

BAB II

LANDASAN TEORI

2.1. Pengertian Sistem

Menurut Hamidini, (2017), menyatakan bahwa “sistem adalah kumpulan elemen bersifat data, program jaringan yang saling berhubungan, sumber daya manusia, teknologi, perangkat keras dan perangkat lunak yang berinteraksi secara keseluruhan untuk menggapai obyek khusus yang sama.”

Menurut Moch.Irfan, (2014), “Sistem adalah kumpulan dari beberapa bagian terkait yang bekerja sama satu sama lain dan secara bersama-sama mencapai tujuan sistem.”

Muhamad Muslihudin, (2016), mendefinisikan sistem adalah “Komponen atau jaringan sekelompok program, mereka saling berhubungan dan membentuk jaringan untuk mencapai tujuan tertentu.”

Berdasarkan uraian beberapa ahli di atas, maka penulis dapat menarik kesimpulan sebagai berikut: Konsep suatu sistem merupakan kumpulan atau sekelompok elemen atau komponen atau jaringan yang saling berhubungan atau berinteraksi, saling bergantung dan mempunyai hubungan, serta dapat bekerja sama. Raih tujuan dengan orang lain.

2.2. Pengertian Informasi

Menurut Pratama, (2014), Info merupakan hasil pengolahan knowledge dari satu atau lebih sumber, lantas diolah untuk beri tambahan nilai, arti dan manfaat.

Menurut Moch.Irfan, (2014), Informasi adalah sejenis data atau objek yang diproses sebelumnya yang dapat diklasifikasikan dengan benar untuk membuatnya

bermakna bagi penerima, dan kemudian bagi penerima adalah pengetahuan tentang masalah tertentu, yang membantu membuat keputusan yang tepat.

Menurut Djahir, (2014), menjelaskan bahwa informasi adalah hasil berasal dari pengolahan knowledge menjadi wujud tertentu, wujud yang lebih bermanfaat bagi penerimanya, yang melukiskan peristiwa aktual dan bisa digunakan sebagai alat pengambilan ketetapan.

Berdasarkan pendapat pakar yang dikutip di atas, dapat diartikan bahwa Info adalah information yang telah diproses, terstruktur dan disusun dari berbagai sumber jadi wujud yang sangat perlu bagi pengguna dan penerima informasi, yang dimaksudkan untuk pengambilan ketetapan saat ini dan berkelanjutan yang akan mampir.

2.3. Pengertian Sistem Informasi

Menurut Mahatmyo, (2014). Sistem informasi adalah serangkaian prosedur resmi dimana information dikumpulkan, diproses dan dikomunikasikan informasi kepada pengguna.

Menurut Pratama, (2014) menjelaskan bahwa proses Info adalah kombinasi berasal dari empat bagian utama. Empat bagian utama meliputi software, hardware, infrastruktur dan sumber kekuatan manusia (SDM) yang terlatih. Bagian utama keempat dihubungkan dengan untuk membentuk sebuah proses yang sanggup mengubah *knowledge* menjadi Info yang bermanfaat.

Berdasarkan pengertian diatas maka penulis menyimpulkan bahwa konsep sistem informasi adalah suatu sistem yang memberikan informasi manajemen pada saat pengambilan keputusan tentang menjalankan suatu bisnis, dimana sistem

tersebut merupakan gabungan antara manusia, teknologi informasi dan prosedur yang terorganisir.

2.4. Data Mining

Menurut Choiriah, (2019), *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. Dalam data mining terdapat dua pendekatan metode pelatihan, yaitu:

- a. *Unsupervised learning*, metode ini diterapkan tanpa adanya latihan (*training*) dan tanpa ada guru (*teacher*). Guru di sini adalah label dari data.
- b. *Supervised learning*, yaitu metode belajar dengan adanya latihan dan pelatih. Dalam pendekatan ini, untuk menemukan fungsi keputusan, fungsi pemisah atau fungsi regresi, digunakan beberapa contoh data yang mempunyai output atau label selama proses training.

Ada beberapa teknik yang dimiliki *data mining* berdasarkan tugas yang bisa dilakukan, setiap teknik memiliki algoritma masing-masing. Teknik dalam data mining terbagi menjadi enam kategori, yaitu:

- a. Deskripsi

Para peneliti biasanya mencoba menemukan cara untuk mendeskripsikan pola dan trend yang tersembunyi dalam data.

- b. Estimasi

Estimasi mirip dengan klasifikasi, kecuali variabel tujuan yang lebih kearah numerik dari pada kategori.

c. Prediksi

Prediksi memiliki kemiripan dengan estimasi dan klasifikasi. Hanya saja, prediksi hasilnya menunjukkan sesuatu yang belum terjadi (mungkin terjadi dimasa depan).

d. Klasifikasi

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

e. Klastering

Clustering lebih ke arah pengelompokan *record*, pengamatan, atau kasus dalam kelas yang memiliki kemiripan.

f. Asosiasi

Mengidentifikasi hubungan antara berbagai peristiwa yang terjadi pada satu waktu.

2.5. Algoritma *Naïve Bayes*

Bayesian Classification (NBC) adalah pengklasifikasian statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu class. Metode NBC menempuh dua tahap dalam proses klasifikasi teks, yaitu tahap pelatihan dan tahap pengujian/ klasifikasi. Pada tahap pelatihan dilakukan proses analisis terhadap sampel dokumen berupa pemilihan vocabulary, yaitu kata yang mungkin muncul dalam koleksi dokumen sampel, sedapat mungkin dapat menjadi representasi dokumen. Selanjutnya adalah penentuan probabilitas prior bagi tiap kategori berdasarkan sampel dokumen. Pada tahap pengujian/ klasifikasi ditentukan nilai

kategori dari suatu dokumen berdasarkan term yang muncul dalam dokumen yang diklasifikasi. (Kusrini, 2019).

Teorema Bayes merupakan teorema yang mengacu konsep probabilitas bersyarat (Tan, Y., Shenoy, P. P., Chan, M. W., & Romberg, 2016). Teorema Bayes dapat dinotasikan pada persamaan berikut:

$$P(C|X) = \frac{P(x|c)P(c)}{P(x)}$$

Keterangan :

x : Data dengan class yang belum diketahui c

$P(c|x)$: Hipotesis data merupakan suatu class spesifik $P(c|x)$:

Probabilitas hipotesis berdasar kondisi (posteriori probability) $P(c)$:

Probabilitas hipotesis (prior probability)

$P(x/c)$: Probabilitas berdasarkan kondisi pada hipotesis $P(x)$

: Probabilitas c

2.6. *Unified Modelling Language (UML)*

Menurut Nugroho, (2015) menyatakan bahwa dengan adanya *UML*, diharapkan dapat mengurangi kekacauan dalam bahasa pemodelan yang selama ini terjadi dalam lingkungan industri. *UML* diharapkan juga dapat menjawab masalah penotasian dan mekanisme tukar menukar model yang terjadi selama ini.

Menurut A.S & Shalahuddin, (2013) menyatakan bahwa “*UML (Unified Modeling Language)* adalah salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain,

serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah *system*.

a. Tujuan UML

- 1) Memberikan model yang siap pakai, bahasa permodelan *visual* yang *ekspresif* untuk mengembangkan model dan dimengerti secara umum.
- 2) Memberikan bahasa permodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- 3) Menyatukan praktek-praktek yang terdapat dalam permodelan.


b. Diagram-diagram dalam UML




Ada beberapa diagram dalam UML (*Unified Modelling Language*) antara lain :



1) Use Case Diagram

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Adapun simbol dari *use case* diagram antara lain :

Tabel 2. 1 Simbol-simbol Use Case Diagram

No	Simbol	Keterangan
1		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit






		atau aktor; biasanya di-nyatakan dengan meng-gunakan kata kerja di awal frase nama <i>use case</i> .
2	<p>Aktor / <i>Actor</i></p>  <p>Nama Aktor</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanyadinyatakan menggunakan kata benda di awal frase nama aktor.
3	<p>Asosiasi / <i>Association</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4	<p>Ekstensi / <i>Extend</i></p>  <p><<extend>></p>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada

		<p>pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>
5	<p>Generalisasi /</p> 	<p>Hubungan generalisasi dan spesialisasi (umum- khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
6	<p>Menggunakan /</p> <p><i>Include / uses</i></p>  <p><<include>></p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>usecase</i> tambahan dijalankan.</p>

2) Activity Diagram

Activity Diagram menggambarkan berbagai alur aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, dan *decision* yang mungkin terjadi, dan bagaimana alur berakhir. Adapun simbol dari *Activity Diagram* antara lain :



Tabel 2. 2 Simbol Activity Diagram

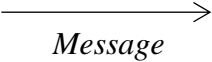
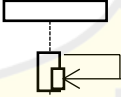
No	Gambar	Keterangan
1.	 <i>Activity</i>	Aktivitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja
2.	 <i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
3.	 Status Awal	Status awal aktiviatas sistem, sebuah diagram aktivitas memiliki status awal.
4.	 Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memilki sebuah status akhir.
5.	 <i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

3) *Sequence Diagram*

Sequence Diagram menggambarkan interaksi antar objek di dalam dan disekitar sistem (termasuk pengguna dan *display*) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek–objek yang terkait). Setiap pesan yang dikirimkan bisa memberikan respon (*return*) relatif pada skenario yang dirancang di *Use Case diagram*. Interaksi yang terjadi bisa bersifat instansiasi sebuah objek maupun *static method* dari sebuah *class*. Berikut ini adalah simbol–simbol dari *sequence diagram*.

Tabel 2. 3 Simbol Sequence Diagram

No	Gambar	Keterangan
1.	 Aktor	Aktor adalah pengguna sistem, pengguna dapat berarti manusia, mesin atau sistem lain atau subsistem dari model apapun yang berinteraksi dengan sistem dari <i>boundary</i> sistem
2.	 <i>Lifeline</i>	Peserta individu dalam interaksi (yaitu jalur hidup yang tidak dapat memiliki multiplisitas)



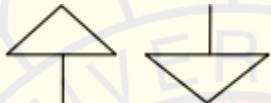

3.		Menunjukkan aliran informasi atau kendali transaksi antar elemen
4.	 <i>Sel-Message</i>	Mencerminkan proses baru atau metode pemanggilan operasi <i>lifeline</i> . Ini adalah spesifikasi pesan biasanya dalam <i>sequence diagram</i>

2.7. Entity Relationship Diagram (ERD)

Menurut Saputra, Saputri, & Akbar (2021) *Entity Relationship Diagram (ERD)* adalah salah satu model yang digunakan untuk mendesain *database* dengan tujuan menggambarkan data yang berelasi pada sebuah *database*. Umumnya setelah perancangan *Entity Relationship Diagram* selesai berikutnya adalah mendesain *database* secara fisik yaitu pembuatan Tabel, *index* tetap dengan mempertimbangkan *performance*. Kemudian setelah *database* selesai dilanjutkan dengan merancang aplikasi yang melibatkan *database*.

Tabel 2. 4 Simbol ERD

No	Simbol	Keterangan
----	--------	------------

1		Memperlihatkan entitas
2		Menunjukkan arah hubungan membaca dari satu entitas ke entitas lain
3		Menunjukkan generalisasi, agregasi, spesialisasi dan pengkomposisi.
4		Hubungan antara entitas dengan atribut mereka.

2.8. MySQL

Menurut (Syahida. Priyanto, 2015) “MySQL adalah salah satu aplikasi DBMS yang sudah banyak oleh para pemogram aplikasi *web*. *SQL* adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis(Kadir, 2014). Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah *SQL*, yang dibuat oleh user maupun program-program aplikasinya. *MySQL* biasanya digunakan atau diinstall bersamaan dengan *XAMPP* sehingga untuk melihat isi tabel bisa menggunakan *PHPmyAdmin*

Contoh DBMS lainnya adalah: PostgreSQL (*freeware*), SQL Server, MS Access dari Microsoft, DB2 dari IBM, Oracle dan Oracle Corp, Dbase, FoxPro, dsb”.

2.9. Python

Python adalah bahasa pemrograman yang diciptakan oleh Guido van Rossum dan populer sebagai bahasa skripting dan pemrograman *Web*. *Python* adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Menurut Syahrudin & Kurniawan (2018), *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Menurut Ahmad & Pambudi, (2015), *Python* mendukung multi paradigma pemrograman, utamanya; namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada *Python* adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, *Python* umumnya digunakan sebagai bahasa skrip meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. *Python* dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. *Python* merupakan salah satu contoh bahasa tingkat tinggi. Contoh lain bahasa tingkat tinggi adalah *pascal*, *c++*, *perl*, *java*, dan sebagainya

2.10. Penelitian Terkait

Berikut ini beberapa penelitian yang terkait dan menjadi referensi pada penelitian ini :

Penelitian dilakukan oleh (Muin, 2016) dengan judul Metode *Naive Bayes* Untuk Prediksi Kelulusan (Studi Kasus: Data Mahasiswa Baru Perguruan Tinggi). *Naive Bayes* dapat melakukan pengklasifikasian dengan metode probabilitas dan statistik, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya. Nilai Presentase keakuratan menunjukkan keefektifan dataset Penerimaan Mahasiswa Baru yang diterapkan ke dalam metode *Naive Bayes Classification*. Implementasi *Naive Bayes* menggunakan aplikasi WEKA dapat menelusuri karakteristik atribut dari dataset dengan luaran Pilihan Lulus. Pengelompokan Pilihan Lulus dilakukan berdasarkan atribut terpilih yaitu Prodi, Pilihan Pertama, Pilihan Kedua dan Nilai Rata-rata.

Penelitian juga dilakukan oleh Castaka Agus Sugianto dan Nadya Selvi Pujiyanita (2019), dengan judul “OPTIMALISASI ALGORITMA C4.5 MENGGUNAKAN ALGORITMA GENETIKA UNTUK PREDIKSI KELULUSAN SISWA SMKN 2 CIMAHI” . Pada angkatan tahun 2015 yang berjumlah 300 siswa, siswa yang tidak lulus pada ujian nasional berjumlah 1 siswa, pada angkatan tahun 2016 yang berjumlah 299 siswa, siswa yang tidak lulus pada ujian nasional berjumlah 1 siswa, pada angkatan tahun 2017 yang berjumlah 302 siswa 100% siswa lulus tepat waktu, namun pada tahun yang akan mendatang masih belum diketahui apakah siswa akan lulus 100% atau tidak. Maka dari itu penelitian ini dilakukan untuk memprediksi kelulusan siswa dengan menggunakan algoritma

C4.5+ algoritma genetika. Algoritma C4.5 mempunyai tingkat accuracy sebesar 99,78%, precision 99,78%, recall 100% dan execution time 1 second. Sedangkan algoritma C4.5 + algoritma genetika mempunyai tingkat accuracy sebesar 99,78%, precision 99,78%, recall 100% AUC 0,500, dan execution time 36 second dan setelah dilakukan pengujian T-test antara algoritma C4.5 dan algoritma C4.5 + algoritma genetika dikatakan signifikan jika $\alpha=0.050$. Dalam hasil penelitian ini didapat $\alpha=1.000$.

Sedangkan menurut Riszki Wijayatun Pratiwi (2017) dalam penelitan berjudul Prediksi Rating Film Menggunakan Metode *Naïve Bayes*. Agar suatu film dapat terus berkembang, tentunya membutuhkan penilaian-penilaian dari para penikmat film, untuk mengetahui selera film yang sesuai dengan para penikmat film. Untuk itu dibutuhkan analisis agar dapat mengetahui bagaimana minat penikmat film yaitu dengan membuat penilaian-penilaian yang nantinya digunakan untuk mengetahui rating suatu film menggunakan metode naïve bayes yaitu metode yang melakukan pendekatan statistika yang fundamental dalam pengenalan pola (pattern recognition). Pendekatan ini didasarkan pada kuantifikasi trade-off antara berbagai keputusan klasifikasi dengan menggunakan probabilitas dan resiko yang ditimbulkan dalam keputusan-keputusan tersebut. Metode tersebut merupakan salah satu metode dari data mining, dengan atribut yang sudah ditentukan, yaitu meliputi genre film, aktor film, bahasa, warna, durasi film, negara, dan lainnya yang dapat digunakan sebagai tolak ukur sutradara untuk membuat film.



**TEKNOLOGI INFORMASI
UNIVERSITAS DARMA PERSADA**