

BAB II

LANDASAN TEORI

2.1 Definisi Penelitian dan Pengembangan

Penelitian dan pengembangan adalah metode penelitian dengan tujuan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut. Menurut Sugiono (2011: 297)

Bahwa penelitian dan pengembangan adalah “penelitian dan pengembangan bertujuan untuk menghasilkan produk baru melalui proses pengembangan “Secara umum penelitian dan pengembangan merupakan penelitian yang bertujuan untuk menghasilkan produk baru dan menguji keefektifan produk tersebut. Pendapat lain diutarakan oleh Mulyatiningsi (2011:161)

2.2 Jenis-jenis Penelitian dan Pengembangan

Menjelaskan ada beberapa langkah dalam model pengembangan ini. Borg and Gall (dalam Mulyatiningsih, 2011: 193) .

Langkah-langkah pelaksanaannya adalah :

a. Penelitian dan Pengumpulan Data

Pada tahap ini ada tiga kegiatan utama yang harus dilakukan yaitu analisis kebutuhan, studi literatur, dan penelitian skala kecil.

b. Perencanaan

Pada tahap perencanaan dilakukan identifikasi kecakapan keahlian yang diperlukan dalam pelaksanaan penelitian kemudian membuat desain atau langkah-langkah penelitian dan merancang kemudian pengujian dalam lingkup terbatas.

c. Mengembangkan Produk Awal

Mengembangkan bentuk permulaan awal produk yang akan dihasilkan. Kegiatan utama yang dilakukan pada tahap ini antara lain penyiapan bahan ajar dan komponen pendukung lainnya, penyiapan pedoman dan buku petunjuk serta melakukan evaluasi terhadap kelayakan alat-alat pendukung.

2.3 Lembar Kerja Siswa (LKS) Berbasis Android

LKS berbasis android ini digunakan untuk meningkatkan kemampuan pengetahuan dan keterampilan siswa dalam standard kompetensi. LKS berbasis android ini dibuat untuk memanfaatkan aplikasi yang ada di play store. Lembar kerja siswa, yaitu materi ajar yang sudah dikemas sedemikian rupa. Pentingnya LKS bagi kegiatan Pembelajaran berbicara mengenai pentingnya LKS bagi kegiatan pembelajaran, maka kita tidak bisa lepas dari pengkajian tentang fungsi LKS itu sendiri berikut adalah penjabaran dari masing-masing kajian tersebut. (Prastowo,2015:204).

1. Sebagai bahan ajar yang bisa meminimalkan peran pendidik namun lebih mengaktifkan peserta didik.
2. Sebagai bahan ajar yang mempermudah peserta didik untuk memahami materi yang diberikan.
3. Sebagai bahan ajar yang ringkas dan kaya tugas untuk berlatih serta
4. Memudahkan pelaksanaan pengajaran kepada peserta didik

2.4 Android

Android adalah sebuah sistem operasi untuk *smartphone* dan tablet. Sistem operasi dapat diilustrasikan sebagai 'jembatan' antara piranti (*device*) dan penggunanya, sehingga pengguna bisa berinteraksi dengan *device*-nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device*. (Sari, 2016).

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. Android adalah sistem operasi untuk telepon seluler yang berbasis *linux*. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh Android Inc, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh *Google Inc*. Untuk pengembangannya, dibentuklah Open Handset Alliance (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan

telekomunikasi termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile,* dan *Nvidia* (Nazrudin Safaat H, 2015, p.1).

2.4.1 Versi Android

Adapun versi-versi android yang pernah dirilis adalah sebagai berikut :

- a. Android Versi 1.0 Alpha - Dirilis pada 23 September 2008.
- b. Android Versi 1.1 Bender (Beta) - Dirilis pada 9 Februari 2009.
- c. Android Versi 1.5 CupCake - Dirilis pada 27 April 2009.
- d. Android Versi 1.6 Donut - Dirilis pada 15 September 2009.
- e. Android Versi 2.0 - 2.1 Eclair - Dirilis pada 26 Oktober 2009.
- f. Android Versi 2.2 Frozen Yoghurt - Dirilis pada 10 Mei 2010.
- g. Android Versi 2.3 GingerBread - Dirilis pada 6 Desember 2010.
- h. Android Versi 3.0 - 3.2 HoneyComb - Dirilis pada 22 Pebruari 2011.
- i. Android Versi 4.0 Ice Cream Sandwich - Dirilis pada 19 Oktober 2011.
- j. Android Versi 4.1 - 4.3 Jeally Bean - Dirilis pada 27 Juni 2012.
- k. Android Versi 4.4 KitKat - Dirilis pada 31 Oktober 2013.
- l. Android Versi 5.0 Lollipop - Dirilis pada 5 Juni 2014.
- m. Android Versi 6.0 Marshmallow - Dirilis pada 17 Agustus 2015.
- n. Android Versi 7.0 Nougat - Dirilis pada 18 Juli 2016.
- o. Android Versi 8.0 Oreo - Dirilis pada 21 Agustus 2017.



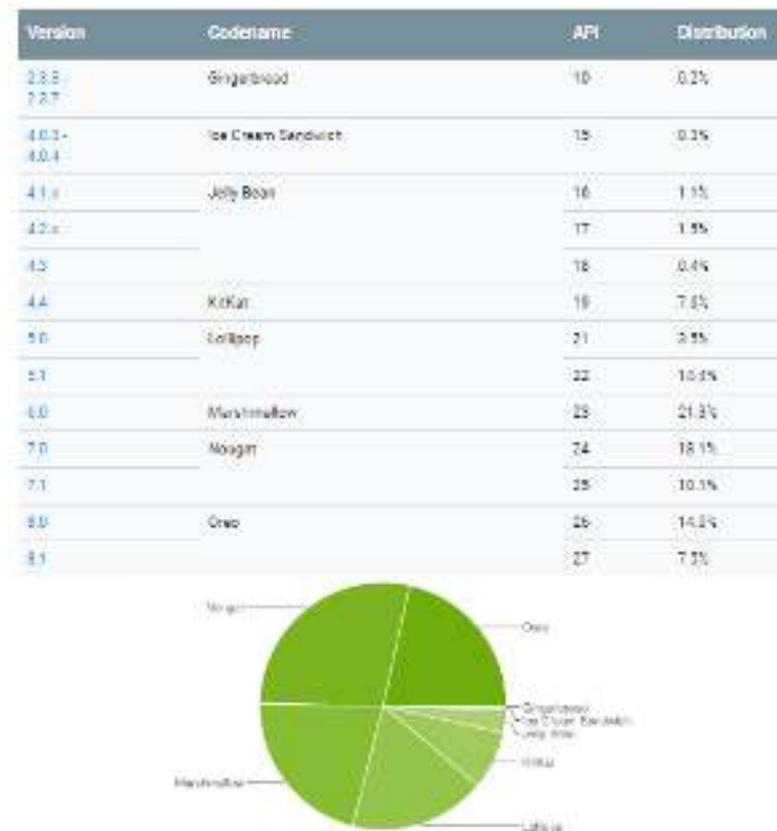
Gambar 2.1 Evolusi Android.

(sumber : <http://en.miui.com/thread-475925-1-1.html>)

2.4.2 Statistik Distribusi OS Android

Google baru-baru ini merilis angka statistik distribusi sistem operasi Android di berbagai perangkat mobile di seluruh dunia pada 26 Oktober 2018. Dalam statistik tersebut terlihat Android 6.0 *Marshmallow* menjadi yang tertinggi dengan persentase total 21.3%. Di urutan kedua berhasil dikuasai oleh Android 7.0 *Nougat* yang memiliki persentase mencapai 18.1%. Kedua sistem operasi ini memang layak menduduki angka tertinggi karena saat ini smartphone Android memang masih banyak yang menggunakan Android *Marshmallow* dan *Nougat* mengingat vendor-vendor di seluruh dunia masih mempercayakan kedua sistem operasi ini pada smartphone buatannya. Sementara pada urutan ketiga diduduki Android 5.1 *Lollipop* dengan persentase 14.4% yang disusul oleh Android 8.0 *Oreo* dengan persentase 14.0%. Kedua sistem operasi ini masih banyak digunakan pada smartphone *entry-level* di berbagai negara. Selanjutnya, Android 7.1 *Nougat* dengan persentase total 10.1%. Selanjutnya, Android 4.4 *Kitkat* dengan persentase 7.6%. Selanjutnya Android 8.1 *Oreo* dengan persentase 7.5% dan android versi 5.0 *Lollipop* sudah mulai ditinggalkan pengguna Android karena hanya memiliki persentase 3.5% dan untuk urutan di bawahnya yaitu oleh android dengan versi 4.1.x, 4.2.x dan 4.3 *Jelly Bean* dengan total 3.0% berikutnya android versi 4.0.3 dan 4.0.4 *Ice cream sandwich* dengan persentase 0.3% di urutan terakhir yaitu android versi 2.3.3-2.3.7 *Ginger bread* dengan persentase 0.2%. Hal ini tentu wajar karena Google sendiri sudah menghentikan dukungan untuk sistem operasi *Ginger bread* ini pada Q1 2017. Sementara versi *Ice Cream Sandwich* yang meski terus melempem tetapi masih akan didukung dalam waktu lebih lama.

Berikut data statistik pengguna sistem operasi android yang dikumpulkan selama periode 7 hari yang berakhir pada 26 Oktober 2018, Setiap versi dengan distribusi kurang dari 0,1% tidak ditampilkan.



Gambar 2.2 Statistik distribusi OS Android (26 Oktober 2018)

(sumber :<https://developer.android.com/about/dashboards/>)

2.5 Android Studio

Android Studio adalah sebuah IDE untuk Android Development yang dikenalkan pihak google pada acara Google I/O di tahun 2013. Android Studio merupakan suatu pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android. Android Studio adalah Lingkungan Pengembangan Terpadu Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA . Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya:

1. Sistem versi berbasis Gradle yang fleksibel
2. Emulator yang cepat dan kaya fitur

3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
4. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
5. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
6. Alat pengujian dan kerangka kerja yang ekstensif
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
8. Dukungan C++ dan NDK
9. Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine

2.5.1 Pengenalan Layout

Membicarakan masalah tampilan atau layout, pada android studio juga sama halnya seperti pada eclipse, layout di android studio ini dibedakan menjadi 2, yaitu design dan juga text, secara default pada saat dibuka layout xml di android studio menggunakan mode design, tapi dalam hal ini kalian bisa mengubah menjadi mode text, sehingga terlihat source code atau barisan susunan kode dari text XML nya.

2.5.2 Barisan atau Struktural Folder

Pada bagian folder build seperti gen/ yang terdapat pada eclipse, isinya hanya merupakan file-file hasil generate dari IDE, jadi tidak perlu diubah-ubah isi folder tersebut. Kemudian folder libs/ sama seperti di eclipse. Jika kalian menggunakan library dalam bentuk .jar maka tempatkan di folder libs. Selanjutnya folder src/. , src adalah folder dimana tersimpan source java dan layout dalam bentuk XML, secara default android studio akan mengenerate kedua folder yaitu, androidtest dan main. Folder androidtest adalah folder khusus UnitTest.

2.5.3 Gradle

Salah satu fitur teranyar pada Android Studio adalah fitur gradle, gradle adalah sebuah featured build automation. Seperti yang tertera pada nama yang diusungnya, fitur ini dapat membantu kalian membuat suatu fitur animasi dengan

cukup mudah. Bagi kalian yang ingin mengetahuinya lebih lanjut, kalian dapat mengunjungi situs resminya di <http://www.gradle.org/>. File Gradle berisi library yang digunakan, versi aplikasi, signed key properties, lokasi repository dll.

2.5.4 Elemen Android

a. *Dalvik Virtual Machine (DVM)*

Salah satu element kunci dari Android adalah *Dalvik Virtual Machine (DVM)*. Android berjalan di *Dalvik Virtual Machine (DVM)* bukan di Java Virtual Machine (JVM), sebenarnya banyak persamaannya dengan Java Virtual Machine (JVM) seperti *Java ME (Java Mobile Edition)*, tetapi Android menggunakan virtual *machine* sendiri yang dirancang untuk memastikan beberapa fitur-fitur berjalan lebih efisien pada perangkat mobile.

b. *Android SDK (Software Development Kit)*

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada plat *form* Android menggunakan bahasa pemrograman *Java*. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi *middleware* dan aplikasi kunci yang di-*release* oleh *Google*. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman *Java*. Sebagai *platform* aplikasi -*netral*, Android memberi kesempatan untuk membuat aplikasi yang dibutuhkan.

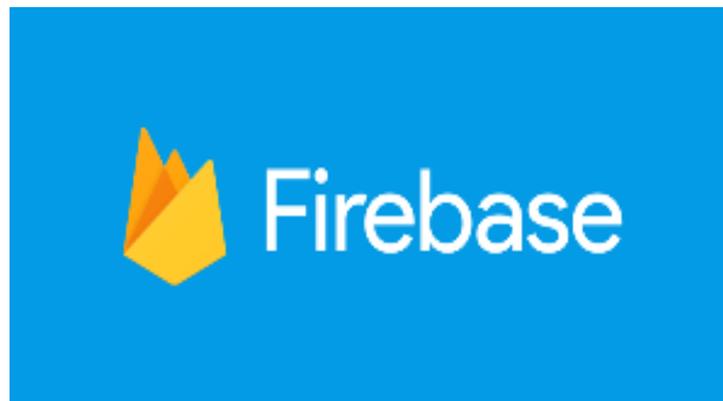
c. *ADT (Android Development Tools)*

Android *development tools* adalah plugin yang di desain untuk IDE Android Studio yang memberikan kemudahan dalam mengembangkan aplikasi Android dengan menggunakan IDE Android Studio. Dengan menggunakan ADT untuk Android Studio akan memudahkan dalam membuat aplikasi *project* Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya.

2.6 Firebase

Firebase adalah layanan pihak ketiga, *Firebase* bisa dikatakan sebagai layanan [DbaaS](#) (*Database as a Service*) dengan konsep *realtime*. Tidak hanya berfungsi sebagai tempat penyimpanan data, tapi juga di sediakan API untuk implementasi web *socket*. *Firebase* menyediakan *library* untuk berbagai *client platform*. Browser menggunakan Java script API dan mobile menggunakan OBJ-C atau Android API.

Berikut penjelasan beberapa fitur yang tersedia pada *firebase* :



Gambar 2.3 logo firebase

(sumber : <https://firebase.google.com/>)

a. *Firebase Cloud Messaging* (FCM)

Firebase Cloud Messaging untuk Android (FCM) adalah layanan yang membantu pengembang mengirim data dari server untuk aplikasi mereka Android pada perangkat Android. Ini bisa menjadi pesan ringan memberitahu aplikasi Android bahwa ada data baru yang akan diambil dari server (misalnya, film diunggah oleh seorang teman), atau bisa juga pesan yang berisi sampai dengan 4KB data payload (sehingga aplikasi seperti instant messaging dapat mengkonsumsi pesan langsung). Layanan FCM menangani semua aspek antrian pesan dan pengiriman ke aplikasi target Android berjalan pada perangkat target.

b. *Firestore Authentication*

Firestore Authentication menyediakan layanan backend, SDK yang mudah digunakan, dan library UI yang siap pakai untuk mengautentikasi pengguna ke aplikasi Anda. *Firestore Authentication* mendukung autentikasi menggunakan sandi, nomor telepon, penyedia identitas gabungan yang populer, seperti *Google*, *Facebook*, dan *Twitter*, dan lain-lain.

c. *Firestore Storage*

Cloud Storage untuk *Firestore* adalah layanan penyimpanan objek yang andal, sederhana, dan hemat biaya yang dibuat untuk skala *Google*. *Firestore* SDK untuk *Cloud Storage* menambahkan keamanan *Google* pada upload dan download file untuk aplikasi *Firestore* Anda, bagaimana pun kualitas jaringannya. Anda dapat menggunakan SDK kami untuk menyimpan gambar, audio, video, atau konten buatan penggunalainnya. Di server, Anda dapat menggunakan [Google Cloud Storage](#) untuk mengakses file yang sama.

d. *Firestore Realtime Database*

Firestore Realtime Database adalah *database* yang di-host di *cloud*. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap *klien* yang terhubung. Ketika Anda membuat aplikasi lintas *platform* dengan SDK Android, iOS, dan *JavaScript*, semua *klien* akan berbagi sebuah *instance Realtime Database* dan menerima update data terbaru secara otomatis.

2.7 *E-Learning*

Melihat perkembangan *E-Learning* dari masa kemasa yang terus berkembang mengikuti perkembangan teknologi, maka dapat disimpulkan bahwa *E-Learning* akan menjadi sistem pembelajaran masa depan. Alasan efektifitas dan fleksibilitas akan menjadi alasan utama. Pemanfaatan dari perkembangan *E-Learning* berdasarkan waktu, terbagi menjadi dua jenis yaitu *synchronous* dan *asynchronous*. *Synchronous* berarti pada waktu yang sama interaksi terjadi antara guru dan murid melalui website. Implementasi *synchronous* ini adalah virtual *classroom*, Sedangkan *asynchronous* memberikan keleluasaan kepada murid

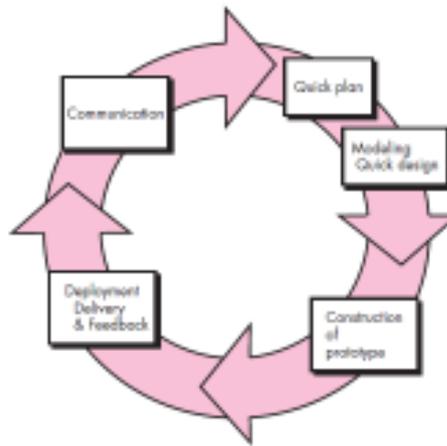
untuk belajar kapan pun tanpa harus secara langsung pada waktu yang sama berinteraksi dengan guru. Metode *asynchronous* dapat berupa *embedded learning*, *course* dan *discussion groups*.(Hasibuan & Meitro, 2014).

2.8 Model Prototype

Metode *Prototype* adalah proses pembuatan model sederhana *software* yang mengizinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal. *Prototype* memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat lunak yang akan di buat. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat lunak yang dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna. Kemudian membuat sebuah rancangan kilat yang selanjutnya akan dievaluasi kembali sebelum di produksi secara benar. *Prototype* bukanlah merupakan sesuatu yang lengkap, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi pada saat *prototype* dibuat untuk memenuhi kebutuhan pengguna dan saat yang sama memungkinkan pengembangan untuk lebih memahami kebutuhan pengguna secara baik (Pressman, 2012).

Berikut adalah tahapan dalam metode *prototype* :

1. Komunikasi (*Communication*) dan pengumpulan data awal, yaitu komunikasi dengan klien dan *user* untuk menentukan kebutuhan.
2. Perencanaan cepat (*Quick Plan*), yaitu pembuatan perencanaan analisis terhadap kebutuhan pengguna.
3. Pemodelan perancangan cepat (*Modeling Quick Design*), yaitu membuat rancangan desain program.
4. Pembentukan *prototype* (*Construction of prototype*), yaitu pembuatan aplikasi berdasarkan dari pemodelan desain yang telah dibuat.
5. Penyerahan sistem dan umpan balik (*Development Delivery and Feedback*), yaitu memproduksi perangkat secara benar sehingga dapat digunakan oleh pengguna.



Gambar 2.5 Diagram Prototype

(sumber : Roger S.Pressman, Ph.D. (2012). Rekayasa Perangkat Lunak)

2.9 Unified Modelling Language (UML)

Mendefinisikan bahwa, Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak, UML menawarkan sebuah standar untuk merancang model sebuah sistem. Tujuan Penggunaan UML yaitu untuk memodelkan suatu sistem yang menggunakan konsep berorientasi objek dan menciptakan bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin. (Rosa dan Shalahuddin2016, p.133).

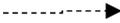
Menurut Rosa dan Shalahuddin (2016, p.140) tipe-tipe diagram UML adalah sebagai berikut :

2.9.1 Use Case Diagram

Use case diagram adalah gambar dari beberapa atau seluruh aktor dan *use case* dengan tujuan yang mengenali interaksi mereka dalam suatu sistem. *Use case* diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara *actor* dan sistem.

Dalam *use case* diagram terdapat istilah seperti aktor, *use case* dan *case relationship*. Penjelasan simbol *use case* diagram ditunjukkan pada tabel 2.1.

Tabel 2.1 Simbol Use Case Diagram.

Simbol	Keterangan
	Aktor : Seseorang atau sesuatu yang berinteraksi dengan sistem yang sedang dikembangkan.
	<i>Use case</i> : perangkat tertinggi dari fungsionalitas yang dimiliki sistem.
	Association : adalah relasi antara <i>actor</i> dan <i>use case</i> .
	Generalisasi : untuk memperlihatkan struktur pewaris yang terjadi.

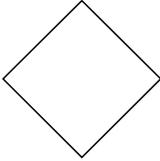
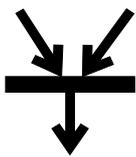
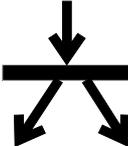
2.9.2 Activity Diagram

Activity Diagram menggambarkan rangkaian aliran dari aktifitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi *Activity* Diagram berupa *flow chart* yang digunakan untuk memperlihatkan aliran kerja dari sistem. Notasi yang digunakan dalam *activity* diagram ditunjukkan pada tabel 2.2.

Tabel 2.2 Simbol Activity Diagram.

Simbol	Keterangan
	<i>Activity</i> : Menunjukkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Initial Node</i> : Bagaimana objek dibentuk atau diawali
	<i>Activity Final Node</i> : Bagaimana objek dibentuk dan diakhiri.

Lanjutan Tabel 2.2

	<p><i>Decision</i> : Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.</p>
	<p><i>Swim lane</i> : Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.</p>
	<p>Join : Digunakan untuk menunjukkan kegiatan yang digabungkan.</p>
	<p>Fork : Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel</p>

2.9.3 Sequence Diagram

Sequence Diagram menggambarkan kolaborasi dinamis antara sejumlah dan untuk menunjukkan rangkaian pesan yang dikirim antar objek juga interaksi antar objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Dalam sequence diagram terdapat 2 simbol yaitu : *Actor*, untuk menggambarkan pengguna sistem. *Lifeline*, untuk menggambarkan kelas dan objek.

2.9.4 Class Diagram

Class diagram menggambarkan struktur data dan deskripsi class, *package*, dan objek beserta hubungan satu sama lain. *Class* diagram berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain. *Class* memiliki 3 area pokok yaitu nama, atribut dan metode.

2.10 Pengujian *Black Box Testing*

Black-Box Testing atau Pengujian Kotak Hitam atau juga disebut *Behavioral Testing*, berfokus pada persyaratan fungsional dari perangkat lunak. Artinya, teknik *Black-Box Testing* memungkinkan untuk mendapatkan *set* kondisi masukan yang sepenuhnya akan melaksanakan semua persyaratan fungsional untuk suatu program (Pressman, 2012).

Black-Box Testing bukan merupakan *alternatif* dari pengujian *White-Box Testing*. Sebaliknya, *Black-Box Testing* adalah pendekatan komplementer yang mungkin untuk mengungkap kelas yang berbeda dari kesalahan daripada metode *White-Box Testing*.

Black-Box Testing mencoba untuk menemukan kesalahan dalam kategori berikut :

- a. Fungsi tidak benar atau hilang.
- b. Kesalahan *interface* atau antarmuka.
- c. Kesalahan dalam struktur data atau akses *database* eksternal.
- d. Kesalahan kinerja atau perilaku.
- e. Kesalahan inisialisasi dan terminasi.

2.11 Penelitian Terdahulu

Tabel 2.3 berikut ini adalah beberapa penelitian terdahulu yang berkaitan dengan Pengembangan Lembar Kerja Siswa Berbasis Android.

Tabel 2.3 Penelitian Terdahulu

No.	Judul	Nama	Tahun
1	Analisis Pelaksanaan Pembelajaran Kurikulum 2013 Oleh Guru Mata Pelajaran Produktif di SMK Negeri 5 Bantaeng	Rismayanti Syam, Heru Winarno & Panennungi T	2018
2	Pengembangan Lember Kerja Siswa Elektronik (LKS) Berbasis Android Sebagai Media	Cut Tiara Permata Sari	2018

	Pembelajaran Pendukung Implementasi Kurikulum 2013 Pada Siswa Kelas X Akuntansi SMK IRA Medan Tahun Pelajaran 2017/2018		
3	Pengembangan Media Pembelajaran Berbasis Android Untuk Meningkatkan Aktivitas Belajar Siswa	Mukhammad Mukhlis Abdilah	2017
4	Membangun kamus obstetri berbasis android (studi kasus: kampus kebidanan adila bandar lampung)	Yuni Puspita Sarti	2016
5	System E-Learnig Dengan Pendekatan Evaluasi Pembelajaran	Muhammad Said Hasibuan	2009