



UNIVERSITAS INDONESIA

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

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 Networks**

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 Networks

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 Networks

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 Networks

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 Networks

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

HALAMAN JUDUL	i
LEMBAR PERNYATAAN ORISINALITAS	ii
LEMBAR PENGESAHAN	iii
KATA PENGANTAR	iv
LEMBAR PERSETUJUAN PUBLIKASI ILMIAH	vi
ABSTRAK	vii
Daftar Isi	ix
Daftar Gambar	xi
Daftar Tabel	xii
1 PENDAHULUAN	1
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	4
1.6 Sistematika Penulisan	5
2 LANDASAN TEORI	7
2.1 Pengenalan Entitas Kesehatan	7
2.2 Deep Learning	9
2.3 Recurrent Neural Networks	9
2.3.1 Long Short Term Memory	11
2.3.2 Penerapan RNNs untuk MER	13
2.4 Word Embedding	13
3 METODOLOGI	14
3.1 Rancangan Sistem	14
3.1.1 Pengumpulan Data	14
3.1.2 Pra-Pemrosesan	14
3.1.3 Pelabelan	15
3.1.4 Pengembangan Model	16
3.1.4.1 Ekstraksi Fitur	16
3.1.4.2 Pengusulan Arsitektur RNNs	18

3.1.5	Eksperimen	19
3.1.6	Evaluasi	20
4	EKSPERIMEN	23
4.1	Normalisasi Data	23
4.2	Ekstraksi Fitur	24
4.2.1	Ekstraksi Fitur MFCC	25
4.2.2	Ekstraksi Fitur SDCC	27
4.3	Pemodelan	29
4.3.1	Pemodelan dengan Support Vector Machine (SVM)	30
4.3.2	Pemodelan dengan Gaussian Mixture Model (GMM)	31
4.3.3	Pemodelan dengan Gabungan SVM dan GMM	31
4.4	Pengujian	32
5	HASIL EKSPERIMEN DAN ANALISIS	34
5.1	Hasil dengan Fitur MFCC	34
5.1.1	Hasil dengan Fitur MFCC dan Metode Klasifikasi SVM	34
5.1.2	Hasil dengan Fitur MFCC dan Metode Klasifikasi GMM	36
5.1.3	Hasil dengan Fitur MFCC dan Metode Klasifikasi Gabungan	37
5.2	Hasil dengan Fitur SDCC	39
5.2.1	Hasil dengan Fitur SDCC dan Metode Klasifikasi SVM	39
5.2.2	Hasil dengan Fitur SDCC dan Metode Klasifikasi GMM	40
5.2.3	Hasil dengan Fitur SDCC dan Metode Klasifikasi Gabungan	42
5.3	Perbandingan Hasil	43
5.3.1	Perbandingan Metode Klasifikasi pada Fitur MFCC	43
5.3.2	Perbandingan Metode Klasifikasi pada Fitur SDCC	44
5.3.3	Perbandingan Fitur pada Metode Klasifikasi SVM	45
5.3.4	Perbandingan Fitur pada Metode Klasifikasi GMM	46
5.3.5	Perbandingan Fitur pada Metode Klasifikasi Gabungan	47
5.4	Analisis Lanjut	48
6	KESIMPULAN DAN SARAN	51
6.1	Kesimpulan	51
6.2	Saran	52
	Daftar Referensi	53

DAFTAR GAMBAR

2.1	Ilustrasi Sistem MER	7
2.2	<i>Recurrent Neural Networks</i> sederhana	10
2.3	1 buah <i>timestep</i> dalam RNNs	11
2.4	1 buah blok memori dalam LSTM	12
3.1	Rancangan Arsitektur Sistem	14
3.2	LSTM 1 layer	19
4.1	Distribusi <i>Gaussian</i> Terhadap Nilai Jarak <i>Mahalanobis</i>	24
4.2	Alur Ekstraksi Fitur MFCC	27
4.3	Alur Ekstraksi Fitur SDCC	29
5.1	Histogram Metrik Evaluasi dengan Fitur MFCC dan Metode Klasifikasi SVM	35
5.2	Histogram Metrik Evaluasi dengan Fitur MFCC dan Metode Klasifikasi GMM	37
5.3	Histogram Metrik Evaluasi dengan Fitur MFCC dan Metode Klasifikasi Gabungan	38
5.4	Histogram Metrik Evaluasi dengan Fitur SDCC dan Metode Klasifikasi SVM	40
5.5	Histogram Metrik Evaluasi dengan Fitur SDCC dan Metode Klasifikasi GMM	41
5.6	Histogram Metrik Evaluasi dengan Fitur SDCC dan Metode Klasifikasi Gabungan	43
5.7	Perbandingan Metode Klasifikasi untuk Fitur MFCC	44
5.8	Perbandingan Metode Klasifikasi untuk Fitur SDCC	45
5.9	Perbandingan Fitur untuk Metode Klasifikasi SVM	46
5.10	Perbandingan Fitur untuk Metode Klasifikasi GMM	47
5.11	Perbandingan Fitur untuk Metode Klasifikasi Gabungan	48

DAFTAR TABEL

4.1	Parameter Pemanggilan Fungsi <i>mfcc</i>	25
4.2	Parameter Pemanggilan Fungsi <i>mfcc2sdc</i>	28
4.3	Parameter SVM dalam Eksperimen	30
4.4	Parameter GMM dalam Eksperimen	31
5.1	Rangkuman Hasil Eksperimen dengan Fitur MFCC dan Metode Klasifikasi SVM	34
5.2	Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur MFCC dan Metode Klasifikasi SVM	35
5.3	Rangkuman Hasil Eksperimen dengan Fitur MFCC dan Metode Klasifikasi GMM	36
5.4	Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur MFCC dan Metode Klasifikasi GMM	36
5.5	Rangkuman Hasil Eksperimen dengan Fitur MFCC dan Metode Klasifikasi Gabungan	37
5.6	Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur MFCC dan Metode Klasifikasi Gabungan	38
5.7	Rangkuman Hasil Eksperimen dengan Fitur SDCC dan Metode Klasifikasi SVM	39
5.8	Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur SDCC dan Metode Klasifikasi SVM	39
5.9	Rangkuman Hasil Eksperimen dengan Fitur SDCC dan Metode Klasifikasi GMM	40
5.10	Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur SDCC dan Metode Klasifikasi GMM	41
5.11	Rangkuman Hasil Eksperimen dengan Fitur SDCC dan Metode Klasifikasi Gabungan	42
5.12	Frekuensi Kemunculan Interval Persentase pada Eksperimen dengan Fitur SDCC dan Metode Klasifikasi Gabungan	42
5.13	Ayat-Ayat yang Diurutkan dari Akurasi Tertinggi	49
5.14	Ayat-Ayat yang Diurutkan dari Akurasi Terendah	49

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 struktural. 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 MER. 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 MER dilakukan oleh Abacha dan Zweigenbaum (2011). Penelitian tersebut menggunakan dokumen medis rumah sakit berbahasa Inggris.

Saat ini, pengembangan sistem MER pada dokumen berbahasa Indonesia masih belum banyak. Ada beberapa penelitian terkait sistem MER, namun hasil yang

diberikan belum memuaskan. Salah satu penelitian terkait MER dilakukan oleh Herwando (2016). MER 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 MER. Penulis mengusulkan sebuah sistem MER dengan menggunakan *Recurrent Neural Network*. Berdasarkan penelitian terkait MER 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. Sistem MER yang dihasilkan dapat digunakan untuk membuat aplikasi lain. Misalnya dengan adanya MER 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 MER 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 MER 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 MER 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 MER 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 MER 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 MER, *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 (*symptom*), 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 METODOLOGI**

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**

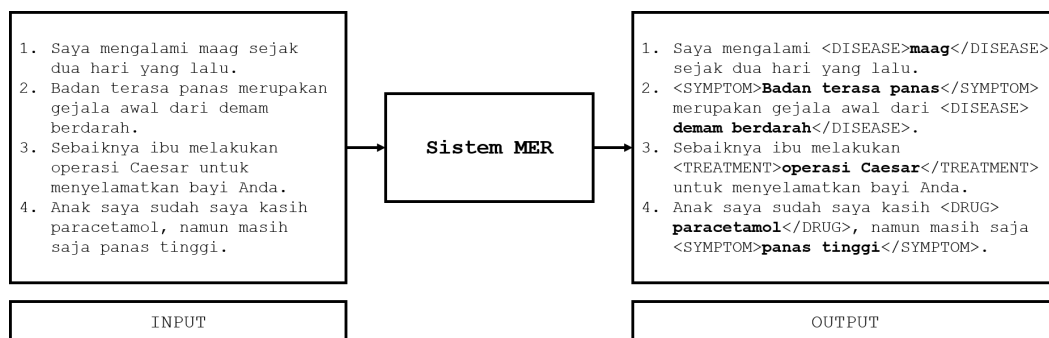
Pada bab ini penulis memberikan kesimpulan berdasarkan hasil eksperimen dan analisis yang telah dilakukan pada penelitian ini. Selain itu penulis juga memberikan saran dan masukan untuk penelitian dan pengembangan sistem mengenai MER berbahasa Indonesia selanjutnya.

BAB 2

LANDASAN TEORI

2.1 Pengenalan Entitas Kesehatan

Pengenalan Entitas Kesehatan (MER) merupakan salah satu cabang dari Pengenalan Entitas Bernama (NER) dengan dokumen sumber berupa teks kesehatan. NER sendiri merupakan suatu sistem/aplikasi yang memanfaatkan teknik pada *Natural Language Processing* dan *Information Extraction* untuk mengenali entitas yang telah dikategorikan sebelumnya seperti nama, lokasi, organisasi, waktu dll. Sedangkan pada sistem MER, entitas yang akan dikenali yaitu nama penyakit (*disease*), gejala penyakit (*symptom*), obat (*drug*) dan langkah penyembuhan *treatment*, nama protein, DNA, RNA dll. Gambar 2.1 merupakan ilustrasi dari sebuah sistem MER.



Gambar 2.1: Ilustrasi Sistem MER

Dari ilustrasi di atas, sebuah sistem MER akan diberikan *input* berupa dokumen kesehatan, kemudian sistem diharapkan dapat memberikan *output* berupa dokumen yang mendapatkan pelabelan dengan benar. Dokumen kesehatan yang menjadi *input* dapat berupa dokumen formal seperti dokumen suatu rumah sakit atau dokumen non-formal seperti dokumen forum kesehatan *online*.

Implementasi sistem MER dapat memberikan manfaat pada beberapa bidang, seperti pada aplikasi *Question Answering* (Abacha dan Zweigenbaum, 2011) yang hasil pelabelan dari sistem MER dapat mempermudah identifikasi entitas yang ditanyakan. Selain itu, hasil pelabelan sistem MER juga dapat dimanfaatkan untuk pembuatan sistem *indexing* dokumen forum sehingga pencarian dokumen kesehatan dapat dilakukan dengan lebih efisien. Sistem MER juga dapat digunakan untuk mendukung aplikasi *entity linking* yang memungkinkan seseorang untuk

mengetahui hubungan antar entitas (Hachey et al., 2013). Misalnya dengan adanya aplikasi *entity linking*, kita dapat mengetahui obat apabila hanya diberikan *query* nama penyakit dengan berdasarkan dokumen-dokumen kesehatan yang telah mendapatkan pelabelan dari sistem MER. Masih banyak manfaat lain dari implementasi sistem MER ini.

Sebelumnya Abacha dan Zweigenbaum (2011) telah melakukan penelitian terkait sistem MER pada dokumen berbahasa Inggris. Sistem MER yang dibuat bertujuan untuk melabeli entitas *treatment*, *problem* dan *test* dengan menggunakan 3 metode, yaitu (i) metode semantik dengan menggunakan *tools* MetaMap (*domain knowledge*), (ii) ekstraksi frasa berdasarkan *chunker* dan klasifikasi dengan SVM (*machine learning based*) dan (iii) gabungan 2 metode sebelumnya dengan menggunakan CRF (*hybrid*). Hasil yang terbaik didapatkan dengan menggunakan metode *hybrid* yang menggabungkan 2 metode sebelumnya (*domain knowledge* dan *machine learning*) dan dengan *precision* 72.18%, *recall* 83.78% dan *f-measures* 77.55%.

Selain penelitian di atas, Mujiono et al. (2016) juga melakukan penelitian terkait MER dengan tujuan untuk mendapatkan representasi data yang berdasarkan karakteristik *training data*. Mujiono et al. (2016) mengusulkan tiga teknik representasi data, yaitu (i) evaluasi dengan model *neural networks* standar, (ii) evaluasi dengan dua *deep network classifiers*, yaitu DBN (*Deep Belief Networks*), dan SAE (*Stacked Denoising Encoders*) serta (iii) representasi kalimat sebagai *sequence* yang dievaluasi dengan *recurrent neural networks* yaitu LSTM (*Long Short Term Memory*). Hasil yang didapatkan yaitu kalimat sebagai *sequence* yang dievaluasi dengan LSTM memberikan hasil yang terbaik, yaitu *f-measure* 86.45%.

Penelitian terkait MER pada dokumen berbahasa Indonesia sudah dilakukan sebelumnya oleh Herwando (2016). Dalam penelitiannya, Herwando (2016) menggunakan CRF (*Conditional Random Fields*) dengan tujuan untuk mendapatkan kombinasi fitur yang menghasilkan akurasi terbaik. Entitas yang akan diberi label yaitu nama penyakit (*disease*), gejala penyakit (*symptom*), obat (*drug*) dan langkah penyembuhan *treatment*. Dokumen yang menjadi input penelitian merupakan hasil *crawling* dari forum kesehatan *online* dari berbagai situs yang berisi tanya jawab. Hasil yang didapatkan yaitu *precision* 70.97%, *recall* 57.83% dan *f-measure* 63.69% dengan fitur *its own word*, frasa, kamus (*symptom*, *disease*, *treatment* dan *drug*), kata pertama sebelum dan panjang kata.

Selain itu, Suwarningsih et al. (2014) juga melakukan penelitian terkait MER pada dokumen berbahasa Indonesia dengan menggunakan SVM (*Support Vector Machine*). Entitas yang akan dikenali yaitu *location*, *facility*, *diagnosis*,

definition dan *person*. Data yang digunakan sebagai korpus merupakan data dari situs *health.detik.com*, *detikhealth.com* dan *health.kompas.com/konsultasi* dengan total keseluruhan sebanyak 1000 kalimat. AKurasi yang dihasilkan yaitu 90% dengan menggunakan fitur *baseline*, *word level (morphology, POS-Tag, dll)* dan fitur dari dalam dokumen tersebut.

2.2 Deep Learning

–Belum

2.3 Recurrent Neural Networks

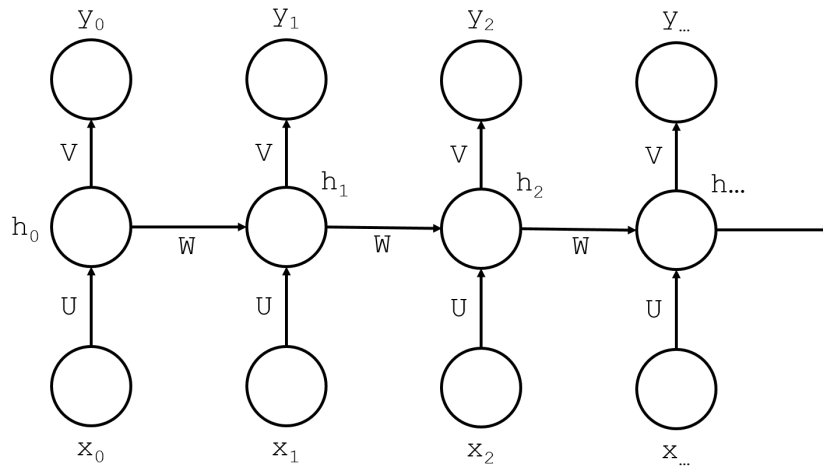
Recurrent neural networks (RNNs) merupakan *Artificial Neural Networks* (ANNs) yang memiliki koneksi siklik (Graves, 2012). RNNs memiliki *neuron* yang terkoneksi dengan *neuron* lain sehingga membentuk *loop* umpan balik (Haykin et al. (2009)), tidak seperti *feedforward neural network* (FNNs) dimana 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 posisi invarian sebuah kata dapat ditangani. Hal ini tidak dapat dilakukan

pada FNNs,

Banyak variasi RNNs yang telah diusulkan oleh beberapa peneliti, seperti *Elman networks* (Elman, 1990), *Jordan networks* (Jordan, 1986), *time delay neural networks* (Lang et al., 1990) dll. Gambar berikut merupakan conroh dari RNNs sederhana



Gambar 2.2: *Recurrent Neural Networks* sederhana

Dari gambar 2.2, sebuah jaringan pada RNNs memiliki *input layer* x , *hidden layer* h dan *output layer* y . Terdapat 3 buah parameter yang dicari dalam tahap *learning*, yaitu U, V, W yang masing-masing parameter tersebut di-*share* untuk semua *timestep* t dengan:

$$U, V, W \in \mathbb{R} \quad (2.1)$$

Untuk suatu *timestep* t , *input* RNNs dinotasikan sebagai $x(t)$, *timestep* dinotasikan sebagai $h(t)$ dan *output* dinotasikan sebagai $y(t)$, dengan:

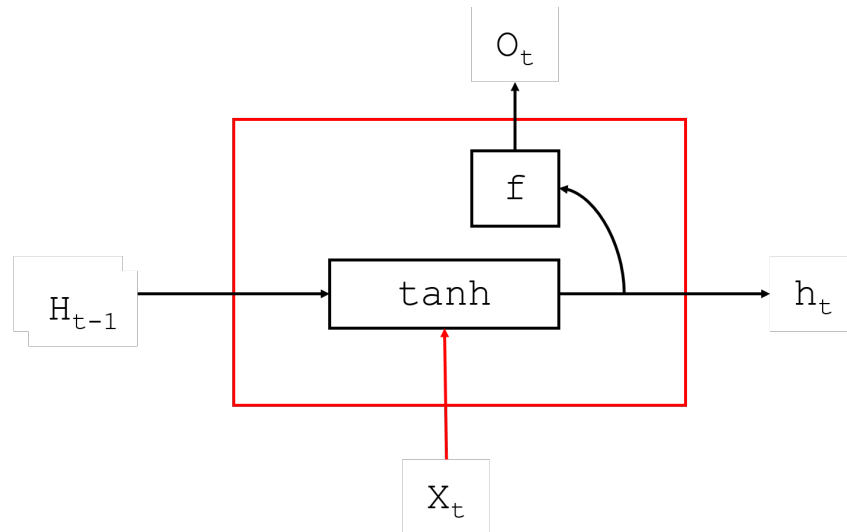
$$y(t) = f(V \cdot h(t)) \quad (2.2)$$

$$h(t) = f(U \cdot x(t) + W \cdot h(t-1)) \quad (2.3)$$

dimana

$$h(0) = f(W \cdot x(0)) \quad (2.4)$$

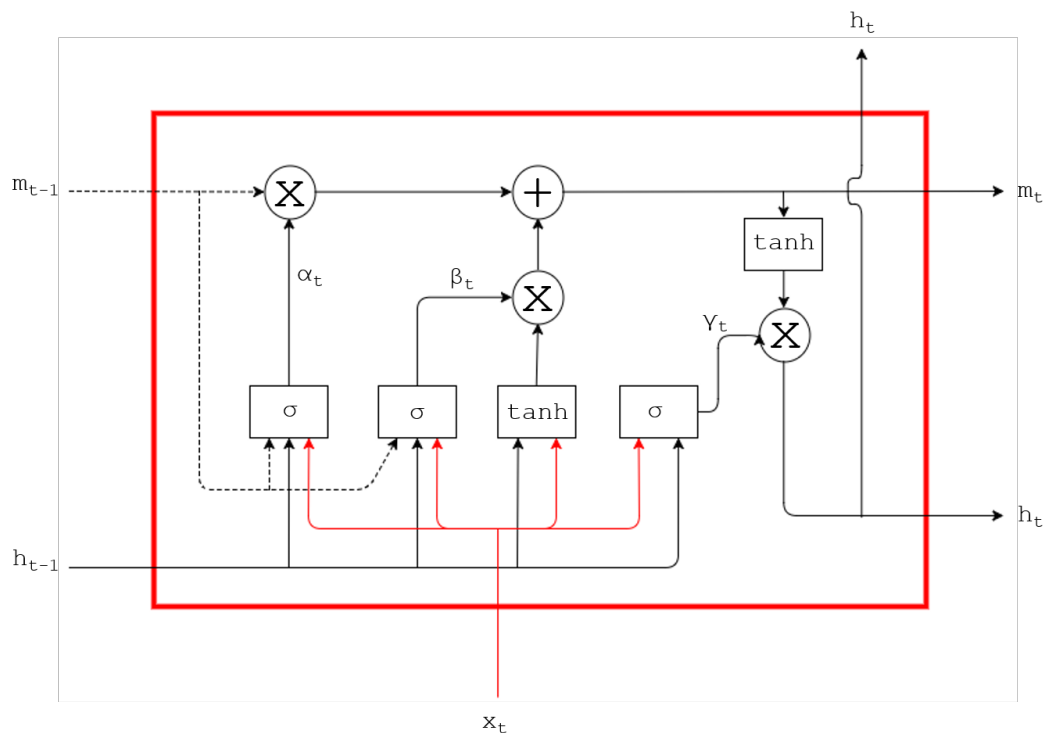
dengan f sebagai *activation function*. Untuk lebih jelasnya, berikut merupakan gambar dari satu buah *timestep* di dalam RNNs



Gambar 2.3: 1 buah *timestep* dalam RNNs

2.3.1 Long Short Term Memory

Pada penjelasan di atas, RNNs memiliki kelebihan mempertimbangkan konteks untuk mengolah *input* menjadi *output*. Sayangnya, *range* konteks yang dapat digunakan dalam satu blok terbatas (Graves, 2012). Efek dari keterbatasan ini yaitu informasi pada suatu blok akan hilang atau terganggu dalam perjalanan *timestep* sehingga *output* yang dihasilkan tidak sesuai harapan. Oleh karena itu RNNs tidak dapat menangani permasalahan dependensi jangka panjang. Permasalahan ini disebut dengan *vanishing gradient problem*. Banyak upaya untuk mengatasi masalah ini, seperti dengan menggunakan *simulated annealing* dan *discrete error propagation*, menggunakan *time delays* atau *time constant*, dan *hierarchical sequence compression*. Namun sejauh ini solusi yang paling bagus yaitu dengan arsitektur *Long Short Term Memory* (LSTM).



Gambar 2.4: 1 buah blok memori dalam LSTM

LSTM diperkenalkan oleh Hochreiter dan Schmidhuber (1997) dan saat ini banyak digunakan dalam berbagai *task*. Gambar 2.4 merupakan ilustrasi satu buah blok memori di dalam LSTM. Pada dasarnya, arsitektur LSTM mirip dengan RNNs, namun unit *nonlinear* pada *hidden layer* di dalam RNNs diganti menjadi blok memori. Sebuah blok memori memiliki gerbang *multiplicative* yang berfungsi untuk menyimpan dan mengakses informasi dari blok sebelumnya namun dengan batasan yang jauh lebih besar dibanding RNNs, sehingga mampu menghindari *vanishing gradient problem*. Apabila *input gate* selalu tertutup, maka memori tidak akan pernah ditimpa sehingga isi memori tidak berubah.

Pada gambar 2.4, kita dapat melihat bahwa 1 blok memori pada LSTM tersebut memiliki 3 buah gerbang, yang berfungsi untuk sebagai pengatur suatu informasi apakah ditambahkan, dipertahankan atau dihapus di dalam sebuah sel. Masing-masing gerbang terdiri dari komponen *sigmoid layer* dan komponen untuk melakukan operasi penjumlahan atau perkalian untuk masing-masing *element-wise*. *Sigmoid layer* tersebut memiliki nilai antara nol sampai dengan satu, yang mendeskripsikan perilaku gerbang dalam menerima *input*. Semakin kecil nilai dari layer tersebut maka semakin kecil pula informasi masuk ke gerbang terkait dan sebaliknya.

1. Forget Gate

Gerbang ini memiliki fungsi untuk menentukan informasi yang akan disimpan di dalam memori dengan formula berikut

$$\alpha_t = \tau(W_{x\alpha} + W_{h\alpha} \cdot h_{t-1} + W_{m\alpha} \cdot h_{t-1}) \quad (2.5)$$

2. *Input Gate*

3. *Output Gate*

2.3.2 Penerapan RNNs untuk MER

– Belum

2.4 Word Embedding

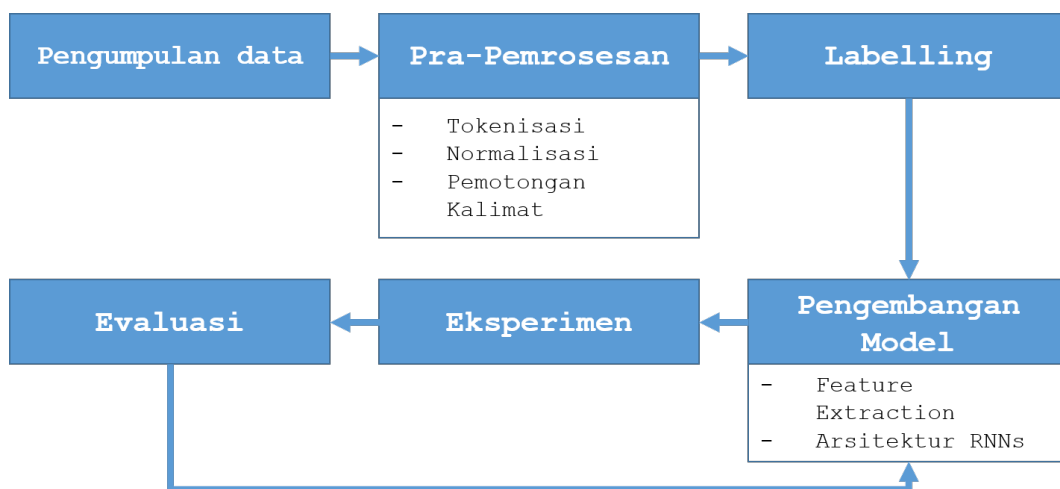
– Belum

BAB 3

METODOLOGI

3.1 Rancangan 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

Pengumpulan data dilakukan dengan tujuan untuk mendapatkan data *training* dan *testing* sebagai *input* sistem MER yang diusulkan. Data yang dimaksud merupakan teks dari forum kesehatan *online* dari berbagai sumber.

3.1.2 Pra-Pemrosesan

Pra-pemrosesan dilakukan dengan tujuan supaya teks yang diberikan mampu dibaca oleh sistem MER. Dalam tahap ini, ada dua pekerjaan utama yang perlu dilakukan, yaitu:

1. Pembersihan data

Langkah ini dilakukan dengan tujuan untuk membersihkan dokumen dari berbagai karakter yang tidak terdapat dalam karakter ASCII. Selain itu,

perlu juga pengubahan beberapa tanda baca/token menjadi 1 token tersendiri. Misalnya sebuah token tautan (www.alodokter.com/asma/pengobatan) diganti menjadi token "url".

2. Tokenisasi

Tokenisasi dilakukan untuk mendapatkan token yang paling tepat sebagai satu buah kata. Hal ini perlu dilakukan untuk menghindari beberapa kelompok token berbeda yang tergabung. Misalnya karakter abjad dengan karakter angka atau karakter abjad dengan karakter tanda baca dipisahkan berdasarkan kelompoknya.

3. Pemotongan kalimat

Untuk menghindari jumlah token yang timpang dalam kalimat yang berbeda dan data yang *sparse*, diperlukan pemotongan kata yang tepat. Langkah-langkah yang diperlukan untuk melakukan pemotongan kata yaitu:

- (a) memisahkan kalimat berdasarkan tanda baca (.,!?,),
- (b) apabila suatu kalimat memiliki jumlah kata yang sedikit (misal diberikan batasan minimal 10 kata dalam satu kalimat), kalimat tersebut digabungkan dengan kalimat setelahnya.

3.1.3 Pelabelan

Pada tahap ini, penulis melakukan pelabelan pada dokumen teks yang merupakan hasil pada tahap sebelumnya dengan label *disease*, *symptom*, *drug* dan *treatment*. Berikut merupakan penjelasan dari masing-masing label:

1. *Disease*

Entitas *disease* yang dimaksud pada penelitian ini yaitu nama dari suatu penyakit. Penyakit merupakan keadaan abnormal yang timbul pada tubuh manusia. Contoh dari entitas *disease* yaitu:

- Skizofrenia
- Trikotilomania
- Diabetes melitus

2. *Symptom*

Entitas *symptom* yang dimaksud pada penelitian ini yaitu fenomena yang dialami oleh seseorang yang terkena suatu penyakit. Contoh dari entitas *symptom* yaitu:

- Napas berbunyi
- Benjolan di daerah perut
- Nyeri saat BAK

3. *Drug*

Entitas *drug* merupakan entitas nama obat dari suatu penyakit yang memiliki fungsi untuk mengurangi atau menyembuhkan penyakit tersebut. Contoh dari entitas *drug* yaitu:

- Paracetamol
- Diltiazem
- eritropoetin-alfa

4. *Treatment*

Entitas *treatment* merupakan cara atau langkah penyembuhan dari suatu penyakit. Contoh dari entitas *treatment* yaitu:

- Pemeriksaan darah rutin
- Penilaian denyut kapiler
- Terapi inhalasi

3.1.4 Pengembangan Model

Pada tahap ini, penulis melakukan pengusulan dan perancangan model yang nantinya akan penulis evaluasi pada tahap eksperimen. Dalam mengembangkan model, terdapat dua pekerjaan yang penulis lakukan, yaitu:

3.1.4.1 Ekstrasi Fitur

Pada tahap ini, penulis melakukan ekstraksi fitur dari dokumen yang telah diberi label entitas. Ada beberapa fitur yang penulis usulkan dalam penelitian ini yang nantinya penulis kombinasikan supaya mendapatkan hasil terbaik. Fitur-fitur tersebut yaitu:

1. Fitur 1: Kata itu sendiri

Fitur ini merupakan fitur kata dalam representasi vektor. Untuk mendapatkan representasi vektor dari masing-masing kata, penulis menggunakan *word embedding*. Terdapat beberapa langkah yang perlu penulis lakukan dalam memanfaatkan *word embedding* ini, yaitu:

- (a) Pengumpulan data *training* untuk *word embedding*
- (b) *Training* untuk mendapatkan model *word embedding*
- (c) Pengubahan kata menjadi vektor dari model yang didapatkan

2. Fitur 2: *Part of Speech Tag* (POS-Tag)

Fitur ini merupakan fitur *tag* yang dimiliki setiap kata yang diusulkan oleh Abacha dan Zweigenbaum (2011) dalam penelitiannya di bidang MER. Model yang digunakan merupakan model POS-Tag berbahasa Indonesia.

3. Fitur 3: *Stopword*

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

4. Fitur 4: Kamus kesehatan

Fitur kamus kata merupakan fitur yang berisi informasi suatu kata terdapat di dalam kamus kesehatan atau tidak. Pada penelitian ini, kamus kesehatan yang dipakai ada sejumlah entitas yang akan dikenali, yaitu kamus *disease*, kamus *symptom*, kamus *drug* dan kamus *treatment*.

5. Fitur 5: Frasa kata benda

Menurut Hs (2005), frasa kata benda sendiri merupakan kelompok kata benda yang dibentuk dengan memperluas kata benda ke sekelilingnya. Fitur frasa kata benda yang penulis gunakan dalam penelitian merupakan fitur yang berisi informasi suatu kata atau kumpulan kata merupakan frasa kata benda atau bukan. Dalam menentukan suatu kata merupakan frasa atau bukan, penulis menggunakan aturan pembentukan frasa yang digunakan pada bahasa Indonesia, yaitu:

- NP : NN
- NP : NNP
- NP : PR
- NP : PRP
- NP : NN + NN
- NP : NN + NNP
- NP : NN + PR
- NP : NN + PRP
- NP : NN + JJ

- NP : DT + NN
- NP : RB + NN
- NP : CD + NN
- NP : NND + NN

6. Fitur 6: Frasa verbal

Menurut Hs (2005), frasa verbal merupakan kelompok kata benda yang dibentuk dengan mkata kerja. Fitur frasa verbal yang penulis gunakan dalam penelitian merupakan fitur yang berisi informasi suatu kata atau kumpulan kata merupakan frasa verbal atau bukan. Dalam menentukan suatu kata merupakan frasa atau bukan, penulis menggunakan aturan pembentukan frasa yang digunakan pada bahasa Indonesia, yaitu:

- VP : VB
- VP : VB + NP

7. Fitur 7: 1 kata sebelum

Fitur ini merupakan fitur yang berisi informasi kata sebelum kata saat ini yang direpresentasikan dalam bentuk vektor untuk masing-masing kata.

8. Fitur 8: 1 kata sesudah

Fitur ini merupakan fitur yang berisi informasi kata sesudah kata saat ini yang direpresentasikan dalam bentuk vektor untuk masing-masing kata.

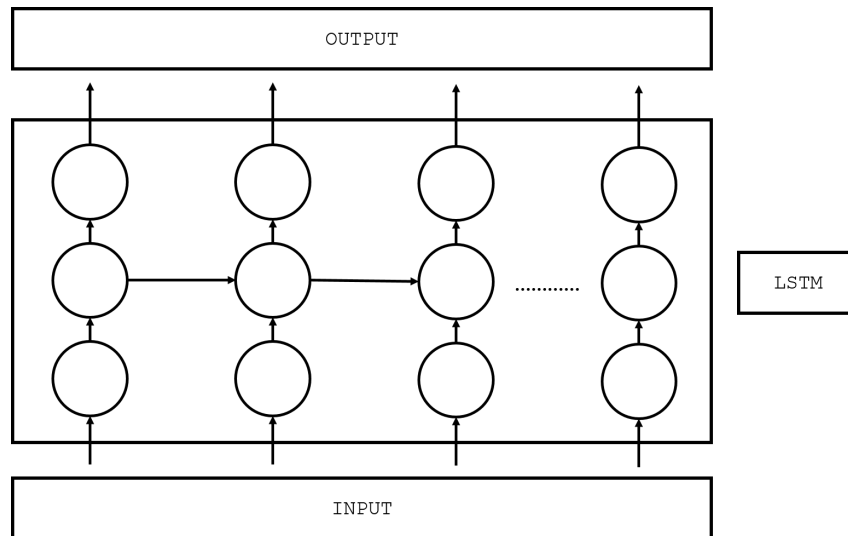
9. Fitur 9: 2 kata sebelum
Fitur ini merupakan fitur yang berisi informasi 2 kata sebelum kata saat ini yang direpresentasikan dalam bentuk vektor untuk masing-masing kata.

3.1.4.2 Pengusulan Arsitektur RNNs

Pada tahap ini penulis mengusulkan arsitektur RNNs yang akan digunakan pada tahap eksperimen. Ada dua arsitektur yang penulis gunakan dalam penelitian ini, yaitu

1. LSTM 1 layer

Pada LSTM 1 layer, semua fitur yang menjadi input digabung menjadi satu. Misalnya Berikut merupakan ilustrasi LSTM 1 layer yang penulis gunakan dalam penelitian ini.



Gambar 3.2: LSTM 1 layer

2. LSTM layer bertingkat Pada LSTM 2 layer, penulis mendefinisikan 2 layer, dimana layer terbawah merupakan layer dengan jumlah LSTM sebanyak n kelompok fitur. Pertama-tama fitur dikelompokkan terlebih dahulu, kemudian dijadikan input untuk LSTM layer pertama. Setelah itu, hasil dari layer pertama tersebut akan digabung menjadi satu, dengan menggunakan layer penggabung. *Output* dari layer penggabung kemudian dimasukkan ke dalam LSTM layer kedua, yang *output*-nya merupakan hasil klasifikasi label. Berikut merupakan ilustrasi dari LSTM layer bertingkat yang penulis gunakan.

Gambar LSTM 2(Belum)

3.1.5 Eksperimen

Dalam melakukan eksperimen, arsitektur *deep learning* yang penulis gunakan ada;ah *Recurrent Neural Networks*, dalam hal ini penulis menggunakan LSTM. Hal ini penulis lakukan karena pada penelitian Mujiono et al. (2016), LSTM memberikan *output* terbaik dalam MER yang dirancang. Selain itu, LSTM juga sangat baik dalam masalah *sequence labelling* seperti yang dilakukan oleh Graves et al. (2013) dan merupakan *state-of-the-art* dalam bidang ini. Masih banyak penelitian lain yang membuktikan bahwa LSTM merupakan arsitektur *deep learning* yang sangat baik dalam masalah *sequence labelling* seperti *Offline Handwriting Recognition* (Graves dan Schmidhuber, 2009), *sequence tagging* (Huang et al., 2015), *Sequence to Sequence Learning* (Sutskever et al., 2014) dan lain lain.

Eksperimen yang penulis lakukan menggunakan *10-cross fold validation*, karena keterbatasan data *training* yang penulis miliki. Sebelum melakukan eksperimen, penulis membagi data *training* menjadi 10 bagian, kemudian melakukan iterasi sebanyak 10 kali dimana pada masing-masing iterasi ke-*i*, bagian data ke-*i* menjadi data *testing* dan yang lainnya digabung menjadi data *training*.

Setelah melakukan pembagian dan pengelompokan data berdasarkan nomor iterasi, penulis membuat model dari data *training* tersebut. Setelah penulis mendapatkan model, penulis melakukan testing terhadap masing-masing model dengan data *testing* yang telah disediakan sebelumnya. Hasil dari pelabelan data *testing* ini akan penulis evaluasi di tahap selanjutnya. Setelah itu penulis kembali melakukan pembuatan model dengan fitur yang berbeda, atau dengan tambahan fitur lain. Dalam perjalanan melakukan pengujian, apabila fitur yang diuji memberikan hasil yang bagus atau menambah akurasi, penulis menggabungkan fitur ini ke percobaan selanjutnya. Namun apabila fitur pada saat ini memberikan akurasi yang lebih jelek, penulis tidak menggunakan fitur tersebut di percobaan selanjutnya.

3.1.6 Evaluasi

Pada tahap ini, penulis melakukan serangkaian evaluasi dari data *testing* yang telah dilabeli dengan model yang dihasilkan pada tahap eksperimen. Penulis melakukan evaluasi dengan menggunakan metode *partial evaluation* di mana sebuah token yang diprediksi entitas oleh model dihitung benar apabila terdapat fragmen yang menyusun entitas bernama tersebut (Seki dan Mostafa, 2003). Aturan yang penulis gunakan dalam melakukan evaluasi adalah sebagai berikut:

1. Perhitungan nilai *True Positive* (TP)

Untuk masing-masing kata yang mendapat label entitas benar, nilai *TP* bertambah sejumlah kata yang diprediksi benar.

Misal:

Contoh 1

True: Bu Ani <Disease>**sakit kepala** sebelah</Disease>

Predicted: Bu Ani <Disease>**sakit kepala**</Disease> sebelah

Dari contoh di atas, nilai $TP = 2$, karena ada 2 kata yang mendapatkan label entitas yang benar.

Contoh 2

True : <Disease>Masuk angin</Disease> dan <Sympton>**suhu badan tinggi**</Sympton>

Predicted : <Sympton>Masuk angin</Sympton> dan <Sympton>**suhu badan tinggi**</Sympton>

Dari contoh di atas, nilai $TP = 3$, karena ada 3 kata yang mendapatkan label entitas yang benar

2. Perhitungan nilai *False Positive* (FP)

Untuk masing-masing kata yang mendapat label entitas namun seharusnya tidak berentitas, nilai FP bertambah sejumlah kata yang diprediksi salah.

Misal:

Contoh 1

True : <Disease>Sakit kepala</Disease> sudah **beberapa hari istirahat**

Predicted : <Disease>Sakit kepala</Disease> sudah <Treatment>**beberapa hari istirahat**</Treatment>

Dari contoh di atas, nilai $FP = 3$, karena ada 3 kata yang mendapat label entitas yang seharusnya tidak berlabel, yaitu "beberapa hari istirahat".

3. Perhitungan nilai *False Negative* (FN)

Untuk masing-masing kata yang mendapat label entitas salah, nilai FP bertambah sejumlah kata yang diprediksi salah.

Misal:

Contoh 1

True : Bu Ani <Disease>sakit kepala sebelah</Disease>

Predicted : Bu Ani <Disease>sakit kepala</Disease> sebelah

Dari contoh di atas, nilai $FN = 0$, karena tidak ada kata yang mendapat label entitas salah (kata "sebelah" tidak mendapat label).

Contoh 2

True : <Symptom>Badan terasa pegal</Symptom>, sepertinya akan <Disease>**demam**</Disease>.

Predicted : <Symptom>Badan terasa pegal</Symptom>, sepertinya akan <Symptom>**demam**</Symptom>.

Dari contoh di atas, nilai $FN = 1$, karena ada 1 kata yang mendapat label entitas salah, yaitu kata "demam".

Setelah mendapatkan angka TP , FP dan FN , penulis menghitung *f-measure*,

precision dan *recall* untuk masing-masing entitas dengan menggunakan formula:

$$Precision = \frac{TP}{TP + FP} \quad (3.1)$$

$$Recall = \frac{TP}{TP + FN} \quad (3.2)$$

$$F - Measeure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (3.3)$$

Angka-angka hasil evaluasi ini akan menjadi pertimbangan untuk penggunaan fitur pada saat ini di eksperimen selanjutnya. Apabila akurasi dari penggunaan fitur saat ini lebih baik atau meningkat dari eksperimen sebelumnya, penulis menggunakan fitur ini pada eksperimen selanjutnya. Selain itu, penulis juga mengevaluasi arsitektur RNNs yang penulis gunakan.

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 (μ) dan standar deviasi (σ) dari 10 ms sampel pertama pada audio. Jika fs menyatakan *bit rate* dari audio, maka terdapat sebanyak s sampel dalam 10 ms.

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

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

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

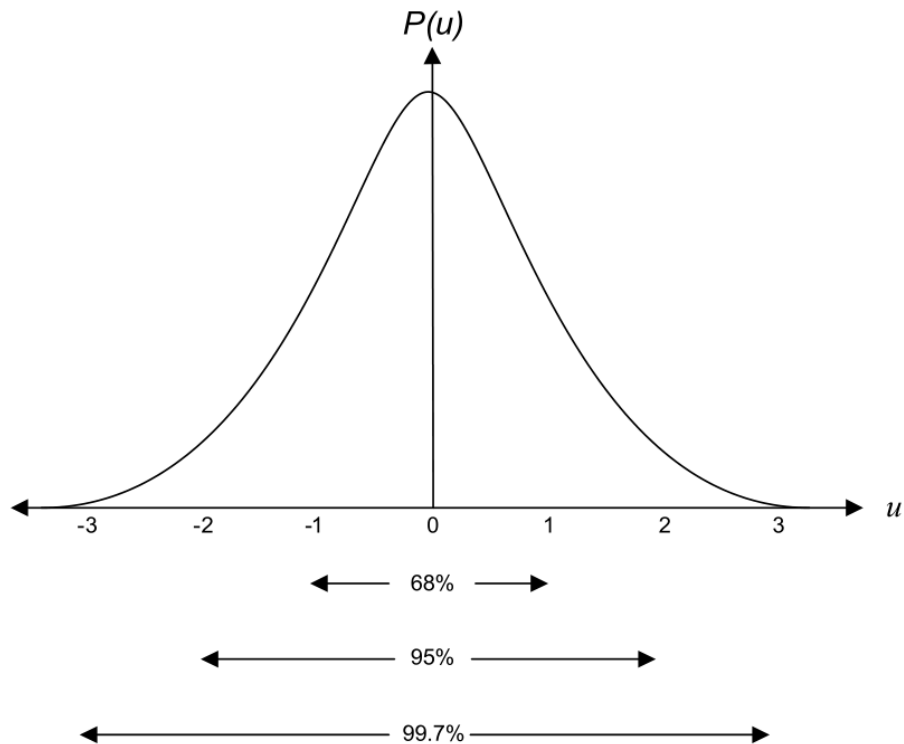
Background noise juga dikenali dari berdasarkan nilai rata-rata (μ) dan standar deviasinya (σ).

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* ≤ 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 ???. 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

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

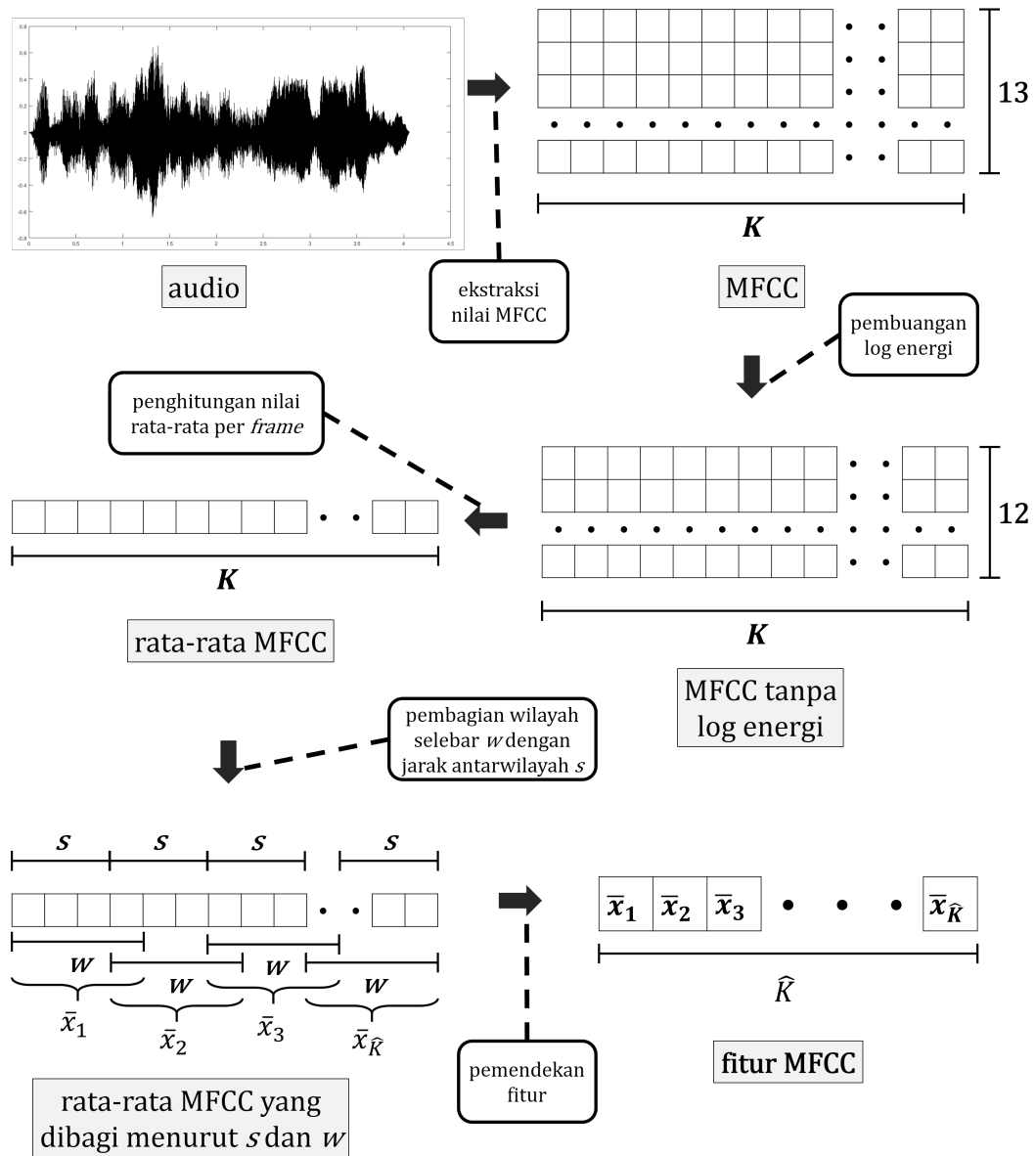
2. Panggil fungsi *mfcc* dengan parameter yang telah disebutkan pada tabel 4.1 dan nilai *bit rate* sama dengan \hat{fs} , sehingga memberikan *output* berupa matriks MFCC $13 \times K$, di mana K menyatakan banyaknya *frame*.
3. Buang matriks MFCC pada baris pertama, karena nilai tersebut merupakan 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 ???. 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.

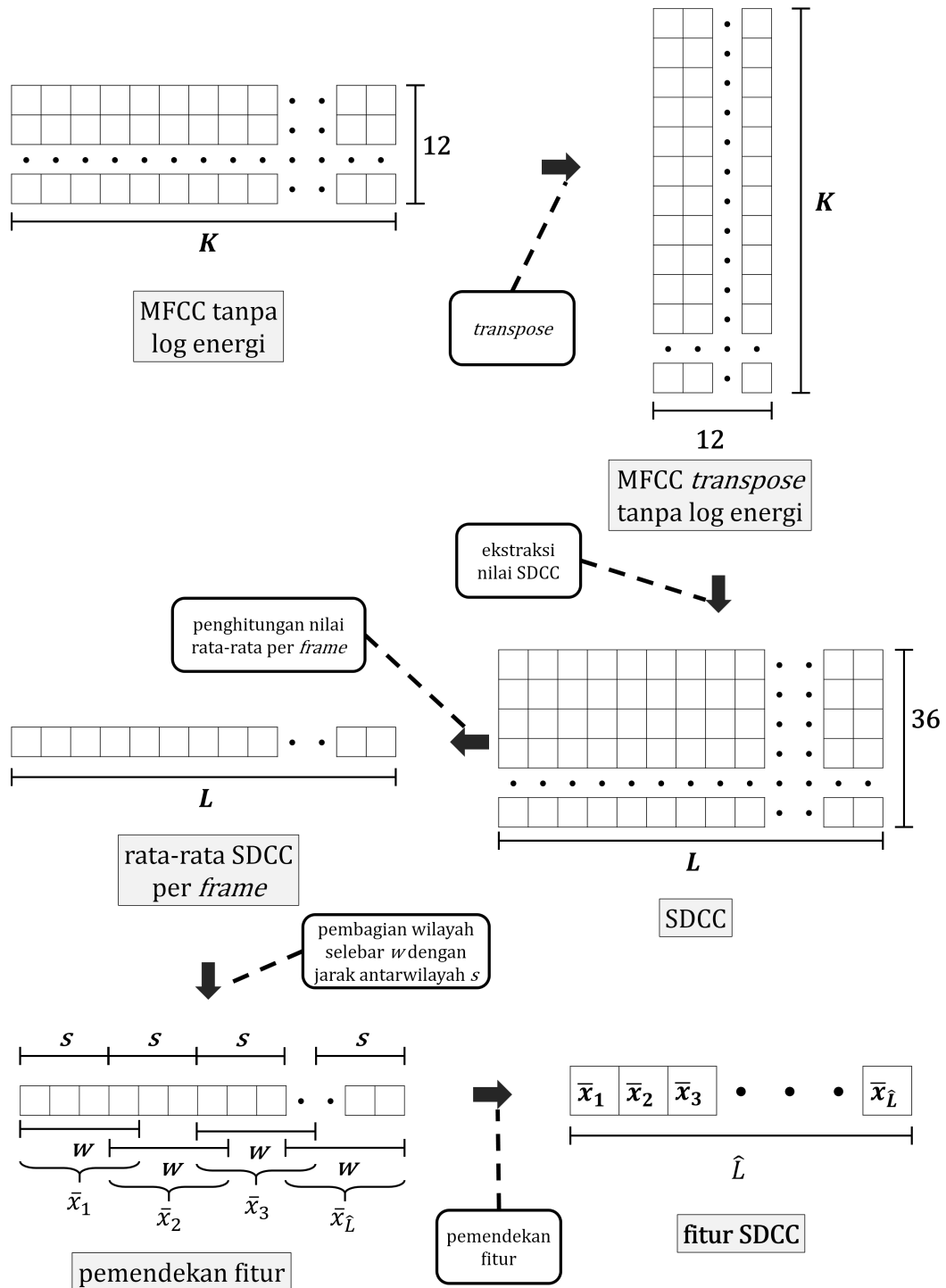
1. Hitung nilai MFCC menggunakan fungsi *mfcc* seperti sudah dijelaskan di Bab 4.2.1, langkah 1 sampai 3, sehingga diperoleh nilai MFCC berupa matriks berukuran $12 \times K$. Nilai MFCC berukuran $12 \times K$ perlu di-*transpose* terlebih dahulu menjadi ukuran $K \times 12$ untuk menyesuaikan kebutuhan 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 ??.

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*¹ dalam proses pemodelan dan *svmclassify*² 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

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

²<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*³ 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.

³<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 2×2 yang merepresentasikan Tabel ??.
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 ?? 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

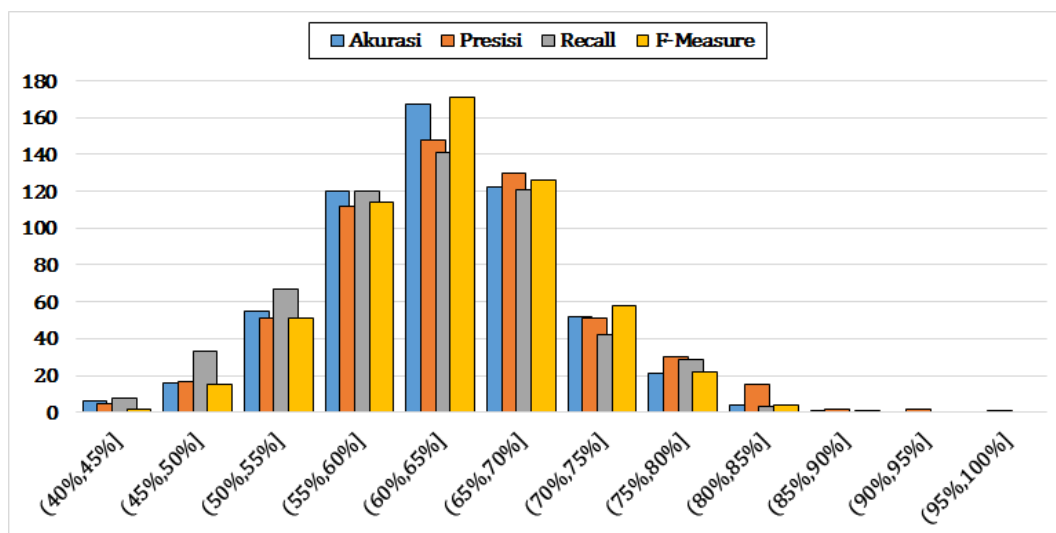
	Akurasi	Presisi	<i>Recall</i>	<i>F-Measure</i>
Minimal	41.3%	42.2%	42.5%	44.4%
Maksimal	87.5%	96.2%	82.5%	86.8%
Rata-rata	63.1%	63.7%	63.0%	63.1%
Standar Deviasi	7.24%	8.09%	7.75%	6.86%

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

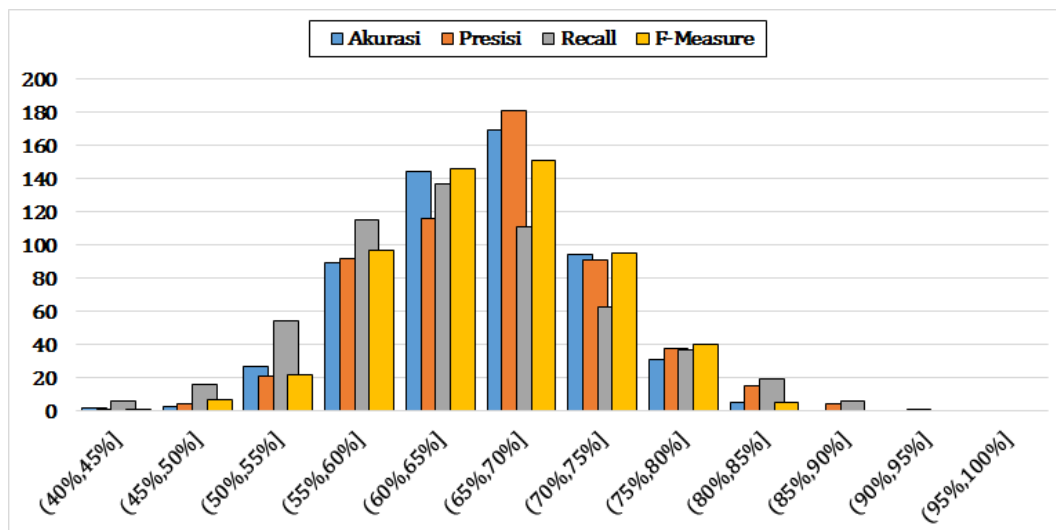
	Akurasi	Presisi	<i>Recall</i>	<i>F-Measure</i>
Minimal	43.8%	43.9%	42.5%	44.4%
Maksimal	85.0%	93.3%	90.0%	85.0%
Rata-rata	65.8%	66.3%	65.1%	65.4%
Standar Deviasi	6.50%	7.19%	8.56%	6.72%

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

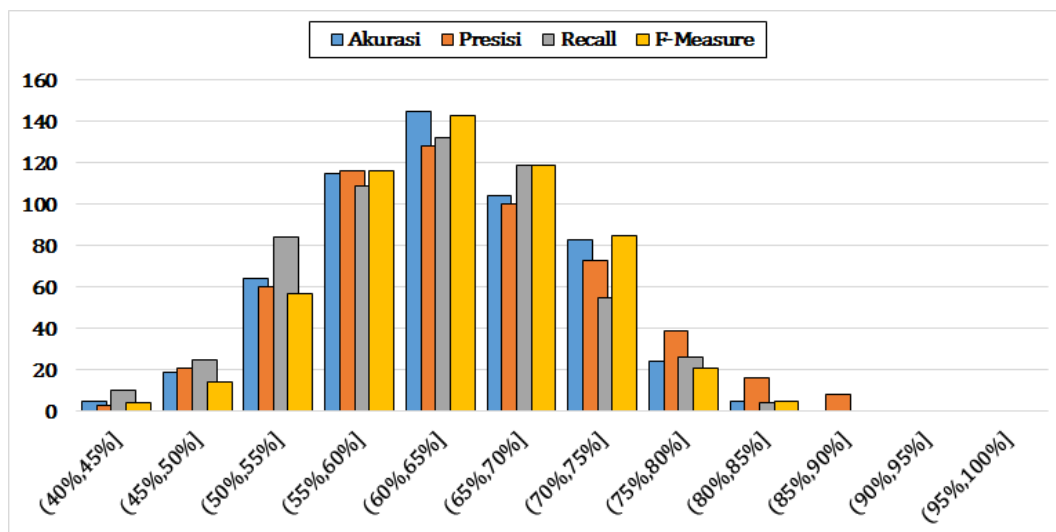
	Akurasi	Presisi	Recall	F-Measure
Minimal	41.3%	42.2%	37.5%	43.9%
Maksimal	85.0%	90.0%	85.0%	84.2%
Rata-rata	63.5%	64.1%	63.1%	63.4%
Standar Deviasi	7.64%	8.71%	8.00%	7.26%

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

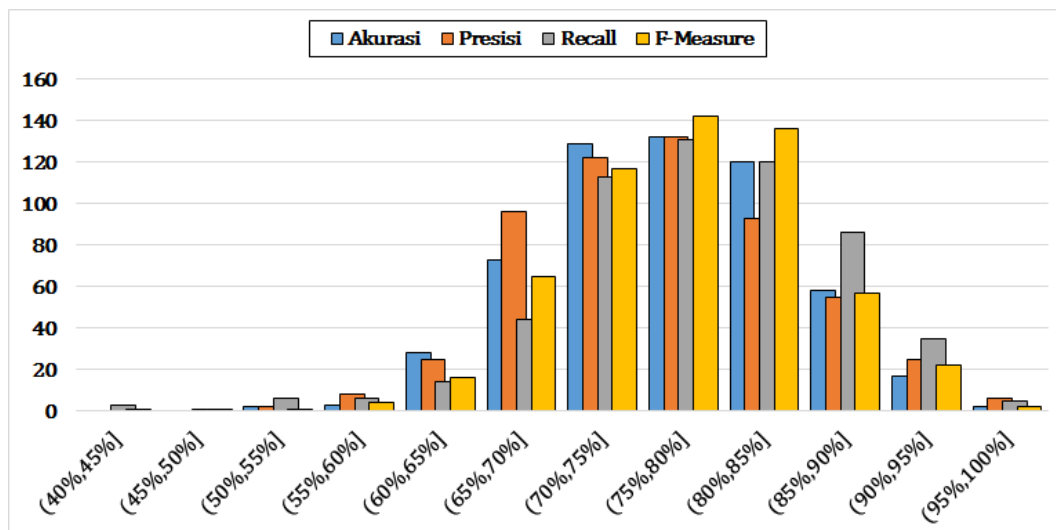
	Akurasi	Presisi	<i>Recall</i>	<i>F-Measure</i>
Minimal	51.3%	51.2%	30.0%	43.6%
Maksimal	96.3%	100.0%	100.0%	96.4%
Rata-rata	77.3%	76.6%	79.5%	77.8%
Standar Deviasi	7.47%	8.18%	8.74%	7.41%

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

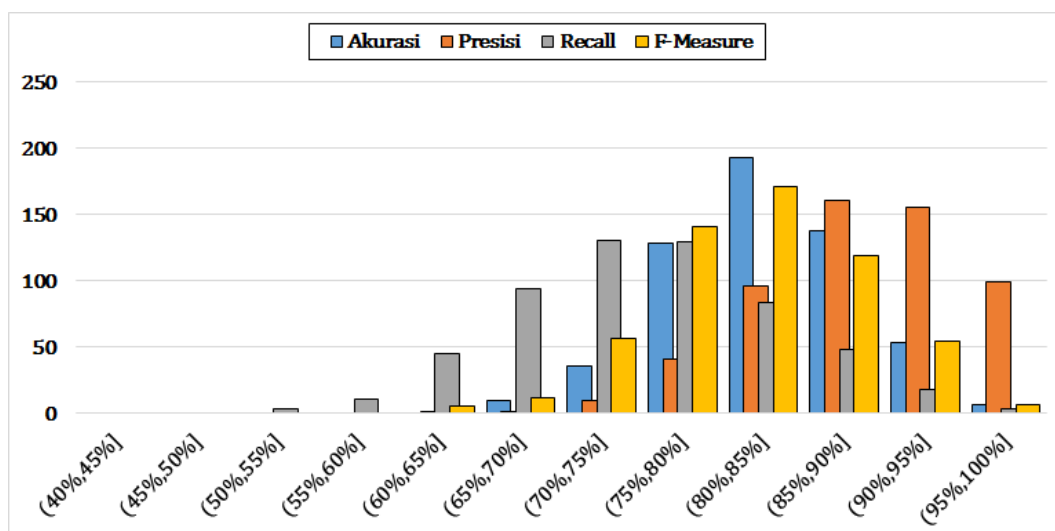
	Akurasi	Presisi	Recall	F-Measure
Minimal	67.5%	64.6%	55.0%	63.8%
Maksimal	98.8%	100.0%	100.0%	98.8%
Rata-rata	83.4%	89.0%	76.4%	82.0%
Standar Deviasi	5.61%	6.39%	8.18%	6.33%

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

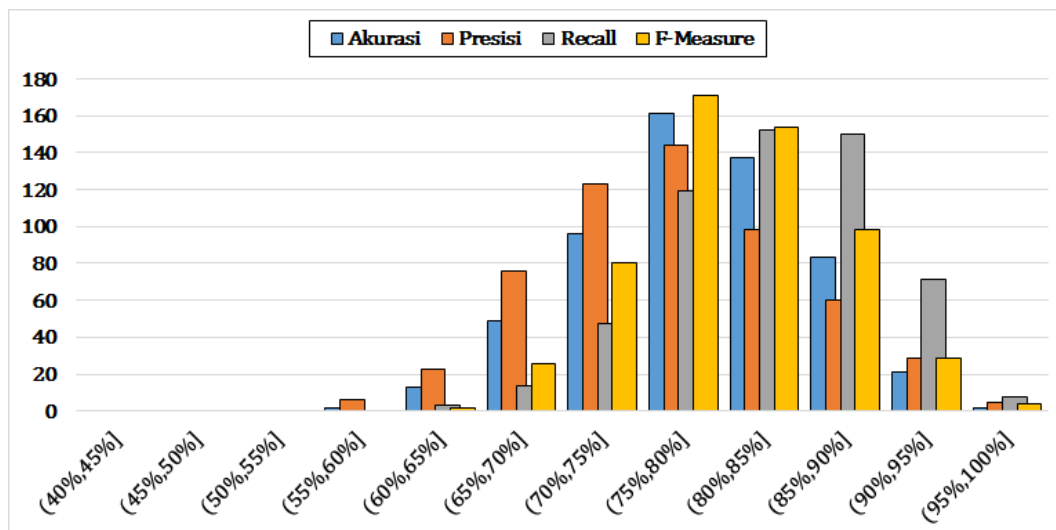
	Akurasi	Presisi	<i>Recall</i>	<i>F-Measure</i>
Minimal	57.5%	56.5%	65.0%	60.5%
Maksimal	96.3%	100.0%	100.0%	96.4%
Rata-rata	79.4%	77.3%	84.2%	80.4%
Standar Deviasi	6.86%	7.80%	6.48%	6.16%

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.



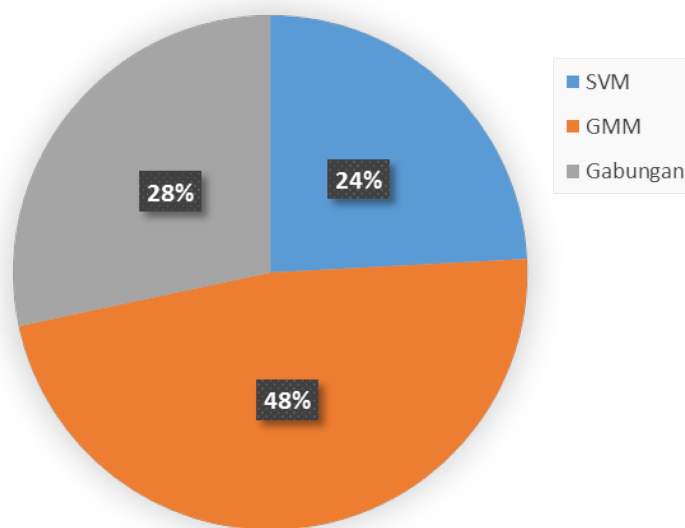
Gambar 5.6: Histogram Metrik Evaluasi dengan Fitur SDCC dan Metode Klasifikasi Gabungan

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.

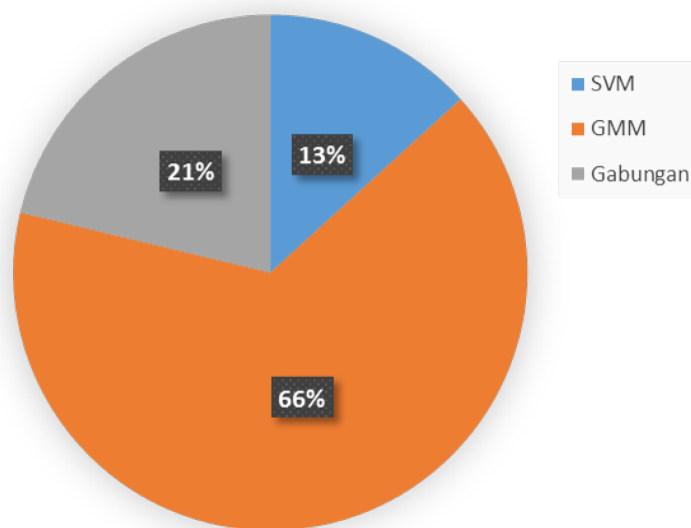


Gambar 5.7: Perbandingan Metode Klasifikasi untuk Fitur MFCC

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.



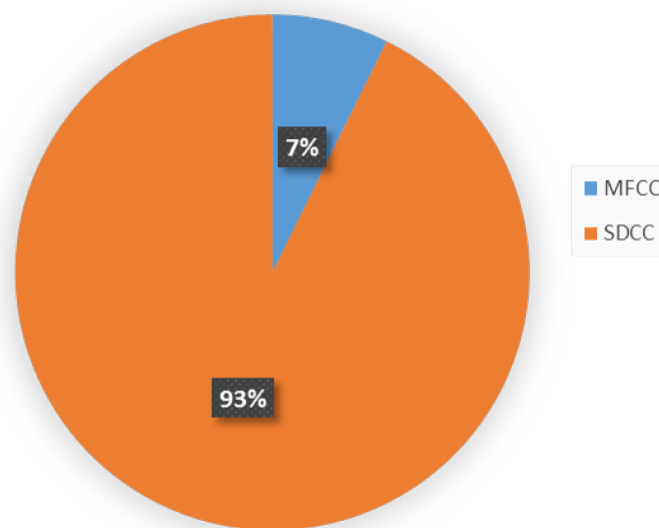
Gambar 5.8: Perbandingan Metode Klasifikasi untuk Fitur SDCC

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.

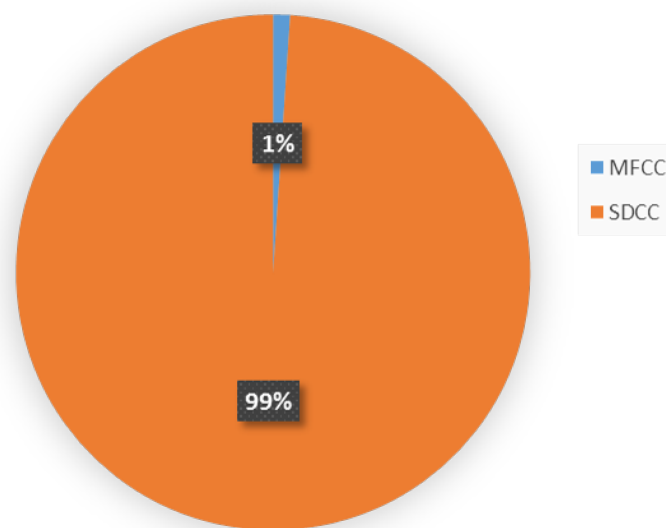


Gambar 5.9: Perbandingan Fitur untuk Metode Klasifikasi SVM

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.

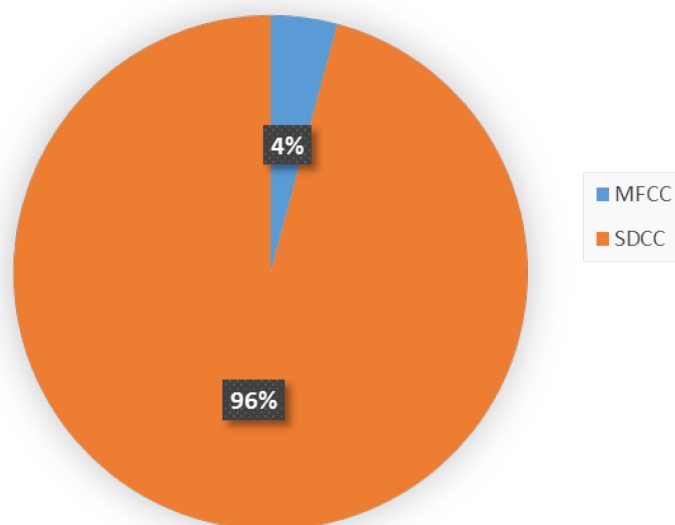


Gambar 5.10: Perbandingan Fitur untuk Metode Klasifikasi GMM

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.
- Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York.
- 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.
- Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.
- Graves, A. (2012). Neural networks. In *Supervised Sequence Labelling with Recurrent Neural Networks*, pages 15–35. Springer.
- 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.
- Graves, A. dan Schmidhuber, J. (2009). Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in neural information processing systems*, pages 545–552.
- Hachey, B., Radford, W., Nothman, J., Honnibal, M., dan Curran, J. R. (2013). Evaluating entity linking with wikipedia. *Artificial intelligence*, 194:130–150.
- 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 conditional random fields. Master's thesis, Universitas Indonesia, Kampus UI Depok.
- Hochreiter, S. dan Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

- Hs, W. (2005). *Bahasa Indonesia: mata kuliah pengembangan kepribadian di perguruan tinggi*. Gramedia Widiasarana Indonesia.
- Huang, Z., Xu, W., dan Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.
- Jordan, M. I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine.
- Lang, K. J., Waibel, A. H., dan Hinton, G. E. (1990). A time-delay neural network architecture for isolated word recognition. *Neural networks*, 3(1):23–43.
- Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., dan Khudanpur, S. (2010). Recurrent neural network based language model. In *Interspeech*, volume 2, page 3.
- 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.
- Mujiono, S., Fanany, M. I., dan Basaruddin, C. (2016). A new data representation based on training data characteristics to extract drug named-entity in medical text. *arXiv preprint arXiv:1610.01891*.
- Rabiner, L. dan Schafer, R. (1978). *Digital Processing of Speech Signals*. Prentice-Hall signal processing series. Prentice-Hall.
- 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.
- Seki, K. dan Mostafa, J. (2003). A probabilistic model for identifying protein names and their name boundaries. In *Bioinformatics Conference, 2003. CSB 2003. Proceedings of the 2003 IEEE*, pages 251–258. IEEE.
- 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.

- Sutskever, I., Vinyals, O., dan Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., dan Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.
- Suwarningsih, W., Supriana, I., dan Purwarianti, A. (2014). Inner indonesian medical named entity recognition. In *Technology, Informatics, Management, Engineering, and Environment (TIME-E), 2014 2nd International Conference on*, pages 184–188. IEEE.
- 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*.
- Wyse, N., Dubes, R., dan Jain, A. K. (1980). A critical evaluation of intrinsic dimensionality algorithms. *Pattern recognition in practice*, pages 415–425.
- 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.