BAB II
LANDASAN TEORI

2.1 Data Mining

Data mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam *database*. *Data mining* adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* besar (Ridwan, 2013).

Definisi umum dari *data mining* itu sendiri adalah proses pencarian pola-pola yang tersembunyi (*hidden patern*) berupa pengetahuan (*knowledge*) yang tidak diketahui sebelumnya dari suatu sekumpulan data yang mana data tersebut dapat berada di dalam *database*, data *werehouse*, atau media penyimpanan informasi yang lain. Hal penting yang terkait di dalam data mining adalah:

1. *Data mining* merupakan suatu proses otomatis terhadap data yang sudah ada.
2. Data yang akan diproses berupa data yang sangat besar.
3. Tujuan data mining adalah mendapatkan hubungan atau pola yang mungkin memberikan indikasi yang bermanfaat (Iswahyudi, 2015).

Data mining dilakukan dengan *tool* khusus untuk mengeksekusi operasi data yang telah didefinisikan berdasarkan model analisis. *Data mining* merupakan proses analisis terhadap data dengan penekanan menemukan informasi yang tersembunyi pada sejumlah data besar yang disimpan ketika menjalankan bisnis perusahaan. Kemajuan luar biasa yang terus berlanjut dalam bidang *data mining* didorong oleh beberapa faktor antara lain:

1. Pertumbuhan yang cepat dalam kumpulan data.
2. Penyimpanan data dalam data *warehouse*, sehingga seluruh perusahaan memiliki akses ke dalam database yang andal.
3. Adanya peningkatan akses data melalui navigasi web dan internet.
4. Tekanan kompetisi bisnis untuk meningkatkan penguasaan pasar dalam globalisasi ekonomi.

5. Perkembangan teknologi perangkat lunak untuk data mining (ketersediaan teknologi).
6. Perkembangan yang hebat dalam kemampuan komputasi dan pengembangan kapasitas media penyimpanan (Iswahyudi, 2015).

Ada beberapa teknik yang dimiliki data mining berdasarkan tugas yang bisa dilakukan, yaitu :

1. Deskripsi

   Terkadang penelitian analisis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.
2. Estimasi

   Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik daripada ke arah kategori. Model dibangun menggunakan record lengkap yang menyediakan nilai dari variabel target sebagai nilai Klasifikasi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel Klasifikasi.
3. Prediksi

   Prediksi memiliki kemiripan dengan estimasi dan klasifikasi. Hanya saja, Prediksi hasilnya menunjukkan sesuatu yang belum terjadi (mungkin terjadi di masa depan).
4. Klasifikasi

   Dalam klasifikasi variabel, tujuan bersifat kategorik. Misalnya, kita akan mengklasifikasikan pendapatan dalam tiga kelas, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.
5. *Clustering*

   Pengklusteran merupakan pengelompokan record, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. *Cluster* adalah kumpulan record yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan record-record dalam *cluster* lain.

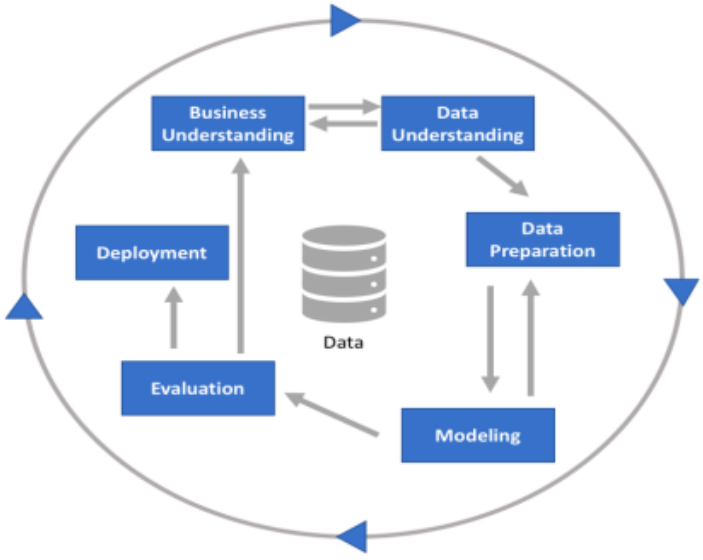
Pengklusteran berbeda dengan klasifikasi yaitu tidak adanya variabel target dalam pengklusteran.

6. Asosiasi

   Mengidentifikasi hubungan antara berbagai peristiwa yang terjadi pada satu waktu (Hermawati, 2013).

2.2 Cross Industry Standart Process for Data Mining

Cross-Industry Standard Process for Data Mining (CRISP-DM) merupakan suatu konsorsium perusahaan yang dikembangkan pada tahun 1996 oleh analis dari beberapa industri seperti Daimler Chrysler, SPSS dan NCR. CRISP-DM menyediakan standar proses *data mining* sebagai strategi pemecahan masalah secara umum dari bisnis atau unit penelitian. Dalam CRISP-DM, sebuah proyek *data mining* memiliki proses yang terbagi dalam enam fase. Keseluruhan fase berurutan yang ada tersebut bersifat adaptif. Fase berikutnya dalam urutan bergantung pada keluaran dari fase sebelumnya. Adapun Gambar 2.1 proses siklus hidup pengembangan dari CRISP-DM sebagai berikut:



Gambar 2.1 Model Fase Data mining dalam CRISP-DM (Ridwan, 2013)

Berikut enam fase proses *data mining* berdasarkan CRISP-DM:

1. Fase Pemahaman Bisnis (*Business Understanding Phase*), yaitu fase untuk memahami tujuan dan kebutuhan dalam lingkup bisnis, kemudian menerjemahkan tujuan dan batasan menjadi formula dari permasalahan *data mining*. Selanjutnya akan ditentukan strategi untuk mencapai tujuan.
2. Fase Pemahaman Data (*Data Understanding Phase*), yaitu fase pengumpulan data yang kemudian akan dilakukan analisis penyelidikan untuk mengenali data lebih lanjut dan pencarian pengetahuan awal, mengidentifikasi kualitas data serta jika diinginkan, memilih sebagian kecil grup data yang mungkin mengandung pola dari permasalahan.
3. Fase Pengolahan Data (*Data Preparation Phase*), dimana fase ini meliputi pengumpulan data yang akan digunakan untuk keseluruhan fase berikutnya. Fase ini juga mencakup pemilihan variabel yang ingin di analisis, melakukan perubahan pada beberapa variabel jika dibutuhkan serta menyiapkan data awal untuk kemudian dijadikan masukan dalam fase pemodelan.
4. Fase Pemodelan (*Modeling Phase*), dimana akan dilakukan pemilihan dan pengaplikasian berbagai teknik pemodelan dan beberapa parameternya akan disesuaikan untuk mengoptimalkan hasil. Secara khusus, beberapa teknik yang berbeda dapat digunakan pada permasalahan *data mining* yang sama. Dan untuk menjadikan data ke dalam bentuk yang sesuai dengan spesifikasi kebutuhan teknik *data mining* tertentu, pada fase ini proses dapat kembali ke fase sebelumnya (pengolahan data).
5. Fase Evaluasi (*Evaluation Phase*), dimana pada fase ini dilakukan evalusi terhadap model yang telah terbentuk untuk mendapatkan kualitas dan efektivitas sebelum disebarkan untuk digunakan. Pada fase ini pula ditetapkan apakah terdapat model yang memenuhi tujuan pada fase awal dan apakah terdapat permasalahan penting dari bisnis atau penelitian yang tidak tertangani dengan baik. Di akhir fase ini kemudian diambil keputusan berkaitan dengan penggunaan hasil dari *data mining*.
6. Fase Penyebaran (*Deployment Phase*), dimana pada fase ini model yang dihasilkan telah dapat digunakan. Contoh sederhana pada fase penyebaran

yakni pembuatan laporan, sedangkan contoh kompleks fase penyebaran yakni penerapan proses *data mining* secara paralel pada departemen lain (Istyfaiyah dan Wati, 2017)

2.3 Classification

Classification adalah proses untuk mencari model atau fungsi yang menggambarkan dan membedakan kelas-kelas atau konsep data. Fungsi dari *Classification* adalah untuk mengklasifikasikan suatu target *class* ke dalam kategori yang dipilih (Sundari dkk,2019).

Ada banyak metode untuk membangun model klasifikasi seperti *naïve-bayesian classification, support vector machine* (SVM) dan *k-nearest neighbor classication.*

2.4 Algoritma Naïve Bayes (NB)

Klasifikasi *Bayesian* adalah klasifikasi statistik yang bisa diklasifikasi probabilitas sebuah kelas. Klasifikasi Bayesian ini dihitung berdasarkan *Teorema Bayes. Teorema Bayes* adalah perhitungan statistik dengan menghitung probabilitas kemiripan kasus lama yang ada dibasis kasus dengan kasus baru. *Teorema Bayes* tingkat akurasi yang tinggi dan kecepatan yang baik ketika diterapkan pada database yang besar (Han dan Kamber, 2012).

Bayes merupakan teknik Klasifikasi berbasis probabilistik sederhana yang berdasarkan pada *Teorema Bayes* atau aturan *bayes* dengan asumsi independensi yang kuat (naif). Dengan kata lain, dalam NB, model yang digunakan adalah "model fitur independen" (Afiyah dan Dengen, 2017).

Dalam *Bayes* (terutama *Naïve Bayes*), maksud independen yang kuat pada fitur adalah bahwa sebuah fitur pada sebuah data tidak berkaitan dengan ada atau tidaknya fitur lain dalam data yang sama. Contohnya, pada kasus hewan dengan fitur penutup kulit, melahirkan, berat dan menyusui. Di sini ada ketergantungan pada fitur menyusui karena hewan yang menyusui biasanya melahirkan, atau hewan yang bertelur biasanya tidak menyusui. Dalam *bayes* hal tersebut tidak

dipandang sehingga masing-masing fitur seolah tidak memiliki hubungan apapun (Prasetyo, 2012).

NB Classifier termasuk ke dalam pembelajaran *supervised*, *Naïve Bayes* mengestimasi peluang kelas bersyarat dengan mengasumsikan bahwa atribut adalah independen secara bersyarat yang diberikan dengan label y, Asumsi independen bersyarat dapat dinyatakan dalam bentuk berikut (Andriani, 2013).

Klasifikasi dengan NB bekerja berdasarkan teori probabilitas yang memandang semua data sebagai bukti dalam probabilitas. Hal ini memberikan karakteristik NB sebagai berikut.

1. Metode *Naïve Bayes* teguh (*robust*) terhadap data-data yang terisolasi yang biasanya merupakan data dengan karakteristik berbeda (*outliner*). NB juga bisa menangani nilai atribut yang salah dengan mengabaikan data latih selama proses pembangunan model dan Klasifikasi.
2. Tangguh menghadapi atribut yang tidak relevan.
3. Atribut yang mempunyai korelasi bias mendegredasi kinerja klasifikasi *Naïve Bayes* karena asumsi independensi atribut tersebut sudah tidak ada (Indrawan, 2017).

Adapun Algoritma NB memiliki kelebihan yaitu Relatif mudah untuk diimplementasi karena tidak menggunakan optimasi numerik, perhitungan matriks dan lainnya, Efisien dalam pelatihan dan penggunaannya, Bisa menggunakan data *binary* atau *polinom*, Karena diamsumsikan independen maka memungkinkan metode ini diimplementasikan dengan berbagai macam dataset, Akurasi yang relatif tinggi. Algoritma NB juga memiliki kekurangan yaitu Perkiraan kemungkinan kelas yang tidak akurat dan Batasan atau *threshold* harus ditentukan secara manual dan bukan secara analisis.

Menurut (Indrawan, 2017), persamaan dari Teorema Bayes pada umumnya adalah sebagai berikut:

$$P(X) = \frac{P(H)P(H)}{P(X)} \quad (2.1)$$

- $P(H|X)$ posterior probability dari *class* (*target*) tiap *predictor* (*attribute*).
- $P(H)$ prior probability dari *class*.
- $P(X|H)$ likelihood : probability dari *predictor* tiap *class*.
- $P(X)$ prior probability dari *predictor*.

Pada proses klasifikasi diperlukan sejumlah petunjuk yaitu variabel-variabel untuk menentukan kelompok mana yang tepat bagi obyek yang dianalisis tersebut sehingga teorema Bayes disesuaikan sebagai berikut:

$$P(Y \mid X_1, X_2, \ldots, X_p) = \frac{P(Y)P(X_1, X_1, \ldots., X_p \mid Y)}{P(X_1, X_1, \ldots., X_p \mid Y)} \quad (2.2)$$

Dengan

- $P(Y \mid X_1, X_2, \ldots, X_p)$ adalah peluang masuknya obyek dengan karakteristik variabel tertentu dalam kelompok Y (posterior)
- $P(X_1, X_1, \ldots, X_p \mid Y)$ adalah peluang kemunculan variabel-variabel pada obyek yang masuk kelompok Y (likelihood)
- $P(Y)$ adalah peluang munculnya kelompok Y sebelum masuknya obyek (prior)
- $P(X_1, X_1, \ldots, X_p \mid Y)$ adalah peluang kemunculan variabel-variabel pada obyek umum (evidence).

Pada persamaan (2.2) dapat juga dituliskan secara sederhana sebagai berikut:

$$\text{Posterior} = \frac{\text{Prior} * \text{Likelihood}}{\text{Evidence}} \quad (2.3)$$

Nilai dari *posterior* tersebut nantinya akan dibandingkan dengan nilai-nilai *posterior* kelompok lainnya untuk menentukan kelompok suatu obyek akan

diklasifikasikan. Mengklasifikasikan suatu obyek dapat ditentukan dengan memilih kelompok yang memiliki *posterior* terbesar, nilai *evidence* selalu tetap untuk setiap kelompok pada satu sampel yaitu bernilai 1 dan merupakan pembagi pada setiap kelompok sehingga dalam perhitungan *posterior* hanya cukup mengalikan nilai *prior* dengan *likelihood*. Nilai prior yang merupakan peluang munculnya kelompok *Y* sebelum masuknya obyek dapat dihitung menggunakan persamaan 2.4 sebagai berikut:

$$P(Y_g) = \frac{n_g}{N} \quad (2.4)$$

Dengan

$P(Y_g)$ = Peluang munculnya Y ke-g dimana g= 1,2,3,…,q

N_g = Banyaknya pengamatan pada kelompok ke-g

Penjabaran lebih lanjut rumus Bayes tersebut dilakukan dengan menjabarkan $P(Y \mid X_1, X_2, \ldots, X_p)$ menggunakan aturan perkalian sebagai berikut:

$$\begin{aligned} P(Y \mid X_1, X_2, \ldots, X_p) &= P(Y)P(X_1 \mid Y)P(X_2, X_3, \ldots, X_p \mid Y, X_1) \\ &= P(Y)P(X_1 \mid Y)P(X_2, X_3, \ldots, X_p \mid Y, X_1) \\ &\quad P(X_3, X_4, \ldots, X_p \mid Y, X_1 X_2) \\ &= P(Y)P(X_1 \mid Y)P(X_2 \mid Y, X_1)\, P(X_3 \mid Y, X_1 X_2)\ldots \\ &\quad P(X_4, X_5, \ldots, X_p \mid Y, X_1, X_2, X_3) \\ &= P(Y)P(X_1 \mid Y)P(X_2 \mid Y, X_1)P(X_3 \mid Y, X_1, X_2)\ldots \\ &\quad P(X_p \mid Y, X_1, X_2, X_3, \ldots, X_{p-1}) \end{aligned} \quad (2.5)$$

Dapat dilihat bahwa hasil penjabaran tersebut menyebabkan semakin banyak dan semakin kompleksnya faktor-faktor syarat yang mempengaruhi nilai peluang yang hampir mustahil untuk dianalisa satu persatu. Kompleksnya faktor-faktor syarat yang mempengaruhi nilai peluang menyebabkan perhitungan tersebut menjadi sulit untuk dilakukan, maka digunakan asumsi independensi yang sangat tinggi (*naive*), bahwa masing-masing petunjuk ($X_1, X_2, \ldots, X_p$) saling bebas

(*independent*) satu sama lain sehingga berlaku suatu persamaan (2.6) sebagai berikut:

$$P(X_a \mid X_b) = \frac{P(X_a \cap X_b)}{P(X_b)} = \frac{P(X_a)P(X_b)}{P(X_b)} = P(X_a) \quad (2.6)$$

untuk $a \neq b$ sehingga $P(X_a \mid Y, X_b) = P(X_a \mid Y)$

Pada Persamaan (2.6) dapat disimpulkan bahwa asumsi independensi *naïve* tersebut membuat syarat peluang menjadi sederhana sehingga perhitungan menjadi mungkin dilakukan. Selanjutnya penjabaran $P(Y \mid X_1, X_2, ..., X_p)$ dapat disederhanakan menjadi:

$$P(Y \mid X_1, X_2, ..., X_p) = P(Y)\, P(X_1 \mid Y)\, P(X_2 \mid Y)\, P(X_3 \mid Y) ... P(X_p \mid Y)$$

$$= P(Y) \prod_{K=1}^{p} P(Y_K \mid Y) \quad (2.7)$$

Persamaan (2.7) merupakan model dari teorema *naive* Bayes yang selanjutnya akan digunakan dalam proses klasifikasi.

2.5 Algoritma *K-Nearest Neighbors* (KNN)

K-Nearest Neighbors (KNN) adalah metode klasifikasi untuk menghitung kedekatan antar atribut baru dengan atribut lama berdasarkan bobot setiap atribut tersebut (Kusrini & Lutfi, 2009). Metode ini membutuhkan waktu untuk menentukan K (jumlah tetangga terdekat).

Adapun rumus untuk melakukan penghitungan kedekatan antara dua kasus dengan menghitung jarak euclidean seperti pada persamaan (2.8) berikut:

$$d(x_{ik}, x^*_{jk}) = \sqrt{\sum_{K=1}^{p} (X_{ik} - X^*_{jk})^2} \quad (2.8)$$

dimana

$d(x_{ik}, x^*_{jk})$ = *jarak Euclideance data training ke-i dengan data testing ke-j*

X_{ik} = nilai varabel bebas ke –k dari training ke-I , i= 1,2,…..,n

x^*_{jk} = nilai variabel bebas ke-k dari data testing ke-j , J= 1,2,….n

P = Banyaknya variabel bebas

Kedekatan biasanya berada pada nilai antara 0 s.d 1. Nilai 0 artinya kedua kasus mutlak tidak mirip, sebaliknya untuk 1 kasus mirip dengan mutlak (Afiyah dan Dengen, 2017). Menurut (Suyanto, 2017) algoritma KNN adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Ketepatan algoritma KNN ini sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi.

Algoritma KNN adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Data pembelajaran diproyeksikan k ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data pembelajaran. Sebuah titik pada ruangan ini ditandai dengan kelas c, jika kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat titik tersebut KNN merupakan metode yang bersifat *supervised*, dimana hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas kategori pada KNN. Pada fase training, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data *training sample*. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk testing data (klasifikasinya belum diketahui). Jarak dari vektor yang baru ini terhadap seluruh vaktor training sample dihitung, dan sejumlah k buah yang paling dekat diambil. Titik yang baru klasifikasinya dipredikasikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut. Ketepatan algoritma KNN oleh ada atau tidak adanya fitur-fitur yang tidak relevan, atau jika bobot fitur tersebut setara dengan relevansinya terhadap klasifikasi (Afiyah dan Dengen, 2017).

Algoritma KNN memiliki kelebihan yaitu dapat menghasilkan data yang kuat atau jelas dan efektif digunakan pada data yang besar. Dari beberapa kelebihan tersebut, KNN juga memiliki kekurangan yaitu membutuhkan nilai K sebagai parameter jarak dari data percobaan tidak dapat jelas dengan tipe jarak yang digunakan dan dengan atribut yang digunakan untuk memperoleh hasil yang terbaik, maka harus menggunakan semua atribut atau hanya satu atribut yang telah pasti, dan perhitungan harga sangat tinggi karena percobaan ini membutuhkan perhitungan jarak dari beberapa *query* untuk semua data percobaan (Afiyah dan Dengen, 2017). Menurut Prasetyo (2012), pada metode K-NN, nilai *K* menyatakan jumlah tetangga terdekat yang dilibatkan dalam proses penentuan Klasifikasi label kelas pada data *testing*. Dari *K* tetangga terdekat yang terpilih kemudian dilakukan voting kelas dari nilai *K* tersebut. Kelas dengan jumlah suara tetangga terbanyaklah yang diberikan sebagai label kelas hasil Klasifikasi pada data *training* tersebut. Berikut adalah algoritma K-NN.

1. Menentukan jumlah K tetangga terdekat.
2. Melakukan perhitungan jarak antar data testing dan semua data training menggunakan rumus jarak Euclid.
3. Mengurutkan jarak (ranking).
4. Menggunakan voting kelas sebagai Klasifikasi dari data testing tersebut.

Ada beberapa hal yang mempengaruhi kinerja K-NN, diantaranya adalah pemilihan nilai K . Jika K terlalu kecil maka berakibat hasil Klasifikasi yang didapat bisa sensitif terhadap keberadaan noise. Jika K terlalu besar maka tetangga terdekat yang terpilih terlalu banyak dari kelas lain yang sebenarnya tidak relevan karena jarak yang terlalu jauh. Pemilihan nilai K genap atau ganjil juga menjadi perhatian. Untuk K genap dengan jumlah klasifikasi genap akan ada kemungkinan voting dari kedua klasifikasi mendapat suara yang sama. Akan tetapi untuk K ganjil dengan jumlah klasifikasi genap akan memudahkan karena dijamin kedua kelas tidak akan mendapat suara yang sama (Prasetyo, 2012).

2.6 RapidMiner

RapidMiner sebelumnya dikenal sebagai YALE (Yet Another Learning Environment), dikembangkan mulai tahun 2001 oleh Ralf Klinkenberg, Ingo mierswa, dan Simon Fischer di Unit Artificial Intelligence dari Technical University of Dortmund. Mulai tahun 2006, perkembangannya adalah didorong oleh cepat, sebuah perusahaan yang didirikan oleh Ingo mierswa dan Ralf Klinkenberg pada tahun yang sama. pada tahun 2007, nama software diubah dari YALE ke RapidMiner dan perusahaan cepat-I GmbH didirikan. Pada akhir Mei, bebas open-source Suite data mining YALE berganti nama menjadi RapidMiner. Sekarang, Rilis ini memberikan semua fungsi yang diketahui dari YALE dan menambahkan sejumlah besar fungsi-fungsi baru bersama dengan antarmuka pengguna sepenuhnya direvisi. berharap bahwa perbaikan dari YALE ke RapidMiner lebih berguna untuk analisis pekerjaan sehari-hari.

RapidMiner dan plugin yang sekarang menyediakan lebih dari 400 belajar dan preprocessing operator dan kombinasi yang tak terhitung jumlahnya dari. Oleh karena itu, RapidMiner adalah pelengkap pengetahuan penemuan Suite yang dapat digunakan untuk semua tugas data mining. Di antara fitur baru adalah ruang kerja untuk proyek yang berbeda dengan meningkatkan visualisasi dari kriteria kinerja seperti kurva ROC rata-rata atau plot 3D dari matriks .

Rapid Miner adalah aplikasi data mining yang tidak perlu dipertanyakan lagi dan berbasis sistem open-source dunia yang terkemuka dan ternama. Tersedia sebagai aplikasi yang berdiri sendiri untuk analisis data dan sebagai mesin data mining untuk integrasi ke dalam produk sendiri. Ribuan aplikasi RapidMiner di lebih dari 40 negara memberikan pengguna mereka keunggulan yang kompetitif. Solusi yang di usung antara lain :Integrasi data, Analitis ETL, Data Analisis, dan Pelaporan dalam satu suite tunggal. Powerfull tapi memiliki antarmuka pengguna grafis yang intuitif untuk desain analisis proses.Repositori untuk proses, data dan penanganan meta data Hanya solusi dengan transformasi meta data: lupakan trial and error dan memeriksa hasil yang telah di inspeksi selama desain.

2.6.1 *Split Data*

Split Data adalah sebuah teknik yang digunakan dalam pengolahan data dan pemodelan statistik untuk membagi dataset menjadi dua bagian yang terpisah: dataset pelatihan (training set) dan dataset pengujian (test set). (Vercellis, 2009). Penggunaan dataset pelatihan dan dataset pengujian dalam teknik Split Data sangat penting untuk menghindari overfitting, yaitu kondisi saat model sangat baik dalam mengenali pola pada dataset pelatihan tetapi tidak dapat melakukan prediksi dengan baik pada data baru atau data yang belum pernah dilihat sebelumnya. Salah satu pendekatan yang umum digunakan dalam teknik Split Data adalah dengan membagi dataset secara acak dalam proporsi tertentu, misalnya 80% dataset digunakan untuk pelatihan dan 20% digunakan untuk pengujian.

2.6.2 Akurasi

Akurasi adalah ukuran dari seberapa baik model mengkorelasikan antara hasil dengan atribut dalam data yang telah disediakan. Terdapat berbagai model akurasi, tetapi semua model akurasi tergantung pada data yang digunakan. Pada tahap ini dilakukan evaluasi kinerja dari algoritma Naïve Bayes dan K-*Nearest Neighbor* dengan menggunakan perhitungan *percision*, recall dan error rate untuk mendapatkan informasi model yang akurat. Beberapa persyaratan standar yang telah ditetapkan untuk matriks klasifikasi dua kelas.

Berikut adalah rumus dari akurasi, *precision* dan *recall* . yang dapat dilihat pada persamaan (2.9), (2.10), (2.11) sebagai berikut.

$$\text{Akurasi} = \frac{TP+FN}{P+N} \tag{2.9}$$

$$\textit{Precision} = \frac{TP}{TP+FP} \tag{2.10}$$

$$\textit{Recall} = \frac{TP}{TP+FN} \tag{2.11}$$

dimana,

1. P (Positive) adalah hasil yang benar.
2. N (Negative) adalah hasil yang salah.
3. TP merupakan hasil dari prediksi sistem yang positif dan sesuai dengan target yang positif.
4. TN, merupakan hasil dari predisi sistem yang negatif dan sesuai dengan target yang negative.
5. FP, merupakan hasil dari prediksi sistem yang positif, namun hasil targetnya negatif.
6. FN, merupakan hasil dari prediksi sistem yang negatif, namun hasil targetnya positif.

1.3 Penelitian Terdahulu

Penelitian terdahulu yang berkaitan dengan penelitian ini dapat dilihat pada tabel 2.1 berikut :

Tabel 2.1 Penelitian Terdahulu

No.	Penulis dan Judul	Hasil
1.	Nur Khotimah, Deden Istiawan (2018), Perbandingan Algoritma C4.5, *Naïve Bayes* dan *K-Nearest Neighbor* untuk Klasifikasi lahan kritis di Kabupaten Pemalang	Kesimpulan yang diperoleh pada penelitian ini adalah algoritma klasifikasi yang memiliki tingkat akurasi paling tinggi dalam diklasifikasi kekritisan lahan kritis daerah kawasan hutan lindung dan hutan konservasi Kabupaten Pemalang dibandingkan algoritma klasifikasi *Naïve Bayes*dan *k-Nearest Neighbour* yaitu mencapai 77.75% disusul *Naïve Bayes* 77,49% *dan* terakhir KNN memiliki akurasi sebesar *73,91%.*
2.	Anisa Nur Afiyah (2018), Analisis perbandingan kinerja Algoritma C4.5, Naïve Bayes Dan K-Nearest Neighbor dalam penentuan rehabilitas narkoba	Kesimpulan hasil penelitian ini dalam melakukan perbandingan kinerja algoritma C4.5, *Naïve Bayes* dan KNN untuk menentukan algoritma terbaik dalam rehabilitas narkoba adalah algoritma *naïve bayes* karena memiliki akurasi yang cukup tinggi 80.55%, error rate yang rendah 19.45% dan didukung dengan kualitas data training sebesar 70.10%.

No.	Penulis dan Judul	Hasil
3.	Derick Iskandar, Yoyon K Suprapto (2015), Perbandingan akurasi klasifikasi tingkat kemiskinan antara Algoritma C4.5 dan *Naïve Bayes*	Berdasarkan hasil komparasi antara algoritma C4.5 dan Naïve *Bayes* untuk mengklasifikasikan tingkat kemiskinan dengan 14 atribut dan jumlah data yang telah di *cleaning* sebesar 13.928 data set dapat disimpulkan bahwa algoritma C4.5 memilliki tingkat akurasi yang lebih baik 3% dibandingkan dengan metode *Naïve Bayes* yang bernilai 63%.
4.	Femi Dwi Astuti, Mohammad Guntara (2018), Analisis Performa Algoritma K-NN Dan C4.5 Pada Klasifikasi Data Penduduk Miskin	Berdasarkan hasil kesimpulan bahwa algoritma K-NN dengan parameter *setting* k=1 memiliki performa yang lebih baik dibandingkan dengan nilai k=10, 100, 1000 maupun algoritma C4.5 dengan nilai *accuracy* 94,71%, *precision* sebesar 84,96% dan *recall* sebesar 83,6%.
5.	Wahyu Indrawan (2017), Komparasi algoritma *naive* bayes *classifier* dan *tree* c4.5 untuk evaluasi kinerja akademik mahasiswa Program studi teknik informatika universitas mulawarman	Hasil uji kinerja algoritma *naïve Bayes classifier* memiliki akurasi terbaik pada rasio data *training* sebesar. 80% dengan nilai akurasi sebesar 76.79% dengan kualitas data training terhadap algoritma sebesar 81.92% sedangkan algoritma *tree* C4.5 memiliki akurasi terbaik pada rasio data *training* sebesar. 90% dengan nilai akurasi sebesar 78.57% dengan kualitas data training terhadap algoritma sebesar 76.72%.
6.	Sri Wahyuningsih, Dyah Retno Utari (2018), Perbandingan Metode, KNN, Naïve Bayes dan Decision Tree untuk Klasifikasi Kelayakan Pemberian Kredit	Metode Decision Tree memiliki tingkat akurasi yang baik yaitu sebesar 92,21% untuk Klasifikasi kelayakan pemberian kredit kepada nasabah, metode K-Nearest Neighbor sebesar 81,82% dan metode Naïve Bayes memiliki akurasi sebesar 81,83%.
7	Yeyen Dwi Atma, Arif Setyanto (2018), Perbandingan Algoritma C4.5 dan KNN Dalam Identifikasi Mahasiswa Berpotensi Drop OUT	Hasil dari pengujian algoritma klasifikasi kasus Klasifikasi mahasiswa drop out untuk algoritma C4.5 tanpa penambahan fitur seleksi Forward selection didapatkan akurasi sebesar 95.96%, kemudian setelah ditambahkan fitur seleksi Forward selection meningkat menjadi 96.66%. Sedangkan pada algoritma K-Nearest Neighbour tanpa penambahan fitur seleksi Forward selection diperoleh nilai akurasi sebesar 95.07%

No.	Penulis dan Judul	Hasil
		setelah ditambahkan fitur seleksi Forward selection meningkat menjadi 98.34%.
8	Devi Yunita (2017), Perbandingan Algoritma *K-Nearest Neighbor* dan *Decision Tree* Untuk Penetuan Risiko Kredit Kepemilikan Mobil	Pada hasil penelitian menunjukan penggunaan Algoritma *K-Nearest Neighbor* lebih akurat dalam penentuan kelayakan konsumen dengan nilai akurasi 98,18%. Nilai keakuratan ini bertujuan untuk menghindari terjadinya kredit macet.