

**PENGENALAN TANDA TANGAN MENGGUNAKAN  
ALGORITMA SINGLE LAYER PERCEPTRON**



**DIMAS AGUNG RIANSA  
5235127220**

**PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA DAN  
KOMPUTER  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI JAKARTA  
2017**

**HALAMAN PENGESAHAN**

**NAMA DOSEN**

**TANDA TANGAN**

**TANGGAL**

**Widodo, M.Kom**  
(Dosen Pembimbing I)



20-02-2017

**Bambang P. Adhi, S.Pd, M.Kom**  
(Dosen Pembimbing II)



17-02-2017

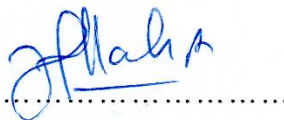
**PENGESAHAN PANITIA UJIAN SKRIPSI**

**NAMA DOSEN**

**TANDA TANGAN**

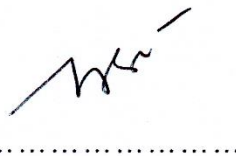
**TANGGAL**

**Hamidillah Ajie, MT**  
(Ketua Penguji)



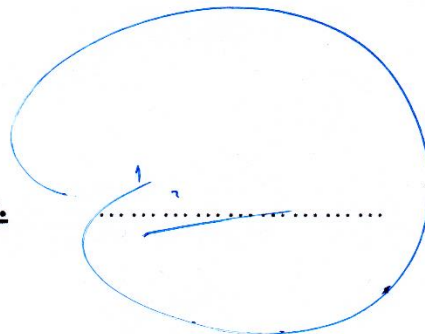
17-02-2017

**Bachren Zaini, M.Pd.**  
(Sekretaris)



16-02-2017

**M. Ficky Duskarnaen, M.Sc.**  
(Dosen Ahli)



17-02-2017

Tanggal lulus: .....

## HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Karya tulis skripsi saya yang berjudul Pengenalan Tanda Tangan Menggunakan Algoritma Single-Layer Perceptron adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri Jakarta maupun di perguruan tinggi lain.
2. Karya tulis yang berjudul Pengenalan Tanda Tangan Menggunakan Algoritma Single-Layer Perceptron adalah murni gagasan, rumusan, dan penelitian saya sendiri dengan arahan dosen pembimbing.
3. Dalam karya tulis, tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma yang berlaku di Universitas Negeri Jakarta.

Jakarta, Januari 2016

Yang membuat pernyataan

Dimas Agung Riansa

5235127220

## KATA PENGANTAR

Dengan memanjatkan puji dan syukur kehadiran Allah SWT yang telah memberikan rahmat, karunia, dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Pengenalan Tanda Tangan menggunakan Algoritma Single-Layer Perceptron” yang merupakan persyaratan untuk meraih gelar Sarjana Pendidikan pada Universitas Negeri Jakarta.

Pada kesempatan ini saya ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak Widodo, M.Kom. selaku pembimbing I dan Bapak Bambang Prasetya Adhi, S.Pd., M.Kom. selaku pembimbing II yang telah memberikan motivasi, arahan, dan kepercayaan kepada penulis dalam menyelesaikan skripsi ini.
2. Kedua Orang tua, kakak, dan adik yang tiada hentinya mendoakan dan memberikan semangat dalam menyelesaikan skripsi ini.
3. Teman-teman PTIK Non Reguler 2012 yang saling mendukung dan menyemangati agar dapat menyelesaikan skripsi.

Penulis menyadari skripsi ini masih banyak kekurangan baik dalam bentuk, isi maupun teknik penyajiannya. Oleh sebab itu kritikan yang bersifat membangun dari berbagai pihak penulis terima dengan tangan terbuka serta sangat diharapkan. Semoga kehadiran skripsi ini memenuhi sarannya.

Jakarta, Juni 2016

Dimas Agung Riansa  
5235127220

# PENGENALAN TANDA TANGAN MENGGUNAKAN ALGORITMA SINGLE LAYER PERCEPTRON

DIMAS AGUNG RIANSA

## ABSTRAK

Tanda tangan yang berasal dari Bahasa latin *Signare* adalah sebuah tulisan tangan yang digunakan untuk mengesahkan sebuah dokumen atau surat. Keterdapatannya tanda tangan dalam sebuah dokumen mengartikan bahwa pihak yang menandatangani mengetahui dan menyetujui seluruh isi dari dokumen. Hal ini menyebabkan tanda tangan dapat dipalsukan oleh pihak yang tidak bertanggung jawab. Tanda tangan dapat dikenali keasliannya secara manual atau dengan penggunaan komputer dengan menggunakan jaringan syaraf tiruan (JST). *Perceptron* adalah salah satu algoritma jaringan syaraf tiruan yang dapat digunakan untuk mengenali tanda tangan dengan akurat. Algoritma *Perceptron* merupakan sebuah algoritma yang digunakan untuk *supervised learning* (Pembelajaran Terarah) yang dapat mengklasifikasi sebuah input yang bersifat *linearly seperable* (dapat dipisahkan secara linier) kedalam kelas-kelas tertentu. Peneliti menggunakan tanda tangan dari 5 pejabat fakultas teknik universitas negeri Jakarta, terdapat 15 tanda tangan asli masing masing pejabat dan terdapat juga 15 tanda tangan palsu masing masing pejabat, secara keseluruhan terdapat 150 tanda tangan yang akan dijadikan sebagai data uji (*data test*) dan data latih (*data train*). *K fold-cross validation* digunakan untuk mendapatkan tingkat akurasi yang valid dari penggunaan algoritma perceptron, data dibagi kedalam 10 kelompok untuk membentuk *10 fold-cross validation* dari 150 data uji berjumlah 15 tanda tangan dan data latih berjumlah 135 tanda tangan. Hasil pengenalan tanda tangan menggunakan algoritma *perceptron* yang tingkat akurasinya diukur dengan menggunakan *k fold-cross validation*, memiliki rata-rata akurasi algoritma 78.667%

**Kata Kunci:** Klasifikasi, Tanda Tangan, Perceptron

# Signature Recognition using Single-Layer Perceptron

DIMAS AGUNG RIANSA

## ABSTRACT

Signature derived from the Latin language *Signare* is a handwriting that is used to validate a document or letter. Existence of a signature in a document meant that the signatories acknowledge and agree to the entire content of the document. This causes a signature can be forged by those who are not responsible. The signature can be recognized its authenticity either manually or with the use of computers by using artificial neural network (ANN). Perceptron is an artificial neural network algorithm which can be used to identify the signature accurately. Perceptron is an algorithm used for supervised learning (Focused Learning) to classify an input that is linearly separable (can be separated linearly) into certain classes. Researcher used the signature of five officials of the faculty of engineering state university of Jakarta, there are 15 original signature of each official and there are also 15 fake signature of each official, overall there were 150 signatures that will serve as *test* data and *train* data. K-fold cross validation is used to obtain a valid accuracy of the perceptron, the data is divided into 10 groups to form 10-fold cross validation, of the 150 signatures, 15 of it is *test* data while 135 of the rest is *train* data. The result of the signature recognition using perceptron algorithm's accuracy which measured using k-fold cross validation, had an average of 78.667%

**Keywords:** Classification, Signature, Perceptron

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PENGESAHAN</b> .....	ii
<b>LEMBAR PERNYATAAN</b> .....	iii
<b>KATA PENGANTAR</b> .....	iv
<b>ABSTRAK</b> .....	v
<b>ABSTRACT</b> .....	vi
<b>DAFTAR ISI</b> .....	vii
<b>DAFTAR TABEL</b> .....	ix
<b>DAFTAR GAMBAR</b> .....	x
<b>DAFTAR LAMPIRAN</b> .....	xi
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Identifikasi Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Rumusan Masalah .....	3
1.5 Tujuan Penelitian .....	3
1.6 Manfaat Penelitian .....	3
<b>BAB II TINJAUAN PUSTAKA</b>	
2.1. Kerangka Teori .....	4
2.1.1. Tanda Tangan .....	4
2.1.2. Jaringan Syaraf Tiruan (JST) .....	5
2.1.2.1. Arsitektur Jaringan Syaraf Tiruan .....	6
2.1.2.2. Jaringan Syaraf Tiruan Perceptron .....	6
2.1.2.3. Pelatihan pada Perceptron .....	8
2.1.3. Pengolahan Citra .....	9
2.1.3.1. Citra Digital .....	10
2.1.3.2. Grayscale .....	11
2.1.3.3. Penajaman Citra .....	12
2.1.4. Pengenalan Pola .....	12
2.1.5. Normalisasi Data .....	13
2.1.6. Format File BMP .....	14
2.1.7. Format File PBM .....	15
2.1.8. K fold-cross validation .....	16
2.2. Kerangka Berpikir .....	17
2.3. Penelitian yang Relevan .....	18

<b>BAB III METODOLOGI PENELITIAN</b>	
3.1. Tempat dan Waktu Penelitian .....	21
3.2. Alat dan Bahan Penelitian .....	21
3.3. Diagram Alir Penelitian .....	21
3.4. Teknik dan Prosedur Pengumpulan Data .....	24
3.4.1. Praproses Citra .....	25
3.4.2. Fase Pembelajaran dan Pengujian .....	27
3.5. Teknik Analisis Data .....	28
<b>BAB IV HASIL PENELITIAN</b>	
4.1. Deskripsi Hasil Penelitian .....	30
4.2. Analisis Data Penelitian .....	45
4.3. Pembahasan .....	52
4.4. Aplikasi Hasil Penelitian .....	53
<b>BAB V KESIMPULAN DAN SARAN</b>	
5.1. Kesimpulan .....	54
5.2. Saran .....	54
<b>DAFTAR PUSTAKA .....</b>	<b>56</b>
<b>LAMPIRAN .....</b>	<b>57</b>
<b>TENTANG PENULIS .....</b>	<b>64</b>



## DAFTAR TABEL

Tabel 2.1 Perhitungan pelatihan perceptron.....	9
Tabel 2.2 Pola dengan cirinya masing-masing .....	13
Tabel 4.1 Data <i>train</i> percobaan 1 .....	33
Tabel 4.2 Data <i>test</i> percobaan 1 .....	33
Tabel 4.3 Data <i>train</i> percobaan 2 .....	34
Tabel 4.4 Data <i>test</i> percobaan 2 .....	34
Tabel 4.5 Data <i>train</i> percobaan 3 .....	35
Tabel 4.6 Data <i>test</i> percobaan 3 .....	35
Tabel 4.7 Data <i>train</i> percobaan 4 .....	36
Tabel 4.8 Data <i>test</i> percobaan 4 .....	36
Tabel 4.9 Data <i>train</i> percobaan 5 .....	37
Tabel 4.10 Data <i>test</i> percobaan 5 .....	37
Tabel 4.11 Data <i>train</i> percobaan 6 .....	38
Tabel 4.12 Data <i>test</i> percobaan 6 .....	38
Tabel 4.13 Data <i>train</i> percobaan 7 .....	39
Tabel 4.14 Data <i>test</i> percobaan 7 .....	39
Tabel 4.15 Data <i>train</i> percobaan 8 .....	40
Tabel 4.16 Data <i>test</i> percobaan 8 .....	40
Tabel 4.17 Data <i>train</i> percobaan 9 .....	41
Tabel 4.18 Data <i>test</i> percobaan 9 .....	41
Tabel 4.19 Data <i>train</i> percobaan 10 .....	42
Tabel 4.20 Data <i>test</i> percobaan 10 .....	42
Tabel 4.21 Penjelasan hasil .....	45
Tabel 4.22 Data <i>test</i> percobaan 1 .....	45
Tabel 4.23 Hasil Percobaan 1 .....	46
Tabel 4.24 Hasil Percobaan 2 .....	47
Tabel 4.25 Hasil Percobaan 3 .....	47
Tabel 4.26 Hasil Percobaan 4 .....	48
Tabel 4.27 Hasil Percobaan 5 .....	48
Tabel 4.28 Hasil Percobaan 6 .....	49
Tabel 4.29 Hasil Percobaan 7 .....	49
Tabel 4.30 Hasil Percobaan 8 .....	50
Tabel 4.31 Hasil Percobaan 9 .....	51
Tabel 4.32 Hasil Percobaan 10 .....	51
Tabel 4.33 Hasil Tingkat Akurasi 10 fold-cross validation .....	52

## DAFTAR GAMBAR

Gambar 2.1 Struktur Neuron Jaringan Syaraf Tiruan .....	6
Gambar 2.2 Single Layer Perceptron .....	7
Gambar 2.3 Fungsi Pemisah Linear untuk <i>Perceptron</i> .....	8
Gambar 2.4 Perhitungan Pelatihan <i>Perceptron</i> .....	9
Gambar 2.5 (a) Citra Asli, (b) Citra Setelah Pengolahan Citra .....	10
Gambar 2.6 Koordinat Citra Digital .....	11
Gambar 2.7 Matriks Citra Digital .....	11
Gambar 2.8 Skala Keabuan Grayscale .....	12
Gambar 2.9 Citra Asli (Kiri) Citra Setelah Penajaman (Kanan) .....	12
Gambar 2.10 Kerangka Berfikir .....	17
Gambar 2.11 Mekanisme Sistem Biometrik .....	18
Gambar 3.1. Diagram Alir Penelitian .....	22
Gambar 3.2 Tanda tangan yang telah di scan .....	24
Gambar 3.3 Citra asli (kiri) dan sesudah grayscale (kanan) .....	25
Gambar 3.4 Citra sebelum ditajamkan (kiri) dan sesudah (kanan) .....	26
Gambar 3.5 Citra sebelum normalisasi (kiri) dan sesudah (kanan) .....	26
Gambar 3.6 Citra sebelum dipotong (kiri) dan sesudah (kanan) .....	26
Gambar 4.1. Tanda Tangan Sebelum Pra-proses .....	31
Gambar 4.2. Tanda Tangan Sesudah Pra-proses .....	31
Gambar 4.3 Matriks biasa .....	43
Gambar 4.4 Matriks 1-dimensi dari gambar 4.3 .....	43
Gambar 4.5 Proses <i>training</i> perceptron di matlab .....	44

## **DAFTAR LAMPIRAN**

Lampiran 1. Keseluruhan Tanda Tangan Sebelum Pra-proses .....	58
Lampiran 2. Keseluruhan Tanda Tangan Setelah Pra-proses .....	59
Lampiran 3. Barisan Kode Program Pra-proses dalam MATLAB .....	60
Lampiran 4. Barisan Kode Program Perceptron Dalam MATLAB .....	62

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Tanda tangan yang berasal dari Bahasa latin *Signare* adalah sebuah tulisan tangan yang digunakan untuk mengesahkan sebuah dokumen atau surat. Tanda tangan dapat berbentuk nama seseorang yang diberikan gaya penulisan tertentu, hal ini menyebabkan tanda tangan sebagai sebuah hal yang unik karena setiap orang memiliki perbedaan. Tanda tangan dapat digunakan sebagai identifikasi dari pemilik tanda tangan. Keterdapatannya dalam sebuah dokumen mengartikan bahwa pihak yang menandatangani mengetahui dan menyetujui seluruh isi dari dokumen. Hal ini menyebabkan tanda tangan dapat dipalsukan oleh pihak yang tidak bertanggung jawab.

Tanda tangan dapat dikenali secara manual akan tetapi hal ini membutuhkan keahlian khusus dan tingkat ketelitian yang tinggi. tetapi pada zaman modern ini pengenalan keaslian tanda tangan dapat dengan mudah dilakukan oleh sebuah komputer dengan menggunakan jaringan syaraf tiruan (JST), berbeda dengan pengenalan manual pengenalan dengan komputer tidak memerlukan keahlian khusus ataupun tingkat ketelitian yang tinggi dari pengguna. Pengenalan dengan komputer dapat dengan mudah menentukan keaslian sebuah tanda tangan, tetapi pengenalan dengan komputer sudah sewajarnya masih berada dibawah pengawasan ahli atau hanya sebagai alat bantu. Pengenalan tanda tangan menggunakan komputer membutuhkan algoritma tertentu.

*Perceptron* adalah salah satu algoritma jaringan syaraf tiruan (JST) yang dapat digunakan untuk mengenali tanda tangan dengan akurat. Algoritma *Perceptron* merupakan sebuah algoritma yang digunakan untuk *supervised learning* (Pembelajaran Terarah) yang dapat mengklasifikasi sebuah input yang bersifat *linearly seperable* (dapat dipisahkan secara linier) kedalam kelas-kelas tertentu. Untuk pengenalan tanda tangan pola akan terlebih melalui fase pembelajaran, pada fase tersebut *perceptron* mempelajari seluruh pola tanda tangan yang dimasukan, setelah fase pembelajaran terdapat fase pengenalan dimana algoritma *perceptron* akan mengenali pemilik tanda tangan yang dimasukan.

Hal tersebut yang membuat penulis tertarik untuk memahami algoritma *single-layer Perceptron* dan menggunakannya dalam pembuatan sistem pengenalan tanda tangan.

## **1.2 Identifikasi Masalah**

Berdasarkan latar belakang masalah yang telah dikemukakan diatas, maka dapat diidentifikasi masalah, yaitu:

1. Pemalsuan tanda-tangan sering dilakukan
2. Bagaimana cara mengenali tanda tangan seseorang
3. Bagaimana cara mengembangkan sebuah sistem pengenalan tanda tangan yang menggunakan algoritma *Single-Layer Perceptron*

## **1.3 Batasan Masalah**

Batasan permasalahan pada penulisan skripsi ini adalah:

1. Tanda tangan yang digunakan adalah tanda tangan pejabat-pejabat Fakultas Teknik Universitas Negeri Jakarta

2. Tanda tangan yang digunakan merupakan tulisan tangan dengan menggunakan media kertas dan pena yang kemudian akan di scan kedalam komputer.
3. Penggunaan *Single-Layer Perceptron* untuk mengidentifikasi tanda tangan dari hasil scan yang berbentuk .bmp

#### **1.4 Rumusan Masalah**

Berdasarkan latar belakang dan batasan masalah yang telah dijabarkan maka masalah yang dibahas dalam skripsi ini adalah Bagaimana pengembangan sistem pengenalan tanda tangan yang menggunakan algoritma *Single-layer Perceptron* dan seberapa tinggi akurasi algoritma tersebut?

#### **1.6 Tujuan Penelitian**

Tujuan dari penelitian ini adalah untuk Mengetahui kinerja algoritma *single-layer perceptron* dalam pengenalan tanda tangan

#### **1.7 Manfaat Penelitian**

Manfaat dari penelitian ini adalah

1. Menambah wawasan penulis tentang cara kerja jaringan syaraf tiruan khususnya *perceptron*.
2. Pembuatan sebuah sistem pengenalan tanda tangan yang layak digunakan

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Kerangka Teori**

##### **2.1.1 Tanda Tangan**

Tanda tangan adalah sebuah tulisan tangan yang diberi gaya tertentu dan dapat digunakan untuk mengesahkan sesuatu dokumen atau perjanjian. Tanda tangan merupakan salah satu bentuk identitas seseorang.

Contoh-contoh tanda tangan setiap orang umumnya identik namun tidak sama. Artinya tanda tangan seseorang sering berubah-ubah setiap waktu. Perubahan ini menyangkut posisi, ukuran maupun faktor tekanan tanda tangan. Pada kenyataannya, perubahan-perubahan tersebut dipengaruhi oleh waktu, umur, kebiasaan dan keadaan mental tertentu (Abbas,1994).

Menurut N. Siregar (2012) pada skripsinya menyatakan bahwa tanda tangan mempunyai karakteristik sebagai berikut:

1. Tanda tangan adalah bukti otentik
2. Tanda tangan jarang dilupakan.
3. Tanda tangan tidak dapat dipindah untuk digunakan ulang.
4. Tanda tangan tidak dapat disangkal.
5. Dokumen yang telah ditandatangani tidak dapat diubah.
6. Tanda tangan sulit dipalsukan.

### 2.1.2 Jaringan Syaraf Tiruan (JST)

Jaringan syaraf tiruan (JST) merupakan sistem yang mampu melakukan pengolahan informasi dengan memiliki kesamaan berpikir serta jaringan syaraf biologi manusia. JST tersebut akan dibentuk dengan menggunakan model matematika. JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi dengan asumsi (Siang, 2009):

1. *Neuron* merupakan elemen sederhana yang berguna melakukan pemrosesan informasi.
2. *Neuron-neuron* akan mendapatkan sinyal dari penghubung-penghubung.
3. Penghubung antar *neuron* memiliki bobot atau nilai yang mampu memperkuat atau memperlemah sinyal.
4. Setiap *neuron* menggunakan fungsi aktivasi (bukan fungsi *linear*) yang dihubungkan pada jumlah *input* yang diterima dan menghasilkan nilai *output* dan dibandingkan dengan suatu batas ambang.

JST ditentukan oleh 3 hal (Siang, 2009):

1. Pola hubungan antar *neuron* disebut arsitektur jaringan.
2. Metode untuk menentukan bobot penghubung disebut metode *training/learning* algoritma.
3. Fungsi aktivasi.

Gambar dari struktur *neuron* yang terdiri dari input, bobot, fungsi aktivasi, dan output dapat dilihat pada gambar 2.1





**Gambar 2.1 Struktur Neuron Jaringan Syaraf Tiruan**

### 2.1.2.1 Arsitektur Jaringan Syaraf Tiruan

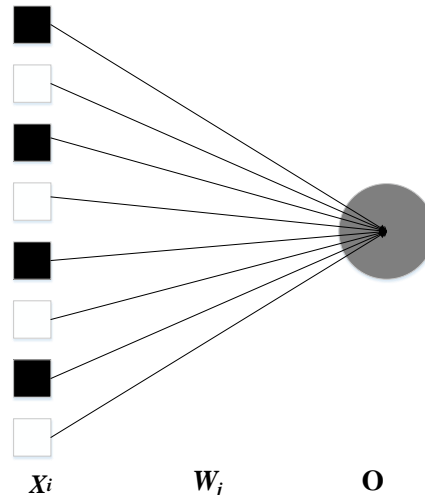
Arsitektur jaringan syaraf tiruan terdiri dari tiga lapisan penyusun (*layer*) (Siahaan, 2011), yaitu:

1. Lapisan masukan (*input layer*) Merupakan lapisan yang terdiri dari beberapa neuron yang akan menerima sinyal dari luar dan kemudian meneruskan ke neuron-neuron lain dalam jaringan.
2. Lapisan tersembunyi (*hidden layer*) Berfungsi meningkatkan kemampuan jaringan dalam memecahkan masalah. Konsekuensi dari adanya lapisan ini adalah pelatihan menjadi makin sulit atau lama.
3. Lapisan keluaran (*output layer*) Merupakan lapisan yang menyalurkan sinyal-sinyal keluaran hasil pemrosesan jaringan. Lapisan ini terdiri dari sejumlah neuron.

### 2.1.2.2 Jaringan Syaraf Tiruan *Perceptron*

Jaringan lapis tunggal *perceptron* (*Single Layer Perceptron*) terdiri dari beberapa unit pemroses (neuron) seperti gambar 2.2, yang terhubung dan mempunyai beberapa input serta satu atau lebih output. *Single Layer Perceptron* sering juga disebut dengan *Perceptron*. *Perceptron* menghitung jumlah nilai perkalian penimbang dan masukan dari parameter permasalahan yang kemudian

dibandingkan dengan nilai *threshold* bila nilai keluaran lebih besar dari *threshold* maka keluarannya adalah satu, sebaliknya adalah nol.



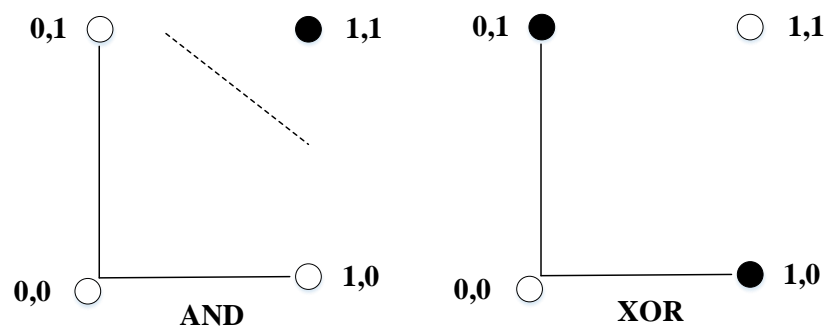
**Gambar 2.2 Single Layer Perceptron**

Pernyataan ini merupakan hasil proses pelatihan yang dengan kata lain adalah pernyataan ya dan tidak (satu dan nol). Secara matematis dapat dituliskan sebagai berikut: Jumlah perkalian penimbang dan parameter masukan adalah,

$$I = w_j x_i$$

Dimana  $x_i$ , adalah input dan  $w_j$  adalah penimbang (bobot). Bila  $I > T$  maka keluaran  $O = 1$  sebaliknya  $O = 0$  dimana  $T$  adalah *threshold*. Nilai awal penimbang adalah bilangan kecil antara nol sampai dengan satu yang dipanggil secara acak.

*Perceptron* dapat dipresentasikan dan hanya fungsi pemisah linear yang dapat dipisahkan dengan *perceptron* seperti pada gambar 2.3



**Gambar 2.3 Fungsi Pemisah Linear untuk *Perceptron***

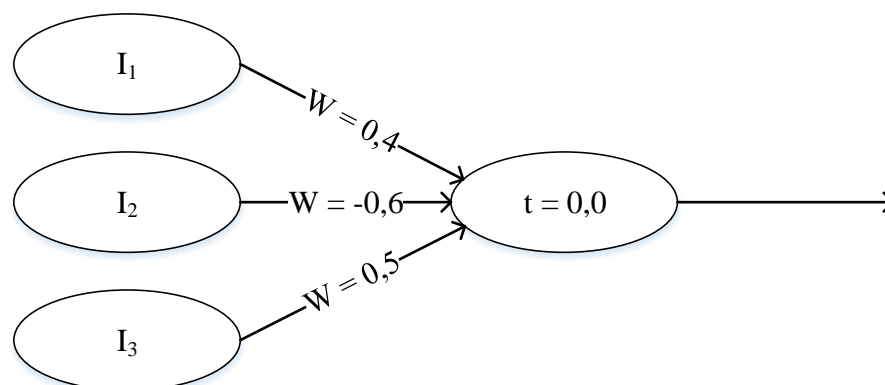
### 2.1.2.3 Pelatihan pada *Perceptron*

Pelatihan pada perceptron dilakukan dengan merubah nilai penimbangannya sehingga sesuai dengan kebutuhan yang dilakukan dengan membandingkan keluaran dari jaringan dengan targetnya :

$$w_{baru_{ji}} = w_{lama_{ji}} + a ( t_j - O_j ) x_i$$

$t_j$  adalah target, dan  $a$  adalah bilangan konstan bernilai kecil antara 0,1 sampai 0,9 yang disebut sebagai laju pelatihan (*learning rate*).

Contoh kasus operasi AND seperti pada gambar 2.3. pertama lakukan pengacakan pada bobot ( $W$ ) kemudian dilakukan perhitungan perkalian antara input ( $I$ ) dan bobot ( $W$ ) seperti gambar 2.4



**Gambar 2.4 Perhitungan Pelatihan *Perceptron***

**Tabel 2.1 Perhitungan pelatihan perceptron**

<b>I<sub>1</sub></b>	<b>I<sub>2</sub></b>	<b>I<sub>3</sub></b>	<b>Penjumlahan</b>	<b>Output</b>
0	0	1	$(0*0,4) + (0*-0,6) + (1*+0,5) = 0,5$	1
0	1	1	$(0*0,4) + (1*-0,6) + (1*+0,5) = -0,1$	0
1	0	1	$(1*0,4) + (0*-0,6) + (1*+0,5) = 0,9$	1
1	1	1	$(1*0,4) + (1*-0,6) + (1*+0,5) = 0,3$	1

Output diperoleh dengan melakukan fungsi aktivasi, dalam kasus AND dilakukan dengan unit step yaitu bila kurang dari nol output = 0, bila lebih output = 1. Error diperoleh apabila terjadi perbedaan antara target dengan output jaringan. Jika error tidak sama dengan nol maka bobot (W) diperbaiki dengan cara membandingkan bobot baru dengan bobot lama di tambah *learning rate*, selisih output dengan target, dan input :

$$w_{baru_{ji}} = w_{lama_{ji}} + a ( t_j - O_j ) x_i$$

### 2.1.3 Pengolahan Citra

Citra secara harafiah berarti gambar pada bidang dwimatra (dua dimensi) (Munir, 2002) sedangkan menurut KBBI citra adalah gambar; tiruan barang (orang, binatang, tumbuhan, dsb.) yang dibuat dengan coretan pensil dsb. pada kertas dsb; lukisan.

Meskipun sebuah citra kaya informasi, namun seringkali citra yang kita miliki mengalami penurunan mutu (degradasi), misalnya mengandung cacat atau derau (*noise*), warnanya terlalu kontras, kurang tajam, kabur (*blurring*), dan sebagainya. Tentu saja citra semacam ini menjadi lebih sulit diinterpretasi karena informasi yang disampaikan oleh citra tersebut menjadi berkurang.

Agar citra yang mengalami gangguan mudah diinterpretasi (baik oleh manusia maupun mesin), maka citra tersebut perlu dimanipulasi menjadi citra lain

yang kualitasnya lebih baik. Bidang studi yang menyangkut hal ini adalah **pengolahan citra** (*image processing*).

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan computer, menjadi citra yang kualitasnya lebih baik sebagai contoh citra

pada gambar 2.5 nampak gambar (a, kiri) berwarna, dengan operasi pengolahan citra dapat menghilangkan warna dari gambar (b, kanan)

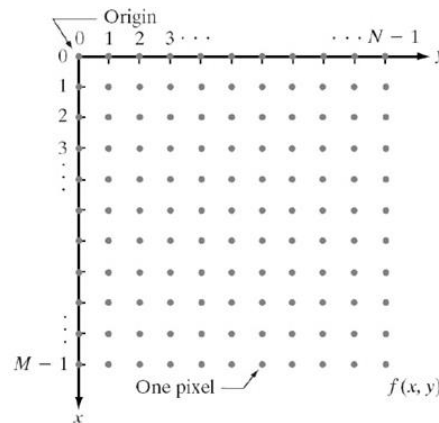


**Gambar 2.5 (a) Citra Asli, (b) Citra Setelah Pengolahan Citra**

Pengolahan citra bertujuan untuk memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini computer) (Rinaldi, 2002).

### 2.1.3.1 Citra Digital

Suatu citra dapat didefinisikan sebagai fungsi  $f(x,y)$  berukuran M baris dan N kolom dengan  $x$  dan  $y$  adalah koordinat spasial, dan amplitudo  $f$  dititik  $(x,y)$  dinamakan intensitas citra pada titik tersebut, Apabila nilai  $x,y$  dan nilai amplitudo  $f$  secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital. (Putra, 2010) gambar 2.6 menunjukkan posisi koordinat citra digital



**Gambar 2.6 Koordinat Citra Digital**

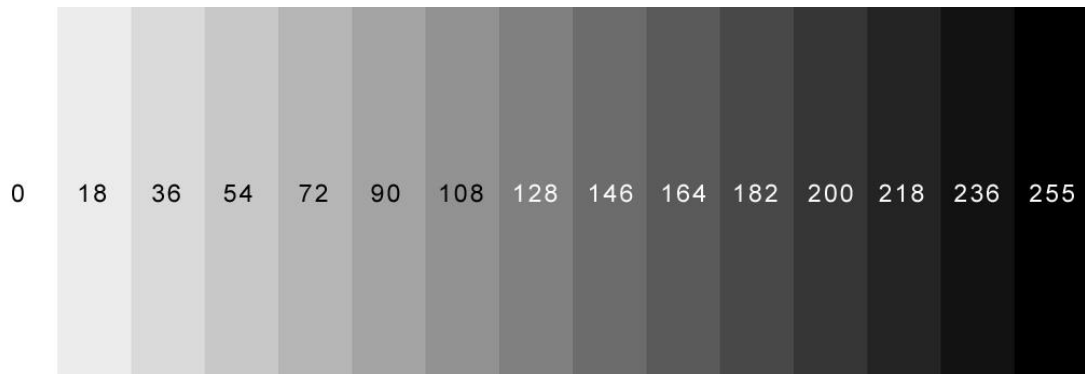
Jika ditulis dalam bentuk matriks

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

**Gambar 2.7 Matriks Citra Digital**

### 2.1.3.2 *Grayscale*

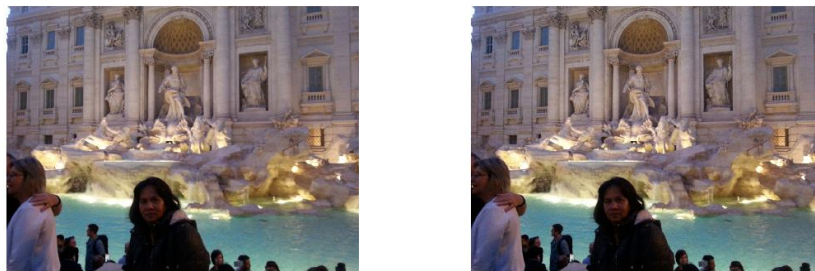
Citra *Grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixelnya, dengan kata lain nilai bagian RED = GREEN = BLUE. (Putra ,2010) Nilai tersebut digunakan untuk menentukan intensitas keabuan dari hitam hingga mendekati putih. Citra *grayscale* memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan).



**Gambar 2.8 Skala Keabuan *Grayscale***

### 2.1.3.3 Penajaman Citra

Penajaman Citra (*Image Sharpening*) bertujuan untuk memperjelas tepi pada objek di dalam citra. Operasi penajaman dilakukan dengan melewati citra pada penapis lolos-tinggi (*high-pass filter*). Penapis lolos-tinggi akan meloloskan (atau memperkuat) komponen yang berfrekuensi tinggi (misalnya tepi atau pinggiran objek) dan akan menurunkan komponen berfrekuensi rendah. Akibatnya, pinggiran objek terlihat lebih tajam dibandingkan sekitarnya.



**Gambar 2.9 Citra Asli (Kiri) Citra Setelah Penajaman (Kanan)**

### 2.1.4 Pengenalan Pola

Pola adalah entitas yang terdefinisi dan dapat diidentifikasi melalui ciri-cirinya (*features*) (Hendradjaya, 1995, diacu dalam Rinaldi, 2002). Ciri-ciri tersebut

digunakan untuk membedakan suatu pola dengan pola lainnya. Ciri yang bagus adalah ciri yang memiliki daya pembeda yang tinggi, sehingga pengelompokan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan keakuratan yang tinggi. Sebagai contoh,

**Tabel 2.2 Pola dengan cirinya masing-masing**

<b>Pola</b>	<b>Ciri</b>
huruf	tinggi, tebal, titik sudut, lengkungan garis, dll
suara	amplitudo, frekuensi, nada, intonasi, warna, dll
tanda tangan	panjang, kerumitan, tekanan, dll
sidik jari	lengkungan, jumlah garis, dll

Ciri pada suatu pola diperoleh dari hasil pengukuran terhadap objek uji. Khusus pada pola yang terdapat di dalam citra, ciri-ciri yang dapat diperoleh berasal dari informasi:

1. Spasial : Intensitas pixel, Histogram
2. Tepi : Arah, Kekuatan
3. Kontur : Garis, Elips, Lingkaran
4. Wilayah/Bentuk : Keliling, Luas, Pusat Massa
5. Hasil Transformasi Fourier : Frekuensi

### **2.1.5 Normalisasi Data**

Normalisasi (*Normalization*) adalah proses transformasi dimana sebuah atribut numerik diskalakan dalam range yang lebih kecil seperti -1.0 sampai 1.0 atau 0.0



sampai 1.0 (Hartarto, dkk., 2011). Ada beberapa metode yang diterapkan untuk normalisasi data diantaranya:

- *Min-max Normalization*
- *Z-Score Normalization*
- *Normalization by Decimal Scaling*

### **2.1.6 Format File BMP**

Format file bmp merupakan sebuah format file citra yang umum digunakan oleh sistem operasi windows untuk menyimpan citra berbasis *bitmap*, citra berbasis *bitmap* berarti citra dibentuk dengan *raster/pixel/dot*/titik koordinat (Sunarto, 2008) dimana semakin banyak jumlah titik yang membentuk gambar maka semakin bagus resolusi gambar tersebut. File bitmap windows disimpan dalam format *device-independent bitmap* (DIB) yang memungkinkan windows untuk menampilkan gambar *bitmap* pada beragam jenis layar. Istilah “*device-independent*” berarti bahwa *bitmap* menentukan warna *pixel* dalam sebuah bentuk independen dari yang digunakan oleh layar untuk mewakili warna (A. P. Godse, D. A. Godse, 2007). Gambar dalam format bmp biasanya disimpan dengan kedalaman warna 2 (1-bit), 16 (4-bit), 256 (8-bit), 65526(16-bit) atau 16.7 juta (24-bit).

Format file bmp terdiri atas *bitmap-file header*, *bitmap information header*, *optional color table* dan *pixel data area*.

- Sebuah *bitmap-file header* berisikan informasi tentang tipe, ukuran dan layout dari file *bitmap*.
- *Bitmap-information header* merincikan dimensi, tipe kompresi, dan format warna untuk *bitmap*.

- *Optional Color Table* berisi elemen sebanyak jumlah warna dalam *bitmap*, table ini bersifat opsional, *optional color table* tidak terdapat untuk *bitmap* dengan 24-bit karena setiap pixel sudah diwakili oleh nilai 24-bit warna RGB di *pixel data area*.
- *Pixel data area* merupakan tempat data pixel berada (M. Reddy)

### 2.1.7 Format File PBM

*Portable bitmap (PBM) format* merupakan sebuah common denominator terendah format file monokrom. Pada awalnya dirancang untuk mewajarkan mengirimkan *bitmap* menggunakan e-mail antara berbagai jenis mesin menggunakan jaringan yang umum digunakan pada saat itu (J. Poskanzer, 1991). Sekarang *pbm* berfungsi sebagai bahasa umum keluarga besar filter konversi *bitmap*.

Format file *pbm* terbagi atas:

- sebuah kode untuk mengidentifikasi tipe file, kode untuk tipe file *.pbm* adalah "P4".
- Lebar gambar dalam ukuran pixel, diformat sebagai karakter ASCII dalam desimal.
- Tinggi gambar dalam ukuran pixel, diformat dalam ASCII desimal.
- Sebuah raster gambar dalam bit. Setiap bit mewakili pixel, putih: 1 dan hitam: 0

Format file *pbm* terbagi menjadi 2 versi yaitu *raw-pbm* dan *plain-pbm*. *Plain-pbm* merupakan versi pertama dari format file *pbm* tetapi karena *plain-pbm* menggunakan sumber daya yang sangat berlebihan penggunaan *plain-pbm*

dihentikan. Raw-pbm merupakan versi file pbm yang lebih efisien dan umum digunakan pada saat ini.

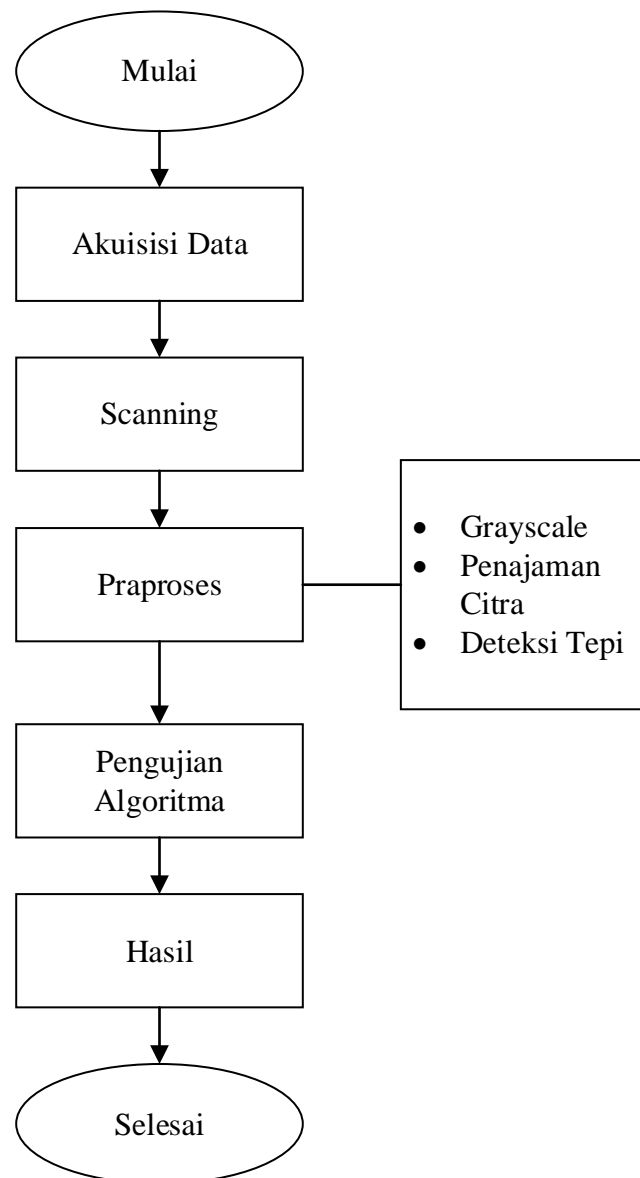
### **2.1.8 K fold Cross-validation**

*Cross validation* atau yang terkadang disebut *rotation estimation* adalah sebuah teknik validasi untuk menilai bagaimana hasil analisis statistik akan menyamakan sebuah kelompok data independen. *Cross validation* sering digunakan dalam percobaan yang bertujuan untuk memprediksi. Dalam *cross validation* kelompok data  $K$  akan dipisah secara acak kedalam kelompok yang jumlahnya sama ( $K_1, K_2, \dots, K_k$ ) lalu setiap kelompok data masing masing akan dijadikan data *train* dan data *test* jumlah percobaan yang dilakukan sama dengan jumlah kelompok yang ada. Jika hanya terdapat 2 kelompok atau 2 *fold cross validation* maka apabila  $K_1$  dijadikan data *test* maka  $K_2$  merupakan data *train* untuk percobaan kedua  $K_1$  merupakan data *train* dan  $K_2$  merupakan data *test*. Nilai akurasi dari  $k$  fold *cross validation* merupakan nilai rerata dari hasil yang benar setiap percobaan.

## **2.2 Kerangka Berpikir**

Seiring dengan kemajuan jaman pemalsuan tanda tangan pada dokumen semakin sering dilakukan, karena itu dibutuhkan sebuah sistem yang dapat mengenali tanda tangan. Proses sistem pengenalan tanda tangan ini dimulai dari mendapatkan tanda tangan fisik yang kemudian akan dilakukan scanning untuk memperoleh citra digital dari tanda tangan tersebut setelah citra digital didapatkan selanjutnya dilakukan praproses yang terdiri dari *Grayscale*, Penajaman Citra, dan Deteksi Tepi. Setelah ini akan dilakukan pengenalan pola agar sistem dapat menentukan ciri dari tanda tangan tersebut setelah sistem mendapatkan ciri dari

sebuah tanda tangan maka sudah dapat dilakukan pengenalan tanda tangan oleh sistem.

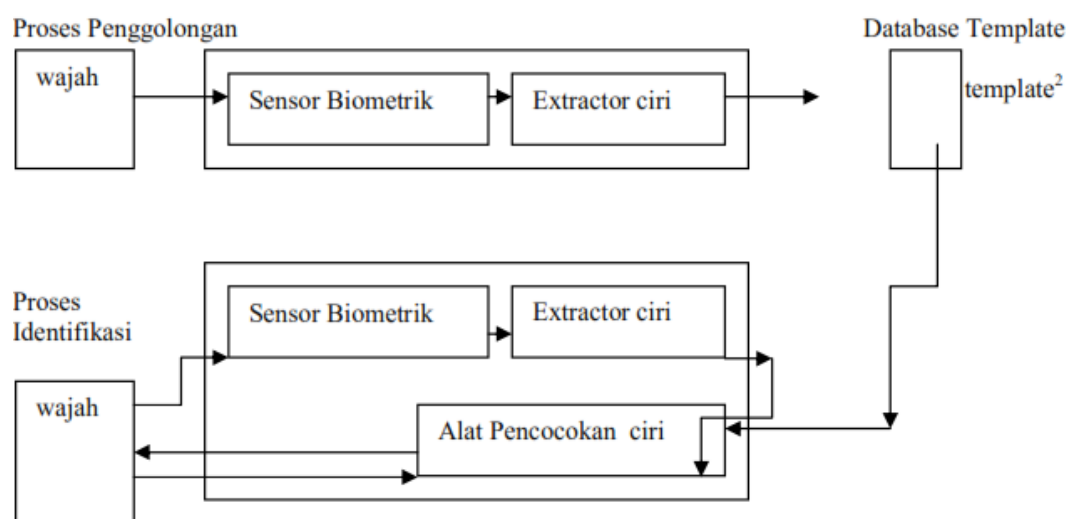


**Gambar 2.10 Kerangka Berfikir**

### 2.3 Penelitian yang Relevan

Sarwoko, Eko Adi (2006). "MEKANISME SISTEM IDENTIFIKASI BIOMETRIK."

Mekanisme sistem biometrik dapat digambarkan dengan beberapa fase, pertama fase penggolongan (enrollment). Pada fase ini masukan akan di pindai (scan) oleh sensor biometrik, yang merupakan representasi karakteristik digital. Selanjutnya fase pencocokan, dalam fase ini inputan database akan dicocokkan dengan identifikasi data. Dapat dimungkinkan adanya reduksi, sehingga dihasilkan representasi digital. Hasil ini akan diproses dengan ekstraktor ciri untuk menghasilkan suatu representasi yang ekspresif dalam bentuk template. Sedang pada fase pengenalan, karakteristik individu dibaca oleh pembaca biometrik (reader). Selanjutnya dikonversi dengan format digital, untuk diproses sebagai ekstraktor ciri (template). Hasil template ini selanjutnya dicocokkan dengan identifikasi individu. Lihat gambar 2.11



**Gambar 2.11 Mekanisme Sistem Biometrik**

Sarwoko menyimpulkan bahwa Mekanisme identifikasi biometrik ini mampu mengidentifikasi individu-individu berdasarkan perbedaan lingkup karakteristik behaviour/psikologi (biometric identifier) yang diyakini bersifat unik. Selain itu identifier biometrik dianggap lebih reliable dibandingkan berdasarkan pemasukan token dan pengenalan knowledge.

Ainun, J., Mohammad Isa, I., & Imam, M. (2011). Pengenalan Pola Tanda Tangan Menggunakan Metode *Moment Invariant* Dan Jaringan Syaraf Radial Basis Function (RBF).

Jaringan Syaraf Tiruan (JST) Radial Basis Function (RBF) dikenal sebagai salah satu bentuk dari jaringan syaraf Feedforward lapis banyak yang handal dalam memecahkan masalah aproksimasi dan klasifikasi/pengenalan pola. Dalam penelitian ini JST RBF digunakan untuk klasifikasi pola-pola tanda tangan berdasarkan output jaringan yang mempunyai kemiripan dengan target (pemilik tanda tangan). Metode *Moment Invariant* digunakan untuk mengekstraksi citra tanda tangan menjadi suatu vektor input yang merepresentasikan setiap citra tanda tangan. Satu citra tanda tangan diwakili oleh 7 *Moment Invariant*. Hasil klasifikasi tanda tangan yang dikenali dengan tingkat akurasi tertentu. Untuk menguji hasil klasifikasi dilakukan uji coba. Dari hasil uji coba program menunjukkan bahwa JST RBF dapat mengenali pola tanda tangan dengan akurasi pada data uji 80 %

Salambue, R. (2013). Pengenalan Pola Tanda Tangan dengan Metode *Moment Invariant* dan *Euclidean Distance*.

Dalam makalah ini dilakukan penelitian tentang pengenalan pola tanda tangan berbasis citra digital. Citra tanda tangan merupakan hasil digitasi yang direpresentasikan dalam bentuk matrik. Dalam melakukan pengenalan pola, matrik yang mempunyai dimensi besar berpengaruh pada waktu komputasi dan akurasi pengenalan. Berdasarkan hal tersebut maka digunakan metode *Moment Invariant* untuk mereduksi dimensi matrik. Prinsip kerja *Moment Invariant* adalah mengelompokkan matrik ke dalam vektor yang dihasilkan oleh fungsi posisi dan arah piksel citra yang invariant terhadap rotasi, translasi dan skala dan didefinisikan dalam momen geometri citra digital. Vektor hasil *Moment Invariant* menjadi data input dalam proses pengenalan citra digital. Metode pengenalan menggunakan *Euclidean Distance* yang mengukur selisih antara vektor. Ukuran kemiripan citra digital berdasarkan pada selisih minimum yang terjadi antara citra pengujian dengan citra yang tersimpan dalam basis data. Salambue menyimpulkan bahwa *Moment Invariant* dapat digunakan untuk memperoleh matrik berdimensi rendah sehingga mempercepat waktu komputasi, Metode ini tidak terpengaruh terhadap rotasi, translasi dan skala citra, dan Semua citra uji dapat dikenali.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1. Tempat dan Waktu Penelitian**

Penelitian dilakukan di laboratorium komputer Jurusan Teknik Elektro Universitas Negeri Jakarta, dimulai dari bulan April 2016 sampai bulan Oktober 2016.

#### **3.2. Alat dan Bahan Penelitian**

Perangkat keras dalam penelitian ini menggunakan seperangkat komputer dengan spesifikasi seperti di bawah ini :

1. Prosesor Intel Core i7-4700 CPU @ 2.40GHz(8CPUs)
2. Memori 8192 RAM
3. NVIDIA GeForce 755m
4. Mouse
5. Keyboard

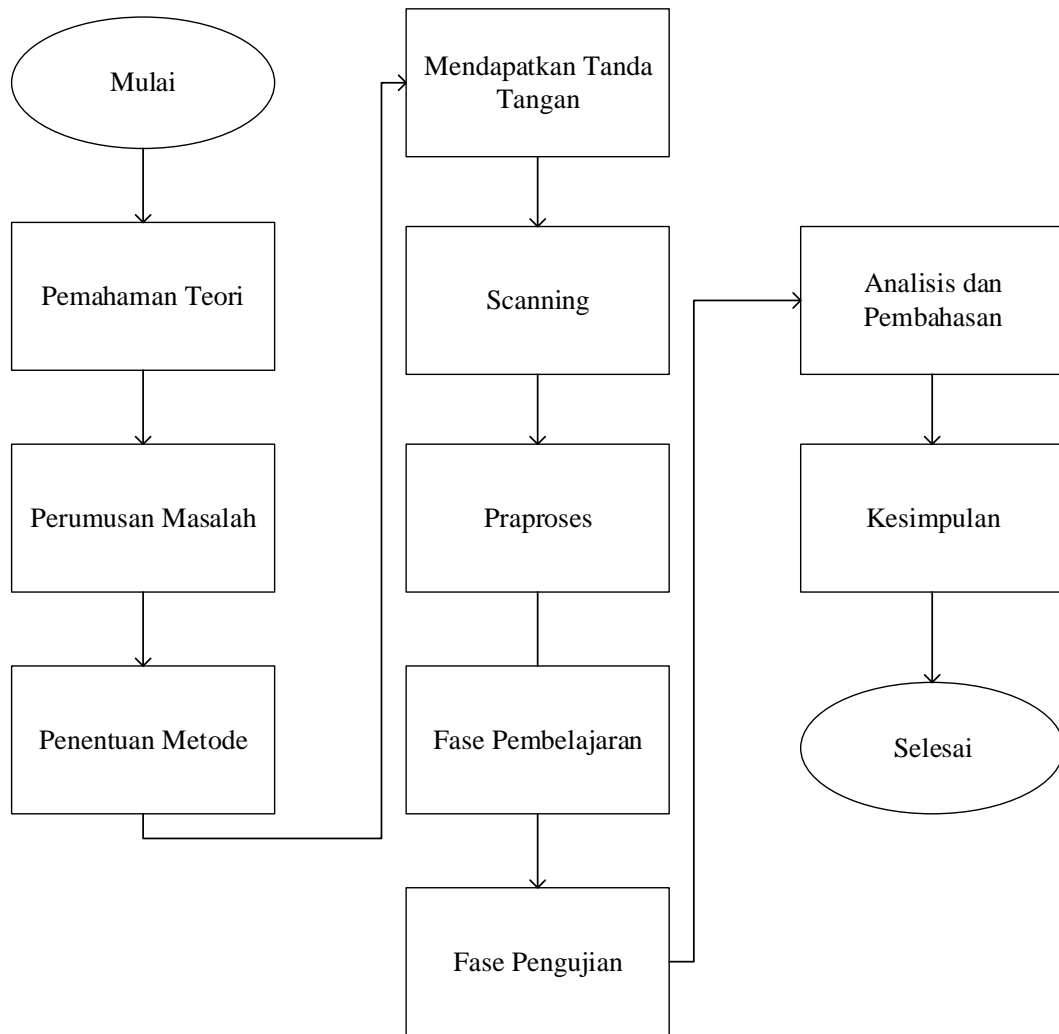
Adapun perangkat lunak yang digunakan adalah sebagai berikut :

1. Sistem operasi Windows 8.1 SL 64-bit
2. Microsoft Office 2016
3. Epson Scan
4. Matlab R2014b

#### **3.3. Diagram Alir Penelitian**

Untuk diagram alir pada penelitian dapat dilihat pada gambar 3.1. mulai dari proses awal penelitian hingga selesai.





**Gambar 3.1. Diagram Alir Penelitian**

Penjelasan dari gambar di atas adalah sebagai berikut:

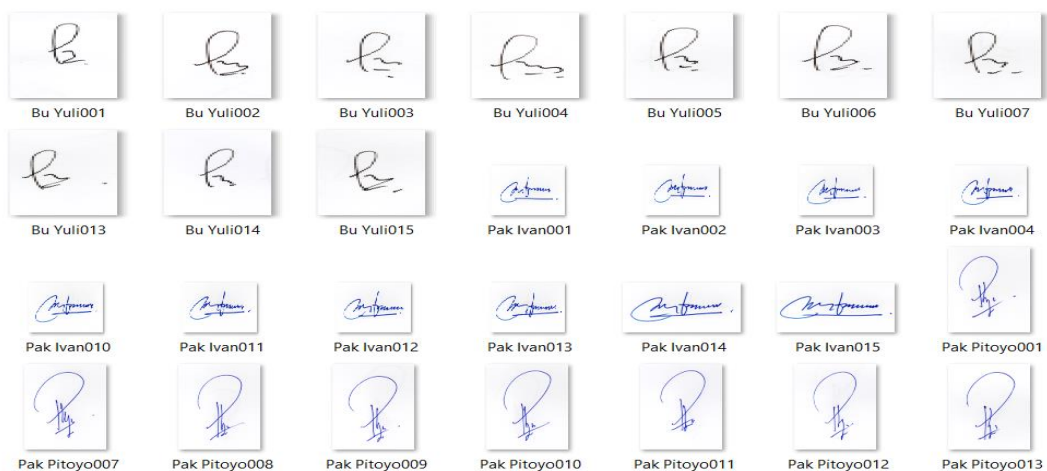
1. Pemahaman teori, yaitu dilakukannya pencarian dan pemahaman terkait materi dan bahan pendukung penelitian sehingga didapatkan beberapa permasalahan untuk dilakukannya penelitian.
2. Perumusan masalah dan tujuan penelitian, setelah permasalahan didapat kemudian dilakukannya perumusan dan tujuan dari penelitian.
3. Penentuan metode penelitian, yaitu menentukan metode apa yang tepat untuk menyelesaikan permasalahan yang didapat pada penelitian ini.

4. Mendapatkan tanda tangan asli pejabat-pejabat fakultas teknik universitas negeri jakarta dengan media kertas dan pulpen yang berjumlah 15 tanda tangan setiap pejabat fakultas teknik dan 15 tanda tangan palsu untuk masing masing tanda tangan asli. Pejabat fakultas teknik yang diambil tanda tangannya adalah sebagai berikut:
  - 15 Tanda Tangan Asli Dr. Yuliatri S., M.Pd.
  - 15 Tanda Tangan Palsu Dr. Yuliatri S., M.Pd.
  - 15 Tanda Tangan Asli Drs.Pitoyo Yuliatmojo, MT.
  - 15 Tanda Tangan Palsu Drs.Pitoyo Yuliatmojo, MT.
  - 15 Tanda Tangan Asli Prof. Dr. Ivan Hanafi, M.Pd.
  - 15 Tanda Tangan Palsu Prof. Dr. Ivan Hanafi, M.Pd.
  - 15 Tanda Tangan Asli Massus Subekti, S.Pd, MT
  - 15 Tanda Tangan Palsu Massus Subekti, S.Pd, MT
  - 15 Tanda Tangan Asli Syufrijal, MT.
  - 15 Tanda Tangan Palsu Syufrijal, MT.
5. *Scan* tanda tangan fisik menjadi digital dengan format file BMP untuk diproses lebih lanjut.
6. Proses Pengubahan data gambar (*grayscale*, normalisasi, penajaman citra, dan pemotongan citra) dan konversi format file dari BMP menjadi PBM.
7. Fase pembelajaran yaitu memasukan 13 dari tanda tangan masing-masing pejabat-pejabat fakultas teknik universitas negeri Jakarta dan tanda tangan palsu kedalam algoritma.

8. Fase Pengenalan yaitu memasukan sisa dari tanda tangan untuk mengetahui apakah tanda tangan dapat dikenali oleh algoritma dengan benar atau tidak.
9. Analisis dan pembahasan, pada tahap ini dilakukan analisis terhadap hasil keluaran yang telah didapatkan serta pembahasan terkait data tersebut.
10. Kesimpulan dan saran, ini merupakan tahap akhir pada penelitian dengan memberikan kesimpulan atas penelitian yang dilakukan dan memberikan saran untuk penelitian selanjutnya.

### 3.4. Teknik dan Prosedur Pengumpulan Data

Pengumpulan data dilakukan secara manual dengan mengumpulkan data-data yang diperlukan untuk mendukung penelitian yang dilakukan. Pengenalan menggunakan matlab dengan menggunakan fungsi yang tersedia di matlab. Masukan berasal dari tanda tangan fisik yang kemudia akan di *scan* menggunakan printer langsung kedalam bentuk BMP (gambar 3.2).



**Gambar 3.2 Tanda tangan yang telah di scan**

Hal ini bertujuan untuk memudahkan proses selanjutnya karena BMP berisikan pemetaan angka-angka yang dapat digunakan kembali untuk pengolahan data.

Setelah data masukan didapatkan, berikutnya adalah Praproses dimana data yang telah didapat akan dirubah dari berwarna menjadi hitam-putih, penambahan ketajaman citra, pemotongan citra, perubahan ukuran pixel citra dan normalisasi citra. Lalu masuk ke fase pembelajaran dan fase pengenalan.

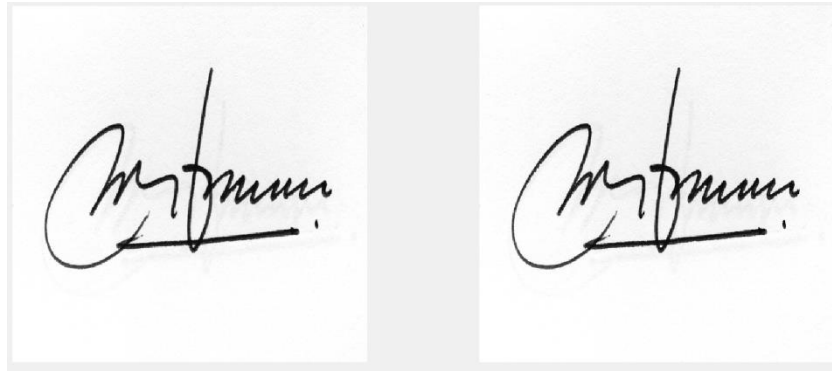
#### 3.4.1. Praproses Citra

Dalam Praproses citra, citra dari hasil scan akan melalui serangkaian perubahan agar citra dapat diproses oleh *perceptron* untuk fase pembelajaran dan pengenalan. Langkah awal yang akan dilakukan adalah penyaman ukuran citra tanda tangan yang beragam menjadi 512x512 menggunakan fungsi “*Imresize*” dari matlab, setelah ukuran citra disamakan maka citra akan dirubah menjadi citra *grayscale* (gambar 3.2) dengan menggunakan fungsi “*rgb2gray*”.



**Gambar 3.3 Citra asli (kiri) dan sesudah *grayscale* (kanan)**

Setelah citra dirubah menjadi *grayscale* selanjutnya citra akan ditajamkan menggunakan fungsi “*imsharpen*” yang bertujuan untuk menonjolkan tepian citra (gambar 3.3).



**Gambar 3.4** Citra sebelum ditajamkan (kiri) dan sesudah (kanan)

Citra yang telah ditajamkan akan dinormalisasi agar algoritma *perceptron* dapat menerima citra hal ini dilakukan dengan menggunakan fungsi matlab “im2bw” (gambar 3.4).



**Gambar 3.5** Citra sebelum normalisasi (kiri) dan sesudah (kanan)

Setelah normalisasi maka akan dilakukan pemotongan citra (gambar 3.5)



**Gambar 3.6** Citra sebelum dipotong (kiri) dan sesudah (kanan)

Setelah Pemotongan citra maka akan dilakukan penyaman ukuran citra tanda tangan yang berubah akibat dari pemotongan citra menjadi 256x256. Setelah disesuaikan maka praproses telah selesai dan siap untuk dimasukkan kedalam algoritma *perceptron*.

### 3.4.2. Fase Pembelajaran dan Pengujian

Fase pembelajaran bertujuan untuk mengambil ciri dari citra tanda tangan yang akan digunakan untuk membentuk jaringan *perceptron*. Data yang digunakan adalah 135 tanda tangan untuk pembelajaran dan 15 tanda tangan untuk pengujian. Pengambilan tanda tangan yang digunakan untuk pengujian dilakukan secara sampling dengan menyertakan setidaknya satu dari setiap tanda tangan yang telah dilakukan praproses.

Setelah pemisahan data pembelajaran dan pengujian selanjutnya data akan dijadikan matriks 1 dimensi sehingga matriks 256x256 akan menjadi 65536x1 setelah seluruh data pembelajaran dirubah menjadi matriks 1 dimensi maka selanjutnya matriks 1 dimensi akan digabungkan menjadi satu matriks menjadi 65536x135 dimana masing-masing tanda tangan menempati satu kolom untuk selanjutnya matriks ini akan disebut sebagai matriks "P".

Setelah Matriks P didapat maka selanjutnya matriks penentuan target akan dibentuk. Tetapi karena algoritma *single layer perceptron* bersifat *linearly seperable*, yang hanya memungkinkan *perceptron* untuk memisahkan tanda tangan menjadi dua kategori sehingga akan dilakukan penyesuaian dalam penentuan target dimana akan terdapat 5 target yang masing-masing bertujuan untuk memisahkan tanda tangan asli salah satu pejabat UNJ dengan tanda tangan palsu dan tanda

tangan asli pejabat UNJ lainnya. Untuk selanjutnya matriks penentuan target akan disebut sebagai Matriks “T1, T2, T3, T4, T5”.

Setelah ini Matriks P dan T1 sampai T5 akan dimasukkan kedalam *Perceptron* yang dilakukan dengan menggunakan fungsi matlab “net = perceptron” untuk membuat jaringan syaraf tiruan *perceptron* baru dan “net = train (net, P, T1)” untuk melatih *perceptron* baru tersebut. Proses akan diulang sampai seluruh matriks T digunakan, sehingga akan terdapat 5 net yang berbeda (net1-net5) untuk masing masing target. Setelah selesai akan dilakukan fase pengujian dimana jaringan *perceptron* yang telah dilatih oleh 135 tanda tangan akan dicoba untuk mengenali 15 tanda tangan yang tidak termasuk kedalam matriks P.

Sebelum memulai pengujian, ke 15 tanda tangan harus terlebih dahulu dijadikan matriks 1 dimensi seperti matriks P, Matriks ini selanjutnya akan disebut sebagai matriks C. pengujian akan dilakukan menggunakan fungsi dari matlab “sim(net, C)” proses ini akan berlangsung sesuai dengan jumlah net yang ada, hasil proses selanjutnya akan disebut sebagai “R” setelah mendapatkan 5 hasil dari proses pengujian selanjutnya kelima matriks R akan digabung menjadi satu dengan melakukan pejumlahan matriks biasa dan akan menghasilkan matriks TR atau hasil total baru setelah itu data bisa dianalisis.

### **3.5 Teknik Analisis Data**

Analisis data dilakukan dengan menggunakan *k-fold cross-validation*. Pengujian *k-fold cross-validation* adalah salah satu teknik untuk menilai atau memvalidasi keakuratan sebuah model yang dibangun berdasarkan sebuah data tertentu. Pembuatan model biasanya bertujuan untuk melakukan klasifikasi terhadap suatu data baru yang boleh jadi belum pernah muncul dalam data. Data

yang digunakan dalam proses pembangunan disebut dengan data latih, sedangkan data yang digunakan untuk memvalidasi adalah data *test*. Dalam teknik *k-fold cross-validation* dataset dibagi menjadi sejumlah “k” partisi secara acak. Kemudian dilakukan eksperimen sejumlah “k”, dimana masing-masing eksperimen menggunakan data partisi ke “k” sebagai data *test* dan memanfaatkan partisi lainnya sebagai data latih. Sebagai contoh *10-fold cross-validation* memiliki desain eksperimen sebagai berikut:

Data:

K1 | K2 | K3 | K4 | K5 | K6 | K7 | K8 | K9 | K10

Eksperimen

Eksperimen Ke-	Data Latih	Data <i>Test</i>
1	K2   K3   K4   K5   K6   K7   K8   K9   K10	K1
2	K1   K3   K4   K5   K6   K7   K8   K9   K10	K2
3	K1   K2   K4   K5   K6   K7   K8   K9   K10	K3
4	K1   K2   K3   K5   K6   K7   K8   K9   K10	K4
5	K1   K2   K3   K4   K6   K7   K8   K9   K10	K5
6	K1   K2   K3   K4   K5   K7   K8   K9   K10	K6
7	K1   K2   K3   K4   K5   K6   K8   K9   K10	K7
8	K1   K2   K3   K4   K5   K6   K7   K9   K10	K8
9	K1   K2   K3   K4   K5   K6   K7   K8   K10	K9
10	K1   K2   K3   K4   K5   K6   K7   K8   K9	K10

Untuk mendapatkan nilai hasil akurasi dari hasil eksperimen yang dilakukan, dapat dilakukan dengan mengambil nilai rata-rata dari seluruh eksperimen.



## **BAB IV HASIL PENELITIAN**

### **4.1. Deskripsi Hasil Penelitian**

Data tanda tangan yang diperoleh dalam penelitian ini diambil secara langsung dari dari pejabat fakultas teknik UNJ. Jumlah data tanda tangan yang diperoleh adalah 75 tanda tangan asli yang diambil dari 5 pejabat fakultas teknik yaitu:

- Dr. Yuliatri S., M.Pd.
- Drs.Pitoyo Yuliatmojo, MT.
- Prof. Dr. Ivan Hanafi, M.Pd
- Massus Subekti, S.Pd, MT
- Syufrijal, MT.

Dan 75 tanda tangan palsu untuk setiap tanda tangan asli yang diambil dari beberapa mahasiswa fakultas teknik sehingga keseluruhan tanda tangan berjumlah 150 tanda tangan.

Setiap tanda tangan diberikan nama *file*, sebagai contoh tanda tangan Dr. Yuliatri S., M.Pd. akan memiliki nama *file* sebagai berikut Bu Yuli001 untuk tanda tangan asli dan Bu Yuli Palsu001 untuk tanda tangan palsu maka *file* akan memiliki angka 001 sampai 015 untuk setiap tanda tangan asli atau palsu yang ada. Tiga angka belakang nama *file* ini untuk selanjutnya akan disebut sebagai nomor *file*



**Gambar 4.1. Tanda Tangan Sebelum Pra-proses**

Pra-proses akan terlebih dahulu dilakukan untuk seluruh tanda tangan, pra-proses meliputi beberapa hal grayscale, normalisasi, pemotongan citra, perubahan jenis file, penajaman citra.



**Gambar 4.2. Tanda Tangan Sesudah Pra-proses**

Setelah semua tanda tangan telah di pra-proses maka selanjutnya akan dilakukan proses pelatihan dan pengujian yang menggunakan model *10 fold-cross validation* untuk mendapatkan hasil yang dapat dipergunakan, seperti yang telah dibahas pada poin 3.5 teknik analisis data model *10 fold-cross validation* berarti proses pelatihan dan pengujian dilakukan sebanyak 10 kali atau terdapat 10 percobaan. setiap percobaan wajib memiliki setidaknya satu dari setiap tanda tangan yang ada (asli dan palsu). Pengambilan tanda tangan dilakukan dengan berdasarkan kepada sorting komputer secara pengurutan berdasarkan nama *file* . Percobaan ganjil / genap akan memiliki jumlah tanda tangan masing masing pejabat yang sama hal terjadi karena pengambilan tanda tangan dilakukan berdasarkan kepada sorting nama *file* secara berurutan.

Berikut adalah kelompok percobaan untuk *10 fold-cross validation* berdasarkan nama *file* masing masing tanda tangan:

## Percobaan 1

*Train***Tabel 4.1 Data *train* percobaan 1**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	003-015	13
2	Bu Yuli	003-015	13
3	Pak Ivan Palsu	003-015	13
4	Pak Ivan	003-015	13
5	Pak Massus	003-015	13
6	Pak Massus Palsu	002-015	14
7	Pak Pitoyo Palsu	002-015	14
8	Pak Pitoyo	002-015	14
9	Pak Syufrijal Palsu	002-015	14
10	Pak Syufrijal	002-015	14

*Test***Tabel 4.2 Data *test* percobaan 1**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	001-002	2
2	Bu Yuli	001-002	2
3	Pak Ivan Palsu	001-002	2
4	Pak Ivan	001-002	2
5	Pak Massus	001-002	2
6	Pak Massus Palsu	001	1
7	Pak Pitoyo Palsu	001	1
8	Pak Pitoyo	001	1
9	Pak Syufrijal Palsu	001	1
10	Pak Syufrijal	001	1

Pada tabel 4.1 terdapat 135 tanda tangan asli dan palsu yang akan digunakan sebagai data *train* dan tabel 4.2 yang berisi 15 tanda tangan asli dan palsu yang digunakan sebagai data *test*

## Percobaan 2

*Train***Tabel 4.3 Data *train* percobaan 2**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	001-002, 004-015	14
2	Bu Yuli	001-002, 004-015	14
3	Pak Ivan Palsu	001-002, 004-015	14
4	Pak Ivan	001-002, 004-015	14
5	Pak Massus	001-002, 004-015	14
6	Pak Massus Palsu	001, 004-015	13
7	Pak Pitoyo Palsu	001, 004-015	13
8	Pak Pitoyo	001, 004-015	13
9	Pak Syufrijal Palsu	001, 004-015	13
10	Pak Syufrijal	001, 004-015	13

*Test***Tabel 4.4 Data *test* percobaan 2**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	003	1
2	Bu Yuli	003	1
3	Pak Ivan Palsu	003	1
4	Pak Ivan	003	1
5	Pak Massus	003	1
6	Pak Massus Palsu	002-003	2
7	Pak Pitoyo Palsu	002-003	2
8	Pak Pitoyo	002-003	2
9	Pak Syufrijal Palsu	002-003	2
10	Pak Syufrijal	002-003	2

Pada tabel 4.3 terdapat 135 tanda tangan asli dan palsu yang akan digunakan sebagai data *train* dan tabel 4.4 yang berisi 15 tanda tangan asli dan palsu yang digunakan sebagai data *test*

## Percobaan 3

*Train***Tabel 4.5 Data *train* percobaan 3**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	001-003, 006-015	13
2	Bu Yuli	001-003, 006-015	13
3	Pak Ivan Palsu	001-003, 006-015	13
4	Pak Ivan	001-003, 006-015	13
5	Pak Massus	001-003, 006-015	13
6	Pak Massus Palsu	001-003, 005-015	14
7	Pak Pitoyo Palsu	001-003, 005-015	14
8	Pak Pitoyo	001-003, 005-015	14
9	Pak Syufrijal Palsu	001-003, 005-015	14
10	Pak Syufrijal	001-003, 005-015	14

*Test***Tabel 4.6 Data *test* percobaan 3**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	004-005	2
2	Bu Yuli	004-005	2
3	Pak Ivan Palsu	004-005	2
4	Pak Ivan	004-005	2
5	Pak Massus	004-005	2
6	Pak Massus Palsu	004	1
7	Pak Pitoyo Palsu	004	1
8	Pak Pitoyo	004	1
9	Pak Syufrijal Palsu	004	1
10	Pak Syufrijal	004	1

Pada tabel 4.5 terdapat 135 tanda tangan asli dan palsu yang akan digunakan sebagai data *train* dan tabel 4.6 yang berisi 15 tanda tangan asli dan palsu yang digunakan sebagai data *test*

## Percobaan 4

*Train***Tabel 4.7 Data *train* percobaan 4**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	001-005, 007-015	14
2	Bu Yuli	001-005, 007-015	14
3	Pak Ivan Palsu	001-005, 007-015	14
4	Pak Ivan	001-005, 007-015	14
5	Pak Massus	001-005, 007-015	14
6	Pak Massus Palsu	001-004, 007-015	13
7	Pak Pitoyo Palsu	001-004, 007-015	13
8	Pak Pitoyo	001-004, 007-015	13
9	Pak Syufrijal Palsu	001-004, 007-015	13
10	Pak Syufrijal	001-004, 007-015	13

*Test***Tabel 4.8 Data *test* percobaan 4**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	006	1
2	Bu Yuli	006	1
3	Pak Ivan Palsu	006	1
4	Pak Ivan	006	1
5	Pak Massus	006	1
6	Pak Massus Palsu	005-006	2
7	Pak Pitoyo Palsu	005-006	2
8	Pak Pitoyo	005-006	2
9	Pak Syufrijal Palsu	005-006	2
10	Pak Syufrijal	005-006	2

Pada tabel 4.7 terdapat 135 tanda tangan asli dan palsu yang akan digunakan sebagai data *train* dan tabel 4.8 yang berisi 15 tanda tangan asli dan palsu yang digunakan sebagai data *test*

## Percobaan 5

*Train***Tabel 4.9 Data *train* percobaan 5**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	001-006, 009-015	13
2	Bu Yuli	001-006, 009-015	13
3	Pak Ivan Palsu	001-006, 009-015	13
4	Pak Ivan	001-006, 009-015	13
5	Pak Massus	001-006, 009-015	13
6	Pak Massus Palsu	001-006, 008-015	14
7	Pak Pitoyo Palsu	001-006, 008-015	14
8	Pak Pitoyo	001-006, 008-015	14
9	Pak Syufrijal Palsu	001-006, 008-015	14
10	Pak Syufrijal	001-006, 008-015	14

*Test***Tabel 4.10 Data *test* percobaan 5**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	007-008	2
2	Bu Yuli	007-008	2
3	Pak Ivan Palsu	007-008	2
4	Pak Ivan	007-008	2
5	Pak Massus	007-008	2
6	Pak Massus Palsu	007	1
7	Pak Pitoyo Palsu	007	1
8	Pak Pitoyo	007	1
9	Pak Syufrijal Palsu	007	1
10	Pak Syufrijal	007	1

Pada tabel 4.9 terdapat 135 tanda tangan asli dan palsu yang akan digunakan sebagai data *train* dan tabel 4.10 yang berisi 15 tanda tangan asli dan palsu yang digunakan sebagai data *test*



## Percobaan 6

*Train***Tabel 4.11 Data *train* percobaan 6**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	001-008, 010-015	14
2	Bu Yuli	001-008, 010-015	14
3	Pak Ivan Palsu	001-008, 010-015	14
4	Pak Ivan	001-008, 010-015	14
5	Pak Massus	001-008, 010-015	14
6	Pak Massus Palsu	001-007, 010-015	13
7	Pak Pitoyo Palsu	001-007, 010-015	13
8	Pak Pitoyo	001-007, 010-015	13
9	Pak Syufrijal Palsu	001-007, 010-015	13
10	Pak Syufrijal	001-007, 010-015	13

*Test***Tabel 4.12 Data *test* percobaan 6**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	009	1
2	Bu Yuli	009	1
3	Pak Ivan Palsu	009	1
4	Pak Ivan	009	1
5	Pak Massus	009	1
6	Pak Massus Palsu	008-009	2
7	Pak Pitoyo Palsu	008-009	2
8	Pak Pitoyo	008-009	2
9	Pak Syufrijal Palsu	008-009	2
10	Pak Syufrijal	008-009	2

Pada tabel 4.11 terdapat 135 tanda tangan asli dan palsu yang akan digunakan sebagai data *train* dan tabel 4.12 yang berisi 15 tanda tangan asli dan palsu yang digunakan sebagai data *test*

## Percobaan 7

*Train***Tabel 4.13 Data *train* percobaan 7**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	001-009, 012-015	13
2	Bu Yuli	001-009, 012-015	13
3	Pak Ivan Palsu	001-009, 012-015	13
4	Pak Ivan	001-009, 012-015	13
5	Pak Massus	001-009, 012-015	13
6	Pak Massus Palsu	001-009, 011-015	14
7	Pak Pitoyo Palsu	001-009, 011-015	14
8	Pak Pitoyo	001-009, 011-015	14
9	Pak Syufrijal Palsu	001-009, 011-015	14
10	Pak Syufrijal	001-009, 011-015	14

*Test***Tabel 4.14 Data *test* percobaan 7**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	010-011	2
2	Bu Yuli	010-011	2
3	Pak Ivan Palsu	010-011	2
4	Pak Ivan	010-011	2
5	Pak Massus	010-011	2
6	Pak Massus Palsu	010	1
7	Pak Pitoyo Palsu	010	1
8	Pak Pitoyo	010	1
9	Pak Syufrijal Palsu	010	1
10	Pak Syufrijal	010	1

Pada tabel 4.13 terdapat 135 tanda tangan asli dan palsu yang akan digunakan sebagai data *train* dan tabel 4.14 yang berisi 15 tanda tangan asli dan palsu yang digunakan sebagai data *test*

## Percobaan 8

*Train***Tabel 4.15 Data *train* percobaan 8**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	001-011, 013-015	14
2	Bu Yuli	001-011, 013-015	14
3	Pak Ivan Palsu	001-011, 013-015	14
4	Pak Ivan	001-011, 013-015	14
5	Pak Massus	001-011, 013-015	14
6	Pak Massus Palsu	001-010, 013-015	13
7	Pak Pitoyo Palsu	001-010, 013-015	13
8	Pak Pitoyo	001-010, 013-015	13
9	Pak Syufrijal Palsu	001-010, 013-015	13
10	Pak Syufrijal	001-010, 013-015	13

*Test***Tabel 4.16 Data *test* percobaan 8**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	012	1
2	Bu Yuli	012	1
3	Pak Ivan Palsu	012	1
4	Pak Ivan	012	1
5	Pak Massus	012	1
6	Pak Massus Palsu	011-012	2
7	Pak Pitoyo Palsu	011-012	2
8	Pak Pitoyo	011-012	2
9	Pak Syufrijal Palsu	011-012	2
10	Pak Syufrijal	011-012	2

Pada tabel 4.15 terdapat 135 tanda tangan asli dan palsu yang akan digunakan sebagai data *train* dan tabel 4.16 yang berisi 15 tanda tangan asli dan palsu yang digunakan sebagai data *test*

## Percobaan 9

*Train***Tabel 4.17 Data *train* percobaan 9**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	001-012, 015	13
2	Bu Yuli	001-012, 015	13
3	Pak Ivan Palsu	001-012, 015	13
4	Pak Ivan	001-012, 015	13
5	Pak Massus	001-012, 015	13
6	Pak Massus Palsu	001-012, 014-015	14
7	Pak Pitoyo Palsu	001-012, 014-015	14
8	Pak Pitoyo	001-012, 014-015	14
9	Pak Syufrijal Palsu	001-012, 014-015	14
10	Pak Syufrijal	001-012, 014-015	14

*Test***Tabel 4.18 Data *test* percobaan 9**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	013-014	2
2	Bu Yuli	013-014	2
3	Pak Ivan Palsu	013-014	2
4	Pak Ivan	013-014	2
5	Pak Massus	013-014	2
6	Pak Massus Palsu	013	1
7	Pak Pitoyo Palsu	013	1
8	Pak Pitoyo	013	1
9	Pak Syufrijal Palsu	013	1
10	Pak Syufrijal	013	1

Pada tabel 4.17 terdapat 135 tanda tangan asli dan palsu yang akan digunakan sebagai data *train* dan tabel 4.18 yang berisi 15 tanda tangan asli dan palsu yang digunakan sebagai data *test*

## Percobaan 10

*Train***Tabel 4.19 Data *train* percobaan 10**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	001-014	14
2	Bu Yuli	001-014	14
3	Pak Ivan Palsu	001-014	14
4	Pak Ivan	001-014	14
5	Pak Massus	001-014	14
6	Pak Massus Palsu	001-013	13
7	Pak Pitoyo Palsu	001-013	13
8	Pak Pitoyo	001-013	13
9	Pak Syufrijal Palsu	001-013	13
10	Pak Syufrijal	001-013	13

*Test***Tabel 4.20 Data *test* percobaan 10**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	015	1
2	Bu Yuli	015	1
3	Pak Ivan Palsu	015	1
4	Pak Ivan	015	1
5	Pak Massus	015	1
6	Pak Massus Palsu	014-015	2
7	Pak Pitoyo Palsu	014-015	2
8	Pak Pitoyo	014-015	2
9	Pak Syufrijal Palsu	014-015	2
10	Pak Syufrijal	014-015	2

Pada tabel 4.19 terdapat 135 tanda tangan asli dan palsu yang akan digunakan sebagai data *train* dan tabel 4.20 yang berisi 15 tanda tangan asli dan palsu yang digunakan sebagai data *test*

Dalam algoritma *perceptron*, *perceptron* tidak dapat menerima langsung input matriks biasa (gambar 4.3), seluruh matriks harus terlebih dahulu dirubah menjadi bentuk matriks 1-dimensi (gambar 4.4),

10	9	8	
7	10	8	
1	7	4	

**Gambar 4.3 Matriks biasa**

10		
7		
1		
9		
10		
7		
8		
8		
4		

**Gambar 4.4 Matriks 1-dimensi dari gambar 4.3**

Setelah matriks dijadikan matriks 1-dimensi maka matriks telah dapat digunakan untuk masuk kedalam algoritma *perceptron*, tetapi sebelum matriks dari data *train* dimasukan kedalam *perceptron*, sebuah target harus terlebih dahulu ditetapkan.

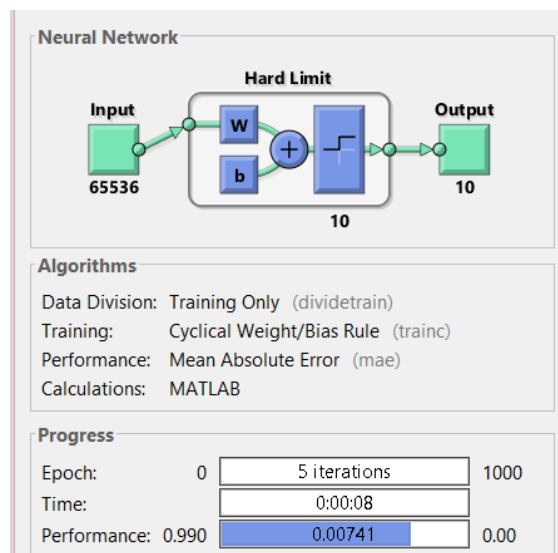
Hal ini dikarenakan *perceptron* merupakan algoritma *supervised learning* yang berarti *perceptron* memerlukan data yang telah dikenali terlebih dahulu sebelum mengklasifikasikan data yang belum dikenali dan dikarenakan *perceptron* juga bersifat *linearly seperable* maka akan terdapat 5 target yang berbeda untuk 5 *perceptron* yang berbeda.

Masing-masing target berfungsi untuk memisahkan satu tanda tangan terhadap tanda tangan lainnya, misal target yang digunakan untuk mengenali tanda tangan bu yuli maka target tersebut hanya akan mengenali tanda tangan bu yuli saja.

Setelah target ditetapkan maka *perceptron* langsung digunakan dan di *train* dengan perintah matlab :

```
net = perceptron;
net = train (net, P, T);
```

dimana P adalah matriks data *train* yang telah dijadikan 1-dimensi dan T adalah target yang telah ditetapkan sebelumnya berikut adalah proses *train* algoritma *perceptron* di matlab



**Gambar 4.5** Proses *training perceptron* di matlab

Setelah seluruh *perceptron* di *train* maka *perceptron* telah siap untuk mengklasifikasi data yang tidak terdapat dalam data *train* untuk mencoba pengklasifikasian data dari *perceptron* yang telah di *train* dapat dengan menggunakan perintah matlab

```
Sim (net, S)
```

Dimana net merupakan *perceptron* yang telah di *train* dan S adalah sebuah data *test* yang tidak terdapat pada data *train*, data *test* juga harus berupa matriks 1-dimensi.

#### 4.2. Analisis Data Penelitian

Setelah seluruh kelompok untuk percobaan 10 *fold-cross validation* terbentuk maka selanjutnya data tiap kelompok akan dimasukkan kedalam *perceptron*, sebelum memasuki pembahasan hasil perhatikan tabel berikut

**Tabel 4.21 Penjelasan hasil**

Nama File	Kode
Bu Yuli Palsu	A
Bu Yuli	B
Pak Ivan Palsu	C
Pak Ivan	D
Pak Massus	E
Pak Massus Palsu	F
Pak Pitoyo Palsu	G
Pak Pitoyo	H
Pak Syufrijal Palsu	I
Pak Syufrijal	J

Hasil mengacu kembali pada daftar percobaan dalam sub-bab sebelumnya misal pada percobaan 1 memiliki data *test* sebagai berikut

**Tabel 4.22 Data *test* percobaan 1**

No	Nama File	Nomor File	Jumlah
1	Bu Yuli Palsu	001-002	2
2	Bu Yuli	001-002	2
3	Pak Ivan Palsu	001-002	2
4	Pak Ivan	001-002	2
5	Pak Massus	001-002	2
6	Pak Massus Palsu	001	1
7	Pak Pitoyo Palsu	001	1
8	Pak Pitoyo	001	1
9	Pak Syufrijal Palsu	001	1
10	Pak Syufrijal	001	1

Terlihat dalam tabel bahwa terdapat dua *file* dengan nama *file* Bu Yuli Palsu (nomor *file* 001 dan 002) dalam percobaan pertama akan terdapat A1 dan A2 dimana A1 mengacu kepada *file* Bu Yuli Palsu001 dan A2 mengacu kepada *file* Bu



Yuli Palsu002, B2 akan mengacu kepada *file* Bu Yuli002 dan seterusnya Sedangkan untuk membaca hasil dari percobaan adalah sebagai berikut

- Tanda tangan palsu harus bernilai 0 pada seluruh baris
- Tanda tangan Asli Harus bernilai 1 hanya pada baris pemilik tanda tangan tersebut

Sebagai contoh pada percobaan 1 D1 (*file* dengan nama Pak Ivan001) harus bernilai 1 pada baris Pak ivan dan J (*file* dengan nama pak Syufrijal001) harus bernilai 1 pada baris Pak Syufrijal (paling bawah). Apabila ada kesalahan nilai maka tanda tangan tersebut berarti tidak dikenali atau salah dikenali. Berikut adalah hasil dari percobaan *10 fold-cross validation*:

Percobaan 1

**Tabel 4.23 Hasil percobaan 1**

Pemilik TTD	A1	A2	B1	B2	C1	C2	D1	D2	E1	E2	F	G	H	I	J
Bu Yuli Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bu Yuli	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1
Pak Massus	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
Pak Massus Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Pada percobaan pertama *perceptron* mengenali hampir seluruh *test* data yang diberikan dengan pengecualian untuk dua *test* data yaitu F dan J dimana F dikenali sebagai tanda tangan asli pak massus dan J dikenali sebagai tanda tangan pak syufrijal dan pak ivan sekaligus. Tingkat akurasi untuk percobaan pertama adalah 13 data dikenali dari 15 data atau 86,67%

## Percobaan 2

**Tabel 4.24 Hasil percobaan 2**

Pemilik TTD	A	B	C	D	E	F1	F2	G1	G2	H1	H2	I1	I2	J1	J2
Bu Yuli Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bu Yuli	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pak Massus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Massus Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Pada Percobaan ke-dua terjadi lebih banyak kesalahan dari percobaan pertama *Perceptron* banyak mengenali tanda tangan asli sebagai tanda tangan palsu, data tanda tangan B, E, H1, dan H2 tidak dikenali secara berurutan sebagai Bu Yuli Asli, Pak Massus asli, dan Pak Pitoyo Asli. Secara keseluruhan tingkat akurasi pada percobaan ke-dua adalah 11 dari 15 data tanda tangan yaitu 73,33%.

## Percobaan 3

**Tabel 4.25 Hasil percobaan 3**

Pemilik TTD	A1	A2	B1	B2	C1	C2	D1	D2	E1	E2	F	G	H	I	J
Bu Yuli Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bu Yuli	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
Pak Massus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Massus Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Pak Syufrijal Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Pada percobaan ke-tiga *Perceptron* mengenali kedua tanda tangan asli pak massus sebagai tanda tangan palsu (E1 & E2) sedangkan tanda tangan asli lainnya dapat dikenali dengan baik. Tingkat akurasi untuk percobaan ke-tiga adalah 13 dari 15 data tanda tangan yaitu 86,67%

## Percobaan 4

**Tabel 4.26 Hasil percobaan 4**

Pemilik TTD	A	B	C	D	E	F1	F2	G1	G2	H1	H2	I1	I2	J1	J2
Bu Yuli Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bu Yuli	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pak Massus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Massus Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
Pak Syufrijal Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Percobaan ke-empat memiliki tingkat akurasi yang sama dengan percobaan ketiga, percobaan keempat gagal mengidentifikasi tanda tangan pak massus asli (E) dan satu tanda tangan pak syufrijal asli (J1). Tingkat akurasi untuk percobaan ke-empat adalah 13 dari 15 data tanda tangan yaitu 86,67%

## Percobaan 5

**Tabel 4.27 hasil percobaan 5**

Pemilik TTD	A1	A2	B1	B2	C1	C2	D1	D2	E1	E2	F	G	H	I	J
Bu Yuli Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bu Yuli	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
Pak Massus	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Pak Massus Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Pada percobaan ke-lima *Perceptron* gagal mengenali tanda tangan asli bu yuli, pak massus, dan pak pitoyo (B2, E2, dan H) sedangkan seluruh tanda tangan palsu berhasil dikenali dengan baik. Tingkat akurasi pada percobaan ke-lima adalah 12 dari 15 tanda tangan dikenali atau 80%

## Percobaan 6

Tabel 4.28 hasil percobaan 6

Pemilik TTD	A	B	C	D	E	F1	F2	G1	G2	H1	H2	I1	I2	J1	J2
Bu Yuli Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bu Yuli	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pak Massus	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Pak Massus Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Pak Syufrijal Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Pada percobaan ke-enam *Perceptron* gagal mengenali salah satu tanda tangan asli pak pitoyo (H1) dan kedua tanda tangan asli pak syufrijal (J1 & J2). Tingkat akurasi untuk percobaan ke-enam adalah 12 dari 15 tanda tangan dikenali atau 80%

## Percobaan 7

Tabel 4.29 hasil percobaan 7

Pemilik TTD	A1	A2	B1	B2	C1	C2	D1	D2	E1	E2	F	G	H	I	J
Bu Yuli Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bu Yuli	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Pak Massus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Massus Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Pak Syufrijal Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Pada percobaan ke-tujuh *perceptron* salah mengenali tanda tangan pak massus palsu sebagai tanda tangan pak syufrijal asli (F) selain itu *perceptron* juga gagal mengenali tanda tangan asli pak ivan dan pak massus (D1, E1, dan E2). Tingkat akurasi untuk percobaan ke-tujuh adalah 11 dari 15 tanda tangan dikenali atau 73,33%

## Percobaan 8

**Tabel 4.30 hasil percobaan 8**

Pemilik TTD	A	B	C	D	E	F1	F2	G1	G2	H1	H2	I1	I2	J1	J2
Bu Yuli Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bu Yuli	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pak Massus	1	0	0	0	1	0	1	0	0	0	0	0	0	1	1
Pak Massus Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Pada percobaan ke-delapan *perceptron* banyak mengalami kegagalan dalam mengenali tanda tangan, *perceptron* mengenali tanda tangan palsu bu yuli (A) dan tanda tangan asli pak syufrijal (J2) sebagai tanda tangan asli pak massus, mengenali tanda tangan palsu pak massus sebagai tanda tangan asli (F2), gagal mengenali tanda tangan asli pak pitoyo (H1 & H2) dan mengenali tanda tangan asli pak syufrijal (J1) sebagai tanda tangan asli pak massus dan pak syufrijal. Tingkat akurasi untuk percobaan ke-delapan adalah 9 dari 15 data dikenali yaitu 60%

## Percobaan 9

**Tabel 4.31 hasil percobaan 9**

Pemilik TTD	A1	A2	B1	B2	C1	C2	D1	D2	E1	E2	F	G	H	I	J
Bu Yuli Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bu Yuli	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
Pak Massus	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Pak Massus Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Pada percobaan ke-sembilan *perceptron* mengenali tanda tangan asli bu yuli (B1), tanda tangan asli pak massus (E2) dan tanda tangan asli pak pitoyo (H) sebagai tanda tangan palsu masing masing. Tingkat akurasi untuk percobaan ke-sembilan adalah 12 dari 15 tanda tangan dikenali atau 80%

## Percobaan 10

**Tabel 4.32 hasil percobaan 10**

Pemilik TTD	A	B	C	D	E	F1	F2	G1	G2	H1	H2	I1	I2	J1	J2
Bu Yuli Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bu Yuli	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Ivan	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Pak Massus	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Pak Massus Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Pitoyo	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal Palsu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pak Syufrijal	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Pada percobaan terakhir *perceptron* gagal mengenali tanda tangan asli bu yuli (B) dan kedua tanda tangan pak pitoyo (H1 & H2) sedangkan tanda tangan asli dan palsu lainnya dapat dikenali dengan baik. Tingkat akurasi untuk percobaan terakhir adalah 12 dari 15 tanda tangan dikenali atau 80%

### 4.3. Pembahasan

Setelah mendapatkan hasil dari seluruh percobaan 10 *fold-cross validation* maka untuk mendapatkan tingkat akurasi yang sebenarnya didapatkan dengan mencari rerata dari seluruh hasil yang didapat (tabel 4.33)

**Tabel 4.33 Hasil Tingkat Akurasi 10 *fold-cross validation***

Percobaan 1	86.67%
Percobaan 2	73.33%
Percobaan 3	86.67%
Percobaan 4	86.67%
Percobaan 5	80%
Percobaan 6	80%
Percobaan 7	73.33%
Percobaan 8	60%
Percobaan 9	80%
Percobaan 10	80%
Hasil	78.667% (79%)

Dari tabel terlihat bahwa percobaan ke-delapan memiliki tingkat akurasi terendah sedangkan tingkat akurasi tertinggi terdapat pada percobaan 1, 3, dan 4.

Secara keseluruhan terjadi 2 dimana *perceptron* salah mengenali tanda tangan palsu sebagai tanda tangan asli, terjadi juga 26 data dimana *perceptron* salah mengenali tanda tangan asli sebagai tanda tangan palsu, dan terjadi 5 data yang salah klasifikasi dimana tanda tangan seseorang dikenali sebagai tanda tangan orang lain.

#### 4.4. Aplikasi Hasil Penelitian

Seperti sudah dijelaskan pada bagian sebelumnya, pada penelitian ini dilakukan pengolahan data tanda tangan dari pejabat FT universitas negeri jakarta. Penelitian ini diharapkan dapat bermanfaat pada penelitian berikutnya. Aplikasi dari penelitian itu sendiri dapat berupa sebuah sistem yang dapat menjamin keaslian dari tanda tangan para pejabat universitas negeri jakarta.

Sistem yang dimaksud merupakan sebuah jaringan program yang dapat memeriksa keaslian sebuah tanda tangan hanya dari foto tanda tangan tersebut saja program ini juga tersambung ke *database* tempat dimana foto tanda tangan tersebut diproses dan apabila tanda tangan dinyatakan asli maka secara otomatis tanda tangan tersebut dimasukkan kedalam kumpulan data *train* untuk selanjutnya dilakukan proses *train* ulang hanya untuk pemilik tanda tangan tersebut dan kebalikannya jika tanda tangan dinyatakan palsu maka foto tersebut akan dimasukkan kedalam *blacklist* untuk selanjutnya ditindak lanjuti oleh *admin* atau pengurus *database*.

Penggunaan sistem seperti ini mungkin akan menjadi rawan karena apabila ada tanda tangan palsu yang dikenali sebagai tanda tangan asli maka hal ini akan berpotensi untuk merusak keseluruhan kumpulan data *train*, pilihan lain adalah tanda tangan yang telah dinyatakan sebagai tanda tangan asli ditampung untuk kemudian akan dimasukkan kedalam kumpulan data *train* setelah *admin* mengkonfirmasi keaslian dari tanda tangan tersebut.



## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Kesimpulan yang diperoleh setelah melalui tahap-tahap dari pengenalan tanda tangan menggunakan algoritma *Perceptron* adalah:

1. Pengenalan tanda tangan pejabat Fakultas Teknik Universitas Negeri Jakarta dapat dilakukan dengan menggunakan algoritma *Single-layer Perceptron*
2. Hasil pengenalan menggunakan algoritma *Perceptron* diukur tingkat akurasi dengan menggunakan metode *K fold-cross validation*, dengan hasil akurasi algoritma sebesar 78.667%
3. Aplikasi dari penelitian ini adalah untuk membuat sebuah sistem yang dapat mengidentifikasi keaslian sebuah tanda tangan untuk mencegah kemungkinan pemalsuan tanda tangan pada dokumen-dokumen penting

#### **5.2. Saran**

Beberapa masukan untuk penelitian berikutnya yang berhubungan dengan pengenalan tanda tangan adalah sebagai berikut:

1. Menambah jumlah data *train*, karena jumlah data *train* yang digunakan berbanding lurus dengan tingkat akurasi dari algoritma *Perceptron*.
2. Menggunakan algoritma lain seperti *Backpropagation*, untuk dapat membandingkan hasilnya agar dapat mengetahui algoritma yang lebih baik untuk digunakan untuk penelitian selanjutnya

3. Jika menggunakan media kertas dan alat tulis pastikan ketebalan tanda tangan agar dapat terbaca dengan baik pada proses *scanning*.
4. Mengimplementasikan penghilangan *noise* agar mendapat hasil yang lebih baik.
5. Menggunakan deteksi tepi atau *edge-detection* agar dapat mendapatkan ciri dari sebuah tanda tangan.

## DAFTAR PUSTAKA

- Siregar, N. (2012) Strategi Otentikasi Pesan Menggunakan Digital Signature Dengan Metode DSA (Digital Standard Algorithm)
- Abbas, R. (1994), *A Prototype System for off-line Signature Verification using Multilayered Feedforward Neural Networks*. Thesis Departement of Computer Science, RMIT
- Siang, JJ. (2009). Aplikasi Jaringan Syaraf Tiruan dan Pemrograman Menggunakan MATLAB. Yogyakarta: Andi.
- Siahaan, A. (2011) Pengenalan Karakter Dan Manajemen *Database* Pada Formulir Isian Menggunakan Jaringan Syaraf Tiruan Dengan Metode Learning Vector Quantization (LVQ)
- Kusumaningtyas, EM. & Badriyah, T. & Sigit, R. (2005) Modul Ajar Kecerdasan Buatan. Politeknik Elektronika Negeri Surabaya
- Munir, R. (2002) Diktat kuliah Pengolahan Citra, Edisi Kedua. Departemen Teknik Informatika ITB
- Putra, D. (2010) Pengolahan Citra Digital. Yogyakarta: Andi
- Hendradjaya, Bayu. (1995) Catatan Kuliah Pengolahan Citra.
- Hartarto, dkk. (2011) DATA TRANSFORMATION PADA DATA MINING. Prosiding Konferensi Nasional “Inovasi dalam Desain dan Teknologi” - IDeaTech 2011
- Godse AP. & Godse DA. (2007) *Digital Computer Fundamentals*. First Edition. Technical Publications pune
- Sunarto. (2008) Buku TIK SMA/MA Kelas XII. Grasindo
- Reddy, M. *BMP MS-Windows bitmap format*. Diakses 30 september 2016 (<http://www.martinreddy.net/gfx/2d/BMP.txt>)
- Poskanzer, J. (1991). *NetPBM User Manual* Diakses 30 september 2016 (<http://netpbm.sourceforge.net/doc/pbm.html>).
- Rostamian, R. (2014). *Programming Projects in C for Students of Engineering, Science, and Mathematics*. the society for industrial and applied mathematics
- Kohavi, Ron (1995). "A study of cross-validation and bootstrap for accuracy estimation and model selection". Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. San Mateo, CA: Morgan Kaufmann. 2 (12): 1137–1143

## **LAMPIRAN**

### Lampiran 1. Keseluruhan Tanda Tangan Sebelum Pra-proses



## Lampiran 2. Keseluruhan Tanda Tangan Setelah Pra-proses



### Lampiran 3. Barisan Kode Program Pra-proses dalam MATLAB

```

folder = 'C:\Users\Dimas\Documents\MATLAB\TTD\';
Files = dir(fullfile(folder, '*.bmp'));
for I = 1 : length(Files)
    FileName = fullfile(folder, Files(I).name);
    Im{I} = imread(FileName);
    Im{I} = imresize(Im{I},[512,512]);
    Im{I} = rgb2gray(Im{I});
    Im{I} = imsharpen(Im{I},'Threshold',0.5);
    Im{I} = im2double(Im{I});
    Im{I} = im2bw(Im{I},0.445);

    xstart=512;
    xend=1;
    ystart=512;
    yend=1;

    for y=1:512
        for x=1:512
            if((Im{I}(y,x)==0)) %(y,x)
                if (y<ystart)
                    ystart=y;
                end
                if((y>yend))
                    yend=y;
                end
                if (x<xstart)
                    xstart=x;
                end
                if (x>xend)
                    xend=x;
                end
            end
        end
    end

    for i=ystart:yend
        for j=xstart:xend
            im{I}((i-ystart+1),(j-xstart+1))=Im{I}(i,j);
        end
    end

    im{I} = imresize(im{I},[256,256]);

Folder = 'C:\Users\Dimas\Documents\MATLAB\contoh\';

```

```
if ~exist(Folder)
    mkdir(Folder);
end
Inputs = {Files.name}';
Outputs = Inputs;
    Outputs{I} = regexprep(Outputs{I}, '.bmp','.pbm');
    imwrite(im{I}, [Folder Outputs{I}])

end
```



#### Lampiran 4. Barisan Kode Program Perceptron Dalam MATLAB

```

folder = 'C:\Users\Dimas\Documents\MATLAB\K-Cross Validation\Q1\Train\';
Files = dir(fullfile(folder, '*.pbm'));
for I = 1 : length(Files)
    FileName = fullfile(folder, Files(I).name);
    Im{I} = imread(FileName);
    p{I} = Im{I}(1:end)';
end
P=cell2mat(p);

load('target 2 ganjil.mat');
net1 = perceptron;
net1 = train(net1, P, T1);

net2 = perceptron;
net2 = train(net2, P, T2);

net3 = perceptron;
net3 = train(net3, P, T3);

net4 = perceptron;
net4 = train(net4, P, T4);

net5 = perceptron;
net5 = train(net5, P, T5);

Folder = 'C:\Users\Dimas\Documents\MATLAB\K-Cross Validation\Q1\Test\';
files = dir(fullfile(Folder, '*.pbm'));
for Y = 1 : length(files);
    filename = fullfile(Folder, files(Y).name);
    test{Y} = imread(filename);
    t{Y} = sim(net1, test{Y}(1:end)');
end
R1=cell2mat(t);

for Y = 1 : length(files);
    filename = fullfile(Folder, files(Y).name);
    test{Y} = imread(filename);
    t{Y} = sim(net2, test{Y}(1:end)');
end
R2=cell2mat(t);

for Y = 1 : length(files);
    filename = fullfile(Folder, files(Y).name);
    test{Y} = imread(filename);
    t{Y} = sim(net3, test{Y}(1:end)');
end

```

```
R3=cell2mat(t);

for Y = 1 : length(files);
    filename = fullfile(Folder, files(Y).name);
    test{Y} = imread(filename);
    t{Y} = sim(net4, test{Y}(1:end));
end
R4=cell2mat(t);

for Y = 1 : length(files);
    filename = fullfile(Folder, files(Y).name);
    test{Y} = imread(filename);
    t{Y} = sim(net5, test{Y}(1:end));
end
R5=cell2mat(t);

TR = R1+R2+R3+R4+R5
```

## TENTANG PENULIS

Penulis bernama Dimas Agung Riansa, lahir di Jakarta pada 10 Desember 1994. merupakan anak kedua dari tiga bersaudara. Penulis memiliki kedua orang tua, seorang ibu yang bernama Viraza Rama Sari dan seorang bapak yang bernama Rianto Dahono. Penulis berkebangsaan Indonesia dan tinggal di Jl Tebet Dalam II No. 17

Penulis memulai pendidikan pada tahun 1999 di TK An-nizomiyah, kemudian pada tahun 2000 melanjutkan pendidikan di SDN 11 Pagi dan menyelesaikan pendidikan dasar pada tahun 2006. Lalu penulis melanjutkan pendidikan di SMPN 227 Jakarta dan menyelesaikannya di tahun 2009. Setelah itu, penulis melanjutkan sekolah di SMAN 55 Jakarta dan lulus pada tahun 2012. Di tahun yang sama, penulis memasuki bangku perkuliahan S1 di Universitas Negeri Jakarta pada program studi Pendidikan Teknik Informatika dan Komputer melalui jalur penmaba.