IMPLEMENTASI ALGORITMA *SUPPORT VECTOR MACHINE* UNTUK SISTEM DIAGNOSA PENYAKIT KANKER PAYUDARA DENGAN OPTIMASI *GRIDSEARCH*

SKRIPSI

Disusun oleh :

CHAISAR ABI PRASETYO

201011400216

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PAMULANG

TANGERANG SELATAN

2024

IMPLEMENTASI ALGORITMA *SUPPORT VECTOR MACHINE* UNTUK SISTEM DIAGNOSA PENYAKIT KANKER PAYUDARA DENGAN OPTIMASI *GRIDSEARCH*

SKRIPSI

Diajukan untuk melengkapi salah satu syarat memperoleh

Gelar Sarjana Komputer

Disusun oleh :

CHAISAR ABI PRASETYO

201011400216

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PAMULANG

TANGERANG SELATAN

2024

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini :

NIM : 201011400216

Nama : CHAISAR ABI PRASETYO

Program Studi : TEKNIK INFORMATIKA

Fakultas : ILMU KOMPUTER

Jenjang Pendidikan : STRATA 1

Menyatakan bahwa skripsi yang saya buat dengan judul:

IMPLEMENTASI ALGORITMA SUPPORT VECTOR MACHINE UNTUK SISTEM DIAGNOSA PENYAKIT KANKER PAYUDARA DENGAN OPTIMASI GRIDSEARCH

1. Merupakan hasil karya tulis ilmiah sendiri, bukan merupakan karya yang pernah diajukan unutk memperoleh gelar akademik oleh pihak lain, dan bukan merupakan hasil plagiat.
2. Saya ijinkan untuk dikelola oleh Universitas Pamulang sesuai dengan norma hokum dan etika yang berlaku.

Pernyataan ini saya buat dengan penuh tanggung jawab dan saya bersedia menerima konsekuensi apapun sesuai dengan aturan yang berlaku apabila di kemudian hari pernyataan ini tidak benar.

Tangeran Selatan, September 2024

(Chaisar Abi Prasetyo)

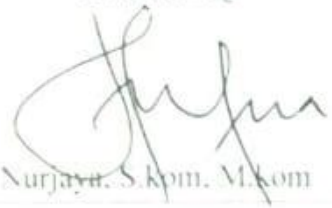
LEMBAR PERSETUJUAN

NIM	:	201011400216
Nama	:	CHAISAR ABI PRASETYO
Program Studi	:	TEKNIK INFORMATIKA
Fakultas	:	ILMU KOMPUTER
Jenjang Pendidikan	:	STRATA 1
Judul Skripsi	:	IMPLEMENTASI ALGORITMA SUPPORT VECTOR MACHINE UNTUK SISTEM DIAGNOSA PENYAKIT KANKER PAYUDARADENGAN OPTIMASI GRIDSEACRH

Skripsi ini telah diperiksa dan disetujui oleh pembimbing untuk persyaratan siding skripsi.

Tangerang Selatan, Oktober 2024

Pembimbing



Nurjaya, S.kom, M.kom

NIDN : 0405078502

Mengetahui,

Ketua Program Studi

Dr. Eng. Ahmad Musyafa, S.kom, M.Kom

NIDN : 042501860

LEMBAR PERSETUJUAN

NIM	:	2010114400216
Nama	:	CHAISAR ABI PRASETYO
Program Studi	:	TEKNIK INFORMATIKA
Fakultas	:	ILMU KOMPUTER
Jenjang Pendidikan	:	STRATA 1
Judul Skripsi	:	IMPLEMENTASI ALGORITMA SUPPORT VECTOR MACHINE UNTUK SISTEM DIAGNOSA PENYAKIT KANKER PAYUDARADENGAN OPTIMASI GRIDSEACRH

Skripsi ini telah diperiksa dan disetujui oleh pembimbing untuk persyaratan siding skripsi.

Tangerang Selatan, Oktober 2024

Penguji 1

Nurhayati, S.Kom, M.Kom

NIDN : 0428059005

Penguji 2

Muhammad Anis, S.T, M.Kom

NIDN : 041488706

Pembimbing

Nurjaya, S.kom, M.kom

NIDN : 0405078502

Mengetahui,

Ketua Program Studi

Dr. Eng. Ahmad Musyafa, S.kom, M.Kom

NIDN : 042501860

DAFTAR ISI

DAFTAR ISI ..... **v**

DAFTAR GAMBAR ..... **viii**

DAFTAR TABEL ..... **ix**

BAB I PENDAHULUAN **1**

1.1 Latar Belakang Masalah 1

1.2 Identifikasi Masalah 2

1.3 Rumusan Masalah 2

1.4 Batasan Masalah 3

1.5 Tujuan Penelitian 3

1.6 Manfaat Penelitian 3

1.7 Metodologi Penelitian 4

1.8 Sistematika Penulisan 4

BAB II TINJUAN PUSTAKA DAN LANDASAN TEORI **6**

2.1 Tinjauan Pustaka 6

2.2 Landasan Teori 10

2.2.1 Data Mining 10

2.2.2 *Machine Learning* 13

2.2.3 *Support Vector Machine* 15

2.2.3.1 Tahapan SVM 17

2.2.4 *Gridsearch* 20

2.2.4.1 Tahapan *Gridseacrh* 20

2.2.4.2 Skema Perhitungan *Gridsearch* 21

2.2.4.3 Hasil Tuning *Hyperparameter* 22

2.2.5 Pengujian Model 22

2.2.5.1 *Confusion matrix* 22

2.2.5.2 *Accuracy* 23
2.2.5.3 *Precision* 23
2.2.5.4 *Recall* 24
2.2.5.5 *F1-Score* 24
2.2.6 *Python* 24
2.2.7 *Library Python* 26
2.2.7.1 *Scikit-learn* 26
2.2.7.2 *NumPy* 26
2.2.7.3 *Pandas* 27
2.2.7.4 *Seaborn* 27
2.2.7.5 *Matplotlib* 28
2.2.8 Aplikasi Pendukung 29
2.2.8.1 *Jupyter Notebook* 29
2.2.9 Kerangka Pemikiran 29
2.3 Tinjauan Objek 30
2.3.1 Kanker Payudara 30
2.3.2 Diagnosa Penyakit Kanker Payudara 31
BAB III METODE PENELITIAN 35
3.1 Analisa Kebutuhan 35
3.2 Teknik Analisis 35
3.2.1 Perancangan Penelitian 35
3.3 Dataset 39
BAB IV HASIL DAN PEMBAHASAN 43
4.1 Hasil 43
4.1.1 Persiapan data 43
4.1.2 Implementasi Metode 43

4.2 *Data Collection* 45
4.3 *Exploratory Data Analisys* (EDA) 48
4.3.1 EDA 48
4.4 *Feature Engineering* 50
4.5 *Data Cleaning* 66
4.5.1 *Missing value* 67
4.5.2 Outliers 67
4.5.3 Dupilkat 67
4.6 Data *Outlier* 67
4.7 *Data Corelation* (Data Kolerasi) 69
4.8 Visualisasi Data 71
4.9 Membangun Model *Machine learning* 76
4.10 Pengujian Model Tanpa *Gridsearch* 77
4.11 Pengujian Model dengan *Gridsearch* 80
4.12 Evaluasi Dan Validasi Akhir 81
BAB V KESIMPULAN DAN SARAN 84
5.1 Kesimpulan 84
5.2 Saran 84
DAFTAR PUSTAKA 85
LAMPIRAN 89

DAFTAR GAMBAR

Gambar 2.1 SVM 16
Gambar 2.2 Kerangka Pemikiran 30
Gambar 3.1 Teknik Analisis 36
Gambar 4.1 *Outlier* sebelum ditangani 68
Gambar 4.2 *Outlier* sesudah ditangani 68
Gambar 4.3 Korelasi Data 70
Gambar 4.4 Distribusi Data Diagnosa 71
Gambar 4.5 Distribusi Data *Concave point_worst* 72
Gambar 4.6 Distribusi Data *Perimeter_worst* 72
Gambar 4.7 Distribusi Data *Concave point_mean* 73
Gambar 4.8 Distribusi Data *Radius worst* 73
Gambar 4.9 Distribusi Data *Perimeter mean* 74
Gambar 4.10 Distribusi Data *Area worst* 74
Gambar 4.11 Distribusi Data *Radius Mean* 75
Gambar 4.12 Distribusi Data *Area Mean* 75
Gambar 4.13 Distribusi Data *Concavity mean* 76
Gambar 4.14 Distribusi Data *Concavity worst* 76
Gambar 4.15 Skema SVM 77
Gambar 4.16 *Confusion matrix* tanpa *Gridsearch* 79
Gambar 4.17 Skema *Gridsearch* 80
Gambar 4.18 *Confusion matrix* data latih 81
Gambar 4.19 *Confusion matrix* Evaluasi data uji 82

DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka 8
Tabel 2.2 Tabel *Confusion matrix* 23
Tabel 3.1 Analisis kebutuhan 35
Tabel 3.2 Rincian Dataset 38
Tabel 4.1 Dataset 43
Tabel 4.2 Data teratas 44
Tabel 4.3 Isi dataset 46
Tabel 4.4 Deskripsi Kolom Dataset 46
Tabel 4.5 Data *Type*s 48
Tabel 4.6 Labeling 50
Tabel 4.7 *Labeling* 51
Tabel 4.8 *Feature Radius mean* 51
Tabel 4.9 *Feature Texture mean* 52
Tabel 4.10 *Feature Perimeter mean* 52
Tabel 4.11 *Feature Area mean* 53
Tabel 4.12 *Feature Smoothness mean* 53
Tabel 4.13 *Feature Compactness mean* 54
Tabel 4.14 *Feature Concavity mean* 54
Tabel 4.15 *Feature Concave point mean* 55
Tabel 4.16 *Feature Symmetry mean* 55
Tabel 4.17 *Feature Fractal dimension mean* 56
Tabel 4.18 *Feature Radius* se 56
Tabel 4.19 *Feature Texture* se 57
Tabel 4.20 *Feature Perimeter* se 57
Tabel 4.21 *Feature Area* se 58
Tabel 4.22 *Feature Smoothness* se 58
Tabel 4.23 *Feature Compactness* se 59
Tabel 4.24 *Feature Concavity* se 59
Tabel 4.25 *Feature Concave point* se 60
Tabel 4.26 *Feature Feature Symmetry* se 60
Tabel 4.27 *Feature Fractal dimension* se 61

Tabel 4.28 *Feature Radius worst* 61
Tabel 4.29 *Feature Texture worst* 62
Tabel 4.30 *Feature Perimeter worst* 62
Tabel 4.31 *Feature Area worst* 63
Tabel 4.32 *Feature Smoothness worst* 63
Tabel 4.33 *Feature Compactness worst* 64
Tabel 4.34 *Feature Concavity worst* 64
Tabel 4.35 *Feature Concave point worst* 65
Tabel 4.36 *Feature Symmetry worst* 65
Tabel 4.37 *Feature Fractal dimension worst* 66
Tabel 4.38 Penghapusan kolom 67
Tabel 4.39 Kolerasi data 69
Tabel 4.40 Split data 77
Tabel 4.41 Hasil tanpa *Gridsearch* 78
Tabel 4.42 Hasil uji 80
Tabel 4.43 Data uji 82
Tabel 4.44 Hasil Evaluasi 82

ABSTRACT

Breast cancer is one of the most common and deadly diseases among women. This study aims to improve the accuracy of breast cancer diagnosis using the Support Vector Machine (SVM) algorithm, optimized with the GridSearch technique. The dataset used is from the UCI Repository, focusing on classifying tumors as benign or malignant.

The methodology includes data cleaning, exploratory data analysis (EDA), and testing the model with and without GridSearch optimization. The results indicate that the application of GridSearch significantly enhances the SVM model's accuracy, achieving optimal performance in predicting tumors through evaluation metrics such as accuracy, precision, and recall. The optimized model demonstrates superior performance compared to the default SVM model.

This research contributes to early breast cancer diagnosis and shows that integrating machine learning techniques with hyperparameter optimization can yield more accurate predictive models.

***Keywords**: Breast cancer, Support Vector Machine (SVM), GridSearch, model optimization.*

ABSTRAK

Kanker payudara merupakan salah satu penyakit paling umum dan mematikan di kalangan wanita. Penelitian ini bertujuan untuk meningkatkan akurasi diagnosa kanker payudara menggunakan algoritma *Support Vector Machine* (SVM) yang dioptimalkan dengan teknik *GridSearch*. Dataset yang digunakan berasal dari UCI Repository, dengan fokus pada klasifikasi tumor menjadi jinak atau ganas.

Metodologi yang diterapkan melibatkan pembersihan data, *exploratory data analysis* (EDA), dan pengujian model dengan dan tanpa optimasi *GridSearch*. Hasil penelitian menunjukkan bahwa penerapan *GridSearch* meningkatkan akurasi model SVM secara signifikan, mencapai performa optimal dalam memprediksi tumor dengan metrik evaluasi seperti akurasi, *precision*, dan *recall*. Model yang dioptimalkan memberikan hasil evaluasi yang lebih baik dibandingkan dengan model SVM default.

Penelitian ini memberikan kontribusi terhadap peningkatan diagnosis dini kanker payudara dan menunjukkan bahwa penggabungan teknik *machine learning* dengan optimasi hiperparameter dapat menghasilkan model prediksi yang lebih akurat.

Kata kunci: Kanker payudara, *Support Vector Machine* (SVM), *GridSearch*, optimasi model.

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penelitian ini. Penelitian ini disusun sebagai salah satu syarat untuk menyelesaikan program studi strata satu (S1) pada program studi Teknik Informatika di Universitas Pamulang.

Penelitian ini bertujuan untuk meneliti apakah algoritma support vector machine dapat di optimasi lebih jauh dengan metode gridsearch, dan diharapkan dapat memberikan kontribusi yang berarti dalam penelitian mengenai bidang machine learning. Dalam proses penyusunan dan penyelesaian penelitian ini, penulis mendapatkan bantuan, dukungan, serta bimbingan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT, yang telah memberikan begitu banyak nikmat dan berkah serta kesehatan dan kelancaran hingga karunianya sehingga penulis dapat menyelesaikan penelitian ini.
2. Bapak Dr.Pranotom, selaku ketua Yayasan Sasmita Jaya yang telah memberikan tempat untuk mencari ilmu.
3. Bapak Dr. E. Nurzaman, AM., MM. Si, selaku Rektor Universitas Pamulang.
4. Bapak Dr. Ir. H. Sarwani. M.T., M.M, selaku Dekan Fakultas Ilmu Komputer Universitas Pamulang.
5. Bapak Dr. Eng. Ahmad Musyafa, S.kom, M.Kom, selaku Ketua Program Studi Teknik Informatika Universitas Pamulang.
6. Bapak Nurjaya, S.Kom, M.Kom, selaku Dosen Pembimbing yang sudah membimbing dan memberikan masukan serta kritik kepada penulis dalam penyusunan penelitian ini.
7. Kedua Orang Tua saya yang selalu memberikan support dan semangat serta rasa percaya kepada penulis sehingga dapat menyelesaikan penelitian ini.

8. Terima Kasih pada teman teman sejawat dan seperjuangan yang saya tidak dapat sebutkan satu persatu, peran kalian dalam memberikan motivasi sangat berpengaruh dalam penulisan penelitian ini.

Penulis menyadari bahwa penelitian ini masih memiliki kekurangan dan jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan guna perbaikan di masa yang akan datang. Semoga penelitian ini dapat bermanfaat bagi pembaca, khususnya dalam machine learning, serta menjadi referensi dalam pengembangan ilmu pengetahuan lebih lanjut.

Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang telah mendukung dan berperan serta dalam penyelesaian penelitian ini. Semoga Allah SWT senantiasa membalas segala kebaikan yang telah diberikan dengan pahala yang setimpal.

Tangerang Selatan, Oktober 2024

Chaisar Abi Prasetyo

BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Kanker payudara merupakan salah satu penyakit kanker yang paling umum dan salah satu penyebab utama kematian pada wanita, menurut data dari Organisasi Kesehatan Dunia (WHO), kanker payudara merupakan jenis kanker yang paling umum di kalangan wanita di seluruh dunia, dengan lebih dari 2 juta kasus baru yang terdiagnosis setiap tahunnya, angka kematian akibat kanker payudara juga masih tinggi, dengan sekitar 685.000 kematian yang dilaporkan pada tahun 2020 (American Cancer Society, 2022). Di Indonesia sendiri terdapat sebanyak 18.150 kasus dari 38 provinsi tertinggi di sebanyak 3.206 orang berasal dari Provinsi Jawa Tengah, kedua di Provinsi Jawa Timur sebanyak 3.077 orang, dan ketiga di Provinsi DI Yogyakarta sebanyak 1.985 orang dan di Provinsi Bengkulu sebanyak 44 orang penderita tumor payudara dan curiga kanker payudara sebanyak 13 orang. (Renita & Puspita Politeknik Kesehatan Kementerian Kesehatan Bengkulu, 2023).

Di sisi lain, pembelajaran dan kemajuan bidang mesin dan teknologi seperti teknik analisis data, pengenalan pola berkembang pesat dalam beberapa tahun terakhir, hal ini berdampak dalam kemajuan berbagai bidang termasuk medis, termasuk diagnosis kanker, beberapa metode yang sering digunakan dalam diagnosis penyakit kanker payudara adalah algoritma *support vector machine* (SVM), *random forest*, *decision tree*, *artificial neural networks* (ANN), *logistic regression*. (Islam et al., 2020)

Metode yang digunakan dalam penelitian ini adalah algoritma *support vector machine* (SVM), yang dikenal karena kemampuan dalam menangani data kompleks dan non-linier (Munawarah et al., 2022).

Metode *support vector machine* (SVM) memiliki kekurangan yaitu ketergantungannya pada pemilihan *Kernel* yang tepat dan parameter C (Septhya et al., 2023), dan kekurangan dari *random forest* adalah *random forest* dapat menjadi *computationally expensive* pada dataset yang besar dan tidak menghasilkan aturan

yang mudah diinterpretasikan secara langsung seperti pohon keputusan tunggal (Kabiraj et al., 2020). Kekurangan dari *decision tree* adalah dapat menjadi tidak stabil dan rentan terhadap *overfitting*, terutama pada dataset yang kompleks atau yang memiliki banyak fitur (Imaduddin et al., 2021). *Artificial Neural Network* memiliki kekurangan cenderung membutuhkan jumlah data yang besar untuk melatih model dengan baik dan rentan terhadap *overfitting* (Faris et al., 2020). Sedangkan *logistic regression* mempunyai kekurangan keterbatasan dalam menangani hubungan non-linier antara variabel prediktor dan variabel respons (Islam et al., 2020).

Penggunaan SVM sebagai pendekatan untuk mendeteksi awal penyakit kanker payudara menawarkan potensi yang luas. SVM dapat memanfaatkan informasi dari berbagai sumber termasuk data citra medis, *histopatologi* dan *citologi* untuk menghasilkan model yang akurat dalam membedakan antara sel jinak dan ganas, serta memprediksi kemungkinan kekambuhan penyaki (Imaduddin et al., 2021). Ditambah dengan *gridsearch* yang mengoptimasi *hyperparameter* yang tidak diperiksa oleh algoritma mesin akan sangat berpengaruh pada hasil akhir dari keakuratan model mesin (Schonlau, 2021).

Pada penelitian ini akan menerapkan *gridsearch* pada *metode support vector machine* untuk meningkatkan akurasi model mesin pembelajaran untuk diagnosa penyakit kanker payudara.

1.2 Identifikasi Masalah

Berdasarkan latar belakang diatas diketahui bahwa masalah yang dihadapi dalam penelitian ini adalah rendahnya tingkat akurasi metode *support vector machine* untuk diagnosa penyakit kanker payudara.

1.3 Rumusan Masalah

Berdasarkan identifikasi masalah yang telah diuraikan diatas, maka dapat dirumuskan permasalahannya sebagai berikut apakah penggunaan *gridsearch* pada algoritma *support vector machine* dapat meningkatkan akurasi model mesin untuk diagnosa penyakit kanker payudara?

1.4 Batasan Masalah

Agar pembahasan penelitian tidak terlalu melebar dari fokusnya, maka batasan masalah pada penelitian ini antara lain:

a. Pada penelitian ini hanya membahas penerapan metode *support vector machine* (SVM) dan *gridsearch* pada model diagnose sel kanker dan tidak membahas metode lainnya.
b. Pada penelitian ini hanya menggunakan metode *support vector machine* (SVM) pada model mesin dan tidak membahas implementasi aplikasinya.
c. Pada penelitian ini hanya menggunakan dataset dari *UCI Repository*, tidak membahas dan menggunakan dataset dari website lain.

1.5 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah menerapkan *gridsearch* meningkatkan akurasi model mesin pembelajaran *support vector machine* untuk diagnosa penyakit kanker payudara.

1.6 Manfaat Penelitian

Terdapat beberapa manfaat atau keuntungan yang diperoleh dari penelitian ini yaitu :

a. Manfaat Pribadi

   Menambah pemahaman dan wawasan peneliti yang ingin melakukan penelitian dan penelitian ini memberikan pengalaman praktis dalam mengolah data, membersihkan dan merancang skema algoritma *machine learning*
b. Manfaat untuk institusi

   Penelitian ini dapat mendorong inovasi dalam penggunaan teknologi, metode analisis data atau algoritma prediksi. Ini dapat mendorong universitas untuk mengadopsi teknologi terbaru dalam proses pembelajaran dan penelitian
c. Manfaat untuk umum

Sebagai acuan dalam membuat tugas akhir atau dalam kehidupan sehari haribaik pribadi atau tempat medis yang ingin membuat pendekteksi sel kanker dan penelitian ini juga dapat dikembangkan lagi oleh pembaca.

1.7 Metodologi Penelitian

Metodologi yang diterapkan pada penelitian ini melalui rangkaian tahapan seperti dibawah ini :

a. Observasi

   Tujuan dari observasi adalah pengumpulan data yang dilakukan dengan cara mengamati langsung objek yang diteliti. Dalam penelitian ini, observasi dilakukan untuk mendapatkan data yang akurat dan mendalam mengenai penerapan algoritma *support vector machine* (SVM) dalam deteksi dini kanker payudara. Dataset yang digunakan dapat dilihat pada *UCI Repository* https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic

b. Studi Pustaka

   Tujuan dari studi pustaka adalah pengumpulan data yang dilakukan dengan cara menelaah literatur atau sumber-sumber tertulis yang relevan dengan topik penelitian.

1.8 Sistematika Penulisan

Untuk lebih memudahkan dalam proses penyusunan tentang topik dasar penyuisunan tugas akhir dan memperjelas konten setiap bab, maka dibuat suatu sistematika penulisan sebagai berikut :

a. BAB I PENDAHULUAN

   Bab ini berisi tentang penjelasan terstruktur tentang dasar topic penelitian, mencakup komponen seperti latar belakang, identifikasi masalah, perumusan masalah, tujuan, manfaat dan batasan masalah. Dan bab ini juga membahas tentang metodologi penelitian yang digunakan.

b. BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bab ini mencakup tinjauan pustaka yang merinci studi studi terkait dan teori-teori yang mendukung penelitian. Dan berisi mengenai penjelasan dan teori tentang semua yang berkaitan dengan penelitian ini.

c. BAB III METODE PENELITIAN

Bab ini menjelaskan secara sistematis bagaimana proses penelitian dilakukan, berisi mengenai metode penelitian yang digunakan, analisa kebutuhan, teknik analisis dan perancangan model mesin.

d. BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini menjelaskan hasil pemrosesan data yang digunakan, melakukan pembuatan model mesin dan melakukan pengujian terhadap model mesin.

e. BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dari hasil penelitian yang sudah dilakukan, serta berisi saran untuk penelitian dimasa depan. Dan juga menjawab tujuan yang hendak dicapai.

BAB 2
TINJUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian ini dibuat tidak terlepas dari penelitian-penelitian terdahulu yang pernah dilakukan sebagai bahan kajian dan komparasi. Adapun hasil-hasil penelitian yang digunakan sebagai kajian dan acuan berkaitan erat dengan penelitian yaitu klasifikasi penyakit dengan *support vector machine* .

Berdasarkan penelitian yang pernah dilakukan oleh Hini Septhya, Kharisma Rahayu, Salsabila Rabbani, Vindi Fitria, Rahmaddeni, Yuda Irawan, Regiolina Hayami mengenai Implementasi dua algoritma yaitu *decision tree* dan *support vector machine* terhadap penyakit kanker paru yang berjudul "*Implementation of Decision tree Algorithm and Support vector machine for Lung Cancer Classification*". Kanker paru merupakan salah satu penyakit yang mematikan karena kanker ini sulit dideteksi sebelum berubah menjadi penyakit yang serius dan saat ini belum ada metode skrining yang efektif untuk deteksi dini kanker paru. Pada penelitian ini dilakukan teknik klasifikasi yang merupakan suatu metode pengelompokkan data yang memiliki karakter yang sama ke dalam beberapa kelompok. Teknik klasifikasi yang diteliti membandingkan 2 algoritma yaitu, algoritma *decision tree* dan *support vector machine* (SVM) untuk mengetahui algoritma yang memberikan hasil terbaik. Dalam penelitian ini akan dilakukan seleksi fitur menggunakan *forward selection* yang bertujuan untuk menaikkan nilai akurasi. Berdasarkan penelitian yang telah dilakukan dapatkan hasil dari algoritma SVM mempunyai nilai akurasi yang lebih unggul yaitu 62,3% menggunakan splitting data 80:20, dan 63.2% untuk algoritma *decision tree*.

Berdasarkan penelitian yang dilakukan oleh Hurriyati, Siti pada tahun 2023 yang berjudul "Implementasi Metode *support vector machine* pada klasifikasi diagnosis penyakit kanker payudara." *support vector machine* merupakan salah satu jenis *machine learning* yang banyak digunakan saat ini. Penerapan metode ini mencakup berbagai bidang, salah satunya bidang medis. Pada bidang medis, *support vector machine* diterapkan pada klasifikasi diagnosis suatu penyakit. Hasil

dari metode ini diharapkan dapat memudahkan pihak-pihak terkait dalam proses penanganan pasien sejak dini. Penelitian ini berfokus pada penyakit kanker payudara. Hasil penelitian menunjukkan bahwa model SVM terbaik ketika nilai variabel *Radius* dan *Perimeter* memiliki pengaruh yang besar pada diagnosis kanker payudara kategori *Benign,* sedangkan variabel *Texture* dan *Smoothness* memiliki pengaruh besar terhadap diagnosis kanker payudara kategori *Malignant*. Tingkat akurasi metode *Support vector machine* pada klasifikasi diagnosis kanker payudara sebesar 96,49%. Hal ini menunjukkan bahwa metode *Support vector machine* bekerja dengan baik pada diagnosis kanker payudara.

Berdasarkan penelitian yang pernah dilakukan oleh Muhammad Ravly Andryan, Muhamad Fajri, Nina Sulistyowati pada tahun 2022, mengenai kinerja *support vector machine* dan algoritma *XGboost* "Komparasi kinerja algoritma *XGboost* dan *Support vector machine* (SVM) untuk diagnosis penyakit kanker payudara". Kanker payudara dapat dideteksi pada tahap dini melalui tumor pada payudara, umumnya dapat terbagi menjadi dua yaitu *Benign* dan *Malignant*. Metode yang digunakan pada penelitian ini adalah *Knowledge Data Discovery* (KDD) dengan menggunakan algoritma *XGboost* dan SVM, kemudian dilakukan klasifikasi untuk menentukan apakah kanker yang dianalisa itu bernilai *Benign* atau *Malignant*. Data yang digunakan dalam penelitian ini adalah data publik yang dirilis oleh UCI *Machine learning* berjudul *Wisconsin Breast Cancer Diagnostic*. Hasil kinerja yang didapat setelah melakukan penelitian menggunakan kedua algoritma adalah *XGboost* yang memiliki akurasi terbaik sebesar 95.12% dan nilai ROC_AUC sebesar 0.99 dan algoritma SVM memiliki akurasi terendah sebesar 90.24% dan nilai ROC_AUC sebesar 0.98.

Berdasarkan penelitian yang pernah dilakukan oleh Lusa Indah Prahartiwi, Wulan Dari pada tahun 2021, mengenai "Komparasi Algoritma *Naive Bayes*, *Decision tree* dan *Support vector machine* untuk Prediksi Penyakit Kanker Payudara". Penyakit kanker payudara dapat diprediksi dengan pengetahuan *Data Mining*. *Data Mining* dapat menemukan korelasi, pola, dan tren baru yang bermakna dengan memilah-milah data dalam jumlah besar yang disimpan dalam repositori, menggunakan teknologi pengenalan pola serta teknik statistik dan matematika. Penelitian ini membandingkan performa Algoritma *Naive Bayes*,

Decision tree dan *Support vector machine* untuk memprediksi penyakit kanker payudara. Dataset yang digunakan adalah data sekunder *Breast Cancer Coimbra* yang diambil dari *UCI Repository*. Hasil dari penelitian ini menunjukan bahwa Algoritma *Support vector machine* menghasilkan tingkat *Accuracy* tertinggi yaitu sebesar 74,29% dibandingkan dengan Algoritma *Naive Bayes* sebesar 72.60% dan *Decision tree* dengan akurasi sebesar 71.25%.

Berdasarkan penelitian yang pernah dilakukan oleh Sajib Kabiraj; M. Raihan, Nasif Alvi, Marina Afrin, Laboni Akter, Shawmi Akhter Sohagi, Etu Podder pada tahun 2020 mengenai teknik pembelajaran mendalam dan pembelajaran mesin untuk pengukuran resiko medis berbasis kanker payudara yang berjudul "*Breast Cancer Risk Prediction using XGboost and Random forest Algorithm*". Kanker payudara adalah salah satu penyebab utama kematian pada wanita di seluruh dunia. Penyakit ini terjadi ketika sel-sel dalam payudara tumbuh secara tidak terkendali. Faktor resiko kanker payudara meliputi riwayat keluarga dengan kanker, kurangnya aktivitas fisik, stres psikologis, dan peningkatan ukuran payudara. Dalam penelitian ini, dataset kanker payudara dianalisis untuk memprediksi keberadaan kanker payudara menggunakan dua algoritma ensemble *Machine learning* yang populer, yaitu *Random forest* dan *Extreme Gradient Boosting* (*XGBoost*). Penelitian ini menggunakan 275 instance dengan 12 fitur. Hasilnya, algoritma *Random forest* mencapai akurasi sebesar 74,73%, sementara *XGboost* mencapai akurasi 73,63%.

Tabel 2.1 Tinjauan Pustaka

Peneliti	Tahun	Topik Penelitian	Metode	Hasil
Hini Septhya, Kharisma Rahayu, Salsabila Rabbani, Vindi Fitria, Rahmaddeni,	2023	*Implementation of Decision tree Algorithm and Support vector machine for Lung Cancer Classification*	*Forward selection* dengan algoritma *Decision tree* dan SVM	SVM akurasi 62.3% , Decision tree, Akurasi 63.2%

Yuda Irawan, Regiolina Hayami				
Hurriyati, Siti	2023	Implementasi metode *Support vector machine* pada klasifikasi diagnosis penyakit kanker payudara	*Support vector machine* (SVM)	SVM: akurasi 96.49%;
Muhammad Ravly Andryan, Muhamad Fajri, Nina Sulistyowati	2022	Komparasi kinerja algoritma *XGboost* dan *Support vector machine* (SVM) untuk diagnosis penyakit kanker payudara	Knowledge Data Discovery (KDD) dengan algoritma *XGboost* dan SVM	*XGBoost*: akurasi 95.12%, ROC_AUC 0.99; SVM: akurasi 90.24%, ROC_AUC 0.98

Lusa Indah Prahartiwi, Wulan Dari	2021	Komparasi Algoritma *Naive Bayes*, *Decision tree* dan *Support vector machine* untuk Prediksi Penyakit Kanker Payudara	Pengetahuan *Data Mining*dengan algoritma *Naive Bayes*, *Decision tree*, dan SVM	SVM dengan akurasi tertinggi 74.29%, *Naive Bayes* sebesar 72.60%, *Decision tree* sebesar 71.25%.
Sajib Kabiraj, M. Raihan, Nasif Alvi, Marina Afrin, Laboni Akter, Shawmi Akhter Sohagi, Etu Podder	2020	*Breast Cancer Risk Prediction using XGboost and Random forest Algorithm*	Penggunaan *XGboost* dan *Random forest* algoritma untuk prediksi resiko kanker payudara.	*Random forest Accuracy* 74.73%, *XGboost Accuracy* 73.76%

2.2 Landasan Teori

2.2.1 Data Mining

Data Mining adalah proses ekstraksi pengetahuan yang bermanfaat atau pola yang menarik dari dataset besar atau kompleks (Tsui et al., 2023). Tujuannya adalah untuk menemukan pola tersembunyi yang dapat digunakan untuk pengambilan keputusan. *Data Mining* melibatkan sejumlah teknik dan metode analisis data yang digunakan untuk mengungkapkan pola-pola yang mungkin tidak terdeteksi secara langsung oleh manusia.

Data Mining dibagi menjadi beberapa kelompok utama berdasarkan tujuan analisis dan metode yang digunakan (Wu et al., 2021). Berikut adalah beberapa kelompok utama dalam *Data Mining*:

a. Klasifikasi

   Memisahkan data ke dalam kelas atau kategori berdasarkan atribut-atribut tertentu. Metodenya melibatkan penggunaan algoritma klasifikasi seperti *naive bayes*, *decision tree*s, dan *support vector machine* (SVM) untuk membangun model yang dapat memprediksi kelas atau label dari data baru.

b. *Clustering*

   Mengelompokkan data menjadi kelompok-kelompok yang serupa berdasarkan pola-pola yang ada dalam data. Metodenya meliputi penggunaan algoritma *Clustering* seperti k-*Mean*s, Hierarchical *Clustering*, dan DBSCAN untuk mengidentifikasi struktur kelompok dalam data.

c. Regresi

   Memprediksi nilai dari variabel dependen berdasarkan variabel independen. Metodenya melibatkan penggunaan teknik regresi seperti Regresi Linier, Regresi Logistik, dan Regresi Ridge untuk memodelkan hubungan antara variabel-variabel dalam data.

d. Asosiasi

   Menemukan hubungan atau korelasi antara item-item dalam dataset. Metodenya melibatkan penggunaan algoritma asosiasi seperti Apriori dan *FP-Growth* untuk menemukan aturan asosiasi antara item-item dalam transaksi atau dataset.

e. Pola Sequential dan *Time Series*

   Mengidentifikasi pola atau tren dalam data yang disusun berdasarkan urutan waktu. Metodenya meliputi penggunaan teknik seperti *Sequential Pattern Mining* dan *Forecasting* untuk menganalisis pola-pola dalam data deret waktu.

f. Pemrosesan Grafik

Menganalisis hubungan dan struktur dalam data yang direpresentasikan sebagai grafik. Metodenya melibatkan penggunaan algoritma pemrosesan grafik seperti *PageRank, HITS*, dan Algoritma Klastering untuk menganalisis jaringan sosial, jaringan Web, atau jaringan lainnya.

Menurut tahapan proses melakukan *Data Mining* meliputi:

a. Seleksi Data

Tahap ini melibatkan pemilihan data yang relevan dan sesuai dengan tujuan analisis. Praktisi *Data Mining* perlu memilih dataset yang memiliki potensi untuk mengungkap pola atau informasi yang berharga terkait dengan masalah yang ingin diselesaikan.

b. Pemilihan Data

Setelah data yang relevan dipilih, tahap selanjutnya adalah pemilihan variabel atau atribut yang akan digunakan dalam analisis. Hal ini melibatkan identifikasi variabel yang paling berpengaruh atau memiliki hubungan yang kuat dengan variabel target atau tujuan analisis.

c. Transformasi

Tahap transformasi data dilakukan untuk mempersiapkan data yang dipilih untuk proses analisis lebih lanjut. Ini meliputi normalisasi data, pengurangan dimensi, atau konversi format data agar sesuai dengan persyaratan algoritma *Data Mining* yang akan digunakan.

d. Mining Data

Tahap ini merupakan inti dari proses *Data Mining*, dimana algoritma dan teknik *Data Mining* diterapkan pada dataset yang telah dipersiapkan sebelumnya. Praktisi *Data Mining* menggunakan berbagai metode seperti klasifikasi, *Clustering*, atau regresi untuk mengidentifikasi pola atau hubungan yang signifikan dalam data.

e. Evaluasi

Setelah model *Data Mining* dibangun, tahap evaluasi dilakukan untuk mengevaluasi kinerja model. Praktisi *Data Mining* menggunakan metrik evaluasi yang sesuai seperti akurasi, presisi, *Recall*, atau *Area* di bawah kurva ROC untuk mengukur seberapa baik model dapat memprediksi atau menemukan pola dalam data.

2.2.2 *Machine Learning*

Machine learning (ML) adalah suatu cabang dari kecerdasan buatan yang memungkinkan sistem komputer untuk belajar dari data dan pengalaman tanpa harus secara eksplisit diprogram (Sarker, 2021). Dengan menggunakan berbagai algoritma dan teknik, sistem komputer dapat mengidentifikasi pola dalam data, membuat prediksi, dan mengambil keputusan tanpa intervensi manusia langsung. Hal ini memungkinkan mesin untuk terus berkembang dan meningkatkan kinerjanya seiring berjalannya waktu, tanpa perlu pembaruan atau perubahan kode secara manual. Dengan demikian, *Machine learning* memiliki aplikasi yang luas dalam berbagai bidang, termasuk pengenalan pola, analisis data, prediksi, dan pengambilan keputusan (Hooshmand & Maserat, 2024).

Terdapat dua kategori berdasarkan jenis dan jumlah supervisi selama pelatihan, yaitu *Supervised Learning* dan *Unsupervised Learning*(Jo, 2021).

a. *Supervised Learning*

Pada *Supervised Learning*, model *Machine learning* diberikan data yang sudah dilabeli, yang berarti setiap contoh data telah diberi label atau kelas yang sesuai. Tujuan utama dari *Supervised Learning* adalah untuk menghasilkan fungsi yang dapat memetakan input ke output yang tepat. Terdapat dua jenis utama *Supervised Learning*:

1. Klasifikasi (*Classification*)

   Dalam klasifikasi, output yang diprediksi adalah kelas atau label diskrit. Model mempelajari hubungan antara input dan kelas output dari contoh-contoh yang sudah dilabeli.

2. Regresi (*Regression*)

   Dalam regresi, output yang diprediksi adalah nilai continou. Model mempelajari hubungan antara input dan nilai output dari contoh-contoh yang sudah dilabeli. Contoh aplikasi regresi termasuk prediksi harga rumah berdasarkan fitur-fitur tertentu, prediksi jumlah penjualan berdasarkan faktor-faktor pasar, dan sebagainya.

b. *Unsupervised Learning*

   Pada *unsupervised Learning*, model *Machine learning* diberikan data yang tidak dilabeli. Tujuan utama dari *Unsupervised Learning* adalah untuk menemukan pola atau struktur dalam data yang tidak diketahui sebelumnya. Terdapat beberapa jenis utama un*supervised Learning*:

   1. *Clustering*

      Dalam *Clustering*, data dikelompokkan ke dalam kelompok-kelompok yang serupa berdasarkan pola-pola yang ada dalam data. Algoritma *Clustering* seperti *K-Means, Hierarchical Clustering*, dan DBSCAN sering digunakan untuk tugas ini.

   2. Reduksi Dimensi (*Dimensionality Reduction*)

      Dalam reduksi dimensi, jumlah atribut atau fitur dalam data dikurangi dengan cara memilih subset fitur yang paling informatif atau dengan proyeksi data ke ruang dimensi yang lebih rendah. Contoh teknik reduksi dimensi termasuk Principal Component Analysis (PCA) dan t-SNE.

   3. Asosiasi (*Association*)

      Dalam asosiasi, hubungan antara item-item dalam dataset ditemukan. Contoh aplikasi asosiasi termasuk rekomendasi produk dalam e-commerce berdasarkan sejarah pembelian pengguna.

   4. Anomali Detection (*Anomaly Detection*)

      Unsupervised Learning juga digunakan untuk mendeteksi anomali atau pencilan dalam data, yaitu data yang tidak mengikuti pola umum dari mayoritas data.

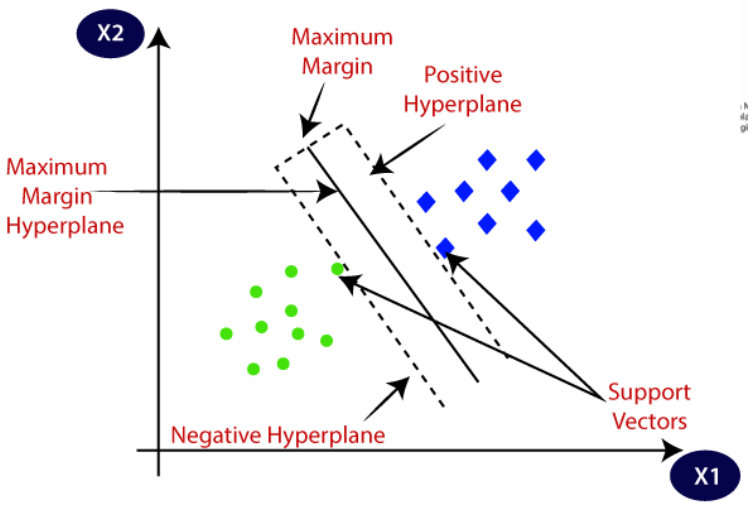
2.2.3 *Support Vector Machine*

Support vector machine (SVM) diperkenalkan oleh Vapnik, Boser dan Guyon pada tahun 1992. SVM adalah salah satu teknik baru dibandingkan dengan teknik lain, tetapi memiliki performansi yang lebih baik di berbagai bidang aplikasi seperti bioinformatika, pengenalan tulisan tangan, klasifikasi teks, klasifikasi diagnosis penyakit dan lain sebagainya (Hurriyati, 2023).

Bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *Hyperplane* terbaik yang memisahkan dua buah *Class* pada input *space*, *Hyperplane* terbaik adalah *Hyperplane* yang terletak ditengah-tengah antara dua set obyek dari dua *Class*, *Hyperplane* pemisah terbaik antara kedua *Class* dapat ditemukan dengan mengukur margin *Hyperplane* tersebut dan mencari titik maksimalnya (Imaduddin et al., 2021). Margin adalah jarak antara *Hyperplane* tersebut dengan *pattern* terdekat dari masing-masing *Class*. *Pattern* yang paling dekat ini disebut sebagai *Support Vector* (Islam et al., 2020). Karakteristik dari *Support vector machine* adalah sebagai berikut:

a. *Support vector machine* adalah algoritma mesin linier *Classifier*.
b. *Pattern Recognition* dilakukan dengan mentransformasikan data pada input *space* ke ruang yang berdimensi lebih tinggi, dan optimisasi dilakukan pada ruang vector yang baru tersebut. Hal ini membedakan SVM dari solusi *pattern Recognition* pada umumnya, yang melakukan optimisasi parameter pada ruang hasil transformasi yang berdimensi lebih rendah daripada dimensi input *space*.
c. Menerapkan strategi *Structural Risk Minimization* (SRM).
d. Prinsip kerja *Support vector machine* pada dasarnya hanya mampu menangani klasifikasi dua *Class*.

Secara sederhana konsep SVM adalah sebagai usaha mencari *Hyperplane* terbaik yang berfungsi sebagai pemisah dua buah pemisah *Class* pada input *space*, dimana dapat dilihat pada gambar dibawah ini:



Gambar 2.1 SVM

Gambar diatas adalah ilustrasi dari prinsip kerja *Support vector machine* (SVM), sebuah algoritma pembelajaran mesin yang digunakan untuk klasifikasi. Berikut adalah penjelasan rinci tentang komponen-komponen yang ada dalam gambar tersebut:

a. *Hyperplane*:
   1. *Positive Hyperplane*: *Hyperplane* yang berada di dekat kelas positif (ditandai dengan titik biru).
   2. *Negative Hyperplane*: *Hyperplane* yang berada di dekat kelas negatif (ditandai dengan titik hijau).
   3. *Maximum* Margin *Hyperplane*: *Hyperplane* yang memisahkan kedua kelas dengan margin terbesar. Ini disebut juga sebagai "*decision boundary*" atau "optimal separating *Hyperplane*".

b. *Support Vectors*:

   Titik-titik yang berada paling dekat dengan *Maximum* Margin *Hyperplane* dari masing-masing kelas disebut *Support Vectors*. Mereka adalah titik data yang paling berpengaruh dalam menentukan posisi *Hyperplane* optimal. Dalam gambar, *Support Vectors* ditandai dengan panah yang menunjukkan mereka berada pada margin dari *Hyperplane*.

1. *Maximum* Margin: Jarak terdekat antara *Hyperplane* optimal dan titik data terdekat dari setiap kelas. SVM bertujuan untuk memaksimalkan margin ini untuk meningkatkan generalisasi model.

c. Dimensi Ruang (X1 dan X2):

Gambar menunjukkan dua dimensi fitur X1 dan X2. SVM dapat bekerja dalam ruang fitur berdimensi tinggi, dan teknik *Kernel* dapat digunakan untuk memetakan data ke ruang berdimensi lebih tinggi jika data tidak dapat dipisahkan secara linear.

2.2.3.1 Tahapan SVM

Tahapan dan langkah dalam perhitungan algoritma *Support vector machine* meliputi :

a. Langkah pertama dalam penerapan *Support vector machine* (SVM) adalah pemilihan dataset yang akan dianalisis. Misalnya, dalam penelitian ini, kita mempertimbangkan dataset sederhana yang terdiri dari dua kelas, yaitu *Class* A dan *Class* B, dengan dua fitur, X1 dan X2. Contoh data adalah sebagai berikut:
   1. *Class* A (Label +1):
      Titik data: (2, 3), (3, 3), (4, 2)
   2. *Class* B (Label -1):
      Titik data: (1, 1), (2, 1), (2, 2)

b. Berikutnya adalah memvisualisasikan data untuk memahami distribusi dan separabilitas antara dua kelas. Pada grafik dua dimensi, titik-titik dari *Class* A dan *Class* B diplot berdasarkan nilai fitur X1 dan X2. Visualisasi ini membantu dalam mengidentifikasi pola awal dan menentukan kebutuhan untuk transformasi fitur jika diperlukan.

c. Lalu menemukan *Hyperplane* yang dapat memisahkan kedua kelas dengan margin maksimum. *Hyperplane* dalam ruang dua dimensi dinyatakan sebagai garis dengan persamaan berikut :

$$w \cdot x - b = 0$$

Di mana:

w adalah vektor bobot.

x adalah vektor fitur (data input).

b adalah bias atau *intercept*.

$$w1X1 + w2X2 + b = 0$$

Di mana *w1X1* dan *w2X2* adalah bobot yang mengatur kemiringan garis, dan *b* adalah bias yang mengatur posisi garis tersebut terhadap sumbu.

d. Margin didefinisikan sebagai jarak antara *Hyperplane* dengan titik data terdekat dari kedua kelas. SVM bertujuan untuk memaksimalkan margin ini agar pemisahan antar kelas lebih jelas. Kondisi ini dapat diformulasikan sebagai berikut:

1. Untuk data dari *Class* A (Label +1):

$$w1X1 + w2X2 + b \geq 1$$

2. Untuk data dari *Class* B (Label -1):

$$w1X1 + w2X2 + b \leq -1$$

Untuk kelas yang berbeda, persamaan *Hyperplane* harus memenuhi:

Untuk kelas Positif : $w \cdot xi - b \geq 1$

Untuk kelas Negatif : $w \cdot xi - b \leq 1$

Margin (M) antara dua *Hyperplane* dapat dihitung sebagai : $M = \frac{2}{|W|}$

Di mana $|W|$ adalah norm Euclidean dari vektor bobot W.

e. Lalu SVM memformulasikan masalah ini sebagai masalah optimasi yang bertujuan untuk meminimalkan fungsi objektif berikut:

1. Minimalkan $\frac{1}{2}||W||2$

Dengan syarat bahwa untuk setiap titik data harus memenuhi:

$$yi(w1X1 + w2X2 + b) \geq 1$$

2. Di sini, *yi* adalah label dari titik data, di mana *yi = +1yi = +1yi*=+1 untuk *Class* A dan *yi*=−*1yi = -1yi*= −1 untuk *Class* B.

Tujuan utama dari SVM adalah memaksimalkan margin M, yang secara ekuivalen dapat ditulis sebagai minimisasi $|W|$, dengan syarat bahwa semua titik data dipisahkan dengan benar. Secara matematis, ini diformulasikan sebagai masalah optimasi : $\min w, b\ \frac{1}{2}|w|2$

Dengan kendala : $y1(w \cdot x1 - b) \geq 1$

Di mana y1 adalah label kelas (+1 atau -1) untuk titik data x1.

f. Selanjutnya Fungsi optimasi tersebut diselesaikan menggunakan metode optimasi tertentu, seperti *Quadratic Programming*, untuk mendapatkan nilai optimal dari bobot *w1,w2* dan bias bobot. Nilai-nilai ini kemudian digunakan untuk menentukan posisi dan orientasi *Hyperplane* yang optimal.
Misalnya, jika setelah proses optimasi ditemukan bahwa:
W1 = 1
W2 = -1
B = 1

Maka persamaan *Hyperplane* menjadi:

$$X1-X2+1 = 0$$

g. Lalu penentuan *Support Vectors* adalah titik data yang berada tepat di tepi margin, yang secara kritis menentukan posisi *Hyperplane*. Dalam kasus ini, misalnya:
   1. Dari *Class* A: Titik (2, 3)
   2. Dari *Class* B: Titik (2, 2)

   Support Vectors ini adalah titik-titik yang paling dekat dengan *Hyperplane* dan memiliki peran penting dalam menentukan bentuk *Hyperplane*.
h. Maka Setelah *Hyperplane* ditentukan, data baru dapat diklasifikasikan berdasarkan posisinya relatif terhadap *Hyperplane*. Misalnya, untuk titik data baru dengan koordinat (3, 2) :

$$\int(x) - 3 - 2 + 1 = 2$$

Karna f(x) > 0, data tersebut diklasifikasikan sebagai *Class* A (Label +1).

2.2.4 *Gridsearch*

Gridsearch adalah salah satu teknik *Hyperparameter* tuning yang bertujuan untuk mencari kombinasi terbaik dari beberapa *Hyperparameter* yang digunakan dalam algoritma pembelajaran mesin(G. & Brindha, 2022). Dalam konteks penelitian ini, *Gridsearch* digunakan untuk menemukan kombinasi optimal dari *Hyperparameter* dalam model *Support vector machine* (SVM), proses *Gridsearch* secara sistematis mencoba semua kemungkinan kombinasi dari *Hyperparameter* yang telah ditentukan, dan memilih yang memberikan kinerja terbaik berdasarkan metrik evaluasi tertentu, misalnya akurasi, *Precision*, *Recall*, atau nilai *F1* (Andryan et al., 2022).

2.2.4.1 Tahapan *Gridseacrh*

Tahapan dan langkah dalam perhitungan *gridsearch* meliputi :

a. Langkah pertama dalam *gridsearch* adalah menentukan parameter-parameter yang akan diuji. Dalam penelitian ini, parameter yang diuji pada model SVM adalah:
   1. C (regularisasi)
   2. *gamma* (parameter *Kernel* RBF)
   3. *Kernel* (jenis *Kernel* yang digunakan)
b. Pembagian Data
   Data dibagi menjadi beberapa subset menggunakan teknik *cross-validation*. Pada penelitian ini, digunakan k-fold *cross-validation* dengan k=5. Data latih dibagi menjadi 5 subset, dan iterasi dilakukan sebanyak 5 kali. Pada setiap iterasi, satu subset digunakan sebagai data uji, sementara empat subset lainnya digunakan untuk melatih model. Hal ini memastikan bahwa semua data diuji dan model tidak overfitting.
c. Evaluasi kinerja dengan Cross validation

gridsearch mencoba semua kombinasi dari *hyperparameter* yang telah ditentukan. Untuk setiap kombinasi, model dilatih menggunakan data latih, dan performa diuji menggunakan subset validasi. Pada setiap iterasi, kinerja model diukur menggunakan metrik yang dipilih, misalnya akurasi atau nilai *F1*. Proses ini diulangi untuk setiap kombinasi, dan hasil evaluasi dicatat.

d. Pemilihan model terbaik
   Setelah semua kombinasi diuji, *gridsearch* memilih kombinasi *hyperparameter* yang memberikan performa terbaik pada validasi *cross-validation*. Sebagai contoh, jika kombinasi "C=10, gamma=0.01, dan *Kernel*='rbf'" memberikan akurasi tertinggi, maka kombinasi tersebut dipilih sebagai model final.

2.2.4.2 Skema Perhitungan *Gridsearch*

Pada dataset ini, *gridsearch* menghasilkan kombinasi *hyperparameter* berikut:

a. C = 10
b. gamma = 0.01
c. *Kernel* = 'rbf'

Jumlah kombinasi *hyperparameter* yang diuji dapat dihitung sebagai berikut:

$$Total\ Kombinasi = (Jumlah\ C)x(Jumlah\ Gamma)x(Jumlah\ kernel)$$

Dengan nilai yang digunakan dalam penelitian ini:

$$Total\ Kombinasi = 4x4x2 = 32$$

Setiap kombinasi diuji menggunakan 5-fold *cross-validation*, sehingga total iterasi yang dilakukan adalah:

$$Total\ iterasi = 32\ x\ 5 = 160\ iterasi$$

Pada setiap iterasi, model akan dilatih dan diuji, dan metrik evaluasi dicatat. Berdasarkan hasil dari evaluasi *cross-validation*, kombinasi *hyperparameter* terbaik dipilih, yaitu C = 10, gamma = 0.01, dan *Kernel* = 'rbf'.

2.2.4.3 Hasil Tuning *Hyperparameter*

Setelah menggunakan *gridsearch*, model yang dihasilkan memiliki performa optimal dibandingkan model dengan *hyperparameter* default. Kinerja model yang optimal ini selanjutnya digunakan untuk prediksi dan evaluasi akhir pada dataset pengujian.

2.2.5 Pengujian Model

Proses klasifikasi merupakan keluaran dari penerapan algoritma SVM. Setelah data dianalisis, SVM akan mengklasifikasikan data ke dalam dua kategori: *Malignant* dan *Benign*. Hasil klasifikasi ini menjadi dasar untuk tahap evaluasi model.

2.2.5.1 *Confusion matrix*

Confusion matrix adalah alat yang digunakan dalam *Machine learning* dan statistik untuk mengukur kinerja model klasifikasi. Ini adalah tabel yang memungkinkan kita untuk melihat seberapa baik model klasifikasi kita dalam memprediksi hasil yang benar. Berikut adalah penjelasan rinci dan tabel yang terkait dengan *Confusion matrix*.

Confusion matrix terdiri dari empat komponen utama:

a. *True Positive* (TP): Jumlah data yang benar-benar positif dan diprediksi positif oleh model.
b. *True Negative* (TN): Jumlah data yang benar-benar negatif dan diprediksi negatif oleh model.
c. *False Positive* (FP): Jumlah data yang benar-benar negatif tetapi diprediksi positif oleh model (*Type* I *error*).
d. *False Negative* (FN): Jumlah data yang benar-benar positif tetapi diprediksi negatif oleh model (*Type* II *error*).

Tabel 2.2 Tabel *Confusion matrix*

Actual / Predicted	***Positive* (Pred)**	***Negative* (Pred)**
Positive (*Actual*)	*True Positive* (TP)	*False Negative* (FN)
Negative (*Actual*)	*False Positive* (FP)	*True Negative* (TN)

2.2.5.2 *Accuracy*

Accuracy adalah metrik yang mengukur proporsi prediksi yang benar dari keseluruhan prediksi yang dibuat oleh model. Ini dihitung dengan membagi jumlah prediksi benar (baik positif maupun negatif) dengan total prediksi.

Rumus dan perhitungan *Accuracy* :

TP (*True Positives*) : 40

TN (*True Negatives*) : 45

FP (*False Positives*) : 5

FN (*False Negatives*) : 10

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

$$Accuracy = \frac{40 + 45}{40 + 45 + 5 + 10} = \frac{85}{100} = 0.85$$

Dengan demikian hasil *Accuracy* model adalah 85%.

2.2.5.3 *Precision*

Precision adalah salah satu metrik evaluasi yang digunakan untuk mengukur kinerja model SVM. *Precision* menghitung proporsi prediksi positif yang benar dari keseluruhan prediksi positif yang dibuat oleh model.

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

$$Precision = \frac{40}{40 + 5} = \frac{40}{45} = 0.89$$

Dengan demikian *Precision* dari model adalah 89%.

2.2.5.4 *Recall*

Recall merupakan metrik evaluasi lainnya yang mengukur kemampuan model dalam menemukan semua data positif yang ada.

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

$$Recall = \frac{40}{40 + 10} = \frac{40}{50} = 0.80$$

Dengan demikian *Recall* dari model adalah 80%.

2.2.5.5 *F1-Score*

F1-Score adalah metrik evaluasi yang mengkombinasikan *Precision* dan *Recall* menjadi satu nilai tunggal. *F1-Score* dihitung sebagai rata-rata harmonik dari *Precision* dan *Recall*.

$$F1 - Score = 2\frac{Precision\ x\ Recall}{Precision + Recall}$$

$$F1 - Score = 2\ x\ \frac{0.89\ x\ 0.80}{0.89 + 0.80} = 2\ x \frac{0.712}{1.69} = 0.843$$

Dengan demikian *F1-Score* model adalah 84.3%.

2.2.6 *Python*

Python adalah bahasa pemrograman tingkat tinggi yang sangat populer dan serbaguna. *Python* telah menjadi pilihan utama dalam berbagai bidang, termasuk analisis medis, pengembangan model medis, mengenalan pola penyakit, dan manajemen risiko (Klemp, 2024).

Menurut (Linge & Langtangen, 2020) karakteristik *Python* yang menonjol disemua bidang termasuk medis seperti :

a. Kemudahan Penggunaan

Python memiliki sintaksis yang mudah dipahami dan ramah pengguna, sehingga memungkinkan para profesional medis untuk dengan cepat mempelajari dan menggunakan bahasa ini dalam pengembangan solusi medis.

b. Ekosistem Pustaka yang Kaya

Python memiliki ekosistem pustaka yang kaya, termasuk pustaka-pustaka seperti *NumPy*, *Pandas*, *Matplotlib*, dan *Scikit-learn*, yang sangat berguna dalam analisis data medis, pemrosesan citra medis, dan pengembangan model *Machine learning* untuk diagnosis dan prediksi penyakit.

c. Fleksibilitas

Python adalah bahasa pemrograman yang sangat fleksibel, yang memungkinkan para peneliti dan praktisi medis untuk mengembangkan berbagai jenis aplikasi, mulai dari aplikasi desktop hingga aplikasi web dan mobile, sesuai dengan kebutuhan mereka.

d. Interpretatif dan Mudah Diperluas

Python adalah bahasa pemrograman interpretatif yang memungkinkan para pengguna untuk menjalankan kode *Python* secara langsung tanpa proses kompilasi yang rumit. Selain itu, *Python* juga mudah diperluas dengan modul-modul tambahan dan pustaka-pustaka pihak ketiga, sehingga memperluas kemampuan bahasa ini untuk menangani berbagai jenis tugas medis.

e. *Open Source* dan Komunitas yang Aktif

Python adalah perangkat lunak *open source* dengan komunitas pengembang yang besar dan aktif. Ini berarti bahwa terdapat banyak sumber daya, tutorial, dan dukungan yang tersedia bagi para praktisi medis untuk mempelajari dan menggunakan *Python* dalam pengembangan solusi medis.

f. Pemrosesan Citra Medis

Python memiliki pustaka-pustaka seperti *OpenCV* dan *scikit-image* yang kuat untuk pemrosesan citra medis, yang memungkinkan para peneliti medis untuk

melakukan segmentasi, deteksi, dan analisis objek dalam citra medis dengan mudah.

2.2.7 *Library Python*

2.2.7.1 *Scikit-learn*

Scikit-learn adalah pustaka *Python* yang menyediakan berbagai algoritma *Machine learning* dan alat untuk memproses dan menganalisis data (Nelli, 2023b). Pustaka ini mengimplementasikan algoritma *Machine learning* seperti klasifikasi, regresi, klastering, dan reduksi dimensi. Selain itu, *Scikit-learn* juga menyediakan alat untuk pemrosesan data, pemodelan statistik, dan evaluasi model. Keunggulan *Scikit-learn* terletak pada kemudahannya digunakan dan dipelajari, dukungan yang luas untuk berbagai jenis algoritma *Machine learning*, serta dokumentasi yang kaya dan beragam tutorial (*Test*as, 2023).

Selain itu, *Scikit-learn* juga menyediakan fitur untuk validasi model, cross-validation, dan tuning *Hyperparameter* menggunakan *Gridsearch*CV dan *RandomizedSearchCV*, implementasi *Scikit-learn* mencakup algoritma populer seperti *Support vector machine* (SVM), Random forest, K-Nearest Neighbors (KNN), dan banyak lagi. Keunggulan *Scikit-learn* terletak pada antarmuka yang seragam, dokumentasi yang lengkap, dan komunitas yang aktif. Pengguna dapat dengan mudah menerapkan model-model Machine learning tanpa harus memahami detail implementasi algoritma yang rumit (Testas, 2023).

2.2.7.2 *NumPy*

NumPy adalah pustaka *Python* untuk komputasi numerik yang menyediakan struktur data array multidimensi dan fungsi-fungsi matematika yang kuat (Gupta & Bagchi, 2024). Fungsinya mencakup manipulasi data numerik seperti array dan matriks, komputasi matematika, aljabar linear, dan integrasi yang baik dengan pustaka lain seperti *Pandas* dan *Scikit-learn*, keunggulan *NumPy* terletak pada kinerja tinggi untuk operasi array besar, fungsi-fungsi matematika yang kuat dan efisien, serta kompatibilitas yang baik dengan pustaka-pustaka ilmiah lainnya (Gupta & Bagchi, 2024).

Satu lagi kekuatan utama *NumPy* adalah kompatibilitasnya dengan pustaka-pustaka ilmiah lainnya seperti *SciPy*, *Pandas*, dan *TensorFlow*, serta integrasinya dengan C dan *Fortran*, yang memungkinkan pengguna untuk mempercepat perhitungan numerik dengan menggunakan kode asli, selain itu, *NumPy* mendukung operasi slicing dan indexing pada array, yang sangat berguna dalam manipulasi data yang kompleks (Häberlein, 2024). Kecepatan dan efisiensi *NumPy* membuatnya menjadi komponen inti dalam sebagian besar proyek komputasi ilmiah dan pembelajaran mesin.

2.2.7.3 *Pandas*

Pandas adalah pustaka *Python* yang menyediakan struktur data tingkat tinggi, seperti *DataFrame*, untuk analisis data tabular (Hetland & Nelli, 2024). Fungsi utamanya mencakup pemrosesan data tabular, pembersihan data, manipulasi data, penggabungan data dari berbagai sumber, serta analisis eksploratif dan pemodelan data. Keunggulan *Pandas* terletak pada fungsionalitas yang kuat untuk bekerja dengan data tabular, kemudahan penggunaan, dan dukungan yang baik untuk operasi berbasis label (Hunt, 2023).

Keunggulan *Pandas* terletak pada kemampuannya dalam menangani data yang tidak terstruktur, termasuk menangani missing values dengan metode imputation atau penghapusan (Hunt, 2023). Selain itu, *Pandas* mendukung operasi berbasis time series, yang sangat berguna untuk analisis data berbasis waktu seperti analisis tren atau forecasting, pustaka ini juga mendukung integrasi dengan pustaka visualisasi seperti *Matplotlib* dan *Seaborn* untuk pembuatan grafik yang lebih kompleks (Gupta & Bagchi, 2024). *Pandas* menjadi alat yang sangat esensial dalam eksplorasi data, pembersihan data, dan preprocessing dalam pipeline Machine learning

2.2.7.4 *Seaborn*

Seaborn adalah pustaka *Python* yang membangun di atas *Matplotlib* dan menyediakan antarmuka tingkat tinggi untuk membuat plot statistik yang menarik dan informatif (Hetland & Nelli, 2024). Fungsi utamanya meliputi pembuatan plot kompleks seperti heatmap, pairplot, dan violin plot, serta visualisasi data statistik

dan distribusi. Keunggulan *Seaborn* terletak pada gaya default yang menarik dan informatif, kemudahan penggunaan untuk pembuatan plot statistik, dan dukungan untuk visualisasi data kompleks.

Salah satu keunggulan utama *Seaborn* adalah kemampuannya untuk bekerja langsung dengan *Pandas DataFrames*, sehingga mempermudah integrasi antara manipulasi data dan visualisasi. Selain itu, *Seaborn* menyediakan default styling yang menarik, memungkinkan visualisasi data yang lebih bersih dan informatif tanpa perlu banyak kustomisasi tambahan (Bose et al., 2024). Pustaka ini mendukung berbagai visualisasi distribusi yang membantu pengguna dalam memahami outliers, tren, dan korelasi antar variabel dalam dataset, kombinasi *Seaborn* dengan *Matplotlib* sering digunakan untuk menghasilkan visualisasi yang lebih informatif dan detail dalam laporan data ilmiah atau presentasi analisis(Nelli, 2023).

2.2.7.5 *Matplotlib*

Matplotlib adalah pustaka *Python* untuk visualisasi data yang menyediakan berbagai jenis plot dan grafik (Bose et al., 2024). Fungsi utamanya termasuk pembuatan plot 2D dan 3D, histogram, grafik garis, dan visualisasi data untuk analisis eksploratif dan presentasi. Keunggulan *Matplotlib* terletak pada fleksibilitasnya dalam menciptakan berbagai jenis plot, dukungan untuk berbagai format output dan media, serta integrasi yang baik dengan lingkungan pengembangan *Python* (Bose et al., 2024).

Matplotlib sangat fleksibel dan memungkinkan pengguna untuk kustomisasi penuh terhadap setiap aspek plot, termasuk warna, label, ukuran font, dan judul. Pustaka ini juga mendukung subplots, yang memungkinkan pembuatan beberapa grafik dalam satu gambar (Gupta & Bagchi, 2024). Salah satu kekuatan *Matplotlib* adalah kemampuannya untuk digabungkan dengan pustaka lain seperti *Seaborn* untuk membuat visualisasi yang lebih kompleks dan menarik. *Matplotlib* sangat penting untuk exploratory data analysis (EDA), di mana visualisasi sering kali digunakan untuk menemukan pola, tren, dan hubungan antar variable (Hetland & Nelli, 2024). Selain itu, *Matplotlib* juga digunakan dalam lingkungan

pengembangan interaktif seperti *Jupyter Notebook*, di mana grafik dapat dihasilkan secara langsung dan diubah secara real-time.

2.2.8 Aplikasi Pendukung

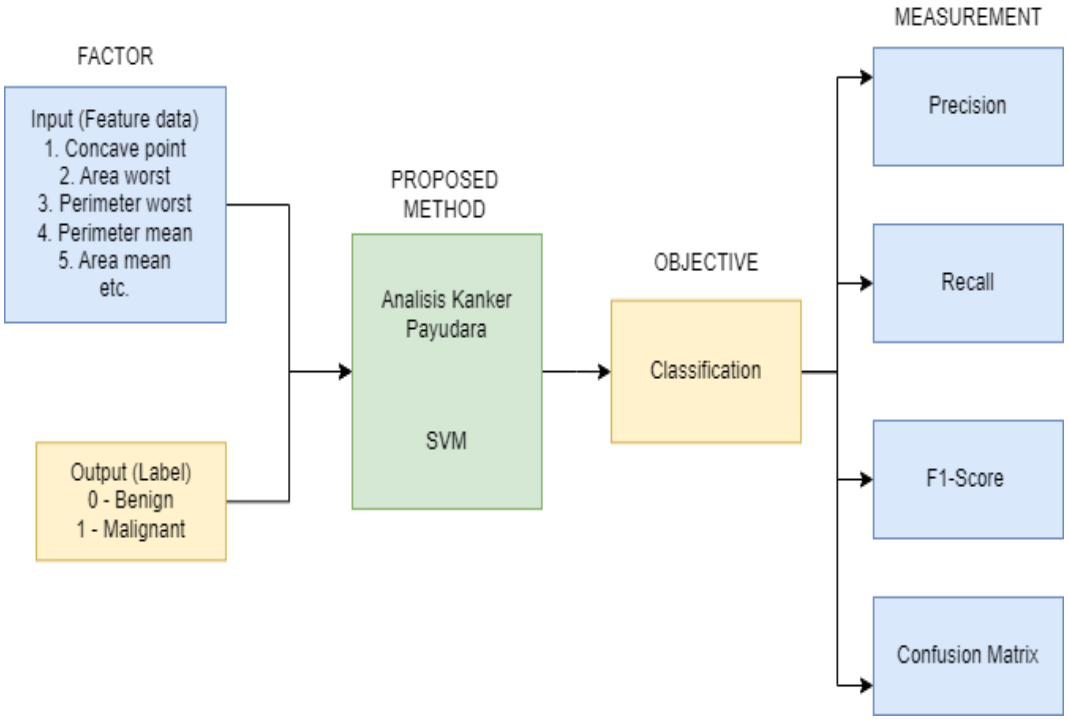
2.2.8.1 *Jupyter Notebook*

Jupyter Notebook adalah sebuah aplikasi *open-source* yang digunakan secara luas dalam dunia penelitian dan pengembangan, khususnya di bidang data science dan *Machine learning* (Silaparasetty, 2020).

Jupyter Notebook memungkinkan pengguna untuk menulis dan menjalankan kode interaktif, visualisasi data, serta menyusun dokumen berbasis teks dalam satu lingkungan terpadu. Fitur ini sangat bermanfaat bagi peneliti karena mempermudah proses eksplorasi data, eksperimen, dan dokumentasi hasil secara real-time. Selain itu, Jupyter Notebook mendukung berbagai bahasa pemrograman seperti Python, R, dan Julia, sehingga menjadi alat yang fleksibel dan mendukung kolaborasi antarpeneliti dari berbagai disiplin ilmu. Integrasi dengan pustaka visualisasi seperti Matplotlib dan Seaborn juga menjadikan Jupyter Notebook sebagai pilihan utama dalam analisis dan presentasi data yang mudah dipahami (Silaparasetty, 2020).

2.2.9 Kerangka Pemikiran

Kerangka pemikiran pada penelitian ini dirancang untuk menguraikan tahapan-tahapan yang dilalui dalam menganalisis data kanker payudara menggunakan algoritma *Support vector machine* (SVM).



Gambar 2.2 Kerangka Pemikiran

2.3 Tinjauan Objek

2.3.1 Kanker Payudara

Kanker payudara adalah jenis kanker yang terbentuk di jaringan payudara, terutama di saluran susu dan kelenjar susu, ini merupakan salah satu penyakit kanker yang paling umum di kalangan wanita di seluruh dunia, kanker payudara dapat berkembang dari berbagai jenis sel dalam payudara, dan perjalanan penyakitnya dapat berbeda-beda dari satu individu ke individu lainnya (Renita & Puspita Politeknik Kesehatan Kementerian Kesehatan Bengkulu, 2023). Faktor resiko untuk kanker payudara meliputi faktor genetik, usia, riwayat keluarga, paparan hormonal, gaya hidup, dan faktor lingkungan. (*American Cancer Society*, 2022).

2.3.2 Diagnosa Penyakit Kanker Payudara

Sistem Diagnosa penyakit kanker adalah proses untuk mendeteksi apakah seseorang mempunyai sel kanker yang bersifat *Malignant* atau *Benign*, sesuai dengan informasi dari data data yang sudah diberikan.

Pada penelitian ini sistem diagnose penyakit kanker payudara menggunakan beberapa input yaitu :

a. *ID*

   Berisi nomor identifikasi unik untuk setiap sampel pasien yang terdapat dalam dataset. Nomor ini digunakan sebagai penanda untuk membedakan sampel satu dengan yang lainnya.

b. *Diagnosis*

   Berisi hasil diagnosis dari tumor, di mana 'M' (*Malignant*) menunjukkan tumor ganas, dan 'B' (*Benign*) menunjukkan tumor jinak. Kolom ini menjadi target klasifikasi dalam analisis menggunakan algoritma pembelajaran mesin.

c. *Radius_mean*

   Merupakan nilai rata-rata dari jarak antara pusat masa sel dengan tepi terluar sel (*Radius*). Nilai ini mencerminkan ukuran rata-rata sel.

d. *Texture_mean*

   Mengukur variasi dalam intensitas pixel di sekitar *Perimeter* sel. Nilai ini digunakan untuk menganalisis tekstur permukaan sel, yang dapat membantu membedakan sel normal dari sel kanker.

e. *Perimeter_mean*

   Mengukur panjang tepi terluar sel. Kolom ini memberikan informasi mengenai lingkaran sel secara rata-rata dan berperan penting dalam pengukuran morfologi sel.

f. *Area_mean*

   Menunjukkan luas *Area* dari sel berdasarkan citra yang dihasilkan. *Area* ini diukur dalam satuan pixel dan memberikan indikasi ukuran fisik dari sel.

g. *Smoothness_mean*

Mengukur seberapa halus permukaan sel. Hal ini dihitung sebagai perubahan lokal dalam panjang *Perimeter* sel. Sel kanker cenderung memiliki permukaan yang kurang halus dibandingkan sel normal.

h. *Compactness_mean*

Mengukur kepadatan sel berdasarkan rumus. Nilai ini menunjukkan seberapa padat sel dibandingkan dengan luasnya, yang dapat digunakan untuk mendeteksi tumor padat.

i. *Concavity_mean*

Menunjukkan tingkat cekungan atau konkavitas di sepanjang batas sel. Nilai yang lebih tinggi mencerminkan adanya cekungan lebih banyak pada permukaan sel, yang merupakan ciri khas dari sel kanker ganas.

j. *Concave points_mean*

Mengukur jumlah titik cekung yang terdapat pada *Perimeter* sel. Titik cekung ini biasanya lebih banyak ditemukan pada sel-sel kanker dibandingkan dengan sel normal.

k. Symmetry_*mean*

Mengukur simetri sel. Sel kanker cenderung lebih asimetris, dan kolom ini memberikan informasi mengenai tingkat keasimetrian sel.

l. *Fractal_dimension_mean*

Mengukur kompleksitas *Perimeter* sel. Ini adalah pengukuran dimensi fraktal yang menunjukkan tingkat kekasaran pada batas sel, yang dapat memberikan indikasi tentang sifat sel.

m. *Radius*_se

Mengukur kesalahan standar (standard error) untuk *Radius* sel, yang menunjukkan seberapa bervariasi ukuran *Radius* dalam setiap sampel.

n. *Texture*_se

Kesalahan standar untuk variasi tekstur permukaan sel. Nilai ini menunjukkan variasi dalam intensitas pixel di sekitar *Perimeter*.

o. *Perimeter*_se

Kesalahan standar untuk pengukuran *Perimeter* sel. Ini memberikan indikasi tentang ketepatan pengukuran *Perimeter* di berbagai titik dalam dataset.

p. *Area*_se

Kesalahan standar untuk luas *Area* sel. Menunjukkan seberapa besar variasi dalam pengukuran *Area* sel dari satu sampel ke sampel lainnya.

q. *Smoothness*_se

Kesalahan standar untuk tingkat kelancaran permukaan sel. Nilai ini menunjukkan variasi kelancaran di berbagai titik sel.

r. *Compactness*_se

Kesalahan standar untuk pengukuran kepadatan sel, menunjukkan variasi tingkat kepadatan antar sel dalam sampel.

s. *Concavity*_se

Kesalahan standar untuk tingkat cekungan pada sel, memberikan gambaran tentang seberapa variatif cekungan di sepanjang *Perimeter* sel.

t. *Concave points*_se

Kesalahan standar untuk jumlah titik cekung pada *Perimeter* sel. Ini menunjukkan variabilitas dalam jumlah titik cekung pada sel kanker.

u. Symmetry_se

Kesalahan standar untuk simetri sel, menunjukkan seberapa variatif tingkat simetri antar sel.

v. *Fractal_dimension*_se

Kesalahan standar untuk dimensi fraktal dari *Perimeter* sel. Nilai ini memberikan gambaran tentang variabilitas dalam tingkat kekasaran *Perimeter* sel.

w. *Radius_worst*

Mengukur *Radius* terbesar yang ditemukan di antara semua sel yang dianalisis pada citra. Ini menunjukkan ukuran maksimal dari *Radius* sel yang dapat membantu dalam identifikasi tumor besar.

x. *Texture_worst*

Mengukur variasi tekstur terbesar di antara semua sel yang dianalisis. Nilai ini penting dalam menentukan perbedaan antara jaringan normal dan jaringan yang terkena kanker.

y. *Perimeter_worst*

Mengukur *Perimeter* terbesar di antara semua sel yang dianalisis. Ini menggambarkan ukuran *Perimeter* terluar terbesar dari tumor, yang digunakan untuk mengidentifikasi massa tumor besar.

z. *Area_worst*

Menunjukkan luas *Area* terbesar di antara semua sel yang dianalisis. Luas terbesar ini dapat menjadi indikator penting untuk mendeteksi tumor yang besar dan berpotensi lebih berbahaya.

BAB 3
METODE PENELITIAN

3.1 Analisa Kebutuhan

Analisa kebutuhan dilakukan untuk memahami kebutuhan pengguna dan system yang berkaitan dengan pengembangan implementasi algoritma *Support vector machine* untuk diagnosa penyakit Kanker payudara dengan menggunakan bahasa pemrograman *Python*.

Pada penelitian ini, spesifikasi perangkat keras dan lunak yang digunakan adalah:

Tabel 3.1 Analisis kebutuhan

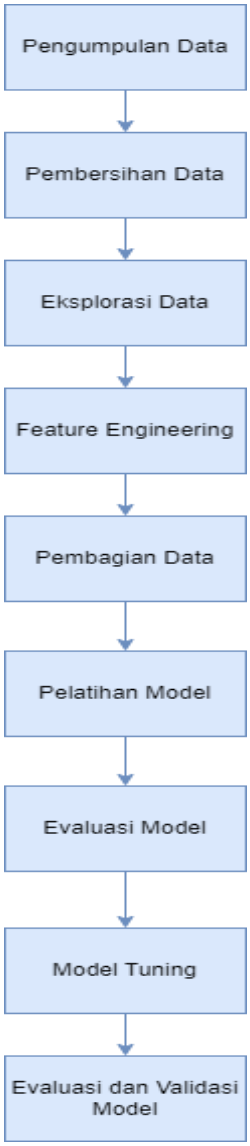
Perangkat Keras	Spesifikasi
Processor	Core I3 gen 9 atau setara
SSD	Minimum 128 Gb
RAM	Minimum 8 Gb
Perangkat Lunak	Spesifikasi
Sistem Operasi	Windows 10
Aplikasi Simulator	Google Colab
Bahasa Pemrograman	Pyhton 3.x
Library *Python*	*Scikit-learn* *Pandas* *Matplotlib* *NumPy* *Seaborn*

3.2 Teknik Analisis

3.2.1 Perancangan Penelitian

Rancangan penelitian merupakan rencana atau kerangka kerja yang akan digunakan untuk mengatur dan mengarahkan langkah langkah yang akan daiambil

dalam penelitian. Rancangan penelitian ini terdiri dari langkah langkah sesuat dengan gambar berikut :



Gambar 3.1 Rancangan Penelitian

a. Pengumpulan Data

Tahap pengumpulan data merupakan langkah awal yang krusial dalam proses penelitian. Data yang digunakan dapat berasal dari berbagai sumber, seperti basis data internal, file eksternal (misalnya CSV), API, atau repositori data publik seperti UCI *Machine learning* Repository.

b. Pembersihan Data

Setelah data terkumpul, dilakukan proses pembersihan data. Tahap ini bertujuan untuk mengidentifikasi dan memperbaiki data yang tidak konsisten, hilang, atau mengandung anomali yang dapat mengganggu analisis. Proses pembersihan meliputi penghapusan data duplikat, penanganan nilai hilang (misalnya, dengan imputasi menggunakan rata-rata atau *median*), koreksi kesalahan data, serta penghapusan *Outlier* jika diperlukan.

c. Eksplorasi Data

Eksplorasi data dilakukan untuk memahami karakteristik data secara lebih mendalam. Pada tahap ini, analisis statistik deskriptif diterapkan untuk memperoleh informasi tentang distribusi, tendensi sentral, dan penyebaran data. Visualisasi data, seperti histogram, scatter plot, dan heatmap, digunakan untuk mengidentifikasi pola dan hubungan antar variabel dalam dataset.

d. Feature Engineering

Feature Engineering merupakan tahap di mana fitur-fitur baru dibentuk atau fitur-fitur yang ada dimodifikasi untuk meningkatkan kinerja model *Machine learning*. Aktivitas dalam tahap ini meliputi penskalaan fitur, pengkodean variabel kategoris, pembuatan fitur baru dari kombinasi fitur yang ada, serta seleksi fitur untuk memilih fitur yang paling signifikan bagi model.

e. Pembagian Data / *Data Splitting*

Setelah fitur-fitur siap digunakan, data kemudian dibagi menjadi dua subset: data pelatihan dan data pengujian. Pembagian ini dilakukan untuk memastikan bahwa model dilatih dan dievaluasi dengan data yang berbeda. Proporsi untuk pembagian ini adalah 70% untuk data pelatihan dan 30% untuk data pengujian. Dengan pembagian ini, model dapat diuji kemampuannya dalam memprediksi hasil pada data yang belum pernah dilihat sebelumnya.

Tabel 3.2 Rincian Dataset

Dataset	Jumlah Baris	Jumlah Kolom
Data Latih (X_*train*)	398	30 *Feature*
Label Latih (Y_*train*) (output)	398	1 (*Malignant*/*Benign*)
Data Uji (X_*test*)	171	30 *Feature*
Label Uji (Y_*test*) (output)	171	1 (*Malignant*/*Benign*)

f. Model *Train*ing

Pada tahap pelatihan model, algoritma *Machine learning* diterapkan pada data pelatihan untuk mempelajari pola yang ada dalam data. Algoritma ini dapat berupa regresi linear, *Decision tree*, *Random forest*, atau jaringan saraf tiruan (neural networks), tergantung pada jenis masalah yang dihadapi. Hasil dari proses pelatihan ini adalah model yang dapat digunakan untuk membuat prediksi atau klasifikasi berdasarkan data input yang diberikan.

g. Model Evaluation

Evaluasi model dilakukan untuk menilai kinerja model menggunakan data pengujian. Metrik evaluasi seperti akurasi, *Precision*, *Recall, F1-Score*, dan AUC-ROC digunakan untuk mengukur seberapa baik model dalam melakukan prediksi. Hasil dari tahap ini memberikan gambaran tentang kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya, yang merupakan indikator penting dari kualitas model.

h. Model *Tuning*

Tahap tuning model dilakukan jika diperlukan untuk meningkatkan kinerja model lebih lanjut. Dalam tahap ini, parameter-parameter model disesuaikan melalui teknik seperti *Gridsearch* untuk menemukan kombinasi parameter yang optimal. Setelah tuning selesai, model dilatih ulang menggunakan parameter terbaik dan dievaluasi kembali untuk memastikan peningkatan kinerja yang dicapai.

i. Evaluasi dan validasi

Setelah model dilatih dan di-tuning, tahap evaluasi dan validasi dilakukan untuk memastikan bahwa model mampu bekerja dengan baik pada data yang belum pernah dilihat sebelumnya. Pada tahap ini, metrik evaluasi seperti akurasi, precision, recall, F1-score, dan AUC-ROC digunakan untuk menilai performa model secara keseluruhan. Selain itu, teknik validasi seperti *cross-validation* juga diterapkan untuk memastikan bahwa model tidak overfitting dan dapat menggeneralisasi dengan baik terhadap data di luar dataset pengujian. Evaluasi ini memberikan gambaran seberapa baik model mampu memprediksi hasil pada data baru.

3.3 Dataset

Dataset yang digunakan pada penelitian ini merupakan dataset Kanker Payudara Wisconsin yang terdapat pada UCI Repository, berikut merupakan rincian dataset yang digunakan dalam penelitian ini :

Judul Dataset	**Breast Cancer Wisconsin (Diagnostic)**
Sumber	UCI Machine Learning Repository (https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic)
Deskripsi	Dataset *Breast Cancer Wisconsin (Diagnostic)* digunakan untuk klasifikasi tumor payudara apakah bersifat ganas atau jinak. Dataset ini berisi 569 data sampel dengan 32 fitur yang menggambarkan karakteristik inti sel yang ada dalam gambar hasil aspirasi jarum halus (Fine Needle Aspiration/FNA) dari massa payudara. Fitur-fitur ini dihitung dari gambar digital aspirasi jarum halus massa payudara dan menggambarkan karakteristik inti sel dalam gambar tersebut.
Jumlah Sampel	569
Jumlah Fitur	32 (termasuk ID, diagnosis (M = ganas, B = jinak), dan 30 fitur numerik).

Informasi Fitur	- Nomor ID
	- Diagnosis (M = ganas, B = jinak)
	- 30 fitur numerik dihitung untuk setiap inti sel:
	- Radius (rata-rata jarak dari pusat ke titik di perimeter)
	- Tekstur (simpangan baku dari nilai skala abu-abu)
	- Perimeter
	- Luas
	- Kelancaran (variasi lokal dalam panjang radius)
	- Kekompakan (perimeter² / luas - 1.0)
	- Konkavitas (keparahan bagian cekung dari kontur)
	- Titik cekung (jumlah bagian cekung dari kontur)
	- Simetri
	- Dimensi fraktal (pendekatan "garis pantai" - 1)
Distribusi Kelas	- Jinak: 357
	- Ganas: 212
Penggunaan	Dataset ini sering digunakan dalam penelitian di bidang diagnostik medis, khususnya dalam pengembangan dan evaluasi algoritma klasifikasi untuk diagnosis kanker payudara.
Sitasi Utama	Wolberg, W.H., & Mangasarian, O.L. (1990). *Metode pemisahan pola multisurface untuk diagnosis medis yang diterapkan pada sitologi payudara*. *Proceedings of the National Academy of Sciences*, 87, 9193-9196.
Penelitian Lain yang Menggunakan Dataset	1. Street, W. N., Wolberg, W. H., & Mangasarian, O. L. (1993). *Nuclear feature extraction for breast tumor diagnosis*. *Biomedical Image Processing and Biomedical Visualization*. Dataset ini digunakan untuk mengekstrak fitur nuklir dari gambar payudara dan mengembangkan metode diagnosis otomatis berbasis machine learning.

	2. Wolberg, W. H., Street, W. N., & Mangasarian, O. L. (1995). *Machine learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates*. *Cancer Letters*. Studi ini berfokus pada pengembangan teknik machine learning untuk mendiagnosis kanker payudara menggunakan fitur nuklir yang diambil dari aspirasi jarum halus.
	3. Bache, K., & Lichman, M. (2013). *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences. Dataset ini diakui secara luas dan digunakan dalam riset akademis.
	4. Abdallah, A., Maarof, M. A., & Zainal, A. (2016). *Fraud detection system: A survey*. *Journal of Network and Computer Applications*. Meskipun topiknya terkait dengan deteksi fraud, dataset ini digunakan sebagai contoh implementasi teknik machine learning.
	5. Bennett, K. P., & Mangasarian, O. L. (1992). *Robust linear programming discrimination of two linearly inseparable sets*. *Optimization Methods and Software*. Studi ini menggunakan dataset ini untuk mengembangkan metode optimasi pemisahan pola.
	6. Chaurasia, V., & Pal, S. (2014). *Data Mining Techniques: To Predict and Resolve Breast Cancer Survivability*. *International Journal of Computer Science and Mobile Computing*. Dataset ini digunakan untuk mengevaluasi berbagai teknik data mining dalam memprediksi kelangsungan hidup pasien kanker.

Bukti Validitas	1. Penggunaan Luas dalam Studi Akademis: Dataset ini telah digunakan dalam lebih dari 700 penelitian, menunjukkan penerimaan luas di bidang machine learning medis.
	2. Kepatuhan pada Standar Etika dan Privasi: Data anonim, tidak mengandung informasi pribadi pasien, sehingga memenuhi standar etika penelitian.
	3. Validasi Algoritma Machine Learning: Dataset ini digunakan untuk memvalidasi berbagai algoritma seperti SVM, Decision Trees, Neural Networks.

BAB 4
HASIL DAN PEMBAHASAN

4.1 Hasil

Bab ini membahas proses implementasi dan analisis yang dilakukan dalam penelitian ini. Tujuan utama dari penelitian ini adalah untuk mengevaluasi performa algoritma *Support vector machine* (SVM) dalam mendeteksi kanker payudara berdasarkan dataset yang diperoleh. Proses ini melibatkan berbagai tahapan, mulai dari pemrosesan awal data, pemilihan fitur, hingga evaluasi model.

4.1.1 Persiapan data

Dataset yang digunakan dalam penelitian ini adalah *Breast Cancer Wisconsin (Diagnostic) Dataset* yang tersedia dalam format CSV pada website UCI https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic Dataset ini terdiri dari 569 *Record*, 33 column *Feature* dan 1 *Class*.

Tabel 4.1 Dataset

Nama	*Record*	*Class*
Breast Cancer Wisconsin	569	1

4.1.2 Implementasi Metode

Pada penelitian ini, digunakan metode Support vector machine (SVM) untuk klasifikasi diagnosis kanker payudara berdasarkan dataset *Breast Cancer Wisconsin Diagnostic*. Data ini terdiri dari berbagai fitur karakteristik sel kanker, dengan label target berupa **0** (benign) dan **1** (malignant). Pada implementasi ini, lima sampel data teratas digunakan untuk menjelaskan perhitungan SVM secara manual.

Tabel 4.2 Data teratas

Index	Mean Radius	Mean Texture	Diagnosis (Label)
1	17.99	10.38	1 (Malignant)
2	20.57	17.77	1 (Malignant)
3	19.69	21.25	1 (Malignant)
4	11.42	20.38	0 (Benign)
5	20.29	14.34	1 (Malignant)

a. Pembentukan Fungsi Keputusan

Metode SVM bertujuan untuk menemukan hyperplane yang memisahkan dua kelas diagnosis, yaitu *benign* (0) dan *malignant* (1). Fungsi keputusan SVM dapat dinyatakan sebagai:

$$\int(x) = w\,x + b$$

Dimana:

W adalah vektor bobot yang menentukan arah *hyperplane*,

x adalah vektor fitur dari masing-masing data (*mean radius* dan *mean texture*),

b adalah bias.

b. Fungsi Optimasi

Tujuan dari SVM adalah untuk memaksimalkan margin, yaitu jarak antara *hyperplane* dengan titik-titik terdekat dari setiap kelas (*benign* dan *malignant*). Fungsi optimasi yang digunakan adalah :

$$min\ \frac{1}{2}|w|2$$

Dengan syarat :

$$y1(w \cdot x1 - b) \geq 1$$

c. Perhitungan Margin

Nilai margin dapat dihitung sebagai :

$$margin = \frac{2}{|W|}$$

d. Penentuan Support Vectors

Support vectors adalah titik-titik yang terletak pada atau sangat dekat dengan margin. Dalam kasus ini, berdasarkan perhitungan manual, *support vectors* yang terpilih adalah :

1. Data ke-4 = $x4 = [11.42, 20.38]$ dengan label 0 (*Benign*).
2. Data ke-1 = $x1 = [17.99, 10{,}38]$ dengan label 1 (*Malignant*)

e. Perhitungan Prediksi

Setelah vektor bobot w dan bias b diperoleh dari proses training, prediksi dapat dilakukan untuk data baru dengan menghitung fungsi keputusan f(x). Jika f(x) > 0 maka kelas yang diprediksi adalah 1 (*malignant*), dan Jika f(x) < 0 maka diprediksi adalah *benign*. Misalkan hasil perhitungan bobot dan bias memberikan nilai :

$w = \begin{bmatrix} 0.5 \\ -0.4 \end{bmatrix}$, b = -3.0

Untuk sampel baru X(baru)=[18.5, 12.3] prediksi dilakukan dengan cara berikut:

$$\text{f(Xbaru)} = \text{wTXbaru} + \text{b} = (0.5 \times 18.5) + (-0.4 \times 12.3) - 3.0$$

$$\text{f(Xbaru)} = 9.25 - 4.92 - 3.0 = 1.33$$

Karena nilai f(Xbaru) > 0 maka sampel ini diprediksi sebagai **1** (*malignant*).

4.2 *Data Collection*

Pada tahap ini akan dilakukan pemruraian dataset yang telah disiapkan, rincian data dapat dilihat pada tabel dibawah :

Tabel 4.3 Isi dataset

id	***diagnosis***	***Perimeter_mean***	***Area_mean***	**…**	*Compactness_mean*	*Compactness_worst*
842302	M	122,8	1001	…	0,2776	0,6656
842517	M	132,9	1326	…	0,07864	0,1866
84300903	M	130	1203	…	0,1599	0,4245
84348301	M	77,58	386,1	…	0,2839	0,8663
84358402	M	135,1	1297	…	0,1328	0,205
843786	M	82,57	477,1	…	0,17	0,5249
844359	M	119,6	1040	…	0,109	0,2576
84458202	M	90,2	577,9	…	0,1645	0,3682
844981	M	87,5	519,8	…	0,1932	0,5401
84501001	M	83,97	475,9	…	0,2396	1,058
845636	M	102,7	797,8	…	0,06669	0,1551

Dan untuk keterangan dari masing-masing atribut yang ada pada dataset diatas dapat dilihat pada table dibawah ini :

Tabel 4.4 Deskripsi Kolom Dataset

Atribut	**Deskripsi**
ID	Nomor identifikasi untuk setiap sampel.
Diagnosis	Diagnosis jaringan payudara: M (*Malignant*/kanker) atau B (*Benign*/tidak kanker).
Radius_mean	Rata-rata jarak dari pusat ke titik-titik pada *Perimeter*.
Texture_mean	Deviasi standar nilai skala abu-abu.
Perimeter_mean	Rata-rata panjang *Perimeter* tumor.
Area_mean	Luas rata-rata tumor.

Smoothness_mean	Rata-rata variasi lokal dalam panjang *Radius*.
Compactness_mean	Tingkat kekompakan, dihitung sebagai (*Perimeter*^2 / *Area* - 1.0).
Concavity_mean	Rata-rata cekungan tumor (ke dalam atau depresi).
Concave_points_mean	Rata-rata jumlah titik cekungan pada permukaan tumor.
Symmetry_mean	Rata-rata simetri tumor.
Fractal_dimension_mean	Rata-rata dimensi fraktal, mengukur kompleksitas kontur (*Perimeter* / *Area*).
Radius_se	Kesalahan standar dari rata-rata *Radius*.
Texture_se	Kesalahan standar dari rata-rata tekstur.
Perimeter_se	Kesalahan standar dari rata-rata *Perimeter*.
Area_se	Kesalahan standar dari rata-rata *Area*.
Smoothness_se	Kesalahan standar dari rata-rata *Smoothness*.
Compactness_se	Kesalahan standar dari rata-rata *Compactness*.
Concavity_se	Kesalahan standar dari rata-rata *Concavity*.
Concave_points_se	Kesalahan standar dari rata-rata *Concave points*.
Symmetry_se	Kesalahan standar dari rata-rata *Symmetry*.
Fractal_dimension_se	Kesalahan standar dari rata-rata *Fractal dimension*.
Radius_worst	Nilai terburuk atau terbesar dari *Radius*.
Texture_worst	Nilai terburuk atau terbesar dari tekstur.
Perimeter_worst	Nilai terburuk atau terbesar dari *Perimeter*.

Area_worst	Nilai terburuk atau terbesar dari *Area*.
Smoothness_worst	Nilai terburuk atau terbesar dari *Smoothness*.
Compactness_worst	Nilai terburuk atau terbesar dari *Compactness*.
Concavity_worst	Nilai terburuk atau terbesar dari *Concavity*.
Concave_points_worst	Nilai terburuk atau terbesar dari *Concave points*.
Symmetry_worst	Nilai terburuk atau terbesar dari *Symmetry*.
Fractal_dimension_worst	Nilai terburuk atau terbesar dari *Fractal dimension*.
Unammed 32	NaN

4.3 *Exploratory Data Analisys* (EDA)

Pada tahap ini dataset diperiksa untuk menentukan jumlah baris dan kolom yang nantinya digunakan dalam penelitian.

4.3.1 EDA

Pada tahap ini dataset akan dibedah untuk melihat tipe data, mencari kolerasi, memahami strukturnya dan melihat statistic deskriptif seperti nilai *mean*, nilai *median*, nilai *modus* dan standar deviasi untuk mengindentifikasi atribut numeric dan kategorikal.

Tabel 4.5 Data *Type*s

Atribut	*Range* Nilai	Tipe Data
ID	8670 - 957719	*Float*
Diagnosis	M, B	*Categorical*
Radius_mean	6.981 - 28.11	*Float*

Texture_mean	9.71 - 39.28	*Float*
Perimeter_mean	43.79 - 188.5	*Float*
Area_mean	143.5 - 2501.0	*Float*
Smoothness_mean	0.05263 - 0.16340	*Float*
Compactness_mean	0.01938 - 0.34540	*Float*
Concavity_mean	0.00000 - 0.42680	*Float*
Concave_points_mean	0.00000 - 0.20120	*Float*
Symmetry_mean	0.1060 - 0.3040	*Float*
Fractal_dimension_mean	0.04996 - 0.09744	*Float*
Radius_se	0.1115 - 2.873	*Float*
Texture_se	0.3602 - 4.885	*Float*
Perimeter_se	0.7570 - 21.98	*Float*
Area_se	6.802 - 542.2	*Float*
Smoothness_se	0.001713 - 0.03113	*Float*
Compactness_se	0.002252 - 0.13540	*Float*
Concavity_se	0.0000000 - 0.39600	*Float*
Concave_points_se	0.0000000 - 0.05279	*Float*
Symmetry_se	0.007882 - 0.07895	*Float*
Fractal_dimension_se	0.0008948 - 0.029840	*Float*
Radius_worst	7.93 - 36.04	*Float*
Texture_worst	12.02 - 49.54	*Float*
Perimeter_worst	50.41 - 251.2	*Float*
Area_worst	185.2 - 4254.0	*Float*
Smoothness_worst	0.07117 - 0.22260	*Float*
Compactness_worst	0.02729 - 1.05800	*Float*

Concavity_worst	0.00000 - 1.25200	*Float*
Concave_points_worst	0.00000 - 0.29100	*Float*
Symmetry_worst	0.1565 - 0.6638	*Float*
Fractal_dimension_worst	0.05504 - 0.20750	*Float*
Unammed 32	0 – NaN	*Float*

4.4 *Feature Engineering*

Dapat dilihat pada deskripsi dataset diatas bahwa kolom "Diagnosis" mempunyai tipe data Kategorikal atau Object, pada saat penanganan data *Outlier* dan perhitungan std,*mean*,*median* atau *max* akan terjadi error maka data yang bertipe kategorikal harus diubah menjadi data numeric dengan value 0 dan 1, dengan *Benign* menjadi 0 dan *Malignant* menjadi 1.

Tabel 4.6 Labeling

Label	Data *Types before*	Data *Types After*
Diagnosis	*Categorical*	*Float*

a. *Class* atau Label

Class pada dataset ini adalah "Diagnosis" yang akan berfungsi sebagai output dari model *Machine learning* yang dibuat. Kolom ini akan digunakan untuk mengidentifikasi apakah pasien terkena kanker atau tidak. Pada kolom ini terdapat 357 data pasien yang bernilai 0 atau tidak mempunyai kanker, sedangkan 212 pasien terkena kanker disimbolkan dengan nilai 1. Hasil analisa dapat dilihat pada tabel dibawah :

Tabel 4.7 *Labeling*

Keterangan	Nilai
Minimal	0
Maksimal	1
Jumlah Data	569
Data unik	2
Benign	357
Malignant	212

b. Atribut atau *Feature*

1. *Radius_mean*

Pada kolom ini berisi rata-rata jarak dari pusat ke titik pada batas terluar tumor, yang merupakan ukuran *Radius* rata-rata dari sel tumor.

Tabel 4.8 *Feature Radius mean*

Keterangan	Nilai
std	3,52405
min	6,981
25%	11,7
50%	13,37
75%	15,78
max	28,11

2. *Texture_mean*:

   Pada kolom ini berisi rata-rata variasi intensitas warna di dalam gambar digital dari inti sel, yang digunakan untuk menilai tekstur permukaan tumor.

Tabel 4.9 *Feature Texture mean*

Keterangan	Nilai
std	4,30104
min	9,71
25%	16,17
50%	18,84
75%	21,8
max	39,28

3. *Perimeter_mean*:

   Pada kolom ini berisi rata-rata keliling dari batas terluar sel tumor, yang merupakan ukuran seberapa luas tumor menyebar pada permukaannya.

Tabel 4.10 *Feature Perimeter mean*

Keterangan	Nilai
std	24,299
min	43,79
25%	75,17
50%	86,24
75%	104,1
max	188,5

4. *Area_mean*:

   Pada kolom ini berisi rata-rata luas *Area* tumor yang dihitung berdasarkan jumlah piksel dalam gambar digital, memberikan ukuran dari besarnya tumor.

Tabel 4.11 *Feature Area mean*

Keterangan	Nilai
std	351,914
min	143,5
25%	420,3
50%	551,1
75%	782,7
max	2501

5. *Smoothness_mean*:

   Pada kolom ini berisi rata-rata kelancaran batas sel, dihitung sebagai perbedaan antara panjang *Perimeter* yang diukur dan keliling idealnya, memberikan informasi tentang kekasaran tepi tumor.

Tabel 4.12 *Feature Smoothness mean*

Keterangan	Nilai
std	0,01406
min	0,05263
25%	0,08637
50%	0,09587
75%	0,1053
max	0,1634

6. *Compactness_mean*:

   Pada kolom ini berisi rata-rata kompaksi sel tumor, dihitung sebagai rasio antara *Perimeter* kuadrat dengan *Area*, yang menunjukkan seberapa padat tumor.

Tabel 4.13 *Feature Compactness mean*

Keterangan	Nilai
std	0,05281
min	0,01938
25%	0,06492
50%	0,09263
75%	0,1304
max	0,3454

7. *Concavity_mean*:

   Pada kolom ini berisi rata-rata cekungan (*Concavity*) pada batas sel, dihitung sebagai derajat dari cekungan di tepi sel tumor, menggambarkan lekukan yang terbentuk pada tumor

Tabel 4.14 *Feature Concavity mean*

Keterangan	Nilai
std	0,07972
min	0
25%	0,02956
50%	0,06154
75%	0,1307
max	0,4268

8. *Concave points_mean*:

   Pada kolom ini berisi rata-rata jumlah titik cekung pada batas sel tumor, yaitu titik-titik di mana tepi tumor melengkung ke dalam.

Tabel 4.15 *Feature Concave point mean*

Keterangan	Nilai
std	0,0388
min	0
25%	0,02031
50%	0,0335
75%	0,074
max	0,2012

9. *Symmetry_mean*:

   Pada kolom ini berisi rata-rata simetri dari sel tumor, dihitung sebagai perbandingan antara sisi-sisi sel, memberikan informasi tentang kesamaan bentuk di sekitar sumbu sentralnya.

Tabel 4.16 *Feature Symmetry mean*

Keterangan	Nilai
std	0,02741
min	0,106
25%	0,1619
50%	0,1792
75%	0,1957
max	0,304

10. *Fractal_dimension_mean*:

    Pada kolom ini berisi rata-rata dimensi fraktal dari batas sel tumor, dihitung sebagai ukuran kerumitan batas tumor, menggambarkan seberapa rumit atau acak pola dari batas tumor tersebut.

Tabel 4.17 *Feature Fractal dimension mean*

Keterangan	Nilai
std	0,00706
min	0,04996
25%	0,0577
50%	0,06154
75%	0,06612
max	0,09744

11. *Radius*_se:

    Pada kolom ini berisi standar deviasi dari pengukuran *Radius* sel tumor, menunjukkan variasi dalam ukuran *Radius* di seluruh sel tumor yang diukur.

Tabel 4.18 *Feature Radius* se

Keterangan	Nilai
std	0,27731
min	0,1115
25%	0,2324
50%	0,3242
75%	0,4789
max	2,873

12. *Texture*_se:

    Pada kolom ini berisi standar deviasi dari pengukuran tekstur sel tumor, menunjukkan variasi dalam intensitas dan corak permukaan sel tumor.

Tabel 4.19 *Feature Texture* se

Keterangan	Nilai
std	0,55165
min	0,3602
25%	0,8339
50%	1,108
75%	1,474
max	4,885

13. *Perimeter*_se:

    Pada kolom ini berisi standar deviasi dari pengukuran *Perimeter* sel tumor, menunjukkan variasi dalam ukuran keliling sel tumor.

Tabel 4.20 *Feature Perimeter* se

Keterangan	Nilai
std	2,02185
min	0,757
25%	1,606
50%	2,287
75%	3,357
max	21,98

14. *Area*_se:

    Pada kolom ini berisi standar deviasi dari pengukuran *Area* sel tumor, menunjukkan variasi dalam ukuran luas tumor.

Tabel 4.21 *Feature Area* se

Keterangan	Nilai
std	45,491
min	6,802
25%	17,85
50%	24,53
75%	45,19
max	542,2

15. *Smoothness*_se:

    Pada kolom ini berisi standar deviasi dari pengukuran kelancaran tepi sel tumor, menunjukkan variasi dalam kekasaran batas sel.

Tabel 4.22 *Feature Smoothness* se

Keterangan	Nilai
std	0,003
min	0,00171
25%	0,00517
50%	0,00638
75%	0,00815
max	0,03113

16. *Compactness*_se:

    Pada kolom ini berisi standar deviasi dari pengukuran kompaksi sel tumor, menunjukkan variasi dalam kepadatan sel.

Tabel 4.23 *Feature* *Compactness* se

Keterangan	Nilai
std	0,01791
min	0,00225
25%	0,01308
50%	0,02045
75%	0,03245
max	0,1354

17. *Concavity*_se:

    Pada kolom ini berisi standar deviasi dari pengukuran cekungan pada tepi sel tumor, menunjukkan variasi dalam derajat cekungan.

Tabel 4.24 *Feature Concavity* se

Keterangan	Nilai
std	0,03019
min	0
25%	0,01509
50%	0,02589
75%	0,04205
max	0,396

18. *Concave points*_se:

    Pada kolom ini berisi standar deviasi dari pengukuran jumlah titik cekung pada tepi sel tumor, menunjukkan variasi dalam jumlah titik cekung.

Tabel 4.25 *Feature Concave point* se

Keterangan	Nilai
std	0,00617
min	0
25%	0,00764
50%	0,01093
75%	0,01471
max	0,05279

19. *Symmetry*_se:

    Pada kolom ini berisi standar deviasi dari pengukuran simetri sel tumor, menunjukkan variasi dalam kesamaan bentuk di sekitar sumbu sentral.

Tabel 4.26 *Feature Feature Symmetry* se

Keterangan	Nilai
std	0,00827
min	0,00788
25%	0,01516
50%	0,01873
75%	0,02348
max	0,07895

20. *Fractal_dimension*_se:

    Pada kolom ini berisi standar deviasi dari pengukuran dimensi fraktal sel tumor, menunjukkan variasi dalam kerumitan atau pola acak dari batas tumor.

Tabel 4.27 *Feature Fractal dimension* se

Keterangan	Nilai
std	0,00265
min	0,00089
25%	0,00225
50%	0,00319
75%	0,00456
max	0,02984

21. *Radius_worst*:

    Pada kolom ini berisi nilai maksimum dari pengukuran *Radius* sel tumor dalam satu sel, menggambarkan ukuran terbesar dari *Radius* sel tumor.

Tabel 4.28 *Feature Radius worst*

Keterangan	Nilai
std	4,83324
min	7,93
25%	13,01
50%	14,97
75%	18,79
max	36,04

22. *Texture_worst*:

Pada kolom ini berisi nilai maksimum dari pengukuran tekstur sel tumor dalam satu sel, menggambarkan variasi intensitas terbesar pada permukaan sel tumor.

Tabel 4.29 *Feature Texture worst*

Keterangan	Nilai
std	6,14626
min	12,02
25%	21,08
50%	25,41
75%	29,72
max	49,54

23. *Perimeter_worst*:

Pada kolom ini berisi nilai maksimum dari pengukuran *Perimeter* sel tumor dalam satu sel, menggambarkan keliling terbesar dari sel tumor.

Tabel 4.30 *Feature Perimeter worst*

Keterangan	Nilai
std	33,6025
min	50,41
25%	84,11
50%	97,66
75%	125,4
max	251,2

24. *Area_worst*:

Pada kolom ini berisi nilai maksimum dari pengukuran *Area* sel tumor dalam satu sel, menggambarkan ukuran terbesar dari luas tumor.

Tabel 4.31 *Feature Area worst*

Keterangan	Nilai
std	569,357
min	185,2
25%	515,3
50%	686,5
75%	1084
max	4254

25. *Smoothness_worst*:

Pada kolom ini berisi nilai maksimum dari pengukuran kelancaran tepi sel tumor dalam satu sel, menggambarkan kekasaran terbesar dari batas sel.

Tabel 4.32 *Feature Smoothness worst*

Keterangan	Nilai
std	0,02283
min	0,07117
25%	0,1166
50%	0,1313
75%	0,146
max	0,2226

26. *Compactness_worst*:

    Pada kolom ini berisi nilai maksimum dari pengukuran kompaksi sel tumor dalam satu sel, menggambarkan kepadatan terbesar dari sel tumor.

Tabel 4.33 *Feature* *Compactness* *worst*

Keterangan	Nilai
std	0,15734
min	0,02729
25%	0,1472
50%	0,2119
75%	0,3391
max	1,058

27. *Concavity_worst*:

    Pada kolom ini berisi nilai maksimum dari pengukuran cekungan pada tepi sel tumor dalam satu sel, menggambarkan derajat cekungan terbesar.

Tabel 4.34 *Feature Concavity worst*

Keterangan	Nilai
std	0,20862
min	0
25%	0,1145
50%	0,2267
75%	0,3829
max	1,252

28. *Concave points_worst*:

    Pada kolom ini berisi nilai maksimum dari pengukuran jumlah titik cekung pada tepi sel tumor dalam satu sel, menggambarkan jumlah titik cekung terbesar.

Tabel 4.35 *Feature Concave point worst*

Keterangan	Nilai
std	0,06573
min	0
25%	0,06493
50%	0,09993
75%	0,1614
max	0,291

29. *Symmetry_worst*:

    Pada kolom ini berisi nilai maksimum dari pengukuran simetri sel tumor dalam satu sel, menggambarkan kesamaan bentuk terbesar di sekitar sumbu sentralnya.

Tabel 4.36 *Feature Symmetry worst*

Keterangan	Nilai
std	0,061867
min	0,1565
25%	0,2504
50%	0,2822
75%	0,3179
max	0,6638

30. *Fractal_dimension_worst*:

    Pada kolom ini berisi nilai maksimum dari pengukuran dimensi fraktal sel tumor dalam satu sel, menggambarkan kerumitan terbesar atau pola acak dari batas tumor.

Tabel 4.37 *Feature Fractal dimension worst*

Keterangan	Nilai
std	0,01806
min	0,05504
25%	0,07146
50%	0,08004
75%	0,09208
max	0,2075

4.5 *Data Cleaning*

Pada tahap ini, proses pembersihan data dilakukan untuk memastikan kualitas data yang baik, sehingga dapat digunakan dengan tepat dalam analisis. Langkah-langkah yang dilakukan mencakup identifikasi dan koreksi kesalahan dalam kumpulan data, seperti menangani data yang hilang (missing value), menghapus duplikasi, serta menangani outlier.

Dalam analisis sebelumnya, ditemukan bahwa terdapat dua kolom yang tidak memiliki fungsi signifikan dan tidak akan digunakan dalam penelitian ini, yaitu kolom ‘ID’ dan kolom ‘Unnamed: 32’. Kolom ‘ID’ hanya berfungsi sebagai nomor identifikasi sampel, sementara kolom ‘Unnamed: 32’ sepenuhnya berisi nilai NaN.

Tabel 4.38 List Data Abnormal

Atribut	Deskripsi
ID	Nomor identifikasi untuk setiap sampel.
Unammed 32	NaN
Missing Value	Tedapat 0 Isi kolom yang kosong
Outliers	Terdapat 171 data *outlier*
Duplikat	Terdapat 0 data duplikat

4.5.1 *Missing value*

Pada dataset ini, tidak terdapat nilai yang hilang (missing value), sehingga tidak diperlukan penanganan khusus untuk kasus ini. Proses pembersihan data pada bagian ini memastikan bahwa seluruh kolom yang digunakan dalam penelitian memiliki nilai yang lengkap dan konsisten.

4.5.2 Outliers

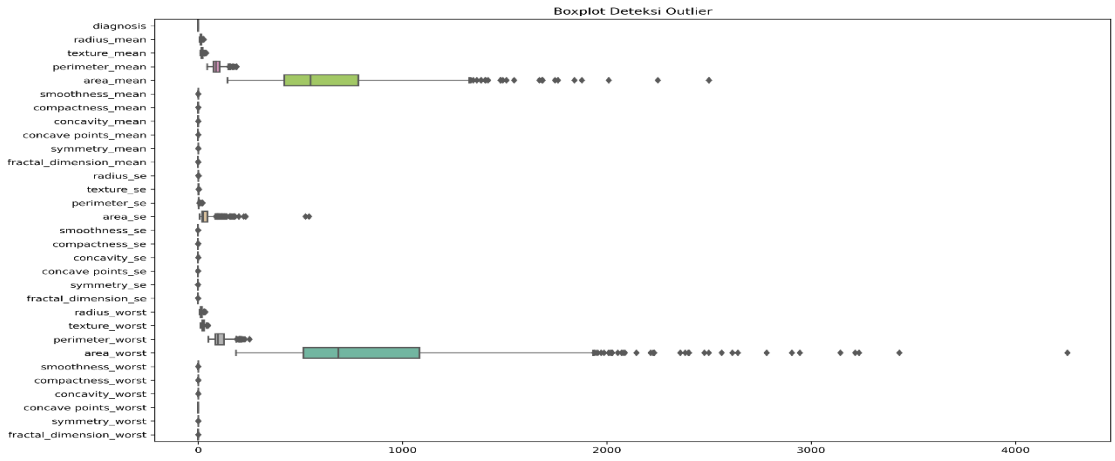
Outlier adalah data yang memiliki nilai yang jauh berbeda atau ekstrem dibandingkan dengan mayoritas data lainnya. Dalam dataset ini, terdapat 171 data yang teridentifikasi sebagai outlier. Penanganan terhadap data outlier ini akan dilakukan dengan metode yang sesuai untuk menjaga kualitas analisis, seperti menggunakan metode trimming atau imputation tergantung dari pengaruh outlier terhadap hasil analisis.

4.5.3 Dupilkat

Pada dataset yang digunakan, tidak ditemukan data duplikat. Hal ini memastikan bahwa setiap entri pada dataset adalah unik, dan tidak ada pengulangan data yang dapat mempengaruhi hasil analisis.

4.6 Data *Outlier*

Outlier adalah nilai yang jauh berbeda dari nilai lainnya dalam kumpulan data. Nilai ini muncul sebagai pengecualian dalam pola data yang ada. Pemrosesan data *Outlier* dapat dilihat pada graph dibawah :

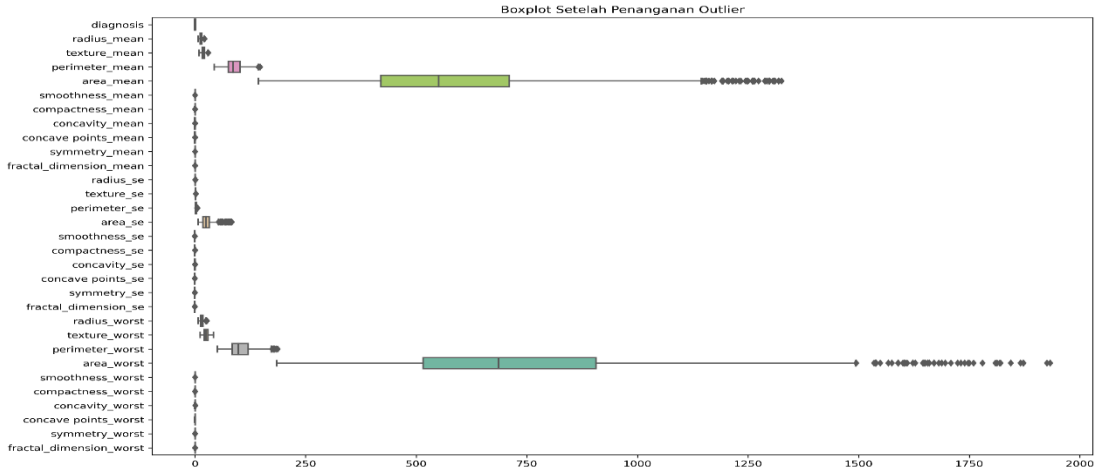


Gambar 4.1 *Outlier* sebelum ditangani

Dapat dilihat dari boxplot diatas Atribut seperti '*Area_mean*' dan '*Area_worst*' memiliki jumlah outliers yang terlihat jelas dengan beberapa titik data jauh berada di sebelah kanan whisker, yang menunjukkan bahwa ada pengamatan dengan nilai yang sangat besar untuk atribut ini.

Pada atribut lain seperti '*Radius*_se, *Perimeter*_se', dan '*Compactness*_se', terlihat juga *outliers* tetapi dengan distribusi yang lebih terpusat dibandingkan dengan atribut lainnya.

Data tersebut dapat diubah dengan nilai yang lebih sesuai atau diubah dengan nilai 0, hasil penangan data *Outlier* dapat dilihat pada boxplot dibawah :



Gambar 4.2 *Outlier* sesudah ditangani

4.7 *Data Corelation* (Data Kolerasi)

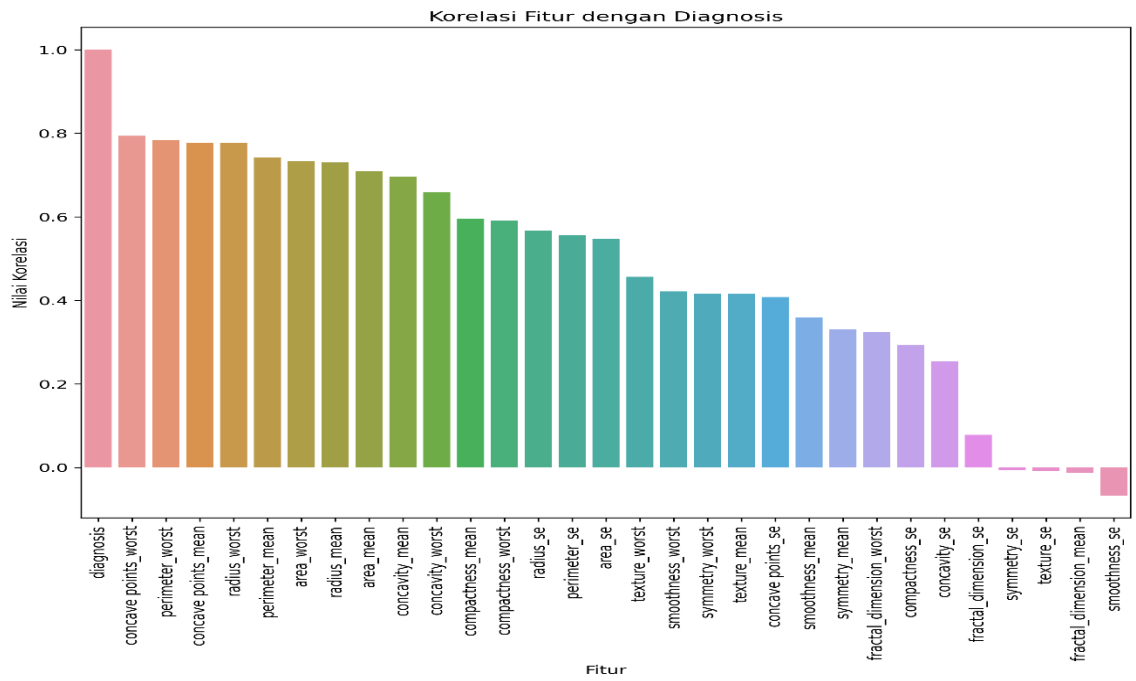
Pada tahap ini dilakukan analisis data set antara kolerasi atribut atribut yang akan digunakan sebagai input terhadap label. Korelasi menunjukan bahwa atribut memiliki pengaruh positif terhdap label. Sedangkan korelasi *Negative* menunjukan pengaruh negative. Seperti yang dapat dilihat pada tabel berikut :

Tabel 4.39 Kolerasi data

Atribut	Diagnosis
Concave points_worst	0.793566
Perimeter_worst	0.782914
Concave points_mean	0.776614
Radius_worst	0.776454
Perimeter_mean	0.742636
Area_worst	0.733825
Radius_mean	0.730029
Area_mean	0.708984
Concavity_mean	0.696360
Concavity_worst	0.659610
Compactness_mean	0.596534
Compactness_worst	0.590998
Radius_se	0.567134
Perimeter_se	0.556141
Area_se	0.548236
Texture_worst	0.456903
Smoothness_worst	0.421465
Symmetry_worst	0.416294
Texture_mean	0.415185
Concave points_se	0.408042
Smoothness_mean	0.358560
Symmetry_mean	0.330499

Fractal_dimension_worst	0.323872
Compactness_se	0.292999
Concavity_se	0.253730
Fractal_dimension_se	0.077972
Symmetry_se	-0.006522
Texture_se	-0.008303
Fractal_dimension_mean	-0.012838
Smoothness_se	-0.067016

Hasil kolerasi antara input terhadap output juga bisa dilihat pada grafik dibawah ini:



Gambar 4.3 Korelasi Data

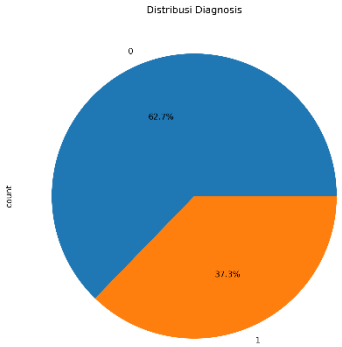
Jika nilai kolerasi mendekati angka 1 itu menunjukan adanya hubungan yang kuat antara atribut input dan output. Semakin dekat nilai korelasi dengan 1 semakin kuat hubungan linier kedua faktor tersebut. Dan sebaliknya pada korelasi atribut yang mendekati angka 0.

4.8 Visualisasi Data

Pada tahap ini akan dilakukan visualisasi data yang telah dioleh dengan menggunakan grafik dan diagram untuk melihat pola data pada dataset.

a. *Class* atau Label

Class pada dataset ini adalah “Diagnosis” yang akan berfungsi sebagai output dari model *Machine learning* yang dibuat. Kolom ini akan digunakan untuk mengidentifikasi apakah pasien terkena kanker atau tidak. Pada kolom ini terdapat 357 data pasien yang bernilai 0 atau tidak mempunyai kanker, sedangkan 212 pasien terkena kanker disimbolkan dengan nilai 1.

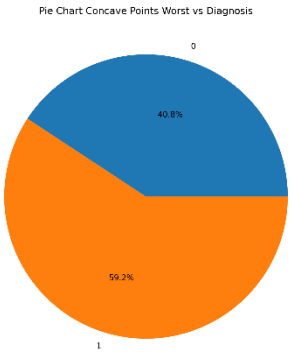


Gambar 4.4 Distribusi Data Diagnosa

Dapat dilihat pada grafik pie diatas bahwa dalam kolom diagnosis terdapat 62.7% pasien yang bernilai 0 yang berarti bahwa pasien tidak mempunyai sel kanker pada tubuhnya, sedangkan 37.3% pasien mempunyai nilai 1 yang berarti mempunyai sel kanker.

b. Atribut atau *Feature*

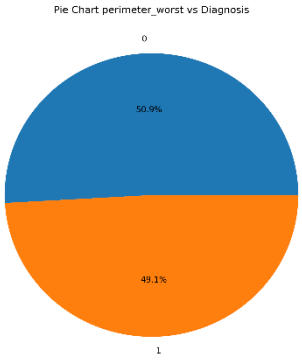
1. *Concave point_worst*



Gambar 4.5 Distribusi Data *Concave point_worst*

Pada grafik dapat disimpulkan bahwa pasien yang berstatus *Malignant* mencapai 59.2% dan 40.8% berstatus *Benign* pada *concave point_worst*.

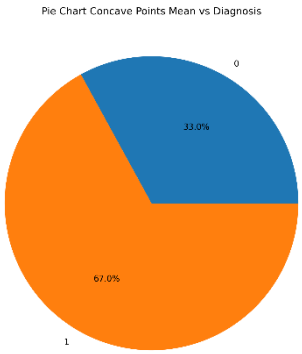
2. *Perimeter_worst*



Gambar 4.6 Distribusi Data *Perimeter_worst*

Pada grafik dapat disimpulkan bahwa pasien yang berstatus *Malignant* berada pada 49.1% dan 50.9% berstatus *Benign* pada *Perimeter_worst*.

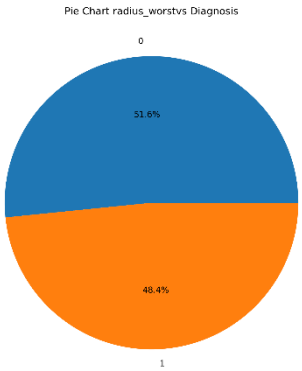
3. *Concave Point_mean*



Gambar 4.7 Distribusi Data *Concave point_mean*

Pada grafik dapat disimpulkan bahwa pasien yang berstatus *Malignant* mendominasi dengan 67% dan 33% berstatus *Benign* pada *Concave point_mean*.

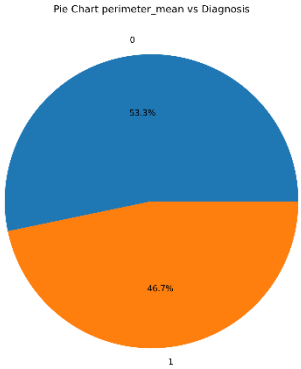
4. *Radius_worst*



Gambar 4.8 Distribusi Data *Radius worst*

Pada grafik dapat disimpulkan bahwa pasien yang berstatus *Malignant* berada pada 48.4% dan 51.6% berstatus *Benign* pada *Radius_worst*.

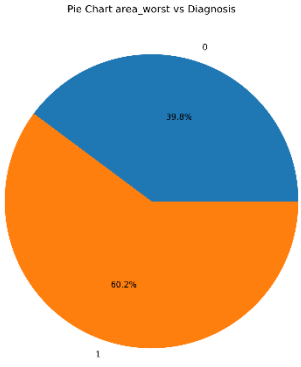
5. *Perimeter_mean*



Gambar 4.9 Distribusi Data *Perimeter mean*

Pada grafik dapat disimpulkan bahwa pasien yang berstatus *Malignant* berada pada 46.7% dan 53.3% berstatus *Benign* pada *Perimeter_mean*.

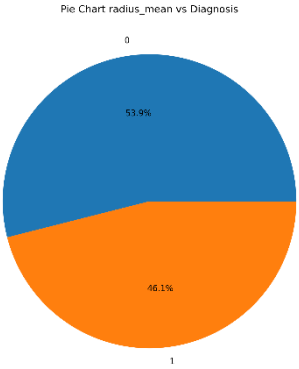
6. *Area_worst*



Gambar 4.10 Distribusi Data *Area worst*

Pada grafik dapat disimpulkan bahwa pasien yang berstatus *Malignant* berada pada 60.2% dan 39.8% berstatus *Benign* pada *Area_worst*.

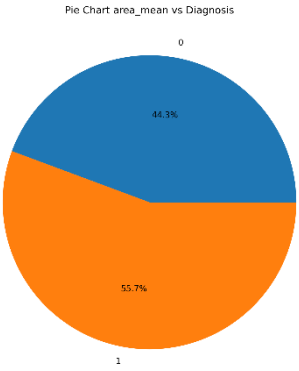
7. *Radius_mean*



Gambar 4.11 Distribusi Data *Radius Mean*

Pada grafik dapat disimpulkan bahwa pasien yang berstatus *Malignant* berada pada 46.1% dan 53.9% berstatus *Benign* pada *Radius_mean*.

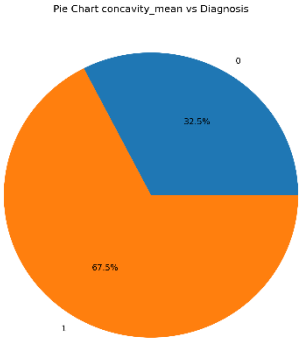
8. *Area_mean*



Gambar 4.12 Distribusi Data *Area Mean*

Pada grafik dapat disimpulkan bahwa pasien yang berstatus *Malignant* berada pada 55.7% dan 44.3% berstatus *Benign* pada *Area_mean*.

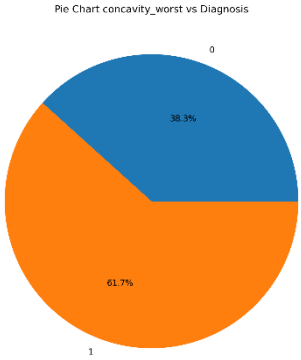
9. *Concavity_mean*



Gambar 4.13 Distribusi Data *Concavity mean*

Pada grafik dapat disimpulkan bahwa pasien yang berstatus *Malignant* berada pada 67.5% dan 32.5% berstatus *Benign* pada *Concavity_mean*.

10. *Concavity_worst*



Gambar 4.14 Distribusi Data *Concavity worst*

Jika dilihat pada grafik dapat disimpulkan bahwa pasien yang berstatus *Malignant* berada pada 61.7% dan 38.3% berstatus *Benign* pada *Concavity_worst*.

4.9 Membangun Model *Machine learning*

Dalam membangun sebuah model mesin kita membuat sesuatu yang merepresentasikan system matematis yang terjadi antar data. Metode yang akan digunakan pada penelitian ini merupakan *Support vector machine* .

a. *Split* data

Pada tahap ini data akan dibagi menjadi 70/30, 70% akan digunakan untuk data latih dan 30% akan digunakan menjadi data uji. Rincian *split* data dapat dilihat pada tabel dibawah :

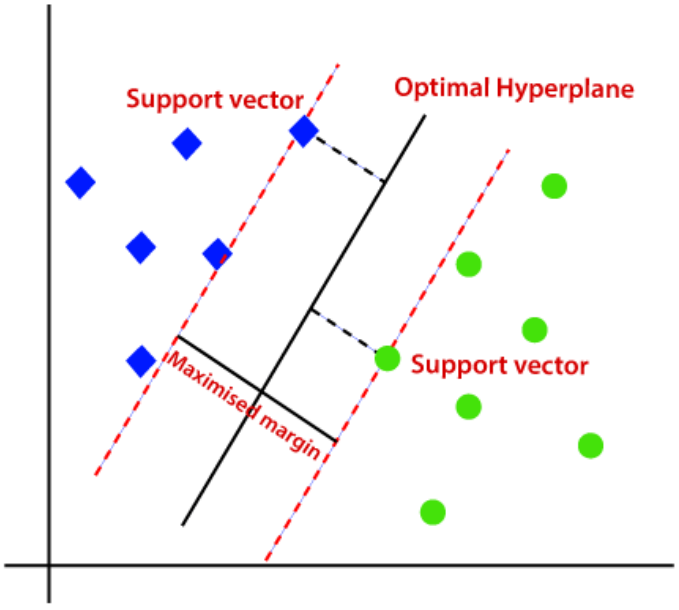
Tabel 4.40 Split data

Dataset	Jumlah
Data Latih	398 baris dan 30 kolom
Data uji	171 baris dan 30 kolom

b. *Train*ing Model

Model menggunakan *Support vector machine* (SVM) yang merupakan salah satu algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. Pada penelitian ini metode yang dipilih adalah klasifikasi. Terdapat beberapa Elemen penting yaitu :

4.10 Pengujian Model Tanpa *Gridsearch*



Gambar 4.15 Skema SVM

Pada tahap ini dilakukan pembuatan model dengan menggunakan parameter yang sudah ditentukan sebelumnya dan memberikan gambaran tentang hasil yang diperoleh oleh model mesin. Ada beberapa matrix evaluasi yang digunakan dalam pengujian model seperti *F1-Score*, Akurasi, Presisi, *Recall*. Model yang dibuat berhasil mendapatkan akurasi sebesar 0.965% yang dapat dilihat pada tabel dibawah ini :

a. *Kernel*

   Kernel disini berfungsi sebagai pengubah data ke dalam bentuk yang lebih tinggi agar lebih mudah dipisahkan. Ketika data tidak dapat dipisahkan secara linear, *Kernel* akan memetakan data ke ruang fitur berdimensi lebih tinggi di mana data tersebut bisa dipisahkan secara linear.

b. *Support Vectors*

   Support Vectors adalah titik data yang paling dekat dengan *Hyperplane* dan memiliki pengaruh langsung terhadap penentuan posisi *Hyperplane*. bekerja dengan menggunakan titik-titik ini untuk menentukan margin maksimum antara dua kelas.

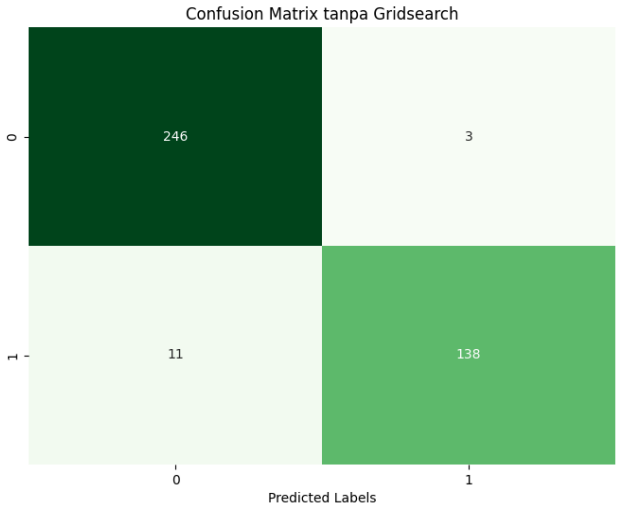
c. *Hyperplane*

   Hyperplane adalah batas keputusan yang digunakan untuk memisahkan kelas-kelas yang berbeda dalam data. *Hyperplane* yang memisahkan dua kelas data dengan margin terbesar (jarak terjauh dari data terdekat dari kedua kelas)

Tabel 4.41 Hasil tanpa *Gridsearch*

Model	**Hasil**
Accuracy	0.965%
Precision	0.96%
Recall	0.99%
F1-Score	0.97%

Pada tahap ini digunakan *Confusion matrix*, untuk memberikan gambaran tentang seberapa baik model dapat mengklasifikasikan input data kedalam kategori yang sudah ditentukan, seperti yang terlihat pada matrix berikut :

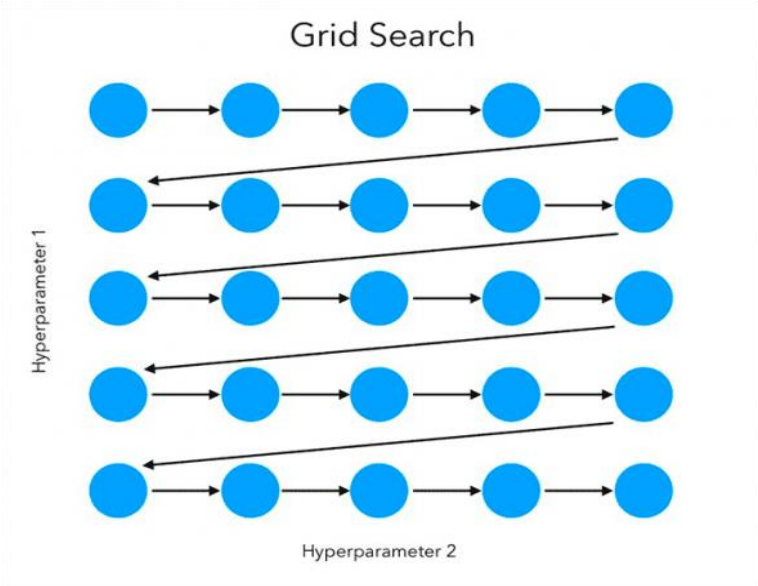


Gambar 4.16 *Confusion matrix* tanpa *Gridsearch*

Berdasarkan hasil dari *Confusion matrix* dapat disimpulkan bahwa :

a. TP (*True Positives*): 138 (Model memprediksi positif dan benar-benar positif)
b. TN (*True Negatives*): 246 (Model memprediksi negatif dan benar-benar negatif)
c. FP (*False Positives*): 11 (Model memprediksi positif, tetapi sebenarnya negatif)
d. FN (*False Negatives*): 3 (Model memprediksi negatif, tetapi sebenarnya positif)

4.11 Pengujian Model dengan *Gridsearch*



Gambar 4.17 Skema *Gridsearch*

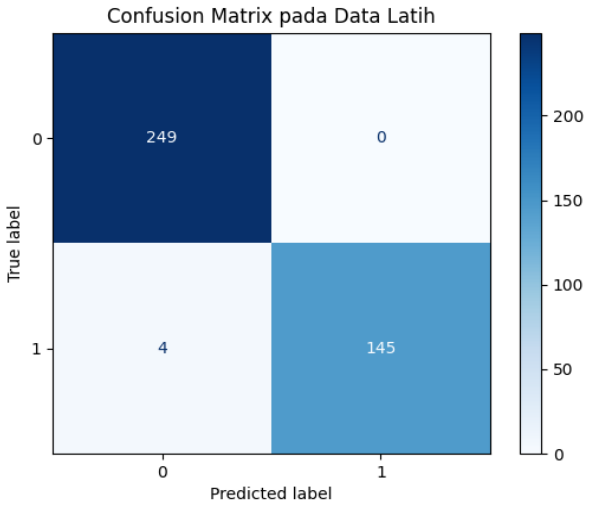
Gridsearch digunakan untuk mengoptimalkan *Hyperparameter* dari model yang sedang diuji, pada penelitian ini yaitu *Support vector machine* (SVM). Dengan menguji berbagai kombinasi *Hyperparameter*, model yang dihasilkan diharapkan memiliki performa yang lebih baik dalam hal akurasi atau metrik evaluasi lainnya.

Pada gambar di atas, *Gridsearch* melakukan eksplorasi sistematis terhadap semua kemungkinan kombinasi *Hyperparameter*, yang digambarkan sebagai titik-titik biru. Setiap sumbu mewakili *Hyperparameter* yang berbeda, dan setiap titik di grid merepresentasikan satu kombinasi nilai *Hyperparameter*. Proses ini mengevaluasi kinerja model pada setiap kombinasi tersebut untuk menentukan yang terbaik.

Tabel 4.42 Hasil uji

Model	**Hasil**
Accuracy	0.990%
Precision	0.98%
Recall	1.00%
F1-Score	0.99%

Pada tahap ini digunakan *Confusion matrix*, untuk memberikan gambaran tentang seberapa baik model dapat mengklasifikasikan input data kedalam kategori yang sudah ditentukan, seperti yang terlihat pada matrix berikut :



Gambar 4.18 *Confusion matrix* data latih

Berdasarkan hasil dari *Confusion matrix* dapat disimpulkan bahwa :

a. TP (*True Positives*): 145 (Model memprediksi positif dan benar-benar positif)
b. TN (*True Negatives*): 249 (Model memprediksi negatif dan benar-benar negatif)
c. FP (*False Positives*): 4 (Model memprediksi positif, tetapi sebenarnya negatif)
d. FN (*False Negatives*): 0 (Model memprediksi negatif, tetapi sebenarnya positif)

4.12 Evaluasi Dan Validasi Akhir

Pada tahap ini dilakukan proses pengujian model mesin learning yang telah dibuat dengan menggunakan pola data yang baru yang telah disiapkan sebelumnya, berikut adalah dataset baru pada tabel dibawah :

Tabel 4.43 Data uji

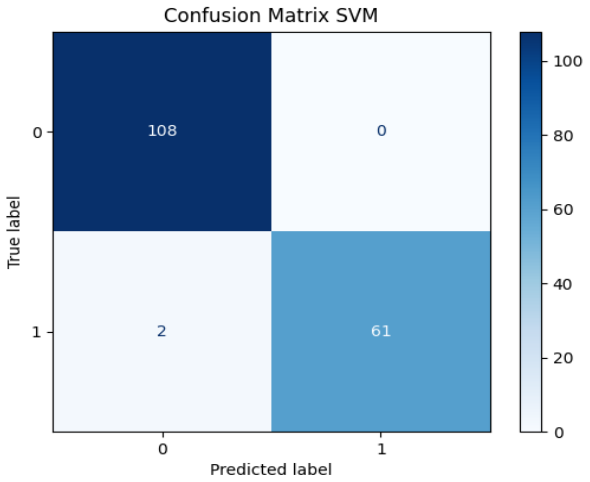
Dataset	Jumlah
Data *test*ing	171 baris data dan 30 kolom

Dari tabel diatas dapat disimpulkan bahwa dataset yang akan digunakan dalam pengujian dan evaluasi didapatkan dari split dataset awal yang berjumlah 569 di split menjadi 30% dan digunakan dalam *test*ing model ini, berikut hasil yang didapatkan setelah menguji sesuai dengan model yang sudah dibangun :

Tabel 4.44 Hasil Evaluasi

Model	Hasil
Accuracy	0.988%
Precision	0.98%
Recall	1.00%
F1-Score	0.99%

Dari tabel diatas dapat disimpulkan bahwa model mesin mengalami penurunan sebanyak 0.02% yang berarti neglible atau tidak berpengaruh pada hasil akurasi model.



Gambar 4.19 *Confusion matrix* Evaluasi data uji

Dapat dilihat pada matrix confusion untuk mengamati hasil pengujuan model mesin dengan lebih akurat :

a. TP (*True Positives*): 61 (Model memprediksi positif dan benar-benar positif)
b. TN (*True Negatives*): 108 (Model memprediksi negatif dan benar-benar negatif)
c. FP (*False Positives*): 2 (Model memprediksi positif, tetapi sebenarnya negatif)
d. FN (*False Negatives*): 0 (Model memprediksi negatif, tetapi sebenarnya positif.

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan untuk implementasi algoritma *Support vector machine* sebagai pendeteksi awal penyakit Kanker Payudara berhsil mencapai tingkat akurasi yang sangat tinggi.

Hasil penelitian menunjukan bahwa pada model mesin *support vector machine* tanpa menggunakan *gridsearch* mencapai tingkat akurasi sebesar 9.65%, *Precision* 96%, *Recall* 99%, *F1-Score* 97%, dan saat model mesin *support vector machine* menggunakan *gridsearch* menunjukan bahwa model mesin mencapai tingkat akurasi sebesar 98.8%, *Precision* 98%, *Recall* 100%, *F1-Score* 99%. Maka dari hasil tersebut dapat disimpulkan bahwa penggunaan *gridsearch* pada algoritma *support vector machine* dapat meningkatkan akurasi model mesin pembelajaran sebanyak 0.23% pada akurasi, 2% pada *Precison,* 1% pada *Recall* dan2 2% pada *F1-Score*.

5.2 Saran

Berdasarkan hasil penelitian ini kami merasa perlu memberikan sejumlah saran yang dapat membantu peneliti di masa depan untuk mengatasi beberapa hambatan atau meningkatkan penelitian ini lebih jauh lagi. Saran-saran mencakup :

a. Dalam penelitian berikutnya disarankan untuk menggunakan dataset yang lebih banyak dan baik.
b. Dalam penelitian selanjutnya disarankan untuk menggunakan metode atau algoritma pembanding.

DAFTAR PUSTAKA

American Cancer Society. (2022). Breast Cancer What is breast cancer? *American Cancer Society. Cancer Facts and Figures Atlanta, Ga: American Cancer Society*, 1–19. http://www.cancer.org/cancer/breast-cancer/about/what-is-breast-cancer.html

Andryan, M. R., Fajri, M., & Sulistyowati, N. (2022). Komparasi Kinerja Algoritma Xgboost Dan Algoritma Support vector machine (Svm) Untuk Diagnosis Penyakit Kanker Payudara. *JIKO (Jurnal Informatika Dan Komputer)*, *6*(1), 1. https://doi.org/10.26798/jiko.v6i1.500

Bose, M., Biswas, N., & Sarkar, D. (2024). *Unraveling the Network Landscape: A Comparative Analytical Approach to Investigate Protein–Protein Interaction Networks in Normal v/s Tumor Cells BT - Proceedings of 4th International Conference on Frontiers in Computing and Systems* (D. K. Kole, S. Roy Chowdhury, S. Basu, D. Plewczynski, & D. Bhattacharjee (eds.); pp. 483–506). Springer Nature Singapore.

Faris, H., Habib, M., Faris, M., Alomari, M., & Alomari, A. (2020). Medical speciality classification system based on binary particle swarms and ensemble of one vs. rest support vector machine s. *Journal of Biomedical Informatics*, *109*, 103525. https://doi.org/10.1016/j.jbi.2020.103525

G., S., & Brindha, S. (2022). Hyperparameters Optimization using Gridsearch Cross Validation Method for machine learning models in Predicting Diabetes Mellitus Risk. *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, 1–4. https://doi.org/10.1109/IC3IOT53935.2022.9768005

Gupta, P., & Bagchi, A. (2024). *Data Manipulation with Pandas BT - Essentials of Python for Artificial Intelligence and Machine learning* (P. Gupta & A. Bagchi (eds.); pp. 197–235). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-43725-0_6

Häberlein, T. (2024). *Numpy BT - Programmieren mit Python: Eine Einführung in*

die Prozedurale, Objektorientierte und Funktionale Programmierung (T. Häberlein (ed.); pp. 149–167). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-68678-2_5

Hetland, M. L., & Nelli, F. (2024). *Activity 1: Data Analysis with Pandas, Matplotlib, and Seaborn BT - Beginning Python: From Novice to Professional* (M. L. Hetland & F. Nelli (eds.); pp. 487–504). Apress. https://doi.org/10.1007/979-8-8688-0196-9_25

Hooshmand, M. N., & Maserat, E. (2024). Application of machine learning and deep learning for cancer vaccine (rapid review). *Multimedia Tools and Applications*, *83*(17), 51211–51226. https://doi.org/10.1007/s11042-023-17589-8

Hunt, J. (2023). *Pandas and Data Analytics BT - Advanced Guide to Python 3 Programming* (J. Hunt (ed.); pp. 611–627). Springer International Publishing. https://doi.org/10.1007/978-3-031-40336-1_54

Hurriyati, S. (2023). *Implementasi metode support vector machine pada klasifikasi diagnosis penyakit kanker payudara*. http://etheses.uin-malang.ac.id/52582/%0Ahttp://etheses.uin-malang.ac.id/52582/1/16610041.pdf

Imaduddin, H., Hermansyah, B. A., & Salsabilla B, F. A. (2021). Comparison of Support vector machine and Decision tree Methods in the Classification of Breast Cancer. *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, *5*(1), 22. https://doi.org/10.22373/cj.v5i1.8805

Islam, M. M., Haque, M. R., Iqbal, H., Hasan, M. M., Hasan, M., & Kabir, M. N. (2020). Breast Cancer Prediction: A Comparative Study Using Machine learning Techniques. *SN Computer Science*, *1*(5), 290. https://doi.org/10.1007/s42979-020-00305-w

Jo, T. (2021). Machine learning foundations: Supervised, unsupervised, and advanced learning. In *Machine learning Foundations: Supervised, Unsupervised, and Advanced Learning*. https://doi.org/10.1007/978-3-030-65900-4

Kabiraj, S., Raihan, M., Alvi, N., Afrin, M., Akter, L., Sohagi, S. A., & Podder, E. (2020). Breast Cancer Risk Prediction using XGBoost and Random forest Algorithm. *2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020*, 1–4. https://doi.org/10.1109/ICCCNT49239.2020.9225451

Klemp, M. (2024). *Python BT - Computer Science in Sport: Modeling, Simulation, Data Analysis and Visualization of Sports-Related Data* (D. Memmert (ed.); pp. 125–131). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-68313-2_15

Linge, S., & Langtangen, H. P. (2020). Programming for Computations - Python. In *Springer Open* (Vol. 15). http://link.springer.com/10.1007/978-3-030-16877-3

Munawarah, R., Soesanto, O., & Faisal, M. R. (2022). Penerapan Metode Support vector machine . *Kumpulan JurnaL Ilmu Komputer (KLIK)*, *04*(01), 103–113.

Nelli, F. (2023a). *Data Visualization with matplotlib and Seaborn BT - Python Data Analytics: With Pandas, NumPy, and Matplotlib* (F. Nelli (ed.); pp. 183–257). Apress. https://doi.org/10.1007/978-1-4842-9532-8_7

Nelli, F. (2023b). *Machine learning with scikit-learn BT - Python Data Analytics: With Pandas, NumPy, and Matplotlib* (F. Nelli (ed.); pp. 259–287). Apress. https://doi.org/10.1007/978-1-4842-9532-8_8

Renita, J., & Puspita Politeknik Kesehatan Kementerian Kesehatan Bengkulu, Y. (2023). Pengaruh Metode Ceramah Kombinasi Media Leaflet Terhadap Pengetahuan Wanita Usia Subur Tentang Pemeriksaan Payudara Sendiri Di Wilayah Kerja Puskesmas Pasar Kepahiang Tahun 2023 the Effect of the Combination Media Leaflet Lecture Method on the Knoewlege o. *Jm*, *11*(2), 265–271.

Sarker, I. H. (2021). Machine learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, *2*(3), 160. https://doi.org/10.1007/s42979-021-00592-x

Schonlau, M. (2021). *GRIDSEARCH: Stata module to optimize tuning parameter levels with a grid search*. https://econpapers.repec.org/RePEc:boc:bocode:s458859

Septhya, D., Rahayu, K., Rabbani, S., Fitria, V., Rahmaddeni, R., Irawan, Y., & Hayami, R. (2023). Implementasi Algoritma Decision tree dan Support vector machine untuk Klasifikasi Penyakit Kanker Paru. *MALCOM: Indonesian Journal of Machine learning and Computer Science*, *3*(1), 15–19. https://doi.org/10.57152/malcom.v3i1.591

Silaparasetty, N. (2020). *Introduction to Jupyter Notebook BT - Machine learning Concepts with Python and the Jupyter Notebook Environment: Using Tensorflow 2.0* (N. Silaparasetty (ed.); pp. 91–118). Apress. https://doi.org/10.1007/978-1-4842-5967-2_6

Testas, A. (2023). *Pipelines with Scikit-Learn and PySpark BT - Distributed Machine learning with PySpark: Migrating Effortlessly from Pandas and Scikit-Learn* (A. Testas (ed.); pp. 441–461). Apress. https://doi.org/10.1007/978-1-4842-9751-3_17

Tsui, K.-L., Chen, V., Jiang, W., Yang, F., & Kan, C. (2023). *Data Mining Methods and Applications BT - Springer Handbook of Engineering Statistics* (H. Pham (ed.); pp. 797–816). Springer London. https://doi.org/10.1007/978-1-4471-7503-2_38

Wu, W.-T., Li, Y.-J., Feng, A.-Z., Li, L., Huang, T., Xu, A.-D., & Lyu, J. (2021). Data mining in clinical big data: the frequently used databases, steps, and methodological models. *Military Medical Research*, *8*(1), 44. https://doi.org/10.1186/s40779-021-00338-z

LAMPIRAN

In [1]:

```
#import library
import pandas as pd
import numpy as np

#library visualisasi
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

# {Pemisah data
from sklearn.model_selection import train_test_split

# data modeling
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier

# performa data
from sklearn.model_selection import RandomizedSearchCV,GridSearchCV
from sklearn.model_selection import KFold
from sklearn.metrics import f1_score, accuracy_score, confusion_matrix, classification_report
from sklearn import metrics

#warnings
import warnings
```

In [2]:

```
#import dataset
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
/kaggle/input/breast-cancer-wisconsin-data/data.csv
```

In [3]:

```
# Dataframe datasets
df = pd.read_csv('/kaggle/input/breast-cancer-wisconsin-data/data.csv')
df.head()
```

Out[3]:

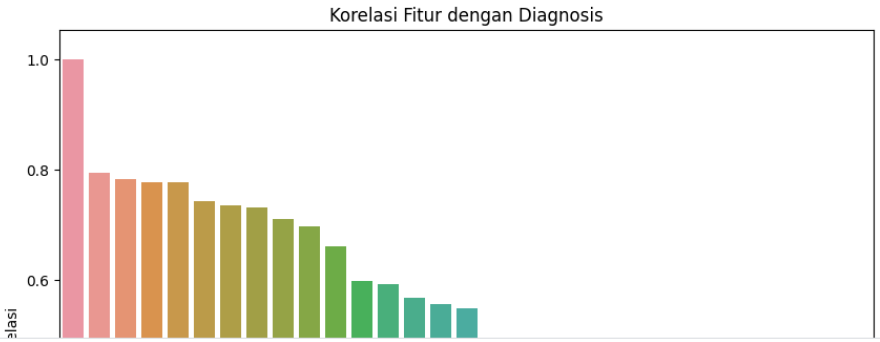
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	con poi
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.1
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.0
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.1
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.1
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.1

In [9]:
```
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1

# Mendefinisikan Outlier sebagai data di luar rentang [Q1 - 1.5*IQR, Q3 + 1.5*IQR]
outliers = (df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))

# Menampilkan baris yang mengandung outlier
outlier_data = df[outliers.any(axis=1)]
outlier_data
```

Out[9]:

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
0	1	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710
1	1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017
2	1	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790
3	1	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520
4	1	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430
...	...	...	...	...	...	...	...	...	...
563	1	20.92	25.09	143.00	1347.0	0.10990	0.22360	0.3174	0.14740
564	1	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.2439	0.13890
565	1	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.1440	0.09791
567	1	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.3514	0.15200
568	0	7.76	24.54	47.92	181.0	0.05263	0.04362	0.0000	0.00000

[76]:
```
# Membuat plot korelasi
plt.figure(figsize=(10, 8))
sns.barplot(x=correlation.index, y=correlation.values)

# Menambahkan judul dan label
plt.title('Korelasi Fitur dengan Diagnosis')
plt.xlabel('Fitur')
plt.ylabel('Nilai Korelasi')
plt.xticks(rotation=90)
plt.show()
```



[79]:
```
import pandas as pd

# Mengkategorikan concave points worst berdasarkan diagnosis
concave_points_by_diagnosis = df.groupby('diagnosis')['perimeter_worst'].sum()

# Membuat pie chart
plt.figure(figsize=(8, 8))
concave_points_by_diagnosis.plot(kind='pie', autopct='%1.1f%%')
plt.title('Pie Chart perimeter_worst vs Diagnosis')
plt.ylabel('')  # Menghilangkan label y

plt.show()
```

[84]:

```
from sklearn.model_selection import train_test_split

# Misalkan df adalah DataFrame yang memuat dataset Anda
# X adalah fitur, y adalah label/target
X = df.drop(columns=['diagnosis'])  # Ganti 'label_column' dengan nama kolom label Anda
y = df['diagnosis']

# Membagi data menjadi 70% data latih dan 30% data uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Menampilkan hasil pembagian
print(f"Ukuran Data Latih: {X_train.shape[0]}")
print(f"Ukuran Data Uji: {X_test.shape[0]}")
```

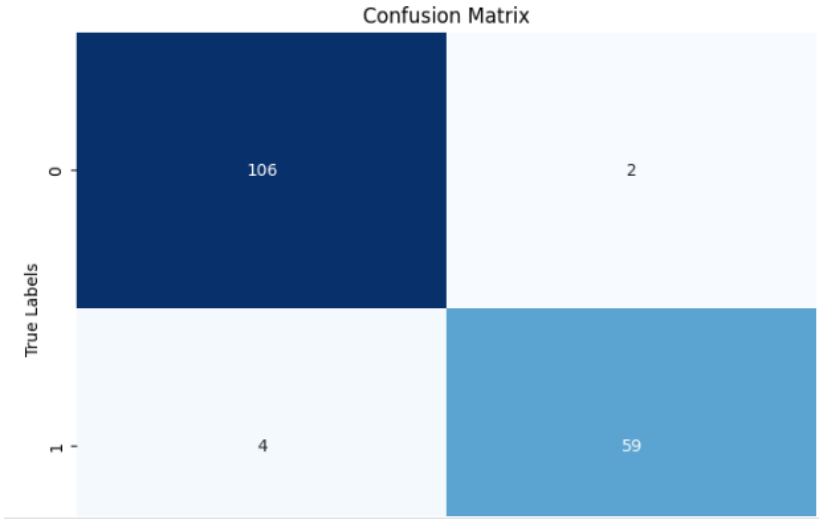
```
Ukuran Data Latih: 398
Ukuran Data Uji: 171
```

```
# Inisialisasi model SVM
model_svm = SVC(kernel='linear', random_state=42)

# Melakukan training pada data latih
model_svm.fit(X_train, y_train)

# Memprediksi data uji
y_pred = model_svm.predict(X_test)

# Evaluasi model
accuracy = accuracy_score(y_test, y_pred)
print(f'Akurasi Model SVM: {accuracy:.2f}')

# Menampilkan laporan klasifikasi
print(classification_report(y_test, y_pred))
```

```
Akurasi Model SVM: 0.96
              precision    recall  f1-score   support

           0       0.96      0.98      0.97       108
           1       0.97      0.94      0.95        63

    accuracy                           0.96       171
   macro avg       0.97      0.96      0.96       171
weighted avg       0.96      0.96      0.96       171
```

+ Code + Markdown

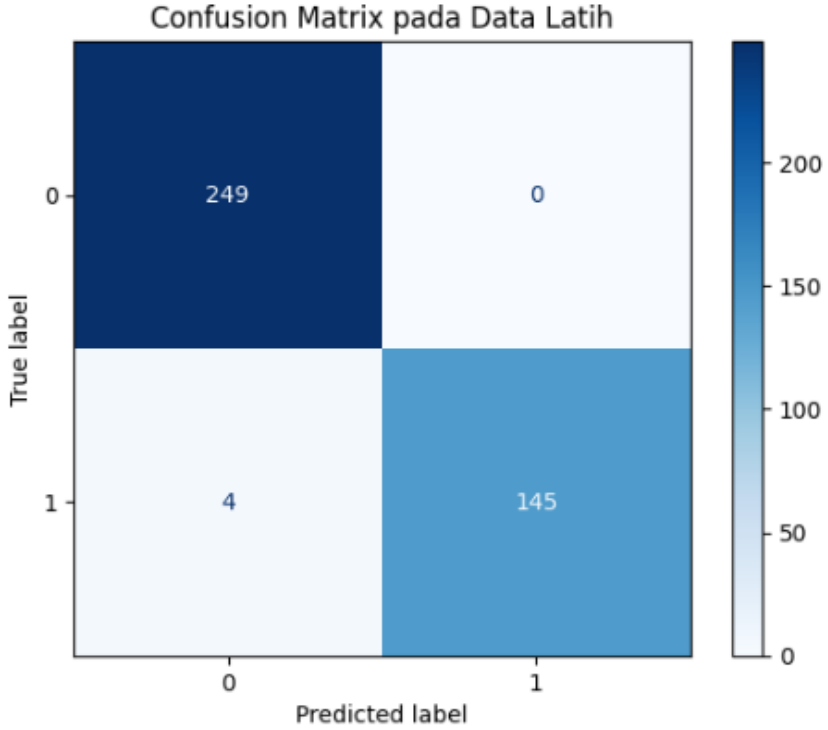
```
Ukuran Data Latih: 398
Ukuran Data Uji: 171
Akurasi Model SVM: 0.96
              precision    recall  f1-score   support

           0       0.96      0.98      0.97       108
           1       0.97      0.94      0.95        63

    accuracy                           0.96       171
   macro avg       0.97      0.96      0.96       171
weighted avg       0.96      0.96      0.96       171
```



[89]:

```
# Me# Melatih model menggunakan data latih
pipeline.fit(X_train, y_train)

# Memprediksi data latih (untuk evaluasi internal)
y_train_pred = pipeline.predict(X_train)

# Evaluasi model pada data latih
accuracy_train = accuracy_score(y_train, y_train_pred)
print(f'Akurasi Model pada Data Latih: {accuracy_train:.2f}')

# Menampilkan laporan klasifikasi
print(classification_report(y_train, y_train_pred))
```

```
Akurasi Model pada Data Latih: 0.99
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       249
           1       1.00      0.97      0.99       149

    accuracy                           0.99       398
   macro avg       0.99      0.99      0.99       398
weighted avg       0.99      0.99      0.99       398
```



Akurasi Model pada Data Latih: 0.990

	precision	recall	f1-score	support
0	0.98	1.00	0.99	249
1	1.00	0.97	0.99	149
accuracy			0.99	398
macro avg	0.99	0.99	0.99	398
weighted avg	0.99	0.99	0.99	398