DAFTAR ISTILAH

Classifier	Alat/metode yang digunakan untuk melakukan klasifikasi terhadap suatu objek
Fitur	Ciri suatu objek
Numpy	Alat bantu tambahan untuk komputasi sains pada Python
Magnitude	Tingkat energi suatu sinyal
Spectrum	Nilai-nilai *magnitude* dalam sebaran frekuensi
Frekuensi	Jumlah getaran yang terjadi dalam satu detik
Amplitudo	Jarak terjauh simpangan dari suatu titik keseimbangan
Silabel	Suku kata
Dimensi	Aspek yang meliputi atribut yang membentuk suatu entitas
Stride	Banyaknya langkah yang diambil dari satu titik untuk menuju ke titik berikutnya

BAB 1 PENDAHULUAN

Bagian ini berisi penjelasan latar belakang pemilihan masalah, perumusan masalah, tujuan, hipotesis, metodologi, dan sistematika penulisan tugas akhir.

1.1 Latar Belakang

Al-Qur'an adalah pedoman pertama dan utama bagi umat Islam yang diberikan oleh Allah SWT dalam Bahasa Arab [1]. Al-Qur'an menjadi suatu hal yang sangat mudah ditemukan di Indonesia. Berdasarkan sensus penduduk yang dilakukan Badan Pusat Statistik pada 2010, ada sekitar 207 juta umat Islam di tanah air dan menjadikan Islam sebagai agama dengan pemeluk terbanyak di Indonesia [2]. Namun, sulitnya membaca dan memahami Al-Qur'an ternyata masih sering dialami oleh sebagian masyarakat Islam di tanah air. Hal ini wajar karena Al-Qur'an merupakan *kitabullah* yang dituliskan berbahasa Arab, sedangkan Bahasa nasional Indonesia adalah Bahasa Indonesia. Masyarakat pun perlu belajar terlebih dahulu bagaimana pelafalan dan cara membaca Al-Qur'an yang baik dan benar. Hal ini dikarenakan kesalahan membaca kata di Al-Qur'an dapat mengubah arti dari kata tersebut.

Kesalahan-kesalahan yang paling umum ditemui dalam pengucapan *lafadz* Al-Qur'an adalah kesalahan pengucapan huruf *hijaiyah* bertanda baca. Masyarakat Indonesia baik yang sudah paham ataupun belum seringkali mengalami kesalahan pada poin di atas. Hal ini tentu dapat dihindari dengan mempelajari *makharijul huruf* dan harakat dengan sebaik-baiknya, senantiasa berlatih dan meminta orang yang lebih ahli untuk mendengarkan bacaan sekaligus mengoreksi apabila terdapat kesalahan.

Di sisi lain, perkembangan teknologi informasi terus melaju pesat. *Speech Recognition* adalah salah satu teknologi yang menjadi *trend* masa kini dan di masa yang akan datang. Dengan *Speech Recognition*, komputer dapat mengenali suara seseorang yang menggunakan microphone atau benda lainnya dan komputer tersebut dapat memahami apa yang dibicarakan oleh orang tersebut [3]. Penelitian *Speech Recognition* terkait pemrosesan suara bacaan Al-Qur'an sudah pernah dilakukan sebelumnya oleh Hyassat dan Abu Zitar pada tahun 2008. Pada kala itu, mereka membangun sebuah sistem *Arabic Speech Recognition* berbasiskan Sphinx4. Mereka juga mengajukan suatu *toolkit* otomatis yang dapat digunakan untuk membangun kamus Al-Qur'an dan beberapa standar dalam Bahasa Arab [4]. Selain penelitian tersebut, Iqbal et al. pernah melakukan penelitian terkait pembagian segmentasi dan identifikasi vokal menggunakan transisi format yang terjadi pada pembacaan Al-Qur'an. Namun penelitian terhadap Bahasa Arab memang lebih sulit ditemukan ketimbang dengan Bahasa lainnya [5]. Setelah sistem selesai dibangun, performansi sistem diharapkan dapat menunjukkan hasil yang memuaskan. Dari banyaknya pembangunan sistem rekognisi suara, cukup jarang ditemukan pembangunan sistem yang menggunakan pendekatan geometrik, sehingga penelitian ini mencoba membangun sistem dengan pendekatan tersebut. *K-Nearest Neighbor* (KNN) merupakan salah satu metode klasifikasi menggunakan pendekatan geometrik. Namun, KNN memiliki sulit diterapkan apabila data yang digunakan memiliki dimensi yang besar sehingga PCA dihadirkan dalam sistem ini. Penggunaan MFCC pun dikarenakan sudah *common* dalam bidang pengolahan data suara [9]. Dengan performansi yang baik, teknologi *Speech Recognition* diharapkan dapat digunakan untuk mengenali Bahasa Arab khususnya untuk

pengecekan bacaan Al-Qur'an oleh seseorang sehingga membaca Al-Qur'an tidak lagi menjadi tantangan yang menyulitkan bagi banyak kalangan. Pengecekan bacaan tersebut dapat dimulai dengan membangun sistem yang dapat mengenali ucapan huruf hijaiyah bertanda baca.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, maka perumusan masalah dalam pembuatan tugas akhir ini adalah sebagai berikut:

1. Bagaimana cara membangun sistem untuk melakukan klasifikasi terhadap data ucapan huruf hijaiyah bertanda baca?
2. Bagaimana menganalisis kinerja performansi hasil klasifikasi oleh sistem yang dibangun?

1.3 Tujuan

Berdasarkan perumusan masalah di atas, maka tujuan pembuatan tugas akhir ini adalah sebagai berikut:

1. Mengimplementasikan MFCC sebagai metode ekstraksi ciri dan KNN sebagai *classifier* untuk melakukan klasifikasi terhadap data ucapan huruf hijaiyah bertanda baca.
2. Menganalisis kinerja MFCC sebagai metode ekstraksi ciri dan KNN sebagai *classifier* berdasarkan *F1-measure* klasifikasi dari sistem yang dibangun.

1.4 Metodologi Penyelesaian Masalah

Metodologi penyelesaian masalah yang akan dilakukan pada penulisan tugas akhir ini adalah:

1. Kajian Pustaka
   Kajian pustaka yang dilakukan berupa pencarian dan pengumpulan jurnal atau buku yang dapat menjadi dasar referensi pembuatan tugas akhir. Tujuan dari dilakukannya tahap ini adalah memahami permasalahan dan konsep tugas akhir yang sedang dikerjakan serta mendapatkan referensi metode yang akan digunakan untuk menyelesaikan permasalahan.
2. Pengumpulan dan Analisis Data
   Pada tahap ini dilakukan pengumpulan data yang diperlukan untuk proses implementasi sistem. Data pada tugas akhir ini berupa data suara pembacaan huruf-huruf hijaiyah beserta harakatnya.
3. Analisis dan Perancangan Sistem
   Setelah melakukan analisis terhadap permasalahan dan referensi yang didapatkan dari kegiatan sebelumnya, sistem yang akan dibangun mulai dirancang. Rancangan sistem dapat dibuat berupa *flowchart* dan *block diagram* agar lebih mudah untuk dipahami. Pada bagian ini pula akan diberikan gambaran terkait proses algoritma yang digunakan.
4. Implementasi
   Dari rancangan sistem yang telah dibuat, sistem akan coba diimplementasikan. Pada tugas akhir ini, sistem akan dibangun menggunakan Python.

5. Analisis Hasil Pengujian
   Hasil yang diperoleh pada tahap implementasi akan dianalisis. Metode yang digunakan pun akan ditinjau dan dilihat hasilnya. Dalam tahap ini pula akan dilakukan proses pengukuran tingkat keberhasilan sistem yang dibangun untuk selanjutnya ditarik kesimpulan dari pembuatan tugas akhir ini.
6. Pembuatan Laporan Tugas Akhir
   Selanjutnya, hasil implementasi dan analisis akan didokumentasikan menjadi sebuah dokumen yaitu Laporan Tugas Akhir. Pada laporan ini pula akan disertakan dokumen-dokumen pendukung berjalannya implementasi tugas akhir ini.

1.5 Sistematika Penulisan

Laporan Tugas Akhir ini disusun dengan sistematika penulisan sebagai berikut:

1. Pendahuluan
   Bab ini berisi penjelasan mengenai latar belakang pemilihan masalah dalam penelitian ini, perumusan masalah, tujuan, metodologi penyelesaian masalah, dan sistematika penulisan.
2. Studi Literatur
   Bab ini berisi penjelasan landasan teori yang mendukung penelitian pada tugas akhir ini. Adapun landasan teori yang dijelaskan meliputi istilah-istilah yang digunakan pada laporan ini, penelitian-penelitian terkait, dan teori lain yang digunakan untuk menyusun laporan ini.
3. Perancangan Sistem
   Bab ini memaparkan analisis kebutuhan sistem dan rancangan sistem yang digunakan serta tahapan-tahapan yang diperlukan untuk implementasi sistem.
4. Pengujian dan Analisis
   Bab ini menguraikan hasil pengujian terhadap sistem yang dibangun beserta analisis terhadap hasil pengujian yang telah dilakukan.
5. Kesimpulan dan Saran
   Bab ini memaparkan kesimpulan dari hasil pengujian dan analisis yang telah dilakukan beserta saran untuk penelitian ini kedepannya.

BAB 2 STUDI LITERATUR

Bagian ini akan memaparkan teori yang mendukung penelitian pada tugas akhir ini. Adapun teori yang dipaparkan meliputi metode yang digunakan, istilah-istilah, penelitian terkait dan teori lain yang digunakan untuk mendukung penelitian ini.

2.1 *Speech Recognition*

Speech Recognition adalah teknologi yang memungkinkan komputer untuk mengidentifikasikan kata-kata yang diucapkan oleh seseorang menggunakan *microphone* atau *telephone*. *Speech Recognition* dapat digunakan untuk banyak hal dan membantu orang-orang cacat untuk dapat berinteraksi dengan lingkungannya [3].

Dalam Bahasa lain, *Speech Recognition* adalah proses mengubah sinyal suara yang ditangkap ke dalam bentuk rangkaian kata dalam bentuk teks. Teknologi *Speech Recognition* telah banyak digunakan, contoh aplikasi yang mengimplementasikan *Speech Recognition* adalah: aplikasi pembelajaran Bahasa asing, aplikasi untuk memperbaiki bacaan Al-Qur'an seseorang, aplikasi dengan antar muka berbasis suara dan aplikasi penanda *database* menggunakan audio atau video.

Speech Recognition juga dapat dikatakan sebagai bentuk konversi gelombang akustik ke bentuk tulisan dengan informasi yang ekivalen. Gambar 2.1 menunjukkan proses dasar yang terjadi pada sistem *Speech Recognition*.

Gambar 2.1. Skema umum Speech Recognition [3]

Speech signal processing menggambarkan suatu operasi yang dilakukan terhadap data sinyal suara yang diperoleh. Operasi yang dilakukan dapat berupa penghilangan *noise,* analisis *spectral,* proses digitalisasi dan lain sebagainya. Sedangkan untuk tahap *feature extraction* menggambarkan sebuah tahap pengenalan pola data yang didapatkan. Pada tahap ini, data *input* akan dibentuk untuk kemudian dapat dikenali oleh *classifier* [6].

Speech Recognition dapat dibedakan berdasarkan jenis *Speaker* dan jenis *Speech*. Berdasarkan jenis *Speaker*, *Speech Recognition* dapat dibagi menjadi dua, yaitu *Speaker-Dependent* dan *Speker-Independent*, sedangkan berdasarkan jenis *Speech*, *Speech Recognition* dapat dibagi menjadi dua, yaitu: *Isolated-Word* dan *Continuous Speech*.

Sistem dengan karakteristik *Speaker-Dependent* merupakan sistem yang hanya dapat mengenali suara yang berasal dari orang yang sama, sedangkan *Speaker-Independent* merupakan sistem yang dapat mengenali suara orang yang berbeda. Untuk sistem dengan karakteristik *Isolated-Word*, sistem hanya dapat menerima data masukan berupa satu ucapan sehingga dibutuhkan kondisi diam pada awal dan akhir kata. Sedangkan untuk sistem dengan karakteristik *Continuous Speech*, sistem dapat menerima data masukan lebih

dari satu kata sekaligus. Pada tugas akhir ini, tipe yang digunakan adalah *Isolated-Word* dan *Speaker-Dependent* serta *Isolated-Word* dan *Speaker-Independent*. [7]

2.2 *Arabic Speech Recognition* (ASR)

Pada dasarnya, pengembangan *Speech Recognition* untuk Bahasa Arab merupakan bentuk kolaborasi dari berbagai cabang ilmu. Beberapa cabang ilmu tersebut antara lain: terkait *Arabic phonetic*, lalu juga melibatkan cabang ilmu teknik pemrosesan suara dalam Bahasa Arab, dan *Natural Language Processing*. Dalam sub-bab ini akan dipaparkan beberapa penelitian terkait *Speech Recognition* untuk Bahasa Arab. Pada 2001, Al-Otaibi membangun sebuah sistem rekognisi untuk Bahasa Arab. Ia menyiapkan dataset berupa rekaman suara MSA dan ia juga mengajukan teknik pelabelan pada data suara. ASR tersebut dibangun menggunakan *Hidden Markov Model* (HMM) tool kit. Selanjutnya pada 2008, Hyassat dan Abu Zitar membangun sistem rekognisi untuk data suara Bahasa Arab menggunakan Sphinx4. Mereka juga membuat toolkit otomatis untuk membangun kamus Al-Qur'an dan standar-standar dalam Bahasa Arab. Hasil dari penelitian ini adalah tiga buah tulisan yang diberi nama Holy Qur'an corpus HQC-1 dengan durasi sekitar 18,5 jam; Command and Control Corpus CAC-1 dengan durasi 1,5 jam dan *Arabic Digit Corpus* dengan durasi kurang dari satu jam [4].

Iqbal et al. pun pernah melaporkan penelitiannya terkait segmentasi dan identifikasi vokal Bahasa Arab menggunakan transisi formant yang terjadi selama pembacaan Al-Qur'an. Akurasi sistem yang dibangun pun mencapai 90% terhadap 1000 data vokal yang digunakan. Selain itu, Razak et al. juga pernah melakukan investigasi terhadap pembacaan ayat Al-Qur'an dengan melakuakn ekstraksi fitur menggunakan pendekatan *Mel-Frequency Cepstral Coefficient* (MFCC) [5].

2.3 Bahasa Arab dan Huruf Hijaiyah

Bahasa arab adalah Bahasa semitik dan salah satu dari Bahasa tertua di dunia. Bahasa arab merupakan Bahasa ke-lima yang sering digunakan di dunia saat ini [3]. Bahasa arab memiliki dua bentuk utama, Bahasa Arab Standar dan Bahasa Arab Dialek. Bahasa Arab standar terdiri atas Bahasa Arab Klasik dan Bahasa Arab Modern atau yang lebih sering dikenal dengan istilah MSA (*Modern Standard Arabic*). Sedangkan Bahasa Arab Dialek terdiri atas banyak bentuk termasuk seluruh bentuk Bahasa Arab yang diucapkan dalam kehidupan sehari-hari.

Dalam MSA terdapat 38 jenis fonem. Fonem tersebut terdiri atas 29 konsonan asli, 3 konsonan asing dan 6 vokal. Sedangkan pada Bahasa Arab Standar terdiri atas 34 fonem yang terbagi menjadi 6 vokal dan 28 konsonan. Fonem vokal adalah bunyi yang diproduksi tanpa penghambat aliran udara yang melalui saluran keluarnya suara sedangkan pada konsonan akan ada penghambat yang menyebabkan adanya *noise* pada suara dengan amplitudo yang lebih lemah. Fonem sendiri adalah unit terkecil dalam suara yang mengindikasikan adanya perbedaan arti, kata atau kalimat. Fonem pada Bahasa Arab mengandung dua kelas yang berbeda yaitu *pharyngeal* dan *emphatic.* Untuk penjelasan lebih lanjut terkait *pharyngeal* dan *emphatic* dapat dilihat pada referensi ke [6].

Suku kata dalam Bahasa Arab dapat berupa CV, CVC, CVV, CVVC, CVVCC dan CVCC dimana C menandakan konsonan dan V menandakan vokal panjang atau pendek. Ucapan-ucapan dalam Bahasa Arab hanya dapat dimulai dengan konsonan. Terdapat beberapa aturan dalam suku kata Arab yaitu:

1. Suku kata harus diawali dengan konsonan dan diikuti dengan vokal.
2. Suku kata tidak pernah diawali dengan dua konsonan.
3. Tidak ada fonem suku kata dalam Bahasa Arab.

Bahasa Arab memiliki 28 huruf yang disebut dengan Huruf Hijaiyah. Seluruh huruf hijaiyah tersebut dapat dilihat pada Tabel 2.1.

Tabel 2.1. Huruf Hijaiyah Bertanda Baca

Kelas	Hijaiyah	Lafal	Kelas	Hijaiyah	Lafal	Kelas	Hijaiyah	Lafal	Kelas	Hijaiyah	Lafal
1	اَ	a	43	دَ	da	85	ضَ	dho	127	كَ	ka
2	اً	an	44	دً	dan	86	ضً	dhon	128	كً	kan
3	اِ	i	45	دِ	di	87	ضِ	dhi	129	كِ	ki
4	اٍ	in	46	دٍ	din	88	ضٍ	dhin	130	كٍ	kin
5	اُ	u	47	دُ	du	89	ضُ	dhu	131	كُ	ku
6	اٌ	un	48	دٌ	dun	90	ضٌ	dhun	132	كٌ	kun
7	بَ	ba	49	ذَ	dza	91	طَ	tho	133	لَ	la
8	بً	ban	50	ذً	dzan	92	طَ	thon	134	لً	lan
9	بِ	bi	51	ذِ	dzi	93	طِ	thi	135	لِ	li
10	بٍ	bin	52	ذٍ	dzin	94	طٍ	thin	136	لٍ	lin
11	بُ	bu	53	ذُ	dzu	95	طُ	thu	137	لُ	lu
12	بٌ	bun	54	ذٌ	dzun	96	طٌ	thun	138	لٌ	lun
13	تَ	ta	55	رَ	ro	97	ظَ	dzo	139	مَ	ma
14	تً	tan	56	رً	ron	98	ظَ	dzon	140	مً	man
15	تِ	ti	57	رِ	ri	99	ظِ	dzi	141	مِ	mi
16	تٍ	tin	58	رٍ	rin	100	ظٍ	dzin	142	مٍ	min
17	تُ	tu	59	رُ	ru	101	ظُ	dzu	143	مُ	mu
18	تٌ	tun	60	رٌ	run	102	ظٌ	dzun	144	مٌ	mun
19	ثَ	tsa	61	زَ	za	103	عَ	‘a	145	نَ	na
20	ثً	tsan	62	زً	zan	104	عً	‘an	146	نً	nan
21	ثِ	tsi	63	زِ	zi	105	عِ	‘i	147	نِ	ni
22	ثٍ	tsin	64	زٍ	zin	106	عٍ	‘in	148	نٍ	nin
23	ثُ	tsu	65	زُ	zu	107	عُ	‘u	149	نُ	nu
24	ثٌ	tsun	66	زٌ	zun	108	عٌ	‘un	150	نٌ	nun
25	جَ	ja	67	سَ	sa	109	غَ	gho	151	وَ	wa
26	جً	jan	68	سً	san	110	غً	ghon	152	وً	wan
27	جِ	ji	69	سِ	si	111	غِ	ghi	153	وِ	wi
28	جٍ	jin	70	سٍ	sin	112	غٍ	ghin	154	وٍ	win
29	جُ	ju	71	سُ	su	113	غُ	ghu	155	وُ	wu
30	جٌ	jun	72	سٌ	sun	114	غٌ	ghun	156	وٌ	wun
31	حَ	ha	73	شَ	sya	115	فَ	fa	157	هَ	ha
32	حً	han	74	شً	syan	116	فً	fan	158	هً	han
33	حِ	hi	75	شِ	syi	117	فِ	fi	159	هِ	hi
34	حٍ	hin	76	شٍ	syin	118	فٍ	fin	160	هٍ	hin
35	حُ	hu	77	شُ	syu	119	فُ	fu	161	هُ	hu
36	حٌ	hun	78	شٌ	syun	120	فٌ	fun	162	هٌ	hun
37	خَ	kho	79	صَ	sho	121	قَ	qo	163	يَ	ya
38	خً	khon	80	صً	shon	122	قً	qon	164	يً	yan
39	خِ	khi	81	صِ	shi	123	قِ	qi	165	يِ	yi
40	خٍ	khin	82	صٍ	shin	124	قٍ	qin	166	يٍ	yin
41	خُ	khu	83	صُ	shu	125	قُ	qu	167	يُ	yu
42	خٌ	khun	84	صٌ	shun	126	قٌ	qun	168	يٌ	yun

Huruf-huruf hijaiyah tersebut memiliki delapan tanda baca, yaitu: *fathah, kasrah, dhammah, fathahtain, kasrahtain, dhammahtain, tasydid* dan *sukun*. Namun sistem ini hanya menggunakan 6 tanda baca, yaitu: *fathah, kasrah, dhammah, fathahtain, kasrahtain, dhammahtain*.

2.4 *Mel-Frequency Cepstrum Coefficients* (MFCC)

Salah satu faktor penting agar sistem mampu menghasilkan *output* yang baik adalah pemilihan metode ekstraksi ciri yang tepat. MFCC telah *common* digunakan dalam menangani kasus pengolahan data suara. [9] MFCC memiliki beberapa tahapan dalam memroses data suara. Tahapan tersebut meliputi: *Pre-emphasis*, *Framing*, *Windowing*, *Fast Fourier Transform*, *Mel-Filter Bank Processing*, *Discrete Cosine Transform* (DCT) serta *Delta Energy* dan *Delta Spectrum*.

1. *Pre-emphasis*

Pada tahap ini, sinyal suara akan melewati suatu filter yang akan memperkuat energi pada data yang memiliki frekuensi tinggi. [10] Hal ini disebabkan sinyal suara dengan frekuensi tinggi memiliki *magnitude* atau energi yang lebih kecil dibandingkan dengan sinyal suara berfrekuensi rendah. Adapun persamaan yang digunakan pada tahap ini dapat dilihat pada persamaan (2).

$$p(t) = x(t) - \alpha x(t-1) \tag{2}$$

Keterangan:
α = koefisien filter, biasanya menggunakan nilai 0,95 dan 0,97 [11]
$x(t)$ = sinyal suara ke-t

$p(t)$ = hasil sinyal yang telah dilakukan *emphasized* ke-t

2. *Framing*

Pada tahap ini, dilakukan proses pembagian sinyal menjadi beberapa *frame* kecil dengan panjang *frame* berkisar antara 20 sampai 40 *milliseconds* (ms). Hal ini karena sinyal dapat diasumsikan stabil dalam kisaran rentang tersebut. Adapun ukuran *frame* yang sering digunakan adalah 25 ms dan *stride* sebesar 10 ms. [11]

3. *Windowing*

Windowing dilakukan untuk mengurangi diskontinuitas pada segmen sinyal yang telah melewati proses *Framing*. Tipe *windowing* yang paling sering digunakan adalah *Hamming Window* [10]. Persamaan *Hamming Window* dapat dilihat pada persamaan (3).

$$W(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \qquad 0 \le n \le N-1 \tag{3}$$

Selanjutnya, sinyal yang menjadi *input* pada tahap *windowing* akan melewati proses dengan persamaan yang dapat dilihat pada persamaan (4).

(4)

$$Y(n) = X(n)\ x\ W(n)$$

Keterangan:
N = jumlah *frame* pada suatu data sinyal suara
n = indeks *frame*
$W(n)$ = persamaan *Hamming Window*
$X(n)$ = sinyal input *frame* ke-n pada tahap ini
$Y(n)$ = sinyal *frame* ke-n yang telah dikalikan dengan persamaan *Hamming Window* ke-n

4. *Fast Fourier Transform*

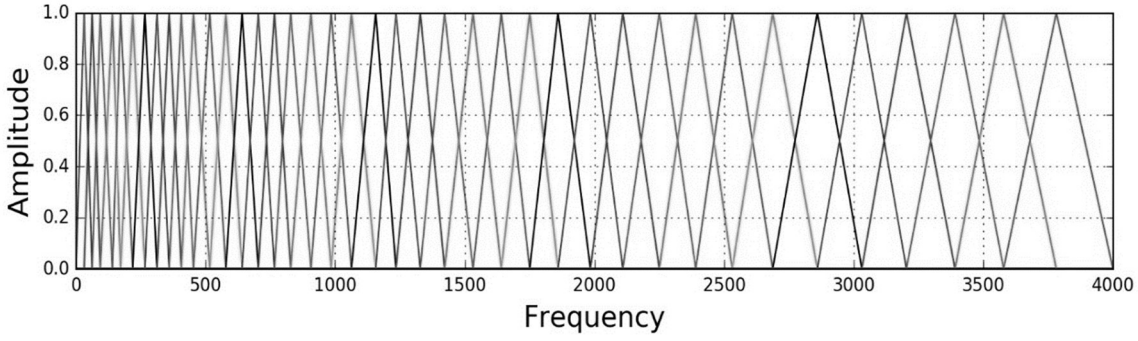
Untuk melakukan konversi pada setiap *frame* dari domain waktu menjadi domain frekuensi dilakukan proses *Fast Fourier Transform* (FFT) [12]. Persamaan FFT dapat dilihat pada persamaan (5).

$$F(w) = \sum_{n=0}^{N-1} Y(n) e^{-j2\frac{\pi}{N}wn} \quad (5)$$

Keterangan:
N = jumlah *frame* pada suatu data sinyal suara
$Y(n)$ = sinyal input *frame* ke-n pada tahap ini
w = panjang DFT, di mana $0 < w < N - 1$
$F(w)$ = Hasil FFT pada DFT ke-w

5. *Mel-Filter Bank Processing*

Spektrum yang merupakan hasil dari tahap sebelumnya mengandung informasi pada setiap frekuensi, sedangkan frekuensi suara yang terdengar oleh manusia berada di atas 1000 Hz. Oleh karena itu, digunakan frekuensi linier untuk frekuensi di bawah 1 KHz dan frekuensi logaritmik untuk frekuensi di atas 1 KHz [11].



Gambar 2.2 Filter Bank pada Mel Scale [11]

Adapun perhitungan *Mel-points* dapat dilakukan dengan menggunakan persamaan (6).

$$m = (2595\ x\ \log_{10}(1 + f)\ 700) \quad (6)$$

Selanjutnya, untuk melakukan konversi dari *Mel-points* ke satuan Hz dapat diperoleh melalui persamaan (7).

$$f = 700(10^{\frac{m}{2595}} - 1) \tag{7}$$

Keterangan:
m = jumlah *Mel-points*
f = frekuensi dari setiap *Mel-points*

Untuk melakukan perhitungan pada *Triangular Window* sehingga mendapatkan *Filterbank* dapat dilakukan dengan menggunakan persamaan (8).

$$H_m(k) = \begin{cases} 0, k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)}, f(m-1) \le k \le f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)}, f(m) \le k \le f(m+1) \\ 0, k > f(m+1) \end{cases} \tag{8}$$

Keterangan:
$f(m)$ = frekuensi pada indeks *filter* ke-*m*
m = indeks *filter*
k = titik frekuensi pada *Triangular Window*

6. Discrete Cosine Transform

Pada tahap ini, dilakuakn proses konversi nilai log *Mel Spectrum* ke domain waktu. Pada tahap ini juga lah diperoleh koefisien MFCC [12]. Perhitungan DCT dapat dilakukan dengan menggunakan persamaan (9).

$$y_t(k) = \sum_{m=1}^{M} \log(|y_t(m)|) \cos\left(k(m - 0.05)\frac{\pi}{M}\right), \quad k = 0, \dots, j \tag{9}$$

Keterangan:
$y_t(m)$ = frekuensi mel ke-*m*
M = jumlah *frame*
m = indeks *frame*

2.5 *Principal Component Analysis* (PCA)

Principal Component Analysis (PCA) adalah metode parametric untuk mengestrak informasi yang relevan dari dataset yang besar. PCA juga dapat digunakan untuk reduksi dimensi dari data yang kompleks ke bentuk data dengan dimensi yang lebih kecil. [13]

Langkah-langkah yang dapat dilakukan untuk mengimplementasikan PCA adalah sebagai berikut [13]:

1. Lakukan proses pengurangan untuk setiap dimensi dengan nilai rata-rata dari dimensi tersebut
2. Lakukan perhitungan matriks *covariance*

$$cov(x,y) = \sum_{i=1}^{n} \frac{(X_i - \bar{X})\,(Y_i - \bar{Y})}{(n-1)} \quad (10)$$

Keterangan:
X_i = elemen ke-i pada dimensi *X*
Y_i = elemen ke-i pada dimensi *Y*
$\bar{X}$ = *mean* dimensi *X*
$\bar{\bar{Y}}$ = *mean* dimensi *Y*

3. Lakukan perhitungan nilai eigen dan vektor eigen dari matriks *covariance* yang telah diperoleh.
4. Ambil vektor eigen yang bersesuaian dengan nilai eigen terbesar sebanyak *x* *principal components*. Nilai *x* dapat ditentukan oleh pengguna.
5. Lakukan reduksi dimensi dan bentuk *feature vector* dengan cara mengalikan dataset dengan vektor eigen untuk mengubah domain dataset menjadi domain eigen

2.6 *K-Nearest Neighbor* (KNN)

Dalam sistem pengenalan pola, *K-Nearest Neighbor* adalah metode non-parametrik yang digunakan untuk menangani kasus klasifikasi dan regresi. Dalam kedua kasus tersebut, akan diambil sebanyak K data *training* terdekat untuk menentukan label dari data *testing* [8].

k-Nearest Neighbor
Classify (**X**, **Y**, *x*) // **X**: *Training* data, **Y**: class labels of **X**, *x*: unknown sample
for *i* = 1 **to** m **do**
Compute distance $d(\mathbf{X}_i, x)$
end for
Compute set *I* containing indices for the *k* smallest distances $d(\mathbf{X}_i, x)$.
return majority label for $\{\mathbf{Y}_i$ where $i \in I\}$

Gambar 2.3. Pseudocode KNN [14]

Pada fase klasifikasi, nilai K akan ditentukan oleh pengguna dan data *testing* akan diklasifikasikan berdasarkan nilai label yang paling sering muncul pada K data *training* yang telah dicari. Pengukuran jarak terdekat yang sering digunakan pada algoritma ini untuk menangani variabel yang bersifat kontinu adalah dengan menggunakan perhitungan Jarak *Euclidean*. [8] Adapun persamaan perhitungan Jarak *Euclidean* dapat dilihat pada persamaan (1).

$$d(p,q) = \sqrt{\sum_{i=1}^{N} (q_i - p_i)^2} \quad (1)$$

Keterangan:
$d(p, q)$ = Jarak Euclidean antara vektor p dan vektor q
N = Jumlah data pada kedua vektor
q_i = data ke-i pada vektor q
p_i = data ke-i pada vektor p

2.7 Pengukuran Performansi Sistem

Pengukuran performansi sistem pada tugas akhir ini diukur dengan menggunakan *f1-measure* dan akurasi. *F1-measure* dihitung menggunakan persamaan (11).

$$F1 - Measure_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \tag{11}$$

Precision adalah jumlah hasil positif yang benar dibagi dengan jumlah semua hasil positif dan *recall* merupakan jumlah hasil positif yang benar dibagi dengan jumlah hasil positif yang seharusnya dikembalikan. Adapun perhitungan *precision* dan *recall* dapat dilakukan dengan menggunakan persamaan (12) dan (13).

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \tag{12}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{13}$$

Keterangan:
i = indeks sampel pada kelas ke-i

Perhitungan *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* dapat diperoleh menggunakan confusion matrix.

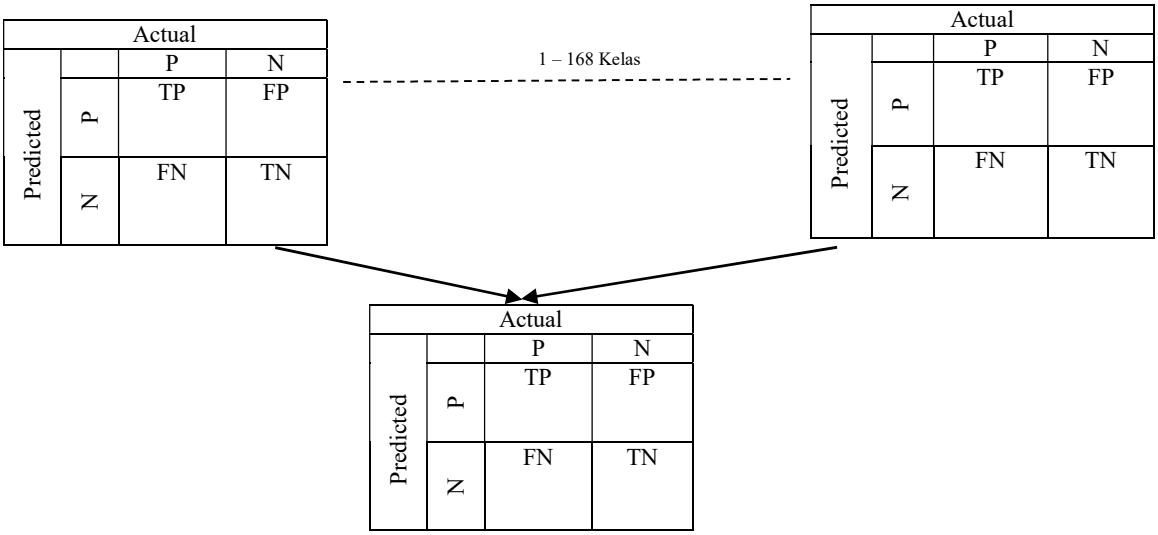
		Actual Value (as confirmed by experiment)	
		Positives	Negatives
Predicted Value (predicted by the test)	Positives	True Positive	False Positive
	Negatives	False Negative	True Negative

Gambar 2.4. Confusion Matrix [15]

Nilai aktual adalah label data sedangkan nilai prediksi adalah hasil keluaran dari sistem yang dibangun. Jika nilai aktual positif dan nilai prediksi positif maka akan diklasifikasikan sebagai *True Positive*, jika nilai aktual negatif dan nilai prediksi positif maka akan diklasifikasikan sebagai *False Positive*, jika nilai aktual positif dan nilai

prediksi negatif, maka akan diklasifikasikan sebagai *False Negative*, dan jika nilai aktual negatif dan nilai prediksi negatif, maka akan diklasifikasikan sebagai *True Negative*.

Perhitungan performansi dilakukan dengan menggunakan *Micro Average F1-Score* dimana seluruh *confusion matrix* untuk setiap kelas digabungkan dan kemudian dihitung nilai *F1-Score* terhadap *confusion matrix* gabungan tersebut. Penggabungan disini bermaksud untuk setiap elemen TP, FP, FN dan TN pada tiap *confusion matrix* per kelas akan dijumlahkan sehingga menghasilkan masing-masing satu nilai TP, FP, FN dan TN pada *confusion matrix* gabungan.



Gambar 2.5. Ilustrasi Micro Average F1-Score

Adapun perhitungan *Micro Average F1-Score* menggunakan rumus *precision, recall* dan *F1-Score* yang sama dengan rumus yang ditunjukkan pada persamaan (11), (12), (13). Namun, nilai TP, FP, FN dan TN diambil dari *confusion matrix* gabungan.

$$F1 - Measure = 2 \times \frac{precision \times recall}{precision + recall} \tag{14}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{15}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{16}$$