

## BAB II LANDASAN TEORI

### 2.1 Definisi Model Pengembangan Perangkat Lunak

Model *Prototype* merupakan metode pengembangan perangkat lunak, yang berupa model fisik kerja sistem dan berfungsi sebagai versi awal dari sistem. Dengan metode *prototype* ini akan dihasilkan *prototype* sistem sebagai perantara pengembang dan pengguna agar dapat berinteraksi dalam proses kegiatan pengembangan sistem informasi. Agar proses pembuatan *prototype* ini berhasil dengan baik adalah dengan mendefinisikan aturan-aturan pada tahap awal, yaitu pengembang dan pengguna harus satu pemahaman bahwa *prototype* dibangun untuk mendefinisikan kebutuhan awal. *Prototype* akan dihilangkan atau ditambahkan pada bagiannya sehingga sesuai dengan perencanaan dan analisis yang dilakukan oleh pengembang sampai dengan ujicoba dilakukan secara simultan seiring dengan proses pengembangan[6]. Ada 4 metodologi *prototype* yang paling utama yaitu:

1. *Illustrative*, menghasilkan contoh laporan dan tampilan layar.
2. *Simulated*, mensimulasikan beberapa alur kerja sistem tetapi tidak menggunakan *data real*.
3. *Functional*, mensimulasikan beberapa alur sistem yang sebenarnya dan menggunakan *data real*.
4. *Evolutionary*, menghasilkan model yang menjadi bagian dari operasional sistem.

Dibuatnya sebuah *Prototype* bagi pengembang sistem bertujuan untuk mengumpulkan informasi dari pengguna sehingga pengguna dapat berinteraksi dengan model *prototype* yang dikembangkan, sebab *prototype* menggambarkan versi awal dari sistem untuk kelanjutan sistem sesungguhnya yang lebih besar. Ogedebe (2012), menegaskan: Telah ditemukan bahwa dalam analisis dan desain sistem, terutama untuk proses transaksi, di mana dialog yang ditampilkan lebih mudah difahami. Semakin besar interaksi antara komputer dan pengguna, besar

pula manfaat yang diperoleh ketika proses pengembangan sistem informasi akan lebih cepat dan membuat pengguna akan lebih interaktif dalam proses pengembangannya. *Prototype* dapat diterapkan pada pengembangan sistem kecil maupun besar dengan harapan agar proses pengembangan dapat berjalan dengan baik, tertata serta dapat selesai tepat waktu. Keterlibatan pengguna secara penuh ketika *prototype* terbentuk akan menguntungkan seluruh pihak yang terlibat, bagi pimpinan, pengguna sendiri serta pengembang sistem[6]. Manfaat lainnya dari penggunaan *prototype* adalah:

1. Mewujudkan sistem sesungguhnya dalam sebuah replika sistem yang akan berjalan, menampung masukan dari pengguna untuk kesempurnaan sistem.
2. Pengguna akan lebih siap menerima setiap perubahan sistem yang berkembang sesuai dengan berjalannya *prototype* sampai dengan hasil akhir pengembangan yang akan berjalan nantinya.
3. *Prototype* dapat ditambah maupun dikurangi sesuai berjalannya proses pengembangan. Kemajuan tahap demi tahap dapat diikuti langsung oleh pengguna.
4. Penghematan sumber daya dan waktu dalam menghasilkan produk yang lebih baik dan tepat guna bagi pengguna.

*Prototype* dimulai dengan pengumpulan kebutuhan, melibatkan pengembang dan pengguna sistem untuk menentukan tujuan, fungsi dan kebutuhan operasional sistem. Langkah-langkah dalam *prototype* adalah sebagai berikut:

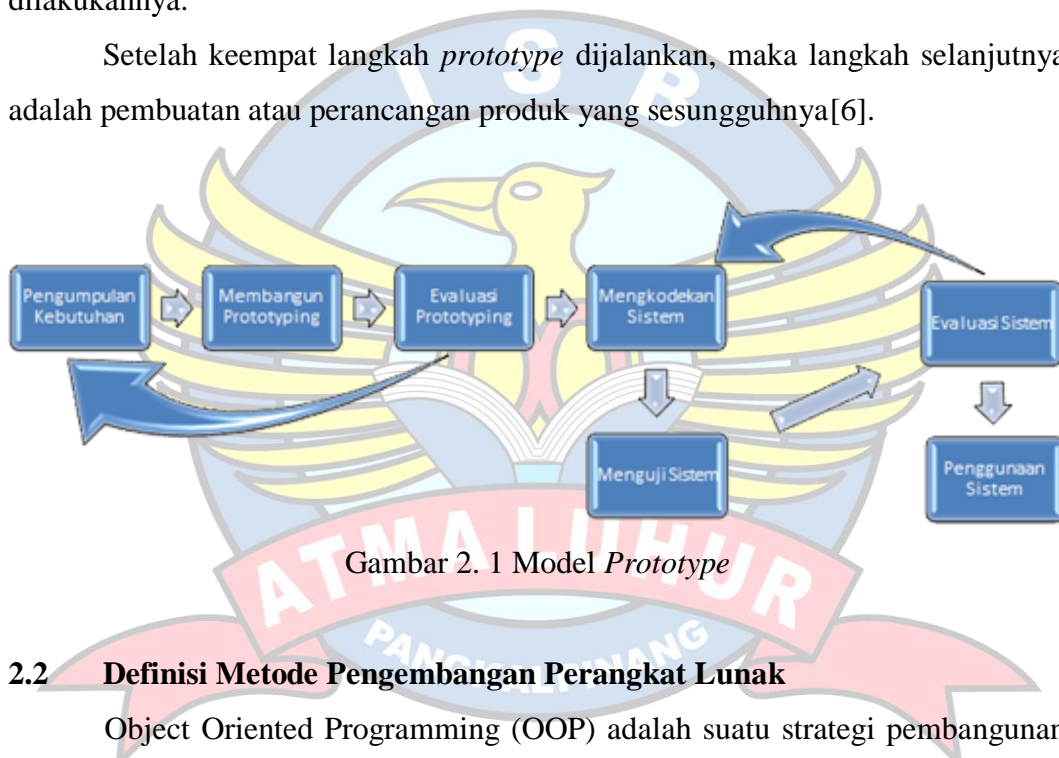
1. Pengumpulan Kebutuhan.
2. Proses desain yang cepat.
3. Membangun *prototype*.
4. Evaluasi dan perbaikan.

Mengumpulkan kebutuhan melibatkan pertemuan antara pengembang dan pelanggan untuk menentukan keseluruhan tujuan dibuatnya perangkat lunak

mengidentifikasi kebutuhan berupa garis besar kebutuhan dasar dari sistem yang akan dibuat.

Desain berfokus pada representasi dari aspek perangkat lunak dari sudut pengguna; ini mencakup input, proses dan format *output*. Desain cepat mengarah ke pembangunan prototipe, prototipe dievaluasi oleh pengguna dan bagian analisis desain dan digunakan untuk menyesuaikan kebutuhan perangkat lunak yang akan dikembangkan. *prototype* diatur untuk memenuhi kebutuhan pengguna, dan pada saat itu pula pengembang memahami secara lebih jelas dan detail apa yang perlu dilakukannya.

Setelah keempat langkah *prototype* dijalankan, maka langkah selanjutnya adalah pembuatan atau perancangan produk yang sesungguhnya[6].



Gambar 2. 1 Model *Prototype*

## 2.2 Definisi Metode Pengembangan Perangkat Lunak

Object Oriented Programming (OOP) adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. OOP ialah paradigma atau tehnik pemrograman yang berorientasi kepada objek. Berdasarkan pengertian yang ada dapat disimpulkan bahwa OOP merupakan suatu strategi atau cara baru untuk membuat program atau merancang sistem dengan memperhatikan objek.

Pada saat ini, metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya,

misalnya pada metode pendekatan terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan pada saat ini sangat beragam (aplikasi bisnis, real-time, utility, dan sebagainya) dengan platform yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut.

Pemrograman OOP merupakan pendekatan yang melihat sebuah permasalahan dan sistem pada objek. Memandang sistem berorientasi objek yang dikembangkan, sebagai kumpulan objek yang berkorespondensi dengan objek pada dunia nyata. Sistem yang berorientasi objek ini, memiliki fungsi serta data-data dikelompokkan dalam sebuah komponen yang dibungkus kedalam bentuk objek, sehingga setiap objek dapat mewariskan sifatnya atau setiap objek yang berbeda. Kumpulan dari objek-objek itu akan berinteraksi satu sama lainnya untuk menghasilkan output yang diinginkan[7].

### 2.3 Definisi *Tools* Pengembangan Perangkat Lunak

UML (*Unified Modeling Language*) adalah metode pemodelan secara visual sebagai sarana untuk merancang dan atau membuat software berorientasi objek. Karena UML ini merupakan bahasa visual untuk pemodelan bahasa berorientasi objek, maka semua elemen dan diagram berbasiskan pada paradigma *object oriented*.

UML adalah salah satu *tool* atau model untuk merancang pengembangan *software* yang berbasis *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*. UML sebagai sebuah bahasa yang memberikan *vocabulary* dan tatanan penulisan kata-kata dalam 'MS Word' untuk kegunaan komunikasi. Sebuah bahasa model adalah sebuah bahasa yang mempunyai *vocabulary* dan konsep tatanan / aturan penulisan serta secara fisik mempresentasikan dari sebuah sistem.

UML adalah sebuah bahasa standar untuk pengembangan sebuah *software* yang dapat menyampaikan bagaimana membuat dan membentuk model-model,



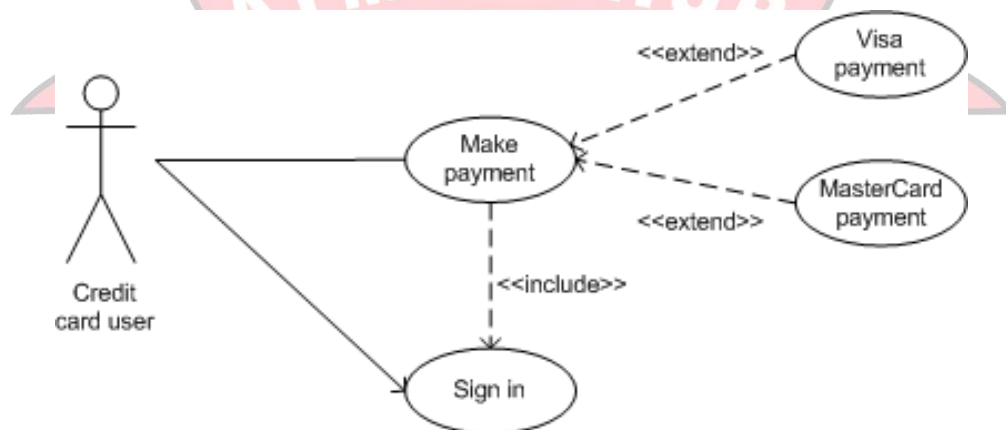
tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan *software*.

UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman, seperti Java, C++, Visual Basic, atau bahkan dihubungkan secara langsung ke dalam sebuah *object-oriented database*.

UML sendiri terdiri atas pengelompokan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model[8]. Dari macam-macam diagram UML diatas ada 4 diagram diantaranya *use case diagram*, *class diagram*, *activity diagram*, dan *sequence diagram* :

1. *Use Case Diagram*

*Use Case Diagram* menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. yang menjadi persoalan itu apa yang dilakukan bukan bagaimana melakukannya. *Use Case Diagram* dekat kaitannya dengan kejadian-kejadian. Kejadian (skenario) merupakan contoh apa yang terjadi ketika seseorang berinteraksi dengan sistem.



Gambar 2. 2 Contoh *Use Case Diagram*

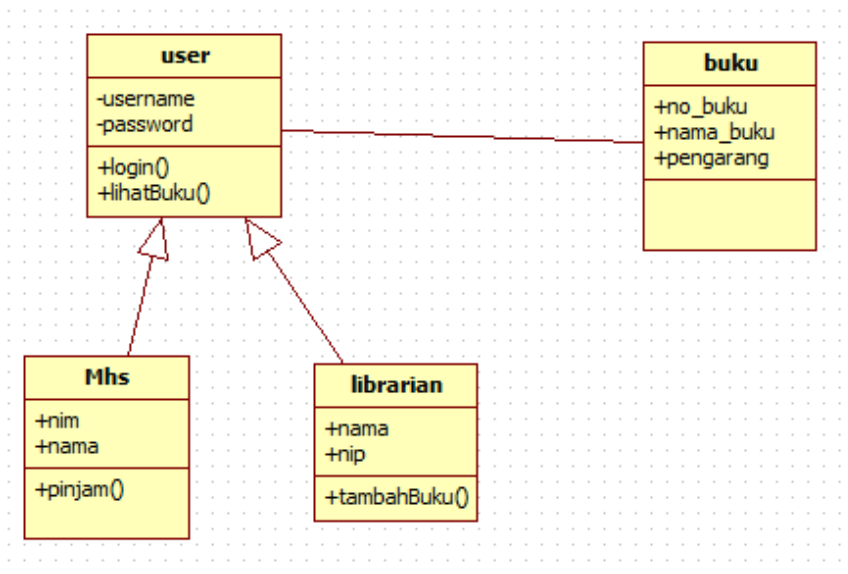
*Use Case Diagram* berguna dalam tiga hal:

- a. Menjelaskan fasilitas yang ada (*requirements*). *Use Case* baru selalu menghasilkan fasilitas baru ketika sistem di analisa, dan design menjadi lebih jelas.
- b. Komunikasi dengan klien. Penggunaan notasi dan simbol dalam *Use Case diagram* membuat pengembang lebih mudah berkomunikasi dengan klien-kliennya.
- c. Membuat *test* dari kasus-kasus secara umum. Kumpulan dari kejadian-kejadian untuk *Use Case* bisa dilakukan *test* kasus layak untuk kejadian-kejadian tersebut.

## 2. *Class Diagram*

*Class Diagram* memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan hubungan mereka. *Class Diagram* bersifat statis; menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan. *Class Diagram* mempunyai 3 macam *relationships* (hubungan), sebagai berikut:

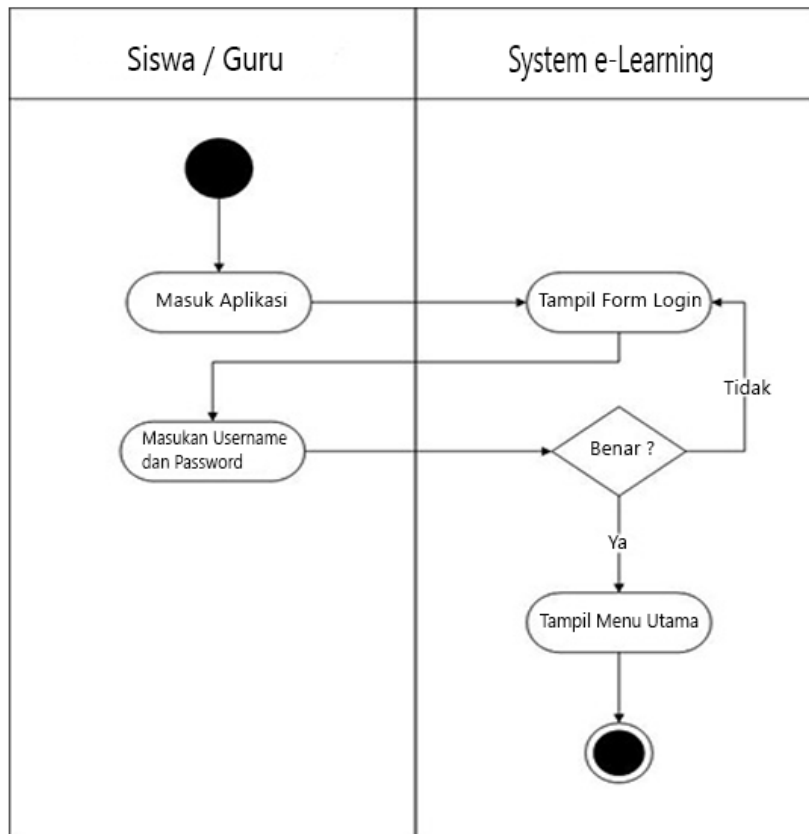
- a. *Association*  
Suatu hubungan antara bagian dari dua kelas. Terjadi *association* antara dua kelas jika salah satu bagian dari kelas mengetahui yang lainnya dalam melakukan suatu kegiatan. Didalam diagram, sebuah *association* adalah penghubung yang menghubungkan dua kelas.
- b. *Aggregation*  
Suatu *Aggregation* dimana salah satu kelasnya merupakan bagian dari suatu kumpulan. *Aggregation* memiliki titik pusat yang mencakup keseluruhan bagian. Sebagai contoh : *Order Detail* merupakan kumpulan dari *Order*.
- c. *Generalization*  
Suatu hubungan turunan dengan mengasumsikan satu kelas merupakan suatu *super class* (kelas super) dari kelas yang lain. *Generalization* memiliki tingkatan yang berpusat pada *super class*.



Gambar 2. 3 Contoh *Class Diagram*

3. *Activity Diagram*

Pada dasarnya *Activity Diagram* sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan *Statechart Diagram*. *Statechart Diagram* berfokus pada objek yang dalam suatu proses (atau proses menjadi suatu objek), *Activity Diagram* berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain. Contoh *Activity Diagram* sebagai berikut:

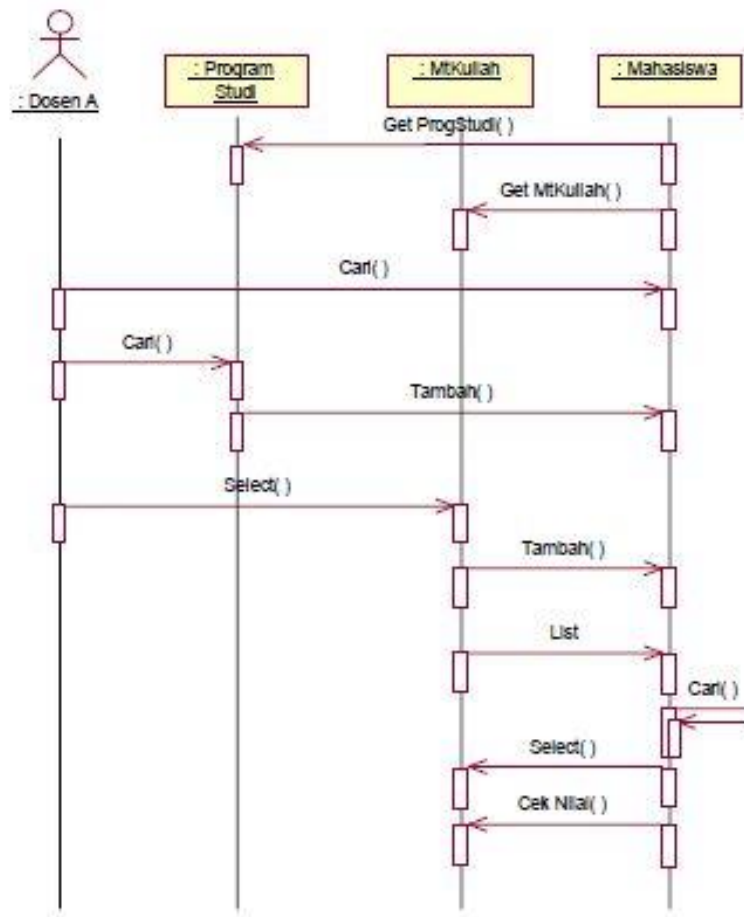


Gambar 2. 4 Contoh *Activity Diagram*

4. *Sequence Diagram*

*Sequence Diagram* merupakan salah satu diagram *Interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan; *message*(pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Objek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut. Contoh *Diagram sequence* sebagai berikut:





Gambar 2. 5 Contoh *Diagram Sequence*

## 2.4 Teori pendukung

Berikut ini teori pendukung yang digunakan sebagai referensi untuk mempermudah melakukan penelitian ini:

### 2.4.1 Aplikasi

Aplikasi yaitu perangkat lunak yang digunakan untuk membantu pemakai komputer untuk melaksanakan pekerjaannya. Jika ingin mengembangkan program aplikasi sendiri, maka untuk menulis program aplikasi tersebut, dibutuhkan suatu bahasa pemrograman, yaitu *language software*, yang dapat berbentuk *assembler*, *compiler* ataupun *interpreter*. Sekarang banyak sekali program-program aplikasi yang tersedia dalam bentuk paket-paket program. Ini adalah program-program

aplikasi yang sudah ditulis oleh orang lain atau perusahaan-perusahaan perangkat lunak. Beberapa perusahaan perangkat lunak telah memproduksi paket-paket perangkat lunak yang mempunyai reputasi internasional[6].

#### **2.4.2 Pengertian Penerimaan Siswa Baru**

Secara umum penerimaan siswa baru (PSB) dapat diartikan sebagai suatu proses administrasi yang terjadi setiap tahun untuk seleksi calon siswa berdasarkan nilai akademik agar dapat melanjutkan pendidikan pada jenjang yang lebih tinggi. Calon siswa yang dimaksud adalah siswa baru yang akan mendaftar pada jenjang SD, SMP atau SMA/SMK. Dapat dipahami bahwa penerimaan siswa baru di sini pada dasarnya hanya untuk memperlancar dan mempermudah dalam proses pendaftaran siswa siswi baru, pendataan dan pembagian kelas seorang siswa siswi, sehingga dapat terorganisir, teratur dengan cepat dan tepat dengan beberapa persyaratan yang telah ditentukan oleh sekolah[10].

#### **2.4.3 Pengertian Sekolah**

Sekolah adalah sebuah lembaga yang dirancang untuk pengajaran siswa atau murid di bawah pengawasan pendidik atau guru. Sebagian besar negara memiliki sistem pendidikan formal yang umumnya wajib dalam upaya menciptakan anak didik yang mengalami kemajuan setelah mengalami proses melalui pembelajaran. Menurut negara, nama-nama untuk sekolah-sekolah itu bervariasi, akan tetapi umumnya termasuk sekolah dasar untuk anak-anak muda dan sekolah menengah untuk remaja yang telah menyelesaikan pendidikan dasar[11].

#### **2.4.4 Pengertian Android**

Android adalah sistem operasi seluler yang didasarkan pada versi Linux yang dimodifikasi. Itu awalnya dikembangkan oleh startup dengan nama yang sama, *Android, Inc.* Pada tahun 2005, sebagai bagian dari strateginya untuk memasuki ruang seluler, Google membeli Android, Inc. dan mengambil alih pekerjaan pengembangannya (serta tim pengembangan).

Google ingin OS Android terbuka dan gratis, sehingga sebagian besar kode Android dirilis di bawah Lisensi Apache *open source*. Artinya, siapa pun yang ingin menggunakan Android dapat melakukannya mengunduh kode sumber Android lengkap. Selain itu, *vendor* (biasanya produsen perangkat keras) dapat menambahkan ekstensi miliknya sendiri ke Android dan menyesuaikan Android untuk membedakannya produk dari orang lain. Model pengembangan ini membuat Android sangat diminati oleh *vendor*, khususnya perusahaan-perusahaan yang terpengaruh oleh fenomena iPhone Apple, yang sangat sukses produk yang merevolusi industri *smartphone*. Saat iPhone diluncurkan, banyak *smartphone* produsen harus berebut mencari cara baru untuk merevitalisasi produk mereka. Produsen ini melihat Android sebagai solusi, artinya mereka akan terus mendesain perangkat keras dan menggunakan Android sebagai sistem operasi yang menjalankannya. Beberapa perusahaan yang telah memanfaatkan Kebijakan sumber terbuka Android termasuk Motorola dan Sony Ericsson, yang telah mengembangkannya memiliki sistem operasi seluler selama bertahun-tahun.

Keuntungan utama mengadopsi Android adalah ia menawarkan pendekatan terpadu untuk pengembangan aplikasi. Pengembang hanya perlu mengembangkan untuk Android secara umum, dan aplikasinya harus mampu berjalan di banyak perangkat berbeda, selama perangkat tersebut diberdayakan menggunakan Android. Di dalam dunia *smartphone*, aplikasi adalah bagian terpenting dari rantai kesuksesan[7].

#### 2.4.5 Android Studio

**Android Studio** adalah *Integrated Development Enviroment* (IDE) untuk sistem operasi Android, yang dibangun di atas perangkat lunak JetBrains IntelliJ IDEA dan didesain khusus untuk pengembangan Android. IDE ini merupakan pengganti dari *Eclipse Android Development Tools* (ADT) yang sebelumnya merupakan IDE utama untuk pengembangan aplikasi Android.

Android Studio sendiri pertama kali diumumkan di Google I/O *conference* pada tanggal 16 Mei 2013. Ini merupakan tahap *preview* dari versi 0.1 pada Mei

2013, dan memasuki tahap beta sejak versi 0.8 dan mulai diliris pada Juni 2014. [7]

Versi rilis stabil yang pertama diliris pada Desember 2014, dimulai sejak versi 1.0. Sedangkan versi stabil yang sekarang adalah versi 4.0 yang diliris pada Mei 2020. Fitur-fitur yang tersedia saat ini dalam *stable version*, sebagai berikut:

1. Dukungan Gradle-based build.
2. Android-specific *refactoring* dan perbaikan cepat.
3. Lint tools untuk menangkap kinerja, kegunaan, kompatibilitas versi, dan masalah lainnya.
4. Integrasi Proguard dan kemampuan penanda tangan aplikasi.
5. Template-based *wizards* untuk membuat *template design* umum seperti *drawer* atau *empty activity*.
6. Mendukung untuk pengembangan aplikasi Android Wear.
7. Editor tata letak yang memungkinkan pengguna untuk menyeret dan menjatuhkan (*drag-and-drop*) komponen UI, opsi untuk melihat tata letak pada beberapa konfigurasi layar.
8. Dukungan bawaan untuk Google Cloud Platform, memungkinkan integrasi dengan Firebase Cloud Messaging ('Perpesanan Google Cloud' Sebelumnya) dan Google App Engine.
9. Android Virtual Device (*Emulator*) untuk menjalankan dan *debug* aplikasi di Android Studio.

#### 2.4.6 Java

Java merupakan bahasa pemrograman yang berbasis objek, maksudnya adalah semua aspek yang terdapat pada Java adalah objek sehingga sangat memudahkan untuk mendesain, membuat dan mengembangkan program java dengan cepat. Kelebihan ini membuat program Java menjadi mudah untuk digunakan oleh banyak orang. Bahasa pemrograman ini juga dapat digunakan oleh banyak sistem operasi, seperti : Microsoft Windows, Linux, Mac OS, dan Sun Solaris (*Multiplatform* ) [8]. Kelebihan Java:



1. Berorientasi objek, Memudahkan untuk mendesign dan mengembangkan program dengan cepat dan teliti, sehingga mudah digunakan. Salah satu bahasa pemrograman yang berorientasi objek secara murni
2. Mirip C++ Mempunyai sintaks yang mirip dengan bahasa pemrograman C++, Sehingga para pengguna C++ banyak yang hijrah menggunakan Java.
3. *Multiplatform*, Dapat digunakan dibanyak sistem operasi.
4. Perkembangan yang luas, Dari game sampai sistem operasi *handphone* menggunakan program Java. Misalnya *Handphone* Sonny Ericsson dan Opera Mini 3 yang bertipe jar (Java Archive).
5. Mempunyai pengumpulan sampah otomatis.

#### 2.4.7 PHP

PHP adalah bahasa pemrograman *web*. Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat *web* yang bersifat *server-side scripting*. PHP memungkinkan kita untuk membuat halaman *web* yang bersifat dinamis. Sistem manajemen basis data yang sering digunakan bersama PHP adalah MySQL. namun PHP juga mendukung sistem manajemen *database* Oracle, Microsoft Access, Interbase, d-base, PostgreSQL, dan sebagainya[8].

#### 2.4.8 MySQL

MySQL adalah sistem manajemen *database SQL* yang bersifat *Open Source* dan paling populer saat ini. Sistem *Database* MySQL mendukung beberapa fitur seperti *multithreaded*, *multi-user*, dan *SQL database* manajemen sistem (DBMS). Database ini dibuat untuk keperluan sistem database yang cepat, handal dan mudah digunakan. Ulf Micheal Widenius adalah penemu awal versi pertama MySQL yang kemudian pengembangan selanjutnya dilakukan oleh perusahaan MySQL AB. MySQL AB yang merupakan sebuah perusahaan komersial yang didirikan oleh para pengembang MySQL. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL



sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL(*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis[8].

#### **2.4.9 XAMPP**

XAMPP adalah program aplikasi pengembang yang berguna untuk pengembangan website berbasis PHP dan MySQL. Versi terbaru program ini adalah XAMPP 1.7.7, yang dirilis pada tanggal 20 September 2011. *Software* XAMPP dibuat dan dikembangkan oleh *Apache Friends*. Perangkat lunak komputer ini memiliki kelebihan untuk bisa berperan sebagai *server web* Apache untuk simulasi pengembangan *website*. *Tool* pengembangan *web* ini mendukung teknologi *web* populer seperti PHP, MySQL, dan Perl. Melalui program ini, *programmer web* dapat menguji aplikasi web yang dikembangkan dan mempresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet. XAMPP juga dilengkapi fitur manajemen *database* PHPMyAdmin seperti pada *server hosting* sungguhan, sehingga pengembang *web* dapat mengembangkan aplikasi *web* berbasis *database* secara mudah. Program XAMPP banyak diaplikasikan dan digunakan oleh kalangan pengguna komputer di bidang pemrograman *web*. XAMPP merupakan *software* gratis. XAMPP dapat dijalankan di sistem operasi Windows 2000/XP/Vista/7 dan sistem operasi lain. Untuk memasang versi terbaru program ini, Anda cukup menginstal *file installer* XAMPP ke komputer Windows[8].

#### **2.4.10 Black Box**

*Black Box* adalah pengujian yang dilakukan untuk mengamati hasil *input* dan *output* dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik[8]. Ada

beberapa teknik yang biasanya digunakan untuk menguji perangkat lunak. Berikut ini adalah teknik-tekniknya:

1. *All Pair testing*
2. *Boundary value analysis*
3. *Cause-effect graph*
4. *Equivalence partitioning*
5. *Fuzzing*
6. *Orthogonal array testing*
7. *State transition*

## 2.5 Penelitian Terdahulu

Tabel 2. 1 Tinjauan Penelitian Terdahulu

Peneliti	Judul	Basis	Hasil
Sarwindah (2018)	Sistem Pendaftaran Siswa Baru Pada SMP Negeri 1 Kelapa Berbasis Web	Web	Sistem Pendaftaran siswa baru berbasis web menggunakan PHP dan MySQL ini telah dikembangkan. Sistem ini mampu mengelola pelaksanaan penerimaan siswa baru, yaitu pendaftaran siswa baru, pencetakan kartu pendaftaran, perengkingan penerimaan siswa baru, dan dengan adanya sistem pendaftaran siswa baru ini pendaftaran siswa baru menjadi lebih mudah dan

			efisien.
Yuni Prasetyanin gtyas (2019)	Aplikasi Berbasis Android Penerimaan Siswa Baru pada Universitas PGRI Madiun	Android	Dengan adanya sistem informasi. pendaftaran berbasis android ini dapat mempermudah dan memperlancar proses administrasi pendaftaran ketika Universitas PGRI Madiun membuka pendaftaran mahasiswa baru. Dan dapat menghemat waktu dan biaya para peserta didik baru. Peserta didik baru dapat dengan mudah mendapatkan segala informasi pendaftaran hanya dengan menggunakan smartphone.
Umi Salamah, Dedi Irawan (2017)	Analisis Perancangan Aplikasi Penerimaan Siswa Baru Pada SMA Negeri 1 Kalirejo dengan menggunakan	Android, Web	Bahwa aplikasi siswa baru pada SMA Negeri 1 Kalirejo dapat digunakan dalam pendaftaran siswa baru dengan aplikasi web mobile
Mohamad Randy Utomo, Muhammad Nur	Perancangan Sistem Informasi Pendaftaran Siswa Baru Berbasis Java Dekstop pada Madrasah Ibtidaiyah Al- Ihsan	Desktop	Perancangan sistem informasi pendaftaran siswa baru pada Madrasah Ibtidaiyah Al- Ihsan akan lebih terkomputerisasi dan

<p>Witama, Ria Asep Sumarni (2020)</p>			<p>membantu Staff Admin dalam melakukan penyimpanan data. Karena sistem ini dapat menghasilkan hasil penyimpanan pendataan pendaftaran siswa baru berupa laporan yang dapat dicetak. Dengan dibuatnya perancangan sistem informasi ini kinerja Staff Admin menjadi lebih baik, efektif dan bekerja secara maksimal</p>
<p>Siti Suherni, Nur Rubiati, Hayatullah Khumaini (2017)</p>	<p>Aplikasi Pendaftaran dan Penerimaan Siswa Baru di SMK Negeri 1 Rupert Berbasis Web dan SMS Gateway</p>	<p>Web, SMS Gateway</p>	<ol style="list-style-type: none"> <li>1. Dengan dibuatnya sistem ini dapat mempermudah pihak sekolah dalam pemberitahuan informasi pendaftaran dan penerimaan siswa baru kepada orang tua/wali siswa.</li> <li>2. Dengan adanya sistem ini siswa tidak perlu lagi datang ke sekolah SMK Negeri 1 Rupert untuk mendaftarkan diri ke sekolah tersebut karena sudah ada sistem online.</li> </ol>