



**KLASIFIKASI TINGKAT OBESITAS BERDASARKAN INDEKS MASA
TUBUH MENGGUNAKAN ALGORITMA *SUPPORT VECTOR*
MACHINES (SVM)**

TUGAS AKHIR

NAUFALAGUNG RIZKINA

NPM 20670051

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

2024



**KLASIFIKASI TINGKAT OBESITAS BERDASARKAN INDEKS MASSA
TUBUH MENGGUNAKAN ALGORITMA SUPPORT VECTOR
MACHINES (SVM)**

TUGAS AKHIR

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

NAUFALAGUNG RIZKINA

NPM 20670051

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

2024

TUGAS AKHIR
KLASIFIKASI TINGKAT OBESITAS BERDASARKAN INDEKS MASSA
TUBUH MENGGUNAKAN ALGORITMA SUPPORT VECTOR
MACHINES (SVM)

Disusun dan diajukan oleh

NAUFALAGUNG RIZKINA

NPM 20670051

Telah disetujui oleh pembimbing untuk dilanjutkan untuk disusun menjadi

TUGAS AKHIR pada tahun 2024

Semarang, 5 Agustus 2024

Pembimbing Utama,



Aris Tri Joko Harjanto, S.Kom., M.Kom.

NIDN. 0619048202

Pembimbing Pendamping,



Nugroho Dwi S, S.Kom., M.Kom

NPP. 178801537

TUGAS AKHIR

**KLASIFIKASI TINGKAT OBESITAS BERDASARKAN INDEKS MASSA
TUBUH MENGGUNAKAN ALGORITMA SUPPORT VECTOR
MACHINES (SVM)**

Disusun dan diajukan oleh

**NAUFALAGUNG RIZKINA
NPM 20670051**

Telah dipertahankan di depan Dewan Penguji

Pada Tanggal 7 Agustus 2024 dan dinyatakan telah Memenuhi syarat

Dewan Penguji

Ketua


Ibnu Toto Husodo, S.T., M.T.
NPP. 136901387

Sekretaris



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


Penguji 1


Aris Tri Joko Harjanto, S.Kom., M.Kom
NIDN. 0619048202

Penguji 2


Nugroho Dwi Saputro, S.Kom. M.Kom
NPP. 178801537

Penguji 3


Febrian Murti Dewanto, SE., M.Kom.
NPP. 057801172

MOTTO DAN PERSEMBAHAN

Moto :

“Teguh melangkah menuju masa depan yang cerah, dibangun di atas semangat pantang menyerah, kasih sayang keluarga yang selalu mendukung, dan kepercayaan kepada Tuhan, adalah pondasi untuk kesuksesan besar di masa depan.” – Naufal Agung Rizkina

Persembahan:

Tugas akhir ini saya persembahkan kepada:

1. Kepada kedua almarhum orang tua saya
2. Kepada kakak saya yang Enik Supriyanti dan Effendi Dwi Susanto yang telah memberi support sehingga saya dapat menyelesaikan seluruh proses perkuliahan hingga detik ini.
3. Seluruh keluarga besar saya yang saya cintai.
4. Teman-teman saya yang selalu memberi motivasi dan semangat untuk penulisan tugas akhir ini
5. Seluruh dosen Program Studi Informatika Universitas PGRI Semarang yang senantiasa membimbing saya untuk menyelesaikan tugas akhir ini.
6. Almamater saya Universitas PGRI Semarang yang selalu saya banggakan.

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini :

Nama : Naufal Agung Rizkina

NPM : 20670051

Program Studi : Informatika

Fakultas : Teknik dan Informatika

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

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

Semarang, 7 Agustus 2024
Yang membuat pernyataan

Naufal Agung Rizkina
NPM 20670051



ABSTRAK

Obesitas merupakan kondisi gangguan kesehatan yang diakibatkan oleh penumpukan lemak di dalam tubuh, yang sering kali dapat diukur melalui kondisi berat badan yang di atas normal. Konsumsi kalori yang berlebihan dibandingkan dengan aktivitas pembakaran kalori mengakibatkan kalori menumpuk berlebihan menjadi lemak. Postur tubuh yang ideal ditentukan oleh bentuk tubuh masing-masing individu, sehingga perbandingan tinggi badan dan berat badan yang baik akan menghasilkan postur tubuh yang ideal. Penentuan berat badan yang ideal akan menentukan tingkat penampilan seseorang. Untuk mengetahui postur tubuh yang ideal, seseorang harus mengetahui berat badan ideal dan asupan kalori yang dibutuhkan. Keadaan obesitas dalam jangka panjang berdampak pada kenyamanan hidup seseorang dan dapat menimbulkan penyakit kronis seperti diabetes, stroke, dan jantung koroner. Pengukuran obesitas dapat dilakukan dengan *antropometri*, menggunakan perhitungan Indeks Massa Tubuh (IMT) dengan perbandingan rasio berat badan dan tinggi badan. Teknik *Support Vector Machines* (SVM) digunakan untuk menentukan *hyperplane* yang dapat memisahkan data set dari dua kelas yang berbeda. SVM dapat mengatasi masalah klasifikasi dan regresi dengan linier maupun non-linier. Pemilihan algoritma *Support Vector Machines* sebagai salah satu metode *machine learning* yang dinamis menjadi pilihan dalam mengatasi masalah yang berkaitan dengan obesitas. Penelitian ini menggunakan metode pengembangan waterfall yang diawali dengan analisis kebutuhan sampai pengujian. Penelitian ini menghasilkan *accuracy* yang memiliki presentase 92% dari hasil pelatihan model menggunakan *hyperparameter grid search cross-validation* untuk mencapai hasil yang maksimal. Penelitian ini diakhiri dengan 2 jenis pengujian aplikasi, yaitu *black box testing*, dan *user acceptance testing* (UAT). Hasil pengujian *black box* menunjukkan kesuksesan dalam mencapai presentase 100%, sementara UAT mendapatkan presentase 79,8% dengan hasil kategori baik.

Kata Kunci: Obesitas, Postur Tubuh, Indeks Massa Tubuh, *waterfall*, *Support Vector Machines*, *Machine Learning*

PRAKATA

Puji syukur atas kehadiran Allah SWT karena atas limpahan rahmat-Nya, penulis dapat menyusun dan menyelesaikan TUGAS AKHIR ini dengan lancar. TUGAS AKHIR yang berjudul “Klasifikasi Tingkat Obesitas Berdasarkan Indeks Massa Tubuh Menggunakan Algoritma *Support Vector Machines* (SVM)” ini disusun untuk memenuhi syarat memperoleh gelar Sarjana Komputer.

Penyusunan TUGAS AKHIR ini tidak lepas dari berbagai hambatan dan rintangan serta kesulitan-kesulitan. Namun, berkat bimbingan, bantuan, nasehat dan dorongan serta saran-saran dari berbagai pihak, khususnya Pembimbing, segala hambatan dan rintangan serta kesulitan tersebut dapat teratasi dengan baik. Oleh karena itu, dalam kesempatan ini dengan tulus hati penulis menyampaikan terima kasih kepada:

1. Allah SWT yang telah memberikan rahmat serta hidayah-Nya serta selalu melindungi penulis dalam proses penyusunan TUGAS AKHIR, sehingga penulis dapat menyelesaikan TUGAS AKHIR ini.
2. Kakak saya tercinta yang telah banyak memberikan dorongan baik moral, material dan doa. Semoga ini menjadi awal langkah sukses untuk penulis.
3. Ibu Dr Sri Suciati, M.Hum selaku Rektor Universitas PGRI Semarang yang telah memberi kesempatan kepada penulis untuk menimba ilmu di Universitas PGRI Semarang.
4. Bapak Ibnu Toto Husodo, S.T., M.T. selaku Dekan Fakultas Teknik dan Informatika Universitas PGRI Semarang yang telah memberikan izin kepada penulis untuk melakukan penelitian.
5. Bapak Bambang Agus Herlambang, S. Kom., M. Kom. selaku Ketua Program Studi Informatika Fakultas Teknik dan Informatika Universitas PGRI Semarang.
6. Aris Tri Jaka Harjanta, S.Kom, M.Kom selaku Pembimbing I Program Studi Informatika Fakultas Teknik dan Informatika Universitas PGRI

Semarang yang telah membimbing penulis dengan penuh dedikasi yang tinggi.

7. Nugroho Dwi Saputro, S.Kom., M.Kom selaku Pembimbing II Program Studi Informatika Fakultas Teknik dan Informatika Universitas PGRI Semarang yang telah membimbing penulis dengan penuh dedikasi yang tinggi.
8. Seluruh Dosen Pengajar, Staff dan Karyawan Fakultas Teknik dan Informatika Universitas PGRI Semarang
9. keluarga besar yang telah memberikan dukungan doa dan semangat demi terselesaikannya Laporan TUGAS AKHIR ini.
10. Teruntuk teman-teman penulis yang tidak bisa penulis sebutkan satu per satu, yang telah membantu penulis dalam bentuk support mental dan bertukar pikiran dalam penulisan TUGAS AKHIR ini.

Akhir kata, penulis berharap laporan TUGAS AKHIR ini dapat bermanfaat bagi pembaca dan kita semua yang membutuhkan.

Semarang, 7 Agustus 2024

Penulis

DAFTAR ISI

SAMPUL DALAM	i
LEMBAR PENGESAHAN	ii
LEMBAR PERSETUJUAN	iii
MOTTO DAN PERSEMBAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
ABSTRAK	vi
PRAKATA	vii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiii
BAB I	1
PENDAHULUAN	1
A. Latar Belakang	1
B. Identifikasi Masalah	2
C. Rumusan Masalah	2
D. Batasan Masalah	2
E. Tujuan Penelitian	2
F. Manfaat Penelitian	3
BAB II	4
KAJIAN PUSTAKA	4
A. Tinjauan Pustaka	4
B. Landasan Teori	6
C. Kerangka Berfikir	21
BAB III	22
METODE PENELITIAN	22
A. Pendekatan Penelitian	22
B. Fokus Penelitian	22
C. Jenis dan Sumber Data	22
D. Teknik Pengumpulan Data	23

E. Langkah Penelitian	23
BAB IV	26
HASIL DAN PEMBAHASAN	26
A. Hasil.....	26
B. Desain	28
C. Pembahasan.....	59
BAB V.....	62
KESIMPULAN DAN SARAN	62
A. Kesimpulan.....	62
B. Saran	63
DAFTAR PUSTAKA	64

DAFTAR TABEL

Tabel 2. 1 Tinjauan Pustaka	4
Tabel 2. 2 <i>Use Case Diagram</i>	15
Tabel 2. 3 Activity Diagram.....	16
Tabel 2. 4 Sequence Diagram.....	17
Tabel 2. 5 Confusion Matrix	20
Tabel 4. 1 Dataset Obesitas.....	26
Tabel 4. 2 Training Data Folder.....	27
Tabel 4. 3 Testing Data Folder.....	27
Tabel 4. 4 Sampel Dataset Obesitas	34
Tabel 4. 5 Labelling.....	36
Tabel 4. 6 Hyperparameter Grid Search Cross-Validation.....	40
Tabel 4. 7 Sampel Data Setelah SMOTE	42
Tabel 4. 8 hyperparameter 5-Fold Cross-Validation	45
Tabel 4. 9 requirements Pembuatan Streamlit	49
Tabel 4. 10 Pengujian Black Box	52
Tabel 4. 11 Hasil Pengujian Black Box	54
Tabel 4. 12 Pengujian User Acceptance Testing.....	56
Tabel 4. 13 Hasil Pengujian UAT	58

DAFTAR GAMBAR

Gambar 2. 1 Model <i>Support Vector Machine</i>	9
Gambar 4. 1 Use Case Diagram	29
Gambar 4. 2 <i>Activity Diagram</i> Halaman Prediksi	30
Gambar 4. 3 <i>Sequence Diagram</i> Halaman Prediksi	31
Gambar 4. 4 Desain Halaman Prediksi	32
Gambar 4. 5 Model Proses CRISP-DM	33
Gambar 4. 6 <i>Read Dataset</i>	34
Gambar 4. 7 <i>labelling Data</i>	35
Gambar 4. 8 Visualisasi <i>Labelling Data</i>	36
Gambar 4. 9 <i>Label encoding Gender</i>	37
Gambar 4. 10 Standarisasi data	37
Gambar 4. 11 <i>Split Data</i>	38
Gambar 4. 12 <i>Parameter GridSearch</i>	39
Gambar 4. 13 <i>Training Model SVC</i>	40
Gambar 4. 14 Evaluasi Model <i>Best_Estimator</i>	41
Gambar 4. 15 SMOTE	41
Gambar 4. 16 Standarisasi Fitur SMOTE	43
Gambar 4. 17 Training Model SMOTE	43
Gambar 4. 18 Parameter XGboost	44
Gambar 4. 19 Training Model XGboost	44
Gambar 4. 20 <i>Confusion Matrik</i>	46
Gambar 4. 21 Evaluasi Model SMOTE SVM	47
Gambar 4. 22 Evaluasi Model XGBoost	47
Gambar 4. 23 Kode Prediksi Streamlit	50
Gambar 4. 24 Halaman Prediksi	51

DAFTAR LAMPIRAN

Lampiran 1 Pengujian Black Box	67
Lampiran 2 lembar Pengujian UAT	76
Lampiran 3 Lembar Pembimbingan skripsi	86
Lampiran 4 Perhitungan Manual SVM.....	88

BAB I

PENDAHULUAN

A. Latar Belakang

Obesitas merupakan kondisi gangguan kesehatan yang diakibatkan oleh lemak yang menumpuk didalam tubuh, obesitas seringkali dapat diukur melalui kondisi berat badan yang diatas normal. Konsumsi kalori yang berlebihan yang berbanding terbalik dengan aktivitas pembakaran kalori mengakibatkan kalori menumpuk berlebihan menjadi lemak. Postur tubuh yang ideal ditentukan oleh bentuk tubuhnya sendiri, oleh karena itu perbandingan tinggi badan dan berat badan yang baik akan menghasilkan postur tubuh yang ideal. Tinggi badan seseorang pada umur tertentu tidak akan berubah tetapi berat badan masih besar kemungkinan berubah. Oleh karena itu, berat badan yang ideal akan menentukan tingkat penampilan seseorang. Untuk mengetahui postur tubuh yang ideal, seseorang harus mengetahui berapa berat badan ideal yang seharusnya mereka miliki, agar mereka juga bisa menentukan berapa asupan kalori yang mereka butuhkan untuk menghasilkan berat badan yang ideal dan tentunya baik untuk kesehatan.[1]

Keadaan obesitas dalam jangka panjang akan berdampak pada kenyamanan hidup seseorang. Dalam melakukan aktifitas, keadaan tubuh obesitas akan menimbulkan rasa tidak nyaman dalam tubuh seperti sendi terasa nyeri atau badan terasa berat. Selain hal tersebut obesitas juga dapat menimbulkan beberapa penyakit kronis, diantaranya diabetes, stroke, dan jantung koroner. Obesitas dapat ditentukan melalui pengukuran *antropometri*, untuk sekarang ini pengukuran *antropometri* menggunakan perhitungan Indeks Massa Tubuh (IMT), dengan perbandingan rasio berat badan dibagi tinggi badan². [2]

Support Vector machines (SVM) merupakan suatu Teknik untuk menentukan *hyperplane* yang bisa memisahkan dua set data dari dua kelas yang berbeda. SVM dapat mengatasi masalah klasifikasi dan regresi dengan linier

maupun non-linier. Pemilihan algoritma *Support Vector Machines* dikarenakan algoritma tersebut merupakan algoritma *machine learning* yang cukup dinamis dalam implementasi klasifikasi *Support Vector Machines*. Kualitas algoritma yang dipilih diharapkan menghasilkan klasifikasi yang akurat dan dapat diandalkan.

B. Identifikasi Masalah

Pada penelitian ini, penulis mengidentifikasi bahwa masalah obesitas sangat menentukan tingkat kepercayaan diri seseorang, kurangnya kesadaran seseorang terhadap berat badan dapat menyebabkan masalah Kesehatan obesitas tersebut.

C. Rumusan Masalah

Berdasarkan latar belakang penelitian yang berjudul “Klasifikasi Tingkat Obesitas Berdasarkan Indeks Massa Tubuh Menggunakan Algoritma *Support Vector Machines* (SVM)” tersebut dapat dirumuskan masalah yaitu bagaimana membangun sebuah sistem *machine learning* yang dapat digunakan untuk mengukur tingkat obesitas seseorang.

D. Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

1. Aplikasi ini dibuat untuk menghasilkan sebuah sistem yang dapat mengukur tingkat obesitas bagi pengguna.
2. Algoritma *Support Vector Machines* (SVM) digunakan untuk proses klasifikasi dalam aplikasi ini.
3. Dalam pengembangan aplikasi penggunaan metode pengembangan *waterfall* hanya sampai tahap pengujian tanpa melakukan pemeliharaan.

E. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk membangun sebuah sistem yang dapat mengklasifikasi Tingkat obesitas seseorang berdasarkan indeks massa

tubuh, agar seseorang dapat menyadari bahwa obesitas sangat beresiko untuk kesehatan.

F. Manfaat Penelitian

Hasil penelitian ini diharapkan dapat memberikan manfaat bagi semua pihak yang terkait dalam proses yang dilaksanakan, sebagai berikut :

1. Bagi Penulis

Untuk melatih dan mengasah serta dapat mengukur seberapa kemampuan daya tangkap penulis dalam mempraktikkan ilmu yang diperoleh dalam bangku kuliah untuk menerapkannya dalam lingkungan kehidupan Masyarakat.

2. Bagi Akademik

Sebagai referensi dan tolak ukur untuk keberhasilan proses belajar mengajar yang dapat digunakan sebagai bahan evaluasi bagi pihak akademik.

3. Bagi Masyarakat

Sebagai tambahan ilmu pengetahuan dan bahan acuan melakukan penelitian lebih lanjut serta tambahan pengetahuan dalam mempelajari masalah yang ada.

BAB II KAJIAN PUSTAKA

A. Tinjauan Pustaka

Dalam penelitian yang dilakukan, penulis menggali beberapa informasi dari penelitian yang sebelumnya telah dilakukan sebagai bahan pertimbangan mengenai kelebihan dan kekurangan tersebut. Penulis juga menggali informasi tentang teori yang berkaitan dengan judul yang diperoleh guna sebagai landasan teori ilmiah. Pada penelitian sebelumnya oleh Sibi S, Widiarti A (2022) yang berjudul “Klasifikasi Tingkat Obesitas Mempergunakan Algoritma KNN”. Pada penelitian ini, Sibi S, Widiarti A, melakukan klasifikasi yang berkaitan tentang obesitas. Persamaan judul dengan penulis adalah sama-sama membahas tentang klasifikasi tingkat obesitas.[3]

Penelitian selanjutnya oleh Sitanggung D (2022) yang berjudul “Model Prediksi Obesitas Menggunakan *Support Vector Machine*”. Pada penelitian tersebut, Sitanggung D merancang sebuah sistem prediksi obesitas menggunakan model machine learning yaitu SVM. Persamaan dalam penelitian ini adalah sama-sama menggunakan Algoritma SVM.[4]

Juan Veron Wie dkk, melakukan penelitian tentang penerapan metode naïve bayes dalam mengklasifikasi tingkat obesitas pada pria. Peneliti ini menggunakan algoritma naïve bayes. Tujuan penelitian ini yaitu untuk mengklasifikasi tingkat obesitas pada pria. Penelitian ini menggunakan 60% data training dan 40% data testing dan menghasilkan akurasi sebesar 84.15%.

Tabel 2. 1 Tinjauan Pustaka

No	Peneliti	Judul	Metode	Hasil
1	Sibi S, Widiarti A	Klasifikasi Tingkat Obesitas Mempergunakan Algoritma KNN	KNN	Pada penelitian ini, akurasi data latih tertinggi yaitu 79.96%

No	Peneliti	Judul	Metode	Hasil
2	Sitanggang D	Model Prediksi Obesitas Menggunakan Support Vector Machine	Support Vector Machine (SVM)	Dalam penelitian ini, Implementasi algoritma SVM dapat dengan baik melakukan regresi dengan tingkat akurasi 71.80%.
3	Wie J, Siddik M	Penerapan Metode Naïve Bayes Dalam Mengklasifikasi Tingkat Obesitas Pada Pria	Naïve Bayes	Penerapan pada algoritma Naïve Bayes dalam memprediksi tingkat obesitas pada pria menghasilkan tingkat akurasi yang memuaskan sebesar 84.15% pada rasio split data 60:40.
4	Oktaviana Bangun, Herman Mawengkang, Syahril Efendi	Metode Algoritma Support Vector Machine (SVM) Linier Dalam Memprediksi Kelulusan Mahasiswa	Support Vector Machine (SVM)	Dalam penelitian ini, menggunakan 792 data training yang dibagi menjadi 70% data training dan 30% data testing untuk menghitung tingkat kelulusan mahasiswa dan menghasilkan 90% lulus dan 10% tidak lulus

B. Landasan Teori

1. Sistem Informasi

Definisi sistem informasi adalah kumpulan informasi didalam sebuah basis data menggunakan model dan media teknologi informasi yang digunakan dalam pengambilan keputusan. Informasi merupakan data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambil keputusan.[5]

2. Obesitas

Masalah kesehatan berupa gangguan status gizi merupakan masalah umum yang sering muncul dalam Masyarakat. Obesitas merupakan salah satu permasalahan gizi yang sedang dihadapi masyarakat global, baik di maju maupun berkembang. Obesitas merupakan kelebihan akumulasi lemak yang dapat menyebabkan resiko bagi kesehatan.[6] Obesitas dapat terjadi karena tingginya asupan makanan dengan kandungan lemak dan gula tinggi. Apabila hasil masukan tersebut tidak dibakar maka akan terjadi penimbunan yang dapat menimbulkan efek obesitas bagi tubuh. Selain itu, gaya hidup dan rendahnya aktifitas fisik terutama di daerah urban karena sosial ekonomi menyebabkan peningkatan prelevansi obesitas. Penentuan status gizi dapat dilihat dari hasil perhitungan IMT (Indeks Massa Tubuh) ataupun lingkaran perut.[6]

3. Klasifikasi

Klasifikasi adalah proses untuk menemukan model atau fungsi yang dapat menggambarkan dan membedakan kelas data atau konsep, dengan tujuan agar model tersebut dapat digunakan untuk memprediksi kelas yang belum diketahui dari suatu objek pengamatan. Metode klasifikasi yang umum digunakan pada disiplin ilmu statistika adalah Analisis Diskriminan dan Regresi Logistik. Namun, semakin populernya era data yang menunjukkan bahwa terjadinya pertumbuhan pesat dari

volume data yang luar biasa banyak sehingga menghasilkan set data besar (big data), maka sangat dibutuhkan alat analisis yang kuat dan serbaguna untuk mengungkap informasi berharga dari set data besar dan untuk mengubah set data besar tersebut menjadi pengetahuan yang terorganisir.[7]

Disisi lain, pesatnya perkembangan teknologi kecerdasan buatan maka berkembanglah metode *machine learning*, yaitu mesin yang dikembangkan untuk bisa belajar dengan sendirinya tanpa arahan dari penggunanya. Dimana dalam pengembangannya berdasarkan disiplin ilmu lain seperti statistika, matematika dan data mining. Metode klasifikasi pada *machine learning* yang sering digunakan pada metode *machine learning* adalah *Classification and Regression Trees (CART)*, *Random Forest*, *Naïve Bayes*, *Support Vector Machines (SVM)*, dan lain-lain.[8]

4. *Machine Learning (ML)*

ML adalah bidang keilmuan yang mempelajari bagaimana membuat program yang dapat menghasilkan pengetahuan baru dari pengetahuan yang sudah ada (*experience*) diluar pengetahuan yang “diprogram” secara langsung oleh program. Istilah umumnya adalah bagaimana membuat komputer yang dapat belajar dari lingkungan sekitar sehingga pengetahuan yang berkembang.[9]

ML ini merupakan bidang keilmuan *computer science* yang paling sering salah dimengerti. Selain konotasi negatif dalam Indonesia dengan istilah “ML”, namun ML tersebut juga merupakan bidang dengan ranah aplikatif yang sangat luas dalam banyak sekali bidang keilmuan. Hampir seluruh bidang yang berhubungan dengan “komputasi cerdas” membutuhkan adanya pengetahuan dalam sebuah program, dan dalam tahap ini ML sangat berperan. Jika AI fokus pada membuat komputer yang cerdas, maka untuk mencapai kecerdasan tersebut digunakan ML.

5. *Support Vector Machine* (SVM)

Support Vector Machine adalah salah satu algoritma dalam *machine learning* yang digunakan untuk klasifikasi dan regresi. SVM bertujuan untuk menemukan *hyperplane* terbaik yang memisahkan dua kelas dalam ruang fitur sedemikian rupa sehingga margin antara *hyperplane* dan titik-titik terdekat dari masing-masing kelas, yang disebut *support vectors* adalah maksimum.

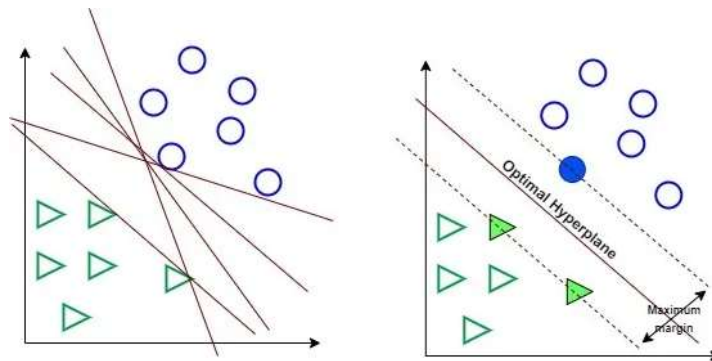
a. *Cross-Industry Standard Process for Data Mining* (CRISP-DM)

CRISP-DM adalah model proses yang banyak digunakan untuk proyek data mining. CRISP-DM merupakan pendekatan yang terstruktur dan sistematis untuk perencanaan dan pelaksanaan proyek data mining. Metode ini terdiri dari enam fase utama:

1. *Business Understanding*, kegiatan untuk mendapatkan pemahaman yang jelas tentang masalah yang ingin kita selesaikan.
2. *Data Understanding*, adalah fase kedua dalam CRISP-DM yang berfokus pada pengumpulan data awal dan eksplorasi data untuk memahami karakteristik dan kualitas data. Tujuan fase ini adalah untuk mengidentifikasi masalah yang ada dalam data dan membentuk hipotesis awal tentang pola yang ada dalam data.
3. *Data Preparation*, merupakan serangkaian kegiatan untuk mengubah data mentah menjadi data yang siap digunakan untuk analisis atau pemodelan. Fase ini bertujuan untuk memastikan bahwa data yang digunakan dalam proses data mining memiliki kualitas tinggi dan relevan untuk analisis.
4. *Modelling*, adalah fase keempat dalam metodologi CRISP-DM yang berfokus pada Pembangunan dan penerapan model analisis menggunakan teknik-teknik data mining. Pada tahap ini, model dibangun untuk menemukan pola dan hubungan dalam data yang dapat digunakan untuk membuat prediksi.

5. *Evaluation*, bertujuan untuk menilai kualitas dan keadilan model yang telah dibangun. Evaluasi model melibatkan penggunaan matrik evaluasi yang relevan untuk menilai kinerja model dan menentukan apakah model tersebut layak untuk diterapkan.
6. *Deployment*, adalah fase terakhir dalam CRISP-DM yang bertujuan untuk mengimplementasikan model yang telah dibangun dan dievaluasi. Pada tahap ini, model diterapkan untuk digunakan oleh end-user atau sistem yang memerlukan output model untuk pengambilan keputusan.

Dengan mengikuti tahapan CRISP-DM, proses pembangunan dan penerapan model SVM menjadi lebih terstruktur dan sistematis, membantu memastikan hasil yang lebih akurat dan relevan.



Gambar 2. 1 Model *Support Vector Machine*

Konsep dasar SVM adalah mencari hyperplane dengan margin terbesar. *Hyperplane* ini adalah suatu d -dimensi (d adalah fitur dalam data) yang membagi ruang fitur menjadi dua bagian, mewakili dua kelas berbeda. Margin adalah jarak terdekat antara *hyperplane* dan *support vectors*. Usaha dalam mencari lokasi hyperplane ini merupakan proses dari pembelajaran pada SVM. Data dinotasikan sebagai $x_i \in R$ sedangkan label masing-masing dinotasikan $y_i \in \{1,0\}$ untuk $i = 1, 2, \dots, l$ yang mana l

adalah banyak data. Asumsi 1 dan 0 dapat terpisah secara sempurna oleh hyperplane berdimensi d yang didefinisikan pada persamaan (1).

$$\vec{w} \cdot x + b = 0 \quad (1)$$

Keterangan :

w : Vector bobot.

x ; Vektor Fitur.

b : Bias atau intercept

Pola \vec{x}_1 yang termasuk kelas 1 dapat dirumuskan sebagai pola yang memenuhi pertidaksamaan (2).

$$\vec{w} \cdot \vec{x}_1 + b \geq 1 \quad (2)$$

Keterangan :

w : Vektor bobot.

x_i : Vektor fitur dari contoh data.

b : Bias atau intercept

Sedangkan pola \vec{x}_1 yang termasuk kelas 0 dirumuskan dengan pertidaksamaan (3).

$$\vec{w} \cdot \vec{x}_1 + b \leq -1 \quad (3)$$

Keterangan :

w : Vektor bobot.

x_i : Vektor fitur dari contoh data.

b : Bias, nilai konstanta yang menyesuaikan hasil prediksi.

Margin terbesar ditemukan dengan memaksimalkan nilai jarak *hyperplane* dan titik terdekatnya dengan persamaan (4).

$$\frac{1}{\|\vec{w}\|} \quad (4)$$

Keterangan :

$\|\vec{w}\|$: panjang dari vektor bobot \vec{w} .

Hal ini dapat dirumuskan sebagai *quadratic programming problem* yaitu mencari titik minimal persamaan (5) dengan memperhatikan *constraint* persamaan (6).

$$\min_w t(w) = \frac{1}{2} \|\vec{w}\|^2 \quad (5)$$

$$y_i(\vec{w} \cdot \vec{x}_i) - 1 \geq 0, \forall i \quad (6)$$

Keterangan :

\min_w : Nilai minimum dari fungsi objektif terhadap vector bobot \vec{w} .

$t(w)$: Fungsi objektif yang ingin diminimalkan.

$\|\vec{w}\|$: panjang dari vektor bobot \vec{w} .

y_i : Label kelas untuk titik data i .

Problem dapat dipecahkan dengan Teknik komputasi *langrange Multiplier*, seperti persamaan berikut.

$$L(\vec{w}, b, \alpha) = \frac{1}{2} \|\vec{w}\|^2 - \sum_{i=1}^l \alpha_i (y_i (\vec{w} \cdot \vec{x}_i + b) - 1), i = 1, 2, \dots, l \quad (7)$$

Keterangan :

w : Vector bobot

b : Bias atau intercept dari hyperplane.

α_i : pengali langrange untuk setiap titik data i .

y_i : Label kelas untuk titik data i .

x_i : Vektor fitur dari contoh data i .

l : Jumlah total titik data.

α_i adalah *Langrange Multipliers*, yang bernilai 0 atau positif $\alpha_i \geq 0$. Nilai optimal dari persamaan dapat dihitung dengan meminimalakan L terhadap \vec{w} dan b dan memaksimalkan L terhadap α_i . Berdasarkan sifat bahwa pada titik optimal $L = 0$, persamaan dapat dimodifikasi sebagai maksimalisasi problem yang mengandung α_i .

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j (\vec{x}_i \cdot \vec{x}_j)$$

Keterangan :

α_i : pengali langrange untuk setiap titik data i .

y_i : Label kelas untuk titik data i .

x_i : Vektor fitur dari contoh data i .

l : Jumlah total titik data.

$\vec{x}_i \cdot \vec{x}_j$: Hasil perkalian titik (dot product) antara vektor fitur \vec{x}_i dan \vec{x}_j

Dari hasil perhitungan diperoleh α_i yang bernilai positif. Data yang berkorelasi dengan α_i yang positif inilah yang disebut *support vector*. [10]

$$\alpha_i \geq 0 (i = 1, 2, \dots, l) \quad \sum_{i=1}^l \alpha_i y_i = 0$$

Keterangan :

α_i : pengali langrange untuk setiap titik data i .

l : Jumlah total titik data.

y_i : Label kelas untuk titik data i .

6. Streamlit

Streamlit merupakan *framework* yang membantu programmer membuat tampilan UI web untuk pemrosesan data dan pembelajaran (*machine learning*) dalam bentuk web [11].

7. Google Colaboratory

Google Colab adalah *compiler* dan *editor* Bahasa pemrograman *Python* secara online, google colab adalah *tool* yang memungkinkan siapa saja untuk belajar coding tanpa membutuhkan sarana komputasi yang canggih dan mahal. Seperti halnya google *docs* dan *sheets*, google colab juga berbasis layanan *cloud* sehingga bisa dipakai dimana saja dan kapan saja [12].

8. Website

Pengertian *website* secara umum adalah Kumpulan halaman web yang saling terkait dan dapat diakses melalui internet. Halaman web tersebut berisi informasi, seperti teks, gambar, video dan animas. Website dapat digunakan untuk berbagai keperluan, seperti bisnis, Pendidikan, hiburan, dan lainnya.

Situs web dapat dirancang dalam berbagai cara, dari situs satu halaman sederhana hingga situs multi halaman yang kompleks dengan

fungsionalitas yang luas. Ini bisa dikembangkan secara statis, artinya konten tetap dan tidak sering berubah, atau dinamis, artinya konten dihasilkan secara dinamis dan dapat sering berubah. Situs web juga bisa *responsive*, artinya menyesuaikan dengan ukuran dan resolusi perangkat yang digunakan untuk mengaksesnya, seperti smartphone dan komputer. Singkatnya, *website* adalah platform digital yang memungkinkan individu dan organisasi berbagi informasi dan terhubung dengan orang lain dalam skala global.

9. Metode *Waterfall*

Metode *Waterfall* merupakan metode pengembangan perangkat lunak model *Software Development Life Cycle* (SDLC) yang bersifat linier dari proses perencanaan, analisa, desain, dan implementasi pada sistem.[13] Metode ini dilakukan pendekatan secara sistematis, mulai dari tahap kebutuhan sistem hingga tahap akhir yaitu pemeliharaan. Langkah yang dilalui harus diselesaikan satu per satu dan berjalan berurutan, oleh karena itu dinamakan “*Waterfall*”, berikut adalah tahapan-tahapan metode *waterfall* :

a) Analisis Kebutuhan (*Requirements*)

Tahap analisis kebutuhan adalah tahap awal dalam pengembangan sistem yang bertujuan untuk memahami dan menetapkan kebutuhan pengguna sistem. Pada tahap ini, dilakukan pengumpulan informasi tentang masalah yang ingin dipecahkan oleh sistem dan kebutuhan fungsional dan non-fungsional dari sistem yang akan dikembangkan serta gambaran dari sistem yang berjalan saat ini.

b) Desain

Pada tahap ini, dilakukan desain sistem berdasarkan kebutuhan yang telah diidentifikasi pada tahap sebelumnya. Tahapan desain sangat penting dalam pengembangan sistem karena hasil dari

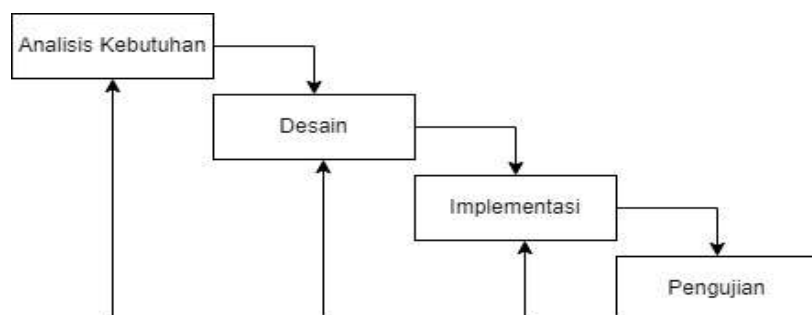
tahapan ini akan menjadi dasar tahapan implementasi dan pengujian sistem.

c) Implementasi

Adapun tahap ini dilakukan proses implementasi hasil dari perancangan sistem pada tahapan sebelumnya ke dalam bentuk sebuah sistem yang siap digunakan. Pada tahap implementasi melibatkan coding menggunakan bahasa pemrograman Python dan penggunaan framework streamlit untuk mengimplementasikan antarmuka pengguna sistem.

d) Pengujian

Setelah sistem selesai dibuat, tahap selanjutnya adalah melakukan pengujian. Pengujian ini bertujuan untuk memastikan bahwa sistem yang telah dibangun berjalan dengan baik dan memenuhi spesifikasi yang telah ditentukan. Tahap pengujian terdiri dari beberapa jenis pengujian, antara lain pengujian blackbox, whitebox, dan UAT.



Gambar 2.2 Tahapan Metode *Waterfall*

10. *Unified Modeling Language (UML)*






UML adalah sebuah “bahasa” yang telah menjadi standar dalam industri merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang sebuah model sistem.


Pada perancangan desain sistem ini menggunakan *Unified Modeling Language* (UML) sebagai permodelannya. *Use-case diagram* UML yang merupakan notasi lengkap untuk membuat visualisasi model untuk suatu sistem.[14]

a) *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor.

Tabel 2. 2 *Use Case Diagram*


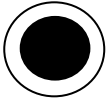



Simbol	Keterangan
	Use Case menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dan aktor.
	Aktor adalah abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem.
	Asosiasi antar aktor dan use case
	Asosiasi antara aktor dan use case yang menggunakan panah terbuka untuk mengidentifikasi bila aktor berinteraksi secara pasif dengan sistem.
	Include, menunjukkan bahwa suatu use case merupakan fungsionalitas dari use case lainnya

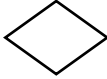

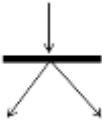
Simbol	Keterangan
	Extend, menunjukkan bahwa use case merupakan tambahan fungsionalitas dari use case lainya jika suatu kondisi terpenuhi

b) *Activity Diagram*

Activity diagram menggambarkan alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggabarkan proses parallel yang mungkin terjadi pada beberapa eksekusi.

Tabel 2. 3 *Activity Diagram*

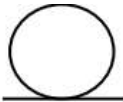
Simbol	Keterangan
	Strat Poin : diletakan pada pojok kiri atas dan merupakan awal aktifitas.
	End Point : menunjukan akhir dari suatu diagram aktivitas.
	Kondisi Transisi : menunjukan transisi antar aktivitas.
	Swimlane : menunjukan actor dari diagram aktivitas yang dibuat.
	Aktivitas : menunjukan aktivitas-aktivitas yang terdapat pada diagram aktivitas.



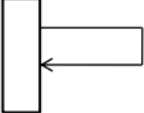


Simbol	Keterangan
	Decision Point : menggambarkan pilihan untuk pengambilan keputusan true, false.
	Join : (penggabungan atau rake) digunakan untuk menunjukkan adanya dekomposisi.
	Fork : (percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan menjadi satu.

c) *Sequence Diagram*

Sequence diagram adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu, *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Objek-objek yang dihubungkan dengan berjalannya proses operasi biasanya diurutkan dari kiri ke kanan.

Tabel 2. 4 *Sequence Diagram*

Simbol	Keterangan
	Entitas Class : merupakan bagian dari sistem yang berisi kelas berupa entitasentitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.

Simbol	Keterangan
	Boundary Class : berisi Kumpulan kelas yang menjadi interface atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan form cetak.
	Control Class : suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah bagian kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	Recursive : menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	Activation : mewakili sebuah eksekusi operasi dari objek, Panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline : garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.

11. Pengujian Blackbox

Pengujian *blackbox* adalah salah satu metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas, khususnya pada input dan output aplikasi (apakah sudah sesuai dengan apa yang diharapkan atau belum). Tahap pengujian merupakan salah satu tahap yang harus ada dalam sebuah siklus pengembangan perangkat lunak [15].

Pengujian *blackbox* berusaha menemukan kesalahan dalam kategori sebagai berikut:

- a. Fungsi yang bermasalah atau hilang
- b. Kesalahan struktur data
- c. Kesalahan interface
- d. Kesalahan kinerja
- e. Inisialisasi dan kesalahan terminasi

Dengan mengumpulkan data pengujian yang menggunakan pendekatan secara wawancara dan observasi langsung terhadap fungsi sistem untuk mendapatkan hasil presentase rata-rata dari jumlah *responden* pengujian yang dilakukan dengan perhitungan sebagai berikut:

$$\text{Jumlah presentase rata – rata tercapai} = \frac{(\text{presentase hasil tercapai})}{(\text{Jumlah Responden})}$$

12. *Software Testing*

Software testing merupakan salah satu bagian penting dan sering digunakan dalam pengembangan dari suatu *software*. Dengan adanya *software testing*, *software engineer* mampu mengenali *error* atau kecacatan yang ada dari suatu *software*. Fitur-fitu utama pada *software testing* adalah verifikasi dan validasi. *Software testing* digunakan untuk menjawab pertanyaan sebagai : Apakah *software* dapat berjalan atau berperilaku sebagaimana yang sudah ditentukan.[16]

13. *User Acceptance Testing (UAT)*

User Acceptance Testing merupakan tahapan pengujian yang dilakukan oleh end user, Dimana disini yang menjadi *end user* adalah pengguna yang menggunakan sistem tersebut. *User Acceptance Testing* bisa disebut dengan pengujian akhir dari pengembangan sebuah produk untuk memvalidasi bahwa sistem yang dibangun sudah sesuai dengan kebutuhan pengguna.

Tahapan *user acceptance testing* merupakan tahapan identifikasi bug pada perangkat lunak, sistem dan jaringan yang mungkin dapat menimbulkan masalah bagi pengguna. Pengguna diberikan kesempatan untuk berinteraksi dengan perangkat lunak sebelum resmi dirilis.[17]

14. *Confusion Matrix*

Confusion matrix merupakan metode yang digunakan untuk mengukur kinerja model klasifikasi. *Confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem untuk diukur keakuratannya. *Confusion matrix* juga merupakan salah satu cara dalam melakukan visualisasi terhadap hasil pembelajaran sistem, visualisasi yang ditampilkan memuat dua kategori atau lebih. Table dibawah ini merupakan contoh hasil *confusion matrix* prediksi dua kelas.[18]

Tabel 2. 5 *Confusion Matrix*

		Kelas sebenarnya	
		1	2
Kelas Prediksi	1	True Positive	False Negative
	2	False Positive	True Negative

Perhitungan Akurasi yang dilakukan confusion matrix berdasarkan tabel diatas dapat menggunakan persamaan berikut:

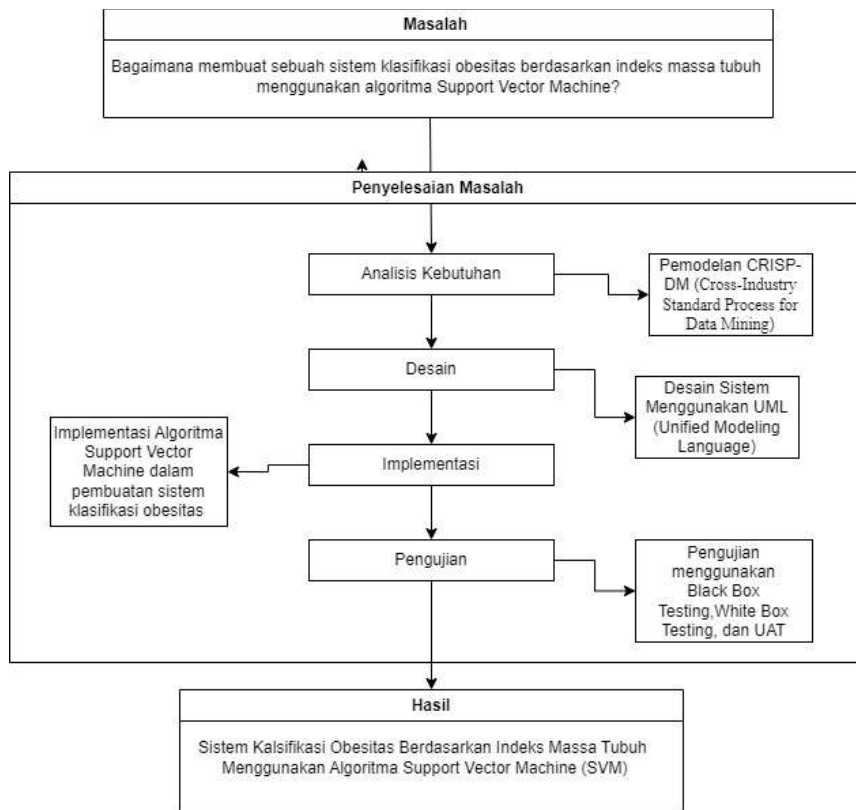
$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN} \times 100\%$$

Keterangan:

- a. *True Positive* (TP): jumlah data dengan kelas positif yang diklasifikasikan positif.

- b. *True Negative* (TN): jumlah data dengan kelas negatif yang diklasifikasikan negatif.
- c. *False Positive* (FP): jumlah data dengan kelas positif yang diklasifikasikan negatif.
- d. *False Negative* (FN): jumlah data dengan kelas negatif diklasifikasikan positif.

C. Kerangka Berfikir



Gambar 2.3 Kerangka Berfikir

BAB III

METODE PENELITIAN

A. Pendekatan Penelitian

Dalam penulisan skripsi ini dilakukan pendekatan penelitian *research and development*. *Research and Development* merupakan penelitian yang menghasilkan produk baru atau mengembangkan produk yang sudah ada agar lebih menarik sesuai dengan pokok pembahasan tertentu. Model pendekatan sistem yang digunakan adalah model perancangan UML. Sedangkan pada tahap metode pengembangan sistem menggunakan metode *waterfall* yang dilakukakan secara berurutan yaitu meliputi analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan.

B. Fokus Penelitian

Fokus penelitian merupakan garis besar dari pengamatan penelitian, sehingga observasi dan analisis hasil penelitian lebih terarah. Oleh sebab itu, digunakan indicator-indikator tertentu agar tidak terjadi pembahasan yang terlalu luas sehingga tidak sesuai dengan napa yang menjadi judul penelitian.

Fokus penelitian ini adalah pembuatan sebuah sistem klasifikasi obesitas berdasarkan indeks massa tubuh menggunakan algoritma *support vector machine*.

C. Jenis dan Sumber Data

1. Data Primer

Data primer merujuk pada dataset yang terdapat pada Kaggle.com yang mencakup informasi tentang BMI dan factor-faktor yang terkait dengan obesitas.

2. Data Sekunder

Data sekunder adalah data yang diperoleh secara tidak langsung dari buku, jurnal ilmiah, laporan riset dan dokumen-dokumen yang mengandung informasi tentang obesitas dan indeks massa tubuh.

D. Teknik Pengumpulan Data

1. Dataset Publik

Pada teknik ini peneliti mengumpulkan data yang berformat .csv yang ada diinternet. Data yang digunakan dalam penelitian ini berupa dataset *500 Person Gender-Height-Weight-Body Mass Index dataset* yang berisi 500 data yang diperoleh dari *Kaggle.com*.

2. Studi Pustaka

Merujuk pada sumber-sumber yang tertulis seperti literatur/jurnal yang memiliki relevansi dengan topik skripsi ini, pengumpulan data tambahan dilakukan untuk mendukung implementasi algoritma *support vector machine* dalam pembuatan sistem kalsifikasi obesitas berdasarkan indeks massa tubuh.

E. Langkah Penelitian

Langkah penelitian merupakan serangkaian tahapan yang melibatkan peneliti dengan logika desain interpretasi hasil. Langkah-langkah dalam penelitian adalah sebagai berikut:

1. Identifikasi masalah

Kunci utama dalam identifikasi masalah pembuatan tugas akhir, tugas akhir ini dibuat guna mengatasi pemasalahan yang ada. Identifikasi masalah terkait penyebaran kasus obesitas dilingkungan masyarakat.

2. Penyelesaian Masalah

a) Model Pengembangan Sistem *Waterfall*

Penyelesaian adalah tahapan selanjutnya dimana pengembangan sistem dengan Metode *Waterfall* menjadi Langkah berikutnya yang terdiri:

1) Analisis Kebutuhan

Pada tahap analisis kebutuhan dilakukan analisis yang berkaitan dengan perancangan aplikasi web tersebut. Mencari informasi terkait dilakukan dengan cara mencari informasi di media internet.

2) Perancangan

Dalam tahap ini dilakukan perancangan berupa arsitektur, desain, spesifikasi software. Perancangan juga melibatkan pembuatan diagram alur dan desain antarmuka pengguna.

3) Implementasi

Pada tahapan selanjutnya yaitu implementasi, dilakukan pembuatan kode program dan dilakukan penerapan pada aplikasi web.

4) Pengujian

Dalam tahapan ini dilakukan pengujian aplikasi web agar sesuai kebutuhan menggunakan black-box testing dan User acceptance Testing (UAT).

b) Model Pengembangan *Support Vector Machine* (SVM)

Dalam pengembangan sistem SVM digunakan metode *Cross-Industry Standard Process for Data Mining* (CRISP-DM). dalam pengembangan CRISP-DM terdapat enam fase yaitu :

1) Business Understanding

kegiatan untuk mendapatkan pemahaman yang jelas tentang masalah yang ingin kita selesaikan.

2) Data Understanding

Mengumpulkan data awal dan memahami data yang tersedia, serta mengidentifikasi masalah kualitas data.

3) Data Preparation

Mempersiapkan data akhir yang akan digunakan dalam pemodelan, melalui proses seleksi dan transformasi data.

4) Modelling

Membangun dan penerapan model analisis menggunakan teknik-teknik data mining. melatih model dengan data yang telah dipersiapkan, dan mengevaluasi kinerja model menggunakan metrik seperti akurasi

5) Evaluation

Mengevaluasi model dengan lebih mendalam untuk memastikan kinerjanya sesuai dengan yang diharapkan.

6) Deployment

Merencanakan bagaimana model akan digunakan, dan mengimplementasikan model dalam sistem produksi.

3. Hasil

Setelah dilakukan penyelesaian masalah tahapan selanjutnya yaitu hasil pembuatan aplikasi sistem klasifikasi tingkat obesitas berdasarkan indeks massa tubuh dengan menerapkan algoritma *support vector machine*.

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini akan membahas mengenai hasil, pembahasa dan penelitian yang telah dilakukan oleh penulis. Hal ini meliputi perancangan dan pengembangan aplikasi klasifikasi obesitas berdasarkan indek massa tubuh.

A. Hasil

Metode yang digunakan penulis dalam mengembangkan aplikasi ini yaitu metode *Waterfall* berikut merupakan tahapan pembuatan aplikasi berdasarkan metode *Waterfall* :

1. Analisis Kebutuhan

Pada tahapan ini mencakup kebutuhan yang diperlukan dalam pembuatan aplikasi. :

a. Analisis Kebutuhan Data

Data yang dibutuhkan untuk penelitian adalah dataset public yang diperoleh dari internet di *website* Kaggle.com. Dataset ini dipilih karena memiliki berbagai variabel yang relevan untuk analisis klasifikasi obesitas, seperti jenis kelamin, berat badan, tinggi badan. Data diproses untuk dibagi menjadi 2 folder yaitu training data folder dan testing data folder. Pembagian data untuk training data sebanyak 80% dari semua data dan untuk testing data sebanyak 30%. Alasan menggunakan pembagian presentase 80%:30% karena dataset ini bersifat tidak seimbang ada lebih sedikit sampel dari satu kelas dibandingkan dengan yang lain. Adapun dataset sebelum dibagi menjadi 2 jenis data dapat dilihat pada tabel 4.1.

Tabel 4. 1 Dataset Obesitas

No	Nama Label	Jumlah
1	Sangat Kurus	13
2	Kurus	22

3	Normal	69
4	Kelebihan Berat	68
5	Obesitas	130
6	Obesitas Ekstrim	198

1) Training Data

Training data digunakan untuk melatih model. Untuk pembagian data jumlah data di training data folder pada penelitian ini dapat dilihat pada tabel 4.2,

Tabel 4. 2 *Training Data Folder*

No	Kelas	Jumlah Data
1	Sangat Kurus	12
2	Kurus	18
3	Normal	51
4	Kelebihan Berat	60
5	Obesitas	100
6	Obesitas Ekstrim	159

2) Testing Data

Testing data digunakan untuk menguji kinerja model setelah prose pelatihan. Untuk pembagian jumlah data testing pad apenelitian ini dapat dilihat pada tabel 4.3.

Tabel 4. 3 *Testing Data Folder*

No	Kelas	Jumlah Data
1	Sangat Kurus	1
2	Kurus	4
3	Normal	18
4	Kelebihan Berat	8
5	Obesitas	30

6	Obesitas Ekstrim	39
---	------------------	----

b. Analisis Kebutuhan Sistem

Dalam membuat dan menjalankan sistem klasifikasi obesitas diperlukan hardware dan software sebagai penunjang. Dalam penelitian ini alat bantu yang digunakan berupa :

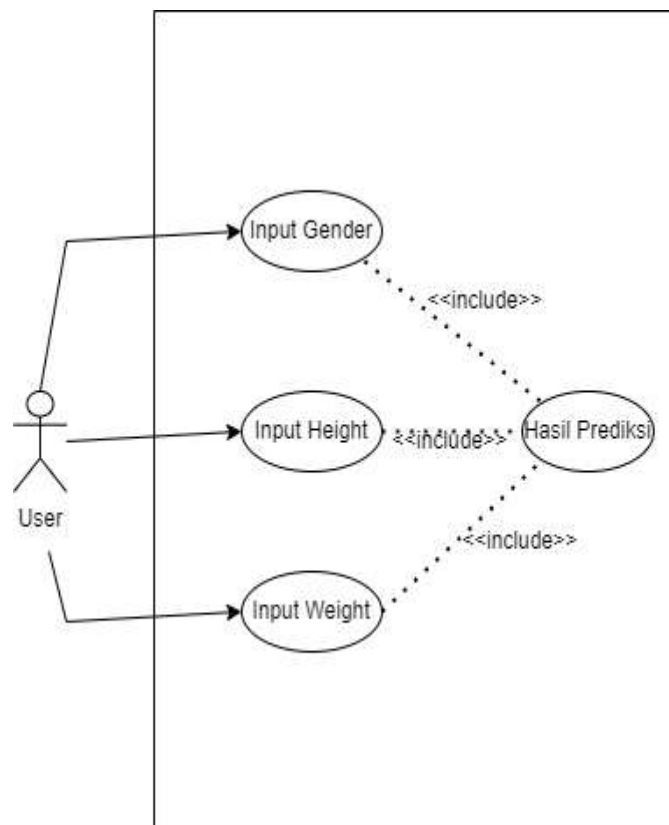
- 1) Hardware
 - a) Processor amd ryzen 3 2200u
 - b) RAM 4GB
 - c) Hardisk kapasitas minimal 500GB atau lebih
- 2) Software
 - a) Google Colaboratory
 - b) Github
 - c) GIT
 - d) Draw.io
 - e) Kaggle
 - f) Figma
- 3) Analisis Kebutuhan Fungsional
 - a) Aplikasi dapat menampilkan halaman prediksi yang memuat input data gender, tinggi badan, berat badan dan tombol prediksi.
 - b) Aplikasi dapat memprediksi hasil input data dan mengeluarkan hasil prediksi

B. Desain

Fungsi dari desain sistem adalah untuk mempermudah dalam implementasi dan penulisan kode, karena telah digambarkan secara jelas. Desain yang telah dibuat penulis untuk klasifikasi obesitas adalah sebagai berikut :

a. *Use Case Diagram*

Use case diagram adalah proses penggambaran yang dilakukan untuk menunjukkan hubungan antara pengguna dengan sistem yang dirancang. Skema dibuat sederhana yang bertujuan untuk memudahkan user dalam memahami informasi yang diberikan.

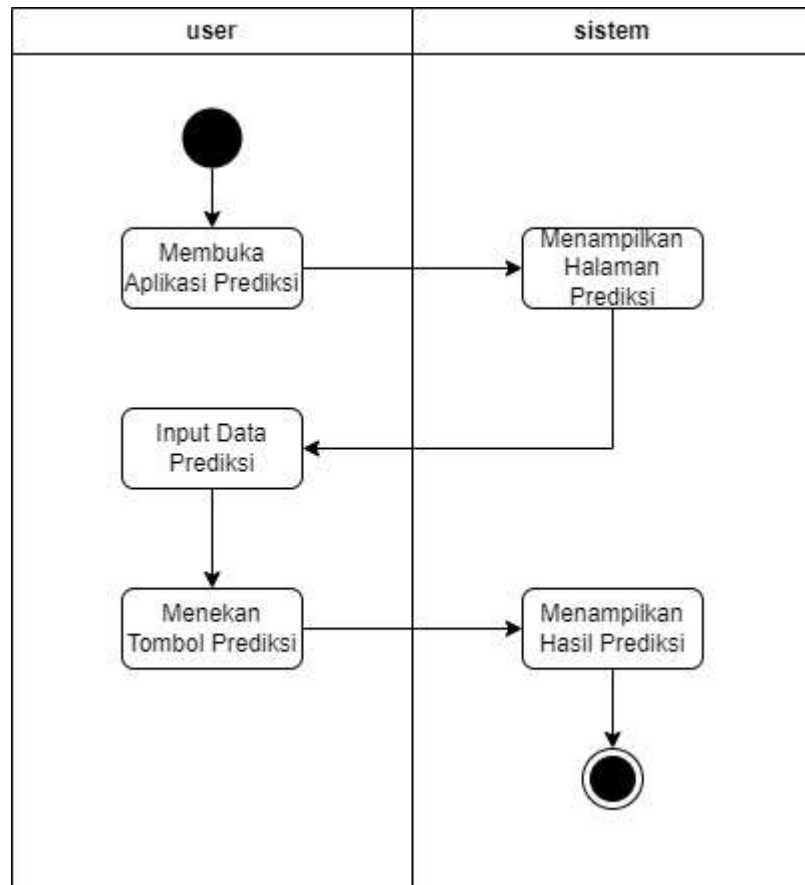


Gambar 4. 1 *Use Case Diagram*

Berdasarkan gambar 4.3, dapat diambil penjelasan bahwa didalam sistem terdapat satu aktor yaitu *user*. *User* mendapatkan akses penuh dalam aplikasi untuk dari input data hingga prediksi.

b. *Activity Diagram*

Activity Diagram merupakan proses penggambaran aktivitas terstruktur yang ada didalam aplikasi sebagaimana aplikasi dimulai hingga berakhir.

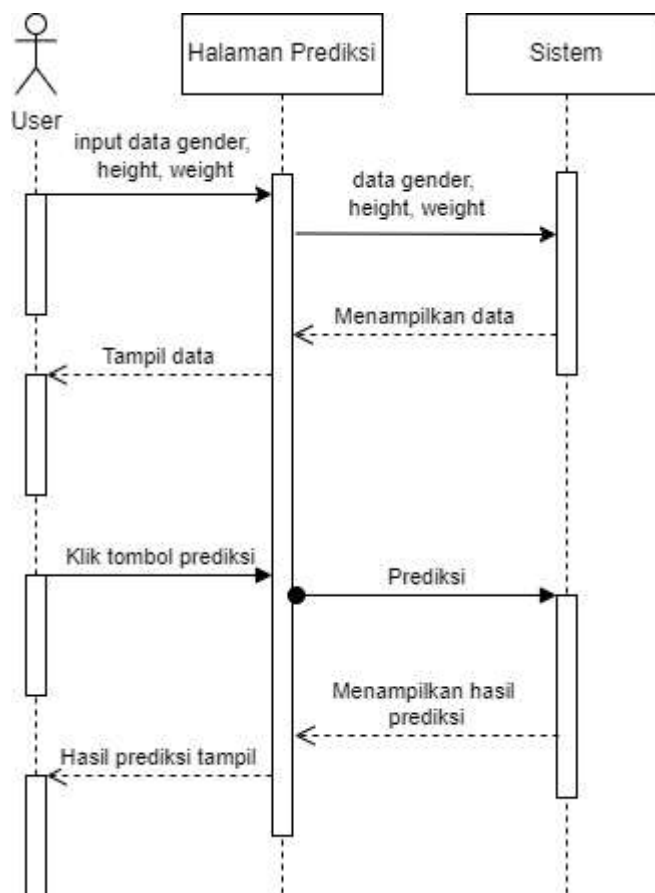


Gambar 4. 2 *Activity Diagram* Halaman Prediksi

Berdasarkan gambar 4.4, digunakan untuk memprediksi tingkat obesitas, merupakan menu yang dikembangkan oleh penulis. Dalam menu ini *user* memasukan input data jenis kelamin, berat badan, tinggi badan, jika inputan sudah sesuai makan user dapat diminta untuk menekan tombol prediksi kemudian sistem akan menampilkan hasil prediksi.

c. *Sequence Diagram*

Sequence diagram digunakan untuk membantudalam memahami persyaratan sistem baru, mendokumentasikan proses dan memvisualisasikan skenario teknis saat sistem sedang dijalankan, sehingga pengguna dapat memahami dan memprediksi bagaimana suatu sistem akan berperilaku.



Gambar 4. 3 *Sequence Diagram* Halaman Prediksi

2. **Desain *User Interface***

User Interface adalah tampilan visual sebuah produk yang menghubungkan sistem dengan pengguna (*user*), yang bertujuan untuk merencanakan dan memvisualisasikan struktur dasar serta tata letak elemen-elemen yang akan ada pada layar. Langkah pertama yang penulis lakukan adalah memilih metode dsain wareframe. *Wireframe* berfungsi

sebagai gambaran kerangka kerja antarmuka, tanpa detail visual atau desain grafis akhir. *Wireframe* adalah langkah penting dalam proses desain UI karena memungkinkan tim untuk fokus pada aspek fungsional dan struktur sebelum melangkah ke tahap desain visual yang lebih rinci.



Gambar 4. 4 Desain Halaman Prediksi

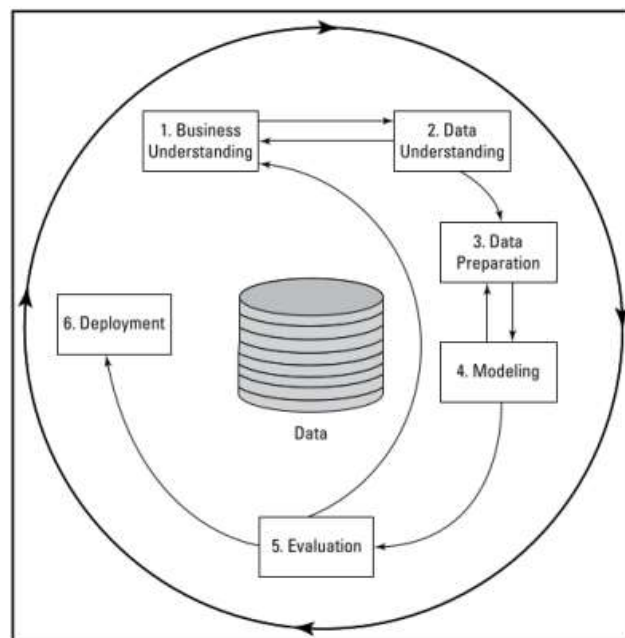
Desain pada gambar 4.4 merupakan rancangan tampilan halaman prediksi yang akan digunakan user atau pengguna. Terdapat beberapa komponen yang disajikan yaitu

- a) Header atau identitas nama aplikasi yang telah dibuat,
- b) Judul aplikasi yang memuat ajakan untuk mengklasifikasikan obesitas terhadap *user* atau pengguna.
- c) Tombol yang berformat 'st.radio' tombol radio berfungsi untuk memilih salah satu dari dua opsi jenis kelamin yang disediakan yaitu *male* atau *female*.
- d) Form yang berisikan input data height atau tinggi badan dalam satuan centimeter(cm).
- e) Kemudian form yang berisikan input data weight atau berat badan user dalam satuan kilogram(kg).

- f) Tombol prediksi yang digunakan untuk memprediksi setelah pengguna telah memasukan data yang sesuai dalam form yang disediakan.
- g) Merupakan kalimat yang muncul berisikan skor BMI pengguna yang telah di prediksi.
- h) Hasil prediksi yang tampil ketika semua form telah diinput dan tombol prediksi yang sudah ditekan.

3. Implementasi Model SVM

Dalam implementasi model SVM digunakan metode *Cross-Industry Standard Process for Data Mining (CRISP-DM)* yang menyediakan kerangka kerja yang terstruktur untuk memahami, menyiapkan, memodelkan, dan mengevaluasi data sehingga dapat digunakan untuk membuat keputusan. Berikut adalah bagaimana setiap fase CRISP-DM diterapkan dalam penggunaan SVM gambar 4.5 :



Gambar 4. 5 Model Proses CRISP-DM

A. Business Understanding

Tujuan penelitian ini adalah untuk membangun model klasifikasi yang dapat mengidentifikasi tingkat obesitas (Sangat Kurus, Kurus, Normal, Kelebihan Berat, Obesitas, dan Obesitas Extreme) dan menghitung skor indeks massa tubuh menggunakan metode SVM dengan akurasi yang baik.

B. Data Understanding

Dataset yang digunakan dalam penelitian ini diunduh dari Kaggle.com, berisi informasi seperti berat badan, tinggi badan, gender, dan label obesitas.

Sebelum melakukan tahapan data preparation penulis melakukan tahapan *read dataset*. Tahapan ini dilakukan untuk menyiapkan data yang kemudian akan di beri label. *Read dataset* dapat dilihat pada gambar 4.6.

```
bmi = pd.read_csv('/content/500_Person_Gender_Height_Weight_Index.csv')  
bmi
```

Gambar 4. 6 *Read Dataset*

dataset yang penulis ambil berasal dari *website Kaggle.com* yang Bernama *500_Person_Gender_Height_Weight_Index.csv*, pengambilan dataset tersebut dikarenakan dataset tersebut berisi 500 sample data dengan kategori berat yang dapat dikategorikan dalam klasifikasi obesitas. Sampel data dapat dilihat pada tabel 4.4.

Tabel 4. 4 Sampel Dataset Obesitas

No	Nama Label	Jumlah Data
1	Sangat Kurus	13
2	Kurus	22
3	Normal	69

4	Kelebihan Berat	68
5	Obesitas	130
6	Obesitas Ekstrim	198

C. Data Preparation

Data preparation merupakan langkah yang dilakukan untuk menyiapkan data sebelum digunakan dalam pelatihan model. Elemen-elemen yang terdapat dalam data *preparation* meliputi *labelling*, *encoding*, normalisasi/standarisasi, dan *split data*. Berikut adalah tahap data *preparation*

1) Labelling Data

Labeling data adalah memberi tanda tertentu pada data yang akan digunakan untuk klasifikasi. Proses ini sangat penting supervised learning, di mana model dapat memproses dari data berlabel untuk memuat prediksi dan klasifikasi. *Labelling data* dapat dilihat pada gambar 4.7.

```

index_labels = {
    0: 'Sangat Kurus',
    1: 'Kurus',
    2: 'Normal',
    3: 'Kelebihan Berat',
    4: 'Obesitas',
    5: 'Obesitas Extreme'
}

plt.figure(figsize=(10, 6))
for category in sorted(bmi['Index'].unique()):
    subset = bmi[bmi['Index'] == category]
    plt.scatter(subset['Height'], subset['Weight'], label=index_labels[category], alpha=0.7)

plt.title('Diagram sebar Height vs Weight')
plt.xlabel('Height (cm)')
plt.ylabel('Weight (kg)')
plt.legend(title='Kategori BMI')
plt.grid(True)
plt.show()

```

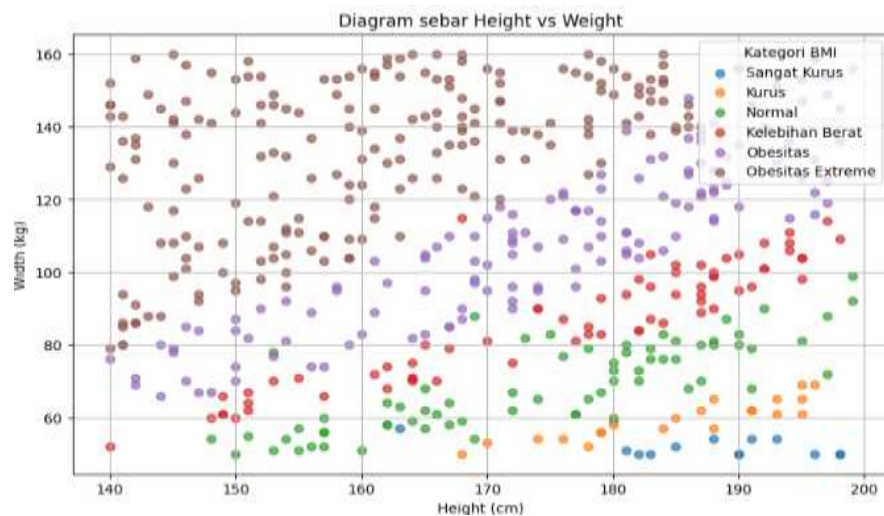
Gambar 4. 7 *labelling Data*

Dalam gambar 4.9, penulis memberi label pada kategori yang berbeda. Setiap nilai dari kolom 'index' dalam dataset diberi label sesuai *dictionary* diatas. Kemudian penulis membuat

diagram sebar (scatter plot) untuk memvisualisasikan hubungan antara tinggi dan berat badan berdasarkan kategori pada *dictionary*. Hasil labelling dapat dilihat pada tabel 4.5.

Tabel 4.5 *Labelling*

Index	Label
Sangat Kurus	0
Kurus	1
Normal	2
Kelebihan Berat	3
Obesitas	4
Obesitas Ekstreme	5



Gambar 4.8 Visualisasi *Labelling* Data

2) *Label Encoding*

Label encoding disini yaitu merubah nilai gender menjadi numerik. Hal ini dilakukan untuk mempermudah pemrosesan data. *Label encoding* dapat dilihat pada gambar 4.9.

```

value_mapping = {'Male': 0, 'Female': 1}
bmi['Gender'] = bmi['Gender'].replace(value_mapping)
bmi

```

Gambar 4. 9 Label encoding Gender

Dalam gambar 4.11, *value mapping* berfungsi mengkonversi tipe data kategori gender menjadi numerik yaitu 'male' menjadi 0 dan 'female' menjadi 1. *Mapping* berfungsi agar dataset konsisten terutama saat pelatihan model.

3) Standarisasi Data

Proses ini membantu dalam memastikan bahwa setiap fitur berkontribusi secara setara dalam analisis atau pemodelan.

```

column_to_exclude = 'Index'
columns_to_scale = [col for col in bmi.columns if col != column_to_exclude]

scaler = StandardScaler()
scaler.fit(bmi[columns_to_scale])
bmi[columns_to_scale] = scaler.transform(bmi[columns_to_scale])
bmi

```

Gambar 4. 10 Standarisasi data

Dalam gambar 4.10, Standarisasi data menggunakan *StandardScaler* kecuali kolom 'Index'. Proses standarisasi mengubah nilai-nilai pada data menjadi nilai dengan distribusi normal dengan mean 0 dan standar deviasi 1. Ini penting untuk memastikan bahwa setiap fitur memiliki kontribusi yang sama pada model.

4) Split Data

berfungsi untuk memisahkan fitur dan label dalam dataset, dan kemudian membagi data menjadi *training* dan *testing* set. Split data dapat dilihat pada gambar 4.11.

```
X = bmi.drop('Index', axis = 1)
y = bmi['Index']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Gambar 4. 11 *Split Data*

Pada gambar 4.13, memisahkan atribut yang digunakan untuk memprediksi label, dengan cara memindahkan atribut *gender*, *height*, dan *weight* kedalam variable x, kemudian memindahkan atribut *index* kedalam variable y, kemudian membagi dataset menjadi dua bagian, 80% data akan digunakan sebagai data *training* dan 20% data akan digunakan untuk *testing*. Dengan membagi data menjadi *training* dan *testing*, kita dapat menghindari overfitting.

D. Modelling

Dalam tahap *modelling* terdapat beberapa tahapan yaitu, mencari parameter yang terbaik seperti 'C', 'gamma'. Selanjutnya menentukan kernel yang sesuai berdasarkan karakteristik data (*linear*, *polynomial*, *rbf*, dan *sigmoid*), kemudian melatih model SVM dengan data pelatihan.

1. Pemilihan Kernel

Proses pertama dalam SVM yang penulis lakukan yaitu mencari parameter yang cocok untuk mendapatkan performa kinerja model yang baik. Dalam mencari parameter yang cocok dapat dilihat pada gambar 4.12.

```

param_grid = [
    {"C": [0.1, 1, 10, 100, 1000], "kernel": ["linear"]},
    {"C": [0.1, 1, 10, 100, 1000], "gamma": [0.001, 0.0001], "kernel": ["rbf"]},
    {
        "C": [0.1, 1, 10, 100, 1000],
        "gamma": [0.001, 0.0001],
        "coef0": [0.0, 1.0],
        "degree": [2, 3, 4],
        "kernel": ["poly"],
    },
]

```

Gambar 4. 12 *Parameter GridSearch*

Dalam gambar 4.13, dengan menggunakan ‘param_grid’, *Grid Search* mencoba berbagai kombinasi parameter untuk menemukan yang terbaik. Kombinasi parameter yang dicoba termasuk berbagai nilai untuk parameter ‘C’, ‘gamma’, ‘coef0’, dan ‘degree’, serta jenis kernel (linear, rbf, poly). Dengan mencoba berbagai kombinasi ini, *Grid Search* bertujuan menemukan parameter yang menghasilkan kinerja terbaik pada data pelatihan.

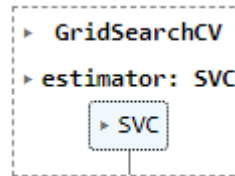
2. *Training Model*

Setelah melakukan *gridsearch* penulis melakukan *training model* menggunakan *Support Vector Classifier* (SVC), *training model* dapat dilihat pada gambar 4.13.

```

clf = svm.SVC()
grid_search = GridSearchCV(estimator=clf, param_grid=param_grid, cv=5, n_jobs=-1)
grid_search.fit(X_train, y_train)

```



```

best_params = grid_search.best_params_
best_estimator = grid_search.best_estimator_
best_params

```

```
{'C': 1000, 'kernel': 'linear'}
```

Gambar 4. 13 *Training Model SVC*

Berdasarkan gambar 4.13, setelah melatih model SVC dengan mencari parameter terbaik menggunakan teknik *Grid Search Cross-Validation*,

Tabel 4. 6 *Hyperparameter Grid Search Cross-Validation*

No	hayperparameter	
1	Parameter Regularisasi C	1000
2	Kernel	Linear

menunjukkan bahwa kernel yang cocok digunakan untuk klasifikasi obesitas adalah kernel : '*linear*' yang berarti data akan dipisahkan secara linier. kemudian dengan nilai parameter regularisasi 'C' : '1000' yang berarti akan mencoba mengklasifikasikan semua titik data pelatihan dengan benar. Setelah melakukan *training* model, tahapan selanjutnya adalah mengevaluasi kinerja model menggunakan '*best_estimator*' yang terdapat dalam *grid search*. Hasil evaluasi model dapat dilihat pada gambar 4.14.

```
y_pred = best_estimator.predict(X_test)
print(accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Gambar 4. 14 Evaluasi Model *Best_Estimator*

3. *Synthetic Minority Over-sampling Technique* (SMOTE)

Tahapan selanjutnya adalah mengatasi imbalance data yang dihasilkan dari parameter regularisasi yang terlalu tinggi. Imbalance data adalah kondisi dalam dataset di mana jumlah sampel dari setiap kelas tidak merata, dengan beberapa kelas memiliki lebih banyak sampel dibandingkan kelas lainnya. Situasi ini sering terjadi dalam masalah klasifikasi, di mana salah satu kelas (kelas mayoritas) memiliki jumlah sampel yang jauh lebih besar dibandingkan dengan kelas lainnya (kelas minoritas). Menangani imbalance data merupakan langkah penting dalam memastikan model machine learning memiliki performa yang baik untuk semua kelas dalam dataset.

SMOTE adalah teknik yang digunakan untuk mengatasi masalah imbalance data dengan cara melakukan *oversampling* pada kelas minoritas. SMOTE menciptakan sampel sintetis baru dengan menginterpolasi antara sampel yang ada dalam kelas minoritas. Implementasi SMOTE dapat dilihat dalam gambar 4.15.

```
smote = SMOTE(random_state=42)
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
```

Gambar 4. 15 SMOTE

Berdasarkan gambar 4.16, menggunakan SMOTE untuk membuat sample sintetis baru dari kelas minoritas dengan membuat contoh sintetis.

- a) *random_state=42*, Parameter ini digunakan untuk mengatur seed pada generator acak yang digunakan oleh SMOTE, sehingga hasil yang dihasilkan dapat direproduksi. Nilai 42 adalah pilihan yang umum digunakan, tetapi bisa berupa nilai integer lainnya.
- b) *X_train_smote*, merupakan Data fitur pelatihan setelah resampling, yang sekarang memiliki kelas minoritas yang disintesis untuk menyeimbangkan jumlah kelas. Kemudian
- c) *y_train_smote*, adalah label kelas pelatihan setelah resampling, dengan kelas minoritas yang telah diperbanyak sehingga seimbang dengan kelas mayoritas.
- d) *fit_resample*, merupakan metode yang digunakan untuk menyesuaikan SMOTE de data pelatihan dan kemudian melakukan resampling untuk menyeimbangkan kelas.
- e) *X_train*, Data fitur pelatihan asli sebelum resampling.
- f) *y_train*, Label kelas pelatihan asli sebelum resampling.

Tabel 4. 7 Sampel Data Setelah SMOTE

No	Nama Label	Jumlah Data
1	Sangat Kurus	159
2	Kurus	159
3	Normal	159
4	Kelebihan Berat	159
5	Obesitas	159
6	Obesitas Ekstrim	159

4. Standarisasi Fitur

Menginisialisasi objek *StandardScaler* untuk Menghitung parameter standarisasi dari data pelatihan yang telah di resample ‘*X_train_smote*’, kemudian menerapkan standarisasi pada data pelatihan yang telah di resample menghasilkan

'X_train_scaled', dan menerapkan standarisasi yang sama pada data pengujian 'X_test' untuk memastikan bahwa fitur pengujian berada pada skala yang sama dengan fitur pelatihan 'X_test_scaled'. Standarisasi fitur SMOTE dapat dilihat pada gambar 4.16.

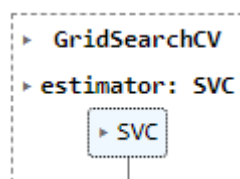
```
scaler = StandardScaler()  
scaler.fit(X_train_smote)  
  
X_train_scaled = scaler.transform(X_train_smote)  
X_test_scaled = scaler.transform(X_test)
```

Gambar 4. 16 Standarisasi Fitur SMOTE

5. Training Model SVM SMOTE

Melatih model SVM dengan data pelatihan yang telah di resample dan di standarisasi 'X_train_scaled', 'y_train_smote'. Training model SMOTE dapat dilihat pada gambar 4.17.

```
clf_smote = svm.SVC()  
grid_search_smote = GridSearchCV(estimator=clf_smote, param_grid=param_grid, cv=5, n_jobs=-1)  
grid_search_smote.fit(X_train_scaled, y_train_smote)
```



Gambar 4. 17 Training Model SMOTE

6. Gradient Boosting

Boosting adalah teknik ensemble dalam *machine learning* yang menggabungkan beberapa model sederhana (*weak learners*) untuk membentuk model yang kuat. *Boosting* bertujuan untuk meningkatkan akurasi prediksi dengan memperbaiki kesalahan yang dibuat oleh model-model sederhana. Metode yang penulis gunakan dalam tahapan *boosting* adalah metode XGBoost,

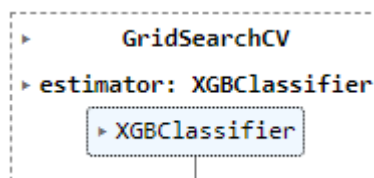
Tahapan dalam *boosting* metode XGBoost dapat dilihat pada gambar 4.18.

```
param_grid = {  
    'n_estimators': [100, 200],  
    'learning_rate': [0.01, 0.1, 0.3],  
    'max_depth': [3, 5, 7],  
    'colsample_bytree': [0.7, 1],  
    'subsample': [0.7, 1]  
}
```

Gambar 4. 18 Parameter XGboost

Pada gambar 4.18, melakukan kombinasi dan setiap nilai parameter akan diuji untuk menentukan hyperparameter terbaik yang memberikan performa optimal.

```
bst = XGBClassifier(objective='multi:softprob', random_state=42)  
grid_search = GridSearchCV(estimator=bst, param_grid=param_grid, cv=5, n_jobs=-1)  
grid_search.fit(X_train, y_train)
```



```
best_params = grid_search.best_params_  
best_estimator = grid_search.best_estimator_  
best_params
```

```
{'colsample_bytree': 1,  
 'learning_rate': 0.1,  
 'max_depth': 3,  
 'n_estimators': 200,  
 'subsample': 0.7}
```

Gambar 4. 19 Training Model XGboost

Dalam gambar 4.19, kode tersebut untuk melakukan pencarian grid hyperparameter untuk model XGBoost dengan tujuan untuk menemukan kombinasi hyperparameter yang

memberikan performa terbaik pada data pelatihan. Kombinasi hyperparameter terbaik ditentukan melalui evaluasi 5-fold cross-validation.

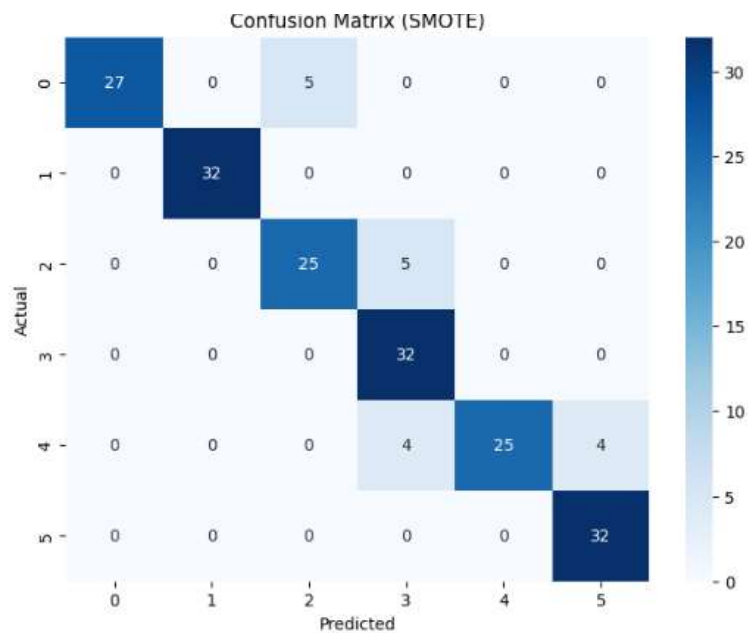
Tabel 4. 8 *hyperparameter 5-Fold Cross-Validation*

No	<i>Hyperparameter</i>	
1	colsample_bytree	1
2	learning_rate	0.1
3	max_depth	3
4	n_estimators	200
5	subsample	0.7

Pada tabel 4.8, penulis menggunakan *5-fold-cross-validation*, dengan `colsample_bytree` 1 yang berarti menggunakan semua fitur yang akan dipilih secara acak yang akan digunakan untuk membangun setiap pohon dalam proses pembentukan model. Menggunakan *learning rate* 0.1 berarti kontribusi dari setiap pohon baru terhadap model akhir dikurangkan menjadi 10%. Kemudian untuk *max_dept* 3 berarti setiap pohon akan memiliki maksimal 3 tingkatan, yang membantu mengurangi overfitting dan menjaga model tetap sederhana. Berikutnya `n_estimator` 200 menunjukkan bahwa model akan membangun 200 pohon secara total, dengan learning rate yang rendah, jumlah pohon yang lebih banyak membantu model mencapai kinerja yang baik. Yang terakhir adalah `subsample` 0.7 yang berarti setiap pohon akan dibangun menggunakan 70% dari data pelatihan yang dipilih secara acak, yang bertujuan untuk memastikan bahwa setiap pohon tidak dilatih pada data yang sama, memberi variasi dalam training model.

E. Evaluation

Setelah melakukan pelatihan model, tahap berikutnya adalah mengevaluasi model. Evaluasi model digunakan untuk menentukan seberapa baik model tersebut dapat digeneralisasikan terhadap pola yang telah dipelajari dari data pelatihan ke data baru yang telah dilihat sebelumnya. *Confusion matrix* digunakan untuk mengevaluasi kinerja model klasifikasi dengan membandingkan hasil prediksi dengan nilai sebenarnya dari data uji, *confusion matrik* membantu dalam memahami kesalahan yang dibuat oleh model serta jenis kesalahan yang terjadi. Untuk melihat hasil evaluasi kinerja model dapat dilihat pada gambar 4.20.



Gambar 4. 20 *Confusion Matrik*

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1
1	0.67	1.00	0.80	4
2	1.00	0.89	0.94	18
3	0.67	1.00	0.80	8
4	0.93	0.87	0.90	30
5	1.00	0.95	0.97	39
accuracy			0.92	100
macro avg	0.88	0.95	0.90	100
weighted avg	0.94	0.92	0.92	100

Gambar 4. 21 Evaluasi Model SMOTE SVM

Pada gambar 4.21, evaluasi model SVM setelah menerapkan teknik SMOTE didapatkan hasil akurasi yang baik sebesar 92%. Setelah melakukan evaluasi model terhadap SVM, penulis mencoba melakukan tahapan boosting agar mendapatkan akurasi yang optimal dengan menggunakan metode XGBoost. Evaluasi model dengan metode XGBoost dapat dilihat pada gambar 4.22.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	1
1	0.67	1.00	0.80	4
2	0.89	0.94	0.92	18
3	0.75	0.75	0.75	8
4	0.92	0.80	0.86	30
5	0.90	0.95	0.92	39
accuracy			0.88	100
macro avg	0.69	0.74	0.71	100
weighted avg	0.88	0.88	0.88	100

Gambar 4. 22 Evaluasi Model XGBoost

Setelah melakukan tahapan evaluasi, penulis menemukan bahwa evaluasi model menggunakan metode SVM memiliki akurasi sebesar 92%, sedangkan metode XGBoost hanya mendapatkan akurasi sebesar 88%, sehingga dapat disimpulkan bahwa metode SVM adalah metode yang cocok untuk digunakan

untuk membuat sistem prediksi klasifikasi obesitas berdasarkan indeks massa tubuh tersebut

Untuk mengetahui nilai dari precision, recall, f1-score terdapat tahapan perhitungannya. Berikut adalah perhitungan yang digunakan :

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Keterangan

True Positive (TP) : kelas positif yang diklasifikasikan positif.

True Negative (TN) : kelas negatif yang diklasifikasikan positif

False Positive (FP) : kelas positif yang diklasifikasikan negatif

False Negative (FN) : kelas negative yang diklasifikasikan negatif

4. Implementasi sistem

Setelah melatih model dengan menggunakan SVM, tahapan berikutnya adalah mendeploy model dan pembuatan aplikasi *web*. Tahapan implementasi sistem adalah sebagai berikut :

a. Pembuatan Streamlit

Dalam penelitian ini penulis menggunakan streamlit sebagai *framework* pembuatan sistem. Streamlit merupakan *tools open-source* yang berbasis *web-development* dengan menggunakan bahasa *python*. Sebelum pembuatan streamlit, penulis membuat *requirements* untuk mendukung pembuatan streamlit. Adapaun *requirements* yang dipakai dalam pembuatan streamlit dapat dilihat dalam tabel 4.9.

Tabel 4. 9 requirements Pembuatan Streamlit

No	Requirements	Versi
1	Streamlit	1.36.0
2	Pandas	2.2.2
3	Numpy	2.0.1
4	Matplotlib	3.9.1
5	Scikit-learn	1.5.1
6	Joblib	1.4.2

Berdasarkan tabel 4.9 terdapat berbagai requirements yang dipakai untuk mengelola dependensi yang dibutuhkan dalam pembuatan projek. Dari beberapa requirements diatas terdapat fungsinya, yaitu yang pertama streamlit yang digunakan untuk membangun aplikasi, pandas yang digunakan mengelola dan menganalisis data dalam bentuk tabel, numpy digunakan untuk komputasi numerik, matplotlib berfungsi untuk memvisualisasi data dalam bentuk grafik dan plot, scikit-learn digunakan untuk Membangun model prediktif, evaluasi model, pengoptimalan hyperparameter, preprocessing data, joblib digunakan untuk menyimpan model dan memuat model yang sudah dilatih.

Tahap berikutnya adalah penulisan kode untuk pembuatan streamlit. Pembuatan kodel diperlukan model dengan format .py yang digunakan untuk proses klasifikasi dan prediksi label. Dalam pembuatan kode juga mengubah label yang awalnya dari encoding yang bertipe integer kemudian saat pelatihan model svm diubah menjadi *string*, dalam pembuatan kode ini untuk label 0 akan dirubag menjadi 'Male' dan 1 akan dirubah menjadi 'Female'. Berikut adalah penulisan kode untuk sistem prediksi obesitas dapat dilihat pada gambar 4.23.

```

%%writefile app.py
import joblib
import streamlit as st

# Load save model dan scaler
model = joblib.load('klasifikasi_obesitas_svm.pkl')
scaler = joblib.load('scaler.pkl')

st.title('Klasifikasi Obesitas App')

# User input
gender = st.radio('Gender', ['Male', 'Female'])
height = st.number_input('Height (cm)', min_value=0.0)
weight = st.number_input('Weight (kg)', min_value=0.0)

# Validasi input
def validate_input(height, weight):
    if height <= 0:
        st.error("Tinggi harus lebih dari 0 cm.")
        return False
    if weight <= 0:
        st.error("Berat harus lebih dari 0 kg.")
        return False
    if height > 300:
        st.error("Tinggi tidak realistis (lebih dari 300 cm).")
        return False
    if weight > 500:
        st.error("Berat tidak realistis (lebih dari 500 kg).")
        return False
    return True

if st.button('Prediksi', key='predict_button'):
    if validate_input(height, weight):
        # hitung skor BMI
        bmi_score = weight / ((height / 100) ** 2)
        st.write(f'Skor BMI Anda: {bmi_score:.2f}')

        gender_num = 0 if gender == 'Female' else 1
        # preprocessing
        input_data = [[gender_num, height, weight]]
        input_scaled = scaler.transform(input_data)

        # Prediksi
        prediction = model.predict(input_scaled)[0]
        index_labels = {
            0: 'Sangat Kurus',
            1: 'Kurus',
            2: 'Normal',
            3: 'Kelebihan Berat',
            4: 'Obesitas',
            5: 'Obesitas Extreme'
        }

        st.write(f'Kategori BMI Anda: {index_labels[prediction]}')

```

Gambar 4. 23 Kode Prediksi Streamlit

b. Hasil Aplikasi

a) Halaman Menu Prediksi

Halaman menu prediksi merupakan halaman utama dalam aplikasi ini. Pada ahalaman ini user dapat memprediksi jenis bmi *user* apakah termasuk dalam kategori obesitas atau tidak. Halaman ini memuat *input* data *gender*, tinggi badan, dan berat badan, kemudian tombol prediksi dan hasil prediksi. Adapun tampilan halaman menu prediksi dapat dilihat pada gambar 4.24.



Gambar 4. 24 Halaman Prediksi

5. Pengujian

Pengujian dilakukan untuk memastikan kualitas aplikasi berfungsi dengan baik. Pada penelitian ini, penulis menggunakan 2 jenis pengujian yaitu *blackbox testing* dan *user acceptance testing*. Untuk pengujian pada penelitian ini dapat dirinci sebagai berikut :

a. Pengujian *Black Box*

Pengujian *blackbox* digunakan untuk memastikan aplikasi berfungsi dengan baik sesuai dengan harapan pengguna. Pengujian dilakukan

oleh tiga dosen informatika. Untuk pengujian pada penelitian ini dapat dilihat sebagai berikut :

1) *Form Pengujian Black Box*

Form prngujian black box dapat dilihat pada tabel 4.10 :

Tabel 4. 10 Pengujian *Black Box*

Uji Coba	Skenario Uji Coba yang diharapkan	Hasil Pengamatan	Keterangan	
			Diterima	Ditolak
Input valid	Pilih gender 'Male', Masukan tinggi 180 cm. Masukan berat 75 kg. Klik tombol prediksi	Skor BMI dihitung dan ditampilkan. Kategori BMI yang benar ditampilkan sesuai dengan model prediksi.		
Input invalid (negative case)	Pilih gender 'Female', Masukan tinggi -160 cm, Masukan berat 50 kg, Klik tombol prediksi	Aplikasi menangani input invalid dan menampilkan pesan kesalahan atau tidak dapat melakukan prediksi.		
Input boundary (boundary case)	Pilih gender 'Male', Masukan tinggi 0 cm, Masukan berat 0 kg, Klik tombol prediksi	Aplikasi menangani input boundary dan menampilkan pesan kesalahan		

Uji Coba	Skenario Uji Coba yang diharapkan	Hasil Pengamatan	Keterangan	
			Diterima	Ditolak
		atau tidak bisa melakukan prediksi.		
Input maksimum (Stress case)	Pilih gender 'Female', Masukan tinggi 250 cm, Masukan berat 200 kg, Klik tombol prediksi	Aplikasi dapat menangani input besar tanpa kesalahan dan memberikan prediksi yang sesuai.		
Input kombinasi	Pilih gender 'Female', Masukan tinggi 185,7 cm, Masukan berat 90,5 kg, Klik tombol prediksi	Aplikasi menghitung skor BMI dengan benar dan menampilkan kategori BMI yang sesuai.		

2) *Form* Hasil Pengujian *Black Box*

Hasil pengujian *black box* dapat dilihat pada gambar 4.11.

Tabel 4. 11 Hasil Pengujian *Black Box*

Uji Coba	Skenario Uji Coba yang diharapkan	Hasil Pengamatan	Keterangan					
			Diterima			Ditolak		
			1	2	3	1	2	3
Input valid	Pilih gender 'Male', Masukan tinggi 180 cm. Masukan berat 75 kg. Klik tombol prediksi	Skor BMI dihitung dan ditampilkan. Kategori BMI yang benar ditampilkan sesuai dengan model prediksi.	√	√	√			
Input invalid (negative case)	Pilih gender 'Female', Masukan tinggi -160 cm, Masukan berat 50 kg, Klik tombol prediksi	Aplikasi menangani input invalid dan menampilkan pesan kesalahan atau tidak dapat melakukan prediksi.	√	√	√			
Input boundary (boundary case)	Pilih gender 'Male', Masukan tinggi 0 cm, Masukan berat 0 kg, Klik tombol prediksi	Aplikasi menangani input boundary dan menampilkan pesan kesalahan atau tidak bisa melakukan prediksi.	√	√	√			

Uji Coba	Skenario Uji Coba yang diharapkan	Hasil Pengamatan	Keterangan					
			Diterima			Ditolak		
			1	2	3	1	2	3
Input maksimum (Stress case)	Pilih gender 'Female', Masukan tinggi 250 cm, Masukan berat 200 kg, Klik tombol prediksi	Aplikasi dapat menangani input besar tanpa kesalahan dan memberikan prediksi yang sesuai.	√	√	√			
Input kombinasi	Pilih gender 'Female', Masukan tinggi 185,7 cm, Masukan berat 90,5 kg, Klik tombol prediksi	Aplikasi menghitung skor BMI dengan benar dan menampilkan kategori BMI yang sesuai.	√	√	√			

3) Kesimpulan Hasil Pengujian *Black Box*

Berdasarkan pengujian *Black Box* dari 5 pengujian pada aplikasi yang didapat dari 3 responden, berikut hasil pengujian *black box*:

➤ Pengujian Pertama

$$\text{Tercapai} = \frac{5}{5} \times 100\% = 100\%$$

$$\text{Gagal} = \frac{0}{5} \times 100\% = 0\%$$

➤ Pengujian Kedua

$$\text{Tercapai} = \frac{5}{5} \times 100\% = 100\%$$

$$\text{Gagal} = \frac{0}{5} \times 100\% = 0\%$$

➤ Pengujian Ketiga

➤ Tercapai = $\frac{5}{5} \times 100\% = 100\%$

➤ Gagal = $\frac{0}{5} \times 100\% = 0\%$

Berdasarkan perhitungan diatas, disimpulkan bahwa dari 5 pengujian oleh 3 responden menghasilkan tingkat keberhasilan pengujian *block box* memiliki presentase 100%, sedangkan gagal memiliki presentase 0%. Dapat disimpulkan bahwa aplikasi berjalan sesuai fungsionalitasnya.

b. Pengujian *User Acceptance Testing* (UAT)

User acceptance testing dilakukan untuk memastikan bahwa aplikasi klasifikasi tingkat obesitas berdasarkan indeks massa tubuh sesuai dengan kebutuhan dan harapan *user*. Pengujian ini merupakan kesempatan dari *user* untuk memastikan bahwa aplikasi siap digunakan dalam lingkup produksi dan operasional dan dapat memberikan umpan balik. UAT diujikan pada 5 responden. Kelayakan pada aplikasi dapat dilihat pada tabel 4.12.

Tabel 4. 12 Pengujian *User Acceptance Testing*

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Angket Aspek kemanfaatan						
1	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi kinerja?					
2	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi keefektifan?					
3	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi kegunaan?					

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Angket aspek kemudahan						
4	Sejauh mana anda merasa aplikasi ini mudah untuk dipahami dan digunakan tanpa bantuan tambahan?					
5	Apakah aplikasi klasifikasi ini mudah digunakan?					
6	Apakah aplikasi ini cukup mudah dipahami dan dijalankan?					
7	Apakah aplikasi klasifikasi ini mudah dipahami?					
Angket aspek <i>User Interface</i>						
8	Apakah tampilan pada aplikasi menarik?					
9	Apakah font tulisan pada aplikasi dapat terbaca dengan baik?					
10	Apakah tata letak pada menu aplikasi sudah sesuai?					
11	Apakah penggunaan warna pada aplikasi sudah terlihat nyaman dan sesuai dengan pengguna?					

Keterangan :

1. Tidak Setuju
2. Kurang Setuju
3. Cukup Setuju
4. Setuju
5. Sangat Setuju

Berikut merupakan hasil kuesioner pengujian *user acceptance testing* yang telah disebar kepada 5 responden. Hasil pengujian UAT dapat dilihat pada tabel 4.13.

Tabel 4. 13 Hasil Pengujian UAT

Pertanyaan	Hasil Pengujian				
	Responden 1	Responden 2	Responden 3	Responden 4	Responden 5
1	5	4	4	4	4
2	4	5	4	4	4
3	5	4	4	4	4
4	5	5	4	4	5
5	5	4	4	4	4
6	5	5	4	3	4
7	4	4	4	3	3
8	4	5	3	3	3
9	4	4	3	3	4
10	5	5	4	3	4
11	3	4	3	3	4
Jumlah Skor	49	49	41	38	43
Persentase	89%	89%	74%	69%	78%
Total	399%				

Dari hasil persentase setiap pertanyaan dari aspek kemanfaatan dan kegunaan, serta aspek *user interface* (UI) yang diuji oleh 5 responden, kemudian dicari nilai rata-ratanya untuk mencapai tingkat penerimaan terdapat yang dibuat oleh responden. Nilai rata-ratanya dapat dihitung dengan menggunakan persamaan sebagai berikut :

$$\text{Presentase rata-rata} = \frac{\text{jumlah total presentase}}{\text{jumlah responden}}$$

$$\text{Presentase rata-rata} = \frac{399\%}{5} = 79.8\%$$

Daftar keterangan kategori presentase sebagai berikut :

0% - 20%	= Tidak Setuju
21% - 40%	= Kurang Setuju
41% - 60%	= Cukup Setuju
61% - 80%	= Setuju
81% - 100%	= Sangat Setuju

Dengan perhitungan yang telah didapatkan diatas presentase rata-rata sebesar 79.8%. dapat diambil kesimpulan bahwa pengujian UAT pada aplikasi ini mendapatkan kategori Baik.

C. Pembahasan

Dari hasil penelitian menggunakan metode *waterfall* diatas, maka aplikasi klasifikasi tingkat obesitas berdasarkan indeks massa tubuh dengan 4 tahapan yang meliputi analisis kebutuhan, desain, implementasi, dan pengujian.

1. Analisis Kebutuhan

Pada tahapan pertama yaitu tahapan analisis kebutuhan. Pada tahapan ini diperlukan analisis kebutuhan yang dipakai untuk pengembangan aplikasi. Beberapa analisis kebutuhan yaitu analisis kebutuhan data, analisis kebutuhan sistem atau aplikasi, dan analisis kebutuhan fungsional. Dalam tahapan ini, kebutuhan untuk aplikasi klasifikasi tingkat obesitas berdasarkan indeks massa tubuh akan dikembangkan dikumpulkan, dipahami, dan diuraikan, secara rinci.

2. Desain

Setelah pengumpulan kebutuhan sistem selesai, tahapan berikutnya adalah desain. Dalam tahapan ini, penulis penulis menggunakan model perancangan Unified Modeling Language (UML). Penulis menggunakan 3 jenis diagram UML. Pertama *use case diagram*, pada digram ini user sebagai aktor yang mempunyai akses penuh dalam aplikasi ini. Kedua *activity diagram*, hanya terdapat satu *activity diagram*

yaitu *activity diagram* halaman menu prediksi. Terakhir jenis diagram yang digunakan pada tahapan desain yaitu *sequence diagram*, terdapat satu *sequence diagram* prediksi, yang menggambarkan interaksi antar objek dalam sebuah sistem secara berurutan dalam satu waktu.

Salain menggunakan *Unified Modeling Language* (UML). Penulis juga menggunakan menggunakan desain antarmuka atau user (UI). Pada desain antarmuka penulis merancang sebuah desain jenis wareframe yang dimana desain masih berwarna hitam putih dan tidak kompleks. Pada desain antarmuka penulis merancang sebuah desain halaman menu prediksi.

3. Implementasi

Tahapan selanjutnya yaitu tahapan implementasi. Pada tahapan ini penulis membagi 2 implementasi yaitu implementasi model SVM dan implementasi sistem. Pada implementasi model SVM, penulis menerapkan model *Cross-Industry Standard Process for Data Mining* (CRISP-DM) yang menyediakan kerangka kerja yang terstruktur untuk memahami, menyiapkan, memodelkan, dan mengevaluasi data sehingga dapat digunakan untuk membuat model SVM. *Output* pada pembuatan model pada penelitian ini yaitu model yang ber ekstensi (.pkl). Setelah (.pkl) didapatkan masuk kedalam implementasi sistem, Dimana pengembangan sistem atau aplikasi berawal dari pembuatan streamlit dan diakhiri dengan hasil aplikasi yang telah dikembangkan.

4. Pengujian

Tahapan terakhir yaitu pengujian yang dilakukan untuk memastikan aplikasi berfungsi dengan baik. Penulis melakukan 2 pengujian yaitu pengujian *Black Box* dan pengujian *User Acceptance Testing*. Pada pengujian *Block Box* mendapatkan hasil presentasi 100% untuk pengujian berhasil, sedangkan pengujian gagal mendapatkan presentasi 0% dari 3 responden dan 5 pengujian diakhiri saran yang

terdapat dalam lampiran. Berikutnya pengujian *User Acceptance Testing* (UAT) mendapatkan hasil baik dengan presentase sebesar 79.8% dari 5 responden dan 11 pertanyaan.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan oleh penulis, dapat ditarik Kesimpulan sebagai berikut :

1. Dari hasil proses pelatihan model *Support Vector Macine* (SVM) menggunakan teknik *Synthetic Minority Over-sampling Technique* (SMOTE) dengan melakukan *oversampling* pada kelas minoritas mendapatkan hasil *Accuracy* sebesar 91%. Proses pelatihan menggunakan 2 jenis *Hyperparameter Grid Search Cross-Validation* untuk meningkatkan proses pelatihan model. Kemudian penulis menerapkan teknik *boosting* menggunakan *XGBoost* untuk meningkatkan performa model, namun setelah pelatihan dilakukan menggunakan *hyperparameter 5-Fold Cross-Validation* hanya mendapatkan *Accuracy* sebesar 88%. Hal ini menunjukkan bahwa metode SVM adalah metode yang cocok untuk digunakan pembuatan model Klasifikasi Tingkat Obesitas Berdasarkan Indeks Massa Tubuh ini.
2. Pengujian *Black Box* yang dilakukan oleh 3 responden dan 5 pengujian. untuk halaman menu prediksi, hasil pengujian mendapatkan tingkat keberhasilan sebesar 100%.
3. Pengujian *User Acceptance Testing* dilakukan oleh 5 responden dan 11 pertanyaan. Aspek yang dipertanyakan adalah aspek kemanfaatan, aspek kemudahan, dan aspek *User Interface* (UI). Hasil pengujian mendapatkan presentase sebesar 79.8% yang berarti mendapatkan kategori baik.
4. Pada penelitian ini diharapkan dapat mengedukasi masyarakat secara luas untuk menerapkan pola hidup yang sehat, agar tidak mengalami kondisi gangguan kesehatan yang berupa obesitas.

B. Saran

Penelitian ini memberikan kontribusi yang signifikan dalam klasifikasi tingkat obesitas berdasarkan Indeks Massa Tubuh (IMT) menggunakan algoritma SVM, disarankan untuk menggunakan dataset yang lebih besar dan lebih beragam untuk meningkatkan generalisasi model. Selain itu, penambahan fitur-fitur lain yang relevan seperti usia, tingkat aktivitas fisik, dan pola makan dapat memperbaiki akurasi model dan memberikan gambaran yang lebih komprehensif mengenai faktor-faktor yang mempengaruhi obesitas.

DAFTAR PUSTAKA

- [1] D. Haryadi, S. Nusa Maniri Jakarta Jl Kramat Raya No, and S. Jakarta Pusat -Indonesia, “Perancangan Aplikasi Perhitungan Indeks Massa Tubuh pada Penderita Obesitas Berbasis Android,” 2018. [Online]. Available: <https://i-msdn.sec.s-msft.com>
- [2] J. V. Wie and M. Siddik, “Penerapan Metode Naïve Bayes Dalam Mengklasifikasi Tingkat Obesitas Pada,” *JOISIE Journal Of Information System And Informatics Engineering*, vol. 6, no. Desember, pp. 69–77, 2022, [Online]. Available: <https://www.kaggle.com/>,
- [3] S. Y. Sibi and A. R. Widiarti, “Klasifikasi Tingkat Obesitas Menggunakan Algoritma KNN.”
- [4] D. Sitanggang, “Model Prediksi Obesitas Dengan Menggunakan Support Vector Machine,” *Jurnal Sistem Informasi dan Ilmu Komputer Prima*), vol. 5, no. 2, 2022.
- [5] H. Nopriandi, I. Kuantan Singingi, and T. Kuantan, “Perancangan Sistem Informasi Registrasi Mahasiswa,” vol. 1, no. 1, 2018.
- [6] S. Arifani and Z. Setiyaningrum, “Faktor Perilaku Berisiko yang Berhubungan Dengan Kejadian Obesitas Pada Usia Dewasa di Provinsi Banten Tahun 2018,” *Jurnal Kesehatan*, vol. 14, no. 2, pp. 160–168, Dec. 2021, doi: 10.23917/jk.v14i2.13738.
- [7] A. P. Wibawa, M. Guntur, A. Purnama, M. Fathony Akbar, and F. A. Dwiyanto, “Metode-metode Klasifikasi,” *Prosiding Seminar Ilmu Komputer dan Teknologi Informasi*, vol. 3, no. 1, 2018.
- [8] P. R. Sihombing and I. F. Yuliati, “Penerapan Metode Machine Learning dalam Klasifikasi Risiko Kejadian Berat Badan Lahir Rendah di Indonesia,” *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 20, no. 2, pp. 417–426, May 2021, doi: 10.30812/matrik.v20i2.1174.

- [9] A. A. Soebroto, “Buku Ajar AI, Machine Learning & Deep Learning,” 2019. [Online]. Available: <https://www.researchgate.net/publication/348003841>
- [10] O. Bangun, H. Mawengkang, and S. Efendi, “Metode Algoritma Support Vector Machine (SVM) Linier Dalam Memprediksi Kelulusan Mahasiswa,” *Jurnal Media Informatika Budidarma*, vol. 6, no. 4, p. 2006, Oct. 2022, doi: 10.30865/mib.v6i4.4572.
- [11] A. Putranto, N. L. Azizah, I. Ratna, I. Astutik, F. Sains, and D. Teknologi, “Sistem Prediksi Penyakit Jantung Berbasis Web Menggunakan Metode SVM dan Framework Streamlit,” 2023. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/heart+disease>
- [12] G. I. E. Soen, M. Marlina, and R. Renny, “Implementasi Cloud Computing dengan Google Colaboratory pada Aplikasi Pengolah Data Zoom Participants,” *JITU: Journal Informatic Technology And Communication*, vol. 6, no. 1, pp. 24–30, Jun. 2022, doi: 10.36596/jitu.v6i1.781.
- [13] B. Riswanto, W. Setiawan, and S. C. E. Sahputro, “Sistem Pakar Diagnosa Stunting pada Balita Berbasis Website Menggunakan Metode Forward Chaining dan Metode Waterfall,” *Digital Transformation Technology*, vol. 3, no. 2, pp. 468–477, Sep. 2023, doi: 10.47709/digitech.v3i2.2881.
- [14] R. W. S Dharwiyanti, “Pengantar unified modeling language (uml),” 2003.
- [15] L. Setiyani, “Techno Xplore Jurnal Ilmu Komputer dan Teknologi Informasi Pengujian Sistem Informasi Inventory Pada Perusahaan Distributor Farmasi Menggunakan Metode Black Box,” 2019.
- [16] I. R. Dhaifullah, M. Muttanifudin, A. A. Salsabila, and M. A. Yakin, “Survei Teknik Pengujian Software,” 2022.
- [17] E. Suprpto, “User Acceptance Testing (UAT) Refreshment PBX Outlet Site BNI Kanwil Padang,” *Jurnal Civronlit Unbari*, vol. 6, no. 2, p. 54, Oct. 2021, doi: 10.33087/civronlit.v6i2.85.

- [18] D. Normawati and S. A. Prayogi, “Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter,” 2021.

LAMPIRAN

Lampiran 1 Pengujian *Black Box*

LEMBAR PENGUJIAN BLACK BOX KLASIFIKASI TINGKAT OBESITAS BERDASARKAN INDEKS MASSA TUBUH

A. Identitas Penguji

Nama : Ramadhan Renaldy, S.kom, m.kom
Jabatan : Dosen Informatika
Tanggal Pengujian : 31 Juli 2024

B. Identitas Peneliti

Nama : Naufal Agung Rizkiana
NPM : 20670051

C. Tabel Pengujian

1. Pengujian Black Box User

Uji Coba	Skenario Uji Coba yang diharapkan	Hasil Pengamatan	Keterangan	
			Diterima	Ditolak
Input valid	Pilih gender 'Male', Masukan tinggi 180 cm. Masukan berat 75 kg. Klik tombol prediksi	Skor BMI dihitung dan ditampilkan. Kategori BMI yang benar ditampilkan sesuai dengan model prediksi.	✓	
Input invalid (negative case)	Pilih gender 'Female', Masukan tinggi -160 cm, Masukan berat 50 kg, Klik tombol prediksi	Aplikasi menangani input invalid dan menampilkan pesan kesalahan	✓	

		atau tidak dapat melakukan prediksi.		
Input boundary (boundary case)	Pilih gender 'Male', Masukan tinggi 0 cm. Masukan berat 0 kg. Klik tombol prediksi	Aplikasi menangani input boundary dan menampilkan pesan kesalahan atau tidak bisa melakukan prediksi.	✓	
Input maksimum (Stress case)	Pilih gender 'Female', Masukan tinggi 250 cm, Masukan berat 200 kg, Klik tombol prediksi	Aplikasi dapat menangani input besar tanpa kesalahan dan memberikan prediksi yang sesuai.	✓	
Input kombinasi	Pilih gender 'Female', Masukan tinggi 185,7 cm. Masukan berat 90,5 kg. Klik tombol prediksi	Aplikasi menghitung skor BMI dengan benar dan menampilkan kategori BMI yang sesuai.	✓	

D. Saran dan Masukan

1. Sediakan skema sistem yang lengkap
2. Setelah hasil klasifikasi berikan penjelasan dan juga saran kepada user sesuai hasil klasifikasi

Penguji



(Ramadhan Renaldi.....)

LEMBAR PENGUJIAN BLACK BOX
KLASIFIKASI TINGKAT OBESITAS BERDASARKAN INDEKS MASSA
TUBUH

A. Identitas Penguji

Nama : Noora O.N.
 Jabatan : Dosen
 Tanggal Pengujian : 31 Juli 2024.

B. Identitas Peneliti

Nama : Naufal Agung Rizkina
 NPM : 20670051

C. Tabel Pengujian

1. Pengujian Black Box User

Uji Coba	Skenario Uji Coba yang diharapkan	Hasil Pengamatan	Keterangan	
			Diterima	Ditolak
Input valid	Pilih gender 'Male', Masukan tinggi 180 cm. Masukan berat 75 kg. Klik tombol prediksi <i>Bisa melakukan prediksi.</i>	Skor BMI dihitung dan ditampilkan. Kategori BMI yang benar ditampilkan sesuai dengan model prediksi.	✓	
Input invalid (negative case)	Pilih gender 'Female', Masukan tinggi -160 cm. Masukan berat 50 kg. Klik tombol prediksi <i>Bisa melakukan prediksi.</i>	Aplikasi menangani input invalid dan menampilkan pesan kesalahan	✓	

		atau tidak dapat melakukan prediksi.		
Input boundary (boundary case)	Pilih gender 'Male', Masukan tinggi 0 cm, Masukan berat 0 kg, Klik tombol prediksi	Aplikasi menangani input boundary dan menampilkan pesan kesalahan atau tidak bisa melakukan prediksi.	✓	
Input maksimum (Stress case)	Pilih gender 'Female', Masukan tinggi 250 cm, Masukan berat 200 kg, Klik tombol prediksi	Aplikasi dapat menangani input besar tanpa kesalahan dan memberikan prediksi yang sesuai.	✓	
Input kombinasi	Pilih gender 'Female', Masukan tinggi 185,7 cm, Masukan berat 90,5 kg, Klik tombol prediksi	Aplikasi menghitung skor BMI dengan benar dan menampilkan kategori BMI yang sesuai.	✓	

D. Saran dan Masukan

agar ~~pk~~ ukr mendapatkan manfaat
boleh diberikan solusi / so what
falo sy terklasifikasi x.

Penguji


(.....)
Noora .

LEMBAR PENGUJIAN BLACK BOX
KLASIFIKASI TINGKAT OBESITAS BERDASARKAN INDEKS MASSA
TUBUH

A. Identitas Penguji

Nama : Nur Hafidhah Dwi MS, MKom
 Jabatan : Dosen
 Tanggal Pengujian : 31 Juli 2024

B. Identitas Peneliti

Nama : Naufal Agung Rizkina
 NPM : 20670051

C. Tabel Pengujian

1. Pengujian Black Box User

Uji Coba	Skenario Uji Coba yang diharapkan	Hasil Pengamatan	Keterangan	
			Diterima	Ditolak
Input valid	Pilih gender 'Male', Masukan tinggi 180 cm, Masukan berat 75 kg. Klik tombol prediksi	Skor BMI dihitung dan ditampilkan. Kategori BMI yang benar ditampilkan sesuai dengan model prediksi.	✓	
Input invalid (negative case)	Pilih gender 'Female', Masukan tinggi -160 cm, Masukan berat 50 kg. Klik tombol prediksi	Aplikasi menangani input invalid dan menampilkan pesan kesalahan	✓	

		atau tidak dapat melakukan prediksi.		
Input boundary (boundary case)	Pilih gender 'Male', Masukan tinggi 0 cm, Masukan berat 0 kg, Klik tombol prediksi	Aplikasi menangani input boundary dan menampilkan pesan kesalahan atau tidak bisa melakukan prediksi.	✓	
Input maksimum (Stress case)	Pilih gender 'Female', Masukan tinggi 250 cm, Masukan berat 200 kg, Klik tombol prediksi	Aplikasi dapat menangani input besar tanpa kesalahan dan memberikan prediksi yang sesuai.	✓	
Input kombinasi	Pilih gender 'Female', Masukan tinggi 185,7 cm, Masukan berat 90,5 kg, Klik tombol prediksi	Aplikasi menghitung skor BMI dengan benar dan menampilkan kategori BMI yang sesuai.	✓	

D. Saran dan Masukan

Penguji


(Nir. Cahya, S1, MS, MKM)

Lampiran 2 lembar Pengujian UAT

**LEMBAR PENGUJIAN UAT (USER ACCEPTANCE TESTING)
KLASIFIKASI TINGKAT OBESITAS BERDASARKAN INDEKS MASSA
TUBUH**

- A. Identitas Penguji :
- Nama : *Muhammad Afrizal Pico*
- Tanggal Pengujian : *Sebelum 30 Juli 2024*
- B. Identitas Peneliti
- Nama : Naufal Agung Rizkita
- NPM : 20670051
- C. Tabel Pengujian

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Angket Aspek kemanfaatan						
1	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi kinerja?				✓	
2	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi keefektifan?				✓	
3	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi kegunaan?				✓	
Angket aspek kemudahan						
4	Sejauh mana anda merasa aplikasi ini mudah untuk dipahami dan digunakan tanpa bantuan tambahan?				✓	
5	Apakah aplikasi klasifikasi ini mudah digunakan?				✓	
6	Apakah aplikasi ini cukup mudah dipahami dan dijalankan?			✓		
7	Apakah aplikasi klasifikasi ini mudah dipahami?			✓		
Angket aspek User Interface						
8	Apakah tampilan pada aplikasi menarik?			✓		

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
9	Apakah font tulisan pada aplikasi dapat terbaca dengan baik?			✓		
10	Apakah tata letak pada menu aplikasi sudah sesuai?			✓		
11	Apakah penggunaan warna pada aplikasi sudah terlihat nyaman dan sesuai dengan pengguna?			✓		

Penguji



(Muhammad Apriadi P...)

**LEMBAR PENGUJIAN UAT (USER ACCEPTANCE TESTING)
 KLASIFIKASI TINGKAT OBESITAS BERDASARKAN INDEKS MASSA
 TUBUH**

- A. Identitas Penguji :
 Nama : Ri2a Ashari
 Tanggal Pengujian : 30 Juli 2024
- B. Identitas Peneliti
 Nama : Naufal Agung Rizkina
 NPM : 20670051
- C. Tabel Pengujian

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Angket Aspek kemanfaatan						
1	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi kinerja?				✓	
2	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi keefektifan?				✓	
3	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi kegunaan?				✓	
Angket aspek kemudahan						
4	Sejauh mana anda merasa aplikasi ini mudah untuk dipahami dan digunakan tanpa bantuan tambahan?				✓	
5	Apakah aplikasi klasifikasi ini mudah digunakan?				✓	
6	Apakah aplikasi ini cukup mudah dipahami dan dijalankan?				✓	
7	Apakah aplikasi klasifikasi ini mudah dipahami?				✓	
Angket aspek User Interface						
8	Apakah tampilan pada aplikasi menarik?			✓		

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
9	Apakah font tulisan pada aplikasi dapat terbaca dengan baik?			✓		
10	Apakah tata letak pada menu aplikasi sudah sesuai?				✓	
11	Apakah penggunaan warna pada aplikasi sudah terlihat nyaman dan sesuai dengan pengguna?			✓		

Penguji



(.....Rizal Ashari.....)

**LEMBAR PENGUJIAN UAT(*USER ACCEPTANCE TESTING*)
**KLASIFIKASI TINGKAT OBESITAS BERDASARKAN INDEKS MASSA
TUBUH****

A. Identitas Penguji : *DIVA Dewarna P*
Nama :
Tanggal Pengujian : *30 Juli 2024*

B. Identitas Peneliti
Nama : *Naufal Agung Rizkina*
NPM : *20670051*

C. Tabel Pengujian

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
<i>Angket Aspek kemanfaatan</i>						
1	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi kinerja?				✓	
2	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi keefektifan?					✓
3	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi kegunaan?				✓	
<i>Angket aspek kemudahan</i>						
4	Sejauh mana anda merasa aplikasi ini mudah untuk dipahami dan digunakan tanpa bantuan tambahan?					✓
5	Apakah aplikasi klasifikasi ini mudah digunakan?				✓	
6	Apakah aplikasi ini cukup mudah dipahami dan dijalankan?					✓
7	Apakah aplikasi klasifikasi ini mudah dipahami?				✓	
<i>Angket aspek User Interface</i>						
8	Apakah tampilan pada aplikasi menarik?					✓

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
9	Apakah font tulisan pada aplikasi dapat terbaca dengan baik?				✓	
10	Apakah tata letak pada menu aplikasi sudah sesuai?					✓
11	Apakah penggunaan warna pada aplikasi sudah terlihat nyaman dan sesuai dengan pengguna?				✓	

Penguji



(Dina Dewara f.....)

**LEMBAR PENGUJIAN UAT (USER ACCEPTANCE TESTING)
 KLASIFIKASI TINGKAT OBESITAS BERDASARKAN INDEKS MASSA
 TUBUH**

A. Identitas Penguji : *Erna Rosalina*
 Nama :
 Tanggal Pengujian : *30 Juli 2024*

B. Identitas Peneliti
 Nama : Naufal Agung Rizkina
 NPM : 20670051

C. Tabel Pengujian

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Angket Aspek kemanfaatan						
1	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi kinerja?					✓
2	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi keefektifan?				✓	
3	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi kegunaan?					✓
Angket aspek kemudahan						
4	Sejauh mana anda merasa aplikasi ini mudah untuk dipahami dan digunakan tanpa bantuan tambahan?					✓
5	Apakah aplikasi klasifikasi ini mudah digunakan?					✓
6	Apakah aplikasi ini cukup mudah dipahami dan dijalankan?					✓
7	Apakah aplikasi klasifikasi ini mudah dipahami?				✓	
Angket aspek User Interface						
8	Apakah tampilan pada aplikasi menarik?				✓	

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
9	Apakah font tulisan pada aplikasi dapat terbaca dengan baik?				✓	
10	Apakah tata letak pada menu aplikasi sudah sesuai?					✓
11	Apakah penggunaan warna pada aplikasi sudah terlihat nyaman dan sesuai dengan pengguna?			✓		

Penguji


 (.....)

**LEMBAR PENGUJIAN UAT (USER ACCEPTANCE TESTING)
KLASIFIKASI TINGKAT OBESITAS BERDASARKAN INDEKS MASSA
TUBUH**

- A. Identitas Penguji : Fadh Dumental Juannah
 Nama :
 Tanggal Pengujian : 30 Juli 2024
- B. Identitas Peneliti
 Nama : Naufal Agung Rizkina
 NPM : 20670051
- C. Tabel Pengujian

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
Angket Aspek kemanfaatan						
1	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi kinerja?				✓	
2	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi keefektifan?				✓	
3	Apakah aplikasi ini bermanfaat bagi pengguna dalam segi kegunaan?				✓	
Angket aspek kemudahan						
4	Sejauh mana anda merasa aplikasi ini mudah untuk dipahami dan digunakan tanpa bantuan tambahan?					✓
5	Apakah aplikasi klasifikasi ini mudah digunakan?				✓	
6	Apakah aplikasi ini cukup mudah dipahami dan dijalankan?				✓	
7	Apakah aplikasi klasifikasi ini mudah dipahami?			✓		
Angket aspek User Interface						
8	Apakah tampilan pada aplikasi menarik?			✓		

No	Pertanyaan	Skor				
		Tidak Setuju	Kurang Setuju	Cukup Setuju	Setuju	Sangat Setuju
9	Apakah font tulisan pada aplikasi dapat terbaca dengan baik?				✓	
10	Apakah tata letak pada menu aplikasi sudah sesuai?				✓	
11	Apakah penggunaan warna pada aplikasi sudah terlihat nyaman dan sesuai dengan pengguna?				✓	

Penguji


 (..... Fachri D.J.)

Lampiran 3 Lembar Pembimbingan skripsi



UNIVERSITAS PGRI SEMARANG
FAKULTAS TEKNIK DAN INFORMATIKA

Kampus : Jalan Sidodadi Timur No 24 Dr. Cipto, Semarang - Indonesia 50125

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

LEMBAR PEMBIMBINGAN SKRIPSI

Nama Mahasiswa : Naufal Agung Rizkina
 N P M : 20670051
 Program Studi : Informatika
 Judul Skripsi : Klasifikasi Tingkat Obentasi Berdasarkan Indeks Massa Tubuh Menggunakan Algoritma Support Vector Machine (SVM)
 Dosen Pembimbing I : Aris Tri Joko Harjanto, S.Kom, M.Kom
 Dosen Pembimbing II : Nugroho Dwi Saputro S.Kom, M.Kom

No	Hari, tanggal	Uraian Bimbingan	Paraf
	22 April 2024	Judul BAB I AOC BAB I	2/3/24
	29 April 2024	BAB II	3/3/24
	30 April 2024	AOC BAB II	3/3/24
	8 Mei 2024	BAB III	3/3/24
	24 Juli 2024	AOC BAB III	3/3/24
	5 Agustus 2024	BAB IV	3/3/24
		AOC uraian.	3/3/24

Dosen Pembimbing I,

Aris Tri Joko Harjanto, M.Kom
 NIDN. 0619048202

Mahasiswa,

Naufal Agung Rizkina
 NPM. 20670051



UNIVERSITAS PGRI SEMARANG
FAKULTAS TEKNIK DAN INFORMATIKA

Kampus : Jalan Sidodadi Timur No 24 Dr. Cipto, Semarang - Indonesia 50125

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

LEMBAR PEMBIMBINGAN SKRIPSI

Nama Mahasiswa : Naufal Agung Rizkina
 N P M : 20670071
 Program Studi : Informatika
 Judul Skripsi : Klasifikasi Tingkat Obesitas Berdasarkan Indeks Massa Tubuh Menggunakan Algoritma Support Vector Machine (SVM)
 Dosen Pembimbing I : Anis Tri Joko
 Dosen Pembimbing II : Nugroho Dwi Saputro, S.Kom., M.Kom.

No	Hari, tanggal	Uraian Bimbingan	Paraf
	Senin 22 Juli 2024	Judul	
	Selasa 23 Juli 2024	Bab I, latar belakang data teks	
	Rabu 24 Juli 2024	Bab II Def - Arayabaga Sistem	
	Kamis 25 Juli 2024	Bab III	
	Jumat 26 Juli 2024	Bab I, II, III Def dan data teks	
	Senin 29 Juli 2024	Bab 1 ³ pemodelan data	
	Selasa 30 Juli 2024	Pemodelan dan manual coding	
	Kamis 1 Agustus 2024	tanpaoba 1 ensemble	
	Jumat 2 Agustus 24	uji black box	
	Senin 5 Agustus 2024	Bab V aka. Abstrak Att 1 coding	

Dosen Pembimbing II,

Nugroho Dwi Saputro, S.Kom., M.Kom
 NPP. 178801537

Mahasiswa,

Naufal Agung Rizkina
 NPM. 20670071

Lampiran 4 Perhitungan Manual SVM

Sampel Perhitungan SVM

$$x_1 = [160, 90, 1]$$

$$x_2 = [170, 70, 1]$$

$$y_1 = -1 \quad = \text{Obesitas}$$

$$y_2 = 1 \quad = \text{Normal}$$

$$\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

Dengan kendala

$$\sum_{j=1}^n \alpha_j y_j = 0 \text{ dan } \alpha_i \geq 0$$

Menghitung kernel linier

$$K(x_i, x_j) = x_i \cdot x_j = 160 \times 170 + 90 \times 70 + 1 \times 1 = 33501$$

Dual problem

Maximize

$$\alpha_1 + \alpha_2 - \frac{1}{2} (\alpha_1^2 K(x_1, x_1) + \alpha_2^2 K(x_2, x_2) + 2\alpha_1 \alpha_2 y_1 y_2 K(x_1, x_2))$$

Dengan kendala

$$\alpha_1 y_1 + \alpha_2 y_2 = 0$$

Menggunakan kendala

Karena $y_1 = -1$ dan $y_2 = 1$, dengan kendalanya

$$-\alpha_1 + \alpha_2 = 0 \quad \alpha_1 = \alpha_2$$

Substitusi kendala ke dual objective

$$2\alpha - \frac{1}{2} (\alpha^2 K(x_1, x_1) + \alpha^2 K(x_2, x_2) + 2\alpha(-1)(1)K(x_1, x_2))$$

Menghitung kernel untuk titik data yang sama

Kerne linier, $K(x_i, x_i) = (x_i \cdot x_i)$:

$$K(x_1, x_1) = 160^2 + 90^2 + 1^2 = 25.600 + 8.100 + 1 = 33.701$$

$$K(x_2, x_2) = 170^2 + 70^2 + 1^2 = 28.900 + 4.900 + 1 = 33.801$$

Dual objective kernel linier

$$\begin{aligned} \text{Maximize } \alpha_2 - \frac{1}{2} (\alpha^2 (33.701 + 33.801 - 2 \cdot 33.501)) \\ = 2\alpha - \frac{1}{2} (\alpha^2 (33.701 + 33.801 - 67.002)) \\ = 2\alpha - \frac{1}{2} (\alpha^2 (500)) \\ = 2\alpha_2 - 250\alpha^2 \end{aligned}$$

Mencari nilai maksimum

$$\frac{d}{d\alpha} (2\alpha - 250\alpha^2) = 2 - 500\alpha = 0$$

$$250\alpha = 2$$

$$\alpha = \frac{2}{500} = \frac{1}{250}$$

Menghitung vector bobot w

$$w = \sum_{i=1}^n \alpha_i y_i x_i$$

$$w = \alpha_1 y_1 x_1 + \alpha_2 y_2 x_2$$

Dengan $\alpha_i = \alpha_i = \frac{1}{250}$

$$w = \frac{1}{250} \cdot (-1) \cdot [160, 90, 1] + \frac{1}{250} \cdot 1 [170, 70, 1]$$



$$w = \left[-\frac{160}{250}, -\frac{90}{250}, -\frac{1}{250} \right] + \left[\frac{170}{250}, \frac{70}{250}, \frac{1}{250} \right]$$

$$w = [-0.64, -0.36, -0.004] + [0.68, 0.28, 0.004]$$

$$w = [0.04, -0.08, 0]$$

Menghitung bias b

$$b = y_1 - w \cdot x_1$$

$$b = -1 - ([0.04, -0.08, 0] \cdot [160, 90, 1])$$

$$b = -1 - (0.04 \cdot 160 + (-0.08) \cdot 90 + 0.1)$$

$$b = -1 - (6.4 - 7.2)$$

$$b = -1 - (-0.8)$$

$$b = -1 + 0.8$$

$$b = -0.2$$

Vector bobot adalah [0.04, -0.08, 0]

Bias adalah -0.2

Prediksi data baru

Data [160, 90, 1]

$$f(x) = w \cdot x + b$$

$$f([160, 90, 1]) = [0.04, -0.08, 0] \cdot [160, 90, 1] + (-0.2)$$

$$= (0.04 \cdot 160) + (-0.08 \cdot 90) + (0.1) - 0.2$$

$$= 6.4 - 7.2 - 0.2$$

$$= -1$$

Prediksi untuk data [160,90,1] adalah -1, yang berarti data tersebut masuk dalam kategori obesitas.