© HAK CIPTA MILIK UNIVERSITAS ANDALAS

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
   a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
   b. Pengutipan tidak merugikan kepentingan yang wajar Unand.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Unand.

SISTEM PENGENALAN PLAT NOMOR KENDARAAN DENGAN MENGGUNAKAN EKSTRAKSI CIRI INTEGRAL PROYEKSI DAN PENGENALAN DENGAN JARINGAN SYARAF TIRUAN BACKPROPAGATION

SKRIPSI

ANTON JEFFRY HIDAYAT
NIM. 06 1750 61

FAKULTAS TEKNIK
UNIVERSITAS ANDALAS
PADANG 2012

ABSTRACT

Detection system of vehicle police number plate is one of computer based applications which enable to proceed data management of vehicle police number. It is applied in public parking-lot management of a place, like office building, shopping centre or likewise, where some duty officers are responsible for recording in the system vehicles in and out. This is an automatic system to help out parking-lot management. An engineering method can be a basic knowledge to indicate detection pattern by using backpropagation neural imitation networks.

This Final Project consists of two phases: training and testing. In training phase, it captures characters (letters and numbers) imaging, conversion of RGB images into the intensity image. Intensity images will be segmented by application of bilevel luminance thresholding method. The next steps are labeling and filtering, and extracting types by application of integral projections method. The out put of this extracting will be trained by neural network with backpropagation algorithm. The end result of the whole training phase will be tested. It is expected to recognize vehicle police numbers both alphabetics (A-Z) and numerics (0-9) characters. This step will be varying parameters of neural network: goal value, sum of epoch, and threshold value.

The test result indicate parameters of networks which contribute maximal level of capacity of this Final Project has both tolerance value of 0.008 and threshold value of 0.4. The acceptable success percentage of recognition of neural networks training with architecture and optimal networks parameters is 60%.

Key words: recognition of vehicle police number, integral projection, backpropagation neural network.

UNTUK KEDJAJAAN BANGSA

ABSTRAK

Sistem deteksi plat nomor kendaraan merupakan salah satu aplikasi dalam ilmu komputer yang dapat membantu proses pengolahan data plat nomor kendaraan. Praktiknya dapat ditemukan dalam suatu area parkir sebuah gedung instansi atau pertokoan, selalu ditemukan petugas pintu masuk parkir yang bertugas mencatat plat nomor kendaraan. Dengan menerapkan kondisi tersebut, maka diharapkan adanya suatu sistem pencatat plat nomor kendaraan secara otomatis,Salah satu metode yang dapat digunakan untuk pengenalan pola deteksi tersebut adalah dengan menggunakan jaringan syaraf tiruan backpropagation.

Pada penelitian ini terdapat dua tahap yaitu tahap pelatihan dan tahap pengujian. Pada tahap pelatihan terdapat proses capture, merupakan pengambilan gambar citra huruf, selanjutnya proses konversi citra huruf RGB ke citra intensitas. Citra intensitas ini selanjutnya akan disegmentasi menggunakan metode bilevel luminance thresholding. Proses selanjutnya adalah pelabelan dan filtering, dan dilanjutkan proses ekstraksi ciri menggunakan metode Integral Proyeksi. Hasil ekstraksi ini akan dilatih oleh jaringan syaraf tiruan menggunakan algoritma backpropagation. Dari hasil pelatihan ini akan dilajutkan proses pengujian. Hasil yang diharapkan mampu mengenali plat nomor kendaraan bermotor yang terdiri dari semua huruf A-Z dan angka 0-9. Percobaan dilakukan dengan memvariasikan parameter jaringan syaraf tiruan berupa nilai toleransi, jumlah epoch dan nilai threshold.

Hasil pengujian menunjukkan parameter jaringan yang memberikan kerja paling optimal dari penelitian ini adalah nilai toleransi 0.008 dan nilai threshold 0.4. Persentase pengenalan dari pelatihan jaringan dengan arsitektur dan parameter jaringan optimal adalah 60%.

Kata kunci: pengenalan plat nomor kendaraan, integral proyeksi, jaringan syaraf tiruan backpropagation

UNTUK KEDJAJAAN BANGSA

DAFTAR ISI

ABSTRAK i
DAFTAR ISI ii
DAFTAR GAMBAR vi
DAFTAR TABEL viii
DAFTAR ISTILAH ix
DAFTAR LAMPIRAN xii
BAB I PENDAHULUAN
1.1 Latar Belakang 1
1.2 Tujuan Penelitian 3
1.3 Manfaat Penelitian 3
1.4 Batasan Masalah 4
1.5 Sistematika Penulisan 4
BAB II TINJAUAN PUSTAKA
2.1 Video 6
2.1.1 Pengertian Video 6
2.1.2 *Capture* (Proses Pengambilan Video) 6
2.2 Citra 7
2.2.1 Pengertian citra 7
2.2.2 Pembentukan citra 7
2.2.3 Digitalisasi Citra 8
2.2.4. Histogram citra 10
2.3 Sistem Komponen Ruang Warna 11
2.3.1 RGB (*Red Green Blue*) 11
2.3.2 Kombinasi Warna 12
2.4 Segmentasi Citra 13
2.4.1 Definisi Segmentasi 13
2.4.2 *Bilevel Luminance Thresholding* 13
2.5 Pelabelan dan Filtering 14
2.6 Integral Proyeksi 15
2.7 Jaringan syaraf tiruan 17

2.7.1 Definisi jaringan syaraf tiruan 17
2.7.2 Kelebihan jaringan syaraf tiruan 18
2.7.3 Karakteristik jaringan syaraf tiruan 19
2.7.4 Model dasar jaringan syaraf tiruan 19
2.7.5 Arsitektur jaringan 19
2.7.5.1 Jaringan layar jamak *(multi layer network)* 19
2.7.6 Bias dan *threshold* 20
2.7.7 Pelatihan dengan dan tanpa Supervisi 21
2.7.8 Model jaringan syaraf tiruan 21
2.7.8.1 Model jaringan syaraf tiruan *Backpropagation* 21
2.7.8.1.1 Arsitektur *Backpropagation* 22
2.7.8.1.2 Fungsi Aktivasi 22
BAB III METODOLOGI PENELITIAN
3.1 Jenis Penelitian 24
3.2 Objek Penelitian 24
3.3 Desain Penelitian 26
3.4 Teknik Analisa Sistem 28
3.5 Prosedur Penelitian 28
BAB IV PERANCANGAN SISTEM
4.1 Blok diagram sistem 30
4.1.1 Citra huruf 32
4.1.2 *Capture* 32
4.1.3 Konversi ke citra intensitas 33
4.1.4 Segmentasi 34
4.1.5 Pelabelan dan *filtering* 34
4.1.6 Ekstraksi ciri 35
4.1.7 Pelatihan jaringan syaraf tiruan *backpropagation* 35
4.1.8 Pengujian sistem 35
4.2 Algoritma program 36
4.2.1 Segmentasi 36
4.2.2 Pelabelan dan *filtering* 36
4.2.3 Ekstraksi ciri Integral Proyeksi 37

4.2.4 Jaringan syaraf tiruan *backprogation* 38
4.3 Visualisasi sistem 40
4.3.1 Pengambilan data 42
4.3.2 Proses merubah citra RGB ke citra intensitas 43
4.3.3 Proses segmentasi 44
4.3.4 Proses *cropping*, *filtering* dan ekstraksi ciri 44
4.3.5 Proses pelatihan jaringan syaraf tiruan *backprogation* 45
4.3.6 Pengujian sistem 45
4.3.7 Mengakhiri program 46
4.4 Skenario percobaan 47
BAB V HASIL DAN PEMBAHASAN
5.1 Hasil percobaan 48
5.1.1 Percobaan pengujian sistem dengan jumlah neuron pada layer tersembunyi pertama 50 dan nilai toleransi 0.007 48
5.1.2 Percobaan pengujian sistem dengan jumlah neuron pada layer tersembunyi pertama 50 dan nilai toleransi 0.008 49
5.1.3 Percobaan pengujian sistem dengan jumlah neuron pada layer tersembunyi pertama 50 dan nilai toleransi 0.009 50
5.1.4 Percobaan pengujian sistem dengan jumlah neuron pada layer tersembunyi pertama 50 dan nilai toleransi 0.0001 51
5.1.5 Percobaan pengujian sistem dengan nilai toleransi 0.008 dan *Threshold* 0.35 53
5.1.6 Percobaan pengujian sistem dengan nilai toleransi 0.008 dan *Threshold* 0.4 54
5.1.7 Percobaan pengujian sistem dengan nilai toleransi 0.008 dan *Threshold* 0.45 55

5.2 Analisa..... 56

BAB VI PENUTUP

6.1 Simpulan..... 57

6.2 Saran..... 57

DAFTAR KEPUSTAKAAN

LAMPIRAN

UNIVERSITAS ANDALAS
UNTUK KEDJAJAAN BANGSA

DAFTAR GAMBAR

Gambar 2.1 Komponen video..... 6

Gambar 2.2 Proses pembentukan citra..... 7

Gambar 2.3 Matriks citra digital N x M..... 8

Gambar 2.4 Sistem ruang warna RGB..... 12

Gambar 2.5 Contoh hasil segmentasi *bilevel luminance thresholding*..... 14

Gambar 2.6 Contoh pemberian label pada objek dalam citra.....15

Gambar 2.7 Daerah penghitungan integral proyeksi..... 16

Gambar 2.8 Grafik integral proyeksi pada gambar huruf C..... 17

Gambar 2.9 Model jaringan layar jamak..... 20

Gambar 2.10 Arsitektur *Backpropagation*..... 22

Gambar 2.11 Grafik fungsi sigmoid biner..... 23

Gambar 2.12 Grafik fungsi sigmoid bipolar..... 23

Gambar 3.1 Plat nomor kendaraan bermotor..... 24

Gambar 3.2 *Webcam Genius* 25

Gambar 3.3 Blok diagram rancangan system secara keseluruhan..... 26

Gambar 4.1 Blok diagram rancangan sistem untuk proses pelatihan..... 30

Gambar 4.2 Blok diagram pengujian sistem pengenalan plat nomor kendaraan..... 31

Gambar 4.3 Gambar citra huruf C..... 32

Gambar 4.4 *webcam Genius iSlim 310*..... 33

Gambar 4.5 Citra intensitas..... 33

Gambar 4.6 Citra hasil segmentasi..... 34

Gambar 4.7 Tampilan GUI sistem pengenalan citra huruf dan angka pada plat 41
Gambar 4.8 Tampilan GUI Pengambilan data citra huruf dan angka 42
Gambar 4.9 Tampilan GUI pada proses memuat citra hasil *capture* 43
Gambar 4.10 Tampilan GUI pada proses konversi citra RGB ke citra intensitas..43
Gambar 4.11 Tampilan GUI pada proses segmentasi 44
Gambar 4.12 Tampilan GUI pada proses filtering dan ekstraksi ciri 44
Gambar 4.13 Tampilan GUI pada proses cropping 45
Gambar 4.14 Tampilan GUI pada proses pengujian 46
Gambar 4.15 Tampilan GUI pada proses mengakhiri program 46
Gambar 5.1 Tabel hasil pengujian untuk toleransi 0.007 48
Gambar 5.2 Tabel hasil pengujian untuk toleransi 0.008 49
Gambar 5.3 Tabel hasil pengujian untuk toleransi 0.009 50
Gambar 5.4 Tabel hasil pengujian untuk toleransi 0.0001 52
Gambar 5.5 Tabel hasil pengujian untuk *threshold* 0.35 53
Gambar 5.6 Tabel hasil pengujian untuk *threshold* 0.4 54
Gambar 5.7 Tabel hasil pengujian untuk *threshold* 0.45 55

DAFTAR TABEL

Tabel 2.1. Kuantisasi citra dengan skala keabuan yang berbeda..... 10

Tabel 2.2. Contoh Pengkombinasian Komponen Warna..... 12

UNIVERSITAS ANDALAS
UNTUK KEDJAJAAN BANGSA

DAFTAR ISTILAH

Istilah	Pengertian	Pemakaian pertama pada halaman
Frame	Struktur dari gambar atau image yang mendukung komponen lainnya dari sebuah konstruksi fisiknya	6
Key frame	Sebuah frame dimana sebuah gambar utuh disimpan dalam data stream.	6
Webcam	Alat penangkap video terhubung kesebuah komputer atau jaringan komputer, sering menggunakan port usb, jika dikoneksi ke jaringan ethernet atau Wi-Fi.	2
Scene	Informasi yang dapat bergerak dari suatu lingkungan fisik kedalam suatu sistem perceptual melalui transducer pancaindra.	6
Shot boundary	Pemisah untuk beberapa shot.	6
Reed green blue (RGB)	Sebuah model warna dimana cahaya merah, hijau, dan biru dicampurkan secara bersama dalam bermacam variasi cara untuk menghasilkan kembali sebuah susunan warna yang lebar.	11
True colour	Sebuah metode penggambaran dan penyimpanan informasi gambar grafik yang menyediakan jumlah warna yang sangat besar, bayangan, dan corak warna yang ditampilkan dalam sebuah citra, seperti gambar fotografi kualitas tinggi grafik kompleks.	12
Thresholding	Metode sederhana dari segmentasi gambar. Thresholding digunakan untuk menghasilkan gambar biner.	13
Scanning	Proses mengkonversi teks dan grafik kertas dokumen, film fotografi, kertas fotografi, atau file lainnya ke gambar digital	14

Filtering	Suatu sistem yang melakukan operasi matematika pada suatu sampel, sinyal waktu diskrit untuk mengurangi mempertinggi aspek tertentu dari gambar.	15
Noise	Gambar yang tidak dinginkan, yang dihasilkan oleh sensor kamera	18
Neuron	Sebuah sel yang diangsang secara elektrik yang memproses dan mengirimkan informasi dengan pennyalan elektris dan kimia	18
Feddback-loop	Hubungan sebab garis edar yang menuntun dari pembangkit awal sinyal feedback ke perubahan berikutnya dari sebuah kejadian.	23
GUI	*Grafical User Interface* adalah metode interaksi secara grafis antara pengguna dan komputer	38

UNIVERSITAS ANDALAS
UNTUK KEDJAJAAN BANGSA

DAFTAR LAMPIRAN

Lampiran A. Listing program menggunakan Matlab 7.04..... A-0

Lampiran B. Tabel data hasil percobaan..... B-0

Lampiran C. Tabel rincian letak piksel pada Integral Proyeksi..... C-0

UNIVERSITAS ANDALAS
UNTUK KEDJAJAAN BANGSA

BAB I
PENDAHULUAN

1.1. Latar Belakang

Pada saat sekarang, perkembangan teknologi dan informasi begitu pesat. Hal ini dilatarbelakangi keinginan manusia untuk terus maju dan membuat segala hal menjadi mudah untuk dilakukan. Dalam melaksanakan kegiatan sehari-hari baik itu kegiatan pribadi maupun kegiatan yang berhubungan dengan pekerjaan, saat ini telah banyak dibantu oleh hasil dari kemajuan teknologi dan informasi itu sendiri. Sehingga dalam beberapa aktivitas manusia hanya perlu memencet beberapa tombol atau hanya dengan bersuara, maka apa yang ia inginkan dilakukan oleh alat-alat yang menjadi pengganti tangan, kaki atau pikiran kita[16].

Bidang komputer merupakan salah satu contoh kemajuan teknologi tersebut. Dapat dilihat pada segi *computer vision* yang mencapai hasil yang berarti diperoleh oleh para peneliti. *Computer vision* merupakan istilah yang bertujuan untuk membuat keputusan berguna mengenai fisik sebenarnya dari sebuah objek dan tentang pemandangan berdasarkan *image* yang didapat dari sensor seperti kamera atau yang lainnya[16].

Salah satu aplikasi pengembangan teknologi berkaitan dengan computer vision adalah bagaimana sebuah mesin komputer dapat mengenali objek pada citra. Meskipun pengenalan objek sangat mudah dilakukan dengan penglihatan manusia, tetapi automatisasi pengolahan pada komputer memerlukan berbagai macam teknik pengolahan citra untuk mengenali sebuah objek. Mata manusia adalah sistem visual yang sangat kompleks. Proses perekaman dan pengenalan objek (*object recognition*) pada mata manusia berada dalam satu sistem yang utuh, sehingga mata manusia dapat langsung mengenali objek dan latar belakangnya, segera setelah mata menangkap dan merekam bayangan suatu citra. Maka untuk membuat suatu sistem visual mesin (*machine vision*), hasil perekaman alat optik membutuhkan proses pengolahan terlebih dahulu

Pengenalan objek layaknya mata ini mengilhami para peneliti untuk mengembangkan aplikasi komputer untuk automatisasi pengolahan citra untuk mengenali objek. Dalam pewujudan aplikasi ini dibutuhkan setidaknya sebuah mesin komputer dan kamera sebagai sensor. Kamera digunakan untuk menangkap gambar objek yang ada di depannya. Komputer digunakan sebagai suatu mesin yang dapat mengolah data gambar dari kamera dan pada akhirnya mengenali gambar objek yang ada di depannya.

Telah cukup banyak penelitian dan tulisan tentang pengenalan objek oleh sistem komputer diantaranya adalah :

a) Hendra Syahputra[15], dalam tugas akhirnya yang berjudul "*Perancangan dan Implementasi Sistem Pengenalan Gambar Kata dan Bilangan dengan Keluaran Suara*" yang membahas tentang pengenalan huruf alfabet. Yang pertama adalah proses segmentasi, labelisasi dan filtering. Kemudian pada tahap kedua adalah ekstraksi karakter objek yang dilakukan dengan metode integral proyeksi. Dengan metode ini setiap karakter diubah kedalam bentuk data atau grafik yang masing-masing karakter memiliki data atau grafik integral proyeksi yang berbeda. Tahap yang ketiga atau terakhir adalah pencocokan data dengan metode template matching. Perlakuan variasi warna pada objek bilangan didapatkan % kesalahan sebesar 0 % pada warna hitam, hijau dan biru, dan 6,67 %pada warna merah. Perlakuan variasi jarak terhadap objek bilangan didapatkan % kesalahan sebesar 0 % pada penambahan 1 cm, 70 % pada penambahan 2 cm, 30 % pada pengurangan jarak 1 cm, 30 % pada pengurangan 2 cm dan 86,67 % pada pengurangan jarak 3 cm dari jarak normal. Perubahan jarak tangkap *webcam* ke objek dari jarak normal berpengaruh terhadap % luas piksel objek.

b) Muhammad Khaidir [7] dalam tugas akhirnya yang berjudul " *Sistem Pengenalan Huruf Alfabet Kapital dengan Variasi Ukuran dan Jarak menggunakan Ekstraksi Ciri Integral Proyeksi dan Pengenalan dengan Jaringan Syaraf Tiruan Backpropagation*". Pada penelitian ini proses pertama yang dilakukan adalah memperoleh data informasi dari obyek yang diubah menjadi data digital. Data digital tersebut dapat diperoleh dengan melakukan *capture* gambar. selanjutnya proses

konversi citra huruf RGB ke citra intensitas, citra intensitas ini selanjutnya akan disegmentasi, proses selanjutnya adalah pelabelan dan filtering, dan dilanjutkan proses ekstraksi ciri menggunakan metode Integral Proyeksi serta hasil ekstraksi ini akan dilatih oleh jaringan syaraf tiruan menggunakan algoritma backpropagation kemudian dari hasil pelatihan ini akan dilakukan proses pengujian. Persentase pengenalan dari pelatihan jaringan dengan arsitektur dan parameter jaringan optimal ini untuk data latih dan data baru masing-masing adalah 98.6% dan 80.3%.

Berdasarkan pemikiran yang telah penulis paparkan, penulis tertarik untuk mengimplimentasikannya berupa pengenalan huruf yang difokuskan pada huruf alfabet kapital dan angka pada plat nomor kendaraan. Proses ini disusun dalam penelitian tugas akhir yang diberi judul **"Sistem Pengenalan Plat Nomor Kendaraan Menggunakan Ekstraksi Ciri Integral Proyeksi dan Pengenalan dengan Jaringan Syaraf Tiruan *Backpropagation*".**

1.2 Tujuan Penelitian

Tujuan dari penelitian ini adalah mendapatkan dan menganalisa proses identifikasi plat kendaraan menggunakan ekstrasi ciri integral proyeksi dan pengenalan jaringan syaraf tiruan *backpropagation.* Sistem ini akan dilatih untuk bekerja dalam pengenalan huruf dan angka pada plat nomor kendaraan

1.3 Manfaat Penelitian

Manfaat dari penelitian ini adalah diharapkan dapat menjadi sistem informasi pengenalan angka dan huruf pada plat nomor kendaraan secara digital.

1.4 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah :

1. Kamera yang digunakan adalah *webcame genius iSlim 310* dengan ukuran 480x640
2. Citra huruf yang akan dimasukkan ke komputer melalui kamera adalah huruf alfabet kapital dan angka pada plat kendaraan standar kepolisian.
3. Jumlah karakter Plat yang akan dijadikan objek adalah 8 karakter
4. Plat yang digunakan adalah *background* warna hitam dan karakter berwarna putih.
5. Teknik pengambilan sampel yang digunakan untuk pengambilan sampel huruf dan *font* huruf adalah sample tidak acak *(nonrandom sampling)* dengan pertimbangan kemudahan (*convenience sampling).* Huruf alfabet kapital yang akan dikenali oleh komputer nantinya adalah semua huruf alfabet dari huruf A sampai Z dan angka 0 sampai angka 9, Warna latar citra huruf haruslah homogen, begitu juga warna objek (huruf) pada citra tersebut.
6. Jarak antara kamera dan objek saat pengambilan gambar adalah 30 cm
7. Simulasi pada penelitian ini menggunakan bahasa pemrograman Matlab 7.04.

1.5 Sistematika Penulisan

1. **Bab I Pendahuluan**

   Bab ini berisikan latar belakang penelitian, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika penulisan.

2. **Bab II Tinjauan Pustaka**

   Bab ini berisi teori dasar yang mendukung penelitian Tugas Akhir ini.

3. **Bab III Metodologi Penelitian**

   Bab ini berisikan tentang langkah–langkah beserta penjelasan mengenai sistem yang akan dirancang.

4. **Bab IV Hasil dan Pembahasan**

   Bab ini akan berisikan analisa terhadap keluaran sistem yang diperoleh dari pengujian sistem itu sendiri.

5. **Bab V Penutup**

   Bab ini berisikan beberapa kesimpulan dan saran yang bisa ditarik dan disampaikan dengan didasari hasil dan pembahasan dari penelitian ini.

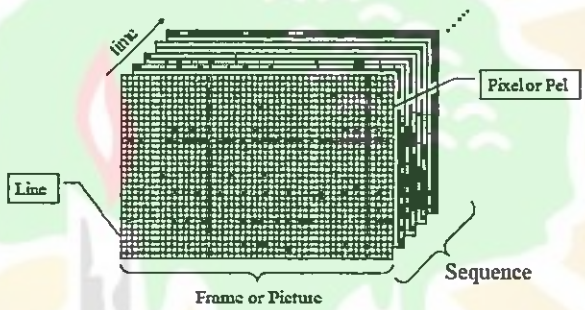
UNIVERSITAS ANDALAS
UNTUK KEDJAJAAN BANGSA

BAB II
TINJAUAN PUSTAKA

2.1 Video

2.1.1 Pengertian Video

Video adalah teknologi pemrosesan sinyal elektronik yang mewakilkan citra bergerak[4]. Video direpresentasikan sebagai serangkaian *frame* yang berurutan dan dihubungkan oleh suatu interval waktu. Untuk menganalisa suatu video, diperlukan pembagian video menjadi unit yang lebih kecil dan memiliki beberapa hubungan semantik. Video dapat dibagi menjadi beberapa *shot, key frame* dan *scene*[17] seperti terlihat pada *gambar 2.1*.



Gambar 2.1 *Komponen video*

Video merupakan serangkaian *frame* utuh yang tidak terputus yang diproduksi dari kamera atau peralatan lain. *Key frame* adalah suatu *frame* yang menunjukkan bagian penting dari *shot*. Dalam satu *shot* dapat terdiri dari beberapa *key frame* tergantung kompleksitas *shot*. Beberapa *shot* dipisahkan oleh *shot boundary*. Sedangkan video *scene* adalah koleksi dari *shot* berdekatan yang memiliki hubungan semantik.

2.1.2 Capture (Proses Pengambilan Video)

Capture adalah proses pengambilan video yang dilakukan secara *frame* demi *frame*. *Frame-frame* yang diambil tadi disimpan dalam memori *buffer* pada kamera, dan disini akan dilihat pengolahan video dari satu buah *frame*. Teknik pengambilan gambar yang digunakan adalah meletakkan gambar huruf yang akan dikenali pada tempat yang disediakan di depan kamera.

2.2 Citra

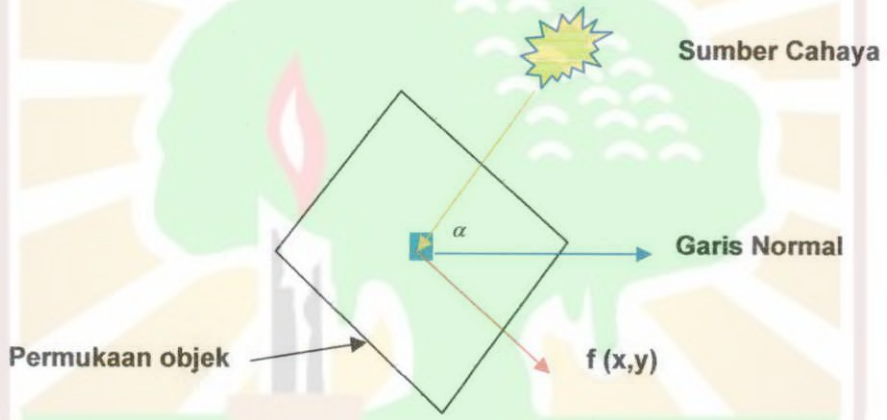
2.2.1 Pengertian citra

Citra merupakan suatu istilah teknis yang biasa dipakai untuk menyebut gambar. Dalam kehidupan sehari-hari, citra dipersepsikan sebagai tampilan visual yang terekam pada alat optik atau indra optik manusia.

Secara harfiah, citra merupakan tampilan objek di bidang dua dimensi. Ditinjau dari sudut pandang matematis, citra merupakan fungsi kontinyu dari intensitas cahaya pada bidang dua dimensi.

2.2.2 Pembentukan citra

Proses pembentukan citra dapat direpresentasikan seperti *gambar 2.2*.



Gambar 2.2 *Proses pembentukan citra*

Permukaan objek mendapatkan sinar dari sumber cahaya. Jumlah pancaran (iluminasi) cahaya yang diterima objek pada koordinat *(x,y)* adalah *i(x,y)*. Objek memantulkan cahaya yang diterimanya dengan derajat pantulan *r(x,y)*. Hasil kali antara *i(x,y)* dan *r(x,y)* menyatakan intensitas cahaya pada koordinat *(x,y)* yang ditangkap oleh sensor visual pada sistem optik, sehingga dapat dinyatakan dengan persamaan di bawah ini.

$$f(x,y) = i(x,y) \bullet r(x,y) \quad \text{.....(1)}$$

dimana:

- f (x,y) = intensitas cahaya pada koordinat titik (x,y) pada citra.
- i (x,y) = jumlah cahaya yang dipancarkan oleh sumber cahaya (*illumination*), bernilai antara 0 dan tak hingga.

- r (x,y) = derajat kemampuan objek dalam memantulkan cahaya (*reflection*), bernilai antara 0 dan 1.

Sumber cahaya mempengaruhi besarnya nilai *i(x,y)*, sedangkan *r(x,y)* ditentukan oleh karakteristik objek dalam gambar. Nilai *r(x,y)* = 0 mengindikasikan penyerapan total, sedangkan *r(x,y)* = 1 menyatakan pemantulan total. Jika permukaan mempunyai derajat pemantulan nol, maka fungsi intensitas cahaya *f(x,y)* juga nol, yang artinya gambar tidak dapat ditangkap oleh sensor visual. Sebaliknya, jika permukaan mempunyai derajat pemantulan 1, maka fungsi intensitas cahaya sama dengan iluminasi yang diterima oleh permukaan tersebut, ini berarti citra yang ditangkap oleh sensor visual mendekati aslinya, tergantung kualitas sensor visual tersebut.

2.2.3 Digitalisasi Citra [9]

Suatu citra harus direpresentasikan secara numerik dengan nilai – nilai diskrit dengan tujuan agar dapat diolah dengan komputer digital. Representasi citra dari fungsi kontinu menjadi nilai – nilai diskrit disebut digitalisasi. Citra yang dihasilkan inilah yang disebut citra digital (*digital image*). Pada umumnya citra digital berbentuk empat persegi panjang, dan dimensi ukurannya dinyatakan sebagai tinggi x lebar (atau lebar x panjang). Masing–masing elemen pada citra digital (elemen matriks) disebut *image element*, *picture element* atau *pixel* (piksel). Citra digital yang berukuran N x M lazim dinyatakan dengan matriks yang berukuran N baris dan M kolom. Jadi, citra yang berukuran NxM mempunyai NM buah *pixel* seperti yang terlihat pada *gambar 2.3*.

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

Gambar 2.3 *Matriks Citra Digital N x M*

Indeks baris (*i*) dan indeks kolom (*j*) menyatakan suatu koordinat titik pada citra, sedangkan *f(i,j)* merupakan intensitas (derajat keabuan) pada titik *(i,j)*. Sebagai contoh, misalkan sebuah citra berukuran 256x256 piksel dan

direpresentasikan secara numerik dengan matriks yang terdiri dari 256 baris (di-indeks dari 0 sampai 255) dan 256 buah kolom (di-indeks dari 0 sampai 255) seperti contoh berikut:

$$\begin{bmatrix} 0 & 125 & 135 & \dots & 232 \\ 0 & 130 & 231 & \dots & 189 \\ : & : & : & : & : \\ 122 & 210 & 213 & \dots & 154 \end{bmatrix}$$

Proses digitalisasi citra terdiri dari dua tahap :

a) Digitalisasi spasial (x, y), sering disebut sebagai sampling.

Sampling menyatakan besarnya kotak-kotak yang disusun dalam baris dan kolom. Dengan kata lain sampling pada citra menyatakan besar kecilnya ukuran piksel pada citra.

Untuk memudahkan implementasi, jumlah sampling biasanya diasumsikan perpangkatan dari dua,

$$N = 2^n \quad \text{.....(2)}$$

dimana ,

N= jumlah sampling pada suatu baris/kolom

n = bilangan bulat positif

Pembagian gambar menjadi ukuran tertentu menentukan resolusi spasial yang diperoleh. Semakin tinggi resolusinya, yang berarti semakin kecil ukuran piksel (atau semakin banyak jumlah pikselnya), semakin halus gambar yang diperoleh karena informasi yang hilang akibat pengelompokkan derajat keabuan pada pen-sampling-an semakin kecil.

b) Digitalisasi intensitas $f(x, y)$, sering disebut sebagai kuantisasi.

Setelah proses sampling pada citra maka proses selanjutnya adalah kuantisasi. Kuantisasi menyatakan besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan (*grey scale*) sesuai dengan jumlah bit biner yang digunakan, dengan kata lain kuantisasi pada citra menyatakan jumlah warna yang ada pada citra. Proses kuantisasi membagi skala keabuan (0, L) menjadi G buah level yang dinyatakan dengan suatu harga bilangan bulat (integer), biasanya G diambil perpangkatan dari 2,

$$G = 2^m \quad \text{.....(3)}$$

MILIK
UPT PERPUSTAKAAN
UNIVERSITAS ANDALAS

dimana,

G = derajat keabuan

m = bilangan bulat positif

Tabel 2.1 adalah tabel kuantisasi citra dengan skala keabuan yang berbeda-beda.

Tabel 2.1. *Kuantisasi citra dengan skala keabuan yang berbeda*

Skala keabuan	**Rentang nilai keabuan**	***Pixel depth***
2^1 (2 nilai)	0,1	1 bit
2^2 (4 nilai)	0 sampai 3	2 bit
2^4 (16 nilai)	0 sampai 15	4 bit
2^8 (256 nilai)	0 sampai 255	8 bit

Jumlah bit yang dibutuhkan untuk mempresentasikan nilai keabuan piksel disebut kedalaman piksel (*pixel depth*). Citra sering diasosiasikan dengan kedalaman pikselnya. Jadi, citra dengan kedalaman 8 bit disebut juga citra 8-bit (atau citra 256 warna, $G = 256 = 2^8$). Semakin banyak jumlah derajat keabuan (berarti jumlah bit kuantisasinya makin banyak), semakin bagus gambar yang diperoleh.

2.2.4. Histogram Citra [9]

Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas piksel dari suatu citra atau bagian tertentu di dalam citra[2]. Citra dibentuk dari oleh piksel-piksel yang merupakan bagaian terkecil dari sebuah citra, jadi pengolahan citra dapat dikatakan sebagai pengolahan piksel-piksel. Untuk dapat mengolah piksel inilah, informasi tentang sebaran piksel dengan intensitas tertentu dalam suatu citra harus diketahui terlebih dahulu. Informasi ini dapat diperoleh dengan tepat dari sebuah histogram. Histogram citra akan menampilkan frekuensi kemunculan relatif dari intensitas pada citra tersebut. Disamping itu histogram juga mampu menampilkan tingkat kecerahan dan kontras citra.

Pembuatan histogram dari sebuah citra akan memudahkan operasi-operasi pengolahan citra. Dari histogram dapat diketahui kualitas citra, juga dalam penentuan nilai ambang batas yang tepat. Histogram tidak secara langsung

MILIK
UPT PERPUSTAKAAN
UNIVERSITAS ANDALAS

merepresentasikan sebuah citra. Histogram dari sebuah citra dapat diperoleh dari perbandingan jumlah piksel dengan derajat keabuan tertentu terhadap jumlah keseluruhan piksel. Persamaan 4 memperlihatkan bagaimana suatu histogram citra dibuat[4].

$$h_i = \frac{n_i}{n} \quad \text{.....} \quad (4)$$

dimana: h_i = amplitudo histogram

n_i = jumlah *pixel* yang memiliki derajat keabuan i

n = jumlah keseluruhan *pixel* pada citra

i = 1,2, 3, ..., 255 (untuk citra 8 bit)

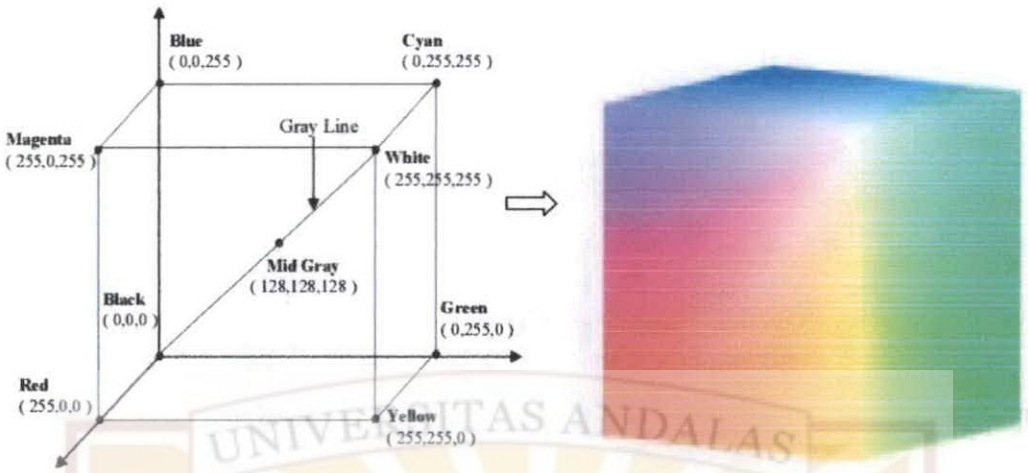
2.3 Sistem Komponen Ruang Warna

2.3.1 RGB (Red Green Blue)[9]

Ada beberapa sistem ruang warna yang diciptakan untuk keperluan tertentu atau diciptakan khusus untuk perangkat keras tertentu seperti monitor, printer. Berikut ini akan dijelaskan mengenai sistem ruang warna yang digunakan dalam pemrosesan citra pada tugas akhir ini yaitu sistem warna *Red Green Blue* (RGB).

Sistem ruang warna RGB banyak digunakan untuk menampilkan citra berwarna pada monitor komputer. Sistem ruang warna RGB ini memiliki 3 warna dasar, yaitu merah (*Red*), hijau (*Green*), biru (*Blue*). Setiap warna dasar, misalnya "merah", dapat diberi rentang-nilai. Untuk monitor komputer, nilai rentangnya paling kecil = 0 dan paling besar = 255. Pilihan skala 256 ini semata-mata didasarkan pada cara mengungkap 8-digit bilangan biner yang digunakan oleh mesin komputer.

Dengan cara ini, akan diperoleh warna campuran sebanyak 256 x 256 x 256 = 1677726 jenis warna. Sistem ruang warna RGB ini diilustrasikan dengan menggunakan sistem koordinat 3 dimensi .



Gambar 2.4. *Sistem Ruang Warna RGB*

Pada *gambar 2.4* tampak bahwa setiap warna akan diwakili oleh tiga buah nilai dalam koordinat tersebut yang menyatakan komponen warna RGB. Seperti warna merah yang diwakili oleh titik (255,0,0).

2.3.2 Kombinasi Warna [9]

Dasar pengolahan citra adalah pengolahan warna RGB (*Red,Green, Blue)* pada posisi tertentu. Dalam pengolahan citra, warna direpresentasikan dengan nilai hexadesimal dari 0x00000000 sampai 0x00FFFFFF. Warna hitam adalah adalah 0x00000000 dan warna putih adalah 0x00FFFFFF. Berikut terlihat definisi nilai warna dan variabel 0x00 menyatakan angka dibelakangnya adalah hexadesimal, yang juga merupakan aturan penulisan kombinasi warna RGB.

0 x 00 XX XX XX

Nilai *Blue* Nilai *Green* Nilai *Red*

Setiap warna mempunyai range nilai 00 (angka desimalnya adalah 0) dan FF (angka desimalnya adalah 255) atau mempunyai nilai derajat keabuan (*grey scale*) $256 = 2^8$. Dengan demikian, range warna yang digunakan adalah $2^8 \cdot 2^8 \cdot 2^8 = 2^{24}$ (atau dikenal dengan istilah *true colour* pada windows). Dalam *tabel 2.2* ditampilkan contoh pengkombinasian komponen warna .

Tabel 2.2. *Contoh Pengkombinasian Komponen Warna*

NILAI	WARNA
0 x 00000000	Hitam
0 x 000000FF	Merah

0 x 0000FF00	Hijau
0 x 00FF0000	Biru
0 x 0000FFFF	Kuning
0 x 00FFFFFF	Putih

Nilai warna yang digunakan merupakan gabungan warna cahaya merah, hijau dan biru. Oleh karena itu, penentuan nilai suatu warna yang bukan warna dasar merupakan gabungan skala kecerahan setiap warnanya. Jadi kita mudah menyajikan warna tertentu, yaitu dengan mencampurkan ketiga warna RGB.

2.4 Segmentasi Citra [9]

2.4.1 Definisi Segmentasi

Segmentasi adalah proses mempartisi citra menjadi daerah-daerah (*region*). Jika kita ingin mempartisi citra berdasarkan warnanya, maka kita dapat melakukannya pada setiap lapisan warna (baik pada HSI atau RGB). Segmentasi citra merupakan salah satu proses pengolahan citra tingkat dasar. Karena hanya dilakukan proses analisa citra untuk kemudian mengubah tampilan citra secara keseluruhan.

Beberapa manfaat dari segmentasi adalah mengeluarkan informasi yang tidak diperlukan dan meningkatkan informasi yang dibutuhkan untuk proses pengolahan citra pada tingkatan selanjutnya. Selain itu juga dengan adanya pemisahan objek citra menjadi pecahan-pecahan yang dibedakan berdasarkan homogenitas ini, dapat dikenali bentuk objek yang sebenarnya (nyata).

2.4.2 Bilevel Luminance Thresholding

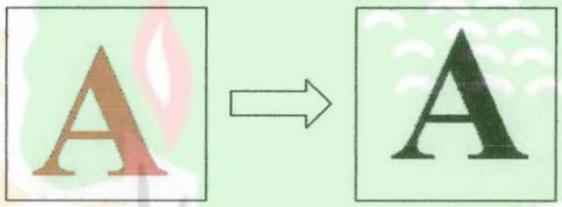
Bilevel Luminance Threholding merupakan metode segmentasi yang digunakan dalam penelitian ini. Metode segmentasi ini dipergunakan hanya untuk memisahkan 2 daerah pada citra, yaitu daerah gelap dan daerah terang saja. Metode ini sangat bermanfaat untuk mensegmentasi citra biner.

Sebagian besar dari pemrosesan citra dilakukan dengan memproses warna-warna yang ada dalam citra tersebut. Warna merupakan komponen yang penting dalam melakukan proses pengenalan objek, karena warna dapat digunakan untuk membedakan antara objek yang dicari dengan latar belakang.

MILIK
UPT PERPUSTAKAAN
UNIVERSITAS ANDALAS

Proses segmentasi dimulai dari penentuan warna yang ingin disegmentasi. Ruang warna intensitas/*luminance* digunakan pada penelitian ini sebagai parameter *threshold*. Proses dari segmentasi dapat dilihat dari tahapan di bawah ini:

a. *Scanning* koordinat piksel (i,j) pada citra hasil *capture*.
b. Dapatkan nilai r, g dan b pada masing-masing koordinat piksel. Konversi warna RGB tersebut ke dalam nilai intensitas.
c. Apabila ditemukan nilai intensitas diatas nilai *threshold* seperti pada *gambar 2.5* (dianggap sebagai objek) ubah menjadi r = 255, g = 255, b = 255 (warna putih), sedangkan yang nilai intensitas dibawah nilai *threshold* (dianggap sebagai latar belakang) diubah menjadi r = 0, g = 0, b = 0 (warna hitam).



Gambar 2.5. Contoh hasil segmentasi *bilevel luminance thresholding*

Objek citra hasil segmentasi pada penelitian ini berwarna hitam sedangkan latarnya berwarna putih. Ini bertujuan untuk memudahkan dalam pemrosesan lebih lanjut, yaitu pada pemfilteran dan integral proyeksi.

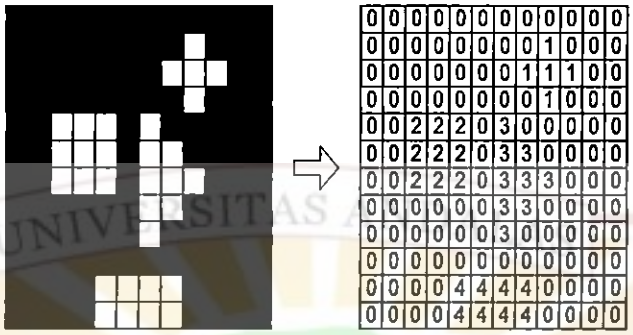
2.5 Pelabelan dan Pemfilteran [9]

Dengan operasi pelabelan, maka ciri-ciri mendasar dari masing-masing objek dapat dihitung secara sendiri-sendiri. Tanpa pelabelan, maka perhitungan ciri-ciri dari objek-objek akan rancu, sebab hasil yang didapat merupakan gabungan dari semua objek yang ada, yang dapat sangat berbeda dengan ciri-ciri objek tadi bila dianalisa secara individual seperti yang terlihat pada gambar 2.6.

Algoritma tersebut menerapkan langkah-langkah sebagai berikut:

1. Baca citra secara sistematis (misalnya dari kiri ke kanan, atas ke bawah) untuk menemukan piksel objek yang belum diberi label dan beri label baru.

2. Beri label yang sama pada semua piksel tetangganya.
3. Berhenti bila tidak ada lagi tetangga yang merupakan piksel objek.
4. Ulangi langkah 1 sampai 3 hingga tidak ada piksel objek yang belum diberi label.



Gambar 2.6 Contoh pemberian label pada objek dalam citra

Untuk *filtering* digunakan parameter ukuran atau luas dari objek, artinya seberapa banyak piksel yang berlabel sama. Jika luas objek tersebut kurang dari *threshold* luas yang telah ditetapkan maka objek tersebut dianggap *noise* dan dihilangkan.

2.6 Integral Proyeksi [13]

Integral proyeksi merupakan suatu metode yang digunakan untuk mencari daerah atau lokasi dari objek[4]. Metode ini dapat digunakan untuk mendeteksi batas dari daerah gambar yang berbeda, sehingga kita bisa mendapatkan daerah lokasi gambar beserta fitur-fiturnya.

Integral Proyeksi ini terbagi dua, yaitu integral proyeksi vertikal dan integral proyeksi horizontal. Oleh sebab itu, metode ini juga bisa disebut dengan integral baris (horizantal) dan kolom (vertikal) dari piksel, karena integral ini menjumlahkan piksel per baris dan piksel per kolom.

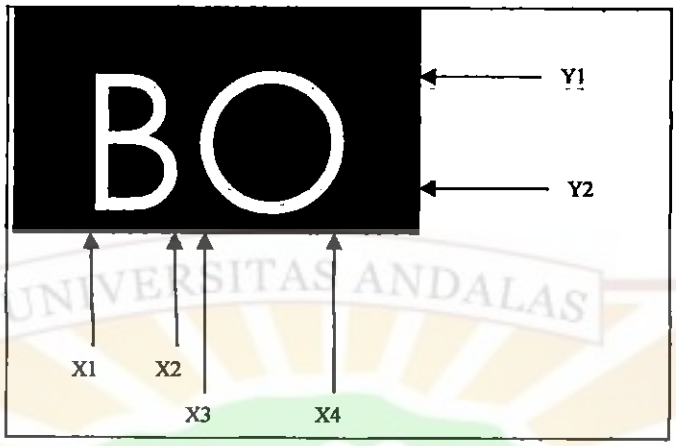
Persamaan untuk integral proyeksi horizontal ditunjukkan oleh persamaan (6), sedangkan persamaan untuk integral proyeksi vertikal ditunjukkan oleh persamaan (7)

$$IPF_v(X) = \int_{y1}^{y2} I(x,y)dy \quad \text{..... (6)}$$

$$IPF_h(y) = \int_{x1}^{x2} I(x,y)dy \quad \text{.....(7)}$$

dimana,

I (x,y) = intesitas derajat keabuan pada piksel (x,y) dari gambar objek.



Gambar 2.7. *Daerah penghitungan integral proyeksi*

Gambar 2.7 menunjukkan pembagian daerah penghitungan integral proyeksi dari sebuah *frame* yang memiliki dua huruf.

Daerah dari x1 sampai x2 merupakan integral proyeksi horizontal dari huruf B. Kemudian dilakukan penjumlahan dari piksel warna putih (objek gambar) dari masing-masing baris sepanjang y1 ke y2 sehingga akan didapatkan grafik hasil penjumlahan piksel. Daerah dari y1 sampai y2 merupakan integral proyeksi vertikal dari gambar objek. Kemudian dilakukan penjumlahan dari piksel warna putih dari masing-masing kolom sepanjang x1 ke x2 sehingga didapat grafik hasil integral proyeksi vertikalnya.

Sedangkan integral proyeksi untuk huruf O. Daerah dari x3 sampai x4 merupakan integral proyeksi horizontal dari huruf B. Kemudian dilakukan penjumlahan dari piksel warna putih (objek gambar) dari masing-masing baris sepanjang y1 ke y2 sehingga akan didapatkan grafik hasil penjumlahan piksel. Daerah dari y1 sampai y2 merupakan integral proyeksi vertikal dari gambar objek. Kemudian dilakukan penjumlahan dari piksel warna putih dari masing-masing kolom sepanjang x3 ke x4 sehingga didapat grafik hasil integral proyeksi vertikalnya.

Dari metode integral proyeksi ini akan kita dapatkan hasil penjumlahan baris dan kolom yang nantinya akan kita olah atau proses lebih lanjut. Tiap piksel citra atau gambar yang diproses memiliki nilai *cluster*, jadi dalam satu gambar

2. Sinyal mengalir diantara sel syaraf/*neuron* melalui suatu sambungan penghubung.

3. Setiap sambungan penghubung memiliki bobot yang bersesuaian.

   Bobot ini digunakan untuk menggandakan/mengalikan sinyal yang dikirim melaluinya.

4. Setiap sel syaraf akan menerapkan fungsi aktifasi terhadap sinyal hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan sinyal keluarannya.

Definisi lain tentang jaringan syaraf tiruan yaitu suatu model komputasi yang terinspirasi secara biologi yang terdiri dari elemen-elemen pemroses (yang disebut neuron) dan penghubung dintaranya dengan batas koefisien-koefisien (bobot) ke penghubung, yang merupakan struktur *neuronal,* pelatihan dan pemanggilan kembali algoritma yang diletakkan ke struktur[14].

2.7.2 Kelebihan Jaringan Syaraf Tiruan[5]

Jaringan syaraf tiruan memiliki sejumlah besar kelebihan dibandingkan dengan metode perhitungan lainnya, yaitu:

a. Kemampuan mengakuisisi pengetahuan walaupun dalam kondisi ada gangguan dan ketidakpastian. Hal ini karena jaringan syaraf tiruan mampu melakukan generalisasi, abstraksi, dan ekstraksi terhadap properti statistik dari data.
b. Kemampuan merepresentasikan pengetahuan secara fleksibel. Jaringan syaraf tiruan dapat menciptakan sendiri representasi melalui pengaturan diri sendiri atau kemampuan belajar (*self organizing*).
c. Kemampuan memberikan toleransi atas suatu distorsi (*error/fault*), di mana gangguan kecil pada data dapat dianggap hanya sebagai *noise* (guncangan) belaka.
d. Kemampuan memproses pengetahuan secara efisien karena memakai sistem paralel, sehingga waktu yang diperlukan untuk mengoperasikannya menjadi lebih singkat.

2.7.3 Karakteristik Jaringan Syaraf Tiruan[11]

Karakteristik Jaringan syaraf tiruan ditentukan oleh 3 hal:

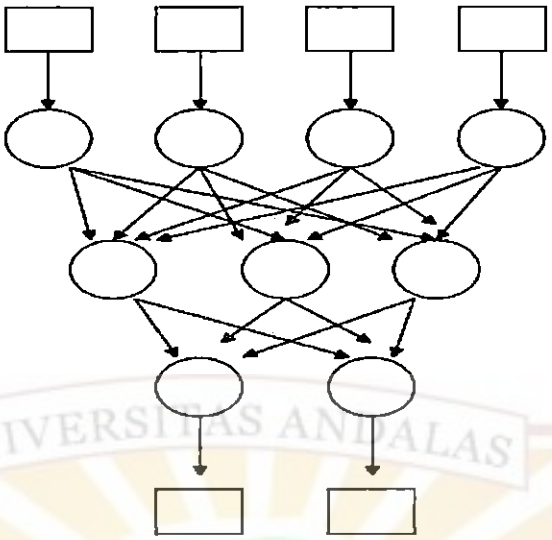
a. Pola hubungan antar neuron (disebut dengan arsitektur jaringan)
b. Metode untuk menentukan bobot-bobot sambungan (disebut dengan pelatihan atau proses belajar jaringan)
c. Fungsi aktivasi

2.7.4 Model Dasar Jaringan Syaraf Tiruan[16]

Mengadopsi esensi dasar dari sistem syaraf biologi, syaraf tiruan digambarkan sebagai berikut: Menerima masukan baik dari data yang dimasukkan atau dari keluaran sel syaraf pada jaringan syaraf. Setiap masukan datang melalui suatu koneksi atau hubungan yang mempunyai sebuah bobot (*weight*). Setiap sel syaraf mempunyai sebuah nilai ambang. Jumlah bobot dari masukan dan dikurangi dengan nilai ambang kemudian akan mendapatkan suatu aktivasi dari sel syaraf. Sinyal aktivasi kemudian menjadi fungsi aktivasi untuk menghasilkan masukan dari sel syaraf. Jika tahapan fungsi aktivasi digunakan (keluaran sel saraf = 0 jika masukan <0 dan 1 jika masukan >= 0) maka tindakan sel syaraf sama dengan sel syaraf biologi yang dijelaskan di atas (pengurangan nilai ambang dari jumlah bobot dan membandingkan dengan 0 adalah sama dengan membandingkan jumlah bobot dengan nilai ambang).

2.7.5 Arsitektur Jaringan: Jaringan Layar Jamak (*multi layer network*)

Jaringan layar jamak merupakan perluasan dari layar tunggal. Dalam jaringan ini, selain unit masukan dan keluaran, ada unit-unit lain (sering disebut layar tersembunyi). Dimungkinkan pula ada beberapa layar tersembunyi. Sama seperti pada unit masukan dan keluaran, unit-unit dalam satu layar tidak saling berhubungan. *Gambar 2.9* adalah gambar model jaringan syaraf layar jamak.



Gambar 2.9 Model jaringan layar jamak

Jaringan layar jamak dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan layar tunggal, meskipun kadang kala proses pelatihan lebih kompleks dan lama.

2.7.6 Bias dan *Threshold*[11]

Terkadang dalam jaringan ditambahkan sebuah unit masukan yang nilainya selalu = 1. Unit yang sedemikian itu disebut bias. Bias dapat dipandang sebagai sebuah input yang nilainya = 1. Bias berfungsi untuk mengubah nilai *threshold* menjadi = 0 (bukan = a). Jika melibatkan bias, maka keluaran unit penjumlahan adalah

$$net = b + \sum_i x_i w_i \quad \text{.....(8)}$$

Fungsi aktivasi *threshold* menjadi:

$$f(net) = \begin{cases} 1 & jika\ net \geq 0 \\ -1 & jika\ net < 0 \end{cases} \quad \text{.....(9)}$$

2.7.7 Pelatihan dengan dan Tanpa Supervisi[11]

Berdasarkan cara memodifikasi bobotnya, ada 2 macam pelatihan yang dikenal yaitu dengan supervisi (*supervised*) dan tanpa supervisi (*unsupervised*).

Metode pelatihan dengan supervisi digunakan jika keluaran yang diharapkan telah diketahui sebelumnya. Terdapat sejumlah pasangan data (masukan – target keluaran) yang telah ada yang dipakai untuk melatih jaringan hingga diperoleh bobot yang diinginkan. Pasangan data tersebut berfungsi sebagai "guru" untuk melatih jaringan hingga diperoleh bentuk yang terbaik. "Guru akan memberikan informasi yang jelas tentang bagaimana sistem harus mengubah dirinya untuk meningkatkan unjuk kerjanya.

Sebaliknya, dalam pelatihan tanpa supervisi (*unsupervised learning*) tidak memerlukan target keluaran atau tidak ada "guru" yang akan mengarahkan proses pelatihan. Selama proses pelatihan, nilai bobot disusun dalam suatu rentang tertentu tergantung pada nilai masukan yang diberikan. Tujuan pelatihan ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pelatihan seperti ini biasanya sangat cocok untuk pengelompokkan (klasifikasi pola).

2.7.8 Model-Model Jaringan Syaraf Tiruan[10]

2.7.8.1 Model Jaringan *Backpropagation*

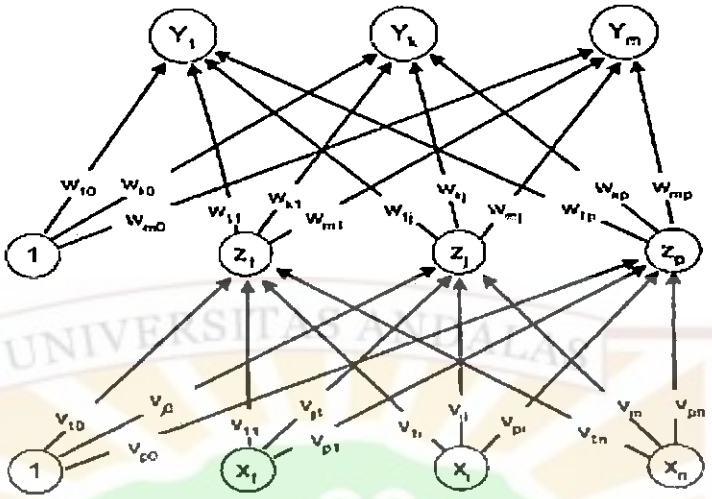
Jaringan *backpropagation* merupakan salah satu algoritma yang sering digunakan dalam menyelesaikan masalah-masalah yang rumit. Hal ini dimungkinkan karena jaringan dengan algoritma ini dilatih dengan menggunakan metoda belajar terbimbing. Pada jaringan diberikan sepasang pola yang terdiri atas pola masukan dan pola yang diinginkan. Ketika suatu pola diberikan kepada jaringan, bobot-bobot diubah untuk memperkecil perbedaan pola keluaran dan pola yang diinginkan. latihan ini dilakukan berulang-ulang sehingga semua pola yang dikeluarkan jaringan dapat memenuhi pola yang diinginkan.

Algoritma pelatihan jaringan syaraf *backpropagation* terdiri atas dua langkah, yaitu perambatan maju dan perambatan mundur ini dilakukan pada jaringan untuk setiap pola yang diberikan selama jaringan mengalami pelatihan.

2.7.8.1.1 Arsitektur *Backpropagation*

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih layar tersembunyi. *Gambar 2.10* adalah arsitektur *backpropagation* dengan n

buah masukan (ditambah sebuah bias), sebuah layar tersembunyi yang terdiri dari p unit (ditambah sebuah bias), serta m buah unit keluaran.



Gambar 2.10 Arsitektur *Backpropagation*

V_{ji} merupakan bobot garis dari unit masukan x_i ke unit layar tersembunyi Z_j (V_{j0} merupakan bobot garis yang menghubungkan bias di unit masukan ke unit layar tersembunyi Z_j).

W_{kj} merupakan bobot dari unit layar tersembunyi Z_j ke unit keluaran Y_k (W_{k0} merupakan bobot dari bias di layar tersembunyi ke unit keluaran Z_k).

2.7.8.1.2. Fungsi Aktivasi

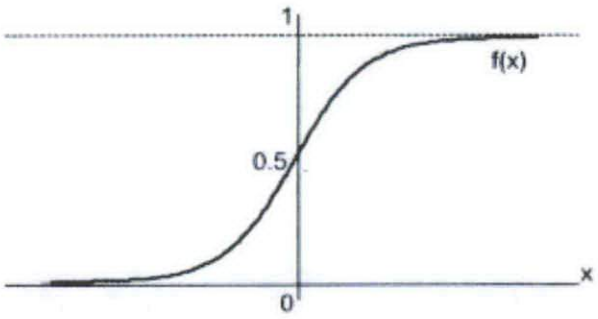
Dalam *backpropagation*, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu: kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering dipakai adalah fungsi sigmoid biner yang memiliki range (0,1).

$$f(x) = \frac{1}{1+e^{-x}} \quad \text{.....(10)}$$

dengan turunan :

$$f'(x) = f(x)(1-f(x)) \quad \text{.....(11)}$$

Grafik fungsinya tampak pada *gambar 2.11*



Gambar 2.11 Grafik fungsi sigmoid biner

Fungsi lain yang sering dipakai adalah fungsi sigmoid bipolar yang bentuk fungsinya mirip dengan fungsi sigmoid biner, tapi dengan range (-1,1).

$$f(x) = \frac{2}{1+e^{-x}} - 1 \quad \text{.....(12)}$$

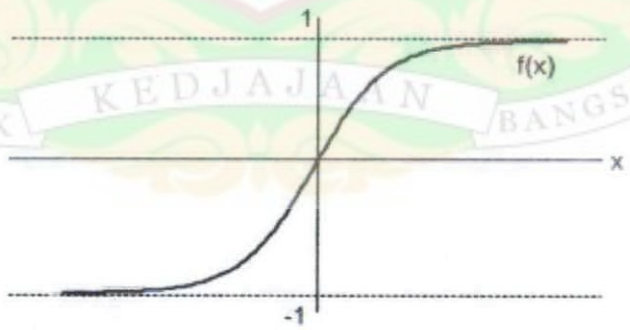
dengan turunan :

$$f'(x) = \frac{(1+(f(x))(1-(f(x))}{2} \quad \text{.....(13)}$$

Grafik fungsinya tampak pada gambar 2.12.

Fungsi sigmoid memiliki nilai maksimum=1. Maka untuk pola yang targetnya > 1, pola masukan dan keluaran harus terlebih dahulu ditransformasi sehingga semua polanya memiliki rentang yang sama seperti fungsi sigmoid yang dipakai.

Alternatif lain adalah menggunakan fungsi aktivasi sigmoid hanya pada layar yang bukan layar keluaran. Pada layar keluaran, fungsi aktivasi yang dipakai adalah fungsi identitas : $f(x) = x$.



Gambar 2.12 Grafik fungsi sigmoid bipolar

BAB III
METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Berdasarkan tujuan dan manfaat tugas akhir ini, maka penelitian ini termasuk ke dalam jenis penelitian terapan (aplikatif), dimana penelitian ini merupakan pemecahan terhadap suatu masalah untuk tujuan tertentu dan merupakan pengembangan dari aplikasi dan penelitian yang telah ada dengan berpedoman pada data sekunder (data dari hasil penelitian) yang relevan. Aplikasi dari penelitian ini diharapkan dapat dipergunakan untuk mengenali karakter dari plat nomor kendaraan bermotor. Adapun jika ditinjau dari sifat-sifat masalahnya penelitian ini bersifat eksperimen.

3.2 Objek Penelitian

Berdasarkan batasan masalah yang telah dijabarkan dalam bab I tugas akhir ini, maka objek penelitian yang digunakan dalam penelitian ini terdiri dari beberapa kondisi, yaitu :

1. Objek

   Dalam perencanaan penelitian ini, objek yang akan digunakan adalah sebuah plat nomor kendaraan sepeda motor standar kepolisiandengan deapan karakter. Warna objek yang digunakan adalah *background* berwarna hitam dan karakter berwarna putih seperti pada **Gambar 3.1** berikut ini.

Gambar 3.1 Plat Nomor kendaraan bermotor

2. Webcam

Pada penelitian ini digunakan *webcam* produk dari *Genius* dengan tipe *iSlim 301*.

Gambar 3.2 *Webcam Genius*

3. Papan

Papan ini digunakan sebagai tempat plat nomor saat dilakukan pengambilan gambar oleh webcam. Jarak antara *webcam* dengan objek adalah 30 cm.

Gambar 3.3 Papan untuk plat nomor kendaraan

Penelitian ini memakai objek berupa citra masukan yang di-*capture* langsung melalui *webcam*. Pengenalan objek berupa karakter dari plat bisa langsung dilakukan dengan melalui beberapa tahapan pengolahan citra.

Ada beberapan tahapan dalam desain sistem ini :

1. Tahapan dimulai dengan proses *capture* atau pengambilan gambar huruf menggunakan *webcam* yang terhubung ke komputer. Yang ditangkap oleh *webcam* adalah *frame-frame* yang kemudian disimpan di memori *buffer* kamera.
2. Hasil *capture* tadi kemudian diproses dengan tahapan pengolahan citra yaitu segmentasi. Dalam tahap ini dilakukan pemisahan terhadap daerah atau bagian tertentu terhadap objek sehingga didapat bentuk huruf yang telah disegmentasi.
3. Selanjutnya dilakukan proses pengolahan citra berikutnya yaitu pemfilteran, dengan pemfilteran akan diberikan label pada masing-masing objek pada koordinat piksel, dimana objek yang berbeda akan memiliki label yang berbeda pula. Pemfilteran ini bertujuan untuk menghilangkan *noise* atau bagian-bagian tertentu pada objek.
4. Dari hasil penfilteran tadi dilakukan pendeteksian ciri/fitur bentuk dari satu persatu gambar huruf. Disini digunakan metode ektraksi ciri Integral Proyeksi.
5. Hasil dari ekstraksi ciri ini akan dimasukkan ke jaringan syaraf tiruan dan selanjutnya akan dilakukan proses pelatihan. Proses ini berlangsung terhadap dua buah parameter yang telah ditentukan, input dan target. Proses pelatihan akan dilakukan sampai diperoleh *error* atau kesalahan terkecil yang telah ditetapkan sebelumnya.
6. Setelah proses pelatihan, sistem akan masuk ke proses pengujian, dimana sistem akan mengenali pola huruf alfabet kapital berdasarkan pola-pola yang telah dilatih sebelumnya.
7. Data yang akan diambil nantinya adalah image plat kendaraan bermotor.
8. Data yang akan diuji mengalami proses RGB ke intesnitas, kemudian disegmentasi. Setelah proses segmentasi, data melalui proses *cropping* yang ukurannya sudah ditentukan. Pada saat *cropping* secara bersamaan data di *filtering* dan di ekstraksi menggunakan integral proyeksi.

9. Data untuk karakteristik jaringan syaraf tiruan yang akan ditentukan nantinya berupa arsitektur *(layer hidden), neuron input*, *neuron output*, inisialisasi bobot, fungsi aktivasi, toleransi galat, *epoch*, laju pembelajaran.

3.4 Teknik Analisa Sistem

Kinerja sistem yang akan dieksperimenkan dianalisa dengan menggunakan penilaian objektif. Hasil-hasil yang didapatkan melalui eksperimen (keluaran sistem) dibandingkan dengan teori-teori yang berasal dari literatur yang ada.

3.5 Prosedur Penelitian

Agar penelitian dapat lebih terarah dan efektif, penulis telah menyusun dan akan mengikuti prosedur penelitian sebagai berikut:

1. Tinjauan dan Studi Kepustakaan

Dilakukan untuk mendapatkan pemahaman tentang konsep-konsep teoritis yang berhubungan dengan tema penelitian. Studi literatur berupa diskusi dengan pembimbing, mengambil bahan dari internet dan buku-buku panduan. Studi literatur juga dilakukan terhadap bahasa pemograman Matlab 7.04 yang dalam penelitian ini menjadi sarana aplikatif.

2. Penyusunan Algoritma Program

Program yang dirancang dalam tugas akhir ini terbagi atas 4 (empat) algoritma pokok, yaitu:

a. Algoritma segmentasi citra berdasarkan metode *bilevel luminance thresholding*. Algoritma ini bertujuan memisahkan antara objek dengan latar (*background*).
b. Algoritma pelabelan dan *filtering* objek. Pelabelan dan *filtering* objek bertujuan untuk menghilangkan *noise* terhadap objek yang tidak diinginkan.
c. Algoritma ekstraksi ciri untuk mengambil ciri-ciri tertentu dari karakter yang diamati.
d. Algoritma jaringan syaraf tiruan *backpropagation* untuk pelatihan dan pengujian.

3. Perancangan dan Pembuatan Program

Berdasarkan algoritma yang telah ditentukan diatas, dibuat program pengolahan citra masukan dengan menggunakan bahasa pemograman Matlab 7.04.

4. Analisa Keluaran Sistem

Analisa keluaran sistem didasarkan pada teknik analisa sistem yang telah dijabarkan pada subbab 3.4. Keluaran dari sistem ini adalah informasi huruf alfabet kapital yang dikenali komputer. Analisa terhadap keluaran sistem dilakukan dengan cara menggunakan jarigan syaraf tiruan *backpropagation* untuk mempelajari dan mengenali pola yang terdapat gambar huruf yang ditangkap *webcam*.

5. Penyusunan Laporan Penelitian

Berisikan penjelasan secara tertulis dan tergambar keseluruhan proses penelitian ini, dengan rincian isi yang telah dijabarkan didalam subbab 1.5 tugas akhir ini mengenai sistematika penulisan.

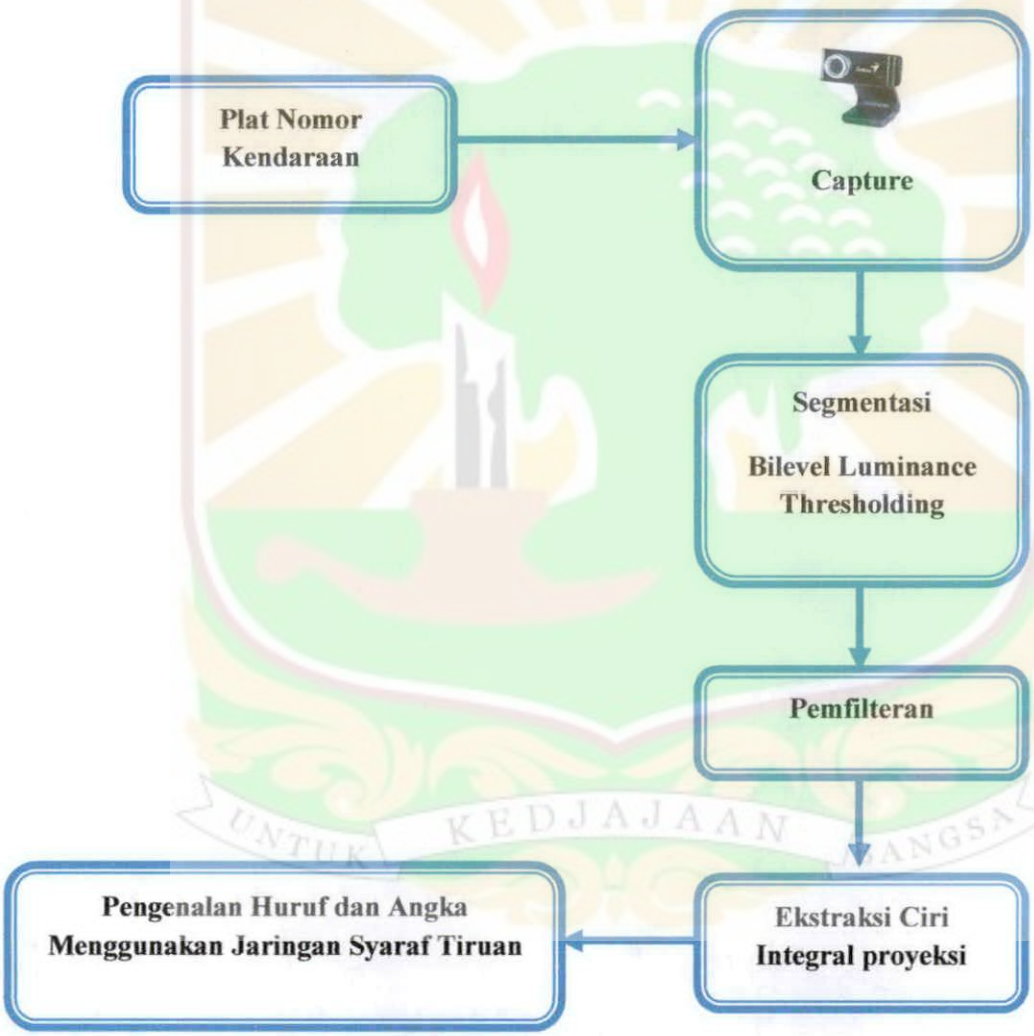
UNIVERSITAS ANDALAS
UNTUK KEDJAJAAN BANGSA

BAB IV

PERANCANGAN SISTEM

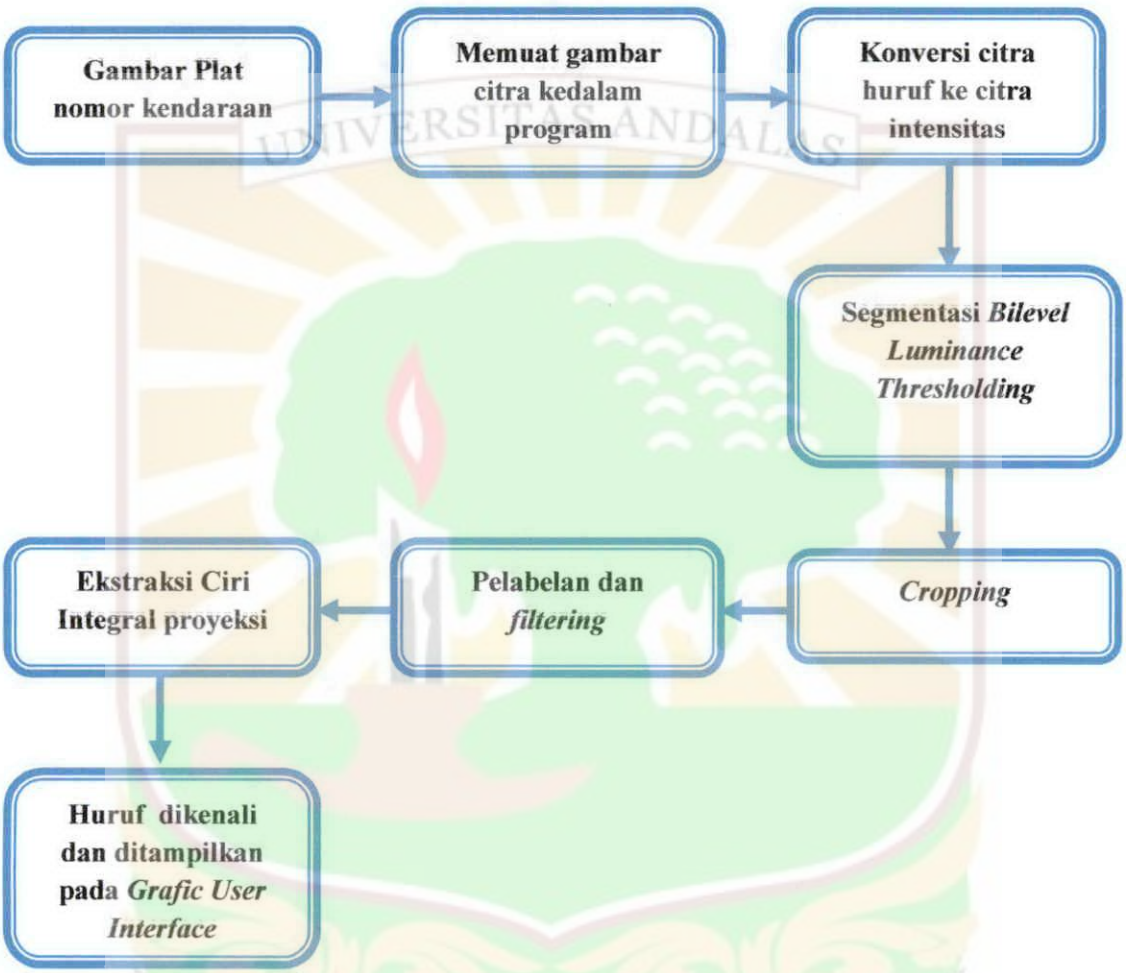
4.1 Blok Diagram Sistem

Pelatihan menggunakan jaringan syaraf tiruan dapat dilihat pada diagram blok berikut:



Gambar 4.1 Blok diagram rancangan sistem untuk proses pelatihan

Pengujian sistem ini dapat dilihat pada gambar 4.2, dimana sistem pengenalan huruf adan angka yang telah dilatih menggunakan jaringan syaraf tiruan *backpropagation* akan diuji menggunakan citra huruf yang dimasukkan ke dalam sistem ini.



Gambar 4.2. Blok diagram pengujian sistem pengenalan plat nomor kendaraan

4.1.1 Citra huruf

Citra huruf dalam penelitian ini memiliki spesifikasi yaitu:

- Semua huruf dan angka di*crop* dari plat kendaraan dengan objek berwarna putih dan latar berwarna hitam. Gambar citra huruf dapat dilihat pada gambar 4.3
- Plat yang digunakan merupakan plat standar kepolisian dengan font huruf untuk sebelum tahun 2011.
- Tiap karakter memiliki 5 bentuk yang berbeda dalam font yang sama

Gambar 4.3 Citra huruf C

4.1.2 *Capture*

Citra huruf yang akan diproses oleh sistem haruslah berbentuk data digital. Oleh karena itu citra huruf yang diperoleh terlebih dahulu diubah ke dalam bentuk data digital yaitu dengan cara di-*capture* menggunakan *webcam*. Pada penelitian ini digunakan *webcam* yang dapat dilihat pada gambar 4.4, dengan spesifikasi sebagai berikut:

Jenis : Genius iSlim 310

Interface to PC : *High Speed USB 2.0 (Compatible with USB 1.1)*

Tipe Sensor : *High Quality VGA Sensor (300k pixel CMOS)*

Resolusi : 640 x 480

Lens Features : *Premium Multi-Layer Adjustable Lens*

Frame Rate : CIF (320 x 240) up to 30fps ;VGA (640 x 480) up to 30fps

Gambar 4.4 *webcam Genius iSlim 310*

4.1.3 Konversi ke Citra Intensitas

Citra yang dihasilkan saat *capture* kamera berupa citra dengan ruang warna RGB, oleh karena itu dilakukan konversi warna RGB ke dalam nilai intensitas, hal ini dilakukan karena pada perangkat lunak Matlab 7.04 hanya dapat memproses citra intensitas ke citra biner dan tidak dapat memproses citra RGB ke citra biner. Gambar citra intensitas dapat dilihat pada gambar 4.5.

Gambar 4.5 Citra intensitas

4.1.4 Segmentasi

Segmentasi adalah suatu proses untuk memisahkan sejumlah objek dalam suatu citra dari latar belakangnya. Pada penelitian ini menggunakan metode segmentasi *bilevel luminance thresholding*. Metode ini memisahkan daerah gelap dan terang pada citra dengan menggunakan ruang warna intensitas atau *luminance* sebagai parameter *threshold*-nya. Gambar hasil segmentasi dapat dilihat pada gambar 4.6.

Gambar 4.6 Citra hasil segmentasi

4.1.5 Pelabelan dan *Filtering*

Citra huruf yang telah disegmentasi kemudian diproses ketahap selanjutnya yaitu pelabelan dan *filtering*. Hal ini bertujuan untuk meningkatkan kualitas citra yaitu dengan menghilangkan *noise* yang ada. Penghilangan *noise* perlu dilakukan karena memberi pengaruh pada saat pelatihan dan pengujian.

Pelabelan merupakan proses pemberian label pada setiap piksel terkoneksi yang membentuk suatu objek, dimana masing-masing objek yang berbeda akan diberi label yang berbeda. Proses labelisasi ini nantinya akan digunakan untuk menentukan objek mana yang akan tetap dipertahankan dan objek mana yang akan dihilangkan. Proses penghilangan objek merupakan bagian dari *filtering*.

4.1.6 Ekstraksi Ciri

Ekstraksi ciri adalah proses untuk mengambil ciri-ciri tertentu dari karakter huruf yang diamati. Pada penelitian ini, ekstraksi ciri menggunakan integral proyeksi dilakukan dengan cara menghitung jumlah piksel aktif pada setiap baris citra huruf kapital tersebut.

4.1.7 Pelatihan Jaringan Syaraf Tiruan *Backpropagation*

Metode yang digunakan pada penelitian ini adalah jaringan syaraf tiruan *backpropagation*. Struktur jaringan syaraf tiruan yang digunakan pada penelitian adalah sebagai berikut :

a) Pelatihan menggunakan 180 karakter yang diambil dari gambar *alphanumeric* (0-9 dan A-Z). Setiap karakter memiliki dua buah gambar yang berbeda.
b) Terdapat 71 *neuron* pada lapisan *input*. Hal ini disesuaikan dengan jumlah elemen matriks yang dihasilkan pada ekstraksi ciri pola huruf.
c) Terdapat 2 lapisan tersembunyi dengan jumlah neuron sebanyak 50 buah dan 2 buah *neuron*.
d) Jumlah neuron *output* sebanyak 6 neuron, yang ditentukan berdasarkan pertimbangan berikut :
    - Citra yang akan kita kenali adalah 26 huruf dan 10 angka dengan total 36 karakter
    - Setiap karakter huruf dikonversikan ke dalam bentuk pengkodean bit. Jumlah bit yang mungkin adalah 6 bit, yang menghasilkan 2^6 variasi *output* yang berbeda.

4.1.8 Pengujian Sistem

Hasil kinerja sistem nantinya akan dievaluasi melalui pengujian yang dilakukan berdasarkan hasil pelatihan yang telah dilakukan. Pada pengujian ini data

citra huruf dimasukkan kedalam sistem untuk dapat dikenali dan kemudian ditampilkan pada GUI *(Grafic User Interface)*.

Pengujian pada penelitian ini menggunakan Plat standar kepolisian yang memiliki ukuran 10,5x25 cm.

4.2. Algoritma Program

4.2.1 Segmentasi

Langkah-langkah yang digunakan dalam segmentasi citra huruf alfabet:

1. Baca citra (*frame* yang akan disegmentasi).
2. Ubah nilai RGB dengan range nilai antara 0 sampai dengan 1.
3. Ekstrak masing-masing komponen warna RGB.
4. Konversi warna RGB tersebut ke dalam nilai intensitas dengan persamaan: I = (r+b+g)/3, dengan nilai intensitas antara 0 sampai dengan 1.
5. Tentukan nilai *threshold* pada parameter intensitas yang tepat untuk warna objek.
6. Apabila ditemukan piksel dengan nilai intensitas di bawah nilai *threshold* (dianggap sebagai latar belakang) ubah nilai piksel menjadi 0 (warna hitam), sedangkan piksel yang memiliki nilai intensitas di atas nilai *threshold* (dianggap sebagai objek) nilainya diubah menjadi 1 (warna putih).

4.2.2 Pelabelan dan *Filtering*

Langkah-langkah yang dilakukan dalam menghilangkan *noise* adalah sebagai berikut:

1. Citra hasil segmentasi diberi label dengan memanfaatkan fasilitas fungsi yang tersedia pada Matlab, yaitu fungsi bwlabel.
2. Tentukan banyaknya komponen piksel terkoneksi (jumlah objek).
3. Tentukan jumlah piksel pada masing-masing objek.
4. Cari objek dengan jumlah piksel terbanyak.
5. Hapus objek yang tidak diperlukan yaitu dengan cara mengubah nilai piksel objek yang tidak dibutuhkan menjadi 0 dan nilai piksel objek yang dibutuhkan menjadi 1.

4.2.3 Ekstraksi ciri Integral Proyeksi

Langkah-langkah yang digunakan untuk mengekstraksi ciri pola huruf :

1. Setiap citra huruf yang diamati (yang berupa matriks) akan dihitung jumlah piksel aktif nya pada setiap kolom. Citra huruf yang berukuruan 71x113 piksel dibagi menjadi 113 daerah pengamatan yang mana jumlah piksel dari setiap kolom citra huruf dengan letak baris yang sama yakni dari baris 1 sampai 113 akan menjadi daerah pengamatan dan membentuk matriks yang baru dengan rincian yang dapat dilihat pada lampiran C.
2. Cari piksel aktif atau yang bernilai 1, pada masing-masing daerah pengamatan.
3. Tentukan jumlah piksel aktif pada setiap daerah pengamatan.
4. Bentuk matriks baru dengan nilai elemen berasal dari nilai jumlah piksel aktif pada daerah pengamatan.
5. Normalisasikan nilai masing-masing elemen matriks dengan rentang nilai antara 0 sampai dengan 1 yaitu dengan cara :
    1. Tentukan nilai elemen maksimum dari matriks tersebut
    2. Bagi masing-masing nilai elemen matriks dengan nilai elemen maksimum matriks tersebut.

3. Simpan matriks. Matriks ini akan menjadi *input* bagi jaringan syaraf tiruan.

4.2.4 Jaringan Syaraf Tiruan *Backpropagation*

a. Pelatihan

Dikarenakan jaringan syaraf tiruan yang digunakan pada penelitian ini adalah *backpropagation* maka terlebih dahulu perlu dibangun jaringan yang terdiri dari lapisan *input*, lapisan tersembunyi, lapisan *output*, fungsi aktivasi pada layer tersembunyi dan layer output, dan model pelatihan yang digunakan pada jaringan syaraf tiruan *backpropagation* menggunakan fungsi:

net = newff (minmax(P), [50 2 6], {'logsig' 'logsig' 'logsig'}, 'trainrp')

Dengan:

net = nama jaringan yang akan dibangun

newff = fungsi untuk membentuk jaringan *backpropagation*

minmax = fungsi untuk mencari nilai minimum dan maksimum dari matriks input P

P = matriks data input

50 = jumlah neuron pada layer tersembunyi pertama

2 = jumlah neuron pada layer tersembunyi kedua

6 = jumlah neuron pada layer output

logsig = fungsi aktivasi sigmoid biner pada layer tersembunyi pertama, kedua dan layer output

trainrp = model pelatihan dengan menggunakan metode resilient

Setelah selesai membangun jaringan, langkah selanjutnya yaitu menentukan parameter yang digunakan selama proses pelatihan, dengan fungsi:

```
net.trainParam.epochs    = 2000;
net.trainParam.goal      = 0.008;
net.trainParam.lr        = 0.7;
net.trainParam.mc        = 0.2;
net.trainParam.show      = 250;
```

Dengan:

net.trainParam.epochs = epoch yang digunakan yaitu 2000

net.trainParam.goal = toleransi yang digunakan yaitu 0,008

net.train.Param.lr = jumlah learning rate yang digunakan yaitu 0,7

net.trainParam.mc = jumlah momentum yang digunakan yaitu 0,2

net.trainParam.show = jumlah epoch yang ditampilkan pada grafik yaitu 250

Setelah selesai membentuk jaringan dan menentukan parameter-parameter yang akan digunakan dalam proses pelatihan, selanjutnya adalah pelatihan yang akan dilakukan dengan fungsi:

```
net = init (net)

[net,tr] = train(net,P,T)
```

Dengan:

init = fungsi untuk menginisialisasi bobot dan bias dengan nilai acak baru

train = fungsi untuk melatih jaringan

P = matriks data input

T = target

Pelatihan akan dilakukan sampai dicapai toleransi yang telah ditentukan yaitu 0.0001.

b. Pengujian

Fungsi pengujian yang digunakan pada jaringan syaraf tiruan *backpropagation* adalah:

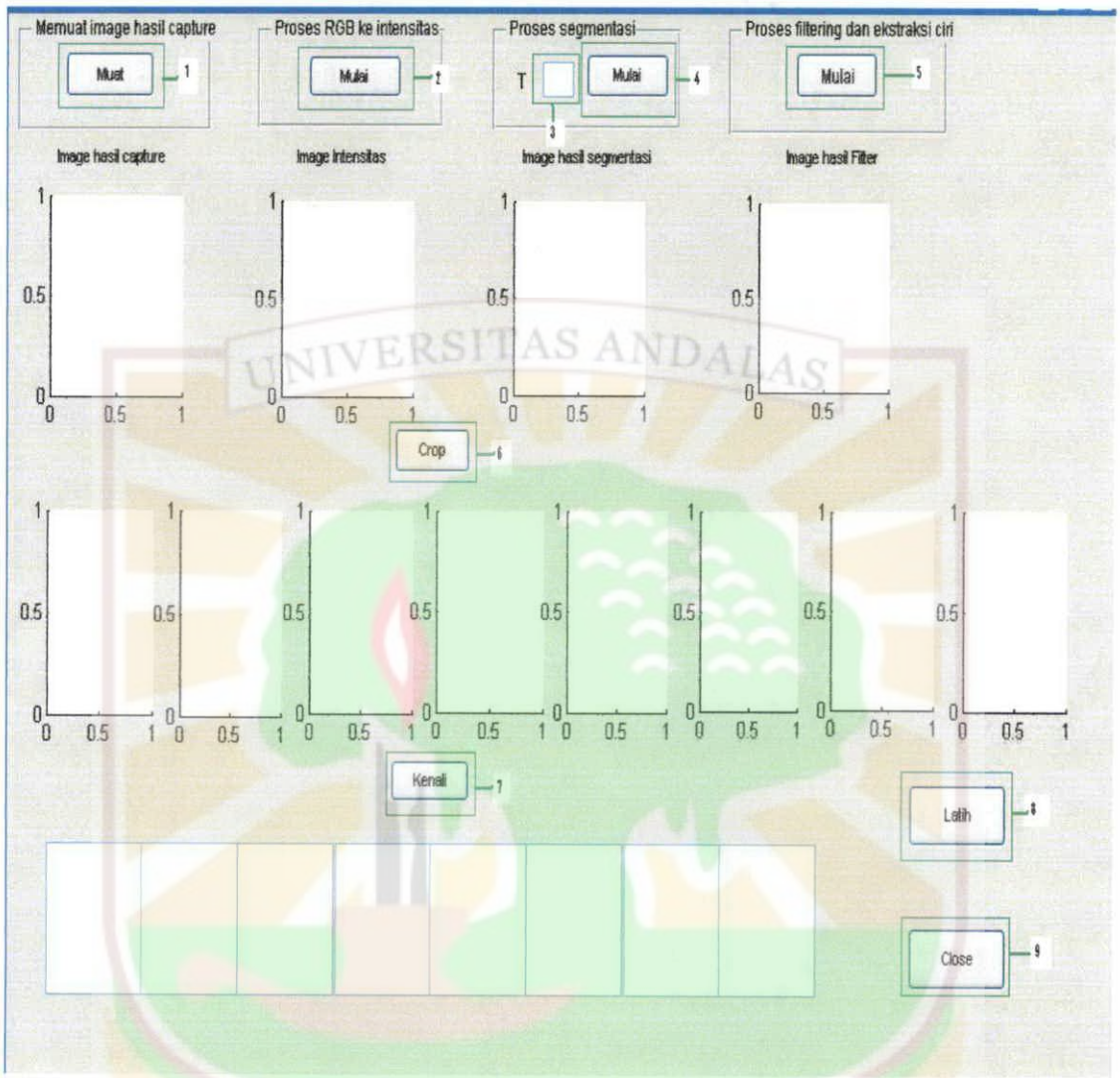
A = sim(net,P)

Pada tahap pengujian ini akan dapat diketahui sejauh mana sistem dapat mengenali pola huruf, terhadap pola yang telah dilatih.

4.3 Visualisasi Sistem

Sistem ini dibuat menggunakan perangkat lunak matlab 7.04 dan menggunakan GUI (*Grafic User Interface)* sebagai tampilan visual agar lebih mudah dalam penggunaannya. Gambar Tampilan GUI sistem ini dapat dilihat pada gambar 4.7.

UNIVERSITAS ANDALAS
UNTUK KEDJAJAAN BANGSA



Gambar 4.7 Tampilan GUI sistem pengenalan citra huruf

Komponen-komponen yang terdapat pada tampilan GUI sistem ini terdiri dari:

1. Tombol "Muat", yang terdapat pada proses capture.
2. Tombol "Mulai", yang terdapat pada proses RGB ke intensitas
3. Input untuk nilai threshold
4. Tombol "Mulai", untuk proses segmentasi.

5. Tombol "Mulai", untuk proses filtering dan ekstraksi ciri.

6. Tombol "Crop", untuk proses cropping

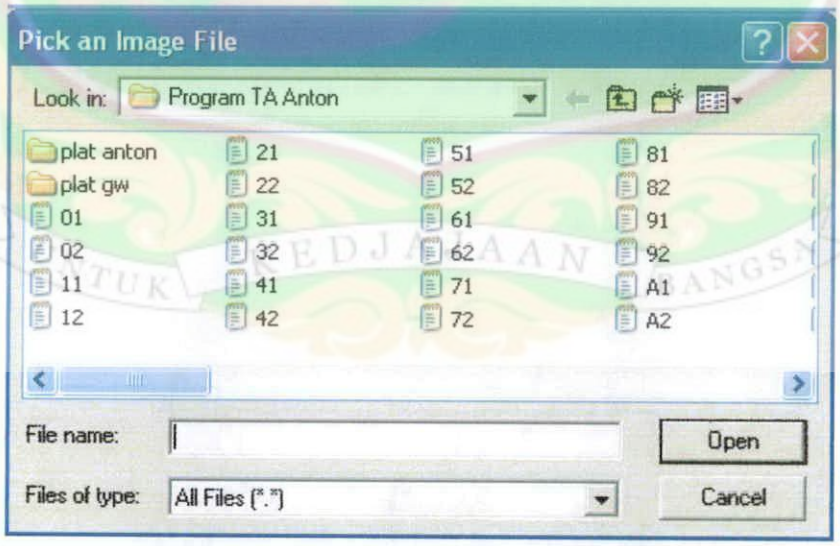
7. Tombol Kenali

8. Tombol "Latih".

9. Tombol "Close".

4.3.1 Pengambilan Data

Pada bagian ini, data citra huruf dengan format JPG yang merupakan hasil *capture* diambil dan kemudian dimasukkan kedalam sistem untuk ditampilkan dan diproses lebih lanjut. Tombol 'Muat' pada proses *capture* yang terdapat pada GUI sistem jika dieksekusi berfungsi untuk mengambil data citra huruf yang mana merupakan data masukkan untuk sistem ini. Proses pengambilan data menggunakan GUI dapat dilihat pada gambar 4.8.



Gambar 4.8 Tampilan GUI pengambilan data citra huruf

Setelah data citra huruf dipilih selanjutnya data tersebut akan ditampilkan pada GUI sistem yang dapat dilihat pada gambar 4.9, yang mana citra huruf ini merupakan citra RGB.

Gambar 4.9 Tampilan GUI pada proses memuat citra hasil capture

4.3.2 Proses Merubah Citra RGB ke Citra Intensitas

Pada bagian ini, data citra RGB akan diubah ke citra intensitas. Proses ini dilakukan dengan mengeksekusi tombol 'Mulai' yang terdapat pada proses RGB ke intensitas pada GUI sistem yang dapat dilihat pada gambar 4.10.

Gambar 4.10 Tampilan GUI pada proses konversi citra RGB ke citra intensitas

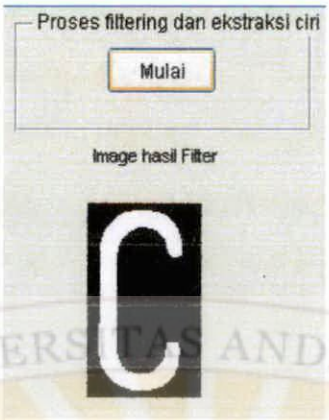
4.3.3 Proses Segmentasi

Pada bagian ini, data citra intensitas akan disegmentasi menjadi citra biner atau citra hitam putih. Proses ini dilakukan dengan mengeksekusi tombol 'Mulai' yang terdapat pada proses segmentasi pada GUI sistem yang dapat dilihat pada gambar 4.11. Sebelum melakukan proses segmentasi, terlebih dahulu kita masukkan nilai threshold yang akan digunakan dalam proses segmentasi.

Gambar 4.11 Tampilan GUI pada proses segmentasi

4.3.4 Proses Cropping, Proses Filtering dan Ekstraksi Ciri

Pada bagian ini, data citra hasil segmentasi atau citra biner akan mengalami proses *cropping,* yang bertujuan memisahkan karakter masing-masing plat sehingga berada pada piksel 71x113 untuk masing-masing karakter. Kemudian citra yang sudah melalui proses *cropping* difilter dan diekstraksi menggunakan metode Integral Proyeksi. Proses ini dilakukan dengan mengeksekusi tombol 'Mulai' yang terdapat pada proses filtering dan ekstraksi ciri pada GUI sistem yang dapat dilihat pada gambar 4.12.



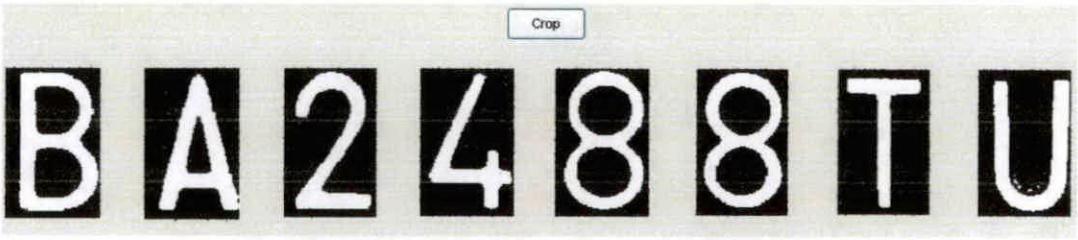
Gambar 4.12 Tampilan GUI pada proses filtering dan ekstraksi cirri

4.3.5 Proses Pelatihan Jaringan Syaraf Tiruan Backpropagation

Pada bagian ini, data citra hasil filtering dan ekstraksi ciri akan menjadi data masukkan bagi jaringan syaraf tiruan *backpropagation* yang mana selanjutnya akan dilatih. Proses pelatihan ini dilakukan dengan mengeksekusi tombol 'Latih' yang terdapat pada GUI sistem.

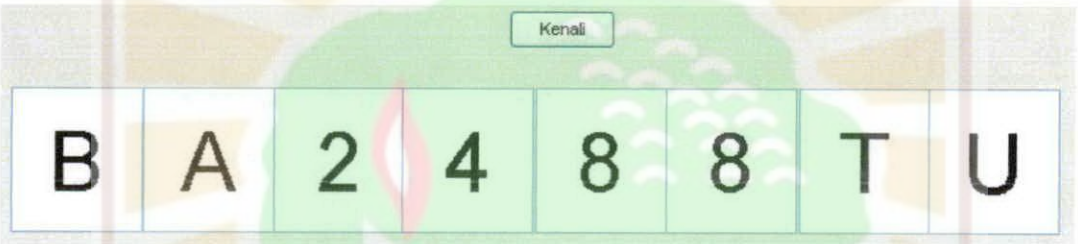
4.3.6 Pengujian Sistem

Pada bagian ini, data hasil pelatihan selanjutnya akan diuji berdasarkan hasil pelatihan yang diperoleh. Proses pengujian ini dilakukan hal yang sama dengan pada saat data di latih. Yaitu melalui proses pengambilan data, proses RGB ke intensitas, kemudian melewati proses segmentasi. Setelah data di segementasi, data di*crop* dengan ukuran tertentu yang sama tiap karakternya. Seperti yang dapat dilihat pada gambar gambar 4.13.



Gambar 4.13 Tampilan GUI pada proses cropping

Setelah dicropping, data dikenali untuk dapat mengenali karakter masing-masing huruf dan angka agar sesuai dengan data latih.

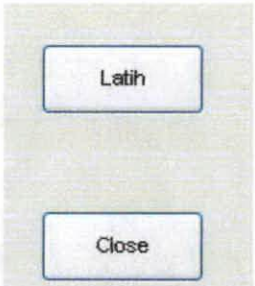


Gambar 4.14 Tampilan GUI pada proses pengujian

4.3.7 Mengakhiri Program

Pada bagian ini, merupakan akhir dari proses sistem pengenalan citra huruf dan untuk keluar dari GUI sistem. Proses ini dilakukan dengan mengeksekusi tombol 'Close' yang terdapat pada GUI sistem, proses mengakhiri dan keluar dari GUI sistem dapat dilihat pada gambar 4.15.

UNIVERSITAS ANDALAS
UNTUK KEDJAJAAN BANGSA



Gambar 4.15 Tampilan GUI pada proses mengakhiri program

4.4 Skenario Percobaan

Pada penelitian sistem pengenalan citra huruf ini dilakukan percobaan seperti pengenalan dengan menvariasikan nilai toleransi dengan jumlah *neuron* konstan. Kemudian menvariasikan nilai *threshold* dan nilai toleransi konstan.

UNIVERSITAS ANDALAS
UNTUK KEDJAJAAN BANGSA

BAB V
HASIL DAN PEMBAHASAN

5.1 Hasil Percobaan

5.1.1 Percobaan Pengujian Sistem dengan Jumlah Neuron pada Layer Tersembunyi Pertama 50 dan Nilai Toleransi 0.007

Pada percobaan ini bertujuan untuk melihat kemampuan sistem ketika parameter jumlah neuron pada layer tersembunyi konstan yaitu 50 dan nilai toleransi 0.007. Kemampuan sistem ini diukur berdasarkan tingkat keberhasilan dalam mengenali citra huruf yang dimasukkan. Hasil dari percobaan ini dapat dilihat pada pada gambar 5.1 dibawah ini.

Plat nomor	Hasil Output
BA2488TU	**BA2488TU**
BA2557JM	**BA2557JM**
BA3241UY	**BA3241UY**
BA3285AE	BH3115HE
BA4008CP	BA4008CF
BA6139SE	**BA6139SE**
BH2435RG	BH2130RG
BH5238LF	8A2238LF
BK5002WX	BKS002WX
BM4224DI	**BM4224DI**

Gambar 5.1 Tabel hasil pengujian untuk toleransi 0.007

Setelah dilakukan percobaan, dapat dilihat bahwa pada toleransi 0.007 ini sistem kesulitan membedakan beberapa karakter antara huruf A dan H, ini dapat dilihat pada plat dengan nomor polisi BA3285AE dan BH5238LF, dan beberapa angka yang tidak sesuai hasil output dengan inputnya.

Secara keseluruhan dari sepuluh data yang diujikan terlihat bahwa pada toleransi 0.007 diperoleh tingkat keberhasilan sebesar 50%.

5.1.2 Percobaan Pengujian Sistem dengan Jumlah Neuron pada Layer Tersembunyi Pertama 50 dan Nilai Toleransi 0.008

Pada percobaan ini di*setting* nilai toleransinya 0.008 dengan jumlah neuron pada layer tersembunyi konstan yaitu 50. Kemampuan sistem ini diukur berdasarkan tingkat keberhasilan dalam mengenali citra huruf yang dimasukkan. Hasil dari percobaan ini dapat dilihat pada pada gambar 5.2 dibawah ini.

Plat nomor	Hasil Output
BA2488TU	**BA2488TU**
BA2557JM	**BA2557JM**
BA3241UY	**BA3241UY**
BA3285AE	**BA3285AE**
BA4008CP	**BA4008CP**
BA6139SE	**BA6139SE**
BH2435RG	BA2439AG
BH5238LF	BA3037LF

BA3241UY	BA3241UT
BA3285AE	BA3285AA
BA4008CP	BA4008CF
BA6139SE	BA61395E
BH2435RG	BH2445RG
BH5238LF	**BH5238LF**
BK5002WX	BH5002WX
BM4224DI	BM4224BI

Gambar 5.3 Tabel hasil pengujian untuk toleransi 0.009

Setelah dilakukan percobaan, dengan ditingkatkannya toleransi menjadi 0.009 terjadi penurunan dalam pengenalan karakter pada plat yang menjadi data uji. Plat yang bias dikenali hanya tiga buah yaitu nomor plat BA2488TU, BA2557JM, dan BH5238LF. Secara keseluruhan dari sepuluh data yang diujikan terlihat bahwa pada toleransi 0.009 diperoleh tingkat keberhasilan hanya sebesar 30%.

5.1.4 Percobaan Pengujian Sistem dengan Jumlah Neuron pada Layer Tersembunyi Pertama 50 dan Nilai Toleransi 0.0001

Pada percobaan ini di*setting* nilai toleransinya 0.0001 dengan jumlah neuron pada layer tersembunyi konstan yaitu 50. Kemampuan sistem ini diukur berdasarkan tingkat keberhasilan dalam mengenali citra huruf yang dimasukkan. Hasil dari percobaan ini dapat dilihat pada pada gambar 5.4 dibawah ini.

Plat nomor	Hasil Output
BA2488TU	**BA2488TU**
BA2557JM	**BA2557JM**
BA3241UY	BH3241UT
BA3285AE	BA3288AE
BA4008CP	BA4088CP
BA6139SE	BA6119SS
BH2435RG	BH2435DG
BH5238LF	BA1138LL
BK5002WX	BK5OO2WX
BM4224DI	BM4224MI

Gambar 5.4 Tabel hasil pengujian untuk toleransi 0.0001

Setelah dilakukan percobaan, dengan ditingkatkannya toleransi menjadi 0.0001 terjadi penurunan dalam pengenalan karakter pada plat yang menjadi data uji. Plat yang bisa dikenali hanya dua buah yaitu nomor plat BA2488TU dan BA2557JM. Secara keseluruhan dari sepuluh data yang diujikan terlihat bahwa pada toleransi 0.0001 diperoleh tingkat keberhasilan hanya sebesar 20%.

5.1.5 Percobaan Pengujian Sistem dengan Nilai Toleransi 0.008 dan *Threshold* 0.35

Tujuan dari percobaan ini adalah untuk melihat pengaruh nilai *Threshold* terhadap kinerja sistem dan dapat mengetahui dimana letak kinerja terbaik dan komposisi yang efektif dari sistem secara keseluruhan. Hasil yang diperoleh saat nilai threshold disetting menjadi 0.35 dapat dilihat pada gambar 5.5 dibawah ini.

Plat nomor	Hasil Output
BA2488TU	B...88YU
BA2557JM	BA25..DM
BA3241UY	BA3222UT
BA3285AE	BA...8AE
BA4008CP	BA4..8CP
BA6139SE	BA6130EE
BH2435RG	BH2255DG
BH5238LF	BH5233LL
BK5002WX	BX5002WK
BM4224DI	BM4...DI

Gambar 5.4 Tabel hasil pengujian untuk *Threshold* 0.35

Dapat dilihat pada tabel diatas, sistem tidak dapat mengenali karakter pada plat secara utuh. Bahkan pada beberapa plat ada karakter yang tidak terbaca. Dari sepuluh data yang diujikan dengan nilai threshold 0.35 tidak satupun plat yang dikenali secara utuh. Persentase keberhasilan pada percobaan ini adalah 0%.

5.1.6 Percobaan Pengujian Sistem dengan Nilai Toleransi 0.008 dan *Threshold* 0.4

Hasil yang diperoleh saat nilai threshold di*setting* menjadi 0.4 dapat dilihat pada gambar 5.6 dibawah ini.

Plat nomor	Hasil Output
BA2488TU	**BA2488TU**
BA2557JM	**BA2557JM**
BA3241UY	**BA3241UY**
BA3285AE	**BA3285AE**
BA4008CP	**BA4008CP**
BA6139SE	**BA6139SE**
BH2435RG	BA2439AG
BH5238LF	BA3037LF
BK5002WX	BA2049UG
BM4224DI	BH2044BI

Gambar 5.6 Tabel hasil pengujian untuk *threshold* 0.4

Dapat dilihat pada tabel diatas, sistem dapat mengenali sebagian besar karakter pada plat. Ini membuktikan bahwa nilai *threshold* yang terbaik untuk sistem ini berada pada nilai 0.4. Dari sepuluh data yang diujikan dengan nilai threshold 0.4 persentase keberhasilan yang diperoleh adalah 60%.

5.1.7 Percobaan Pengujian Sistem dengan Nilai Toleransi 0.008 dan *Threshold* 0.45

Hasil yang diperoleh saat nilai threshold di*setting* menjadi 0.45 dapat dilihat pada gambar 5.7 dibawah ini.

Plat nomor	Hasil Output
BA2488TU	BA2080SU
BA2557JM	**BA2557JM**
BA3241UY	**BA3241UY**
BA3285AE	RA328SRE
BA4008CP	BA40O8CP
BA6139SE	BH6139DE
BH2435RG	BHZA35RG
BH5238LF	BH5238LP
BK5002WX	BK50O8WX
BM4224DI	BM4222D1

Gambar 5.7 Tabel hasil pengujian untuk *threshold* 0.45

Dapat dilihat pada tabel diatas, sistem mengalami penurunan dalam pengenalan karakter pada plat. Hanya terdapat dua buah plat yang memiliki kesamaan dengan inputnya. Dari sepuluh data yang diujikan dengan nilai threshold 0.45 persentase keberhasilan yang diperoleh adalah 20%.

5.2 Analisa

Dari hasil percobaan yang diperoleh, pada sistem ini mempunyai suatu nilai titik balik pada tingkat keberhasilan. Pada saat nilai toleransi semakin diperbesar, maka sistem akan menjadi tidak stabil dan menyebabkan penurunan tingkat keberhasilan pada sistem ini.

Banyaknya pelatihan yang dilakukan berdampak pada tingkat keberhasilan pada sistem ini, semakin banyak pelatihan yang dilakukan maka tingkat keberhasilan juga semakin tinggi namun hal ini juga bergantung dari nilai toleransi yang ditetapkan. Apabila pada saat dilatih hasil yang diperoleh tidak lagi menunjukkan hasil yang memuaskan. Maka sistem dianggap tidak mampu lagi megeneralisasi data lagi.

Nilai *threshold* sangat berpengaruh terhadap kemampuan sistem ini dalam mengenali citra huruf, kesalahan pada pemberian nilai *threshold* dalam proses segmentasi akan membuat citra huruf tidak dikenali oleh sistem. Pada sistem ini nilai *threshold* yang digunakan yaitu $0.35 \leq T \leq 0.45$.

Faktor penyebab dari kesalahan ini adalah kemampuan pada jaringan syaraf tiruan yang sudah mulai mengambil sifat yang hanya dimiliki spesifik oleh data pelatihan (tapi tidak dimiliki oleh data pengujian) sehingga terjebak pada kondisi maksimum lokal yakni kondisi dimana target yang ingin di capai sistem tidak ditemukan karena kesalahan pada pelatihan yang sudah meningkat dan tidak stabil yang menjadikan sistem mengambil keputusan yang dianggap mendekati pola target yang diinginkan.

Pengaturan cropping cukup berpengaruh untuk tingkat keberhasilan sistem ini. Penggunaan plat yang memiliki huruf dan angka yang sedikit berbeda walaupun dalam bentuk font yang sama, pencahayaan yang kurang bagus sehingga memberikan efek pantulan pada plat, kualitas *webcam* yang digunakan, menjadi faktor kesalahan dalam sistem.

MILIK
UPT PERPUSTAKAAN
UNIVERSITAS ANDALAS

BAB VI

PENUTUP

6.1. Simpulan

Berdasarkan analisa terhadap hasil percobaan yang didapat, maka dapat diambil beberapa simpulan sebagai berikut:

1. Sistem mampu mendapatkan beberapa ciri atau fitur bentuk dari citra huruf dan angka pada plat.
2. Tingkat keberhasilan terbesar yaitu 60% pada toleransi 0.008. ketika nilai toleransi diperbesar maka diperoleh tingkat keberhasilan yang semakin kecil yaitu toleransi 0.001 mempunyai tingkat keberhasilan 20%
3. Nilai threshold yang terbaik untuk penelitian ini berada pada nilai 0.4

6.2. Saran

Setelah menganalisa cara kerja sistem dan keluaran sistem, untuk penelitian dan pengembangan sistem ini selanjutnya, penulis menyarankan beberapa hal:

1. Kemampuan sistem dapat dikembangkan dengan metode ekstraksi ciri dan metode pengenalan yang lain.
2. Adanya pengembangan dari sistem ini agar dapat bekerja secara *online* dan meng-*update* parameter jaringan syaraf tiruan *backpropagation* secara otomatis.

DAFTAR KEPUSTAKAAN

[1]. Arman, Arry Akhmad.2002. *Konversi dari Teks ke Fonem.* ITB:Bandung

[2]. Awaludin, Ujang Fajar. 2006. *Mendeteksi Wajah Menggunakan Sketsa.* Skripsi, Jurusan Teknologi Informasi Politeknik Elektronika Negeri Surabaya : Surabaya.

[3]. Brunelli, Roberto . 2009. *Template Matching Techniques in Computer Vision Theory and Ppractice.* Itali : A John Wiley and sons Ltd

[4]. Hendrawan. 2003. "*Pendahuluan: Handout Mata Kuliah Pemrosesan Citra Dan Video*". Bandung: Teknik Elektro ITB

[5]. Hermawan, Arief. 2006. *Jaringan Saraf Tiruan, Teori dan Aplikasi.* Andi: Yogyakarta.

[6]. Kasabov, Nikolai K. 1996. *Foundation of Neural Networks, Fuzzy Systems, and Knowlege Engineering.* Second printing. The MIT Press: London.

[7]. Khaidir, Muhammad. 2011. *Sistem Pengenalan Huruf Alfabet Kapital dengan Variasi Ukuran dan Jarak menggunakan Ekstraksi Ciri Integral Proyeksi dan Pengenalan dengan Jaringan Syaraf Tiruan Backpropagation.* Tugas Akhir. Jurusan Teknik Elektro Universitas Andalas : Padang

[8]. Munir, Renaldi. 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik.* Informatika : Bandung.

[9]. Pratt, William K. 2001. *Digital Image Processing, 3rd Ed.* John Wiley & Sons: New York.

[10]. Priyatma, J Eka dan St Eko Hari Parmadi.. 2004. *Perluasan Algoritma Pengenal Huruf Metode Tupel-N Memakai Logika Kabur.* Jurusan Ilmu Komputer Fakultas MIPA Universitas Sanata Dharma : Yogyakarta.

[11]. SiaKng, Jong Jek. 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab.* Andi: Yogyakarta.

[12]. Sigit, Riyanto,dkk.2005.*Step by Step Pengolahan Citra Digital.* Yogyakarta: Andi

[13]. SutamaMandasari, Fibriana. 2006. *Sistem Deteksi Penempatan Barang Melalui Kamera*. Skripsi, Jurusan Teknologi Informasi Politeknik Elektronika Negeri Surabaya : Surabaya.

[14]. Sutama, Willy.2008. *Pengenalan Huruf dengan Metode Integral Proyeksi.* Tugas Akhir. Jurusan Teknik Elektro Universitas Andalas : Padang

[15]. Syahputra, Hendra. 2010. *Perancangan dan Implementasi Sistem Pengenalan Gambar Kata dan Bilangan dengan Keluaran Suara.* Tugas Akhir. Jurusan Teknik Elektro Universitas Andalas : Padang

[16]. Yani, Eli. 2005. *Pengantar Jaringan Syaraf Tiruan.* http://trirezqiariantoro.files.wordpress.com/2007/05/jaringan_syaraf_tiruan.pdf

[17]. Yu-Jin, Zhang. 2006. *Advance in Image and Video Segmentation*. Harshey: IRM Press

LAMPIRAN A

LISTING PROGRAM MENGGUNAKAN MATLAB 7.04

***Listing Program Pengambilan Data**

```
function pushbutton2_Callback(hObject, eventdata, handles)
 [filename, pathname] = uigetfile({'*.*';'*.bmp';'*.jpg';'*.gif'}, 'Pick an Image File');
S = imread([pathname,filename]);
axes(handles.axes3);
imshow(S);
handles.S = S;
guidata(hObject, handles);
```

***Listing Program Citra Intensitas**

```
function pushbutton3_Callback(hObject, eventdata, handles)
S = handles.S;
axes(handles.axes3);
rgb = S;
rgb=im2double(rgb);
 r=rgb(:,:,1);
g=rgb(:,:,2);
b=rgb(:,:,3);
 I=(r+g+b)/3;
 axes(handles.axes8);
imshow(I);
handles.I = I;
guidata(hObject, handles);
 save data1 I;
```

***Listing Program Segmentasi**

```
function pushbutton4_Callback(hObject, eventdata, handles)
load data1;
I = handles.I;
axes(handles.axes8);
T = str2double(get(handles.T_input,'String'));
I1 = I;
```

```
I1(I>=T) = 1;
I1(I<T)  = 0;
axes(handles.axes9);
imshow(I1)
handles.I1 = I1;
guidata(hObject, handles);
save data2 I1;
```

***Listing Program Pelabelan dan Filtering**

```
function pushbutton10_Callback(hObject, eventdata, handles)
load data2;
 I1 = handles.I1;
axes(handles.axes9);
L = bwlabel (I1);
max (max(L));
k = ans;
for x=1:k
    P = L(L==x);
   lab (x) = sum (P)/x;
 end;
 t = lab;
max (t);
find (t==ans);
 l = ans;
la = (L==0);
lb = (L~=0);
F=L;
F(la)=0;
F(lb)=1;
 save data3 F;
axes(handles.axes10);
imshow(F)
handles.F = F;
guidata(hObject, handles);
load data3;
```

```
p17= sum (ans);

f18 = F (1:113,18);
f18 (f18==1);
sum (ans);
p18= sum (ans);

f19 = F (1:113,19);
f19 (f19==1);
sum (ans);
p19= sum (ans);

f20 = F (1:113,20);
f20 (f20==1);
sum (ans);
p20= sum (ans);

f21 = F (1:113,21);
f21 (f21==1);
sum (ans);
p21= sum (ans);

f22 = F (1:113,22);
f22 (f22==1);
sum (ans);
p22= sum (ans);

f23= F (1:113,23);
f23 (f23==1);
sum (ans);
p23= sum (ans);

f24 = F (1:113,24);
f24 (f24==1);
sum (ans);
p24= sum (ans);

f25 = F (1:113,25);
f25 (f25==1);
sum (ans);
p25= sum (ans);

f26 = F (1:113,26);
f26 (f26==1);
sum (ans);
p26= sum (ans);

f27 = F (1:113,27);
f27 (f27==1);
sum (ans);
p27= sum (ans);

f28 = F (1:113,28);
f28 (f28==1);
sum (ans);
p28= sum (ans);

f29 = F (1:113,29);
f29 (f29==1);
sum (ans);
p29= sum (ans);

f30 = F (1:113,30);
f30 (f30==1);
sum (ans);
p30= sum (ans);

f31 = F (1:113,31);
f31 (f31==1);
sum (ans);
p31= sum (ans);

f32 = F (1:113,32);
f32 (f32==1);
sum (ans);
p32= sum (ans);

f33 = F (1:113,33);
f33 (f33==1);
sum (ans);
p33= sum (ans);

f34 = F (1:113,34);
f34 (f34==1);
sum (ans);
p34= sum (ans);

f35 = F (1:113,35);
```

```
f35 (f35==1);
sum (ans);
p35= sum (ans);

f36 = F (1:113,36);
f36 (f36==1);
sum (ans);
p36= sum (ans);

f37 = F (1:113,37);
f37 (f37==1);
sum (ans);
p37= sum (ans);

f38 = F (1:113,38);
f38 (f38==1);
sum (ans);
p38= sum (ans);

f39 = F (1:113,39);
f39 (f39==1);
sum (ans);
p39= sum (ans);

f40 = F (1:113,40);
f40 (f40==1);
sum (ans);
p40= sum (ans);

f41 = F (1;113,41);
f41 (f41==1);
sum (ans);
p41= sum (ans);

f42 = F (1:113,42);
f42 (f42==1);
sum (ans);
p42= sum (ans);

f43 = F (1:113,43);
f43 (f43==1);
sum (ans);
p43= sum (ans);

f44 = F (1:113,44);
f44 (f44==1);
sum (ans);
p44= sum (ans);

f45 = F (1:113,45);
f45 (f45==1);
sum (ans);
p45= sum (ans);

f46 = F (1:113,46);
f46 (f46==1);
sum (ans);
p46= sum (ans);

f47 = F (1:113,47);
f47 (f47==1);
sum (ans);
p47= sum (ans);

f48 = F (1:113,48);
f48 (f48==1);
sum (ans);
p48= sum (ans);

f49= F (1:113,49);
f49 (f49==1);
sum (ans);
p49= sum (ans);

f50 = F (1:113,50);
f50 (f50==1);
sum (ans);
p50= sum (ans);

f51 = F (1:113,51);
f51 (f51==1);
sum (ans);
p51= sum (ans);


f52 = F (1:113,52);
f52 (f52==1);
sum (ans);
```

```
P13 = load ('C3.txt');
P14 = load ('C4.txt');
P15 = load ('C5.txt');
P16 = load ('D1.txt');
P17 = load ('D2.txt');
P18 = load ('D3.txt');
P19 = load ('D4.txt');
P20 = load ('D5.txt');
P21 = load ('E1.txt');
P22 = load ('E2.txt');
P23 = load ('E3.txt');
P24 = load ('E4.txt');
P25 = load ('E5.txt');
P26 = load ('F1.txt');
P27 = load ('F2.txt');
P28 = load ('F3.txt');
P29 = load ('F4.txt');
P30 = load ('F5.txt');
P31 = load ('G1.txt');
P32 = load ('G2.txt');
P33 = load ('G3.txt');
P34 = load ('G4.txt');
P35 = load ('G5.txt');
P36 = load ('H1.txt');
P37 = load ('H2.txt');
P38 = load ('H3.txt');
P39 = load ('H4.txt');
P40 = load ('H5.txt');
P41 = load ('I1.txt');
P42 = load ('I2.txt');
P43 = load ('I3.txt');
P44 = load ('I4.txt');
P45 = load ('I5.txt');
P46 = load ('J1.txt');
P47 = load ('J2.txt');
P48 = load ('J3.txt');
P49 = load ('J4.txt');
P50 = load ('J5.txt');
P51 = load ('K1.txt');
P52 = load ('K2.txt');
P53 = load ('K3.txt');
P54 = load ('K4.txt');
P55 = load ('K5.txt');
P56 = load ('L1.txt');
```

```
P57 = load ('L2.txt');
P58 = load ('L3.txt');
P59 = load ('L4.txt');
P60 = load ('L5.txt');
P61 = load ('M1.txt');
P62 = load ('M2.txt');
P63 = load ('M3.txt');
P64 = load ('M4.txt');
P65 = load ('M5.txt');
P66 = load ('N1.txt');
P67 = load ('N2.txt');
P68 = load ('N3.txt');
P69 = load ('N4.txt');
P70 = load ('N5.txt');
P71 = load ('O1.txt');
P72 = load ('O2.txt');
P73 = load ('O3.txt');
P74 = load ('O4.txt');
P75 = load ('O5.txt');
P76 = load ('P1.txt');
P77 = load ('P2.txt');
P78 = load ('P3.txt');
P79 = load ('P4.txt');
P80 = load ('P5.txt');
P81 = load ('Q1.txt');
P82 = load ('Q2.txt');
P83 = load ('Q3.txt');
P84 = load ('Q4.txt');
P85 = load ('Q5.txt');
P86 = load ('R1.txt');
P87 = load ('R2.txt');
P88 = load ('R3.txt');
P89 = load ('R4.txt');
P90 = load ('R5.txt');
P91 = load ('S1.txt');
P92 = load ('S2.txt');
P93 = load ('S3.txt');
P94 = load ('S4.txt');
P95 = load ('S5.txt');
P96 = load ('T1.txt');
P97 = load ('T2.txt');
P98 = load ('T3.txt');
P99 = load ('T4.txt');
P100= load ('T5.txt');
```

```
P101= load ('U1.txt');
P102= load ('U2.txt');
P103= load ('U3.txt');
P104= load ('U4.txt');
P105= load ('U5.txt');
P106= load ('V1.txt');
P107= load ('V2.txt');
P108= load ('V3.txt');
P109= load ('V4.txt');
P110= load ('V5.txt');
P111= load ('W1.txt');
P112= load ('W2.txt');
P113= load ('W3.txt');
P114= load ('W4.txt');
P115= load ('W5.txt');
P116= load ('X1.txt');
P117= load ('X2.txt');
P118= load ('X3.txt');
P119= load ('X4.txt');
P120= load ('X5.txt');
P121= load ('Y1.txt');
P122= load ('Y2.txt');
P123= load ('Y3.txt');
P124= load ('Y4.txt');
P125= load ('Y5.txt');
P126= load ('Z1.txt');
P127= load ('Z2.txt');
P128= load ('Z3.txt');
P129= load ('Z4.txt');
P130= load ('Z5.txt');
P131= load ('01.txt');
P132= load ('02.txt');
P133= load ('03.txt');
P134= load ('04.txt');
P135= load ('05.txt');
P136= load ('11.txt');
P137= load ('12.txt');
P138= load ('13.txt');
P139= load ('14.txt');
P140= load ('15.txt');
P141= load ('21.txt');
P142= load ('22.txt');
P143= load ('23.txt');
P144= load ('24.txt');
```

```
P145= load ('25.txt');
P146= load ('31.txt');
P147= load ('32.txt');
P148= load ('33.txt');
P149= load ('34.txt');
P150= load ('35.txt');
P151= load ('41.txt');
P152= load ('42.txt');
P153= load ('43.txt');
P154= load ('44.txt');
P155= load ('45.txt');
P156= load ('51.txt');
P157= load ('52.txt');
P158= load ('53.txt');
P159= load ('54.txt');
P160= load ('55.txt');
P161= load ('61.txt');
P162= load ('62.txt');
P163= load ('63.txt');
P164= load ('64.txt');
P165= load ('65.txt');
P166= load ('71.txt');
P167= load ('72.txt');
P168= load ('73.txt');
P169= load ('74.txt');
P170= load ('75.txt');
P171= load ('81.txt');
P172= load ('82.txt');
P173= load ('83.txt');
P174= load ('84.txt');
P175= load ('85.txt');
P176= load ('91.txt');
P177= load ('92.txt');
P178= load ('93.txt');
P179= load ('94.txt');
P180= load ('95.txt');

P = [P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15 P16 P17 P18 P19 P20
P21 P22 P23 P24 P25 P26 P27 P28 P29 P30 P31 P32 P33 P34 P35 P36 P37 P38 P39
P40 P41 P42 P43 P44 P45 P46 P47 P48 P49 P50 P51 P52 P53 P54 P55 P56 P57 P58
P59 P60 P61 P62 P63 P64 P65 P66 P67 P68 P69 P70 P71 P72 P73 P74 P75 P76 P77
P78 P79 P80 P81 P82 P83 P84 P85 P86 P87 P88 P89 P90 P91 P92 P93 P94 P95 P96
P97 P98 P99 P100 P101 P102 P103 P104 P105 P106 P107 P108 P109 P110 P111
```

P112 P113 P114 P115 P116 P117 P118 P119 P120 P121 P122 P123 P124 P125 P126
P127 P128 P129 P130 P131 P132 P133 P134 P135 P136 P137 P138 P139 P140 P141
P142 P143 P144 P145 P146 P147 P148 P149 P150 P151 P152 P153 P154 P155 P156
P157 P158 P159 P160 P161 P162 P163 P164 P165 P166 P167 P168 P169 P170 P171
P172 P173 P174 P175 P176 P177 P178 P179 P180]

ta = [0;0;0;0;0;0];
tb = [0;0;0;0;0;1];
tc = [0;0;0;0;1;0];
td = [0;0;0;1;0;0];
te = [0;0;1;0;0;0];
tf = [0;1;0;0;0;0];
tg = [1;0;0;0;0;0];
th = [1;1;0;0;0;0];
ti = [1;0;1;0;0;0];
tj = [1;0;0;1;0;0];
tk = [1;0;0;0;1;0];
tl = [1;0;0;0;0;1];
tm = [0;0;0;0;1;1];
tn = [0;0;0;1;0;1];
to = [0;0;1;0;0;1];
tp = [0;1;0;0;0;1];
tq = [0;0;0;1;1;0];
tr = [0;0;1;0;1;0];
ts = [0;1;0;0;1;0];
tt = [0;1;0;1;0;0];
tu = [0;1;1;0;0;0];
tv = [1;1;1;0;0;0];
tw = [1;1;0;1;0;0];
tx = [1;1;0;0;1;0];
ty = [1;1;0;0;0;1];
tz = [1;0;1;1;0;0];
t0 = [1;0;1;0;1;0];
t1 = [1;0;1;0;0;1];
t2 = [1;0;0;1;0;1];
t3 = [1;0;0;0;1;1];
t4 = [1;0;0;1;1;0];
t5 = [0;1;1;1;0;0];
t6 = [0;1;1;0;1;0];
t7 = [0;1;1;0;0;1];
t8 = [1;1;1;1;0;0];
t9 = [1;1;1;0;0;1];

```
T = [ta ta ta ta ta ta tb tb tb tb tb tc tc tc tc tc td td td td td te te te te te tf tf tf tf tf tg tg tg
tg tg th th th th th ti ti ti ti ti tj tj tj tj tj tk tk tk tk tk tl tl tl tl tl tm tm tm tm tm tn tn tn
tn tn to to to to to tp tp tp tp tp tq tq tq tq tq tr tr tr tr tr ts ts ts ts ts tt tt tt tt tt tu tu tu tu
tu tv tv tv tv tv  tw tw tw tw tw tx tx tx tx tx ty ty ty ty ty tz tz tz tz tz t0 t0 t0 t0 t0 t1
t1 t1 t1 t1 t2 t2 t2 t2 t2 t3 t3 t3 t3 t3 t4 t4 t4 t4 t4 t5 t5 t5 t5 t5 t6 t6 t6 t6 t6 t7 t7 t7 t7
t7 t8 t8 t8 t8 t8 t9 t9 t9 t9 t9]


net = newff(minmax(P),[50 2 6],{'logsig' 'logsig' 'logsig'},'trainrp')

BobotAwal_Input          = net.IW{1,1}
BobotAwal_Bias_Input     = net.b{1,1}
BobotAwal_Lapisan1       = net.LW{2,1}
BobotAwal_Bias_Lapisan1  = net.b{2,1}
BobotAwal_Lapisan2       = net.LW{3,2}
BobotAwal_Bias_Lapisan2  = net.b{3,1}

 net.trainParam.epochs   = 2000;
 net.trainParam.goal     = 0.008;
 net.trainParam.lr       = 0.7;
 net.trainParam.mc       = 0.2;
 net.trainParam.show     = 250;

net = init (net)
[net,tr] = train(net,P,T)
save trainning ta tb tc td te tf tg th ti tj tk tl tm tn to tp tq tr ts tt tu tv tw txt y tz t0 t1 t2
t3 t4 t5 t6 t7 t8 t9 P net

BobotAkhir_Input         = net.IW{1,1}
BobotAkhir_Bias_Input    = net.b{1,1}
BobotAkhir_Lapisan1      = net.LW{2,1}
BobotAkhir_Bias_Lapisan1 = net.b{2,1}
BobotAkhir_Lapisan2      = net.LW{3,2}
BobotAkhir_Bias_Lapisan2  = net.b{3,1}
```

***Listing Program Cropping**

- **Cropping Karakter pertama**

```
load data3;
 I1 = handles.I1;
axes(handles.axes9);

L = bwlabel (I1);

max (max(L));
k = ans;

for x=1:k
    P = L(L==x);
   lab (x) = sum (P)/x;

end;

t = lab;
max (t);
find (t==ans);

 l = ans;
la = (L==0);
lb = (L~=0);
F=L;
F(la)=0;
F(lb)=1;
F=F(15:128,5:75);

 size(F)

save data3 F;
axes(handles.axes11);
imshow(F)
handles.F = F;
guidata(hObject, handles);

load data3;


f01 = F (1:113,1);
f01 (f01==1);
sum (ans);
p01= sum (ans);

f02 = F (1:113,2);
f02 (f02==1);
sum (ans);
p02= sum (ans);

f03 = F (1:113,3);
f03 (f03==1);
sum (ans);
p03= sum (ans);

f04 = F (1:113,4);
f04 (f04==1);
sum (ans);
p04= sum (ans);

f05 = F (1:113,5);
f05 (f05==1);
sum (ans);
p05= sum (ans);

f06 = F (1:113,6);
f06 (f06==1);
sum (ans);
p06= sum (ans);

f07 = F (1:113,7);
f07 (f07==1);
sum (ans);
p07= sum (ans);

f08 = F (1:113,8);
f08 (f08==1);
sum (ans);
p08= sum (ans);

f09 = F (1:113,9);
f09 (f09==1);
sum (ans);
p09= sum (ans);

f10 = F (1:113,10);
f10 (f10==1);
sum (ans);
p10= sum (ans);
```

```
f11 = F (1:113,11);
f11 (f11==1);
sum (ans);
p11= sum (ans);

f12 = F (1:113,12);
f12 (f12==1);
sum (ans);
p12= sum (ans);

f13 = F (1:113,13);
f13 (f13==1);
sum (ans);
p13= sum (ans);

f14 = F (1:113,14);
f14 (f14==1);
sum (ans);
p14= sum (ans);

f15 = F (1:113,15);
f15 (f15==1);
sum (ans);
p15= sum (ans);

f16 = F (1:113,16);
f16 (f16==1);
sum (ans);
p16= sum (ans);

f17 = F (1:113,17);
f17 (f17==1);
sum (ans);
p17= sum (ans);

f18 = F (1:113,18);
f18 (f18==1);
sum (ans);
p18= sum (ans);

f19 = F (1:113,19);
f19 (f19==1);
sum (ans);
p19= sum (ans);

f20 = F (1:113,20);
f20 (f20==1);
sum (ans);
p20= sum (ans);

f21 = F (1:113,21);
f21 (f21==1);
sum (ans);
p21= sum (ans);

f22 = F (1:113,22);
f22 (f22==1);
sum (ans);
p22= sum (ans);

f23= F (1:113,23);
f23 (f23==1);
sum (ans);
p23= sum (ans);

f24 = F (1:113,24);
f24 (f24==1);
sum (ans);
p24= sum (ans);

f25 = F (1:113,25);
f25 (f25==1);
sum (ans);
p25= sum (ans);

f26 = F (1:113,26);
f26 (f26==1);
sum (ans);
p26= sum (ans);

f27 = F (1:113,27);
f27 (f27==1);
sum (ans);
p27= sum (ans);

f28 = F (1:113,28);
f28 (f28==1);
sum (ans);
```

```
p28= sum (ans);

f29 = F (1:113,29);
f29 (f29==1);
sum (ans);
p29= sum (ans);
f30 = F (1:113,30);
f30 (f30==1);
sum (ans);
p30= sum (ans);

f31 = F (1:113,31);
f31 (f31==1);
sum (ans);
p31= sum (ans);

f32 = F (1:113,32);
f32 (f32==1);
sum (ans);
p32= sum (ans);

f33 = F (1:113,33);
f33 (f33==1);
sum (ans);
p33= sum (ans);

f34 = F (1:113,34);
f34 (f34==1);
sum (ans);
p34= sum (ans);

f35 = F (1:113,35);
f35 (f35==1);
sum (ans);
p35= sum (ans);

f36 = F (1:113,36);
f36 (f36==1);
sum (ans);
p36= sum (ans);

f37 = F (1:113,37);
f37 (f37==1);
sum (ans);
p37= sum (ans);

f38 = F (1:113,38);
f38 (f38==1);
sum (ans);
p38= sum (ans);
f39 = F (1:113,39);
f39 (f39==1);
sum (ans);
p39= sum (ans);

f40 = F (1:113,40);
f40 (f40==1);
sum (ans);
p40= sum (ans);

f41 = F (1:113,41);
f41 (f41==1);
sum (ans);
p41= sum (ans);

f42 = F (1:113,42);
f42 (f42==1);
sum (ans);
p42= sum (ans);

f43 = F (1:113,43);
f43 (f43==1);
sum (ans);
p43= sum (ans);

f44 = F (1:113,44);
f44 (f44==1);
sum (ans);
p44= sum (ans);

f45 = F (1:113,45);
f45 (f45==1);
sum (ans);
p45= sum (ans);

f46 = F (1:113,46);
f46 (f46==1);
sum (ans);
```

```
f64 = F (1:113,64);
f64 (f64==1);
sum (ans);
p64= sum (ans);

f65 = F (1:113,65);
f65 (f65==1);
sum (ans);
p65= sum (ans);

f66 = F (1:113,66);
f66 (f66==1);
sum (ans);
p66= sum (ans);

f67 = F (1:113,67);
f67 (f67==1);
sum (ans);
p67= sum (ans);

f68 = F (1:113,68);
f68 (f68==1);
sum (ans);
p68= sum (ans);

f69 = F (1:113,69);
f69 (f69==1);
sum (ans);
p69= sum (ans);

f70 = F (1:113,70);
f70 (f70==1);
sum (ans);
p70= sum (ans);

f71 = F (1:113,71);
f71 (f71==1);
sum (ans);
p71= sum (ans);

h   = [p01; p02; p03; p04; p05; p06; p07; p08; p09; p10; p11; p12; p13; p14; p15;
p16; p17; p18; p19; p20; p21; p22; p23; p24; p25; p26; p27; p28; p29; p30; p31; p32;
p33; p34; p35; p36; p37; p38; p39; p40; p41; p42; p43; p44; p45; p46; p47; p48; p49;
p50; p51; p52; p53; p54; p55; p56; p57; p58; p59; p60; p61; p62; p63; p64; p65; p66;
p67; p68; p69; p70; p71]


max (max(h));
hmax = ans
hn = (h/hmax)

dlmwrite('anton.txt',hn, ';')
```

- **Cropping karakter kedua**

```
load data3;
 I1 = handles.I1;
axes(handles.axes9);

L = bwlabel (I1);

max (max(L));
k = ans;

for x=1:k
    P = L(L==x);
   lab (x) = sum (P)/x;
```

```
end;
t = lab;
max (t);
find (t==ans);

l = ans;
la = (L==0);
lb = (L~=0);
F=L;
F(la)=0;
F(lb)=1;
F=F(15:128,81:151);

size(F)

save data3 F;
axes(handles.axes12);
imshow(F)
handles.F = F;
guidata(hObject, handles);

load data3;

f01 = F (1:113,1);
f01 (f01==1);
sum (ans);
p01= sum (ans);

f02 = F (1:113,2);
f02 (f02==1);
sum (ans);
p02= sum (ans);

f03 = F (1:113,3);
f03 (f03==1);
sum (ans);
p03= sum (ans);

f04 = F (1:113,4);
f04 (f04==1);
sum (ans);
p04= sum (ans);

f05 = F (1:113,5);
f05 (f05==1);
sum (ans);
p05= sum (ans);

f06 = F (1:113,6);
f06 (f06==1);
sum (ans);
p06= sum (ans);

f07 = F (1:113,7);
f07 (f07==1);
sum (ans);
p07= sum (ans);

f08 = F (1:113,8);
f08 (f08==1);
sum (ans);
p08= sum (ans);

f09 = F (1:113,9);
f09 (f09==1);
sum (ans);
p09= sum (ans);

f10 = F (1:113,10);
f10 (f10==1);
sum (ans);
p10= sum (ans);

f11 = F (1:113,11);
f11 (f11==1);
sum (ans);
p11= sum (ans);

f12 = F (1:113,12);
f12 (f12==1);
sum (ans);
p12= sum (ans);

f13 = F (1:113,13);
f13 (f13==1);
sum (ans);
p13= sum (ans);
```

```
f14 = F (1:113,14);
f14 (f14==1);
sum (ans);
p14= sum (ans);

f15 = F (1:113,15);
f15 (f15==1);
sum (ans);
p15= sum (ans);

f16 = F (1:113,16);
f16 (f16==1);
sum (ans);
p16= sum (ans);

f17 = F (1:113,17);
f17 (f17==1);
sum (ans);
p17= sum (ans);

f18 = F (1:113,18);
f18 (f18==1);
sum (ans);
p18= sum (ans);

f19 = F (1:113,19);
f19 (f19==1);
sum (ans);
p19= sum (ans);

f20 = F (1:113,20);
f20 (f20==1);
sum (ans);
p20= sum (ans);

f21 = F (1:113,21);
f21 (f21==1);
sum (ans);
p21= sum (ans);

f22 = F (1:113,22);
f22 (f22==1);
sum (ans);
p22= sum (ans);

f23= F (1:113,23);
f23 (f23==1);
sum (ans);
p23= sum (ans);

f24 = F (1:113,24);
f24 (f24==1);
sum (ans);
p24= sum (ans);

f25 = F (1:113,25);
f25 (f25==1);
sum (ans);
p25= sum (ans);

f26 = F (1:113,26);
f26 (f26==1);
sum (ans);
p26= sum (ans);

f27 = F (1:113,27);
f27 (f27==1);
sum (ans);
p27= sum (ans);

f28 = F (1:113,28);
f28 (f28==1);
sum (ans);
p28= sum (ans);

f29 = F (1:113,29);
f29 (f29==1);
sum (ans);
p29= sum (ans);

f30 = F (1:113,30);
f30 (f30==1);
sum (ans);
p30= sum (ans);

f31 = F (1:113,31);
f31 (f31==1);
```

```
sum (ans);
p31= sum (ans);

f32 = F (1:113,32);
f32 (f32==1);
sum (ans);
p32= sum (ans);

f33 = F (1:113,33);
f33 (f33==1);
sum (ans);
p33= sum (ans);

f34 = F (1:113,34);
f34 (f34==1);
sum (ans);
p34= sum (ans);

f35 = F (1:113,35);
f35 (f35==1);
sum (ans);
p35= sum (ans);

f36 = F (1:113,36);
f36 (f36==1);
sum (ans);
p36= sum (ans);

f37 = F (1:113,37);
f37 (f37==1);
sum (ans);
p37= sum (ans);

f38 = F (1:113,38);
f38 (f38==1);
sum (ans);
p38= sum (ans);

f39 = F (1:113,39);
f39 (f39==1);
sum (ans);
p39= sum (ans);

f40 = F (1:113,40);
f40 (f40==1);
sum (ans);
p40= sum (ans);

f41 = F (1:113,41);
f41 (f41==1);
sum (ans);
p41= sum (ans);

f42 = F (1:113,42);
f42 (f42==1);
sum (ans);
p42= sum (ans);

f43 = F (1:113,43);
f43 (f43==1);
sum (ans);
p43= sum (ans);

f44 = F (1:113,44);
f44 (f44==1);
sum (ans);
p44= sum (ans);

f45 = F (1:113,45);
f45 (f45==1);
sum (ans);
p45= sum (ans);

f46 = F (1:113,46);
f46 (f46==1);
sum (ans);
p46= sum (ans);

f47 = F (1:113,47);
f47 (f47==1);
sum (ans);
p47= sum (ans);

f48 = F (1:113,48);
f48 (f48==1);
sum (ans);
p48= sum (ans);
```

```
f49= F (1:113,49);
f49 (f49==1);
sum (ans);
p49= sum (ans);

f50 = F (1:113,50);
f50 (f50==1);
sum (ans);
p50= sum (ans);

f51 = F (1:113,51);
f51 (f51==1);
sum (ans);
p51= sum (ans);


f52 = F (1:113,52);
f52 (f52==1);
sum (ans);
p52= sum (ans);

f53 = F (1:113,53);
f53 (f53==1);
sum (ans);
p53= sum (ans);

f54 = F (1:113,54);
f54 (f54==1);
sum (ans);
p54= sum (ans);

f55 = F (1:113,55);
f55 (f55==1);
sum (ans);
p55= sum (ans);

f56 = F (1:113,56);
f56 (f56==1);
sum (ans);
p56= sum (ans);

f57 = F (1:113,57);
f57 (f57==1);
sum (ans);
p57= sum (ans);

f58 = F (1:113,58);
f58 (f58==1);
sum (ans);
p58= sum (ans);

f59 = F (1:113,59);
f59 (f59==1);
sum (ans);
p59= sum (ans);

f60 = F (1:113,60);
f60 (f60==1);
sum (ans);
p60= sum (ans);

f61 = F (1:113,61);
f61 (f61==1);
sum (ans);
p61= sum (ans);

f62 = F (1:113,62);
f62 (f62==1);
sum (ans);
p62= sum (ans);

f63 = F (1:113,63);
f63 (f63==1);
sum (ans);
p63= sum (ans);

f64 = F (1:113,64);
f64 (f64==1);
sum (ans);
p64= sum (ans);

f65 = F (1:113,65);
f65 (f65==1);
sum (ans);
p65= sum (ans);

f66 = F (1:113,66);
f66 (f66==1);
```

```
sum (ans);
p66= sum (ans);

f67 = F (1:113,67);
f67 (f67==1);
sum (ans);
p67= sum (ans);

f68 = F (1:113,68);
f68 (f68==1);
sum (ans);
p68= sum (ans);

f69 = F (1:113,69);
f69 (f69==1);
sum (ans);
p69= sum (ans);

f70 = F (1:113,70);
f70 (f70==1);
sum (ans);
p70= sum (ans);

f71 = F (1:113,71);
f71 (f71==1);
sum (ans);
p71= sum (ans);

h    = [p01; p02; p03; p04; p05; p06; p07; p08; p09; p10; p11; p12; p13; p14; p15;
p16; p17; p18; p19; p20; p21; p22; p23; p24; p25; p26; p27; p28; p29; p30; p31; p32;
p33; p34; p35; p36; p37; p38; p39; p40; p41; p42; p43; p44; p45; p46; p47; p48; p49;
p50; p51; p52; p53; p54; p55; p56; p57; p58; p59; p60; p61; p62; p63; p64; p65; p66;
p67; p68; p69; p70; p71]

max (max(h));
hmax = ans
hn = (h/hmax)

dlmwrite('anton2.txt',hn, ';')
```

- **Cropping Karakter Ketiga**

```
load data3;
 I1 = handles.I1;
axes(handles.axes9);

L = bwlabel (I1);

max (max(L));
k = ans;

for x=1:k
    P = L(L==x);
   lab (x) = sum (P)/x;
end;
t = lab;
max (t);
find (t==ans);

 l = ans;
la = (L==0);
lb = (L~=0);
F=L;
F(la)=0;
F(lb)=1;
F=F(15:128,174:244);

 size(F)

save data3 F;
axes(handles.axes13);
imshow(F)
handles.F = F;
guidata(hObject, handles);

load data3;
```

```
f01 = F (1:113,1);
f01 (f01==1);
sum (ans);
p01= sum (ans);

f02 = F (1:113,2);
f02 (f02==1);
sum (ans);
p02= sum (ans);

f03 = F (1:113,3);
f03 (f03==1);
sum (ans);
p03= sum (ans);

f04 = F (1:113,4);
f04 (f04==1);
sum (ans);
p04= sum (ans);

f05 = F (1:113,5);
f05 (f05==1);
sum (ans);
p05= sum (ans);

f06 = F (1:113,6);
f06 (f06==1);
sum (ans);
p06= sum (ans);

f07 = F (1:113,7);
f07 (f07==1);
sum (ans);
p07= sum (ans);

f08 = F (1:113,8);
f08 (f08==1);
sum (ans);
p08= sum (ans);

f09 = F (1:113,9);
f09 (f09==1);
sum (ans);
p09= sum (ans);

f10 = F (1:113,10);
f10 (f10==1);
sum (ans);
p10= sum (ans);

f11 = F (1:113,11);
f11 (f11==1);
sum (ans);
p11= sum (ans);

f12 = F (1:113,12);
f12 (f12==1);
sum (ans);
p12= sum (ans);

f13 = F (1:113,13);
f13 (f13==1);
sum (ans);
p13= sum (ans);

f14 = F (1:113,14);
f14 (f14==1);
sum (ans);
p14= sum (ans);

f15 = F (1:113,15);
f15 (f15==1);
sum (ans);
p15= sum (ans);

f16 = F (1:113,16);
f16 (f16==1);
sum (ans);
p16= sum (ans);

f17 = F (1:113,17);
f17 (f17==1);
sum (ans);
p17= sum (ans);

f18 = F (1:113,18);
f18 (f18==1);
```

```
sum (ans);
p18= sum (ans);

f19 = F (1:113,19);
f19 (f19==1);
sum (ans);
p19= sum (ans);

f20 = F (1:113,20);
f20 (f20==1);
sum (ans);
p20= sum (ans);

f21 = F (1:113,21);
f21 (f21==1);
sum (ans);
p21= sum (ans);

f22 = F (1:113,22);
f22 (f22==1);
sum (ans);
p22= sum (ans);

f23= F (1:113,23);
f23 (f23==1);
sum (ans);
p23= sum (ans);

f24 = F (1:113,24);
f24 (f24==1);
sum (ans);
p24= sum (ans);

f25 = F (1:113,25);
f25 (f25==1);
sum (ans);
p25= sum (ans);

f26 = F (1:113,26);
f26 (f26==1);
sum (ans);
p26= sum (ans);

f27 = F (1:113,27);
f27 (f27==1);
sum (ans);
p27= sum (ans);

f28 = F (1:113,28);
f28 (f28==1);
sum (ans);
p28= sum (ans);

f29 = F (1:113,29);
f29 (f29==1);
sum (ans);
p29= sum (ans);

f30 = F (1:113,30);
f30 (f30==1);
sum (ans);
p30= sum (ans);

f31 = F (1:113,31);
f31 (f31==1);
sum (ans);
p31= sum (ans);

f32 = F (1:113,32);
f32 (f32==1);
sum (ans);
p32= sum (ans);

f33 = F (1:113,33);
f33 (f33==1);
sum (ans);
p33= sum (ans);

f34 = F (1:113,34);
f34 (f34==1);
sum (ans);
p34= sum (ans);

f35 = F (1:113,35);
f35 (f35==1);
sum (ans);
p35= sum (ans);
```

```
f36 = F (1:113,36);
f36 (f36==1);
sum (ans);
p36= sum (ans);

f37 = F (1:113,37);
f37 (f37==1);
sum (ans);
p37= sum (ans);

f38 = F (1:113,38);
f38 (f38==1);
sum (ans);
p38= sum (ans);

f39 = F (1:113,39);
f39 (f39==1);
sum (ans);
p39= sum (ans);

f40 = F (1:113,40);
f40 (f40==1);
sum (ans);
p40= sum (ans);

f41 = F (1:113,41);
f41 (f41==1);
sum (ans);
p41= sum (ans);

f42 = F (1:113,42);
f42 (f42==1);
sum (ans);
p42= sum (ans);

f43 = F (1:113,43);
f43 (f43==1);
sum (ans);
p43= sum (ans);

f44 = F (1:113,44);
f44 (f44==1);
sum (ans);
p44= sum (ans);

f45 = F (1:113,45);
f45 (f45==1);
sum (ans);
p45= sum (ans);

f46 = F (1:113,46);
f46 (f46==1);
sum (ans);
p46= sum (ans);

f47 = F (1:113,47);
f47 (f47==1);
sum (ans);
p47= sum (ans);

f48 = F (1:113,48);
f48 (f48==1);
sum (ans);
p48= sum (ans);

f49= F (1:113,49);
f49 (f49==1);
sum (ans);
p49= sum (ans);

f50 = F (1:113,50);
f50 (f50==1);
sum (ans);
p50= sum (ans);

f51 = F (1:113,51);
f51 (f51==1);
sum (ans);
p51= sum (ans);


f52 = F (1:113,52);
f52 (f52==1);
sum (ans);
p52= sum (ans);

f53 = F (1:113,53);
f53 (f53==1);
```

```
sum (ans);
p53= sum (ans);

f54 = F (1:113,54);
f54 (f54==1);
sum (ans);
p54= sum (ans);

f55 = F (1:113,55);
f55 (f55==1);
sum (ans);
p55= sum (ans);

f56 = F (1:113,56);
f56 (f56==1);
sum (ans);
p56= sum (ans);

f57 = F (1:113,57);
f57 (f57==1);
sum (ans);
p57= sum (ans);

f58 = F (1:113,58);
f58 (f58==1);
sum (ans);
p58= sum (ans);

f59 = F (1:113,59);
f59 (f59==1);
sum (ans);
p59= sum (ans);

f60 = F (1:113,60);
f60 (f60==1);
sum (ans);
p60= sum (ans);

f61 = F (1:113,61);
f61 (f61==1);
sum (ans);
p61= sum (ans);

f62 = F (1:113,62);
f62 (f62==1);
sum (ans);
p62= sum (ans);

f63 = F (1:113,63);
f63 (f63==1);
sum (ans);
p63= sum (ans);

f64 = F (1:113,64);
f64 (f64==1);
sum (ans);
p64= sum (ans);

f65 = F (1:113,65);
f65 (f65==1);
sum (ans);
p65= sum (ans);

f66 = F (1:113,66);
f66 (f66==1);
sum (ans);
p66= sum (ans);

f67 = F (1:113,67);
f67 (f67==1);
sum (ans);
p67= sum (ans);

f68 = F (1:113,68);
f68 (f68==1);
sum (ans);
p68= sum (ans);

f69 = F (1:113,69);
f69 (f69==1);
sum (ans);
p69= sum (ans);

f70 = F (1:113,70);
f70 (f70==1);
sum (ans);
p70= sum (ans);
```

```
f71 = F (1:113,71);
f71 (f71==1);
sum (ans);
p71= sum (ans);
```

```
h   = [p01; p02; p03; p04; p05; p06; p07; p08; p09; p10; p11; p12; p13; p14; p15;
p16; p17; p18; p19; p20; p21; p22; p23; p24; p25; p26; p27; p28; p29; p30; p31; p32;
p33; p34; p35; p36; p37; p38; p39; p40; p41; p42; p43; p44; p45; p46; p47; p48; p49;
p50; p51; p52; p53; p54; p55; p56; p57; p58; p59; p60; p61; p62; p63; p64; p65; p66;
p67; p68; p69; p70; p71]

max (max(h));
hmax = ans
hn = (h/hmax)

dlmwrite('anton3.txt',hn, ';')
```

- **Cropping Karakter Keempat**

```
load data3;
 I1 = handles.I1;
axes(handles.axes9);

L = bwlabel (I1);

max (max(L));
k = ans;

for x=1:k
    P = L(L==x);
   lab (x) = sum (P)/x;

end;
t = lab;
max (t);
find (t==ans);

l = ans;
la = (L==0);
lb = (L~=0);
F=L;
F(la)=0;
F(lb)=1;
F=F(15:128,254:324);

 size(F)

save data3 F;
axes(handles.axes14);
imshow(F)
handles.F = F;
guidata(hObject, handles);

load data3;

f01 = F.(1:113,1);
f01 (f01==1);
sum (ans);
p01= sum (ans);

f02 = F (1:113,2);
f02 (f02==1);
sum (ans);
p02= sum (ans);
```

```
f03 = F (1:113,3);
f03 (f03==1);
sum (ans);
p03= sum (ans);

f04 = F (1:113,4);
f04 (f04==1);
sum (ans);
p04= sum (ans);

f05 = F (1:113,5);
f05 (f05==1);
sum (ans);
p05= sum (ans);

f06 = F (1:113,6);
f06 (f06==1);
sum (ans);
p06= sum (ans);

f07 = F (1:113,7);
f07 (f07==1);
sum (ans);
p07= sum (ans);

f08 = F (1:113,8);
f08 (f08==1);
sum (ans);
p08= sum (ans);

f09 = F (1:113,9);
f09 (f09==1);
sum (ans);
p09= sum (ans);

f10 = F (1:113,10);
f10 (f10==1);
sum (ans);
p10= sum (ans);

f11 = F (1:113,11);
f11 (f11==1);
sum (ans);
p11= sum (ans);

f12 = F (1:113,12);
f12 (f12==1);
sum (ans);
p12= sum (ans);

f13 = F (1:113,13);
f13 (f13==1);
sum (ans);
p13= sum (ans);

f14 = F (1:113,14);
f14 (f14==1);
sum (ans);
p14= sum (ans);

f15 = F (1:113,15);
f15 (f15==1);
sum (ans);
p15= sum (ans);

f16 = F (1:113,16);
f16 (f16==1);
sum (ans);
p16= sum (ans);

f17 = F (1:113,17);
f17 (f17==1);
sum (ans);
p17= sum (ans);

f18 = F (1:113,18);
f18 (f18==1);
sum (ans);
p18= sum (ans);

f19 = F (1:113,19);
f19 (f19==1);
sum (ans);
p19= sum (ans);

f20 = F (1:113,20);
f20 (f20==1);
```

```
sum (ans);
p20= sum (ans);

f21 = F (1:113,21);
f21 (f21==1);
sum (ans);
p21= sum (ans);

f22 = F (1:113,22);
f22 (f22==1);
sum (ans);
p22= sum (ans);

f23= F (1:113,23);
f23 (f23==1);
sum (ans);
p23= sum (ans);

f24 = F (1:113,24);
f24 (f24==1);
sum (ans);
p24= sum (ans);

f25 = F (1:113,25);
f25 (f25==1);
sum (ans);
p25= sum (ans);

f26 = F (1:113,26);
f26 (f26==1);
sum (ans);
p26= sum (ans);

f27 = F (1:113,27);
f27 (f27==1);
sum (ans);
p27= sum (ans);

f28 = F (1:113,28);
f28 (f28==1);
sum (ans);
p28= sum (ans);

f29 = F (1:113,29);
f29 (f29==1);
sum (ans);
p29= sum (ans);

f30 = F (1:113,30);
f30 (f30==1);
sum (ans);
p30= sum (ans);

f31 = F (1:113,31);
f31 (f31==1);
sum (ans);
p31= sum (ans);

f32 = F (1:113,32);
f32 (f32==1);
sum (ans);
p32= sum (ans);

f33 = F (1:113,33);
f33 (f33==1);
sum (ans);
p33= sum (ans);

f34 = F (1:113,34);
f34 (f34==1);
sum (ans);
p34= sum (ans);

f35 = F (1:113,35);
f35 (f35==1);
sum (ans);
p35= sum (ans);

f36 = F (1:113,36);
f36 (f36==1);
sum (ans);
p36= sum (ans);

f37 = F (1:113,37);
f37 (f37==1);
sum (ans);
p37= sum (ans);
```

```
f38 = F (1:113,38);
f38 (f38==1);
sum (ans);
p38= sum (ans);

f39 = F (1:113,39);
f39 (f39==1);
sum (ans);
p39= sum (ans);

f40 = F (1:113,40);
f40 (f40==1);
sum (ans);
p40= sum (ans);

f41 = F (1:113,41);
f41 (f41==1);
sum (ans);
p41= sum (ans);

f42 = F (1:113,42);
f42 (f42==1);
sum (ans);
p42= sum (ans);

f43 = F (1:113,43);
f43 (f43==1);
sum (ans);
p43= sum (ans);

f44 = F (1:113,44);
f44 (f44==1);
sum (ans);
p44= sum (ans);

f45 = F (1:113,45);
f45 (f45==1);
sum (ans);
p45= sum (ans);

f46 = F (1:113,46);
f46 (f46==1);
sum (ans);
p46= sum (ans);

f47 = F (1:113,47);
f47 (f47==1);
sum (ans);
p47= sum (ans);

f48 = F (1:113,48);
f48 (f48==1);
sum (ans);
p48= sum (ans);

f49= F (1:113,49);
f49 (f49==1);
sum (ans);
p49= sum (ans);

f50 = F (1:113,50);
f50 (f50==1);
sum (ans);
p50= sum (ans);

f51 = F (1:113,51);
f51 (f51==1);
sum (ans);
p51= sum (ans);

f52 = F (1:113,52);
f52 (f52==1);
sum (ans);
p52= sum (ans);

f53 = F (1:113,53);
f53 (f53==1);
sum (ans);
p53= sum (ans);

f54 = F (1:113,54);
f54 (f54==1);
sum (ans);
p54= sum (ans);

f55 = F (1:113,55);
f55 (f55==1);
```

```
sum (ans);
p55= sum (ans);

f56 = F (1:113,56);
f56 (f56==1);
sum (ans);
p56= sum (ans);

f57 = F (1:113,57);
f57 (f57==1);
sum (ans);
p57= sum (ans);

f58 = F (1:113,58);
f58 (f58==1);
sum (ans);
p58= sum (ans);

f59 = F (1:113,59);
f59 (f59==1);
sum (ans);
p59= sum (ans);

f60 = F (1:113,60);
f60 (f60==1);
sum (ans);
p60= sum (ans);

f61 = F (1:113,61);
f61 (f61==1);
sum (ans);
p61= sum (ans);

f62 = F (1:113,62);
f62 (f62==1);
sum (ans);
p62= sum (ans);

f63 = F (1:113,63);
f63 (f63==1);
sum (ans);
p63= sum (ans);

f64 = F (1:113,64);
f64 (f64==1);
sum (ans);
p64= sum (ans);

f65 = F (1:113,65);
f65 (f65==1);
sum (ans);
p65= sum (ans);

f66 = F (1:113,66);
f66 (f66==1);
sum (ans);
p66= sum (ans);

f67 = F (1:113,67);
f67 (f67==1);
sum (ans);
p67= sum (ans);

f68 = F (1:113,68);
f68 (f68==1);
sum (ans);
p68= sum (ans);

f69 = F (1:113,69);
f69 (f69==1);
sum (ans);
p69= sum (ans);

f70 = F (1:113,70);
f70 (f70==1);
sum (ans);
p70= sum (ans);

f71 = F (1:113,71);
f71 (f71==1);
sum (ans);
p71= sum (ans);


h   = [p01; p02; p03; p04; p05; p06;
p07; p08; p09; p10; p11; p12; p13;
p14; p15; p16; p17; p18; p19; p20;
p21; p22; p23; p24; p25; p26; p27;
```

```
p28; p29; p30; p31; p32; p33; p34;
p35; p36; p37; p38; p39; p40; p41;
p42; p43; p44; p45; p46; p47; p48;
p49; p50; p51; p52; p53; p54; p55;
p56; p57; p58; p59; p60; p61; p62;
p63; p64; p65; p66; p67; p68; p69;
p70; p71]
```

```
max (max(h));
hmax = ans
hn = (h/hmax)

dlmwrite('anton4.txt',hn, ';')
```

- **Cropping Karakter Kelima**

```
load data3;
 I1 = handles.I1;
axes(handles.axes9);

L = bwlabel (I1);

max (max(L));
k = ans;

for x=1:k
    P = L(L==x);
   lab (x) = sum (P)/x;

end;
t = lab;
max (t);
find (t==ans);


l = ans;
la = (L==0);
lb = (L~=0);
F=L;
F(la)=0;
F(lb)=1;
F=F(15:128,335:405);


 size(F)

save data3 F;
axes(handles.axes15);
imshow(F)
```

```
handles.F = F;
guidata(hObject, handles);

load data3;


f01 = F (1:113,1);
f01 (f01==1);
sum (ans);
p01= sum (ans);

f02 = F (1:113,2);
f02 (f02==1);
sum (ans);
p02= sum (ans);

f03 = F (1:113,3);
f03 (f03==1);
sum (ans);
p03= sum (ans);

f04 = F (1:113,4);
f04 (f04==1);
sum (ans);
p04= sum (ans);

f05 = F (1:113,5);
f05 (f05==1);
sum (ans);
p05= sum (ans);

f06 = F (1:113,6);
f06 (f06==1);
```

```
sum (ans);
p06= sum (ans);

f07 = F (1:113,7);
f07 (f07==1);
sum (ans);
p07= sum (ans);

f08 = F (1:113,8);
f08 (f08==1);
sum (ans);
p08= sum (ans);

f09 = F (1:113,9);
f09 (f09==1);
sum (ans);
p09= sum (ans);

f10 = F (1:113,10);
f10 (f10==1);
sum (ans);
p10= sum (ans);

f11 = F (1:113,11);
f11 (f11==1);
sum (ans);
p11= sum (ans);

f12 = F (1:113,12);
f12 (f12==1);
sum (ans);
p12= sum (ans);

f13 = F (1:113,13);
f13 (f13==1);
sum (ans);
p13= sum (ans);

f14 = F (1:113,14);
f14 (f14==1);
sum (ans);
p14= sum (ans);

f15 = F (1:113,15);
f15 (f15==1);
sum (ans);
p15= sum (ans);

f16 = F (1:113,16);
f16 (f16==1);
sum (ans);
p16= sum (ans);

f17 = F (1:113,17);
f17 (f17==1);
sum (ans);
p17= sum (ans);

f18 = F (1:113,18);
f18 (f18==1);
sum (ans);
p18= sum (ans);

f19 = F (1:113,19);
f19 (f19==1);
sum (ans);
p19= sum (ans);

f20 = F (1:113,20);
f20 (f20==1);
sum (ans);
p20= sum (ans);

f21 = F (1:113,21);
f21 (f21==1);
sum (ans);
p21= sum (ans);

f22 = F (1:113,22);
f22 (f22==1);
sum (ans);
p22= sum (ans);

f23= F (1:113,23);
f23 (f23==1);
sum (ans);
p23= sum (ans);
```

```
p41= sum (ans);

f42 = F (1:113,42);
f42 (f42==1);
sum (ans);
p42= sum (ans);

f43 = F (1:113,43);
f43 (f43==1);
sum (ans);
p43= sum (ans);

f44 = F (1:113,44);
f44 (f44==1);
sum (ans);
p44= sum (ans);

f45 = F (1:113,45);
f45 (f45==1);
sum (ans);
p45= sum (ans);

f46 = F (1:113,46);
f46 (f46==1);
sum (ans);
p46= sum (ans);

f47 = F (1:113,47);
f47 (f47==1);
sum (ans);
p47= sum (ans);

f48 = F (1:113,48);
f48 (f48==1);
sum (ans);
p48= sum (ans);

f49= F (1:113,49);
f49 (f49==1);
sum (ans);
p49= sum (ans);

f50 = F (1:113,50);
f50 (f50==1);
sum (ans);
p50= sum (ans);

f51 = F (1:113,51);
f51 (f51==1);
sum (ans);
p51= sum (ans);

f52 = F (1:113,52);
f52 (f52==1);
sum (ans);
p52= sum (ans);

f53 = F (1:113,53);
f53 (f53==1);
sum (ans);
p53= sum (ans);

f54 = F (1:113,54);
f54 (f54==1);
sum (ans);
p54= sum (ans);

f55 = F (1:113,55);
f55 (f55==1);
sum (ans);
p55= sum (ans);

f56 = F (1:113,56);
f56 (f56==1);
sum (ans);
p56= sum (ans);

f57 = F (1:113,57);
f57 (f57==1);
sum (ans);
p57= sum (ans);

f58 = F (1:113,58);
f58 (f58==1);
sum (ans);
p58= sum (ans);
```

```
f59 = F (1:113,59);
f59 (f59==1);
sum (ans);
p59= sum (ans);

f60 = F (1:113,60);
f60 (f60==1);
sum (ans);
p60= sum (ans);

f61 = F (1:113,61);
f61 (f61==1);
sum (ans);
p61= sum (ans);

f62 = F (1:113,62);
f62 (f62==1);
sum (ans);
p62= sum (ans);

f63 = F (1:113,63);
f63 (f63==1);
sum (ans);
p63= sum (ans);

f64 = F (1:113,64);
f64 (f64==1);
sum (ans);
p64= sum (ans);

f65 = F (1:113,65);
f65 (f65==1);
sum (ans);
p65= sum (ans);

f66 = F (1:113,66);
f66 (f66==1);
sum (ans);
p66= sum (ans);

f67 = F (1:113,67);
f67 (f67==1);
sum (ans);
p67= sum (ans);

f68 = F (1:113,68);
f68 (f68==1);
sum (ans);
p68= sum (ans);

f69 = F (1:113,69);
f69 (f69==1);
sum (ans);
p69= sum (ans);

f70 = F (1:113,70);
f70 (f70==1);
sum (ans);
p70= sum (ans);

f71 = F (1:113,71);
f71 (f71==1);
sum (ans);
p71= sum (ans);

h  = [p01; p02; p03; p04; p05; p06; p07; p08; p09; p10; p11; p12; p13; p14; p15;
p16; p17; p18; p19; p20; p21; p22; p23; p24; p25; p26; p27; p28; p29; p30; p31; p32;
p33; p34; p35; p36; p37; p38; p39; p40; p41; p42; p43; p44; p45; p46; p47; p48; p49;
p50; p51; p52; p53; p54; p55; p56; p57; p58; p59; p60; p61; p62; p63; p64; p65; p66;
p67; p68; p69; p70; p71]

max (max(h));
```

```
hmax = ans
hn = (h/hmax)

dlmwrite('anton5.txt',hn, ';')
```

- **Cropping Karakter Keenam**

```
load data3;
 I1 = handles.I1;
axes(handles.axes9);

L = bwlabel (I1);

max (max(L));
k = ans;

for x=1:k
    P = L(L==x);
   lab (x) = sum (P)/x;

end;
t = lab;
max (t);
find (t==ans);


l = ans;
la = (L==0);
lb = (L~=0);
F=L;
F(la)=0;
F(lb)=1;
F=F(15:128,416:486);

 size(F)

save data3 F;
axes(handles.axes16);
imshow(F)
handles.F = F;
guidata(hObject, handles);
```

```
load data3;


f01 = F (1:113,1);
f01 (f01==1);
sum (ans);
p01= sum (ans);

f02 = F (1:113,2);
f02 (f02==1);
sum (ans);
p02= sum (ans);

f03 = F (1:113,3);
f03 (f03==1);
sum (ans);
p03= sum (ans);

f04 = F (1:113,4);
f04 (f04==1);
sum (ans);
p04= sum (ans);

f05 = F (1:113,5);
f05 (f05==1);
sum (ans);
p05= sum (ans);

f06 = F (1:113,6);
f06 (f06==1);
sum (ans);
p06= sum (ans);

f07 = F (1:113,7);
f07 (f07==1);
```

```
sum (ans);
p07= sum (ans);

f08 = F (1:113,8);
f08 (f08==1);
sum (ans);
p08= sum (ans);

f09 = F (1:113,9);
f09 (f09==1);
sum (ans);
p09= sum (ans);

f10 = F (1:113,10);
f10 (f10==1);
sum (ans);
p10= sum (ans);

f11 = F (1:113,11);
f11 (f11==1);
sum (ans);
p11= sum (ans);

f12 = F (1:113,12);
f12 (f12==1);
sum (ans);
p12= sum (ans);

f13 = F (1:113,13);
f13 (f13==1);
sum (ans);
p13= sum (ans);

f14 = F (1:113,14);
f14 (f14==1);
sum (ans);
p14= sum (ans);

f15 = F (1:113,15);
f15 (f15==1);
sum (ans);
p15= sum (ans);

f16 = F (1:113,16);
f16 (f16==1);
sum (ans);
p16= sum (ans);

f17 = F (1:113,17);
f17 (f17==1);
sum (ans);
p17= sum (ans);

f18 = F (1:113,18);
f18 (f18==1);
sum (ans);
p18= sum (ans);

f19 = F (1:113,19);
f19 (f19==1);
sum (ans);
p19= sum (ans);

f20 = F (1:113,20);
f20 (f20==1);
sum (ans);
p20= sum (ans);

f21 = F (1:113,21);
f21 (f21==1);
sum (ans);
p21= sum (ans);

f22 = F (1:113,22);
f22 (f22==1);
sum (ans);
p22= sum (ans);

f23= F (1:113,23);
f23 (f23==1);
sum (ans);
p23= sum (ans);

f24 = F (1:113,24);
f24 (f24==1);
sum (ans);
p24= sum (ans);
```

```
f25 = F (1:113,25);
f25 (f25==1);
sum (ans);
p25= sum (ans);

f26 = F (1:113,26);
f26 (f26==1);
sum (ans);
p26= sum (ans);

f27 = F (1:113,27);
f27 (f27==1);
sum (ans);
p27= sum (ans);

f28 = F (1:113,28);
f28 (f28==1);
sum (ans);
p28= sum (ans);

f29 = F (1:113,29);
f29 (f29==1);
sum (ans);
p29= sum (ans);

f30 = F (1:113,30);
f30 (f30==1);
sum (ans);
p30= sum (ans);

f31 = F (1:113,31);
f31 (f31==1);
sum (ans);
p31= sum (ans);

f32 = F (1:113,32);
f32 (f32==1);
sum (ans);
p32= sum (ans);

f33 = F (1:113,33);
f33 (f33==1);
sum (ans);
p33= sum (ans);

f34 = F (1:113,34);
f34 (f34==1);
sum (ans);
p34= sum (ans);

f35 = F (1:113,35);
f35 (f35==1);
sum (ans);
p35= sum (ans);

f36 = F (1:113,36);
f36 (f36==1);
sum (ans);
p36= sum (ans);

f37 = F (1:113,37);
f37 (f37==1);
sum (ans);
p37= sum (ans);

f38 = F (1:113,38);
f38 (f38==1);
sum (ans);
p38= sum (ans);

f39 = F (1:113,39);
f39 (f39==1);
sum (ans);
p39= sum (ans);

f40 = F (1:113,40);
f40 (f40==1);
sum (ans);
p40= sum (ans);

f41 = F (1:113,41);
f41 (f41==1);
sum (ans);
p41= sum (ans);

f42 = F (1:113,42);
f42 (f42==1);
sum (ans);
```

```
p42= sum (ans);

f43 = F (1:113,43);
f43 (f43==1);
sum (ans);
p43= sum (ans);

f44 = F (1:113,44);
f44 (f44==1);
sum (ans);
p44= sum (ans);

f45 = F (1:113,45);
f45 (f45==1);
sum (ans);
p45= sum (ans);

f46 = F (1:113,46);
f46 (f46==1);
sum (ans);
p46= sum (ans);

f47 = F (1:113,47);
f47 (f47==1);
sum (ans);
p47= sum (ans);

f48 = F (1:113,48);
f48 (f48==1);
sum (ans);
p48= sum (ans);

f49= F (1:113,49);
f49 (f49==1);
sum (ans);
p49= sum (ans);

f50 = F (1:113,50);
f50 (f50==1);
sum (ans);
p50= sum (ans);

f51 = F (1:113,51);
f51 (f51==1);
sum (ans);
p51= sum (ans);


f52 = F (1:113,52);
f52 (f52==1);
sum (ans);
p52= sum (ans);

f53 = F (1:113,53);
f53 (f53==1);
sum (ans);
p53= sum (ans);

f54 = F (1:113,54);
f54 (f54==1);
sum (ans);
p54= sum (ans);

f55 = F (1:113,55);
f55 (f55==1);
sum (ans);
p55= sum (ans);

f56 = F (1:113,56);
f56 (f56==1);
sum (ans);
p56= sum (ans);

f57 = F (1:113,57);
f57 (f57==1);
sum (ans);
p57= sum (ans);

f58 = F (1:113,58);
f58 (f58==1);
sum (ans);
p58= sum (ans);

f59 = F (1:113,59);
f59 (f59==1);
sum (ans);
p59= sum (ans);
```

```
f60 = F (1:113,60);
f60 (f60==1);
sum (ans);
p60= sum (ans);

f61 = F (1:113,61);
f61 (f61==1);
sum (ans);
p61= sum (ans);

f62 = F (1:113,62);
f62 (f62==1);
sum (ans);
p62= sum (ans);

f63 = F (1:113,63);
f63 (f63==1);
sum (ans);
p63= sum (ans);

f64 = F (1:113,64);
f64 (f64==1);
sum (ans);
p64= sum (ans);

f65 = F (1:113,65);
f65 (f65==1);
sum (ans);
p65= sum (ans);

f66 = F (1:113,66);
f66 (f66==1);
sum (ans);
p66= sum (ans);

f67 = F (1:113,67);
f67 (f67==1);
sum (ans);
p67= sum (ans);

f68 = F (1:113,68);
f68 (f68==1);
sum (ans);
p68= sum (ans);

f69 = F (1:113,69);
f69 (f69==1);
sum (ans);
p69= sum (ans);

f70 = F (1:113,70);
f70 (f70==1);
sum (ans);
p70= sum (ans);

f71 = F (1:113,71);
f71 (f71==1);
sum (ans);
p71= sum (ans);


%Membentuk matrix baru dengan elemen2 yang berasal dari jumlah piksel aktif
%pada masing2 baris matrix di atas

h   = [p01; p02; p03; p04; p05; p06; p07; p08; p09; p10; p11; p12; p13; p14; p15;
p16; p17; p18; p19; p20; p21; p22; p23; p24; p25; p26; p27; p28; p29; p30; p31; p32;
p33; p34; p35; p36; p37; p38; p39; p40; p41; p42; p43; p44; p45; p46; p47; p48; p49;
p50; p51; p52; p53; p54; p55; p56; p57; p58; p59; p60; p61; p62; p63; p64; p65; p66;
p67; p68; p69; p70; p71]


max (max(h));
hmax = ans
```

```
hn = (h/hmax)

dlmwrite('anton6.txt',hn, ';')
```

- **Cropping Karakter Ketujuh**

```
load data3;
 I1 = handles.I1;
axes(handles.axes9);

L = bwlabel (I1);

max (max(L));
k = ans;

for x=1:k
    P = L(L==x);
   lab (x) = sum (P)/x;

end;
t = lab;
max (t);
find (t==ans);

l = ans;
la = (L==0);
lb = (L~=0);
F=L;
F(la)=0;
F(lb)=1;
F=F(15:128,500:570);

 size(F)

save data3 F;
axes(handles.axes17);
imshow(F)
handles.F = F;
guidata(hObject, handles);

load data3;

f01 = F (1:113,1);
f01 (f01==1);
sum (ans);
p01= sum (ans);

f02 = F (1:113,2);
f02 (f02==1);
sum (ans);
p02= sum (ans);

f03 = F (1:113,3);
f03 (f03==1);
sum (ans);
p03= sum (ans);

f04 = F (1:113,4);
f04 (f04==1);
sum (ans);
p04= sum (ans);

f05 = F (1:113,5);
f05 (f05==1);
sum (ans);
p05= sum (ans);

f06 = F (1:113,6);
f06 (f06==1);
sum (ans);
p06= sum (ans);

f07 = F (1:113,7);
f07 (f07==1);
sum (ans);
p07= sum (ans);

f08 = F (1:113,8);
f08 (f08==1);
sum (ans);
p08= sum (ans);
```

```
f09 = F (1:113,9);
f09 (f09==1);
sum (ans);
p09= sum (ans);

f10 = F (1:113,10);
f10 (f10==1);
sum (ans);
p10= sum (ans);

f11 = F (1:113,11);
f11 (f11==1);
sum (ans);
p11= sum (ans);

f12 = F (1:113,12);
f12 (f12==1);
sum (ans);
p12= sum (ans);

f13 = F (1:113,13);
f13 (f13==1);
sum (ans);
p13= sum (ans);

f14 = F (1:113,14);
f14 (f14==1);
sum (ans);
p14= sum (ans);

f15 = F (1:113,15);
f15 (f15==1);
sum (ans);
p15= sum (ans);

f16 = F (1:113,16);
f16 (f16==1);
sum (ans);
p16= sum (ans);

f17 = F (1:113,17);
f17 (f17==1);
sum (ans);
p17= sum (ans);

f18 = F (1:113,18);
f18 (f18==1);
sum (ans);
p18= sum (ans);

f19 = F (1:113,19);
f19 (f19==1);
sum (ans);
p19= sum (ans);

f20 = F (1:113,20);
f20 (f20==1);
sum (ans);
p20= sum (ans);

f21 = F (1:113,21);
f21 (f21==1);
sum (ans);
p21= sum (ans);

f22 = F (1:113,22);
f22 (f22==1);
sum (ans);
p22= sum (ans);

f23= F (1:113,23);
f23 (f23==1);
sum (ans);
p23= sum (ans);

f24 = F (1:113,24);
f24 (f24==1);
sum (ans);
p24= sum (ans);

f25 = F (1:113,25);
f25 (f25==1);
sum (ans);
p25= sum (ans);

f26 = F (1:113,26);
f26 (f26==1);
sum (ans);
```

```
p26= sum (ans);

f27 = F (1:113,27);
f27 (f27==1);
sum (ans);
p27= sum (ans);

f28 = F (1:113,28);
f28 (f28==1);
sum (ans);
p28= sum (ans);

f29 = F (1:113,29);
f29 (f29==1);
sum (ans);
p29= sum (ans);

f30 = F (1:113,30);
f30 (f30==1);
sum (ans);
p30= sum (ans);

f31 = F (1:113,31);
f31 (f31==1);
sum (ans);
p31= sum (ans);

f32 = F (1:113,32);
f32 (f32==1);
sum (ans);
p32= sum (ans);

f33 = F (1:113,33);
f33 (f33==1);
sum (ans);
p33= sum (ans);

f34 = F (1:113,34);
f34 (f34==1);
sum (ans);
p34= sum (ans);

f35 = F (1:113,35);
f35 (f35==1);
sum (ans);
p35= sum (ans);

f36 = F (1:113,36);
f36 (f36==1);
sum (ans);
p36= sum (ans);

f37 = F (1:113,37);
f37 (f37==1);
sum (ans);
p37= sum (ans);

f38 = F (1:113,38);
f38 (f38==1);
sum (ans);
p38= sum (ans);

f39 = F (1:113,39);
f39 (f39==1);
sum (ans);
p39= sum (ans);

f40 = F (1:113,40);
f40 (f40==1);
sum (ans);
p40= sum (ans);

f41 = F (1:113,41);
f41 (f41==1);
sum (ans);
p41= sum (ans);

f42 = F (1:113,42);
f42 (f42==1);
sum (ans);
p42= sum (ans);

f43 = F (1:113,43);
f43 (f43==1);
sum (ans);
p43= sum (ans);

f44 = F (1:113,44);
```

```
f44 (f44==1);
sum (ans);
p44= sum (ans);

f45 = F (1:113,45);
f45 (f45==1);
sum (ans);
p45= sum (ans);

f46 = F (1:113,46);
f46 (f46==1);
sum (ans);
p46= sum (ans);

f47 = F (1:113,47);
f47 (f47==1);
sum (ans);
p47= sum (ans);

f48 = F (1:113,48);
f48 (f48==1);
sum (ans);
p48= sum (ans);

f49= F (1:113,49);
f49 (f49==1);
sum (ans);
p49= sum (ans);

f50 = F (1:113,50);
f50 (f50==1);
sum (ans);
p50= sum (ans);

f51 = F (1:113,51);
f51 (f51==1);
sum (ans);
p51= sum (ans);


f52 = F (1:113,52);
f52 (f52==1);
sum (ans);
p52= sum (ans);

f53 = F (1:113,53);
f53 (f53==1);
sum (ans);
p53= sum (ans);

f54 = F (1:113,54);
f54 (f54==1);
sum (ans);
p54= sum (ans);

f55 = F (1:113,55);
f55 (f55==1);
sum (ans);
p55= sum (ans);

f56 = F (1:113,56);
f56 (f56==1);
sum (ans);
p56= sum (ans);

f57 = F (1:113,57);
f57 (f57==1);
sum (ans);
p57= sum (ans);

f58 = F (1:113,58);
f58 (f58==1);
sum (ans);
p58= sum (ans);

f59 = F (1:113,59);
f59 (f59==1);
sum (ans);
p59= sum (ans);

f60 = F (1:113,60);
f60 (f60==1);
sum (ans);
p60= sum (ans);

f61 = F (1:113,61);
f61 (f61==1);
sum (ans);
```

```
p61= sum (ans);

f62 = F (1:113,62);
f62 (f62==1);
sum (ans);
p62= sum (ans);

f63 = F (1:113,63);
f63 (f63==1);
sum (ans);
p63= sum (ans);

f64 = F (1:113,64);
f64 (f64==1);
sum (ans);
p64= sum (ans);

f65 = F (1:113,65);
f65 (f65==1);
sum (ans);
p65= sum (ans);

f66 = F (1:113,66);
f66 (f66==1);
sum (ans);
p66= sum (ans);

f67 = F (1:113,67);
f67 (f67==1);
sum (ans);
p67= sum (ans);

f68 = F (1:113,68);
f68 (f68==1);
sum (ans);
p68= sum (ans);

f69 = F (1:113,69);
f69 (f69==1);
sum (ans);
p69= sum (ans);

f70 = F (1:113,70);
f70 (f70==1);
sum (ans);
p70= sum (ans);

f71 = F (1:113,71);
f71 (f71==1);
sum (ans);
p71= sum (ans);


h   = [p01; p02; p03; p04; p05; p06; p07; p08; p09; p10; p11; p12; p13; p14; p15;
p16; p17; p18; p19; p20; p21; p22; p23; p24; p25; p26; p27; p28; p29; p30; p31; p32;
p33; p34; p35; p36; p37; p38; p39; p40; p41; p42; p43; p44; p45; p46; p47; p48; p49;
p50; p51; p52; p53; p54; p55; p56; p57; p58; p59; p60; p61; p62; p63; p64; p65; p66;
p67; p68; p69; p70; p71]


max (max(h));
hmax = ans
hn = (h/hmax)

dlmwrite('anton7.txt',hn, ';')
```

- **Cropping Karakter Kedelapan**

```
load data3;
 I1 = handles.I1;
axes(handles.axes9);

L = bwlabel (I1);

max (max(L));
k = ans;

for x=1:k
    P = L(L==x);
   lab (x) = sum (P)/x;

end;
t = lab;
max (t);
find (t==ans);


l = ans;
la = (L==0);
lb = (L~=0);
F=L;
F(la)=0;
F(lb)=1;
F=F(15:128,570:640);

 size(F)

save data3 F;
axes(handles.axes18);
imshow(F)
handles.F = F;
guidata(hObject, handles);

load data3;

f01 = F (1:113,1);
f01 (f01==1);
sum (ans);
p01= sum (ans);

f02 = F (1:113,2);
f02 (f02==1);
sum (ans);
p02= sum (ans);

f03 = F (1:113,3);
f03 (f03==1);
sum (ans);
p03= sum (ans);

f04 = F (1:113,4);
f04 (f04==1);
sum (ans);
p04= sum (ans);

f05 = F (1:113,5);
f05 (f05==1);
sum (ans);
p05= sum (ans);

f06 = F (1:113,6);
f06 (f06==1);
sum (ans);
p06= sum (ans);

f07 = F (1:113,7);
f07 (f07==1);
sum (ans);
p07= sum (ans);

f08 = F (1:113,8);
f08 (f08==1);
sum (ans);
p08= sum (ans);

f09 = F (1:113,9);
f09 (f09==1);
sum (ans);
p09= sum (ans);

f10 = F (1:113,10);
f10 (f10==1);
sum (ans);
p10= sum (ans);
```

```
f11 = F (1:113,11);
f11 (f11==1);
sum (ans);
p11= sum (ans);

f12 = F (1:113,12);
f12 (f12==1);
sum (ans);
p12= sum (ans);

f13 = F (1:113,13);
f13 (f13==1);
sum (ans);
p13= sum (ans);

f14 = F (1:113,14);
f14 (f14==1);
sum (ans);
p14= sum (ans);

f15 = F (1:113,15);
f15 (f15==1);
sum (ans);
p15= sum (ans);

f16 = F (1:113,16);
f16 (f16==1);
sum (ans);
p16= sum (ans);

f17 = F (1:113,17);
f17 (f17==1);
sum (ans);
p17= sum (ans);

f18 = F (1:113,18);
f18 (f18==1);
sum (ans);
p18= sum (ans);

f19 = F (1:113,19);
f19 (f19==1);
sum (ans);
p19= sum (ans);

f20 = F (1:113,20);
f20 (f20==1);
sum (ans);
p20= sum (ans);

f21 = F (1:113,21);
f21 (f21==1);
sum (ans);
p21= sum (ans);

f22 = F (1:113,22);
f22 (f22==1);
sum (ans);
p22= sum (ans);

f23= F (1:113,23);
f23 (f23==1);
sum (ans);
p23= sum (ans);

f24 = F (1:113,24);
f24 (f24==1);
sum (ans);
p24= sum (ans);

f25 = F (1:113,25);
f25 (f25==1);
sum (ans);
p25= sum (ans);

f26 = F (1:113,26);
f26 (f26==1);
sum (ans);
p26= sum (ans);

f27 = F (1:113,27);
f27 (f27==1);
sum (ans);
p27= sum (ans);

f28 = F (1:113,28);
f28 (f28==1);
```

```
sum (ans);
p28= sum (ans);

f29 = F (1:113,29);
f29 (f29==1);
sum (ans);
p29= sum (ans);

f30 = F (1:113,30);
f30 (f30==1);
sum (ans);
p30= sum (ans);

f31 = F (1:113,31);
f31 (f31==1);
sum (ans);
p31= sum (ans);

f32 = F (1:113,32);
f32 (f32==1);
sum (ans);
p32= sum (ans);

f33 = F (1:113,33);
f33 (f33==1);
sum (ans);
p33= sum (ans);

f34 = F (1:113,34);
f34 (f34==1);
sum (ans);
p34= sum (ans);

f35 = F (1:113,35);
f35 (f35==1);
sum (ans);
p35= sum (ans);

f36 = F (1:113,36);
f36 (f36==1);
sum (ans);
p36= sum (ans);

f37 = F (1:113,37);
f37 (f37==1);
sum (ans);
p37= sum (ans);

f38 = F (1:113,38);
f38 (f38==1);
sum (ans);
p38= sum (ans);

f39 = F (1:113,39);
f39 (f39==1);
sum (ans);
p39= sum (ans);

f40 = F (1:113,40);
f40 (f40==1);
sum (ans);
p40= sum (ans);

f41 = F (1:113,41);
f41 (f41==1);
sum (ans);
p41= sum (ans);

f42 = F (1:113,42);
f42 (f42==1);
sum (ans);
p42= sum (ans);

f43 = F (1:113,43);
f43 (f43==1);
sum (ans);
p43= sum (ans);

f44 = F (1:113,44);
f44 (f44==1);
sum (ans);
p44= sum (ans);

f45 = F (1:113,45);
f45 (f45==1);
sum (ans);
p45= sum (ans);
```

```
f46 = F (1:113,46);
f46 (f46==1);
sum (ans);
p46= sum (ans);
f47 = F (1:113,47);
f47 (f47==1);
sum (ans);
p47= sum (ans);

f48 = F (1:113,48);
f48 (f48==1);
sum (ans);
p48= sum (ans);

f49= F (1:113,49);
f49 (f49==1);
sum (ans);
p49= sum (ans);

f50 = F (1:113,50);
f50 (f50==1);
sum (ans);
p50= sum (ans);

f51 = F (1:113,51);
f51 (f51==1);
sum (ans);
p51= sum (ans);


f52 = F (1:113,52);
f52 (f52==1);
sum (ans);
p52= sum (ans);

f53 = F (1:113,53);
f53 (f53==1);
sum (ans);
p53= sum (ans);

f54 = F (1:113,54);
f54 (f54==1);
sum (ans);
p54= sum (ans);
```

```
f55 = F (1:113,55);
f55 (f55==1);
sum (ans);
p55= sum (ans);

f56 = F (1:113,56);
f56 (f56==1);
sum (ans);
p56= sum (ans);

f57 = F (1:113,57);
f57 (f57==1);
sum (ans);
p57= sum (ans);

f58 = F (1:113,58);
f58 (f58==1);
sum (ans);
p58= sum (ans);

f59 = F (1:113,59);
f59 (f59==1);
sum (ans);
p59= sum (ans);

f60 = F (1:113,60);
f60 (f60==1);
sum (ans);
p60= sum (ans);

f61 = F (1:113,61);
f61 (f61==1);
sum (ans);
p61= sum (ans);

f62 = F (1:113,62);
f62 (f62==1);
sum (ans);
p62= sum (ans);

f63 = F (1:113,63);
f63 (f63==1);
sum (ans);
```

```
p63= sum (ans);

f64 = F (1:113,64);
f64 (f64==1);
sum (ans);
p64= sum (ans);

f65 = F (1:113,65);
f65 (f65==1);
sum (ans);
p65= sum (ans);

f66 = F (1:113,66);
f66 (f66==1);
sum (ans);
p66= sum (ans);

f67 = F (1:113,67);
f67 (f67==1);
sum (ans);
p67= sum (ans);

f68 = F (1:113,68);
f68 (f68==1);
sum (ans);
p68= sum (ans);

f69 = F (1:113,69);
f69 (f69==1);
sum (ans);
p69= sum (ans);

f70 = F (1:113,70);
f70 (f70==1);
sum (ans);
p70= sum (ans);

f71 = F (1:113,71);
f71 (f71==1);
sum (ans);
p71= sum (ans);

h    = [p01; p02; p03; p04; p05; p06; p07; p08; p09; p10; p11; p12; p13; p14; p15;
p16; p17; p18; p19; p20; p21; p22; p23; p24; p25; p26; p27; p28; p29; p30; p31; p32;
p33; p34; p35; p36; p37; p38; p39; p40; p41; p42; p43; p44; p45; p46; p47; p48; p49;
p50; p51; p52; p53; p54; p55; p56; p57; p58; p59; p60; p61; p62; p63; p64; p65; p66;
p67; p68; p69; p70; p71]


max (max(h));
hmax = ans
hn = (h/hmax)

dlmwrite('anton8.txt',hn, ';')
```

***Listing Program Simulasi Hasil Pelatihan**

```
function pushbutton11_Callback(hObject, eventdata, handles)

nntwarn off
load trainning;
PA1 = load ('anton.txt');
P = PA1
A = sim(net,P)
```

```
round (A)
output = ans
if    (output == [0;0;0;0;0;0]);
      set(handles.edithasil, 'string','A');
elseif (output == [0;0;0;0;0;1]);
      set(handles.edithasil, 'string','B');
elseif (output == [0;0;0;0;1;0]);
      set(handles.edithasil, 'string','C');
elseif (output == [0;0;0;1;0;0]);
      set(handles.edithasil, 'string','D');
elseif (output == [0;0;1;0;0;0]);
      set(handles.edithasil, 'string','E');
elseif (output == [0;1;0;0;0;0]);
      set(handles.edithasil, 'string','F');
elseif (output == [1;0;0;0;0;0]);
      set(handles.edithasil, 'string','G');
elseif (output == [1;1;0;0;0;0]);
      set(handles.edithasil, 'string','H');
elseif (output == [1;0;1;0;0;0]);
      set(handles.edithasil, 'string','I');
elseif (output == [1;0;0;1;0;0]);
      set(handles.edithasil, 'string','J');
elseif (output == [1;0;0;0;1;0]);
      set(handles.edithasil, 'string','K');
elseif (output == [1;0;0;0;0;1]);
      set(handles.edithasil, 'string','L');
elseif (output == [0;0;0;0;1;1]);
      set(handles.edithasil, 'string','M');
elseif (output == [0;0;0;1;0;1]);
      set(handles.edithasil, 'string','N');
elseif (output == [0;0;1;0;0;1]);
      set(handles.edithasil, 'string','O');
elseif (output == [0;1;0;0;0;1]);
      set(handles.edithasil, 'string','P');
elseif (output == [0;0;0;1;1;0]);
      set(handles.edithasil, 'string','Q');
elseif (output == [0;0;1;0;1;0]);
      set(handles.edithasil, 'string','R');
elseif (output == [0;1;0;0;1;0]);
      set(handles.edithasil, 'string','S');
elseif (output == [0;1;0;1;0;0]);
      set(handles.edithasil, 'string','T');
elseif (output == [0;1;1;0;0;0]);
      set(handles.edithasil, 'string','U');
```

```
elseif (output == [1;1;1;0;0;0]);
      set(handles.edithasil, 'string','V');
elseif (output == [1;1;0;1;0;0]);
      set(handles.edithasil, 'string','W');
elseif (output == [1;1;0;0;1;0]);
      set(handles.edithasil, 'string','X');
elseif (output == [1;1;0;0;0;1]);
      set(handles.edithasil, 'string','Y');
elseif (output == [1;0;1;1;0;0]);
      set(handles.edithasil, 'string','Z');
elseif (output == [1;0;1;0;1;0]);
      set(handles.edithasil, 'string','0');
elseif (output == [1;0;1;0;0;1]);
      set(handles.edithasil, 'string','1');
elseif (output == [1;0;0;1;0;1]);
      set(handles.edithasil, 'string','2');
elseif (output == [1;0;0;0;1;1]);
      set(handles.edithasil, 'string','3');
elseif (output == [1;0;0;1;1;0]);
      set(handles.edithasil, 'string','4');
elseif (output == [0;1;1;1;0;0]);
      set(handles.edithasil, 'string','5');
elseif (output == [0;1;1;0;1;0]);
      set(handles.edithasil, 'string','6');
elseif (output == [0;1;1;0;0;1]);
      set(handles.edithasil, 'string','7');
elseif (output == [1;1;1;1;0;0]);
      set(handles.edithasil, 'string','8');
elseif (output == [1;1;1;0;0;1]);
      set(handles.edithasil, 'string','9');
else
      set(handles.edithasil, 'string','.');

end

nntwarn off
load trainning;
PA1 = load ('anton2.txt');
P = PA1
A = sim(net,P)
round (A)
output = ans
if    (output == [0;0;0;0;0;0]);
      set(handles.edit6, 'string','A');
```

```
elseif (output == [0;0;0;0;0;1]);
      set(handles.edit6, 'string','B');
elseif (output == [0;0;0;0;1;0]);
      set(handles.edit6, 'string','C');
elseif (output == [0;0;0;1;0;0]);
      set(handles.edit6, 'string','D');
elseif (output == [0;0;1;0;0;0]);
      set(handles.edit6, 'string','E');
elseif (output == [0;1;0;0;0;0]);
      set(handles.edit6, 'string','F');
elseif (output == [1;0;0;0;0;0]);
      set(handles.edit6, 'string','G');
elseif (output == [1;1;0;0;0;0]);
      set(handles.edit6, 'string','H');
elseif (output == [1;0;1;0;0;0]);
      set(handles.edit6, 'string','I');
elseif (output == [1;0;0;1;0;0]);
      set(handles.edit6, 'string','J');
elseif (output == [1;0;0;0;1;0]);
      set(handles.edit6, 'string','K');
elseif (output == [1;0;0;0;0;1]);
      set(handles.edit6, 'string','L');
elseif (output == [0;0;0;0;1;1]);
      set(handles.edit6, 'string','M');
elseif (output == [0;0;0;1;0;1]);
      set(handles.edit6, 'string','N');
elseif (output == [0;0;1;0;0;1]);
      set(handles.edit6, 'string','O');
elseif (output == [0;1;0;0;0;1]);
      set(handles.edit6, 'string','P');
elseif (output == [0;0;0;1;1;0]);
      set(handles.edit6, 'string','Q');
elseif (output == [0;0;1;0;1;0]);
      set(handles.edit6, 'string','R');
elseif (output == [0;1;0;0;1;0]);
      set(handles.edit6, 'string','S');
elseif (output == [0;1;0;1;0;0]);
      set(handles.edit6, 'string','T');
elseif (output == [0;1;1;0;0;0]);
      set(handles.edit6, 'string','U');
elseif (output == [1;1;1;0;0;0]);
      set(handles.edit6, 'string','V');
elseif (output == [1;1;0;1;0;0]);
      set(handles.edit6, 'string','W');
```

```
elseif (output == [1;1;0;0;1;0]);
      set(handles.edit6, 'string','X');
elseif (output == [1;1;0;0;0;1]);
      set(handles.edit6, 'string','Y');
elseif (output == [1;0;1;1;0;0]);
      set(handles.edit6, 'string','Z');
elseif (output == [1;0;1;0;1;0]);
      set(handles.edit6, 'string','0');
elseif (output == [1;0;1;0;0;1]);
      set(handles.edit6, 'string','1');
elseif (output == [1;0;0;1;0;1]);
      set(handles.edit6, 'string','2');
elseif (output == [1;0;0;0;1;1]);
      set(handles.edit6, 'string','3');
elseif (output == [1;0;0;1;1;0]);
      set(handles.edit6, 'string','4');
elseif (output == [0;1;1;1;0;0]);
      set(handles.edit6, 'string','5');
elseif (output == [0;1;1;0;1;0]);
      set(handles.edit6, 'string','6');
elseif (output == [0;1;1;0;0;1]);
      set(handles.edit6, 'string','7');
elseif (output == [1;1;1;1;0;0]);
      set(handles.edit6, 'string','8');
elseif (output == [1;1;1;0;0;1]);
      set(handles.edit6, 'string','9');
else
      set(handles.edit6, 'string','.');

end

nntwarn off
load trainning;
PA1 = load ('anton3.txt');
P = PA1
A = sim(net,P)
round (A)
output = ans
if    (output == [0;0;0;0;0;0]);
      set(handles.edit7, 'string','A');
elseif (output == [0;0;0;0;0;1]);
      set(handles.edit7, 'string','B');
elseif (output == [0;0;0;0;1;0]);
      set(handles.edit7, 'string','C');
```

```
elseif (output == [0;0;0;1;0;0]);
    set(handles.edit7, 'string','D');
elseif (output == [0;0;1;0;0;0]);
    set(handles.edit7, 'string','E');
elseif (output == [0;1;0;0;0;0]);
    set(handles.edit7, 'string','F');
elseif (output == [1;0;0;0;0;0]);
    set(handles.edit7, 'string','G');
elseif (output == [1;1;0;0;0;0]);
    set(handles.edit7, 'string','H');
elseif (output == [1;0;1;0;0;0]);
    set(handles.edit7, 'string','I');
elseif (output == [1;0;0;1;0;0]);
    set(handles.edit7, 'string','J');
elseif (output == [1;0;0;0;1;0]);
    set(handles.edit7, 'string','K');
elseif (output == [1;0;0;0;0;1]);
    set(handles.edit7, 'string','L');
elseif (output == [0;0;0;0;1;1]);
    set(handles.edit7, 'string','M');
elseif (output == [0;0;0;1;0;1]);
    set(handles.edit7, 'string','N');
elseif (output == [0;0;1;0;0;1]);
    set(handles.edit7, 'string','O');
elseif (output == [0;1;0;0;0;1]);
    set(handles.edit7, 'string','P');
elseif (output == [0;0;0;1;1;0]);
    set(handles.edit7, 'string','Q');
elseif (output == [0;0;1;0;1;0]);
    set(handles.edit7, 'string','R');
elseif (output == [0;1;0;0;1;0]);
    set(handles.edit7, 'string','S');
elseif (output == [0;1;0;1;0;0]);
    set(handles.edit7, 'string','T');
elseif (output == [0;1;1;0;0;0]);
    set(handles.edit7, 'string','U');
elseif (output == [1;1;1;0;0;0]);
    set(handles.edit7, 'string','V');
elseif (output == [1;1;0;1;0;0]);
    set(handles.edit7, 'string','W');
elseif (output == [1;1;0;0;1;0]);
    set(handles.edit7, 'string','X');
elseif (output == [1;1;0;0;0;1]);
    set(handles.edit7, 'string','Y');
```

```
elseif (output == [1;0;1;1;0;0]);
      set(handles.edit7, 'string','Z');
elseif (output == [1;0;1;0;1;0]);
      set(handles.edit7, 'string','0');
elseif (output == [1;0;1;0;0;1]);
      set(handles.edit7, 'string','1');
elseif (output == [1;0;0;1;0;1]);
      set(handles.edit7, 'string','2');
elseif (output == [1;0;0;0;1;1]);
      set(handles.edit7, 'string','3');
elseif (output == [1;0;0;1;1;0]);
      set(handles.edit7, 'string','4');
elseif (output == [0;1;1;1;0;0]);
      set(handles.edit7, 'string','5');
elseif (output == [0;1;1;0;1;0]);
      set(handles.edit7, 'string','6');
elseif (output == [0;1;1;0;0;1]);
      set(handles.edit7, 'string','7');
elseif (output == [1;1;1;1;0;0]);
      set(handles.edit7, 'string','8');
elseif (output == [1;1;1;0;0;1]);
      set(handles.edit7, 'string','9');
else
      set(handles.edit7, 'string','.');

end



nntwarn off
load trainning;
PA1 = load ('anton4.txt');
P = PA1
A = sim(net,P)
round (A)
output = ans
if    (output == [0;0;0;0;0;0]);
      set(handles.edit8, 'string','A');
elseif (output == [0;0;0;0;0;1]);
      set(handles.edit8, 'string','B');
elseif (output == [0;0;0;0;1;0]);
      set(handles.edit8, 'string','C');
elseif (output == [0;0;0;1;0;0]);
      set(handles.edit8, 'string','D');
```

```
elseif (output == [0;0;1;0;0;0]);
      set(handles.edit8, 'string','E');
elseif (output == [0;1;0;0;0;0]);
      set(handles.edit8, 'string','F');
elseif (output == [1;0;0;0;0;0]);
      set(handles.edit8, 'string','G');
elseif (output == [1;1;0;0;0;0]);
      set(handles.edit8, 'string','H');
elseif (output == [1;0;1;0;0;0]);
      set(handles.edit8, 'string','I');
elseif (output == [1;0;0;1;0;0]);
      set(handles.edit8, 'string','J');
elseif (output == [1;0;0;0;1;0]);
      set(handles.edit8, 'string','K');
elseif (output == [1;0;0;0;0;1]);
      set(handles.edit8, 'string','L');
elseif (output == [0;0;0;0;1;1]);
      set(handles.edit8, 'string','M');
elseif (output == [0;0;0;1;0;1]);
      set(handles.edit8, 'string','N');
elseif (output == [0;0;1;0;0;1]);
      set(handles.edit8, 'string','O');
elseif (output == [0;1;0;0;0;1]);
      set(handles.edit8, 'string','P');
elseif (output == [0;0;0;1;1;0]);
      set(handles.edit8, 'string','Q');
elseif (output == [0;0;1;0;1;0]);
      set(handles.edit8, 'string','R');
elseif (output == [0;1;0;0;1;0]);
      set(handles.edit8, 'string','S');
elseif (output == [0;1;0;1;0;0]);
      set(handles.edit8, 'string','T');
elseif (output == [0;1;1;0;0;0]);
      set(handles.edit8, 'string','U');
elseif (output == [1;1;1;0;0;0]);
      set(handles.edit8, 'string','V');
elseif (output == [1;1;0;1;0;0]);
      set(handles.edit8, 'string','W');
elseif (output == [1;1;0;0;1;0]);
      set(handles.edit8, 'string','X');
elseif (output == [1;1;0;0;0;1]);
      set(handles.edit8, 'string','Y');
elseif (output == [1;0;1;1;0;0]);
      set(handles.edit8, 'string','Z');
```

```
elseif (output == [1;0;1;0;1;0]);
      set(handles.edit8, 'string','0');
elseif (output == [1;0;1;0;0;1]);
      set(handles.edit8, 'string','1');
elseif (output == [1;0;0;1;0;1]);
      set(handles.edit8, 'string','2');
elseif (output == [1;0;0;0;1;1]);
      set(handles.edit8, 'string','3');
elseif (output == [1;0;0;1;1;0]);
      set(handles.edit8, 'string','4');
elseif (output == [0;1;1;1;0;0]);
      set(handles.edit8, 'string','5');
elseif (output == [0;1;1;0;1;0]);
      set(handles.edit8, 'string','6');
elseif (output == [0;1;1;0;0;1]);
      set(handles.edit8, 'string','7');
elseif (output == [1;1;1;1;0;0]);
      set(handles.edit8, 'string','8');
elseif (output == [1;1;1;0;0;1]);
      set(handles.edit8, 'string','9');
else
      set(handles.edit8, 'string','.');

end


nntwarn off
load trainning;
PA1 = load ('anton5.txt');
P = PA1
A = sim(net,P)
round (A)
output = ans
if    (output == [0;0;0;0;0;0]);
      set(handles.edit9, 'string','A');
elseif (output == [0;0;0;0;0;1]);
      set(handles.edit9, 'string','B');
elseif (output == [0;0;0;0;1;0]);
      set(handles.edit9, 'string','C');
elseif (output == [0;0;0;1;0;0]);
      set(handles.edit9, 'string','D');
elseif (output == [0;0;1;0;0;0]);
      set(handles.edit9, 'string','E');
elseif (output == [0;1;0;0;0;0]);
```

```
        set(handles.edit9, 'string','F');
elseif (output == [1;0;0;0;0;0]);
        set(handles.edit9, 'string','G');
elseif (output == [1;1;0;0;0;0]);
        set(handles.edit9, 'string','H');
elseif (output == [1;0;1;0;0;0]);
        set(handles.edit9, 'string','I');
elseif (output == [1;0;0;1;0;0]);
        set(handles.edit9, 'string','J');
elseif (output == [1;0;0;0;1;0]);
        set(handles.edit9, 'string','K');
elseif (output == [1;0;0;0;0;1]);
        set(handles.edit9, 'string','L');
elseif (output == [0;0;0;0;1;1]);
        set(handles.edit9, 'string','M');
elseif (output == [0;0;0;1;0;1]);
        set(handles.edit9, 'string','N');
elseif (output == [0;0;1;0;0;1]);
        set(handles.edit9, 'string','O');
elseif (output == [0;1;0;0;0;1]);
        set(handles.edit9, 'string','P');
elseif (output == [0;0;0;1;1;0]);
        set(handles.edit9, 'string','Q');
elseif (output == [0;0;1;0;1;0]);
        set(handles.edit9, 'string','R');
elseif (output == [0;1;0;0;1;0]);
        set(handles.edit9, 'string','S');
elseif (output == [0;1;0;1;0;0]);
        set(handles.edit9, 'string','T');
elseif (output == [0;1;1;0;0;0]);
        set(handles.edit9, 'string','U');
elseif (output == [1;1;1;0;0;0]);
        set(handles.edit9, 'string','V');
elseif (output == [1;1;0;1;0;0]);
        set(handles.edit9, 'string','W');
elseif (output == [1;1;0;0;1;0]);
        set(handles.edit9, 'string','X');
elseif (output == [1;1;0;0;0;1]);
        set(handles.edit9, 'string','Y');
elseif (output == [1;0;1;1;0;0]);
        set(handles.edit9, 'string','Z');
elseif (output == [1;0;1;0;1;0]);
        set(handles.edit9, 'string','0');
elseif (output == [1;0;1;0;0;1]);
```

```
        set(handles.edit9, 'string','1');
elseif (output == [1;0;0;1;0;1]);
        set(handles.edit9, 'string','2');
elseif (output == [1;0;0;0;1;1]);
        set(handles.edit9, 'string','3');
elseif (output == [1;0;0;1;1;0]);
        set(handles.edit9, 'string','4');
elseif (output == [0;1;1;1;0;0]);
        set(handles.edit9, 'string','5');
elseif (output == [0;1;1;0;1;0]);
        set(handles.edit9, 'string','6');
elseif (output == [0;1;1;0;0;1]);
        set(handles.edit9, 'string','7');
elseif (output == [1;1;1;1;0;0]);
        set(handles.edit9, 'string','8');
elseif (output == [1;1;1;0;0;1]);
        set(handles.edit9, 'string','9');
else
        set(handles.edit9, 'string','.');

end

nntwarn off
load trainning;
PA1 = load ('anton6.txt');
P = PA1
A = sim(net,P)
round (A)
output = ans
if    (output == [0;0;0;0;0;0]);
        set(handles.edit10, 'string','A');
elseif (output == [0;0;0;0;0;1]);
        set(handles.edit10, 'string','B');
elseif (output == [0;0;0;0;1;0]);
        set(handles.edit10, 'string','C');
elseif (output == [0;0;0;1;0;0]);
        set(handles.edit10, 'string','D');
elseif (output == [0;0;1;0;0;0]);
        set(handles.edit10, 'string','E');
elseif (output == [0;1;0;0;0;0]);
        set(handles.edit10, 'string','F');
elseif (output == [1;0;0;0;0;0]);
        set(handles.edit10, 'string','G');
elseif (output == [1;1;0;0;0;0]);
```

```
        set(handles.edit10, 'string','H');
elseif (output == [1;0;1;0;0;0]);
        set(handles.edit10, 'string','I');
elseif (output == [1;0;0;1;0;0]);
        set(handles.edit10, 'string','J');
elseif (output == [1;0;0;0;1;0]);
        set(handles.edit10, 'string','K');
elseif (output == [1;0;0;0;0;1]);
        set(handles.edit10, 'string','L');
elseif (output == [0;0;0;0;1;1]);
        set(handles.edit10, 'string','M');
elseif (output == [0;0;0;1;0;1]);
        set(handles.edit10, 'string','N');
elseif (output == [0;0;1;0;0;1]);
        set(handles.edit10, 'string','O');
elseif (output == [0;1;0;0;0;1]);
        set(handles.edit10, 'string','P');
elseif (output == [0;0;0;1;1;0]);
        set(handles.edit10, 'string','Q');
elseif (output == [0;0;1;0;1;0]);
        set(handles.edit10, 'string','R');
elseif (output == [0;1;0;0;1;0]);
        set(handles.edit10, 'string','S');
elseif (output == [0;1;0;1;0;0]);
        set(handles.edit10, 'string','T');
elseif (output == [0;1;1;0;0;0]);
        set(handles.edit10, 'string','U');
elseif (output == [1;1;1;0;0;0]);
        set(handles.edit10, 'string','V');
elseif (output == [1;1;0;1;0;0]);
        set(handles.edit10, 'string','W');
elseif (output == [1;1;0;0;1;0]);
        set(handles.edit10, 'string','X');
elseif (output == [1;1;0;0;0;1]);
        set(handles.edit10, 'string','Y');
elseif (output == [1;0;1;1;0;0]);
        set(handles.edit10, 'string','Z');
elseif (output == [1;0;1;0;1;0]);
        set(handles.edit10, 'string','0');
elseif (output == [1;0;1;0;0;1]);
        set(handles.edit10, 'string','1');
elseif (output == [1;0;0;1;0;1]);
        set(handles.edit10, 'string','2');
elseif (output == [1;0;0;0;1;1]);
```

```
        set(handles.edit10, 'string','3');
elseif (output == [1;0;0;1;1;0]);
        set(handles.edit10, 'string','4');
elseif (output == [0;1;1;1;0;0]);
        set(handles.edit10, 'string','5');
elseif (output == [0;1;1;0;1;0]);
        set(handles.edit10, 'string','6');
elseif (output == [0;1;1;0;0;1]);
        set(handles.edit10, 'string','7');
elseif (output == [1;1;1;1;0;0]);
        set(handles.edit10, 'string','8');
elseif (output == [1;1;1;0;0;1]);
        set(handles.edit10, 'string','9');
else
        set(handles.edit10, 'string','.');

end


nntwarn off
load trainning;
PA1 = load ('anton7.txt');
P = PA1
A = sim(net,P)
round (A)
output = ans
if    (output == [0;0;0;0;0;0]);
        set(handles.edit11, 'string','A');
elseif (output == [0;0;0;0;0;1]);
        set(handles.edit11, 'string','B');
elseif (output == [0;0;0;0;1;0]);
        set(handles.edit11, 'string','C');
elseif (output == [0;0;0;1;0;0]);
        set(handles.edit11, 'string','D');
elseif (output == [0;0;1;0;0;0]);
        set(handles.edit11, 'string','E');
elseif (output == [0;1;0;0;0;0]);
        set(handles.edit11, 'string','F');
elseif (output == [1;0;0;0;0;0]);
        set(handles.edit11, 'string','G');
elseif (output == [1;1;0;0;0;0]);
        set(handles.edit11, 'string','H');
elseif (output == [1;0;1;0;0;0]);
        set(handles.edit11, 'string','I');
```

```
elseif (output == [1;0;0;1;0;0]);
      set(handles.edit11, 'string','J');
elseif (output == [1;0;0;0;1;0]);
      set(handles.edit11, 'string','K');
elseif (output == [1;0;0;0;0;1]);
      set(handles.edit11, 'string','L');
elseif (output == [0;0;0;0;1;1]);
      set(handles.edit11, 'string','M');
elseif (output == [0;0;0;1;0;1]);
      set(handles.edit11, 'string','N');
elseif (output == [0;0;1;0;0;1]);
      set(handles.edit11, 'string','O');
elseif (output == [0;1;0;0;0;1]);
      set(handles.edit11, 'string','P');
elseif (output == [0;0;0;1;1;0]);
      set(handles.edit11, 'string','Q');
elseif (output == [0;0;1;0;1;0]);
      set(handles.edit11, 'string','R');
elseif (output == [0;1;0;0;1;0]);
      set(handles.edit11, 'string','S');
elseif (output == [0;1;0;1;0;0]);
      set(handles.edit11, 'string','T');
elseif (output == [0;1;1;0;0;0]);
      set(handles.edit11, 'string','U');
elseif (output == [1;1;1;0;0;0]);
      set(handles.edit11, 'string','V');
elseif (output == [1;1;0;1;0;0]);
      set(handles.edit11, 'string','W');
elseif (output == [1;1;0;0;1;0]);
      set(handles.edit11, 'string','X');
elseif (output == [1;1;0;0;0;1]);
      set(handles.edit11, 'string','Y');
elseif (output == [1;0;1;1;0;0]);
      set(handles.edit11, 'string','Z');
elseif (output == [1;0;1;0;1;0]);
      set(handles.edit11, 'string','0');
elseif (output == [1;0;1;0;0;1]);
      set(handles.edit11, 'string','1');
elseif (output == [1;0;0;1;0;1]);
      set(handles.edit11, 'string','2');
elseif (output == [1;0;0;0;1;1]);
      set(handles.edit11, 'string','3');
elseif (output == [1;0;0;1;1;0]);
      set(handles.edit11, 'string','4');
```

```
elseif (output == [0;1;1;1;0;0]);
      set(handles.edit11, 'string','5');
elseif (output == [0;1;1;0;1;0]);
      set(handles.edit11, 'string','6');
elseif (output == [0;1;1;0;0;1]);
      set(handles.edit11, 'string','7');
elseif (output == [1;1;1;1;0;0]);
      set(handles.edit11, 'string','8');
elseif (output == [1;1;1;0;0;1]);
      set(handles.edit11, 'string','9');
else
      set(handles.edit11, 'string','.');

end

nntwarn off
load trainning;
PA1 = load ('anton8.txt');
P = PA1
A = sim(net,P)
round (A)
output = ans
if    (output == [0;0;0;0;0;0]);
      set(handles.edit12, 'string','A');
elseif (output == [0;0;0;0;0;1]);
      set(handles.edit12, 'string','B');
elseif (output == [0;0;0;0;1;0]);
      set(handles.edit12, 'string','C');
elseif (output == [0;0;0;1;0;0]);
      set(handles.edit12, 'string','D');
elseif (output == [0;0;1;0;0;0]);
      set(handles.edit12, 'string','E');
elseif (output == [0;1;0;0;0;0]);
      set(handles.edit12, 'string','F');
elseif (output == [1;0;0;0;0;0]);
      set(handles.edit12, 'string','G');
elseif (output == [1;1;0;0;0;0]);
      set(handles.edit12, 'string','H');
elseif (output == [1;0;1;0;0;0]);
      set(handles.edit12, 'string','I');
elseif (output == [1;0;0;1;0;0]);
      set(handles.edit12, 'string','J');
elseif (output == [1;0;0;0;1;0]);
      set(handles.edit12, 'string','K');
```

```
elseif (output == [1;0;0;0;0;1]);
      set(handles.edit12, 'string','L');
elseif (output == [0;0;0;0;1;1]);
      set(handles.edit12, 'string','M');
elseif (output == [0;0;0;1;0;1]);
      set(handles.edit12, 'string','N');
elseif (output == [0;0;1;0;0;1]);
      set(handles.edit12, 'string','O');
elseif (output == [0;1;0;0;0;1]);
      set(handles.edit12, 'string','P');
elseif (output == [0;0;0;1;1;0]);
      set(handles.edit12, 'string','Q');
elseif (output == [0;0;1;0;1;0]);
      set(handles.edit12, 'string','R');
elseif (output == [0;1;0;0;1;0]);
      set(handles.edit12, 'string','S');
elseif (output == [0;1;0;1;0;0]);
      set(handles.edit12, 'string','T');
elseif (output == [0;1;1;0;0;0]);
      set(handles.edit12, 'string','U');
elseif (output == [1;1;1;0;0;0]);
      set(handles.edit12, 'string','V');
elseif (output == [1;1;0;1;0;0]);
      set(handles.edit12, 'string','W');
elseif (output == [1;1;0;0;1;0]);
      set(handles.edit12, 'string','X');
elseif (output == [1;1;0;0;0;1]);
      set(handles.edit12, 'string','Y');
elseif (output == [1;0;1;1;0;0]);
      set(handles.edit12, 'string','Z');
elseif (output == [1;0;1;0;1;0]);
      set(handles.edit12, 'string','0');
elseif (output == [1;0;1;0;0;1]);
      set(handles.edit12, 'string','1');
elseif (output == [1;0;0;1;0;1]);
      set(handles.edit12, 'string','2');
elseif (output == [1;0;0;0;1;1]);
      set(handles.edit12, 'string','3');
elseif (output == [1;0;0;1;1;0]);
      set(handles.edit12, 'string','4');
elseif (output == [0;1;1;1;0;0]);
      set(handles.edit12, 'string','5');
elseif (output == [0;1;1;0;1;0]);
      set(handles.edit12, 'string','6');
```

```
elseif (output == [0;1;1;0;0;1]);
      set(handles.edit12, 'string','7');
elseif (output == [1;1;1;1;0;0]);
      set(handles.edit12, 'string','8');
elseif (output == [1;1;1;0;0;1]);
      set(handles.edit12, 'string','9');
else
      set(handles.edit12, 'string','.');

end
```

1 sampai 113	39	39
1 sampai 113	40	40
1 sampai 113	41	41
1 sampai 113	42	42
1 sampai 113	43	43
1 sampai 113	44	44
1 sampai 113	45	45
1 sampai 113	46	46
1 sampai 113	47	47
1 sampai 113	48	48
1 sampai 113	49	49
1 sampai 113	50	50
1 sampai 113	51	51
1 sampai 113	52	52
1 sampai 113	53	53
1 sampai 113	54	54
1 sampai 113	55	55
1 sampai 113	56	56
1 sampai 113	57	57
1 sampai 113	58	58
1 sampai 113	59	59
1 sampai 113	60	60
1 sampai 113	61	61
1 sampai 113	62	62
1 sampai 113	63	63
1 sampai 113	64	64
1 sampai 113	65	65
1 sampai 113	66	66
1 sampai 113	67	67
1 sampai 113	68	68
1 sampai 113	69	69
1 sampai 113	70	70
1 sampai 113	71	71

UNIVERSITAS ANDALAS
UNTUK KEDJAJAAN BANGSA