BAB II
LANDASAN TEORI

2.1 Multimedia Interaktif

Multimedia adalah gabungan berbagai media (format file) dari teks, suara, citra, maupun video. Dari gabungan media tersebut diintegrasikan kedalam komputer untuk disimpan kemudian diolah dan disajikan secara bersamaan. Multimedia sendiri telah banyak dimanfaatkan dalam berbagai bidang diantaranya : media pembelajaran, game, film, me-dis, militer, bisnis, desain, arsitektur, olahraga, hobi, iklan atau promosi, dll. Multimedia bermaksud memaksimalkan setiap indera dalammenerima suatu informasi, sehingga dapat digunakan untuk menyampaikan pesan kepada publik (Danang, 2014)

Sedangkan pengertian interaktif terkait dengan komunikasi dua arah atau lebih dari komponen – komponen komunikasi. Dimana komponen komunikasi dalam sebuah multimedia interaktif berupa : hubungan antara manusia (sebagai *user*/pengguna produk) dan computer (softwere/aplikasi/produk dalam format file tertentu biasanya dalam bentuk CD). Dalam multimedia interaktif, interaksi merupakan salah satu fitur yang menonjol dalam multimedia yang memungkinkan pembelajaran yang aktif (*active learning*), yang tidak saja memungkinkan pengguna melihat atau mendengar (*see and hear*) tetapi juga melakukan sesuatu (*do*). Dalam konteks multimedia *do* disini dapat berupa memberikan respon terhadap pertanyaan yang diajukan komputer atau aktif dalam simulasi yang disediakan komputer.

2.2 Metode Inferensi : *Fisher Yates Shuffel*

Menurut Kresna, I. B. dkk. (2015), Metode Pengacakan Fisher-Yates diadopsi dari nama penemunya, yaitu Ronald Fihser dan Frank Yates, yang pertama kali diperkenalkan pada tahun 1934 dan kemudian di revisi kembali pada tahun 1948. Algoritma Fisher-Yates Shuffel adalah sebuah algoritma untuk menghasilkan suatu permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak suatu himpunan tersebut. Jika diimplementasikan dengan benar, maka hasil dari algoritma ini tidak akan berat sebelah, sehingga setiap permutasi memiliki

kemungkinan yang sama. Metode dasar yang digunakan untuk menghasilkan suatu permutasi acak untuk angka 1 sampai N adalah sebagai berikut:

a. Tuliskan angka dari 1 sampai N.
b. Pilih sebuah angka acak K diantara 1 sampai dengan jumlah angka yang belum dicoret.
c. Dihitung dari bawah, coret angka K yang belum dicoret, dan tuliskan angka tersebut di lain tempat.
d. Ulangi langkah 2 dan langkah 3 sampai semuat angka sudah tercoret.
e. Urutan angka yang dituliskan pada langkah 3 adalah permutasi acak dari angka awal.

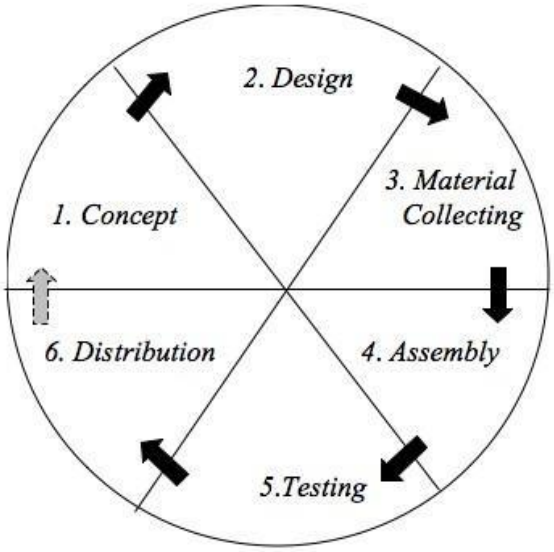
2.3 Metode Pengembangan Multimedia

Selanjutnya untuk metodologi pengembangan sistem mengacu pada Metode Pengembangan Multimedia Versi LutherSutopo yang dikemukakan oleh Ariesto Hadi Sutopo (2003). Menurut Luther (1994), metodologi pengembangan multimedia terdiri dari enam tahap, yaitu:

1. Konsep *(Concept)*
2. Desain *(Design)*
3. Pengumpulan Materi *(Material collecting)*
4. Pembuatan *(Assembly)*
5. Pengujian *(Testing)*
6. Pendistribusian *(Distribution)*

Keenam tahap ini tidak harus berurutan dalam praktiknya, tahap-tahap tersebut dapat saling bertukar posisi. Meskipun begitu, tahap *concept* memang harus menjadi hal yang pertama kali dikerjakan.

Sutopo (2003) mengadopsi metodologi Luther dengan modifikasi, seperti yang terlihat pada Gambar 2.1



Gambar 2.1 Metodologi Pengembangan Multimedia (Sutopo, 2003)

2.3.1 Konsep *(Concept)*

Tahap konsep *(Concept)* adalah tahap untuk menentukan tujuan dan siapa pengguna program (identifikasi *audience*). Selain itu menentukan macam aplikasi (presentasi, interaktif, dll) dan tujuan aplikasi (hiburan, pelatihan, pembelajaran, dll).

2.3.2 Desain *(Design)*

Tahap desain *(Design)* adalah tahap membuat spesifikasi mengenai arsitektur program, gaya, tampilan dan kebutuhan material/bahan untuk program.

2.3.3 Pengumpulan Materi *(Material collecting)*

Tahap Pengumpulan Materi *(Material collecting)* adalah tahap dimana pengumpulan bahan yang sesuai dengan kebutuhan dilakukan. Tahap ini dapat dikerjakan paralel dengan tahap *assembly*. Pada beberap kasus, tahap *Material Collecting* dan tahap *Assembly* akan dikerjakan secara linear tidak paralel.

2.3.4 Pembuatan *(Assembly)*

Tahap *assembly* (pembuatan) adalah tahap dimana semua objek atau bahan multimedia dibuat. Pembuatan aplikasi didasarkan pada tahap *design*.

2.3.5 Pengujian *(Testing)*

Dilakukan setelah selesai tahap pembuatan (*assembly*) dengan menjalankan aplikasi/program dan dilihat apakah ada kesalahan atau tidak. Tahap ini disebut juga sebagai tahap pengujian *alpha* (*alpha test*) dimana pengujian dilakukan oleh pembuat atau lingkungan pembuatnya sendiri.

2.3.6 Pendistribusian *(Distribution)*

Tahapan dimana aplikasi disimpan dalam suatu media penyimpanan. Pada tahap ini jika media penyimpanan tidak cukup untuk menampung aplikasinya, maka dilakukan kompresi terhadap aplikasi tersebut.

2.4 Murai Batu (*Copsychus Malabaricus*)

Menurut Mua'rif (2012), burung muraii batu (*Copsychus malabaricus*) secara luas dikenal sebagai salah satu burung berkicau yang termasuk dalam famili Turdidae. Burung ini memiliki kicauan dengan suara merdu, bermelodi, dan sangat bervariasi (Forum Agri, 2012). Burung murai batu juga dikenal dengan nama yang berbeda-beda dan biasanya diberi nama berdasarkan daerah asal. Misalnya burung murai batu Medan, burung murai batu Aceh dan burung murai batu Kalimantan. Ditambahkan oleh Mua'rif (2012) bahwa burung famili Turdidae pada umumnya memiliki pola warna yang beragam dan menarik. Ukuran tubuhnya ratarata sedang, kepalanya bulat, kakinya agak panjang, paruhnya runcing dan ramping, serta sayapnya lebar. Hampir semua jenis burung yang termasuk dalam familia Turdidae merupakan burung peniru kicauan burung lain dan mempunyai kicauan yang bagus.

2.5 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).

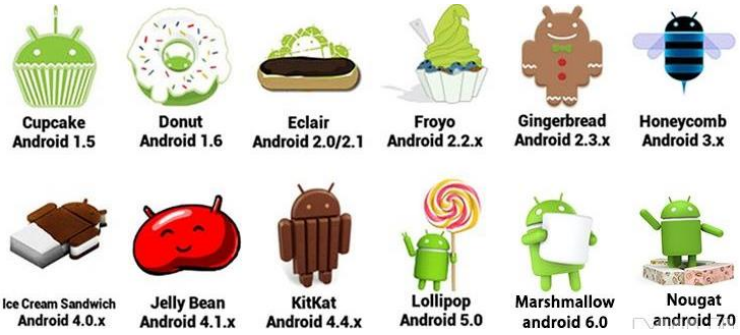
2.5.1 Versi Android

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 2.2 ***Evolusi Android.***

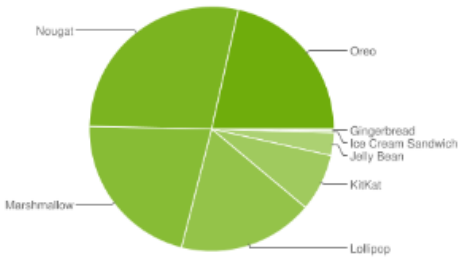
2.5.2 Statistik Distribusi OS Android

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 Android5.1 *Lollipop* dengan *persentase* 14.4% yang dususul oleh Android 8.0 *Oreo* dengan presentase 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, Andorid 4.4 *Kitkat* dengan presentase 7.6%. Selanjtnya Android 8.1 *Oreo* dengan *presentase* 7.5% dan android versi 5.0 *Lollipop* sudah mulai ditinggalkan pengguna Android karena hanya memiliki persentase 3.5% dam umtuk urutan di bawah nya 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 *presentase* 0.3 % di urutan terakhir yaitu android versi 2.3.3-2.3.7 *Ginger bread* dengan *presentase* 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.

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.1%
4.2.x		17	1.5%
4.3		18	0.4%
4.4	KitKat	19	7.6%
5.0	Lollipop	21	3.5%
5.1		22	14.4%
6.0	Marshmallow	23	21.3%
7.0	Nougat	24	18.1%
7.1		25	10.1%
8.0	Oreo	26	14.0%
8.1		27	7.5%



Gambar 2.3 Statistik distribusi OS Android (26 Oktober 2018)

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

2.6 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 tanpa5. 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

2.6.1 Pengenalan Layout

Membicarakan masalah tampilan aau 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.

2.6.2 Barisan atau Struktural Folder

Pada bagian folder build sepertti 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 difolder 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.

2.6.3 Grandle

Salah satu fitur teranyar pada Android Studio adalah fitur gradle, grandle 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.

2.6.4 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 Programing 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.

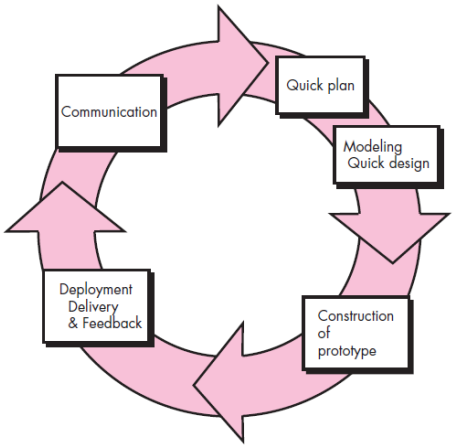
2.7 Model Prototype

Metode *Prototype* adalah proses pembuatan model sederhana *software* yang mengijinkan 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 perencaan 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 Delevery and Feedback*), yaitu memproduksi perangkat secara benar sehingga dapat digunakan oleh pengguna.



Gambar 2.4 *Diagram Prototype*

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

2.8 Sistem Pemodelan

2.8.1 *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan gambar untuk menvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek. *Unified Modeling Language* (UML) bukanlah merupakan bahasa pemprograman tetapi model-model yang tercipta berhubungan langsung dengan berbagai macam bahasa pemprograman berorientasi obyek, seperti Java (Syafitri 2016). UML tersusun atas sejumlah elemen grafis membentuk diagram-diagram. Dalam penelitian ini melakukan desain hanya 2 diagram yaitu *Use Case* Diagram dan Activity Diagram..

2.8.2 *Use Case* Diagram

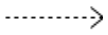
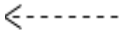
Munawar (2018 : 89) menguraikan bahwa *Use Case* adalah deskripsi fungsi sebuah system dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah system dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah system dipakai. Urutan langkah-langkah yang

menerangkan antara pengguna dan system disebut sebagai *Scenario*. Setiap *Scenario* menggambarkan urutan kejadian. Setiap urutan di inisialisasi oleh orang, system yang lain, perangkat keras atau urutan waktu. Dengan demikian, secara singkat bias dikatakan *Use Case* adalah serangkaian *Scenario* yang digabungkan bersaa-sama oleh tujuan umum pengguna.

Use Case dibuat berdasarkan kebutuhan Aktor. *Use Case* harus merupakan 'apa' yang dikerjakan software aplikasi, bukan 'bagaimana' software aplikasi mengerjakannya.

Tabel 2.1 pada halaman berikut ini adalah Simbol-simbol yang digunakan dalam *Use Case* Diagram :

Tabel 2.1 Simbol *Use Case* Diagram.

GAMBAR	NAMA	KETERANGAN
	*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
	*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
	*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
	*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
	Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.

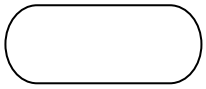
Tabel 2.1 (lanjutan)

	*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
	*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

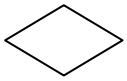
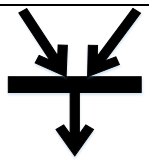
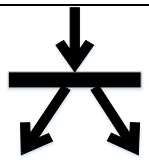
2.8.3 *Activity* Diagram

Munawar (2018 : 127) mengurakan bahwa *Activity Diagram* adalah bagian penting dari UML yang menggambarkan aspek dinamis dari Sistem. Logika Prosedural, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah di deskripsikan dalam *Activity Diagram. Activity Diagram* mempunyai peran seperti halnya *Flowchart*, akan tetapi perbedaannya dengan *Flowchart* adalah `; *Activity Diagram* bisa mendukung perilaku paralel sedangkan *Flowcharti* tidak bisa. Simbol-simbol yang digunakan dalam *Activity* diagram dapat dilihat pada tabel 2.2 dibawah ini :

Tabel 2.2 Simbol *Activity* Diagram.

Simbol	Keterangan
	*Activity* : Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	*Initial Node* : Bagaimana objek dibentuk atau diawali
	*Actifity Final Node* : Bagaimana objek dibentuk dan diakhiri.

Tabel 2.2 (lanjutan)

	*Decision* : Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.
	*Swimlane* : Memisalkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.
	*Join* : Digunakan untuk menunjukkan kegiatan yang digabungkan.
	*Fork* : Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel

2.9 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-BoxTesting*. Sebaliknya, *Black-Box Testing* adalah pendekatan komplementer yang mungkin untuk mengungkap kelas yang berbeda dari kesalahan daripada metode *White-Box Testing*.

Black-BoxTesting 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.

2.10 Penelitian Terkait

Dalam penyusunan skripsi ini, penulis terinspirasi dan mereferensi dari penelitian-penelitian sebelumnya yang berkaitan dengan skripsi ini. Daftar penelitian terkait sebagai berikut :

Tabel 2.3 Penelitian Terkait

No.	Nama	Judul	Tahun	Uraian
1.	Beki Subaeki, Dieky Ardiansyah.	Implementasi Algoritma Fisher Yates Shuffel Pada AplikasiMultimedia Interaktif Untuk Pembelajaran Tenses Bahasa Inggris	2017	Penelitian ini membahas mengenai media penyampian informasi mengenai materi tenses dibuat dalam bentuk PBK (Pembelajaran Berbantuan Komputer), yakni Multimedia Interaktif. Multimedia Interaktif dapat memberikan sebuah interaktifitas antara pengguna dengan aplikasiß.

2.	Indri Ayuningtyas, Muhammad Arief Fadhilah, Rita Wahyuni Arifin	Media Pembelajaran Mengenal Hewan Dalam Bahasa Inggris Berbasis Multimedia Interaktif	2018	Tujuan dari perancangan media pembelajaran ini adalah membantu para guru dan murid khususnya anak sekolah dasar usia enam dan tujuh tahun dalam memahami pengenalan dunia satwa dalam bahasa inggris.
3.	Adi Fitria Andikos	Perancangan Aplikasi Multimedia Interaktif Sebagai Media Pembelajaran Pengenalan Hewan Pada TK Islam Bakti 113 Kota Salak	2019	Aplikasi ini bertujuan untuk membantu anak TK untuk mengenalkan hewan,