

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Teori**

##### **2.1.1 Pinjaman Kredit**

Pinjaman kredit merupakan salah satu bentuk pembiayaan yang memungkinkan pengaju untuk menerima sejumlah uang guna memenuhi kebutuhan atau tujuan masing-masing. Institusi-institusi keuangan seperti bank dan *credit union* menyediakan fasilitas pinjaman kredit dengan berbagai macam kriteria tersendiri yang harus dipenuhi oleh pengaju pinjaman [21]. Pengaju pinjaman kredit diharuskan membuat kesepakatan dengan pihak penyedia pinjaman secara hukum, dimana beberapa aspek penting didefinisikan, seperti bunga (*interest*), durasi pinjaman (*duration*) dan konsekuensinya bila pengaju gagal memenuhi perjanjian yang dibuat [22]. Tipe pinjaman kredit terdiri dari berbagai varian, yang antara lain mencakup tapi tidak terbatas pada, konsolidasi hutang (*debt consolidation*), hutang mahasiswa (*student*), hutang hipotek (*mortgages*), hutang pribadi (*personal*), hutang bisnis berskala kecil (*small business*), serta hutang uang muka (*cash advances*) [23].

##### **2.1.2 Kelayakan Nasabah**

Pada setiap bank, nasabah memiliki tingkat risiko kredit (*credit risk*) yang dijadikan sebagai dasar acuan kelayakan nasabah dalam pinjaman kredit yang diajukan. *Credit risk*, umumnya berupa skor yang memiliki rentang nilai spesifik, merupakan risiko kredit nasabah yang dihitung berdasarkan beberapa faktor latar belakang serta probabilitas kemampuan untuk membayar pinjaman kredit secara tepat waktu atau secara terlambat, yang disebut sebagai risiko *defaulting* [24]. Setiap bank mengimplementasikan sistem pengukuran risiko kredit yang berbeda-beda, baik menggunakan metode tradisional maupun metode modern, dimana kriteria yang wajib dipenuhi oleh nasabah juga bervariasi. Bila hasil perhitungan tingkat risiko nasabah tidak memenuhi kriteria yang

ditentukan, maka nasabah akan dianggap tidak layak dan pengajuan pinjaman kredit akan ditolak. Sebaliknya, bila hasil sistem perhitungan menunjukkan bahwa nasabah memiliki risiko kredit yang rendah, maka nasabah dianggap layak dan pengajuan pinjaman kredit akan disetujui.

Faktor penentu kelayakan nasabah pada penelitian ini didasari dengan riwayat kredit nasabah melalui nilai variabel “Credit History” yang ditemukan di dalam dataset nasabah Bank XY tahun 2021. Dimana bila nilai variabel “Credit History” adalah 0, maka nasabah memiliki riwayat kredit yang tidak memuaskan (buruk), sehingga dinilai tidak layak. Sementara, bila nilai variabel “Credit History” adalah 1, maka nasabah memiliki riwayat kredit yang memuaskan (baik).

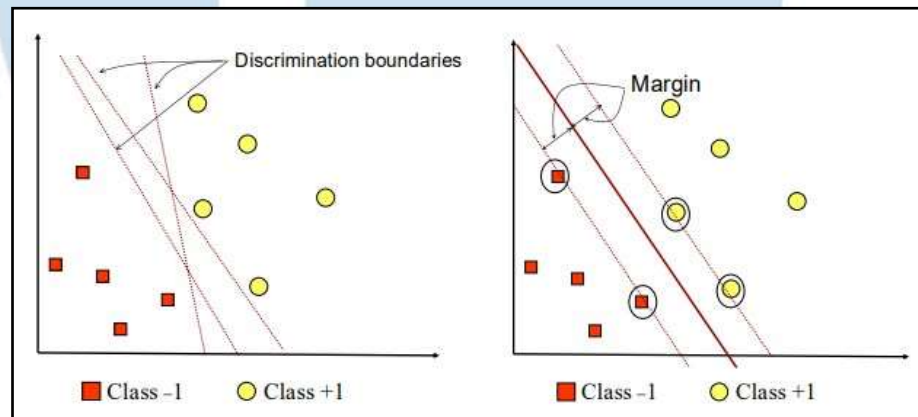
### **2.1.3 Pembelajaran Mesin**

Pembelajaran Mesin (*machine learning*) merupakan sebuah istilah yang pertama kali dikenalkan oleh Alan Turing pada tahun 1950, yang mengacu terhadap kemampuan program komputer untuk belajar dan berpikir layaknya seorang manusia secara tersendiri [25]. Bahan pembelajaran yang digunakan oleh mesin-mesin ini bersumber dari pola yang ditemukan pada data. Terdapat puluhan tipe pembelajaran mesin, dimana dua tipe yang paling umum digunakan adalah *supervised learning* dan *unsupervised learning* [26]. Menurut [26], *supervised learning* atau pembelajaran dengan pengawasan diterapkan ketika data terdiri dari variabel *input* dan nilai *target output*, dimana algoritma mempelajari fungsi pemetaan dari *input* serta *output*; memerlukan baik *input* maupun *output*. *Unsupervised learning* atau pembelajaran tanpa pengawasan diterapkan ketika data hanya terdiri dari variabel *input*, tanpa nilai *target output* yang diketahui, dimana algoritma melakukan pembelajaran berdasarkan pola yang ditemukan pada model; hanya memerlukan *input* [27]. Pada seluruh tipe pembelajaran, digunakan berbagai macam algoritma yang berbeda antar satu sama lain dengan keunggulan dan kekurangan masing-masing. Implementasi algoritma-algoritma pembelajaran mesin berfokus pada pembangunan model guna mencapai

beberapa tugas, seperti klasifikasi, regresi, prediksi dan pengelompokan (*clustering*) [25].

#### 2.1.4 Support Vector Machine

*Support Vector Machine*, juga sering disingkat menjadi *SVM*, merupakan salah satu algoritma *supervised learning* yang didasari dengan pencarian fungsi pemisah (*hyperplane*) terbaik ataupun yang paling tepat untuk memisahkan beberapa *class* yang ada pada dataset [28]. Metode yang dikembangkan oleh Boser, Guyon, Vapnik pada tahun 1992 ini menerapkan klasifikasi linear (*linear classification*) sebagai prinsip dasarnya [29].



Gambar 2.1. SVM Melakukan Pencarian *Hyperplane* Terbaik

Secara sederhana, konsep bekerja algoritma *SVM* adalah mencari *hyperplane* terbaik guna memisahkan dua buah *class* pada *input space*. Gambar 2.1. menunjukkan beberapa *data points* yang terbagi menjadi dua kelas, yaitu *class -1* dan *+1*. Sementara, garis-garis berwarna merah menunjukkan berbagai alternatif garis pemisah (*discrimination boundaries*) [28]. Pencarian *hyperplane* terbaik dapat dikalkulasikan dengan mengukur titik maksimal dari *margin*, dimana *margin* merupakan jarak antar garis-garis tersebut dengan *data points* terdekat. *Data points* terdekat ini yang ditandai dengan lingkaran pada grafik di sebelah kanan gambar 2.1. disebut sebagai *support vectors*. Garis berwarna merah tebal menandakan *hyperplane* terbaik yang berlaku sebagai pemisah antar kedua *class* (*separating hyperplane*) [29].

Langkah-langkah pencarian *separating hyperplane* yang paling optimal yang memerlukan berbagai macam kalkulasi dan fungsi, termasuk penggunaan fungsi Lagrange, serta beberapa detail penjelasan teknik *SVM* lainnya mencakup *kernel selection*, *hard-margin SVM* dan *soft-margin SVM* dapat didalami dengan mengacu kepada [29] dan [30].

### 2.1.5 Naïve Bayes

*Naive Bayes* adalah salah satu algoritma *supervised learning* yang digunakan untuk melakukan klasifikasi secara statistik dengan mengacu pada konsep probabilitas bersyarat pada Teorema Bayes [31] dan [32]. Metode yang dikembangkan dengan konsep dari Thomas Bayes ini memiliki asumsi bahwa terdapat independensi yang kuat antar *features* [33] dan [34]. Adapun persamaan umum dari Teorema Bayes yang digunakan dalam algoritma *Naive Bayes* adalah sebagai berikut.

$$P(c | x) = P(x | c)P(c)P(x) \quad (2.1)$$

Dengan  $P(c | x) = P(x_1 | c) P(x_2 | c) \dots P(x_n | c) P(c)$

Dimana  $P(c | x)$  adalah *posterior probability* dari *class (target)*,  $P(c)$  adalah *prior probability* dari *class*,  $P(x | c)$  adalah *likelihood* yakni probabilitas dari prediktor, dan  $P(x)$  adalah *prior probability* dari prediktor [35]. Secara sederhana, Teorema Bayes juga dapat ditulis dengan rumus sebagai berikut.

$$\text{Posterior} = \text{Prior} \times \text{Likelihood} / \text{Evidence} \quad (2.2)$$

### 2.1.6 Random Forest

*Random Forest* merupakan salah satu algoritma *ensembled learning*, yang artinya bersifat mengkombinasikan algoritma-algoritma baik yang berbeda maupun yang sama secara berulang kali untuk membentuk model prediksi yang lebih kuat [36]. Algoritma yang diperkenalkan pertama kali oleh Leo Breiman ini seringkali digunakan untuk menyelesaikan permasalahan klasifikasi dan regresi [37]. Cara bekerja *Random Forest* didasari oleh kumpulan algoritma *Decision Tree* yang dibangun menggunakan dua metode utama yang umum ditemukan

pada *ensembled learning* yaitu *bagging* dan *boosting*.

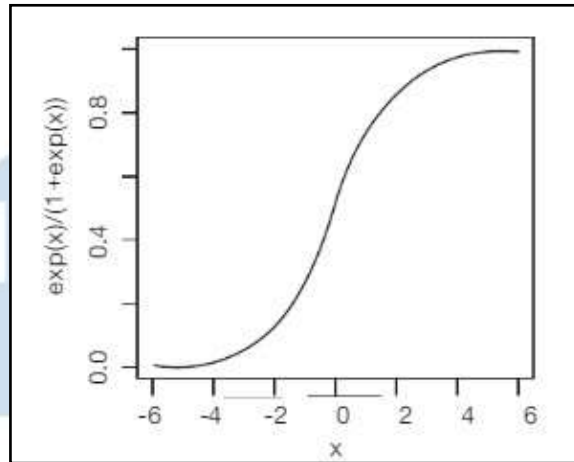
*Bagging*, yang merupakan singkatan dari *bootstrap aggregating*, adalah salah satu metode *ensembled learning* untuk meningkatkan estimasi yang tidak stabil pada skema klasifikasi [38]. Breiman menyebutkan bahwa *bagging* merupakan sebuah teknik pengurangan varians (*variance reduction*) untuk prosedur-prosedur dasar seperti *Decision Tree* [38]. Selain dapat meningkatkan kinerja atau performa model prediktif pada kasus klasifikasi dan regresi, *bagging* juga dapat mengurangi *mean squared error* yang dihasilkan oleh model [38].

Tidak seperti *bagging* yang merupakan metode *parallel ensembled*, *boosting* merupakan metode *sequential ensembled*, dimana sebuah *base model* memiliki ketergantungan dengan model sebelumnya yang sudah dibangun untuk dikombinasikan agar memiliki tingkat akurasi yang tertinggi [39].

### **2.1.7 Logistic Regression**

*Logistic Regression* merupakan salah satu algoritma regresi yang menganalisis hubungan antar beberapa variabel independen (*independent variables*) dengan sebuah variabel dependen kategorikal (*categorical dependent variable*) [40]. *Logistic Regression* menghitung estimasi atau memperkirakan probabilitas terjadinya suatu peristiwa (*event*) dengan melakukan *fitting* model pada kurva logistik [40].





**Gambar 2.2. Kurva Logistik**

*Logistic Regression* memiliki dua model yang berbeda, yaitu *Binary Logistic Regression* dan *Multinomial Logistic Regression*; *Binary Logistic Regression* biasanya digunakan ketika hanya terdapat dua kategori dimana variabel dependen bertipe *dichotomous* (terbagi menjadi dua) dan variabel independen bertipe *continouous* atau *categorical*, sementara *Multinomial Logistic Regression* biasanya digunakan ketika variabel dependen tidak bertipe *dichotomous* dan terdapat lebih dari dua kategori [40].

### **2.1.8 K-Nearest Neighbors**

*K-Nearest Neighbors (K-NN)* merupakan sebuah algoritma klasifikasi atau regresi yang bersifat *non-parametic* dan dapat diimplementasikan secara mudah, namun tetap efektif [41]. *K-NN* melakukan klasifikasi berdasarkan *similarity measure*, yaitu perhitungan jarak antar satu titik dengan yang lainnya. Algoritma ini memerlukan pemilihan jumlah *reference point* atau jumlah tetangga (*neighbors*) yaitu *k*.

Hal pertama yang harus dilakukan adalah menentukan jumlah *k* yang ingin digunakan. Perhitungan jumlah *k* yang optimal dapat menggunakan rumus *Euclidean Distance* yang ditunjukkan pada gambar 2.3. [42]. Setelah menentukan jumlah *k* yang akan digunakan, *data points*

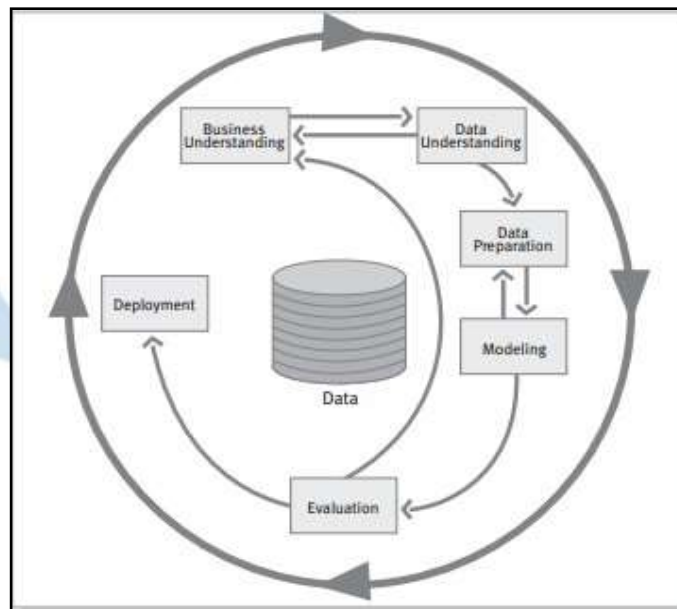
yang berada di dalam jangkauan (terdekat)  $k$  dengan titik yang dipilih akan dijadikan data *training* dan diobservasi, dimana kelas mayoritas yang didapatkan akan mendefinisikan *data point* yang terpilih tersebut.

$$EuDis(m_1, m_2) = \sqrt{\sum_{i=\{X,Y,Z\}} (m_{1,i} - m_{2,i})^2}$$

Gambar 2.3. Rumus Euclidean Distance

### 2.1.9 CRISP-DM

*CRISP-DM*, sebuah akronim untuk *Cross Industry Standard Process for Data Mining*, merupakan sebuah teknik *data mining* yang populer digunakan untuk mengatasi permasalahan-permasalahan *data science* [20]. Metodologi *CRISP-DM* pertama kali disusun pada akhir tahun 1996 oleh DaimlerChrysler, SPSS dan NCR. Menurut [20], siklus hidup metodologi *CRISP-DM* didirikan oleh 6 fase utama berurut yang saling berhubungan sebagaimana yang digambarkan pada gambar 2.4..



Gambar 2.4. Siklus Hidup *CRISP-DM*

*Business understanding* atau pemahaman bisnis sebagai fase pertama berfokus pada pemahaman tujuan dan persyaratan proyek dari perspektif bisnis. Pengetahuan yang telah diperoleh diubah menjadi definisi permasalahan *data mining*, dimana kemudian dimulai perancangan rencana awal untuk mencapai objektif yang ditetapkan.

*Data understanding* atau pemahaman data sebagai fase kedua dimulai dengan mengumpulkan data dan dilanjutkan dengan melaksanakan kegiatan-kegiatan yang mampu membantu memahami data, mengidentifikasi permasalahan kualitas data, menemukan *insights* awal terkait data ataupun membentuk hipotesis yang menarik.

*Data preparation* atau preparasi data sebagai fase ketiga mencakup semua kegiatan yang diperlukan untuk membangun dataset yang siap digunakan untuk *modeling*. Proses preparasi data berkemungkinan besar untuk dilakukan secara berulang kali, dimana beberapa kegiatan di antaranya meliputi tetapi tidak terbatas pada transformasi dan pembersihan data mentah.

*Modeling* atau pembangunan model sebagai fase keempat merupakan proses pemilihan dan penerapan berbagai macam algoritma untuk membangun model yang diinginkan guna memecah permasalahan *data mining*. Dalam fase ini, parameter-parameter dikalibrasikan agar memiliki nilai-nilai yang optimal.

*Evaluation* atau evaluasi sebagai fase kelima merupakan proses mengevaluasi dan meninjau ulang secara menyeluruh hasil akhir dan performa model yang telah dirancang. Fase evaluasi bertujuan untuk memastikan bahwa model yang dibangun mampu mencapai objektif yang ditetapkan sebelumnya dan memecahkan permasalahan *data mining*.



*Deployment* atau peluncuran sebagai fase keenam dan terakhir merupakan proses ketika model diaplikasikan secara *live* di dunia nyata dalam proses pengambilan keputusan suatu organisasi oleh *end-user*. Model yang telah diluncurkan dan dipakai pada dunia nyata akan menghasilkan berbagai macam wawasan (*insights*) dan pengetahuan (*knowledge*) baru yang bernilai atau memiliki *value*. Alhasil, wawasan dan pengetahuan baru yang diperoleh mampu membantu dalam proses pembuatan keputusan bisnis, serta merekomendasikan aksi-aksi baru yang perlu diambil (*actions*).

#### **2.1.10 Python**

Python merupakan sebuah bahasa pemrograman tingkat tinggi dan serba guna yang bersifat gratis, *open source* serta *cross platform* [43]. Diciptakan oleh Guido van Rossum pada tahun 1991, Python meraih ketenaran selama 10 tahun terakhir dan membuatnya sebagai salah satu bahasa pemrograman paling populer, menduduki peringkat ketiga dalam kategori bahasa pemrograman yang paling banyak digunakan berdasarkan peringkat pada IEEE Spectrum [43].

Beberapa karakteristik Python lainnya menurut [43], mendeskripsikan Python sebagai bahasa pemrograman yang *highly extendable* dikarenakan memiliki *package* dan *library* yang melimpah. Kemudian, Python dapat digunakan pada semua perangkat lunak, baik Windows, macOS maupun Linux. Python mampu dipakai untuk memprogram aplikasi web, aplikasi perusahaan (*enterprise*) maupun aplikasi tertanam (*embedded*).

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

### 2.1.11 Feature Selection

*Feature selection* atau pemilihan fitur merupakan sebuah teknik pengurangan dimensi yang bertujuan untuk memilih sebuah *subset* kecil yang berisikan fitur-fitur yang relevan, melalui pemindahan fitur-fitur yang tidak relevan, redundan dan *noisy* [44]. Beberapa manfaat yang dihasilkan oleh teknik *feature selection* mencakup peningkatan pada performa algoritma pembelajaran, mengurangi waktu komputasi, mengurangi biaya komputasi, serta mengidentifikasi fitur relevan untuk setiap masalah tertentu [45]. Umumnya, metode *feature selection* terbagi menjadi tiga klasifikasi, yaitu metode *filter*, *wrapper* dan *embedded* [45].

Menurut [46], metode *filter* memilih fitur berdasarkan metrik performa yang digunakan, terlepas dari algoritma pembelajaran mesin yang sedang diimplementasikan. Metode *filter* sangat hemat dari segi biaya komputasi dan mampu mencegah terjadinya permasalahan *overfitting*. Namun, metode *filter* menghiraukan *dependencies* antara fitur sehingga *subset* terpilih yang dihasilkan memiliki kemungkinan untuk menjadi tidak optimal dan menjadi redundan. Beberapa teknik yang tergolong ke dalam metode *filter* adalah *Information Gain*, *Chi-square Test*, *Correlation-Based Feature Selection (CBFS)* dan *Euclidean Distance*.

Menurut [45], metode *wrapper* menggunakan teknik *backward elimination* untuk menghilangkan fitur-fitur tidak relevan dari sebuah *subset*. Tidak seperti metode *filter*, metode *wrapper* bekerja dengan melibatkan algoritma pembelajaran mesin yang diterapkan untuk mengidentifikasi fitur-fitur yang relevan. Metode *wrapper* memakan lebih banyak biaya dan waktu komputasi jika dibandingkan dengan metode *filter*. Namun, metode *wrapper* mampu menghasilkan performa akurasi yang lebih optimal dan menjaga *dependencies* antar fitur. Salah satu contoh teknik yang tergolong ke dalam metode *wrapper* adalah *Support Vector Machine-Recursive Feature Elimination (SVM-RFE)*.

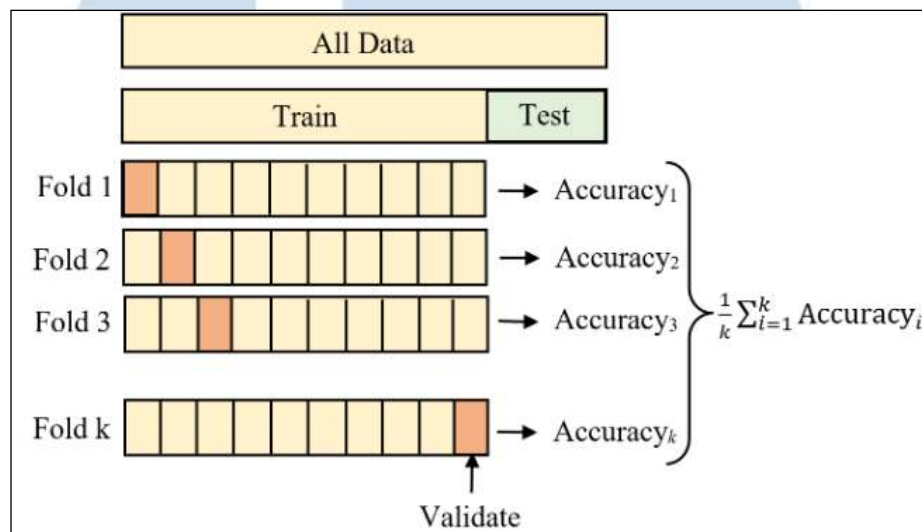
Menurut [47], metode *embedded* merupakan campuran atau gabungan metode *filter* dan *wrapper*. Metode *embedded* mengikutsertakan algoritma pembelajaran mesin seperti metode *wrapper*, dan sangat efisien dari segi waktu komputasi sejak tidak memerlukan evaluasi *subset* secara berulang kali seperti metode *filter*. Teknik-teknik populer yang tergolong ke dalam metode *embedded* adalah *CART*, *C4.5*, *Random Forest* dan *Multinomial Logistic Regression* [46].

### 2.1.12 Hyperparameter Tuning

*Hyperparameter tuning* merupakan proses pendefinisian parameter yang wajib ditentukan sebelum melatih model pembelajaran mesin, sejak parameter-parameter ini mendefinisikan arsitektur model yang akan dilatih [48]. Menurut [48], beberapa alasan yang melandaskan tujuan penerapan *hyperparameter tuning* pada model pembelajaran mesin antara lain mencakup: untuk mengurangi usaha dan waktu yang dikeluarkan oleh pengembang pembelajaran mesin, meningkatkan performa model pembelajaran mesin pada berbagai dataset dan permasalahan, serta membantu mengidentifikasi model terbaik untuk permasalahan tertentu. Tidak seperti parameter normal yang dipelajari selama masa pelatihan model, *hyperparameter* diatur oleh pengguna secara manual, dimana proses pengaturan tersebut dikenal sebagai istilah *tuning* [49]. Tujuan akhir dari proses *hyperparameter tuning* adalah mencari nilai terbaik untuk *hyperparameter* tertentu yang ditemukan dalam sebuah algoritma pembelajaran mesin.

Salah satu proses terpenting dalam *hyperparameter tuning* adalah *Cross Validation (CV)*. Menurut [50], *Cross Validation* merupakan sebuah metode statistik untuk mengestimasi akurasi model pembelajaran mesin. *Cross Validation* memiliki beberapa teknik; salah satu teknik terpopuler bernama *K-Fold*, dimana parameter *K* mengindikasikan jumlah lipatan (*fold*). Teknik *K-Fold Cross Validation* bekerja dengan menjadikan salah satu *fold* sebagai set validasi, dimana pembelajaran mesin dilatih

menggunakan  $K-1$  *fold* yang tersisa. Pada satu titik, seluruh *fold* akan digunakan sebagai set validasi, dimana nilai  $K$  dihasilkan yang berupa akurasi. Pada akhirnya, skor akhir model akan diperoleh melalui rata-rata performa model pada setiap *fold*. Gambar 2.5. menggambarkan proses *K-Fold Cross Validation* secara jelas.



Gambar 2.5. Proses *K-Fold Cross Validation*

Berdasarkan [51], terdapat sejumlah teknik *hyperparameter tuning* yang umum digunakan, yang mencakup tetapi tidak terbatas pada teknik *manual search*, *grid search*, *randomized search*, *halving grid search*, *halving randomized search*, *hyperopt-sklearn* dan *bayes grid search*. *Manual search* memerlukan pengguna untuk mencoba beberapa kombinasi nilai-nilai terbaik untuk setiap parameter secara tersendiri. *Grid search* dapat dianggap sebagai versi otomatis dari *manual search*, dimana sebuah *library* Python mendukung pencarian kombinasi nilai-nilai terbaik secara otomatis yaitu *library* GridSearchCV. *Randomized search* melatih kombinasi parameter secara teracak, dimana sebuah *library* Python juga tersedia untuk fungsi ini yang bernama RandomizedSearchCV. *Halving grid search* menerapkan pendekatan algoritma *successful halving*, yang mulai bekerja dengan mengevaluasi seluruh kandidat kombinasi parameter dalam sebuah sampel yang kecil, dan ukuran sampel mulai bertumbuh

seiring dengan bertambahnya iterasi. *Halving randomized search* menggunakan pendekatan algoritma yang sama seperti *halving grid search*, yaitu *successful halving*, dan diterapkan pada kombinasi parameter teracak. *Hyperopt-sklearn* merupakan sebuah *library* Python yang bersifat khusus untuk *Bayesian optimization*, dirancang secara spesifik untuk model dengan ratusan jumlah parameter. *Bayes grid search* memanfaatkan teknik *Bayesian optimization* untuk menemukan parameter terbaik secepat mungkin, dengan menggunakan hasil evaluasi terdahulu untuk memilih kandidat parameter yang memiliki kemungkinan tertinggi untuk menghasilkan nilai terbaik.

## 2.2 Penelitian Terdahulu

Sebagai bahan acuan dan referensi dalam penulisan penelitian “Klasifikasi Persetujuan Pinjaman Kredit Berdasarkan Kelayakan Nasabah Bank: Pendekatan Menggunakan Berbagai Algoritma Pembelajaran Mesin (Studi Kasus: Bank XY)”, digunakan beberapa penelitian terdahulu, yaitu [52], [53], [24] dan [54], yang antara lain dijelaskan dalam tabel 2.1. berikut:

**Tabel 2.1. Penelitian Terdahulu**

Judul Jurnal	Nama Jurnal	Penulis / Tahun	Kesimpulan
Predicting Bank Loan Risks Using Machine Learning Algorithms	Raf. J. of Comp. & Math's, Vol 14, No. 1	Maan Y. Alsaleem, Safwan O. Hasoon / 2020	Penerapan beragam metrik evaluasi model yang berhasil dibangun yang mencakup penggunaan metrik <i>Root Mean Squared Error (RMSE)</i> , <i>Root Relative Squared Error (RRSE)</i> , <i>kappa statistic</i> , <i>TP</i> dan <i>FP rate</i> , <i>precision</i> , <i>recall</i> , <i>F-measure</i> , <i>MCC</i> , <i>ROC area</i> serta <i>PRC area</i> .
An Empirical Comparison of Machine- Learning Methods on Bank Client Credit Assessments	Sustainability 2019, 11(3), 699	L. Munkhdalai, T. Munkhdalai, Oyun-Erdene Namsrai, Jong Yun Lee, Keun Ho Ryu / 2019	Tahap <i>preprocessing data</i> hanya berfokus pada menjalankan tugas seleksi variabel ( <i>feature selection</i> ) dan <i>hyperparameter tuning</i> yang terbaik, agar model yang dibangun memiliki tingkat akurasi yang semaksimal



			<p> mungkin. Algoritma <i>TSFFS</i> dan <i>NAP</i> dimanfaatkan untuk <i>feature selection</i>.</p>
<p>Loan default prediction using decision trees and random forest: A comparative study</p>	<p>IOP Conf. Series: Materials Science and Engineering 1022 (2021) 012042</p>	<p>Mehul Madaan, Aniket Kumar, Chirag Keshri, Rachna Jain, Preeti Nagrath / 2021</p>	<p>Penggunaan <i>Confusion Matrix</i> sebagai satu-satunya metrik evaluasi. Kedua, <i>NULL values</i> yang ditemukan tidak dibersihkan atau dibiarkan saja. Serta, penggunaan berbagai macam grafik pada proses <i>Exploratory Data Analysis</i> untuk mengetahui lebih dalam terkait karakteristik variabel yang ditemukan dalam dataset.</p>
<p>Loan Default Prediction using Supervised Machine Learning Algorithm</p>	<p>TRITA-SCI-GRU 2019:073, MAT-E 2019:30</p>	<p>Daria Granstrom, Johan Abrahamsson / 2019</p>	<p>Teknik <i>feature selection</i> yang digunakan mencakup metode analisis korelasi dengan <i>Kendall's Tau Coefficient</i> dan <i>Recursive Feature Elimination</i>. Algoritma <i>SMOTE</i> digunakan untuk mengatasi data yang tidak seimbang (<i>imbalanced data</i>). Dataset dibagi menjadi 3 yaitu <i>training</i>, <i>validation</i> dan <i>test</i>. Teknik validasi model yang digunakan adalah metode <i>K-fold Cross Validation</i> dengan <i>sensitivity</i>, <i>precision</i>, <i>F-score</i> dan <i>ROC score</i> sebagai metrik evaluasi utama.</p>

Berdasarkan penelitian-penelitian terdahulu, terdapat beberapa *novelty* atau kebaruan yang membedakan penelitian “Klasifikasi Persetujuan Pinjaman Kredit Berdasarkan Kelayakan Nasabah Bank: Pendekatan Menggunakan Berbagai Algoritma Pembelajaran Mesin (Studi Kasus: Bank XY)” dengan sejumlah penelitian pada tabel 2.1., yaitu teknik *feature selection* yang diimplementasikan adalah melakukan analisis korelasi (*correlation analysis*) antar variabel menggunakan teknik Pearson dan Spearman. Perbandingan hasil performa model antar model yang mengimplementasikan *feature selection* dan

*hyperparameter tuning* dengan model yang tidak mengimplementasikan *feature selection* dan *hyperparameter tuning* dilakukan pada setiap algoritma. Sebagai bentuk *deployment*, dibangun purwarupa (*prototype*) yang mampu mengklasifikasikan status pinjaman kredit berdasarkan *input* data dari pengguna menggunakan algoritma pemenang.

