

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terdahulu**

Adapun penelitian terkait yang akan digunakan sebagai sumber acuan yang relevan dan terkini yaitu:

Penelitian dari Eka Wulansari (2018) dengan judul “Rancang Bangun Sistem Informasi *E-Learning* Berbasis *Web* Pada SMK Daarut Taufiq Tangerang”. Peneliti menghasilkan sebuah sistem *e-learning* yang membantu mengoptimalkan proses belajar-mengajar. Memudahkan para guru mendistribusikan materi secara *online* dan membantu siswa/i dalam memperoleh mata pelajaran lebih cepat dan akurat.

Penelitian yang dilakukan oleh Jimi Asmara (2020) Dengan judul “Perancangan Sistem *E-Learning* Berbasis *Web* Pada SMP N 2 Busalangga”. Penelitian yang dilakukan menghasilkan sebuah aplikasi *e-learning* pada SMP Negeri 2 Busalangga. Rancangan aplikasi ini akan mempermudah guru dan siswa/i dalam kegiatan belajar-mengajar, latihan soal, dapat mengatasi kendala kegiatan proses belajar-mengajar, dan implementasi *e-learning* di SMP Negeri 2 Busalangga. Guru juga dapat login ke sistem untuk mengunggah berita, materi pembelajaran, artikel, soal-soal latihan. Sedangkan siswa/i dapat melihat berita, artikel pengumuman, dan mengunduh materi pelajaran.

Penelitian yang dilakukan Putu Dian (2017) dengan judul “Perancangan Dan Implementasi *E Learning* Berbasis *Web* Pada Universitas Dhyana Pura Dengan Metode *Random Matrix*”. Hasil penelitian ini adalah sistem informasi berbasis *web* yang berisi beberapa fasilitas seperti kuis yang memungkinkan mahasiswa melaksanakan kuis secara *online*. Sistem pembelajaran *online* berbasis *web* ini dibangun untuk menunjang proses perkuliahan dengan menggunakan suatu sistem elektronik, sehingga mampu untuk mendukung suatu proses pembelajaran. Juga dapat memberikan fasilitas baru mengenai proses belajar-mengajar khususnya bagi mahasiswa pada Universitas Dhyana Pura.

Penelitian yang dilakukan Juniar Hutagalung (2019) dengan judul “Perancangan Dan Implementasi *E-Learning* Berbasis *Web* Pada SMA Negeri 1 Siantar”. Penelitian dapat mendukung proses kegiatan belajar-mengajar, sehingga melalui *e-learning* proses belajar-mengajar dapat dilakukan dengan cepat dan mudah serta dapat meningkatkan efisiensi dan efektifitas waktu juga biaya. Sistem *e-learning* berbasis *web* pada SMA Negeri 1 Siantar ini memberikan kemudahan pada siswa untuk melakukan pembelajaran.

Penelitian yang dilakukan oleh Romindo (2017) dengan judul “Perancangan Aplikasi *E-Learning* Berbasis *Web* Pada SMA Padamu Negeri Medan”. Penelitian ini menghasilkan aplikasi *e-learning* dalam memberikan fasilitas kuis *online* dengan memberikan soal-soal secara acak, sehingga proses belajar mengajar pun terjadi. Memberikan kemudahan dalam hal mengunduh materi-materi, artikel, dan pengerjaan kuis *online*.

## **II.2. Landasan Teori**

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa tinjauan pustaka pada penelitian ini yaitu:

### **II.2.1. Sistem**

Sistem adalah kumpulan elemen yang saling berhubungan dan berinteraksi dalam satu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama. Sistem dapat terdiri dari sistem-sistem bagian (*subsystem*). Sebagai misal, sistem komputer dapat terdiri dari subsistem perangkat keras dan subsistem perangkat lunak. Masing-masing subsistem dapat terdiri dari subsistem-subsistem yang lebih kecil atau terdiri dari komponen-komponen. Subsistem perangkat keras (*hardware*) dapat terdiri dari alat masukan, alat pemroses, alat keluaran, dan simpanan luar. Kemudian subsistem-subsistem tersebut akan berinteraksi sedemikian rupa sehingga dapat mencapai satu kesatuan yang terpadu.

Perkembangan ilmu pengetahuan dan teknologi saat ini, telah menciptakan beragam perangkat dengan fungsi dan manfaat yang membuatnya lebih mudah untuk semua orang, apalagi dengan banyak perangkat seperti komputer. Salah satu perangkat komputer itu adalah perangkat ponsel. Kualitas dan efektivitas di tempat kerja meningkat. Ponsel menawarkan kenyamanan melalui aplikasi untuk membantu manusia dalam menyelesaikan pekerjaan. Dan ponsel juga membantu aktivitas lain untuk berkomunikasi (Tanti : 2018) .

### II.2.1.1. Karakteristik Sistem

Sistem mempunyai karakteristik atau sifat-sifat tertentu, diantaranya adalah sebagai berikut:

1. Komponen Sistem (*Components*)

Sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environments*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga merugikan suatu sistem.

4. Penghubung Sistem (*Interface*)

Penghubung merupakan media penghubung antara suatu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari suatu subsistem ke subsistem lainnya.

5. Masukan Sistem (*Input*)

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran sistem (*Output*)

Keluaran adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran dapat berupa masukan untuk subsistem yang lain.

7. Pengolah Sistem (*Process*)

Sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolah. Pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objectives*)

Sistem mempunyai tujuan atau sasaran yang akan menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem (Firman et al : 2018).

## **II.2.2. Informasi**

Informasi dapat diibaratkan sebagai darah yang mengalir di dalam tubuh manusia, seperti halnya informasi di dalam sebuah perusahaan yang sangat penting untuk mendukung kelangsungan perkembangannya, sehingga terdapat alasan bahwa informasi sangat dibutuhkan bagi sebuah perusahaan. Akibat bila kurang mendapatkan informasi, dalam waktu tertentu perusahaan akan mengalami ketidakmampuan mengontrol sumber daya, sehingga dalam mengambil keputusan-keputusan strategis sangat terganggu yang pada akhirnya akan

mengalami kekalahan dalam bersaing dengan lingkungan pesaingnya. Disamping itu, sistem informasi yang dimiliki sering kali tidak dapat bekerja dengan baik. Masalah utamanya adalah bahwa sistem informasi tersebut terlalu banyak informasi yang tidak bermanfaat atau berarti sistem terlalu banyak data (Anwar : 2018).

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

1. Akurat (*Accurate*)

Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.

2. Tepat waktu (*Timelines*)

Berarti informasi yang datang pada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai logika karena informasi merupakan landasan dalam pengambilan keputusan.

3. Relevan (*Relevance*)

Berarti informasi tersebut bermanfaat bagi pemakainya (Deppi Linda : 2019).

### **II.2.3. Sistem Informasi**

Sistem informasi adalah sekumpulan komponen-komponen sistem yang berada dalam suatu ruang lingkup organisasi, saling berinteraksi untuk menghasilkan sebuah informasi yang bertujuan untuk pihak manajemen tertentu

dan untuk mencapai tujuan tertentu. Faktor-faktor yang menentukan kehandalan dari suatu sistem informasi dapat dikatakan baik jika memenuhi kriteria-kriteria sebagai berikut :

1. Keunggulan (*Usefulness*)

Yaitu suatu sistem yang harus dapat menghasilkan informasi yang tepat dan relevan untuk mengambil keputusan manajemen dan personil operasi dalam organisasi.

2. Ekonomis

Kemampuan sistem yang mempengaruhi sistem harus bernilai manfaat minimal, sebesar biayanya.

3. Kehandalan (*Reliability*)

Keluaran dari sistem harus mempunyai tingkat ketelitian tinggi dan sistem tersebut harus beroperasi secara efektif.

4. Pelayanan (*Customer Service*)

Yakni suatu sistem memberikan pelayanan yang baik dan efisien kepada para pengguna sistem pada saat berhubungan dengan organisasi.

5. Kapasitas (*Capacity*)

Setiap sistem harus mempunyai kapasitas yang memadai untuk menangani setiap periode sesuai yang dibutuhkan.

6. Sederhana dalam kemudahan (*Simplicity*)

Sistem tersebut lebih sederhana (umum) sehingga struktur dan operasinya dapat dengan mudah dimengerti.

7. Fleksibel (*Flexibility*)

Sistem informasi ini harus dapat digunakan dalam kondisi yang bagaimana diinginkan oleh organisasi tersebut atau pengguna tertentu.

#### 8. Komponen Sistem Informasi

Istilah dalam komponen sistem informasi adalah blok bangunan (*building block*) yang dapat di bagi menjadi enam blok yaitu:

##### a. Blok masukan (*Input block*)

Blok *input* merupakan data-data yang masuk ke dalam sistem informasi, yang dapat berupa dokumen-dokumen dasar yang dapat diolah menjadi suatu informasi tertentu.

##### b. Blok model (*Model block*)

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan mengolah data *input* untuk menghasilkan suatu informasi yang dibutuhkan.

##### c. Blok keluaran (*Output block*)

Merupakan informasi yang menghasilkan sekumpulan data yang nantinya akan disimpan berupa data cetak laporan.

##### d. Blok teknologi (*Technologi block*)

Blok teknologi merupakan penunjang utama dalam berlangsungnya sistem informasi. Blok teknologi memiliki beberapa komponen yaitu diantaranya alat memasukkan data (*input device*), alat untuk menyimpan dan mengakses data (*storege device*), alat untuk menghasilkan dan mengirimkan keluaran (*output divice*), dan alat untuk membentuk pengendalian sistem secara keseluruhan (*control device*).

Teknologi informasi terdiri dari 3 (tiga) bagian utama, yaitu teknisi (*humanware* atau *braiware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

e. Blok basis data (*Database block*)

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dan perlu di organisir sedemikian rupa, supaya informasi yang dihasilkan berkualitas.

f. Blok kendali (*Control block*)

Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat di cegah bila terlanjur terjadi (Adnyana : 2018).

#### **II.2.4. E-Learning**

*E-learning* adalah salah satu bentuk pendidikan yang menggabungkan motivasi, komunikasi, efisiensi dan teknologi. Sering kali proses pembelajaran mengalami keterbatasan waktu dan jarak tetapi tidak dengan e-learning yang bisa diakses di mana saja. Pada pembelajaran e-learning pelajar secara individu pada waktu tertentu dan bertanggungjawab untuk pelajarannya sendiri. E-learning sendiri menyediakan perangkat yang dapat memperkaya nilai belajar konvensional. Kapasitas siswa sangat bervariasi tergantung pada bentuk isi dan

penyampaiannya, semakin baik keselarasan antar konten dan alat penyampaian belajar maka akan memberi hasil yang baik (Hutagalung : 2019).

Media pembelajaran merupakan salah satu komponen pembelajaran, sehingga media tidak bisa luput dari pembahasan sistem pembelajaran secara menyeluruh. Pemanfaatan media seharusnya mendapatkan perhatian lebih dalam setiap kegiatan pembelajaran. Untuk membantu penyampaian pembelajaran dalam menunjang majunya dunia pendidikan saat ini. Namun kenyataannya penggunaan media pembelajaran masih belum maksimal karena belum banyak modul-modul media pembelajaran yang dibuat oleh tenaga pengajar (Yusfrizal : 2018).

#### **II.2.5. *Android***

*Android* adalah sistem operasi berbasis *linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai macam peranti bergerak. Awalnya, *Google Inc.* membeli *Android Inc.* pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi. Termasuk *Google, HTC, Intel, Motorola, Qualcomm, TMobile, dan Nvidia*. Pada saat perilisan perdana *android*, 5 November 2007, *android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, *Google* merilis kode *android* di bawah lisensi *Apache*,

sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler (Harni Kusniyati Dan Dicky : 2018).

*Android* merupakan sistem operasi *mobile*. *Android* tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. *Application Programming Interface* (API) yang disediakan menawarkan akses ke *hardware*, maupun data-data ponsel sekalipun, atau data sistem sendiri. *Android* merupakan sebuah sistem operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. Beberapa pengertian lain dari *Android*, yaitu :

1. Merupakan *platform* terbuka (*Open Source*) bagi para pengembang (*Programmer*) untuk membuat aplikasi.
2. Merupakan sistem operasi yang dibeli *Google Inc.* dari *Android Inc.*
3. Bukan bahasa pemrograman, tetapi hanya menyediakan lingkungan hidup atau *run time enviroment* yang disebut DVM (*Dalvik Virtual Machine*) yang telah dioptimasi untuk alat/*device* dengan sistem memori yang kecil (Ni Kadek Ceryna, dkk: 2018).

*Android* merupakan sistem operasi berbasis *linux* yang dirancang untuk perangkat bergerak layar sentuh (*mobile*) seperti *smartphone* dan *tablet*. *Android* adalah sistem operasi yang bersifat *open source*. *Google* merilis kodenya di bawah lisensi *apache*. *Android* awalnya dikembangkan oleh *Android Inc.* dengan dukungan finansial dari *google* dan kemudian *google* membelinya pada tahun 2005. *Android* dibuat menggunakan bahasa pemrograman *C(core)*, *C++*, *Java(UI)* dan dukungan berbagai bahasa pemrograman lain. Berdasarkan *dashboard developer android*, *Android 8.0/1 Oreo* adalah versi *android* yang

paling banyak digunakan, yakni sekitar 28.3% dari keseluruhan perangkat *android* di seluruh dunia. Hingga saat ini, *android* sistem operasi berbasis *mobile* yang paling banyak digunakan di seluruh dunia sudah di rilis versi 10 pada 3 September 2019 (Roberto Kaban, dkk: 2019).

*Android* merupakan sistem operasi berbasis *linux* yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. *Android* bersifat *open source* dan memungkinkan penggunanya untuk memasang aplikasi baik yang diperoleh dari toko aplikasi seperti *Google Play* atau dengan mengunduh dan memasang berkas apk. Apk adalah paket aplikasi *android* yang digunakan untuk menyimpan sebuah aplikasi atau program yang akan dijalankan pada perangkat *android* (Nelly Monica : 2018).

#### **II.2.6. Andorid Studio**

*Android studio* adalah *Integrated Development Enviroment* (IDE) untuk sistem operasi *android* yang dibangun berdasarkan perangkat lunak *JetBrains IntelliJ IDEA*. Selain editor kode dan alat pengembang yang didesain khusus untuk pengembangan *android*, *android studio* merupakan pengganti dari *Eclipse Android Development Tools* (ADT) yang sebelumnya merupakan IDE utama untuk pengembangan aplikasi *android*. *Android studio* menyediakan banyak fitur untuk meningkatkan produktivitas dalam pengembangan aplikasi berbasis *android* seperti tersedianya sistem pembangunan berbasis *gradle* yang *fleksibel*, emulator yang memiliki banyak fitur, integrasi dengan *GitHub*, dukungan bawaan untuk *Google Cloud Platform* sehingga mempermudah pengembang aplikasi untuk

mengintegrasikan *Google Cloud Messaging* dan *App Engine* (Roberto Kaban, dkk: 2019).

*Android Incorporation* didirikan di Palo Alto, California, Amerika Serikat pada bulan Oktober, 2003 oleh Andy Rubin: *co-founded Danger Inc. (Danger Inc.* adalah sebuah perusahaan eksklusif di *platform, software, desain dan layanan untuk perangkat komputasi mobile*), Rich Miner: *co-founded Wildfire Communications*, Nick Sears: *Vice President di T- Mobile*, dan Chris White: *desain kepala dan pengembangan antar muka di TV Web*. Dari mulai pendirian *android* dioperasikan diam-diam, hanya mengekspos bahwa itu bekerja pada perangkat lunak *mobile* (Nelly Monica : 2018).

### **II.2.7. Tools Pembangunan Android**

Untuk membangun sebuah sistem operasi *android* dapat menggunakan Mac, *Windows PC*, ataupun *Linux*. *Tools* yang dibutuhkan tersedia gratis dan dapat mengunduhnya melalui *web*. Berikut adalah beberapa *tools* yang digunakan untuk membangun aplikasi *android*.

1. JDK (*Java Development Kit*)
2. Android SDK
3. ADT (*Android Development Tools*) (Ni Kadek Ceryna, dkk : 2018).

### **II.2.8. Perkembangan Android**

Adapun perkembangan *android* adalah sebagai berikut:

- a. *Android versi 1.1*

*Android* memang diluncurkan pertama kali pada tahun 2007, namun sistem operasi ini mulai dirilis dan diterapkan ke berbagai *gadget* pada tanggal 9 Maret 2009 silam. *Android versi 1.1* merupakan *android* awal yang dimana versi ini baru memberikan sentuhan di beberapa aplikasinya seperti sistem antar muka bagi pengguna (*user interface*) yang lebih baik, serta beberapa aplikasi yang lain.

b. *Android versi 1.5 (Cupcake)*

Pada bulan Mei 2009 *android* kembali mengalami perubahan versi. *Android* versi 1.1 kemudian disempurnakan dengan *android versi 1.5* atau yang dikenal sebagai *Android Cupcake*.

c. *Android versi 1.6 (Donut)*

*Donut (versi 1.6)* diluncurkan dalam tempo kurang dari 4 bulan semenjak peluncuran perdana *Android Cupcake*, yaitu pada bulan September 2009.

d. *Android versi 2.0/2.1 (Eclair)*

Masih ditahun yang sama, *android* kembali merilis operating sistem versi terbarunya, yaitu *Android versi 2.0/2.1 Eclair*. *Android Eclair* diluncurkan oleh *Google* 3 bulan setelah peluncuran.

e. *Android versi 2.2 (Froyo: Frozen Yoghurt)*

Butuh 5 bulan bagi *Google* untuk melakukan regenerasi dari *Android Eclair* versi sebelumnya ke versi *Froyo Frozen Yoghurt*. Pada tanggal 20 Mei 2010, *Android versi 2.2* alias *Android Froyo* ini dirilis.

f. *Android versi 2.3 (Gingerbread)*

7 bulan kemudian *android* kembali melakukan gebrakan dengan merilis kembali

*Android versi 2.3* atau yang dikenal sebagai *Android Gingerbread*.

g. *Android versi 3.0/3.1 (Honeycomb)*

Pada bulan Mei 2011 *android versi 3.0/3.1* atau *Android Honeycom* dirilis.

*Android Honeycomb* merupakan sebuah sistem operasi *android* yang tujuannya memang dikhususkan bagi penggunaan tablet berbasis *android*.

h. *Android versi 4.0 (ICS: Ice Cream Sandwich)*

*Android ICS* atau *Ice Cream Sandwich* juga dirilis pada tahun yang sama dengan *Honeycomb*, yaitu pada bulan Oktober 2011.

i. *Android versi 4.1 (Jelly Bean)*

*Android Jelly Bean* merupakan versi *android* yang terbaru pada saat ini. Salah satu *gadget* yang menggunakan sistem operasi *Jelly Bean* adalah *Google Nexus 7* yang diprakarsai oleh *ASUS*, *vendor* asal Taiwan yang juga menjadi teman satu kampung halaman dengan *Acer*.

j. *Android versi 4.4 (Kit Kat)*

Kehadiran *android kitkat* merupakan peluncuran produk *OS* anyar yang diluncurkan pada 4 September 2013, sebelumnya banyak kabar beredar kalau *android* akan meluncurkan *OS* baru yang bernama *Android Key Lime Pie* namun setelah di analisa tidak sesuai dengan ejaan orang umum, sehingga namanya diganti dengan *OS Android KitKat* yang sebagian besar orang sudah familiar dengan itu.

k. *Android versi 5.0.2 (Lollipop)*

*Android Lollipop* merupakan keberadaan *OS android* yang memang saat ini sudah menjadi *trend* baru di industri *smartphone*, hal ini tak lepas dari keunikan dan kelebihan yang banyak di miliki dari *OS* tersebut. Kehadiran *android* versi ini amat di nanti oleh sekian banyak orang karna diharapkan sistem operasi *lollipop* ini bisa lebih baik dibandingkan versi-versi sebelumnya.

l. *Android versi 6.0 (Marshmallow)*

*Android 6.0 Marshmallow* adalah versi dari sistem operasi *mobile android*. Pertama kali diperkenalkan Mei 2015 di *Google I / O* di bawah kode nama *Android M*, secara resmi dirilis pada Oktober 2015. *Android Marshmallow* memperkenalkan model izin aplikasi di desain ulang sekarang ada hanya delapan kategori izin, dan aplikasi yang tidak lagi secara otomatis diberikan semua hak akses mereka ditentukan pada waktu instalasi.

m. *Android 7.0 (Nougat)*

Untuk saat ini, sistem operasi *android* ini masih digunakan beberapa *smartphone* yang baru dirilis belakangan ini. *Android 7.0 & 7.1 Nougat* pertama kali diperkenalkan pada Juni 2016 dengan menampilkan ikon robot *android* dengan batangan *nougat*. Sistem operasi *android 7.0 & 7.1 nougat* mengalami perubahan dari segi tampilan antarmuka. Selain itu ada juga fitur *splitscreen* untuk membagi tampilan layar untuk dua aplikasi sekaligus (Muhamad Zaelani : 2017).

n. *Android Oreo 8.0*

*Oreo 8.0* hadir pada 21 Agustus 2017. *User interface* pada *Android Oreo* lebih simpel agar memudahkan dalam mengakses aplikasi. Pembaruan pada versi *oreo* membawa beberapa fitur seperti fitur *autofill* yang memberikan kemudahan dalam mengisi formulir misal, dukungan gambar dalam gambar dan pengoptimalan *booting* agar lebih cepat.

o. *Android Pie 9.0*

*Pie 9.0* dirilis pada 6 Agustus 2018 dengan membawa *smart reply*, *digital wellbeing*, *adaptive brightness* yang akan menyesuaikan kecerahan layar secara otomatis dan dukungan pada ponsel *bazel less*.

p. *Android 10*

Versi *android* terbaru diberi nama *android 10* untuk memperingati bahwa *android* telah mencapai 1 dekade secara komersial. Versi *android 10* lebih berfokus pada penyempurnaan mode malam atau gelap serta peningkatan fitur *sound amplifier* untuk mengatur kualitas audio.

**Tabel. II.1 Versi Android**

Versi	Nama	Rillis	Catatan
1.0	<i>Android 1.0</i>	23 September 2008	<i>Android</i> pertama hanya untuk <i>smartphone</i>
1.1	<i>Android 1.1</i>	9 Februari 2008	
1.5	<i>Cupcake</i>	30 April 2009	Mulai pakai kode nama
1.6	<i>Donut</i>	15 September 2009	
2.0- 2.1	<i>Éclair</i>	26 Oktober 2009 (2.0) 12 Januari 2010 (2.1)	
2.2	<i>Froyo</i>	20 Mei 2010	

2.3	<i>Gingerbread</i>	6 Desember 2010	Digunakan pada <i>smartphone</i> jenis lama
3.0-3.2	<i>Honeycomb</i>	22 Februari 2011 (3.0) 10 Mei 2011 (3.1) 15 Juli 2011 (3.2)	Hanya untuk tablet
4.0	<i>ICS (Ice Cream Sandwich)</i>	19 Oktober 2011	<i>Smartphone</i> dan tablet
4.1-4.3	<i>Jelly Bean</i>	9 Juli 2012 (4.1) 13 November 2012 (4.2) 24 Juli 2013 (4.3)	<i>Update</i> untuk memperbaiki dan menambah fitur pada <i>ICS</i>
4.4	<i>Kit kat</i>	3 September 2013	
5.0	<i>Lollipop</i>	12 november 2014 (5.0) 9 Maret 2015 (5.1)	
6.0	<i>Marshmallow</i>	5 oktober 015	Terdapat <i>daze mode</i> , <i>Do Not Disturb mode</i> , mendukung <i>USB tipe C</i> , mendukung pembacaan <i>fingerprint</i> .
7.0	<i>Nougat</i>	22 Agustus 2016	Multitasking
8.0	<i>Android Oreo</i>	21 Agustus 2017	<i>multitasking</i> serta penampilan <i>UI</i> yang lebih rapi.
9.0	<i>Android Pie</i>	6 Agustus 2018	<i>mart reply</i> , <i>digital wellbeing</i> , <i>adaptive brightness</i>
10	<i>Android 10</i>	2020	

(Sumber: Harni Kusniyati Dan Dicky : 2018)

### II.2.9. Basis Data (Database)

Basis data merupakan kumpulan dari data yang saling berhubungan antara satu dengan yang lain. Basis data atau *database* merupakan salah satu komponen yang penting dalam sistem informasi, karena berfungsi sebagai basis penyedia informasi bagi pemakainya. Sistem basis data adalah suatu sistem menyusun dan mengelola *record-record* menggunakan komputer untuk menyimpan atau

merekam serta memelihara data operasional lengkap sebuah organisasi atau perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan (Sutopo : 2018).

### II.2.10. Normalisasi

Normalisasi merupakan sebuah teknik dalam *logical desain* sebuah basis data / *database*, teknik pengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik tanpa redundansi. Tujuan normalisasi adalah mengorganisasikan data ke dalam tabel-tabel untuk memenuhi kebutuhan pemakai, menghilangkan kerangkapan data, mengurangi kompleksitas, dan mempermudah modifikasi data. (Latief : 2018).

1. Proses Normalisasi
  - a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.
  - b. Apabila tabel yang diuji belum memenuhi persyaratan tertentu maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.
2. Tahapan Normalisasi
  - a. Bentuk Tidak Normal : Menghilangkan perulangan grup.

**Tabel II.2. Contoh bentuk tidak normal**

No-Mhs	Nama Mhs	Jurusan	Kode-MK	Nama-MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
			M1465	Analisis Perc. Sistem	B317	Dita	B
5432	Bakti	AK	M1350	Manajemen DV	B104	Ati	C

			Akn201	Akuntansi	D310	Lia	B
			MKT300	Dasar Pemasaran	B212	Lola	A

(Sumber : Mukhlisulfatih Latief : 2018)

- b. Bentuk Normal Pertama (1NF) : Menghilangkan ketergantungan sebagian. Suatu relasi dikatakan sudah memenuhi bentuk normal kesatu bila setiap data bersifat atomik yaitu setiap irisan baris dan kolom hanya mempunyai satu nilai data.

**Tabel II.3. Contoh Bentuk Normal Pertama (1NF)**

No-Mhs	Nama Mhs	Jurusan	Kode-MK	Nama-MK	Kode Dosen	Nama Dosen	Nilai
2683	Welli	MI	M1350	Manajemen DB	B104	Ati	A
2683	Welli	MI	M1465	Analisis Perc. Sistem	B317	Dita	B
5432	Bakti	AK	M1350	Manajemen DV	B104	Ati	C
5432	Bakti	AK	Akn201	Akuntansi	D310	Lia	B
5432	Bakti	AK	MKT300	Dasar Pemasaran	B212	Lola	A

(Sumber : Mukhlisulfatih Latief : 2018)

- c. Bentuk Normal Kedua (2NF) : Menghilangkan ketergantungan transitif. Suatu relasi dikatakan sudah memenuhi bentuk normal kedua bila relasi tersebut sudah memenuhi bentuk normal kesatu dan atribut yang bukan key sudah tergantung penuh terhadap key-nya.

**Tabel II.4. Contoh Bentuk Normal Kedua (2NF)**

Kode-MK	Nama-MK	Kode Dosen	Nama Dosen
M1350	Manajemen DB	B104	Ati
M1465	Analisis Perc. Sistem	B317	Dita
M1350	Manajemen DV	B104	Ati
Akn201	Akuntansi	D310	Lia
MKT300	Dasar Pemasaran	B212	Lola

(Sumber : Mukhlisulfatih Latief : 2018)

- d. Bentuk Normal Ketiga (3NF) : Menghilangkan anomali-anomali hasil dari ketergantungan fungsional. Suatu relasi dikatakan sudah memenuhi bentuk normal ketiga bila relasi tersebut sudah memenuhi bentuk normal kedua dan atribut yang bukan key tidak tergantung transitif terhadap key-nya.

**Tabel II.5. Contoh Tabel Mahasiswa Dan Tabel Kuliah (3NF)**

No_Mhs	Nama Mhs	Jurusan
2683	Welli	MI
5432	Bakti	AK

(Sumber : Mukhlisulfatih Latief : 2018)

### II.2.11. XAMPP

*XAMPP* adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari x (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. Untuk mendapatkannya dapat mengunduhnya langsung dari *web* resminya. (Palit : 2018).

### II.2.12. MySQL

*MySQL (My Structure Query Language)* merupakan sebuah program pembuat

*database* yang bersifat *open source*, artinya semua orang dapat menggunakannya dan dapat dijalankan pada semua *platform* baik *windows* maupun *linux*. *MySQL* juga merupakan sebuah perangkat lunak sistem manajemen basis data *sql* yang bersifat jaringan sehingga dapat digunakan untuk aplikasi *multi user*. *MySQL* juga sering dikenal dengan nama sistem manajemen *database* relasional. Suatu *database* relasional menyimpan data dalam tabel yang terpisah. Tabel-tabel tersebut terhubung oleh suatu relasi terdefinisi yang memungkinkan memperoleh kombinasi data dari beberapa tabel dalam suatu permintaan. Untuk administrasi *database*, pembuatan *database*, pembuatan tabel, dan sebagainya dapat digunakan aplikasi berbasis *web* seperti *PHP MyAdmin* dengan aplikasi *XAMPP*. (Saipul Anwar : 2018).

### **II.2.13. PHP**

*PHP* atau *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah *script HTML*. Bahasa *php* dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti *C*, *Java*, dan *Perl* serta mudah untuk dipelajari. *PHP* merupakan bahasa *scripting server – side*, dimana pemrosesan data dilakukan pada sisi *server*. Sederhananya, *server* yang akan menterjemahkan *script* program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Adapun pengertian lain *php* adalah *akronim* dari *hypertext preprocessor*, yaitu suatu bahasa pemrograman berbasis kode-kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode *HTML*". (Firman : 2018).

#### II.2.14. UML (*Unified Modelling Language*)

*Unified Modelling Language (UML)* merupakan suatu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. *UML* merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena *UML* menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. *UML* berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan.

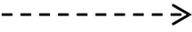
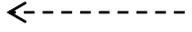
*Unified Modeling Language (UML)* biasa digunakan untuk:

- a. Menggambarkan batasan sistem dan fungsi-fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
- c. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.
- f. Menyampaikan atau memperluas *functionality* dengan *stereotypes*. (Alfina & Harahap : 2019).

### II.2.14.1. Use Case Diagram

*Use Case Diagram* adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.6 dibawah ini:

**Tabel II.6. Simbol Use Case**

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidkasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2018)

### II.2.14.2. Class Diagram

*Class Diagram* menggambarkan struktur dan deskripsi *class*, *package* dan *object* beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau *cardinality*. Dapat dilihat pada tabel II.7 dibawah ini :

**Tabel II.7. Multiplicity Class Diagram**

<b>Multiplicity</b>	<b>Penjelasan</b>
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

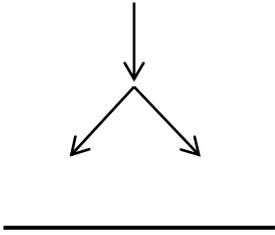
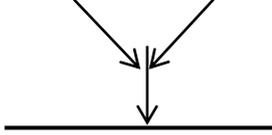
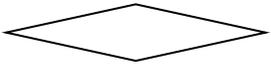
(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2018)

### II.2.14.3. Activity Diagram

*Activity Diagram* menggambarkan alur aktifitas dalam sistem yang sedang dirancang. Bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Sebuah aktifitas dapat direalisasikan oleh satu *use case* atau lebih. Aktifitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana *actor* menggunakan sistem untuk melakukan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses *parallel* (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis *horizontal* atau *vertical*. *Activity* diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktifitas tertentu.

Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.8 dibawah ini:

**Tabel II.8. Simbol Activity Diagram**

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

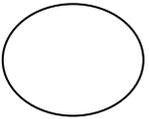
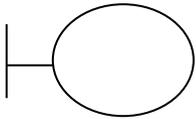
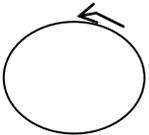
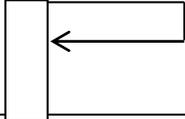
(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2018)

#### II.2.14.4. Sequence Diagram

*Sequence Diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* terdiri atas dimensi *vertical* (waktu) dan dimensi *horizontal* (obyek–obyek yang terkait). *Sequence Diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu.

Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.9 dibawah ini:

**Tabel II.9. Simbol Sequence Diagram**

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

	<p><i>Activation, activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2018)