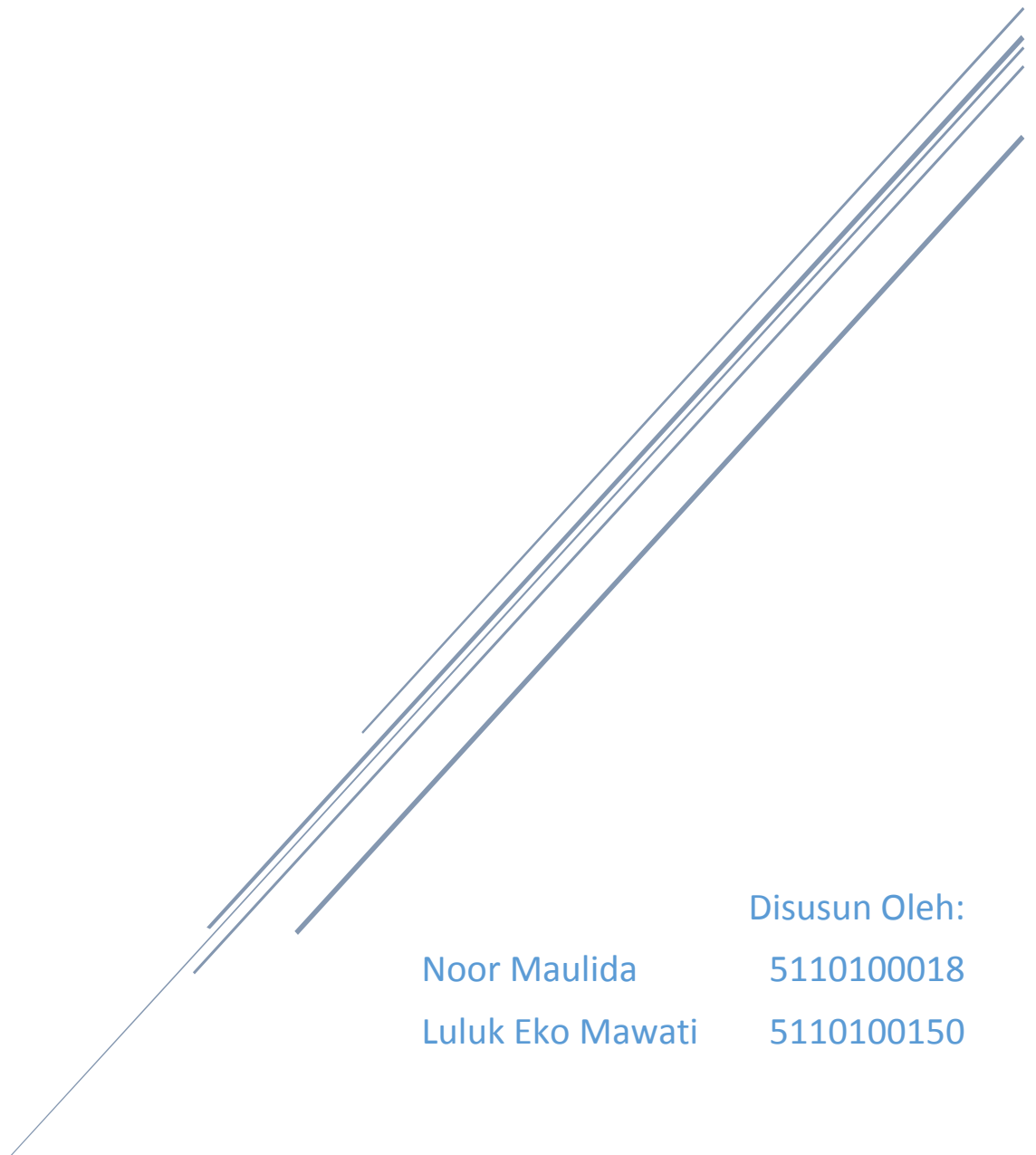


LAPORAN FINAL PROJECT PENGOLAHAN CITRA DIGITAL

“Klasifikasi Kanker Payudara”



Disusun Oleh:

Noor Maulida 5110100018

Luluk Eko Mawati 5110100150

Jurusan Teknik Informatika
Institut Teknologi Sepuluh Nopember Surabaya

Daftar Isi

Daftar Isi	1
Pendahuluan	2
Rancangan <i>Final Project</i>	2
<i>Preprocessing</i> (Peningkatan Kualitas Citra)	3
Algoritma Possibility Distribution	3
Baris Kode Program Peningkatan Kualitas Citra	5
Contoh Peningkatan Kualitas Citra	7
Hasil Percobaan Peningkatan Kualitas Citra	8
Segmentasi	9
Algoritma Thresholding.....	9
Algoritma <i>Cropping</i> Citra	10
Baris Kode Program Segmentasi.....	10
Hasil Percobaan Segmentasi	13
Ekstraksi Fitur	14
Algoritma GLCM (Gray Level Co-Occurrence Matrix).....	14
Mean	14
Varian	14
Standart Deviasi	15
Contrast.....	15
Kurtosis.....	16
Smoothness.....	17
Baris Kode Program Ekstraksi Fitur	17
Hasil Percobaan Ekstraksi Fitur	18
Klasifikasi	19
Algoritma Neural Network.....	19
Baris Kode Program Training	20
Hasil Percobaan Training	21
Baris Kode Program Klasifikasi	22
Hasil Percobaan Klasifikasi	22
Kesimpulan	23
Daftar Pustaka	24

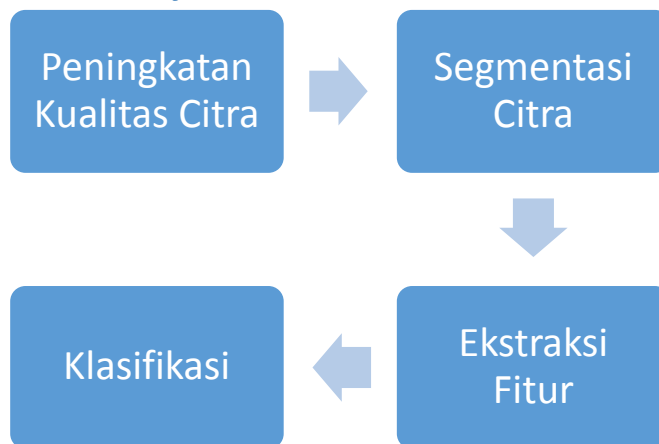
Pendahuluan

Peningkatan kualitas citra pada komputasi di bidang medis adalah penggunaan computer untuk memperjelas citra [1]. Jenis-jenis peningkatan kualitas citra meliputi pengurangan derau, peningkatan titik, dan peningkatan kontras. Peningkatan kualitas dapat digunakan untuk mengembalikan citra yang buruk atau meningkatkan fitur-fitur tertentu pada citra [2].

Mamografi adalah pemeriksaan kelenjar payudara dng menggunakan sinar X yang digunakan untuk mendeteksi awal kanker payudara. Mamografi digital ditujukan pada aplikasi pada teknik sistem digital pada *mammogram* digital. Sistem digital memiliki kapasitas untuk mendeteksi kanker payudara. 12 bit deteksi resolusi biasanya dibutuhkan untuk menghasilkan *mammogram* digital dengan resolusi tinggi tanpa kehilangan informasi dari *mammogram* asli.

Analisis mamografi adalah sebuah tantangan karena rendahnya iluminasi dan tingginya derau pada citra yang dapat mencapai 10-15% dari maksimum intensitas piksel. Mammogram adalah citra yang paling sulit untuk dianalisis dan diinterpretasikan [2].

Rancangan *Final Project*



Gambar 1. Tahapan pengerjaan *final project*

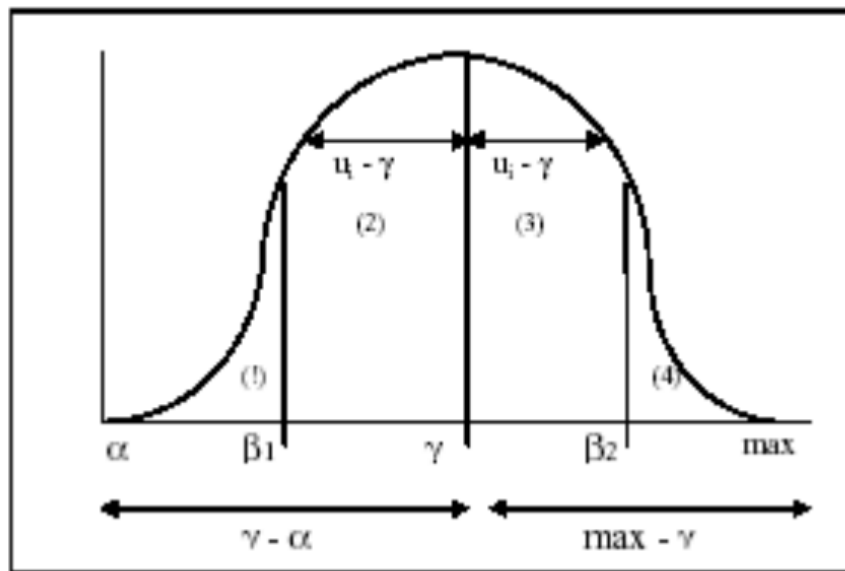
Topik untuk pengerjaan *final project* yang kami gagas adalah “Deteksi kanker payudara berdasarkan hasil analisis mamografi”. Adapun tahapan pengerjaan *final project* sebagaimana terdapat pada Gambar 5, berikut penjelasannya:

- **Peningkatan kualitas citra**, pada tahapan ini dilakukan peningkatan kualitas citra sehingga mudah dilakukan pengolahan terhadap citra mamografi.
- **Segmentasi citra**, tahapan segmentasi bertujuan untuk membagi citra menjadi segmen-segmen sehingga didapatkan bentuk yang sesuai untuk proses pengolahan.
- **Ekstraksi fitur**, hal ini dilakukan untuk mendapatkan fitur-fitur apa saja yang ada pada citra tersebut..
- **Klasifikasi**, setelah mendapatkan informasi dari citra, citra yang ada diklasifikasikan menjadi dua, apakah normal atau kanker.

Preprocessing (Peningkatan Kualitas Citra)

Algoritma Possibility Distribution

Algoritma *possibility distribution* untuk peningkatan citra menggunakan pendekatan logika fuzzy dengan menggunakan 5 parameter, yaitu α , β_1 , γ , β_2 dan \max seperti yang terdapat di Gambar 2



Gambar 2. Fungsi Possibility Distribution

Dari parameter yang dibutuhkan, α merepresentasikan nilai minimum distribusi, γ merepresentasikan nilai rata-rata distribusi dan \max merepresentasikan nilai maksimum distribusi. Fungsi transformasi fuzzy untuk mendapatkan keseluruhan nilai didefinisikan sebagai berikut:

- $\alpha = \min$
- $\beta_1 = (\alpha + \gamma) / 2$
- $\gamma = \text{mean}$
- $\beta_2 = (\max + \gamma) / 2$
- $\max = \max$

Tujuan dari penggunaan algoritma *possibility distribution* dalam peningkatan kualitas citra adalah untuk menurunkan tingkat keabuan piksel yang memiliki nilai keabuan di antara β_1 dan β_2 . Cara yang dilakukan adalah memberikan nilai intensitas baru di piksel di antara β_1 dan γ , γ dan β_2 dengan nilai direksi yang berlawanan menuju nilai mean γ .

Aturan fuzzy di bawah ini digunakan untuk melakukan peningkatan kontras citra berdasarkan Gambar 2:

1. **Rule-1:** If $\alpha \leq u_i \leq \beta_1$ then $P = 2((u_i - \alpha) / (\gamma - \alpha))^2$
2. **Rule-2:** If $\beta_1 \leq u_i \leq \gamma$ then $P = 1 - 2((u_i - \gamma) / (\gamma - \alpha))^2$
3. **Rule-3:** If $\gamma \leq u_i \leq \beta_2$ then $P = 1 - 2((u_i - \gamma) / (\max - \gamma))^2$
4. **Rule-4:** If $\beta_2 \leq u_i \leq \max$ then $P = 2((u_i - \gamma) / (\max - \gamma))^2$

dimana $u_i = f(x,y)$ adalah intensitas piksel ke-i.

Peraturan yang menurunkan tingkat keabuan piksel yang memiliki nilai keabuan di antara β_1 dan β_2 direpresentasikan dengan peraturan 2 dan 3. Langkah-langkah algoritma *possibility distribution* dapat dijelaskan sebagai berikut:

Langkah-1: Inisialisasi Parameter

- Matriks data diset gambar yang akan ditingkatkan kualitas citranya
- Dari gambar tersebut dilakukan pencarian nilai minimum, maksimum dan rata-rata (mean) nilai keabuan
- Nilai β_1 diset dengan hasil dari $(\min + \text{mean})/2$
- Nilai β_2 diset dengan hasil dari $(\text{mean} + \max)/2$

Langkah-2: Fuzzification

1. Untuk semua piksel, lakukan penerapan aturan fuzzy (Rule-1, Rule-2, Rule-3 dan Rule-4) untuk mendapatkan nilai keabuan yang baru

Langkah-3: Modification

2. Untuk semua piksel, dihitung nilai fuzzy data yang didapatkan dari nilai keabuan yang baru di pangkatkan dua

Langkah-4: Defuzzification

3. Untuk semua piksel, dihitung data kontras baru yang didapatkan dari nilai fuzzy data dikalikan dengan matriks data gambar awal

Baris Kode Program Peningkatan Kualitas Citra

Berikut penjelasan baris kode program dari program peningkatan kualitas citra.

Inisialisasi data awal:

```
data2 = imread('image.jpg');
data = rgb2gray(data2);
sizes = size(data);
height = sizes (1);
width = sizes (2);

newgraylevel = zeros(height,width);
fuzzydata = zeros(height,width);
enhanceddata = zeros(height,width);

min = 256;
maks = 1;

total = 0;
```

Menghitung nilai minimum dan maksimum dari semua piksel:

```
if data(i,j) < min
    min = data(i,j);
end

if data(i,j) > maks
    maks = data(i,j);
end
```

Menghitung nilai rata-rata dari intensitas piksel:

```
means = mean2(data);
```

Langkah 1: Inisialisasi Parameter

*Menentukan β_1 yang didapat dari perhitungan $(\min + \text{means})/2$ dan β_2 yang didapat dari perhitungan $(\text{maks} + \text{means})/2$ dimana,

Min : nilai intensitas piksel minimum pada citra

Maks : nilai intensitas piksel maksimum pada citra

Means : nilai rata-rata intensitas piksel pada citra

```
b1 = (min+means)/2;
b2 = (maks+means)/2;
```

Langkah 2: Fuzzification

Untuk semua piksel yang ada pada matriks data, dilakukan *fuzzification* sesuai dengan aturan yang ada:

1. Jika piksel lebih besar sama dengan nilai min dan piksel kurang dari nilai β_1 , maka tingkat keabuan diubah menjadi $2((\text{piksel} - \min) / (\text{means} - \min))^2$

```
if (data(i,j) >= min) && (data(i,j) < b1)
    newgraylevel(i,j) = 2*((data(i,j)-min)/(means-min))^2;
end
```

2. Jika piksel lebih besar sama dengan nilai β_1 dan piksel kurang dari nilai means, maka tingkat keabuan diubah menjadi $1-2((\text{piksel}-\text{min})/(\text{means}-\text{min}))^2$

```
if (data(i,j)>= b1) && (data(i,j) < means)
    newgraylevel(i,j) = 1-(2*((data(i,j)-means)/(means-min))^2);
end
```

3. Jika piksel lebih besar sama dengan means dan piksel kurang dari nilai β_2 , maka tingkat keabuan diubah menjadi $1-2((\text{piksel}-\text{means})/(\text{maks}-\text{means}))^2$

```
if (data(i,j)>= means) && (data(i,j) < b2)
    newgraylevel(i,j) = 1-(2*((data(i,j)-means)/(maks-means))^2);
end
```

4. Jika piksel lebih besar sama dengan nilai β_2 dan piksel kurang dari maks, maka tingkat keabuan diubah menjadi $2((\text{data}-\text{means})/(\text{maks}-\text{means}))^2$

```
if (data(i,j)>= b2) && (data(i,j) < maks)
    newgraylevel(i,j) = 2*((data(i,j)-means)/(maks-means))^2;
end
```

Langkah 3: Modifikasi

Menghitung setiap data *fuzzy* pada setiap piksel, untuk setiap nilai (tingkat keabuan) nilainya dipangkatkan dua.

```
fuzzydata(i,j) = newgraylevel(i,j)^2;
```

Langkah 4: Defuzzification

Untuk semua piksel yang ada pada citra dilakukan perhitungan data yang sudah ditingkatkan kualitasnya dengan cara mengalikan data *fuzzy* dengan data intensitas tiap piksel.

```
enhanceddata(i,j) = fuzzydata(i,j)*data(i,j);
```

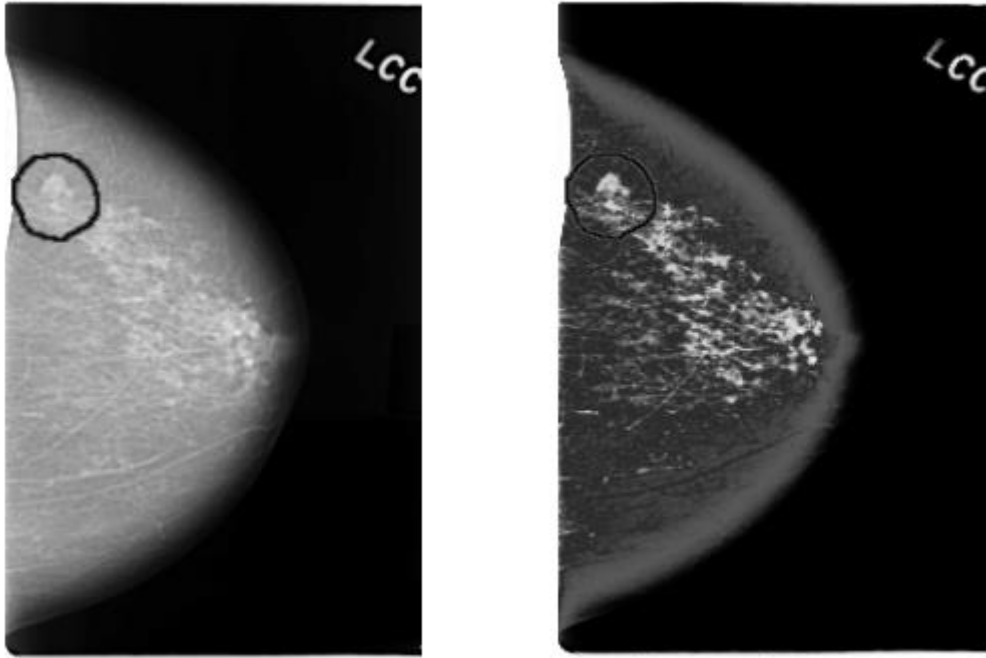
Menampilkan gambar pada figure

```
figure
subplot(1,2,1);
imshow(data)

subplot(1,2,2);
imshow(enhanceddata)
```

Contoh Peningkatan Kualitas Citra

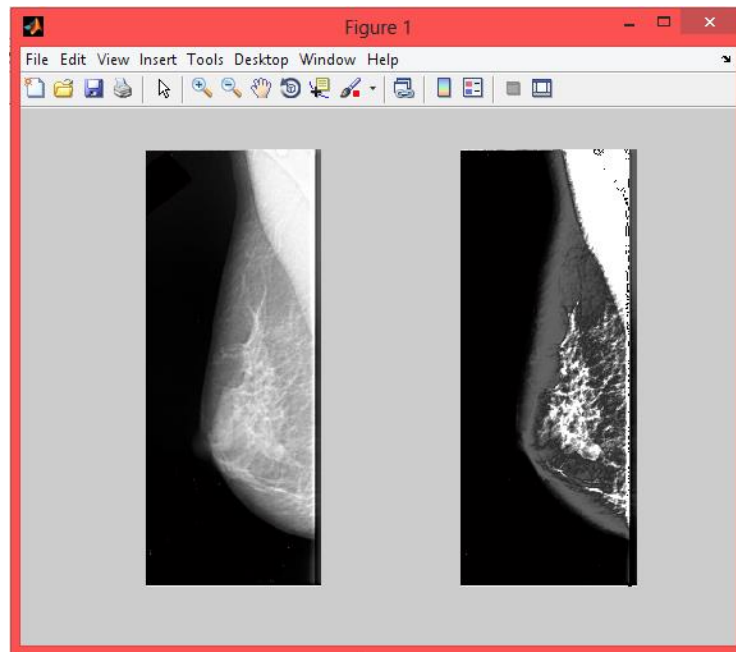
Gambar 2 merupakan contoh mamografi kanker yang ditunjukkan oleh penulis paper. Untuk gambar sebelah kiri merupakan citra asli, sedangkan gambar sebelah kanan merupakan citra setelah dilakukan peningkatan kualitas citra menggunakan algoritma *possibility distribution*.



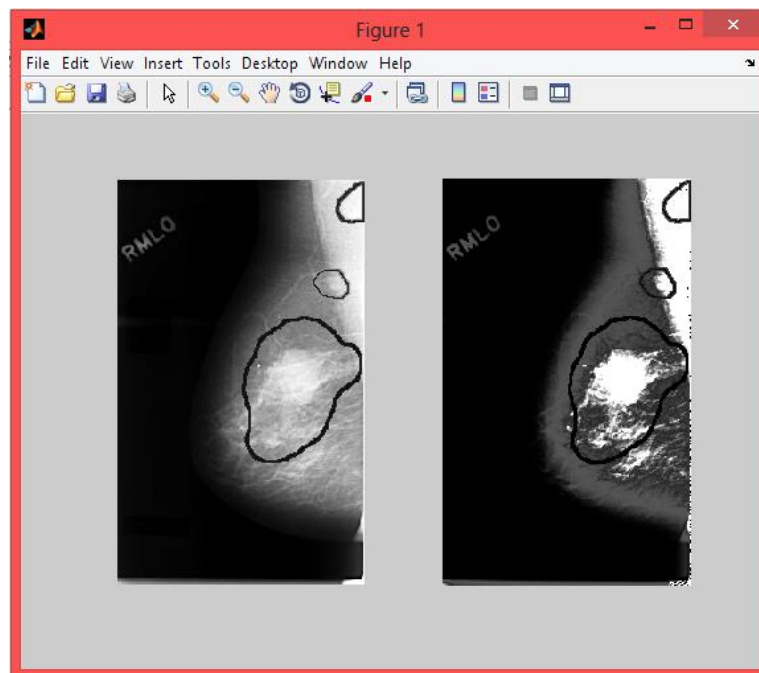
Gambar 3. Contoh peningkatan kualitas citra

Hasil Percobaan Peningkatan Kualitas Citra

Berikut adalah hasil percobaan peningkatan kualitas citra pada dataset citra mamografi. Gambar 3 merupakan hasil percobaan dengan dataset mamografi normal, sedangkan Gambar 4 merupakan hasil percobaan dengan dataset mamografi kanker.



Gambar 4. Hasil percobaan menggunakan data mamografi normal



Gambar 5. Hasil percobaan menggunakan data mamografi kanker

Segmentasi

Algoritma Thresholding

Tujuan dari algoritma Thresholding Otsu adalah melakukan segmentasi gambar dengan cara dibedakan menjadi 2 kelas, yaitu background (nilai diset ke 0) dan objek (nilai diset ke 1) menggunakan level tertentu sebagai pembatas. Langkah-langkah yang dilakukan untuk melakukan thresholding Otsu adalah:

1. Dari gambar yang telah ditingkatkan citranya, diambil jumlah piksel di level i , yang direpresentasikan dengan n_i , total dari jumlah piksel di definisikan dengan $N = n_1 + n_2 + \dots + n_n$. Lalu didapat nilai p dari jumlah piksel dibagi dengan total dari jumlah piksel tersebut.

$$p_i = n_i / N, \quad p_i \geq 0, \quad \sum_{i=1}^L p_i = 1.$$

```
p = counts / sum(counts);
```

2. Setelah itu dilakukan pembagian piksel menjadi 2 kelas, C_0 sebagai background, dan C_1 sebagai objek.

$$\omega_0 = \Pr(C_0) = \sum_{i=1}^k p_i = \omega(k)$$
$$\omega_1 = \Pr(C_1) = \sum_{i=k+1}^L p_i = 1 - \omega(k)$$

```
C0 -> omega = cumsum(p);  
C1 -> 1 - omega;
```

3. Setelah itu dilakukan pembagian piksel menjadi 2 kelas, C_0 sebagai background, dan C_1 sebagai objek.

$$\mu(k) = \sum_{i=1}^k i p_i$$

```
mu = cumsum(p .* (1:num_bins)');
```

4. Lalu untuk mendapatkan mean direpresentasikan dengan rumus sebagai berikut:

$$\mu_T = \mu(L) = \sum_{i=1}^L i p_i$$

```
mu_t = mu(end);
```

5. Untuk mendapatkan class variance didefinisikan dengan rumus sebagai berikut:

$$\sigma_W^2 = \omega_0 \sigma_0^2 + \omega_1 \sigma_1^2$$

$$\begin{aligned}\sigma_B^2 &= \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \\ &= \omega_0 \omega_1 (\mu_1 - \mu_0)^2\end{aligned}$$

[due to (9)) and

$$\sigma_T^2 = \sum_{i=1}^L (i - \mu_T)^2 p_i$$

```
sigma_b_squared = (mu_t * omega - mu).^2 ./ (omega .* (1 - omega));
```

6. Setelah itu dilakukan pencarian nilai maksimum dari semua nilai variance yang terlihat, dan ditentukan nilai threshold menggunakan rata-rata dari ditemukannya nilai variance. Jika tidak ditemukan, nilai threshold berarti 0.0.

$$\sigma_B^2(k^*) = \max_{1 \leq k < L} \sigma_B^2(k).$$

$$S^* = \{k; \omega_0 \omega_1 = \omega(k)[1 - \omega(k)] > 0, \quad \text{or } 0 < \omega(k) < 1\}.$$

```
maxval = max(sigma_b_squared);
if isfinite(maxval)
    idx = mean(find(sigma_b_squared == maxval));
    level = (idx - 1) / (num_bins - 1);
else
    level = 0.0;
```

Algoritma Cropping Citra

Citra hasil segmentasi dipotong sehingga objek citra yang diproses menjadi lebih terfokus. Untuk memotong citra, perlu untuk menentukan batas atas, batas bawah, batas kanan, dan batas kiri dari citra. Untuk mendapatkan batas-batas tersebut dilakukan dengan cara perulangan pada setiap piksel, jika dalam perulangan telah menemukan piksel dengan intensitas satu (1), maka piksel tersebut menjadi batas citra. Perulangan ini dilakukan secara horizontal dan vertikal, baik dari sisi atas, bawah, kanan, atau kiri. Setelah batas ini ditemukan, selanjutnya adalah memotong citra dari batas yang telah ditentukan sebelumnya.

Baris Kode Program Segmentasi

Berikut penjelasan baris kode program untuk tahapan segmentasi

```
%mengambil background dari citra
background = imopen(enhanceddata, strel('disk', 15));

%mendapatkan gambar tanpa background
I2 = enhanceddata - background;

%mengatur nilai intensitas citra atau warna
I3 = imadjust(I2);

%mengambil nilai threshold berdasarkan Otsu thresholding
```

```

level_threshold = fungsi_graythresh(I3);

citra = zeros(row, col);

%membuat matriks citra hitam putih berdasarkan thresholding sebelumnya
for i=1:row
    for j=1:col
        if enhanceddata(i,j) > level_threshold
            citra(i,j) = 1;
        else
            citra(i,j) = 0;
        end
    end
end
end

```

Algoritma Thresholding

```

function level = fungsi_graythresh(I)

if ~isempty(I)
    I = im2uint8(I(:));
    num_bins = 256;
    counts = imhist(I,num_bins);

    % rumus dari Otsu paper.
    p = counts / sum(counts);
    omega = cumsum(p);
    mu = cumsum(p .* (1:num_bins)');
    mu_t = mu(end);

    sigma_b_squared = (mu_t * omega - mu).^2 ./ (omega .* (1 - omega));
    maxval = max(sigma_b_squared);
    if isfinite(maxval)
        idx = mean(find(sigma_b_squared == maxval));
        level = (idx - 1) / (num_bins - 1);
    else
        level = 0.0;
    end
else
    level = 0.0;
end
end

```

Cropping Citra

```

%inisialisasi awal batas untuk memotong citra
batas_atas = 1;
batas_bawah = 1;
batas_kanan = 1;
batas_kiri = 1;

%mencari batas bawah potongan citra
while (true)
    for i=1:row
        for j=1:col
            if citra(i,j) == 1
                batas_bawah = i;
                break;
            end
        end
    end
end

```

```

        end
    end
    break;
end

%mencari batas kanan potongan citra
while (true)
    for j=1:col
        for i=1:row
            if citra(i,j) == 1
                batas_kanan = j;
                break;
            end
        end
    end
    break;
end

%mencari batas atas potongan citra
while (true)
    for i=row:-1:1
        for j=col:-1:1
            if citra(i,j) == 1
                batas_atas = i;
                break;
            end
        end
    end
    break;
end

%mencari batas kiri potongan citra
while (true)
    for j=col:-1:1
        for i=row:-1:1
            if citra(i,j) == 1
                batas_kiri = j;
                break;
            end
        end
    end
    break;
end

data_cropped_double = zeros(batas_bawah-batas_atas+1, batas_kanan-
batas_kiri+1);
data_cropped_int = zeros(batas_bawah-batas_atas+1, batas_kanan-
batas_kiri+1);

%set crop
for i=batas_atas:batas_bawah
    for j=batas_kiri:batas_kanan
        %memotong 'citra' dengan batas yang sudah dihitung
        data_cropped_double ((i-batas_atas)+1, (j-batas_kiri)+1) = citra
(i,j);

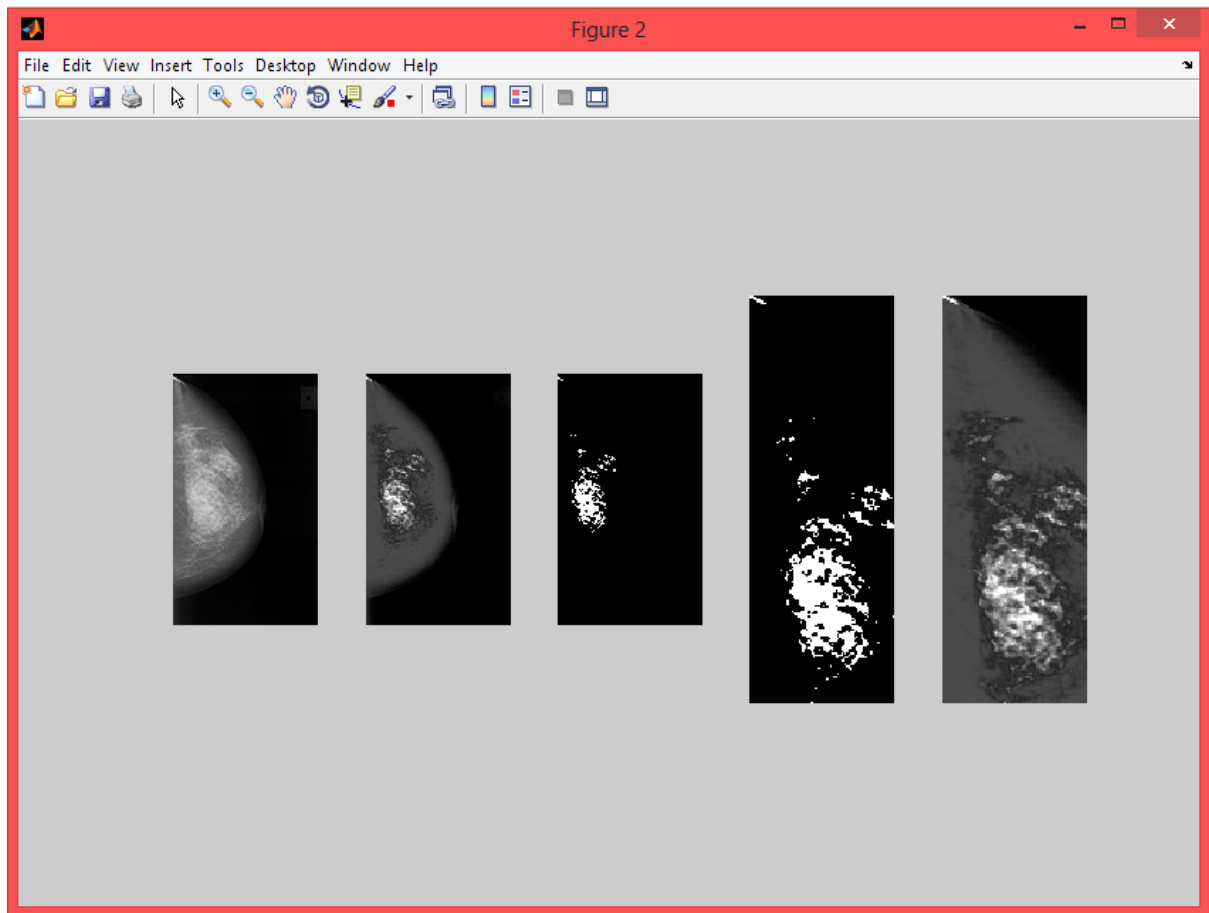
        %memotong 'enhanceddata' berdasarkan batas yang telah dihitung
        data_cropped_int ((i-batas_atas)+1, (j-batas_kiri)+1) = enhanceddata(i,j);
    end
end

```

end

Hasil Percobaan Segmentasi

Gambar 2 menjelaskan hasil percobaan tahapan segmentasi dan ekstraksi fitur sebagai hasil dari progress kami.



Gambar 6. (a) citra asli, (b) citra setelah dilakukan *enhancement*, (c) citra setelah dilakukan *thresholding*, potongan dari hasil segmentasi citra dengan cara 'Otsu' (d) dan potongan citra dari citra yang sudah di *enhancement*

Ekstraksi Fitur

Algoritma GLCM (Gray Level Co-Occurrence Matrix)

GLCM (*Gray Level Co-occurrence Matrix*) adalah matriks dua dimensi yang merepresentasikan variasi tingkat keabuan pada citra. Pengukuran semacam korelasi, energy, kontras, entropi, homogeneity merupakan perhitungan dari GLCM.

Mean

Persamaan Mean:

$$\mu = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N p(i, j)$$

Contoh matriks:

1	2	2	3
3	3	4	5
1	3	1	1
2	4	1	5

Perhitungan mean:

$$(1 + 3 + 1 + 2)/4$$

1.75

$$(2 + 3 + 3 + 4)/4$$

3

$$(2 + 4 + 1 + 1)/4$$

2

$$(3 + 5 + 1 + 5)/4$$

3.5

$$(1.75 + 3 + 2 + 3.5)/4$$

2.5625

Varian

Persamaan Varian:

$$\text{Varian} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (p(i, j) - \mu)^2$$

Contoh matriks:

1	2	2	3
3	3	4	5
1	3	1	1
2	4	1	5

Perhitungan Variance

Mean: 2.5625

$(1 - 2.5625)^2 = 2,44140625$	$(2 - 2.5625)^2 = 0,31640625$	$(2 - 2.5625)^2 = 0,31640625$	$(3 - 2.5625)^2 = 0,19140625$
$(3 - 2.5625)^2 = 0,19140625$	$(3 - 2.5625)^2 = 0,19140625$	$(4 - 2.5625)^2 = 2,06640625$	$(5 - 2.5625)^2 = 5,94140625$
$(1 - 2.5625)^2 = 2,44140625$	$(3 - 2.5625)^2 = 0,19140625$	$(1 - 2.5625)^2 = 2,44140625$	$(1 - 2.5625)^2 = 2,44140625$
$(2 - 2.5625)^2 = 0,31640625$	$(4 - 2.5625)^2 = 2,06640625$	$(1 - 2.5625)^2 = 2,44140625$	$(5 - 2.5625)^2 = 5,94140625$

Varian = $(2,44140625 + 0,31640625 + 0,31640625 + 0,19140625 + 0,19140625 + 0,19140625 + 2,06640625 + 5,94140625 + 2,44140625 + 0,19140625 + 2,44140625 + 2,44140625 + 2,06640625 + 5,94140625)/16$

Varian = 1,871094

Standart Deviasi

Persamaan Standart Deviasi

$$\sigma = \sqrt{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (p(i,j) - \mu)^2}$$

Contoh matriks:

1	2	2	3
3	3	4	5
1	3	1	1
2	4	1	5

Perhitungan Standart Deviasi:

Varian: 1,871094

STD : $\sqrt{1,871094}$

STD : 1,367879

Contrast

Persamaan Contrast:

$$\text{Contrast} = \sum_{i,j=0}^{G-1} (i - j)^2 P(i, j)$$

Penjelasan Contrast:

Dibuat matriks contrast yang ukurannya (baris) i x (kolom) j. Ketika baris (i) dan kolom (j) sama, nilai piksel adalah pada diagonal (ij) = 0. Nilai-nilai ini mewakili piksel sepenuhnya mirip dengan tetangga dari piksel tersebut, sehingga diberi bobot 0.

Jika i dan j berbeda 1 (perubahan kontras yang sedikit), ada perbedaan kecil, dan beratnya adalah $1^2 = 1$. Jika i dan j berbeda 2 (perubahan kontras yang cukup besar), kontras meningkat dan beratnya adalah $2^2 = 4$. Bobot terus meningkat secara eksponensial apabila (ij) meningkat.

Sehingga didapatkan matriks contrast:

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

Perhitungan Contrast:

Contoh matriks:

1	2	2	3
3	3	4	5
1	3	1	1
2	4	1	5

$$1*(0-0)^2 + 2*(0-1)^2 + 2*(0-2)^2 + 3*(0-3)^2 + 3*(1-0)^2 + 3*(1-1)^2 + 4*(1-2)^2 + 5*(1-3)^2 + 1*(2-0)^2 + 3*(2-1)^2 + 1*(2-2)^2 + 1*(2-3)^2 + 2*(3-0)^2 + 4*(3-1)^2 + 1*(3-2)^2 + 5*(3-3)^2$$

Contrast = 106

Kurtosis

Persamaan Kurtosis:

$$K = \left\{ \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N \left[\frac{p(i,j) - \mu}{\sigma} \right]^4 \right\} - 3$$

Perhitungan Kurtosis:

Contoh matriks:

1	2	2	3
3	3	4	5
1	3	1	1
2	4	1	5

K =

$$\begin{aligned} & (((1-2,5625)/1,367879)^4 + ((2-2,5625)/1,367879)^4 + ((2-2,5625)/1,367879)^4 + \\ & ((3-2,5625)/1,367879)^4 + ((3-2,5625)/1,367879)^4 + ((4-2,5625)/1,367879)^4 + \\ & ((5-2,5625)/1,367879)^4 + ((1-2,5625)/1,367879)^4 + ((3-2,5625)/1,367879)^4 + \\ & ((1-2,5625)/1,367879)^4 + ((1-2,5625)/1,367879)^4 + ((2-2,5625)/1,367879)^4 + \\ & ((4-2,5625)/1,367879)^4 + ((1-2,5625)/1,367879)^4 + ((5-2,5625)/1,367879)^4) / 16) - 3 \end{aligned}$$

$$K = ((8,51254 + 0,08578 + 1,279352 + 2,101794 + 20,165967)/16) - 3$$

$$K = -0,9909191875$$

Smoothness

Persamaan Smoothness:

$$R = 1 - \frac{1}{1 + \sigma^2}$$

Perhitungan Smoothness:

Contoh matriks:

1	2	2	3
3	3	4	5
1	3	1	1
2	4	1	5

σ^2 (Variance) = 1,871094

$$R = 1 - \frac{1}{1 + 1,871094^2}$$

$$R = 1 - \frac{1}{4,500993}$$

R = 0,777827

Baris Kode Program Ekstraksi Fitur

Berikut baris kode program untuk tahapan ekstraksi fitur.

```
function [cont,vars, stds, kurt, men,
smo]=feature_extraction(data_cropped_int)
    glcm = graycomatrix(data_cropped_int);
    %mencari nilai-nilai contrast, variance, standard deviation,
    %curtosis, mean, dan smoothness dari gambar yang telah disegmentasi
    stats = graycoprops(glcm,{'contrast'});
    cont = struct2array(stats);
    vars = var(data_cropped_int(:));
    stds = std(data_cropped_int(:));
    kurt = kurtosis(data_cropped_int(:));
    men = mean(data_cropped_int(:));
    smo = 1-(1/(1+vars));

    %mencetak nilai-nilai dari ekstraksi fitur
    fprintf('\t\t%f\n', cont);
    fprintf('\t\t%f\n', vars);
    fprintf('\t\t%f\n', stds);
    fprintf('\t\t%f\n', kurt);
    fprintf('\t\t%f\n', men);
    fprintf('\t\t%f\n', smo);
end
```

Hasil Percobaan Ekstraksi Fitur

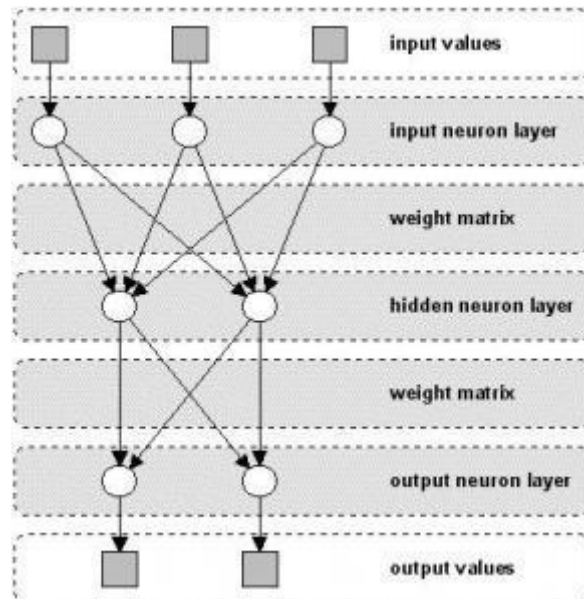
Berikut hasil dari ekstraksi fitur menggunakan kode program di atas yang kami ambil dari beberapa data mamogram:

file	contrast	std dev	curtosis	mean	variance	smoothness
cancer001	1.027975	0.628802	9.45129	0.539653	0.395393	0.283356
cancer002	1.39875	0.685388	9.736556	0.57835	0.469757	0.319615
cancer004	0.737317	0.605021	7.130871	0.527126	0.36605	0.267962
cancer005	0.649256	0.952167	3.081397	0.95672	0.906621	0.475512
cancer008	0.791496	0.59201	14.38957	0.418011	0.350476	0.25952
normal001	0.1923	0.148253	34.16894	0.188447	0.021979	0.021506
normal002	0.369617	0.22563	27.85677	0.22431	0.050909	0.048443
normal003	0.313964	0.17338	6.763178	0.195019	0.03006	0.029183
normal005	0.750176	0.444575	15.89192	0.41034	0.197647	0.16503

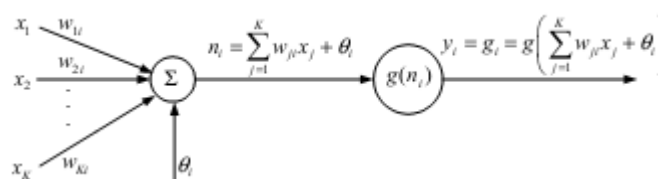
Klasifikasi

Algoritma Neural Network

Neural network merupakan suatu model yang dibuat untuk meniru fungsi belajar yang dimiliki otak manusia. Dalam Neural Network, neuron dikelompokkan dalam layer, yang disebut neuron layer. Biasanya setiap neuron dari sebuah layer dihubungkan ke semua neuron yang ada di layer belakang maupun depannya (kecuali input dan output). Informasi yang dikirim dalam sebuah Neural Network, dipropagasi layer – per – layer mulai dari input hingga output tanpa atau melalui satu atau lebih hidden layers. Bergantung pada algoritma yang digunakan, informasi juga dapat dipropagasi ke arah belakang (*backpropagation*). Gambar berikut menunjukkan Neural Network dengan tiga neuron layer.



Salah satu jenis *neural network* adalah *Multi Layer Perceptron* (MLP). MLP memiliki *input layer*, beberapa *hidden layer*, dan *output layer*. Titik *i* pada gambar 7 merupakan neuron pada jaringan dan *g* merupakan fungsinya.



Gambar 7 Single node pada jaringan MLP

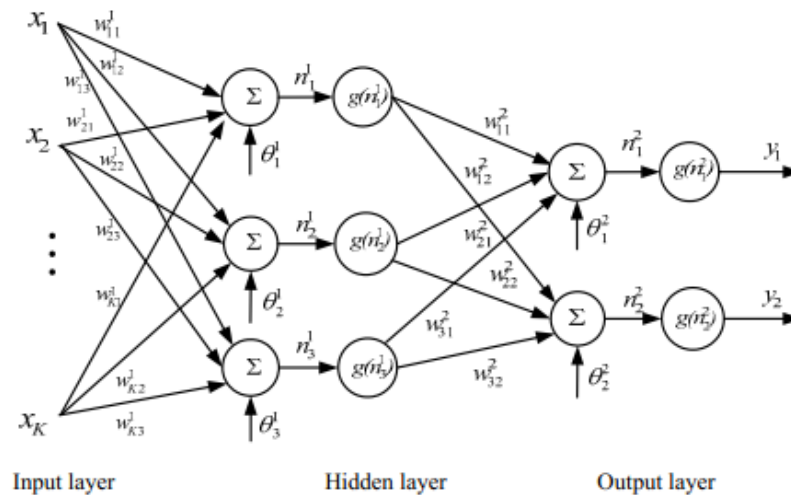
Input x_k , $k=1, \dots, K$ ke neuron dikalikan dengan bobot w_{ki} dan dijumlahkan dengan konstanta bias θ_i . Hasilnya n_i merupakan input untuk mengaktifkan fungsi g . Fungsi aktivasi ini dipilih untuk tujuan matematis *tangen* hiperbolik (*tanh*) atau fungsi sigmoid yang biasa digunakan. Tangen hiperbolik didefinisikan sebagai:

$$\tanh(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

Sehingga titik keluaran menjadi seperti ini:

$$y_i = g_i = g\left(\sum_{j=1}^K w_{ji} x_j + \theta_i\right)$$

Dengan menghubungkan beberapa titik secara seri dan parallel, maka jaringan MLP akan terbentuk, sebagaimana pada gambar 8.



Gambar 8 MLP dengan satu hidden layer

Keluaran y_i , $i=1,2$, dari MLP diatas menjadi:

$$y_i = g\left(\sum_{j=1}^3 w_{ji}^2 g(n_j^1) + \theta_j^2\right) = g\left(\sum_{j=1}^3 w_{ji}^2 g\left(\sum_{k=1}^K w_{kj}^1 x_k + \theta_j^1\right) + \theta_j^2\right)$$

Dari persamaan diatas, dapat disimpulkan bahwa jaringan MLP adalah map dengan parameter non-linier dari input $x \in R^k$ ke output $y \in R^m$ (disini $m=3$). Parameternya adalah bobot w_{ji}^k dan bias θ_j^k . Fungsi aktivasi g biasanya diasumsikan sama pada masing-masing layer.

Baris Kode Program Training

Berikut baris kode program untuk melakukan pelatihan terhadap data latih mammogram. Kami menggunakan *hiddenlayer* sebanyak 5.

```
inputs = hasil';
targets = target';
hiddenLayerSize = 5;
net = patternnet(hiddenLayerSize);
net.divideParam.trainRatio = 100/100;
net.trainParam.showWindow = false;
net.trainParam.showCommandLine = false;
[net,tr] = train(net,inputs,targets);
```

Hasil Percobaan Training

Neural Network

```
        name: 'Pattern Recognition Neural Network'
    efficiency: .cacheDelayedInputs, .flattenTime,
               .memoryReduction, .flattenedTime
    userdata: (your custom info)

    dimensions:

        numInputs: 1
        numLayers: 2
        numOutputs: 1
        numInputDelays: 0
        numLayerDelays: 0
    numFeedbackDelays: 0
    numWeightElements: 47
        sampleTime: 1

    connections:

        biasConnect: [1; 1]
        inputConnect: [1; 0]
        layerConnect: [0 0; 1 0]
        outputConnect: [0 1]

    subobjects:

        inputs: {1x1 cell array of 1 input}
        layers: {2x1 cell array of 2 layers}
        outputs: {1x2 cell array of 1 output}
        biases: {2x1 cell array of 2 biases}
        inputWeights: {2x1 cell array of 1 weight}
        layerWeights: {2x2 cell array of 1 weight}

    functions:

        adaptFcn: 'adaptwb'
        adaptParam: (none)
        derivFcn: 'defaultderiv'
        divideFcn: 'dividerand'
        divideParam: .trainRatio, .valRatio, .testRatio
        divideMode: 'sample'
        initFcn: 'initlay'
        performFcn: 'mse'
        performParam: .regularization, .normalization
        plotFcns: {'plotperform', plottrainstate, ploterrhist,
                  plotconfusion, plotroc}
        plotParams: {1x5 cell array of 5 params}
        trainFcn: 'trainscg'
        trainParam: .showWindow, .showCommandLine, .show, .epochs,
                   .time, .goal, .min_grad, .max_fail, .sigma,
                   .lambda

    weight and bias values:

        IW: {2x1 cell} containing 1 input weight matrix
        LW: {2x2 cell} containing 1 layer weight matrix
        b: {2x1 cell} containing 2 bias vectors
```

```

methods:

    adapt: Learn while in continuous use
    configure: Configure inputs & outputs
    gensim: Generate Simulink model
    init: Initialize weights & biases
    perform: Calculate performance
    sim: Evaluate network outputs given inputs
    train: Train network with examples
    view: View diagram
    unconfigure: Unconfigure inputs & outputs

evaluate:      outputs = ans(inputs)

```

Baris Kode Program Klasifikasi

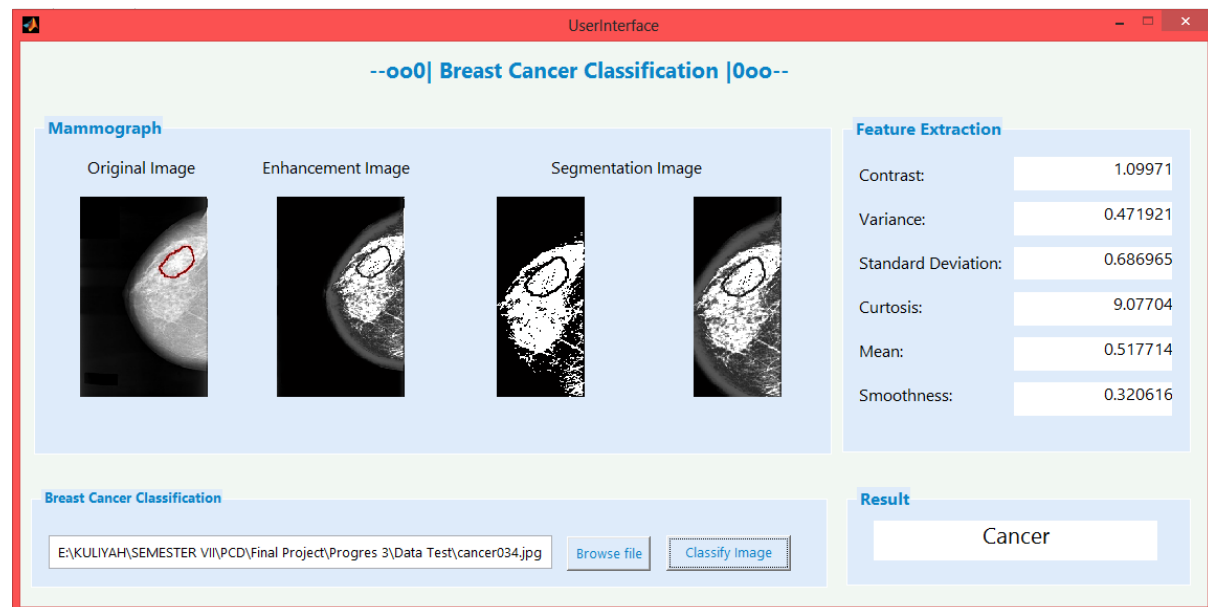
Berikut merupakan kode program tahapan klasifikasi:

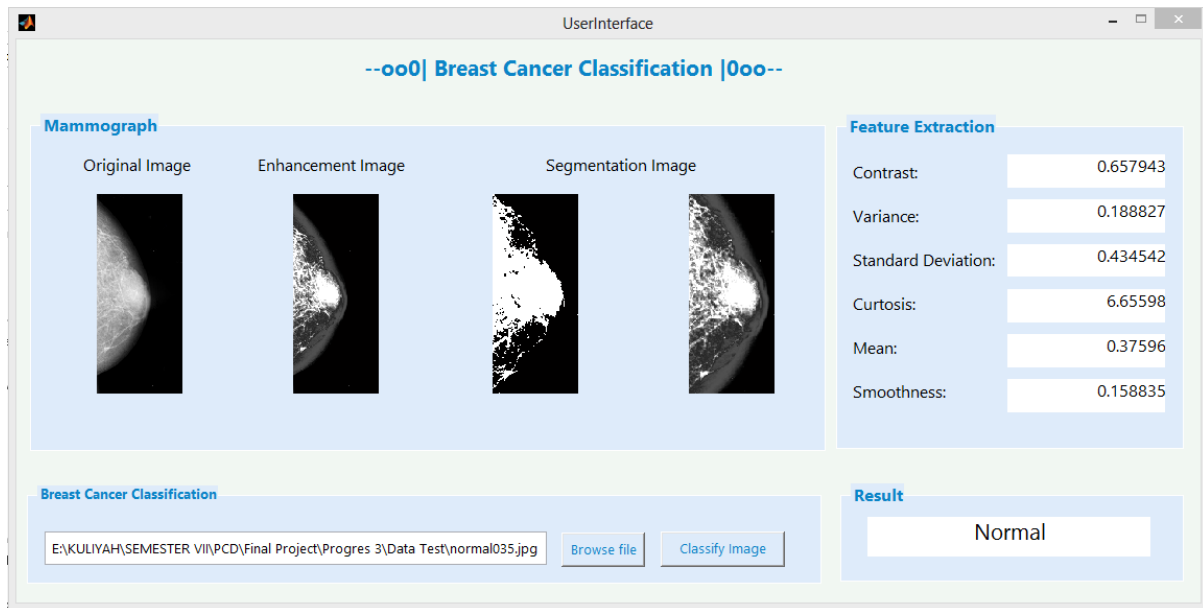
```

function [result,c]=classification(net, new_inputs, new_targets)
% Test Network
outputs = sim(net,new_inputs);
[c,cm] = confusion(new_targets,outputs);
if cm(1,1)==1 || cm(2,1)==1
result='Normal';
elseif cm(1,2)==1 || cm(2,2)==1
result='Cancer';
end

```

Hasil Percobaan Klasifikasi





Kesimpulan

1. Pada paper "A Comparative Study on Digital Mammography Enhancement Algorithms Based on Fuzzy Theory" dijelaskan mengenai peningkatan kualitas citra data mamografi untuk mendeteksi kanker payudara dengan menggunakan 5 (lima) algoritma pendekatan fuzzy. Salah satu algoritma yang digunakan dalam penyelesaian tugas ini adalah algoritma *possibility distribution*. Secara garis besar algoritma ini menurunkan tingkat keabuan piksel yang memiliki nilai keabuan di antara β_1 dan β_2 . Dari hasil program yang dijalankan dapat disimpulkan bahwa algoritma ini tetap mempertahankan nilai yang sangat gelap dan sangat terang dengan pendekatan fuzzy, sedangkan nilai yang berada di sekitar rata-rata (mean) diberi nilai direksi yang berlawanan menuju nilai mean γ . Sehingga hasil yang didapat adalah gambar yang menonjolkan sisi gambar dataset yang sangat terang dan sangat gelap, yang merepresentasikan jaringan kulit dan dapat diolah lebih lanjut untuk menganalisis kanker payudara.
2. Segmentasi menggunakan threshold menemukan bagian yang lebih kecil untuk dipotong dan dianalisis ke tahapan selanjutnya.
3. Pada tahapan ekstraksi fitur, setelah mencoba beberapa fitur, kami menemukan 6 fitur yang merepresentasikan mammogram. Kami mengambil dari GLCM, fitur intensitas, dan fitur histogram intensitas. Dari fitur GLCM kami mengambil nilai *contrast*, dari fitur intensitas kami mengambil nilai *mean*, *variance*, *standard deviation*, dari fitur histogram intensitas kami mengambil *curtosis* dan *smoothness*.
4. Klasifikasi menggunakan algoritma *Neural Network* membagi 6 input dari ekstraksi fitur menjadi dua output (normal dan *cancer*) dengan *hidden layer* berukuran 5. Training data menggunakan 60 data mamogram (30 data normal dan 30 data *cancer*) dan dengan data test 18 data mamogram (8 data normal dan 10 data *cancer*). Dari data test tersebut, 3 data tidak tepat dalam klasifikasi.

Daftar Pustaka

- [1] T. C. J. Astley S M, "Combining cues for mammographic abnormalities Proc," in *1st British Machine Vision Conference Oxford*, UK, 1990.
- [2] A. B. Aboul Ella Hassanien, "A Comparative Study on Digital Mamography Enhancement Algorithms Based on Fuzzy Theory," *Studies in Informatics and Control*, Vols. 12, No.1, pp. 21-31, 2003.
- [3] "DDSM: Digital Database for Screening Mammography," 2006. [Online]. Available: <http://marathon.csee.usf.edu/Mammography/Database.html>. [Accessed 18 Oktober 2013].
- [4] "MathWorks," [Online]. Available: <http://www.mathworks.com/>. [Accessed 12 Desember 2013].
- [5] N. Otsu, "A Threshold Selection Method from Gray-Level Histogram," *Transactions on Systems, Man, and Cybernetics*, Vols. SMC-9 No.1, pp. 62-66, 1979.
- [6] R. Nithya and B. Santhi, "Classification of Normal and Abnormal Patterns in Digital Mammograms for Diagnosis of Breast Cancer," *International Journal of Computer Applications (0975 – 8887)*, Vols. 28-No.6, pp. 21-25, 2011.
- [7] "Documentation Center," [Online]. Available: <http://www.mathworks.com/help/nnet/gs/classify-patterns-with-a-neural-network.html>. [Accessed 24 Desember 2013].