

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 ; 2012 : 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; 2012 : 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 Perancangan

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; 2012 : 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.

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. (Insap Santoso; 2012: 188-189).

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, yaitu sebagai berikut : (Insap Santoso; 2012: 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.

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. Kriptografi

Ditinjau dari asal usulnya, kata algoritma mempunyai sejarah yang menarik. Kata ini muncul di dalam kamus Webster sampai akhir tahun 1957. Kata algorisma mempunyai arti proses perhitungan dalam bahasa Arab. Algoritma berasal dari nama penulis buku Arab yang terkenal, yaitu Abu Ja'far Muhammad Ibnu Musa Al-Khuwarizmi (al-Khuwarizmi dibaca oleh orang barat sebagai algorism). Kata algorism lambat laun berubah menjadi algorithm.

Definisi terminologi algoritma adalah urutan langkah-langkah logis untuk menyelesaikan masalah yang disusun secara sistematis. Algoritma kriptografi merupakan langkah-langkah logis bagaimana menyembunyikan pesan dari orang-

orang yang tidak berhak atas pesan tersebut. Algoritma kriptografi terdiri dari tiga fungsi dasar, yaitu :

1. Enkripsi merupakan hal yang sangat penting dalam kriptografi, merupakan pengamanan data yang dikirimkan agar terjaga kerahasiannya. Pesan asli disebut plaintext, yang diubah menjadi kode-kode. Sama halnya dengan tidak mengerti akan sebuah kata maka akan melihatnya di dalam kamus atau daftar istilah. Bedan halnya dengan enkripsi, untuk mengubah teks asli ke bentuk teks kode menggunakan algoritma yang dapat mengkodekan data yang diinginkan.
2. Deskripsi merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya teks asli, disebut dengan enkripsi pesan. Algoritma yang digunakan untuk deskripsi tentu berbeda dengan algoritma yang digunakan untuk enkripsi.
3. Kunci yang dimaksud di sini adalah kunci yang dipakai untuk melakukan enkripsi dan deskripsi. Kunci terbagi menjadi dua bagian, kunci rahasia (*private key*) dan kunci umum (*public key*).
4. Keamanan dari algoritma kriptografi tergantung pada bagaimana algoritma itu bekerja. Oleh sebab itu algoritma semacam ini disebut dengan algoritma terbatas. Algoritma terbatas merupakan algoritma yang dipakai sekelompok orang untuk merahasiakan pesan yang mereka kirim. Jika salah satu dari anggota kelompok itu keluar dari kelompoknya amak algoritma yang dipakai diganti dengan yang baru. Jika tidak maka hal itu bisa menjadi masalah di kemudian hari.

II.3. Macam-macam algoritma Kriptografi

Algoritma kriptografi dibagi menjadi tiga bagian berdasarkan kunci yang dipakainya, sebagai berikut :

1. Algoritma Simetri.

Pada sistem kriptografi kunci simetris, kunci untuk enkripsi sama dengan kunci untuk dekripsi, oleh karena itulah dinamakan kriptografi simetris. Keamanan sistem kriptografi simetris terletak pada kerahasiaan kuncinya. Kriptografi simetris merupakan satu-satunya jenis kriptografi yang dikenal dalam catatan sejarah hingga tahun 1976.

2. Algoritma Asimetri.

Algoritma asimetris menggunakan dua jenis kunci, yaitu kunci publik dan kunci rahasia. Nama lainnya adalah kriptografi kunci publik, sebab kunci untuk enkripsi tidak rahasia dan dapat diketahui oleh siapapun, sementara kunci untuk dekripsi hanya diketahui oleh penerima pesan.

3. *Hash Function*.

Algoritma ini sering disebut dengan algoritma klasik karena memakai kunci yang sama untuk kegiatan enkripsi dan dekripsi. Algoritma ini sudah ada sejak lebih dari 4000 tahun yang lalu. Bila mengirim pesan dengan menggunakan algoritma ini, si penerima pesan harus diberitahu kunci dari pesan tersebut agar bisa mendekripsi pesan yang dikirim.

Dari ketiga kunci yang telah dijelaskan diatas, kunci tersebut bersifat rahasia baik itu kunci enkripsi maupun kunci dekripsi. Kunci ini dapat digunakan sebagai

private maupun publik, dan kunci ini dapat digunakan secara bersamaan antara kunci enkripsi dan kunci dekripsi.

II.4. Hill Cipher

Hill cipher yang merupakan polyalphabetic cipher dapat dikategorikan sebagai block cipher karena teks yang akan diproses akan dibagi menjadi blok-blok dengan ukuran tertentu. Setiap karakter dalam satu blok akan saling mempengaruhi karakter lainnya dalam proses enkripsi dan dekripsinya, sehingga karakter yang sama tidak dipetakan menjadi karakter yang sama.

Dasar dari teknik hill cipher adalah aritmatika modulo terhadap matriks. Dalam penerapannya, hill cipher menggunakan teknik perkalian matriks dan teknik invers terhadap matriks. Kunci pada hill cipher adalah matriks $n \times n$ dengan n merupakan ukuran blok. Jika matriks kunci disebut dengan K , maka matriks K adalah sebagai berikut (Arya Widyanarko;2012:2)

$$K = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1m} \\ k_{21} & k_{22} & \dots & k_{2m} \\ \dots & \dots & \dots & \dots \\ k_{m1} & k_{m2} & \dots & k_{mn} \end{bmatrix}$$

Matriks K yang menjadi kunci ini harus merupakan matriks yang invertible, yaitu memiliki multiplicative inverse K^{-1} sehingga $K \cdot K^{-1} = 1$. Kunci harus memiliki invers karena matriks K^{-1} tersebut adalah kunci yang digunakan untuk melakukan dekripsi

II.5. Teknik enkripsi

Proses enkripsi pada hill cipher dilakukan per blok plainteks. Ukuran blok tersebut sama dengan ukuran matriks kunci. Sebelum membagi teks menjadi deretan blok-blok, plainteks terlebih dahulu dikonversi menjadi angka, masing-masing sehingga A=0, B=1, hingga Z=25. Secara matematis, proses enkripsi pada hill cipher adalah $C = K \cdot P$. (Arya Widyanarko;2012:3)

C = Cipherteks

K = Kunci

P = Plainteks

Jika terdapat plainteks $P = B E S O K M A L A M$ Maka plainteks tersebut dikonversi menjadi $P = 1 4 18 14 10 12 0 11 0 12$. Plainteks tersebut akan dienkripsi dengan teknik hill ciphe, dengan kunci K yang merupakan matriks 2×2 . Karena matriks kunci K berukuran 2, maka plainteks dibagi menjadi blok yang masing-masing bloknya berukuran 2 karakter. Blok plainteks ini kemudian dienkripsi dengan kunci K melalui persamaan.

Karakter yang berkorespondensi dengan 9 dan 7 adalah J dan H. Maka karakter BE pada plainteks berubah menjadi karakter JH pada cipherteks. Pada contoh karakter ketiga dan keempat, hasil perhitungan menghasilkan angka yang tidak berkorespondensi dengan huruf-huruf, maka lakukan modulo 26 pada hasil tersebut. Maka karakter yang berkorespondensi dengan 20 dan 16 adalah U dan Q. Setelah melakukan enkripsi semua blok pada plainteks P maka dihasilkan cipherteks C sebagai berikut:

$P = BESOKMALAM$

C=JHUQIQWLYM

Dari cipherteks yang dihasilkan terlihat bahwa hill cipher menghasilkan cipherteks yang tidak memiliki pola yang mirip dengan plainteksnya

II.6. Teknik Deskripsi

Proses dekripsi pada hill cipher pada dasarnya sama dengan proses enkripsinya. Namun matriks kunci harus dibalik (invers) terlebih dahulu. Secara matematis, proses dekripsi pada hill cipher dapat diturunkan dari persamaan.

$$C = K.P$$

$$K^{-1}.C = K^{-1}.K.P$$

$$K^{-1}.C = I.P$$

$$P = K^{-1}.C$$

Menjadi persamaan proses dekripsi $P = K^{-1}.C$ dengan menggunakan kunci $K = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}$, proses dekripsi diawali dengan menghitung invers dari matriks K dengan menggunakan metode operasi baris (row operation). (Arya Widyanarko;2012:5)

$$\begin{aligned} [K | I] &= \left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 1 \end{array} \right] (9 \times R2) \\ &= \left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 27 & 9 & 0 & 9 \end{array} \right] (\text{mod } 26) \\ &= \left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 1 & 9 & 0 & 9 \end{array} \right] (R2 - R1) \\ &= \left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 0 & 7 & -1 & 9 \end{array} \right] (15 \times R2) \\ &= \left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 0 & 105 & -15 & 135 \end{array} \right] (\text{mod } 26) \\ &= \left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 0 & 1 & -15 & 5 \end{array} \right] (R1 - 2R2) \\ &= \left[\begin{array}{cc|cc} 1 & 0 & 31 & -10 \\ 0 & 1 & -15 & 5 \end{array} \right] (\text{mod } 26) \\ &= \left[\begin{array}{cc|cc} 1 & 0 & 5 & 16 \\ 0 & 1 & 11 & 5 \end{array} \right] \end{aligned}$$

Setelah melakukan perhitungan, didapat matriks K^{-1} yang merupakan invers dari matriks K , yaitu :

$$K^{-1} = \begin{bmatrix} 5 & 16 \\ 11 & 5 \end{bmatrix}$$

Kunci K^{-1} yang digunakan untuk melakukan dekripsi karena

$$\begin{aligned} K K^{-1} &= \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 5 & 16 \\ 11 & 5 \end{bmatrix} \\ &= \begin{bmatrix} 27 & 26 \\ 26 & 53 \end{bmatrix} \bmod 26 \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I \end{aligned}$$

Cipherteks $C = J H U Q I Q W L Y M$, akan didekripsi dengan menggunakan kunci dekripsi K^{-1} . Proses dekripsi ini dilakukan blok per blok seperti pada proses enkripsi. Pertama-tama ubah huruf-huruf pada cipherteks menjadi urutan numerik. $C = 9 7 20 16 8 16 22 11 24 12$.

II.7. Penerapan Pengolahan Citra Digital

Pengolahan citra digital dimulai sekitar awal tahun 1920-an dari dunia pemberitaan media cetak, di mana sebuah citra dikirim melalui kabel bawah laut dari London menuju ke New York. Proses transmisi ini menghemat waktu pengiriman dari seminggu menjadi kurang dari tiga jam. Sebelum dikirim, citra terlebih dulu dikodekan dan setelah diterima citra direkonstruksi ulang. Contoh ini sebenarnya masih kurang tepat digunakan sebagai awal mula pengolahan citra digital karena dalam prosesnya belum menggunakan teknologi komputer. (Darma Putra ; 2010 : 2).

Pengolahan citra digital sebenarnya mulai dari sekitar tahun 1960-an ketika saat itu diluncurkan computer yang mampu melakukan pengolahan citra.

Komputer tersebut adalah pemicu cepatnya perkembangan teknologi pengolahan citra digital. Pada tahun 1964 terjadi proses pengolahan citra berupa perbaikan kualitas citra bulan dari distorsi di laboratorium *jet propulsion*. Citra bulan kemudian dikirim oleh pesawat ulang alik ranger 7.

II.8. Pengenalan Digital Image

Digital image merupakan obyek nyata yang direpresentasikan secara elektronis. Obyek dapat bersumber dari dokumen, foto, barang cetakan, hingga lukisan. Unsur utama digital image adalah grid-grid berisi elemen obyek yang sangat dasar, yaitu picture element (pixel). Setiap pixel mempunyai tingkatan nilai tertentu, sehingga menghasilkan representasi data yang ditangkap oleh mata manusia sebagai bentuk tingkatan warna hitam, putih, abu-abu, hingga penuh dengan warna. (Edi S. Mulyanta ; 2010 : 5).

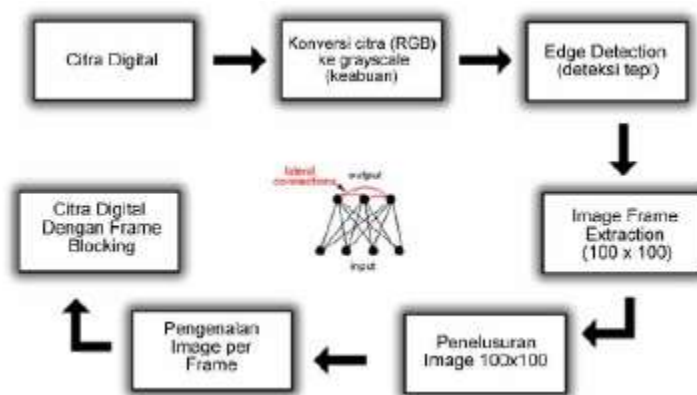
Setiap bit dalam pixel akan disimpan dalam urutan tertentu oleh computer dengan penghitungan matematis agar menghasilkan file yang optimal dibaca oleh media perangkat yang mendukungnya. Setiap informasi bit digital akan diinterpretasikan dan dibaca oleh computer agar menghasilkan versi analog untuk ditampilkan dan dicetak oleh media lain. Versi yang dilihat oleh mata manusia adalah data bersifat analog yang dirangkai oleh peralatan digital pada media komputer. (Edi S. Mulyanta ; 2010 : 5).

Berbagai peralatan elektronis umum telah beralih ke serba digital. Era digital tampaknya akan semakin mencengkeram kuat berbagai segi kehidupan manusia. Tampaknya, teknologi lama semakin tersisih dan dipandang telah tidak

ekonomis apabila masih digunakan. Pemakaian teknologi lama mungkin masih efisien digunakan, namun tuntutan selanjutnya adalah apakah masih efektif digunakan. Tidak semua orang setuju bahwa teknologi baru selalu memberikan dampak positif bagi manusia penggunaanya. Kadang harus mengedepankan sikap bijak dalam memaknai teknologi baru yang semakin cepat hadir. Pemaksaan pengguna teknologi baru semakin mendesak dan terkadang hadir tanpa disadari oleh penggunaanya. (Edi S. Mulyanta ; 2010 : 2).

1. Proses Pengolahan Citra

Proses pengolahan citra dan deteksi dilakukan dengan berbagai tahap proses yaitu mengambil citra grayscale dari citra asli, kemudian deteksi tepi hingga proses pengenalan,



Gambar II.1. Blok Proses Pencarian Pola
(Sumber : Nazrul Effendy, dkk ; 2010 : 5)

2. Citra Digital

Citra (*image*) adalah representasi optis dari sebuah obyek yang disinari oleh sebuah sumber radiasi. Pada dasarnya citra yang dilihat terdiri atas berkas-berkas cahaya yang dipantulkan oleh benda-benda disekitarnya, jadi secara

alamiah fungsi intensitas cahaya merupakan fungsi sumber cahaya yang menerangi obyek, serta jumlah cahaya yang dipantulkan oleh obyek, dinotasikan

$$(x, y) = i(x, y) \cdot r(x, y) \dots\dots\dots (2)$$

dimana : $0 < i(x, y) < \infty$ merupakan iluminasi sumber cahaya

$0 < r(x, y) < 1$ merupakan koefisien pantul obyek

Salah satu bentuk citra adalah citra yang mengandung abstrak dari citra matematis yang berisi fungsi kontinyu dan fungsi diskrit atau citra digital. Citra yang memiliki fungsi diskrit inilah yang dapat diolah oleh komputer. Setiap citra digital memiliki beberapa karakteristik, antara lain ukuran citra, resolusi dan format nilainya. Untuk itu citra digital harus mempunyai *format* tertentu yang sesuai sehingga dapat merepresentasikan obyek pencitraan dalam bentuk kombinasi data *biner*. (Muhtadan, dkk ; 2008 : 468).

Format citra digital yang banyak digunakan adalah citra biner, skala keabuan (*grayscale*), warna dan warna berindeks. Film radiografi merupakan citra fisik yang menunjukkan distribusi materi atau energy dari radiasi pengion dimana radiasi pengion menghitamkan film sehingga tingkat kehitaman merupakan wujud dari densitas benda uji, sedangkan bentuk dari struktur benda uji ditunjukkan dengan bentuk citra yang nampak dalam film. Pengolahan citra digital memfokuskan transformasi suatu citra pada format digital dan pengolahannya oleh komputer digital. Input dan output dari sistem pengolahan citra digital adalah citra digital. (Muhtadan, dkk ; 2008 : 468).

Citra adalah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi yang kontinu menjadi gambar diskrit melalui proses sampling. Gambar

analog dibagi menjadi N baris dan M kolom sehingga menjadi gambar diskrit.

Persilangan antara baris dan kolom tertentu disebut dengan piksel. Contohnya adalah gambar atau titik diskrit pada baris n dan kolom m disebut dengan piksel $[n,m]$. Sedangkan Sampling adalah proses untuk menentukan warna pada piksel tertentu pada citra dari sebuah gambar yang kontinu. Pada proses sampling biasanya dicari warna rata-rata dari gambar analog yang kemudian dibulatkan kedalam angka bulat. Proses sampling sering juga disebut proses digitisasi.

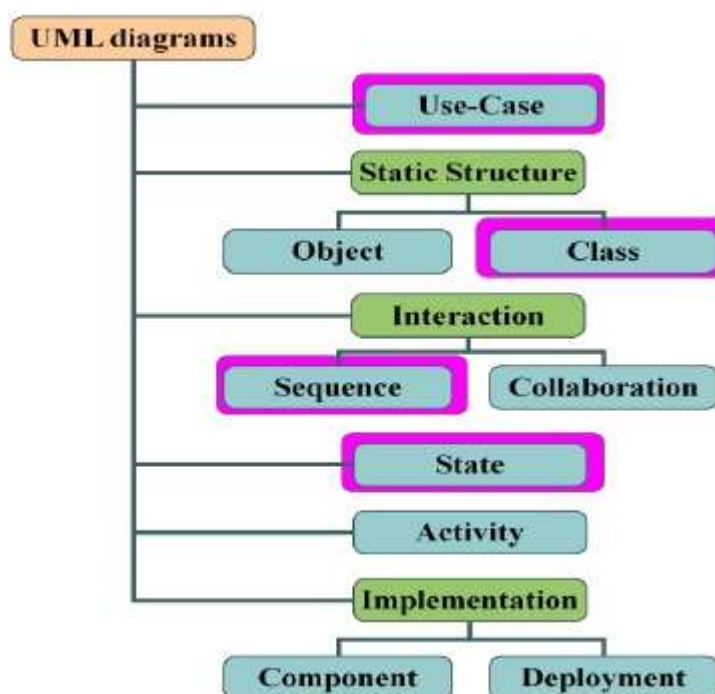
II.9. *Unified Modeling Language (UML)*

Unified Modelling Language (UML) Menurut (Haviluddin) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara *visual* (Braun, *et. al.* 2012). Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek.

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.

Saat ini 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.2 seperti dibawah in :



Gambar II.2. Diagram UML
(Sumber : Havaluddin; 2)

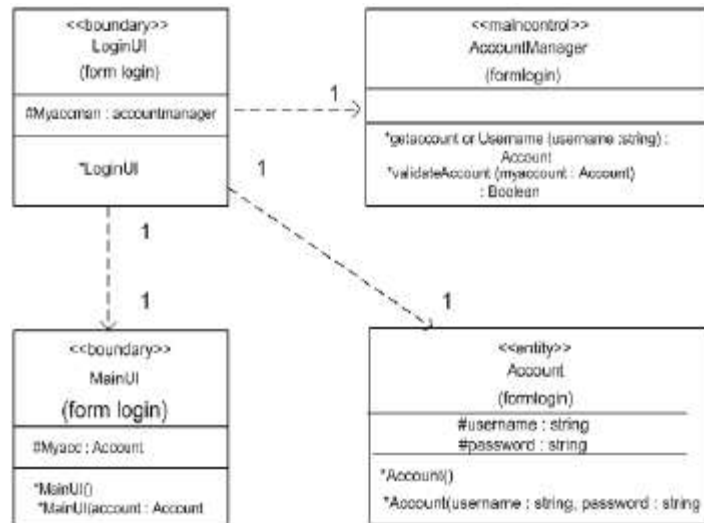
II.9.1. Tujuan Pemanfaatan UML

Tujuan dari penggunaan diagram seperti diungkapkan oleh Schmuller 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 (Havaluddin; 2015: 23)

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.
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 sistem.
7. Memiliki integrasi praktik terbaik.

II.9.2. Struktur Diagram

Menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem. *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.3. Notasi class diagram
(Sumber : Havaluddin; 3)

II.10. Activity diagram

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas.

Tabel II.1. Activity Diagram

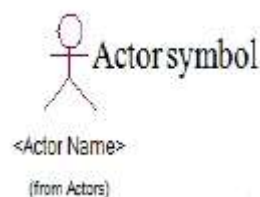
	<i>Activity</i>
	<i>Transition</i>
	<i>Decision</i>
	<i>Synchronization Bars</i>

(Sumber : Havaluddin; 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*

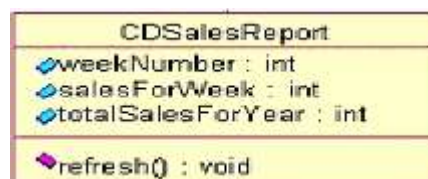
Actor menentukan peran yang dimainkan oleh *user* atau sistem lain yang berinteraksi dengan subjek. *Actor* adalah segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer, seperti orang, benda atau lainnya. Tugas *actor* adalah memberikan informasi kepada sistem dan dapat memerintahkan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



Gambar II.4. Notasi *actor*
(Sumber : Haviluddin; 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

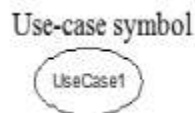


Gambar II.5. Notasi *class*
(Sumber : Haviluddin; 6)

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

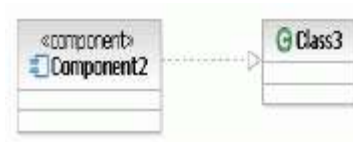
antara pengguna dan sistem disebut skenario. *Use case* merupakan awal yang sangat baik untuk setiap *fase* pengembangan berbasis objek, *design*, *testing*, dan dokumentasi yang menggambarkan kebutuhan sistem dari sudut pandang di luar sistem. Perlu diingat bahwa *use case* hanya menetapkan apa yang 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.6. Notasi *use case*
(Sumber : Haviluddin; 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.7. Notasi *realization*
(Sumber : Haviluddin; 6)

5. *Interaction*

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



Gambar II.8. Notasi *Interaction*
(Sumber : Haviluddin; 6)

II.11. Bahasa Pemrograman Visual Basic 2010

Visual Basic merupakan salah satu bahasa pemrograman yang andal dan banyak digunakan oleh pengembang untuk membangun berbagai macam aplikasi Windows. Visual Basic 2010 atau Visual Basic 9 adalah versi terbaru yang telah diluncurkan oleh Microsoft bersama C#, visual C++, dan Visual Web Developer dalam satu paket Visual Studio 2010 (Wahana Komputer;2012:20).

Visual Basic 2010 merupakan aplikasi pemrograman yang menggunakan teknologi *.NET Framework*. Teknologi *.NET Framework* merupakan komponen Windows yang terintegrasi serta mendukung pembuata, penggunaan aplikasi, dan halaman web. Teknologi *.NET Framework* mempunyai 2 komponen utama, yaitu CLR (*Common Language Runtime*) dan *Class Library*, CLR digunakan untuk menjalankan aplikasi yang berbasis .NET, sedangkan *Library* adalah kelas pustaka atau perintah yang digunakan untuk membangun aplikasi.

Sebelum menginstall komputer harus memenuhi beberapa persyaratan agar Visual Basic 2010 dapat dijalankan dengan baik. Adapun, persyaratan (*System Requirements*) yang harus dipenuhi dapat Anda lihat pada Tabel II.2.

Tabel II.2. Sistem Requirements Visual Basic 2010

Sistem	Syarat Minimal	Syarat yang direkomendasikan
Arsitektur	X86 dan x64 (WOW)	
Sistem Operasi	Microsoft Windows XP Service Pack 2 Microsoft Windows Server 2003 Windows Vista	
Prosesor	CPU 1.6 GHz (Giga Hertz)	Windows XP dan Windows Server 2003:CPU 2,2 GHz atau yang lebih tinggi. Windows Vista : CPU

		2,4 GHz
RAM	Windows XP dan Windows Server 2003 384 MB (Mega byte) Windows Vista : 768 MB	RAM 1024 MB / 1 GB atau yang lebih besar.
Harddisk	Tanpa MSDN Ruang Kosong harddisk pada drive penginstalan 2 GB. Sisa ruang harddisk kosong 1 GB Dengan MSDN Ruang kosong harddisk pada drive penginstalan 3,8 GB (MSDN diinstal full) 2,8 GB untuk menginstal MSDN default. Kecepatan Harddisk 5400 RPM.	Kecepatan harddisk 7200 RPM atau yang lebih tinggi.
Display Layar	1024 x 768 display	1280 x 1024 display

(Sumber : Wahana Komputer ; 2)

II.12. Keamanan Data

Keamanan data merupakan bagian dari perkembangan teknologi informasi. Ketika berpikir bahwa data yang dimiliki merupakan data yang sangat penting, semua berusaha untuk melindunginya agar jangan sampai jatuh ke tangan orang yang tidak bertanggung jawab. Tetapi buat sebagian orang, mereka justru tidak mengetahui seberapa penting apakah data yang mereka miliki. Karena ketidaktahuan tersebut, mereka baru menyadari bahwa data yang mereka miliki sangat penting setelah mengalami kecurian data dan mengalami kerugian. Data di sini bisa bersifat umum tidak terbatas pada data digital saja, tetapi juga seperti data diri (ktp, ijasah, sertifikat, dan lain-lain). Data yang menyangkut informasi pribadi tidak seharusnya diumbar sembarang seperti pada blog, situs jejaring pertemanan, email, selebaran, fotokopi KTP di buang sembarangan dan lain-lain. (Andik Susilo, 2012:16).

Masalah keamanan merupakan salah satu aspek terpenting dari sebuah sistem informasi. Masalah keamanan sering kurang mendapat perhatian dari para perancang dan pengelola sistem informasi. Masalah keamanan sering berada di urutan setelah tampilan, atau bahkan di urutan terakhir dalam daftar hal-hal yang dianggap penting. Apabila mengganggu performansi sistem, masalah keamanan sering tidak dipedulikan, bahkan ditiadakan.

Informasi menentukan hampir setiap elemen dari kehidupan manusia. Informasi sangat penting artinya bagi kehidupan karena tanpa informasi maka hampir semuanya tidak dapat dilakukan dengan baik. Contohnya, jika membeli tiket penerbangan dan membayarnya dengan menggunakan kartu kredit, informasi mengenai diri nantinya disimpan dan dikumpulkan serta digunakan oleh bank dan penerbangan. Demikian juga halnya saat membeli obat di apotik. Harus mendapat resep dari dokter dan memberikan resep tersebut ke pelayan apotik. Resep itu merupakan satu informasi yang disampaikan dokter ke pihak apotik tentang obat yang dibutuhkan.

Kemajuan sistem informasi memberikan banyak keuntungan bagi kehidupan manusia. Meski begitu, aspek negatifnya juga banyak, seperti kejahatan komputer yang mencakup pencurian, penipuan, pemerasan, kompetisi, dan banyak lainnya. Jatuhnya informasi ke pihak lain, misalnya lawan bisnis, dapat menimbulkan kerugian bagi pemilik informasi. Sebagai contoh, banyak informasi milik perusahaan yang hanya boleh diketahui oleh orang-orang tertentu di perusahaan tersebut, seperti misalnya informasi tentang produk yang sedang dalam pengembangan. Algoritma dan teknik yang digunakan untuk menghasilkan

2produk tersebut. Untuk itu keamanan dari sistem informasi yang digunakan harus terjamin dalam batas tertentu.