BAB II
LANDASAN TEORI

2.1. Media Pembelajaran

Daryanto (2013, p.4) mendefinisikan bahwa "Media merupakan salah satu komponen komunikasi yaitu sebagai pembawa pesan dari komunikator menuju komunikan, berdasarkan definisi tersebut dapat dikatakan bahwa proses pembelajaran merupakan proses komunikasi". Media pembelajaran merupakan suatu alat atau sarana bantu yang digunakan sebagai pembelajaran, sehingga terjadi proses perubahan dalam kemampuan sikap dan prilaku siswa yang relatif permanen sehingga akibat dari pengalaman ataupun pelatihan.

Media pembelajaran adalah segala sesuatu yang digunakan untuk menyalurkan pesan serta dapat merangsang pikiran, perasaan, perhatian, dan kemauan si belajar sehingga dapat mendorong terjadinya proses belajar. Peran seorang tenaga pengajar adalah membuat proses pembelajaran menjadi efektif dengan menggunakan media yang sesuai dengan tujuan pembelajaran yang telah ditetapkan.

2.1.1. Fungsi Media Pembelajaran

Daryanto (2013, p.8) mendefinisikan, "Media memiliki fungsi sebagai pembawa informasi dari sumber (dosen) menuju penerima (mahasiswa) dalam kegiatan interaksi antara mahasiswa dengan lingkungan, fungsi media dapat diketahui berdasarkan adanya kelebihan media dan hambatan yang mungkin timbul dalam proses pembelajaran".

Secara umum manfaat yang dapat diperoleh adalah proses pembelajaran lebih menarik, lebih interaktif, jumlah waktu mengajar dapat dikurangi kualitas belajar siswa dapat ditingkatkan dan proses belajar mengajar dapat dilakukan di mana dan kapan saja, serta sikap belajar siswa dapat ditingkatkan.

2.2. Bahasa Pemrograman C++

Bahasa pemrograman adalah kumpulan instruksi dalam bahasa yang dapat dimengerti manusia untuk dijalankan oleh komputer (Abdul Kadir, 2012:2). Bahasa pemrograman C++ dikembangkan dari bahasa C, pada awal tahun 1980 oleh Bjarne Stroustrup dari *AT & T Bell Laboratories*.

Komputer membutuhkan penerjemah untuk dapat menjalankan instruksi yang ditulis dalam bahasa pemrograman, hal ini dikarenakan komputer hanya mengerti bahasa mesin yang terdiri dari angka 0 dan 1. Dalam menerjemahkan bahasa pemrograman ke dalam bahasa mesin, dibutuhkan *interpreter* dan *compiler*.

Interpreter menerjemahkan bahasa pemrograman secara baris per baris sebelum di eksekusi atau dijalankan oleh komputer, sedangkan *compiler* menerjemahkan bahasa pemrograman ke dalam bahasa mesin secara keseluruhan, hasil dari proses kompilasi ini biasanya berbentuk aplikasi atau *executable file*.

2.3. *Mobile*

Echols dan Shadily (2013, p.1) dalam kamus bahasa Inggris-Indonesia menuliskan bahwa "*Mobile* adalah kata sifat yang berarti dapat bergerak atau dapat digerakkan dengan bebas dan mudah". Namun *mobile* dapat pula diartikan sebuah benda yang berteknologi tinggi dan dapat bergerak tanpa menggunakan kabel. Contohnya seperti *smartphone, PDA,* dan *tablet*.

2.4. Aplikasi

Setiawati (2009, p.8) mendefinisikan bahwa "Aplikasi merupakan penerapan, pengimplementasian suatu hal, data, permasalahan, pekerjaan kedalam suatu sarana atau media yang dapat digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan tersebut sehingga berubah menjadi suatu bentuk yang baru, tanpa menghilangkan nilai-nilai dasar dari hal, data, permasalahan atau pekerjaan".

2.5. *Android*

2.5.1. Pengertian Android

Irawan (2012, p.2) menyatakan bahwa "*Android* merupakan sebuah sistem operasi yang berbasis *Linux* untuk perangkat *portable* seperti *smartphone* dan komputer tablet". *Android* menyediakan *platform* terbuka bagi programmer untuk mengembangkan aplikasi sendiri pada berbagai perangkat dengan sistem operasi android.

Menurut Gargenta (2011, p8), Android adalah sebuah *comprehensive open-source platform* yang didesain untuk perangkat *mobile*. *Comprehensive platform* disini adalah setumpuk lengkap perangkat lunak yang dipakai pada perangkat *mobile*. Android dipelopori oleh Google dan dimiliki oleh *Open Handset Alliance*. Android merupakan *platform open source* pertama yang memisahkan perangkat keras dan perangkat lunak yang berjalan. Menurut Darcey dan Conder (2012, p4), Android adalah sebuah *mobile platform* pertama yang lengkap, *open source*, dan *gratis* yang dikembangkan dengan menggunakan *Software Development Kit* (SDK) yang *comprehensive* dengan *tools* yang cukup untuk mengembangkan aplikasi yang *powerful* dan kaya akan fitur.

2.5.2. Versi Pengembangan Android

Seperti halnya *software* yang lain, Android juga mengalami perkembangan setiap waktunya yang dapat dilihat dari jumlah versinya dimana terdapat perbedaan antara versi yang satu dengan yang sesudahnya. Pengembangan versi android dapat dilihat pada tabel 2.1 berdasarkan pada www.developer.android.com di tahun 2017.

Tabel 2.1 Versi Pengembangan Android

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.4%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.5%
4.1.x	Jelly Bean	16	2.0%
4.2.x		17	3.0%
4.3		18	0.9%
4.4	KitKat	19	13.4%
5.0	Lollipop	21	6.1%
5.1		22	20.2%
6.0	Marshmallow	23	29.7%
7.0	Nougat	24	19.3%
7.1		25	4.0%
8.0	Oreo	26	0.5%

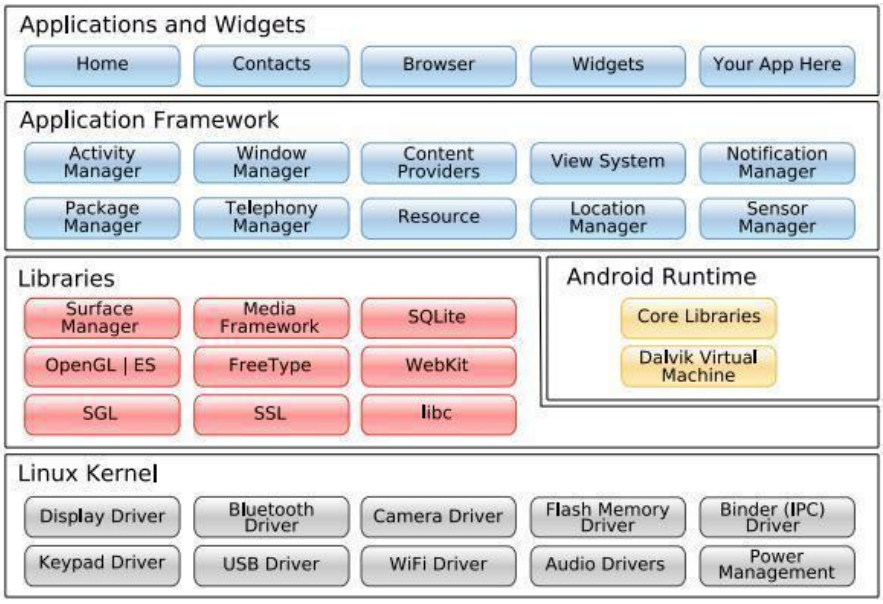
Penamaan versi-versi android tersebut berdasarkan pada nama-nama hidangan penutup (*dessert*). Pendistribusian Android dengan versi *Gingerbread*, merupakan versi yang paling banyak digunakan oleh para pengguna Android sampai tahun 2012.

2.5.3. Arsitektur Android

Arsitektur Android dapat digambarkan seperti pada gambar 1 dan secara garis besar Arsitektur Android dapat dijelaskan sebagai berikut:

a. *Application* dan *Widgets*

Application dan *Widgets* ini adalah layer dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita download aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Hampir semua aplikasi ditulis menggunakan bahasa pemrograman Java.



Gambar 2.1 Arsitektur Android

b. Application Frameworks

Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resource, menjalankan service background, mengatur alarm, dan menambah status *notifications*, dan sebagainya. Pengembang memiliki akses penuh menuju API *framework* seperti yang dilakukan oleh aplikasi kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan

(reuse). Sehingga bisa kita simpulkan Application Frameworks ini adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti *content providers* yang berupa sms dan panggilan telepon. Komponen-komponen yang termasuk di dalam *Application Frameworks* adalah sebagai berikut:

1. *Views*
2. *Content Provider*
3. *Resource Manager*
4. *Notification Manager*
5. *Activity Manager*

c. *Libraries*

Libraries ini adalah layer dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan di atas Kernel, layer ini meliputi berbagai library C/C++ inti seperti Libc SSL, serta:

a. *Libraries* media untuk pemutaran media audio dan video
b. *Libraries* untuk manajemen tampilan
c. *Libraries* Graphics mencakup SGL dan OpenGL
d. *Libraries* SQLite untuk dukungan database
e. *Libraries* SSL dan WebKit terintegrasi web browser dan security
f. *Libraries* LiveWebcore mencakup modern web browser dengan engine embedded webview
g. *Libraries* 3D yang mencakup implementasi OpenGL ES1.0 API's.

d. Android Run Time

Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux. Dalvik Virtual Machine (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam Android Run Time dibagi menjadi dua bagian yaitu:

1. Core Libraries: Aplikasi Android dibangun dalam bahasa Java, sementara Dalvik sebagai virtual mesinnya bukan Virtual Machine Java, sehingga diperlukan sebuah libraries yangberfungsi untuk menterjemahkan bahasa Java/C yang ditangani oleh Core Libraries.
2. Dalvik Virtual Machine: Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat Linux Kernel untuk melakukan threading dan manajemen tingkat rendah.

e. Linux Kernel

Linux Kernel adalah layer dimana inti dari system operasi Android itu berada. Berisi file-file system yang mengatur sistem processing, memory, resource, drivers, dan sistem-sistem operasi Android lainnya. Linux Kernel yang digunakan Android adalah Linux Kernel release 2.6.

2.5.4. APK

Irawan (2012, p.3) menulis bahwa "*Application Package File* atau berkas paket aplikasi *Android (APK)* adalah format berkas yang digunakan untuk mendistribusikan dan memasang *software* ke *ponsel* dengan sistem operasi *Android*, mirip dengan paket *MSI* pada *Windows* atau *Deb* pada *OS Debian*.

2.6. Software Pendukung dan Bahasa Pemrograman

2.6.1. Android Studio

Android Studio merupakan IDE (*Integrated Development Environment*) terbaru yang berbasis IntelliJ yang diluncurkan oleh Google pada konferensi para pembangun aplikasi pada tahun 2013 (Hohensee, 2014). Android Studio kedepannya akan digunakan sebagai aplikasi yang digunakan untuk membangun sebuah aplikasi berbasis Android menggantikan Eclipse. Ant sebagai sistem pembangun telah digantikan pada Android Studio ke Gradle.

Salah satu produk unggulan dari Android Studio adalah editor kode dengan pembangun fitur seperti "*Smart Editing*" untuk menyediakan kode yang lebih bisa dibaca oleh para penggunanya. Fitur kelebihan lain dari Android Studio

adalah sistem pembangun yang berbasis Gradle. Gradle mengizinkan para pembangun aplikasi untuk mengaplikasikan pengaturan yang berbeda pada kode yang sama, untuk memproduksi versi yang berbeda dari aplikasi yang sama. Secara umum Gradle meningkatkan penggunaan kembali kode dan integrasi dari server yang dibangun.

Gambar 2.2 Logo Android Studio

Fitur desain dan teks dari editor Android Studio telah berkembang daripada Eclipse. Editor terbaru menampilkan tampilan langsung dari rancangan aplikasi untuk resolusi, versi Android dan karakteristik tersendiri dari sebuah negara. Android Studio memiliki beberapa servis baru dan terintegrasi yang memudahkan para penggunanya untuk menangani terjemahan atau untuk koneksi dengan *Google Cloud Messaging* (CGM) yang mampu untuk mengirimkan pesan ke aplikasi dan menerima pesan dari aplikasi.

2.6.2. Java Programming Language

Java adalah suatu teknologi di dunia *software* komputer, yang merupakan suatu bahasa pemrograman, dan sekaligus suatu *platform*. Sebagai bahasa pemrograman, *Java* dikenal sebagai bahasa pemrograman tingkat tinggi. *Java* mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++. *Java* merupakan bahasa pemrograman berorientasi objek yang merupakan paradigma pemrograman masa depan. Sebagai bahasa pemrograman *Java*

dirancang menjadi handal dan aman. *Java* juga dirancang agar dapat dijalankan di semua platform, dan juga dirancang untuk menghasilkan aplikasi-aplikasi dengan performansi yang terbaik, seperti aplikasi database *Oracle* 8i/9i yang *core*-nya dibangun menggunakan bahasa pemrograman Java. Sedangkan Java bersifat *neutral architecture*, karena *Java Compiler* yang digunakan untuk mengkompilasi kode program Java dirancang untuk menghasilkan kode yang netral terhadap semua arsitektur perangkat keras. Bahasa ini digunakan dalam pengembangan perangkat lunak berbasis *android*.

2.6.3. XML (Extensible Markup Language)

XML (Erick, 2003) adalah singkatan dari eXtensible Markup Language. Bahasa markup adalah sekumpulan aturan-aturan yang mendefinisikan suatu sintaks yang digunakan untuk menjelaskan, dan mendeskripsikan teks atau data dalam sebuah dokumen melalui penggunaan tag. Bahasa markup lain yang populer seperti HTML, menggambarkan kepada browser web tentang bagaimana menampilkan format teks, data, dan grafik ke layar komputer ketika sedang mengunjungi sebuah situs web.

XML adalah sebuah bahasa markup yang digunakan untuk mengolah meta data (informasi tentang data) yang menggambarkan struktur dan maksud/tujuan data yang terdapat dalam dokumen XML, namun bukan menggambarkan format tampilan data tersebut. XML adalah sebuah standar sederhana yang digunakan untuk medeskripsikan data teks dengan cara *self-describing* (deskripsi diri). XML juga digunakan dalam pengembangan *android* untuk merancang *user interface*.

2.6.4. SQLite

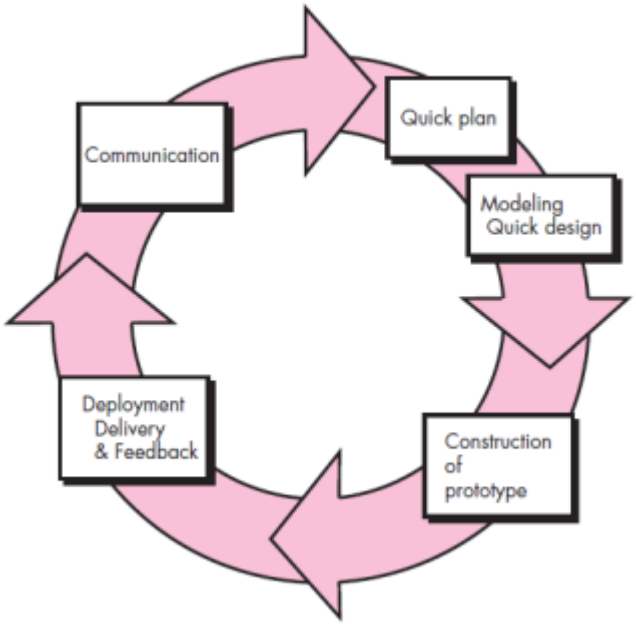
SQLite adalah paket perangkat lunak publik-domain yang menyediakan sebuah manajemen sistem relasi database, atau RDBMS (Relational Database Management System) (Kreibich, 2010). Sistem relasi database digunakan untuk menyimpan rekor yang ditetapkan oleh pengguna pada sebuah tabel yang besar. Kata "*Lite*" pada *SQLite* tidak ditujukan untuk kapabilitasnya.

Melainkan, SQLite merupakan aplikasi yang mudah digunakan untuk tingkat instalasi, administrasi dan penggunaan sumber daya.

2.7. Metode Pengembangan Perangkat Lunak

2.7.1. Model Prototype

Pressman (2012, p40) menyatakan bahwa *Prototyping pardigma* dimulai dengan pengumpulan kebutuhan, pengembang bertemu dengan pengguna dan mengidentifikasikan objektif keseluruhan dari perangkat lunak, selanjutnya mengidentifikasi segala kebutuhan yang diketahui secara garis besar di mana definisi-definisi lebih jauh merupakan keharusan, kemudian dilakukan perancangan kilat, lalu diakhiri dengan evaluasi *prototyping* yang dapat dilihat pada gambar 2.3 berikut:



Gambar 2.3 Model *Prototype*

Tahap–tahap rekayasa *software* dalam *prototype model* pada gambar 2.5 di atas adalah sebagai berikut:

1. Communication (komunikasi), pada tahapan ini, pengguna dan pengembang bersama-sama mengidentifikasi kebutuhan, format perangkat lunak, garis besar perangkat lunak yang akan dikembangkan.

2. Quick Plan, menggunakan hasil dari komunikasi dengan pengguna untuk membuat suatu tahapan perencanaan secara cepat.
3. Modeling, pada tahap permodelan, termasuk didalamnya tahapan perancangan sederhana dilakukan untuk menggambarkan representasi dari perangkat lunak pada pengguna, misalnya interface atau format output dari perangkat lunak.
4. Construction of Prototype, setelah melakukan tahap permodelan, tahapan selanjutnya adalah pembuatan prototype dari perangkat lunak tersebut berdasarkan tahap-tahap sebelumnya.
5. Deployment and Feedback, prototype di gunakan dan di evaluasi oleh pengguna perangkat lunak. Hasil dari evaluasi digunakan sebagai bahan masukan untuk proses pengembangan kembali dengan mengidentifikasi ulang kebutuhan perangkat lunak berdasarkan kekurangan dari protoype yang telah dihasilkan.

2.8. *Unified Modeling Language* (*UML*)

Widodo dan Herlawati (2011, p.6) menjelaskan bahwa "*UML* adalah singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. Dalam proses iteratif, pengembang melakukan beberapa langkah berulang-ulang dan setiap waktu berfokus pada bagian-bagian yang berbeda dari sistem".

UML diaplikasikan untuk maksud tertentu sebagai berikut:

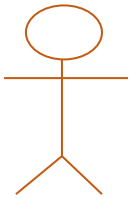
1) Merancang perangkat lunak.
2) Sarana komunikasi antara perangkat lunak dengan proses.
3) Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan system.
4) Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

Alat-alat yang digunakan dalam suatu metodologi umumnya berupa suatu gambar diagram atau grafik. Adapun alat pengembangan sistem yang digunakan yaitu sebagai berikut:

1) *Use Case Diagram*

Widodo dan Herlawati, (2011, p. 21) berpendapat bahwa "*Use case* sebagai urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomatis maupun secara manual untuk tujuan melengkapi satu tugas tunggal seperti yang terdapat dalam penjelasan simbol pada tabel 2.2 berikut:

Tabel 2.2 Bagan *Use Case Diagram*

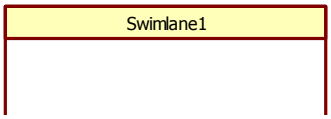
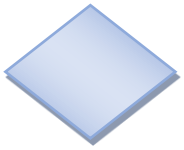
Simbol	Keterangan
ACTOR 	Mempresentasikan sesorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.
Use Case 	Gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.
Relasi 	Associaton, menghubungkan link antar element

2) *Activity Diagram*

Activity Diagram merupakan tipe khusus dari *diagarm* status yang memperlihatkan aliran suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Widodo dan Herlawati (2011, p.144) menggambarkan aktifitas-aktifitas, objek, state, transisi sistem dan event.

Dengan kata lain kegiatan *diagarm* alur kerja menggambarkan perilaku sistem untuk aktivitas *diagarm* ini terutama penting dalam pemodelan fungsi – fungsi suatu sistem dan memberi tekanan pada aliran kendali antar object sperti yang terdapat dalam penjelasan simbol pada tabel 2.3 berikut:

Tabel 2.3 Bagan *Activity Diagram*

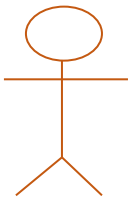
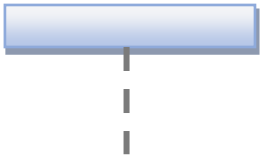
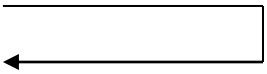
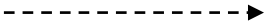
Simbol	**Keterangan**
Swimlane 	Digunakan untuk menunjukan siapa yang melakukan apa atau siapa melakukan proses apa, bukan menunjukan apa yang terjadi.
Initial State 	Memperlihatkan dimana aliran berawal
Action State 	Merupakan langkah atau aksi – aksi yang terjadi
Desicion 	Memperlihatkan dimana keputusan perlu diambil selama terjadi aliran kerja
Syncronization 	Merupakan percabangan dari sebuah aksi – aksi yang lain yang bisa saja terjadi secara paralel
Final State 	Memperlihatkan dimana aliran kerja berakhir

3) Sequence Diagram

Widodo dan Herlawati (2011, p.173) berpendapat bahwa "*Diagarm sequence* adalah *interaction diagarm* yang memperlihatkan event-event yang berurutan sepanjang berjalannya waktu". *Sequence diagram* menjelaskan interaksi objek yang disusun berdasarkan urutan waktu.

Secara mudahnya sequence *diagarm* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan use case *diagarm*, seperti yang terdapat dalam penjelasan simbol pada tabel 2.4 berikut:

Tabel 2.4 Bagan *Sequence Diagram*

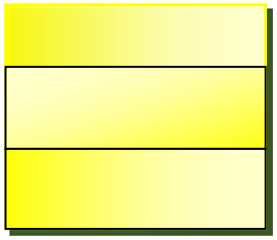
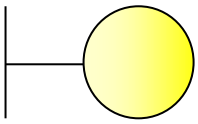
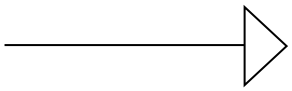
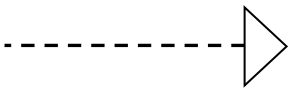
Simbol	**Keterangan**
ACTOR 	Pelaku dalam sequence *diagarm*
Object 	Menambahkan objek baru pada *Diagarm*
Stimulus 	Menggmbar pesan *(message)* antar dua objek
SelfStimulus 	Menggambar pesan *(message)* yang menuju dirinya sendiri
Return message 	Menggambarkan pengembalian dari pemanggilan prosedur

4) *Class Digram*

Widodo dan Herlawati (2011 p.37) mengatakan bahwa "*Class diagarm* menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain lain". *Class Digram* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut / properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode / fungsi).

Penjelasan simbol *class diagram* terdapat pada tabel 2.5 berikut:

Tabel 2.5 Bagan *Class Digram*

Simbol	Keterangan
CLASS 	Menambahkan kelas baru pada *Diagram*
Association 	Menggambar relasi asosiasi
Interface 	Menambahkan kelas antar muka *(interface)* pada diagram
Generalization 	Menggambarkan relasi generalisasi
Realize 	Menggambarkan relasi realisasi
Association class	Menghubungkan kelas asosiasi (association class) pada suatu relasi asosiasi
Return message	Menggambarkan pengembalian dari pemanggilan prosedur
Aggregation 	Menggambarkan relasi agregasi

2.9. Penelitian Terdahulu

Fungsi penelitian terdahulu, salah satunya adalah untuk mendapatkan permasalahan yang bisa dijadikan sebagai dasar penulisan latar belakang masalah dalam penelitian ini. Pada tabel 2.6 berikut ini menguraikan tentang penelitian yang pernah dilakukan sebelumnya:

Tabel 2.6 Penelitian Terdahulu

Nama	Judul/Tahun	Terbit	Keterangan
Tri Sakti Soangkupon	Rancang Bangun Media Pembelajaran Bimbingan Konseling pada SMA Immanuel berbasis Mobile/2015	Jurusan Teknik Informatika Fakultas Ilmu Komputer Intitut Informatika dan Bisnis Darmajaya Lampung	Hasil output dari penelitian ini adalah alternatif media belajar selain buku berupa aplikasi media pembelajaran bimbingan konseling yang dapat diakses secara mobile. Aplikasi ini menggunakan bahasa pemrograman Action Script 3 dan dibangun menggunakan Adobe Flash CS6. Dengan media pembelajaran ini diharapkan siswa-siswi dapat belajar secara interaktif dan efektif melalui smartphone android dengan fleksibel.
Rico Chandra	Rancang Bangun Media Pembelajaran Agama Budha Berbasis Online/2016	Jurusan Teknik Informatika Fakultas Ilmu Komputer Intitut Informatika dan Bisnis Darmajaya Lampung	Hasil output dari penelitian ini adalah alternatif media belajar selain buku berupa aplikasi media pembelajaran agama Budha yang dapat diakses dengan perangkat mobile. Dibangun menggunakan Adobe Flash CS6. Media belajar ini membantu anak-anak untuk dapat belajar secara interaktif dan efektif melalui smartphone android dengan fleksibel.