



**APLIKASI PENGELOLAAN DISTRIBUSI PRODUK USAHA KECIL  
MENENGAH BERBASIS WEB DENGAN INTEGRASI *CONTENT  
MANAGEMENT SYSTEM* SHOPIFY DI NETASIA SINGAPORE**

**SKRIPSI**

**ARSHA RAULNADI TRIKUSUMA**

**NPM 18670028**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK DAN INFORMATIKA  
UNIVERSITAS PGRI SEMARANG**

**2022**



**APLIKASI PENGELOLAAN DISTRIBUSI PRODUK USAHA KECIL  
MENENGAH BERBASIS WEB DENGAN INTEGRASI *CONTENT  
MANAGEMENT SYSTEM* SHOPIFY DI NETASIA SINGAPORE**

**SKRIPSI**

**Diajukan kepada Fakultas Teknik dan Informatika  
Universitas PGRI Semarang untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer**

**ARSHA RAULNADI TRIKUSUMA**

**NPM 18670028**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK DAN INFORMATIKA  
UNIVERSITAS PGRI SEMARANG**

**2022**

**SKRIPSI**

**APLIKASI PENGELOLAAN DISTRIBUSI PRODUK USAHA KECIL  
MENENGAH BERBASIS WEB DENGAN INTEGRASI *CONTENT  
MANAGEMENT SYSTEM* SHOPIFY DI NETASIA SINGAPORE**

**Disusun dan diajukan oleh**

**ARSHA RAULNADI TRIKUSUMA  
NPM 18670028**

**telah disetujui oleh pembimbing untuk dilanjutkan di  
hadapan Dewan Penguji**

**Pembimbing I,**



**Mega Novita, S.Si., M.Si., M.Nat.Sc., Ph.D.**

**NPP 158801493**

**Semarang, 23 Maret 2022**

**Pembimbing II,**



**Noora Ootrun N., S.T., M.Eng.**

**NPP 158201485**

**SKRIPSI**

**APLIKASI PENGELOLAAN DISTRIBUSI PRODUK USAHA KECIL  
MENENGAH BERBASIS WEB DENGAN INTEGRASI *CONTENT*  
*MANAGEMENT SYSTEM* SHOPIFY DI NETASIA SINGAPORE**

Disusun dan diajukan oleh  
**ARSHA RAULNADI TRIKUSUMA**  
NPM 18670028

telah dipertahankan di depan Dewan Penguji  
pada tanggal 30 Maret 2022  
dan dinyatakan telah memenuhi syarat

**Dewan Penguji**



Dr. Slamet Supriyadi, M.Env.St.  
NIP 195912281986031003

Sekretaris,



Bambang Agus H., S.Kom., M.Kom.  
NPP 148201433

Penguji I,



Bambang Agus H., S.Kom., M.Kom.  
NPP 148201433

Penguji II,



Mega Novita, S.Si., M.Si., M.Nat.Sc., Ph.D.  
NPP 158801493

Penguji III,



Noora Qotrun Nada, S.T., M.Eng.  
NPP 158201485

## MOTO DAN PERSEMBAHAN

### Moto:

1. “Maka sesungguhnya bersama kesulitan pasti ada kemudahan. Sesungguhnya bersama kesulitan pasti ada kemudahan. Maka apabila kamu telah selesai (dari suatu urusan), tetaplah bekerja keras (untuk urusan yang lain), dan hanya kepada Tuhanmulah kamu berharap” (Q.S. Al-Insyirah: 5-8).
2. “Jika kamu tidak sanggup menahan lelahnya belajar maka kamu harus sanggup menahan perihnya kebodohan” (Imam Syafi’i).
3. *Just because you took longer than others, doesn't mean you failed. Because everyone has their own way. So, don't measure your progress using someone else's ruler. It's just about time* – Hanya karena kamu membutuhkan waktu lebih lama dari yang lain, bukan berarti kamu gagal. Karena setiap orang punya caranya masing-masing. Jadi, jangan mengukur kemajuanmu menggunakan “penggaris” orang lain. Semua ini hanya tentang waktu (Penulis).

### Persembahan:

Skripsi ini saya persembahkan untuk:

1. Ayah dan ibuku tercinta
2. Kakak dan adikku tercinta
3. Teman-temanku tercinta
4. Almamater Universitas PGRI Semarang

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Arsha Raulnadi Trikusuma

NPM : 18670028

Progdi : Informatika

Fakultas : Teknik dan Informatika

Menyatakan dengan sebenarnya bahwa skripsi yang saya buat ini benar-benar merupakan hasil karya saya sendiri, bukan plagiarisme.

Apabila pada kemudian hari skripsi ini terbukti hasil plagiarisme, saya bersedia menerima sanksi atas perbuatan tersebut.

Semarang, 23 Maret 2022

Yang membuat pernyataan



Arsha Raulnadi Trikusuma

NPM 18670028

## ABSTRAK

Netasia merupakan merek dari perusahaan Golden Style Management Lte. Ptd. yang berbasis di Singapura dan bertindak sebagai mitra lokal dan distributor untuk Usaha Kecil Menengah (UKM) dari Indonesia. Netasia memanfaatkan *Content Management System* (CMS) Shopify untuk menjangkau *enduser*, namun belum memiliki *platform* pengelolaan pendistribusian produk mulai dari *preorder* kepada vendor hingga distribusi ke toko. Penelitian ini dilakukan untuk mengembangkan aplikasi pengelolaan distribusi produk UKM berbasis web dengan integrasi CMS Shopify di Netasia Singapore. Penelitian ini dilakukan dengan metode *Research and Development* untuk pendekatan penelitian dan metode *prototyping* untuk metode pengembangan sistem. Penggunaan metode *prototyping* sering melibatkan komunikasi antara pengembang dan pengguna aplikasi, sehingga aplikasi beberapa kali mengalami perubahan. Hasil penelitian berhasil mengintegrasikan data perusahaan di CMS Shopify dengan aplikasi yang dikembangkan melalui API dengan menggunakan cURL. Aplikasi diuji dengan beberapa metode pengujian seperti 1) pengujian *black-box* dengan persentase ketercapaian 100% sehingga dinyatakan layak digunakan, 2) pengujian *white-box* dengan kompleksitas siklomatis sebesar 4 serta persentase ketercapaian pengujian nilai 100% sehingga dinyatakan memenuhi kriteria rekayasa perangkat lunak, 3) *User Acceptance Testing* dengan persentase rata-rata 94% sehingga dinyatakan memenuhi harapan pengguna, serta 4) pengujian kompatibilitas aplikasi *browser* dengan hasil secara fungsional dapat dijalankan pada semua aplikasi *browser* yang telah ditentukan, sedangkan dari aspek *user interface* kurang kompatibel dengan Internet Explorer. Saran yang diajukan antara lain perlunya pelatihan penggunaan aplikasi kepada pengguna, aplikasi yang dapat dikembangkan lebih optimal lagi sesuai kebutuhan perusahaan, pengujian pada aplikasi *browser* selain yang digunakan dalam penelitian ini, serta aplikasi yang dapat dikembangkan untuk pengelolaan distribusi produk di perusahaan lain.

Kata Kunci: aplikasi web, *content management system*, Shopify, API, cURL

## PRAKATA

Puji dan syukur penulis panjatkan ke hadirat Allah *subhanahu wata'ala* yang telah melimpahkan rahmat serta hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan lancar dan tepat waktu. Skripsi yang berjudul “Aplikasi Pengelolaan Distribusi Produk Usaha Kecil Menengah Berbasis Web dengan Integrasi *Content Management System* Shopify di Netasia Singapore” ini disusun untuk memenuhi syarat mendapatkan gelar Sarjana Komputer.

Penyusunan skripsi ini tentunya tidak lepas dari berbagai hambatan dan kesulitan. Namun berkat semangat, bimbingan, nasehat, dorongan moril, serta saran-saran dari berbagai pihak, khususnya pembimbing, segala hambatan dan kesulitan dapat teratasi dengan baik. Oleh karena itu, pada kesempatan ini dengan tulus hati penulis sampaikan terima kasih kepada:

1. Dr. Muhti S.H, M.Hum. selaku Rektor Universitas PGRI Semarang yang telah memberikan kesempatan kepada penulis untuk menimba ilmu di Universitas PGRI Semarang.
2. Dr. Drs. Slamet Supriyadi, M.Env.St. selaku Dekan Fakultas Teknik dan Informatika Universitas PGRI Semarang yang telah memberikan izin kepada penulis untuk melakukan penelitian.
3. Bambang Agus Herlambang, S.Kom., M.Kom. selaku Ketua Program Studi Informatika yang telah menyetujui topik skripsi penulis.
4. Mega Novita, S.Si., M.Si., M.Nat.Sc., Ph.D. selaku Pembimbing I yang telah membimbing dan mengarahkan penulis dengan penuh ketekunan dan kecermatan serta dedikasi yang tinggi.
5. Noora Qotrun Nada, S.T., M.Eng. selaku Pembimbing II yang telah membimbing dan mengarahkan penulis dengan penuh ketekunan dan kecermatan serta dedikasi yang tinggi.
6. Bapak dan Ibu Dosen Program Studi Informatika yang telah memberi bekal ilmu kepada penulis selama belajar di Universitas PGRI Semarang.



7. Ir. Sripeni Puspasari, M.Eng. selaku Direktur Golden Style Management Pte. Ltd. Singapura yang telah mengizinkan peneliti melakukan penelitian di perusahaan yang dipimpinnya.
8. Kedua orang tua, Bapak Sutopo dan Ibu Kusmidah serta nenek tersayang, Ibu Marijanah yang selalu memberikan motivasi, semangat, dukungan, serta doa kepada penulis.
9. Aisha Syafriati Ikakusuma, Iasha Syafebi Dwikusuma, dan Resha Meiranadi Caturkusuma, saudara kandung yang selalu membantu dan memberikan semangat kepada penulis.
10. Novia Cecilia Medina, Sukma Maulana Hakim, Dhimas Aria Wardhana, dan Mona Rizqa, sahabat yang selalu membantu dan memberikan semangat kepada penulis.
11. Teman-teman yang tidak dapat penulis sebutkan satu persatu yang senantiasa membantu dan mendukung penulis.
12. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu dalam penyusunan skripsi ini.

Akhirnya skripsi ini dapat tersusun dengan baik sekaligus menjadi salah satu syarat mendapatkan gelar Sarjana Komputer yang penulis raih dengan masa studi selama kurang dari empat tahun. Dengan kerendahan hati, penulis juga berharap agar skripsi ini dapat bermanfaat.

Semarang, 23 Maret 2022

Penulis

## DAFTAR ISI

	Halaman
SAMPUL LUAR .....	i
SAMPUL DALAM.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN .....	iv
MOTO DAN PERSEMBAHAN .....	v
PERNYATAAN KEASLIAN TULISAN .....	vi
ABSTRAK.....	vii
PRAKATA.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR .....	xiv
DAFTAR LAMPIRAN.....	xviii
BAB I PENDAHULUAN.....	1
A. Latar Belakang Masalah .....	1
B. Identifikasi Masalah .....	3
C. Pembatasan Masalah.....	3
D. Perumusan Masalah.....	4
E. Tujuan Penelitian .....	4
F. Manfaat Penelitian .....	4
G. Penegasan Istilah .....	5
H. Sistematika Penulisan Skripsi.....	5
BAB II KAJIAN PUSTAKA/TEORI.....	7
A. Tinjauan Pustaka.....	7
B. Landasan Teori .....	9
C. Kerangka Berpikir .....	24
BAB III METODE PENELITIAN .....	25
A. Pendekatan Penelitian.....	25
B. Lokasi Penelitian .....	25
C. Populasi dan Sampel.....	26
D. Desain Penelitian .....	27
E. Proses Eksperimen.....	115

F. Teknik Pengumpulan Data .....	115
G. Teknik Analisis Data .....	116
BAB IV HASIL DAN PEMBAHASAN .....	118
A. Hasil Pengembangan .....	118
B. Hasil Pengujian.....	145
C. Pembahasan .....	162
BAB V PENUTUP .....	168
A. Simpulan.....	168
B. Saran .....	169
DAFTAR PUSTAKA .....	170
LAMPIRAN.....	173

## DAFTAR TABEL

	Halaman
Tabel 2.1. Tinjauan Pustaka .....	7
Tabel 2.2. Simbol pada <i>Use Case Diagram</i> .....	15
Tabel 2.3. Simbol pada <i>Activity Diagram</i> .....	17
Tabel 2.4. Simbol pada <i>Class Diagram</i> .....	18
Tabel 2.5. Simbol pada <i>Sequence Diagram</i> .....	19
Tabel 3.1. Kebutuhan Nonfungsional Sistem .....	30
Tabel 3.2. Skenario <i>Use Case Login</i> .....	34
Tabel 3.3. Skenario <i>Use Case Admin Melihat Data Vendor</i> .....	35
Tabel 3.4. Skenario <i>Use Case Admin Menambah Akun Vendor</i> .....	36
Tabel 3.5. Skenario <i>Use Case Admin Mengedit Akun Vendor</i> .....	37
Tabel 3.6. Skenario <i>Use Case Admin Melihat Data Toko</i> .....	37
Tabel 3.7. Skenario <i>Use Case Admin Menambah Data Toko</i> .....	38
Tabel 3.8. Skenario <i>Use Case Admin Mengedit Data Toko</i> .....	39
Tabel 3.9. Skenario <i>Use Case Admin Menghapus Data Toko</i> .....	40
Tabel 3.10. Skenario <i>Use Case Admin Melihat Data Admin</i> .....	40
Tabel 3.11. Skenario <i>Use Case Admin Menambah Data Admin</i> .....	41
Tabel 3.12. Skenario <i>Use Case Admin Mengedit Data Admin</i> .....	41
Tabel 3.13. Skenario <i>Use Case Admin Menghapus Data Admin</i> .....	42
Tabel 3.14. Skenario <i>Use Case Admin Menerima Preorder</i> .....	43
Tabel 3.15. Skenario <i>Use Case Admin Mengirim Preorder</i> .....	44
Tabel 3.16. Skenario <i>Use Case Admin Melihat Riwayat Preorder</i> .....	45
Tabel 3.17. Skenario <i>Use Case Pengaturan Akun Admin</i> .....	45
Tabel 3.18. Skenario <i>Use Case Vendor Melihat Data Produk</i> .....	46
Tabel 3.19. Skenario <i>Use Case Vendor Menerima Preorder</i> .....	46
Tabel 3.20. Skenario <i>Use Case Vendor Melihat Riwayat Preorder</i> .....	47
Tabel 3.21. Skenario <i>Use Case Pengaturan Akun Vendor</i> .....	48
Tabel 3.22. Skenario <i>Use Case Toko Mengirim Preorder</i> .....	48
Tabel 3.23. Skenario <i>Use Case Toko Melihat Riwayat Preorder</i> .....	49
Tabel 3.24. Skenario <i>Use Case Pengaturan Akun Toko</i> .....	50
Tabel 3.25. Tabel <i>Database Pengguna</i> .....	73
Tabel 3.26. Tabel <i>Database Keranjang</i> .....	74

Tabel 3.27. Tabel <i>Database</i> Prapemesanan .....	75
Tabel 3.28. Formulir Pengujian <i>Black-box</i> Halaman <i>Login</i> .....	97
Tabel 3.29. Formulir Pengujian <i>Black-box</i> Halaman Admin.....	98
Tabel 3.30. Formulir Pengujian <i>Black-box</i> Halaman Vendor.....	105
Tabel 3.31. Formulir Pengujian <i>Black-box</i> Halaman Toko .....	107
Tabel 3.32. Formulir <i>User Acceptance Testing</i> (UAT) .....	112
Tabel 3.33. Bobot Penilaian <i>User Acceptance Testing</i> (UAT).....	114
Tabel 3.34. Formulir Pengujian Kompatibilitas Aplikasi <i>Browser</i> .....	115
Tabel 4.1. Hasil Pengujian <i>Black-box</i> Halaman <i>Login</i> .....	145
Tabel 4.2. Hasil Pengujian <i>Black-box</i> Halaman Admin .....	146
Tabel 4.3. Hasil Pengujian <i>Black-box</i> Halaman Vendor .....	151
Tabel 4.4. Hasil Pengujian <i>Black-box</i> Halaman Toko .....	153
Tabel 4.5. Hasil Pengujian <i>White-box</i> .....	155
Tabel 4.6. Hasil <i>User Acceptance Testing</i> (UAT) .....	160
Tabel 4.7. Hasil Pengujian Kompatibilitas Aplikasi <i>Browser</i> .....	162
Tabel 4.8. Hasil Pengujian Nilai pada Pengujian <i>White-box</i> .....	165

## DAFTAR GAMBAR

	Halaman
Gambar 1.1. Proses Pengelolaan Distribusi Produk di Netasia .....	2
Gambar 2.1. Kerangka Berpikir .....	24
Gambar 3.1. Metode <i>Prototyping</i> .....	27
Gambar 3.2. <i>Use Case Diagram</i> .....	32
Gambar 3.3. <i>Sequence Diagram Login</i> .....	52
Gambar 3.4. <i>Sequence Diagram Admin Melihat Data Vendor</i> .....	52
Gambar 3.5. <i>Sequence Diagram Admin Menambah Akun Vendor</i> .....	52
Gambar 3.6. <i>Sequence Diagram Admin Mengedit Akun Vendor</i> .....	53
Gambar 3.7. <i>Sequence Diagram Admin Melihat Data Toko</i> .....	53
Gambar 3.8. <i>Sequence Diagram Admin Menambah Data Toko</i> .....	53
Gambar 3.9. <i>Sequence Diagram Admin Mengedit Data Toko</i> .....	54
Gambar 3.10. <i>Sequence Diagram Admin Menghapus Data Toko</i> .....	54
Gambar 3.11. <i>Sequence Diagram Admin Melihat Data Admin</i> .....	54
Gambar 3.12. <i>Sequence Diagram Admin Menambah Data Admin</i> .....	55
Gambar 3.13. <i>Sequence Diagram Admin Mengedit Data Admin</i> .....	55
Gambar 3.14. <i>Sequence Diagram Admin Menghapus Data Admin</i> .....	55
Gambar 3.15. <i>Sequence Diagram Admin Menerima Preorder</i> .....	56
Gambar 3.16. <i>Sequence Diagram Admin Mengirim Preorder</i> .....	56
Gambar 3.17. <i>Sequence Diagram Admin Melihat Riwayat Preorder</i> .....	56
Gambar 3.18. <i>Sequence Diagram Pengaturan Akun Admin</i> .....	57
Gambar 3.19. <i>Sequence Diagram Vendor Melihat Data Produk</i> .....	57
Gambar 3.20. <i>Sequence Diagram Vendor Menerima Preorder</i> .....	57
Gambar 3.21. <i>Sequence Diagram Vendor Melihat Riwayat Preorder</i> .....	58
Gambar 3.22. <i>Sequence Diagram Pengaturan Akun Vendor</i> .....	58
Gambar 3.23. <i>Sequence Diagram Toko Mengirim Preorder</i> .....	58
Gambar 3.24. <i>Sequence Diagram Toko Melihat Riwayat Preorder</i> .....	59
Gambar 3.25. <i>Sequence Diagram Pengaturan Akun Toko</i> .....	59
Gambar 3.26. <i>Activity Diagram Login</i> .....	60
Gambar 3.27. <i>Activity Diagram Admin Melihat Data Vendor</i> .....	61
Gambar 3.28. <i>Activity Diagram Admin Menambahkan Akun Vendor</i> .....	61
Gambar 3.29. <i>Activity Diagram Admin Mengedit Akun Vendor</i> .....	62

Gambar 3.30. <i>Activity Diagram</i> Admin Melihat Data Toko.....	62
Gambar 3.31. <i>Activity Diagram</i> Admin Menambahkan Data Toko .....	63
Gambar 3.32. <i>Activity Diagram</i> Admin Mengedit Data Toko.....	63
Gambar 3.33. <i>Activity Diagram</i> Admin Menghapus Data Toko .....	64
Gambar 3.34. <i>Activity Diagram</i> Admin Melihat Data Admin.....	64
Gambar 3.35. <i>Activity Diagram</i> Admin Menambah Data Admin .....	65
Gambar 3.36. <i>Activity Diagram</i> Admin Mengedit Data Admin .....	65
Gambar 3.37. <i>Activity Diagram</i> Admin Menghapus Data Admin.....	66
Gambar 3.38. <i>Activity Diagram</i> Admin Menerima <i>Preorder</i> .....	66
Gambar 3.39. <i>Activity Diagram</i> Admin Mengirim <i>Preorder</i> .....	67
Gambar 3.40. <i>Activity Diagram</i> Admin Melihat Riwayat <i>Preorder</i> .....	67
Gambar 3.41. <i>Activity Diagram</i> Pengaturan Akun Admin .....	68
Gambar 3.42. <i>Activity Diagram</i> Vendor Melihat Data Produk.....	68
Gambar 3.43. <i>Activity Diagram</i> Vendor Menerima <i>Preorder</i> .....	69
Gambar 3.44. <i>Activity Diagram</i> Vendor Melihat Riwayat <i>Preorder</i> .....	69
Gambar 3.45. <i>Activity Diagram</i> Pengaturan Akun Vendor .....	70
Gambar 3.46. <i>Activity Diagram</i> Toko Mengirim <i>Preorder</i> .....	70
Gambar 3.47. <i>Activity Diagram</i> Toko Melihat Riwayat <i>Preorder</i> .....	71
Gambar 3.48. <i>Activity Diagram</i> Pengaturan Akun Toko.....	71
Gambar 3.49. <i>Class Diagram</i> .....	72
Gambar 3.50. Desain Halaman <i>Login</i> .....	76
Gambar 3.51. Desain Halaman Dasbor Admin.....	77
Gambar 3.52. Desain Halaman Daftar Vendor di Admin.....	77
Gambar 3.53. Desain Halaman Detail Vendor di Admin .....	78
Gambar 3.54. Desain Halaman Daftar Toko di Admin .....	79
Gambar 3.55. Desain Halaman Detail Toko di Admin.....	79
Gambar 3.56. Desain Tab <i>Preorder</i> dari Toko .....	80
Gambar 3.57. Desain Tab <i>Preorder</i> ke Vendor .....	81
Gambar 3.58. Desain Halaman Proses <i>Preorder</i> dari Toko .....	81
Gambar 3.59. Desain Halaman Proses <i>Preorder</i> ke Vendor .....	82
Gambar 3.60. Desain Tab Riwayat <i>Preorder</i> dari Toko.....	83
Gambar 3.61. Desain Tab Riwayat <i>Preorder</i> ke Vendor.....	83
Gambar 3.62. Desain Halaman Detail <i>Preorder</i> dari Toko .....	84
Gambar 3.63. Desain Halaman Detail <i>Preorder</i> ke Vendor .....	85

Gambar 3.64. Desain Halaman Daftar Admin.....	85
Gambar 3.65. Desain Halaman Pengaturan Akun Admin .....	86
Gambar 3.66. Desain Halaman Dasbor Vendor.....	87
Gambar 3.67. Desain Halaman Daftar Produk di Vendor .....	87
Gambar 3.68. Desain Halaman <i>Preorder</i> di Vendor .....	88
Gambar 3.69. Desain Halaman Proses <i>Preorder</i> di Vendor .....	89
Gambar 3.70. Desain Halaman Riwayat <i>Preorder</i> di Vendor .....	89
Gambar 3.71. Desain Halaman Detail <i>Preorder</i> di Vendor.....	90
Gambar 3.72. Desain Halaman Pengaturan Akun Vendor .....	91
Gambar 3.73. Desain Halaman Dasbor Toko .....	91
Gambar 3.74. Desain Halaman <i>Preorder</i> di Toko.....	92
Gambar 3.75. Desain Halaman Daftar Pilihan Produk di Toko.....	93
Gambar 3.76. Desain Halaman Detail Produk di Toko .....	93
Gambar 3.77. Desain Halaman Keranjang di Toko .....	94
Gambar 3.78. Desain Halaman Riwayat <i>Preorder</i> di Toko.....	94
Gambar 3.79. Desain Halaman Detail <i>Preorder</i> di Toko .....	95
Gambar 3.80. Desain Halaman Pengaturan Akun Toko .....	96
Gambar 4.1. Tampilan (a) Sebelum Revisi dan (b) Sesudah Revisi.....	121
Gambar 4.2. Ilustrasi Proses Integrasi Data Melalui API.....	122
Gambar 4.3. Tampilan Halaman <i>Login</i> .....	123
Gambar 4.4. Tampilan Halaman Dasbor Admin .....	124
Gambar 4.5. Tampilan Halaman Daftar Vendor di Admin.....	125
Gambar 4.6. Tampilan Halaman Detail Vendor di Admin .....	125
Gambar 4.7. Tampilan Halaman Daftar Toko di Admin .....	126
Gambar 4.8. Tampilan Halaman Detail Toko di Admin.....	127
Gambar 4.9. Tampilan Tab <i>Preorder</i> dari Toko.....	127
Gambar 4.10. Tampilan Tab <i>Preorder</i> ke Vendor.....	128
Gambar 4.11. Tampilan Halaman Proses <i>Preorder</i> dari Toko .....	129
Gambar 4.12. Tampilan Halaman Proses <i>Preorder</i> ke Vendor .....	129
Gambar 4.13. Tampilan Tab Riwayat <i>Preorder</i> dari Toko.....	130
Gambar 4.14. Tampilan Tab Riwayat <i>Preorder</i> ke Vendor .....	131
Gambar 4.15. Tampilan Halaman Detail <i>Preorder</i> dari Toko.....	132
Gambar 4.16. Tampilan Halaman Detail <i>Preorder</i> Ke Vendor.....	133
Gambar 4.17. Tampilan Halaman Daftar Admin.....	133



Gambar 4.18. Tampilan Halaman Pengaturan Akun Admin .....	134
Gambar 4.19. Tampilan Halaman Dasbor Vendor .....	135
Gambar 4.20. Tampilan Halaman Daftar Produk di Vendor .....	135
Gambar 4.21. Tampilan Halaman <i>Preorder</i> di Vendor .....	136
Gambar 4.22. Tampilan Halaman Proses <i>Preorder</i> di Vendor.....	137
Gambar 4.23. Tampilan Halaman Riwayat <i>Preorder</i> di Vendor.....	138
Gambar 4.24. Tampilan Halaman Detail <i>Preorder</i> di Vendor .....	138
Gambar 4.25. Tampilan Halaman Pengaturan Akun Vendor .....	139
Gambar 4.26. Tampilan Halaman Dasbor Toko .....	140
Gambar 4.27. Tampilan Halaman <i>Preorder</i> di Toko.....	140
Gambar 4.28. Tampilan Halaman Daftar Pilihan Produk di Toko .....	141
Gambar 4.29. Tampilan Halaman Detail Produk di Toko .....	141
Gambar 4.30. Tampilan Halaman Keranjang di Toko .....	142
Gambar 4.31. Tampilan Halaman Riwayat <i>Preorder</i> di Toko .....	143
Gambar 4.32. Tampilan Halaman Detail <i>Preorder</i> di Toko .....	144
Gambar 4.33. Tampilan Halaman Pengaturan Akun Toko.....	144
Gambar 4.34. Pengujian Berbasis Alur pada Pengujian <i>White-box</i> .....	164

## DAFTAR LAMPIRAN

Lampiran 1. Surat Permohonan Izin Penelitian .....	173
Lampiran 2. Lembar Pembimbingan Skripsi Pembimbing 1.....	174
Lampiran 3. Lembar Pembimbingan Skripsi Pembimbing 2.....	176
Lampiran 4. Formulir Pengujian <i>Black-box</i> Responden 1 .....	178
Lampiran 5. Formulir Pengujian <i>Black-box</i> Responden 2.....	187
Lampiran 6. Formulir Pengujian <i>Black-box</i> Responden 3.....	196
Lampiran 7. Formulir <i>User Acceptance Testing</i> (UAT) Responden 1 .....	205
Lampiran 8. Formulir <i>User Acceptance Testing</i> (UAT) Responden 2 .....	207
Lampiran 9. Formulir <i>User Acceptance Testing</i> (UAT) Responden 3 .....	209
Lampiran 10. Bukti Pengujian Kompatibilitas Aplikasi <i>Browser</i> .....	211
Lampiran 11. Dokumentasi Penelitian.....	213
Lampiran 12. Lembar Revisi Ujian Skripsi Penguji 1 .....	214
Lampiran 13. Lembar Revisi Ujian Skripsi Penguji 2 .....	215
Lampiran 14. Lembar Revisi Ujian Skripsi Penguji 3.....	216
Lampiran 15. Biodata Penulis.....	217

# BAB I PENDAHULUAN

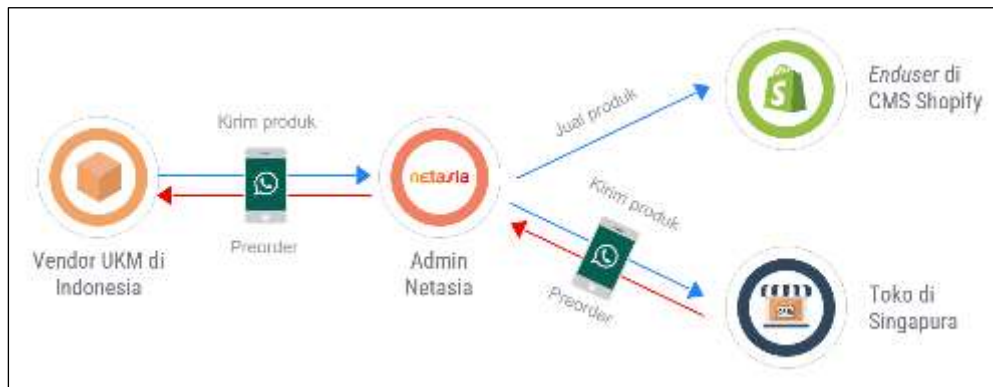
## A. Latar Belakang Masalah

Netasia atau yang dikenal dengan Netasia Singapore merupakan salah satu merek (*brand activation*) dari perusahaan Golden Style Management Lte. Ptd. yang berbasis di Singapura. Netasia merupakan mitra lokal dan distributor untuk Usaha Kecil Menengah (UKM) dari Indonesia di Singapura. Produk yang didistribusikan juga menyesuaikan kebutuhan dan tren pasar di Singapura. Netasia menjaga produk mulai dari kedatangan ke toko, memantau penjualan, dan menembus ke tempat baru atau target pasar baru. Netasia bertindak melakukan prapemesanan (*preorder*) kepada vendor UKM yang ada di Indonesia, kemudian mendistribusikannya baik kepada pelanggan akhir (*enduser*) maupun kepada beberapa toko di Singapura yang telah bekerja sama dengan Netasia.

Sebagai upaya mengoptimalkan pendistribusian produk UKM, Netasia sedang memulai menjangkau *enduser* dengan memanfaatkan *Content Management System* (CMS) bernama Shopify. *Content Management System* (CMS) merupakan sebuah sistem yang digunakan untuk memudahkan pengguna dalam mengelola konten web secara lebih mudah tanpa banyak melakukan pemrograman [1]. Pengguna hanya mengunggah konten ke CMS dan melakukan konfigurasi melalui fitur-fitur yang disediakan CMS. Sementara itu Shopify merupakan salah satu CMS berbentuk *e-commerce* yang digunakan sebagai tempat jual beli secara *online*. Netasia menggunakan CMS Shopify tersebut untuk menjual beberapa produk dari vendor UKM secara *online* kepada siapa saja sebagai *enduser*.

Selain menjual produk kepada *enduser*, Netasia juga perlu mengelola pendistribusian produk UKM mulai dari *preorder* kepada vendor sampai dengan distribusi kepada toko di Singapura. Namun saat ini Netasia belum memiliki *platform* untuk mengelola pendistribusian produk UKM tersebut. Pengelolaan yang dilakukan Netasia saat ini masih dilakukan secara manual

menggunakan aplikasi komunikasi seperti WhatsApp dan pencatatan manual menggunakan Google Spreadsheet atau Microsoft Excel. Hal ini dapat menyebabkan adanya penumpukan berkas yang berpotensi mempersulit pengelolaan distribusi produk. Proses pengelolaan distribusi produk UKM di Netasia yang dilakukan selama ini diilustrasikan pada Gambar 1.1.



Gambar 1.1. Proses Pengelolaan Distribusi Produk di Netasia

Salah satu cara yang dapat digunakan untuk mengelola distribusi produk secara lebih mudah adalah dengan membuat sebuah aplikasi berbasis web. Aplikasi berbasis web merupakan aplikasi yang berjalan pada *web server* dan dapat diakses menggunakan *web browser* [2]. Berbeda dengan *website*, aplikasi berbasis web lebih interaktif dan responsif terhadap tindakan pengguna [3]. Pengelolaan distribusi produk akan lebih terorganisir jika memanfaatkan aplikasi berbasis web. Proses *preorder* dan distribusi produk dapat dilakukan dengan mudah dan cepat, serta data dan riwayat distribusi produk dapat tersimpan dengan baik.

Aplikasi berbasis web yang digunakan untuk mengelola distribusi produk harus dapat dioperasikan secara lebih efektif dan efisien dibandingkan dengan menggunakan cara manual. Salah satunya adalah mengintegrasikan data produk yang telah dimiliki Netasia di Shopify dengan aplikasi berbasis web yang dikembangkan agar tidak perlu melakukan pengunggahan ulang data produk. Integrasi data antara Shopify dengan aplikasi berbasis web dapat dilakukan karena Shopify sebagai CMS menyediakan layanan web (*web service*) yang memungkinkan aplikasi lain dapat mengakses beberapa fitur

yang dimilikinya [4]. Salah satu fitur yang dapat diakses oleh aplikasi lain adalah data produk yang dijual di Shopify.

Data produk yang ada pada Shopify dapat diintegrasikan ke aplikasi lain dengan memanfaatkan *Application Programming Interface* (API) milik Shopify menggunakan *client URL* (cURL). *Application Programming Interface* (API) merupakan sebuah *library* yang menjadi perantara sebuah aplikasi untuk berinteraksi dengan aplikasi lain [5]. Sedangkan *client URL* (cURL) merupakan baris perintah untuk mentransfer data dari dan ke *server* menggunakan sintaks URL melalui bermacam-macam protokol [6]. Shopify menyediakan prosedur untuk mengakses API tersebut.

Berdasarkan kondisi yang dihadapi oleh Netasia, maka diperlukan suatu *platform* yang dapat digunakan untuk mengelola pendistribusian produk UKM mulai dari *preorder* kepada vendor sampai dengan distribusi kepada toko di Singapura. Oleh karena itu, dalam penelitian ini dilakukan pengembangan sebuah aplikasi pengelolaan distribusi produk UKM berbasis web dengan integrasi CMS Shopify di Netasia Singapore. Diharapkan aplikasi berbasis web ini dapat membantu Netasia dalam mengelola distribusi produk UKM secara lebih efektif dan efisien.

## **B. Identifikasi Masalah**

Berdasarkan latar belakang masalah yang telah diuraikan, maka dapat diidentifikasi beberapa permasalahan sebagai berikut:

1. Pengelolaan distribusi produk di Netasia masih dilakukan secara manual menggunakan aplikasi komunikasi seperti WhatsApp dan pencatatan manual menggunakan Google Spreadsheet atau Microsoft Excel.
2. Belum adanya *platform* atau sistem pengelolaan distribusi produk yang lebih efektif dan efisien.

## **C. Pembatasan Masalah**

Berdasarkan identifikasi masalah yang telah diuraikan, maka penelitian ini hanya dibatasi pada pengembangan aplikasi berbasis web dengan integrasi

*Content Management System* Shopify sesuai dengan prosedur pengelolaan distribusi produk Usaha Kecil Menengah yang ada di Netasia Singapore, tanpa menggunakan prosedur pengelolaan distribusi produk di tempat lain.

#### **D. Perumusan Masalah**

Berdasarkan identifikasi masalah dari latar belakang yang telah diuraikan, maka rumusan masalah dari penelitian ini adalah “Bagaimana pengembangan aplikasi pengelolaan distribusi produk Usaha Kecil Menengah berbasis web dengan integrasi *Content Management System* Shopify di Netasia Singapore?”.

#### **E. Tujuan Penelitian**

Berdasarkan perumusan masalah yang telah diuraikan, maka tujuan dari penelitian ini adalah untuk mengembangkan aplikasi pengelolaan distribusi produk Usaha Kecil Menengah berbasis web dengan integrasi *Content Management System* Shopify di Netasia Singapore.

#### **F. Manfaat Penelitian**

Adapun manfaat yang didapat dari penelitian ini antara lain diuraikan sebagai berikut:

1. Secara teoretis/akademis, hasil penelitian ini diharapkan dapat menambah wawasan dan memperkaya khazanah kepustakaan akademik mengenai pengembangan aplikasi berbasis web terintegrasi *Content Management System* Shopify, dengan subjek atau objek penelitian yang lebih beragam maupun dengan penambahan fitur pada aplikasi.
2. Secara praktis, bagi perusahaan hasil penelitian ini diharapkan dapat memberikan solusi dalam mempermudah pengelolaan distribusi produk secara lebih efektif dan efisien dibanding menggunakan pengelolaan manual. Sementara bagi peneliti, hasil penelitian ini diharapkan dapat menjadi referensi dalam mengembangkan penelitian selanjutnya dengan fitur aplikasi maupun subjek atau objek penelitian yang berbeda.

## G. Penegasan Istilah

Istilah yang perlu ditegaskan dalam penelitian ini antara lain diuraikan sebagai berikut:

### 1. Aplikasi berbasis web

Aplikasi berbasis web merupakan aplikasi yang berjalan pada *web server* dan dapat diakses menggunakan *web browser* [2]. Berbeda dengan *website*, aplikasi berbasis web lebih interaktif dan responsif terhadap tindakan pengguna [3].

### 2. Pengelolaan distribusi produk

Pengelolaan distribusi produk yang dimaksud dalam penelitian ini adalah proses pencatatan dan pengolahan data distribusi produk mulai dari prapemesanan kepada vendor, sampai dengan distribusi kepada toko.

### 3. *Content Management System (CMS) Shopify*

*Content Management System (CMS)* merupakan sebuah sistem yang digunakan untuk memudahkan pengguna dalam mengelola konten web secara lebih mudah tanpa banyak melakukan pemrograman [1]. Sedangkan *Shopify* adalah salah satu CMS yang berbentuk *e-commerce* atau *platform* jual beli secara *online*.

## H. Sistematika Penulisan Skripsi

Skripsi ini disusun dengan sistematika penulisan yang secara garis besar terdiri dari dua bagian yakni bagian awal dan bagian inti. Bagian awal secara berurutan terdiri dari sampul luar, sampul dalam, halaman persetujuan, halaman pengesahan, halaman moto dan persembahan, abstrak, prakata, daftar isi, daftar tabel, daftar gambar, dan daftar lampiran. Sementara itu bagian inti secara berurutan terdiri dari Bab I Pendahuluan, Bab II Kajian Pustaka/Teori, Bab III Metode Penelitian, Bab IV Hasil dan Pembahasan, Bab V Penutup, daftar pustaka, dan lampiran.

Bab I Pendahuluan membahas tentang latar belakang masalah, identifikasi masalah, pembatasan masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, penegasan istilah, dan sistematika penulisan

skripsi. Bab II Kajian Pustaka/Teori berisi tinjauan pustaka, landasan teori yang berisi beberapa teori yang digunakan dalam penelitian ini, dan kerangka berpikir. Bab III Metode Penelitian membahas tentang pendekatan penelitian, lokasi/fokus penelitian, populasi dan sampel, desain penelitian yang berisi tahap-tahap pada metode pengembangan aplikasi, proses eksperimen, teknik pengumpulan data, dan teknik analisis data. Bab IV Hasil dan Pembahasan membahas tentang hasil pengembangan, hasil pengujian, dan pembahasan. Bab V Penutup berisi simpulan dan saran.



**BAB II**  
**KAJIAN PUSTAKA/TEORI**

**A. Tinjauan Pustaka**

Penulis melakukan tinjauan pustaka berdasarkan beberapa penelitian sejenis yang pernah dilakukan sebelumnya. Beberapa penelitian sejenis yang digunakan penulis sebagai referensi diuraikan dalam Tabel 2.1.

Tabel 2.1. Tinjauan Pustaka

No	Nama Peneliti dan Tahun	Judul	Hasil
1	Anthonius Hendriyanto dan Doro Edi Tahun 2021	Integrasi Data Toko-pedia dengan <i>Content Management Systems Point of Sales</i>	Data produk yang ada di Tokopedia dapat terintegrasi ke aplikasi yang bernama <i>Point of Sales</i> sehingga pemilik toko tidak perlu merekap ulang data.
2	Raka Yusuf, Harni Kusniyati, Rani Eva Dewi Tahun 2015	Aggregator Otomatis Pencari Produk dengan Pemberitahuan Melalui Surel Menggunakan Fungsi cURL	Aplikasi pencari produk pada bisnis <i>online</i> yang melibatkan konsumen, produsen, penyedia layanan, dan pedagang perantara melalui jaringan Internet dengan menggunakan fungsi cURL untuk mentransfer data dari dan menuju <i>server</i> .

3	Yusup Maulanadireja Tahun 2020	Optimalisasi Distri- busi Barang PT. XYZ Menggunakan <i>Hill Climbing</i> Berbasis Android	Aplikasi bernama Gasku yang digunakan untuk membuat pemesanan gas secara <i>online</i> , membuat sebuah <i>database</i> untuk mengintegrasikan data, menggunakan metode <i>hill climbing</i> untuk dapat mengetahui jalur yang optimal dengan teknologi GPS ( <i>Global Positioning System</i> ), serta penggunaan Mapbox untuk mendukung navigasi dan peta, melalui API.
---	--------------------------------------	---	---

Menurut penelitian yang dilakukan Hendriyanto dkk. (2021) dengan judul “Integrasi Data Tokopedia dengan *Content Management Systems Point of Sales*”, dihasilkan sebuah integrasi antara data yang ada di Tokopedia dengan CMS bernama *Point of Sales* (POS). Penerapan integrasi tersebut menjadikan data-data manual seperti hasil penjualan produk, harga produk, dan lain-lain menjadi data digital yang bisa dilihat oleh penjual atau pemilik toko tanpa harus khawatir kehilangan data tersebut. Selain itu dengan adanya fitur *import order* maka penjual atau pemilik toko tidak perlu repot-repot untuk merekap ulang, dan hanya perlu memasukkan *file* penjualan Tokopedia ke CMS POS agar data menjadi satu [1].

Menurut penelitian yang dilakukan Yusuf dkk. (2015) dengan judul “Aggregator Otomatis Pencari Produk dengan Pemberitahuan Melalui Surel Menggunakan Fungsi cURL”, dihasilkan sebuah aplikasi pencari produk pada bisnis online yang melibatkan konsumen, produsen, penyedia layanan, dan pedagang perantara melalui jaringan Internet. Aplikasi ini mengambil data di

situs lain secara otomatis, kemudian menggunakan fungsi cURL untuk mentransfer data dari dan menuju *server*. Aplikasi ini juga dapat mencari dan memberitahukan pengguna via surel (surat elektronik) jika ada produk yang dicarinya. Keberadaan aplikasi ini membuat pengguna juga tidak perlu lagi untuk mencari produk di berbagai situs perdagangan secara satu per satu [6].

Menurut penelitian yang dilakukan Maulanadireja (2020) dengan judul “Optimalisasi Distribusi Barang PT. XYZ Menggunakan *Hill Climbing* Berbasis Android”, dihasilkan aplikasi bernama Gasku yang digunakan untuk membuat pemesanan gas secara *online*, membuat sebuah *database* untuk mengintegrasikan data, menggunakan metode *hill climbing* untuk dapat mengetahui jalur yang optimal dengan teknologi GPS (*Global Positioning System*), serta penggunaan Mapbox untuk mendukung navigasi dan peta. Beberapa fitur yang ada di aplikasi tersebut mengambil data melalui API [5].

Tinjauan pustaka yang digunakan dalam penelitian ini antara lain membahas tentang integrasi CMS, penggunaan fungsi cURL, dan penggunaan API dalam optimalisasi distribusi produk. Berdasarkan tinjauan pustaka yang diuraikan, maka peneliti mengembangkan aplikasi pengelolaan distribusi produk Usaha Kecil Menengah berbasis web dengan integrasi *Content Management System* Shopify di Netasia Singapore. Data perusahaan yang berada di CMS Shopify diintegrasikan dengan aplikasi pengelolaan distribusi produk berbasis web melalui API dengan menggunakan cURL.

## **B. Landasan Teori**

### **1. Aplikasi berbasis web**

Aplikasi berbasis web merupakan perangkat lunak yang berjalan pada *server* web dan dapat diakses menggunakan web *browser* [2]. Keunggulan aplikasi berbasis web ini antara lain dapat dijalankan di berbagai sistem operasi, dapat dijalankan tanpa proses instalasi, serta dapat dijalankan kapan saja dan di mana saja [7]. Perbedaan antara *website* dengan aplikasi berbasis web adalah, jika *website* bersifat informatif, maka aplikasi berbasis web bersifat interaktif. Aplikasi

berbasis web hadir dengan beberapa fitur tambahan yang dapat digunakan pengguna untuk berinteraksi di dalam *website*, sehingga aplikasi berbasis web lebih responsif terhadap tindakan pengguna [3]. Kelebihan menggunakan aplikasi berbasis web salah satunya adalah dapat dioperasikan di berbagai perangkat maupun sistem operasi karena aplikasi berjalan di web yang dapat diakses melalui *browser*.

## 2. *Content Management System (CMS)*

*Content Management System (CMS)* atau sistem manajemen konten merupakan sebuah sistem yang digunakan untuk memudahkan pengguna dalam mengelola konten web secara lebih mudah tanpa banyak melakukan pemrograman [1]. Pengguna hanya mengunggah konten web melalui CMS dan melakukan konfigurasi dengan fitur-fitur yang disediakan CMS. Kemampuan atau fitur dari sebuah CMS berbeda-beda tergantung dari pengembang CMS itu sendiri. Meskipun begitu, kebanyakan dari CMS memiliki fitur publikasi berbasis web, manajemen format, kontrol revisi, pembuatan indeks, pencarian, dan pengarsipan [8]. Setiap CMS juga memiliki jenis yang berbeda-beda tergantung pada kegunaannya antara lain seperti *website* profil, blog, toko *online (e-commerce)*, pembelajaran (*e-learning*), dan lain-lain.

## 3. *Shopify*

Shopify merupakan salah satu CMS berbentuk *e-commerce* yang digunakan sebagai tempat jual beli *online*. Secara teknis Shopify adalah *e-commerce web builder* yang berguna untuk membuat toko online, mulai dari manajemen produk, inventaris harga, pembayaran, hingga pengiriman. Shopify cocok untuk pemilik bisnis yang ingin membuat toko *online*, tanpa harus banyak melakukan pemrograman. Keuntungan penggunaan Shopify dibandingkan dengan CMS *e-commerce* yang lain antara lain mudah digunakan, *template* yang menarik, tampilan yang responsif terhadap layar perangkat, serta pilihan fitur yang cukup lengkap. Shopify menawarkan beberapa pilihan paket dengan harga yang

beragam berdasarkan fungsionalitasnya, mulai dari *Shopify Basic*, *Shopify Standard*, *Shopify Advanced*, hingga *Shopify Plus* [9].

#### 4. **Application Programming Interface (API)**

*Application Programming Interface* (API) merupakan perangkat lunak antarmuka (*software interface*) yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk *library* dan menjelaskan bagaimana agar suatu aplikasi dapat berinteraksi dengan aplikasi lain [5]. Jadi ketika sebuah aplikasi ingin mengakses data dari aplikasi lain, maka dapat menggunakan API yang dimiliki oleh aplikasi yang akan diambil datanya. Namun tidak semua aplikasi menyediakan API agar datanya dapat diakses aplikasi lain. Bahkan juga tidak semua data pada aplikasi dapat diakses menggunakan API.

#### 5. **Client Uniform Resource Locators (cURL)**

*Client Uniform Resource Locators* (cURL) yang dikembangkan oleh Daniel Stanberg pada tahun 1998 merupakan baris perintah untuk mentransfer data dari dan ke *server* menggunakan sintaks URL melalui bermacam-macam protokol seperti FTP, HTTP, HTTPS, TELNET, dan sebagainya [6]. Sebenarnya ada banyak perangkat transfer data layaknya cURL. Namun cURL memiliki fitur yang lebih lengkap di antara perangkat-perangkat lainnya. Terdapat empat tahapan ketika menggunakan cURL, antara lain inisialisasi, *set option*, *execute*, serta *close* [10]. Sementara itu penggunaan cURL berada di dalam bahasa pemrograman PHP atau pada sisi *server* (*backend*).

#### 6. **Framework Codeigniter**

*Framework* adalah kerangka kerja untuk mengembangkan aplikasi berbasis *website* maupun *desktop*. Kerangka kerja di sini sangat membantu pengembangan web dengan lebih terstruktur dan tersusun rapi. Terdapat berbagai macam *framework* yang dapat digunakan dalam membangun sebuah web, salah satunya adalah Codeigniter. Codeigniter adalah *framework* aplikasi web yang bersifat *open source* dan digunakan untuk membangun aplikasi PHP yang dinamis. Tujuan utamanya adalah

untuk membantu pengembangan aplikasi secara lebih cepat daripada menulis semua kode mulai dari nol [11]. Kelebihan menggunakan *framework* Codeigniter antara lain mudah dipelajari dan digunakan, ukuran relatif kecil dibandingkan dengan *framework* lain, serta tidak terlalu banyak konfigurasi.

#### 7. ***HyperText Markup Language (HTML)***

*HyperText Markup Language (HTML)* adalah sintaks atau bahasa *markup* yang digunakan untuk membuat tampilan dari sebuah *website*. HTML berbentuk berkas teks yang ditulis menggunakan aturan-aturan kode tertentu (sintaks) untuk kemudian disajikan ke pengguna melalui suatu *browser*. HTML inilah yang akan dieksekusi oleh *browser* dan ditampilkan di jendela *browser* [12]. HTML menjadi kerangka dasar sebuah web, dan menjadi penyusun struktur sebuah web. Berbeda dengan bahasa pemrograman, HTML hanyalah sintaks yang tidak dapat menjalankan logika dan aritmetika.

#### 8. ***Cascading Style Sheets (CSS)***

*Cascading Style Sheets (CSS)* adalah sebuah dokumen yang berisi sintaks yang digunakan untuk mengatur beberapa komponen dalam halaman-halaman web sehingga lebih terstruktur [12]. CSS dapat mengatur tampilan atau visualisasi sebuah web seperti warna, ukuran, bentuk, dan lain-lain. Tanpa adanya CSS tampilan sebuah web akan terlihat polos dan kurang teratur. Terdapat tiga cara dalam menerapkan CSS pada sebuah web antara lain secara *inline* atau langsung di *tag* HTML, *internal* atau di dalam berkas HTML, serta *external* atau dengan membuat berkas CSS di luar berkas HTML. Sejenis dengan HTML, CSS bukan merupakan bahasa pemrograman, tetapi hanya sintaks yang tidak dapat menjalankan logika dan aritmetika.

#### 9. **JavaScript**

JavaScript adalah bahasa pemrograman yang pemrosesannya dilakukan di sisi *client* untuk memberikan efek dinamis, interaktif, dan bersifat dependen yang hanya dapat dijalankan dengan menggunakan

*browser* [13]. JavaScript dapat meningkatkan fungsionalitas halaman web dan membuat sebuah web menjadi lebih menarik dan hidup. Berbeda dengan HTML dan CSS yang berbentuk sintaks, JavaScript merupakan bahasa pemrograman yang dapat menjalankan logika dan aritmetika. Oleh karena itu JavaScript mampu membuat sebuah web lebih interaktif. Bersama dengan HTML dan CSS dalam sebuah web, JavaScript berinteraksi langsung dengan pengguna atau biasa disebut dengan *frontend*.

#### 10. *Hypertext Preprocessor (PHP)*

*Hypertext Preprocessor (PHP)* adalah bahasa pemrograman di sisi *server (server-side scripting)* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Maksud dari *server-side scripting* adalah sintaks dan perintah-perintah yang diberikan sepenuhnya akan dijalankan dalam *server* [14]. PHP juga merupakan bahasa pemrograman yang dapat menjalankan logika dan aritmetika. PHP akan membuat sebuah web menjadi dinamis atau dengan kata lain pengguna dapat melakukan interaksi secara aktif dengan web, seperti mengolah atau merubah data pada web. PHP dijalankan bersama dengan HTML, CSS, dan JavaScript dalam membangun sebuah web yang dinamis. Namun berbeda dengan HTML, CSS, dan JavaScript dalam sebuah web, PHP berjalan di sisi *server* atau biasa disebut dengan *backend*.

#### 11. MySQL

MySQL adalah sebuah sistem manajemen basis data atau disebut dengan *Database Management System (DBMS)* yang menggunakan bahasa *Structured Query Language (SQL)*. MySQL mampu menangani data yang cukup besar [14]. MySQL digunakan untuk menjalankan sintaks dan perintah-perintah dalam manajemen *database* antara lain untuk membaca, menyimpan, mengubah, dan menghapus data atau dikenal dengan CRUD (*create, read, update, and delete*). MySQL dituliskan di dalam bahasa pemrograman PHP pada pengembangan

sebuah web. Bersama dengan PHP dalam sebuah web, MySQL berjalan di sisi *server* atau biasa disebut dengan *backend*.

## 12. *JavaScript Object Notation (JSON)*

*JavaScript Object Notation (JSON)* merupakan sebuah format penyimpanan dan pertukaran data yang ringan, mudah dibaca dan ditulis, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format JSON hampir sama dengan fungsi sebuah tabel yang berisi kolom dan baris atau bisa juga hanya kolom dan baris saja [15]. JSON biasanya digunakan sebagai format pertukaran data pada sebuah API. Jadi jika sebuah web akan mengakses data dari sebuah API, maka format data yang disajikan dalam bentuk JSON. Sebenarnya terdapat format penyimpanan dan pertukaran data selain JSON seperti XML, YAML, CSV, dan lain-lain. Namun saat ini JSON lebih banyak digunakan karena lebih mudah dibaca.

## 13. *Unified Modeling Language (UML)*






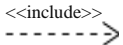
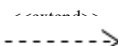
*Unified Modeling Language (UML)* adalah suatu metode yang menyediakan pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat perancangan atau cetak biru (*blueprint*) dalam bentuk yang baku, mudah dimengerti, serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan yang telah dibuat [16]. Diagram-diagram UML yang digunakan dalam penelitian ini antara lain diuraikan sebagai berikut:

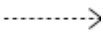


### a. *Use case diagram*

Diagram ini digunakan untuk mendeskripsikan urutan aksi-aksi atau interaksi yang ditampilkan sistem dari masing-masing entitas atau aktor. *Use case diagram* digunakan untuk mengetahui fungsi-fungsi apa saja yang ada dalam sebuah sistem dan untuk mengetahui siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang sering digunakan dalam *use case diagram* diuraikan pada Tabel 2.2.



Tabel 2.2. Simbol pada *Use Case Diagram*








NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
3		<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
4		<i>Association</i>	Penghubung antara objek satu dengan objek lainnya.
5		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek di atasnya/objek induk ( <i>ancestor</i> ).
6		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber membutuhkan <i>use case</i> target untuk menjalankan fungsinya.
7		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.

NO	GAMBAR	NAMA	KETERANGAN
8		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

b. *Activity diagram*

Diagram ini digunakan untuk memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Alur kerja pada *activity diagram* hanya menjelaskan satu kegiatan dari pengguna/aktor untuk satu kasus dalam sistem. Artinya satu *activity diagram* hanya menjelaskan satu *use case* saja. Activity diagram digunakan untuk mengetahui aktivitas pengguna/aktor berdasarkan *use case diagram* yang telah dibuat sebelumnya. Simbol-simbol yang sering digunakan dalam *activity diagram* diuraikan pada Tabel 2.3.

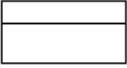






Tabel 2.3. Simbol pada *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas saling berinteraksi satu sama lain.
2		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Transition arrow</i>	Notasi ini mewakili arah aliran grafik. Panah menunjukkan arah kegiatan.
4		<i>Decision</i>	Notasi pilihan aktivitas di mana ada dua kondisi yang dilakukan untuk melanjutkan aktivitas berikutnya.
5		<i>Initial node</i>	Bagaimana objek dibentuk atau diawali.
6		<i>Activity final node</i>	Bagaimana objek berakhir dan dihancurkan.
7		<i>Fork node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

c. *Class diagram*

Diagram ini digunakan untuk menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). Penggambaran dilakukan dalam sebuah model diagram yang menunjukkan kelas dan hubungan antarkelas satu sama lain. Simbol-simbol yang sering digunakan dalam *class diagram* diuraikan pada Tabel 2.4.

Tabel 2.4. Simbol pada *Class Diagram*


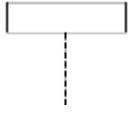

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
2		<i>Association</i>	Relasi antarkelas dengan makna umum.
3		<i>Directional association</i>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
4		<i>Generalization</i>	Relasi antara kelas anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya/kelas induk ( <i>ancestor</i> ).
5		<i>Aggregation</i>	Relasi antarkelas dengan makna semua bagian.
6		<i>Nary association</i>	Upaya untuk menghindari asosiasi dengan lebih dari dua objek.
7		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
8		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> )


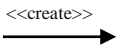
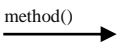
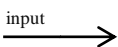
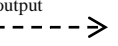
NO	GAMBAR	NAMA	KETERANGAN
			akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.

d. *Sequence diagram*

Diagram ini digunakan untuk menggambarkan kelakuan (*behaviour*) objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima objek. Sama dengan *activity diagram*, satu *sequence diagram* hanya menjelaskan satu *use case* saja. Simbol-simbol yang sering digunakan dalam *sequence diagram* diuraikan pada Tabel 2.5.

Tabel 2.5. Simbol pada *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan dalam sebuah sistem.
2		<i>Lifeline</i>	Menyatakan objek yang berinteraksi dan masa hidup objek.
3		<i>Lifetime</i>	Menyatakan suatu objek dalam keadaan aktif dan berinteraksi. Semua yang berhubungan dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.

NO	GAMBAR	NAMA	KETERANGAN
4		<i>Message</i>	Spesifikasi dari komunikasi antarobjek yang memuat informasi aktivitas yang terjadi.
5		<i>Type create message</i>	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.
6		<i>Type call message</i>	Menyatakan Suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
7		<i>Type send message</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lain dengan arah panah mengarah pada objek yang menerima.
8		<i>Type return message</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu dengan arah panah mengarah pada objek yang menerima pengembalian.

#### 14. Pengujian *black-box*

Pengujian *black-box* digunakan untuk menguji spesifikasi fungsional dari sebuah sistem atau perangkat lunak [17]. Pengujian ini hanya berfokus pada masukan (*input*) yang diberikan atau keluaran (*output*) yang dihasilkan pada fungsi-fungsi yang ada dalam sistem atau perangkat lunak, tanpa harus mengetahui proses dari *input* atau *output* tersebut [18]. Oleh karena itu pengujian tidak perlu mengetahui bahasa pemrograman atau memeriksa struktur kode dari sistem atau perangkat lunak yang diuji. Hasil pengujian yang didapatkan adalah bagaimana fungsi-fungsi pada sistem atau perangkat lunak dapat bekerja sesuai dengan yang dirancang. Terdapat beberapa teknik dalam pengujian *black-box* antara lain diuraikan sebagai berikut [19]:

- a. *Equivalence partitioning*, yaitu pengujian berdasarkan masukan (*input*), kemudian membagi masukan tersebut menjadi kelas data untuk selanjutnya diuji.
- b. *Boundary value analysis/limit testing*, yaitu menguji batasan nilai input minimal dan maksimal, dan merupakan komplemen (kebalikan) dari *Equivalence Partitioning*.
- c. *Comparison testing*, yaitu menguji setiap versi pada sistem atau perangkat lunak dengan membandingkan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama.
- d. *Sample testing*, yaitu memastikan nilai yang terpilih dapat menghasilkan data yang sesuai dengan data masukan dari *user*.
- e. *Robustness testing*, yaitu data masukan dipilih di luar spesifikasi yang telah didefinisikan, untuk membuktikan bahwa tidak ada kesalahan jika masukan data yang tidak valid.
- f. *Behaviour testing*, yaitu melakukan beberapa kali pengujian dengan cara membuat data baru berkali-kali untuk menghindari data *stack*.
- g. *Performance testing*, yaitu mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan

misalnya seperti aliran data, ukuran pemakaian memori, kecepatan eksekusi, dan lain-lain.

- h. *Requirement testing*, yaitu menguji spesifikasi kebutuhan dari sistem atau perangkat lunak yang diidentifikasi pada tahap spesifikasi kebutuhan dan desain.
- i. *Endurance testing*, yaitu pengujian yang melibatkan kasus uji yang diulang-ulang dengan jumlah tertentu.
- j. *Cause-effect relationship testing*, yaitu pengujian berdasarkan pada hubungan antar sebab (*input*) dan akibat (*output*) yang terdapat pada spesifikasi kebutuhan sistem atau perangkat lunak.

### 15. Pengujian *white-box*

Pengujian *white-box* digunakan untuk menguji detail perancangan dan menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian [17]. Berbeda dengan pengujian *black-box* yang hanya fokus pada *input* yang diberikan atau *output* yang dihasilkan, maka pengujian *white-box* berfokus pada aliran atau prose *input* dan *output* dari suatu sistem atau perangkat lunak. Oleh karena itu penguji harus mengetahui bahasa pemrograman atau mampu memeriksa struktur kode dari sistem atau perangkat lunak yang diuji. Hasil pengujian yang didapatkan adalah bagaimana alur data, struktur kode, dan kemungkinan adanya kegagalan pada sistem atau perangkat lunak. Terdapat beberapa teknik dalam pengujian *white-box* antara lain diuraikan sebagai berikut [20][21]:

- a. Pengujian berbasis alur (*basis path testing*)

Pengujian untuk mengukur kompleksitas logika dari rancangan prosedural. Skenario uji coba yang digunakan pada teknik ini dijamin akan mengeksekusi setiap *statement* dari aplikasi yang diujikan setidaknya satu kali saat tahap pengujian.

- b. Pengujian graf alur (*flowgraph*)

Dihasilkan dari pemetaan diagram alir (*flowchart*) program yang ada untuk merepresentasikan aliran kontrol logika program



yang ada. Teknik ini digunakan pada tahapan pengujian yang berfokus pada penggambaran aliran dari sebuah program.

c. Kompleksitas siklomatis (*cyclomatic complexity*)

*Cyclomatic complexity* merupakan perangkat lunak matrik yang menyediakan ukuran kuantitatif dari kompleksitas logikal suatu program. Hasil kompleksitas siklomatis menggambarkan banyaknya *path* dan batas atas sejumlah ujicoba yang harus dirancang dan dieksekusi untuk seluruh perintah dalam program.

d. Jalur independen (*independent path*)

Jalur independen merupakan seluruh jalur sebuah program yang menuju sekurang-kurangnya satu himpunan dari *statement* proses atau kondisi baru. Jadi sebuah jalur independen harus berpindah sekurang-kurangnya satu *edge* yang belum pernah dilalui dengan catatan setiap alur baru melewati *edge* baru. Hasil dari teknik pengujian ini adalah adanya beberapa jalur (*path*) yang ada dalam sebuah aliran program.

e. Pengujian nilai (*value test*)

Pengujian nilai yang merupakan teknik untuk menguji suatu proses masukan dan keluaran (*input and output*) dari sebuah jalur independen yang telah ditentukan. Hasil dari teknik pengujian ini adalah berhasil atau tidaknya suatu proses *input* dan *output* dari setiap jalur independen.

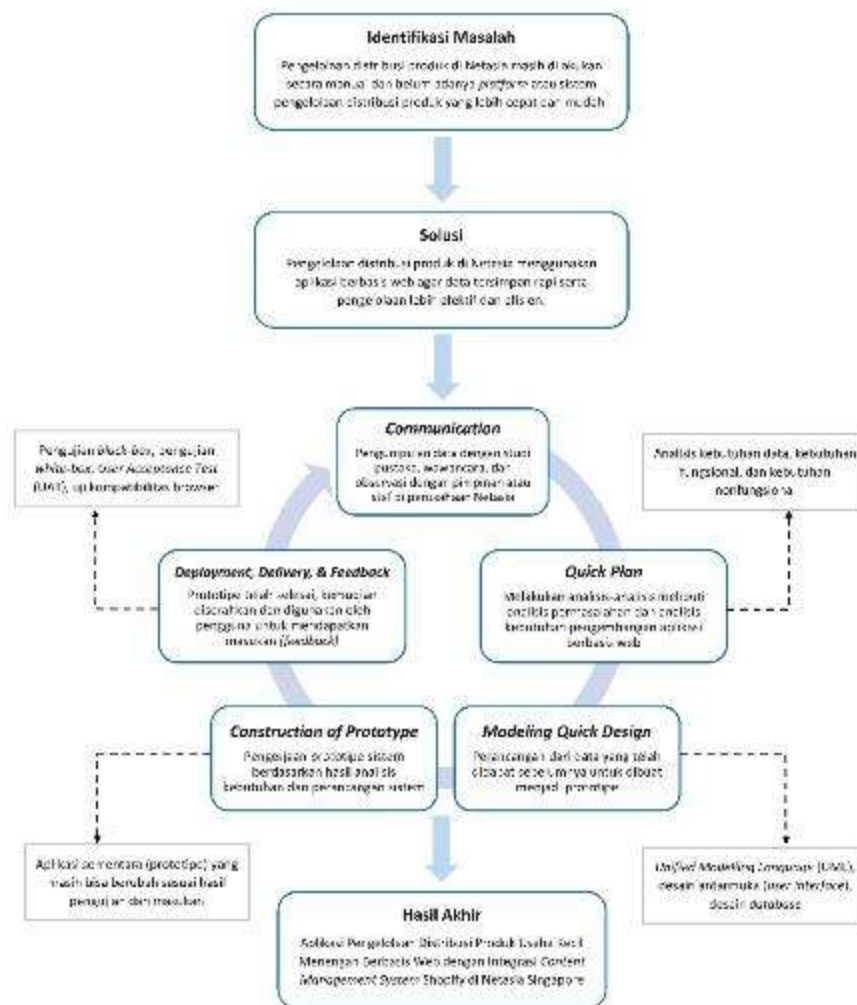
## 16. *User Acceptance Testing* (UAT)

*User Acceptance Testing* (UAT) merupakan proses verifikasi bahwa solusi yang dibuat dalam sistem atau perangkat lunak sudah sesuai untuk pengguna. Berbeda dengan pengujian sistem seperti *black-box* atau *white-box* berfokus pada fungsionalitas dan struktur kode sistem atau perangkat lunak yang diuji, proses UAT berfokus pada solusi dalam sistem atau perangkat lunak apakah diterima oleh pengguna [22]. Proses UAT dilakukan dengan memberikan sistem atau perangkat lunak kepada pengguna untuk digunakan. Hasil yang dicapai pada UAT adalah

masukan (*feedback*) dari pengguna mengenai pengalaman menggunakan sistem atau perangkat lunak. Kemudian penguji atau pengembang akan mengidentifikasi masalah serta melakukan evaluasi dan perbaikan pada sistem atau perangkat lunak yang telah diujikan.

### C. Kerangka Berpikir

Penelitian yang dilakukan melalui beberapa tahapan kegiatan yang tertuang dalam kerangka berpikir yang meliputi identifikasi masalah, solusi permasalahan, proses pengembangan dan pengujian aplikasi, serta hasil akhir. Kerangka berpikir pada penelitian ini dapat dilihat pada Gambar 2.1.



Gambar 2.1. Kerangka Berpikir

## **BAB III**

### **METODE PENELITIAN**

#### **A. Pendekatan Penelitian**

Penelitian dilakukan dengan menggunakan metode penelitian dan pengembangan atau *Research and Development* (R&D). Metode R&D merupakan sebuah metode yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut [23]. Metode R&D merupakan pendekatan sistematis untuk mengembangkan sistem perangkat lunak (*software*) dengan beberapa tahapan antara lain: 1) pemetaan potensi dan masalah, 2) pengumpulan data, 3) desain produk, 4) validasi desain, 5) revisi desain, 6) uji coba produk, 7) revisi produk, 8) uji coba pemakaian, 9) revisi produk, dan 10) produksi massal.

Metode R&D digunakan pada penelitian ini karena hasil akhir yang dicapai dari penelitian ini adalah sebuah produk berupa aplikasi berbasis web. Selain itu penelitian ini juga sering melibatkan interaksi antara pengguna dengan pengembang, sehingga aplikasi berbasis web yang dikembangkan juga beberapa kali mengalami perubahan atau revisi berdasarkan hasil interaksi dari pengguna. Hal tersebut sesuai dengan tahapan-tahapan yang ada dalam metode R&D. Namun karena keterbatasan waktu penelitian, berdasarkan tahapan yang ada dalam metode R&D, maka penelitian tidak dilakukan hingga tahap terakhir atau tahap produksi massal [24].

#### **B. Lokasi Penelitian**

##### **1. Lokasi Penelitian**

Lokasi penelitian berada di perusahaan Golden Style Management Pte. Ltd. Singapura. Perusahaan ini beralamat di 190 Clemenceau Avenue #06-08, Singapura. Perusahaan ini memiliki merek bernama Netasia yang menjadi subjek pada penelitian ini. Karena keterbatasan waktu dan biaya, maka penelitian di perusahaan ini dilakukan secara daring menggunakan alat komunikasi jarak jauh.

## 2. Fokus Penelitian

Penelitian yang dilakukan berfokus pada pengembangan dan pengujian aplikasi pengelolaan distribusi produk UKM berbasis web dengan integrasi CMS Shopify di Netasia Singapore untuk mendapatkan perubahan kondisi pada permasalahan yang akan diselesaikan, dalam hal ini efektifitas dan efisiensi distribusi produk UKM di Netasia Singapore.

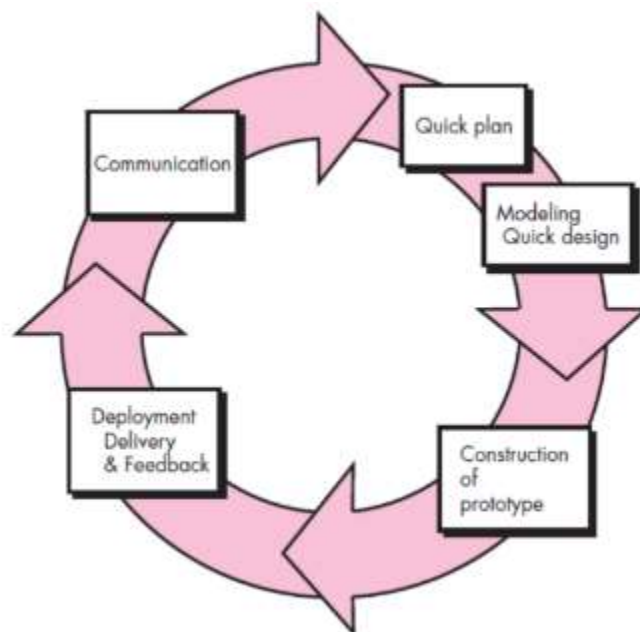
### C. Populasi dan Sampel

Populasi merupakan wilayah generalisasi yang terdiri atas objek atau subjek dengan karakteristik tertentu yang ditetapkan oleh peneliti untuk diteliti [25]. Atau dengan kata lain populasi merupakan keseluruhan subjek penelitian. Berdasarkan pengertian tersebut, maka populasi dari penelitian ini adalah pengguna aplikasi yang terdiri dari tiga level yakni admin, vendor, dan toko. Populasi untuk masing-masing level tersebut antara lain pimpinan dan staf perusahaan Golden Style Management Lte. Ptd. sebagai admin, vendor UKM di Indonesia sebagai vendor, dan beberapa toko di Singapura yang telah bekerja sama dengan Netasia sebagai toko.

Sementara itu sampel merupakan bagian dari jumlah dan karakteristik yang dimiliki oleh populasi [25]. Sampel diambil dari populasi dan harus dapat mewakili populasi tersebut. Berdasarkan pengertian tersebut, maka sampel untuk masing-masing level pengguna aplikasi pada penelitian ini antara lain satu orang staf dari perusahaan Golden Style Management Lte. Ptd. sebagai admin, satu orang dari vendor UKM di Indonesia sebagai vendor, dan satu orang dari toko di Singapura yang telah bekerja sama dengan Netasia sebagai toko, dengan jumlah sampel keseluruhan ganjil. Sementara itu teknik *sampling* yang digunakan adalah *simple random sampling* atau penentuan sampel dari populasi untuk masing-masing level pengguna aplikasi dilakukan secara acak tanpa memperhatikan kondisi atau spesifikasi tertentu karena anggota populasi dianggap homogen.

#### D. Desain Penelitian

Penelitian dilakukan dengan menggunakan salah satu metode pengembangan sistem dari *System Development Lifecycle* (SDLC) yaitu metode *prototyping*. *Prototyping* merupakan suatu metode pengembangan sistem yang menggunakan pendekatan untuk membuat sesuatu program dengan cepat dan bertahap sehingga segera dapat dievaluasi oleh pengguna [26]. Metode *prototyping* memungkinkan pengguna untuk berinteraksi dengan pengembang selama proses pengembangan sistem sedang berlangsung. Hal ini bertujuan agar pengembang mendapatkan gambaran sistem yang akan dikembangkan melalui pengembangan prototipe terlebih dahulu yang akan dievaluasi oleh pengguna [27]. Terdapat beberapa tahap dalam metode *prototyping* seperti yang dapat dilihat pada Gambar 3.1 [28].



Gambar 3.1. Metode *Prototyping*

Berdasarkan tahapan-tahapan pada metode *prototyping* yang disajikan pada Gambar 3.1, maka tahapan-tahapan tersebut diuraikan sebagai berikut:

##### 1. Komunikasi (*communication*)

Komunikasi meliputi pengumpulan data dan informasi yang dilakukan dengan tiga metode antara lain studi pustaka, wawancara, dan

observasi. Studi pustaka dilakukan dengan mencari sumber-sumber referensi berupa artikel dari jurnal, buku, dan berbagai sumber informasi lainnya yang berkaitan dengan penelitian ini. Selanjutnya wawancara dilakukan dengan pimpinan dan staf perusahaan Golden Style Management Lte. Ptd. menggunakan aplikasi komunikasi jarak jauh. Kemudian dilakukan observasi secara langsung terhadap data yang ada di CMS Shopify milik perusahaan.

## 2. **Perencanaan cepat (*quick plan*)**

Perencanaan cepat terdiri dari analisis masalah dan analisis kebutuhan yang meliputi kebutuhan data, kebutuhan fungsional sistem, dan kebutuhan nonfungsional sistem pada aplikasi berbasis web yang dikembangkan. Tahapan dalam perencanaan diuraikan sebagai berikut:

### a. Analisis masalah

Selain menjual produk kepada pelanggan akhir (*enduser*) melalui CMS Shopify, Netasia juga perlu mengelola pendistribusian produk mulai dari *preorder* kepada vendor hingga distribusi produk kepada toko di Singapura. Namun saat ini Netasia belum memiliki *platform* untuk mengelola pendistribusian produk UKM tersebut. Pengelolaan yang dilakukan Netasia saat ini masih sebatas menggunakan aplikasi komunikasi seperti WhatsApp dan pencatatan manual menggunakan Google Spreadsheet atau Microsoft Excel. Hal ini dapat menyebabkan adanya penumpukan berkas yang berpotensi mempersulit pengelolaan distribusi produk.

Berdasarkan permasalahan yang telah diuraikan, maka dikembangkanlah suatu aplikasi berbasis web untuk mengelola pendistribusian produk UKM yang dapat menggantikan pengelolaan yang sebelumnya dilakukan secara manual. Namun aplikasi berbasis web tersebut harus dapat dioperasikan secara lebih efektif dan efisien. Salah satunya adalah mengintegrasikan data produk yang telah dimiliki Netasia di CMS Shopify dengan aplikasi berbasis web

yang dikembangkan agar tidak perlu melakukan pengunggahan ulang data produk.

b. Analisis kebutuhan

Kebutuhan dalam pengembangan aplikasi pengelolaan distribusi produk UKM berbasis web dengan integrasi CMS Shopify di Netasia Singapore antara lain terdiri dari kebutuhan data, kebutuhan fungsional sistem, dan kebutuhan nonfungsional sistem. Kebutuhan dalam pengembangan aplikasi berbasis web ini diuraikan sebagai berikut:

1) Kebutuhan data

Data yang dibutuhkan dalam pengembangan aplikasi berbasis web ini antara lain data vendor dan data produk vendor yang didapatkan dari CMS Shopify milik perusahaan Golden Style Management Lte. Ptd., serta data toko yang didapatkan dari hasil wawancara dengan pimpinan dan staf perusahaan.

2) Kebutuhan fungsional sistem

Berdasarkan hasil komunikasi dengan pihak perusahaan yang meliputi pengumpulan data dan informasi baik dengan studi pustaka, wawancara, dan observasi, didapatkan kebutuhan fungsional sistem yang diinginkan oleh perusahaan antara lain sistem *multi-user login* (admin, vendor, toko), pengelolaan akun, tabel daftar vendor, daftar produk vendor, dan daftar toko, sistem *preorder* dari toko ke admin, sistem *preorder* dari admin ke vendor, riwayat *preorder* dari toko beserta statusnya, serta riwayat *preorder* kepada vendor beserta statusnya.

3) Kebutuhan nonfungsional sistem

Kebutuhan nonfungsional sistem meliputi kebutuhan perangkat lunak (*software*) dan kebutuhan perangkat keras (*hardware*) untuk menunjang pengembangan sebuah sistem. Kebutuhan nonfungsional sistem pada pengembangan aplikasi berbasis web ini dapat dilihat pada Tabel 3.1.

Tabel 3.1. Kebutuhan Nonfungsional Sistem

No	Kebutuhan Nonfungsional	Keterangan
Perangkat Lunak		
1	Aplikasi XAMPP (rekomendasi versi 7)	Web <i>server</i> lokal untuk menguji aplikasi selama tahap pengembangan
2	<i>Web browser</i> (rekomendasi Google Chrome versi 99)	Sebagai penjelajah web dan tempat aplikasi berjalan
3	<i>Text editor</i> (Sublime Text/ Visual Studio Code)	Digunakan untuk membuat atau menulis program komputer
4	<i>Framework</i> Codeigniter (rekomendasi versi 3)	Membantu pengembangan web lebih terstruktur dan tersusun rapi
5	<i>HyperText Markup Language</i> (HTML)	Sintaks untuk menyusun kerangka web
6	<i>Cascading Style Sheets</i> (CSS)	Sintaks untuk mengatur tampilan web
7	JavaScript	Bahasa pemrograman di sisi <i>client</i> untuk memberikan efek dinamis dan interaktif pada suatu web
8	<i>Hypertext Preprocessor</i> (PHP) versi 7	Bahasa pemrograman di sisi <i>server</i> ( <i>backend</i> )
9	MySQL versi 10	Digunakan untuk pengelolaan <i>database</i>
10	Adobe Photoshop	Aplikasi untuk desain dan edit gambar



Perangkat Keras		
11	Komputer/laptop dengan spesifikasi minimal CPU Intel Core i3 atau setara, memori (RAM) 4GB, <i>harddisk/SSD</i> 50GB	Perangkat keras yang digunakan dalam pengembangan aplikasi

### 3. Perancangan model cepat (*modeling quick design*)

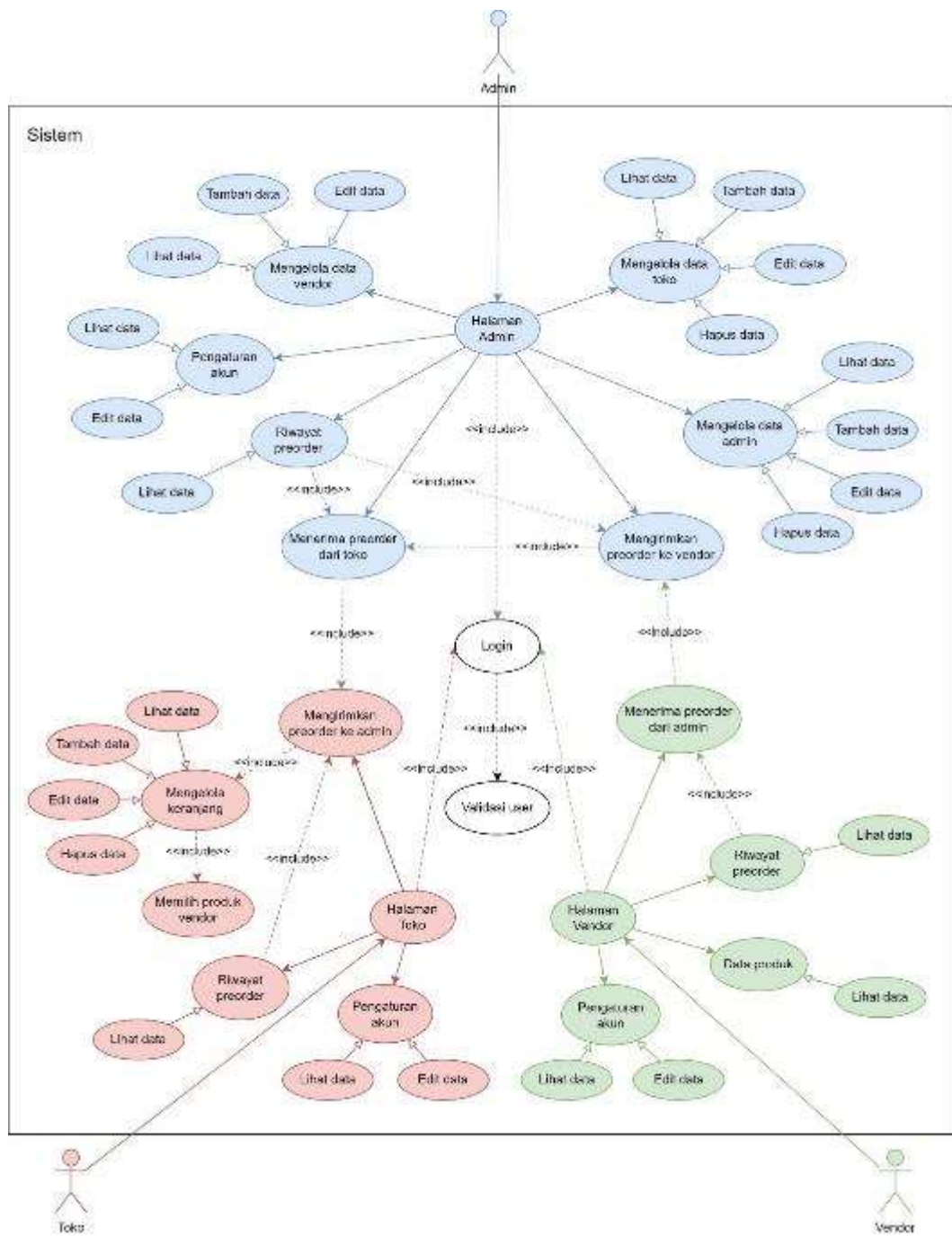
Perancangan model dilakukan berdasarkan data yang didapat sebelumnya untuk selanjutnya divisualisasikan dalam model sistem menggunakan *Unified Modeling Language* (UML), desain *database*, dan desain antarmuka pengguna (*user interface*). Perancangan model diuraikan sebagai berikut:

#### a. Model sistem

Perancangan aplikasi diawali dengan pembuatan model sistem. Pembuatan model sistem dilakukan dengan menggunakan diagram-diagram UML. Terdapat empat diagram UML yang digunakan dalam perancangan aplikasi ini, antara lain *use case diagram*, *sequence diagram*, *activity diagram*, dan *class diagram* yang diuraikan sebagai berikut:

##### 1) *Use case diagram*

Diagram ini digunakan untuk mendeskripsikan urutan aksi-aksi yang ditampilkan sistem dari masing-masing entitas atau aktor. Terdapat beberapa *use case* untuk setiap aktor dalam sebuah *use case diagram*. Berdasarkan *use case diagram* yang dibuat seperti yang ditunjukkan pada Gambar 3.2, terdapat tiga aktor dalam aplikasi berbasis web yang dikembangkan pada penelitian ini yakni admin, vendor, dan toko. Setiap aktor memiliki *use case* masing-masing. Namun terdapat satu *use case* yang melibatkan seluruh aktor yaitu *use case login*.



Gambar 3.2. Use Case Diagram

Berdasarkan *use case diagram* yang telah dibuat, maka diperlukan juga skenario untuk memahami alur jalannya *use case*. Skenario *use case* digunakan untuk menjelaskan proses *use case* baik dari sisi aktor maupun dari sisi sistem. Terdapat dua skenario dalam suatu *use case* yakni skenario normal jika sistem berjalan normal tanpa *error* dan skenario alternatif jika sistem tidak berjalan normal atau mengalami *error*. Skenario *use case* ini juga menjadi dasar dalam pembuatan *sequence diagram* dan *activity diagram*.

Skenario *use case* yang digunakan dalam pengembangan aplikasi berbasis web pada penelitian ini terbagi dalam empat halaman utama yakni halaman *login*, halaman admin, halaman vendor, dan halaman toko. Halaman *login* memiliki 1 *use case*, halaman admin memiliki 15 *use case*, halaman vendor memiliki 4 *use case*, serta halaman toko memiliki 3 *use case* dengan total *use case* sebanyak 23 *use case*. Terdapat satu *use case* yang melibatkan seluruh aktor yaitu *use case login*. Skenario *use case login* dapat dilihat pada Tabel 3.2.

Skenario *use case* untuk aktor admin yang dijelaskan dalam tabel antara lain melihat data vendor pada Tabel 3.3, menambah akun vendor pada Tabel 3.4, dan mengedit akun vendor pada Tabel 3.5. Selanjutnya melihat data toko pada Tabel 3.6, menambah data toko pada Tabel 3.7, mengedit data toko pada Tabel 3.8, dan menghapus data toko pada Tabel 3.9. Selanjutnya melihat data admin pada Tabel 3.10, menambah data admin pada Tabel 3.11, mengedit data admin pada Tabel 3.12, dan menghapus data admin pada Tabel 3.13. Selanjutnya menerima *preorder* dari toko pada Tabel 3.14, mengirimkan *preorder* ke vendor pada Tabel 3.15, melihat riwayat *preorder* pada Tabel 3.16, serta pengaturan akun pada Tabel 3.17.

Skenario *use case* untuk aktor vendor yang dijelaskan dalam tabel antara lain melihat data produk pada Tabel 3.18, menerima *preorder* dari admin pada Tabel 3.19, melihat riwayat *preorder* pada Tabel 3.20, serta pengaturan akun pada Tabel 3.21. Sementara itu skenario *use case* untuk aktor toko yang dijelaskan dalam tabel antara lain mengirimkan *preorder* ke admin termasuk mengelola keranjang pada Tabel 3.22, melihat riwayat *preorder* pada Tabel 3.23, dan pengaturan akun pada Tabel 3.24. Skenario untuk masing-masing *use case* pada setiap halaman dalam aplikasi berbasis web yang dikembangkan pada penelitian ini antara lain diuraikan sebagai berikut:

a) Skenario *use case login*

Tabel 3.2. Skenario *Use Case Login*

AKSI AKTOR	REAKSI SISTEM
Aktor: admin, vendor, toko	
<b>Skenario Normal</b>	
1. Memasukkan <i>username</i> dan <i>password</i>	
	2. Validasi akun berdasarkan <i>username, password, level</i>
	3. Menampilkan dasbor sesuai level
<b>Skenario Alternatif</b>	
1. Memasukkan <i>username</i> dan <i>password</i>	
	2. Validasi akun berdasarkan <i>username, password, level</i>
	3. Menampilkan pesan <i>error</i> bahwa <i>login</i> tidak valid

<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
4. Memasukkan <i>username</i> dan <i>password</i> valid	
	5. Validasi akun berdasarkan <i>username</i> , <i>password</i> , <i>level</i>
	6. Menampilkan dasbor sesuai level

b) Skenario *use case* admin melihat data vendor

Tabel 3.3. Skenario *Use Case* Admin Melihat Data Vendor

<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
Aktor: admin	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai admin	
	2. Menampilkan dasbor admin
3. Memilih menu <i>vendor partner</i>	
	4. Menampilkan halaman <i>vendor partner</i>
5. Melihat data vendor	
6. Klik tombol detail di salah satu vendor	
	7. Menampilkan detail vendor berisi daftar produk vendor
8. Melihat daftar produk vendor	

c) Skenario *use case* admin menambahkan akun vendor

Tabel 3.4. Skenario *Use Case* Admin Menambah Akun Vendor

AKSI AKTOR	REAKSI SISTEM
Aktor: admin	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai admin	
	2. Menampilkan dasbor admin
3. Memilih menu <i>vendor partner</i>	
	4. Menampilkan halaman <i>vendor partner</i>
5. Klik tombol <i>add account</i> di salah satu vendor	
	6. Menampilkan <i>modal form</i> untuk menambahkan akun vendor
7. Mengisi <i>form</i> dan klik tombol <i>register</i>	
	8. Menambahkan data akun vendor baru
	9. Menampilkan halaman <i>vendor partner</i>
	10. Tombol <i>add account</i> berubah menjadi <i>edit account</i>

d) Skenario *use case* admin mengedit akun vendorTabel 3.5. Skenario *Use Case* Admin Mengedit Akun Vendor

AKSI AKTOR	REAKSI SISTEM
Aktor: admin	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai admin	
	2. Menampilkan dasbor admin
3. Memilih menu <i>vendor partner</i>	
	4. Menampilkan halaman <i>vendor partner</i>
5. Klik tombol <i>edit account</i> di salah satu vendor	
	6. Menampilkan <i>modal form</i> untuk mengedit akun vendor
7. Mengisi <i>form</i> dan klik tombol <i>update</i>	
	8. Memperbarui akun vendor
	9. Menampilkan halaman <i>vendor partner</i>

e) Skenario *use case* admin melihat data tokoTabel 3.6. Skenario *Use Case* Admin Melihat Data Toko

AKSI AKTOR	REAKSI SISTEM
Aktor: admin	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai admin	
	2. Menampilkan dasbor admin
3. Pilih menu <i>store partner</i>	

<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
	4. Menampilkan halaman <i>store partner</i>
5. Melihat data toko	
6. Klik tombol detail di salah satu toko	
	7. Menampilkan detail toko berisi riwayat <i>preorder</i> toko
8. Melihat detail toko berisi riwayat <i>preorder</i> toko	

f) Skenario *use case* admin menambahkan data toko

Tabel 3.7. Skenario *Use Case* Admin Menambah Data Toko

<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
Aktor: admin	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai admin	
	2. Menampilkan dasbor admin
3. Pilih menu <i>store partner</i>	
	4. Menampilkan halaman <i>store partner</i>
5. Klik tombol <i>add new store</i>	
	6. Menampilkan <i>modal form</i> untuk menambahkan data akun toko baru
7. Mengisi <i>form</i> dan klik tombol <i>register</i>	
	8. Menambahkan data akun toko baru



<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
	9. Menampilkan halaman <i>store partner</i>

g) Skenario *use case* admin mengedit data toko

Tabel 3.8. Skenario *Use Case* Admin Mengedit Data Toko

<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
Aktor: admin	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai admin	
	2. Menampilkan dasbor admin
3. Pilih menu <i>store partner</i>	
	4. Menampilkan halaman <i>store partner</i>
5. Klik tombol <i>edit</i> di salah satu toko	
	6. Menampilkan <i>modal form</i> untuk mengedit akun toko
7. Mengisi <i>form</i> dan klik tombol <i>update</i>	
	8. Memperbarui akun toko
	9. Menampilkan halaman <i>store partner</i>

h) Skenario *use case* admin menghapus data tokoTabel 3.9. Skenario *Use Case* Admin Menghapus Data Toko

AKSI AKTOR	REAKSI SISTEM
Aktor: admin	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai admin	
	2. Menampilkan dasbor admin
3. Pilih menu <i>store partner</i>	
	4. Menampilkan halaman <i>store partner</i>
5. Klik tombol hapus di salah satu toko	
	6. Menghapus akun toko
	7. Menampilkan halaman <i>store partner</i>

i) Skenario *use case* admin melihat data adminTabel 3.10. Skenario *Use Case* Admin Melihat Data Admin

AKSI AKTOR	REAKSI SISTEM
Aktor: admin	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai admin	
	2. Menampilkan dasbor admin
3. Pilih menu <i>admin accounts</i>	
	4. Menampilkan halaman <i>admin accounts</i>
5. Melihat data admin	

j) Skenario *use case* admin menambahkan data admin

Tabel 3.11. Skenario *Use Case* Admin Menambah Data Admin

AKSI AKTOR	REAKSI SISTEM
Aktor: admin	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai admin	
	2. Menampilkan dasbor admin
3. Pilih menu <i>admin accounts</i>	
	4. Menampilkan halaman <i>admin accounts</i>
5. Klik tombol <i>add new admin</i>	
	6. Menampilkan <i>modal form</i> untuk menambahkan akun admin baru
7. Mengisi <i>form</i> dan klik tombol <i>register</i>	
	8. Menambahkan data akun admin baru
	9. Menampilkan halaman <i>admin accounts</i>

k) Skenario *use case* admin mengedit data admin

Tabel 3.12. Skenario *Use Case* Admin Mengedit Data Admin

AKSI AKTOR	REAKSI SISTEM
Aktor: admin	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai admin	
	2. Tampil dasbor admin

<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
3. Pilih menu <i>admin accounts</i>	
	4. Menampilkan halaman <i>admin accounts</i>
5. Klik tombol edit di salah satu admin	
	6. Menampilkan <i>modal form</i> untuk edit akun admin
7. Mengisi <i>form</i> dan klik tombol <i>update</i>	
	8. Memperbarui akun admin
	9. Menampilkan halaman <i>admin accounts</i>

l) Skenario *use case* admin menghapus data admin

Tabel 3.13. Skenario *Use Case* Admin Menghapus Data Admin

<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
Aktor: admin	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai admin	
	2. Menampilkan dasbor admin
3. Pilih menu <i>admin accounts</i>	
	4. Menampilkan halaman <i>admin accounts</i>
5. Klik tombol hapus di salah satu admin	
	6. Menghapus akun admin
	7. Menampilkan halaman <i>admin accounts</i>

m) Skenario *use case* admin menerima *preorder* dari toko

Tabel 3.14. Skenario *Use Case* Admin Menerima *Preorder*

AKSI AKTOR	REAKSI SISTEM
Aktor: admin	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai admin	
	2. Menampilkan dasbor admin
3. Memilih menu <i>preorders</i> dan klik tab <i>preorder from store</i>	
	4. Menampilkan tab <i>preorder from store</i> di halaman <i>preorders</i> berisi daftar <i>preorder</i> dari toko
5. Melihat daftar <i>preorder</i> dari toko	
6. Klik tombol <i>process</i> di salah satu <i>preorder</i>	
	7. Menampilkan halaman <i>process preorder</i>
8. Melihat detail <i>preorder</i> dan klik tombol <i>add to demand</i>	
	9. Memproses <i>preorder</i> untuk dikirimkan ke vendor

n) Skenario *use case* admin mengirim *preorder* ke vendor

Tabel 3.15. Skenario *Use Case* Admin Mengirim *Preorder*

AKSI AKTOR	REAKSI SISTEM
Aktor: admin	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai admin	
	2. Menampilkan dasbor admin
3. Memilih menu <i>preorders</i> dan klik tab <i>preorder to vendor</i>	
	4. Menampilkan tab <i>preorder to vendor</i> di halaman <i>preorders</i> berisi daftar <i>preorder</i> ke vendor
5. Melihat daftar <i>preorder</i> ke vendor	
6. Klik tombol <i>process to vendor</i> di salah satu <i>preorder</i>	
	7. Menampilkan halaman <i>process preorder</i>
8. Melihat detail <i>preorder</i> , mengatur informasi <i>pre-order</i> , dan klik tombol <i>submit preorder</i>	
	9. Mengirimkan informasi <i>preorder</i> ke vendor

o) Skenario *use case* admin melihat riwayat *preorder*

Tabel 3.16. Skenario *Use Case* Admin Melihat Riwayat *Preorder*

AKSI AKTOR	REAKSI SISTEM
Aktor: admin	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai admin	
	2. Menampilkan dasbor admin
3. Memilih menu riwayat <i>preorder</i>	
	4. Menampilkan halaman riwayat <i>preorder</i>
5. Melihat riwayat <i>preorder</i> beserta status	
6. Klik tombol detail di salah satu riwayat	
	7. Menampilkan halaman detail <i>preorder</i>

p) Skenario *use case* pengaturan akun admin

Tabel 3.17. Skenario *Use Case* Pengaturan Akun Admin

AKSI AKTOR	REAKSI SISTEM
Aktor: admin	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai admin	
	2. Menampilkan dasbor admin
3. Memilih menu <i>settings</i>	
	4. Menampilkan halaman pengaturan akun
5. Memperbarui data akun	

<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
6. Klik tombol <i>update</i>	
	7. Menampilkan halaman <i>login</i> dan pesan sukses

q) Skenario *use case* vendor melihat data produk

Tabel 3.18. Skenario *Use Case* Vendor Melihat Data Produk

<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
Aktor: vendor	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai vendor	
	2. Menampilkan dasbor vendor
3. Memilih menu <i>products</i>	
	4. Menampilkan halaman <i>products</i>
5. Melihat data produk	

r) Skenario *use case* vendor menerima *preorder* dari admin

Tabel 3.19. Skenario *Use Case* Vendor Menerima *Preorder*

<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
Aktor: vendor	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai vendor	
	2. Menampilkan dasbor vendor
3. Memilih menu <i>preorders</i>	
	4. Menampilkan halaman <i>preorders</i>
5. Melihat daftar <i>preorder</i> dari admin	



<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
6. Klik tombol <i>process</i> di salah satu <i>preorder</i>	
	7. Menampilkan halaman <i>process preorder</i>
8. Melihat detail <i>preorder</i> dan klik tombol <i>accept preorder</i>	
	9. Menyimpan status <i>preorder</i>
	10. Menampilkan halaman <i>preorders</i>

s) Skenario *use case* vendor melihat riwayat *preorder*

Tabel 3.20. Skenario *Use Case* Vendor Melihat Riwayat *Preorder*

<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
Aktor: vendor	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai vendor	
	2. Menampilkan dasbor vendor
3. Memilih menu riwayat <i>preorder</i>	
	4. Menampilkan halaman riwayat <i>preorder</i>
5. Melihat riwayat <i>preorder</i> beserta status	
6. Klik tombol detail di salah satu riwayat	
	7. Menampilkan halaman detail <i>preorder</i>

t) Skenario *use case* pengaturan akun vendorTabel 3.21. Skenario *Use Case* Pengaturan Akun Vendor

AKSI AKTOR	REAKSI SISTEM
Aktor: vendor	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai vendor	
	2. Menampilkan dasbor vendor
3. Memilih menu <i>settings</i>	
	4. Menampilkan halaman pengaturan akun
5. Memperbarui data akun	
6. Klik tombol <i>update</i>	
	7. Menampilkan halaman <i>login</i> dan pesan sukses

u) Skenario *use case* toko mengirim *preorder* ke adminTabel 3.22. Skenario *Use Case* Toko Mengirim *Preorder*

AKSI AKTOR	REAKSI SISTEM
Aktor: toko	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai toko	
	2. Menampilkan dasbor toko
3. Memilih menu <i>preorders</i>	
	4. Menampilkan halaman <i>preorders</i> berupa daftar vendor
5. Memilih salah satu vendor	
	6. Menampilkan daftar produk-produk vendor

<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
7. Memilih salah satu produk	
	8. Menampilkan halaman detail produk
9. Memilih opsi produk dan klik tombol <i>add to cart</i>	
	10. Memasukkan produk ke dalam keranjang
	11. Menampilkan halaman detail produk
12. Memilih menu <i>cart</i>	
	13. Menampilkan halaman keranjang
14. Mengatur jumlah produk	
15. Klik tombol <i>submit preorder</i>	
	16. Menyimpan <i>preorder</i> dan mengirim ke admin

v) Skenario *use case* toko melihat riwayat *preorder*

Tabel 3.23. Skenario *Use Case* Toko Melihat Riwayat *Preorder*

<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
Aktor: toko	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai toko	
	2. Menampilkan dasbor toko
3. Memilih menu riwayat <i>preorder</i>	
	4. Menampilkan halaman riwayat <i>preorder</i>

<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
5. Melihat riwayat <i>preorder</i> beserta status	
6. Klik tombol detail di salah satu riwayat	
	7. Menampilkan halaman detail <i>preorder</i>

w) Skenario *use case* pengaturan akun toko

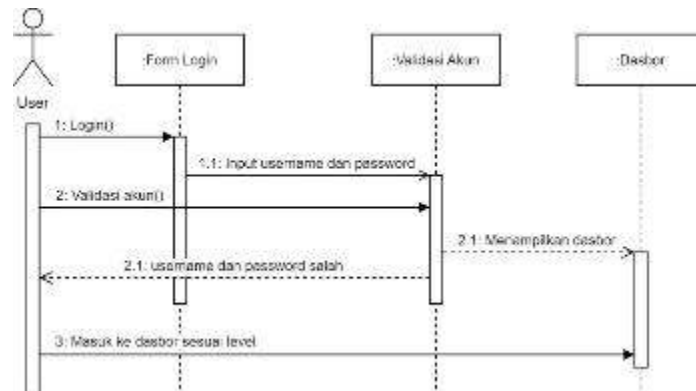
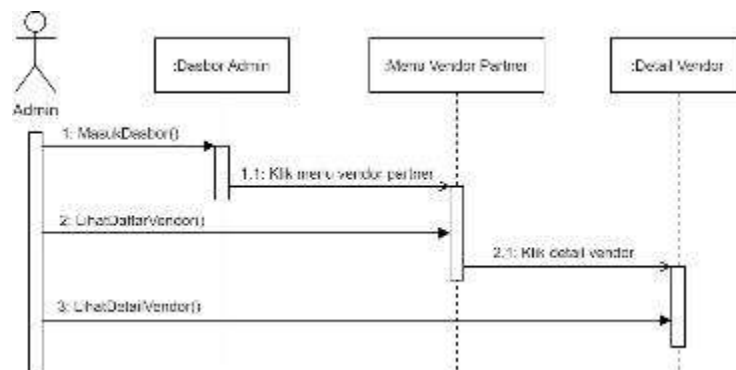
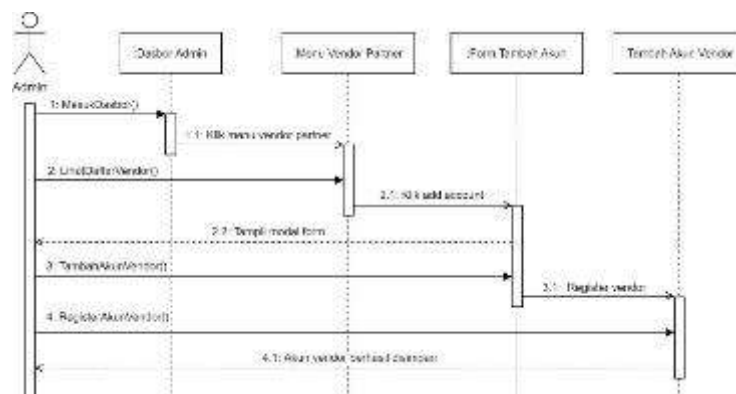
Tabel 3.24. Skenario *Use Case* Pengaturan Akun Toko

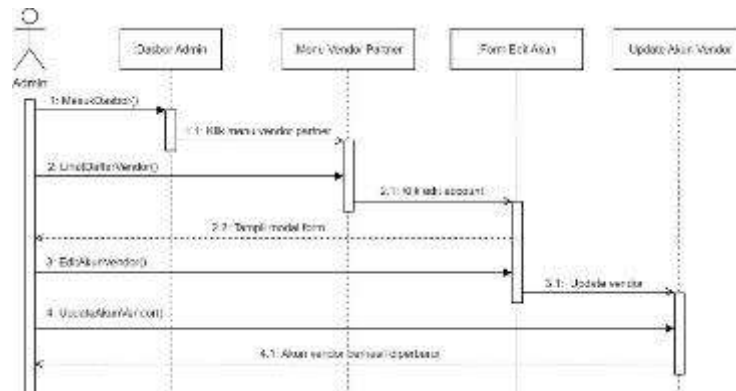
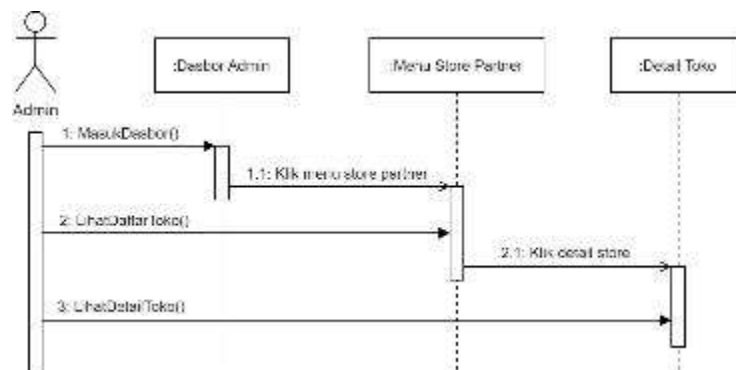
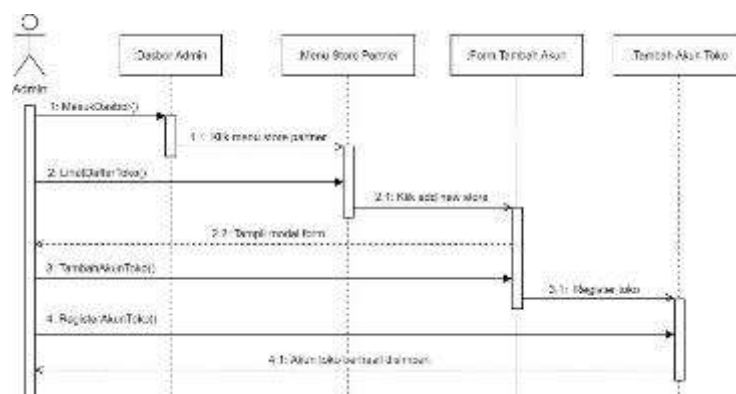
<b>AKSI AKTOR</b>	<b>REAKSI SISTEM</b>
Aktor: toko	
<b>Skenario Normal</b>	
1. <i>Login</i> sebagai toko	
	2. Menampilkan dasbor toko
3. Klik menu <i>settings</i>	
	2. Menampilkan halaman pengaturan akun
3. Memperbarui data akun	
4. Klik tombol <i>update</i>	
	5. Menampilkan halaman <i>login</i> dan pesan sukses

## 2) *Sequence diagram*

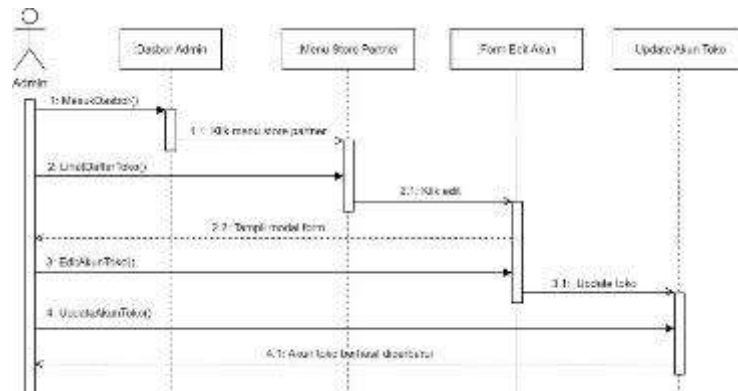
Diagram ini digunakan untuk menggambarkan kelakuan (*behaviour*) objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima objek. *Sequence diagram* ini disusun berdasarkan skenario *use case* yang telah dibuat sebelumnya. Terdapat 23 *use case* yang digambarkan dalam *sequence diagram* yang terbagi ke dalam empat halaman utama pada aplikasi ini antara lain halaman *login* dengan 1 *use case*, halaman admin dengan 15 *use case*, halaman vendor dengan 4 *use case*, dan halaman toko dengan 5 *use case*.

*Sequence diagram* untuk halaman *login* ditunjukkan pada Gambar 3.3. *Sequence diagram* untuk halaman admin antara lain melihat data vendor pada Gambar 3.4, menambahkan akun vendor pada Gambar 3.5, mengedit akun vendor pada Gambar 3.6, melihat data toko pada Gambar 3.7, menambahkan data toko pada Gambar 3.8, mengedit data toko pada Gambar 3.9, menghapus data toko pada Gambar 3.10, melihat data admin pada Gambar 3.11, menambah data admin pada Gambar 3.12, mengedit data admin pada Gambar 3.13, menghapus data admin pada Gambar 3.14, menerima *preorder* pada Gambar 3.15, mengirim *preorder* pada Gambar 3.16, melihat riwayat *preorder* pada Gambar 3.17, serta pengaturan akun pada Gambar 3.18. *Sequence diagram* untuk halaman vendor antara lain melihat data produk pada Gambar 3.19, menerima *preorder* pada Gambar 3.20, melihat riwayat *preorder* pada Gambar 3.21, serta pengaturan akun pada Gambar 3.22. *Sequence diagram* untuk halaman toko antara lain mengirim *preorder* pada Gambar 3.23, melihat riwayat *preorder* pada Gambar 3.24, serta pengaturan akun pada Gambar 3.25. *Sequence diagram* untuk masing-masing *use case* selengkapnya diuraikan sebagai berikut:

a) *Sequence diagram login*Gambar 3.3. *Sequence Diagram Login*b) *Sequence diagram admin melihat data vendor*Gambar 3.4. *Sequence Diagram Admin Melihat Data Vendor*c) *Sequence diagram admin menambahkan akun vendor*Gambar 3.5. *Sequence Diagram Admin Menambah Akun Vendor*

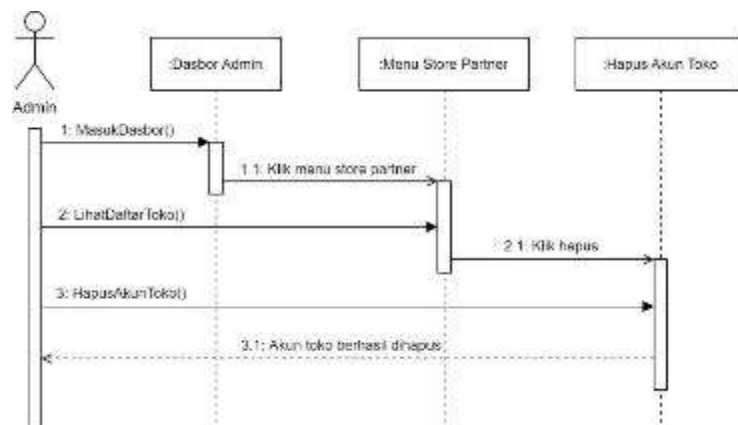
d) *Sequence diagram* admin mengedit akun vendorGambar 3.6. *Sequence Diagram* Admin Mengedit Akun Vendore) *Sequence Diagram* admin melihat data tokoGambar 3.7. *Sequence Diagram* Admin Melihat Data Tokof) *Sequence diagram* admin menambahkan data tokoGambar 3.8. *Sequence Diagram* Admin Menambah Data Toko

g) *Sequence diagram* admin mengedit data toko



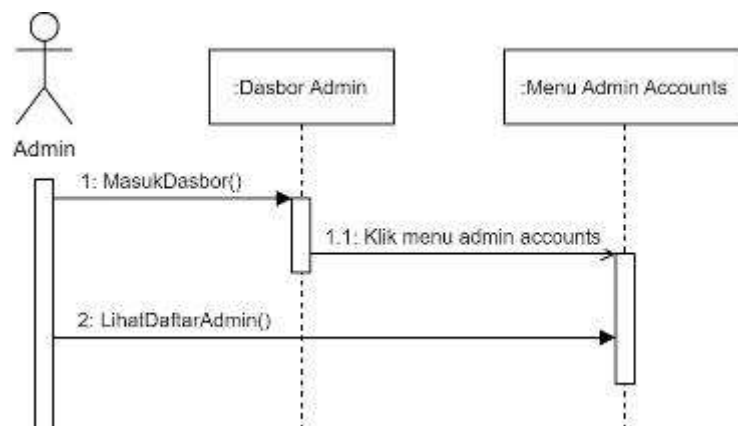
Gambar 3.9. *Sequence Diagram* Admin Mengedit Data Toko

h) *Sequence diagram* admin menghapus data toko



Gambar 3.10. *Sequence Diagram* Admin Menghapus Data Toko

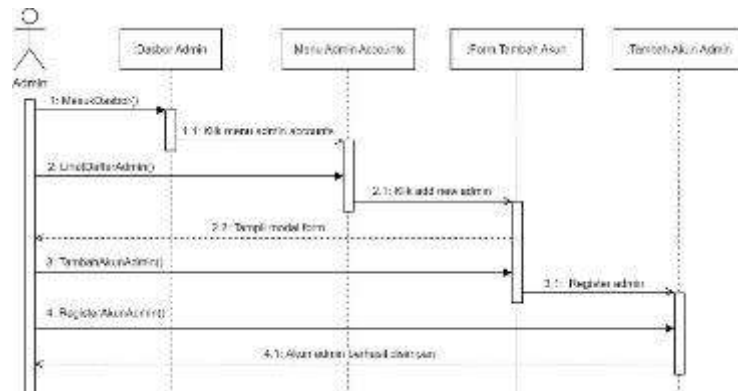
i) *Sequence diagram* admin melihat data admin



Gambar 3.11. *Sequence Diagram* Admin Melihat Data Admin

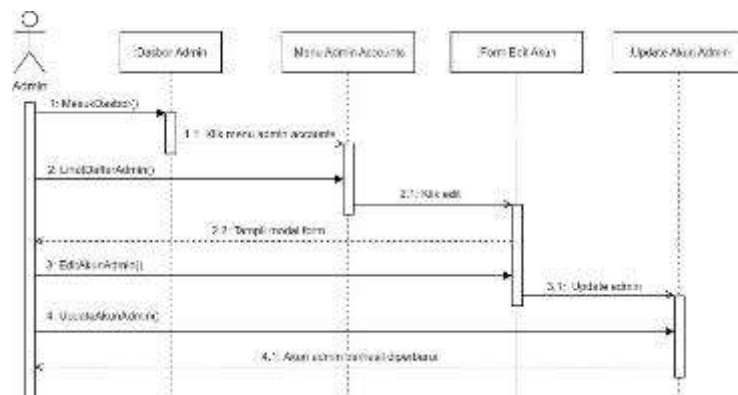


j) *Sequence diagram* admin menambahkan data admin



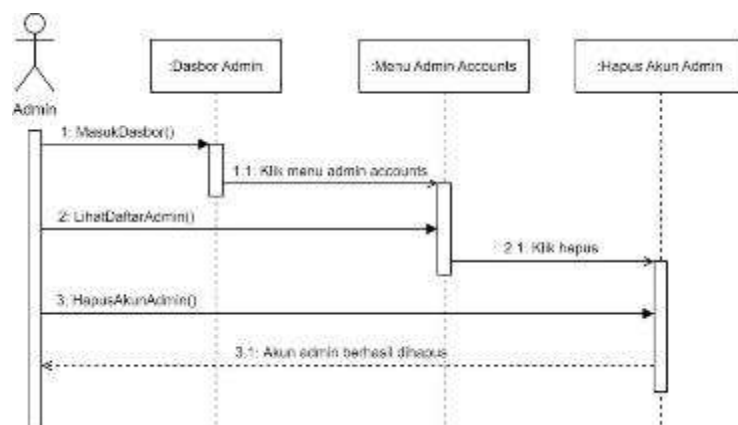
Gambar 3.12. *Sequence Diagram* Admin Menambah Data Admin

k) *Sequence diagram* admin mengedit data admin



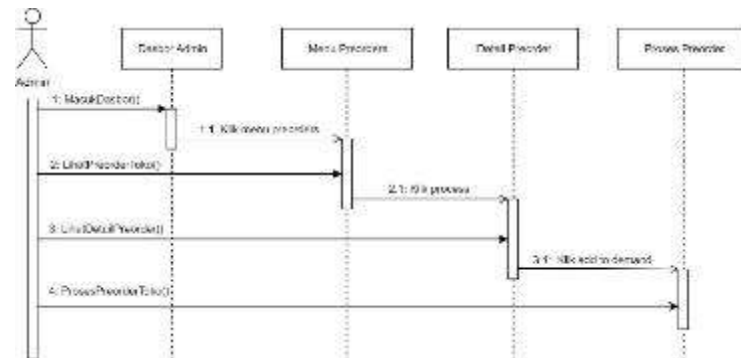
Gambar 3.13. *Sequence Diagram* Admin Mengedit Data Admin

l) *Sequence diagram* admin menghapus data admin



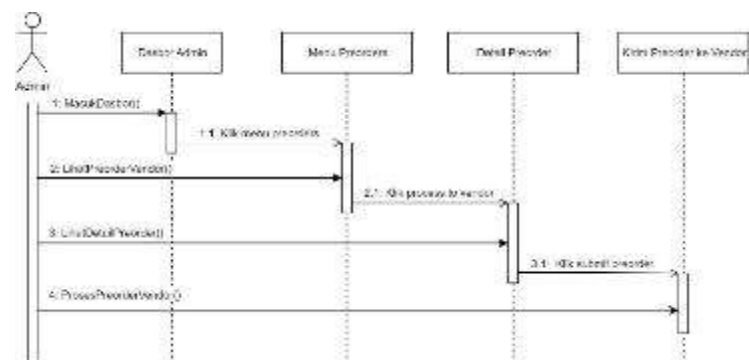
Gambar 3.14. *Sequence Diagram* Admin Menghapus Data Admin

m) *Sequence diagram admin menerima preorder dari toko*



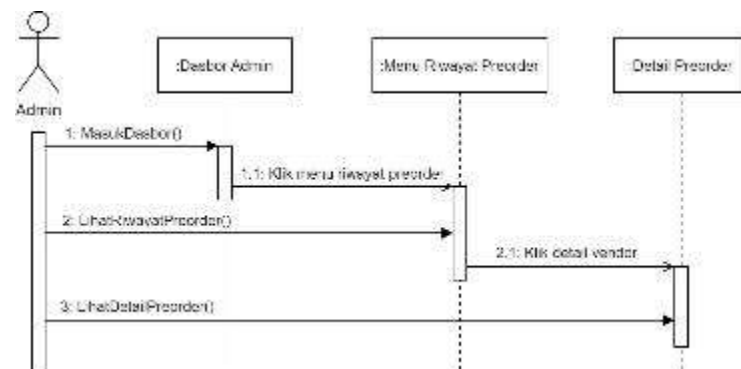
Gambar 3.15. *Sequence Diagram Admin Menerima Preorder*

n) *Sequence diagram admin mengirim preorder ke vendor*



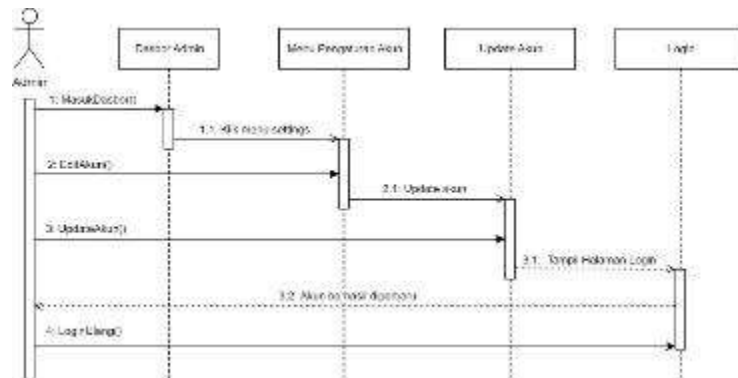
Gambar 3.16. *Sequence Diagram Admin Mengirim Preorder*

o) *Sequence diagram admin melihat riwayat preorder*



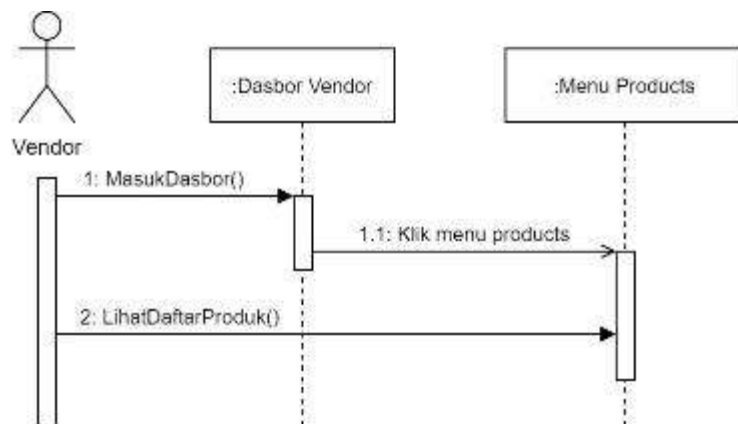
Gambar 3.17. *Sequence Diagram Admin Melihat Riwayat Preorder*

p) *Sequence diagram* pengaturan akun admin



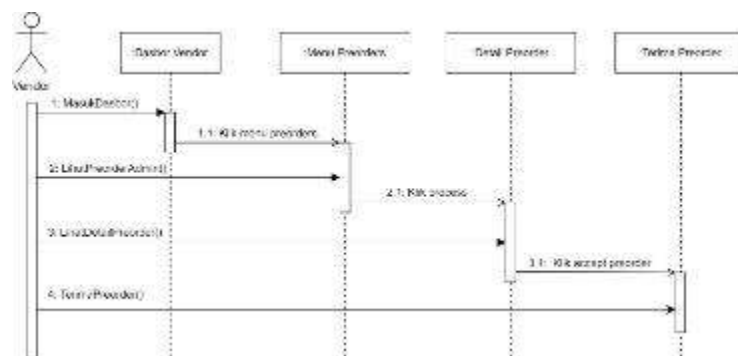
Gambar 3.18. *Sequence Diagram* Pengaturan Akun Admin

q) *Sequence diagram* vendor melihat data produk



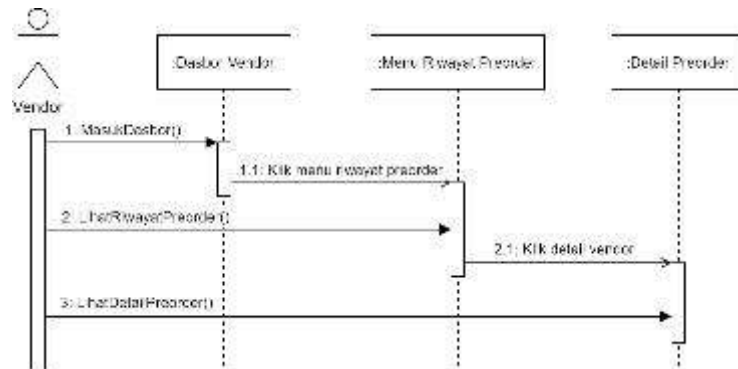
Gambar 3.19. *Sequence Diagram* Vendor Melihat Data Produk

r) *Sequence diagram* vendor menerima preorder dari admin



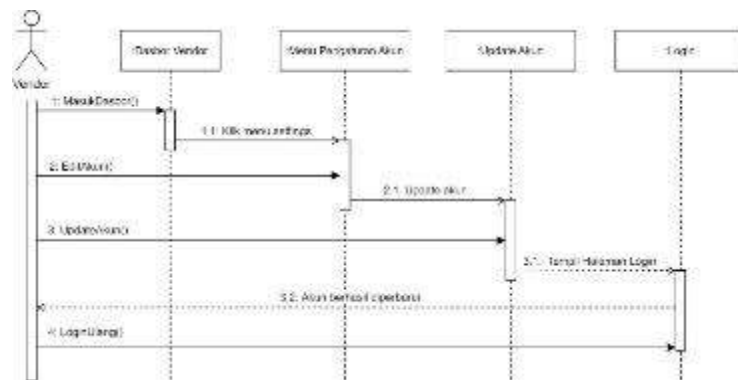
Gambar 3.20. *Sequence Diagram* Vendor Menerima Preorder

s) *Sequence diagram* vendor melihat riwayat preorder



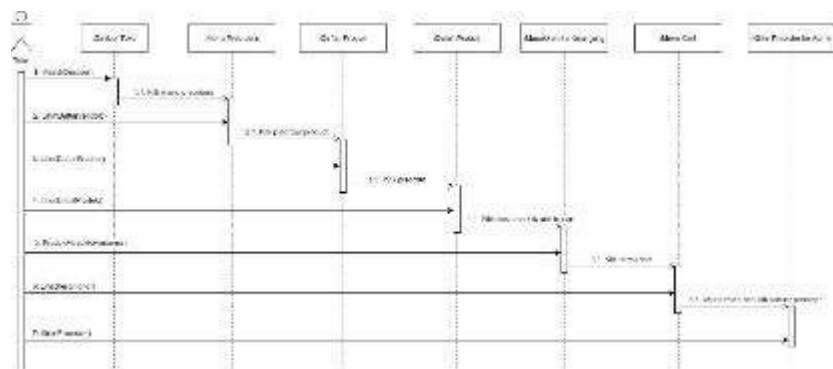
Gambar 3.21. *Sequence Diagram* Vendor Melihat Riwayat Preorder

t) *Sequence diagram* pengaturan akun vendor



Gambar 3.22. *Sequence Diagram* Pengaturan Akun Vendor

u) *Sequence diagram* toko mengirim preorder ke admin



Gambar 3.23. *Sequence Diagram* Toko Mengirim Preorder

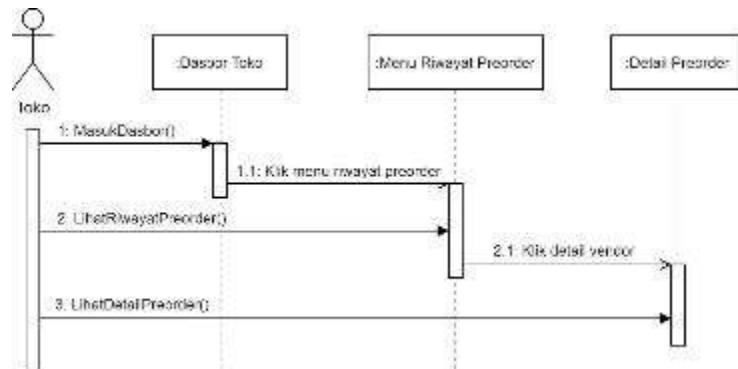
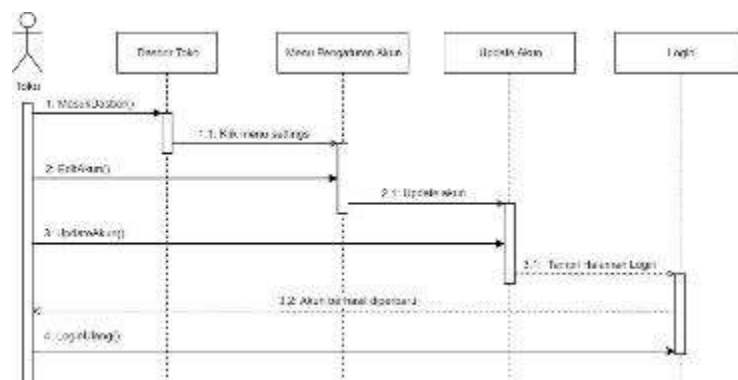
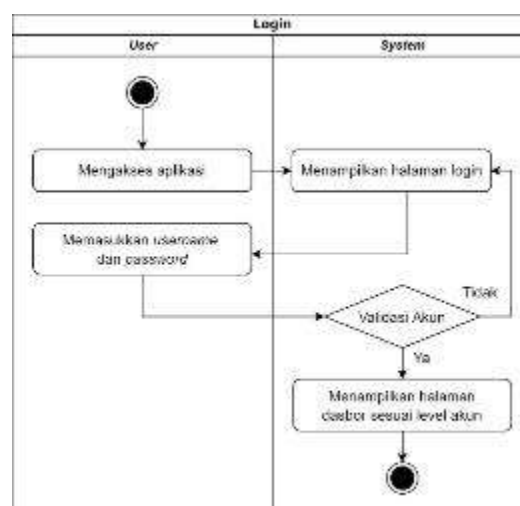
v) *Sequence diagram* toko melihat riwayat preorderGambar 3.24. *Sequence Diagram* Toko Melihat Riwayat Preorderw) *Sequence diagram* pengaturan akun tokoGambar 3.25. *Sequence Diagram* Pengaturan Akun Toko3) *Activity diagram*

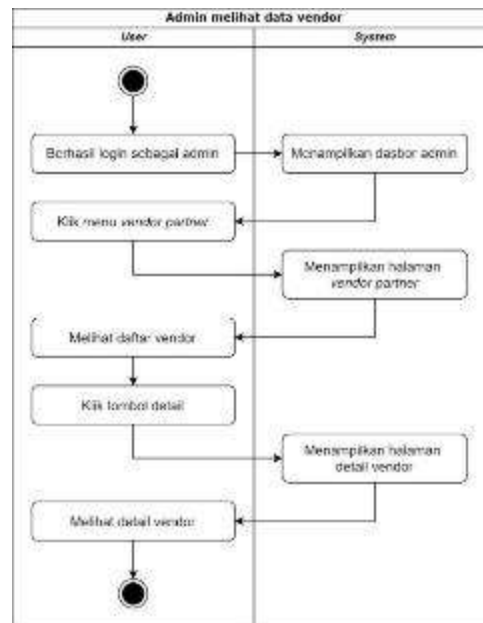
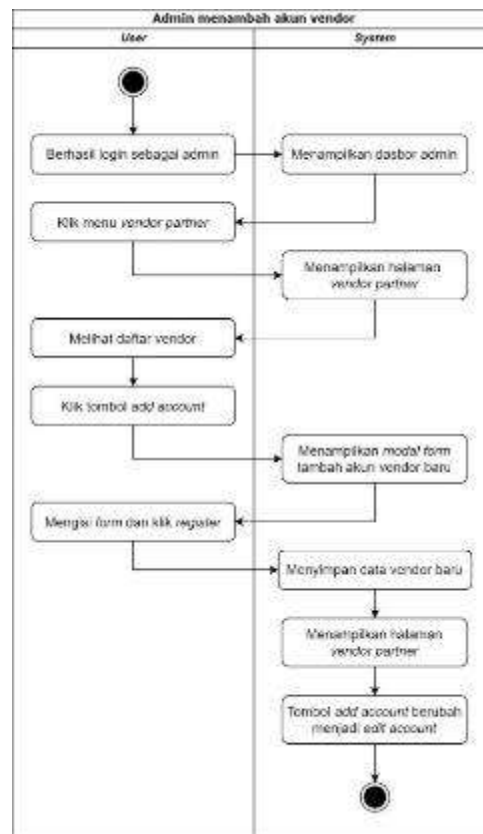
Diagram ini digunakan untuk memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. *Activity diagram* ini disusun berdasarkan skenario *use case* yang telah dibuat sebelumnya. Sama seperti *sequence diagram*, terdapat 23 *use case* yang dimodelkan dalam *activity diagram*. *Use case* tersebut terbagi ke dalam empat halaman utama pada aplikasi ini antara lain halaman *login* dengan 1 *use case*, halaman admin dengan 15 *use case*, halaman vendor dengan 4 *use case*, dan halaman toko dengan 5 *use case*.

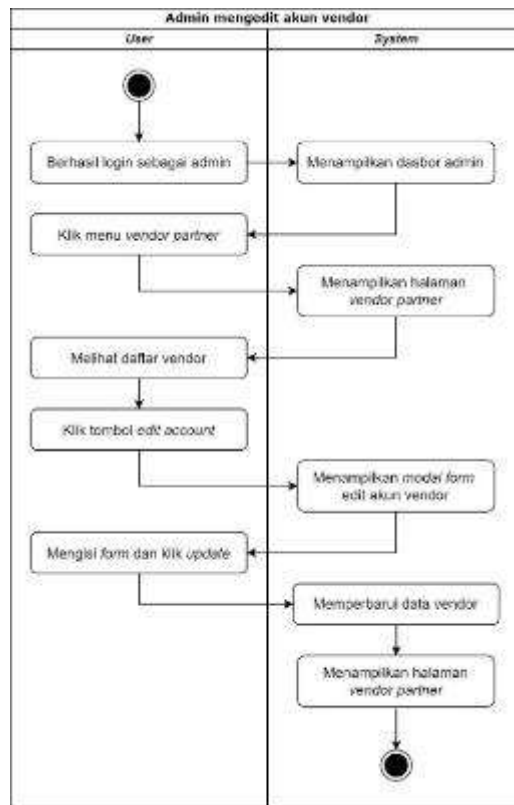
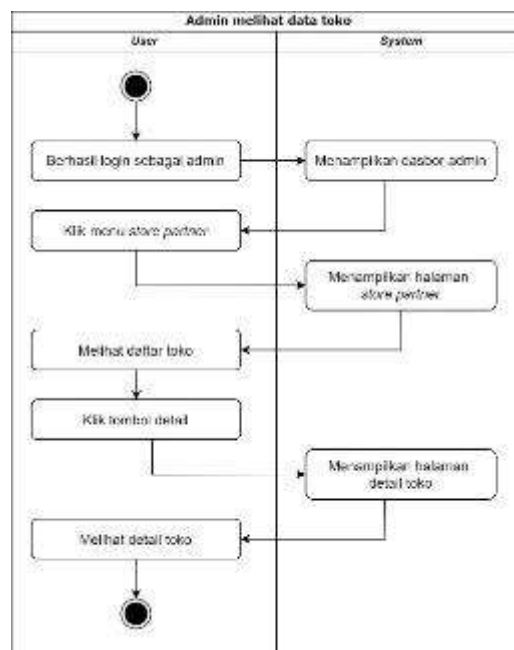
*Activity diagram* untuk halaman *login* ditunjukkan pada Gambar 3.26. *Activity diagram* untuk halaman admin antara lain melihat data vendor pada Gambar 3.27, menambahkan akun vendor pada Gambar 3.28, mengedit akun vendor pada Gambar 3.29, melihat data toko pada Gambar 3.30, menambahkan data toko pada Gambar 3.31, mengedit data toko pada Gambar 3.32, menghapus data toko pada Gambar 3.33, melihat data admin pada Gambar 3.34, menambah data admin pada Gambar 3.35, mengedit data admin pada Gambar 3.36, menghapus data admin pada Gambar 3.37, menerima *preorder* pada Gambar 3.38, mengirim *preorder* pada Gambar 3.39, melihat riwayat *preorder* pada Gambar 3.40, serta pengaturan akun pada Gambar 3.41. *Activity diagram* untuk halaman vendor antara lain melihat data produk pada Gambar 3.42, menerima *preorder* pada Gambar 3.43, melihat riwayat *preorder* pada Gambar 3.44, serta pengaturan akun pada Gambar 3.45. *Activity diagram* untuk halaman toko antara lain mengirim *preorder* pada Gambar 3.46, melihat riwayat *preorder* pada Gambar 3.47, serta pengaturan akun pada Gambar 3.48. *Activity diagram* untuk masing-masing *use case* selengkapnya diuraikan sebagai berikut:

a) *Activity diagram Login*



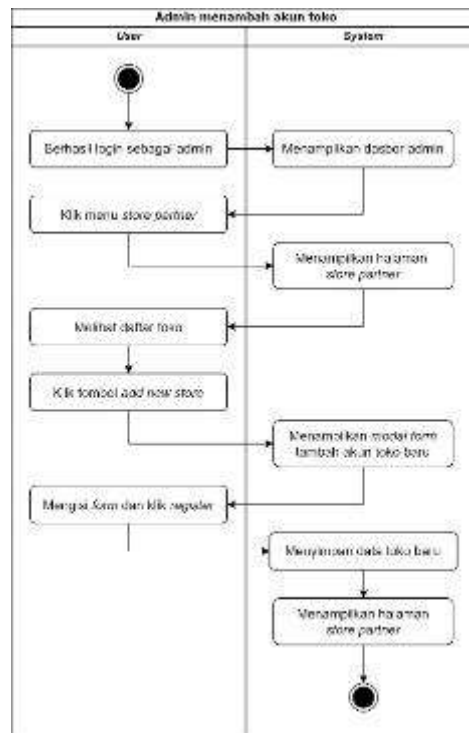
Gambar 3.26. *Activity Diagram Login*

b) *Activity diagram* admin melihat data vendorGambar 3.27. *Activity Diagram* Admin Melihat Data Vendorc) *Activity diagram* admin menambahkan akun vendorGambar 3.28. *Activity Diagram* Admin Menambahkan Akun Vendor

d) *Activity diagram* admin mengedit akun vendorGambar 3.29. *Activity Diagram* Admin Mengedit Akun Vendore) *Activity diagram* admin melihat data tokoGambar 3.30. *Activity Diagram* Admin Melihat Data Toko



f) *Activity diagram* admin menambahkan data toko



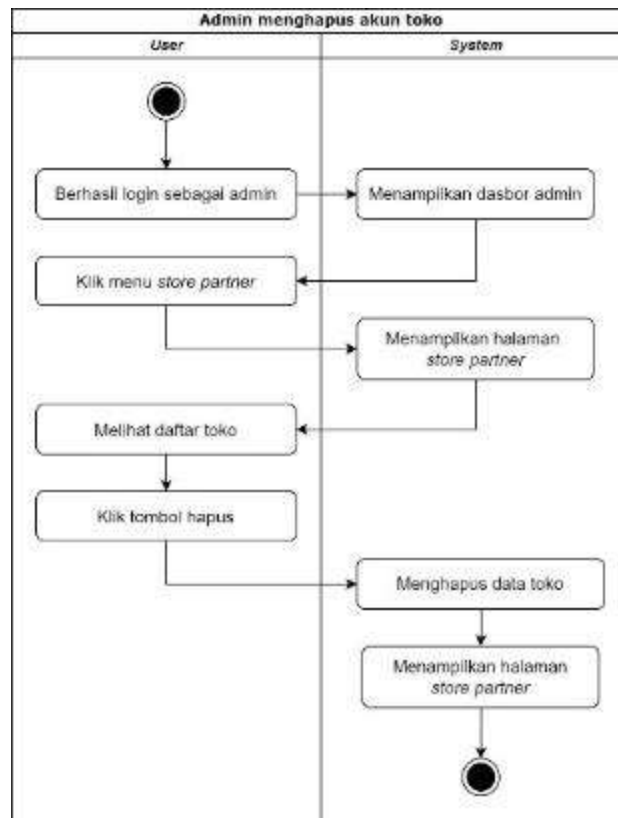
Gambar 3.31. *Activity Diagram* Admin Menambahkan Data Toko

g) *Activity diagram* admin mengedit data toko



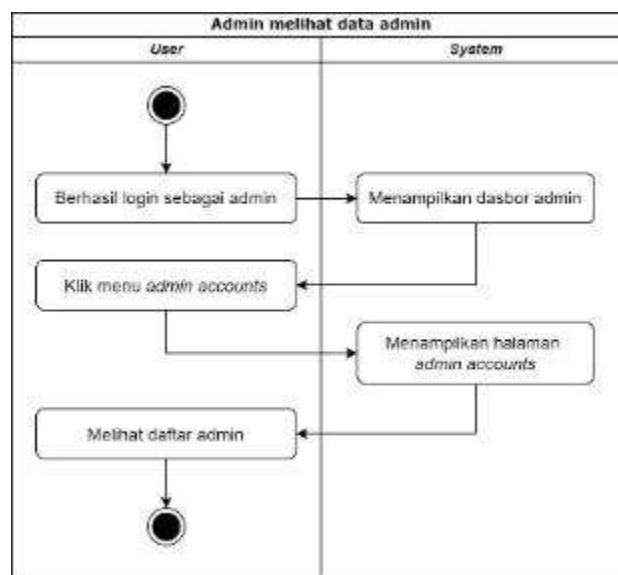
Gambar 3.32. *Activity Diagram* Admin Mengedit Data Toko

h) *Activity diagram* admin menghapus data toko



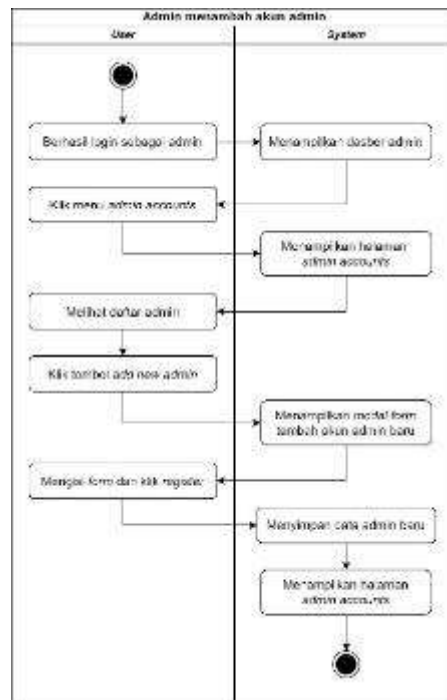
Gambar 3.33. *Activity Diagram* Admin Menghapus Data Toko

i) *Activity diagram* admin melihat data admin



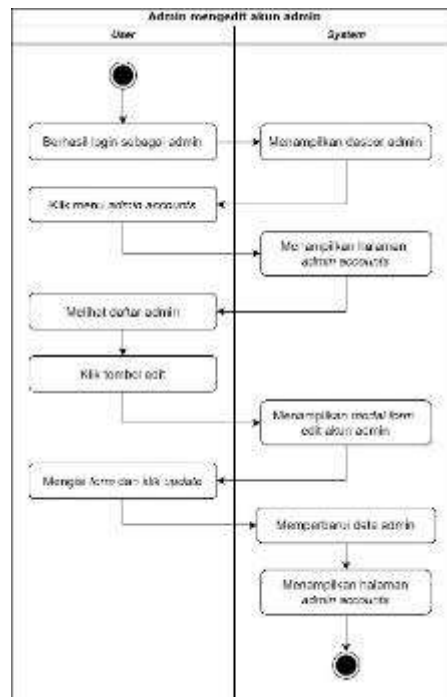
Gambar 3.34. *Activity Diagram* Admin Melihat Data Admin

j) *Activity diagram* admin menambahkan data admin



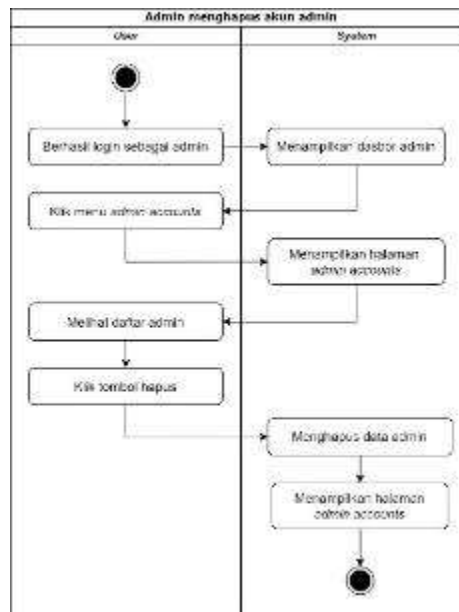
Gambar 3.35. *Activity Diagram* Admin Menambah Data Admin

k) *Activity diagram* admin mengedit data admin



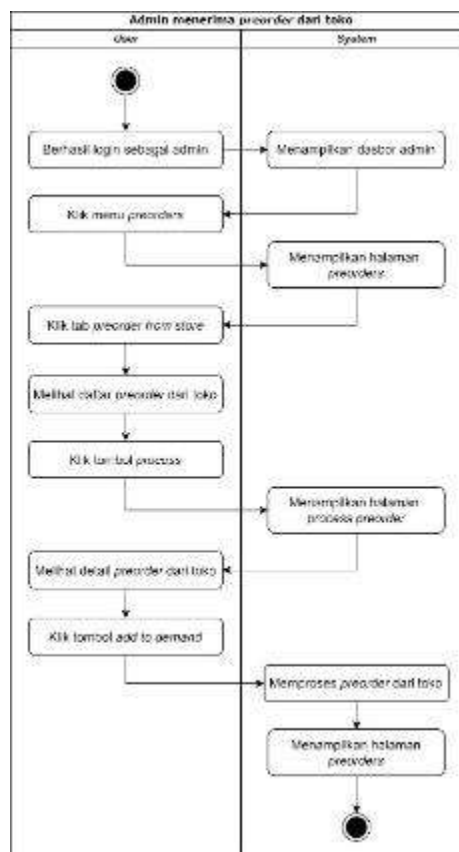
Gambar 3.36. *Activity Diagram* Admin Mengedit Data Admin

1) *Activity diagram* admin menghapus data admin



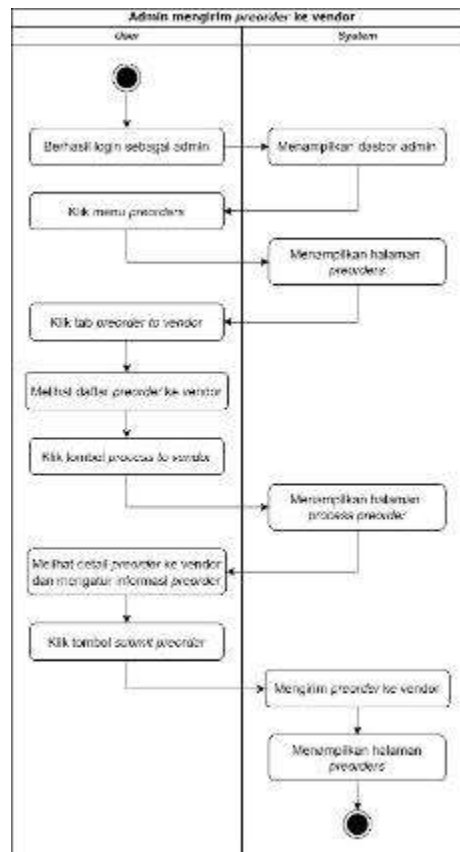
Gambar 3.37. *Activity Diagram* Admin Menghapus Data Admin

m) *Activity diagram* admin menerima preorder dari toko



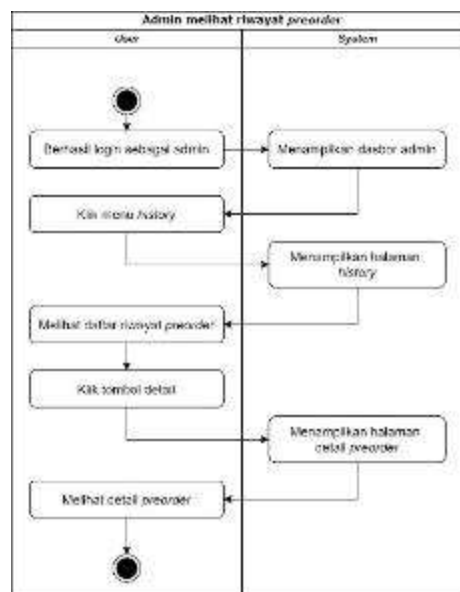
Gambar 3.38. *Activity Diagram* Admin Menerima Preorder

n) *Activity diagram admin mengirim preorder ke vendor*



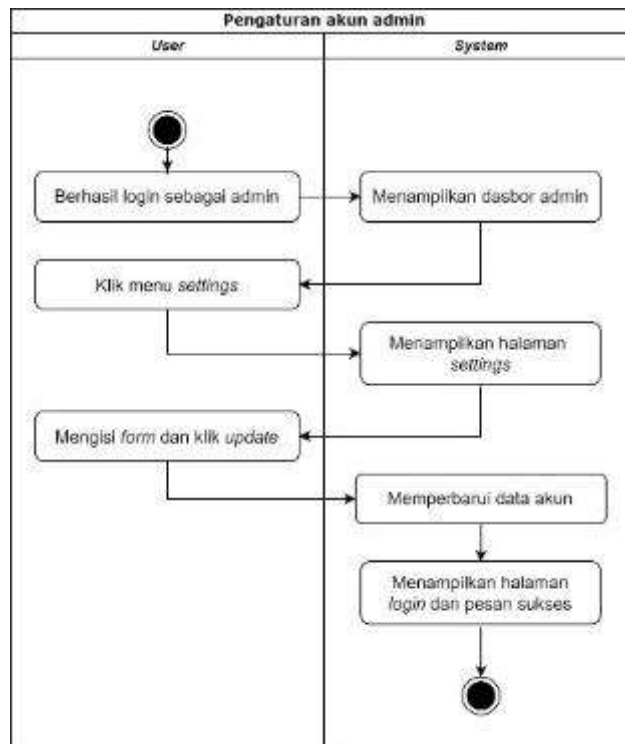
Gambar 3.39. *Activity Diagram Admin Mengirim Preorder*

o) *Activity diagram admin melihat riwayat preorder*



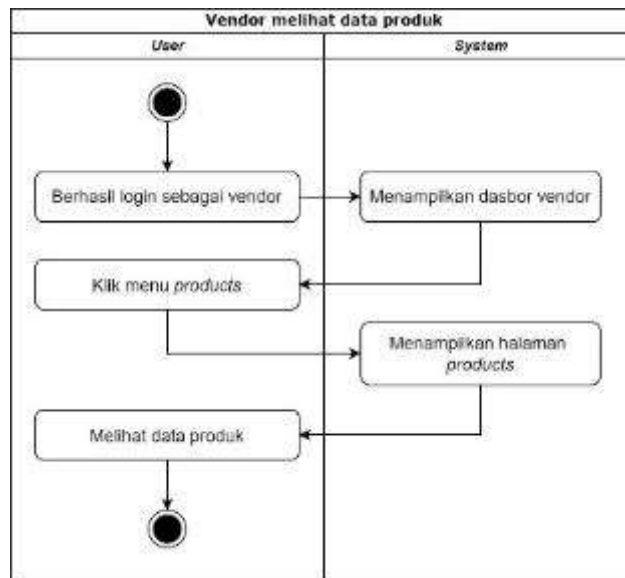
Gambar 3.40. *Activity Diagram Admin Melihat Riwayat Preorder*

p) *Activity diagram* pengaturan akun admin



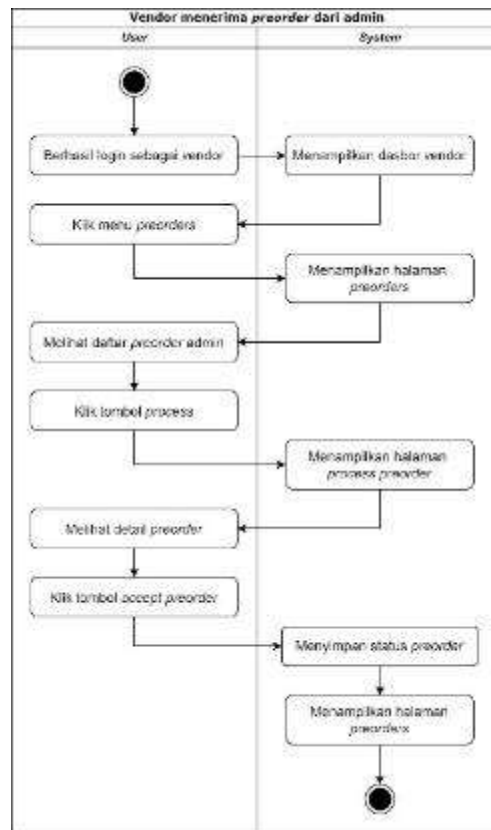
Gambar 3.41. *Activity Diagram* Pengaturan Akun Admin

q) *Activity Diagram* vendor melihat data produk



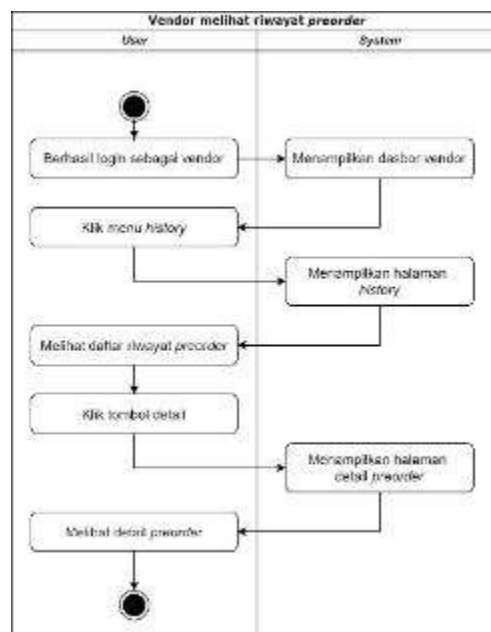
Gambar 3.42. *Activity Diagram* Vendor Melihat Data Produk

r) *Activity diagram* vendor menerima *preorder* dari admin



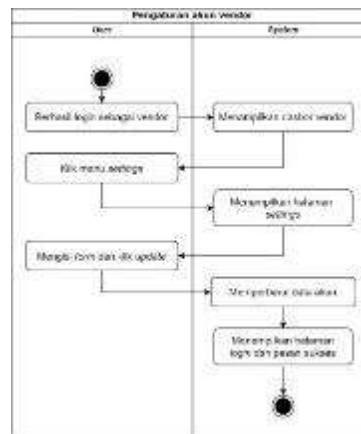
Gambar 3.43. *Activity Diagram* Vendor Menerima Preorder

s) *Activity diagram* vendor melihat riwayat preorder



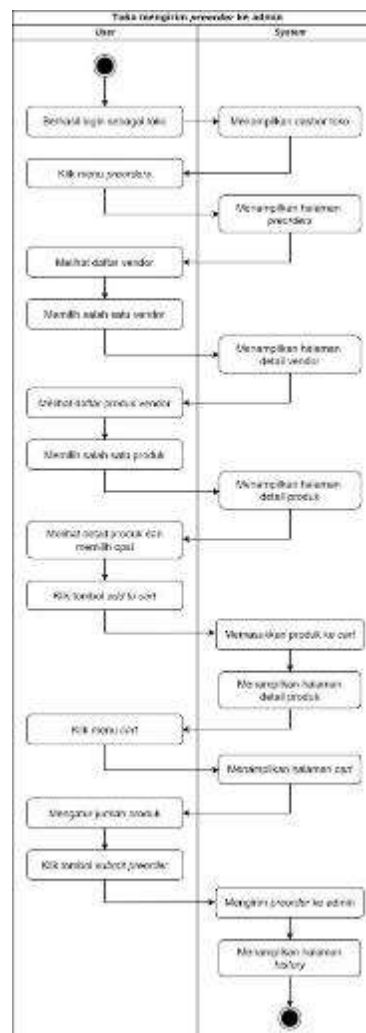
Gambar 3.44. *Activity Diagram* Vendor Melihat Riwayat Preorder

t) *Activity diagram* pengaturan akun vendor



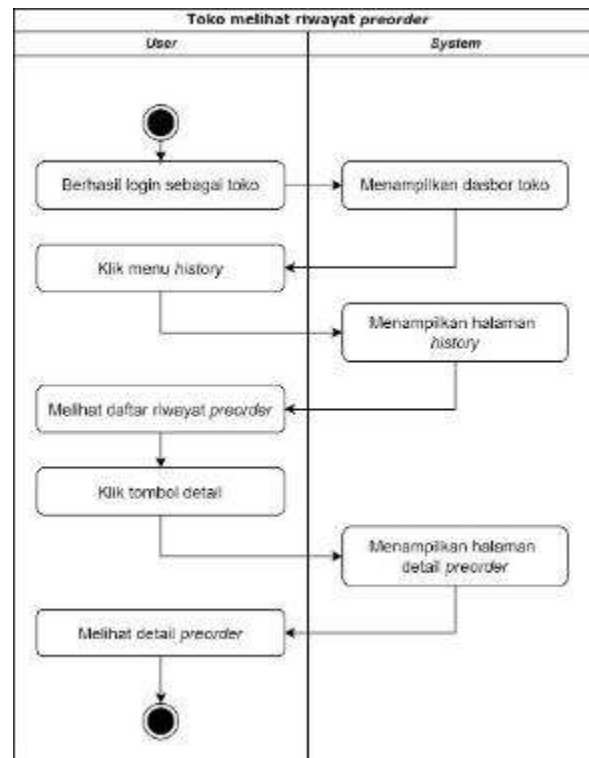
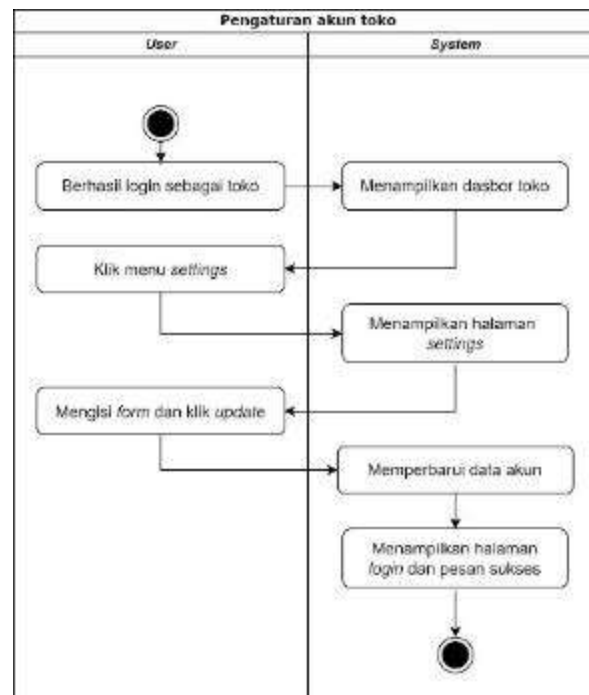
Gambar 3.45. *Activity Diagram* Pengaturan Akun Vendor

u) *Activity diagram* toko mengirim *preorder* ke admin



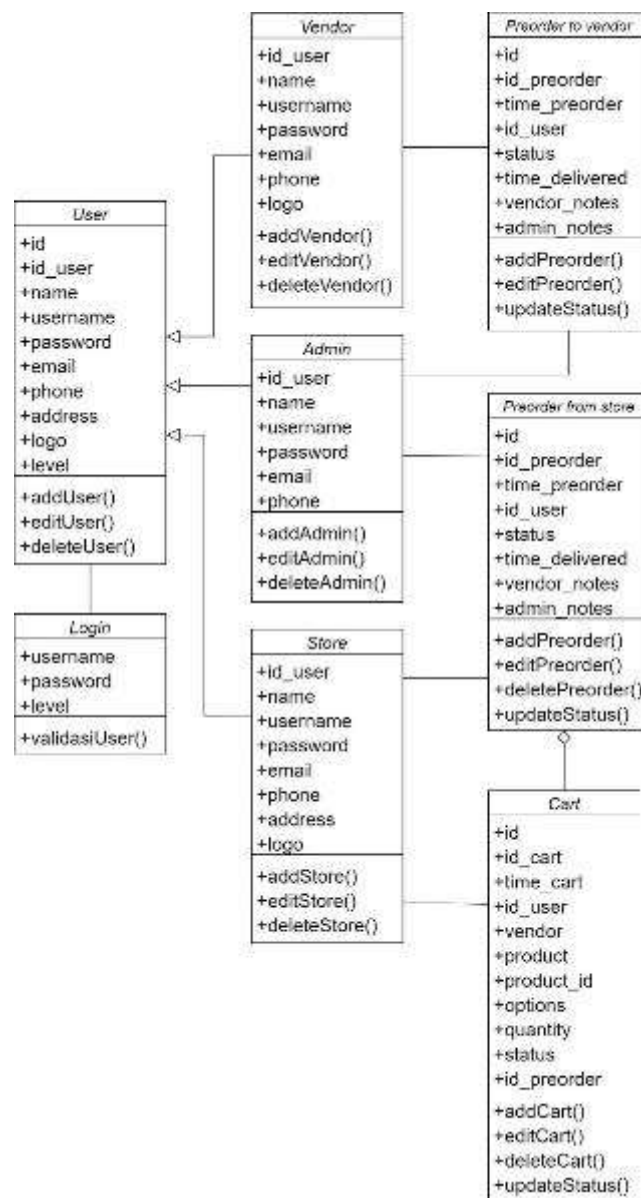
Gambar 3.46. *Activity Diagram* Toko Mengirim Preorder



v) *Activity diagram toko melihat riwayat preorder*Gambar 3.47. *Activity Diagram Toko Melihat Riwayat Preorder*w) *Activity diagram pengaturan akun toko*Gambar 3.48. *Activity Diagram Pengaturan Akun Toko*

#### 4) Class diagram

Diagram ini digunakan untuk menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan (metode/fungsi). Beberapa kelas pada aplikasi ini antara lain kelas *user*, kelas *login*, kelas *admin*, kelas *vendor*, kelas  *toko*, kelas *keranjang*, kelas *preorder* dari toko, dan kelas *preorder* ke vendor. *Class diagram* yang dibuat untuk perancangan aplikasi ini dapat dilihat pada Gambar 3.49.



Gambar 3.49. Class Diagram

b. Desain *database*

Proses perancangan berikutnya adalah pembuatan desain *database*. *Database* berfungsi sebagai basis penyimpanan data dalam suatu sistem informasi. Sementara itu tujuan dari desain *database* adalah untuk menentukan data-data yang dibutuhkan dalam sistem, sehingga informasi yang dihasilkan dapat terpenuhi [29]. Desain *database* berupa tabel berisi informasi yang terdiri dari kolom (*field*), tipe data (*type*), lebar data (*length*), dan catatan (*notes*). Terdapat tiga tabel dalam *database* aplikasi berbasis web yang dikembangkan pada penelitian ini antara lain tabel pengguna (*user*) yang ditampilkan pada Tabel 3.25, tabel keranjang (*cart*) yang ditampilkan pada Tabel 3.26, dan tabel prapemesanan (*preorder*) yang ditampilkan pada Tabel 3.27. Desain *database* dari aplikasi ini antara lain diuraikan sebagai berikut:

1) Tabel pengguna (*user*)

Tabel 3.25. Tabel *Database* Pengguna

<i>Field</i>	<i>Type</i>	<i>Length</i>	<i>Note</i>
<i>id</i>	<i>int</i>	10	ID record ( <i>Primary Key</i> )
<i>id_user</i>	<i>varchar</i>	30	ID pengguna ( <i>Unique</i> )
<i>level</i>	<i>varchar</i>	20	Level pengguna: admin, vendor, toko
<i>name</i>	<i>varchar</i>	100	Nama pengguna
<i>username</i>	<i>varchar</i>	20	Username pengguna ( <i>Unique</i> )
<i>password</i>	<i>varchar</i>	20	Password pengguna
<i>email</i>	<i>varchar</i>	50	Email pengguna
<i>phone</i>	<i>varchar</i>	15	Nomor telepon pengguna
<i>address</i>	<i>varchar</i>	100	Alamat pengguna
<i>logo</i>	<i>varchar</i>	500	Logo bagi pengguna yang memiliki logo

2) Tabel keranjang (*cart*)Tabel 3.26. Tabel *Database* Keranjang

<b>Field</b>	<b>Type</b>	<b>Length</b>	<b>Note</b>
<i>id</i>	<i>int</i>	10	ID <i>record</i> ( <i>Primary Key</i> )
<i>id_cart</i>	<i>varchar</i>	50	ID keranjang ( <i>Unique</i> )
<i>time_cart</i>	<i>datetime</i>	-	Waktu barang masuk keranjang
<i>id_user</i>	<i>varchar</i>	30	ID pemesan
<i>name</i>	<i>varchar</i>	100	Nama pemesan
<i>vendor</i>	<i>varchar</i>	100	Vendor yang dipilih
<i>product</i>	<i>varchar</i>	100	Produk yang dipesan
<i>handle</i>	<i>varchar</i>	100	<i>Username</i> untuk produk
<i>product_id</i>	<i>varchar</i>	30	ID produk
<i>options</i>	<i>varchar</i>	50	Opsi varian produk
<i>product_photo</i>	<i>varchar</i>	500	URL foto produk
<i>price</i>	<i>varchar</i>	10	Harga pokok produk
<i>quantity</i>	<i>varchar</i>	10	Jumlah produk yang dipesan
<i>subtotal</i>	<i>varchar</i>	10	Jumlah harga per item
<i>amount</i>	<i>varchar</i>	10	Jumlah seluruh tagihan
<i>status</i>	<i>varchar</i>	30	Status pemesanan
<i>id_preorder</i>	<i>varchar</i>	30	ID pemesanan
<i>time_preorder</i>	<i>datetime</i>	-	Waktu pemesanan ( <i>checkout</i> )
<i>id_to_vendor</i>	<i>varchar</i>	30	ID pemesanan ke vendor
<i>time_to_vendor</i>	<i>datetime</i>	-	Waktu pemesanan ke vendor
<i>qty_to_vendor</i>	<i>varchar</i>	11	Jumlah produk yang dipesan admin ke vendor

3) Tabel prapemesanan (*preorder*)Tabel 3.27. Tabel *Database* Prapemesanan

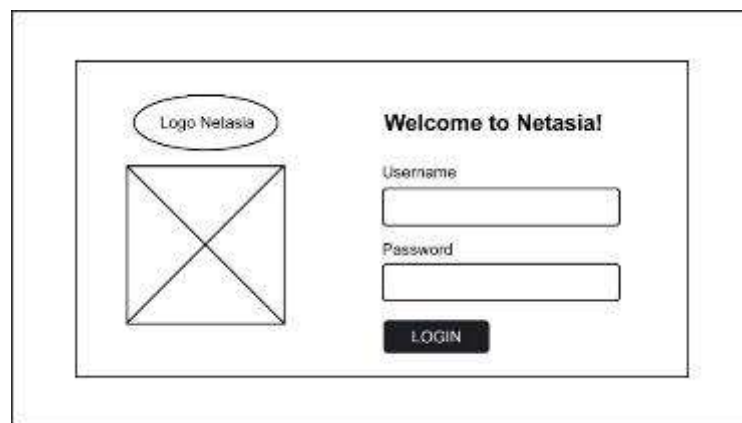
Field	Type	Length	Note
<i>id</i>	<i>int</i>	10	ID <i>record</i> ( <i>Primary Key</i> )
<i>id_preorder</i>	<i>varchar</i>	30	ID pemesanan
<i>preorder_time</i>	<i>datetime</i>	-	Waktu pemesanan ( <i>checkout</i> )
<i>id_user</i>	<i>varchar</i>	30	ID pemesan
<i>name</i>	<i>varchar</i>	100	Nama pemesan
<i>amount</i>	<i>varchar</i>	10	Jumlah seluruh tagihan
<i>status</i>	<i>varchar</i>	30	Status pemesanan
<i>time_delivered</i>	<i>datetime</i>	-	Waktu vendor mengirim pesanan
<i>vendor_notes</i>	<i>varchar</i>	250	Catatan dari vendor
<i>admin_notes</i>	<i>varchar</i>	250	Catatan dari admin
<i>update_notes1</i>	<i>datetime</i>	-	Waktu terakhir <i>update</i> catatan vendor
<i>update_notes2</i>	<i>datetime</i>	-	Waktu terakhir <i>update</i> catatan admin

c. Desain antarmuka pengguna (*user interface*)

Setelah membuat model sistem dengan diagram-diagram UML dan membuat desain *database*, proses perancangan berikutnya adalah pembuatan desain antarmuka pengguna (*user interface*). Terdapat empat halaman utama dalam aplikasi ini, yakni halaman *login*, halaman admin, halaman vendor, dan halaman toko. Masing-masing halaman tersebut memiliki beberapa menu dengan fungsi yang berbeda-beda. Halaman utama beserta masing-masing menu dalam aplikasi berbasis web yang dikembangkan pada penelitian ini diuraikan sebagai berikut:

### 1) Halaman *login*

Halaman *login* merupakan halaman yang pertama kali ditampilkan saat pengguna mengakses aplikasi berbasis web ini. Halaman ini berisi formulir yang terdiri dari kolom untuk memasukkan *username* dan *password* serta tombol untuk *login*. Desain halaman *login* dapat dilihat pada Gambar 3.50.



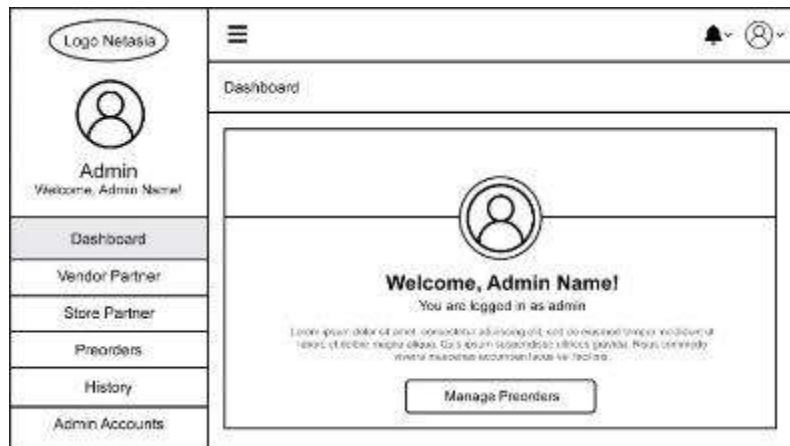
Gambar 3.50. Desain Halaman *Login*

### 2) Halaman admin

Halaman admin memiliki enam menu utama yakni dasbor (*dashboard*), daftar vendor (*vendor partner*), daftar toko (*store partner*), prapemesanan (*preorders*), riwayat prapemesanan (*history*), dan akun admin (*admin accounts*). Selain itu halaman admin juga memiliki beberapa fitur lain seperti detail vendor, detail toko, pemrosesan *preorder*, pengiriman *preorder* ke vendor, detail *preorder*, dan pengaturan akun. Desain halaman yang ada di level admin diuraikan sebagai berikut:

#### a) Halaman dasbor admin (*dashboard*)

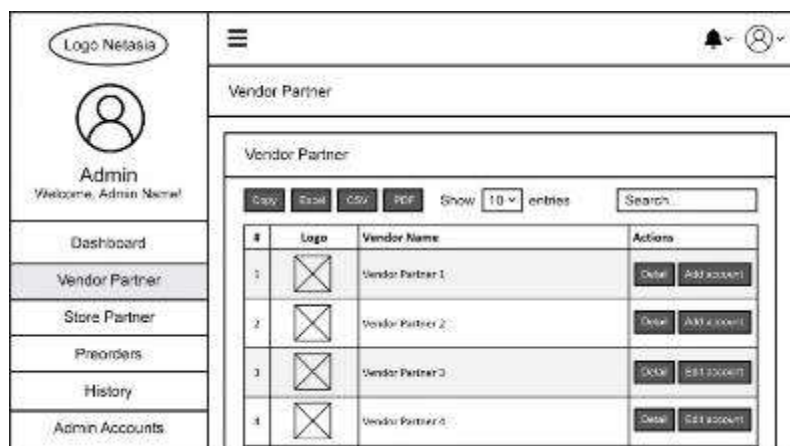
Halaman dasbor admin berisi pesan selamat datang, deskripsi peran admin, dan tombol pintasan untuk menuju ke fitur utama sebagai admin yaitu halaman *preorder* untuk mengelola *preorder* dari toko dan ke vendor. Desain halaman dasbor admin dapat dilihat pada Gambar 3.51.



Gambar 3.51. Desain Halaman Dasbor Admin

b) Halaman daftar vendor (*vendor partner*)

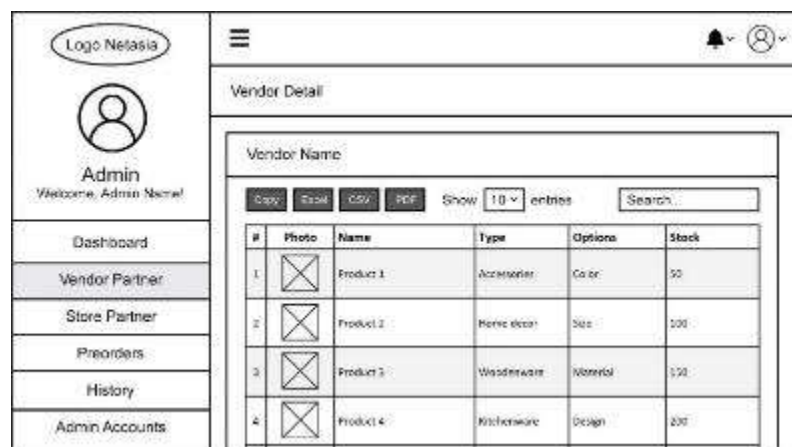
Halaman daftar vendor berisi daftar vendor UKM yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain logo vendor, nama vendor, dan aksi. Selain itu terdapat juga tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, *CSV*, dan *PDF*, serta kolom pencarian vendor. Desain halaman daftar vendor dapat dilihat pada Gambar 3.52.



Gambar 3.52. Desain Halaman Daftar Vendor di Admin

c) Halaman detail vendor (*vendor detail*)

Halaman detail vendor berisi daftar produk dari vendor UKM yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama produk, tipe produk, opsi produk, dan stok produk. Selain itu terdapat juga tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, *CSV*, dan *PDF*, serta kolom pencarian produk. Desain halaman detail vendor dapat dilihat pada Gambar 3.53.



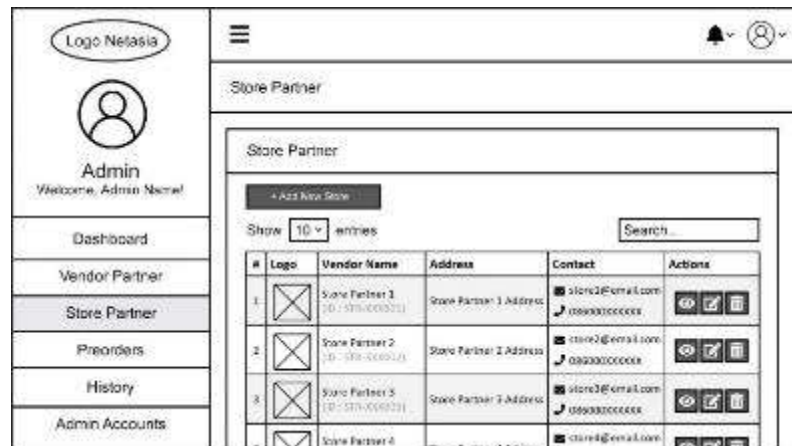
#	Photo	Name	Type	Options	Stock
1		Product 1	Accessories	Color	50
2		Product 2	Home Decor	Size	100
3		Product 3	Woodenware	Material	150
4		Product 4	Kitchenware	Design	200

Gambar 3.53. Desain Halaman Detail Vendor di Admin

d) Halaman daftar toko (*store partner*)

Halaman daftar toko berisi daftar toko yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain logo toko, nama toko, alamat toko, kontak toko, dan aksi. Terdapat juga tombol untuk menambah toko baru (*add new store*), tombol detail, edit, dan hapus pada masing-masing toko, serta kolom pencarian toko. Desain halaman daftar toko dapat dilihat pada Gambar 3.54.

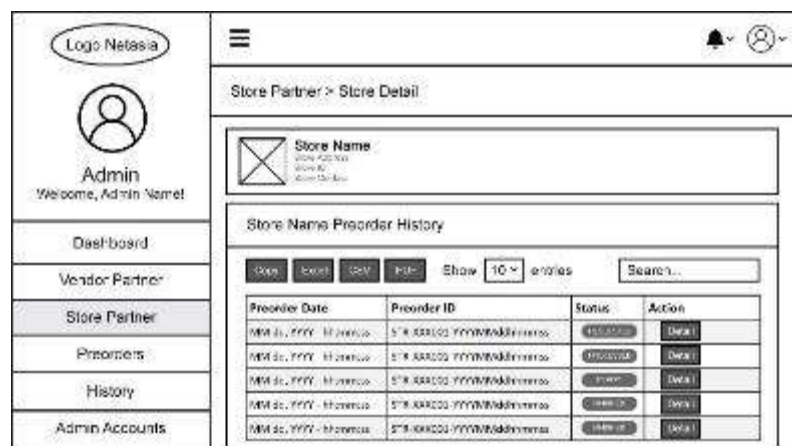




Gambar 3.54. Desain Halaman Daftar Toko di Admin

e) Halaman detail toko (*store detail*)

Halaman detail toko berisi detail toko yang terdiri dari nama toko, alamat toko, id toko, dan kontak toko serta daftar riwayat *preorder* toko yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain tanggal *preorder*, id *preorder*, status *preorder*, dan aksi. Terdapat juga tombol detail pada masing-masing *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, CSV, dan PDF, serta kolom pencarian *preorder*. Desain halaman detail toko dapat dilihat pada Gambar 3.55.



Gambar 3.55. Desain Halaman Detail Toko di Admin

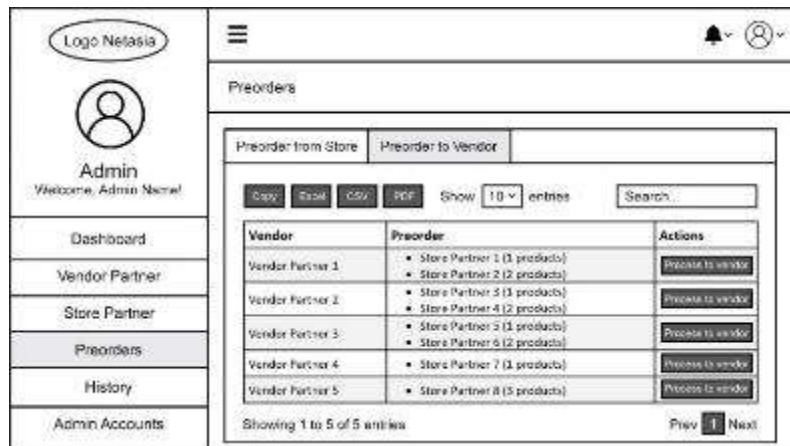
f) Halaman prapemesanan (*preorders*)

Halaman prapemesanan terdiri dari dua tab, yakni tab *preorder* dari toko (*preorder from store*) dan tab *preorder* ke vendor (*preorder to vendor*). Tab *preorder* dari toko berisi daftar permintaan *preorder* dari toko yang disajikan dalam tabel. Kolom yang ada di tabel tersebut antara lain tanggal *preorder*, nama toko, id *preorder*, status, dan aksi. Terdapat juga tombol proses *preorder* (*process*) pada setiap *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, CSV, dan PDF, serta kolom pencarian. Desain tab *preorder* dari toko dapat dilihat pada Gambar 3.56.

Preorder Date	Store	Preorder ID	Status	Actions
MM/dd/yyyy - hh:mm:ss	Store Partner 1	STP-000001-YYYYMMDDhhmmss	Pending	Process
MM/dd/yyyy - hh:mm:ss	Store Partner 2	STP-000002-YYYYMMDDhhmmss	Pending	Process
MM/dd/yyyy - hh:mm:ss	Store Partner 3	STP-000003-YYYYMMDDhhmmss	Pending	Process
MM/dd/yyyy - hh:mm:ss	Store Partner 4	STP-000004-YYYYMMDDhhmmss	Pending	Process
MM/dd/yyyy - hh:mm:ss	Store Partner 5	STP-000005-YYYYMMDDhhmmss	Pending	Process

Gambar 3.56. Desain Tab *Preorder* dari Toko

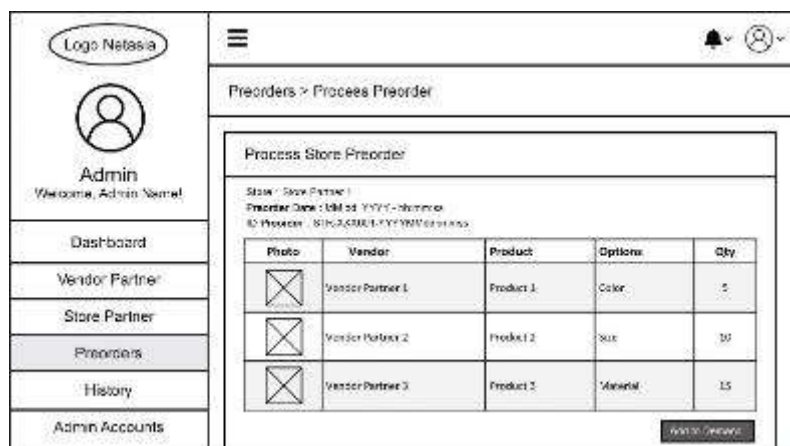
Sementara itu tab *preorder* ke vendor berisi daftar *preorder* dari toko untuk diteruskan ke vendor yang disajikan dalam tabel. Kolom yang ada di tabel tersebut antara lain nama vendor, keterangan *preorder*, dan aksi. Terdapat juga tombol proses *preorder* ke vendor (*process to vendor*) pada masing-masing *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, CSV, dan PDF, serta kolom pencarian *preorder*. Desain tab *preorder* ke vendor di halaman prapemesanan dapat dilihat pada Gambar 3.57.



Gambar 3.57. Desain Tab *Preorder* ke Vendor

g) Halaman proses *preorder* dari toko

Halaman proses *preorder* dari toko berisi daftar produk pesanan toko yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama vendor, nama produk, opsi produk, dan jumlah produk yang dipesan. Terdapat juga tombol untuk meneruskan pesanan toko ke vendor (*add to demand*). Desain halaman proses *preorder* dari toko dapat dilihat pada Gambar 3.58.

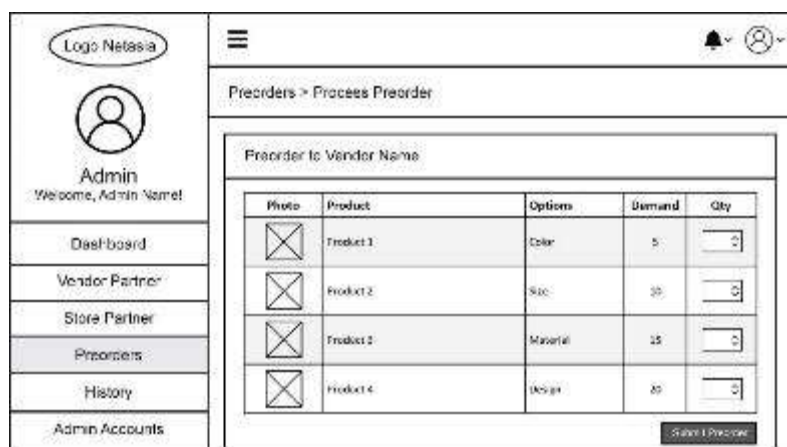


Gambar 3.58. Desain Halaman Proses *Preorder* dari Toko

h) Halaman proses *preorder* ke vendor

Halaman proses *preorder* ke vendor berisi daftar produk pesanan toko yang akan diteruskan kepada vendor

yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama produk, opsi produk, total jumlah permintaan dari toko, dan jumlah produk yang dipesan ke vendor. Terdapat juga tombol untuk mengirim *preorder* ke vendor (*submit preorder*). Desain halaman proses *preorder* ke vendor dapat dilihat pada Gambar 3.59.



Gambar 3.59. Desain Halaman Proses *Preorder* ke Vendor

i) Halaman riwayat prapemesanan (*history*)

Halaman riwayat prapemesanan terdiri dari dua tab, yakni *preorder* dari toko (*preorder from store*) dan *preorder* ke vendor (*preorder to vendor*). Tab *preorder* dari toko berisi daftar riwayat *preorder* dari toko yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain tanggal *preorder*, nama toko, id *preorder*, status, dan aksi. Terdapat juga tombol detail pada masing-masing *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, *CSV*, dan *PDF*, serta kolom pencarian *preorder*. Desain tab *preorder* dari toko di halaman riwayat prapemesanan dapat dilihat pada Gambar 3.60.

Preorder Data	Store	Preorder ID	Status	Actions
Metasys, MYR - Indonesia	Store Partner 1	STR-00001-MYR-INDONESIA	Finished	Detail
Metasys, MYR - Indonesia	Store Partner 2	STR-00002-MYR-INDONESIA	Ready	Detail
Metasys, MYR - Indonesia	Store Partner 3	STR-00003-MYR-INDONESIA	Ready	Detail
Metasys, MYR - Indonesia	Store Partner 4	STR-00004-MYR-INDONESIA	Ready	Detail
Metasys, MYR - Indonesia	Store Partner 5	STR-00005-MYR-INDONESIA	Ready	Detail

Gambar 3.60. Desain Tab Riwayat *Preorder* dari Toko

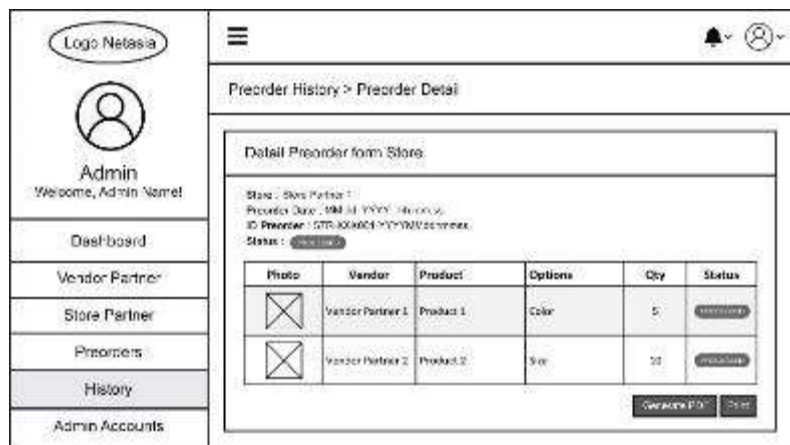
Sementara itu tab *preorder* ke vendor berisi daftar riwayat *preorder* ke vendor yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain tanggal *preorder*, nama vendor, id *preorder*, status, dan aksi. Terdapat juga tombol detail pada masing-masing *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, *CSV*, dan *PDF*, serta kolom pencarian *preorder*. Desain tab *preorder* ke vendor di halaman riwayat prapemesanan dapat dilihat pada Gambar 3.61.

Preorder Date	Vendor	Preorder ID	Status	Actions
09/01/2020 - Indonesia	Vendor Partner 1	AVR-00001-MYR-INDONESIA	Received	Detail
09/01/2020 - Indonesia	Vendor Partner 2	AVR-00002-MYR-INDONESIA	Accepted	Detail
09/01/2020 - Indonesia	Vendor Partner 3	AVR-00003-MYR-INDONESIA	Declined	Detail
09/01/2020 - Indonesia	Vendor Partner 4	AVR-00004-MYR-INDONESIA	Declined	Detail
09/01/2020 - Indonesia	Vendor Partner 5	AVR-00005-MYR-INDONESIA	Requested	Detail

Gambar 3.61. Desain Tab Riwayat *Preorder* ke Vendor

j) Halaman detail *preorder* dari toko

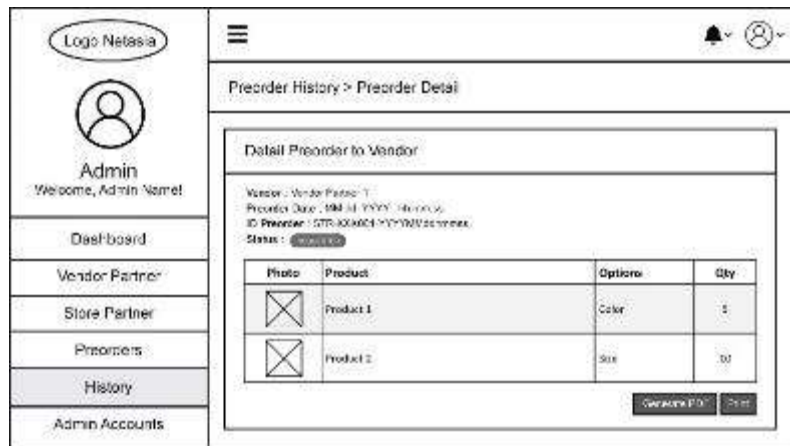
Halaman detail *preorder* dari toko berisi informasi *preorder* yang terdiri dari nama toko, tanggal *preorder*, id *preorder*, dan status *preorder*, serta daftar produk pesanan toko yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama vendor, nama produk, opsi produk, jumlah produk yang dipesan, dan status produk. Terdapat juga tombol untuk mencetak halaman (*print*) dan tombol ekspor halaman ke PDF. Desain halaman detail *preorder* dari toko dapat dilihat pada Gambar 3.62.



Gambar 3.62. Desain Halaman Detail *Preorder* dari Toko

k) Halaman detail *preorder* ke vendor

Halaman detail *preorder* ke vendor berisi informasi *preorder* yang terdiri dari nama vendor, tanggal *preorder*, id *preorder*, dan status, serta daftar produk pesanan admin ke vendor yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama produk, opsi produk, dan jumlah produk yang dipesan. Terdapat juga tombol untuk mencetak halaman (*print*) dan tombol ekspor halaman ke PDF. Desain halaman detail *preorder* ke vendor dapat dilihat pada Gambar 3.63.



Gambar 3.63. Desain Halaman Detail *Preorder* ke Vendor

#### 1) Halaman daftar admin (*admin accounts*)

Halaman daftar admin berisi daftar admin yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain id admin, nama admin, *username* admin, kontak admin, dan aksi. Terdapat juga tombol untuk menambah admin baru (*add new admin*), tombol edit dan hapus pada setiap admin, serta kolom pencarian admin. Desain halaman daftar admin dapat dilihat pada Gambar 3.64.



Gambar 3.64. Desain Halaman Daftar Admin

m) Halaman pengaturan akun admin (*settings*)

Halaman pengaturan akun admin berisi informasi akun yang terdiri dari nama admin, *username*, email, dan nomor telepon, formulir untuk memperbarui akun (*update account*) yang terdiri dari kolom untuk memasukkan nama, email, nomor telepon, *username*, dan password, serta tombol untuk memperbarui akun (*update*). Desain halaman pengaturan akun admin dapat dilihat pada Gambar 3.65.



Gambar 3.65. Desain Halaman Pengaturan Akun Admin

3) Halaman vendor

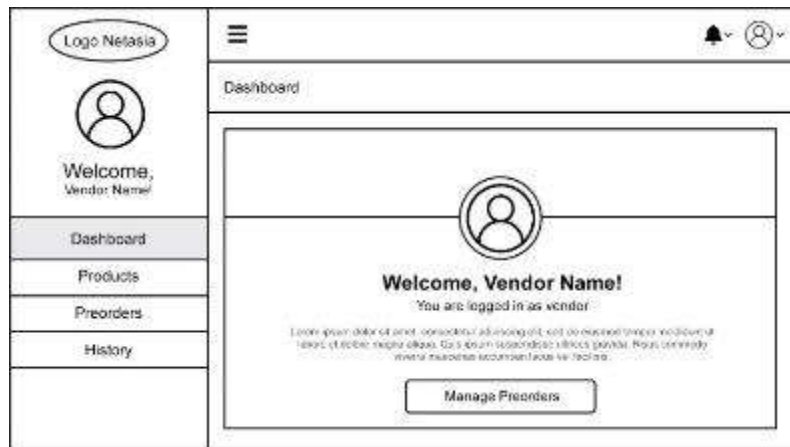
Halaman vendor memiliki empat menu utama antara lain dasbor (*dashboard*), daftar produk (*products*), prapemesanan dari admin (*preorders*), dan riwayat rapemesanan (*history*). Selain itu halaman vendor juga memiliki beberapa fitur lain seperti pemrosesan *preorder*, detail *preorder*, dan pengaturan akun. Desain halaman yang ada di level vendor diuraikan sebagai berikut:

a) Halaman dasbor vendor (*dashboard*)

Halaman dasbor vendor berisi pesan selamat datang, deskripsi peran vendor, dan tombol pintasan untuk menuju ke fitur utama sebagai vendor yaitu halaman *preorder* untuk



mengelola *preorder* dari admin. Desain halaman dasbor vendor dapat dilihat pada Gambar 3.66.



Gambar 3.66. Desain Halaman Dasbor Vendor

b) Halaman daftar produk (*products*)

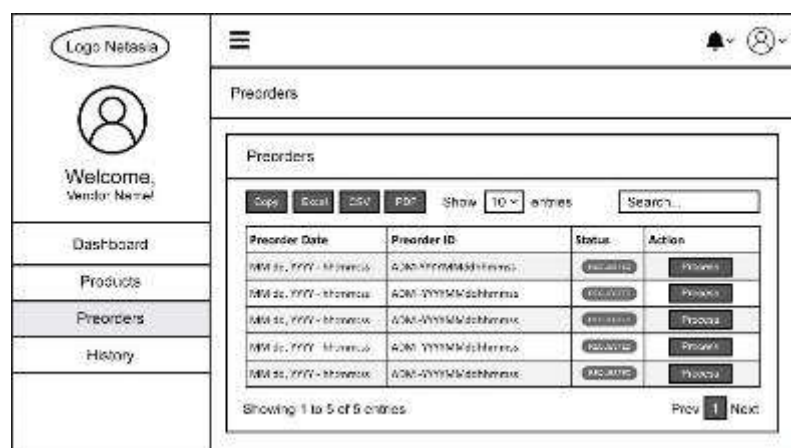
Halaman daftar produk berisi daftar produk vendor UKM yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama produk, tipe produk, opsi produk, dan stok produk. Selain itu terdapat juga tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, *CSV*, dan *PDF*, serta kolom pencarian produk. Desain halaman daftar produk dapat dilihat pada Gambar 3.67.

#	Photo	Name	Type	Options	Stock
1	⊗	Product 1	Accessories	Color	50
2	⊗	Product 2	Home decor	Size	100
3	⊗	Product 3	Weddingware	Material	150
4	⊗	Product 4	Kitchenware	Design	200

Gambar 3.67. Desain Halaman Daftar Produk di Vendor

c) Halaman *preorder* vendor (*preorders*)

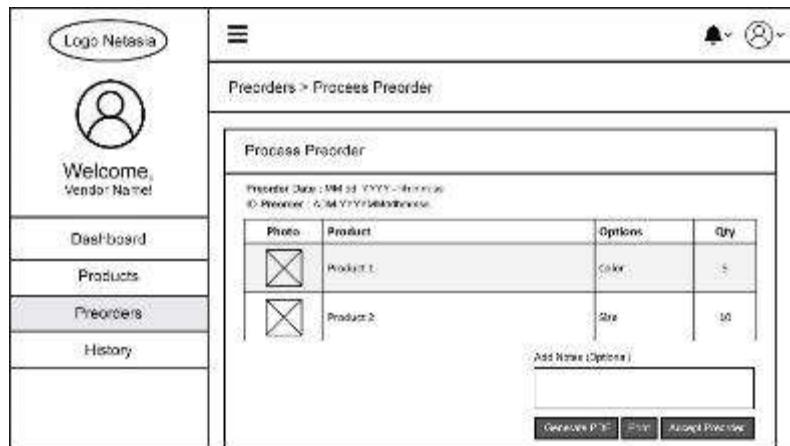
Halaman *preorder* vendor berisi daftar permintaan *preorder* dari admin yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain tanggal *preorder*, id *preorder*, status, dan aksi. Terdapat juga tombol proses *preorder* (*process*) pada masing-masing *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, *CSV*, dan *PDF*, serta kolom pencarian *preorder*. Desain halaman *preorder* vendor dapat dilihat pada Gambar 3.68.



Gambar 3.68. Desain Halaman *Preorder* di Vendor

d) Halaman proses *preorder* vendor

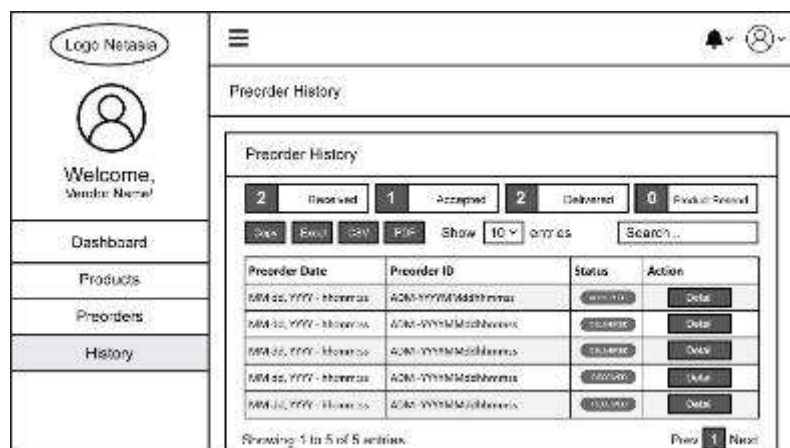
Halaman proses *preorder* vendor berisi daftar produk pesanan admin yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama produk, opsi produk, dan jumlah produk. Terdapat juga tombol untuk menerima *preorder* (*accept preorder*). Desain halaman proses *preorder* vendor dapat dilihat pada Gambar 3.69.



Gambar 3.69. Desain Halaman Proses *Preorder* di Vendor

e) Halaman riwayat *preorder* vendor (*history*)

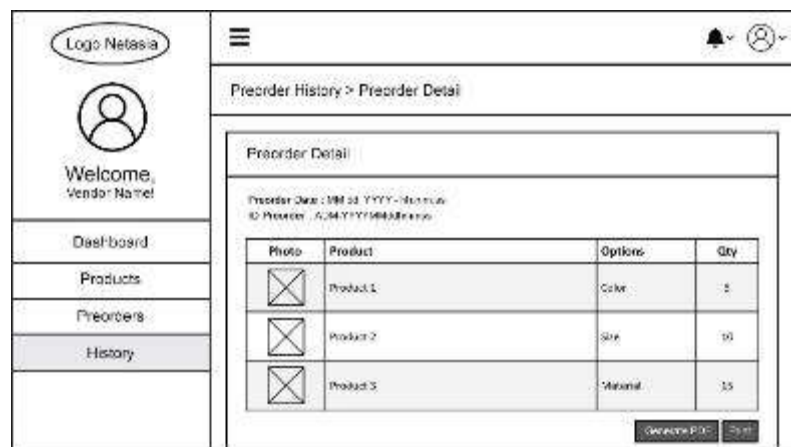
Halaman riwayat *preorder* vendor berisi daftar riwayat *preorder* vendor yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain tanggal *preorder*, id *preorder*, status, dan aksi. Terdapat juga tombol untuk detail atau kirim (*deliver*) pada masing-masing *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, CSV, dan PDF, serta kolom pencarian *preorder*. Desain halaman riwayat *preorder* vendor dapat dilihat pada Gambar 3.70.



Gambar 3.70. Desain Halaman Riwayat *Preorder* di Vendor

f) Halaman detail *preorder* vendor

Halaman detail *preorder* vendor berisi informasi *preorder* yang terdiri dari tanggal *preorder* dan id *preorder*, serta daftar produk pesanan admin yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama produk, opsi produk, dan jumlah produk yang dipesan. Terdapat juga tombol untuk mencetak halaman (*print*) dan tombol ekspor halaman ke PDF. Desain halaman detail *preorder* vendor dapat dilihat pada Gambar 3.71.



Gambar 3.71. Desain Halaman Detail *Preorder* di Vendor

g) Halaman pengaturan akun vendor (*settings*)

Halaman pengaturan akun vendor berisi informasi akun yang terdiri dari nama vendor, *username*, email, dan nomor telepon, formulir untuk memperbarui akun (*update account*) yang terdiri dari kolom untuk memasukkan email, nomor telepon, *username*, dan password, serta tombol untuk memperbarui akun (*update*). Desain halaman pengaturan akun vendor dapat dilihat pada Gambar 3.72.



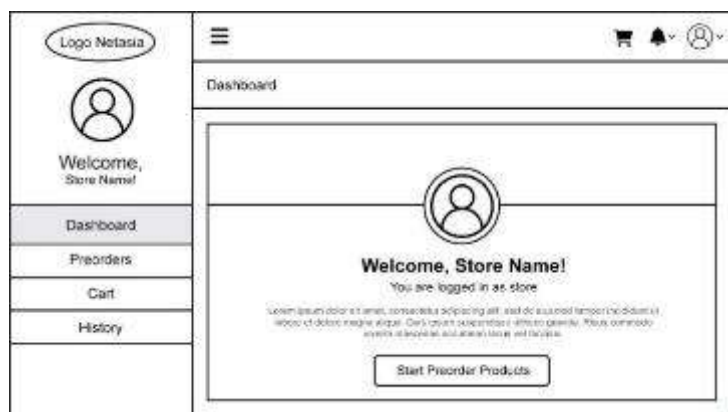
Gambar 3.72. Desain Halaman Pengaturan Akun Vendor

#### 4) Halaman toko

Halaman toko memiliki beberapa menu antara lain dasbor (*dashboard*), prapemesanan (*preorders*), keranjang (*cart*), dan riwayat prapemesanan (*history*). Selain itu halaman toko juga memiliki beberapa fitur lain seperti daftar pilihan produk, detail produk, detail *preorder*, dan pengaturan akun. Desain halaman yang ada di level toko diuraikan sebagai berikut:

##### a) Halaman dasbor toko (*dashboard*)

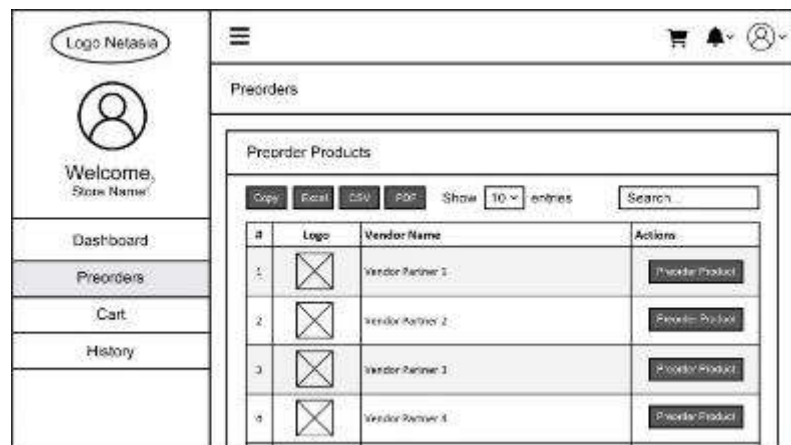
Halaman dasbor toko berisi pesan selamat datang, deskripsi peran toko, dan tombol pintasan untuk menuju ke fitur utama sebagai toko yaitu halaman *preorder* untuk memesan produk kepada admin. Desain halaman dasbor toko dapat dilihat pada Gambar 3.73.



Gambar 3.73. Desain Halaman Dasbor Toko

### b) Halaman *preorder* toko

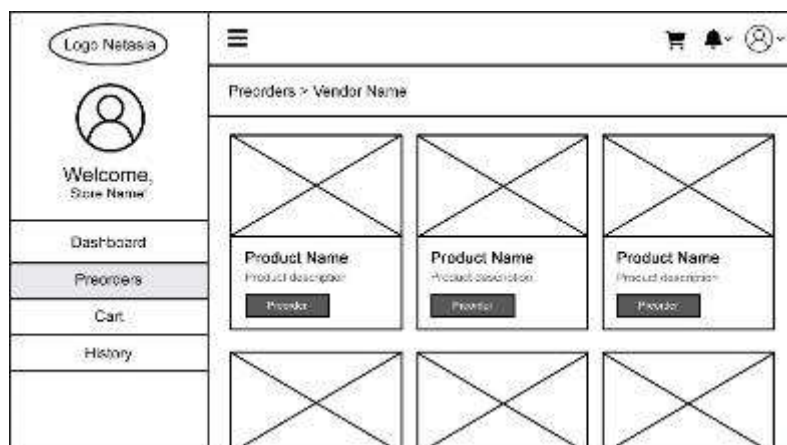
Halaman *preorder* toko berisi daftar vendor UKM yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain logo vendor, nama vendor, dan aksi. Terdapat juga tombol *preorder* produk (*preorder product*) pada masing-masing vendor. Selain itu terdapat juga tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, CSV, dan PDF, serta kolom pencarian vendor berdasarkan kata kunci dari seluruh teks yang ada pada tabel. Desain halaman *preorder* toko dapat dilihat pada Gambar 3.74.



Gambar 3.74. Desain Halaman *Preorder* di Toko

### c) Halaman daftar pilihan produk

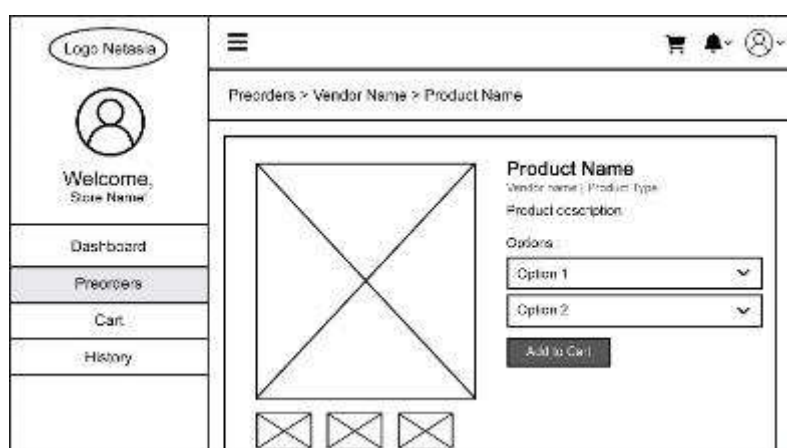
Halaman daftar pilihan produk berisi produk-produk yang akan dipesan oleh toko. Informasi yang ditampilkan pada halaman ini antara lain foto produk, nama produk, dan deskripsi singkat produk. Terdapat juga tombol untuk memesan produk (*preorders*) pada masing-masing produk. Desain halaman daftar pilihan produk dapat dilihat pada Gambar 3.75.



Gambar 3.75. Desain Halaman Daftar Pilihan Produk di Toko

#### d) Halaman detail produk

Halaman detail produk berisi informasi produk antara lain foto produk, nama produk, deskripsi produk, tipe produk, dan opsi produk. Terdapat juga tombol untuk memasukkan produk dalam keranjang (*add to cart*). Desain halaman detail produk dapat dilihat pada Gambar 3.76.

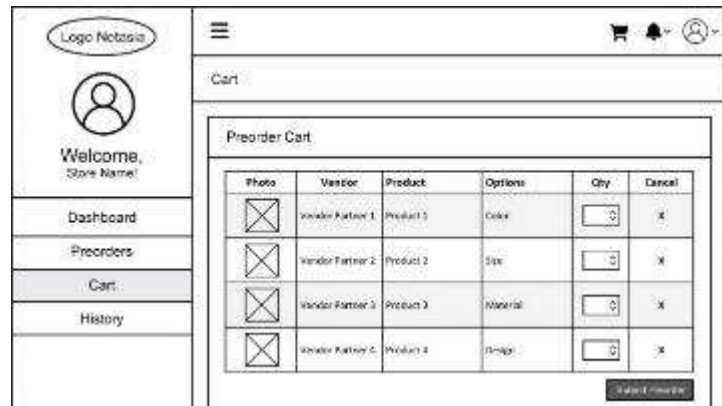


Gambar 3.76. Desain Halaman Detail Produk di Toko

#### e) Halaman keranjang (*cart*)

Halaman keranjang berisi daftar produk yang akan dipesan oleh toko yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama vendor, nama produk, opsi produk, jumlah produk yang akan dipesan,

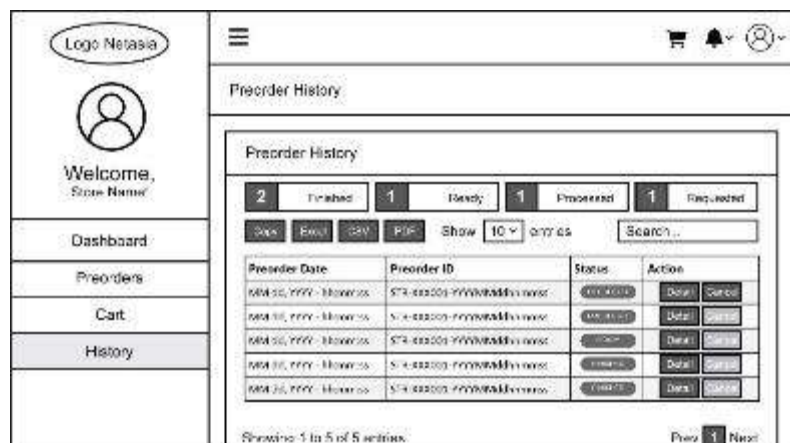
dan tombol pembatalan. Terdapat juga tombol untuk mengirim *preorder* ke admin (*submit preorder*). Desain halaman keranjang dapat dilihat pada Gambar 3.77.



Gambar 3.77. Desain Halaman Keranjang di Toko

f) Halaman riwayat *preorder* toko (*history*)

Halaman riwayat *preorder* toko berisi daftar riwayat *preorder* toko yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain tanggal *preorder*, id *preorder*, status, dan aksi. Terdapat juga tombol detail *preorder* dan batalkan *preorder* pada setiap *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, CSV, dan PDF, serta kolom pencarian *preorder*. Desain halaman riwayat *preorder* toko dapat dilihat pada Gambar 3.78.

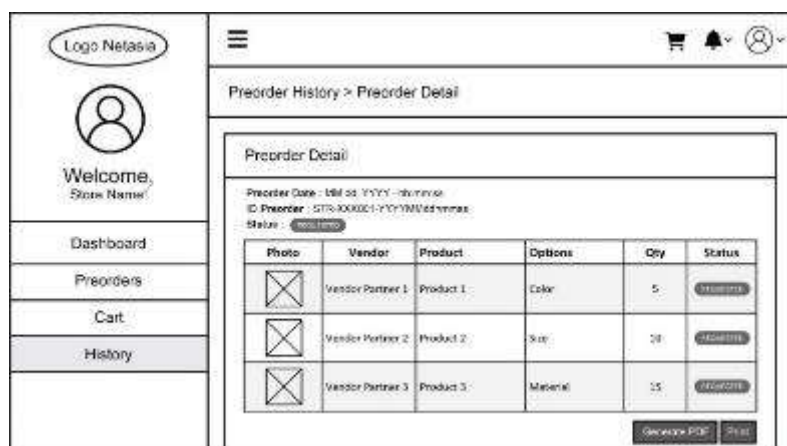


Gambar 3.78. Desain Halaman Riwayat *Preorder* di Toko



g) Halaman detail *preorder* toko

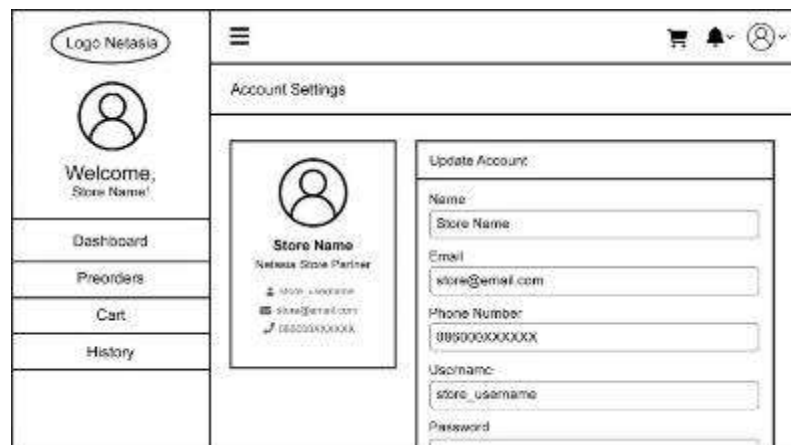
Halaman detail *preorder* toko berisi informasi *preorder* yang terdiri dari tanggal *preorder*, id *preorder*, dan status, serta daftar produk pesanan toko yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama vendor, nama produk, opsi produk, jumlah produk yang dipesan, dan status produk. Terdapat juga tombol untuk mencetak halaman (*print*) dan tombol ekspor halaman ke PDF. Tampilan halaman detail *preorder* toko dapat dilihat pada Gambar 3.79.



Gambar 3.79. Desain Halaman Detail *Preorder* di Toko

h) Halaman pengaturan akun toko (*settings*)

Halaman pengaturan akun toko berisi informasi akun yang terdiri dari nama toko, *username*, email, dan nomor telepon, formulir untuk memperbarui akun (*update account*) yang terdiri dari kolom untuk memasukkan nama, alamat, email, nomor telepon, *username*, dan password, serta tombol untuk memperbarui akun (*update*). Tampilan halaman pengaturan akun toko dapat dilihat pada Gambar 3.80.



Gambar 3.80. Desain Halaman Pengaturan Akun Toko

#### 4. Konstruksi prototipe (*construction of prototype*)

Konstruksi prototipe meliputi pengembangan perangkat lunak berdasarkan hasil analisis kebutuhan dan perancangan sistem yang telah dilakukan pada tahap sebelumnya. Tahap ini menjadi tahap yang menjadi fokus utama dalam metode *prototyping* karena pada tahap ini perangkat lunak sudah mulai dikembangkan. Prototipe perangkat lunak yang dikembangkan selanjutnya diujikan, salah satunya untuk menerima masukan dan saran dari pengguna dalam hal ini pihak perusahaan, sehingga dapat dijadikan sebagai evaluasi dan perbaikan perangkat lunak agar benar-benar bisa digunakan.

#### 5. Penyebaran, pengiriman, masukan (*deployment, delivery, feedback*)

Jika prototipe perangkat lunak yang dikembangkan telah selesai, selanjutnya perangkat lunak diserahkan dan digunakan oleh pengguna untuk mendapatkan masukan (*feedback*). Tahap ini juga meliputi tahap pengujian perangkat lunak yang terdiri dari pengujian *black-box*, pengujian *white-box*, *User Acceptance Testing* (UAT), dan pengujian kompatibilitas aplikasi *browser*. Instrumen yang digunakan dalam pengujian tersebut adalah angket atau kuesioner berupa formulir. Pengujian perangkat lunak berupa aplikasi berbasis web pada penelitian ini diuraikan sebagai berikut:

a. Pengujian *black-box*

Pengujian *black-box* dilakukan untuk menguji spesifikasi fungsional dari sebuah perangkat lunak. Tujuan dari pengujian ini adalah untuk mengetahui bagaimana fungsi-fungsi pada perangkat lunak dapat bekerja sesuai dengan yang dirancang. Pengujian ini dilakukan bersama responden yang paham perangkat lunak dengan jumlah ganjil menggunakan instrumen kuesioner berupa formulir. Teknik pengujian *black-box* yang digunakan pada penelitian ini adalah *sample testing*, yaitu memastikan nilai yang terpilih dapat menghasilkan data yang sesuai dengan data masukan dari *user*.

Terdapat total 76 kasus pengujian (*test case*) yang dilakukan pada pengujian ini dengan rincian 4 kasus pengujian pada halaman *login* yang ditampilkan pada Tabel 3.28, 41 kasus pengujian pada halaman admin yang ditampilkan pada Tabel 3.29, 15 kasus pengujian pada halaman vendor yang ditampilkan pada Tabel 3.30, dan 16 kasus pengujian pada halaman toko yang ditampilkan pada Tabel 3.31. Hasil pengujian untuk setiap *test case* lalu dituliskan pada kolom “Keterangan” berdasarkan dua kondisi yakni “Berhasil” atau “Gagal”. Formulir pengujian *black-box* untuk masing-masing halaman pada aplikasi berbasis web ini diuraikan sebagai berikut:

1) Formulir pengujian halaman *login*

Tabel 3.28. Formulir Pengujian *Black-box* Halaman *Login*

<i>Test case</i>	Hasil yang diharapkan	Hasil yang didapatkan	Ket.
<b>1. Formulir <i>login</i></b>			
<i>Username</i> dan <i>password</i> diisi dengan data yang benar dengan level admin	Pengguna dapat masuk ke halaman utama admin (dasbor)	Sistem akan menampilkan halaman utama admin (dasbor)	

<i>Test case</i>	<b>Hasil yang diharapkan</b>	<b>Hasil yang didapatkan</b>	<b>Ket.</b>
<i>Username dan password</i> diisi dengan data yang benar dengan level vendor	Pengguna dapat masuk ke halaman utama vendor (dasbor)	Sistem akan menampilkan halaman utama vendor (dasbor)	
<i>Username dan password</i> diisi dengan data yang benar dengan level toko	Pengguna dapat masuk ke halaman utama toko (dasbor)	Sistem akan menampilkan halaman utama toko (dasbor)	
<i>Username dan password</i> diisi dengan data yang salah	Pengguna tidak dapat masuk ke halaman utama	Sistem kembali ke halaman <i>login</i> dan menampilkan pesan <i>error</i>	

## 2) Formulir pengujian halaman admin

Tabel 3.29. Formulir Pengujian *Black-box* Halaman Admin

<i>Test case</i>	<b>Hasil yang diharapkan</b>	<b>Hasil yang didapatkan</b>	<b>Ket.</b>
<b>1. Menu dasbor (<i>dashboard</i>)</b>			
Memilih menu dasbor	Masuk ke menu dasbor admin	Menampilkan halaman dasbor admin	
<b>2. Menu daftar vendor (<i>vendor partner</i>)</b>			
Memilih menu <i>vendor partner</i>	Masuk ke menu <i>vendor partner</i>	Menampilkan halaman <i>vendor partner</i>	

<i>Test case</i>	<b>Hasil yang diharapkan</b>	<b>Hasil yang didapatkan</b>	<b>Ket.</b>
Klik tombol detail	Masuk ke menu detail vendor untuk melihat daftar produk vendor	Menampilkan halaman detail vendor berisi daftar produk vendor	
Klik tombol <i>add account</i>	Muncul <i>modal form</i> untuk menambahkan akun vendor baru	Menampilkan <i>modal form</i> untuk menambahkan akun vendor baru	
Mengisi form penambahan akun vendor baru dan klik tombol <i>register</i>	Data akun vendor baru berhasil tersimpan	Menampilkan daftar vendor di halaman <i>vendor partner</i> dan tombol <i>add account</i> berubah menjadi tombol <i>edit account</i>	
Klik tombol <i>edit account</i>	Muncul <i>modal form</i> untuk mengedit akun vendor	Menampilkan <i>modal form</i> untuk mengedit akun vendor	
Mengisi form edit akun vendor dan klik tombol <i>update</i>	Data akun vendor berhasil diperbarui	Menampilkan daftar vendor di halaman <i>vendor partner</i>	
<b>3. Menu daftar toko (<i>store partner</i>)</b>			
Memilih menu <i>store partner</i>	Masuk ke menu <i>store partner</i>	Menampilkan halaman <i>store partner</i>	
Klik tombol <i>add</i>	Muncul <i>modal</i>	Menampilkan	

<i>Test case</i>	<b>Hasil yang diharapkan</b>	<b>Hasil yang didapatkan</b>	<b>Ket.</b>
<i>new store</i>	<i>form</i> untuk menambahkan akun toko baru	<i>modal form</i> untuk menam-bahkan akun toko baru	
Mengisi form penambahan akun toko baru dan klik tombol <i>register</i>	Data akun toko baru berhasil tersimpan	Menampilkan daftar toko di halaman <i>store partner</i>	
Klik tombol detail	Masuk ke menu detail toko untuk melihat daftar <i>preorder</i> toko	Menampilkan halaman detail toko berisi daftar <i>preorder</i> toko	
Klik tombol edit	Muncul <i>modal form</i> un-tuk mengedit akun toko	Menampilkan <i>modal form</i> untuk mengedit akun toko	
Mengisi form edit akun toko dan klik tombol <i>update</i>	Data akun toko berhasil diperbarui	Menampilkan daftar toko di halaman <i>store partner</i>	
Klik tombol hapus	Data akun toko berhasil dihapus	Menampilkan daftar toko di halaman <i>store partner</i>	
<b>4. Menu prapemesanan (<i>preorders</i>)</b>			
Memilih menu <i>preorders</i>	Masuk ke menu <i>preorders</i>	Menampilkan halaman <i>preorders</i>	

<i>Test case</i>	<b>Hasil yang diharapkan</b>	<b>Hasil yang didapatkan</b>	<b>Ket.</b>
Klik tab <i>preorder from store</i>	Melihat daftar permintaan <i>preorder</i> dari toko	Menampilkan daftar permintaan <i>preorder</i> dari toko	
Klik tab <i>preorder to vendor</i>	Melihat daftar permintaan <i>preorder</i> ke vendor	Menampilkan daftar permintaan <i>preorder</i> ke vendor	
Klik tombol <i>process</i> di tab <i>preorder from store</i>	Memproses permintaan <i>preorder</i> dari toko	Menampilkan halaman <i>process preorder</i>	
Klik tombol <i>add to demand</i> di halaman <i>process preorder</i>	Memasukkan pesanan toko ke daftar permintaan <i>preorder</i> ke vendor	Menampilkan halaman <i>preorders</i>	
Klik tombol <i>process to vendor</i> di tab <i>preorder to vendor</i>	Memproses permintaan <i>preorder</i> ke vendor	Menampilkan daftar produk yang dipesan di halaman <i>process preorder to vendor</i>	
Klik tombol <i>submit preorder to vendor</i>	<i>Checkout</i> produk dan mengirimkan <i>preorder</i> kepada vendor	Menampilkan tab <i>preorder to vendor</i> di halaman <i>preorders</i>	
<b>5. Menu riwayat prapemesanan (<i>history</i>)</b>			
Memilih menu <i>history</i>	Masuk ke menu <i>history</i>	Menampilkan halaman <i>history</i>	

<i>Test case</i>	<b>Hasil yang diharapkan</b>	<b>Hasil yang didapatkan</b>	<b>Ket.</b>
Klik tab <i>preorder from store</i>	Melihat daftar riwayat <i>preorder</i> dari toko	Menampilkan daftar riwayat <i>preorder</i> dari toko	
Klik tab <i>preorder to vendor</i>	Melihat daftar riwayat <i>preorder</i> ke vendor	Menampilkan daftar riwayat <i>preorder</i> ke vendor	
Klik tombol detail di tab <i>preorder from store</i>	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan toko	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan toko	
Klik tombol detail di tab <i>preorder to vendor</i>	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan ke vendor	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	
Jika status di tab <i>preorder to vendor</i> adalah “ <i>delivered</i> ” klik tombol <i>check</i>	Memeriksa apakah produk yang dikirimkan sesuai dengan yang diterima	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	
Klik tombol <i>request product resend</i> di halaman detail <i>preorder</i>	Meminta pengiriman ulang produk karena ada yang tidak sesuai	Muncul <i>modal form</i> keterangan permintaan pengiriman ulang	



<i>Test case</i>	<b>Hasil yang diharapkan</b>	<b>Hasil yang didapatkan</b>	<b>Ket.</b>
Mengisi form keterangan permintaan pengiriman ulang dan klik <i>request</i>	Permintaan pengiriman ulang beserta keterangan berhasil dikirim ke vendor	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>request product resend</i>	
Klik tombol <i>receive product</i> di halaman detail <i>preorder</i>	Menerima produk yang telah terkirim dari vendor ke admin	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>received</i>	
Jika status di tab <i>preorder from store "ready"</i> klik tombol detail	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan ke vendor	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	
Klik tombol <i>finish preorder</i> di halaman detail <i>preorder</i>	Produk telah didistribusikan kepada toko	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>finished</i>	
<b>6. Menu akun admin (<i>admin accounts</i>)</b>			
Memilih menu <i>admin accounts</i>	Masuk ke menu <i>admin accounts</i>	Menampilkan halaman <i>admin accounts</i>	
Klik tombol <i>add new admin</i>	Muncul <i>modal form</i> untuk menambahkan akun admin baru	Menampilkan <i>modal form</i> untuk menambahkan akun admin baru	
Mengisi form	Data akun admin	Menampilkan	

<i>Test case</i>	<b>Hasil yang diharapkan</b>	<b>Hasil yang didapatkan</b>	<b>Ket.</b>
penambahan akun admin baru dan klik tombol <i>register</i>	baru berhasil tersimpan	daftar admin di halaman <i>admin account</i>	
Klik tombol edit	Muncul <i>modal form</i> untuk mengedit akun admin	Menampilkan <i>modal form</i> untuk mengedit akun admin	
Mengisi form edit akun admin dan klik tombol <i>update</i>	Data akun admin berhasil diperbarui	Menampilkan daftar admin di halaman <i>store partner</i>	
Klik tombol hapus	Data akun toko berhasil dihapus	Menampilkan daftar admin di halaman <i>admin account</i>	
<b>7. Menu pengaturan akun</b>			
Klik tombol <i>settings</i>	Masuk ke menu <i>settings</i>	Menampilkan <i>settings</i>	
Mengisi form pengaturan akun dan klik tombol <i>update</i>	Data akun berhasil diperbarui	Menampilkan halaman <i>login</i>	
<b>8. Keluar dari halaman admin</b>			
Klik tombol <i>logout</i>	Kembali ke halaman <i>login</i>	Menampilkan halaman <i>login</i>	

## 3) Formulir pengujian halaman vendor

Tabel 3.30. Formulir Pengujian *Black-box* Halaman Vendor

<i>Test case</i>	<b>Hasil yang diharapkan</b>	<b>Hasil yang didapatkan</b>	<b>Ket.</b>
<b>1. Menu dasbor (<i>dashboard</i>)</b>			
Memilih menu dasbor	Masuk ke menu dasbor vendor	Menampilkan halaman dasbor vendor	
<b>2. Menu daftar produk (<i>products</i>)</b>			
Memilih menu <i>products</i>	Masuk ke menu <i>products</i>	Menampilkan halaman <i>products</i> dan daftar produk	
<b>3. Menu prapemesanan (<i>preorders</i>)</b>			
Memilih menu <i>preorders</i>	Masuk ke menu <i>preorders</i>	Menampilkan halaman <i>preorders</i>	
Klik tombol <i>process</i> di salah satu pesanan	Memproses permintaan <i>preorder</i> dari admin	Menampilkan halaman <i>process preorder</i>	
Klik tombol <i>accept preorder</i> di halaman <i>process preorder</i>	Menerima permintaan <i>preorder</i> dari admin	Menampilkan halaman <i>preorders</i>	
<b>4. Menu riwayat prapemesanan (<i>history</i>)</b>			
Memilih menu <i>history</i>	Masuk ke menu <i>history</i>	Menampilkan halaman <i>history</i>	
Klik tombol detail	Masuk ke menu detail <i>preorder</i> untuk melihat	Menampilkan halaman detail <i>preorder</i> berisi	

<i>Test case</i>	<b>Hasil yang diharapkan</b>	<b>Hasil yang didapatkan</b>	<b>Ket.</b>
	daftar produk yang dipesan	daftar produk yang dipesan	
Jika status “ <i>accepted</i> ” klik tombol <i>deliver</i>	Masuk ke menu detail <i>preorder</i> untuk memperbarui status	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	
Klik tombol <i>deliver product</i> di halaman detail <i>preorder</i>	Memperbarui status <i>preorder</i> bahwa produk sedang dikirim	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>delivered</i>	
Jika status “ <i>request product resend</i> ” klik tombol <i>respond</i>	Masuk ke menu detail <i>preorder</i> untuk merespon permintaan	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor dan keterangan	
Klik tombol <i>contact admin</i> di halaman detail <i>preorder</i>	Menghubungi kontak admin	Menuju ke nomor telepon admin	
Klik tombol <i>resend product</i> di halaman detail <i>preorder</i>	Memperbarui status <i>preorder</i> bahwa produk sedang dikirim ulang	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>delivered</i>	

<i>Test case</i>	<b>Hasil yang diharapkan</b>	<b>Hasil yang didapatkan</b>	<b>Ket.</b>
<b>5. Menu pengaturan akun</b>			
Klik tombol <i>settings</i>	Masuk ke menu <i>settings</i>	Menampilkan menu <i>settings</i>	
Mengisi form pengaturan akun dan klik tombol <i>update</i>	Data akun berhasil diperbarui	Menampilkan halaman <i>login</i>	
<b>6. Keluar dari halaman vendor</b>			
Klik tombol <i>logout</i>	Kembali ke halaman <i>login</i>	Menampilkan halaman <i>login</i>	

## 4) Formulir pengujian halaman toko

Tabel 3.31. Formulir Pengujian *Black-box* Halaman Toko

<i>Test case</i>	<b>Hasil yang diharapkan</b>	<b>Hasil yang didapatkan</b>	<b>Ket.</b>
<b>1. Menu dasbor (<i>dashboard</i>)</b>			
Memilih menu dasbor	Masuk ke menu dasbor toko	Menampilkan halaman dasbor toko	
<b>2. Menu prapemesanan (<i>preorders</i>)</b>			
Memilih menu <i>preorders</i>	Masuk ke menu <i>preorders</i>	Menampilkan daftar vendor di halaman <i>preorders</i>	
Klik tombol <i>preorder product</i> di salah satu vendor	Masuk ke menu detail vendor untuk melihat daftar produk vendor	Menampilkan halaman detail vendor berisi daftar produk vendor	

<i>Test case</i>	<b>Hasil yang diharapkan</b>	<b>Hasil yang didapatkan</b>	<b>Ket.</b>
Klik tombol <i>preorder</i> pada salah satu produk	Masuk ke menu detail produk untuk memesan produk	Menampilkan halaman detail produk	
Memilih opsi produk dan klik tombol <i>add to cart</i>	Memilih produk sesuai opsi dan memasukkan produk ke dalam keranjang	Menampilkan halaman detail produk	
<b>3. Menu keranjang (<i>cart</i>)</b>			
Memilih menu <i>cart</i>	Masuk ke menu <i>cart</i>	Menampilkan halaman <i>cart</i>	
Klik tombol <i>submit preorder</i>	<i>Checkout</i> produk dan mengirimkan <i>preorder</i> ke admin	Menampilkan halaman <i>history</i>	
<b>4. Menu riwayat prapemesanan (<i>history</i>)</b>			
Memilih menu <i>history</i>	Masuk ke menu <i>history</i>	Menampilkan halaman <i>history</i>	
Klik tombol detail	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan	Menampilkan halaman detail vendor berisi daftar produk yang dipesan	
Klik tombol <i>cancel preorder</i>	<i>Preorder</i> dibatalkan	Menampilkan halaman <i>history</i> dan status menjadi “ <i>canceled</i> ”	
Jika status “ <i>canceled</i> ” klik	Mengubah data dalam <i>preorder</i>	Menampilkan halaman <i>cart</i>	

<i>Test case</i>	<b>Hasil yang diharapkan</b>	<b>Hasil yang didapatkan</b>	<b>Ket.</b>
tombol <i>edit preorder</i>		berisi data <i>preorder</i> yang diedit	
Memperbarui data <i>preorder</i> dan klik tombol <i>submit prorder</i> di halaman <i>cart</i>	Data <i>preorder</i> berhasil diperbarui	Menampilkan halaman <i>history</i>	
Jika status “ <i>canceled</i> ” klik tombol <i>delete preorder</i>	Data <i>preorder</i> berhasil dihapus	Menampilkan halaman <i>history</i>	
<b>5. Menu pengaturan akun</b>			
Klik tombol <i>settings</i>	Masuk ke menu <i>settings</i>	Menampilkan menu <i>settings</i>	
Mengisi form pengaturan akun dan klik tombol <i>update</i>	Data akun berhasil diperbarui	Menampilkan halaman <i>login</i>	
<b>6. Keluar dari halaman toko</b>			
Klik tombol <i>logout</i>	Kembali ke halaman <i>login</i>	Menampilkan halaman <i>login</i>	

Setelah formulir kuesioner pengujian *black-box* untuk masing-masing halaman telah diisi oleh seluruh responden, maka dilakukan perhitungan untuk masing-masing responden dengan rumus sebagai berikut:

$$\text{Tercapai} = \frac{\text{Jumlah tercapai}}{\text{Jumlah semua aspek}} \times 100\% \quad (3.1)$$

$$\text{Gagal} = \frac{\text{Jumlah gagal}}{\text{Jumlah semua aspek}} \times 100\% \quad (3.2)$$

Kemudian dihitung rata-rata persentase hasil pengujian dari seluruh responden untuk mendapatkan hasil pengujian *black-box* dengan rumus sebagai berikut:

$$\text{Hasil Pengujian} = \frac{\text{Jumlah persentase}}{\text{Jumlah responden}} \quad (3.3)$$

b. Pengujian *white-box*

Pengujian *white-box* dilakukan untuk menguji desain dan struktur program secara prosedural. Tujuan dari pengujian ini adalah untuk mengetahui bagaimana alur data, struktur kode, dan kemungkinan adanya kegagalan pada sistem atau perangkat lunak. Pengujian ini dilakukan oleh peneliti sendiri yang paham bahasa pemrograman dan mampu memeriksa struktur kode dari sistem atau perangkat lunak yang diuji. Teknik pengujian *white-box* yang digunakan pada penelitian ini antara lain diuraikan sebagai berikut:

- 1) Pengujian berbasis alur (*basis path testing*) yang merupakan penggambaran diagram alur (*flow diagram*) dari struktur kode (*script*) yang nantinya digunakan untuk menghitung kompleksitas siklomatis (*cyclomatic complexity*).
- 2) Kompleksitas siklomatis (*cyclomatic complexity*) yang merupakan pengukuran kuantitatif terhadap kompleksitas logika sebuah program. Hasil kompleksitas siklomatis menggambarkan banyaknya *path* dan batas atas sejumlah ujicoba yang harus dirancang dan dieksekusi untuk seluruh perintah dalam program. Kompleksitas siklomatis dihitung menggunakan rumus berikut:



$$V(G) = E - N + 2 \quad (3.4)$$

Keterangan:

$V(G)$  : Kompleksitas siklomatis untuk alur graf (G)

$E$  : Panah (*edge*)

$N$  : Lingkaran (*node*)

- 3) Jalur independen (*independent path*) untuk memperkenalkan sedikitnya satu rangkaian proses baru atau suatu kondisi baru.
- 4) Pengujian nilai (*value test*) yang merupakan keterangan proses masukan dan keluaran (*input and output*) dari sebuah jalur independen yang telah ditentukan. Pengujian nilai dihitung menggunakan rumus berikut:

$$\text{Tercapai} = \frac{\text{Jumlah path tercapai}}{\text{Jumlah semua path}} \times 100\% \quad (3.5)$$

$$\text{Gagal} = \frac{\text{Jumlah path gagal}}{\text{Jumlah semua path}} \times 100\% \quad (3.6)$$

c. *User Acceptance Testing* (UAT)

*User Acceptance Testing* (UAT) dilakukan untuk menguji apakah perangkat lunak sudah memenuhi harapan pengguna dan bekerja seperti yang telah direncanakan. Pengujian ini dilakukan bersama responden yang tidak perlu paham perangkat lunak dengan jumlah ganjil menggunakan instrumen kuesioner berupa formulir. Aspek yang diujikan dibagi menjadi tiga bagian yakni aspek kemanfaatan, aspek kemudahan penggunaan, dan aspek antarmuka pengguna (*user interface*). Terdapat 13 pertanyaan yang diajukan pada pengujian ini. Pertanyaan yang diajukan pada formulir UAT dapat dilihat pada Tabel 3.32, sedangkan penjelasan skor dari masing-masing pertanyaan dan persentase kelayakan aplikasi dapat dilihat pada Tabel 3.33.

Tabel 3.32. Formulir *User Acceptance Testing* (UAT)

No	Pertanyaan	Skor				
		1	2	3	4	5
Aspek Kemanfaatan						
1	Apakah aplikasi berbasis web ini bermanfaat untuk membantu pengelolaan distribusi produk UKM di Netasia secara keseluruhan?					
2	Apakah aplikasi berbasis web ini bermanfaat untuk admin dalam mengelola distribusi produk UKM di Netasia?					
3	Apakah aplikasi berbasis web ini bermanfaat untuk vendor dalam menerima <i>preorder</i> dari Netasia?					
4	Apakah aplikasi berbasis web ini bermanfaat untuk toko dalam melakukan <i>preorder</i> produk UKM di Netasia?					
5	Apakah aplikasi berbasis web ini membuat pengelolaan distribusi produk UKM di Netasia menjadi lebih efektif dan efisien?					

No	Pertanyaan	Skor				
		1	2	3	4	5
Aspek Kemudahan Penggunaan						
6	Apakah aplikasi berbasis web ini mudah dipahami?					
7	Apakah aplikasi berbasis web ini mudah digunakan?					
8	Apakah aplikasi berbasis web ini dapat berjalan sesuai yang diharapkan?					
9	Apakah aplikasi berbasis web ini mudah dimengerti oleh admin?					
10	Apakah aplikasi berbasis web ini mudah dimengerti oleh vendor?					
11	Apakah aplikasi berbasis web ini mudah dimengerti oleh toko?					
Aspek Antarmuka Pengguna ( <i>User Interface</i> )						
12	Apakah aplikasi berbasis web ini memiliki tampilan yang jelas?					
13	Apakah aplikasi berbasis web ini memiliki tampilan yang baik?					

Tabel 3.33. Bobot Penilaian *User Acceptance Testing* (UAT)

Jawaban	Skor	Persentase
Sangat Setuju	5	81% - 100%
Setuju	4	61% - 80%
Cukup Setuju	3	41% - 60%
Kurang Setuju	2	21% - 40%
Tidak Setuju	1	0% - 20%

Setelah formulir kuesioner UAT telah diisi oleh seluruh responden pada masing-masing level, maka akan didapatkan total persentase dari masing-masing responden. Kemudian dilakukan perhitungan dengan rumus sebagai berikut:

$$\text{Hasil UAT} = \frac{\text{Total persentase}}{\text{Jumlah responden}} \quad (3.7)$$

d. Pengujian kompatibilitas aplikasi *browser*

Pengujian kompatibilitas aplikasi *browser* dilakukan untuk mengetahui tingkat kompatibilitas perangkat lunak pada aplikasi *browser*. Pengujian ini dilakukan dengan menjalankan aplikasi berbasis web yang dikembangkan pada penelitian ini pada beberapa tipe aplikasi *browser* yang telah ditentukan antara lain Google Chrome, Internet Explorer, Mozilla Firefox, Opera Browser, dan Safari Browser. Aspek yang diujikan adalah fungsionalitas dan tampilan antarmuka pengguna (*user interface*) aplikasi. Pengujian ini dilakukan oleh peneliti sendiri dengan menggunakan instrumen kuesioner berupa formulir. Hasil pengujian kemudian dituliskan pada setiap kolom aplikasi *browser* berdasarkan dua kondisi yakni “Normal” atau “Berubah”. Formulir pengujian kompatibilitas aplikasi *browser* pada penelitian ini dapat dilihat pada Tabel 3.34.

Tabel 3.34. Formulir Pengujian Kompatibilitas Aplikasi *Browser*

No	Aspek	Google Chrome	Internet Explorer	Mozilla Firefox	Opera Browser	Safari Browser
1	Fungsi menu					
2	Tampilan halaman					

### E. Proses Eksperimen

1. Tahap persiapan, meliputi:
  - a. Perancangan penelitian.
  - b. Studi pustaka.
  - c. Perancangan aplikasi.
  - d. Pembuatan aplikasi.
  - e. Validasi aplikasi.
2. Tahap pelaksanaan, meliputi:
  - a. Pengelompokkan sampel pada satu kelas penelitian.
  - b. Pengujian *black-box*, *white-box*, dan UAT untuk mengetahui kelayakan aplikasi.
  - c. Penggunaan aplikasi oleh pengguna.
3. Tahap akhir, meliputi:
  - a. Pengolahan dan analisis data hasil penelitian.
  - b. Menyimpulkan hasil penelitian dengan alur yang sistematis.

### F. Teknik Pengumpulan Data

Pengumpulan data dilakukan untuk merancang, melaksanakan, dan mendapatkan hasil dari penelitian yang dilakukan. Oleh karena itu diperlukan suatu teknik untuk mengumpulkan data yang dibutuhkan dalam penelitian. Data dikumpulkan berdasarkan jenis data dan instrumen penelitian yang digunakan. Teknik pengumpulan data berdasarkan jenis data dan instrumen yang digunakan pada penelitian ini antara lain diuraikan sebagai berikut:

### 1. Jenis data

Data yang didapatkan dalam penelitian ini terdiri dari dua data, yaitu data kualitatif dan kuantitatif dengan penjelasan sebagai berikut:

- a. Data kualitatif merupakan data mengenai proses pengembangan dan hasil pengujian aplikasi berbasis web yang dikembangkan. Data ini didapatkan dari hasil wawancara dengan pihak perusahaan.
- b. Data kuantitatif merupakan data pokok penelitian yang berupa data kebutuhan dan data hasil pengujian aplikasi yang dikembangkan pada penelitian. Data ini didapatkan dari hasil studi pustaka, observasi, dan pengujian yang dilakukan.

### 2. Instrumen penelitian

Instrumen penelitian yang digunakan adalah angket atau kuesioner. Angket atau kuesioner merupakan instrumen penelitian dalam pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawab. Angket digunakan untuk mengukur kualitas aplikasi yang dibuat. Instrumen angket pada penelitian ini digunakan untuk memperoleh data dari pihak perusahaan serta hasil pengujian aplikasi.

## **G. Teknik Analisis Data**

Setelah semua data hasil dari penelitian telah terkumpul, maka langkah selanjutnya adalah menganalisis data tersebut. Teknik analisis data yang digunakan pada penelitian ini adalah sebagai berikut:

1. Mengumpulkan data yang diperlukan dengan observasi, wawancara, dan studi kepustakaan. Data yang dikumpulkan antara lain sebagai berikut:
  - a. Data kebutuhan fungsional sistem dan nonfungsional sistem dari perangkat lunak yang dikembangkan.
  - b. Data hasil pengembangan perangkat lunak.
  - c. Data hasil pengujian perangkat lunak.

2. Menganalisis data dengan cara mengidentifikasi kebutuhan yang diperlukan dalam pembuatan perangkat lunak serta menganalisis hasil pengembangan dan pengujian perangkat lunak.
3. Mengolah data-data yang sudah terkumpul untuk kemudian diproses dengan metode yang sudah ditentukan hingga mendapatkan hasil penelitian sesuai yang direncanakan.
4. Membuat kesimpulan akhir berdasarkan hasil pengembangan dan pengujian perangkat lunak yang dikembangkan.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **A. Hasil Pengembangan**

Hasil pengembangan aplikasi didapatkan setelah melalui beberapa tahap pada metode *prototyping* yang berlangsung selama sekitar delapan bulan. Aplikasi yang dikembangkan merupakan aplikasi berbasis web yang menggunakan *framework* Codeigniter. Selain itu aplikasi berbasis web ini juga terintegrasi dengan data dari API Shopify milik perusahaan Golden Style Management Lte. Ptd. Penggunaan metode *prototyping* dalam pengembangan aplikasi berbasis web pada penelitian ini sering melibatkan komunikasi antara pengembang dan pengguna aplikasi dalam hal ini pihak perusahaan. Hal ini juga membuat aplikasi yang dikembangkan banyak mengalami perubahan dan pembaruan berdasarkan masukan dari pengguna. Hasil implementasi pengembangan aplikasi berbasis web dengan metode *prototyping* pada penelitian ini diuraikan sebagai berikut:

##### **1. Komunikasi dan perencanaan**

Tahap awal dari pengembangan perangkat lunak dengan menggunakan metode *prototyping* adalah komunikasi (*communication*). Komunikasi yang dimaksud adalah pengumpulan data dan informasi yang dilakukan dengan tiga cara yakni studi pustaka dari berbagai literatur, wawancara dengan pihak perusahaan, dan observasi secara langsung terhadap data yang ada di CMS Shopify milik perusahaan. Pengembangan perangkat lunak dengan menggunakan metode *prototyping* akan terus melibatkan perusahaan karena pada metode ini aplikasi dikatakan sebagai sebuah prototipe yang dapat selalu berbuah sesuai dengan kebutuhan perusahaan.

Hasil yang didapatkan dari tahap komunikasi ini digunakan untuk melakukan tahap berikutnya yaitu tahap perencanaan cepat (*quick plan*) yang terdiri dari analisis masalah dan analisis kebutuhan yang meliputi kebutuhan data, kebutuhan fungsional sistem, dan kebutuhan



nonfungsional sistem. Setelah mengetahui permasalahan dan kebutuhan dari perusahaan, maka selanjutnya dilakukan perencanaan model cepat (*quick modelling design*) yang meliputi pembuatan model sistem menggunakan diagram-diagram UML, desain antarmuka pengguna (*user interface*), dan desain *database*.

## 2. Prototipe aplikasi

Sebelum diimplementasikan, aplikasi yang dikembangkan pada penelitian ini akan melalui tahap konstruksi prototipe (*construction of prototype*) atau dapat dikatakan sebagai aplikasi sementara. Terdapat beberapa kali perubahan dalam aplikasi yang dikembangkan pada penelitian ini sesuai dengan masukan dari pengguna dalam hal ini pihak perusahaan pada tahap komunikasi (*communication*) dan penyebaran, pengiriman, masukan (*deployment, delivery, feedback*). Pihak dari perusahaan yang terlibat dalam proses pengembangan aplikasi ini adalah direktur perusahaan, staf perusahaan, dan mitra kerja perusahaan dalam bidang *IT Management*. Perubahan pada prototipe aplikasi yang dikembangkan pada penelitian ini antara lain diuraikan sebagai berikut:

### a. Prototipe aplikasi ke-1

Aplikasi berbasis web untuk pengelolaan distribusi produk UKM di Netasia awalnya dikembangkan dengan metode CRUD (*create, read, update, delete*) untuk keperluan pengelolaan data vendor dan produk vendor. Namun pihak perusahaan sebagai pengguna aplikasi meminta perubahan pada metode pengelolaan data tersebut karena telah memiliki data vendor dan produk vendor yang ada di CMS Shopify milik perusahaan. Perusahaan meminta adanya integrasi data antara aplikasi pengelola distribusi produk dengan data vendor dan produk vendor yang ada di CMS Shopify milik perusahaan agar tidak perlu melakukan mengunggahan ulang data vendor dan data produk vendor pada sistem.

b. Prototipe aplikasi ke-2

Setelah diintegrasikan dengan CMS Shopify milik perusahaan, aplikasi berbasis web untuk pengelolaan distribusi produk UKM di Netasia berhasil mendapatkan data vendor dan produk vendor. Salah satu data produk vendor yang didapatkan adalah harga produk. Kemudian harga produk tersebut ditampilkan dalam aplikasi untuk keperluan *preorder* baik dari toko maupun dari admin. Namun pihak perusahaan selaku pengguna aplikasi meminta agar harga produk tersebut tidak ditampilkan dalam aplikasi pengelolaan distribusi produk karena aplikasi ini hanya berfokus pada pengontrolan status pendistribusian.

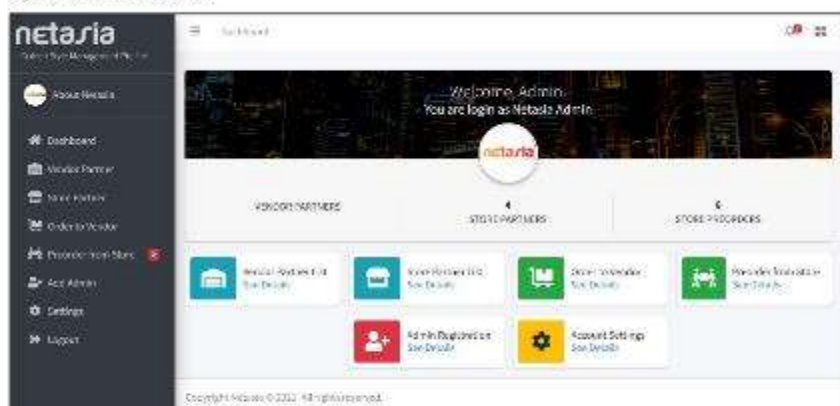
c. Prototipe aplikasi ke-3

Pihak perusahaan selaku pengguna aplikasi meminta adanya fitur catatan pada saat melakukan permintaan atau menerima permintaan *preorder*. Fitur catatan ini hanya tersedia untuk proses permintaan *preorder* dari admin ke vendor dan proses penerimaan *preorder* oleh vendor. Admin dan vendor dapat memberikan catatan baik saat mengirim permintaan, menerima permintaan, maupun menerima produk.

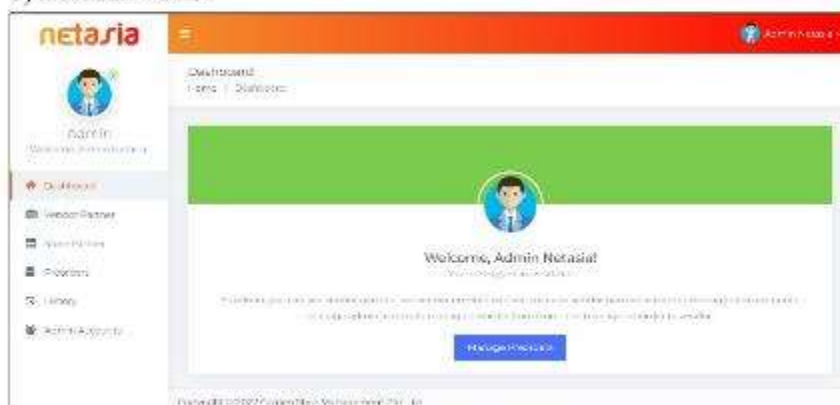
d. Prototipe aplikasi ke-4

Pengembang melakukan perubahan pada tampilan antarmuka pengguna (*user interface*) keseluruhan aplikasi. Perubahan bertujuan untuk membuat tampilan lebih menarik dan sesuai dengan tema warna dari Netasia. Pengembang hanya merubah *template* tanpa merubah fungsionalitas dari aplikasi. Selain itu terdapat juga perubahan pada menu-menu yang ada di setiap halaman untuk masing-masing level. Perubahan menu ini bertujuan untuk meringkas menu agar pengguna bisa lebih mudah dalam menggunakan aplikasi. Perubahan tampilan antarmuka pengguna (*user interface*) di salah satu halaman aplikasi berbasis web yang dikembangkan pada penelitian ini dapat dilihat pada Gambar 4.1.

## a) Sebelum revisi



## b) Sesudah revisi



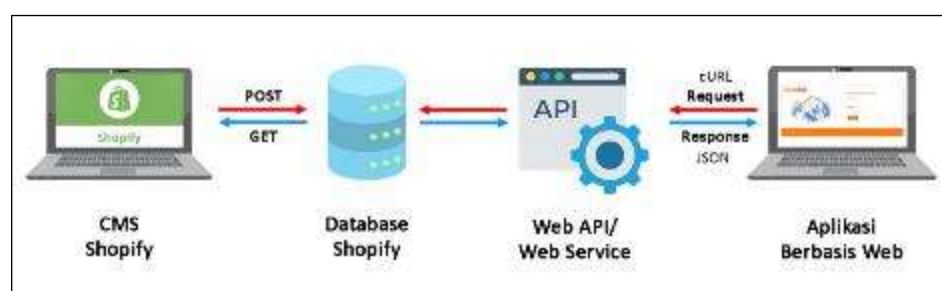
Gambar 4.1. Tampilan (a) Sebelum Revisi dan (b) Sesudah Revisi

### 3. Integrasi data dengan API Shopify

Setelah melalui beberapa kali pengembangan prototipe sampai dengan prototipe terakhir, pengembangan aplikasi dilanjutkan dengan melakukan integrasi data dari CMS Shopify melalui API. Data yang berada di CMS Shopify milik perusahaan antara lain adalah data vendor dan data produk dari vendor. Kedua data inilah yang akan diintegrasikan dengan aplikasi pengelolaan distribusi produk berbasis web yang dikembangkan pada penelitian ini. Integrasi data dilakukan agar pihak perusahaan tidak perlu memasukkan ulang data vendor dan produk vendor ketika menggunakan aplikasi pengelolaan distribusi produk berbasis web yang dikembangkan pada penelitian ini.

Data vendor dan data produk yang ada pada CMS Shopify dapat diintegrasikan dengan aplikasi berbasis web yang dikembangkan pada penelitian ini dengan memanfaatkan *Application Programming Interface* (API) milik Shopify menggunakan *Client URL* (cURL). Integrasi data tersebut dapat dilakukan karena Shopify sebagai CMS menyediakan layanan web (*web service*) yang memungkinkan aplikasi lain dapat mengakses beberapa fitur yang dimilikinya. Data tersebut disajikan dalam format *JavaScript Object Notation* (JSON) untuk selanjutnya dipanggil dalam struktur kode PHP.

Terdapat empat tahapan ketika menggunakan cURL, secara berurutan antara lain 1) menginisialisasi fungsi cURL dengan perintah program *curl\_init()*, 2) mengatur opsi nilai (*set option*) seperti alamat URL yang akan dituju, format hasilnya, header, dan lain-lain dengan perintah program *curl\_setopt()*, 3) melakukan eksekusi (*execute*) dengan perintah program *curl\_exec()* di mana cURL akan melakukan HTTP *request* sesuai dengan opsi yang diberikan pada tahap sebelumnya, dan 4) menutup cURL (*close*) dengan perintah program *curl\_close()* karena proses sudah tidak digunakan lagi. Data yang diambil dari API disajikan dalam format JSON (*JavaScript Object Notation*). Kemudian dilakukan *decode* pada data tersebut dengan perintah kode *json\_decode()*. Fungsi tersebut digunakan untuk menerjemahkan JSON agar mudah saat akan diolah menjadi variabel PHP. Proses integrasi data dari CMS Shopify ke aplikasi berbasis web yang dikembangkan pada penelitian ini dapat dilihat pada Gambar 4.2.



Gambar 4.2. Ilustrasi Proses Integrasi Data Melalui API

#### 4. Hasil implementasi aplikasi

##### a. Halaman *login*

Ketika pengguna mengakses aplikasi melalui web, maka tampilan pertama yang muncul adalah halaman *login*. Halaman ini menampilkan formulir yang terdiri dari kolom untuk memasukkan *username* dan *password* serta tombol untuk *login*. Kemudian pengguna akan masuk ke dalam aplikasi dan diarahkan ke halaman utama sesuai level masing-masing baik sebagai admin, vendor, maupun toko sesuai dengan *username* dan *password* yang telah dimasukkan. Tampilan halaman *login* dapat dilihat pada Gambar 4.3.



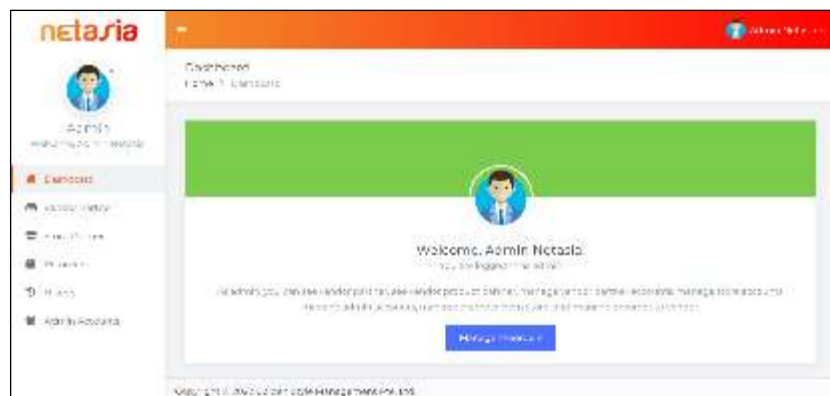
Gambar 4.3. Tampilan Halaman *Login*

##### b. Halaman admin

Halaman admin memiliki enam menu utama antara lain dasbor (*dashboard*), daftar vendor (*vendor partner*), daftar toko (*store partner*), prapemesanan (*preorders*), riwayat prapemesanan (*history*), dan akun admin (*admin accounts*). Selain itu halaman admin juga memiliki beberapa fitur lain seperti detail vendor, detail toko, pemrosesan *preorder*, pengiriman *preorder* ke vendor, detail *preorder*, dan pengaturan akun. Menu dan fitur yang ada di halaman admin diuraikan sebagai berikut:

1) Halaman dasbor admin (*dashboard*)

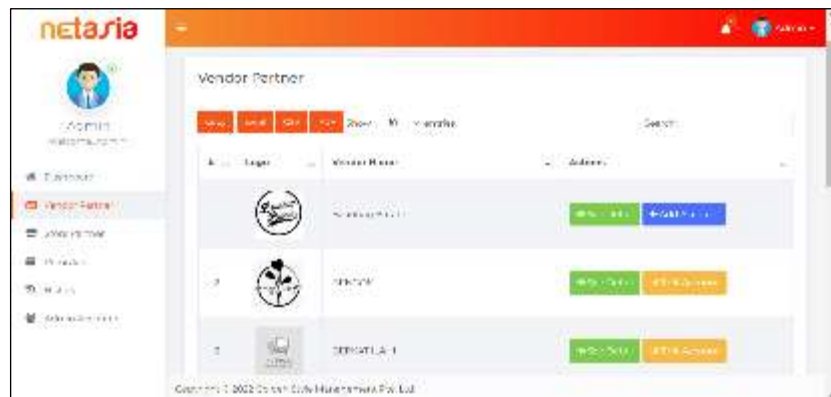
Ketika pengguna *login* ke aplikasi sebagai admin, maka halaman pertama yang muncul setelah berhasil *login* adalah halaman dasbor admin. Halaman dasbor admin berisi pesan selamat datang, deskripsi peran admin, dan tombol pintasan untuk menuju ke fitur utama sebagai admin yaitu halaman *preorder* untuk mengelola *preorder* dari toko dan ke vendor. Tampilan halaman dasbor admin dapat dilihat pada Gambar 4.4.



Gambar 4.4. Tampilan Halaman Dasbor Admin

2) Halaman daftar vendor (*vendor partner*)

Halaman daftar vendor berisi daftar vendor UKM yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain logo vendor, nama vendor, dan aksi. Daftar vendor ini merupakan data yang didapatkan dari API Shopify milik perusahaan. Terdapat juga tombol detail (*see detail*) dan tambah akun (*add account*) pada masing-masing vendor. Namun tombol tambah akun akan berubah menjadi tombol edit akun (*edit account*) jika telah dilakukan registrasi akun vendor. Selain itu terdapat juga tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, *CSV*, dan *PDF*, serta kolom pencarian vendor berdasarkan kata kunci dari seluruh teks yang ada pada tabel. Tampilan halaman daftar vendor dapat dilihat pada Gambar 4.5.



Gambar 4.5. Tampilan Halaman Daftar Vendor di Admin

### 3) Halaman detail vendor (*vendor detail*)

Halaman detail vendor berisi daftar produk dari vendor UKM yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama produk, tipe produk, opsi produk, dan stok produk. Daftar produk ini juga merupakan data yang didapatkan dari API Shopify milik perusahaan. Selain itu terdapat juga tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, *CSV*, dan *PDF*, serta kolom pencarian produk berdasarkan kata kunci dari seluruh teks yang ada pada tabel. Tampilan halaman detail vendor dapat dilihat pada Gambar 4.6.



Gambar 4.6. Tampilan Halaman Detail Vendor di Admin

#### 4) Halaman daftar toko (*store partner*)

Halaman daftar toko berisi daftar toko yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain logo, nama, alamat, kontak, dan aksi. Terdapat juga tombol untuk menambah toko baru (*add new store*), tombol detail, edit, dan hapus pada masing-masing toko, serta kolom pencarian toko berdasarkan kata kunci dari seluruh teks yang ada pada tabel. Tampilan halaman daftar toko dapat dilihat pada Gambar 4.7.

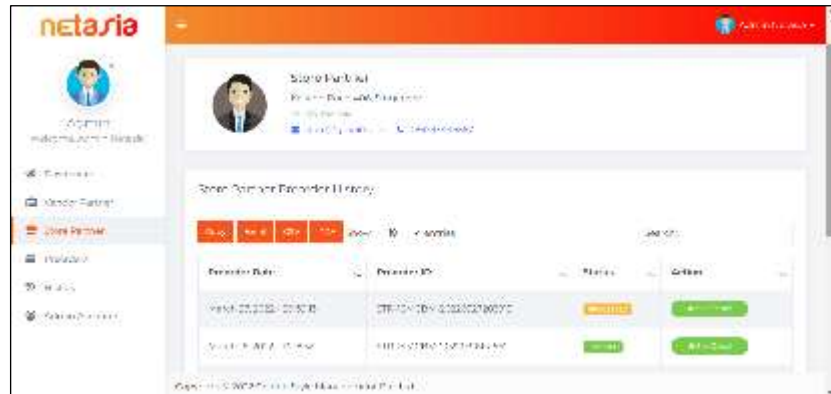


Gambar 4.7. Tampilan Halaman Daftar Toko di Admin

#### 5) Halaman detail toko (*store detail*)

Halaman detail toko berisi detail toko yang terdiri dari nama, alamat, id, dan kontak serta daftar riwayat *preorder* yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain tanggal *preorder*, id *preorder*, status, dan aksi. Status yang ditampilkan antara lain “*processed*” yang berarti *preorder* sedang diproses oleh admin maupun vendor, “*ready*” yang berarti produk telah siap untuk didistribusikan kepada toko, serta “*finished*” yang berarti *preorder* telah selesai. Terdapat juga tombol detail pada masing-masing *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, *CSV*, dan *PDF*, serta kolom pencarian *preorder* berdasarkan kata kunci dari seluruh teks yang ada pada tabel. Tampilan halaman detail toko dapat dilihat pada Gambar 4.8.





Gambar 4.8. Tampilan Halaman Detail Toko di Admin

6) Halaman prapemesanan (*preorders*)

Halaman prapemesanan terdiri dari dua tab, yakni tab *preorder* dari toko (*preorder from store*) dan tab *preorder* ke vendor (*preorder to vendor*). Tab *preorder* dari toko berisi daftar permintaan *preorder* dari toko yang disajikan dalam tabel. Kolom yang ada di tabel tersebut antara lain tanggal *preorder*, nama toko, id *preorder*, status, dan aksi. Hanya *preorder* dari toko yang berstatus “*requested*” yang tampil pada tab ini. Terdapat juga tombol proses *preorder* (*process*) pada masing-masing *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, *CSV*, dan *PDF*, serta kolom pencarian berdasarkan kata kunci dari seluruh teks yang ada pada tabel. Tampilan tab *preorder* dari toko dapat dilihat pada Gambar 4.9.



Gambar 4.9. Tampilan Tab *Preorder* dari Toko

Sementara itu tab *preorder* ke vendor berisi daftar *preorder* dari toko yang akan diteruskan kepada vendor yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain nama vendor, keterangan *preorder*, dan aksi. Terdapat juga tombol proses *preorder* ke vendor (*process to vendor*) pada masing-masing *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, CSV, dan PDF, serta kolom pencarian *preorder* berdasarkan kata kunci dari seluruh teks yang ada pada tabel. Tampilan tab *preorder* ke vendor di halaman prapemesanan dapat dilihat pada Gambar 4.10.



Gambar 4.10. Tampilan Tab *Preorder* ke Vendor

#### 7) Halaman proses *preorder* dari toko

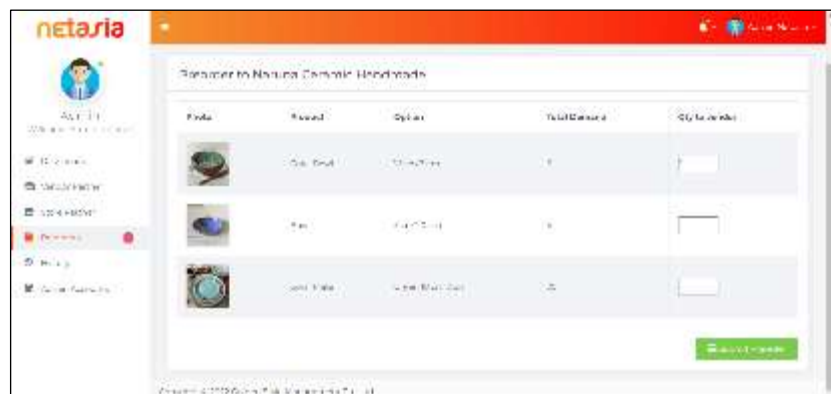
Halaman proses *preorder* dari toko (*process store preorder*) berisi daftar produk pesanan toko yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama vendor, nama produk, opsi produk, dan jumlah produk yang dipesan. Terdapat juga tombol untuk meneruskan pesanan toko ke vendor (*add to demand*). Tampilan halaman proses *preorder* dari toko dapat dilihat pada Gambar 4.11.



Gambar 4.11. Tampilan Halaman Proses *Preorder* dari Toko

8) Halaman proses *preorder* ke vendor

Halaman proses *preorder* ke vendor (*process to preorder*) berisi daftar produk pesanan toko yang akan diteruskan kepada vendor yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama produk, opsi produk, total jumlah permintaan dari toko, dan jumlah produk yang dipesan ke vendor. Terdapat juga tombol untuk mengirim *preorder* ke vendor (*submit preorder*). Tampilan halaman proses *preorder* ke vendor dapat dilihat pada Gambar 4.12.



Gambar 4.12. Tampilan Halaman Proses *Preorder* ke Vendor

9) Halaman riwayat prapemesanan (*history*)

Halaman riwayat prapemesanan terdiri dari dua tab, yakni *preorder* dari toko (*preorder from store*) dan *preorder* ke vendor (*preorder to vendor*). Tab *preorder* dari toko berisi daftar riwayat *preorder* dari toko yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain tanggal *preorder*, nama toko, id *preorder*, status, dan aksi. Status yang ditampilkan antara lain “*processed*” yang berarti *preorder* sedang diproses oleh admin maupun vendor, “*ready*” yang berarti produk telah siap untuk didistribusikan kepada toko, serta “*finished*” yang berarti *preorder* telah selesai. Terdapat juga tombol detail pada masing-masing *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, CSV, dan PDF, serta kolom pencarian *preorder* berdasarkan kata kunci dari seluruh teks yang ada pada tabel. Tampilan tab *preorder* dari toko di halaman riwayat prapemesanan dapat dilihat pada Gambar 4.13.

Preorder ID	Toko	Preorder ID	Status	Aksi
1234567890123456	PT. ABCD EFGHIJK	1234567890123456	Processed	Detail Export
1234567890123456	PT. ABCD EFGHIJK	1234567890123456	Ready	Detail Export
1234567890123456	PT. ABCD EFGHIJK	1234567890123456	Finished	Detail Export
1234567890123456	PT. ABCD EFGHIJK	1234567890123456	Processed	Detail Export

Gambar 4.13. Tampilan Tab Riwayat *Preorder* dari Toko

Sementara itu tab *preorder* ke vendor berisi daftar riwayat *preorder* ke vendor yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain tanggal *preorder*, nama vendor, id *preorder*, status, dan aksi. Status yang ditampilkan antara lain “*requested*” yang berarti *preorder* sedang dimintakan kepada vendor, “*accepted*” yang berarti *preorder* telah diterima oleh

vendor, “*delivered*” yang berarti produk telah dikirim oleh vendor, “*request product resend*” yang berarti admin meminta pengiriman ulang produk, serta “*received*” yang berarti produk telah diterima oleh admin. Terdapat juga tombol detail pada masing-masing *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, CSV, dan PDF, serta kolom pencarian *preorder* berdasarkan kata kunci dari seluruh teks yang ada pada tabel. Tampilan tab *preorder* ke vendor di halaman riwayat prapemesanan dapat dilihat pada Gambar 4.14.

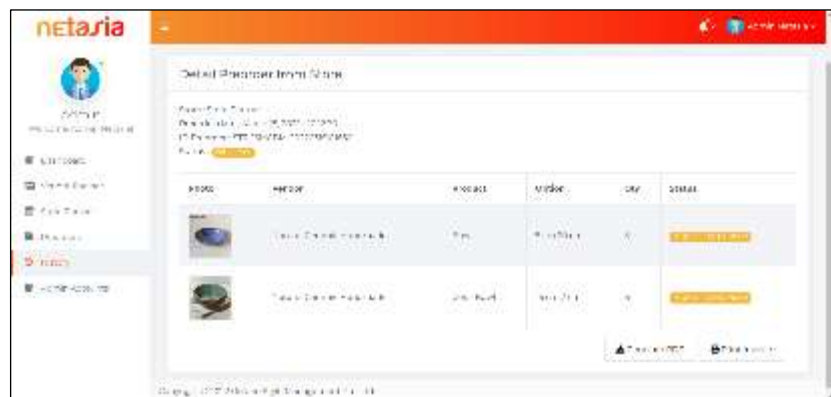


Gambar 4.14. Tampilan Tab Riwayat *Preorder* ke Vendor

#### 10) Halaman detail *preorder* dari toko

Halaman detail *preorder* dari toko berisi informasi *preorder* yang terdiri dari nama toko, tanggal *preorder*, id *preorder*, dan status *preorder*, serta daftar produk pesanan toko yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama vendor, nama produk, opsi produk, jumlah produk yang dipesan, dan status produk. Status produk yang ditampilkan pada halaman ini antara lain “*processed by admin*” yang berarti *preorder* sedang diproses oleh admin, “*requested to vendor*” yang berarti *preorder* telah diteruskan ke vendor, “*accepted by vendor*” yang berarti *preorder* telah diterima oleh vendor, “*delivered*” yang berarti produk telah dikirim oleh vendor, “*pending*” yang berarti

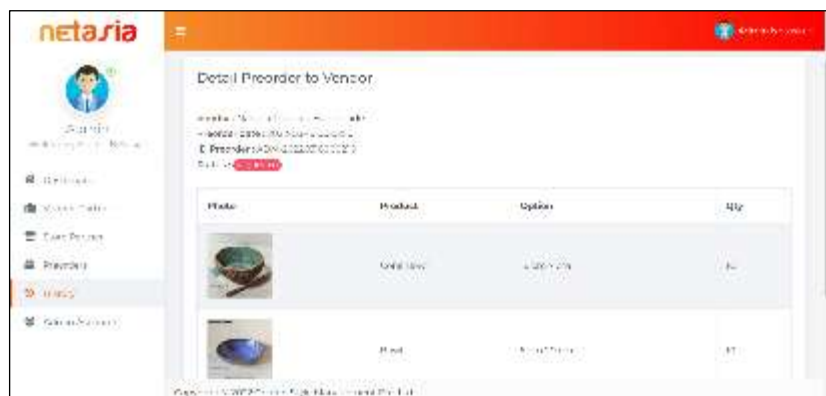
sedang dilakukan pengiriman ulang produk oleh vendor, “*ready*” yang berarti produk telah siap untuk didistribusikan, serta “*finished*” yang berarti *preorder* telah selesai. Terdapat juga tombol untuk mencetak halaman (*print*) dan tombol ekspor halaman ke PDF. Tampilan halaman detail *preorder* dari toko dapat dilihat pada Gambar 4.15.



Gambar 4.15. Tampilan Halaman Detail *Preorder* dari Toko

#### 11) Halaman detail *preorder* ke vendor

Halaman detail *preorder* ke vendor berisi informasi *preorder* yang terdiri dari nama vendor, tanggal *preorder*, id *preorder*, dan status, serta daftar produk pesanan admin ke vendor yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama produk, opsi produk, dan jumlah produk yang dipesan. Terdapat juga tombol untuk mencetak halaman (*print*) dan tombol ekspor halaman ke PDF. Tampilan halaman detail *preorder* ke vendor dapat dilihat pada Gambar 4.16.



Gambar 4.16. Tampilan Halaman Detail *Preorder* Ke Vendor

## 12) Halaman daftar admin (*admin accounts*)

Halaman daftar admin berisi daftar admin yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain id admin, nama admin, *username* admin, kontak admin, dan aksi. Terdapat juga tombol untuk menambah admin baru (*add new admin*), tombol edit dan hapus pada masing-masing admin, serta kolom pencarian admin berdasarkan kata kunci dari seluruh teks yang ada pada tabel. Tampilan halaman daftar admin dapat dilihat pada Gambar 4.17.



Gambar 4.17. Tampilan Halaman Daftar Admin

## 13) Halaman pengaturan akun admin (*settings*)

Halaman pengaturan akun admin berisi informasi akun yang terdiri dari nama admin, *username*, email, dan nomor

telepon, formulir untuk memperbarui akun (*update account*) yang terdiri dari kolom untuk memasukkan nama, email, nomor telepon, *username*, dan password, serta tombol untuk memperbarui akun (*update*). Tampilan halaman pengaturan akun admin dapat dilihat pada Gambar 4.18.



Gambar 4.18. Tampilan Halaman Pengaturan Akun Admin

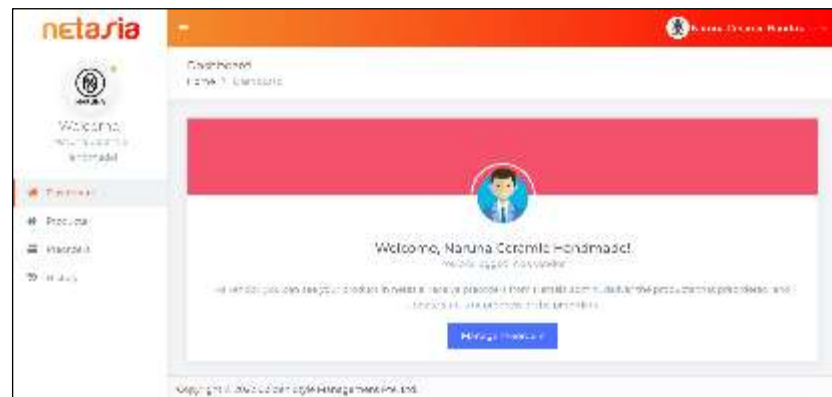
#### c. Halaman vendor

Halaman vendor memiliki empat menu utama antara lain dasbor (*dashboard*), daftar produk (*products*), prapemesanan dari admin (*preorders*), dan riwayat rapemesanan (*history*). Selain itu halaman vendor juga memiliki beberapa fitur lain seperti pemrosesan *preorder*, detail *preorder*, dan pengaturan akun. Menu dan fitur yang ada di halaman vendor diuraikan sebagai berikut:

##### 1) Halaman dasbor vendor (*dashboard*)

Ketika pengguna *login* ke aplikasi sebagai vendor, maka halaman pertama yang muncul setelah berhasil *login* adalah halaman dasbor vendor. Halaman dasbor vendor berisi pesan selamat datang, deskripsi peran vendor, dan tombol pintasan untuk menuju ke fitur utama sebagai vendor yaitu halaman *preorder* untuk mengelola *preorder* dari admin. Tampilan halaman dasbor vendor dapat dilihat pada Gambar 4.19.

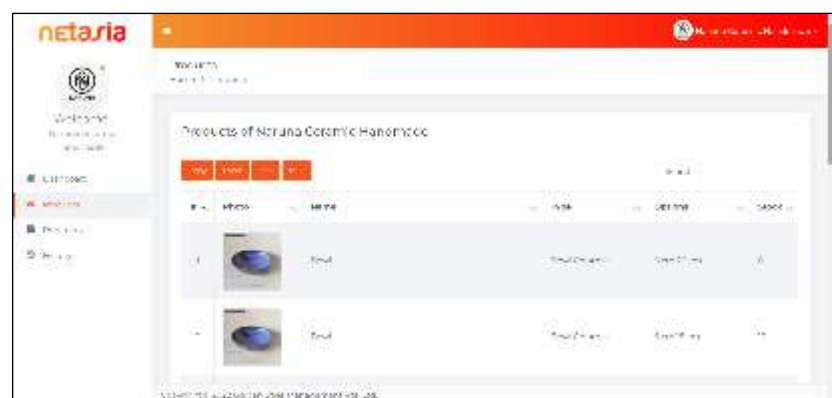




Gambar 4.19. Tampilan Halaman Dasbor Vendor

## 2) Halaman daftar produk (*products*)

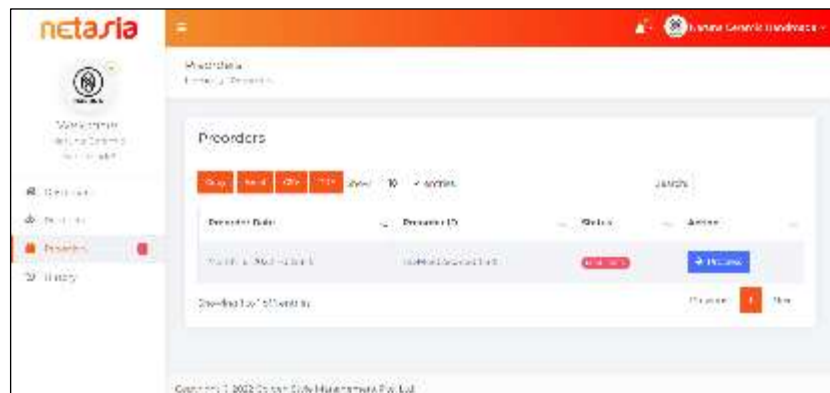
Halaman daftar produk berisi daftar produk vendor UKM yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama produk, tipe produk, opsi produk, dan stok produk. Daftar produk ini juga merupakan data yang didapatkan dari API Shopify milik perusahaan. Terdapat juga tombol salin dan ekspor tabel ke beberapa tipe *file* seperti *excel*, *CSV*, dan *PDF*, serta kolom pencarian produk berdasarkan kata kunci dari seluruh teks yang ada pada tabel. Tampilan halaman daftar produk dapat dilihat pada Gambar 4.20.



Gambar 4.20. Tampilan Halaman Daftar Produk di Vendor

### 3) Halaman *preorder* vendor (*preorders*)

Halaman *preorder* vendor berisi daftar permintaan *preorder* dari admin yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain tanggal *preorder*, id *preorder*, status, dan aksi. Hanya *preorder* yang berstatus “*requested*” yang tampil pada halaman ini. Terdapat juga tombol proses *preorder* (*process*) pada masing-masing *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, CSV, dan PDF, serta kolom pencarian *preorder* berdasarkan kata kunci dari seluruh teks yang ada pada tabel. Tampilan halaman *preorder* vendor dapat dilihat pada Gambar 4.21.



Gambar 4.21. Tampilan Halaman *Preorder* di Vendor

### 4) Halaman proses *preorder* vendor

Halaman proses *preorder* vendor berisi daftar produk pesanan admin yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama produk, opsi produk, dan jumlah produk yang dipesan. Terdapat juga tombol untuk menerima *preorder* (*accept preorder*). Tampilan halaman proses *preorder* vendor dapat dilihat pada Gambar 4.22.



Gambar 4.22. Tampilan Halaman Proses *Preorder* di Vendor

##### 5) Halaman riwayat *preorder* vendor (*history*)

Halaman riwayat *preorder* vendor berisi daftar riwayat *preorder* vendor yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain tanggal *preorder*, id *preorder*, status, dan aksi. Status yang ditampilkan antara lain “*accepted*” yang berarti *preorder* telah diterima, “*delivered*” yang berarti produk telah dikirim, “*request product resend*” yang berarti admin meminta pengiriman ulang produk, serta “*received*” yang berarti produk telah diterima oleh admin. Terdapat juga tombol untuk detail atau kirim (*deliver*) pada masing-masing *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, *CSV*, dan *PDF*, serta kolom pencarian *preorder* berdasarkan kata kunci dari seluruh teks yang ada pada tabel. Tampilan halaman riwayat *preorder* vendor dapat dilihat pada Gambar 4.23.



Gambar 4.23. Tampilan Halaman Riwayat *Preorder* di Vendor

6) Halaman detail *preorder* vendor

Halaman detail *preorder* vendor berisi informasi *preorder* yang terdiri dari tanggal *preorder* dan id *preorder*, serta daftar produk pesanan admin yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama produk, opsi produk, dan jumlah produk yang dipesan. Terdapat juga tombol untuk mencetak halaman (*print*) dan tombol ekspor halaman ke PDF. Tampilan halaman detail *preorder* vendor dapat dilihat pada Gambar 4.24.



Gambar 4.24. Tampilan Halaman Detail *Preorder* di Vendor

7) Halaman pengaturan akun vendor (*settings*)

Halaman pengaturan akun vendor berisi informasi akun yang terdiri dari nama vendor, *username*, email, dan nomor

telepon, formulir untuk memperbarui akun (*update account*) yang terdiri dari kolom untuk memasukkan email, nomor telepon, *username*, dan password, serta tombol untuk memperbarui akun (*update*). Tampilan halaman pengaturan akun vendor dapat dilihat pada Gambar 4.25.



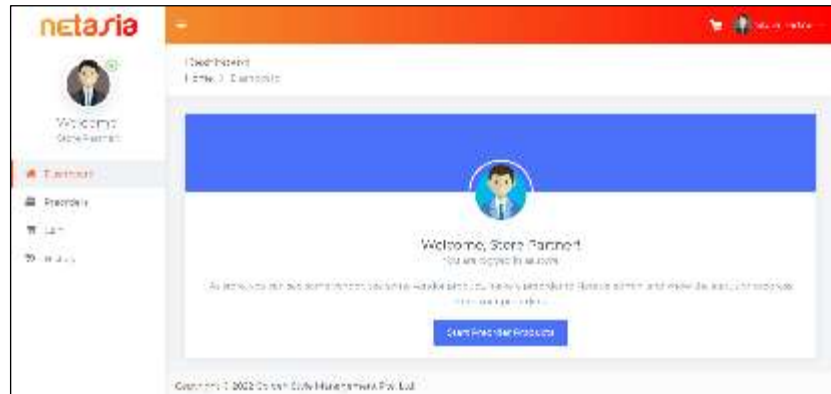
Gambar 4.25. Tampilan Halaman Pengaturan Akun Vendor

#### d. Halaman toko

Halaman toko memiliki beberapa menu antara lain dasbor (*dashboard*), prapemesanan (*preorders*), keranjang (*cart*), dan riwayat prapemesanan (*history*). Selain itu halaman toko juga memiliki beberapa fitur lain seperti daftar pilihan produk, detail produk, detail *preorder*, dan pengaturan akun. Menu dan fitur yang ada di halaman toko diuraikan sebagai berikut:

##### 1) Halaman dasbor toko (*dashboard*)

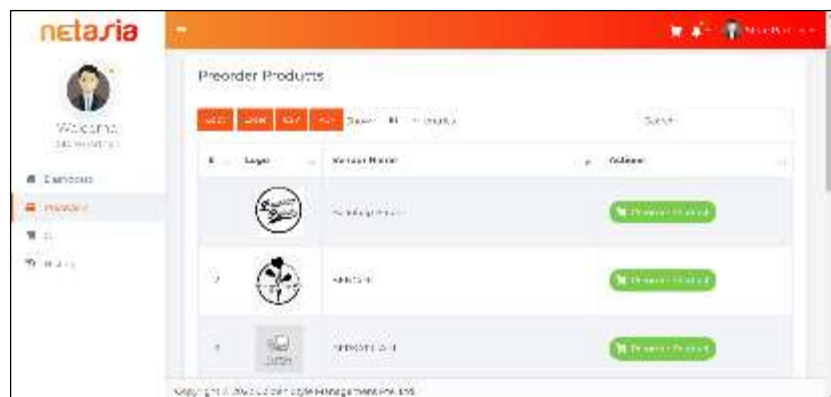
Ketika pengguna *login* ke aplikasi sebagai toko, maka halaman pertama yang muncul setelah berhasil *login* adalah halaman dasbor toko. Halaman dasbor toko berisi pesan selamat datang, deskripsi peran toko, dan tombol pintasan untuk menuju ke fitur utama sebagai toko yaitu halaman *preorder* untuk memesan produk kepada admin. Tampilan halaman dasbor toko dapat dilihat pada Gambar 4.26.



Gambar 4.26. Tampilan Halaman Dasbor Toko

## 2) Halaman *preorder* toko (*preorders*)

Halaman *preorder* toko berisi daftar vendor UKM yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain logo vendor, nama vendor, dan aksi. Daftar vendor ini merupakan data yang didapatkan dari API Shopify milik perusahaan. Terdapat juga tombol *preorder* produk (*preorder product*) pada masing-masing vendor. Selain itu terdapat juga tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, *CSV*, dan *PDF*, serta kolom pencarian vendor berdasarkan kata kunci dari seluruh teks yang ada pada tabel. Tampilan halaman *preorder* toko dapat dilihat pada Gambar 4.27.



Gambar 4.27. Tampilan Halaman *Preorder* di Toko

### 3) Halaman daftar pilihan produk

Halaman daftar pilihan produk berisi produk-produk yang akan dipesan oleh toko. Informasi yang ditampilkan pada halaman ini antara lain foto produk, nama produk, dan deskripsi singkat produk. Terdapat juga tombol untuk memesan produk (*preorders*) pada masing-masing produk. Tampilan halaman daftar pilihan produk dapat dilihat pada Gambar 4.28.



Gambar 4.28. Tampilan Halaman Daftar Pilihan Produk di Toko

### 4) Halaman detail produk

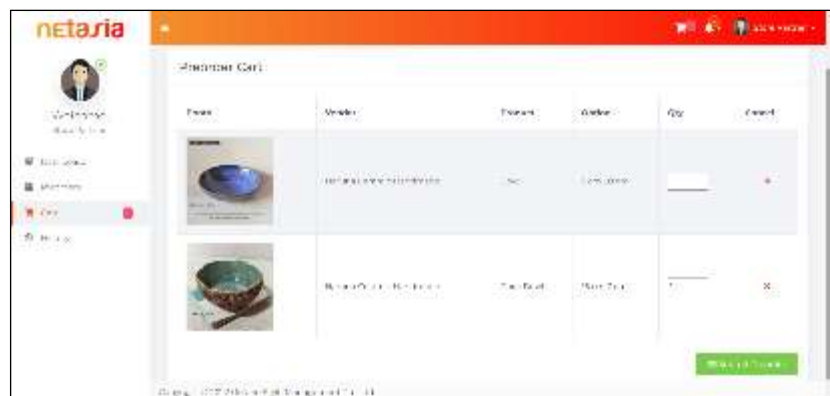
Halaman detail produk berisi informasi produk antara lain foto produk, nama produk, deskripsi produk, tipe produk, dan opsi produk. Terdapat juga tombol untuk memasukkan produk dalam keranjang (*add to cart*). Tampilan halaman detail produk dapat dilihat pada Gambar 4.29.



Gambar 4.29. Tampilan Halaman Detail Produk di Toko

5) Halaman keranjang (*cart*)

Halaman keranjang berisi daftar produk yang akan dipesan oleh toko yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama vendor, nama produk, opsi produk, jumlah produk yang akan dipesan, dan tombol pembatalan. Terdapat juga tombol untuk mengirim *preorder* ke admin (*submit preorder*). Tampilan halaman keranjang dapat dilihat pada Gambar 4.30.



Gambar 4.30. Tampilan Halaman Keranjang di Toko

6) Halaman riwayat *preorder* toko (*history*)

Halaman riwayat *preorder* toko berisi daftar riwayat *preorder* toko yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain tanggal *preorder*, id *preorder*, status, dan aksi. Status yang ditampilkan antara lain “*processed*” yang berarti *preorder* sedang diproses oleh admin maupun vendor, “*ready*” yang berarti produk telah siap untuk didistribusikan kepada toko, serta “*finished*” yang berarti *preorder* telah selesai. Terdapat juga tombol detail *preorder* dan batalkan *preorder* pada masing-masing *preorder*, tombol salin dan ekspor tabel ke beberapa tipe *file* antara lain *excel*, CSV, dan PDF, serta kolom pencarian *preorder* berdasarkan kata kunci dari seluruh teks yang ada pada tabel. Tampilan halaman riwayat *preorder* toko dapat dilihat pada Gambar 4.31.

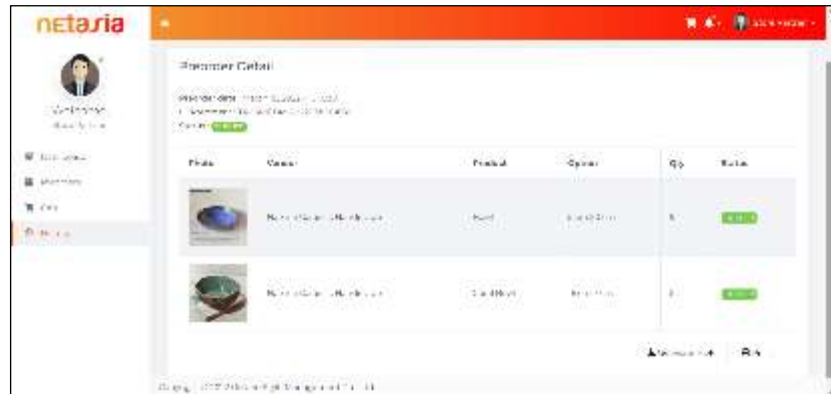




Gambar 4.31. Tampilan Halaman Riwayat *Preorder* di Toko

#### 7) Halaman detail *preorder* toko

Halaman detail *preorder* toko berisi informasi *preorder* yang terdiri dari tanggal *preorder*, id *preorder*, dan status, serta daftar produk pesanan toko yang disajikan dalam tabel. Kolom yang ada pada tabel tersebut antara lain foto produk, nama vendor, nama produk, opsi produk, jumlah produk yang dipesan, dan status produk. Status produk yang ditampilkan pada halaman ini antara lain “*processed by admin*” yang berarti *preorder* sedang diproses oleh admin, “*requested to vendor*” yang berarti *preorder* telah diteruskan ke vendor, “*accepted by vendor*” yang berarti *preorder* telah diterima oleh vendor, “*delivered*” yang berarti produk telah dikirim oleh vendor, “*pending*” yang berarti sedang dilakukan pengiriman ulang produk oleh vendor, “*ready*” yang berarti produk telah siap untuk didistribusikan, serta “*finished*” yang berarti *preorder* telah selesai. Terdapat juga tombol untuk mencetak halaman (*print*) dan tombol ekspor halaman ke PDF. Tampilan halaman detail *preorder* toko dapat dilihat pada Gambar 4.32.



Gambar 4.32. Tampilan Halaman Detail *Preorder* di Toko

8) Halaman pengaturan akun toko (*settings*)

Halaman pengaturan akun toko berisi informasi akun yang terdiri dari nama toko, *username*, email, dan nomor telepon, formulir untuk memperbarui akun (*update account*) yang terdiri dari kolom untuk memasukkan nama, alamat, email, nomor telepon, *username*, dan password, serta tombol untuk memperbarui akun (*update*). Tampilan halaman pengaturan akun toko dapat dilihat pada Gambar 4.33.



Gambar 4.33. Tampilan Halaman Pengaturan Akun Toko

## B. Hasil Pengujian

Pengujian merupakan salah satu tahap yang penting dalam pengembangan perangkat lunak. Pengujian ini bertujuan untuk mengetahui apakah perangkat lunak yang dikembangkan sudah sesuai dengan kriteria dan tujuan perancangan perangkat lunak tersebut. Pengujian perangkat lunak yang dilakukan dalam penelitian ini antara lain dengan pengujian *black-box*, pengujian *white-box*, *User Acceptance Testing (UAT)*, dan pengujian kompatibilitas aplikasi *browser*. Hasil pengujian perangkat lunak berupa aplikasi berbasis web yang dikembangkan pada penelitian ini diuraikan sebagai berikut:

### 1. Pengujian *black-box*

Pengujian *black-box* dilakukan dengan melibatkan tiga responden yang merupakan dosen dari Program Studi Informatika Universitas PGRI Semarang. Hasil pengujian pada halaman *login* ditampilkan pada Tabel 4.1, hasil pengujian pada halaman admin ditampilkan pada Tabel 4.2, hasil pengujian pada halaman vendor ditampilkan pada Tabel 4.3, dan hasil pengujian pada halaman toko ditampilkan pada Tabel 4.4. Hasil pengujian *black-box* dalam menguji spesifikasi fungsional masing-masing halaman pada aplikasi berbasis web ini diuraikan secara rinci sebagai berikut:

#### a. Pengujian halaman *login*

Tabel 4.1. Hasil Pengujian *Black-box* Halaman *Login*

<i>Test case</i>	Keterangan (berhasil/gagal)		
	Responden 1	Responden 2	Responden 3
<b>1. Formulir <i>login</i></b>			
<i>Username</i> dan <i>password</i> diisi dengan data yang benar dengan level admin	Berhasil	Berhasil	Berhasil

<i>Test case</i>	<b>Keterangan (berhasil/gagal)</b>		
	<b>Responden 1</b>	<b>Responden 2</b>	<b>Responden 3</b>
<i>Username dan password</i> diisi dengan data yang benar dengan level vendor	Berhasil	Berhasil	Berhasil
<i>Username dan password</i> diisi dengan data yang benar dengan level toko	Berhasil	Berhasil	Berhasil
<i>Username dan password</i> diisi dengan data yang salah	Berhasil	Berhasil	Berhasil

b. Pengujian halaman admin

Tabel 4.2. Hasil Pengujian *Black-box* Halaman Admin

<i>Test case</i>	<b>Keterangan (berhasil/gagal)</b>		
	<b>Responden 1</b>	<b>Responden 2</b>	<b>Responden 3</b>
<b>1. Menu dasbor (<i>dashboard</i>)</b>			
Memilih menu dasbor	Berhasil	Berhasil	Berhasil
<b>2. Menu daftar vendor (<i>vendor partner</i>)</b>			
Memilih menu <i>vendor partner</i>	Berhasil	Berhasil	Berhasil
Klik tombol detail	Berhasil	Berhasil	Berhasil
Klik tombol <i>add</i>	Berhasil	Berhasil	Berhasil

<i>Test case</i>	<b>Keterangan (berhasil/gagal)</b>		
	<b>Responden 1</b>	<b>Responden 2</b>	<b>Responden 3</b>
<i>account</i>			
Mengisi form penambahan akun vendor baru dan klik tombol <i>register</i>	Berhasil	Berhasil	Berhasil
Klik tombol <i>edit account</i>	Berhasil	Berhasil	Berhasil
Mengisi form edit akun vendor dan klik tombol <i>update</i>	Berhasil	Berhasil	Berhasil
<b>3. Menu daftar toko (<i>store partner</i>)</b>			
Memilih menu <i>store partner</i>	Berhasil	Berhasil	Berhasil
Klik tombol <i>add new store</i>	Berhasil	Berhasil	Berhasil
Mengisi form penambahan akun toko baru dan klik tombol <i>register</i>	Berhasil	Berhasil	Berhasil
Klik tombol detail	Berhasil	Berhasil	Berhasil
Klik tombol edit	Berhasil	Berhasil	Berhasil
Mengisi form edit akun toko dan klik tombol <i>update</i>	Berhasil	Berhasil	Berhasil

<i>Test case</i>	<b>Keterangan (berhasil/gagal)</b>		
	<b>Responden 1</b>	<b>Responden 2</b>	<b>Responden 3</b>
Klik tombol hapus	Berhasil	Berhasil	Berhasil
<b>4. Menu prapemesanan (<i>preorders</i>)</b>			
Memilih menu <i>preorders</i>	Berhasil	Berhasil	Berhasil
Klik tab <i>preorder from store</i>	Berhasil	Berhasil	Berhasil
Klik tab <i>preorder to vendor</i>	Berhasil	Berhasil	Berhasil
Klik tombol <i>process</i> di tab <i>preorder from store</i>	Berhasil	Berhasil	Berhasil
Klik tombol <i>add to demand</i> di halaman <i>process preorder</i>	Berhasil	Berhasil	Berhasil
Klik tombol <i>process to vendor</i> di tab <i>preorder to vendor</i>	Berhasil	Berhasil	Berhasil
Klik tombol <i>submit preorder to vendor</i>	Berhasil	Berhasil	Berhasil
<b>5. Menu riwayat prapemesanan (<i>history</i>)</b>			
Memilih menu <i>history</i>	Berhasil	Berhasil	Berhasil
Klik tab <i>preorder</i>	Berhasil	Berhasil	Berhasil

<i>Test case</i>	<b>Keterangan (berhasil/gagal)</b>		
	<b>Responden 1</b>	<b>Responden 2</b>	<b>Responden 3</b>
<i>from store</i>			
Klik tab <i>preorder to vendor</i>	Berhasil	Berhasil	Berhasil
Klik tombol detail di tab <i>preorder from store</i>	Berhasil	Berhasil	Berhasil
Klik tombol detail di tab <i>preorder to vendor</i>	Berhasil	Berhasil	Berhasil
Jika status di tab <i>preorder to vendor "delivered"</i> klik tombol <i>check</i>	Berhasil	Berhasil	Berhasil
Klik tombol <i>request product resend</i> di halaman detail <i>preorder</i>	Berhasil	Berhasil	Berhasil
Mengisi form keterangan permintaan pengiriman ulang dan klik <i>request</i>	Berhasil	Berhasil	Berhasil
Klik tombol <i>receive product</i> di halaman detail <i>preorder</i>	Berhasil	Berhasil	Berhasil
Jika status di tab <i>preorder from</i>	Berhasil	Berhasil	Berhasil

<i>Test case</i>	<b>Keterangan (berhasil/gagal)</b>		
	<b>Responden 1</b>	<b>Responden 2</b>	<b>Responden 3</b>
<i>store “ready”</i> klik tombol detail			
Klik tombol <i>finish</i> <i>preorder</i> di halaman detail <i>preorder</i>	Berhasil	Berhasil	Berhasil
<b>6. Menu akun admin (<i>admin accounts</i>)</b>			
Memilih menu <i>admin accounts</i>	Berhasil	Berhasil	Berhasil
Klik tombol <i>add</i> <i>new admin</i>	Berhasil	Berhasil	Berhasil
Mengisi form penambahan akun admin baru dan klik tombol <i>register</i>	Berhasil	Berhasil	Berhasil
Klik tombol edit	Berhasil	Berhasil	Berhasil
Mengisi form edit akun admin dan klik tombol <i>update</i>	Berhasil	Berhasil	Berhasil
Klik tombol hapus	Berhasil	Berhasil	Berhasil
<b>7. Menu pengaturan akun</b>			
Klik tombol <i>settings</i>	Berhasil	Berhasil	Berhasil
Mengisi form pengaturan akun	Berhasil	Berhasil	Berhasil



<i>Test case</i>	<b>Keterangan (berhasil/gagal)</b>		
	<b>Responden 1</b>	<b>Responden 2</b>	<b>Responden 3</b>
dan klik tombol <i>update</i>			
<b>8. Keluar dari halaman admin</b>			
Klik tombol <i>logout</i>	Berhasil	Berhasil	Berhasil

c. Pengujian halaman vendor

Tabel 4.3. Hasil Pengujian *Black-box* Halaman Vendor

<i>Test case</i>	<b>Keterangan (berhasil/gagal)</b>		
	<b>Responden 1</b>	<b>Responden 2</b>	<b>Responden 3</b>
<b>1. Menu dasbor (<i>dashboard</i>)</b>			
Memilih menu dasbor	Berhasil	Berhasil	Berhasil
<b>2. Menu daftar produk (<i>products</i>)</b>			
Memilih menu <i>products</i>	Berhasil	Berhasil	Berhasil
<b>3. Menu prapemesanan (<i>preorders</i>)</b>			
Memilih menu <i>preorders</i>	Berhasil	Berhasil	Berhasil
Klik tombol <i>process</i> di salah satu pesanan	Berhasil	Berhasil	Berhasil
Klik tombol <i>accept preorder</i> di halaman <i>process preorder</i>	Berhasil	Berhasil	Berhasil
<b>4. Menu riwayat prapemesanan (<i>history</i>)</b>			

<i>Test case</i>	<b>Keterangan (berhasil/gagal)</b>		
	<b>Responden 1</b>	<b>Responden 2</b>	<b>Responden 3</b>
Memilih menu <i>history</i>	Berhasil	Berhasil	Berhasil
Klik tombol detail	Berhasil	Berhasil	Berhasil
Jika status “ <i>accepted</i> ” klik tombol <i>deliver</i>	Berhasil	Berhasil	Berhasil
Klik tombol <i>deliver product</i> di halaman detail <i>preorder</i>	Berhasil	Berhasil	Berhasil
Jika status <i>request product resend</i> klik <i>respond</i>	Berhasil	Berhasil	Berhasil
Klik tombol <i>contact admin</i> di halaman detail <i>preorder</i>	Berhasil	Berhasil	Berhasil
Klik tombol <i>resend product</i> di halaman detail <i>preorder</i>	Berhasil	Berhasil	Berhasil
<b>5. Menu pengaturan akun</b>			
Klik tombol <i>settings</i>	Berhasil	Berhasil	Berhasil
Mengisi form pengaturan akun dan klik tombol <i>update</i>	Berhasil	Berhasil	Berhasil

<i>Test case</i>	<b>Keterangan (berhasil/gagal)</b>		
	<b>Responden 1</b>	<b>Responden 2</b>	<b>Responden 3</b>
<b>6. Keluar dari halaman vendor</b>			
Klik tombol <i>logout</i>	Berhasil	Berhasil	Berhasil

## d. Pengujian halaman toko

Tabel 4.4. Hasil Pengujian *Black-box* Halaman Toko

<i>Test case</i>	<b>Keterangan (berhasil/gagal)</b>		
	<b>Responden 1</b>	<b>Responden 2</b>	<b>Responden 3</b>
<b>1. Menu dasbor (<i>dashboard</i>)</b>			
Memilih menu dasbor	Berhasil	Berhasil	Berhasil
<b>2. Menu prapemesanan (<i>preorders</i>)</b>			
Memilih menu <i>preorders</i>	Berhasil	Berhasil	Berhasil
Klik tombol <i>pre-order product</i> di salah satu vendor	Berhasil	Berhasil	Berhasil
Klik tombol <i>preorder</i> pada salah satu produk	Berhasil	Berhasil	Berhasil
Memilih opsi produk dan klik tombol <i>add to cart</i>	Berhasil	Berhasil	Berhasil
<b>3. Menu keranjang (<i>cart</i>)</b>			
Memilih menu <i>cart</i>	Berhasil	Berhasil	Berhasil
Klik tombol	Berhasil	Berhasil	Berhasil

<i>Test case</i>	<b>Keterangan (berhasil/gagal)</b>		
	<b>Responden 1</b>	<b>Responden 2</b>	<b>Responden 3</b>
<i>submit preorder</i>			
<b>4. Menu riwayat prapemesanan (<i>history</i>)</b>			
Memilih menu <i>history</i>	Berhasil	Berhasil	Berhasil
Klik tombol detail	Berhasil	Berhasil	Berhasil
Klik tombol <i>cancel preorder</i>	Berhasil	Berhasil	Berhasil
Jika status “ <i>canceled</i> ” klik tombol <i>edit preorder</i>	Berhasil	Berhasil	Berhasil
Memperbarui data <i>preorder</i> dan klik tombol <i>submit prorder</i> di halaman <i>cart</i>	Berhasil	Berhasil	Berhasil
Jika status “ <i>canceled</i> ” klik tombol <i>delete preorder</i>	Berhasil	Berhasil	Berhasil
<b>5. Menu pengaturan akun</b>			
Klik tombol <i>settings</i>	Berhasil	Berhasil	Berhasil
Mengisi form pengaturan akun dan klik tombol <i>update</i>	Berhasil	Berhasil	Berhasil

<i>Test case</i>	Keterangan (berhasil/gagal)		
	Responden 1	Responden 2	Responden 3
<b>6. Keluar dari halaman toko</b>			
Klik tombol <i>logout</i>	Berhasil	Berhasil	Berhasil

## 2. Pengujian *white-box*

Pengujian *white-box* dilakukan oleh peneliti sendiri dengan menguji struktur kode dari aplikasi berbasis web yang dikembangkan pada penelitian ini. Struktur kode yang diuji adalah alur program yang memiliki perulangan, yaitu proses *preorder* mulai dari toko ke admin sampai dengan admin ke vendor. Terdapat beberapa kali perulangan dalam proses tersebut seperti ketika toko membatalkan *preorder* dan memperbarui informasi *preorder* serta ketika admin meminta pengiriman ulang produk kepada vendor. Hasil pengujian *white-box* pada penelitian ini dapat dilihat pada Tabel 4.5.

Tabel 4.5. Hasil Pengujian *White-box*

Node	Struktur Kode ( <i>script</i> )	Keterangan
1	<pre> public function add_to_cart() {     \$id_user=\$this-&gt;input-&gt;post('id_user');     \$id_cart=\$this-&gt;input-&gt;post('id_cart');     \$time_cart = date('Y m d H:i:s');     \$name=\$this-&gt;input-&gt;post('name');     \$vendor=\$this-&gt;input-&gt;post('vendor');     \$product=\$this-&gt;input-&gt;post('product');     \$handle=\$this-&gt;input-&gt;post('handle');     \$product_id=\$this-&gt;input-&gt;post('product_id');     \$options=\$this-&gt;input-&gt;post('options');     \$option2=\$this-&gt;input-&gt;post('option2');     \$option3=\$this-&gt;input-&gt;post('option3');     \$product_photo=\$this-&gt;input-&gt;post('product_photo');     \$price=\$this-&gt;input-&gt;post('price');      \$replace = str_replace(" ", "-", \$id_cart);      \$data = array('name' =&gt;\$name, 'id_user' =&gt;\$id_user,     'handle' =&gt;\$handle, 'product' =&gt;\$product,     'product_id' =&gt;\$product_id, 'vendor' =&gt;\$vendor,     'id_cart' =&gt;\$replace, 'price' =&gt;\$price, 'options'     =&gt; '\$option1.'.'\$option2.'.'\$option3.'.'     'product_photo' =&gt;\$product_photo, 'time_cart'     =&gt;\$time_cart, 'status' =&gt;'CART');     \$this-&gt;M_all-&gt;insert_data('cart',\$data);     redirect('start/preorder/product/' . \$handle); } </pre>	Toko memilih produk dan memasukkannya ke keranjang

2	<pre> public function submit_preorder() {     \$status = \$_POST['status'];     \$idpreorder = \$_POST['idpreorder'];     \$total_bayar = \$_POST['total_bayar'];     \$jml_produk = \$_POST['jml_produk'];     \$subtotal = \$_POST['subtotal'];     \$id_cart = \$_POST['id_cart'];     \$id_user = \$_POST['id_user'];     \$time_preorder = date('Y-n-d H:i:s');     \$id_preorder = \$id_user."-".str_replace(array(":",     "-", " "), "", \$time_preorder);     \$name = \$_POST['name'];      if(\$status=="CART"){         if(!empty(\$subtotal)){             for(\$i = 0; \$i &lt; count(\$subtotal); \$i++){                 if(!empty(\$subtotal[\$i])){                     \$data = array(                         'amount' =&gt; \$total_bayar,                         'time_preorder' =&gt; \$                             time_preorder,                         'id_preorder' =&gt; \$id_preorder,                         'quantity' =&gt; \$jml_produk[\$i],                         'subtotal' =&gt; \$subtotal[\$i],                         'status' =&gt; 'REQUESTED',                         'id_cart' =&gt; NULL                     );                     \$where = array('id_cart' =&gt; \$                         id_cart[\$i]);                     \$this-&gt;M_all-&gt;update_data('cart',\$                         data,\$where);                 }             }              \$value = array(                 'amount' =&gt; \$total_bayar,                 'time_preorder' =&gt; \$time_preorder,                 'id_preorder' =&gt; \$id_preorder,                 'name' =&gt; \$name,                 'id_user' =&gt; \$id_user,                 'status' =&gt; 'REQUESTED',             );             \$this-&gt;db-&gt;insert('preorder',\$value);             redirect('store/history');         }     } } </pre>	Produk yang dimasuk- sukkan ke keranjang oleh toko dikirim ke admin
3	<pre> public function cancel_preorder(\$id_preorder) {     \$where = array('id_preorder' =&gt; \$id_preorder, );     \$data = array(         'status' =&gt; 'CANCELLED'     );     \$this-&gt;M_all-&gt;update_data('cart',\$data,\$where);     \$this-&gt;M_all-&gt;update_data('preorder',\$data,\$where);     redirect('store/history'); } </pre>	Toko membatalkan <i>preorder</i> ke admin
4	<pre> } else if(\$status=="CANCELLED"){     if(!empty(\$subtotal)){         for(\$i = 0; \$i &lt; count(\$subtotal); \$i++){             if(!empty(\$subtotal[\$i])){                 \$data = array(                     'amount' =&gt; \$total_bayar,                     'time_preorder' =&gt; \$                         time_preorder,                     'id_preorder' =&gt; \$id_preorder,                     'quantity' =&gt; \$jml_produk[\$i],                     'subtotal' =&gt; \$subtotal[\$i],                     'status' =&gt; 'REQUESTED',                     'id_cart' =&gt; NULL                 );                 \$where = array('id_preorder' =&gt; \$                     idpreorder);                 \$this-&gt;M_all-&gt;update_data('cart',\$                     data,\$where);             }         }     } } </pre>	Toko memperbarui informasi <i>preorder</i> dan mengirim ke admin

	<pre> \$value = array(     'amount' =&gt; \$total_bayar,     'line_preorder' =&gt; \$line_preorder,     'id_preorder' =&gt; \$id_preorder,     'name' =&gt; \$name,     'id_user' =&gt; \$id_user,     'status' =&gt; 'REQUESTED', ); \$where = array('id_preorder' =&gt; \$idpreorder); \$this-&gt;M_all-&gt;update_data('preorder',\$value,\$ where); redirect('store/history'); } } </pre>	
5	<pre> public function process_preorder(\$id_preorder) {     \$where = array('id_preorder' =&gt; \$id_preorder);     \$data = array('status' =&gt; 'PROCESSED BY ADMIN');     \$data2 = array('status' =&gt; 'PROCESSED');     \$this-&gt;M_all-&gt;update_data('cart',\$data,\$where);     \$this-&gt;M_all-&gt;update_data('preorder',\$data2,\$where);     redirect('admin/preorder'); } </pre>	Admin menerima <i>preorder</i> dari toko dan memasukkan ke antrian <i>preorder</i> ke vendor
6	<pre> public function submit_preorder() {     \$vendor = \$_POST['vendor'];     \$jml_produk = \$_POST['jml_produk'];     \$id_user = \$_POST['id_user'];     \$time_tovendor = date('Y m d H:i:s');     \$id_tovendor = "ADM".str_replace(array(":", " ",     " "), "", \$time_tovendor);      for(\$i = 0; \$i &lt; count(\$jml_produk); \$i++){         if(!empty(\$jml_produk[\$i])){             \$data = array(                 'time_tovendor' =&gt; \$time_tovendor,                 'id_tovendor' =&gt; \$id_tovendor,                 'qty_tovendor' =&gt; \$jml_produk[\$i],                 'status' =&gt; 'REQUESTED TO VENDOR'             );             \$where = array('vendor' =&gt; \$vendor, '             status' =&gt; 'PROCESSED BY ADMIN');             \$this-&gt;M_all-&gt;update_data('cart',\$data,\$             where);         }     }      \$value = array(         'amount' =&gt; NULL,         'line_preorder' =&gt; \$line_tovendor,         'id_preorder' =&gt; \$id_tovendor,         'name' =&gt; 'Admin',         'id_user' =&gt; \$id_user,         'status' =&gt; 'REQUESTED',     );     \$this-&gt;db-&gt;insert('preorder',\$value);     redirect('admin/preorder#vendor'); } } </pre>	Admin meneruskan permintaan <i>preorder</i> dari toko ke vendor

7	<pre> public function accept_preorder() {     \$time_now = date('Y-m-d H:i:s');     \$notes=\$this-&gt;input-&gt;post('notes');     \$id_preorder=\$this-&gt;input-&gt;post('id_preorder');      \$where = array('id_tovendor' =&gt; \$id_preorder);     \$where2 = array('id_preorder' =&gt; \$id_preorder);     \$data = array('status' =&gt; 'ACCEPTED BY VENDOR');     \$data2 = array('status' =&gt; 'ACCEPTED',         'vendor_notes' =&gt; \$notes,         'update_notes2' =&gt; \$time_now);     \$this-&gt;M_all-&gt;update_data('cart',\$data,\$where);     \$this-&gt;M_all-&gt;update_data('preorder',\$data2,\$where2);     redirect('vendor/preorder'); } </pre>	Vendor menerima <i>preorder</i> dari admin
8	<pre> public function deliver(\$id_preorder) {     \$time_now = date('Y-m-d H:i:s');     \$where = array('id_tovendor' =&gt; \$id_preorder);     \$where2 = array('id_preorder' =&gt; \$id_preorder);     \$data = array('status' =&gt; 'DELIVERED');     \$data2 = array('status' =&gt; 'DELIVERED',         'time_delivered' =&gt; \$time_now);     \$this-&gt;M_all-&gt;update_data('cart',\$data,\$where);     \$this-&gt;M_all-&gt;update_data('preorder',\$data2,\$where2);     redirect('vendor/history'); } </pre>	Vendor mengirim status ke admin bahwa produk sedang dikirimkan
9	<pre> public function detail_preorder(\$id_preorder) {     \$where = array('id_tovendor' =&gt; \$id_preorder);     \$where2 = array('id_preorder' =&gt; \$id_preorder);     \$data['penduduk']=\$this-&gt;M_all-&gt;         select_groupBy_where('',             'cart','product_id',\$where)-&gt;result();     \$data['preorder']=\$this-&gt;M_all-&gt;select_where(         'preorder',         \$where2)-&gt;result();     \$this-&gt;load-&gt;view('admin/layout/header');     \$this-&gt;load-&gt;view('admin/preorder_detail_tovendor',         \$data);     \$this-&gt;load-&gt;view('admin/layout/footer'); } </pre>	Admin menerima status dari vendor dan memeriksa produk yang dikirim
10	<pre> public function resend(){     \$time_now = date('Y-m-d H:i:s');     \$notes=\$this-&gt;input-&gt;post('notes');     \$id_preorder=\$this-&gt;input-&gt;post('id_preorder');      \$where = array('id_tovendor' =&gt; \$id_preorder);     \$where2 = array('id_preorder' =&gt; \$id_preorder);     \$data = array('status' =&gt; 'PENDING');     \$data2 = array('status' =&gt; 'REQUEST PRODUCT RESEND',         'admin_notes' =&gt; \$notes, 'update_notes2' =&gt; \$time_now);     \$this-&gt;M_all-&gt;update_data('cart',\$data,\$where);     \$this-&gt;M_all-&gt;update_data('preorder',\$data2,\$where2);     redirect('admin/history#vendor'); } </pre>	Admin meminta pengiriman ulang ( <i>resend</i> ) produk yang tidak sesuai ke vendor
11	<pre> public function detail(\$id_preorder) {     \$where = array('id_tovendor' =&gt; \$id_preorder);     \$data['penduduk']=\$this-&gt;M_all-&gt;         select_groupBy_where('',             'cart','product_id',\$where)-&gt;result();     \$this-&gt;load-&gt;view('vendor/layout/header');     \$this-&gt;load-&gt;view('vendor/preorder_detail',\$data);     \$this-&gt;load-&gt;view('vendor/layout/footer'); } </pre>	Vendor menerima permintaan <i>resend</i> produk dengan opsi menghubungi admin/kirim ulang produk



12	<pre> public function receive(\$id_tovendor) {     \$where = array('id_tovendor' =&gt; \$id_tovendor);     \$where2 = array('id_preorder' =&gt; \$id_tovendor);     \$data = array('status' =&gt; 'PRODUCT IS READY');     \$data2 = array('status' =&gt; 'RECEIVED');     \$data3 = array('status' =&gt; 'READY');      \$sql = "SELECT cart.id_preorder, COUNT(cart.id_preorder) AS jml_all, jml_finished.jml AS jml_finished, (COUNT(cart.id_preorder))-(jml_finished.jml) AS selisih FROM cart LEFT JOIN (SELECT id_preorder, COUNT(id_preorder) AS jml FROM 'cart' WHERE id_tovendor='\$id_tovendor' GROUP BY id_preorder) AS jml_finished ON cart.id_preorder=jml_finished.id_preorder WHERE id_tovendor='\$id_tovendor' GROUP BY cart.id_preorder"; \$query=\$this-&gt;db-&gt;query(\$sql); foreach (\$query-&gt;result() as \$prd) if(\$prd-&gt;selisih == 0) {     \$where2 = array('id_preorder' =&gt; \$prd-&gt;id_preorder);     \$this-&gt;M_all-&gt;update_data('preorder',\$data3,\$where2); }  \$this-&gt;M_all-&gt;update_data('cart',\$data,\$where); \$this-&gt;M_all-&gt;update_data('preorder',\$data2,\$where2); redirect('admin/history#vendor'); } </pre>	Admin menerima produk yang dikirim oleh vendor dan mengirim status ke toko bahwa produk telah siap didistribusikan
13	<pre> public function finished(\$id_preorder) {     \$where = array('id_preorder' =&gt; \$id_preorder);     \$data = array('status' =&gt; 'FINISHED');     \$this-&gt;M_all-&gt;update_data('cart',\$data,\$where);     \$this-&gt;M_all-&gt;update_data('preorder',\$data,\$where);     redirect('admin/history'); } </pre>	Admin mengirim status ke toko bahwa <i>preorder</i> telah selesai

Hasil pengujian *white-box* yang telah didapatkan kemudian diolah dengan menggunakan beberapa teknik dalam pengujian *white-box* antara lain pengujian berbasis alur (*basis path testing*), kompleksitas siklomatis (*cyclomatic complexity*), jalur independen (*independent path*), dan pengujian nilai (*value test*). Pengolahan hasil pengujian ini akan dibahas pada bagian pembahasan.

### 3. *User Acceptance Testing (UAT)*

Pengujian UAT dilakukan dengan melibatkan 3 responden yang terdiri dari 1 pimpinan perusahaan sebagai admin, 1 perwakilan dari pihak vendor, dan 1 perwakilan dari pihak toko. Hasil pengujian UAT pada penelitian ini dapat dilihat pada Tabel 4.6.

Tabel 4.6. Hasil *User Acceptance Testing (UAT)*

No	Pertanyaan	Skor dari responden		
		Admin	Vendor	Toko
Aspek Kemanfaatan				
1	Apakah aplikasi berbasis web ini dapat membantu pengelolaan distribusi produk UKM di Netasia?	4	5	4
2	Apakah aplikasi berbasis web ini bermanfaat untuk admin dalam mengelola distribusi produk UKM di Netasia?	5	5	5
3	Apakah aplikasi berbasis web ini bermanfaat untuk vendor dalam menerima <i>preorder</i> dari Netasia?	5	5	5
4	Apakah aplikasi berbasis web ini bermanfaat untuk toko dalam melakukan <i>preorder</i> produk UKM di Netasia?	5	5	5
5	Apakah aplikasi berbasis web ini membuat pengelolaan distribusi produk UKM di Netasia menjadi lebih efektif dan efisien?	4	4	4

No	Pertanyaan	Skor dari responden		
		Admin	Vendor	Toko
Aspek Kemudahan Penggunaan				
6	Apakah aplikasi berbasis web ini mudah dipahami?	4	5	4
7	Apakah aplikasi berbasis web ini mudah digunakan?	5	5	4
8	Apakah aplikasi berbasis web ini dapat berjalan sesuai yang diharapkan?	4	5	5
9	Apakah aplikasi berbasis web ini mudah dimengerti oleh admin?	5	5	5
10	Apakah aplikasi berbasis web ini mudah dimengerti oleh vendor?	5	5	5
11	Apakah aplikasi berbasis web ini mudah dimengerti oleh toko?	5	5	5
Aspek Antarmuka Pengguna ( <i>User Interface</i> )				
12	Apakah aplikasi berbasis web ini memiliki tampilan yang jelas?	5	5	5
13	Apakah aplikasi berbasis web ini memiliki tampilan yang baik?	5	5	4
Jumlah skor/total skor		61/65	64/65	60/65
Persentase		93%	98%	92%
Total persentase		283%		

#### 4. Pengujian kompatibilitas aplikasi *browser*

Pengujian kompatibilitas aplikasi *browser* dilakukan oleh peneliti sendiri dengan menguji fungsionalitas dan tampilan antarmuka (*user interface*) aplikasi berbasis web yang dikembangkan dalam penelitian ini pada beberapa sampel aplikasi *browser* yang telah ditentukan. Hasil pengujian kompatibilitas aplikasi *browser* pada penelitian ini dapat dilihat pada Tabel 4.7.

Tabel 4.7. Hasil Pengujian Kompatibilitas Aplikasi *Browser*

No	Aspek	Google Chrome	Internet Explorer	Mozilla Firefox	Opera Browser	Safari Browser
1	Fungsi menu	Normal	Normal	Normal	Normal	Normal
2	Tampilan halaman	Normal	Berubah	Normal	Normal	Normal

### C. Pembahasan

Pembahasan hasil pengembangan dan pengujian perangkat lunak berupa aplikasi berbasis web yang dikembangkan pada penelitian ini antara lain diuraikan sebagai berikut:

#### 1. Pembahasan hasil implementasi aplikasi

Aplikasi ini berhasil dikembangkan dengan menggunakan metode *prototyping* yang terdiri dari beberapa tahap meliputi komunikasi (*communication*), perencanaan cepat (*quick plan*), perancangan model cepat (*modeling quick design*), konstruksi prototipe (*construction of prototype*), serta penyebaran, pengiriman, dan masukan (*deployment, delivery, and feedback*). Aplikasi ini juga berhasil diintegrasikan dengan data dari CMS Shopify milik perusahaan, dengan memanfaatkan API dari Shopify menggunakan cURL, sehingga tidak perlu dilakukan pengunggahan ulang data vendor dan data produk vendor.

## 2. Pembahasan hasil pengujian

### a. Pengujian *black-box*

Hasil pengujian *black-box* yang telah didapatkan dari ketiga responden kemudian dihitung dengan menggunakan perhitungan sebagai berikut:

#### 1) Responden 1

$$\text{Tercapai} = \frac{76}{76} \times 100\% = 100\% \quad (4.1)$$

$$\text{Gagal} = \frac{0}{76} \times 100\% = 0\% \quad (4.2)$$

#### 2) Responden 2

$$\text{Tercapai} = \frac{76}{76} \times 100\% = 100\% \quad (4.3)$$

$$\text{Gagal} = \frac{0}{76} \times 100\% = 0\% \quad (4.4)$$

#### 3) Responden 3

$$\text{Tercapai} = \frac{76}{76} \times 100\% = 100\% \quad (4.5)$$

$$\text{Gagal} = \frac{0}{76} \times 100\% = 0\% \quad (4.6)$$

Persentase pengujian masing-masing responden kemudian dirata-rata dan didapatkan hasil pengujian *black-box* sebagai berikut:

$$\text{Tercapai} = \frac{100\% + 100\% + 100\%}{3} = \frac{300\%}{3} = 100\% \quad (4.7)$$

$$\text{Gagal} = \frac{0\% + 0\% + 0\%}{3} = \frac{0\%}{3} = 0\% \quad (4.8)$$

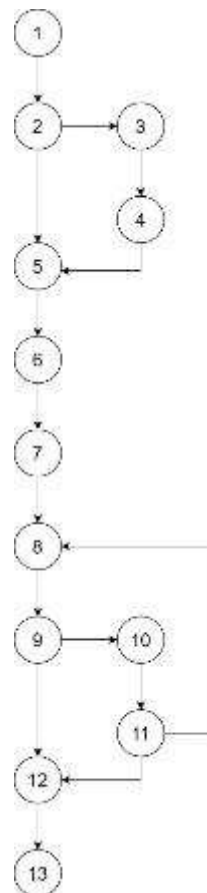
Berdasarkan hasil perhitungan rata-rata pengujian *black-box* dari tiga responden, didapatkan persentase ketercapaian fungsional perangkat lunak berupa aplikasi berbasis web yang dikembangkan adalah sebesar 100% berhasil dan 0% gagal dengan predikat layak atau berfungsi.

b. Pengujian *white-box*

Hasil pengujian *white-box* yang telah didapatkan kemudian diolah dengan menggunakan beberapa teknik sebagai berikut:

1) Pengujian berbasis alur (*basis path testing*)

Hasil dari pengujian berbasis alur adalah sebuah diagram alur (*flow diagram*) dari struktur kode (*script*) yang digambarkan dengan lingkaran. Hasil pengujian berbasis alur pada penelitian ini dapat dilihat pada Gambar 4.34.



Gambar 4.34. Pengujian Berbasis Alur pada Pengujian *White-box*

2) Kompleksitas siklomatis (*cyclomatic complexity*)

Berdasarkan hasil pengujian berbasis alur, maka diperoleh jumlah panah dan jumlah lingkaran untuk menghitung kompleksitas siklomatis dengan perhitungan sebagai berikut:

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 15 - 13 + 2 \\ &= 4 \end{aligned} \quad (4.9)$$

Keterangan:

$V(G)$  : Kompleksitas siklomatis untuk alur graf (G)

E : Jumlah panah (*edge*)

N : Jumlah lingkaran (*node*)

3) Jalur independen (*independent path*)

Berdasarkan hasil perhitungan kompleksitas siklomatis yang menghasilkan nilai 4, maka diperoleh 4 jalur independen sebagai berikut:

*Path 1*: 1, 2, 5, 6, 7, 8, 9, 12, 13

*Path 2*: 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13

*Path 3*: 1, 2, 5, 6, 7, 8, 9, 10, 11, 12, 13

*Path 4*: 1, 2, 5, 6, 7, 8, 9, 10, 11, 8, 9, 12, 13

4) Pengujian nilai (*value test*)

Berdasarkan jalur independen yang telah didapatkan, selanjutnya dilakukan pengujian pada proses masukan dan keluaran (*input and output*) dari sebuah jalur independen yang telah ditentukan. Hasil pengujian nilai pada penelitian ini dapat dilihat pada Tabel 4.8.

Tabel 4.8. Hasil Pengujian Nilai pada Pengujian *White-box*

No	<i>Path</i>	<i>Input</i>	<i>Output</i>	Ket
1	1, 2, 5, 6, 7, 8, 9, 12, 13	Toko mengirim <i>preorder</i>	<i>Preorder</i> selesai	Berhasil

No	Path	Input	Output	Ket
2	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13	Toko mengedit <i>preorder</i>	<i>Preorder</i> terkirim ke admin	Berhasil
3	1, 2, 5, 6, 7, 8, 9, 10, 11, 12, 13	Admin meminta pengiriman ulang produk	Vendor menerima permintaan admin	Berhasil
4	1, 2, 5, 6, 7, 8, 9, 10, 11, 8, 9, 12, 13	Produk ditolak admin	Vendor mengirim ulang produk	Berhasil

Hasil pengujian nilai untuk setiap jalur independen yang telah didapatkan kemudian dihitung dengan perhitungan sebagai berikut:

$$\mathbf{Tercapai} = \frac{4}{4} \times 100\% = 100\% \quad (4.10)$$

$$\mathbf{Gagal} = \frac{0}{4} \times 100\% = 0\% \quad (4.11)$$

Berdasarkan hasil pengujian nilai pada pengujian *white-box*, didapatkan persentase sebesar 100% tercapai dan 0% gagal.

c. *User Acceptance Testing* (UAT)

Hasil UAT yang telah didapatkan kemudian dihitung dengan menggunakan perhitungan sebagai berikut:

$$\mathbf{Hasil\ UAT} = \frac{\text{Total persentase}}{\text{Jumlah responden}} = \frac{283\%}{3} = 94\% \quad (4.12)$$

Berdasarkan hasil perhitungan UAT, rata-rata persentase yang didapatkan dari 3 responden yang terdiri dari 1 pimpinan perusahaan sebagai admin, 1 perwakilan dari pihak vendor, dan 1 perwakilan dari pihak toko berdasarkan aspek kemanfaatan, aspek kemudahan



penggunaan, dan aspek antarmuka pengguna (*user interface*) adalah sebesar 94% dengan predikat sudah memenuhi harapan pengguna dan bekerja seperti yang telah direncanakan.

d. Pengujian kompatibilitas aplikasi *browser*

Berdasarkan hasil pengujian kompatibilitas aplikasi *browser* yang telah dilakukan, aplikasi berbasis web yang dikembangkan pada penelitian ini secara fungsional dapat dijalankan pada semua tipe aplikasi *browser* yang telah ditentukan antara lain Google Chrome, Internet Explorer, Mozilla Firefox, Opera Browser, dan Safari Browser. Sementara itu dari aspek tampilan antarmuka pengguna (*user interface*), terdapat satu aplikasi *browser* yang kurang kompatibel dengan tampilan aplikasi berbasis web yang dikembangkan pada penelitian ini yaitu Internet Explorer. Terdapat salah satu fitur JavaScript yaitu *alert* yang tidak dapat ditampilkan dengan sempurna di aplikasi *browser* tersebut. Namun hal tersebut hanya bagian kecil yang tidak terlalu mempengaruhi fungsionalitas aplikasi berbasis web yang dikembangkan pada penelitian ini.

## **BAB V**

### **PENUTUP**

#### **A. Simpulan**

Berdasarkan penelitian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi pengelolaan distribusi produk UKM berbasis web dengan integrasi CMS Shopify di Netasia Singapore berhasil dikembangkan dengan menggunakan metode *prototyping* yang terdiri dari beberapa tahap meliputi komunikasi (*communication*), perencanaan cepat (*quick plan*), perancangan model cepat (*modeling quick design*), konstruksi prototipe (*construction of prototype*), serta penyebaran, pengiriman, dan masukan (*deployment, delivery, and feedback*).
2. Aplikasi pengelolaan distribusi produk UKM berbasis web di Netasia Singapore berhasil diintegrasikan dengan data dari CMS Shopify milik perusahaan, dengan memanfaatkan API dari Shopify menggunakan cURL pada aplikasi yang dikembangkan sehingga tidak perlu dilakukan pengunggahan ulang data vendor dan data produk vendor.
3. Berdasarkan hasil pengujian *black-box* yang telah dilakukan, aplikasi berbasis web yang dikembangkan pada penelitian ini layak digunakan dengan persentase sebesar 100% tercapai dan 0% gagal.
4. Berdasarkan hasil pengujian *white-box* yang telah dilakukan, aplikasi berbasis web yang dikembangkan pada penelitian ini memenuhi kriteria rekayasa perangkat lunak dengan kompleksitas siklomatis (*cyclomatic complexity*) di bawah 10 yaitu sebesar 4 serta persentase hasil pengujian nilai (*value test*) sebesar 100% tercapai dan 0% gagal.
5. Berdasarkan hasil *User Acceptance Testing* (UAT) yang dilakukan bersama tiga responden berdasarkan aspek kemanfaatan, aspek kemudahan penggunaan, dan aspek antarmuka pengguna (*user interface*), didapatkan persentase rata-rata sebesar 94%, sehingga aplikasi berbasis

web yang dikembangkan pada penelitian ini sudah memenuhi harapan pengguna dan bekerja seperti yang direncanakan.

6. Berdasarkan hasil pengujian kompatibilitas aplikasi *browser* yang telah dilakukan, aplikasi berbasis web yang dikembangkan pada penelitian ini secara fungsional dapat dijalankan pada semua tipe aplikasi *browser* yang telah ditentukan antara lain Google Chrome, Internet Explorer, Mozilla Firefox, Opera Browser, dan Safari Browser, sedangkan dari aspek tampilan antarmuka pengguna (*user interface*) kurang kompatibel dengan aplikasi *browser* Internet Explorer terhadap fitur *alert*, namun tidak terlalu berpengaruh pada fungsionalitas aplikasi berbasis web yang dikembangkan pada penelitian ini.

## **B. Saran**

Berdasarkan penelitian yang telah dilakukan, saran yang diajukan adalah sebagai berikut:

1. Sebelum aplikasi pengelolaan distribusi produk UKM berbasis web dengan integrasi CMS Shopify di Netasia Singapore diimplementasikan dan digunakan secara fungsional oleh perusahaan, sebaiknya dilakukan pelatihan kepada pengguna yang terdiri dari staf perusahaan sebagai admin, pihak vendor, dan pihak toko agar dapat digunakan dengan mudah dan lancar.
2. Aplikasi berbasis web yang dikembangkan pada penelitian ini masih dapat dikembangkan dan diperbarui lebih optimal lagi seiring dengan perkembangan spesifikasi dan kebutuhan perusahaan.
3. Aplikasi berbasis web yang dikembangkan pada penelitian ini masih dapat diujikan pada aplikasi *browser* selain yang digunakan dalam penelitian ini.
4. Aplikasi berbasis web yang dikembangkan pada penelitian ini juga dapat dikembangkan untuk pengelolaan distribusi produk di perusahaan lain dengan prosedur yang berbeda.

## DAFTAR PUSTAKA

- [1] A. Hendriyanto and D. Edi, “Integrasi Data Tokopedia dengan Content Management Systems Point of Sales,” *J. Strateg.*, vol. 3, no. November, pp. 311–319, 2021.
- [2] Sulton and R. Setiawan, “Rancang Bangun Aplikasi Computer Laboratory Resource Inventory System Berbasis Web pada Sekolah Menengah Kejuruan Plus Pontren Yabafa,” *J. Algoritm. Sekol. Tinggi Teknol. Garut*, pp. 50–57, 2014.
- [3] Feradhita NKD, “Web App VS Website – Apa Saja Perbedaan Keduanya?,” *Logique*, 2019. <https://www.logique.co.id/blog/2019/05/10/web-app-> (accessed Nov. 27, 2021).
- [4] A. Zein, E. S. Eriana, S. Farizy, and G. N. Persada, “Pembuatan Website CMS (Content Management System) Pada SMK Muhammadiyah Parung Bogor,” *J. Ilmu Komput.*, vol. IV, no. 1, pp. 70–75, 2021.
- [5] Y. Maulanadireja, “Optimalisasi Distribusi Barang PT. XYZ Menggunakan Hill Climbing Berbasis Android,” *Tesis*, 2020.
- [6] R. Yusuf, H. Kusniyati, and R. E. Dewi, “Aggregator Otomatis Pencari Produk dengan Pemberitahuan Melalui Surel Menggunakan Fungsi cURL,” *J. Tek. Inform.*, vol. 8, no. 1, pp. 48–56, 2015, doi: 10.15408/jti.v8i1.1936.
- [7] R. S. Wijaya, “Aplikasi Fairuz Wedding Organizer Berbasis Web Application,” *Rhesa Setya Wijaya*, vol. 3, no. 3, pp. 1576–1582, 2017.
- [8] A. Wahana, “Rancang Bangun Marketplace Produk Kewirausahaan Mahasiswa UPY Berbasis Content Management System,” *J. Din. Inform.*, vol. 7, no. 1, pp. 73–81, 2018.
- [9] K. Howe-Patterson, “An Analysis of a Canadian E-commerce Platform’s Marketing Strategy and Activities in an International Market,” *Thesis*, 2020.
- [10] B. E. W. Asrul, “Sistem Temu Kembali untuk Data Harga Murah Barang Toko Online Menggunakan Web Service,” *J. IT STMIK Handayani*, vol. 14, pp. 13–19, 2014.

- [11] P. Irawan, D. A. P. Prasetya, and P. Sokibi, "Rancang Bangun Sistem Pengarsipan Surat Kedinasan Berbasis Web Menggunakan Framework Codeigniter," *J. Manaj. Inform. dan Sist. Inf.*, vol. 3, no. 2, pp. 157–165, 2020.
- [12] S. Ali and A. Ambarita, "Information System of Inventory Goods Web-Based on The State Prosecutor Ternate," *Indones. J. Inf. Syst. Sist.*, vol. 1, no. April 2016, pp. 31–38, 2016.
- [13] S. Rosa, "Politeknik Negeri Sriwijaya 4," *Pembangkitan Energi List.*, vol. 7, no. 1, pp. 8–31, 2016.
- [14] M. R. R. Widodo, M. R. Zainuddin, and L. S. Nusantara, "Sistem Informasi dan Pengolahan Data Kursus Mobil Berbasis Web dengan SMS," *J. Inform. Merdeka Pasuruan*, vol. 1, no. 3, pp. 85–104, 2016.
- [15] M. A. Rosid, "Implementasi JSON untuk Minimasi Penggunaan Jumlah Kolom Suatu Tabel Pada Database PostgreSQL," *JOINCS (Journal Informatics, Network, Comput. Sci.)*, vol. 1, no. 1, p. 33, 2017.
- [16] H. F. Siregar and N. Sari, "Rancang Bangun Aplikasi Simpan Pinjam Uang Mahasiswa Fakultas Teknik Universitas Asahan Berbasis Web," *J. Teknol. Inf.*, vol. 2, no. 1, p. 53, 2018, doi: 10.36294/jurti.v2i1.409.
- [17] T. Hidayat and M. Muttaqin, "Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis," *J. Tek. Inform. UNIS JUTIS*, vol. 6, no. 1, pp. 2252–5351, 2018.
- [18] R. Parluka, T. A. Nisaa, S. M. Ningrum, and B. A. Haque, "Studi Literatur Kekurangan dan Kelebihan Pengujian Black Box," *Teknomatika*, vol. 10, no. 02, pp. 131–140, 2020.
- [19] U. Hanifah, R. Alit, and S. Sugiarto, "Penggunaan Metode Black Box Pada Pengujian Sistem Informasi Surat Keluar Masuk," *SCAN - J. Teknol. Inf. dan Komun.*, vol. 11, no. 2, pp. 33–40, 2016.
- [20] M. F. Londjo, "Seri Sains dan Teknologi Implementasi White Box Testing dengan Teknik Basis Path Kata Kunci : White Box , Basis Path , Form Login Seri Sains dan Teknologi P-ISSN 2477-3891 E-ISSN 2615-4765,"

- vol. 7, no. 2, pp. 35–40, 2021.
- [21] Handy and J. Susilo, “Aplikasi Pengujian White-Box Ibi Online Judge,” *J. Inform. dan Bisnis*, vol. 3, pp. 56–68, 2014.
- [22] E. Suprpto, “User Acceptance Testing (UAT) Refreshment PBX Outlet Site BNI Kanwil Padang,” *J. Civronlit Unbari*, vol. 6, no. 2, p. 54, 2021, doi: 10.33087/civronlit.v6i2.85.
- [23] Y. Lestari, Sunardi, and A. Fadlil, “Analisis Pengembangan Sistem Penerimaan Peserta Didik Baru (PPDB) Menggunakan Metode R&D,” *Semin. Nas. Inform. 2018 (semnasIF 2018)*, vol. 1, no. 1, pp. 160–165, 2018.
- [24] S. Haryati, “Research and Development (R&D) Sebagai Salah Satu Model Penelitian dalam Bidang Pendidikan,” *Res. Dev. Sebagai Salah Satu Model Penelit. Dalam Bid. Pendidik.*, vol. 37, no. 1, pp. 11–26, 2012.
- [25] Sugiyono, *Metode Penelitian Pendidikan “Pendekatan Kuantitatif, Kualitatif, dan R&D.”* Bandung: Alfabeta, 2016.
- [26] R. C. Noor Santi, “Perancangan Interaksi Pengguna (User Interaction Design) Menggunakan Metode Prototyping,” *J. Tek. Inform.*, vol. 9, no. 2, pp. 108–113, 2018, doi: 10.15408/jti.v9i2.5599.
- [27] A. A. dk. Pradipta, “Pengembangan Web E-Commerce Bojana Sari Menggunakan Metode Prototype,” *J. Tugas Akhir / Fak. Rekayasa Ind.*, vol. 3, no. 1, pp. 341–347, 2019.
- [28] A. Abdullah and R. Kurniawan, “Perancangan Sistem Informasi Akuntansi Desa Wisata Pentingsari Menggunakan Metode Prototyping,” *Automata*, p. 7, 2021.
- [29] B. A. Kusuma, Andriani, and S. H. Fitriasih, “Aplikasi penjualan barang secara multiuser pada distro thinker matesih karanganyar,” *J. TiKomSiN*, vol. 5, pp. 39–44, 2017.

## LAMPIRAN

### Lampiran 1. Surat Permohonan Izin Penelitian



**UNIVERSITAS PGRI SEMARANG**  
**FAKULTAS TEKNIK DAN INFORMATIKA**

Kampus : Jalan Sidodadi Timur Nomor 24 Dr. Cipto, Semarang - Indonesia 50125

Telp. (024) 8452230, Faks. (024) 8448217, E-mail : [fti@upgris.ac.id](mailto:fti@upgris.ac.id). Website : <http://fti.upgris.ac.id>

Nomor : 225 /AM/FTI/III/2022

7 Maret 2022

Lamp. : --

Hal : Permohonan izin penelitian

Kepada Yth.  
Pimpinan Netasia Singapura  
Clemenceau Avenue #06-08 Singapura  
SINGAPURA

Kami beritahukan dengan hormat, bahwa mahasiswa kami:

NO.	N P M	NAMA MAHASISWA	PROGRAM STUDI
1.	18670028	ARSHA RAULNADI TRIKUSUMA	Informatika
2.			
3.			
4.			
5.			

Akan mengadakan penelitian dengan judul:

**Aplikasi Pengelolaan Distribusi Produk Usaha Kecil Menengah Berbasis Web dengan Integrasi Content Management System Shopify di Netasia Singapura**

Sehubungan dengan hal tersebut kami mohon perkenan Bapak/Ibu memberikan izin bagi mahasiswa tersebut untuk melakukan penelitian.

Atas perkenan dan kerjasama yang baik, kami ucapkan terima kasih.



Dr. SLAMET SUPRIYADI, M.Env.St.  
NIP. 195912281986031003

## Lampiran 2. Lembar Pembimbingan Skripsi Pembimbing 1



**UNIVERSITAS PGRI SEMARANG**

**FAKULTAS TEKNIK DAN INFORMATIKA**

Kampus : Jalan Siddadi Timur Nomor 24 Dr. Cipto, Semarang – Indonesia 50125

Telp. (024) 8316377, Faks. (024) 8448217, E-mail : [upgrismg@gmail.com](mailto:upgrismg@gmail.com), Homepage : [www.upgrismg.ac.id](http://www.upgrismg.ac.id)

### LEMBAR PEMBIMBINGAN SKRIPSI

Nama Mahasiswa : ARSHA RAULWADI TRIKUSUMA  
 N P M : 18670028  
 Program Studi : INFORMATIKA  
 Judul Skripsi : APLIKASI PENGELOLAAN DISTRIBUSI PRODUK  
 USAHA KECIL MENENGAH BERBASIS WEB  
 DENGAN INTEGRASI CONTENT MANAGEMENT  
 SYSTEM SHOPIFY DI NETASIA SINGAPORE  
 Dosen Pembimbing I : MEGA NOVITA, S.Si, M.Si, M.Nat.Sc., Ph.D.  
 Dosen Pembimbing II : NOORA ACHUN NADA, S.T., M.Eng.

No.	Hari, tanggal	Uraian Bimbingan	Paraf
①	Jumat, 14 Januari 2022	Bimbingan judul dan proposal, judul ACC pembimbing I.	
②	Jumat, 11-03-2022	- BAB IV - Diagram $\Rightarrow$ UML ditaruh di metode BAB III - Form pengujian black-box dan UAT	
③	Selasa, 15-03-2022	BAB III Hasil $\rightarrow$ Integrasi CMS Shopify dengan API	

Dosen Pembimbing I,

MEGA NOVITA  
 NIK/NPP 158801493

Mahasiswa

ARSHA RAULWADI TRIKUSUMA  
 NPM 18670028





**UNIVERSITAS PGRI SEMARANG**

**FAKULTAS TEKNIK DAN INFORMATIKA**

Kampus : Jalan Sidodadi Timur Nomor 24 Dr. Cipto, Semarang – Indonesia 50125

Telp. (024) 8316377, Faks. (024) 8448217, E-mail : upgrisng@gmail.com Homepage : www.upgrisng.ac.id

**LEMBAR PEMBIMBINGAN SKRIPSI**

Nama Mahasiswa : ARSHA RAULNADI TRIKUSUMA  
 N P M : 18670028  
 Program Studi : INFORMATIKA  
 Judul Skripsi : APLIKASI PENGELOLAAN DISTRIBUSI PRODUK USAHA KECIL MENENGAH BERBASIS WEB DENGAN INTEGRASI CONTENT MANAGEMENT SYSTEM SHOPIFY DI NETASIA SINGAPORE.  
 Dosen Pembimbing I : MEGA NOVITA, S.Si., M.Si., M.Nae.Sc., Ph.D.  
 Dosen Pembimbing II : NOORA SOTRUM NADA, S.T., M.Eng.

No.	Hari, tanggal	Uraian Bimbingan	Paraf
(A)	Jumat, 18-03-2022	- Pengujian white-box - Kerangka bertitik - BAB IV → Alur pengembangan aplikasi	
(S)	Rabu, 23 Maret 2022	ACC Sidang	

Dosen Pembimbing I,

MEGA NOVITA  
 NIP/NPP 188001993

Mahasiswa

ARSHA RAULNADI TRIKUSUMA  
 NPM 18670028

### Lampiran 3. Lembar Pembimbingan Skripsi Pembimbing 2



**UNIVERSITAS PGRI SEMARANG**

**FAKULTAS TEKNIK DAN INFORMATIKA**

Kampus : Jalan Sidoarjo Timur Nomor 24 Dr. Cipto, Semarang – Indonesia 50125

Telp. (024) 8316377, Faks. (024) 8448217, E-mail : [ungrismg@gmail.com](mailto:ungrismg@gmail.com), Homepage : [www.upgrisng.ac.id](http://www.upgrisng.ac.id)

#### LEMBAR PEMBIMBINGAN SKRIPSI

Nama Mahasiswa : ARSHA RAULNADI TRIKUSUMA  
 N P M : 18670028  
 Program Studi : INFORMATIKA  
 Judul Skripsi : APLIKASI PENEBLOLAAN DISTRIBUSI PRODUK USAHA KECIL MENENGAH BERBASIS WEB DENGAN INTEGRASI CONTENT MANAGEMENT SYSTEM SHOPIFY DI NETASIA SINGAPORE  
 Dosen Pembimbing I : MEGA NOVITA, S.Si, M.Si., M.Nat.Sc., Ph.D.  
 Dosen Pembimbing II : NOORA GOTRIAN NADA, S.T., M.Eng.

No.	Hari, tanggal	Uraian Bimbingan	Paraf
1.	17 Jan 2021	① Bimbingan Judul. ② Bab. 1-3. tambahan metode prototyping	<i>[Signature]</i>
2.	15 Maret 2021	① Tambahkan white box. ② Perbaikan → page listen to customer → hub dgn user di perjalan / dibatas	<i>[Signature]</i>

Dosen Pembimbing II,

*[Signature]*

NIP/NPP 158201985

NIDN 0626028201

Mahasiswa

*[Signature]*

ARSHA RAULNADI TRIKUSUMA

NPM 18670028



**UNIVERSITAS PGRI SEMARANG**

FAKULTAS TEKNIK DAN INFORMATIKA

Kampus : Jalan Sidodadi Timur Nomor 24 Dr. Cipto, Semarang – Indonesia 50125

Telp. (024) 8316377, Faks. (024) 8448217, E-mail : upgrismg@gmail.com, Homepage : www.upgrismg.ac.id

**LEMBAR PEMBIMBINGAN SKRIPSI**

Nama Mahasiswa : ARSHA RAULIADI TRIKUSUMA  
 N P M : 18670028  
 Program Studi : INFORMATIKA  
 Judul Skripsi : APLIKASI PENGELOLAAN DISTRIBUSI PRODUK USAHA KECIL MENENGAH BERBASIS WEB DENGAN INTEGRASI CONTENT MANAGEMENT SYSTEM SHOPIFY DI NETASIA SINGAPORE  
 Dosen Pembimbing I : MEGA NOVITA, S.Si., M.Si., M.Nat.Sc., Ph.D.  
 Dosen Pembimbing II : NOORA BOTRAN NADA, S.T., M.Eng.

No.	Hari, tanggal	Uraian Bimbingan	Paraf
3.	16 Maret 2022	① BAB III → desain penelitian diseleksi ② pengujian white-box Arlegkap	f
4.	18 Maret 2022	① Seleksi sequence diagram ② seleksi desain UI ③ BAB IV Penulisan implementasi	f
5.	23 Maret 2022	acc. Sidang.	f

Dosen Pembimbing II.

NIP/NPP 158201485

Mahasiswa

ARSHA RAULIADI TRIKUSUMA  
 NPM 18670028

### Lampiran 4. Formulir Pengujian *Black-box* Responden 1

**FORM KUESIONER PENGUJIAN *BLACK-BOX***  
**APLIKASI PENGELOLAAN DISTRIBUSI PRODUK USAHA KECIL MENENGAH**  
**BERBASIS WEB DENGAN INTEGRASI *CONTENT MANAGEMENT SYSTEM***  
**SHOPIFY DI NETASIA SINGAPORE**

Tanggal pengujian : 16/3/2022  
 Nama Responden : Febrian M.D., M.Kom  
 Jabatan/status : Dosen

Mohon perhatikan pertanyaan dengan seksama dan harap mengisi form kuesioner berikut ini dengan jujur dan sesuai dengan kondisi yang terjadi. Beri tanda centang (✓) pada kolom yang sudah disediakan apabila *test case* berhasil dijalankan.

1) Formulir pengujian halaman login

<i>Test Case</i>	Hasil yang diharapkan	Hasil yang didapatkan	Berhasil
<b>1. Formulir Login</b>			
<i>Username</i> dan <i>password</i> diisi dengan data yang benar dengan level admin	Pengguna dapat masuk ke halaman utama admin (dasbor)	Sistem akan menampilkan halaman utama admin (dasbor)	✓
<i>Username</i> dan <i>password</i> diisi dengan data yang benar dengan level vendor	Pengguna dapat masuk ke halaman utama vendor (dasbor)	Sistem akan menampilkan halaman utama vendor (dasbor)	✓
<i>Username</i> dan <i>password</i> diisi dengan data yang benar dengan level toko	Pengguna dapat masuk ke halaman utama toko (dasbor)	Sistem akan menampilkan halaman utama toko (dasbor)	✓
<i>Username</i> dan <i>password</i> diisi dengan data yang salah	Pengguna tidak dapat masuk ke halaman utama	Sistem kembali ke halaman login dan menampilkan pesan error	✓

## 2) Formulir pengujian halaman admin

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
<b>1. Menu dasbor (<i>dashboard</i>)</b>			
Memilih menu dasbor	Masuk ke menu dasbor admin	Menampilkan halaman dasbor admin	✓
<b>2. Menu daftar vendor (<i>vendor partner</i>)</b>			
Memilih menu <i>vendor partner</i>	Masuk ke menu <i>vendor partner</i>	Menampilkan halaman <i>vendor partner</i>	✓
Klik tombol detail	Masuk ke menu detail vendor untuk melihat daftar produk vendor	Menampilkan halaman detail vendor berisi daftar produk vendor	✓
Klik tombol <i>add account</i>	Muncul <i>modal form</i> untuk menambahkan akun vendor baru	Menampilkan <i>modal form</i> untuk menambahkan akun vendor baru	✓
Mengisi form penambahan akun vendor baru dan klik tombol <i>register</i>	Data akun vendor baru berhasil tersimpan	Menampilkan daftar vendor di halaman <i>vendor partner</i> dan tombol <i>add account</i> berubah menjadi tombol <i>edit account</i>	✓
Klik tombol <i>edit account</i>	Muncul <i>modal form</i> untuk mengedit akun vendor	Menampilkan <i>modal form</i> untuk mengedit akun vendor	✓
Mengisi form edit akun vendor dan klik tombol <i>update</i>	Data akun vendor berhasil diperbarui	Menampilkan daftar vendor di halaman <i>vendor partner</i>	✓
<b>3. Menu daftar toko (<i>store partner</i>)</b>			
Memilih menu <i>store partner</i>	Masuk ke menu <i>store partner</i>	Menampilkan halaman <i>store partner</i>	✓
Klik tombol <i>add new store</i>	Muncul <i>modal form</i> untuk menambahkan akun toko baru	Menampilkan <i>modal form</i> untuk menambahkan akun toko baru	✓

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
Mengisi form penambahan akun toko baru dan klik tombol <i>register</i>	Data akun toko baru berhasil tersimpan	Menampilkan daftar toko di halaman <i>store partner</i>	✓
Klik tombol detail	Masuk ke menu detail toko untuk melihat daftar <i>preorder</i> toko	Menampilkan halaman detail toko berisi daftar <i>preorder</i> toko	✓
Klik tombol edit	Muncul <i>modal form</i> untuk mengedit akun toko	Menampilkan <i>modal form</i> untuk mengedit akun toko	✓
Mengisi form edit akun toko dan klik tombol <i>update</i>	Data akun toko berhasil diperbarui	Menampilkan daftar toko di halaman <i>store partner</i>	✓
Klik tombol hapus	Data akun toko berhasil dihapus	Menampilkan daftar toko di halaman <i>store partner</i>	✓
<b>4. Menu prapemesanan (<i>preorders</i>)</b>			
Memilih menu <i>preorders</i>	Masuk ke menu <i>preorders</i>	Menampilkan halaman <i>preorders</i>	✓
Klik tab <i>preorder from store</i>	Melihat daftar permintaan <i>preorder</i> dari toko	Menampilkan daftar permintaan <i>preorder</i> dari toko	✓
Klik tab <i>preorder to vendor</i>	Melihat daftar permintaan <i>preorder</i> ke vendor	Menampilkan daftar permintaan <i>preorder</i> ke vendor	✓
Klik tombol <i>process</i> di tab <i>preorder from store</i>	Memproses permintaan <i>preorder</i> dari toko	Menampilkan halaman <i>process preorder</i>	✓
Klik tombol <i>add to demand</i> di halaman <i>process preorder</i>	Memasukkan pesanan toko ke daftar permintaan <i>preorder</i> ke vendor	Menampilkan halaman <i>preorders</i>	✓
Klik tombol <i>process to vendor</i> di tab <i>preorder to vendor</i>	Memproses permintaan <i>preorder</i> ke vendor	Menampilkan daftar produk yang dipesan di halaman <i>process preorder to vendor</i>	✓

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
Klik tombol <i>submit preorder to vendor</i>	<i>Checkout</i> produk dan mengirimkan <i>preorder</i> kepada vendor	Menampilkan tab <i>preorder to vendor</i> di halaman <i>preorders</i>	✓
<b>5. Menu riwayat prapemesanan (<i>history</i>)</b>			
Memilih menu <i>history</i>	Masuk ke menu <i>history</i>	Menampilkan halaman <i>history</i>	✓
Klik tab <i>preorder from store</i>	Melihat daftar riwayat <i>preorder</i> dari toko	Menampilkan daftar riwayat <i>preorder</i> dari toko	✓
Klik tab <i>preorder to vendor</i>	Melihat daftar riwayat <i>preorder</i> ke vendor	Menampilkan daftar riwayat <i>preorder</i> ke vendor	✓
Klik tombol detail di tab <i>preorder from store</i>	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan toko	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan toko	✓
Klik tombol detail di tab <i>preorder to vendor</i>	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan ke vendor	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	✓
Jika status di tab <i>preorder to vendor</i> " <i>delivered</i> " klik tombol <i>check</i>	Memeriksa apakah produk yang dikirimkan sesuai dengan yang diterima	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	✓
Klik tombol <i>request product resend</i> di halaman detail <i>preorder</i>	Meminta pengiriman ulang produk karena ada yang tidak sesuai	Muncul <i>modal form</i> keterangan permintaan pengiriman ulang	✓
Mengisi form keterangan permintaan pengiriman ulang dan klik <i>request</i>	Permintaan pengiriman ulang beserta keterangan berhasil dikirim ke vendor	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>request product resend</i>	✓

<i>Test Case</i>	<b>Hasil yang diharapkan</b>	<b>Hasil yang didapatkan</b>	<b>Berhasil</b>
Klik tombol <i>receive product</i> di halaman detail <i>preorder</i>	Menerima produk yang telah terkirim dari vendor ke admin	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>received</i>	✓
Jika status di tab <i>preorder from store</i> "ready" klik tombol detail	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan ke vendor	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	✓
Klik tombol <i>finish preorder</i> di halaman detail <i>preorder</i>	Produk telah didistribusikan kepada toko	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>finished</i>	✓
<b>6. Menu akun admin (<i>admin accounts</i>)</b>			
Memilih menu <i>admin accounts</i>	Masuk ke menu <i>admin accounts</i>	Menampilkan halaman <i>admin accounts</i>	✓
Klik tombol <i>add new admin</i>	Muncul <i>modal form</i> untuk menambahkan akun admin baru	Menampilkan <i>modal form</i> untuk menambahkan akun admin baru	✓
Mengisi form penambahan akun admin baru dan klik tombol <i>register</i>	Data akun admin baru berhasil tersimpan	Menampilkan daftar admin di halaman <i>admin account</i>	✓
Klik tombol edit	Muncul <i>modal form</i> untuk mengedit akun admin	Menampilkan <i>modal form</i> untuk mengedit akun admin	✓
Mengisi form edit akun admin dan klik tombol <i>update</i>	Data akun admin berhasil diperbarui	Menampilkan daftar admin di halaman <i>store partner</i>	✓
Klik tombol hapus	Data akun toko berhasil dihapus	Menampilkan daftar admin di halaman <i>admin account</i>	✓



<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
<b>7. Menu Pengaturan Akun</b>			
Klik tombol <i>settings</i>	Masuk ke menu <i>settings</i>	Menampilkan <i>settings</i>	✓
Mengisi form pengaturan akun dan klik tombol <i>update</i>	Data akun berhasil diperbarui	Menampilkan halaman <i>login</i>	✓
<b>8. Keluar dari Halaman Admin</b>			
Klik tombol <i>logout</i>	Kembali ke halaman <i>login</i>	Menampilkan halaman <i>login</i>	✓

## 3) Formulir pengujian halaman vendor

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
<b>1. Menu dasbor (<i>dashboard</i>)</b>			
Memilih menu dasbor	Masuk ke menu dasbor vendor	Menampilkan halaman dasbor vendor	✓
<b>2. Menu daftar produk (<i>products</i>)</b>			
Memilih menu <i>products</i>	Masuk ke menu <i>products</i>	Menampilkan halaman <i>products</i> dan daftar produk	✓
<b>3. Menu prapemesanan (<i>preorders</i>)</b>			
Memilih menu <i>preorders</i>	Masuk ke menu <i>preorders</i>	Menampilkan halaman <i>preorders</i>	✓
Klik tombol <i>process</i> di salah satu pesanan	Memproses permintaan <i>preorder</i> dari admin	Menampilkan halaman <i>process preorder</i>	✓
Klik tombol <i>accept preorder</i> di halaman <i>process preorder</i>	Menerima permintaan <i>preorder</i> dari admin	Menampilkan halaman <i>preorders</i>	✓
<b>4. Menu riwayat prapemesanan (<i>history</i>)</b>			
Memilih menu <i>history</i>	Masuk ke menu <i>history</i>	Menampilkan halaman <i>history</i>	✓
Klik tombol detail	Masuk ke menu detail <i>preorder</i> untuk melihat	Menampilkan halaman detail <i>preorder</i> berisi	✓

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
	daftar produk yang dipesan	daftar produk yang dipesan	✓
Jika status "accepted" klik tombol <i>deliver</i>	Masuk ke menu detail <i>preorder</i> untuk memperbarui status	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	✓
Klik tombol <i>deliver product</i> di halaman detail <i>preorder</i>	Memperbarui status <i>preorder</i> bahwa produk sedang dikirim	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>delivered</i>	✓
Jika status "request <i>product resend</i> " klik tombol <i>respond</i>	Masuk ke menu detail <i>preorder</i> untuk merespon permintaan	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor dan keterangan	✓
Klik tombol <i>contact admin</i> di halaman detail <i>preorder</i>	Menghubungi kontak admin	Menuju ke nomor telepon admin	✓
Klik tombol <i>resend product</i> di halaman detail <i>preorder</i>	Memperbarui status <i>preorder</i> bahwa produk sedang dikirim ulang	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>delivered</i>	✓
<b>5. Menu Pengaturan Akun</b>			
Klik tombol <i>settings</i>	Masuk ke menu <i>settings</i>	Menampilkan menu <i>settings</i>	✓
Mengisi form pengaturan akun dan klik tombol <i>update</i>	Data akun berhasil diperbarui	Menampilkan halaman <i>login</i>	✓
<b>6. Keluar dari Halaman Admin</b>			
Klik tombol <i>logout</i>	Kembali ke halaman <i>login</i>	Menampilkan halaman <i>login</i>	✓

## 4) Formulir pengujian halaman toko

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
<b>1. Menu dashbor (<i>dashboard</i>)</b>			
Memilih menu dashbor	Masuk ke menu dashbor toko	Menampilkan halaman dashbor toko	✓
<b>2. Menu prapemesanan (<i>preorders</i>)</b>			
Memilih menu <i>preorders</i>	Masuk ke menu <i>preorders</i>	Menampilkan daftar vendor di halaman <i>preorders</i>	✓
Klik tombol <i>preorder product</i> di salah satu vendor	Masuk ke menu detail vendor untuk melihat daftar produk vendor	Menampilkan halaman detail vendor berisi daftar produk vendor	✓
Klik tombol <i>preorder</i> pada salah satu produk	Masuk ke menu detail produk untuk memesan produk	Menampilkan halaman detail produk	✓
Memilih opsi produk dan klik tombol <i>add to cart</i>	Memilih produk sesuai opsi dan memasukkan produk ke dalam keranjang	Menampilkan halaman detail produk	✓
<b>3. Menu keranjang (<i>cart</i>)</b>			
Memilih menu <i>cart</i>	Masuk ke menu <i>cart</i>	Menampilkan halaman <i>cart</i>	✓
Klik tombol <i>submit preorder</i>	<i>Checkout</i> produk dan mengirimkan <i>preorder</i> kepada admin	Menampilkan halaman <i>history</i>	✓
<b>4. Menu riwayat prapemesanan (<i>history</i>)</b>			
Memilih menu <i>history</i>	Masuk ke menu <i>history</i>	Menampilkan halaman <i>history</i>	✓
Klik tombol detail	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan	Menampilkan halaman detail vendor berisi daftar produk yang dipesan	✓

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
Klik tombol <i>cancel preorder</i>	<i>Preorder</i> dibatalkan	Menampilkan halaman <i>history</i> dan status menjadi " <i>cancelled</i> "	✓
Jika status " <i>cancelled</i> " klik tombol <i>edit preorder</i>	Mengubah data dalam <i>preorder</i>	Menampilkan halaman <i>cart</i> berisi data <i>preorder</i> yang diedit	✓
Memperbarui data <i>preorder</i> dan klik tombol <i>submit preorder</i> di halaman <i>cart</i>	Data <i>preorder</i> berhasil diperbarui	Menampilkan halaman <i>history</i>	✓
Jika status " <i>cancelled</i> " klik tombol <i>delete preorder</i>	Data <i>preorder</i> berhasil dihapus	Menampilkan halaman <i>history</i>	✓
<b>5. Menu Pengaturan Akun</b>			
Klik tombol <i>settings</i>	Masuk ke menu <i>settings</i>	Menampilkan menu <i>settings</i>	✓
Mengisi form pengaturan akun dan klik tombol <i>update</i>	Data akun berhasil diperbarui	Menampilkan halaman <i>login</i>	✓
<b>6. Keluar dari Halaman Admin</b>			
Klik tombol <i>logout</i>	Kembali ke halaman <i>login</i>	Menampilkan halaman <i>login</i>	✓

Saran dan masukan:

Responden:

  
 (.....) Febrina

## Lampiran 5. Formulir Pengujian *Black-box* Responden 2

**FORM KUESIONER PENGUJIAN *BLACK-BOX***  
**APLIKASI PENGELOLAAN DISTRIBUSI PRODUK USAHA KECIL MENENGAH**  
**BERBASIS WEB DENGAN INTEGRASI *CONTENT MANAGEMENT SYSTEM***  
**SHOPIFY DI NETASIA SINGAPORE**

Tanggal pengujian: 16 - 3 - 2022  
 Nama Responden: Rahmat Dobi Wahyungah, M.Kom.  
 Jabatan/status: Dosen Informatika UPGHS

Mohon perhatikan pertanyaan dengan seksama dan harap mengisi form kuesioner berikut ini dengan jujur dan sesuai dengan kondisi yang terjadi. Beri tanda centang (✓) pada kolom yang sudah disediakan apabila *test case* berhasil dijalankan.

1) Formulir pengujian halaman login

Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Berhasil
<b>I. Formulir Login</b>			
<i>Username</i> dan <i>password</i> diisi dengan data yang benar dengan level admin	Pengguna dapat masuk ke halaman utama admin (dasbor)	Sistem akan menampilkan halaman utama admin (dasbor)	✓
<i>Username</i> dan <i>password</i> diisi dengan data yang benar dengan level vendor	Pengguna dapat masuk ke halaman utama vendor (dasbor)	Sistem akan menampilkan halaman utama vendor (dasbor)	✓
<i>Username</i> dan <i>password</i> diisi dengan data yang benar dengan level toko	Pengguna dapat masuk ke halaman utama toko (dasbor)	Sistem akan menampilkan halaman utama toko (dasbor)	✓
<i>Username</i> dan <i>password</i> diisi dengan data yang salah	Pengguna tidak dapat masuk ke halaman utama	Sistem kembali ke halaman login dan menampilkan pesan error	✓

## 2) Formulir pengujian halaman admin

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
<b>1. Menu dasbor (<i>dashboard</i>)</b>			
Memilih menu dasbor	Masuk ke menu dasbor admin	Menampilkan halaman dasbor admin	✓
<b>2. Menu daftar vendor (<i>vendor partner</i>)</b>			
Memilih menu <i>vendor partner</i>	Masuk ke menu <i>vendor partner</i>	Menampilkan halaman <i>vendor partner</i>	✓
Klik tombol detail	Masuk ke menu detail vendor untuk melihat daftar produk vendor	Menampilkan halaman detail vendor berisi daftar produk vendor	✓
Klik tombol <i>add account</i>	Muncul <i>modal form</i> untuk menambahkan akun vendor baru	Menampilkan <i>modal form</i> untuk menambahkan akun vendor baru	✓
Mengisi form penambahan akun vendor baru dan klik tombol <i>register</i>	Data akun vendor baru berhasil tersimpan	Menampilkan daftar vendor di halaman <i>vendor partner</i> dan tombol <i>add account</i> berubah menjadi tombol <i>edit account</i>	✓
Klik tombol <i>edit account</i>	Muncul <i>modal form</i> untuk mengedit akun vendor	Menampilkan <i>modal form</i> untuk mengedit akun vendor	✓
Mengisi form edit akun vendor dan klik tombol <i>update</i>	Data akun vendor berhasil diperbarui	Menampilkan daftar vendor di halaman <i>vendor partner</i>	✓
<b>3. Menu daftar toko (<i>store partner</i>)</b>			
Memilih menu <i>store partner</i>	Masuk ke menu <i>store partner</i>	Menampilkan halaman <i>store partner</i>	✓
Klik tombol <i>add new store</i>	Muncul <i>modal form</i> untuk menambahkan akun toko baru	Menampilkan <i>modal form</i> untuk menambahkan akun toko baru	✓

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
Mengisi form penambahan akun toko baru dan klik tombol <i>register</i>	Data akun toko baru berhasil tersimpan	Menampilkan daftar toko di halaman <i>store partner</i>	✓
Klik tombol detail	Masuk ke menu detail toko untuk melihat daftar <i>preorder</i> toko	Menampilkan halaman detail toko berisi daftar <i>preorder</i> toko	✓
Klik tombol edit	Muncul <i>modal form</i> untuk mengedit akun toko	Menampilkan <i>modal form</i> untuk mengedit akun toko	✓
Mengisi form edit akun toko dan klik tombol <i>update</i>	Data akun toko berhasil diperbarui	Menampilkan daftar toko di halaman <i>store partner</i>	✓
Klik tombol hapus	Data akun toko berhasil dihapus	Menampilkan daftar toko di halaman <i>store partner</i>	✓
<b>4. Menu prapemesanan (<i>preorders</i>)</b>			
Memilih menu <i>preorders</i>	Masuk ke menu <i>preorders</i>	Menampilkan halaman <i>preorders</i>	✓
Klik tab <i>preorder from store</i>	Melihat daftar permintaan <i>preorder</i> dari toko	Menampilkan daftar permintaan <i>preorder</i> dari toko	✓
Klik tab <i>preorder to vendor</i>	Melihat daftar permintaan <i>preorder</i> ke vendor	Menampilkan daftar permintaan <i>preorder</i> ke vendor	✓
Klik tombol <i>process</i> di tab <i>preorder from store</i>	Memproses permintaan <i>preorder</i> dari toko	Menampilkan halaman <i>process preorder</i>	✓
Klik tombol <i>add to demand</i> di halaman <i>process preorder</i>	Memasukkan pesanan toko ke daftar permintaan <i>preorder</i> ke vendor	Menampilkan halaman <i>preorders</i>	✓
Klik tombol <i>process to vendor</i> di tab <i>preorder to vendor</i>	Memproses permintaan <i>preorder</i> ke vendor	Menampilkan daftar produk yang dipesan di halaman <i>process preorder to vendor</i>	✓

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
Klik tombol <i>submit preorder to vendor</i>	<i>Checkout</i> produk dan mengirimkan <i>preorder</i> kepada vendor	Menampilkan tab <i>preorder to vendor</i> di halaman <i>preorders</i>	✓
<b>5. Menu riwayat prapemesanan (<i>history</i>)</b>			
Memilih menu <i>history</i>	Masuk ke menu <i>history</i>	Menampilkan halaman <i>history</i>	✓
Klik tab <i>preorder from store</i>	Melihat daftar riwayat <i>preorder</i> dari toko	Menampilkan daftar riwayat <i>preorder</i> dari toko	✓
Klik tab <i>preorder to vendor</i>	Melihat daftar riwayat <i>preorder</i> ke vendor	Menampilkan daftar riwayat <i>preorder</i> ke vendor	✓
Klik tombol detail di tab <i>preorder from store</i>	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan toko	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan toko	✓
Klik tombol detail di tab <i>preorder to vendor</i>	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan ke vendor	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	✓
Jika status di tab <i>preorder to vendor</i> " <i>delivered</i> " klik tombol <i>check</i>	Memeriksa apakah produk yang dikirimkan sesuai dengan yang diterima	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	✓
Klik tombol <i>request product resend</i> di halaman detail <i>preorder</i>	Meminta pengiriman ulang produk karena ada yang tidak sesuai	Muncul <i>modal form</i> keterangan permintaan pengiriman ulang	✓
Mengisi form keterangan permintaan pengiriman ulang dan klik <i>request</i>	Permintaan pengiriman ulang beserta keterangan berhasil dikirim ke vendor	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>request product resend</i>	✓



<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
Klik tombol <i>receive product</i> di halaman detail <i>preorder</i>	Menerima produk yang telah terkirim dari vendor ke admin	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>received</i>	✓
Jika status di tab <i>preorder from store</i> "ready" klik tombol detail	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan ke vendor	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	✓
Klik tombol <i>finish preorder</i> di halaman detail <i>preorder</i>	Produk telah didistribusikan kepada toko	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>finished</i>	✓
<b>6. Menu akun admin (<i>admin accounts</i>)</b>			
Memilih menu <i>admin accounts</i>	Masuk ke menu <i>admin accounts</i>	Menampilkan halaman <i>admin accounts</i>	✓
Klik tombol <i>add new admin</i>	Muncul <i>modal form</i> untuk menambahkan akun admin baru	Menampilkan <i>modal form</i> untuk menambahkan akun admin baru	✓
Mengisi form penambahan akun admin baru dan klik tombol <i>register</i>	Data akun admin baru berhasil tersimpan	Menampilkan daftar admin di halaman <i>admin account</i>	✓
Klik tombol edit	Muncul <i>modal form</i> untuk mengedit akun admin	Menampilkan <i>modal form</i> untuk mengedit akun admin	✓
Mengisi form edit akun admin dan klik tombol <i>update</i>	Data akun admin berhasil diperbarui	Menampilkan daftar admin di halaman <i>store partner</i>	✓
Klik tombol hapus	Data akun toko berhasil dihapus	Menampilkan daftar admin di halaman <i>admin account</i>	✓

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
<b>7. Menu Pengaturan Akun</b>			
Klik tombol <i>settings</i>	Masuk ke menu <i>settings</i>	Menampilkan <i>settings</i>	✓
Mengisi form pengaturan akun dan klik tombol <i>update</i>	Data akun berhasil diperbarui	Menampilkan halaman <i>login</i>	✓
<b>8. Keluar dari Halaman Admin</b>			
Klik tombol <i>logout</i>	Kembali ke halaman <i>login</i>	Menampilkan halaman <i>login</i>	✓

## 3) Formulir pengujian halaman vendor

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
<b>1. Menu dasbor (<i>dashboard</i>)</b>			
Memilih menu dasbor	Masuk ke menu dasbor vendor	Menampilkan halaman dasbor vendor	✓
<b>2. Menu daftar produk (<i>products</i>)</b>			
Memilih menu <i>products</i>	Masuk ke menu <i>products</i>	Menampilkan halaman <i>products</i> dan daftar produk	✓
<b>3. Menu prapemesanan (<i>preorders</i>)</b>			
Memilih menu <i>preorders</i>	Masuk ke menu <i>preorders</i>	Menampilkan halaman <i>preorders</i>	✓
Klik tombol <i>process</i> di salah satu pesanan	Memproses permintaan <i>preorder</i> dari admin	Menampilkan halaman <i>process preorder</i>	✓
Klik tombol <i>accept preorder</i> di halaman <i>process preorder</i>	Menerima permintaan <i>preorder</i> dari admin	Menampilkan halaman <i>preorders</i>	✓
<b>4. Menu riwayat prapemesanan (<i>history</i>)</b>			
Memilih menu <i>history</i>	Masuk ke menu <i>history</i>	Menampilkan halaman <i>history</i>	✓
Klik tombol detail	Masuk ke menu detail <i>preorder</i> untuk melihat	Menampilkan halaman detail <i>preorder</i> berisi	✓

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
	daftar produk yang dipesan	daftar produk yang dipesan	
Jika status " <i>accepted</i> " klik tombol <i>deliver</i>	Masuk ke menu detail <i>preorder</i> untuk memperbarui status	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	✓
Klik tombol <i>deliver product</i> di halaman detail <i>preorder</i>	Memperbarui status <i>preorder</i> bahwa produk sedang dikirim	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>delivered</i>	✓
Jika status " <i>request product resend</i> " klik tombol <i>respond</i>	Masuk ke menu detail <i>preorder</i> untuk merespon permintaan	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor dan keterangan	✓
Klik tombol <i>contact admin</i> di halaman detail <i>preorder</i>	Menghubungi kontak admin	Menuju ke nomor telepon admin	✓
Klik tombol <i>resend product</i> di halaman detail <i>preorder</i>	Memperbarui status <i>preorder</i> bahwa produk sedang dikirim ulang	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>delivered</i>	✓
<b>5. Menu Pengaturan Akun</b>			
Klik tombol <i>settings</i>	Masuk ke menu <i>settings</i>	Menampilkan menu <i>settings</i>	✓
Mengisi form pengaturan akun dan klik tombol <i>update</i>	Data akun berhasil diperbarui	Menampilkan halaman <i>login</i>	✓
<b>6. Keluar dari Halaman Admin</b>			
Klik tombol <i>logout</i>	Kembali ke halaman <i>login</i>	Menampilkan halaman <i>login</i>	✓

## 4) Formulir pengujian halaman toko

<i>Test Case</i>	Hasil yang diharapkan	Hasil yang didapatkan	Berhasil
<b>1. Menu dasbor (dashboard)</b>			
Memilih menu dasbor	Masuk ke menu dasbor toko	Menampilkan halaman dasbor toko	✓
<b>2. Menu prapemesanan (preorders)</b>			
Memilih menu <i>preorders</i>	Masuk ke menu <i>preorders</i>	Menampilkan daftar vendor di halaman <i>preorders</i>	✓
Klik tombol <i>preorder product</i> di salah satu vendor	Masuk ke menu detail vendor untuk melihat daftar produk vendor	Menampilkan halaman detail vendor berisi daftar produk vendor	✓
Klik tombol <i>preorder</i> pada salah satu produk	Masuk ke menu detail produk untuk memesan produk	Menampilkan halaman detail produk	✓
Memilih opsi produk dan klik tombol <i>add to cart</i>	Memilih produk sesuai opsi dan memasukkan produk ke dalam keranjang	Menampilkan halaman detail produk	✓
<b>3. Menu keranjang (cart)</b>			
Memilih menu <i>cart</i>	Masuk ke menu <i>cart</i>	Menampilkan halaman <i>cart</i>	✓
Klik tombol <i>submit preorder</i>	<i>Checkout</i> produk dan mengirimkan <i>preorder</i> kepada admin	Menampilkan halaman <i>history</i>	✓
<b>4. Menu riwayat prapemesanan (history)</b>			
Memilih menu <i>history</i>	Masuk ke menu <i>history</i>	Menampilkan halaman <i>history</i>	✓
Klik tombol detail	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan	Menampilkan halaman detail vendor berisi daftar produk yang dipesan	✓

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
Klik tombol <i>cancel preorder</i>	<i>Preorder</i> dibatalkan	Menampilkan halaman <i>history</i> dan status menjadi " <i>cancelled</i> "	✓
Jika status " <i>cancelled</i> " klik tombol <i>edit preorder</i>	Mengubah data dalam <i>preorder</i>	Menampilkan halaman <i>cart</i> berisi data <i>preorder</i> yang diedit	✓
Memperbarui data <i>preorder</i> dan klik tombol <i>submit preorder</i> di halaman <i>cart</i>	Data <i>preorder</i> berhasil diperbarui	Menampilkan halaman <i>history</i>	✓
Jika status " <i>cancelled</i> " klik tombol <i>delete preorder</i>	Data <i>preorder</i> berhasil dihapus	Menampilkan halaman <i>history</i>	✓
<b>5. Menu Pengaturan Akun</b>			
Klik tombol <i>settings</i>	Masuk ke menu <i>settings</i>	Menampilkan menu <i>settings</i>	✓
Mengisi form pengaturan akun dan klik tombol <i>update</i>	Data akun berhasil diperbarui	Menampilkan halaman <i>login</i>	✓
<b>6. Keluar dari Halaman Admin</b>			
Klik tombol <i>logout</i>	Kembali ke halaman <i>login</i>	Menampilkan halaman <i>login</i>	✓

Saran dan masukan:

-

Responden

*Rahmat Rabi W. M. (Kor.)*  
NIDN. 060109803

### Lampiran 6. Formulir Pengujian *Black-box* Responden 3

**FORM KUESIONER PENGUJIAN *BLACK-BOX***  
**APLIKASI PENGELOLAAN DISTRIBUSI PRODUK USAHA KECIL MENENGAH**  
**BERBASIS WEB DENGAN INTEGRASI *CONTENT MANAGEMENT SYSTEM***  
**SHOPIFY DI NETASIA SINGAPORE**

Tanggal pengujian : 16 Maret 2022  
 Nama Responden : Nugroho Ari S  
 Jabatan/status : Dean Informatika UPERIS

Mohon perhatikan perlakuan dengan seksama dan harap mengisi form kuesioner berikut ini dengan jujur dan sesuai dengan kondisi yang terjadi. Beri tanda centang (✓) pada kolom yang sudah disediakan apabila *test case* berhasil dijalankan.

1) Formulir pengujian halaman login

<i>Test Case</i>	Hasil yang diharapkan	Hasil yang didapatkan	Berhasil
<b>I. Formulir Login</b>			
<i>Username</i> dan <i>password</i> diisi dengan data yang benar dengan level admin	Pengguna dapat masuk ke halaman utama admin (dasbor)	Sistem akan menampilkan halaman utama admin (dasbor)	✓
<i>Username</i> dan <i>password</i> diisi dengan data yang benar dengan level vendor	Pengguna dapat masuk ke halaman utama vendor (dasbor)	Sistem akan menampilkan halaman utama vendor (dasbor)	✓
<i>Username</i> dan <i>password</i> diisi dengan data yang benar dengan level toko	Pengguna dapat masuk ke halaman utama toko (dasbor)	Sistem akan menampilkan halaman utama toko (dasbor)	✓
<i>Username</i> dan <i>password</i> diisi dengan data yang salah	Pengguna tidak dapat masuk ke halaman utama	Sistem kembali ke halaman login dan menampilkan pesan error	✓

## 2) Formulir pengujian halaman admin

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
<b>1. Menu dasbor (<i>dashboard</i>)</b>			
Memilih menu dasbor	Masuk ke menu dasbor admin	Menampilkan halaman dasbor admin	✓
<b>2. Menu daftar vendor (<i>vendor partner</i>)</b>			
Memilih menu <i>vendor partner</i>	Masuk ke menu <i>vendor partner</i>	Menampilkan halaman <i>vendor partner</i>	✓
Klik tombol detail	Masuk ke menu detail vendor untuk melihat daftar produk vendor	Menampilkan halaman detail vendor berisi daftar produk vendor	✓
Klik tombol <i>add account</i>	Muncul <i>modal form</i> untuk menambahkan akun vendor baru	Menampilkan <i>modal form</i> untuk menambahkan akun vendor baru	✓
Mengisi form penambahan akun vendor baru dan klik tombol <i>register</i>	Data akun vendor baru berhasil tersimpan	Menampilkan daftar vendor di halaman <i>vendor partner</i> dan tombol <i>add account</i> berubah menjadi tombol <i>edit account</i>	✓
Klik tombol <i>edit account</i>	Muncul <i>modal form</i> untuk mengedit akun vendor	Menampilkan <i>modal form</i> untuk mengedit akun vendor	✓
Mengisi form edit akun vendor dan klik tombol <i>update</i>	Data akun vendor berhasil diperbarui	Menampilkan daftar vendor di halaman <i>vendor partner</i>	✓
<b>3. Menu daftar toko (<i>store partner</i>)</b>			
Memilih menu <i>store partner</i>	Masuk ke menu <i>store partner</i>	Menampilkan halaman <i>store partner</i>	✓
Klik tombol <i>add new store</i>	Muncul <i>modal form</i> untuk menambahkan akun toko baru	Menampilkan <i>modal form</i> untuk menambahkan akun toko baru	✓

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
Mengisi form penambahan akun toko baru dan klik tombol <i>register</i>	Data akun toko baru berhasil tersimpan	Menampilkan daftar toko di halaman <i>store partner</i>	✓
Klik tombol detail	Masuk ke menu detail toko untuk melihat daftar <i>preorder</i> toko	Menampilkan halaman detail toko berisi daftar <i>preorder</i> toko	✓
Klik tombol edit	Muncul <i>modal form</i> untuk mengedit akun toko	Menampilkan <i>modal form</i> untuk mengedit akun toko	✓
Mengisi form edit akun toko dan klik tombol <i>update</i>	Data akun toko berhasil diperbarui	Menampilkan daftar toko di halaman <i>store partner</i>	✓
Klik tombol hapus	Data akun toko berhasil dihapus	Menampilkan daftar toko di halaman <i>store partner</i>	✓
<b>4. Menu prapemesanan (<i>preorders</i>)</b>			
Memilih menu <i>preorders</i>	Masuk ke menu <i>preorders</i>	Menampilkan halaman <i>preorders</i>	✓
Klik tab <i>preorder from store</i>	Melihat daftar permintaan <i>preorder</i> dari toko	Menampilkan daftar permintaan <i>preorder</i> dari toko	✓
Klik tab <i>preorder to vendor</i>	Melihat daftar permintaan <i>preorder</i> ke vendor	Menampilkan daftar permintaan <i>preorder</i> ke vendor	✓
Klik tombol <i>process</i> di tab <i>preorder from store</i>	Memproses permintaan <i>preorder</i> dari toko	Menampilkan halaman <i>process preorder</i>	✓
Klik tombol <i>add to demand</i> di halaman <i>process preorder</i>	Memasukkan pesanan toko ke daftar permintaan <i>preorder</i> ke vendor	Menampilkan halaman <i>preorders</i>	✓
Klik tombol <i>process to vendor</i> di tab <i>preorder to vendor</i>	Memproses permintaan <i>preorder</i> ke vendor	Menampilkan daftar produk yang dipesan di halaman <i>process preorder to vendor</i>	✓



<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
Klik tombol <i>submit preorder to vendor</i>	<i>Checkout</i> produk dan mengirimkan <i>preorder</i> kepada vendor	Menampilkan tab <i>preorder to vendor</i> di halaman <i>preorders</i>	✓
<b>5. Menu riwayat prapemesanan (<i>history</i>)</b>			
Memilih menu <i>history</i>	Masuk ke menu <i>history</i>	Menampilkan halaman <i>history</i>	✓
Klik tab <i>preorder from store</i>	Melihat daftar riwayat <i>preorder</i> dari toko	Menampilkan daftar riwayat <i>preorder</i> dari toko	✓
Klik tab <i>preorder to vendor</i>	Melihat daftar riwayat <i>preorder</i> ke vendor	Menampilkan daftar riwayat <i>preorder</i> ke vendor	✓
Klik tombol detail di tab <i>preorder from store</i>	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan toko	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan toko	✓
Klik tombol detail di tab <i>preorder to vendor</i>	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan ke vendor	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	✓
Jika status di tab <i>preorder to vendor</i> " <i>delivered</i> " klik tombol <i>check</i>	Memeriksa apakah produk yang dikirimkan sesuai dengan yang diterima	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	✓
Klik tombol <i>request product resend</i> di halaman detail <i>preorder</i>	Meminta pengiriman ulang produk karena ada yang tidak sesuai	Muncul <i>modal form</i> keterangan permintaan pengiriman ulang	✓
Mengisi form keterangan permintaan pengiriman ulang dan klik <i>request</i>	Permintaan pengiriman ulang beserta keterangan berhasil dikirim ke vendor	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>request product resend</i>	✓

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
Klik tombol <i>receive product</i> di halaman detail <i>preorder</i>	Menerima produk yang telah terkirim dari vendor ke admin	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>received</i>	✓
Jika status di tab <i>preorder from store</i> "ready" klik tombol detail	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan ke vendor	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	✓
Klik tombol <i>finish preorder</i> di halaman detail <i>preorder</i>	Produk telah didistribusikan kepada toko	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>finished</i>	✓
<b>6. Menu akun admin (<i>admin accounts</i>)</b>			
Memilih menu <i>admin accounts</i>	Masuk ke menu <i>admin accounts</i>	Menampilkan halaman <i>admin accounts</i>	✓
Klik tombol <i>add new admin</i>	Muncul <i>modal form</i> untuk menambahkan akun admin baru	Menampilkan <i>modal form</i> untuk menambahkan akun admin baru	✓
Mengisi form penambahan akun admin baru dan klik tombol <i>register</i>	Data akun admin baru berhasil tersimpan	Menampilkan daftar admin di halaman <i>admin account</i>	✓
Klik tombol edit	Muncul <i>modal form</i> untuk mengedit akun admin	Menampilkan <i>modal form</i> untuk mengedit akun admin	✓
Mengisi form edit akun admin dan klik tombol <i>update</i>	Data akun admin berhasil diperbarui	Menampilkan daftar admin di halaman <i>store partner</i>	✓
Klik tombol hapus	Data akun toko berhasil dihapus	Menampilkan daftar admin di halaman <i>admin account</i>	✓

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
<b>7. Menu Pengaturan Akun</b>			
Klik tombol <i>settings</i>	Masuk ke menu <i>settings</i>	Menampilkan <i>settings</i>	✓
Mengisi form pengaturan akun dan klik tombol <i>update</i>	Data akun berhasil diperbarui	Menampilkan halaman <i>login</i>	✓
<b>8. Keluar dari Halaman Admin</b>			
Klik tombol <i>logout</i>	Kembali ke halaman <i>login</i>	Menampilkan halaman <i>login</i>	✓

## 3) Formulir pengujian halaman vendor

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
<b>1. Menu dasbor (<i>dashboard</i>)</b>			
Memilih menu dasbor	Masuk ke menu dasbor vendor	Menampilkan halaman dasbor vendor	✓
<b>2. Menu daftar produk (<i>products</i>)</b>			
Memilih menu <i>products</i>	Masuk ke menu <i>products</i>	Menampilkan halaman <i>products</i> dan daftar produk	✓
<b>3. Menu prapemesanan (<i>preorders</i>)</b>			
Memilih menu <i>preorders</i>	Masuk ke menu <i>preorders</i>	Menampilkan halaman <i>preorders</i>	✓
Klik tombol <i>process</i> di salah satu pesanan	Memproses permintaan <i>preorder</i> dari admin	Menampilkan halaman <i>process preorder</i>	✓
Klik tombol <i>accept preorder</i> di halaman <i>process preorder</i>	Menerima permintaan <i>preorder</i> dari admin	Menampilkan halaman <i>preorders</i>	✓
<b>4. Menu riwayat prapemesanan (<i>history</i>)</b>			
Memilih menu <i>history</i>	Masuk ke menu <i>history</i>	Menampilkan halaman <i>history</i>	✓
Klik tombol detail	Masuk ke menu detail <i>preorder</i> untuk melihat	Menampilkan halaman detail <i>preorder</i> berisi	✓

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
	daftar produk yang dipesan	daftar produk yang dipesan	
Jika status "accepted" klik tombol <i>deliver</i>	Masuk ke menu detail <i>preorder</i> untuk memperbarui status	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor	✓
Klik tombol <i>deliver product</i> di halaman detail <i>preorder</i>	Memperbarui status <i>preorder</i> bahwa produk sedang dikirim	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>delivered</i>	✓
Jika status "request <i>product resend</i> " klik tombol <i>respond</i>	Masuk ke menu detail <i>preorder</i> untuk merespon permintaan	Menampilkan halaman detail <i>preorder</i> berisi daftar produk yang dipesan ke vendor dan keterangan	✓
Klik tombol <i>contact admin</i> di halaman detail <i>preorder</i>	Menghubungi kontak admin	Menuju ke nomor telepon admin	✓
Klik tombol <i>resend product</i> di halaman detail <i>preorder</i>	Memperbarui status <i>preorder</i> bahwa produk sedang dikirim ulang	Menampilkan halaman <i>history</i> dan status berubah menjadi <i>delivered</i>	✓
<b>5. Menu Pengaturan Akun</b>			
Klik tombol <i>settings</i>	Masuk ke menu <i>settings</i>	Menampilkan menu <i>settings</i>	✓
Mengisi form pengaturan akun dan klik tombol <i>update</i>	Data akun berhasil diperbarui	Menampilkan halaman <i>login</i>	✓
<b>6. Keluar dari Halaman Admin</b>			
Klik tombol <i>logout</i>	Kembali ke halaman <i>login</i>	Menampilkan halaman <i>login</i>	✓


## 4) Formulir pengujian halaman toko

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
<b>1. Menu dasbor (<i>dashboard</i>)</b>			
Memilih menu dasbor	Masuk ke menu dasbor toko	Menampilkan halaman dasbor toko	✓
<b>2. Menu prapemesanan (<i>preorders</i>)</b>			
Memilih menu <i>preorders</i>	Masuk ke menu <i>preorders</i>	Menampilkan daftar vendor di halaman <i>preorders</i>	✓
Klik tombol <i>preorder product</i> di salah satu vendor	Masuk ke menu detail vendor untuk melihat daftar produk vendor	Menampilkan halaman detail vendor berisi daftar produk vendor	✓
Klik tombol <i>preorder</i> pada salah satu produk	Masuk ke menu detail produk untuk memesan produk	Menampilkan halaman detail produk	✓
Memilih opsi produk dan klik tombol <i>add to cart</i>	Memilih produk sesuai opsi dan memasukkan produk ke dalam keranjang	Menampilkan halaman detail produk	✓
<b>3. Menu keranjang (<i>cart</i>)</b>			
Memilih menu <i>cart</i>	Masuk ke menu <i>cart</i>	Menampilkan halaman <i>cart</i>	✓
Klik tombol <i>submit preorder</i>	<i>Checkout</i> produk dan mengirimkan <i>preorder</i> kepada admin	Menampilkan halaman <i>history</i>	✓
<b>4. Menu riwayat prapemesanan (<i>history</i>)</b>			
Memilih menu <i>history</i>	Masuk ke menu <i>history</i>	Menampilkan halaman <i>history</i>	
Klik tombol detail	Masuk ke menu detail <i>preorder</i> untuk melihat daftar produk yang dipesan	Menampilkan halaman detail vendor berisi daftar produk yang dipesan	✓

<i>Test Case</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang didapatkan</i>	<i>Berhasil</i>
Klik tombol <i>cancel preorder</i>	<i>Preorder</i> dibatalkan	Menampilkan halaman <i>history</i> dan status menjadi " <i>cancelled</i> "	✓
Jika status " <i>cancelled</i> " klik tombol <i>edit preorder</i>	Mengubah data dalam <i>preorder</i>	Menampilkan halaman <i>cart</i> berisi data <i>preorder</i> yang diedit	✓
Memperbarui data <i>preorder</i> dan klik tombol <i>submit preorder</i> di halaman <i>cart</i>	Data <i>preorder</i> berhasil diperbarui	Menampilkan halaman <i>history</i>	✓
Jika status " <i>cancelled</i> " klik tombol <i>delete preorder</i>	Data <i>preorder</i> berhasil dihapus	Menampilkan halaman <i>history</i>	✓
<b>5. Menu Pengaturan Akun</b>			
Klik tombol <i>settings</i>	Masuk ke menu <i>settings</i>	Menampilkan menu <i>settings</i>	✓
Mengisi form pengaturan akun dan klik tombol <i>update</i>	Data akun berhasil diperbarui	Menampilkan halaman <i>login</i>	✓
<b>6. Keluar dari Halaman Admin</b>			
Klik tombol <i>logout</i>	Kembali ke halaman <i>login</i>	Menampilkan halaman <i>login</i>	✓

Saran dan masukan:

Responden

  
(Nugraha Ari S.)

## Lampiran 7. Formulir *User Acceptance Testing* (UAT) Responden 1

**FORM KUESIONER *USER ACCEPTANCE TEST* (UAT)  
 APLIKASI PENGELOLAAN DISTRIBUSI PRODUK USAHA KECIL MENENGAH  
 BERBASIS WEB DENGAN INTEGRASI *CONTENT MANAGEMENT SYSTEM*  
 SHOPIFY DI NETASIA SINGAPORE**

Tanggal pengujian : 3/17/2022 14:57:55 PM

Nama Responden : Sripeni Puspasari

Jabatan/status : Director

Level dalam sistem : Admin

Sebelum mengisi form kuesioner, mohon perhatikan keterangan skor berikut.

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Cukup Setuju	3
Kurang Setuju	2
Tidak Setuju	1

Mohon perhatikan pertanyaan dengan seksama dan harap mengisi form kuesioner berikut ini dengan jujur dan sesuai dengan kondisi yang terjadi. Beri tanda centang (√) pada kolom sesuai skor yang diberikan.

No	Pertanyaan	Skor
<b>Aspek Kemanfaatan</b>		
1	Apakah aplikasi berbasis web ini bermanfaat untuk membantu pengelolaan distribusi produk UKM di Netasia secara keseluruhan?	4
2	Apakah aplikasi berbasis web ini bermanfaat untuk admin dalam mengelola distribusi produk UKM di Netasia?	5
3	Apakah aplikasi berbasis web ini bermanfaat untuk vendor dalam menerima preorder dari Netasia?	5
4	Apakah aplikasi berbasis web ini bermanfaat untuk toko dalam melakukan preorder produk UKM di Netasia?	5
5	Apakah aplikasi berbasis web ini membuat pengelolaan distribusi produk UKM di Netasia menjadi lebih efektif dan efisien?	4

No	Pertanyaan	Skor
<b>Aspek Kemudahan Penggunaan</b>		
6	Apakah aplikasi berbasis web ini mudah dipahami?	4
7	Apakah aplikasi berbasis web ini mudah digunakan?	5
8	Apakah aplikasi berbasis web ini dapat berjalan sesuai yang diharapkan?	4
9	Apakah aplikasi berbasis web ini mudah dimengerti oleh admin?	5
10	Apakah aplikasi berbasis web ini mudah dimengerti oleh vendor?	5
11	Apakah aplikasi berbasis web ini mudah dimengerti oleh toko?	5
<b>Aspek Antarmuka Pengguna (<i>User Interface</i>)</b>		
12	Apakah aplikasi berbasis web ini memiliki tampilan yang jelas?	5
13	Apakah aplikasi berbasis web ini memiliki tampilan yang baik?	5

Saran dan masukan:

Bisa ditambahkan agregasi stok tersedia di admin dan notifikasi jika ada perubahan status preorder



## Lampiran 8. Formulir *User Acceptance Testing* (UAT) Responden 2

**FORM KUESIONER *USER ACCEPTANCE TEST* (UAT)  
APLIKASI PENGELOLAAN DISTRIBUSI PRODUK USAHA KECIL MENENGAH  
BERBASIS WEB DENGAN INTEGRASI *CONTENT MANAGEMENT SYSTEM*  
SHOPIFY DI NETASIA SINGAPORE**

Tanggal pengujian : 3/17/2022 15:03:12 PM

Nama Responden : Yeste Sujut Triyanta

Jabatan/status : IT Management

Level dalam sistem : Store

Sebelum mengisi form kuesioner, mohon perhatikan keterangan skor berikut.

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Cukup Setuju	3
Kurang Setuju	2
Tidak Setuju	1

Mohon perhatikan pertanyaan dengan seksama dan harap mengisi form kuesioner berikut ini dengan jujur dan sesuai dengan kondisi yang terjadi. Beri tanda centang (√) pada kolom sesuai skor yang diberikan.

No	Pertanyaan	Skor
<b>Aspek Kemanfaatan</b>		
1	Apakah aplikasi berbasis web ini bermanfaat untuk membantu pengelolaan distribusi produk UKM di Netasia secara keseluruhan?	5
2	Apakah aplikasi berbasis web ini bermanfaat untuk admin dalam mengelola distribusi produk UKM di Netasia?	5
3	Apakah aplikasi berbasis web ini bermanfaat untuk vendor dalam menerima preorder dari Netasia?	5
4	Apakah aplikasi berbasis web ini bermanfaat untuk toko dalam melakukan preorder produk UKM di Netasia?	5
5	Apakah aplikasi berbasis web ini membuat pengelolaan distribusi produk UKM di Netasia menjadi lebih efektif dan efisien?	4

No	Pertanyaan	Skor
<b>Aspek Kemudahan Penggunaan</b>		
6	Apakah aplikasi berbasis web ini mudah dipahami?	5
7	Apakah aplikasi berbasis web ini mudah digunakan?	5
8	Apakah aplikasi berbasis web ini dapat berjalan sesuai yang diharapkan?	5
9	Apakah aplikasi berbasis web ini mudah dimengerti oleh admin?	5
10	Apakah aplikasi berbasis web ini mudah dimengerti oleh vendor?	5
11	Apakah aplikasi berbasis web ini mudah dimengerti oleh toko?	5
<b>Aspek Antarmuka Pengguna (<i>User Interface</i>)</b>		
12	Apakah aplikasi berbasis web ini memiliki tampilan yang jelas?	5
13	Apakah aplikasi berbasis web ini memiliki tampilan yang baik?	5

Saran dan masukan:

Ada beberapa halaman yang belum responsif terhadap layar device seperti hp

### Lampiran 9. Formulir *User Acceptance Testing* (UAT) Responden 3

**FORM KUESIONER *USER ACCEPTANCE TEST* (UAT)  
 APLIKASI PENGELOLAAN DISTRIBUSI PRODUK USAHA KECIL MENENGAH  
 BERBASIS WEB DENGAN INTEGRASI *CONTENT MANAGEMENT SYSTEM*  
 SHOPIFY DI NETASIA SINGAPORE**

Tanggal pengujian : 3/17/2022 15:08:11 PM  
 Nama Responden : Dipta  
 Jabatan/status : Netasia Shopify Administrator  
 Level dalam sistem : Vendor

Sebelum mengisi form kuesioner, mohon perhatikan keterangan skor berikut.

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Cukup Setuju	3
Kurang Setuju	2
Tidak Setuju	1

Mohon perhatikan pertanyaan dengan seksama dan harap mengisi form kuesioner berikut ini dengan jujur dan sesuai dengan kondisi yang terjadi. Beri tanda centang (√) pada kolom sesuai skor yang diberikan.

No	Pertanyaan	Skor
<b>Aspek Kemanfaatan</b>		
1	Apakah aplikasi berbasis web ini bermanfaat untuk membantu pengelolaan distribusi produk UKM di Netasia secara keseluruhan?	4
2	Apakah aplikasi berbasis web ini bermanfaat untuk admin dalam mengelola distribusi produk UKM di Netasia?	5
3	Apakah aplikasi berbasis web ini bermanfaat untuk vendor dalam menerima preorder dari Netasia?	5
4	Apakah aplikasi berbasis web ini bermanfaat untuk toko dalam melakukan preorder produk UKM di Netasia?	5
5	Apakah aplikasi berbasis web ini membuat pengelolaan distribusi produk UKM di Netasia menjadi lebih efektif dan efisien?	4

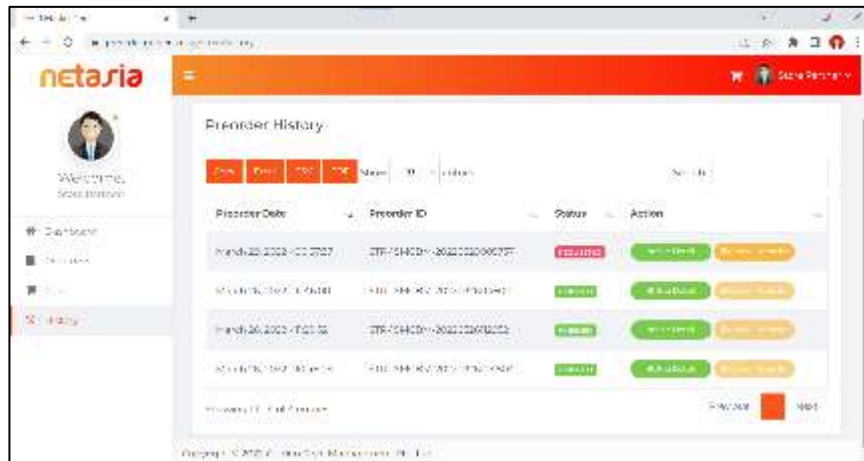
No	Pertanyaan	Skor
<b>Aspek Kemudahan Penggunaan</b>		
6	Apakah aplikasi berbasis web ini mudah dipahami?	4
7	Apakah aplikasi berbasis web ini mudah digunakan?	4
8	Apakah aplikasi berbasis web ini dapat berjalan sesuai yang diharapkan?	5
9	Apakah aplikasi berbasis web ini mudah dimengerti oleh admin?	5
10	Apakah aplikasi berbasis web ini mudah dimengerti oleh vendor?	5
11	Apakah aplikasi berbasis web ini mudah dimengerti oleh toko?	5
<b>Aspek Antarmuka Pengguna (<i>User Interface</i>)</b>		
12	Apakah aplikasi berbasis web ini memiliki tampilan yang jelas?	5
13	Apakah aplikasi berbasis web ini memiliki tampilan yang baik?	4

Saran dan masukan:

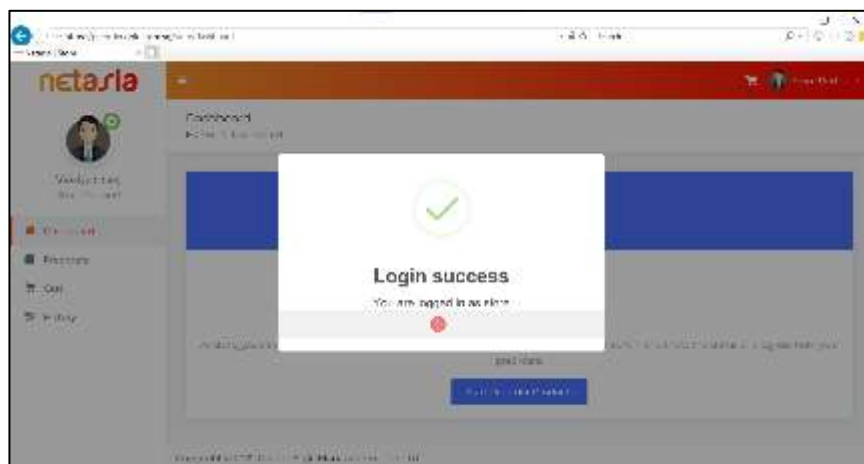
Kalau bisa ada notifikasi

## Lampiran 10. Bukti Pengujian Kompatibilitas Aplikasi *Browser*

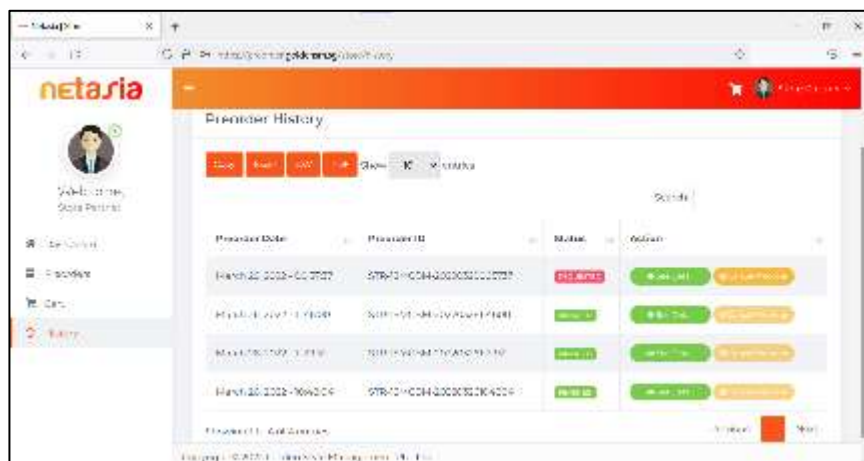
### 1. Google Chrome



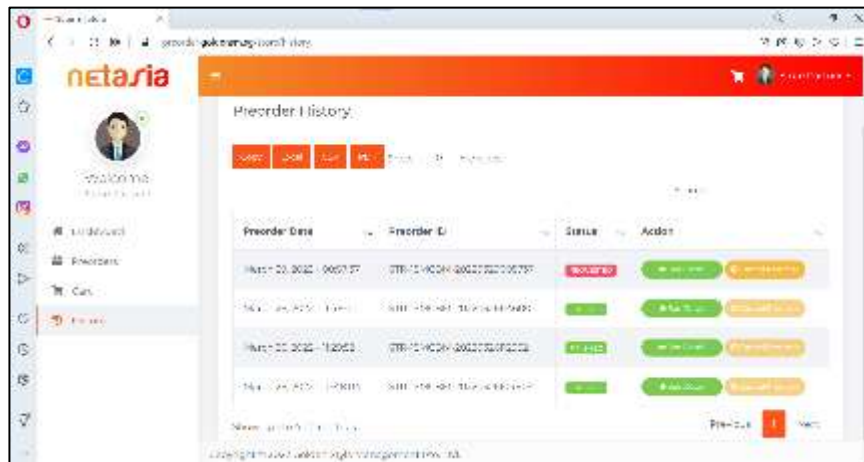
### 2. Internet Explorer



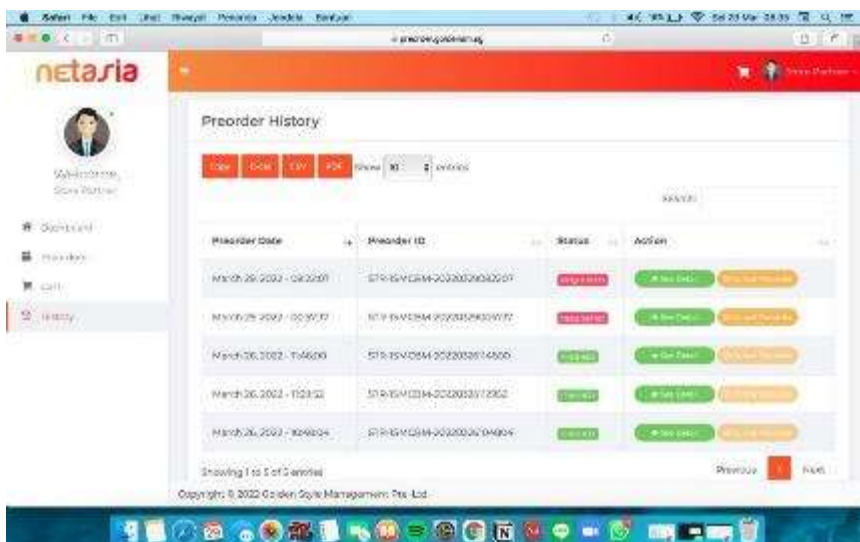
### 3. Mozilla Firefox



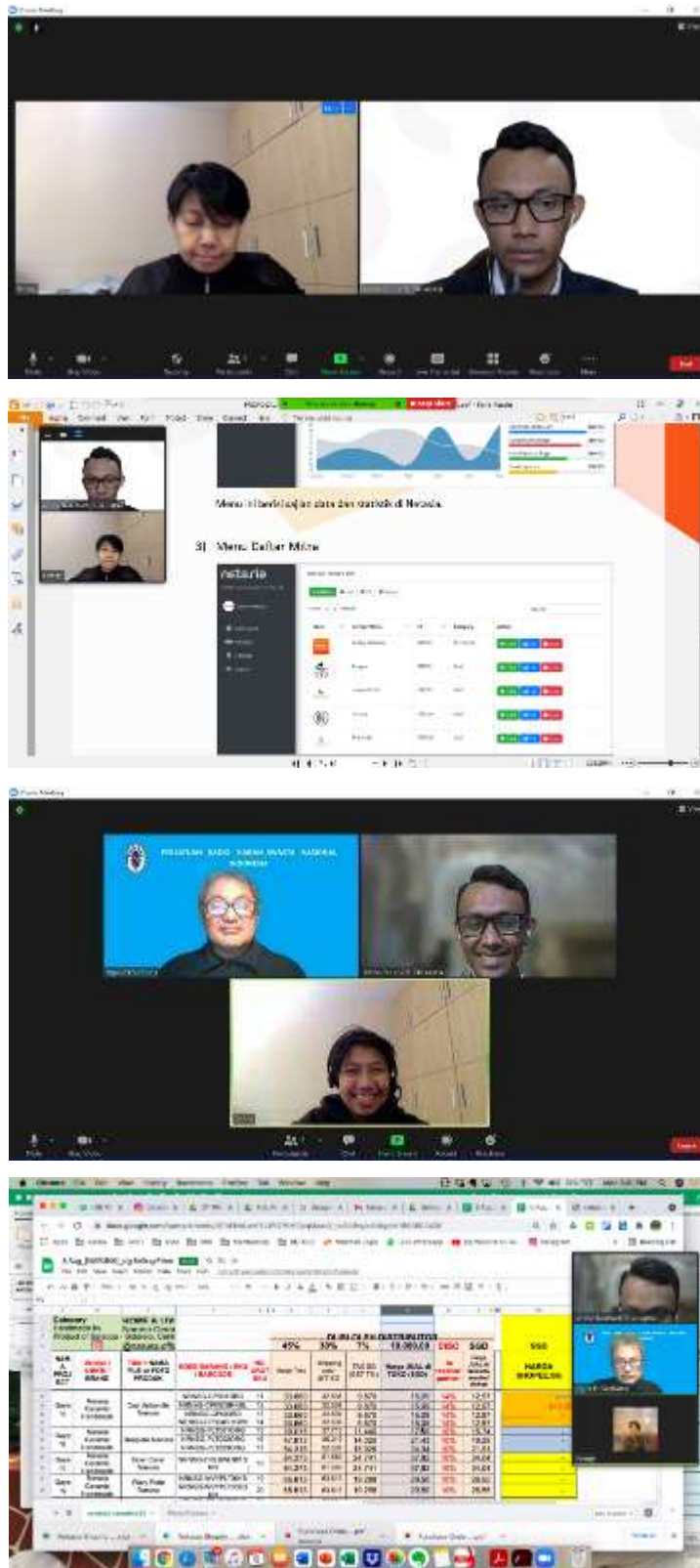
#### 4. Opera Browser



#### 5. Safari Browser



Lampiran 11. Dokumentasi Penelitian




## Lampiran 12. Lembar Revisi Ujian Skripsi Penguji 1

### LEMBAR REVISI UJIAN SKRIPSI

Nama Mahasiswa : Ansha Raihadi Trikusuma  
 N P M : 18670028  
 Judul : Aplikasi Pengelolaan Distribusi Produk Usaha Kecil Menengah Berbasis Web dengan Integrasi Content Management System Shopify di Netasia Singapore

No	Uraian Revisi	Keterangan
①	Abstrak diperjelas. ✓	acc 9/9-2022
②	pada Desain I/O jelaskan mengapa? keaktifan pd tiap gambar. ✓	✓
③	Class Susunakan Use Case. ✓	✓

Pengesahan Penguji I

  
 Bambang Agus H, S. Kom, M. Kom  
 NIP/NPP. 148201433

\*) Revisi Maksimal 7 Hari Setelah Pelaksanaan Ujian Skripsi



### Lampiran 13. Lembar Revisi Ujian Skripsi Penguji 2

#### LEMBAR REVISI UJIAN SKRIPSI

Nama Mahasiswa : Arsha Raunadi Trikusuma  
 N P M : 18670028  
 Judul : Aplikasi Pengelolaan Distribusi Produk Usaha Kecil Menengah Berbasis Web dengan Integrasi Content Management System Shopify di Netasia Singapore

No	Uraian Revisi	Keterangan
1.	Abstrak disejajarkan.	2/
2.	Semua gambar di ppt dimasukkan ke skripsi	2/
3.	Gambar disesuaikan sesuai TMR 10 ppt. white box	2/
4.	Aplikasi ditambah status/laporan barang ready?	2/
		all - 4/4 21

Pengesahan Penguji II

Mega Novita, Ph.D  
 NIP/NPP. 158801493


\*) Revisi Maksimal 7 Hari Setelah Pelaksanaan Ujian Skripsi

### Lampiran 14. Lembar Revisi Ujian Skripsi Penguji 3

**LEMBAR REVISI UJIAN SKRIPSI**

Nama Mahasiswa : Arsha Raulnadi Trikusuma  
 N P M : 18670028  
 Judul : Aplikasi Pengelolaan Distribusi Produk Usaha Kecil Menengah Berbasis Web dengan Integrasi Content Management System Shopify di Netasia Singapore

No	Uraian Revisi	Keterangan
①	Abstract.	✓
②	Variabel penelitian penelitian 7. Karena penelitian ini R&D. tp. tdk. sampai menguji variabel 3 yg dibutuhkan.	✓ Revisi acc Horu 9/10/22 9
③	Status. tiap. user pengguna s: vendor. dan admin.	✓

Pengesahan Penguji III  
  
 Noora Cotrun Nada, S.T., M.ENG  
 NIP/NPP. 158201485

\*) Revisi Maksimal 7 Hari Setelah Pelaksanaan Ujian Skripsi

## Lampiran 15. Biodata Penulis

### A. Identitas Diri

1	Nama Lengkap	Arsha Raulnadi Trikusuma
2	Nama Sapaan	Arsha
3	Jenis Kelamin	Laki-laki
4	Program Studi	Informatika
5	Fakultas	Teknik dan Informatika
6	Perguruan Tinggi	Universitas PGRI Semarang
7	NPM	18670028
8	Tempat dan Tanggal Lahir	Semarang, 9 Juli 2000
9	Alamat Rumah	Jl. Bukit Beringin Selatan II Blok F/129, Kelurahan Gondoriyo, Kecamatan
10	Alamat E-mail	arsha.trikusuma88@gmail.com
11	Nomor Telepon/HP	083838863174/087736121113

### B. Riwayat Pendidikan

	SD	SMP	SMA	S-1
Almamater	SD Negeri Kalibanteng Kidul 03	SMP Negeri 30 Semarang	SMA Negeri 5 Semarang	Informatika Universitas PGRI Semarang
Tahun Masuk-Lulus	2006-2012	2012-2015	2015-2018	2018-2022

### C. Kegiatan Kemahasiswaan

No.	Jenis Kegiatan	Peran	Waktu dan Tempat
1	Latihan Keterampilan Manajemen Mahasiswa Tingkat Pra-Dasar	Peserta	2019, Universitas PGRI Semarang
2	Latihan Keterampilan Manajemen Mahasiswa Tingkat Dasar	Peserta	2020, Universitas PGRI Semarang

3	Training Legislatif Dewan Perwakilan Mahasiswa Universitas PGRI Semarang	Peserta	2020, Universitas PGRI Semarang
4	Program Merdeka belajar Kampus Merdeka: Bangkit 2021 – Kemdikbud, Google, Traveloka, Tokopedia	Peserta	2021, Daring

#### D. Pengalaman Organisasi/Komunitas

No.	Jenis Kegiatan	Jabatan	Waktu dan Tempat
1	UKM Kajian Ilmiah dan Penelitian Mahasiswa	Sekretaris Departemen Sarana Publikasi dan Informasi	2019, Universitas PGRI Semarang
2	UKM Kajian Ilmiah dan Penelitian Mahasiswa	Kepala Departemen Sarana Publikasi dan Informasi	2020, Universitas PGRI Semarang
3	UKM Kajian Ilmiah dan Penelitian Mahasiswa	Ketua Umum	2021, Universitas PGRI Semarang
4	Google Developer Students Clubs Batch 2020	DSC Lead chapter UPGRIS	2020-2021, Universitas PGRI Semarang

#### E. Penghargaan

No.	Jenis Penghargaan	Pihak Pemberi Penghargaan	Tahun
1	Juara 2 Lomba Karya Tulis Ilmiah Mahasiswa Nasional	Universitas Nusantara PGRI Kediri	2019
2	Semifinalis Lomba Karya	Universitas Negeri Medan	2019

	Tulis Ilmiah Mahasiswa Nasional		
3	Semifinalis Lomba Karya Tulis Ilmiah Mahasiswa Nasional	Universitas Negeri Jember	2019
4	10 Finalis Terbaik Gebyar PKM Tingkat Universitas PGRI Semarang	Universitas PGRI Semarang	2019
5	Juara 1 Lomba Poster <i>Informatics Fun Competition</i> Tingkat Program Studi Informatika UPGRI	HIMA Informatika Universitas PGRI Semarang	2019
6	<i>Best Presentation</i> Lomba Karya Tulis Ilmiah Mahasiswa Nasional	Universitas Muhammadiyah Surakarta	2019
7.	Juara 3 Lomba Poster Perpustakaan Tingkat Universitas PGRI Semarang	UPT Perpustakaan Universitas PGRI Semarang	2020
8.	Juara 2 Pemilihan Mahasiswa Berprestasi Tingkat Fakultas Teknik dan Informatika Universitas PGRI Semarang	Fakultas Teknik dan Informatika Universitas PGRI Semarang	2020
9.	Pendanaan Program Kreativitas Mahasiswa bidang Karsa Cipta 2020	Ditjen Belmawa Kemdikbud	2020
10.	Pendanaan Kompetisi Inovasi Bisnis Mahasiswa 2020	Puspresnas Kemdikbud	2020

11.	Pendanaan Program Kreativitas Mahasiswa bidang Artikel Ilmiah 2021	Ditjen Belmawa Kemdikbud	2021
12.	Juara 1 Pemilihan Mahasiswa Berprestasi Tingkat Fakultas Teknik dan Informatika Universitas PGRI Semarang	Fakultas Teknik dan Informatika Universitas PGRI Semarang	2021
13.	Juara 1 Pemilihan Mahasiswa Berprestasi Tingkat Universitas PGRI Semarang	Universitas PGRI Semarang	2021
14.	Mahasiswa Berprestasi Tingkat Wilayah LLDIKTI-VI Jawa Tengah	LLDIKTI Wilayah VI Jawa Tengah	2021

#### F. Publikasi

No.	Judul Publikasi	Penulis Lainnya	Jurnal/Penerbit, Tahun
1	Buku ISBN - “Mengukir Cerita: Kumpulan Tulisan Pengurus UKM KIPM”	Afiatunnisa’ dan 33 lainnya	Beruang Cipta Literasi, 2020
2	Buku ISBN - “Mahasiswa dan Pandemi Covid-19”	Nurun Nadzifah dan 26 lainnya	Tiga Cakrawala, 2021
3	Artikel Ilmiah - “Branket” Design as a Safe Deposit Box Security System using Arduino-Based Tap Sensor	Mona Rizqa, Dhimas Aria Wardhana, Noora Qotrun Nada, S.T., M.Eng.	Advance Sustainable Science, Environmental Engineering and Technology (ASSET), 2021

4	Artikel Ilmiah – “Aplikasi Pengelolaan Distribusi Produk Usaha Kecil Menengah Berbasis Web di Netasia Singapore”	Mega Novita, S.Si., M.Si., M.Nat.Sc. Ph.D.	Science and Engineering National Seminar 6 2021
---	--	--	---

### G. Hak Cipta

No.	Nama Hak Cipta	Jenis Hak Cipta	Waktu
1	"Mengukir Cerita: Kumpulan Tulisan Pengurus UKM KIPM"	Hak Cipta Buku	2021
2	SIPAMDES (Sistem Informasi PAM Desa) Berbasis Website	Hak Cipta Program Komputer	2021