

BAB II

TINJAUAN PUSTAKA

II.1. Perancangan

Untuk membuat tampilan yang menarik memang tidak mudah dilakukan. Seorang perancang tampilan selain harus mempunyai jiwa seni yang memadai, ia juga harus mengerti selera pengguna secara umum. Hal lain yang perlu disadari oleh seorang perancang tampilan adalah bahwa ia harus bisa meyakinkan pemrogramnya bahwa apa yang ia bayangkan dapat diwujudkan dengan peranti bantu yang tersedia (Insap Santoso ; 2009 : 185)

Perancangan merupakan proses pengolahan hasil analisis perangkat lunak menjadi rencana pengembangan perangkat lunak dan batasan-batasan perangkat lunak atau masalah yang mungkin dihadapi dalam pengembangan perangkat lunak. Perancangan yang dilakukan meliputi perancangan arsitektur, perancangan modul, dan perancangan antarmuka. (Y. Yohakim Marwanta ; 2010 : 5).

Bagi perancang antarmuka, hal yang sangat penting untuk ia perhatikan adalah mendokumentasikan semua pekerjaan yang dilakukan. Dokumentasi rancangan dapat dikerjakan atau dilakukan dengan beberapa cara :

1. Membuat sketsa pada kertas
2. Menggunakan peranti purwarupa GUI
3. Menuliskan keterangan yang menjelaskan tentang kaitan antara jendela.
4. Menggunakan peranti bantu CASE (*Computer Aided Software Engineering*).

Cara kedua dan keempat tidak selalu dapat diterapkan, karena peranti tersebut biasanya harus dibeli dan seringkali cukup mahal. Cara ini kebanyakan diterapkan pada pembuatan antarmuka grafis untuk suatu jenis pekerjaan berskala besar.

II.1.1. Cara Pendekatan

Sebuah program aplikasi pastilah ditujukan kepada pengguna, yang utama, bukan perancangan program aplikasi tersebut. Program aplikasi pada dasarnya dapat dikelompokkan dalam dua kategori besar, yakni program aplikasi untuk keperluan khusus dengan pengguna yang khusus pula dan program aplikasi yang akan digunakan oleh pengguna umum, yang juga sering dikenal dengan sebutan public software. Karena perbedaan pada calon pengguna, maka perancang program antarmuka perlu memperhatikan hal ini (Insap Santoso ; 2009 : 186).

Pada kelompok pertama, yakni pada program aplikasi untuk keperluan khusus, misalnya program aplikasi untuk inventori gudang, pengelolaan data akademis mahasiswa, pelayanan reservasi hotel, dan program-program aplikasi yang serupa, kelompok calon pengguna yang akan memanfaatkan program aplikasi tersebut dapat dengan mudah diperkirakan, baik dalam hal keahlian pengguna maupun ragam antarmuka yang akan digunakan. Untuk kelompok ini ada satu pendekatan yang dapat dilakukan, yakni pendekatan yang disebut dengan pendekatan perancangan berpusat ke pengguna (*user centered design approach*). Cara pendekatan ini berbeda pendekatan perancangan oleh pengguna (*user design approach*).

Pendekatan perancangan berpusat ke pengguna adalah perancangan antarmuka yang melibatkan pengguna. Pelibatan pengguna di sini tidak diartikan bahwa pengguna harus ikut memikirkan bagaimana implementasinya nanti, tetapi pengguna diajak untuk aktif berpendapat ketika perancangan antarmuka sedang menggambar wajah antarmuka yang mereka inginkan. Dengan kata lain, perancangan dan pengguna duduk bersama-sama untuk merancang wajah antarmuka yang diinginkan pengguna. Pengguna menyampaikan keinginannya. Sementara perancangan menggambar keinginan pengguna tersebut sambil menjelaskan keuntungan dan kerugian wajah antarmuka yang diinginkan oleh pengguna, seolah-olah sudah mempunyai gambaran nyata tentang antarmuka yang nanti akan mereka gunakan (Insap Santoso ; 2009 : 187).

Pada perancangan oleh pengguna, pengguna sendirilah yang merancang wajah antarmuka yang diinginkan. Di satu sisi, cara ini akan mempercepat proses pengimplementasian modul antarmuka. Tetapi di sisi yang lain, hal ini justru sangat memberatkan pemrogram karena apa yang diinginkan pengguna belum tentu dapat diimplementasikan dengan mudah, atau bahkan tidak dapat dikerjakan dengan menggunakan peranti bantu yang ada.

Perancang program aplikasi yang dimasukkan dalam kelompok kedua, atau *public software*, perlu menganggap bahwa program aplikasi tersebut akan digunakan oleh pengguna dengan berbagai tingkat kepandaian dan karakteristik yang sangat beragam. Di satu sisi keadaan ini dapat ia gunakan untuk memaksa pengguna menggunakan antarmuka yang ia buat, tetapi pada sisi lain pemaksaan itu akan berakibat bahwa program aplikasinya menjadi tidak banyak

penggunanya. Satu kunci penting dalam pembuatan modul antarmuka untuk program-program aplikasi pada kelompok ini adalah dengan melakukan customization. Dengan customization pengguna dapat menggunakan program aplikasi dengan wajah antarmuka yang sesuai dengan selera masing-masing pengguna.

Salah satu contoh dari adanya kemampuan yang dimiliki oleh sebuah program aplikasi atau sistem operasi yang dapat disesuaikan dengan karakteristik pengguna adalah pengaturan desktop pada OS X versi 10.5 favoritnya, sehingga pengguna dapat mengubahnya sesuai keinginan justru akan membuat mata pengguna itu sakit, dikarenakan mata harus melakukan akomodasi maksimum terus menerus untuk menyesuaikan dengan warna tampilan yang ada.

Selain cara pendekatan yang dijelaskan di atas, Anda yang terbiasa menulis program-program aplikasi mungkin mempunyai cara khusus untuk berhadapan dengan pengguna. Tetapi perlu Anda ingat bahwa apapun cara yang Anda gunakan, Anda tetap harus mempunyai pedoman bahwa pada akhirnya program itu bukan untuk Anda sendiri, tetapi akan digunakan oleh orang lain. Dengan kata lain, jangan pernah mengabaikan pendapat (calon) pengguna program aplikasi Anda.

II.1.2. Prinsip Dan Petunjuk Perancangan

Antarmuka pengguna secara alamiah terbagi menjadi empat komponen model pengguna, bahasa perintah, umpan balik, dan penampilan informasi. Model pengguna merupakan dasar dari tiga komponen yang lain. (Insap Santoso ; 2009 : 188)

Model mental pengguna merupakan model konseptual yang dimiliki oleh pengguna ketika ia menggunakan sebuah sistem atau program aplikasi. Model ini memungkinkan seorang pengguna untuk mengembangkan pemahaman mendasar tentang bagian yang dikerjakan oleh program, bahkan oleh pengguna yang sama sekali tidak mengetahui teknologi komputer. Dengan pertolongan model itu pengguna dapat mengantisipasi pengaruh suatu tindakan yang dilakukan dan dapat memilih strategi yang cocok untuk mengoperasikan program tersebut. Model pengguna dapat berupa suatu simulasi tentang keadaan yang sebenarnya dalam dunia nyata, sehingga ia tidak perlu mengembangkannya sendiri dari awal.

Setelah pengguna mengetahui dan memahami model yang diinginkan, dia memerlukan peranti untuk memanipulasi model itu. Peranti pemanipulasian model ini sering disebut dengan bahasa perintah (*command language*), yang sekaligus merupakan komponen kedua dari antarmuka pengguna. Idealnya program komputer kita mempunyai bahasa perintah yang alami, sehingga model pengguna dengan cepat dapat dioperasionalkan.

Komponen ketiga adalah umpan balik. Umpan balik di sini diartikan sebagai kemampuan sebuah program yang membantu pengguna untuk mengoperasikan program itu sendiri. Umpan balik dapat berbentuk pesan penjelasan, pesan penerimaan perintah, indikasi adanya obyek terpilih, dan penampilan karakter yang diketikkan lewat papan ketik. Beberapa bentuk umpan balik terutama ditujukan kepada pengguna yang belum berpengalaman dalam menjalankan program sebuah aplikasi. Umpan balik dapat digunakan untuk

member keyakinan bahwa program telah menerima perintah pengguna dan dapat memahami maksud perintah tersebut.

Komponen keempat adalah tampilan informasi. Komponen ini digunakan untuk menunjukkan status informasi atau program ketika pengguna melakukan suatu tindakan. Pada bagian ini perancang harus menampilkan pesan-pesan tersebut seefektif mungkin sehingga mudah dipahami oleh pengguna. Setelah memahami beberapa prinsip dalam perancangan antarmuka pengguna. Pada bagian berikut ini akan diberikan petunjuk singkat tentang perancangan antarmuka yang akan Anda lakukan sebagai seorang perancang tampilan.

II.1.3. Urutan Perancangan

Perancangan dialog, seperti halnya perancangan sistem yang lain, harus dikerjakan secara atas ke bawah. Proses perancangannya dapat dikerjakan secara bertahap sampai rancangan yang diinginkan terbentuk. Adapun urutan rancangan tersebut adalah sebagai berikut : (Insap Santoso ; 2009 : 190)

1. Pemilihan ragam dialog

Untuk suatu tugas tertentu, pilihlah ragam dialog yang menurut perkiraan cocok untuk tugas tersebut. Ragam dialog dapat dipilih dari sejumlah ragam dialog yang telah dijelaskan pada bab-bab sebelumnya. Pemilihan ragam dialog dipengaruhi oleh karakteristik populasi pengguna, tipe dialog yang diperlukan, dan kendala teknologi yang ada untuk mengimplementasikan ragam dialog tersebut. Ragam dialog yang terpilih dapat berupa sebuah ragam tunggal, atau sekumpulan ragam dialog yang satu sama lain saling mendukung.

2. Perancangan Struktur Dialog

Tahap kedua adalah melakukan analisis tugas dan menentukan model pengguna dari tugas tersebut untuk membentuk struktur dialog yang sesuai. Dalam tahap ini pengguna sebaiknya banyak dilibatkan, sehingga pengguna langsung mendapatkan umpan balik dari diskusi yang terjadi. Pada tahap ini suatu purwarupa dialog seringkali dibuat untuk memberik gambaran yang lebih jelas kepada calon pengguna.

3. Perancangan format pesan

Pada tahap ini tata letak tampilan dan keterangan tekstual secara terinci harus mendapat perhatian lebih. Selain itu, kebutuhan data masukan yang mengharuskan pengguna untuk memasukkan data ke dalam komputer juga harus dipertimbangkan dari segi efisiensinya. Salah satu contohnya adalah dengan mengurangi pengetikan yang tidak perlu dengan cara mengefektifkan pengguna tombol.

II.2. Multimedia

Multimedia, ditinjau dari bahasanya, terdiri dari 2 kata, yaitu multi dan media. Multi memiliki arti banyak atau lebih dari satu. Sedangkan media merupakan bentuk jamak dari mediun, juga diartikan sebagai saran, wadah, atau alat. Istilah multimedia dapat diartikan sebagai transmisi data dan manipulasi semua bentuk informasi, baik berbentuk kata-kata, gambar, video, music, angka, atau tulisan tangan di mana dalam dunia komputer, bentuk informasi tersebut diolah dari dan dalam bentuk data digital (Darma, dkk ; 2012:1).

Banyak orang mempresentasikan mengenai pengertian dari multimedia itu sendiri. Dalam industry elektronika, multimedia adalah kombinasi dari computer dan video atau multimedia secara umum merupakan kombinasi tiga elemen, yaitu: suara, gambar dan teks. Di sisi lain, multimedia adalah kombinasi dari paling sedikit dua media *input* atau *output* dari data, di mana media tersebut dapat berupa *audio* (suara, musik), animasi, *video*, teks, grafik dan gambar atau multimedia merupakan alat yang menciptakan presentasi yang dinamis dan interaktif yang mengkombinasikan teks, grafik, animasi, *audio* dan gambar *video*.

Multimedia dimanfaatkan juga dalam dunia pendidikan dan bisnis. Di dunia pendidikan, multimedia digunakan sebagai media pengajaran, atau media persentasi, baik dalam kelas maupun secara sendiri-sendiri. Di dunia bisnis, multimedia digunakan sebagai media profil perusahaan, profil produk, bahkan sebagai media kios informasi dan pelatihan dalam *system elearning*.

Multimedia dimanfaatkan menggunakan aplikasi multimedia. Perangkat-perangkat lunak tersebut secara umum bekerja mengolah data digital dapat diterjemahkan dan ditampilkan, sehingga para pemakai dapat melihat dan memahami isi dari informasi yang terdapat dari multimedia tersebut. Perangkat lunak ini bermacam-macam, tergantung dari jenis multimedia itu sendiri, karena sangat berkaitan dengan *format* data yang digunakan. Aplikasi multimedia umumnya dipisahkan lagi menjadi aplikasi yang digunakan untuk membuat, yang hanya digunakan untuk menampilkan saja dan aplikasi pengaturan. Jika ingin menikmati multimedia di komputer, setidaknya harus memiliki fasilitas sebagai berikut : (Darma, dkk; 2012: 2)

1. Perangkat lunak/aplikasi multimedia.

Perangkat lunak ini digunakan untuk menjalankan fungsi multimedia pada komputer. Misalnya, *windows media player* yang dapat digunakan untuk menjalankan CD dan DVD pada komputer kita.

2. CD/DVD ROM

CD/DVD ROM digunakan untuk memutar berbagai jenis CD, VCD, dan DVD.

3. *Sound Card*

Sound card (kartu suara) adalah perangkat yang terhubung pada papan induk (*motherboard*) yang berfungsi sebagai alat untuk mengolah dan mengontrol suara, baik suara yang masuk (merekam) dan suara yang keluar melalui *speaker* tersebut. Hal ini dimungkinkan karena pada *sound card* terdapat masukan (*Line in*, *Mic*, dan *MIDI*) serta keluaran (*line out/speaker out*) pada perangkat tersebut.

4. Kartu grafis (*Graphic Card / Display Adapter*)

Kartu grafis merupakan perangkat yang terhubung langsung di papan induk komputer yang berfungsi untuk mengolah citra (gambar) agar mempunyai kualitas yang baik. Saat ini kartu grafis yang sering digunakan adalah kartu grafis yang menggunakan teknologi AGP (*Accelerated Graphics Port*) dengan kapasitas diatas 128 MB.

5. *Speaker*

Speaker (pengeras suara) merupakan perangkat output untuk menghasilkan suara.

II.2.1. Konsep Rancangan

Multimedia secara sederhana dapat diartikan sebagai lebih dari satu media yaitu kombinasi antara teks, grafik, animasi, suara, video. Penggabungan ini merupakan suatu kesatuan yang secara bersama-sama menampilkan informasi, pesan, atau isi materi. (Iva Khususia Hidayat, 2012:4).

Dalam perancangan multimedia interaktif ini, dibutuhkan persiapan yang sangat rumit. Ada beberapa hal yang dipertimbangkan dalam pembuatan multimedia interaktif ini diantaranya :

1. Multimedia interaktif ini ditujukan untuk orang dewasa dalam masyarakat luas, tentunya gaya desain dan penyampaiannya disesuaikan dengan karakter-karakter orang dewasa. Diantaranya desain dibuat tidak terlalu rame dan sedikit menggunakan efek-efek visual, kemudian penataan tipografi yang dibuat sederhana tapi tepat langsung pada bahasan.
2. Unsur media dalam mewarnai gambar 2 dimensi yang digunakan adalah teks, gambar, animasi, dan audio. Animasi yang digunakan cukup sederhana dan tidak terlalu rame.
3. Mewarnai gambar 2 dimensi ini dirancang dengan pengoperasian secara interaktif dan komunikatif. Dimana para pengguna atau user bisa secara langsung memilih dan menekan tombol menu yang telah disajikan sesuai dengan keinginannya.

Multimedia interaktif ini, memiliki konsep desain modern dengan fasilitas memberikan objek untuk diberikan hiasan sesuai warna yang diinginkan. Agar interaksi antar komputer dapat berjalan dengan baik.

II.2.2. Konsep Ruang Desain

Dalam kehidupan sehari-hari sudah sangat terbiasa dengan ruang 3 dimensi, contohnya ketika menanyakan teman dimana letak sebuah kotak pensil, seseorang itu menunjukkan kalau kotak itu berada di atas meja, di belakang kotak yang agak besar, dan dengan cepat menemukannya dan cepat menilai bentuk, ukuran, ketebalan warna dan sebagainya. (Arif Ramadhan; 2012: 1).

Coba bandingkan ketika memiliki gambar kotak pensil yang sudah discan dan dibuka di komputer bersama tumpukan *window* dan kotak dialog yang buka lainnya. Saat seseorang tersebut menanyakan kepada teman dimana gambar kotak pensil itu, teman pasti bilang mungkin berada dibelakang *windows explorer* yang sedang dibuka, dan akan mulai mencari di belakang satu per satu *window* yang dibuka atau bisa melihat dari taskbar *windows*.

Dari dua kasus ini bisa menyimpulkan bahwa komputer hanya memiliki ruang dua dimensi yang terdiri dari panjang dan lebar, sedangkan di dunia nyata memiliki ruang 3 dimensi yang terdiri dari panjang lebar dan ketebalan. Lalu bagaimana komputer bisa menyajikan objek 3 dimensi jawabnya adalah dengan melakukan pendekatan konsep ruang 3 dimensi di mana bisa membuat komputer seolah-olah mampu menampilkan objek 3 dimensi. Dari kemampuan inilah kemudian melahirkan desain 3 dimensi dalam dunia komputer.

Konsep desain dalam komputer berarti bahwa setiap objek yang dibangun, akan ditempatkan pada ruang yang memiliki media bidang. Media bidang itu dapat berupa bidang flat atau yang kemudian disebut 2 dimensi (2D),

terdiri dari dua sumbu koordinat dan berupa bidang dimensional atau juga disebut 3 dimensi (3D) terdiri dari tiga sumbu koordinat.

II.3. Animasi

Animasi adalah suatu teknik yang banyak sekali digunakan dalam dunia film dewasa ini, baik sebagai suatu kesatuan yang utuh, bagian dalam suatu film maupun bersatu dalam *film live*. Dunia film sebenarnya berakar dari fotografi, sedangkan animasi berakar dari dunia gambar, yaitu ilustrasi design grafis (desain komunikasi visual). Dapat dikatakan bahwa animasi merupakan suatu media yang lahir dari dua konversi atau disiplin, yaitu film dan gambar. Untuk dapat mengerti dan memakai teknik animasi, dua konversi tersebut harus dipahami dan dimengerti. (Yunita Syahfitri, lppm.trigunadharma.ac.id ; 2011 : 213)

Film biasanya dipakai untuk merekam suatu keadaan atau mengemukakan sesuatu. Film digunakan untuk memenuhi suatu kebutuhan umum yaitu mengkomunikasikan suatu gagasan, pesan atau kenyataan. Karena keunikan dimensinya dan karena sifat hiburannya, film telah diterima sebagai salah satu media audio visual yang paling populer dan sebagai media yang paling efektif.

Keinginan manusia untuk membuat gambar atau santiran (*image*) yang hidup dan bergerak sebagai perantara dari pengungkapan (*expression*) mereka, merupakan perwujudan dari bentuk dasar animasi yang hidup berkembang. Kata *animasi* itu sendiri sebenarnya penyesuaian dari kata *animation* yang berasal dari kata dasar *to animate* dalam kamus umum Inggris – Indonesia berarti menghidupkan. secara umum animasi adalah suatu kegiatan menghidupkan,

menggerakkan benda mati. Suatu benda mati diberikan dorongan kekuatan, semangat dan emosi untuk menjadi hidup dan bergerak atau hanya berkesan hidup.

II.3.1. Animasi Masking Strip

Animasi masking dipahami sebagai animasi yang menggunakan objek tertentu untuk membedakan mana yang akan terlihat secara utuh di dalam stage dan yang tidak. Animasi masking dapat diibaratkan seperti ketika mengenakan topeng yang hanya memiliki satu lubang mata saja. Yang terlihat di depan hanyalah pemandangan yang terlihat di satu mata itu. Begitu pun animasi masking. Animasi masking dapat diwakili oleh objek persegi empat, garis, lingkaran, atau objek berbentuk bebas lainnya. Hanya saja yang terlihat di dalam stage hanyalah pemandangan yang tertembus oleh objek itu. (Jubilee Enterprise ; 2012 : 1)

Animasi sendiri berasal dari bahasa latin yaitu *anima* yang berarti jiwa, hidup, semangat. Sedangkan karakter adalah orang, hewan maupun objek nyata lainnya yang dituangkan dalam bentuk gambar 2D maupun 3D. Sehingga karakter animasi dapat diartikan sebagai gambar yang memuat objek yang seolah-olah hidup, disebabkan kumpulan gambar itu berubah beraturan dan bergantian ditampilkan. Objek dalam gambar dapat berupa tulisan, bentuk benda, warna.

II.3.2. Animasi Masking Blok

Animasi masking blok yaitu memanfaatkan blok-blok persegi empat yang menggeser satu foto sehingga menampakkan lapisan foto lainnya. Dengan demikian, yang dibutuhkan adalah beberapa fotoyang diletakkan berlapis-lapis,

dan bidang persegi empat yang di animasikan yang nanti menyapu foto-foto tersebut.

II.4. Adobe Flash CS4

Adobe Flash CS4 merupakan versi lanjutan dari Adobe Flash CS4 yang dirilis pada pertengahan oktober 2008. Secara umum, Adobe Flash CS4 memiliki fungsi yang dengan Flash versi-versi sebelumnya. Animasi ini bisa berupa iklan atau film animasi, variasi komponen-komponen halaman web, aplikasi berbasis internet, hingga teknologi game yang sedang marak akhir-akhir ini. Walaupun fungsi Flash CS4 secara umum sama dengan versi-versi sebelumnya, sebagai versi terbaru yang membawa fitur-fitur baru yang tidak dimiliki oleh Flash versi-versi sebelumnya. (Wahana Komputer; 2012: 1).

Adobe Flash merupakan program yang digunakan untuk membuat sebuah objek bergerak atau animasi. Program ini dapat dimanfaatkan untuk beberapa keperluan seperti pembuatan presentasi, animasi kartu, dan tampilan interaktif, serta digunakan sebagai program pendukung dalam pembuatan desain web.

Menggunakan dan membuat objek animasi dengan adobe flash dapat dilakukan secara mudah dan cepat. Ada 2 pilihan cara pembuatan animasi yang dapat digunakan, yaitu secara langsung atau menggunakan *actionscript*.

II.5. Unified Modeling Language (UML)

UML adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara

visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. (Haviluddin; 2013: 2)

Sejarah UML sendiri terbagi dalam dua fase sebelum dan sesudah munculnya UML. Dalam *fase* sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990an namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi.

Sebagian besar para perancang sistem informasi dalam menggambarkan informasi dengan memanfaatkan UML diagram dengan tujuan utama untuk membantu tim proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program. Secara filosofi UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* karena konsep ini menganalogikan system seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik. Diagram Uml dapat dilihat pada gambar II.1 seperti dibawah ini :



Gambar II.1. Diagram UML
(Sumber : Haviluddin; 2013 : 2)

II.5.1. Tujuan Pemanfaatan UML

Tujuan dari penggunaan diagram seperti diungkapkan oleh Schuller J. (2004), "*The purpose of the diagrams is to present multiple views of a system; this set of multiple views is called a model*". Berikut tujuan utama dalam desain UML adalah (Haviluddin; 2012:10)

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan *visual* yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan dalam membuat sistem.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu system yang dibuat dalam penyelesaian proyek.
7. Memiliki integrasi praktik terbaik.

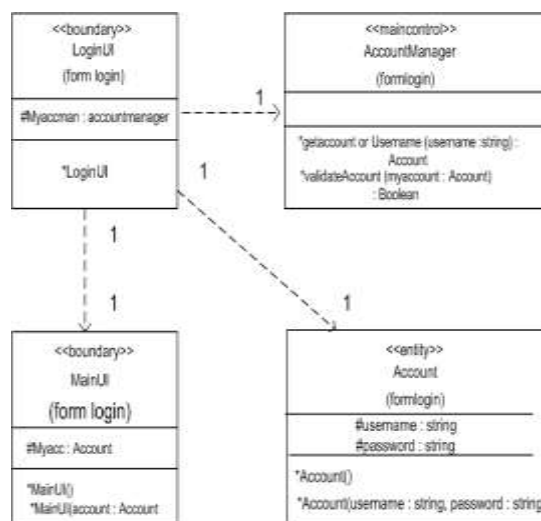
II.5.2. Struktur Diagram

Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem.

Struktur diagram dalam UML terdiri atas :

1. Class Diagram

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. *Class* memiliki tiga area pokok : nama (dan *stereo type*), atribut, metoda.







Gambar II.2. Notasi class diagram

(Sumber : Haviluddin; 2013 : 3)

II.6. Activity diagram

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas. Peralatan yang digunakan untuk membuat struktur diagram activity dapat dilihat pada Tabel II.1.

Tabel II.1. Activity Diagram

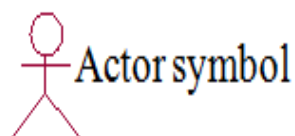
	<i>Activity</i>
	<i>Transition</i>
	<i>Decison</i>
	<i>Synchronization Bars</i>

(Sumber : Havaluddin; 2013 : 4)

UML memiliki seperangkat notasi yang akan digunakan ke dalam tiga kategori diatas yaitu struktur diagram, *behaviour* diagram dan *interaction* diagram. Berikut beberapa notasi dalam UML diantaranya :

1. *Actor*

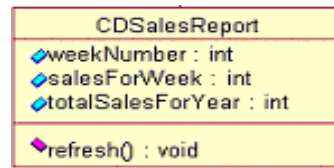
Actor menentukan peran yang dimainkan oleh *user* atau sistem lain yang berinteraksi dengan subjek. *Actor* adalah segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer, seperti orang, benda atau lainnya. Tugas *actor* adalah memberikan informasi kepada sistem dan dapat memerintahkan sistem untuk melakukan sesuatu tugas.

**Gambar II.3. Notasi actor**

(Sumber : Havaluddin; 2013 : 6)

2. *Class Diagram*

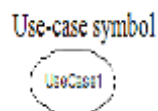
Notasi utama dan yang paling mendasar pada diagram UML adalah notasi untuk mempresentasikan suatu *class* beserta dengan atribut dan operasinya. *Class* adalah pembentuk utama dari sistem berorientasi objek



Gambar II.4. Notasi class
(Sumber : Havaluddin; 2013 : 6)

3. Use Case dan Use Case Specification

Use case adalah deskripsi fungsi dari sebuah sistem perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut skenario. *Use case* merupakan awal yang sangat baik untuk setiap fase pengembangan berbasis objek, *design*, *testing*, dan dokumentasi yang menggambarkan kebutuhan sistem dari sudut pandang di luar sistem. Perlu diingat bahwa *use case* hanya menetapkan apa yang seharusnya dikerjakan oleh sistem, yaitu kebutuhan fungsional sistem dan tidak untuk menentukan kebutuhan non- fungsional, misalnya: sasaran kinerja, bahasa pemrograman dan lain sebagainya.



Gambar II.5. Notasi use case
(Sumber : Havaluddin; 2013 : 6)

4. Realization

Realization menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah.



Gambar II.6. Notasi *realization*
(Sumber : Havaluddin; 2013 : 6)

5. *Interaction*

Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek.



Gambar II.7. Notasi *Interaction*
(Sumber : Havaluddin; 2013 : 6)

II.7. Algoritma *Depth First Search* (DFS)

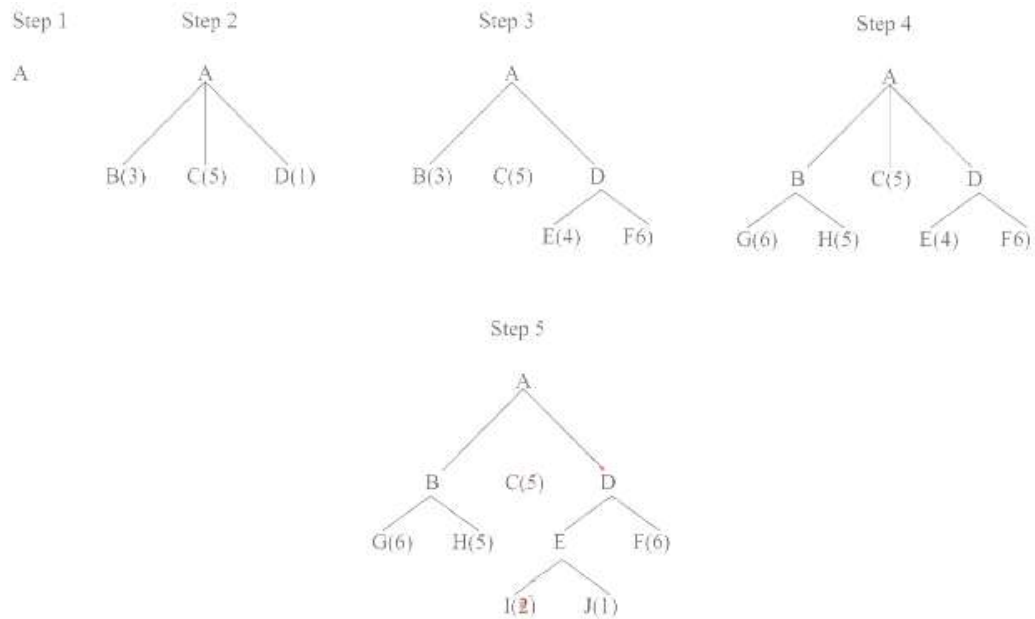
Sesuai dengan namanya, *Breadth First Search* membangkitkan simpul berikutnya dari sebuah simpul yang sejauh ini terbaik diantara semua *leaf nodes* yang pernah dibangkitkan. Pertanyaannya bagaimana menentukan sebuah simpul terbaik saat ini. Penentuan simpul terbaik dapat dilakukan menggunakan informasi berupa (Ayu Putri Sri Rezeki; 2015: 19).

Biaya perkiraan dari suatu simpul menuju ke goal atau gabungan antara biaya sebenarnya dan biaya perkiraan tersebut. Biaya perkiraan dapat diperoleh dengan menggunakan fungsi heuristic. Oleh karena itu algoritma *Breadth First Search* ini digolongkan sebagai algoritma heuristic. Secara informal, pseudo code algoritma *Breadth First Search* diilustrasikan.

1. *Open* berisi initial state dan *Closed* masih kosong.
2. Ulangi sampai goal ditemukan atau sampai tidak ada nodes didalam *Open*.

- a. Jika simpul tersebut sama dengan goal maka sukses.
- b. Jika simpul tersebut sama dengan goal maka sukses.
- c. Jika tidak, masukkan simpul tersebut sampai *Closed*.
- d. Bangkitkan semua suksesor dari simpul tersebut.
- e. Untuk setiap suksesor kerjakan :
 - 1) Jika suksesor tersebut belum pernah dibangkitkan, evaluasi suksesor tersebut, tambahkan ke *Open*, dan catat parent atau orang tuanya.
 - 2) Jika suksesor tersebut sudah pernah dibangkitkan, ubah parent-nya jika jalur melalui parent ini lebih baik daripada jalur melalui parent yang sebelumnya. selanjutnya, perbarui biaya untuk suksesor tersebut dan nodes lain yang berada di *level* bawahnya.

Pada algoritma tersebut diatas, *Open* dan senarai (*list*) yang digunakan untuk menyimpan simpul-simpul yang pernah dibangkitkan dan nilai heuristic-nya telah dihitung tetapi belum terpilih sebagai simpul terbaik (*best node*). Dengan kata lain *Open* berisi simpul-simpul yang masih memiliki peluang (peluannya masih terbuka) untuk terpilih sebagai simpul terbaik. sedangkan *Closed* adalah senarai untuk menyimpan simpul-simpul yang sudah pernah dibangkitkan dan sudah pernah terpilih sebagai simpul terbaik. Artinya *Closed* berisi simpul-simpul yang tidak mungkin terpilih sebagai simpul terbaik (peluang untuk terpilih sudah tertutup). Perhatikan langkah-langkah pada gambar II.8 berikut ini. (Ayu Putri Sri Rezeki; 2015: 20)



Gambar II.8. Ilustrasi Pencarian Terbaik Pertama BFS

(Sumber : Ayu Putri Sri Rezeki; 2015 : 21)

1. Open=[A5]; closed=[]
2. Evaluate A5; Open=[A4,C4,D6]; closed=[A5]
3. Evaluate B4; Open=[C4,E5,F5,D6]; closed=[B4,A5]
4. Evaluate C4; Open=[H3,G4,E5,F5,D6]; closed=[C4,B4,A5]
5. Evaluate H3; Open=[O2,P3,G4,E5,F5,D6]; closed=[H3,C4,B4,A5]
6. Evaluate O2; Open=[P3,G4,E5,F5,D6]; closed=[O2,H3,C4,B4,A5]
7. Evaluate P3; Solusi ditemukan.

Ada beberapa keuntungan menggunakan algoritma ini, diantaranya adalah tidak akan menemui jalan buntu dan jika ada lebih dari satu solusi maka akan mudah menemukannya. namun ada tipe persoalan yang berkenaan dengan algoritma ini yaitu :

1. Membutuhkan memori yang besar, karena menyimpan semua node dalam satu pohon.
2. Membutuhkan sejumlah besar pekerjaan, khususnya jika lintasan solusi terpendek cukup panjang.
3. Tidak relevannya operator akan menambah jumlah node yang harus diperiksa.

Oleh karena itu, proses ini menjamin ditemukannya lintasan terpendek dari keadaan awal ke keadaan tujuan (akhir). Lebih jauh karena mula-mula semua keadaan ditemukan melalui lintasan terpendek sehingga setiap keadaan yang ditemui pada kali ke-*n* didapati pada sepanjang sebuah lintasan yang sama atau lebih panjang. Kemudian jika ada kesempatan ditemukannya keadaan yang identik pada sepanjang lintasan yang lebih baik maka algoritma akan menghapusnya.

II.8. Dragon Ball

Saya ingat zaman PS1 dimana kita suka mengadu Kamehameha dan combo “Meteor” di Dragon Ball GT Final Bout. *Game* yang bagus, untuk merusak stik karena kita harus barbar memencet tombol agar menang adu Kamehameha atau Meteor. Tapi combo pukulan dan tendangan biasa terasa sangat kaku dan kurang seru. Apalagi itu adalah *game fighting* 3D, bukan 2D seperti Blazblue atau yang sejenis. (Nugrahadi; 2015: 25).

Tampaknya akhirnya doa para gamer yang menanti *game fighting* 2D terjawab oleh *game* amatir ini. Hyper Dragon Ball Z memang memiliki daftar

karakter yang jauh lebih sedikit dibanding game resminya, tapi mekanisme game yang ditawarkan sukses membuat saya tergiur. Salah satu alasan utama kami membuat game ini meskipun banyak sekali *game* Dragon Ball di pasaran, resmi maupun tidak, selalu terasa kurang komentar sang developer, Team Z2. *Game* ini dibuat berbasis engine MUGEN yang kita semua kenal. Tidak hanya pertarungan cepat dan kombo keren khas 2D fighting, tapi juga ada beberapa kombo yang bisa diakhiri dengan adegan sinematik khas tiap karakter.