

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan proses penguraian suatu sistem informasi ke dalam komponen-komponen yang lebih sederhana yang bertujuan untuk mengidentifikasi serta mengevaluasi masalah-masalah dan hambatan-hambatan yang mungkin terjadi serta kebutuhan-kebutuhan yang diharapkan menjadi acuan yang dapat mengarah kepada solusi yang terbaik. Pada analisis sistem akan dijelaskan beberapa macam analisis yang berhubungan dengan aplikasi yang akan dibangun seperti analisis masalah, analisis penggunaan, analisis metode yang digunakan, analisis kebutuhan fungsional dan non fungsional.

3.1.1 Analisis Masalah

Berdasarkan dari identifikasi masalah yang sudah didapat, maka akan dijabarkan secara lebih terperinci tentang masalah yang ditemukan dalam penelitian ini. Berikut adalah masalah yang ditemukan:

1. Dengan penggunaan metode *learning vector quantization* dalam penelitian ini bagaimana metode ini dapat mengenali karakter seseorang berdasarkan citra tulisan tangan.
2. Akurasi dari metode *learning vector quantization* lebih baik dibandingkan metode *backpropagation* dalam penelitian pengenalan pola citra pada barcode [3], dalam penelitian ini masalah yang diangkat adalah pengenalan pola tulisan tangan dengan menggunakan metode *learning vector quantization* dengan tujuan mengecek bagaimana akurasi metode *learning vector quantization* dalam penelitian ini.

3.1.2 Analisis Aplikasi

Aplikasi pengenalan tulisan tangan adalah aplikasi simulasi dari tulisan tangan yang menghasilkan karakter orang berdasarkan fitur tulisan tangan. Aplikasi ini bekerja dengan cara menganalisis citra tulisan tangan dari hasil *scanning*.

Aplikasi simulasi ini mengimplementasikan metode *learning vector quantization*, citra dari tulisan tangan tersebut akan dilatih dan juga diuji, dan kemudian akan menghasilkan karakter dari orang yang tulisan tangannya sudah dianalisis sebagai hasil akhirnya.

3.2 Analisis Penyelesaian Masalah

Dalam menyelesaikan permasalahan ini, maka aplikasi simulasi ini dibangun. Aplikasi ini bekerja dengan cara memasukan citra dari hasil *scanning* yang akan diolah dan menghasilkan karakter orang berdasarkan dari analisis tulisan tangan yang terdapat dalam aplikasi.

3.2.1 Analisis Data Masukan

Data masukan yang dibutuhkan untuk aplikasi ini adalah sebuah citra. Citra dapat berupa hasil dari *scanning*. Citra yang digunakan untuk aplikasi ini berformat ‘.jpg’ atau juga ‘.png’. Citra yang dijadikan data masukan ini akan dibagi menjadi beberapa kelas berdasarkan hasil analisis yang dilakukan oleh aplikasi. Tabel 3.1 berikut adalah kelas-kelas yang tersedia untuk data citra masukan, data citra masukan akan dianalisis dan dimasukan ke dalam kelas yang tersedia: [1]

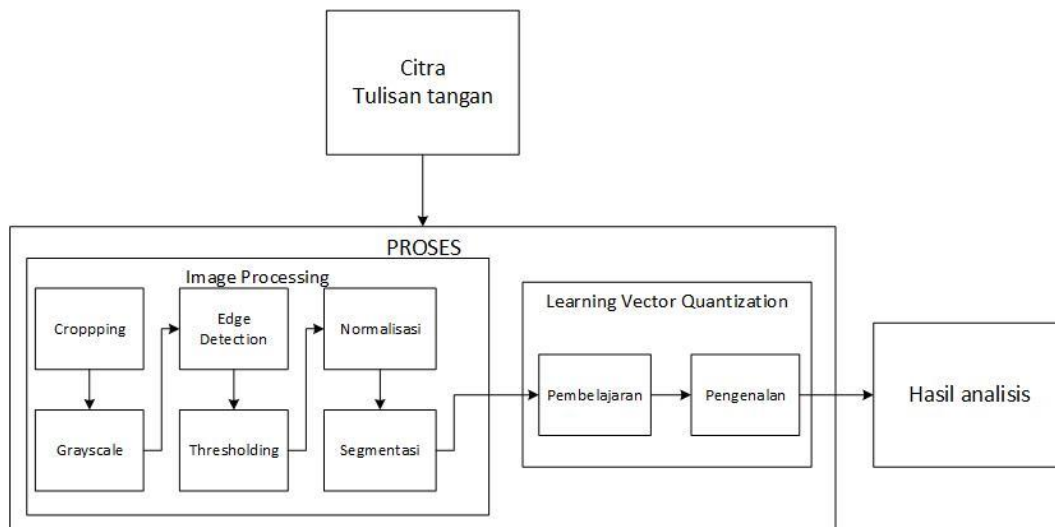
Tabel 3.1 Kelas Data Masukan

Kelas	Fitur	Singkatan	Karakter
1	Antar Baris : Lebar Antar Kata : Lebar Antar Huruf : Lebar	LLL	Karakter dari orang ini adalah seseorang yang senang berada dibalik layar, merupakan seseorang yang individualis dan penuh kehati-hatian dalam bersosial
2	Antar Baris : Lebar Antar Kata : Lebar Antar Huruf : Kecil	LLK	Karakter dari orang ini adalah seseorang yang senang berada dibalik layar, merupakan seseorang yang individualis tetapi terbuka dalam bersosial
3	Antar Baris : Lebar Antar Kata : Kecil Antar Huruf : Lebar	LKL	Karakter dari orang ini adalah seseorang yang senang berada dibalik layar, kemudian orang yang

			memiliki jiwa sosial tinggi tetapi penuh kehati-hatian dalam bersosial
4	Antar Baris : Lebar Antar Kata : Kecil Antar Huruf : Kecil	LKK	Karakter dari orang ini adalah seseorang yang senang berada dibalik layar tetapi mudah bersosial dan juga terbuka dalam bersosial
5	Antar Baris : Kecil Antar Kata : Lebar Antar Huruf : Lebar	KLL	Karakter dari orang ini adalah seseorang yang senang terlibat langsung dalam pekerjaan, merupakan seseorang yang individualis tetapi terbuka dalam bersosial
6	Antar Baris : Kecil Antar Kata : Lebar Antar Huruf : Kecil	KLK	Karakter dari orang ini adalah seseorang yang senang terlibat langsung dalam pekerjaan tetapi merupakan seseorang yang individualis dan tertutup dalam bersosial
7	Antar Baris : Kecil Antar Kata : Kecil Antar Huruf : Lebar	KKL	Karakter dari orang ini adalah seseorang yang senang terlibat langsung dalam pekerjaan dan mudah dalam bersosial tetapi seseorang yang tertutup
8	Antar Baris : Kecil Antar Kata : Kecil Antar Huruf : Kecil	KKK	Karakter dari orang ini adalah seseorang yang senang terlibat langsung dalam pekerjaan mudah bersosial dan terbuka dalam bersosial

3.2.2 Analisis Proses

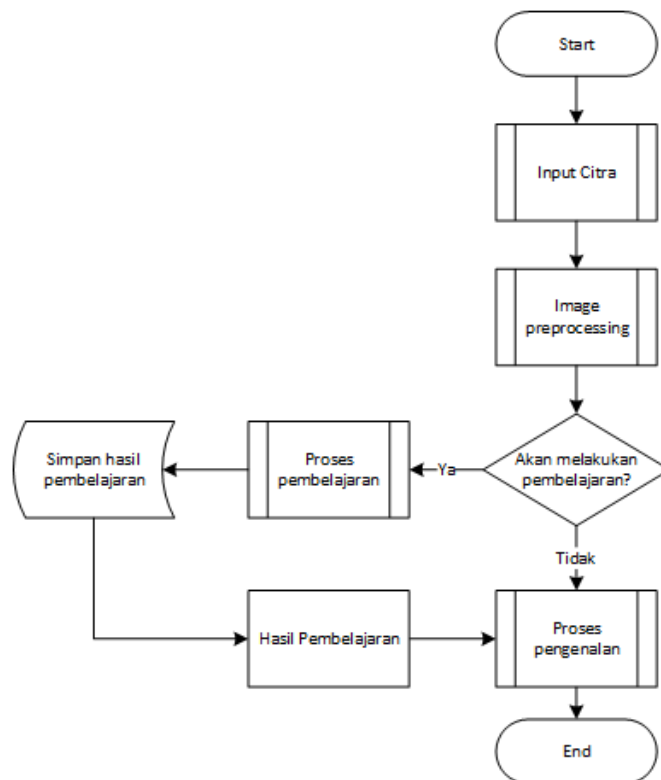
Secara umum aplikasi memiliki beberapa proses dalam melakukan analisis terhadap citra yang dimasukkan kedalam aplikasi. Setiap proses memiliki peran masing-masing untuk dapat mengenali citra yang dimasukkan. Gambar 3.1 dapat dilihat pada gambar berikut:



Gambar 3.1 Proses-proses Pada Simulator

Gambar 3.1 adalah proses-proses yang terjadi pada simulator yang dibuat, terdapat dua proses utama yaitu proses *image processing* dan proses *learning vector quantization* yang terdiri dari proses pembelajaran dan pengenalan kemudian menghasilkan hasil analisis.

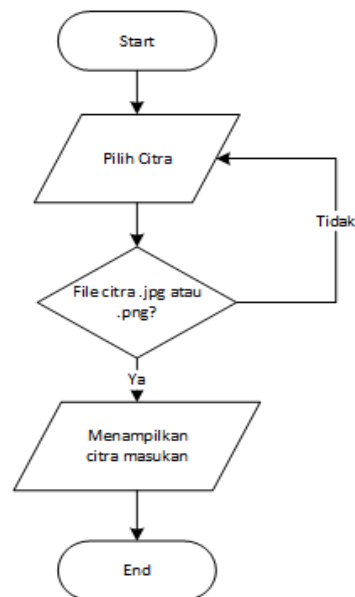
Gambar 3.2 berikut adalah rencana berjalannya aplikasi mulai dari memasukkan citra hingga proses pengenalan dan menampilkan hasil analisisnya:



Gambar 3.2 Flowchart Rencana Aplikasi

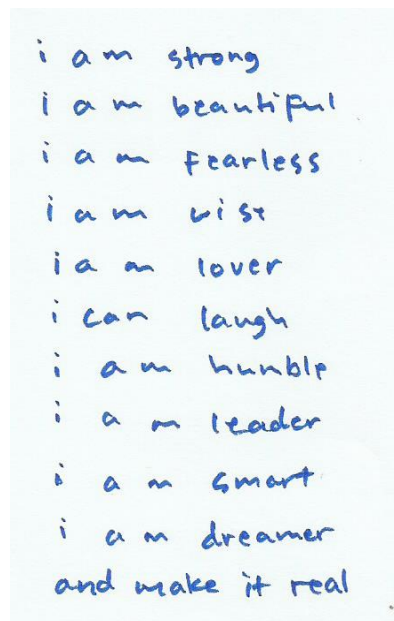
3.2.2.1 Input citra

Proses pengambilan citra yang digunakan sebagai masukan pada sistem yang dibangun adalah citra dari tulisan tangan seseorang yang sudah di *scan* atau di foto yang berformat *.jpg* atau *.png*. Gambar 3.3 adalah *flowchart* dari *input* citra, proses ini menjelaskan cara memasukkan citra hingga citra dapat tampil pada simulator:



Gambar 3.3 Flowchart Proses Input Citra

Gambar 3.4 berikut merupakan citra masukan yang akan digunakan pada proses *input* citra.

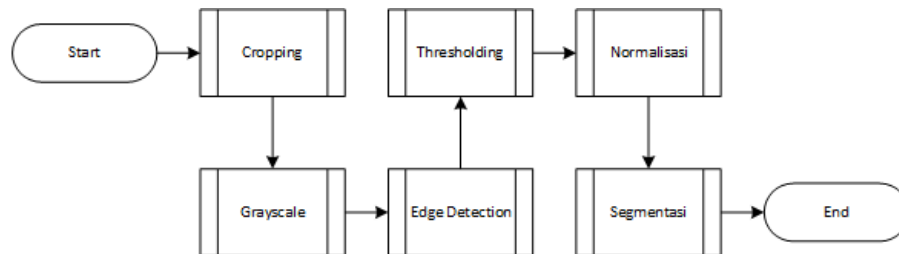


Gambar 3.4 Citra Gambar Masukan

3.2.2.2 Proses Image Preprocessing

Pada proses ini terdapat beberapa beberapa proses pengolahan citra yang dibutuhkan pada simulasi yang akan dibangun mulai dari *scaling*, *grayscale*, *edge detection*, *thresholding*, segmentasi, normalisasi dan ekstraksi ciri. Gambar 3.5

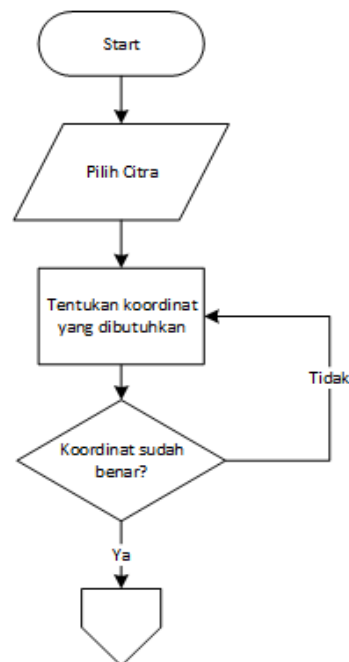
berikut adalah langkah-langkah yang dilakukan pada proses *image processing* mulai dari proses pertama hingga menghasilkan citra hasil dari proses *image processing*:



Gambar 3.5 Flowchart Proses Image Preprocessing

3.2.2.2.1 Cropping

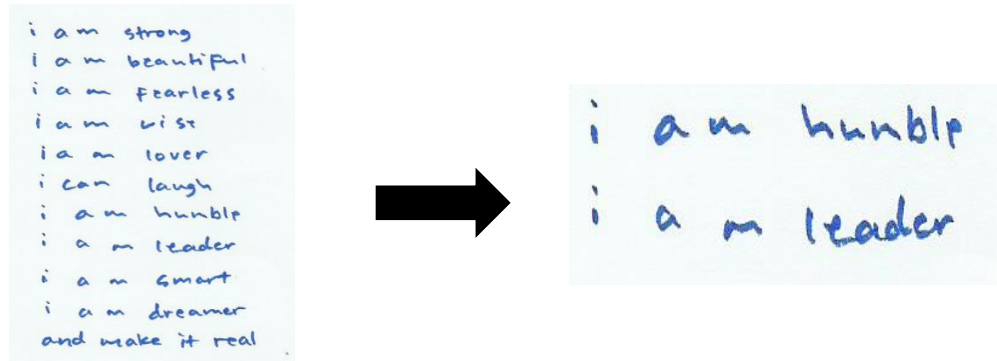
Cropping atau pemotongan citra dilakukan untuk mendapatkan citra yang digunakan sebagai data masukan. Gambar 3.6 berikut adalah *flowchart* dari proses pemotongan citra untuk mendapatkan potongan citra yang akan digunakan dalam sistem:



Gambar 3.6 Flowchart Proses Pemotongan Citra

Dari citra awal yang terdapat beberapa baris kalimat yang digunakan hanya 2/3 kalimat dari total baris yang ada, kalimat yang digunakan bukan pada awal kalimat karena tulisannya masih kontrol dan tulisan masih belum merupakan proyeksi dari penulis dan bukan kalimat terakhir yang digunakan karena tulisan

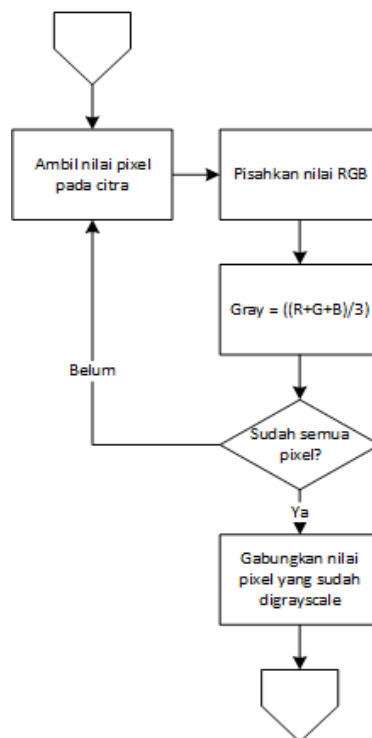
tersebut penulis sudah kelelahan menulis tulisannya. Gambar 3.7 berikut adalah citra hasil pemotongan dari citra masukan:



Gambar 3.7 Citra Hasil Pemotongan

3.2.2.2.2 Grayscale

Proses image processing adalah pengolahan citra yang normal (RGB) menjadi citra *grayscale*. Gambar 3.8 berikut adalah *flowchart* dari proses perubahan citra masukan menjadi citra dengan nilai *grayscale*:



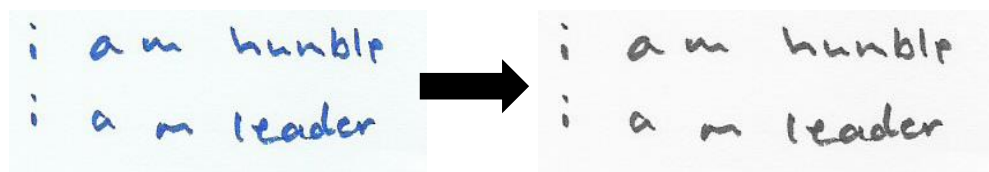
Gambar 3.8 Flowchart Proses Grayscale

Pengubahan gambar ke dalam bentuk *grayscale* ini dilakukan dengan mengambil nilai pixel dari satu gambar masukan kemudian dihitung menggunakan Persamaan 2.5.

Berikut adalah alur kerja dari proses *grayscale*:

- Citra dipisahkan berdasarkan warnanya yaitu Red, Green atau Blue.
- Warna-warna yang sudah dipisahkan akan langsung diubah menjadi warna *grayscale*/keabuan.
- Setelah proses selesai maka setiap pixel yang diubah menjadi *grayscale*/keabuan disusun menjadi ukuran pixel yang sudah ditentukan.

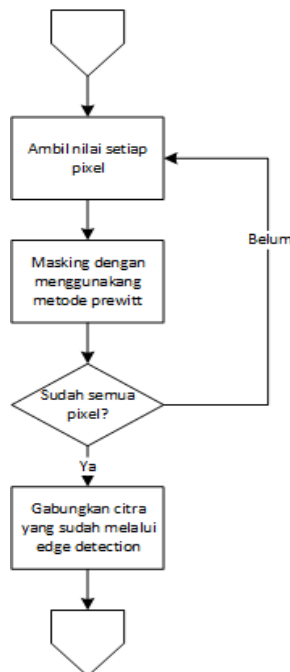
Gambar 3.9 berikut adalah citra hasil proses *grayscale* dari citra masukan yang sudah melalui proses pemotongan:



Gambar 3.9 Citra Hasil Proses Grayscale

3.2.2.2.3 Edge Detection

Gambar 3.10 berikut adalah *flowchart* dari sebuah citra masukan yang diproses untuk mendapatkan citra hasil *edge detection*.



Gambar 3.10 Flowchart Proses Edge Detection

Pada proses ini citra yang digunakan adalah citra berwarna *grayscale*. Proses *edge detection* ini dijalankan dengan menggunakan prewitt, metode prewitt menggunakan dua matriks berukuran 3x3 yaitu matriks Gx dan Gy. Kedua matriks tersebut digunakan untuk menghitung perbedaan warna pada pixel yang sedang dihitung dengan pixel disekitarnya secara horizontal dan vertikal. Matriks Gx digunakan untuk menghitung secara horizontal dan matriks Gy secara matriks vertikal. Gambar 3.11 berikut adalah matriks Gx dan Gy berukuran 3x3 pada metode Prewitt yang digunakan dalam proses deteksi tepi:

-1	0	+1
-1	0	+1
-1	0	+1

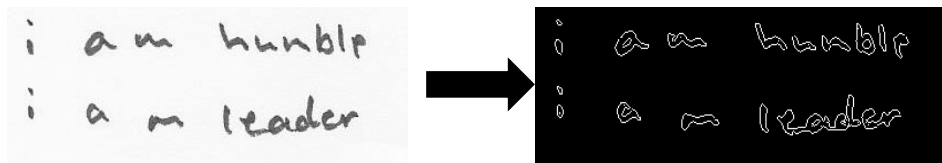
Gx

+1	+1	+1
0	0	0
-1	-1	-1

Gy

Gambar 3.11 Matriks Metode Prewitt

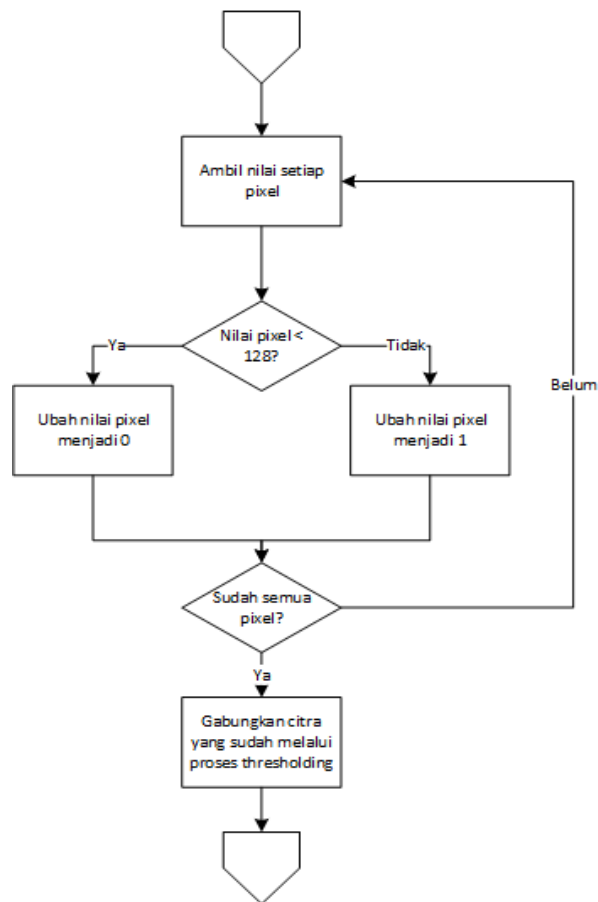
Gambar 3.12 berikut adalah citra hasil *edge detection* dengan menggunakan metode prewitt:



Gambar 3.12 Citra Hasil Edge Detection

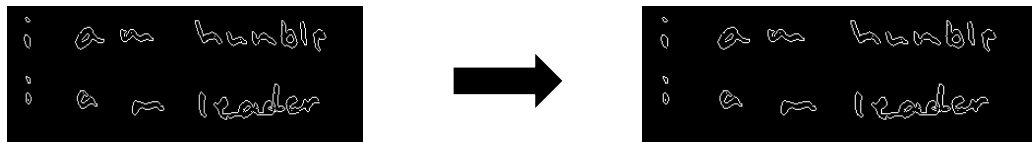
3.2.2.2.4 Thresholding

Thresholding merupakan proses untuk merepresentasikan gambar digital hasil *edge detection* ke dalam bentuk matriks dengan isi 0 atau 1. Nilai 0 dan 1 didapat dari merubah nilai intensitas *threshold* yang sudah ditentukan. Apabila nilai pixel berada di atas sebuah nilai intensitas maka akan diubah menjadi 0 (putih) yang berarti pixel tersebut merupakan *background*, sedangkan jika nilai intensitas berada di bawah batas nilai yang ditentukan maka pixel tersebut akan diubah menjadi berwarna hitam yang berarti dianggap sebuah karakter. Gambar 3.13 berikut adalah *flowchart* dari proses mendapatkan citra hasil *thresholding* dari sebuah citra masukan:



Gambar 3.13 Flowchart Proses Thresholding

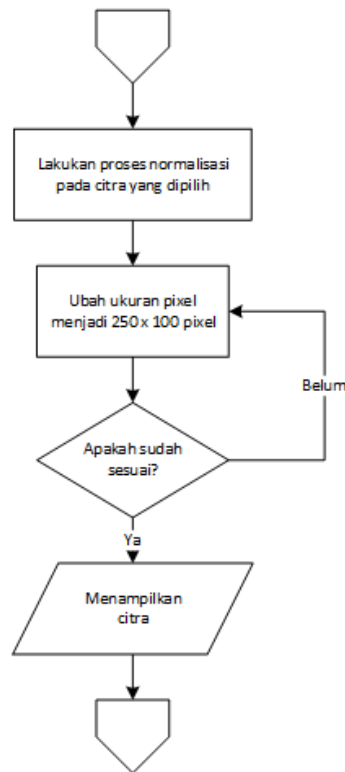
Citra masukan yang hasil *edge detection* akan dilakukan proses *threshold* yang, kemudian akan diubah ke dalam sebuah matriks yang berukuran sesuai dengan ukuran pixel citra masukan. Selanjutnya akan dilakukan perubahan ke dalam vektor matriks biner yang hanya bernilai 0 dan 1 pada setiap pixelnya. Gambar 3.14 berikut adalah hasil dari proses *thresholding*:



Gambar 3.14 Citra Hasil Thresholding

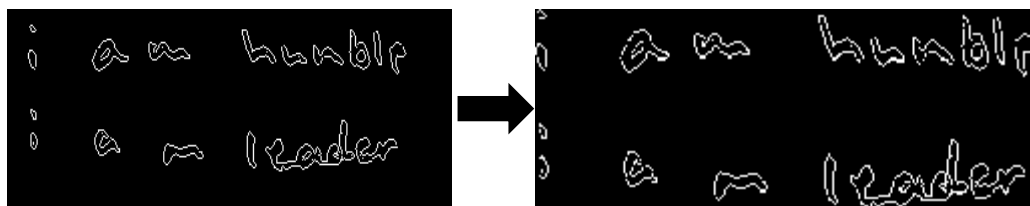
3.2.2.2.5 Normalisasi

Normalisasi adalah proses penyeragaman ukuran citra menjadi ukuran yang sama. Gambar 3.15 berikut adalah *flowchart* dari proses normalisasi sebuah citra dalam sistem:



Gambar 3.15 Flowchart Proses Normalisasi

Karena data yang dihasilkan dari proses segmentasi dapat berbeda-beda dimensinya. Oleh karena itu, untuk mendapatkan data yang seragam, akurat dan konsisten dari setiap sampel, data citra hasil segmentasi akan dinormalisasikan menjadi citra gambar dengan ukuran 250 x 100 pixel. Gambar 3.16 berikut adalah hasil dari proses normalisasi citra masukan:



Gambar 3.16 Hasil Proses Normalisasi

3.2.2.2.6 Segmentasi

Segmentasi merupakan salah satu proses yang sangat penting dalam pengenalan karakter. Karena dalam proses ini simulasi memisahkan setiap fitur spasi antar kata, antar huruf dan juga antar baris. Gambar 3.17 berikut adalah citra masukan setelah proses *image processing* lain yang akan digunakan untuk proses segmentasi:

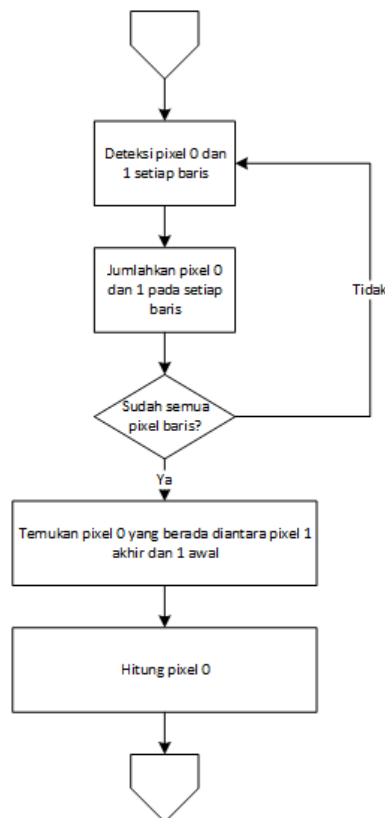


Gambar 3.17 Citra Proses Segmentasi

Dalam proses segmentasi, terdapat beberapa proses segmentasi, yaitu sebagai berikut: [14]

a. Segmentasi Baris (*Line Segmentation*)

Proses yang dilakukan pada tahap segmentasi baris ini adalah dengan memetakan keseluruhan pixel citra yang berisi rangkaian karakter pada sumbu-y. Gambar 3.18 berikut adalah *flowchart* dari proses segmentasi baris untuk mendapatkan jarak setiap baris:

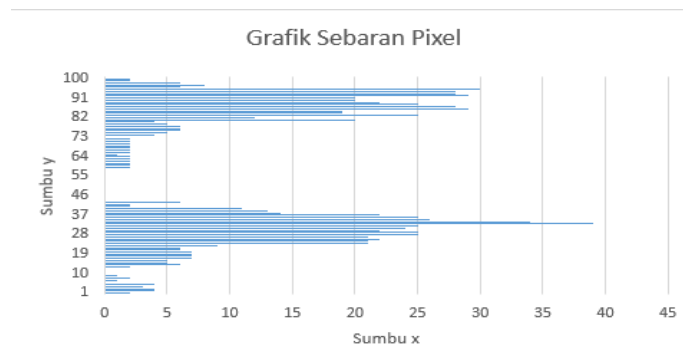


Gambar 3.18 Flowchart Segmentasi Baris

Hal ini dilakukan dengan cara mendeteksi pixel bernilai 0 dan 1 setiap baris kemudian dilakukan penjumlahan setiap pixel 0 dan 1 pada setiap barisnya, setelah mendapatkan baris yang memiliki pixel 1 dan maka akan dilakukan pencarian pixel

0 yang berada diantara pixel 1 kemudian akan dijumlahkan nilai pixel 0 yang terdeteksi jika pixel 0 pada baris ≤ 20 maka jaraknya “KECIL” dan jika lebih adalah “LEBAR”.

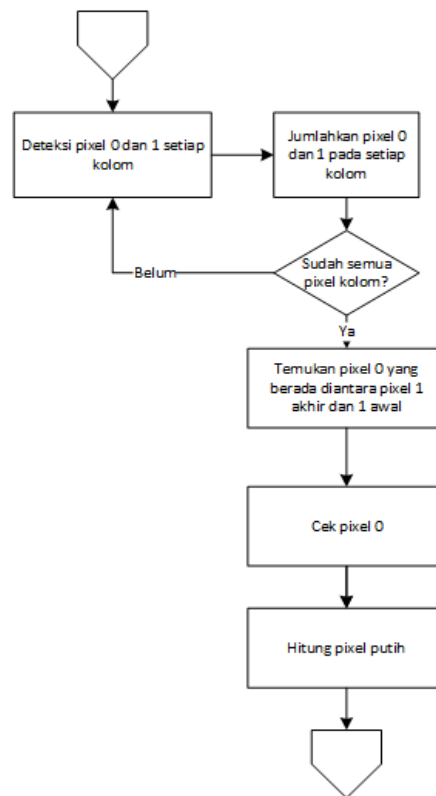
Gambar 3.19 berikut adalah grafik dari sebaran nilai pixel 0. Sumbu y menunjukkan baris yang terdapat pada citra masukan, sedangkan sumbu x menunjukkan jumlah pixel 0 pada setiap baris dari total pixel berukuran 250 x 100 pixel.



Gambar 3.19 Sebaran Pixel Perbaris

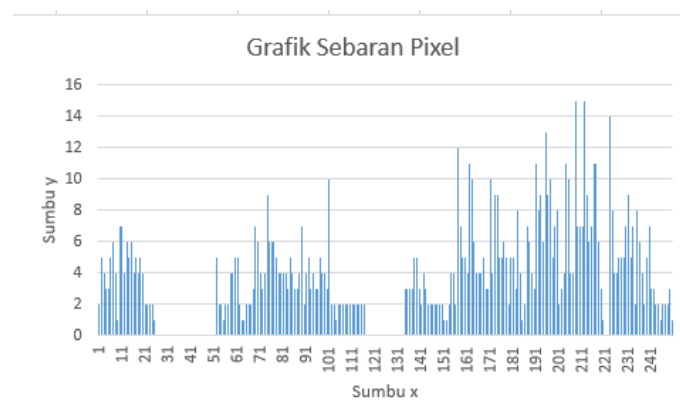
b. Segmentasi Kata (*Word Segmentation*)

Data citra masukan akan dipetakan seluruhnya pada sumbu-x dan dilakukan pendeteksian pixel 0 dan 1. Seperti pada tahap segmentasi baris, tetapi untuk mendapatkan kata setiap baris akan dilihat jumlah nilai pixel 0. Gambar 3.20 adalah *flowchart* dari proses segmentasi kata untuk mendapatkan jarak setiap kata:



Gambar 3.20 Flowchart Segmentasi Kata

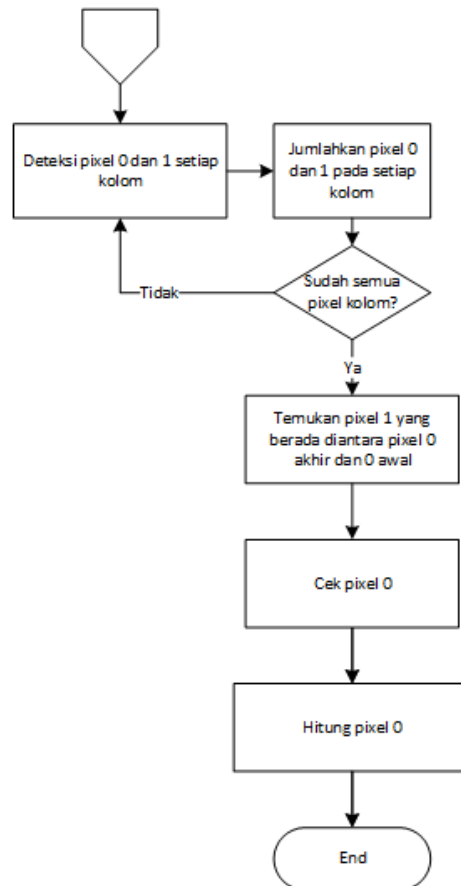
Untuk mendapatkan kata dalam satu baris akan dilakukan pengecekan jumlah pixel 0, jika sebaran nilai pixel 0 diantara pixel 1 ≤ 12 , maka itu dikatakan sebagai kata. Kemudian jika nilai pixel 0 ≤ 10 maka jaraknya “KECIL” dan jika lebih adalah “LEBAR”. Gambar 3.21 grafik dari sebaran nilai pixel 0. Sumbu x menunjukkan kolom yang terdapat pada citra masukan, sedangkan sumbu y menunjukkan jumlah pixel 0 pada setiap kolom dari total pixel berukuran 250 x 100 pixel.



Gambar 3.21 Sebaran Pixel Perkolom

c. Segmentasi Huruf

Proses segmentasi huruf adalah proses untuk mendapatkan jarak pada fitur spasi antar huruf. Gambar 3.22 berikut adalah *flowchart* dari proses segmentasi huruf untuk mendapatkan jarak setiap huruf:

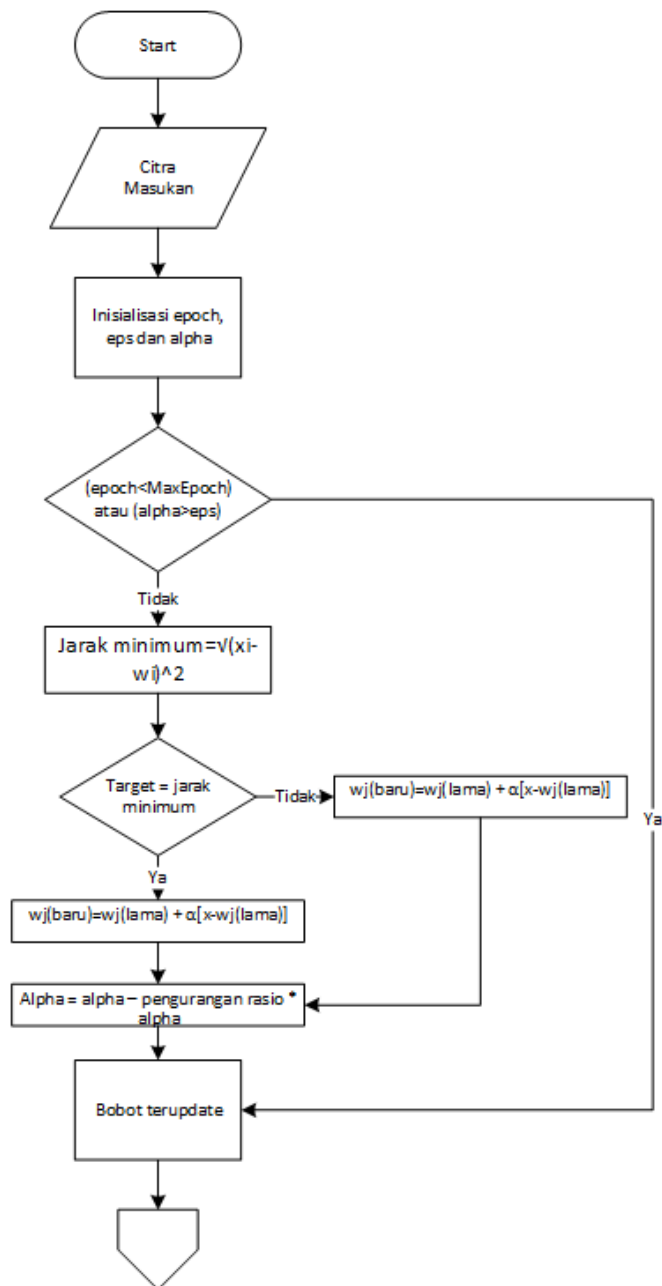


Gambar 3.22 Flowchart Segmentasi Huruf

Proses segmentasi yang dilakukan sama dengan segmentasi kata, tetapi yang berbeda adalah nilai pixel $0 \leq 6$. Kemudian jika nilai pixel $0 \leq 6$ maka jaraknya “KECIL” dan jika lebih adalah “LEBAR”. Hasil dari proses ini adalah mendapatkan jarak antar huruf.

3.2.2.3 Proses Pembelajaran Learning Vector Quantization

Analisis metode yang dilakukan dalam penelitian adalah meneliti bagaimana kerja dari metode *learning vector quantization* pada simulasi yang akan dibangun. Berikut adalah gambar *flowchart* dari proses pembelajaran *learning vector quantization*:





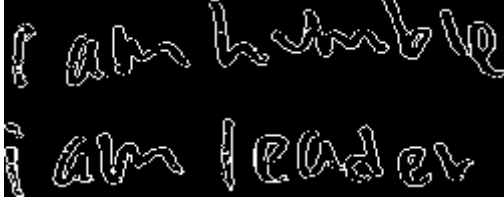


Gambar 3.23 Flowchart Proses Pembelajaran Learning Vector Quantization




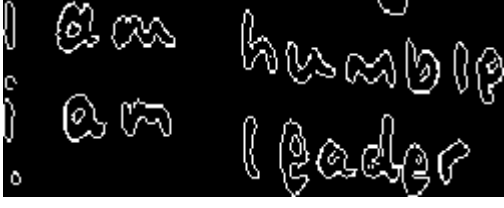
Gambar 3.23 adalah proses pembelajaran metode *learning vector quantization* dimana citra masukan akan dijadikan bobot kemudian akan dihitung menggunakan Persamaan 2.1 dan mencari jarak terkecil dari vektor masukan, kemudian jarak terkecil bobotnya akan diubah.

Berikut adalah langkah-langkah yang dilakukan dalam proses pembelajaran *learning vector quantization*:

1. Setiap kelas diambil salah satu array vektornya dan dijadikan sebagai inisialisasi bobot (w). Sedangkan array lainnya menjadi data masukan (x). Tabel 3.2 berikut adalah data citra masukan dari setiap kelas yang akan digunakan pada proses *learning vector quantization*, matriks dari data citra masukan terdapat pada Lampiran D:

Tabel 3.2 Data Citra Masukan

Kelas	Citra	
Kelas 1		w_1
Kelas 2		w_2
Kelas 3		w_3
Kelas 4		w_4
Kelas 5		w_5

Kelas 6		W ₆
Kelas 7		W ₇
Kelas 8		W ₈
Kelas 1		X ₁

2. Tentukan nilai maksimum perulangan, eror minimum, rasio pembelajaran dan pengurangan rasio. Berikut nilai awal yang digunakan sebagai berikut:

$$\alpha \text{ (learning rate/rasio pembelajaran)} = 0.1$$

$$\text{Pengurangan rasio} = 0.04$$

$$\text{Eror minimum} = 0.002$$

$$\text{MaxEpoh (Maksimum Perulangan)} = 100$$

3. Perhitungan nomor 4 sampai nomor 6 dilakukan selama perulangan masih lebih kecil dari maksimum perulangan atau rasio pembelajaran masih lebih besar dari eror minimum.
4. Untuk setiap data masukan (x) dihitung terhadap setiap data (w) dengan menggunakan Persamaan 2.1 berikut :

$$j = [1, 2]$$

$$n = 25000 \text{ (didapat dari ukuran pixel citra masukan } 250 \times 100 \text{ pixel)}$$

sehingga persamaan menjadi:

$$C1 = \sqrt{\sum_{i=1}^{25000} (xi - w1i)^2}$$

Perulangan = 1

Data ke-1 (x1)

Jarak pada :

a. Bobot ke-1 (w_1)

$$C1 = \sqrt{((0 - 1)^2 + \dots + (0 - 0)^2 + (0 - 0)^2)} = 48.836$$

b. Bobot ke-2 (w_2)

$$C2 = \sqrt{((0 - 0)^2 + \dots + (0 - 1)^2 + (0 - 0)^2)} = 48.435$$

c. Bobot ke-3 (w_3)

$$C3 = \sqrt{((0 - 1)^2 + \dots + (0 - 0)^2 + (0 - 1)^2)} = 46.227$$

d. Bobot ke-4 (w_4)

$$C4 = \sqrt{((0 - 0)^2 + \dots + (0 - 0)^2 + (0 - 0)^2)} = 48.682$$

e. Bobot ke-5 (w_5)

$$C5 = \sqrt{((0 - 1)^2 + \dots + (0 - 1)^2 + (0 - 0)^2)} = 52.952$$

f. Bobot ke-6 (w_6)

$$C6 = \sqrt{((0 - 0)^2 + \dots + (0 - 1)^2 + (0 - 0)^2)} = 50.050$$

g. Bobot ke-7 (w_7)

$$C7 = \sqrt{((0 - 0)^2 + \dots + (0 - 1)^2 + (0 - 1)^2)} = 46.540$$

h. Bobot ke-8 (w_8)

$$C8 = \sqrt{((0 - 1)^2 + \dots + (0 - 1)^2 + (0 - 0)^2)} = 51.068$$

5. Dilihat jarak (C) terkecil yang diperoleh pada perhitungan terhadap x_1 . Kemudian diperiksa apakah target x_1 sesuai dengan kelas bobot dengan jarak terkecil. Hasil dari perhitungan di atas menunjukkan bahwa jarak terkecil ada pada bobot ke-7. Target x_1 adalah kelas 1 sedangkan kelas bobot ke-7 adalah kelas 7, hal ini berarti target tidak sesuai dengan hasil perhitungan.
6. Dilakukan perhitungan perubahan terhadap bobot dengan jarak terkecil yaitu bobot ke-7, jika kelas target sesuai dengan bobot maka persamaan yang

digunakan Persamaan 2.2 sedangkan jika kelas target tidak sesuai dengan bobot maka persamaan yang digunakan adalah Persamaan 2.3 Dalam contoh perhitungan ini, hasil dari perhitungan menunjukkan bahwa kelas target jarak terkecil tidak sesuai dengan kelas bobot, sehingga perhitungannya adalah sebagai berikut:

$$w_{2,1} = 0 - 0.05 * (0-0) = 0$$

$$w_{2,2} = 0 - 0.05 * (0-0) = 0$$

$$w_{2,3} = 1 - 0.05 * (0-1) = 0$$

...

$$w_{2,24999} = 1 - 0.05 * (1-1) = 0$$

$$w_{2,25000} = 0 - 0.05 * (0-0) = 0$$

Setelah itu hasil perhitungan bobot pada iterasi awal akan ditemukan.

7. Langkah nomor 4 sampai dengan nomor 7 dikerjakan terhadap semua masukan (x_1 dan x_2), setelah selesai dilakukan perhitungan rubah nilai rasio dengan menggunakan Persamaan 2.4 berikut:

$$\alpha = 0.05 - 0.1 * 0.05 = 0.045$$

8. Perhitungan diulang hingga perulangan maksimum dan mendapatkan nilai bobot akhir dan mendapatkan bobot akhir dan dilakukan pembaruan bobot. Hingga iterasi terakhir hasil bobot yang berubah hanya bobot w_2 , w_3 dan w_7 , hasil perhitungan bobot pada iterasi terakhir terdapat pada Lampiran E.

3.2.2.4 Proses Pengenalan Metode Learning Vector Quantization

Proses pengenalan dengan metode *learning vector quantization* menggunakan rumus yang sama dengan perhitungan bobot pada proses pembelajaran.

Berikut langkah-langkah pengenalan:

1. Data *test* dihitung terhadap setiap data bobot (w) dengan Persamaan 2.1:

$$j = [1, 2]$$

$$n = 25000 \text{ (didapat dari ukuran pixel citra masukan } 250 \times 100 \text{ pixel)}$$

$$C1 = \sqrt{\sum_{i=1}^{25000} (xi - W1i)^2}$$

Jarak pada:

a. Bobot ke1 (w_1)

$$C1 = \sqrt{((1 - 1)^2 + \dots + (0 + 0)^2 + (0 + 0)^2)} = 49.2849$$

b. Bobot ke2 (w_2)

$$C2 = \sqrt{((1 - 1)^2 + \dots + (0 + 0)^2 + (0 + 0)^2)} = 43.6320$$

c. Bobot ke3 (w_3)

$$C3 = \sqrt{((1 - 1)^2 + \dots + (0 + 0)^2 + (0 + 0)^2)} = 50.2272$$

d. Bobot ke4 (w_4)

$$C4 = \sqrt{((1 - 1)^2 + \dots + (0 + 0)^2 + (0 + 0)^2)} = 49.8598$$

e. Bobot ke5 (w_5)

$$C5 = \sqrt{((1 - 1)^2 + \dots + (0 + 0)^2 + (0 + 0)^2)} = 53.7401$$

f. Bobot ke6 (w_6)

$$C6 = \sqrt{((1 - 1)^2 + \dots + (0 + 0)^2 + (0 + 0)^2)} = 51.0979$$

g. Bobot ke7 (w_7)

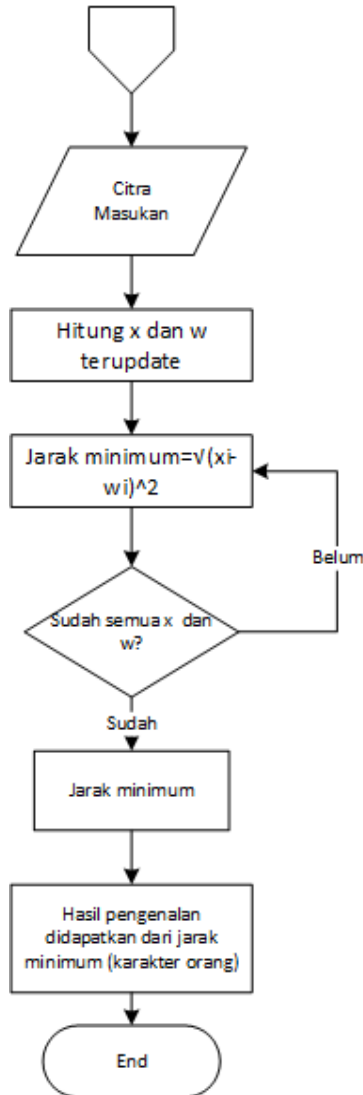
$$C7 = \sqrt{((1 - 1)^2 + \dots + (0 + 0)^2 + (0 + 0)^2)} = 49.5754$$

h. Bobot ke8 (w_8)

$$C8 = \sqrt{((1 - 1)^2 + \dots + (0 + 0)^2 + (0 + 0)^2)} = 52.0192$$

2. Dilihat dari nilai (C) terkecil yang diperoleh pada perhitungan terhadap data *test* . Kelas bobot yang memiliki jarak terkecil menjadi hasil dari proses pengenalan ini. Dalam perhitungan di atas untuk fitur spasi antar kata menghasilkan jarak terkecil ada pada bobot ke-2 (w_2) dan w_2 merupakan kelas 2 (LLK), sehingga hasil dari data *test* adalah “Karakter dari orang ini adalah seseorang yang senang berada dibalik layar, merupakan seseorang yang individualis tetapi terbuka dalam bersosial”.

Gambar 3.24 berikut adalah *flowchart* dari proses pengenalan algoritma *learning vector quantization* dalam mengenali data citra masukan:



Gambar 3.24 Flowchart Proses Pengenalan Learning Vector Quantization

3.3 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional berisi tentang kebutuhan-kebutuhan pendukung yang diperlukan saat pembangunan aplikasi. Berikut adalah kebutuhan-kebutuhan yang dibutuhkan:

3.3.1 Analisis Kebutuhan Perangkat Keras

Pada aplikasi simulasi yang akan dibangun diperlukan perangkat keras untuk membantu pembangunan aplikasi. Adapun perangkat keras yang dibutuhkan adalah:

1. *Processor* : Intel Dual Core 2.0 Ghz
2. *Memory* : RAM DDR3 1 GB
3. *VGA* : 512 GB
4. *Harddisk* : 120 GB
5. *Monitor* : 14 inch
6. *Scanner*
7. *Mouse*
8. *Keyboard*

3.3.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak merupakan kebutuhan yang penting dalam pembangunan aplikasi ini, karena perangkat lunak digunakan sebagai sarana pembangunan aplikasi sampai aplikasi selesai dibangun. Perangkat lunak yang dibutuhkan dalam pembangunan aplikasi ini adalah sebagai berikut:

1. Sistem operasi : Microsoft Windows 7
2. Matlab

3.3.3 Analisis Pengguna

Untuk menjalankan aplikasi yang dibangun dibutuhkan spesifikasi dari pengguna agar dapat menjalankan fungsi-fungsi yang terdapat pada aplikasi. Hanya terdapat jenis pengguna yang dibutuhkan yaitu penguji. Adapun pengguna yang dibutuhkan untuk menjalankan aplikasi adalah sebagai berikut:

1. Bisa mengoperasikan komputer
2. Bisa menggunakan Matlab

3.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan untuk memberikan gambaran aliran data yang ada pada aplikasi yang dibangun yang meliputi diagram konteks, *data flow diagram* (DFD) dan spesifikasi proses.

3.4.1 Diagram Konteks

Diagram konteks (*Context Diagram*) merupakan gambaran umum dari keseluruhan suatu sistem. Gambar 3.25 berikut adalah diagram konteks yang merupakan gambaran umum dari sistem pengenalan karakter tulisan tangan:

3.4.3 Spesifikasi Proses

Untuk menjelaskan lebih lanjut tentang proses-proses yang ada didalam *data flow diagram* dibuatlah spesifikasi proses. Adapun spesifikasi proses dari sistem pengenalan karakter tulisan tangan dapat dilihat pada Tabel 3.3 dibawah ini:

Tabel 3.3 Spesifikasi Proses

No	Proses	Keterangan
1	No. Proses	1
	Nama Proses	Inisialisasi
	<i>Source</i>	Penguji (Pengguna), Citra
	<i>Input</i>	Data citra, Potongan Citra
	<i>Output</i>	Info data citra, Info Potongan Citra, Bobot
	<i>Destination</i>	Pengguna
	Logika Proses	1. Pengguna mengambil citra tulisan tangan dari penyimpanan citra 2. Sistem menampilkan citra 3. Pengguna melakukan pemotongan citra 4. Sistem menampilkan hasil pemotongan citra 5. Sistem merubah data citra hasil pemotongan menjadi bobot 6. Sistem menyimpan bobot
2	No. Proses	2
	Nama Proses	Pembelajaran
	<i>Source</i>	Penguji (Pengguna), Citra
	<i>Input</i>	Data citra, Potongan Citra, Parameter Masukan, Jarak Antar Baris, Jarak Antar Kata, Jarak Antar Huruf, Bobot
	<i>Output</i>	Info citra, Info Potongan Citra, Info Parameter Masukan, Info Jarak Antar Baris, Info Jarak Antar Kata, Info Jarak Antar Huruf, Bobot Baru
	<i>Destination</i>	Pengguna

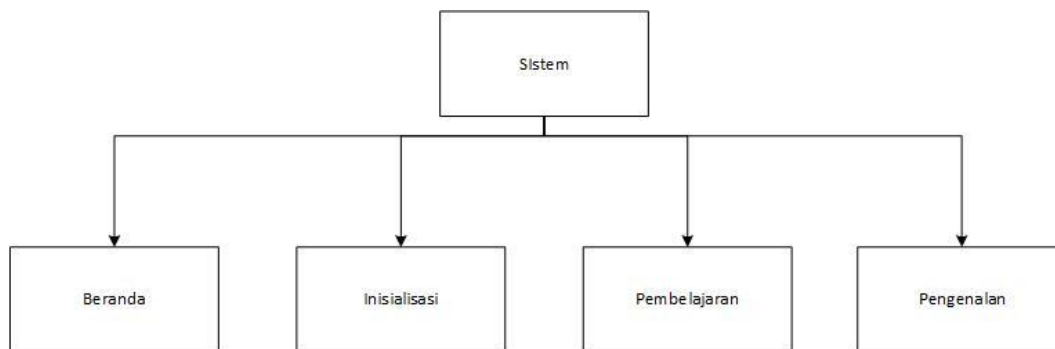
	Logika Proses	1. Pengguna mengambil citra tulisan tangan dari penyimpanan citra 2. Sistem menampilkan citra 3. Pengguna melakukan pemotongan citra 4. Sistem menampilkan hasil potongan citra 5. Pengguna memasukkan jarak antar baris, jarak antar kata dan jarak antar huruf 6. Pengguna memasukkan parameter masukan (learning rate, pengurangan rasio, eror minimum dan maksimum epoch) 7. Bobot hasil inisialisasi dijadikan bobot masukan 8. Bobot baru dihasilkan dari proses pembelajaran
3	No. Proses	3
	Nama Proses	Pengenalan
	<i>Source</i>	Penguji (Pengguna), Citra
	<i>Input</i>	Data citra, Potongan Citra, Bobot Baru
	<i>Output</i>	Info pengenalan, Info citra, Info Potongan Citra
	<i>Destination</i>	Penguji (Pengguna)
	Logika Proses	1. Pengguna mengambil citra tulisan tangan dari penyimpanan citra 2. Sistem menampilkan citra 3. Pengguna melakukan pemotongan citra 4. Sistem menampilkan hasil pemotongan citra 5. Bobot baru digunakan dalam proses pengenalan 6. Sistem akan menampilkan karakter orang

3.5 Perancangan Sistem

Perancangan sistem dimulai setelah tahap analisis terhadap sistem selesai dilakukan. Perancangan sistem dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi.

3.5.1 Perancangan Struktur Menu

Perancangan struktur menu dan submenu yang berfungsi untuk memudahkan pengguna di dalam menggunakan sistem. Gambar 3.27 menjelaskan struktur menu dari simulator yang akan dibuat, dimana beranda, inisialisasi, pembelajaran dan pengenalan memiliki tingkatan level yang sama, berikut ini adalah gambar dari struktur menu yang digunakan:



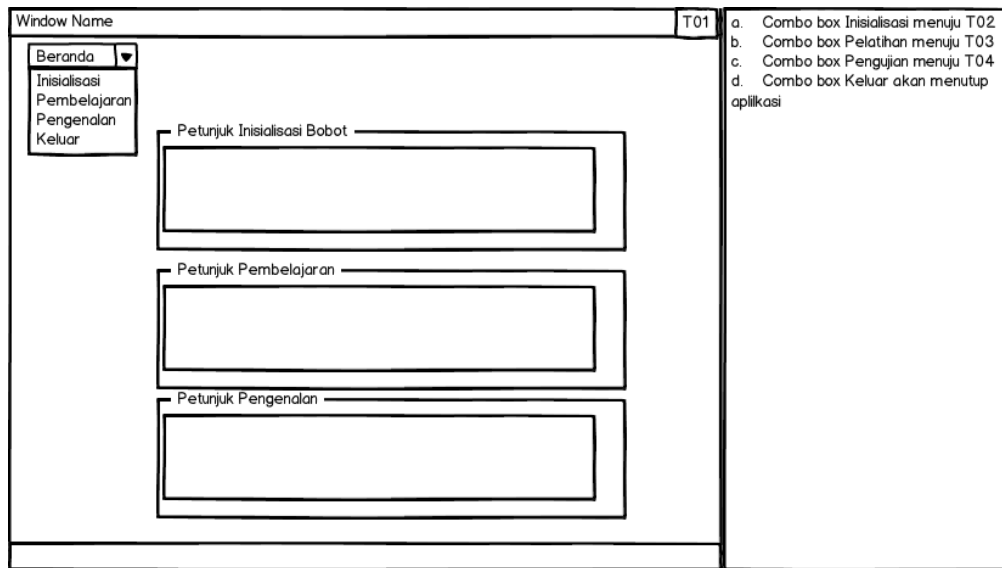
Gambar 3.27 Struktur Menu Aplikasi Pengenalan Karakter Tulisan Tangan

3.5.2 Perancangan Antarmuka

Spesifikasi antarmuka merupakan suatu bentuk tampilan dari aplikasi yang akan dibangun untuk kebutuhan *interface* dengan pengguna. Spesifikasi antarmuka terdiri dari perancangan tampilan menu, tampilan form, keluaran dan jaringan semantik.

3.5.2.1 Perancangan Antarmuka Menu Halaman Utama

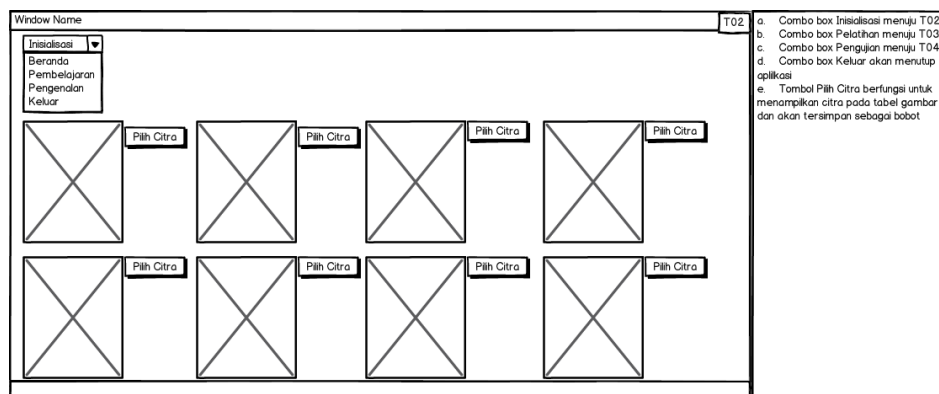
Pada menu halaman utama terdapat *combo box* dan juga terdapat langkah-langkah penggunaan aplikasi. Gambar 3.28 berikut adalah perancangan antarmuka menu halaman utama pada saat sistem dijalankan pertama kali:



Gambar 3.28 Perancangan Antarmuka Menu Tampilan Utama

3.5.2.2 Perancangan Antarmuka Menu Inisialisasi Bobot

Pada menu halaman inisialisasi terdapat *combo box*, tombol pilih citra yang digunakan untuk memasukkan citra ke dalam aplikasi dan tombol simpan digunakan untuk menyimpan citra sebagai bobot awal dalam aplikasi. Gambar 3.29 berikut adalah perancangan antarmuka menu inisialisasi bobot sistem dimana pada menu ini pengguna harus memasukkan data citra untuk menjadikannya bobot awal:



Gambar 3.29 Perancangan Antarmuka Menu Inisialisasi Bobot

3.5.2.3 Perancangan Antarmuka Menu Pembelajaran

Pada menu halaman inisialisasi terdapat *combo box*, tombol pilih citra yang digunakan untuk memasukkan citra ke dalam aplikasi, tombol latih digunakan untuk melatih citra yang digunakan dalam aplikasi, *text edit* pada grup parameter berfungsi untuk memasukkan nilai parameter-parameter yang digunakan pada

analisis dan *radiobutton* digunakan untuk memilih jarak pada setiap fitur. Gambar 3.30 berikut adalah perancangan antarmuka menu pembelajaran sistem dimana menu ini berfungsi untuk melakukan proses pembelajaran terhadap data citra masukan:

Window Name T03

Pembelajaran ▼

Pilih Citra Latih

Parameter

Learning Rate

Pengurangan Rasio

Error Minimum

Maksimum Epoch

Jarak Antar Baris ☐ Lebar ☐ Kecil

Jarak Antar Kata ☐ Lebar ☐ Kecil

Jarak Antar Huruf ☐ Lebar ☐ Kecil

Hasil Perhitungan

Jumlah Perulangan	0	C1	0	C5	0
Jarak Antar Baris (Pixel)	0	C2	0	C6	0
Jarak Antar Kata (Pixel)	0	C3	0	C7	0
Jarak Antar Huruf (Pixel)	0	C4	0	C8	0
Target Kelas	0				

a. Combo box Inisialisasi menuju T02
b. Combo box Pelatihan menuju T03
c. Combo box Pengujian menuju T04
d. Combo box Keluar akan menutup aplikasi
e. Tombol Pilih Citra berfungsi untuk menampilkan citra pada tabel gambar
f. Tombol Latih berfungsi untuk melatih citra masukan
g. Text Edit pada grup parameter berfungsi untuk nilai parameter masukan
h. Radiobutton digunakan untuk memilih jarak setiap fitur

Gambar 3.30 Perancangan Antarmuka Menu Pembelajaran

3.5.2.4 Perancangan Antarmuka Menu Pengujian

Pada menu halaman inisialisasi terdapat *combo box*, tombol pilih citra yang digunakan untuk memasukkan citra ke dalam aplikasi dan tombol uji digunakan untuk menguji citra yang digunakan dan menampilkan karakter orang dari tulisan yang sudah dianalisis. Gambar 3.31 berikut adalah perancangan antarmuka menu pelatihan sistem yang berfungsi untuk melakukan pengujian terhadap data citra masukan dan menghasilkan karakter orang setelah dianalisis:

Window Name T04

Pengujian ▼

Beranda
Pembelajaran
Keluar

Pilih Citra Uji

Hasil Analisis

a. Combo box Inisialisasi menuju T02
b. Combo box Pelatihan menuju T03
c. Combo box Pengujian menuju T04
d. Combo box Keluar akan menutup aplikasi
e. Tombol Pilih Citra berfungsi untuk menampilkan citra pada tabel gambar dan akan menampilkan hasil analisis
f. Tombol Latih berfungsi untuk melatih citra masukan

Gambar 3.31 Perancangan Antarmuka Menu Pengujian

3.5.2.5 Perancangan Pesan

Pada Tabel 3.4 berikut adalah perancangan pesan yang terdapat pada simulator, hanya ada pesan yang akan tampil yaitu “Tolong Pilih Citra” dan “Data Berhasil Disimpan” seperti berikut:

Tabel 3.4 Perancangan Pesan

Kode Pesan	Pesan	No.Form
M01	Tolong Pilih Citra	T02, T03, T04
M02	Data Berhasil Disimpan	T02

Berikut adalah gambar perancangan pesan M01 yang dibuat:

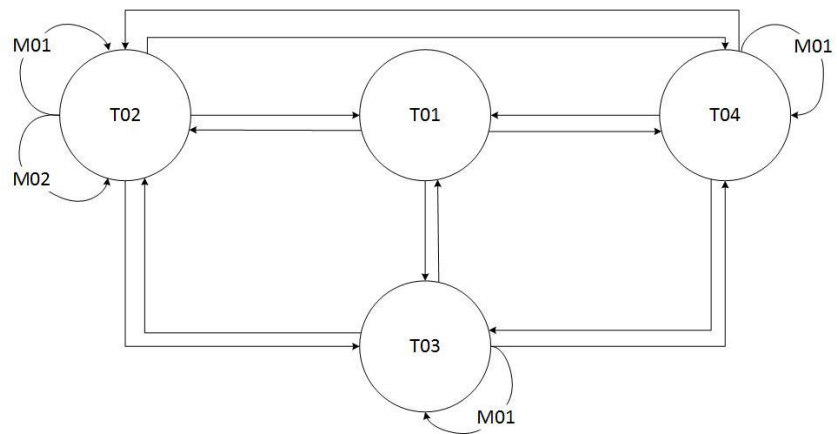
Gambar 3.32 Perancangan Pesan Tolong Pilih Citra

Berikut adalah gambar perancangan pesan M02 yang dibuat:

Gambar 3.33 Perancangan Pesan Data Berhasil Disimpan

3.5.2.6 Jaringan Semantik

Jaringan semantik adalah gambaran yang menunjukkan hubungan antar objek yang terdapat pada sistem. Gambar 3.34 berikut adalah jaringan semantik yang menjelaskan hubungan dari setiap menu yang terdapat pada; sistem pengenalan karakter melalui tulisan tangan orang:



Gambar 3.34 Jaringan Semantik