**PEMBELAJARAN HURUF DAN KALIMAT ARAB BERBASIS PENGENALAN CITRA**

**TESIS**

**Karya tulis sebagai salah satu syarat**

**untuk memperoleh gelar Magister dari**

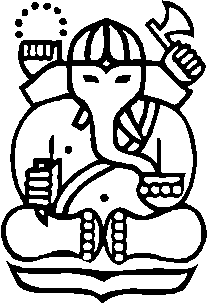
**Institut Teknologi Bandung**

**Oleh**

**AINATUL RADHIAH**

**NIM: 23215145**

**(Program Studi Magister Teknik Elektro)**

****

**INSTITUT TEKNOLOGI BANDUNG**

**Mei 2017**

**ABSTRAK**

**PEMBELAJARAN HURUF DAN KALIMAT ARAB BERBASIS PENGENALAN CITRA**

**Oleh**

**Ainatul Radhiah**

**NIM: 23215145**

**PROGRAM STUDI MAGISTER TEKNIK ELEKTRO**

Optical Character Recognition (OCR) berfungsi untuk memindai citra digital menjadi teks yang ada didalam citra tersebut. Dengan adanya OCR, citra yang berisi tulisan tangan, tulisan mesin ketik atau computer text, dapat dimanipulasi. Bahasa arab digunakan oleh lebih dari 1 miliyar pengguna di dunia. Jika sistem OCR tersedia untuk karakter arab, akan memiliki manfaat yang banyak. Telah banyak penelitian tentang pengenalan karakter latin, Jepang dan Cina, tetapi sedikit penelitian tentang huruf Arab, ini dikarenakan terdapat kesulitan dalam huruf Arab, yakni pada bagian segmentasi, karakteristik tulisan Arab yang bersambung membuat tahap segmentasi huruf menjadi sulit. Pada penelitian ini akan dirancang sistem yang dapat mengenali huruf arab dalam rangkaian kata. Adapun tahap yang dilakukan ialah *pre-processing*, segmentasi, *thinning*, ekstraksi fitur, dan klasifikasi. Pada tahap *pre-processing* dimulai dengan *binarization, skew detection, baseline estimation, median filter* dan *resizing*. Pada tahap segmentasi, segmentasi baris dilakukan dengan verifikasi lintas *horizontal projection* dengan standar deviasi, segmentasi kata dilakukan dengan komponen pixel yang terhubung, dan segmentasi huruf dilakukan dengan algoritma Zidouri. Pada tahap *Thinning* dilakukan dengan algoritma Zhang Suen. Pada tahap ekstraksi fitur dilakukan dengan metode *Freman Chain Code,* jumlah komponen yang terhubung, jumlah lubang, jumlah titik dan posisi titik. Pada tahap klasifikasi menggunakan metode *Hidden Markov Model* (HMM), dan *Support Vector Machine* (SVM) untuk dibandingkan hasil dari kedua metode tersebut.

Kata kunci: OCR, *Freeman Chain Code*, HMM, SVM

**ABSTRACT**

**LEARNING ARABIC LETTERS AND SENTENCE BASED ON IMAGE RECOGNITION**

**By**

**Ainatul Radhiah**

**NIM: 23215145**

**ELECTRICAL ENGINEERING MASTER PROGRAM**

Optical Character Recognition (OCR) allows you to scan a digital image into text. With OCR, the image of handwritten text, writing text typewriter or computer, can be manipulated. Arabic is used by more than 1 billion users worldwide. If the OCR system is available for Arabic characters, will have many benefits. It has been a lot of research on Latin character recognition, Japan and China, but a little research on the Arabic letter, is because there are difficulties in the Arabic letter, which is on the segmentation, the characteristics of the Arabic letters is connected making stage segmentation becomes difficult. This research will be designed a system that can recognize Arabic letters in a series of words. The stage is carried out pre-processing, segmentation, thinning, feature extraction and classification. In the pre-processing stage began with binarization, skew detection, estimation baseline, median filter and resizing. In segmentation stage, segmentation line is done with cross verification with a standard deviation of horizontal projection, segmentation of words is done by pixel components are connected, and segmentation letter is done by Zidouri algorithms. In Thinning stage is done by Zhang Suen algorithms. In the feature extraction stage was conducted by Freman Chain Code, the number of connected components, number of holes, the number of points and the position of dots. In the classification stage using Hidden Markov Model (HMM), and Support Vector Machine (SVM) to compare the results of the two methods.

Keywords: OCR, *Freeman Chain Code*, HMM, SVM.

**KATA PENGANTAR**

Puji syukur penulis panjatkan ke hadirat Allah SWT, yang atas rahmat dan karunia-Nya penulis dapat menyelesaikan tesis ini. Shalawat dan salam penulis ucapkan kepada Rasulullah Muhammad SAW beserta keluarganya.

Tesis yang berjudul “Pembelajaran Huruf dan Kalimat Bahasa Arab Berbasis Pengolahan Citra” ini dibuat untuk memenuhi salah satu syarat dalam mendapatkan gelar Magister Teknik dari Program Studi Magister Teknik Elektro, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

Tesis ini tidak mungkin selesai tanpa bantuan dari banyak pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada pihak-pihak yang turut terlibat dalam proses pengerjaan tesis ini, yaitu:

1. Bapak Prof. Dr. Carmadi Machbub, selaku pembimbing pertama yang telah  
   memberikan bimbingan, arahan dan motivasi dalam menyelesaikan tesis ini.
2. Bapak Dr. Dr.techn. Ary Setijadi P, ST.,MT., selaku dosen wali dan pembimbing kedua yang telah memberikan bimbingan, arahan dan motivasi dalam menyelesaikan perkuliahan.
3. Bapak Dr. Egi Muhammad Idris Hidayat, selaku pembimbing ketiga yang telah memberikan bimbingan, arahan dan motivasi dalam menyelesaikan tesis ini.
4. Ayah dan Ibu penulis untuk doa, semangat, nasehat dan dukungannya selama proses penyelesaian skripsi ini.
5. Adikku Husnul Fitri dan Nurfitri Aulia yang telah selalu memberikan doa dan semangat dalam proses penyelesaian tugas akhir ini.
6. Sahabat – sahabat penulis Kak Hendi Irawan, Arif, Gazandi, Bening Qias Ranum, Ulfa Asmi, Agis Baswara, mas Hanif yang senantiasa memberikan dukungan, semangat dan bantuan dalam proses penyelesaian tugas akhir ini serta bantuannya selama proses pembelajaran di kampus.
7. Dan pihak – pihak yang juga telah banyak memberikan bantuan kepada penulis  
   yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa penelitian ini masih jauh dari sempurna, banyak kelemahan baik dalam penyajian maupun penulisannya. Oleh karena itu penulis mengharapkan kritik dan saran yang membangun guna penyempurnaan penelitian dan penulisan selanjutnya. Akhir kata penulis berharap semoga penulisan tesis penelitian ini dapat bermanfaat bagi kita semua.

Bandung, Mei 2017

Penulis

|  |
| --- |
|  |

**DAFTAR ISI**

Halaman

|  |  |
| --- | --- |
| ABSTRAK ....................................................................................................... | i |
| ABSTRACT....................................................................................................... | ii |
| PEDOMAN PENGGUNAAN ......................................................................... | iii |
| HALAMAN PERSEMBAHAN........................................................................ | iv |
| KATA PENGANTAR ..................................................................................... | v |
| DAFTAR ISI .................................................................................................... | vii |
| DAFTAR GAMBAR ....................................................................................... | viii |
| DAFTAR TABEL ............................................................................................ | xii |
| Bab I Pendahuluan ........................................................................................ | 1 |
| I.1 Latar Belakang ................................................................................ | 1 |
| I.2 Tujuan .............................................................................................. | 2 |
| I.3 Batasan Masalah ............................................................................. | 2 |
| I.4 Metodologi Penelitian ..................................................................... | 2 |
| I.5 Sistematika Penulisan ..................................................................... | 3 |
| Bab II Tinjauan Pustaka ................................................................................. | 5 |
| II.1 Catur ................................................................................................ | 5 |
| II.2 *Engine* Catur ................................................................................... | 6 |
| II.2.1 Macam-Macam *Engine* Catur ...................................................... | 6 |
| Bab III Perancangan Pengujian ....................................................................... | 13 |
| III.1 Analisis Kebutuhan......................................................................... | 13 |
|  |  |
| Bab IV Hasil Pengujian dan Analisis ............................................................. | 21 |
| 4.1 Hasil Perbandinga............................................................................ | 21 |
| Bab V Kesimpulan dan Saran ......................................................................... | 32 |
| 5.1 Kesimpulan ..................................................................................... | 32 |
| 5.2 Saran ............................................................................................... | 32 |
| DAFTAR PUSTAKA ...................................................................................... | 34 |

**Bab I**

**Pendahuluan**

1. **Latar Belakang**

*Optical Character Recognition* (OCR) berfungsi untuk memindai citra digital menjadi teks yang ada didalam citra tersebut. Dengan adanya OCR, citra yang berisi tulisan tangan, tulisan mesin ketik atau teks computer dapat dimanipulasi. Teks yang discan dengan OCR dapat dicari kata per-kata atau per-kalimat. Kinerja pengenalan huruf pada OCR tergantung pada kualitas input teks yang dimasukkan, dan teknik yang berbeda yang digunakan untuk meningkatkan pengenalan dari karakter. Bahasa arab digunakan oleh lebih dari 1 miliyar pengguna di dunia. Jika sistem OCR tersedia untuk karakter arab, akan memiliki manfaat yang banyak.

Telah banyak penelitian tentang pengenalan huruf latin, jepang dan cina, tetapi sedikit penelitian tentang huruf Arab, ini dikarenakan terdapat kesulitan dalam pengenalan huruf Arab, yakni pada bagian segmentasi, karakteristik tulisan Arab yang bersambung membuat tahap segmentasi huruf menjadi sulit. Oleh karena itu perlu dirancang sebuah teknik segmentasi yang canggih, agar bisa memisahkan tulisan Arab yang bersambung.

Penelitian sebelumnya tentang pengenalan huruf arab telah berkembang, seperti yang dilakukan Albadr (2013) dengan mengekstraksi 24 fitur, diantaranya adalah *boundary chain code,* dan klasifikasi dengan menggunakan *decision tree* yang dihasilkan oleh algoritma C4.5 menghasilkan akurasi pengenalan sebesar 82%. Kemudian beberapa penelitian yang menggunakan *Hidden Markov Model* (HMM)dan *Support Vector Machine* (SVM) sebagai pengklasifikasi. Dalam penelitiam yang dilakukan Anugrah Redja Kusuma (2006) menggunakan HMM mencapai tingkat akurasi 50% hingga 60%. Penelitian menggunakan SVM dengan akurasi 99% dilakukan oleh Mehmmood Abdulla Abd. Berdasarkan hal tersebut, maka penulis ingin membangun sistem pengenalan huruf arab dalam rangkaian kata, dengan menggunakan klasifikasi *Hidden Markov Model* (HMM)dan *Support Vector Machine* (SVM) untuk dibandingkan hasil pengenalan dari keduanya.

1. **Tujuan**

Berdasarkan latar belakang dan rumusan masalah, berikut tujuan yang ingin dicapai dalam penelitian ini.

1. Merancang sistem yang dapat mengenali huruf arab dalam rangkaian kata. Jika diberikan masukan berupa kata atau kalimat dalam bahasa Arab, maka sistem akan mampu mengenali huruf tersebut.
2. Melakukan sistem pengenalan huruf Arab melalui *Hidden Markov Model* (HMM) dan *Support Vector Machine* (SVM)
3. **Batasan Masalah**

Hal-hal yang menjadi batasan pada penelitian ini adalah sebagai berikut.

1. Bahasa pemograman yang digunakan adalah Bahasa Java ™ SE Runtime Environtment (build 1.8.0.72-b15).
2. Tulisan huruf arab yang digunakan adalah dari tulisan cetak dengan font Times New Roman dengan besar font 72.
3. Laptop dengan processor core i3 dengan RAM 8GB.
4. **Metodologi Penelitian**

Tahapan-tahapan kegiatan yang dilakukan dalam penelitian ini dijelaskan sebagai berikut.

1. Studi literatur

Mempelajari dan menganalisis berbagai sumber informasi seperti buku – buku dan literatur referensi yang serupa dengan penelitian yang akan dilakukan

1. Analisis desain

Berdasarkan hasil studi literatur akan dibuat analisis desain dalam menyelesaikan proses perbandingan algoritma untuk setiap metode.

1. Pembuatan sistem

Hasil perancangan diimplementasikan dengan menggunakan Java sebagai pemrosesan citra.

1. Pengujian dan evaluasi

Berdasarkan aplikasi yang telah dibuat kemudian dilakukan uji coba sistem dan  
mengevaluasi sistem sesuai dengan tujuan penelitian

1. Pengambilan data

Setelah pengujian dan didapatkan hasil yang sesuai dengan tujuan penelitian  
kemudian dilakukan pengambilan data

1. Penulisan laporan

Penulisan laporan hasil pengujian dan pengambilan data dibukukan dalam  
laporan penelitian

1. **Sistematika Penelitian**

Dalam penulisan laporan tesis ini terdiri dari lima bab dengan masing–masing bab menguraikan beberapa hal yang terkait dengan perancangan yang dilakukan.

Bab I Pendahuluan. Bab ini mengemukakan latar belakang, tujuan, batasan masalah, metode penelitian, dan sistematika penulisan.

Bab II Tinjauan Pustaka. Bab ini menjelaskan teori dasar mengenai *Optical Character Recognition* (OCR), *Freeman Chain Code, Thinning dan Hidden Markov Model* (HMM) dan *Support Vector Machine* (SVM).

Bab III Perancangan Sistem dan Eksperimen. Bab ini membahas perancangan awal sistem dan eksperimen pengenalan huruf Arab, perancangan sistem dan eksperimen yakni ekstraksi fitur, dan perancangan sistem dan eksperimen untuk pengklasifikasi, serta perangkat lunak untuk pengenalan huruf Arab.

Bab IV Pengujian dan Pembahasan. Bab ini membahas pengujian sistem pengenalan huruf arab yang menghasilkan data yang dibutuhkan sebagai analisis pembahasan perbandingan metode.

Bab V Penutup. Bab ini mengemukakan bagian penutup dari pelaporan penelitian yang meliputi kesimpulan dan saran agar dapat dikembangkan dengan metode lainnya untuk sistem kerja yang sama.

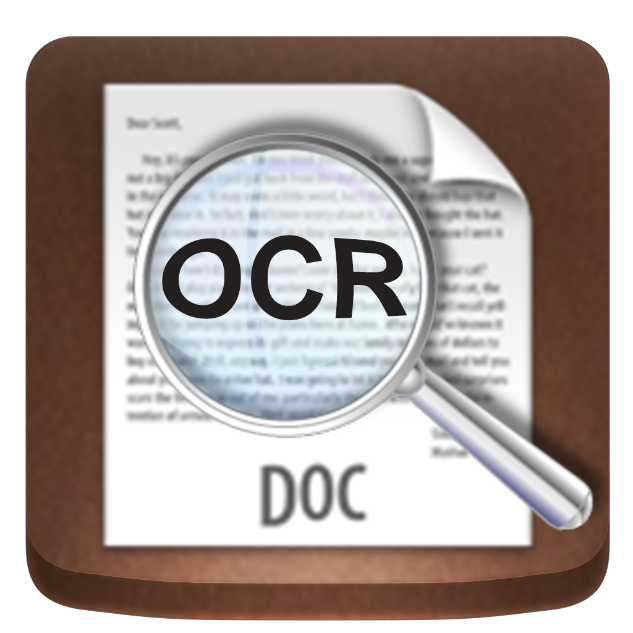
Dalam bab selanjutnya akan disampaikan teori-teori dasar hasil dari peninjauan beberapa literatur yang digunakan. Literatur tersebut dapat berupa buku-buku. makalah penelitian yang telah dipublikasikan, dan sebagainya yang dijadikan sebagai referensi dalam melakukan penelitian ini.

**Bab II**

**Tinjauan Pustaka**

## II.1 Optical Character Recognition (OCR)

OCR (Optical Character Recognition) berfungsi untuk memindai citra digital menjadi teks yang ada didalam citra tersebut. OCR juga bisa menjadi pendukung atau aplikasi tambahan untuk scanner. Dengan adanya OCR, citra yang bertulisan tangan, tulisan mesin ketik atau teks komputer, dapat dimanipulasi. Teks yang discan dengan OCR dapat dicari kata per kata atau per kalimat, dan setiap teks dapat dimanipulasi, diganti, atau diberikan barcode.



Gambar II.1 *Optical Character Recognition* (OCR)

(<https://learn.eprolabs.com/wp-content/uploads/2017/01/OCR-Feature.png>)

Pada Arabic Optical Character Recognition dilakukan dengan 5 tahap yakni sebagai berikut.

1. Tahap *pre-processing* untuk meningkatkan kualitas citra masukan dan karakter yang akan dikenali.
2. Tahap Segmentasi untuk memotong huruf yang terdapat pada rangkaian kata atau kalimat menjadi huruf – huruf yang berdiri sendiri.
3. Tahap *Thinning* berguna untuk mengambil tulang dari huruf yang telah disegmentasi, penulangan beguna untuk mengambil jalur dari gambar.

4. Tahap Ekstraksi Ciri untuk mengambil ciri pembeda dari citra untuk dimasukkan ke proses klafisikasi.

5. Tahap Klasifikasi untuk memproses ciri yang mengidentifikasikan citra karakter dan mengklasifikasikannya kedalam masing – masing kelas.

**II.2 Machine Learning**

Menurut Tom M. Mitchel (1997), defenisi fomal tentang machine learning sebagai berikut :

“Sebuah program komputer dikatakan belajar dari pengalaman E yang bergabung pada target T dan ukuran kinerja program P jika kinerja pada target T, menggunakan ukuran P, ditingkatkan oleh pengalaman E.”

Dengan kata lain, *machine learning* meningkatkan mesin untuk memberikan hasil prediksi pada data baru setelah mendapat pengalaman dari sekumpulan data (data training). Berdasarkan input yang diberikan pada data training, *machine learning* dikelompokkan menjadi dua jenis, yaitu :

**II.2.1 *Supervised Learning***

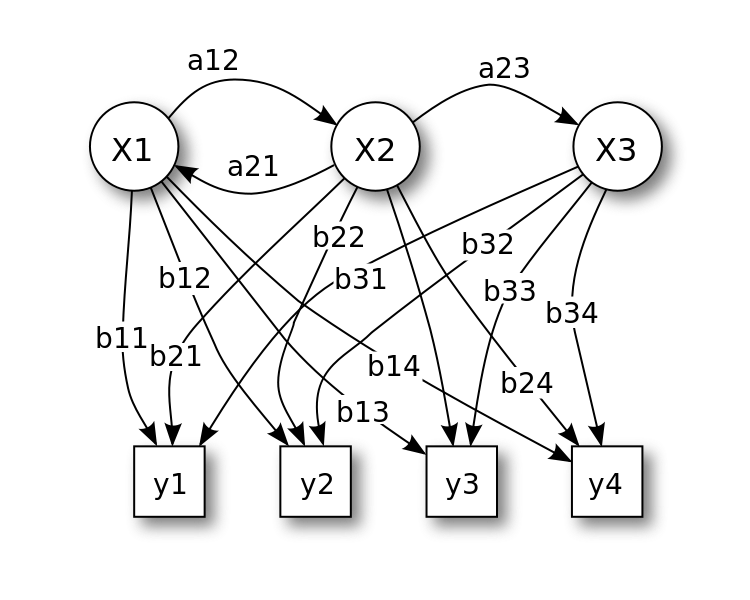
Pada *supervised learning*, *data training* disertai target pada setiap datanya, {xi,ti}, i = 1…N, dengan **x** adalah vector input dan t adalah target. Tujuan dari *supervised learning* adalah membangun model yang dapat memberikan hasil/output secara benar untuk suatu data input. *Supervised learning* digunakan untuk *classification, regression, ordinal regression, ranking,* dll.

**II.2.2 *Unsupervised Learning***

Pada *unsupervised Learning,* data training tidak disertai target pada setiap datanya, {xi}, i = 1…N, dengan **x** adalah vector input. Tujuan dari *unsupervised learning* adalah membangun model yang dapat menemukan variable tersembunyi dari *data training*. *Unsupervised learning* digunakan untuk *clustering, concept extraction, 0-97recommendation, density estimation, dimensionality reduction,* dll.

## II.3 Hidden Markov Model (HMM)

*Hidden Markov Model* (HMM) adalah perluasan dari rantai Markov dimana *state*nya tidak dapat diamati secara langsung (tersembunyi). Perubahan *hidden state* hanya dapat diobservasi melalui pengamatan variable lain.



Gambar II.2 *Hidden Markov Model* (HMM)

(<https://en.wikipedia.org/wiki/Hidden_Markov_model#/media/File:HiddenMarkovModel.svg>)

Terdapat tiga masalah utaa yang dapat diselesaikan oleh Hidden Markov Model.

**II.3.1 Evaluasi**

Operasi evaluasi dalam HMM adalah perhitungan probabilitas dari urutan nilai observasi yang diberikan oleh *Hidden Markov Models.* Algoritma yang banyak digunakan untuk penyelesaian masalah observasi adalah algoritma maju (*Forward Algorithm*) yang keadaannya mengalir kedepan. Algoritma mundur (*Backward Algorithm*) yang keadaannya mengalir kebelakang dari observasi terakhir pada saat *T,* dan algoritma maju – mundur (*forward-backward algorithm*) yang merupakan gabungan dari algoritma *forward – backward.* .[10]

1. Algoritma maju

Jika variable forward , pada saat *t* dan state *i*, maka persamannya

(II.1)

Dengan *O* = indeks matrik terobservasi

Penyelesain dengan n keadaan dan observasi sampai T secara iterasi

* Inisialisasi :

, (II.2)

dimana = matrik state awal dan = matriks pertama yang terobservasi.

* Induksi :

… (II.3)

dimana :

= jumlah state

= matrik transisi.

* Terminasi :

(II.4)

1. Algoritma mundur

Keadaan mengalir ke belakang dari observasi terakhir saat *t.* Persamaan probabilitas mundur sebagai berikut:

(II.5)

dan dianalogikan dengan prosedur *forward* dengan langkah:

* Inisialisasi

(II.6)

* Induksi

, (II.7)

, (II.8)

**II.3.2 Pengkodean (*Decoding*)**

Permasalahan *decoding* ini yaitu menemukan barisan *state* terbaik (optimal) yang berasosiasi dengan barisan observasi dari sebuah model yang juga telah diketahui. Barisan *state* yang mempunyai probabilitas tertinggi dalam menghasilkan barisan observasi yang telah diketahui sebelumnya. Untuk menentukan keadaan tersembunyi dari suatu barisan observasi perlu digunakan suatu metode yang mempertimbangkan probabilitas transisi *state* pada proses pencarian barisan *state* yang paling optimal. Metode yang digunakan untuk penyelesaian masalah ini antara lain algoritma Viterbi. Langkah – langkah pengkodean sebagai berikut.

* Inisialisasi :

, (II.9)

* Rekursi :

, (II.10)

* Terminasi :

(II.11)

Dimana :

*P* = Probabilitas

**II.3.3 Learning**

Operasi learning dalam HMM adalah melatih parameter HMM jika diberikan data set barisan – barisan tertentu agar dapat menemukan himpunan transisi state yang paling mungkin beserta probabilitas output. Untuk menyelesaikan permasalahan terakhir pada HMM ini, biasanya digunakan algoritma *Baum-Welch* yang merupakan kasus khusus dari algoritma EM (Ekspektasi Maksimum). Algoritma EM sendiri merupakan algoritma yang digunakan untuk mempelajari model – model probabilistik dalam suatu kasus yang melibatkan keadaan – keadaan tersembunyi.

1. Parameter A

Parameter A ditunjukan dengan kumpulan status transisi yang membentuk matrik tertentu yaitu

(II.12)

bentuk

(II.13)

Merupakan peluang ketika state *j* pada waktu *t+1* jika pada waktu *t*  berada di state *i*

= Probabilitas transisi dari state i ke state j

*P*  = probabilitas

*n* = banyaknya hidden state dalam model

1. Parameter B

Parameter B ditunjukkan bahwa

(II.14)

yang merupakan

(II.15)

dan (II.16)

= Probabilitas distribusi matriks observasi

*m* = banyaknya simbol observasi yang berbeda pada tiap state

Pada *continuos density* HMM sering dikarakterisasi oleh fungsi kerapatan (*density function*) atau campuran fungsi kepadatan tertentu di setiap *state* . Dengan asumsi penggunaan *Gaussian* *Mixture*, kepadatan emisi *state* 𝑗 didefinisikan sebagai:

(II.17)

Dimana :

K = *number of mixture*

= *mixing coefficient*

untuk *Gaussian* pada *state* j dengan batasan stokastik dengan persamaan berikut.

(II.18)

Dimana :

= *Gaussian density function* dengan mean dan matrik kovarian untuk *campuran*.

1. Matriks awal di state *i*

Diperlukan inisialisasi matriks awal status yang ditunjukan oleh.

dengan :

(II.19)

Sehingga HMM dapat dilambangkan dengan .

## II.4 Support Vector Machine

Support Vector Machine (SVM) merupakan salah satu metode *machine learning* untuk *pattern recognition*, yang dalam penelitian ini adalah *image recognition*. Algoritma SVM pertama kali dikembangkan oleh Vladimir Vapnik (Bishop,2006). Konsep dasar SVM ialah mengklasifikasikan data menjadi dua kelas yang berbeda dengan cara membuat membuat *hyperplane* atau suatu bidang yang merupakan fungsi klasifikator antara dua kelas dengan menggunakan konsep memaksimumkan margin.

Bentuk umum klasifikator SVM ialah sebagai berikut :

y = *sign*(wTx + b) (II.20) Keterangan :

y merupakan nilai target pada setiap vector baris

x Rn merupakan vector yang dimensinya bergantung dari *n* banyaknya vitur

w Rn vector yang menjadi parameter bobot

b bias atau eror berupa skalar

Hyperplane yang dihasilkan SVM dapat menklasifikasikan data menjadi dua kelas, yaitu kelas positf dan kelas negative yang dimodelkan sebagai berikut :

wTxi + b ≥1, untuk yi bernilai 1 (II.21)

wTxi + b ≥1, untuk yi bernilai -1 (II.22)

dapat dirumuskan sebagai berikut

yi (wTxi + b) ≥1 (II.23)

Misal dibuat dua bidang yang sejajar dengan hyperplane, untuk kelas B+ dapat dirumuskan sebagai berikut,

B+ : wTxi + b = 1 (II.24)

Untuk kelas B- dapat dirumuskan sebagai berikut,

B- : wTxi + b = -1 (II.25)

Margin adalah jarak antara bidang B+ dan B-

X1

Hyperplane

yi = 1

B+ : wTxi + b = 1

B- : wTxi + b = -1

yi = -1

||W||

X2

Gambar II.3 *Hyperplane*

**II.4.1 Fungsi Kernel**

Pada umumnya masalah dalam domain dunia nyata jarang bersifat linear separable. Kebanyakan bersifat non linear. Oleh karena itu untuk menyelesaikan masalah non linear, SVM dapat dimodifikasikan dengan memasukkan fungsi kernel. Dalam non linear SVM, data dipetakan oleh fungsi Ф () ke ruang vector yang berdimensi lebih tinggi. Pada ruang vector yang baru ini, hyperplane yang memisahkan dua kelas tersebut dapat di konstruksikan.

Hal ini sejalan dengan teori Cover yang menyatakan “Jika suatu transformasi bersifat non linear dan dimensi dari feature space cukup tinggi, maka data pada input space dapat dipetakan ke feature space yang baru, dimana pattern-pattern tersebut pada probabilitas tinggi dapat dipisahkan secara linear” (Cover, 1965)

Hyperplane

Gambar II.4 Kernel pada SVM

Pada gambar diatas dapat dilihat bahwa pada kelas kuning dan pada kelas merah yang berada pada input space berdimensi dua tidak dapat dipisahkan secara linear. Selanjutnya ditunjukkan bahwa fungsi Ф memetakan tiap data pada input space tersebut ke ruang vector baru yang berdimensi lebih tinggi (dimensi 3), dimana kedua kelas dapat dipisahkan secara linear oleh sebuah hyperplane. Pemetaan ini dilakukan dengan cara menjaga topologi data, dalam artian dua data yang berjarak dekat pada input space, sebaliknya dua data yang berjarak jauh pada feature space.

**II.4.2 Multi class SVM**

*Hyperplane* yang dihasilkan SVM hanya bisa mengklasifikasikan dua kelas. Sedangkan pada kenyataannya akan banyak ditemukan kasus yang lebih dari dua kelas. Oleh karena itu dapat digunakan *Multiclass* SVM untuk mengklasifikasikan permasalahan yang memiliki lebih dari dua kelas. Terdapat beberapa pendekatan yang dapat menyelesaikan masalah *Multiclass* dengan menggunakan SVM, diantaranya dalah *One Vs All* (OVA) dan *One Vs One* (OVO).

**II.4.2.1 *One Vs All* (OVA)**

Memisahkan permasalahan yang ditemui dari *N* kelas. Sehingga akan dibuat *N decision boundary. Decision boundary* yang dihasilkan merupakan hasil dari pencarian *hyperplane* dari kelas ke *i* dengan kelas sisa yang lainnya. Berikut dilakukan simulasi dengan menggunakan tiga kelas.

A

A

A

A

B

B

B

B

C

C

C

C

Gambar 2.4 Contoh pendekatan OVA

Pada gambar diatas terdapat tiga kelas yaitu kelas A, kelas B, dan kelas C. karena menggunakan metode OVA akan dibuat tiga *decision boundary* dari tiga kelas tersebut. Setiap kelas dicari *decision boundary* dengan kelas sisanya.

A

A

A

A

B

B

B

D1

B

C

C

C

C

Gambar 2.5 *Decision boundary D1*pada pendekatan OVA

Pada gambar 2.5 dibuat *decision boundary* dari kelas A dengan kelas sisanya (kelas B dan C). Kelas A dilabelkan sebagai positif, sedangkan kelas sisanya dilabelkan menjadi negative, sehingga D1 merupakan decision boundary yang dihasilkan dari kelas A dan kelas sisanya (kelas B dan kelas C).

A

A

A

A

B

B

B

D2

B

C

C

C

C

Gambar 2.6 *Decision boundary D2* pada pendekatan OVA

Selanjutnya, sesuai dengan Gambar 2.6, dibuat *decision boundary* dari kelas B dengan kelas sisanya (kelas A dan kelas C). Kelas B dilabelkan sebagai positif, sedangkan kelas sisanya dilabelkan menjadi negatif, sehingga D2 merupakan *decision boundary* yang dihasilkan dari kelas B dan kelas sisanya (kelas A dan kelas C).

B

B

A

A

A

A

D3

B

B

c

c

c

c

Gambar 2.7 *Decision boundary D3* pada pendekatan OVA

Lalu yang terakhir adalah mencari *decision boundary* yang terakhir. Sesuai dengan gambar 2.7, dibuat *decision boundary* dari kelas C dengan kelas sisanya (kelas A dan kelas B). kelas C dilabelkan sebagai positif, sedangkan kelas sisanya dilabelkan menjadi negative, sehingga D3 merupakan *decision boundary* yang dihasilkan dari kelas C dan kelas sisanya (kelas A dan kelas B).

A

A

A

A

B

B

B

D2

D1

B

D3

C

C

C

C

Gambar 2.7 *Decision boundary D3* pada pendekatan OVA

**II.6.2.2 *One Vs One* (OVO)**

Misalkan permasalahan yang ditemui *N* kelas. Sehingga akan dibuat *N(N-1)/2 Decision Boundary, decision boundary* yang dihasilkan merupakan hasil pencarian *hyperplane* dari setiap kelas yang lainnya. Berikut dilakukan simulasi dengan menggunakan tiga kelas.

C

C

C

B

B

B

A

A

B

A

A

C

Gambar 2.8 contoh pendekatan OVO

Pada Gambar 2.8 terdapat tiga kelas, yaitu kelas A, elas B dan kelas C. Karena menggunakan metode OVO dibuat tiga *decision boundary* dari tiga kelas tersebut. Setiap kelas dicari  *decision boundary* dengan masing-masing kelas lainnya.

D1

B

B

A

A

B

A

A

B

Gambar 2.9 *Decision Boundary D1* pada pendekatan OVO

Sesuai dengan Gambar 2.9, dibuat *decision boundary* dari kelas A dengan kelas B. kelas A dilabelkan sebagai positif, sedangkan kelas B dilabelkan menjadi negative, sehingga *D1*merupakan *decision boundary* yang dihasilkan kelas A dan elas B.

B

B

B

B

D2

C

C

C

C

Gambar 2.10 *Decision Boundary D2* pada pendekatan OVO

Lalu selanjutnya, sesuai Gambar 2.10, dibuat *decision boundary* dari kelas B dengan kelas C. Kelas B dilabelkan sebagai positif, sedangkan kelas C dilabelkan menjadi negative, sehingga *D2* merupakan *decision boundary* yang dihasilkan dari kelas B dan kelas C.

A

A

C

C

C

A

A

D3

Gambar 2.11 *Decision Boundary D3* pada pendekatan OVO

Lalu yang terakhir sesuai dengan Gambar 2.11, dibuat *decision boundary* dari kelas A dengan kelas C. Kleas A dilabelkan sebagai positif, sedangkan kelas C dilabelkan menjadi negative *D3*merupakan *decision boundary* yang dihasilkan dari kelas A dan kelas C.

A

D1

B

B

A

A

B

B

A

D3

C

C

D3

C

C

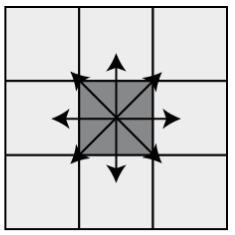
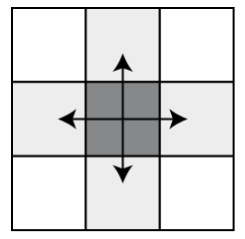
Gambar 2.12 Hasil pendekatan OVO

Sehingga metode OVO menghasilkan area seperti Gambar 2.12, yang dibentuk oleh *decision boundary D1, D2 dan D3.*

**II.7 Chain Code**

Pada pengenalan objek, *chain code* adalah suatu atribut unik yang dapat diekstraksi dari suatu objek di dalam citra dengan cara menelusuri *pixel* batas objek tersebut berdasarkan arah-arah yang telah ditetapkan. *Chain code* pertama kali diusulkan oleh Freeman pada tahun 1961. Keluaran dari pengambilan *chain code* adalah angka-angka yang menunjukkan arah, yang merepresentasikan batas objek. Untuk mempermudah penentuan batas objek, pencarian *chain code* hanya dapat dilakukan pada citra biner. Pada kasus ini, objek dapat didefinisikan sebagai kumpulan *pixel* hitam yang terhubung membentuk satu kelompok.

Ada dua skema ketetanggaan yang dapat digunakan dalam pencarian *chain code*: *4-  
connectivity* dan *8-connectivity*. Skema *4-connectivity* hanya memperbolehkan pemeriksaan ketetanggaan pada empat arah utama (atas, kanan, bawah, dan kiri). *8-connectivity* memperbolehkan pemeriksaan ketetanggaan pada 8 arah (atas, atas-kanan, kanan, kanan-bawah, bawah, kiri-bawah, kiri, dan kiri-atas). Gambar 2.2 mengilustrasikan perbedaan kedua skema ini.



Gambar 2.2 Chain Code dengan 4 – Connectivity dan 8 – Connectivity

Sistem ini menggunakan skema *8-connectivity* karena *chain code* yang dihasilkan lebih fleksibel dibandingkan dengan menggunakan skema *4-connectivity*. Untuk memangkuskan perubahan koordinat titik observasi, arah *chain code* dapat digunakan  
untuk mengubah koordinat x dan y dari titik observasi. Pada aplikasi ini, pemeriksaan pada setiap iterasi dimulai dari *pixel* di atas *pixel* yang sedang diperiksa dikunjungi dan dilakukan searah jarum jam (*clockwise*). Setiap arah diberi nilai antara 1 hingga 8, dengan 1 menyatakan arah awal kiri atas (atas, ↖), 2 menyatakan arah selanjutnya kearah atas dan seterusnya hingga 8 menyatakan arah terakhir yakni kiri.

*Chain code* memiliki empat kekurangan utama sebagai fitur rekognisi:

1. *Chain code* tidak bersifat *scale-invariant*. Objek yang sama dengan ukuran yang berbeda akan memiliki *chain code* yang berbeda, sehingga *chain code* huruf A dengan ukuran 36 tidak bisa digunakan untuk mengenali huruf A dengan ukuran 12.
2. *Chain code* tidak fleksibel. Objek dengan tipe yang berbeda akan menghasilkan *chain code* yang berbeda, contohnya huruf A dengan font Arial akan menghasilkan chain code yang berbeda dengan huruf A pada font Times New Roman, padahal keduanya adalah huruf A. Hal tersebut akan mengurangi keakuratan pengenalan, atau malah membuat pengenalan jadi tidak bisa dilakukan.

3. *Chain code* tidak dapat mendeteksi bagian dalam dari objek. Jika suatu objek memiliki satu atau lebih “lubang” yang membedakan objek tersebut dengan objek lain yang mirip, *chain code* tidak dapat digunakan. Hal ini terjadi karena *chain code* hanya memeriksa batas luar objek. Sebagai contoh, angka 0 *7-segment* dan angka 8 *7-segment* memiliki *chain code* yang sama (dengan asumsi ukurannya juga sama), namun seharusnya diklasifikasikan sebagai dua objek yang berbeda.

4. Tidak dapat merepresentasikan percabangan. Untuk merepresentasikan percabangan pada suatu objek (terutama pada objek yang sudah menjadi kerangka), *chain code* harus dibagi menjadi beberapa bagian, sesuai dengan jumlah cabang yang terbentuk.

## II.8 Thinning

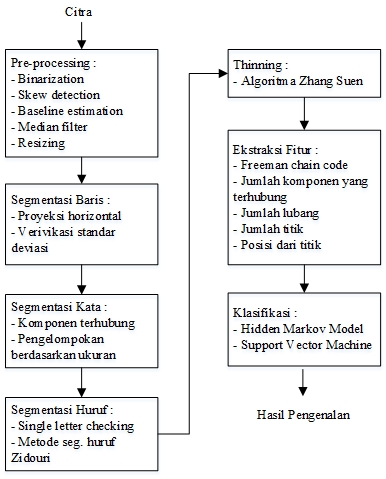
*Thinning* (penipisan) adalah proses mengurangi suatu obyek didalam citra digital menjadi ukuran yang minimum (objek (region) direduksi menjadi rangka (skeleton)) . *Thinning* merupakan operasi morfologi yang digunakan untuk menghilangkan bagian depan piksel dari citra biner, yang dapat digunakan pada beberapa aplikasi namun sangat berguna untuk skeletonisasi [9]. Pada penelitian ini akan dilakukan Thinning dengan Algoritma Zhang Suen.

**Bab III**

**Perancangan Pengujian**

## 3.1 Perancangan Awal Sistem dan Eksperimen Pengenalan Huruf Arab

Sistem pengenalan huruf Arab terdiri dari tahap *pre-processing,* segmentasi baris, segmentasi kata, segmentasi huruf, *thinning,* ekstraksi fitur dan klasifikasi. Gambaran umum dari sistem diberikan oleh gambar berikut.



Gambar 3.1 Diagram blok utama dalam sistem pengenalan huruf Arab

Dalam penelitian ini dikenal data latih dan data uji. Berikut penjelasan dari kedua jenis data tersebut.

1. Data latih

Data latih merupakan kumpulan data huruf arab sesuai sesuai posisinya pada kalimat yakni terpisah, di awal, di tengah, dan di akhir yang akan dilatih untuk kemudian disimpan sebagai *data set* huruf arab.

1. Data uji

Data uji merupakan kumpulan data kalimat Arab yang akan dimasukan dalam tahap pengujian untuk kemudian di segmentasi, diambil fiturnya dan dibandingkan dengan *data set* dari data latih dan ditentukan kemiripan data huruf Arab tersebut.

Jumlah huruf arab yang akan dilatih sebanyak 29 huruf arab, 1 huruf arab memiliki 4 posisi yang berbeda-beda, yaitu posisi terpisah, diawal, ditengah dan diakhir. Berikut teknik yang digunakan dalam pengambilan data latih.

1. Ketik huruf 29 hurufa Arab dari ا sampai ى menggunakan Microsoft Word , huruf arab ditulis sesuai posisinya pada kalimat, yakni posisi terpisah, di awal, di tengah dan di akhir.
2. Potong setiap huruf menggunakan Snipping Tool
3. Hasil pemotongan huruf disimpat sebagai *data set* huruf arab

Saat pengambilan data uji, dilakukan dalam tiga tipe tingkat kesulitan, yaitu:

1. Pengujian huruf

Saat pengujian huruf ini dilakukan sebanyak 29 huruf arab, dimana 1 huruf memiliki 4 posisi yang berbeda, yakni posisi terpisah, diawal, ditengah dan diakhir.

1. Pengujian kalimat

Saat mengujian kalimat dilakukan dengan memasukkan 10 kalimat yang berbeda kedalam sistem. Kalimat kemudian di segmentasi dan diambil fiturnya. Hasil pengujian kalimat berupa persentase keberhasilan segmentasi dan persentase pengenalan huruf Arab.

**3.2 Perancangan sistem dan Pre- Processing**

Input dari tahap preprocessing adalah dokumen gambar. Tahap ini mengimplementasikan beberapa proses yakni *binarization, skew detection,* *baseline estimation*, *median filter,* dan *resizing.* Berikut penjelasan masing – masing tahap.

1. Binarization : Mengubah gambar jadi *binary* yakni 0 dan 1.
2. Skew detection : Mengkoreksi gambar yang miring
3. Baseline estimation :
4. Median filter :
5. Resizing :

**3.3 Perancangan Sistem dan Eksperimen Ekstraksi Fitur**

Sistem ekstraksi fitur dalam penelitian ini menggunakan *Freeman Chain Code*, jumlah komponen yang terhubung, jumlah lubang, jumlah titik, dan posisi titik.

1. Ekstraksi fitur dengan freeman chain code

Tahap ekstraksi fitur dengan Freeman chain code adalah sebagai berikut.

1. Cari objek dengan cara menelusuri pixel pada citra dimulai dari pojok kiri atas sampai  
ditemukan pixel yang berwarna hitam (merupakan pixel batas objek).  
2. Mulai telusuri batas objek, dimulai dari pixel batas objek yang pertama ditemukan sampai kembali lagi ke pixel tersebut atau tidak ada pixel hitam lain yang dapat dikunjungi. Tandai setiap pixel yang menjadi bagian dari objek tersebut (semua pixel hitam yang dapat mencapai pixel batas objek pertama yang didapatkan). Hal ini sama dengan proses segmentasi objek pada suatu citra.  
3. Keluarkan chain code dari objek tersebut.

4. Lanjutkan penelusuran citra, dengan melewati semua pixel yang telah dicatat sebelumnya.Lakukan hingga pixel berwarna hitam yang belum tercatat ditemukan sebagai pixel batas objek baru.

5. Ulangi algoritma mulai dari langkah 2.

2. Ekstraksi fitur dengan jumlah komponen terhubung :

3. Ekstraksi fitur dengan jumlah lubang

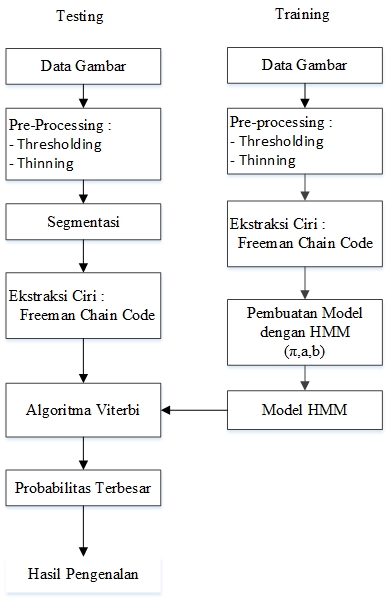
4. Ekstraksi fitur dengan jumlah titik

5. Ekstraksi fitur dengan posisi titik

**3.3 Perancangan Sistem dan Eksperimen Pengklasifikasi**

**3.3.1 Tahap Pelatihan dan Pengujian dengan Metode Hidden Markov Model**

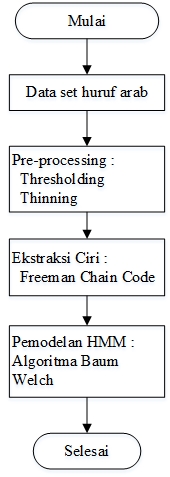
Dalam penelitian ini menggunakan Hidden Markov Model (HMM) untuk tahap klasifikasi. Secara umum, diagram blok dari proses pengenalan huruf arab menggunakan metode Hidden Markov Model dalam Gambar 3.2.



Gambar 3.2 Proses pengenalan huruf arab menggunakan Hidden Markov Model

1. Tahap pelatihan dengan metode Hidden Markov Model

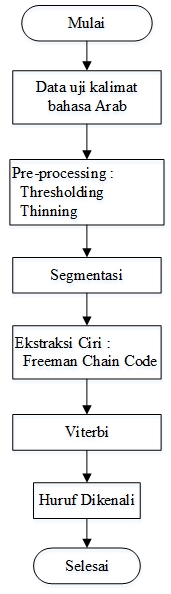
Tahap pelatihan dilakukan untuk menentukan parameter estimasi, sehingga  
terbentuk model markov tersembunyi berupa . Berikut diagram alir pelatihan dengan metode model markov tersembunyi.



Gambar 5.11 Diagram alir pelatihan dengan metode *Hidden Markov Model* (HMM)

1. Tahap pengujian dengan metode model markov tersembunyi

Pengujian dilakukan untuk mengetahui seberapa besar nilai peluang data suara uji  
terhadap kecocokannya dengan data latih. Jika dalam tahap pelatihan, inisialisasi  
matriks transisi dan matriks emisi diisi dengan acak, sedangkan untuk tahap  
pengujian, inisialisasi matriks transisi dan matriks emisi berasal dari data proses  
pelatihan. Di awal proses pengujian tetap dilakukan ekstraksi ciri, kemudian keluarannya dijadikan *chain code.* Proses pengujian atau evaluasi ini menggunakan algoritma *viterbi* karena yang ingin dicari hanya nilai peluang dari data uji di setiap basis data hasil pelatihan. Nilai peluang yang paling tinggi, menentukan kecocokan antara data uji terhadap basis data latih. Berikut algoritma *viterbi* yang digunakan dalam tahap pengujian dengan metode HMM.



Gambar 5.12 Diagram alir pengujian dengan Hidden Markov Model

Tahapan pertama adalah memasukkan gambar rangkaian kata dalam huruf Arab. Terhadap gambar tersebut dilakukan pemrosesan awal guna membersihkan gambar huruf dari kerusakan atau karena posisi yang tidak sesuai serta mengeliminasi hal-hal yang tidak diperlukan. Warna gambar kemudian diubah ke dalam warna abu- *(gray scale)*, dan selanjutnya diubah lagi ke dalam warna hitam-putih, selanjutnya dilakukan proses *thinning* dengan algoritma *Zhang Suen* yang berguna untuk mendapatkan jalur huru, kemudian dilakukan tahapan segmentasi dari kalimat menjadi huruf. Huruf yang telah berhasil di segmentasi dimasukkan kedalam metode *Freeman Chain code* dalam tahap ekstraksi fitur pengambilan ciri masing-masing huruf. Hasil dari ekstraksi fitur ini akan digunakan sebagai masukan untuk tahap klasifikasi dengan *Hidden Markov Model* (HMM) yang merupakan proses identifikasi masing-masing karakter dan menetapkan kedalam kelasnya sendiri.