BAB II
LANDASAN TEORI

Untuk mendukung sebuah pelaksanaan penelitian landasan teori merupakan dasar teori yang akan membuktikan bahwa penelitian tersebut berkualitas, memiliki dasar pengetahuan yang dapat dipertangung jawabkan untuk melanjutkan penelitian sebelumnya, memperbaiki atau dapat juga mematahkan teori-teori sebelumnya.

2.1 *Android*

Menurut Silvia, Haritman dan Muladi (2014:2), Android adalah *platform open source* yang komprehensif dan dirancang untuk *mobile devices*. Dikatakan komprehensif karena Android menyediakan semua *tools* dan *frameworks* yang lengkap untuk pengembangan aplikasi pada suatu *mobile device*. Sistem Android menggunakan *database* untuk menyimpan informasi penting yang diperlukan agar tetap tersimpan meskipun *device* dimatikan.

Menurut DiMarzio (2017:2-3), Android adalah sistem operasi *mobile* yang didasarkan pada versi modifikasi dari Linux. Ini pada awalnya dikembangkan ole *startup* dengan nama yang sama, Android, Inc pada tahun 2005, sebagai bagian dari strategi untuk memasuki ruang *mobileI,* google membeli Android, Inc dan mengambil alih pekerjaan pembangunan (serta tim pengembang).

2.2 Perpustakaan

Menurut Noerhayati (1987:1), perpustakaan perguruan tinggi adalah suatu unit kerja yang merupakan bagian integral dari suatu lembaga induknya yang bersama-sama unit lainnya tetapi dalam peranan yang berbeda, bertugas membantu perguruan tinggi yang bersangkutan melaksanakan Tri Dharmanya.

Menurut Sutarno (2003:35) Perpustakaan Perguruan Tinggi merupakan perpustakaan yang berada dalam suatu perguruan tinggi dan yang sederajat yang berfungsi mencapai Tri Dharma Perguruan Tinggi, sedangkan penggunanya adalah seluruh civitas akademika.

2.3 Definisi Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya (*Hutahaean, 2014:9*).

2.4 Sistem Informasi

Menurut Abdul Kadir (2014:08) Dalam praktik, istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis komputer walaupun dalam kenyataannya komputer merupakan bagian yang penting. Ada beragam definisi sistem informasi sebagai berikut:

a. Menurut Alter (1992), Sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.
b. Menurut Bodnar dan Hopwood (1993), Sistem informasi adalah kumpulan perangkat lunak dan keras yang dirancang untuk mentransformasikan data kedalam bentuk informasi yang berguna.
c. Menurut Gelinas, Oram, dan Wiggins (1990), Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengolah data serta menyediakan informasi keluaran kepada pemakai.
d. Menurut Hall (2001), Sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokan, diproses menjadi informasi dan didistribusikan kepada pemakai.
e. Menurut Turban, McLean, dan Wetherbe (1999), Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik.
f. Menurut Wilkinson (1992), Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi), guna mencapai sasaran-sasaran perusahaan.

2.5 Basis data

Menurut Rosa A.S M. Shalahudin (2014:43) Basis Data adalah sistem terkomputerisasi yang tujuan utamanaya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah *file text* ataupun *Database Management System* (DBMS) Kebutuhan basis data dalam sistem informasi adalah untuk memasukkan, menyimpan dan mengambil data, selain itu untuk membuat laporan berdasarkan data yang telah disimpan. Tujuan dari dibuat nya tabel-tabel pada *database* adalah untuk menyimpan data kedalam tabel-tabel agar mudah diakses. Oleh karena itu, untuk merancang tabel-tabel yang akan dibuat maka dibutuhkan pola pikir penyimpanan data nantinya jika dalam bentuk baris-baris data *(record)* dimana setiap baris terdiri dari beberapa kolom.

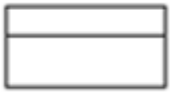
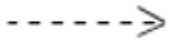
Menurut Abdul Kadir (2014:218) Basis data adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data dimaksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas.

2.6 Definisi *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa A.S M. *Shalahudin,* 2014).

Tabel 2.1. Simbo *Class Diagram*.

No	Gambar	Nama	Keterangan
1		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk

			(*ancestor*).
2		*N-Ary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribu tserta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
5		*Relatization*	Operasi yang benar-benar dilakukan oleh suatu objek
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.

2.7 Definisi*Object Oriented Programming* (OOP)

Object Oriented Programming (OOP) merupakan model pemrograman yang berbasis pada konsep obyek, diantaranya berisi data, sering dikenal sebagai atribut dan kode, dalam bentuk prosedur, sering dikenal sebagai metode. Sebuah fitur dari obyek adalah bahwa prosedur obyek dapat mengakses dan sering memodifikasi data dari obyek yang saling berhubungan. Dalam OOP, program dirancang dengan membuat obyek yang dapat berinteraksi satu sama lain (*Kindler, E.; Krivy, I.* (2011)). Pemrograman berorientasi obyek didefinisikan

menggunakan objek, tetapi tidak semua teknik terkait dan struktur dalam bahasa pemrograman mendukung OOP (*Michael Lee Scott, Programming language pragmatics, Edition 2, Morgan Kaufmann* (2006)). Terkadang obyek-obyek sesuai dengan hal-hal yang ditemukan di dunia nyata. Misalnya, program grafis mungkin memiliki objek seperti "lingkaran", "kotak", "menu". Sebuah sistem belanja *online* mungkin memiliki objek seperti "keranjang belanja", "pelanggan", dan "produk" (*Booch, Grady* (1986)). Kadang-kadang objek mewakili entitas yang lebih abstrak, seperti objek yang mewakili file terbuka, atau benda yang menyediakan layanan menerjemahkan pengukuran. Bahasa yang mendukung *class* hampir selalu mendukung *Inheritance*. Hal ini memungkinkan *Class* yang akan diatur dalam hirarki yang mewakili hubungan. Misalnya, *Class* karyawan mungkin mewarisi dari *Class* Person. Semua data dan metode yang tersedia untuk *Class* induk juga muncul di *Class* anak dengan nama yang sama. Misalnya, *Class* Orang mungkin mendefinisikan variabel *first_name* dan *last_name* dengan metode *make_full_name* (). Ini juga akan tersedia dalam *Class* karyawan, yang bisa menambahkan variabel posisi dan gaji. Teknik ini memungkinkan mudah digunakan kembali prosedur dan data yang sama definisi, selain berpotensi *mirroring* hubungan dunia nyata dengan cara yang intuitif. *Class-class* dan *subclass* sesuai dengan *set* dan *subset* di logika matematika. Dari pada menggunakan table *database* dan pemrograman subrutin, pengembang memanfaatkan objek, pengguna mungkin lebih akrab dengan Objek dari domain aplikasi merek(*Jacobsen, Ivar; Magnus Christerson; Patrik Jonsson; Gunnar Overgaard* (1992).

2.8 Definisi *Use Case* Diagram

Use case atau *use case*diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang

disebut aktor dan *use case*. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor (*Rosa dan Shalahuddin, 2013:155*).

Tabel 2.2. Simbol-Simbol *UseCase.*

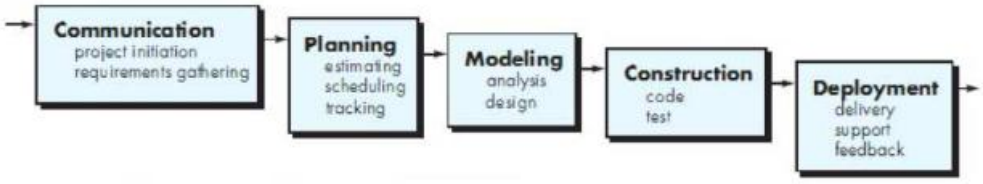
No	Gambar	Nama	Keteragan
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu element mandiri (*Iindependent*) akan mempengaruhi element yang bergantung padanya element yang tidak mandiri.
3		*Generlization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *Use Case* sumber secara *eksplisit.*
5		*Extend*	Menspesifikasikan bahwa *Use case* target memperluas perilaku dari *Use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menggabungkan antara objek satu dengan objek lainnya.

7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
9		*Collaboration*	Interaksi anturan-aturan dan elemen lain yang lebihi besar dari jumlah dan elemen-elemen (Sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2.9 Metode *Waterfall*

Metode *Waterfall* Menurut Pressman (2015:42), model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah "*Linear Sequential Model*". Model ini sering disebut juga dengan "*classic life cycle*" atau metode waterfall. Model ini termasuk ke dalam model generic pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh *Winston Royce* sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

Fase-fase dalam WaterfallModel menurut referensi Pressman :



Gambar 2.1 *Waterfall* Pressman (Pressman, 2015:42)

a. *Communication* (*Project Initiation & Requirements Gathering*) Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan customer demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi s*oftware*. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, dan internet.
b. *Planning* (*Estimating, Scheduling, Tracking*)Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan *tracking* proses pengerjaan sistem.
c. *Modeling* (*Analysis & Design*) Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektur *software*, tampilan *interface*, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.
d. *Construction* (*Code & Test*)Tahapan *Construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin. Setelah pengkodean selesai, dilakukan pengujian

terhadap sistem dan juga kode yang sudah dibuat. Tujuannya untuk menemukan kesalahan yang mungkin terjadi untuk nantinya diperbaiki.

e. *Deployment* (*Delivery, Support, Feedback*) Tahapan *Deployment* merupakan tahapan implementasi *software* ke customer, pemeliharaan *software* secara berkala, perbaikan *software*, evaluasi *software*, dan pengembangan softwareberdasarkan umpan balik yang diberikan agar sistem dapat tetap berjalan dan berkembang sesuai dengan fungsinya. (Pressman, 2015:17)

2.10 *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. Bahasa pemrograman utama yang digunakan adalah Java, sedangkan untuk membuat tampilan atau layout, digunakan bahasa XML. *Android studio* juga terintegrasi dengan *Android Software Development Kit* (SDK) untuk deploy ke perangkat *android*. *Android Studio* juga merupakan pengembangan dari *eclipse*, dikembangkan menjadi lebih kompleks dan professional yang telah tersedia didalamnya *Android Studio* IDE, *Android SDK tools*

Gambar 1.2 *Software Android Studio*

Sumber :

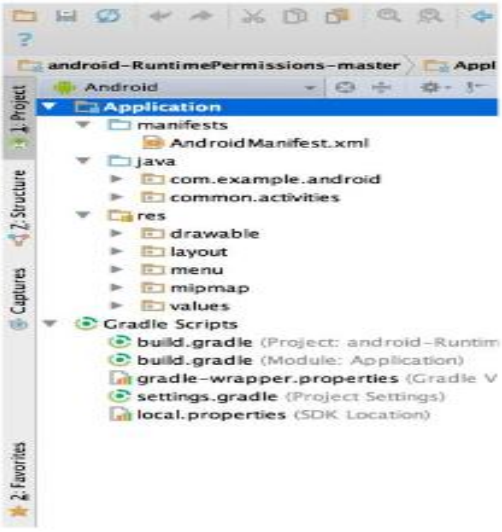
https://android-developers.googleblog.com/2014/12/android-studio-10.html

Setiap proyek di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis modul mencakup:

a. Modul aplikasi Androidb.
b. Modul Pustaka
c. Modul Google App Engine

Secara default, Android Studio akan menampilkan file proyek dalam tampilan proyek Android, seperti yang ditampilkan dalam gambar 1.2. Tampilan disusun berdasarkan modul untuk memberikan akses cepat ke file sumber utama proyek.Semua file versi terlihat di bagian atas di bawahGradle Scriptsdan masing-masing modul aplikasi berisi folder berikut:

a. manifests: Berisi fileAndroidManifest.xml.
b. java: Berisi file kode sumber Java, termasuk kode pengujian JUnit.
c. res: Berisi semua sumber daya bukan kode, seperti tata letak XML, string UI, dan gambar bitmap (Hidayah, 2017, http://eprints.polsri.ac.id/4490/3/File%203.pdf Jam 16:21 dan tanggal akses 24 November 2019).



Gambar 1.3 File Proyek di tampilan Android

Sumber :

https://developer.android.com/studio/intro/index.html?hl=id

2.11 Definisi *XAMPP*

Xampp adalah perangkat yang menggabungkan tiga aplikasi kedalam satu paket yaitu *Apache*, *MySQL*, dan *PHPMyAdmin*. Dengan *Xampp* pekerjaan anda sangat dimudahkan karena dapat menginstalasi dan mengkonfigurasi ketiga aplikasi tersebut dengan sekaligus dan otomatis.

Apache adalah sebuah *web server opensource*, jadi semua orang dapat menggunakannya secara gratis,bahkan anda bisa mengedit kode programnya. Fungsi utama dari *Apache* yakni menghasilkan halaman *web* yang benar sesuai dengan yang dibuat oleh seorang *web programmer*, dengan menggunakan kode *PHP*.

PHP adalah bahasa pemograman untuk membuat *web*, dengan *PHP* anda dapat membuat halaman *web* yang dinamis. Selain mendukung di sistem operasi *Windows*, *PHP* juga dapat di gunakan pada *mac* OS, *Linux*, dan sistemoperasi yang lainnya.

MySQL adalah sistem manajemen *database* yang sering digunakan bersama *PHP*. *PHP* juga mendukung pada *Microsoft Access*, *Database Oracle*, *d-Base*, dan sistem manajemen *database* lainnya. *SQL* (*Structured Query Language*) adalah bahasa terstruktur yang digunakan secara khusus untuk mengolah database. Dan *MySQL* merupakan sebuah sistem manajemen *database*.

Dengan aplikasi yang juga *open source* ini, anda dapat membuat dan mengolah *database* beserta isinya, menambahkan, mengubah, danmenghapus data yang berada dalam *database*. Diperlukan *MySQL*, dan *PHPMyAdmin*a dalah salah satu aplikasi yang anda bisa gunakan. Dengan *PHP MyAdmin* anda dapat membuat tabel, mengisi data,dan pekerjaan lainnya dengan mudah, tanpa harus mengafal perintahnya (*Affandi, Khaerul.* *http://khaerulaffandi.weebly.com/mengenal-apa-*

itu-xamppapachephp-dan-mysql.html. Jam 16.30 dan tanggalakses 24 November 2019).

2.12 UML (*Unified Modeling Language*)

"*Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangka tlunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem" (Windu dan Grace, 2013). *Unified Modeling Language* (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen - komponen yang diperlukan dalam sistem *software* (http://www.omg.org).Diagram *Unified Modelling Language* (UML) (Siti Fatima, 2015) antara lain sebagaiberikut:

2.1 *Use Case* Diagram,*Use case* menggambarkan *external view* dari sistem yang akan kita buat modelnya (Prabowo Pudjo Widodo, 2011) Model *use case* dapat dijabarkan dalam diagram *use case*, tetapi perlu di ingat, diagram tidak indetik dengan model karena model lebih luas dari diagram. (Pooley, 2003:15).*Use case* harus mampu mengambarkan urutan aktor yang menghasilkan nilai terukur (Prabowo Pudjo Widodo, 2011).

2.2 *Class* Diagram, Kelas sebagai suatu set objek yang memiliki atribut dan perilaku yang sama, kelas kadang disebut kelas objek (Whitten, 2004:410). Class memiliki tiga area pokok yaitu : 1) Nama, kelas harus mempunyai sebuah nama. 2) Atribut, adalah kelengkapan yang melekat pada kelas. Nilai dari suatu kelas hanya bisa diproses sebatas atribut yang dimiliki. 3) Operasi, adalah proses yang dapat dilakukan oleh sebuah kelas, baik pada kelas itu sendiri ataupun kepada kelas lainnya.

2.13 Kamus Data

(Rosa A.S M. Shalahudin,2014:73) kamus data (*data dictionary*) di pergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input)* dan keluaran (*output*) dapat dipahami secara umum. Kamus data biasanya berisi :

1. Nama, nama dari data.
2. Digunakan pada, merupakan proses-proses yang terkai data.
3. Deskripsi, merupakan deskripsi data.
4. Informasi tambahan, seperti tipe data, nial data, batas nilai data, dan komponen yang membentuk data.
5. Kamus data memiliki beberapa symbol untuk menjelaskan informasi tambahan sebagai berikut :

Table 2.5 Simbol-Simbol Kamus Data.

Simbol	Keterangan
=	Disusun atau terdiri dari
+	Dan
[|]	Baik...atau.....
{}n	N kali diulang atau bernilai banyak
( )	Data optional
....	Batas komentar

Kamus data pada DFD nanti harus dapat dipetakan dengan hasil perancangan basis data yang dilakukan sebelumnya. Jika ada kamus data yang tidak dapat dipetakan pada table hasil perancangan basis data dengan perancangan dengan DFD masih belum sesuai, sehingga harus ada yang diperbaiki baik perancangan basis datanya, peracangan DFD nya atau keduanya.