



**UNIVERSITAS INDONESIA**

**Pengenalan Entitas Kesehatan pada Forum Online  
dengan Menggunakan Recurrent Neural Network**

**SKRIPSI**

**WAHID NUR ROHMAN**  
**1306381856**

**FAKULTAS ILMU KOMPUTER**  
**PROGRAM STUDI ILMU KOMPUTER**  
**DEPOK**  
**JULI 2016**



**UNIVERSITAS INDONESIA**

**Pengenalan Entitas Kesehatan pada Forum Online  
dengan Menggunakan Recurrent Neural Network**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Sarjana Ilmu Komputer**

**WAHID NUR ROHMAN**

**1306381856**

**FAKULTAS ILMU KOMPUTER  
PROGRAM STUDI ILMU KOMPUTER**

**DEPOK  
JULI 2016**

## **HALAMAN PERNYATAAN ORISINALITAS**

**Skripsi ini adalah hasil karya saya sendiri,  
dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar.**

**Nama : Wahid Nur Rohman**  
**NPM : 1306381856**  
**Tanda Tangan :**

**Tanggal : 22 Juli 2016**

## **HALAMAN PENGESAHAN**

Skripsi ini diajukan oleh :

Nama : Wahid Nur Rohman

NPM : 1306381856

Program Studi : Ilmu Komputer

Judul Skripsi : Pengenalan Entitas Kesehatan pada Forum Online  
dengan Menggunakan Recurrent Neural Network

**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Ilmu Komputer pada Program Studi Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Indonesia.**

## **DEWAN PENGUJI**

Pembimbing 1 : Dra. Mirna Adriani, Ph.D. ( )

Pembimbing 2 : Alfian Farizki Wicaksono S.T., M.Sc. ( )

Penguji : Dr. Indra Budi S.Kom., M.Kom ( )

Penguji : Ir. Ito Wasito M.Sc., Ph.D. ( )

Ditetapkan di : Depok

Tanggal : 27 Juni 2016

## KATA PENGANTAR

الْحَمْدُ لِلَّهِ الَّذِي هَدَانَا لِهَذَا وَمَا كُنَّا لِنَهْتَدِيَ لَوْلَا أَنْ هَدَانَا اللَّهُ

*Segala puji bagi Allah yang telah menunjuki kami kepada (surga) ini. Dan kami sekali-kali tidak akan mendapat petunjuk kalau Allah tidak memberi kami petunjuk. [Al-A'raf:43]*

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ، وَالصَّلَاةُ وَالسَّلَامُ عَلَى سَيِّدِنَا مُحَمَّدٍ سَيِّدِ الْأَوَّلِينَ وَالْآخِرِينَ، وَعَلَى آلِهِ وَصَحْبِهِ وَمَنْ اهْتَدَى يَهْدِيهِ إِلَى يَوْمِ الدِّينِ

*Segala puji bagi Allah, Tuhan sekalian alam, semoga keselamatan dan kesejahteraan tetap terlimpahkan atas junjungan kita Nabi Muhammad SAW, penghulu manusia, baik yang dahulu maupun yang belakangan, begitu juga kepada segenap keluarga dan semua orang yang mengikuti petunjuk, sampai saat Hari Kemudian. Segala puji dan syukur kehadiran Allah SWT, Tuhan Yang Maha Esa, yang senantiasa memberikan ramhat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini.*

Penulisan skripsi ini ditujukan untuk memenuhi salah satu syarat untuk menyelesaikan pendidikan pada Program Sarjana Ilmu Komputer, Universitas Indonesia. Penulis sadar bahwa dalam perjalanan menuntut ilmu di universitas hingga dalam menyelesaikan skripsi ini, penulis tidak sendiri. Penulis ingin berterima kasih kepada pihak-pihak yang selalu peduli, mendampingi, dan mendukung penulis, yaitu:

1. Kedua Orang Tua penulis yang selalu memberikan dukungan dan do'a kepada penulis.
2. Dra. Mirna Adriani, Ph.D. dan Dr. Amalia Zahra selaku dosen pembimbing yang banyak memberikan arahan, masukan, dan bantuan dalam menyelesaikan skripsi ini.
3. Alfian Farizki Wicaksono, ST., M.Sc. dan Rahmad Mahendra, S.Kom., M.Sc. yang memberi dukungan dari awal sampai akhir pengerjaan skripsi ini, dan juga memberikan tips-tips dalam mengerjakan skripsi.
4. Andreas Febrian yang telah membuat *template* dokumen skripsi ini, sehingga penulis menjadi terbantu dalam menulis skripsi.

5. Erik Dominikus yang telah mempublikasikan dan mempopulerkan *template* dokumen skripsi ini, sehingga penulis menjadi tahu bahwa ada *template* tersebut.
6. Mohammad Syahid Wildan dan Abid Nurul Hakim, sebagai rekan yang banyak memberi masukan dan berbagi ide dengan penulis.
7. Teman-teman Lab Information Retrieval yang memberi dukungan dan semangat kepada penulis untuk menyelesaikan skripsi ini.
8. Teman-teman Forum Remaja Masjid UI yang memberi dukungan serta do'a kepada penulis untuk menyelesaikan skripsi ini.
9. Pihak-pihak lain yang tidak dapat penulis sebutkan satu-persatu yang sudah memberikan bantuan dan dukungannya kepada penulis.

Depok, Juni 2016

Wahid Nur Rohman

## HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

**Nama** : Wahid Nur Rohman  
**NPM** : 1306381856  
**Program Studi** : Ilmu Komputer  
**Fakultas** : Ilmu Komputer  
**Jenis Karya** : Skripsi

demikian demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty Free Right)** atas karya ilmiah saya yang berjudul:

Pengenalan Entitas Kesehatan pada Forum Online dengan Menggunakan  
Recurrent Neural Network

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia-  
/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok  
Pada tanggal : 22 Juli 2016  
Yang menyatakan

(Wahid Nur Rohman)

## ABSTRAK

Nama : Wahid Nur Rohman  
Program Studi : Ilmu Komputer  
Judul : Pengenalan Entitas Kesehatan pada Forum Online dengan Menggunakan Recurrent Neural Network

Banyak umat Muslim yang ingin menghafalkan Al-Qur'an. Namun orang yang menghafalkan Al-Qur'an membutuhkan rekan untuk membantu mengevaluasi hafalannya. Untuk membantu proses tersebut, penelitian ini mengembangkan sebuah sistem yang mampu mengevaluasi pembacaan Al-Qur'an secara otomatis. Sistem tersebut menggunakan fitur *mel frequency cepstral coefficient* (MFCC) dan *shifted delta cepstral coefficient* (SDCC), dengan metode klasifikasi *support vector machine* (SVM) dan *Gaussian mixture model* (GMM). Eksperimen dalam penelitian ini dilakukan terhadap setiap ayat di juz 30 Al-Qur'an, dan hasilnya menunjukkan bahwa kombinasi yang paling tepat untuk digunakan dalam sistem tersebut adalah fitur SDCC dengan metode klasifikasi GMM.

Kata Kunci:

Al-Qur'an, evaluasi, MFCC, SDCC, SVM, GMM



## ABSTRACT

Name : Wahid Nur Rohman  
Program : Computer Science  
Title : Medical Entity Recognition on the Online Health Forum using  
Recurrent Neural Network

Many Moslems want to recite Al-Qur'an. Unfortunately, someone who is reciting Al-Qur'an needs a partner to help evaluating the recitation. To help that process, this research develops a system that is able to automatically evaluate Al-Qur'an recitation. The system uses *mel frequency cepstral coefficient* (MFCC) and *shifted delta cepstral coefficient* (SDCC) feature, with *support vector machine* (SVM) and *Gaussian mixture model* (GMM) classification method. The experiment is applied to every ayah in juz 30 of Al-Qur'an, and the result shows that the best combination to use in the system is SDCC feature with GMM classification method.

Keywords:

Al-Qur'an, evaluation, MFCC, SDCC, SVM, GMM

## DAFTAR ISI

<b>HALAMAN JUDUL</b>	<b>i</b>
<b>LEMBAR PERNYATAAN ORISINALITAS</b>	<b>ii</b>
<b>LEMBAR PENGESAHAN</b>	<b>iii</b>
<b>KATA PENGANTAR</b>	<b>iv</b>
<b>LEMBAR PERSETUJUAN PUBLIKASI ILMIAH</b>	<b>vi</b>
<b>ABSTRAK</b>	<b>vii</b>
<b>Daftar Isi</b>	<b>ix</b>
<b>Daftar Gambar</b>	<b>xii</b>
<b>Daftar Tabel</b>	<b>xiii</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Perumusan Masalah . . . . .	2
1.3 Tujuan dan Manfaat Penelitian . . . . .	3
1.4 Metodologi Penelitian . . . . .	3
1.5 Ruang Lingkup Penelitian . . . . .	5
1.6 Sistematika Penulisan . . . . .	5
<b>2 LANDASAN TEORI</b>	<b>7</b>
2.1 Pengenalan Entitas Kesehatan . . . . .	7
2.2 Recurrent Neural Networks . . . . .	7
2.2.1 Deep Learning . . . . .	8
2.2.2 Vanilla RNNs . . . . .	8
2.2.3 Long Short Term Memory . . . . .	8
2.2.4 Penerapan RNNs untuk MER . . . . .	8
2.3 Word Embedding . . . . .	8
2.4 Pengenalan Suara Otomatis . . . . .	9
2.5 Fitur . . . . .	10
2.5.1 Mel Frequency Cepstral Coefficients (MFCCs) . . . . .	11
2.5.2 Shifted Delta Cepstral Coefficients (SDCCs) . . . . .	11
2.6 Longest Common Subsequence (LCS) . . . . .	12
2.7 Metode <i>Clustering</i> dan Klasifikasi . . . . .	14
2.7.1 K-Means Clustering . . . . .	14
2.7.2 Support Vector Machine (SVM) . . . . .	15

2.7.3	Gaussian Mixture Model (GMM)	18
2.8	Evaluasi	19
2.9	Penelitian Terkait	20
<b>3</b>	<b>PERANCANGAN SISTEM</b>	<b>22</b>
3.1	Rancangan Arsitektur Sistem	22
3.1.1	Pengumpulan Data	22
3.1.2	Proses Normalisasi	23
3.1.3	Pembuatan Model <i>Word Embedding</i>	24
3.1.4	Ekstraksi Fitur	24
3.1.5	Eksperimen	25
3.2	Data	25
3.2.1	Pengambilan Data	25
3.2.2	Pembuangan Data Duplikat	26
3.2.3	Penyaringan Data	28
3.3	Perangkat dan Fungsi Pendukung	29
3.3.1	Perangkat Pendukung	29
3.3.2	Fungsi Pendukung	29
<b>4</b>	<b>EKSPERIMEN</b>	<b>32</b>
4.1	Normalisasi Data	32
4.2	Ekstraksi Fitur	33
4.2.1	Ekstraksi Fitur MFCC	34
4.2.2	Ekstraksi Fitur SDCC	36
4.3	Pemodelan	38
4.3.1	Pemodelan dengan Support Vector Machine (SVM)	39
4.3.2	Pemodelan dengan Gaussian Mixture Model (GMM)	40
4.3.3	Pemodelan dengan Gabungan SVM dan GMM	40
4.4	Pengujian	41
<b>5</b>	<b>HASIL EKSPERIMEN DAN ANALISIS</b>	<b>43</b>
5.1	Hasil dengan Fitur MFCC	43
5.1.1	Hasil dengan Fitur MFCC dan Metode Klasifikasi SVM	43
5.1.2	Hasil dengan Fitur MFCC dan Metode Klasifikasi GMM	45
5.1.3	Hasil dengan Fitur MFCC dan Metode Klasifikasi Gabungan	46
5.2	Hasil dengan Fitur SDCC	48
5.2.1	Hasil dengan Fitur SDCC dan Metode Klasifikasi SVM	48
5.2.2	Hasil dengan Fitur SDCC dan Metode Klasifikasi GMM	49
5.2.3	Hasil dengan Fitur SDCC dan Metode Klasifikasi Gabungan	51
5.3	Perbandingan Hasil	52
5.3.1	Perbandingan Metode Klasifikasi pada Fitur MFCC	52
5.3.2	Perbandingan Metode Klasifikasi pada Fitur SDCC	53
5.3.3	Perbandingan Fitur pada Metode Klasifikasi SVM	54
5.3.4	Perbandingan Fitur pada Metode Klasifikasi GMM	55
5.3.5	Perbandingan Fitur pada Metode Klasifikasi Gabungan	56
5.4	Analisis Lanjut	57

<b>6 KESIMPULAN DAN SARAN</b>	<b>60</b>
6.1 Kesimpulan . . . . .	60
6.2 Saran . . . . .	61
<b>Daftar Referensi</b>	<b>62</b>

## DAFTAR GAMBAR

2.1	<i>Simple Recurrent Neural Network</i> . . . . .	8
2.2	Contoh Perhitungan Sebuah Vektor SDCC . . . . .	12
2.3	Prosedur LCS-Length . . . . .	13
2.4	Tabel $c$ dan $b$ yang Dihasilkan oleh Prosedur LCS-Length . . . . .	14
2.5	Ilustrasi Data dalam Dimensi 2 . . . . .	16
2.6	Ilustrasi Data dalam Dimensi 3 . . . . .	17
2.7	<i>Gaussian Mixture Distribution</i> dengan 7 Komponen . . . . .	19
3.1	Rancangan Arsitektur Sistem . . . . .	22
4.1	Distribusi <i>Gaussian</i> Terhadap Nilai Jarak <i>Mahalanobis</i> . . . . .	33
4.2	Alur Ekstraksi Fitur MFCC . . . . .	36
4.3	Alur Ekstraksi Fitur SDCC . . . . .	38
5.1	Histogram Metrik Evaluasi dengan Fitur MFCC dan Metode Klasifikasi SVM . . . . .	44
5.2	Histogram Metrik Evaluasi dengan Fitur MFCC dan Metode Klasifikasi GMM . . . . .	46
5.3	Histogram Metrik Evaluasi dengan Fitur MFCC dan Metode Klasifikasi Gabungan . . . . .	47
5.4	Histogram Metrik Evaluasi dengan Fitur SDCC dan Metode Klasifikasi SVM . . . . .	49
5.5	Histogram Metrik Evaluasi dengan Fitur SDCC dan Metode Klasifikasi GMM . . . . .	50
5.6	Histogram Metrik Evaluasi dengan Fitur SDCC dan Metode Klasifikasi Gabungan . . . . .	52
5.7	Perbandingan Metode Klasifikasi untuk Fitur MFCC . . . . .	53
5.8	Perbandingan Metode Klasifikasi untuk Fitur SDCC . . . . .	54
5.9	Perbandingan Fitur untuk Metode Klasifikasi SVM . . . . .	55
5.10	Perbandingan Fitur untuk Metode Klasifikasi GMM . . . . .	56
5.11	Perbandingan Fitur untuk Metode Klasifikasi Gabungan . . . . .	57

## DAFTAR TABEL

2.1	Kontingensi . . . . .	19
3.1	Informasi Detil Surat An-Naba' Ayat 7 dengan Qari Mishary Rashid Alafasy . . . . .	26
3.2	Informasi Data Eksperimen . . . . .	29
3.3	Spesifikasi <i>Hardware</i> . . . . .	29
3.4	Parameter Fungsi <i>mfcc</i> . . . . .	30
3.5	Parameter Fungsi <i>mfcc2sdc</i> . . . . .	31
4.1	Parameter Pemanggilan Fungsi <i>mfcc</i> . . . . .	34
4.2	Parameter Pemanggilan Fungsi <i>mfcc2sdc</i> . . . . .	37
4.3	Parameter SVM dalam Eksperimen . . . . .	39
4.4	Parameter GMM dalam Eksperimen . . . . .	40
5.1	Rangkuman Hasil Eksperimen dengan Fitur MFCC dan Metode Klasifikasi SVM . . . . .	43
5.2	Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur MFCC dan Metode Klasifikasi SVM . . . . .	44
5.3	Rangkuman Hasil Eksperimen dengan Fitur MFCC dan Metode Klasifikasi GMM . . . . .	45
5.4	Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur MFCC dan Metode Klasifikasi GMM . . . . .	45
5.5	Rangkuman Hasil Eksperimen dengan Fitur MFCC dan Metode Klasifikasi Gabungan . . . . .	46
5.6	Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur MFCC dan Metode Klasifikasi Gabungan . . . . .	47
5.7	Rangkuman Hasil Eksperimen dengan Fitur SDCC dan Metode Klasifikasi SVM . . . . .	48
5.8	Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur SDCC dan Metode Klasifikasi SVM . . . . .	48
5.9	Rangkuman Hasil Eksperimen dengan Fitur SDCC dan Metode Klasifikasi GMM . . . . .	49
5.10	Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur SDCC dan Metode Klasifikasi GMM . . . . .	50
5.11	Rangkuman Hasil Eksperimen dengan Fitur SDCC dan Metode Klasifikasi Gabungan . . . . .	51
5.12	Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur SDCC dan Metode Klasifikasi Gabungan . . . . .	51
5.13	Ayat-Ayat yang Diurutkan dari Akurasi Tertinggi . . . . .	58
5.14	Ayat-Ayat yang Diurutkan dari Akurasi Terendah . . . . .	58

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Saat ini berbagai kegiatan manusia semakin dimudahkan dengan adanya teknologi. Salah satu kegiatan tersebut yaitu melakukan konsultasi terkait masalah kesehatan kepada dokter. Teknologi yang dapat membantu seseorang dalam hal ini yaitu forum kesehatan *online*. Melalui forum tersebut, seseorang hanya perlu menuliskan keluhan dan pertanyaan di formulir yang tersedia. Kemudian, dokter yang memiliki akun di forum kesehatan *online* tersebut dapat memberikan jawaban atas pertanyaan orang tersebut.

Dari forum kesehatan *online*, banyak informasi yang dapat diambil. Informasi tersebut misalnya informasi keluhan yang dialami pasien, obat yang sebaiknya digunakan atau langkah penyembuhan yang dapat dilakukan. Orang lain dapat mencari obat atau langkah penyembuhan dari forum tersebut melalui pertanyaan yang sudah ada. Yang menjadi permasalahan adalah *post* yang ditulis oleh pelapor kurang struktur. Dokumen *post* tidak dibagi menjadi beberapa bagian seperti bagian keluhan, penyakit, obat dll, namun hanya menjadi 1 bagian saja. Misalnya seseorang menanyakan tentang keluhannya, orang tersebut hanya diberikan 2 buah isian berupa judul dan isi pertanyaan. Jawaban yang diberikan oleh dokter juga sama, hanya menjadi 1 bagian saja. Jawaban yang diberikan tidak terstruktur seperti memiliki bagian langkah penyembuhan, nama penyakit dan obat secara terpisah. Hal ini menyebabkan orang sulit melakukan ekstraksi informasi dari dokumen tersebut.

Dari permasalahan tersebut, terdapat sebuah solusi untuk melakukan ekstraksi informasi penyakit dalam suatu dokumen. Solusi tersebut yaitu dengan menggunakan sistem Pengenalan Entitas Kesehatan. Sistem ini dapat mengenali entitas kesejatan dalam sebuah dokumen. Diberikan sebuah dokumen, sistem ini akan mengembalikan dokumen dengan entitas kesehatan yang sudah diberi label di dalamnya. Sistem ini udah banyak dikembangkan oleh beberapa peneliti. Salah satu penelitian yang mengembangkan sistem Pengenalan Entitas Kesehatan dilakukan oleh Abacha dan Zweigenbaum (2011). Pebelitian tersebut menggunakan dokumen medis rumah sakit berbahasa Inggris.

Saat ini, pengembangan sistem Pengenalan Entitas Kesehatan pada dokumen

berbahasa Indonesia masih belum banyak. Ada beberapa penelitian terkait sistem Pengenalan Entitas Kesehatan, namun hasil yang diberikan belum memuaskan. Salah satu penelitian terkait Pengenalan Entitas Kesehatan dilakukan oleh Herwando (2016). Pengenalan Entitas Kesehatan yang dikembangkan menggunakan dokumen forum kesehatan *online* berbahasa Indonesia dari beberapa situs. Herwando (2016) menggunakan algoritma *Conditional Random Field*. Penelitian yang dilakukan bertujuan untuk mencari kombinasi fitur yang dapat menghasilkan akurasi terbaik. Hasil akhir yang didapatkan yaitu *precision* 70.97%, *recall* 57.83% dan *f-measure* 63.69%. Fitur-fitur yang digunakan yaitu fitur kata itu sendiri, frasa, kamus: *symptom*, *disease*, *treatment*, *drug*, kata pertama sebelumnya dan panjang kata.

Dalam penelitian ini, penulis mengusulkan *framework* lain untuk mengembangkan sistem Pengenalan Entitas Kesehatan. Penulis mengusulkan sebuah sistem Pengenalan Entitas Kesehatan dengan menggunakan *Recurrent Neural Network*. Berdasarkan penelitian terkait Pengenalan Entitas Kesehatan dengan menggunakan *Recurrent Neural Network* pada dokumen Bahasa Inggris yang dilakukan oleh **SIAPA**, hasil yang didapatkan sangat baik, yaitu *precision* 70.97%, *recall* 57.83% dan *f-measure* 63.69%. Oleh karena itu, penulis menerapkan *Recurrent Neural Network* pada penelitian ini.

Penulis berharap bahwa penelitian ini akan memberikan banyak manfaat. Model Pengenalan Entitas Kesehatan yang dihasilkan dapat digunakan untuk membuat aplikasi lain. Misalnya dengan adanya Pengenalan Entitas Kesehatan pada dokumen bahasa Indonesia, dapat dibuat sistem untuk melakukan *indexing* dokumen forum sehingga pencarian dokumen kesehatan dapat dilakukan dengan lebih efisien. Selain itu, keluaran dari Pengenalan Entitas Kesehatan juga dapat digunakan untuk mengidentifikasi tren penyakit pada waktu tertentu dari suatu sumber, sehingga pihak terkait mampu melakukan langkah dan kebijakan yang tepat. Penulis berharap bahwa penelitian Pengenalan Entitas Kesehatan pada dokumen berbahasa Indonesia ini dapat dilanjutkan sehingga dapat menghasilkan model yang lebih baik dan membuat suatu aplikasi yang memanfaatkan keluaran dari penelitian ini. Masih banyak manfaat lain yang didapatkan dengan adanya sistem Pengenalan Entitas Kesehatan yang memiliki hasil akurat.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, dalam penelitian ini penulis mengajukan rumusan masalah sebagai berikut:



1. Bagaimana fitur yang membuat sistem Pengenalan Entitas Kesehatan memiliki performa terbaik?
2. Bagaimana kendala dalam melakukan penelitian ini?

### 1.3 Tujuan dan Manfaat Penelitian

Penelitian ini bertujuan untuk membangun sistem yang mampu melakukan ekstraksi entitas kesehatan dari forum *online*. Sebenarnya, pada penelitian yang dilakukan oleh Herwando (2016) sudah menghasilkan sebuah sistem yang sama. Namun, fokus penelitian ini yaitu mencoba menggunakan metode yang berbeda. Metode tersebut yaitu dengan menggunakan *Recurrent Neural Network* dengan harapan mampu memberikan hasil yang lebih baik. Penelitian ini juga bertujuan untuk mendapatkan fitur-fitur yang membuat sistem memiliki performa terbaik. Selain itu, penelitian ini juga bertujuan untuk mendapatkan informasi baru terkait pembuatan sistem Pengenalan Entitas Kesehatan berbahasa Indonesia.

Manfaat dari penelitian ini adalah menghasilkan rancangan sistem dan metode yang dapat digunakan sebagai bahan penelitian lanjutan. Saat ini, sistem dan metode yang dihasilkan hanya mampu mengenali entitas kesehatan saja. Selain itu, model yang dihasilkan dapat digunakan untuk melakukan pengindeksan dokumen kesehatan sehingga pencarian dokumen dapat dilakukan dengan lebih efisien. Hal ini dapat digunakan untuk membuat sistem informasi tentang suatu jenis penyakit lengkap dengan gejala, obat dan cara penyembuhannya. Selama ini, masyarakat yang menanyakan suatu penyakit melalui forum *online* tidak membaca terlebih dahulu riwayat pertanyaan yang telah ditanyakan oleh orang lain. Oleh karena itu, diharapkan dengan sistem informasi tersebut, penanya hanya perlu mencari penyakit yang akan ditanyakan pada sistem informasi tersebut. Apabila tidak ada, penanya dapat mengajukan pertanyaan, kemudian pertanyaan dan jawaban yang diberikan akan terindeks oleh sistem dan menambah informasi.

Selain itu, hasil penelitian ini juga dapat digunakan untuk membangun sistem yang mengenali tren penyakit pada masyarakat, sehingga pihak terkait mampu menentukan langkah strategis yang tepat.

### 1.4 Metodologi Penelitian

Berikut merupakan metode penelitian yang penulis lakukan.

1. Studi Literatur

Pada tahapan ini penulis mencari literatur yang terkait dengan penelitian ini.

Literatur ini digunakan sebagai bahan pembelajaran dan untuk mendukung penelitian yang penulis lakukan. Literatur yang penulis gunakan memiliki keterkaitan terhadap kasus Pengenalan Entitas Kesehatan , *Sequence Labelling* dan *Recurrent Neural Network*.

## 2. Perumusan Masalah

Pada tahapan ini penulis membuat rumusan masalah yang akan dijawab melalui serangkaian penelitian ini. penulis membuat rumusan masalah setelah melakukan studi literatur dan mempelajari penelitian sebelumnya yang telah dilakukan oleh []

## 3. Perancangan Sistem

Pada tahapan ini, penulis membuat rancangan sistem yang akan penulis gunakan dalam penelitian ini. Tujuan dibuatnya perancangan sistem ini yaitu untuk memudahkan penulis dalam melakukan penelitian dari awal sampai akhir.

## 4. Pengumpulan Data

Pada tahapan ini, penulis mengumpulkan data percobaan yang diperlukan. Data tersebut penulis dapatkan dari penelitian sebelumnya ditambah dengan data dari forum *online* yang lebih baru.

## 5. Pengolahan Data

Pada tahapan ini, penulis melakukan pengolahan data supaya data dapat dengan mudah dibaca oleh sistem. Selain itu, pengolahan data dilakukan juga untuk membuat korpus.

## 6. Eksperimen

Tahapan ini merupakan bagian inti dari penelitian. penulis melakukan langkah eksperimen dengan tujuan mendapatkan jawaban dari pertanyaan yang telah dirumuskan pada rumusan masalah.

## 7. Evaluasi dan Analisis Hasil

Pada tahapan ini penulis melakukan evaluasi dan analisis dari hasil eksperimen.

## 8. Penarikan Kesimpulan

Tahap ini merupakan tahap terakhir dari penelitian. Setelah melakukan serangkaian eksperimen, evaluasi dan analisis, penulis memberikan kesimpulan dan informasi penting terkait penelitian ini. Selain itu penulis juga memberikan saran untuk penelitian selanjutnya.

## 1.5 Ruang Lingkup Penelitian

Pada penelitian ini terdapat beberapa batasan yang penulis tentukan, yaitu:

### 1. Kriteria Data

Data yang digunakan dalam penelitian ini merupakan data penelitian Herwando (2016) dan data dari forum kesehatan *online* berbahasa Indonesia di internet.

### 2. Penerapan Metode

Penelitian ini berfokus pada kombinasi pemilihan fitur dan metode yang memberikan hasil terbaik.

### 3. Aplikasi

Penelitian ini hanya dilakukan sampai tahap metode, tidak sampai ke tahap pembuatan aplikasi. Karena penelitian ini hanya memberikan output pelabelan pada dokumen teks kesehatan, belum sampai pada pembuatan sistem informasi kesehatan.

### 4. Entitas Kesehatan

Pengenalan entitas kesehatan pada penelitian ini berfokus pada pengenalan nama penyakit (*disease*), gejala-gejala penyakit (*online*), nama obat (*drug*) dan langkah pengobatan (*treatment*),

### 5. Domain Pengenalan

Pengenalan entitas kesehatan dilakukan pada bagian judul pertanyaan, isi pertanyaan/keluhan dan isi jawaban dari dokter.

## 1.6 Sistematika Penulisan

Sistematika penulisan dalam laporan penelitian ini sebagai berikut:

- Bab 1 PENDAHULUAN

Pada bab ini penulis menjelaskan latar belakang, perumusan masalah, tujuan dan manfaat penelitian, metodologi penelitian, ruang lingkup penelitian dan sistematika penulisan.

- Bab 2 LANDASAN TEORI

Pada bab ini penulis menjelaskan teori-teori yang terkait dengan penelitian ini. Teori-teori tersebut penulis dapatkan dari studi literatur dari berbagai sumber.

- **Bab 3 PERANCANGAN SISTEM**

Pada bab ini penulis menjelaskan rancangan arsitektur sistem yang akan dikembangkan dalam penelitian ini.

- **Bab 4 EKSPERIMEN**

Pada bab ini penulis menjelaskan proses implementasi sistem dan eksperimen berdasarkan rancangan yang telah Wahid Nur Rohman tentukan pada bab sebelumnya.

- **Bab 5 HASIL EKSPERIMEN DAN ANALISIS**

Pada bab ini penulis menjelaskan analisis dari hasil eksperimen yang telah penulis kerjakan pada tahap sebelumnya.

- **Bab 6 KESIMPULAN DAN SARAN**

Pada bab ini penulis memberikan kesimpulan berdasarkan hasil eksperimen dan analisis yang telah dilakukan pada penelitian ini. Selain itu penulis juga memberikan saran dan masukan untuk penelitian dan pengembangan sistem mengenai Pengenalan Entitas Kesehatan berbahasa Indonesia selanjutnya.

## BAB 2

### LANDASAN TEORI

#### 2.1 Pengenalan Entitas Kesehatan

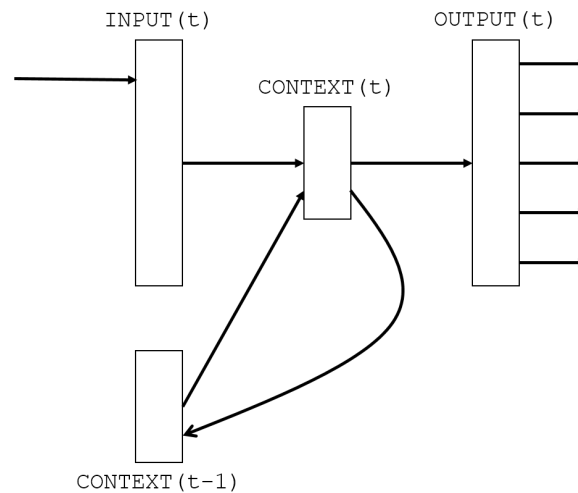
#### 2.2 Recurrent Neural Networks

*Recurrent neural networks* (RNNs) merupakan salah satu arsitektur dalam *deep learning*. RNNs memiliki *neuron* yang terkoneksi dengan *neuron* lain sehingga membentuk *loop* umpan balik (Haykin et al. (2009)), tidak seperti *feedforward neural network* (FNNs) dimana aliran informasi hanya berjalan searah. RNNs memungkinkan *output* yang dihasilkan akan menjadi *input* untuk menghasilkan *output* yang lain. Hal ini menyebabkan perilaku RNNs tidak hanya bergantung pada *input* saat ini saja, namun juga bergantung pada *output* sebelumnya. Oleh karena itu, RNNs memiliki kemampuan yang sangat bagus sebagai model dalam permasalahan *sequence data* dibandingkan dengan FNNs. RNNs sendiri memiliki kemampuan yang sangat bagus dalam beberapa *task*, seperti *language model* (Mikolov et al. (2010)) dan *speech recognition* (Graves et al. (2013)).

Dibandingkan dengan FNNs, RNNs memiliki beberapa kelebihan (Mikolov et al. (2010)), yaitu:

1. Pada RNNs, kata-kata sebelumnya direpresentasikan dengan *recurrent connections*, sehingga RNNs dapat menyimpan informasi kata sebelumnya dalam jumlah tak hingga. Pada FNNs, representasi kata sebelumnya berupa konteks dari  $n-1$  kata. Oleh karena itu, FNNs terbatas dalam penyimpanan informasi kata sebelumnya terbatas seperti pada model  $n$ -gram.
2. RNNs dapat melakukan kompresi keseluruhan riwayat kata menjadi ruang dimensi yang lebih kecil, sedangkan FNNs melakukan kompresi/proyeksi hanya dengan sebuah kata saja.
3. RNNs memiliki kemampuan membentuk *short term memory*, sehingga dapat dapat posisi invarian sebuah kata dapat ditangani. Hal ini tidak dapat dilakukan pada FNNs,

Berikut merupakan gambar struktur RNNs yang diusulkan oleh Elman (1990).



**Gambar 2.1:** *Simple Recurrent Neural Network*

Sumber gambar: Mikolov et al. (2010)

Sebuah jaringan pada RNNs memiliki *input layer*  $x$ , *hidden layer*  $s$  dan *output layer*  $y$ . Untuk suatu waktu  $t$ , *input* RNNs dinotasikan sebagai  $x(t)$ , *state* dinotasikan sebagai  $s(t)$  dan *output* dinotasikan sebagai  $y(t)$ .

Perhitungan matematis pada RNNs

RNNs have demonstrated great success in sequence labeling and prediction tasks such as handwriting recognition and language modeling. In acoustic modeling for speech recognition, however, where deep neural networks (DNNs) are the established state-of-the-art, recently RNNs have received little attention beyond small scale phone recognition tasks, notable exceptions being the work of Robinson [1], Graves [2], and Sak [3].

### 2.2.1 Deep Learning

### 2.2.2 Vanilla RNNs

### 2.2.3 Long Short Term Memory

### 2.2.4 Penerapan RNNs untuk MER

## 2.3 Word Embedding

Satu Al-Qur'an dibagi sama rata ke dalam 30 bagian, yang masing-masing disebut dengan *juz*. Dalam Al-Qur'an terdapat 114 *surat*, di mana setiap surat terdiri dari beberapa *ayat*, bervariasi mulai dari 3 ayat sampai 286 ayat. Total seluruh ayat dalam Al-Qur'an ada sebanyak 6,236 ayat. Ayat-ayat Al-Qur'an tidak seluruhnya

unik, karena ada beberapa pasang ayat yang sama dan ada beberapa pasang ayat yang mirip. Hal ini bisa dideteksi menggunakan algoritma yang akan dijelaskan di Bab 2.6.

## 2.4 Pengenalan Suara Otomatis

Pengenalan Suara Otomatis atau yang lebih umum dikenal dengan *Automatic Speech Recognition* (ASR) memiliki beberapa definisi dari sumber yang berbeda. Menurut Forsberg (2003), ASR adalah proses yang dilakukan komputer untuk menafsirkan ucapan manusia. Definisi lebih teknis diberikan oleh Jurafsky dan Martin (2009), yaitu ASR adalah sebuah sistem untuk memetakan sinyal akustik ke dalam untaian kata. Selain definisi-definisi yang telah disebutkan sebelumnya, Chigier (1997) menambahkan bahwa ASR merupakan proses komputasi yang bertujuan untuk menangkap sinyal akustik yang merepresentasikan ucapan dan menentukan kata-kata yang diucapkan menggunakan pencocokan pola.

Teknik dalam membangun sistem ASR berbeda-beda, namun secara garis besar ada kesamaan. Sistem pengenalan suara umumnya memiliki sekumpulan model akustik dan bahasa yang direpresentasikan sebagai pola, dan disimpan dalam sebuah basis data komputer. Model-model ini kemudian dibandingkan dengan sinyal yang ditangkap. Untuk sistem ASR yang mengandung sedikit kosakata (kurang dari 50 kata), sebuah model dapat merepresentasikan sebuah kata. Namun untuk kosakata yang lebih dari 50 kata, proses pemodelan dan algoritma pengenalan membutuhkan komputasi yang banyak dan tidak praktis. Oleh karena itu, sistem dengan kosakata yang banyak (lebih dari 1000 kata) membuat sebuah model yang merepresentasikan komponen yang lebih kecil dari kata, sebagai contoh adalah *fonem*. Deretan fonem dapat dirangkai untuk menghasilkan sebuah model kata.

Juang dan Rabiner (2005) dalam artikelnya yang berjudul *Automatic Speech Recognition – A Brief History of the Technology Development* menjelaskan bahwa saat ini teknologi ASR sudah tersedia secara komersial, walaupun hanya mampu menjalankan tugas-tugas yang terbatas. Teknologi ini memungkinkan mesin untuk menanggapi perintah manusia melalui suara dengan baik. Namun teknologi tersebut masih jauh dari kemampuan untuk bercakap-cakap dengan manusia dalam berbagai topik bebas. Perkembangan dalam bidang sains dan teknologi semakin mendorong sistem ASR untuk menjadi sistem yang mampu mengenali dan memahami ucapan manusia dengan baik.

Sejarah singkat mengenai perkembangan ASR dapat dikelompokkan ke dalam beberapa era. Dimulai pada sekitar tahun 1960, di mana pada saat itu telah

berhasil dikembangkan teknologi yang mampu mengenali kosakata dalam skala kecil (10 sampai 100 kata). Pengenalan ini dilakukan dengan mengenali kata demi kata secara terpisah berdasarkan sifat-sifat *acoustic-phonetic* yang terdapat dalam suara. *Acoustic-phonetic* adalah ilmu tentang karakteristik akustik dari suara. Kunci dari keberhasilan teknologi ASR di era ini adalah analisis *filter-bank*, metode normalisasi *simple time*, dan permulaan dari metodologi *dynamic programming* yang canggih. *Filter-bank* adalah pembagian sinyal *input*  $x(n)$  ke dalam himpunan sinyal analisis  $x_1(n), x_2(n), \dots$ , yang masing-masing berkorespondensi terhadap wilayahnya di dalam spektrum  $x(n)$ .

Sekitar tahun 1970, teknologi yang berkembang mampu untuk mengenali kosakata dalam skala sedang (100 sampai 1000 kata). Kunci perkembangan teknologi di era ini adalah model pengenalan pola (*pattern recognition*), permulaan metode *Linear Predictive Coding* (LPC) untuk merepresentasikan spektrum, metode *pattern clustering* untuk *speaker-independent recognizers*, dan permulaan metode *dynamic programming* untuk menyelesaikan permasalahan *connected word recognition*. Sekitar tahun 1980, teknologi ASR mampu menangani kosakata dalam skala besar (1000 kata atau lebih). Kunci dari keberhasilan di era ini adalah model statistik *Hidden Markov Model* (HMM) dan model bahasa *stochastic*.

Teknologi semakin maju pada sekitar tahun 1990. Sistem ASR dengan kosakata berskala besar yang lebih canggih dari era sebelumnya telah berhasil dikembangkan pada era ini, dengan adanya *continuous speech recognition and understanding*. Adapun keberhasilan teknologi ASR di era ini berkat metode untuk melakukan *stochastic language understanding*, pembelajaran statistik dari model akustik dan model bahasa, permulaan *finite state transducer framework* (dan juga *FSM Library*), serta metode untuk determinasi dan minimalisasi untuk implementasi sistem *speech understanding* yang efisien dengan kosakata berskala besar. Akhirnya sekitar tahun 2000, muncul permulaan dari sistem ASR dengan kosakata berskala sangat besar disertai model semantik penuh, terintegrasi dengan sistem sintesis *text-to-speech* (TTS), dan *multi-modal inputs*.

## 2.5 Fitur

Informasi tertentu yang diambil dari data disebut *fitur*. Proses pengambilan fitur dari data dinamakan proses *ekstraksi fitur*. Dalam sistem ASR, fitur yang populer digunakan antara lain adalah *Mel Frequency Cepstral Coefficients* (MFCCs) dan *Shifted Delta Cepstral Coefficients* (SDCCs). Implementasi fungsi untuk melakukan ekstraksi fitur MFCCs dan SDCCs akan dijelaskan pada Bab 3.3.2.



### 2.5.1 Mel Frequency Cepstral Coefficients (MFCCs)

*Mel frequency cepstral coefficients* (MFCCs) adalah salah satu fitur modern dalam teknologi *speech*, yang diperkenalkan oleh Davis dan Mermelstein (1980). Fitur ini sudah banyak digunakan dalam ASR (Young et al., 2002), verifikasi *speaker* (Ganchev et al., 2005), dan juga di dalam identifikasi bahasa lisan (Yin et al., 2006). MFCCs adalah fitur yang berbasis spektrum *short-term*. Dalam penelitian tentang *speech*, *short term* sering dipilih sebagai fitur dengan asumsi bahwa secara statistik sinyal bersifat stasioner dalam periode waktu singkat. Namun jika terlalu singkat, banyaknya sampel tidak akan cukup untuk menghasilkan estimasi spektrum tepat. Umumnya interval waktu untuk sebuah *frame* adalah 20 ms sampai dengan 40 ms (Zahra dan Carson-Berndsen, 2013).

Lyons (2016) menjelaskan secara garis besar cara menghitung nilai MFCCs dari sebuah data audio. Langkah-langkahnya adalah sebagai berikut.

1. Bagi sebuah sinyal ke dalam beberapa *frame* singkat.
2. Hitung estimasi *periodogram* dari *power spectrum* pada setiap *frame*.
3. Terapkan *mel filter-bank* ke setiap *power spectrum*, lalu jumlahkan energi di setiap *filter*.
4. Hitung nilai logaritma dari seluruh energi *filter-bank*.
5. Hitung nilai *discrete cosine transform* (DCT) dari logaritma energi *filter-bank*.

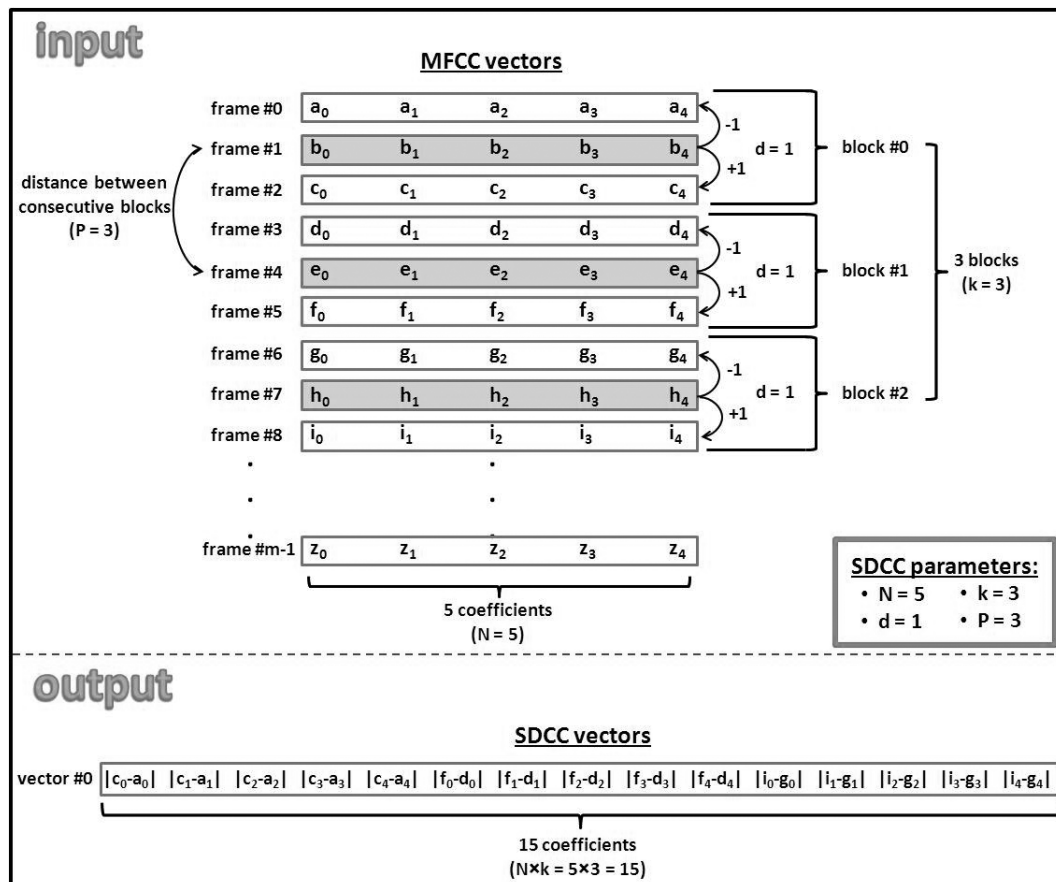
### 2.5.2 Shifted Delta Cepstral Coefficients (SDCCs)

*Shifted delta cepstral coefficients* (SDCCs) adalah pengembangan dari MFCCs (Torres-Carrasquillo et al., 2002). Untuk menghitung SDCCs diperlukan nilai MFCCs sebagai salah satu parameter dalam komputasi SDCCs. Vektor fitur SDCCs diperoleh dengan cara menumpuk *delta cepstral* yang dihitung dari beberapa *frame*. Selain nilai MFCCs, fitur SDCCs membutuhkan 4 parameter tambahan sebagai berikut.

1. *N*: banyaknya koefisien-koefisien *cepstral* yang dihitung pada setiap *frame*
2. *d*: waktu untuk *advance* dan *delay* untuk komputasi *delta*
3. *P*: pergeseran waktu antarblok berurutan

4.  $k$ : banyaknya blok di mana koefisien *delta* disambungkan untuk membentuk vektor fitur akhir

Menurut Zahra dan Carson-Berndsen (2013), ide dari perhitungan ini adalah menghitung jarak antara pasangan  $N$  koefisien *cepstral* paralel yang dipisahkan oleh waktu *advance* dan *delay*  $t$ , dilakukan pada  $k$  blok dengan waktu geser  $P$  antarblok yang berurutan. Semua nilai jarak ini kemudian disatukan menjadi sebuah vektor. Langkah ini dilakukan secara iteratif dari awal sampai akhir *frame* pada *speech*, sehingga diperoleh kumpulan vektor yang menjadi nilai SDCCs. Gambar 2.2 mengilustrasikan bagaimana cara menghitung sebuah vektor SDCC.



**Gambar 2.2:** Contoh Perhitungan Sebuah Vektor SDCC

Sumber gambar: Zahra dan Carson-Berndsen (2013)

## 2.6 Longest Common Subsequence (LCS)

Cormen et al. (2009) menjelaskan tentang algoritma untuk menyelesaikan permasalahan *longest common subsequence* (LCS) dalam bukunya yang berjudul *Introduction to Algorithms*. Diberikan sebuah barisan (*sequence*)  $X = \langle x_1, x_2, \dots, x_m \rangle$ . Barisan  $Z = \langle z_1, z_2, \dots, z_k \rangle$  disebut *subsequence* dari  $X$  jika terdapat barisan menaik

$\langle i_1, i_2, \dots, i_k \rangle$  berupa indeks dari  $X$  sedemikian hingga untuk semua  $j = 1, 2, \dots, k$ , terpenuhi  $x_{i_j} = z_j$ . Sebagai contoh,  $Z = \langle B, C, D, B \rangle$  adalah *subsequence* dari  $X = \langle A, B, C, B, D, A, B \rangle$  dengan barisan indeks yang berkorespondensi adalah  $Z = \langle 2, 3, 5, 7 \rangle$ .

Diberikan dua buah barisan  $X$  dan  $Y$ .  $Z$  disebut **common subsequence** dari  $X$  dan  $Y$  jika  $Z$  merupakan *subsequence* dari  $X$  dan  $Y$  sekaligus. Sebagai contoh, jika  $X = \langle A, B, C, B, D, A, B \rangle$ , dan  $Y = \langle B, D, C, A, B, A \rangle$ , barisan  $\langle B, C, A \rangle$  adalah *common subsequence* dari  $X$  dan  $Y$  sekaligus. Barisan  $\langle B, C, A \rangle$  bukan merupakan *common subsequence* terpanjang dari  $X$  dan  $Y$  karena hanya memiliki panjang 3, sedangkan terdapat barisan  $\langle B, C, B, A \rangle$  yang memiliki panjang 4 dan juga merupakan *common subsequence* dari  $X$  dan  $Y$ . Karena tidak ada *common subsequence* dari  $X$  dan  $Y$  dengan panjang 5, maka barisan  $\langle B, C, B, A \rangle$  merupakan *common subsequence* terpanjang atau **longest common subsequence** (LCS) dari  $X$  dan  $Y$ .

LCS dapat diselesaikan menggunakan *dynamic programming*. Prosedur pada gambar 2.3 menjelaskan cara untuk menghitung nilai LCS. Prosedur tersebut membutuhkan dua buah barisan sebagai *input*, yaitu  $X = \langle x_1, x_2, \dots, x_m \rangle$  dan  $Y = \langle y_1, y_2, \dots, y_n \rangle$ , lalu memberikan tabel  $b$  dan  $c$  sebagai *output*. Nilai LCS terdapat pada  $c[m, n]$ .

```

LCS-LENGTH( $X, Y$ )
1   $m = X.length$ 
2   $n = Y.length$ 
3  let  $b[1..m, 1..n]$  and  $c[0..m, 0..n]$  be new tables
4  for  $i = 1$  to  $m$ 
5       $c[i, 0] = 0$ 
6  for  $j = 0$  to  $n$ 
7       $c[0, j] = 0$ 
8  for  $i = 1$  to  $m$ 
9      for  $j = 1$  to  $n$ 
10         if  $x_i == y_j$ 
11              $c[i, j] = c[i - 1, j - 1] + 1$ 
12              $b[i, j] = "\diagup"$ 
13         elseif  $c[i - 1, j] \geq c[i, j - 1]$ 
14              $c[i, j] = c[i - 1, j]$ 
15              $b[i, j] = "\uparrow"$ 
16         else  $c[i, j] = c[i, j - 1]$ 
17              $b[i, j] = "\leftarrow"$ 
18  return  $c$  and  $b$ 

```

**Gambar 2.3:** Prosedur LCS-Length

Sumber gambar: Cormen et al. (2009)

Gambar 2.4 memperlihatkan bagaimana isi tabel yang dihasilkan dari prosedur

LCS-Length pada barisan  $X = \langle A, B, C, B, D, A, B \rangle$  dan  $Y = \langle B, D, C, A, B, A \rangle$ .

		$j$	0	1	2	3	4	5	6
		$y_j$		B	D	C	A	B	A
$i$	$x_i$								
0			0	0	0	0	0	0	0
1	A		0	0	0	0	1	1	1
2	B		0	1	1	1	1	2	2
3	C		0	1	1	2	2	2	2
4	B		0	1	1	2	2	3	3
5	D		0	1	2	2	2	3	3
6	A		0	1	2	2	3	3	4
7	B		0	1	2	2	3	4	4

**Gambar 2.4:** Tabel  $c$  dan  $b$  yang Dihasilkan oleh Prosedur LCS-Length

Sumber gambar: Cormen et al. (2009)

## 2.7 Metode *Clustering* dan Klasifikasi

Metode *clustering* dan klasifikasi adalah dua hal yang mirip. *Clustering* mengelompokkan data tanpa diketahui labelnya, sedangkan klasifikasi mengelompokkan data dengan diketahui labelnya.

### 2.7.1 K-Means Clustering

*Clustering* adalah proses mengelompokkan data menurut kemiripannya. Ada beberapa metode *clustering* yang dikembangkan oleh peneliti, salah satunya adalah *k-means clustering*. Metode ini mengelompokkan data menurut kedekatannya ke dalam  $k$  kelompok. Data dalam *k-means clustering* berupa sebuah vektor, sehingga kedekatan dua buah vektor ditentukan oleh jarak dari dua buah vektor tersebut. Jarak dua buah vektor salah satunya dihitung menggunakan *euclidean distance*. Anton dan Rorres (2010) menjelaskan tentang *euclidean distance* dalam bukunya yang berjudul *Elementary Linear Algebra* sebagai berikut. Jika  $\mathbf{u} = (u_1, u_2, \dots, u_n)$  dan  $\mathbf{v} = (v_1, v_2, \dots, v_n)$  adalah dua buah titik di dalam  $\mathbb{R}^n$ , maka *euclidean distance*

antara  $\mathbf{u}$  dan  $\mathbf{v}$  didefinisikan sebagai

$$d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\| = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \cdots + (u_n - v_n)^2}. \quad (2.1)$$

Kanungo et al. (2002) menjelaskan cara kerja *k-means* sebagai berikut. Diberikan himpunan  $n$  titik dalam dimensi  $d$ ,  $\mathbb{R}^d$ , dan sebuah bilangan bulat  $K$ . Tugas dari *k-means* adalah menentukan himpunan  $K$  titik dalam  $\mathbb{R}^d$ , yang disebut *centers*, untuk meminimalkan nilai *mean squared distance* dari setiap titik ke *center* terdekat. *Center* dalam istilah lain disebut *centroid*. Secara matematis, untuk himpunan *disjoint*  $S_j$  yang berisi  $N_j$  titik, *k-means* bekerja dengan cara meminimalkan nilai  $J$  pada persamaan

$$J = \sum_{j=1}^K \sum_{n \in S_j} |x_n - \mu_j|^2, \quad (2.2)$$

di mana  $x_n$  adalah vektor yang merepresentasikan titik ke- $n$  dan  $\mu_j$  merepresentasikan *centroid* dari titik-titik dalam himpunan  $S_j$  (Weisstein, 2016). Walaupun algoritma *k-means clustering* tidak menghasilkan solusi *global optimum* untuk  $J$ , namun algoritma ini sering digunakan oleh banyak praktisi karena mudah untuk diimplementasikan.

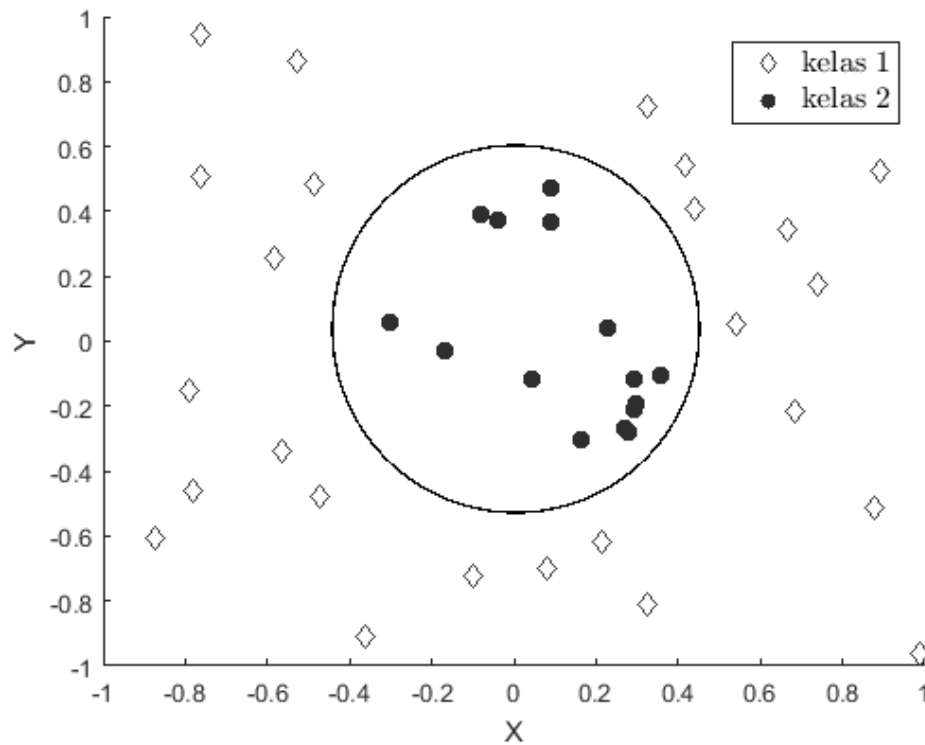
Metode *k-means clustering* secara singkat terdiri dari langkah-langkah sebagai berikut.

1. Masukkan seluruh titik ke dalam  $K$  himpunan secara acak.
2. Hitung *centroid* pada setiap himpunan.
3. Kelompokkan ulang setiap titik ke kelompok yang memiliki *centroid* terdekat dengan titik tersebut.
4. Ulangi langkah 2 sampai 3 hingga syarat berhenti terpenuhi, misalnya tidak adanya perubahan kelompok pada setiap titik.

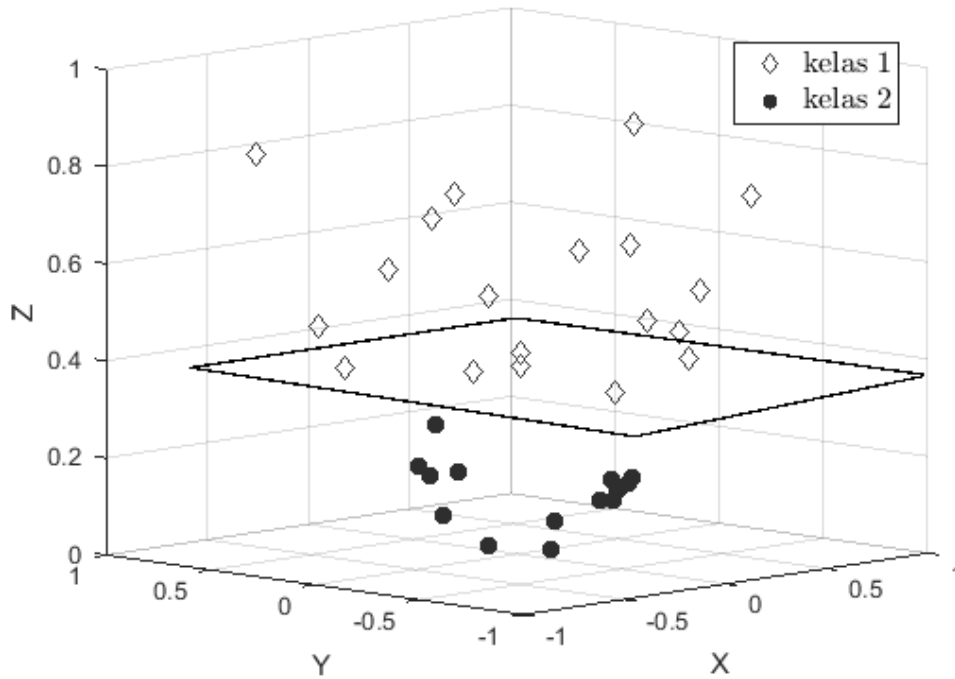
### 2.7.2 Support Vector Machine (SVM)

*Support Vector Machine* (SVM) adalah metode pembelajaran mesin yang relatif baru untuk klasifikasi biner, yaitu klasifikasi yang hanya terdiri dari dua kelas (Boswell, 2016). Ide dasar dari SVM adalah mencari *hyperplane* yang memisahkan data  $d$ -dimensi secara linier menjadi dua kelas. Namun data sampel terkadang tidak dapat dipisahkan secara linier, sehingga SVM memperkenalkan konsep *kernel induced feature space*, yaitu memetakan data ke dimensi ruang yang lebih tinggi

supaya data dapat dipisahkan secara linier. Gambar 2.5 mengilustrasikan data dalam dimensi 2 yang tidak dapat dipisahkan secara linier. Gambar 2.6 mengilustrasikan data tersebut setelah dipetakan ke dimensi 3 dengan fungsi  $\phi(x,y) = (x,y,x^2 + y^2)$ , sehingga dapat dipisahkan secara linier menggunakan *hyperplane*.



**Gambar 2.5:** Ilustrasi Data dalam Dimensi 2



**Gambar 2.6:** Ilustrasi Data dalam Dimensi 3

Secara matematis, cara kerja SVM dalam melakukan klasifikasi adalah sebagai berikut. Diberikan  $l$  sampel *training*,  $\{\vec{x}_i, y_i\}$  untuk  $i = 1, 2, \dots, l$ ;  $\vec{x}_i \in \mathbb{R}^d$ ; dan  $y_i \in \{-1, 1\}$ . Semua *hyperplane* di  $\mathbb{R}^d$  memiliki parameter berupa sebuah vektor  $\vec{w}$  dan sebuah konstanta  $b$ , yang dinyatakan dalam bentuk persamaan

$$\vec{w} \cdot \vec{x} + b = 0. \quad (2.3)$$

Diberikan sebuah *hyperplane*  $(\vec{w}, b)$  yang memisahkan data, maka terdapat fungsi

$$f(\vec{x}) = \text{sign}(\vec{w} \cdot \vec{x} + b) \quad (2.4)$$

yang mengklasifikasi data *training* dengan benar. Syarat data terklasifikasi dengan benar adalah  $\forall_i (f(\vec{x}_i) y_i \geq 1)$ .

Pemetaan ke dimensi ruang yang lebih tinggi membutuhkan proses komputasi yang lebih banyak, serta dapat membuat proses klasifikasi mengalami *overfitting*. Karena itu pemetaan vektor tidak dilakukan dalam SVM secara langsung. Pemetaan terjadi secara tidak langsung di dalam *kernel function*. *Kernel function* adalah fungsi yang digunakan untuk menghitung nilai perkalian dua buah vektor dalam SVM. *Kernel function*  $K(\vec{x}_a, \vec{x}_b)$  menerima *input* berupa dua buah vektor dan memberikan *output* berupa sebuah nilai skalar. Salah satu contoh *kernel function*

adalah  $K(\vec{x}_a, \vec{x}_b) = (\vec{x}_a \cdot \vec{x}_b + 1)^2$ .

### 2.7.3 Gaussian Mixture Model (GMM)

Menurut Valverde et al. (2012), *Gaussian mixture distribution* atau *Gaussian mixture model* (GMM) adalah gabungan dari  $L$  komponen *Gaussian distribution*. Untuk *random variable* satu dimensi,  $Y$ , *probability density function* (PDF) dari  $Y$  didefinisikan sebagai

$$f_Y(y) = \sum_{i=1}^L \omega_i f_{N(\mu_i, \sigma_i^2)}(y), \quad (2.5)$$

di mana  $\omega_i$ ,  $\mu_i$ , dan  $\sigma_i^2$  secara berurutan menyatakan proporsi, rata-rata, dan varian dari komponen ke- $i$  di *Gaussian mixture* (Sirisena dan Brown, 1983). PDF adalah fungsi pada suatu distribusi *random variable* kontinu yang menyatakan probabilitas suatu nilai berada pada interval distribusi tersebut.

Untuk mempertahankan karakteristik dari *probability distribution*, parameter memiliki batasan yaitu,  $0 < \omega_i \leq 1$  dan  $\sum_{i=1}^L \omega_i = 1$ . Menurut buku *Applied Statistics and Probability for Engineers* yang ditulis oleh Montgomery (2013), cara menghitung distribusi dari komponen *Gaussian* ke- $i$ ,  $f_{N(\mu_i, \sigma_i^2)}$ , adalah sebagai berikut.

$$f_{N(\mu_i, \sigma_i^2)}(x) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{\left(\frac{-(x-\mu_i)^2}{2\sigma_i^2}\right)}. \quad (2.6)$$

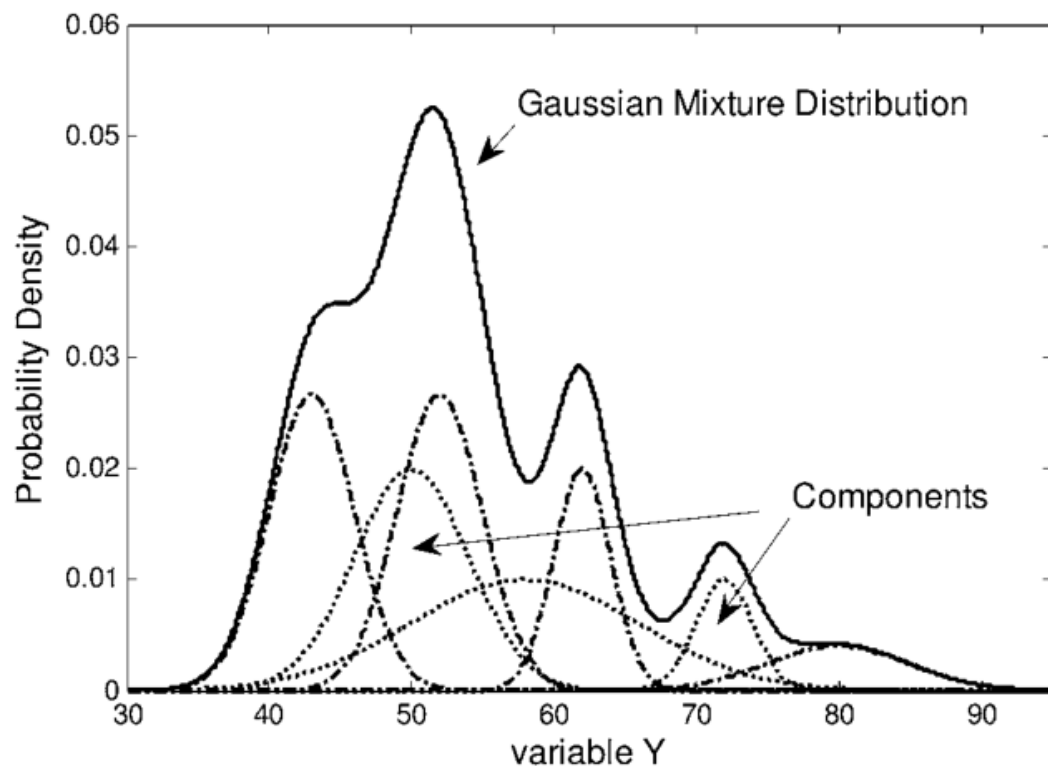
Salmond (2009) menjelaskan cara menghitung rata-rata ( $\mu$ ) dan varian ( $\sigma^2$ ) dari *random variable*  $Y$  adalah sebagai berikut.

$$\mu_Y = \sum_{i=1}^L \omega_i \mu_i \quad (2.7)$$

$$\sigma_Y^2 = \sum_{i=1}^L \omega_i (\sigma_i^2 + (\mu_i - \mu_Y)^2). \quad (2.8)$$

Gambar 2.7 menunjukkan *probability density* dari *random variable*  $Y$  yang dimodelkan menggunakan *Gaussian distribution* dengan 7 komponen. Jumlah setiap komponen berbobot *Gaussian* menghasilkan *Gaussian mixture distribution*





**Gambar 2.7:** *Gaussian Mixture Distribution* dengan 7 Komponen

Sumber gambar: Valverde et al. (2012)

Keuntungan menggunakan GMM adalah sembarang PDF dengan banyak komponennya terbatas dapat didekati. Biasanya semakin banyak komponen yang ada pada GMM, maka pendekatan tersebut semakin akurat. Metode paling efektif untuk menentukan GMM yang paling mendekati distribusi sampel *random variable*  $Y$  adalah menggunakan algoritma *expectation maximization* (Singh et al., 2010).

## 2.8 Evaluasi

Hasil pada klasifikasi biner dapat dikategorikan menjadi dua, yaitu *true* (+) dan *false* (-). Perbandingan hasil klasifikasi dengan hasil yang diharapkan dapat dilihat pada tabel 2.1.

**Tabel 2.1:** Kontingensi

Hasil Klasifikasi	Hasil yang Diharapkan	
	+	-
+	True Positive (TP)	False Positive (FP)
-	False Negative (FN)	True Negative (TN)

Dari tabel 2.1 dapat dihitung beberapa metrik evaluasi, yang dapat digunakan untuk melihat dan menganalisis hasil eksperimen. Beberapa contoh metrik evaluasi antara lain sebagai berikut.

$$Accuracy = \frac{TP + FN}{TP + FN + FP + TN} \quad (2.9)$$

$$Precision = \frac{TP}{TP + FP} \quad (2.10)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.11)$$

$$F - measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (2.12)$$

Nilai-nilai pada metrik evaluasi memiliki makna yang berbeda-beda. *Accuracy* (persamaan 2.9) menyatakan rasio antara banyaknya hasil klasifikasi yang sesuai dengan hasil yang diharapkan dengan seluruh data yang ada. Rasio antara banyaknya hasil positif yang sesuai harapan, terhadap banyaknya hasil klasifikasi yang memberikan nilai positif (+) disebut *precision* (persamaan 2.10). Kemudian *recall* (persamaan 2.11) menyatakan rasio antara banyaknya yang diklasifikasikan positif dengan banyaknya data yang diharapkan positif. Namun terkadang nilai *precision* bersaing dengan nilai *recall*. Semakin tinggi nilai *precision* umumnya akan membuat nilai *recall* semakin rendah, dan sebaliknya. Nilai *F-measure* (persamaan 2.12) menyatakan rata-rata harmonik antara nilai *precision* dengan nilai *recall*.

## 2.9 Penelitian Terkait

Penelitian ini membahas sistem *automatic speech recognition* (ASR) dalam domain bacaan Al-Qur'an. Beberapa pekerjaan yang sudah dilakukan oleh peneliti sebelumnya terkait bidang ASR dengan domain Al-Qur'an antara lain, yang *pertama* adalah penelitian yang dilakukan oleh Muhammad et al. (2010). Penelitiannya membahas tentang pencocokan suara bacaan Al-Qur'an. Tujuannya adalah membangun sistem yang dapat memberitahukan kesalahan pada pelafalan bacaan Al-Qur'an yang tidak sesuai dengan pelafalan yang seharusnya. Penelitiannya menggunakan MFCC sebagai fitur.

Penelitian *kedua* dilakukan oleh Putra et al. (2012) yang dimuat dalam jurnalnya, *Developing Speech Recognition System for Quranic Verse Recitation Learning Software*. Jurnal tersebut membahas mengenai pengembangan perangkat lunak untuk membantu pembelajaran dalam membaca Al-Qur'an. Penelitian dalam

jurnal tersebut menggunakan MFCC sebagai fitur dan GMM sebagai model.

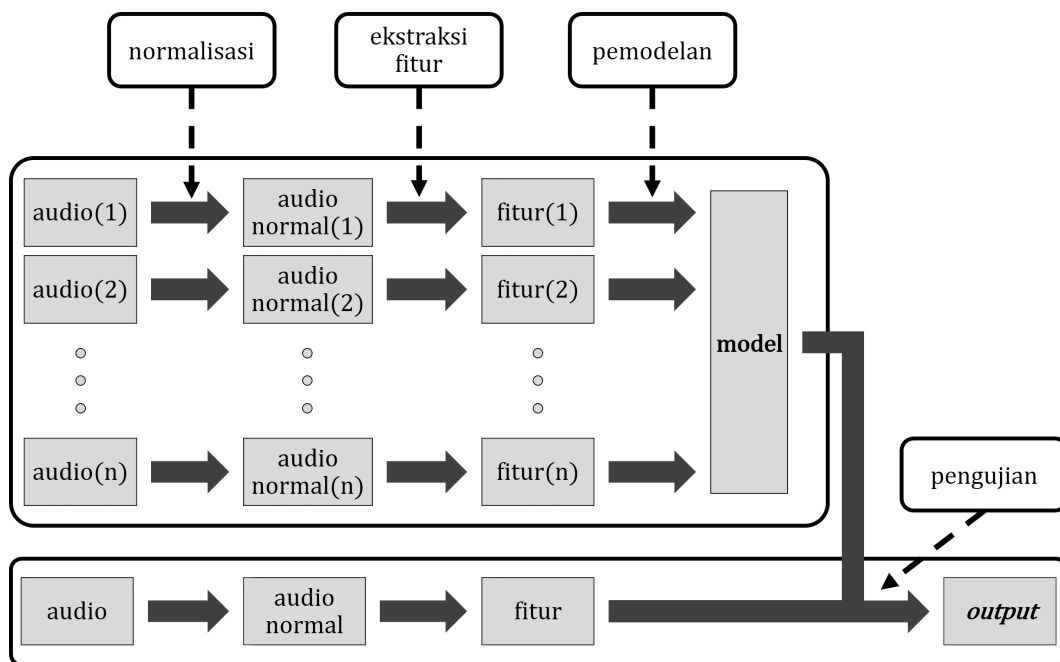
Penelitian *ketiga* dilakukan oleh Mohammed et al. (2015). Penelitiannya bertujuan untuk mencegah pengubahan suara bacaan Al-Qur'an yang banyak tersebar di *internet* dan media sosial secara otomatis. Pengubahan yang dimaksud adalah penambahan, pengurangan, atau kesalahan dalam melafalkan bacaan ayat-ayat Al-Qur'an. Penelitian tersebut menggunakan teks sebagai fitur. Proses ekstraksi teks dari audio lebih kompleks dibandingkan dengan proses ekstraksi MFCC. Untuk mengekstrak teks dari suara setidaknya diperlukan model akustik dan model bahasa. Proses ini menggunakan *Hidden Markov Model* (HMM) sebagai model untuk mencari tahu teks mana yang paling memungkinkan untuk menjadi *output*. Penelitian tersebut memberikan kesimpulan bahwa untuk Bahasa Arab, fitur yang tepat digunakan adalah MFCC dengan model HMM, berdasarkan teknologi yang berkembang saat ini.

## BAB 3

### PERANCANGAN SISTEM

#### 3.1 Rancangan Arsitektur Sistem

Sistem yang akan penulis buat dalam penelitian ini sesuai dengan rancangan arsitektur sistem pada gambar 3.1.



**Gambar 3.1:** Rancangan Arsitektur Sistem

Berikut merupakan penjelasan alur dalam rancangan arsitektur sistem.

##### 3.1.1 Pengumpulan Data

Penulis mengumpulkan data dari beberapa situs forum kesehatan *input*. Forum tersebut yaitu:

1. lala
2. lili

Dalam proses pengumpulan data, penulis menggunakan data yang digunakan Herwando (2016) dalam penelitiannya, ditambah dengan data dari forum terbaru. Penulis menggunakan program untuk melakukan otomatisasi. Program

pertama merupakan program untuk mengumpulkan data *input* untuk model *word embedding* dan program kedua merupakan program untuk mengumpulkan data forum kesehatan *online* di situs di atas. Program kedua merupakan program hasil modifikasi dari Herwando (2016).

### 3.1.2 Proses Normalisasi

Proses normalisasi penulis lakukan untuk setiap data, baik data dari penelitian Herwando (2016) maupun data yang penulis dapatkan dari forum kesehatan terbaru. Pada tahap ini, penulis melakukan normalisasi 2 data, yaitu data untuk membuat model *word embedding* dan data sebagai *input* sistem. Normalisasi yang dilakukan dalam penelitian ini adalah:

1. Penghapusan karakter yang tidak ada di dalam daftar ASCII
2. Perbaikan karakter spasi dalam dokumen
3. Pengubahan beberapa tanda baca/token menjadi 1 token tersendiri. Misalnya sebuah token url diganti menjadi kata "url".

Setelah penulis melakukan proses di atas, sama seperti yang dilakukan Herwando (2016), penulis menggabung masing-masing data menjadi 2 kelompok besar, yaitu data sebagai *input* model *word embedding* dan data sebagai *input* sistem. Untuk data sebagai *input* model *word embedding*, setiap baris merupakan 1 buah kalimat yang masing-masing kata dipisahkan dengan karakter spasi. Selain itu, setiap tanda baca juga dipisahkan dengan karakter spasi. Untuk data sebagai *input* sistem, penulis mengubah format data menjadi format yang dipakai oleh Raditya Herwando dalam penelitiannya:

FOOOORRRMMMAAATTT

Setelah proses pemformatan selesai, penulis memberikan label secara manual pada data yang baru (karena data dari Raditya Herwando sudah diberikan label sebelumnya). Label yang penulis gunakan sama dengan label yang Raditya Herwando gunakan dalam penelitiannya, yaitu:

1. *online*: gejala penyakit,
2. *disease*: nama penyakit,
3. *treatment*: pengobatan penyakit, dan
4. *drug*: nama obat.

Format pelabelan yang penulis gunakan adalah format format xml. Satu buah entitas yang masuk dalam kategori entitas kesehatan akan diberikan tag. Entitas yang terdiri dari lebih dari 1 kata akan diberikan 1 buah tag saja. Berikut merupakan contoh teks yang sudah diberikan label:

CCOOONNNTTTTOOOHHH

Setelah pelabelan seara manual selesai dilakukan, penulis akan membuat statistik dari data yang ada. Statistik ini bertujuan untuk mengetahui informasi dari masing-masing kategori yang penulis definisikan di atas. Selain itu, penulis juga ingin mengetahui distribusi penyakit yang ada dalam data sehingga penulis dapat membagi data yang seimbang dalam melakukan eksperimen nantinya.

### 3.1.3 Pembuatan Model *Word Embedding*

Model *word embedding* digunakan untuk mendapatkan representasi vektor dari kata dalam data.

### 3.1.4 Ekstraksi Fitur

Sebelum masuk tahap eksperimen, penulis melakukan ekstraksi fitur. Ada beberapa fitur yang penulis usulkan dalam penelitian ini. Fitur-fitur tersebut penulis kombinasikan supaya mendapatkan hasil terbaik. Fitur-fitur tersebut yaitu:

#### 1. Fitur 1: Kata itu sendiri

Fitur ini merupakan fitur kata dalam representasi vektor. Dalam mengubah kata menjadi vektor, penulis menggunakan *tools word embedding* yang merupakan hasil penelitian Řehůřek dan Sojka (2010). KENAPA?

#### 2. Fitur 2: *Part of Speech Stanford postagger*

Fitur ini merupakan fitur *tag* yang dimiliki setiap kata. Dalam mengubah setiap kata menjadi *tag* dalam POS Tag, penulis menggunakan *tools Stanford POS Tagger* yang menggunakan bahasa pemrograman Java. Penulis membuat sebuah kelas yang mengubah setiap baris kalimat menjadi baris *tag*. Kemudian, penulis mengubah masing-masing kalimat menjadi kumpulan vektor *tag*. KENAPA?

#### 3. Fitur 3: *Stopword*

Fitur ini merupakan fitur yang berisi vektor suatu kata merupakan *stopword* atau bukan. KENAPA?

#### 4. Fitur 4: Kamus kata

### 3.1.5 Eksperimen

Dalam melakukan Eksperimen, penulis menggunakan *Recurrent Neural Network Tools* yang penulis gunakan yaitu Keras. Keras merupakan *library* untuk *Deep Learning* yang *high level* yang ditulis dalam bahasa pemrograman python. Keras yang penulis gunakan berjalan di atas Theano. Sebelum memulai eksperimen, penulis membagi data menjadi beberapa bagian untuk melakukan *n-fold cross validation*. Hal ini penulis lakukan karena data yang digunakan sedikit. Dalam melakukan pembagian data, penulis menggunakan program buatan penulis sendiri dengan menggunakan bahasa pemrograman Python. Setelah data telah terbagi menjadi n-bagian, penulis melakukan pengkombinasian fitur-fitur supaya mendapatkan hasil terbaik.

## 3.2 Data

Penelitian ini memerlukan data berupa berkas audio bacaan seluruh ayat Al-Qur'an di juz 30 dari berbagai qari. Karena sistem yang dikembangkan adalah pengenalan suara per ayat, maka data yang diperlukan berupa bacaan Al-Qur'an yang sudah dipotong per ayat. Dengan kata lain, satu *instance* data merepresentasikan satu ayat Al-Qur'an yang dibaca oleh seorang qari. Berikut adalah langkah-langkah untuk memperoleh data tersebut dan memilih data yang tepat untuk digunakan dalam eksperimen.

### 3.2.1 Pengambilan Data

Data diperoleh dengan cara mengunduh dari beberapa sumber, yaitu <http://everyayah.com>, <http://tanzil.net>, dan <http://recitequran.com>. Tiga sumber tersebut memberi penomoran yangurut pada datanya. Penomoran berkas yangurut memudahkan proses pengunduhan untuk dilakukan secara otomatis menggunakan bantuan program. Nama berkas untuk surat ke-*s* dan ayat ke-*a* adalah *sssaaa.mp3*. Contoh untuk surat ke-78 ayat pertama adalah *078001.mp3* dan surat ke-114 ayat ke-6 adalah *114006.mp3*. Data yang diambil adalah seluruh surat, mulai dari surat ke-78 sampai dengan surat ke-114 (seluruh surat dalam Al-Qur'an di juz 30). Total ayat dari surat-surat yang diambil adalah 564 ayat untuk masing-masing qari.

Di dalam tiga *website* yang telah disebutkan sebelumnya tersedia data berupa berkas mp3, di mana satu berkas merepresentasikan satu ayat. Masing-masing berkas memiliki informasi detil yang berbeda-beda, terutama pada *duration*, *bit rate*, dan *channel*. Tabel 3.1 menyajikan informasi detil salah satu berkas, yaitu surat An-Naba' ayat 7 dengan qari Mishary Rashid Alafasy.

**Tabel 3.1:** Informasi Detil Surat An-Naba' Ayat 7 dengan Qari Mishary Rashid Alafasy

Format	MPEG Audio
Format version	Version 1
Format profile	Layer 3
Mode	Joint stereo
Mode extension	MS Stereo
Duration	3.866 s
Bit rate mode	Constant
Bit rate	128 Kbps
Channel(s)	2 channels
Sampling rate	44.1 KHz
Compression mode	Lossy
Stream size	60.4 KiB (99%)
Writing library	LAME3.96r
Encoding settings	-m j -V 4 -q 3 -lowpass 17.5 -b 128

### 3.2.2 Pembuangan Data Duplikat

Dari tiga sumber yang telah disebutkan pada Bab 3.2.1, diperoleh data bacaan Al-Qur'an lebih dari 50 qari berbeda. Namun di antara data-data tersebut, ada beberapa data yang duplikat, yaitu data dengan nama berkas berbeda tetapi sebenarnya memiliki suara yang sama persis. Hal ini dikarenakan ada sedikit perbedaan dari masing-masing sumber data dalam menamai qari. Misalnya dalam <http://everyayah.com> terdapat qari dengan nama "Alafasy\_128kbps" dan dalam <http://tanzil.net> terdapat qari dengan nama "afasy". Dua nama tersebut merujuk ke orang yang sama. Untuk mengetahui bahwa dua berkas memiliki konten audio yang sama tidak dilihat dari nama berkasnya, tetapi menggunakan teknik lain. Dalam penelitian ini teknik yang digunakan adalah perbandingan nilai MFCC. Langkah-langkah dari teknik tersebut adalah sebagai berikut.

#### 1. Menentukan satu ayat yang akan menjadi acuan

Ayat yang dipilih boleh yang mana saja, tetapi disarankan bukan ayat pertama



dari setiap surat. Ayat pertama dari setiap surat terkadang mengandung bacaan *basmalah*<sup>1</sup> yang sebenarnya bukan merupakan bagian dari ayat tersebut, karena bacaan tersebut sifatnya opsional untuk dibaca di awal surat. Ayat yang digunakan pada langkah selanjutnya hanya ayat yang terpilih pada langkah ini. Dalam penelitian ini ayat yang digunakan sebagai acuan adalah ayat ke-6 dari surat ke-114.

## 2. Ekstraksi MFCC dari semua qari

Dengan mengacu pada ayat yang ditentukan di langkah 1, audio bacaan Al-Qur'an masing-masing qari diekstrak nilai MFCC-nya. Dalam penelitian ini, parameter koefisien MFCC yang digunakan adalah 13. Hasil dari ekstraksi nilai MFCC berupa matriks berukuran  $13 \times K$ , di mana  $K$  merepresentasikan banyaknya *frame*. Nilai  $K$  bervariasi, bergantung pada durasi audio yang diekstrak. Kemudian, nilai rata-rata dari matriks  $13 \times K$  pada setiap barisnya dihitung, sehingga terbentuk vektor kolom dengan panjang 13. Vektor kolom adalah matriks yang memiliki tepat 1 kolom. Penghitungan rata-rata dilakukan per baris supaya panjang vektor dari hasil perhitungan ini konstan, yaitu 13. Karena jika yang dihitung nilai rata-rata per *frame*, maka hasilnya tentu bervariasi sesuai dengan banyak *frame* dari masing-masing nilai MFCC, yang dipengaruhi oleh durasi audio. Panjang setiap vektor dari langkah ini haruslah sama supaya masing-masing vektor dapat dihitung nilai jaraknya.

## 3. Menghitung jarak dari setiap pasang data

Setiap pasang vektor yang terbentuk dari langkah 2 kemudian dihitung nilai jaraknya. Sehingga akan terbentuk matriks  $D$  berukuran  $N \times N$  yang berisi nilai-nilai jarak tersebut, di mana  $N$  adalah banyaknya qari. Baris ke- $i$  dan kolom ke- $j$  menyatakan nilai jarak dari qari ke- $i$  dengan qari ke- $j$ . Penelitian ini menggunakan nilai jarak *euclidean distance*<sup>2</sup>.

## 4. Mengelompokkan matriks $D$

Kumpulan sel di matriks  $D$  hasil dari langkah 3 kemudian dikelompokkan ke dalam 4 kelompok menggunakan teknik *k-means clustering*. Pembagian ke dalam 4 kelompok ini dimaksudkan supaya data terbagi ke dalam kelompok *mirip*, *sedikit mirip*, *tidak terlalu mirip*, dan *tidak mirip*. Kelompok yang mengandung nilai terendah adalah kelompok *mirip* yang ditetapkan sebagai pasangan data yang duplikat. Jika pembagiannya hanya 3 atau 2 kelompok saja, maka akan ada pasangan-pasangan data yang hanya *sedikit mirip*,

<sup>1</sup>sebutan untuk bacaan “بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ”.

<sup>2</sup>persamaan 2.1

yang jika diamati secara manual berbeda (tidak duplikat), tetapi masuk ke dalam kelompok *mirip*, sehingga dianggap duplikat. Dari langkah ini akan dihasilkan matriks C yang berukuran sama dengan matriks D, di mana setiap sel pada matriks C berisi label kelompok. Nilai sel pada baris ke-*i* dan kolom ke-*j* menyatakan level kemiripan qari ke-*i* dengan qari ke-*j*.

#### 5. Membuang data mirip

Dengan mengacu pada matriks C yang diperoleh pada langkah 4, setiap pasangan data yang memiliki hubungan *mirip*, salah satu dari keduanya dihilangkan dari himpunan data eksperimen pada penelitian ini.

### 3.2.3 Penyaringan Data

Setelah data yang terkumpul dipastikan tidak mengandung data yang duplikat, maka langkah selanjutnya adalah menyaring data. Proses ini dilakukan secara manual. Adapun kriteria data yang lolos saringan adalah sebagai berikut:

#### 1. Terdengar jelas dan bacaannya benar.

Suara bacaan Al-Qur'an harus terdengar jelas oleh pendengaran manusia. Data yang tidak jelas menurut pendengaran manusia tidak sesuai standar dalam penelitian ini, sehingga tidak disertakan dalam eksperimen. Di samping itu, bacaan dalam data tersebut juga harus sesuai dengan cara membaca Al-Qur'an yang benar.

#### 2. Tidak mengandung bacaan *basmalah*.

Sebagian qari mengawali pembacaan ayat pertama di setiap surat dengan *basmalah*. Karena bacaan ini bukan bagian dari ayat pertama dan sifatnya opsional, ada qari yang membacanya dan ada juga yang tidak. Baik membaca *basmalah* maupun tidak, keduanya benar menurut aturan membaca Al-Qur'an. Oleh karena itu, dalam penelitian ini perlu diterapkan standar, yaitu data yang digunakan hanya data yang tidak mengandung bacaan *basmalah*.

#### 3. Tidak mengandung pantulan suara.

Beberapa suara bacaan Al-Qur'an mengandung pantulan suara atau *echo*. *Echo* dalam rekaman bacaan Al-Qur'an disebabkan oleh efek audio digital yang sengaja ditambahkan untuk memberikan kesan bagus. Namun hal tersebut justru akan mengganggu proses klasifikasi. Maka dalam penelitian ini, data yang mengandung *echo* tidak disertakan dalam eksperimen.

Informasi mengenai data yang tersisa setelah proses pembuangan data duplikat dan penyaringan, dapat dilihat pada tabel 3.2.

**Tabel 3.2:** Informasi Data Eksperimen

Banyak Qari	40 qari
Surat Mulai	An-Naba' (surat ke-78)
Surat Selesai	An-Nas (surat ke-114)
Banyak Surat	37 surat
Banyak Ayat	564 ayat
Total Seluruh <i>Instance</i>	22,560 <i>instance</i>

### 3.3 Perangkat dan Fungsi Pendukung

#### 3.3.1 Perangkat Pendukung

Dalam melakukan eksperimen, penulis menggunakan perangkat komputer yang disediakan di lab. Perangkat komputer tersebut memiliki spesifikasi seperti pada tabel 3.3.

**Tabel 3.3:** Spesifikasi *Hardware*

Processor	i7-4770S
Banyak Core	8 core
Frekuensi Processor	3.1 GHz per core
RAM	8 GB

Perangkat lunak yang penulis gunakan dalam mengembangkan sistem yaitu Sublime. Bahasa pemrograman yang penulis pilih yaitu Python untuk pengolahan dan pengembangan sistem dan Ruby untuk pengambilan data dari internet. *Framework Recurrent Neural Network* yang penulis gunakan adalah Keras. Penulis memilih Keras karena *framework* ini lebih *high level* dibandingkan dengan *framework* lain dan memiliki *library* yang cukup lengkap. Selain itu, pengembang Keras juga masih aktif dalam mengembangkan *framework* dan diskusi mengenai Keras. Sehingga penulis lebih mudah dalam membangun sistem. Untuk mengubah kata menjadi vektor, penulis menggunakan *word embedding* dari Gensim. Alasan penulis memilih *library* tersebut yaitu relatif lebih mudah digunakan.

#### 3.3.2 Fungsi Pendukung

MATLAB menyediakan banyak fungsi yang dapat digunakan untuk membantu eksperimen. Namun selain fungsi-fungsi yang sudah disediakan MATLAB,

diperlukan fungsi tambahan untuk menjalankan eksperimen ini, antara lain sebagai berikut.

#### 1. audioread

*audioread* adalah fungsi untuk membaca berkas mp3 yang sudah tersedia di MATLAB mulai dari versi R2012b<sup>3</sup>. Jika fungsi *audioread* belum ada di MATLAB, maka alternatifnya adalah fungsi *wavread*. Fungsi *audioread* menerima *input* berupa *string* yang menyatakan lokasi penyimpanan berkas, lalu memberikan *output* berupa matriks sinyal  $N \times C$  dan *bit rate*.  $N$  menyatakan panjang sinyal, sedangkan  $C$  menyatakan banyaknya *channel*.

#### 2. mfcc

*mfcc* adalah fungsi untuk menghitung MFCC. Implementasi fungsi *mfcc* menggunakan *source code* dari HTK MFCC MATLAB<sup>4</sup>. Fungsi *mfcc* membutuhkan parameter seperti yang tercantum dalam tabel 3.4.

**Tabel 3.4:** Parameter Fungsi *mfcc*

Nama Parameter	Keterangan
Speech	Data audio dalam bentuk matriks $n \times 1$
Fs	<i>Bit rate</i>
Tw	<i>Analysis frame duration</i> (ms)
Ts	<i>Analysis frame shift</i> (ms)
Alpha	<i>Preemphasis coefficient</i>
Window	<i>Windowing function</i>
R	<i>Frequency range</i>
M	Banyaknya <i>filterbank channels</i>
C	Banyaknya <i>cepstral coefficients</i>
L	Parameter <i>cepstral sine lifter</i>

#### 3. mfcc2sdc

*mfcc2sdc* adalah fungsi untuk menghitung SDCC. Implementasi fungsi tersebut diperoleh dari *source code* di HTK MFCC MATLAB<sup>5</sup>. Fungsi *mfcc2sdc* membutuhkan parameter seperti yang tercantum pada tabel 3.5.

<sup>3</sup><http://www.mathworks.com/help/matlab/ref/audioread.html>

<sup>4</sup><http://www.mathworks.com/matlabcentral/fileexchange/32849-htk-mfcc-matlab>

<sup>5</sup><http://www.mathworks.com/matlabcentral/fileexchange/31478-shifted-delta-coefficients-sdc-computation-from-mel-frequency-cepstral-coefficients-mfcc->

**Tabel 3.5:** Parameter Fungsi *mfcc2sdc*

Nama Parameter	Keterangan
CepCoeff	Matriks MFCC dalam bentuk $K \times C$ , di mana K menyatakan banyaknya <i>frame</i> dan C menyatakan banyaknya koefisien MFCC
D	Nilai <i>shift</i> untuk <i>delta computation</i>
P	Nilai <i>shift</i> untuk <i>frame</i> selanjutnya
K	Banyaknya blok di mana koefisien <i>delta</i> disambungkan

## BAB 4

### EKSPERIMEN

#### 4.1 Normalisasi Data

Data yang sudah diperoleh perlu dinormalisasi terlebih dahulu untuk membuang *noise* yang dapat mengganggu jalannya eksperimen. Normalisasi dilakukan dengan cara membuang suara senyap atau *silence*. Ada beberapa teknik yang digunakan untuk membuang *silence*. Salah satu teknik klasik dan populer yang digunakan untuk membuang *silence* pada suara adalah kombinasi *short term energy* (STE) dengan *zero crossing rate* (ZCR) (Rabiner dan Schafer, 1978). Saha et al. (2005) memaparkan pendekatan lain yang lebih baik akurasi dalam membuang *silence* jika dibandingkan dengan STE maupun kombinasi ZCR dengan STE, yaitu dengan memodelkan sinyal audio menggunakan distribusi *Gaussian*.

Penelitian ini menggunakan pendekatan yang dipaparkan oleh Saha et al. (2005) dalam membuang *silence*, dengan langkah-langkah sebagai berikut.

1. Hitung rata-rata ( $\mu$ ) dan standar deviasi ( $\sigma$ ) dari 10 ms sampel pertama pada audio. Jika  $fs$  menyatakan *bit rate* dari audio, maka terdapat sebanyak  $s$  sampel dalam 10 ms.

$$s = 0.01 \cdot fs \quad (4.1)$$

$$\mu = \frac{\sum_{i=1}^s x(i)}{s} \quad (4.2)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^s (x(i) - \mu)^2}{s}} \quad (4.3)$$

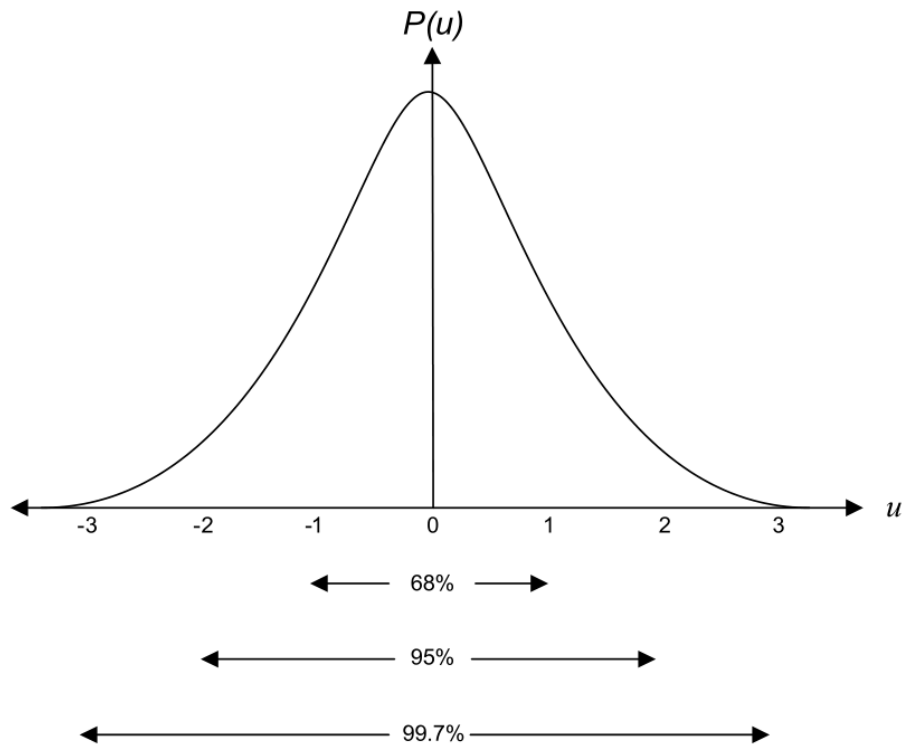
*Background noise* juga dikenali dari berdasarkan nilai rata-rata ( $\mu$ ) dan standar deviasinya ( $\sigma$ ).

2. Periksa apakah jarak *Mahalanobis* mulai dari sampel pertama sampai dengan sampel terakhir pada audio lebih dari 3 atau tidak. Jarak *Mahalanobis* dihitung menggunakan fungsi  $f$  sebagai berikut.

$$d(x) = \frac{|x - \mu|}{\sigma} \quad (4.4)$$

Nilai 3 dipilih karena pada distribusi *Gaussian*, 99.7% data memiliki nilai jarak *Mahalanobis*  $\leq 3$ . Gambar 4.1 menunjukkan persebaran data pada

distribusi *Gaussian*.



**Gambar 4.1:** Distribusi *Gaussian* Terhadap Nilai Jarak Mahalanobis

Sumber gambar: Saha et al. (2005)

3. Tandai sampel yang dianggap bukan sebagai *silence* (selanjutnya disebut *voiced*) dengan 1 dan sampel yang dianggap sebagai *silence* dengan 0. Bagi keseluruhan sinyal audio ke dalam beberapa bagian yang tidak saling *overlap*, dengan durasi pada setiap bagian adalah 10 ms.
4. Asumsikan ada  $M$  sampel yang bernilai 0 dan  $N$  sampel yang bernilai 1. Jika  $M \geq N$ , maka ubah setiap tanda 1 menjadi tanda 0, dan sebaliknya.
5. Kumpulkan seluruh bagian yang dianggap sebagai *voiced*, yaitu bagian-bagian yang bertanda 1 hasil dari langkah 1 sampai langkah 4.

## 4.2 Ekstraksi Fitur

Ekstraksi fitur adalah proses mengubah data *input* menjadi himpunan fitur-fitur yang dapat merepresentasikan data *input* dengan baik. Ekstraksi fitur merupakan bentuk istimewa dari *dimensionality reduction* (Sharma dan Sarma, 2016). Wyse et al. (1980) menjelaskan bahwa ekstraksi fitur adalah proses yang mengekstrak

himpunan fitur-fitur baru dari fitur asli melalui serangkaian fungsi pemetaan. Penelitian ini menggunakan dua jenis fitur, yaitu *mel frequency cepstral coefficient* (MFCC) dan *shifted delta cepstral coefficient* (SDCC). Fitur MFCC dipilih karena MFCC merupakan fitur yang banyak digunakan dalam penelitian di bidang *speech recognition* (Young et al., 2002). Contohnya adalah penelitian yang berjudul *Voice Content Matching System for Quran Readers* oleh Muhammad et al. (2010). Sedangkan fitur SDCC dipilih karena SDCC merupakan fitur yang memuat lebih banyak konteks dalam setiap *frame*-nya jika dibandingkan dengan MFCC.

#### 4.2.1 Ekstraksi Fitur MFCC

Ekstraksi fitur MFCC menggunakan fungsi *mfcc* yang telah dijelaskan di Bab 3.3.2. Fungsi *mfcc* pada eksperimen ini dipanggil dengan parameter fungsi yang dijelaskan pada tabel 4.1.

**Tabel 4.1:** Parameter Pemanggilan Fungsi *mfcc*

Nama Parameter	Nilai Parameter
Tw	25 ms
Ts	10 ms
Alpha	0.97
Window	<i>hamming window</i>
R	[300 3700]
M	20
C	13
L	22

Berikut adalah langkah-langkah untuk memproses fitur MFCC.

1. Tentukan nilai *bit rate* untuk dimasukkan dalam perhitungan MFCC, yang akan mempengaruhi durasi audio. Durasi sebuah audio (dalam detik),  $d$ , dapat dihitung dengan rumus

$$d = \frac{L}{fs}, \quad (4.5)$$

di mana  $L$  adalah panjang *frame* audio dan  $fs$  adalah *bit rate*. Banyaknya *frame* (panjang kolom) pada hasil ekstraksi nilai MFCC dipengaruhi oleh durasi audionya. Maka untuk menyamakan panjang kolom hasil ekstraksi MFCC, durasi audio-audio yang akan diproses harus disamakan terlebih dahulu. Durasi yang digunakan untuk mengekstrak nilai MFCC pada satu ayat tertentu adalah *nilai rata-rata durasi* keseluruhan audio dari berbagai



qari pada ayat tersebut. Setelah diperoleh nilai rata-rata durasi,  $\bar{d}$ , langkah selanjutnya adalah membuat masing-masing audio memiliki durasi yang sama, dengan cara mengubah nilai  $fs$  masing-masing audio. Jika diberikan nilai  $\bar{d}$  dan  $L$ , maka nilai  $fs$  yang baru,  $\hat{fs}$ , dihitung menggunakan rumus

$$\hat{fs} = \frac{L}{\bar{d}}. \quad (4.6)$$

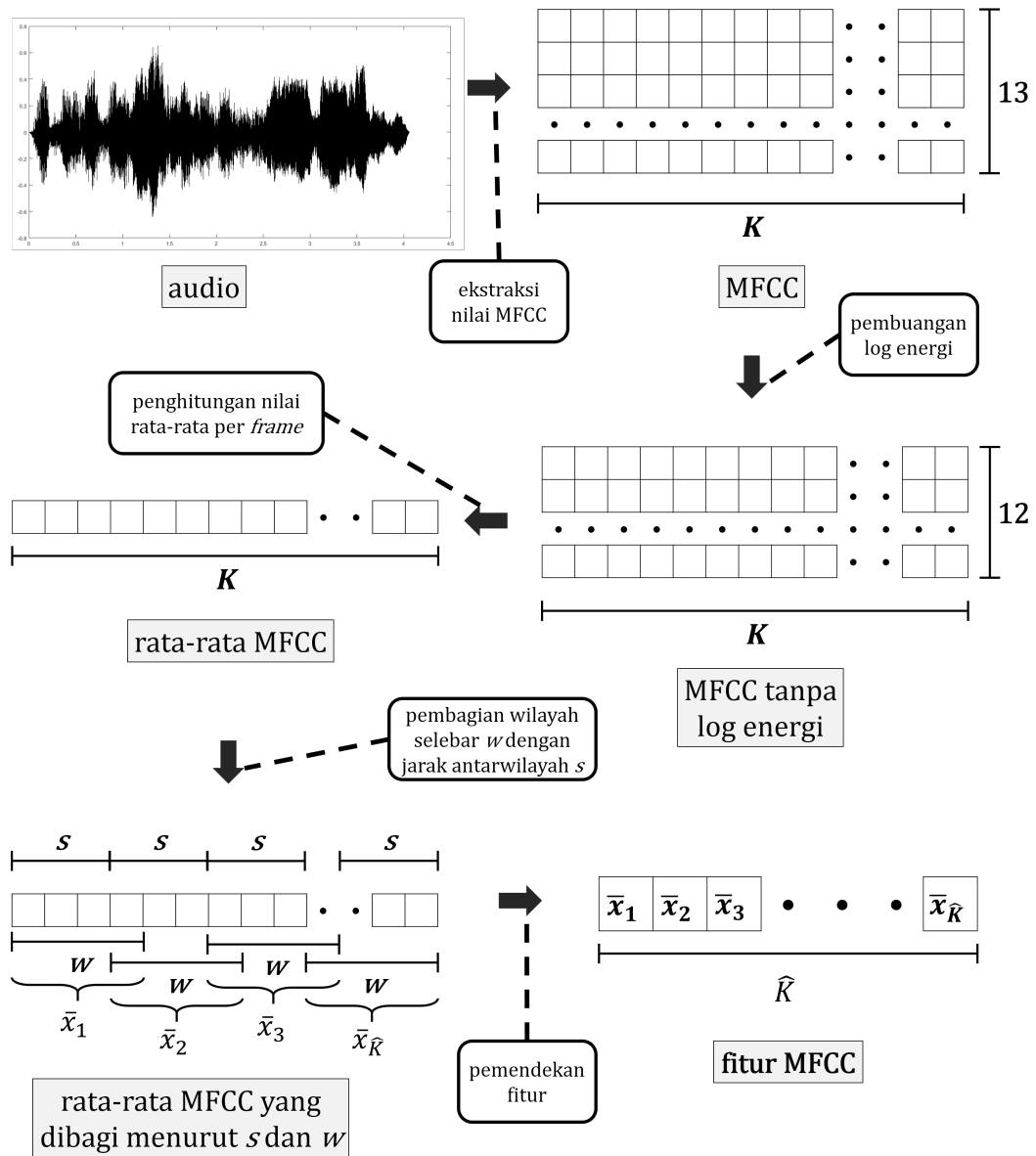
2. Panggil fungsi *mfcc* dengan parameter yang telah disebutkan pada tabel 4.1 dan nilai *bit rate* sama dengan  $\hat{fs}$ , sehingga memberikan *output* berupa matriks MFCC  $13 \times K$ , di mana  $K$  menyatakan banyaknya *frame*.
3. Buang matriks MFCC pada baris pertama, karena nilai tersebut merupakan log energi yang bukan merupakan bagian dari fitur dalam eksperimen ini. Sehingga matriks MFCC yang tersisa berukuran  $12 \times K$ .
4. Hitung nilai rata-rata pada setiap *frame* matriks MFCC. Perhitungan tersebut akan menghasilkan vektor kolom,  $\vec{v}$ , dengan panjang  $K$ . Vektor kolom adalah matriks yang hanya memiliki satu baris.
5. Selanjutnya vektor tersebut dipendekkan dengan cara sebagai berikut.
  - 5.1. Tentukan nilai  $s$  (*shift*).  $s = \lceil \frac{K}{30} \rceil$ .
  - 5.2. Tentukan nilai  $w$  (*width*).  $w = \lceil 1.5 \times s \rceil$ .
  - 5.3. Setiap  $s$  elemen sekali, hitung nilai rata-rata dari  $w$  elemen berurutan pada vektor  $\vec{v}$ .

Hasil pemendekan tersebut berupa vektor  $\hat{\vec{v}}$  dengan panjang  $\hat{K}$ , di mana

$$\hat{K} = \left\lceil \frac{K - w + 1}{s} \right\rceil = \left\lceil \frac{K - 1.5 \lceil \frac{K}{30} \rceil + 1}{\lceil \frac{K}{30} \rceil} \right\rceil \leq 29.$$

Tujuan dari pemendekan vektor ini adalah untuk mengurangi kompleksitas fitur sehingga diharapkan hasil klasifikasi akan lebih akurat.

Gambar 4.2 menunjukkan alur proses ekstraksi fitur MFCC.



**Gambar 4.2:** Alur Ekstraksi Fitur MFCC

#### 4.2.2 Ekstraksi Fitur SDCC

Ekstraksi fitur SDCC menggunakan fungsi *mfcc2sdc* yang telah dijelaskan di Bab 3.3.2. Fungsi *mfcc2sdc* pada eksperimen ini dipanggil dengan parameter fungsi yang dijelaskan pada tabel 4.2.

**Tabel 4.2:** Parameter Pemanggilan Fungsi *mfcc2sdc*

Nama Parameter	Nilai Parameter
D (nilai <i>shift</i> untuk <i>delta computation</i> )	1
P (nilai <i>shift</i> untuk <i>frame</i> selanjutnya)	3
K (banyaknya blok di mana koefisien <i>delta</i> disambungkan)	3

Berikut adalah langkah-langkah untuk memproses fitur SDCC.

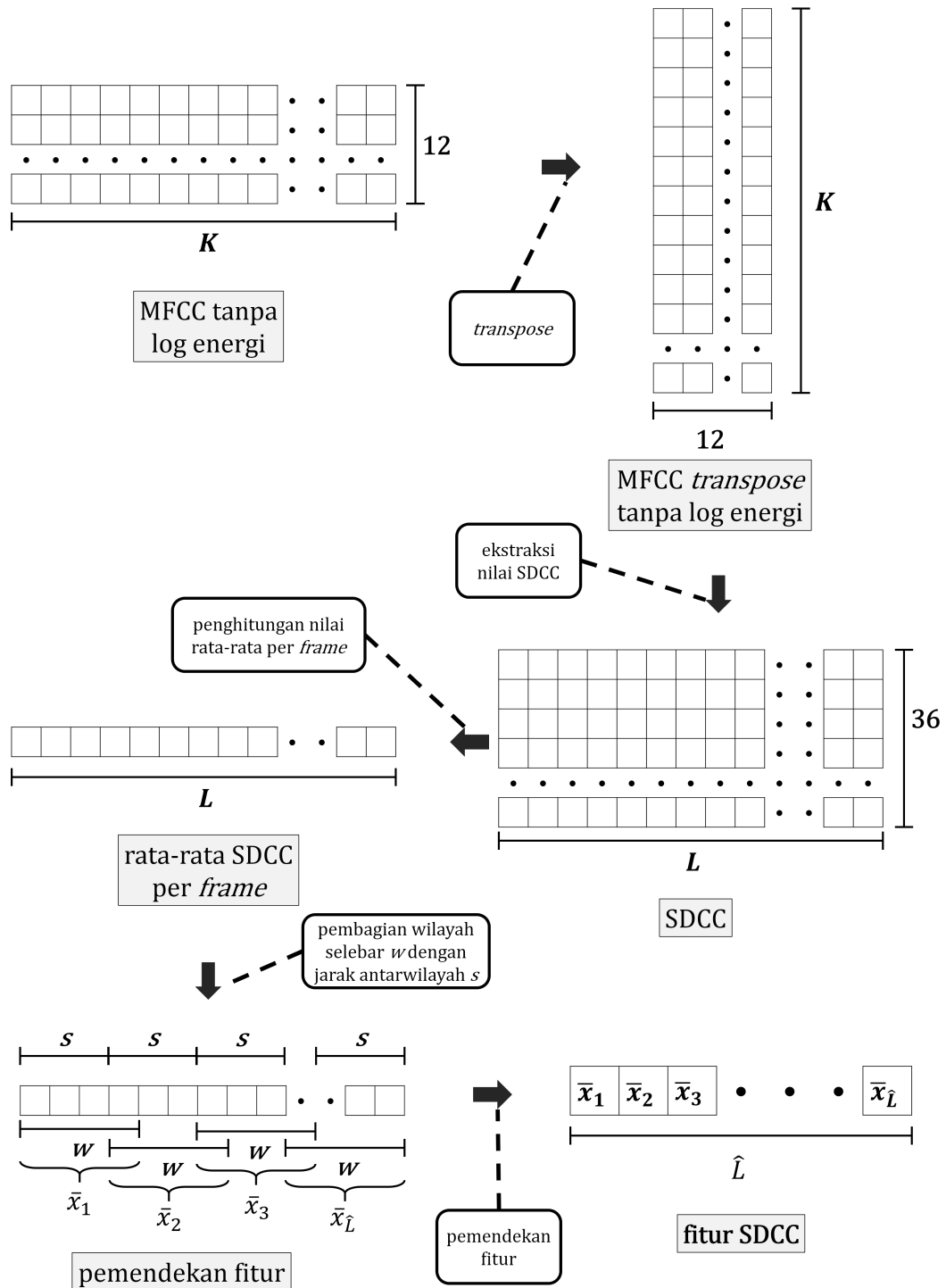
1. Hitung nilai MFCC menggunakan fungsi *mfcc* seperti sudah dijelaskan di Bab 4.2.1, langkah 1 sampai 3, sehingga diperoleh nilai MFCC berupa matriks berukuran  $12 \times K$ . Nilai MFCC berukuran  $12 \times K$  perlu di-*transpose* terlebih dahulu menjadi ukuran  $K \times 12$  untuk menyesuaikan kebutuhan fungsi *mfcc2sdc*, yaitu baris sebagai *frame* dan kolom sebagai koefisien.
2. Panggil fungsi *mfcc2sdc* dengan parameter yang telah disebutkan pada tabel 4.2, sehingga memberikan *output* berupa matriks SDCC  $36 \times L$ , di mana  $L$  menyatakan banyaknya *frame*.
3. Hitung nilai rata-rata pada setiap *frame* matriks SDCC. Perhitungan tersebut akan menghasilkan vektor kolom,  $w$ , dengan panjang  $L$ .
4. Selanjutnya vektor tersebut dipendekkan dengan cara sebagai berikut.
  - 4.1. Tentukan nilai  $s$  (*shift*).  $s = \lceil \frac{L}{30} \rceil$ .
  - 4.2. Tentukan nilai  $w$  (*width*).  $w = \lceil 1.5 \times s \rceil$ .
  - 4.3. Setiap  $s$  elemen sekali, hitung nilai rata-rata dari  $w$  elemen berurutan pada vektor  $\vec{w}$ .

Hasil pemendekan tersebut berupa vektor  $\hat{\vec{w}}$  dengan panjang  $\hat{L}$ , di mana

$$\hat{L} = \left\lceil \frac{L - w + 1}{s} \right\rceil = \left\lceil \frac{L - 1.5 \lceil \frac{L}{30} \rceil + 1}{\lceil \frac{L}{30} \rceil} \right\rceil \leq 29.$$

Tujuan dari pemendekan vektor ini adalah untuk mengurangi kompleksitas fitur sehingga diharapkan hasil klasifikasi akan lebih akurat.

Gambar 4.3 menunjukkan alur proses ekstraksi fitur SDCC.



**Gambar 4.3:** Alur Ekstraksi Fitur SDCC

### 4.3 Pemodelan

Penelitian ini menggunakan tiga variasi metode klasifikasi untuk mengenali pola dari fitur-fitur yang dihasilkan melalui proses pada Bab 4.2, yaitu *support vector machine* (SVM), *Gaussian mixture model* (GMM), serta gabungan SVM dengan

GMM. SVM dipilih untuk digunakan dalam penelitian ini karena SVM merupakan metode klasifikasi biner yang *powerful* (Bishop, 2006). Sedangkan GMM dipilih untuk digunakan dalam penelitian ini karena GMM merupakan model yang banyak digunakan dalam penelitian di bidang *speaker verification* (Reynolds et al., 2000) dan *language identification* (Torres-Carrasquillo et al., 2002), serta sudah menjadi salah satu pendekatan terbaik dalam kedua bidang tersebut (Zahra dan Carson-Berndsen, 2013).

Satu ayat dari berbagai qari dimodelkan menjadi sebuah model klasifikasi. Ayat-ayat yang sama dari para qari akan dijadikan data sampel dengan label *benar*. Ayat-ayat selain ayat tersebut yang tidak mirip secara tekstual akan dijadikan data sampel dengan label *salah*. Cara mengetahui dua buah ayat mirip secara tekstual adalah menggunakan pendekatan algoritma *dynamic programming* yang sudah dijelaskan pada Bab 2.6.

#### 4.3.1 Pemodelan dengan Support Vector Machine (SVM)

SVM dipilih untuk digunakan dalam eksperimen ini karena memiliki kemampuan yang terbukti kuat dalam melakukan klasifikasi pola. Hal tersebut dijelaskan oleh Campbell et al. (2006) dalam jurnalnya yang berjudul *Support Vector Machines for Speaker and Language Recognition*. SVM didasari oleh teori-teori matematika yang kuat, antara lain dengan adalah pemetaan data ke dimensi ruang yang lebih tinggi, pencarian *margin* terbesar, dan generalisasi. Alasan lain yang memperkuat penggunaan SVM dalam penelitian ini adalah karena SVM merupakan metode klasifikasi dua kelas, sesuai dengan kebutuhan sistem untuk mengklasifikasi data ke dalam dua kelas, yaitu kelas *benar* dan kelas *salah*.

Metode klasifikasi SVM dalam eksperimen ini menggunakan fungsi MATLAB *svmtrain*<sup>1</sup> dalam proses pemodelan dan *svmclassify*<sup>2</sup> dalam proses pengujian, dengan parameter yang dijelaskan pada Tabel 4.3 berikut.

**Tabel 4.3:** Parameter SVM dalam Eksperimen

Nama Parameter	Nilai Parameter
kernel_function	linear
method	Sequential Minimal Optimization (SMO)
MaxIter	15000

<sup>1</sup><http://www.mathworks.com/help/stats/svmtrain.html>

<sup>2</sup><http://www.mathworks.com/help/stats/svmclassify.html>

### 4.3.2 Pemodelan dengan Gaussian Mixture Model (GMM)

GMM dipilih dalam eksperimen ini karena banyak penelitian yang menyatakan bahwa GMM dapat memodelkan fitur-fitur *speech recognition* dengan baik, salah satunya adalah penelitian yang dilakukan oleh Zahra dan Carson-Berndsen (2013). Ayat-ayat yang berlabel *benar* dimodelkan menjadi sebuah GMM, sedangkan ayat-ayat yang berlabel *salah* dimodelkan menjadi sebuah GMM lain, sehingga untuk satu ayat terdapat dua GMM yang menjadi model dalam melakukan klasifikasi. Cara menentukan apakah satu data uji masuk ke dalam kelas *benar* atau *salah* adalah dengan membandingkan nilai PDF dari kedua model tersebut. Jika nilai PDF pada GMM yang memodelkan kelas *benar* lebih besar atau sama dengan nilai PDF pada GMM yang memodelkan kelas *salah*, maka data uji tersebut diklasifikasikan sebagai data *benar*. Selain itu maka data uji tersebut diklasifikasikan sebagai data *salah*.

Metode klasifikasi GMM dalam eksperimen ini menggunakan fungsi MATLAB *gmdistribution.fit*<sup>3</sup> dalam proses pemodelan dengan parameter yang dijelaskan pada Tabel 4.4 berikut.

**Tabel 4.4:** Parameter GMM dalam Eksperimen

Nama Parameter	Nilai Parameter
k	8
CovType	diagonal
SharedCov	true
MaxIter	100

### 4.3.3 Pemodelan dengan Gabungan SVM dan GMM

Kemampuan SVM maupun GMM untuk menjadi model sistem ASR sudah teruji baik oleh beberapa penelitian (Zahra dan Carson-Berndsen, 2013). Maka dalam eksperimen ini dicoba pula gabungan dari dua metode klasifikasi tersebut. Cara menggabungkan dua metode tersebut dalam proses pemodelan adalah sebagai berikut.

1. Buat model pertama,  $M_{SVM}$  menggunakan metode klasifikasi SVM dan data sampel.
2. Buat model kedua,  $M_0$  menggunakan GMM dengan sampel data berlabel *salah* dan data sampel.

<sup>3</sup><http://www.mathworks.com/help/stats/gmdistribution.fit.html>

3. Buat model ketiga,  $M_1$  menggunakan GMM dengan sampel data berlabel *benar* dan data sampel.

Sedangkan cara menggabungkan dua metode tersebut dalam proses pengujian adalah sebagai berikut.

1. Hitung nilai PDF dari data uji pada  $M_0$ ,  $P_0$ , sebagai nilai probabilitas *salah*.
2. Hitung nilai PDF dari data uji pada  $M_1$ ,  $P_1$ , sebagai nilai probabilitas *benar*.
3. Klasifikasikan data uji menggunakan metode klasifikasi SVM. dengan model  $M_{SVM}$  Jika hasil klasifikasinya bernilai *salah*, tambahkan nilai 1 pada  $P_0$ , dan jika hasil klasifikasinya bernilai *benar*, tambahkan nilai 1 pada  $P_1$ .
4. Jika nilai  $P_1$  lebih besar atau sama dengan  $P_0$ , maka data uji tersebut diklasifikasikan sebagai data *benar*. Selain itu maka data uji tersebut diklasifikasikan sebagai data *salah*.

#### 4.4 Pengujian

Pengujian dilakukan per ayat menggunakan teknik *k-fold cross validation*, dengan nilai  $k$  yang umum digunakan, yaitu  $k = 10$ . Langkah-langkah dalam melakukan *k-fold cross validation* adalah sebagai berikut.

1. Dari 40 qari yang ada dalam koleksi data eksperimen, gunakan 10% qari (4 qari) *secara acak* sebagai data uji, sedangkan qari-qari lainnya digunakan sebagai data model. Pengacakan tersebut bertujuan untuk membuat proses pengujian menjadi *fair* karena tidak ada beberapa qari yang selalu berada kelompok yang sama, baik sebagai kelompok data uji maupun kelompok data model.
2. Bangun model sesuai penjelasan pada Bab 4.3 menggunakan data model yang diperoleh pada langkah 1.
3. Lakukan proses klasifikasi menggunakan model yang dibangun pada langkah 2 terhadap dua kelompok data berikut.
  - (a) Data berlabel *benar* diperoleh dari ayat-ayat yang sama. Terdapat 4 *instance* berlabel *benar* dari 4 qari yang dijadikan data uji.
  - (b) Data berlabel *salah* diperoleh dari ayat-ayat yang tidak mirip secara tekstual. Ambil 4 *instance* berlabel *salah* secara acak dari 4 qari yang dijadikan data uji.

4. Lakukan evaluasi terhadap hasil klasifikasi pada langkah 3. Evaluasi tersebut akan menghasilkan *confusion matrix*,  $C$ , berukuran  $2 \times 2$  yang merepresentasikan Tabel 2.1.
5. Ulangi langkah 1 sampai langkah 4 sebanyak 10 kali. Qari yang sudah menjadi data uji diganti dengan qari lainnya yang belum pernah menjadi data uji. Sehingga seluruh data qari akan mendapat giliran menjadi data uji. Langkah tersebut akan menghasilkan 10 *confusion matrix*,  $\{C_1, C_2, \dots, C_{10}\}$ .
6. Jumlahkan 10 *confusion matrix* yang diperoleh pada langkah 5, sehingga menghasilkan satu *confusion matrix* baru,  $C_{total} = \sum_{i=1}^{10} C_i$ . Nilai akurasi, presisi, *recall*, serta *f-measure* diperoleh dari perhitungan yang sudah dijelaskan pada Bab 2.8 dengan mengacu pada matriks  $C_{total}$ .



## BAB 5

### HASIL EKSPERIMEN DAN ANALISIS

Sistem pengenalan suara otomatis pada penelitian ini adalah sistem pengenalan per ayat. Bab 4 menjelaskan bagaimana eksperimen dilakukan terhadap satu ayat. Setiap ayat mendapat perlakuan sama dalam eksperimen. Pengujian suatu ayat akan menghasilkan sebuah *confusion matrix*, lalu dari *confusion matrix* diperoleh metrik evaluasi berupa nilai akurasi, presisi, *recall*, dan *f-measure*. Banyaknya ayat yang diproses dalam eksperimen yang sudah dilakukan adalah 564 ayat, sehingga akan dihasilkan pula 564 metrik evaluasi tersebut.

Untuk menampilkan keseluruhan data hasil eksperimen secara ringkas, data tersebut disajikan dalam bentuk histogram. Sumbu X pada histogram menyatakan interval persentase, sedangkan sumbu Y pada histogram menyatakan banyaknya hasil eksperimen yang berada pada interval tersebut. Contoh interval dalam histogram yang disajikan adalah (90%,95%]. Interval tersebut merepresentasikan rentang *lebih dari 90% dan kurang dari atau sama dengan 95%*. Semakin tinggi *bar* histogram akurasi pada interval (90%,95%], artinya semakin banyak ayat yang berhasil diklasifikasikan oleh sistem dengan  $90\% < \text{akurasi} \leq 95\%$ . Jadi jika pada suatu metode, *bar* histogram tinggi berkumpul pada interval tinggi, menunjukkan bahwa metode tersebut baik. Sebaliknya, jika *bar* histogram tinggi berkumpul pada interval rendah, menunjukkan bahwa metode tersebut kurang baik.

## 5.1 Hasil dengan Fitur MFCC

### 5.1.1 Hasil dengan Fitur MFCC dan Metode Klasifikasi SVM

Eksperimen menggunakan fitur MFCC dan menggunakan metode klasifikasi SVM, memberikan hasil yang dirangkum pada Tabel 5.1 berikut.

**Tabel 5.1:** Rangkuman Hasil Eksperimen dengan Fitur MFCC dan Metode Klasifikasi SVM

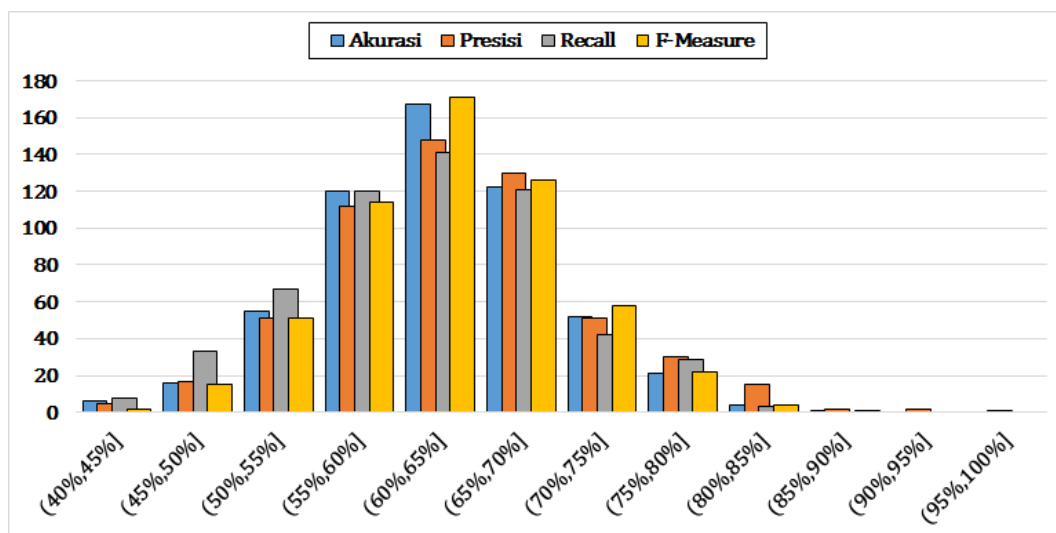
	Akurasi	Presisi	<i>Recall</i>	<i>F-Measure</i>
Minimal	41.3%	42.2%	42.5%	44.4%
Maksimal	87.5%	96.2%	82.5%	86.8%
Rata-rata	63.1%	63.7%	63.0%	63.1%
Standar Deviasi	7.24%	8.09%	7.75%	6.86%

Tabel 5.2 menunjukkan frekuensi kemunculan persentase tertentu yang dikelompokkan ke dalam interval 5%.

**Tabel 5.2:** Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur MFCC dan Metode Klasifikasi SVM

Persentase	Frekuensi Nilai Akurasi	Frekuensi Nilai Presisi	Frekuensi Nilai <i>Recall</i>	Frekuensi Nilai <i>F-Measure</i>
(40%,45%]	6	5	8	2
(45%,50%]	16	17	33	15
(50%,55%]	55	51	67	51
(55%,60%]	120	112	120	114
(60%,65%]	167	148	141	171
(65%,70%]	122	130	121	126
(70%,75%]	52	51	42	58
(75%,80%]	21	30	29	22
(80%,85%]	4	15	3	4
(85%,90%]	1	2	0	1
(90%,95%]	0	2	0	0
(95%,100%]	0	1	0	0

Gambar 5.1 merepresentasikan Tabel 5.2 dalam bentuk histogram.



**Gambar 5.1:** Histogram Metrik Evaluasi dengan Fitur MFCC dan Metode Klasifikasi SVM

Dari Gambar 5.1 dapat diamati bahwa mayoritas ayat dalam eksperimen terklasifikasi dengan rentang akurasi (60%,65%].

### 5.1.2 Hasil dengan Fitur MFCC dan Metode Klasifikasi GMM

Eksperimen menggunakan fitur MFCC dan menggunakan metode klasifikasi GMM, memberikan hasil yang dirangkum pada Tabel 5.3 berikut.

**Tabel 5.3:** Rangkuman Hasil Eksperimen dengan Fitur MFCC dan Metode Klasifikasi GMM

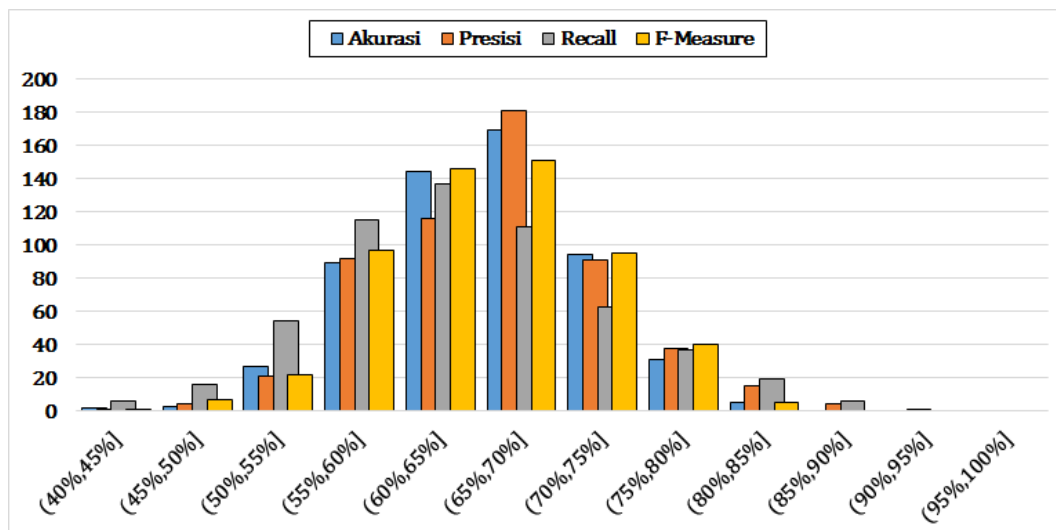
	Akurasi	Presisi	<i>Recall</i>	<i>F-Measure</i>
Minimal	43.8%	43.9%	42.5%	44.4%
Maksimal	85.0%	93.3%	90.0%	85.0%
Rata-rata	65.8%	66.3%	65.1%	65.4%
Standar Deviasi	6.50%	7.19%	8.56%	6.72%

Tabel 5.4 menunjukkan frekuensi kemunculan persentase tertentu yang dikelompokkan ke dalam interval 5%.

**Tabel 5.4:** Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur MFCC dan Metode Klasifikasi GMM

Persentase	Frekuensi Nilai Akurasi	Frekuensi Nilai Presisi	Frekuensi Nilai <i>Recall</i>	Frekuensi Nilai <i>F-Measure</i>
(40%,45%]	2	1	6	1
(45%,50%]	3	4	16	7
(50%,55%]	27	21	54	22
(55%,60%]	89	92	115	97
(60%,65%]	144	116	137	146
(65%,70%]	169	181	111	151
(70%,75%]	94	91	63	95
(75%,80%]	31	38	37	40
(80%,85%]	5	15	19	5
(85%,90%]	0	4	6	0
(90%,95%]	0	1	0	0
(95%,100%]	0	0	0	0

Gambar 5.2 merepresentasikan Tabel 5.4 dalam bentuk histogram.



**Gambar 5.2:** Histogram Metrik Evaluasi dengan Fitur MFCC dan Metode Klasifikasi GMM

Dari Gambar 5.2 dapat diamati bahwa mayoritas ayat dalam eksperimen terklasifikasi dengan rentang akurasi (65%,70%]. Hasil tersebut sedikit lebih tinggi jika dibandingkan dengan hasil pada eksperimen yang menggunakan fitur MFCC dan metode klasifikasi SVM. Hal tersebut mengindikasikan bahwa metode klasifikasi GMM lebih tepat untuk digunakan dalam sistem evaluasi pembacaan Al-Qur'an dalam penelitian ini.

### 5.1.3 Hasil dengan Fitur MFCC dan Metode Klasifikasi Gabungan

Eksperimen menggunakan fitur MFCC dengan metode klasifikasi gabungan antara SVM dan GMM, memberikan hasil yang dirangkum pada Tabel 5.5 berikut.

**Tabel 5.5:** Rangkuman Hasil Eksperimen dengan Fitur MFCC dan Metode Klasifikasi Gabungan

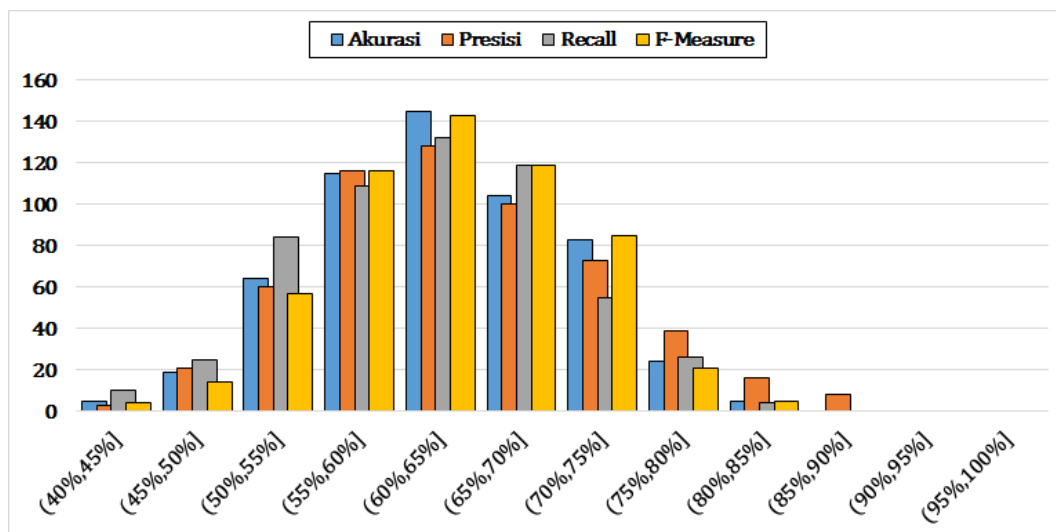
	Akurasi	Presisi	Recall	F-Measure
Minimal	41.3%	42.2%	37.5%	43.9%
Maksimal	85.0%	90.0%	85.0%	84.2%
Rata-rata	63.5%	64.1%	63.1%	63.4%
Standar Deviasi	7.64%	8.71%	8.00%	7.26%

Tabel 5.6 menunjukkan frekuensi kemunculan persentase tertentu yang dikelompokkan ke dalam interval 5%.

**Tabel 5.6:** Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur MFCC dan Metode Klasifikasi Gabungan

Persentase	Frekuensi Nilai Akurasi	Frekuensi Nilai Presisi	Frekuensi Nilai <i>Recall</i>	Frekuensi Nilai <i>F-Measure</i>
(40%,45%]	5	3	10	4
(45%,50%]	19	21	25	14
(50%,55%]	64	60	84	57
(55%,60%]	115	116	109	116
(60%,65%]	145	128	132	143
(65%,70%]	104	100	119	119
(70%,75%]	83	73	55	85
(75%,80%]	24	39	26	21
(80%,85%]	5	16	4	5
(85%,90%]	0	8	0	0
(90%,95%]	0	0	0	0
(95%,100%]	0	0	0	0

Gambar 5.3 merepresentasikan Tabel 5.6 dalam bentuk histogram.



**Gambar 5.3:** Histogram Metrik Evaluasi dengan Fitur MFCC dan Metode Klasifikasi Gabungan

Dari Gambar 5.3 dapat diamati bahwa mayoritas ayat dalam eksperimen terklasifikasi dengan rentang akurasi (60%,65%]. Hasil tersebut sedikit lebih rendah jika dibandingkan dengan hasil pada eksperimen yang menggunakan fitur MFCC dan metode klasifikasi GMM. Hal tersebut mengindikasikan bahwa gabungan dua metode klasifikasi, yaitu SVM dengan GMM, tidak lebih tepat untuk digunakan

dalam sistem evaluasi pembacaan Al-Qur'an dalam eksperimen ini jika dibandingkan dengan metode klasifikasi GMM.

## 5.2 Hasil dengan Fitur SDCC

### 5.2.1 Hasil dengan Fitur SDCC dan Metode Klasifikasi SVM

Eksperimen menggunakan fitur SDCC dan menggunakan metode klasifikasi SVM, memberikan hasil yang dirangkum pada Tabel 5.7 berikut.

**Tabel 5.7:** Rangkuman Hasil Eksperimen dengan Fitur SDCC dan Metode Klasifikasi SVM

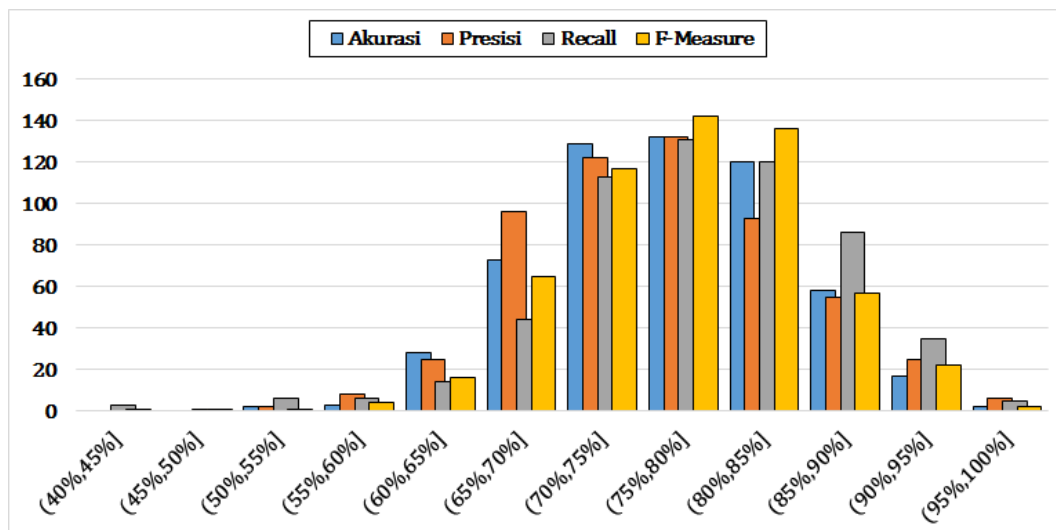
	Akurasi	Presisi	<i>Recall</i>	<i>F-Measure</i>
Minimal	51.3%	51.2%	30.0%	43.6%
Maksimal	96.3%	100.0%	100.0%	96.4%
Rata-rata	77.3%	76.6%	79.5%	77.8%
Standar Deviasi	7.47%	8.18%	8.74%	7.41%

Tabel 5.8 menunjukkan frekuensi kemunculan persentase tertentu yang dikelompokkan ke dalam interval 5%.

**Tabel 5.8:** Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur SDCC dan Metode Klasifikasi SVM

Persentase	Frekuensi Nilai Akurasi	Frekuensi Nilai Presisi	Frekuensi Nilai <i>Recall</i>	Frekuensi Nilai <i>F-Measure</i>
(40%,45%]	0	0	3	1
(45%,50%]	0	0	1	1
(50%,55%]	2	2	6	1
(55%,60%]	3	8	6	4
(60%,65%]	28	25	14	16
(65%,70%]	73	96	44	65
(70%,75%]	129	122	113	117
(75%,80%]	132	132	131	142
(80%,85%]	120	93	120	136
(85%,90%]	58	55	86	57
(90%,95%]	17	25	35	22
(95%,100%]	2	6	5	2

Gambar 5.4 merepresentasikan Tabel 5.8 dalam bentuk histogram.



**Gambar 5.4:** Histogram Metrik Evaluasi dengan Fitur SDCC dan Metode Klasifikasi SVM

Dari Gambar 5.4 dapat diamati bahwa mayoritas ayat dalam eksperimen terklasifikasi dengan rentang akurasi (75%,80%]. Hasil tersebut secara signifikan lebih tinggi jika dibandingkan dengan beberapa hasil pada eksperimen yang menggunakan fitur MFCC. Hal tersebut mengindikasikan bahwa penggunaan fitur SDCC lebih tepat daripada fitur MFCC, untuk digunakan dalam sistem evaluasi pembacaan Al-Qur'an dalam penelitian ini.

### 5.2.2 Hasil dengan Fitur SDCC dan Metode Klasifikasi GMM

Eksperimen menggunakan fitur SDCC dan menggunakan metode klasifikasi GMM, memberikan hasil yang dirangkum pada Tabel 5.9 berikut.

**Tabel 5.9:** Rangkuman Hasil Eksperimen dengan Fitur SDCC dan Metode Klasifikasi GMM

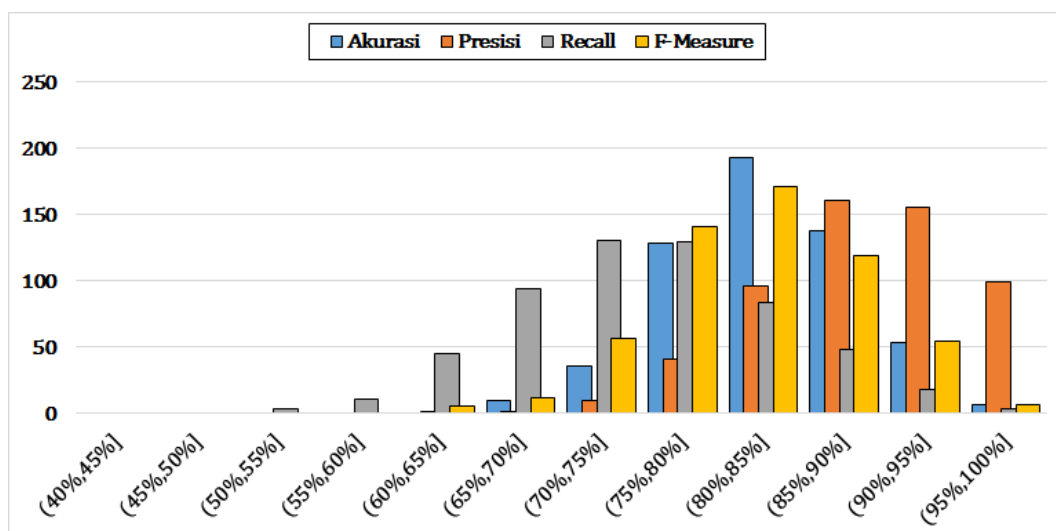
	Akurasi	Presisi	Recall	F-Measure
Minimal	67.5%	64.6%	55.0%	63.8%
Maksimal	98.8%	100.0%	100.0%	98.8%
Rata-rata	83.4%	89.0%	76.4%	82.0%
Standar Deviasi	5.61%	6.39%	8.18%	6.33%

Tabel 5.10 menunjukkan frekuensi kemunculan persentase tertentu yang dikelompokkan ke dalam interval 5%.

**Tabel 5.10:** Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur SDCC dan Metode Klasifikasi GMM

Persentase	Frekuensi Nilai Akurasi	Frekuensi Nilai Presisi	Frekuensi Nilai <i>Recall</i>	Frekuensi Nilai <i>F-Measure</i>
(40%,45%]	0	0	0	0
(45%,50%]	0	0	0	0
(50%,55%]	0	0	3	0
(55%,60%]	0	0	11	0
(60%,65%]	0	1	45	5
(65%,70%]	10	1	94	12
(70%,75%]	36	10	130	56
(75%,80%]	128	41	129	141
(80%,85%]	193	96	83	171
(85%,90%]	138	161	48	119
(90%,95%]	53	155	18	54
(95%,100%]	6	99	3	6

Gambar 5.5 merepresentasikan Tabel 5.10 dalam bentuk histogram.



**Gambar 5.5:** Histogram Metrik Evaluasi dengan Fitur SDCC dan Metode Klasifikasi GMM

Dari Gambar 5.5 dapat diamati bahwa mayoritas ayat dalam eksperimen terklasifikasi dengan rentang akurasi (80%,85%]. Hasil tersebut sedikit lebih tinggi jika dibandingkan dengan hasil pada eksperimen yang menggunakan fitur SDCC dan metode klasifikasi SVM. Hal tersebut semakin mengindikasikan bahwa metode klasifikasi GMM lebih tepat untuk digunakan dalam sistem evaluasi pembacaan Al-Qur'an dalam eksperimen ini.



### 5.2.3 Hasil dengan Fitur SDCC dan Metode Klasifikasi Gabungan

Eksperimen menggunakan fitur SDCC dan menggunakan metode klasifikasi gabungan SVM dengan GMM, memberikan hasil yang dirangkum pada Tabel 5.11 berikut.

**Tabel 5.11:** Rangkuman Hasil Eksperimen dengan Fitur SDCC dan Metode Klasifikasi Gabungan

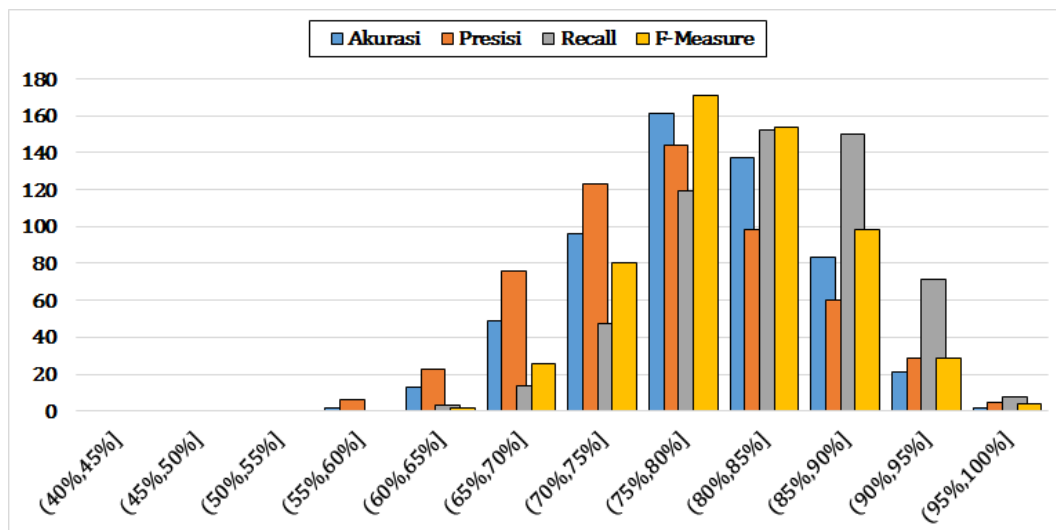
	Akurasi	Presisi	<i>Recall</i>	<i>F-Measure</i>
Minimal	57.5%	56.5%	65.0%	60.5%
Maksimal	96.3%	100.0%	100.0%	96.4%
Rata-rata	79.4%	77.3%	84.2%	80.4%
Standar Deviasi	6.86%	7.80%	6.48%	6.16%

Tabel 5.12 menunjukkan frekuensi kemunculan persentase tertentu yang dikelompokkan ke dalam interval 5%.

**Tabel 5.12:** Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur SDCC dan Metode Klasifikasi Gabungan

Persentase	Frekuensi Nilai Akurasi	Frekuensi Nilai Presisi	Frekuensi Nilai <i>Recall</i>	Frekuensi Nilai <i>F-Measure</i>
(40%,45%]	0	0	0	0
(45%,50%]	0	0	0	0
(50%,55%]	0	0	0	0
(55%,60%]	2	6	0	0
(60%,65%]	13	23	3	2
(65%,70%]	49	76	14	26
(70%,75%]	96	123	47	80
(75%,80%]	161	144	119	171
(80%,85%]	137	98	152	154
(85%,90%]	83	60	150	98
(90%,95%]	21	29	71	29
(95%,100%]	2	5	8	4

Gambar 5.6 merepresentasikan Tabel 5.12 dalam bentuk histogram.



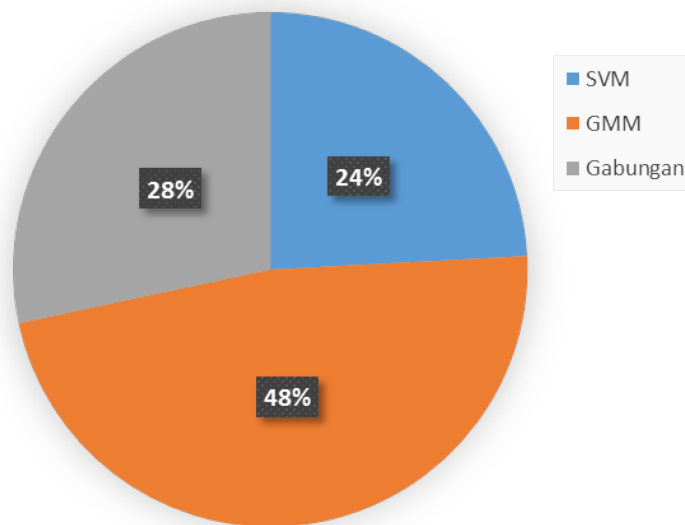
**Gambar 5.6:** Histogram Metrik Evaluasi dengan Fitur SDCC dan Metode Klasifikasi Gabungan

Dari Gambar 5.6 dapat diamati bahwa mayoritas ayat dalam eksperimen terklasifikasi dengan rentang akurasi (75%,80%]. Hasil tersebut sedikit lebih rendah jika dibandingkan dengan hasil pada eksperimen yang menggunakan fitur SDCC dan metode klasifikasi GMM. Hal tersebut semakin mengindikasikan bahwa gabungan dua metode klasifikasi, yaitu SVM dengan GMM, tidak lebih tepat untuk digunakan dalam sistem evaluasi pembacaan Al-Qur'an dalam eksperimen ini jika dibandingkan dengan metode klasifikasi GMM.

## 5.3 Perbandingan Hasil

### 5.3.1 Perbandingan Metode Klasifikasi pada Fitur MFCC

Eksperimen menggunakan fitur MFCC dengan berbagai metode klasifikasi, memberikan nilai akurasi yang berbeda-beda pada setiap ayat. Gambar 5.7 menunjukkan perbandingan metode klasifikasi yang memiliki nilai akurasi paling tinggi di antara metode-metode yang digunakan dalam eksperimen.

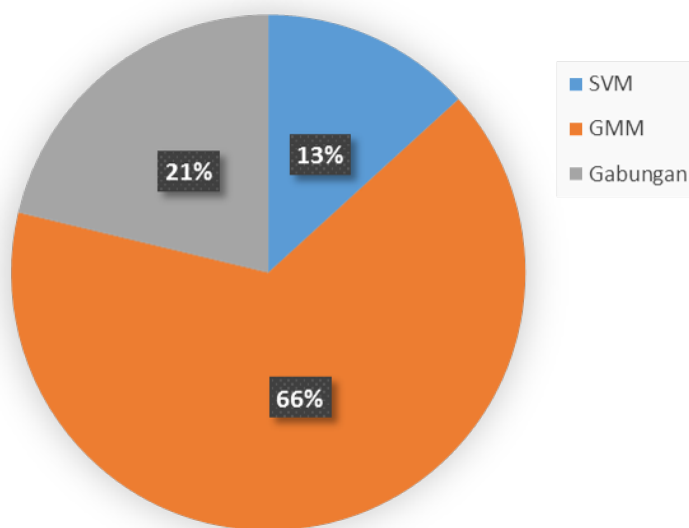


**Gambar 5.7:** Perbandingan Metode Klasifikasi untuk Fitur MFCC

Berdasarkan Gambar 5.7, terlihat bahwa pada 48% ayat dalam eksperimen, metode klasifikasi GMM menghasilkan akurasi tertinggi jika dibandingkan dengan dua metode klasifikasi lainnya. Nilai tersebut mendominasi perolehan akurasi tertinggi pada ketiga metode klasifikasi. Sehingga dapat dikatakan bahwa metode klasifikasi GMM memiliki peluang paling besar untuk menjadi metode yang menghasilkan akurasi tertinggi.

### 5.3.2 Perbandingan Metode Klasifikasi pada Fitur SDCC

Eksperimen menggunakan fitur SDCC dengan berbagai metode klasifikasi, memberikan nilai akurasi yang berbeda-beda pada setiap ayat. Gambar 5.8 menunjukkan perbandingan metode klasifikasi yang memiliki nilai akurasi paling tinggi di antara metode-metode yang digunakan dalam eksperimen.



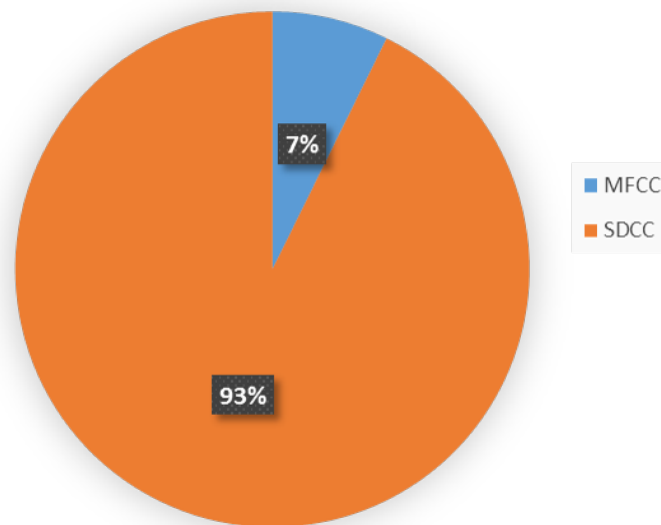
**Gambar 5.8:** Perbandingan Metode Klasifikasi untuk Fitur SDCC

Berdasarkan Gambar 5.8, terlihat bahwa pada 66% ayat dalam eksperimen, metode klasifikasi GMM menghasilkan akurasi tertinggi jika dibandingkan dengan dua metode klasifikasi lainnya. Nilai tersebut mendominasi perolehan akurasi tertinggi pada ketiga metode klasifikasi. Sehingga dapat dikatakan bahwa metode klasifikasi GMM memiliki peluang paling besar untuk menjadi metode yang menghasilkan akurasi tertinggi. Hal tersebut juga konsisten pada eksperimen yang menggunakan fitur MFCC.

Suatu pembacaan ayat Al-Qur'an dapat dinilai benar, salah, jauh dari benar, hampir benar, dan lain sebagainya. Nilai kebenaran suatu pembacaan Al-Qur'an tidak bersifat diskrit. Dalam penelitian ini metode klasifikasi GMM lebih banyak menghasilkan akurasi yang lebih tinggi daripada metode klasifikasi SVM karena klasifikasi pembacaan ayat Al-Qur'an merupakan klasifikasi yang tidak diskrit, sehingga GMM yang bersifat generatif dapat memodelkan data dengan lebih baik daripada SVM yang bersifat diskriminatif.

### 5.3.3 Perbandingan Fitur pada Metode Klasifikasi SVM

Eksperimen menggunakan metode klasifikasi SVM dengan berbagai fitur, memberikan nilai akurasi yang berbeda-beda pada setiap ayat. Gambar 5.9 menunjukkan perbandingan fitur yang memiliki nilai akurasi paling tinggi di antara fitur-fitur yang digunakan dalam eksperimen.

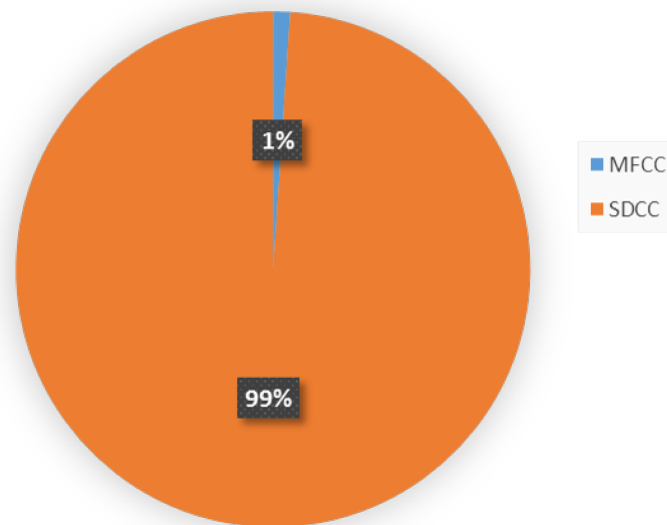


**Gambar 5.9:** Perbandingan Fitur untuk Metode Klasifikasi SVM

Berdasarkan Gambar 5.9, terlihat bahwa pada 93% ayat dalam eksperimen, fitur SDCC menghasilkan akurasi yang lebih tinggi jika dibandingkan dengan fitur MFCC. Nilai tersebut mendominasi perolehan akurasi tertinggi pada kedua fitur. Sehingga dapat dikatakan bahwa fitur SDCC memiliki peluang paling besar untuk menjadi fitur yang menghasilkan akurasi tertinggi.

#### **5.3.4 Perbandingan Fitur pada Metode Klasifikasi GMM**

Eksperimen menggunakan metode klasifikasi GMM dengan berbagai fitur, memberikan nilai akurasi yang berbeda-beda pada setiap ayat. Gambar 5.10 menunjukkan perbandingan fitur yang memiliki nilai akurasi paling tinggi di antara fitur-fitur yang digunakan dalam eksperimen.

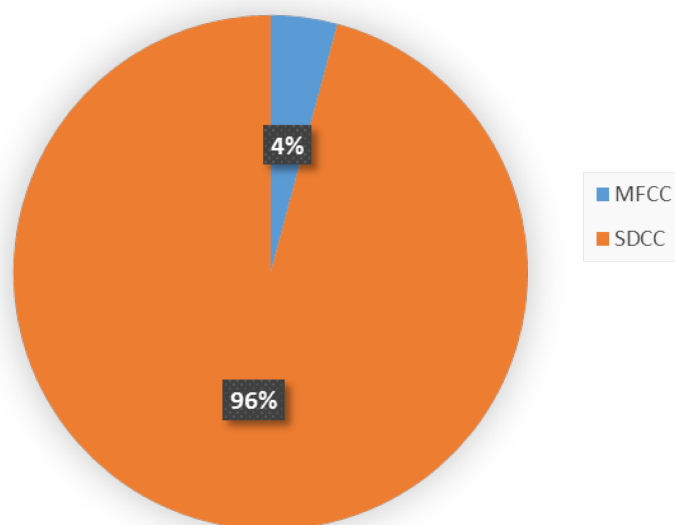


**Gambar 5.10:** Perbandingan Fitur untuk Metode Klasifikasi GMM

Berdasarkan Gambar 5.10, terlihat bahwa pada 99% ayat dalam eksperimen, fitur SDCC menghasilkan akurasi yang lebih tinggi jika dibandingkan dengan fitur MFCC. Nilai tersebut mendominasi perolehan akurasi tertinggi pada kedua fitur. Hal tersebut memperkuat pernyataan bahwa fitur SDCC memiliki peluang paling besar untuk menjadi fitur yang menghasilkan akurasi tertinggi dibandingkan fitur MFCC.

### 5.3.5 Perbandingan Fitur pada Metode Klasifikasi Gabungan

Eksperimen menggunakan metode klasifikasi Gabungan dengan berbagai fitur, memberikan nilai akurasi yang berbeda-beda pada setiap ayat. Gambar 5.11 menunjukkan perbandingan fitur yang memiliki nilai akurasi paling tinggi di antara fitur-fitur yang digunakan dalam eksperimen.



**Gambar 5.11:** Perbandingan Fitur untuk Metode Klasifikasi Gabungan

Berdasarkan Gambar 5.11, terlihat bahwa pada 96% ayat dalam eksperimen, fitur SDCC menghasilkan akurasi yang lebih tinggi jika dibandingkan dengan fitur MFCC. Nilai tersebut mendominasi perolehan akurasi tertinggi pada kedua fitur. Hal tersebut konsisten dengan hasil pada metode klasifikasi SVM maupun GMM, dan semakin memperkuat pernyataan bahwa fitur SDCC memiliki peluang paling besar untuk menjadi fitur yang menghasilkan akurasi tertinggi dibandingkan fitur MFCC.

Nilai SDCC pada setiap *frame* merupakan kombinasi dari beberapa nilai MFCC yang berdekatan. Dalam penelitian ini penggunaan fitur SDCC lebih banyak akurat daripada penggunaan fitur MFCC karena SDCC memuat lebih banyak konteks dalam setiap *frame*-nya jika dibandingkan dengan MFCC.

## 5.4 Analisis Lanjut

Penggunaan fitur SDCC dan metode klasifikasi GMM pada eksperimen memberikan hasil terbaik secara rata-rata, baik pada nilai akurasi, presisi, *recall*, ataupun *f-measure*. Setiap ayat memiliki hasil klasifikasi yang berbeda-beda. Ada ayat-ayat yang diklasifikasikan dengan akurasi tinggi dan ada juga ayat-ayat yang diklasifikasikan dengan akurasi rendah. Beberapa ayat dengan akurasi tertinggi yang lebih dari atau sama dengan 95% pada penggunaan fitur SDCC dan metode klasifikasi GMM dapat dilihat pada Tabel 5.13.

**Tabel 5.13:** Ayat-Ayat yang Diurutkan dari Akurasi Tertinggi

Surat	Ayat	Akurasi	Presisi	<i>Recall</i>	<i>F-Measure</i>
89	22	98.8%	97.6%	100.0%	98.8%
91	13	97.5%	100.0%	95.0%	97.4%
92	19	97.5%	97.5%	97.5%	97.5%
87	7	96.3%	95.1%	97.5%	96.3%
93	7	96.3%	97.4%	95.0%	96.2%
110	3	96.3%	100.0%	92.5%	96.1%
78	19	95.0%	95.0%	95.0%	95.0%
79	31	95.0%	97.4%	92.5%	94.9%
79	46	95.0%	100.0%	90.0%	94.7%
80	2	95.0%	100.0%	90.0%	94.7%
80	25	95.0%	95.0%	95.0%	95.0%
88	21	95.0%	100.0%	90.0%	94.7%
90	4	95.0%	100.0%	90.0%	94.7%

Beberapa ayat dengan akurasi terendah yang kurang dari atau sama dengan 70% pada penggunaan fitur SDCC dan metode klasifikasi GMM dapat dilihat pada Tabel 5.14.

**Tabel 5.14:** Ayat-Ayat yang Diurutkan dari Akurasi Terendah

Surat	Ayat	Akurasi	Presisi	<i>Recall</i>	<i>F-Measure</i>
78	4	67.5%	71.9%	57.5%	63.9%
83	4	67.5%	64.6%	77.5%	70.5%
85	21	67.5%	69.4%	62.5%	65.8%
101	10	67.5%	71.9%	57.5%	63.9%
80	24	68.8%	75.9%	55.0%	63.8%
106	1	68.8%	75.9%	55.0%	63.8%
106	3	68.8%	74.2%	57.5%	64.8%
106	4	68.8%	71.4%	62.5%	66.7%
82	12	70.0%	73.5%	62.5%	67.6%
85	6	70.0%	76.7%	57.5%	65.7%

Suatu ayat dapat diklasifikasikan dengan akurasi tinggi dikarenakan ayat tersebut memiliki ciri khusus dalam pelafalannya jika dibandingkan dengan ayat-ayat lainnya. Ciri tersebut antara lain adalah bacaan *mad wajib* (bacaan panjang 3 huruf), *ghunnah* (bacaan dengung), serta irama panjang pendek dalam pembacaan



ayat. Contoh beberapa ayat yang dapat diklasifikasikan dengan akurasi tinggi antara lain sebagai berikut.

- Surat ke-89 ayat 22, yaitu “وَجَاءَ رَبُّكَ وَالْمَلَكُ صَفًّا صَفًّا”. Ayat tersebut mengandung *mad wajib* pada kata وَجَاءَ serta *ghunnah* pada dua kata صَفًّا صَفًّا.
- Surat ke-92 ayat 19, yaitu “وَمَا لِأَحَدٍ عِنْدَهُ مِنْ نِعْمَةٍ تُجْزَىٰ”. Ayat tersebut mengandung *ghunnah* pada kata مِنْ نِعْمَةٍ.
- Surat ke-110 ayat 3, yaitu “فَسَبِّحْ بِحَمْدِ رَبِّكَ وَاسْتَغْفِرْهُ إِنَّهُ كَانَ تَوَّابًا”. Ayat tersebut mengandung *ghunnah* pada kata إِنَّهُ dan تَوَّابًا.

## **BAB 6**

### **KESIMPULAN DAN SARAN**

#### **6.1 Kesimpulan**

Setelah mengimplementasikan rancangan arsitektur sistem, menjalankan eksperimen, serta menganalisis hasil eksperimen, diperoleh kesimpulan sebagai berikut.

1. Penggunaan fitur SDCC berpeluang lebih baik daripada fitur MFCC. Hal ini dapat diamati dari Gambar 5.9, Gambar 5.10, dan Gambar 5.11. Ketiga gambar tersebut menunjukkan hasil yang konsisten bahwa fitur SDCC lebih mendominasi dalam memberikan hasil yang lebih akurat daripada fitur MFCC.
2. Penggunaan metode klasifikasi GMM berpeluang lebih baik daripada metode klasifikasi SVM maupun metode gabungan SVM dengan GMM. Hal ini dapat diamati dari 5.7 dan Gambar 5.8. Kedua gambar tersebut menunjukkan bahwa metode klasifikasi GMM lebih mendominasi dalam memberikan hasil yang paling akurat dibandingkan metode klasifikasi lainnya.
3. Kombinasi pengambilan fitur SDCC dengan metode klasifikasi GMM adalah kombinasi yang memberikan hasil paling akurat secara rata-rata dalam penelitian ini. Hal ini dapat diamati dari nilai rata-rata akurasi pada Tabel 5.1, Tabel 5.3, Tabel 5.5, Tabel 5.7, Tabel 5.9, dan Tabel 5.11. Kombinasi fitur SDCC dengan metode klasifikasi GMM menghasilkan nilai rata-rata akurasi tertinggi, yaitu sebesar 83,4%.
4. Penggabungan dua metode klasifikasi yang sama-sama memiliki kinerja yang baik tidak menjamin akan menghasilkan metode klasifikasi baru yang lebih akurat secara rata-rata. Nilai rata-rata akurasi dari metode klasifikasi GMM turun setelah digabung dengan metode klasifikasi SVM, baik pada pengambilan fitur MFCC maupun pada pengambilan fitur SDCC.
5. Akurasi dari masing-masing ayat berbeda-beda. Ada beberapa ayat yang dapat diklasifikasikan dengan akurasi tinggi, dan ada juga beberapa ayat yang hanya diklasifikasikan dengan akurasi rendah.

## 6.2 Saran

Setelah melakukan eksperimen dan menganalisis hasilnya, ada beberapa saran untuk penelitian selanjutnya, antara lain sebagai berikut.

1. Fitur MFCC dan fitur SDCC memungkinkan untuk dikombinasikan. Kombinasi tersebut layak untuk dicoba dalam penelitian selanjutnya karena ada peluang akurasi dari sistem akan meningkat.
2. Sistem dalam penelitian ini dapat dikembangkan lebih lanjut dengan cara mensegmentasi ayat-ayat yang panjang, seperti yang ada dalam Surat Al-Baqarah. Dalam surat tersebut terdapat ayat sepanjang satu halaman Al-Qur'an. Dengan cara disegmentasi, ayat-ayat yang panjang dapat diperlakukan seperti ayat-ayat yang pendek di juz 30. Sehingga sistem yang dikembangkan dalam penelitian ini dapat digunakan untuk seluruh ayat dalam Al-Qur'an.
3. Banyaknya data sampel dapat ditingkatkan, tidak hanya 40 sampel qari. Data sampel yang semakin banyak diharapkan akan menghasilkan sistem yang semakin akurat dan presisi.
4. Masih ada metode-metode klasifikasi lain yang layak untuk dicoba dalam penelitian selanjutnya, seperti *deep learning*, *i-vector*, dan *Gaussian process*. Ada kemungkinan metode klasifikasi lain akan menghasilkan sistem yang lebih akurat dan presisi.
5. Metode klasifikasi yang dilakukan dalam penelitian ini masih menggunakan parameter yang konstan. Ada potensi untuk meningkatkan akurasi sistem dengan memilih parameter yang lebih tepat pada setiap metode klasifikasi.
6. Penelitian ini dapat dikembangkan untuk mencari tahu qari mana yang pelafalannya paling sering diidentifikasi dengan benar.

## DAFTAR REFERENSI

- Abacha, A. B. dan Zweigenbaum, P. (2011). Medical entity recognition: A comparison of semantic and statistical methods. In *Proceedings of BioNLP 2011 Workshop*, pages 56–64. Association for Computational Linguistics.
- Anton, H. dan Rorres, C. (2010). *Elementary Linear Algebra: Applications Version*. John Wiley & Sons.
- Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York.
- Boswell, D. (2016). Introduction to support vector machines. <http://dustwell.com/PastWork/IntroToSVM.pdf>. [Diakses 7 Juni 2016].
- Campbell, W. M., Campbell, J. P., Reynolds, D. A., Singer, E., dan Torres-Carrasquillo, P. A. (2006). Support vector machines for speaker and language recognition. *Computer Speech & Language*, 20(2):210–229.
- Chigier, B. (1997). Automatic speech recognition. US Patent 5,638,487.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., dan Stein, C. (2009). *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition.
- Davis, S. dan Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366.
- Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.
- Forsberg, M. (2003). Why is speech recognition difficult? *Chalmers University of Technology*.
- Ganchev, T., Fakotakis, N., dan Kokkinakis, G. (2005). Comparative evaluation of various mfcc implementations on the speaker verification task. *Proceedings of the SPECOM*, 1:191–194.
- Graves, A., Mohamed, A.-r., dan Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.

- Haykin, S. S., Haykin, S. S., Haykin, S. S., dan Haykin, S. S. (2009). *Neural networks and learning machines*, volume 3. Pearson Upper Saddle River, NJ, USA:.
- Herwando, R. (2016). Pengenalan entitas kesehatan pada forum kesehatan *Online* berbahasa indonesia menggunakan algoritma textitConditional Random Fields. An optional note.
- Juang, B.-H. dan Rabiner, L. R. (2005). Automatic speech recognition—a brief history of the technology development. *Encyclopedia of Language and Linguistics*.
- Jurafsky, D. dan Martin, J. H. (2009). *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., dan Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):881–892.
- Lyons, J. (2016). Mel frequency cepstral coefficient (mfcc) tutorial. <http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/>. [Diakses 12 Juni 2016].
- Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., dan Khudanpur, S. (2010). Recurrent neural network based language model. In *Interspeech*, volume 2, page 3.
- Mohammed, A., Sunar, M. S., dan Salam, M. S. H. (2015). Quranic verses verification using speech recognition techniques. *Jurnal Teknologi*, 73(2).
- Montgomery, D. (2013). *Applied Statistics and Probability for Engineers, 6th Edition*. John Wiley & Sons.
- Muhammad, W. M., Muhammad, R., Muhammad, A., dan Martinez-Enriquez, A. M. (2010). Voice content matching system for quran readers. In *Proceedings of the 2010 Ninth Mexican International Conference on Artificial Intelligence, MICAI '10*, pages 148–153, Washington, DC, USA. IEEE Computer Society.
- Putra, B., Atmaja, B., dan Prananto, D. (2012). Developing speech recognition system for quranic verse recitation learning software. *International Journal on Informatics for Development*, 1(2).

- Rabiner, L. dan Schafer, R. (1978). *Digital Processing of Speech Signals*. Prentice-Hall signal processing series. Prentice-Hall.
- Řehůřek, R. dan Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. <http://is.muni.cz/publication/884893/en>.
- Reynolds, D. A., Quatieri, T. F., dan Dunn, R. B. (2000). Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1):19–41.
- Saha, G., Chakroborty, S., dan Senapati, S. (2005). A new silence removal and endpoint detection algorithm for speech and speaker recognition applications. *Proceedings of the 11th National Conference on Communications (NCC)*, pages 291–295.
- Salmond, D. J. (2009). Mixture reduction algorithms for point and extended object tracking in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 45(2):667–686.
- Sharma, M. dan Sarma, K. K. (2016). *Soft-Computational Techniques and Spectro-Temporal Features for Telephonic Speech Recognition: An Overview and Review of Current State of the Art*. IGI Global.
- Singh, R., Pal, B. C., dan Jabr, R. A. (2010). Statistical representation of distribution system loads using gaussian mixture model. *Power Systems, IEEE Transactions on*, 25(1):29–37.
- Sirisena, H. R. dan Brown, E. P. M. (1983). Representation of non-gaussian probability distributions in stochastic load-flow studies by the method of gaussian sum approximations. *IEE Proceedings C - Generation, Transmission and Distribution*, 130(4):165–171.
- Torres-Carrasquillo, P. A., Singer, E., Kohler, M. A., Greene, R. J., Reynolds, D. A., dan Deller Jr, J. R. (2002). Approaches to language identification using gaussian mixture models and shifted delta cepstral features. In *INTERSPEECH*.
- Valverde, G., Saric, A. T., dan Terzija, V. (2012). Probabilistic load flow with non-gaussian correlated random variables using gaussian mixture models. *IET Generation, Transmission Distribution*, 6(7):701–709.

- Weisstein, E. W. (2016). K-means clustering algorithm. MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/K-MeansClusteringAlgorithm.html>. [Diakses 7 Juni 2016].
- Wyse, N., Dubes, R., dan Jain, A. K. (1980). A critical evaluation of intrinsic dimensionality algorithms. *Pattern recognition in practice*, pages 415–425.
- Yin, B., Ambikairajah, E., dan Chen, F. (2006). Combining cepstral and prosodic features in language identification. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, pages 254–257. IEEE.
- Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Valtchev, V., dan Woodland, P. (2002). The htk book. *Cambridge University Engineering Department*, 3.
- Zahra, A. dan Carson-Berndsen, J. (2013). Unique n-phone ranking based spoken language identification. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2013 Fifth International Conference on*, pages 239–244. IEEE.