

BAB II TINJAUAN PUSTAKA

2.1 Perancangan

Menurut (Rosa & Shalahudin, 2014), Desain atau perancangan sistem dalam pembangunan perangkat lunak merupakan upaya untuk mengkonstruksi sebuah sistem yang memberikan (kepuasan informal) akan spesifikasi kebutuhan *funksional*, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain segi biaya, waktu, dan perangkat lunak.

2.2 *Augmented Reality*

Menurut (Arifitama, 2017) *Augmented Reality* adalah kombinasi antara dunia maya (*virtual*) dan dunia nyata (*real*) yang dibuat oleh komputer. *Augmented Reality* dapat diaplikasikan untuk semua indera, termasuk pendengaran, dan sentuhan. Selain digunakan dalam bidang-bidang seperti kesehatan, militer, industri manufaktur, *Augmented Reality* juga telah diaplikasikan dalam perangkat-perangkat yang digunakan orang banyak, seperti pada telepon genggam.

Augmented Reality dapat diklasifikasikan menjadi dua berdasarkan adanya tindakan pengguna marker yaitu:

a. *Marker Augmented Reality*

Sebuah metode yang memanfaatkan marker yang biasanya berupa ilustrasi hitam putih berbentuk persegi atau lainnya dengan batasan hitam tebal dan latar belakang putih. Melalui posisi yang diharapkan pada sebuah kamera komputer atau *smartphone*, maka komputer atau *smartphone* akan melakukan proses menciptakan dunia *virtual 2D* atau *3D*.

b. *Markerless Augmented Reality*

Salah satu metode *augmented reality* yang saat ini sedang berkembang adalah metode *markerless augmented reality*. Dengan metode ini pengguna tidak perlu lagi menggunakan sebuah *marker* untuk menampilkan objek 3D atau yang lainnya. Sekalipun dinamakan dengan *markerless* namun aplikasi tetap berjalan dengan melakukan pemindaian terhadap objek, namun ruang lingkup yang akan dipindai lebih luas dibandingkan dengan *marker based tracking*.

2.3 Profesi

Profesi adalah suatu jabatan atau pekerjaan yang menuntut keahlian tertentu. artinya, jabatan profesional tidak bisa dilakukan atau dipegang oleh sembarang orang yang tidak terlatih dan tidak disiapkan secara khusus untuk melakukan pekerjaan tersebut. melainkan melalui proses pendidikan dan pelatihan yang disiapkan secara khusus untuk bidang tertentu. pada umumnya pekerjaan dibagi menjadi dua berupa barang dan jasa. contoh pekerjaan yang menghasilkan barang yaitu Desainer, Perakit Robot, Pengrajin Furniture. Pekerjaan yang menghasilkan jasa yaitu Guru, Polisi, Tentara dan Dokter.

Menurut (Hanafri, Ramdhan, & Nisa, 2017) Profesi merupakan suatu pekerjaan yang menuntut pengembangan profesi tersebut untuk terus memperbaharui keterampilannya sesuai dengan perkembangan ilmu dan teknologi, Profesi juga merupakan bagian dari pekerjaan, namun tidak setiap pekerjaan adalah profesi. Profesi merupakan suatu pekerjaan yang mengandalkan keterampilan dan keahlian khusus yang didapatkan melalui pengalaman kerja.

Menurut (Martinis, 2013) profesi mempunyai pengertian seseorang yang menekuni pekerjaan berdasarkan keahlian, kemampuan, teknik, dan prosedur berlandaskan intelektualitas.

2.4 Media Pembelajaran

Pembelajaran anak usia dini memegang peranan yang sangat penting dalam pembentukan kemampuan dan sikap belajar pada tahap yang lebih lanjut. Proses pembelajaran peran guru bukan semata-mata memberikan informasi, melainkan juga mengarahkan, dan memberi fasilitas belajar (*Directing and Facilitating the Learning*) agar proses belajar menjadi lebih efektif.

Menurut (Susanto, 2017), media pembelajaran adalah sarana dan prasarana membawa pesan yang dapat dimanfaatkan untuk keperluan pembelajaran, jadi media pembelajaran adalah alat bantu yang dapat digunakan untuk pembelajaran.

Menurut (Ilmawan Mustaqim., 2016) media pembelajaran adalah segala sesuatu yang dapat digunakan untuk menyalurkan pesan dari pengirim ke penerima sehingga merangsang pikiran, perasaan, perhatian dan minat serta kemauan peserta didik sedemikian rupa sehingga proses belajar terjadi dalam rangka mencapai tujuan pembelajaran secara efektif

Sementara menurut (Arsyad, 2013) dalam dunia pendidikan kata 'media' disebut dengan media pembelajaran. menyampaikan bahwa media pembelajaran adalah segala sesuatu yang dapat digunakan untuk menyampaikan pesan atau informasi dalam proses belajar mengajar sehingga dapat merangsang perhatian dan minat siswa dalam belajar

2.5 Anak Usia Dini

Definisi anak usia dini menurut (Zaman, Pd, Eliyawati, Pd, & Eliyawati, 2010) merupakan anak yang berada dalam prinsip pendidikan anak usia dini adalah individu unik yang memiliki pola pertumbuhan dan perkembangan dalam aspek fisik, kognitif, sosial emosional, kreatifitas, bahasa dan komunikasi yang khusus sesuai dengan tahapan yang sedang dilalui oleh anak tersebut.

Menurut (Susanto, 2017) Cara pembelajaran anak usia dini disesuaikan dengan tingkat perkembangan seiring dengan bertambahnya usia anak. Cara belajar tersebut sebagai berikut.

1. Usia 0 - 1 tahun, anak belajar dengan mengendalikan kemampuan pancaindranya dengan pendengaran, penglihatan, penciuman, perabaan, dan perasa. Secara bertahap, pancaindra anak akan berfungsi lebih sempurna. Pada usia satu tahun, anak ingin mempelajari apa saja yang dilihat dengan mengarahkan seluruh pancaindranya. Hal ini tampak pada aktifitas memasukan segala macam benda ke dalam mulut sebagai bagian dari proses belajar, serta mempelajari komunikasi social.
2. Usia 2 - 3 tahun, anak melakukan proses belajar dengan lebih sungguh-sungguh. Ia memperlihatkan apa saja yang ada dilingkungannya kemudia ditiru. Jadi , cara belajar anak yang utama untuk usia ini adalah meniru. Meniru segala hal yang dilihat dan didengar. Selain itu , perkembangan anak pada usia ini juga sudah mulai berkembang. Anak mengembangkan kemampuan berbahasa dengan cara bermain. Perkembangan emosi anak didasarkan pada respons lingkungan memperlakukan anak. Sebab emosi bukan ditimbulkan oleh bawaan, melainkan lebih banyak karena lingkungan.
3. Usia 4 – 6 tahun, kemampuan bahasa anak akan semakin baik. Anak mampu berkomunikasi dengan baik maka diikuti proses

belajar anak dengan cara bertanya. Anak akan menanyakan segala sesuatu yang dilihat. Dengan kognisi anak berkembang pesat, dan keinginan anak untuk belajar sangat tinggi maka anak belajar melalui bertanya dan berkomunikasi.

Menurut (Wijana, 2014) Anak usia dini merupakan anak terbagi menjadi tiga bagian yaitu kelompok usia bayi hingga dua tahun, kelompok usia tiga hingga lima tahun, dan kelompok enam hingga usia delapan tahun. Pembagian kelompok tersebut dapat mempengaruhi kebijakan penerapan kurikulum dalam pendidikan dan pengasuhan anak.

2.6 *Android*

Menurut (Nazruddin Safaat H, 2012) Android adalah sistem operasi berbasis Linux bagi telepon seluler seperti telepon pintar dan komputer tablet. Android juga menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk berbagai macam piranti gerak.

Menurut (Yuniar, 2015) merupakan sebuah sistem informasi perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Sistem operasi Android ini bersifat *open source* sehingga banyak sekali programmer yang berbondong-bondong membuat aplikasi maupun memodifikasi sistem ini. Para programmer memiliki peluang yang sangat besar untuk terlibat mengembangkan aplikasi Android karena alasan *open source* 31 tersebut. Sebagian besar aplikasi yang terdapat dalam *Play Store* bersifat gratis dan ada juga yang berbayar.

1. Kelebihan *Android*
 - a. *Multitasking*.
 - b. Kemudahan dalam Notifikasi
 - c. Akses Mudah terhadap Ribuan Aplikasi *Android* lewat Google Android
 - d. *App Market*
 - e. Pilihan Ponsel yang beranekaragam .
 - f. Bisa menginstal ROM yang dimodifikasi
 - g. Widget.
 - h. *Google Maniak* (layanan *Google* mulai dari Gmail sampai Google Reader, ponsel *Android* telah terintegrasi dengan layanan Google, sehingga Anda bisa dengan cepat mengecek email dari Gmail.)

2. Kelemahan *Android*

- a. Koneksi *Internet* yang terus menerus
- b. Iklan

2.7 Perangkat Lunak Yang Digunakan

2.7.1 *Vuforia SDK*

Menurut (Arifitama, 2017) *Vuforia SDK* *Vuforia* adalah merupakan sebuah software *development kit SDK* yang dikeluarkan oleh *qualcomm*, untuk pengembangan sebuah aplikasi di bidang *computer vision* khususnya teknologi *virtual reality* dan *Augmented Reality*. *Vuforia* menggunakan teknologi *Computer Vision* untuk mengenali dan melacak marker atau image target dan objek 3D sederhana, seperti kotak, secara *real-time*.

AR Vuforia memberikan cara berinteraksi yang memanfaatkan kamera mobile phones untuk digunakan sebagai perangkat masukan, sebagai mata elektronik yang mengenali penanda tertentu, sehingga di layar bisa ditampilkan perpaduan antara dunia nyata dan dunia yang digambar oleh aplikasi. Dengan kata lain, *Vuforia* adalah SDK untuk *computer vision based AR*. Jenis aplikasi *Augmented Reality* yang lain adalah *GPS-based AR*.

2.7.2 *Unity 3D*

Menurut (Arifitama, 2017) *Unity 3D* merupakan sebuah *platform* pengembangan game engine 2 dimensi maupun 3 dimensi yang dapat digunakan oleh pengembang baru maupun pengembang yang sudah berpengalaman. *Unity 3D* dapat digunakan untuk membuat sebuah game yang bisa digunakan pada perangkat komputer, Android, iPhone, Playstation, dan X-Box. *Unity 3D* adalah sebuah *tool* yang terintegrasi untuk membuat game, arsitektur bangunan dan simulasi. *Unity 3D* bisa digunakan untuk PC games dan online games. Penggunaan dalam online game diperlukan sebuah plugin, yaitu *Unity Web Player* seperti *Flash Player* pada browser. *Unity* dapat di jalankan dengan *sistem requirement* sebagai berikut :

1. *System requirements for Unity development*

- a. Windows XP SP2 or later; Windows 7 SP1; Mac OS X "Snow Leopard" 10.6 or later. Note that *Unity* was not tested on server versions of Windows and OS X.

- b. *Graphics card with DirectX 9 level (shader model 2.0) capabilities. Any card made since 2004 should work.*
 - c. *Using Occlusion Culling requires GPU with Occlusion Query support (some Intel GPUs do not support that).*
2. *System requirements for Unity Android development In addition to the general system requirements:*
- a. *Android SDK and Java Development Kit (JDK)*
 - b. *Android authored content requires devices equipped with:*
 - c. *Android OS 2.3.1 or later*
 - d. *Device powered by an ARMv7 (Cortex family) CPU*
 - e. *GPU support for OpenGL ES 2.0 is recommended*

2.7.3 Blender

Menurut (Hendratman Hendi, 2017) Pada saat ini grafis komputer telah mengalami perkembangan yang sangat cepat. Grafis komputer mampu menghasilkan animasi dengan model 3 dimensi yang sudah sangat detail, selain itu terdapat game 3 dimensi yang sudah dapat dibuat dengan lebih baik dibandingkan sebelumnya. Banyak software aplikasi yang beredar saat ini , salah satunya adalah Blender. Blender adalah software animasi yang cukup populer saat ini. Selain untuk membuat animasi , Blender juga sudah terdapat *game engine* (mesin permainan).

Blender merupakan *free* dan *open source* software. Blender dapat dipakai di beberapa system operasi yaitu Windows (XP, Vista, 7), Mac OS (X 10.4 keatas) , Linux, dan FreeBSD 6.2 i386.

2.7.4 Bahasa Pemrograman C#

Menurut (R.H SIANIPAR, 2014) C# (dibaca C-Sharp) merupakan bahasa pemrograman yang dibuat khusus untuk platform .NET. mendefinisikan C# sebagai pemrograman *object oriented*, dan *event driven* karena di dalam C# dapat merespon pengguna dengan *event* klik atau dengan penekanan tombol, dan sebagai bahasa pemrograman visual. Dari penjelasan di atas dapat disimpulkan bahwa bahasa pemrograman C# merupakan cabang dari bahasa C, C++ dan java, dimana Bahasa pemrograman C# khusus digunakan untuk platform .NET oleh microsoft.

2.8 Metode Analisis *PIECES*

Menurut (Wukil Ragil, 2010) metode *PIECES* adalah metode analisis sebagai dasar untuk memperoleh pokok-pokok permasalahan yang lebih spesifik. Dalam menganalisis sebuah sistem, biasanya akan dilakukan terhadap beberapa aspek antara lain adalah kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi dan pelayanan pelanggan. Analisis ini disebut dengan *PIECES Analysis (Performance, Information, Economy, Control, Eficiency and Service)*.

Analisis *PIECES* ini sangat penting untuk dilakukan sebelum mengembangkan sebuah sistem informasi karena dalam analisis ini biasanya akan ditemukan beberapa masalah utama maupun masalah yang bersifat gejala dari masalah utama. Metode ini menggunakan enam variable evaluasi yaitu :

1. *Performance* (kinerja)

Kinerja merupakan variable pertama dalam metode analisis *PIECES*. Dimana memiliki peran penting untuk menilai apakah proses atau prosedur yang ada masih mungkin ditingkatkan kinerjanya, dan melihat sejauh mana dan seberapa handalkah suatu sistem informasi dalam berproses untuk menghasilkan tujuan yang diinginkan. Dalam hal ini kinerja diukur dari:

- a. *Throughput* yaitu jumlah pekerjaan/*output/ deliverables* yang dapat dilakukan/ dihasilkan pada saat tertentu.
- b. *Response time*, yaitu waktu yang dibutuhkan untuk menyelesaikan serangkaian kegiatan untuk menghasilkan *output/ deliverables* tertentu.

2. *Information* (informasi)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki sehingga kualitas informasi yang dihasilkan menjadi semakin baik. Informasi yang disajikan haruslah benar-benar mempunyai nilai yang berguna. Hal ini dapat diukur dengan:

- a. Keluaran (*outputs*): Suatu sistem dalam memproduksi keluaran.
- b. Masukan (*inputs*): Dalam memasukkan suatu data sehingga kemudian diolah untuk menjadi informasi yang berguna.

3. *Economic* (ekonomi)

Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan manfaatnya (nilai gunanya) atau diturunkan biaya penyelenggaraannya.

4. *Control* (pengendalian)
Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan sehingga kualitas pengendalian menjadi semakin baik, dan kemampuannya untuk mendeteksi kesalahan/ kecurangan menjadi semakin baik pula.
5. *Efficiency* (efisiensi)
Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki, sehingga tercapai peningkatan efisiensi operasi, dan harus lebih unggul dari pada sistem manual.
6. *Service* (layanan)
Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki kemampuannya untuk mencapai peningkatan kualitas layanan. Buatlah kualitas layanan yang sangat *user friendly* untuk *end – user* (pengguna) sehingga pengguna mendapatkan kualitas layanan yang baik.

2.9 Metode Pengembangan *Prototyping*

Menurut (Pressman, 2010) dalam melakukan perancangan sistem yang akan dikembangkan dapat menggunakan metode *prototyping*. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat yang akan dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna, dalam hal ini pengguna dari perangkat yang dikembangkan adalah peserta didik. Kemudian membuat sebuah rancangan kilat yang selanjutnya akan dievaluasi kembali sebelum diproduksi secara benar.

Prototyping bukanlah sesuatu yang lengkap, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi pada saat *prototyping* dibuat untuk memenuhi kebutuhan pengguna dan pada saat yang sama memungkinkan pengembang untuk lebih memahami kebutuhan pengguna secara lebih baik.

Metode atau tahap-tahap dalam perancangan aplikasi ini menggunakan *Prototype Model* dimana sistem ini nantinya dapat dikembangkan kembali. Model *Prototype* dimulai dari mengumpulkan kebutuhan pengguna terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program *prototype* agar pengguna lebih terbayang dengan apa yang sebenarnya diinginkan. Tahapan-tahapan *Prototype* yang dimulai dari *Listen to Customer*, *Build/Revise*, *Customer Test-Drives Mock-Up*.

2.10 Metode *WhiteBox Testing*

Menurut (Rouf, 2012) Pengujian dilakukan dengan metode *whitebox*, karena dengan pengujian *whitebox*, pengujian ini yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara *procedural* untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan *whitebox* testing merupakan petunjuk untuk mendapatkan program yang berjalan dengan benar.

a. Kelebihan *White Box Testing*

1. Kesalahan logika. Digunakan pada sintaks 'if' dan pengulangan. Dimana *White Box Testing* akan mendeteksi kondisi-kondisi yang tidak sesuai dan mendeteksi kapan proses pengulangan akan berhenti.
2. Ketidakesuaian asumsi. Menampilkan asumsi yang tidak sesuai dengan kenyataan, untuk di analisa dan diperbaiki.
3. Kesalahan ketik. Mendeteksi bahasa pemrograman yang bersifat *case sensitive*.

b. Kelemahan *White Box Testing*

1. Untuk perangkat lunak yang tergolong besar, *White Box Testing* dianggap sebagai strategi yang tergolong boros, karena akan melibatkan sumber daya yang besar untuk melakukannya.

2.11 *Unified Modeling Language (UML)*


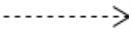

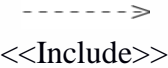
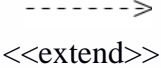




Menurut (Rosa & Shalahudin, 2014) *unified Modeling Language (UML)* adalah salah satu standar bahas yang banyak digunakan didunia industri untuk mengidentifikasi requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

Beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis *diagram*, yang lain menyebutkan delapan karena ada beberapa *diagram* yang digabung, misalnya *diagram* komunikasi, *diagram* urutan dan *diagram* pewaktuan digabung menjadi *diagram* interaksi. Macam-macam *diagram* UML sebagai berikut:


a. *Use Case Diagram*

Menurut (Rosa & Shalahudin, 2014) *Use case* merupakan pemodelan untuk kelakuan (*behavior*) system informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antar satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Tabel 2.1 Simbol-Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem itu sendiri.
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
3		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.
4		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri.
5		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.
6		<i>Association</i>	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).

Lanjutan Tabel 2.1 Simbol-Simbol *Use Case Diagram*




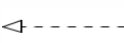
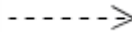

10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
----	---	-------------	---

(Sumber: Rosa A.S dan M. Shalahuddin. 2014)

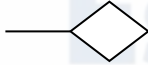
b. *Class Diagram*

Menurut (Rosa & Shalahudin, 2014) *Class diagram* menggambarkan struktur dan dekripsi *class*, *package*, dan objek beserta hubungan satu sama lain seperti *containment*, pewaris, asosiasi, dan lain-lain. *Class diagram* berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain. Objek adalah nilai tertentu dari setiap attribute kelas entity. *Class* adalah sebuah spesifikasi yang jika diinstansikan akan menghasilkan sebuah objek dan merupakan inti dari pengembangan berorientasi objek. *Class* menggambarkan keadaan (*attribute/property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi).

Tabel 2.2 Simbol - simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Association</i>	Hubungan antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
5		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
6		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah <i>Use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.

Lanjutan **Tabel 2.2** Simbol - simbol *Class Diagram*











7		<i>Agregasi/aggregation</i>	Hubungan antar kelas dengan makna semua-bagian (<i>whole part</i>)
---	---	-----------------------------	--

(Sumber: Rosa A.S dan M. Shalahuddin . 2014)

c. *Activity Diagram*

Menurut (Rosa & Shalahudin, 2014) *Activity diagram* adalah gambaran berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Berikut adalah simbol-simbol yang sering digunakan dalam perancangan *activity diagram* :

Tabel 2.3 Simbol – Simbol *Activity Diagram*







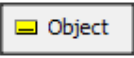
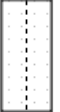
SIMBOL	KETERANGAN
	Titik Awal
	Titik Akhir
	<i>Activity</i>
	Pilihan untuk pengambilan keputusan
	<i>Fork</i> ; digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Rake</i> ; menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (<i>Flow Final</i>)

(Sumber : Rosa A.S dan M. Shalahuddin. 2014)


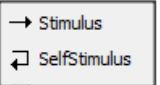
d. *Diagram Sequence*

Menurut (Rosa & Shalahudin, 2014) *Diagram Sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar *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 sequence* juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Tabel 2.4 Simbol-Simbol *Diagram Sequence*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem itu sendiri.
2		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah penggambaran dari form.
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel.
5		<i>Lifeline</i>	Menyatakan kehidupan suatu objek
6		<i>Line Message</i>	Menggambarkan pengiriman pesan.
7		<i>Object</i>	Menyatakan objek yang berinteraksi pesan
8		Garis Hidup (<i>Lifeline</i>)	Menyatakan kehidupan suatu objek.

Lanjutan Tabel 2.4 Simbol-Simbol *Diagram Sequence*

9		Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya aktor tidak memiliki waktu aktif.
10		Stimulus	Menyatakan suatu objek mengirimkan pesan untuk menjalankan operasi yang ada pada objek lain.

(Sumber: Rosa A.S dan M. Shalahuddin, 2014)

e. *State Machine Diagram*

Menurut (Rosa & Shalahudin, 2014) *State Machine Diagram* atau *statechart diagram* atau dalam bahasa Indonesia disebut dengan *diagram* mesin status atau sering juga disebut *diagram* status digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek. Jika *diagram* sekuen digunakan untuk interaksi antar objek maka *diagram* status digunakan untuk interaksi di dalam sebuah objek. Perubahan tersebut digambarkan dalam suatu graf berarah.

f. *Component Diagram*

Menurut (Rosa & Shalahudin, 2014) *Diagram* Komponen atau *Component Diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. *Diagram* komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem.

g. *Deployment Diagram*

Menurut (Rosa & Shalahudin, 2014) *Diagram Deployment* atau *Deployment Diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. *Diagram Deployment* juga dapat digunakan untuk memodelkan hal-hal berikut :

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node*, dan *hardware*.
2. Sistem *client/server*

2.12 Usability Testing (Uji Kegunaan)

Menurut (Sriwulandari, Hidayati, & Pudjoatmojo, 2014) *Usability Testing* adalah salah satu kategori metode dalam evaluasi *usability* yang mengobservasi pengguna sebuah desain kemudian diambil data dan menganalisisnya. Biasanya, selama tes, pengguna akan mencoba menyelesaikan tugasnya, sementara pengamat melihat, mendengar dan membuat catatan. Tujuannya adalah untuk mengidentifikasi masalah kegunaan, mengumpulkan data kualitatif dan kuantitatif dan menentukan kepuasan pengguna dengan aplikasi. Prinsip-prinsip heuristik yang dihasilkan adalah sebagai berikut :

1. *Learnability* : Seberapa mudah bagi pengguna memahami saat pertama kali melihat. Kemudahan tersebut diukur dari pemakaian fungsi-fungsi dan fitur yang tersedia.
2. *Effeciency* : Seberapa cepat dapat menyelesaikan perintah (*input*).
3. *Memorability* : Saat pengguna menggunakan lagi seberapa ingat (terbisa) terhadap penggunaannya. Berkaitan dengan kemampuan pengguna mempertahankan pengetahuannya setelah jangka waktu tertentu. Kemampuan tersebut diarahkan oleh tata letak desain *interface* yang relative tetap. *Interface* yang relative tetap ini memiliki komponen seperti penempatan fitur, huruf, dan warna.
4. *Errors* : Berapa banyak kesalahan yang diperbuat saat menggunakannya. Berkaitan dengan kesalahan-kesalahan yang dibuat oleh yang dilakukan oleh pengguna selama berinteraksi dengan *website* atau aplikasi teetentu.
5. *Satisfaction* : Seberapa nyaman pengguna dengan antarmuka tersebut. Berkaitan dengan kepuasan pengguna setelah menggunakan aplikasi. Pengukuran terhadap kepuasan juga meliputi aspek manfaat yang didapat dari pengguna selama menggunakan perangkat tertentu.

Teknik analisis data menggunakan *Skala Linkert*. Menurut (sugiyono, 2011) *Skala Linkert* dapat digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial yang merupakan skala kontinum bipolar, pada ujung sebelah kiri (angka rendah) menggambarkan suatu jawaban yang bersifat *negative*. Sedangkan ujung sebelah kanan (angka tinggi), menggambarkan suatu jawaban yang bersifat positif. Berikut untuk mengetahui jumlah jawaban dari para responden melalui presentase yaitu digunakan rumus sebagai berikut :

$$p = \frac{f}{n} \times 100\%$$

Gambar 2.1 Presentase Persetujuan

Keterangan :

p : Prosentase

f : Frekuensi dari setiap jawaban angket

n : Jumlah skor ideal

100 : Bilangan tetap