BAB II

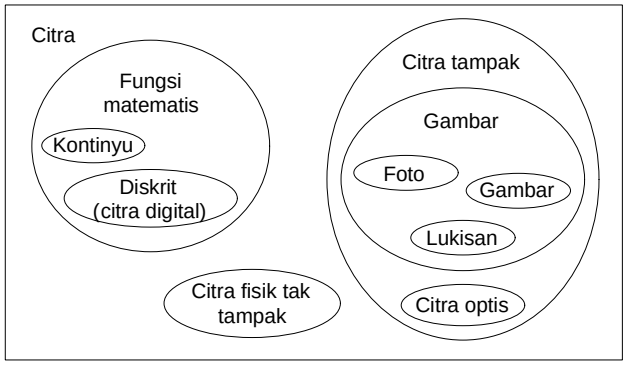
LANDASAN TEORI

2.1 Citra

Definisi citra menurut Kamus Webster adalah “suatu representasi, kemiripan, atau imitasi dari suatu objek atau benda”.

Secara harafiah, citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Sedangkan ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera pemindai (*scanner*), dan sebaginya, sehingga bayangan objek yang disebut citra tersebut terekam.

Citra dapat dikelompokkan menjadi citra tampak dan citra tak tampak:



Gambar 2.1 Pengelompokan jenis-jenis citra

Diantara jenis-jenis citra tersebut, hanya citra digital yang dapat diolah menggunakan komputer. Jenis citra lain, jika hendak diolah dengan komputer, harus diubah dulu menjadi citra digital.

2.2 Pengolahan Citra

Pengolahan citra adalah kegiatan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia/mesin(komputer). Inputannya adalah citra dan keluarannya juga citra tapi dengan kualitas lebih baik daripada citra masukan. Misal citra warnanya kurang tajam, kabur (blurring), mengandung noise (misal bintikbintik putih), dll sehingga perlu ada pemrosesan untuk memperbaiki citra karena citra tersebut menjadi sulit diinterpretasikan karena informasi yang disampaikan menjadi berkurang.

Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukkannya adalah suatu citra dan keluarannya juga citra, namun citra keluaran mempunyai kualitas lebih baik daripada citra masukkan.

Citra sebagai keluaran dari suatu sistem perekaman data dapat bersifat :

1. optik berupa foto
2. analog berupa sinyal video seperti gambar pada monitor televisi
3. digital yang dapat langsung disimpan pada media penyimpan magnetic.

Operasi-operasi pada pengolahan citra diterapkan pada citra bila :

1. Perbaikan atau memodifikasi citra dilakukan untuk meningkatkan kualitas

penampakan citra/menonjolkan beberapa aspek informasi yang terkandung dalam citra *(image enhancement)* contoh : perbaikan kontras gelap/terang, perbaikan tepian objek, penajaman, pemberian warna semu, dll

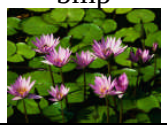
2. Adanya cacat pada citra sehingga perlu dihilangkan/diminimumkan *(image restoration)* contoh : penghilangan kesamaran (debluring) yaitu citra tampak kabur karena pengaturan fokus lensa tidak tepat / kamera goyang, penghilangan noise.
3. Elemen dalam citra perlu dikelompokkan, dicocokan atau diukur *(image segmentation)* Operasi ini berkaitan erat dengan pengenalan pola.
4. Diperlukannya ekstraksi ciri-ciri tertentu yang dimiliki citra untuk membantu dalam pengidentifikasian objek *(image analysis).* Proses segementasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Contoh : pendeteksian tepi objek.
5. Sebagian citra perlu digabung dengan bagian citra yang lain *(image reconstruction)* contoh : beberapa foto rontgen digunakan untuk membentuk ulang gambar organ tubuh.
6. Citra perlu dimampatkan *(image compression)*

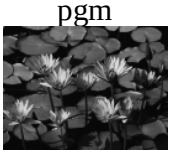
   contoh : suatu file citra berbentuk BMP berukuran 258 KB dimampatkan dengan metode JPEG menjadi berukuran 49 KB.

7. Menyembunyikan data rahasia (berupa teks/citra) pada citra sehingga keberadaan data rahasia tersebut tidak diketahui orang *(steganografi & watermarking).*

Berikut ini adalah contoh format citra

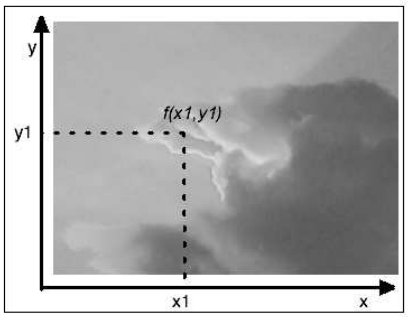
Tabel 2.1. Macam-macam Format Citra

Ekstensi	Nama	Keterangan
bmp 	Windows Bitmap	Biasanya digunakan oleh aplikasi dan sistem opeasi Microsoft Windows.Merupakan kompresi tipe lossless.
gif 	Graphics Interchange Gif	biasanya digunakan di website. Format Format gif mendukung citra bergerak. Namun format gif hanya mendukung 255 warna tiap frame. Format gif juga Mendukung citra transparan. Format gif merupakan kompresi tipe lossy.
jpg/jpeg 	Joint Photographic Ex- perts Group atau	JPEG biasanya digunakan untuk foto citra di website. JPEG menggunakan kompresi tipe lossy. Kualitas JPEG 2000 bisa bervariasi tergantung setting kompresi yang digunakan. Kompresi JPEG berbasis DCT(Discreete Cosine Transform)
jp2/jpg2/j2k  .	Joint Photographic Ex perts Group 2000	Merupakan pengembangan dari JPEG yang berbasis transformasi wavelet. Format ini mendukung kompresi tipe lossless dan lossy. Namun, support JPEG 2000 dalam berbagai aplikasi masih kurang, disebabkan kebutuhan hardware yang tangguh dan paten
pbm	Portable Bitmap Format	Merupakan format citra hitam putih yang sederhana. PBM memerlukan 1 bit tiap pixel. Tidak seperti format citra lainnya, format PBM merupakan plain

		text yang bisa diolah dengan menggunakan pengolah text. Format PBM merupakan bagian dari PNM (Portable Pixmap File Format).
pgm 	Portable Graymap Format	Merupakan format citra abu-abu yang sederhana. Format PGM memerlukan 8 bit tiap pixel. PGM merupakan citra mentah dengan kompresi tipe lossless. Format PGM merupakan bagian dari PNM (Portable Pixmap File Format).
ppm 	Portable Pixmap Format	Merupakan format citra berwarna yang sederhana. PPM memerlukan 24 bit tiap pixel. PPM merupakan citra mentah dengan kompresi tipe lossless. Format PPM merupakan bagian dari PNM (Portable Pixmap File Format).

2.3 Citra Digital

Citra digital dapat didefinisikan sebagai fungsi dua variabel, f(x,y), dimana x dan y adalah koordinat spasial dan nilai f(x,y) adalah intensitas citra pada koordinat tersebut, hal tersebut diilustrasikan pada gambar 2.2.



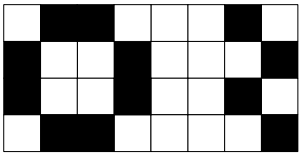
Gambar 2.2. Citra Digital

2.3.1 Jenis-jenis Citra Digital

Ada banyak cara untuk menyimpan citra digital di dalam memori. Cara penyimpanan menentukan jenis citra digital yang terbentu. Beberapa jenis citra digital yang sering digunakan adalah citra biner, citra *grayclale*, dan citra warna.

2.3.1.1 Citra Biner (Monokrom)

Citra biner adalah citra yang hanya mempunyai dua nilai derajat keabuan : hitam dan putih. Dibutuhkan 1 bit di memori untuk menyimpan kedua warna.
Gradasi warnanya, bit 0 untuk warna hitam, bit 1 untuk warna putih.



Gambar 2.3 Citra biner

2.3.1.2 Citra Grayscale (Citra Keabuan)

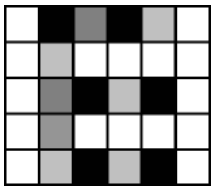
Citra skala keabuan memberi kemungkinan warna yang lebih banyak dari pada citra biner. Banyaknya warna tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna. Semakin besar jumlah bit warna yang disediakan di memori, semakin halus gradasi warna yang terbentuk.

Contoh :

skala keabuan 4 bit
jumlah kemungkinan 2_4 = 16 warna
kemungkinan warna 0 (min) sampai 15 (max)

skala keabuan 8 bit

jumlah kemungkinan 2_8 = 256 warna
kemungkinan warna 0 (min) sampai 255 (max)



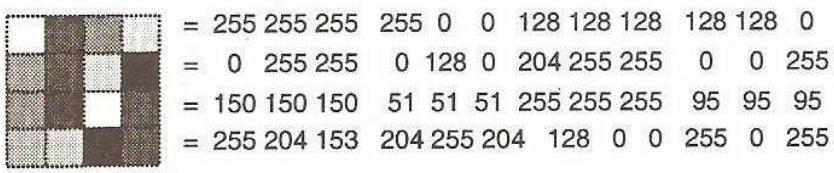
Skala keabuan 4 bit (hitam = 0, putih = 15)
= 15 0 6 0 13 15 = 15 12 15 15 15 15 = 15
5 0 12 0 15 = 15 8 15 15 15 15 = 15 10
0 13 0 15

Gambar 2.4 Citra skala keabuan

2.3.1.3 Citra Warna (*True Color*)

Pada citra warna setiap titik mempunyai warna yang spesifik, yang merupakan kombinasi dari 3 warna dasar, yaitu merah, hijau, dan biru. Format citra ini sering disebut RGB (*red-green-blue*). Setiap warna dasar mempunyai intensitas sendiri dengan nilai maksimum 255 (8 bit). Setiap titik pada citra warna membutuhkan data 3 *byte*.

Jumlah kombinasi warna yang memungkinkan untuk format citra ini adalah 2^{24} atau lebih dari 16 juta warna, dengan demikian bisa dianggap mencakup semua warna yang ada. Inilah sebabnya format ini dinamakan *true color*.



Gambar 2.5 Citra warna (*true color*)

2.4 Histogram Citra

Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas *pixel* dari suatu citra atau bagian tertentu dari citra. Dengan sebuah histogram dapat diketahui frekuensi kemunculan relatif dari intensitas pada citra, selain itu, informasi mengenai tingkat kecerahan (*brightness*) dan kontras (*contrast*) dapat diketahui melalui histogram. Jika suatu citra memiliki *L* derajat keabuan, yaitu dari nilai 0 sampai *L*-1, maka secara matematis dapat dirumuskan sebagai berikut:

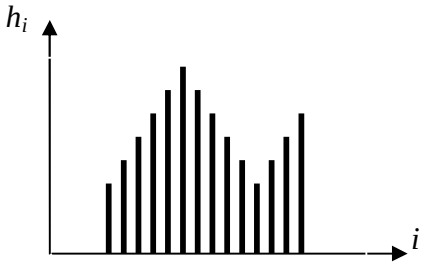
$$h_i = \frac{n_i}{n} \quad , \; i = 0,1,...,L-1 \tag{2.1}$$

dimana,

n_i = jumlah *pixel* yang memiliki derajat keabuan i

n = jumlah seluruh *pixel* didalam citra

Secara grafis histogram ditampilkan dengan diagram batang. Untuk citra berwarna histogram dibuat untuk setiap kanal RGB.



Gambar 2.6 Histogram Citra

Puncak histogram menunjukkan intensitas *pixel* yang menonjol. Lebar dari puncak menunjukkan rentang kontras dari gambar. Citra yang mempunyai kontras yang terlalu terang (*overexposed*) atau terlalu gelap (*underexposed*) memiliki histogram yang sempit. Histogramnya terlihat hanya menggunakan setengah dari daerah derajat keabuan. Citra yang baik akan memiliki histogram yang mengisi daerah derajat keabuan secara penuh dengan distribusi yang merata pada setiap nilai intensitas *pixel*.

2.5 Grafika Komputer

Grafika komputer atau dalam bahasa inggris *computer graphics* dapat diartikan sebagai sebagai perangkat alat yang terdiri dari hardware dan software untuk membuat gambar, grafik atau citra realistic untuk seni, game, komputer, foto dan film animasi.

Grafika komputer merupakan bagian yang paling menarik dari bidang ilmu komputer, ini dikarenakan untuk dapat memahaminya dengan baik diperlukan kemampuan metematika dan juga keahlian untuk memprogram dan kreatifitas.

2.5.1 Elemen gambar untuk menciptakan gambar dalam komputer

Penghasilan citra pada grafik komputer mengunakan primitif grafik dasar. primitif ini memudahkan untuk merender (menggambar pada layar monitor) sebagai

penggunaan persamaan geometri sederhana. Grafik dasar yang digunakan biasanya adalah titik dan garis.

2.5.1.1 Titik (Point)

Titik pada komputer merupakan sebuah koordinat yang mengandung identitas warna yang diletakkan pada Cathoda-Ray-Tube (CRT) monitor. Untuk meletakakan titik pada koordinat x dan y (x dan y merupakan besaran yang posisi pada arah horizontal dan vertical). Pada CRT monitor diperlukan program aplikasi untuk mengkonversi koordinat titik tersebut menjadi suatu nilai yang menyatakan lokasi dari peta layar monitor.

Pada peta layar monitor sebenarnya adalah linier, artinya layar monitor tidak mengenal sistem koordinat. Sebagai contoh, jika lokasi awal layar monitor (0,0) dan lokasi akhir monitor (X_{max}, Y_{max}) maka untuk mengkonversi koordinat suatu titik (x,y) menjadi suatu lokasi pada layar monitor digunakan rumus :

$$alamat\,(x,y) = y(x_{max}) + 1 \qquad (2.2)$$

Dimana alamat (x,y) merupakan lokasi sebenarnya pada layar monitor, x dan y adalah koordinat titik, x_{max} adalah maksimum pada arah horizontal, dan y_{max} adalah koordinat y maksimum pada arah vertical.

2.5.1.2 Garis (Line)

Penggambaran garis dilakukan dengan menghitung posisi-posisi sepanjang jalur lurus antara dua posisi titik. Jalur lurus yang menghubungkan dua titik tersebut

sebenarnya adalah titik-titik yang saling berdempetan mengikuti arah jalur lurus tersebut.

Jika suatu titik (x,y) berada pada garis, dan garis tersebut dinyatakan dengan koordinat (x_1,y_1) dan (x_2,y_2), maka persamaan untuk garis lurus adalah :

$$y = m.\, x + b \tag{2.3}$$

2.6 Matriks

Matriks adalah himpunan objek yang disusun secara persegi panjang (yang terdiri dari baris dan kolom) yang biasanya dibatasi dengan tanda kurung siku atau biasa.

2.6.1 Jenis-jenis Matriks

Matriks dapat dibedakan menurut jenisnya, antara lain:

1. Matriks Nol

   Suatu matriks dikatakan sebagai matriks nol, jika semua elemennya sama dengan nol. Misalnya,

   $$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

2. Matriks Baris

   Suatu matriks dikatakan sebagai matriks baris, jika matriks tersebut hanya terdiri atas satu baris, misalnya

   $$\begin{bmatrix} 1 & 7 \end{bmatrix}, \begin{bmatrix} 5 & -3 & 2 & 6 \end{bmatrix}$$

3. Matriks kolom

Suatu matriks dikatakan sebagai matriks kolom, jika matriks tersebut hanya terdiri dari satu kolom. Misalnya,

$$\begin{bmatrix} 2 \\ -5 \end{bmatrix}, \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}$$

4. Matriks persegi dan matriks kuadrat

   Suatu matriks dikatakan sebagai matriks persegi atau matriks kuadrat, jika jumlah baris pada matriks tersebut sama dengan jumlah kolomnya.

   Misalnya,

$$\begin{bmatrix} 2 & -3 \\ 4 & 1 \end{bmatrix}, \begin{bmatrix} 3 & 7 & -5 \\ 6 & -3 & 1 \\ -1 & 8 & -2 \end{bmatrix}$$

   Pada suatu matriks persegi ada yang dinamakan sebagai diagonal utama dan diagonal sekunder. Perhatikan matriks berikut.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

   Komponen-komponen yang terletak pada diagonal utama pada matriks tersebut adalah $a11$, $a22$ dan $a33$ (sesuai dengan arsiran yang berasal dari kiri atas ke kanan bawah). Sebaliknya, komponenkomponen yang terletak pada diagonal sekunder sesuai dengan arsiran yang berasal dari kiri bawah ke kanan atas, dalam hal ini $a11$, $a22$, $a33$.

5. Matriks segitiga

   Suatu matriks persegi dikatakan sebagai matriks segitiga jika elemenelemen yang ada di bawah atau di atas diagonal utamanya (salah satu, tidak kedua-duanya) bernilai nol. Jika elemen-elemen yangada di bawah diagonal utama bernilai nol maka disebut sebagai matriks segitiga atas. Sebaliknya, jika elemen-elemen yang ada di atas diagonal utamanya bernilai nol maka disebut sebagai matriks segitiga bawah.

   Misalnya,

   $$\begin{bmatrix} -5 & -1 & 2 \\ 0 & 4 & 3 \\ 0 & 0 & 4 \end{bmatrix} \qquad \begin{bmatrix} 7 & 0 & 0 \\ 5 & 1 & 0 \\ -4 & -2 & 3 \end{bmatrix}$$

   Matriks segitiga bawah Matriks segitiga atas

6. Matriks Diagonal

   Suatu matriks persegi dikatakan sebagai matriks diagonal jika elemenelemen yang ada di bawah dan di atas diagonal utamanya bernilai nol, atau dengan kata lain elemen-elemen selain diagonal utamanya bernilai nol.

   Misalnya,

   $$\begin{bmatrix} -1 & 0 \\ 0 & 4 \end{bmatrix} \quad \begin{bmatrix} -4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

7. Matriks Skalar

Suatu matriks diagonal dikatakan sebagai matriks skalar jika semua elemen-elemen yang terletak pada diagonal utamanya memiliki nilai yang sama, misalnya,

$$\begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix} \quad \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

8. Matriks Identitas dan materiks satuan

   Suatu matriks skalar dikatakan sebagai matriks identitas jika semua elemen yang terletak pada diagonal utamanya bernilai satu, sehingga

   matriks identitas disebut juga matriks satuan.

   Misalnya,

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2.7 Matriks Transformasi

Matrik transformasi adalah matrik yang membuat sebuah obyek mengalami perubahan baik berupa perubahan posisi, maupun perubahan ukuran.

2.7.1 Transformasi 2 dimensi

Transformasi 2 dimensi adalah suatu model atau bentuk yang dapat mengubah nilai posisi objek dalam sistem koordinat 2 dimensi. Pemindahan objek ini dapat diartikan sebagai pemindahan titik. Untuk dapat melakukan pemindahan maka digunakan 2 variabel dasar didalam matematika, yaitu vektor dan matriks, dimana

vektor menunjukkan titik objek yang dinyatakan dalm vektor posisi, sedangakan matriks adalah sebuah operator yang akan memindahkan objek.

2.8 Translasi

Translasi digunakan pada suatu objek untuk meletakkan kembali objek tersebut sepanjang jalur lurus dari suatu objek lokasi koordinat ke lokasi koordinat lain. Untuk mentranslasi suatu titik (x,y) dengan pergerakan sebesar tx dan ty menjadi titik (x',y') adalah

$$x' = x + tx$$

$$y' = y + ty \qquad (2.4)$$

Dimana x,y adalah koordinat asal dari titik tx dan ty adalah pergeseran pada sumbu x dan y, x' dan y' adalah koordinat titik sesudah ditranslasi.

Secara singkat persamaan (10) titik ditulis kedalam bentuk matrik dua dimensi. Jika P mempresentasikan koordinat (x,y), P' mempresentasikan koordinat (x',y') dan T mempresentasikan tx dengan ty, maka persamaan (10) dapat ditulis kedalam bentuk matriks menjadi :

$P = \begin{bmatrix} x \\ y \end{bmatrix}$ $P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$ $T = \begin{bmatrix} tx \\ ty \end{bmatrix}$

Dimana untuk mentranslasi suatu titik P menjadi P' adalah

$$P' = P + T \qquad (2.5)$$

atau

$$P = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 1 & 0 & 1 \end{bmatrix} \qquad (2.6)$$

2.9 Skala

Skala adalah operasi untuk membuat sebuah objek lebih besar atau lebih kecil dari bentuk aslinya. Matriks transformasi untuk penskalaan dengan faktor skala terhadap sumbu x dan sumbu y adalah:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (2.7)$$

Dimana Sx > 0 dan Sy > 0

2.10 Rotasi

Rotasi adalah bentuk transformasi dengan dengan memutar sebuah objek, ada dua macam cara atau rumus untuk merotasi object di java yaitu berpatokan pada titik acuan (0,0) atau (p,q) dibawah ini adalah rumus yang digunakan untuk operasi tersebut.Rotasi dengan sudut putar

dengan titik acuan (0,0) adalah:

Maka, koordinat baru hasil rotasi adalah :

x’ = x cosθ – y sinθ

$y' = x \sin y + y \cos\theta$

2.11 Approximate String Matching

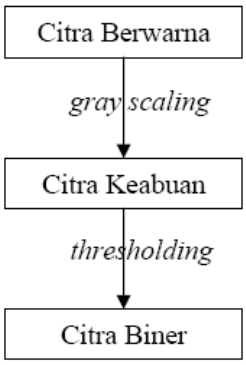
Metode *Approximate String Matching* merupakan metode pencocokkan string. Terdapat dua objek pada metode ini yaitu teks dan pola, tujuan pendekatan ini adalah untuk mencari string didalam sebuah teks atau pola yang mirip dengan string yang ada pada target. Penghitungan kemiripan antara dua string ini menggunakan algoritma *Levensthein distance*.

2.12 Algoritma Levenshtein Distance

Algoritma *levenshtein distance* merupakan algoritma yang digunakan untuk mencari jumlah operasi string yang paling sedikit untuk mentransformasikan suatu *string* menjadi *string* yang lain. Algoritma ini digunakan dalam pencarian *string* dengan pendekatan perkiraan (*Approximate String Matching*). Dengan pendekatan perkiraan ini, pencarian *string* target menjadi tidak harus sama persis dengan yang ada di dalam *string* sumber.

Metode algoritma *levenshtain distance* dapat diterapkan dengan cara mengubah citra sumber dan target menjadi deret karakter atau angka, berikut adalah langkah-langkah proses pengenalan citranya :

1. Merubah citra target menjadi citra biner.



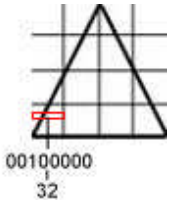
2. Normalisasi citra target menggunakan proses tranformasi agar dapat menjadi citra yang siap untuk dibandingkan, citra yang siap dibandingkan adalah memiliki ukuran yang sama dengan citra sumber yang akan dibandingkan dalam hal ini berukuran 64 pixel x 64 pixel.
3. Segmentasi pada citra sumber dan target dalam bentuk ukuran 8 pixel x 8 pixel.



Citra Sumber Citra Target

Gambar 2.7 Segmentasi Citra

4. Translasi setiap baris pada setiap segment citra sumber dan citra target ditransalasikan menjadi deret angka atau karakter string dengan mengubah setiap baris pixel pada segment menjadi rangkaian angka biner dengan merubah pixel berwarna hitam menjadi angka 1 dan pixel berwarna putih menjadi 0, kemudian rangkaian biner tersebut ditransalasikan kembali menjadi angka desimal atau karakter.



Gambar 2.8 Ilustrasi proses transalasi baris pada segmentasi.

Untuk mempercepat proses pencocokan, tidak perlu melalukan proses sekmentasi dan transalasi untuk citra sumbernya. Dengan cara menyiapkan database yang berisi deret angka atau karakter yang merupakan hasil citra sumber yang di inginkan.

5. Pencarian jarak kecocokan setiap segment pada citra sumber kepada citra target dengan memanfaatkan algoritma *leveinshtain distance.*

Tabel 2.2 Ilustrasi pencocokan segment

t\s		64	32	16	16	8	8	4	4
	0	1	2	3	4	5	6	7	8
64	1	0	1	2	3	4	5	6	7
32	2	1	0	1	2	3	4	5	6
2	3	2	1	1	2	3	4	5	6
16	4	3	2	0	1	2	3	4	5
8	5	4	3	1	1	1	1	2	3
8	6	5	4	2	2	1	1	2	3
8	7	6	5	3	3	1	1	2	3
4	8	7	6	4	4	2	2	1	2

Ilustarasi proses pencocokan segement sumber dengan segment target menggunakan *leveinshtain distance*, pada contoh jarak perbedaan segment antara sumber dan target adalah 2.

2.12.1 Grayscaling

Grayscaling atau *Greyscale* adalah sebuah teknik yang digunakan dalam pengolahan citra untuk menghasilkan sebuah citra yang memiliki nilai dari putih yang

memiliki intensitas paling besar sampai hitam yang memiliki intensitas paling rendah. *Greyscale* sering sekali dipergunakan untuk menghitung intensitas cahaya pada sebuah gambar berwarna. *Greyscale* memiliki 256 intensitas pada gambar 8-bit yang dimulai dari nol(putih) sampai 255(hitam). Berikut tampilkan sebuah contoh gambar berwarna dengan gambar greyscale-nya.

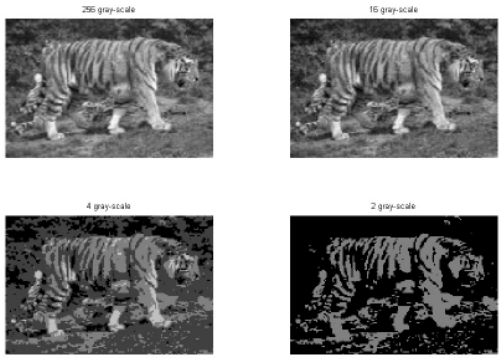


Gambar 2.9 Contoh gambar *grayscaling*

2.12.2 Thresholding

Tresholding digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Default yang ada pada visual basic adalah 256. Dengan menggunakan Tresholding maka derajat keabuan bisa diubah sesuai keinginan.

Sebagai contoh misalkan menggunakan derajat keabuan 16, maka tinggal membagi nilai derajat keabuan dengan 16.



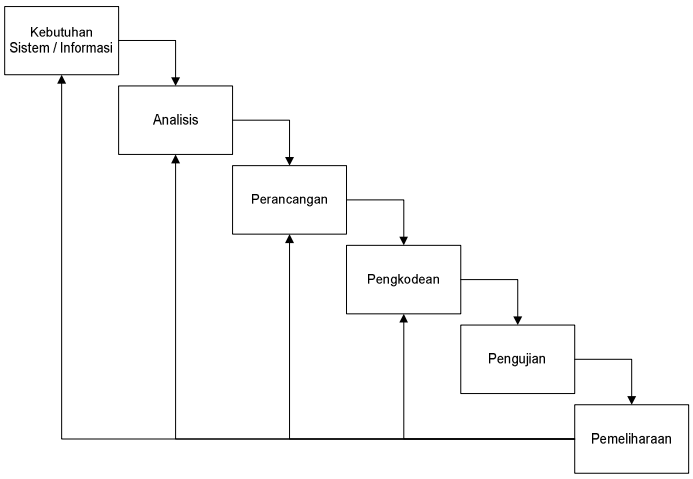
Gambar 2.10 Contoh *thresholding*

2.13 Pengembangan Sistem

1. Model Waterfall

Pada prinsipnya pemodelan sistem waterfall pengembangannya dilakukan secara sistematis dan terarah dari tahap sistem, secara berurutan melalui tahap analisa, tahap desain sistem, coding, testing dan maintenance dan dapat kembali ketahap awal apabila semua tahapan pengembangan sistem telah dilalui. Pemodelan seperti ini juga dikenal sebagai model sekuensial linear "Linear Sequential model".

Metode pengembangan perangkat lunak secara *Waterfall* yang meliputi:



Gambar 2.11 Metode *Waterfall*

Teknik analisis data dalam pembuatan perangkat lunak menggunakan paradigma perangkat lunak secara *waterfall*, yang meliputi beberapa proses diantaranya:

a. KebutuhanSistem / Informasi

   Merupakan bagian dari sistem yang terbesar dalam pengerjaan suatu proyek, di mulai dengan menetapkan berbagai kebutuhan dari semua elemen yang diperlukan sistem dan mengalokasikannya ke dalam pembentukan perangkat lunak.

b. Analisis

   Merupakan tahap menganalisis hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan perangkat lunak, seperti analisis kebutuhan fungsional dan non fungsional.

c. Perancangan

   Tahap perancangan interface yang mudah dimengerti *user* yang mengacu pada data – data analisis.

d. Pengkodean

   Tahap penerjemahan data yang telah dirancang kedalam bahasa pemrograman tertentu.

e. Pengujian

   Merupakan tahap pengujian terhadap perangkat lunak yang di bangun. Pada tahap ini menggunakan dua metode pengujian, yaitu pengujian *alpha* dan

pengujian *beta*. Pengujian *alpha* dilakukan dengan metode *black box* untuk menguji fungsional sistem seperti kesalahan sintak, terminasi dan kesalahan interface apakah sudah sesuai dengan fungsi yang dibangun. Sedangkan pengujian *beta* dilakukan dengan menyebarkan kuesioner.

f. Pemeliharaan

Tahap akhir dimana suatu perangkat lunak yang sudah selesai dapat mengalami perubahan–perubahan atau penambahan sesuai dengan permintaan *user*.

2. Model *Prototype*

Model *prototype* dibangun dari mengumpulkan berbagai kebutuhan, kemudian tim pengembang akan bertemu dengan pelanggan untuk menentukan tujuan dari perangkat lunak, dan mengidentifikasi kebutuhan-kebutuhan yang telah diketahui oleh pelanggan, dan batasan-batasan apa saja yang dapat dikategorikan sebagai tugas utama. Hasilnya akan dibangun rancangan sementara yang mewakili berbagai aspek dari perangkat lunak yang kelak akan digunakan oleh pelanggan/pengguna (seperti bentuk pendekatan input yang digunakan dan bentuk output).

Idealnya model *prototype* melayani sebuah mekanisme untuk mengidentifikasi kebutuhan perangkat lunak. Dimana jika nanti nya sebuah model *prototype* berhasil di buat, seorang *developr* harus berusaha mendayagunakan *tools* yang ada (semisal, *report generator*, *windows manager*) dapat bekerja dengan baik (cepat).

3. Model *RAD*

Rapid Aplication Development (RAD) adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek. Model *RAD* ini merupakan sebuah adaptasi "kecepatan tinggi" dari model sequensial linier dimana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen.

4. Model *Evolusioner*

Model *Evolusioner* adalah model *iterative*. Model itu di tandai dengan tingkah laku yang memungkinkan perekayasa perangkat lunak mengembangkan versi perangkat yang lebih lengkap sedikit demi sedikit.

2.14 Metode Analisis dan Perancangan Terstruktur

Salah satu pendekatan yang digunakan dalam suatu analisis dan desain adalah pendekatan terstruktur. Suatu pendekatan yang bekerja dari sudut pandang yang lebih tinggi menuju tingkat lebih rendah yang lebih rinci, dimana keinginan pemakai disajikan dalam diagram aliran data. Desain terstruktur adalah implementasi secara fisik dan pembagian struktur modular secara hirarki dengan pendekatan atas bawah. Beberapa alat bantu yang digunakan dalam analisis dan desain tersetruktur antara lain:

2.14.1 Diagram Konteks

Diagram konteks merupakan rancangan aliran data utama yang perlu dilakukan mengurai ke level yang lebih tinggi agar proses yang terjadi dapat terlihat jelas.

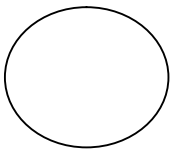
2.14.2 *Data Flow Diagram* (DFD)

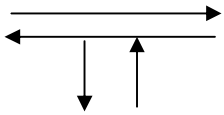
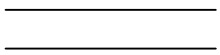
DFD adalah penjelasan lebih rinci dari diagram konteks dan proses fungsional yang ada dalam sistem. DFD mejelaskan tentang aliran masuk, aliran keluar, proses serta penyuntingan file yang digunakan.

DFD digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang dikembangkan secara logika tanpa memperhatikan lingkungan fisik dimana data tersebut mengalir atau disimpan.

DFD sangat berguna untuk mengetahui prosedur suatu program. Keuntungan yang lain adalah mempermudah pemakai atau user yang kurang menguasai komputer untuk mengerti sistem yang akan dibuat.

Tabel 2.3 Tabel Simbol DFD

NO	SIMBOL	NAMA	KETERANGAN
1		Entitas	Menggambarkan kegiatan proses dari operasi program komputer.
2		Proses	Menggambarkan kegiatan atau kerja yang dilakukan oleh manusia atau komputer

3		Garis Alir	Menunjukan arus data yang mengalir antara proses, simpanan data dan kesatuan luar dari ke suatu arus
4		Penyimpanan Data	Menggambarkan suatu tempat penyimpanan data

2.14.3 Kamus Data

"Kamus data atau data directory adalah catalog data tentang fakta dan kebutuhan-kebutuhan informasi dari suatu sistem informasi"

Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Pada tahap perancangan sistem, kamus data dapat digunakan untuk merancang *input*, *output* (laporan-laporan) dan merancang *database* program.

2.14.4 *Entity Relationship Diagram* (ERD)

ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antara penyimpanan. ERD digunangan untuk memodelkan struktur data dan hubungan antar data, karena hal ini relative kompleks. Dengan ERD kita dapat menguji model dengan mengabaikan proses yang dilakukan.

ERD menggunakan Sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data, pada dasarnya ada 3 macam simbol yang digunakan:

1. *Entity*

   Entity adalah suatu objek yang dapat di identifikasi dalam lingkaran pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat.

2. Atribut

   Entity mempunyai elemen yang disebut atribut dan berfungsi mendeskripsikan karakter *entity*.

3. Relasi

   Relasi merupakan gugusan entitas yang berhubungan antar entitas atau beberapa entitas. Macam-macam relasi :

   a. Relasi satu ke satu (*one to one*)

      Setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B, dan berlaku sebaliknya.

   b. Relasi satu ke banyak (*one to many*)

      Setiap entitas pada himpunan A berhubungan dengan banyak entitas pada himpunan entitas B, dan B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.

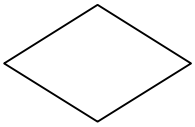
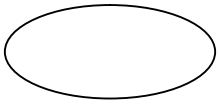
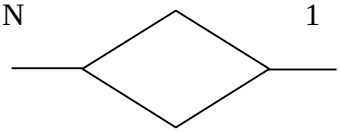
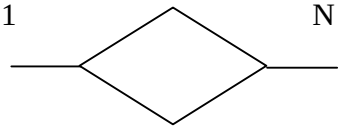
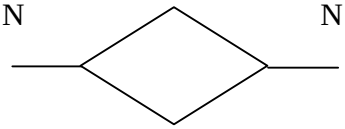
   c. Relasi banyak ke satu (*many to one*)

      Setiap entitas pada himpunan a berhubungan paling banyak dengan satu entitas pada himpunan entitas B, tapi tidak sebaliknya dimana setiap entitas di himpunan A berhubungan paling banyak satu entitas pada himpunan entitas.

   d. Relasi banyak ke banyak (*many to many*)

Setiap entitas pada himpunan A berhubungan paling banyak dengan banyak entitas pada himpunan entitas.

Tabel 2.4 Tabel Simbol ERD

NO	SIMBOL	KETERANGAN
1		Menunjukan suatu objek berupa benda, konsep atau entitas
2		Menunjukan hubungan *logical* antara satu *file* dengan *file* lainnya.
3		Menunjukan atribut suatu entitas
4	N 1 	Menunjukan hubungan banyak ke satu (N-1)
5	1 N 	Menunjukan hubungan satu ke banyak (1-N)
6	N N 	Menunjukan hubungan banyak ke banyak (N-N)

2.15 *Database* MySQL

MySQL adalah multi user *database* yang menggunakan bahasa *Strucktured Query Language* (SQL). MySQL dalam operasi client - *server* melibatkan *server* daemon MySQL di sisi *server* dan berbagai macam program serta *library* yang

berjalan di sisi client. MySQL mampu menangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku menyimpan data lebih dari 40 *database*, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 Gigabyte data.

SQL adalah bahasa standar yang digunakan untuk mengakses *database server*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai standar industri. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user - friendly* dibandingkan dengan menggunakan dBASE atau Clipper yang masih menggunakan perintah - perintah pemrograman.

MySQL merupakan *software database* yang paling populer di lingkungan Linux, kepopuleran ini karena ditunjang performa query dari *database*nya yang saat ini bisa dikatakan paling cepat dan jarang bermasalah. MySQL ini juga sudah dapat berjalan pada lingkungan Windows.

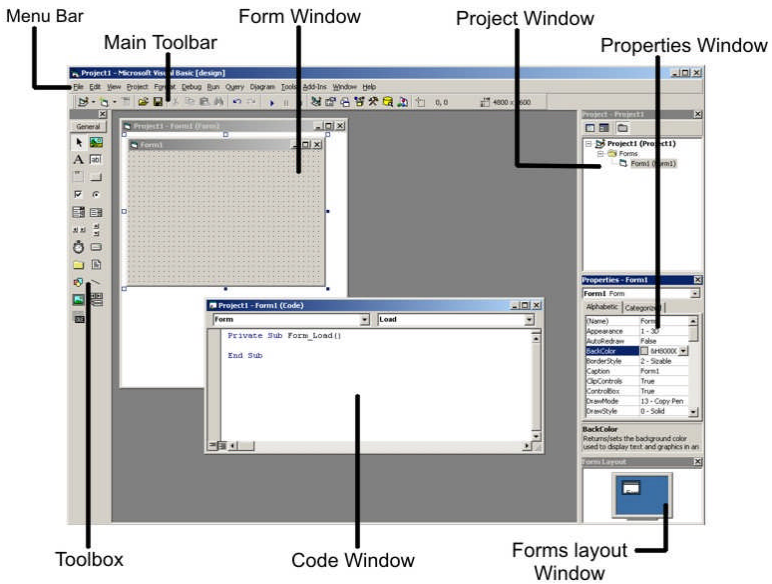
2.16 Pemrograman Visual Basic 6.0

Visual Basic termasuk kategori bahasa pemrograman tingkat tinggi (*high-level programming language*) yang merupakan pengembangan dari versi DOS yang dikenal dengan nama BASIC. BASIC merupakan singkatan dari Beginner's Allpurpose Symbolic Instruction Code. Bahasa ini cukup mudah untuk dipelajari. Kode-kode yang digunakan mirip dengan bahasa Inggris pada umumnya. Bahasa BASIC dikembangankan oleh beberapa perusahaan pengembang perangkat lunak,

sehingga banyak bermunculan varian dari BASIC, seperti : Microsoft QBASIC, QUICKBASIC, GWBASIC, IBM BASICA dan lain-lain.

Visual Basic adalah versi "visual" dari bahasa BASIC. Pada bahasa BASIC, pembuatan program hanya dilakukan di lingkungan berbasis teks (*text-base*) dan dieksekusi secara sekuensial. Dengan Visual Basic, pembuatan program dilakukan dalam lingkungan grafik (graphical-environment). Pada pemakaiannya, progamer cukup melakukan "*double-click*" pada objek tertentu dan setiap objek dapat diprogram secara independen sehingga dapat merespon perintah-perintah yang dituliskan.

Versi program Visual Basic yang akan digunakan adalah versi 6.0.



Gambar 2.12 Lingkungan Visual Basic dengan window antarmuka standar

Keterangan :

1. Menu Bar

Terdiri dari menu-menu standar aplikasi berbasis Windows™ dan seluruh fasilitas pengaturan penggunaan aplikasi Visual Basic dan/atau aplikasi yang sedang dibangun. Menu yang sering diakses biasanya terdapat pada *toolbar*.

2. Main Toolbar

   Visual Basic memiliki toolbar standar yang memuat perintah-perintah umum, seperti : membuka dan menyimpan project, menjalankan program dan untuk membuka window-window yang lain.

3. Form Window

   Digunakan untuk mendesain antarmuka dengan cara meletakkan komponen-komponen yang digunakan dalam aplikasi.

4. Project Window

   Berisi ikhstisar atau gambaran dari seluruh modul yang digunakan dalam aplikasi yang sedang dibangun.

5. Properties Window

   Terdiri dari daftar properti yang dimiliki oleh sebuah komponen yang sedang dipilih dan dapat juga digunakan untuk memodifikasi nilai-nilai properti dari komponen yang bersangkutan.

6. Forms layout Window

   Digunakan untuk menunjukkan dan mengatur tata letak form yang dipilih jika aplikasi dijalankan.

7. Code Window

Digunakan untuk menulis kode-kode program untuk menentukan perilaku dari suatu komponen atau beberapa komponen yang digunakan pada aplikasi.

8. Toolbox

   Terdiri dari sekumpulan komponen yang dapat diletakkan pada form.