

## BAB II

### LANDASAN TEORI

#### 2.1 Machine Learning

*Machine learning* adalah ilmu yang mempelajari algoritma komputer yang dapat mengenali pola-pola dalam data dengan tujuan untuk mengubah berbagai jenis data menjadi tindakan yang nyata dengan sedikit kemungkinan campur tangan manusia. Menurut IBM, *machine learning* merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) dan ilmu komputer yang berfokus pada penggunaan data dan algoritma untuk meniru cara manusia belajar dan secara bertahap dapat meningkatkan akurasi. Semakin bagus algoritma *machine learning* yang digunakan maka akan semakin baik pula keputusan yang keluar [1].

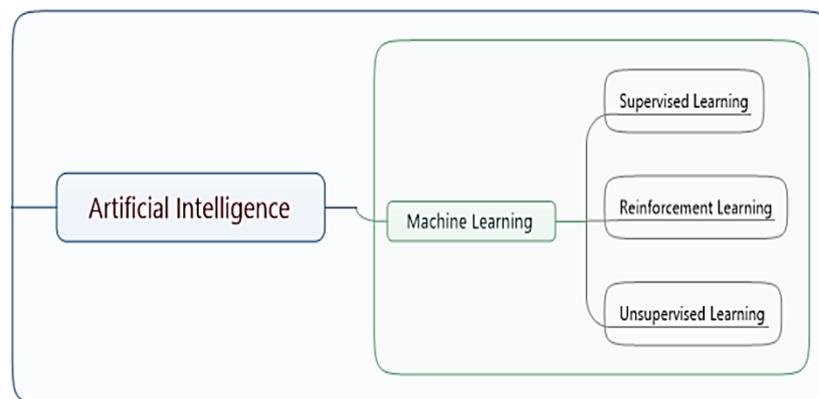
Algoritma *machine learning* ini juga dapat mempelajari data dari data historis saja. Meskipun *machine learning* menjanjikan kesuksesan dalam pemrograman, ada beberapa masalah yang harus diperhatikan saat menggunakannya. *Machine learning* merujuk pada sebuah metode yang membuat komputer memiliki kemampuan dalam mempelajari dan melakukan sebuah pekerjaan secara otomatis. Proses *machine learning* dilakukan melalui algoritma tertentu, sehingga pekerjaan yang diperintahkan kepada komputer dapat dilakukan secara otomatis [2].

Machine learning dilakukan melalui 2 fase, yaitu fase *training* dan fase *application*. Fase *training* adalah proses pemodelan dari algoritma yang digunakan akan dipelajari oleh sistem melalui training data, sedangkan fase *application* adalah

proses pemodelan yang telah dipelajari sistem melalui fase training akan digunakan untuk menghasilkan sebuah keputusan tertentu, dengan menggunakan testing data.

### 2.1.1 Teknik Machine Learning

Penelitian terkini mengungkapkan bahwa *machine learning* terbagi menjadi tiga kategori yaitu: *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning* [3]. Skema keterkaitan *artificial intelligence* dan *machine learning* dapat dijelaskan dalam Gambar 2.1.



**Gambar 2. 1** Skema Artificial Intelligence dan Machine Learning [3]

Teknik yang digunakan oleh *Supervised Learning* adalah metode klasifikasi di mana kumpulan data sepenuhnya diberikan label untuk mengklasifikasikan kelas yang tidak dikenal. Sedangkan teknik *Unsupervised Learning* sering disebut *cluster* dikarenakan tidak ada kebutuhan untuk pemberian label dalam kumpulan data dan hasilnya tidak mengidentifikasi contoh di kelas yang telah ditentukan [4]. Sedangkan *Reinforcement Learning* biasanya berada antara *Supervised Learning* dan *Unsupervised Learning*, teknik ini bekerja dalam lingkungan yang dinamis di mana konsepnya harus menyelesaikan tujuan tanpa adanya pemberitahuan dari komputer secara eksplisit jika tujuan tersebut telah tercapai.

Metode *supervised learning* didasarkan pada kumpulan sampel data yang memiliki label. Kumpulan sampel digunakan untuk meringkas karakteristik distribusi ukuran perilaku dalam setiap jenis aplikasi sehingga membentuk model perilaku dari data. *Supervised learning* dikelompokkan lebih lanjut dalam masalah klasifikasi dan regresi. Masalah klasifikasi adalah ketika variabel output berbentuk kategori, seperti merah atau biru atau penyakit dan tidak ada penyakit. Sedangkan masalah regresi adalah ketika variabel output adalah nilai riil, seperti dollar atau berat [3].

*Supervised learning* memiliki beberapa algoritma populer seperti Back-propagation, Linear regression, Random Forest, Support Vector Machines, *Naïve Bayesian*, Metode Rocchio, Decision Tree, k-Nearest Neighbor, Neural Network, Logistic Regression, dan Neural Network. Kemudian beberapa algoritma untuk klasifikasi pun disebutkan dalam seperti Support Vector Machines (SVM), Normal Bayesian Classifier (NBC), K-Nearest Neighbor (KNN), Trees Gradient Boosted (GBT), Random Trees (RT), dan Artificial Neural Networks (ANN). Algoritma lainnya pun dibahas dalam seperti Gaussian Mixture models, Hidden Markov Models, logistic regression, Kernel Regression, Deep neural networks, Deep belief networks, PCA, Kernel Perceptron.

Dalam jenis pembelajaran *Unsupervised Learning*, sistem disediakan dengan beberapa input sampel tetapi tidak ada output yang hadir. Karena tidak ada output yang diinginkan di sini kategorisasi dilakukan sehingga algoritma membedakan dengan benar antara kumpulan data. Ini adalah tugas mendefinisikan fungsi untuk menggambarkan struktur yang tersembunyi dari data yang tidak

berlabel. *Unsupervised learning* dikelompokkan lebih lanjut dalam masalah clustering dan asosiasi. Masalah pengelompokan (*clustering*) adalah tempat untuk menemukan pengelompokan yang melekat dalam data, seperti mengelompokkan pelanggan berdasarkan pada perilaku pembelian. Sedangkan masalah asosiasi adalah aturan yang menggambarkan sebagian besar data yang ada, seperti orang yang membeli A juga cenderung membeli.

*Reinforcement learning* berasal dari teori belajar hewan. Pembelajaran ini tidak memerlukan pengetahuan sebelumnya, dapat secara mandiri mendapatkan kebijakan opsional dengan pengetahuan yang diperoleh melalui coba-coba dan terus berinteraksi dengan lingkungan yang dinamis. Masalah reinforcement learning diselesaikan dengan mempelajari pengalaman baru melalui trial-and-error. Algoritma reinforcement learning terkait dengan algoritma pemrograman dinamis yang sering digunakan untuk menyelesaikan masalah optimisasi. Banyak masalah machine learning dalam dunia nyata termasuk dalam kategori ini. Hal ini dikarenakan bisa mencapai harga yang mahal bahkan memakan waktu untuk pemberian label pada data terkait kemungkinan untuk memerlukan akses ke bagian pakar. Padahal data yang tidak memiliki label itu didapatkan dengan harga yang murah dan mudah untuk dikumpulkan dan disimpan.

Secara khusus metode reinforcement learning berdasarkan model proses pengambilan keputusan Markov (*Markov decision process*) mencakup dua jenis. Pertama yaitu metode berbasis model seperti algoritma SARSA, dimana reinforcement learning pertama kali mempelajari pengetahuan model, kemudian mendapatkan strategi yang optimal dari pengetahuan model tersebut. Kedua adalah

metode yang relevan model seperti algoritma Temporal Difference dan algoritma Q-learning, di mana reinforcement learning secara langsung menghitung strategi yang optimal tanpa pengetahuan model [3].

## 2.2 *Naïve Bayes*

*Naïve Bayes* adalah salah satu algoritma pembelajaran induktif yang paling efektif dan efisien untuk *machine learning* dan data mining. *Naïve Bayes* merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang dimasa depan berdasarkan pengalaman dimasa sebelumnya sehingga dikenal sebagai Teorema Bayes [5]. *Naïve Bayes* merupakan Teknik prediksi berbasis probabilistik sederhana yang merujuk pada penerapan teorema Bayes dengan asumsi independensi yang kuat. Metode Bayes menggunakan probabilitas bersyarat sebagai dasarnya. Algoritma *Naïve Bayes* didasarkan pada fungsi probabilitas untuk setiap instance dan menjelaskan pemetaan antara atribut dan klasifikasi atribut dalam sistem yang memiliki efisiensi klasifikasi stabil dan kompleksitas rendah. *Naïve Bayes* mengasumsikan bahwa nilai atribut saling independen [6].

*Naïve Bayes* ini bisa dikategorikan ke dalam pembelajaran *supervised learning*, dimana *supervised learning* merupakan teknik *machine learning* yang membuat mesin dapat mengidentifikasi label input dengan menggunakan pembelajaran pada data trining yang sudah diberi label untuk melakukan prediksi atau klasifikasi sehingga pada tahapan pembelajaran *Naïve Bayes* memerlukan data pelatihan untuk mendapatkan hasil probabilitas. Setelah memperoleh hasil

probabilitas, tahapan berikutnya adalah proses pengklasifikasian, dimana dengan menggunakan data testing akan dihitung [7]. *Naïve Bayes* merupakan sebuah pengklasifikasian data secara probabilistik sederhana yang menghitung beberapa probabilitas dengan mengkombinasikan frekuensi nilai dari sebuah dataset. Proses metode ini dihitung dari kemiripan probabilitas kasus lama dengan kasus yang baru [8]. Teorema Bayes memiliki persamaan umum sebagai berikut [5]:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \quad (1)$$

Dimana pada persamaan (1),

- X : Data class yang belum diketahui
- H : Hipotesis data X merupakan suatu class spesifik
- P(H) : Probabilitas dari hipotesis H
- P(X) : Probabilitas dari X
- P(H|X) : Probabilitas hipotesis H berdasarkan kondisi X
- P(X|H) : Probabilitas X berdasarkan kondisi pada hipotesis H

Berdasarkan persamaan (1) teorema bayes maka inilah rumus dari *Naïve Bayes Classifier* [9]:

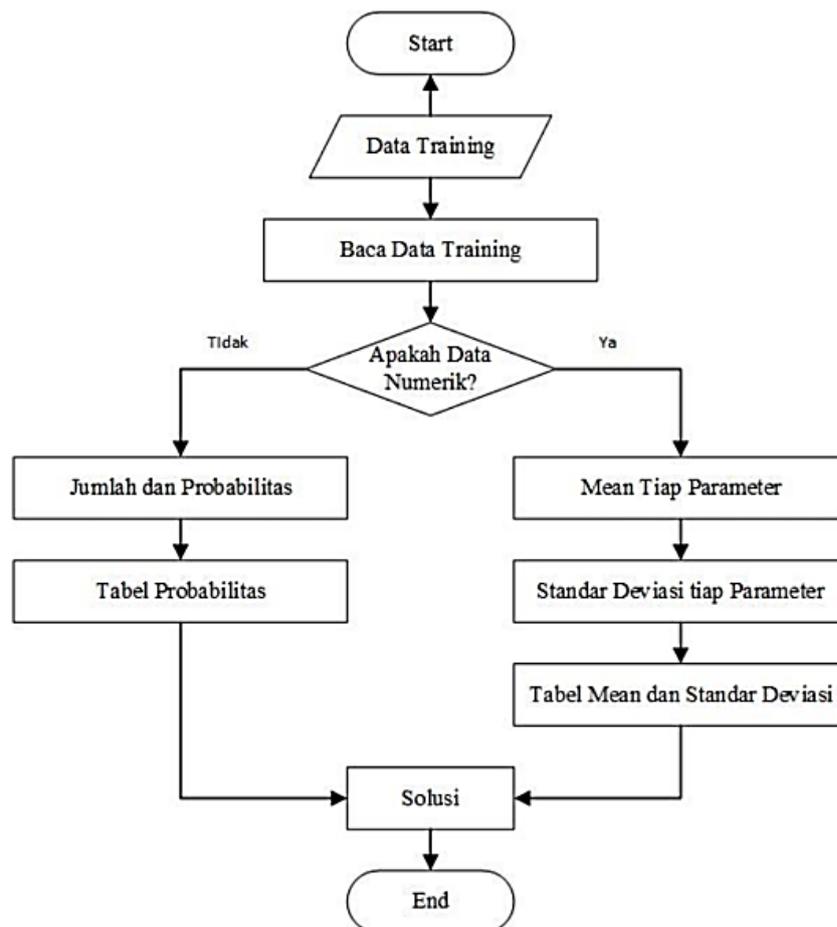
$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \quad (2)$$

Dimana pada persamaan (2),

- X : Data class yang belum diketahui
- C<sub>i</sub> : Hipotesis data X merupakan suatu class spesifik
- P(C<sub>i</sub>|X) : Probabilitas hipotesis C<sub>i</sub> berdasarkan kondisi X
- P(C<sub>i</sub>) : Probabilitas hipotesis C<sub>i</sub>

$P(X|C_i)$  : Probabilitas X berdasarkan kondisi pada hipotesis  $C_i$

$P(X)$  : Probabilitas X



**Gambar 2. 2** Alur Kerja Algoritma *Naïve Bayes* [10]

Dari gambar 2.2 dapat dijelaskan alur kerja dari algoritma *Naïve Bayes* dijelaskan sebagai berikut [10]:

1. Baca data *training*.
2. Hitung jumlah atau probabilitas, namun apabila data numerik maka:
3. Cari nilai mean dan standar deviasi dari masing-masing parameter yang merupakan data numerik.

4. Cari nilai probabilitas dengan cara menghitung jumlah data yang sesuai dari dari kategori yang sama dibagi dengan jumlah data pada kategori tersebut.
5. Mendapatkan nilai dalam table mean, standar deviasi dan probabilitas.

### 2.3 Confusion Matrix

*Confusion Matrix* adalah alat (*tools*) visualisasi yang biasa digunakan pada *supervised learning*. Tiap kolom pada matrix adalah contoh kelas prediksi, sedangkan tiap baris mewakili kejadian di kelas yang sebenarnya. Metode ini hanya menggunakan tabel matriks terdapat pada proses jika dataset memiliki kelas yaitu kelas yang dianggap positif dan kelas lainnya merupakan kelas negatif. Evaluasi dengan *confusion matrix* ini menghasilkan nilai akurasi, *precision*, dan *recall* terhadap klasifikasi yang telah dilakukan [9]. Akurasi dalam klasifikasi merupakan presentasi ketepatan *record* data diklasifikasikan secara benar setelah dilakukan pengujian pada hasil klasifikasi. *Precision* merupakan proposikasi yang diprediksi positif yang juga positif benar pada data sebenarnya. *Recall* merupakan proporsi kasus positif yang sebenarnya diprediksi positif secara benar [10].

*Confusion matrix* adalah sebuah tabel yang menyatakan jumlah data uji yang benar diklasifikasikan dan jumlah data uji yang salah diklasifikasikan. Confusion matrix adalah tool yang berguna untuk menganalisis seberapa baik *classifier* mengenali kelas yang berbeda. TP dan TN menjelaskan ketika pengklasifikasi mendapatkan sesuatu dengan benar, sementara FP dan FN menjelaskan ketika pengklasifikasi mendapatkan hal yang salah [11]. Contoh *confusion matrix* untuk klasifikasi biner ditunjukkan pada tabel 2.1.

**Tabel 2. 1** Confusion Table

Confusion Matrix		Nilai Prediksi	
		Positive	Negative
Nilai Sebenarnya	Positive	TP	FN
	Negative	FN	TN

Persamaan *Accuracy* :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

Persamaan *Recall* :

$$Recall = \frac{TP}{FN+TP} \quad (4)$$

Persamaan *Precision* :

$$Precision = \frac{TP}{FP+TP} \quad (5)$$

Persamaan F-I Score

$$F1Score = 2 \times \frac{precision \times recall}{precision+recall} \quad (6)$$

Keterangan:

- TP = *True Positive*
- TN = *True Negative*
- FP = *False Positive*
- FN = *False Negative*

*True positive* dan *false positive* adalah jumlah *record* positif dan negatif yang diklasifikasikan sebagai positif, sedangkan *false negative* dan *true negative* adalah jumlah *record* positif dan negatif yang diklasifikasikan sebagai negatif. Lalu masukkan data uji, setelah itu hitung nilai yang telah dimasukkan tersebut untuk

dihitung sensitivitasnya, spesifikasinya, presisinya dan akurasi. Berdasarkan isi matriks pada tabel tersebut maka dapat diketahui jumlah data dari masing-masing kelas yang diprediksi secara benar yaitu (True positives + True negatives) dan data yang diklasifikasikan secara salah adalah (False positive + False negatives) [9].

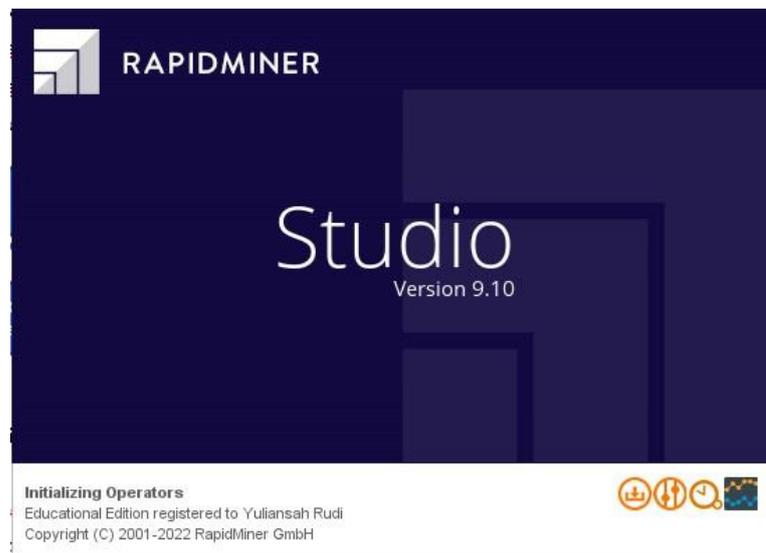
## **2.4 Alat Bantu Pemrograman**

### **2.4.1 Rapidminer**

*RapidMiner* merupakan sebuah platform perangkat lunak di bidang ilmu data yang dikembangkan oleh perusahaan dengan nama yang sama. Platform ini menyediakan lingkungan terintegrasi untuk berbagai kegiatan, termasuk persiapan data, pembelajaran mesin, pembelajaran dalam, penambahan teks, dan analisis prediktif. RapidMiner digunakan tidak hanya dalam konteks bisnis dan komersial, tetapi juga untuk penelitian, pendidikan, pelatihan, pengembangan prototipe cepat, dan pengembangan aplikasi. Selain itu, platform ini mendukung seluruh langkah dalam proses pembelajaran mesin, yang mencakup persiapan data, visualisasi hasil, validasi model, dan optimasi. *RapidMiner* merupakan salah satu perangkat lunak yang digunakan untuk pengolahan data mining. Fungsi yang dilaksanakan oleh RapidMiner dalam bidang text mining mencakup analisis teks, ekstraksi pola-pola dari dataset yang besar, serta penggabungan teknik-teknik statistika, kecerdasan buatan, dan database.

*RapidMiner* dikenal dengan istilah YALE (Yet Another Learning Environment), yang diprakarsai oleh Ralf Klinkenberg, Ingo Mierswa, dan Simon Fischer di Unit Kecerdasan Buatan Universitas Teknik Dortmund pada tahun 2001.

Pada tahun 2006, pengembangan RapidMiner didorong oleh Rapid-I, sebuah perusahaan yang didirikan oleh Ingo Mierswa dan Ralf Klinkenberg pada tahun yang sama. Pada tahun 2007, perangkat lunak ini mengalami perubahan nama dari YALE menjadi RapidMiner. Selanjutnya, pada tahun 2013, perusahaan melakukan rebranding dari Rapid-I menjadi RapidMiner. Saat ini, RapidMiner digunakan di berbagai bidang, termasuk bisnis, pendidikan, penelitian, pelatihan, serta pengembangan prototipe, yang mendukung proses pembelajaran mesin melalui visualisasi, pengembangan model, persiapan data, dan optimasi.



**Gambar 2. 3** Tampilan Aplikasi *RapidMiner*

Sumber: A. M. Khalimi, 2022

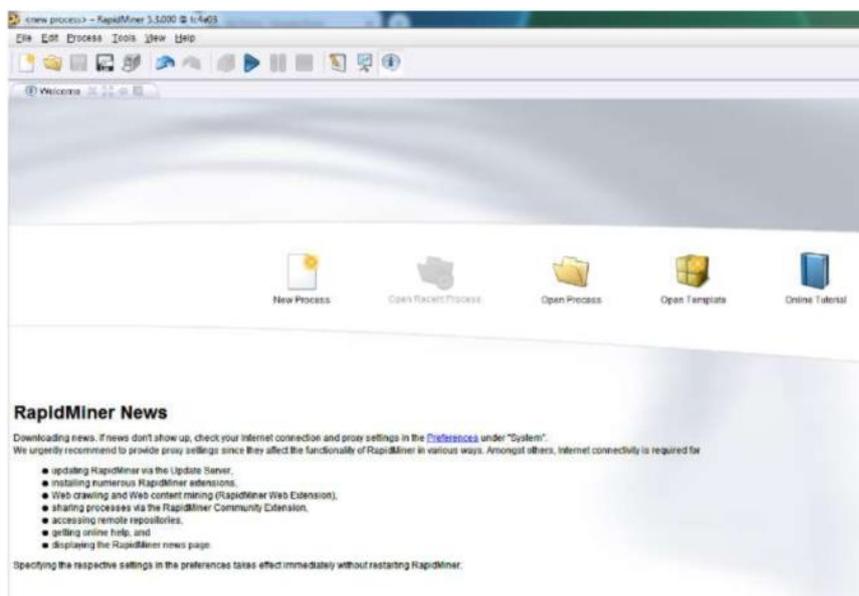
*RapidMiner* adalah perangkat lunak open source yang dirancang sebagai solusi untuk analisis data mining, text mining, dan analisis prediktif. Perangkat ini berfungsi sebagai aplikasi mandiri untuk analisis data serta sebagai mesin data mining yang dapat diintegrasikan ke dalam produk lain. RapidMiner dikembangkan menggunakan bahasa pemrograman Java, sehingga dapat beroperasi di berbagai

sistem operasi. Dengan memanfaatkan berbagai teknik deskriptif dan prediktif, RapidMiner memberikan wawasan yang berharga kepada pengguna, yang pada gilirannya dapat membantu pengambilan keputusan yang lebih optimal. RapidMiner dilengkapi dengan kurang lebih 500 operator data mining, termasuk operator untuk penginputan, pengeluaran, pra-pemrosesan data, dan visualisasi.

#### **2.4.2 Rapidminer Bagian-Bagian *RapidMiner***

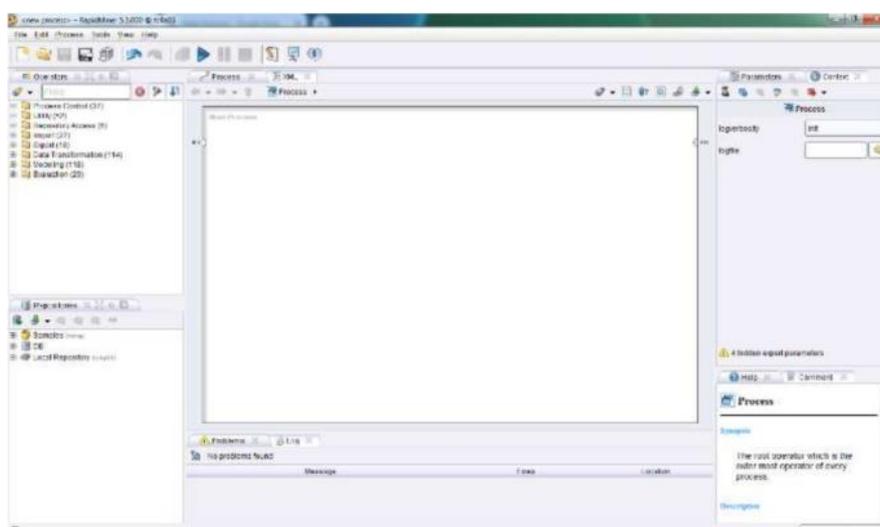
*RapidMiner* menyediakan tampilan (UI) yang ramah pengguna, sehingga memudahkan pengguna saat menggunakannya. Tampilan yang terdapat pada *RapidMiner* disebut *Perspective*. Terdapat 3 *Perspective*, yaitu *Welcome Perspective*, *Design Perspective* dan *Result Perspective*.

Ketika pengguna membuka aplikasi RapidMiner, mereka akan disambut dengan tampilan *Welcome Perspective*, seperti yang terlihat pada Gambar 2.3. Selanjutnya, pengguna akan beralih ke *Design Perspective*, yang merupakan area utama untuk bekerja dalam RapidMiner. *Design Perspective* berfungsi sebagai ruang untuk merancang dan mengelola proses analisis. Dalam tampilan ini, sebagaimana ditunjukkan pada Gambar 2. 4, terdapat beberapa view dengan fungsi yang berbeda-beda, yang semuanya mendukung proses analisis data. Selain itu, terdapat juga *Result Perspective*, yang menampilkan hasil dari analisis. Hasil yang ditampilkan dapat bervariasi, mulai dari grafik, teks, pohon, tabel, dan lain sebagainya, tergantung pada jenis data dan metode yang digunakan dalam analisis.



**Gambar 2. 4** *Welcome Perspective RapidMiner*

Sumber: V. R. Prasetyo, 2021



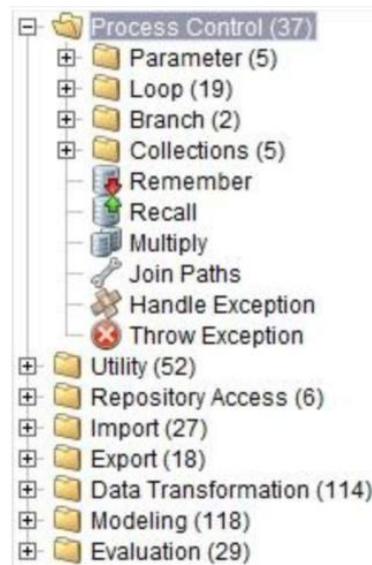
**Gambar 2. 5** *Design Perspective RapidMiner*

Sumber: V. R. Prasetyo,2021

Dalam lingkungan kerja Design Perspective, terdapat beberapa tampilan yang dapat digunakan. Tampilan yang paling utama dan esensial adalah Operator View. Di sinilah semua operator dan langkah-langkah kerja dalam RapidMiner

ditampilkan dalam bentuk grup hierarkis, memungkinkan pengguna untuk mengintegrasikannya dalam proses analisis, seperti yang terlihat pada Gambar 2.5. Berdasarkan gambar tersebut, Operator View dibagi menjadi beberapa kelompok. Pertama, terdapat kelompok Process Control, yang berisi logika looping dan percabangan untuk mengatur alur proses analisis data. Kedua, terdapat kelompok Utility yang berfungsi sebagai operator pendukung, seperti macros, login, subproses, dan lain-lain. Kelompok selanjutnya adalah Repository Access, yang digunakan untuk membaca atau menulis akses ke repository. Berikutnya adalah kelompok Import, yang berperan untuk membaca data dan objek dari berbagai sumber seperti file dan database. Kemudian ada kelompok Export, yang memiliki fungsi berlawanan dengan Import, digunakan untuk menulis data ke dalam format tertentu.

Selanjutnya, terdapat operator Data Transformation yang berguna untuk melakukan transformasi data dan metadata. Kemudian, kita memiliki operator Modelling, yang mencakup berbagai metode dan teknik data mining untuk pengolahan data. Terakhir, ada kelompok Evaluation, yang berfungsi untuk mengevaluasi kualitas output yang dihasilkan.



**Gambar 2. 6** *Operator View RapidMiner*

Sumber: V. R. Prasetyo,2021

Selain *operator view*, *repository view* juga merupakan komponen yang cukup penting dalam *Design Perspective*. Pada *view* ini, user dapat mengelola dan mengatur proses analisis data menjadi sebuah proyek, di mana pada saat yang sama *view* tersebut berperan sebagai sumber data dan hal-hal yang terkait dengan metadata. *Process view* juga berfungsi untuk menunjukkan tahaptahap tertentu dalam proses analisis dan juga berfungsi sebagai tautan ke langkah-langkah berikutnya.

## 2.5 Penelitian Terdahulu

Penelitian terdahulu memberikan dasar teoritis yang kuat untuk penelitian baru. Dengan memahami hasil dan temuan dari studi sebelumnya, peneliti dapat membangun kerangka kerja yang lebih baik. **Tabel 2.2** di bawah ini menampilkan lima penelitian sebelumnya.

**Tabel 2. 2** Penelitian Terdahulu

No	Judul	Penulis	Metode dan Data	Hasil
1	Implementasi Teorema <i>Naïve Bayes</i> Pada Prediksi Prestasi Mahasiswa	(Hasudungan dan Pranoto, 2021)	<ul style="list-style-type: none"> <li>• Penelitian ini menggunakan metode <i>Naïve Bayes</i>.</li> <li>• Data diambil dari mahasiswa aktif semester 3 angkatan 2018. Terdapat 16 parameter yang digunakan, termasuk: Umur, Jenis Kelamin, Kategori Tempat Tinggal, Jarak dari tempat tinggal ke kampus, Jenis Pendidikan SLTA, Pekerjaan dan Pendidikan Orang</li> </ul>	Model klasifikasi yang diusulkan menggunakan <i>Naïve Bayes</i> menunjukkan akurasi sebesar 77,5%. Hasil ini lebih baik dibandingkan dengan percobaan sebelumnya yang tidak menggunakan deteksi anomali, di mana akurasi hanya 69%. Penelitian ini menunjukkan bahwa faktor-faktor keluarga memiliki

			Tua, Penghasilan Orang Tua, dan Jumlah Anggota Keluarga.	<p>pengaruh terhadap prestasi akademik mahasiswa. Penelitian ini berhasil menunjukkan bahwa algoritma <i>Naïve Bayes</i> dapat digunakan untuk memprediksi prestasi akademik mahasiswa dengan akurasi yang baik. Disarankan untuk meningkatkan akurasi model dengan memilih atribut yang tepat dan menambah jumlah data.</p>
2	<p>Penerapan Data Mining dalam Proses Prediksi Perceraian Menggunakan Algoritma <i>Naïve Bayes</i> di Kabupaten Aceh Tengah</p>	(Sari et al., 2023)	<ul style="list-style-type: none"> <li>• Penelitian ini menggunakan algoritma <i>Naïve Bayes</i>.</li> <li>• Data yang digunakan adalah data perceraian diambil dari Mahkamah Syari'ah Aceh. Data mencakup faktor-faktor yang mempengaruhi</li> </ul>	<p>Hasil penelitian telah berhasil membangun sistem berbasis <i>web</i> untuk memprediksi perceraian. Pengguna dapat melakukan prediksi secara mandiri sebelum mengajukan gugatan. Model yang dibangun menggunakan</p>

			<p>perceraian, seperti usia, lama pernikahan, dan jumlah anak.</p>	<p>algoritma <i>Naïve Bayes</i> menunjukkan hasil yang baik dalam memprediksi perceraian. Penelitian menemukan bahwa faktor-faktor tertentu, seperti usia dan lama pernikahan, berpengaruh signifikan terhadap kemungkinan perceraian. Penelitian ini menunjukkan bahwa penerapan algoritma <i>Naïve Bayes</i> dalam memprediksi perceraian sangat efektif dan dapat membantu dalam pengambilan keputusan untuk mencegah perceraian di masa depan.</p>
3	<p>Mengimplementasikan Metode <i>Naïve Bayes</i></p>	<p>(Hengky P et al., 2023)</p>	<ul style="list-style-type: none"> <li>• Penelitian ini menggunakan metode <i>Naïve Bayes</i> sebagai metode</li> </ul>	<p>Hasil penelitian menunjukkan bahwa probabilitas posterior</p>

	<p>Untuk Memprediksi Tingkat Kelulusan Mahasiswa Tepat Waktu</p>		<p>klasifikasi berbasis probabilitas untuk memprediksi tingkat kelulusan mahasiswa.</p> <ul style="list-style-type: none"> <li>• Dataset yang digunakan adalah data mahasiswa Sistem Informasi Universitas Prabumulih tahun angkatan 2020, berjumlah 36 data. Data mencakup atribut seperti tahun masuk, jenis kelamin, jumlah SKS yang diambil per semester, dan Indeks Prestasi Kumulatif (IPK).</li> </ul>	<p>untuk kelas Cepat: 0,079, Tepat: 0,658, dan Terlambat: 0,211. Ini menunjukkan bahwa mahasiswa dengan atribut yang mengarah pada kelas "Tepat" memiliki kemungkinan besar untuk lulus dalam waktu tepat 4 tahun. Penelitian ini menyimpulkan bahwa metode <i>Naïve Bayes</i> dapat digunakan secara efektif untuk memprediksi kelulusan mahasiswa berdasarkan atribut yang relevan, dan hasilnya dapat membantu universitas dalam meningkatkan kualitas pendidikan dan layanan. Penelitian ini memberikan wawasan tentang bagaimana metode <i>Naïve Bayes</i> dapat</p>
--	--	--	---	---

				diterapkan untuk memprediksi tingkat kelulusan mahasiswa, serta pentingnya data yang relevan dalam proses pengambilan keputusan di institusi pendidikan.
4	Implementasi Algoritma <i>Naïve Bayes</i> Classifier untuk Memprediksi Tingkat Produktivitas Kinerja Karyawan	(Nuryansah dan Ary, 2024)	<ul style="list-style-type: none"> <li>• Penelitian ini menggunakan algoritma <i>Naïve Bayes</i> Classifier untuk memprediksi tingkat produktivitas kinerja karyawan di PT. Focon Indo Beton.</li> <li>• Data yang digunakan yaitu sampel data: 50 data karyawan dari PT. Focon Indo Beton. Atribut Data: Meliputi NPP, nama karyawan, dan skor penilaian untuk atribut seperti tanggung jawab, kerja sama, kedisiplinan, dan ketepatan waktu.</li> </ul>	Hasil penelitian menunjukkan tingkat akurasi model <i>Naïve Bayes</i> Classifier mencapai akurasi sebesar 98,80%, nilai presisi sebesar 98,33%, recall sebesar 99,09%, dan <i>Area Under Curve</i> (AUC) mencapai 0,999, menunjukkan bahwa model sangat baik dalam klasifikasi. Hasil penelitian menunjukkan bahwa algoritma <i>Naïve Bayes</i> Classifier efektif dalam

				memprediksi tingkat produktivitas kinerja karyawan, dan dapat digunakan sebagai alat evaluasi untuk meningkatkan produktivitas di perusahaan.
5	Prediksi Kehadiran Mahasiswa Universitas Teknologi Digital Menggunakan <i>Naïve Bayes</i>	(Saepudin dan Usman, 2024)	<ul style="list-style-type: none"> <li>• Penelitian ini menggunakan metode <i>Naïve Bayes</i>.</li> <li>• Data menggunakan kuesioner yang disebarkan kepada mahasiswa Universitas Teknologi Digital. Data yang dikumpulkan mencakup berbagai atribut seperti jenis mahasiswa, jenis kelamin, umur, semester, transportasi sehari-hari, jarak domisili ke kampus, dan kondisi kehadiran. Dataset terdiri dari 224 data dengan 8 atribut: Jenis Mahasiswa</li> </ul>	Hasil pengujian menggunakan pemrograman Python menunjukkan bahwa terdapat 12 prediksi benar dan 8 prediksi salah, dengan akurasi model sebesar 60%. Perhitungan manual menunjukkan akurasi sebesar 65%. Penelitian menyimpulkan bahwa atribut kondisi mata kuliah dan semester sangat mempengaruhi prediksi kehadiran mahasiswa. Penelitian ini menunjukkan

			(Reguler/Karyawan), Jenis Kelamin (Laki-laki/Perempuan), Umur (Remaja/Dewasa), Semester, Transportasi Sehari-hari, Jarak Domisili ke Kampus, Kondisi (Jumlah Mata Kuliah), dan Hasil Kehadiran (Hadir/Ragu/Tidak).	bahwa metode <i>Naïve Bayes</i> dapat digunakan untuk memprediksi kehadiran mahasiswa dengan akurasi yang cukup baik, meskipun ada ruang untuk perbaikan dengan menambah jumlah data yang digunakan.
--	--	--	--	--