



### **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 PENELITIAN DAN PERANCANGAN SISTEM

#### 3.1 Metodologi penelitian

Beberapa metodologi yang akan digunakan untuk merealisasikan tujuan penyusunan tugas akhir ini adalah.

##### a. Studi Literatur

Bertujuan untuk mencari dan mempelajari teori-teori dasar dari sejumlah masalah yang di bahas pada penelitian ini dan metode metode yang digunakan. Literatur yang didapat berasal dari jurnal ilmiah, *paper*, internet dan tugas akhir terdahulu.

##### b. Analisis Kebutuhan

Dalam tahap ini dilakukan analisa terhadap kebutuhan-kebutuhan apa saja yang diperlukan untuk menunjang proses penelitian, seperti *device*/alat yang akan digunakan (spesifikasi kebutuhan *software*).

##### c. Pengumpulan Sampel

Mengumpulkan beberapa sampel suara yang akan digunakan dalam proses *learning* sistem.

##### d. Perancangan dan Pembuatan

Merancang bangun aplikasi *lock screen* pada android. Dimulai dari tahap pembuatan desain, diikuti dengan pemrograman seluruh aplikasi

#### e. Pengujian Aplikasi dan *Debugging*

Melakukan pengujian apakah aplikasi dapat berjalan dengan baik dan seberapa akurat tingkat akurasi, dan melakukan *debugging* jika ditemuinya kesalahan *coding* saat pengujian

#### f. Penulisan Laporan

Bertujuan untuk memberikan data terkait penelitian termasuk tahap-tahap melakukan penelitian dan hasil penelitian

### 3.1.1 Variabel Penelitian

Variabel penelitian ini dibagi menjadi 2, yaitu variabel bebas dan variabel terikat. Variabel bebas pada penelitian ini adalah suara yang dimasukan dari *user* sedangkan variabel terikat dari penelitian ini adalah tingkat akurasi dari respon sistem pada pencocokan suara pengujian dengan suara *learning*.

### 3.1.2 Teknik Pengambilan Sampel

Seperti pada penelitian Iqbal dkk. (2011), akan diambil jumlah sampel tiap orang sebanyak 15 kali sebagai data training dan 3 data testing. Minimal jumlah orang yang akan di test sebanyak 10 orang mengikuti penelitian Iqbal (2011).

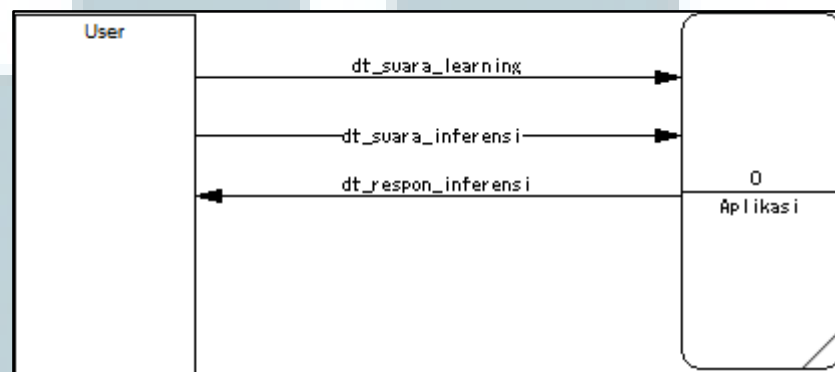
### 3.2 Perancangan Sistem

Perancangan sistem pada aplikasi ini dilakukan dengan membuat *Data Flow Diagram (DFD)*, *flowchart* dan desain antarmuka.

### 3.2.1 Data Flow Diagram (DFD)

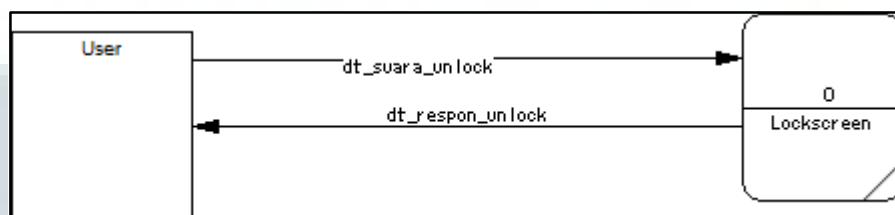
DFD adalah gambaran aliran data yang terdapat pada suatu proses. Penelitian ini memiliki DFD sebagai berikut.

#### A. DFD Level 0



Gambar 3.1 DFD Level 0 Aplikasi

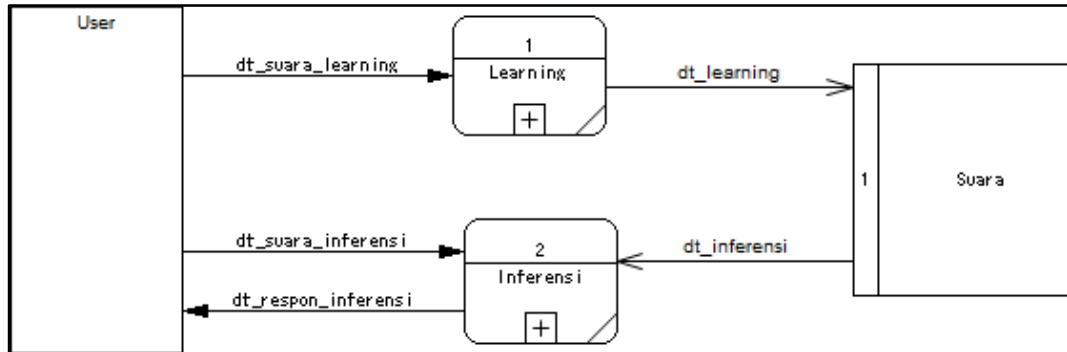
Gambar 3.1 menjelaskan aliran data yang terjadi pada bagian aplikasi. *User* akan memberikan data suara *learning* untuk dijadikan *data store* pada aplikasi ini. Lalu *User* juga dapat memberikan data suara *inferensi* untuk melakukan otentikasi saat melakukan perubahan data suara pada aplikasi ini. *User* akan menerima respon apakah data suara cocok data *training* pada aplikasi ini.



Gambar 3.2 DFD Level 0 Lockscreen

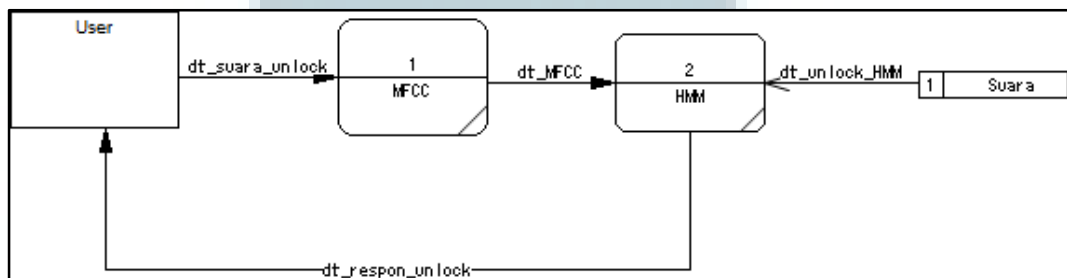
Gambar 3.2 menjelaskan aliran data yang terjadi pada *lockscreen*. *User* akan memberikan suara untuk membuka *lockscreen*. *Lockscreen* akan memberikan respon apakah suara yang diberikan *user* sesuai atau tidak dengan data *learning* yang ada.

## B. DFD Level 1



Gambar 3.3 DFD Level 1 Aplikasi

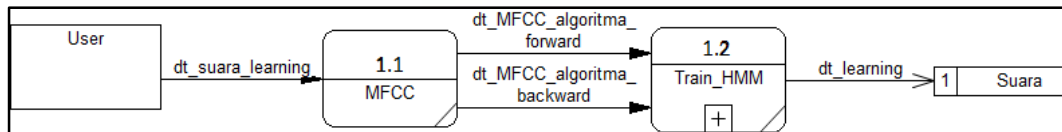
Gambar 3.3 merupakan proses aliran data yang ada didalam aplikasi ini. *User* dapat melakukan proses *learning* dengan memberikan data suaranya. Lalu data suara tersebut akan dimasukkan ke dalam *data store* berbentuk *file*. Lalu *user* juga melakukan proses inferensi saat ingin mengubah data suara yang ada. *User* akan memberikan suara yang sesuai dengan *learning*, lalu akan dicocokkan dengan *datastore* suara. Proses inferensi akan membalikkan hasil dari inferensi tersebut.



Gambar 3.4 DFD Level 1 Lockscreen

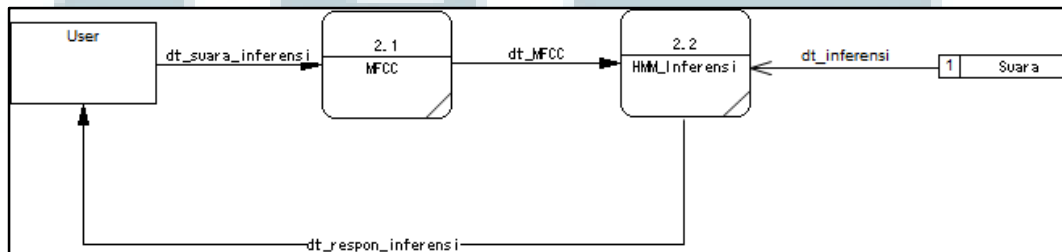
Gambar 3.4 merupakan aliran data pada *lockscreen*. Disini suara yang *user* berikan akan diubah menjadi bentuk MFCC. Lalu suara yang telah diubah dalam MFCC akan dicocokkan dengan *datastore* suara melalui HMM. Hasil dari pencocokkan akan memberikan respon kepada *user* apakah suara yang *user* berikan sesuai dengan suara *learning* pada *datastore* suara.

### C. DFD Level 2



Gambar 3.5 DFD Level 2 Aplikasi

Gambar 3.5 merupakan aliran data dari proses *learning* pada aplikasi. Suara *learning* diubah menjadi *format* MFCC. Setelah diubah, maka data MFCC akan dimasukkan kedalam proses *train* HMM untuk menentukan probabilitas dari tiap sinyal suara. Data tersebut akan dimasukkan ke dalam *datastore* suara.

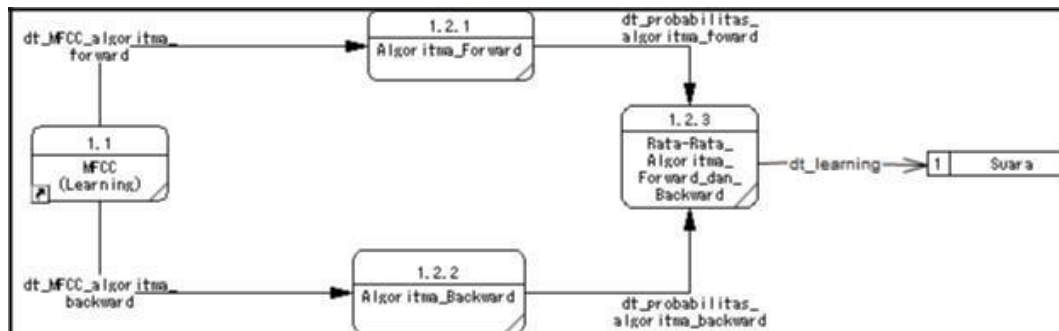


Gambar 3.6 DFD Level 2 Inferensi Aplikasi

Gambar 3.6 menjelaskan aliran data dari proses inferensi aplikasi. Suara akan diubah menjadi *format* MFCC. Lalu suara akan dicocokkan dengan *datastore* suara dalam proses inferensi HMM. Setelah itu dicocokkan, maka hasilnya akan diberikan kepada *user*.

U  
M  
M  
N

### D.DFD Level 3



Gambar 3.7 DFD Level 3 *Train HMM*

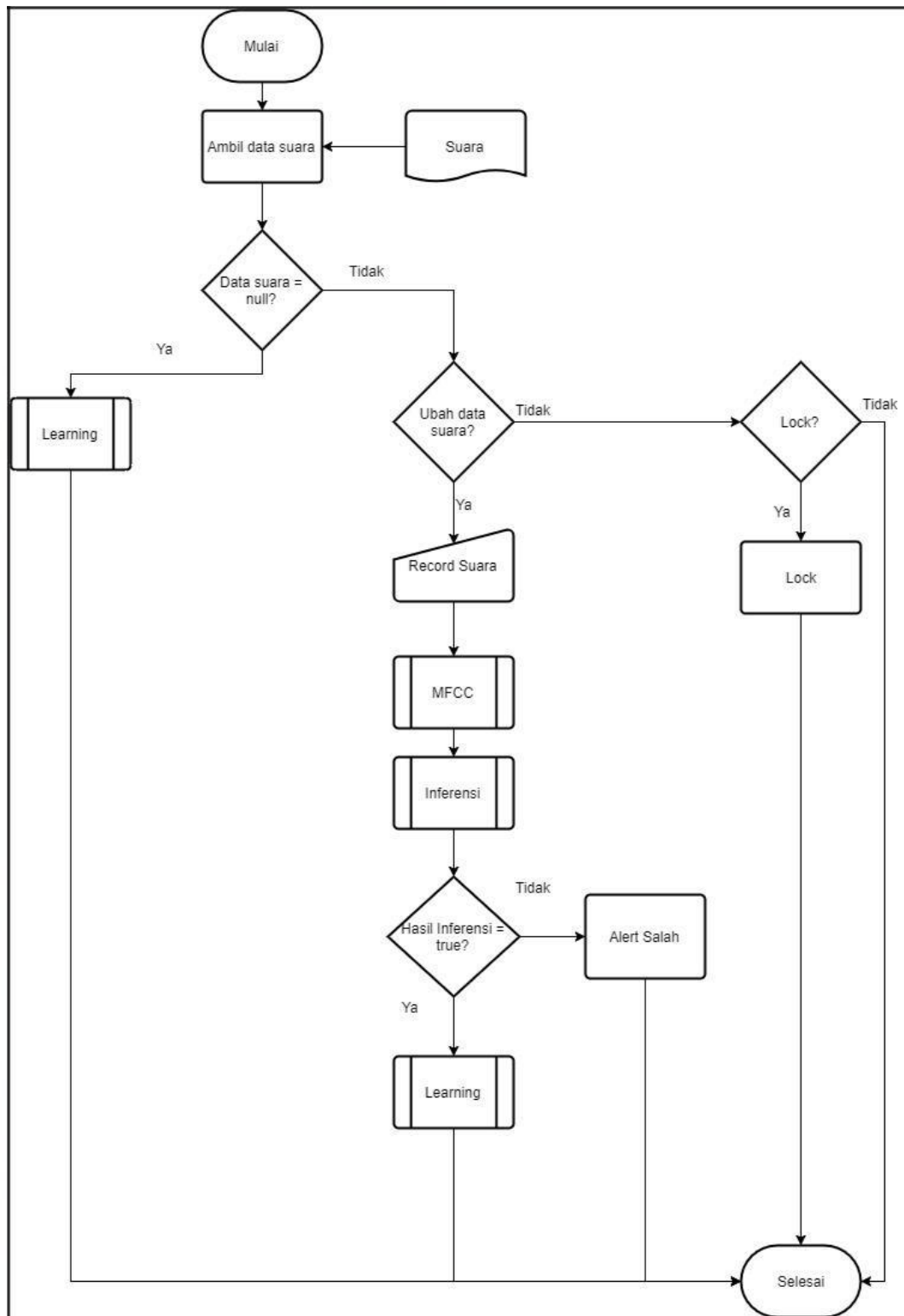
Gambar 3.7 Merupakan aliran data yang terjadi dalam *train HMM*. Disini data yang didapat dari MFCC diproses kedalam Algoritma *Forward* dan *Backward*. Hasil dari kedua algoritma ini menghasilkan probabilitas yang akan di rata-rata dan dimasukkan kedalam *datastore* suara

### 3.2.2 Flowchart

*Flowchart* adalah alur proses yang ada di dalam suatu sistem. Dalam penelitian ini terdapat dua *flowchart* utama yaitu *flowchart* aplikasi dan *flowchart* *lockscreen*. *Flowchart* aplikasi terdiri dari *flowchart* *learning*, *flowchart* MFCC, *flowchart* *train HMM*, dan *flowchart* inferensi, *flowchart* Algoritma *Forward* dan *Backward*.

U  
M  
M  
N

## A. Flowchart Aplikasi

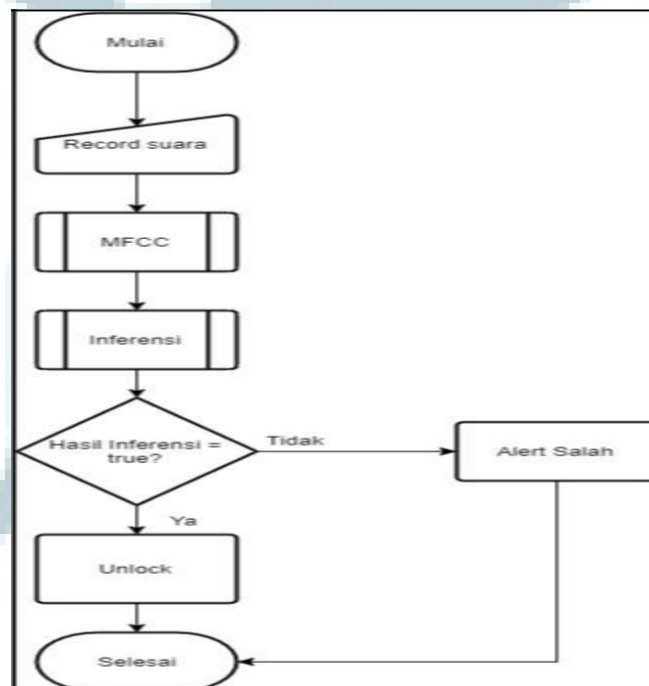


Gambar 3.8 Flowchart Aplikasi



Gambar 3.8 menggambarkan alur dari aplikasi yang dibuat. Saat aplikasi dijalankan, aplikasi akan melakukan pemeriksaan apakah isi data suara pada *mobile device* bernilai *null* atau tidak. Jika data suara bernilai *null*, maka aplikasi akan melakukan *learning*. Jika data suara tidak bernilai *null* saat aplikasi dijalankan, maka *user* dapat memilih untuk mengubah data suara atau melakukan penguncian. Jika *user* memilih untuk mengubah data suara, maka *user* akan diminta kata kunci suara sebelumnya. Jika hasil inferensi benar, maka *user* akan diminta melakukan *learning* untuk data suara yang baru. Jika inferensi salah maka akan muncul *alert* yang menunjukkan suara yang diucapkan tidak sesuai dengan data suara. Jika *user* memilih untuk melakukan *lock*, aplikasi akan melakukan *lock* pada *mobile device* tersebut.

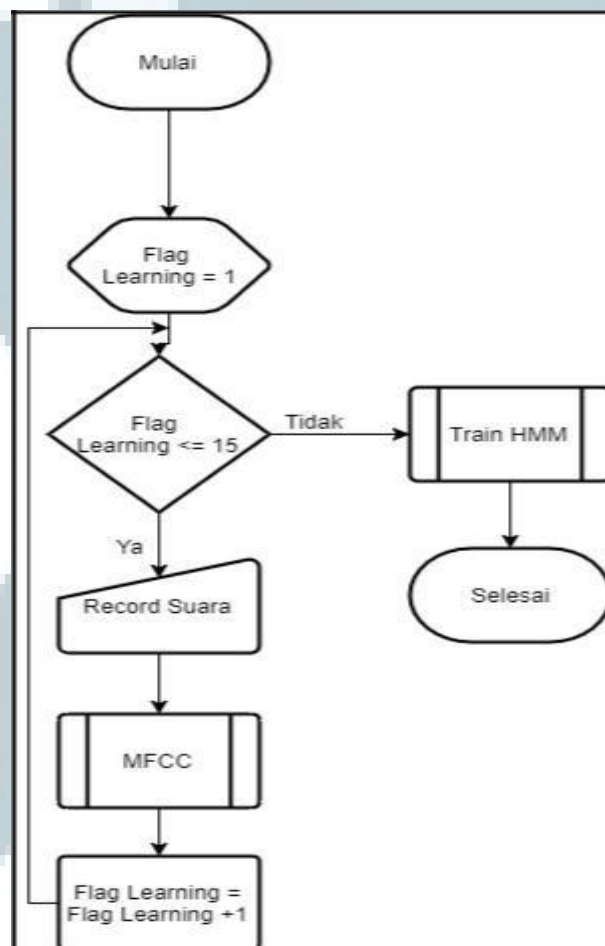
## B. Flowchart Lockscreen



Gambar 3.9 Flowchart Lockscreen

Gambar 3.9 merupakan *flowchart* dari *lockscreen* yang dirancang. Pertama *user* akan merekam suara untuk membuka *lockscreen*. Setelah direkam, suara tersebut diubah kedalam *format* MFCC. Sesudah itu nilai tersebut akan diinferensikan apakah sesuai dengan data suara yang dimiliki. Jika inferensi bernilai *true*, maka *lockscreen* akan terbuka. Jika hasil inferensi bernilai *false*, maka akan muncul *alert* yang memberikan bahwa suara tidak sesuai dengan data suara atau salah.

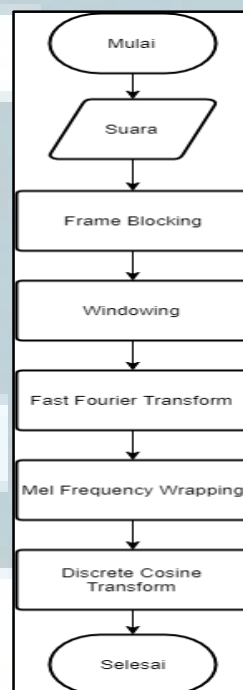
### C. Flowchart Learning



Gambar 3.10 *Flowchart Learning*

Gambar 3.10 merupakan *flowchart* bagaimana proses *learning* dilakukan. *Learning* dilakukan sebanyak 15 kali. Setiap *learning* akan merekam suara yang akan dimasukkan ke dalam data suara. Suara yang masuk diubah ke dalam *format* MFCC. Setelah 15 kali *learning*, semua suara yang sudah di *format* akan di *train* dengan HMM untuk dimasukkan ke dalam data suara.

#### D. Flowchart MFCC

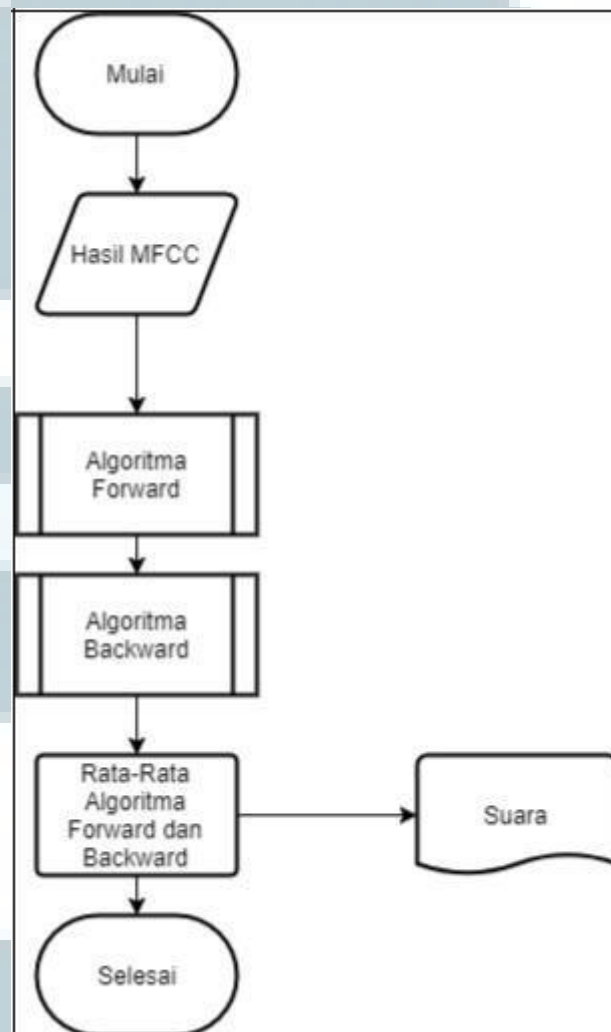


Gambar 3.11 *Flowchart* MFCC

Gambar 3.11 menjelaskan alur dari perubahan suara kedalam *format* MFCC. Suara yang dihasilkan akan melalui proses *frame blocking* yaitu memisahkan suara kedalam beberapa *frame*. Setiap *frame* yang didapat akan melalui proses *windowing* yaitu menghilangkan *aliasing*. Setelah itu suara akan diubah dalam satuan frekuensi dengan menggunakan *Fast Fourier Transform*. Setelah itu, suara akan diubah ke

skala mel dan akan dilakukan *Discrete Cosine Transform* untuk mendapatkan nilai dari tiap *frame* dalam domain waktu yang menotasikan suara tersebut.

#### E. Flowchart Train HMM

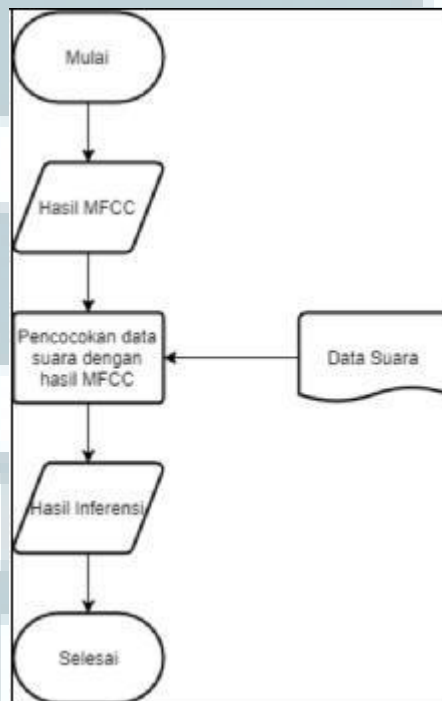


Gambar 3.12 *Flowchart Train HMM*

Gambar 3.12 menjelaskan alur *training* suara dengan HMM. Pertama data suara akan dimasukan ke dalam Algoritma *Forward* untuk mendapatkan probabilitas tiap *frame* suara yang ada. Setelah itu data suara akan dimasukan juga ke dalam Algoritma *Backward* untuk mendapatkan probabilitasnya. Setelah kedua

probabilitas tersebut didapatkan, maka kedua probabilitas itu akan dikali untuk menjadi probabilitas pada data suara yang menjadi *training*.

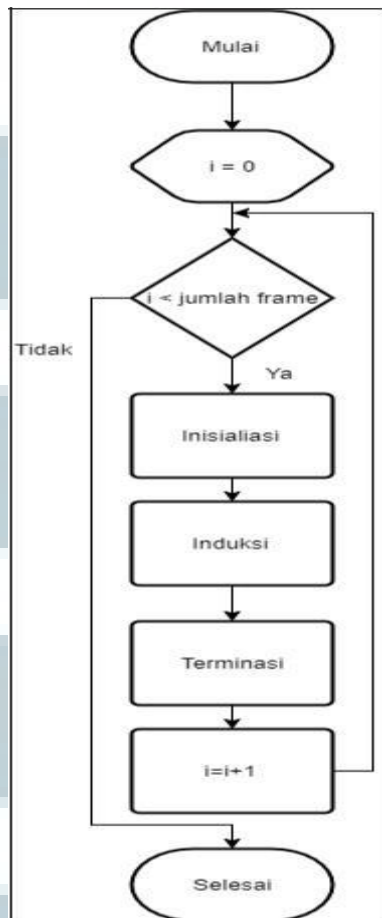
#### F. Flowchart Inferensi



Gambar 3.13 *Flowchart Inferensi*

Gambar 3.13 menjelaskan alur dari inferensi dari suara yang dicocokkan dengan data suara. Frame suara akan dicocokkan dengan data suara training yang ada. Setiap frame yang cocok dengan data training akan dijumlahkan lalu dibagi dengan total frame suara yang dihasilkan. Jika presentase nya menunjukkan diatas 50%, bisa dikatakan suara tersebut valid dengan kata kunci pada training (Ferdyanto, 2014).

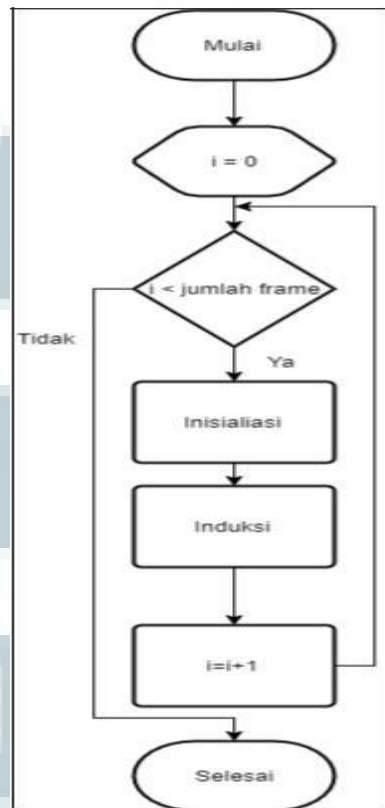
## G. Flowchart Algoritma Forward



Gambar 3.14 *Flowchart Algoritma Forward*

Gambar 3.14 merupakan alur proses dari Algoritma Forward. Data yang sudah didapat dari MFCC akan diproses tiap frame. Tiap frame akan mengalami proses inisialisasi untuk mengetahui probabilitas awal tiap sampel sinyal. Setelah itu akan dimasukkan ke dalam proses induksi untuk menentukan kemungkinan output dalam satu framenya. Setelah itu proses terminasi yang digunakan untuk menentukan nilai probabilitas tiap sampel sinyal nya.

## H. Flowchart Algoritma Backward



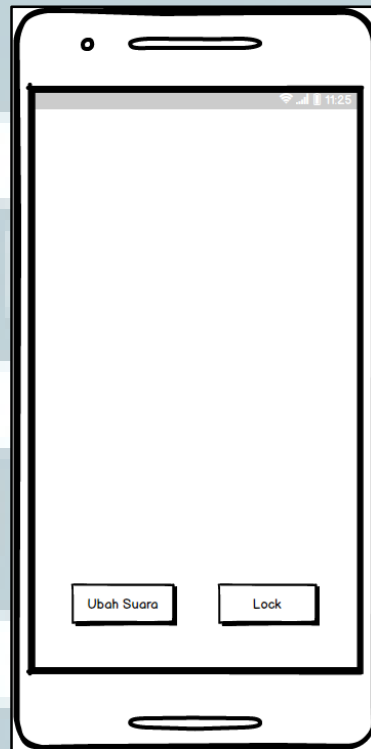
Gambar 3.15 Flowchart Algoritma Backward

Gambar 3.15 menggambarkan alur proses dari Algoritma Backward. Sinyal diproses tiap framanya. Pertama akan dilakukan proses inisialisasi dimana nilai 29 probabilitas sinyal dalam frame adalah satu. Lalu dilanjutkan dengan proses induksi dimana setiap sampel sinyal indeks terakhir dihilangkan dan dicari probabilitasnya

### 3.2.3 Desain Antarmuka

Desain antarmuka pada penelitian ini dapat digambarkan sebagai berikut.

#### A. Desain Antarmuka Aplikasi



Gambar 3.16 Desain Antarmuka Halaman Utama Aplikasi

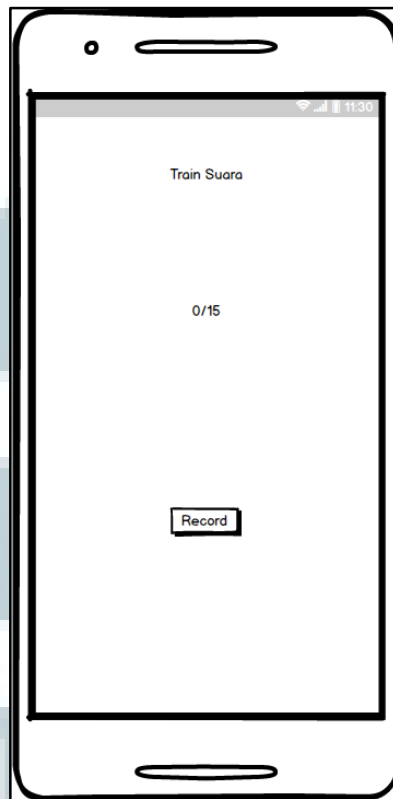
Gambar 3.16 menggambarkan desain antarmuka halaman utama aplikasi ini. Tombol sebelah kiri adalah tombol ubah data suara dan tombol sebelah kanan merupakan tombol *lock*. Saat tombol ubah suara ditekan, maka *user* akan diminta untuk memasukkan suara untuk diinferensikan dengan data suara. Desain saat *user* akan memasukkan suara, digambarkan seperti Gambar 3.17.





Gambar 3.17 Desain Antarmuka Halaman Inferensi Aplikasi

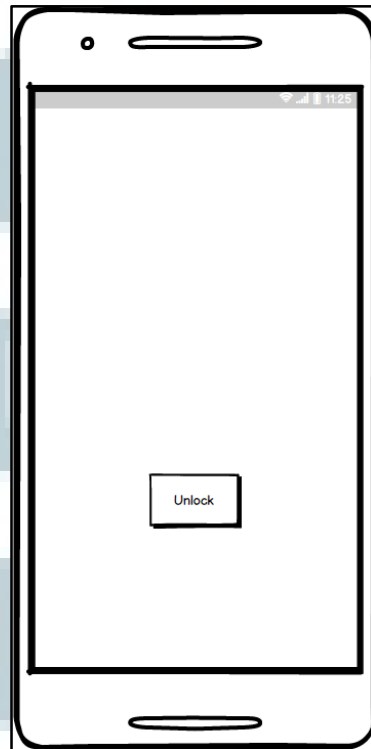
Pada Gambar 3.17, terlihat sebuah tombol yang jika ditekan akan merekam suara *user* yang akan diinferensikan dengan data suara yang ada. Jika hasil inferensi adalah salah, maka akan muncul *alert* yang mengatakan bahwa data suara salah. Jika Inferensi benar, maka halaman akan dipindahkan ke halaman *training*. Halaman ini digambarkan pada Gambar 3.18.



Gambar 3.18 Desain Antarmuka Halaman *Training*

Gambar 3.18 menggambarkan halaman *training* pada aplikasi. Tombol record akan memulai merekam suara untuk di *training*. Setelah suara pertama selesai diucapkan, maka *counter* pada label akan bertambah satu. Setelah proses *training* telah berjalan 15 kali, maka halaman ini akan berpindah kembali ke halaman utama pada Gambar 3.16. Gambar 3.18 ini juga merupakan tampilan saat aplikasi belum memiliki data suara atau saat pertama kali di *install*.

## B. Desain Antarmuka Lockscreen



Gambar 3.19 Desain Antarmuka *Lockscreen*

Gambar 3.19 menggambarkan desain antarmuka pada *lockscreen*. Pada desain ini, hanya terdapat satu tombol untuk merekam suara. Jika suara yang direkam sesuai dengan data suara yang ada, maka *lockscreen* akan dimatikan.

U  
M  
M  
N