

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Tinjauan pustaka dalam melaksanakan penelitian dan untuk menyusun tugas akhir ini, dilakukan penelitian untuk mencari beberapa *website* Program Studi Desain Interior yang sudah diterapkan di beberapa Universitas yang sudah mempunyai *website*. Hal ini berguna untuk jadi bahan referensi dan pembandingan. Sebagai referensi dalam membangun *website* Program Studi Desain Interior Universitas Sahid Surakarta dilakukan pengamatan dari berbagai contoh *website* misalnya seperti *webiste* www.interior.petra.ac.id dan www.tarumanagara.ac.id. Dari *website* tadi banyak dilakukan fasilitas dan fitur-fitur yang dimiliki *website* tersebut.

2.1.1. Website Program Studi Desain Interior Universitas Kristen Petra

Website Universitas Kristen Petra yang dapat diakses melalui alamat www.interior.petra.ac.id, *website* ini dari sisi penampilannya sangat *simple* bahkan tidak terlalu banyak *content-content*, juga monoton tidak ada paduan warna. Jika dilihat dari segi penggunaannya mudah digunakan, jika dari sisi informasi *website* ini juga kurang detail untuk isinya. Berikut adalah tampilan *website* program studi Desain Interior Universitas Kristen Petra terlihat pada Gambar 2.1.

Selamat Datang di Desain Interior UK Petra

OVERVIEW

Selamat Datang di Situs Web Program Studi Desain Interior Universitas Kristen Petra Surabaya-Jawa Timur, Indonesia Kami senang untuk dapat memperkenalkan profil, program dan kegiatan kami. Bila anda memerlukan bantuan, anda dapat menghubungi kami melalui email: kj-intrr@petra.ac.id Bergabunglah dengan grup Facebook kami: DI UKP Grup Facebook untuk mahasiswa: DESAIN INTERIOR PETRA

ACARA

22
SEP
2015

Dies Natalis UK
Petra

BERITA TERBARU

No recent news!

PENDAFTARAN MAHASISWA BARU

Universitas Kristen Petra menerima pendaftaran mahasiswa melalui 3 jalur, yaitu: Jalur Kerjasama, Jalur Prestasi, dan Jalur Umum. Ingin tahu lebih banyak? Silakan klik link di samping.

[PENERIMAAN
MAHASISWA BARU](#)



UNIVERSITAS
KRISTEN
PETRA

Sivalankerto 121-131 Surabaya
+62 31 8439040, 8394830-31
+62 31 8436418



Let Your Light
Shine

Love - Integrity - Growth - Humility - Truth

Universitas Kristen Petra

[Download](#)

[Term of use](#)

Gambar 2.1 Website Program Studi Desain Interior. Universitas Kristen Petra

Sumber url : www.interior.petra.ac.id

2.1.2. Website Program Studi Desain Interior Universitas Tarumanegara

Website Program Studi Desain Interior Universitas Tarumanegara yang dapat diakses melalui alamat www.tarumanagara.ac.id, website ini sangat bagus dan mempunyai fitur dan fasilitas banyak di halaman website seperti informasi akademik yang sangat lengkap dan berita terbaru. Dari sisi tampilan sangat *simple* mudah digunakan oleh user, kemudian perpaduan warnanya juga bagus akan tetapi di tampilan home terlalu monoton, seperti yang terlihat pada Gambar 2.2.

The screenshot shows the website for the Interior Design Program at Tarumanegara University. The header includes the UNTAR logo and navigation links: BERANDA, TENTANG UNTAR, MAHASISWA BARU, AKADEMIK, RISET, KEMAHAGISWAAN, KERJASAMA, UNTAR KARIR, and KONTAK. The main content area is titled "PROGRAM STUDI DESAIN INTERIOR" and includes the following sections:

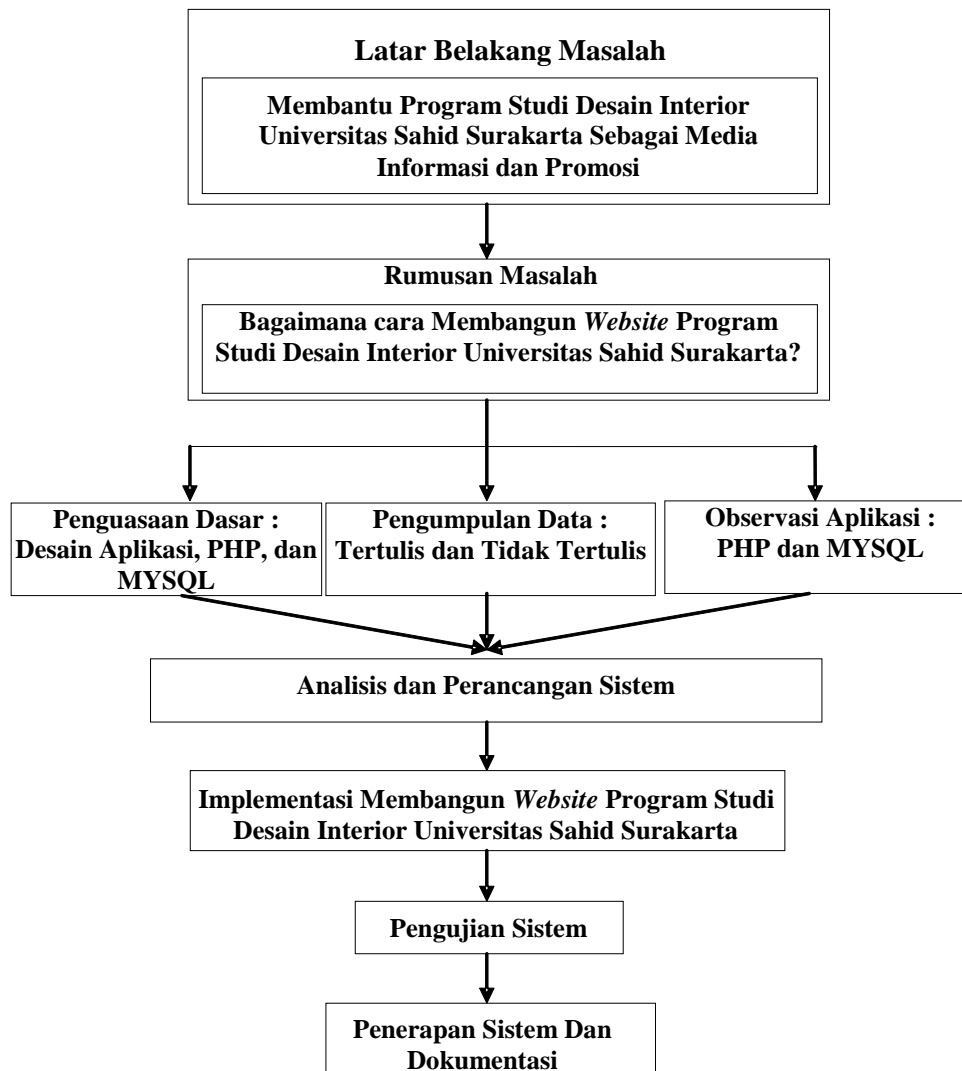
- 2011-07-18**
TERAKREDITASI PERINGKAT A
- TUJUAN PENDIDIKAN**
Program Studi Desain Interior diarahkan untuk menghasilkan traga ahilsarjana sedain (S1) yang memiliki kemampuan menganalisa data/informasi dan memecahkan persoalan desain, serta mengolah hasil analisa solusi menjadi rancangan interior yang fungsional, kreatif dan estetis, serta dapat dipertanggung jawabkan secara sosio kultural.
- LABORATORIUM/STUDIO**
 - Laboratorium komputer grafis
 - Laboratorium Fotografi digital & teknik cetak foto (kamar gelap)
 - Laboratorium seni rupa (cetak grafis & lukisan)
- PROSPEK LAPANGAN PEKERJAAN**
 - Desainer interior
 - Desainer furnitur
 - Kontraktor interior
 - Konsultan spesialis untuk desain
 - Exhibition, desain museum

On the right side, there is a "Link Fakultas" section with a list of links: [Ekonomi](#), [Teknik](#), [Psikologi](#), [Teknologi Informasi](#), [Seni Rupa dan Desain](#), [Hukum](#), and [Ilmu Komunikasi](#). Below the main content, there is a "Link :" label.

Gambar 2.2 Website Program Studi Desain Interior. Website Untar

Sumber url : <http://www.tarumanagara.ac.id/>

2.2. Kerangka Pemikiran



Gambar 2.5. Diagram Kerangka Pemikiran

2.3. Landasan Teori

2.3.1. Pengertian *Website*

Website atau situs dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana

masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*). Bersifat statis apabila isi informasi website selalu berubah-ubah, dan isi informasinya interaktif dua arah berasal dari pemilik serta pengguna *website* (Anggiana Septa Riyadi, 2012).

Website adalah halaman yang ditampilkan di internet yang memuat informasi tertentu (khusus). Web pertama kali diperkenalkan pada tahun 1992. Hal ini sebagai hasil usaha pengembangan yang dilakukan CERN di Swiss. Internet dan web adalah dua hal yang berbeda. Internet yaitu yang dapat menampilkan web-nya, sedangkan web adalah yang ditampilkannya yang berupa susunan dari halaman-halaman yang menggunakan teknologi web dan saling berkaitan satu sama lain (Yosefa Ferdianto, 2013).

Pada pendapat yang ada di atas maka dapat diambil kesimpulan bahwa *website* adalah kumpulan dari halaman - halaman situs, yang terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam *World Wide Web* (WWW) di dalam Internet. Sebuah halaman web biasanya berupa dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang selalu bisa diakses melalui HTTP, yaitu sebuah protokol yang menyampaikan informasi dari server *website* untuk ditampilkan kepada para pemakai melalui *web browser*. dan *Website* atau situs dapat juga diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*)

2.3.2 Pengertian PHP

Pengertian *Hypertext Preprocessor* (PHP) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis (M.Rusdyanto Arief, 2011:43). Karena PHP merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di *server* kemudian hasilnya akan dikirim ke *browser* dalam format HTML.

Kode program yang ditulis dalam PHP tidak akan terlihat oleh *user* sehingga keamanan *web* lebih terjamin. PHP dirancang untuk membentuk halaman *web* yang dinamis, yaitu halaman *web* yang dapat membentuk suatu tampilan berdasarkan perintah terkini, seperti menampilkan isi basis data ke halaman *web*.

Sistem manajemen *database* yang sering digunakan bersama PHP adalah MySQL. Namun PHP juga mendukung sistem manajemen *database oracle*, *Microsoft Accses*, *Interbase*, *d-Base*, *PostgreSQL* dan sebagainya. PHP mendukung penuh *Object Oriented Programing* (OOP), *integrasi XML*, mendukung semua ekstensi terbaru MySQL, serta sama dengan *web server* lainnya. PHP juga bersifat *open source* sehingga setiap orang dapat menggunakan dengan gratis (Nugroho, 2004).

2.3.2.1 Kelebihan PHP

Kelebihan yang dimiliki PHP adalah kemampuannya untuk melakukan koneksi ke berbagai macam *software system* manajemen basis data / *Database Manajement System* (DBMS), sehingga dapat menciptakan suatu halaman *web* yang dinamis. PHP mempunyai koneksitas yang baik dengan DBMS antara lain: *Oreacle*, *Sybase*, *MySQL*, *Microsoft SQL Server*, *Solid*, *PostgresSQL*, *Adabas*, *FilePro*, *Velocis*, *dBase*, *unix dbm*, dan tidak terkecuali semua *database* ber-*interface ODBC*.

2.3.2.2 Kekurangan PHP

PHP tidak memiliki sistem pemrograman berorientasi objek yang sesungguhnya. PHP memiliki kelemahan pada *security* tertentu apabila programmer tidak jeli dalam melakukan pemrograman dengan kurang memperhatikan isu dan konfigurasi PHP.

2.3.3 HTML (*Hyper Text Markup Language*)

HTML merupakan kependekan dari *Hyper Text Markup Language*. Biasanya html ini digunakan sebagai pondasi dari suatu aplikasi *Web*. Pengerjaan *file html* dapat dilakukan sembarang, maksudnya ialah menggunakan *editor* apa saja, *html* dapat dikerjakan. Untuk menciptakan *file html*, setidaknya ada dua

macam *ekstensi* yang dapat kita gunakan, diantaranya *.htm* dan *.html*. *Ekstensi* dokumen html yang menggunakan tiga karakter awalnya adalah untuk mengkomodasikan sistem penamaan yang ada dalam sistem operasi DOS. Dokumen html merupakan dokumen yang disajikan dalam *browser Web* dan biasanya hanya untuk menampilkan informasi maupun *interface* (Blanco : 2009) .

Menurut Sunyoto (2007) Secara umum, *document Web* dibagi menjadi dua bagian, yaitu *head* dan *body*, sehingga setiap *dokumenhtml* harus mempunyai pola dasar sebagai berikut.

```

< html >
< head >
      Desain Interior Universitas Sahid Surakarta
< /head >
< body >
      “ Gerbang Menuju Kesuksesan “
< /body >
< /html >

```

Setiap *dokumenhtml* harus diawali dengan perintah *tag* *< html >* dan diakhiri dengan *< /html >*. *Tag* tersebut menandakan bahwa kita akan mengerjakan suatu *file* dokumen *html*.

Elemenhead ditandai dengan perintah *< head >*, dan diakhiri dengan tag *< /head >*, pada bagian ini berisi tentang *dokumenhtml*, misalnya judul dokumen yang nanti akan ditampilkan pada bagian pojok atau kiri *Web browser* melalui *tag* *< title >.... < /title >*. Cara penggunaan dapat dilihat pada struktur sebagai berikut:

Bagian *body* merupakan tempat untuk menampilkan informasi yang akan ditampilkan pada jendela *Web browser*. bagian ini ditandai dengan tag *< body >* dan diakhiri dengan tag *< /body >*.

```

< head >

< title > RoadToSolo “ Guide Your Trip To Solo “ < / title >

< / head >

```

2.3.4 Pengertian *Web Browser*

Merupakan program yang berfungsi untuk menampilkan dokumen-dokumen *web* dalam format HTML. (M.Rusdyanto Arief, 2011: 19). Halaman *web* yang dibuat ditampilkan sangat tergantung pada *webengine* yang digunakan oleh masing-masing *broser* yang cukup populer beserta *web engine* yang digunakan:

- 1) *Web Engine Webkit* : Safari, Google , Chrome.
- 2) *Web Engine Trident* : Microsoft Internet Exploler, Maxthon.
- 3) *Web Engine Gecko* : Mozilla Firefox.
- 4) *Web Engine Presto* : Opera .

2.3.5 Pengertian Basis Data

Basis data adalah kumpulan dari item data yang saling berhubungan satu dengan yang lainnya, yang diorganisasikan berdasar sebuah skema atau struktur tertentu, tersimpan di *hardware* komputer dengan *software* tertentu digunakan untuk melakukan manipulasi data untuk tujuan tertentu.

Suatu basis data adalah koleksi data yang bisa mencari secara menyeluruh dan sistematis memelihara informasi (Janner Simarmata: 2009). Basis data dapat diartikan sebagai himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat mudah (Fatansyah: 2007).

2.3.5.1 Istilah-istilah dalam basis data

Dalam basis data ada Istilah-istilah sebagai berikut:

- 1) *Entity* adalah orang, tempat, kejadian atau konsep yang informasinya direkam.
- 2) Atribut adalah sebuah sebutan yang digunakan untuk mewakili suatu entiti.
- 3) Data *value* (nilai atau isi data) Adalah data aktual atau informasi yang disimpan pada tiap data elemen atau atribut.

- 4) *Record/ Tuple* Adalah kumpulan elemen-elemen data yang saling berkaitan menginformasikan suatu entiti secara lengkap.
- 5) *File* Adalah kumpulan *record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama namun berbeda data *valuenya*.
- 6) *Database* Adalah kumpulan *file-file* yang mempunyai kaitan antara satu *file* dengan *file* lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu.
- 7) *Database Management Sistem (DBMS)* Adalah kumpulan *file* yang saling berkaitan bersama program untuk pengelolaanya disebut DBMS. *Database* adalah kumpulan datanya, sedang program pengelolanya berdiri sendiri dalam satu paket program yang komersial untuk membaca, mengisi, menghapus, melaporkan data dalam *database*.
- 8) Kunci elemen data Adalah suatu tanda pengenal yang secara unik mengidentifikasi sebuah entitas dari suatu himpunan entitas.

2.3.6 Pengertian MySQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *databasenya* (A Kadir: 2008). MySQL selain mudah instalasinya juga *database* ini mampu menampung *record* sebanyak ratusan giga (M Syafi'i: 2004).

MySQL memiliki dua bentuk lisensi, yaitu: *free software* dan *shareware*. MySQL yang biasa digunakan adalah MySQL *free software* yang berada dibawah lisensi *General Public Licence (GNU/GPL)*. Selain itu juga dapat memiliki produk MySQL yang bersifat komersial, biasa disebut MySQL AB, MySQL menggunakan logo standar berbentuk seperti Lumba-Lumba.

MySQL merupakan sebuah *database server* yang *free*, artinya bebas menggunakan sebuah *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. Selain sebagai *database server*, MySQL juga merupakan program yang dapat mengakses suatu *database* MySQL yang berposisi *server*. Pada saat itu berarti program berposisi sebagai *client*. Jadi

MySQL adalah sebuah *database* yang dapat digunakan baik sebagai *client* maupun *server*.

2.3.6.1 Keistimewaan MySQL

MySQL memiliki beberapa keistimewaan, antara lain :

1) *Portability*

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows*, *Linux*, *FreeBSD*, *Mac Os X Server*, *Solaris*, *Amiga*, dan masih banyak lagi.

2) *Open Source*

MySQL didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3) *Multiuser*

MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa banyak masalah atau konflik. Hal ini memungkinkan sebuah *database server* MySQL dapat diakses *client* secara bersamaan.

4) *Performance Tuning*

Memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL persatuan waktu.

5) *Column Types*

MySQL memiliki banyak kolom yang sangat kompleks, seperti: *signed/unsignedinteger*, *float*, *double*, *char*, *varchar*, *text*, *blob*, *date*, *time*, *datetime*, *timestamp*, *year*, *set* serta *enum*.

6) *Command dan Functions*

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam *query*.

2.4. Analisis dan Perancangan Sistem

2.4.1 Analisis Sistem

Analisis Sistem adalah penguraian dari suatu sistem informasi yang utuh kedalam bagian - bagian komponennya dengan maksud untuk mendefinisikan dan mengevaluasi permasalahan - permasalahan, kesempatan - kesempatan, hambatan-hambatan yang terjadi dan kebutuhan - kebutuhan yang diharapkan

sehingga diusulkan perbaikan – perbaikan. Analisis sistem bertujuan untuk mengidentifikasi permasalahan - permasalahan yang ada pada sistem dimana aplikasi dibangun yang meliputi perangkat keras (*hardware*), perangkat lunak (*software*) dan pengguna (*User*). Analisis ini diperlukan sebagai dasar bagi tahapan perancangan sistem. Analisis sistem meliputi spesifikasi aplikasi, spesifikasi pengguna, dan lingkungan operasi Cristian Putri (2003) .

2.4.2 Perancangan Sistem

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering) (Sri Dharwiyanti,2003:2).

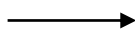
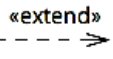
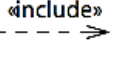
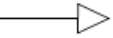


Seperti juga tercantum pada bukunya UML mendefinisikan diagram-diagram sebagai berikut:, yaitu

1. use case diagram 2. class diagram 3. statechart diagram 4. activity diagram 5. sequence diagram 6. collaboration diagram 7. component diagram 8. deployment diagram

2.4.1 Usecase Diagram

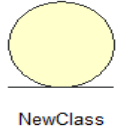
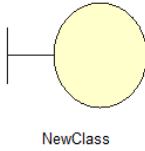
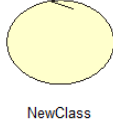
Usecase diagram adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Usecase* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan Martin Flower (2005:141). Diagram *Usecase* dekat 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*. Simbol-simbol dari *Usecase* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol-simbol *Usecase Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) perilaku dan struktur data dari objek yang ada di atasnya objek induk
2		<i>Extent</i>	<i>Insertion</i> tambahan ke <i>Usecase</i> yang tidak diketahui
3		<i>Include</i>	Menspesifikasikan bahwa <i>Usecase</i> sumber secara explicit.
4		<i>Usecase generalization</i>	Hubungan antara <i>Usecase</i> satu dengan <i>Usecase</i> yang lainnya.
5		<i>Actor</i>	Pengguna di luar sistem
6		<i>Usecase</i>	Sebuah spesifikasi dari perilaku sebuah <i>entitas</i> dalam interaksinya dengan agen luar

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

Tabel 2.2 Simbol-Simbol *Usecase Realization*

NO	SIMBOL	NAMA	KETERANGAN
1		<i>Boundary Class</i>	Digunakan untuk menggambarkan bagaimana <i>Usecase</i> direalisasikan sebagai interaksi antara <i>object</i> .
2		<i>Entity Class</i>	Digunakan untuk menangani informasi yang mungkin akan selalu di simpan dalam proses bisnis.
3		<i>Control Class</i>	Kelas <i>control</i> untuk satu <i>Usecase</i> yang digunakan mengatur kejadian dalam <i>Usecase</i> tersebut.

2.4.2. *Class Diagram*

Diagram kelas mempunyai 3 macam *relationships* (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, Di dalam diagram, sebuah asosiasi adalah penghubung yang menghubungkan dua kelas.


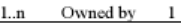
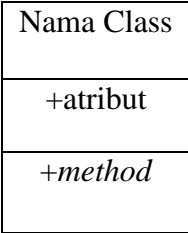


2) *Aggregation*

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*.

Tabel 2.3 Simbol-simbol *Class Diagram*





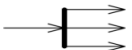
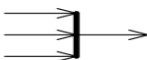
NO	SIMBOL	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Association</i>	Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum antara 2 <i>class</i>
3		<i>Class</i>	<i>Class</i> adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian.
4		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
5		<i>Dependency</i>	Kadangkala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain.

2.4.3 *Activity Diagram*

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). Simbol-simbol yang digunakan pada *activity diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Simbol - Simbol *Activity Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Decision node</i>	suatu titik/ <i>point</i> pada <i>activity diagram</i> yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Actifty Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
6		<i>Join node</i>	Beberapa aliran masukan tertentu berubah menjadi satu aliran

2.4.4 Statechart Diagram

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 Sri Dharwiyanti(2003), menunjukkan kondisi yang dapat dialami atau terjadi pada sebuah objek, status awal (kondisi awal alur hidup objek dan setiap *state diagram* hanya memiliki satu status awal), Status

akhir (kondisi akhir alur hidup objek), Transisi (garis transisi antar status pada alur hidup objek). Simbol-simbol yang digunakan pada *statechart diagram* dapat dilihat pada Tabel 2.5 di bawah ini

Tabel 2.5 Simbol-simbol *Statechart Diagram*

No	SIMBOL	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Transition</i>	Sebuah kejadian yang memicu sebuah <i>state</i> objek dengan cara memperbaharui satu atau lebih nilai atributnya
4		<i>Final State</i>	Kondisi akhir alur hidup objek
5		<i>Concurrent composite state</i>	sebuah <i>state</i> yang dibagi menjadi dua atau lebih concurrent substate, semuanya concurrently aktif ketika composite state aktif.
6		<i>Junction state</i>	Sebuah pseudo statesegmen rantai transisi dalam single run untuk menyelesaikan transisi
7		<i>Suquential composite state</i>	State yang berisi satu atau lebih substate, tepatnya salah satunya aktif pada satu waktu ketika composite state aktif
8		<i>History state</i>	Sebuah pseudostate mengembalikan keadaan state sebelumnya di dalam composite state

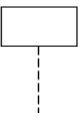
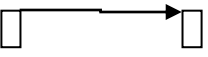


2.4.5 Sequence Diagram

Sequencediagram adalah penjabaran *behavior* sebuah skenario tunggal. *Sequencediagram* menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek ini di dalam *Usecase* Martin Flower (2005:81). *Sequencediagram* 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.

Dibawah ini merupakan simbol-simbol dari *Sequencediagram*, dapat dilihat ada simbol dalam *Sequencediagram*. Simbol-simbol yang digunakan pada *sequence diagram* dapat dilihat pada Tabel 2.6 di bawah ini





Tabel 2.6 Simbol-simbol *Sequence Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Actor</i>	Pengguna di luar system
4		<i>Fragment</i>	Menggambarkan batas grafis suatu diagram

2.4.6 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 *executablecomponent*. Sebuah komponent berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*. Sehingga *component diagram* merepresentasikan dunia nyata yaitu *componentsoftware* yang mengandung *component*, *interface* dan *relationship*. Simbol-simbol yang digunakan pada *component diagram* dapat dilihat pada Tabel 2.7.




Tabel 2.7 Simbol-simbol *Component Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		<i>Componant</i>	<i>Physical</i> dari sebuah system
2		<i>Interface</i>	A <i>name state</i> informasi yang menjadi ciri dari perilaku
3		<i>Usage</i>	situasi di manasatu elemen membutuhkan lain untuk fungsi yang benar
4		<i>Realization</i>	hubungan antara spesifikasi dan implementasinya

2.4.7 *Deployment Diagram*

Deployment Diagram menunjukkan 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. Simbol-simbol yang digunakan pada *deyploment diagram* dapat dilihat pada Tabel 2.8.

Tabel 2.8 Simbol-Simbol *Deployment Diagram*

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

2.5 Metode Penentuan Kualitas Web

2.5.1. Metode *Webqual*

Webqual merupakan salah satu metode atau teknik pengukuran kualitas website berdasarkan persepsi pengguna akhir. Metode ini merupakan pengembangan dari *Servqual*- yang disusun oleh Parasuraman, yang banyak digunakan sebelumnya pada pengukuran kualitas jasa. Instrumen penelitian pada Webqual tersebut dikembangkan dengan metode *Quality Function Development (QFD)*.

Webqual sudah mulai dikembangkan sejak tahun 1998 dan telah mengalami beberapa iterasi dalam penyusunan dimensi dan butir-butir pertanyaannya. Penelitian Barnes dan Vidgen (2003) yang menggunakan Webqual 4.0 untuk mengukur kualitas website yang dikelola oleh *OECD (Organization for Economic Cooperation and Development)*. Webqual 4.0 tersebut disusun berdasarkan penelitian pada tiga area yaitu :

- 1) Kualitas informasi dari penelitian sistem informasi,
- 2) Interaksi dan kualitas layanan dari penelitian kualitas sistem informasi, *e-commerce*, dan pemasaran,
- 3) *Usability* dari *human-computer interaction*.

Webqual merupakan metode pengukuran kualitas website berdasarkan persepsi pengguna akhir (masyarakat). Metode ini merupakan pengembangan dari *servqual* (Zeithaml et al. 1990) yang banyak digunakan sebelumnya pada pengukuran kualitas jasa (Diana 2012). *WebQual 4.0* disusun berdasarkan penelitian pada tiga area yaitu :

1. *Usability*

Usability adalah mutu yang berhubungan dengan rancangan site, sebagai contoh penampilan, kemudahan penggunaan, navigasi dan gambaran yang disampaikan kepada pengguna. (Barnes dan Vidgen 2002 didalam Vidiantoro 2012).

Menurut Tullis dan Albert (2008) Tiga komponen umum *usability* (Ichsani 2012):

- Adanya keterlibatan seorang pengguna,
- Pengguna melakukan suatu pekerjaan, dan
- Pengguna melakukan sesuatu dengan adanya produk, sistem atau hal

2. *Information Quality*

Kualitas informasi dari penelitian system informasi (*Information Quality*) *Information Quality* adalah mutu dari isi yang terdapat pada site, pantas tidaknya informasi untuk tujuan pengguna seperti akurasi, format dan keterkaitannya. (Barnes dan Vidgen; 2002 didalam Vidiantoro 2012)

3. *Service Interaction*

Service Interaction Quality adalah mutu dari interaksi pelayanan yang dialami oleh pengguna ketika mereka menyelidiki kedalam site lebih dalam, yang terwujud dengan kepercayaan dan empati, sebagai contoh isu dari keamanan transaksi dan informasi, pengantaran produk, personalisasi dan komunikasi dengan pemilik site. (Barnes dan Vidgen 2002 didalam Vidiantoro 2012).

Model pengukuran mutu website, Barnes and Vidgen (2003), dengan kuesionernya berdasarkan 3 dimensi WebQual 4.0 (Vidiantoro 2012).