BAB 2

LANDASAN TEORI

2.1 Teori Umum

Teori-teori umum yang berkaitan dengan skripsi ini dan menjadi landasan bagi teori-teori lainnya yaitu :

2.1.1 Teknologi Informasi

Menurut Proctor (2011, p3), teknologi informasi memiliki karakteristik sebagai pelajaran, desain, pengembangan, aplikasi, implementasi, mengatur komputer berdasarkan sistem informasi, khususnya aplikasi perangkat lunak dan perangkat keras pada komputer. Informasi sebagai pengetahuan yang berasal dari investigasi, pelajaran, dan instruksi. Teknologi sebagai aplikasi yang diperoleh dari pengetahuan.

Menurut Aksoy (2008, p8), teknologi informasi adalah sistem perangkat keras dan perangkat lunak yang menangkap, memproses, menyimpan, dan menyajikan informasi menggunakan energi listrik, magnetik dan elektromagnetik. Pada tahun 2000, beberapa orang hanya mengetahui *Wi-Fi* atau MP3 *file sharing*, dan belum ada YouTube®. Pada tahun 1990, belum ada akses internet.

2.1.2 Interaksi Manusia dan Komputer

Menurut Santoso (2010, p5), dari perspektif ilmu komputer, IMK identik dengan interaksi khususnya interaksi antar manusia, baik satu atau lebih manusia (sebagai pengguna komputer) dengan satu atau lebih mesin komputasi (komputer).

Interaksi manusia dan komputer merupakan interaksi atau komunikasi antara pengguna dengan sistem komputer melalui tampilan antarmuka untuk mencapai tujuan tertentu.

Menurut Santoso (2010, p20), 8 aturan emas (*eight golden rules*)perancangan antarmuka pemakai:

1. Berusaha untuk konsisten.

   Konsisten dalam banyak hal, seperti urutan aksi, istilah-istilah, menu, layar bantuan, warna, tata letak, penulisan huruf dan *font*.

2. Menyediakan *shortcuts.*

   Shortcut digunakan untuk mempercepat dan mempermudah pengguna dalam menggunakan suatu fungsi, sehingga pada akhirnya dapat mempercepat penyelesaian pekerjaannya.

3. Memberikan umpan balik yang informatif.

   Setiap aksi yang dilakukan oleh pengguna, sebaiknya harus ada umpan balik dari sistem.

4. Merancang kotak dialog.

   Urutan dari aksi-aksi harus diorganisasikan secara teratur apakah termasuk di dalam urutan awal, tengah, atau akhir. Kotak dialog akan mempermudah pengguna untuk mengingat urutan aksi yang telah dilakukannya. Hal ini akan membuat para pengguna dapat merencanakan aksi apa yang akan dilakukan berikutnya.

5. Memberikan pencegahan kesalahan dan penanganan kesalahan yang sederhana.

   Sistem harus dirancang dengan baik agar pengguna tidak dapat melakukan kesalahan yang berdampak serius. Jika terjadi kesalahan (*error),* maka sistem harus dapat langsung mendeteksi dan menawarkan mekanisme penanganan yang sederhana namun terjamin keberhasilannya.

6. Memungkinkan pembalikan aksi yang mudah.

   Setiap aksi yang dilakukan oleh pengguna sebaiknya dapat dikembalikan atau dibatalkan dengan mudah, agar pengguna tidak kesulitan ketika mengeksplorasi suatu aplikasi. Contoh pengembalian aksi adalah *undo.*

7. Mendukung pusat kendali internal (*internal locus of control).*

   Sistem harus memastikan agar pengguna benar-benar memegang kontrol akan sistem dan sistem tersebut merespon aksi yang dilakukan oleh pengguna.

8. Mengurangi beban ingatan jangka pendek.

   Tampilan dan fungsi sistem sebaiknya dibuat sesederhana mungkin agar pengguna mudah mengingat dan dapat diingat dalam jangka waktu yang panjang.Akses *online* untuk *command-syntax forms*¸singkatan, kode, dan informasi lainnya juga harus disediakan oleh sistem.

2.1.3 *Unified Modelling Language (UML)*

Menurut Whitten, Bentley, dan Dittman (2007,p430), *Unified Modelling Language (UML)* adalah suatu pendekatan yang digunakan untuk mempelajari objek-objek yang ada, dengan melihat apakah objek tersebut dapat digunakan kembali atau dimodifikasi untuk kegunaan baru. UML juga mendefinisikan objek baru maupun objek yang telah dimodifikasi, yang akan digabungkan dengan objek yang ada untuk membuat aplikasi bisnis.

UML terdiri dari beberapa diagram, yaitu :

1. *Use case diagram*

   Use case diagram menggambarkan interaksi antara aktor dan sistem (mendeskripsikan siapa yang akan menggunakan sistem dan apa yang dilakukan).

2. *Class diagram*

   Class diagram menggambarkan struktur objek sistem, dengan menunjukkan *class* dan objek beserta hubungan antara satu dengan yang lain.

3. *Activity diagram*

   Activity diagram menggambarkan aktivitas sistem yang sedang dirancang, keputusan yang mungkin terjadi dan bagaimana berakhirnya. Diagram ini lebih menggambarkan proses dan jalur secara umum.

4. *Sequence diagram*

Sequence diagram menggambarkan interaksi antara objek dengan sistem dan langkah-langkah yang dilakukan sebagai timbal balik dari suatu aktivitas untuk menghasilkan *output* tertentu.

2.1.4 *Database System*

Database (basis data) adalah sekumpulan data yang saling berhubungan.*Database system* adalah suatu sistem yang mengatur dan mengelola *database* menggunakan program komputer.Tujuan utamanya adalah untuk memudahkan dalam menyimpan dan mengambil data.Sebuah *database* umumnya tidak dapat dikelola begitu saja oleh pengguna.Karena itu, dibutuhkan sebuah sistem perangkat lunak yang disebut *Database Management System* (DBMS). Menurut Connolly (2005, p15), DBMS biasanya digunakan untuk menjembatani penggunaan *database* sehingga pengguna dapat mendefinisikan, membuat, memelihara dan mengontrol *database* tersebut.

Di dalam DBMS, disediakan 2 fasilitas yaitu DDL (*Data Definition Language*) dan DML (*Data Manipulation Language*) yang keduanya dapat diakses menggunakan *query language*. *Query language* yang paling umum digunakan sekarangadalah SQL (*Structured Query Language*) yang telah menjadi bahasa standar untuk DBMS.

Komponen utama *database system* ialah :

1. Perangkat keras *(*h*ardware)*
2. Basis data *(database)*

3. Perangkat lunak *(software)*
4. Pengguna *(user)* :
    a. Pengguna awam (*end user)*

       Pengguna awam adalah orang yang menggunakan suatu aplikasi.

    b. Pemrogram aplikasi

       Pemrogram aplikasi adalah orang yang membuat program aplikasi menggunakan *database* yang sesuai dengan kebutuhan pemakai.

    c. Administrator basis data (*Database Administrator)*

       Database Administrator adalah orang yang bertanggung jawab untuk mengelola *database*.

2.1.5 Aplikasi

Menurut anonim 1,aplikasi adalah penerapan konsep ke dalam bentuk perangkat lunak sesuai dengan keinginan pengguna, dengan tujuan untuk membantu pengguna dalam melaksanakan tugas. Suatu aplikasi saling berkaitan antara satu dengan yang lainnya. Contohnya, aplikasi komputer terdiri dari *software, hardware,* dan *brainware. Hardware* tidak dapat bekerja tanpa adanya *software* dan keduanya tidak dapat bekerja tanpa adanya *brainware.* Ketiganya tidak dapat dipisahkann dan bertanggung jawab dalam memproses masukan (*input*) dan menghasilkan keluaran (*output*).

2.1.6 *Web Browser*

Menurut anonim 2, w*eb browser* adalah suatu aplikasi yang digunakan untuk mencari, mengambil dan menampilkanhalaman-halaman di web yang berada di internet.*Web browser* tidak hanya berorientasi pada teks, tetapi dapat menampilkan file multimedia seperti video dan suara. Beberapa contoh *web browser* yang ada yaitu Internet Explorer®, Mozilla Firefox®, Opera® dan Google Chrome®.

Fungsi utama *web browser* adalah menampilkan halaman *web* yang berisi informasi yang dicari dari internet. Halaman *web* juga dapat disimpan, sehingga suatu saat dapat membuka halaman tersebut dengan mudah (*bookmark*). Informasi di dalam *website* pun dapat diambil dan disimpan sesuai dengan keinginan pengguna.

2.1.7 *Crowdsourcing*

Menurut Yuswohady(2009, p46), *Crowdsourcing* adalah suatu upaya menyerahkan segala proses di dalam perusahaan kepada komunitas pelanggan. Metode yang mengandalkan partisipasi komunitas dalam memenuhi tugas tertentu bisa disebut dengan *The Power of Crowd.* Kekuatan dari *social media network* adalah salah satu inovasi dimana kita bisa menerapkan konsep *crowdsourcing* tersebut. Dengan kekuatan dari *social media* kita dapat mempromosikan suatu produk hingga banyak orang dapat mengetahui produk tesebut. Contohnya pada film *The Girl With The Dragon Tattoo,* Distributor film tersebut bekerja sama dengan Amazon dan

Google untuk menargetkan para penggemar, kemudian dengan Facebook untuk menjaga mereka terlibat (engage), dan memungkinkan mereka untuk menyampaikan cerita film tersebut kepada temannya. Agensi digital jam membuat promosi tersebut dan surat kabar Metro, mitra kunci di sisi non digital dari promosi juga mendorong beberapa aktivitas di Facebook.

Gambar 2. 1 Kolaborasi Promosi di Berbagai Media

Kampanye film The Girl With The Dragon Tattoo dilakukan 19 minggu sebelum film tersebut diputar di bioskop. Trailer film dapat diakses di yahoo dan diiklankan juga di situs web surat kabar Metro. Selain itu interactive video website juga diluncurkan di www.thegirl.co.uk. Dengan promosi yang sedemikian gencarnya dalam waktu yang relatif cukup lama sebelum peluncuran film dengan menggunakan berbagai *touch point* menyebabkan komunikasi yang tercipta begitu baik sehingga berdampak pada suksesnya film tersebut.

Gambar 2. 2 Promosi Film The Girl with the Dragon Tattoo

Menurut Yuswohady (2009, p2), Internet yang sudah teragregasi menjadi suatu ribuan bahkan jutaan komunitas umat manusia melalui situs-situs seperti Friendster, YouTube, Facebook, Twitter, MySpace, Second Life, atau Blooger juga memunculkan potensi luar biasa untuk membentuk komunitas konsumen yang tak pernah terbayangkan dalam sejarah umat manusia. Contoh yang paling mudah adalah dengan blog. Dengan blog kita bisa menulis ide apapun yang berseliweran di kepala kita. Setelah ide ditulis kita juga bisa mengajak teman-teman untuk aktif berpartisipasi dengan berdiskusi atau sekedar ngobrol , memberikan komentar, menuangkan ide, atau memberi tanggapan. Contoh lain adalah wikipedia. Wikipedia ditulis oleh ribuan pakar berbagai bidang di seluruh dunia yang bekerja secara sukarela dalam platform yang terbuka (*open source*).

Ketika memiliki komunitas yang solid , maka akan ada potensi untk menjadikan orang tersebut sebagai “evangelists” atau “advocators” yang membicarakan produk kita dan mereka akan merekomendasikan segala karya kita.

2.1.8 Facebook

Facebook adalah sebuah layanan jejaring sosial yang diluncurkan pada bulan Februari 2004, dimiliki dan dioperasikan oleh Facebook, Inc. Pada September 2012, Facebook memiliki lebih dari satu miliar pengguna aktif, lebih dari separuhnya menggunakan telepon genggam.Pengguna harus mendaftar sebelum dapat menggunakan situs ini. Setelah itu, pengguna dapat membuat profil pribadi, menambahkan pengguna lain sebagai teman, dan bertukar pesan, termasuk pemberitahuan otomatis ketika mereka memperbarui profilnya. Selain itu, pengguna dapat bergabung dengan grup pengguna dengan ketertarikan yang sama, diurutkan berdasarkan tempat kerja, sekolah atau perguruan tinggi, atau ciri khas lainnya, dan mengelompokkan teman-teman mereka ke dalam daftar seperti "Rekan Kerja" atau "Teman Dekat".

Facebook didirikan oleh Mark Zuckerberg bersama teman sekamarnya dan sesama mahasiswa Universitas Harvard, Eduardo Saverin, Andrew McCollum, Dustin Moskovitz dan Chris Hughes.Keanggotaan situs web ini awalnya terbatas untuk mahasiswa Harvard saja, kemudian diperluas ke perguruan lain di Boston, Ivy League, dan Universitas Stanford. Situs ini secara perlahan membuka diri kepada mahasiswa di universitas lain sebelum dibuka untuk siswa sekolah menengah atas, dan akhirnya untuk setiap orang yang berusia minimal 13 tahun. Meski begitu, menurut survei Consumer Reports bulan Mei 2011, ada 7,5 juta anak di

bawah usia 13 tahun yang memiliki akun Facebook dan 5 juta lainnya di bawah 10 tahun, sehingga melanggar persyaratan layanan situs ini.

Studi Compete.com bulan Januari 2009 menempatkan Facebook sebagai layanan jejaring sosial yang paling banyak digunakan menurut jumlah pengguna aktif bulanan di seluruh dunia. Entertainment Weekly menempatkannya di daftar "terbaik" akhir dasawarsa dengan komentar, "Bagaimana caranya kita menguntit mantan kekasih kita, mengingat ulang tahun rekan kerja kita, mengganggu teman kita, dan bermain Scrabulous sebelum Facebook diciptakan?" Quantcast memperkirakan Facebook memiliki 138,9 juta pengunjung bulanan di AS pada Mei 2011. Menurut Social Media Today pada April 2010, sekitar 41,6% penduduk Amerika Serikat memiliki akun Facebook. Meski begitu, pertumbuhan pasar Facebook mulai turun di sejumlah wilayah dengan hilangnya 7 juta pengguna aktif di Amerika Serikat dan Kanada pada Mei 2011.

Nama layanan ini berasal dari nama buku yang diberikan kepada mahasiswa pada tahun akademik pertama oleh beberapa pihak administrasi universitas di Amerika Serikat dengan tujuan membantu mahasiswa mengenal satu sama lain. Facebook memungkinkan setiap orang berusia minimal 13 tahun menjadi pengguna terdaftar di situs ini.

2.1.9 Twitter

Twitter adalah sebuah situs web yang dimiliki dan dioperasikan oleh Twitter Inc., yang menawarkan jejaring sosial berupa mikroblog sehingga memungkinkan penggunanya untuk mengirim dan membaca pesan yang

disebut kicauan (tweets). Kicauan adalah teks tulisan hingga 140 karakter yang ditampilkan pada halaman profil pengguna. Kicauan bisa dilihat secara luar, namun pengirim dapat membatasi pengiriman pesan ke daftar teman-teman mereka saja. Pengguna dapat melihat kicauan penulis lain yang dikenal dengan sebutan pengikut ("follower").

Semua pengguna dapat mengirim dan menerima kicauan melalui situs Twitter, aplikasi eksternal yang kompatibel (telepon seluler), atau dengan pesan singkat (SMS) yang tersedia di negara-negara tertentu. Situs ini berbasis di San Bruno, California dekat San Francisco, di mana situs ini pertama kali dibuat. Twitter juga memiliki server dan kantor di San Antonio, Texas dan Boston, Massachusetts.

Sejak dibentuk pada tahun 2006 oleh Jack Dorsey, Twitter telah mendapatkan popularitas di seluruh dunia dan saat ini memiliki lebih dari 100 juta pengguna. Hal ini kadang-kadang digambarkan sebagai "SMS dari internet".

Twitter memiliki logo berupa seekor burung berwarna biru yang bernama "Larry the "Bird", dinamai setelah nama seorang mantan pemain basket NBA, Larry Bird.

2.2 Teori Khusus

Teori-teori khusus yang berhubungan dengan topik dalam skripsi ini adalah :

2.2.1 Object Oriented Programming (OOP)

2.2.1.1 Pengertian OOP

Menurut Deitel (2008, p446),*Object Oriented Programming (OOP)* adalah teknik membuat suatu program yang terdiri dari objek-objek yang saling berinteraksi.

2.2.1.2 Konsep-Konsep dalam OOP

Menurut Deitel (2008, p452), konsep-konsep dari *Object Oriented Programming* adalah:

1. *Encapsulation*

   Encapsulation sering disebut dengan komponen atau modul.Fungsinya yaitu membuat sebuah kumpulan kode dalam sebuah objek menjadi objek sendiri. Dalam konteks OOP, *encapsulation* sering disebut kotak hitam, yang berarti dapat melihat sebuah benda bekerja meskipun tidak melihat cara kerja detail di dalamnya.

   Encapsulation yaitu menyembunyikan proses dari pihak luar atau objek lain untuk menyederhanakan sebuah sistem. *Encapsulation* juga memisahkan antara bagian publik (bisa dilihat oleh pihak luar) dan bagian *private* (tidak bisa dilihat oleh pihak luar), tujuannya agar dapat bekerja dengan aspek internal tanpa bergantung dengan aspek eksternal.

2. *Inheritance*

Inheritance mengarah kepada bagaimana sebuah kelas dari suatu objek menurunkan *properties* ,metode dan *events* dari kelas yang lain. Jika kelas A memiliki metode X, Y, Z dan kelas B adalah *sub*-kelas dari kelas A, maka kelas B juga memiliki metode X, Y, Z seperti kelas A.

Inheritance (pewarisan sifat) merupakan suatu objek yang memiliki perilaku sama dengan objek lain, sehingga dengan hubungan ini, suatu *class* dapat didefinisikan melalui *class* yang lain.

3. *Polymorphism*

Polymorphism adalah sebuah kata yang mendeskripsikan proses metamorfosis. *Metamorphoun* diambil dari bahasa Latin yang artinya berubah bentuk, sebuah kelas bisa mewakili berbagai bentuk tipe data.*Polymorphism* yaitu suatu fitur yang sama dengan aksi yang berbeda.

2.2.2 PHP

Menurut Kadir (2008, p46), PHP (*Hypertext Processor*) adalah sebuah bahasa pemrograman berbasis *web* yang berbentuk *scipt*. PHP ditempatkan dan diproses di *server*. PHP adalah bahasa pemrograman yang didasarkan oleh bahasa C. Bahasa C berkembang menjadi Perl, dan Perl

adalah sumber dari PHP. Oleh karena itu, struktur pemrograman yang ada di PHP sama dengan yang ada di C. PHP bersifat *server side*, sehingga untuk menjalankan PHP pada *browser* harus menginstal *web server* seperti Apache.PHP dapat dihubungkan dengan *database* seperti MySQL, SQL server, Oracle dan lain-lain.

2.2.3 MySQL

Menurut anonim 3, MySQL merupakan *software database* yang bersifat *open source* dan paling popular saat ini, dengan lebih dari 100 juta salinan *software* unduhan. Beberapa kelebihan MySQL seperti kecepatan yang tinggi, kehandalan dan penggunaan yang mudah, membuat MySQL menjadi pilihan utama bagi banyak perusahaan telekomunikasi dan manajer perusahaan IT.Banyak organisasi besar di dunia dan perusahaan menggunakan MySQL untuk menghemat waktu dan biaya dalam mendukung *website*, sistem bisnis dan *software*.Beberapa pemimpin industri yang menggunakan MySQL yaitu Yahoo!®, Google®, Nokia®, Youtube® dan lain-lain.

MySQL ditemukan dan dikembangkan di Swedia oleh David Axmark, Allan Larsson, dan Michael “Monty” Widenius. Mereka telah bekerja sama sejak tahun 1980-an. MySQL merupakan bagian terpenting dari LAMP (Linux, Apache, MySQL, PHP/Perl). Perusahaan banyak menggunakan LAMP sebagai alternatif dari *software* yang mahal, dikarenakan biaya LAMP yang rendah dan kebebasan dari *platform.*

2.2.4 AJAX(Asynchronous JavaScript and XML)

Menurut Th0R (2008, p133), AJAX adalah sebuah teknik atau fitur yang dapat digunakan untuk membantu pembangunan suatu *website.*Tujuannya yaitu untuk membangun sebuah *website* yang lebih responsif atau cepat tanggap, dengan melakukan pertukaran data dalam ukuran lebih kecil dengan server secara diam-diam.

Aplikasi AJAX membuat proses antara antarmuka penggunadengan respon aplikasi lebih cepat dengan menambahkan suatu *layer* diantara antarmuka pengguna dan komunikasi dengan *server.* Aplikasi AJAX yang cepat ini membuat penggunamenjadi terbiasa dengan respon yang cepat dengan mengadopsi metodologi AJAX, sehingga AJAX menyebar dengan sangat cepat melalui *website-website*, salah satu contohnya adalah *Google Maps.*

2.2.5 HTML

Menurut Turban (2007, p436), HTML *(Hypertext Markup Language)* merupakan standar bahasa pemrograman yang dipakai untuk membuat *web* dan mengenali dokumen *hypertext.*2 fitur penting yang dimiliki oleh HTML yaitu *hypertext* dan *universality.*

Menurut Castro (2007, p12), *hypertext* adalah suatu pendekatan untuk pengelolaan data dimana data disimpan dalam jaringan node dihubungkan dengan *link (hyperlink). Hypertext* dapat membuat link pada halaman *web* yang mengarahkan pengguna ke halaman *web* lain atau untuk hal praktis apapun di internet, ini berarti informasi dalam *web* dapat diakses

dari sember yang berbeda. *Universality* berarti semua komputer dapat membaca suatu halaman *web*, karena dokumen HTML yang merupakan dasar pembuatan *web* disimpan dalam bentuk *file Text Only,* sehingga *web* terbuka untuk semua.

Struktur HTML :

1. *Tag*

   Tag adalah tanda yang digunakan untuk menentukan tampilan suatu teks. *Tag* harus ditulis secara berpasangan, jika ada *tag* pembuka maka harus ada *tag* penutup.

   Contoh *tag* : *tag* pembuka <body> dan *tag* penutup </body>

2. Element

   Element terdiri dari *tag* pembuka, isi dan *tag* penutup. *Tag-tag* harus tersusun dengan baik.

   Contoh *element* : *tag* pembuka <body> isi tulisan *tag* penutup </body>

3. Attribute

   Attribute terdiri dari nama dan nilai, dimana nilai *attribute* tersebut berada dalam tanda petik.

   Contoh *attribute* : membuat warna latar halaman web menjadi biru dengan warna tulisan putih <body bgcolor="blue" text="white">.

4. *Element* HTML

   Element HTML digunakan untuk menyatakan kepada *browser* bahwa dokumen web yang dipakai adalah HTML.

Penulisan *element HTML* :<html></html>

2.2.6 Design Pattern

Menurut Shi (2007, p43), *design pattern* biasanya digunakan untuk menjembatani komunikasi antar *developers* dalam menggabungkan ide-ide mereka. Setiap *pattern* yang didefinisikan *developers*harus dijabarkan secara sederhana dan mudah dimengerti untuk menjelaskan *design pattern* tersebut. Karena itu, *design pattern* menggunakan *visual language* yang merupakan gabungan dari shapes, lines, dan *text* untuk mempermudah penjabaran dari setiap *entities* dan hubungan antar *entities.*Berdasarkan hal tersebut, *design pattern* menggunakan *syntax* dari UML yang merupakan *modelling language* yang biasa digunakan dalam mendefinisikan model dari *design patterns.*

Contoh dari *single pattern* yaitu :

1. *Singleton*

   Sebuah *design pattern* dikatakan *singleton* jika tiga hal berikut ini terpenuhi :

   a. Kepemilikan dari *single instance* tidak dapat di *assigned.*

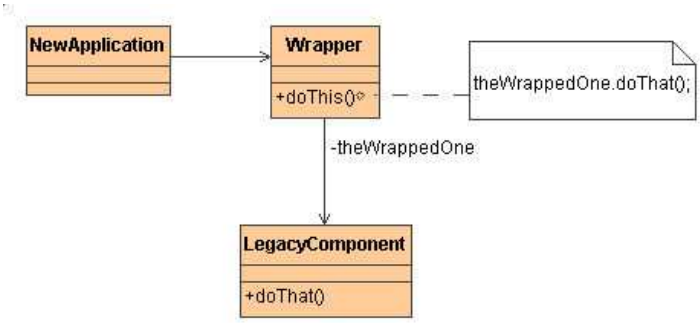
   b. Diinginkan inisialisasi yang minimal.

   c. Tidak terdapat akses global

Gambar 2. 3 *Singleton*

2. *Adapter*

Tipe adapter ini dirancang agar segala tugas dikerjakan setelah semuanya selesai dirancang. Terdapat bridge yang dirancang agar abstraksi dan implementasi dapat berdiri sendiri. Adapter digunakan untuk menjembatani class yang tidak berhubungan dapat bekerja sama.

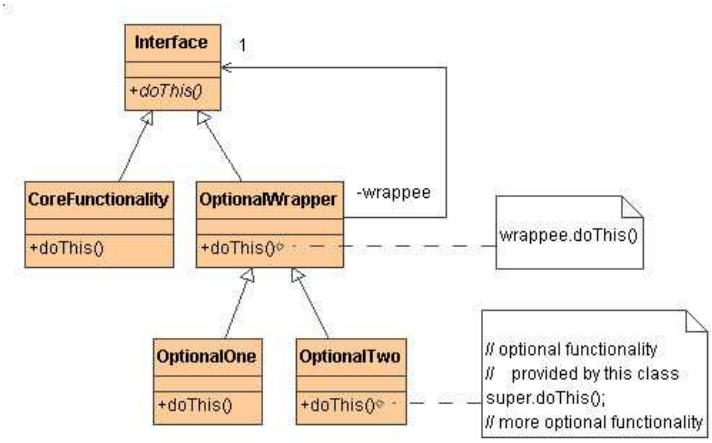


Gambar 2. 4 *Adapter*

3. Decorator

Tipe *decorator* ini dirancang untuk memberikan tanggung jawab pada objek tanpa *subclassing.*Fokus komposit bukan berada pada rancangan tapi tiap-tiap representasi. Konsikuensinya, komposit dan dekorator mempunyai cara pandang yang berbeda namun dapat saling melengkapi.

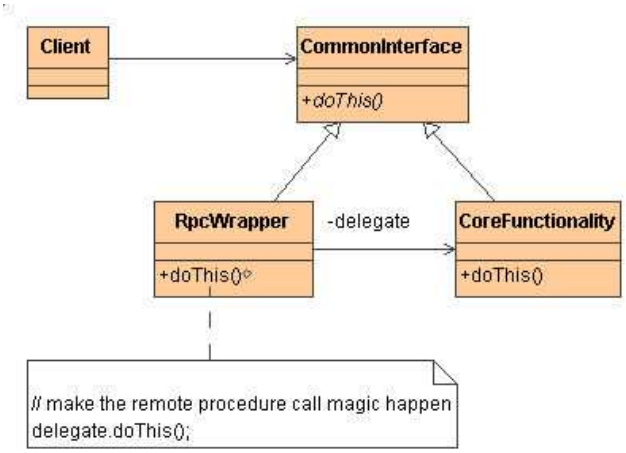
Karena itu, tipe komposit dan dekorator biasa digunakan dalam projek *event organizer.*



Gambar 2. 5 *Decorator*

4. Proxy

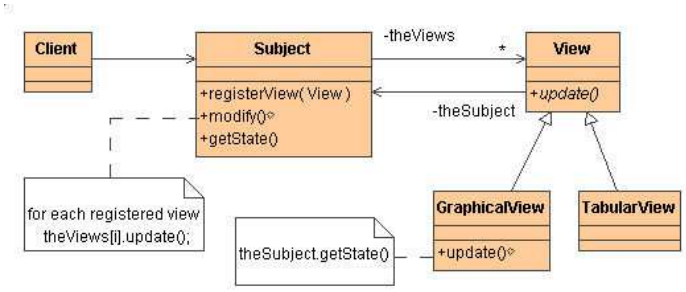
Decorator dan *Proxy* mempunyai tujuan yang berbeda tapi mempunyai struktur yang sama. Kedua tipe ini menjelaskan bagaimana tingkatan *indirection* ke objek lainnya, dan implementasi yang tetap menjaga referensi ke tiap objek dimana *request* akan diteruskan.*Adapter* menyediakan tampilan yang berbeda ke setiap subjek.*Proxy* menyediakan tampilan yang sama. *Decorator* memberikan tampilan yang ditingkatkan per *level.*



Gambar 2. 6 *Proxy*

5. *Observer*

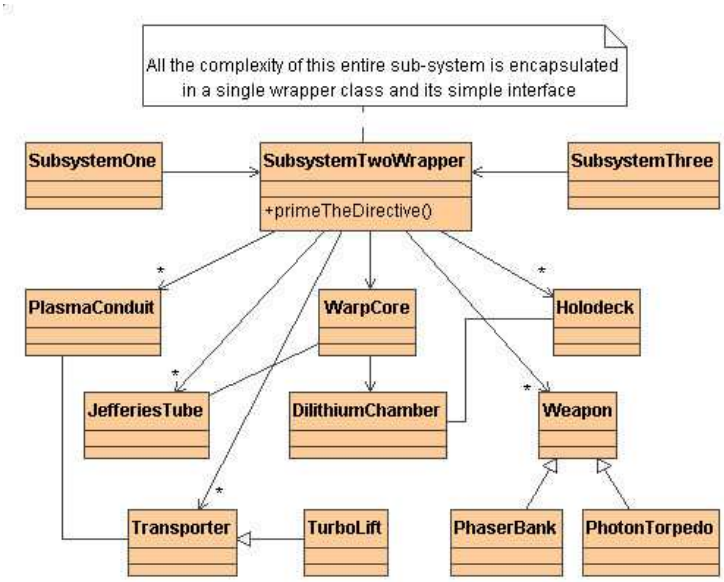
Mediator dan *Observer* merupakan *patterns* yang berbeda. Perbedaan keduangan adalah *Observer* mendistribusikan komunikasi melalui mendefinisikan "*observer*" dan "*subject*" sebagai objek, sedangkan *Mediator* melakukan enkapsulasi objek pada saat berkomunikasi diantara objek-objek lainnya.*Observer* secara dinamis mendaftarkan setiap *colleagues*dan berkomunikasi dengan *colleagues* tersebut.



Gambar 2. 7 *Observer*

6. *Facade*

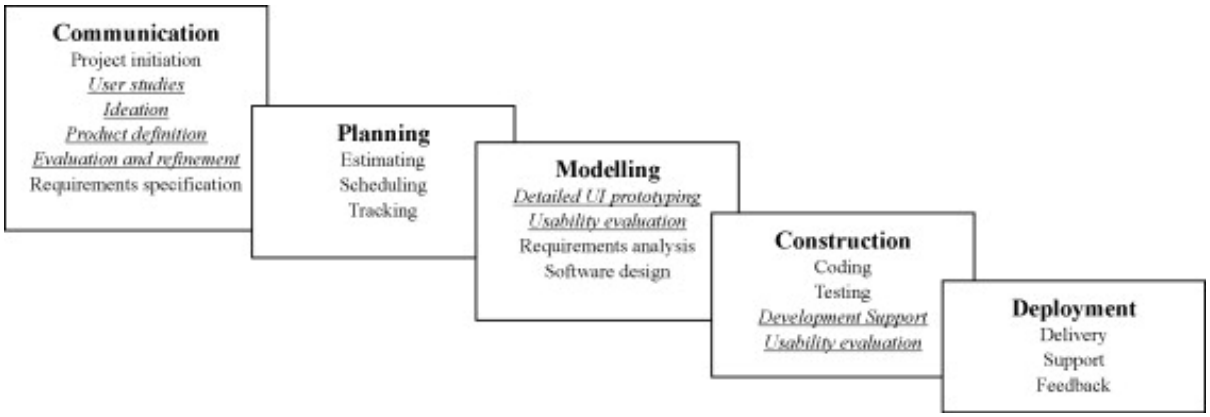
Facade mendefinisikan tampilan yang baru, sedangkan *Adapter* menggunakan tampilan yang lama. Perlu diingat bahwa *adapter* membuat dua tampilan yang ada bekerja bersama sebagai perbandingan terhadap *facade* yang hanya mendefinisikan tampilan baru.



Gambar 2. 8 *Façade*

2.2.7 *Waterfall Model*

Menurut Pressman (2010, p39) model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software.* Berikut ini ada dua gambaran dari *waterfall* model.



Gambar 2. 9 *Waterfall Model*

Fase – fase dalam model *waterfall* menurut referensi Pressman:

1. *Communication*

   Langkah ini merupakan analisis terhadap kebutuhan *software,* dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan *customer,* maupun mengumpulkan data – data tambahan baik yang ada di jurnal, artikel, maupun internet

2. *Planning*

   Proses *planning* merupakan kelanjutan proses *communication (analysis requirement).* Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan *software,* termasuk rencana yang akan dilakukan.

3. *Modelling*

   Proses *modelling* ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan *software* yang dapat diperkirakan sebelum dibuat *coding.* Proses ini berfokus pada rancangan struktur data, arsitektur *software,* representasi *interface,* dan detail (algoritma) prosedural. Tahapan ini akan mengahasilkan dokumen yang disebut *software requirement.*

4. *Construction*

   Construction merupakan proses membuat kode. *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjamahkan transaksi yang diminta oleh *user.* Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software,* artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan – kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki

5. *Deployement*

   Tahapan ini bisa dikatan final dalam pembuatan sebuah *Software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user.* Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

Adapun keunggulan dan kelemahan *Waterfall model* yaitu:

1. Keunggulan

   Memberikan template tentang metode analisis, desain, pengkodean, pengujian, dan pemeliharaan.

2. Kelemahannya

   Jarang sekali proyek riil mengikuti aliran sekuensial yang dianjurkan model karena model ini bisa melakukan itersi tidak langsung . Hal ini berakibat ada perubahan yang diragukan pada saat proyek berjalan. Sebuah kesalahan jika tidak diketahui dari awal akan menjadi masalah besar karena harus mengulang dari awal. Membutuhkan waktu pengerjaan proyek yang lama.