



BAB II

TINJAUAN PUSTAKA

2.1 Teori Judul

2.1.1 Aplikasi

Menurut Marimin, dkk. (2011:43) Aplikasi merupakan program yang secara langsung dapat melakukan proses-proses yang digunakan dalam komputer oleh pengguna, kumpulan dari file-file tertentu yang berisi kode program yang menghubungkan antara pengguna dan perangkat keras Komputer.

Menurut Hengky W. Pramana (2020:53) mengatakan bahwa Aplikasi merupakan suatu perangkat lunak yang dibuat khusus untuk memenuhi kebutuhan berbagai aktivitas dan pekerjaan layaknya aktivitas perniagaan, periklanan, pelayanan masyarakat, *game*, dan beraneka aktivitas lainnya yang dikerjakan oleh manusia.

Dari kedua pengertian diatas dapat disimpulkan bahwa Aplikasi adalah sekumpulan perintah atau kode yang disusun secara sistematis untuk menjalankan suatu perintah yang diberikan oleh manusia melalui komponen atau *hardware* komputer yang digunakan oleh manusia dalam menjalankan program aplikasi, dengan demikian bisa membantu untuk memberikan solusi dari apa yang diinginkan.

2.1.2 Pengolahan Data

Menurut Jaka Abdul Rohim (2016:12) “Pengolahan data adalah serangkaian operasi atas informasi yang direncanakan guna mencapai tujuan atau hasil yang diinginkan”.

2.1.3 Kerja Sama

Menurut Kamus Besar Bahasa Indonesia, “Kerja sama adalah kegiatan atau usaha yang dilakukan oleh beberapa orang (lembaga, pemerintah dan sebagainya) untuk mencapai tujuan bersama. peningkatan kemampuan efektivitas tenaga pengajar untuk melakukan pekerjaan.



2.1.4 Perguruan Tinggi Swasta

Menurut Undang-Undang Nomor 12 Tahun 2012 tentang Pendidikan Tinggi, perguruan tinggi swasta adalah perguruan tinggi yang didirikan atau diselenggarakan oleh masyarakat dengan membentuk badan penyelenggara berbadan hukum yang berprinsip nirlaba, misalnya yayasan. Perbedaannya dengan perguruan tinggi negeri, perguruan tinggi swasta tidak didirikan oleh pemerintah atau negara tujuan bersama. peningkatan kemampuan efektivitas tenaga pengajar untuk melakukan pekerjaan.

2.1.5 Lembaga Layanan Pendidikan Tinggi (LLDIKTI)

Lembaga Layanan Pendidikan Tinggi Wilayah (LLDIKTI) adalah instansi pemerintah di lingkungan Kementerian Pendidikan dan Kebudayaan di suatu wilayah tertentu yang merupakan transformasi dari Koordinasi Pendidikan Tinggi Swasta (Kopertis) yang dahulu hanya mengkoordinasikan perguruan tinggi swasta di wilayah kerja masing-masing. Dengan memperhatikan luas wilayah dan jumlah Perguruan Tinggi Swasta di wilayah bersangkutan, hingga saat ini ada 12 (dua belas) Lembaga Layanan Pendidikan Tinggi Wilayah di seluruh Indonesia.

2.1.6 Metode *Prototype*

Menurut Andikos dan Gusteti dalam Michael dan Gustina (2019:61), “*Prototype* merupakan suatu metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat suatu program dengan cepat dan bertahap sehingga segera dapat dievaluasi oleh pemakai. *Prototype* mewakili model produk yang akan dibangun atau mensimulasikan struktur, fungsional, dan operasi sistem”.

Menurut Darmawan dalam kusumah dan permana (2018:112-113), “*Prototype* adalah suatu versi dari sebuah sistem potensial yang memberikan ide bagi para pengembang dan calon pengguna, bagaimana sistem akan berfungsi dalam bentuk yang telah selesai”.



Berdasarkan pernyataan tersebut, dapat disimpulkan bahwa Metode *prototype* adalah metode dalam pengembangan sistem yang didasarkan pada konsep model kerja sehingga dapat digunakan dengan baik. *Prototype* juga bisa dibangun melalui beberapa *tool* pengembangan untuk menyederhanakan proses. Adapun tahapan-tahapan dalam metode *prototype* adalah sebagai berikut :

1. Analisis Kebutuhan Sistem
2. Desain Sistem
3. Pengujian Sistem
4. Implementasi

Kelebihan :

1. Prototype melibatkan user dalam analisa dan desain.
2. Punya kemampuan menangkap requirement secara konkret.
3. Digunakan untuk memperluas SDLC.

Kekurangan :

1. Proses analisis dan perancangan terlalu singkat.
2. Mengesampingkan alternatif pemecahan masalah.
3. Bisanya kurang fleksible dalam menghadapi perubahan.
4. *Prototype* yang dihasilkan tidak selamanya mudah dirubah dan cepat selesai.

2.1.7 Aplikasi Pengolahan Data Kerja Sama Perguruan Tinggi Swasta Berbasis Android Pada Lembaga Layanan Pendidikan Tinggi Wilayah II Daerah Sumatera Bagian Selatan Menggunakan Metode *Prototype*

Aplikasi Pengolahan Data Kerja Sama Perguruan Tinggi Swasta Berbasis Android Pada Lembaga Layanan Pendidikan Tinggi Wilayah II Daerah Sumatera Bagian Selatan Menggunakan Metode *Prototype* merupakan suatu program komputer yang digunakan untuk mengelola data kerja sama.



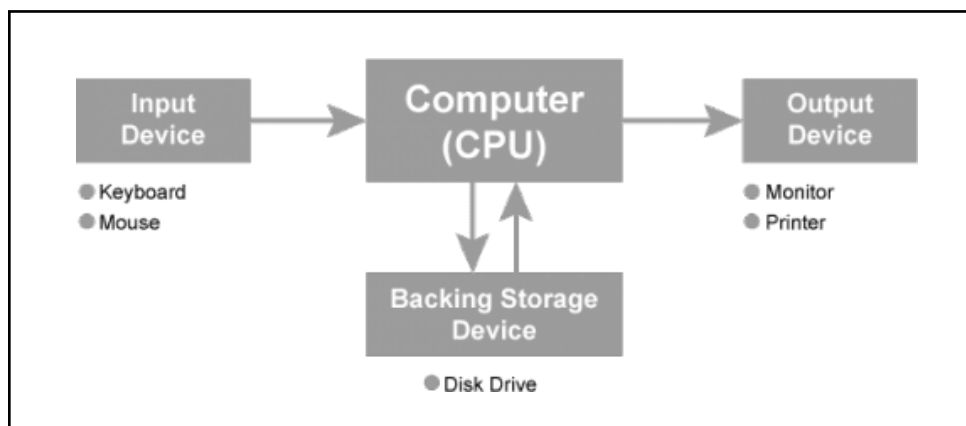
2.2 Teori Umum

2.2.3 Pengertian Komputer

Menurut Donald H. Sanders (1999:1), “Komputer merupakan sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan supaya secara otomatis menerima dan menyimpan data input, memprosesnya dan menghasilkan output dibawah pengawasan suatu langkah-langkah instruksi-instruksi program yang tersimpan di memori (*stored program*)”.

Sedangkan, menurut Sawyer dan Krisbiantoro (2018), "Komputer merupakan sebuah mesin multiguna yang dapat dilakukan pemrograman, dan dapat diprogram, dimana mesin-mesin tersebut dapat menerima data berupa fakta dan gambar kasar, lalu kemudian memproses atau memanipulasi data tersebut menjadi sebuah informasi yang bisa digunakan oleh user-nya." Disimpulkan bahwakomputer adalah alat elektronik yang dapat mengolah data digital, dengan mengikuti serangkaian program yang digunakan untuk membantu pelaksanaan pekerjaan.

Proses kerja komputer biasa disebut juga proses pengolahan data terdiri dari 3 tahapan dasar yang disebut dengan siklus pengolahan data (*data processing cycle*) yaitu *input*, *processing* dan *output*. Cara kerja komputer dapat kita gambarkan sebagai berikut:



Gambar 2.1 Cara Kerja Komputer



2.2.3.1 *Input Device*

Input Device adalah perangkat keras komputer yang berfungsi untuk memasukkan data ke dalam memori komputer, seperti *keyboard, mouse, joystick* dan lain-lain.

2.2.3.2 Prosesor

Prosesor adalah perangkat utama komputer yang mengelola seluruh aktifitas komputer itu sendiri.

2.2.3.3 Memori

Memori adalah media penyimpanan data pada komputer. Memori ini terbagi atas dua macam, yaitu *Read Only Memory (ROM)* dan *Random Access Memory (RAM)*. Komputer terdiri dari tiga komponen utama yang tidak dapat dipisahkan,

1. *Hardware* (Perangkat Keras), merupakan peralatan fisik dari komputer yang dapat kita lihat dan rasakan. *Hardware* ini terdiri dari *Input/Output Device (I/O Device)*, *Storage Device*, *Monitor Screen*, *Casing Unit*, *Central Processing Unit (CPU)*.
2. *Software* (Perangkat Lunak), merupakan program-program komputer yang berguna untuk menjalankan suatu pekerjaan sesuai dengan yang dikehendaki. Program tersebut ditulis dengan bahasa khusus yang dimengerti oleh komputer. *Software* terdiri dari beberapa jenis, yaitu sistem operasi, program *utility*, Program Aplikasi, program paket, bahasa pemrograman.
3. *Brainware (User)*, adalah unsur yang paling menentukan di dalam sistem komputer, karena manusia mempunyai otak (*brain*) yang dinamis dan bisa membuat kebijaksanaan. Manusia bisa mempertimbangkan banyak elemen sekaligus dalam membuat keputusan, termasuk menggunakan intuisi.

2.2.4 Pengertian Software (Perangkat Lunak)

Menurut Husein dan Fauzi (2019), Perangkat lunak komputer adalah



perangkat berbasis program (tidak berbentuk fisik) atau *algoritma* yang digunakan untuk mengoperasikan dan mengontrol perangkat keras komputer. Komputer tidak dapat dioperasikan tanpa perangkat lunak. Oleh karena itu, setiap komputer membutuhkan instalasi perangkat lunak agar dapat digunakan.

Sedangkan, menurut Jayanti dan Sumiari (2018:5). "*Software* adalah perangkat lunak yang digunakan dalam suatu sistem basis data". Berdasarkan beberapa definisi perangkat lunak diatas, dapat disimpulkan bahwa perangkat lunak adalah instruksi - instruksi atau data yang diformat secara digital, yang bisa dibaca dan ditulis oleh komputer sesuai kehendak pemakai.

Software terdiri dari beberapa jenis, yaitu:

2.2.4.1 Sistem Operasi, seperti *DOS, Unix, Novell, OS/2, Windows*.

Software yang berfungsi untuk mengaktifkan seluruh perangkat yang terpasang pada komputer sehingga masing-masing dapat saling berkomunikasi.

2.2.4.2 Program *Utility*, seperti *Norton Utility, Scandisk, PC Tools*.

Program *utility* berfungsi untuk membantu atau mengisi kekurangan/kelemahan dari sistem operasi, misalnya *PC Tools* dapat melakukan perintah format sebagaimana *DOS*, tapi *PC Tools* mampu memberikan keterangan dan animasi yang bagus dalam proses pemformatan.

2.2.4.3 Program Aplikasi, seperti *GL, MYOB, Payroll*.

Program yang khusus melakukan suatu pekerjaan tertentu, seperti program gaji pada suatu perusahaan. Maka program ini hanya digunakan oleh bagian keuangan saja tidak dapat digunakan oleh departemen yang lain.

2.2.4.4 Program Paket, seperti *MS-Word, MS-Excel, Power Point*.

Program yang disusun sedemikian rupa sehingga dapat digunakan. oleh banyak orang dengan berbagai kepentingan. Seperti MS- Word, dapat digunakan oleh departmen keuangan untuk membuat nota, atau bagian administrasi untuk membuat surat penawaran dan lain sebagainya.



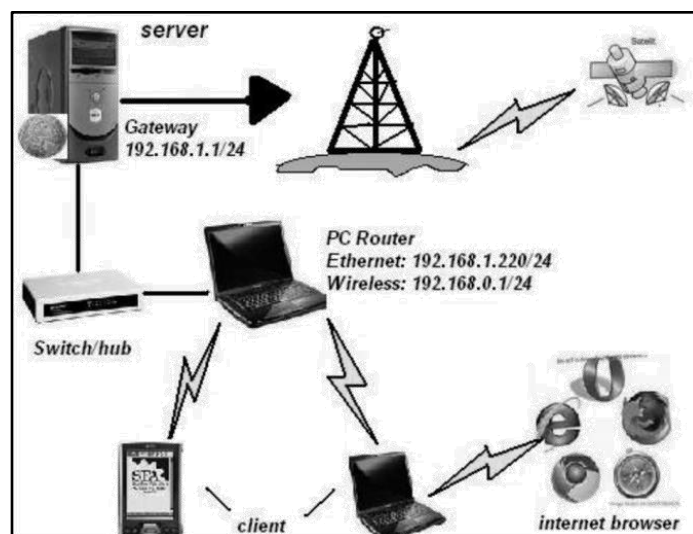
2.2.4.5 Bahasa Pemrograman, *Pascal, Fortran, Clipper, dBase.*

Software yang khusus digunakan untuk membuat program komputer, apakah itu sistem operasi, program paket. Bahasa pemrograman ini biasanya dibagi atas 3 tingkatan, yaitu *Low Level Language, Midle Level Language, High Level Language.*

2.2.5 Pengertian Internet

Menurut Anhar (2016). "Internet adalah jaringan atau sistem pada jaringan komputer yang saling berhubungan (terhubung) dengan menggunakan *Sistem Global Transmission Control Protocol/Internet Protocol Suite (TCP/IP)* sebagai protokol pertukaran paket (*packet switching communication protocol*) untuk melayani miliaran pengguna di seluruh dunia."

Menurut Subandi dan Syahidi (2018:16-17) menyatakan bahwa Internet merupakan singkatan dari *International Networking* (jaringan internasional), secara teknis merupakan dua komputer atau lebih yang saling berhubungan jaringan komputer hingga meliputi jutaan komputer di dunia yang saling berinteraksi dan bertukar informasi. Maka dapat disimpulkan dari kedua pengertian diatas, bahwa internet adalah jaringan global yang menghubungkan seluruh jaringan di dunia melalui jalur satelit.



Gambar 2.2 Skema cara kerja internet



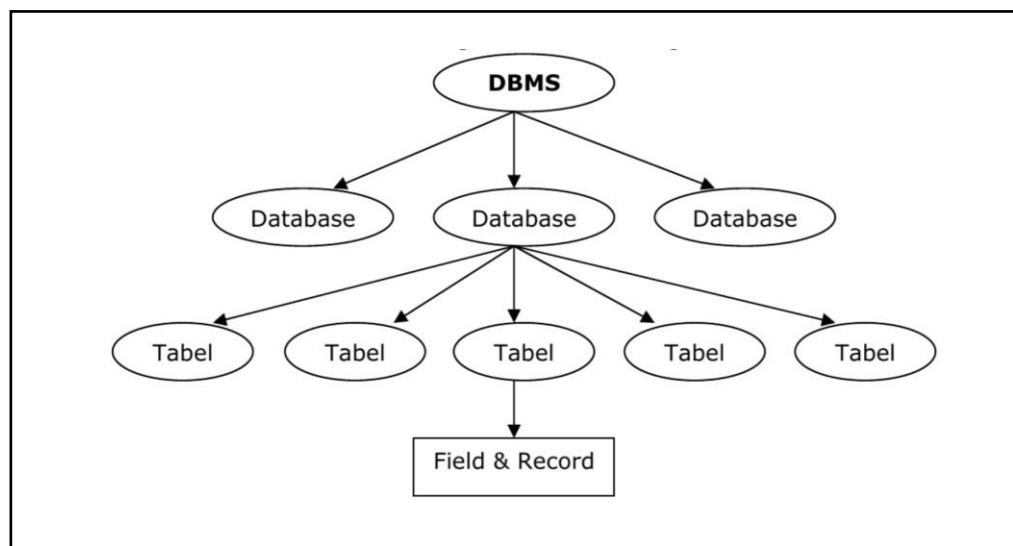
2.2.6 Pengertian Basis Data (*Database*)

Menurut Kristanto (2018:25) “Basis Data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut”.

Menurut Gadek (2017:06) Terpusat dan berbagi data user memasukan dan menyimpan semua data didalam satu komputer atau lebih, tindakan itu mengurangi penggunaan kertas, file, folder dan juga kemungkinan kehilangan atau kesalahan saat menempatkan data.

Berdasarkan pengertian diatas dapat disimpulkan bahwa basis data (*database*) adalah kumpulan data yang saling berhubungan, diorganisasi secara sistematis, menggunakan suatu program komputer agar dapat dimanipulasi dan diperoleh dengan cepat dan mudah.

Sekali data disimpan dalam komputer, banyak pengguna lain mengaksesnya via jaringan komputer (*publisher*). Struktur basis data dapat dilihat pada gambar dibawah ini :



Gambar 2.3 Struktur *Database*



2.3 Teori Khusus

2.3.1 Kamus Data

Menurut Jogiyanto dalam Purwati dan Aziz (2019:20), “Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi”.

Menurut Rusmawan (2019:36), “Kamus data merupakan katalog fakta data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data dibuat berdasarkan data yang ada pada diagram alur data”.

Berdasarkan pernyataan tersebut, dapat disimpulkan bahwa kamus data adalah katalog fakta dan kebutuhan-kebutuhan informasi dari suatu sistem informasi.

Simbol-simbol yang digunakan dalam kamus data sebagai berikut:

Tabel 2.1 Simbol-simbol Kamus Data

No.	Simbol	Uraian
1.	=	Terdiri atas, mendefinisikan, diuraikan menjadi, artinya
2.	+	Dan
3.	()	Optional (Boleh atau boleh tidak)
4.	{ }	Pengulangan



Lanjutan Tabel 2.1 Simbol-simbol Kamus Data

No.	Simbol	Uraian
5.	[]	Memilih salah satu dari jumlah alternative, seleksi.
6.	* *	Komentar
7.	@	Identifikasi atribut
8.	!	Pemisah sejumlah alternatif pilihan antara simbol []

(Sumber : Supardi, 2015:7-8)

2.3.2 Unified Modeling Language (UML)



Gambar 2.4 Logo UML

Menurut Mubarak (2019:20), “UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*)”.

Menurut Mulyani (2016: 42), “UML adalah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan



melakukan spesifikasi pada sistem”.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa *Unified Modeling Language* (UML) adalah teknik untuk pemodelan desain program dan aplikasinya.


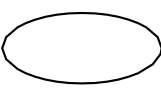
2.3.3 Use Case Diagram

Menurut Destiningrum dan Adrian (2017: 34), “ *Use Case Diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”.


Menurut Mulyani (2016: 42), “ *Use Case Diagram* yaitu diagram yang digunakan untuk menggambarkan hubungan antara sistem dengan aktor”.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa *Use Case Diagram* adalah pemodelan mengenai gambaran dari sistem informasi yang akan dibuat. Adapun simbol-simbol dari *Use Case Diagram*.

Tabel 2.2 Simbol-simbol *Use Case Diagram*

No.	Simbol	Nama	Keterangan
1.		Actor	Seseorang atau apa saja yang berhubungan dengan sistem yang sedang dibangun
2.		<i>Use case</i>	Menggambarkan bagaimana seseorang menggunakan sistem



3.		Relasi asosiasi	Relasi yang dipakai untuk menunjukkan hubungan antara aktor dan <i>use case</i>
----	---	-----------------	---

Lanjutan Tabel 2.2 Simbol-simbol *Use Case Diagram*

4.	<p data-bbox="443 1236 622 1265"><< include >></p>	Relasi <i>include</i>	Memungkinkan satu <i>use case</i> menggunakan fungsionalitas yang disediakan oleh <i>use case</i> lainnya
5.	<p data-bbox="443 1473 635 1503">.....></p> <p data-bbox="443 1514 622 1543"><< extends >></p> <p data-bbox="443 1704 635 1733">.....></p>	Relasi <i>extend</i>	Memungkinkan suatu <i>use case</i> secara optional menggunakan fungsionalitas yang disediakan oleh <i>use case</i> lainnya

(Sumber : Rusmawan, 2019:72-73)



2.3.4 *Class Diagram*

Menurut Zulkifli dan Farabi (2018:43), “*Class Diagram* adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem atau perangkat lunak yang sedang kita gunakan. *Class diagram* memberi kita gambaran (diagram statis) tentang sistem/perangkat lunak dan relasi-relasi yang ada didalamnya”.

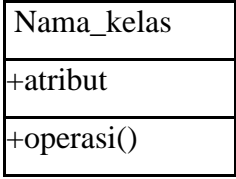
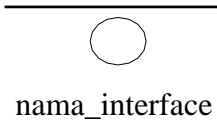

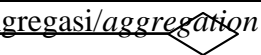
Menurut Saputra (2018:182), “*Class diagram* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek”.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa *Class Diagram* (Diagram Kelas) adalah diagram yang digunakan untuk menampilkan beberapa kelas yang ada dalam sistem atau perangkat lunak yang sedang kita gunakan.



Berikut adalah simbol-simbol yang digunakan pada *Class Diagram*:

Tabel 2.3 Simbol-simbol *Class Diagram*

No.	Simbol	Deskripsi
1.	<p>Kelas</p> 	Kelas pada struktur sistem
2.	<p>Antarmuka/<i>Interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	<p>Asosiasi/<i>association</i></p>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	<p>Asosiasi berarah/<i>directed association</i> →</p>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	<p>Generalisasi →</p>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	<p>Kebergantungan/<i>dependency</i></p> 	Relasi antar kelas dengan makna kebergantungan antarmuka
7.	<p>Agregasi/<i>aggregation</i></p> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

(Sumber : Mukhtar, 2019:85)



2.3.5 Activity Diagram



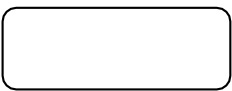
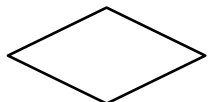

Menurut Zulkifli dan Farabi (2018:43), “*Activity Diagram* adalah diagram yang menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis”.

Menurut Saputra (2018:182), “*Activity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor, jadi aktivitas yang dapat dilakukan oleh sistem”.

Berdasarkan pengertian tersebut, dapat disimpulkan bahwa *Activity Diagram* adalah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem.

Adapun simbol-simbol yang sering digunakan dalam *Activity Diagram* dapat dilihat pada tabel berikut ini:

Tabel 2.4 Simbol-simbol *Activity Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Start state</i>	Titik Awal atau permulaan
2.		<i>End state</i>	Titik akhir atau akhir dari aktivitas
3.		<i>Activity</i>	<i>Activity</i> atau aktivitas yang dilakukan oleh actor
4.		<i>Decision</i>	Pilihan untuk mengambil keputusan
5.		<i>Interaction</i>	Alur

(Sumber : Rusmawan, 2019:81)



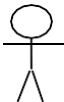
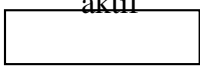

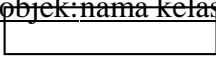

2.3.6 Sequence Diagram

Menurut Saputra (2018:182), “*Sequence diagram* adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu”.

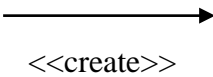
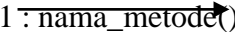
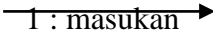
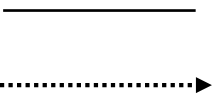
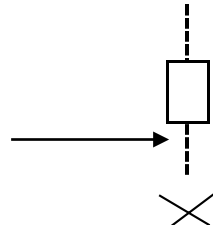
Berdasarkan pengertian tersebut, dapat disimpulkan bahwa *Sequence Diagram* adalah diagram yang menampilkan hubungan antar objek yang dijelaskan dengan rangkaian waktu.

Berikut adalah simbol-simbol yang digunakan pada *Sequence Diagram*:

Tabel 2.5 Simbol-simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>Atau</p> <p>Nama_Aktor</p> <p>Tanpa waktu aktif</p> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu berupa orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor
2.	<p>Garis hidup/<i>lifeline</i></p> 	Menyatakan kehidupan suatu objek
3.	<p>Objek <u>Nama</u></p> <p>objek:nama kelas</p> 	Menyatakan objek yang berinteraksi pesan
4.	<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, yang terhubung dengan waktu aktif adalah sebuah tahapan yang dilakukan didalam

Lanjutan Tabel 2.5 Simbol-simbol *Sequence Diagram*

No.	Simbol	Deskripsi
5.	Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.	Pesan tipe <i>call</i> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri
7.	Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8.	Pesan tipe <i>return</i> 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9.	Pesan tipe <i>destroy</i> <<destroy>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

(Sumber : Mukhtar, 2019:84-85)



2.4 Teori Program

2.4.1 Pengertian Java



Gambar 2.5 Logo Java

Haqi dan Setiawan (2019) juga menjelaskan bahwa, *Java* adalah sebuah nama untuk sekumpulan teknologi yang digunakan untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan.

Pemrograman java sebagai bahasa interpreter yang bersifat *Multi Platform*, tentu saja membutuhkan sebuah mesin virtual agar dapat berjalan pada sistem operasi yang disebut sebagai *java runtime environment* (JRE). Khusus untuk kategori pengembangan aplikasi, tidak hanya dibutuhkan sekedar sebuah virtual machine tapi juga antarmuka pemrograman aplikasi (API) yang dikemas dalam sebuah paket perangkat lunak pengembang yaitu *Java Development Kit* (JDK).

Teknologi Java memiliki tiga komponen penting, yaitu:

- *Programming-language specification*
- *Application-programming interface*
- *Virtual-machine specification*



Adapun karakteristik Java menurut Rickyanto (2020) adalah:

- a. Sederhana: Java tidak memiliki sintaks yang aneh tetapi banyak menggunakan sintaks c++ yang sudah banyak dikenal, sehingga java tidak menyulitkan bagi para programmer. Bahkan java memberikan banyak keunggulan dan kemudahan dibanding C++
- b. Berorientasi objek: Java merupakan pemrograman berorientasi objek yang murni. Dalam pemrograman Java semua adalah objek, terkecuali tipe primitif
- c. Dapat didistribusikan Dengan mudah: Sifat distribusi dari Java sangat tampak sebagai *applet* dan *library* yang mampu bekerja dalam jaringan dan bekerja dengan objek terdistribusi (RMI) dengan sangat baik.
- d. Aman: Program Java memiliki *library security* serta *policy* yang membatasi akses *applet* di komputer *client*.
- e. Diinterpretasi oleh interpreter: Java memerlukan *virtual machine* yang bertindak sebagai interpreter yang menterjemahkan bytecode (file class) menjadi bahasa mesin yang dimengerti oleh komputer *host*.
- f. Portabel: Java dapat dijalankan diberbagai *platform* tanpa perubahan kode.
- g. Multi *threading*: Java memiliki banyak kemampuan untuk menangani dan menjalankan banyak *thread* sekaligus.
- h. Dinamis: Java merupakan teknologi yang terus berkembang dan hal ini tampak nyata sekali dengan *library* yang terus ditingkatkan kemampuan dan kelengkapannya.
- i. Netral terhadap arsitektur *hardware*: Java dapat dijalankan dengan baik pada komputer yang memiliki arsitektur berbeda-beda.
- j. Robust: Java merupakan teknologi yang mampu menolong programmer untuk menghasilkan program secara cepat dan handal karena Java mencegah adanya memori *leaking*, meniadakan *pointer* serta mencegah berbagai eror yang mungkin terjadi dengan adanya proses pengecekan awal pada kompilasi.



Dibawah ini adalah timeline perjalanan Java dari awal hingga kini.

Tahun	Versi	Perkembangan Java
1991-1994	-	Pertama kali diciptakan dengan nama Oak.
1995	-	Nama Oak diganti menjadi Java.
1996	Java Kit 1.0	Sistem operasi Java diluncurkan.
1997-2000	Java Kit 1.1 Java Kit 1.2 Java Kit 1.3	Java Community Process (JCP) dibentuk untuk membuat mekanisme pengembangan dari pihak luar.
2001-2004	J2SE 1.4, J2SE 5.0	Website Java.com dirilis supaya pengguna bisa download dan menjalankan aplikasi Java di desktop dan laptop.
2005		10 tahun Java, Java telah digunakan oleh 4,5 juta developer dan 2,5 miliar perangkat.
2006-2008	Java SE 6	- Java tersedia dalam versi open source di bawah GNU (General Public License). - Pertama kalinya Java Virtual Machine (JVM) diluncurkan.
2009-2011	Java SE 7	Oracle mengakuisisi Sun. Oracle launching Java Magazine untuk publikasi teknikal bagi para developer.
2012-2014	Java SE 8	Publikasi Java 8 di terjemahkan dalam 8 bahasa dunia
2015		- Java menjadi platform bahasa pemrograman nomor satu di dunia. - Telah dijalankan di 13 miliar perangkat.
2016	-	Java Magazine telah di subscribe 250.000 orang dan dijalankan di lebih dari 15 miliar perangkat.
2017	Java SE 9	Ada 38 juta JVM yang aktif digunakan.
2018	Java SE 10 Java SE 11	Fitur terbaru termasuk HTTP client untuk website.
2019-2020	Java SE 12 – 15	Java telah digunakan berbagai aplikasi mobile, seperti Spotify, Siemens, NASA Open Code Project, Minecraft, dll



2.4.2 Pengertian Android Studio



Gambar 2.6 Logo Android Studio

Menurut Yeka, dkk (2020:117) Android Studio merupakan sebuah *Intregated Development Environment* (IDE) untuk platform Android. Android Studio ini diumumkan pada tanggal 16 Mei 2013 pada Konferensi Google I/O oleh Produk Manajer Google, Ellie Powers. Android Studio bersifat *free* dibawah *Apache License 2.0*. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna.

Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android. Untuk dapat digunakan dalam pembuatan aplikasi android maka android studio membutuhkan plug-in yang disebut ADT, ADT adalah kepanjangan dari android *development tools* yang menjadi penghubung antara IDE android studio dengan *Android Software Development Kit*.

A. Jenis Android Studio

Android Studio yang saat ini digunakan orang memiliki beberapa versi. Jenis-jenis Android Studio yang tersedia adalah sebagai berikut:



1. Android Studio versi Canary

Android Studio Canary dapat digunakan jika kita ingin mendapatkan jenis rilis apa pun. Jika menggunakan versi canary ini, programmer bisa menggunakan fitur baru saat baru diluncurkan tanpa harus menunggu lama. Biasanya update untuk versi canary ini dirilis setiap minggu.

2. Android Studio versi Dev

Versi canary yang telah teruji secara klinis dan untuk beberapa waktu akan dipilih dan menjadi versi Dev. Versi Dev digunakan untuk menguji fitur baru secepat mungkin kepada pengguna.

3. Android Studio versi Beta

Jika kamu ingin menggunakan Android Studio dengan fitur baru tetapi risiko bug kecil, gunakan versi beta. Versi ini dirilis biasanya sudah memiliki fitur baru tetapi dengan beberapa bug.

4. Android Studio versi Stable

Android Studio versi stabil adalah versi yang semua fiturnya telah diuji sepenuhnya melalui versi-versi di atas. Jadi versi ini memiliki stabilitas tinggi dan kinerja yang sangat baik.

B. Fitur di Android Studio

Selain sebagai editor kode dan fitur developer intelligence yang andal, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas dalam membuat aplikasi Android, seperti:

1. Sistem build berbasis Gradle yang fleksibel.
2. Emulator cepat dan kaya fitur.
3. Lingkungan terpadu dapat mengembangkan aplikasi untuk semua perangkat Android.
4. Terapkan perubahan untuk mendorong perubahan kode dan sumber daya ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi.
5. Template kode dan integrasi GitHub untuk membantu kita membangun fitur aplikasi umum dan mengimpor kode sampel.
6. Framework dan alat pengujian yang lengkap.



7. Alat Lint untuk merekam kinerja, kegunaan, kompatibilitas versi, dan masalah lainnya.
8. Dukungan C++ dan NDK.
9. Dukungan bawaan untuk Google Cloud Platform, yang memudahkan integrasi Google Cloud Messaging dan App Engine. Halaman ini memberikan pengenalan fitur dasar Android Studio.

2.4.3 Pengertian Basis Data (*Database*)



Gambar 2.7 Logo *Database*

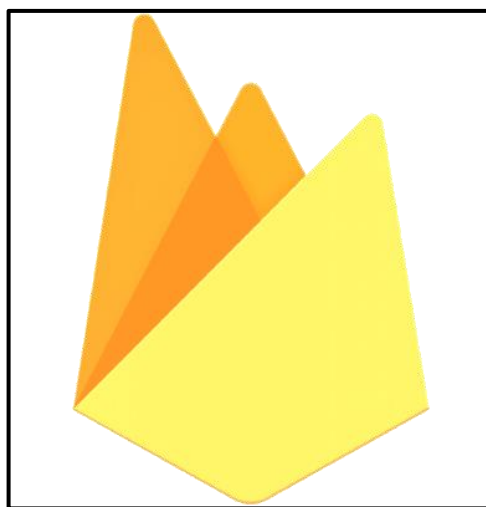
Menurut Kristanto (2018:79), “Basis Data adalah kumpulan data yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi”. Sedangkan menurut Nugroho (2017:175), “*Database*/Basis Data merupakan sekumpulan data yang sangat kompleks, kemudian data tersebut memiliki hubungan antara data yang satu dengan yang lainnya”.

Dari kedua pengertian diatas dapat disimpulkan bahwa, Basis Data adalah kumpulan data yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi, kemudian data tersebut memiliki hubungan antara



data yang satu dengan yang lainnya.

2.4.4 Pengertian *Firestore Realtime Database*



Gambar 2.8 Logo *Firestore*

Google firestore (2011). *Firestore Realtime Database* adalah database yang di-host di cloud. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung. Ketika Anda membuat aplikasi lintas-platform dengan SDK Android, iOS, dan *JavaScript*, semua klien akan berbagi sebuah instance Realtime Database dan menerima update data terbaru secara otomatis.

2.4.4.1 Fitur *Firestore*

Menurut Google Firestore (2016), setelah API Firestore dimasukkan ke Dalam aplikasi Android atau iOS, pengembang bisa menggunakan fitur Firestore dengan coding yang simpel. Beberapa fitur yang disediakan Firestore.

1. *Analytics*

Fitur ini membuat para pengembang aplikasi dapat mengerti bagaimana para pengguna menggunakan aplikasi mereka. SDK analytic



menangkap sendiri data yang dibutuhkan oleh para developer. Dashboard juga menyediakan detil seperti pengguna paling aktif atau fitur apa saja yang paling digunakan dalam aplikasi tersebut. Analytic juga menyediakan data yang telah dirangkum.

2. *Authentication*

Fitur ini membuat para pengembang dapat mengizinkan pengguna untuk mengakses aplikasi. Firebase menyediakan fitur login melalui Gmail, Github, Twitter, Facebook dan juga autentikasi buatan sendiri.

3. *Messaging FCM (Firebase Cloud Messaging)*

Membuat para pengguna dapat mengirimkan pesan ke berbagai platform tanpa biaya tambahan. Messaging juga dapat digunakan untuk kebutuhan notifikasi.

4. *Realtime Database*

Database di Firebase adalah database berbasis cloud dan tidak membutuhkan SQL untuk mengambil dan menyimpan data atau bisa disebut juga NoSQL. Database ini sangatlah cepat dan dapat diandalkan yang artinya data dapat dibaharui dan disinkronisasikan dengan cepat. Data juga dijaga meskipun pengguna kehilangan koneksi internetnya.

5. *Storage*

Firebase juga menyediakan fasilitas penyimpanan. Firebase dapat menyimpan dan mengambil konten seperti gambar, video, dan audio langsung dari SDK. Meng-upload dan men-download juga dilakukan di background. Data yang disimpan akan aman dan hanya pengguna yang diijinkan yang dapat mengaksesnya.

6. *Hosting*

Firebase juga menyediakan fitur hosting. *Firebase* mengirimkan konten web secara cepat dan konten selalu dikirim dengan aman.

7. *Crash reporting*

Fitur *crash reporting* di *Firebase* membuat pengembang dapat mengetahui kesalahan ketika terjadi crash.





2.5 Referensi Jurnal

Untuk menambah pengetahuan informasi penulis mengenai tugas akhir ini, penulis telah mengumpulkan sumber literatur yang berupa jurnal ilmiah. Berikut ini adalah rangkuman (*resume*) dari 5 jurnal ilmiah, yaitu :

Tabel 2.6 *Resume* Referensi Jurnal

NO	JUDUL/PENULIS/TAHUN	MASALAH	TEORI	METODE	HASIL
1	Rancang Bangun Sistem Informasi Pengelolaan Arsip Data Kerjasama Tri Dharma Perguruan Tinggi Menggunakan Metode <i>Alphabetical Filing System</i> Penulis : Mohamad Rully, Petrus Sokibi, Rinaldi Adam p-ISSN : 2640-7363 e-ISSN : 2614-6606 Volume 6 Nomor 2, September 2020 1-44	Adanya aktivitas perkembangan teknologi sudah menyentuh dalam berbagai bidang, salah satunya dalam bidang pendidikan, dimana memerlukan sebuah sistem untuk pengolahan data agar tersimpan secara aman dan tersusun rapih	Sistem Informasi	<i>Alphabetical Filing System</i>	Dapat mempermudah Lembaga Penelitian Dan Pengabdian Masyarakat dalam melakukan pengarsipan data. Kearsipan elektronik menjadi media pengarsipan data yang aman dan efisien dalam penyimpanan data.



Lanjutan Tabel 2.6 Resume Referensi Jurnal

NO	JUDUL/PENULIS/TAHUN	MASALAH	TEORI	METODE	HASIL
2	Penerapan Data Mining Pengarsipan Dokumen Berdasarkan Kontrak Kerja Dengan Metode <i>Alphabetical Filing System</i> Penulis : Eferoni Ndruru, Samuel Sihaloho, Erbin Sitorus ISSN:26145154 Volume 4 No.1 Februari 202032, pp 177 - 182	Proses ini sangat penting karena kumpulan dokumen yang disimpan secara teratur berencana mempunyai suatu kegunaan agar setiap kali diperlukan dapat cepat ditemukan kembali, oleh karena itu kearsipan surat harus disimpan secara teratur dan berencana serta di upayakan untuk seminimal mungkin terjadinya kesalahan.	Arsip, <i>Outsourcing</i> , <i>Alphabetic Filing System</i> , WEKA	Metode <i>Alphabetical Filing System</i> (AFS)	Berdasarkan hasil penelitian didapatkan Dalam pengarsipan dokumen berdasarkan kontrak kerja dengan metode <i>alphabetic filing system</i> menunjukkan nama korespondennya serta disusun berdasarkan abjad



Lanjutan Tabel 2.6 Resume Referensi Jurnal

NO	JUDUL/PENULIS/TAHUN	MASALAH	TEORI	METODE	HASIL
3	Penerapan Firebase Realtime Database Pada Prototype Aplikasi Pemesanan Makanan Berbasis Android Penulis : George Richard Payara1 , Radius Tanone p-ISSN : 2443-2210 e-ISSN : 2443-2229, Volume 4 Nomor 3 Desember 2018 pp 21-29	Pada proses pemesanan makanan di warung Lawas masih menggunakan cara biasa, terdapat dua cara manual dalam proses pemesanan makanan yaitu yang pertama pelanggan harus menghampiri pelayan untuk memesan makanan dengan cara mencatat semua pesanan makanannya pada nota pemesanan dan yang kedua adalah pelanggan harus menunggu pelayan datang ke meja makanan untuk memberikan nota pemesanan.	Penerapan Aplikasi Pemesanan Makanan	<i>Prototype</i>	Perancangan aplikasi pemesanan makanan dapat dibuat dengan menerapkan teknologi firebase realtime database berbasis android karena firebase memiliki banyak fitur seperti authentication, database, storage, hosting, pemberitahuan sehingga dapat mempermudah pelanggan dalam melakukan proses pemesanan makanan



Lanjutan Tabel 2.6 Resume Referensi
Jurnal

NO	JUDUL/PENULIS/TAHUN	MASALAH	TEORI	METODE	HASIL
4	Sistem Pengelolaan Dokumen Surat Menggunakan Metode <i>Alphabetical Filing</i> Dan <i>Chronology System</i> Penulis : Ria Resti, Langgeng Listiyoko, Meliana Saragih, Anggun Presilia	Diperlukan sistem pengendalian intern yang tepat dengan tujuan agar pemilik dapat mengontrol kegiatan operasional dalam perusahaan.	Sistem Pengelolaan dokumen surat	Metode <i>Alphabetical Filing</i> Dan <i>Chronology System</i>	Dengan adanya indexing dan penyimpanan secara digital. Efisiensi dapat diperoleh dalam hal data retrieval dan preview terhadap dokumen yang telah disimpan dalam bentuk digital. Hasil UAT menunjukkan 55% responden sangat setuju, 45% setuju, sementara tidak satupun yang tidak setuju dengan sistem yang diusulkan



Lanjutan Tabel 2.6 Resume Referensi Jurnal

NO	JUDUL/PENULIS/TAHUN	MASALAH	TEORI	METODE	HASIL
5	Sistem Informasi Pengelolaan Kerjasama Bidang Humas pada STMIK STIKOM Indonesia. Penulis : Welda, Brigida Arie Minartiningtyas Konferensi Nasional Sistem Informasi Volume 06, Nomor 02, September 2017	Pencatatan dan pengarsipan data-data kerjasama dan partner juga dilakukan secara tidak terstruktur, dimana setiap terjalin kerjasama maka Humas hanya akan fokus pada pembuatan MoU dan dalam waktu yang tidak ditentukan Humas akan mencatat kerjasama-kerjasama yang telah atau akan terjalin hanya pada file Microsoft Excel	Sistem Informasi	-	Sistem Pengelolaan Kerjasama Humas pada STMIK STIKOM Indonesia ini memiliki fitur pencatatan dan penyimpanan data Partner, Pencatatan dan penyimpanan data Kerjasama (PKS) serta pembuatan laporan data partner dan data kerjasama.