BAB II

LANDASAN TEORI

2.1 Citra

Citra merupakan suatu fungsi dari intensitas cahaya dalam bidang dua dimensi yang dinotasikan dengan *f(x,y)*, dimana nilai *x* dan *y* menyatakan koordinat citra dan nilai *f* pada koordinat *(x,y)* menyatakan kecerahan atau informasi warna citra. Secara matematis persamaan untuk fungsi intensitas *f(x,y)* adalah :

$$0 < f(x,y) < \infty \qquad (2.1)$$

Pada hakekatnya citra yang dilihat oleh mata kita terdiri dari berkas-berkas cahaya yang dipantulkan oleh benda-benda di sekitar kita. Jadi fungsi intensitas *f(x,y)* merupakan fungsi sumber cahaya *i(x,y)* yang menerangi objek serta jumlah cahaya yang dipantulkan oleh objek *r(x,y)*. Dengan demikian fungsi *f(x,y)* dapat dinyatakan dengan persamaan:

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

dengan :

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

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

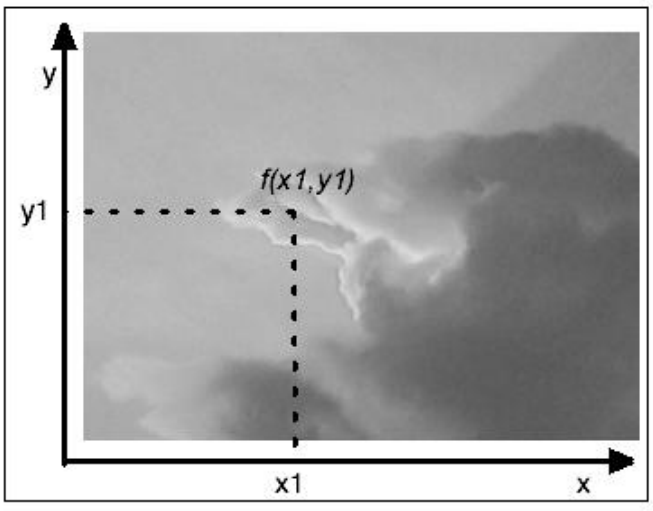
Semakin terang sumber cahaya, maka nilai iluminasi sumber semakin besar, dan semakin terang warna suatu objek maka koefisien pantul objek tersebut semakin besar. Pengolahan citra digital secara umum dapat didefinisikan sebagai

pemrosesan sebuah gambar dua dimensi dengan menggunakan komputer untuk menghasilkan suatu citra digital lain yang sesuai dengan keinginan kita.

Pengolahan citra mempunyai aplikasi yang sangat luas dalam berbagai bidang, antara lain di bidang militer, kedokteran, biologi, komunikasi data, hiburan, robotika, hukum, dan lain-lain.

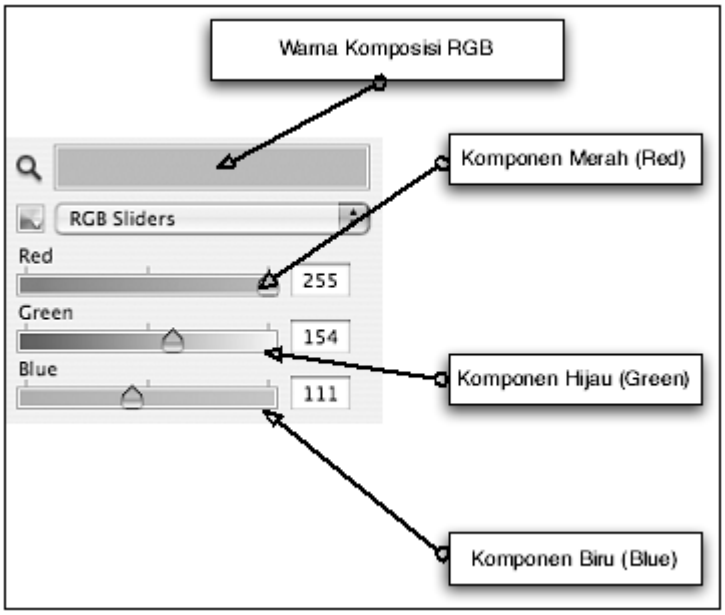
2.2 Pengolahan Citra Digital

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 2.1. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (Red, Green, Blue - RGB). Komposisi warna
RGB tersebut dapat dijelaskan pada 2.2.



Gambar 2.1. Citra Digital

Sebuah citra diubah ke bentuk digital agar dapat disimpan dalam memori komputer atau media lain. Proses mengubah citra ke bentuk digital bisa dilakukan dengan beberapa perangkat, misalnya scanner, kamera digital, dan handycam. Ketika sebuah citra sudah diubah ke dalam bentuk digital (selanjutnya disebut citra digital), bermacam-macam proses pengolahan citra dapat diperlakukan terhadap citra tersebut



Gambar 2.2. Komposisi Warna RGB

Pengolahan citra digital dapat dilakukan dengan cara-cara sebagai berikut :

1. Representasi dan permodelan citra
2. Peningkatan kualitas citra
3. Restorasi citra
4. Analisis citra
5. Rekonstruksi citra

6. Kompresi citra

JPEG dikembangkan oleh Joint Photographic Expert Group pada akhir tahun 80an dan kemudian dikenal karena teknik kompresi-nya yang bagus. Kompresi JPEG berdasarkan pada Discrete Cosine Transform. Pada tahun 1997, komite JPEG memutuskan untuk mengembangkan standar baru untuk kompresi citra. Sejak saat itulah JPEG-2000 mulai dikembangkan.

Berikut ini adalah contoh format citra

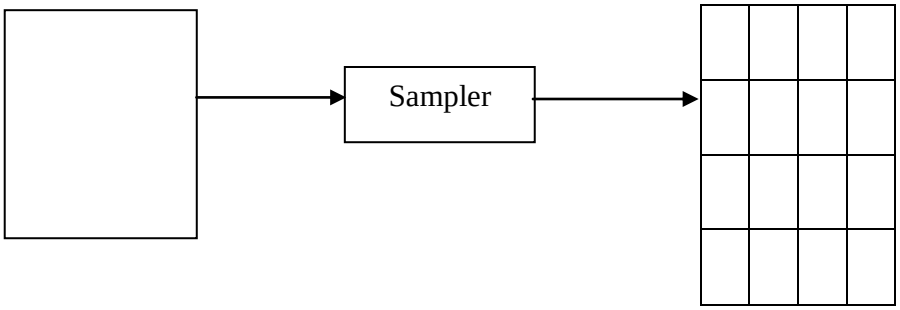
Tabel 2.1. Macam-macam Format Citra

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-	JPEG biasanya digunakan untuk foto
	perts Group	atau 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 Experts 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	Merupakan format citra hitam putih yang sederhana. PBM memerlukan 1 bit tiap pixel. Tidak seperti format citra

	Format	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 Representasi Citra Digital

Agar dapat diproses dengan menggunakan komputer, maka suatu citra harus direpresentasikan secara numerik dengan cara *digitalisasi*, yaitu merepresentasikan citra kontinu kedalam bentuk diskrit (digital), baik terhadap koordinat citra, maupun terhadap intensitasnya.



Gambar 2.3 Proses *Sampling*

Citra digital merupakan suatu matriks berukuran *M x N* dimana baris dan kolomnya merupakan koordinat titik pada citra dan elemen matiksnya merupakan nilai intensitas dari titik tersebut. Elemen-elemen dari array digital disebut sebagai *pixel*. Citra digital yang berupa matriks dengan ukuran *M x N* dapat digambarkan sebagai berikut:

$$f(x,y) = \begin{pmatrix} f(0,0) & f(0,1) & \ldots. & f(0,M-1) \\ f(1,0) & f(1,1) & \ldots. & f(1,M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \ldots. & f(N-1,M-1) \end{pmatrix}$$

Gambar 2.4 Citra *f(x,y)* berukuran *M x N*

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:

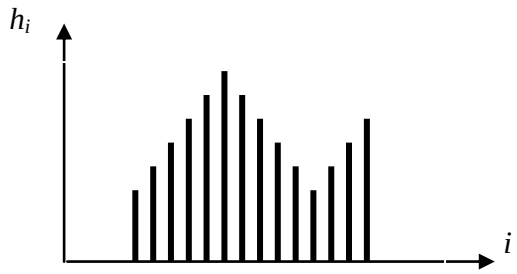
$$h_i = \frac{n_i}{n} \quad , \; i = 0,1,...,L-1 \tag{2.5}$$

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 2.5 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 Mean square error (MSE)

MSE adalah rata-rata kuadrat nilai kesalahan antara citra asli dengan citra hasil pengolahan yang secara matematis dapat dirumuskan sebagai berikut :

$$MSE = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left|\left(f(x,y)-g(x,y)\right)^2\right|$$

Keterangan :

M= panjang citra N=Lebar citra

F=Citra asli G=Citra penggabungan (x,y)= Titik Citra

2.6 Metode Penggabungan Image

2.6.1 Metode *pixel averaging*

Pixel averaging digunakan untuk mengurangi timbulnya suatu efek visualisasi pada hasil penyalinan. Berdasarkan input tersebut maka akan dilakukan perhitungan letak dari citra tertentu terhadap citra lainnya setelah mengetahui letak citra tertentu. Tujuan yang ingin dicapai dalam penelitian ini adalah diperolehnya cara untuk membentuk panorama 360^0 dengan menggabungkan sejumlah citra tunggal yang diperoleh dengan menggunakan kamera biasa. Berbeda dengan cara yang dilakukan oleh perangkat lunak QuickTime VR dimana digunakan alat khusus untuk memutar kamera dengan sudut yang tepat, cara yang kami gunakan di sini tidak memerlukan alat khusus dan posisi sudut pengambilan setiap citra tidak perlu diketahui. Lebih dari itu, posisi horisontal citra satu dengan yang lainnya pada pada saat pengambilan citra dengan kamera dimungkinkan untuk berbeda tanpa mengorbankan kualitas gambar terlalu banyak. Dalam percobaan yang telah dilakukan,setiap gambar panorama dibentuk dari 15-25 gambar tunggal yang memiliki wilayah tumpang-tindih sekitar 20-30%. Wilayah tumpang tindih yang semakin besar akan memberikan gambar panorama yang lebih baik namun memerlukan jumlah gambar tunggal yang lebih

banyak.Kelemahan yang kami temukan pada metoda yang sudah diterapkan pada [5] adalah di dalam transformasi geometrik yang dilakukan terhadap seluruh wilayah citra untuk menyamakan sudut dan posisi perspektif dua buah citra. Jika hal ini dilakukan untuk jumlah citra yang besar maka citra yang terakhir akan mengalami distorsi yang sangat besar bahkan mungkin mencapai keadaan degenerasi.Untuk mengatasi kelemahan ini, transformasi geometrik dilakukan hanya terhadap wilayah gambar yang tumpang tindih saja.

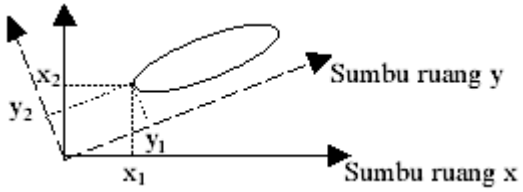
2.6.2 Metode pixel split

Algoritma clustering telah dikembangkan oleh J.J Simpson yakni split dan marge dan classification dengan mengganbungkan proses split dan marge. Nampak bahwa mekanisme split pada algoritma tersebut tidak mempertimbangkan lokasi tempat berkumpulnya mayoritas pixel. Namun hanya mempertimbangkan jarak terjauh antara pixel. Hal ini bisa mengakibatkan pemotongan cluster yang berada dikeduanya penyebabnya bisa berupa perbedaan distribusi atau ukuran cluster yang terlalu besar dengan emikian dibutuhkan metode split yang memperhatika distribusi pixel dalam feature space disribusi ini dapat digambarkan melaului histrogram. Dimana tiap kurva yang terbentuk dapat diasosiasikan sebagai sebuah cluster .

1. **Proses *Split***

Citra multispektral yang memiliki banyak dimensi itu,ditransformasikan dengan metode Principle ComponentTranformation (PCT). Proses ini sebenarnya adalah pemutaran sumbu sedemikian hingga variansi maksimum dicapai oleh

komponen utama. Sebagaimana Gambar 3.2,sumbu ruang x dapat digantikan oleh sumbu ruang y,dengan variansi maksimum pada komponen utama.



Gambar 2.6. Ilustrasi modifikasi koordinat

Dengan demikian array multidimensi tersebut dapat ditransformasikan menjadi berdimensi tunggal. Distribusi gray level array ini divisualisasikan oleh histogram yang

akan menampakkan sejumlah kurva. Tiap kurva ini dapat diasosiasikan sebagai satu cluster, yang telah melakukan hal yang sama namun dalam feature space 2 dimensi. Oleh karena perlu dicari interval batas tiap kurva, agar tiap pixel yang nilainya berada pada interval yang sama dapat digabungkan sebagai anggota suatu cluster yang sama

pula. Selanjutnya, untuk tiap cluster yang terbentuk, perlu dicari vektor mean yang menjadi prototype cluster tersebut. Caranya dengan menghitung rata-rata dari seluruh vektor pixel yang menjadi anggotanya. Hasil akhir tahap ini adalah n vektor yang mewakili n cluster.

2. Proses *Merge*

Diantara tiap *cluster* yang dihasilkan oleh proses di atas sangat mungkin terdapat pasangan *cluster* yang sangat mirip, sehingga layak untuk digabung.

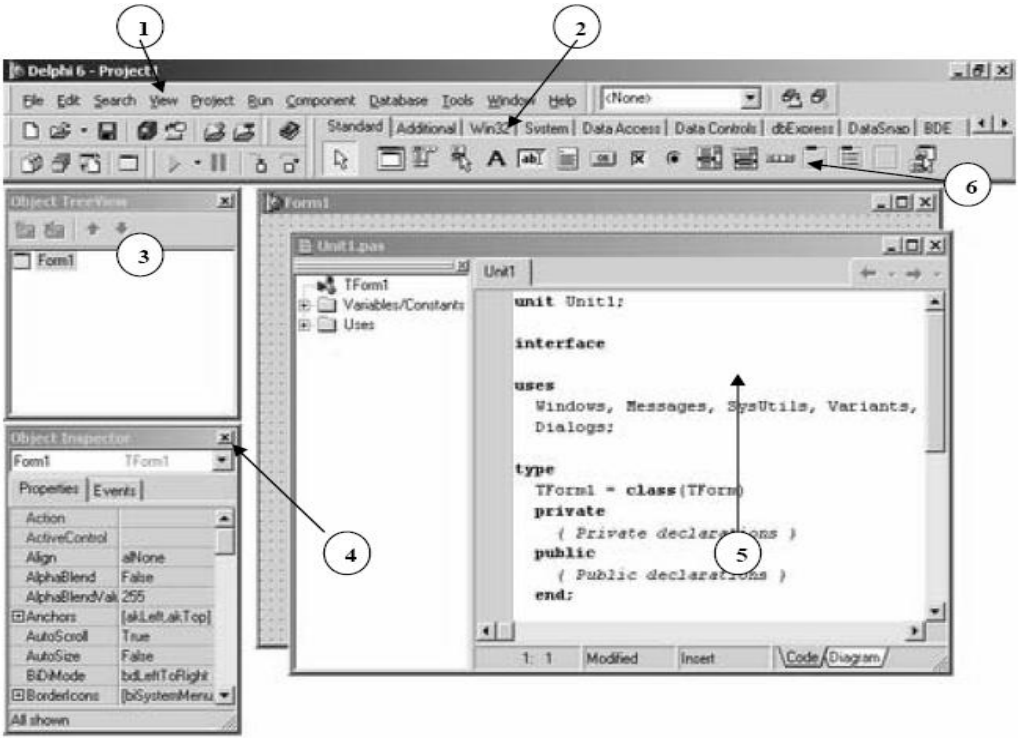
Penggabungan (*merge*) ini harus dilakukan secara ketat, agar tiap *cluster* benar-benar kompak. Artinya, seluruh *pixel* dalam sebuah *cluster* yang sama, harus memiliki kemiripan yang tinggi satu sama lain. Selain itu, perlu dihindari terjadinya *chain effect*, yakni kondisi tergabungnya dua *cluster* yang sebenarnya tidak mirip, sebagai akibat terdapatnya *noise* yang menghubungkan keduanya. Metode yang diharapkan dapat mengakomodasi kebutuhan tersebut adalah *complete link*. Sedangkan ukuran kemiripan dihitung berdasarkan jarak *euclidean*.

Pada tahap awal, perlu dibuat tabel jarak antar *cluster* untuk mencari pasangan *cluster* yang paling mirip. Pasangan yang jaraknya paling dekat inilah, yang harus digabung. Selanjutnya tabel jarak antar *cluster* tersebut dihitung ulang, berkaitan dengan berkurangnya sebuah *cluster* ini. Jarak suatu *cluster* terhadap *cluster* baru ini ditentukan oleh jarak terdekatnya terhadap *cluster* baru tersebut. Proses penggabungan dan pengevaluasian tabel ini diulang terus menerus, hingga jarak terkecil yang ada telah melampaui *threshold merge*. Hasil akhir proses *merge* ini adalah vektor mean baru hasil penggabungan.

2.7 Pemrograman Borland Delphi 6.0

Borland Delphi 6.0, untuk selanjutnya disebut Delphi, adalah paket bahasa pemrograman yang bekerja pada sistem operasi Windows. Delphi merupakan bahasa pemrograman yang mempunyai cakupan kemampuan yang luas dan sangat canggih. Berbagai aplikasi dapat dibuat dengan Delphi, termasuk aplikasi untuk mengolah teks, grafik, database, dan aplikasi web.

Untuk mempermudah programer dalam membuat program aplikasi, Delphi menyediakan fasilitas pemrograman yang lengkap. Fasilitas tersebut terbagi kedalam dua kelompok, yaitu *object* dan bahasa pemrograman. Secara ringkas, *object* adalah suatu komponen yang mempunyai bentuk fisik dan biasanya dapat dilihat (visual). Sedangkan bahasa pemrograman secara singkat dapat disebut sebagai sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu serta untuk menjalankan tugas tertentu. Delphi menggunakan struktur bahasa pemrograman *Object Pascal* yang sudah sangat dikenal dikalangan pemrogram professional. Gabungan dari *object* dan bahasa pemrograman ini sering disebut dengan bahasa pemrograman berorientasi *object* atau *Object Oriented Programming* (*OOP*).



Gambar 2.7 Interface Delphi 6.0

Keterangan :

1. Jendela utama Delphi. jendela ini berisi baris menu, *toolbar*, dan *component palette*.
2. *Component palette*, berisi *icon-icon* yang mewakili komponen tertentu.
3. Jendela *Object TreeView* yang berisi hirarki objek dengan *root* adalah *form* yang sedang aktif.
4. Jendela *Object Inspector* yang berisi tab *Property* dan *Event* dari objek yang sedang aktif (terpilih).
5. Jendela *Editor* yang digunakan untuk menulis kode program.
6. *Form* yang digunakan untuk mendesain tampilan program