

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Perancangan**

Perancangan adalah analisis sistem, persiapan untuk merancang dan implementasi agar dapat menyelesaikan apa yang harus diselesaikan serta menkonfigurasi komponen-komponen perangkat lunak keperangkat keras. Perancangan adalah strategi untuk memecahkan masalah dan mengembangkan solusi terbaik bagi permasalahan itu. [12]

#### **II.2. Aplikasi**

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju. Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer langsung untuk melakukan tugas yang diinginkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media. [3]

#### **II.3. Citra**

Warna adalah respon persepsi dari mata dan otak manusia terhadap pancaran energi berbagai panjang gelombang dan intensitasnya. Pancaran energi tersebut diserap oleh mata dan dipersepsikan oleh otak sebagai warna. Ilmu tentang warna

pada dasarnya merupakan karakteristik sensor dari mata. Citra adalah kumpulan dari beberapa warna yang diatur sedemikian rupa ang bertujuan untuk menyampaikan informasi. Oleh kaena itu warna merupakan komponen yang penting dari suatu citra. [10]

Istilah citra, digunakan untuk menyatakan intensitas cahaya dua dimensi dalam fungsi  $f(x,y)$ , dimana  $(x,y)$  menyatakan koordinat spasial dan nilai dari  $f$  pada titik  $(x,y)$  menyatakan tingkat kecerahan citra pada titik tersebut. Fungsi  $f(x,y)$ , dipengaruhi oleh banyaknya sumber cahaya yang jatuh pada daerah yang diamati dan banyaknya sumber cahaya yang dipantulkan oleh objek pada daerah tersebut (refleksi). Hal ini daapat ditulis secara matematis sebagai :

$$f(x,y) = i(x,y) \cdot r(x,y) \dots \dots \dots (1)$$

dimana:

$$0 < i(x,y) < 8 \quad 0 < r(x,y) < 1$$

Jika  $r(x,y) = 0$ , maka semua cahaya diserap, sedangkan jika  $r(x,y) = 1$ , maka semua cahaya dipantulkan. Bila nilai  $r(x,y)$  berada diantara kedua nilai tersebut, maka akan dihasilkan warna yang berbeda. [10]

Citra  $f(x,y)$  yang kontinu, dapat dinyatakan sebagai nilai-nilai sampel yang dipisahkan pada jarak sama dan disusun dalam bentuk matriks  $N \times M$  dimana tiap elemen dari matriks menunjukkan entitas diskrit. Level keabuan dalam bentuk diskrit terpisah dalam range 0 sampai 225. Suatu citra digital dapat dipandang sebagai array dua dimensi seperti berikut:

Citra digital dapat dipandang sebagai sebuah matrik yang indeks baris dan kolomnya menyatakan titik pada citra dan elemen matriknya menyatakan level keabuan pada titik tersebut.

Secara prinsip ada 2 jenis metode untuk merepresentasikan citra digital, yaitu:

1. BitMap/Raster

Sebuah citra dibagi menjadi kotak-kotak kecil dimana setiap kotak berisi informasi tentang nilai intensitas yang menunjuk kepada index tabel warna (*palettes*), kotak-kotak ini disebut sebagai pixel. Posisi dari kotak (*bit*) yang dipresentasikan merupakan pemetaan sebagian dari citra pada posisi tersebut, oleh karena itu disebut bitmap.

2. *Vektor*

Sebuah citra didekripsikan sebagai sekumpulan garis atau bentuk. [10]

## **II.3.1. Elemen-Elemen Citra Digital**

### **II.3.1.1. Color**

Warna adalah elemen dasar gambar digital yang paling penting. Setiap gambar yang didigitalisasi kita representasikan dalam bentuk matriks warna (matrix of color). Warna adalah persepsi yang dirasakan oleh system visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh sebuah objek. Tiap warna memiliki panjang gelombang yang berbeda. Warna merah memiliki panjang gelombang paling tinggi, sedangkan warna ungu memiliki panjang gelombang paling rendah. Warna-warna yang diterima oleh mata manusia merupakan gabungan dari cahaya dengan panjang gelombang yang berbeda-beda.

Kombinasi warna yang memberikan rentang warna paling besar adalah kombinasi warna merah (R), hijau (G), dan biru (B). Nilai R, G, dan B pada gambar RGB 8 bit adalah 0 sampai 255.

### **II.3.1.2. *Brightness***

*Brightness* adalah nama lain dari tingkat kecerahan/intensitas cahaya. Elemen ini menyatakan banyaknya cahaya yang diterima oleh mata. Elemen ini dapat dirasakan sebagai lampu penerang berwarna putih ketika kita melihat suatu benda. Semakin terang cahaya lampu tersebut (Tingkat kecerahan/*brightness* tinggi), benda yang kita lihat akan semakin putih. Semakin redup (Tingkat kecerahan/*brightness* rendah), benda yang kita lihat semakin gelap. Dan ketika tidak ada cahaya lampu (Tingkat kecerahan/*brightness* = 0), benda yang kita lihat berwarna hitam.

### **II.3.1.3. *Saturation***

*Saturation* adalah tingkat kepekatan/konsentrasi suatu warna. Semakin tinggi nilai saturasi, semakin pekat warna tersebut. Warna merah adalah warna dengan nilai saturasi tinggi. Warna merah muda adalah warna dengan nilai saturasi rendah. Warna putih adalah warna dengan tingkat saturasi 0.

## **II.4. Kriptografi**

Kriptografi berasal dari bahasa Yunani, “*kryptos*” yang berarti tersembunyi dan “*graphein*” yang berarti tulisan. Sehingga kata kriptografi dapat diartikan menjadi “tulisan tersembunyi”. Menurut Request for Comments (RFC), kriptografi adalah ilmu matematika yang berhubungan dengan transformasi data

agar arti dari data tersebut menjadi aulit untuk dipahami (untuk menyembunyikan maknanya), mencegahnya dari perubahan tanpa izin, atau mencegahnya dari penggunaan yang tidak sah. Jika transformasinya dapat dikembalikan, kriptografi juga dapat diartikan sebagai proses mengubah kembali data yang terenkripsi menjadi bentuk yang mudah dipahami. Sehingga, kriptografi juga dapat diartikan sebagai proses untuk melindungi data dalam arti yang luas. [1]

Pengertian kriptografi dalam kamus bahasa inggris Oxford adalah sebagai berikut:

“ Sebuah teknik rahasia dalam penulisan, dengan karakter khusus, dengan menggunakan huruf dan karakter diluar bentuk aslinya, atau dengan metode-metode lainya yang hanya dapat dipahami oleh pihak-pihak yang memproses kuncinya, juga semua hal yang ditulis dengan cara seperti ini.”

Jadi, secara umum kriptografi diartikan sebagai seni menulis atau memecahkan cipher”. [1]

#### **II.4.1 Jenis Algoritma Kriptografi**

Berdasarkan jenis kunci, algoritma kriptografi dikelompokkan menjadi dua bagian, yaitu: algoritma simetris (algoritma kunci privat) dan algoritma asimetris (algoritma kunci publik). [1]

Algoritma simetris adalah salah satu jenis kunci pada algoritma kriptografi yang menggunakan kunci enkripsi yang sama dengan kunci dekripsinya. Istilah lain untuk kriptografi kunci simetri adalah kriptografi kunci-simetri (*private key cryptography*). Sistem kriptografi kunci simetri diasumsikan sebagai pengirim dan

penerima pesan yang sudah berbagi kunci yang sama sebelum bertukar pesan. Keamanan sistem kriptografi simetris terletak pada kerahasiaan kuncinya.

Proses enkripsi dan dekripsi pada kriptografi simetri menggunakan kunci yang sama. Sehingga timbul persoalan untuk menjaga kerahasiaan kunci yang sama. Sehingga timbul persoalan untuk menjaga kerahasiaan kunci. Contohnya pada saat pengiriman kunci dilakukan melalui media yang tidak aman seperti internet. Jika kunci ini hilang atau sudah diketahui orang lain yang tidak berhak, maka kriptosistem ini dinyatakan tidak aman lagi. Kelemahan lain adalah masalah efisiensi jumlah kunci . jika terdapat  $n$  user, maka diperlukan  $n(n-1)/2$  kunci, sehingga untuk jumlah user yang sangat banyak, sistem ini tidak efisien lagi. [1]

Algoritma asimetris atau dapat disebut juga dengan algoritma kunci public , didesain sebaik mungkin sehingga kunci yang digunakan untuk enkripsi berbeda dengan kunci dekripsinya. Dimana kunci enkripsi tidak rahasia (diumumkan ke publik), sementara kunci dekripsinya bersifat rahasia (hanya diketahui oleh penerima pesan). [1]

## II.5. Vigenere Chiper

Sandi *vigenere* merupakan sistem sandi poli-alfabetik yang sederhana. Sistem sandi poli-alfabetik mengenkripsi sekaligus sebuah teks yang terdiri dari beberapa huruf. Sandi vigenere menggunakan substitusi dengan fungsi *shift* Seperti pada sandi *caesar*. [7]

*Vigenere Chiper* termasuk kedalam *chiper* abjad majemuk (*poly alphabetic substitution cipher*). Ditemukan oleh diplomat (sekaligus kriptologis)

Perancis, Blaise de Vigenere pada abad 16. sudah berhasil dipecahkan pada abad 19. *Vigenere Cipher* menggunakan Bujursangkar *Vigenere* untuk melakukan enkripsi. [4]

## II.6. RGB

Model warna RGB adalah model warna yang aditif yang dapat dikombinasikan untuk membuat warna yang lainnya. Sistem warna RGB terdiri dari tiga bagian utama. Yaitu *Red* (Merah), *Green* (Hijau), dan *Blue* (Biru). [9]

Suatu citra biasanya mengacu ke citra RGB. Sebenarnya bagaimana citra disimpan dan dimanipulasi dalam komputer diturunkan dari teknologi televisi, yang pertama kali mengaplikasikannya untuk tampilan grafis dikomputer. jika dilihat dengan kaca pembesar, tampilan monitor komputer akan terdiri dari sejumlah *triplet* titik warna merah (*red*), hijau (*green*), biru (*blue*). Tergantung pada pabrik monitornya untuk menentukan apakah titik tersebut merupakan titik bulat atau kotak kecil, tetapi akan selalu terdiri dari 3 *triplet red, green dan blue*. Citra dalam komputer tidak lebih dari sekumpulan sejumlah *triplet* dimana setiap *triplet* terdiri dari variasi tingkat keterangan (*brightness*) dari elemen *red, green, blue*. Setiap triplet akan merepresentasikan 1 pixel (*picture element*).

## II.7. UML (*Unified Modelling Language*)

Penjadwalan telah lama diteliti, contohnya dalam penghasilan tenaga oleh *Windu Gata* dan *Grace Gata* (2013), UML (*Unified Modelling Language*) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan,

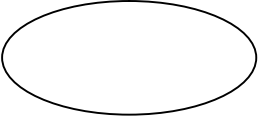
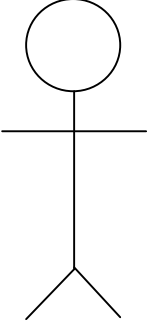
menspesifikasikan dan membangun perangkat lunak. UML (*Unified Modelling Language*) merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.



### II.7.1 UseCase Diagram

*UseCase Diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Simbol-simbol yang digunakan dalam *UseCase Diagram*, yaitu:

**Tabel II.1 UseCase Diagram**

Gambar	Keterangan
	<p><i>UseCase</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal nama <i>UseCase</i>.</p>
	<p><i>Actor</i> atau Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>usecase</i>, tetapi tidak memiliki kontrol terhadap <i>usecase</i>.</p>
	<p>Asosiasi antara aktor dan <i>usecase</i>, digambarkan</p>

	<p>dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>usecase</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>

(Sumber : Windu Gata dan Grace Gata ; 2013)


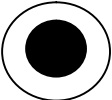
### II.7.2 Class Diagram (Diagram Kelas)


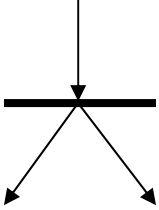
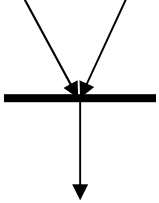
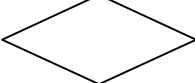
*Class Diagram* merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

### II.7.3 Activity Diagram (Diagram Aktivitas)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu:

**Tabel II.2 Activity Diagram**

Gambar	Keterangan
	<p><i>Start Point</i> diletakkan pada pojok kiri atas dan merupakan awal aktifitas.</p>
	<p><i>End Point</i>, akhir aktifitas.</p>

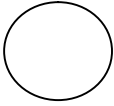
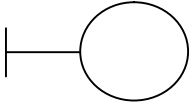
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>Rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .

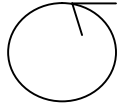

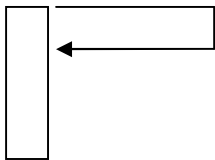


(Sumber : Windu Gata dan Grace Gata ; 2013)

#### II.8.4 Sequence Diagram (Diagram Urutan)

*Sequence Diagram* menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu:

**Tabel II.3 Sequence Diagram**

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan

	<i>formentry</i> dan <i>form</i> cetak.
	<i>Control Class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek. <i>Control object</i> mengkoordinir pesan antara <i>boundary</i> dengan entitas.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gata dan Grace Gata ; 2013)

## II.8. Visual Basic 2010

*Visual Basic* 2010 adalah inkarnasi dari bahasa *Cisual Basic* yang sangat populer dan telah dilengkapi dengan fitur serta fungsi yang setara dengan bahasa tingkat tinggi lainnya seperti C++. [2]

Anda dapat menggunakan *Visual Basic* 2010 untuk membuat aplikasi *Windows*, *mobile*, *Web*, dan *Office* yang kompleks dengan menggunakan kode

yang anda tulis, atau kode yang telah ditulis oleh orang lain dan kemudian dimasukkan ke dalam program anda.

*Visual Basic* menyediakan beberapa tools dan fitur canggih yang memungkinkan anda untuk menulis kode, menguji dan menjalankan program tunggal atau terkadang serangkaian program yang terkait dengan satu aplikasi.

*Microsoft Visual Basic 2010* merupakan salah satu bagian dari produk pemrograman yang dikeluarkan oleh *Microsoft*, yaitu *microsoft visual studio 2010*. *Microsoft visual studio 2010* merupakan produk lingkungan terintegrasi atau IDE yang dikeluarkan oleh *microsoft*. [9]

*Visual Basic 2010* merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu *visual basic 2008*. Bahasa *Visual Basic 2010* awalnya berasal dari bahasa pemrograman yang sangat populer dikalangan *programmer*, yaitu bahasa *Basic*. [9]