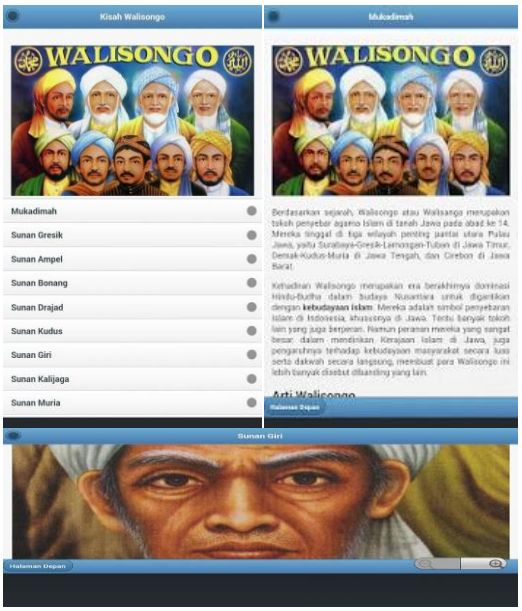
BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka dalam penelitian ini mengenai penilaian tampilan dari beberapa Aplikasi tentang edukasi walisonggo ini sudah banyak ditemukan di *Playstoremarket* untuk pengguna *android*. Aplikasi ini berisikan tentang profil dan cerita sejarah para Wali di Indonesia dalam penyebaran Agama Islam,dalam aplikasi tersebut banyak mengetahui bagaimana informasi yang disajikan dan fitur – fitur apa saja yang disajikan, adapun dari hasil penilaian yang peneliti lakukan dari aplikasi tersebut adalah sebagai berikut :

2.1.1 Aplikasi Android Kisah Walisongo



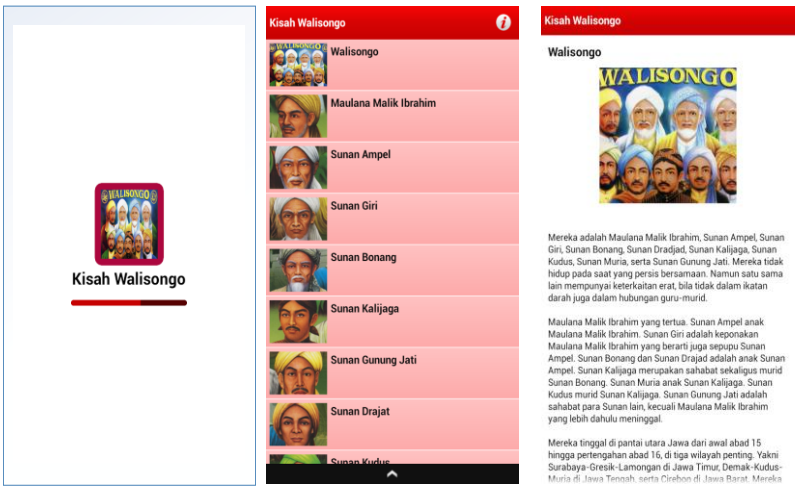
Gambar 2.1. Aplikasi Kisah Walisongo

Sumber : *Playstore*, 2015, Aplikasi Kisah Walisongo, *Playstore Android*, Minggu, 5 Juli 2015 (11:01:53 PM)

Gambar 2.1 merupakan tampilan dari salah satu aplikasi kisah walisongo yang berjalan saat ini, aplikasi tersebut diatas menampilkan beberapa fitur atau

menu yaitu menu nama walisongo, dari menu walisongo tersebut menampilkan sejarah atau kisah dari walisongo yang berupa *text* dan gambar.

2.1.2 Aplikasi Sejarah Walisongo



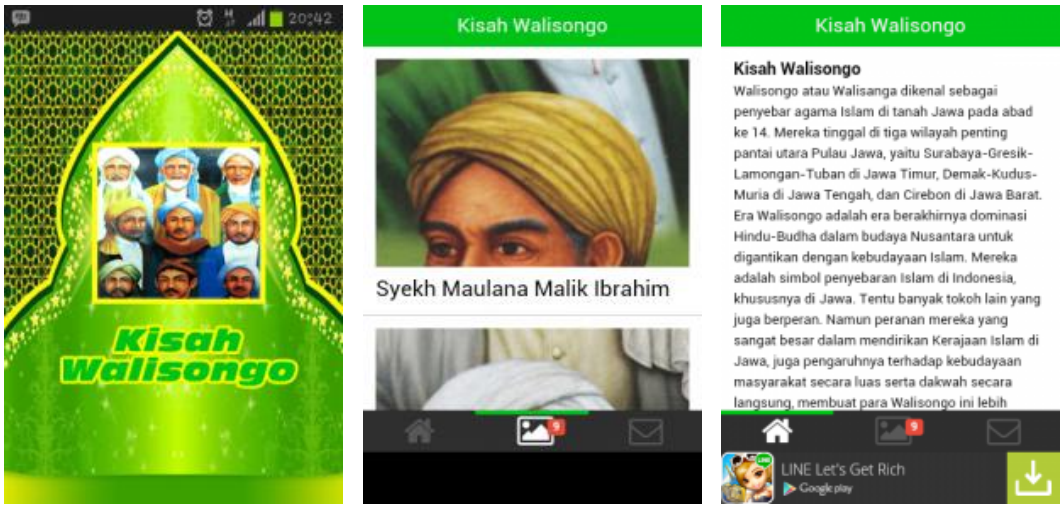
Gambar 2.2 Aplikasi Sejarah Walisongo

Sumber : *Playstore,* 2015, Aplikasi Sejarah Walisongo, *Playstore Android*, Minggu, 5 Juli 2015 (11:05:53 PM)

Gambar 2.2 merupakan tampilan dari salah satu aplikasi sejarah walisongo yang berjalan saat ini, seperti pada gambar 2.1 pada gambar 2.2 mempunyai menu – menu pelengkap seperti menu nama walisongo, dari menu tersebut menampilkan sejarah atau kisah dari walisongo.

2.1.3 Aplikasi Walisongo

Aplikasi *Android* walisongo dalam hal ini ditunjukkan pada gambar 2.3 dari aplikasi tersebut menerapkan warna hijau menyala sebagai warna dasar, berbeda dengan gambar 2.1 dan gambar 2.2 pada gambar 2.3 ini menampilkan tiga menu *icon*, *icon* yang pertama adalah menu Home dari tampilan Home ini menampilakn menu utama, *icon* kedua menampilkan menu *Picture* dimana menu ini menampilkan nama – nama walisongo, *icon* yang ketiga adalah Menu *Massage* menu ini menampilkan *contact person* dan menampilakn profil dari aplikasi ini.



Gambar 2.3 Aplikasi Walisongo

Sumber : *Playstore*, 2015, Aplikasi Walisongo, *Playstore Android*, Minggu, 5 Juli 2015 (11:05:53 PM)

Dari ketiga sempel aplikasi diatas yang bersumber aplikasi di *playstore* menunjukan beberapa kelemahan, Aplikasi pertama mempunyai menu yang sederhana dan kelemahan hanya menampilkan sejarah tanpa ada menu edukasi. Aplikasi kedua menu yang sangat sederhana dan kelemahan hanya menampilkan sejarah walisongo. Aplikasi yang ketiga mempunyai menu yang tidak menarik dan mempunyai kelemahan hanya ada kisah sejarah walisongo tanpa ada menu edukasi. dengan khusus diatas ingin membuat aplikasi yang serupa untuk diterapkan di SMP Negeri 1 Kartosuro. Aplikasi diberi nama aplikasi Edukasi Pengenalan Walisongo aplikasi ini secara prinsip sama dengan semple aplikasi diatas, penulis mencoba mengembangkan dengan lebih menarik lagi seperti berikut:

Menu utama atau tampilan utama aplikasi edukasi pembelajaran ini menyediakan 4 menu utama yaitu menu kisah wali, tebak gambar, kuis, tentang, aplikasi juga didukung *backsound* supaya membuat pengguna tidak merasakan jenuh dan membuat imajinasi pengguna lebih bervariasi, dan juga ditambahkan efect efect agar aplikasi terlihat menarik..

Dari perbandingan dan antara ketiga aplikasi dengan aplikasi yang dibuat, aplikasi pembelajaran yang telah dibuat lebih banyak variasi dari sisi manfaat dan

kegunaan aplikasi edukasi pengenalan walisongo ini di ditujukan untuk para siswa siswi, aplikasi ini sangat membantu belajar, tidak menutup kemungkinan aplikasi ini masih kurang dari segi grafis.

2.2 Pengertian Aplikasi

Menurut (Kadir, 2003) aplikasi adalah suatu program yang dibuat oleh pengguna yang ditujukan untuk melakukan suatu tugas khusus. Berdasarkan definisi di atas dapat disimpulkan bahwa aplikasi adalah program yang dibuat untuk melakukan tugas khusus dalam perusahaan. Menurut (Supriyanto, 2005) aplikasi adalah program yang memiliki aktivitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu.

2.2.1 *Game Maker*

Game Maker merupakan aplikasi yang di gunakan untuk membuat game, baik 2D maupun 3D. GameMaker memungkinkan anda tidak perlu mempelajari bahasa pemrograman lagi ketika menggunakannya. Anda hanya memprogram game secara *drag* and *drop*. Dengan cara ini, memungkinkan anda membuat game dengan asyik tanpa perlu belajar keras mempelajari bahasa pemrograman, GameMaker telah menaglami banyak pembaruan bahkan dimulai sebelum penulis membuat naskah buku ini. Mengembangkan *game* adalah sesuatu yang diimpikan orang, baik dari kalangan muda maupun tua. Banyak pengembang aplikasi yang berusaha untuk membuat perkakas berupa aplikasi pengembangan *game* dengan tujuan mempermudah untuk mengembangkan game yang di inginkan, salah satunya yaitu *Game Maker*. Demikian, tidak ada istilah sulit dalam pengembangan game. *Game Maker* memungkinkan anda tidak perlu mempelajari bahasa pemrograman lagi ketika menggunakannya. Hanya memprogram *game* secara *drag and drop*, dengan cara ini, memungkinkan anda membuat game dengan asyik tanpa belajar keras mempelajari bahasa pemrograman. *Game Maker Language* (GML), adalah sebuah bahasa pemrograman yang dibuat oleh Overmars sebagai bahasa penunjang untuk *software* buatanya yakni *Game Make*r. *Game Maker language* ini mempunyai memberikan akses bagi para pembuat

game (baik pemula atau yang sudah mahir) untuk membuat serta mengembangkan *game* buatannya sendiri dengan cepat dan mudah (Nanda Prasetio : 1, 2014).

Tujuannya untuk menghindari hal-hal yang tidak diinginkan pada saat menjalankan aplikasi tersebut. Berikut persyaratan sistemnya.

a. Sistem operasi: *Windows XP/ Windows* 7*/ Windows* 8.
b. *Prosessor:Core* I5.
c. *Ram*: *Speed*2 Gb.
d. *VGA / Kartu grafis:SpeedMinimal*512 GB.
e. Resolusi layar: *Minimal* 800x600 dengan 16 bit warna
f. *Driver grafis (graphics driver)*: *MinimalDirectX* 8.

2.2.2 *CorelDRAW*

CorelDRAW merupakan aplikasi grafis yang fenomenal. Kemampuannya yang sangat fleksibel dalam mengolah objek membuat aplikasi ini begitu dikenal sehingga pengguna komputer terbiasa mendengar keunggulan aplikasi ini dan mungkin juga telah menjadikannya *default* pekerjaan grafis (Edi S Mulyanta:1, 2004)

Menurut Agnessia Atitatita (2011), program *CorelDraw* adalah program aplikasi desain grafis yang berguna untuk membuat desain vektor, logo, dana *layout* halaman, hampir setiap tahun Program *CorelDraw* memperbaruhi versi dan fasilitas di dalamnya sehingga mempermudah dalam membuat desain desainya. *Program CorelDraw versi X5* memiliki banyak kelebihan dibanding *corelDraw* sebelumnya. *Corel Draw X5* ini membawa beberapa fitur antara lain ;

1) *Video Tutorial*

   CorelDRAW X5 menyediakan *video turorial* yang berguna untuk mempelajari program *CorelDRAW X5* dengan cepat.

2) *Tooltips*

   Ketika posisi *pointer* berada diatas ikon atau tombol, maka *tool* tip akan muncul menampilkan informasi tentang nama, fungsi dan cara penggunaanya.

3) *Corel PowerTRACE X5*

Dengan perintah *PowerTrace*, kamu dapat mengkonversi objek bitmap menjadi objek vektor. Objek vektor yang menghasilkan dapat diedit lebih halus dan akurat dibanding versi program *CorelDRAW* sebelumnya.

4) *Drawing Tools*

   Ada beberapa *Drawing tool* atau alat gambar baru di dalam *CorelDRAW X5* seperti: *B-Spline tool*, skala mata panah, *Connector tool* dan *Dimension tool.*

5) *Mesh Fill tool*

   Mesh Fill tool digunakan untuk mewarnai objek dengan berbagai campuran warna, Dengan tombol *Smooth mesh color* yang baru pada *property bar,* kamu dapat menghasilkan transisi warna yang labih halus. Selain itu, jumlah *node per mesh* telah berkurang sehingga lebih mudah untuk memanipulasi objek.

6) *Bounding Box*

   Ketika menggambar garis atau kurva dengan *curve tool,* kamu dapat menampilkan atau menyembunyikan *bounding box,* sehingga kamu lebih mudah mengatur bentuk dan arah garis tanpa terganggu oleh *bounding box.*

7) *Round corners*

   Saat membuat kotak dengan *Rectangle tool*, kamu dapat mengatur sudut kotak dengan bentuk *chamfer, scallop,* atau *round corners* dari *property bar.*

8) *Lock Toolbar*

   Tampilan *Toolbar* dapat dikunci diposisi tertentu sehingga tidak mungkin tergeser saat kamu mengklik *tool.* Jika kamu ingin memindah *toolbar,* kamu dapat membuka kuncinya dan mengatur posisi *toolbar* pada layar.

9) Kotak *dialog Export for Web*

   Kotak *dialog Export for Web* dapat kamu gunakan untuk mengekspor desain dari *program CorelDRAW* ke tampilan web dengan perintah *File Export for Web.*

2.2.3 *Adobe Photoshop* CS5

Adope Photoshop CS5 adalah salah satu software untuk mengelolah foto ataupun gambar yang sangat populer saat ini. Dengan menggunakan adope photoshop kita dapat memperbaiki dan mempercantik foto yang ingin kita cetak dengan menambahkan beberapa efect dalam foto tersebut, sehingga foto yang biasa akan menjadi sebuah foto dengan tampilan yang berbeda dan menarik (Madcoms:2,2011).

2.3 Pengertian Android

Menurut Arif Dwi Susanto (2013), *Android* adalah susunan dari beberapa perangkat lunak (*software stack*). Stack ini secara umum meliputi sistem operasi, *middleware*, dan aplikasi-aplikasi kunci. Android pada awalnya tidak dikembangkan oleh google, melainkan dikembangkan oleh sebuah perusahaan bernama *Android Inc*. Karena google melihat banyaknya user yang online dengan perangkat mobile, maka google mengira bahwa perangkat mobile ini memiliki masa depan yang cerah, sehingga *Android Inc* diakuisi oleh Google pada tahun 2005. Beberapa hal penting seputar android :

a. *Android* adalah sistem operasi embedded yang sangat bergantung pada kernel linux untuk layanan-layanan core-nya, tapi Android bukanlah linux *embedded*.
b. Penulisan program untuk android menggunakan *framework* java, tapi ini bukanlah java. Karena library standar java seperti Swing tidak didukung. *Library* lain seperti *timer* tidak disarankan, karena sudah diganti dengan *library default* dari *Android*, yang dioptimalkan untuk penggunaan di lingkungan *embedded* yang terbatas.
c. OS android merupakan sistem operasi *open source*, artinya *developer* bisa melihat semua *source code* sistem, termasuk stack radio.

Android adalah nama *software* yang diapakai pada perangkat *mobile* yang mencakup berbagai komponen, yaitu sistem operasi, *middleware* dan aplikasi kunci yang dirilis oleh *google*. Jadi, *Android* ini mencakup keseluruhan aplikasi, mulai dari sistem operasi hingga pengembangan aplikasi itu sendiri. Dan

pengembangan aplikasi pada platform. *Android* ini menggunakan dasar bahasa pemograman *Java*.

Platform pengembang aplikasi *Android* yang merupakan bagian dari *Android* memiliki lisensi *open-source* atau terbuka, sehingga Anda dapat membangun aplikasi kaya dan inovatif (Tim EMS:1,2013).

2.4 Pengertian Multimedia

Multimedia adalah kombinasi dari komputer dan video (Roseh, 1996) atau Multimedia secara umum merupakan kombinasi tiga elemen yaitu suara, gambar, dan teks (McConnick 1996) atau Multimedia adalah kombinasi dari paling sedikit dua media input atau output dari data, media ini dapat audio (suara, musik), animasi, video, teks, grafik dan gambar (Turban dkk, 2002) Multimedia merupakan alat yang dapat menciptakan presentasi yang dinamis dan interaktif yang mengkombinasikan teks, grafik, animasi, audio, dan gambar video (Robin dan Linda, 2001).

Definisi yang lain dari multimedia yaitu dengan menempatkannya dalam konteks, seperti yang dilakukan oleh Hofstertter (2001), multimedia adalah pemanfaatan komputer untuk membuat dan menggabungkan linkdan tools yang memungkinkan pemakai melakukan navigasi, berinteraksi, berkreasi dan berkomunikasi (M.Suyanto:20,2003).

2.4.1 Multimedia *Linear*

Multimedia *linear* adalah multimedia yang bersifat sekuensial atau berurutan, setiap siswa atau pemakai multimedia ini menggunakanya sesuai dengan urutan setahap demi setahapsesuai dengan pengemasan materi yang ditentukan. Siswa belajar berdasarkan bagian-bagian yang didesain sedemikan rupa secara berurutan dengan waktu yang telah ditentukan, bentuk multimedia yang bersifat linear memiliki kelebihan di antaranya sebagai berikut:

1) Lebih mudah dalam pengembanganya, Hal ini disebakan multimedia yang bersifat linier bentuknya lebih sederhana yang tidak banyak menggunakan fungsi kontrol.

2) Multimedia ini mudah digunakan siswa karena siswa tidak dihadapkan pada berbagai *frame* dan menu pilihan.
3) Multimedia liniear terdiri atas bagian-bagian atau unit-unit terkecil bahan pelajaran, dengan demikan lebih mudah dalam kontrol penguasaan materi oleh siswa.
4) Bentuk umpan balik dapat dilakukan sehingga segera pula siswa dapat mempebaiki apabila diperlukan.

2.4.2 Multimedia Interaktif

Mmultimedia interaktif adalah multimedia yang tidak bersifat linier, namu siswa memiliki pilihan sesuai dengan menu yang ditawarkan. Dalam mempelajari satu topik bahasan siswa dapat memilih mana yang akan dipelajari lebih dahulu. Dengan demikian ciri khas dari multimedia interaktif adalah semacam pengontrol yang biasa disebut *graphical user interface (GUI),* yang bisa, *scroll* berupa *icon, button* atau yang lainya. Setia *GUI* tersebut dapat dioperasikan oleh siswa (pemakai) untuk mencari informasi yang diinginkan. Beberapa manfaat penggunaan multimedia interaktif diantaranya:

1) Multimedia interaktif sifatnya lebih dinamis sehingga tidak membosankan.
2) Multimedia interaktif memberikan plihan menu yang lebih beragam sehingga siswa lebih menyukai sebagai pemakai media ini.
3) Kajian materi pelajaran yang lebih lengkap memungkin multimedia interaktif lebih memiliki keanekargaman materi yang dapat dipahami siswa

2.5 Analisis dan Perancangan Sistem

2.5.1 Analisis Sistem

Analisa sistem didefinisikan sebagai bagaimana memahami dan menspesifikasi dengan detail apa yang harus dilakukan oleh sistem. Sedangkan sistem desain diartikan sebagai menjelaskan dengan detail bagaimana bagian-bagian dari sistem informasi diimplementasikan. Sehingga Analisa dan desain sistem informasi (ANSI) bisa didefinisikan sebagai: Proses organisasional

kompleks dimana sistem informasi berbasis komputer diimplementasikan (Kartika Imam Santoso:18,2011).

Metodologi pengembangan Sistem Proses-proses standard yang digunakan untuk membangun suatu sistem informasi meliputi langkah-langkah berikut ini:

1) Analisa
2) Desain
3) Implementasi
4) Maintenance

Pada perkembangannya, proses-propses standar tadi dituangkan dalam satu metode yang dikenal dengan nama Systems Development Life Cycle (SDLC) yang merupakan metodologi umum dalam pengembangan sistem yang menandai kemajuan dari usaha analisa dan desain. SDLC meliputi fase-fase sebagai berikut:

1) Identifikasi dan seleksi proyek
2) Inisiasi dan perencanaan proyek
3) Analisa
4) Desain
    a) Desain logikal
    b) Desain Fisikal
6) Implementasi
7) Maintenance

2.5.2 Perancangan Sistem

Setelah analisis sistem langkah selanjutnya adalah perancangan sistem. Perancangan sistem secara umum didefinisikan sebagai pengindentifikasian komponen-komponen sistem informasi dengan tujuan untuk dikomunikasikan dengan pemakai (Hans-Erik Eriksson : 2004).

Object Oriented Program(OOP) merupakan paradigma baru dalam rekayasa *software* yang didasarkan pada objek dan kelas (Hans-Erik Eriksson: 2004). Diakui para ahli bahwa *object - oriented* merupakan metodologi terbaik yang ada saat ini dalam rekayasa *software*. *Object-oriented* memandang *software* bagian perbagian dan menggambarkan satu bagian tersebut dalam satu objek.

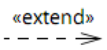
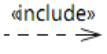
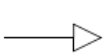
Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh objek, dengan demikian keunggulan teknologi objek adalah bahwa model yang dibuat akan sangat mendekati dunia nyata yang masalahnya akan dipecahkan oleh sistem yang dibangun. Model objek, *atribut* dan perlakuannya bisa langsung diambil dari objek yang ada di dunia nyata.

Unified Modeling Language (UML) merupakan standar yang relatif terbuka yang dikendalikan oleh OMG *(Object Menegement Group),* sebuah konsorium terbuka yang terdiri dari banyak perusahaan.OMG dibentuk untuk membuat standar-standar yang mendukung interoperabilitas sistem berorientasi objek. Menurut (Russ Milles & Kim Hamilton : 2006), UML terdiri 7 buah diagram, adapun diagram yang sering digunakan dalam pemodelan, yaitu *Usecase Diagram, Class Diagram, Sequence Diagram, Activity Diagram, State Diagram, Componen Diagram, Deploymen Diagram,* dalam pembuatan UML Diagram digunakan Program Aplikasi *Rational Rose,* berikut pembahasannya.

2.5.2.1 *Usecase Diagram*

Berikut ini adalah Tabel Simbol – simbol dari *Usecase Diagram*

Tabel2.1 Simbol-simbol *Usecase*

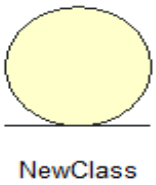
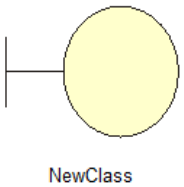
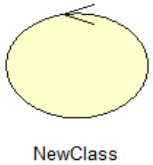
NO	SIMBOL	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak (*descendent*) perilaku dan struktur data dari objek yang ada di atasnya objek induk
2		*Exten*	*Insertion* tambahan ke *Usecase* yang tidak diketahui
3		*Include*	Menspesifikasikan bahwa *Usecase*sumber secara *explicit.*
4		*Usecasegeneralization*	Hubungan antara *Usecase*satu dengan *Usecase* yang lainnya.
5		*Actor*	Pengguna di luar sistem

6		*Usecase*	Sebuah spesifikasi dari perilaku sebuah *entitas* dalam interaksinya dengan agenluar

Usecase diagram adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Usecase* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan Martin Flower (2005:141). Diagram *Usecase* dekat kaitannya dengan kejadian-kejadian. Kejadian (skenario) merupakan contoh apa yang terjadi ketika seseorang berinteraksi dengan sistem. *Usecase* Diagram dibuat untuk memvisualisasikan atau menggambarkan hubungan antara *Actor* dan *Usecase*.

2.5.2.2 *Usecase Realization*

Tabel 2.2 *Simbol-Simbol Usecase Realization*

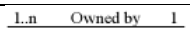
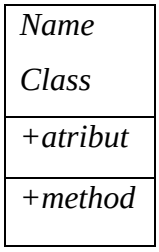
NO	SIMBOL	NAMA	KETERANGAN
1		*Boundary Class*	Digunakan untuk menggambarkan bagaimana *Usecase* direalisasikan sebagai interaksi antara *object.*
2		*Entity Class*	Digunakan untuk menangani informasi yang mungkinakan selalu di simpan dalam proses bisnis.
3		*Control Class*	Kelas kontrol untuk satu *Usecase* yang digunakan mengatur kejadian dalam *Usecase* tersebut.

Fungsionalitas Usecase direpresentasikan dengan aliran peristiwa-peristiwa. Skenario digunakan untuk menggambarkan bagaimana *Usecase-Usecase* direalisasikan sebagai interaksi antara objek-objek. *Usecase realization* menggambarkan bagaimana realisasi dari setiap *usecase* yangada pada *Usecase* model, untuk menggambarkan bagaimana realisasi dari suatu *usecase* dapat menggunakan beberapa diagram, diantaranya adalah *Class Diagram owned by Usecase Realization* serta *Interaction Diagram*.

2.5.2.3 *Class Diagram*

Berikut ini adalah tabel dari Simbol – simbol Class Diagram

Tabel 2.3 Simbol-simbol *Class Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Association*	Sebuah asosiasi merupakan sebuah *relationship* paling umum antara 2 *class*
3		*Class*	*Class* adalah blok - blok pembangun pada pemrograman berorientasi objek. Sebuah *class* digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian.
4		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.

5		*Dependency*	Kadangkala sebuah *class* menggunakan *class* yang lain, hal ini disebut *dependency*. Umumnya penggunaan *dependency* digunakan untuk menunjukkan operasi pada suatu *class* yang menggunakan *class* yang lain.

Diagram kelas mempunyai 3 macam *relationalships* (hubungan) yaitu :

1) *Association*

Suatu hubungan antara bagian dari dua kelas, terjadi asosiasi antara dua kelas jika salah satu bagian dari kelas mengetahui yang lainnya dalam melakukan suatu kegiatan, didalam diagram, sebuah asosiasi adalah penghubung yang menghubungkan dua kelas.

2) *Aggregation*

Suatu *association* dimana salah satu kelasnya merupakan bagian dari suatu kumpulan, *aggregation* memiliki titik pusat yang mencakup keseluruhan bagian.

3) *Generalization*

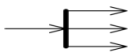
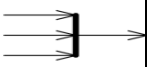
Suatu hubungan turunan dengan mengasumsikan satu kelas merupakan suatu *super class* (kelas super) dari kelas yang lain, contoh: *Payment* adalah *super class* dari *Cash, Check, Credit*.

2.5.2.4 *Activity Diagram*

Berikut ini adalah Tabel Simbol – simbol *Activity Diagram*

Tabel 2.4 Simbol - Simbol *Activity Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Decision node*	suatu titik / *point* pada *activity* diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi

3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
6		*Join node*	Beberapa aliran masukan tertentu berupah menjadi satu aliran

Activity Diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja, dalam beberapa hal, diagram ini memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antara diagram ini dan notasi diagram alir adalah diagram ini mendukung *behavior* paralel (Martin Flower, 2005:163).

2.5.2.5 *Statechart Diagram*

Berikut ini adalah Simbol – simbol *Statchart Diagram*

Tabel 2.5 Simbol-simbol *Statechart Diagram*

No	SIMBOL	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial State*	Bagaimana objek dibentuk atau diawali
3		*Transition*	Sebuah kejadian yang memicu sebuah *state* objek dengan cara memperbaharui satu atau lebih nilai atributnya
4		*Final State*	Kondisi akhir alur hidup objek
5		*Concurrent composite state*	sebuah *state* yang dibagi menjadi duaatau lebih *concurrent substate*, semua dari *concurrently* aktif ketika *composite state* aktif.
6		*Junction*	Sebuah *pseudo statesegmen* rantai transisi

		state	dalam *single run* untuk menyelesaikan transisi
7		*Suquencial composite state*	*State* yang berisi satu atau lebih *substate*, tepatnya salah satunya aktif pada satu waktu ketika *composite state* aktif
8		*History state*	Sebuah *pseudostate* mengembalikan keadaan state sebelumnya di dalam *composite state*

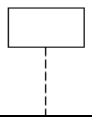
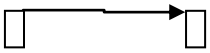
Statechart Diagram adalah teknik yang umum digunakan untuk menggambarkan sifat sebuah sistem Martin Flower (2005:151), *Statechart diagram* menggambarkan semua *state* atau kondisi yang dimiliki oleh suatu *object* dari suatu *class* dan kejadian yang menyebabkan *state* berubah. Kejadian dapat berupa objek lain yang mengirim pesan.

Statechart Diagram menurut Rahmad Hidayat (2010), *Statechart diagram* menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

2.5.2.6 *Sequence Diagram*

Berikut ini adalah Tabel dari Simbol – simbol *Sequence Diagram*

Tabel 2.6 Simbol-simbol *Sequence Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

3		*Actor*	Pengguna di luar sistem
4		*Fragment*	Menggambarkan batas grafis suatu diagram

Sequence diagram adalah penjabaran *behavior* sebuah skenario tunggal. *Sequence diagram* menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek ini di dalam *Usecase* Martin Flower (2005:81). *Sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah skenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antara objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

2.5.2.7 *Component Diagram*

Component merupakan bagian fisik dari sebuah sistem, karena menetap di komputer tidak berada dianalis. *Component* terhubung melalui antarmuka yang digunakan dan dibutuhkan (Martin Flower, 2005:189). *Component* merupakan implementasi *software* dari sebuah atau lebih *class*. *Component* dapat berupa *sourcecode*, komponen biner, atau *executable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*. Sehingga *component diagram* merepresentasikan dunia *real* yaitu *component software* yang mengandung *component*, *interface* dan *relationship*. Simbol-simbol yang digunakan pada *component diagram* dapat dilihat pada Tabel di bawah

Tabel 2.7 Simbol-simbol *Component Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		*Componant*	*Physical* dari sebuah sistem

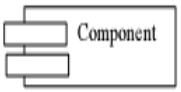
2		*Interface*	*Aname state* informasi yang menjadi ciri dari perilaku
3		*Usage*	situasi di mana satu elemen membutuhkan lain untuk fungsiyang benar
4		*Realization*	hubungan antara spesifikasi dan implementasinya

2.5.2.8 *Deployment Diagram*

Simbol-simbol yang digunakan pada *Deployment Diagram* dapat dilihat pada Tabel di atas.

Deployment Diagram menujukkan susunan fisik sebuah sistem, menunjukkan bagian perangkat lunak mana yang berjalan pada perangkat keras mana (Martin Flower,2005:137). *Deployment Diagram* juga menggambarkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya.

Tabel. 2.8 Simbol-Simbol *Deployment Diagram*

No	SIMBOL	NAMA	KETERANGAN
1		***Component***	Pada *deployment diagram*, komponen-komponen yang ada diletakkan didalam *node* untuk memastikan keberadaan posisi mereka.
2		***Node***	*Node* menggambarkan bagian-bagian *hardware* dalam sebuah sistem. Notasi untuk *node* digambarkan sebagai sebuah kubus 3 dimensi.
3		***Association***	Sebuah *association* digambarkan sebagai sebuah garis yang menghubungkan dua *node* yang mengindikasikan jalur komunikasi

			antara *node* ke *node* yang lain.

2.6 Metode Five View

Software Testing and *QA Theoryand Practice* (*Chapter* 17: *Software Quality*) *Universityof Waterloo*, pengujian *fiveview* adalah pengujian yang sifatnya deskriptif dimana *software* yang diuji dinilai melalui lima sudut pandang *(metode five view)* atau kategori yang berbeda melalui penilaian dari *expertise* atau orang yang berpengalaman dari masing-masing sudut pandang. Lima sudut pandang tersebut adalah sebagai berikut:

1) *Transcendental View*

Kualitas menurut pandangan ini ialah sesuatu yang dapat dikenali melalui pengalaman tapi tidak dapat selalu digambarkan. Objek atau *software* yang bagus itu menonjol dan dapat dengan mudah dikenali.

2) *User View*

Kualitas menyangkut sejauh mana produk memenuhi kebutuhan dan harapan pengguna dan apakah suatu produk cocok untuk digunakan. Pendapat ini bersifat sangat *personal*. Sebuah produk berkualitas baik jika memuaskan sebagian besar pengguna, Hal ini berguna untuk mengidentifikasi fitur dari produk yang pengguna anggap penting. Pandangan ini dapat mencakup banyak unsur subjek, seperti kegunaan, keandalan, dan ke-efisiensasian.

3) *Manufacturing View*

Pandangan ini berkaitan dengan faktor dakam industri manufaktur, apakah produk memenuhi persyaratan atau tidak Setiap penyimpangan dari persyaratan yang dinilai mengurangi kualitas produk. Konsep proses memainkan peran kunci. Produk yang dibuat harus orisinil sehingga biaya berkurang, misal biaya pembangunan dan biaya pemeliharaan.

Kesesuaian dengan persyaratan dan spesifikasi menyebabkan keseragaman dalam produk tapi beberapa berpendapat bahwa keseragaman tersebut tidak

menjamin kualitas. Kualitas produk dapat secara bertahap ditingkatkan dengan memperbaiki proses.

4) *Product View*

Jika sebuah produk diproduksi dengansifat internal(misalnya bahan dan tindakan) yang baik, maka produk akan memiliki sifat *eksternal* atau *output* yang baik dan dapat dieksplorasi hubungan antara sifat internal dan kualitas eksternal.

5) *Value-based view*

Value-based view merupakan penggabungan dari dua konsep yaitu keunggulan dan kelayakan. Kualitas adalah ukuran dari keunggulan dan nilai adalah ukuran layak. Berapa banyak penggunabersedia membayar untuk tingkat kualitas tertentu. Kualitas tidak berarti jika produk memenuhi nilai ekonomi. Pandangan berbasis nilai antara biaya dan kualitas.