

**SISTEM PENDUKUNG KEPUTUSAN UNTUK SELEKSI CALON  
KONSUMEN PADA PERUSAHAAN PEMBIAYAAN DENGAN  
MENGGUNAKAN *MACHINE LEARNING***

**TUGAS AKHIR**



**AGNES GIOVANNI ALIANTO  
311910027**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS MA CHUNG  
MALANG  
2023**

**LEMBAR PENGESAHAN**  
**TUGAS AKHIR**

**SISTEM PENDUKUNG KEPUTUSAN UNTUK SELEKSI CALON  
KONSUMEN PADA PERUSAHAAN PEMBIAYAAN DENGAN  
MENGGUNAKAN *MACHINE LEARNING***

Oleh:  
**AGNES GIOVANNI ALIANTO**  
**NIM. 311910027**

dari:  
**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS SAINS dan TEKNOLOGI**  
**UNIVERSITAS MA CHUNG**

Telah dinyatakan lulus dalam melaksanakan Tugas Akhir sebagai syarat kelulusan  
dan berhak mendapatkan gelar Sarjana Komputer (S.Kom.)

**Dosen Pembimbing I,**

  
Hendry Setiawan, ST., M.Kom.  
NIP. 20100006

**Dosen Pembimbing II,**

  
Paulus Lucky Tirma Irawan, S.Kom.,  
MT.  
NIP. 20100005

Dekan Fakultas Sains dan Teknologi,



Dr. Kestella Rega Prilianti, M.Si.  
NIP. 20120035

## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan “Sistem Pendukung Keputusan Untuk Seleksi Calon Konsumen Pada Perusahaan Pembiayaan dengan Menggunakan *Maching Learning*” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Malang, 21 Juli 2023



A handwritten signature in blue ink, appearing to read "Agnes Giovanni Alianto", is placed to the right of the QR code watermark.

**Agnes Giovanni Alianto**

**NIM. 311910027**

# **SISTEM PENDUKUNG KEPUTUSAN UNTUK SELEKSI CALON KONSUMEN PADA PERUSAHAAN PEMBIAZAAN DENGAN MENGGUNAKAN *MACHINE LEARNING***

Agnes Giovanni Alianto, Hendry Setiawan, Paulus Lucky Tirma Irawan

## **Abstrak**

Pada masa kini, permintaan masyarakat yang tinggi akan barang membuat banyak orang ingin melakukan pengajuan kredit kepada pemberian. Meningkatnya pengajuan kepada pemberian dapat membuat pihak pemberian melakukan kesalahan dalam memilih pelanggan yang memiliki potensi dapat membayar kredit dengan baik. Oleh karena itu, dibutuhkan sistem yang dapat mendukung pemberian dalam mengambil keputusan untuk memilih calon konsumen.

Pada pembuatan sistem akan menggunakan metode algoritma *machine learning* di mana akan membandingkan performa dari beberapa algoritma seperti *Decision Tree*, *K-Nearest Neighbor*, *Support Vector Machine*, *Bagging Classifier*, dan *Ensemble Method*. Tujuan dari perbandingan ini adalah untuk mengetahui potensi paling baik dari metode algoritma *machine learning* mana yang paling cocok untuk menyeleksi calon konsumen. Pengujian akan dilakukan menggunakan *cross validation* dan *confusion matrix*.

Hasil dari *cross validation* menunjukkan bahwa metode algoritma *K-Nearest Neighbor* terhadap nilai *standard deviation* mendapatkan nilai paling kecil, yaitu 0.07 sedangkan hasil dari *confusion matrix* menunjukkan bahwa metode *Support Vector Machine* mencapai akurasi paling tinggi, yaitu 90%.

**Kata Kunci :** Kredit, Pengambilan Keputusan, *Machine Learning*

# **DECISION SUPPORT SYSTEM FOR SELECTION OF PROSPECTIVE CONSUMERS IN FINANCING COMPANIES USING MACHINE LEARNING**

Agnes Giovanni Alianto, Hendry Setiawan, Paulus Lucky Tirma Irawan

## **Abstract**

At present, the high public demand for goods makes many people want to apply for credit for financing. Increasing submissions to financing can make financing parties make mistakes in choosing customers who have the potential to pay credit well. Therefore, a system is needed that can support financing in making decisions to choose potential customers.

In making the system will use a machine learning algorithm method which will compare the performance of several algorithms such as Decision Tree, K-Nearest Neighbor, Support Vector Machine, Bagging Classifier, and Ensemble Method. The purpose of this comparison is to find out the best potential of which machine learning algorithm method is most suitable for selecting potential customers. Testing will be carried out using cross validation and confusion matrix.

The results of the cross validation show that the K-Nearest Neighbor algorithm method for the standard deviation value gets the smallest value, namely 0.07 while the results of the confusion matrix show that the Support Vector Machine method achieves the highest accuracy, namely 90%.

**Key Words :** Credit, Decision Making, Machine Learning

## KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa atas berkat dan rahmat-Nya sehingga laporan Tugas Akhir yang berjudul “Sistem Pendukung Keputusan Untuk Seleksi Calon Konsumen Pada Perusahaan Pembiayaan dengan Menggunakan *Machine Learning*” dapat diselesaikan dengan baik dan tepat waktu. Dalam proses penyusunan laporan ini tidak lupa penulis hendak menyampaikan ucapan terima kasih kepada banyak pihak yang telah membantu dan mendukung penulis. Oleh karena itu, penulis mengucapkan terima kasih kepada :

1. Ibu Dr. Kestrilia Rega Prilianti, M.Si selaku Dekan dari Fakultas Sains dan Teknologi Universitas Ma Chung dan Ketua Pengaji Tugas Akhir,
2. Bapak Hendry Setiawan, ST, M.Kom, selaku Kepala Program Studi Teknik Informatika dan Dosen Pembimbing I Tugas Akhir,
3. Bapak Paulus Lucky Tirma Irawan, S.Kom., MT, selaku Dosen Pembimbing II Tugas Akhir,
4. Orang tua yang selalu mendoakan dan memberikan semangat serta dukungan,
5. Teman-teman seperjuangan yang telah memberikan dukungan selama Tugas Akhir berlangsung.

Akhir kata, penulis menyadari bahwa laporan ini tidak lepas dari kekurangan, namun penulis telah berusaha semaksimal mungkin untuk menyajikan hasil yang terbaik. Penulis berharap bahwa laporan ini dapat memberikan manfaat dan kontribusi dalam bidang studi yang relevan. Apabila terdapat hal-hal lain yang dapat diperbaiki atau ditambahkan di masa mendatang, penulis dengan senang hati akan menerimanya untuk meningkatkan kualitas penulisan dan penelitian yang lebih baik. Terima kasih atas perhatian dan dukungan semua pihak yang telah membantu dalam penyelesaian Tugas Akhir ini.

Malang, 21 Juli 2023

Agnes Giovanni Alianto

## **DAFTAR ISI**

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iv
DAFTAR TABEL	vi
BAB I	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Batasan Masalah	3
1.4 Rumusan Masalah	3
1.5 Tujuan Penelitian	3
1.6 Manfaat Penelitian	3
1.7 Luaran Tugas Akhir	4
1.8 Sistematika Penulisan	4
BAB II	6
2.1 Data	6
2.2 <i>Customer</i>	6
2.3 Kredit	7
2.4 <i>Machine Learning</i>	8
2.5 <i>Decision Tree</i>	10
2.6 <i>K-Nearest Neighbor</i> (KNN)	12
2.7 <i>Support Vector Machine</i> (SVM)	13
2.8 <i>Bagging Classifier</i>	14
2.9 <i>Ensemble Method</i>	14
2.10 <i>Confusion Matrix</i>	15
2.11 <i>Cross Validation</i>	16
2.12 Python	17
2.13 Scikit-Learn	19
2.14 HTML	22
2.15 Bootstrap	23
2.16 Flask	24

2.17 PyCharm	25
2.18 MySQL	26
2.19 Postman	27
2.20 Penelitian Terdahulu	28
BAB III	30
3.1 Tahapan Penelitian	30
3.2 Analisis Kebutuhan	31
3.2.1 Atribut Data	32
3.3 Perancangan Sistem	32
3.4 Pembuatan Sistem	42
3.5 Pengujian Sistem	42
BAB IV	44
4.1 Pengumpulan Data	44
4.2 Preprocessing Data	44
4.3 Gambaran Profil Konsumen	47
4.4 Pengujian Menggunakan Cross Validation	48
4.5 Pengujian Akurasi Metode Menggunakan Confusion Matrix	53
4.6 Pengujian Sistem	67
4.7 Evaluasi Pengujian	73
BAB V	76
5.1 Kesimpulan	76
5.2 Saran	76
DAFTAR PUSTAKA	78

## DAFTAR GAMBAR

Gambar 2.1 Cara Kerja <i>Machine Learning</i>	9
Gambar 2.2 Skema <i>Decision Tree</i>	11
Gambar 2.3 <i>Bagging Classifier</i>	14
Gambar 2.4 10-folds <i>Cross Validation</i>	16
Gambar 2.5 Logo Python	17
Gambar 2.6 Logo Scikit-Learn	19
Gambar 2.7 Logo Flask	24
Gambar 2.8 Contoh Penggunaan Postman	27
Gambar 3.1 <i>Flowchart</i> Tahapan Penelitian	30
Gambar 3.2 <i>Flowchart</i> Proses Perancangan Sistem Model <i>Machine Learning</i>	33
Gambar 3.3 <i>Flowchart</i> Sistem Pendukung Keputusan untuk Seleksi Konsumen	35
Gambar 3.4 <i>Mockup</i> Halaman <i>Log In</i>	36
Gambar 3.5 <i>Mockup</i> Halaman <i>Dashboard</i>	37
Gambar 3.6 Mockup Halaman Data Konsumen	38
Gambar 3.7 Mockup Halaman Tambah Data	38
Gambar 3.8 <i>Mockup</i> Halaman Formulir Prediksi	39
Gambar 3.9 <i>Mockup</i> Halaman Lihat Prediksi	41
Gambar 4.1 <i>Confusion Matrix</i> Data <i>Test</i> dari Metode <i>Decision Tree</i>	55
Gambar 4.2 Grafik Nilai <i>n_neighbors</i>	57
Gambar 4.3 <i>Confusion Matrix</i> Data <i>Test</i> dari Metode <i>K-Nearest Neighbor</i>	57
Gambar 4.4 <i>Confusion Matrix</i> Data <i>Test</i> dari Metode <i>Support Vector Machine</i>	59
Gambar 4.5 Grafik Nilai <i>n_estimators Decision Tree</i>	60
Gambar 4.6 <i>Confusion Matrix</i> Data <i>Test</i> dari <i>Bagging Classifier Decision Tree</i>	61
Gambar 4.7 Grafik Nilai <i>n_estimators KNN</i>	62
Gambar 4.8 <i>Confusion Matrix</i> Data <i>Test</i> dari <i>Bagging Classifier KNN</i>	62
Gambar 4.9 Grafik Nilai <i>n_estimators SVM</i>	63
Gambar 4.10 <i>Confusion Matrix</i> Data <i>Test</i> dari <i>Bagging Classifier SVM</i>	64
Gambar 4.11 <i>Confusion Matrix</i> Data <i>Test</i> dari <i>Ensemble Method Individu</i>	65
Gambar 4.12 <i>Confusion Matrix</i> Data <i>Test</i> dari <i>Ensemble Method Bagging</i>	66
Gambar 4.13 Tampilan Halaman Login	68

Gambar 4.14 Tampilan Halaman Dashbor	68
Gambar 4.15 Tampilan Halaman Menu Data Konsumen	69
Gambar 4.16 Tampilan Halaman Unggah CSV	69
Gambar 4.17 Tampilan Form Tambah Data	70
Gambar 4.18 Tampilan Form Ubah Data	70
Gambar 4.19 Tampilan Saat Ingin Menghapus Data	71
Gambar 4.20 Tampilan Formulir Prediksi	71
Gambar 4.21 Tampilan Formulir Prediksi Saat Menekan Tombol Cek	72
Gambar 4.22 Tampilan Halaman Lihat Prediksi	72

## DAFTAR TABEL

Tabel 2.1 Contoh Data	6
Tabel 2.2 Model <i>Confusion Matrix</i>	15
Tabel 3.1 Spesifikasi Komputer	31
Tabel 3.2 Data Uji <i>Dummy</i>	40
Tabel 3.3 <i>Confusion Matrix</i>	40
Tabel 3.4 Rancangan Pengujian	42
Tabel 3.5 Rancangan Pengujian (Lanjutan)	43
Tabel 4.1 Pengelompokkan Jangka Waktu	45
Tabel 4.2 Pengelompokkan Rentang Umur	45
Tabel 4.3 Pengelompokkan Rentang Pendapatan	46
Tabel 4.4 Pengelompokkan Rentang Pengeluaran	46
Tabel 4.5 Pengelompokkan Jumlah Pengajuan Konsumen	46
Tabel 4.6 <i>Cross Validation</i> Menggunakan Metode <i>Decision Tree</i>	48
Tabel 4.7 <i>Cross Validation</i> Menggunakan Metode <i>K-Nearest Neighbor</i> (KNN)	49
Tabel 4.8 <i>Cross Validation</i> Menggunakan Metode <i>Support Vector Machine</i>	49
Tabel 4.9 <i>Cross Validation</i> Menggunakan <i>Bagging Classifier Decision Tree</i>	50
Tabel 4.10 <i>Cross Validation</i> Menggunakan <i>Bagging Classifier KNN</i>	50
Tabel 4.11 <i>Cross Validation</i> Menggunakan <i>Bagging Classifier KNN</i> (Lanjutan)	51
Tabel 4.12 <i>Cross Validation</i> Menggunakan <i>Bagging Classifier SVM</i>	51
Tabel 4.13 <i>Cross Validation Ensemble Method</i>	52
Tabel 4.14 Nilai <i>Standard Deviation</i>	52
Tabel 4.15 Akurasi <i>Decision Tree</i> dari Rasio Tertentu	55
Tabel 4.16 Akurasi <i>K-Nearest Neighbor</i> (KNN) dari Rasio Tertentu	58
Tabel 4.17 Akurasi <i>Support Vector Machine</i> (SVM) dari Rasio Tertentu	59

Tabel 4.18 Hasil *Confusion Matrix* Secara Keseluruhan 67

Tabel 4.19 Hasil Pengujian Aplikasi 74

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Pada era globalisasi saat ini, banyak masyarakat yang ingin membeli suatu barang namun karena harganya yang tinggi, mereka tidak dapat membayarnya secara tunai, sehingga banyak masyarakat yang mengajukan peminjaman dana ke pembiayaan. Pembiayaan sendiri adalah perusahaan yang bergerak di bidang keuangan yang menyediakan penawaran dalam bentuk kredit untuk pembelian suatu barang. Kredit merupakan kegiatan peminjaman dana kepada lembaga keuangan yang dilunasi dengan cara mengangsur dalam jangka waktu tertentu.

Saat ini banyak orang yang membeli suatu barang secara kredit. Hal ini tentunya mengakibatkan banyaknya pengajuan permohonan kredit kepada pihak pembiayaan. Menurut survei permintaan dan penawaran pembiayaan perbankan yang diterbitkan oleh Bank Indonesia, pada bulan Juli 2022 permintaan kredit kendaraan bermotor mengalami peningkatan dari bulan sebelumnya yaitu 23,4% menjadi 25,1%. Masalah yang sering dihadapi pihak pembiayaan dalam memberikan kredit kepada calon konsumen adalah kewajiban konsumen yang tidak terpenuhi kepada pembiayaan. Ini dapat disebabkan oleh beberapa faktor, yaitu faktor yang disengaja maupun faktor ekonomi sehingga dapat mempengaruhi kesejahteraan keuangan perusahaan (Junaedi, Siregar, & Nurlaelasari, 2022). Ketika seseorang mengajukan permohonan kredit, pembiayaan harus secara hati-hati mengevaluasi orang yang bersangkutan untuk memastikan bahwa pengajuan kredit tersebut dapat diterima dan tidak berpotensi menjadi negatif konsumen. Negatif konsumen adalah orang yang tidak mampu membayar kembali pinjamannya dengan kata lain pembayaran kredit tersebut tidak berjalan dengan baik atau macet. Jika pihak pembiayaan tidak memperhitungkan hal ini dengan benar, maka dapat merugikan pembiayaan itu sendiri.

Dengan meningkatnya jumlah pengajuan kredit kepada pembiayaan, memungkinkan pihak pembiayaan melakukan kesalahan dalam mengevaluasi orang yang bersangkutan. Hal ini dapat terjadi karena beberapa perusahaan masih melakukan analisis data secara manual, yaitu dengan melakukan tinjauan secara

langsung ke lokasi usaha atau menganalisis informasi dari berkas-berkas yang dibutuhkan oleh pihak pemberian kredit dari pihak peminjam yang berkaitan dengan persyaratan pengajuan kredit seperti penghasilan setiap bulan, jumlah tanggungan keluarga, dan lain sebagainya. Pihak pemberian diharapkan tidak menggunakan waktu yang cukup lama untuk melakukan proses penilaian dan cermat dalam menganalisis data yang diajukan oleh pihak peminjam (Lestari, Rahimah, Army, & Habibie, 2021).

Peran teknologi diperlukan untuk mengekstraksi informasi dari pihak peminjam agar dapat memprediksi kelayakan kredit calon konsumen (Junaedi, Siregar, & Nurlaelasari, 2022). Oleh karena itu, diperlukan sistem yang dapat mendukung pengambilan keputusan pemberian dalam menyeleksi dan mengevaluasi pihak peminjam. Dalam pengembangan sistem ini, akan dilakukan perbandingan antara 3 metode dari *machine learning*, yaitu metode *Decision Tree*, *K-Nearest Neighbor* (KNN), dan *Support Vector Machine* (SVM). Setiap metode akan dilakukan evaluasi terhadap hasil akurasi dari masing-masing metode tersebut untuk membandingkan kinerjanya. Dengan menggunakan *machine learning* diharapkan sistem ini dapat mendukung pengambilan keputusan dan memberikan gambaran besar terhadap pemberian dalam menyeleksi calon konsumen yang berpotensi menjadi negatif konsumen sehingga pihak pemberian dapat meningkatkan hasil keputusan yang lebih baik.

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka identifikasi masalah tersebut adalah kekurangan sistem pendukung keputusan dalam proses seleksi calon konsumen yang mengajukan kredit. Kekurangan sistem ini dapat menimbulkan kesalahan analisis yang dapat disebabkan oleh faktor manusia. Selain itu, kurangnya bahan pertimbangan yang efektif bagi pihak pemberian dalam pengambilan keputusan terkait kredit tersebut. Oleh karena itu, diperlukan pengembangan sebuah aplikasi yang mampu memberikan gambaran besar dari calon konsumen dalam menyeleksi calon konsumen.

### **1.3 Batasan Masalah**

Batasan masalah dalam tugas akhir ini antara lain sebagai berikut.

- a. Data yang diambil adalah data konsumen yang mengajukan kredit di pemberian XYZ dengan jumlah data 1000 pada periode 2022 – 2023.
- b. Pengembangan aplikasi berbasis *website* dilakukan dengan menggunakan *framework* Flask.
- c. Atribut yang digunakan dalam proses pembuatan metode *machine learning* ini adalah umur, pendapatan, pengeluaran, jangka waktu, model konsumen, memiliki guarantor, jumlah pengajuan konsumen, dan keterangan.
- d. Data yang diperoleh tidak memperhatikan kolom kriminal. Kolom ini akan diabaikan dan tidak akan dipertimbangkan dalam analisis data.

### **1.4 Rumusan Masalah**

Rumusan masalah dalam tugas akhir ini adalah bagaimana merancang aplikasi yang dapat mendukung pengambilan keputusan sehingga dapat membantu pemberian untuk mendapatkan gambaran besar dari calon konsumen dalam menyeleksi calon konsumen.

### **1.5 Tujuan Penelitian**

Tujuan yang ingin dicapai dalam tugas akhir ini adalah dengan menerapkan *machine learning* pada aplikasi berbasis *website* untuk mendukung pengambilan keputusan agar dapat menyeleksi konsumen tersebut dapat berpotensi menjadi negatif konsumen atau tidak berpotensi menjadi negatif konsumen.

### **1.6 Manfaat Penelitian**

- a. Bagi pemberian, dapat digunakan untuk menyeleksi konsumen yang mengajukan kredit.
- b. Bagi pembaca, dapat digunakan sebagai acuan untuk bahan pembelajaran dan kajian untuk penelitian selanjutnya.
- c. Bagi penulis, dapat menambah wawasan, pengetahuan, pengalaman, dan kesempatan untuk belajar bagaimana mengimplementasikan *machine*

- learning* untuk menyeleksi konsumen dengan menggunakan bahasa pemrograman Python.
- d. Bagi Universitas Ma Chung, dengan harapan dapat memberikan kontribusi terhadap Universitas Ma Chung sebagai bahan referensi di kalangan mahasiswa yang sedang melaksanakan studi di Universitas Ma Chung.

## 1.7 Luaran Tugas Akhir

Luaran dari tugas akhir ini adalah sebuah *website* untuk membantu pembiayaan dalam menyeleksi konsumen yang mengajukan kredit. *Website* ini khususnya untuk mendukung pengambilan keputusan bagi pembiayaan untuk menentukan apakah konsumen yang mengajukan kredit tersebut berpotensi menjadi negatif konsumen atau tidak berpotensi menjadi negatif konsumen. Tugas akhir ini kemudian akan dilanjutkan dengan publikasi jurnal ilmiah.

## 1.8 Sistematika Penulisan

Sistematika penulisan tugas akhir ini dibagi dalam lima bab, yaitu sebagai berikut.

Bab I              Pendahuluan

Pada bab ini berisi latar belakang, identifikasi masalah, batasan masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, luaran tugas akhir, dan sistematika penulisan yang berkaitan dengan tugas akhir.

Bab II              Tinjauan Pustaka

Pada bab ini berisi landasan-landasan teori yang berkaitan dengan tugas akhir, seperti penjelasan mengenai data, *customer*, kredit, *Machine Learning*, *Decision Tree*, *K-Nearest Neighbor* (KNN), *Support Vector Machine* (SVM), *Bagging Classifier*, *Ensemble Method*, *Confusion Matrix*, *Cross Validation*, Python, Scikit-Learn, HTML, Bootstrap, Flask, Pycharm, MySQL, Postman, dan penelitian terdahulu.

Bab III              Analisis dan Perancangan Sistem

Pada bab ini berisi analisis dan perancangan sistem yang meliputi tahapan penelitian, analisis kebutuhan, desain sistem, implementasi sistem, dan rancangan pengujian.

Bab IV

Hasil dan Pembahasan

Pada bab ini berisi hasil dan pembahasan yang berkaitan dengan tugas akhir. Hasil dan pembahasan ini berupa hasil pengujian yang menunjukkan tingkat akurasi dari metode yang digunakan dan hasil pengujian sistem yang dikembangkan.

Bab V

Kesimpulan dan Saran

Pada bab ini berisi kesimpulan dari hasil pengujian yang telah dilakukan dan saran perbaikan untuk penelitian selanjutnya.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Data**

Data merupakan kumpulan bahan informasi berupa fakta. Data diperoleh melalui observasi yang akurat. Data bersifat mentah karena belum bisa menyimpulkan isi yang terkandung di dalamnya. Data perlu diolah terlebih dahulu untuk menghasilkan informasi yang berguna. Menurut (Hasbiyalloh & Jakaria, 2018) data adalah acuan bukti kenyataan yang dapat dicerminkan dalam suatu objek ber bentuk angka, huruf, simbol, bunyi, atau kombinasinya. Pada penelitian ini, data yang didapat sebanyak 1000 data dimana data ini masih merupakan data kotor yang belum di preprocessing. Untuk gambarannya dapat dilihat pada tabel 2.1.

Tabel 2.1 Contoh Data

No	Nama	Umur	Pendapatan	...	...	...	...	...	...	...	Keterangan
1	A	32	15000000	...	...	...	...	...	...	...	Customer Negatif
2	B	35	16000000	...	...	...	...	...	...	...	Bukan Customer Negatif
...	...	...	...	...	...	...	...	...	...	...	...
1000	C	28	18000000	...	...	...	...	...	...	...	Customer Negatif

#### **2.2 Customer**

*Customer* atau pelanggan adalah orang yang membeli suatu produk yang diproduksi oleh perusahaan secara terus menerus (Kristianto, 2018). Menurut Gasperz, *customer* dapat dibedakan menjadi 3 jenis, yaitu

1. *Internal customer*

*Internal customer* adalah orang yang berada di dalam perusahaan yang berperan penting dalam kinerja perusahaan.

2. *Intermediate customer*

*Intermediate customer* adalah orang yang berperan sebagai perantara dan bukan merupakan pengguna akhir produk.

3. *External customer*

*External customer* adalah orang yang membeli produk dan merupakan pengguna akhir produk tersebut.

### 2.3 Kredit

Pengertian kredit merupakan kegiatan untuk melakukan pembelian atau melakukan pinjaman dengan janji kesepakatan antara dua belah pihak dan pembayaran pinjaman akan dilakukan pada waktu yang telah disepakati bersama (Abdurahman & Riswaya, 2014). Dalam kredit, terdapat prinsip yang digunakan untuk menilai atau menjadi bahan pertimbangan mengenai kelayakan kredit *customer* yang dikenal dengan prinsip 6 C (Abdurahman & Riswaya, 2014). Prinsip 6 C, yaitu

1. *Character*

*Character* terkait dengan kepribadian dari calon pihak peminjam, yaitu kesediannya untuk memenuhi kewajiban yang telah disepakati bersama.

2. *Capacity*

*Capacity* terkait dengan kemampuan dari calon pihak peminjam untuk melunasi semua kewajiban tepat pada waktunya sesuai dengan perjanjian yang telah disepakati dari usaha yang dijalankan.

3. *Capital*

*Capital* terkait dengan modal dari calon pihak peminjam saat mengajukan kredit.

4. *Collateral*

*Collateral* terkait dengan jaminan dari calon pihak peminjam.

5. *Condition of Economic*

*Condition of economic* terkait dengan kondisi ekonomi dari calon pihak peminjam dengan mempertimbangkan kemungkinan yang mempengaruhi kondisi ekonomi di masa yang akan mendatang.

6. *Constrain*

*Constrain* terkait dengan hambatan yang tidak memungkinkan untuk menjalankan usaha di tempat tertentu.

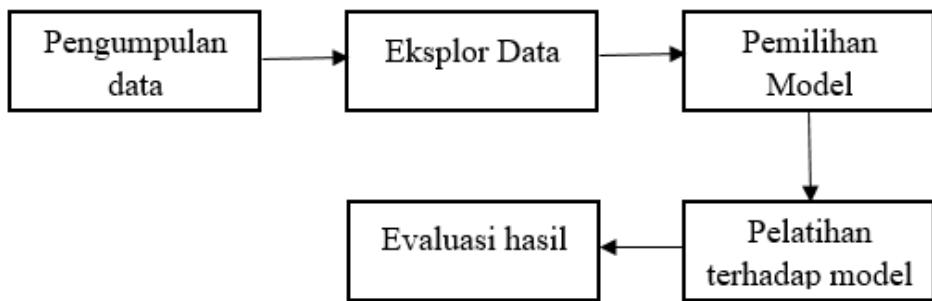
Untuk proses pengajuan kredit pada perusahaan *multifinance*, sebagai langkah pertama setelah mengajukan permohonan kredit, pihak *multifinance* akan

menghasilkan skor kredit. Kemudian setelah skor kredit diperoleh dan calon *customer* dinyatakan memenuhi syarat untuk mendapatkan kredit, calon *customer* dapat meminjam uang dan harus membayar kembali kredit tersebut dengan mencicil bersama dengan bunga yang disepakati sesuai produk pembiayaan yang dipilih dan sesuai dengan waktu yang telah disepakati. Jika *customer* terlambat dalam pembayaran, *customer* akan dikenakan penalti. Demikian pula, jika *customer* ingin mengembalikan pinjaman di tengah masa pelunasan, *customer* akan membayar denda keterlambatan.

## 2.4 Machine Learning

*Machine Learning* (ML) adalah suatu mesin yang dirancang untuk memahami pembelajaran tanpa panduan atau pedoman pengguna. *Machine learning* menggunakan disiplin ilmu lain seperti statistik, matematika, dan data *mining* sehingga mesin ini dapat belajar dengan sendirinya menganalisis data tanpa harus diprogram ulang (Setiadi T., 2022). *Machine learning* dapat mempelajari dan menganalisis data dari data yang diperoleh di awal pengembangan dan dari data saat *machine learning* digunakan untuk dapat melakukan tugasnya. Tugas ini bergantung pada apa yang mereka pelajari seperti metode yang digunakan selama pengembangan. Beberapa matematikawan seperti Adrien Marie Legendre, Thomas Bayes, dan Andrey Markov pertama kali menciptakan istilah *machine learning* pada tahun 1920.

Peran *machine learning* saat ini banyak membantu orang diberbagai bidang misalnya fitur kunci ponsel yang menggunakan pengenalan wajah. Secara garis besar, *machine learning* bersifat *soft computing* yang artinya hasil algoritma tidak sepenuhnya benar. Terdapat beberapa prasyarat yang dibutuhkan dalam pengimplementasian *machine learning*. Prasyarat ini jika tidak terpenuhi maka akan mempengaruhi akurasi dari machine learning tersebut. Dalam *machine learning* seringkali hasil yang muncul tidak akurat karena datanya bias. Pada dasarnya prinsip cara kerja *machine learning* dapat dilihat pada gambar 2.1



Gambar 2.1 Cara Kerja *Machine Learning*

Sumber : (Setiadi T. , 2022)

Terdapat dua teknik dasar dari *machine learning* yang umum digunakan, yaitu *supervised learning* dan *unsupervised learning*.

### 1. *Supervised learning*

*Supervised learning* adalah teknik *machine learning* yang dapat dilakukan pada data yang memiliki informasi atau label yang telah diketahui sebelumnya. Dalam *supervised learning*, algoritma akan mempelajari kembali data-data yang telah diketahui untuk memprediksi data (Permana & S, 2019). Pada *supervised learning*, dataset akan dibagi menjadi dua bagian menjadi data latih dan data uji. Data latih akan digunakan untuk melatih model sedangkan data uji akan digunakan untuk menguji akurasi dari algoritma tersebut. Contoh algoritma yang termasuk dalam teknik *supervised learning* adalah *Decision Tree*, *K-Nearest Neighbor* (KNN), *Support Vector Machine* (SVM), Naïve Bayes, dan lain sebagainya.

### 2. *Unsupervised learning*

*Unsupervised learning* adalah teknik *machine learning* yang dapat dilakukan pada data yang tidak memiliki informasi atau label yang telah ditentukan sebelumnya (Permana & S, 2019). Dalam *unsupervised learning*, algoritma akan dibiarkan menemukan pola yang tersembunyi pada data tanpa label yang jelas. Contoh algoritma yang termasuk dalam teknik *unsupervised learning* adalah *K-Means*, *Fuzzy C-Means*, dan lain sebagainya.

## 2.5 Decision Tree

*Decision tree* atau yang dikenal dengan pohon keputusan adalah sebuah algoritma yang memungkinkan untuk mengolah suatu keputusan dengan pernyataan bersyarat yang berisi cabang-cabang untuk merepresentasikan tahapan keputusan dalam struktur seperti pohon. Pada dasarnya konsep pohon keputusan adalah mengolah data menjadi aturan keputusan yang dapat disederhanakan dalam pengembangan selanjutnya dengan membuang cabang atau aturan yang tidak diperlukan. Atribut pada *decision tree* menunjukkan parameter yang digunakan sebagai perbandingan atau tolak ukur saat membuat *decision tree* (Tahir, 2019).

Pohon keputusan dimulai dengan satu simpul atau titik yang disebut dengan *node*. *Node* ditentukan dengan cara melihat nilai *gain* tertinggi dari atribut yang lainnya. Untuk menghitung nilai *gain* diperlukan nilai *entropy*. Dalam menghitung nilai *entropy* dapat menggunakan rumus sebagai berikut.

$$Entropy(S) = \sum_{i=1}^n -p_i \times \log_2 p_i \quad (2-1)$$

Keterangan :

S : Himpunan kasus

n : Jumlah partisi S

$p_i$  : Proporsi dari  $S_i$  ke S

Untuk menghitung nilai *gain* dapat menggunakan rumus sebagai berikut.

$$Gain(S, A) = Entropy(S) - \left( \sum_{i=1}^n \frac{|A_i|}{|S|} \times Entropy(A_i) \right) \quad (2-2)$$

Keterangan :

S : Himpunan kasus

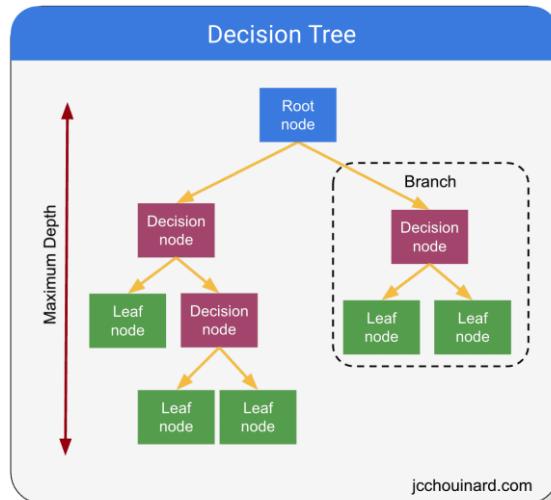
A : Atribut

n : Jumlah partisi atribut A

$|A_i|$  : Jumlah kasus pada partisi ke-i

$|S|$  : Jumlah kasus pada S

*Node* ini kemudian akan bercabang untuk mewakili opsi yang tersedia. Pada pohon keputusan kita dapat mengatur opsi yang berbeda dan mengeksplorasi kemungkinan hasilnya. Selain itu, kita juga dapat melihat potensi risiko dan keuntungan dari setiap opsi yang tersedia. Pohon keputusan memiliki tiga elemen, yaitu simpul akar (*root*), ranting (*branches*), simpul daun (*leaf node*). Simpul akar (*root*) adalah keputusan utama yang akan dibuat. Ranting (*branches*) adalah opsi untuk pilihan keputusan yang berbeda. Simpul daun (*leaf node*) adalah kemungkinan dari hasil keputusan tersebut. Berikut ini adalah contoh skema *decision tree* yang menggambarkan simpul akar (*root*), ranting (*branches*), dan simpul daun (*leaf node*).



Gambar 2.2 Skema *Decision Tree*

Sumber : <https://www.jcchouinard.com/decision-trees-in-machine-learning/>

Pada *decision tree*, terdapat beberapa tahapan algoritma hingga terbentuknya *decision tree* tersebut. Tahapan tersebut adalah sebagai berikut (Yuliana & Pratomo, 2017) :

1. Mempersiapkan data *training*. Data *training* adalah data yang pernah terjadi sebelumnya yang telah dikelompokan ke dalam kelas tertentu (Nurellisa & Fitrianah, 2020).
2. Menentukan akar pohon.

3. Menghitung nilai *entropy*. Titik awalnya adalah perhitungan nilai *entropy* karena hasil perhitungan ini digunakan sebagai acuan dalam perhitungan berikutnya. Rumus *entropy* dapat dilihat di (2-1).
4. Menghitung nilai *gain*. Untuk menentukan atribut sebagai akar pohon dilihat pada nilai *gain* yang tertinggi dari atribut-atribut yang ada. Rumus *gain* dapat dilihat di (2-2).
5. Ulangi langkah kedua sampai setiap cabang terisi.
6. Proses partisi dalam *decision tree* berhenti ketika semua cabang dari simpul N menerima kelas yang sama.

Terdapat beberapa manfaat saat menggunakan *decision tree*, yaitu

1. Saat menggunakan *decision tree*, *decision tree* dapat mengikuti pemikiran manusia saat mengambil keputusan yang ada sehingga mudah dipahami.
2. Risiko-risiko setiap opsi yang ada dapat dianalisis dengan baik karena memakai probabilitas.
3. Dapat mempertimbangkan kemungkinan untung dan rugi dari setiap opsi yang tersedia dimana setiap opsi ini dijelaskan secara transparan sehingga membiarkan setiap orang ikut terlibat untuk mengetahui pertimbangan-pertimbangan apa yang diperhitungkan dari keputusan yang dipilih.

## 2.6 K-Nearest Neighbor

*K-Nearest Neighbor* (KNN) adalah sebuah algoritma klasifikasi *machine learning* paling sederhana yang memiliki acuan membandingkan data terdekat dalam menentukan kelas data barunya, dimana algoritma ini akan mengelompokkan data berdasarkan kesamaannya dengan data lain. KNN bersifat *lazy learning* yang berarti dalam membuat model algoritma ini tidak menggunakan fase *training*. KNN dapat dilakukan dengan mencari kelompok k objek terdekat pada data *training* dengan objek pada data *testing*. Untuk menentukan jarak antara dua objek dapat menggunakan rumus jarak Euclidean dengan rumus sebagai berikut.

$$d = \sqrt{\sum_{i=1}^n (x_{2i} - x_{1i})^2} \quad (2-3)$$

Keterangan :

$d$  : Jarak Euclidean

$x_2$  : Data *testing*

$x_1$  : Data *training*

Berikut ini adalah algoritma KNN (Tedi, Siregar, Masruriyah, & Juwita, 2020):

1. Tentukan parameter k.
2. Hitung jarak dengan menggunakan rumus jarak Euclidean yang dapat dilihat di (2-3).
3. Urutkan objek tersebut mulai dari jarak Euclidean terkecil hingga terbesar.
4. Klasifikasi KNN.

## 2.7 *Support Vector Machine (SVM)*

*Support Vector Machine (SVM)* adalah sebuah algoritma *machine learning* yang digunakan untuk klasifikasi. Selain digunakan untuk klasifikasi, SVM juga dapat digunakan untuk regresi. SVM bermula dari permasalahan mengenai pengelompokan antara dua kelas yang memerlukan data *training* dengan data positif dan data negatif. SVM memiliki tujuan untuk mencari *hyperlane* (pemisah) yang terbaik dengan memaksimalkan jarak antar kedua kelas.

Pada tahun 1992, SVM dikenalkan oleh Vapnik sebagai konsep dalam bidang pola *recognition* yang dapat digunakan untuk memilih model secara otomatis. SVM termasuk algoritma yang sangat baik digunakan untuk prediksi karena algoritma ini dapat mengurangi kesalahan dalam melakukan klasifikasi dan anomali data pada saat training data. SVM dibagi menjadi 2 jenis, yaitu SVM Linear dan SVM Non-Linear. SVM linear digunakan pada kumpulan data yang dapat diklasifikasikan menjadi dua kelas secara linear atau garis lurus sedangkan SVM Non-linear digunakan pada kumpulan data yang tidak dapat diklasifikasikan menjadi dua kelas secara linear. Pada kasus non-linear dapat menggunakan fungsi kernel dimana fungsi kernel ini digunakan untuk menggambarkan dimensi awal himpunan data ke dimensi yang relatif lebih tinggi. Ada beberapa fungsi kernel, beberapa diantaranya, yaitu (Fadilah, Agfiannisa, & Azhar, 2020)

## 1. Kernel Gaussian Radial Basic Function (RBF)

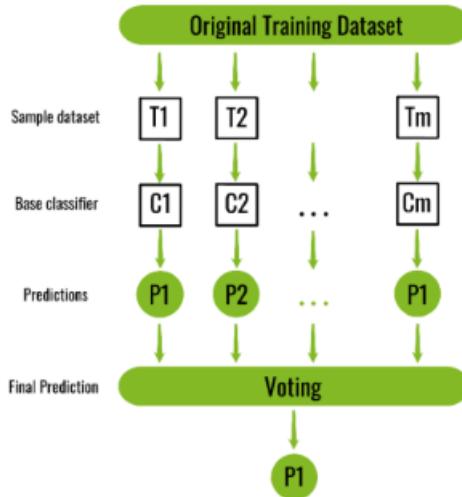
$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (2-4)$$

## 2. Kernel Polynomial

$$K(x_i, x_j) = \exp\left((x_i \cdot x_j) + c\right)^d \quad (2-5)$$

### 2.8 Bagging Classifier

*Bagging* singkatan dari *Bootstrap Aggregating* merupakan teknik *machine learning ensemble* yang menggabungkan prediksi beberapa model untuk meningkatkan kinerja sistem secara keseluruhan. *Bagging classifier* menggunakan *resampling bootstrap* untuk menghasilkan beberapa subset berbeda dari data training lalu melatih model terpisah pada setiap subset tersebut. Prediksi akhir akan dibuat dengan menggabungkan prediksi dari semua model (Dey, 2023).



Gambar 2.3 *Bagging Classifier*

Sumber : (Dey, 2023)

### 2.9 Ensemble method

*Ensemble* berarti sekelompok elemen yang dilihat secara keseluruhan bukan secara individual (Apurva, 2021). *Ensemble* membuat beberapa model dan

menggabungkannya. Salah satu metode *ensemble* dasar adalah *max voting*. *Max voting* digunakan untuk masalah klasifikasi dimana metode ini akan bekerja dengan membangun beberapa model secara mandiri dan mendapatkan hasilnya masing-masing. Sebagai contoh terdapat tiga model klasifikasi algoritma *machine learning*, kemudian tiga model tersebut digabungkan menggunakan *library* Scikit-Learn *VotingClassifier*. *VotingClassifier* bertugas mengumpulkan prediksi dari masing-masing model dan mengembalikan hasil dengan jumlah suara terbanyak sebagai output akhir.

## 2.10 Confusion Matrix

*Confusion matrix* adalah metode untuk menghitung akurasi di dalam konsep data mining (Rahmianti, 2022). *Confusion matrix* berbentuk tabel yang berisi nilai prediksi dan nilai sebenarnya. Berikut ini adalah model tabel dari *confusion matrix*.

Tabel 2.2 Model *Confusion Matrix*

<i>Confusion Matrix</i>	<i>Positive</i>	<i>Negative</i>
<i>Positive</i>	<i>True Positive (TP)</i>	<i>False Negative (FN)</i>
<i>Negative</i>	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>

Dapat dilihat pada tabel di atas terdapat empat istilah untuk model *confusion matrix*, yaitu *true positive*, *false negative*, *false positive*, dan *true negative*. *True positive* merupakan istilah untuk jumlah data positif dan diklarifikasi benar. *False negative* merupakan istilah untuk jumlah data negative dan diklarifikasi salah. *False positif* merupakan istilah untuk jumlah data positif dan diklarifikasi salah. *True negative* merupakan istilah untuk jumlah data negative dan diklarifikasi benar (Rahmianti, 2022). Dari model *confusion matrix* juga didapatkan empat persamaan, yaitu (Nirwana, Siregar, & Rahmat, 2022)

1. *Recall* digunakan untuk melihat seberapa baik model mengklasifikasikan ke las positif dengan benar.

$$Recall = \frac{TP}{TP + FN} \quad (2-6)$$

2. *Precision* digunakan untuk melihat seberapa baik model mengklasifikasikan kelas yang diprediksi positif dengan benar.

$$Precision = \frac{TP}{TP + FP} \quad (2-7)$$

3. *Accuracy* digunakan untuk melihat seberapa baik model mengklasifikasikan dengan benar.

$$Accuracy = \frac{\text{Jumlah prediksi benar}}{\text{Jumlah total prediksi}} \quad (2-8)$$

4. *F-Measure* digunakan untuk melihat keseimbangan antara precision dan recall. Semakin tinggi nilai F-Measure maka model semakin baik.

$$F - Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2-9)$$

## 2.11 Cross Validation

*Cross validation* (CV) adalah suatu metode dengan memisahkan data menjadi 2 bagian, yaitu data *train* dan data *test* yang digunakan untuk mengevaluasi kinerja dari suatu model *machine learning*. Proses *cross validation* umumnya dilakukan dengan membagi dataset menjadi beberapa bagian yang sama besar.

	Dataset dibagi menjadi 10 bagian secara random (acak)											Akurasi
	10%	10%	10%	10%	10%	10%	10%	10%	10%	10%	100%	
Percobaan 1	10%											a1
Percobaan 2		10%										a2
Percobaan 3			10%									a3
Percobaan 4				10%								a4
Percobaan 5					10%							a5
Percobaan 6						10%						a6
Percobaan 7							10%					a7
Percobaan 8								10%				a8
Percobaan 9									10%			a9
Percobaan 10										10%		a10

Gambar 2.4 10-folds Cross Validation

Sumber : (Rilvani, Trisnawan, & Santoso, 2019)

Sebagai contoh pada gambar 2.4 dataset awal dibagi menjadi 10 bagian yang sama besar. Kemudian salah satu bagian akan dipilih menjadi data *test* sementara 9 bagian lainnya digunakan sebagai data *train*. Data yang dipilih menjadi data *test* dilambangkan dengan simbol kotak berwarna kuning. Langkah ini diulang sebanyak 10 kali dengan setiap bagian bergantian sebagai data *train* maupun data *test*. Akhirnya hasil evaluasi tersebut akan mendapatkan rata-rata akurasi sebagai performa model secara keseluruhan. Pada *cross validation*, metode 10 *folds cross validation* telah terbukti menjadi metode yang umum digunakan dan memberikan performa model yang lebih stabil dalam berbagai penelitian (Rilvani, Trisnawan, & Santoso, 2019).

## 2.12 Python

Python adalah salah satu bahasa pemrograman yang berorientasi objek dan banyak digunakan dalam pengembangan *software*. Pengembangan *software* ini mencakup berbagai bidang, mulai dari pengembangan *website*, *data science*, hingga *machine learning*. Python dapat dijalankan di berbagai *platform* dan dapat terintegrasi baik dengan semua jenis sistem yang ada (AWS, 2023).



Gambar 2.5 Logo Python

Sumber : <https://www.python.org/>

Python pertama kali diciptakan oleh seorang *programmer* asal Belanda bernama Guido Van Rossum yang lahir pada tanggal 31 Januari 1956. Pada tahun 1989, Guido Van Rossum memulai pengembangan bahasa pemrograman Python di *Centrum Wiskunde and Informatica* (CWI) sebagai proyek hobi untuk menyibukkan diri selama Natal berlangsung. Nama Python terinspirasi dan diambil dari acara televisi “*Monty Python’s Flying Circus*” yang disiarkan oleh BBC pada tahun 1980-an dimana acara tersebut adalah acara komedi dan Guido Van Rossum adalah penggemar acara tersebut.

Saat Guido Van Rossum menciptakan bahasa pemrograman Python, Guido Van Rossum terinspirasi dari bahasa pemrograman ABC. Bahasa pemrograman ABC sendiri adalah bahasa pemrograman untuk komputasi personal yang dirancang oleh CWI sebagai bahasa pemrograman dengan tujuan untuk pengajaran dan pembuatan *prototype*. Meskipun bahasa pemrograman Python bersifat *open source* atau terbuka yang berarti memungkinkan banyak orang untuk berkontribusi dalam pengembangan bahasa pemrograman tersebut termasuk memperbaikinya, Guido Van Rossum adalah penulis utama bahasa pemrograman tersebut.

Bahasa pemrograman Python memiliki banyak kelebihan yang dapat membantu *developer* terkait dalam pengembangan *software*, yaitu

1. Python dapat digunakan di berbagai sistem operasi komputer seperti Windows, MacOS, Linux, dan Unix.
  2. *Developer* dapat dengan mudah mengoperasikan bahasa pemrograman Python karena Python memiliki sintaks dasar seperti bahasa Inggris yang mudah dibaca dan dipahami.
  3. Python memiliki baris kode yang lebih sedikit dibandingkan dengan bahasa pemrograman yang lain dimana hal ini membuat *developer* lebih produktif.
  4. Python memiliki banyak *library* dimana hal ini membuat *developer* tidak perlu membuat kode lagi dari awal dan *library* Python ini selalu dapat digunakan kembali oleh *developer*. Berikut ini adalah beberapa contoh dari *library* Python.
    - Matplotlib
- Matplotlib adalah *library* Python yang berguna untuk menyajikan data dalam bentuk diagram dengan kualitas yang tinggi. Matplotlib memungkinkan untuk memvisualisasikan data dengan berbagai macam diagram seperti diagram batang, diagram lingkaran, diagram garis, dan lain-lain.
- Pandas
- Pandas adalah *library* Python yang berguna untuk menyajikan struktur data dan menganalisis data. Pandas memungkinkan untuk membuat tabel, memfilter, mengelompokkan data, dan lain sebagainya.
- Numpy

Numpy adalah *library* Python yang mendukung dalam hal *scientific computer*. Numpy berguna untuk membuat array dan mengoperasikan aljabar linear seperti vektor dan matriks.

- *Request*

*Request* adalah *library* Python yang berguna untuk pengembangan *website* dimana *library* ini digunakan untuk mengirim *request* HTTP, berinteraksi dengan *website*, dan lain sebagainya.

5. Terdapat banyak komunitas Python yang aktif dan banyak sumber yang berguna di internet terkait bahasa pemrograman Python seperti contoh banyak video pembelajaran Python yang ada internet.

### 2.13 Scikit-Learn

Scikit-Learn adalah *library* Python yang digunakan dalam kebutuhan *data science* seperti *machine learning*. Scikit-Learn pertama kali dikembangkan sebagai proyek *Google Summer of Code*. Proyek ini dikembangkan pada tahun 2007 oleh David Cournapeau. Seorang bernama Matthieu Brucher kemudian bergabung dalam proyek tersebut. Saat bergabung, Matthieu Brucher menggunakan proyek tersebut sebagai bagian dari tesisnya. Pada tahun 2010, proyek ini dipimpin oleh empat orang dari INRIA yang bernama Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, dan Vincent Michel. Mereka merilis Scikit-Learn pertama kali kepada publik pada tanggal 1 Februari tahun 2010 (Scikit-Learn, 2013).



Gambar 2.6 Logo Scikit-Learn

Sumber : <https://scikit-learn.org/stable/about.html>

Dasar-dasar dalam machine learning menggunakan Scikit-Learn antara lain :

1. Memuat data set.
2. Split data atau membagi data untuk data train dan data test.

3. Memilih dan membuat model.
4. Memprediksi data.

Scikit-Learn sebagai *library* Python memiliki beberapa keunggulan. Beberapa keunggulan Scikit-Learn, yaitu

1. Scikit-Learn dapat bekerja dengan *library* Python yang berbeda seperti NumPy dan SciPy.
2. Scikit-Learn dapat dengan cepat membandingkan kumpulan data yang berbeda dan dapat dengan mudah dalam penggunaan API.

Scikit-Learn menyediakan beberapa fitur algoritma untuk *data science* terkait *machine learning* seperti regresi, naive bayes, *classification*, *decision tree*, dan lain sebagainya. Pada penelitian ini, Scikit-Learn akan digunakan untuk mengimplementasikan *decision tree*, *K-Nearest Neighbor* (KNN), dan *Support Vector Machine* (SVM), dimana pada *library* ini terdapat beberapa fungsi, yaitu

- Fungsi `train_test_split` yang berfungsi untuk membagi data train dengan data test. Berikut ini adalah contoh pengimplementasian dari fungsi ini.

```
x_train, x_test, y_train, y_test = train_test_split(x, y,  
test_size=0.20)
```

Pada fungsi diatas `x_train` adalah data `x` yang ingin dilatih, `x_test` adalah data `x` yang ingin ditest, `y_train` adalah variable dependen yang dilatih, `y_test` adalah variable dependen yang ditest, `x` adalah variable independen, `y` adalah variable dependen, `test_size` menunjukkan data test yang diambil sebesar 20%.

- Fungsi `DecisionTreeClassifier` untuk membuat model klasifikasi *decision tree* dari data train. Berikut ini adalah contoh pengimplementasian dari fungsi ini.

```
classifier = DecisionTreeClassifier(criterion="entropy",  
max_depth=4)
```

Pada fungsi diatas *criterion* adalah perhitungan menggunakan rumus entropy, *max\_depth* adalah jumlah kedalaman branch sebanyak 4.

- Fungsi fit digunakan untuk melatih classifier pada data train. Berikut ini adalah contoh pengimplementasian dari fungsi ini.

```
classifier.fit(x_train, y_train)
```

- Fungsi untuk menyimpan hasil prediksi yang diperoleh dari fungsi *train\_test\_split*. Berikut ini adalah contoh pengimplementasian dari fungsi ini.

```
y_pred = classifier.predict(x_test)
```

- Fungsi *accuracy\_score* untuk menampilkan akurasi dari data test. Berikut ini adalah contoh pengimplementasian dari fungsi ini.

```
accuracy_score(y_test, y_pred)
```

- Fungsi *confusion\_matrix* berfungsi untuk menghitung akurasi dari hasil prediksi. Berikut ini adalah contoh pengimplementasian dari fungsi tersebut.

```
confusion_matrix(y_test, y_pred)
```

- Fungsi *KNeighborsClassifier* untuk membuat model klasifikasi *K-Nearest Neighbor* (KNN). Berikut ini adalah contoh pengimplementasian dari fungsi ini.

```
knn = KNeighborsClassifier(n_neighbors=3)
```

Pada fungsi di atas *n\_neighbors* adalah jumlah tetangga sebanyak 3.

- Fungsi SVC untuk membuat model klasifikasi *Support Vector Machine* (SVM). Berikut ini adalah contoh pengimplementasian dari fungsi ini.

```
svm = SVC()
```

## 2.14 HTML

HTML adalah bahasa markup yang digunakan untuk membuat halaman sebuah *website* (Benefita, 2023). HTML telah berkembang dengan berbagai versi pembaruannya. Setiap versi telah memungkinkan penggunanya untuk membuat halaman *website* dengan cara yang lebih mudah dan efisien. HTML sendiri terdiri dari simbol dan kombinasi teks yang disimpan dalam sebuah file berbentuk .html. HTML pertama kali dibuat oleh Tim Berners-Lee, yaitu seorang ahli fisika di lembaga penelitian CERN di Swiss pada tahun 1991. HTML ini terdiri dari 18 tag HTML. Saat itu Tim Berners-Lee membuat HTML untuk mempermudah para ilmuwan lain mengakses dokumen satu dengan yang lainnya.

Secara umum HTML memiliki komponen-komponen utama, yaitu *tag*, elemen, dan atribut. Masing-masing komponen memiliki fungsi tersendiri. Untuk penjelasan lebih lanjut mengenai fungsinya, mari simak uraian di bawah ini.

### 1. Tag

*Tag* adalah suatu teks yang menentukan bagaimana menampilkan suatu konten yang dapat dibaca oleh *website*. *Tag* terdiri dari pembuka dan penutup yang berpasangan dimana diantara pembuka dan penutup tersebut terdapat konten. Namun, tidak semua *tag* berpasangan. Ada beberapa *tag* yang tidak mempunyai penutup seperti *tag* br. *Tag* ditandai dengan tanda lebih kecil (<) dan tanda lebih besar (>). Pada penutup terdapat garis miring setelah tanda lebih kecil (</). Saat ini sudah terdapat lebih dari 250 *tag* dimana masing-masing *tag* memiliki fungsi yang berbeda-beda.

### 2. Elemen

Elemen adalah suatu komponen yang terdiri dari *tag* pembuka, konten, dan *tag* penutup. Secara singkat elemen adalah semua kode yang ditulis dalam *website*. Elemen dalam HTML memiliki keterkaitan satu dengan yang lainnya.

### 3. Atribut

Atribut merupakan informasi tambahan yang ada pada elemen dan terletak di dalam *tag*. Atribut dapat ditambahkan lebih dari satu. Nilai atribut dalam HTML ditandai dengan tanda petik. HTML juga dapat membuat sebuah komentar menggunakan *tag* <!-- --> untuk mencatat atau menjelaskan bagian-bagian penting yang ada di halaman *website*.

## 2.15 Bootstrap

Bootstrap adalah suatu *framework open source* HTML, CSS, dan JavaScript untuk memperindah tampilan *website* dengan cepat dan mudah. Bootstrap dapat membuat tampilan *website* menjadi responsif dan *mobile-friendly*. *Framework open source* ini dibuat pada tahun 2011 oleh Mark Otto dan Jacob Thornton yang bekerja di Twitter. Itu sebabnya Bootstrap dulu disebut sebagai Twitter Blueprint.

Dalam perkembangannya, *framework* ini telah mengeluarkan sebanyak 20 versi. Bootstrap memiliki beberapa fungsi utama yang penting untuk *developer* dalam mengembangkan *website* (Pradana, 2023), antara lain :

### 1. Dapat mengembangkan *website mobile-friendly*

Bootstrap dikenal sebagai *framework mobile-first*. *Framework* ini memiliki sistem *grid*, yaitu fitur untuk menciptakan *website* yang bekerja secara sempurna di *mobile device*.

### 2. Menjadikan *website* lebih interaktif

Bootstrap yang memiliki fitur *plugin JQuery* dapat membuat *website* menjadi lebih interaktif karena dapat berkreasi dengan tampilan seperti *pop-up*, *carousel*, dan lain-lain.

### 3. Dapat menambahkan elemen *website* lebih mudah

Dengan adanya elemen bawaan, *developer* dapat memanfaatkan langsung elemen tersebut dengan menambahkan menu seperti *dropdown*, *navigasi*, dan lain-lain.

### 4. Dapat mempercepat resize gambar

Bootstrap menyediakan *class* untuk gambar agar responsif sehingga ukuran gambar dalam halaman *website* akan menyesuaikan ukuran di *device user* secara otomatis. *Class* tersebut adalah *class.img-responsive*. Jadi, *developer*

- tidak perlu lagi membuat *file* gambar dalam banyak ukuran untuk membuat tampilan gambar responsif.
5. Tampilan *website* lebih modern  
Tampilan *website* dengan menggunakan *framework* Bootstrap tampak lebih modern dengan tampilan yang relatif minimalis, *fresh*, dan *clean*.
  6. Tidak membuang waktu *developer*  
Dengan *framework* Bootstrap, *developer* tidak perlu membuang waktu membuat semua kode dari awal untuk mengembangkan *website*. Cukup dengan menggunakan fitur bawaan Bootstrap maka *website* akan menampilkan halaman yang menarik.  
Fungsi-fungsi diatas terbukti memudahkan *developer* dalam hal mengembangkan *website*.

## 2.16 Flask

Flask merupakan *framework* dari Python yang berfungsi untuk pengembangan *website*. Flask pertama kali diciptakan oleh Pocoo pada tahun 2004 dimana flask dibangun dengan dasar dari Werkzeug dan Jinja2. Flask dikategorikan sebagai *framework* mikro karena tidak bergantung pada *library* tambahan sehingga dapat digunakan dengan lebih ringan (Alexandromeo, 2022).



Gambar 2.7 Logo Flask

Sumber : <https://www.sviluppomania.com/en/install-and-test-flask/>

Sebagai *framework*, flask memiliki beberapa kelebihan, yaitu

1. Keuntungan pertama yang bisa dirasakan adalah performa penggunaan *framework*. Flask adalah *framework* pengembangan *web* yang sangat ringan karena desain modularnya yang sangat sederhana.
2. *Framework* ini cocok untuk pemula karena menggunakan bahasa yang sangat mudah dipahami bagi pemula.
3. Flask memiliki fungsionalitas dasar yang mendukung penambahan *library*, modul, dan *plugin* lain sebagai ekstensi tambahan dimana dengan keunggulan tersebut akan memudahkan pengguna untuk membangun pengembangan *web*.
4. Dengan sifat *open source* pengguna dapat menggunakan secara gratis dan pengguna dapat membuat *website* untuk keperluan bisnis.
5. *Framework* ini memungkinkan untuk pengembangan *web* yang mencakup pengembangan *front-end* dan *back-end* sehingga memungkinkan untuk mengembangkan *web* yang lebih fleksibel dalam satu *framework*.
6. Dapat menggunakan fitur bawaan seperti pengembangan *server* dan *timing*, dukungan integrasi pada pengujian unit, *secure cookies*, ReSTful *Request dispatching*, dan lain sebagainya.

## 2.17 PyCharm

PyCharm adalah salah satu IDE Python yang dikembangkan oleh JetBrains. PyCharm pertama kali dipublikasi pada tahun 2010. PyCharm digunakan untuk pengembangan *web* dan salah satu integrasi terlengkap untuk *development* dengan menggunakan bahasa pemrograman Python. PyCharm dapat dengan mudah beroperasi di berbagai *platform* seperti Windows, Linux, dan macOS. Berikut ini adalah fitur yang dimiliki oleh PyCharm (Appkey, 2021).

### 1. Intelligent Editor Code

PyCharm sebagai editor *code* mendukung berbagai macam fasilitas dalam hal penulisan *code* Python yang berkualitas tinggi.

### 2. Ketersediaan *tool* yang terintegrasi

PyCharm mendukung alat integrasi yang berguna untuk meningkatkan produktivitas kode hingga menyediakan fasilitas untuk proyek *data science*.

### 3. *Data Science* dan *Machine Learning*

PyCharm mendukung banyak *library* untuk *data science* dan *machine learning* seperti contoh Matplotlib, SciPy, dan lain-lain.

4. *Debugging* dan pengujian terintegrasi

PyCharm mendukung program *debug* dan pengujian. *Debug* dilakukan untuk menemukan dan memperbaiki *bug* di setiap baris kode yang telah dijalankan.

Selain fitur di atas PyCharm memiliki beberapa kelebihan, yaitu

1. Instalasi pada Pycharm mudah dilakukan.
2. Fasilitas yang tersedia lengkap sehingga *developer* dalam hal pengembangan kode dapat melakukannya dengan lebih cepat.
3. PyCharm relative mudah digunakan oleh *developer*.
4. PyCharm adalah aplikasi yang produktif untuk pengembangan dalam menggunakan bahasa pemrograman Python.
5. *Plugin* yang tersedia berbagai macam.

## 2.18 MySQL

MySQL adalah salah satu *database management system* (DBMS) yang bersifat *open source* yang menggunakan bahasa SQL sebagai perantara antara aplikasi dan server *database*. MySQL dikembangkan oleh Michael Monty Widenius dan TcX yang merupakan perusahaan *software* asal Swedia. MySQL dikembangkan dari proyek UNIREG yang merupakan pengembangan lanjutan dari proyek tersebut. Pada saat itu UNIREQ tidak terlalu kompatibel dengan *database* yang digunakan di *web* sehingga TcX mencari opsi lain dan menemukan *software* yang bernama miniSQL atau mSQL yang dikembangkan oleh David Hughes. Namun, miniSQL atau mSQL tidak mensupport *indexing* sehingga masih belum kompatibel dengan kebutuhan yang diperlukan oleh TcX. Alhasil, pengembang UNIREG, mSQL, dan TcX melakukan kolaborasi. Tujuan dari kolaborasi ini adalah untuk membangun sistem database yang baru. Hingga akhirnya, MySQL dirilis pada tahun 1995 dan berada di bawah Oracle hingga saat ini (K, 2022).

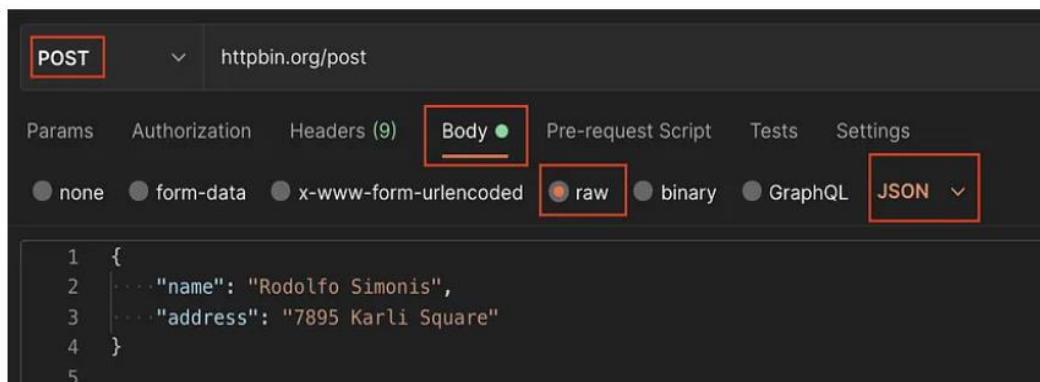
SQL adalah bahasa pemrograman yang berhubungan dengan tabel dan *database* serta antar *database*. SQL mempunyai tiga macam bentuk, yaitu

1. *Data Definition Language* (DDL) yang berfungsi untuk mendefinisikan data atau membuat dan mengubah objek di dalam *database*.

2. *Data Manipulation Language* (DML) yang berfungsi untuk menambah atau mengubah isi data dari tabel yang sudah ada.
3. *Data Control Language* (DCL) yang berfungsi untuk memberikan akses *database* dan mengalokasikan ruang *database*.

## 2.19 Postman

*Postman* adalah suatu platform *Application Programming Interface* (API) yang memungkinkan penggunanya untuk membuat dan menggunakan API. *Postman* pertama kali dibuat oleh Abhinav Asthana pada tahun 2012 sebagai proyek ekstensi *Google Chrome* untuk membantu penggunanya berinteraksi dengan API. Seiring berjalanannya waktu, *postman* telah beralih menjadi aplikasi *native*. *Postman* memberikan *experience* dengan berbagai fitur dan memiliki cara kerja yang mudah serta sederhana. Postman bekerja dengan cara mengirim *request* API ke server *website* dan akan menerima *response* dari server tersebut (Sudirman, 2022). Pada gambar 2.8 adalah salah satu contoh penggunaan *postman* yang paling umum, yaitu mengirim *request* dalam bentuk format *body json* ke API.



Gambar 2.8 Contoh Penggunaan *Postman*

Sumber : <https://medium.com/apis-with-valentine/all-types-of-post-requests-with-postman-1cd2307ed6aa>

*Postman* memiliki tampilan yang *friendly* bagi penggunanya sehingga memudahkan pengguna untuk mengirim *request* dengan mengisi data yang diperlukan ke dalam *body* dan setelah itu dapat mengklik tombol *send* untuk

memprosesnya. *Postman* juga mendukung fitur untuk merapikan format tulisan secara otomatis sehingga pengguna tidak kebingungan untuk melihatnya.

## 2.20 Penelitian Terdahulu

Penelitian mengenai prediksi kelayakan pemberian kredit dengan menggunakan metode *decision tree* sebelumnya dilakukan oleh (Wahyuningsih & Utari, 2018). Pada penelitian ini dilakukan perbandingan antara tiga metode, yaitu metode *K-Nearest Neighbor*, metode *Naïve Bayes*, dan metode *Decision Tree*. Penelitian tersebut dilakukan dengan cara membagi dua data yang diperoleh menjadi data latih sekaligus data uji dan data evaluasi. Dari data latih hasil pengujinya akan dibandingkan dengan menggunakan *k-fold cross validation* dengan  $k = 10$ . Dari hasil penelitiannya dikatakan bahwa metode *decision tree* memiliki tingkat akurasi yang paling baik yaitu sebesar 92,21 % dibandingkan dengan metode yang lainnya.

Penelitian lainnya dilakukan oleh (Setiadi & Lareno, 2015) mengenai penerapan algoritma *decision tree* untuk penilaian agunan pengajuan kredit. Pada penelitian ini telah dilakukan perbandingan antara tiga metode yang sama seperti penelitian yang dilakukan oleh (Wahyuningsih & Utari, 2018). Dari penelitian ini diketahui bahwa *decision tree* C4.5 memiliki nilai akurasi yang paling tinggi, yaitu 71% dibandingkan dengan metode yang lainnya sehingga diperoleh hasil bahwa *decision tree* C4.5 dapat memprediksi lebih akurat dan dapat digunakan untuk penilaian agunan pengajuan kredit.

Kemudian terdapat penelitian yang dilakukan oleh (Rani, 2016) mengenai klasifikasi nasabah menggunakan algoritma C4.5 sebagai dasar pemberian kredit. Pada penelitian ini disebutkan bahwa metode *decision tree* dapat mengidentifikasi kelayakan kredit dengan baik dimana algoritma C4.5 dianggap sebagai algoritma yang dapat membantu melakukan pengelompokan data.

Setelah itu terdapat penelitian yang dilakukan oleh (Rahmianti, 2022) mengenai analisis kelayakan pemberian kredit koperasi dengan metode *data mining decision tree*. Dari penelitian didapatkan hasil bahwa penerapan algoritma C4.5 terhadap data calon kreditur koperasi dengan menggunakan perangkat lunak

*rapidminer* memperoleh nilai akurasi sebesar 94,27 % dimana dari hasil perhitungan manual mempunyai tingkat akurasi sebesar 98 %.

Penelitian yang dilakukan oleh (Akbar & Samsudin, 2016) mengenai sistem pendukung keputusan kelayakan pemberian kredit menggunakan metode *decision tree* dalam studi kasus Bank PD. BPR Gemilang Tembilahan juga menyebutkan bahwa dengan menggunakan metode *decision tree* dapat membantu kinerja dari proses seleksi dimana sistem pendukung keputusan ini dapat dijadikan solusi baru dalam proses seleksi keputusan kelayakan pemberian kredit.

Penelitian yang dilakukan oleh (Handayani, Wahyono, Trianto, & Permana, 2021) mengenai prediksi tingkat risiko kredit dengan data *mining* menggunakan algoritma *decision tree* C.45 disebutkan bahwa algoritma C.45 diimplementasikan dan dianalisa menggunakan aplikasi WEKA. Dari hasil penelitian yang telah dilakukan didapatkan akurasi sebesar 79% dengan menggunakan *confusion matrix*.

Selain itu, terdapat penelitian mengenai implementasi algoritma *decision tree* untuk klasifikasi produk laris yang dilakukan oleh (Nasrullah, 2021). Pada penelitian ini dilakukan pengujian akurasi dari algoritma C4.5 untuk melakukan klasifikasi produk laris. Dari penelitian tersebut didapatkan hasil akurasi sebesar 90% dan dapat disimpulkan bahwa model klasifikasi data mining algoritma *decision tree* C4.5 akurat dalam melakukan klasifikasi untuk produk laris.

Penelitian yang dilakukan oleh (Widjaja, Prilianti, & Setiawan, 2014) mengenai sistem pendukung keputusan pemberian kredit rumah menggunakan *analytical hierarchy process* berbasis *web*. Pada penelitian ini telah dikembangkan aplikasi untuk melakukan proses penyeleksian terhadap calon nasabah agar dapat menggunakan cara yang lebih mudah dimana metode yang digunakan dalam proses pembuatan aplikasi ini adalah metode *Analytical Hierarchy Process*. Dari hasil uji coba program didapatkan nilai persentasi tingkat validasi sistem sebesar 85%.

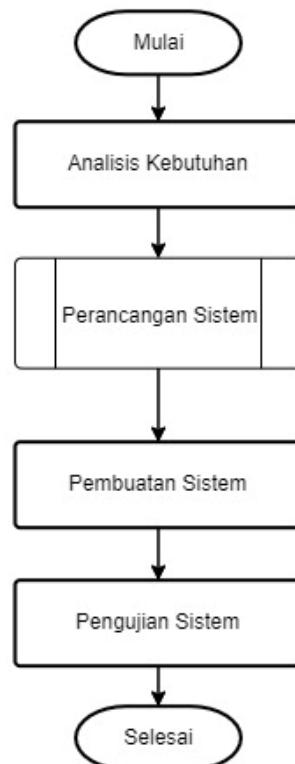
Penelitian yang dilakukan oleh (Bangun, Mawengkang, & Efendi, 2022) mengenai metode algoritma support vector machine (SVM) linier dalam memprediksi kelulusan mahasiswa menunjukkan bahwa hasil akurasi yang dihasilkan sebesar 90% dengan rasio data 70:30.

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **3.1 Tahapan Penelitian**

Tahapan penelitian dalam perancangan sistem untuk menyeleksi konsumen yang mengajukan kredit tertera pada gambar 3.1 sebagai berikut.



Gambar 3.1 *Flowchart* Tahapan Penelitian

Tahapan awal dimulai dengan menganalisis kebutuhan yang diperlukan terkait dengan proses perancangan sistem ini. Setelah itu adalah tahapan perancangan sistem yang berguna dalam pembuatan sistem kedepannya. Pada tahapan perancangan sistem akan dirancangkan bagaimana cara kerja sistem secara garis besarnya. Setelah itu dilanjutkan dengan pembuatan sistem dan kemudian akan dilanjutkan dengan pengujian sistem.

#### **3.2 Analisis Kebutuhan**

Meningkatnya jumlah pengajuan kredit kepada pembiayaan memungkinkan pihak pembiayaan melakukan kesalahan dalam mengevaluasi calon konsumen. Hal

ini membuat peneliti ingin membantu permasalahan pembiayaan tersebut. Saat ini peneliti dapat memberikan solusi berupa pembuatan sebuah sistem berbasis *website* dimana sistem ini diharapkan dapat mendukung pengambilan keputusan pembiayaan dalam menyeleksi sebagai bahan pertimbangan dalam mengevaluasi calon konsumen.

Dalam analisis kebutuhan dibutuhkan data konsumen yang mengajukan kredit. Pada proses bisnis, data yang menjadi dasar untuk sistem pendukung pengambil keputusan pembiayaan berasal dari tim *marketing*. Tim *marketing* merupakan pihak yang berinteraksi langsung dengan calon konsumen dan mengumpulkan informasi terkait pengajuan kredit atau pembiayaan. Data ini mencakup umur, pendapatan, pengeluaran, jumlah pengajuan kredit, dan informasi lain yang relevan. Data-data ini kemudian disampaikan melalui perantara ke divisi data *analyst*. Divisi data *analyst* akan menganalisis data yang telah diterima dari tim *marketing*.

Selain itu, dibutuhkan tampilan web untuk menunjang aplikasi tersebut, *user requirement*, dan spesifikasi komputer yang digunakan. Dalam pengembangan sistem ini, terdapat satu *user* utama, yaitu *data analyst*. *User requirement* untuk *data analyst* adalah dapat melakukan prediksi untuk mendukung pengambilan keputusan dan akses penuh ke semua menu yang terdapat dalam *website* ini. Selain itu, diperlukan juga *system external* untuk prediksi berupa API sehingga dapat diintegrasikan ke dalam sistem lain sesuai kebutuhan. Berikut ini adalah spesifikasi komputer yang digunakan dalam pembuatan aplikasi ini yang dapat dilihat pada tabel 3.1.

Tabel 3.1 Spesifikasi Komputer

Komponen	Keterangan
Sistem operasi	Microsoft Windows 10 Pro
Prosesor	Processor Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz, 2208 Mhz, 6 Core(s), 12 Logical Processor(s)
Memori	8.00 GB
Resolusi layar	1920 x 1080

### **3.2.1 Atribut Data**

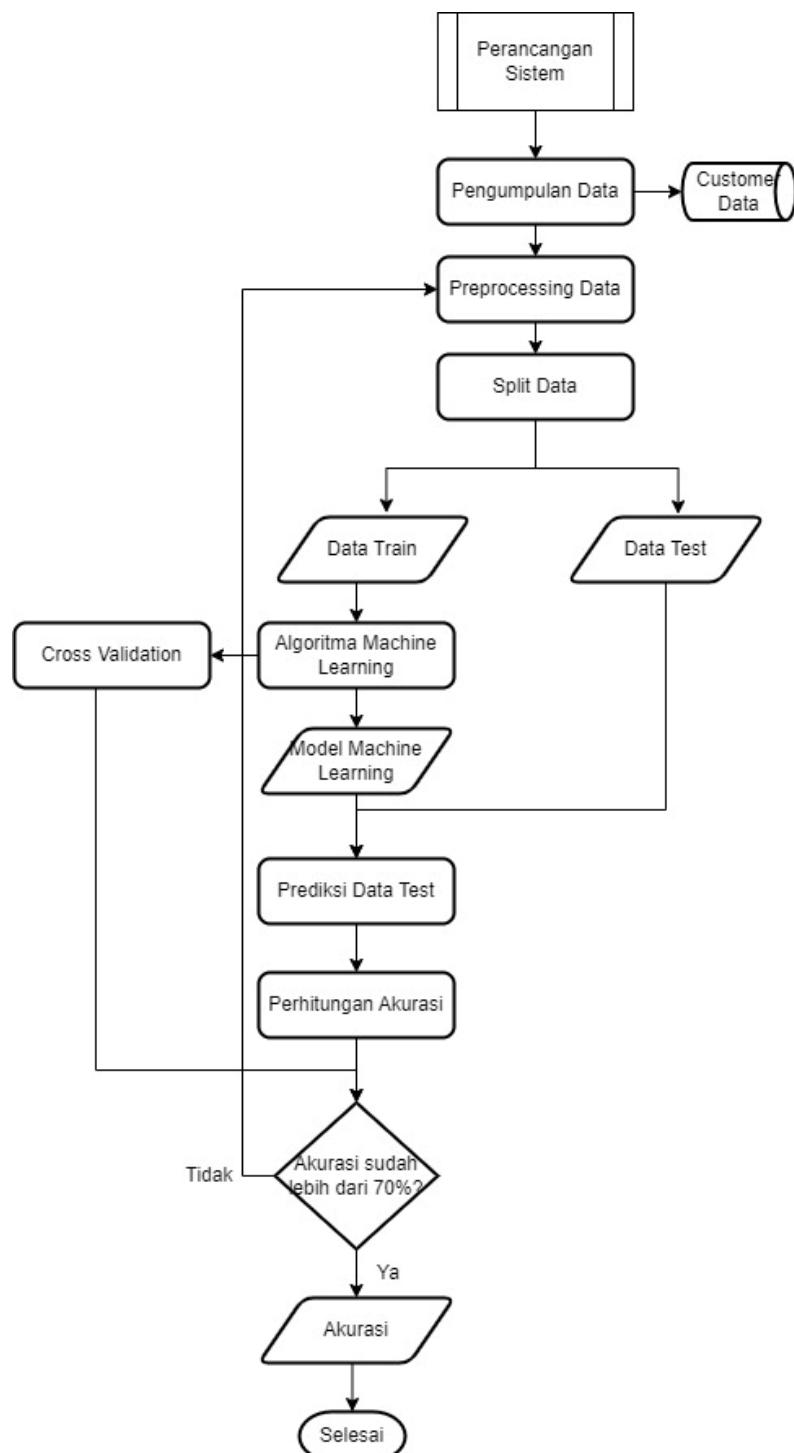
Pada data konsumen yang mengajukan kredit terdapat beberapa atribut. Di bawah ini adalah deskripsi singkat untuk setiap atribut yang digunakan.

1. Umur : Atribut ini mencatat usia konsumen yang mengajukan kredit. Data umur akan memberikan wawasan tentang rentang usia konsumen termasuk usia paling muda dan paling tua dari pemohon kredit.
2. Jangka waktu : Atribut ini merupakan variabel yang mengukur periode waktu tertentu, seperti berapa bulan atau tahun, dalam pengajuan kredit oleh konsumen. Data jangka waktu akan memberikan informasi tentang preferensi konsumen terkait durasi kredit yang diajukan, serta rentang waktu paling umum yang dipilih oleh pemohon kredit.
3. Pendapatan : Atribut ini mencatat jumlah penghasilan konsumen yang mengajukan kredit. Informasi tentang pendapatan akan membantu dalam menilai kemampuan finansial calon konsumen untuk membayar kredit yang diajukan.
4. Pengeluaran : Atribut ini mengukur total belanja atau pengeluaran konsumen. Data pengeluaran akan memberikan gambaran tentang pola pengeluaran dan perilaku konsumen terkait keuangan mereka.
5. Jumlah pengajuan konsumen : Atribut ini mencatat jumlah pengajuan kredit yang diajukan oleh konsumen.
6. Model konsumen : Atribut ini memberikan informasi tentang kelompok pekerjaan atau profesi konsumen yang mengajukan kredit.
7. Memiliki guarantor : Atribut ini mencatat apakah konsumen memiliki penjamin dalam pengajuan kredit. Informasi ini akan membantu dalam mengevaluasi tingkat kepercayaan dan risiko terkait kredit yang diajukan.
8. Keterangan : Atribut ini merupakan kelas atau label dari calon konsumen berdasarkan hasil analisis dan prediksi. Keterangan berisi informasi apakah konsumen tersebut termasuk *customer negatif* atau bukan *customer negatif*.

### **3.3 Perancangan Sistem**

Dalam tahap desain sistem, terdapat langkah-langkah yang diperlukan untuk membuat suatu sistem berbasis *website*. Hal ini dilakukan agar sistem yang dibuat

dapat memenuhi aspek yang diperlukan sehingga sistem ini nantinya dapat digunakan oleh pihak pembiayaan dengan baik. Sub proses perancangan sistem untuk pembuatan model *machine learning* diilustrasikan pada *flowchart* yang terdapat pada gambar 3.2 berikut.



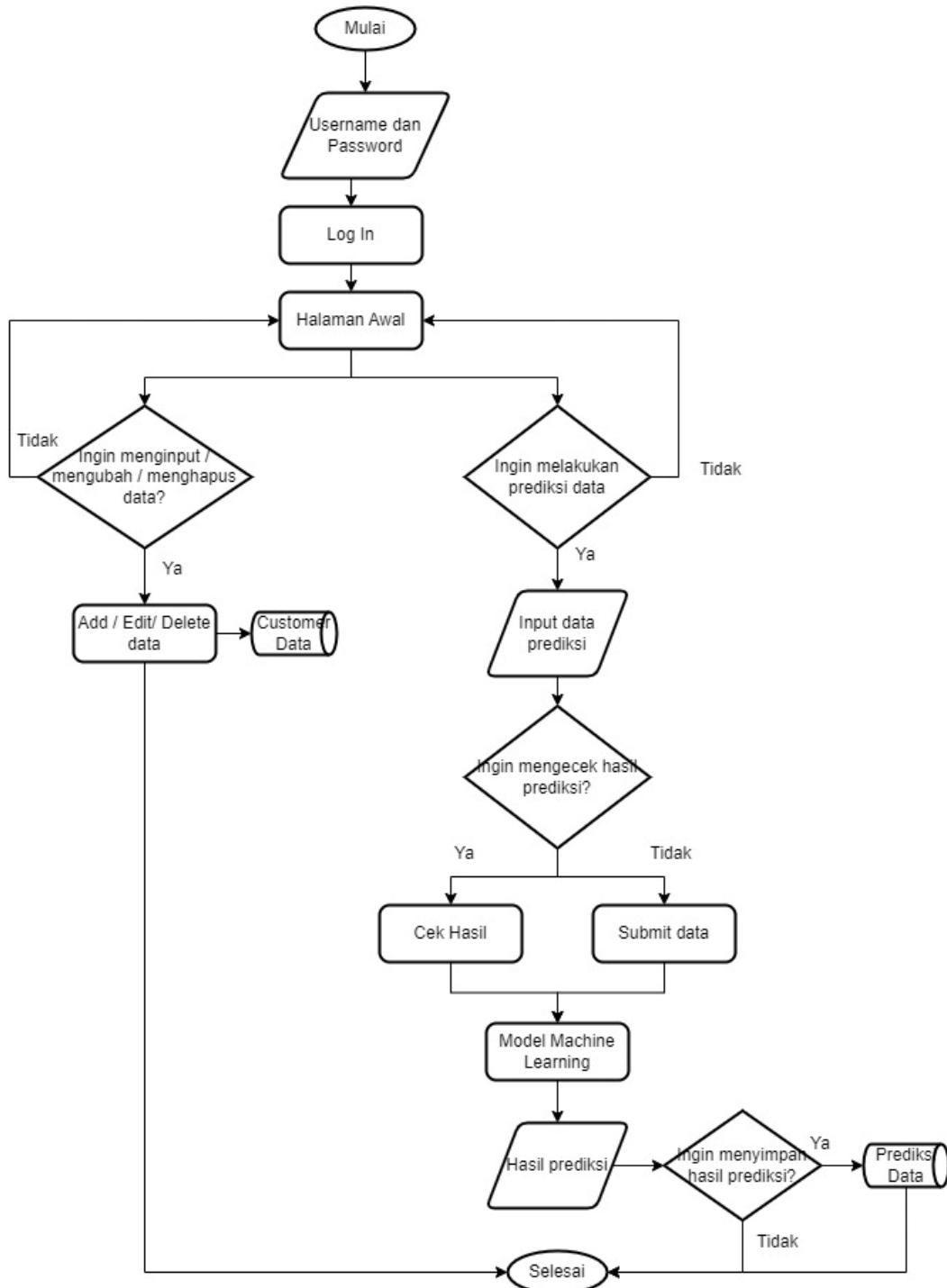
Gambar 3.2 *Flowchart* Proses Perancangan Sistem Model *Machine Learning*

Pada tahap awal perancangan sistem dilakukan pengumpulan data terlebih dahulu. Kemudian data yang telah dikumpulkan akan dimasukkan ke dalam database pada tabel *customer* data. Selanjutnya dilakukan tahap *preprocessing* pada data yang telah diperoleh, yaitu dikelompokkan sesuai dengan kriteria yang telah ditetapkan. Setelah melakukan pengelompokan data, data tersebut akan dibagi menjadi *data training* dan data test. *Data training* yang didapat akan diimplementasikan ke dalam algoritma *machine learning* untuk dilatih. Setelah itu akan dihasilkan model dari *machine learning* tersebut. Kemudian akan dilakukan prediksi data *test* dimana prediksi ini akan menggunakan model yang telah dilatih. Berikutnya akan dilakukan perhitungan akurasi. Jika hasil akurasi sudah lebih dari 70%, hasil akurasi telah didapatkan. Jika tidak, akan dilakukan penyesuaian kembali pada tahap *preprocessing* data untuk meningkatkan performa model. Pada pengimplementasian algoritma *machine learning* juga dilakukan *cross validation*. Dimana hasil rata-rata *cross validation* dilihat apakah sudah lebih dari 70%. Jika sudah maka dapat diperoleh hasil akurasi. Jika tidak maka akan kembali ke *preprocessing* data.

Selain itu, dalam tahap perancangan dibutuhkan tabel yang menyusun *database*. Hasil penyusunan ini nantinya akan dihubungkan dengan *website*. Terdapat tiga tabel pada *database* yang akan dibuat dalam pembuatan *website* sistem pendukung keputusan untuk menyeleksi konsumen negatif dalam mengajukan kredit, yaitu

1. Tabel CUSTOMER\_DATA berfungsi untuk menyimpan data-data konsumen yang telah dikumpulkan.
2. Tabel USER berfungsi untuk menyimpan *username* dan *password* yang telah didaftarkan.
3. Tabel PREDIKSI\_DATA berfungsi untuk menyimpan data-data yang telah diprediksi oleh pemberian.
4. Tabel TEMP\_PREDIKSI\_DATA berfungsi sebagai penyimpanan sementara untuk data yang telah diprediksi yang diunggah melalui file CSV.

Selanjutnya adalah rancangan *flowchart* untuk alur proses sistem pendukung keputusan untuk menyeleksi konsumen negatif dalam mengajukan kredit berbasis *website* yang diilustrasikan pada gambar 3.3 di bawah ini.

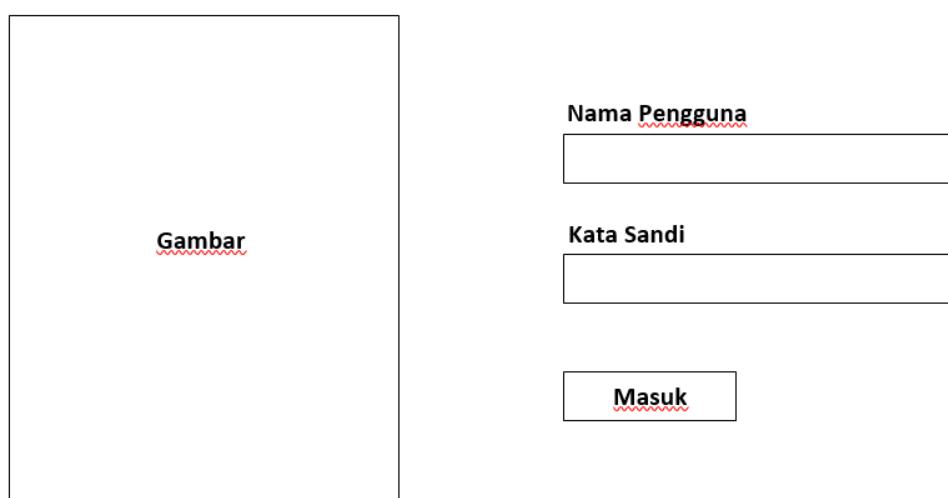


Gambar 3.3 Flowchart Sistem Pendukung Keputusan untuk Menyeleksi Konsumen

Proses dimulai dengan penginputan *username* dan *password*. Setelah *username* dan *password* diinput, proses login akan terjadi, membawa pengguna ke halaman awal website. Ada dua kondisi yang dapat dihadapi pengguna, yaitu penginputan data dan prediksi data. Pada kondisi pertama, jika pengguna ingin menginput,

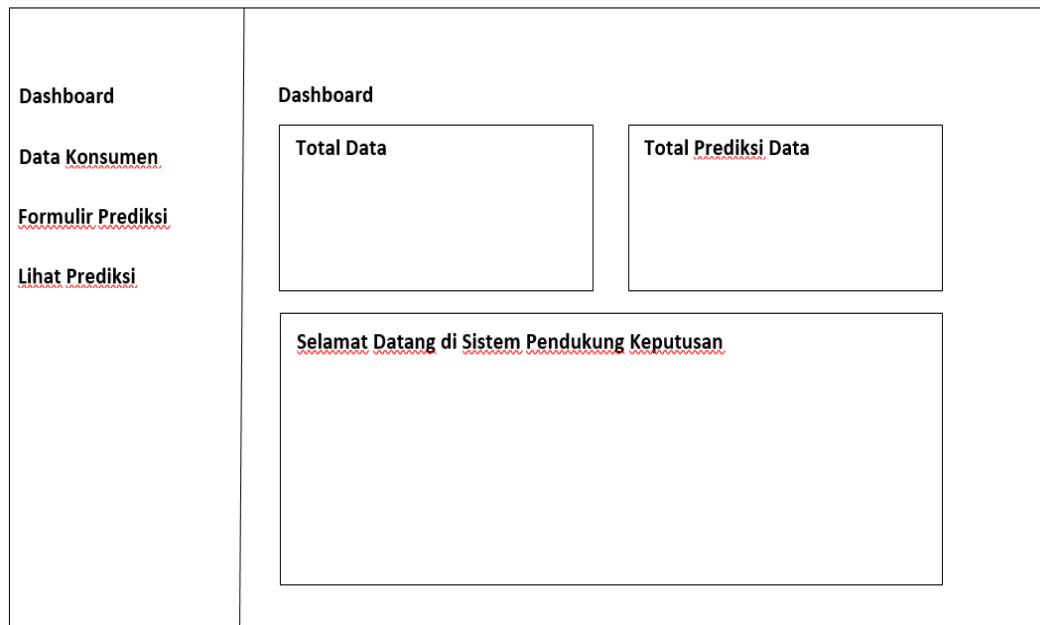
mengedit, atau menghapus data, pengguna dapat menambahkan, mengedit, atau menghapus data yang ada. Data yang diinput akan disimpan ke dalam *database* pada tabel *customer data*. Jika pengguna tidak ingin melakukan operasi tersebut, pengguna dapat kembali ke halaman awal. Pada kondisi kedua, jika pengguna ingin melakukan prediksi data, pengguna dapat menginputkan data yang akan diprediksi. Setelah penginputan data prediksi, ada opsi untuk memeriksa hasil prediksi atau tidak. Jika pengguna ingin memeriksa hasil, proses cek hasil akan dilakukan. Ini melibatkan penggunaan model machine learning untuk menghasilkan hasil prediksi. Setelah itu, pengguna dapat melihat hasil prediksi. Jika pengguna tidak ingin memeriksa hasil tersebut, pengguna dapat melakukan proses *submit data*. Data yang diinput akan diproses oleh model *machine learning* yang telah dibuat dan menghasilkan hasil prediksi. Setelah mendapatkan hasil prediksi, pengguna memiliki opsi untuk menyimpan hasil atau tidak. Jika pengguna tidak ingin menyimpan hasil, proses selesai. Jika pengguna ingin menyimpan hasil, data yang diinput akan dimasukkan ke dalam *database* dalam tabel prediksi data.

Setelah rancangan *flowchart* di atas, tahap selanjutnya adalah merancang sebuah *mockup* untuk *website* tersebut. *Mockup* merupakan suatu rancangan atau konsep gambar yang akan dibuat ke dalam produk. Berikut ini adalah *mockup* yang dirancang untuk pembuatan *website* sistem pendukung keputusan untuk menyeleksi konsumen negatif dalam mengajukan kredit.



Gambar 3.4 Mockup Halaman Log In

Gambar 3.4 merupakan gambar *mockup* untuk halaman *log in*. Pada *mockup* halaman *log in* ini terdapat nama pengguna dan kata sandi yang digunakan untuk masuk ke dalam *website*. Halaman *log in* ini dapat diakses oleh pembiayaan. Setelah merancang halaman *log in* dilakukan perancangan pada halaman *mockup dashboard*.



Gambar 3.5 *Mockup* Halaman *Dashboard*

Gambar 3.5 merupakan gambar *mockup* dari halaman *dashboard*. Pada halaman *dashboard* terdapat informasi untuk total data dan total prediksi data. Dimana untuk nilai total data diambil dari data pada menu data konsumen dan total prediksi data diambil dari data pada menu lihat prediksi. Setelah itu pada halaman ini juga terdapat tulisan berisi “Selamat Datang di Sistem Pendukung Keputusan”.

Setelah itu terdapat rancangan *mockup* untuk menu data konsumen. Dimana pada halaman tersebut terdapat *button* tambah data untuk menambah data konsumen. Kemudian terdapat *button* cari yang digunakan untuk mencari data. Pada halaman tersebut juga terdapat tabel yang berisi data konsumen yang dapat diubah atau dihapus melalui *button* yang terdapat pada kolom aksi. Untuk halaman data konsumen dapat dilihat pada gambar 3.6 di bawah ini.

<p><b>Dashboard</b></p> <p><b>Data Konsumen</b></p> <p><b>Formulir Prediksi</b></p> <p><b>Lihat Prediksi</b></p>	<p><b>Data Konsumen</b></p> <p style="text-align: right;"><b>Tambah Data</b></p> <p style="text-align: right;"><b>Cari</b></p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>NO</th><th>NAMA</th><th>UMUR</th><th>PENDAPATAN</th><th>PENGELUARAN</th><th>JUMLAH PENGAJUAN</th><th>JANGKA WAKTU</th><th>MODEL KONSUMEN</th><th>MEMILIKI GUARANTOR</th><th>KETERANGAN</th><th>AKSI</th></tr> </thead> <tbody> <tr> <td>1</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td> </td></tr> <tr> <td>2</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td> </td></tr> <tr> <td>3</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td> </td></tr> <tr> <td>4</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td>TEXT</td><td> </td></tr> </tbody> </table> <p style="text-align: right;"><b>Halaman</b></p>	NO	NAMA	UMUR	PENDAPATAN	PENGELUARAN	JUMLAH PENGAJUAN	JANGKA WAKTU	MODEL KONSUMEN	MEMILIKI GUARANTOR	KETERANGAN	AKSI	1	TEXT		2	TEXT		3	TEXT		4	TEXT																																	
NO	NAMA	UMUR	PENDAPATAN	PENGELUARAN	JUMLAH PENGAJUAN	JANGKA WAKTU	MODEL KONSUMEN	MEMILIKI GUARANTOR	KETERANGAN	AKSI																																														
1	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT																																															
2	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT																																															
3	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT																																															
4	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT																																															

Gambar 3.6 *Mockup* Halaman Data Konsumen

Setelah itu terdapat rancangan untuk *mockup* tambah data. Dimana pada halaman ini terdapat *form* yang dapat diisi oleh pembiayaan untuk menambahkan data. Pada *form* ini terdapat nama, umur, pendapatan, pengeluaran, jumlah pengajuan, model konsumen, memiliki garantor, dan keterangan. Untuk halaman ubah data sama seperti halaman tambah data. Gambar 3.7 adalah gambar untuk *mockup* halaman tambah data.

<p><b>Dashboard</b></p> <p><b>Data Konsumen</b></p> <p><b>Formulir Prediksi</b></p> <p><b>Lihat Prediksi</b></p>	<p><b>Tambah Data</b></p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;"><b>Nama</b></td> <td><input type="text"/></td> </tr> <tr> <td><b>Umur</b></td> <td><input type="text"/></td> </tr> <tr> <td><b>Pendapatan</b></td> <td><input type="text"/></td> </tr> <tr> <td><b>Pengeluaran</b></td> <td><input type="text"/></td> </tr> <tr> <td><b>Jumlah Pengajuan</b></td> <td><input type="text"/></td> </tr> <tr> <td><b>Jangka Waktu</b></td> <td><input type="text"/></td> </tr> <tr> <td><b>Model Konsumen</b></td> <td><input type="text"/></td> </tr> <tr> <td><b>Memiliki Guarantor</b></td> <td><input type="radio"/></td> </tr> <tr> <td><b>Keterangan</b></td> <td><input type="text"/></td> </tr> </table> <p style="text-align: right;"><b>Batal</b> <b>Tambah</b></p>	<b>Nama</b>	<input type="text"/>	<b>Umur</b>	<input type="text"/>	<b>Pendapatan</b>	<input type="text"/>	<b>Pengeluaran</b>	<input type="text"/>	<b>Jumlah Pengajuan</b>	<input type="text"/>	<b>Jangka Waktu</b>	<input type="text"/>	<b>Model Konsumen</b>	<input type="text"/>	<b>Memiliki Guarantor</b>	<input type="radio"/>	<b>Keterangan</b>	<input type="text"/>
<b>Nama</b>	<input type="text"/>																		
<b>Umur</b>	<input type="text"/>																		
<b>Pendapatan</b>	<input type="text"/>																		
<b>Pengeluaran</b>	<input type="text"/>																		
<b>Jumlah Pengajuan</b>	<input type="text"/>																		
<b>Jangka Waktu</b>	<input type="text"/>																		
<b>Model Konsumen</b>	<input type="text"/>																		
<b>Memiliki Guarantor</b>	<input type="radio"/>																		
<b>Keterangan</b>	<input type="text"/>																		

Gambar 3.7 *Mockup* Halaman Tambah Data

Kemudian terdapat *mockup* formulir prediksi, dimana pada halaman ini pembiayaan dapat melakukan prediksi terhadap data konsumen. Pada halaman ini prediksi dapat dilakukan dengan cara mengisi data konsumen seperti nama, umur, pendapatan, pengeluaran, jumlah pengajuan, jangka waktu, model konsumen, memiliki garantor. Setelah itu pada halaman ini akan menampilkan hasil keterangan apakah konsumen tersebut berpotensi *negatif* atau tidak dengan mengklik *button* cek. Setelah itu jika pembiayaan sudah mengecek dan mengisi semua data yang ada pada halaman formulir prediksi maka pembiayaan dapat melakukan simpan data untuk menyimpan data tersebut. Untuk *mockup* halaman formulir prediksi dapat dilihat pada gambar 3.8 di bawah ini.

<a href="#">Dashboard</a> <a href="#">Data Konsumen</a> <a href="#">Formulir Prediksi</a> <a href="#">Lihat Prediksi</a>	<b>Formulir Prediksi</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Nama</td> <td><input type="text"/></td> </tr> <tr> <td>Umur</td> <td><input type="text"/></td> </tr> <tr> <td>Pendapatan</td> <td><input type="text"/></td> </tr> <tr> <td>Pengeluaran</td> <td><input type="text"/></td> </tr> <tr> <td>Jumlah Pengajuan</td> <td><input type="text"/></td> </tr> <tr> <td>Jangka Waktu</td> <td><input type="dropdown"/></td> </tr> <tr> <td>Model Konsumen</td> <td><input type="dropdown"/></td> </tr> <tr> <td>Memiliki Guarantor</td> <td><input type="radio"/></td> </tr> <tr> <td>Keterangan</td> <td><b>Hasil Prediksi</b></td> </tr> </table> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="Cek"/> <input type="button" value="Simpan"/> </div>	Nama	<input type="text"/>	Umur	<input type="text"/>	Pendapatan	<input type="text"/>	Pengeluaran	<input type="text"/>	Jumlah Pengajuan	<input type="text"/>	Jangka Waktu	<input type="dropdown"/>	Model Konsumen	<input type="dropdown"/>	Memiliki Guarantor	<input type="radio"/>	Keterangan	<b>Hasil Prediksi</b>
Nama	<input type="text"/>																		
Umur	<input type="text"/>																		
Pendapatan	<input type="text"/>																		
Pengeluaran	<input type="text"/>																		
Jumlah Pengajuan	<input type="text"/>																		
Jangka Waktu	<input type="dropdown"/>																		
Model Konsumen	<input type="dropdown"/>																		
Memiliki Guarantor	<input type="radio"/>																		
Keterangan	<b>Hasil Prediksi</b>																		

Gambar 3.8 *Mockup* Halaman Formulir Prediksi

Pada *button* cek juga akan ditampilkan nilai akurasi yang didapat dari *confusion matrix*. Dari *confusion matrix* juga dapat menghitung nilai *precision*, *recall*, dan *f-measure*. Tabel 3.2 adalah contoh data *dummy* yang akan diimplementasikan ke dalam *confusion matrix*.

Tabel 3.2 Data Uji *Dummy*

Data Uji	Actual	Predicted
1	Tidak berpotensi menjadi negatif customer	Tidak berpotensi menjadi negatif customer
2	Berpotensi negatif customer	Tidak berpotensi menjadi negatif customer
3	Tidak berpotensi menjadi negatif customer	Tidak berpotensi menjadi negatif customer
4	Tidak berpotensi menjadi negatif customer	Tidak berpotensi menjadi negatif customer
...	...	...
10	Tidak berpotensi menjadi negatif customer	Berpotensi menjadi negatif customer

Berikut ini adalah *confusion matrix* dengan menggunakan data *dummy* pada tabel 3.2 di atas.

Tabel 3.3 Confusion Matrix

Actual		Predicted	
		Tidak Berpotensi	Potensi
Tidak Berpotensi		<i>True Positive (TP) = 3</i>	<i>False Negative (FN) = 2</i>
Potensi		<i>False Positive (FP) = 1</i>	<i>True Negative (TN) = 4</i>

Setelah didapatkan angka untuk *confusion matrix* pada tabel 3.3 maka langkah selanjutnya adalah menghitung *accuracy*, *precision*, *recall*, dan *f-measure*. Berikut ini adalah contoh perhitungan untuk nilai *accuracy*, *precision*, *recall*, dan *f-measure*. Untuk menghitung *accuracy* dapat menggunakan rumus sebagai berikut.

$$\begin{aligned}
 Accuracy &= \frac{TP + TN}{(TP + TN + FP + FN)} \\
 &= \frac{7}{10} = 0.7
 \end{aligned}$$

Untuk menghitung *precision* dapat menggunakan rumus sebagai berikut.

$$\begin{aligned} \textit{Precision} &= \frac{TP}{TP + FP} \\ &= \frac{3}{4} = 0.75 \end{aligned}$$

Untuk menghitung *recall* dapat menggunakan rumus sebagai berikut.

$$\begin{aligned} \textit{Recall} &= \frac{TP}{TP + FN} \\ &= \frac{3}{5} = 0.6 \end{aligned}$$

Untuk menghitung *f-measure* dapat menggunakan rumus sebagai berikut.

$$\begin{aligned} \textit{F - Measure} &= 2 \times \frac{\textit{Precision} \times \textit{Recall}}{\textit{Precision} + \textit{Recall}} \\ &= 2 \times \frac{0.6 \times 0.8}{0.6 + 0.8} = 0.69 \end{aligned}$$

Setelah itu terdapat *mockup* untuk halaman lihat prediksi. Dimana pada halaman ini terdapat data-data yang telah dilakukan prediksi oleh pembiayaan pada halaman formulir prediksi. Gambar 3.9 adalah gambar *mockup* dari lihat prediksi.

Lihat Prediksi										
<input type="button" value="Cari"/>										
NO	NAMA	UMUR	PENDAPATAN	PENGELUARAN	JUMLAH PENGAJUAN	JANGKA WAKTU	MODEL KONSUMEN	MEMILIKI GUARANTOR	KETERANGAN	
1	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	
2	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	
3	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	
4	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	TEXT	

Gambar 3.9 *Mockup* Halaman Lihat Prediksi

### **3.4 Pembuatan Sistem**

Pada pembuatan sistem pendukung keputusan untuk menyeleksi konsumen negatif menggunakan algoritma *machine learning* diperlukan *text editor* berupa Pycharm. Dalam pembuatan sistem ini juga diperlukan *database* untuk menyimpan data. *Database* yang digunakan adalah *database MySQL* yang diakses menggunakan *localhost* phpmyadmin. Untuk bahasa pemrograman yang digunakan adalah bahasa pemrograman Python dengan menggunakan *framework* Flask. Untuk desain antar muka akan menggunakan HTML dan bootstrap. Setelah itu untuk *library* yang akan digunakan adalah *library* scikit-learn dimana *library* ini akan diimport untuk mengimplementasikan algoritma *machine learning*. Modul merupakan file ekstensi dari bahasa pemrograman yang akan digunakan. Adapun modul yang digunakan adalah .py dan .html.

### **3.5 Pengujian Sistem**

Pada pengujian sistem ini dilakukan untuk menguji sistem apakah sistem yang dibuat dapat memprediksi data konsumen dengan baik dan akurat. Setelah itu akan diuji fitur-fitur yang terdapat di dalam *website* tersebut apakah berjalan dengan baik atau tidak. Tabel 3.4 adalah rincian dari rancangan pengujian sistem.

Tabel 3.4 Rancangan Pengujian

No	Deskripsi Pengujian	Keterangan	Hasil yang diharapkan
1	Melakukan pengecekan tingkat akurasi pengujian algoritma <i>machine learning</i>	Melihat persentase akurasi pengujian algoritma <i>machine learning</i>	Berhasil mencapai tingkat akurasi di atas 70%
2	Melakukan prediksi data konsumen	Memasukkan data konsumen di formulir prediksi	Berhasil melakukan prediksi data konsumen

Tabel 3.5 Rancangan Pengujian (Lanjutan)

No	Deskripsi Pengujian	Keterangan	Hasil yang diharapkan
3	Melihat data prediksi pada menu lihat prediksi	Membuka menu lihat prediksi	Berhasil melihat data yang telah diprediksi
4	Menguji API prediksi yang telah dibuat di <i>postman</i>	Menjalankan API di <i>postman</i>	Berhasil menjalankan API di <i>postman</i>

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Pengumpulan Data**

Pengumpulan data merupakan tahap awal dalam melakukan penelitian. Pengumpulan data adalah proses yang dilakukan dengan mengumpulkan data yang diperlukan untuk kebutuhan penelitian. Dalam penelitian ini, data dikumpulkan melalui dokumentasi perusahaan yang tersimpan dalam *database product* pembiayaan XYZ. *Database* ini berisi data konsumen yang namanya telah disamarkan, yang bertujuan untuk menjaga kerahasiaan dan privasi konsumen yang terlibat dalam penelitian. Selain itu, *database product* pembiayaan XYZ telah melalui proses *go live*, yang berarti *database* tersebut sudah aktif dan digunakan secara nyata dalam aktivitas bisnis sehari-hari.

Sebelum melakukan pengambilan data dari *database product* pembiayaan XYZ, menurut sumber, data ini adalah data yang diinput oleh tim *marketing* dan merupakan data yang valid. Hal ini menunjukkan bahwa data yang dikumpulkan sudah melewati proses verifikasi dan validasi sehingga dapat diandalkan untuk penelitian. Dalam dataset yang diperoleh, terdapat kemungkinan kasus data konsumen yang memiliki keterangan *customer negatif* disebabkan oleh beberapa faktor. Misalnya, konsumen tersebut memiliki riwayat dalam tindakan kriminal.

Dalam tahap pengambilan data tersebut diperoleh 1000 data. Data yang diperoleh mencakup berbagai jenis informasi seperti data umur, pendapatan, pengeluaran, jangka waktu, *customer model*, memiliki guarantor, jumlah pengajuan konsumen, dan keterangan. Namun, sebelum data tersebut dapat digunakan untuk penelitian diperlukan tahap *preprocessing* data.

#### **4.2 Preprocessing Data**

*Preprocessing* data adalah proses yang dilakukan untuk mengolah data mentah menjadi data yang lebih efisien dan siap digunakan untuk proses penelitian. Pada *preprocessing* data akan dilakukan data *cleaning*. Data *cleaning* adalah proses mengeliminasi atau menghilangkan data-data yang kurang sesuai. Data yang kurang sesuai dieliminasi atau dihilangkan karena bisa menyebabkan permasalahan

pada saat pemrosesan data berlangsung. Pada saat proses *cleaning* terdapat data-data anomali atau data yang kurang sesuai seperti contoh pendapatan lebih rendah daripada pengeluaran konsumen tersebut. Dari hasil data *cleaning*, diperoleh 894 data. Setelah dilakukan data *cleaning*, ada beberapa atribut yang memerlukan pengelompokkan data. Atribut tersebut adalah jangka waktu, umur, pendapatan, pengeluaran, dan jumlah pengajuan konsumen. Pada atribut jangka waktu pengelompokkan dibagi menjadi 5 bagian. Tabel 4.1 adalah tabel yang menggambarkan pengelompokkan untuk atribut jangka waktu.

Tabel 4.1 Pengelompokkan Jangka Waktu

Jangka Waktu	Keterangan
1 tahun	12 bulan
2 tahun	24 bulan
3 tahun	36 bulan
4 tahun	48 bulan
5 tahun	60 bulan

Setelah itu terdapat pengelompokan terhadap atribut umur. Pengelompokan terhadap atribut umur dibagi menjadi 9 bagian. Tabel 4.2 adalah tabel yang menggambarkan pengelompokan rentang umur berdasarkan sumber dari Badan Pusat Statistik.

Tabel 4.2 Pengelompokan Rentang Umur

Rentang Umur	Keterangan
20-24 tahun	Usia 20 hingga 24 tahun
25-29 tahun	Usia 25 hingga 29 tahun
30-34 tahun	Usia 30 hingga 34 tahun
35-39 tahun	Usia 35 hingga 39 tahun
40-44 tahun	Usia 40 hingga 44 tahun
45-49 tahun	Usia 45 hingga 49 tahun
50-54 tahun	Usia 50 hingga 54 tahun
55-59 tahun	Usia 55 hingga 59 tahun
>= 60 tahun	Usia 60 tahun ke atas

Kemudian terdapat pengelompokan untuk atribut pendapatan. Pengelompokan rentang pendapatan dibagi menjadi 3 bagian. Tabel 4.3 adalah tabel yang menggambarkan pengelompokan rentang pendapatan berdasarkan sumber dari (Triwijayati & Pradipta, 2018).

Tabel 4.3 Pengelompokkan Rentang Pendapatan

Rentang Pendapatan	Keterangan
<5.000.000 juta	Kurang dari 5.000.000 juta
5.000.000 juta – 10.000.000 juta	5.000.000 hingga 10.000.000 juta
>10.000.000 juta	Lebih dari 10.000.000 juta

Pada pembagian rentang pengeluaran dibagi menjadi 4 bagian. Tabel 4.4 adalah tabel yang menggambarkan pengelompokan rentang pendapatan berdasarkan sumber dari (Triwijayati & Pradipta, 2018) pada tabel 4.3.

Tabel 4.4 Pengelompokkan Rentang Pengeluaran

Rentang Pengeluaran	Keterangan
<3.000.000 juta	Kurang dari 3.000.000 juta
3.000.000 juta – 5.000.000 juta	3.000.000 hingga 5.000.000 juta
5.000.001 juta – 10.000.000 juta	5.000.001 hingga 10.000.000 juta
>10.000.000 juta	Lebih dari 10.000.000 juta

Selain itu, terdapat pengelompokan untuk atribut jumlah pengajuan konsumen yang dibagi menjadi 2 bagian. Tabel 4.5 adalah tabel yang menggambarkan pengelompokan jumlah pengajuan konsumen berdasarkan sumber dari (Indonesia, 2023).

Tabel 4.5 Pengelompokkan Jumlah Pengajuan Konsumen

Rentang Jumlah Pengajuan	Keterangan
<500.000.000 juta	Kurang dari 500.000.000 juta
>500.000.000 juta	Lebih dari 500.000.000 juta

### **4.3 Gambaran Profil Konsumen**

Profil konsumen memberikan gambaran tentang karakteristik dan kecenderungan para calon konsumen dalam mengajukan kredit. Berikut adalah gambaran profil konsumen berdasarkan data yang telah dikumpulkan.

1. Profil model konsumen: mayoritas konsumen sebesar 75.6% merupakan kategori Small Medium Enterprise, diikuti Employee sebesar 11.6%, Professional sebesar 2%, dan Non Professional sebesar 0.2%.
2. Profil jangka waktu: mayoritas konsumen sebesar 45.9% memilih jangka waktu kredit selama 5 tahun, dengan jangka waktu terpendek adalah 1 tahun (7.4%) dan 2 tahun (5.2%).
3. Profil umur: mayoritas konsumen berada pada rentang usia 35-54 tahun (54.3%), dengan rentang usia 45-49 tahun (16.7%) sebagai kelompok terbesar. Hal ini menunjukkan bahwa sebagian besar konsumen berada dalam kelompok usia yang produktif dan mungkin telah memiliki pekerjaan atau usaha stabil.
4. Profil pendapatan: mayoritas konsumen sebesar 81.6% memiliki pendapatan >10.000.000 juta.
5. Profil pengeluaran: mayoritas konsumen sebesar (54.1%) memiliki total pengeluaran dalam kisaran >10.000.000 juta.
6. Profil punya guarantor: sekitar 29.2% konsumen memiliki guarantor dalam pengajuan kredit mereka, sementara 60.2% tidak memiliki guarantor. Keberadaan guarantor dapat meningkatkan tingkat kepercayaan pembiayaan terhadap calon konsumen yang memiliki potensi risiko kredit lebih tinggi.
7. Profil jumlah pengajuan customer: mayoritas konsumen (71.6%) mengajukan kredit dengan jumlah di bawah 500.000.000 juta, sedangkan ">500.000.000 juta" (17.8%) merupakan kelompok dengan jumlah pengajuan paling rendah.
8. Profil keterangan (hasil klasifikasi): dari hasil klasifikasi, jumlah konsumen yang tergolong sebagai "BUKAN CUSTOMER NEGATIF" dan "CUSTOMER NEGATIF" memiliki jumlah yang sama, masing-masing sebanyak 447 data.

#### 4.4 Pengujian Menggunakan *Cross Validation*

Pengujian *cross validation* adalah pengujian yang dilakukan untuk mengukur kinerja dari suatu model *machine learning*. Pengujian *cross validation* dilakukan dengan menggunakan *library* Scikit-Learn di Python dengan menggunakan fungsi *cross\_val\_score*. Pengujian *cross validation* ini menggunakan nilai  $k = 10$ . Berikut ini adalah hasil dari pengujian *cross validation* pada masing-masing metode, yaitu *Decision Tree*, *K-Nearest Neighbor* (KNN), *Support Vector Machine* (SVM), *bagging classifier* masing-masing metode, dan *ensemble method*.

##### a. *Decision Tree*

Dalam pengujian *cross validation* menggunakan metode *decision tree*. Dapat dilihat bahwa dari hasil pengujian menunjukkan bahwa akurasi tertinggi diperoleh saat nilai  $k$  berada di 0 dan  $k$  berada di 5, yaitu sebesar 100% sedangkan untuk akurasi terendah diperoleh saat nilai  $k$  berada di 1, yaitu sebesar 71.11%. Dari *cross validation*  $k = 10$  menggunakan metode *decision tree* didapatkan rata-rata sebesar 85.93%. *Cross validation* dari *decision tree* yang telah dibuat dapat dilihat pada tabel 4.6.

Tabel 4.6 *Cross Validation* Menggunakan Metode *Decision Tree*

<b>K</b>	<b>Akurasi</b>
0	1.000000
1	0.711111
2	0.722222
3	0.777778
4	0.943820
5	1.000000
6	0.876404
7	0.943820
8	0.887640
9	0.730337

##### b. *K-Nearest Neighbor* (KNN)

Dalam pengujian *cross validation* menggunakan metode *K-Nearest Neighbor* (KNN). Dapat dilihat bahwa dari hasil pengujian menunjukkan bahwa akurasi tertinggi diperoleh saat nilai  $k$  berada di 5, yaitu sebesar 85.39% sedangkan untuk akurasi terendah diperoleh saat nilai  $k$  berada di 9, yaitu sebesar 61.79%. Dari *cross validation*  $k = 10$  menggunakan metode *K-Nearest Neighbor* (KNN) didapatkan

rata-rata sebesar 77.28%. *Cross validation* dari *K-Nearest Neighbor* (KNN) yang telah dibuat dapat dilihat pada tabel 4.7

Tabel 4.7 *Cross Validation* Menggunakan Metode *K-Nearest Neighbor* (KNN)

<b>K</b>	<b>Akurasi</b>
0	0.844444
1	0.744444
2	0.744444
3	0.788889
4	0.831461
5	0.853933
6	0.730337
7	0.820225
8	0.752809
9	0.617978

### c. *Support Vector Machine* (SVM)

Pada SVM digunakan kernel *Gaussian Radial Basic Function* (RBF). Dalam pengujian *cross validation* menggunakan metode *Support Vector Machine* (SVM). Dapat dilihat bahwa dari hasil pengujian menunjukkan bahwa akurasi tertinggi diperoleh saat nilai k berada di 0 dan k berada di 5, yaitu sebesar 100% sedangkan untuk akurasi terendah diperoleh saat nilai k berada di 2, yaitu sebesar 63.33%. Dari *cross validation* k = 10 menggunakan metode *Support Vector Machine* (SVM) didapatkan rata-rata sebesar 84.48%. *Cross validation* dari *Support Vector Machine* (SVM) yang telah dibuat dapat dilihat pada tabel 4.8

Tabel 4.8 *Cross Validation* Menggunakan Metode *Support Vector Machine* (SVM)

<b>K</b>	<b>Akurasi</b>
0	1.000000
1	0.688889
2	0.633333
3	0.744444
4	0.966292
5	1.000000
6	0.898876
7	0.966292
8	0.898876
9	0.651685

**d. Bagging Classifier**

Dalam pengujian *cross validation* menggunakan *bagging classifier* dengan 3 metode berbeda, dapat dilihat bahwa

**1. Bagging Classifier dengan metode Decision Tree**

Dari hasil pengujian menunjukkan bahwa akurasi tertinggi diperoleh saat nilai k berada di 5, yaitu sebesar 100% sedangkan untuk akurasi terendah diperoleh saat nilai k berada di 2, yaitu sebesar 70.00%. Dari *cross validation* k = 10 menggunakan *bagging classifier decision tree* didapatkan rata-rata sebesar 84.92%. *Cross validation* dari *bagging classifier decision tree* yang telah dibuat dapat dilihat pada tabel 4.9

Tabel 4.9 *Cross Validation* Menggunakan *Bagging Classifier Decision Tree*

K	Akurasi
0	0.988889
1	0.711111
2	0.700000
3	0.777778
4	0.932584
5	1.000000
6	0.865169
7	0.943820
8	0.853933
9	0.719101

**2. Bagging Classifier dengan metode K-Nearest Neighbor (KNN)**

Dari hasil pengujian menunjukkan bahwa akurasi tertinggi diperoleh saat nilai k berada di 5, yaitu sebesar 95.50% sedangkan untuk akurasi terendah diperoleh saat nilai k berada di 2, yaitu sebesar 71.11%. Dari *cross validation* k = 10 menggunakan *bagging classifier K-Nearest Neighbor (KNN)* didapatkan rata-rata sebesar 82.01 %. *Cross validation* dari *bagging classifier K-Nearest Neighbor (KNN)* yang telah dibuat dapat dilihat pada tabel 4.10

Tabel 4.10 *Cross Validation* Menggunakan *Bagging Classifier KNN*

K	Akurasi
0	0.833333
1	0.722222

Tabel 4.11 *Cross Validation* Menggunakan *Bagging Classifier KNN* (Lanjutan)

K	Akurasi
2	0.711111
3	0.777778
4	0.898876
5	0.955056
6	0.865169
7	0.842697
8	0.876404
9	0.719101

### 3. *Bagging classifier* dengan metode *Support Vector Machine* (SVM)

Dari hasil pengujian menunjukkan bahwa akurasi tertinggi diperoleh saat nilai k berada di 0, yaitu sebesar 100 % sedangkan untuk akurasi terendah diperoleh saat nilai k berada di 2, yaitu sebesar 62.22%. Dari *cross validation* k = 10 menggunakan *bagging classifier Support Vector Machine* (SVM) didapatkan rata-rata sebesar 83.92%. *Cross validation* dari *bagging classifier Support Vector Machine* (SVM) yang telah dibuat dapat dilihat pada tabel 4.12

Tabel 4.12 *Cross Validation* Menggunakan *Bagging Classifier SVM*

K	Akurasi
0	1.000000
1	0.700000
2	0.622222
3	0.744444
4	0.955056
5	0.988764
6	0.876404
7	0.943820
8	0.910112
9	0.651685

#### e. *Ensemble Method*

Dalam pengujian *cross validation ensemble method* menggunakan metode individu dan *ensemble method* menggunakan *bagging*. Dapat dilihat bahwa dari hasil pengujian menunjukkan bahwa akurasi tertinggi diperoleh saat nilai k berada di 0 dan 5, yaitu sebesar 100% sedangkan untuk akurasi terendah diperoleh saat nilai k berada di 2. Dari *cross validation* k = 10 menggunakan *ensemble method* individu didapatkan rata-rata sebesar 85.93% sedangkan untuk ensemble method

bagging didapatkan rata-rata sebesar 85.14%. *Cross validation* dari *ensemble method* yang telah dibuat dapat dilihat pada tabel 4.13

Tabel 4.13 *Cross Validation Ensemble Method*

<b>K</b>	<b>Akurasi Ensemble Individu</b>	<b>Akurasi Ensemble <i>Bagging</i></b>
0	1.000000	1.000000
1	0.711111	0.711111
2	0.700000	0.688889
3	0.777778	0.777778
4	0.943820	0.932584
5	1.000000	1.000000
6	0.876404	0.876404
7	0.955056	0.921348
8	0.910112	0.898876
9	0.719101	0.707865

Dalam pengujian *cross validation* dapat dilihat bahwa masing-masing metode menunjukkan perilaku yang hampir serupa pada *fold* 0 atau 5 dengan menghasilkan nilai yang tertinggi. Hal ini menunjukkan bahwa data pada *fold* tersebut mungkin memiliki karakteristik yang konsisten terhadap keseluruhan *dataset* dan model memiliki kemampuan yang konsisten dalam melakukan prediksi pada kedua subset tersebut. Selain itu dilakukan perhitungan untuk nilai *standard deviation* untuk masing-masing model. Semakin kecil nilainya maka semakin konsisten dan stabil hasil prediksi model. Hasil nilai *standard deviation* dari masing-masing metode dapat dilihat pada tabel 4.14

Tabel 4.14 Nilai *Standard Deviation*

<b>Model</b>	<b>Nilai <i>Standard Deviation</i></b>
<i>Decision Tree</i>	0.11
<i>K-Nearest Neighbors</i>	0.07
<i>Support Vector Machine</i>	0.14
<i>Decision Tree Bagging</i>	0.11
<i>K-Nearest Neighbors Bagging</i>	0.08
<i>Support Vector Machine Bagging</i>	0.14
<i>Ensemble Method Individu</i>	0.12
<i>Ensemble Method Bagging</i>	0.11

Dari tabel 4.14, dapat dilihat bahwa model *K-Nearest Neighbors* (KNN), memiliki nilai *standard deviation* terkecil, yaitu 0.07. Hal ini menunjukkan bahwa

model KNN tersebut memberikan hasil prediksi yang konsisten dan stabil dalam proses *cross validation*. Sementara itu, model *Support Vector Machine* (SVM) dan *Support Vector Machine* (SVM) *Bagging* memiliki nilai *standard deviation* tertinggi, yaitu 0.14. Hal ini menunjukkan bahwa model tersebut memiliki variasi yang lebih tinggi.

#### **4.5 Pengujian Akurasi Metode Menggunakan *Confusion Matrix***

Pengujian akurasi metode adalah suatu proses yang dilakukan untuk mengukur seberapa baik akurasi metode dalam melakukan prediksi pada data. Setelah proses pengelompokan selesai, tahap selanjutnya adalah tahap *labeling* data. Pada tahap ini, data akan diberikan label berdasarkan pengelompokkannya. Setelah tahap labeling data selesai, proses selanjutnya adalah melakukan *split* data.

*Split* data merupakan pembagian *dataset* menjadi dua bagian, yaitu data *train* dan data *test*. Pada proses *split* data, dilakukan uji coba pembagian data *train* dan data *test* dengan rasio tertentu. Rasio tersebut adalah 60:40, 70:30, 80:20, dan 90:10. Pembagian dengan rasio ini dilakukan untuk melihat bagaimana akurasi dari masing-masing metode yang digunakan dalam melakukan prediksi pada data dengan ukuran data *train* dan data *test* yang berbeda. Dalam pengujian akurasi ini terdapat 3 metode yang digunakan, yaitu *Decision Tree*, *K-Nearest Neighbor* (KNN), *Support Vector Machine* (SVM), *bagging classifier* masing-masing metode, dan *ensemble method*.

##### **1. *Decision Tree***

Tahap berikutnya setelah melakukan pembagian data dalam rasio tertentu adalah pembuatan model dengan menggunakan *decision tree*. Pembuatan model *decision tree* ini dilakukan dengan menggunakan *library* Scikit-Learn di Python. Kode 4.1 merupakan cara pembuatan model menggunakan *decision tree*. Pada kode 4.1 akan dipanggil suatu fungsi *DecisionTreeClassifier* untuk membuat objek *decision tree* yang akan dilatih dan dievaluasi.

Fungsi *DecisionTreeClassifier* memiliki beberapa parameter. Parameter yang pertama adalah *criterion* yang berfungsi untuk mengukur kualitas *split*. Pada kode 4.1, parameter *criterion* diberi nilai *entropy* untuk menghasilkan nilai *gain* yang digunakan sebagai perhitungan dalam membentuk *decision tree*. Parameter kedua

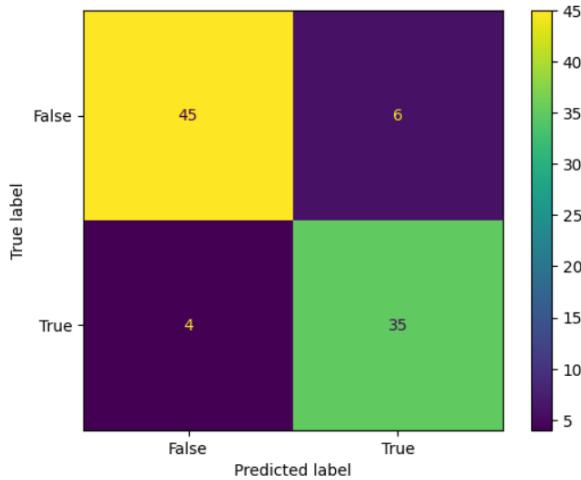
adalah *max\_depth* yang berfungsi untuk membatasi kedalaman maksimum pohon yang dimiliki oleh *decision tree*. Pada kode 4.1 *max\_depth* diberi nilai 4 karena dalam pengujian menggunakan rasio 90:10, grafik akurasi mencapai titik tertingginya pada nilai 4.

```
1.      dt = DecisionTreeClassifier(criterion="entropy",
max_depth = 4)
```

Kode 4.1 Pembuatan Model Decision Tree

Sebelum menggunakan parameter *max\_depth*, didapatkan nilai akurasi sebesar 78.89% dari pengujian menggunakan rasio 90:10. Dari hasil percobaan, kemudian dilakukan eksperimen dengan menggunakan parameter *max\_depth* dan ditemukan bahwa dengan menggunakan *max\_depth* = 4, akurasi model meningkat menjadi 88.89% pada rasio 90:10. Hal ini menunjukkan bahwa dengan membatasi kedalaman pohon, model *decision tree* dapat menghasilkan akurasi yang lebih tinggi dibandingkan tanpa menggunakan parameter *max\_depth*.

Setelah membuat model *decision tree* selanjutnya akan menghitung akurasi dari *decision tree* tersebut. Akurasi tersebut didapatkan dari *confusion matrix*. *Confusion matrix* memberikan informasi yang lebih rinci mengenai kinerja model pada data *test*. Berdasarkan hasil evaluasi *confusion matrix* untuk data *test* pada metode *decision tree* dengan rasio 90:10 didapatkan hasil dengan nilai akurasi sebesar 89% artinya sekitar 89% data pada data *test* telah diklasifikasikan dengan benar oleh model. Dari hasil tersebut, nilai *precision* yang didapatkan sebesar 85% artinya sebesar 85% data yang diklasifikasikan sebagai positif oleh model adalah benar-benar positif. Selain itu, didapatkan nilai *recall* sebesar 90% artinya model mampu menemukan sekitar 90% dari semua data positif yang ada dan nilai *f1 score* sebesar 88% yang menunjukkan bahwa model memiliki keseimbangan antara *precision* dan *recall* sekitar 88%. Selain itu, dilakukan pengujian *confusion matrix* untuk data *training* dan didapatkan hasil akurasi sebesar 86%, *precision* sebesar 84%, *recall* sebesar 91%, serta *f1 score* 87%. *Confusion matrix* dari *decision tree* untuk data *test* yang telah dibuat dapat dilihat pada gambar 4.1.



Gambar 4.1 *Confusion Matrix Data Test* dari Metode *Decision Tree*

Pada tabel 4.15 terdapat hasil dari pengujian akurasi metode *decision tree* dengan menggunakan variasi rasio pembagian data *train* dan data *test*. Dapat dilihat bahwa dari pengujian menggunakan metode *decision tree* akurasi tertinggi diperoleh saat menggunakan rasio 90:10, yaitu sebesar 88.89% pada akurasi data *test* sedangkan untuk akurasi terendah diperoleh saat menggunakan rasio 80:20, yaitu sebesar 82.68% pada akurasi data *test*.

Tabel 4.15 Akurasi *Decision Tree* dari Rasio Tertentu

Rasio Data <i>Train</i> dan Data <i>Test</i>	Akurasi Data <i>Test</i>	Akurasi Data <i>Training</i>
60 : 40	87.15%	86.19%
70 : 30	86.99%	86.40%
80 : 20	82.68%	86.85%
90 : 10	88.89%	86.32%

## 2. *K-Nearest Neighbor* (KNN)

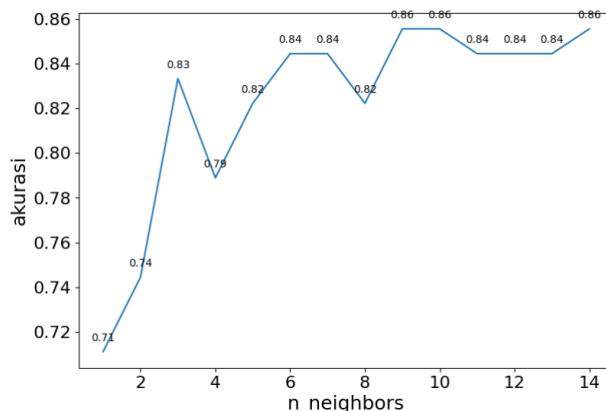
Tahap berikutnya setelah melakukan pembagian data dalam rasio tertentu adalah pembuatan model dengan menggunakan *K-Nearest Neighbor* (KNN). Pembuatan model *K-Nearest Neighbor* (KNN) ini dilakukan dengan menggunakan *library* Scikit-Learn di Python. Kode 4.2 merupakan cara pembuatan model menggunakan *K-Nearest Neighbor* (KNN). Pada kode 4.2 akan dipanggil suatu

fungsi *KNeighborsClassifier* untuk membuat objek *K-Nearest Neighbor* (KNN) yang akan dilatih dan dievaluasi.

```
1. for i in range(1, 15):
2.     # Create KNN classifier
3.     clf = KNeighborsClassifier(n_neighbors=i)
4.     # Fit the model
5.     clf.fit(X_train, y_train)
6.     # Append the model and score to their respective list
7.     models.append(clf)
8.     scores.append(accuracy_score(y_true = y_test, y_pred =
clf.predict(X_test)))
```

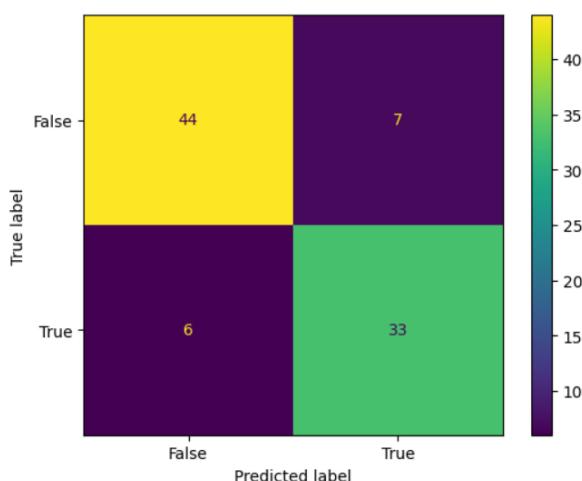
Kode 4.2 Pembuatan Model *K-Nearest Neighbor* (KNN)

Fungsi ini memiliki parameter *n\_neighbors* yang menentukan jumlah *neighbor*. Pada pengujian ini, dilakukan uji dengan berbagai nilai *n\_neighbors* untuk melihat nilai berapa yang memberikan akurasi maksimal. Untuk metode *K-Nearest Neighbor* (KNN), nilai *n\_neighbors* yang memberikan akurasi maksimal adalah 9. Grafik hasil pengujian menunjukkan bahwa akurasi mencapai puncaknya pada nilai tersebut. Oleh karena itu, *n\_neighbors* sebanyak 9 dipilih untuk metode *K-Nearest Neighbor* (KNN) dalam *KNeighborsClassifier*. Gambar 4.2 adalah grafik yang menunjukkan nilai *n\_neighbors* untuk metode *K-Nearest Neighbor* (KNN).



Gambar 4.2 Grafik Nilai *n\_neighbors*

Berdasarkan hasil evaluasi *confusion matrix* untuk data *test* pada metode *K-Nearest Neighbor* (KNN) dengan rasio 90:10 didapatkan hasil dengan nilai akurasi sebesar 86% artinya sekitar 86% data pada data test telah diklasifikasikan dengan benar oleh model. Dari hasil tersebut, nilai *precision* yang didapatkan sebesar 82% artinya sebesar 82% data yang diklasifikasikan sebagai positif oleh model adalah benar-benar positif. Selain itu, didapatkan nilai *recall* sebesar 85% artinya model mampu menemukan sekitar 85% dari semua data positif yang ada dan nilai *f1 score* sebesar 84% yang menunjukkan bahwa model memiliki keseimbangan antara *precision* dan *recall* sekitar 84%. Selain itu, dilakukan pengujian *confusion matrix* untuk data *training* dan didapatkan hasil akurasi sebesar 85%, *precision* sebesar 82%, *recall* sebesar 90%, serta *f1 score* 86%. *Confusion matrix* dari *K-Nearest Neighbor* (KNN) untuk data *test* yang telah dibuat dapat dilihat pada gambar 4.3.



Gambar 4.3 *Confusion Matrix* Data *Test* dari Metode *K-Nearest Neighbor*

Pada tabel 4.16 terdapat hasil dari pengujian akurasi metode *K-Nearest Neighbor* (KNN) dengan menggunakan variasi rasio pembagian data *train* dan data *test*. Dapat dilihat dari pengujian menggunakan metode *K-Nearest Neighbor* (KNN) akurasi tertinggi diperoleh saat menggunakan rasio 90:10, yaitu sebesar 85.56% pada akurasi data *test* sedangkan untuk akurasi terendah diperoleh saat menggunakan rasio 60:40, yaitu sebesar 81.01% pada akurasi data *test*.

Tabel 4.16 Akurasi *K-Nearest Neighbor* (KNN) dari Rasio Tertentu

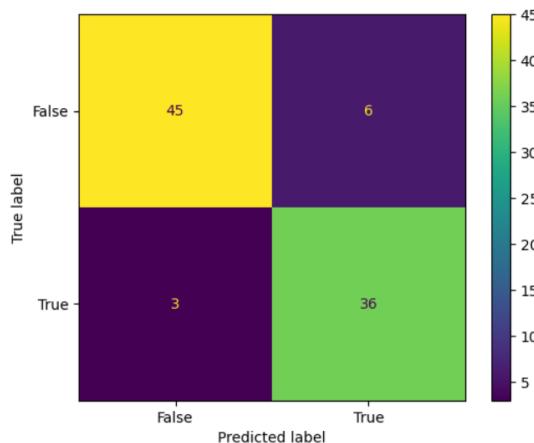
Rasio Data <i>Train</i> dan Data <i>Test</i>	Akurasi Data <i>Test</i>	Akurasi Data <i>Training</i>
60 : 40	81.01%	84.89%
70 : 30	83.27%	83.52%
80 : 20	82.12%	83.78%
90 : 10	85.56%	84.83%

### 3. *Support Vector Machine* (SVM)

Tahap berikutnya setelah melakukan pembagian data dalam rasio tertentu adalah pembuatan model dengan menggunakan *support vector machine* (SVM). Pembuatan model *support vector machine* (SVM) ini dilakukan dengan menggunakan *library* Scikit-Learn di Python. Dalam implementasi ini, akan dipanggil suatu fungsi SVC untuk membuat objek *support vector machine* (SVM) yang akan dilatih dan dievaluasi.

Berdasarkan hasil evaluasi *confusion matrix* untuk data *test* pada metode *support vector machine* (SVM) dengan rasio 90:10 didapatkan hasil dengan nilai akurasi sebesar 90% artinya sekitar 90% data pada data *test* telah diklasifikasikan dengan benar oleh model. Dari hasil tersebut, nilai *precision* yang didapatkan sebesar 86% artinya sebesar 86% data yang diklasifikasikan sebagai positif oleh model adalah benar-benar positif. Selain itu, didapatkan nilai *recall* sebesar 92% artinya model mampu menemukan sekitar 92% dari semua data positif yang ada dan nilai *f1 score* sebesar 89% yang menunjukkan bahwa model memiliki keseimbangan antara *precision* dan *recall* sekitar 89%. Selain itu, dilakukan pengujian *confusion matrix* untuk data *training* dan didapatkan hasil akurasi sebesar

87%, *precision* sebesar 83%, *recall* sebesar 94%, serta *f1 score* 88%. *Confusion matrix* dari *support vector machine* (SVM) untuk data *test* yang telah dibuat dapat dilihat pada gambar 4.4



Gambar 4.4 *Confusion Matrix* Data *Test* dari Metode *Support Vector Machine* (SVM)

Pada tabel 4.17 terdapat hasil dari pengujian akurasi metode *support vector machine* (SVM) dengan menggunakan variasi rasio pembagian data train dan data test. Dapat dilihat dari pengujian menggunakan metode *support vector machine* (SVM) akurasi tertinggi diperoleh saat menggunakan rasio 90:10, yaitu sebesar 90.00% pada akurasi data *test* sedangkan untuk akurasi terendah diperoleh saat menggunakan rasio 80:20, yaitu sebesar 87.15% pada akurasi data *test*.

Tabel 4.17 Akurasi *Support Vector Machine* (SVM) dari Rasio Tertentu

Rasio Data Train dan Data Test	Akurasi Data <i>Test</i>	Akurasi Data <i>Training</i>
60 : 40	87.71%	86.01%
70 : 30	88.48%	86.08%
80 : 20	87.15%	87.13%
90 : 10	90.00%	86.94%

#### 4. *Bagging Classifier*

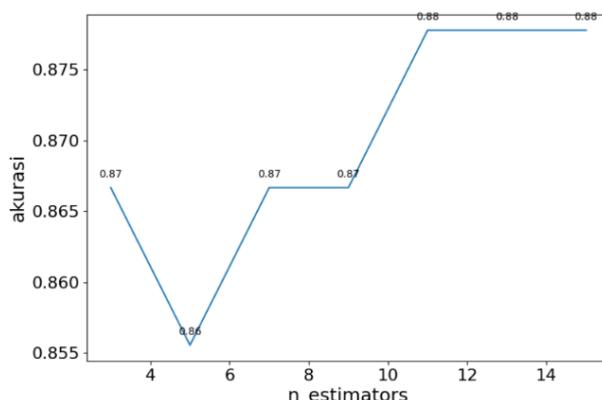
Dalam *bagging classifier*, subset data dibuat dengan mengambil sampel acak. Setiap subset digunakan untuk melatih model terpisah, kemudian hasil dari model-

model tersebut digabungkan menggunakan teknik voting. Dalam pengujian ini dilakukan bagging terhadap *decision tree*, *K-Nearest Neighbor* (KNN), dan *support vector machine* (SVM) dengan menggunakan fungsi *BaggingClassifier*. Pembuatan bagging ini dilakukan dengan menggunakan *library* Scikit-Learn di Python. Kode 4.4 merupakan cara memanggil fungsi *BaggingClassifier*.

```
1. bagging_dt = BaggingClassifier(estimator=dt,  
n_estimators=11, random_state = 42)  
  
2. bagging_cbr = BaggingClassifier(estimator=cbr,  
n_estimators=15, random_state = 42)  
  
3. bagging_svm = BaggingClassifier(estimator=svm,  
n_estimators=13, random_state = 42)
```

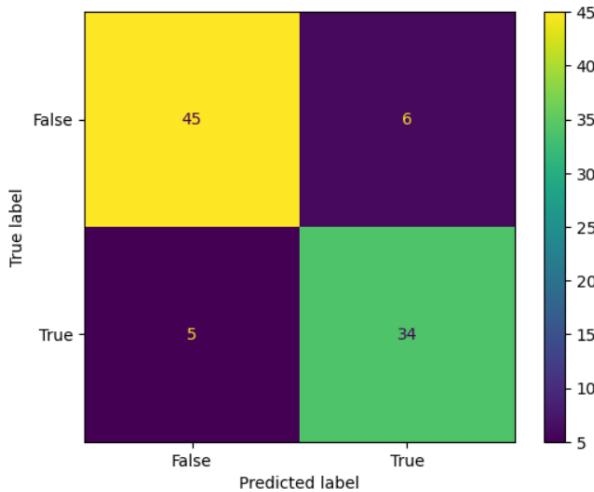
Kode 4.4 Pembuatan *Bagging Classifier*

Fungsi ini memiliki parameter *n\_estimator* yang menentukan jumlah model terpisah yang akan dilatih dalam proses bagging. Pada pengujian ini, dilakukan uji dengan berbagai nilai *n\_estimator* untuk melihat nilai berapa yang memberikan akurasi maksimal. Untuk metode *decision tree*, nilai *n\_estimator* yang memberikan akurasi maksimal adalah 11. Grafik hasil pengujian menunjukkan bahwa akurasi mencapai puncaknya pada nilai tersebut. Oleh karena itu, *n\_estimator* sebanyak 11 dipilih untuk metode *decision tree* dalam *Bagging Classifier*. Gambar 4.5 adalah grafik yang menunjukkan nilai *n\_estimator* untuk metode *decision tree*.



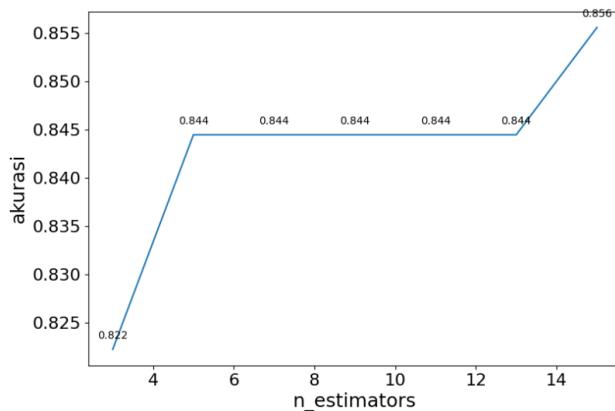
Gambar 4.5 Grafik Nilai *n\_estimators* *Decision Tree*

Berdasarkan hasil evaluasi *confusion matrix* untuk data *test* pada *bagging classifier decision tree* didapatkan hasil dengan nilai akurasi sebesar 88% artinya sekitar 88% data pada data *test* telah diklasifikasikan dengan benar oleh model. Dari hasil tersebut, nilai *precision* yang didapatkan sebesar 85% artinya sebesar 85% data yang diklasifikasikan sebagai positif oleh model adalah benar-benar positif.. Selain itu, didapatkan nilai *recall* sebesar 87% artinya model mampu menemukan sekitar 87% dari semua data positif yang ada dan nilai *f1 score* sebesar 86% yang menunjukkan bahwa model memiliki keseimbangan antara *precision* dan *recall* sekitar 86%. Selain itu, dilakukan pengujian *confusion matrix* untuk data *training* dan didapatkan hasil akurasi sebesar 87%, *precision* sebesar 84%, *recall* sebesar 91%, serta *f1 score* 87%. *Confusion matrix* dari *bagging classifier decision tree* untuk data *test* yang telah dibuat dapat dilihat pada gambar 4.6



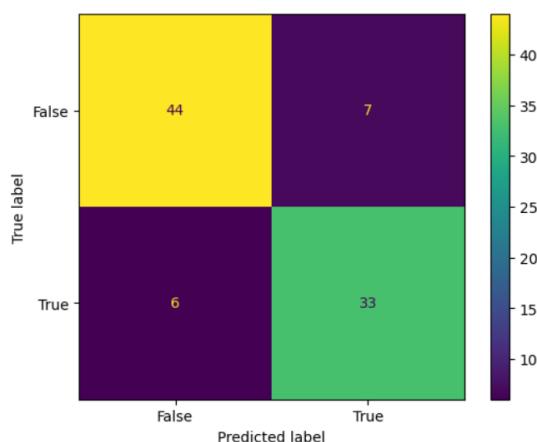
Gambar 4.6 *Confusion Matrix* Data *Test* dari *Bagging Classifier Decision Tree*

Untuk metode *K-Nearest Neighbor* (KNN), nilai *n\_estimator* yang memberikan akurasi maksimal adalah 15. Grafik hasil pengujian menunjukkan bahwa akurasi mencapai puncaknya pada nilai tersebut. Oleh karena itu, *n\_estimator* sebanyak 15 dipilih untuk metode KNN dalam *Bagging Classifier*. Gambar 4.7 adalah grafik yang menunjukkan nilai *n\_estimator* untuk metode KNN



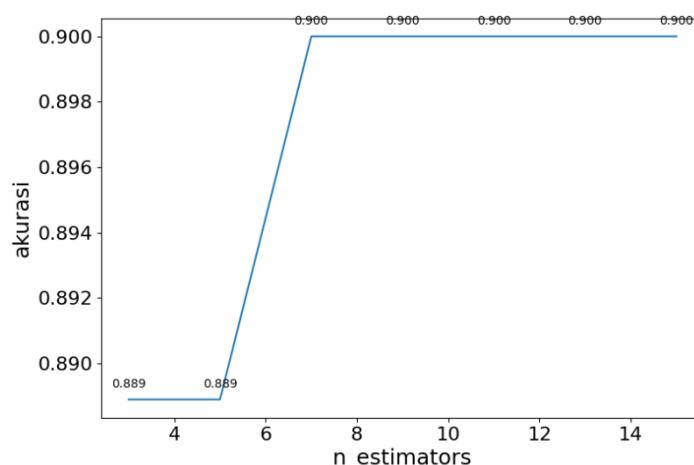
Gambar 4.7 Grafik Nilai n\_estimators KNN

Berdasarkan hasil evaluasi *confusion matrix* pada *bagging classifier* KNN didapatkan hasil dengan nilai akurasi sebesar 86% artinya sekitar 86% data pada data test telah diklasifikasikan dengan benar oleh model. Dari hasil tersebut, nilai *precision* yang didapatkan sebesar 82% artinya sebesar 82% data yang diklasifikasikan sebagai positif oleh model adalah benar-benar positif. Selain itu, didapatkan nilai *recall* sebesar 85% artinya model mampu menemukan sekitar 85% dari semua data positif yang ada dan nilai *f1 score* sebesar 84% yang menunjukkan bahwa model memiliki keseimbangan antara *precision* dan *recall* sekitar 84%. Selain itu, dilakukan pengujian *confusion matrix* untuk data training dan didapatkan hasil akurasi sebesar 85%, precision sebesar 82%, recall sebesar 90%, serta f1 score 86%. *Confusion matrix* dari *bagging classifier* KNN untuk data *test* yang telah dibuat dapat dilihat pada gambar 4.8



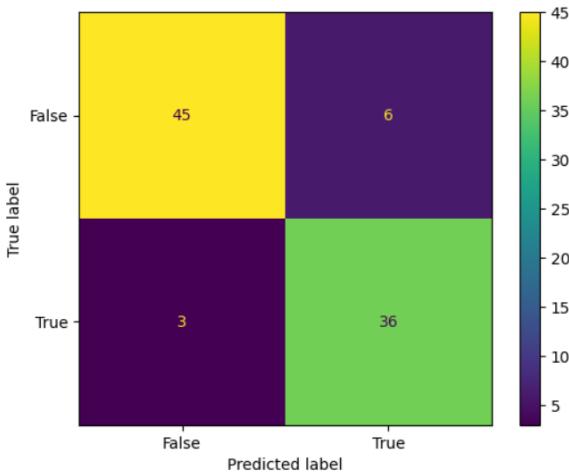
Gambar 4.8 *Confusion Matrix* Data *Test* dari *Bagging Classifier* KNN

Sementara itu, untuk metode *support vector machine* (SVM), nilai n\_estimator yang memberikan akurasi maksimal adalah 7. Grafik hasil pengujian menunjukkan bahwa akurasi mencapai titik tertinggi pada nilai tersebut. Oleh karena itu, n\_estimator sebanyak 7 dipilih untuk metode *support vector machine* (SVM) dalam *Bagging Classifier*. Gambar 4.9 adalah grafik yang menunjukkan nilai n\_estimator untuk metode SVM



Gambar 4.9 Grafik Nilai n\_estimators SVM

Berdasarkan hasil evaluasi *confusion matrix* untuk data *test* pada *bagging classifier support vector machine* (SVM) didapatkan hasil dengan nilai akurasi sebesar 90% artinya sekitar 90% data pada data test telah diklasifikasikan dengan benar oleh model. Dari hasil tersebut, nilai *precision* yang didapatkan sebesar 86% artinya sebesar 86% data yang diklasifikasikan sebagai positif oleh model adalah benar-benar positif. Selain itu, didapatkan nilai *recall* sebesar 92% artinya model mampu menemukan sekitar 92% dari semua data positif yang ada dan nilai *f1 score* sebesar 89% yang menunjukkan bahwa model memiliki keseimbangan antara *precision* dan *recall* sekitar 89%. Selain itu, dilakukan pengujian *confusion matrix* untuk data *training* dan didapatkan hasil akurasi sebesar 87%, *precision* sebesar 82%, *recall* sebesar 93%, serta *f1 score* 88%. *Confusion matrix* dari *bagging classifier support vector machine* (SVM) yang telah dibuat dapat dilihat pada gambar 4.10



Gambar 4.10 *Confusion Matrix* Data Test dari *Bagging Classifier SVM*

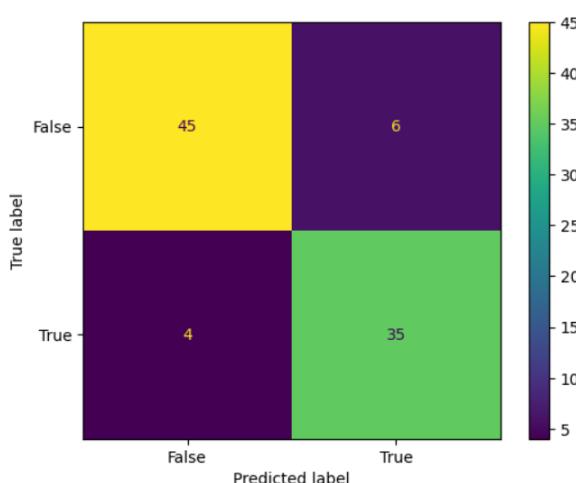
Setelah menerapkan nilai *n\_estimator* yang terbaik untuk masing-masing metode dalam bagging classifier, diperoleh hasil pengujian bahwa bagging *decision tree* mencapai akurasi 87.78 %, bagging *K-Nearest Neighbors* (KNN) mencapai akurasi 85.56 %, dan bagging *support vector machine* (SVM) mencapai akurasi 90.00 %.

#### 4. Ensemble Method

Dalam pengujian *ensemble method* diterapkan teknik voting untuk menggabungkan prediksi dari metode yang berbeda, yaitu *decision tree*, *K-Nearest Neighbors* (KNN), dan *support vector machine* (SVM). Pada *ensemble method* akan dipanggil suatu fungsi *VotingClassifier*. Terdapat dua pendekatan *ensemble method* yang diuji, yaitu *ensemble method* dengan menggunakan metode individu dan *ensemble method* dengan menggunakan *bagging*. Hasil pengujian menunjukkan bahwa *ensemble method* dengan menggunakan metode individu mencapai akurasi sebesar 88.89% dan *ensemble method* dengan menggunakan *bagging* mencapai akurasi sebesar 87.78%.

Berdasarkan hasil evaluasi *confusion matrix* untuk data *test* pada *ensemble method* dengan menggunakan metode individu didapatkan hasil dengan nilai akurasi sebesar 89% artinya sekitar 89% data pada data test telah diklasifikasikan dengan benar oleh model. Dari hasil tersebut, nilai *precision* yang didapatkan sebesar 85% artinya sebesar 85% data yang diklasifikasikan sebagai positif oleh

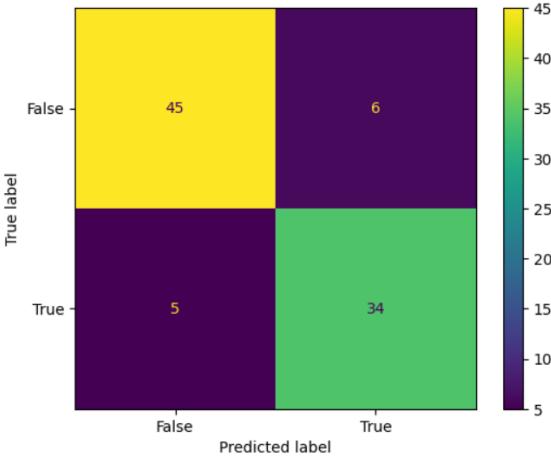
model adalah benar-benar positif. Selain itu, didapatkan nilai *recall* sebesar 90% artinya model mampu menemukan sekitar 90% dari semua data positif yang ada dan nilai *f1 score* sebesar 88% yang menunjukkan bahwa model memiliki keseimbangan antara precision dan recall sekitar 88%. Selain itu, dilakukan pengujian confusion matrix untuk data *training* dan didapatkan hasil akurasi sebesar 86%, precision sebesar 83%, recall sebesar 91%, serta f1 score 87%. *Confusion matrix* dari *ensemble method* untuk data *test* dengan menggunakan metode individu yang telah dibuat dapat dilihat pada gambar 4.11



Gambar 4.11 *Confusion Matrix* Data *Test* dari *Ensemble Method* Metode Individu

Sedangkan untuk hasil evaluasi confusion matrix untuk data *test* pada ensemble method dengan menggunakan *bagging* didapatkan hasil dengan nilai akurasi sebesar 88% artinya sekitar 88% data pada data test telah diklasifikasikan dengan benar oleh model. Dari hasil tersebut, nilai *precision* yang didapatkan sebesar 85% artinya sebesar 85% data yang diklasifikasikan sebagai positif oleh model adalah benar-benar positif. Selain itu, didapatkan nilai *recall* sebesar 87% artinya model mampu menemukan sekitar 87% dari semua data positif yang ada dan nilai *f1 score* sebesar 86% yang menunjukkan bahwa model memiliki keseimbangan antara precision dan recall sekitar 86%. Selain itu, dilakukan pengujian confusion matrix untuk data *training* dan didapatkan hasil akurasi sebesar 86%, precision sebesar 83%, recall sebesar 91%, serta f1 score 87%. *Confusion matrix* dari *ensemble*

*method* untuk data *test* dengan menggunakan *bagging* yang telah dibuat dapat dilihat pada gambar 4.12



Gambar 4.12 *Confusion Matrix Data Test* dari *Ensemble Method Bagging*

Dari pengujian *confusion matrix* masing-masing metode di atas, dapat dikatakan bahwa semakin tinggi nilai akurasi, semakin baik model dalam mengklasifikasikan data dengan tingkat kebenaran yang tinggi. Dalam pengujian ini, metode yang mencapai akurasi tertinggi adalah SVM dengan nilai akurasi sebesar 90%. Hal ini menunjukkan bahwa model SVM mampu mengklasifikasikan data calon konsumen dengan tingkat kebenaran mencapai 90% dari total data yang dievaluasi. Selain itu, semakin tinggi nilai *precision*, semakin baik model menghindari kesalahan dalam mengklasifikasikan data negatif sebagai positif. *Precision* yang tinggi menunjukkan bahwa sebagian besar prediksi positif yang dibuat oleh model adalah benar. Dalam pengujian ini, metode dengan nilai *precision* tertinggi adalah metode SVM dengan nilai *precision* sebesar 86%. Selanjutnya, semakin tinggi nilai *recall*, semakin baik model dalam mengidentifikasi dan mengambil data positif secara akurat. Dalam pengujian ini, metode dengan nilai *recall* tertinggi adalah metode SVM dengan nilai *recall* sebesar 92%. Selain itu, semakin tinggi nilai *f1 score* maka semakin baik model dalam mengklasifikasikan data positif dan negatif. *F1 score* menggabungkan *precision* dan *recall* yang menunjukkan keseimbangan antara keduanya. Dalam pengujian ini, metode dengan nilai *f1 score* tertinggi adalah metode SVM dengan nilai *f1 score* sebesar 89%. Secara keseluruhan, hasil pengujian *confusion matrix* menunjukkan bahwa metode SVM memiliki kinerja

yang paling tinggi dalam hal akurasi, *precision*, *recall*, maupun *f1 score*. Tabel 4.18 merupakan hasil *confusion matrix* dari masing-masing metode secara keseluruhan.

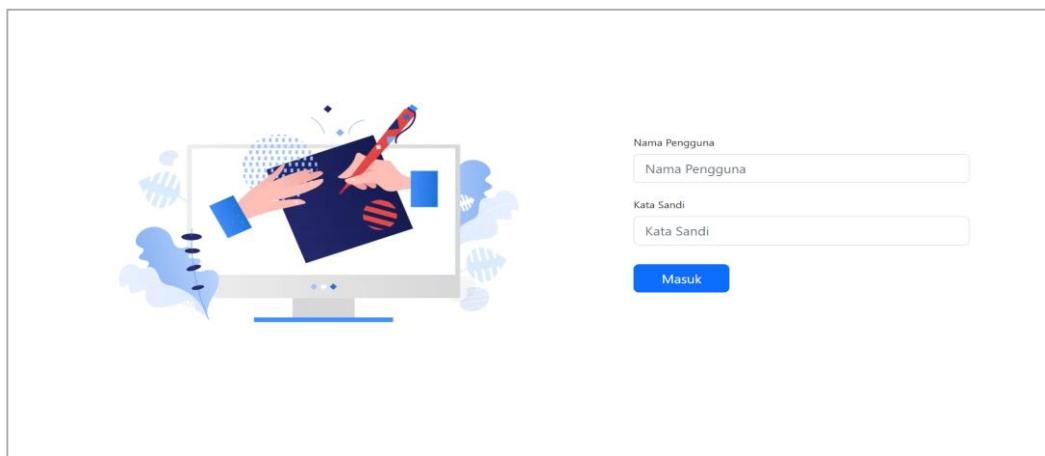
Tabel 4.18 Hasil *Confusion Matrix* Secara Keseluruhan

Model	<i>Confusion Matrix</i>	Nilai Data <i>Test</i> (%)	Nilai Data <i>Train</i> (%)
Decision Tree	Akurasi	89	86
	Precision	85	84
	Recall	90	91
	F1-Score	88	87
K-Nearest Neighbors (KNN)	Akurasi	86	85
	Precision	82	82
	Recall	85	90
	F1-Score	84	86
Support Vector Machine (SVM)	Akurasi	90	87
	Precision	86	83
	Recall	92	94
	F1-Score	89	88
Decision Tree Bagging	Akurasi	88	87
	Precision	85	84
	Recall	87	91
	F1-Score	86	87
K-Nearest Neighbors (KNN) Bagging	Akurasi	86	85
	Precision	82	82
	Recall	85	90
	F1-Score	84	86
Support Vector Machine (SVM) Bagging	Akurasi	90	87
	Precision	86	82
	Recall	92	93
	F1-Score	89	88
Ensemble Method Individu	Akurasi	89	86
	Precision	85	83
	Recall	90	91
	F1-Score	88	87
Ensemble Method Bagging	Akurasi	88	86
	Precision	83	83
	Recall	91	91
	F1-Score	87	87

## 4.6 Pengujian Sistem

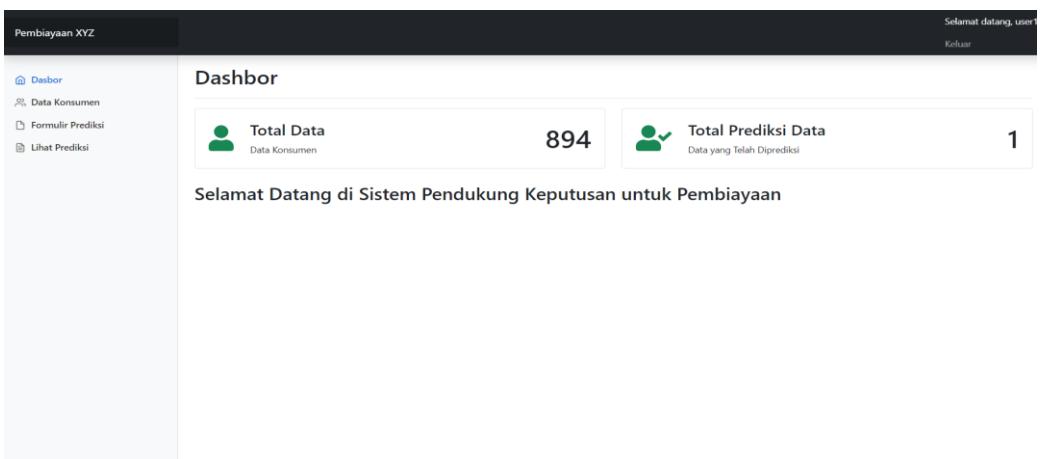
Dalam pengujian sistem langkah pertama yang perlu dilakukan adalah melakukan *login* untuk mengakses halaman utama *website*. Halaman *login* ini untuk memastikan bahwa pengguna yang ingin mengakses sistem sudah terdaftar dalam database pengguna. Gambar 4.13 adalah tampilan halaman *login* dari *website*

sistem pendukung keputusan untuk seleksi calon konsumen pada perusahaan pembiayaan dengan menggunakan machine learning.



Gambar 4.13 Tampilan Halaman *Login*

Pada halaman *login* nama pengguna diharapkan sudah didaftarkan oleh admin ke dalam *database*. Pengguna harus memasukkan kata sandi yang sesuai dengan akun mereka. Setelah itu pengguna akan diarahkan ke halaman dashbor sistem pendukung keputusan. Di dalam website terdapat 4 halaman utama, yaitu halaman utama, data konsumen, formulir prediksi, dan lihat prediksi. Gambar 4.14 ini adalah tampilan halaman dashbor sistem pendukung keputusan



Gambar 4.14 Tampilan Halaman Utama

Pada halaman utama, pengguna akan melihat informasi terkait total data konsumen yang telah diinput sebanyak 894 data dan total prediksi data yang telah diprediksi sebanyak 1 data. Selanjutnya terdapat halaman data konsumen. Pada halaman ini terdapat fitur untuk menambahkan data, search data berdasarkan nama, mengunggah csv, mengubah data, dan menghapus data. Gambar 4.15 adalah tampilan dari halaman menu data konsumen.

The screenshot shows a web application interface for managing consumer data. At the top, there's a navigation bar with links for 'Halaman Utama', 'Data Konsumen' (which is the active page), 'Formulir Prediksi', and 'Lihat Prediksi'. On the right side of the header, it says 'Selamat datang, user1' and 'Keluar'. Below the header, the main content area is titled 'Data Konsumen'. It features a table with 10 rows of consumer data. The columns include: No, Nama, Umur, Pendapatan, Pengeluaran, Jumlah Pengajuan, Jangka Waktu, Customer Model, Memiliki Guarantor, Keterangan, and Aksi. Each row has a set of buttons for 'Ubah' (Edit) and 'Hapus' (Delete). Below the table is a pagination bar with numbers 1 through 10. At the bottom right of the page, it says 'Total data : 894'.

No	Nama	Umur	Pendapatan	Pengeluaran	Jumlah Pengajuan	Jangka Waktu	Customer Model	Memiliki Guarantor	Keterangan	Aksi
1	DOUGLAS WONG	38	168,000,000	84,000,000	489,000,000	60	Small Medium Enterprise	YES	CUSTOMER NEGATIF	<button>Ubah</button> <button>Hapus</button>
2	MICHELLE PEREZ	60	160,000,000	80,000,000	482,500,000	48	Small Medium Enterprise	YES	CUSTOMER NEGATIF	<button>Ubah</button> <button>Hapus</button>
3	DAVID ENGLISH	37	125,000,000	50,000,000	733,000,000	60	Small Medium Enterprise	YES	CUSTOMER NEGATIF	<button>Ubah</button> <button>Hapus</button>
4	CHRISTOPHER PECK	63	86,000,000	15,000,000	607,000,000	12	Small Medium Enterprise	YES	CUSTOMER NEGATIF	<button>Ubah</button> <button>Hapus</button>
5	KATHLEEN MURRAY	52	150,000,000	80,000,000	900,000,000	60	Small Medium Enterprise	NO	CUSTOMER NEGATIF	<button>Ubah</button> <button>Hapus</button>
6	ALICE MCCOY	58	144,000,000	74,000,000	421,000,000	36	Small Medium Enterprise	NO	CUSTOMER NEGATIF	<button>Ubah</button> <button>Hapus</button>
7	RAYMOND SALINAS	45	80,000,000	10,000,000	596,700,000	36	Employee	NO	CUSTOMER NEGATIF	<button>Ubah</button> <button>Hapus</button>
8	ARIANA COOK	51	160,704,000	96,422,400	502,000,000	48	Small Medium Enterprise	NO	CUSTOMER NEGATIF	<button>Ubah</button> <button>Hapus</button>
9	SHERRI HURLEY	34	122,500,000	60,000,000	509,500,000	48	Small Medium Enterprise	NO	CUSTOMER NEGATIF	<button>Ubah</button> <button>Hapus</button>
10	CHERYL RIVERA	25	65,000,000	3,000,000	307,100,000	60	Employee	NO	CUSTOMER NEGATIF	<button>Ubah</button> <button>Hapus</button>

Gambar 4.15 Tampilan Halaman Menu Data Konsumen

Ketika pengguna ingin mengunggah csv, pengguna dapat menekan tombol unggah csv. Setelah itu halaman akan mengarah ke halaman unggah csv seperti pada gambar 4.16



Gambar 4.16 Tampilan Halaman Unggah CSV

Ketika pengguna ingin menambah data, pengguna dapat menekan tombol tambah data pada gambar 4.15. Saat menekan tombol tambah data halaman akan mengarah ke form tambah data. Pada form tambah data pengguna dapat menginput nama, umur, pendapatan, pengeluaran, jumlah pengajuan, jangka waktu, model konsumen, memiliki garantor, dan keterangan. Setelah menginput pengguna dapat menekan tombol tambah untuk menyimpan inputan tersebut. Jika pengguna tidak jadi menambah data maka pengguna dapat menekan tombol batal. Gambar 4.17 adalah tampilan form tambah data.

Tambah Data	
Nama	<input type="text" value="Nama"/>
Umur	<input type="text" value="Umur"/>
Pendapatan	<input type="text" value="Pendapatan"/>
Pengeluaran	<input type="text" value="Pengeluaran"/>
Jumlah Pengajuan	<input type="text" value="Jumlah Pengajuan"/>
Jangka Waktu	<input type="text" value="Jangka Waktu"/>
Model Konsumen	<input type="text" value="Employee"/>
Memiliki Guarantor	<input checked="" type="radio"/> Tidak <input type="radio"/> Ya
Keterangan	<input type="text" value="BUKAN CUSTOMER NEGATIF"/>
<input type="button" value="Batal"/> <input type="button" value="Tambah"/>	

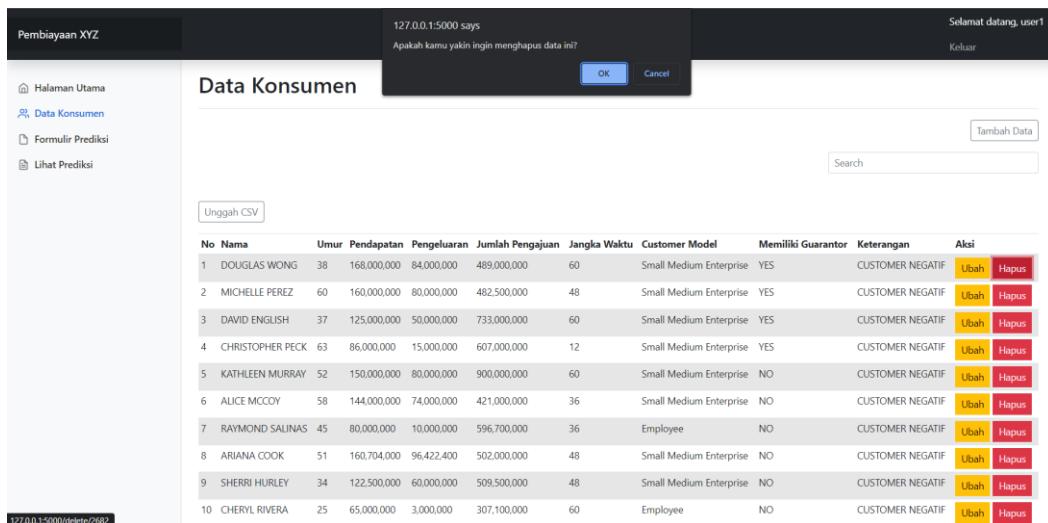
Gambar 4.17 Tampilan Form Tambah Data

Kemudian jika pengguna ingin mengubah data, pengguna dapat menekan tombol ubah yang terdapat pada gambar 4.15. Saat menekan tombol ubah, halaman akan mengarah ke form edit data. Gambar 4.18 adalah tampilan form ubah data.

Ubah Data	
Nama	<input type="text" value="AARON WILSON"/>
Umur	<input type="text" value="43"/>
Pendapatan	<input type="text" value="2500000"/>
Pengeluaran	<input type="text" value="1000000"/>
Jumlah Pengajuan	<input type="text" value="50250000"/>
Jangka Waktu	<input type="text" value="60"/>
Model Konsumen	<input type="text" value="Small Medium Enterprise"/>
Memiliki Guarantor	<input checked="" type="radio"/> Tidak <input type="radio"/> Ya
Keterangan	<input type="text" value="BUKAN CUSTOMER NEGATIF"/>
<input type="button" value="Batal"/> <input type="button" value="Ubah"/>	

Gambar 4.18 Tampilan Form Ubah Data

Selain tambah data dan ubah data, pengguna juga dapat menghapus data yang tidak diperlukan. Saat menekan tombol hapus maka data yang dipilih akan dihapus dan terdapat konfirmasi berupa apakah kamu yakin ingin menghapus data ini. Jika pengguna yakin ingin menghapus data tersebut dapat menekan tombol ok. Gambar 4.19 adalah tampilan saat ingin menghapus data.



Gambar 4.19 Tampilan Saat Ingin Menghapus Data

Selanjutnya terdapat halaman formulir prediksi. Di halaman ini pengguna dapat melakukan prediksi data dengan mengisi formulir yang disediakan. Halaman formulir prediksi ini dirancang dengan menggunakan metode-metode *machine learning* yang dapat menghasilkan prediksi. Gambar 4.20 adalah tampilan halaman formulir prediksi.

Metode yang Digunakan	Decision Tree
Nama	<input type="text" value="Nama"/>
Umur	<input type="text" value="Umur"/>
Pendapatan	<input type="text" value="Pendapatan"/>
Pengeluaran	<input type="text" value="Pengeluaran"/>
Jumlah Pengajuan	<input type="text" value="Jumlah Pengajuan"/>
Jangka Waktu	<input type="text" value="Jangka Waktu"/>
Model Konsumen	Employee
Memiliki Garantor	<input checked="" type="radio"/> Tidak <input type="radio"/> Ya

Gambar 4.20 Tampilan Formulir Prediksi

Pada formulir prediksi, setelah pengguna mengisi formulir tersebut pengguna dapat menekan tombol cek untuk memperoleh keterangan mengenai apakah customer tersebut berpotensi menjadi negatif atau tidak dan akan terdapat akurasi saat menggunakan metode *machine learning* tersebut. Jika pengguna menekan tombol cek, data tidak akan disimpan ke dalam database. Gambar 4.21 adalah tampilan halaman formulir prediksi ketika pengguna menekan tombol cek.

Selamat datang, user1  
Keluar

Pembayaran XYZ

[Halaman Utama](#) [Data Konsumen](#) [Formulir Prediksi](#) [Lihat Prediksi](#)

**Formulir Prediksi**

Unggah CSV | Metode yang Digunakan: Decision Tree

Nama:	TEST DATA
Umur:	25
Pendapatan:	100,000,000
Pengeluaran:	65,000,000
Jumlah Pengajuan:	500,000,000
Jangka Waktu:	12
Model Konsumen:	Employee
Memiliki Guarantor:	<input checked="" type="radio"/> Tidak <input type="radio"/> Ya

Keterangan : BUKAN CUSTOMER NEGATIF  
Akurasi menggunakan metode Decision Tree adalah : 88.89%

Cek Simpan

Gambar 4.21 Tampilan Formulir Prediksi Saat Menekan Tombol Cek

Jika pengguna ingin menyimpan data tersebut maka pengguna dapat menekan tombol simpan. Data yang disimpan dapat dilihat di halaman lihat prediksi. Halaman lihat prediksi dirancang untuk pengguna dapat dengan mudah melihat data prediksi yang telah disimpan sebelumnya. Gambar 4.22 adalah tampilan halaman lihat prediksi.

Selamat datang, user1  
Keluar

Pembayaran XYZ

[Halaman Utama](#) [Data Konsumen](#) [Formulir Prediksi](#) [Lihat Prediksi](#)

**Lihat Prediksi**

Unduh Laporan | Search

No	Nama	Umur	Pendapatan	Pengeluaran	Jumlah Pengajuan	Jangka Waktu	Customer Model	Memiliki Guarantor	Keterangan	Model	Akurasi
1	b	26	10,000,000	5,000,000	20,000,000	12	Employee	YES	BUKAN CUSTOMER NEGATIF	Decision Tree	88.8889 %
2	A	25	10,000,000	5,000,000	20,000,000	12	Employee	YES	BUKAN CUSTOMER NEGATIF	Decision Tree	88.8889 %
3	A	25	10,000,000	5,000,000	20,000,000	12	Employee	YES	BUKAN CUSTOMER NEGATIF	Decision Tree	88.8889 %
4	A	25	10,000,000	5,000,000	20,000,000	12	Employee	YES	BUKAN CUSTOMER NEGATIF	Decision Tree	88.8889 %
5	A	25	10,000,000	5,000,000	20,000,000	12	Employee	YES	BUKAN CUSTOMER NEGATIF	Decision Tree	88.8889 %
6	TEST DATA	25	100,000,000	65,000,000	500,000,000	12	Employee	YES	BUKAN CUSTOMER NEGATIF	K-Nearest Neighbor Bagging	84.44 %
7	A	25	10,000,000	5,000,000	20,000,000	12	Employee	YES	BUKAN CUSTOMER NEGATIF	Decision Tree	85.5556 %
8	A	25	10,000,000	5,000,000	20,000,000	12	Employee	YES	BUKAN CUSTOMER NEGATIF	Decision Tree	85.5556 %
9	A	25	10,000,000	5,000,000	20,000,000	12	Employee	YES	BUKAN CUSTOMER NEGATIF	Decision Tree	85.5556 %

Gambar 4.22 Tampilan Halaman Lihat Prediksi

## **4.7 Evaluasi Pengujian**

### **1. Evaluasi Pengujian Akurasi**

#### a. Evaluasi *cross validation*

Setelah melakukan evaluasi *cross validation* dengan  $k = 10$  menggunakan tiga metode *machine learning*, yaitu *decision tree*, KNN, dan SVM, serta menerapkan *bagging* untuk masing-masing metode tersebut dan *ensemble method*, didapatkan hasil sebagai berikut :

##### 1. *Decision Tree*

- Rata-rata Akurasi *Decision Tree* : 85.93%
- Rata-rata Akurasi *Decision Tree* dengan *Bagging* : 84.92%

##### 2. KNN

- Rata-rata Akurasi KNN : 77.28%
- Rata-rata Akurasi KNN dengan *Bagging* : 82.01%

##### 3. SVM

- Rata-rata Akurasi SVM : 84.48%
- Rata-rata Akurasi SVM dengan *Bagging* : 83.92%

##### 4. *Ensemble Method*

- Rata-rata Akurasi *Ensemble Method* : 85.93%
- Rata-rata Akurasi *Ensemble Method Bagging* : 85.14%

Dari evaluasi *cross validation* tersebut didapatkan hasil bahwa metode *decision tree* memiliki rata-rata akurasi yang paling tinggi, yaitu sebesar 85.93%.

#### b. Evaluasi *confusion matrix*

Setelah melakukan evaluasi *confusion matrix* menggunakan tiga metode *machine learning*, yaitu *decision tree*, KNN, dan SVM, serta menerapkan *bagging* untuk masing-masing metode tersebut dan *ensemble method*, didapatkan hasil sebagai berikut :

##### 1. *Decision Tree*

- Akurasi *Decision Tree* : 88.89%
- Akurasi *Decision Tree* dengan *Bagging* : 87.78%

##### 2. KNN

- Akurasi KNN : 85.56%

- Akurasi KNN dengan *Bagging* : 85.56%
3. SVM
- Akurasi SVM : 90.00%
  - Akurasi SVM dengan *Bagging* : 90.00%
4. Ensemble
- Akurasi *Ensemble Method* dengan menggunakan metode individu : 88.89%
  - Akurasi *Ensemble Method* dengan menggunakan *bagging* : 87.78%
- Dari hasil evaluasi tersebut, dapat dilihat bahwa metode SVM menghasilkan akurasi yang paling tinggi, yaitu 90.00%. Setelah itu dilanjutkan dengan *decision tree* yang menghasilkan akurasi 88.89% dan KNN yang menghasilkan akurasi 85.56%.
- Hasil uji coba pengujian akurasi dengan menggunakan *confusion matrix* dan cross validation menghasilkan akurasi yang berbeda. Pada pengujian *confusion matrix*, metode SVM menghasilkan akurasi tertinggi, yaitu sebesar 90.00% sedangkan pada pengujian *cross validation*, rata-rata tertinggi diraih oleh metode *decision tree*, yaitu sebesar 85.70%.

## 2. Evaluasi Pengujian Aplikasi

Proses pengujian dilakukan secara *blackbox*. Dari evaluasi pengujian aplikasi, secara keseluruhan aplikasi dapat berjalan dengan lancar. Pada sistem aplikasi *website*, semua fitur dapat digunakan dengan baik seperti penambahan data konsumen, pengubahan data konsumen, dan penghapusan data konsumen. Selain itu, pengguna dapat melakukan unggah file dengan format csv untuk menambah data dan memprediksi data. Tabel 4.19 merupakan rincian dari hasil pengujian sistem yang berjalan.

Tabel 4.19 Hasil Pengujian Aplikasi

No	Pengujian	Keterangan
1	Perhitungan persentase akurasi metode <i>machine learning</i> mencapai tingkat akurasi di atas 70% berhasil dilakukan dengan melakukan	Berhasil

Tabel 4.18 Hasil Pengujian Aplikasi (Lanjutan)

No	Pengujian	Keterangan
	evaluasi menggunakan <i>cross validation</i> dan <i>confusion matrix</i>	
2	Pada menu formulir prediksi, pengguna dapat melakukan prediksi data dengan mengisi semua input yang diperlukan pada formulir tersebut dan kemudian aplikasi akan memberikan hasil prediksi	Berhasil
3	Pengguna dapat melihat data yang telah diprediksi sebelumnya di menu lihat prediksi	Berhasil
4	Uji coba API untuk hasil prediksi di <i>postman</i> dan API tersebut memberikan output yang sesuai berisi keterangan calon konsumen	Berhasil

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil dari pembuatan sistem pendukung keputusan untuk seleksi calon konsumen pada perusahaan pembiayaan dengan menggunakan *machine learning*, dapat disimpulkan bahwa aplikasi tersebut telah berjalan dengan baik untuk mendukung pengambilan keputusan dalam seleksi calon konsumen. Implementasi *machine learning* pada *website* memungkinkan pengguna, khususnya *data analyst* untuk melakukan pengelolaan data konsumen dan melakukan prediksi dengan bantuan algoritma *machine learning*.

Dalam proses pengembangan sistem, berbagai metode *machine learning* telah diimplementasikan dan dievaluasi. Berdasarkan hasil evaluasi *cross validation* dengan nilai  $k = 10$ , metode *Decision Tree* memiliki rata-rata akurasi tertinggi sebesar 85,93%, diikuti oleh metode *bagging Decision Tree* sebesar 84.92%, *Support Vector Machine* (SVM) sebesar 84.48%, *bagging Support Vector Machine* (SVM) sebesar 83.92%, *bagging K-Nearest Neighbor* (KNN) sebesar 82.01%, *K-Nearest Neighbor* (KNN) sebesar 77.28% dan *ensemble method* sebesar 77.28%. Selain itu, pada *cross validation* dilakukan uji coba untuk nilai *standard deviation*. Didapatkan bahwa model *K-Nearest Neighbor* (KNN) memiliki nilai *standard deviation* terkecil, yaitu 0.07 yang berarti model tersebut memberikan hasil prediksi yang konsisten.

Selanjutnya, hasil evaluasi menggunakan confusion matrix menunjukkan bahwa metode *Support Vector Machine* (SVM) mencapai akurasi tertinggi sebesar 90% diikuti oleh metode *Decision Tree* mencapai akurasi 88.89% dan *K-Nearest Neighbor* (KNN) dengan akurasi 85.56%. Dalam metode menggunakan *bagging*, hasil akurasi tidak menunjukkan perbedaan yang signifikan dibandingkan dengan metode individual yang digunakan. Dengan demikian berdasarkan hasil evaluasi metode *Support Vector Machine* (SVM) memiliki performa akurasi paling tinggi dalam prediksi calon konsumen.

## **5.2 Saran**

Berdasarkan kesimpulan yang telah diperoleh, saran yang dapat diberikan untuk aplikasi ini selanjutnya adalah dengan memperbanyak variasi data dan lebih mendetailkan data untuk meningkatkan performa model *machine learning*. Dengan menggunakan *dataset* yang lebih beragam model dapat belajar dengan lebih baik. Setelah itu, untuk meningkatkan pengalaman pengguna dapat menambahkan beberapa fitur untuk website tersebut seperti fitur untuk visualisasi data dan melakukan evaluasi performa sistem secara berkala untuk memastikan sistem tetap optimal. Diharapkan aplikasi dapat terus berkembang dan memberikan manfaat dalam mendukung pengambilan keputusan dalam seleksi calon konsumen.

## **DAFTAR PUSTAKA**

- Abdurahman, H., & Riswaya, A. R. (2014). Aplikasi Pinjaman Pembayaran Secara Kredit Pada Bank Yudha Bhakti. *Jurnal Computech & Bisnis*.
- Akbar, M. F., & Samsudin. (2016). Sistem Pendukung Keputusan Kelayakan Pemberian Kredit Menggunakan Metode Decision Tree (Studi Kasus : Bank Pd. Bpr Gemilang Tembilahan). *Jurnal SISTEMASI*.
- Alexandromeo. (2022). Retrieved from Apa itu Flask? Pengertian, Kelebihan, dan Kegunaannya: <https://makinrajin.com/blog/flask-adalah/>
- Appkey. (2021). Retrieved from Apa itu PyCharm? Mengenal PyCharm Python, Kekurangan, serta Kelebihannya: <https://appkey.id/pembuatan-aplikasi/mobile-programming/pycharm-python/>
- Apurva. (2021). Retrieved from Ensemble Method in Python: <https://www.geeksforgeeks.org/ensemble-methods-in-python/>
- AWS. (2023). Retrieved from Apa itu Python?: <https://aws.amazon.com/id/what-is/python/>
- Bangun, O., Mawengkang, H., & Efendi, S. (2022). Metode Algoritma Support Vector Machine (SVM) Linier Dalam Memprediksi Kelulusan Mahasiswa. *Jurnal Media Informatika Budidarma*.
- Benefita. (2023). Retrieved from Pengertian HTML, Fungsi dan Cara Kerjanya: <https://www.niagahoster.co.id/blog/html-adalah/>
- Dey, D. (2023). Retrieved from ML Bagging Classifier: <https://www.geeksforgeeks.org/ml-bagging-classifier/>
- Fadilah, W. R., Agfiannisa, D., & Azhar, Y. (2020). Analisis Prediksi Harga Saham PT. Telekomunikasi Indonesia Menggunakan Metode Support Vector Machine. *Fountain of Informatics Journal*.
- Handayani, N., Wahyono, H., Trianto, J., & Permana, D. S. (2021). Prediksi Tingkat Risiko Kredit dengan Data Mining Menggunakan Algoritma Decision Tree C.45. *JURIKOM (Jurnal Riset Komputer)*.
- Hasbiyalloh, M., & Jakaria, D. A. (2018). Aplikasi Penjualan Barang Perlengkapan Hand Phone Di Zildan Cell Singaparna Kabupaten Tasikmalaya. *Jurnal Manajemen dan Teknik Informatika (JUMANTAKA)*.

- Indonesia, B. (2023). Retrieved from Data Kredit Usaha Mikro, Kecil, dan Menengah: <https://www.bi.go.id/umkm/kredit/default.aspx>
- Junaedi, E., Siregar, A. M., & Nurlaelasari, E. (2022). Implementasi C4.5 Dan Algoritma K Nearest Neighbor Untuk Prediksi Kelayakan Pemberian Kredit Menggunakan RapidMiner Studio. Scientific Student Journal for Information, Technology and Science.
- K, Y. (2022). Retrieved from Pengertian MySQL, Fungsi, dan Cara Kerjanya: <https://www.niagahoster.co.id/blog/mysql-adalah/>
- Kristianto, M. (2018). Klasifikasi Potensi Pembayaran Kredit Customer Dengan Metode C4.5 Pada PT. Autochem Industry. Fakultas Sains dan Teknologi, Universitas Buddhi Dharma, Tangerang: Tidak Diterbitkan.
- Lestari, W. J., Rahimah, Army, W. L., & Habibie, D. R. (2021). Analisa Risiko Kredit Macet Dengan Pendekatan Data Mining (Studi Kasus: Koperasi Putra Kembar). Jurnal Sistem Informasi dan Manajemen.
- Nasrullah, A. H. (2021). Implementasi Algoritma Decision Tree Untuk Klasifikasi Produk Laris. Jurnal Ilmiah Ilmu Komputer.
- Nirwana, A., Siregar, A. M., & Rahmat. (2022). Klasifikasi Permasalahan Kredit Macet Pada Bank Menggunakan Algoritma Decision Tree C4.5. Scientific Student Journal for Information, Technology and Science ISSN: 2715-2766.
- Nurellisa, L., & Fitrianah, D. (2020). Analisis Rekomendasi Calon Debitur Motor Pada PT. XYZ Menggunakan Algoritma C4.5. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK).
- Permana, A. A., & S, W. (2019). Machine Learning. In Machine Learning. PT Global Eksekutif Teknologi Anggota IKAPI.
- Pradana, T. P. (2023). Retrieved from 10 Kelebihan Bootstrap dan Cara Mudah Menggunakannya: <https://idwebhost.com/blog/bootstrap-adalah/>
- Rahmianti, I. (2022). Analisis Kelayakan Pemberian Kredit Koperasi Dengan Metode Data Mining Decision Tree. JIRE (Jurnal Informatika & Rekayasa Elektronika).
- Rani, L. N. (2016). Klasifikasi Nasabah Menggunakan Algoritma C4.5 Sebagai Dasar Pemberian Kredit. Jurnal INOVTEK POLBENG.

- Rilvani, E., Trisnawan, A. B., & Santoso, P. P. (2019). Penentuan Kelulusan Siswa Yayasan Cerdas Bakti Pertiwi Dengan Menggunakan Algoritma Naive Bayes dan Cross Validation. Pelita Teknologi Jurnal Ilmiah Informatika.
- Scikit-Learn. (2013). Retrieved from History from Scikit Learn: <https://scikit-learn.org/stable/about.html#history>
- Setiadi, B., & Lareno, B. (2015). Penerapan Algoritma Decision Tree Untuk Penilaian Agunan Pengajuan Kredit. Konferensi Nasional Sistem & Informatika.
- Setiadi, T. (2022). Retrieved from Apa Itu Machine Learning? Beserta Pengertian Dan Cara Kerjanya: <https://sistem-komputer-s1.stekom.ac.id/informasi/baca/Apa-itu-Machine-Learning-Beserta-Pengertian-dan-Cara-Kerjanya/a17ca13ab81788f8a4b73ad10ee4b3d3caac8cc3>
- Sudirman, B. (2022). Retrieved from Mengenal Aplikasi Postman Untuk Testing API: <https://teknik-informatika-s1.stekom.ac.id/informasi/baca/Mengenal-Aplikasi-Postman-Untuk-Testing-API/eb1f10d77482d9ff4016f0fcfd2c703edd2fcf548>
- T. P., Siregar, A. M., Masruriyah, A. F., & Juwita, A. R. (2020). Perbandingan Hasil Prediksi Kredit Macet Pada Koperasi Menggunakan Algoritma KNN dan C5.0. CIASTECH 2020.
- Tahir, M. A. (2019). Perancangan Aplikasi Data Mining Menggunakan Metode Decision Tree Untuk Analisa Pemberian Kredit Pada Bri Unit Lalabata Rilau. Jurnal Ilmiah Sistem Informasi dan Teknik Informatika “JISTI”.
- Triwijayati, A., & Pradipta, D. B. (2018). Kelas Sosial Vs Pendapatan: Eksplorasi Faktor Penentu Pembelian Consumer Goods dan Jasa. Jurnal Ekonomi.
- Wahyuningsih, S., & Utari, D. R. (2018). Perbandingan Metode K-Nearest Neighbor, Naïve Bayes dan Decision Tree untuk Prediksi Kelayakan Pemberian Kredit. Konferensi Nasional Sistem Informasi.
- Widjaja, D., Prilianti, K. R., & Setiawan, H. (2014). Sistem pendukung keputusan pemberian kredit rumah menggunakan analytical hierarchy process berbasis web. Symbol Vol.1 No.1.

Yuliana, A., & Pratomo, D. B. (2017). Algoritma Decision Tree (C4.5) Untuk Memprediksi Kepuasan Mahasiswa Terhadap Kinerja Dosen Politeknik Tedc Bandung. Seminar Nasional Inovasi Teknologi.