BAB II

LANDASAN TEORI

2.1 Internet

Internet merupakan sebuah jaringan yang berfungsi untuk menghubungkan antara satu media elektronik dengan media lainnya. Jaringan komunikasi inilah yang akan mentransfer data secara tepat dan cepat melalui frekuensi tertentu. Adapun standar global penggunaan internet sendiri telah memakai *Internet Protocol* atau *Transmission Control Protocol (IP/TCP)*.

Jika belum tahu, istilah tersebut adalah bentuk protokol pertukaran paket, yang mana telah dipakai secara global. Kemudian, proses menghubungkan rangkaian internetnya disebut *internet working*. Pengertian internet dijelaskan salah satu ahli IT, Onno W Purbo sebagai suatu media yang digunakan untuk mengefisiensikan proses komunikasi memakai aplikasi (Amira, 2022).

2.2 Perancangan Sistem

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi sebagai perancangan sistem dapat dirancang dalam bentuk bagan alir sistem, yang merupakan alat bantu grafik yang dapat digunakan untuk menunjukan urutan-urutan proses dari sistem. Dalam perancangan sistem informasi, pada umumnya ada 2 (Dua) pemodelan sistem yang lazim digunakan yaitu

pemodelan terstruktur dan pemodelan berorientasi objek. Pada prakteknya kedua pemodelan ini sama penting fungsinya. Pemodelan terstruktur sering kita kenal dengan bagan alir seperti aliran sistem informasi(Flowchart System), Digram Konteks dan Diagran Alir Data (DAD).Sementara untuk pemodelan berorientasi objek umum kita lihat menggunakan *Unified Modeling Language* (UML). UML Digunakan untuk menspesifikasikan, menggambarkan, mambangundan mendokumentasikan dari sistem perangkat lunak (Julianto Simatupang, 2019).

2.3 Klasifikasi KNN

Algoritma *K-Nearest Neighbors* (KNN) adalah salah satu algoritma pembelajaran mesin yang sederhana namun sangat efektif. KNN termasuk dalam kategori algoritma pembelajaran berbasis instance (instance-based learning) dan sering digunakan untuk tugas klasifikasi dan regresi. Prinsip dasar KNN adalah bahwa objek yang serupa atau berdekatan satu sama lain dalam ruang fitur kemungkinan besar memiliki label kelas yang sama atau nilai target yang serupa. Dengan kata lain, KNN memprediksi label atau nilai berdasarkan mayoritas tetangga terdekatnya.

Cara kerja KNN dimulai dengan menentukan nilai K, yaitu jumlah tetangga terdekat yang akan digunakan dalam proses prediksi. Setelah nilai K ditentukan, langkah selanjutnya adalah menghitung jarak antara data yang ingin diprediksi dengan semua data dalam dataset pelatihan. Berbagai metrik jarak dapat digunakan untuk ini, seperti jarak *Euclidean*, jarak *Manhattan*, atau jarak *Minkowski*. Jarak ini digunakan untuk mengidentifikasi K tetangga terdekat.

Setelah K tetangga terdekat teridentifikasi, langkah berikutnya tergantung pada apakah tugasnya klasifikasi atau regresi. Dalam tugas klasifikasi, kelas data yang ingin diprediksi ditentukan berdasarkan mayoritas kelas dari K tetangga terdekat. Misalnya, jika mayoritas dari K tetangga terdekat termasuk dalam kelas tertentu, maka data baru akan diberi label kelas tersebut. Untuk tugas regresi, nilai target dari data baru ditentukan sebagai rata-rata atau median dari nilai target K tetangga terdekat.

Kelebihan KNN antara lain adalah kesederhanaannya dan kemudahan dalam implementasi. KNN juga fleksibel karena tidak membuat asumsi tentang distribusi data dan dapat digunakan untuk klasifikasi serta regresi.

Formula penghitungan KNN adalah sebagai berikut:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Dengan keterangan :

D : jarak kedekatan

x : data training

y : data testing

n : jumlah atribut individu antara 1 s.d. n

f : fungsi similitary atribut i antara kasus X dan kasus Y

i : Atribut individu antara 1 sampai dengan n

2.4 *Naïve Bayes*

Algoritma Naïve Bayes adalah salah satu algoritma pembelajaran mesin yang didasarkan pada teorema Bayes dengan asumsi independensi yang kuat (naïve) antara fitur-fitur dalam dataset. Algoritma ini termasuk dalam kategori algoritma pembelajaran terawasi dan sering digunakan untuk tugas klasifikasi, terutama untuk masalah klasifikasi teks seperti spam detection dan sentiment analysis.

Prinsip dasar Naïve Bayes adalah menghitung probabilitas bahwa sebuah data masuk ke dalam kelas tertentu berdasarkan fitur-fitur yang dimilikinya. Dengan kata lain, algoritma ini memprediksi kelas data baru dengan menghitung probabilitas posterior dari setiap kelas yang diberikan fitur-fitur tertentu, kemudian memilih kelas dengan probabilitas tertinggi.

Cara kerja Naïve Bayes dimulai dengan menghitung probabilitas prior dari setiap kelas dalam dataset pelatihan. Selanjutnya, algoritma menghitung probabilitas likelihood dari setiap fitur diberikan kelas tertentu. Pada saat prediksi, probabilitas posterior dari setiap kelas diberikan fitur-fitur dari data baru dihitung menggunakan teorema Bayes. Kelas dengan probabilitas posterior tertinggi dipilih sebagai prediksi.

Kelebihan dari Naïve Bayes antara lain adalah kecepatan dan efisiensi dalam proses pelatihan dan prediksi, serta performa yang baik pada dataset dengan banyak fitur dan sedikit data pelatihan. Algoritma ini juga tidak memerlukan banyak parameter untuk diatur, sehingga mudah diimplementasikan. Namun, Naïve Bayes memiliki kekurangan, yaitu asumsi independensi antar fitur yang jarang terpenuhi dalam dunia

nyata. Selain itu, algoritma ini juga sensitif terhadap fitur-fitur yang tidak relevan atau berkorelasi tinggi.

Naïve Bayes memiliki beberapa varian, seperti Gaussian Naïve Bayes untuk data kontinu yang diasumsikan berdistribusi normal, Multinomial Naïve Bayes untuk data diskrit seperti frekuensi kata dalam klasifikasi teks, dan Bernoulli Naïve Bayes untuk data biner. Setiap varian memiliki kelebihan dan kekurangan tersendiri tergantung pada jenis data dan masalah yang dihadapi.

Formulasi Naïve Bayes untuk klasifikasi menurut Prasetyo (2012) adalah sebagai berikut:

$$P(Y|X) = \frac{P(Y)\prod_{i=1}^{q} P(X_i|Y)}{P(X)} \quad (1)$$

Dimana:

P(Y|X) = probabilitas data dengan vektor X pada kelas Y.

P(Y) = probabilitas awal kelas Y (prior probability).

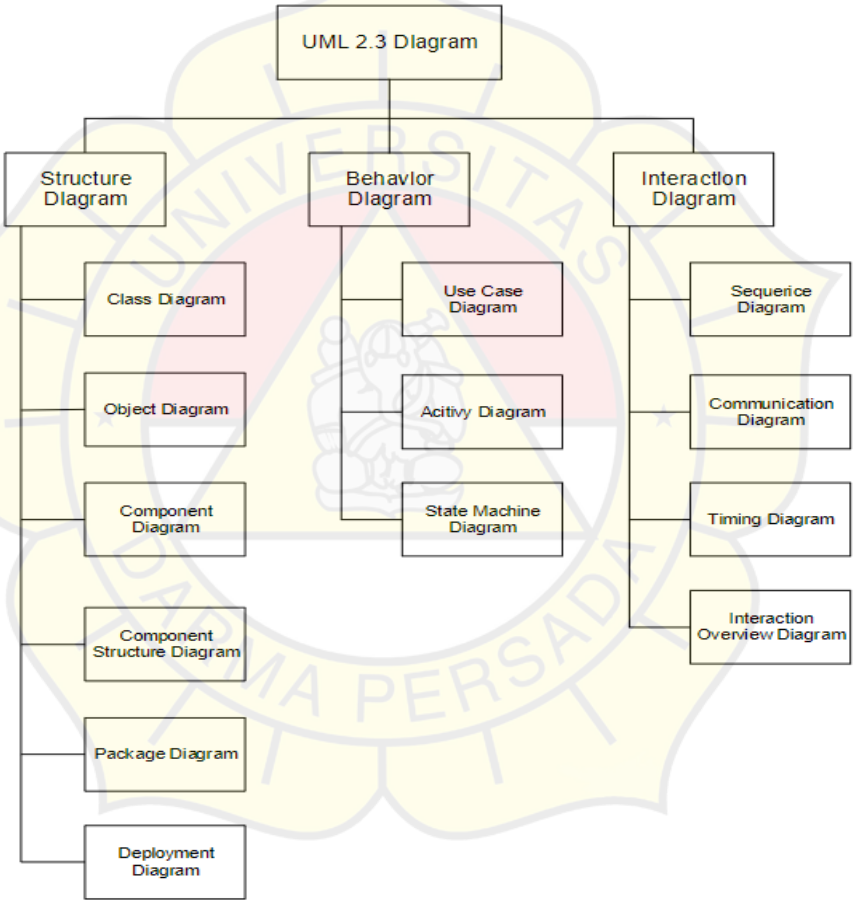
$\prod_{i=1}^{q} P(X_i|Y)$ = probabilitas independen kelas Y dari semua fitur dalam vektor X.

Nilai P(X) = probabilitas dari X.

2.5 Unified Modeling Language (UML)

UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML

hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek. UML terdiri dari 13 diagram yang dikelompokkan dalam 3 kategori.Pembagian kategori dan macam-macam diagram tersebut dapat dilihatpada gambar dibawah ini (Julianto Simatupang, 2019)



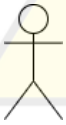
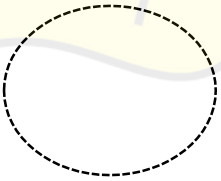
Gambar 2.1 Diagarm UML

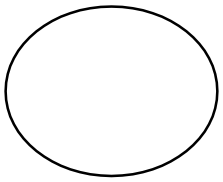
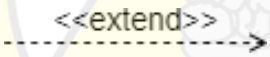
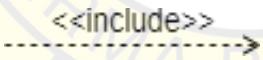
a. *Use case Diagram*

Diagram use case merupakan pemodelan untuk kelakuan system informasi yang akan dibangun. *Usecase* mendeskripsikan sebuah interaksi anatra satu

atau lebih *actor* dengan sistem informasi yang akan dibangun. *Usecase* digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut .Berikut ini adalah simbol-simbol diagram *usecase*, seperti yang terlihat pada tabel dibawah ini (Julianto Simatupang, 2019).

Tabel 2.1 Simbol-Simbol Use Case Diagram

Simbol	Keterangan
 *Generalization*	Merupakan hubungan dimana descendent berbagi perilaku dan struktur data
 *Actor*	Menunjukan siapa saja yang dapat mengakses kegiatan fitur
 *Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar

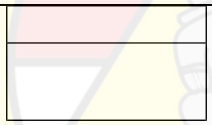
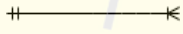
 *Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem untuk menghasilkan suatu hasil yang lebih terukur bagi suatu *actor*
 *System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas
	Menunjukan bahwa suatu usecase merupakan tambahan fungsionalitas dari use case lainnya.
	Menunjukan bahwa suatu usecase seluruhnya merupakan fungsionalitas dari use case lainnya.
	Mengidentifikasi interaksi yang dilakukan oleh *actor* tertentu dengan *use case* tertentu

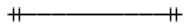
Association	

b. *Class Diagram*

Class diagram menggambar kanstruktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut adalah simbol-simbol *class diagram* seperti terlihat pada tabel 2.2 berikut in (Julianto Simatupang, 2019).

Tabel 2.2 Simbol *Class Diagram*

Simbol	Keterangan
 *Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
 *One to many*	Menggambarkan relasi antar tabel yang bersifat *one to many*.
	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar

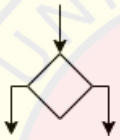
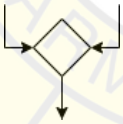
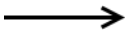
 *One to one*	

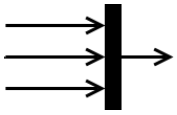
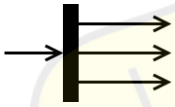
c. *Activity Diagram*

Diagram aktivitas merupakan dari alur kerja atau pergerakan yang terjadi dalam suatu sistem ataupun proses bisnis pada perangkat lunak. Tujuan dari diagram ini ialah untuk memvisualisasikan kegiatan yang dilakukan sistem atau fungsi pergerakan yang dapat dilakukan oleh sistem, bukan aktivitas yang dilakukan oleh aktor. Diagram aktivitas digunakan untuk menggambarkan alur kerja atau proses dalam sistem. Berikut ini adalah beberapa simbol yang ada dalam diagram aktivitas (Julianto Simatupang, 2019).

Tabel 2.3 Simbol *Activity Diagram*

Simbol	Keterangan
 *Activity*	Memperlihatkan bagaimana kegiatan atau aktivitas yang bekerja dalam aliran kerja.
	Awal dimulainya suatu aliran kerja pada *activity diagram*

Initial state	
 *Final state*	Bagian akhir dari suatu aliran kerja pada *activity diagram*.
 *Decision*	Menggambarkan pilihan kondisi yang membuat aliran kerja terbagi menjadi lebih dari satu aliran atau jalur.
 *Merge*	Untuk menggabungkan kembali aliran kerja yang sebelumnya telah terpecah oleh *decision*.
 *Transition*	Menunjukan aliran proses.

 *Join*	Menggabungkan kembali aktivitas yang paralel.
 *Fork*	Memecah *behaviour* menjadi aktivitas yang paralel.

2.6 Android

Android adalah sebuah sistem operasi untuk smartphone dan tablet. Sistem operasi dapat diilustrasikan sebagai jembatan antara piranti *(device)* dan penggunanya, sehingga pengguna bisa berinteraksi dengan *device*-nya dan menjalankan aplikasi – aplikasi yang tersedia pada *device*. Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencangkup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel / *smartphone*. Android merupakan generasi baru *platform mobile* yang memberikan pengembangan untuk melakukan pengembangan sesuai dengan yang diharapkannya. Sistem operasi yang mendasari Android dilisensikan dibawah GNU, *General Public Lisensi* versi 2

(GPLv2), yang sering dikenal dengan istilah "*copyleft*" lisensi dimana setiap perbaikan pihak ketiga harus terus jatuh dibawah *terms*. Android didistribusikan di bawah lisensi Apache Software (ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya (Sari, Y. P., & Ali, R, 2019).

2.7 Android Studio

Android Studio merupakan sebuah *Integrated Development Environment* (IDE) khusus untuk membangun aplikasi yang berjalan pada *platform* android. Android studio ini berbasis pada IntelliJ IDEA, sebuah IDE untuk Bahasa pemrograman Java.Yeka Hendriyani dan Karmila Suryani (2019: 117) mengungkapkan: "Android Studio merupakan sebuah integrated development environment (IDE) untuk platform android".

2.8 Flutter

Flutter adalah sebuah platform pengembangan aplikasi mobile yang dikembangkan oleh Google untuk menciptakan aplikasi yang dapat berjalan pada iOS, Android, web, dan juga desktop. Menurut Kurosaki (2020), platform ini memiliki beberapa keunggulan seperti kemampuan hot reload, beragam widget, dan desain antarmuka yang menarik. Tujuan utama Flutter, sebagaimana diungkapkan oleh Giordano (2019), adalah untuk mengubah cara pengembangan aplikasi seluler dengan menyediakan semua alat yang diperlukan untuk menghasilkan aplikasi yang menakjubkan tanpa mengorbankan kinerja dan skalabilitas. Dalam struktur intinya, Flutter memiliki konsep-konsep yang berfokus pada kinerja aplikasi dan antarmuka pengguna, dengan menggunakan bahasa pemrograman Dart untuk meningkatkan

produktivitas pengembang dan membangun aplikasi yang dioptimalkan untuk distribusi publik (Biessek, 2019).

Flutter memanfaatkan Dart untuk mengembangkan antarmuka pengguna (UI). Secara deklaratif, Flutter membangun UI untuk mencerminkan status aplikasi; ketika status (data) berubah, UI pun ikut berubah. Flutter menciptakan instance baru dari widget untuk setiap perubahan, seperti yang.

2.9 Dart

Dart adalah bahasa pemrograman utama yang dipakai untuk mengembangkan framework flutter dan dikembangkan langsung oleh google. (Kurosaki, 2020) Flutter membutuhkan bahasa modern tingkat tinggi agar mampu memberikan pengalaman terbaik kepada pengembang dan memungkinkan pembuatan aplikasi seluler yang mengagumkan. Dart bisa digunakan untuk membuat aplikasi web, android, ios dan juga menjalankan server. (Biessek, 2019) Dart dapat digunakan untuk membuat satu aplikasi yang kodingannya dapat digunakan di berbagai platform. Aplikasi yang dibuat juga bisa digunakan pada android maupun ios. Dart juga bisa digunakan untuk web development. (Wiraganda, 2019).

Dart adalah bahasa lintas platform modern yang terus meningkatkan fiturfiturnya, membuatnya lebih baik dan fleksibel. Itulah mengapa flutter memilih dart sebagai bahasa yang digunakan. (Biessek, 2019)

2.10 JSON (*JavaScript Object Notation*)

JSON (*JavaScript Object Notation*) merupakan format yang ringan untuk memasukan data ke dalam sebuah variabel. Sangat mudah dimengerti dan diimplementasikan oleh manusia, dan mudah juga untuk komputer dalam melakukan parsingnya. JSON merupakan bagian dari bahasa pemrograman JavaScript (Standard ECMA-262 3rd Edition – December 1999). JSON merupakan format teks yang sepenuhnya independen tetapi menggunakan konvensi yang familiar dengan bahasa pemrograman dari keluarga-C, termasuk C, C++, C#, Java, JavaScript, Perl, Python, dan sebagainya. JSON adalah struktur data yang universal, dalam artian bisa digunakan dalam berbagai bahasa pemrograman. Hampir semua bahasa pemrograman mendukung penuh JSON dalam berbagai format. Hal ini memungkinkan format data yang dapat dipertukarkan menggunakan bahasa pemrograman juga menggunakan dasar dari struktur JSON. (Dwija. 2015)

2.11 Basis data *(database)*

basis data terdiri dari kata basis dan data. basis data dapat di artikan sebagai marks atau gudang. sedangkan data adalah catatan atas kumpulan fakta nyata yang mewakilkan objek seperti manusia,barang,hewan yang di wujudkan dalam bentuk huruf, angka, simbol, gambar,teks bunyi atau kombinasinya. Pengertian dari basis data atau database itu sendiri adalah himpunan kemlopok data yang saling terhubung dan diorganisasi sedemikian rupa supaya kelak dapat dimanfaatkan kembali secara cepat dan mudah. kumpulan data dalam bentuk file/table/arsip yang saling terhubung dan

tersimpan dalam media penyimpanan elektronis, untuk kemudahan dalam pengaturan, pemilihan,penglomokan dan pengogranisasian data sesuai tujuan (Rachmadi, 2020).

2.12 MySQL

MySQL adahlah salah satu jenis *database server* yang sangat populer, hal ini disebabkan karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. MySQL bersifat *Open Source*, *Software* ini dilengkapi dengan *Source code* (kode yangdipakai untuk membuat MySQL) (Winanjar & Susanti, 2021)

2.13 Tinjauan Pustaka

Tabel 2.4 Tinjauan Pustaka

No	Penulis	Judul	Publikasi	Metode	Tujuan	Temuan
1	Naisah Marito Putry dan Betha Nurina	Komparasi Algoritma Knn Dan Naïve Bayes Untuk Klasifikasi Diagnosis Penyakit Diabetes Melitus	Jurnal Sains dan Manajemen	Penelitian ini menggunakan dua algoritma klasifikasi, yaitu Naïve Bayes dan KNN.	Penelitian dilakukan untuk membandingkan algoritma Naïve Bayes dan K-Nearest Neighbor (KNN) dalam klasifikasi diagnosis penyakit	Nilai akurasi algoritma Naïve Bayes lebih tinggi dibandingkan dengan algoritma KNN. Artinya Algoritma

					diabetes melitus	Naïve Bayes lebih unggul dalam hal akurasi untuk klasifikasi diagnosis penyakit diabetes melitus dibanding kan algoritma KNN
2	Putri Septiani Indah Pratiwi, M. Ghofar Rohman, Miftahus Sholihin	Sistem Pakar Penyakit Telinga Mengguna kan Metode Naïve Bayes	*Generati on Journal*	Penelitian ini menggun akan adalah metode Naïve Bayes.	Penelitian ini dilakukan untuk menerapkan pendekatan Naïve Bayes dalam sistem diagnosis penyakit telinga	Sistem pakar diagnosis penyakit telinga dengan metode Naïve Bayes diuji sebanyak 10 kali dengan hasil 9 data

						uji sesuai dan 1 data uji tidak sesuai. Sistem ini memiliki akurasi sebesar 90%.
3	Unius Pratama, Fauziah, dan Ira Diana Sholihati	Metode K-Nearest Neighbor Dan Naive Bayes Dalam Menentukan Status Gizi Balita	Penerapan Kecerdasan Buatan	Penelitian ini mengguna kan dua algoritma klasifikasi, yaitu Naïve Bayes dan K-Nearest Neighbor	Mempermudah masyarakat, khususnya orang tua, dalam mengatasi permasalahan gizi balita dengan menyediakan sistem informasi berbasis website	Tingkat akurasi metode Naïve Bayes lebih unggul dengan persentase nilai sebesar 87,5% dibanding kan dengan metode KNN yang memiliki persentase akurasi

						sebesar 71,25% dengan k = 3.