BAB II

TEORI PENUNJANG

2.1 Pucuk Daun Teh

Teh adalah hasil pengolahan pucuk (daun muda) dari tanaman teh (*Camellia sinensis*) yang dipakai sebagai bahan minuman. Teh yang baik dihasilkan dari bagian pucuk peko (pucuk paling ujung atau tunas yang sedang aktif) ditambah 2-3 helai daun muda, karena pada daun muda tersebut kaya akan senyawa polifenol. Istilah teh juga digunakan untuk minuman yang dibuat dari buah, rempah-rempah atau tanaman obat lain yang diseduh, misalnya teh *rosehip, camomile, krisan* dan *Jiaogulan*. Teh yang tidak mengandung daun teh disebut teh herbal.

Teh merupakan sumber alami kafein, *teofilin* dan antioksidan dengan kadar lemak, karbohidrat atau protein mendekati nol persen. Teh bila diminum terasa sedikit pahit yang merupakan kenikmatan tersendiri dari teh. Teh bunga dengan campuran kuncup bunga melati yang disebut teh melati atau teh wangi melati merupakan jenis teh yang paling populer di Indonesia. Konsumsi teh di Indonesia sebesar 0,8 kilogram per kapita per tahun masih jauh di bawah negara-negara lain di dunia, walaupun Indonesia merupakan negara penghasil teh terbesar nomor lima di dunia.

Pengelompokan teh berdasarkan tingkat oksidasi:

a. Teh Putih.

Teh yang dibuat dari pucuk daun yang tidak mengalami proses oksidasi dan sewaktu belum dipetik dilindungi dari sinar matahari untuk menghalangi pembentukan klorofil (zat hijau). Teh putih diproduksi dalam jumlah lebih sedikit dibandingkan teh jenis lain sehingga harga menjadi lebih mahal. Teh putih kurang terkenal di luar Tiongkok, walaupun secara perlahan-lahan teh putih dalam kemasan teh celup juga mulai populer.

b. Teh Hijau

Daun teh yang dijadikan teh hijau biasanya langsung diproses setelah dipetik. Setelah daun mengalami oksidasi dalam jumlah minimal, proses oksidasi dihentikan dengan pemanasan (cara tradisional Jepang dengan menggunakan uap atau cara tradisional Tiongkok dengan memasak di atas wajan panas). Teh yang sudah dikeringkan bisa dijual dalam bentuk lembaran daun teh atau digulung rapat berbentuk seperti bola-bola kecil (teh yang disebut *gun powder*).

c. Teh Hitam atau Teh Merah.

Daun teh dibiarkan teroksidasi secara penuh sekitar 2 minggu hingga 1 bulan. Teh hitam merupakan jenis teh yang paling umum di Asia Selatan (India, Sri Langka, Bangladesh) dan sebagian besar negara-negara di Afrika seperti: Kenya, Burundi, Rwanda, Malawi dan Zimbabwe. Orang Barat menyebutnya sebagai teh hitam karena daun teh berwarna hitam. Di Afrika Selatan, teh merah adalah sebutan untuk teh *rooibos* yang termasuk golongan teh herbal. Teh hitam masih dibagi menjadi 2 jenis: Ortodoks (teh diolah dengan metode pengolahan tradisional) atau CTC (metode produksi teh *Crush, Tear, Curl* yang berkembang sejak tahun 1932). Teh hitam yang belum diramu (*unblended*) dikelompokkan berdasarkan asal perkebunan, tahun produksi, dan periode pemetikan (awal musim semi, pemetikan kedua, atau musim gugur). Teh jenis Ortodoks dan CTS masih dibagi-bagi lagi menurut kualitas daun pasca produksi sesuai standar *Orange Pekoe*.

d. *Pu-erh* atau Teh Matang(*Póu léi* dalam bahasa Kantonis)

Teh *pu-erh* terdiri dari dua jenis: mentah dan matang. Teh *pu-erh* yang masih mentah bisa langsung digunakan untuk dibuat teh atau disimpan beberapa waktu hingga matang. Selama penyimpanan, teh *pu-erh* mengalami oksidasi mikrobiologi tahap kedua. Teh *pu-erh* (matang) dibuat dari daun teh yang mengalami oksidasi secara artifisial supaya menyerupai rasa teh *pu-erh* mentah yang telah lama disimpan dan mengalami proses penuaan alami. Teh *pu-erh* matang dibuat dengan mengontrol kelembaban dan temperatur daun teh mirip dengan proses pengomposan. Teh *pu-erh* biasanya dijual dalam bentuk padat setelah dipres menjadi seperti batu bata, piring kecil atau mangkuk. Teh *pu-erh* dipres agar proses oksidasi tahap kedua bisa berjalan, karena teh *pu-erh* yang tidak dipres tidak akan mengalami proses pematangan. Semakin lama disimpan,

aroma teh *pu-erh* menjadi semakin enak. Teh *pu-erh* yang masih mentah kadang-kadang disimpan sampai 30 tahun bahkan 50 tahun supaya matang.

2.1.1 Kualitas

Kualitas atau mutu adalah tingkat baik buruknya pucuk daun teh yang ada pada pohon teh, kualitas akan menentukan rasa dan aroma.

2.2 Definisi Pengolahan Citra

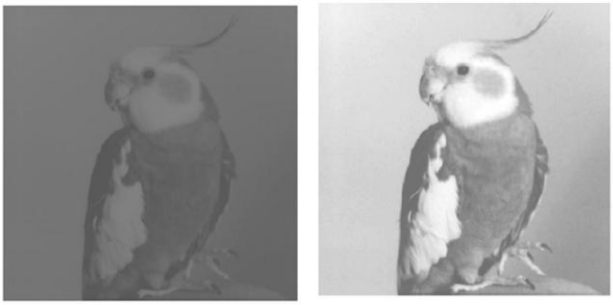
Pengolahan Citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Dalam definisi yang lebih luas, pengolahan citra digital juga mencakup semua data dua dimensi. Citra digital adalah barisan bilangan nyata maupun kompleks yang diwakili oleh bit-bit tertentu.

Citra merupakan istilah lain dari gambar yang merupakan komponen multimedia yang memegang peranan sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu kaya akan informasi. Citra digital adalah citra hasil digitalisasi citra kontinu (analog). Tujuan dibuatnya citra digital adalah agar citra tersebut dapat diolah menggunakan komputer atau piranti digital dan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain yang mempunyai kualitas lebih baik.



Gambar 2.1 Diagram Citra Menjadi Citra lain yang Mempunyai Kualitas Lebih Baik.

Dibawah ini adalah contoh gambar yang melalui proses perbaikan citra menjadi kualitas lebih baik:



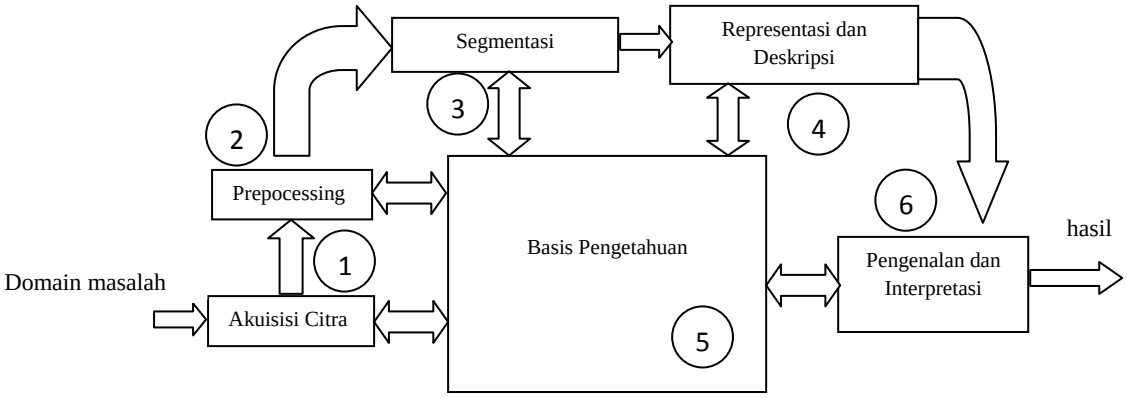
Gambar 2.2 Proses citra menjadi citra lain yang mempunyai kualitas lebih baik.

Keterangan:

Proses dimana gambar (a) yang semula dengan gambar yang kurang pencahayaan dan pelembutan dan gambar (b) proses dimana setelah atau sesedah mengalami pelembutan dan penambahan kontras yang menjadi lebih bagus.

2.3 Langkah-Langkah Penting dalam Pengolahan Citra (*image prosesing*)

Secara umum, langkah-langkah dalam pengolahan citra dapat di jabarkan menjadi beberapa langkah yang digambakan oleh gambar 2.3.



Gambar 2.3 Tahap-Tahap Pengolahan Citra Digital.

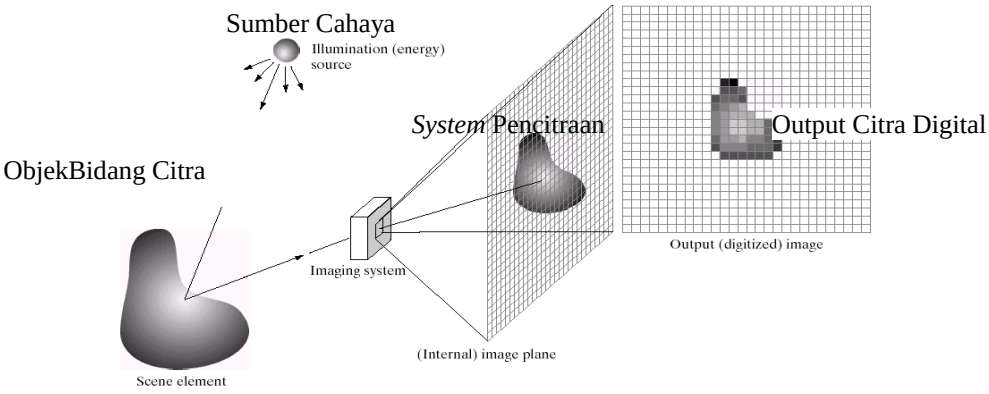
1. Akuisisi Citra

Akuisisi citra adalah tahap awal untuk mendapatkan citra digital. Tujuan akuisisi citra adalah untuk menentukan data yang diperlukan dan memilih metode perekaman citra digital. Tahap ini dimulai dari objek yang akan diambil gambarnya, persiapan alat-alat, sampai pada pencitraan. Pencitraan adalah kegiatan transformasi dari citra tampak (foto, lukisan, gambar, patung,

pemandangaan dan lain-lain) menjadi citra digital. Beberapa alat yang dapat digunakan untuk pencitraan adalah:

a. *Video* kamera
b. Kamera digital
c. Kamera konvesional dan *converter* analog to digital
d. Scanner
e. Photo sinar-x atau sinar infra merah

Gambar proses akuisisi citra:



Gambar 2.4 Proses Akuisisi Citra.

Dimana hasil dari akuisisi citra ini ditentukan oleh kemampuan sensor untuk proses digitalisasi sinyal yang terkumpul pada sensor tersebut. Kemampuan digitalisai alat ditentukan oleh resolusi alat tersebut.

2. Preprocessing

Tahapan ini diperlukan untuk menjamin kelancaran pada proses berikutnya. Hal-hal penting yang dilakukan pada tingkatan ini diantaranya adalah:

a. Peningkatan kualitas citra (kontras, *brightness*, dan lain-lain)
b. Menghilangkan *nois*
c. Perbaikan citra (*image restoration*)
d. Transformasi (*image transformasi*)
e. Menentukan bagian citra yang akan diobservasi

3. Segmentasi

Tahapan ini bertujuan untuk mempartisi citra menjadi bagian-bagian poko yang mengandung informasi penting. Misalnya, memisahkan objek dari latar belakang.

4. Representasi dan deskripsi

Dalam hal ini representasi merupakan suatu proses untuk merepresentasikan suatu wilayah sebagai suatu daftar titik-titik koordinat dalam kurva yang tertutup, dengan deskripsi luasan atau perimeternya. Setelah suatu wilayah dapat direpresentasikan, proses selanjutnya adalah melakukan deskripsi citra dengan cara seleksi ciri dan ekstraksi ciri (*Feature Extraction and selection*). Seleksi ciri bertujuan untuk memilih informasi kuantitatif dari ciri yang ada, yang dapat membedakan kelas-kelas objek secara baik, sedangkan ekstraksi ciri bertujuan untuk mengukur besaran kuantitatif ciri setiap piksel, misalnya rata-rata, standar deviasi, koefisien variasi, signal to nois ratio (SNR), dan lain-lain.

5. Pengenalan dan interpretasi

Tahap pengenalan bertujuan untuk memberi label pada suatu objek yang informasinya disediakan oleh *descriptor,* sedangkan tahap interpretasi bertujuan untuk memberi arti atau makna kepada kelompok objek-objek yang dikenali.

6. Basis pengetahuan

Basis pengetahuan sebagai basis data pengetahuan berguna untuk memandu operasi dari masing-masing modul proses dan mengkontrol interaksi antara modul-modul tersebut. Selain itu, basis pengetahuan juga digunakan sebagai referensi pada proses *template matching* atau pada pengenalan pola.

2.4 Dasar-Dasar Pengolahan Citra Digital

Hal penting pada pengolahan citra digital antara lain:

a. Cara-cara pengambilan citra
b. Format dan model citra digital
c. Sampling dan kuantisasi
d. *Threshold*

e. Histogram

f. Proses *filtering*, perbaikan citra sampai pada pengolahan citra digital yang lebih lanjut seperti *segmentasi*, *image clustering* dan ekstraksi ciri.

2.5 Macam-Macam Cara Pengambilan Citra Digital

Beberapa cara pengambilan citra digital dilakukan antara lain dengan menggunakan kamera digital (bisa juga *web-cam*) atau menggunakan scanner (bisa scanner umum, *finger-print* scanner atau *bar-code* scanner). Hasil dari citra yang diambil menggunakan kamera atau scanner berupa citra raster (atau citra dengan model matrik).

Citra juga merupakan bentuk dari dua dimensi untuk di jadikan fisik nyata menjadi tiga dimensi. Dalam perwujudannya, citra dibagi menjadi dua yaitu *still images* (citra diam) dan *moving images* (citra bergerak). Citra diam adalah citra tunggal yang tidak bergerak, sedangkan citra bergerak adalah rangkaian citra diam yang ditampilkan secara beruntun (*sekuensial*) sehingga memberi kesan pada mata kita sebagai gambar yang bergerak.

Gambar atau citra dapat disebut sebagai sebuah bidang datar yang mempunyai fungsi dua dimensi *f(x, y)*, dimana nilai *x* dan *y* merupakan koordinat pada sebuah bidang datar dan amplitudo dari *f* dapat disebut sebagai intensitas atau *gray*-level atau biasa disebut tingkat ke abu-abuan dari sebuah gambar pada koordinat *x* dan *y*.

2.6 Pembentukan Citra

Pembentukan citra membutuhkan suatu cahaya sebagai salah satu parameter utamanya. Sejumlah cahaya akan menerangi objek secara berulang dalam kurun waktu tertentu. Oleh objek, cahaya tersebut lalu dipantulkan kembali hingga ditangkap oleh alat-alat optik.

Pembentukan citra ada dua macam yaitu:

a. Citra Kontinu

   Dihasilkan dari sistem optik yang menerima sinyal analog.

   Contoh : mata manusia, kamera analog.

b. Citra Diskrit atau Citra Digital

Dihasilkan melalui proses digitalisasi terhadap citra kontinu.

Contoh : kamera digital, scanner.

2.6.1 Macam-Macam Operasi Pengolahan Citra

Pada dasarnya pengolahan citra dilakukan dengan cara mengubah nilai setiap titik (píxel) pada citra tersebut, macam-macam cara mengubah nilai pixel dapat dikelompokan sebagi berikut :

1). Operasi titik, Pada operasi ini setiap titik (píksel) di rubah-rubah tanpa melibatkan titik yang lain pada citra tersebut.
2). Operasi global, Pada operasi ini terlebih dahulu diambil karakteristik global dari suatu citra kemudian dengan menggunakan karakter global sebuah citra maka proses pengubahan titik yang lain dilakukan.
3). Operasi berbasis bingkai, Pada operasi ini proses pengubahan nilai citra memerlukan citra yang lain, dengan kata lain citra 1 dioperasikan dengan citra 2 menghasilkan citra 3.
4). Operasi geometri, Pada operasi ini orientasi citra (bentuk, ukuran, kemiringan) dirubah secara geometris.
5). Opersi banyak titik bertetangga, Pada operasi ini sebuah pixel atau titik dirubah nilainya dengan melibatkan nilai tetangganya.
6). Operasi morfologi, Pada operasi ini sebuah pixel atau titik dirubah dengan melihat bagian dalam yang dimiliki sebuah citra yang mendapat perhatian khusus.

Pengolahan citra dari tahun ke tahun berikutnya telah mengalami berbagai perkembangan dan dalam banyak penerapan pengolahan citra ini digunakan sebagai pengganti mata yang diletakkan pada komputer. Untuk mengambil sebuah citra komputer memerlukan sebuah peralatan yang disebut *image capture*.

Peralatan *image capture* diantaranya adalah kamera dan scanner. Data citra yang telah didapat dari kamera dan scanner selanjutnya diproses oleh komputer dengan menggunakan langkah-langkah tertentu (algoritma) untuk menghasilkan sebuah citra yang lain atau sebuah keputusan.

2.7 Citra Analog dan Citra Digital

Citra analog adalah citra yang bersifat kontinu, seperti gambar pada monitor televisi, foto sinar-X, foto yang tercetak dikertas foto, lukisan, pemandangan, hasil CT scan, gambar-gambar yang terekam pada pita kaset, dan lain-lain sebagainya.

Citra analog tidak dapat direpresentasikan dalam komputer sehingga tidak dapat diproses dikomputer secara langsung. Oleh sebab itu, agar citra ini dapat diproses dikomputer, proses konversi analog ke digital harus dilakukan terlebih dahulu. Citra analog dihasilkan dari alat-alat analog, *video* kamera analog, kamera foto analog, *Web Cam*, *CT scan*, sensor *ultrasound* pada system USG, dan lain-lain .

Citra Digital adalah citra yang dapat diolah oleh komputer dan citra digital yaitu gambar pada bidang dua dimensi. Dalam tinjauan matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi. Ketika sumber cahaya menerangi objek, objek memantulkan kembali sebagian cahaya tersebut. Pantulan ini ditangkap oleh alat-alat pengindera optik, misalnya mata manusia, kamera, scanner dan sebagainya. Bayangan objek tersebut akan terekam sesuai intensitas pantulan cahaya. Ketika alat optik yang merekam pantulan cahaya itu merupakan mesin digital, misalnya kamera digital, maka citra yang dihasilkan merupakan citra digital. Pada citra digital, kontinuitas intensitas cahaya dikuantisasi sesuai resolusi alat perekam.

2.8 Model Citra Sederhana

Sensor optik yang terdapat di dalam *system* pencitraan disusun sedemikian rupa sehingga membentuk dua dimensi (x, y). Besar intensitas yang diterima sensor di setiap titik (x, y) disimbolkan oleh $f\ (x, y)$ dan besarnya tergantung pada intensitas yang dipantulkan oleh objek. Ini berarti $f(x, y)$ sebanding dengan energi yang dipancarkan oleh sumber cahaya. Konsekuensinya, besar intensitas $f(x, y)$ tidak boleh nol dan harus berhingga yaitu:

$$0 < f(x, y) < \infty \quad \text{.....(2.1)}$$

Fungsi $f(x, y)$ dapat dipisahkan menjadi dua komponen yaitu:

1. Jumlah cahaya yang berasal dari sumbernya disimbolkan oleh $i(x,y)(illumination)$ nilainya antara 0 dan ∞.
2. Derajat kemempuan objek memantulkan cahaya $r(x,y)(reflection)$, nilainya antara 0 dan 1. Besar $f(x,y)$ merupakan kombinasi perkalian dari keduanya.

$$f(x,y) = i(x,y).r(x,y) \quad (2.2)$$

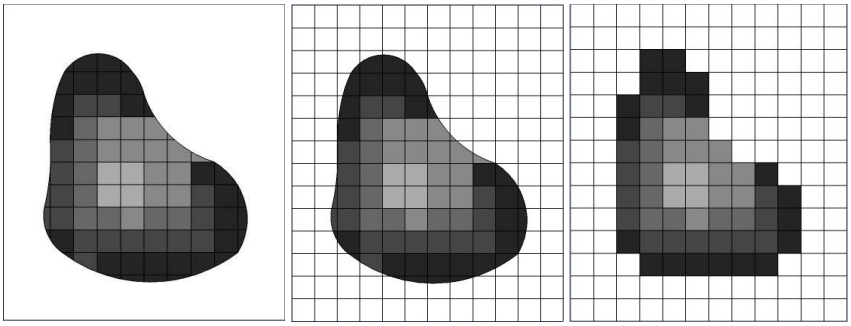
dimana

$$0 < i(x,y) < \infty \text{ dan } 0 < r(x,y) < 1 \quad (2.3)$$

Nilai $i(x,y)$ ditentukan oleh sumber cahaya, nilai $i(x,y) = 1$ menyatakan pemantulan total, nilai $r(x,y)$ ditentukan oleh karakteristik objek didalam gambar, nilai $r(x,y) = 0$ mengindikasikan penyerapan total. Jika permukaan mempunyai derajat pemantulan nol maka fungsi intensitas cahaya $f(x,y)$ juga nol.

2.9 Sampling

Sampling adalah transformasi citra kontinu menjadi citra digital dengan cara membagi citra analog (kontinu) menjadi M kolom dan N baris sehinnga menjadi citra diskrit. Semakin besar nilai M dan N, semakin halus citra digital yang dihasilkan dan artinya resolusi citra semakin semakin tinggi. Persilangan antara baris dan kolom tertentu disebut dengan piksel.



Gambar 2.5 (a) Citra Analog, (b) Citra Analog Disampling Menjadi 14 Baris dan 12 Kolom, (c) Citra Digital Hasil Sampling Berukuran 14 x 12 Piksel.

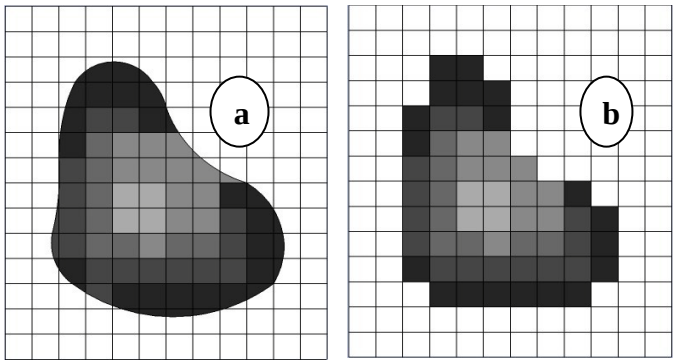
Gambar diatas menunjukkan proses *sampling* dari citra analog menjadi citra digital berukuran 14 x 12 piksel yang mengakibatkan adanya beberapa bagian informasi citra yang hilang.

2.10 Kuantisasi

Warna sebuah citra digital ditentukan oleh besar intensitas piksel-piksel penyusunnya. Warna ini diperoleh dari besar kecilnya intensitas cahaya yang ditangkap oleh sensor. Sedangkan skala intensitas cahaya dialam tidak terbatas, yang bisa menghasilkan warna dengan jumlah yang tak terhingga. Sampai saat ini belum ada satu sensor pun yang mampu menangkap seluruh gradasi warna tersebut.

Keterbatasan inilah yang mengharuskan kita membuat gradasi warna sesuai dengan kebutuhan. Transformasi intensitas analog yang bersifat kontinu ke daerah intensitas diskrit disebut kuantisasi. Proses kuantisasi dihasilkan oleh peralatan digital, misalnya *scanner*, foto digital, dan kamera digital.

Perhatikan Gambar 2.6 (a) dan (b). Misalnya besar memori yang digunakan untuk meyimpan warna adalah 3 bit maka gradasi warna adalah citra analog. Dan Gambar 2.6 (a) hanya diwakili oleh gradasi warna 3 bit. Kemudian, dilakukan kuantisasi untuk setiap piksel. Warna tiap-tiap piksel disesuaikan dengan gradasi warna yang disediakan oleh memori.



Gambar 2.6 Proses Kuantisasi

7	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**
7	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**
7	**7**	**0**	**0**	**7**	**7**	**7**	**7**	**7**	**7**	**7**
7	**7**	**0**	**0**	**0**	**7**	**7**	**7**	**7**	**7**	**7**
7	**0**	**0**	**2**	**4**	**7**	**7**	**7**	**7**	**7**	**7**
7	**0**	**3**	**4**	**4**	**4**	**7**	**7**	**7**	**7**	**7**
7	**0**	**3**	**5**	**5**	**4**	**4**	**0**	**7**	**7**	**7**
7	**0**	**3**	**5**	**5**	**4**	**4**	**0**	**0**	**7**	**7**
7	**0**	**3**	**2**	**4**	**0**	**0**	**0**	**0**	**7**	**7**
7	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**7**	**7**
7	**7**	**0**	**0**	**0**	**0**	**0**	**0**	**7**	**7**	**7**
7	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**
7	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**

Gambar 2.7 Matriks Berisikan Intensitas.

Hasil citra digital yang disimpan oleh memori hanyalah nilai-nilai intensitas yang ditunjukkan pada Gambar 2.7 yang berbentuk matriks berukuran 14 baris x 11 kolom. Setelah tiap-tiap piksel dikuantisasi, nilai-nilai intensitas diperoleh sebagai berikut.

Nilai-nilai diatas diperoleh setelah dikuantisasi, kemudian untuk selanjutnya akan ditulis dalam bentuk asimetris. Bila cita digital tersebut ditulis dalam bentuk asimetris sebagai fungsi $f(x, y)$ maka :

1	2	3	4	5	6	7	8	9	10	11	
7	7	7	7	7	7	7	7	7	7	7	1
7	7	7	7	7	7	7	7	7	7	7	2
7	7	0	0	7	7	7	7	7	7	7	3
7	7	0	0	0	7	7	7	7	7	7	4
7	0	0	0	0	7	7	7	7	7	7	5
7	0	0	2	4	7	7	7	7	7	7	6
7	0	3	4	4	4	7	7	7	7	7	7
7	0	3	5	5	4	4	0	7	7	7	8
7	0	3	5	5	4	4	0	0	7	7	9
7	0	3	2	4	0	0	0	0	7	7	10
7	0	0	0	0	0	0	0	0	7	7	11
7	7	0	0	0	0	0	0	7	7	7	12
7	7	7	7	7	7	7	7	7	7	7	13
7	7	7	7	7	7	7	7	7	7	7	14

Gambar 2.8 Matriks Nilai Asimetris

Keterangan:

F(x,y).

F(3,10) = 7 artinya piksel di titik (3, 10) mempunyai nilai intensitas sebesar 7

F(6,5) = 4 artinya piksel di titik (6,5) mempunyai nilai intensitas sebesar 4

F(8,5) = 5 artinya piksel di titik (8,5) mempunyai nilai intensitas sebesar 5

F(10,4) = 2 artinya piksel di titik (10,4) mempunyai nilai intensitas sebesar 2

F(12,8) = 0 artinya piksel di titik (12,8) mempunyai nilai intensitas sebesar 0

2.11 Representasi Citra Digital

Sebuah citra digital dapat mewakili oleh sebuah matriks yang terdiri dari M kolom N baris, dimana perpotongan antara kolom dan baris disebut piksel (piksel = *picture* element), yaitu elemen terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x, y) adalah $f(x, y)$, yaitu besar intensitas atau warna dari piksel di titik itu. Oleh sebab itu, sebuah citra digital dapat ditulis dalam bentuk matriks berikut.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \dots & \dots & & \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1,M-1) \end{bmatrix}$$

Pada Matlab representasi ini bisa ditulis menjadi :

$$\begin{bmatrix} f(1,1) & f(0,1) & \dots & f(1,N) \\ f(2,1) & f(2,2) & \dots & f(2,N) \\ \dots & \dots & \dots & \dots \\ f(M,1) & f(M,2) & \dots & f(3,N) \end{bmatrix}$$

Yang perlu diperhatikan adalah bahwa indeks matriks pada Matlab selalu dimulai dengan angka 1 sehingga untuk $f(0,0)$ akan sama dengan $f(1,1)$ pada Matlab.

Berdasarkan gambaran tersebut, secara matematis citra digital dapat dituliskan sebagai fungsi intensitas $f(x,y)$, dimana x (baris) dan y (kolom) merupakan koordinat posisi dan $f(x,y)$ adalah nilai fungsi pada setiap titik (x,y) yang menyatakan besar intensitas citra atau tingkat keabuan atau warna dari piksel di titik tersebut. Pada proses digitalisasi (sampling dan kuantitas) diperoleh besar baris M dan kolom N hingga citra membentuk matriks $M\ x\ N$ dan jumlah tingkat keabuan piksel G, biasanya besar M, N dan G adalah perpangkatan dari dua.

$$M = 2^m N = 2^n\ dan\ G = 2^k \quad \text{……..…...……(2.4)}$$

Dalam hal ini $m, n, dan\ k$ adalah bilanga bulat positif. Jika b menyatakan jumlah bit yang diperlukan untuk menyimpan citra digital dalam memori, maka:

$$b = mxnxk \quad \text{……..……..…(2.5)}$$

Bentuk matriks ini kemudian diolah menurut teori-teori tertentu yang bertujuan untuk memecahkan masalah tertentu, bentuk matriks adalah perwujudan dari bentuk sinyal digital sehingga proses pemecahan dan pengolahan matriks dari gambar ini biasanya disebut dengan digital *image processing*.

2.12 Jenis-jenis Citra Digital

Nilai suatu *pixel* memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Namun secara umum jangkauannya adalah 0 – 255. Citra dengan penggambaran seperti ini digolongkan kedalam citra *integer*. Berikut adalah jenis-jenis citra berdasarkan nilai *pixel*-nya.

2.12.1 Citra Biner (Monokrom)

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai *pixel* yaitu hitam dan putih. Citra biner juga disebut sebagai citra B&W (*black and white*) atau citra monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap *pixel* dari citra biner. Citra biner diperoleh melalui proses pemisahan piksel-piksel berdasarkan derajat keabuan yang dimilikinya.

Citra biner sering kali muncul sebagai hasil dari proses pengolahan seperti segmentasi, pengambangan, morfologi ataupun *dithering*.

0	1



Gambar 2.9 Contoh Citra Biner.

Bit 0 = warna hitam

Bit 1 = warna putih

2.12.2 Citra *Grayscale* (Sekala Keabuan)

Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap *pixel*-nya, dengan kata lain nilai bagian *RED* = *GREEN* = *BLUE*. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas warna. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Tingkatan keabuan di sini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih. Citra *grayscale* berikut memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan).

Semakin besar jumlah bit warna yang disediakan dimemori, maka semakin halus gradasi warna yang terbentuk.

Citra 2 bit mewakili 4 warna dengan gradasi warna berikut:



Citra 3 bit mewakili 8 warna dengan gradasi warna berikut:





Gambar 2.10 Contoh Citra *Grayscale*.

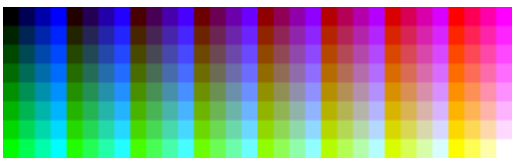
Semakin besar jumlah bit warna yang disediakan di memori, smakin halus gradasi warna yang terbentuk.

2.12.3 Citra Warna (8 bit)

Setiap *pixel* dari citra warna (8-bit) hanya di wakili oleh 8 bit=1 $byte$ dengan jumlah warna maksimum yang dapat digunakan adalah 256 warna. Ada dua jenis citra warna 8 bit. Pertama, citra warna 8 bit dengan menggunakan palet warna 256 dengan setiap paletnya memiliki pemetaan nilai (*colormap*) RGB tertentu. Model ini lebih sering digunakan. Kedua, setiap *pixel* memiliki format 8 bit sebagai berikut.

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
R	R	R	R	R	R	R	R

Bentuk kedua ini dinamakan 8 bit *truecolor*. Berikut ini adalah warna-warnanya :





Gambar 2.11 Citra Warna (8 bit)

2.12.4 Citra Warna (16 bit)

Citra warna 16 bit (biasanya disebut sebagai citra *high color*) dengan setiap *pixel*-nya diwakili dengan 2 *byte* memori (16 bit). Warna 16 Bit memiliki 65.536 warna. Dalam formasi bitnya, nilai merah dan biru mengambil tempat di 5 bit di kanan dan kiri. Komponen hijau memiliki 5 bit ditambah 1 bit ekstra. Pemilihan komponen hijau dengan deret 6 bit dikarenakan penglihatan manusia lebih sensitif terhadap warna hijau.

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R	R	R	R	R	G	G	G	G	G	G	B	B	B	B	B

Berikut ini adalah deret warna yang dihasilkan dari warna 16 bit.



Gambar 2.12 Contoh Citra Warna (16 bit)

2.12.5 Citra Warna (24 bit) atau *True Color*

Setiap *pixel* dari citra warna 24 bit diwakili dengan 24 bit sehingga total 16.777.216 variasi warna. Variasi ini sudah lebih dari cukup untuk memvisualisasikan seluruh warna yang dapat dilihat penglihatan manusia. Penglihatan manusia dipercaya hanya dapat membedakan hingga 10 juta warna saja.

Setiap poin informasi *pixel* (RGB) disimpan ke dalam 1 *byte* data. 8 bit pertama menyimpan nilai biru, kemudian diikuti dengan nilai hijau pada 8 bit kedua dan pada 8 bit terakhir merupakan warna merah.



Gambar 2.13 Contoh Citra Warna (24 bit).

Sehingga dalam aplikasi ini yang dipakai adalah citra 24 bit atau *true color*.

2.12.5.1 Citra Warna (24 bit) atau *True Color*

Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB= *Red, Green, Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 *byte*, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak $2^8 .2^8 .2^8 = 2^{24} = 16$ juta warna lebih. Itulah sebabnya format ini dinamakan *true color* karena mempunyai jumlah warna yang cukup besar sehingga bias dikatakan hampir mencakup semua warna di alam.

Penyimpanan citra *true color* di dalam memori berbeda dengan citra *grayscale.* Setiap piksel dari citra *grayscale* 255 gradasi warna diwakili oleh 1 *byte.* Sedangkan 1 piksel citra *true color* diwakili oleh 3 *byte*, dimana masing-

masing *byte* merepresentasikan warna merah (*Red*), hijau (*Green*), dan biru (*Blue*).

2.13 Elemen Citra

Citra mengandung sejumlah elemen dasar. Elemen dasar tersebut di manipulasi dalam pengolahan citra, elemen tersebut adalah.

1. Warna

   Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang. Warna yang diterima oleh mata merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda. Kombinasi warna yang memberikan rentang warna yang paling lebar adalah *red* (R), *green* (G), *blue* (B).
2. Kecerahan (*brightness*)

   Kecerahan disebut juga intensitas cahaya. Kecerahan pada sebuah piksel (titik) didalam citra bukanlah intensitas yang *rell*, tetapi sebenarnya adalah intensitas rata-rata dari suatu area yang melingkupinya.
3. Kontras (*contrast*)

   Kontras menyatakan sebaran terang dan gelap di dalam sebuah gambar. Citra dengan kontras rendah dicirikan oleh sebagian besar komposisi citranya adalah terang atau sebagian besar gelap. Pada citra dengan kontras yang baik, komposisi gelap dan terang tersebar secara merata.
4. Kontur (*contour*)

   Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada piksel yang bertetangga. Karena adanya perubahan intensitas, mata manusia dapat mendeteksi tepi objek didalam citra.
5. Bentuk (*shape*)

   Bentuk adalah properti intrinsik dari objek tiga dimensi, dengan pengertian bahwa *shape* merupakan properti intrinsik utama untuk sistem visual manusia. Pada umumnya citra yang dibentuk oleh mata merupakan citra *dwimatra* (dua dimensi), sedangkan objek yang dilihat umumnya berbentuk *trimatra* (tiga

dimensi). Informasi bentuk objek dapat diekstraksi dari citra pada permulaan pra-pengolahan dan segmentasi citra.

6. Tekstur (*texture*)

   Tekstur diartikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan piksel-piksel yang bertetangga. Jadi tekstur tidak dapat didefinisikan untuk sebuah piksel. Sistem visual manusia menerima informasi citra sebagai suatu kesatuan. Resolusi citra yang diamati ditentukan oleh skala dimana tekstur tersebut dipersepsi.

7. Waktu dan Pergerakan

   Respon suatu sistem visual tidak hanya berlaku pada faktor ruang, tetapi juga pada faktor waktu. Sebagai contoh, bila citra diam ditampilkan secara cepat, akan berkesan melihat citra yang bergerak.

8. Deteksi dan Pengenalan

   Dalam mendeteksi dan mengenali suatu citra, ternyata tidak hanya sistem visual manusia saja yang bekerja, tetapi juga ikut melibatkan ingatan dan daya pikir manusia.

2.14 Format File Citra

Sebuah format *file* citra harus dapat menyatukan kualitas citra, ukuran file dan kompabilitas dengan berbagai aplikasi. Format *file* citra standar yang digunakan saat ini terdiri dari beberapa jenis. Format- format ini digunakan untuk menyimpan citra dalam sebuah *file*. Setiap format memiliki karakteristik masing-masing. Ini adalah contoh format umum, yaitu : *Bitmap (.bmp), tagged image format (.tif, .tiff), Portable Network Graphics (.png), JPEG (.jpg),* dll.

Citra tertentu dapat disimpan dengan baik (dalam arti ukuran *file* lebih kecil dan kualitas gambar tidak berubah) pada format *file* citra tertentu, karena jika disimpan pada format lain, maka terkadang dapat menyebabkan ukuran *file* menjadi lebih besar dari aslinya dan kualitas citra dapat menurun. Oleh karena itu, untuk menyimpan suatu citra harus diperhatikan citra dan format *file* citra apa yang sesuai. Misalnya format citra GIF sangat tidak cocok untuk citra fotografi karena biasanya citra fotografi kaya akan warna, sedangkan format GIF hanya mendukung sejumlah warna sebanyak 256 (8 bit) saja. Format JPEG merupakan

pilihan yang tepat untuk citra–citra fotografi karena JPEG sangat cocok untuk citra dengan perubahan warna yang halus.

Pada format *bitmap*, citra disimpan sebagai suatu matriks dimana masing–masing elemennya digunakan untuk menyimpan informasi warna untuk setiap *pixel*. Jumlah warna yang dapat disimpan ditentukan dengan satuan *bit*-per-*pixel*. Semakin besar ukuran *bit*-per-*pixel* dari suatu *bitmap*, semakin banyak pula jumlah warna yang dapat disimpan. Format *bitmap* ini cocok digunakan untuk menyimpan citra digital yang memiliki banyak variasi dalam bentuknya maupun warnanya, seperti foto, lukisan, dan *frame video*.

Format *file* citra standar yang digunakan saat ini terdiri dari beberapa jenis. Format-format ini digunakan dalam menyimpan citra dalam sebuah *file*. Setiap format memiliki karakteristik masing-masing:

1. *Bitmap* (.bmp)

   Format .bmp adalah format penyimpanan standar tanpa kompresi yang umum dapat digunakan untuk menyimpan citra biner hingga citra warna. Format ini terdiri dari beberapa jenis yang setiap jenisnya ditentukan dengan jumlah bit yang digunakan untuk menyimpan sebuah nilai *pixel*.

2. *Tagged Image Format* (.tif, .tiff)

   Format .tif merupahkan format penyimpanan citra yang dapat digunakan untuk menyimpan citra *bitmap* hingga citra dengan warna palet terkompresi. Format ini dapat digunakan untuk menyimpan citra yang tidak terkompresi dan juga citra terkompresi.

3. *Portable Network Graphics* (.png)

   Format .png adalah format penyimpanan citra terkompresi. Format ini dapat digunakan pada citra *grayscale*, citra dengan palet warna, dan juga citra *fullcolor*. Format .png juga mampu menyimpan informasi hingga kanal alpha dengan penyimpanan sebesar 1 hingga 16 bit per kanal.

4. *Joint Photographic Expert Group* (.jpg)

   Format jpg adalah format yang sangat umum digunakan saat ini khususnya untuk transmisi citra. Format ini digunakan untuk menyimpan citra hasil kompresi dengan metode JPEG.

5. *Graphics Interchange Format* (.gif)

Format ini dapat diigunakan pada citra warna dengan palet 8 bit. Penggunaan umumnya pada aplikasi web. Kualitas yang rendah menyebabkan format ini tidak terlalu populer dikalangan peneliti pengolahan citra digital.

6. RGB (.rgb)

Format ini merupahkan format penyimpanan citra yang dibuat oleh *silicon graphics* untuk menyimpan citra berwarna.

7. RAS (.ras)

Format .ras diigunakan untuk menyimpan citra dengan format RGB tanpa kompresi.

Sedangkan pada format *file* citra vektor merupakan citra vektor yang dihasilkan dari perhitungan matematis dan tidak terdapat piksel, yaitu data yang tersimpan dalam bentuk vektor posisi, dimana yang tersimpan hanya informasi vektor posisi dengan bentuk sebuah fungsi. Pada citra vektor, mengubah warna lebih sulit dilakukan, tetapi membentuk objek dengan cara mengubah nilai lebih mudah. Oleh karena itu, bila citra diperbesar atau diperkecil, kualitas citra relatif tetap baik dan tidak berubah. Citra vektor biasanya dibuat menggunakan aplikasi-aplikasi citra vektor seperti CorelDRAW, Adobe Illustrator, Macromedia Freehand, Autocad, dll.

2.14.1 Grafis portabel jaringan (*Portable Network Graphics* (.png))

Format *file* ini berfungsi sebagai alternatif lain dari format *file* GIF. Format *file* ini digunakan untuk menampilkan objek dalam halaman web. Kelebihan dari format *file* ini dibandingkan dengan GIF adalah kemampuannya menyimpan *file* dalam bit depth hingga 24 bit serta mampu menghasilkan latar belakang (*background*) yang transparan dengan pinggiran yang halus. Format *file* ini mampu menyimpan alpha *channel*.

Format PNG (*Portable Network Graphic*), Adapun beberapa keuntungan dan kekurangan dari format tersebut adalah:

Keuntungan:

- PNG mendukung Kompresi *lossless* tingkat tinggi.
- PNG mendukung transparansi *alpha channel*.
- PNG mendukung Koreksi gamma.

- PNG mendukung *interlace.*
- PNG didukung oleh browser Web yang lebih baru.

Kekurangan:

- Browser yang lebih tua dan program mungkin tidak mendukung PNG file.
- Sebagai format berkas Internet, PNG menyediakan kurang kompresi daripada kompresi *lossy* JPEG.
- Sebagai format berkas Internet, PNG menawarkan tidak ada dukungan untuk multi gambar file atau file animasi. Mendukung GIF format berkas multi gambar dan animasi.

2.15 Kompresi Citra

2.15.1 Resolusi

Ada dua jenis resolusi yang perlu diketahui, yaitu resolusi spasial dan resolusi kecemerlangan.

1. Resolusi spasial

   Resolusi spasial adalah ukuran halus atau kasarnya pembagian kisi-kisi baris dan kolom pada saat dilakukan smpling. Resolusi spasial dipakai untuk menentukan jumlah piksel per satuan panjang. Biasanya satuan resolusi ini adalah dpi (*dot per inch*). Resolusi ini sangat berpengaruh pada detail citra dan perhitungan gambarnya. Sebagai contoh, citra dengan resolusi 50 dpi, artinya 1 (satu) *inch* mempunyai 50 piksel dan bila luas citra 1 $inch^2$ berarti citra tersebut mempunyai jumlah piksel 50X50 piksel. Bila ukuran citra diperbesar menjadi 10x10 $inch^2$ maka jumlah pikselnya tetap 50x50, tetapi resolusinya berubah menjadi 50:10= 5 dpi. Artinya 1 *inch* hanya diisi 5 piksel. Hal ini mengakibatkan gambar menjadi kabur, pecah-pecah, dan kasar.

2. Resolusi kecemerlangan

   Resolusi kecemerlangan (intensitas atau *brightness*) atau biasanya disebut dengan kedalaman bit atau kedalaman warna (*Bit Depth*) adalah ukuran halus kasarnya pembagian tingkat gradasi warna saat dilakukan kuantisasi. *Bit Depth* menentukan berapa banyak informasi warna yang tersedia untuk ditampilkan dalam setiap piksel. Semakin besar nilanya, semakin bagus

kualitas gambar yang dihasilkan dan tentu ukuran juga semakin besar. Resolusi citra merupakan tingkat detailnya suatu citra. Semakin tinggi resolusinya semakin tinggi pula tingkat detail dari citra tersebut.

2.16 Peningkatan Mutu Citra

Peningkatan mutu citra dilakukan untuk memperoleh keindahan citra yang akan digunakan untuk kepentingan analisis citra. Teknik penajaman (peningkatan) meliputi semua operasi yang menghasilkan citra baru dengan kenampakan visual dan karaktersitik spektal yang berbeda. Proses peningkatan tidak menaikan sistem informasi yang ada pada citra itu, tetapi menaikan kisaran dinamis pada objek yang dipilih, sehingga dapat dideteksi dengan mudah. Peningkatan citra meliputi manipulasi aras keabuan, dan kontras, penurunan derau, peningkatan kejelasan, dan penajaman tepi, penapisan, perbesaran, pewarnaan buatan, dan lain-lain.

Kesulitan terbesar dalam peningkatan citra mengukur kriteria peningkatan. Oleh karena itu sejumlah peningkatan citra membutuhkan prosedur interaktif untuk memperoleh hasil yang memuaskan. Teknik peningkatan citra merupakan pengolahan awal (*Prepocessing*) dalam pengolahan citra. Operasi pengolahan yang paling sederhana adalah operasi titik, pengolahan titik adalah operasi secara terpisah setiap titik, yaitu operasi secara terpisah setiap nilai kecerahan piksel masukan untuk dipetakan kedalam nilai baru sesuai dengan pada transformasi tertentu.

Operasi spasial mengolah sekelompok piksel masukan disekitar piksel pusat. Piksel-piksel menyediakan informasi bernilai kecenderungan (trend) kecerahan dalam daerah yang akan diolah. Penggunaan trend kecerahan ini mengatakan kedalam hal penapisan spasial.

Teknik peningkatan citra merupakan pengolahan awal dalam pengolahan citra. Selanjutnya diadakan segmentasi citra menjadi komponen-komponennya. Segmentasi citra digunakan untuk memecah suatu citra kedalam beberapa bagian kecil (segmen-segmen) berdasarkan suatu kriteria tertentu.

Peningkatan kualitas citra dibagi dalam dua kategori, yaitu metode domain spasial (ruang atau waktu) dan metode domain frekuensi. Teknik pemrosesan metode domain spasial adalah berdasarkan manipulasi langsung dari piksel

didalam citra. Sedangkan teknik pemrosesan metode domain frekuensi adalah berdasarkan perubahan *transformasi fourier* pada citra.

2.17 Konsep Deteksi Tepi

Tepi adalah perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat. Perbedaan intensitas inilah yang memperlihatkan rincian pada gambar. Tepi dapat diorientasikan dengan suatu arah, dan arah ini berbeda-beda, tergantung pada perubahan intensitas. Deteksi tepi merupakan langkah pertama untuk melingkupi informasi didalam citra. Tepi mencirikan batas-batas objek dan karena itu tepi berguna untuk proses segmentasi dan identifikasi objek di dalam citra.

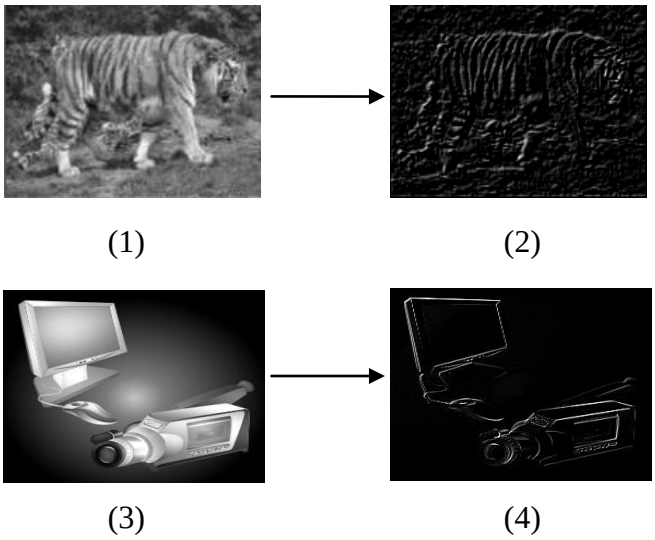
Suatu titik (*x*,*y*) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya.

2.17.1 Prinsip-Prinsip Deteksi Tepi

Deteksi tepi (*Edge Detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra, tujuannya adalah :

1. Untuk menandai bagian yang menjadi detail citra.
2. Untuk memperbaiki detail dari citra yang kabur, yang terjadi karena *error* atau adanya efek dari proses akuisisi citra.
3. Serta untuk mengubah citra 2D menjadi bentuk kurva.

Perhatikan hasil deteksi dari beberapa citra:



(1) (2)

(3) (4)

Gambar 2.14 Hasil Beberapa Deteksi Tepi

Pada gambar 2.14 terlihat bahwa gambar (1) dan (3) adalah gambar yang telah melalui proses pengambangan atau *threshold* dan gambar (2) dan (4) adalah gambar hasil deteksi tepi. Hasil deteksi tepi berupa tepi-tepi dari suatu gambar. Bila diperhatikan bahwa tepi suatu gambar terletak pada titik-titik yang memiliki perbedaan tinggi. Berdasarkan prinsip-prinsip filter pada citra maka tepi suatu gambar dapat diperoleh menggunakan High Pass Filter (HPF), yang mempunyai karakteristik:

$$\sum_y \sum_x H(x,y) = 0 \text{……..……...……(2.6)}$$

Contoh:

Diketahui fungsi citra $f(x,y)$ sebagai berikut:

1	1	1	1	1
1	1	1	1	0
1	1	1	0	0
1	1	0	0	0
1	0	0	0	0

Gambar 2.15 Matriks yang berisikan nilai piksel citra biner.

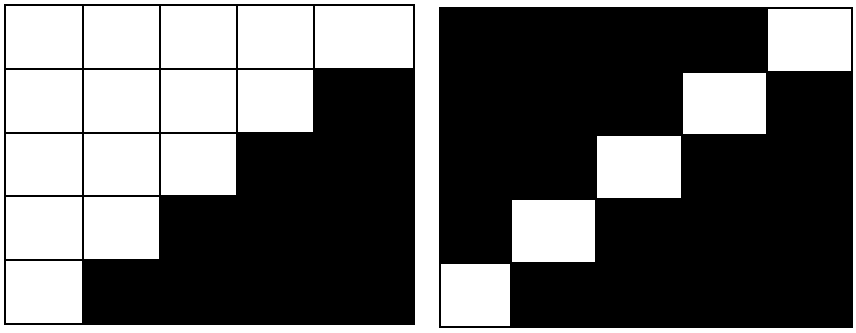
Dengan menggunakan filter : $H(x,y) = [-1 \;\; 1]$

Maka Hasil filter adalah :

0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
1	0	0	0	0

Gambar 2.16 Hasil filter dari citra biner

Bila digambarkan maka proses filter di atas mempunyai masukan dan keluaran sebagai berikut:



(a) (b)

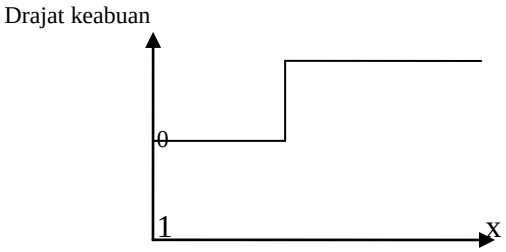
Gambar 2.17 (a). Citra masukan yaitu citra biner berupa bit-bit 0 dan 1, yaitu 0 = hitam, dan 1 = putih dan (b). Hasil filter dari citra biner.

2.17.2 Macam-Macam Tepi dalam Citra Digital

a. Tepi curam, arah berkisar 90^0.
b. Tepi landai atau lebar, sudut arah kecil. Dianggap terdiri dari sejumlah tepi-tepi lokal yang lokasinya berdekatan.
c. Tepi yang mengandung *noise*. Perlu dilakukan *image enhancement*.

a. Tepi Curam

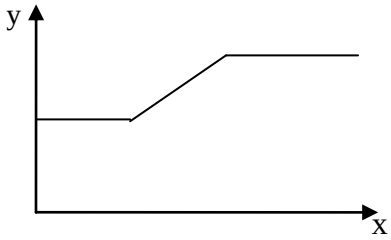
Tepi curam adalah tepi dengan perubahan intensitas yang tajam. Arah tepi berkisar 90^0. Gambar 2.18 menunjukan deteksi tepi tipe curam.



Gambar 2.18 Menunjukan Kurva untuk Deteksi Tepi Curam.

b. Tepi Landai

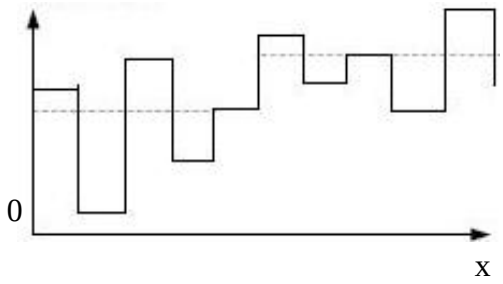
Tepi landai yaitu tepi dengan sudut arah yang kecil. Tepi landai dapat dianggap terdiri dari sejumlah tepi-tepi lokal yang lokasinya berdekatan. Gambar 2.19 menunjukan deteksi tepi tipe tepi landai.



Gambar 2.19 Menunjukan Kurva untuk Deteksi Tepi Landai.

c. Tepi yang Mengandung Derau

Umumnya tepi yang terdapat pada aplikasi visi komputer mengandung derau. Operasi peningkatan kualitas citra dapat dilakukan terlebih dahulu sebelum pendektesian tepi. Gambar 2.20 menunjukan deteksi tepi tipe curam dengan derau.



Gambar 2.20 Menunjukan Kurva untuk Deteksi Tepi Derau

2.18 Operator dalam Mendeteksi Tepi Suatu Citra

2.18.1 Operator *Roberts*

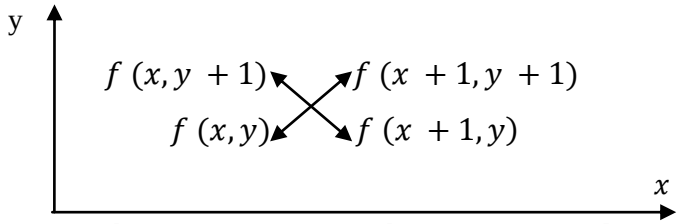
Operator *Roberts* sering disebut juga operator silang. Gradien *Roberts* dalam arah-x dan arah-y dihitung dengan persamaan berikut dan ditunjukkan pada gambar 2.21:

$$R_x(x,y) = f(x+1, y+1) - f(x,y) \quad \text{.....(2.7)}$$

$$R_y(x,y) = f(x, y+1) - f(x+1, y) \quad \text{.....(2.8)}$$

Bila ditulis dalam komponen gradien menjadi:

$$R = |R_x| + |R_y| \quad \text{.....(2.9)}$$



Gambar 2.21 Operator *Roberts* (operator silang)

Dimana R_x dan R_y dihitung menggunakan *mask* konvolusi, operator *Roberts* adalah:

$$R_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad R_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

1	0
0	-1

0	1
-1	0

Titik ini berada diantara 4 piksel dalam ukuran 2x2 piksel yang berdekatan atau dipusat kernel, yaitu koordinat hasil interpolasi $\left(\frac{x+1}{2}, \frac{y+1}{2}\right)$.

Contoh perhitungan operator *Roberts*:

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

(i) citra semula (ii) hasil konvolusi

S_x=(1)(3)+(0)(2)+(0)(4)+(-1)(1) = 2

S_y=(0)(3)+(1)(4)+(-1)(2)+(0)(1) = 2

$S_x + S_y = 4$.

2.18.2 Operator *Sobel*

Operator *sobel* adalah salah satu operator yang menghindari adanya perhitungan gradien dititik interpolasi. Operator ini menggunakan kernel ukuran 3x3 piksel untuk perhitungan gradien sehingga perkiraan gradien berada tepat ditengah jendela.

Misal, suatu pengaturan piksel di sekitar piksel (*x*,*y*):

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x,y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

a_0	a_1	a_2
a_7	(x,y)	a_3
a_6	a_5	a_4

Operator *Sobel* adalah magnitude dari gradien yang dihitung dengan:

$$M = \sqrt{{S_x}^2 + {S_y}^2} \quad \text{……...….(2.10)}$$

Dimana *M* adalah besaran gradien dititik tengah kernel dan turunan parsial dihitung menggunakan persamaan berikut:

$$S_x = (a_2 + ca_{3+}a_4) - (a_0 + ca_{7+}a_6) \text{…….……...…..(2.11)}$$

$$S_y = (a_0 + ca_{1+}a_2) - (a_6 + ca_{5+}a_4) \text{…….....…..(2.12)}$$

Dengan konstanta c adalah 2, dalam bentuk kedok (*mask*), *Sx* dan *Sy* dapat dinyatakan sebagai:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} S_y = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 2 \\ -1 & -2 & -1 \end{bmatrix}$$

Arah tepi dihitung dengan persamaan:

$$\alpha(x,y) = tan^{-1}\left(\frac{S_y}{S_x}\right) \text{……...……...…..(2.13)}$$

Contoh perhitungan operator *Sobel:*

$$\begin{bmatrix} 3 & 4 & 2 & 5 & 1 \\ 2 & 1 & 6 & 4 & 2 \\ 3 & 5 & 7 & 1 & 3 \\ 4 & 2 & 5 & 7 & 1 \\ 2 & 5 & 1 & 3 & 2 \end{bmatrix} \qquad \begin{bmatrix} * & * & * & * & * \\ * & 18 & & & \\ & & & & \\ & & & & \\ & & & & \end{bmatrix}$$

(i) citra semula (ii) hasil konvolusi

Nilai 18 pada citra hasil konvolusi diperoleh dengan perhitungan berikut

$S_x = (3)(-1) + (2)(-2) + (3)(-1) + (2)(1) + (6)(2) + (7)(1) = 11$

$S_y = (3)(1) + (4)(2) + (2)(1) + (3)(-1) + (5)(-2) + (7)(-1) = -7$

$$M = \sqrt{s_x^2 + s_y^2} = \sqrt{11^2 + (-7)^2} \cong |S_x| + |S_y| = |11| + |-7| = 18$$

2.18.3 Operator *Prewitt*

Operator ini menggunakan persamaan yang sama dengan operator *sobel*, hanya saja konstanta c yang digunakan bernilai 1 (c=1) sehingga bentuk kernel dari operator *prewitt* adalah:

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad P_y = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

-1	0	1
-1	0	1
-1	0	1

1	0	1
0	0	0
-1	-1	-1

Berbeda dengan operator *sobel*, operator *prewitt* tidak menekankan pembobotan pada piksel-piksel yang lebih dekat dengan titik pusat kernel. Untuk contoh perhitungan operator *prewitt* sama seperti operator *sobel* hanya konstanta yang digunakan adalah c = 1.

2.19 Pengenalan Pola (*Pattern Recognition*)

Menurut Fairhurst (1998), pengenalan pola merupakan bidang yang berhubungan dengan proses identifikasi objek pada citra atau interpretasi citra. Proses ini bertujuan untuk mengekstrak informasi yang disampaikan oleh citra. Pengenalan pola mempunyai arti bidang studi yang melakukan proses analisis citra yang bentuk masukannya adalah citra itu sendiri atau dapat juga berupa citra digital dan bentuk keluarannya adalah suatu deskripsi. Tujuan pengenalan pola adalah untuk meniru kemampuan manusia dalam mengenali objek tertentu.

Pengenalan pola merupakan proses pengenalan suatu objek dengan menggunakan berbagai metode. Teknik pencocokan pola adalah salah satu teknik dalam pengolahan citra digital yang berfungsi untuk mencocokkan tiap-tiap bagian dari suatu citra dengan citra yang menjadi acuan (*template*). Metode pencocokan pola adalah salah satu metode terapan dari teknik konvolusi. Teknik konvolusi pada penelitian ini dilakukan dengan mengkombinasikan citra daun masukan dengan citra daun sumber acuan, hingga akan didapatkan nilai koefisien korelasi yang besarnya antara -1 dan +1. Saat nilai koefisien korelasi semakin mendekati +1, bisa dikatakan citra masukan semakin sama (mirip) dengan citra acuannya. Rumus yang digunakan adalah:

$$r = \frac{\sum_{i=o}^{M-1}\sum_{j=0}^{N-1}(x_{ij}-\bar{x})(y_{ij}-\bar{y})}{\sqrt{\sum_{i=o}^{M-1}\sum_{j=0}^{N-1}(x_{ij}-\bar{x})^2.\sum_{i=o}^{M-1}\sum_{j=0}^{N-1}(y_{ij}-\bar{y})^2}} \text{.....(2.14)}$$

Keterangan :

r : Koefisien korelasi

x : Citra acuan (*template*)

$\bar{x}$: Nilai rata-rata citra acuan$\bar{x}$

y : Nilai rata-rata citra masukan

$\bar{y}$: Citra masukan

M, N : Jumlah piksel citra

2.19.1 *Template Matching*

Menurut Roland Miezianko, definisi *Template Matching* adalah :

1). Teknik yang digunakan untuk mengklasifikasikan objek.

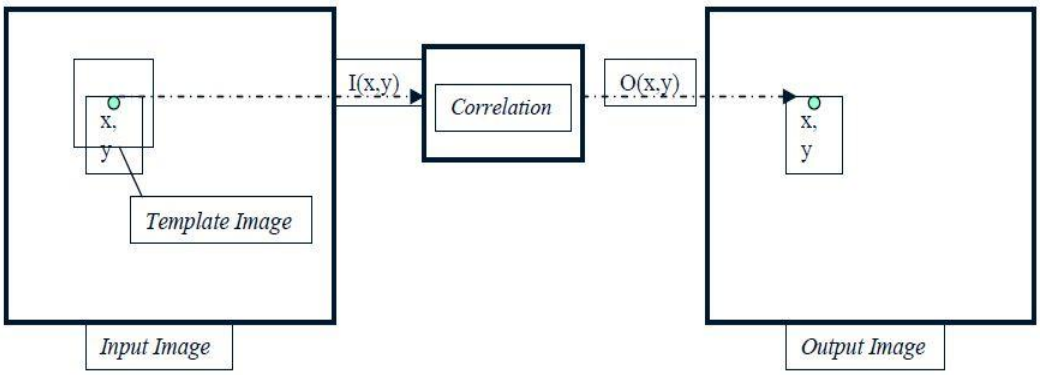
2). Teknik *Template Matching* membandingkan potongan-potongan citra yang satu dengan lainnya.

3). Citra sampel dapat digunakan untuk mengenali objek yang serupa didalam *source image*.

4). Jika standard deviasi dari *template image* yang dibandingkan dengan *source image* cukup kecil, maka metode *template matching* dapat digunakan.

5). *Template* sering digunakan untuk mengenali *printed characters*, angka dan objek sederhana lainnya.

Template matching merupakan teknik pengenalan pola yang paling sederhana. Pola diidentifikasi dengan cara membandingkan pola masukkan (input *pattern*) dengan daftar representasi pola yang telah disimpan. Representasi pola yang disimpan tersebut disebut *template*.



Gambar 2.22 Metode *Template Matching*

Proses *matching* memindahkan *template image* ke tempat yang memungkinkan didalam *source image*, lalu menghitung index yang megindikasikan seberapa besar kecocokan *template* itu dalam posisi tersebut. Proses *matching* dilakukan berdasarkan *pixel-by-pixel.*

2.20 *Matriks Laboratory* (MatLab)

Matlab merupakan bahasa pemrograman yang hadir dengan fungsi dan karakteristik yang berbeda dengan bahasa pemrograman lain yang sudah ada lebih dahulu seperti Delphi, Basic, maupun C++. Matlab merupakan bahasa pemrograman level tinggi yang dikhususkan untuk kebutuhan komputasi teknis, visualisasi dan pemrograman seperti komputasi matematik, analisis data, pengembangan algoritma, simulasi dan pemodelan dan grafik-grafik perhitungan.

Matlab merupakan suatu sisem interaktif yang memiliki elemen data dalam suatu array sehingga tidak lagi kita dipusingkan dengan masalah dimensi. Hal ini memungkinkan kita untuk memecahkan banyak masalah teknis yang terkait dengan komputasi, khususnya yang berhubungan dengan matrix dan formulasi vektor, yang mana masalah tersebut akan sulit dilaukan bila menggunakan bahasa level rendah seperti Pascall, C dan Basic.

Matlab hadir dengan membawa warna yang berbeda. Hal ini karena matlab membawa keistimewaan dalam fungsi-fungsi matematika, fisika, statistik, dan visualisasi. Nama Matlab merupakan singkatan dari *matrix laboratory*. Matlab pada awalnya dikembangkan oleh MathWorks untuk memberikan kemudahan mengakses data matrik pada proyek LINPACK dan EISPACK. Saat ini matlab memiliki ratusan fungsi yang kompleks dari berbagai disiplin ilmu.



Gambar 2.23 Logo Matlab

Ada perbedaan penggunaan Matlab di dalam *platform* yang berbeda:

A. **Macintosh**

Terdapat sebuah built-in editor untuk *m-files*. *File* harus di *save* ke dalam format text, dan dapat diload bila akan mulai menggunakan Matlab.

B. **Windows**

Menggunakan Matlab dalam Windows serupa dengan cara menggunakan di dalam Macintosh. Akan tetapi, perlu diketahui bahwa *m-file* akan disimpan di dalam *elipboard*. *M-file* perlu disimpan dalam bentuk format *namafile.m*.

C. Unix

Di dalam Unix, editor dijalankan secara terpisah dari Matlab. Cara terbaik adalah dengan membuat sebuah direktori untuk semua *m-files*, kemudian cd ke direktori yang dimaksud sebelum menjalankan Matlab maupun editor. Untuk mulai menggunakan Matlab dari *window.Xterm* hanya perlu dengan mengetikkan "Matlab".

Dalam pembuatan aplikasi disini hanya menggunakan Matlab *for windows*, karena bahasa pemrograman yang digunakan adalah bahasa C, C++ dan sebagainya agar dapat mudah mempelajari dan memahami dari semua *toolbox* Matlab.

2.20.1 Lingkungan Kerja Matlab

Matlab adalah program interaktif untuk komputasi numerik dan visualisasi data yang umum digunakan oleh ilmuan kendali untuk analisa dan perancangan.

Ada berbagai *toolbox* yang berbeda yang dapat digunakan untuk berbagai area aplikasi yang berbeda. Sebagai sebuah sistem, Matlab tersusun dari 5 bagian utama, yaitu:

1. **Development Environment.**

   Merupakan sekumpulan perangkat dan fasilitas untuk fungsi-fungsi dan *file-file* Matlab. Beberapa perangkat ini merupakan sebuah *graphical user interface* (GUI). Termasuk didalamnya adalah Matlab desktop dan *Command Window*, *Command history*, sebuah editor dan *debugger*, dan *browsers* untuk melihat *help*, *workspace*, files, dan *search path*.

2. **Matlab *Mathematical Function Library*.**

   Merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi yang lebih komplek seperti: sum, sin, cos, dan *complex arithmetic*, sampai dengan fungsi-fungsi yang lebih komplek seperti *matrix invers, matrix eigenvalues, bessel function*, dan *fast fourier transform*.

3. **Matlab *Laguage*.**

   Merupakan suatu high-level matrix atau *array laguage* dengan *control flow statements, functions, data structures, input/output,* dan fitur-fitur *object-oriented programming*. Ini memungkinkan bagi kita untuk melakukan kedua hal baik pemrograman dalam lingkup sederhana untuk mendapatkan hasil yang cepat dan pemrograman dalam lingkup yang lebih besar untuk memperoleh hasil-hasil dan aplikasi yang komplek.

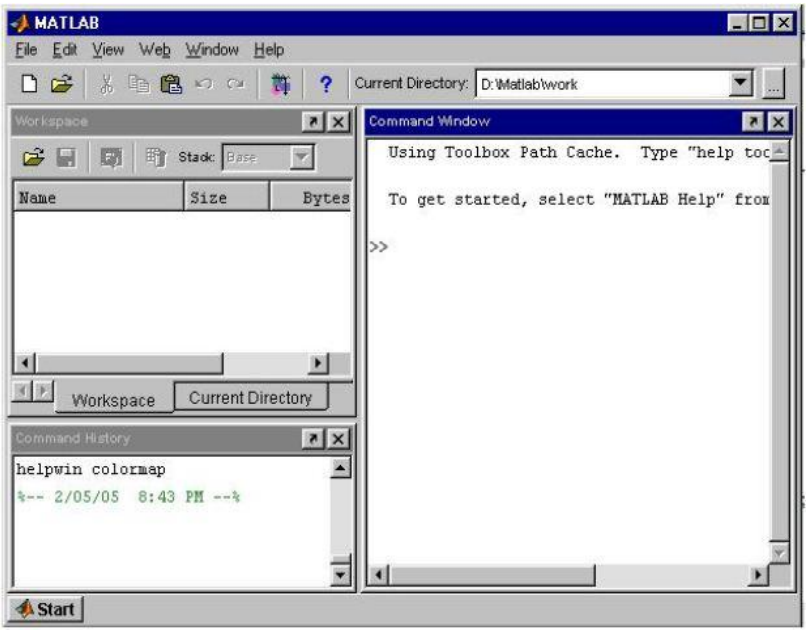
4. ***Graphics*.**

   Matlab memiliki fasilitas untuk menampilkan *vector* dan *matrices* sebagai suatu grafik. Didalamnya melibatkan *high-level functions* (fungsi-fungsi level tinggi) untuk visualisasi data dua dimensi dan data tiga dimensi, *image processing*, *animation*, dan *presentation graphics*. Ini juga melibatkan fungsi level rendah yang memungkinkan bagi pengguna untuk membiasakan diri untuk memunculkan grafik mulai dari bentuk yang sederhana sampai yang tingkat *graphical user interface* pada aplikasi Matlab.

5. **Matlab *Application Program Interface* (API).**

   Merupakan suatu *library* yang memungkinkan program yang telah ditulis dalam bahasa C dan fortran mampu berinteraksi dengan Matlab. Ini

melibatkan fasilitas untuk memanggil *routines* dari Matlab (*dynamic linking*), pemanggilan Matlab sebagai sebuah *computational engine*, dan untuk membaca dan menuliskan *MAT-files*.



Gambar 2.24 Jendela utama Matlab

Beberapa bagian dari *Window* Matlab adalah:

A. Current Directory

Window ini menampilkan isi dari direktori kerja saat menggunakan Matlab. Kita dapat mengganti direktori ini sesuai dengan tempat direktori kerja yang diinginkan. *Default* dari alamat direktori berada dalam *folder works* tempat *program files* Matlab berada.

B. Command History

Window ini berfungsi untuk menyimpan perintah-perintah apa saja yang sebelumnya dilakukan oleh pengguna terhadap Matlab.

C. Command Window

Window ini adalah *window* utama dari Matlab. Disini adalah tempat untuk menjalankan fungsi, mendeklarasikan *variable*, menjalankan proses-proses, serta melihat isi *variable*.

D. Workspace

Workspace berfungsi untuk menampilkan seluruh *variable-variable* yang sedang aktif pada saat pemakaian Matlab. Apabila *variable* berupa data

matriks berukuran besar, maka user dapat melihat isi dari seluruh data dengan melakukan *double* klik pada variable tersebut. Matlab secara otomatis akan menampilkan *window* "*array editor*" yang berisikan data pada setiap variable yang dipilih user.