# ABSTRAK

OCR (Optical Character Recognition) berfungsi untuk men scan gambar pada image dan dijadikan text, dan aplikasi ini juga bisa menjadi support atau aplikasi tambahan untuk scanner. Kinerja pengenalan tergantung pada kualitas input teks dan teknik yang berbeda yang digunakan untuk meningkatkan pengenalan dari karakter. Pada penelitian ini akan dirancang sistem yang dapat mengenali huruf arab. Adapun tahap yang dilakukan ialah Pre-processing, segmentasi, ekstraksi, dan klasifikasi. Pada tahap pre-processing gambar diubah menjadi abu – abu kemudian di jadikan hitam putih dengan metode *otsu threshold*. Pada tahap segmentasi dilakukan tiga proses yakni segmentasi baris, segmentasi kata dan segmentasi huruf. Pada tahap feature ekstraksi digunakan metode fremanchain code untuk mendapat ciri dari tiap – tiap fitur. Kemudian pada tahap klasifikasi kami menggunakan metode Hidden Markov Model (HMM). Berdasarkan hasil penelitian sistem mampu mengenali huruf arab dalam rangkaian kata.

# ABSTRACT

# OCR (Optical Character Recognition) allows you to scan an image on the image and text used, and these applications can also be a support or additional applications for the scanner. Recognition performance depends on the quality of input text and different techniques are used to improve the recognition of the characters. This research will be designed a system that can recognize the Arabic alphabet. The stage is done is pre-processing, segmentation, extraction and classification. At the stage of pre-processing the image is converted into gray - then made in black and white with Otsu threshold method. At this stage of the process of segmentation is done three lines of segmentation, segmentation and segmentation letter word. At this stage of feature extraction methods fremanchain code used to obtain the characteristics of each - each feature. Then, at the stage of our classification using Hidden Markov Model (HMM). Based on the research system is able to recognize the Arabic alphabet in a series of words.

# PEDOMAN PENGGUNAAN THESIS

Tesis S2 yang tidak dipublikasikan terdaftar dan tersedia di Perpustakaan Institut Teknologi Bandung, dan terbuka untuk umum dengan ketentuan bahwa hak cipta ada pada pengarang dengan mengikuti aturan HaKI yang berlaku di Institut Teknologi Bandung. Referensi kepustakaan diperkenankan dicatat, tetapi pengutipan atau peringkasan hanya dapat dilakukan seizin pengarang dan harus disertai dengan kebiasaan ilmiah untuk menyebutkan sumbernya.

Sitasi hasil penelitian Tesis ini ditulis dalam bahasa Indonesia sebagai berikut:

Radhiah, Ainatul. (2016): *Pembelajaran Huruf dan Kalimat Arab Berbasis Pengenalan Citra,* Tesis Program Magister, Institut Teknologi Bandung.

dan dalam bahasa Inggris sebagai berikut:

Radhiah, Ainatul. (2016): *Pembelajaran Huruf dan Kalimat Arab Berbasis Pengenalan Citra,* Master’s Program Thesis, Institut Teknologi Bandung.

Memperbanyak atau menerbitkan sebagian atau seluruh tesis haruslah seizin Dekan Sekolah Pascasarjana, Institut Teknologi Bandung.

# KATA PENGANTAR

*Alhamdulillaahi Robbil’alamin*, penulis ucapkan syukur yang setinggi-tinggi ke-hadirat Allah SWT, karena atas segala limpahan rahmat dan karuniahnya yang diberikan sehingga penulis dapat menyelesaikan penelitian sekaligus penulisan laporan tesis ini. *Allahumma sholli’ala Muhammad wa’ala ali sayyidina Muhammad*, yang tidak lupa penulis haturkan juga untuk junjungan alam, kekasih Allah, Rasul Allah, dan tauladan kita yakni Nabi Muhammad SAW.

Tesis ini disusun guna memenuhi persyaratan untuk menyelesaikan  
jenjang magister dari Program Studi Magister Teknik Elektro, Institut Teknologi  
Bandung. Selama menyelesaikan tugas akhir ini, penulis telah banyak mendapatkan bantuan, bimbingan, dan petunjuk dari banyak pihak baik secara langsung maupun tidak langsung. Dalam kesempatan ini penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada:

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 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, Bening Qias Ranum, Ulfa Asmi, Agis Baswara, Arif 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, Desember, 2016 |
| Penulis |
|  |

# DAFTAR ISI

[ABSTRAK i](#_Toc467765768)

[ABSTRACT ii](#_Toc467765769)

[PEDOMAN PENGGUNAAN THESIS iii](#_Toc467765770)

[KATA PENGANTAR iv](#_Toc467765771)

[DAFTAR ISI vi](#_Toc467765772)

[DAFTAR LAMPIRAN vii](#_Toc467765773)

[DAFTAR GAMBAR viii](#_Toc467765774)

[DAFTAR TABEL ix](#_Toc467765775)

[Bab 1 Pendahuluan 1](#_Toc467765776)

[I.1 Latar Belakang 1](#_Toc467765777)

[I.2 Tujuan 2](#_Toc467765778)

[I.3 Batasan Masalah 2](#_Toc467765779)

[I.4 Metode Penelitian 2](#_Toc467765780)

[I.5 Sistematika Penulisan 3](#_Toc467765781)

[Bab II Tinjauan Pustaka 4](#_Toc467765782)

[II.1 Optical Character Recognition (OCR) 4](#_Toc467765783)

[II.3 Markov Model 4](#_Toc467765784)

[II.3 Hidden Markov Model (HMM) 5](#_Toc467765785)

[II.3 Chain Code 9](#_Toc467765786)

[II.4. Thinning 10](#_Toc467765787)

[Bab III Perancangan Sistem dan Eksperimen 13](#_Toc467765788)

[Bab IV Pengujian dan Pembahasan 14](#_Toc467765789)

[IV.1 Pengujian Sistem 14](#_Toc467765790)

[IV.2 Pembahasan 14](#_Toc467765791)

[Bab V Kesimpulan dan Saran 15](#_Toc467765792)

[V. 1 Kesimpulan 15](#_Toc467765793)

[V.2 Saran 15](#_Toc467765794)

[DAFTAR PUSTAKA 16](#_Toc467765795)

# DAFTAR LAMPIRAN

# DAFTAR GAMBAR

# DAFTAR TABEL

# Bab 1 Pendahuluan

## I.1 Latar Belakang

Al-Quran merupakan firman Allah yang diwahyukan kepada Nabi Muhammad SAW melalui malaikat Jibril, diturunkan pertama kali pada 17 Ramadhan, di Gua Hira’ dalam Bahasa Arab. Al-Qur'an turun secara berangsur-angsur selama 22 tahun 2 bulan 22 hari, terdiri atas 30 juz,114 surah. Alqur’an ditulis dengan bahasa Arab. Huruf arab terdiri dari 28 huruf , ditulis dari kanan ke kiri. Bentuk dari setiap huruf berebda sesuai dengan posisinya pada kalimat, berdiri sediri, diawal, ditengah atau di akhir. Pada penelitian ini akan dirancang sebuah sistem yang bisa mengenali huruf arab pada rangkaia kata dengan teknik *Optical Character Recognition* (OCR). Optical Character Recognition yang disingkat sebagai OCR ialah sebuah aplikasi yang dapat melakukan scan pada gambar berisi teks, kemudian aplikasi akan memproses gambar tersebut hingga menjadi teks yang dapat di edit. dimanipulasi, dan dapat dicari perkata dan perkalimat. Manfaat dari Optical Character Recognition (OCR) diantaranya adalah untuk membaca plat mobil otomatis, menyortir surat dan mengedit dokumen [2], proses pemerik saan Bank, pengecekan tanda tangan, dan identifikasi produk [3]. 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 penilitian tentang Bahasa arab, ini dikarenakan terdapat kesulitan dalam pengenalan Bahasa arab, yakni pada bagian segmentasi, karakteristik tulisan Bahasa arab yang bersambung membuat tahap segmentasi huruf menjadi sulit. Oleh karena itu perlu teknik yang canggih yang dikembangkan untuk meakukan segmentasi huruf dalam Bahasa arab. Pemotongan sebuah kata menjadi huruf tanpa sebuah kesalahan merupakan hal sulit karena huruf di dalam tulisan terhubung sepenuhnya dan tidak memiliki parameter yang jelas untuk menemukan batas huruf yang tepat.

## I.2 Tujuan

Tujuan dari penelitian pada tahap ini adalah membangun sebuah 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.

## I.3 Batasan Masalah

Untuk menyederhanakan masalah dan efisiensi waktu serta biaya, dalam penelitian  
ini terdapat pembatasan masalah 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.
3. Laptop dengan processor core i3 dengan RAM 8GB

## I.4 Metode Penelitian

Dalam menyelesaikan penelitian tesis ini diperlukan langkah – langkah sebagai  
berikut:

1. Studi literature

Mempelajari dan menganalisis berbagai sumber informasi seperti buku – buku dan literature 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

## I.5 Sistematika Penulisan

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).*

Bab III Perancangan Sistem dan Eksperimen. Bab ini membahas perancangan awal sistem dan eksperimen *Optical Character Recognition*, dan *Hidden Markov Model (HMM)*

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) adalah aplikasi yang berfungsi untuk men scan gambar pada image dan dijadikan text, dan aplikasi ini juga bisa menjadi support /aplikasi tambahan untuk scanner. Dengan adanya OCR, Image yang bertulisan tangan, tulisan mesin ketik atau computer text, dapat dimanipulasi. Text yang discan dengan OCR dapat dicari kata per kata atau per kalimat. Dan setiap text dapat dimanipulasi, diganti, atau diberikan barcode.

Pada OCR, karakter pada dokumen yang telah dipindai menggunakan scanner atau alat pindai optis lainnya akan melalui 3 tahap pada sistem pengenalan [3], yaitu sebagai berikut.

1. Tahap preprocessing untuk meningkatkan kualitas citra masukan dan karakter yang akan dikenali.

2. Tahap Ekstraksi Ciri untuk mengambil ciri pembeda dari citra karakter digital untuk pengenalan.

3. Tahap Klasifikasi untuk memproses vektor ciri yang mengidentifikasikan citra karakter dan kata. Angka tulisan tangan merupakan salah satu karakter cukup sulit untuk dikenali. Setiap orang memiliki cara penulisan angka yang berbeda, bahkan karakter angka yang ditulis oleh seseorang pun tidak selalu sama. Keberagaman tulisan tangan angka ini menjadi salah satu topik yang menantang dalam riset OCR [12].

**II.3 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.3.1 *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.3.2 *Unsupervised Learning***

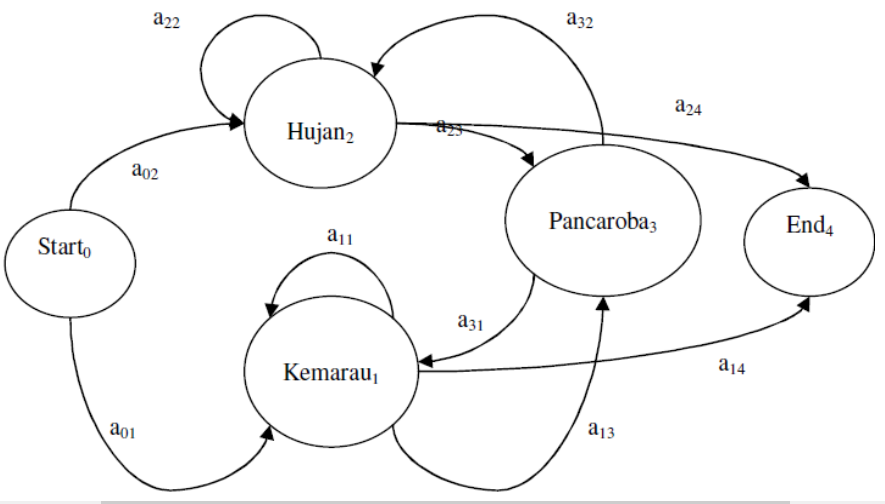
Pada *unsupervised Learning,* data training tidak disertai target pada setiap datanya, {xi}, i = 1…N, dengan **x** adalah vector input. Tujuan dari *unsupervised learning* adalah membangun model yang dapat menemukan variable tersembunyi dari *data training*. *Unsupervised learning* digunakan untuk *clustering,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

## 

## II.4 Markov Model

*Markov Model* ditemukan oleh Andrey Markov dan merupakan bagian dari proses stokastik yang memiliki property *Markov*. Dengan memiliki property tersebut berarti apabila diberikan inputan keadaan saat ini, keadaan akan datang dapat diprediksi dan ia lepas dari keadaan masa lampau. Artinya, deskripsi kondisi saat ini menangkap semua informasi yang mempengaruhi evolusi dari suatu sistem dimasa depan. Dengan kata lain Kondisi masa depan dituju dengan menggunakan probabilitas bukan determinitas [12].

Model ini merupakan bagian dari *finite automaton*. *Finite automation* sendiri adalah kumpulan state yang transisi antar statenya dilakukan berdasarkan masukkan observasi. Pada Markov Chain, setiap busur antar state berisi probabilitas yang mengindikasikan kemungkinan jalur tersebut akan diambil. Jumlah probabilitas semua busur yang keluar dari sebuah simpul adalah satu. Gambar dibawah memperlihatkan contoh Markov Chain yang menggambarkan kondisi cuaca.



Gambar 2.1 Markov Chain yang menggambarkan perubahan cuaca

Markov Chain bermanfaat untuk menghitung probabilitas urutan kejadian yang dapat diamati. Permasalahannya terkadang ada urutan kejadian yang ingin diketahui tetapi tidak dapat diamati. Untuk menyelesaikan kasus tersebut, dikembangkanlah model baru yang memodelkan kejadian yang tersembunyi, Hidden Markov Model.

## II.5 Hidden Markov Model (HMM)

HMM adalah sebuah proses stokastik dimana salah satu prosesnya tidak dapat diobservasi (*hidden*). Proses yang tidak dapat diobservasi ini hanya dapat diobservasi melalui proses yang dapat di observasi. Sebuah HMM menggabungkan dua atau lebih rantai Markov dengan hanya satu rantai yang terdiri dari state yang dapat diobservasi dan rantai lainnya membentuk state yang tidak dapat diobservasi (*hidden*), yang mempengaruhi hasil dari state yang dapat diobservasi.[10]

Terdapat tiga masalah utaa yang dapat diselesaikan oleh Hidden Markov Model, yaitu :

1. Evaluasi

Operasi evaluasi dalam HMM adalah perhitungan probabilitas dari urutan nilai observasi yang diberikan oleh *Hidden Markov Models.* Untuk menghitung peluang terjadinya suatu barisan observasi dibutuhkan algoritma yang lebih efisien untuk menyelesaikan masalah evaluasi. 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]

Algoritma *Forward*

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 dan = matrik transisi.

**Terminasi :**

(II.4)

Algoritma *Backward*

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

(20) (II.5)

Dan dianalogikan dengan prosedur *forward* dengan langkah:

**Inisialisasi**

(21) (II.6)

**Induksi**

, (II.7)

, (II.8)

1. Decoding (Menentukan Barisan Keadaan Tersembunyi)

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 banyak digunakan untuk penyelesaian masalah ini antara lain algoritma Viterbi dan Entropi.

**Inisialisasi :**

, (II.9)

**Rekursi :**

, (II.10)

**Terminasi :**

(II.11)

Dimana P = Probabilitas

1. Learning (Menaksir Parameter – parameter HMM)

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 probabilistic dalam suatu kasus yang melibatkan keadaan – keadaan tersembunyi.

**Parameter A**

(II.12)

, peluang ketika state *j* pada waktu *t+1* jika pada waktu *t*  berada di state *I* (II.13)

= Probabilitas transisi dari state I ke state j

*P*  = probabilitas

*n* = banyaknya hidden state dalam model

**Parameter B**

(II.14)

(II.15)

dan (II.16)

= Probabilitas distribusi matriks observasi

*m* = banyaknya symbol 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 adalah *number of mixture* dan adalah *mixing coefficient* untuk *Gaussian* pada *state* j dengan batasan stokastik :

(II.18)

dimana menunjukkan *Gaussian density function* dengan mean dan matrik kovarian untuk *mixture*.

**Matriks awal di state *i***

(II.19)

Sehingga HMM dapat dilambangkan dengan [13].

## II.6 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 beriukut,

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 2.2

Untuk mendapatkan jarak dari bidang B+ dengan data yang berlabel positif, dapat dilakukan dengan mencari jarak minimum antara data yang mempunyai target positif terdekat dengan bidang B+. Sehingga secara matematis kita dapat meminimumkan xTx dengan syarat wTxi + b 1. Dengan pengali Lagrange (λ) didapatkan bentuk.

L = xTx + λ (-wTx – b + 1) (II.26)

Kemudian turunkan fungsi L terhadap x sehingga didapatkan hasil

= 0 = xT- λwT

xT = λwT x = λw (II.27)

subtitusikan x pada persamaan (II.27) ke dalam persamaan bidang (II.24) B+ : wTxi + b = 1, sehingga menjadi

wT λw + b = 1

λ = (II.28)

Subtitusikan λ pada persamaan (II.28) kedalam **x** pada persamaan (II.27) menjadi

x = = (II.29)

sehingga didapat persamaan untuk xTx sebagai berikut :

xTx = (II.30)

Jadi, jara terdekat bidang B+ ke bidang pemisah adalah

S1 = ||x|| = = = (II.31)

Sedangkan untuk mendapatkan jarak dari bidang B- dengan data yang berlabel negative, dapat dilakukan dengan mencari jarak minimum antara data yang mempunyai target negative terdekat dengan bidang B-. Sehingga secara matematis kita dapat meminimumkan xTx dengan syarat wTxi + b -1. Dengan pengali Lagrange (λ) didapatkan bentuk.

L = xTx + λ (wTx – b + 1) (II.32)

Lalu turunkan fungsi L terhadap x sehingga didapatkan

= 0 = xT+ λwT

xT =- λwT x = - λw (II.33)

subtitusikan x pada persamaan (II.33) ke dalam persamaan bidang (II.25) B- : wTxi + b = -1, sehingga menjadi

-wT λw + b = -1

λ = (II.34)

Subtitusikan λ pada persamaan (II.3.4) kedalam **x** pada persamaan (II.33) menjadi

x = = (II.35)

sehingga didapat persamaan untuk xTx sebagai berikut :

xTx = (II.36)

Jadi, jarak terdekat bidang B- ke *hyperline* adalah

S1 = ||x|| = = =

Untuk mencari margin yang sesuai, diperlukan jarak yang maksimum dari data yang terdekat dengan *hyperline* dari masing – masing kelas. Dari persamaan (II.31 dan II.36) dapat dibentuk,

=

(II.37)

Sehingga persamaan dapat dibentuk menjadi,

(II.38)

Dengan syarat :

min wTxi + b = 1 , yi = +1

max wTxi + b = -1 , yi = -1

sehingga akan menghasilkan fungsi klasifikator sebagai berikut :

y = *sign*(wTx + b) (II.39)

Masalah (II.38) juga dapat ditulis sebagai berikut :

atau (II.40)

Dengan syarat :

yi (wTxi + b) ≥1 (II.41)

Dalam pembentukan fungsi klasifikator, ada beberapa yang tidak dapat diklasifikasikan secara benar atau permasalahan optimasinya bersifat tidak layak. Untuk menghindari masalah ini, ditambahkan variable *slack* ( ≥ 0) pada masalah optimasi (II.40) sehingga menjadi :

(II.42)

Dengan syarat :

yi (wTxi + b) ≥1 - i (II.43)

i ≥ 0 , (II.44)

Dimana C akan mengontrol *tradeoff* antara variable slack dan margin.

Permasalahan (II.42) merupakan bentuk *quadratic programing*. Untuk menyelesaikan pemograman kuadrat tersebut, cara yang umum digunakan adalah mencari bentuk dual dengan menggunakan pengali Lagrange *(Lagrange multipliers)*. Dari permasalahan diatas, bentuk *Lagrange*-nya adalah sebagai berikut :

L (w, α, λ) = ||w||2 + C i+ i (1 - yi (wTxi + b - i ) + - i )

= ||w||2 + C i + i - i yi (wTxi + b) - i i - i i (II.45)

Dengan kendala variable dualnya adalah sebagai berikut

αi ≥ 0 , i ≥ 0 (II.46)

Pada bentuk dual (II.45), αi dan i adalah pengali *Lagrange.*

Soulusi dari masalah optimasi (II.45) dapat ditentukan melalui mencari *saddle point*-nya dengan cara menghitung turunan parsial dari *L* yang sama dengan 0 sebagai berikut :

* Turunan terhadap **w :**

= w - i yi xi = 0

w = i yi xi (II.47)

* Turunan terhadap *b* :

= i yi  = 0 (II.48)

* Turunan terhadap i :

= αi - i  = 0 (II.49)

Dengan syarat i ≥ 0, αi ≥ C.

Dengan mensubtitusikan turunan parsial L (II.47), (II.48) dan (II.49) ke masalah dual (II.45) menghasilkan :

Maks L = (iyixi) (iyixi) + C i + i - iyi ((iyixi) xi + b) - ii - ii

= i yi αj yj (xi . xj) -  i yi αj yj (xi . xj) - b i yi + i + C i - i i - i i

= i - i yi αj yj (xi . xj) (II.50)

Dengan syarat

0 ≤ αi ≤ C, I = 1, …, N (II.51)

i yi = 0 (II.52)

Meskipun telah menggunakan variable *slack* tidak semua data merupakan data yang dapat dipisahkan secara linier oleh *hyperlane*.Oleh Karena itu untuk mengatasi masalah ini digunakan fungsi kernel yang berfungsi untuk memetakan data ke dimensi yang lebih tinggi, sehingga diharapkan data menjadi bersifat linearly separable. Fungsi kernel Radial Basis Function (RBF) secara umum direkomendasikan sebagai pilihan utama, sehingga pada penelitian ini akan digunakan fungsi kernel RBF. Hal ini dikarenakan 3 hal yaitu :

1. Fungsi kernel RBF dapat memetakan secara tidak linear data ke ruang dimensi yang lebih tinggi, sehingga diharapkan dapat menangani kasus di mana relasi antara kelas dan fitur yang tidak linear.
2. Fungsi kernel RBF menggunakan lebih sedikit parameter
3. Fungsi kernel RBF memiliki kesulitan numerik yang lebih sedikit dibandingkan dengan fungsi kernel lain.

Dalam pemakaiannya, data xi akan dipetakan dengan fungsi Ф(xi) dan setiap perkalian (xi . xj) akan dihitung menggunakan K (xi . xj) yang memiliki bentuk sebagai berikut

*K (xi . xj)* = exp (-|| xi - xj||2); > 0, (II.53)

Dengan menambahkan fungsi kernel RBF ke dalam persamaan (II.50), maka bentuk, maka bentuk masalah dual yang baru adalah sebagai berikut :

maks L = i - i yi αj yj K(xi . xj) (II.54)

dengan syarat

0 ≤ αi ≤ C, i = 1, …, N (II.55)

i yi = 0 (II.56)

Permasalahan SVM dalam bentuk dual Lagrange (II.54) dapat diselesaikan dengan menggunakan metode Sequential Minimal Optimization (SMO) (Platt, 1998). Jika solusi dari permasalahan (II.54) adalah α\*, maka **w** dapat dicari dengan subtitusi α\* kedalam persamaan (II.47) sebagai berikut :

w = yi xi  (II.57)

Dengan mensubtitusikan persamaan (II.57) kedalam persamaan (II.20) sehingga fungsi klasifikator (*hyperplane*) adalah persamaan dengan bentuk sebagai berikut:

y = *sign*(wTx + b)

y = *sign* (yi x + b)

y = *sign* (yi K(xi . xj) + b) (II.58)

Pada persamaan (3.8) hanya data dengan > 0 (*support vectors*) yang akan berperan pada model persamaan fungsi klasifikator (*hyperplane*) atau dengan kata lain

y = *sign* (yi K(xi . xj) + b) (II.59)

dengan S adalah himpunan indeks dari *support vectors.*

Bentuk dual dari persamaan (II.59) memenuhi KKT sebagai berikut :

αi ≥ 0, [KKT – 1]

yi(wTxi + b) -1 + i ≥ 0, [KKT – 2]

αi(yi(wTxi + b) – 1 + i ) = 0, [KKT – 3]

i ≥ 0, [KKT – 4]

i ≥ 0, [KKT – 5]

ii = 0 [KKT – 6]

Karena *support vectors* adalah data training dengan αi ≥ 0, maka KKT – 3 menjadi

yi(wTxi + b) -1 + i = 0

yi(wTxi + b) = 1 – i (II.60)

Dari syarat persamaan (II.55) dan dari hasil turunan *L* terhadap i pada persamaan (II.49), akan didapat :

1. Bila< C maka i > 0, dan berdasarkan KKT – 6 maka I = 0 yang berarti data training terletak pada *hyperplane*
2. Bila = C maka i = 0, dan berdasarkan KKT – 6 maka i ≠ 0 yang berarti *data training* terletak di dalam margin, baik terklasifikasi dengan benar (i ≤ 1) maupun salah (i >1).

Dari fakta tersebut, maka b dapat dicari dengan persamaan berikut :

yi(wTxi + b) = 1

yiyi K(xi . xj) + b) = 1

b = i  yi K(xi . xj) (II.61)

dengan S adalah himpunan indeks dari *support vector*, dan Ns adalah banyaknya data train yang menjadi *support vectors.*

**II.6.1 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 2.3

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.6.2 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.6.2.1 *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

D3

C

C

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.6.3**

**Contoh penggunaan SVM**

Misal diberikan *data training* ,} ϵ kelas (-1) dan ϵ kelas (+1), atau dapat digambarkan dengan vitur x1 dan x2 sebagai berikut :

x2

(2,3)

Keterangan :

* berarti yi = - 1

berarti yi = +1

3

(1,1)

1

X1

2

1

## Gambar 3.3 contoh representasi geometric dari data training dalam contoh

Perhatikan bahwa elemen terdekat dari masing – masing kelas adalah elemen Untuk membuat *hyperlane*, maka ambil satu titik yang pasti dilewati yaitu **w** = , yang akan mengimplikasikan vector bobot **w =** .

Ambil ϵ kelas (-1) maka dari persamaan (II.25) yaitu B- : wTxi + b = -1 diperoleh :

wTxi + b = -1 () + b = -1

+ + b = -1

b = -1 - 3 (II. 53)

Ambil ϵ kelas (+1) maka dari persamaan (II.24) yaitu B+ : wTxi + b = 1 diperoleh :

wTxi + b = +1 () + b = 1

+ + b =1

b = 1 - 8 (II. 54)

Dari persamaan (II. 53) dan (II. 54) didapatkan

-1 - 3 = b = 1 - 8

5 = 2

= (II. 55)

Dengan mensubtitusikan persamaan (II. 55) ke persamaan (II. 53), maka diperoleh

b = -1 – 3 = - (II. 56)

dari persamaan (II. 55), maka vector bobot *hyperlane*  menjadi

**w =**  , ) = ( , ) (II. 57)

Sehingga *hyperlane* akan berbentuk sebagai berikut

y (x1, x2) = wTxi + b

y (x1, x2) = ( ) ( ) -

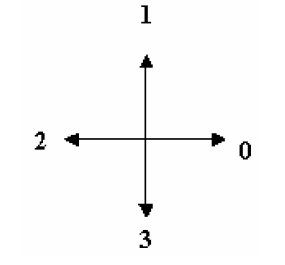
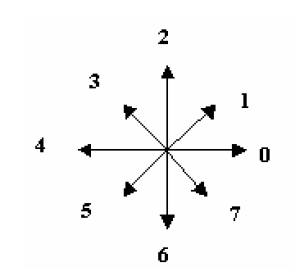
y (x1, x2) = x1 + x2 -

sehingga didapatkan *hyperplane* sebagai berikut

y (x1, x2) = 2x1 + 4x2 = 11

## II.7 Chain Code

*Chain Code* merupakan *shape descriptor* yang berlandaskan tepian atau *border based descriptor*. *Chain Code* yang digagas oleh Freeman pada tahun 1961 ini memberi kode pada tepian dengan cara melihat arah dari *pixel*-*pixel* yang saling terhubung (Sleit, 2006). Proses pemberian kode ini dapat menggunakan pendekatan *4-neighborhood* maupun *8-neighborhood*.

**Gambar 2.4 *4-neighborhood* dan *8-neighborhood Chain Code***

**(Sumber: A Chain Code Approach for Recognizing Basic Shapes oleh Sleit)**

Setiap hubungan antar *pixel* akan dinyatakan dalam kode sesuai dengan aturan *4-neighborhood* atau *8-neighborhood*. Misalkan *pixel* p(x, y) terhubung dengan *pixel* p(x+1, y) di mana x merupakan baris dan y adalah kolom pada koordinat gambar, maka hubungan antara p(x,y) dengan p(x+1, y) akan dikodekan dengan nilai 3 pada *4-neighborhood* atau 6 pada *8-neighborhood*. Metode ini akan menelusuri tepian objek hingga bertemu titik permulaan, menghasilkan serentetan kode yang merepresentasikan objek tersebut.

Representasi *Chain Code* banyak digunakan dalam pengenalan objek karena kompresi informasinya yang dianggap cukup baik. Selain itu representasi *Chain Code* memiliki sifat *invariant* terhadap translasi 2 dimensi, akan tetapi bersifat *variant* terhadap rotasi dan skala serta sangat peka terhadap noise.

## 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, berikut tahap proses dari algoritma Zhang Suen.

1. Gambar hasil biner diubah menjadi array biner dengan warna putih menjadi nilai 0 dan hitam menjadi 1.
2. Kemudian dilakukan proses thinning yang dilakukan dengan Zhang Suen, adapun langkah – langkah algoritma Zhang Suen adalah sebagai berikut.
   1. Asumsi

Nilai 1



Nilai 0

* + 1. Citra adalah *binary image*
    2. Piksel 1 adalah *foreground*
    3. Piksel 0 adalah *background*
    4. Aturan 8 tetangga

Gambar 2.1 Proes Algoritma Zhang Suen (1)

* 1. Keterangan Istilah
     1. N(P1) : Jumlah tetangga P1 bernilai 1
     2. S(P1) : Jumlah perpindahan nilai 0-1; P2,P3…P9
     3. P2 \* P3 = 0 : P2 atau P3 bernilai 0 (nol)



Keterangan :

N(P1) = 3

S(P1) = 2

P2 \* P3 = 0; TRUE

Gambar 2.2 Proes Algoritma Zhang Suen (2)

* 1. Kondisi
     1. 2 ≤ N (P1) ≤ 6
     2. S(P1) = 1
     3. P2 \* P4 \* P6 = 0
     4. P4 \* P6 \* P8 = 0
     5. 2 ≤ N (P1) ≤ 6
     6. S(P1) = 1
     7. P2 \* P4 \* P8 = 0
     8. P2 \* P6 \* P8 = 0
  2. Algoritma thinning :
     1. Ulangi Langkah A sampai D hingga tidak ada perubahan :
     2. Tandai semua piksel foreground yang memenuhi kondisi 1 sampai dengan 4
     3. Rubah nilai piksel menjadi 0 (*background*)
     4. Tandai semua piksel foreground yang memenuhi kondisi 5 sampai dengan 8
     5. Rubah nilai piksel menjadi 0 (*background*)



Langkah A

* + 1. 2 ≤ N (P1) ≤ 6
    2. S(P1) = 1
    3. P2 \* P4 \* P6 = 0
    4. P4 \* P6 \* P8 = 0

Gambar 2.3 Proes Algoritma Zhang Suen (3)

Langkah B



Nilai = 0

Rubah nilai menjadi 0

Gambar 2.4 Proes Algoritma Zhang Suen (4)

Langkah C

* + 1. 2≤ N (P1) ≤ 6
    2. S(P1) = 1
    3. P2 \* P4 \* P8 = 0
    4. P2 \* P6 \* P8 = 0

Gambar 2.5 Proes Algoritma Zhang Suen (5)



Langkah D

1. Rubah nilai menjadi 0
2. Nilai = 0
3. Langkah A-B, tidak berubah

selesai

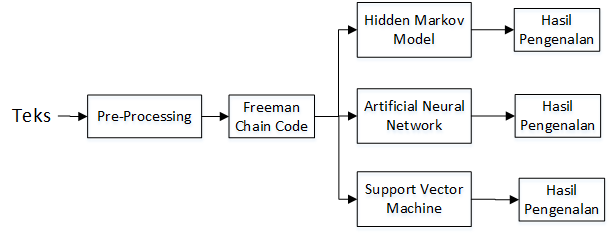
Gambar 2.6 Proes Algoritma Zhang Suen (6)

# 

# Bab III Perancangan Sistem dan Eksperimen

## 3.1 Perancangan Awal Sistem dan Eksperimen Pengenalan Huruf Arab

Sistem pengenalan huruf arab dirancang dengan menggunakan Freeman Chain Code sebagai ekstraksi fiturnya, sehingga didapat chain code dari setiap huruf arab yang kemudian di distribusikan sebagai masukan untuk klasifikasi. Hidden Markov Model (HMM) , Artificial Neural Network (ANN) dan Support Vector Machine (SVM) sebagai pengklasifikasi akan melakukan tugasnya untuk mengenali huruf Arab dari huruf Arab yang diujikan. Gambaran umum dari sistem diberikan oleh gambar berikut.

Gambar III.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 suara responden yang akan dilatih untuk kemudian disimpan sebagai basis data huruf arab

1. Data uji

Data uji merupakan kumpulan data suara responden yang akan dimasukan dalam tahap pengujian untuk kemudian dibandingkan dengan basis data dari data latih dan ditentukan kemiripan data huruf Arab tersebut.

Jumlah huruf arab yang akan dilatih dan diuji sebanyak 29 huruf arab, 1 huruf arab memiliki 4 posisi yang berbeda-beda, yaitu posisi terpisah, diawal, ditengah dan diakhir.

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.

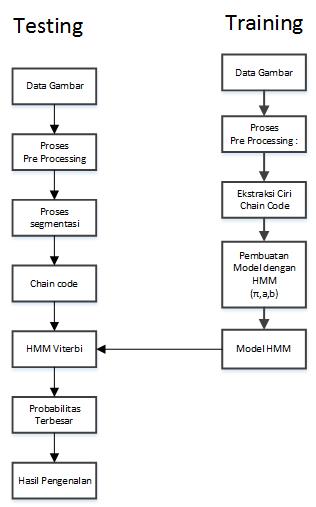
1. Pengujian kata

Pengujian kata dilakukan dengan 5 kata Bahasa Arab yang berbeda.

1. Pengujian kalimat

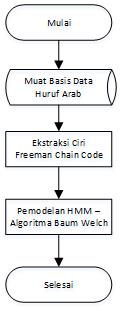
**III.3.2 Tahap Pelatihan dan Pengujian dengan Metode Model Markov Tersembunyi**

Dalam penelitian ini menggunakan Hidden Markov Model, Secara umum, diagram blok dari proses pengenalan huruf arab menggunakan metode model dalam  
Gambar III.5.



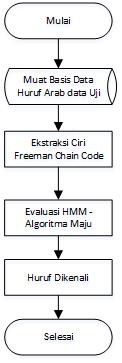
1. Tahap pelatihan dengan metode Hidden Markov Model

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



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  
maju 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. Nilai peluang yang didapat merupakan hasil  
penyekalaan (*scaling*) dikarenakan rentang datanya terlalu jauh (Nilsson, 2005).  
Berikut maju yang digunakan dalam tahap pengujian dengan metode model markov  
tersembunyi.



Tahapan pertama adalah memasukkan gambar rangkaian kata dalam huruif 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 segmentasi segmentasi baris, dan segmentasi huruf. Setelah tahapan segmentasi, dilanjutkan tahapan Thinning dengan algoritma Zhang Suen yang berguna untuk mendapatkan jalur huruf sebelum di masukkan ke 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.

* 1. **Perancangan sistem dan eksperimen pre-processing**

Seperti yang telah disampaikan pada bab sebelumnya, langkah pertama dalam sistem pengenalan karakter adalah pengolahan awal data/citra yang akan diujikan. Tujuan utama langkah ini adalah menghasilkan citra baru yang memiliki ciri-ciri atau fitur yang lebih menonjol dibandingkan citra awal guna memudahkan proses ekstraksi fitur. Tahapan yang dilakukan dalam pengolahan awal ini adalah sebagai berikut:

1. Konversi citra berwana RGB menjadi citra grayscale.

Pada citra berwarna RGB, setiap pixel memiliki nilai warna yang dinyatakan dalam vektor tiga dimensi yang menyatakan karakteristik warna dasar. Konversi citra RGB menjadi grayscale menghasilkan citra baru dengan setiap pixelnya hanya diwakili oleh satu parameter yakni derajat keabuan. Parameter ini dapat bernilai dalam rentang 0-255, dengan 0 merepresentasikan warna hitam dan 255 merepresntasikan warna putih.

1. Konversi citra grayscale menjadi hitam dan putih.

Tahapan berikutnya adalah klasifikasi citra menjadi gambar hitam dan putih berdasarkan histogram citra grayscale. Dari histogram dapat terlihat dua puncak dominan yang menyatakan ada atau tidaknya suatu karakter dalam sebuah citra. Dengan menentukan batas diantara dua puncak tersebut, citra grayscale diubah menjadi gambar hitam dan putih, yang setiap pixelnya hanya memiliki nilai antara 0 (hitam) atau 1 (putih). Langkah ini perlu dilakukan agar proses berikutnya dalam pengenalan karakter dapat dieksekusi.

* 1. **Segmentasi**

Segmentasi pada konteks laporan ini didefinisikan sebagai proses pemotongan suatu citra menjadi komponen-komponen yang lebih kecil. Proses segmentasi yang terjadi dalam sistem pengenalan karakter Arab dapat dibedakan menjadi tiga tahapan, yakni segmentasi baris, kata, dan huruf. Pada tahapan penelitian ini, telah berhasil dilakukan segmentasi baris dan segmentasi huruf.

1. Segmentasi baris

Proses segmentasi garis dilakukan dengan cara melakukan penelusuran citra dari tepi atas hingga barisan pixel terbawah. Barisan-barisan yang seluruhnya terisi oleh pixel-pixel bernilai satu (putih) disinyalir sebagai spasi baris dalam suatu teks, dan dinyatakan sebagai posisi dimulainya baris baru dalam teks.

1. Segmentasi huruf

Tahap segmentasi huruf merupakan tahap tersulit dari proses pengenalan huruf arab, dikarenakan tulisan arab yang bersambung, berikut langkah – langkah segmentasi huruf :

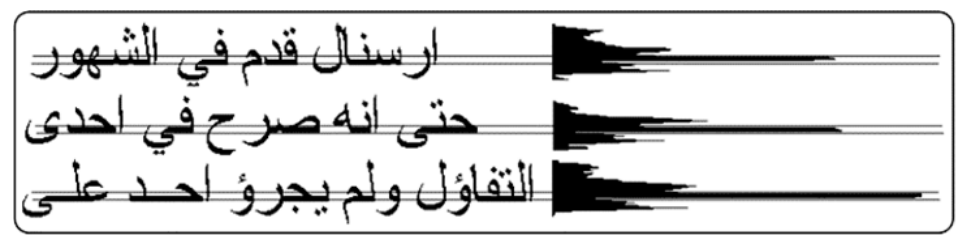
1. Hitung horizontal projection dar sebuah gambar
2. Ambil baseline dari horizontal projection dengan mencari kemunculan titik terbanyak
3. Dari base line di iterasi keatas dan kebawah, apakah ada titik hitam ditemukan , jika tidak ada titik hitam maka akan dilakukan pemotongan.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **0** |  |  |  |  |
| **1** |  |  |  |  |
| **2** |  |  |  |  |
| **3** |  |  |  |  |
| **4** |  |  |  |  |
| **5** |  |  |  |  |

**Gambar 5.5 Contoh huruf Arab yang berada dalam sebuah kotak**

**Gambar 5.6 Contoh Horizontal Projection dari Huruf Arab**

**Gambar 5.7 Contoh Horizontal Projection**

****

**5.8 Proyeksi horisontal untuk teks Arab yang diberikan**

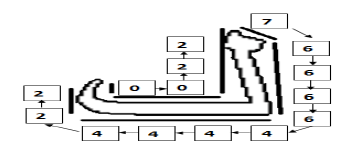
Dari hasil segmentasi huruf diatas pada kata nomor 1 berhasil di segmentasi dengan sempurna, namun pada kata ke – 2 atau ke – 3 , hasil segmentasi yang diperoleh belum sempurna.

* 1. **Thinning menggunakan algoritma Zhang Suen**

Pada tahap ini akan dilakukan Thinning terhadap huruf hasil segmentasi, yakni proses pengambilan tulang suatu gambar agar terlihat jalur suatu gambar tersebut, sebelum digunakan pada metode FreemanChain Code.

* 1. **Ekstraksi Fitur menggunakan Freeman Chain Code**

Pada tahap ini akan dilakukan ekstraksi fitur dengan metode *Freeman Chain Code* untuk membangun representasi obyek dengan rangkaian kode. Kode rantai disusun berdasarkan kenyataan bahwa titik berurutan pada kurva kontinyu yang saling berdekatan satu sama lain, dan bahwa masing–masing titik data secara berurutan berbatasan dengan salah satu dari delapan kode rantai yang mengelilingi titik data tersebut. Kode rantai digunakan untuk merepresentasikan batas tepi dengan urutan garis lurus yang terhubung dengan ukuran dan arah tertentu. Biasanya, kode rantai direpresentasikan dengan 8 arah mata angin. Ilustrasi penelusuran kode rantai ditunjukkan pada Gambar 6.2.



**Gambar 6.2 Contoh Freeman chain code**

Pada tahap chain code ini sistem akan melakukan scan dari kiri atas sampai bertemu titik hitam pertama, kemudian dia akan melakukan cek tetangga, jika tetangganya adalah titik hitam selanjutnya maka dia akan bergerak menuju titik hitam tersebut, jika tetangganya tidak merupakan titik hitam maka tidak akan dikunjungi. kemudian titik hitam yang sudah dilalui akan dijadikan seperti warna background, sehingga tidak dapat dikunjungi kembali.

**5.5 Klasifikasi dengan Hidden Markov Model**

Sistem pengenalan huruf Arab dirancang dengan menggunakan *Freeman Chain Code* sebagai ekstraksi fiturnya, sehingga didapatkan ciri angka yang kemudian akan didistribusikan sebagai masukan untuk klasifikasi. Model markov tersembunyi sebagai pengklasifikasi akan melakukan tugasnya untuk menentukan huruf Arab, dari kalimat yang diujikan. Gambaran umum dari sistem diberikan oleh diagram berikut.



Gambar III.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 suara responden yang akan dilatih untuk kemudian disimpan sebagai basis data huruf arab

1. Data uji

Data uji merupakan kumpulan data suara responden yang akan dimasukan dalam tahap pengujian untuk kemudian dibandingkan dengan basis data dari data latih dan ditentukan kemiripan data huruf Arab tersebut.

**5.6 Tahap Pelatihan dan Pengujian dengan Metode Model Markov Tersembunyi**

# Bab IV Pengujian dan Pembahasan

## IV.1 Pengujian Sistem

Pengujian dilaksanakan dengan memasukkan rangkaian kata Bahasa Arab ke sistem, lalu akan dilakukan proses pengenalan dengan Optical Character Recognition. Hasil eksperimen berupa huruf arab yang dapat dikenali. Diagram blok pengujian perbandingan hasil eksperimen pengenalan pembicara ini diberikan dalam gambar berikut.

## IV.2 Pembahasan

# Bab V Kesimpulan dan Saran

## V. 1 Kesimpulan

## V.2 Saran

# DAFTAR PUSTAKA

[1] Safwa Taha, Y. B. (2012). Optical Character Recognition of Arabic Printed Text. *IEEE Student Conference on Research and Development*, 1-3.

[2] Ismail Bouazizi, F. B.-A. (2013). Arabic Reading Machine for Visually Impaired People using TTS and OCR. *4th International Conference on Intelligent Systems, Modelling and Simulation*, 1 - 3.

[3] Iping Supriana\*, A. N. (2013). Arabic Character Recognition System Development. *4th International Conference on Electrical Engineering and Informatics (ICEEI 2013)*, 5.

[4] Zidouri, A. (2010). On Multiple Typeface Arabic Script Recognition. *Research Journal of Applied Sciences Engineering and Technology 2(5): 428-435, 2010*, 1 - 3.

[5] *Pengertian OCR dan OMR (2)*. (2016, 11 15). Retrieved from Information and Communication Technology Solution: <http://www.skill.co.id/front/index.php/jakarta-tech-update/296-pengertian-ocr-dan-omr-2>

[6] Vidya Dwi Turisqi Wijaya, S. N. (2014). Deteksi Huruf Arab Menggunakan Metode Freeman Chain Code. 2.

[7] Alginahi, Y. M. (2013). A survey on Arabic character segmentation. *International Journal on Document Analysis and Recognition*, 1.

[8] Anwar Ali Sanjrani, J. B. (2016). Handwritten Optical Character Recognition System for Sindhi Numerals. *International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, 3.

[9] Ningsih, S. F. (2016, 11 16). *Penipisan Pola Citra (Thinning)*. Retrieved from Sri Fitri Ningsih: http://srifitri07.blogspot.co.id/2013/12/penipisan-pola-citra-thinning.html

[10] (n.d.). Retrieved from <http://a-research.upi.edu/operator/upload/s_mat_0706660_chapter3.pdf>

[11] <https://www.google.co.id/url?sa=t&rct=j&q=&esrc=s&source=web&cd=13&cad=rja&uact=8&ved=0ahUKEwjXlIv737HQAhUHq48KHVKhC704ChAWCCswAg&url=http%3A%2F%2Flibrary.binus.ac.id%2FeColls%2FeThesisdoc%2FBab2DOC%2F2013-1-00535-SK%2520Bab2001.doc&usg=AFQjCNH2PmZxJTr_Nrco-O5op1sPHItENQ&sig2=KIRk3lG7C_cYAtDEww7ZpQ>

[12] Prasetyo, M. E. (n.d.). Teori Dasar Hidden Markov Model.

[13] Devi Handaya, H. F. (2016). Perbandingan pada Pengenalan Pembicara Bahasa Indonesia Menggunakan Vektor Kuantisasi (VK) dan Hidden Markov Model (HMM). 3-5.

Skripsi LAN HERAHMAN berjudul “analisa perbandingan kinerja sistem vervikasi tanda tangan online menggunakan Gaussian mixture model (GMM) dan support vector machine (SVM) , Fakultas teknik, universitas Indonesia , departemen tekni elektro, depok januari 2013

Skripsi adhimas yudha prawira berjudul analisis kinerja pemilihan ftur untuk support vector machine (svm) pada masalah analisis sentiment, fakultas matematika dan ilmu pengetahuan alam, program sarjana matematika depok, juni 2014s