

**Pengenalan Aksara Incung Menggunakan  
Metode *HIDDEN MARKOV MODEL***

**SKRIPSI**

Diajukan untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
Program Studi Informatika



Oleh:

Agung Kristanto

175314052

**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS SANATA DHARMA  
YOGYAKARTA**

**2021**

# **RECOGNIZING THE INCUNG SCRIPT USING HIDDEN MARKOV MODEL METHOD**

## **THESIS**

Presented as Partial Fullfillment of the Requirements  
to Obtain *Sarjana Komputer* Degree  
in the Informatics Study Program



Created by:  
Agung Kristanto  
175314052

**INFORMATICS STUDY PROGRAM  
FACULTY OF SCIENCE AND TECHNOLOGY  
SANATA DHARMA UNIVERSITY  
YOGYAKARTA  
2021**

**HALAMAN PERSETUJUAN**  
**SKRIPSI**  
**PENGENALAN AKSARA INCUNG MENGGUNAKAN**  
**METODE *HIDDEN MARKOV MODEL***

Oleh:  
Agung Kristanto  
NIM: 175314052

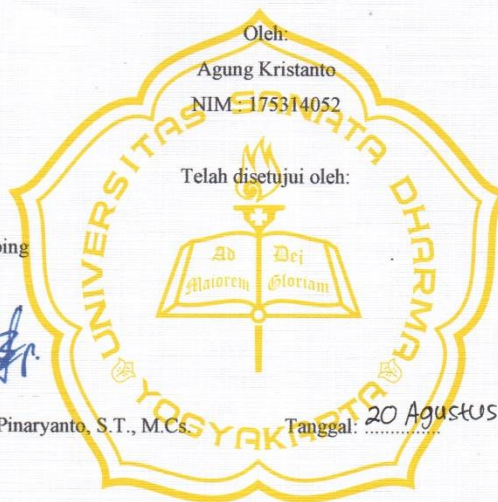
Telah disetujui oleh:

Pembimbing



Kartono Pinaryanto, S.T., M.Cs.

Tanggal: 20 Agustus 2021



**HALAMAN PENGESAHAN**  
**SKRIPSI**  
**Pengenalan Aksara Incung Menggunakan**  
**METODE *HIDDEN MARKOV MODEL***

Dipersiapkan dan disusun oleh:

Agung Kristanto

175314052

Telah dipertahankan di depan Tim Penguji

Pada tanggal: 27 Juli 2021

Dan dinyatakan memenuhi syarat

Susunan Tim Penguji:

Ketua

: Dr. Anastasia Rita Widiarti, M.Kom.

Sekretaris

: Rosalia Arum Kumalasanti, M.T.

Anggota

: Kartono Pinaryanto, S.T., M.Cs.

Tanda Tangan



Yogyakarta, 23 Agustus 2021

Fakultas Sains dan Teknologi

Universitas Sanata Dharma

Dekan



Sudi Mungkasi, S.Si., M.Math.Sc., Ph.D.

## **HALAMAN PERSEMBAHAN**

“Percayalah kepada TUHAN dengan segenap hatimu, dan janganlah bersandar kepada pengertianmu sendiri. Akuilah Dia dalam segala lakumu, maka Ia akan meluruskan jalanmu.”

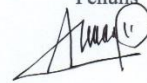
Amsal 3:5-6

#### **PERNYATAAN KEASLIAN KARYA**

Saya menyatakan dengan sesungguhnya bahwa di dalam skripsi yang saya tulis ini tidak memuat karya atau bagian karya orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka sebagaimana layaknya karya ilmiah.

Yogyakarta, 18 Agustus 2021

Penulis



Agung Kristanto



#### **LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH**

Yang bertanda tangan dibawah ini, saya mahasiswa Universitas Sanata Dharma:

Nama : Agung Kristanto

NIM : 175314052

Demi pengembangan ilmu pengetahuan, saya memberikan kepada perpustakaan Universitas Sanata Dharma karya ilmiah yang berjudul:

#### **PENGENALAN AKSARA INCUNG MENGGUNAKAN METODE *HIDDEN MARKOV MODEL***

Dengan demikian saya memberikan kepada perpustakaan Universitas Sanata Dharma hak untuk menyimpan, mengalihkan dalam bentuk media lain, mengelolanya dalam bentuk pangkalan data mendistribusikan secara terbatas, dan mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya maupun memberikan royalti kepada saya selama tetap mencantumkan nama saya sebagai penulis.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Yogyakarta,

Pada tanggal: 18 Agustus 2021

Yang menyatakan,



Agung Kristanto

## ABSTRAK

Aksara Incung sebagai kebudayaan berupa tulisan aksara kuno yang berasal dari daerah Kerinci, Jambi. Dalam upaya untuk melestarikan aksara Incung supaya tidak punah, maka diperlukan sebuah sistem untuk mengenali aksara Incung. Ini dapat dilakukan dengan menggunakan metode *Hidden Markov Model* untuk proses pengenalan pada aksara Incung. Penulis menggunakan konsonan aksara Incung yang disusun oleh L.C Westenek.

Setiap citra aksara yang sudah dilakukan proses *preprocessing* kemudian dilakukan proses ekstraksi ciri dengan metode *Intensity of Character* dan *Mark Direction*. Dari ciri tersebut digunakan untuk input pada metode *Hidden Markov Model* (HMM). Sehingga dari proses pengenalan ini diperoleh akurasi terbaik sebesar 89.3% dengan menggunakan ekstraksi ciri *Intensity of Character* berukuran 5x5 dengan nilai *K-Fold* 5.

**Kata kunci :** Aksara Incung, *Intensity of Character*, *Mark Direction*, *Hidden Markov Model* (HMM)



## ABSTRACT

Incung script as a culture takes the form of ancient script from Kerinci, Jambi. In an effort to preserve the Incung script so that it does not become extinct, a system is needed to recognize the Incung script. This can be done using the Hidden Markov Model method for the recognition process of the Incung script. The author uses incung consonants compiled by L.C Westenenk.

Each character *image* that has been through the preprocessing process then goes through the process of extracting features with the Intensity of Character and *Mark Direction* method. The output of this process is used for input on the Hidden Markov Model (HMM) method. So from this recognition process obtained the best accuracy of 89.3% by using the extraction of Intensity of Character size 5x5 the value of *K-Fold* 5.

**Keywords :** The Incung Script, *Intensity of Character*, *Mark Direction*, *Hidden Markov Model* (HMM)

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, atas segala kebaikan dan kasih setia-Nya, sehingga penulis dapat menyelesaikan tugas akhir dengan judul “Pengenalan Aksara Incung Menggunakan Metode *Hidden Markov Model*”.

Dalam penyelesaian tugas akhir ini tentunya tidak lepas dari bantuan berbagai pihak yang sudah mendukung baik secara langsung maupun tidak langsung. Oleh karena itu, dengan tulus hati penulis mengucapkan terima kasih kepada:

1. Bapak Kartono Pinaryanto, S.T., M.Cs. selaku dosen pembimbing yang telah bersedia memberikan waktu, arahan, motivasi, masukan serta bimbingan kepada penulis selama menyelesaikan tugas akhir ini.
2. Seluruh dosen Informatika Universitas Sanata Dharma yang telah mendidik dan memberikan ilmu pengetahuan kepada penulis yang digunakan sebagai bekal dalam menyelesaikan tugas akhir ini.
3. Keluarga tercinta Bapak Heru Prayitno, Ibu Kristi Handayani dan Adik Dwi Handoko yang selalu mendukung serta selalu mendoakan penulis sehingga penulis dapat menyelesaikan tugas akhir ini.
4. Saudari Geovani Putri Wiradi yang selalu memberikan dukungan dalam doa, memberikan perhatian, semangat dan motivasi kepada penulis dalam penyelesaian tugas akhir ini.
5. Saudara Joseph Hutagalung, Edrick Hernando, Atanasius Ivannoel Rio Aji, Gabriel Ryan Prima, dan Saudari Fransiska Apri Wulandari yang telah bersedia meluangkan waktunya untuk berdiskusi mengenai permasalahan dalam pengerjaan tugas akhir ini.
6. Semua pihak yang tidak bisa disebutkan satu persatu yang telah membantu atau terlibat dalam penulisan tugas akhir ini.

Penulis menyadari bahwa masih banyak kekurangan dari tugas akhir ini, sehingga penulis mengharapkan kritik dan saran yang bersifat membangun untuk

menyempurnakannya. Semoga tugas akhir ini dapat bermanfaat bagi para pembaca, terutama bagi teman-teman Informatika.

Yogyakarta, 18 Agustus 2021

Penulis

A handwritten signature in black ink, featuring a stylized 'A' and 'K' with a circular flourish at the end.

Agung Kristanto

## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN.....	iii
HALAMAN PERSEMBAHAN .....	v
HALAMAN PERNYATAAN .....	vii
ABSTRAK .....	viii
ABSTRACT.....	ix
KATA PENGANTAR .....	x
DAFTAR ISI.....	xii
DAFTAR ISI TABEL .....	xv
DAFTAR ISI GAMBAR .....	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian.....	2
1.5 Batasan Penelitian .....	3
1.6 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI .....	5
2.1 Aksara Incung.....	5
2.2 Pemrosesan Citra.....	6
2.2.1 Binerisasi.....	7
2.2.2 Reduksi Derau.....	7
2.2.3 Segmentasi .....	8
2.2.4 <i>Resize</i> .....	8
2.2.5 Penipisan .....	8
2.3 Pengenalan Pola .....	9
2.5 Ekstraksi Ciri .....	9
2.5.1 <i>Intensity of Character</i> .....	10
2.5.2 <i>Mark Direction</i> .....	11

2.6 Metode <i>Hidden Markov Model</i> .....	12
2.6.1 Algoritma <i>Hidden Markov Model</i> .....	15
2.6.2 Tahap Pelatihan HMM.....	18
2.6.3 Tahap Pengujian HMM.....	26
2.7 Metode Evaluasi <i>K-Fold Cross Validation</i> .....	29
BAB III METODE PENELITIAN .....	30
3.1 Data .....	30
3.2 Perancangan Sistem.....	31
3.2.1 Akusisi Data.....	31
3.2.2 <i>Cropping</i> .....	32
3.2.3 <i>Preprocessing</i> .....	32
3.2.4 Ekstraksi Ciri .....	34
3.2.2 Tahap Pelatihan HMM.....	37
3.2.3 Tahap Pengujian HMM.....	39
3.2.5 Skenario Pengujian .....	41
3.3 Kebutuhan Perangkat Lunak .....	42
3.4 Kebutuhan Perangkat Keras .....	42
3.5 Desain GUI.....	43
BAB IV HASIL DAN ANALISIS.....	44
4.1 Data .....	44
4.1.1 Akusisi Data.....	44
4.1.2 <i>Cropping</i> .....	45
4.2 Implementasi <i>Preprocessing</i> .....	45
4.2.1 Binerisasi Citra.....	45
4.2.2 Reduksi Derau.....	46
4.2.3 Segmentasi .....	47
4.2.4 <i>Resize</i> Citra .....	48

4.2.5 Penipisan Citra .....	49
4.3 Implementasi Ekstraksi Ciri .....	49
4.3.1 Data Ciri 1 .....	51
4.3.2 Data Ciri 2 .....	51
4.3.3 Data Ciri 3 .....	52
4.3.4 Data Ciri 4 .....	52
4.3.4 Data Ciri 5 .....	52
4.3.4 Data Ciri 6 .....	53
4.4 Implementasi Pembagian Data Pelatihan dan Pengujian .....	53
4.5 Implementasi Klasifikasi <i>Hidden Markov Model</i> .....	54
4.5.1 Inisialisasi parameter <i>Hidden Markov Model</i> .....	54
4.5.2 Tahap Pelatihan <i>Hidden Markov Model</i> .....	55
4.5.3 Tahap Pengujian <i>Hidden Markov Model</i> .....	55
4.6 Pengujian dan Evaluasi .....	56
4.6.1 Pengujian Data Ciri 1 .....	56
4.6.2 Pengujian Data Ciri 2 .....	59
4.6.3 Pengujian Data Ciri 3 .....	62
4.6.4 Pengujian Data Ciri 4 .....	66
4.6.5 Pengujian Data Ciri 5 .....	70
4.6.6 Pengujian Data Ciri 6 .....	74
4.6.7 Pengujian Data Tunggal .....	79
BAB V PENUTUP .....	86
5.1 Kesimpulan .....	86
5.2 Saran .....	86
DAFTAR PUSTAKA .....	87
LAMPIRAN .....	90

## DAFTAR ISI TABEL

Tabel 2.1 Representasi Data Citra Ukuran 9x9.....	10
Tabel 2.2 Hasil Data Citra IoC 3x3.....	10
Tabel 2.3 Tabel Representasi <i>Mark Direction</i> (Vertikal) .....	11
Tabel 2.4 Tabel Representasi <i>Mark Direction</i> (Horizontal) .....	11
Tabel 2.5 Tabel Representasi <i>Mark Direction</i> (Diagonal Kiri) .....	11
Tabel 2.6 Tabel Representasi <i>Mark Direction</i> (Diagonal Kanan) .....	11
Tabel 2.7 Fungsi <i>toolbox</i> Inisialisasi HMM.....	13
Tabel 2.8 Fungsi <i>toolbox</i> Pelatihan HMM.....	14
Tabel 2.9 Fungsi <i>toolbox</i> Pengujian HMM.....	15
Tabel 2.10 Probabilitas Membawa Payung.....	17
Tabel 2.11 Estimasi Probabilitas.....	23
Tabel 3.1 Skenario Pengujian .....	42
Tabel 4.1 Kelas pada Aksara Incung.....	49
Tabel 4.2 Data Ciri 1 .....	51
Tabel 4.3 Data Ciri 2 .....	51
Tabel 4.4 Data Ciri 3 .....	52
Tabel 4.5 Data Ciri 4 .....	52
Tabel 4.6 Data Ciri 5 .....	52
Tabel 4.7 Data Ciri 6 .....	53
Tabel 4.8 Pembagian Data Latih 3 <i>fold</i> .....	53
Tabel 4.9 Pembagian Data Latih 3 <i>fold</i> .....	53
Tabel 4.10 Pembagian Data Latih 5 <i>fold</i> .....	54
Tabel 4.11 Pembagian Data Uji 5 <i>fold</i> .....	54
Tabel 4.12 Hasil Pengujian Data Ciri 1 dengan 3 <i>fold</i> .....	57
Tabel 4.13 Hasil Pengujian Data Ciri 1 dengan 5 <i>fold</i> .....	58
Tabel 4.14 Hasil Pengujian Data Ciri 2 dengan 3 <i>fold</i> .....	60
Tabel 4.15 Hasil Pengujian Data Ciri 2 dengan 5 <i>fold</i> .....	61
Tabel 4.16 Hasil Pengujian Data Ciri 3 dengan 3 <i>fold</i> .....	63
Tabel 4.17 Hasil Pengujian Data Ciri 3 dengan 5 <i>fold</i> .....	65
Tabel 4.18 Hasil Pengujian Data Ciri 4 dengan 3 <i>fold</i> .....	67



Tabel 4.19 Hasil Pengujian Data Ciri 4 dengan 5 <i>fold</i> .....	69
Tabel 4.20 Hasil Pengujian Data Ciri 5 dengan 3 <i>fold</i> .....	71
Tabel 4.21 Hasil Pengujian Data Ciri 5 dengan 5 <i>fold</i> .....	73
Tabel 4.22 Hasil Pengujian Data Ciri 6 dengan 3 <i>fold</i> .....	75
Tabel 4.23 Hasil Pengujian Data Ciri 6 dengan 5 <i>fold</i> .....	77
Tabel 4.24 Durasi Klasifikasi HMM.....	78
Tabel 4.25 Hasil Pengujian Data Tunggal .....	79
Tabel 4.26 Hasil Persentase Rata-rata Akurasi Keseluruhan.....	84

## DAFTAR ISI GAMBAR

Gambar 2.1 Konsonan Surat Incung yang disusun oleh L.C Westenenk .....	6
Gambar 2.2 Struktur elemen tanda arah (Surinta, 2010) .....	11
Gambar 2.3 Representasi parameter HMM .....	15
Gambar 2.4 Ilustrasi dari operasi pada Baum-Welch .....	20
Gambar 2.5 Ilustrasi pencarian nilai terbaik dengan <i>Viterbi</i> .....	27
Gambar 2.6 Ilustrasi Model 3-Fold Cross Validation .....	29
Gambar 3.1 Angket Aksara Incung.....	30
Gambar 3.2 Diagram Perancangan Sistem.....	31
Gambar 3.3 Diagram Tahap Pelatihan .....	37
Gambar 3.4 Diagram Tahap Pengujian.....	39
Gambar 4.1 Contoh Angket Aksara Incung.....	44
Gambar 4.2 Data Citra Hasil <i>Cropping</i> .....	45
Gambar 4.3 Proses Binerisasi Citra Aksara Da .....	45
Gambar 4.4 Proses <i>Invers</i> Citra Aksara Da .....	46
Gambar 4.5 Proses Menghilangkan Derau Aksara Da.....	46
Gambar 4. 6 Hasil Citra Proses <i>Imfill</i> Citra Aksara Da .....	47
Gambar 4.7 Proses <i>Invers</i> Citra Aksara Da .....	47
Gambar 4.8 Proses Profil Proyeksi Citra .....	48
Gambar 4.9 <i>Resize</i> Citra.....	48
Gambar 4.10 Penipisan Citra .....	49
Gambar 4.11 Grafik Akurasi dengan IoC = 3 dan <i>fold</i> 3 .....	56
Gambar 4.12 Grafik Akurasi dengan IoC = 3 dan <i>fold</i> 5 .....	58
Gambar 4.13 Grafik Akurasi dengan IoC = 4 dan <i>fold</i> 3 .....	59
Gambar 4.14 Grafik Akurasi dengan IoC = 4 dan <i>fold</i> 5 .....	60
Gambar 4.15 Grafik Akurasi dengan IoC = 5 dan <i>fold</i> 3 .....	62
Gambar 4.16 Grafik Akurasi dengan IoC=5 dan <i>fold</i> 5 .....	64
Gambar 4.17 Grafik Akurasi dengan <i>Mark Direction</i> Horizontal dan <i>Fold</i> 3.....	66
Gambar 4.18 Grafik Akurasi dengan <i>Mark Direction</i> Horizontal dan <i>Fold</i> 5.....	68
Gambar 4.19 Grafik Akurasi dengan <i>Mark Direction</i> Vertikal dan <i>Fold</i> 3.....	70
Gambar 4.20 Grafik Akurasi dengan <i>Mark Direction</i> Vertikal dan <i>Fold</i> 5.....	72

Gambar 4.21 Grafik Akurasi dengan <i>Mark Direction</i> dan <i>fold 3</i> .....	74
Gambar 4.22 Grafik Akurasi dengan <i>Mark Direction</i> dan <i>fold 5</i> .....	76
Gambar 4.23 Tampilan Menu Pengujian Data Tunggal .....	78

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Sebagai negara yang luas dan terdiri dalam kepulauan Indonesia memiliki warisan kekayaan budaya yang beragam, salah satu kebudayaan itu terwujud dalam bentuk aksara. Begitu banyak kebudayaan aksara di Indonesia salah satunya aksara Incung. Aksara Incung berasal dari kebudayaan masyarakat di daerah Kerinci, Jambi. Keberadaan aksara Incung telah digunakan secara luas pada abad ke 14 M (Powa, 2020).

Melestarikan kebudayaan aksara Incung ini sebagai upaya untuk memperkaya kebudayaan daerah Kerinci maupun untuk bangsa Indonesia (Utnasari, 2018). Pada penelitian ini diharapkan mampu menarik minat generasi muda untuk mempelajari aksara Incung yang merupakan kebudayaan daerah Kerinci, Jambi sebagai kontribusi dalam pelestarian budaya bangsa Indonesia. Hal ini akan membawa pengaruh yang baik bagi orang-orang untuk mengenal kebudayaan Indonesia. Sehingga warisan leluhur dapat terjaga dan lestari.

Perkembangan teknologi modern yang semakin pesat membuat permasalahan mampu dipecahkan melalui komputasi menggunakan komputer (Widyastuti, 2016). Salah satu model komputasi *machine learning* seperti *Hidden Markov Model* yang memberikan memanfaatkan kemampuan komputer dalam menangani masalah-masalah yang kompleks di antaranya adalah pengenalan pola, pengenalan wajah, pencitraan medis, prediksi saham dan lain sebagainya (Ganeshamoorthy, 2008).

Beberapa peneliti telah melakukan penelitian menggunakan metode *Hidden Markov Model*, para peneliti tersebut mendapatkan hasil akurasi yang tinggi. Berdasarkan penelitian yang dilakukan oleh Metwally, dkk, menggunakan metode HMM untuk pengenalan tulisan tangan Arab mendapatkan akurasi 87% (Metwally et al., 2017). Pada penelitian Pratiwi, dkk, melakukan pengenalan wajah menggunakan HMM mencapai akurasi 95.9% (Pratiwi et al., 2018). Lokhande telah menggunakan Hidden Markov Model untuk pengenalan tulisan tangan online yang menghasilkan 80% efisiensi (Lokhande, 2017).

Maka dengan hal ini, penulis ingin menguji tingkat keberhasilan pengenalan aksara Incung menggunakan metode *Hidden Markov Model*. Sehingga metode *Hidden Markov Model* akan diharapkan memberikan keberhasilan pada pengenalan aksara Incung.

### **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah dibuat, maka dapat dirumuskan masalah dalam penelitian ini sebagai berikut:

1. Bagaimana penerapan metode *Hidden Markov Model* pada pengenalan aksara Incung?
2. Berapa akurasi yang diperoleh dari penerapan metode *Hidden Markov Model* untuk pengenalan aksara Incung?

### **1.3 Tujuan Penelitian**

Tujuan dari penelitian ini adalah sebagai berikut:

1. Menerapkan metode *Hidden Markov Model* untuk pengenalan aksara Incung.
2. Mengetahui tingkat akurasi dalam mengenali aksara Incung menggunakan metode *Hidden Markov Model*.

### **1.4 Manfaat Penelitian**

Manfaat penelitian adalah sebagai berikut:

1. Membantu orang-orang untuk lebih mudah mengenali dan belajar mengenai aksara Incung sebagai kebudayaan dari Kerinci, Jambi.
2. Memberikan referensi bagi penelitian yang berkaitan pada pengenalan aksara Incung menggunakan metode *Hidden Markov Model*.

### 1.5 Batasan Penelitian

Batasan – batasan masalah dari penelitian ini adalah :

1. Aksara yang digunakan adalah konsonan aksara Incung yang disusun oleh L.C Westenenk.
2. Pada pengumpulan data aksara Incung menggunakan ukuran kertas A4 dengan ukuran pena 0.8 dan posisi responden duduk.
3. Data diperoleh dari 90 responden yang menuliskan aksara Incung kemudian hasil penulisan tersebut dipindai yang menghasilkan citra dalam format jpg.
4. Hasil citra aksara yang telah dipindai dilakukan pemotongan setiap aksara secara manual.
5. Ukuran citra aksara Incung yang digunakan berukuran 60x60 piksel.
6. Ekstraksi ciri menggunakan *Intensity of Character* dengan ukuran 3x3, 4x4, dan 5x5.
7. Ekstraksi ciri yang digunakan *Mark Direction* dengan ukuran 3x3.
8. Menggunakan validasi *K-Fold Cross Validation* dengan nilai k-3 dan 5.

## **1.6 Sistematika Penulisan**

### **BAB 1: PENDAHULUAN**

Bab ini berisi tentang latar belakang masalah, rumusan masalah, tujuan, manfaat, batasan masalah serta sistematika penulisan dari penelitian yang akan dilakukan

### **BAB 2: LANDASAN TEORI**

Bab ini berisi tentang teori-teori dasar yang berkaitan dengan penelitian yang akan dilakukan, meliputi jenis data citra yang digunakan, metode *preprocessing*, ekstraksi ciri, dan metode untuk klasifikasi.

### **BAB 3: METODOLOGI PENELITIAN**

Bab ini berisi tentang langkah-langkah yang dilakukan dalam penelitian yang bertujuan untuk menjawab dan menyelesaikan rumusan masalah yang dimiliki.

### **BAB 4: HASIL PENELITIAN DAN ANALISIS**

Bab ini berisi tentang penjelasan mengenai sistem yang akan dibangun, penerapan algoritma serta rancangan yang telah dibuat, cara penggunaan sistem, serta hasil analisa dari pengujian-pengujian yang dilakukan.

### **BAB 5: PENUTUP**

Bab ini berisi kesimpulan dari hasil penelitian serta saran yang diusulkan untuk dapat mengembangkan penelitian.



## BAB II

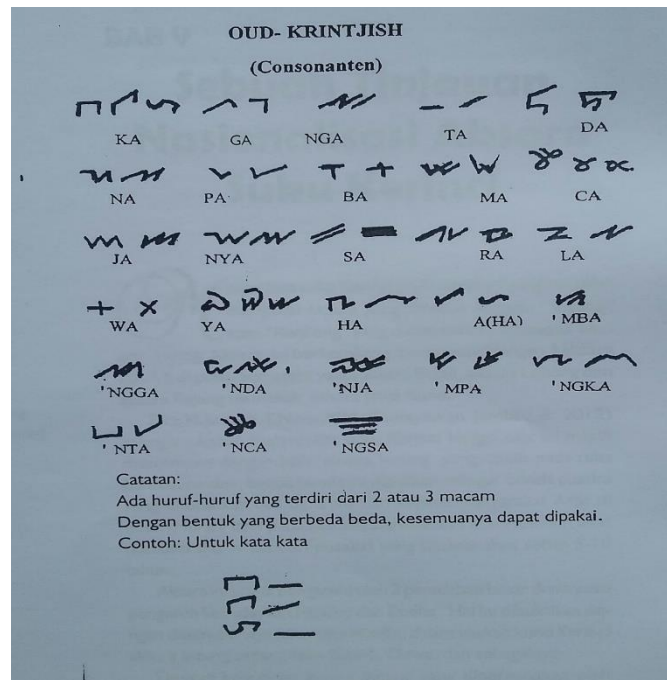
### LANDASAN TEORI

#### 2.1 Aksara Incung

Surat Incung atau biasa disebut aksara Incung adalah salah satu aksara yang ada di Indonesia. Aksara Incung sebagai bentuk keragaman yang dimiliki oleh suku Kerinci. Aksara ini merupakan bagian dari surat Ulu yang banyak digunakan di Sumatera Selatan. Kozok melakukan pengelompokan terhadap surat Ulu menjadi tiga sub-kelompok yaitu surat Incung yang digunakan di wilayah Kerinci, surat Rencong di Bengkulu dan Sumatera Selatan termasuk Komeriing, Lebong, Lembak, Lintang, Ogan, Pasemah, Rejang, dan Serawai, dan Surat Lampung yang digunakan di wilayah Lampung (Kozok, 2006).

Naskah-naskah kuno aksara Incung berisi dua hal yaitu *tembo* dan *karang mindu* (Alimin dkk, 2003). *Tembo* merupakan histografi tradisional yang berisi perjalanan silsilah nenek moyang atau klan (bahasa lokal: *kelebu*, *luhah*) masyarakat Kerinci sebagai pemilik naskah. Banyak naskah *tembo* beraksara Incung dituliskan pada media tanduk kerbau dan tanduk kambing. Sedangkan *karang mindu* merupakan prosa berisi ratapan kesedihan dari pembuat naskah. Pada naskah karang beraksara Incung kebanyakan ditulis pada media bambu dan kertas lama.

Surat Incung merupakan bagian dari kelompok surat Ulu yang berkembang di Sumatera Selatan. Westenenk mengidentifikasi sebanyak 28 karakter huruf dalam Surat Incung yang memiliki beberapa varian. Keberadaan aksara Incung pertama kali dilaporkan oleh William Marsden di tahun 1834 M (Voorhoeve, 1970).



Gambar 2.1 Konsonan Surat Incung yang disusun oleh L.C Westenenk

Setiap konsonan dalam aksara Incung terdiri atas sebuah konsonan yang diikuti oleh vokal 'a' seperti: Setiap karakter huruf dapat diubah menggunakan sandangan. Sandangan tersebut merubah vokal a menjadi i dan u selain itu terdapat pula sandangan untuk menambah bunyi dan menghilangkan vokal a sebagai contoh konsonan. Namun untuk naskah surat Incung kuno tidak terdapat kata sandangan untuk mengubah vokal a menjadi e atau o.

## 2.2 Pemrosesan Citra

Citra merupakan suatu kumpulan titik yang diberi nama piksel (Kadir & Susanto, 2013). Titik- titik ini yang mengandung informasi untuk menyusun suatu citra, sehingga dapat memperoleh informasi dari citra tersebut. Citra digital dapat diartikan sebagai citra  $f(x, y)$  maupun nilai intensitas cahaya pada citra. Citra digital digambarkan sebagai matrik yang terdiri dari baris dan kolom, pada setiap baris dan kolom akan berisikan elemen atau nilai intensitas kecerahan tertentu untuk setiap  $f(x, y)$  atau piksel (titik) pada bidang citra digital (Widiarti & Himamunanto, 2013).

Pemrosesan citra merupakan pengolahan gambar menggunakan komputer supaya kualitas pada suatu citra menjadi lebih baik. Pada pengolahan atau pemrosesan citra dilakukan pada setiap  $(x, y)$  untuk mendapatkan hasil yang diinginkan. Pada dasarnya metode pemrosesan citra ini digunakan untuk memodifikasi setiap piksel atau titik dari bidang citra sesuai dengan kebutuhannya.

### **2.2.1 Binerisasi**

Binerisasi merupakan tahap awal di dalam pengolahan tingkat piksel yaitu proses memisahkan objek dari latar belakang dengan cara menentukan nilai ambang *threshold* (Widiarti & Himamunanto, 2013). Pada citra biner adalah citra yang pikselnya memiliki kedalaman bit sebesar 1 bit sehingga hanya memiliki dua nilai intensitas warna yaitu 0 (hitam) dan 1 (putih). Binerisasi mengubah citra dengan dimensi dua menjadi citra hitam putih menjadi dimensi 1 dengan mengubah nilai elemen dalam matriks citra menjadi 1 dan 0 yang mana nilai 0 sebagai warna hitam dan nilai 1 sebagai warna putih.

### **2.2.2 Reduksi Derau**

Pada citra aksara yang terdapat bintik-bintik, itu terjadi sebagai kotoran-kotoran pada citra aksara. Setiap gangguan tersebut disebut dengan *noise* atau derau. Dengan ini membutuhkan proses dalam menghilangkan derau tersebut agar citra lebih baik.

Reduksi derau sebagai tahapan penghilangan *noise* dari sebuah citra digital. Proses yang dapat dilakukan untuk mengurangi kesalahan pada pengenalan objek dengan mengurangi derau atau *noise*. Ini akan menghilangkan piksel-piksel yang tidak dibutuhkan (Widiarti & Himamunanto, 2013). Tujuan utama pada reduksi derau untuk mengurangi sebanyak mungkin derau sehingga diperoleh citra yang lebih baik.

Beberapa jenis *noise* yang mungkin dapat terjadi antar lain:

- a. *Gaussian noise* : *noise* berupa titik-titik yang berwarna yang memiliki jumlah sama dengan presentase *noise*.
- b. *Speckle noise* : *noise* berupa warna hitam pada titik yang terkena *noise*.
- c. *Salt and pepper noise* : *noise* yang berupa warna putih pada titik yang terkena *noise*.

### 2.2.3 Segmentasi

Segmentasi merupakan teknik untuk membagi suatu citra menjadi beberapa daerah di mana setiap daerah memiliki kemiripan atribut. Pada tahap segmentasi ini untuk mendapatkan huruf tersebut secara individu dengan mendapatkan kolom-kolom karakter citra aksara dari baris yang telah dipotong. Tahap ini menggunakan profil proyeksi yang berfungsi memisahkan latar belakang dengan dipotong secara vertikal dan horizontal. Pada kolom citra diambil menggunakan proyeksi vertikal, sedangkan baris citra diambil menggunakan proyeksi horizontal.

### 2.2.4 Resize

*Resize* berfungsi untuk membuat setiap citra dapat diproses dengan ukuran yang sama. Melakukan hal ini yang bertujuan untuk meningkatkan nilai akurasi. Ukuran yang akan digunakan pada citra aksara Incung adalah 60x60 piksel.

### 2.2.5 Penipisan

Penipisan atau disebut *thinning* merupakan suatu operasi dalam proses *preprocessing*, yang berfungsi mereduksi suatu objek menjadi garis tengah yang menjadi kerangka dari objek tersebut (Widiarti & Himamunanto, 2013). Tujuan utama dari penipisan adalah membuat sederhana suatu objek menjadi suatu kerangka objek. Sehingga proses analisa pada objek menjadi lebih sederhana dan tidak mengambil memori terlalu besar.

### 2.3 Pengenalan Pola

Pengenalan pola merupakan disiplin ilmu yang bertujuan untuk klasifikasi suatu objek dalam beberapa kategori atau kelas (Theodoridis & Koutroumbas, 2008). Pola adalah suatu entitas yang terdefinisi dan dapat didefinisikan serta diberi nama. Pengertian dari pola itu sendiri adalah sekumpulan hasil atau pemantauan yang dapat dinyatakan ke dalam bentuk notasi vektor atau matriks (Putra, 2010).

Pengenalan pola dapat dilakukan sebagai tindakan dalam mengelola data mentah yang membuat suatu aksi berdasarkan pada kategori dari pola data tersebut. Pada pengenalan pola terdiri dari 3 langkah utama yaitu *preprocessing*, ekstraksi ciri, dan klasifikasi. *Preprocessing* sebagai langkah untuk memfokuskan objek data yang akan digunakan dan menghilangkan bagian yang tidak diperlukan dari objek data. Dalam hal pemrosesan awal yang dilakukan terhadap objek untuk mengubah citra digital menjadi citra biner. Ekstraksi ciri sebagai cara dalam menyederhanakan data dengan melakukan pengukuran dari ciri objek tertentu untuk mendapatkan informasi dasar dari data tersebut. Proses selanjutnya, klasifikasi yaitu suatu tindakan untuk pengelompokan data menjadi pola sesuai target atau kelas. Dalam proses klasifikasi dipilih ekstraksi ciri yang paling optimal untuk menghasilkan hasil terbaik pada tahap pengenalan aksara.

### 2.5 Ekstraksi Ciri

Ekstraksi ciri sebagai proses dalam pengambilan informasi atau ciri yang penting dari suatu citra sebagai pembeda dari suatu citra yang lain. Hasil ekstraksi ciri ini yang akan digunakan untuk membuat model pada *Hidden Markov Model*. Kemudian pada ciri diekstraksi untuk diproses dalam perhitungan saat pelatihan maupun pengujian. Pada langkah ini data yang diolah berupa aksara biner dengan ukuran tertentu yang telah dikenakan proses *preprocessing*.

### 2.5.1 *Intensity of Character*

*Intensity of Character* merupakan salah satu metode yang digunakan untuk melakukan ekstraksi ciri. *Intensity of Character* dalam bentuk matriks dengan ukuran  $N \times N$ , dimana setiap unit  $1 \times 1$  berisikan piksel hasil penjumlahan piksel yang memiliki nilai 0 (hitam). Proses ini data citra akan dibagi menjadi beberapa segmen yang lebih kecil sehingga lebih mudah dalam proses pencirian data. Sebagai contoh berikut merupakan proses gambaran untuk mendapatkan ciri IoC berukuran  $3 \times 3$ :

- a. Terdapat citra berukuran sebesar  $9 \times 9$ , maka untuk setiap unit IoC  $1 \times 1$  nya akan mencakup matriks sebesar  $3 \times 3$ .

Tabel 2.1 Representasi Data Citra Ukuran  $9 \times 9$

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

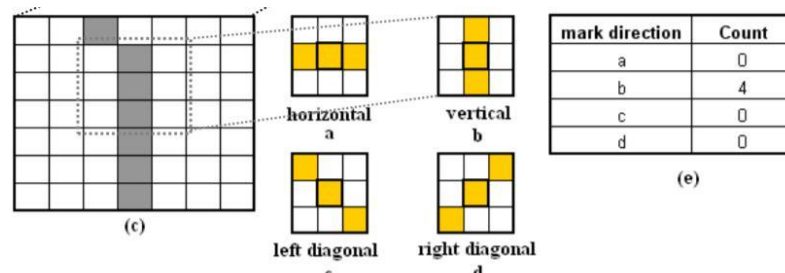
- b. Berikut bagian dari unit  $1 \times 1$  yang menyimpan nilai yang merupakan banyaknya piksel yang berwarna hitam atau bernilai 0 untuk setiap area  $3 \times 3$ . Berikut pada tabel 2.2 sebagai bentuk IoC  $3 \times 3$  yang telah dilakukan penjumlahan yang pada setiap bagian area  $3 \times 3$  yang memiliki nilai 0.

Tabel 2.2 Hasil Data Citra IoC  $3 \times 3$

3	1	4
3	6	1
1	2	3

### 2.5.2 Mark Direction

Ekstraksi ciri *Mark Direction* adalah ekstraksi ciri dari tanda arah horizontal, vertikal, diagonal ke kanan, dan diagonal ke kiri.



Gambar 2.2 Struktur elemen tanda arah (Surinta, 2010)

1. Dalam menentukan nilai arah vertikal yang memenuhi

Tabel 2.3 Tabel Representasi *Mark Direction* (Vertikal)

1	0	1
1	0	1
1	0	1

Tabel 2.4 Tabel Representasi *Mark Direction* (Horizontal)

1	1	1
0	0	0
1	1	1

Tabel 2.5 Tabel Representasi *Mark Direction* (Diagonal Kiri)

1	1	0
1	0	1
0	1	1

Tabel 2.6 Tabel Representasi *Mark Direction* (Diagonal Kanan)

0	1	1
1	0	1
1	1	0



## 2.6 Metode *Hidden Markov Model*

Metode klasifikasi pada pengenalan pola memiliki banyak jenis yang menghasilkan akurasi yang baik. Salah satu metode klasifikasi paling populer yang banyak digunakan yaitu metode *Hidden Markov Model* pada pengenalan pola. Dalam *Markov Model* dapat diasumsikan setiap keadaan dapat terlihat langsung oleh pengamat. Oleh karena itu, kemungkinan dari transisi antara kondisi menjadi satu-satunya parameter teramati. Dalam HMM itu suatu keadaan tidak terlihat secara langsung. Tetapi *output* yang bergantung terhadap keadaan tersebut terlihat.

*Hidden Markov Model* adalah model statistika dari suatu sistem yang diasumsikan dari rantai *Markov* dengan parameter yang tidak diketahui, dan memiliki tantangan dalam menentukan parameter-parameter tersembunyi (*hidden*) dari parameter-parameter yang dapat diamati (Przytycka, 2000). Pada parameter yang sudah ditentukan kemudian digunakan untuk analisis pada pengenalan pola.

Penulis menggunakan *toolbox* HMM dari Kevin Murphy pada tahap pembuatan sistem pengenalan aksara Incung menggunakan metode *Hidden Markov Model*. HMM *toolbox* mempunyai 3 fungsi utama yang digunakan untuk:

1. Estimasi parameter *likelihood* maksimum menggunakan algoritma *Baum-Welch*.
2. Klasifikasi deret, digunakan untuk mengevaluasi *log-likelihood* dari model yang sudah dilatih dengan data pengujian.
3. Menghitung deret dengan probabilitas tertinggi menggunakan algoritma *Viterbi*.

Beberapa hal dalam penggunaan HMM *toolbox* untuk inisialisasi HMM, pelatihan HMM, dan tahap pengujian.

### 1. Inisialisasi HMM

Sebelum melakukan tahap pelatihan dilakukan inisialisasi parameter pada HMM. Fungsi yang digunakan pada *toolbox* HMM.

Tabel 2.7 Fungsi *toolbox* Inisialisasi HMM

Fungsi pada <i>toolbox</i> HMM	<i>Output</i>	Penjelasan
mk_stochastic(rand(Q,Q))	Inisialisasi parameter distribusi peluang transisi keadaan (A)	Fungsi ini digunakan untuk menentukan parameter A dengan melakukan <i>random</i> berdasarkan pada nilai suatu keadaan (Q).
mixgauss_init(Q*M, data, cov_type)	Inisialisasi parameter distribusi probabilitas simbol observasi tiap keadaan (B). Ini dalam bentuk <i>output mean</i> dan sigma sebagai nilai dari tiap keadaan (B).	Dalam menentukan parameter tersebut. Dengan mencari nilai mean dan sigma karena menggunakan distribusi normal. Q sebagai jumlah <i>state</i> . M sebagai jumlah <i>mixture</i> ini menggunakan nilai 1 yang berarti tanpa adanya <i>mixture</i> . Cov_type sebagai pengelompokan data menggunakan <i>spherical</i> .
normalise(rand(Q,1))	Inisialisasi suatu keadaan awal ( $\pi$ )	Menentukan nilai parameter keadaan awal berdasarkan jumlah <i>state</i> dengan dinormalisasi random ke satu.

## 2. Tahap Pelatihan HMM

Pada tahap ini sebagai proses dalam membuat model dari HMM. Ini dilakukan dengan mencari nilai optimal dari setiap parameter dan *likelihood*.

Tabel 2.8 Fungsi *toolbox* Pelatihan HMM

Fungsi pada <i>toolbox</i> HMM	Output	Penjelasan
mhmm_em(... Train_Data, prior0, transmat0, mu0, sigma0, mixmat0,'max_iter', max_iter, 'cov_type', cov_type, 'verbose', verbose)	Menemukan nilai optimal dari setiap parameter <i>likelihood</i> , prior1, transmat1, mu1, Sigma1, mixmat1	Parameter yang telah diinisialisasi digunakan untuk tahap pelatihan. Ditambahkan dengan max iter sebagai jumlah iterasi yang ditentukan dalam proses menemukan nilai parameter yang optimal. <i>Verbose</i> sebagai fungsi untuk ditampilkan atau tidak hasil proses iterasi <i>likelihood</i> .

## 3. Tahap Pengujian HMM

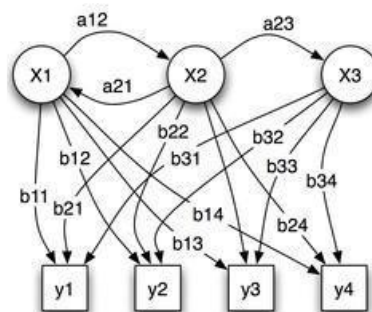
Pada tahap ini fungsi yang digunakan untuk mencari *likelihood* terbesar dengan menggunakan parameter hasil dari pelatihan HMM menggunakan data dari aksara Incung. Ini akan berguna menentukan nilai *likelihood* yang terbesar sebagai hasil prediksi pada kelas dari aksara Incung.

Tabel 2.9 Fungsi *toolbox* Pengujian HMM

Fungsi pada <i>toolbox</i> HMM	Output	Penjelasan
<code>mhmm_logprob(data, prior1, transmat1, mu1, Sigma1, mixmat1)</code>	<i>log-likelihood</i>	Menemukan hasil <i>likelihood</i> dari parameter yang optimal dari hasil pelatihan HMM.
<code>max(log-likelihood)</code>	<i>Likelihood</i> yang memiliki nilai terbesar dan menentukan indeks atau kelas yang diprediksi.	Menemukan <i>likelihood</i> yang terbesar dari hasil optimasi menggunakan parameter hasil pelatihan HMM.

### 2.6.1 Algoritma *Hidden Markov Model*

*Hidden Markov Model* dapat dibangkitkan, setelah mendapatkan ciri. Data ciri dari citra berupa diskret, sehingga dapat dibuat rangkaian *state* dengan angka probabilitas transisi sebagai penghubung. Bentuk pada *Hidden Markov Model* adalah setiap *state* saling terhubung dengan *state* yang lain.



Gambar 2.3 Representasi parameter HMM

x = Kondisi keadaan

y = Observasi yang mungkin

a = Kemungkinan keadaan transisi

b = Kemungkinan output contoh kasus HMM

Algoritma *Hidden Markov Model* dibangkitkan dari beberapa parameter yaitu:

$$\lambda = (N, M, A, B, \pi) \quad (2.1)$$

Permodelan dengan parameter tersebut dapat dijelaskan sebagai berikut:

1. N, jumlah keadaan (*state*) dalam model. Setiap *state* dinotasikan sebagai  $S = \{S_1, S_2, S_3, \dots, S_N\}$  dan *state* pada waktu t ialah  $q_t$ ,
2. M, banyaknya simbol observasi tiap *state*. Simbol individual dinotasikan sebagai  $V = \{V_1, V_2, V_3, \dots, V_M\}$ , simbol dengan panjang T sebagai  $Q_t$
3. Matriks transisi probabilitas  $A = \{a_{ij}\}$ ,

Di mana

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i, \leq N \quad (2.2)$$

$$\text{Dan dengan syarat } a_{ij} \geq 0 \text{ dan } \sum_{j=1}^N a_{ij} = 1 \quad (2.3)$$

4. Simbol distribusi probabilitas pada *state* j,  $B = \{b_j(k)\}$  di mana

$$b_j(k) \geq 0, 1 \leq j \leq N, 1 \leq k \leq M \quad (2.4)$$

$$\sum_{k=1}^M b_j(k) = 1, 1 \leq j \leq N \quad (2.5)$$

5. Probabilitas kondisi awal  $\Pi = \{\pi_i\}$  dengan  $\pi_i$  ialah probabilitas bahwa model berada pada keadaan  $S_i$  saat  $t = 0$  dengan rumus untuk menentukan probabilitas kondisi awal

$$\pi_i = p \{q_i = i\} \text{ dan } 1 \leq i \leq N \quad (2.6)$$

Keterangan:

$\pi_i$  = Probabilitas kondisi awal

$i$  = Jumlah probabilitas pada kondisi awal

$N$  = Jumlah *state* atau keadaan pada model

### 2.6.1.1 Contoh Perhitungan HMM

Keadaan cuaca dapat diobservasi secara langsung. Pada *Hidden Markov Model*, cuaca-cuaca dalam keadaan “*hidden*”. Hal ini dapat diilustrasikan jika kita dikunci di sebuah kamar untuk beberapa hari (tidak dapat melihat kondisi cuaca luar). Cuaca dapat diprediksi yang dilihat dari tanda pada orang-orang yang setiap hari datang ke kamar untuk membawa makanan membawa payung atau tidak.

Tabel 2.10 berikut merupakan tampilan dari probabilitas orang membawa payung berdasarkan keadaan cuaca. Jika orang tersebut membawa payung maka 0.1 cuaca hari itu “*sunny*”, 0.5 cuaca hari itu “*rainy*” dan 0.3 cuaca “*foggy*”.

Tabel 2.10 Probabilitas  $P(x_i|q_i)$  membawa payung berdasarkan cuaca  $q_i$  pada hari  $i$

Tabel 2.10 Probabilitas Membawa Payung

Cuaca	Probabilitas membawa payung
<i>Sunny</i>	0.1
<i>Rainy</i>	0.8
<i>Foggy</i>	0.3

- a. Jika hari saat seseorang terkunci cuaca “*sunny*”. Pada hari berikutnya orang itu yang membawa payung. Berapakah probabilitas masing-masing cuaca pada hari itu?

1. Kemungkinan pada hari kedua adalah “*sunny*”:

$$L(q_2 = \text{sunny} | q_1 = \text{sunny}, x_2 = \text{bawa payung}) =$$

$$P(x_2 = \text{membawa payung} | q_2 = \text{sunny}) *$$

$$P(q_2 = \text{sunny} | q_1 = \text{sunny})$$

$$= 0.1 * 0.8 = 0.08$$

2. Kemungkinan pada hari kedua adalah “rainy”:

$$L(q_2 = \text{rainy} | q_1 = \text{sunny}, x_2 = \text{Bawa payung}) =$$

$$P(x_2 = \text{membawa payung} | q_2 = \text{rainy}) *$$

$$P(q_2 = \text{rainy} | q_1 = \text{sunny}) = 0.8 * 0.05 = 0.04$$

3. Kemungkinan pada hari kedua adalah “foggy”:

$$L(q_2 = \text{foggy} | q_1 = \text{sunny}, x_2 = \text{bawa payung}) =$$

$$P(x_2 = \text{bawa payung} | q_2 = \text{foggy}) * P(q_2 =$$

$$\text{foggy} | q_1 = \text{sunny}) = 0.3 * 0.15 = 0.045$$

- b. Jika Anda tidak mengetahui cuaca pada hari Anda dikunci dalam kamar, tiga hari berturut-turut orang yang mengantarkan makanan tidak membawa payung. Berapakah *likelihood* cuaca tersebut adalah  $\{q_1 = \text{sunny}, q_2 = \text{foggy}, q_3 = \text{sunny}\}$ ?

Jawab:

$$L(q_1 = \text{sunny}, q_2 = \text{foggy}, q_3 = \text{sunny} | x_1 = \text{tidak bawa payung}, x_2 = \text{tidak bawa payung}, x_3 = \text{tidak bawa payung}) = P(x_1 = \text{tidak bawa payung} | q_1 = \text{sunny}) *$$

$$P(x_2 = \text{tidak bawa payung} | q_2 = \text{foggy}) *$$

$$P(x_3 = \text{tidak bawa payung} | q_3 = \text{sunny}) *$$

$$P(q_1 = \text{sunny}) * P(q_2 = \text{foggy} | q_1 = \text{sunny}) *$$

$$P(q_3 = \text{sunny} | q_2 = \text{foggy}) = 0.9 * 0.7 * 0.9 * 1/3 * 0.15 * 0.2 = 0.0057$$

## 2.6.2 Tahap Pelatihan HMM



Pada tahap pelatihan HMM permasalahan yang ditemukan ketika menggunakan metode *Hidden Markov Model* adalah perlu mengatur parameter model  $\lambda = \{A, B, \pi\}$  agar probabilitas urutan observasi yaitu  $P(O|\lambda)$  menjadi maksimal. Parameter di model  $\lambda = \{A, B, \pi\}$  untuk memaksimalkan  $P(O|\lambda)$  dapat diperoleh dengan prosedur iterasi yaitu dengan Algoritma *forward-backward* atau biasa disebut dengan metode *Baum-Welch*.

Variabel *forward* didefinisikan sebagai observasi parsial dan probabilitas *state* sekuens yang dilambangkan dengan  $O_1, O_2, \dots, O_t$  (hingga waktu  $t$ ) dan *state*  $S_i$  pada waktu  $t$ , dengan model  $\lambda$ , dan  $\alpha$  sebagai  $t(i)$ . Pada variabel *backward* didefinisikan sebagai observasi parsial dari probabilitas *state* sekuens dari  $t+1$  ke *state* sebelumnya, yang mana terdapat *state*  $S_i$  pada waktu  $t$ , dengan model  $\lambda$ , dan  $\alpha$  sebagai  $t(i)$ . Observasi dari probabilitas *state* sekuens ini dihitung dengan rumus:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_t(i) \beta_t(i) = \sum_{i=1}^N \alpha_t(i) \quad (2.7)$$

Keterangan :

$O$  = Nilai pada fitur

$\lambda$  = Probabilitas model pada HMM

$N$  = Jumlah keadaan

$\alpha_t$  = *Alfa* sebagai nilai perhitungan dari *forward* yang memiliki probabilitas optimal

$\beta_t$  = *Beta* sebagai nilai perhitungan dari *backward* yang memiliki probabilitas optimal

$i$  = Banyaknya perulangan yang dilakukan

Probabilitas pada saat berada pada *state*  $S_i$  pada waktu  $t$ , dan diberikan sekuens observasi), dan model  $\lambda$  ialah:

$$\gamma_t = \frac{\alpha_t(i) \beta_t(i)}{P(O|\lambda)} \quad (2.8)$$

Keterangan:

$\gamma$  = Probabilitas dari suatu kondisi

$\alpha$  = Alfa sebagai hasil probabilitas pada *forward*

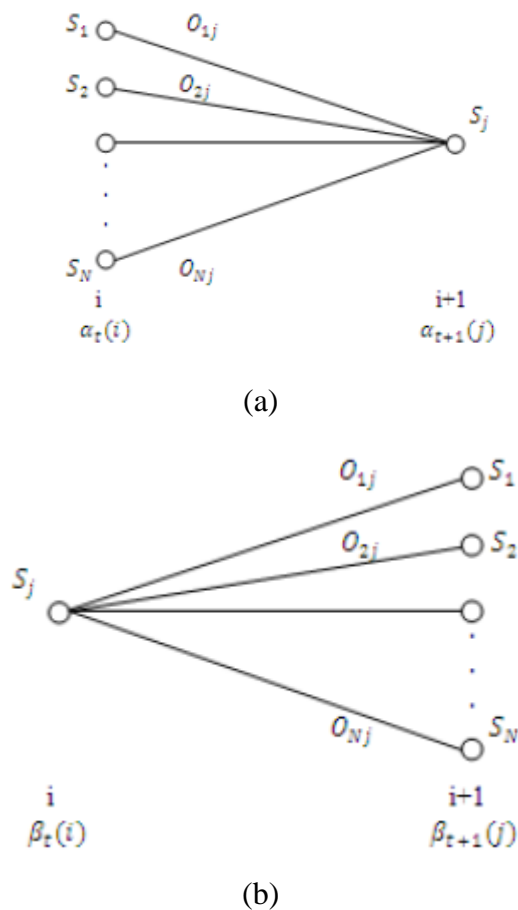
$\beta$  = Beta sebagai hasil probabilitas pada *backward*

$O$  = Nilai pada fitur

$t$  = Jumlah observasi

$i$  = Jumlah perulangan yang digunakan

$\lambda$  = Probabilitas dari model HMM



Gambar 2.4 Ilustrasi dari operasi pada *Baum-Welch*

(a) Variabel *Forward*

(b) Variabel *Backward*

Perhitungan pada model *Hidden Markov Model* banyak sampel data dari kelas yang akan dimodelkan. Parameter ini yang dipetakan sebagai ekuivalensi dari  $O_t$ .

### 2.6.2.1 Contoh Perhitungan dengan Algoritma *Baum-Welch*

Diketahui:

$$A = \begin{bmatrix} 0.8 & 0.04 & 0.16 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.4 & 0.5 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \\ 0.6 & 0.4 \end{bmatrix}$$

$$\pi = \begin{bmatrix} 0.8 \\ 0 \\ 0.2 \end{bmatrix}$$

Pada kasus perasaan orang yang masuk ke ruangan, setelah dilakukan observasi selama tiga hari berturut-turut adalah *happy*, *sad*, dan *happy*. Tentukan peluang barisan observasi  $O = \{Happy, Sad, Happy\}$  dengan model  $\lambda = (A, B, \pi)$ .

Solusi:

Algoritma *Forward*

1. Inisialisasi

$$a_1(1) = \pi_1 b_1(O_1) = (0.8)(0.8) = 0.64$$

$$a_1(2) = \pi_2 b_2(O_1) = (0)(0.1) = 0$$

$$a_1(3) = \pi_3 b_3(O_1) = (0.2)(0.6) = 0.12$$

2. Pengenalan

$$t = 2, O_2 = sad$$

$$a_2(j) = \left[ \sum_{i=1}^N a_1(i) a_{ij} \right] b_j(O_2)$$

$$\begin{aligned} a_2(1) &= [(0.64)(0.8) + (0) + (0.12)(0.1)]0.2 \\ &= 0.1048 \end{aligned}$$

$$\begin{aligned} a_2(2) &= [(0.0256) + (0) + (0.048)]0.9 \\ &= 0.06624 \end{aligned}$$

$$\begin{aligned} a_2(3) &= [(0.1024) + (0) + (0.06)]0.4 \\ &= 0.06496 \end{aligned}$$

$$t = 3, O_3 = happy$$

$$a_3(j) = \left[ \sum_{i=1}^N a_2(i) a_{ij} \right] b_j(O_3)$$

$$\begin{aligned} a_3(1) &= [(0.08384) + (0.013248) \\ &\quad + (0.006496)]0.8 \\ &= 0.08286 \end{aligned}$$

$$\begin{aligned} a_3(2) &= [(0.04192) + (0.03974) \\ &\quad + (0.002598)]0.1 \\ &= 0.00699 \end{aligned}$$

$$\begin{aligned} a_3(3) &= [(0.016768) + (0.013248) \\ &\quad + (0.03248)]0.6 \\ &= 0.03749 \end{aligned}$$

### 3. Terminasi

$$\begin{aligned} P(O = \text{Happy, Sad, Happy} \mid \lambda) \\ &= \sum_{i=1}^N a_T(i) \\ &= a_3(1) + a_3(2) + a_3(3) \\ &= 0.08286 + 0.00699 + 0.03749 = 0.13 \end{aligned}$$

### Algoritma *Backward*

#### 1. Inisialisasi

$$\beta_T(i) = 1$$

$$\beta_T(i) = 1$$

$$\beta_3(1) = \beta_3(2) = \beta_3(3) = 1$$

#### 2. Pengenalan

$$t = 2, O_3 = \text{Happy}$$

$$\beta_2(i) = \sum_{j=1}^N b_j(O_3) \beta_3(j) a_{ij}$$

$$\beta_2(1) = [(0.64) + (0.004) + (0.096)] = 0.74$$

$$\beta_2(2) = [(0.16) + (0.06) + (0.12)] = 0.34$$

$$\beta_2(3) = [(0.08) + (0.04) + (0.3)] = 0.42$$

$$t = 1, O_2 = \text{Sad}$$

$$\beta_2(i) = \sum_{j=1}^N b_j(O_2) \beta_2(j) a_{ij}$$

$$\begin{aligned}\beta_2(1) &= [(0.1184) + (0.01224) + (0.02688)] \\ &= 0.15752\end{aligned}$$

$$\begin{aligned}\beta_2(2) &= [(0.0296) + (0.1836) + (0.0336)] \\ &= 0.2468\end{aligned}$$

$$\begin{aligned}\beta_2(3) &= [(0.0148) + (0.0544) + (0.189)] \\ &= 0.2582\end{aligned}$$

### 3. Terminasi

$$\begin{aligned}P(O = \textit{Happy, Sad, Happy} \mid \lambda) \\ &= \sum_{i=1}^N \beta_1(i) \pi(i) b_i(O_1) \\ &= 0.10081 + 0 + 0.03098 = 0.13\end{aligned}$$

Pada kasus suasana hati seorang yang masuk pada ruangan, setelah dilakukan observasi, urutan suasana hati *happy, sad, happy*.

Menentukan estimasi parameter  $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$  untuk maksimal dari  $P(O|\lambda)$ .

Penyelesaian:

$$\xi_t(i, j) = \frac{a_t(1) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} \quad (2.8)$$

Keterangan:

$\xi_t(i, j)$  = probabilitas yang paling optimal

$a$  = Probabilitas pada kondisi transisi

$b$  = Probabilitas pengamatan

$O$  = Nilai fitur dari model

$\beta$  = *Beta* sebagai hasil probabilitas optimal pada *backward*

$\lambda$  = *Lambda* pada model HMM dengan nilai probabilitas

Tabel 2.11 Estimasi Probabilitas

For t = 1	For t = 2
-----------	-----------

$\xi_1(1,1) = \frac{(0.64)(0.8)(0.2)(0.74)}{0.13}$ $= 0.58289$	$\xi_2(1,1) = \frac{(0.1048)(0.8)(0.8)(1)}{0.13}$ $= 0.51593$
$\xi_1(1,2) = \frac{(0.64)(0.04)(0.9)(0.34)}{0.13}$ $= 0.06025$	$\xi_2(1,2) = \frac{(0.1048)(0.8)(0.2)(1)}{0.13}$ $= 0.00322$
$\xi_1(1,3) = \frac{(0.64)(0.8)(0.2)(0.74)}{0.13}$ $= 0.58289$	$\xi_2(1,3) = \frac{(0.1048)(0.8)(0.2)(1)}{0.13}$ $= 0.07739$
$\xi_1(2,1) = \frac{(0)(0.2)(0.2)(0.74)}{0.13} = 0$	$\xi_2(2,1) = \frac{(0.06624)(0.8)(0.2)(1)}{0.13}$ $= 0.08152$
$\xi_1(2,2) = \frac{(0)(0.6)(0.9)(0.34)}{0.13} = 0$	$\xi_2(2,2) = \frac{(0.06624)(0.6)(0.1)(1)}{0.13}$ $= 0.03057$
$\xi_1(2,3) = \frac{(0)(0.2)(0.4)(0.42)}{0.13} = 0$	$\xi_2(2,3) = \frac{(0.06624)(0.2)(0.6)(1)}{0.13}$ $= 0.06114$
$\xi_1(3,1) = \frac{(0.12)(0.1)(0.2)(0.74)}{0.13}$ $= 0.01366$	$\xi_2(3,1) = \frac{(0.06496)(0.1)(0.8)(1)}{0.13}$ $= 0.03997$
$\xi_1(3,2) = \frac{(0.12)(0.4)(0.2)(0.34)}{0.13}$ $= 0.11298$	$\xi_2(3,2) = \frac{(0.06496)(0.4)(0.1)(1)}{0.13}$ $= 0.01998$
$\xi_1(3,3) = \frac{(0.12)(0.5)(0.4)(0.42)}{0.13}$ $= 0.07753$	$\xi_2(3,3) = \frac{(0.06496)(0.5)(0.6)(1)}{0.13}$ $= 0.14991$

Mencari hasil dari nilai  $\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j)$  (2.9)

For t = 1

$$\gamma_1(1) = [0.58289 + 0.06025 + 0.13233] = 0.8$$

$$\gamma_1(2) = [0 + 0 + 0] = 0$$

$$\gamma_1(3) = [0.01366 + 0.11298 + 0.07753] = 0.2$$

For t = 2

$$\gamma_2(1) = [0.51593 + 0.00322 + 0.07739] = 0.59$$

$$\begin{aligned}\gamma_2(2) &= [0.08152 + 0.03057 + 0.06114] \\ &= 0.17\end{aligned}$$

$$\gamma_2(3) = [0.03997 + 0.01998 + 0.14991] = 0.21$$

Kemudian, dengan menggunakan hasil perhitungan dapat ditemukan estimasi HMM  $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$

Keterangan:

$\bar{A}$  = Estimasi *state* transisi probabilitas distribusi

$\bar{B}$  = Estimasi observasi probabilitas distribusi

$\bar{\pi}$  = Estimasi inisial *state* distribusi

$$\bar{\pi} = \begin{bmatrix} \gamma_1(1) \\ \gamma_1(2) \\ \gamma_1(3) \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0 \\ 0.2 \end{bmatrix}$$

$\bar{\pi}$  adalah estimasi matriks probabilitas keadaan awal. Sehingga terpenuhi, probabilitas keadaan awal saat proses dalam keadaan “*sunny*” adalah 0.8 estimasi probabilitas keadaan awal saat proses dalam keadaan “*rainy*” *state* adalah 0 dan estimasi probabilitas keadaan awal saat proses dalam keadaan “*cloudy*” *state* adalah 0.2.

$$\begin{aligned}\bar{A} &= \begin{bmatrix} \frac{\sum_{i=1}^T \xi_t(1,1)}{\sum_{i=1}^T \gamma_t(1)} & \frac{\sum_{i=1}^T \xi_t(1,2)}{\sum_{i=1}^T \gamma_t(1)} & \frac{\sum_{i=1}^T \xi_t(1,3)}{\sum_{i=1}^T \gamma_t(1)} \\ \frac{\sum_{i=1}^T \xi_t(2,1)}{\sum_{i=1}^T \gamma_t(2)} & \frac{\sum_{i=1}^T \xi_t(2,2)}{\sum_{i=1}^T \gamma_t(2)} & \frac{\sum_{i=1}^T \xi_t(2,3)}{\sum_{i=1}^T \gamma_t(2)} \\ \frac{\sum_{i=1}^T \xi_t(3,1)}{\sum_{i=1}^T \gamma_t(3)} & \frac{\sum_{i=1}^T \xi_t(3,2)}{\sum_{i=1}^T \gamma_t(3)} & \frac{\sum_{i=1}^T \xi_t(3,3)}{\sum_{i=1}^T \gamma_t(3)} \end{bmatrix} \\ &= \begin{bmatrix} 0.8 & 0.05 & 0.15 \\ 0.5 & 0.2 & 0.3 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}\end{aligned}$$

$\bar{A}$  adalah estimasi matriks transisi A. jadi  $P(O|\bar{\lambda}) \geq P(O|\lambda)$  terpenuhi, probabilitas dari “*sunny*” *state* ke “*sunny*” *state* adalah 0.8, ke “*rainy*” *state* adalah 0.05 dan “*cloudy*”

$$\bar{B} = \begin{bmatrix} \frac{\sum_{i=1}^T O_{t=1} \gamma_t(1)}{\sum_{i=1}^T \gamma_t(1)} & \frac{\sum_{i=1}^T O_{t=2} \gamma_t(1)}{\sum_{i=1}^T \gamma_t(1)} \\ \frac{\sum_{i=1}^T O_{t=1} \gamma_t(2)}{\sum_{i=1}^T \gamma_t(2)} & \frac{\sum_{i=1}^T O_{t=2} \gamma_t(2)}{\sum_{i=1}^T \gamma_t(2)} \\ \frac{\sum_{i=1}^T O_{t=1} \gamma_t(3)}{\sum_{i=1}^T \gamma_t(3)} & \frac{\sum_{i=1}^T O_{t=2} \gamma_t(3)}{\sum_{i=1}^T \gamma_t(3)} \end{bmatrix} = \begin{bmatrix} 0.6 & 0.4 \\ 0 & 1 \\ 0.5 & 0.5 \end{bmatrix}$$

$\bar{B}$  adalah estimasi matriks emisi B. Sehingga  $P(O|\bar{\lambda}) \geq P(O|\lambda)$  terpenuhi peluang penjaga dengan suasana hati “happy” saat cuaca di luar “sunny” adalah 0.6 saat cuaca di luar “rainy” adalah 0 dan saat cuaca luar “cloudy” adalah 0.5. Maka peluang dari seorang dengan suasana hati “sad” saat cuaca di luar “sunny” adalah 0.4 saat cuaca luar “rainy” adalah 1 dan saat cuaca luar “cloudy” adalah 0.5.

### 2.6.3 Tahap Pengujian HMM

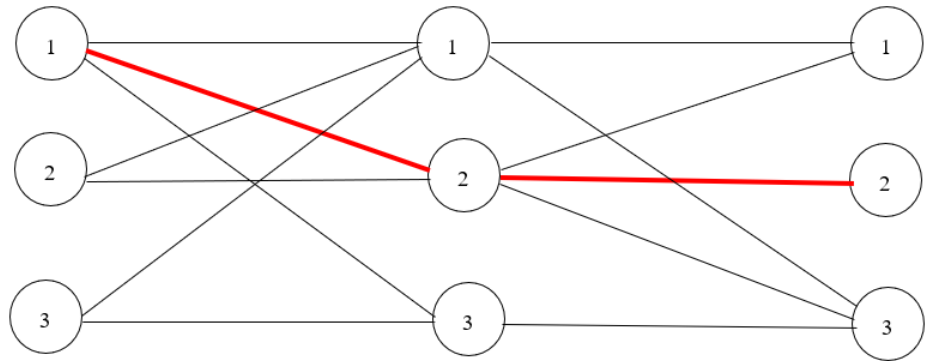
Pada tahap ini dilakukan proses pencocokan kemiripan aksara dihitung dengan algoritma *Viterbi* dengan mencari nilai maksimum dari perbandingan probabilitas data baru dengan probabilitas dari masing-masing model HMM.

Algoritma *Viterbi* digunakan untuk menghitung sekuens *state*  $Q = \{q_1 q_2 \dots q_T\}$ , yang paling dekat probabilitas dari sebuah sekuens observasi  $O = \{O_1 O_2 \dots O_T\}$  yang didefinisikan sebagai:

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_3} P[q_1 q_2 \dots q_t = i, O_1 O_2 \dots O_t | \lambda] \quad (2.10)$$

$\delta(i)$  merupakan nilai probabilitas terbaik pada waktu  $t$ , yang dihitung pada observasi  $t$  pertama dan diakhiri pada *state*  $S_i$ .





Gambar 2.5 Ilustrasi pencarian nilai terbaik dengan *Viterbi*

Garis tebal menunjukkan nilai probabilitas terbaik antar *state*.

### 2.6.3.1 Contoh Perhitungan dengan Algoritma *Viterbi*

Dalam hal suasana hati seorang yang masuk ke ruangan, setelah dilakukan observasi, urutan suasana hati orang itu dalam tiga hari berturut-turut adalah *happy*, *sad*, *happy* dan diberi model  $\lambda = (A, B, \pi)$ . Memilih urutan state tersembunyi yang paling optimal.

Penyelesaian:

#### 1. Inisialisasi

$$\delta_1(i) = \pi_1 b_1(O_1)$$

$$\psi_1(i) = 0$$

$$\delta_1(1) = \pi_1 b_1(O_1) = (0.8)(0.8) = 0.64$$

$$\delta_1(2) = \pi_2 b_2(O_1) = (0)(0.1) = 0$$

$$\delta_1(3) = \pi_3 b_3(O_1) = (0.2)(0.6) = 0.12$$

$$\psi_1(1) = \psi_1(2) = \psi_1(3) = 0$$

#### 2. Rekursif

$$\delta_t(i) = \max_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ji}] b_i(O_t)$$

$$\psi_t(j) = \underset{1 \leq i \leq N}{\operatorname{argmax}} [\delta_{t-1}(i) a_{ij}]$$

For  $t=2$ ,  $O_2 = 2(\text{Sad})$

$$\delta_2(1) = \max \{ (0.512), (0), (0.012) \} 0.2 = 0.1024$$

$$\Psi_2(1) = 1(\text{Sunny})$$

$$\delta_2(2) = \max \{(0.0256), (0), (0.048)\} 0.9 = 0.0432$$

$$\Psi_2(2) = 2(\text{Cloudy})$$

$$\delta_2(3) = \max \{(0.1024), (0), (0.06)\} 0.4 = 0.04096$$

$$\Psi_2(3) = 3(\text{Sunny})$$

$$\text{For } t=3, O_3 = 1(\text{Happy})$$

$$\delta_3(1) = \max \{(0.08192), (0.00864), (0.00409)\} 0.8 = 0.06553$$

$$\Psi_3(1) = 1(\text{Sunny})$$

$$\delta_3(2) = \max \{(0.00409), (0.02592), (0.01638)\} 0.1 = 0.00259$$

$$\Psi_3(2) = 2(\text{Rainy})$$

$$\delta_3(3) = \max \{(0.01638), (0.00864), (0.02048)\} 0.6 = 0.01228$$

$$\Psi_3(3) = 3(\text{Cloudy})$$

### 3. Terminasi

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

$$q_T^* = \operatorname{argmax}_{1 \leq i \leq N} [\delta_T(i)]$$

$$P^* = \max \{\delta_3(1), \delta_3(2), \delta_3(3)\}$$

$$= \max \{(0.06553), (0.00259), (0.01228)\} = 0.06553$$

$$q_3^* = \operatorname{argmax}\{\delta_3(1), \delta_3(2), \delta_3(3)\} = 1 (\text{Sunny})$$

### 4. Backtracking

$$q_2^* = \Psi_3(q_3)^* = \Psi_3(1) = \text{Sunny}$$

$$q_1^* = \Psi_2(q_2)^* = \Psi_2(1) = \text{Sunny}$$

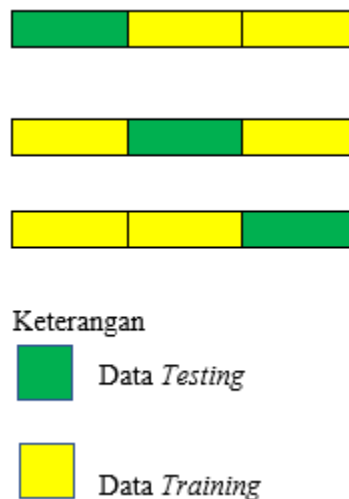
Jadi, ketika seorang yang masuk ke ruangan dengan urutan suasana hati *happy*, *sad*, *happy* urutan keadaan tersembunyi dalam hal ini keadaan cuaca di luar yang optimal adalah

$$q^* = \{1(\text{Sunny}), 1(\text{Sunny}), 1(\text{Sunny})\}.$$

## 2.7 Metode Evaluasi *K-Fold Cross Validation*

Proses yang telah dilakukan menggunakan *Hidden Markov Model* untuk mengukur tingkat keberhasilan dapat diukur menggunakan metode *K-fold cross validation*. Metode ini dilakukan dengan membagi tiap kelompok aksara menjadi beberapa kelompok data yang kemudian bergantian data untuk *testing* maupun *training* dalam beberapa langkah pengujian yang saling silang.

Hasil yang akan diperoleh menggunakan metode ini akan menampilkan hasil akurasi. Gambar 2.6 merupakan ilustrasi pembagian data menggunakan nilai *k-fold cross validation* dengan  $k = 3$ . Setiap *fold* di bagi  $1/3$  sebagai data *testing* dan  $2/3$  sebagai data *training*.



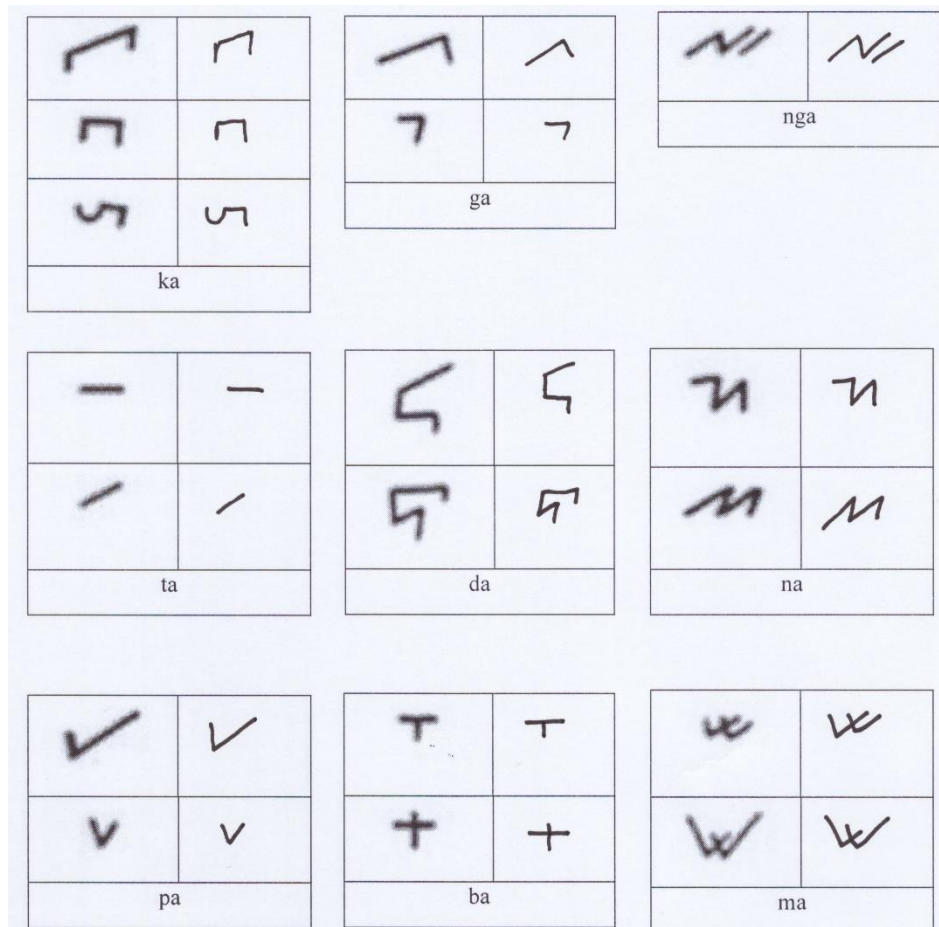
Gambar 2.6 Ilustrasi Model 3-Fold Cross Validation

### BAB III

#### METODE PENELITIAN

##### 3.1 Data

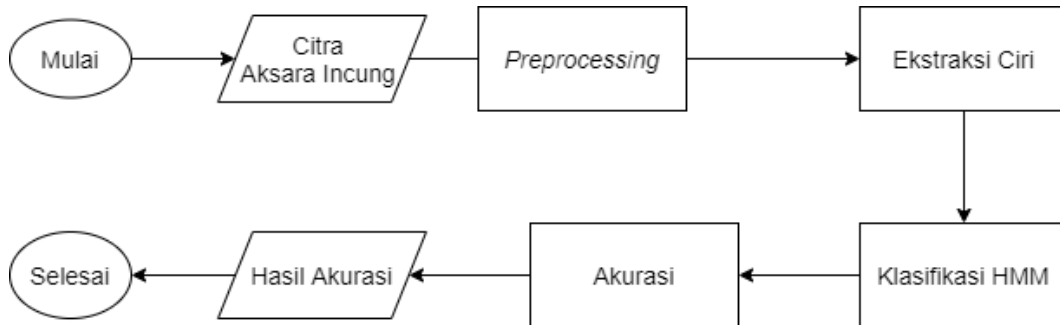
Sistem pengenalan aksara Incung ini digunakan untuk membuat model pada metode *Hidden Markov Model* dari tulisan tangan aksara Incung dari beberapa orang. Penelitian ini menggunakan data yang berupa citra aksara Incung yang berasal dari beberapa responden dalam menuliskan aksara Incung.



Gambar 3.1 Angket Aksara Incung

### 3.2 Perancangan Sistem

Pada tahap ini akan menunjukkan perancangan sistem yang akan dibuat pada penelitian ini. Desain alat uji pada sistem sebagai berikut:



Gambar 3.2 Diagram Perancangan Sistem

Sistem akan melakukan klasifikasi aksara Incung menggunakan metode *Hidden Markov Model*. Tulisan aksara Incung dari reponden kemudian dilakukan pindai untuk mengubah menjadi data citra digital. Citra hasil pindai tersebut sudah dilakukan proses akusisi pada data aksara Incung dengan proses *cropping* secara manual. Citra aksara akan dilakukan proses *preprocessing* dari setiap citra aksara. Hasil dari *preprocessing* dilakukan proses ekstraksi ciri menggunakan IoC atau *Mark Direction*. Ekstraksi ciri tersebut digunakan untuk proses klasifikasi dengan metode HMM. Itu akan mendapatkan hasil klasifikasi dalam bentuk label, maka akan di evaluasi untuk mendapatkan akurasi dari penggunaan metode klasifikasi dengan *Hidden Markov Model*.

#### 3.2.1 Akusisi Data

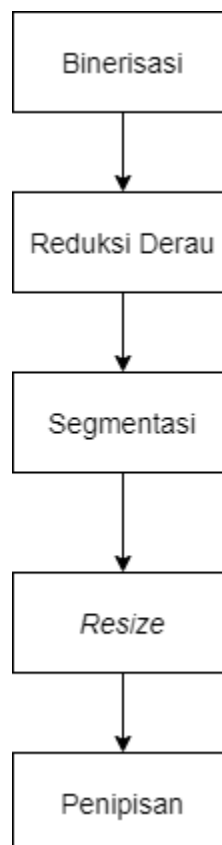
Akusisi citra merupakan proses menangkap (*capture*) atau memindai (*scan*) suatu citra analog menjadi sebuah citra digital. Pada tahap ini diawali dengan melakukan pengambilan gambar pada objek dengan menggunakan media scanner. Objek yang berupa citra aksara Incung yang diambil dari 90 responden yang berbeda dengan kertas ukuran A4.

### 3.2.2 Cropping

*Cropping* merupakan metode untuk mendapatkan bentuk karakter aksara secara utuh, tanpa adanya ‘ruang’ lebih pada citra aksara. Metode yang digunakan untuk melakukan *cropping* atau pemotongan citra pada penelitian ini yaitu secara manual menggunakan Photoshop 2018.

### 3.2.3 Preprocessing

Citra aksara yang telah dipindai berupa RGB (*Red Green Blue*) yang perlu dilakukan proses *processing* supaya bisa diolah dalam proses pengenalan dan pengujian aksara. Dalam proses *preprocessing* yang diperlukan dalam sistem ini adalah proses binerisasi, reduksi derau, segmentasi, *resize*, dan penipisan.



Gambar 3.4 Diagram Blok *Preprocessing*

### 3.2.3.1 Binerisasi

Data citra yang akan dilakukan proses *preprocessing* untuk mempermudah proses ekstraksi ciri. Hal ini dilakukan setelah melakukan pemotongan citra aksara Incung. Binerisasi sebagai proses mengubah citra menjadi hitam putih. Dalam hal ini digunakan fungsi dari MATLAB yaitu *im2bw*.

### 3.2.3.2 Reduksi Derau

Pada tahap ini citra hasil binerisasi akan di tukar warna *background* yang sebelumnya putih (1) menjadi hitam (0) dan *foreground* sebelumnya hitam (0) menjadi putih (1). Digunakan *function* dari MATLAB yaitu *imcomplement*.

Reduksi derau ini dilakukan untuk menghilangkan derau pada citra. Dalam hal ini menggunakan *Area open* untuk menghilangkan piksel yang bukan bagian dari karakter aksara. Itu dilakukan dengan melihat jumlah piksel tersebut pada suatu area. Apabila kurang dari nilai batas yang ditentukan, maka piksel tersebut akan dijadikan *background*. Dalam hal ini menggunakan fungsi dari MATLAB yaitu *bwareaopen*. Kemudian untuk menghilangkan lubang-lubang pada objek citra maka digunakan *function* dari MATLAB yaitu *imfill*.

Setelah itu dilakukan proses *invers* kembali untuk mengembalikan citra yang seperti awal yaitu di tukar warna *background* yang sebelumnya hitam (0) menjadi putih (1) dan *foreground* sebelumnya putih (1) menjadi hitam (0).

### 3.2.3.3 Segmentasi

Tahap yang digunakan untuk memotong tepi pada setiap aksara, agar dapat dilakukan ekstraksi ciri, sehingga ciri tersebut dapat digunakan untuk melakukan klasifikasi. Proses ini menggunakan metode profil proyeksi.

Algoritma metode profil proyeksi sebagai berikut:

1. Proyeksi Horizontal

Langkah 1: Hitung piksel berwarna hitam pada setiap baris pada citra.

Langkah 2: Lakukan pemotongan citra dengan batas baris yang bernilai 0

2. Proyeksi Vertikal

Langkah 3: Dari hasil pemotongan citra, hitung piksel berwarna hitam pada setiap kolom pada citra hasil pemotongan

Langkah 4: Lakukan pemotongan citra dengan batas kolom yang bernilai 0

Langkah 5: Selesai

#### 3.2.3.4 *Resize*

*Resize* merupakan metode untuk mengubah ukuran citra menjadi besar atau menjadi kecil. Hal ini digunakan karena pada tahap sebelumnya ukuran citra masih berbeda-beda. Pada penelitian ini, penulis akan menggunakan fungsi pada MATLAB yaitu *imresize*.

#### 3.2.3.5 Penipisan

Penipisan yang akan dilakukan pada setiap aksara Incung menggunakan metode *Rosenfeld*. Pada metode *Rosenfeld* merupakan algoritma yang digunakan untuk melakukan penipisan (*thinning*) terhadap citra aksara, sehingga hanya akan menyisihkan citra kerangka aksara saja. Metode ini bekerja dengan cara membuang bagian piksel yang merupakan bagian tepi (*edge*) (Widiarti, 2011).

#### 3.2.4 Ekstraksi Ciri

Ekstraksi ciri sebagai proses dalam memperoleh sifat pola dari suatu objek yang ada pada citra. Sehingga citra dapat



dibedakan dengan citra yang lainnya. Dalam hal ini ekstraksi ciri yang digunakan adalah *Intensity of Character* dan *Mark Direction*.

### 1. *Intensity of Character*

Pada tahap ini data aksara yang telah melalui tahap *preprocessing* dan penipisan akan dilakukan proses ekstraksi ciri dengan IoC. Pada perhitungan IoC dilakukan dengan mengubah matrik hasil segmentasi menjadi matriks baru dengan ukuran NxN melalui cara menambahkan jumlah piksel hitam sebanyak ukuran gambar dibagi dengan ukuran matriks IoC.

Algoritma ekstraksi ciri menggunakan IoC

1. Buka berkas masukan, misal = aksara
2. Set tinggi = size(aksara,1)
3. Set lebar = size(aksara,2)
4. Bagi aksara menjadi bagian yang dikehendaki
5. Cari jumlahan piksel yang bernilai 0 di setiap bagian karakter hasil langkah 2
6. Simpan ciri tersebut menjadi ciri dari karakter yang bersangkutan ke dalam file ciriAksara.mat

### 2. *Mark Direction*

Pada tahap ini data aksara akan dilakukan ekstraksi ciri dengan menemukan jumlah hitam dalam bentuk diagonal kiri, diagonal kanan, horizontal, dan vertikal.

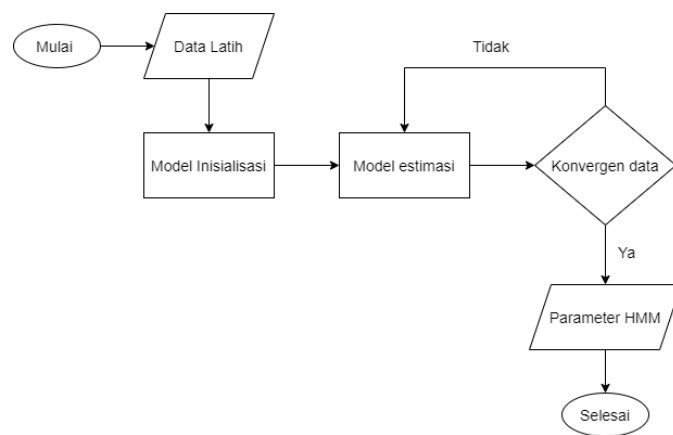
### Algoritma *Mark Direction*

1. Mulai
2. Membaca kolom dan baris pada sebuah citra.
3. Membuat variabel baru bernama *vert*, *horz*, *dig1*, *dig2*
4. Membaca kolom mulai dari 1 dari banyaknya kolom
5. Membaca baris mulai dari 1 dari banyaknya baris.
6. Jika *image* (*i*, *j*) sama dengan 0 lakukan langkah 7
7. Jika *image* (*i*, *j*) + *image* (*i*-1, *j*) + *image* (*i*+1, *j*) sama dengan 0 lakukan langkah 8
8. Jika *image* (*i*-1, *j*-1) + *image* (*i*-1, *j*+1) + *image* (*i*, *j*+1) + *image* (*i*+1, *j*-1) + *image* (*i*+1, *j*+1) sama dengan 6, maka tambahkan counter *vert*=*vert*+1
9. Jika *image* (*i*, *j*) + *image* (*i*, *j*-1) + *image* (*i*, *j*+1) sama dengan 0 lakukan langkah 10
10. Jika *image* (*i*-1, *j*-1) + *image* (*i*-1, *j*) + *image* (*i*+1, *j*-1) + *image* (*i*+1, *j*) + *image* (*i*+1, *j*+1) sama dengan 6, maka tambahkan counter *horz*=*horz*+1
11. Jika *image* (*i*, *j*) + *image* (*i*-1, *j*-1) + *image* (*i*+1, *j*+1) sama dengan 0 lakukan langkah
12. Jika *image* (*i*-1, *j*) + *image* (*i*-1, *j*+1) + *image* (*i*, *j*-1) + *image* (*i*, *j*+1) + *image* (*i*, *j*+1) + *image* (*i*+1, *j*-1) + *image* (*i*+1, *j*) sama dengan 6, maka tambahkan counter *dig1*=*dig1*+1
13. Jika *image* (*i*, *j*) + *image* (*i*-1, *j*+1) + *image* (*i*+1, *j*-1) sama dengan 0 lakukan langkah 14
14. Jika *image* (*i*-1, *j*-1) + *image* (*i*-1, *j*) + *image* (*I*, *j*-1) + *image* (*I*, *j*+1) + *image* (*i*+1, *j*) + *image* (*i*+1, *j*+1) sama dengan 6, maka tambahkan counter *dig2*=*dig2*+1
15. Selesai

Nilai dari ekstraksi ciri *Mark Direction* yang terbentuk dari konvolusi matriks 3x3.

### 3.2.2 Tahap Pelatihan HMM

Ada dua permasalahan pokok pada aplikasi HMM untuk pengenalan aksara. Masalah pertama adalah pembuatan model untuk mewakili suatu kontur aksara, digunakan algoritma *Baum-Welch* dengan re-estimasi dalam memecahkannya yang dilakukan pada tahap pelatihan (*training*). Gambar 3.3 ini sebagai tahap pelatihan yang dilakukan untuk mendapatkan parameter HMM.



Gambar 3.3 Diagram Tahap Pelatihan

#### 3.2.2.1 Inisialisasi Elemen-elemen HMM

Pemodelan HMM terdiri dari 3 matriks probabilitas, yaitu matriks transisi antara *state*(A), matriks probabilitas pengamatan suatu *state* (B), dan matriks probabilitas awal *state* ( $\pi$ ). Jenis HMM yang digunakan merupakan tipe *ergodic*, dimana parameter HMM seperti matriks A, B, dan  $\pi$  dibangkitkan secara random dengan nilai yang dinormalisasi ke satu. Nilai matriks A, B dan  $\pi$  tersebut kemudian akan dilakukan re-estimasi melalui proses pelatihan untuk mendapatkan nilai parameter yang optimal.

### 3.2.2.2 Re-estimasi Parameter HMM

Re-estimasi parameter sebagai proses *training* pada HMM dengan menggunakan algoritma *Baum-Welch*. Dengan re-estimasi ini tiap model  $\lambda = (A, B, \pi)$  awal akan diadaptasi berulang-ulang sampai diperoleh model terbaik. Iterasi untuk re-estimasi akan berhenti jika iterasi sudah mencapai nilai maksimal.

Pada  $\lambda = (A, B, \pi)$  dimana, A, B, dan  $\pi$  berturut-turut menyatakan distribusi probabilitas transisi antara *state*, distribusi probabilitas simbol observasi, dan distribusi inisialisasi *state*. Pada simbol O sebagai objek pengamatan.

Nilai  $\varepsilon$  adalah nilai batasan dalam menentukan tingkat konvergensi model yang dibuat, sehingga nilai  $\varepsilon$  akan diambil cukup kecil, untuk meminimalkan selisih perubahan nilai *likelihood*. Pada sistem ini proses *training* dan *testing* menggunakan nilai  $\varepsilon = 10^{-3}$ .

Algoritma *Baum-Welch*:

1. Estimasi sebuah model HMM sebagai  $\lambda = (A, B, \pi)$ .
2. Dengan nilai  $\lambda$  dan sekuens observasi O, menghitung sebuah model baru:

$$\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi}), \text{seperti } P(O | \bar{\lambda}) > P(O | \lambda) \quad (3.1)$$

Keterangan:

$\bar{A}$  = estimasi *state* transisi probabilitas distribusi

$\bar{B}$  = estimasi observasi probabilitas distribusi

$\bar{\pi}$  = estimasi inisial *state* distribusi

O = nilai fitur dari model

$\lambda$  = nilai probabilitas dari model HMM

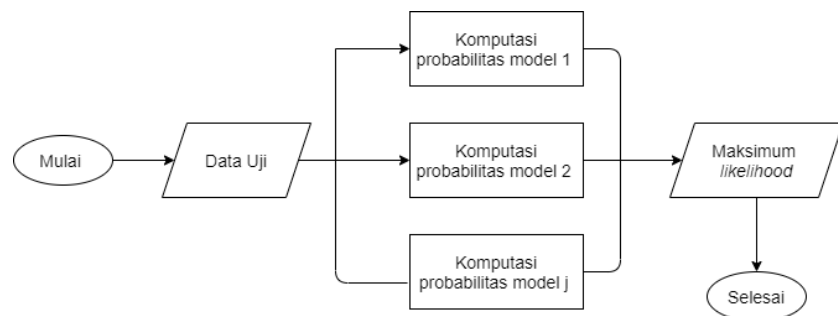
3. Jika menghasilkan  $\frac{P(O|\bar{\lambda}) - P(O|\lambda)}{P(O|\bar{\lambda})} < threshold$ , maka (3.2)

hentikan langkah ini. Jika tidak, maka memasukan nilai  $\bar{\lambda}$  untuk menggantikan  $\lambda$  dan ulangi dilangkah 1.

### 3.2.3 Tahap Pengujian HMM

Pada tahap pengujian ini sebagai proses hasil dari pemodelan dengan HMM akan dibandingkan probabilitas perhitungan citra pelatihan dengan HMM setiap kelas yaitu dengan menghitung peluang maksimum suatu deretan observasi yang dibangkitkan oleh model dari aksara citra tertentu.

Proses klasifikasi citra aksara dilakukan dengan menghitung *likelihood* dari data pengujian. Ini akan dikenali terhadap semua model  $\lambda = (A, B, \pi)$  setiap kelas telah dikenai proses pelatihan dengan HMM. *Output* dari proses pengujian ini akan memberikan indeks dari model kelas yang mempunyai *likelihood* paling besar. Pada indeks tersebut akan dicari dalam *database* dengan indeks *database* yang sudah disimpan pada proses pelatihan. Itu akan menampilkan dari jenis citra aksara sesuai dengan kelas pada aksara Incung. Gambar 3.4 sebagai proses pengujian dalam menemukan nilai maksimum pada *likelihood*.



Gambar 3.4 Diagram Tahap Pengujian

Langkah-langkah dalam algoritma *Viterbi* untuk menentukan barisan *state* terbaik yaitu:

1. Inisialisasi:

Rumus untuk mencari perubahan nilai probabilitas:

$$\delta_1(i) = \pi_i b_i(O_1) \quad (3.3)$$

$$\psi_1(i) = 0 \quad (3.4)$$

Keterangan:

$\delta$  = Perubahan nilai probabilitas yang optimal

$\pi$  = Distribusi kondisi awal

$b$  = Probabilitas pengamatan

$O$  = Nilai probabilitas yang terbesar

$i$  = Jumlah perulangan yang digunakan

$\psi$  = Fungsi *polygamma*

2. Rekursi:

Rumus perubahan nilai probabilitas yang convergen:

$$\delta_t(i) = \max_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ji}] b_i(O_t) \quad (3.5)$$

$$\psi_t(i) = \operatorname{argmax}_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ji}] \quad (3.6)$$

Keterangan:

$N$  = Jumlah suatu (*sate*) keadaan

$i$  = Jumlah perulangan

$\delta$  = Perubahan nilai probabilitas yang optimal

$\psi$  = fungsi *polygamma*

$a$  = Probabilitas kondisi transisi

$b$  = Probabilitas pengamatan

$O$  = Nilai probabilitas terbesar

3. Terminasi:

Rumus untuk menentukan probabilitas yang optimal:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (3.7)$$

$$q_T^* = \operatorname{argmax}_{1 \leq i \leq N} [\delta_T(i)] \quad (3.8)$$

Keterangan:

$P^*$  = Nilai probabilitas optimal

$i$  = Nilai pada probabilitas

$q^*$  = kondisi pada deret optimal

$T$  = Jumlah observasi

$N$  = Jumlah suatu (*state*) keadaan

$\delta$  = Perubahan pada probabilitas

4. Lintasan (kondisi deret) jejak balik:

Rumus menentukan kondisi deret jejak balik:

$$q_t^* = \Psi_{t+1}(q_{t+1}^*), t = T - 1, T - 2, \dots, 1 \quad (3.9)$$

Keterangan:

$q^*$  = Keadaan nilai probabilitas yang paling optimal

$\psi$  = Fungsi *polygamma*

$T$  = Jumlah observasi

### 3.2.5 Skenario Pengujian

1. Pengujian Data Jamak

Pada penelitian ini hasil ekstraksi ciri yang digunakan adalah *Intensity of Character* dan *Mark Direction*. Pengujian yang akan dilakukan dengan memilih ukuran dari ekstraksi ciri IoC dengan ukuran 3x3, 4x4, dan 5x5 menggunakan nilai *k-fold cross validation* adalah 3 dan 5. Kemudian menggunakan ekstraksi ciri *Mark Direction* pada bagian horizontal dan vertikal seperti yang ada pada Tabel 3.1 dan *Mark Direction* untuk diagonal kanan, diagonal kiri, horizontal dan vertikal.

Tabel 3.1 Skenario Pengujian Data Jamak

Nama Ciri	Ekstraksi Ciri	Jumlah Ciri
Data Ciri 1	IoC (3x3)	9
Data Ciri 2	IoC (4x4)	16
Data Ciri 3	IoC (5x5)	25
Data Ciri 4	<i>Mark Direction</i> (Horz)	9
Data Ciri 5	<i>Mark Direction</i> (Vert)	9
Data Ciri 6	<i>Mark Direction</i> (Dig1, Dig2, Horz, Vert)	36

## 2. Pengujian Data Tunggal

Pada pengujian data tunggal menggunakan satu responden yang mengisi angket aksara Incung yang terdiri dari 53 aksara. Setiap aksara Incung akan dilakukan pengujian menggunakan model HMM yang telah dilakukan pelatihan data. Model HMM yang dipilih berdasarkan dari pengujian data jamak yang memiliki akurasi paling optimal yang akan digunakan untuk pengujian data tunggal.

### 3.3 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang diperlukan adalah MATLAB R2014b digunakan untuk membuat dan menjalankan sistem. Dibutuhkan aplikasi Photoshop 2018 yang digunakan untuk melakukan pemotongan citra hasil pindai dari responden.

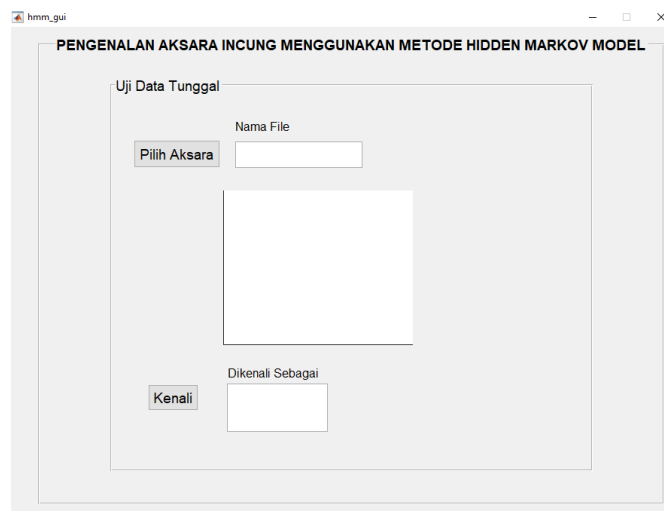
### 3.4 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras dalam menyelesaikan tugas akhir ini adalah sebagai sistem komputer dengan spesifikasi sebagai berikut:

- a. Prosesor : AMD Ryzen 3 3200U
- b. Sistem Operasi : Windows 10 Home Single Language 64-bit
- c. Memory : 8192 Mbyte
- d. Media tampilan : AMD Radeon™ Vega 3 Graphics
- e. Media masukan : Alat pindai (*scanner*)
- f. Media penyimpanan : Hard Disk Drive 1 TB



### 3.5 Desain GUI



Gambar 3.4 Tampilan Pengenalan Aksara Incung

Di halaman ini pengguna dapat memasukan citra aksara Incung yang sudah dilakukan *cropping* dengan menekan tombol “Pilih Aksara”. Citra aksara Incung akan tampil pada kotak citra. Tombol “Kenali” digunakan untuk melakukan proses pengenalan citra aksara Incung.

## BAB IV

### HASIL DAN ANALISIS

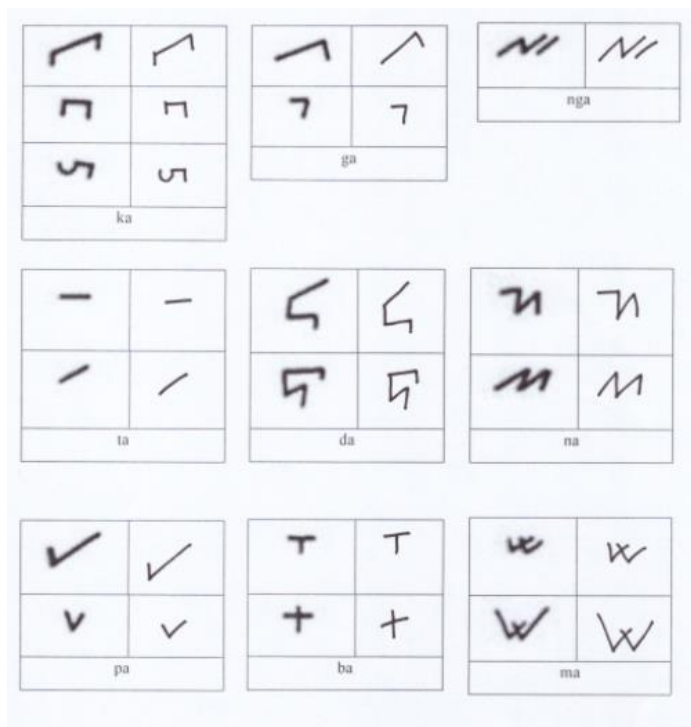
Pada bab ini membahas mengenai tahapan proses pengenalan aksara Incung dengan hasil pengujian pengenalan aksara Incung menggunakan metode *Hidden Markov Model*.

#### 4.1 Data

Data yang digunakan adalah aksara Incung yang berasal dari Jambi. Tahap untuk memperoleh data aksara Incung dilakukan dengan metode kuisioner. Setiap responden akan menuliskan ulang aksara Incung pada kertas A4.

##### 4.1.1 Akusisi Data

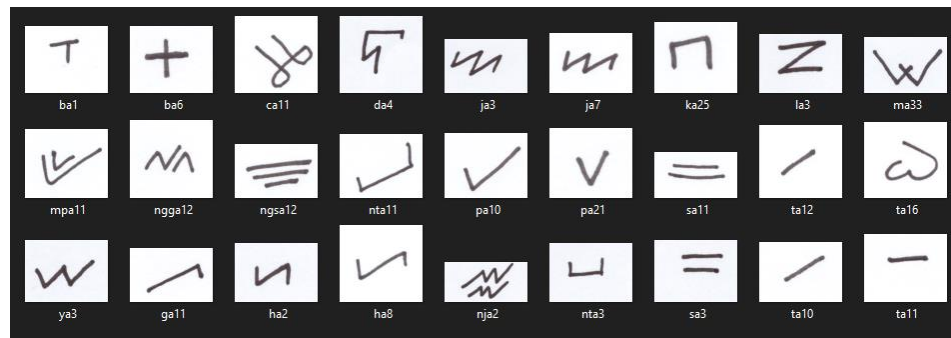
Data aksara yang digunakan berisikan aksara Incung yang diperoleh dari pengisian angket oleh responden. Berikut contoh dari angket dan beserta isinya. Kemudian di pindai menggunakan alat pindai dari seluruh pengumpulan angket aksara Incung.



Gambar 4.1 Contoh Angket Aksara Incung

#### 4.1.2 Cropping

Pada tahap ini dilakukan *cropping* pada citra aksara Incung hasil pindai. Hal ini dilakukan secara manual menggunakan perangkat lunak Adobe Photoshop CC 2018. Hasil awal pindai citra aksara Incung terdiri dari kumpulan beberapa aksara Incung kemudian dilakukan pemotongan menjadi bagian-bagian dari satu aksara.



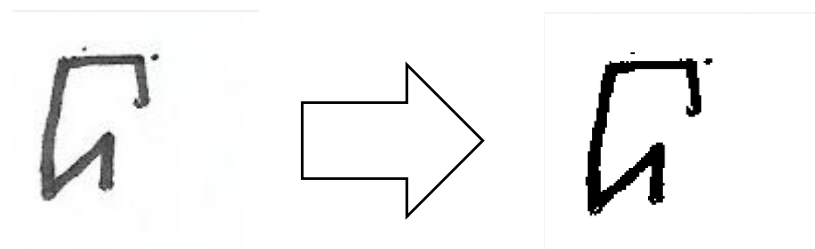
Gambar 4.2 Data Citra Hasil *Cropping*

#### 4.2 Implementasi *Preprocessing*

*Preprocessing* adalah proses awal dalam membuat sistem pengenalan aksara, ini bertujuan untuk memastikan data yang diolah dalam keadaan bersih. Seperti terdapat *noise* pada citra, tulisan yang tidak jelas, dan masih banyak hal yang dipersiapkan agar data siap untuk diolah.

##### 4.2.1 Binerisasi Citra

Binerisasi merupakan proses untuk mengubah citra input menjadi citra biner yang terdiri dari hitam dan putih. Fungsi yang akan digunakan dalam proses binerisasi menggunakan fungsi *im2bw* pada MATLAB, citra aksara yang sudah dilakukan *cropping* kemudian diubah menjadi citra hitam putih yang memiliki nilai piksel 0 dan 1.

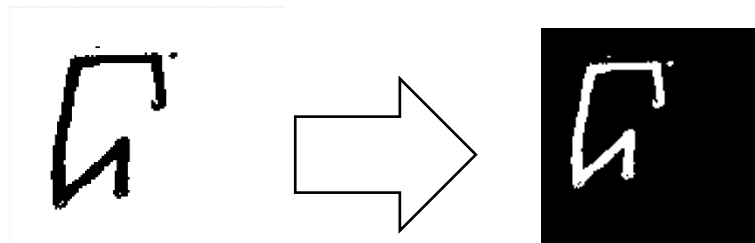


Gambar 4.3 Proses Binerisasi Citra Aksara Da

#### 4.2.2 Reduksi Derau

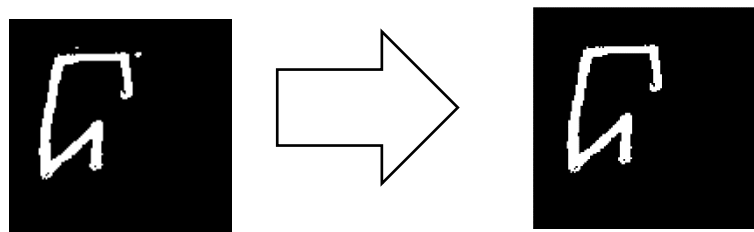
Dilakukan proses reduksi derau sebagai upaya untuk membuat citra data bersih dari *noise*. Jika hal ini tidak dilakukan pada citra yang terdapat *noise* akan mempengaruhi saat dilakukan proses klasifikasi.

Pada Gambar 4.4 proses *invers* citra aksara da sebelum dilakukan proses filtering terlebih dahulu diubah menjadi *background* hitam dan objek putih. Pada proses ini menggunakan *toolbox* dari MATLAB yaitu fungsi *imcomplement*.



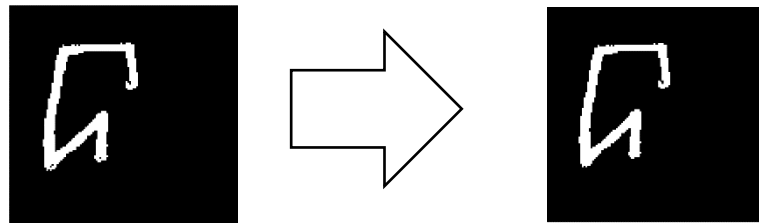
Gambar 4.4 Proses *Invers* Citra Aksara Da

Pada Gambar 4.5 merupakan proses menghilangkan derau menggunakan *toolbox* dari MATLAB yaitu fungsi *bwareaopen* setelah dilakukan proses *invers*.



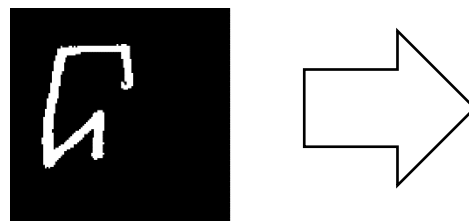
Gambar 4.5 Proses Menghilangkan Derau Aksara Da

Pada Gambar 4.6 merupakan proses mengisi bagian objek yang berlubang menggunakan *toolbox* dari MATLAB yaitu fungsi *imfill*.



Gambar 4.6 Hasil Citra Proses *Imfill* Citra Aksara Da

Pada Gambar 4.7 merupakan proses *invers* atau mengembalikan citra menjadi *background* putih dan objek hitam. Ini menggunakan *toolbox* dari MATLAB yaitu fungsi *imcomplement*.



Gambar 4.7 Proses *Invers* Citra Aksara Da

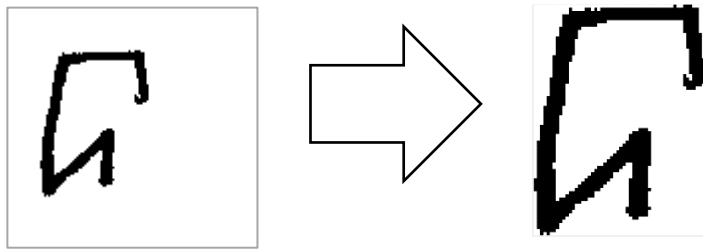
#### 4.2.3 Segmentasi

Pada Gambar 4.8, citra aksara memiliki tepi pada bagian sisi kiri, atas, dan bawah. Proses reduksi derau sudah dilakukan, maka citra aksara tersebut perlu dilakukan segmentasi untuk menghilangkan tepi-tepi batas dari citra. Tujuan dalam penghilangan tepi pada aksara ini berpengaruh pada proses ekstraksi ciri. Tepi-tepi pada citra aksara ini akan terkena proses ekstraksi ciri, hal ini akan mempengaruhi proses klasifikasi. Pada proses segmentasi ini dilakukan dengan menggunakan metode profil proyeksi.

Proyeksi dilakukan dua kali secara horizontal dan vertikal. Proyeksi secara horizontal dilakukan untuk mengetahui

banyaknya piksel putih yang ada pada objek horizontal. Proyeksi vertikal dilakukan untuk mengetahui banyaknya piksel putih pada objek vertikal.

Fungsi ini digunakan untuk menghilangkan batas tepi dari setiap citra. Sehingga citra yang akan digunakan hanya citra aksara data yang dibutuhkan.



Gambar 4.8 Proses Profil Proyeksi Citra Aksara Da

#### 4.2.4 *Resize Citra*

Tahap selanjutnya adalah mengubah ukuran citra menjadi ukuran 60x60 piksel. Tujuan dilakukan *resize* supaya mempermudah dalam proses ekstraksi ciri. Pada tahap ini menggunakan *toolbox* dari MATLAB yaitu fungsi *imresize*.



Gambar 4.9 *Resize Citra Aksara Da*

#### 4.2.5 Penipisan Citra

Proses penipisan citra aksara dilakukan menggunakan fungsi *Rosenfeld* yang membuat ketebalan pada citra aksara menjadi 1 piksel.



Gambar 4.10 Penipisan Citra Aksara Da

#### 4.3 Implementasi Ekstraksi Ciri

Pada proses ekstraksi ciri menggunakan metode *Intensity of Character* (IoC) dan *Mark Direction*. Metode ekstraksi ciri IoC digunakan untuk menghitung nilai piksel hitam atau nol. Metode ekstraksi ciri *Mark Direction* digunakan untuk menghitung nilai horizontal, vertikal, diagonal ke kanan, diagonal ke kiri dari setiap segmen yang memiliki ukuran 20 x 20 piksel. Data aksara Incung memiliki 53 kelas. Setiap kelas akan diberikan nama dengan angka, hal ini untuk mempermudah dalam proses pengujian atau pengenalan aksara Incung.

Tabel 4.1 Kelas pada Aksara Incung

Aksara	Kelas	Jumlah
Ba	1	90
	2	90
Ca	3	90
	4	90
	5	90
Da	6	90
	7	90
Ga	8	90
	9	90
A(ha)	10	90
	11	90
Ha	12	90

	13	90
Ja	14	90
	15	90
Ka	16	90
	17	90
	18	90
La	19	90
	20	90
Ma	21	90
	22	90
Mba	23	90
Mpa	24	90
	25	90
Na	26	90
	27	90
Nca	28	90
Nda	29	90
	30	90
Nga	31	90
Ngga	32	90
	33	90
	34	90
Ngsa	35	90
Nja	36	90
Nta	37	90
	38	90
Nya	39	90
	40	90
Pa	41	90
	42	90
Ra	43	90
	44	90
Sa	45	90
	46	90
Ta	47	90
	48	90
Wa	49	90
	50	90
Ya	51	90
	52	90
	53	90



Dari Tabel 4.1 merupakan pelabelan dari data citra aksara Incung dengan memberikan nama label angka dari 1 sampai 53. Setiap masing-masing data aksara Incung memiliki 90 data citra aksara. Pada tahap ini akan melakukan skenario pengujian berdasarkan Tabel 3.1 setiap tahap masing-masing data dilakukan ekstraksi ciri sesuai dengan tabel tersebut. Pada Tabel 3.1 akan melakukan skenario pengujian dengan data ciri 1, data ciri 2, data ciri 3, data ciri 4, data ciri 5 dan data ciri 6.

#### 4.3.1 Data Ciri 1

Pada data ciri 1 menggunakan ekstraksi ciri IoC ukuran 3x3. Ketika dalam piksel memenuhi piksel berwarna 0 (hitam) maka akan bertambah 1.

Tabel 4.2 Data Ciri 1

4770x10 double

	1	2	3	4	5	6	7	8	9
1	22	0	0	29	20	21	29	0	0
2	22	0	0	31	20	21	28	0	0
3	23	0	0	43	20	21	34	0	0
4	32	0	0	32	20	23	28	0	0
5	23	0	0	32	20	20	32	0	0
6	22	0	0	29	20	22	29	0	0
7	21	0	0	30	20	15	23	0	8
8	21	0	0	32	20	22	30	0	0
9	21	0	0	24	20	21	32	0	0
10	25	0	0	28	20	22	27	0	0
11	5	18	0	24	20	21	30	0	0
12	25	0	0	30	20	20	33	0	0
13	23	0	0	31	20	22	26	0	0
14	23	0	0	29	20	21	29	0	0
15	23	1	0	28	20	23	33	0	0

#### 4.3.2 Data Ciri 2

Pada data ciri 2 menggunakan ekstraksi ciri IoC ukuran 4x4 yang menghasilkan 16 segmen ciri.

Tabel 4.3 Data Ciri 2

4770x17 double

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	8	9	0	0	15	0	0	0	19	15	15	16	24	0	0	0
2	17	0	0	0	15	10	15	13	21	5	0	3	23	0	0	0
3	18	0	0	0	22	15	15	4	31	0	0	12	24	0	0	0
4	26	0	0	0	23	15	15	18	15	0	0	0	20	3	0	0
5	17	1	0	0	15	0	0	0	22	15	15	15	27	0	0	0
6	16	1	0	0	15	11	15	11	19	4	0	6	24	0	0	0
7	16	0	0	0	15	0	0	0	20	15	15	18	18	0	0	0
8	16	0	0	0	15	0	0	0	22	15	15	17	25	0	0	0
9	0	16	0	0	0	15	0	0	14	15	15	16	27	0	0	0
10	20	0	0	0	15	0	0	0	18	15	15	17	22	0	0	0
11	0	18	0	0	0	15	0	0	14	15	15	16	25	0	0	0
12	18	2	0	0	15	0	0	2	20	15	15	13	28	0	0	0
13	18	0	0	0	15	0	0	0	21	15	15	17	21	0	0	0
14	18	0	0	0	19	15	15	16	15	0	0	0	24	0	0	0
15	2	17	0	0	18	15	15	18	15	0	0	0	28	0	0	0

### 4.3.3 Data Ciri 3

Pada data ciri 3 menggunakan ekstraksi ciri IoC ukuran 5x5 yang menghasilkan 25 segmen ciri.

Tabel 4.4 Data Ciri 3

10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	12	0	0	0	0	13	12	12	12	13	21	0	0	0	0
0	15	12	12	12	13	12	0	0	0	0	20	0	0	0	0
0	20	12	12	12	13	29	0	0	0	0	16	0	0	0	0
13	12	0	0	0	2	7	5	0	0	0	3	17	0	0	0
0	16	12	12	12	12	12	0	0	0	0	24	0	0	0	0
0	13	12	12	12	14	12	0	0	0	0	21	0	0	0	0
0	12	0	0	0	0	14	12	12	12	15	15	0	0	0	0
0	16	12	12	12	13	12	0	0	0	1	22	0	0	0	0
0	0	20	12	12	13	10	2	0	0	0	24	0	0	0	0
0	12	12	12	12	14	12	0	0	0	0	19	0	0	0	0
0	0	20	12	12	13	8	4	0	0	0	22	0	0	0	0
0	14	12	12	12	12	12	0	0	0	0	25	0	0	0	0
0	15	12	12	12	13	12	0	0	0	1	18	0	0	0	0
0	13	12	12	12	13	12	0	0	0	0	21	0	0	0	0
0	12	12	12	12	15	12	0	0	0	0	25	0	0	0	0
12	13	2	n	n	n	13	n	n	n	n	21	n	n	n	n

### 4.3.4 Data Ciri 4

Pada data ciri 4 menggunakan ekstraksi ciri *Mark Direction* bagian horizontal ukuran 3x3. Ini akan menghasilkan 9 segmen ciri.

Tabel 4.5 Data Ciri 4

4770x10 double

	1	2	3	4	5	6	7	8	9	10
1	13	9	10	0	0	0	0	0	0	1
2	13	11	12	0	0	0	0	0	0	1
3	10	10	9	0	0	0	0	0	0	1
4	9	8	12	0	0	0	0	0	0	1
5	6	12	12	0	0	0	0	0	0	1
6	13	9	8	0	0	0	0	0	0	1
7	11	11	10	0	0	0	0	0	0	1
8	12	12	10	0	0	0	0	0	0	1
9	13	9	5	0	0	0	0	0	0	1
10	13	9	7	0	0	0	0	0	0	1
11	0	7	4	7	0	0	0	0	0	1
12	9	11	10	0	0	0	0	0	0	1

### 4.3.4 Data Ciri 5

Pada data ciri 5 menggunakan ekstraksi ciri *Mark Direction* bagian vertikal ukuran 3x3. Ini akan menghasilkan 9 segmen ciri.

Tabel 4.6 Data Ciri 5

4770x10 double

	1	2	3	4	5	6	7	8	9	10
1	0	7	3	0	16	0	0	13	0	1
2	0	10	1	0	18	0	0	13	0	1
3	0	9	5	0	14	0	0	10	0	1
4	2	6	0	0	18	0	0	15	0	1
5	0	10	2	0	18	0	0	17	0	1
6	0	6	1	0	18	0	0	15	0	1
7	0	7	0	0	17	0	0	6	0	1
8	0	9	0	0	18	0	0	10	0	1
9	0	2	1	0	16	0	0	15	0	1
10	0	5	1	0	12	0	0	17	0	1
11	0	0	1	0	18	0	0	17	0	1
12	0	7	2	0	16	0	0	14	0	1

#### 4.3.4 Data Ciri 6

Pada data ciri 6 menggunakan ekstraksi ciri *Mark Direction* yang terdiri dari dig1, dig2, horz, vert yang menghasilkan 36 segemen ciri.

Tabel 4.7 Data Ciri 6

	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	9	10	0	0	0	0	0	0	7	3	0	16	0	0	13	0
2	11	12	0	0	0	0	0	0	10	1	0	18	0	0	13	0
3	10	9	0	0	0	0	0	0	9	5	0	14	0	0	10	0
4	8	12	0	0	0	0	0	2	6	0	0	18	0	0	15	0
5	12	12	0	0	0	0	0	0	10	2	0	18	0	0	17	0
6	9	8	0	0	0	0	0	0	6	1	0	18	0	0	15	0
7	11	10	0	0	0	0	0	0	7	0	0	17	0	0	6	0
8	12	10	0	0	0	0	0	0	9	0	0	18	0	0	10	0
9	9	5	0	0	0	0	0	0	2	1	0	16	0	0	15	0
10	9	7	0	0	0	0	0	0	5	1	0	12	0	0	17	0
11	7	4	7	0	0	0	0	0	0	1	0	18	0	0	17	0
12	11	10	0	0	0	0	0	0	7	2	0	16	0	0	14	0
13	11	10	0	0	0	0	0	0	10	0	0	16	0	0	13	0
14	7	11	0	0	0	0	0	0	5	1	0	14	0	0	10	0
15	7	12	0	0	0	0	0	0	5	4	0	14	0	0	17	0
16	9	11	0	0	0	0	0	0	2	0	0	13	0	0	15	0
17	14	13	0	0	0	0	0	0	8	1	0	18	0	0	14	0
18	11	11	0	0	0	0	0	0	11	0	0	16	0	0	15	0
19	13	5	0	0	0	0	0	0	5	0	0	16	0	0	12	0
20	8	9	0	0	0	0	0	0	4	1	0	18	0	0	16	0
21	11	11	0	0	0	0	0	0	7	0	0	14	0	0	16	0

#### 4.4 Implementasi Pembagian Data Pelatihan dan Pengujian

Pada tahap ini setelah dilakukan proses *preprocessing* dan ekstraksi ciri maka data dapat dibagi. Data hasil ekstraksi ciri di bagi menjadi data *training* (data latih) dan data *testing* (data uji). Pembagian data latih dan data uji dilakukan berdasarkan dengan nilai *fold* yang akan digunakan untuk menghitung akurasi menggunakan *k-fold cross validation*.


Tabel 4.8 Pembagian Data Latih 3 fold

	1	2	3
1	3180x10 do...	3180x10 do...	3180x10 do...
2			

Tabel 4.9 Pembagian Data Latih 3 fold


	1	2	3
1	1590x10 do...	1590x10 do...	1590x10 do...
2			

Tabel 4.10 Pembagian Data Latih 5 *fold*

 1x5 [cell](#)

	1	2	3	4	5
1	3816x19 do...	3816x19 do...	3816x19 do...	3816x19 do...	3816x19 do...
2					

Tabel 4.11 Pembagian Data Uji 5 *fold*

 1x5 [cell](#)

	1	2	3	4	5
1	954x10 dou...	954x10 dou...	954x10 dou...	954x10 dou...	954x10 dou...
2					

#### 4.5 Implementasi Klasifikasi *Hidden Markov Model*

Pada tahap ini terdiri dari beberapa langkah yang perlu dilakukan dalam implementasi pengenalan aksara Incung menggunakan metode HMM. Setelah data yang telah di ekstraksi ciri dan dilakukan proses *cross validation* sebagai evaluasi dari penggunaan sistem pengenalan menggunakan HMM. Dilakukan proses inisialisasi parameter pada HMM, kemudian tahapan pelatihan untuk menemukan parameter yang maksimal. Setelah itu dilakukan tahapan pengujian, hasil dari pelatihan yang akan digunakan dalam tahap pengujian untuk melihat tingkat akurasi dari pengenalan aksara Incung.

##### 4.5.1 Inisialisasi parameter *Hidden Markov Model*

Tahap sebelum melakukan proses pelatihan menggunakan metode *Hidden Markov Model* terlebih dahulu menginisialisasi parameter yang akan digunakan pada dalam proses klasifikasi. Setiap data citra yang telah dibagi masing-masing akan dilakukan pengujian menggunakan *state* 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, dan 30.

1. Menentukan parameter  $\pi$  dengan Q adalah jumlah *state* yang akan digunakan:

$$\text{phi} = \text{normalise}(\text{rand}(Q, 1));$$

2. Menentukan parameter nilai probabilitas transisi (A) nilai Q sebagai jumlah *state* yang digunakan.

```
A = mk_stochastic(rand(Q,Q));
```

3. Menentukan parameter nilai probabilitas emisi (B) dengan menginisialisasi nilai *mean* dan *sigma*.

```
[mu0, sigma0] = mixgauss_init(Q * M,
cell2mat(data), cov_type);
mu0 = reshape(mu0, [O Q M]);
sigma0 = reshape(sigma0, [O O Q M]);
mixmat0 = mk_stochastic(rand(Q, M));
```

#### 4.5.2 Tahap Pelatihan *Hidden Markov Model*

Pada tahap pelatihan ini digunakan program HMM *toolbox* yang dikembangkan oleh Kevin Murphy, 2005 [Murphy]. Hasil pembagian data pelatihan akan digunakan untuk melakukan tahap pelatihan menggunakan algoritma *Baum-Welch*. Ini digunakan untuk membuat model sesuai dengan pembagian data latih.

```
[LL, prior1, transmat1, mu1, Sigma1, mixmat1] =
... mhmm_em(dataLatih, prior0, transmat0, mu0,
Sigma0, mixmat0);
```

Fungsi yang digunakan tersebut untuk menemukan estimasi parameter *likelihood* maksimum menggunakan algoritma *Baum-Welch*.

#### 4.5.3 Tahap Pengujian *Hidden Markov Model*

Pada tahap pengujian ini dilakukan perbandingan pada setiap data *testing* yang telah dilakukan. Setiap data ini akan dilakukan klasifikasi deret, ini digunakan untuk mengevaluasi *log-likelihood* dari setiap model yang sudah dilatih dengan data pengujian.

```
loglik = mhmm_logprob(datauji, prior, transmat, mu,
Sigma, mixmat);
```

Fungsi yang digunakan untuk mencari *log-likelihood* dari data pengujian dan model pelatihan sebelumnya.

## 4.6 Pengujian dan Evaluasi

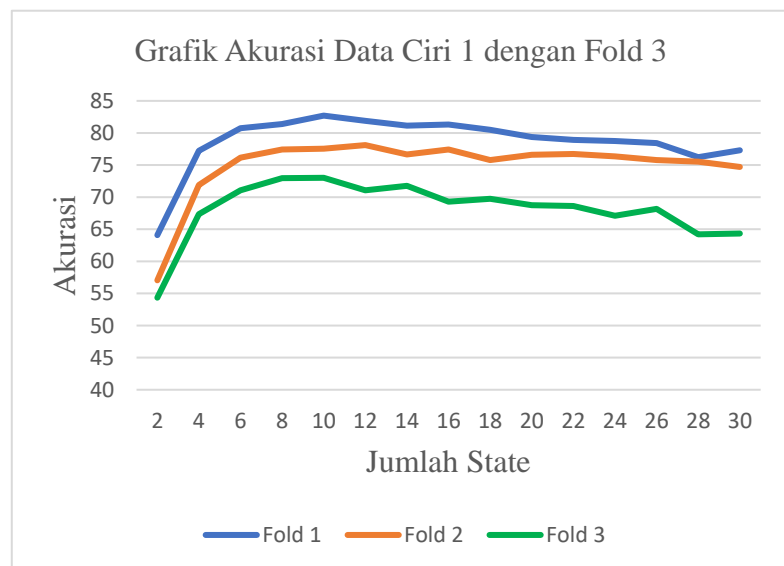
Sesuai dengan skenario pengujian yang akan dilakukan dengan ukuran citra *resize* 60x60 piksel dilakukan ekstraksi ciri menggunakan IoC dengan ukuran 3x3, 4x4, dan 5x5. Kemudian melakukan ekstraksi ciri *Mark Direction* (horizontal). Ciri yang digunakan dengan ekstraksi ciri *Mark Direction* (vertikal). Kemudian menggunakan data ekstraksi ciri *Mark Direction* terdiri dari diagonal ke kanan, diagonal ke kiri, horizontal, dan vertikal.

Pada proses klasifikasi data latih dan data uji menggunakan *k-fold cross validation*. Menggunakan nilai *k-fold cross validation* adalah 3 dan 5. Sehingga terdapat 6 jenis pengujian data citra aksara yang akan dilakukan.

### 4.6.1 Pengujian Data Ciri 1

#### a. Percobaan dengan 3-fold cross validation

Hasil pengujian dari data ciri 1 menggunakan ekstraksi ciri IoC ukuran 3x3. Menggunakan 3-fold cross validation dengan jumlah *state* dari 2 hingga 30 dengan kenaikan 2.



Gambar 4.11 Grafik Akurasi dengan IoC = 3 dan *fold* 3

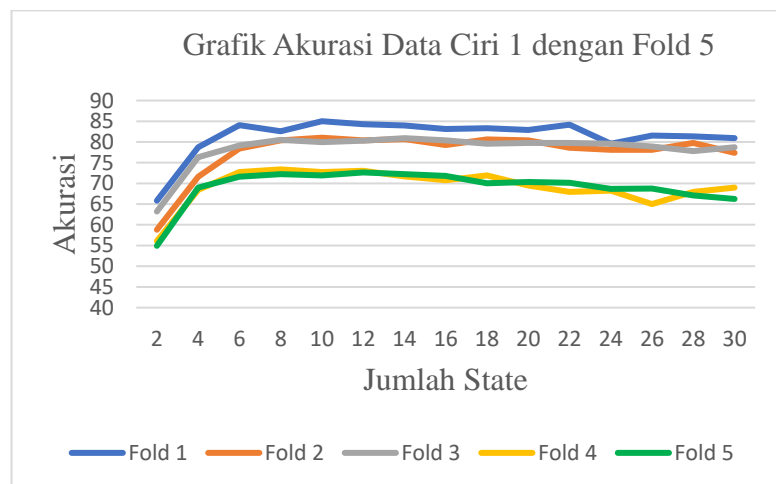
Tabel 4.12 Hasil Pengujian Data Ciri 1 dengan 3 *fold*

State	Persentase Akurasi ( <i>Fold</i> ke)		
	<i>Fold 1</i>	<i>Fold 2</i>	<i>Fold 3</i>
2	64.08805	57.04403	54.33962
4	77.2327	71.88679	67.35849
6	80.75472	76.16352	71.06918
8	81.38365	77.42138	72.95597
10	82.7044	77.54717	73.01887
12	81.88679	78.11321	71.06918
14	81.13208	76.66667	71.76101
16	81.32075	77.42138	69.30818
18	80.50314	75.78616	69.74843
20	79.37107	76.60377	68.74214
22	78.93082	76.72956	68.61635
24	78.74214	76.3522	67.10692
26	78.42767	75.78616	68.1761
28	76.22642	75.53459	64.21384
30	77.2956	74.71698	64.33962

Dari Tabel 4.12 menunjukkan hasil akurasi paling besar pada *fold* 1 di *state* 10 dengan akurasi 82.70%.

b. Percobaan dengan 5-fold cross validation

Hasil pengujian dari data ciri 1 menggunakan ekstraksi ciri IoC ukuran 3x3. Menggunakan 5-fold cross validation dengan jumlah *state* dari 2 hingga 30 dengan kenaikan 2.



Gambar 4.12 Grafik Akurasi dengan IoC = 3 dan *fold* 5  
Tabel 4.13 Hasil Pengujian Data Ciri 1 dengan 5 *fold*

State	Persentase Akurasi ( <i>Fold</i> ke)				
	<i>Fold</i> 1	<i>Fold</i> 2	<i>Fold</i> 3	<i>Fold</i> 4	<i>Fold</i> 5
2	65.8281	58.805	63.2075	56.0797	54.9266
4	78.7212	71.5933	76.3103	68.4486	68.9727
6	84.0671	78.4067	79.1405	72.7463	71.5933
8	82.5996	80.3983	80.5031	73.3753	72.2222
10	85.0105	81.0273	79.979	72.7463	71.9078
12	84.2767	80.3983	80.2935	73.0608	72.6415
14	83.9623	80.7128	80.9224	71.6981	72.2222
16	83.1237	79.2453	80.3983	70.7547	71.8029
18	83.3333	80.608	79.5597	71.9078	70.021
20	82.914	80.3983	79.7694	69.4969	70.3354
22	84.1719	78.6164	79.7694	67.9245	70.1258
24	79.5597	78.0922	79.5597	68.239	68.6583
26	81.5514	78.0922	78.9308	64.9895	68.7631
28	81.3417	79.7694	77.7778	67.9245	67.086
30	80.9224	77.3585	78.7212	68.9727	66.2474

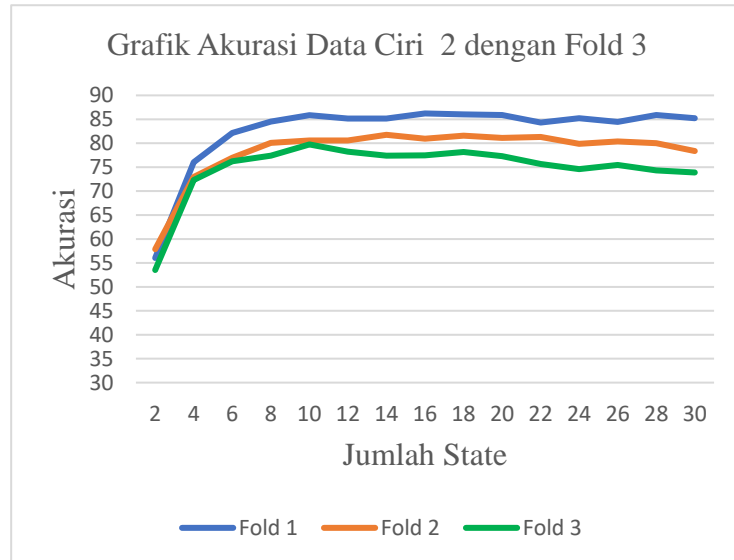
Dari Tabel 4.13 menunjukkan hasil akurasi paling besar pada *fold* 1 di *state* 10 dengan akurasi 85.01%.



#### 4.6.2 Pengujian Data Ciri 2

##### a. Percobaan dengan 3-fold cross validation

Hasil pengujian dari data ciri 2 menggunakan ekstraksi ciri IoC ukuran 4x4. Menggunakan 3-fold cross validation dengan jumlah *state* dari 2 hingga 30 dengan kenaikan 2.



Gambar 4.13 Grafik Akurasi dengan IoC = 4 dan *fold* 3

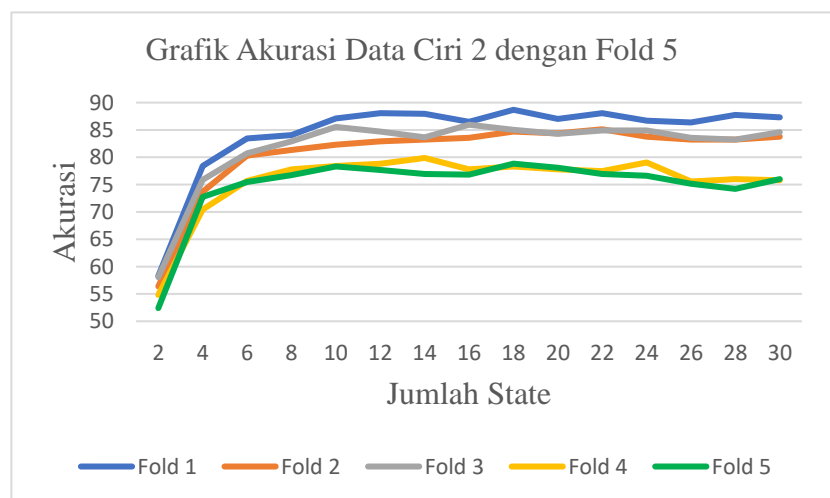
Tabel 4.14 Hasil Pengujian Data Ciri 2 dengan 3 *fold*

State	Persentase Akurasi ( <i>Fold</i> ke)		
	<i>Fold 1</i>	<i>Fold 2</i>	<i>Fold 3</i>
2	56.0377	57.8616	53.522
4	76.0377	72.956	72.327
6	82.1384	76.9811	76.2264
8	84.5283	80.0629	77.4214
10	85.8491	80.566	79.7484
12	85.1572	80.566	78.239
14	85.1572	81.761	77.4214
16	86.2264	80.9434	77.4843
18	86.0377	81.5723	78.1761
20	85.9119	81.1321	77.2956
22	84.3396	81.3208	75.6604
24	85.2201	79.8742	74.5912
26	84.4654	80.3774	75.4717
28	85.9119	80	74.3396
30	85.2201	78.3648	73.8994

Dari Tabel 4.14 menunjukkan hasil akurasi paling besar pada *fold* 1 di *state* 16 dengan akurasi 86.22%.

b. Percobaan dengan 5-*fold cross validation*

Hasil pengujian dari data ciri 2 menggunakan ekstraksi ciri IoC ukuran 4x4. Menggunakan 5-*fold cross validation* dengan jumlah *state* dari 2 hingga 30 dengan kenaikan 2.

Gambar 4.14 Grafik Akurasi dengan IoC = 4 dan *fold* 5

Tabel 4.15 Hasil Pengujian Data Ciri 2 dengan 5 *fold*

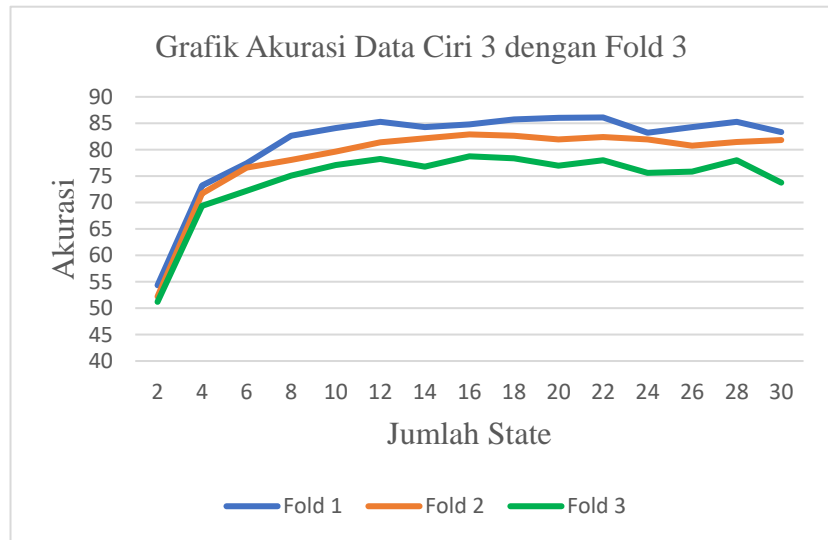
State	Persentase Akurasi ( <i>Fold</i> ke)				
	<i>Fold 1</i>	<i>Fold 2</i>	<i>Fold 3</i>	<i>Fold 4</i>	<i>Fold 5</i>
2	58.2809	56.3941	58.0713	54.8218	52.4109
4	78.4067	73.6897	75.891	70.4403	72.7463
6	83.4382	80.2935	80.7128	75.6813	75.4717
8	84.0671	81.3417	82.914	77.7778	76.7296
10	87.1069	82.2851	85.5346	78.4067	78.3019
12	88.0503	82.914	84.696	78.826	77.673
14	87.9455	83.2285	83.6478	79.8742	76.9392
16	86.478	83.543	85.9539	77.7778	76.8344
18	88.6792	84.696	85.0105	78.3019	78.826
20	87.0021	84.3816	84.2767	77.7778	78.0922
22	88.0503	85.1153	84.9057	77.4633	76.9392
24	86.6876	83.7526	84.9057	79.0356	76.6247
26	86.3732	83.2285	83.543	75.5765	75.1572
28	87.7358	83.2285	83.2285	75.9958	74.2138
30	87.3166	83.7526	84.5912	75.7862	75.9958

Dari Tabel 4.15 menunjukkan hasil akurasi paling besar pada *fold* 1 di *state* 18 dengan akurasi 88.67%.

### 4.6.3 Pengujian Data Ciri 3

#### a. Percobaan dengan 3-fold cross validation

Hasil pengujian dari data ciri 3 menggunakan ekstraksi ciri IoC ukuran 5x5. Menggunakan 3-fold cross validation dengan jumlah *state* dari 2 hingga 30 dengan kenaikan 2.



Gambar 4.15 Grafik Akurasi dengan IoC = 5 dan *fold* 3

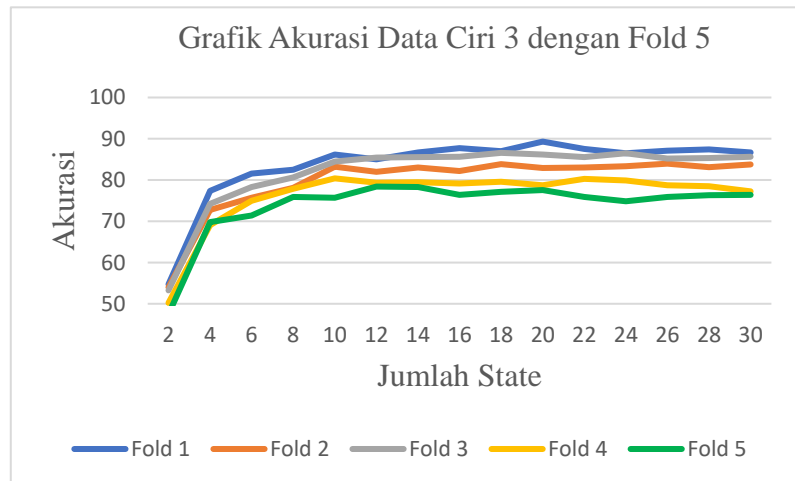
Tabel 4.16 Hasil Pengujian Data Ciri 3 dengan 3 *fold*

State	Persentase Akurasi ( <i>Fold</i> ke)		
	<i>Fold 1</i>	<i>Fold 2</i>	<i>Fold 3</i>
2	54.3396	52.2013	51.195
4	73.2075	71.6981	69.3711
6	77.4214	76.6038	72.2013
8	82.6415	78.0503	75.0943
10	84.0881	79.6226	77.1069
12	85.283	81.3836	78.239
14	84.2767	82.1384	76.7925
16	84.7799	82.8931	78.7421
18	85.7233	82.6415	78.3648
20	86.0377	81.9497	76.9811
22	86.1006	82.3899	77.9874
24	83.2075	81.9497	75.5975
26	84.2767	80.7547	75.8491
28	85.283	81.4465	77.9874
30	83.3333	81.8239	73.7736

Dari Tabel 4.16 menunjukkan hasil akurasi paling besar pada *fold* 1 di *state* 22 dengan akurasi 86.10%.

b. Percobaan dengan *5-fold cross validation*

Hasil pengujian dari data ciri 3 menggunakan ekstraksi ciri IoC ukuran 5x5. Menggunakan *5-fold cross validation* dengan jumlah *state* dari 2 hingga 30 dengan kenaikan 2.



Gambar 4.16 Grafik Akurasi dengan IoC=5 dan *fold* 5

Tabel 4.17 Hasil Pengujian Data Ciri 3 dengan 5 *fold*

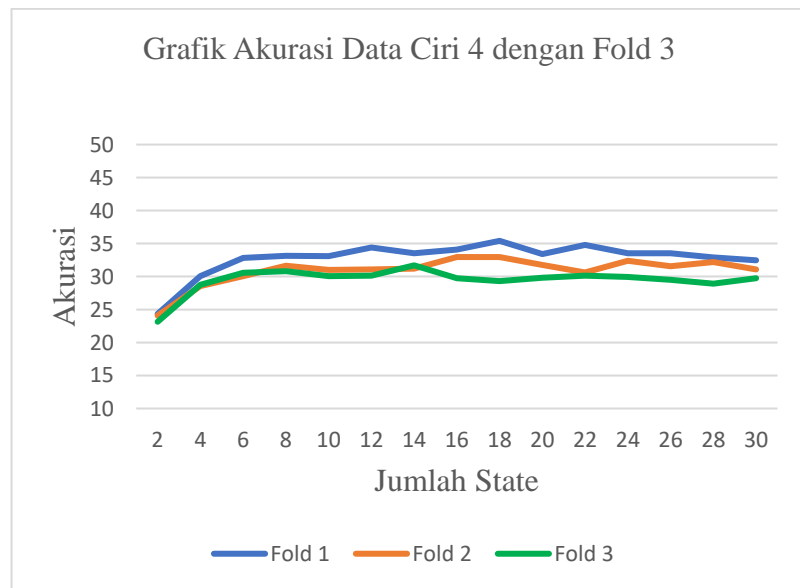
State	Persentase Akurasi ( <i>Fold</i> ke)				
	<i>Fold 1</i>	<i>Fold 2</i>	<i>Fold 3</i>	<i>Fold 4</i>	<i>Fold 5</i>
2	54.717	53.9832	53.2495	50.2096	47.4843
4	77.3585	72.7463	74.2138	69.0776	69.8113
6	81.5514	75.6813	78.3019	74.9476	71.3836
8	82.4948	78.0922	80.608	77.8826	75.891
10	86.1635	83.2285	84.4864	80.3983	75.6813
12	85.0105	81.9706	85.4298	79.3501	78.4067
14	86.6876	83.0189	85.5346	79.4549	78.3019
16	87.7358	82.1803	85.6394	79.1405	76.4151
18	87.0021	83.8574	86.5828	79.5597	77.1488
20	89.3082	82.914	86.1635	78.7212	77.5681
22	87.5262	83.0189	85.5346	80.2935	75.891
24	86.478	83.3333	86.478	79.8742	74.8428
26	87.1069	83.9623	85.2201	78.7212	75.891
28	87.4214	83.1237	85.3249	78.5115	76.3103
30	86.6876	83.7526	85.6394	77.2537	76.4151

Dari Tabel 4.17 menunjukkan hasil akurasi paling besar pada *fold* 1 di *state* 20 dengan akurasi 89.30%.

#### 4.6.4 Pengujian Data Ciri 4

##### a. Percobaan dengan 3-fold cross validation

Hasil pengujian dari data ciri 4 menggunakan ekstraksi ciri *Mark Direction Horizontal*. Menggunakan 3-fold cross validation dengan jumlah *state* dari 2 hingga 30 dengan kenaikan 2.



Gambar 4.17 Grafik Akurasi dengan *Mark Direction Horizontal* dan *fold 3*



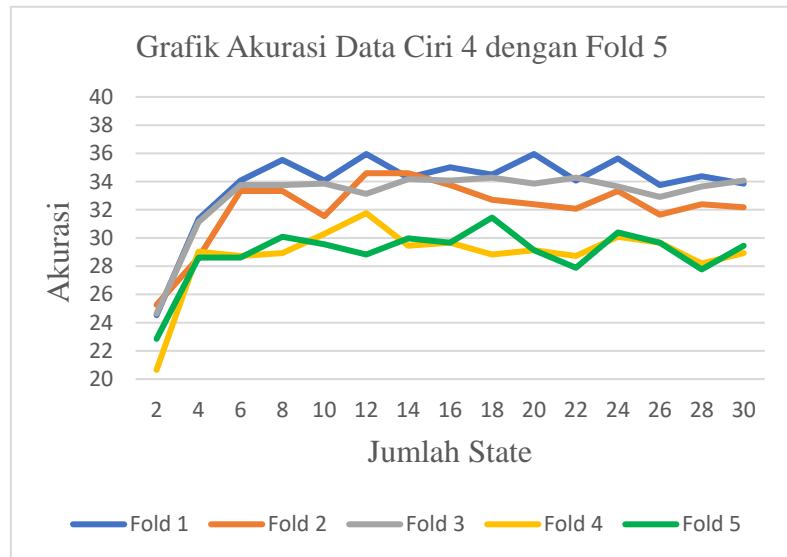
Tabel 4.18 Hasil Pengujian Data Ciri 4 dengan 3 *fold*

State	Persentase Akurasi ( <i>Fold</i> ke)		
	<i>Fold 1</i>	<i>Fold 2</i>	<i>Fold 3</i>
2	24.3396	24.0881	23.1447
4	30.0629	28.5535	28.7421
6	32.8302	30.0629	30.566
8	33.1447	31.6352	30.8176
10	33.0818	31.0063	30.0629
12	34.4025	31.0692	30.1258
14	33.522	31.195	31.6981
16	34.0881	32.956	29.7484
18	35.4088	32.956	29.3082
20	33.3962	31.761	29.8113
22	34.7799	30.6289	30.1258
24	33.522	32.3899	29.9371
26	33.522	31.5723	29.4969
28	32.8931	32.2013	28.9308
30	32.4528	31.0692	29.7484

Dari Tabel 4.18 menunjukkan hasil akurasi paling besar pada *fold* 1 di *state* 18 dengan akurasi 35.40%.

b. Percobaan dengan 5-fold cross validation

Hasil Pengujian dari data ciri 4 menggunakan ekstraksi ciri *Mark Direction* horizontal. Menggunakan 5-fold cross validation dengan jumlah *state* dari 2 hingga 30 dengan kenaikan 2.



Gambar 4.18 Grafik Akurasi dengan *Mark Direction* Horizontal dan *fold 5*

Tabel 4.19 Hasil Pengujian Data Ciri 4 dengan 5 *fold*

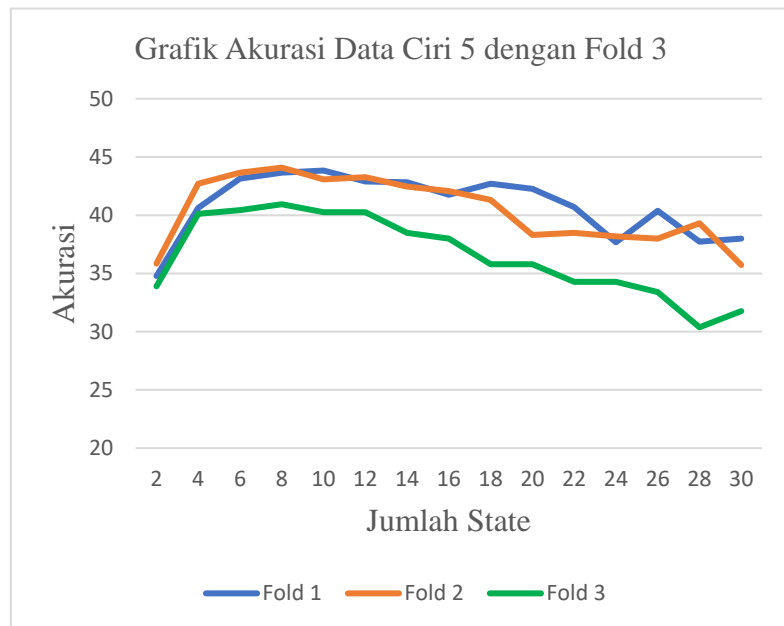
State	Persentase Akurasi ( <i>Fold</i> ke)				
	<i>Fold 1</i>	<i>Fold 2</i>	<i>Fold 3</i>	<i>Fold 4</i>	<i>Fold 5</i>
2	24.5283	25.2621	24.6331	20.6499	22.8512
4	31.3417	28.6164	31.1321	29.0356	28.6164
6	34.0671	33.3333	33.7526	28.7212	28.6164
8	35.5346	33.3333	33.7526	28.9308	30.0839
10	34.0671	31.5514	33.8574	30.2935	29.5597
12	35.9539	34.5912	33.1237	31.761	28.826
14	34.2767	34.5912	34.1719	29.4549	29.979
16	35.0105	33.7526	34.0671	29.6646	29.6646
18	34.4864	32.7044	34.2767	28.826	31.4465
20	35.9539	32.3899	33.8574	29.1405	29.1405
22	34.0671	32.0755	34.2767	28.7212	27.8826
24	35.6394	33.3333	33.6478	30.0839	30.3983
26	33.7526	31.6562	32.914	29.6646	29.6646
28	34.3816	32.3899	33.6478	28.1971	27.7778
30	33.8574	32.1803	34.0671	28.9308	29.4549

Dari Tabel 4.19 menunjukkan hasil akurasi paling besar pada *fold* 1 di *state* 12 dengan akurasi 35.95%.

#### 4.6.5 Pengujian Data Ciri 5

##### a. Percobaan dengan 3-fold cross validation

Hasil Pengujian dari data ciri 5 menggunakan ekstraksi ciri *Mark Direction* vertikal. Menggunakan 3-fold cross validation dengan jumlah *state* dari 2 hingga 30 dengan kenaikan 2.



Gambar 4.19 Grafik Akurasi dengan *Mark Direction* Vertikal dan *fold 3*

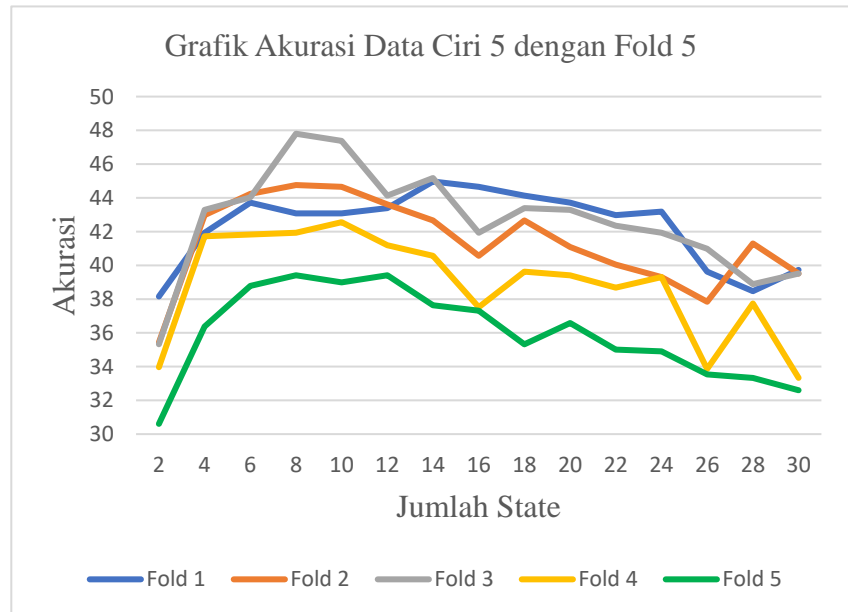
Tabel 4.20 Hasil Pengujian Data Ciri 5 dengan 3 *fold*

State	Persentase Akurasi ( <i>Fold</i> ke)		
	<i>Fold 1</i>	<i>Fold 2</i>	<i>Fold 3</i>
2	34.7799	35.8491	33.8994
4	40.6289	42.7044	40.1258
6	43.1447	43.6478	40.4403
8	43.6478	44.0881	40.9434
10	43.8365	43.0818	40.2516
12	42.8931	43.2704	40.2516
14	42.8302	42.4528	38.4906
16	41.761	42.0755	37.9874
18	42.7044	41.3208	35.7862
20	42.2642	38.3019	35.7862
22	40.6918	38.4906	34.2767
24	37.673	38.1761	34.2767
26	40.3774	37.9874	33.3962
28	37.7358	39.3082	30.3774
30	37.9874	35.7233	31.761

Dari Tabel 4.20 menunjukkan hasil akurasi paling besar pada *fold* 1 di *state* 10 dengan akurasi 43.83%.

b. Percobaan dengan *5-fold cross validation*

Hasil Pengujian dari data ciri 5 menggunakan ekstraksi ciri *Mark Direction* vertikal. Menggunakan *5-fold cross validation* dengan jumlah *state* dari 2 hingga 30 dengan kenaikan 2.



Gambar 4.20 Grafik Akurasi dengan *Mark Direction* Vertikal dan *fold 5*

Tabel 4.21 Hasil Pengujian Data Ciri 5 dengan 5 *fold*

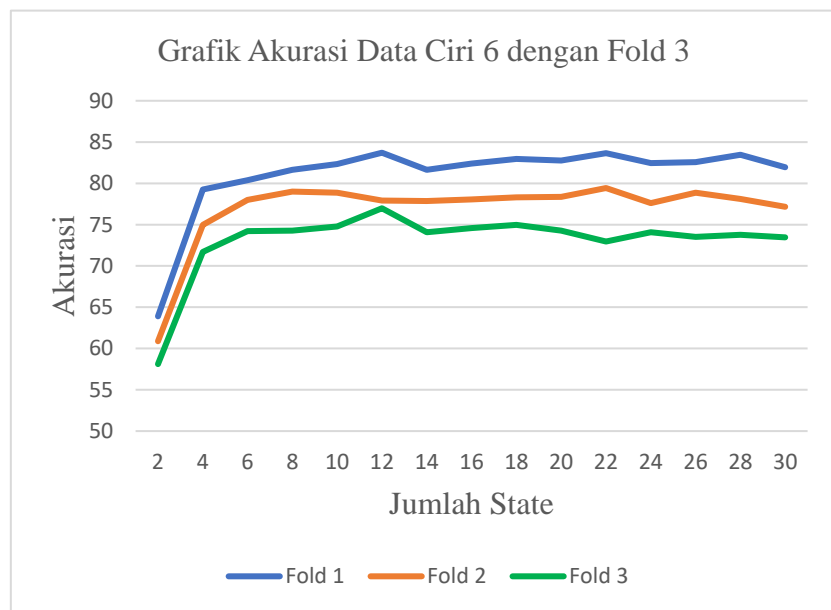
State	Persentase Akurasi ( <i>Fold</i> ke)				
	<i>Fold 1</i>	<i>Fold 2</i>	<i>Fold 3</i>	<i>Fold 4</i>	<i>Fold 5</i>
2	38.1551	35.4298	35.3249	33.9623	30.608
4	41.9287	42.9769	43.2914	41.7191	36.3732
6	43.7107	44.2348	44.0252	41.8239	38.7841
8	43.0818	44.7589	47.7987	41.9287	39.413
10	43.0818	44.6541	47.3795	42.5577	38.9937
12	43.3962	43.6059	44.13	41.195	39.413
14	44.9686	42.6625	45.1782	40.566	37.631
16	44.6541	40.566	41.9287	37.5262	37.3166
18	44.13	42.6625	43.3962	39.6226	35.3249
20	43.7107	41.0901	43.2914	39.413	36.5828
22	42.9769	40.0419	42.348	38.6792	35.0105
24	43.1866	39.3082	41.9287	39.3082	34.9057
26	39.6226	37.8407	40.9853	33.8574	33.543
28	38.4696	41.2998	38.8889	37.7358	33.3333
30	39.7275	39.5178	39.5178	33.3333	32.5996

Dari Tabel 4.21 menunjukkan hasil akurasi paling besar pada *fold* 1 di *state* 14 dengan akurasi 44.96%.

#### 4.6.6 Pengujian Data Ciri 6

##### a. Percobaan dengan 3-fold cross validation

Hasil Pengujian dari data ciri 6 menggunakan ekstraksi ciri *Mark Direction* yang terdiri dari diagonal kiri, diagonal kanan, horizontal, dan vertikal. Menggunakan 3-fold cross validation dengan jumlah *state* dari 2 hingga 30 dengan kenaikan 2.



Gambar 4.21 Grafik Akurasi dengan *Mark Direction* dan *fold 3*



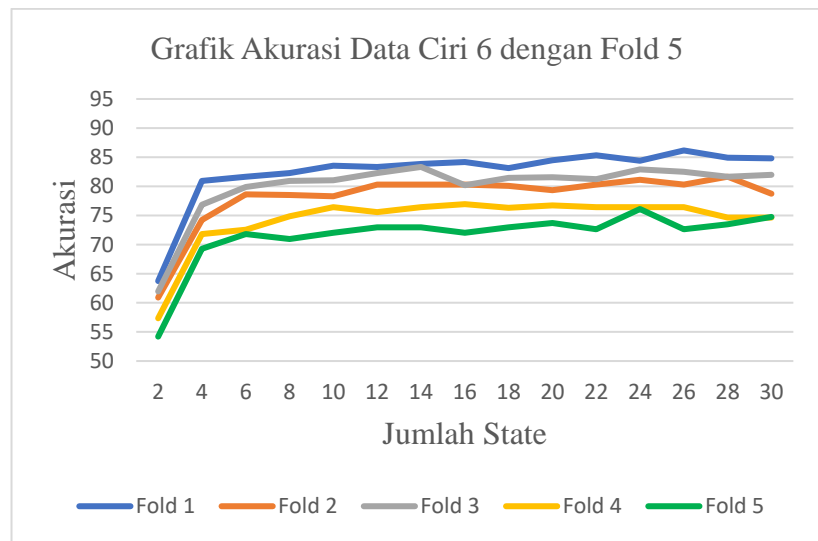
Tabel 4.22 Hasil Pengujian Data Ciri 6 dengan 3 *fold*

State	Persentase Akurasi ( <i>Fold</i> ke)		
	<i>Fold 1</i>	<i>Fold 2</i>	<i>Fold 3</i>
2	63.8994	60.8805	58.1132
4	79.2453	74.9686	71.6981
6	80.3774	77.9874	74.2138
8	81.6352	78.9937	74.2767
10	82.327	78.8679	74.7799
12	83.7107	77.9245	76.9811
14	81.6352	77.8616	74.0881
16	82.3899	78.0503	74.5912
18	82.956	78.3019	74.9686
20	82.7673	78.3648	74.2767
22	83.6478	79.434	72.956
24	82.4528	77.6101	74.0881
26	82.5786	78.8679	73.522
28	83.4591	78.1132	73.7736
30	81.9497	77.1698	73.4591

Dari Tabel 4.22 menunjukkan hasil akurasi paling besar pada *fold* 1 di *state* 12 dengan akurasi 83.71%.

b. Percobaan dengan *5-fold cross validation*

Hasil Pengujian dari data ciri 6 menggunakan ekstraksi ciri *Mark Direction* yang terdiri dari diagonal kiri, diagonal kanan, horizontal, dan vertikal. Menggunakan *5-fold cross validation* dengan jumlah *state* dari 2 hingga 30 dengan kenaikan 2.



Gambar 4.22 Grafik Akurasi dengan *Mark Direction* dan *fold 5*

Tabel 4.23 Hasil Pengujian Data Ciri 6 dengan 5 *fold*

State	Persentase Akurasi ( <i>Fold</i> ke)				
	<i>Fold 1</i>	<i>Fold 2</i>	<i>Fold 3</i>	<i>Fold 4</i>	<i>Fold 5</i>
2	63.7317	60.9015	61.9497	57.3375	54.1929
4	80.9224	74.2138	76.8344	71.8029	69.2872
6	81.6562	78.6164	79.8742	72.5367	71.8029
8	82.2851	78.5115	80.9224	74.8428	70.9644
10	83.543	78.3019	81.0273	76.4151	72.0126
12	83.3333	80.2935	82.2851	75.5765	72.956
14	83.8574	80.2935	83.3333	76.4151	72.956
16	84.1719	80.2935	80.1887	76.9392	72.0126
18	83.1237	80.0839	81.4465	76.3103	72.956
20	84.4864	79.3501	81.5514	76.7296	73.6897
22	85.3249	80.2935	81.2369	76.4151	72.6415
24	84.3816	81.1321	82.914	76.4151	76.1006
26	86.1635	80.2935	82.4948	76.4151	72.6415
28	84.9057	81.6562	81.6562	74.6331	73.4801
30	84.8008	78.7212	81.9706	74.6331	74.7379

Dari Tabel 4.23 menunjukkan hasil akurasi paling besar pada *fold* 1 di *state* 26 dengan akurasi 86.16%.

Hasil dari grafik menunjukkan pemodelan yang memiliki akurasi paling besar terdapat pada data ciri 3 menggunakan ekstraksi ciri IoC dengan ukuran 5x5. Itu terdapat pada percobaan menggunakan nilai *k-fold* 5 pada *fold* 1 dan jumlah *state* 20 yaitu 89,39%. Pemodelan ini yang akan digunakan untuk melakukan pengujian data tunggal pada pengenalan aksara Incung.

Pada penelitian ini, diperoleh waktu proses komputasi dari setiap skenario pengujian dalam pengenalan aksara Incung. Skenario pengujian ini berdasarkan pada Tabel 3.1 skenario pengujian data jamak. Durasi yang dihitung dalam satuan detik sudah termasuk pada tahap pelatihan dan pengujian aksara Incung dengan metode HMM. Berikut ini durasi komputasi dari setiap skenario pengujian:

Tabel 4.24 Durasi Klasifikasi HMM

K-Fold	Skenario Pengujian (detik)					
	Data Ciri 1	Data Ciri 2	Data Ciri 3	Data Ciri 4	Data Ciri 5	Data Ciri 6
Fold 3	8656.88	8418.70	10600.00	8863.51	9979.17	8358.63
Fold 5	10365.29	13587.84	13281.12	12565.20	14877.31	10599.27

Dari Tabel 4.24 durasi waktu yang paling sedikit pada data ciri 6 dengan ekstraksi ciri *Mark Direction* yang terdiri dari diagonal kiri, diagonal kanan, horizontal dan vertikal dengan 3 *fold cross validation* yaitu 8358.63 detik. Sedangkan durasi waktu yang paling banyak pada data ciri 5 dengan ekstraksi ciri *Mark Direction* menggunakan bagian vertikal dengan 3 *fold cross validation* yaitu 14877.31 detik.

Gambar 4.23 ini sebagai implementasi pengenalan aksara Incung menggunakan metode *Hidden Markov Model*. Ini sebagai tampilan yang digunakan dalam proses pengujian data tunggal setiap huruf data aksara Incung yang terdiri dari 53 huruf aksara Incung.



Gambar 4.23 Tampilan Menu Pengujian Data Tunggal

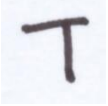
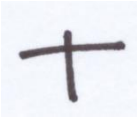
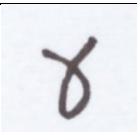
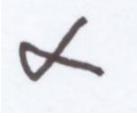

Proses yang dilakukan untuk pengenalan aksara Incung dengan mengklik tombol “Pilih Aksara”. Ini akan diarahkan untuk memilih citra aksara Incung yang akan dilakukan proses pengenalan. Kemudian akan



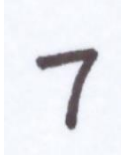
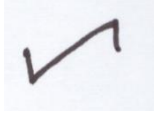
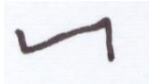
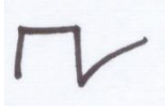
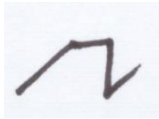
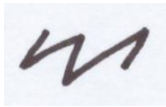
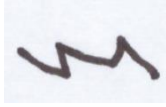

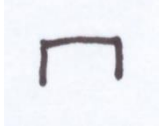
menampilkan hasil dari aksara yang telah dipilih. Ketika tekan tombol “Kenali” akan melakukan proses *preprocessing* pada citra aksara. Hasil *preprocessing* dikenakan ekstraksi ciri IoC dengan ukuran 5x5. Itu akan dilakukan proses pengujian menggunakan Algoritma *Viterbi* menggunakan hasil pemodelan yang telah didapatkan sebelumnya. Hasil dari pengujian tersebut akan menampilkan hasil aksara Incung yang dikenali.

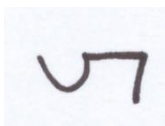
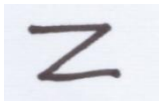

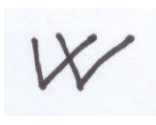
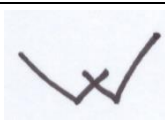


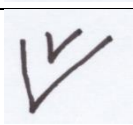
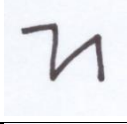
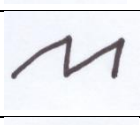
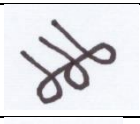
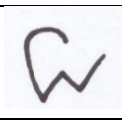
#### 4.6.7 Pengujian Data Tunggal




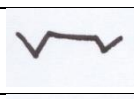

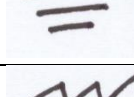
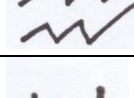
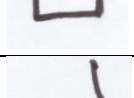
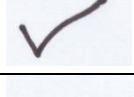
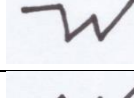
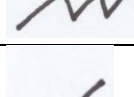
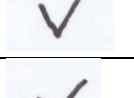
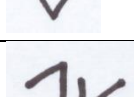
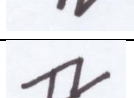
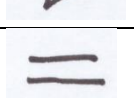
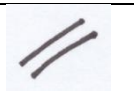

Pada tahap pelatihan diperoleh hasil akurasi terbaik dengan data ciri 3 dengan nilai k sama dengan 5 dengan jumlah *state* 20. Ini yang akan digunakan sebagai pemodelan untuk melakukan pengujian data tunggal aksara Incung.

Tabel 4.25 Hasil Pengujian Data Tunggal





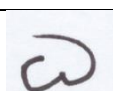
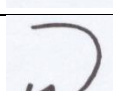
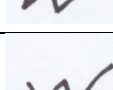
No	Citra Aksara Incung	Huruf Aksara Incung	Hasil Pengenalan Aksara Incung	Status
1		BA	BA	BENAR
2			BA	BENAR
3		CA	CA	BENAR
4			BA	SALAH
5			MA	SALAH
6		DA	DA	BENAR

7			DA	BENAR
8		GA	GA	BENAR
9			GA	BENAR
10		A(HA)	A(HA)	BENAR
11			GA	SALAH
12		HA	HA	BENAR
13			HA	BENAR
14		JA	JA	BENAR
15			JA	BENAR
16		KA	KA	BENAR
17			HA	SALAH

18			KA	BENAR
19		LA	LA	BENAR
20			NYA	SALAH
21		MA	MA	BENAR
22			MA	BENAR
23		MBA	MBA	BENAR
24		MPA	MPA	BENAR
25			MPA	BENAR
26		NA	NA	BENAR
27			NA	BENAR
28		NCA	WA	SALAH
29		NDA	NDA	BENAR

30			NGGA	SALAH
31		NGA	NGA	BENAR
32		NGGA	NGGA	BENAR
33		NGKA	NGA	SALAH
34			NGKA	BENAR
35		NGSA	NGSA	BENAR
36		NJA	NJA	BENAR
37		NTA	NTA	BENAR
38			NTA	BENAR
39		NYA	NYA	BENAR
40			NGA	SALAH
41		PA	PA	BENAR
42			PA	BENAR
43		RA	RA	BENAR
44			WA	SALAH
45		SA	SA	BENAR
46			SA	BENAR



47		TA	TA	BENAR
48			TA	BENAR
49		WA	WA	BENAR
50			BA	SALAH
51		YA	YA	BENAR
52			YA	BENAR
53			MA	SALAH

Setelah melakukan pengujian data tunggal mendapatkan hasil dikenali dengan benar sebanyak 41 huruf aksara Incung. Maka akurasi yang diperoleh dari pengujian data tunggal adalah sebesar:

$$\text{Akurasi} = 41/53 * 100 = 77,35\%$$

Perbedaan hasil akurasi dari pengujian data jamak dengan akurasi pengujian data tunggal karena pengaruh pada jumlah pengujian. Proses pada pengujian data jamak menggunakan jumlah data lebih banyak dibandingkan dengan pengujian data tunggal. Disamping itu, setiap orang memiliki penulisan karakter yang berbeda sehingga mempengaruhi nilai akurasi.

Tabel 4.26 Hasil Persentase Rata-rata Akurasi Keseluruhan

No	Jumlah <i>State</i>	Data Ciri 1		Data Ciri 2		Data Ciri 3		Data Ciri 4		Data Ciri 5		Data Ciri 6	
		K-Fold 3	K-Fold 5	K-Fold 3	K-Fold 5	K-Fold 3	K-Fold 5	K-Fold 3	K-Fold 5	K-Fold 3	K-Fold 5	K-Fold 3	K-Fold 5
1	2	58.49	59.77	55.81	56	52.58	51.93	23.86	23.58	34.84	34.7	60.96	59.62
2	4	72.16	72.81	73.77	74.23	71.43	72.64	29.12	29.75	41.15	41.26	75.3	74.61
3	6	76	77.19	78.45	79.12	75.41	76.37	31.15	31.7	42.41	42.52	77.53	76.9
4	8	77.25	77.82	80.67	80.57	78.6	78.99	31.87	32.33	42.89	43.4	78.3	77.51
5	10	77.76	78.13	82.05	82.33	80.27	81.99	31.38	31.87	42.39	43.33	78.66	78.26
6	12	77.02	78.13	81.32	82.43	81.64	82.03	31.87	32.85	42.14	42.35	79.54	78.89
7	14	76.52	77.9	81.45	82.33	81.07	82.6	32.14	32.49	41.26	42.2	77.86	79.37
8	16	76.02	77.06	81.55	82.12	82.14	82.22	32.26	32.43	40.61	40.4	78.34	78.72
9	18	75.35	77.09	81.93	83.1	82.24	82.83	32.56	32.35	39.94	41.03	78.74	78.78
10	20	74.91	76.58	81.45	82.31	81.66	82.94	31.66	32.1	38.78	40.82	78.47	79.16
11	22	74.76	76.12	80.44	82.49	82.16	82.45	31.84	31.4	37.82	39.81	78.68	79.18
12	24	74.07	74.82	79.9	82.2	80.25	82.2	31.95	32.62	36.71	39.73	78.05	80.19
13	26	74.13	74.47	80.1	80.78	80.29	82.18	31.53	31.53	37.25	37.17	78.32	79.6
14	28	71.99	74.78	80.08	80.88	81.57	82.14	31.34	31.28	35.81	37.95	78.45	79.27
15	30	72.12	74.44	79.16	81.49	79.64	81.95	31.09	31.7	35.16	36.94	77.53	78.97

Dari Tabel 4.26 dapat dilihat hasil rata-rata akurasi menggunakan *k-fold* 3 dan *k-fold* 5. Penggunaan *k-fold* 5 menghasilkan akurasi lebih baik daripada menggunakan *k-fold* 3. Jumlah *state* tidak memiliki pengaruh pada hasil akurasi pengujian. Setiap kenaikan *state* tidak menampilkan kenaikan akurasi yang signifikan. Nilai rata-rata akurasi ini memperlihatkan data ciri yang menghasilkan akurasi yang lebih baik pada data ciri 2 dan data ciri 3. Sedangkan hasil akurasi yang tidak baik pada data ciri 4 dan data ciri 5.

Ini menunjukkan dalam pengujian semakin banyak jumlah ciri akan menghasilkan akurasi yang semakin baik. Itu dapat dilihat dari pengujian data ciri 1 dengan ukuran ciri 3x3 dibandingkan data ciri 3 dengan ukuran ciri 5x5 akan mengalami perbedaan nilai akurasi.

Pengujian data ciri 3 dengan nilai *k-fold* 5 menghasilkan akurasi yang lebih baik dari pengujian data ciri 1, data ciri 2, data ciri 4, data ciri 5, dan data ciri 6. Hal ini menunjukkan penggunaan ekstraksi ciri memiliki pengaruh dalam proses pengenalan aksara Incung menggunakan metode *Hidden Markov Model*.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan penelitian pada pengenalan aksara Incung menggunakan metode *Hidden Markov Model*, dapat disimpulkan sebagai berikut:

1. Pengujian menggunakan ekstraksi ciri *Intensity of Character* dengan ukuran 5x5 menghasilkan 25 atribut ciri. Hasil akurasi terbaik pada pengujian 5-*fold cross validation* dengan jumlah *state* 20 dengan data yang digunakan sebanyak 4770 aksara. Akurasi yang diperoleh dari pengujian tersebut mendapatkan sebesar 89.3%.
2. Berdasarkan akurasi rata-rata menggunakan ekstraksi ciri *Intensity of Character* dan *Mark Direction* yang telah didapatkan, dapat disimpulkan sistem berhasil dibuat untuk pengenalan aksara Incung menggunakan metode *Hidden Markov Model*.
3. Pemodelan menggunakan metode *Hidden Markov Model* mampu mengenali aksara Incung dengan baik karena telah dilakukan uji data tunggal diluar data yang digunakan dalam proses pelatihan dengan nilai akurasi 77.35%.

#### **5.2 Saran**

Dari hasil penelitian pengenalan aksara Incung ini, ada beberapa saran yang dapat diberikan penulis untuk membuat sistem akan semakin baik, yaitu:

1. Pengujian menggunakan jumlah *cross validation* yang berbeda.
2. Menggunakan metode ekstraksi ciri yang berbeda, seperti ekstraksi ciri ICZ-ZCZ, atau PCA.
3. Mengembangkan sistem ini pada aplikasi android untuk pengenalan aksara Incung menggunakan metode HMM.

## DAFTAR PUSTAKA

- Alimin. (2003). *Sastra Incung Kerinci*. Sungaipenuh: Dinas Kebudayaan dan Pariwisata Kabupaten Kerinci.
- Coelho, P. (2019). *Theory and Implementation using Matlab ®*. Science Publishers Book.
- Dymarski, P. (2011). *HIDDEN MARKOV MODEL: Theory and Applications*. BoD–Books on Demand.
- Ganeshamoorthy, K., & Ranasinghe, D. N. (2008). On the performance of parallel neural network implementations on distributed memory architectures. *2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*, 90–97.
- Indyaputra, O. G. (2019). Metode Backpropagation untuk alih aksara Jawa cetak menggunakan ciri ICZ-ZCZ. Skripsi. Program Studi Informatika. Univesitas Sanata Dharma.
- Kadir, A., & Susanto, A. (2013). Teori dan aplikasi pengolahan citra. *Yogyakarta: Andi*.
- Kozok, U. (2006). *Kitab undang-undang Tanjung Tanah: naskah Melayu yang tertua*. Yayasan Obor Indonesia.
- Lokhande, M. S. D. (2017). Online Handwriting Recognition using HMM. *International Journal on Recent and Innovation Trends in Computing and Communication*, 5(8), 206–210.
- Metwally, A. H., Khalil, M. I., & Abbas, H. M. (2017). Offline Arabic handwriting recognition using hidden Markov models and post-recognition lexicon matching. In *2017 12th International Conference on Computer Engineering and Systems (ICCES)* (pp. 238-243). IEEE.
- Murphy, K. (2005), Hidden Markov Model (HMM) *Toolbox* for Matlab. <https://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html> diakses pada 26 Mei 2021
- Prarian, C., Tritasmoro, I. I., Magdalena, R., Elektro, F. T., Telkom, U., & A, P. N. (2013). Analisis Pengenalan Aksara Lampung Menggunakan Modified

- Direction Feature ( Mdf ) Dan *Hidden Markov Model*. Skripsi. Program Studi Teknik Elektro. Universitas Telkom.
- Powa, J. E. (2020). Implementasi Aksara Incung di Sungai Penuh. *Jurnal Ilmiah Dikdaya*, 10(1), 111–118.
- Prasetyo, M. E. B. (2010). Teori Dasar *Hidden Markov Model*. *Makalah II 2092 Probabilitas Dan Statistik*.
- Pratiwi, N. W., Fauziah, F., Andryana, S., & Gunaryati, A. (2018). Deteksi Wajah Menggunakan Hidden Markov Model (HMM) Berbasis Matlab. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 3(1), 44.
- Putra, D. (2010). Pengolahan citra digital. Penerbit Andi.
- Putri, A. R. (2017). Pengenalan Pola Aksara Batak Karo Menggunakan Metode Back Propagation. Skripsi. Program Studi Informatika. Universitas Sanata Dharma.
- Przytycka. (2007) "Encyclopedia Of The Human Genome: Hidden Markov Models, USA: Nature Publishing Group.
- Rabiner, L. R. (1989). Tutorial on HMM and Applications. In *Proceedings of the IEEE* (Vol. 77, pp. 257–286).
- Surinta, O., & Schomaker, L. (2010). Overview of handwritten Thai character recognition. *Lecture Notes Online*.
- Theodoridis, S., & Koutroumbas, K. (2008). Pattern recognition. *IEEE Transactions on Neural Networks*, 19(2), 376.
- Tumilaar, K., Langi, Y., & Rindengan, A. (2015). *Hidden Markov Model*. d'CARTESIAN: Jurnal Matematika dan Aplikasi, 4(1), 86-94.
- Utnasari, I. (2018). Penerapan Jaringan Syaraf Tiruan pada pengenalan karakter pola aksara Incung dengan metode Backpropagation. *Computer Based Information System Journal*, 7–11.
- Voorhoeve, P. (1970). Kerintji documents. *Bijdragen Tot de Taal-, Land-En Volkenkunde/Journal of the Humanities and Social Sciences of Southeast Asia*, 126(4), 369–399.
- Widiarti, Anastasia Rita & Himamunanto, A. R. (2013). *Teori dan Aplikasi Pengolahan Citra Digital: Transliterasi Otomatis Citra Dokumen Teks Aksara*

- Jawa*. Lintang Pustaka Utama Yogyakarta.
- Westenenk, L. C. (1922). *Rèntjong-schrift: II: beschreven hoorns in het landschap Krintji*. Lange.
- Wibowo, T. A. (n.d.). Pengenalan Pola Tulisan Tangan Aksara Jawa Dengan Algoritma Backpropagation. Skripsi. Program Studi Informatika. Universitas Santa Dharma.
- Widiarti, A. R., & Wastu, P. N. (2009). Javanese character recognition using *Hidden Markov Model*. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 3(9), 2201–2204.
- Widiarti, A. R. (2011). Comparing Hilditch, Rosenfeld, Zhang-Suen, and Nagendraprasad-Wang-Gupta Thinning. *International Journal of Computer and Information Engineering*, 5(6), 563-567.
- Widiarti, Anastasia Rita & Himamunanto, A. R. (2013). *Teori dan Aplikasi Pengolahan Citra Digital: Transliterasi Otomatis Citra Dokumen Teks Aksara Jawa*. Lintang Pustaka Utama Yogyakarta.
- Widyastuti, W. (2016). Pengenalan Aksara Pallawa dengan Model Hidden Markov. *ReTII*, 126–131.
- Yuwitaning, E. F., Hidayat, B., & Andini, N. (2014). Implementasi Metode *Hidden Markov Model* untuk Deteksi Tulisan Tangan. *E-Proceeding of Engineering*, 1(1), 396–402.

# LAMPIRAN



## Lampiran 1 Angket Aksara Incung

### ANGKET PENGUMPULAN DATA

NAMA : Andreas Agung Nugroho

PRODI : Teknik Mesin

ka	

ga	

nga	

ta	

da	

na	

pa	

ba	

ma	

ca	

ja	

nya	

sa	

ra	

la	

wa	

ya	

ha	

a(ha)	

'mba	

'ngga	

'nda	

'nja	

'mpa	

'ngka	

'nca	

'nta	

'ngsa	

## Lampiran 2 Syntax GUI

```

function varargout = hmm_gui(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @hmm_gui_OpeningFcn, ...
    'gui_OutputFcn',  @hmm_gui_OutputFcn, ...
    'gui_LayoutFcn',  [] , ...
    'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function hmm_gui_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = hmm_gui_OutputFcn(hObject, eventdata,
handles)

varargout{1} = handles.output;

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all
properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)

```

```

[file_data,nama_path] = uigetfile({'*.jpg'}, 'membuka gambar');
%memilih gambar

if ~isequal(file_data,0)
    I = imread(fullfile(nama_path,file_data));
    axes(handles.axes1);
    imshow(I);
    handles.I = I;
    guidata(hObject,handles)
else
    return
end
assignin('base','namaFile',file_data);
set(handles.edit1,'String',file_data);

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)

dataName = evalin('base','namaFile');
namaData = 'ta1.jpg';
if strcmp(dataName,namaData)
    I = handles.I;
    assignin('base','hasil',I);
    %Mengubah citra keabuan menjadi citra hitam putih
    hasilakhir = preprocessing2 (I);
    fitur = ekstrakciri_6unit(hasilakhir);
else
    I = handles.I;
    assignin('base','hasil',I);
    %% Proses Preprocessing
    hasilakhir = preprocessing (I);
    %% Proses Ekstraksi Ciri dengan IoC 5x5
    fitur = ekstrakciri_5unit(hasilakhir);
end
HMM_Models = evalin('base','HMM_Models');
% HMM_Models = evalin('base','modelF5');
hasil = hmmPengujian(fitur, HMM_Models);

assignin('base','out',hasil);

```

```

if hasil == 1
    hasil_huruf = 'BA';
elseif hasil == 2
    hasil_huruf = 'BA';
elseif hasil == 3
    hasil_huruf = 'CA';
elseif hasil == 4
    hasil_huruf = 'CA';
elseif hasil == 5
    hasil_huruf = 'CA';
elseif hasil == 6
    hasil_huruf = 'DA';
elseif hasil == 7
    hasil_huruf = 'DA';
elseif hasil == 8
    hasil_huruf = 'GA';
elseif hasil == 9
    hasil_huruf = 'GA';
elseif hasil == 10
    hasil_huruf = 'A (HA) ';
elseif hasil == 11
    hasil_huruf = 'A (HA) ';
elseif hasil == 12
    hasil_huruf = 'HA';
elseif hasil == 13
    hasil_huruf = 'HA';
elseif hasil == 14
    hasil_huruf = 'JA';
elseif hasil == 15
    hasil_huruf = 'JA';
elseif hasil == 16
    hasil_huruf = 'KA';
elseif hasil == 17
    hasil_huruf = 'KA';
elseif hasil == 18
    hasil_huruf = 'KA';
elseif hasil == 19
    hasil_huruf = 'LA';
elseif hasil == 20
    hasil_huruf = 'LA';
elseif hasil == 21
    hasil_huruf = 'MA';
elseif hasil == 22
    hasil_huruf = 'MA';
elseif hasil == 23
    hasil_huruf = 'MBA';
elseif hasil == 24
    hasil_huruf = 'MPA';
elseif hasil == 25
    hasil_huruf = 'MPA';
elseif hasil == 26
    hasil_huruf = 'NA';
elseif hasil == 27
    hasil_huruf = 'NA';
elseif hasil == 28

```

```

        hasil_huruf = 'NCA';
elseif hasil == 29
    hasil_huruf = 'NDA';
elseif hasil == 30
    hasil_huruf = 'NDA';
elseif hasil == 31
    hasil_huruf = 'NGA';
elseif hasil == 32
    hasil_huruf = 'NGGA';
elseif hasil == 33
    hasil_huruf = 'NGKA';
elseif hasil == 34
    hasil_huruf = 'NGKA';
elseif hasil == 35
    hasil_huruf = 'NGSA';
elseif hasil == 36
    hasil_huruf = 'NJA';
elseif hasil == 37
    hasil_huruf = 'NTA';
elseif hasil == 38
    hasil_huruf = 'NTA';
elseif hasil == 39
    hasil_huruf = 'NYA';
elseif hasil == 40
    hasil_huruf = 'NYA';
elseif hasil == 41
    hasil_huruf = 'PA';
elseif hasil == 42
    hasil_huruf = 'PA';
elseif hasil == 43
    hasil_huruf = 'RA';
elseif hasil == 44
    hasil_huruf = 'RA';
elseif hasil == 45
    hasil_huruf = 'SA';
elseif hasil == 46
    hasil_huruf = 'SA';
elseif hasil == 47
    hasil_huruf = 'TA';
elseif hasil == 48
    hasil_huruf = 'TA';
elseif hasil == 49
    hasil_huruf = 'WA';
elseif hasil == 50
    hasil_huruf = 'WA';
elseif hasil == 51
    hasil_huruf = 'YA';
elseif hasil == 52
    hasil_huruf = 'YA';
elseif hasil == 53
    hasil_huruf = 'YA';
else
    return
end

assignin('base','hasil_huruf',hasil_huruf);

```

```

set(handles.edit2,'String',hasil_huruf);

function edit2_Callback(hObject, eventdata, handles)

function edit2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

### Lampiran 3 Syntax Inisialisasi Parameter HMM

```

function [prior0, transmat0, mu0, sigma0, mixmat0] =
initializeParam( ...
    data, Q, O, M, cov_type, model_type)

% model type: 1 - 'ergodic', 2 - 'bakis'
switch model_type
    case 1
        % ergodic model
        prior0 = normalise(rand(Q,1));
        transmat0 = mk_stochastic(rand(Q,Q));
    case 2
        % bakis model
        level = 2; % levels of bakis model
        [prior0, transmat0] = initByBakis(Q, level);
end

%% menentukan nilai mu dan sigma
[mu0, Sigma0] = mixgauss_init(Q*M, cell2mat(data), cov_type);
mu0 = reshape(mu0, [O Q M]);
sigma0 = reshape(Sigma0, [O O Q M]);
mixmat0 = mk_stochastic(rand(Q,M));
end

```

### Lampiran 4 Syntax Pelatihan HMM

```

function HMM_Models = hmmTrain(TR_Actions, param, Q, O)
%% training using hmm
%% HMM parameters
M = param.M;
cov_type = param.cov_type;
max_iter = param.max_iter;
verbose = param.verbose;

%% Preparation for HMM
training_number = length(TR_Actions);
all_labels = zeros(training_number, 1);

% get labels
for i = 1:training_number
    all_labels(i) = TR_Actions(i).label;
end

```



```

end

labels = unique(all_labels);
label_number = size(labels, 1);

HMM_Models = struct;

%% Training models
for i = 1:label_number
    Train_Data = cell(1, 1);
    label = labels(i);
    HMM_Models(i).label = label;

    %% Mendapatkan pelatihan data yang memiliki label yang sama
    for j = 1 : training_number
        if TR_Actions(j).label == label
            len = size(Train_Data, 2);
            if isempty(Train_Data{len})
                Train_Data{len} = TR_Actions(j).Observations;
            else
                Train_Data{len + 1} = TR_Actions(j).Observations;
            end
        end
    end
end

%% HMM training
% initial guess (Ergodic model)
model_type = 2;
[prior0, transmat0, mu0, sigma0, mixmat0] =
initializeParam(...
    Train_Data, Q, O, M, cov_type, model_type);

% improve guess by using iterations of EM
[LL, prior1, transmat1, mu1, sigma1, mixmat1] = mhmm_em(...
    Train_Data, prior0, transmat0, mu0, sigma0, mixmat0, ...
    'max_iter', max_iter, 'cov_type', cov_type, 'verbose',
verbose);

HMM_Models(i).LL = LL;
HMM_Models(i).prior = prior1;
HMM_Models(i).transmat = transmat1;
HMM_Models(i).mu = mu1;
HMM_Models(i).sigma = sigma1;
HMM_Models(i).mixmat = mixmat1;
end

```

## Lampiran 5 Syntax Pengujian HMM

```
function [accuracy, predict_label, true_label, count] =
hmmTest(TE_Actions, HMM_Models)
model_number = length(HMM_Models); % model number
test_number = length(TE_Actions);
true_label = zeros(test_number, 1);
predict_label = zeros(test_number, 1);

prob_scores = zeros(test_number, model_number);
count = 0;
for i = 1:test_number
    % get true label
    true_label(i) = TE_Actions(i).label;
    for j = 1:model_number
        data = TE_Actions(i).Observations;
        prob_scores(i, j) = mhmm_logprob(data,
HMM_Models(j).prior, ...
        HMM_Models(j).transmat, HMM_Models(j).mu, ...
        HMM_Models(j).sigma, HMM_Models(j).mixmat);
    end

    [max_prob, idx] = max(prob_scores(i,:));
    predict_label(i) = HMM_Models(idx).label;

    if predict_label(i) == true_label(i)
        count = count + 1;
    end
end
% Hitung nilai akurasi
accuracy = (count / test_number)*100;
end
```