

BAB II

LANDASAN TEORI

2.1 Tinjauan Teori

Banyak penelitian yang sebelumnya dilakukan mengenai sistem informasi pengelolaan data. Sistem informasi pengelolaan data membahas tentang pengumpulan data, pencatatan, pencarian dan laporan, sehingga sesuai dengan bidang dalam penelitian ini. Dalam upaya penerapan sistem informasi Pergerakan Mahasiswa Islam Indonesia (PMII) perlu dilakukan studi pustaka sebagai salah satu dari penerapan metode penelitian yang dilakukan. Diantaranya adalah mengidentifikasi kesenjangan (*identify gaps*), menghindari pembuatan ulang (*reinventing the wheel*), mengidentifikasi metode yang pernah dilakukan, meneruskan penelitian sebelumnya, serta mengetahui orang lain yang spesialisasi dan area penelitiannya sama dibidang ini. Beberapa penelitian sebelumnya tersebut adalah sebagai berikut :

Penelitian yang dilakukan oleh Ade Suryanto dan Ramadon pada tahun 2018 yang berjudul “Perancangan Sistem Informasi Pendaftaran Ekstrakurikuler Berbasis Web pada SMK Malaka Jakarta”[4]. Penelitian ini membahas tentang pendaftaran kegiatan ekstrakurikuler masih tercatat secara manual yaitu dengan pencatatan data peserta pada sebuah buku induk yang disediakan oleh sekolah, kemudian di rekap kembali untuk membuat laporan pendaftaran. Sistem yang berjalan mempunyai banyak kekurangan diantaranya membutuhkan waktu yang lama dalam proses mengelola data siswa yang mengikuti pendaftaran maupun dalam proses pembuatan laporan. Penelitian ini bertujuan untuk menghasilkan sebuah sistem informasi pengelolaan pendaftaran ekstrakurikuler dan data siswa agar lebih cepat, tepat, dan akurat. Metode pengumpulan data dilakukan dengan cara observasi, studi kepustakaan dan wawancara. Hasil dari penelitian ini adalah mempermudah pihak sekolah SMK Malaka Jakarta dalam proses pengelolaan data pendaftaran ekstrakurikuler dan membantu dalam proses pendataan siswa.

Penelitian yang dilakukan oleh Dewa Putu Yudhi Ardiana, I Wayan Dharma Suryawan dan Eddy Hartono pada tahun 2019 yang berjudul “Sistem Informasi

Pengelolaan Administrasi Organisasi Kemahasiswaan di STIKOM Indonesia”[5]. Penelitian ini menjelaskan tentang pengelolaan organisasi mahasiswa (ormawa) di STMIK STIKOM Indonesia selama ini dilakukan secara manual. Pengelolaan secara manual ini mengakibatkan institusi mengalami kesulitan untuk mengetahui kondisi ormawa. Kondisi organisasi mahasiswa meliputi keaktifan anggota organisasi kemahasiswaan, jumlah kegiatan, partisipasi mahasiswa dalam kegiatan kemahasiswaan, dan kedisiplinan organisasi kemahasiswaan dalam hal administrasi. Ketidakmampuan institusi dalam mengetahui kondisi ormawa mengakibatkan proses pembinaan tidak dapat berjalan sebagaimana mestinya. Pada penelitian ini dilakukan dengan menggunakan metode *Simple Additive Weighting* (SAW). Hasil penelitian ini untuk mengembangkan sistem informasi pengelolaan administrasi kemahasiswaan. Sistem informasi yang dikembangkan dapat menghasilkan informasi keaktifan ormawa, kegiatan ormawa dan laporan kondisi ormawa dengan perhitungan SAW.

Penelitian yang dilakukan oleh Riska Audina pada tahun 2015 yang berjudul “Aplikasi Informasi Kegiatan Mahasiswa di Fakultas Ilmu Terapan Universitas Telkom Berbasis Android dan SMS Broadcast”[6]. Penelitian ini menjelaskan tentang kegiatan ormawa di Fakultas Ilmu Terapan ini sering mengadakan acara atau event yang menunjang kreativitas mahasiswa yang masing-masing memiliki kegiatan non akademik. Sistem yang berjalan masih dilakukan secara manual seperti pendaftaran kegiatan, penyebaran informasi dan proses pencatatan mahasiswa. Sulitnya untuk informasi yang akan diselenggarakan, menyebabkan mahasiswa harus mencari informasi dengan bertanya ke mahasiswa lain, sehingga dirasa kurang efisien. Dalam penelitian ini menggunakan metode pengerjaan *SDLC Waterfall* sampai pada tahap pengujian. Penelitian ini bertujuan untuk bertujuan untuk memfasilitasi dan memberikan informasi untuk menarik minat mahasiswa dalam kegiatan yang akan diadakan oleh ormawa kampus.

Dari tiga penelitian terdahulu yang ada, telah banyak penelitian tentang sistem informasi yang di buat menggunakan website maupun android. Dari penelitian diatas belum ada penelitian yang secara khusus membahas mengenai Penerapan Sistem Informasi Pergerakan Mahasiswa Islam Indonesia (PMII) berbasis

Android. Kekurangan dan kelebihan dari penelitian terdahulu akan dijadikan acuan dalam penelitian ini, sehingga diharapkan akan tercapai penerapan sistem informasi yang lebih baik. Sistem informasi yang direncanakan adalah sistem informasi yang terkomputerisasi, dapat memberikan informasi kepada seluruh pengguna, namun juga terdapat menu khusus bagi admin. Selain itu, sistem informasi yang dibuat dapat diakses dan dioperasikan secara mudah oleh pengguna melalui *smartphone*.

2.2 Tinjauan Pustaka

2.2.1 Pengertian Sistem

kata “sistem” berasal dari kata Latin “systema” dan kata Yunani “sustema”. Sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Jadi, secara umum sistem adalah suatu kesatuan yang terdiri dari komponen atau elemen-elemen yang dihubungkan untuk memudahkan aliran informasi[7].

sistem terdiri dari berbagai unsur yang saling melengkapi dalam mencapai tujuan dan sasaran. Unsur-unsur yang terdapat dalam sistem itulah yang disebut dengan subsistem. Subsistem-subsistem tersebut harus saling berhubungan dan berinteraksi melalui komunikasi yang relevan sehingga sistem dapat bekerja secara efektif dan efisien.

2.2.2 Pengertian Informasi

Informasi adalah data yang telah diolah sehingga lebih berarti dan berguna bagi penerima. Jadi, sumber dari informasi adalah data. Informasi juga sekumpulan fakta (data) yang telah diorganisasikan dengan cara tertentu sehingga memiliki arti bagi penerima. Informasi adalah hasil pengolahan data dalam bentuk yang lebih berguna dan lebih berarti, yang menggambarkan kejadian (event) yang nyata (fact), dan dapat digunakan untuk pengambilan keputusan[8].

Informasi merupakan Data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat bagi pengambilan keputusan saat ini atau saat mendatang.

2.2.3 Pengertian Sistem Informasi

Sistem informasi adalah seperangkat komponen yang saling terkait yang mengumpulkan, memproses, menyimpan atau mendistribusikan informasi untuk mendukung koordinasi, pengendalian dan tahap pengambilan keputusan pada suatu organisasi. Informasi adalah data yang berhubungan dengan keputusan[9].

Menurut Abdilah, 2016 sistem informasi terdiri dari tiga konsep dasar, yaitu: masukan (input), proses (processing) dan keluaran (output). Ketiga elemen dasar ini menghasilkan informasi yang dibutuhkan untuk pengambilan keputusan, pengendalian operasi, analisis permasalahan dan menciptakan produk atau jasa baru[9].



Gambar 2.1 Konsep Dasar Sistem Informasi
(Sumber : Jurnal Teknologi Terpadu, 2016)

2.2.4 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* yang bersifat *open source* bagi para pengembang untuk menciptakan sebuah aplikasi. Awalnya Google Inc. Membeli Android Inc yang membuat piranti lunak untuk ponsel. Kemudian untuk mengembangkan android dibentuklah *Open Handset Alliance*. Android sebagai *platform mobile* pertama yang lengkap, terbuka dan bebas maka dari itu android menjadi pesaing utama dari Apple pada sistem operasi[12].

Jenis-jenis versi Android :

1. Android Versi 1.0 (Astro)
2. Android Versi 1.1 (Bender)
3. Android Versi 1.5 (Cupcake)
4. Android Versi 1.6 (Donut)
5. Android Versi 2.0/2.1 (Eclair)

6. Android Versi 2.2 (Froyo)
7. Android Versi 2.3 (Gingerbread)
8. Android Versi 3.0/3.1 (Honeycomb)
9. Android Versi 4.0 (Ice Cream Sandwich)
10. Android Versi 4.1/4.3 (Jelly Bean)
11. Android Versi 4.4 (Kitkat)
12. Android Versi 5.0 / 5.1 (Lolipop)
13. Android Versi 6.0 (Marshmallow)
14. Android Versi 7.0 (Nougat)
15. Android Versi 8.0 (Oreo)

2.2.5 Bahasa Pemrograman Android

1. Kotlin

Kotlin adalah Bahasa pemrograman berbasis *Java Virtual Machine* (JVM) yang dikembangkan oleh JetBrains. Kotlin merupakan bahasa pemrograman yang pragmatis untuk android yang mengkombinasikan *object oriented* (OO) dan pemrograman fungsional. Kotlin juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dalam satu project dengan bahasa pemrograman Java. Bahasa pemrograman ini juga dapat digunakan untuk pengembangan aplikasi berbasis desktop, web dan bahkan untuk *backend*. Beberapa keuntungan yang mungkin akan didapatkan jika pengembangan aplikasi beralih menggunakan Kotlin untuk mengembangkan aplikasi diatas platform JVM adalah sebagai berikut :

1. Dapat mengatasi `NullPointerException` yang umumnya terdapat pada Java.
2. Penulisan kode lebih ringkas dan mudah dibaca dibandingkan kode yang ditulis dengan menggunakan bahasa Java.
3. Mudah dipelajari.
4. Dukungan IDE untuk mempermudah dalam pemrograman[10].

2. Java

Java adalah bahasa pemrograman yang populer, dikembangkan oleh Sun Microsystems. Salah satu penggunaan terbesar Java adalah dalam pembuatan

aplikasi native untuk android. Bahasa pemrograman ini bersifat multiplatform yakni bahasa ini dapat digunakan di berbagai platform, seperti desktop, android dan bahkan untuk sistem operasi Linux[10]. Beberapa ciri dari bahasa pemrograman ini adalah sebagai berikut:

1. *Object oriented language*
2. *Multithreading*
3. *Garbage collector support*
4. *Statically Typed*
5. *Multiplatform*

Dari penjelasan diatas, bahasa pemrograman Kotlin dan java memiliki kelebihan masing-masing yang dapat digunakan untuk mengembangkan sebuah aplikasi berbasis android. Adapun perbedaan dari bahasa pemrograman tersebut sebagai berikut :

Tabel 2.1 Perbedaan Kotlin dan Java

Java	Kotlin
Kelebihan	
Mudah dikembangkan	Bisa dipertukar dengan Java
Sifatnya Multy Platform	Kurva Pembelajaran yang mudah
Memiliki keudahan dalam menyusun suatu script	Dukungan utama Android Studio
Bahasa pemrograman yang berorientasi terhadap objek	<i>Source Code</i> yang ringkas
Kekurangan	
Penggunaan memory yang tinggi	Ukuran runtime tambahan
Mudah di dekompilasi	Kemudahan <i>Source Code</i> dibaca diawal saja
Java adalah bahasa pemrograman yang kompleks	Minim dukungan resmi dari google
Java memiliki kecepatan yang kurang dari C++	Komunitas yang lebih kecil dan sedikit pertolongan tersedia

2.2.6 MySQL

MySQL merupakan sebuah sistem yang dapat digunakan untuk menyimpan data (*database*). Tujuan MySQL yaitu untuk mempermudah dalam mengakses dan menyimpan data secara cepat. Perintis pertama kali MySQL dilakukan oleh seorang programmer database bernama Michael Widenius. Pemrograman MySQL dapat menggunakan SQL dalam mengakses *database* dan dapat menambah, mengubah maupun menghapus data pada tabel SQL. SQL (*Structured Query Language*) merupakan bahasa standar yang dapat digunakan untuk mengakses *database* yang bersifat relasional. MySQL memiliki beberapa keunggulan, antara lain:

- MySQL dapat digunakan dalam menyimpan data dengan ukuran besar
- MySQL mampu melakukan *Multi-Threading* (menerima *query* lebih dari satu)
- MySQL dapat digunakan oleh siapa saja tanpa harus membayar lisensi kepada pembuatnya
- MySQL dalam menggunakan *database* nya dapat dihubungkan ke internet sehingga dapat diakses dari mana saja.
- MySQL memiliki keamanan yang cukup baik

2.2.7 JSON

JSON atau *JavaScript Object Notation* adalah standart terbuka berbasis teks ringan yang dirancang untuk pertukaran data yang dapat dibaca manusia. Format JSON pada awalnya ditentukan oleh Douglas Crockford, dan dijelaskan dalam RFC 4627. Ekstensi nama file JSON adalah *.json*.

JSON sebenarnya hanya sebuah standart yang disepakati oleh dunia internasional untuk mengirimkan data. JSON dapat digambarkan sebagai array yang memiliki dua kriteria terdiri dari *key-value pairs*, dan *ordered list*. Dengan menggunakan JSON kita dapat melakukan proses penerimaan dan pengiriman data dengan fleksibel struktur data yang disimpan pada database ini berupa JSON tree. Tidak seperti SQL database yang terdapat table-table atau records. Saat data ditambahkan ke JSON tree, maka menjadi sebuah node di dalam struktur JSON yang ada dengan sebuah associated key. Associated key dapat berupa user id, semantic names, atau dapat dibuat secara otomatis dengan *push()*

2.2.8 UML (*Unified Modeling Language*)

UML merupakan kependekan dari *Unified Modeling Language* yang artinya adalah sekumpulan alat yang digunakan untuk membuat sebuah model dari suatu sistem berbasis objek yang dimaksudkan untuk menyediakan cara standar atau memvisualisasikan desain sistem[4].

Unified Modeling Language (UML) menggunakan konsep dasar *operation* dan juga *class*, maka akan lebih cocok untuk penulisan perangkat lunak dalam bahasa berorientasi objek contohnya Java, Visual Basic, maupun C++[4]. Pada UML versi 2 terdiri dari 3 kategori dan 13 jenis diagram, yaitu :

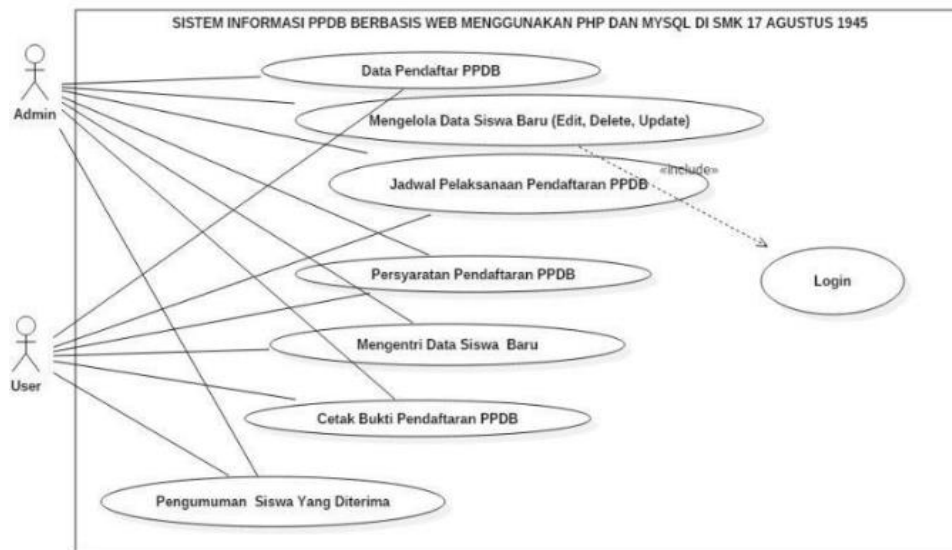
1. Struktur Diagram yang menggambarkan elemen pada spesifikasi dimulai dari obyek, kelas, hubungan mereka dan beralih ke dokumen arsitektur logis dari suatu sistem. *Structure* diagram dalam UML terdiri atas :
 - a) *Class Diagram*
 - b) *Object Diagram*
 - c) *Component Diagram*
 - d) *Deployment Diagram*
 - e) *Composite Structure Diagram*
 - f) *Package Diagram*
2. Behavior Diagram menggambarkan suatu ciri-ciri behavior/metode/fungsi dari suatu sistem atau business process. Behavior diagram dalam UML terdiri atas:
 - a) *Activity Diagram*
 - b) *Usecase Diagram*
 - c) *State Machine Diagram*
3. Interaction Diagram bagian behavior yang menggambarkan suatu interaksi objek. *Interaction* diagram dalam UML terdiri atas:
 - a) *Communication Diagram*
 - b) *Interaction Overview Diagram*
 - c) *Sequence Diagram*
 - d) *Timing Diagram*

2.2.9 Use Case Diagram

Use case adalah abstraksi dari interaksi antara system dan actor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana system akan terlihat di mata *user*. Sedangkan *use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client[13].

Tabel 2.2 Simbol *Use Case Diagram*

No.	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Pengguna sistem yang berhubungan dengan sistem lain atau waktu
2.		<i>Include</i>	Memungkinkan 1 <i>use case</i> menggunakan fungsionalitas yang disediakan oleh <i>use case</i> yang lain
3.		<i>Generalization</i>	Hubungan pada objek anak berbagi perilaku dan struktur data dari objek induk.
4.		<i>Extend</i>	Memungkinkan 1 <i>use case</i> secara optimal menggunakan fungsilitas dari <i>use case</i> yang lain
5.		<i>Use Case</i>	Deskripsi dari urutan aksi yang ditampilkan sistem.
6.		<i>Collaboration</i>	Interaksi elemen lain yang berkerja sama untuk menyediakan perilaku yang besar dari jumlah yang ada.
7.		<i>Association</i>	Menghubungkan antara objek satu ke objek yang lainnya

Gambar 2.2 Contoh *Use Case Diagram*





(Sumber : Jurnal Sistem Informasi Penerimaan Peserta Didik Baru Berbasis Android Menggunakan Metode Agile, 2018)

2.2.10 Class Diagram

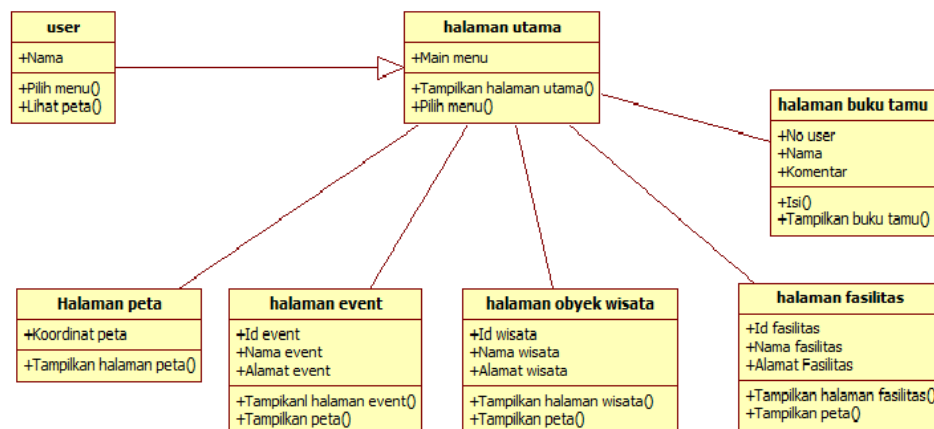
Class adalah deskripsi kelompok obyek-obyek dengan *property*, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *class diagram* dapat memberikan pandangan global atas sebuah *system*. Hal tersebut tercermin dari class-class yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class diagram*. *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu *system*[13].

Tabel 2.3 Simbol *Class Diagram*

No.	Simbol	Nama	Deskripsi
1.		<i>Generalitation</i>	Hubungan antara objek anak dari sebagai perilaku dengan struktur dari dari objek yang ada diatas objek induk.
2.		<i>Nary Assosiation</i>	Menggambarkan asosiasi percabangan jika pilihan aktivitas lebih dari dua objek.
3.		<i>Flow Control</i>	Gabungan dari beberapa objek dengan berbagi atribut serta operasi yang sama.

4.		<i>Collaboration</i>	Menggambarkan deskripsi berurutan yang dapat ditampilkan sistem dalam menghasilkan suatu hasil terstruktur bagi sebuah objek.
5.		<i>Dependency</i>	Menggambarkan suatu hubungan di mana terjadi perubahan pada suatu objek sehingga dapat mempengaruhi objek yang bergantung/tidak dapat berdirisendiri.
6.		<i>Realization</i>	Operasi yang benar-benar dilakukan pada sebuah objek.
7.		<i>Associattion</i>	Menggambarkan antara objek yang satu dengan objek yang lain.

ClassDiagram User








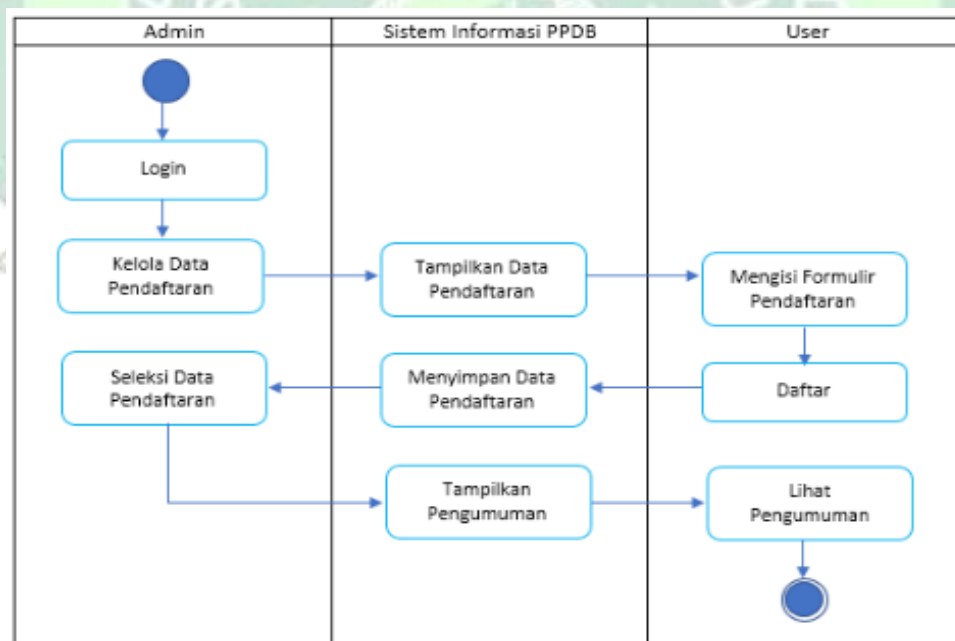
Gambar 2.3 Contoh *Class Diagram*
(Sumber : Jurnal Sistem Informasi, 2016)

2.2.11 Activity Diagram

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity* diagram mempunyai peranan seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah *activity* diagram bisa mendukung perilaku paralel sedangkan flowchart tidak bisa[13].

Tabel 2.4 Simbol *Activity Diagram*

No	Simbol	Nama	Deskripsi
1.		<i>Action State</i>	Menggambarkan keadaan elemen dalam suatu aliran aktifitas.
2.		<i>State</i>	Menggambarkan kondisi suatu elemen.
3.		Flow Control	Menggambarkan aliran aktifitas dari suatu elemen ke elemen lain.
4.		Initial State	Menggambarkan titik awal siklus hidup suatu elemen.
5.		Final State	Menggambarkan titik akhir yang menjadi kondisi akhir suatu elemen.







Gambar 2.4 Contoh *Activity Diagram*

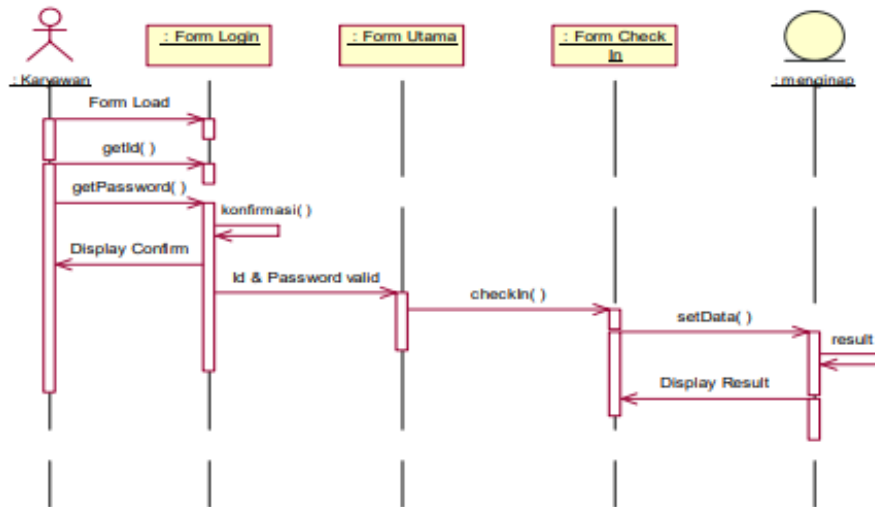
(Sumber : Jurnal Sistem Informasi Penerimaan Peserta Didik Baru Berbasis Android Menggunakan Metode Agile, 2018)

2.2.12 Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan pesan (*message*) yang diletakkan diantara objek – objek ini didalam *use case*. Komponen utama *sequence* diagram terdiri atas objek yang dituliskan dengan kotak segi empat bernama. Message diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progres vertical*[13].

Tabel 2.5 Simbol *Sequence Diagram*

No.	Simbol	Nama	Deskripsi
1.		Object	Menggambarkan pos-pos objek yang pengirim dan penerima message.
2.		Message	Menggunakan aliran pesan yang dikirim oleh pos-pos objek.
3.		Actor	Menggambarkan identitas / subjek yang dapat melakukan suatu proses.
4.		Lifeline	Menggambarkan tempat mulai dan berakhirnya suatu objek
5.		State	Menggambar kondisi suatu objek
6.		Antion State	Menggambar keadaan objek dalam melakukan aktivitas



Gambar 2.5 Contoh *Sequence Diagram*
(Sumber : Journal Speed – Sentra Penelitian Engineering dan Edukasi, 2016)

2.2.12 Jenis Pengujian Sistem

Pengujian sistem difungsikan sebagai menguji kelayakan suatu sistem dan memeriksa apakah hasil implementasi ada kesesuaian dengan analisis kebutuhan dan perancangan sistem yang dibuat. Pengujian bisa berupa *black box testing* (pengujian validasi) dan *white box testing* (pengujian unit).

a) Black box testing

Black Box Testing merupakan suatu pengujian yang difokuskan pada spesifikasi fungsional dari sebuah perangkat lunak, seorang *tester* dapat mendefinisikan sebuah kumpulan kondisi *input* (masukkan) dan melakukan pengujian atau pengecekan pada suatu spesifikasi fungsional sebuah program[14]. Secara garis besar *black box testing* menguji pada fungsional dari sistem tanpa memperhatikan struktur dan proses programnya.



Gambar 2.6 Black Box Testing

Sumber : Jurnal Cendekia, 2014

Pengujian *black box testing* sebagai berikut:

- Dilakukan oleh penguji Independent.
- Melakukan pengujian berdasarkan apa yang dilihat, hanya fokus terhadap fungsionalitas dan output. Pengujian lebih ditujukan pada desain software sesuai standar dan reaksi apabila terdapat celah-cela bug/vulnerabilitas pada program aplikasi tersebut setelah dilakukan white box testing.
- Dilakukan setelah white box testing.

b) White box testing

White Box Testing merupakan suatu pengujian pada pengecekan detail perancangan dan penggunaan struktur kontrol dari sebuah desain program secara prosedural untuk dapat membagi pengujian ke dalam beberapa kasus pengujian[14]. Tujuan dari pengujian *white box testing* untuk mengetahui bagaimana cara kerja perangkat lunak secara internal dan menjamin operasi-operasi internal sesuai dengan spesifikasi yang telah ditetapkan dengan menggunakan struktur kendali dari prosedur yang dirancang.



Gambar 2.7 White Box Testing

Sumber : Jurnal Cendekia, 2014

Pengujian *white box testing* sebagai berikut :

- Dilakukan oleh penguji yang mengetahui tentang *Quality Assurance* (QA).
- Melakukan testing pada software/program aplikasi menyangkut security dan performance program tersebut (meliputi tes code, desain implementasi, security, data flow, software failure).
- Dilakukan seiring dengan tahapan pengembangan software atau pada tahap testing.

2.3 Kerangka Pemikiran

Kerangka pemikiran merupakan tahapan-tahapan yang akan dilakukan dalam proses penyelesaian penelitian dalam rangka untuk memudahkan memecahkan masalah dari awal hingga tercapainya tujuan. Kerangka pemikiran ini berupa kerangka konseptual yang didesain untuk memberikan gambaran penelitian yang akan dilakukan. Adapun kerangka pemikiran sebagai berikut:

Gambar 2.8 Kerangka Pemikiran

