BAB 2
TINJAUAN PUSTAKA

2.1 Pengolahan Citra

Pengolahan citra digital adalah salah satu cabang dari ilmu informatika yang berkutat untuk melakukan transformasi suatu citra atau gambar menjadi citra lain dengan menggunakan teknik tertentu[14]. Pengolahan citra banyak digunakan dalam berbagai bidang seperti bidang pendidikan, kedokteran, industry dll. Pengolahan citra digital merupakan salah satu disiplin ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar(peningkatan kontras, tranformasi, warna, restorasi citra), transformasi gambar(rotasi, translasi, skala, transformasi, geometrik), melakukan pemulihan citra ciri (*feature image*s) yang optimal untuk tujuan analisis, melakukan proses penarikan informasi deskripsi objek atau pengenalan objek yang terkandung pada citra, melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, input pengolahan citra adalah citra, sedangkan outputnya adalah citra hasil pengolahan.

2.2 Operasi Pengolahan Citra

Banyak operasi yang dilakukan dalam pengolahan citra[14], umumnya operasi pengolahan citra dapat diklasifikasikan dalam beberapa jenis sebagai berikut :

1. Perbaikan Kualitas Citra (*image enhancement*)

Operasi citra jenis ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra, ciri khusus yang terdapat didalam citra lebih ditonjolkan. Contoh-contoh operasi citra perbaikan citra :

1. Perbaikan kontras gelap/terang.
2. Perbaikan tepian objek (*edge enhancement*).
3. Penajaman (*sharpening*).
4. Penajaman warna semu (*sharpening*).
5. Penapisan derau (*noise filtering*).

2. Pemugaran Citra (*image restoration*)

Operasi citra ini bertujuan untuk menghilangkan atau meminimumkan cacat pada citra. Tujuan pemugaran citra hampir sama dengan operasi perbaikan citra, perbedaanya pada pemugaran citra penyebab degradasi gambar diketahui.

Contoh operasi pemugaran citra :

a. Penghilangan kesamaran (*debluring*).
b. Penghilangan derau (*noise*).

3. Pemampatan citra (*image compression*)

Jenis operasi ini bertujuan agar citra dipresentasikan dalam bentuk yang lebih kompak sehingga memerlukan memori yang lebih sedikit. Hal penting yang harus diperhatikan dalam pemampatan adalah citra yang telah dimampatkan harus tetap mempunyai kualitas gambar yang bagus.

4. Segmentasi Citra (*image segmentation*)

Operasi citra jenis ini bertujuan untuk memecah suatu citra kedalam beberapa segmen dengan suatu kriteria tertentu. Jenis operasi ini berkaitan erat dengan pengolahan pola.

5. Pengorakan Citra (*image analysis*)

Jenis operasi ini bertujuan menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsinya. Teknik pengorakan citra mengekstraksi ciri-ciri tertentu yang membantu dalam identifikasi objek. Proses segmentasi seringkali diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya.

Contoh-contoh operasi pengorakan citra :

a. Pendeteksian tepi objek (*edge detection*)
b. Ekstraksi batas (*boundary*)
c. Representasi daerah (*region*)

6. Rekonstruksi Citra (*image reconstruction*)

Operasi citra jenis ini bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Operasi rekonstruksi citra banyak digunakan dalam bidang medis misalnya foto rontgen dengan sinar X digunakan untuk membentuk ulang gambar organ tubuh.

7. Perubahan Model Warna

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang yang berbeda. Warna merah mempunyai panjang gelombang yang paling tinggi, sedangkan warna ungu mempunyai panjang gelombang rendah. Warna-warna yang diterima oleh mata adalah hasil kombinasi cahaya dengan panjang gelombang berbeda. Penelitian memperlihatkan kombinasi warna memberikan rentang warna yang paling lebar adalah merah(R), hijau(G), Biru(B).

a. Citra RGB

Disebut citra *true color* disimpan dalam citra berukuran (m x n) x 3 yang mendefinisikan warna merah, hijau dan biru untuk setiap pikselnya. Warna pada setiap piksel ditentukan berdasarkan kombinasi dari warna red, green, dan blue (RGB). RGB merupakan citra 24 bit dengan komponen merah, hijau dan biru yang masing-masing umumnya bernilai 8 bit sehingga intensitas kecerahan warna sampai 256 level, sekitar 16 juta warna.

b. Citra keabuan

Citra dengan derajat keabuan berbeda dengan citra RGB, citra ini didefinisikan oleh satu nilai derajat warna. Umumnya bernilai 8 bit sehingga intensitas kecerahan warna sampai 256 level dan kombinasi warnanya 256 varian. Tingkat kecerahan paling rendah yaitu nilai 0 untuk warna hitam dan nilai 255 untuk warna putih. Ada 3 persamaan untuk mengkonversi citra yang memiliki warna RGB ke derajat keabuan[10].

1. *Lightness*

   $X = (\max(R,G,B)+\min(R,G,B))/2$ (Persamaan 1)

2. *Average*

   $X = (0.21\ R + 0.72\ G + 0.07\ B) / 3$ (Persamaan 2)

3. *Luminocity*

   $X = 0.21\ R + 0.72\ G + 0.07\ B$ (Persamaan 3)

   Dimana :

   X = citra greyscale, R = red, G = green, B = blue

2.3 Segmentasi Citra

Segmentasi Citra Digital adalah teknik untuk membagi / memisahkan suatu citra menjadi beberapa daerah(region) atau beberapa objek dimana setiap daerah memiliki kemiripan atribut[4].

Algoritma segmentasi citra didasarkan pada salah satu sifat dari dasar nilai intensitas, yaitu:

a. Discontinuity, pendekatan dengan membagi citra berdasarkan perubahan besar pada nilai intensitasnya, seperti tepi citra.
b. Similiarity, pendekatan dengan membagi citra ke dalam region-region yang serupa sesuai dengan kriteria awal yang diberikan. Contoh pendekatan ini adalah *thresholding, region growing, region splitting, merging.*

Dengan proses segmentasi masing-masing objek pada citra dapat diambil secara individu sehingga dapat digunakan sebagai input bagi proses lain.

Ada 2 macam segmentation

a. Full segmentation adalah pemisahan suatu objek secara individu dari background dan diberi ID(label) pada tiap-tiap segmen.
b. Partial segmentation adalah pemisahan sejumlah data dari background dimana data yang disimpan hanya data yang dipisahkan saja untuk mempercepat proses selanjutnya.

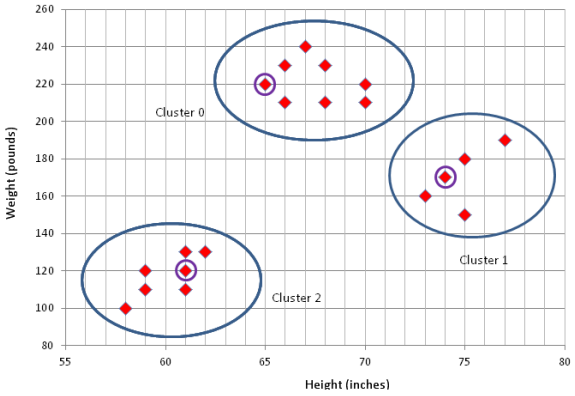
Segmentasi yang digunakan pada penelitian ini adalah segmentasi berbasis *clustering* yaitu menggunakan metode *K-Means Clustering*[15].

Clustering atau analisis cluster adalah pembentukan kelompok data dari himpunan data yang tidak diketahui kelompok-kelompok atau kelas-kelasnya, pengertian lain *clustering* adalah proses menentukan data-data termasuk ke dalam *cluster* yang mana. *Clustering* bukanlah proses klasifikasi karena dalam proses klasifikasi data dikelompokkan ke dalam kelas-kelas yang telah diketahui sebelumnya. Beberapa metode atau model untuk melakukan clustering, seperti :

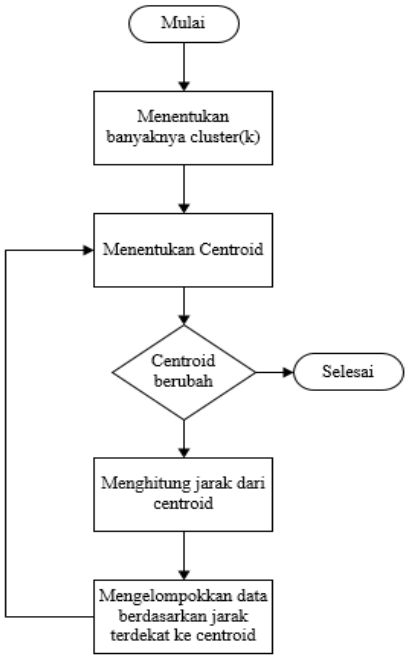
1. Model connectivity
2. Model centroid
3. Model density

4. Model subspace

Algoritma K-Means *clustering* merupakan model *centroid*. Model *centroid* adalah model yang menggunakan centroid untuk membuat *cluster*, *centroid* adalah nilai titik tengah suatu *cluster* dan *centroid* digunakan untuk menghitung jarak suatu objek terhadap *centroid*[9]. Suatu objek data termasuk dalam suatu *cluster* jika memiliki jarak terpendek terhadap *centroid cluster* tersebut.



Gambar 2.1 K-Means Clustering



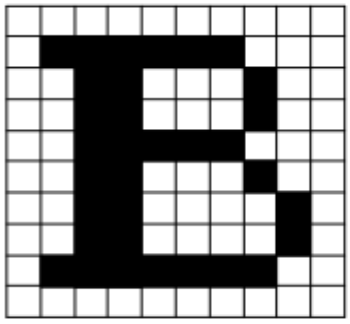
Gambar 2.2 Diagram Alir K-Means Clustering

Tujuan dari algoritma ini adalah membagi data menjadi beberapa kelompok, algoritma ini menerima masukan berupa data tanpa label kelas[9]. Pada algoritma ini komputer mengelompokkan sendiri data-data menjadi masukannya tanpa

mengetahui terlebih dahulu target kelasnya dimana masukan yang diterima adalah data atau objek dan k buah kelompok(*cluster*) yang di inginkan.

2.4 Operasi Biner

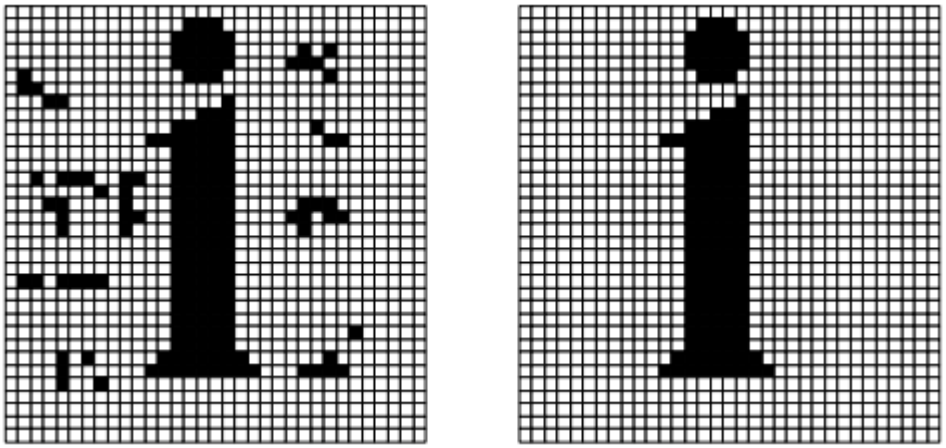
Citra binari adalah citra yang hanya mempunyai dua nilai derajat keabuan yaitu hitam dan putih, dimana piksel objek bernilai 1 dan latar belakang bernilai 0, 0 adalah putih dan 1 adalah hitam, pada citra ini latar belakang berwarna putih sedangkan objek berwarna hitam.



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

Gambar 2.3 Citra Biner

Untuk fokus pada analisis bentuk morfologi maka intensitas piksel tidak terlau penting dibandingkan bentuk, setelah objek dipisahkan dari latar belakang maka objek dapat dihitung dari citra biner.



Gambar 2.4 kiri : gangguan pada citra biner yang mengandung huruf “i”, kanan : citra yang dihasilkan setelah dilakukan penapisan/filtering (T=10) [JAI95]

2.5 Operasi Morfologi

Pemrosesan citra secara morofologi dilakukan pada citra biner, walaupun tidak menutup kemungkinan dilakukan pada citra dengan skala keabuan 0-255. Dengan pendekatan morofologi kita memandang suatu citra sebagai himpunan, bukan sebagai suatu fungsi intensitas terhadap posisi(x,y).

Secara umum pemrosesan citra secara morfologi dilakukan dengan cara passing sebuah *structuring element* terhadap sebuah citra dengan cara yang hampir sama dengan konvolusi. *Structuring element* memiliki titik poros yang disebut titik origin/titik asal/titik acuan.

Beberapa operasi morfologi yang dapat kita lakukan adalah

1. Dilasi, erosi
2. Opening, closing
3. Thinning, shrinking, pruning, thickening, skeletonizing, dll.

Salah satu operasi morfologi yang akan digunakan pada penelitian ini adalah operasi dilasi(*dilation*), dilasi adalah proses penumbuhan / penebalan / penggabungan dari citra biner/titik latar (0) dengan bagian dari objek (1), berdasarkan struktur *structuring element* yang digunakan. Dilasi berguna ketika diterapkan dalam obyek-obyek yang terputus dikarenakan

1. Hasil pengambilan citra yang terganggu oleh noise,
2. Kerusakan objek fisik yang dijadikan citra digital atau disebabkan resolusi yang kurang bagus. Misalkan teks pada kertas yang sudah rusak sehingga bentuk huruf terputus-putus dan lain sebagainya.

Kegunaan operasi morfologi adalah untuk

1. Menghilangkan noise
2. Mengisolasi objek

Algoritma Morfologi Citra

1. Konversi citra ke dalam bentuk skala keabuan (*Grayscale*)
2. Lakukan binerisasi citra (*thresholding*)
3. Morfologi

Operasi dilasi memiliki sifat sama dengan operasi penambahan pada operasi minkowski yaitu menggunakan operasi himpunan gabungan(*union*). Misalkan A dan B adalah himpunan-himpunan pixel.

Dilasi A ke B dinotasikan dengan $A \oplus B$ dan di definisikan sebagai berikut

$$A \oplus B = \bigcap_{x \in B} A_x \qquad \text{(Persamaan 4)}$$

Ini berarti bahwa untuk setiap titik x ∈ *B,* maka dilakukan translasi atau penggeseran dan kemudian

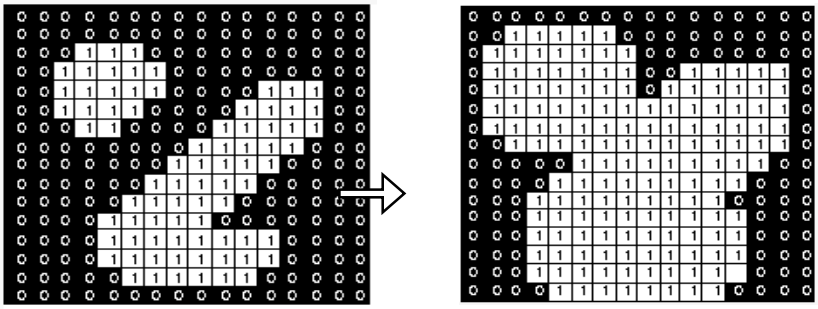
dan kemudian menggabungkan seluruh hasilnya (*union*). Secara matematis dituliskan

$$A \oplus B = \{(x, y) + (u, v) : (x, y) \in A, (u, v) \in B\} \qquad \text{(Persamaan 5)}$$

Dilasi mempunya hukum komutatif, yaitu

$$A \oplus B = B \oplus A \qquad \text{(Persamaan 6)}$$

Berikut contoh citra sebelum dan sesudah dilakukan proses dilasi



Gambar 2. 5 Citra Sebelum dan sesudah dilasi

2.6 Deteksi tepi (*Edge Detection*)

Deteksi Tepi pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari objek-objek 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.

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

Macam metode yang banyak digunakan untuk proses deteksi tepi, antara lain

1. Metode Robert
2. Metode Prewitt
3. Metode Sobel

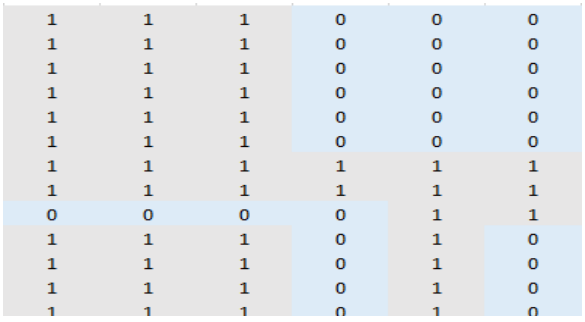
Pada tahap deteksi tepi digunakan metode sobel karena kemampuannya dalam mengurangi noise sebelum melakukan perhitungan deteksi tepi.

Kernel filter yang digunakan dalam metode sobel adalah

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{dan} \quad V = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

2.7 Labeling

Labeling adalah suatu proses pemberian label yang sama pada sekumpulan piksel pembentuk objek yang saling berdekatan pada suatu citra. Objek yang berbeda memiliki label yang berbeda. Labeling termasuk pada pemrosesan citra pada tahap intermediate level, ia memiliki peran yang penting pada pengolahan citra karea dapat mempermudah proses penganalisaan bentuk dan pengenalan pola pada tahap high level. Labeling memiliki sifat yang dependent dan regional, dependent artinya pemerian label harus memperhitungkan pixel label sebelumnya. Sifat regional adalah pemberian label harus memperhitungkan posisi pixel-pixel secara regional didalam citra.



Gambar 2.6 bentuk binari citra labelling

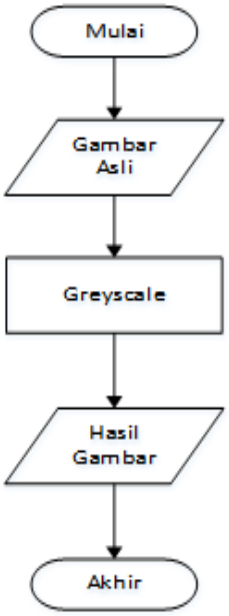
2.8 Cropping

Cropping citra artinya adalah pengambilan bagian tertentu dari suatu citra digital menjadi matriks baru yang independent. Cropping sangat berguna apabila kita hanya membutuhkan bagian tertentu dari suatu citra digital.

2.9 Grayscale

Merupakan proses untuk mengubah warna menjadi keabu-abuan. Dengan mengubah nilai RGB setiap piksel gambar menjadi satu nilai yang sama sehingga setiap piksel memiliki nilai yang sama untuk ketiga unsur warna serta didapatkan nilai matriks grayscale.

Berikut alur grayscale



Gambar 2.7 Diagram Alir Grayscale

2.10 Tekstur

Tekstur merupakan karakteristik intrinsik dari suatu citra yang terkait dengan tingkat kekasaran (*roughness*), granularitas (*granulation*) dan keteraturan (*regularity*) susunan struktural piksel. Aspek tekstural dari sebuah citra dapat dimanfaatkan sebagai dasar dari segmentasi, klasifikasi maupun interpretasi citra. Tesktur dapat didefinisikan sebagai fungsi dari variasi spasial intensitas piksel (nilai keabuan) dalam citra.

Berdasarkan strukturnya, tekstur dapat diklasifikasikan dalam 2 golongan

1. Makrostruktur

Tekstur makrostruktur memiliki perulangan pola lokal secara periodik dalam suatu daerah citra, biasanya terdapat pada pola-pola buatan manusia dan cenderung mudah untuk direpresentasikan secara matematis.

2. Mikrostruktur

Pada tektur mikrostruktur, pola-pola lokal dan perulangan tidak terjadi begitu jelas, sehingga tidak mudah untuk memberikan definisi testur yang komprehensif.

Tekstur dapat dicirikan sebagai berikut

a. Pengulangan pola dari variasi lokal sehingga membentuk kesatuan yang utuh
b. Meyediakan informasi susunan spasial dari warna dan intensitas citra
c. Dicirikan dengan distribusi spasial dari level intensitas dari nilai piksel ketetanggaan
d. Tidak bisa didefinisikan per point karena ia merupakan sebuah pola atau kesatuan.

2.11 Analisis Tekstur

Analisis tekstur merupakan dasar dari berbagai macam aplikasi, aplikasi dari analisis tekstur antara lain, penginderaan jarak jauh, pencitraan medis, identifikasi kualitas suatu bahan (kayu, kulit, batu, dan lain-lain). Analisis tekstur bekerja dengan mengamati pola ketetanggaan antar piksel dalam domain spasial. Dua persoalan yang seringkali berkaitan dengan analisis tekstur adalah :

1. Ekstraksi Ciri

Ekstraksi ciri pada citra merupakan langkah awal dalam melakukan klasifikasi dan interpretasi citra. Proses ini berkaitan dengan kuantisasi karakteristik citra ke dalam sekelompok nilai ciri yang sesuai. Dalam praktikum ini kita akan mengamati metode ekstraksi ciri statistik orde pertama dan kedua.

2. Segmentasi Citra

Segmentasi citra merupakan proses yang bertujuan untuk memisahkan suatu daerah pada citra dengan daerah lainnya. Berbeda dengan citra non tekstural, segmentasi citra tekstural tidak dapat didasarkan pada intensitas piksel per piksel,

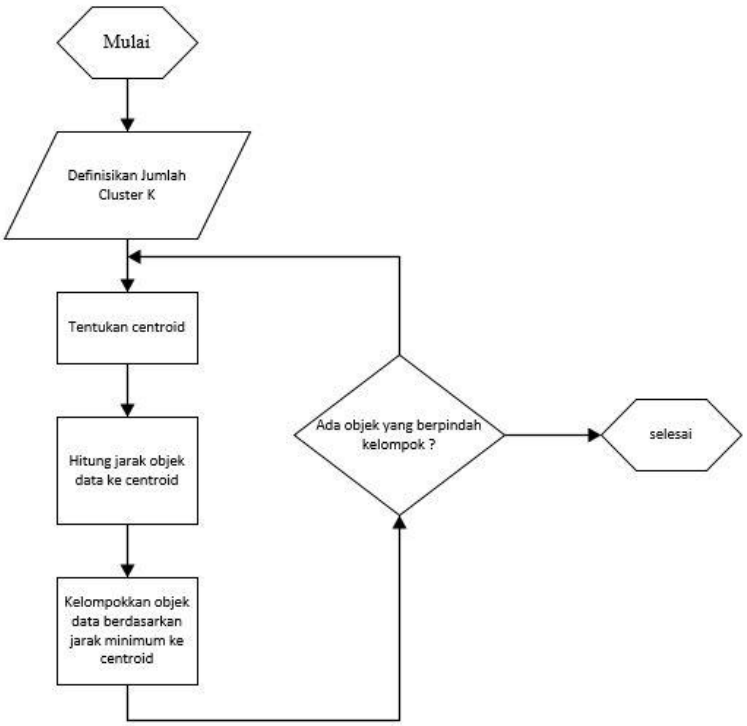
tetapi perlu mempertimbangkan perulangan pola dalam suatu wilayah ketetanggaan lokal.

2.12 K Means *Clustering*

Salah satu algoritma clustering, bertujuan untuk membagi data menjadi beberapa Kelompok. Algoritma ini menerima data tanpa label kelas. Komputer mengelompokkan sendiri data-data yang menjadi masukan tanpa mengetahui terlebih dahulu target kelasnya. Masukan yang diterima adalah data atau objek dan k buah kelompok(cluster) yang diinginkan dimana centroid adalah titik pusat cluster. Alasan digunakannya K *Means Clustering* adalah karena data harus dikelompokkan untuk membantu proses klasifikasi.

Algoritma untuk K *Means Clustering* adalah :

1. Pilih k buah titik centroid secara acak
2. Kelompokkan data sehingga terbetuk K buah cluster dengan titik centroid dari setiap *centroid* dari setiap *cluster* merupakan titik centroid yang telah dipilih sebelumnya.
3. Perbaharui nilai titik *centroid.*
4. Ulangi langkah 2 dan 3 sampai nilai dari titik *centroid* tidak lagi berubah.



Gambar 2.8 Flowchar k means *clustering*

K *Means Clustering* bisa membantu proses klasifikasi KNN, karena metode ini akan mengelompokkan objek sesuai kriteria masing-masing pada tahap preprocessing.

2.13 Ekstraksi Fitur

Ekstraksi fitur (*fitur extraction*) adalah bagian fundamental dari analisis citra. Fitur adalah karakteristik unik dari suatu objek. Karakteristik fitur yang baik sebisa mungkin memenuhi persyaratan berikut

1. Dapat membedakan suatu objek dengan objek lainnya(*discrimination*).
2. Memperhatikan kompleksitas komputasi daam memperoleh fitur.
3. Tidak terikat (*independence*) dalam arti bersifat invarian terhadap berbagai transformasi (rotasi, penskalaan, pergeseran dan lain sebagainya)
4. Jumlahnya sedikit, karena fitur yang jumlahnya sedikit dapat menghemat waktu komputasi dan ruang penyimpanan untuk proses selanjutnya (proses pemanfaatan fitur).

Berbagai metode ekstraksi fitur suatu citra

1. Ampitudo

   Adalah ciri yang paling sederhana dan mungkin paling berguna bagi suatu objek, ciri ini berhubungan dengan properti fisik suatu objek, seperti luminance, nilai tristimulus, nilai spektral

2. Histogram

   Adalah ciri yang didasarkan pada histogram dari suatu citra. Bila x menyatakan tingkat keabuan pada suatu citra maka probabilitas dari x dinyatakan dengan :

$$P(x) = \frac{\text{Banyaknya titik} - \text{titik yang memiliki tingkat keabuan x}}{\text{Total banyaknya titik pada daerah suatu citra}}$$

(Persamaan 7)

   Beberapa ciri yang dihitung berdasarkan histogram, antara lain

   a. Rata-rata / *mean*

$$m_1 = \sum_{x=0}^{L} xP(x)$$ (Persamaan 8)

   b. Variance

$$\mu_2 = \sum_{x=0}^{L} (x - m_1)^2 P(x)$$ (Persamaan 9)

   c. Skewness

$$\mu_3 = \sum_{x=0}^{L} (x - m_1)^3 P(x)$$ (Persamaan 10)

   d. Kurtosis

$$\mu_4 - 3 = \sum_{x=0}^{L} (x - m_1)^4 P(x) - 3$$ (Persamaan 11)

3. Matriks co-occurrence

Adalah matriks yang dibangun menggunakan histogram tingkat kedua, matriks ini berukuran L x L (L menyatakan tingkat keabuan) dengan elemen $P(x_1,x_2)$ yang merupakan distribusi probabilitas bersama dari pasangan pixel

dengan tingkat keabuan x1 yang berlokasi pada koordinat (*j*,*k*) dengan x2 berlokasi pada koordinat (*m*,*n*). koordinat pasangan pixel ini berjarak r dengan sudut Θ. Histogram tingkat kedua $P(x_1,x_2)$ dihitung dengan pendekatan sebagai berikut.

$$P(x1, x2) = \frac{\text{Banyaknya pasangan titik} - \text{titik dengan tingkat keabuan x1 dan x2}}{\text{banyaknya titik pada daerah suatu citra}}$$

(Persamaan 12)

Histogram dan Matriks *Coocurens* dihimpun ke dalam satu analisa yaitu analisa metode statistik, untuk menggambarkan ciri dari setiap citra, metode ini dinamakan analisis tektur karena parameter statistik yang diekstrak dari citra merupakan karakteristik yang merepresentasikan bentuk/pola tekstur dalam citra.

Syarat terbentuknya tekstur :

1. Adanya pola primitif yang terdiri dari satu atau lebih piksel, bentuk nya dapat berupa titik, garis lurus, luasan dan lain-lain yang merupakan elemen dasar dari sebuah bentuk.
2. Pola-pola primitif tadi muncul berulang-ulang dengan interval jarak dan arah tertentu sehingga dapat diprediksi atau ditemukan karakteristik pengulangannya.

Berdasarkan orde statistiknya analisis tektur dapat dikategorikan menjadi 3, yaitu analisis tekstur orde satu, orde dua, dan orde diatas dua. Semakin tinggi orde tingkat kesulitannya juga meningkat.

Pada orde satu kita hanya menganalisa sebaran piksel dengan nilai intensitas tertentu secara global pada citra (menggunakan histogram citra). Analisis orde dua perlu mempertimbangkan korelasi antara 2 piksel dengan nilai intensitas tertentuk yang memiliki hubungan jarak dan arah tertentu.

Untuk kebutuhan analisa, analisis tekstur orde dua memerlukan bantuan matriks Coocurens untuk citra keabuan, biasanya disebut GLCM(*Gray Level Coocurence Matrix*). Analisa orde ini lebih baik dalam merepresentasikan tekstur citra dalam parameter-parameter terukur, seperti *contrast, correlation, homogenity, entropy, energy.*

2.14.1 Orde Satu

Ekstraksi ciri orde pertama merupakan metode pengambilan ciri yang didasarkan pada karakteristik histogram citra. Histogram menunjukkan probabilitas kemunculan nilai derajat keabuan piksel pada suatu citra. Dari nilai-nilai pada histogram yang dihasilkan, dapat dihitung beberapa parameter ciri orde pertama, antara lain adalah *mean, skewness, variance, kurtosis,* dan *entropy*.

1. Mean (μ)

   Menunjukkan ukuran dispersi dari suatu citra

   Rumus

$$\mu = \sum_{n} f_n p(f_n) \qquad \text{(Persamaan 13)}$$

2. Variance (σ^2)

   Menunjukkan variasi elemen pada histogram dari suatu citra

   Rumus

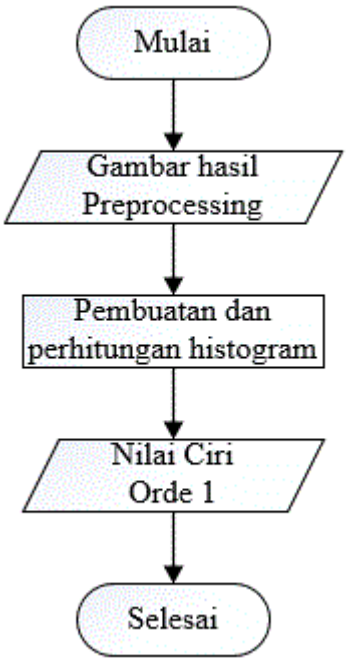
$$\sigma^2 = \sum_{n} (f_n - \mu)^2 p(f_n) \qquad \text{(Persamaan 14)}$$

2.14.2 Orde Dua

Statistik orde dua dibutuhkan ketika orde satu tidak dapat mengenali beberapa perbedaan ciri citra. Salah satu teknik untuk memperoleh ciri statistik orde dua adalah dengan menghitung probabilitas hubungan ketetanggaan antara dua piksel pada jarak dan orientasi sudut tertentu. Pendekatan ini bekerja dengan membentuk sebuah matriks kookurensi dari data citra, dilanjutkan dengan menentukan ciri sebagai fungsi dari matriks tersebut[16].

Kookurensi berarti kejadian bersama yaitu jumlah kejadian satu level nilai piksel bertetangga dengan satu level nilai piksel lain dalam jarak(*d*) dan orientasi (θ) tertentu. Jarak dinyatakan dalam piksel dan orientasi dinyatakan dalam derajat. Orientasi dibentuk dalam empat arah sudut dengan interval 45^o yaitu 0^o, 45^o, 90^o, 135^o. Sedangkan jarak antar piksel biasanya ditetapkan sebesar 1 piksel.

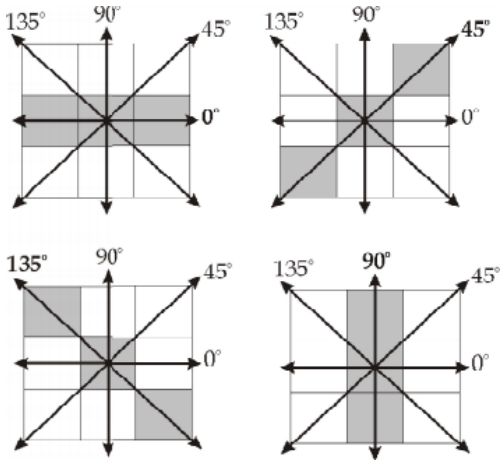
Matriks kookurensi merupakan matriks bujur sangkar dengan jumlah elemen sebanyak kuadrat jumlah level intensitas piksel pada citra. Matriks dihitung dari nilai piksel yang berpasangan dan memiliki intensitas tertentu. Misalkan d adalah jarak antara dua pixel yaitu (x_1,y_1) dan (x_2,y_2) dan Θ tetha didefinisikan sebagai sudut antara keduanya, maka matriks ini merupakan distribusi spasial dari $P_{d\,\Theta}(i,j)$.



Gambar 2.9 Alur Orde Satu

Berikut ilustrasi mengambarkan arah sudut dengan jarak 1 piksel dan 4 jenis sudut yang digunakan

a. $0^o = 180^o$
b. $45^o = 225^o$
c. $90^o = 270^o$
d. $135^o = 315^o$



Gambar 2.10 Hubungan ketetanggan antar piksel

Keterangan Gambar

(a) Citra Masukan

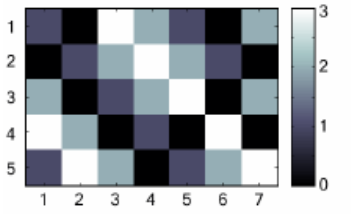
(b) Nilai intensitas citra masukan

(c) Hasil matriks kookurens 0^o setelah normalisasi

(d) Hasil matriks kookurensi 45^o setelah normalisasi

(e) Hasil matriks kookurensi 90^o setelah normalisasi

(f) Hasil matriks kookurensi 135^o setelah normalisasi



(a)

1	0	3	2	1	0	2
0	1	2	3	2	1	0
2	0	1	2	3	0	2
3	2	0	1	0	3	0
1	3	2	0	1	2	3

(b)

0	0.1333	0.0833	0.0667
0.1333	0	0.0833	0.0167
0.0833	0.0833	0	0.1167
0.0667	0.0167	0.1167	0

(c)

0.1667	0	0.0833	0.0417
0	0	0.1042	0.125
0.0833	0.1042	0.0833	0.0208
0.0417	0.125	0.0208	0

(d)

0	0.1429	0.1071	0.0536
0.1429	0	0.0536	0.0179
0.1071	0.0536	0	0.125
0.0536	0.0179	0.125	0

(e)

0.2083	0	0.0833	0
0	0.2083	0.0208	0
0.0833	0.0208	0.1667	0
0	0	0	0.2083

(f)

Gambar 2.11 Pembuatan Matrik Coocurence

Setelah memperoleh matriks kookurensi maka dapat dihitung ciri statistik orde dua yang merepresentasikan citra yang diamati. Haralick et al mengusulkan berbagai jenis ciri tekstural yang dapat diekstraksi pada matrik kookurensi. Pada penelitian ini ada 2 ciri statistik orde dua yang akan dihitung seperti *contrast* dan *autocorrelation.*

Dimana tahap perhitungannya adalah

1. Contrast

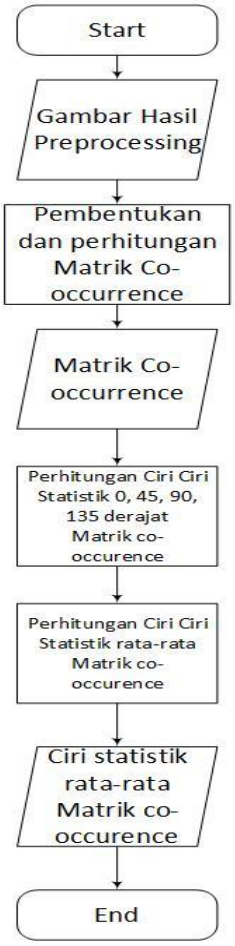
   Mengukur frekuensi spasial dari citra dan perbedaan moment matriks. Perbedaan yang dimaksud adalah perbedaan tinggi dan rendahnya piksel. Contrast akan bernilai 0 jika piksel ketetanggaan mempunya nilai yang sama.

$$f_4 = \sum_i \sum_j (i-j)^2 . p(i,j) \qquad \text{(Persamaan 15)}$$

2. Autocorrelation

Tahap mengukur correlation diantara garis diagonal utama.

$$f_7 = \sum_i \sum_j i.j.p(i,j) \qquad \text{(Persamaan 16)}$$



Gambar 2.12 Alur Orde dua

2.14 Klasifikasi

Klasifikasi secara harfiah adalah penggolongan atau pengelompokan. Menurut kamus besar bahasa indonesia klasifikasi adalah penyusunan bersistem dalam kelompok atau golongan menurut standar yang ditetapkan.

Klasifikasi citra digital merupakan proses pengelompokan piksel ke dalam kelas-kelas tertentu. Tujuan dari proses ini adalah untuk mendapatkan gambar tematik, gambar tematik adalah gambar yang terdiri dari bagian-bagian yang menyatakan suatu objek atau tema tertentu.

Proses klasifikasi ada 2 macam *Supervised* dan *Unsupervised*

1. Klasifikasi citra terawasi (*supervised*)

Klasifikasi ini memasukkan terlebih dahulu setiap piksel citra kedalam suatu kategori objek yang sudah diketahui. Sebelum klasifikasi dilakukan maka harus memasukkan inputan sebagai dasar pengklasifikasian yang akan dilakukan.

Klasifikasi k-nearest neighbor termasuk ke dalam klasifikasi *supervised*, dimana hasil sampel uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada k-NN.

2. Klasifikasi citra tidak terawasi (*unsupervised*)

   Dalam prosesnya klasifikasi ini tidak menggunakan referensi penunjang, proses dilakukan berdasarkan perbedaan tingkat keabuan setiap piksel pada citra. Klasifikasi ini mencari kelompok-kelompok(*cluster*) piksel-piksel, kemudian menandai setiap piksel ke dalam sebuah kelas berdasarkan parameter-parameter penglompokkan awal yang akan didefinisikan.

Dua klasifikasi ini dapat digabungkan penggunaan nya dalam satu kasus, dimana k means digunakan untuk memisahkan objek-objek yang bermacam-macam dari suatu citra dan ini akan mempercepat proses klasifikasi.

2.14.1 Klasifikasi K-Nearest Neighbor

K-Nearest Neighbor merupakan sebuah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumnya. KNN termasuk dalam golongan *supervised* dimana hasil query *instance* yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori yang ada dalam KNN. Nantinya kelas yang baru dari suatu data akan dipilih berdasarkan grup kelas yang dekat jarak vektornya.

Alasan digunakan algoritma KNN untuk klasifikasi karena ia memiliki konsistensi yang kuat, menjamin error rate tidak lebih dari dua kali.

Tujuan dari metode ini adalah mengklasifikasikan obyek baru berdassarkan atribut dan trainning sample. *Classifier* tidak menggunakan model apapun untuk dicocokan dan hanya berdasarkan pada memori. Diberikan titik *query*, akan ditemukan sejumlah k obyek atau (titik training) yang paling dekat dengan titik

query. Klasifikasi menggunakan voting terbanyak diantara klasifikasi dari k obyek. Metode
k-nearest neighbor (KNN) menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari *query instance* yang baru.

Metode k-nearest neighbor (KNN) sangatlah sederhana, bekerja berdasarkan jarak terpendek dari query instance ke training sample untuk menemukan KNN-nya. *Training sample* diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian–bagian berdasarkan klasifikasi *trainning sample*. sebuah titik pada ruang ini ditandai kelas c jika kelas c merupakan klasifikasi yang paling banyak ditemukan pada k buah tetangga terdekat dari titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasrkan *Euclidean Distance*.

Jarak *Euclidian* paling sering digunakan menghitung jarak. Jarak *euclidean* berfungsi menguji ukuran yang bisa dgunakan sebagai interpretasi kedekatan jarak antara dua obyek, yang direpresentasikan sebagai berikut:

$$d(x-y) = \sqrt{\sum_{j=1}^{i}(x_j - y_j)^2} \qquad \text{(Persamaan 17)}$$

Keterangan:

d = jarak data uji ke data pembelajaran.

x_j = data uji ke-j, dengan j = 1, 2, . . . n.

y_j = data pembelejaran ke-j dengan j = 1, 2, . . . n.

Nilai k yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Nilai K yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus dimana klasifikasi diprediksikan berdasarkan trainning data yang paling dekat (dengan kata lain, k=1) disebut metode nearest neighbor.

Ketetapan metode KNN sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap metode ini sebagian besar membahas

bagaimana memilih dan memberi bobot terhadap fitur agar performa klasfikasi menjadi lebih baik.

Langkah-langkah untuk menghitung k-*nearest neighbor* :

1. Menentukan parameter K (jumlah tetangga yang paling dekat).
2. Menghitung kuadrat jarak euclid (*query inctance*) masing-masing obyek terhadap data sampel yang diberikan.
3. Kemudian mengurutkan objek-objek tersebut kedalam kelompok yang mempunyai jarak *euclidean* terkecil.
4. Mengumpulkan kategori Y (Klasifikasi nearest neighbor).
5. Dengan menggunakan kategori nearest neighbor yang paling mayoritas maka dapat diprediksikan nilai *query instance* yang telah dihitung.

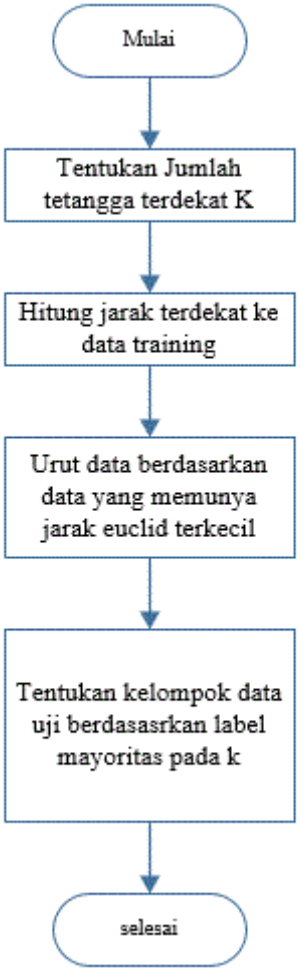
Kelebihan dari metode k-nearest neighbor adalah sebagai berikut:

1. Ketangguhan terhadap data yang memiliki banyak noise.
2. Efektif terhadap data yang berukuran besar.

Kekurangan dari metode k-nearest neighbor adalah sebagai berikut:

1. Nilai k harus ditentukan secara manual.
2. Trainning berdasarkan jarak harus menggunakan banyak sampel untuk mendapatkan hasil yang terbaik.
3. Memerlukan komputasi tingkat tinggi karena perlunya menghitung satu persatu data testing terhadap semua data trainning.

Adapun metode dari KNN pada flowchart berikut :



Gambar 2.13 Flowchart metode KNN

2.15 Pemrograman Terstruktur

Pemrograman Terstruktur adalah tindakan untuk membuat program yang berisi instruksi-instruksi dalam bahasa komputer yang tersusun secara logis dan sistematis supaya mudah dimengerti, mudah dites dan mudah dimodifikasi. Pemrograman terstruktur adalah bahasa pemrograman yang mendukung pembuatan program sebagai kumpulan prosedur. Prosedur saling memanggil dan dipanggil dari manapun dalam program dan dapat menggunakan parameter yang berbeda-beda untuk setiap pemanggilan. Bahasa pemrograman terstruktur mendukung abstraksi data, pengkodean terstruktur dan kontrol program terstruktur,

sedangkan prosedur sendiri adalah bagian dari program untuk melakukan operasi-operasi yang sudah ditentukan dengan menggunakan parameter tertentu.

Prinsip pemrograman terstruktur

1. Gunakan rancangan pendekatan dari atas ke bawah (*top down design*),
2. Bagi program ke dalam modul-modul logika sejenis,
3. Gunakan sub-program untuk proses-proses sejenis yang sering digunakan.
4. Gunakan pengkodean terstruktur, seperti if.. then, do..while, dan lain-lain.
5. Gunakan nama-nama bermakna (mnemonic names)
6. jika suatu proses telah sampai pada suatu titik/langkah tertentu, maka proses selanjutnya tidak boleh mengeksekusi langkah sebelumnya.
7. Buatkan dokumentasi yang akurat dan berarti

Manfaat pemrograman terstruktur

1. Dapat menangani program yang besar dan komplek
2. Dapat menghindari konflik internal team
3. Membagi kerja team berdasarkan modul-modul program yang sudah dirancang
4. Kemajuan pengerjaan sistem dapat dimonitor dan dikaji

Tujuan pemrograman terstruktur

1. Meningkatkan kehandalan suatu program
2. Program mudah dibaca dan ditelusuri
3. Menyederhanakan kerumitan program
4. Biaya perawatan dan dokumentasi yang dibutuhkan rendah

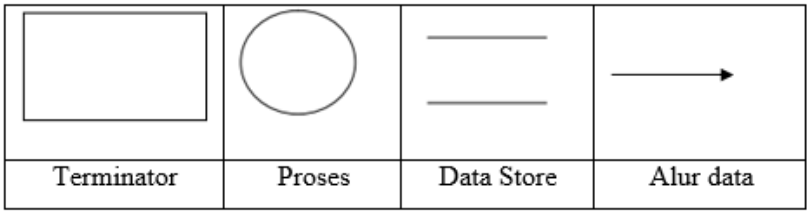
Langkah membuat program yang baik dan terstruktur adalah

1. Mendefiniskan Masalah
2. Menentukan Solusi
3. Memilih Algoritma
4. Menulis Program
5. Menguji Program
6. Menulis Dokumentasi

2.16 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan seorang profesioanal sistem menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data. DFD sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem.

DFD bisa dikatakan sebagai alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. Komponen Data Flow Diagram, menurut yourdan dan deMarco



Gambar 2.14 DFD yourdan dan deMarco

Komponen Terminator/Entitas Luar

Terminator mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Terminator dikenal dengan nama entitas luar (*eksternal entity*), terdapat dua jenis terminator :

1. Terminator sumber(*source*) merupakan terminator yang menjadi sumber.
2. Terminator tujuan (*sink*) merupakan terminator yang menjadi tujuan data/informasi sistem. Terminator dapat berupa orang, sekelompok orang, organisasi, departemen di dalam organisasi atau perusahaan yang sama tetapi diluar kendali sistem yang sedang membuatnya. Terminator dapat berupa departemen, divisi atau sistem di luar sistem yang berkomunikasi dengan sistem yang sedang dikembangkan.

2.17 Matlab (*Matrix Laboratory*)

Matlab adalah salah software aplikasi untuk menyelesaikan berbagai masalah teknis. Matlab mengintegrasikan komputasi, visualisas dan pemrograman daam

suatu model yang sangat mudah untuk dipakai, dimana masalah-masalah dan penyelesaiaannya diekspresikan dalam notasi matematika yang familiar. Penggunaan matlab meliputi bidang-bidang :

1. Pembentukan dan pengembangan Algoritma
2. Matematika dan komputasi
3. Pemodelan, simulasi sistem hingga pembuatan prototype
4. Akusisi data, analisis data, plotting data dan visualisasi
5. Pembuatan software aplikasi termasuk antar muka (Graphical User Interfaces)
6. Matlab memungkinkan manipulasi matriks, pemflotan fungsi dan data.

Matlab merupakan suatu sistem interaktif yang memiliki elemen data dalam suatu *array* sehingga memungkinkan untuk memecahkan banyak masalah teknis yag terkait dengan komputasi, khususnya yang berhubungan dengan matriks dan formulasi vektor.

Fitur matlab sudah banyak dikembangkan yang lebih dikenal dengan nama *toolbox. Toolbox* merupakan kumpulan dari fungsi-fungsi Matlab (*M-Files*) yang telah dikembangkan ke suatu lingkungan kerja matlab untuk memecahkan masalah.

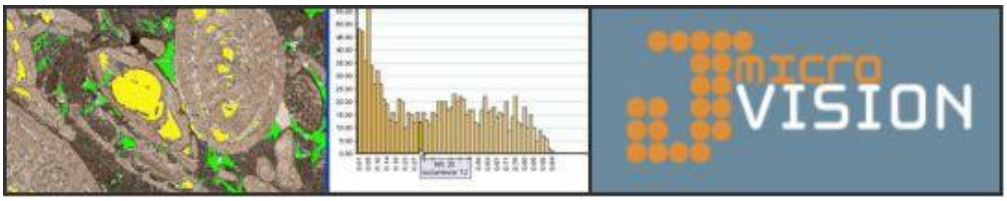
Area-area yang sudah dipecahkan dengan *toolbox* saat ini meliputi pengolahan sinyal, sistem kontrol, *neural network, fuzzy logic, wavelets*, dan lain-lain.

2.18 Sayatan Tipis Batuan Pasir

Sayatan tipis batu adalah batu yang sudah di potong menggunakan mesin pemotong batu, sehingga batu yang sudah dipotong menjadi sayatan kecil bisa di ambil fotomikroskopnya menggunakan mikroskop polar. Setelah pengambilan foto dilakukan dan menghasilkan gambar dalam format *.jpg maka gambar dianalisis kandungan dan persentasi dari kandungan batu mineral oleh para geolog[2].

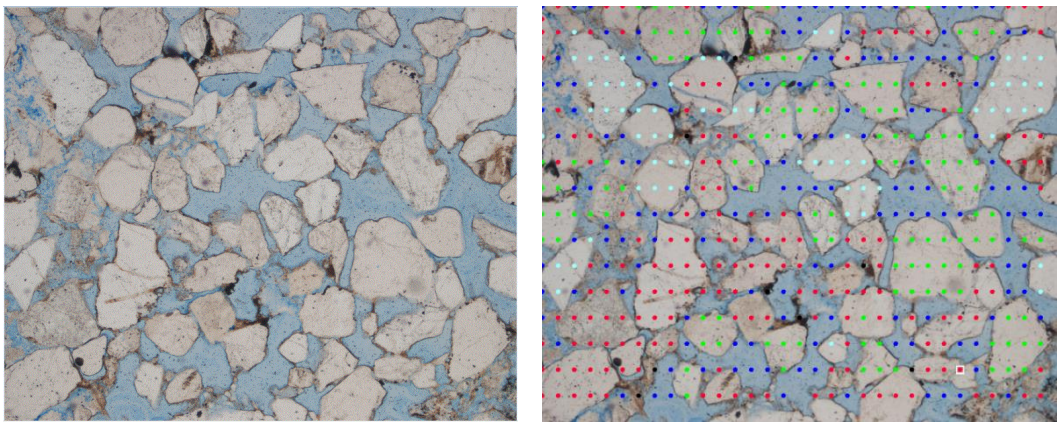
2.19 Point Counting

Adalah metode statistik untuk menghitung persentasi suatu mineral pada citra hasil fotomikroskop[17]. Sistem pengidentifikasian yang sedang berjalan menggunakan software Jmicrovision, dimana software ini digunakan untuk menghasilkan titik warna yang berbeda pada objek yang akan diteliti.



Gambar 2.15 Tampilan Software JMicro

Dimana hasil dari software ini adalah gambar dengan hasil titik dengan warna yang berbeda.



Gambar 2.16 Citra Sayatan hasil JMicro

Pemberian titik pada citra dilakukan melalui inputan pada program khusus dengan jumlah 300-500 titik, untuk warna dari masing-masing titik ditentukan sebelum nya oleh para geolog, pada citra ini diberikan 4 warna

a. Warna biru muda untuk kuarsa
b. Warna merah untuk feldspar
c. Warna biru untuk rongga, rongga tidak berisi mineral
d. Warna hijau untuk mineral lithic, mineral ini adalah mineral gabungan yang jumlah nya hanya 2-3 % dari total jumlah mineral dan mineral ini tidak dihitung.

Pada tugas akhir ini yang dihitung hanya jumlah persentasi dari mineral kuarsa dan feldspar dan jenis mineral yang terkandung.

Cara menghitung manual nya[17]

Rumus Total Area = P/Pt

Keterangan

P = jumlah point dari masing-masing warna

Pt= total dari semua titik

P/Pt = area fraksi

Pada gambar diatas ada 100 warna merah dari total jumlah titik 500

Jadi perhitungannya

P=100

Pt=500

Total area = P/Pt = 100/500 * 100% = 0.2%

Ada 0,2 % jumlah mineral merah/feldspar pada citra ini, pada kenyataannya proses perhitungan pada metode ini masih banyak kesalahan menurut para geolog karena pemberian titik oleh aplikasi tidak sesuai harapan.