



BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Umum

2.1.1 Pengertian Internet

Sofana (2015:5), “Internet adalah interkoneksi jaringan computer skala besar (mirip WAN), yang dihubungkan menggunakan protocol khusus jadi sebenarnya Internet merupakan bagian dari WAN. Cakupan internet adalah satu dunia bahkan tidak menutup kemungkinan antarplanet. Koneksi antar jaringan computer dapat dilakukan berkat dukungan protocol yang khas, yaitu TCP/IP (*Transmission Control Protocol/Internet Protocol*)”.

Dari pendapat diatas dapat disimpulkan bahwa internet adalah sebuah jaringan yang menghubungkan seluruh komputer di seluruh dunia.

2.1.2 Pengertian Android

Setiawan dan Ramdany (2019:2), “Android adalah sistem operasi untuk perangkat mobile berbasis Linux yang awalnya dikembangkan oleh Android Inc.”

Menurut Firly (2020:2), “Dalam bahasa Inggris istilah Android berarti Robot yang menyerupai manusia, hal tersebut dapat terlihat jelas pada icon Android yang menggambarkan sebuah robot berwarna hijau yang memiliki sepasang tangan dan kaki.”

Dari pengertian diatas dapat ditarik kesimpulan, Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet.

2.2 Pengertian Judul

2.2.1 Pengertian Aplikasi

Menurut Reksoatmodjo (2018:24), “Aplikasi adalah program komputer yang digunakan untuk membuat maupun memelihara basis data, kemudian melakukan diseminasi data dan informasi kepada semua pengguna akhir.”



Pane et al, (2020:53),”Aplikasi adalah suatu perangkat lunak (*software*) atau program computer yang beroperasi pada sistem tertentu yang diciptakan dan dikembangkan untuk melakukan perintah tertentu.”

Kesimpulannya, Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari user (pengguna).

2.2.2 Pengertian *Customer Service*

Menurut Kasmir (2018:180),”Customer Service adalah setiap kegiatan yang diperuntukan atau ditujukan memberikan kepuasan nasabah melalui pelayanan yang dapat memenuhi keinginan dan kebutuhan nasabah.”

Sahir et al, (2020:93), “Customer Service ialah setiap aktivitas yang diperuntukkan atau diajukan untuk memberikan kepuasan nasabah.”

Berdasarkan pendapat di atas dapat disimpulkan bahwa customer service adalah kegiatan yang diperuntukan atau ditujukan memberikan kepuasan nasabah melalui pelayanan yang dapat memenuhi keinginan dan kebutuhan nasabah.

2.2.3 Pengertian *Chatbot*

Elviana (dikutip Khan & Das, 2018) Chatbot adalah program komputer yang memproses bahasa alami dari pengguna dan menghasilkan respon kemudian dikirim kembali ke pengguna. Saat ini, chatbot didukung oleh mesin berbasis aturan atau mesin kecerdasan buatan (AI) yang berinteraksi dengan pengguna melalui antarmuka yang berawal berbasis teks, namun seiring kemajuan teknologi suara yang mulai berkembang, perusahaan seperti Google, Apple, dan Amazon telah memulai perkembangan agen kecerdasan buatan (chatbot) untuk suara.

2.2.4 Pengertian *Keluhan*

Keluhan adalah salah satu cara tambahan untuk mengetahui harapan pelanggan. Dengan adanya keluhan kita menjadi tahu mana yang belum memenuhi harapan pelanggan. Keluhan adalah alat ukur paling jujur sekaligus murah mengenai apakah standar yang kita tetapkan sudah berjalan atau sebaliknya.



2.2.5 Pengertian Naive Bayes

Manalu et al, (2017:17) Naive Bayes merupakan sebuah pengklasifikasian probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan. Algoritma menggunakan teorema Bayes dan mengasumsikan semua atribut independen atau tidak saling ketergantungan yang diberikan oleh nilai pada variabel kelas.

Naive Bayes didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai output. Dengan kata lain, diberikan nilai output, probabilitas mengamati secara bersama adalah produk dari probabilitas individu. Keuntungan penggunaan Naive Bayes adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (Training Data) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. Naive Bayes sering bekerja jauh lebih baik dalam kebanyakan situasi dunia nyata yang kompleks dari pada yang diharapkan.

Persamaan dari teorema Bayes adalah:

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

Di mana :

X : Data dengan class yang belum diketahui

H : Hipotesis data merupakan suatu class spesifik

$P(H|X)$: Probabilitas hipotesis H berdasar kondisi X (posteriori probabilitas)

$P(H)$: Probabilitas hipotesis H (prior probabilitas)

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

$P(X)$: Probabilitas X



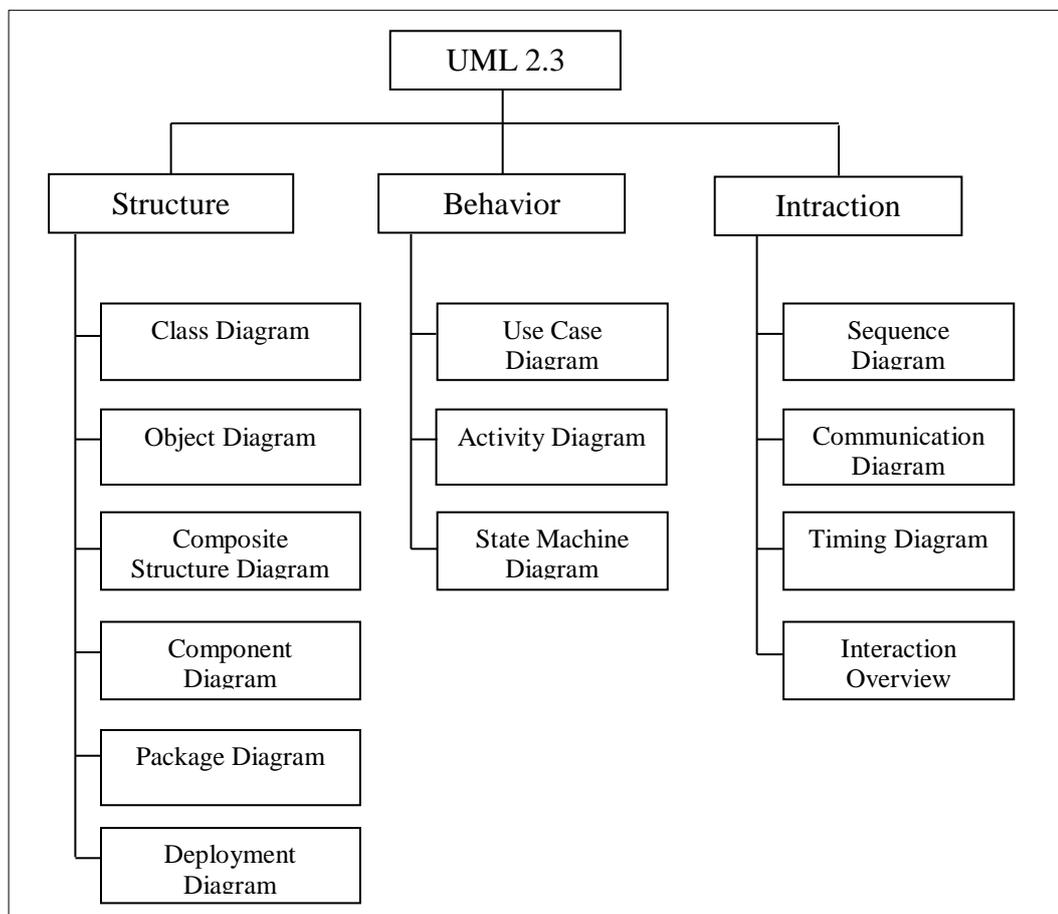
2.2.6 Pengertian Aplikasi *Chatbot Customer Service* pada Plasa Telkom Curup Menggunakan Metode *Naïve Bayes* Berbasis *Android*

Aplikasi *Chatbot Customer Service* pada Plasa Telkom Curup Menggunakan Metode *Naïve Bayes* Berbasis *Android* adalah suatu aplikasi yang dapat membantu dan mempermudah karyawan Plasa Telkom Curup dalam melayani pengaduan keluhan gangguan.

2.3 Pengertian Khusus

2.3.1 Pengertian *UML(Unified Modeling Language)*

Menurut S. Rosa dan Shalahuddin (2018:140), “Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori”. Pembagian kategori dan macam-macam diagram Menurut Sukamto dan Shalahuddin tersebut dapat dilihat pada gambar dibawah:



Gambar 2.1 Macam-macam Diagram UML



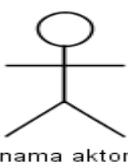
Penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukamto dan Shalahuddin (2018:141):

- 1) *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- 2) *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
- 3) *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

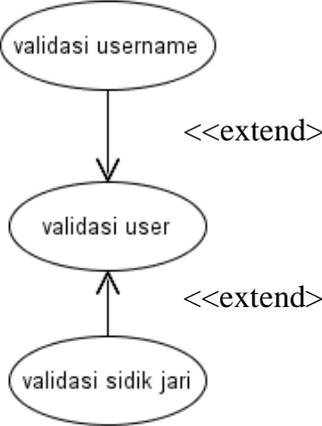
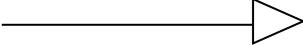
2.3.2 Usecase Diagram

Menurut S. Rosa dan Shalahuddin (2018:155-158) “*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem.” Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

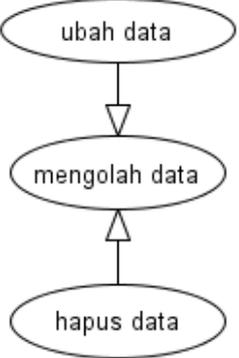
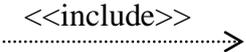
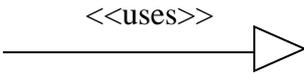
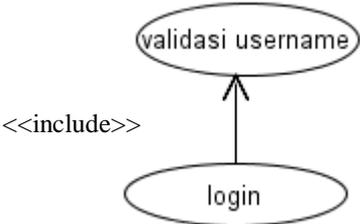
Tabel 2.1 Simbol-simbol *Use case Diagram*

Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>fungsi yang disediakan sistem sebagai unit-unit saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>aktor / <i>actor</i></p> 	<p>orang, proses, atau sistem yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang; biasanya kata benda di awal frase nama aktor</p>

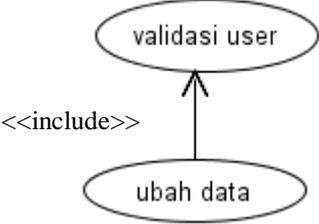
Lanjutan Tabel 2.1 Simbol-simbol *Use case Diagram*

Simbol	Deskripsi
asosiasi / <i>association</i> 	komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> .
ekstensi / <i>extend</i> 	relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya
	 <p>arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya</p>
Generalisasi / <i>generalization</i> 	hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:

Lanjutan Tabel 2.1 Simbol-simbol *Use case Diagram*

Simbol	Deskripsi
	 <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
menggunakan / include / uses 	relasi 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 ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i> : <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut:  <p><i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di</p>

Lanjutan Tabel 2.1 Simbol-simbol *Use case Diagram*

Simbol	Deskripsi
	<p>tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p>  <pre> graph BT UC1(ubah data) -- "<<include>>" --> UC2(validasi user) </pre> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber : S. Rosa. A dan Shalahuddin (2018:155-158)

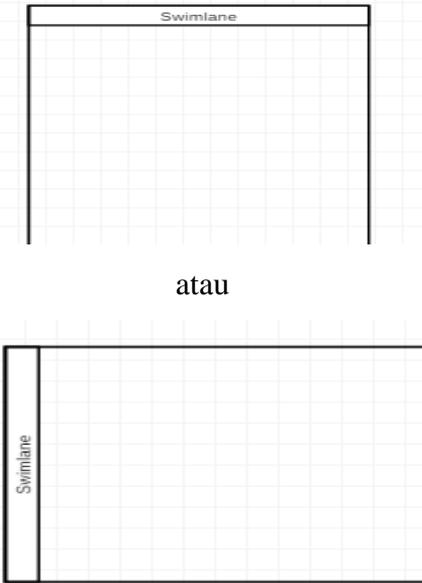
2.3.3 Activity Diagram

S. Rosa dan Shalahuddin (2018:161-163), menjelaskan tentang *activity diagram* sebagai berikut :

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

Tabel 2.2 Simbol-simbol pada *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber : S. Rosa. A dan Shalahuddin (2018:161-163)

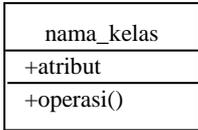
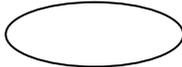
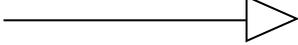


2.3.4 ClassDiagram

S. Rosa dan Shalahuddin (2018:141-146), menyebutkan *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Diagram Class* dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Adapun simbol-simbol yang digunakan dalam *class diagram* adalah sebagai berikut:

Tabel 2.3 Simbol-simbol pada *Class Diagram*

Simbol	Deskripsi
<p>kelas</p> 	Kelas pada struktur sistem
<p>antarmuka / interface</p>  <p>nama_interface</p>	Sama dengan konsep interface dalam pemrograman berorientasi objek
<p>asosiasi / association</p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i>
<p>asosiasi berarah / <i>directed association</i></p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>generalisasi</p> 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus)



Lanjutan Tabel 2.3 Simbol-simbol pada *Class Diagram*

Simbol	Deskripsi
kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas
agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

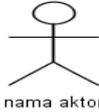
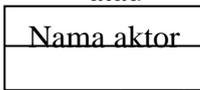
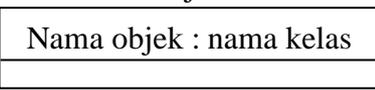
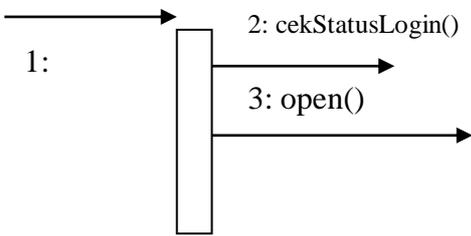
Sumber : S. Rosa. A dan Shalahuddin (2018:141-146)

2.3.5 *Sequence Diagram*

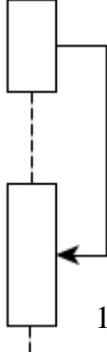
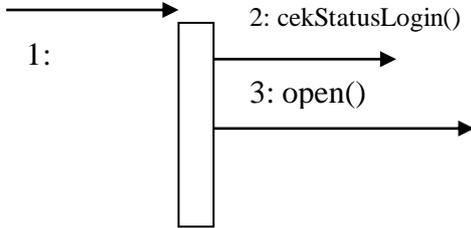
S. Rosa dan Shalahuddin (2018:165-167), menjelaskan bahwa diagram *sequence* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram *sequence* juga dibutuhkan untuk melihat skenario yang ada pada use case.

Banyaknya diagram *sequence* yang harus digambarkan adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram *sequence* sehingga semakin banyak *use case* yang didefinisikan maka diagram *sequence* yang harus dibuat juga semakin banyak. Berikut simbol-simbol pada *Sequence Diagram* :

Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

Simbol	Deskripsi
<p>Actor</p>  <p>nama aktor</p> <p>atau</p>  <p>Nama aktor</p> <p>tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p>  <p>Nama objek : nama kelas</p>	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>Maka <code>cekStatusLogin ()</code> dan <code>open()</code> dilakukan di dalam metode <code>login()</code> aktor tidak memiliki waktu aktif</p>
<p>Pesan tipe create</p> <p><<create>></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>

Lanjutan Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

Simbol	Deskripsi
<p>Pesan tipe call</p> <p><<create>></p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>1: nama_metode()</p> <p>arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>1: 2: cekStatusLogin() 3: open()</p> <p>Maka cekStatusLogin () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif</p>
<p>Pesan tipe create</p> <p><<create>></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>

Sumber : S. Rosa. A dan Shalahuddin (2018:165-167)



2.3.6 Kamus Data

Menurut Kristanto (2018:118) Pada kamus data (*data dictionary*), semua jenis data yang terlibat dalam proses yang terjadi, didefinisikan dan dikumpulkan dalam bentuk penyajian.

Sukanto dan Shalahuddin (2018:73-74) menyatakan bahwa, “Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan)”. Berikut beberapa simbol-simbol yang terdapat pada kamus data :

Tabel 2.5 Simbol-simbol Kamus Data

No	Simbol	Keterangan
1.	=	Disusun atau terdiri dari
2.	+	Dan
3.	[]	Baik...atau...
4.	{ ⁿ }	N kali/ bernilai banyak
5.	()	Data opsional
6.	*...*	Batas komentar

Sumber : S. Rosa. A dan Shalahuddin (2018:73-74)

2.4 Pengertian Program

2.4.1 Pengertian Java

Menurut Haqi (2019:1) menjelaskan, Java adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Menurut define Sun, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada computer standalone ataupun pada lingkungan jaringan.

Menurut Nofriadi (2018:1), Java merupakan salah satu dari sekian banyak bahasa pemrograman yang dapat dijalankan diberbagai sistem operasi termasuk telepon genggam.



2.4.2 Pengertian Android Studio

Menurut Firly (2020:13) Android studio merupakan integrated development environment (IDE) atau dalam artian lain adalah sebuah lingkungan pengembangan terintegrasi resmi yang memang dirancang khusus untuk pengembangan sistem operasi Google Android.

Herlinah dan Musliadi (2019:4) menjelaskan Android Studio merupakan Lingkungan Pengembangan Perangkat Lunak Terpadu - Integrated Development Environment (IDE). Selain merupakan editor kode IntelliJ dan alat pengembangan yang berdaya guna.

2.4.3 Pengertian Basis Data

Kristanto (2018:79) “Basis Data adalah kumpulan data, yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi. Sebagai contoh, basis data universitas berisi informasi mengenai: Entiti, semisal mahasiswa, fakultas, mata kuliah, dan ruang kelas. Relasi diantara entitas, seperti pengambilan kuliah yang dilakukan oleh mahasiswa, staf pengajar di fakultas, dan penggunaan ruang perkuliahan. “

Sedangkan menurut Reaksoatmodjo (2018:24),“basis data adalah salah satu komponen yang menjadi tumpuan keberhasilan sebuah sistem informasi, oleh karena itu gambaran tentang langkah-langkah yang harus diambil dalam perancangan, analisis, dan implementasi basis data tidak dapat dipisahkan dari pengembangan keseluruhan sistem informasi.

Kesimpulannya adalah basis data merupakan kumpulan informasi yang disimpan didalam komputer secara sistematis dan saling berhubungan satu dengan yang lain yang tujuan utamanya adalah memelihara data yang sudah ada yang diolah.