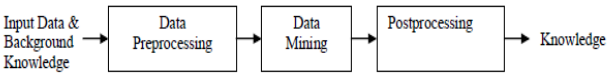
BAB II

LANDASAN TEORI

2.1. Data Mining



Gambar 2.1. Gambar Data *Mining*

Sumber: [15]

Dalam proses pengolahan data atau disebut *Knowledge Discovery* in *database*s (KDD), sangat berguna agar data tidak terjadi *Garbage In Garbage Out*, dan pada proses data *Mining* ini yang pertama yaitu data *preparation* atau bisa disebut juga dengan data *preprocessing* adalah suatu proses/langkah yang dilakukan untuk membuat data mentah menjadi data yang berkualitas atau bisa disebut juga *input* yang baik untuk data *Mining tools* dan inilah *filter* pertama. dan proses terakhir yaitu disebut *Postprocessing* atau bisa kita anggap, *filter* terakhir, dimana data yang sudah di *mining* dicek kembali jangan sampai ada data yang tidak penting maupun *double entry*. *Knowledge* adalah data *valid* yang sudah di olah akan di visualisasikan kedalam bentuk statistika [15].

Setelah proses data *mining* terdapat data *analyzing* yang dibagi menjadi 2 teknik, yang pertama yaitu *Reporting Techniques* atau bisa kita sebut cara Konvensional (*human analyst*), contohnya yaitu laporan keuangan perusahaan, yang kedua yaitu data *mining techniques* atau bisa disebut cara modern (Mengunakan *Tools*/*Software*) yang biasanya dengan visualisasi statistik (lebih akurat dan efektif), dan terakhir dalam data *mining* terdapat sebutan data *Engineering* dimana yang dimaksud itu adalah data apa yang kita miliki, dan apa hasil data olahan yang kita harapkan [15].

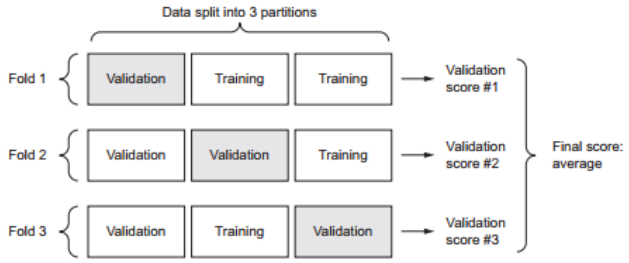
2.2. Klasifikasi

Tujuan dari klasifikasi adalah memprediksi dan mendapatkan akurasi, prediksi didapat dari pembelajaran *dataset* yang awalnya dilatih terlebih dahulu. Klasifikasi termasuk dalam *predictive model* dengan menganalisis dengan varibel terikat dan variabel bebas.

Tujuan lainnya bisa disebut juga untuk memprediksi suatu nilai variabel yang tidak diketahui sebelumnya ataupun menemukan pola baru yang dapat dideskripsikan untuk mempertegas hipotesa. Akurasi dapat ditingkatkan dari data yang sudah diolah dengan baik atau sudah ditingkatkan akurasinya dengan teknik tertentu, seperti menghapus *outlier*, menambahkan *hidden layer*, iterasi atau *epochs* percobaan, komposisi data *train* dan data *test*, dan juga bisa karena data yang digunakan sedikit yang

mengakibatkan model terlalu bagus, jumlah kolom, dan baris juga mempengaruhi akurasi model, umumnya semakin banyak kolom, semakin baik akurasinya[16]–[18].

2.3. K-fold



Gambar 2.2. Gambar K-fold validation

Sumber: [16]–[18]

K-fold validation bagi data 1 set *Training* dan *validation*, yang di *fold* atau experimen validasi atau *testing* di *bin* yang berbeda sebanyak k, contoh gambar diatas dengan k=3, contoh 30 data maka total nilai pada 1 *bin* adalah 30/3=10, dan garis diagonal *validation* yang di bold pada gambar adalah data *testing* atau data validasi, *k-fold* umumnya baik untuk data yang kecil, *sample* lebih bagus supaya data *train* dibagi lagi untuk *traning* dan *validation*, dipelajari berulang-ulang di *k-fold Training* agar makin terfilter dengan baik agar akurasi dapat meningkat, semakin banyak *fold* umumnya semakin bagus, kalau terlalu banyak *fold* pun tidak bagus, dimana ada *output* yang tidak sesuai kebutuhan. Umumnya *k-fold* dapat

mengurangi memori, waktu, dan biaya terutama untuk data yang sedikit, data *train* dan data *test* agar tidak salah pilih komposisinya tidak menambahkan akurasi, tetapi hanya rata-rata *validator* akurasi sebagai tampilan *output* akhir akurasi, validasi itu data *testing*, membuat suatu *valid* pernah di *test* [16]–[18].

2.4. Variabel

Variabel independen/Variabel bebas adalah jenis variabel saling berkaitan dengan variabel yang lain atau saling mempengaruhi, Variabel independen adalah variabel sebab dari munculnya/adanya variabel terikat atau dependen, dengan kriteria dapat saling mempengaruhi diantara variabel lain, dapat memilih/dibuat-buat/dimanipulasikan vairablenya oleh peneliti.

Variabel dependen/Variabel terikat, muncul karena adanya variabel bebas atau akibat dari variabel independen, dan tidak dapat dibuat-buat oleh peneliti [16]–[18].

2.5. *Algoritma Naïve Bayes*

Naïve Bayes merupakan metode *algoritma* yang tidak mempunyai aturan, *Naïve Bayes* didapatkan dari salah satu cabang ilmu matematika yang ada hubunganya dengan teori peluang atau probabilitas yang bertujuan

mencari peluang yang terbesar dari semua kemungkinan yang ada pada metode klasifikasi, dengan cara menganalisis frekuensi atau berapa sering klasifikasi data *Training*. *Naïve Bayes* merupakan salah satu *algoritma* klasifikasi populer dan masuk dalam kategori sepuluh *algoritma* yang terbaik pada metode data *mining*, *algoritma* ini juga biasa dipanggil *Simple Bayes* [16]–[18]**.**

Klasifikasi *algoritma Naïve Bayes* berdasarkan pada teorema bayes, ditemukan oleh ahli matematika, dan menteri Inggris, yaitu Thomas Bayes (1702-1761) [6]. Yaitu:

$$P(x|y) = \frac{P(y|x).P(x)}{P(y)}$$

Keterangan:

Y : data dengan kelas yang belum diketahui

X : Hipotesis data y merupakan suatu kelas spesifik

P(x|y) : Probabilitas hipotesis x berdasarkan kondisi y (*posteriori probability*)

P(x) : Probabilitas hipotesis x (*prior probability*)

P(y|x) : Probabilitas y berdasarkan kondisi pada hipotesis x

p(y) : Probabilitas dari y

Rumus 2.1. Rumus *Naïve Bayes*

2.6. Confusion Matrix

Confusion Matrix untuk memunculkan seberapa baik klasifikasi dari data *testing*. *Confusion Matrix* merupakan matriks berdimensi dua, yaitu satu dimensi di indeks pada kelas sebenarnya dari suatu objek atau *actual* dan 1 dimensi lain berada di indeks oleh kelas yang telah ditentukan *Classifier* atau prediksi. *Confusion Matrix* contohnya bisa dilihat di Tabel 2.1.

Tabel 2.1. Struktur *Confusion Matrix* Secara Umum

Aktual \ Prediksi	**Positif**	**Negatif**
Positif	TP	FN
Negatif	FP	TN

Tabel 2.1. menjelaskan tentang TP (*True Positive*), FN (*False Negative*), FP (*False Positive*), dan TN (*True Negative*) merupakan nilai dari *Confusion Matrix*, yang bertujuan untuk perhitungan performa model. Nilai *accuracy* bertujuan untuk mengukur seberapa tepat prediksi model, untuk pengukuran performa pada model, yaitu menggunakan F1 *score* , *Precision*, dan *recall*.

Nilai *Precision* untuk menghitung perbandingan prediksi positif yang sesuai (*True Positive*) dengan total prediksi positif. Nilai *recall* menghitung perbandingan prediksi positif yang sesuai (*True Positive*) dengan semua nilai prediksi pada kelas aktual. [19]

Accuracy = (TP + TN)/(TP + TN + FP + FN)

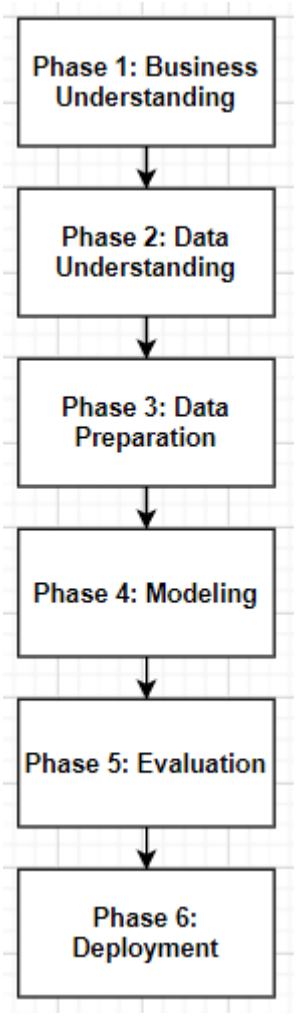
Recall=(TP)/(TP+FN)

Precision=(TP)/(TP+ FP)

Accuracy = (2 x *Precision* x *Recall*)/(*Precision* + *Recall*)

Rumus 2.2. Rumus Performa Model *Confusion Matrix*

2.7. CRISP-DM



Gambar 2.3. Grafik *CRISP-DM*

Sumber:[20]

Berdasarkan pada model referensi yang telah ditampilkan pada gambar 2.4. berikut merupakan penjelasannya:

a. *Business Understanding*, adalah pemahaman kegiatan data *mining* yang akan dilaksanakan, hubungan dengan proses, dan tujuan bisnis dengan data *mining*.
b. *Data Understanding*, merupakan proses pengumpulan data, mempelajari data, menganalisis masalah yang berhubungan dengan data.
c. *Data Preparation*, mempersiapkan struktur basis data untuk mempermudah proses *mining*.
d. *Modeling Phase*, fase memilih model, *tools*, *algoritma* data *mining*.
e. *Evaluation Phase*, fase *analysis* output data *mining* dari proses fase sebelumnya.
f. *Deployment Phase*, atau bisa disebut fase penyebaran adalah biasanya digunakan pada pembuatan *system*, yang menyebarkan model ke karyawan atau bisnis ke departemen lain.

2.8. *Algoritma* Regresi Logistik Biner

Analisis regresi logistik adalah analisis yang bertujuan mencari pengaruh atau relasi, dimana variabel responya adalah kategori yang bersifat satu atau lebih variabel bebas, *algoritma* ini bersifat logaritmik, yang baik untuk 2 pilihan variabel dependen.

fungsi umum regresi logistik adalah sebagai berikut:

$$g(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p \quad (1)$$

sehingga persamaan untuk menentukan peluangnya dapat diperoleh:

$$\pi(x) = \frac{e^{g(x)}}{1+e^{g(x)}} \quad (2)$$

dengan $0 \leq \pi(x) \leq 1$ adalah peluang kejadian berhasil dan βj adalah parameter dengan j = 1,2, . . . , p. Fungsi $\pi(x)$ adalah fungsi non-linier, maka dari itu transformasi ke dalam bentuk logit perlu dilaksanakan. perubahan kedalam fungsi linier untuk mempergampang interpretasi relasi antara prediktor dan respon. Transformasi logit dari (1) dan (2) munculah formula yang lebih simple, yakni: [21]

$$g(x) = \ln \frac{\pi(x)}{(1-\pi(x))} = (\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p) \text{ sehingga}$$

$$\pi(x) = \frac{exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)}{1 + (\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)}$$

Rumus 2.3. Rumus Logistic Regression

Di mana y atau $\pi(x)$ adalah output yang diprediksi, β_0 adalah bias atau suku intersep/*intercept term* dan β_1 adalah koefisien untuk tunggal nilai masukan (x). Setiap kolom dalam data masukan memiliki terkait koefisien β (nilai riil konstan) itu harus dipelajari dari data pelatihan [22].

Coefficient, atau bisa disebut bobot, Nilai koefisien ini memberikan wawasan tentang atribut mana yang paling penting untuk model, beberapa

di antaranya lebih penting daripada yang lain, artinya bila pada topik penelitian ini bobot semakin besar maka makin tepat waktu kalu mines makin tidak tepat waktu[16], [23].

Standardized coefficient, atau Bahasa Indonesianya adalah Koefisien standar (perkiraan koefisien mentah(*raw coefficient estimate*) dibagi dengan kesalahan standar perkiraan(*standard error of estimate*)) merupakan ukuran mentah dari relatif pentingnya terkait dengan variabel itu [23].

Standard error, tingkat kebaikan koefisien estimasi dapat ditunjukkan menggunakan *standard error* [24].

Z-value, atau *z-score*, adalah jumlah *standard deviation* dari *mean* titik data [25].

Standar deviasi sendiri berdefinisi yaitu volatilitas dari rata-rata data, semakin besar standar deviasi, maka outlier semakin banyak, sehingga prediksi data makin tidak akurat [26].

P-value bisa dibilang error pada model, dimana biasanya digunakan untuk uji hipotesis, dimana batas toleransi *error* atau *alpha* (α) sebesar 5% atau tingkat keyakinan diatas 95% dikatakan model yang baik [27].

2.9. Data Visualisasi

Analis bisa memvisualisasikan secara *realtime* dari pengolahan data. Untuk menyelesaikan masalah di dalam ilmu komputer, maka digunakanlah *Tableau* [11].

Tableau adalah *software business intelligent* yang gampang digunakan, terutama untuk membuat visualisasi data, data *analysis*, dan pelaporan, dengan sistem yang *drag and drop*. *Tableau* bisa merelasikan data dari *database, spreadsheet,* dan *cloud* data kedalam satu program [28].

2.10. Penelitian Terdahulu

Tabel 2.2. Tabel Penelitian Tedahulu

Nama Jurnal	Penulis/Tahun	Hasil	Kesimpulan
IMPLEMENTASI KOMPARASI *ALGORITMA* KLASIFIKASI MENENTUKAN KELULUSAN MATA KULIAH *ALGORITMA* UNIVERSITAS BUDI LUHUR	Nahot Frastian/2018	*Naïve Bayes* mendapatkan akurasi 96,67%, dan *Random forest* 95,56%	*Algoritma Naïve Bayes* mendapatkan akurasi yang besar, yaitu 96,67% dan mengalahkan akurasi *random forest*
KLASIFIKASI WAKTU KELULUSANMAHASISWA STIKOM BALI MENGGUNAKAN *CHAID REGRESSION–TREES* DAN REGRESI LOGISTIK BINER	I Ketut Putu Suniantara, Muhammad Rusli/2017	metode *CHAID regression trees* mendapatkan akurasi sebesar 91,2%, sedangkan dengan menggunakan	metode *CHAID regression trees* dengan akurasi sebesar 91,2% lebih unggul 1% dari *algoritma* regresi logistik

Nama Jurnal	Penulis/Tahun	Hasil	Kesimpulan
		algoritma regresi logistik akurasi mencapai 90,2%	
Fitur Seleksi *Forward selection* Untuk Menetukan Atribut Yang Berpengaruh Pada Klasifikasi Kelulusan Mahasiswa Fakultas Ilmu Komputer UNAKI Semarang Menggunakan *Algoritma Naive Bayes*	Mohamad Fajarianditya Nugroho, Setyoningsih Wibowo/2017	Hasil klasifikasi metode *Naïve Bayes* dengan akurasi 90,95%	Hasil algortima *Naïve Bayes* untuk kategori lulusan mendapatkan akurasi diatas 90% bisa dibilang besar, yaitu di angka 90,95%
Perbandingan Kinerja Klasifikasi Support *Vector* Machine (Svm) Dan Regresi Logistik Biner Dalam Mengklasifikasikan Ketepatan Waktu Kelulusan Mahasiswa Fmipa Untad	Utami, I T/2018	Prediksi ketepatan waktu kelulusan mahasiswa, regresi logistik biner, dengan akurasi 80,7%	Prediksi sudah termasuk kategori besar karena prediksi ketepatan waktu kelulusan mahasiswa diatas 80% yiatu 80,7%, menggunakan regresi logistik biner
Seleksi Fitur menggunakan *Algoritma* Particle Swarm Optimization pada Klasifikasi Kelulusan Mahasiswa dengan Metode *Naive Bayes*	Evi Purnamasari, Dian Palupi Rini, Sukemi/2017	menggunakan metode *Naive Bayes* menghasilkan nilai akurasi 80%	Hasil algortima *Naïve Bayes* untuk kategori lulusan mendapatkan akurasi 80% bisa dibilang besar
Peningkatan Akurasi Klasifikasi Ketidaktepatan Waktu Kelulusan Mahasiswa Menggunakan Metode *Boosting Neural network*	I Ketut Putu Suniantara, Gede Suwardika, Siti/2020 Soraya	metode *algoritma boosting* pada *Feedforward Neural network* dengan akurasi 74,44% pada iterasi 500	Bahwa *algoritma neural network* untuk klasifikasi kategori lulusan tergolong kecil dibandingkan dengan penelitian

Nama Jurnal	Penulis/Tahun	Hasil	Kesimpulan
			terdahulu lainnya, yaitu 74,44%

Tabel 2.2. menjelaskan tentang tabel penelitian terdahulu dengan kolom nama jurnal, penulis/tahun, hasil, dan kesimpulan yang diambil dari jurnal, sehingga dapatlah kesimpulan bahwa pada penelitian skripsi ini menggunakan algoritma *Naïve Bayes* karena penelitian terdahulu bisa mendapatkan akurasi yang sangat besar yaitu 96,67%, dan regresi logistik 90,2%.