

APLIKASI PENGENALAN PENUTUR PADA IDENTIFIKASI SUARA PENELEPON MENGGUNAKAN MEL-FREQUENCY CEPSTRAL COEFFICIENT DAN VECTOR QUANTIZATION (Studi Kasus : Layanan Hotline Universitas Pembangunan Nasional “Veteran” Yogyakarta)

Muhammad Fahim Rasyid ⁽¹⁾, Herlina Jayadianti ⁽²⁾, Herry Sofyan ⁽³⁾

Jurusan Teknik Informatika Fakultas Teknik Industri UPN “Veteran” Yogyakarta

Jl. Babarsari 2 Tambakbayan Yogyakarta

e-mail : fahim.rasyid45@gmail.com, herlinajayadianti@gmail.com, herrysofyan@gmail.com

Abstract

Everyone is able to access Universitas Pembangunan Nasional “Veteran” Yogyakarta’s hotline services. Staff and employees using the services to share informations with other departments in the rectorate building. Caller have to identified by the officer to comunicate with intended department. The term of identity such as name, position, and origin of the majors or section are asked during identification process. There is no physical and computer database record of identification results yet. These problems cause hotline services officer do not have any evidiance for incorrect identification of the caller. This research focus to decrease these problems using speaker recognition technology. Voice frequency was extracted using Mel-Frequency Cepstral Coefficient (MFCC) into Mel Frequency Cepstrum Coefficients. Employees training data coefficients compared to caller voice coefficients using Vector Quantization (VQ) method. Speaker recognition application was identified caller voice with 80% accuracy at threshold value of 25.

Keywords: *Hotline, Speaker Recognition, Frequency, Mel-Frequency Cepstral Coefficient (MFCC), Vector Quantization (VQ)*

Abstrak

Layanan *hotline* Universitas Pembangunan Nasional “Veteran” Yogyakarta merupakan layanan yang dapat digunakan oleh semua orang. Layanan tersebut digunakan dosen dan pegawai untuk berbagi informasi dengan bagian-bagian yang berlokasi di gedung rektorat. Penelepon dapat berkomunikasi dengan bagian yang dituju apabila telah teridentifikasi oleh petugas layanan *hotline*. Terminologi identitas yang terdiri dari nama, jabatan serta asal jurusan atau bagian ditanyakan saat proses identifikasi. Tidak terdapat catatan hasil identifikasi penelepon baik dalam bentuk fisik maupun basis data yang terekam pada komputer. Hal tersebut mengakibatkan tidak adanya dokumentasi yang dapat dijadikan barang bukti untuk menindak lanjuti kasus kesalahan identifikasi. Penelitian ini fokus untuk mengurangi resiko kesalahan identifikasi penelepon menggunakan teknologi *speaker recognition*. Frekuensi suara diekstraksi menggunakan metode *Mel-Frequency Cepstral Coefficient (MFCC)* sehingga dihasilkan nilai *Mel Frequency Cepstrum Coefficients*. Nilai *Mel Frequency Cepstrum Coefficients* dari semua data latih suara pegawai Universitas Pembangunan Nasional “Veteran” Yogyakarta kemudian dibandingkan dengan sinyal suara penelpon menggunakan metode *Vector Quantization (VQ)*. Aplikasi pengenalan penutur mampu mengidentifikasi suara penelepon dengan tingkat akurasi 80% pada nilai ambang (*threshold*) 25.

Kata Kunci : *Hotline, Speaker Recognition, Frekuensi, Mel-Frequency Cepstral Coefficient (MFCC), Vector Quantization (VQ)*

PENDAHULUAN

Hotline merupakan saluran telepon untuk pelayanan masyarakat yang membutuhkan informasi berkaitan dengan institusi maupun perusahaan tertentu. Layanan *hotline* juga dapat meningkatkan kinerja dan mobilitas suatu institusi dengan menjadi media penghubung antara karyawan maupun departemen. Teknologi dan perangkat keras yang digunakan yaitu telepon yang terhubung dengan komputer, serta jaringan internet. Identifikasi penelepon dilakukan dengan menanyakan data diri penelepon, yang kemudian dicocokkan dengan *database* melalui komputer. Universitas Pembangunan Nasional “Veteran” Yogyakarta merupakan salah satu institusi yang menyediakan layanan *hotline*.

Hotline Universitas Pembangunan Nasional “Veteran” Yogyakarta dikelola oleh bagian Hubungan Masyarakat. Layanan tersebut merupakan layanan untuk umum, sehingga semua orang yang membutuhkan informasi mengenai kampus dapat mengaksesnya. Dalam meningkatkan mobilitas dari kegiatan kampus, staff maupun dosen memanfaatkan layanan tersebut untuk berbagai keperluan. Dosen beserta staff Universitas Pembangunan Nasional “Veteran” Yogyakarta yang berlokasi di kampus 2 Babarsari menggunakan layanan *hotline* untuk berkomunikasi dengan bagian-bagian yang berlokasi di rektorat kampus 1 Condongcatut. Penerapan teknologi serta sistem yang berjalan pada layanan *hotline* Universitas Pembangunan Nasional “Veteran” Yogyakarta memiliki beberapa kekurangan. Telepon yang digunakan untuk menerima panggilan tidak terhubung dengan komputer, sehingga tidak dilakukan pengecekan data penelepon dengan *database*. Identifikasi penelepon dilakukan dengan menanyakan nama, jabatan beserta asal jurusan atau bagian. Proses identifikasi tersebut beresiko terjadinya ketidaksesuaian data penelepon yang disampaikan dibandingkan dengan identitas sebenarnya penelepon. Informasi-informasi yang bersifat rahasia universitas dapat diketahui oleh masyarakat luas dengan menelepon bermodalkan nama, jabatan, beserta jurusan atau bagian salah seorang staff maupun dosen. Proses identifikasi penelepon dengan menanyakan identitas penelepon oleh petugas saat layanan *hotline* membuat durasi waktu telepon menjadi semakin lama. Semakin lama waktu telepon mengakibatkan biaya telepon yang dibebankan pada penelepon semakin besar.

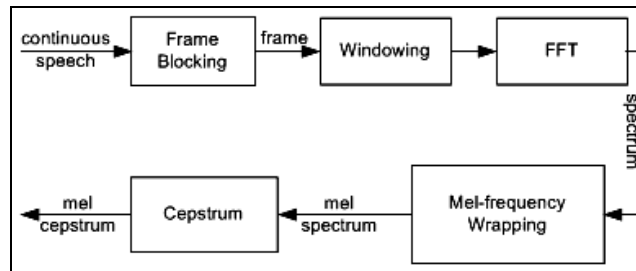
Berdasarkan permasalahan tersebut diperlukan sebuah sistem *hotline* yang dapat mengidentifikasi identitas penelepon secara akurat dan dicocokkan dengan data yang terdapat pada *database* staff dan dosen Universitas Pembangunan Nasional “Veteran” Yogyakarta. Keakuratan identifikasi seseorang dapat dilakukan dengan mengenali ciri khas suaranya. Pengenalan suara saat ini dapat dipelajari dan dikenali melalui proses pembelajaran *speaker recognition* yang merupakan bagian dari *voice recognition*. *Speaker recognition* menangkap frekuensi dari suara seseorang, kemudian dibandingkan dengan frekuensi suara data latih yang ada. Frekuensi suara diekstraksi melalui metode *Mel-Frequency Cepstral Coefficient (MFCC)* sehingga menghasilkan informasi berdasarkan sinyal suara yang diucapkan. Informasi dalam bentuk *Mel Frequency Cepstrum Coefficients* dari semua data latih dosen beserta staff Universitas Pembangunan Nasional “Veteran” Yogyakarta kemudian diklasifikasikan dan dibandingkan dengan sinyal suara dari penelpon yang masuk melalui metode *VQ (Vector Quantization)*. Proses klasifikasi, metode Algoritma *LBG (Linde Buzo Gray)* dipakai untuk membentuk *codebook* berdasarkan vektor ciri masing masing pengucap pada metode *Vector Quantization*

Speaker Recognition

Identifikasi penutur (*Speaker recognition*) dapat didefinisikan sebagai proses memilih penutur (pembicara) yang mempunyai karakteristik ciri suara yang mendekati sama dengan suara masukan. Berbagai-bagai bunyi ucapan (fonem) yang berbeda merupakan cara yang paling mudah untuk membedakan antara pengucap yang satu dengan yang lain. Fonem yang berbeda tersebut antara lain vokal, *diphthong*, semivokal, *unvoiced fricative*, *voiced fricative*, *voice* dan *unvoiced stop* (Apriyono, Wahyudi, & Hidayatno, 2011). Dalam hal ini dapat diimplementasikan untuk autentifikasi dengan menggunakan *password*, no kartu, kode PIN dan lainnya yang dapat memberikan hak akses ke suatu sistem bilamana suara pembicara diterima dengan teks yang sama (Prasetyawan, 2016).

Mel-Frequency Cepstral Coefficient

Ekstraksi ciri sinyal suara menggunakan *MFCC* didasarkan atas variasi bandwidth terhadap frekuensi pada telinga manusia yang merupakan filter, yang bekerja secara linier pada frekuensi rendah dan bekerja secara logaritmik pada frekuensi tinggi. metode *MFCC* merupakan metode yang paling sering digunakan dalam melakukan ekstraksi fitur suara dalam penelitian pengenalan identitas pembicara dan pengenalan perintah suara, karena mampu untuk mengekstraksi karakteristik sinyal suara secara jelas, yang relatif berbeda dari sifat saluran suara pembicara dan lebih efektif digunakan untuk pengenalan suara yang mengandung *noise* (Munggaran, Djamal, & Yuniarti, 2017). Diagram proses *MFCC* dapat dilihat pada Gambar 1.



Gambar 1 Diagram proses MFCC

Frame Blocking merupakan proses pembagian suara menjadi beberapa *frame* dan satu *frame* terdiri dari beberapa sampel. Proses ini diperlukan untuk membentuk sinyal suara yang *non-stationary* menjadi sinyal suara yang *quasi-stationary* sehingga dapat diubah dari domain waktu ke domain frekuensi dengan transformasi Fourier. Langkah selanjutnya adalah *windowing* setiap *frame*. Hal ini dilakukan untuk meminimalisasi diskontinuitas sinyal pada permulaan dan akhir setiap *frame*. Konsepnya adalah meruncingkan sinyal ke angka nol pada permulaan dan akhir setiap *frame*. Ada banyak fungsi window, namun yang sering digunakan dalam *Speaker identification* adalah *Hamming window*. Fungsi window ini menghasilkan *sidelobe level* yang tidak terlalu tinggi (kurang lebih -43 dB), selain itu *noise* yang dihasilkan pun tidak terlalu besar. Pengolahan selanjutnya menggunakan *Fast Fourier Transform* (FFT), yang mengubah setiap *frame* N sampel dari domain waktu ke domain frekuensi. FFT adalah algoritma cepat untuk menerapkan *Discrete Fourier Transform* (DFT) yang didefinisikan pada set N sampel x_n , sebagai berikut:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2j\pi kn}{N}}, k = 0, 1, 2, \dots, N-1 \quad \dots\dots\dots(1)$$

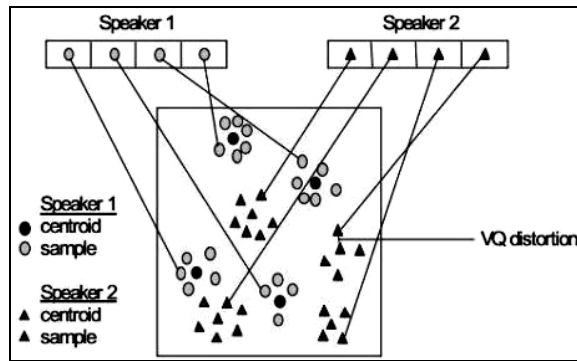
Dalam *mel-frequency wrapping*, sinyal hasil FFT dikelompokkan ke dalam berkas *filter triangular* ini. Maksud pengelompokkan ini adalah setiap nilai FFT dikalikan terhadap *gain filter* yang bersesuaian dan hasilnya dijumlahkan. Langkah terakhir dalam proses MFCC ini mengubah spektrum log *mel* kembali ke waktu. Hasilnya disebut MFCC (*Mel Frequency Cepstrum Coeficients*). *Cepstrum* adalah sebutan kebalikan untuk *spectrum*. *Cepstrum* biasa digunakan untuk mendapatkan informasi dari suatu sinyal suara yang diucapkan oleh manusia. Karena koefisien spektrum *mel* adalah bilangan real, maka konversinya ke domain waktu menggunakan *Discrete Cosine Transform* (DCT). Formula untuk menghitung koefisien MFCC itu adalah:

$$C_n = \sum_{k=1}^K (\log S_k) \cos \left[\frac{n \left(k - \frac{1}{2} \right) \pi}{K} \right] \quad \dots\dots\dots(2)$$

dengan S_k merupakan keluaran dari proses *filterbank* dan jumlah koefisien yang diharapkan (Prasetyawan, 2016).

Metode Virtual Quantization

Hasil ekstraksi MFCC(koefisien MFCC) untuk basisdata kemudian dibuat *codebook*-nya dengan menggunakan metode VQ. dengan menggunakan metode VQ diperoleh *codebook* yang isinya merepresentasikan koefisien MFCC yang telah diperoleh, yang berisi tentang informasi karakteristik dari masing-masing pengucap (Apriyono et al., 2011). *Vector Quantitation* (VQ) merupakan teknik dalam membangun ciri pencocokan (*feature matching*) pada *speaker identification*. Keuntungan dari menggunakan VQ adalah sederhana, mudah diimplementasikan dan akurasi tinggi. VQ adalah proses *mapping* vektor dari ruang vektor yang luas menjadi sejumlah daerah terhitung di ruangan itu. Kumpulan *codeword* disebut *codebook*. Jarak dari vektor akustik ke *codeword* terdekat pada *codebook* disebut *VQdistortion*. Diagram yang menggambarkan ciri pencocokan dengan vektor kuantisasi dapat dilihat pada Gambar 2.



Gambar 2 Konseptual vektor kuantisasi

Perhitungan *VQ-distortion* menggunakan *Euclidean distance*. Pada pengenalan pengucap dengan menggunakan metode VQ, jarak penyimpangan *Euclidean* ini digunakan untuk menghitung jarak penyimpangan antara masing-masing vektor ciri dengan *codeword* pada tiap-tiap *codebook*, sehingga dapat diketahui *codeword* mana yang memiliki jarak penyimpangan terdekat dengan vektor ciri (Apriyono et al., 2011). distorsi diukur berdasarkan minimisasi jarak *Euclidean* yang digunakan saat mencocokkan (*matching*) penutur tak dikenal dengan *database* penutur (Mustofa, 2008). Formula euclidean *distance* dituliskan sebagai berikut :

$$D_i = \sqrt{\sum_{i=0}^N (x_1 - x_2)^2} \dots\dots\dots(3)$$

dengan:

D_i = jarak terhadap tekstur i yang terkecil pada

x_1 = ciri dari tekstur yang diklasifikasikan

x_2 = ciri dari tekstur yang terdapat pada

Tekstur akan diklasifikasikan sebagai tekstur i apabila D_i merupakan jarak terkecil dibandingkan dengan jarak yang lainnya. Algoritma yang terkenal untuk membangun *VQcodebook*, yaitu algoritma LBG. Algoritma tersebut dilaksanakan oleh prosedur rekursif dengan langkah berikut ini:

1. Desain sebuah *codebook* vektor-1, yang merupakan centroid dari seluruh himpunan vektor training (tidak ada iterasi yang diperlukan disini).
2. Gandakan ukuran *codebook* dengan memisahkan setiap *current codebook* yn sesuai dengan aturan.

$$y_n^+ = y_n(1 + \varepsilon) \dots\dots\dots(4)$$

$$y_n^- = y_n(1 - \varepsilon) \dots\dots\dots(5)$$

dengan n bervariasi dari 1 sampai ukuran *current codebook* dan adalah parameter pemisahan ($\varepsilon = 0,01$).

3. *Nearest-Neighbor Search*: untuk setiap vektor training, temukan *codeword* pada *codebook* saat yang jaraknya terdekat (dalam hal pengukuran kesamaan) dan menetapkan vektor tersebut ke *cell* yang sesuai (terkait dengan *codeword* terdekat).
4. Perbarui centroid, memperbarui *codeword* dalam setiap *cell* menggunakan centroid dari vektor training yang ditetapkan ke *cell* tersebut.
5. Iterasi 1: ulangi langkah 3 dan 4 sampai jarak (*distance*) rata-rata turun dibawah ambang (*preset threshold*).
6. Iterasi 2: ulangi langkah 2, 3 dan 4 sampai ukuran *codebook* dirancang (Prasetyawan, 2016).

METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM

Metodologi penelitian yang digunakan adalah metode penelitian kuantitatif. Metode penelitian kuantitatif diterapkan untuk menjawab pertanyaan penelitian berdasarkan variabel yang bisa diukur dengan angka. Variabel pada penelitian ini adalah suara penutur yang diidentifikasi dan diukur berdasarkan nilai dari masing-masing data. Metode penelitian kuantitatif menarik kesimpulan dimulai dari teori-teori yang ada sebelumnya untuk kemudian diuji dengan data di lapangan. Data-data diperoleh dari proses pengumpulan data yang terdiri dari studi pustaka, observasi, dan wawancara. Metodologi pengembangan sistem yang digunakan untuk membangun sistem ini adalah *Guidelines for Rapid Application Engineering (GRAPPLE)*.

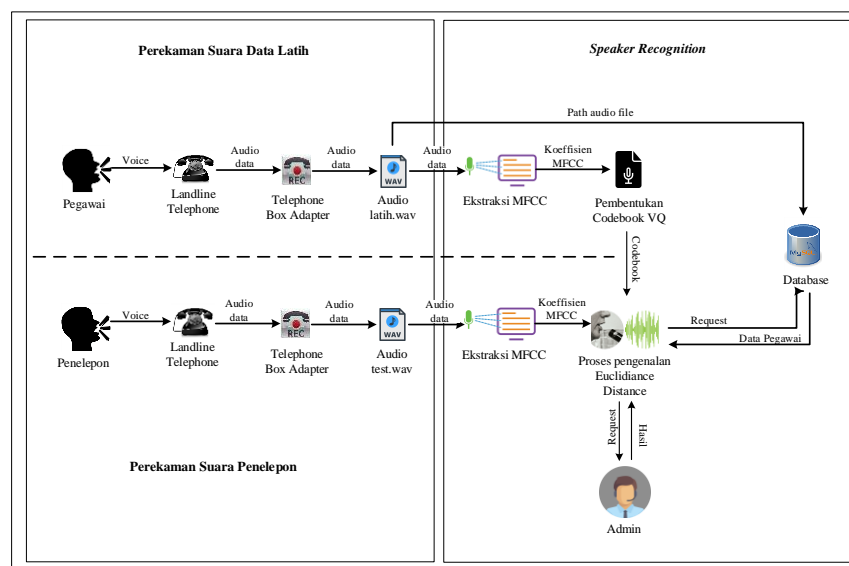
GRAPPLE merupakan metodologi yang fleksibel dan memberikan panduan yang jelas dalam proses pengembangan sistem. Proses dimulai dari tahapan *requirement gathering*, *analysis*, *design*, *development*, dan *deployment* (Schmuller, 2004). Kerangka kerja penelitian digambarkan pada gambar 4.

Perancangan (Design)

Aplikasi pengenalan penutur terdiri dari beberapa *design* (rancangan). Rancangan dibentuk berdasarkan analisa sistem yang akan dibangun. Perancangan meliputi arsitektur sistem, *unified modeling diagram* dan diagram alir (*flowchart*).

Arsitektur Sistem

Proses pengenalan suara dibagi menjadi dua proses perekaman. Proses perekaman pertama dilakukan pada pegawai selaku data latih. Perekaman kedua dilakukan pada penelepon yang masuk saat layanan *hotline*. Pada perekaman suara data latih, pegawai menginput suara pada telepon *landline* dalam kondisi tidak ada telepon masuk maupun keluar. Data *audio* dari telepon kemudian dialirkan ke *telephone adapter box* melalui kabel RJ-11. Data *audio* yang diterima kemudian diteruskan ke komputer melalui kabel *audio* yang dihubungkan dengan kabel *audio splitter*. Data *audio* yang diterima lalu direkam oleh aplikasi dan menghasilkan file *audio* latih dengan format *.wav*. Lokasi dari file kemudian disimpan ke dalam *database*, sedangkan data *audio* dari file data latih di ekstraksi oleh metode *MFCC* untuk menghasilkan nilai koefisien *MFCC*. Nilai koefisien *MFCC* lalu diproses oleh metode *VQ* untuk membentuk *codebook* data latih. Pada perekaman suara penelepon, data *audio* dari telepon kemudian dialirkan ke *telephone adapter box* melalui kabel RJ-11. Data *audio* yang diterima kemudian diteruskan ke komputer melalui kabel *audio* yang dihubungkan dengan kabel *audio splitter*. Data *audio* yang diterima lalu direkam oleh aplikasi dan menghasilkan file *audio* dengan format *.wav*. Data *audio* dari file di ekstraksi oleh metode *MFCC* untuk menghasilkan nilai koefisien *MFCC*. Nilai koefisien *MFCC* dari suara penelepon serta *codebook* data latih kemudian diproses oleh *Euclidean Distance* untuk pengenalan suara penelepon. Setelah dikenali, aplikasi meminta data pegawai yang dikenali kepada *database*. Data pegawai kemudian ditampilkan kepada admin melalui aplikasi. Arsitektur sistem digambarkan pada gambar 3.



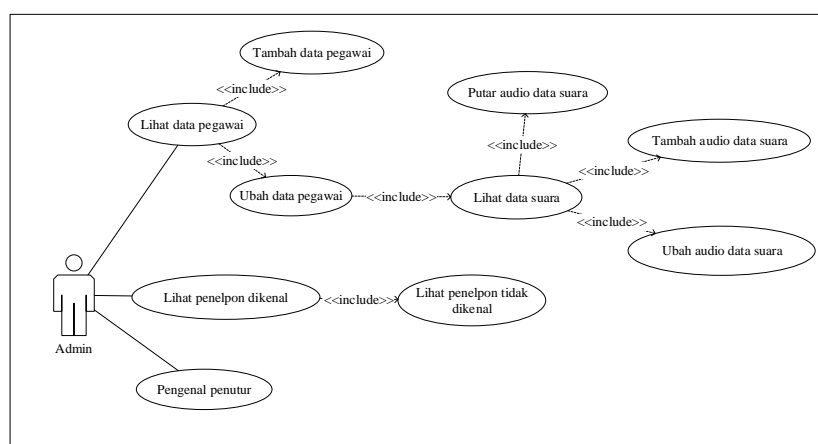
Gambar 3 Arsitektur Sistem

UML (Unified Modeling Language)

UML dapat dibuat model untuk semua jenis aplikasi perangkat lunak, dimana aplikasi tersebut dapat berjalan pada perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam program apapun (Sulistyorini, 2009). *UML* berfungsi untuk memodelkan sistem pada aplikasi pengenalan penutur. *UML* membantu dalam memvisualisasikan serta mendokumentasikan sistem yang dibuat pada penelitian ini. Pemodelan *UML* pada penelitian ini terdiri dari *use case diagram*, *class diagram*, *activity diagram*, dan *sequence diagram*.

a. Use Case Diagram

Use case diagram menggambarkan interaksi antara admin selaku aktor beserta *use case* dari aplikasi pengenalan suara. *Use case* menggambarkan fungsi dari setiap antarmuka pada aplikasi. *Use case diagram* dimodelkan pada tahapan analisis, dimana fungsi dari sistem dimodelkan dengan *use case*. Relasi *include* menjelaskan kepemilikan sebuah *use case* atas *use case* lain dari sistem. *Use case* pengenalan penutur merupakan fungsi utama sistem. *Use case* pengenalan penutur berisi fungsi dalam pengenalan penutur pada layanan *hotline*. *Use case* lihat memfasilitasi aktor untuk dapat melihat data pegawai yang terdaftar pada aplikasi pengenalan penutur. Pada tampilan data pegawai admin dapat berinteraksi dengan dua *use case* lain yaitu *use case* tambah data pegawai beserta ubah data pegawai. Admin dapat mengelola data suara melalui *use case* lihat data suara. Data penelepon yang telah diidentifikasi oleh aplikasi dapat dilihat melalui *use case* lihat penelepon dikenal. *Use case* digambarkan pada gambar 4.



Gambar 4 Use Case Diagram

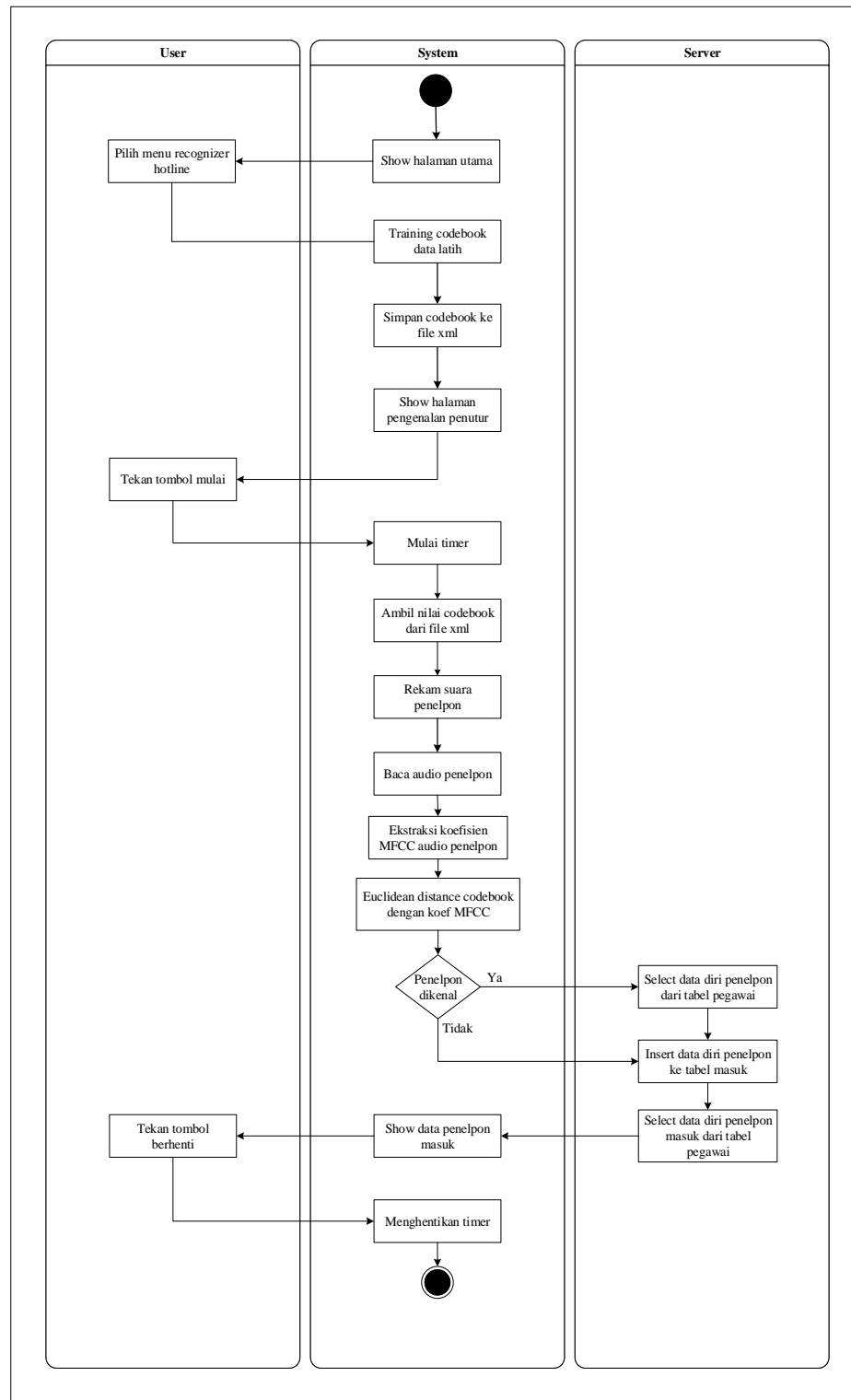
b. *Class Diagram*

Class diagram memodelkan hubungan antara semua *class* pada sistem. Masing masing *class* memiliki *method* dan atribut. *Class diagram* merupakan pemodelan berorientasi objek pada tahapan analisa. *Class diagram* menggambarkan perintah-perintah yang harus dikerjakan sesuai dengan informasi yang didapatkan pada tahapan *requirement gathering*. Terdapat tujuh *class* utama pada sistem, setiap *class* merupakan antarmuka aplikasi pada sistem. Diantara ketujuh *class* tersebut tidak terdapat interaksi, karena setiap fungsi yang dibutuhkan telah dideklarasikan pada setiap *class*. Penggunaan *PyQT4* sebagai *GUI* berpengaruh pada pemodelan *class diagram*, dimana ketujuh *class* utama terhubung ke *class PyQT4* yang mereferensikan semua variabel beserta fungsi dalam menampilkan antarmuka.

c. *Activity Diagram* Pengenalan Penutur

Activity diagram pengenalan penutur merupakan antarmuka yang berfungsi dalam pengidentifikasian penelepon masuk. Setelah tampilan awal aplikasi ditampilkan, kemudian *user* memilih menu *recognizer hotline*. Sebelum sistem menampilkan antarmuka pengenalan penutur, sistem melakukan *training codebook* data latih. Nilai *codebook* kemudian disimpan kedalam *file* penyimpanan *xml*. Sistem kemudian menampilkan antarmuka pengenalan penutur. Pada antarmuka pengenalan penutur, apabila terdapat telepon masuk, *user* dapat menekan tombol mulai untuk memulai mengenali suara penelepon. Sistem kemudian melakukan proses perekaman suara penelepon. Proses perekaman diawali dengan dipanggilnya fungsi *timer*. Melalui fungsi *timer*, diatur *interval* perekaman suara penelepon. Setelah *timer* dimulai, sistem mengambil nilai *codebook* data latih dari *file* penyimpanan *xml*. *Audio* suara rekaman tersebut kemudian diekstraksi untuk kemudian diubah menjadi koefisien *MFCC*. Proses ekstraksi data suara menggunakan metode *MFCC*. Koefisien *MFCC* lalu dibandingkan dengan nilai *codebook* data latih sehingga menghasilkan nilai yang menunjukkan urutan *file* suara. Apabila *file* suara dikenal maka sistem akan menampilkan data penelepon dikenal dengan mengambil data pegawai dari *server*. Apabila penelepon tidak dikenali maka sistem akan menampilkan tulisan "Tidak Dikenal" pada semua label form tampilan pengenalan suara. Tombol berhenti berfungsi untuk menghentikan

proses identifikasi penelepon, apabila kegiatan telepon telah berakhir. *Activity* diagram digambarkan pada gambar 5.

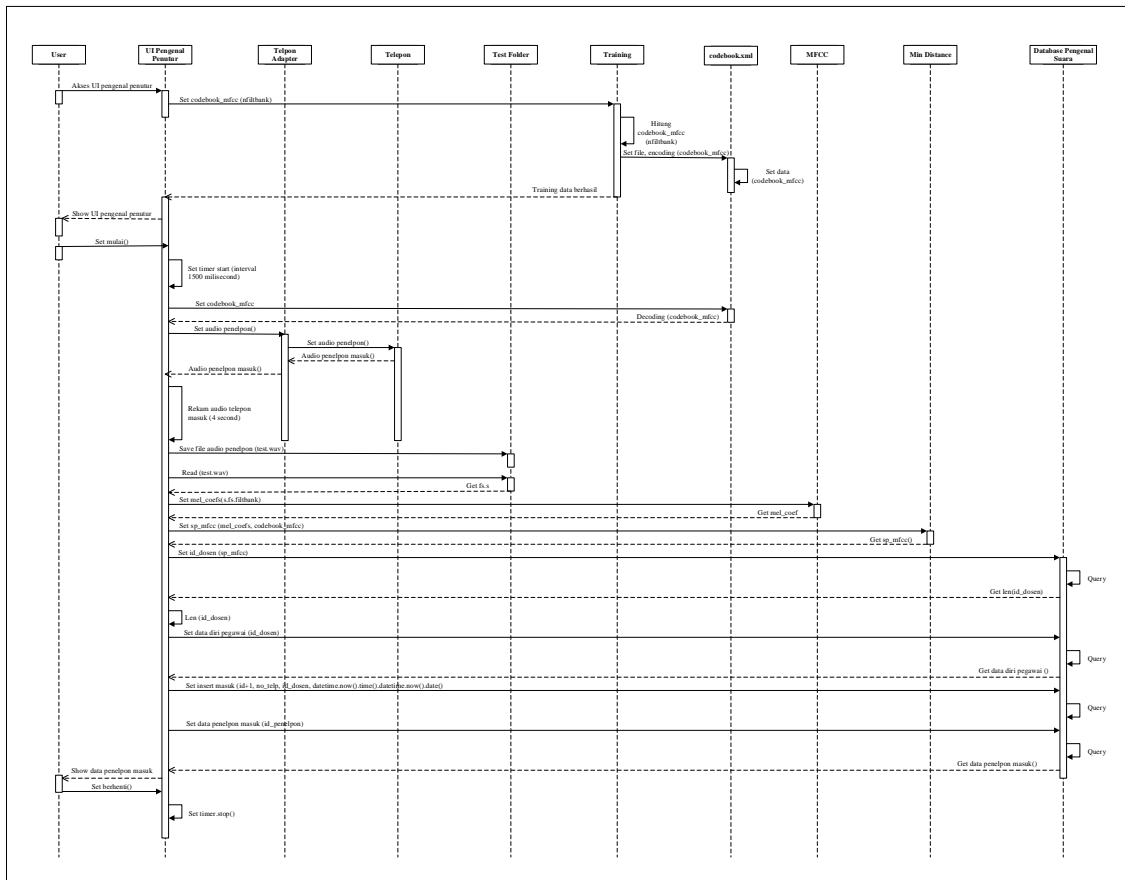


Gambar 5 Activity Diagram Pengenalan Penutur

d. *Sequence* Diagram Pengenalan Penutur

Sequence diagram pengenalan penutur pada gambar 6 menggambarkan interaksi pengiriman pesan antara objek-objek yang berperan dalam fungsi pengenalan penutur berdasarkan waktu. Selain *user* terdapat delapan objek pada *sequence* diagram pengenalan

penutur. Ui pengenalan penutur merepresentasikan *class* antarmuka pengenalan penutur. *User* mengawali proses dengan mengirim pesan akses kepada Ui pengenalan penutur.



Gambar 6 Sequence Diagram Pengenalan Penutur

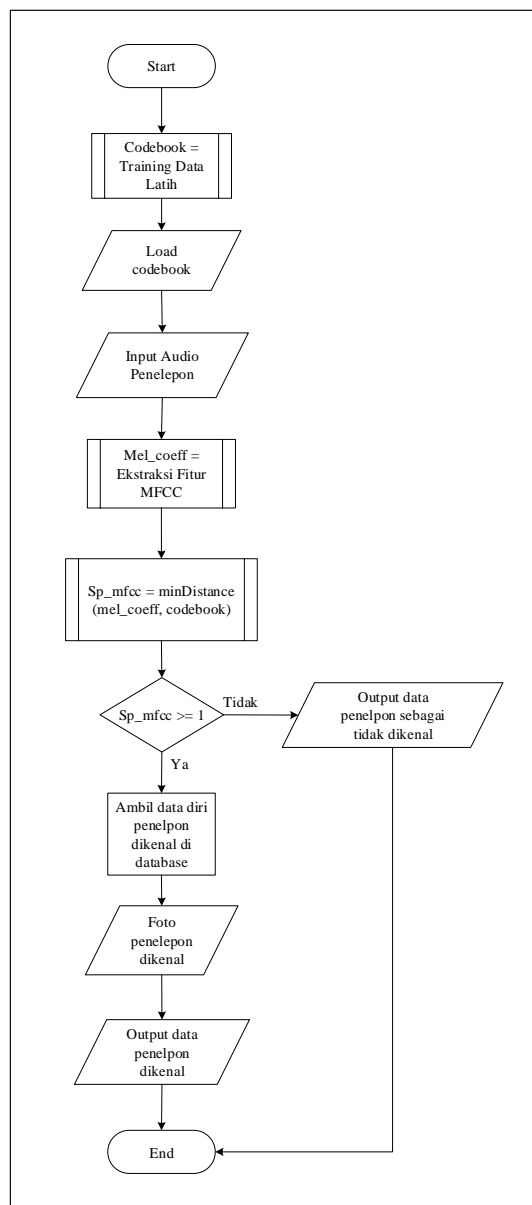
Ui pengenalan penutur kemudian mengembalikan pesan tersebut dengan menampilkan antarmuka pengenalan penutur. Kemudian *user* mengirim pesan kepada antarmuka untuk memulai rekam suara pada *microphone*. Berdasarkan pesan tersebut kemudian antarmuka memulai *timer* dengan interval 500 milidetik. Pada proses perekaman, antarmuka meminta sinyal *audio* dari telepon adapter yang bertindak sebagai masukan *microphone*. Pesan yang diterima kemudian diteruskan ke telepon yang terhubung dengan adapter. Telepon kemudian mengembalikan pesan pesan ke adapter berupa data *audio* penelepon. Data *audio* penelepon masuk kemudian diteruskan oleh adapter ke antarmuka. Data tersebut kemudian direkam oleh class ui pengenalan suara selama 4 detik. Setelah proses perekaman selsesai, antarmuka kemudian mengirim pesan ke folder *test* untuk menyimpan file *audio* yang dihasilkan dan berformat *.wav*. Setelah file suara berhasil tersimpan pada *storage*. Antarmuka mengirim pesan kepada fungsi *training* untuk menghasilkan nilai *codebook mfcc* dengan masukan berupa *nfilbank*. Setelah diproses, kemudian fungsi *training* mengembalikan nilai *codebook mfcc* ke antarmuka. Langkah selanjutnya, antarmuka mengirim pesan untuk membaca file suara penelepon yang terekam pada folder *test*. Nilai yang diterima dari pembacaan file *audio* adalah *fs* dan *s*. Setelah mendapatkan nilai *fs* dan *s*, antarmuka mengirim pesan kepada fungsi *MFCC* untuk memproses *fs,s*, dan *nfilbank* menjadi *mel_coeff*. Setelah diproses kemudian *MFCC* mengembalikan nilai *mel_coeff* kepada antarmuka. Selanjutnya antarmuka mengirim pesan kepada fungsi *minDistance* untuk memproses nilai *mel_coeff* dan *codebook_mfcc* menjadi *sp_mfcc*. Setelah diproses kemudian fungsi *minDistance* mengembalikan nilai *sp_mfcc*.

Flowchart

Flowchart menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam program. Flowchart pada penelitian ini terdiri dari proses pengenalan penutur, *training* data latih, ekstraksi fitur, metode *Vector Quantization*, serta *Minimum Distance*.

a. Flowchart Pengenalan Penutur

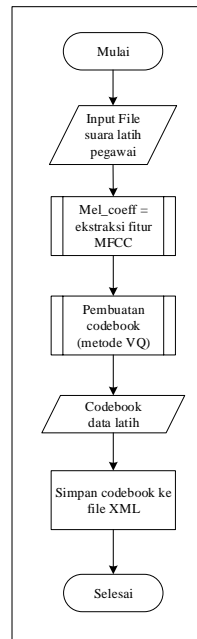
Pengenalan penutur merupakan proses pengidentifikasian suara penelepon yang masuk untuk ditampilkan pada aplikasi. *Admin* selaku pengguna aplikasi dapat melihat identitas penelepon tanpa harus menanyakannya secara langsung. Pengenalan penutur dijelaskan pada *flowchart* dimulai dengan mencari nilai *codebook*. Nilai *codebook* yang disimpan oleh prosedur *training* data latih pada file *xml* kemudian diambil melalui mekanisme *decoding*. File audio penelepon kemudian diproses oleh prosedur ekstraksi *MFCC* untuk dihasilkan nilai *mel koefisien*. Nilai koefisien data suara penelepon dibandingkan dengan *codebook* data latih melalui prosedur *minDistance* untuk menghasilkan nilai *sp_mfcc*. Nilai *sp_mfcc* merepresentasikan urutan file suara data latih yang memiliki selisih koefisien terdekat dengan suara penelepon masuk. Kemudian diberikan kondisi apabila nilai *sp_mfcc* lebih besar sama dengan 1, maka aplikasi akan mengambil data pegawai dengan id sama dengan nilai *sp_mfcc*. Flowchart pengenalan penutur digambarkan pada gambar 7.



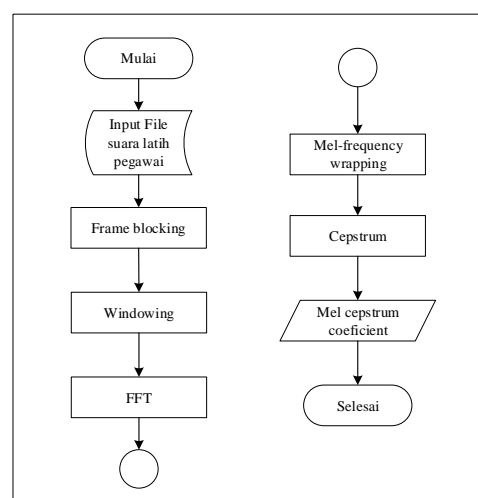
Gambar 7 Flowchart Pengenalan Penutur

b. *Flowchart Training Data Latih*

Training data latih merupakan proses penjabaran dari prosedur pencarian *codebook* pada *flowchart* pengenalan penutur. Proses dimulai dengan membaca input file suara latih pegawai yang terdapat pada *storage folder train*. Folder yang dibaca kemudian ekstraksi oleh prosedur ekstraksi fitur *MFCC*, untuk menghasilkan nilai *mel_coeff*. Nilai *mel_coeff* kemudian menjadi masukan untuk prosedur pada proses selanjutnya yaitu pembuatan *codebook* pada metode *VQ*. Proses pembacaan file hingga prosedur *VQ* dilakukan secara berulang sesuai dengan jumlah file suara data latih. Nilai *codebook* dari semua data latih suara lalu disimpan kedalam file penyimpanan dengan format *xml*. Penyimpanan dilakukan melalui tahapan *encoding* data terlebih dahulu. *Flowchart* digambarkan pada gambar 8.

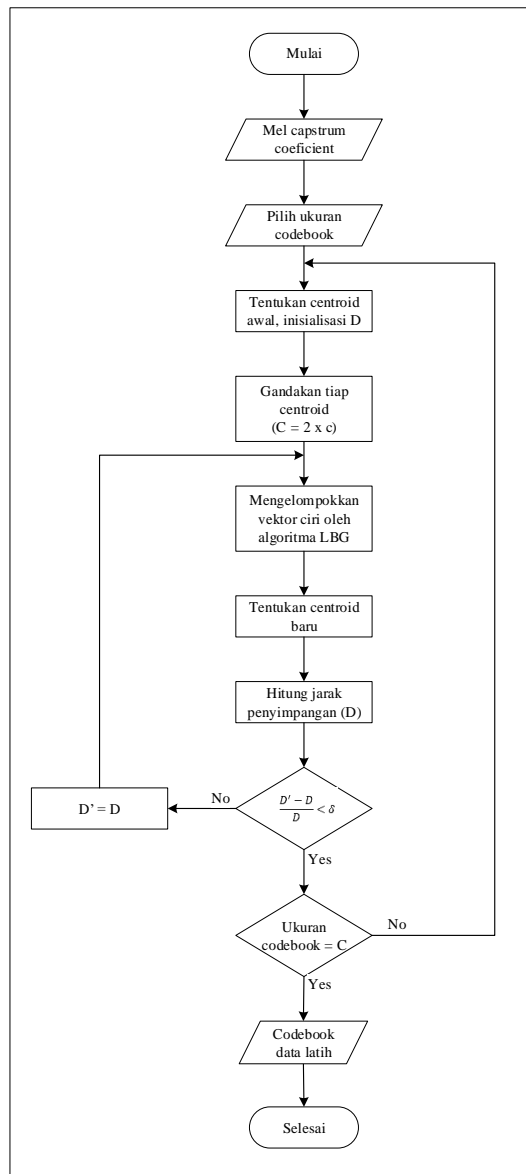
Gambar 8 *Flowchart Training Data Latih*c. *Flowchart Ekstraksi Fitur*

Proses ekstraksi fitur merupakan proses ekstraksi data *audio* dari file suara yang masuk, untuk dihasilkan nilai koefisien. Proses dimulai dengan membaca file suara latih pegawai pada *storage*. Proses *MFCC* ini mengubah spektrum log *mel* kembali ke waktu sehingga menghasilkan *MFCC* (*Mel Frequency Cepstrum Coefficients*). *Flowchart* digambarkan pada gambar 9.

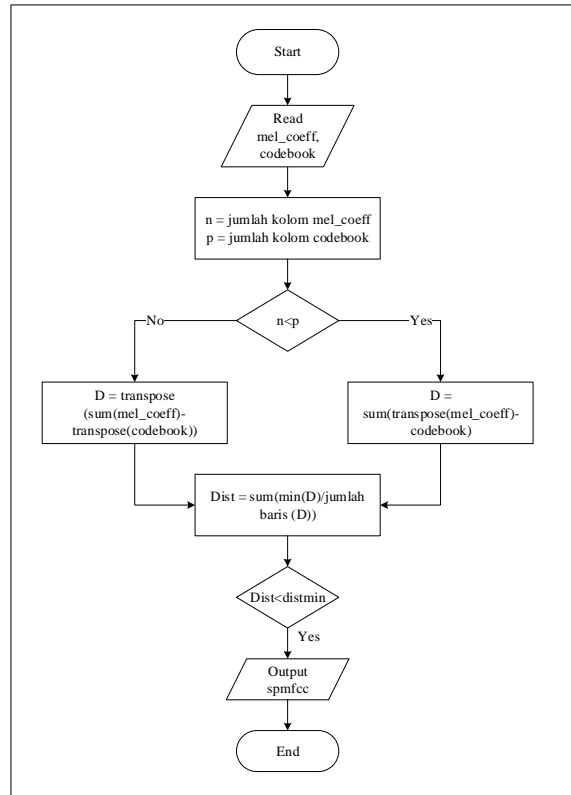
Gambar 9 *Flowchart Ekstraksi Fitur*

d. *Flowchart Metode Vector Quantization*

Flowchart metode *vector quantization* merupakan prosedur yang terdapat pada proses *training* data latih. Proses pada metode ini dimaksudkan untuk membentuk *codebook* dari setiap data latih. Nilai *codebook* yang dihasilkan merupakan nilai dari masing-masing data suara yang masuk dan akan diproses oleh proses data *training*. *Flowchart* digambarkan pada gambar 10.

Gambar 10 *Flowchart* Metode *Vector Quantization*e. *Flowchart Minimum Distance*

Minimum distance dimulai dengan membaca nilai *mel_coeff* penelepon masuk serta *codebook* data latih. Kemudian dideklarasikan variabel *n* sebagai jumlah kolom dari array *mel_coeff*, serta *p* merupakan jumlah kolom array *codebook*. Kemudian pada kondisi apabila nilai *n* lebih kecil dari *p*, maka nilai *D* ditentukan dengan *transpose mel coeff* lalu diproses dengan *codebook* untuk menghasilkan nilai *D*. Apabila tidak memenuhi kondisi maka nilai dari *codebook* yang akan di *tranpose* lalu diproses untuk menghasilkan nilai *D*. Nilai *Dist* merupakan penjumlahan nilai minimum *D* dibagi dengan jumlah baris pada *D*. Apabila nilai *Dist* yang dihasilkan lebih kecil dari *distmin*, maka akan menghasilkan keluaran nilai *sp_mfcc*. *Flowchart* digambarkan pada gambar 11.

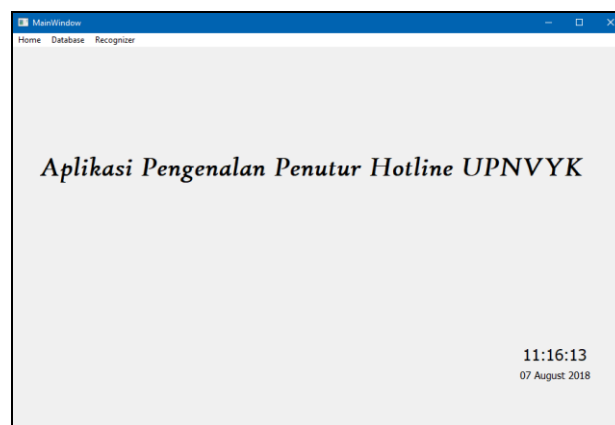


Gambar 11 Flowchart Minimum Distance

HASIL DAN PEMBAHASAN

Halaman Utama

Halaman utama merupakan antarmuka awal dari aplikasi. *User* dapat mengakses antarmuka lain melalui menu-menu pada *menu bar*. Terdapat tiga menu pada *menu bar*, yaitu : *home*, *database*, dan *recognizer*. Menu *home* mengarahkan aplikasi ke antarmuka utama. Menu *database* terdiri dari *submenu* dosen dan penelpon. *Submenu* dosen mengarahkan aplikasi ke antarmuka *database* pegawai. *Submenu* penelpon mengarahkan aplikasi ke antarmuka data penelpon dikenal. Menu *recognizer* terdiri dari *submenu hotline* yang mengarahkan aplikasi ke antarmuka pengenalan penutur. Antarmuka utama juga menampilkan judul aplikasi serta waktu dan tanggal saat ini. Antarmuka halaman utama dapat dilihat pada gambar 12.

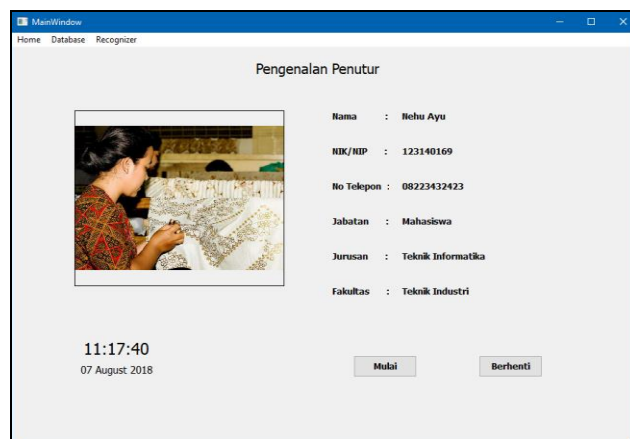


Gambar 12 Halaman Utama

Halaman Pengenalan Penutur

Halaman pengenalan penutur merupakan halaman yang digunakan untuk melakukan pengidentifikasian suara penelepon. Terdapat tiga menu pada Terdapat tiga menu pada *menu bar*, yaitu *home*, *database*, dan *recognizer*. Proses pengidentifikasian suara penelepon dimulai

ketika *user* menekan tombol mulai. Setelah tombol mulai ditekan, aplikasi akan mulai merekam suara. Hasil rekaman suara tersebut kemudian dibandingkan dengan data suara latih pegawai. Apabila suara penelepon masuk memiliki nilai yang mendekati salah satu dari data suara latih pegawai, maka aplikasi akan menampilkan data pegawai. Data pegawai yang ditampilkan diambil dari *database* pegawai yang terdiri dari nama, NIK/NIP, no telepon, jabatan, jurusan, fakultas, serta foto. Data pegawai ditampilkan melalui label-label yang disediakan pada antarmuka halaman pengenalan penutur. Apabila suara penelepon tidak teridentifikasi sebagai pegawai, maka halaman pengenalan penutur akan menampilkan tulisan "Tidak Dikenal" pada semua label, serta mengosongkan label foto. Aplikasi akan menampilkan hasil identifikasi suara setiap 3 sampai 4 detik. Tombol berhenti berfungsi untuk menghentikan aplikasi dalam mengidentifikasi suara penelepon. Waktu beserta tanggal saat ini ditampilkan melalui label yang terletak pada bawah label foto. Antarmuka halaman pengenalan penutur dapat dilihat pada gambar 13.



Gambar 13 Halaman Pengenalan Penutur

Halaman Data Penelepon Dikenal

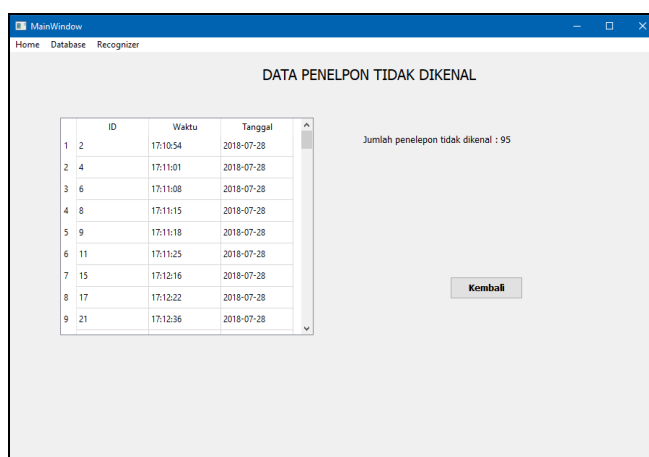
Halaman data penelepon dikenal merupakan halaman yang digunakan untuk menampilkan data penelepon masuk yang dikenal sebagai pegawai yang terdaftar pada basis data. *User* dapat mengakses halaman data penelepon dikenal dengan menekan *action* penelepon pada menu *database*. *Menu bar* terdiri dari *home*, *database*, dan *recognizer*. Judul dari halaman terletak pada bagian atas tabel. Semua data penelepon yang dikenali oleh aplikasi ditampilkan pada tabel. Kolom pada tabel terdiri dari id, status, nama, jurusan, jabatan, no telepon, waktu, dan tanggal. Jumlah penelepon dikenali ditampilkan pada label yang terletak pada bagian kiri atas dari tabel. Pada bagian kanan atas tabel terdapat tombol bertuliskan tidak dikenal yang berfungsi untuk menampilkan antarmuka tampil data penelepon tidak dikenal. Antarmuka halaman data penelepon dikenal dapat dilihat pada gambar 14.

ID	Status	Nama	Jurusan	Jabatan	No telepon	Waktu	Tanggal
1	Dikenal	I Putu Wahyu P...	Teknik Informat...	Teknik Informat...	081220535152	17:10:50	2018-07-28
2	Dikenal	I Putu Wahyu P...	Teknik Informat...	Teknik Informat...	081220535152	17:10:57	2018-07-28
3	Dikenal	I Putu Wahyu P...	Teknik Informat...	Teknik Informat...	081220535152	17:11:04	2018-07-28
4	Dikenal	I Putu Wahyu P...	Teknik Informat...	Teknik Informat...	081220535152	17:11:11	2018-07-28
5	Dikenal	Muhammad Fa...	Teknik Informat...	Mahasiswa	085312453810	17:11:22	2018-07-28
6	Dikenal	I Putu Wahyu P...	Teknik Informat...	Teknik Informat...	081220535152	17:12:05	2018-07-28
7	Dikenal	I Putu Wahyu P...	Teknik Informat...	Teknik Informat...	081220535152	17:12:09	2018-07-28
8	Dikenal	I Putu Wahyu P...	Teknik Informat...	Teknik Informat...	081220535152	17:12:12	2018-07-28
9	Dikenal	I Putu Wahyu P...	Teknik Informat...	Teknik Informat...	081220535152	17:12:19	2018-07-28

Gambar 14 Halaman Data Penelepon Dikenal

Halaman Data Penelepon Tidak Dikenal

Halaman data penelepon tidak dikenal merupakan halaman yang digunakan untuk menampilkan data penelepon masuk yang dikenal sebagai pegawai yang terdaftar pada basis data. User dapat mengakses halaman data penelepon dikenal dengan menekan tombol “Tidak Dikenal” pada halaman data penelepon dikenal. Menu bar terdiri dari *home*, *database*, dan *recognizer*. Judul dari antarmuka terletak pada bagian atas tabel. Semua data penelepon yang dikenali oleh aplikasi ditampilkan pada tabel. Kolom pada tabel terdiri dari id, status, nama, jurusan, jabatan, no telepon, waktu, dan tanggal. Jumlah penelepon tidak dikenali ditampilkan pada label yang terletak pada bagian kanan dari tabel. Pada bagian bawah jumlah penelepon tidak dikenal terdapat tombol bertuliskan kembali yang berfungsi untuk mengembalikan aplikasi ke halaman tampil penelepon dikenal. Antarmuka halaman data penelepon tidak dikenal dapat dilihat pada gambar 15.



Gambar 15 Halaman Data Penelepon Tidak Dikenal

Pengujian Sistem

Pengujian sistem berfungsi untuk mendapatkan nilai ambang (*threshold*) yang tepat serta menghitung akurasi aplikasi pengenalan penutur. *FAR* dan *FRR* mengukur kesalahan keputusan untuk keseluruhan sistem. Pengukuran ini menggabungkan tingkat kesalahan pencocokan, dan kegagalan untuk memperoleh nilai sesuai dengan kebijakan keputusan sistem (Mansfield, Kelly, Chandler, & Kane, 2001). Pengujian dilakukan dengan delapan nilai *threshold* yang berbeda. Pengujian ini dibagi menjadi dua bagian. Pengujian pertama merupakan nilai ambang, dan pengujian kedua adalah pengujian akurasi. Jumlah orang yang diuji pada kedua pengujian tersebut berbeda.

a. Pengujian Nilai Ambang (*Threshold*)

Pengujian nilai ambang dilakukan dengan menguji masing-masing seputuh suara dari dua orang yang berbeda. Orang pertama telah terdaftar sebagai data *training* pada aplikasi. Nilai suara latih dari orang pertama disimpan dalam file *codebook* yang berformat *xml*. Orang kedua merupakan orang yang tidak terdaftar pada aplikasi. Pada orang yang terdaftar sebagai data *training*, pengujian yang dilakukan adalah *FRR*. Orang yang dijadikan objek pengujian *FRR* adalah Wahyu patrika. Data suara uji diambil berdasarkan suara yang direkam pada saat objek melakukan telepon dan terhubung ke aplikasi pengenalan penutur. Pengujian *FRR* menghitung persentase aplikasi ketika tidak mengenali suara objek yang seharusnya dikenali. Persentase *FRR* dari masing-masing *threshold* dihitung berdasarkan rumus berikut:

$$\text{Persentase } FRR = \frac{\text{Jumlah Suara Tidak Dikenal}}{\text{Total Suara}} \times 100\% \dots\dots\dots(6)$$

Berdasarkan perhitungan yang dilakukan melalui rumus, didapatkan persentase *FRR* terbesar adalah 100% pada *threshold* 15. Pada orang yang tidak terdaftar pada aplikasi, pengujian yang dilakukan adalah *FAR*. Orang yang dijadikan objek pengujian *FAR* adalah Riselda Lalusu. Data suara uji diambil berdasarkan suara yang direkam pada saat objek

melakukan telepon dan terhubung ke aplikasi pengenalan penutur. Pengujian *FAR* menghitung persentase aplikasi ketika mengenali suara objek yang seharusnya tidak dikenali. Persentase *FAR* dari masing-masing *threshold* dihitung berdasarkan rumus berikut:

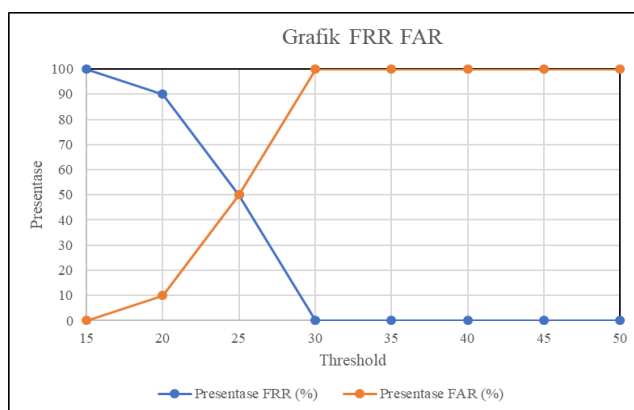
$$\text{Persentase FAR} = \frac{\text{Jumlah Suara Dikenal Benar}}{\text{Total Suara}} \times 100\% \dots\dots\dots(7)$$

Berdasarkan perhitungan yang dilakukan melalui rumus, didapatkan persentase nilai *FAR* adalah 100% pada *threshold* 30, 35, 40, 45, dan 50. Berdasarkan pengujian *FRR* dan *FAR* yang telah dilakukan, maka disusun tabel yang merekap hasil kedua pengujian tersebut. Rekap pengujian mendasari pembentukan grafik hasil pengujian. Tabel rekapitulasi pengujian *FAR* dan *FRR* nilai ambang dapat dilihat pada tabel 1.

Tabel 1 Tabel rekapitulasi pengujian *FAR* dan *FRR* nilai ambang

No	Thershold	FRR (%)	FAR (%)
1	15	100	0
2	20	90	10
3	25	50	50
4	30	0	100
5	35	0	100
6	40	0	100
7	45	0	100
8	50	0	100

Nilai ambang yang dapat memaksimalkan hasil dari pengenalan penutur didapat dari nilai pada titik perpotongan *FAR* dan *FRR*. Berdasarkan grafik *FAR* dan *FRR* didapatkan nilai *threshold* 25 menjadi titik perpotongan pada pengujian. Pada *threshold* 25, nilai persentase yang dihasilkan adalah 50%. Grafik *FAR* dan *FRR* dapat dilihat pada gambar 16.



Gambar 16 Grafik pengujian *FAR* dan *FRR*

b. Pengujian Akurasi

Pengujian akurasi dilakukan dengan menguji sepuluh suara dari enam orang yang berbeda. Keenam orang yang menjadi objek pengujian terdiri dari tiga orang yang terdaftar sebagai data *training* aplikasi, dan tiga orang tidak terdaftar pada data *training* aplikasi. Objek pengujian yang terdaftar pada data *training* adalah Wahyu Patrika, Ifa, dan Muhammad Fahim. Objek pengujian yang tidak terdaftar pada data *training* adalah Riselda Lalu, Batista Maryanto, dan Yolanda. Persentase akurasi dari objek yang terdaftar pada data *training* dihitung berdasarkan rumus:

$$\text{FRR} = \frac{\text{Jumlah Suara Dikenali Benar}}{\text{Total Suara}} \times 100\% \dots\dots\dots(8)$$

Rumus nilai *FRR* yang digunakan merupakan kebalikan dari rumus nilai *FRR* pada pengujian nilai ambang. Persentase akurasi dari objek yang tidak terdaftar pada data *training* dihitung berdasarkan rumus:

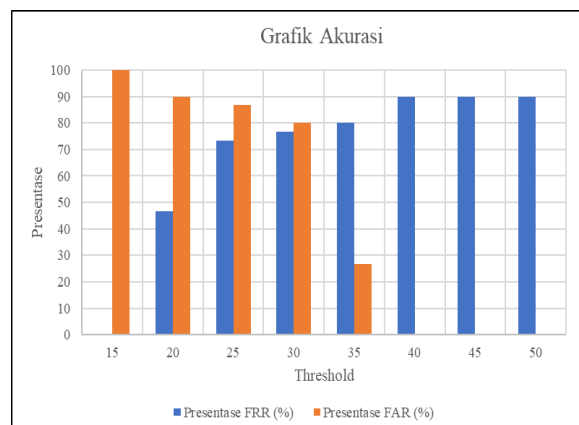
$$FAR = \frac{\text{Jumlah Suara Tidak Dikenali}}{\text{Total Suara}} \times 100\% \dots\dots\dots(9)$$

Rumus nilai *FAR* yang digunakan merupakan kebalikan dari rumus nilai *FAR* pada pengujian nilai ambang. Akurasi identifikasi suara penelepon pada aplikasi pengenalan penutur memiliki nilai persentase tertinggi yaitu 90% pada *threshold* 40, 45, dan 50. Nilai persentase akurasi *FRR* tertinggi adalah 90% dengan rata-rata dari keseluruhan pengujian senilai 68,33%. Nilai persentase akurasi *FAR* tertinggi adalah 100% dengan rata-rata dari keseluruhan pengujian senilai 47,54%. Nilai Rata-rata pengenalan penutur tertinggi berdasarkan pengujian *FRR* dan *FAR* bernilai 80% pada *threshold* 25. Hasil tersebut sesuai dengan pengujian nilai ambang, dimana nilai *threshold* 25 merupakan nilai titik perpotongan pengujian *FAR* dan *FRR*. Hasil rekapitulasi pengujian akurasi dapat dilihat pada tabel 2.

Tabel 2 Tabel rekapitulasi pengujian akurasi

No.	Threshold	Akurasi		Rata-rata (%)
		FRR (%)	FAR (%)	
1	15	0	100	50
2	20	47	90	68
3	25	73	87	80
4	30	77	80	78
5	35	80	27	53
6	40	90	0	45
7	45	90	0	45
8	50	90	0	45
Rata-rata (%)		68,33	47,54	57,94
Max (%)		90	100	

Hasil rekapitulasi pengujian akurasi digambarkan pada grafik akurasi. Grafik akurasi menggambarkan nilai *FAR* dan *FRR* dari masing-masing *threshold*. Grafik akurasi dapat dilihat pada gambar 17.



Gambar 17 Grafik pengujian akurasi

KESIMPULAN

Berdasarkan dari hasil analisis, perancangan dan pembahasan yang telah dilakukan sebelumnya, didapatkan beberapa kesimpulan yaitu:

1. Hasil penelitian ini mampu menghubungkan telepon pada layanan *hotline* dengan komputer. Suara penelepon yang masuk melalui telepon dikenali berdasarkan *database* dosen maupun staff pada komputer. Terhubungnya telepon dan komputer memudahkan petugas dalam memonitor aktivitas telepon yang masuk pada layanan *hotline*.
2. Hasil penelitian ini mampu mengurangi resiko terjadinya ketidaksesuaian data penelepon yang disampaikan dibandingkan dengan identitas sebenarnya penelepon. Proses identifikasi penelepon pada aplikasi pengenalan penutur dilakukan dengan membandingkan ciri khas suara penelepon dibandingkan dengan ciri khas data latih suara dosen maupun staff yang terdaftar pada aplikasi.
3. Hasil penelitian ini mampu mempersingkat proses identifikasi penelepon pada layanan *hotline*. Penelepon yang telah teridentifikasi ditampilkan pada antarmuka aplikasi sekitar tiga sampai empat detik dari telepon masuk. Apabila dibandingkan dengan proses identifikasi yang berjalan pada layanan *hotline* saat ini, aplikasi pengenalan penutur mempersingkat waktu sekitar enam detik.
4. Semua fungsi dari antarmuka pada hasil penelitian ini berjalan dengan baik berdasarkan pengujian *black box*. Pengujian menghasilkan persentase 100% berdasarkan penilaian 7 responden.
5. Metode *Mel-Frequency Cepstral Coefficient* dan *Vector Quantization* dapat mengidentifikasi suara penelepon dengan tingkat akurasi tertinggi yaitu 80% pada nilai ambang (*threshold*) 25.

DAFTAR PUSTAKA

- Apriyono, N., Wahyudi, W., & Hidayatno, A. (2011). *Pengenalan Pengucap Tak Bergantung Teks dengan Metode Vector Quantization (VQ) Melalui Ekstraksi Linear Predictive Coding (LPC)* (PhD Thesis). Jurusan Teknik Elektro Fakultas Teknik.
- Mansfield, T., Kelly, G., Chandler, D., & Kane, J. (2001). Biometric product testing final report. *Contract*, 92(4009), 309.
- Munggaran, A. K., Djamal, E. C., & Yuniarti, R. (2017). Identifikasi Suara Pengontrol Lampu Menggunakan Mel-Frequency Cepstral Coefficients dan Hidden Markov Model. In *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*.
- Prasetyawan, P. (2016). Perbandingan Identifikasi Pembicara Menggunakan Mfcc Dan Sbc Dengan Ciri Pencocokan Lbg-Vq.
- Mustofa, A. (2008). Sistem Pengenalan Penutur dengan Metode Mel-frequency Wrapping. *Jurnal Teknik Elektro*, 7(2), 88–96.
- Schmuller, J. (2004). *Sams Teach Yourself UML in 24 Hours* (3 ed.). Indiana USA: Sams Publishing.
- Sulistiyorini, P. (2009). Pemodelan Visual dengan Menggunakan UML dan Rational Rose. *Dinamik-Jurnal Teknologi Informasi*, 14(1).