



**DIAGNOSA PENYAKIT YANG DISEBABKAN OLEH KELEBIHAN GULA
DARAH BERBASIS *WEB* DENGAN MENGGUNAKAN METODE *CASE
BASED REASONING***

SKRIPSI

ADITIO TRI NUGROHO

15670041

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

2021



**DIAGNOSA PENYAKIT YANG DISEBABKAN OLEH KELEBIHAN GULA
DARAH BERBASIS *WEB* DENGAN MENGGUNAKAN METODE *CASE
BASED REASONING***

SKRIPSI

**Diajukan kepada Fakultas Teknik dan Informatika
Universitas PGRI Semarang untuk Penyusunan Skripsi**

ADITIO TRI NUGROHO

15670041

PROGRAM STUDI INFORMATIKA

UNIVERSITAS PGRI SEMARANG

2021

HALAMAN PERSETUJUAN

SKRIPSI

**DIAGNOSA PENYAKIT YANG DISEBABKAN OLEH KELEBIHAN GULA
DARAH BERBASIS *WEB* DENGAN MENGGUNAKAN METODE *CASE
BASED REASONING***

**Disusun dan diajukan oleh
ADITIO TRI NUGROHO
NPM 15670041**

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

Pembimbing I,

**Aris Tri Jaka Harjanta, M.Kom
NIDN 0619048202**

Pembimbing II,

**Khoiriyah Latifah, M.Kom
NIDN 0617077801**

SKRIPSI

**DIAGNOSA PENYAKIT YANG DISEBABKAN OLEH KELEBIHAN GULA
DARAH BERBASIS *WEB* DENGAN MENGGUNAKAN METODE *CASE
BASED REASONING***

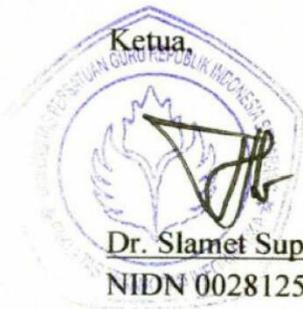
Disusun dan diajukan oleh

ADITIO TRI NUGROHO

NPM 15670041

**Telah dipertahankan di depan Dewan Penguji
Pada tanggal 30 Desember 2021
Dan dinyatakan telah memenuhi syarat**

Dewan Penguji



Dr. Slamet Supriyadi, M.Env.St.
NIDN 0028125901

Penguji I,



Bambang Agus H., S.Kom., M.Kom.
NIDN 0601088201

Sekretaris,



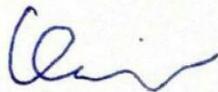
Bambang Agus H., S.Kom., M.Kom.
NIDN 0601088201

Penguji II,



Aris Trijaka Harjanta, S.Kom., M.Kom.
NIDN 0619048202

Penguji III,



Khoiriya Latifah, S.Kom., M.Kom.
NIDN 0617077801

MOTTO DAN PERSEMBAHAN

Motto :

1. Barang siapa yang menjaga, maka dia akan dijaga.
2. Bisa karena biasa, biasa akan menimbulkan kebiasaan.
3. Sedikit tapi membekas lebih baik, dari pada banyak namun hilang.
4. Usaha tidak pernah mengkhianati hasil.

Persembahan :

Kupersembahkan skripsi ini untuk :

Almamater Universitas PGRI Semarang

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini :

Nama : Aditio Tri Nugroho

NPM : 15670041

Prodi : Informatika

Fakultas : Teknik dan Informatika

Menyatakan dengan sebenarnya bahwa skripsi yang saya ini buat 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, 30 Desember 2021

Yang membuat pernyataan,



Aditio Tri Nugroho

NPM 15670041

Abstrak

Sistem pakar adalah salah satu cabang *Artificial Intelligence* yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar. Berdasarkan informasi *American Diabetes Association* (ADA) 2005, ada peningkatan drastis komplikasi penyakit diabetes sejak 2001 hingga 2004. Pada 2001, pasien DM beresiko mengalami penyakit hipertensi. Tahun 2001 38% pasien DM mengalami hipertensi, tahun 2004 mencapai 69% (meningkat 31%). Walaupun DM merupakan penyakit kronik yang tidak menyebabkan kematian secara langsung, tetapi dapat berakibat fatal bila pengelolannya tidak tepat dan kasus ini dapat di jadikan sebagai sistem pakar salah satunya mengidentifikasi penyakit kelebihan gula darah. Metode yang digunakan dalam sistem ini adalah metode *Case Based Reasoning*, *Case Based Reasoning* yaitu salah satu penyelesaian masalah, di mana masalah tersebut diselesaikan dengan melihat pola atau keadaan yang telah terjadi sebelumnya. Pengujian dilakukan menggunakan pengujian *black box* dengan keberhasilan pengujian 100% (layak digunakan), pengujian *white box* dengan hasil perhitungan kompleksitas siklomatis konsultasi adalah 2. Karena nilai tersebut kurang dari 10 berarti termasuk dalam algoritma yang tidak kompleks dan memenuhi kriteria rekayasa perangkat lunak. Berdasarkan *User Acceptance* yang dilakukan pada lima responden penguji. Aplikasi sistem pakar untuk mendiagnosa penyakit kelebihan gula darah ini memiliki rata-rata 88% maka dinyatakan layak pakai untuk pengguna.

Kata Kunci : Sistem Pakar, *Case Based Reasoning*, Penyakit Kelebihan Gula Darah.

PRAKATA

Puji syukur atas kehadiran Tuhan Yang Maha Esa karena atas limpahan rahmatnya, kami dapat menyelesaikan skripsi yang berjudul “Diagnosa Penyakit yang Disebabkan oleh Kelebihan Gula Darah Berbasis *Web* dengan Menggunakan Metode *Case Based Reasoning*” dengan lancar.

Penulisan skripsi ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Fakultas Teknik dan Informatika di Universitas PGRI Semarang. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, untuk itu kritik dan saran kami harapkan dari berbagai pihak.

Atas tersusunanya skripsi ini, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Muhdi S.H., M.Hum., selaku Rektor Universitas PGRI Semarang.
2. Dr. Slamet Supriyadi, M.Env.St., selaku Dekan Fakultas Teknik dan Informatika Universitas PGRI Semarang.
3. Bapak Bambang Agus Herlambang, S.Kom., M.Kom., selaku Ketua Program Studi Informatika.
4. Bapak Aris Tri Jaka Harjanta, S.Kom., M.Kom., selaku Dosen Pembimbing I yang telah memberikan bimbingan dan ilmu yang berarti hingga selesainya penulisan skripsi.
5. Ibu Khoiriya Latifah, S.Kom., M.Kom., selaku Dosen Pembimbing II yang telah mengarahkan penulis dengan penuh ketekunan dan kecermatan.
6. Seluruh Dosen Pengajar Informatika, Staff, dan Karyawan bagian Fakultas Teknik dan Informatika Universitas PGRI Semarang.
7. Orang tua tercinta dan keluarga yang tiada henti memberikan semangat dan selalu mendoakan.
8. KH. Muhaimin AH., dan keluarga yang telah memberikan doa dan berbagai pengalaman hidup yang sangat bermanfaat serta ilmu agama yang bermanfaat di dunia dan akhirat.
9. Saudaraku Pondok Pesantren Al Madinah, yang juga memberikan doa, semangat, dan dukungan.

10. Teman-teman angkatan 2015 Informatika Universitas PGRI Semarang yang selama ini memberikan warna di masa-masa menjadi mahasiswa.

Akhir kata, penulis berharap dan berdoa semoga skripsi ini dapat memberikan manfaat serta menambah wawasan dan ilmu pengetahuan bagi pembaca.

Semarang, 30 Desember 2021

Penulis,

Aditio Tri Nugroho

DAFTAR ISI

HALAMAN Sampul	i
HALAMAN PERSETUJUAN SKRIPSI.....	iii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Identifikasi Masalah	3
C. Rumusan Masalah	3
D. Batasan Masalah	3
E. Tujuan.....	4
F. Manfaat.....	4
BAB II KAJIAN PUSTAKA	5
A. Tinjauan Pustaka	5
B. Landasan Teori.....	7
1. Sistem Pakar	7
2. Metode <i>Case Based Reasoning</i> (CBR)	8
3. Algoritma <i>Manhattan Distance</i>	11
4. Gula Darah	11
5. Metode <i>Waterfall</i>	13
6. <i>Unified Modelling Language</i> (UML).....	14
7. <i>Website</i>	18
8. XAMPP	18
9. PhpMyAdmin	18
10. <i>Notepad++</i>	18
11. Pengujian <i>Black Box</i>	19
C. Kerangka Berpikir	19
BAB III METODE PENELITIAN	21
A. Pendekatan Penelitian.....	21

B.	Fokus Penelitian.....	21
C.	Desain Penelitian.....	21
D.	Teknik Pengumpulan Data	23
E.	Teknik Analisis Data.....	23
F.	Metode Pengembangan Sistem.....	23
BAB IV HASIL DAN PEMBAHASAN.....		28
A.	Hasil.....	28
1.	<i>Use Case Diagram</i>	28
2.	<i>Activity Diagram</i>	32
3.	<i>Sequence Diagram</i>	43
4.	<i>Class Diagram</i>	50
5.	Struktur Database.....	51
6.	Kamus Data.....	52
7.	<i>Design User Interface/Desain Antarmuka</i>	54
B.	Hasil Tahapan (Pembangunan).....	62
C.	Pengujian.....	72
1.	<i>White Box Testing</i>	72
2.	<i>Black Box Testing</i>	75
3.	<i>User Acceptance</i>	77
4.	Validasi Ahli.....	81
D.	Pembahasan.....	82
1.	Survei Sistem.....	82
2.	Analisis Sistem.....	82
3.	Implementasi Sistem.....	82
4.	Desain Sistem.....	83
5.	Analisa Permasalahan.....	83
E.	Contoh Kasus.....	87
BAB V PENUTUP.....		88
A.	Kesimpulan.....	88
B.	Saran.....	88
DAFTAR PUSTAKA		89

DAFTAR GAMBAR

2.1. Struktur Sistem Pakar.....	8
2.2. Ilustrasi Algoritma K-NN.....	9
2.3. Metode <i>Case Based Reasoning</i>	10
2.4. Proses Metode <i>Waterfall</i>	14
2.5. Use Case Diagram.....	15
2.6. Activity Diagram.....	16
2.7. Sequence Diagram.....	17
2.8. Class Diagram.....	17
2.9. Skema Kerangka Berpikir	20
3.1. Skema Desain Penelitian	22
4.1. Use Case Diagram.....	28
4.2. Activity Diagram Login.....	33
4.3. Activity Diagram Konsultasi.....	34
4.4. Activity Diagram Tambah Data Penyakit.....	35
4.5. Activity Diagram Ubah Data Penyakit.....	36
4.6. Activity Diagram Hapus Data Penyakit.....	36
4.7. Activity Diagram Cari Data Penyakit.....	37
4.8. Activity Diagram Tambah Data Gejala.....	38
4.9. Activity Diagram Edit Data Gejala.....	38
4.10. Activity Diagram Hapus Data Gejala.....	39
4.11. Activity Diagram Cari Data Gejala.....	40
4.12. Activity Diagram Tambah Data Basis Kasus.....	41
4.13. Activity Diagram Detail Data Basis Kasus.....	41
4.14. Activity Diagram Hapus Data Basis Kasus.....	42
4.15. Activity Diagram Cari Data Basis Kasus.....	43
4.16. Sequence Diagram Login Admin.....	44
4.17. Sequence Diagram Konsultasi.....	44
4.18. Sequence Diagram Tambah Data Penyakit.....	45

4.19. Sequence Diagram Edit Data Penyakit.....	45
4.20. Sequence Diagram Hapus Data Penyakit.....	46
4.21. Sequence Diagram Cari Data Penyakit.....	46
4.22. Sequence Diagram Tambah Data Gejala.....	47
4.23. Sequence Diagram Edit Data Gejala.....	47
4.24. Sequence Diagram Hapus Data Gejala.....	48
4.25. Sequence Diagram Cari Data Gejala.....	48
4.26. Sequence Diagram Tambah Data Basis Kasus.....	49
4.27. Sequence Diagram Detail Data Basis Kasus.....	49
4.28. Sequence Diagram Hapus Data Basis Kasus.....	49
4.29. Sequence Diagram Cai Data Basis Kasus.....	50
4.30. Class Diagram.....	50
4.31. Desain Beranda User.....	54
4.32. Desain Informasi Penyakit.....	54
4.33. Desain Halaman Konsultasi.....	55
4.34. Desain Hasil Diagnosa.....	55
4.35. Desain Login Admin.....	55
4.36. Desain Halaman Beranda Admin.....	56
4.37. Desain Halaman Data Penyakit.....	56
4.38. Desain Halaman Tambah Data Penyakit.....	57
4.39. Desain Halaman Edit Data Penyakit.....	57
4.40. Desain Hapus Data Penyakit.....	57
4.41. Desain Halaman Data Gejala.....	58
4.42. Desain Halaman Tambah Data Gejala.....	58
4.43. Desain Halaman Edit Data Gejala.....	59
4.44. Desain Hapus Data Gejala.....	59
4.45. Desain Halaman Data Basis Kasus.....	60
4.46. Desain Halaman Tambah Data Basis Kasus.....	60
4.47. Desain Halaman Detail Basis Kasus.....	61
4.48. Desain Hapus Data Basis Kasus.....	61
4.49. Desain Halaman Logout Sistem.....	61

4.50. Tampilan Halaman Beranda User.....	62
4.51. Tampilan Halaman Informasi Penyakit.....	63
4.52. Tampilan Halaman Konsultasi.....	63
4.53. Tampilan Halaman Hasil Diagnosa.....	64
4.54. Tampilan Halaman Login Admin.....	64
4.55. Tampilan Halaman Beranda Admin.....	65
4.56. Tampilan Halaman Data Penyakit.....	65
4.57. Tampilan Halaman Tambah Data Penyakit.....	66
4.58. Tampilan Halaman Edit Data Penyakit.....	66
4.59. Tampilan Halaman Hapus Data Penyakit.....	67
4.60. Tampilan Halaman Data Gejala.....	67
4.61. Tampilan Halaman Tambah Data Gejala.....	68
4.62. Tampilan Halaman Edit Data Gejala.....	68
4.63. Tampilan Halaman Hapus Data Gejala.....	69
4.64. Tampilan Halaman Data Basis Kasus.....	69
4.65. Tampilan Halaman Tambah Data Basis Kasus.....	70
4.66. Tampilan Halaman Detail Basis Kasus.....	70
4.67. Tampilan Halaman Hapus Data Basis Kasus.....	71
4.68. Tampilan Halaman Logout Sistem.....	71
4.69. Basis Path Konsultasi.....	74
4.70. Alur Aplikasi Sistem Pakar Penyakit Gula Darah.....	84

DAFTAR TABEL

4.1.Deskripsi use case diagram login.....	29
4.2.Deskripsi use case diagram konsultasi.....	29
4.3.Deskripsi kelola data penyakit.....	30
4.4.Deskripsi kelola data gejala.....	31
4.5.Deskripsi kelola data basis kasus.....	32
4.6.Struktur database admin.....	51
4.7.Struktur database gejala.....	51
4.8.Struktur database penyakit.....	51
4.9.Struktur database basis kasus.....	52
4.10. Skrip hasil konsultasi.....	72
4.11. Pengujian <i>black box testing</i>	76
4.12. <i>User Acceptance</i> dari segi kemanfaatan admin.....	78
4.13. <i>User Acceptance</i> dari segi kemanfaatan pasien.....	79
4.14. Kemudahan Pengguna.....	80
4.15. Validasi ahli.....	81
4.16. Kode Data Gejala.....	85
4.17. Kode Data Penyakit.....	86
4.18. Representasi.....	86
4.19. Data lama pasien.....	87
4.20. Hasil Perhitungan dengan <i>Manhattan Distance</i>	87

BAB I

PENDAHULUAN

A. Latar Belakang

Pengembangan teknologi saat ini banyak mengalami perubahan, terutama teknologi komputer khususnya di Indonesia. Pemanfaatan teknologi ini bukan hanya memberikan informasi di dunia tertentu saja, bahkan dapat digunakan oleh masyarakat luas. Hal ini mendorong para ahli untuk semakin mengembangkan komputer agar membantu kinerja manusia. Salah satu pengembangannya yaitu sebuah sistem pakar.

Sistem pakar adalah salah satu cabang *Artificial Intelligence* yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar. Seorang pakar adalah orang yang mempunyai keahlian dalam bidang tertentu, yaitu pakar yang mempunyai *knowledge* atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya.[1] Saat ini sistem pakar sudah banyak diterapkan diberbagai bidang, salah satunya di bidang kesehatan.

Sistem pakar dalam bidang kesehatan dapat membantu aktivitas para dokter sebagai asisten yang berpengalaman dan mengkombinasikan kaidah-kaidah penarikan kesimpulan (*inference rules*) dengan basis pengetahuan tertentu yang diberikan oleh satu atau lebih pakar (dokter) dalam bidang kesehatan. Bidang kesehatan adalah suatu bidang yang membutuhkan bantuan teknologi guna meringankan para dokter untuk mengidentifikasi atau mendiagnosa suatu penyakit. Dalam kasus ini sistem pakar yang akan dibuat penulis yaitu sistem pakar untuk mendiagnosa penyakit kelebihan gula darah menggunakan metode *case based reasoning*. Untuk menangani masalah efektifitas dan akurasi dalam mendeteksi penyakit kelebihan gula darah.

Menurut International Diabetes Federation-7 tahun 2015, dalam metabolisme tubuh hormon insulin bertanggung jawab mengatur kadar glukosa darah. Hormon ini diproduksi dalam pancreas kemudian dikeluarkan untuk

digunakan sebagai sumber energi. Apabila di dalam tubuh kekurangan hormon insulin maka dapat menyebabkan hiperglikemia.

Data dari berbagai studi global menyebutkan bahwa penyakit Diabetes Mellitus adalah masalah kesehatan yang besar. Hal ini dikarenakan adanya peningkatan jumlah penderita diabetes dari tahun ke tahun. Pada tahun 2015 menyebutkan sekitar 415 juta orang dewasa memiliki diabetes, kenaikan 4 kali lipat dari 108 juta di tahun 1980an. Apabila tidak ada tindakan pencegahan maka jumlah ini akan terus meningkat tanpa ada penurunan. Diperkirakan pada tahun 2040 meningkat menjadi 642 juta penderita.[20]

Case Based Reasoning adalah salah satu penyelesaian masalah, di mana masalah tersebut diselesaikan dengan melihat pola atau keadaan yang telah terjadi sebelumnya.[2] Pemanfaatan metode *Case Based Reasoning* (CBR) dapat menghasilkan suatu aplikasi yang mengidentifikasi penyakit kelebihan gula darah. Dalam metode *case based reasoning* terdapat empat tahapan proses yaitu : *Retrieve, Reuse, Revise, Retain*. Didalam tahap *reuse* terdapat algoritma *K-Nearest Neighbor* yang berguna untuk menghitung jarak data terdekat dengan data sebelumnya. Algoritma *Nearest Neighbor* (K-NN) merupakan algoritma klasifikasi berdasarkan kedekatan jarak suatu data dengan data yang lain. Pada algoritma *Nearest Neighbor*, data berdimensi q , jarak dari data tersebut ke data yang lain dapat dihitung. Nilai jarak inilah yang digunakan sebagai nilai kedekatan/kemiripan antara data uji dengan data latih. Nilai K pada *Nearest Neighbor* berarti K - data terdekat dari data uji.[3]

Berdasarkan penelitian sebelumnya bahwa kelebihan gula darah dapat mengakibatkan suatu penyakit yang serius seperti *Diabetes*. Dalam metode *Case Based Reasoning* terdapat algoritma *nearest neighbor* yang digunakan untuk membantu dalam mengklasifikasi kedekatan data sebelumnya dengan data yang baru. Dengan menghitung tingkat kemiripan (jarak) suatu kasus terhadap kasus baru berdasarkan beberapa atribut yang didefinisikan, berdasar pembobotan tertentu dan kemudian tingkat kemiripan (jarak) dari keseluruhan atribut akan dijumlahkan.[4]

Berdasarkan latar belakang tersebut, diharapkan sistem pakar diagnosa penyakit yang disebabkan oleh gula darah dapat dilakukan dengan mudah. Untuk itu, dilakukan penelitian dengan judul “ Diagnosa Penyakit yang Disebabkan oleh Kelebihan Gula Darah Berbasis *Web* dengan Menggunakan Metode *Case Based Reasoning*”.

B. Identifikasi Masalah

Permasalahan penelitian dapat diidentifikasi sebagai berikut :

1. Keterbatasan jumlah dokter dalam menangani berbagai macam penyakit.
2. Waktu yang terbatas dalam menangani jumlah pasien yang terus meningkat.

C. Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dijabarkan rumusan masalah sebagai berikut :

Bagaimana merancang dan membuat sistem pakar diagnosa penyakit yang disebabkan oleh kelebihan gula darah dengan metode *case based reasoning*?

D. Batasan Masalah

Mengingat terbatasnya tenaga dan waktu yang ada, serta menjaga dan menghindari pembahasan masalah yang terlalu luas, maka penulis membatasi permasalahan meliputi :

1. Sistem pakar ini hanya menampilkan 6 penyakit dalam data yaitu *Diabetes*, *Jantung*, dan *Obesitas*, *Hiperglikemia*, *Kanker Pankreas*, dan *Darah Tinggi*.
2. Data sistem pakar ini diperoleh dari klinik Pratama Umum Undaan Kudus tahun 2017.
3. Sistem pakar ini hanya sampai tahap pengujian sistem.

E. Tujuan

Berdasarkan rumusan masalah, maka tujuan dari penelitian ini adalah :

1. Merancang sebuah sistem pakar yang dapat mempermudah dokter dalam mengetahui jenis penyakit.
2. Melakukan pengujian aplikasi sehingga dapat diterapkan dengan baik dan berguna.

F. Manfaat

Manfaat yang didapatkan dari penyusunan skripsi ini adalah :

1. Bagi Mahasiswa
Menambah wawasan pengetahuan sesuai dengan program studi serta menambah bekal pengetahuan yang dapat dipergunakan untuk persiapan dalam menghadapi dunia kerja.
2. Bagi Akademik
Sebagai tolak ukur keberhasilan akademik dalam mendidik dan memberikan ilmu sebagai bekal untuk terjun ke masyarakat. Laporan yang dihasilkan diharapkan dapat berguna bagi referensi di perpustakaan. Sebagai bahan evaluasi akademik untuk meningkatkan mutu pendidikan dan pembelajaran di Fakultas Teknik dan Informatika Universitas PGRI Semarang.
3. Bagi Pakar
Mempermudah kinerja dokter dalam memeriksa penyakit pasien dan juga mempersingkat waktu.

BAB II KAJIAN PUSTAKA

A. Tinjauan Pustaka

Tinjauan pustaka yang dipergunakan dalam penelitian disajikan melalui beberapa pendapat terdahulu dengan metode *case based reasoning* :

No	Nama	Tahun	Judul	Hasil dan Kesimpulan
1.	Dito Putro Utomo, Surya Darma Nasution	2016	Sistem Pakar Mendeteksi Kerusakan Toner dengan Menggunakan Metode <i>Case Based Reasoning</i>	<ul style="list-style-type: none"> • Sistem pakar ini dibuat untuk memudahkan dalam mendeteksi penyakit kerusakan Toner didapat nilai kemiripan antara kasus yang lama dan kasus yang baru sebesar 0,67. • Sedangkan penulis membuat sistem pakar untuk mendeteksi penyakit kelebihan gula darah.
2.	Sri Wahyuni Nasution, Nelly Astuti Hasibuan, Putri Ramadhani.	2017	Sistem Pakar Diagnosa Anoreksia Nervosa Menerapkan Metode <i>Case Based Reasoning</i>	<ul style="list-style-type: none"> • Sistem pakar ini dibuat untuk mendiagnosa penyakit Anoreksia Nervosa menggunakan metode <i>case based reasoning</i>. • Sedangkan penulis membuat sistem pakar untuk mendiagnosa penyakit kelebihan gula darah menggunakan metode <i>case based reasoning</i>.
3.	Annahl Riadi	2017	Penerapan Metode <i>Certainty Factor</i> untuk Sistem Pakar Diagnosa Penyakit Diabetes Melitus pada RSUD Bumi Panua Kabupaten Pohuwato	<ul style="list-style-type: none"> • Sistem pakar ini dibuat untuk mendiagnosa penyakit Diabetes Melitus menggunakan metode <i>Certainty Factor</i>. • Sedangkan penulis membuat sistem pakar untuk

				mendiagnosa kelebihan gula darah dengan metode <i>Case Based Reasoning</i>.
4.	Faza Akmal, Sri Winiarti	2014	Sistem Pakar untuk Mendiagnosa Penyakit Lambung dengan Implementasi Metode CBR (<i>Case Based Reasoning</i>) Berbasis Web	<ul style="list-style-type: none"> • Sistem pakar ini dibuat untuk mendiagnosa penyakit lambung berbasis Web. • Sedangkan penulis membuat sistem pakar untuk mendiagnosa penyakit kelebihan gula darah berbasis Web.

Sistem pakar merupakan suatu sistem komputer yang dirancang agar dapat melakukan penalaran seperti layaknya seorang pakar pada suatu bidang keahlian tertentu. Sistem pakar diciptakan bukan untuk menggantikan kedudukan seorang pakar tetapi untuk memasyarakatkan pengetahuan dan pengalaman pakar tersebut untuk orang banyak.[5]

Dalam penelitian ini dikembangkan sistem pakar untuk mendiagnosa penyakit akibat gangguan makan dengan menggunakan metode *Case Based Reasoning*. Metode *Case Based Reasoning* melakukan proses penyelesaian masalah dengan memanfaatkan pengalaman sebelumnya. Metode ini memiliki kelebihan dibandingkan dengan sistem berbasis aturan (*rule base system*) dalam hal pengetahuan yang terletak pada kumpulan pengalaman/kasus-kasus sebelumnya. Metode ini juga tetap dapat melakukan penalaran walaupun terdapat data yang tidak lengkap. Perancangan *Case Based Reasoning* ini bisa digunakan untuk membantu pakar dalam mengidentifikasi penyakit dan memberikan cara penanggulangannya. Hal ini berarti tidak menggantikan kedudukan seorang pakar tetapi hanya membantu dalam mengkonfirmasi keputusannya, karena mungkin bisa terdapat alternatif yang harus dipilih secara tepat.[6]

Penelitian ini umumnya ditujukan pada permasalahan yang cenderung ingin mendapatkan solusi dari hasil berupa konsultasi, diagnosis, estimasi, prediksi dan sejenisnya. Seperti halnya penerapan dalam dunia kedokteran

ataupun dunia medis, konsultasi, dan diagnosis sangatlah diandalkan karena hasil diagnosis ini dapat mengantisipasi beberapa jenis penyakit dengan tepat dan cepat. Demikian halnya dengan Rumah Sakit yang tentu sangat mengharapkan adanya area penambahan nilai, produktivitas serta managerial dengan dukungan hasil diagnosis penyakit *Diabetes Melitus* dengan tepat dan akurat sehingga kerugian dapat dicegah.[7]

Dari hasil penelitian Divisi *Gastreenterologi* Departemen Ilmu Penyakit Dalam FKUI, ditemukan penderita yang mengalami gangguan pencernaan di Indonesia selama tahun 2009 sebanyak 86,4% disebabkan oleh gastritis, 12,59% terdapat ulkus dan 1% disebabkan oleh kanker lambung. Berdasarkan catatan rekamedis di RS PKU Muhammadiyah Yogyakarta menunjukkan bahwa penderita penyakit lambung selama tahun 2011 mencapai angka 257 kasus. Perusahaan obat PT Kalbe Farma Tbk. mengadakan survei yang dilakukan di Jakarta tahun 2007 yang melibatkan 1645 responden diperoleh informasi bahwa pasien dengan masalah sakit maag ini mencapai angka 60 %.[8]

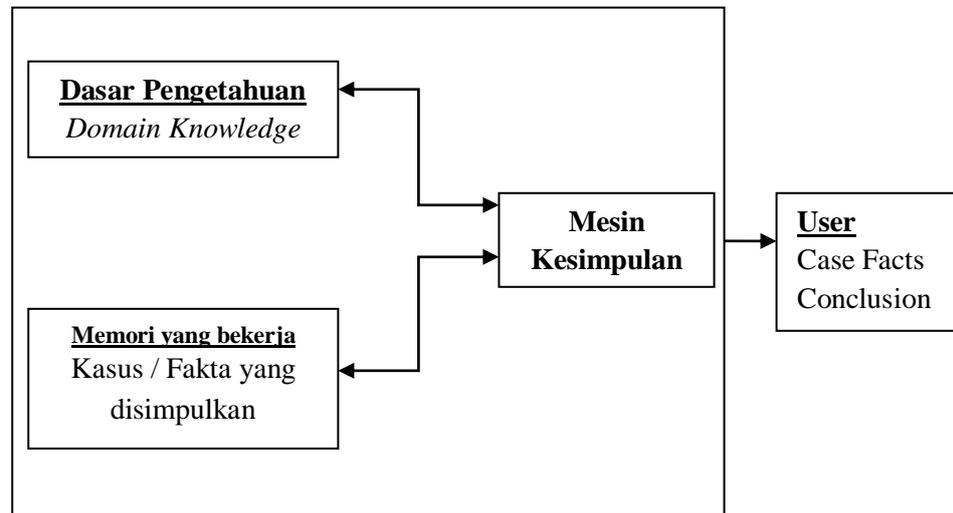
B. Landasan Teori

1. Sistem Pakar

Secara umum, sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar.

Dalam penyusunannya, sistem pakar mengkombinasikan kaidah-kaidah penarikan kesimpulan atau *inference rules* dengan basis pengetahuan tertentu yang diberikan oleh satu atau lebih pakar dalam bidang tertentu. Kombinasi dari kedua hal tersebut disimpan dalam komputer, yang selanjutnya digunakan dalam proses pengambilan keputusan untuk penyelesaian masalah tertentu. Komponen utama pada struktur sistem pakar meliputi Basis Pengetahuan / *Knowledge Base*, Mesin Inferensi / *Inference Engine*, *Working Memory*, dan Antarmuka

Pemakai / *User Interface*. Struktur sistem pakar dapat ditunjukkan pada Gambar 2.1.[9]



Gambar 2.1. Struktur Sistem Pakar

2. Metode *Case Based Reasoning* (CBR)

Case Based Reasoning merupakan sistem yang bertujuan untuk menyelesaikan suatu kasus baru dengan cara mengadaptasi solusi-solusi yang terdapat pada kasus-kasus sebelumnya yang mirip dengan kasus baru tersebut. Sistem penalaran komputer berbasis kasus menurut Riesbeck dan Schank merupakan sebuah penalaran berbasis kasus memecahkan masalah dengan menggunakan atau mengadaptasi solusi kasus lama.[10]

Ada 4 tahapan proses yang ada di dalam sistem penalaran komputer berbasis kasus, yaitu:

1. *Retrieve* (Memperoleh Kembali)

Pada proses ini sistem akan melakukan identifikasi parameter pencocokan yang dapat dijadikan sebagai acuan lalu melakukan pencarian kasus lama yang memiliki kesamaan dengan kasus baru.

2. *Reuse* (Menggunakan)

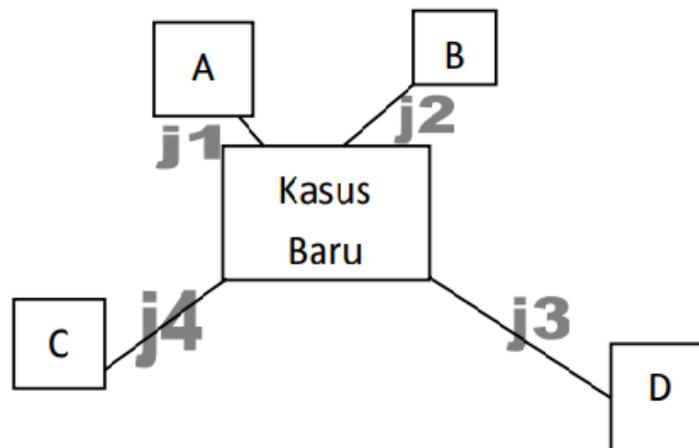
Pada proses ini sistem akan menggunakan kembali informasi yang berasal dari kasus sebelumnya atau sistem akan melakukan adaptasi

terlebih dahulu untuk memecahkan masalah pada kasus yang baru. Pada tahap *Reuse* terdapat algoritma *K- Neirest Neighbor* (K-NN) untuk mengklasifikasikan obyek baru berdasarkan atribut, training *sample* dan juga menggunakan Kemiripan (*Similarity*).

Adapun penerapan metode K-NN melalui beberapa langkah:

- a) Tentukan parameter k .
- b) Hitung jarak antara data yang akan dievaluasi dengan semua pelatihan.
- c) Urutkan jarak yang terbentuk (urut naik).
- d) Tentukan jarak terdekat sampai urutan k .
- e) Pasangkan kelas yang bersesuaian.
- f) Cari jumlah kelas dari tetangga yang terdekat dan tetapkan kelas tersebut sebagai kelas data yang akan dievaluasi.[11]

Berikut adalah gambaran ilustrasi algoritma K-NN :



Gambar 2.2. Ilustrasi Algoritma K-NN[12]

Dari gambar ilustrasi di atas dapat dijelaskan bahwa ada terdapat 4 pasien lama yang disimbolkan oleh (A, B, C dan D). Ketika ada pasien baru maka yang diambil solusi adalah solusi dari kasus pasien lama yang memiliki kedekatan terbesar. Misal j_1 adalah jarak antara pasien baru dengan pasien A, j_2 adalah jarak antara pasien baru dengan pasien B, j_3 adalah jarak antara pasien baru

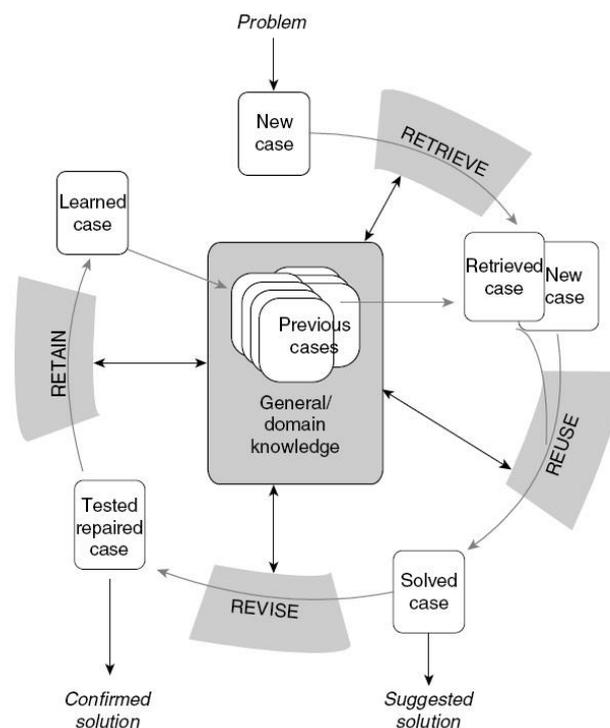
dengan pasien C, j_4 adalah jarak antara pasien baru dengan pasien D. Dari ilustrasi gambar terlihat bahwa j_1 yang paling terdekat dengan kasus baru. Dengan demikian maka solusi dari kasus pasien A yang akan digunakan sebagai solusi dari pasien baru tersebut.

3. *Revise* (Meninjau)

Pada proses ini sistem akan meninjau kembali solusi yang telah didapatkan dari kasus yang lama.

4. *Retain* (Menyimpan)

Pada proses ini apabila ternyata ditemukan solusi baru yang lebih baik dari solusi yang telah ada sebelumnya maka solusi baru tersebut akan diberi indeks dan disimpan untuk kemudian digunakan kembali pada kasus serupa pada masa yang akan datang.[13]



Gambar 2.3. Metode *Case Based Reasoning*. [14]

3. Algoritma *Manhattan Distance*

Manhattan Distance atau sering juga disebut *city block distance*, merupakan sebuah nama pada kota *Manhattan* yang tersusun menjadi blok-blok, maka dari itu sering disebut *city block distance*. *Manhattan distance/city block distance* merupakan algoritma perhitungan yang digunakan untuk menentukan ukuran kedekatan (*similarity*) antar kasus. Dimana rumus *Manhattan distance/city block distance* adalah sebagai berikut :

$$Sim(A, B) = \frac{\sum_{i=1}^p sim_i(a, b)}{p} \quad (1)$$

Keterangan :

A : Kasus Lama
 B : Kasus Baru
 p : Jumlah atribut dalam setiap kasus
 i : Atribut individu dari 1 sampai p
 $sim_i(a, b)$: Fungsi kemiripan untuk atribut I dalam kasus A dan B

Dimana urutan tingkat nilai kemiripan (*similarity*) yang paling tinggi dengan *range* 0 sampai 1. Maka kasus yang memiliki kemiripan paling tinggi dengan kasus baru yang akan disarankan sebagai solusi.

4. Gula Darah

Konsumsi energi yang melebihi kebutuhan tubuh menyebabkan lebih banyak glukosa yang ada dalam tubuh. Gula merupakan sumber makanan dan bahan bakar bagi tubuh yang berasal dari proses pencernaan makanan. Pada penderita *Diabetes Mellitus* (DM) tipe-2, jaringan tubuhnya tidak mampu untuk menyimpan dan menggunakan glukosa, sehingga kadar glukosa darah akan naik dan akan menjadi racun bagi tubuh. Tingginya kadar glukosa darah dipengaruhi oleh tingginya asupan energi dari makanan.

Hasil analisis univariat, diketahui asupan energi pasien DM yang baik 31,6% sedangkan dengan asupan energi kurang baik 68,4%.

Konsumsi energi kurang baik yang melebihi asupan energi yang baik ini cukup mengkuatirkan bagi tubuh pasien mengingat prinsip makanan penderita DM harus tepat jumlah (energi) yang dikonsumsinya.

Pada analisis bivariat diketahui pasien dengan asupan energi yang baik semuanya mengalami penurunan kadar gula darah. Sedangkan pasien dengan asupan energi kurang baik yang mengalami penurunan kadar gula darah (61,5%) dan dengan kenaikan kadar gula darah sebanyak 15 orang (38,5%).

Berdasarkan hasil analisa statistik menunjukkan ada hubungan bermakna antara asupan energi dengan kadar gula darah pasien *diabetes mellitus* dengan nilai $P = 0,001$, hal ini sesuai dengan pendapat bahwa tingginya kadar gula darah dipengaruhi oleh tingginya konsumsi energi.

Pada pasien *diabetes mellitus* tidak diajukan asupan energi lebih ataupun kurang dari kebutuhan , karena pada saat asupan energi kurang dapat terjadi *hipoglikemi*, pada saat asupan energi lebih dapat menyebabkan *hiperglikemia* dengan gejala keton yang meningkat dalam darah, serta gejala khas DM seperti poliuria, olidipsi, dan poliphagia yang dapat mengurangi kualitas hidup pasien.[15] Berikut macam-macam penyakit gula darah:

1. *Diabetes*

Diabetes merupakan penyakit menahun yang dapat diderita seumur hidup.

2. Penyakit Jantung

Penyakit jantung merupakan gangguan yang terjadi pada sistem pembuluh darah besar. Sehingga menyebabkan jantung dan peredaran darah tidak bekerja sebagaimana mestinya.

3. Obesitas

Obesitas dapat diartikan sebagai akumulasi lemak secara berlebihan atau abnormal dalam tubuh yang dapat mengganggu kesehatan.

4. Hiperglikemia

Hiperglikemia adalah Kondisi ketika kadar glukosa (gula darah) berada di bawah normal. Kadar gula darah saat ini mungkin kurang dari 60 mg/dl.

5. Darah Tinggi

Darah Tinggi adalah Kondisi saat tekanan darah berada pada nilai 130/80 mmHg atau lebih. Kondisi ini dapat menjadi berbahaya, karena jantung dipaksa memompa darah lebih keras ke seluruh tubuh.

6. Kanker Pankreas

Kanker Pankreas adalah Penyakit yang disebabkan oleh tumbuhnya tumor di dalam pancreas. Pankreas adalah sebuah kelenjar besar yang merupakan bagian dari sistem pencernaan dan memiliki panjang sekitar 15 cm.

7. Metode *Waterfall*

Metode penelitian yang diterapkan pada penelitian ini adalah dengan pengembangan metode *waterfall*. Metode *waterfall* merupakan model pengembangan sistem informasi yang sistematis dan sekuensial. Metode *Waterfall* memiliki tahapan-tahapan sebagai berikut :

a) Analisis Kebutuhan (*Requirements Analysis and Definition*)

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

b) Desain Sistem (*System and Software Design*)

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

c) Implementasi (*Implementation and Unit Testing*)

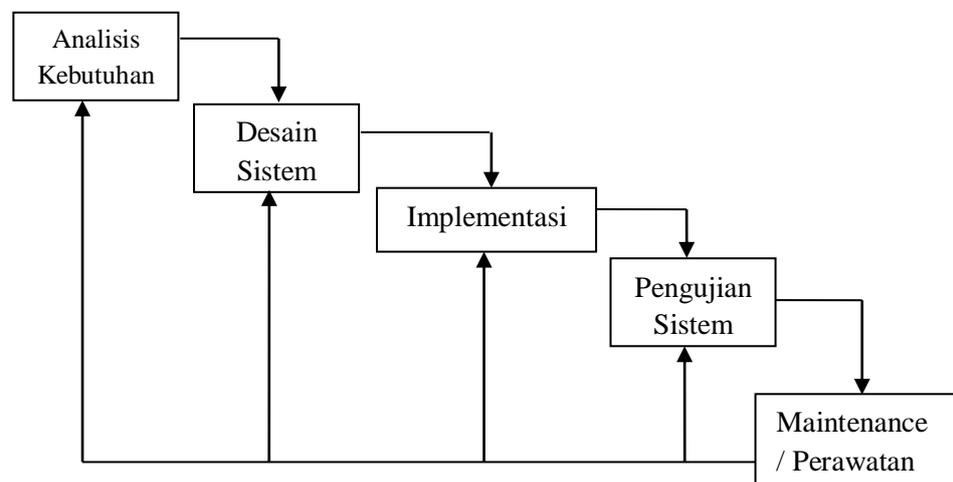
Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

d) Pengujian Sistem (*Integration and System Testing*)

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke *customer*

e) *Maintenance* / Perawatan

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya, meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru. [16]



Gambar 2.4. Proses Metode *Waterfall*. [17]

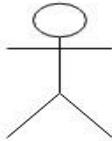
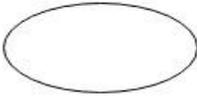
8. *Unified Modelling Language (UML)*

UML (*Unified Modelling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan,

membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem *software*. Untuk membuat suatu model, UML memodifikasikan diagram-diagram berikut ini :

a. *Use case Diagram*

Use case diagram menggambarkan sekelompok *use case* dan aktor yang disertai dengan hubungan diantaranya. Diagram ini menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Simbol yang digunakan dalam *use case diagram* adalah :

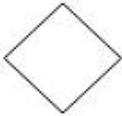
Relasi	Keterangan	Notasi Keterangan
Aktor		Aktor merupakan semua yang ada di luar ruang lingkup sistem perangkat lunak dan berinteraksi dengan sistem perangkat lunak tersebut.
Asosiasi (Association)		Lintasan komunikasi antara actor dengan use case.
Use Case		Fungsi-fungsi atau fitur-fitur apa saja yang disediakan oleh sistem informasi yang akan di bangun kepada pengguna.

Gambar 2.5. *Use case diagram*

b. *Activity Diagram*

Activity diagram merupakan diagram yang menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin

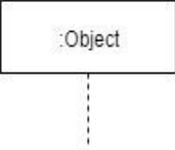
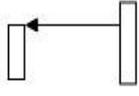
terjadi dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses *pararel* yang mungkin terjadi pada beberapa eksekusi.

Relasi	Keterangan	Notasi Keterangan
Aktiviti		Perilaku objek yang dilakukan saat objek berada dalam state tertentu
Decision		<i>Decision</i> digunakan sebagai pilihan untuk pengambilan keputusan
Asosiasi (Association)		Apa yang menghubungkan antar objek satu dengan objek yang lainnya.
Start state		<i>Start state</i> digunakan untuk memulai diagram statechart.
End state		<i>End state</i> digunakan untuk mengakhiri diagram.

Gambar 2.6. *Activity diagram*

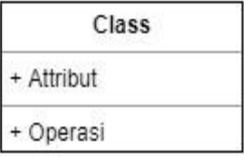
c. *Sequence Diagram*

Sequence diagram menjelaskan interaksi antar objek di dalam dan di sekitar sistem berupa pesan (*message*) yang disusun dalam suatu urutan waktu yaitu urutan kejadian yang dilakukan oleh seorang aktor dalam menjalankan sistem. Diagram ini khusus berasosiasi dengan *use case*. Diagram ini menunjukkan bagaimana detail operasi dilakukan, pesan apa yang dikirim dan kapan terjadinya. *Sequence diagram* terdiri atas dimensi vertikal yaitu waktu dan dimensi horizontal yaitu menggambarkan objek-objek yang berkaitan.

NO.	Gambar	Nama	Keterangan
1.		<i>Life Line</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

Gambar 2.7. *Sequence diagram*d. *Class Diagram*

Class diagram adalah diagram yang mendeskripsikan kelas/objek dan menggambarkan relasi/hubungan antar kelas/objek. *Class diagram* menggunakan notasi kelas dan notasi relasi antar kelas. Simbol yang digunakan dalam *class diagram* sebagai berikut :

Relasi	Notasi	Notasi Keterangan
Kelas		Kelas adalah kategori yang membungkus informasi dan perilaku.
<i>Asosiasi</i> (Association)		Apa yang menghubungkan antar objek satu dengan objek yang lainnya.

Gambar 2.8. *Class diagram*

9. Website

Website adalah suatu dokumen berupa kumpulan halaman *web* yang saling terhubung dan isinya terdiri dari berbagai informasi berbentuk teks, suara, gambar, video, dan lainnya, dimana semua data tersebut disimpan pada server hosting.

10. XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database dan penerjemahan bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, dan Perl. Program ini tersedia dalam General Public License (GNU) dan bebas, merupakan *web server* yang mudah digunakan dan dapat melayani tampilan halaman *web* yang dinamis

11. PhpMyAdmin

PhpMyAdmin merupakan aplikasi berbasis *web* yang dikembangkan menggunakan bahasa pemrograman PHP. Melalui PhpMyAdmin user dapat melakukan perintah *query* tanpa harus mengetikkan seperti pada MS DOS. Perintah tersebut misalnya administrasi *user* dan *privileges*, *export*, dan *import database*, manajemen *database*, manajemen tabel dan struktur tabel, dan sebagainya. PhpMyAdmin sangat *user friendly*, sehingga mudah untuk digunakan walaupun pengguna baru.

12. Notepad++

Notepad++ adalah suatu text editor yang berjalan pada Operating System (OS) Windows.

13. Pengujian *Black Box*

Pengujian *Black Box* berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetestan pada spesifikasi fungsional program.

Black Box bukanlah solusi alternatif dari *White Box* tapi lebih merupakan pelengkap untuk menguji hal-hal yang dicakup oleh *White Box*.

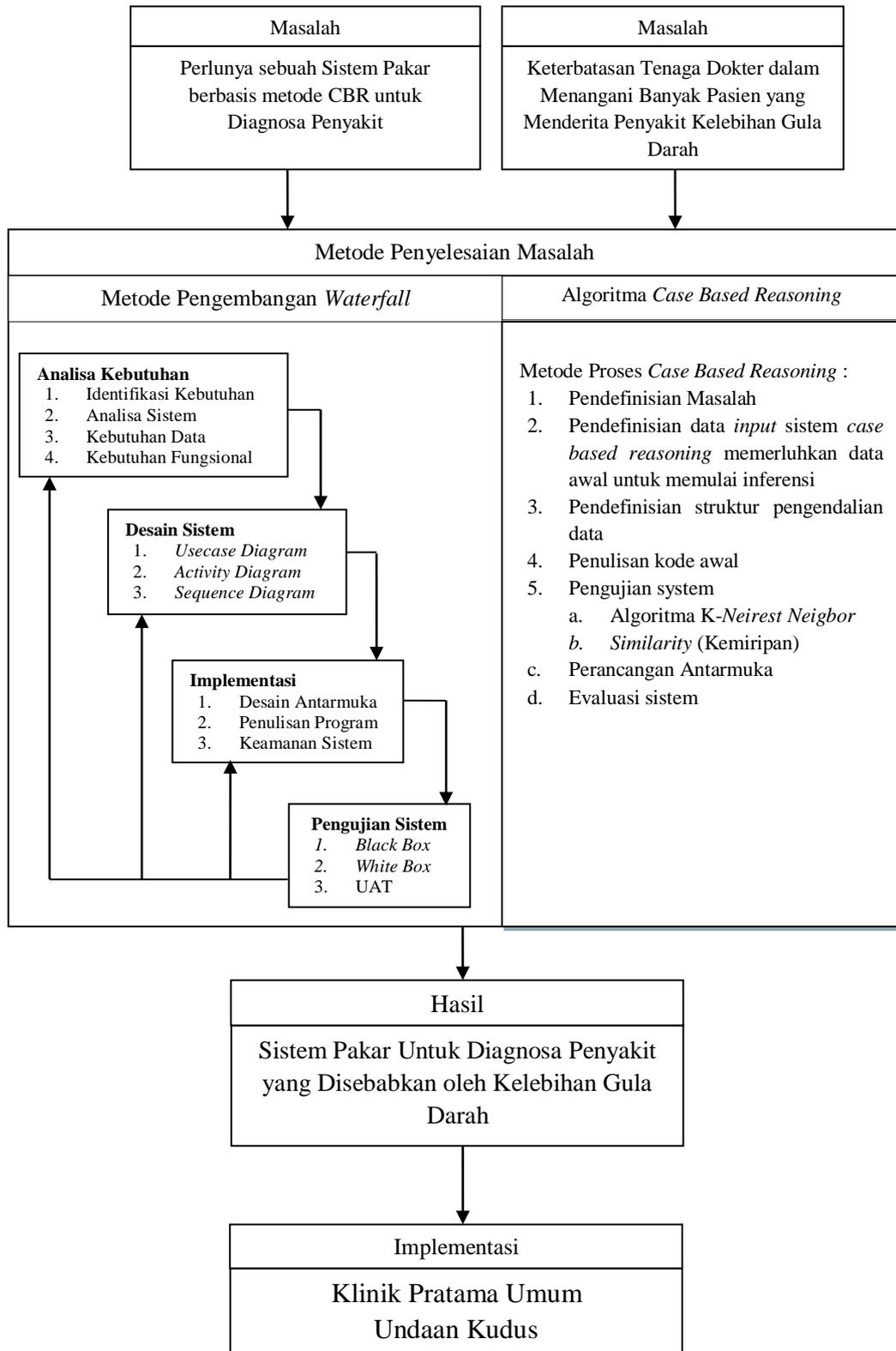
Black Box cenderung untuk menemukan hal-hal berikut:

- 1) Fungsi yang tidak benar atau tidak ada.
- 2) Kesalahan antarmuka (*interface errors*).
- 3) Kesalahan pada struktur data dan akses basis data.
- 4) Kesalahan performansi (*performance errors*).
- 5) Kesalahan inisialisasi dan terminasi.[18]

C. Kerangka Berpikir

Kerangka berpikir merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah yang penting.

Pengembangan teknologi semakin canggih saja bagaimana tidak saat ini di berbagai daerah khususnya di Indonesia sudah banyak memakai yang namanya teknologi salah satunya dalam bidang kesehatan. Dalam bidang kesehatan masih membutuhkan tenaga untuk memudahkan proses pemeriksaan pasien dan mempersingkat waktu juga karena keterbatasan tenaga dokter dan staf-staf perawat lainnya. Hal ini mendorong para ahli komputer untuk membuat sebuah sistem pakar untuk mendiagnosa penyakit yang menggunakan metode *case based reasoning*. Maka akan dihasilkan sebuah sistem pakar untuk diagnosa penyakit kelebihan gula darah. Berikut untuk lebih jelasnya skema kerangka berpikir dapat dilihat pada Gambar 2.9.



Gambar 2.9. Skema Kerangka Berpikir

BAB III

METODE PENELITIAN

A. Pendekatan Penelitian

Dalam pembangunan suatu aplikasi diperlukan suatu pendekatan dan pengembangan sistem yang akan menentukan proses penyelesaian rekayasa perangkat lunak, adapun pendekatan sistem yang digunakan adalah dengan menggunakan pendekatan berorientasi objek.

Metode pendekatan sistem yang digunakan adalah pendekatan terstruktur dengan menggunakan UML (*Unified Modeling Language*).

B. Fokus Penelitian

Fokus penelitian yaitu masyarakat yang sedang menderita penyakit gula darah. Karena saat ini banyak masyarakat sering mengkonsumsi glukosa yang berlebihan tanpa memperdulikan dampaknya.

C. Desain Penelitian

Desain penelitian merupakan tahapan yang akan dilakukan dalam melakukan penelitian supaya untuk memudahkan peneliti dalam melakukan penelitian dan dibutuhkan desain penelitian.

Langkah-langkah yang akan dilakukan untuk melakukan penelitian skripsi ini adalah sebagai berikut:

1. Pengumpulan Data Penelitian

Pengumpulan data penelitian yaitu mengumpulkan data-data yang dibutuhkan untuk membuat aplikasi sistem pakar. Pengumpulan data penelitian dilakukan dengan dua cara yaitu :

a. Wawancara

Pengumpulan data melalui wawancara akan memberikan data yang nyata berupa keluhan atau gejala yang sedang dialami oleh pasien.

b. Studi Literatur

Pengumpulan data melalui studi literatur guna untuk mencari data-data dari hasil penelitian terdahulu.

2. Desain Tampilan dan Sistem Penelitian

Mendesain tampilan dan sistem dilakukan agar *user* mudah memahami dan mengoperasikan aplikasi tersebut.

3. Implementasi Penelitian

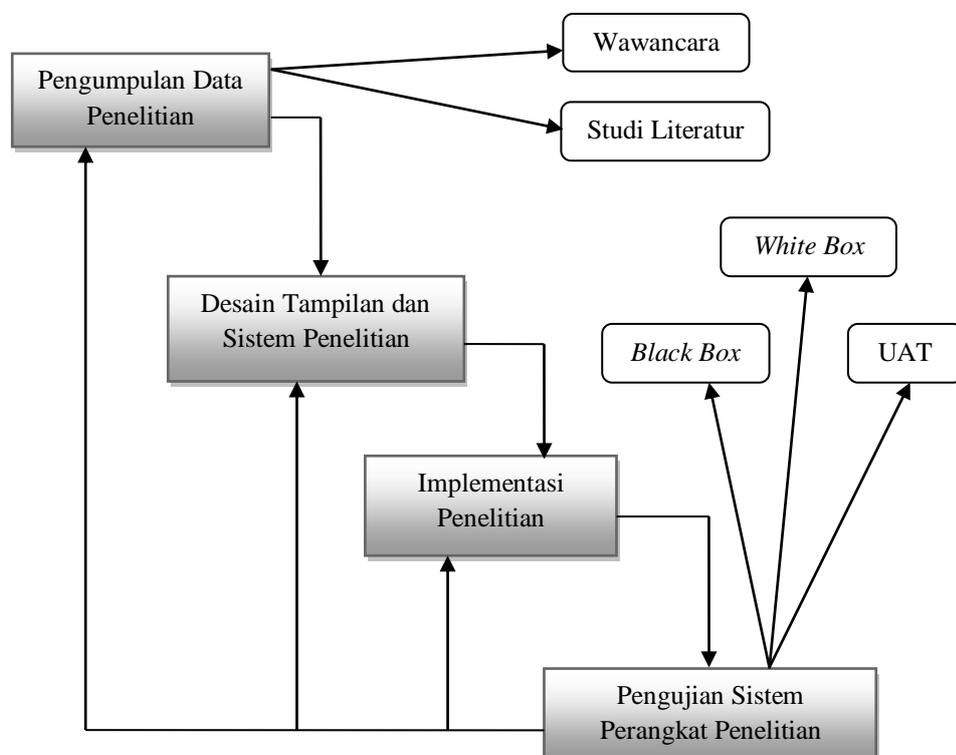
Pada proses implementasi sistem hal yang perlu dilakukan yaitu :

- a. Melakukan *login* oleh *user*.
- b. Menginput keluhan atau gejala untuk *user*.
- c. Menampilkan solusi pengobatan atau saran.

4. Pengujian Sistem Perangkat Penelitian

Pengujian sistem dilakukan menggunakan *White Box*, *Black Box*, dan *User Acceptance* agar aplikasi memang siap digunakan dan tidak terjadi suatu kendala sewaktu-waktu saat digunakan.

Berikut gambaran mengenai skema desain penelitian :



Gambar 3.1. Skema Desain Penelitian

D. Teknik Pengumpulan Data

1. Data Primer

Data yang didapat secara langsung dari sumber penelitian dengan tujuan untuk mendapatkan data yang diperlukan dalam penelitian.

2. Data Sekunder

Merupakan data yang telah dikumpulkan lembaga pengumpulan data dan dipublikasikan kepada masyarakat pengguna data.

E. Teknik Analisis Data

a. Teknik observasi yaitu dengan melakukan pengamatan dan pencatatan secara langsung kepada penderita gula darah.

b. Teknik tanya jawab yaitu dengan cara melakukan pertanyaan mengenal gejala dan solusi meringankan mengenai penyakit gula darah.

c. Teknik kepustakaan yaitu dengan mengumpulkan data dari buku atau bahan tulisan yang ada relevansinya dengan skripsi ini.

F. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah dengan menggunakan metode *Waterfall*. *Waterfall* merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan. *Waterfall* dibangun untuk mendefinisikan kebutuhan. *Waterfall* merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan dalam membuat sebuah sistem.[19] Tahapan pengembangan ada 6 sistem yaitu *Planning*, Analisis sistem, Desain sistem, Implementasi, Analisa Kebutuhan, dan Pengujian sistem. Berikut tahapan-tahapan dalam metode *waterfall* :

1. *Planning*

Tahap perencanaan merupakan tahap yang menjelaskan tentang persiapan menyangkut beberapa hal yang diperlukan sebelum masuk ke tahap analisis, yaitu *Business Need*, *Business Value*, dan menentukan cakupan sistem pakar diagnosa penyakit kelebihan

gula darah berbasis web dengan metode *Case Based Reasoning*.

Dalam tahap ini ada beberapa perencanaan yang perlu dibuat :

- a. *Business Need* : Membantu mempermudah dokter.
- b. *Business Value* : Mendeteksi penyakit dari gejala yang dialami.
- c. Cakupan, yaitu menentukan batasan ruang lingkup sistem yang akan dibangun, dalam kasus ini sistem yang akan dibangun mengenai sistem pakar untuk mendiagnosa penyakit kelebihan gula darah.

2. Analisis

Dalam perencanaan sebuah sistem perlu dilakukan sebuah analisis sistem guna mendapatkan data maupun informasi yang dapat dijadikan sebuah acuan dari sistem yang sedang berjalan. Karena dari data yang diperoleh dapat diketahui apa saja yang sudah dapat dilakukan oleh sistem yang berjalan dan apa saja yang belum dapat dilakukan sistem. Berdasarkan analisa sistem yang berjalan diharapkan dapat diketahui sejauh mana sistem yang berjalan memenuhi kebutuhan dan bagaimana solusi yang diberikan. Analisa dari sistem pakar untuk mendiagnosa penyakit kelebihan gula darah dapat di lihat alur sistem berikut:

- a. Admin melakukan *login*.
- b. Admin dapat memasukan data gejala dan data penyakit baru dalam sistem.
- c. Pasien dapat menginput gejala yang dialami.
- d. Sistem akan menganalisis gejala.
- e. Sistem menampilkan kemungkinan penyakit yang akan terjadi.

Analisa Kebutuhan

a. Identifikasi Masalah

Permasalahan yang terjadi pada sistem pemeriksaan pasien pada layanan kesehatan diantaranya :

- 1) Terlalu banyaknya pasien yang berobat sedangkan jumlah dokter masih sedikit.
- 2) Butuhnya tenaga bantuan agar mempersingkat waktu pemeriksaan.

b. Analisa Sistem

Pengambilan data yang *real* dan juga data dari studi literatur.

c. Kebutuhan Data

Data yang dibutuhkan dalam desain sistem pakar ini diantaranya : data keluhan pasien, data penyakit yang disebabkan gula darah, data solusi pencegahan.

d. Kebutuhan Fungsional

- 1) Proses *Login* untuk admin
- 2) Proses *Login* untuk pasien
- 3) Pengelolaan data pasien oleh admin, meliputi : *Input*, *Update*, dan *Delete*
- 4) Pengelolaan data penyakit oleh pasien, meliputi : *Input* dan *Delete*
- 5) Pengelolaan data solusi oleh pasien.

3. Desain Sistem

Setelah melakukan analisis pada sistem dilanjutkan dengan tahapan desain. Dalam tahapan ini adalah tahap perencanaan dan pemodelan arsitektur sistem. Dalam tahap pemodelan sistem penulis menggunakan perancangan UML (*Unified Modeling Language*) yang terdiri dari *Use Case* diagram, *Activity* diagram, *Sequence* diagram, *Class* diagram, Struktur database, *User interface*.

4. Implementasi

Tahapan ini merupakan menerapkan pada sistem yang sebenarnya, yaitu bagaimana mengimplementasikan sistem berdasarkan metode yang sudah dibuat. Berikut ini tahap-tahap pembuatan sistem:

- a. Sistem implementasi: desain interface, penulisan program, *username* dan *password* admin.
- b. Alokasi waktu: Klinik Pratama Umum Undaan Kudus.
- c. Basis sistem: *Web*

5. Pengujian Sistem

Pengujian sistem adalah teknik untuk mengkontruksi struktur program dengan melakukan pengujian untuk mengungkapkan kesalahan sehubungan dengan menggabungkan modul-modul secara bersama-sama. Pengujian dilakukan dengan beberapa tahap yaitu dengan pengujian *White Box*, *Black Box*, dan *User Acceptance Test (UAT)* :

- a. Pengujian *White Box* adalah salah satu cara untuk menguji suatu aplikasi atau *software* dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang dibuat ada yang salah atau tidak. Kalau modul yang telah dan sudah dihasilkan berupa *output* yang tidak sesuai dengan yang diharapkan maka akan dikompilasi ulang dan di cek kembali kode-kode tersebut hingga sesuai dengan yang diharapkan. Kasus yang sering menggunakan *white box testing* akan di uji dengan beberapa tahapan yaitu:
 1. Pengujian seluruh keputusan yang menggunakan logika.
 2. Pengujian keseluruhan *loop* yang ada sesuai batasan-batasannya.
 3. Pengujian pada struktur data yang sifatnya internal dan yang terjamin validitasnya.
- b. Pengujian *Black Box* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program. *Black box* merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *white box*.

Black Box Testing cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

Pengujian didesain untuk menjawab pertanyaan-pertanyaan berikut:

1. Bagaimana fungsi-fungsi diuji agar dapat dinyatakan valid?
 2. *Input* seperti apa yang dapat menjadi bahan kasus uji yang baik?
 3. Apakah sistem sensitif pada input-input tertentu?
 4. Bagaimana sekumpulan data dapat diisolasi?
- c. Pengujian *User Acceptance Tester* adalah pengujian untuk mengetahui terhadap tampilan pada aplikasi yang kita buat, UAT biasanya dinilai oleh user aplikasi tersebut, pada UAT tidak jauh beda dengan kusioner pada pembuatan aplikasi. *User Acceptance Test* (UAT) terdiri dari beberapa langkah pengujian, yang memverifikasi apakah kebutuhan penggunaan sistem atau aplikasi bisa dimanfaatkan dengan baik dan benar oleh pengguna. Jika pengguna telah puas dan bisa menjalankan sistem atau perangkat lunak, maka bisa dikatakan sistem itu sudah bagus. Uji fungsional di sisi lain, tes persyaratan khusus dan spesifikasi dari perangkat lunak. Ini tidak memiliki komponen pengguna. Sebuah tes fungsional bisa menyimpulkan bahwa perangkat lunak memenuhi spesifikasinya. Namun, itu tidak memverifikasi apakah itu benar-benar bekerja untuk pengguna. Dimensi fungsional hanya salah satu dari banyak.[20]

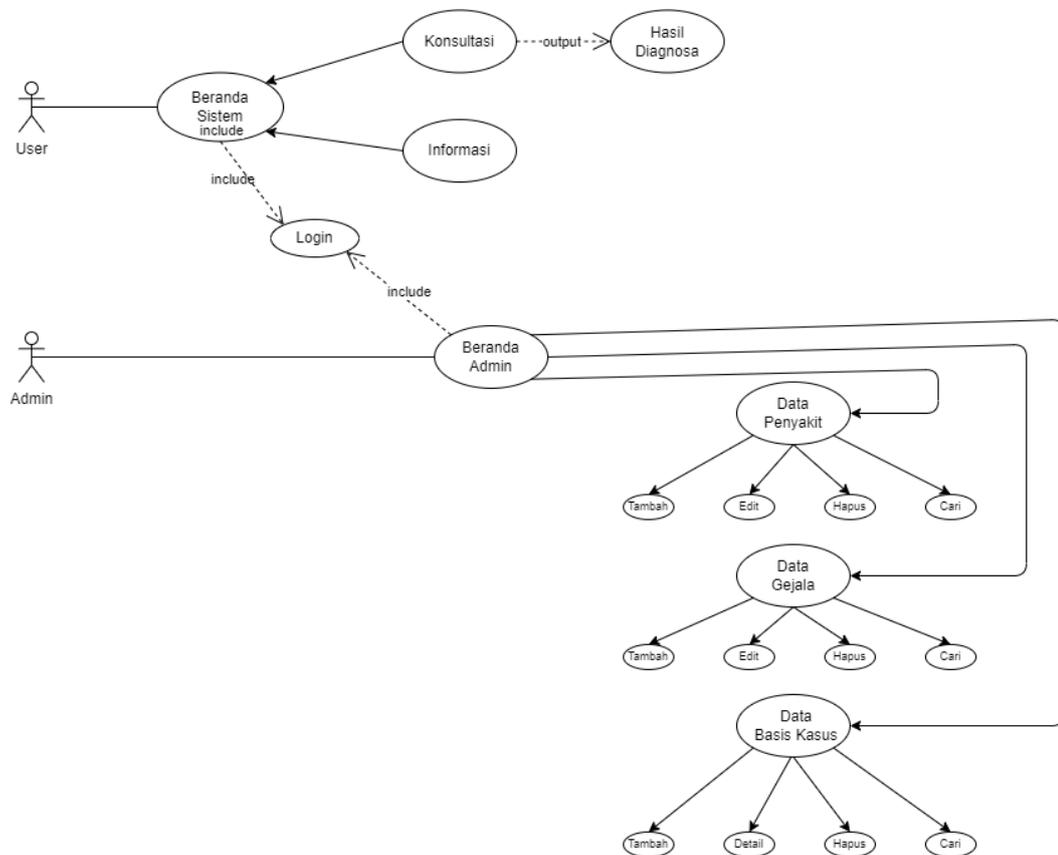
BAB IV

HASIL DAN PEMBAHASAN

A. Hasil

1. Use Case Diagram

Use case diagram adalah salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, *use case diagram* juga dapat mendeskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.



Gambar 4.1 Use Case Diagram

a. Skenario *Use Case (Flow Of Events)*

Use case menjelaskan apa yang sistem akan lakukan, untuk membangun sistem secara aktual memerlukan rancangan yang lebih spesifik yang menjelaskan secara rinci apa yang pemakai akan lakukan dan apa yang sistem itu sendiri lakukan. Dari diagram *use case* sistem di atas, maka dapat dibuatkan beberapa skenario antara lain :

1) Deskripsi *Use Case Diagram Login Admin*

Nama use case : Menu login admin
 Aktor : Adminitator
 Tujuan : Agar admin dapat masuk ke menu dashboard

Tabel 4.1 deskripsi *use case* diagram *login admin*

Aktor	Sistem
1. <i>Input Username dan Password</i>	
2. <i>Klik tombol Login</i>	3. <i>Login berhasil</i>
Aktifitas Alternatif	
Aktor	Sistem
1. <i>Salah memasukkan username dan password</i>	2. <i>Mereset dan kembali ke form isi username dan password</i>

2) Deskripsi *Use Case Diagram Konsultasi*

Nama : Menu konsultasi
 Aktor : *User*
 Tujuan : Agar *user* dapat mengecek jenis penyakit

Tabel 4.2 deskripsi *use case* diagram konsultasi

Aktor	Sistem
1. <i>Klik menu konsultasi</i>	2. <i>Tampil form diagnosa</i>
3. <i>Memilih gejala penyakit</i>	
4. <i>Klik button diagnosa penyakit</i>	5. <i>Proses mendiagnosa</i>

	6. Sukses mendiagnosa penyakit
	7. Tampil hasil analisa penyakit

3) Deskripsi Kelola Data Penyakit

Nama use case : Menu data penyakit

Aktor : Adminitator

Tujuan : Agar admin dapat memasukkan data penyakit

Tabel 4.3 deskripsi kelola data penyakit

Aktor	Sistem
1. Klik menu Penyakit	2. Menampilkan menu tabel penyakit
3. Kli button tambah data penyakit	4. Tampil halaman tambah data penyakit
5. <i>Input</i> data penyakit	
6. Klik tombol simpan	7. Sukses menyimpan data penyakit
8. Klik button edit data penyakit	9. Menampilkan halaman <i>form</i> edit data penyakit
10. Mengedit data penyakit	
11. Klik tombol simpan	12. Sukses mengedit data penyakit
13. Klik detail	14. Menampilkan detail data penyakit
15. Klik hapus	16. Menghapus data penyakit
	17. Sukses menghapus data penyakit
18. Mengisi <i>keyword</i> dan klik tombol cari	19. Menampilkan hasil pencarian
Aktifitas Alternatif	
Aktor	Sistem
1. Jika batal menambahkan data penyakit klik batal	2. Kembali kemenu <i>form</i> kelola data penyakit
3. Jika batal atau ragu untuk menghapus data penyakit klik kembali	4. Kembali kemenu data penyakit

4) Deskripsi Kelola Data Gejala

Nama use case : Menu data gejala

Aktor : Adminitator

Tujuan : Agar admin dapat memasukkan data gejala

Tabel 4.4 deskripsi kelola data gejala

Aktor	Sistem
1. Klik menu gejala	2. Menampilkan menu tabel gejala
3. Klik tombol tambah data gejala	4. Tampil halaman tambah data gejala
5. Input data gejala	
6. Klik tombol simpan	7. Sukses menyimpan data gejala
8. Klik tombol edit data penyakit	9. Menampilkan halaman <i>form</i> edit data gejala
10. Mengedit data gejala	
11. Klik tombol simpan	12. Sukses mengedit data gejala
13. Klik detail	14. Menampilkan detail data gejala
15. Klik hapus	16. Menghapus data gejala
	17. Sukses menghapus data gejala
18. Mengisi <i>keyword</i> dan klik button cari	19. Menampilkan hasil pencarian
Aktifitas Alternatif	
Aktor	Sistem
1. Jika batal menambahkan data gejala klik batal	2. Kembali kemenu <i>form</i> kelola data gejala
3. Jika batal atau ragu untuk menghapus data gejala klik kembali	4. Kembali kemenu data gejala

5) Deskripsi Kelola Basis Kasus

Nama use case : Menu basis kasus

Aktor : Admin

Tujuan : Agar admin dapat menyesuaikan data jenis penyakit beserta gejalanya

Tabel 4.5 deskripsi kelola basis kasus

Aktor	Sistem
1. Klik menu basis kasus	2. Menampilkan menu tabel basis kasus
3. Klik tombol tambah data basis kasus	4. Tampil halaman tambah data basis kasus
5. Pilih penyakit	
6. Klik tombol tambah gejala	
7. Pilih gejala	
8. Klik tombol simpan	9. Sukses menambah data basis kasus
10. Klik detail	11. Menampilkan detail data basis kasus
12. Klik hapus	13. Menghapus data basis kasus
	14. Sukses menghapus data basis kasus
15. Mengisi keyword dan klik tombol cari	16. Menampilkan hasil pencarian
Aktifitas Alternatif	
Aktor	Sistem
1. Jika batal menambahkan data basis kasus klik batal	2. Kembali kemenu <i>form</i> kelola data basis kasus
3. Jika batal atau ragu untuk menghapus data basis kasus klik kembali	4. Kembali kemenu data basis kasus

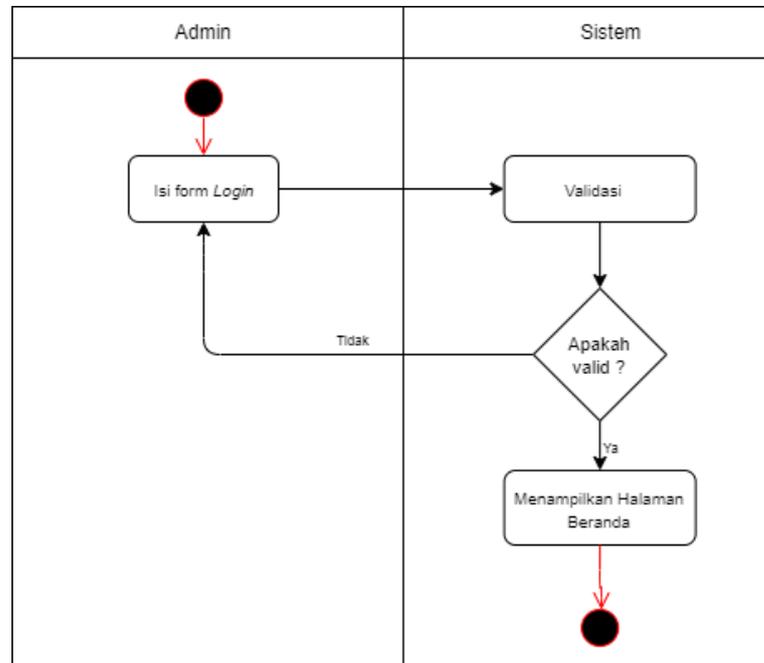
2. Activity Diagram

Activity diagram adalah sesuatu yang menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Berikut adalah *activity* diagram sistem pakar diagnosa penyakit kelebihan gula berbasis web di Klinik Pratama Umum Undaan Kudus.

a. Activity Diagram Login Admin

Activity diagram login admin merupakan scenario perintah login pada halaman sistem pakar diagnosa penyakit kelebihan gula darah.

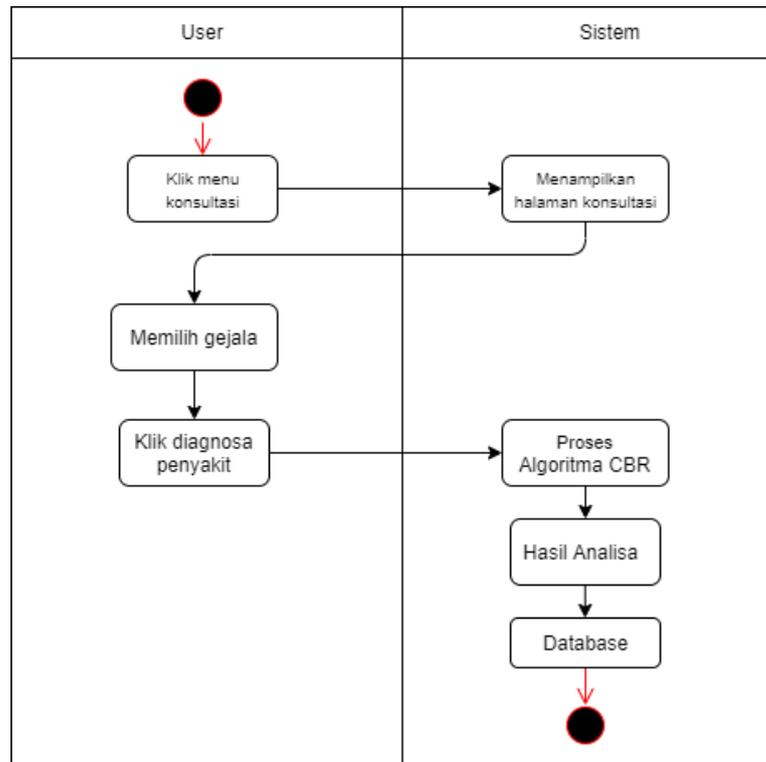
Admin masuk ke menu *login* dengan menggunakan *username* dan *password*. *Username* dan *password* diproses untuk di cek validasinya.



Gambar 4.2 Activity Diagram Login

b. Activity Diagram Konsultasi

Activity diagram konsultasi merupakan halaman untuk user konsultasi. User memasukkan gejala yang dialami, setelah selesai pilih diagnosa maka sistem akan menganalisis penyakit



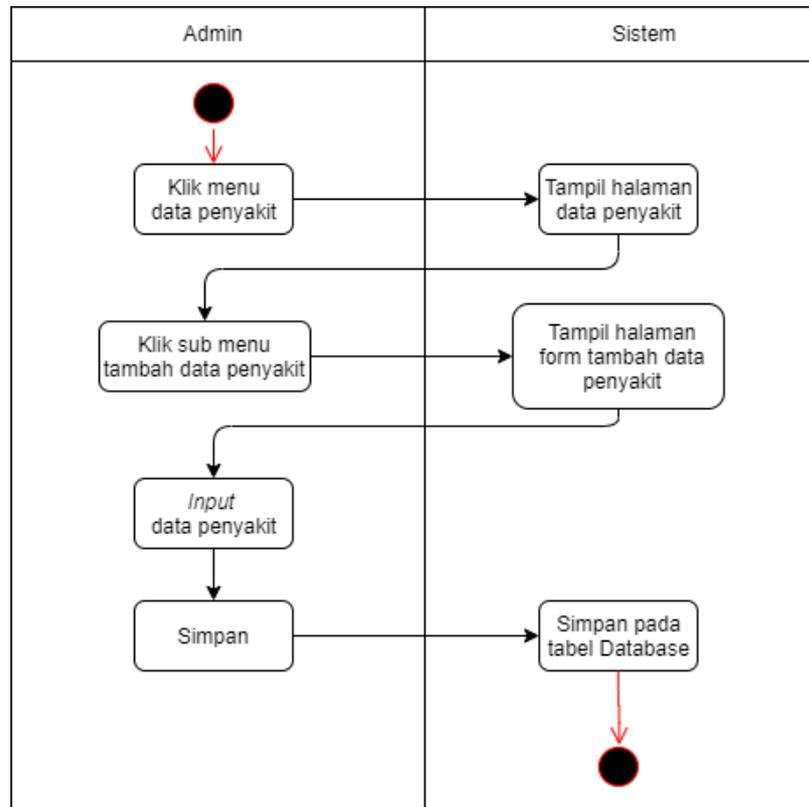
Gambar 4.3 *Activity Diagram* Konsultasi

c. *Activity Diagram* Kelola Data Penyakit

Activity diagram kelola data penyakit merupakan halaman untuk admin mengisi ataupun mengubah data gejala dan penyakit. Dalam kelola data penyakit terdapat menu tambah data, edit data, hapus data dan cari data penyakit.

1) *Activity Diagram* Tambah Data Penyakit

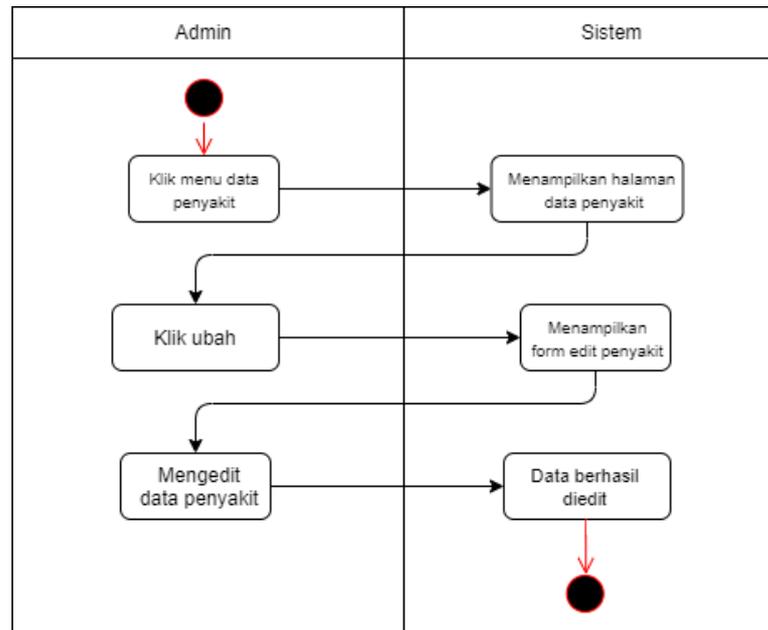
Activity diagram tambah data penyakit merupakan halaman untuk admin menambah data penyakit.



Gambar 4.4 *Activity Diagram* Tambah Data Penyakit

2) *Activity Diagram* Ubah Data Penyakit

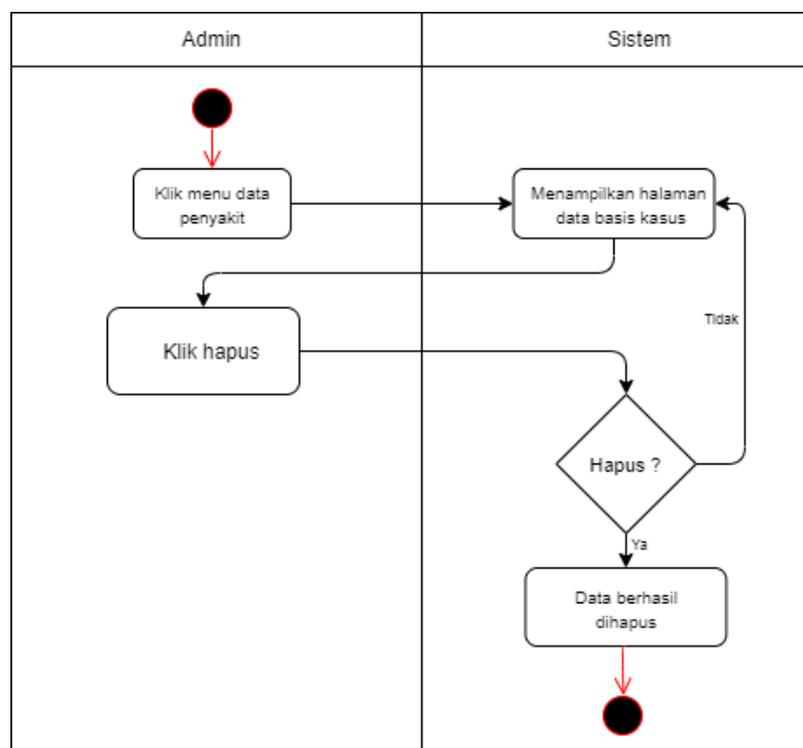
Activity diagram ubah data penyakit merupakan halaman untuk mengubah data penyakit.



Gambar 4.5 Activity Diagram Ubah Data Penyakit

3) Activity Diagram Hapus Data Penyakit

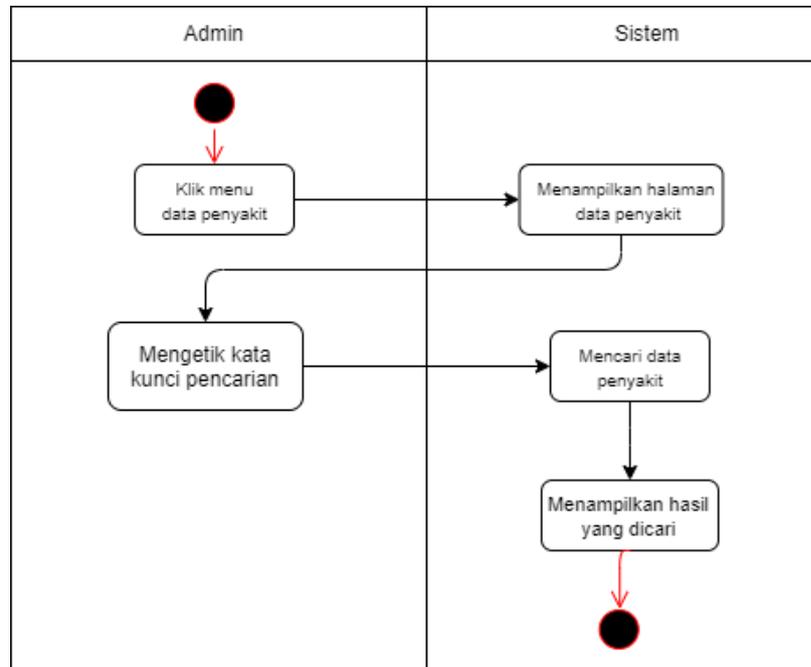
Activity diagram hapus data penyakit merupakan halaman untuk menghapus data penyakit.



Gambar 4.6 Activity Diagram Hapus Data Penyakit

4) *Activity Diagram* Cari Data Penyakit

Activity diagram cari data penyakit merupakan halaman untuk admin mencari data penyakit.



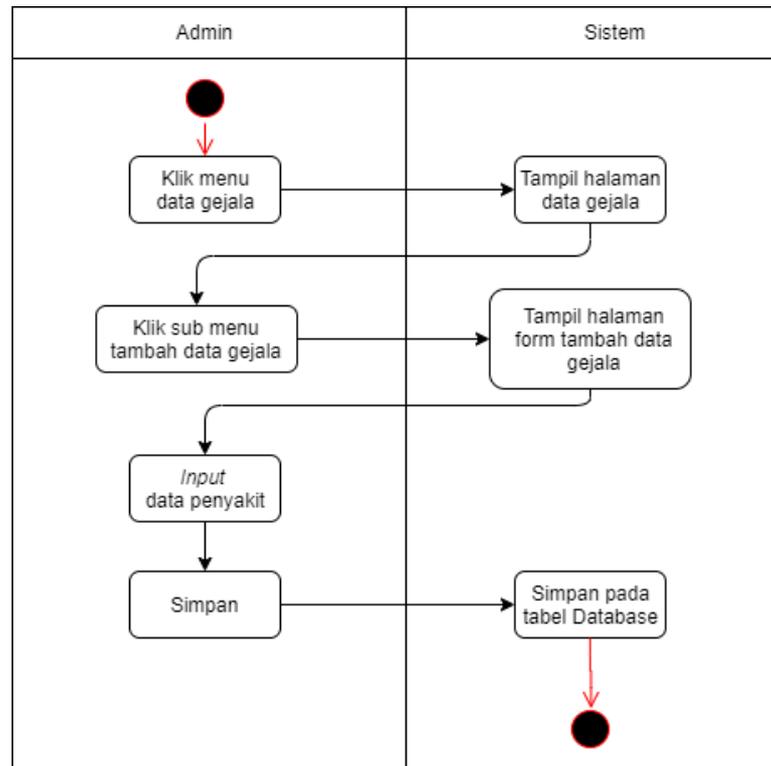
Gambar 4.7 *Activity Diagram* Cari Data Penyakit

d. *Activity Diagram* Kelola Data Gejala

Activity diagram kelola data penyakit merupakan halaman untuk admin mengisi ataupun mengubah data gejala dan penyakit. Dalam kelola data penyakit terdapat menu tambah data, edit data, hapus data dan cari data penyakit.

1) *Activity Diagram* Tambah Data Gejala

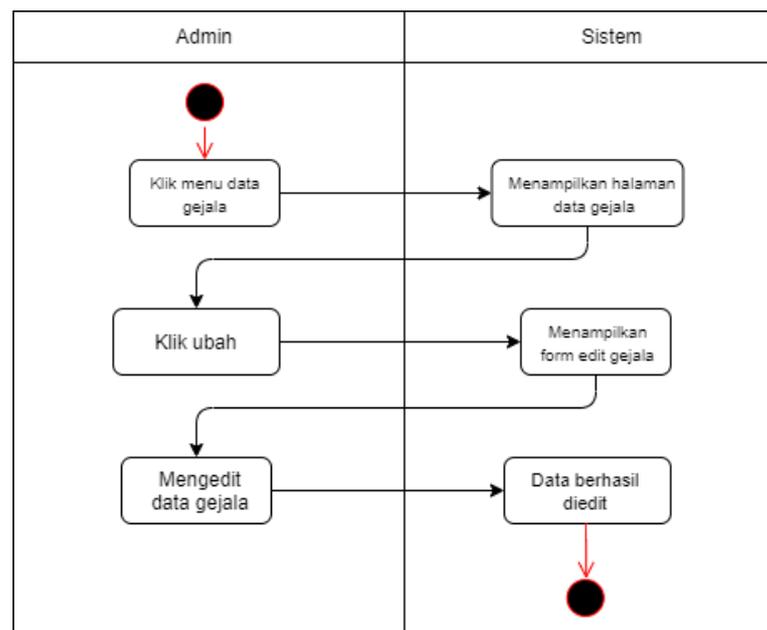
Activity diagram tambah data gejala merupakan halaman untuk admin menambah data gejala.



Gambar 4.8 Activity Diagram Tambah Data Gejala

2) Activity Diagram Edit Data Gejala

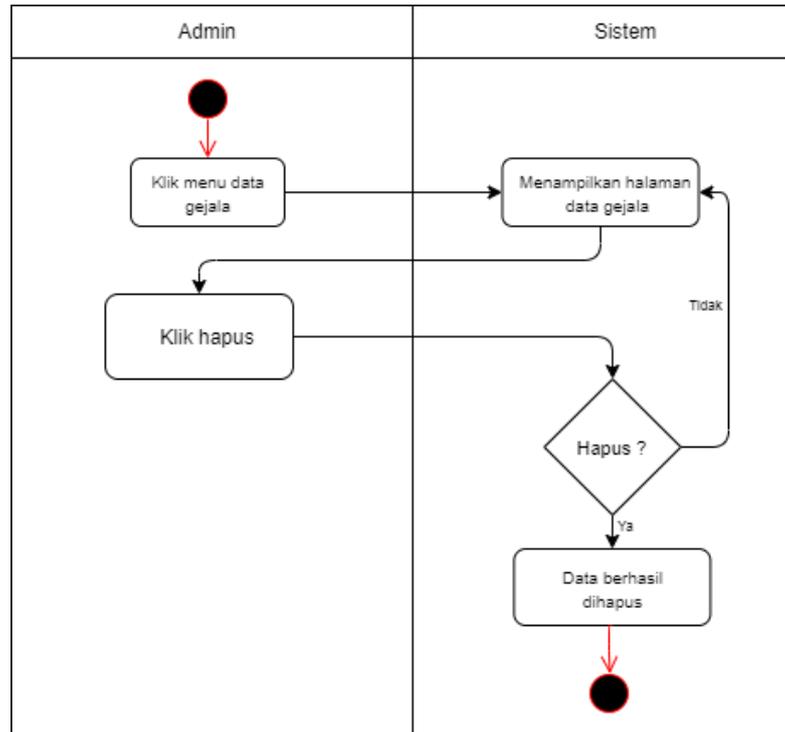
Activity diagram edit data gejala merupakan halaman untuk admin mengedit data gejala.



Gambar 4.9 Activity Diagram Edit Data Gejala

3) *Activity Diagram* Hapus Data Gejala

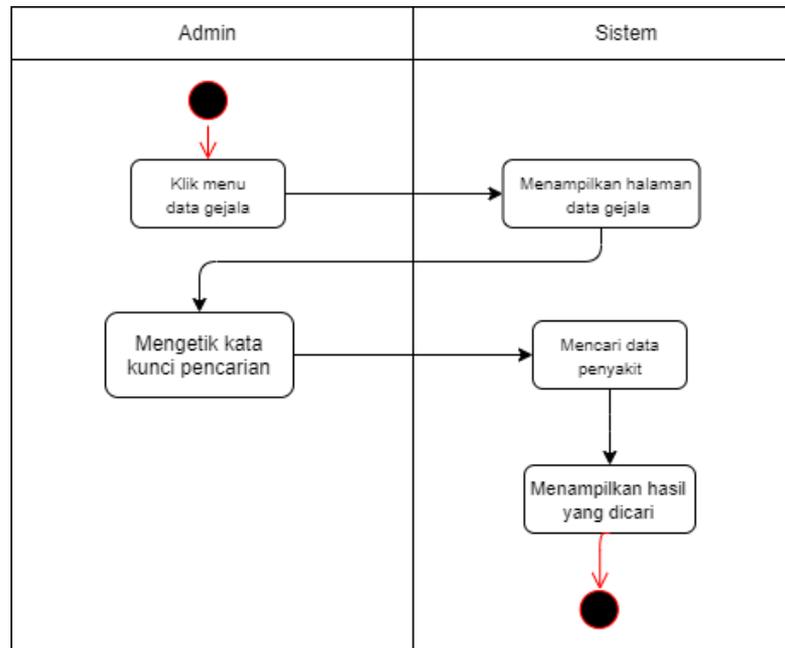
Activity diagram hapus data gejala merupakan halaman untuk admin menghapus data gejala.



Gambar 4.10 *Activity Diagram* Hapus Data Gejala

4) *Activity Diagram* Cari Data Gejala

Activity diagram cari data gejala merupakan halaman untuk admin mencari data gejala.



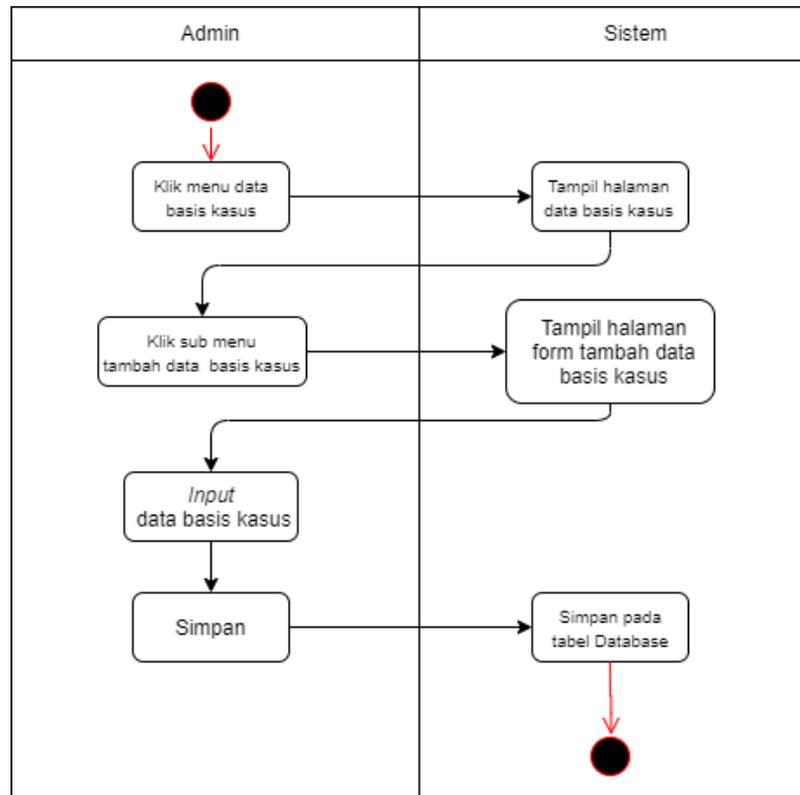
Gambar 4.11 *Activity Diagram* Cari Data Gejala

e. *Activity Diagram* Kelola Data Basis Kasus

Activity diagram kelola data basis kasus merupakan halaman untuk admin mendata jenis penyakit beserta gejalanya. Dalam kelola data basis kasus terdapat menu tambah data, detail data, hapus data dan cari data basis kasus.

1) *Activity Diagram* Tambah Data Basis Kasus

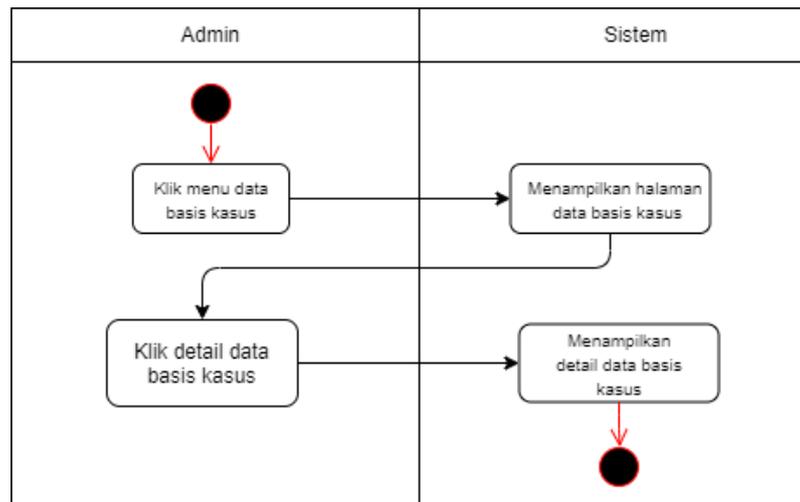
Activity diagram tambah data basis kasus merupakan halaman untuk admin menambah data basis kasus.



Gambar 4.12 *Activity Diagram* Tambah Data Basis Kasus

2) *Activity Diagram* Detail Data Basis Kasus

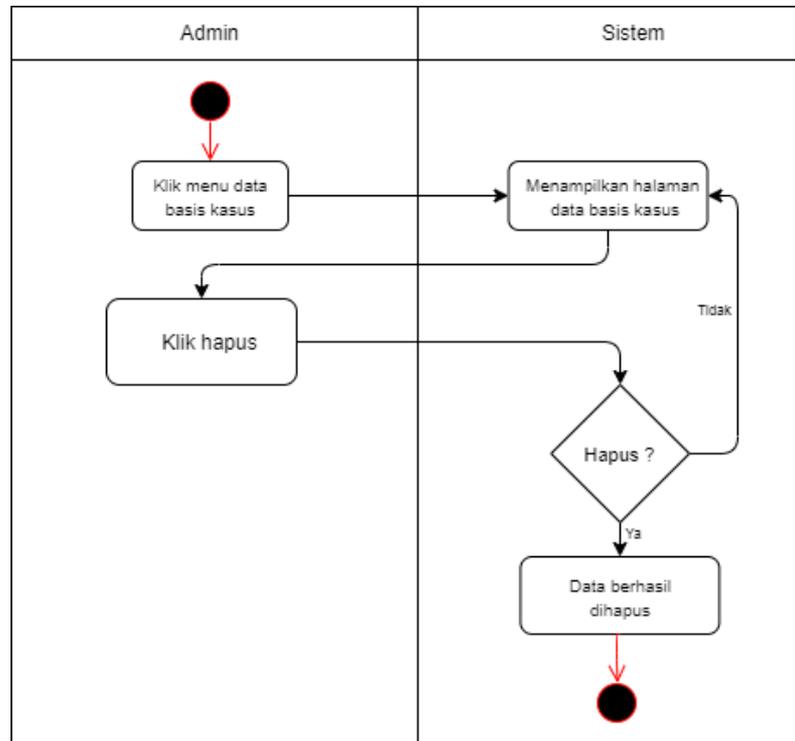
Activity diagram detail data basis kasus merupakan halaman untuk admin melihat secara detail data basis kasus.



Gambar 4.13 *Activity Diagram* Detail Data Basis Kasus

3) *Activity Diagram* Hapus Data Basis Kasus

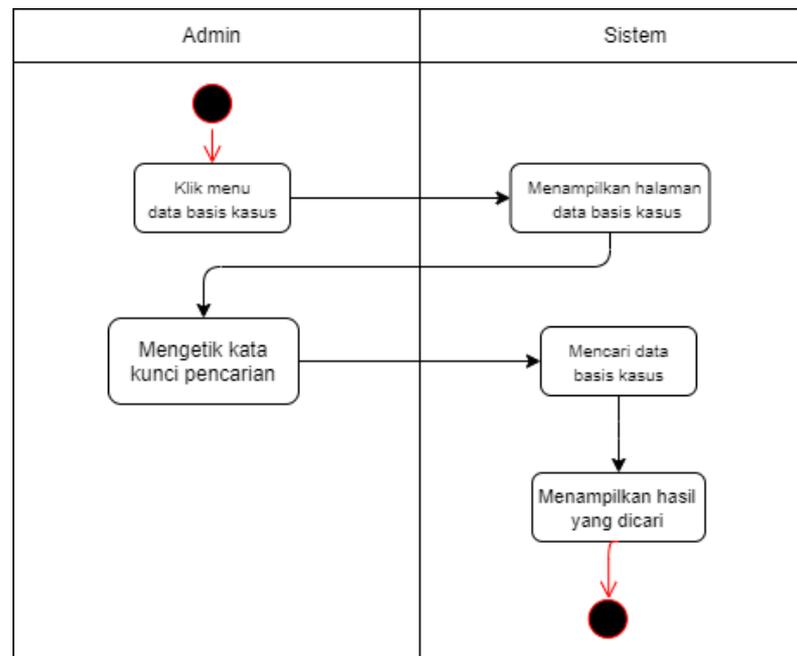
Activity diagram hapus data basis kasus merupakan halaman untuk admin menghapus data basis kasus.



Gambar 4.14. *Activity Diagram* Hapus Data Basis Kasus

4) *Activity Diagram* Cari Data Basis Kasus

Activity diagram cari data basis kasus merupakan halaman untuk admin mencari data basis kasus.



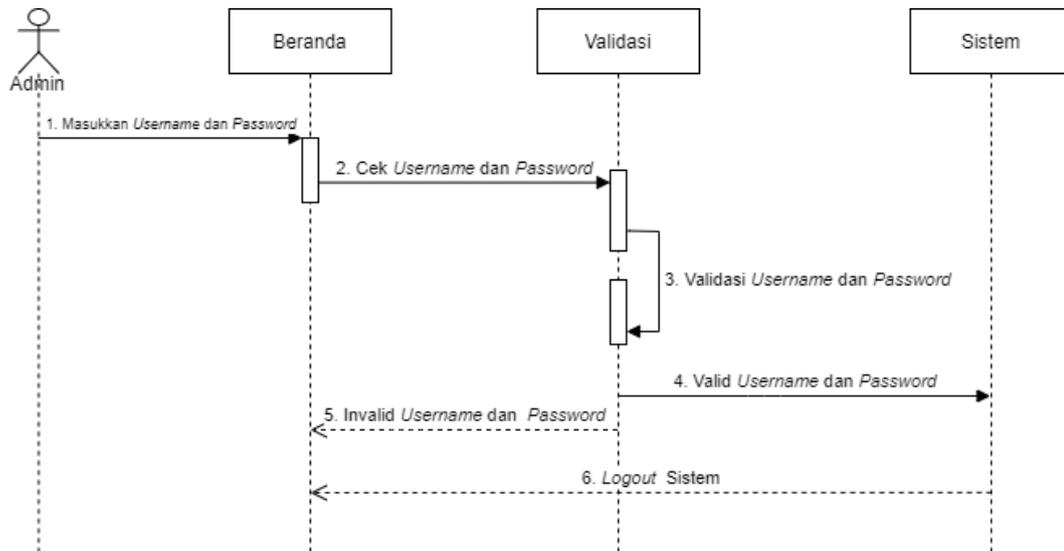
Gambar 4.15. Activity Diagram Cari Data Basis Kasus

3. Sequence Diagram

Sequence Diagram menggambarkan hubungan antara objek pada *use case*, dengan mendeskripsikan waktu objek dan pesan yang dikirimkan antar objek, untuk menggambarkan *Sequence Diagram* terlebih dahulu diketahui objek yang terlibat didalam usecase beserta metode yang dimiliki oleh kelas. Berikut adalah penjelasan *sequence diagram* sistem pakar mendiagnosa penyakit kelebihan gula darah :

a. Sequence Diagram Login Admin

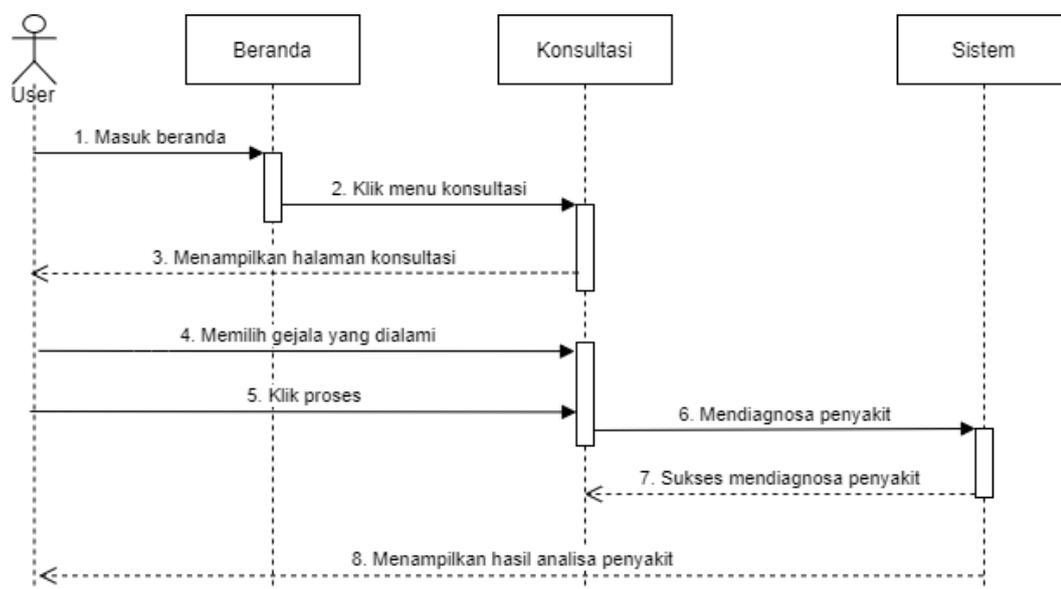
Sequence diagram login admin merupakan skenario perintah login pada halaman aplikasi. Admin masuk ke menu *login* dengan memasukkan *username* dan *password*. Dari menu beranda, sistem akan mengecek *username* dan *password*. *Username* dan *password* diproses untuk dicek validitasnya. Validitas ini bertujuan untuk membuka menu utama sistem, gambar 4.16. menyajikan *sequence diagram login admin*.



Gambar 4.16. *Sequence Diagram Login Admin*

b. *Sequence Diagram Konsultasi*

Sequence Diagram Konsultasi merupakan skenario untuk konsultasi. *User* masuk ke menu beranda. *User* masuk ke menu konsultasi. Sistem akan menampilkan menu konsultasi. *User* memilih gejala yang dialami. Sistem akan memproses gejala yang diinputkan dan mendiagnosa penyakit. Sistem menampilkan kemungkinan penyakit yang dialami oleh user.

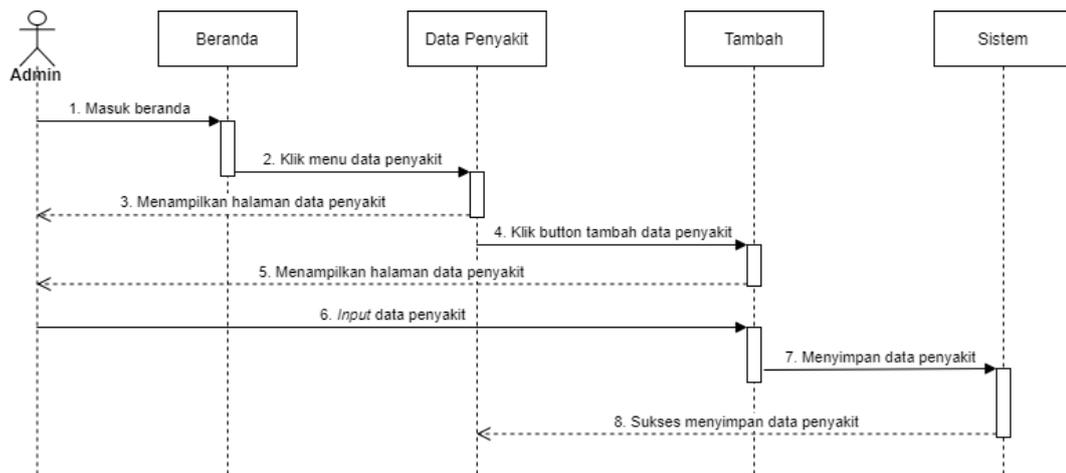


Gambar 4.17. *Sequence Diagram Konsultasi*

c. *Sequence Diagram* Kelola Data Penyakit

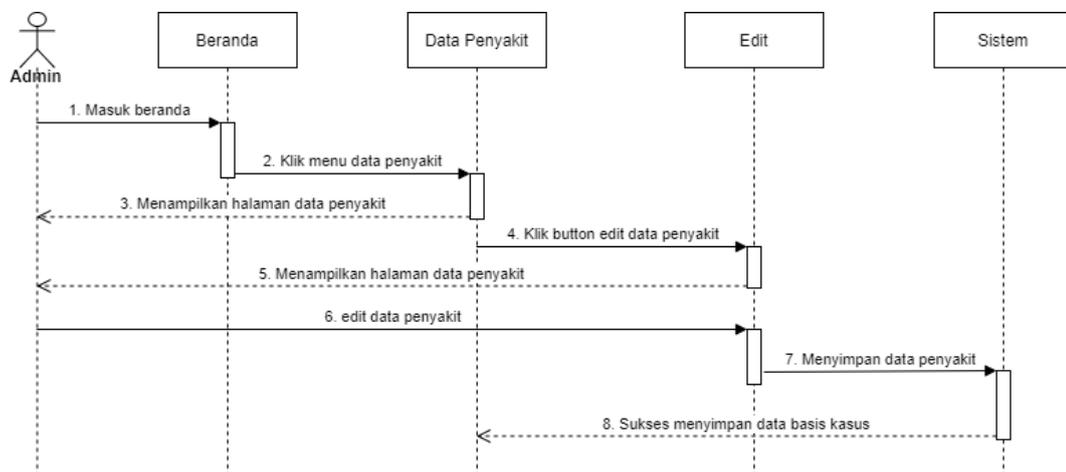
Sequence Diagram kelola data penyakit merupakan skenario data penyakit yang akan ditambahkan, diedit, dihapus, dan dicari.

1) *Sequence Diagram* Tambah Data Penyakit



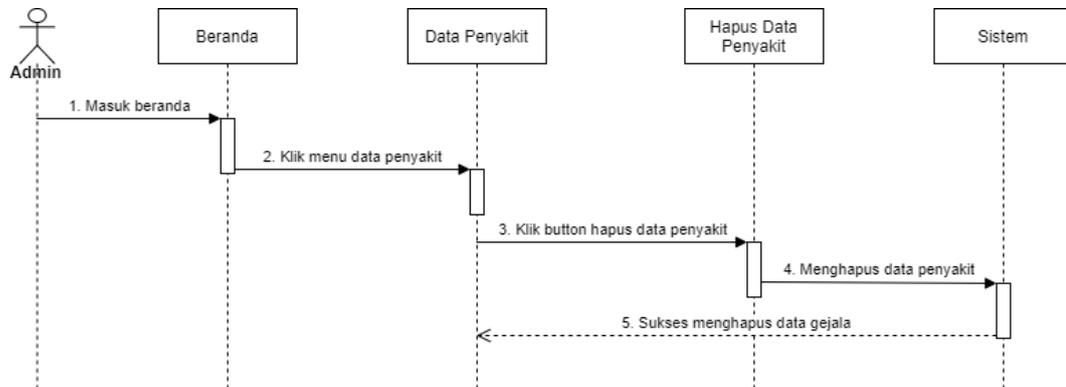
Gambar 4.18. *Sequence Diagram* Tambah Data Penyakit

2) *Sequence Diagram* Edit Data Penyakit



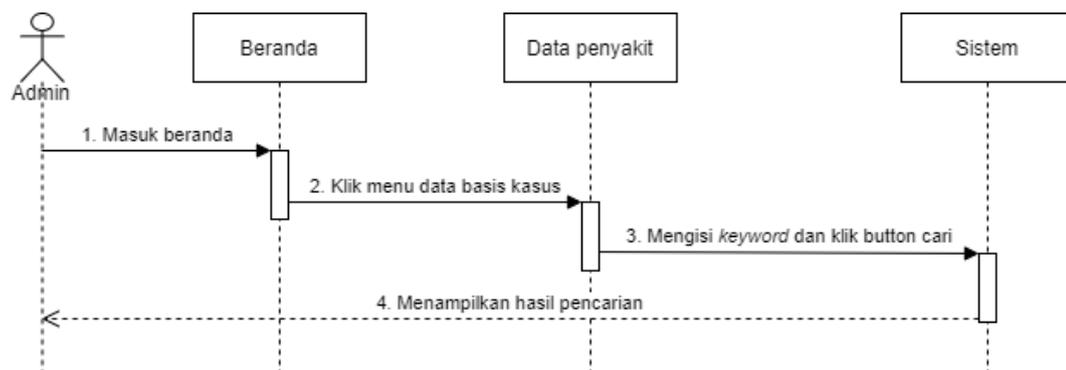
Gambar 4.19. *Sequence Diagram* Edit Data Penyakit

3) *Sequence Diagram* Hapus Data Penyakit



Gambar 4.20. *Sequence Diagram* Hapus Data Penyakit

4) *Sequence Diagram* Cari Data Penyakit

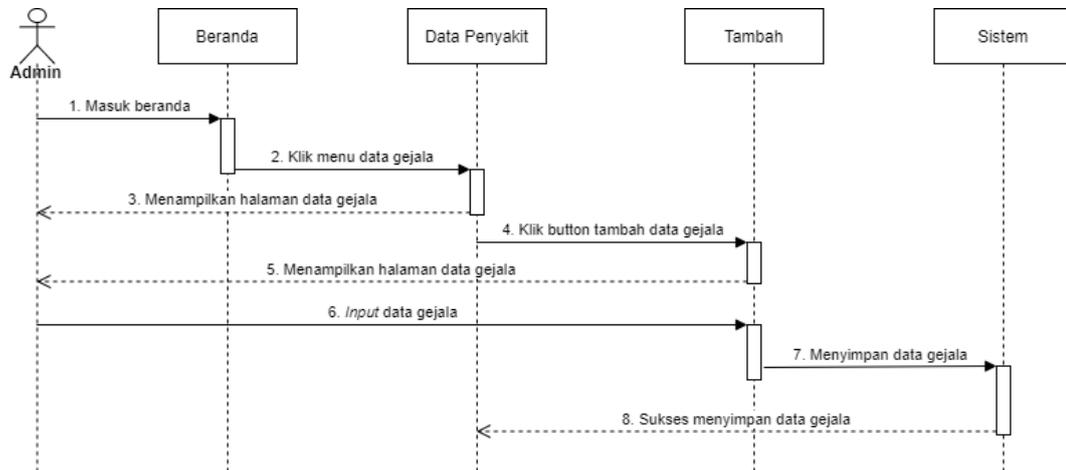


Gambar 4.21. *Sequence Diagram* Cari Data Penyakit

d. *Sequence Diagram* Kelola Data Gejala

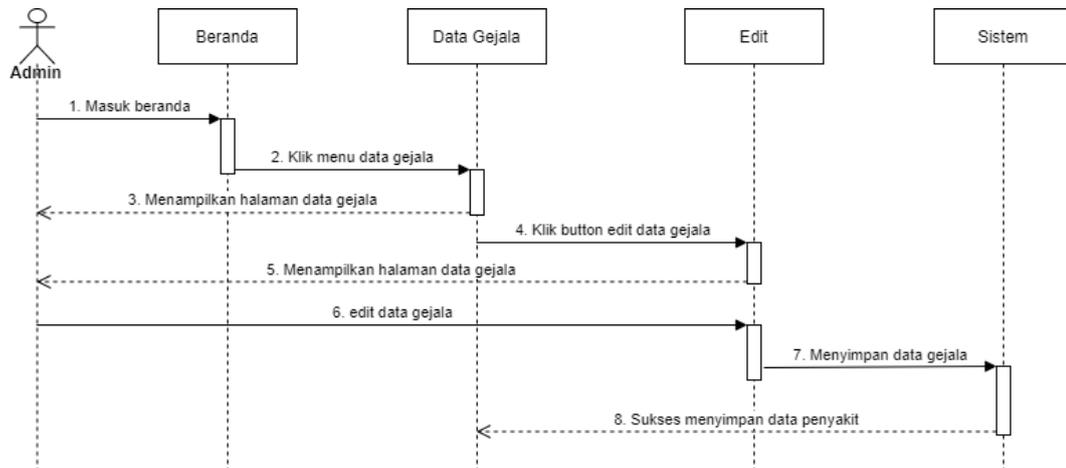
Sequence Diagram kelola data gejala merupakan skenario data gejala yang akan ditambahkan, diedit, dihapus, dan dicari.

1) *Sequence Diagram* Tambah Data Gejala



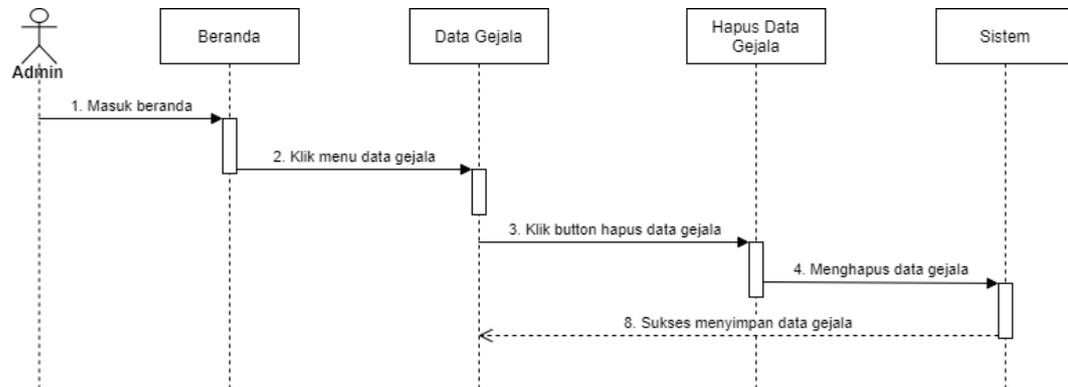
Gambar 4.22. *Sequence Diagram* Tambah Data Gejala

2) *Sequence Diagram* Edit Data Gejala



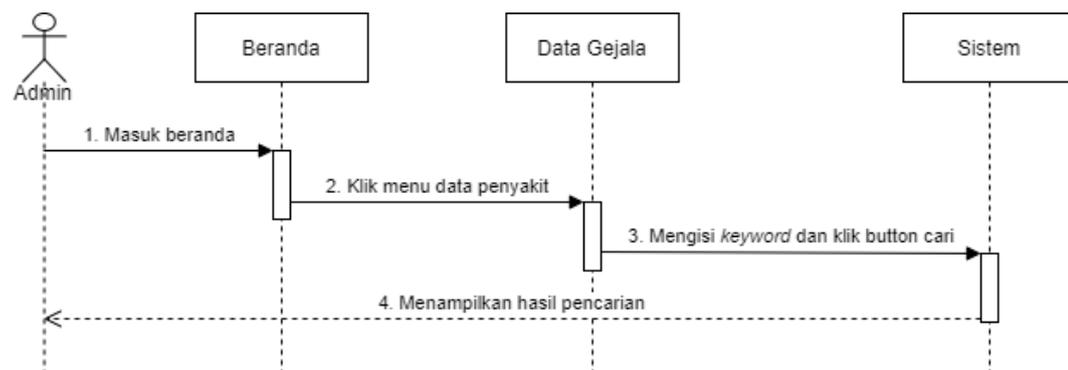
Gambar 4.23. *Sequence Diagram* Edit Data Gejala

3) *Sequence Diagram* Hapus Data Gejala



Gambar 4.24. *Sequence Diagram* Hapus Data Gejala

4) *Sequence Diagram* Cari Data Gejala

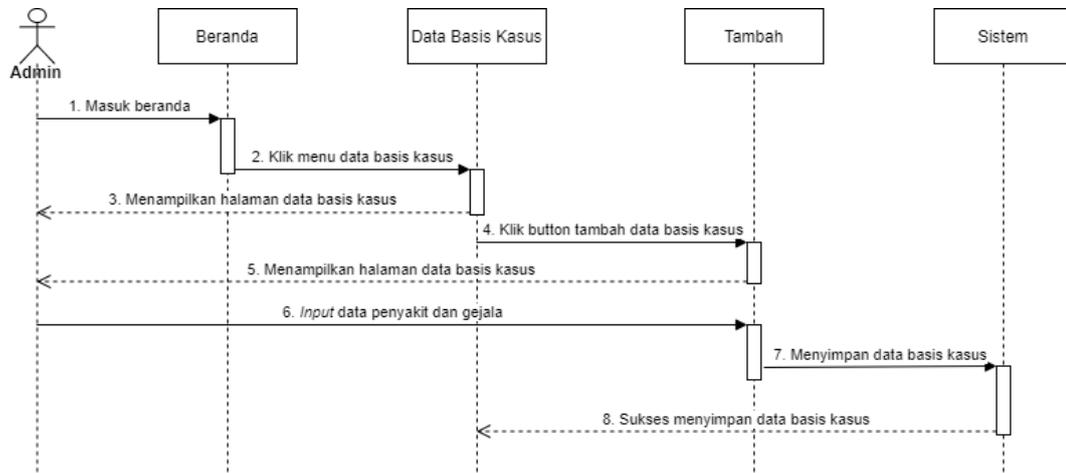


Gambar 4.25. *Sequence Diagram* Cari Data Gejala

e. *Sequence Diagram* Kelola Data Basis Kasus

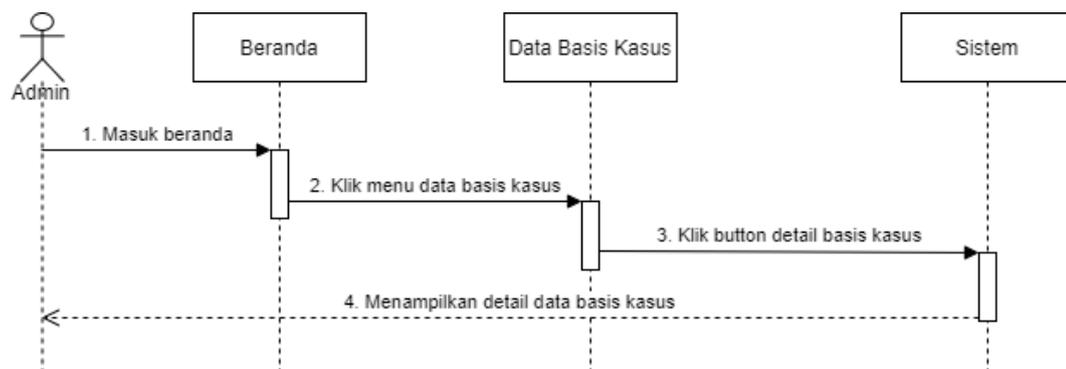
Sequence Diagram kelola data basis kasus merupakan skenario data basis kasus yang akan ditambahkan, detail, dihapus, dan dicari.

1) *Sequence Diagram* Tambah Data Basis Kasus



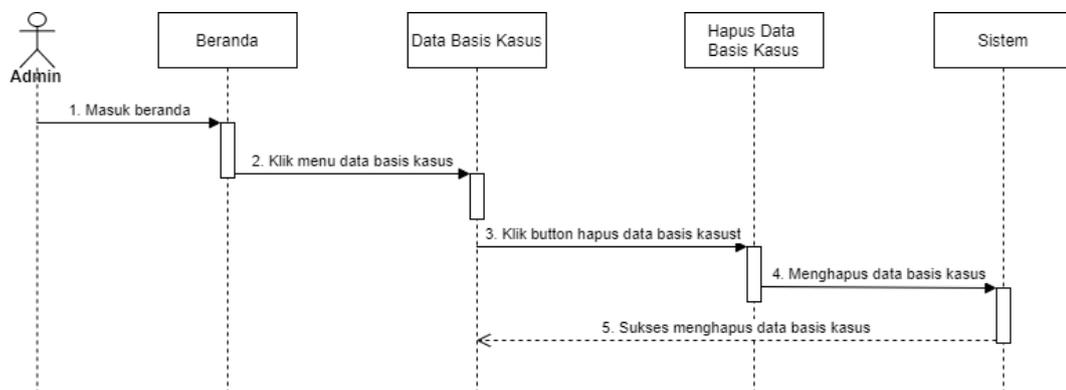
Gambar 4.26. *Sequence Diagram* Tambah Data Basis Kasus

2) *Sequence Diagram* Detail Data Basis Kasus



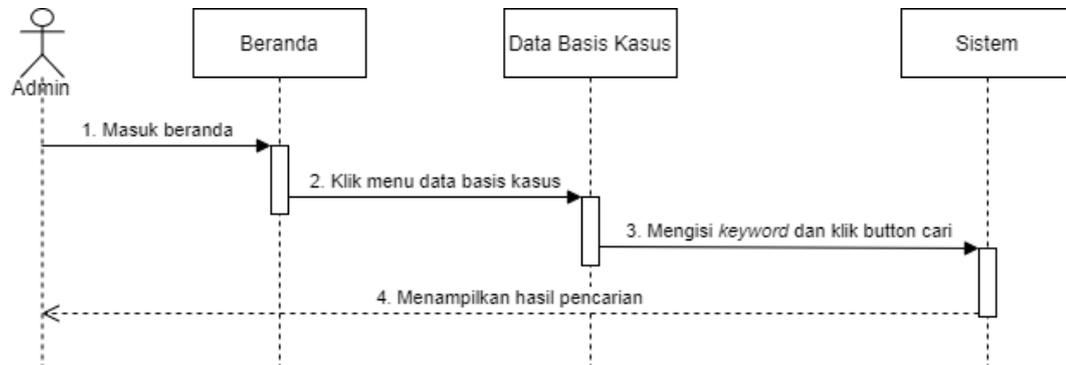
Gambar 4.27. *Sequence Diagram* Detail Data Basis Kasus

3) *Sequence Diagram* Hapus Data Basis Kasus



Gambar 4.28. *Sequence Diagram* Hapus Data Basis Kasus

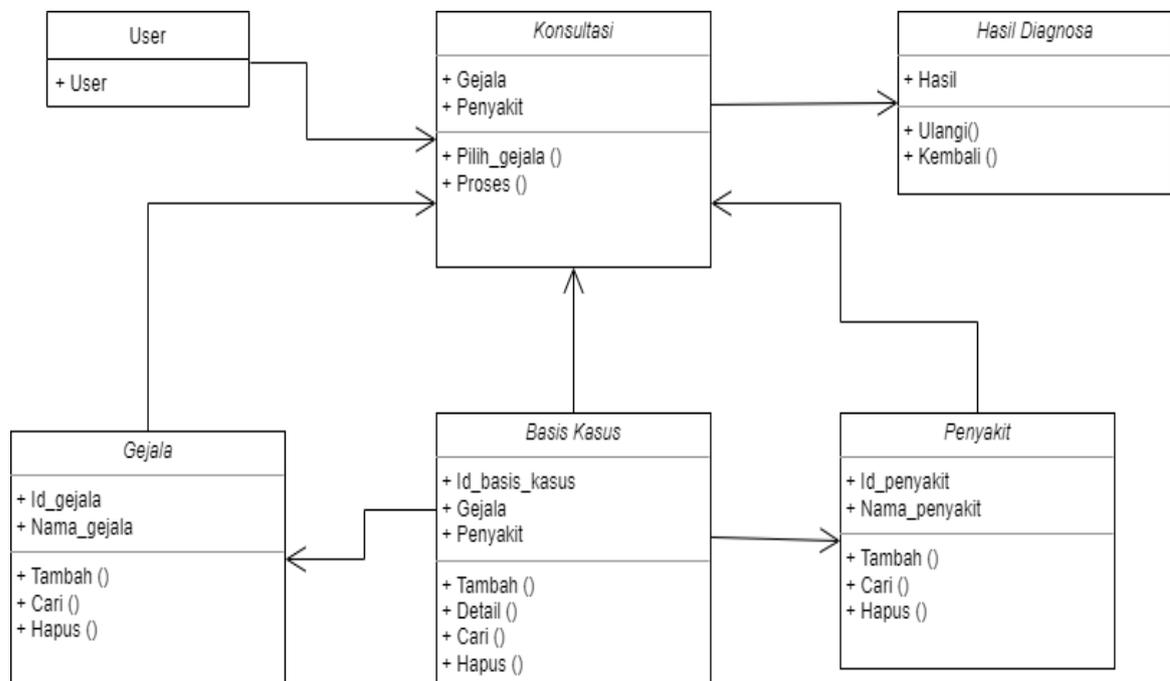
4) *Sequence Diagram* Cari Data Basis Kasus



Gambar 4.29. *Sequence Diagram* Cari Data Basis Kasus

4. *Class Diagram*

Class Diagram adalah salah satu jenis diagram yang digunakan untuk memetakan struktur sistem tertentu dengan memodelkan kelas, atribut, operasi, serta hubungan antar objek. Dalam *class diagram* terdapat 3 bagian yaitu nama kelas, atribut, dan metode/operasi. *Class Diagram* dari sistem ini dapat dilihat pada Gambar 4.30. menyajikan desain perancangan database.



Gambar 4.30. *Class Diagram*

5. Struktur Database

Perancangan data berisi *field-field* yang diperlukan dalam database yang akan digunakan didalam sistem. Berikut adalah rancangan data pada sistem pakar diagnosa penyakit kelebihan gula darah berbasis *web* :

Tabel 4.6. Struktur Database Admin

Field	Type	Lenght	Keterangan
<i>id_admin</i>	int	11	Digunakan untuk menyimpan nomer admin dengan format = 9999
<i>username</i>	var	10	Digunakan untuk menyimpan <i>username</i> dengan format = xxxx
<i>password</i>	var	10	Digunakan untuk menyimpan <i>password</i> dengan format = xxxx

Tabel 4.7. Struktur Database Gejala

Field	Type	Length	Keterangan
<i>id_ciri</i>	int	11	Digunakan untuk menyimpan id gejala dengan format = 9999
<i>nama_ciri</i>	text		Digunakan untuk menyimpan nama gejala dengan format = xxxx
<i>bobot</i>	int	11	Digunakan untuk menyimpan bobot gejala dengan format = 9999

Tabel 4.8. Struktur Database Penyakit

Field	Type	Length	Keterangan
<i>id_jenis</i>	var	5	Digunakan untuk menyimpan id penyakit dengan format = 9999
<i>nama_jenis</i>	var	30	Digunakan untuk menyimpan nama penyakit dengan format = xxxx

detail_jenis	text		Digunakan untuk menyimpan detail penyakit dengan format = xxxx
solusi_jenis	text		Digunakan untuk menyimpan solusi penyakit dengan format = xxxx

Tabel 4.9. Struktur Database Basis Kasus

Field	Type	Length	Keterangan
id_klasifikasi	int	11	Digunakan untuk menyimpan id klasifikasi penyakit dan gejala dengan format = 9999
id_jenis	var	5	Digunakan untuk menyimpan id penyakit dengan format = 9999
id_ciri	int	11	Digunakan untuk menyimpan id gejala dengan format = 9999

6. Kamus Data

Berikut adalah kamus data atau referensi data yang ada pada basis data sistem yang dibangun :

a. Tabel Admin

Admin = id_admin + username + password

id_admin = 4 { Integer } 4

username = 1 { Varchar } 10

password = 1 { Varchar } 10

Number = [0...9]

Char = [A...Z] | [a...z] | [0...9]

b. Tabel Gejala

Gejala = id_ciri + name_ciri + bobot

id_ciri = 2 { Integer } 11

nama_ciri = 1 { Text } 30

bobot = 1 { Integer } 2

Number = [0...9]

Char = [A...Z] | [a...z] | [0...9]

c. Tabel Penyakit

Penyakit = id_jenis + nama_jenis + detail_jenis + solusi_jenis

id_jenis = 1 { Varchar } 5

nama_jenis = 1 { Varchar } 30

detail_jenis = 1 { Text } 100

solusi_jenis = 1 { Text } 100

Number = [0...9]

Char = [A...Z] | [a...z] | [0...9]

d. Tabel Konsultasi

Konsultasi = id_konsultasi + id_jenis + id_ciri

id_konsultasi = 1 { Integer } 11

id_jenis = 1 { Varchar } 5

id_ciri = 1 { Integer } 11

Number = [0...9]

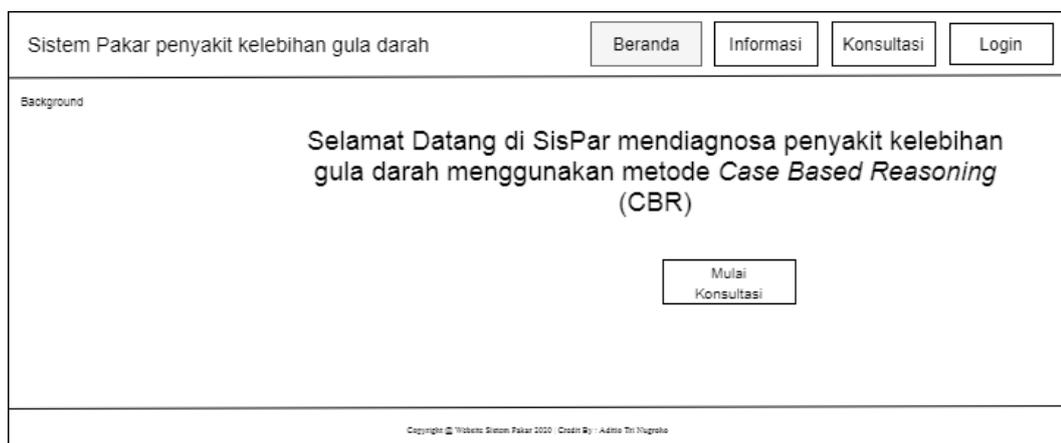
Char = [A...Z] | [a...z] | [0...9]

7. *Design User Interface/Desain Antarmuka*

Design User Interface/desain antar muka berfungsi untuk menjelaskan fungsi dan kontrol yang digunakan pada sistem untuk proses konsultasi, penyakit, gejala, aturan, dan rekaman konsultasi.

a. *Desain Beranda User*

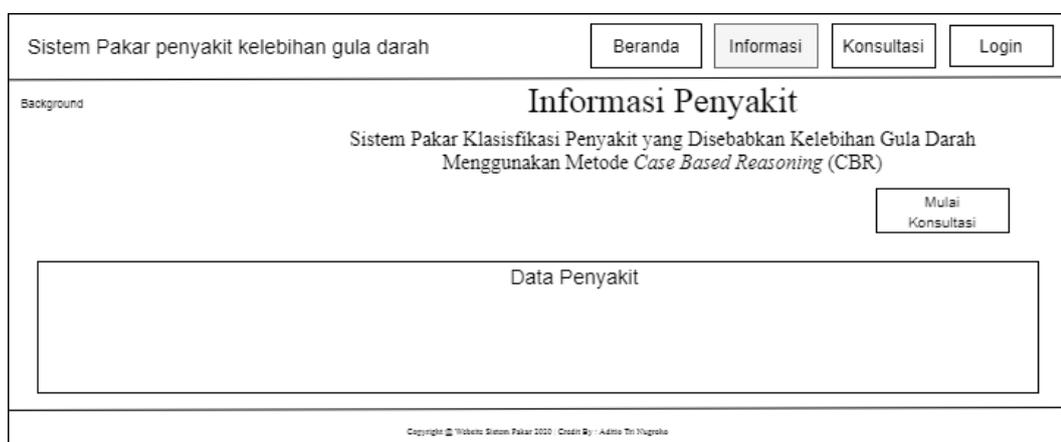
Desain beranda *user* adalah desain tampilan antar muka *user* pada awal tampilan aplikasi sistem pakar.



Gambar 4.31. *Desain Beranda User*

b. *Desain Informasi Penyakit*

Desain informasi penyakit adalah desain tampilan informasi data penyakit yang meliputi pengertian penyakit dan solusi pencegahannya.



Gambar 4.32. *Desain Informasi Penyakit*

c. Desain Halaman Konsultasi

Desain halaman konsultasi merupakan desain tampilan menu konsultasi.

Gambar 4.33. Desain Halaman Konsultasi

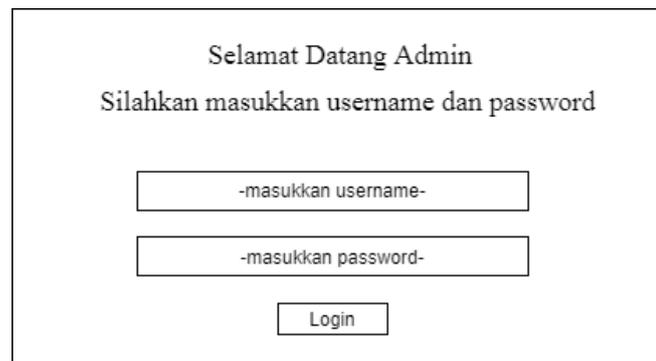
d. Desain Hasil Diagnosa

Desain hasil diagnosa merupakan desain tampilan untuk menu hasil diagnosa.

Gambar 4.34. Desain Hasil Diagnosa

e. Desain Login Admin

Desain login admin merupakan desain tampilan login ketika admin melakukan login.



Selamat Datang Admin
Silahkan masukkan username dan password

-masukkan username-

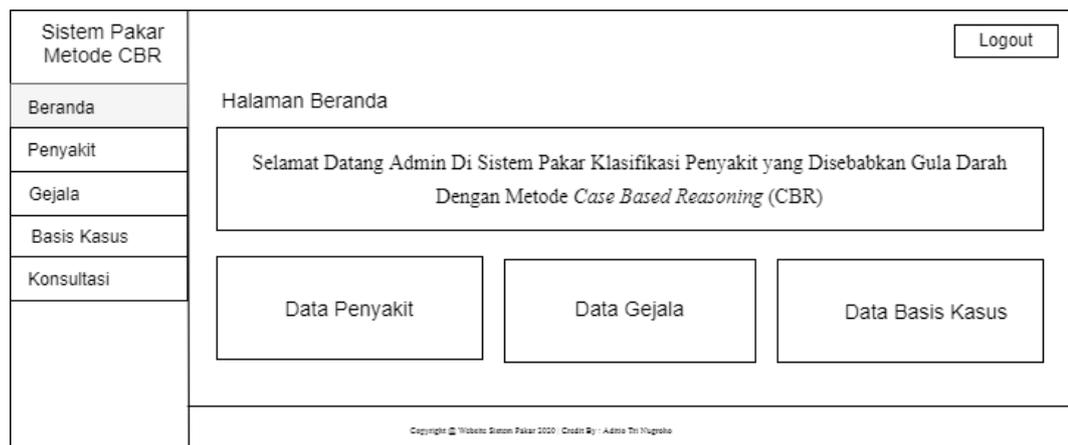
-masukkan password-

Login

Gambar 4.35. Desain Login Admin

f. Desain Halaman Beranda Admin

Desain halaman beranda admin merupakan desain tampilan beranda admin.



Sistem Pakar Metode CBR	Logout
Beranda	Halaman Beranda
Penyakit	Selamat Datang Admin Di Sistem Pakar Klasifikasi Penyakit yang Disebabkan Gula Darah Dengan Metode <i>Case Based Reasoning</i> (CBR)
Gejala	
Basis Kasus	
Konsultasi	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; width: 30%;">Data Penyakit</div> <div style="border: 1px solid black; padding: 5px; width: 30%;">Data Gejala</div> <div style="border: 1px solid black; padding: 5px; width: 30%;">Data Basis Kasus</div> </div>
	Copyright © 2020 Sistem Pakar 2020 Created By : Adina Tri Nugroho

Gambar 4.36. Desain Halaman Beranda Admin

g. Desain Halaman Data Penyakit

Desain halaman data penyakit merupakan desain tampilan halaman data penyakit untuk admin memasukkan data baru atau menghapus data lama dan merubah data.

Sistem Pakar Metode CBR	<input type="button" value="Logout"/>															
Beranda	Data Penyakit <input type="button" value="Tambah"/>															
Penyakit	Tabel Penyakit <input type="button" value="Search"/>															
Gejala	<table border="1"> <thead> <tr> <th>No.</th> <th>ID Penyakit</th> <th>Nama Penyakit</th> <th>Detail</th> <th>Solusi</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td colspan="5" style="text-align: right;"> <input type="button" value="edit"/> <input type="button" value="hapus"/> </td> </tr> </tbody> </table>					No.	ID Penyakit	Nama Penyakit	Detail	Solusi	Aksi	<input type="button" value="edit"/> <input type="button" value="hapus"/>				
No.	ID Penyakit	Nama Penyakit	Detail	Solusi	Aksi											
<input type="button" value="edit"/> <input type="button" value="hapus"/>																
Basis Kasus	<small>Copyright © 2020 Sistem Pakar 2020 Credit © Adito Tri Nugroho</small>															
Konsultasi																

Gambar 4.37. Desain Halaman Data Penyakit

h. Desain Halaman Tambah Data Penyakit

Desain halaman tambah merupakan desain tampilan halaman tambah data penyakit baru.

Sistem Pakar Metode CBR	Tambah Data Penyakit		<input type="button" value="Logout"/>
Beranda	ID Penyakit	Nama Penyakit	
Penyakit	<input type="text"/>	<input type="text"/>	
Gejala	Detail Penyakit	Solusi Penyakit	
Basis Kasus	<input type="text"/>	<input type="text"/>	
Konsultasi	<input type="button" value="Simpan"/>		
	<small>Copyright © 2020 Sistem Pakar 2020 Credit © Adito Tri Nugroho</small>		

Gambar 4.38. Desain Halaman Tambah Data Penyakit

i. Desain Halaman Edit Data Penyakit

Desain halaman edit data penyakit merupakan desain tampilan halaman untuk mengedit data penyakit.

Sistem Pakar Metode CBR	Logout	
Beranda	Edit Data Penyakit	
Penyakit	ID Penyakit	Nama Penyakit
Gejala	<input type="text"/>	<input type="text"/>
Basis Kasus	Detail Penyakit	Solusi Penyakit
Konsultasi	<input type="text"/>	<input type="text"/>
	<input type="button" value="Simpan"/>	
<small>Copyright © 2020 Sistem Pakar 2020. Credit © Adito Tri Nugroho</small>		

Gambar 4.39. Desain Halaman Edit Data Penyakit

j. Desain Hapus Data Penyakit

Desain hapus data penyakit merupakan desain tampilan untuk menghapus data penyakit.

<p style="font-size: 1.2em; font-weight: bold;">Anda yakin menghapus data ini?</p> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <input type="button" value="OK"/> <input type="button" value="Cancel"/> </div>

Gambar 4.40. Desain Hapus Data Penyakit

k. Desain Halaman Data Gejala

Desain halaman data gejala merupakan desain tampilan halaman untuk data gejala.

Sistem Pakar Metode CBR	Logout														
Beranda	Data Gejala <input type="button" value="Tambah"/>														
Penyakit	Tabel Gejala <input type="button" value="Search"/>														
Gejala	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">No.</th> <th style="width: 15%;">ID Gejala</th> <th style="width: 40%;">Ciri-ciri</th> <th style="width: 15%;">Bobot</th> <th style="width: 20%;">Aksi</th> </tr> </thead> <tbody> <tr> <td colspan="5" style="text-align: right; padding-top: 10px;"> <input type="button" value="edit"/> <input type="button" value="hapus"/> </td> </tr> </tbody> </table>					No.	ID Gejala	Ciri-ciri	Bobot	Aksi	<input type="button" value="edit"/> <input type="button" value="hapus"/>				
No.	ID Gejala	Ciri-ciri	Bobot	Aksi											
<input type="button" value="edit"/> <input type="button" value="hapus"/>															
Basis Kasus	<small>Copyright © 2020 Sistem Pakar 2020. Credit © Adito Tri Nugroho</small>														
Konsultasi															

Gambar 4.41. Desain Halaman Data Gejala

l. Desain Halaman Tambah Data Gejala

Desain halaman tambah data gejala merupakan desain tampilan halaman untuk menambah data gejala.

Sistem Pakar Metode CBR	Tambah Data Gejala Logout
Beranda	Masukkan Gejala Baru
Penyakit	<input type="text"/>
Gejala	
Basis Kasus	Bobot
Konsultasi	* Masukkan Nilai dari 1 - 5
	<input type="text"/>
	<input type="button" value="Simpan"/> <small>*Semakin besar nilai bobot, semakin besar pengaruh gejala terhadap penyakit</small>
	<small>Copyright © Woboto Sistem Pakar 2020 Credit By : Adiro Tri Nugroho</small>

Gambar 4.42. Desain Halaman Tambah Data Gejala

m. Desain Halaman Edit Data Gejala

Desain halaman edit data gejala merupakan desain tampilan halaman untuk mengedit data gejala.

Sistem Pakar Metode CBR	Edit Data Gejala Logout
Beranda	Gejala
Penyakit	<input type="text"/>
Gejala	
Basis Kasus	Bobot
Konsultasi	* Masukkan nilai dari 1 - 5 !
	<input type="text"/>
	<input type="button" value="Simpan"/> <small>* Semakin besar nilai bobot, semakin besar pengaruh gejala terhadap penyakit.</small>
	<small>Copyright © Woboto Sistem Pakar 2020 Credit By : Adiro Tri Nugroho</small>

Gambar 4. 43. Desain Halaman Edit Data Gejala

n. Desain Hapus Data Gejala

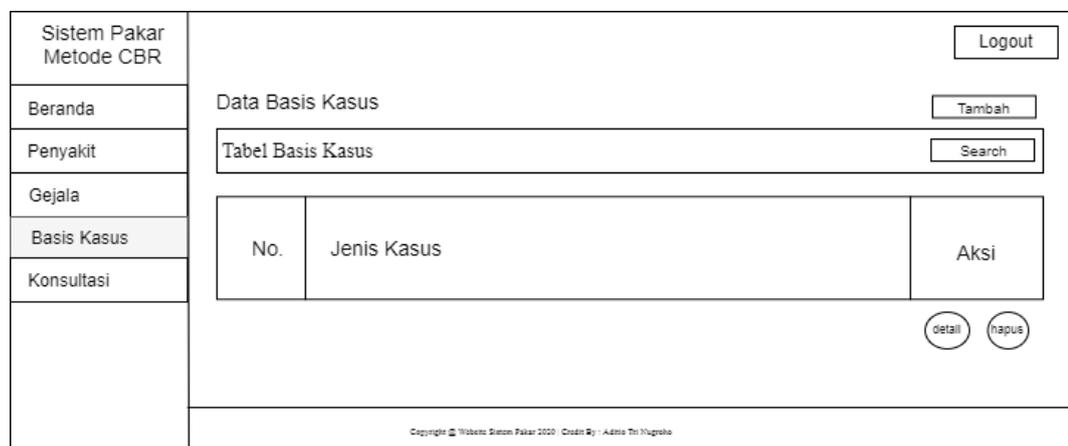
Desain hapus data gejala merupakan desain tampilan halaman untuk menghapus data gejala.



Gambar 4.44. Desain Hapus Data Gejala

o. Desain Halaman Data Basis Kasus

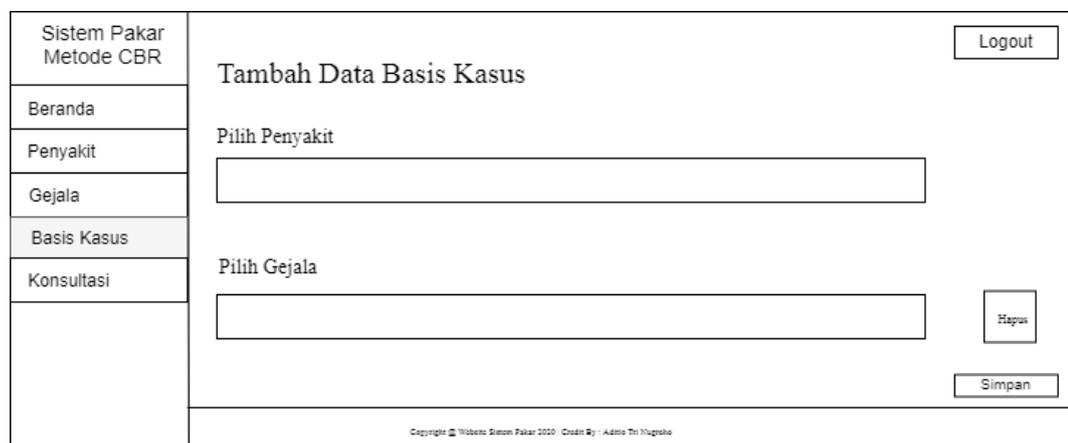
Desain halaman data basis kasus merupakan desain tampilan halaman untuk data basis kasus.



Gambar 4.45. Desain Halaman Data Basis Kasus

p. Desain Halaman Tambah Data Basis Kasus

Desain halaman tambah data basis kasus merupakan desain tampilan halaman untuk menambahkan data basis kasus.



Gambar 4.46. Desain Halaman Tambah Data Basis Kasus

q. Desain Halaman Detail Basis Kasus

Desain halaman detail basis kasus merupakan desain tampilan halaman untuk melihat data detail basis kasus.

Sistem Pakar Metode CBR	Detail Data Basis kasus				Logout
Beranda	ID Penyakit				
Penyakit					
Gejala	Data Gejala				
Basis Kasus	No.	ID Gejala	Gejala	Bobot	Aksi
Konsultasi					
	Copyright © 2020 Sistem Pakar 2020. Credit By : Adma Tri Nugroho				

Gambar 4.47. Desain Halaman Detail Basis Kasus

r. Desain Hapus Data Basis Kasus

Desain hapus data basis kasus merupakan desain tampilan halaman untuk menanyakan yakin ingin menghapus data.

Anda yakin menghapus data ini?

OK Cancel

Gambar 4.48. Desain Hapus Data Basis Kasus

s. Desain Halaman *Logout* Sistem

Desain halaman *logout* sistem merupakan desain tampilan untuk menanyakan akan keluar dari sistem.

Ready to Leave?

Select "Logout" jika ingin keluar dari sistem

Cancel Logout

Gambar 4.49. Desain Halaman *Logout* Sistem

B. Hasil Tahapan (Pembangunan)

Pada tahapan ini, dilakukan pembangunan aplikasi berdasarkan *input use case* yang telah dibuat pada tahapan sebelumnya dengan mengimplementasikan pada sebuah bahasa pemrograman. Pada penelitian ini bahasa pemrograman yang digunakan adalah *PHP*.

1. Tampilan Halaman Beranda Sistem

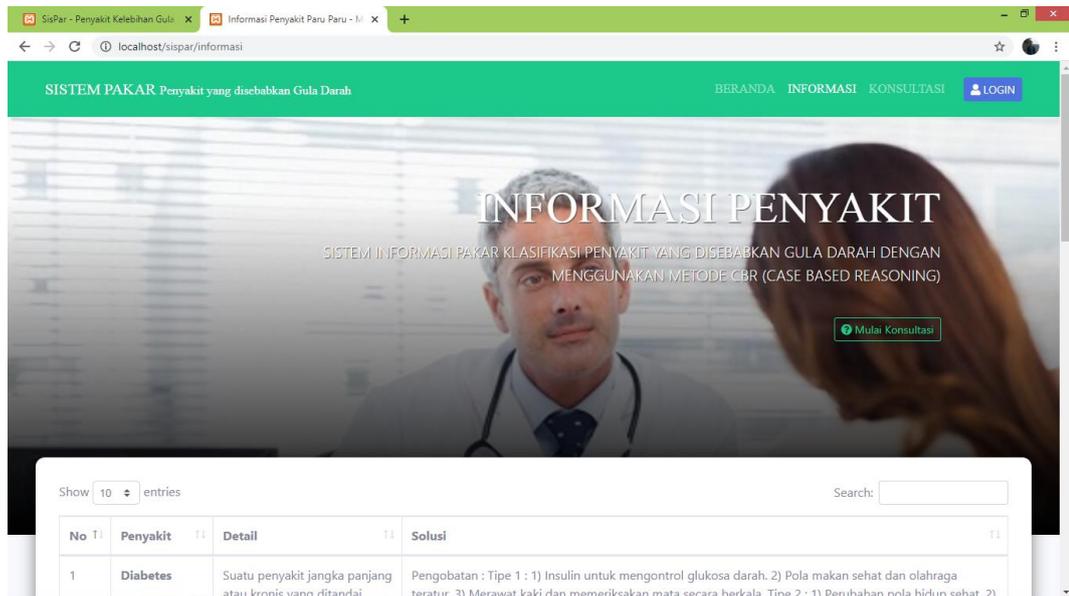
Tampilan halaman beranda merupakan hasil tampilan halaman beranda sistem.



Gambar 4.50. Tampilan Halaman Beranda *User*

2. Tampilan Halaman Informasi Penyakit

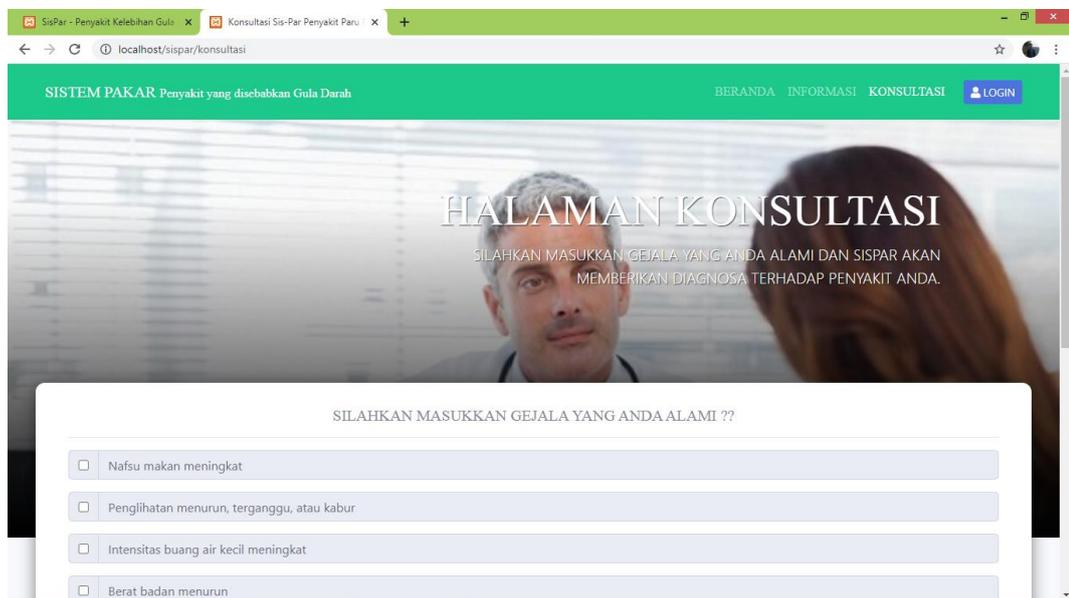
Tampilan halaman informasi merupakan hasil tampilan halaman informasi untuk melihat data penyakit beserta gejala dan solusinya.



Gambar 4.51. Tampilan Halaman Informasi Penyakit

3. Tampilan Halaman Konsultasi

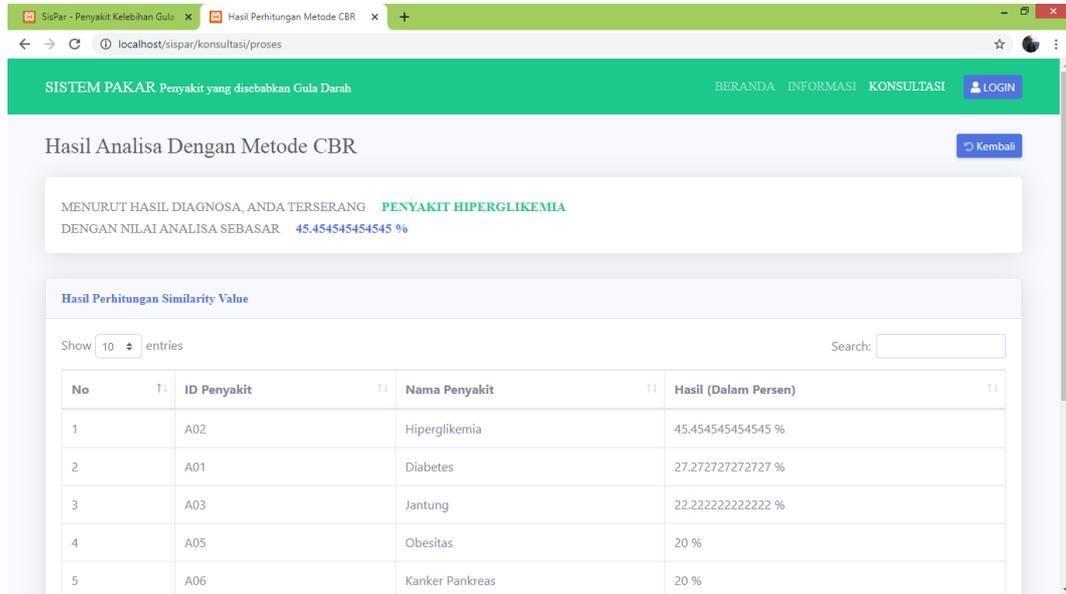
Tampilan halaman konsultasi merupakan hasil tampilan halaman konsultasi.



Gambar 4.52. Tampilan Halaman Konsultasi

4. Tampilan Halaman Hasil Diagnosa

Tampilan halaman hasil diagnosa merupakan hasil tampilan hasil diagnosa.



SISTEM PAKAR Penyakit yang disebabkan Gula Darah

BERANDA INFORMASI KONSULTASI LOGIN

Hasil Analisa Dengan Metode CBR [Kembali](#)

MENURUT HASIL DIAGNOSA, ANDA TERSEKANG **PENYAKIT HIPERGLIKEMIA**
DENGAN NILAI ANALISA SEBASAR **45.454545454545 %**

Hasil Perhitungan Similarity Value

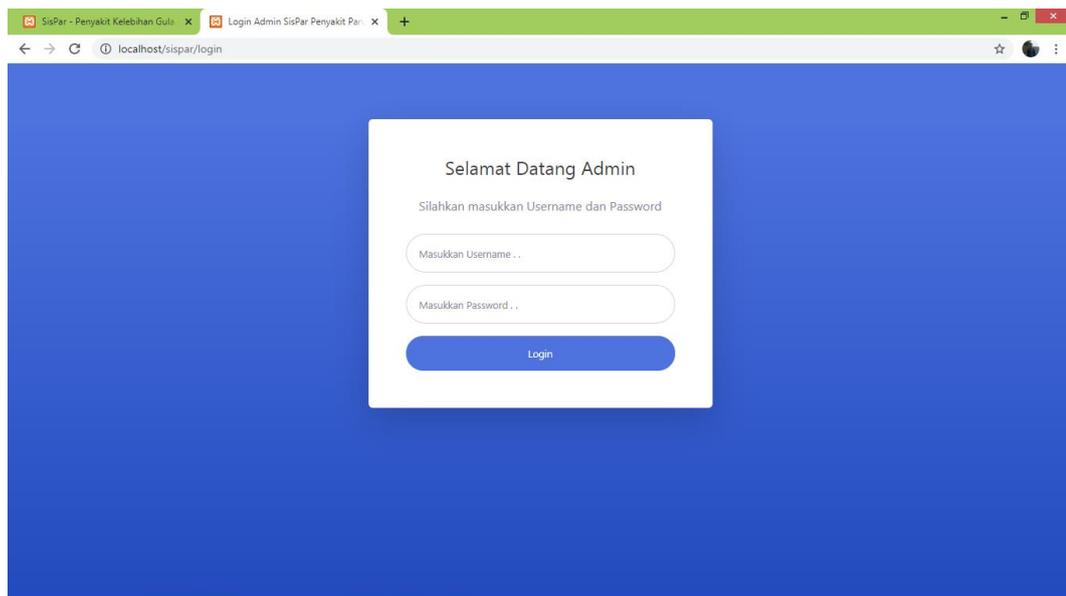
Show 10 entries Search:

No	ID Penyakit	Nama Penyakit	Hasil (Dalam Persen)
1	A02	Hiperglikemia	45.454545454545 %
2	A01	Diabetes	27.272727272727 %
3	A03	Jantung	22.222222222222 %
4	A05	Obesitas	20 %
5	A06	Kanker Pankreas	20 %

Gambar 4.53. Tampilan Halaman Hasil Diagnosa

5. Tampilan Halaman *Login* Admin

Tampilan halaman login admin merupakan hasil tampilan sistem login admin.



SisPar - Penyakit Kelebihan Gula

Login Admin SisPar Penyakit Par

localhost/sispar/login

Selamat Datang Admin

Silahkan masukkan Username dan Password

Masukkan Username ..

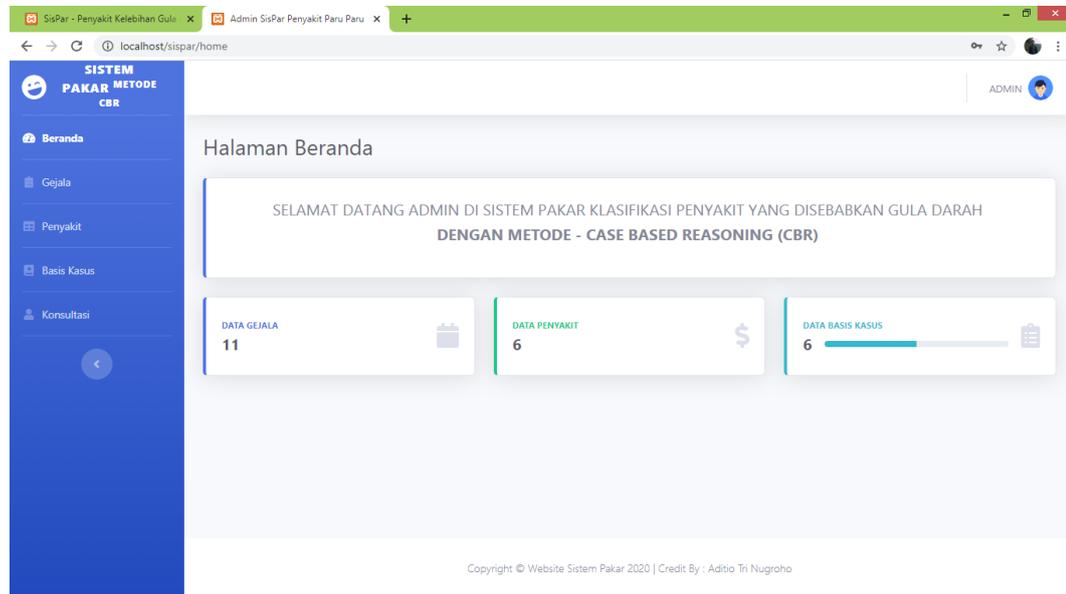
Masukkan Password ..

Login

Gambar 4.54. Tampilan Halaman *Login* Admin

6. Tampilan Halaman Beranda Admin

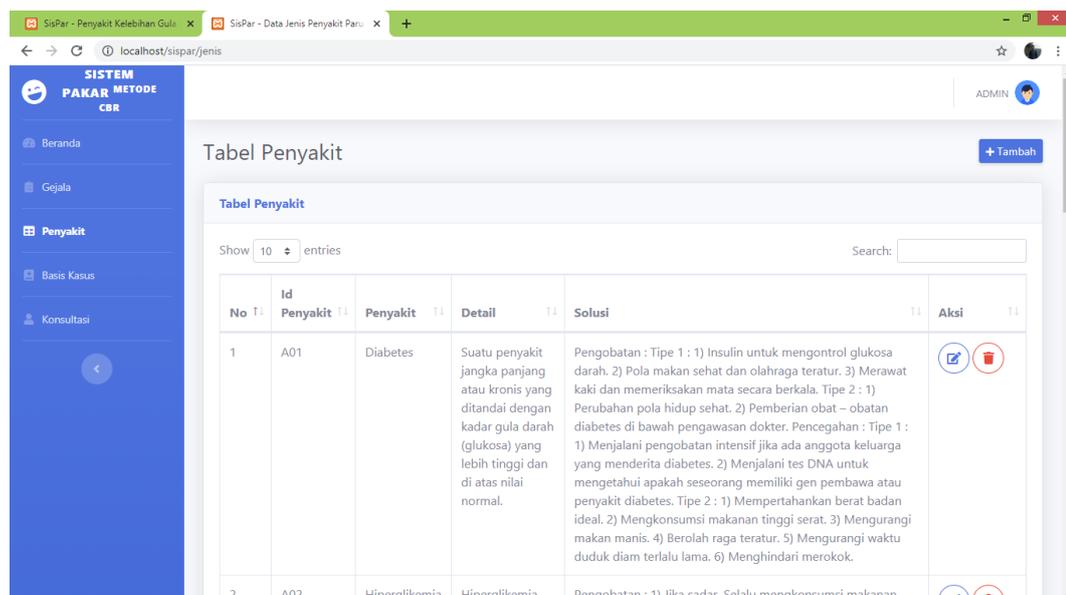
Tampilan halaman beranda admin merupakan hasil tampilan beranda admin.



Gambar 4.55. Tampilan Halaman Beranda Admin

7. Tampilan Halaman Data Penyakit

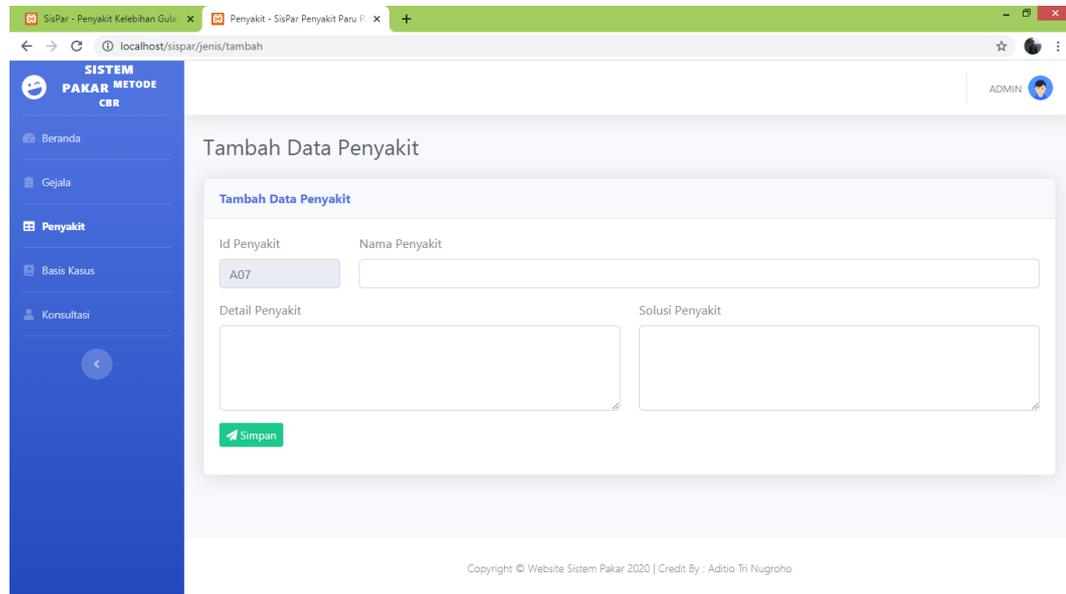
Tampilan halaman data penyakit merupakan hasil tampilan halaman data penyakit.



Gambar 4.56. Tampilan Halaman Data Penyakit

8. Tampilan Halaman Tambah Data Penyakit

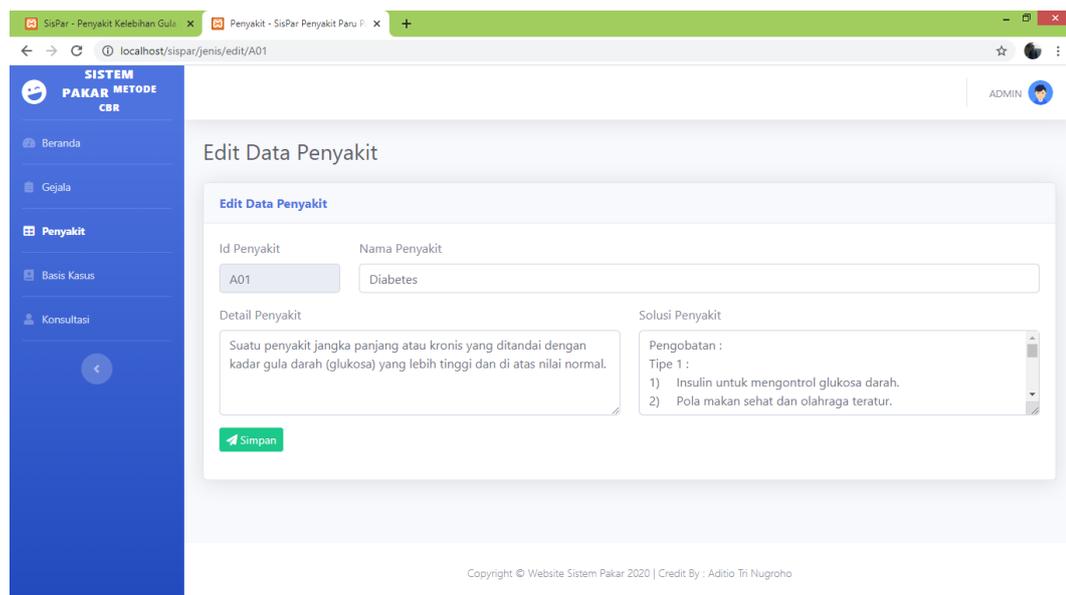
Tampilan halaman tambah data penyakit merupakan hasil sistem tampilan halaman tambah data penyakit.



Gambar 4.57. Tampilan Halaman Tambah Data Penyakit

9. Tampilan Halaman Edit Data Penyakit

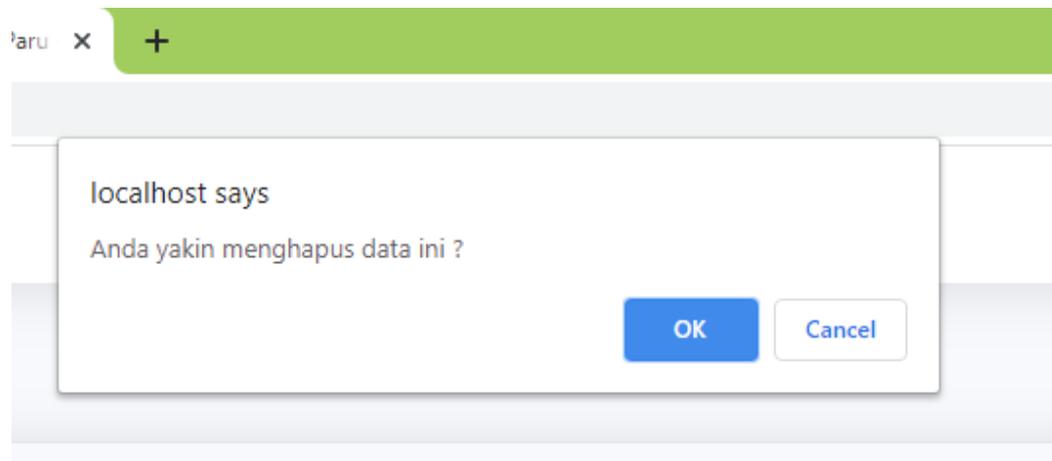
Tampilan halaman edit data penyakit merupakan hasil sistem tampilan halaman edit data penyakit.



Gambar 4.58. Tampilan Halaman Edit Data Penyakit

10. Tampilan Halaman Hapus Data Penyakit

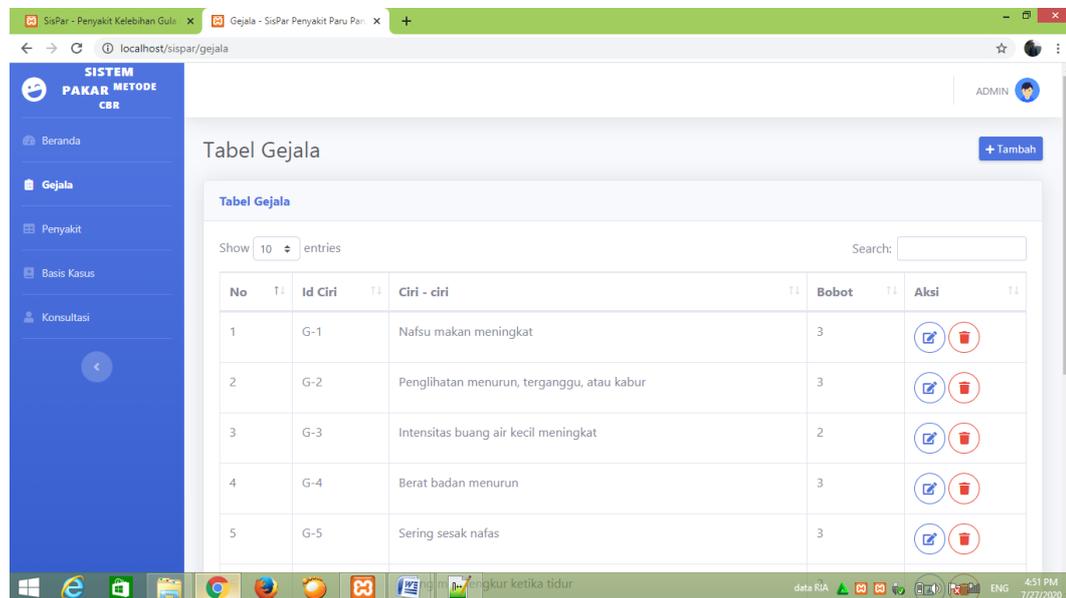
Tampilan halaman hapus data penyakit merupakan hasil sistem tampilan halaman hapus data penyakit.



Gambar 4.59. Tampilan Halaman Hapus Data Penyakit

11. Tampilan Halaman Data Gejala

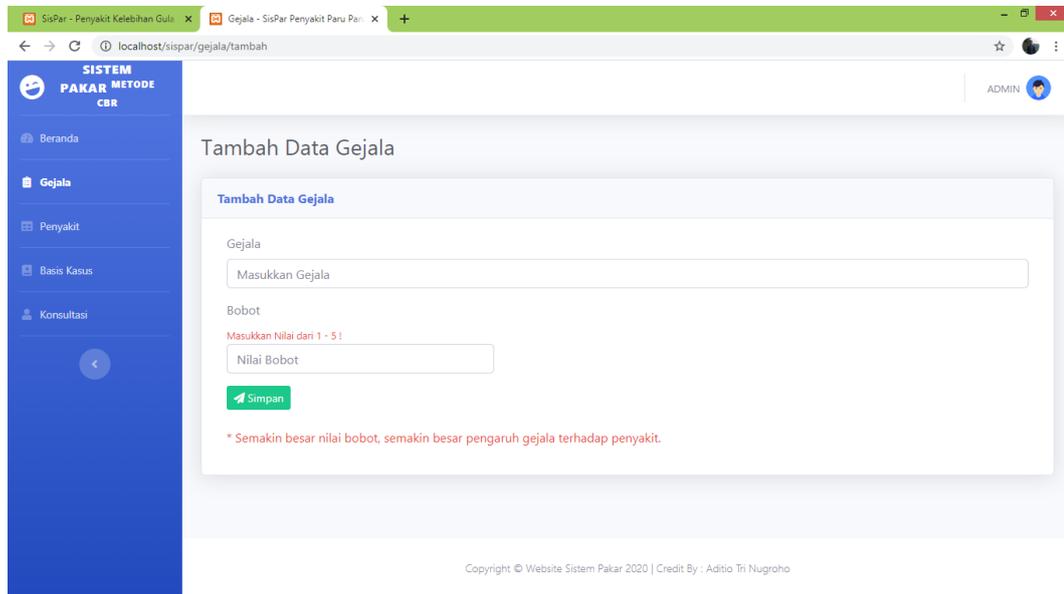
Tampilan halaman data gejala merupakan hasil sistem tampilan halaman data gejala.



Gambar 4.60. Tampilan Halaman Data Gejala

12. Tampilan Halaman Tambah Data Gejala

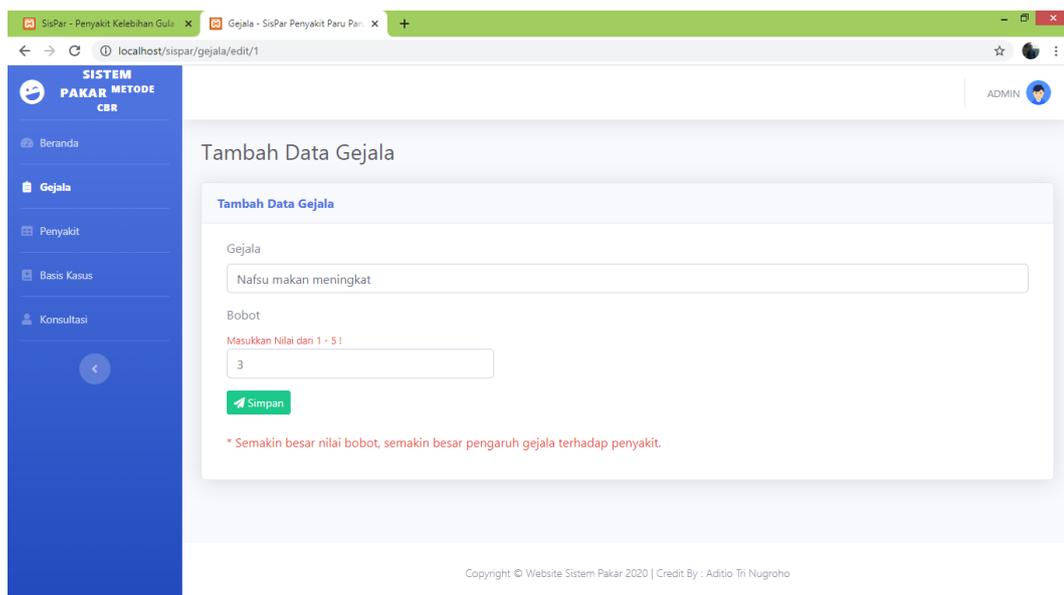
Tampilan halaman tambah data gejala merupakan hasil sistem tampilan halaman tambah data gejala.



Gambar 4.61. Tampilan Halaman Tambah Data Gejala

13. Tampilan Halaman Edit Data Gejala

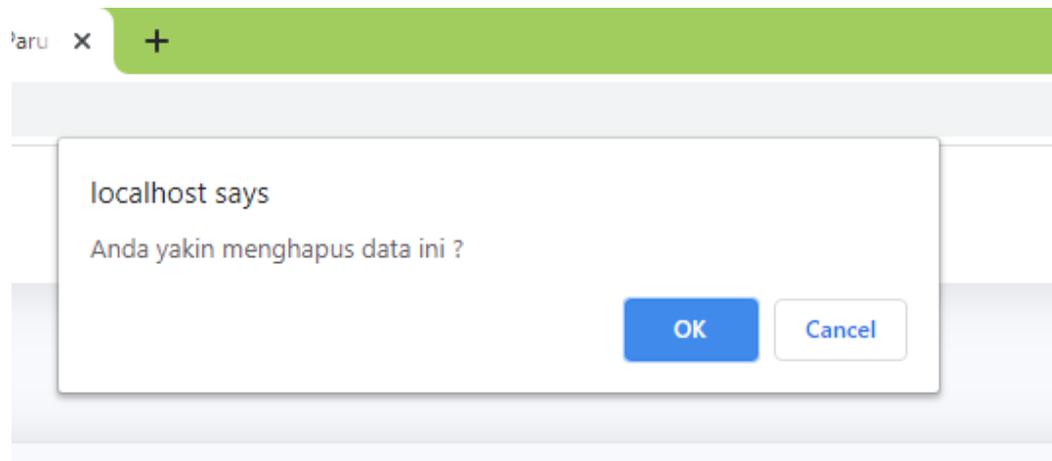
Tampilan halaman edit data gejala merupakan hasil sistem tampilan halaman edit data gejala.



Gambar 4.62. Tampilan Edit Data Gejala

14. Tampilan Halaman Hapus Data Gejala

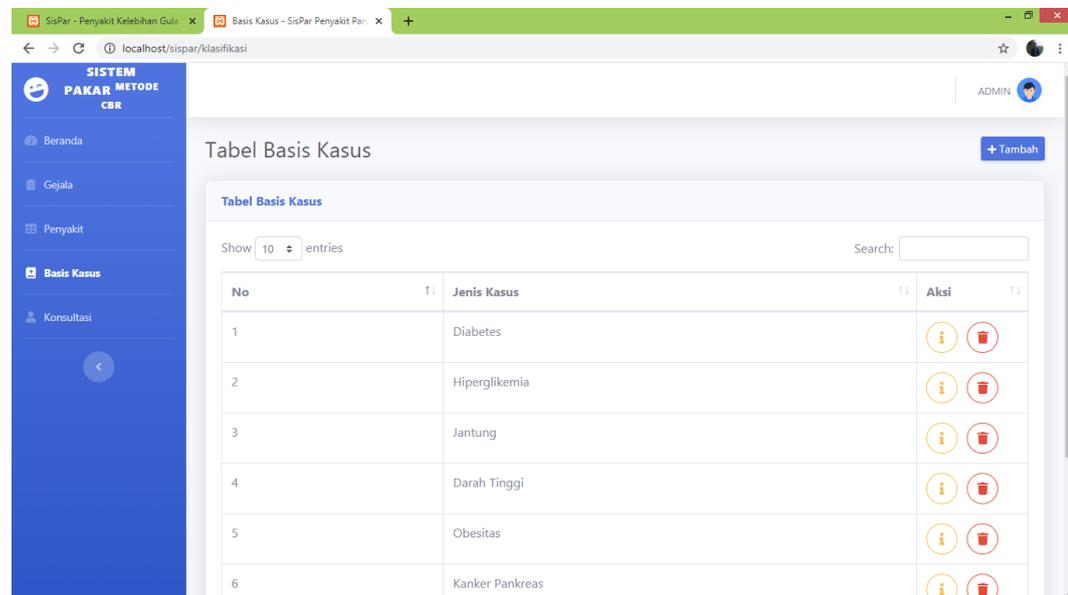
Tampilan halaman hapus data gejala merupakan hasil sistem tampilan halaman hapus data gejala.



Gambar 4.63. Tampilan Halaman Hapus Data Gejala

15. Tampilan Halaman Data Basis Kasus

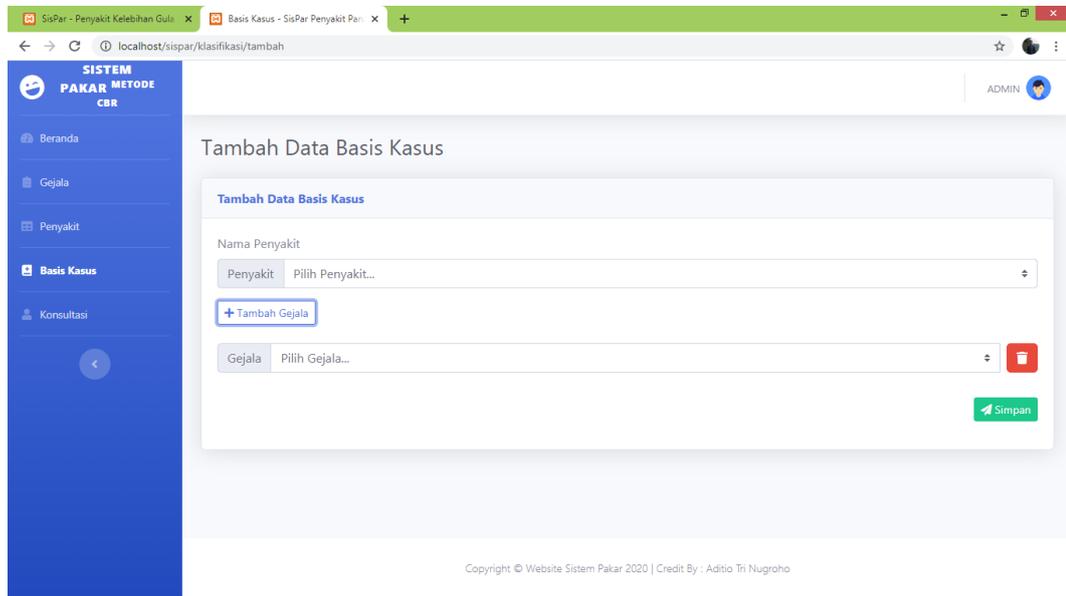
Tampilan halaman data basis kasus merupakan hasil sistem tampilan halaman data basis kasus.



Gambar 4.64. Tampilan Halaman Data Basis Kasus

16. Tampilan Halaman Tambah Data Basis Kasus

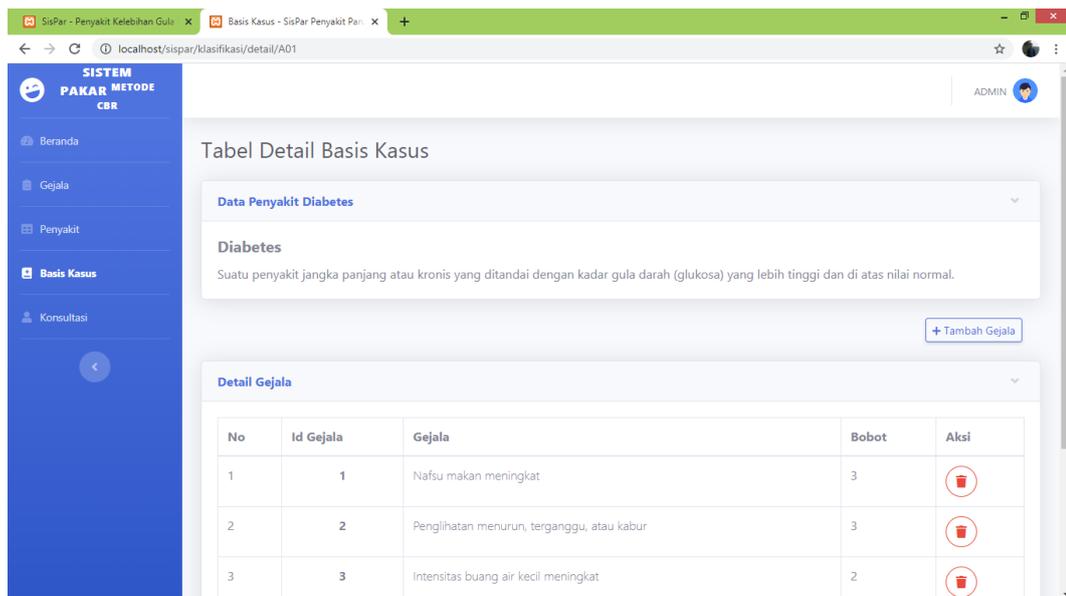
Tampilan halaman tambah data basis kasus merupakan hasil sistem tampilan halaman tambah data basis kasus.



Gambar 4.65. Tampilan Halaman Tambah Data Basis Kasus

17. Tampilan Halaman Detail Basis Kasus

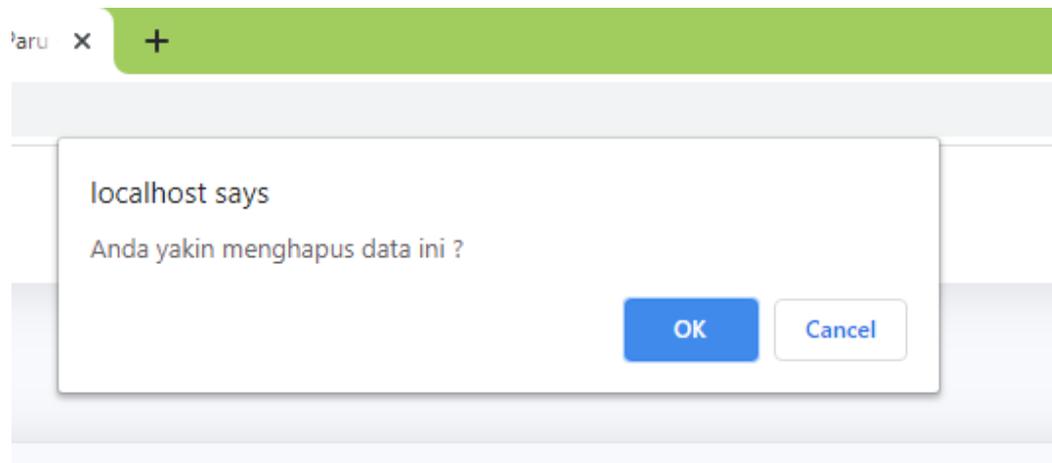
Tampilan halaman detail basis kasus merupakan hasil sistem tampilan halaman detail basis kasus.



Gambar 4.66. Tampilan Halaman Detail Basis Kasus

18. Tampilan Halaman Hapus Data Basis Kasus

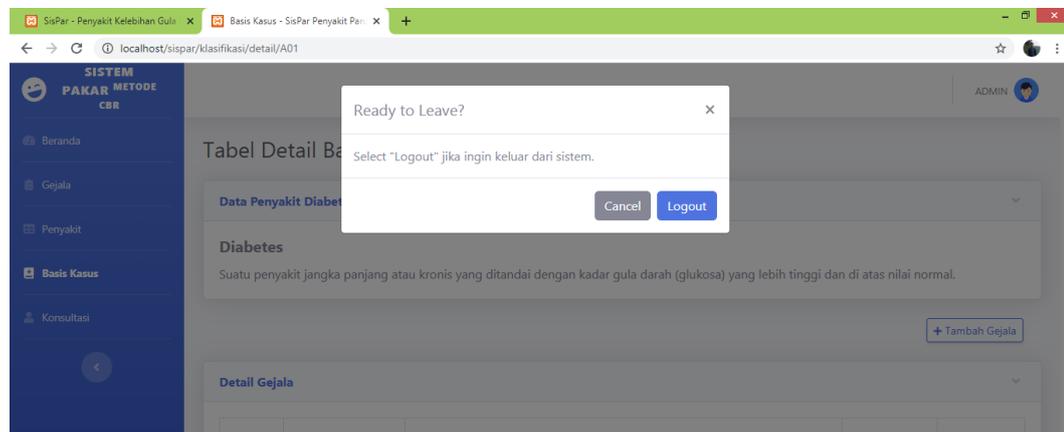
Tampilan halaman hapus data basis kasus merupakan hasil sistem tampilan halaman hapus data basis kasus.



Gambar 4.67. Tampilan Halaman Hapus Data Basis Kasus

19. Tampilan Halaman *Logout* Sistem

Tampilan halaman *logout* sistem merupakan hasil sistem tampilan halaman *logout* sistem.



Gambar 4.68. Tampilan Halaman *Logout* Sistem

C. Pengujian

1. *White Box Testing*

White Box Testing adalah salah satu cara untuk menguji suatu aplikasi atau software dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang dibuat ada yang salah atau tidak. Kalau modul yang telah dan sudah di hasilkan berupa *output* yang tidak sesuai dengan yang diharapkan maka akan dikomplikasi ulang dan dicek kembali kode-kode tersebut hingga sesuai dengan yang diharapkan. Berikut adalah hasil pengujian yang dilakukan:

Pengujian *White Box Testing* Konsultasi

Tabel 4.10. Skrip Hasil Konsultasi

1	<pre>//Konsultasi kasus baru oleh user <form class="user" method="post" action="<?=base_url('konsultasi/proses'); ?>"> <?php \$no = 1; foreach (\$ciri as \$row) { ?> <div class="input-group mb-3"> <div class="input-group-prepend"> <div class="input-group-text"> <input type="checkbox" name="ciri[]" id="ciri[]" value="<?= \$row['id_ciri']; ?>"> </div> </div> <input readonly type="text" name="labelciri" class="form- control" value="<?= \$row['nama_ciri']; ?>"> </div> <?php }; ?>
 <button type="submit" class="btn btn-primary"><i class="fas fa- sync-alt"></i>&nbsp;&nbsp;Proses</button> </form></pre>
2	<pre>// Validasi data yang dimasukkan if (!\$this->input->post('ciri', true)) { redirect('konsultasi');</pre>

	<pre> } else { \$data2 = \$this->konsul->getJenis(); </pre>
3	<pre> //Hitung similarity dengan KNN \$jml = 0; \$nilai = 0; \$jenis = \$row['id_jenis']; \$kasus = \$this->konsul->getPenyakit(\$jenis); \$penyakit = \$kasus['nama_jenis']; \$dipilih = count(\$this->input->post('ciri', true)); // Perulangan perhitungan metode CBR foreach (\$this->input->post('ciri', true) as \$selected) { \$ciri = \$selected; \$row = \$this->konsul->getCiri(\$ciri); \$get = \$this->konsul->getSama(\$jenis, \$ciri); if (isset(\$get)) { \$jml += 1; \$nilai += (1 * \$get['bobot']); } else { \$jml += 0; } } \$pembagi = \$this->konsul->getPembagi(\$jenis); \$jml_ciri = \$this->konsul->getJmlCiri(\$jenis); \$hasil = \$nilai / "\$pembagi"; </pre>
4	<pre> //Hasil hitung similarity kasus \$final[\$i] = array('id_jenis' => \$jenis, 'nama_jenis' => \$penyakit, 'jml_cocok' => \$jml, 'jml_gejala' => \$jml_ciri, 'jml_dipilih' => \$dipilih, 'bobot_sama' => \$nilai, 'total_bobot' => \$pembagi, 'hasil' => \$hasil); </pre>
5	<pre> //Menampilkan hasil diagnosa <div class="card shadow mb-4 alert"> </pre>

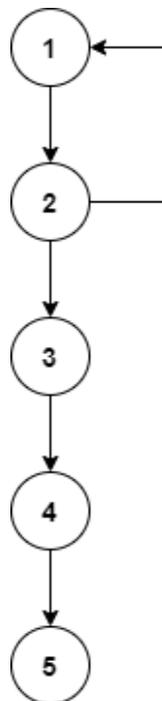
```

<?php
    // Nilai Perhitungan Terbesar
    $max = max(array_column($final, 'hasil'));
    $key = array_search($max, array_column($final,
'hasil'));
    ?>
<h5 class="mt-3" style="font-weight:lighter; font-size:17px; text-
transform:uppercase">Menurut Hasil Diagnosa, Anda Terserang
<span style="font-weight:bold" class="text-success ml-3">
Penyakit <?= $final[$key]['nama_jenis']; ?></span>
</h5>

<h5 style="font-weight:lighter; font-size:17px; text-
transform:uppercase">
    Dengan Nilai Analisa Sebesar <span class="text-
primary ml-3" style="font-weight:bold"><?= $final[$key]['hasil']
* 100; ?> %</span>
</h5>
</div>

```

Basis Path Konsultasi



Gambar 4.69. *Basis Path* Konsultasi

a. *Complexiti Cyclometric*

$$V(G) = E - N + 2$$

$$V(G) = 5 - 5 + 2$$

$$V(G) = 2$$

Keterangan :

$V(G)$: Cyclomatic complexity untuk flow graph G

E : Panah (*Edge*)

N : Lingkaran (*Node*)

b. *Independent Path*

Berdasarkan hasil independent path pada contoh diatas dapat dijabarkan sebagai berikut :

Path 1 : 1, 2, 1

Path 2 : 1, 2, 3, 4, 5

2. *Black Box Testing*

Black Box Testing merupakan pendekatan komplementer dari teknik *white box*, metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Dibawah ini merupakan metode *black box* dari perangkat lunak yang dibuat.

Tabel 4.11. Pengujian *Black Box Testing*

Test Pengujian	Tujuan	Hasil yang diharapkan	Pengujian	
			Valid	Tidak Valid
Menu Beranda	Pasien dapat melihat tampilan beranda dan keterangan cara sistem pakar bekerja	Sistem dapat berjalan dengan baik untuk menampilkan menu beranda dan keterangan cara sistem pakar bekerja		
Menu Informasi	Pasien dapat melihat informasi data gejala dan penyakit lama	Sistem dapat berjalan dengan baik untuk menampilkan halaman menu informasi		
Menu Konsultasi	Pasien memilih gejala yang dirasakan untuk mengetahui hasil analisa penyakit	Sistem dapat berjalan dengan baik untuk menampilkan halaman menu konsultasi		
Menu Login	Admin dapat masuk kedalam sistem	Sistem dapat berjalan untuk menampilkan isi aplikasi dan menampilkan pesan “ <i>login berhasil</i> ”		
Menu Data Penyakit	Admin bisa menambah, mengubah, melihat dan menghapus data penyakit	Sistem dapat berjalan dengan baik untuk menampilkan halaman menu kelola data penyakit		
Menu Data Gejala	Admin bisa menambah, mengubah, melihat dan menghapus data gejala	Sistem dapat berjalan dengan baik untuk menampilkan halaman menu kelola data gejala		
Menu Data Basis Kasus	Admin bisa menghapus dan menambah data gejala	Sistem dapat berjalan dengan baik untuk menampilkan halaman menu basis kasus		

Berdasarkan hasil pengujian pada tabel 4.11. tersebut dapat diketahui presentase ketercapaian sebagai berikut :

Tercapai = $11/11 \times 100\% = 100\%$

Gagal = $0/11 \times 100\% = 0\%$

Maka dari itu hasil pengujian *Black Box* yang didapatkan dengan interpretasi 100% layak digunakan.

3. *User Acceptance*

User Acceptance dilakukan untuk mengetahui tentang pendapat pengguna dari sisi Admin maupun Pasien untuk aplikasi Sistem Pakar untuk Mendiagnosa Penyakit yang disebabkan oleh Kelebihan Gula Darah Berbasis *Web* dengan Menggunakan Metode *Case Based Reasoning* (CBR) dari segi penggunaan dan manfaat sistem. *User Acceptance* dilakukan pada 5 responden. Instrumen *user acceptance* dapat dilihat pada lampiran.

Berikut penjelasan skor *user acceptance* untuk masing – masing pertanyaan.

- a. Skor 5 (sangat setuju)
- b. Skor 4 (setuju)
- c. Skor 3 (ragu - ragu)
- d. Skor 2 (tidak setuju)
- e. Skor 1 (sangat tidak setuju)

1) Berikut hasil pertanyaan dari segi kemanfaatan (usefulness) Admin :

- a) Apakah sistem pakar untuk mendiagnosa penyakit kelebihan gula darah ini bermanfaat bagi admin dari segi pelayanan mendiagnosa penyakit kelebihan gula darah?
- b) Apakah sistem pakar untuk mendiagnosa penyakit kelebihan gula darah ini bermanfaat bagi pasien dari segi kinerja mendiagnosa penyakit kelebihan gula darah?
- c) Apakah sistem pakar untuk mendiagnosa penyakit kelebihan gula darah ini bermanfaat bagi pasien dari segi efektifitas mendiagnosa penyakit kelebihan gula darah?

- d) Apakah sistem pakar untuk mendiagnosa penyakit kelebihan gula darah ini bermanfaat bagi pasien dari segi kegunaan mendiagnosa penyakit kelebihan gula darah?

Tabel 4.12. *User Acceptance* dari segi kemanfaatan Admin

Responden	Pertanyaan dan Skor			
	a	b	c	d
1	4	5	4	5
Jumlah	4	5	4	5
Presentase	80%	100%	80%	100%
Rata-rata Presentase	90%			

- 2) Berikut hasil pertanyaan dari segi kemanfaatan pasien :
- Apakah sistem pakar untuk mendiagnosa penyakit kelebihan gula darah bermanfaat bagi pasien dari segi pelayanan mendiagnosa penyakit kelebihan gula darah?
 - Apakah sistem pakar untuk mendiagnosa penyakit kelebihan gula darah ini bermanfaat bagi pasien dari segi kinerja mendiagnosa penyakit kelebihan gula darah?
 - Apakah sistem pakar untuk mendiagnosa penyakit kelebihan gula darah ini bermanfaat bagi pasien dari segi efektifitas mendiagnosa penyakit kelebihan gula darah?
 - Apakah sistem pakar untuk mendiagnosa penyakit kelebihan gula darah ini bermanfaat bagi pasien dari segi kegunaan dalam proses mendiagnosa penyakit kelebihan gula darah?

Tabel 4.13. *User Acceptance* dari segi kemanfaatan pasien

Responden	Pertanyaan dan Skor			
	1	2	3	4
2	4	4	5	4
3	5	4	4	5
4	4	5	4	5
5	4	5	5	5
Jumlah	17	18	18	19
Presentase	85%	90%	90%	95%
Rata-rata Presentase	90%			

- 3) Berikut daftar pertanyaan dan hasil pada hasil kemudahan penggunaan (*Easy Of Use*) :
- a) Apakah sistem pakar untuk mendiagnosa penyakit kelebihan guladarah ini mudah dipahami?
 - b) Apakah sistem pakar untuk mendiagnosa penyakit kelebihan gula darah ini mudah digunakan?

Tabel 4.14. Kemudahan Pengguna

Responden	Pertanyaan dan skor	
	1	2
1	5	4
2	4	5
3	4	4
4	4	4
Jumlah	17	17
Presentasi	85%	85%
Rata-rata Presentasi	85%	

Untuk mengetahui apakah aplikasi ini layak digunakan yaitu berdasarkan kriteria tersebut :

- 100% = Sangat Layak
- 80% - 99% = Layak
- 70% - 79% = Lumayan Layak
- 60% - 69% = Tidak Layak
- < 59% = Sangat Tidak Layak

Dari hasil rata – rata presentase tersebut dapat diketahui presentase ketercapaian sebagai berikut :

$$\begin{aligned} & \text{Total Jumlah Rata – Rata Presentase / Jumlah Aspek} \\ & = 265\% / 3 \\ & = 88\% \end{aligned}$$

Dengan presentase hasil *User Acceptance* 88% maka Sistem pakat untuk mendiagnosa penyakit kelebihan gula darah berbasis web dengan metode *Case Based Reasoning* (CBR) Di klinik Pratama Umum Asy-Syifa Wates Undaaan Kudus Layak Digunakan.

4. Validasi Ahli

Menguji kevalidan sistem pakar untuk mendiagnosa penyakit kelebihan gula darah berbasis web dengan metode *Case Based Reasoning* (CBR) di klinik Pratama Umum Undaan Kudus oleh para ahli.

Makna point validasi adalah sebagai berikut :

1 = Sangat kurang baik 3 = Cukup baik 5 = Sangat baik
2 = Kurang baik 4 = Baik

Tabel 4.15. Validasi Ahli

Indikator	Pernyataan	Penguji		
		1	2	3
Survei Sistem	Sistem pakar untuk mendiagnosa penyakit kelebihan gula darah sudah sesuai dengan kebutuhan data secara lengkap untuk menentukan problem-problem atau gejala yang timbul.			
Analisis Sistem	Hasil rancangan sistem pakar untuk mendiagnosa penyakit kelebihan gula darah ini sudah sesuai dengan sistem yang berjalan dan sistem dapat mendiagnosa penyakit kelebihan gula darah.			
Desain Sistem	Pembangunan sistem pakar untuk mendiagnosa penyakit kelebihan gula darah ini sudah sesuai dengan hasil perancangan sistem yang dibangun menggunakan bahasa pemrograman PHP.			
Implementasi Sistem	Hasil pengujian <i>white box</i> , <i>black box</i> , dan <i>user acceptance</i> sudah sesuai			

	dengan sistem yang ada.			
Jumlah				
Presentase = (jumlah nilai / nilai max) x 100%				
Rata – Rata presentase = jumlah presentase / jumlah penguji				

D. Pembahasan

Perancangan sistem menggunakan metode *waterfall* yang terdiri dari 5 langkah yaitu Survei sistem, Analisa Sistem, Desain sistem, Implementasi sistem, dan Analisa permasalahan berikut tahapan-tahapan dalam perancangan sistem.

1. Survei Sistem

Proses perancangan sistem ini melakukan analisis kebutuhan data secara lengkap untuk menentukan problem-problem atau gejala yang timbul. Pada hal ini belum adanya sistem pakar untuk mendeteksi penyakit kelebihan gula darah di klinik Pratama Umum 24 Jam Undaan Kudus.

2. Analisa Sistem

Tahap analisis bertitik tolak pada kegiatan-kegiatan dan tugas-tugas dimana sistem yang berjalan dipelajari lebih mendalam, konsepsi dan ulasan dibuat untuk menjadi landasan bagi sistem baru yang akan dibangun. Pada akhir tahap ini separuh kegiatan dari usaha pengembangan sistem pakar untuk mendiagnosa penyakit kelebihan gula darah diselesaikan. Salah satu tujuan terpenting pada tahap ini adalah untuk mendefinisikan sistem berjalan. Data yang dibutuhkan adalah data gejala yang menyebabkan penyakit kelebihan gula darah. Sistem dapat mendiagnosa apakah pasien terkena penyakit Diabetes,

3. Desain Sistem

Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras dan perangkat lunak. Pada tahap ini akan dibuat UML Aplikasi Sistem Pakar untuk Mendiagnosa Penyakit yang Disebabkan oleh Kelebihan Gula

Darah Berbasis *Web* dengan Metode *Case Based Reasoning* (CBR) meliputi *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *User Interface*.

4. Implementasi Sistem

Tahap ini adalah prosedur yang dilakukan untuk menyelesaikan desain sistem yang ada didalam dokumen desain sistem yang telah disetujui dan menguji, menginstal, dan memulai penggunaan sistem baru atau sistem yang telah diperbaiki. Tujuan dari tahap implementasi ini adalah untuk menyelesaikan desain sistem yang telah disetujui, menguji, serta mendokumentasikan program-program dan prosedur sistem yang diperlukan, memastikan bahwa staff yang terlibat dapat mengoperasikan sistem pakar untuk mendiagnosa penyakit kelebihan gula darah berbasis *web* dan memastikan bahwa konversi sistem lama ke sistem yang baru dapat berjalan secara baik dan benar.

5. Analisa Permasalahan

Permasalahan di dalam dunia kerja menjadi semakin kompleks dan rumit seiring dengan majunya teknologi, informasi, dan elektronik. Pada setiap permasalahan itu, tentu saja dibutuhkan solusi/penyelesaian yang tepat untuk kemudian dilakukan tindak lanjut secara cepat, tepat, efektif, dan akurat.

Oleh karena itu, dalam kehidupan manusia seperti yang telah dipaparkan pada awal tadi, maka kebutuhan jasa konsultasi pada seorang/badan yang merupakan pakar atau yang ahli dalam bidang tersebut menjadi begitu penting dan dibutuhkan.

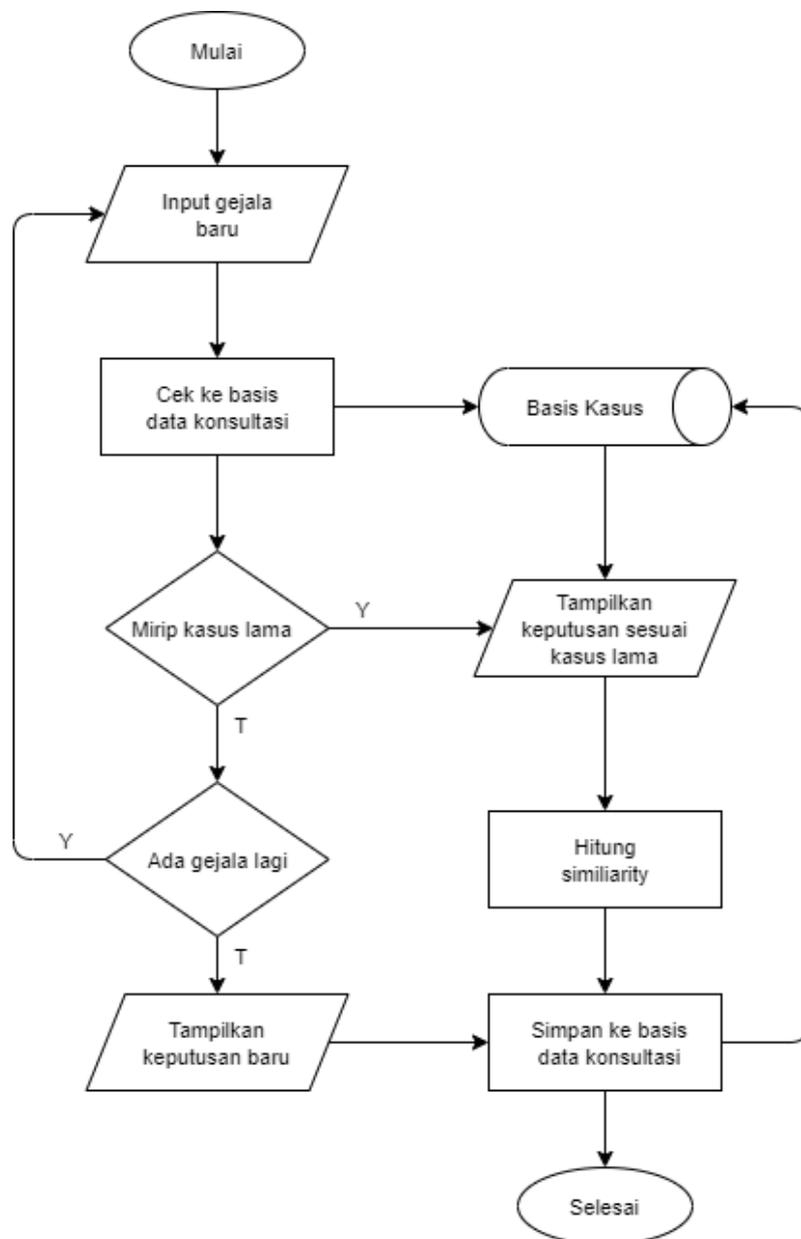
a. Alur aplikasi sistem pakar penyakit gula darah

Cara kerja dari alur aplikasi sistem pakar penyakit kelebihan gula darah adalah sebagai berikut :

- 1.) Menginput atau memilih gejala baru.
- 2.) Pengecekan masukan gejala baru ke dalam basis kasus.
- 3.) Jika masukan gejala baru memiliki kemiripan dengan gejala pada kasus lama maka *output* akan menampilkan sesuai dengan kasus lama.
- 4.) Hasil keputusan kasus lama kemudian di analisa dengan membandingkan gejala yang mirip dengan kasus baru sehingga

informasi yang didapat akan digunakan untuk menghitung nilai kemiripan dari penyakit.

- 5.) Jika masukan gejala baru tidak memiliki kemiripan dengan gejala kasus sebelumnya (kasus lama) atau masih ada gejala yang mau diinputkan maka harus memasukan gejala baru lainnya yang masih ada.
- 6.) Jika sudah tidak ada gejala yang dimiliki maka *output* akan menampilkan keputusan baru yang kemudian disimpan dalam konsultasi dan laporan.



Gambar 4.70. Alur Aplikasi Sistem Pakar Penyakit Gula Darah
Fakta dengan Kode

Perancangan sistem pakar yang penulis buat ini terdiri dari data gejala dan data penyakit yang merupakan sebuah objek yang sangat penting guna membantu proses perancangan sistem pakar berbasis web diagnosa. Berikut adalah tabel data kode gejala dan tabel data kode penyakit yang penulis rancang guna mempermudah dalam proses mengolah data yang dimasukkan ke dalam basis kasus.

Tabel 4.16. Kode Data Gejala

Kode	Gejala
G1	Nafsu makan meningkat
G2	Penglihatan menurun, terganggu, atau kabur
G3	Intensitas buang air kecil meningkat
G4	Berat badan menurun
G5	Sering sesak nafas
G6	Sering mendengkur ketika tidur
G7	Merasa mual-mual
G8	Mulut terasa kering
G9	Detak jantung yang tidak beraturan
G10	Terjadi perubahan warna kulit di leher atau lipatan
G16	Merasa mulas

Tabel 4.17. Kode Data Penyakit

Kode	Nama Penyakit
A1	Diabetes
A2	Hiperglikemia
A3	Jantung
A4	Darah Tinggi
A5	Obesitas
A6	Kanker Pankreas

a. Representasi Pengetahuan

Berdasarkan hasil identifikasi penyakit serta gejala-gejala yang diderita, maka dapat dibuat representasi pengetahuan yang ditunjukkan pada Tabel 4.18. sebagai berikut :

Tabel 4.18. Representasi

Kode	Gejala Penyakit	A1	A2	A3	A4	A5	A6
G1	Nafsu makan meningkat	√	√			√	
G2	Penglihatan menurun, terganggu, atau kabur	√	√		√		
G3	Intensitas buang air kecil meningkat	√		√			√
G4	Berat badan menurun	√	√				
G5	Sering sesak nafas			√	√	√	√
G6	Sering mendengkur ketika tidur			√		√	
G7	Merasa mual mual				√		
G8	Mulut terasa kering		√		√		√
G9	Detak jantung yang tidak beraturan			√	√	√	
G10	Tidak ada gejala dalam data yang dialami						
G16	Merasa mulas						√

b. Contoh Kasus

Dengan menganalisa data, sistem akan memiliki acuan untuk mengambil keputusan terhadap pasien. Membuat prediksi hasil keputusan menjadi lebih akurat.

Tabel 4.19. Data lama pasien

No Kasus	Variabel			Y=Hasil
	x1	x2	x3	
1	8	13	10	Baik
2	10	12	7	Buruk
3	5	9	12	Baik
4	7	11	9	Baik
5	4	7	6	Buruk

1. Tetapkan nilai k
Dalam hal $k = 3$
2. Hitung jarak antara *query* dan semua sampel kasus

Dengan kasus baru dimana :

$$X1=12, X2=10, X3=9$$

maka dapat dihitung jarak antara kasus baru dengan kasus lama

$$\text{Diketahui} = \sum W_k |x_{ik} - c_{jk}|$$

$$\text{Matriks baris pertama kolom pertama} : (8-12)+(13-10)+(10-9)=8$$

Tabel 4.20. Hasil Perhitungan Dengan *Manhattan Distance*

No Kasus	Variabel			Jumlah	Klasifikasi
	x1	x2	x3		
1	4	3	1	8	Baik
2	2	2	2	6	Buruk
3	7	1	3	11	Baik
4	5	1	0	6	Baik
5	8	3	3	14	Buruk

BAB V

PENUTUP

A. Kesimpulan

Dari penelitian ini dapat disimpulkan bahwa :

1. Aplikasi Sistem Pakar untuk Mendiagnosa Penyakit Kelebihan Gula Darah dirancang untuk dapat memberikan informasi kepada pasien tentang kemungkinan penyakit yang akan muncul dengan gejala yang sedang diderita oleh pasien.
2. Aplikasi berhasil dibuat dengan metode *waterfall* dan menggunakan *Unified Modelling Language* (UML).
3. Berdasarkan pengujian *Black Box*, aplikasi Sistem Pakar untuk Mendiagnosa Penyakit yang Disebabkan oleh Kelebihan Gula Darah ini layak digunakan dengan presentase 100% dapat berfungsi dengan baik.
4. Berdasarkan pengujian *White Box*, hasil perhitungan kompleksitas siklomatis konsultasi data yang dihasilkan adalah 2. Karena nilai tersebut kurang dari 10 berarti termasuk dalam algoritma yang tidak kompleks dan memenuhi kriteria rekayasa perangkat lunak.
5. Berdasarkan *User Acceptance* yang dilakukan pada lima responden penguji. Aplikasi sistem pakar untuk mendiagnosa penyakit kelebihan gula darah ini memiliki rata-rata 88% maka dinyatakan layak pakai untuk pengguna.

B. Saran

1. Aplikasi sistem pakar untuk mendiagnosa penyakit kelebihan gula darah ini hanya mengolah data konsultasi berbasis *WEB*, maka dari itu perlu adanya sistem seperti android, karena saat ini sudah berbasis *mobile*.
2. Dalam mengatasi masalah kekurangan sistem, maka pengguna dapat memberikan masukan kepada pengembang sistem untuk memperbaiki sistem agar lebih baik lagi.

DAFTAR PUSTAKA

- [1] Pakar, S., Mendiagnosa, U., & Pada, H. (2017). Sistem pakar untuk mendiagnosis hama pada tanaman jambu biji menggunakan metode *bayes*, (Septembaer).
- [2] Sri Wahyuni Nasution, Nelly Astuti Hasibuan, Putri Ramadhani. (2017). Sistem Pakar Diagnosa Anoreksia Nervosa Menerapkan Metode *Case Based Reasoning*. 1(1). 52-56.
- [3] Ginanjar Wiro Sasmito, (2017). Penerapan Metode *Waterfall* pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal. 2(1). 6-12.
- [4] Mukhammad Shaid, Wawan Laksito YS, Yustina Retno Utami. (2015). Sistem Pakar Pertumbuhan Balita Berbasis Web dengan Metode Case Based Reasoning. 37-44.
- [5] Dito Putro Utomo. (2016). Sistem Pakar Mendeteksi Kerusakan Toner dengan Menggunakan Metode *Case Based Reasoning*, 430-434.
- [6] Annahl Riadi. (2017). Penerapan Metode *Certainty Factor* untuk Sistem Pakar Diagnosa Penyakit Diabetes Meliyus pada RSUD Bumi Panua Kabupaten Pohuwato, 309-316.
- [7] Sri Wahyuni Nasution, Nelly Astuti Hasibuan, Putri Ramadhani. (2017). Sistem Pakar Diagnosa Anoreksia Nervosa Menerapkan Metode *Case Based Reasoning*. 1(1). 52-56.
- [8] Faza Akmal, Sri Winiarti (0516127501), (2014). Sistem Pakar untuk Mendiagnosa Penyakit .Lambung dengan Implementasi Metode CBR (*Case Based Reasoning*) Berbasis Web. 2(1). 790-800.
- [9] Dito Putro Utomo. (2016). Sistem Pakar Mendeteksi Kerusakan Toner dengan Menggunakan Metode *Case Based Reasoning*, 430-434.
- [10] Sri Wahyuni Nasution, Nelly Astuti Hasibuan, Putri Ramadhani. (2017). Sistem Pakar Diagnosa Anoreksia Nervosa Menerapkan Metode *Case Based Reasoning*. 1(1). 52-56.
- [11] Ni Luh Gede Pivin Suwirmayanti. (2017). Penerapan Metode *K-Nearest Neighbor* untuk Sistem Rekomendasi Pemilihan Mobil. 17(2). 120-131.

- [12] Salamun. (2017). Penerapan Algoritma *Nearest Neighbor* dan CBR pada *Expert System* Penyimpanan Perilaku Seksual. 2(2). 63-70.
- [13] Dito Putro Utomo. (2016). Sistem Pakar Mendeteksi Kerusakan Toner dengan Menggunakan Metode *Case Based Reasoning*, 430-434.
- [14] Sri Wahyuni Nasution, Nelly Astuti Hasibuan, Putri Ramadhani. (2017). Sistem Pakar Diagnosa Anoreksia Nervosa Menerapkan Metode *Case Based Reasoning*. 1(1). 52-56.
- [15] Muliani, U. (2013). Asupan Zat-Zat Gizi dan Kadar Gula Darah Penderita DM tipe 2. *Jurnal Kesehatan*, 4(2), 325–332.
- [16] Ginanjar Wiro Sasmito, (2017). Penerapan Metode *Waterfall* pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal. 2(1). 6-12.
- [17] Ginanjar Wiro Sasmito, (2017). Penerapan Metode *Waterfall* pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal. 2(1). 6-12.
- [18] Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). Pengujian Aplikasi Menggunakan *Black Box Testing Boundary Value Analysis*. *Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN)*, 1(3), 34. <https://doi.org/ISSN : 2407 - 3911>
- [19] Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). Pengujian Aplikasi Menggunakan *Black Box Testing Boundary Value Analysis*. *Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN)*, 1(3), 34. <https://doi.org/ISSN : 2407 - 3911>.

LAMPIRAN