



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metodologi

Metodologi penelitian yang digunakan dalam perancangan sistem pengenalan suara mausia menggunakan metode Jaringan Syaraf Tiruan *Backpropagation* untuk menyalakan saklar lampu yaitu sebagai berikut.

1. Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mendapatkan jumlah jenis suara yang digunakan. Jenis suara yang dimaksudkan merupakan jumlah suara yang didapatkan dari beberapa orang dengan jenis kelamin yang berbeda. Pada tahap ini ditentukan berapa banyak data suara yang digunakan untuk proses *learning* dan untuk proses pengujian sistem.

2. Telaah Literatur

Telaah literatur dilakukan guna mempelajari teori-teori dasar yang berhubungan dengan perancangan sistem pengenalan suara. Teori-teori dasar yang dipelajari adalah Jaringan Syaraf Tiruan *Backpropagation*, ekstraksi ciri MFCC, perangkat keras Arduino Uno.

3. Perancangan Sistem

Perancangan sistem dilakukan dengan membuat *use case* diagram, *activity* diagram, *sequence* diagram dan *class* diagram untuk memudahkan pemahaman terhadap alur aplikasi serta perancangan antarmuka dan struktur tabel sebagai tempat penyimpanan data.

4. Pemrograman Sistem

Pemrograman sistem dilakukan menggunakan bahasa pemrograman C pada Arduino Uno. Sistem melakukan ekstraksi ciri suara menggunakan MFCC, dimana suara yang masuk diekstraksi ke dalam angka biner dan dimasukkan ke dalam sebuah matriks $N \times N$ dan digunakan sebagai pembelajaran pola suara dengan menggunakan metode Jaringan Syaraf Tiruan *backpropagation*. Jumlah layer yang digunakan dalam Jaringan Syaraf Tiruan sejumlah 3 buah *layer* yaitu *input layer*, *hidden layer*, serta *output layer*. Jumlah neuron yang digunakan pada tiap *layer* yakni sejumlah 12 neuron pada *input layer*, N neuron pada *hidden layer*, dan 5 neuron pada *output layer*. Jumlah neuron pada *hidden layer* diketahui pada saat menjalankan testing. Weight yang didapatkan dari hasil pembelajaran disimpan ke dalam tabel di *database*, dimana *weight* tersebut dikalkulasikan dengan suara yang diterima secara *real time*.

5. Testing

Testing dilakukan dengan menguji sistem dengan suara baru yang sudah direkam terlebih dahulu. Suara diuji dari 10 individu dengan melatih suara terlebih dahulu setelah itu dilakukan proses pengenalan dan diukur tingkat keberhasilannya dengan batas toleransi kesalahan sebesar 0.01.

6. Evaluasi Sistem

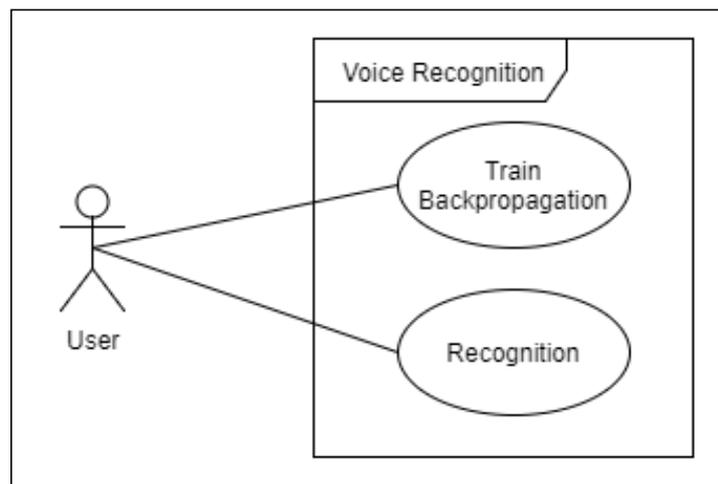
Evaluasi sistem dilakukan untuk menganalisa *output* dari hasil *testing* yang telah dilakukan sebelumnya, dimana hasil analisa dari *output* disimpulkan dalam bentuk persentase. Persentase yang didapatkan merupakan tingkat kecocokan antara ciri suara yang diterima secara *real time* dengan ciri suara yang ada pada *database*.

3.2 Perancangan Sistem

Perancangan sistem yang dimulai dengan pembuatan *use case* diagram, *activity* diagram, *sequence* diagram, *class* diagram, tampilan antarmuka dan struktur tabel.

3.2.1 Use Case Diagram

Use case diagram merupakan gambaran dari beberapa atau semua aktor dan interaksi di antar koomponen-komponen yang menggambarkan system yang dibangun. *Use case* diagram digunakan untuk menggambarkan bagaimana pandangan aktor yang terlibat dari luar sistem, Diagram ini menggambarkan fungsionalitas sistem serta bagaimana sistem berinteraksi dengan bagian yang tidak ada di dalam sistem. Berikut merupakan gambar dari *use case* diagram dari aplikasi yang telah dibuat sebagai berikut.



Gambar 3.1 Use Case Diagram

Pada diagram di atas terdapat satu *actor* saja yaitu *User*. *User* dapat melakukan dua aktivitas, yaitu *train Backpropagation*, *Recognition*. Untuk *train Backpropagation* dilakukan saat *user* ingin melakukan pengenalan suara untuk pertama kali saat suara tidak dapat dikenali. Pada aktivitas *Recognition* *user* dapat

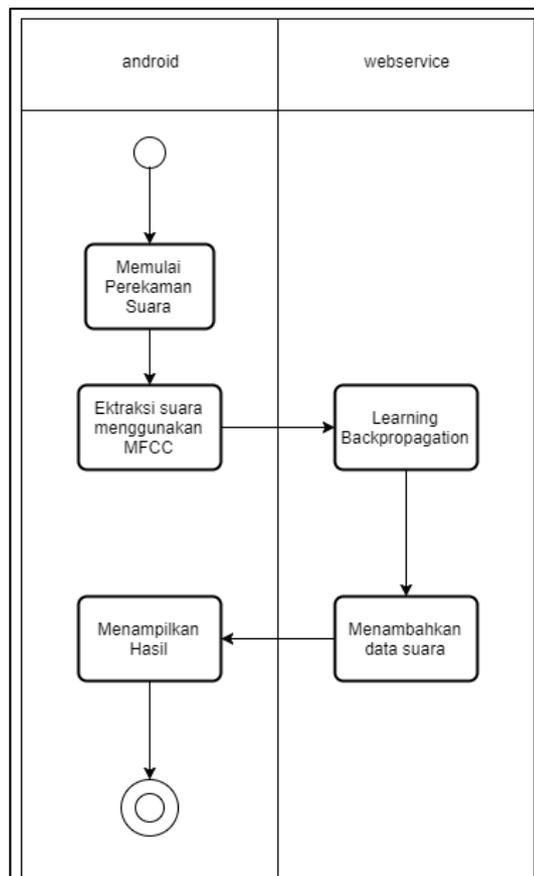
melakukan perekaman suara yang kemudian aplikasi dapat langsung melakukan pengenalan dengan membandingkan suara yang diterima dengan suara yang ada pada *database*.

3.2.2 Activity Diagram

Activity Diagram merupakan diagram yang menggambarkan langkah-langkah dan proses dari suatu sistem. Berikut merupakan gambar dari *activity* diagram yang mewakili beberapa proses, proses-proses tersebut adalah proses *train backpropagation*, dan proses *recognition*

A. Activity Diagram Train Backpropagation

Berikut ini merupakan gambar *activity* diagram pada *train backpropagation*.

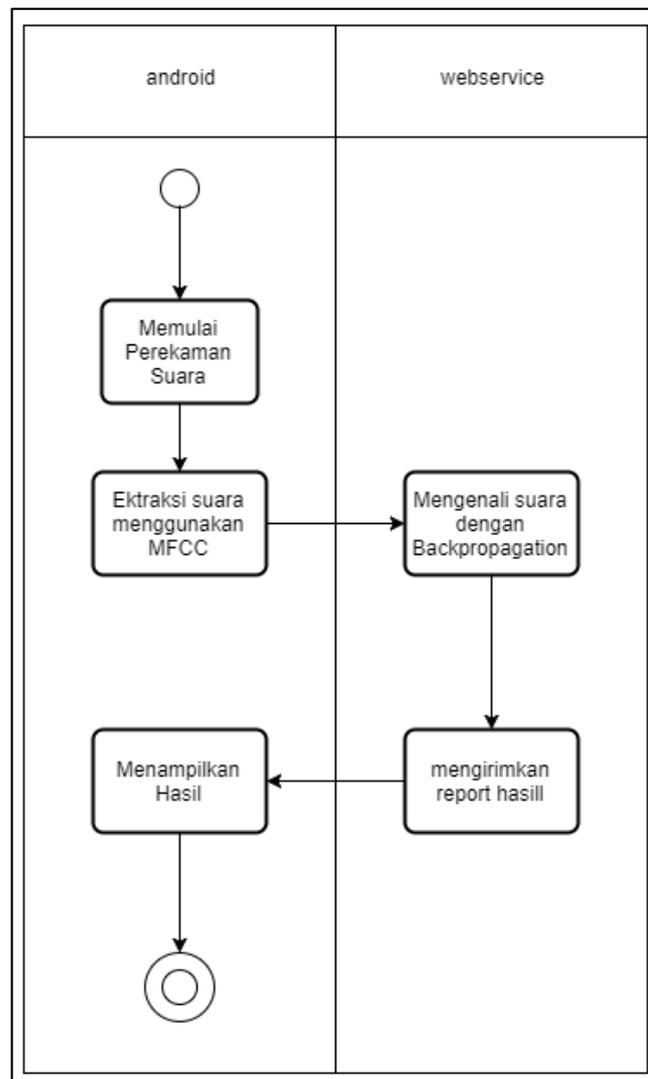


Gambar 3.2 Activity Diagram Train Backpropagation

Pada diagram diatas proses dimulai dengan melakukan perekaman suara yang ingin dikenali. Setelah melakukan perekaman suara, aplikasi langsung mengekstraksi ciri suara menggunakan MFCC, yang kemudian hasil dari ekstraksi ciri digunakan untuk melakukan *training* dengan menggunakan *backpropagation*. Dimana hasil dari *training* disimpan ke dalam *database* yang digunakan untuk proses *recognition*.

B. Activity Diagram Recognition

Berikut ini merupakan gambar *activity* diagram pada *recognition*.

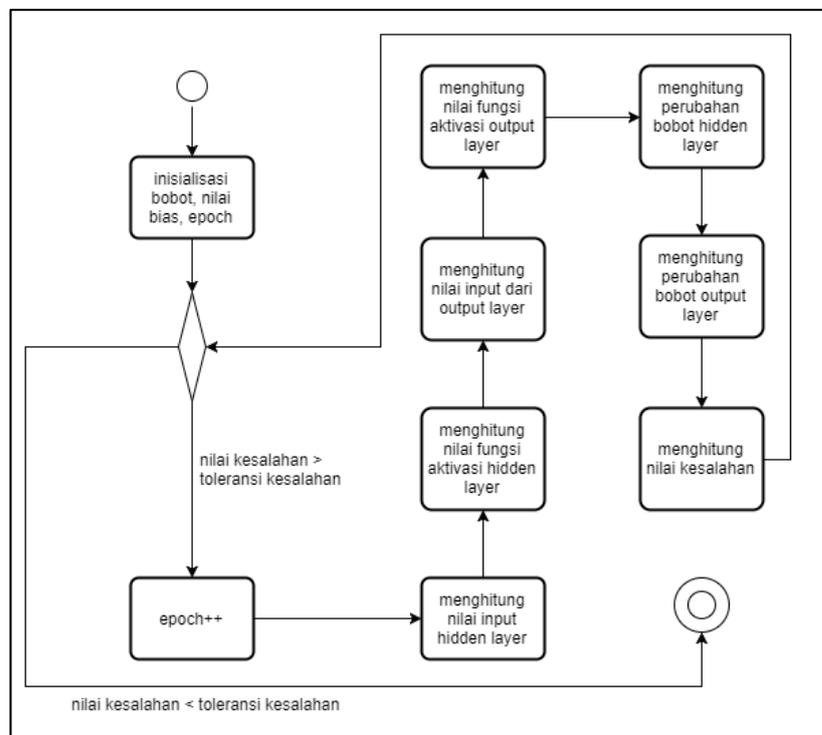


Gambar 3.3 Activity Diagram Recognition

Pada diagram di atas proses dimulai dengan melakukan perekaman suara yang ingin dikenali, setelah merekam suara maka aplikasi langsung melakukan ekstraksi ciri suara dengan menggunakan MFCC. Setelah mendapatkan hasil ekstraksi ciri maka aplikasi melakukan pengenalan suara dengan menggunakan *backropagation* yang berada di server lalu server mengirimkan hasil pengenalan ke client, kemudian ditampilkan hasil dari pengenalan pada Arduino berupa nyala atau matinya lampu.

C. Activity Learning Backpropagation

Berikut ini merupakan gambar *activity* diagram pada *learning backpropagation*.



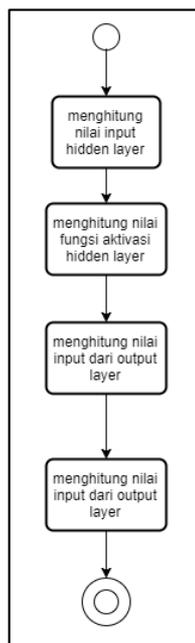
Gambar 3.4 Activity Learning Backpropagation

Pada diagram di atas proses dimulai dengan melakukan inisialisasi bobot dan nilai bias, setelah melakukan inisialisasi dilakukan pengecekan nilai kesalahan yang

ada, proses ini terus berjalan hingga nilai kesalahan lebih kecil dari nilai toleransi kesalahan maka iterasi dihentikan, jika tidak iterasi berjalan kemudian sistem menghitung nilai masukan pada *hidden layer* yang dikalikan dengan bobot yang telah di inisialisasi sebelumnya, kemudian nilai didapatkan dihitung dengan fungsi aktivasi yang telah ditentukan. Pada *output layer* nilai masukan juga dihitung dengan mengalikan nilai masukan dengan bobot yang diterima dari *hidden layer*, kemudian nilai didapatkan dihitung dengan fungsi aktivasi yang telah ditentukan. Penghitungan perubahan bobot pada *hidden layer* dan *output layer* di lakukan untuk mendapatkan bobot baru yang digunakan untuk iterasi selanjutnya, dihitung nilai kesalahannya.

D. Activity Training Backpropagation

Berikut ini merupakan gambar *activity* diagram pada *training backpropagation*.



Gambar 3.5 Activity Training Backpropagation.

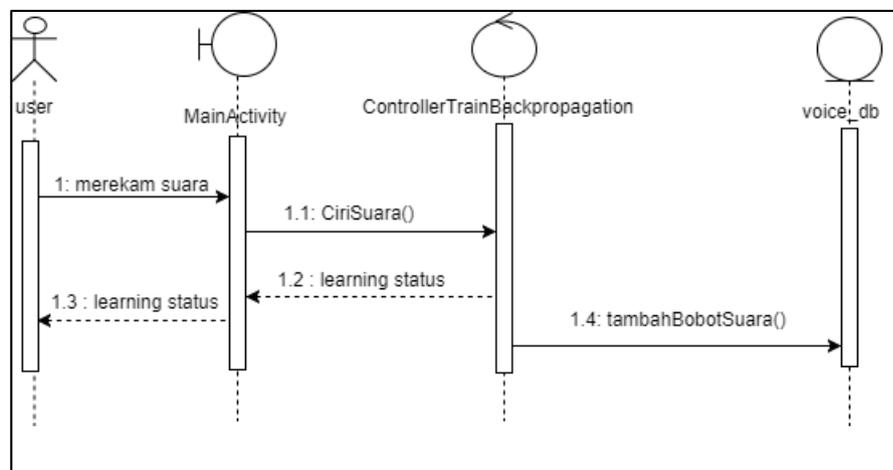
Pada diagram di atas proses dimulai menghitung nilai masukan pada *hidden layer* yang dikalikan dengan bobot yang telah didapatkan dari hasil pembelajaran, kemudian nilai didapatkan dihitung dengan fungsi aktivasi yang telah ditentukan. Pada *output layer* nilai masukan juga dihitung dengan mengalikan nilai masukan dengan bobot yang diterima dari *hidden layer*, kemudian nilai didapatkan dihitung dengan fungsi aktivasi yang telah ditentukan. Hasil *output* yang didapatkan digunakan untuk menentukan apakah suara yang masuk dapat dikenali atau tidak.

3.2.3 Sequence Diagram

Sequence Diagram untuk proses penggunaan *backpropagation* pada aplikasi. Proses dimulai dengan *user* melakukan perekaman suara yang ingin dikenali. Aplikasi melakukan ekstraksi ciri dengan *library* MFCC yang tersedia dan kemudian melakukan *learning* ataupun *traininig* dengan menggunakan *backpropagation*.

A. Sequence Diagram Train Backpropagation

Berikut ini merupakan gambar *sequence diagram train backpropagation*.

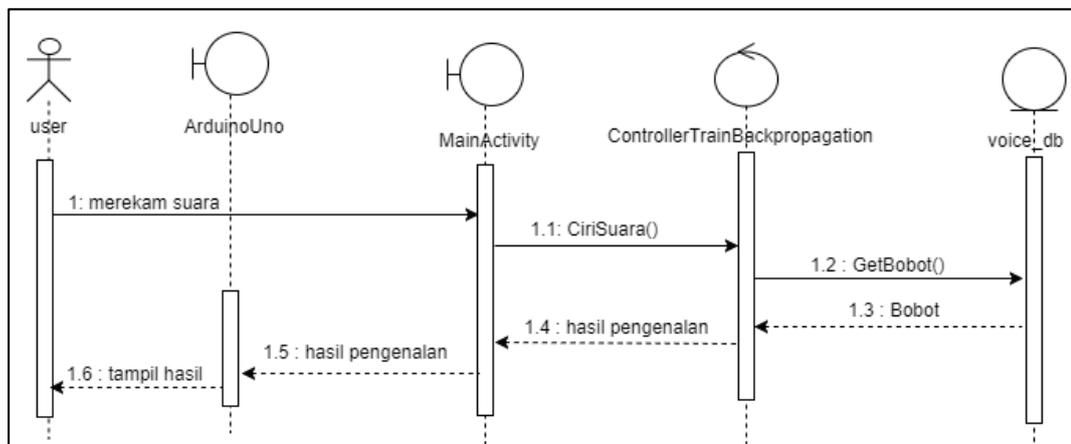


Gambar 3.6 Sequence Diagram Train Backpropagation

Gambar 3.6 merupakan *sequence diagram* untuk proses *train backpropagation* pada aplikasi. Proses dimulai dari *user* merekam suara, aplikasi melakukan ekstraksi ciri menggunakan MFCC kemudian mengirimkan hasil ekstraksi ke *server* dengan menjalankan fungsi *CiriSuara()*. *Server* menjalankan *learning* ketika untuk mendapatkan bobot suara yang digunakan untuk pengenalan. Kemudian *server* menjalankan fungsi *tambahBobotSuara()* yang melakukan penyimpanan bobot yang telah didapatkan dari *learning* ke dalam *database*.

B. Sequence Diagram Recognition

Berikut ini merupakan gambar *sequence diagram recognition*.



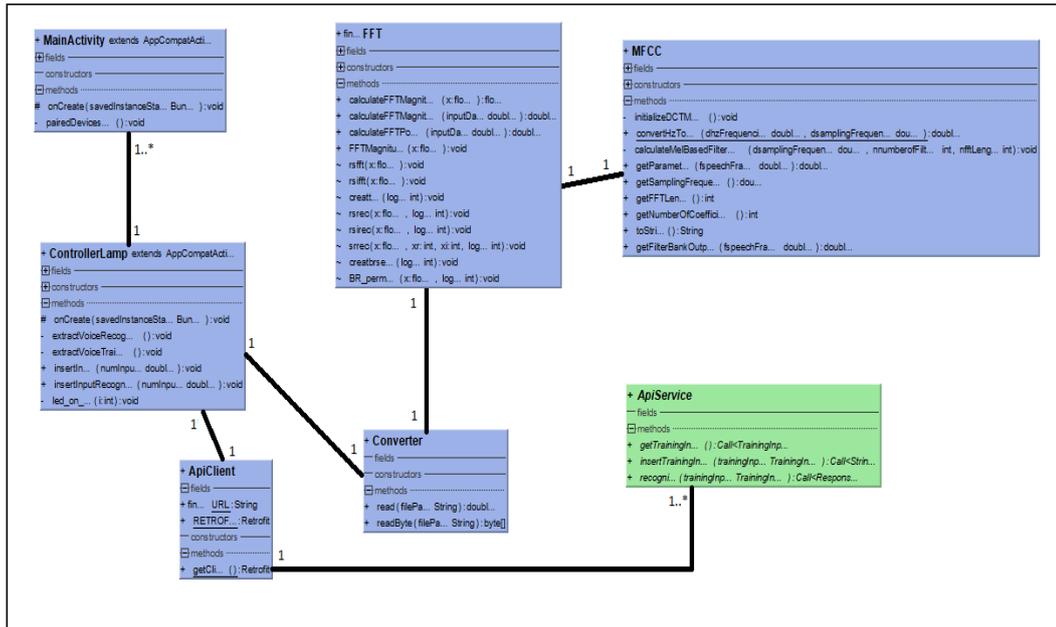
Gambar 3.7 Sequence Diagram Recognition

Gambar 3.7 merupakan *sequence diagram* untuk proses *recognition* pada aplikasi. Proses dimulai dari *user* merekam suara, aplikasi melakukan ekstraksi ciri menggunakan MFCC kemudian mengirimkan hasil ekstraksi ke *server* dengan menjalankan fungsi *CiriSuara()*. *Server* menjalankan *learning* ketika untuk mendapatkan bobot suara yang digunakan untuk pengenalan. Kemudian *server* menjalankan fungsi *GetBobot()* untuk mengambil bobot yang sudah disimpan

sebelumnya dari *database*. Dan kemudian aplikasi menampilkan hasil dikenali atau tidaknya suara tersebut.

3.2.4 Class Diagram

Berikut ini merupakan gambar *class* diagram aplikasi.

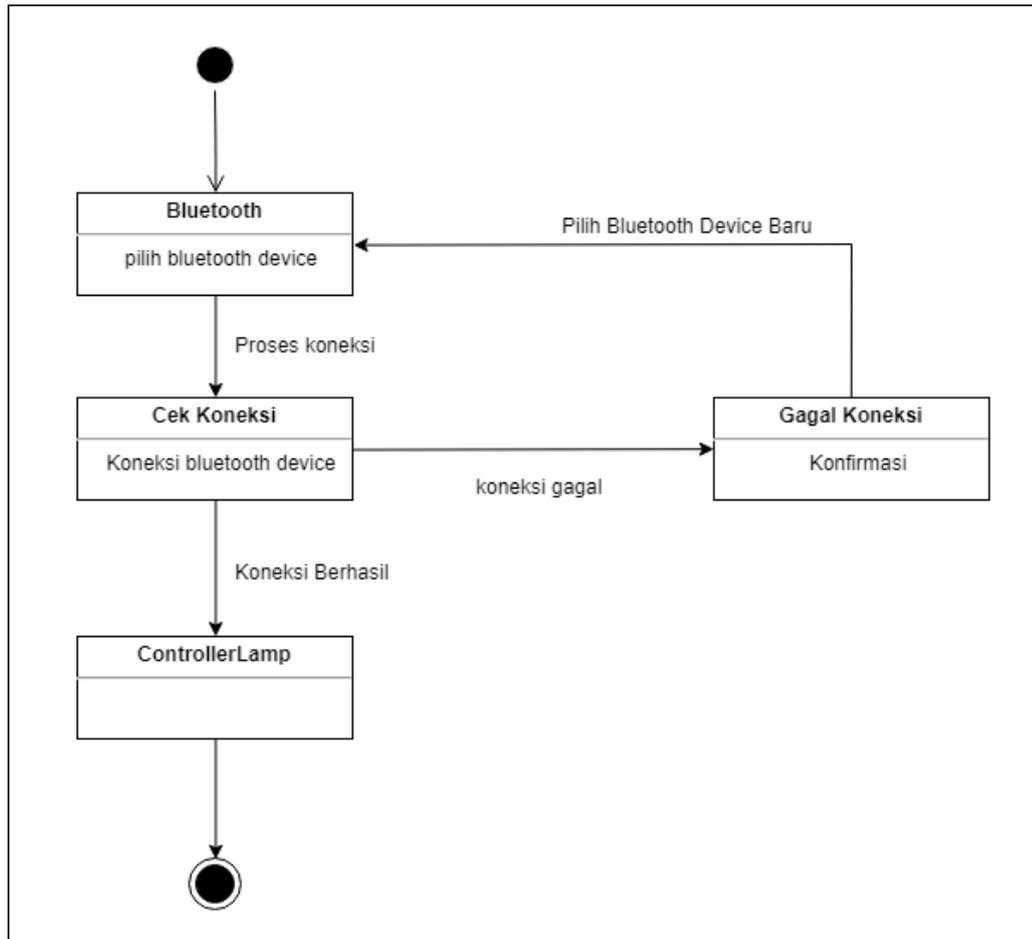


Gambar 3.8 Class Diagram

Gambar 3.8 merupakan *class* diagram aplikasi ini. *Class* MainActivity merupakan *class* yang menghubungkan aplikasi dengan Arduino Uno menggunakan Bluetooth. *Class* ControllerLamp merupakan *class* yang digunakan untuk melakukan perekaman suara dan mengontrol lampu. Kemudian *class* Converter, FFT, dan MFCC digunakan untuk melakukan ekstraksi ciri dari suara yang telah direkam sebelumnya. *Class* ApiService dan ApiClient berguna untuk melakukan koneksi antara aplikasi dengan server.

3.2.5 Statechart Diagram

Berikut ini merupakan gambar *Statechart* diagram pada koneksi Bluetooth.

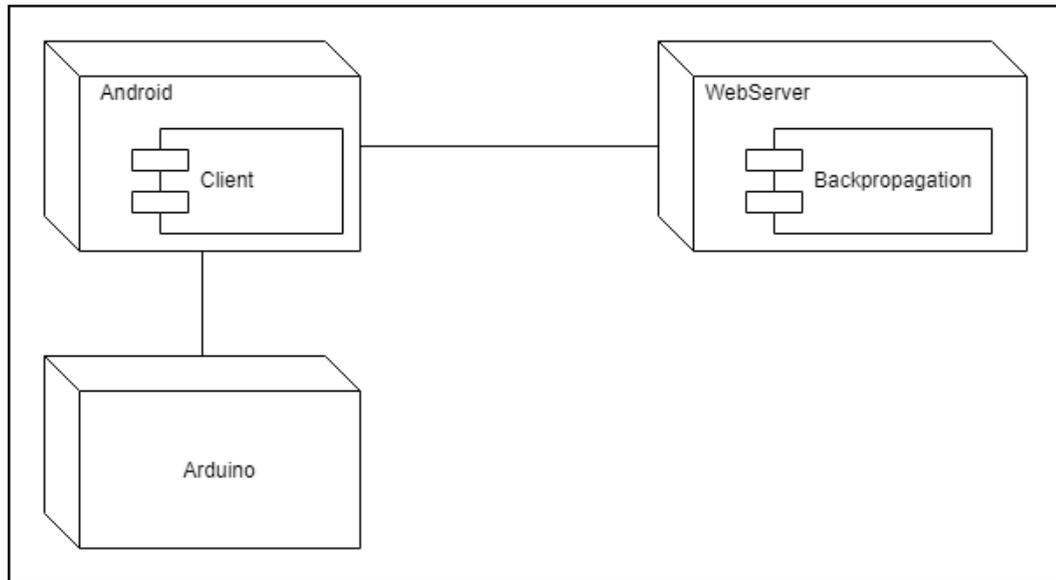


Gambar 3.9 Statechart Diagram Bluetooth

Gambar 3.9 merupakan *statechart* diagram untuk koneksi Bluetooth. Pengguna akan melakukan pemilihan Bluetooth *device* dan akan melakukan koneksi terlebih dahulu. Jika koneksi gagal pengguna akan diminta untuk memilih Bluetooth *device* yang ingin di koneksikan lagi. Jika berhasil melakukan koneksi sistem akan melanjutkan ke *class* ControllerLamp untuk melakukan kontrol terhadap lampu.

3.2.6 Deployment Diagram

Berikut ini merupakan gambar *Deployment* diagram aplikasi.

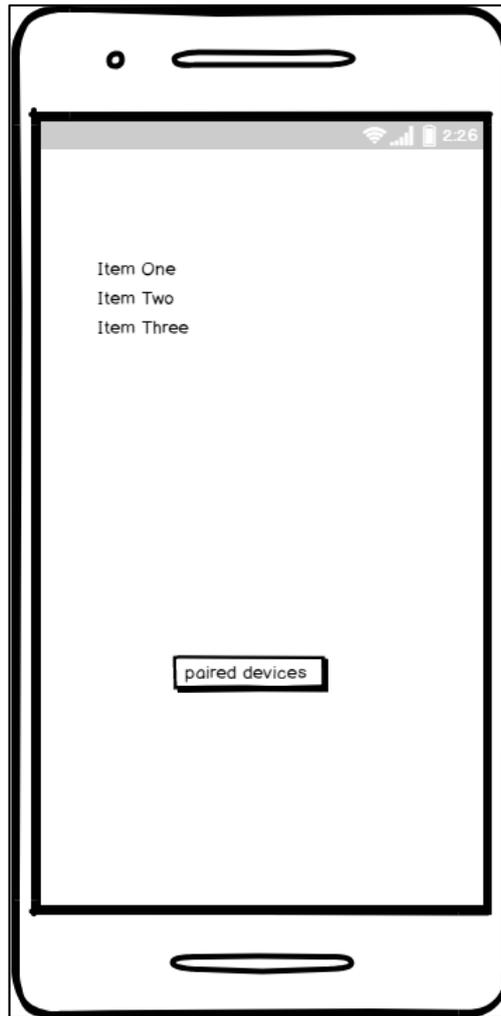


Gambar 3.10 Deployment Diagram

Gambar 3.10 Merupakan *deployment* diagram aplikasi ini. Android merupakan *Client* yang terhubung dengan WebServer yang melakukan proses pelatihan suara dan pengenalan suara dengan *backpropagation*. Android juga terhubung dengan Arduino yang melakukan proses pengontrolan lampu saat suara dikenali.

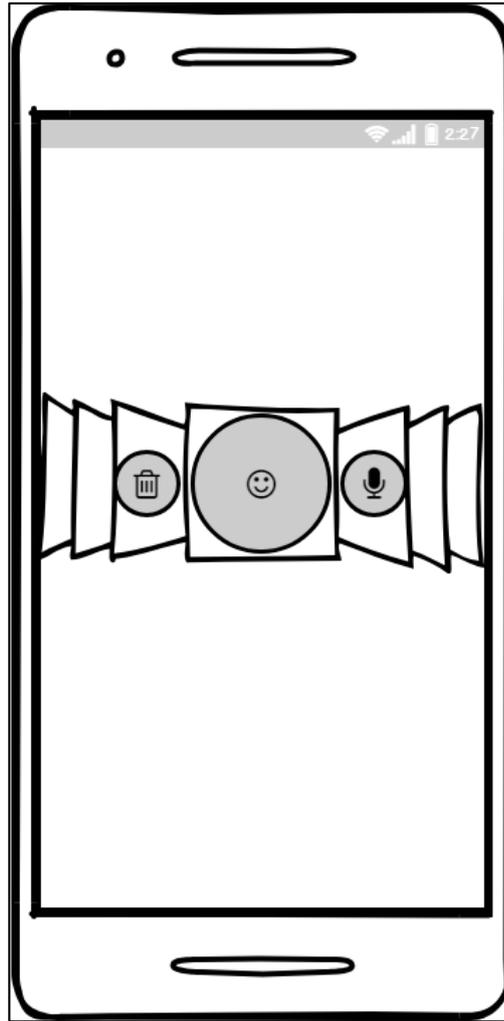
3.2.7 Desain Antarmuka

Pada bagian ini, ditampilkan beberapa Desain antarmuka sistem pengenalan suara dari sisi pengguna.



Gambar 3.11 Desain Antarmuka Koneksi Bluetooth

Gambar 3.11 merupakan desain antarmuka awal ketika pengguna membuka aplikasi, dimana pengguna melakukan koneksi Bluetooth ke Arduino Uno yang telah di *paired* terlebih dahulu sebelumnya. Tombol *paired devices* berguna untuk menampilkan nama serta *mac address* dari Bluetooth yang telah di *paired*.



Gambar 3.12 Desain Antarmuka Halaman Utama

Gambar 3.12 merupakan desain antarmuka halaman utama. Terdapat 3 buah pilihan dimana pada *button* bergambar wajah yang berguna untuk membuka halaman perekaman suara yang bertujuan untuk melatih suara baru sebelum pengenalan. Pada *button* tempat sampah berfungsi untuk melakukan *reset* dengan menghapus semua data suara yang sudah pernah direkam sebelumnya. Kemudian pada *button* dengan gambar mikrofon berguna untuk membuka halaman perekaman suara yang bertujuan untuk melakukan control terhadap lampu.



Gambar 3.13 Desain Antarmuka Halaman Perekaman suara

Gambar 3.13 merupakan desain antarmuka halaman perekaman suara. Terdapat sebuah *button* yang berfungsi untuk melakukan perekaman suara untuk melakukan kontrol terhadap lampu ataupun pembelajaran suara baru.

3.2.8 Struktur Tabel

Pada penelitian ini, basis data yang digunakan memiliki 5 tabel, dikarenakan setiap tabel tidak memiliki hubungan satu sama lain, maka tidak digambarkan Entity Relationship Diagram. pada Tabel 3.1 digunakan untuk menampung nama *user* dan target suara *on* dan target suara *off* untuk proses pengenalan dan pelatihan. Berikut merupakan struktur dari tabel yang digunakan.

Tabel 3.1 Struktur Tabel User

Nama Kolom	Tipe Data	Keterangan
ID	Integer	<i>Primary Key</i> , menyimpan ID dari setiap jenis suara
User	Varchar(255)	Nama <i>user</i> dari setiap jenis suara
targetOn	Int	Target jenis suara <i>on</i> pada <i>user</i> untuk pengenalan suara
targetOff	Int	Target jenis suara <i>off</i> pada <i>user</i> untuk pengenalan suara

Pada Tabel 3.2 digunakan untuk menampung nomor *row*, *col* dan nilai bobot *input* yang telah didapatkan dari hasil pelatihan. Berikut merupakan struktur dari tabel yang digunakan.

Tabel 3.2 Struktur Tabel Weight Input

Nama Kolom	Tipe Data	Keterangan
wIH_id	Integer	<i>Primary Key</i> , menyimpan ID dari setiap setiap <i>weight input</i>
row	Int	Nomor row yang berisikan nilai dari <i>weight</i>
col	Int	Nomor col yang berisikan nilai dari <i>weight</i>
Value	Float	Nilai <i>weight input</i> yang telah di dapatkan dari hasil pelatihan

Pada Tabel 3.3 digunakan untuk menampung nomor *row*, *col* dan nilai bobot *output* yang telah didapatkan dari hasil pelatihan. Berikut merupakan struktur dari tabel yang digunakan.

Tabel 3.3 Struktur Tabel Weight Output

Nama Kolom	Tipe Data	Keterangan
wHO_id	Integer	<i>Primary Key</i> , menyimpan ID dari setiap setiap <i>weight output</i>
row	Int	Nomor row yang berisikan nilai dari <i>weight</i>
col	Int	Nomor col yang berisikan nilai dari <i>weight</i>
Value	Float	Nilai <i>weight output</i> yang telah di dapatkan dari hasil pelatihan

Pada Tabel 3.4 digunakan untuk menampung nomor *node* dan nilai bobot *hidden bias* yang telah didapatkan dari hasil pelatihan. Berikut merupakan struktur dari tabel yang digunakan

Tabel 3.4 Struktur Tabel Hidden Bias

Nama Kolom	Tipe Data	Keterangan
id	Integer	<i>Primary Key</i> , menyimpan ID dari setiap <i>weight hidden bias</i>
node	Int	Nilai <i>node</i> sejumlah yang berisikan nilai dari <i>hidden bias</i>
Value	Float	Nilai <i>hidden bias</i> yang telah didapatkan dari hasil pelatihan

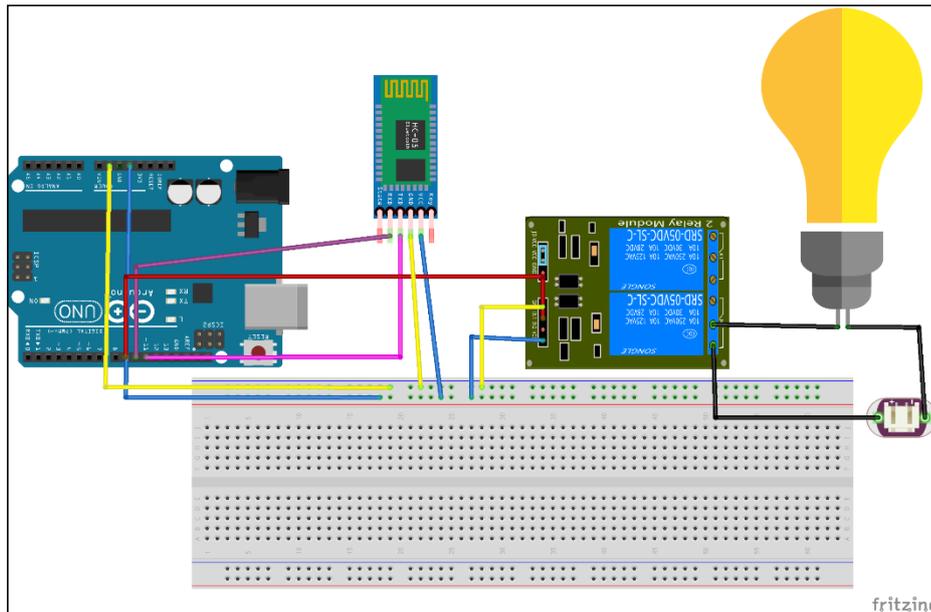
Pada Tabel 3.5 digunakan untuk menampung nomor *node* dan nilai bobot *output bias* yang telah didapatkan dari hasil pelatihan. Berikut merupakan struktur dari tabel yang digunakan

Tabel 3.5 Struktur Tabel Output Bias

Nama Kolom	Tipe Data	Keterangan
id	Integer	<i>Primary Key</i> , menyimpan ID dari setiap <i>weight output bias</i>
node	Int	Nilai <i>node</i> sejumlah yang berisikan nilai dari <i>output bias</i>
Value	Float	Nilai <i>output bias</i> yang telah didapatkan dari hasil pelatihan

3.2.9 Skema Hardware

Berikut ini merupakan skema rancangan hardware yang digunakan.



Gambar 3.14 Skema Hardware

Pada Gambar 3.14 terdapat perangkat Arduino Uno yang terhubung ke Bluetooth HC-05 yang berfungsi untuk menghubungkan dan menerima perintah dari Android ke Arduino. Modul relay yang terhubung dengan Arduino dan lampu berguna untuk mengontrol aliran listrik tegangan tinggi yang masuk.