

BAB II

TINJAUAN PUSTAKA

2.1 Rancang Bangun

(Harsono, 2012) Rancang bangun adalah suatu istilah umum untuk membuat atau mendesain suatu objek dari awal pembuatan sampai akhir pembuatan. Rancang bangun berawal dari kata desain yang artinya perancangan, rancang, desain, bangun. Sedangkan merancang artinya mengatur, cara, perbuatan merancang. Dapat disimpulkan arti kata desain adalah proses, cara, perbuatan dengan mengatur segala sesuatu sebelum bertindak atau merancang.

2.2 Paritta Suci

(Sa gha Therav da Indonesia, 2005) Paritta (P li) atau paritrana (Sanskerta) dan pirit atau pirith menurut pengucapannya dalam bahasa singhala pada pokoknya bermakna ‘perlindungan’. Paritta adalah khotbah sang Buddha yang merupakan suatu perlindungan yang kuat bilamana bisa dihapal. Pada saat mengulangi paritta, pikiran harus ditunjukkan, dipusatkan pada makna paritta tersebut sehingga pada saat itu kesadaran (sati-sampajañña) menjadi kuat, pikiran (citta) bersatu dengan kebajikan, bersih dari kilesa, sebaliknya penuh dengan cinta kasih (mett) dan kebenaran dhamma (sacca). Jadi mengulangi paritta merupakan kondisi untuk mempercepat masakannya buah kamma baik (kusala-kamma-vip ka) yang telah dibuat, sedangkan buah kamma tidak baik (akusala-kamma-vip ka) terhambat masakannya, terkecuali akusala-garuka-kamma-vip ka. Inilah yang dimaksudkan dengan perlindungan dalam paritta.

2.3 Aplikasi

(Faruq, 2015) menguraikan bahwa, Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah yang dieksekusi oleh komputer. Program merupakan kumpulan *instruction set* yang akan dijalankan oleh pemroses, yaitu berupa *software*. Bagaimana

sebuah sistem komputer berpikir diatur oleh program ini. Program inilah yang mengendalikan semua aktivitas yang ada pada proses. Program berisi konstruksi logika yang dibuat manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada *instruction set*. Program aplikasi merupakan program siap pakai. Program direkam untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi akan menggunakan *Operating System* (OS) komputer dan aplikasi yang mendukung.

2.4 Android

(Satyaputra, 2014) *Android* adalah istilah dalam bahasa Inggris yang berarti “Robot yang menyerupai manusia”. Logo *android* sendiri dicerminkan seperti sebuah robot berwarna hijau, yang mengacu kepada arti kata *Android*. *Android* adalah sebuah sistem operasi untuk *smartphone* dan *tablet*. Sistem operasi dapat diilustrasikan sebagai ‘jembatan’ antara piranti (*device*) dan penggunaannya, sehingga pengguna bias berinteraksi dengan *device*-nya dan menjalankan aplikasi – aplikasi yang tersedia pada *device*.

Android adalah sistem operasi yang bersifat *open source* (sumber terbuka). Disebut *open source* karna *source code* (kode sumber) dari sistem operasi android dapat dilihat, di-*download*, dan dimodifikasi secara bebas. Paradigma *open source* ini memudahkan pengembangan teknologi android, karena semua pihak yang tertarik dapat memberikan kontribusi, baik pada pengembangan sistem operasi maupun aplikasi. Pada awal oktober 2013, tercatat ada lebih dari 850,000 aplikasi android yang tersedia di google play (dulu bernama android market).

2.4.1 Versi dan Jenis – Jenis Android

Pengembangan Android dimulai dengan berdirinya Android, Inc. Pada Oktober 2003 dengan tujuan membuat *mobile device* yang lebih *smart* untuk menyaingi *Symbian* dan *Windows Mobile* yang populer saat itu (iPhone dan BlackBerry belum diliris). Pada tahun 2005, Android Inc. Diakuisisi oleh google. Pengembangan terus dilanjutkan sampai android versi beta diluncurkan pada tanggal 5 November 2007

bersamaan dengan berdirinya OHA (Open Handset Alliance).
Dijelaskan pada tabel 2.2 berikut ini :

Tabel 2.1 Versi dan jenis – jenis Android

Versi	Nama	Rilisan	Catatan
1.0	-	23 September 2008	Android pertama, hanya untuk <i>smartphone</i>
1.1	-	9 Febuari 2009	
1.5	<i>Cupcake</i>	30 April 2009	Mulai pakai kode nama
1.6	<i>Donut</i>	15 September 2009	ukuran layar yang lebih besar, dan diberi versi awal fitur
2.0-2.1	<i>Eclair</i>	26 Oktober 2009 (2.0) 12 Januari 2010 (2.1)	menambahkan fitur Bluetooth 2.1, flash dan kamera dengan digital zoom, multi-touch, live wallpaper
2.2	<i>Frozen Yogurt</i>	20 Mei 2010	Jenis <i>smartphone</i> lama
2.3	<i>Gingerbread</i>	6 Desember 2010	Memperbaharui desain antarmuka pengguna dengan meningkatkan kecepatan dan kesederhanaan.
3.0-3.2	<i>Honeycomb</i>	22 Februari 2011 (3.0) 10 Mei 2011 (3.1) 15 Juli 2011 (3.2)	Hanya untuk tablet
4.0	ICS (<i>Ice cream sandwich</i>)	19 Oktober 2011	<i>Smartphone</i> dan Tablet
4.1-4.3	<i>Jelly Bean</i>	19 Juli 2012 (4.1) 13 November 2012 (4.2) 24 Juli 2013 (4.3)	Merupakan versi android dengan antarmuka pengguna yang lebih halus dan cepat.
4.4	<i>KitKat</i>	31 Oktober 2013	Merupakan versi lanjutan dari versi sebelumnya “ <i>jellybean</i> ” <i>KitKat</i> sendiri merupakan nama komersial dari sebuah produk coklat.
5.0	<i>Lollipop</i>	15 Oktober 2014	-
6.0	<i>Marshmallow</i>	-	-

2.4.2 Keunggulan Android

Aplikasi android sangat berperan penting dalam smartphone, maka dari itu ada beberapa keunggulan android yaitu:

a. *User Friendly*

Pada sistem operasi windows milik Microsoft, ibarat kita dengan sangat mudah mengoperasikan komputer hanya dengan belajar beberapa hari bahkan beberapa jam saja pada android yang berjalan pada smartphone.

b. *Notifications*

Anda dengan sangat mudah mendapatkan notifikasi dari smartphone android dengan mengatur beberapa akun E-mail, SMS, *Voice Dial*, Update dan lain sebagainya.

c. Tampilan

Dari segi tampilan, android tidak kalah bagusnya dari iOS milik Apple, karena memang dari awal android hampir mengusung teknologi iOS.

d. *Open Source*

Operating system yang memang dibuat open source oleh penciptannya, karena memang berbasis *Kernel Linux*.

e. Aplikasi

Untuk aplikasi anda akan disajikan jutaan pilihan aplikasi yang menarik dari yang gratis hingga prabayar.

2.5 Adobe Flash Professional CS6

Menurut (Maulana I, 2014) pada dasarnya baik *Flash* versi *Macromedia* maupun versi *Adobe*, area kerjanya terbagi menjadi beberapa bagian, di antaranya menu, *toolbox*, *timeline*, *stage*, dan *panel*.

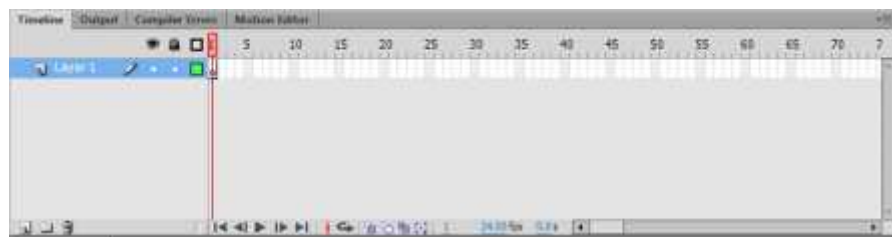
2.5.1 *Toolbox*

Di dalam *toolbox* terdapat macam-macam *tool* yang bisa kita gunakan untuk menggambar objek. *Toolbox* ini terdiri dari beberapa bagian yaitu *selection tool*, *drawing tool*, *painting tool*, dan *navigation tool*. Seperti pada gambar 2.1 berikut:

Gambar 2.1 *Toolbox*

2.5.2 *Timeline*

Timeline pada *Adobe Flash CS6* secara *default* terletak diatas *stage*. *Timeline* berfungsi untuk mengontrol keseluruhan objek dan animasi yang terdapat pada *stage*, seperti pada gambar 2.2 berikut :

Gambar 2.2 *Timeline*

2.5.3 *Layer*

Layer dapat diilustrasikan seperti tumpukan-tumpukan yang berisi objek di dalamnya. Dengan *layer*, kita bisa mengatur *movie* pada *stage* dan menentukan kedalaman atau lapisan suatu objek, seperti pada gambar 2.3 berikut :

Gambar 2.3 *Layer*

Untuk menambahkan *layer* baru, kita dapat memilih menu *Insert > Timeline > Layer* atau dengan mengklik *icon* yang terdapat pada bagian bawah *timeline*.

Terdapat beberapa mode yang dapat dipilih pada *layer* di antaranya :

- *Guide layer*, berfungsi untuk membuat animasi dengan menggunakan jalur atau *track* berupa garis yang telah dibuat.
- *Motion layer*, digunakan untuk membuat pergerakan animasi *tween*.
- *Masking layer*, digunakan untuk membuat animasi efek *masking*.

2.5.4 *Frame*

Frame merupakan bagian dari *layer* yang digunakan untuk mengatur pergerakan animasi. Di dalam *frame* bisa terdiri dari teks, gambar, audio, video, dan kode program *ActionScript*, seperti pada gambar 2.4 berikut :



Gambar 2.4 *Frame*

Frame terdiri dari beberapa bentuk, di antaranya :

- *Keyframe*, *frame* yang memiliki bentuk bulatan hitam, menandakan bahwa di dalamnya terdapat objek.
- *Blank keyframe*, *frame* berbentuk bulat putih, menandakan bahwa *frame* masih kosong.
- *Action frame*, *frame* dengan bulatan putih dan terdapat huruf 'a', menandakan di dalamnya terdapat kode program *ActionScript*.
- *Frame label*, yaitu *frame* yang diberi nama *label* dan ditandai dengan bendera warna merah.

2.5.5 Stage

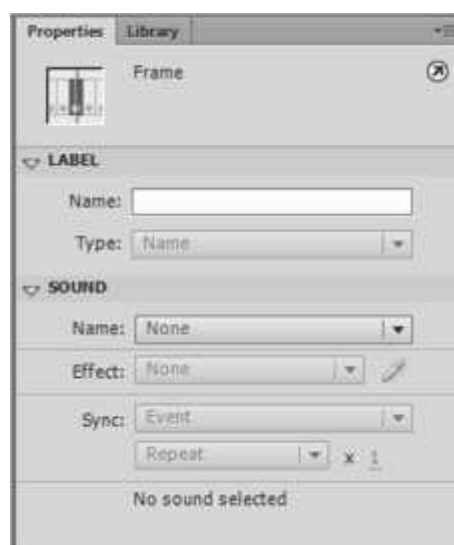
Stage adalah area putih berbentuk kotak yang terletak ditengah area kerja *Flash*. Seperti istilah *stage* pada panggung teater, *stage* di dalam *Flash* berfungsi untuk menampilkan semua objek maupun *movie* yang berjalan di atasnya. Objek yang diletakan pada *stage* bisa berupa teks, gambar, dan video, seperti pada gambar 2.5 berikut :



Gambar 2.5 Stage

2.5.6 Panel Properties

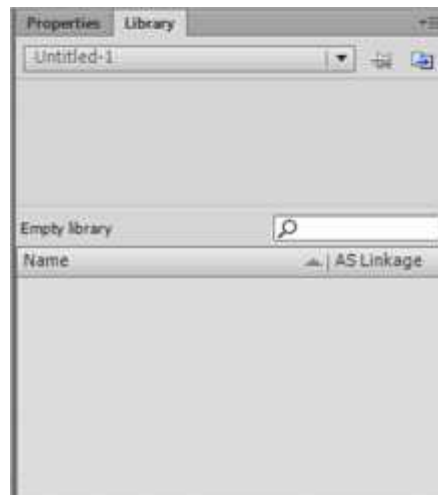
Panel ini menampung semua properti yang terdapat pada *tool-tool* dalam *Flash*. Untuk mengaktifkannya, pilih menu *Windows > Properties*, seperti pada gambar 2.6 berikut :



Gambar 2.6 Panel Properties

2.5.7 Panel Library

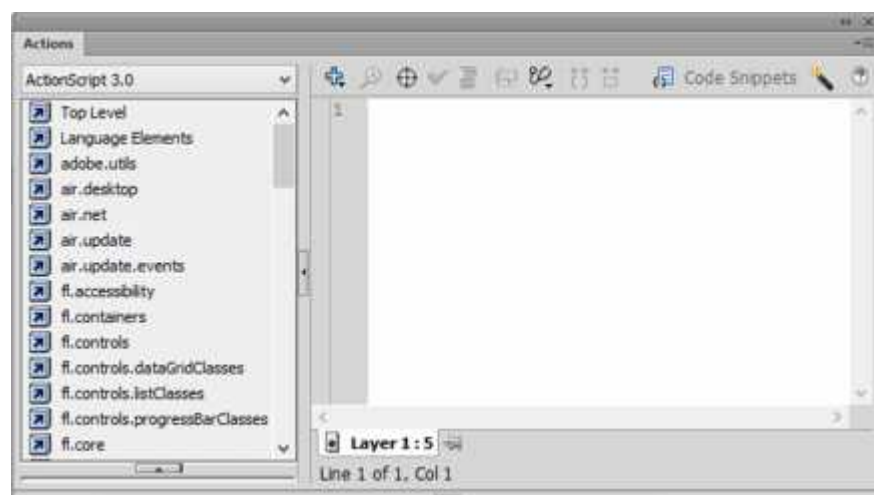
Seperti layaknya perpustakaan, *library* pada *Flash* berfungsi menampung semua simbol yang telah dibuat maupun hasil impor dari file luar seperti audio, video, gambar, dll. Untuk menampilkan *panel library*, bisa melalui menu *Window > Library* atau dengan menekan tombol CTRL+L, seperti pada gambar 2.7 berikut :



Gambar 2.7 Panel Library

2.5.8 Panel Action

Panel ini berfungsi sebagai tempat menuliskan *ActionScript*. Untuk mengaktifkannya, pilih menu *Window > Action* atau menekan tombol F9, seperti pada gambar 2.8 berikut :



Gambar 2.8 Panel Action

2.6 *ActionScript 3.0*

Menurut (Maulana I, 2014) *ActionScript 3.0* atau disingkat AS3 merupakan bahasa pemrograman yang bekerja pada *Adobe Flash*, *Flex*, dan *FlashDevelop*. *ActionScript 3.0* pertama kali dirilis pada tahun 2006 bersamaan dengan diluncurkannya *Flash* versi 9 sekaligus *Flash* pertama yang kini telah diakuisisi oleh *Adobe System Inc.* yaitu *Adobe Flash CS3*.

ActionScript 3.0 adalah bahasa pemrograman yang didasarkan pada *ECMAScript*, yaitu standar bahasa pemrograman yang dikembangkan oleh *ECMA (European Computer Manufacturers Association)*. Dengan standar ini, AS3 mampu melakukan integrasi data yang cepat dengan berbagai bahasa pemrograman lainnya seperti *JavaScript* dan *XML*.

2.7 *Adobe Photoshop CS*

(Hartono D.S, 2013), *Adobe Photoshop* adalah perangkat lunak *editor* citra buatan *Adobe System* yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek. Perangkat ini banyak digunakan oleh fotografer digital dan perusahaan iklan sehingga dianggap sebagai pemimpin besar (*market leader*) untuk perangkat lunak pengolah gambar/foto, dan bersama *Adobe Acrobat*, dianggap sebagai produk terbaik yang pernah diproduksi oleh *Adobe System*. Versi kedelapan aplikasi ini disebut dengan nama *Photoshop CS (Creative Suite)*, versi sembilan disebut *Adobe Photoshop CS2*, versi kesepuluh disebut *Adobe Photoshop CS3*, versi kesebelas disebut *Adobe Photoshop CS4*, versi keduabelas disebut *Adobe Photoshop CS5*, dan versi yang terakhir (ketigabelas) adalah *Adobe Photoshop CS6*.

2.8 *Adobe AIR*

(Seno, 2014) Menguraikan bahwa, *Adobe Air (Adobe Integrated Runtime)* adalah sebuah *cross operating system runtime* yang dikembangkan oleh **Adobe** sehingga memungkinkan pengembang memanfaatkan ketrampilan mereka (seperti *Flash*, *Flex*, *HTML*, *Javascript* dan *PDF*) untuk membangun RIA (*Rich Internet Application*) dan *contenya* kedalam *platform* baru.



Gambar 2.9 *Adobe AIR Setup*

2.9 Goldwave

(Suyanto M, 2005) merupakan perangkat lunak editing audio yang sederhana yang mempunyai kemampuan trimming, denoising atau normalizing.

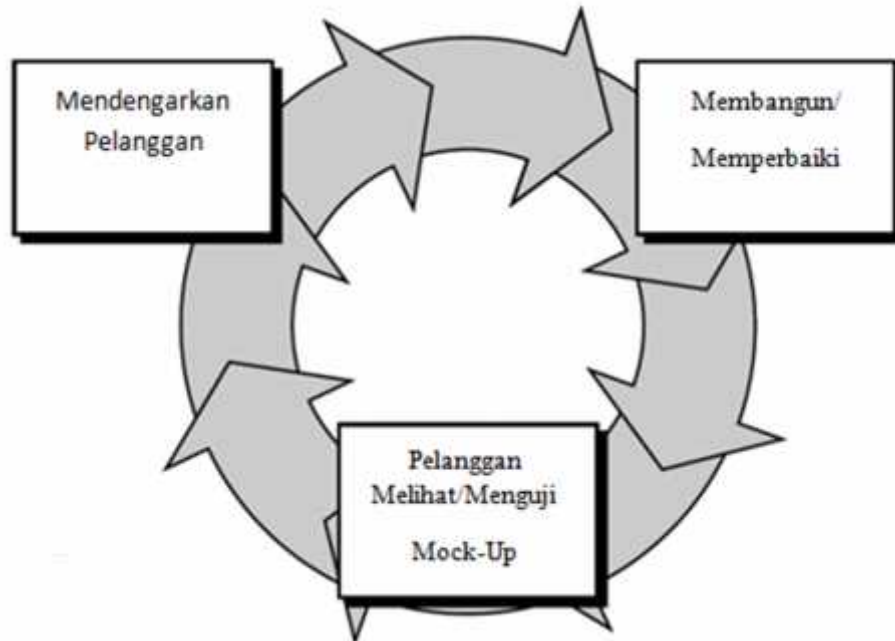


Gambar 2.10 *Device Controls Goldwave*

2.10 Metode Prototype

(Rosa A.S dan M. Shalahuddin, 2015) Menguraikan bahwa, Model prototipe (*prototyping model*) dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program prototipe biasanya merupakan program yang belum jadi. Program ini biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak

yang sudah jadi. Program prototipe ini dievaluasi oleh pelanggan atau user sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau user. Berikut adalah gambar dari model prototipe :



Gambar 2.11 Model Prototipe

Mock-up adalah sesuatu yang digunakan sebagai model desain yang digunakan untuk mengajar, demonstrasi, evaluasi desain, promosi, atau keperluan lain. Sebuah *mock-up* disebut sebagai prototipe perangkat lunak jika menyediakan atau

mampu mendemonstrasikan sebagian besar fungsi sistem perangkat lunak dan memungkinkan pengujian desain sistem perangkat lunak. Iterasi terjadi pada pembuatan prototipe sampai sesuai dengan keinginan pelanggan (*customer*) atau *user*. Proses pada model *prototyping* dapat dijelaskan sebagai berikut:

- a. Pengumpulan kebutuhan : *Developer* dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagiannya yang akan dibutuhkan berikutnya.
- b. Perancangan : Perancangan dilakukan cepat dan rancangan mewakili semua aspek perangkat lunak yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.

- c. Evaluasi *prototype* : Klien mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan perangkat lunak. Perulangan ketiga proses ini terus berlangsung hingga semua kebutuhan terpenuhi. *Prototype-prototype* dibuat untuk memuaskan kebutuhan klien dan untuk membangun perangkat lunak lebih cepat, namun tidak semua *prototype* bisa dimanfaatkan. Demi kebutuhan klien lebih baik *prototype* yang dibuat diusahakan dapat dimanfaatkan.

2.11 Unified Modeling Language (UML)

2.11.1 Pengertian UML

(Haviluddin, 2011) Menguraikan bahwa, *Unified Modeling Language* (UML), adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan system secara visual juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah system software yang terkait dengan objek. Sejarah UML sendiri terbagi dalam dua fase sebelum dan sesudah munculnya UML. Dalam fase sebelum, UML sebenarnya sudah mulai diperkenalkan sejak 1990an namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

2.11.2 Diagram UML

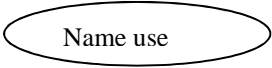
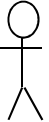

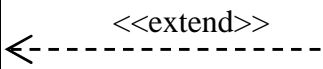
Model-model diagram dapat dikelompokkan berdasarkan sifatnya yaitu statis dan dinamis. Jenis diagram tersebut antara lain:

1. Diagram *Use case*

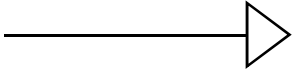
(Haviluddin, 2011) Menguraikan bahwa, *Use Case Diagram* merupakan *diagram* yang menggambarkan *actor* (pengguna atau sistem lain), *use case* (deskripsi fungsi dari sebuah sistem) dan relasinya. *Use Case* adalah abstraksi dan interaksi antara sistem

dan actor. *Use Case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use Case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata user. Sedangkan *Use Case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client.

Tabel 2.2 Simbol *Use Case* Diagram

Simbol	Keterangan
<p data-bbox="483 734 609 768">Use Case</p>  <p data-bbox="443 813 555 835">Name use</p>	<p data-bbox="751 741 1305 920">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya akan diterangkan dengan menggunakan kata kerja di awal frase nama use case.</p>
<p data-bbox="459 972 635 1005">Aktor/Actor</p>  <p data-bbox="467 1193 627 1227">Nama aktor</p>	<p data-bbox="751 972 1305 1189">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi simbol aktor belum tentu menunjukkan orang, biasanya akan dinyatakan menggunakan kata benda diawal frase nama aktor.</p>
<p data-bbox="403 1256 687 1290">Asosiasi/Association</p> 	<p data-bbox="751 1256 1305 1357">Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p data-bbox="435 1442 659 1476">Ekstensi/Extend</p>  <p data-bbox="467 1518 627 1541"><<extend>></p>	<p data-bbox="751 1442 1305 1547">Case tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya</p>




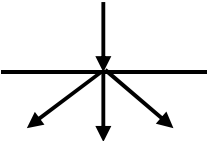
Tabel 2.2 Simbol *Use Case Diagram* (Lanjutan)

Simbol	Keterangan
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

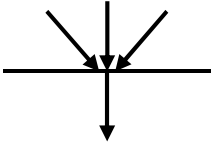

2. Activity Diagram

(Haviluddin, 2011) Menguraikan bahwa, *Activity diagram* menggambarkan kegiatan diagram alur kerja atau aktivitas dari sistem. *Activity diagram* menggambarkan aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana aktivitas itu berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi yang dijalankan dari sebuah sistem.

Tabel 2.3 Simbol Diagram Activity

Simbol	Keterangan
Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal.
Aktivitas 	Aktivitas yang dilakukan sistem. Biasanya diawali dengan kata kerja.
Percabangan/ <i>Decision</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas.
Fork 	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel.

Tabel 2.3 Simbol Diagram Activity (Lanjutan)


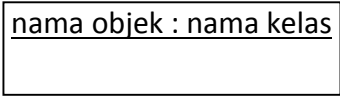

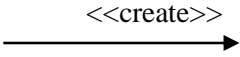
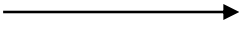
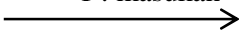
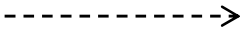
Simbol	Keterangan
Penggabungan/ <i>Join</i> 	Digunakan untuk menunjukkan kegiatan yang digabungkan.
<i>End Point</i> 	Mengakhiri aktivitas sistem.

3. Sequence Diagram

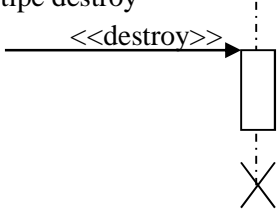
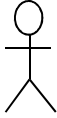
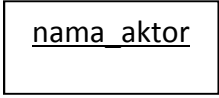
(Rosa A.S dan M. Shalahuddin, 2015) diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat scenario yang ada pada *use case*. Banyaknya diagram *sequen* yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

(Rosa A.S dan M. Shalahuddin, 2015) simbol-simbol yang ada pada diagram sekuen dapat dilihat pada Tabel 2.4 berikut.

Tabel 2.4 Simbol *Sequence Diagram*

Simbol	Keterangan
Garis hidup/ <i>lifeline</i> 	Menyatakan kehidupan suatu objek.
Objek 	Menyatakan objek yang berinteraksi pesan.
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
Pesan tipe create 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe call 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
Pesan tipe send 1 : masukan 	Merupakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe return 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

Tabel 2.4 Simbol *Sequence Diagram* (Lanjutan)

Simbol	Keterangan
Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhirkan sebaliknya jika ada create maka ada destroy.
Aktor  atau 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.

Penomoran pesan berdasarkan urutan interaksi pesan. Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu. Semua metode di dalam kelas harus ada di dalam diagram kolaborasi atau sekuen, jika tidak ada berarti perancangan metode di dalam kelas itu kurang baik. Hal ini dikarenakan ada metode yang tidak dapat dipertanggungjawabkan kegunaannya.

2.12 Penelitian Terdahulu

Table 2.5 berikut ini merupakan penelitian terdahulu yang menerapkan system aplikasi berbasis android dan Paritta Suci.

Table 2.5 Penelitian Terdahulu

No.	Nama	Judul	Terbit/Tahun	Abstrak
1.	Hendry 1) Eklianti 2)	Aplikasi Pembelajaran Paritta Suci Pada Windows Phone	STMIK IBBI Medan/2012	Indonesia yang terdiri banyak suku dan beraneka ragam budaya juga terdapat beraneka ragam agama. Dalam hal ini, setiap agama juga berbeda dalam hal tata cara persembahan terhadap ajaran agamanya. Untuk itu, agar penganut agama bisa lebih memahami pembacaan ayat-ayat suci atau paritta maka perlu dirancang sebuah perangkat lunak untuk pembelajaran yang diaplikasikan kepada teknologi komunikasi. Hasil dari perangkat lunak ini, untuk memudahkan umat atau penganut agama terutama agama Buddha dalam menguasai pembacaan paritta, mengerti arti dari paritta yang dibaca atau didengarkan.
2.	I Dewa Putu Laksana Kariana	Rancang Bangun Aplikasi Kitab Virtual Bhagawadgita Berdasarkan Terjemahan Ida Bagus Mantra Berbasis Android	Sistem Komputer, STMIK STIKOM Bali/2014	Kitab berarti sebuah teks atau tulisan yang dijilid menjadi satu. Weda merupakan kumpulan sastra-sastra kuno dari zaman India Kuno yang jumlahnya sangat banyak dan luas, salah satunya adalah Kitab Bhagawadgita. Saat ini mobile phone digunakan juga sebagai media yang digunakan untuk menambah wawasan akan ilmu pengetahuan.. Kitab Virtual atau Digital berfungsi sebagai media yang digunakan untuk memberi informasi berdasarkan kitab aslinya