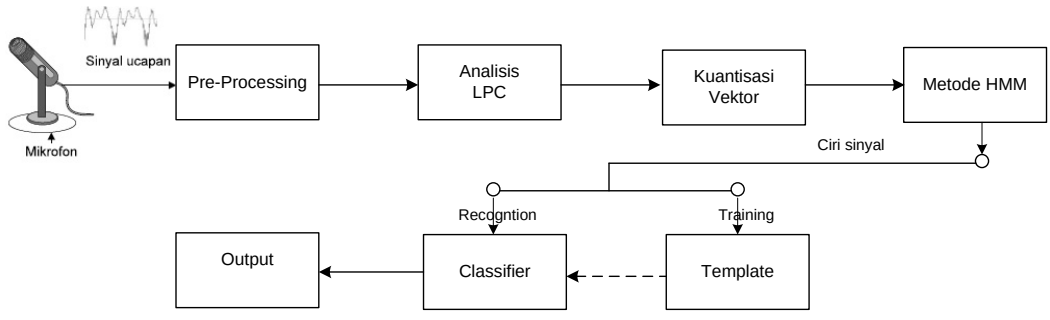
BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1. Deskripsi Umum Sistem

Proses pengidentifikasian kata pada sistem dapat dijelaskan sebagai berikut pertama-tama pengguna menentukan ucapan yang ingin dikenali pengguna kemudian mengucapkannya pada mikrofon pada bagian pelatihan sistem akan mengekstraksi sinyal suara dengan ekstraksi cirri dan menimpanannya sebagai *template.* Setelah itu barulah sistem akan mengenali ucapan dari pengguna karena sistem telah memiliki *template* yang akan dijadikan perbandingan. Bila pengguna menjalankan bagian pengenalan maka sistem akan mengekstraksi sinyal suara dari mikrofon kemudian akan dibandingkan dengan *template*. Sistem kemudian akan mengeluarkan keluaran berdasarkan sinyal suara pada bagian pengenalan terhadap template dimana nilai probabilatas yang terbesar maka itu akan menjadi keluaran yang tepat.



Gambar 3.1. proses pengeidentifikasian kata

Gambar 3.1. menunjukan garis besar proses pengidentifikasian kata yang dirancang. Tahap *pre-processing* adalah proses pengkondisian sinyal suara

menggunakan filter. Tahap analisis LPC adalah mengekstraksi sinyal data sinyal yang diinputkan. Tahap Kuantisasi vektor adalah proses pengkodean ciri-ciri vektor sinyal kedalam bentuk simbol HMM. Tahap HMM adalah model statistik dimana mempunyai keluaran rangkaian simbol dan kuantitas. Dengan memiliki sebuah model yang memberikan kemungkinan dari rangkaian akustik data yang telah diobservasi dari sebuah atau banyak kata (rangkaian kata) akan dapat menyebabkan sistem bekerja dengan rangkaian kata tersebut sesuai dengan aplikasi aturan. Tahap pelatihan adalah pengestimasian ulang parameter HMM dari suatu data berdasarkan data latih yang diajukan pengguna, sehingga hasil *estimasi* memilki kualitas yang baik yang kemudian dijadikan *template.* Tahap pengenalan adalah perhitungan *probabilitas* kemiripan pola dari tiap-tiap model HMM yang mempunyai *probabilitas* kemiripan yang tertinggi dan berdasarkan pada *template.*

3.2. Analisis Sistem

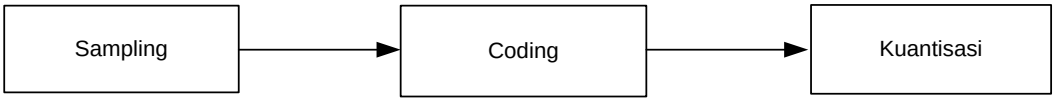
Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh kedalam bagian-bagian komponennnya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

Dalam proses pembuatan suatu sistem mutlak dilakukan penelitian dan penganalisaan tetang sistem yang akan dibangun, berikut adalah beberapa analisis yang dilakukan untuk membangun Aplikasi Identifikasi Kata Menggunakan

Metode HMM *(Hidden Markov Model)* dengan Ekstraksi Ciri LPC *(Linear Predective Coding).*

3.2.1. Analisis Suara Secara Digital

Sinyal suara yang akan diproses bersifat analog sehingga jika akan dilakukan pengolahan secara digital, sinyal suara tersebut harus dikonversi menjadi sinyal digital, berupa urutan angka dengan tingkat presisi tertentu yang dinamakan analog to digital conversion dengan menggunakan *analog-to-digital converter ADC.* Konsep Kerja ADC terdiri dari tiga proses :



Gambar 3.2. konsep kerja ADC (Analog to Digital Converter)

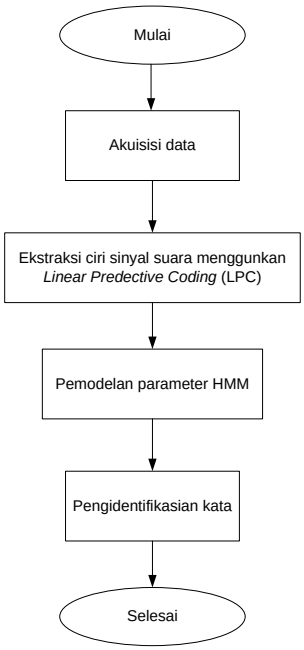
Keterangan konsep kerja ADC :

1. Sampling adalah konversi sinyal kontinu dalam domain waktu menjadi sinyal diskrit, melalui proses sampling sinyal pada selang waktu tertentu. Sehingga jika $x_0(t)$ adalah sinyal input, maka outputnya adalah $x_0(nT)$, dengan T adalah interval sampling.
2. Kuantisasi adalah konversi dari descrete time, continous value, menjadi decrete time dan continous value.
3. Coding, pada proses ini, tiap nilai diskrit yang telah didapat, direpresentasikan dengan angka binary n-bit.

3.2.2. Analisis Pengenalan Suara

Dalam analisis pengenalan suara ini dibagi menjadu 5 tahap, tahap pertama adalah akuisisi data, tahap kedua adalah proses ekstraksi cirri sinyal suara menggunakan LPC, tahap ketiga adalah proses pemodelan parameter HMM dan

tahap yang keempat adalah pengenalan suara. Secara umum program ini mengikuti alur sesuai yang ditunjukan pada gambar 3.3.



Gambar 3.3. Alur Program pengidentifikasian kata

3.2.2.1. Akuisisi Data

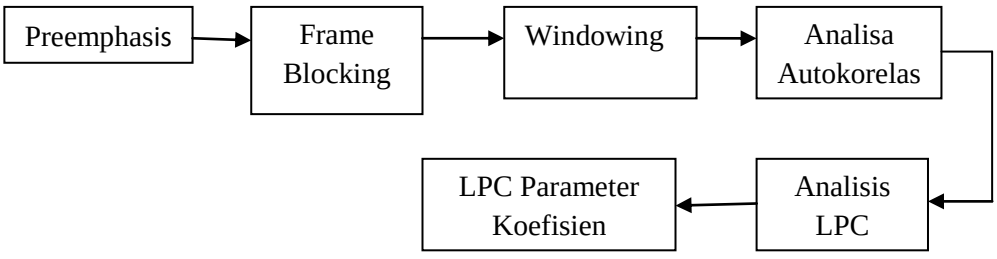
Data berupa sinyal suara diperoleh dengan cara merekam suara melalui mikropon yang dihubungkan dengan komputer pribadi. Perekeman suara dilakukan dengan bantuan program aksesories *windows,* yaitu *sound recorder* dengan frekuensi *sampling* 8000 Hz, 8 bit, mono.

Suara diucapkan oleh 20 orang yang terdiri dari sepuluh orang pria dan sepuluh orang wanita dimana setiap kata diulang sebanyak 10 kali. Dengan sound recorder suara direkam dan berkas suara dibuka kembali dengan cool edit pro untuk kemudian disimpan dalam berkas dengan format 123.wav. dimana angka 1 menunjukan nama pembicara, angka 2 menunjukan kata yang diucapkan dan

angka 3 munujukan suara yang ke (1-10). Kata yang akan dijadikan sample adalah fakultas, jurusan, teknik, informatika, universitas, komputer, Indonesia, bandung.

3.2.2.2. Analisis Ekstraksi Ciri LPC

Analisis sinyal adalah kegiatan melakukan ekstraksi terhadap informasi yang terdapat di suatu sinyal. *Linear Predictive Coding* (LPC) merupakan salah satu teknik analisis sinyal percakapan yang paling *powerful* dan menyediakan ekstraksi fitur yang berkualitas baik dan efisien untuk digunakan dalam perhitungan. LPC pertama kali digunakan pada tahun 1978 untuk membuat alat sintesis sinyal percakapan. LPC melakukan analisis dengan cara memperkirakan *formant*, memisahkan *formant* dari sinyal, yang dinamakan proses *inverse filtering*, lalu mengestimasi intensitas dan frekuensi dari sinyal percakapan yang tersisa, yang disebut *residue.16* Karena sinyal percakapan bervariasi seiring waktu, estimasi tersebut dilakukan untuk setiap potongan kecil dari sinyal, yang dinamakan *frame.* Prosedur untuk mendapatkan koefisien LPC diperlihatkan pada gambar dibawah.



Gambar 3.4 Block Diagram LPC

Gambar diatas menunujukan blok diagram LPC yang biasa digunakan dalam sistem pengenalan suara manusia. Langkah-langkah dasar yamg harus dilakukan mengenai blok diagram tersebut adalah :

1. *Preemphasis* adalah suatu proses produksi suara manusia, radiasi pada bibir dan lidah ketika proses phonation mengakibatkan komponen frekuensi tingginya. Hal ini dapat berpengaruh pada proses spectrum frekuensi sinyal menjadi merata antara frekuensi tinggi dilakukan pemfilteran dengan menggunakan *high pass filter*. Dimana *High pass filter* adalah jenis filter yang melewatkan frekuensi tinggi, tetapi mengurangi amplitudo frekuensi yang lebih rendah daripada frekuensi cutoff.Nilai-nilai pengurangan untuk frekuensi berbeda-beda untuk tiap-tiap filter ini .Terkadang filter ini disebut *low cut filter*, *bass cut filter* atau rumble filter yang juga sering digunakan dalam aplikasi audio.

Preemphasis terhadap cuplikan sinyal dengan persamaan *preemphasizer:*

$$s(n) = s(n) - as(n-1) \quad (1)$$

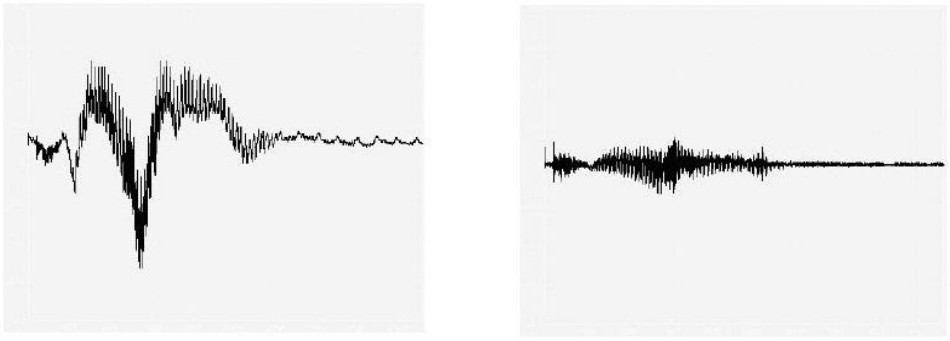
dengan *s*(*n*) adalah sampel ke-n dan harga *a* yang paling sering digunakan ialah 0.95.

Algoritma Procedure Preemphasis

Keterangan :

PSinyal disini adalah Panjang Sinyal yang akan menjadi acuan untuk *procedure preemphasis.*

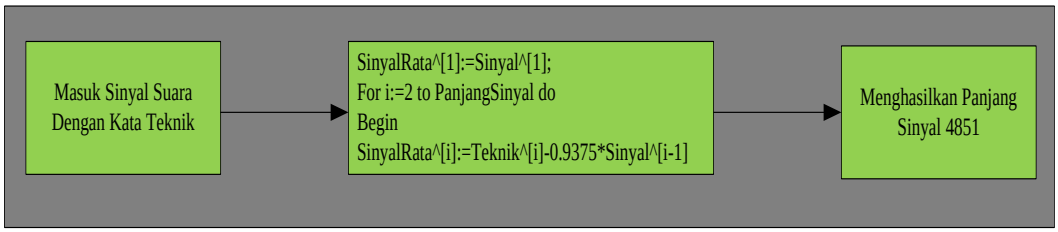
```
Procedure preemphasis (sinyal:Psinyal;var sinyalrata:Psinyalrata);
Kamus
    i: Integer
Algoritma
    sinyal rata^[1]←sinyal^[1]
    for i←2 panjang sinyal do
    begin
        sinyalrata^[i]←sinyal^[i]-0.95*sinnyal^[i-1]
    endfor
```



a. Sebelum preemphasis b. Sesudah preemphasis

Gambar 3.5. Sebelum dan sesudah preemphasis

Alur masukan suara dengan kata teknik dimulai dari sinyal suara masuk, proses, dan hasil dari masukan sinyal suara pada proses *pre emphasis.*



Keterangan : Untuk hasil dari panjang sinyal lebih detailnya dapat dilihat halaman lampiran b.

Analisis Prosedur *Preemphasis* :

```
procedure SA_preemphasis(Sinyal:PSinyal; var SinyalRata:PSinyalRata);
var i:integer;
begin
  SinyalRata^[1]:=Sinyal^[1];
  for i:=2 to PanjangSinyal do
  begin
    SinyalRata^[i]:=Sinyal^[i]-0.9375*Sinyal^[i-1];
  end;
end;
```

2. *Blocking into frames* adalah suatu proses pada tahapan sinyal yang telah di *preemphasis*, diblok menjadi beberapa bagian dengan jumlah sample *N*, dan tiap bagian dipisahkan dengan sejumlah M sample.

Membagi hasil *preemphasis s* (*n*) ke dalam *frame-frame* yang masing-masing memuat *N* buah sampel yang dipisahkan sejauh *M* buah *sample*. Semakin semakin baik perkiraan spektral LPC dari *frame* ke *frame.*

$$s(n)=M < N$$

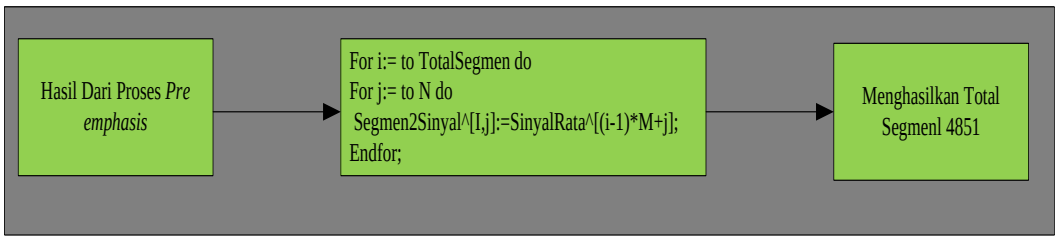
Algoritma Block In To Frame

Keterangan :

PSegmen disini adalah Panjang Segmen dimana panjang segmen ini berfungsi untuk mengetahui maksimum dan nilai-nilai puncak minimum pada selang pengamatan Suara *(1 frame.*)

```
Procedure framing (sinyalrata:Psinyalrata);var
Segmen2sinyal:Psegmen2sinyal;
Kamus
        i,j: Integer
Algoritma
        for i←1 to totalsegmen do
        begin
                for j←1 to N do
                segmen2sinyal^[i,j]←sinyalrata^[(i,j)*M+j]
         endfor
```

Alur masukan suara setelah dilakukan proses *pre emphasis* dengan kata teknik dimulai Masukan hasil dari *pre emphasis*, proses, dan hasil total segmen yang dihasilkan pada proses *Blocking In To Frame.*



Keterangan : Untuk hasil dari total segmen lebih detailnya dapat dilihat halaman lampiran b.

Analisis Prosedur *Block In To Frame* :

```
Procedure SA_framing(SinyalRata:PSinyalRata;
                     var Segmen2Sinyal:PSegmen2Sinyal);
var i,j:integer;
begin
  for i:=1 to TotalSegmen do
    for j:=1 to N do
      Segmen2Sinyal^[i,j]:=SinyalRata^[(i-1)*M+j];
end;
```

3. *FrameWindowing* adalah suatu proses pada setiap window pada bagian sinyal yang telah dibuat sebelumnya. Hal ini dilakukan untuk meminimalkan pada bagian awal dan akhir sinyal. Jika definisikan sebuah *window w(n)* dan sinyal tiap bagian adalah x(n) maka sinyal hasil proses *windowing*.

Melakukan *windowing* terhadap setiap *frame* yang telah dibentuk untuk meminimalkan diskontinuitas pada ujung awal dan ujung akhir setiap *frame* dengan persamaan *Hamming Window* untuk sampel ke-n adalah :

$$W(n) = 0.54 - 0.46 \cos(2\pi n/N-1),\ 0 \leq n \leq N-1 \qquad (2)$$

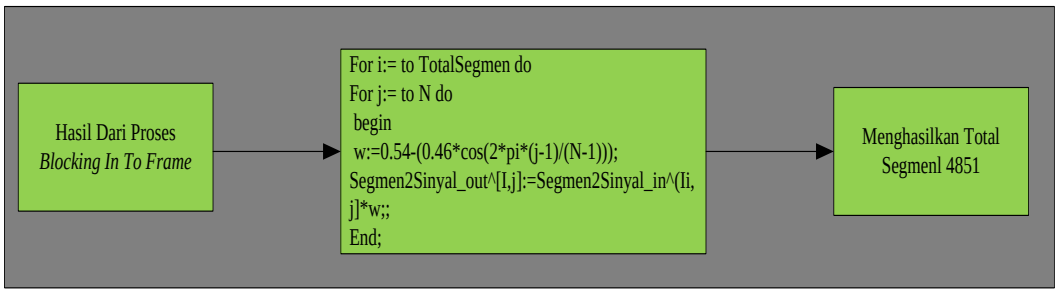
Algoritma Procedure Windowing

Keterangan :

tReal LPC disini adalah sebuah waktu yang dikondisikan untuk sebuah ekstraksi ciri LPC.

```
Procedure windowing (segmen2sinyal_in:Psegmen2sinyal;var
Segmen2sinyal_out:Psegmen2sinyal);
Kamus
        i,j: Integer
        w:tReal LPC
Algoritma
        for i←1 to totalsegmen do
        begin
              for j←1 to N do
               begin
                     w←0.54-(0.46*cos(2*pi*(j-1)/(N-1)))
                     segmensinyal_out^[i,j]←segmen sinyal2_in^[i,j]*w
              endfor
         endfor
```

Alur masukan suara setelah dilakukan proses *Blocking In To Frame* dengan kata teknik dimulai Masukan hasil dari *Blocking In To Frame*, proses, dan hasil total segmen yang dihasilkan pada proses *Frame Windowing*.



Keterangan : Untuk hasil dari total segmen lebih detailnya dapat dilihat halaman lampiran b.

Analisis Prosedur *Windowing* :

```
procedure SA_windowing(Segmen2Sinyal_in:PSegmen2Sinyal;
                       var Segmen2Sinyal_out:PSegmen2Sinyal);
var i,j:integer;
   w:tRealLPC;
```

```
begin
  for i:=1 to TotalSegmen do
    for j:=1 to N do
    begin
      w:=0.54-(0.46*cos(2*pi*(j-1)/(N-1)));
      Segmen2Sinyal_out^[i,j]:=Segmen2Sinyal_in^[i,j]*w;
    end;
end;
```

4. *Auto Correlation Analisys* adalah proses tahapan yang setiap bagian yang telah diberi *window* kemudian akan dibentuk autokorelasinya.

Analisis autokorelasi terhadap setiap *frame* hasil *windowing* x_1 (*n*) dengan persamaan :

$$r_1(m)=\sum^{n-1-m}_{n=0} x_1(n)x_1(n+m) \quad (3)$$

Contoh Algoritma Auto Korelasi

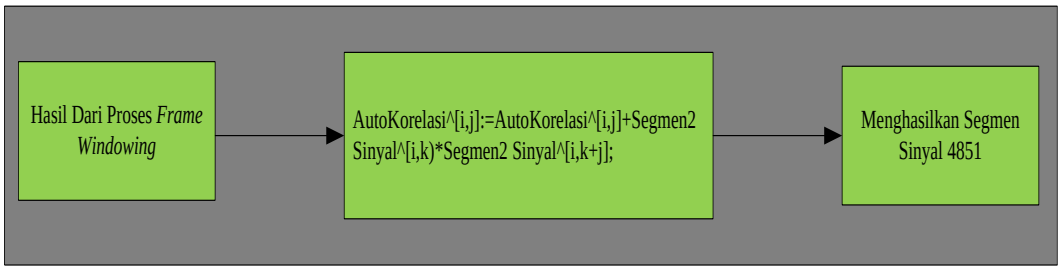
Keterangan :

Pmultivektor disini pendefinisan dari panjang dari sebuah vektor-vektor yang terdapat pada proses analisis auto korelasi, dan AK disini adalah inisialisasi dari Auto Korelasi.

```
Procedure auotocorrelation (segmen2sinyal_in:Psegmen2sinyal;var
Autokorelasi:Pmultivektor AK);
Kamus
        i,j,k: Integer
Algoritma
        for i←1 to totalsegmen do
        begin
                for j←0 to orderLPC do
                begin
                        Autokorelasi^[i,j]←0
                         for k←1 to N-j do
```

```
                    begin
                        Autokorelasi^[i,j]←Autokorelasi^[i,j]+
                        segmen2sinyal ^[i,k]*segmen2sinyal ^[i,k+j]
                    endfor
                endfor
            endfor
```

Alur masukan suara setelah dilakukan proses *Frame Windowing* dengan kata teknik dimulai Masukan hasil dari *Frame Windowing*, proses, dan hasil total segmen yang dihasilkan pada proses AutoKorelasi.



Keterangan : Untuk hasil dari total segmen lebih detailnya dapat dilihat halaman lampiran b.

Analisis Auto Korelasi :

```
Procedure SA_autocorrelation(Segmen2Sinyal:PSegmen2Sinyal;
                             var AutoKorelasi:PMultiVektorAK);
var i,j,k:integer;
begin
  for i:=1 to TotalSegmen do
    for j:=0 to OrdeLPC do
    begin
      AutoKorelasi^[i,j]:=0;
      for k:=1 to N-j do
      begin

AutoKorelasi^[i,j]:=AutoKorelasi^[i,j]+Segmen2Sinyal^[i,k]*Segmen2Sinyal^[i,
k+j];
      end;
    end;
end;
```

5. Analisa LPC adalah suatu proses analisis sinyal dimana semua nilai aotukorelasinya yang telah dihitung pada tahap sebelumnya akan diubah menjadi parameter LPC.

Mengubah p +1 buah hasil autokorelasi pada masingmasing *frame* menjadi koefisien LPC $a_m = \alpha_m^{(p)}$ untuk m = 1,2,..., p dengan persamaan dibawah ini:

$$E^{(o)}=r(0) \quad (4)$$

$$k_m=\{r(m)-\Sigma^{m-1}_{i=1}\alpha_i^{(m-1)}r(|m-j|)/E^{(m-1)},\ 1 \le m \le p \quad (5)$$

$$\alpha_m^{(m)}= k_m \quad (6)$$

$$\alpha_j^{(m)} = \alpha_j^{(m-1)}- k_m\, \alpha_{m-j}^{(m-1)},\ 1 \le j \le m-1 \quad (7)$$

$$E^{(m)}=(1- k_m^2)\, E^{(m-1)} \quad (8)$$

dengan $r(0)$adalah hasil autokorelasi, $E_{(m)}$ adalah *error*, k_m adalah koefisien pantulan, $a_j^{(m)}$ adalah koefisien prediksi untuk $1 \le j \le m$.

Algoritma Analisa LPC

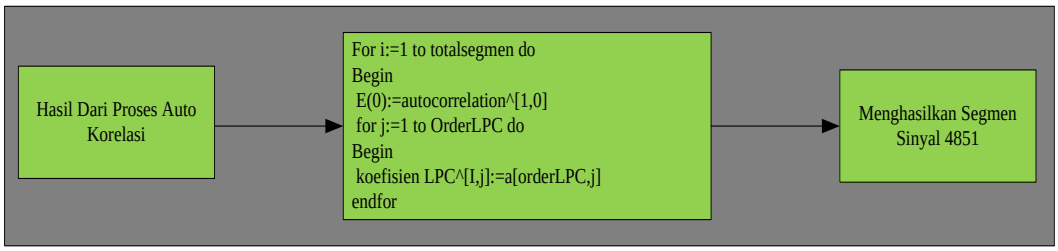
Keterangan :

TVektor tipe data dari sebuah total nilai dari sebuah vektor yang diperlukan pada vektor-vektor yang terdapat pada LPC.

```
Procedure LPC(autocorrelation:Pmultivektor AK;var
Koefisien LPC:Pmultivektor);
Kamus
    i,j: Integer
    E:Tvektor AK
    K:Tvektor
    A:array [0…orderLPC] of Tvektor
Algoritma
     for i←1 to totalsegmen do
     begin
          E[0] ←autocorrelation^[1,0]
          for j←1 to orderLPC do
```

```
        begin
            if (j=1) then
                K[1]←autocorrelation^[i,1]/ autocorrelation^[i,0]
             else
                begin
                  tmp←0
                    for i←1 to j-1 do
                      tmp←tmp+a[j-1,i]*autocorrelation^[i,(j-i)]
                      k[j]←( autocorrelation^[i,j]-tmp)/E[j-i]
                end
                a[i,j]
                  if j>=2 then
                    for i←j-1 down to i  do
                    begin
                      a[j,i]←a[j-1,i]-k[j]*a[j-1,j-i]
                    end
                    E[j]←(1-k[j]*k[j])*E[j-1]
                  end
                  for j←1 to orderLPc do
                  begin
                    koefisien LPC^[i,j] ←a[orderLPC,j]
        endfor
    endfor
```

Alur masukan suara setelah dilakukan proses AutoKorelasi dengan kata teknik dimulai Masukan hasil dari AutoKorelasi, proses, dan hasil total segmen yang dihasilkan pada proses Analisa LPC.



Keterangan : Untuk hasil dari total segmen lebih detailnya dapat dilihat halaman lampiran b.

Analisis Prosedur Analisa LPC :

```
Procedure SA_LPC(AutoKorelasi:PMultiVektorAK;
                 var KoefisienLPC:PMultiVektor);
var i,j,l:integer;
   E:TVektorAK;
   k:TVektor;
   a:array [0..OrdeLPC] of TVektor;
   tmp:tRealLPC;
begin
  for i:=1 to TotalSegmen do
  begin
   E[0]:=AutoKorelasi^[i,0];
   for j:=1 to OrdeLPC do
   begin
    if (j=1) then
     //if(AutoKorelasi^[i,0]<>0)
     then k[1]:=AutoKorelasi^[i,1]/AutoKorelasi^[i,0]
     //else k[1]:=AutoKorelasi^[i,1]
     k[1]:=AutoKorelasi^[i,1]/AutoKorelasi^[i,0]
    else
    begin
     tmp:=0;
     for l:=1 to j-1 do
       tmp:=tmp+a[j-1,l]*AutoKorelasi^[i,abs(j-l)];
     k[j]:=(AutoKorelasi^[i,j]-tmp)/E[j-1];
     //kode dibawah sengaja dicomment, karena ternyata seharusnya memang
tidak ada E[j-1]=0
     //if (E[j-1]<>0) then k[j]:=(AutoKorelasi^[i,j]-tmp)/E[j-1]
     //else k[j]:=(AutoKorelasi^[i,j]-tmp)
    end;
    a[j,j]:=k[j];
    if j>=2 then
     for l:=j-1 downto 1 do
```

```
        a[j,l]:=a[j-1,l]-k[j]*a[j-1,j-l];
    E[j]:=(1-k[j]*k[j])*E[j-1];
   end;
   for j:=1 to OrdeLPC do
    KoefisienLPC^[i,j]:=a[OrdeLPC,j];
  end;
end;
```

6. Pengubahan parameter LPC menjadi koefisien *cepstral* adalah suatu proses pengubahan parameter LPC yang sangat penting yang bisa diturunkan dari koefisien LPC adalah koefisien *cepstral* LPC, *c(m).*

Mengubah parameter LPC a_m ke koefisien cepstral c_m untuk mendapatkan kinerja yang lebih baik dan tahan terhadap *noise*, yaitu dengan persamaan:

$$c_m = a_m + \Sigma^{m-1}_{k=1}(k/m) c_k \, a_{m-k}, \; 1 \leq m \leq p \quad (9)$$

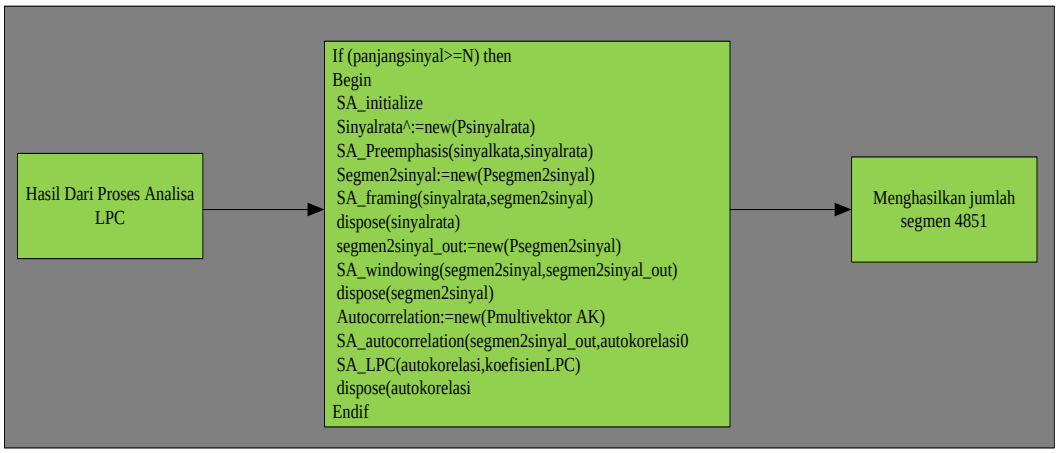
$$c_m = \Sigma^{m-1}_{k=1}(k/m) c_k \, a_{m-k}, \; m > p \quad (10)$$

Algoritma Analisis Sinyal

```
Procedure LPC:Pmultivektor;
kamus
    sinyalrata:Psinyalrata
    segmen2sinyal:Psegmen2sinyal
    segmen2sinyal_out:Psegmen2sinyal
    autocorrelation:Pmultivektor AK
Algoritma
    If (panjangsinyal>=N) then
    begin
      SA_initialize
      Sinyalrata←new(Psinyalrata)
      SA_Preemphasis(sinyalkata,sinyalrata)
      segmen2sinyal←new(Psegmen2sinyal)
      SA_framing(sinyalrata,segmen2sinyal)
      dispose(sinyalrata)
```

```
        segmen2sinyal_out←new(Psegmen2sinyal)
        SA_windowing(segmen2sinyal,segmen2sinyal_out)
        dispose(segmen2sinyal)
        Autocorrelation←new(Pmultivektor AK)
        SA_autocorrelation(segmen2sinyal_out,autokorelasi)
        dispose(segmensinyal_out)
        SA_LPC (autokorelasi,koefisienLPC)
        dispose(autokorelasi)
      endif
```

Alur masukan suara setelah dilakukan proses Analisa LPC dengan kata teknik dimulai Masukan hasil dari Analisa LPC, proses, dan hasil total segmen yang dihasilkan pada proses .



Keterangan : Untuk hasil dari total segmen lebih detailnya dapat dilihat halaman lampiran b.

Analisis Prosedur LPC :

```
Procedure SA_analisis_sinyal(sinyalkata:TSinyal;
                             var KoefisienLPC:PMultiVektor);
var
  sinyalrata:PSinyalRata;
  segmen2sinyal:Psegmen2sinyal;
  segmen2sinyal_out:Psegmen2sinyal;
```

```
  AutoKorelasi:PMultiVektorAK;
begin
  if (panjangsinyal>=N) then
  begin SA_initialize;
    sinyalrata:=new(PSinyalRata);
    SA_preemphasis(@SinyalKata, SinyalRata);
    segmen2sinyal:=new(Psegmen2sinyal);
    SA_framing(SinyalRata, Segmen2Sinyal);
    dispose(sinyalrata);
    segmen2sinyal_out:=new(Psegmen2sinyal);
    SA_windowing(Segmen2Sinyal, Segmen2Sinyal_out);
    dispose(segmen2sinyal);
    AutoKorelasi:=new(PMultiVektorAK);
    SA_autocorrelation(Segmen2Sinyal_out, AutoKorelasi);
    dispose(segmen2sinyal_out);
    //KoefisienLPC:=new(PMultiVektor); // this variable should be already
allocated
    SA_LPC(AutoKorelasi, KoefisienLPC);
    dispose(AutoKorelasi);
    //dispose(KoefisienLPC); // just remind another process which call this func for
doing this
  end;
end;
```

3.2.2.3. Analisis Pemodelan Parameter HMM

Untuk mendapatkan parameter HMM melalui lima tahap, yaitu :

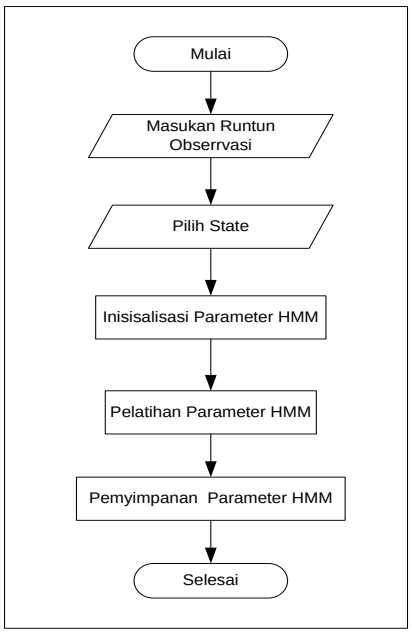
a. Memasukan Runtun *Observasi* hasil dari proses ekstraksi ciri

b. Memilih State, proses pemilihan satate berfungsi untuk pemodelan parameter HMM.

c. Inisialisasi Parameter HMM, nilai awal dari parameter-parameter model HMM diinisialisasi dengan nilai acak (random).

d. Pelatihan HMM, proses pelatihan model HMM merupakan proses reestimasi nilai parameter HMM. Proses tersebut berjalan berulang sampai selisih nilai elemen-elemen parameter HMM hasil estimasi saat ini dengan nilai elemen-elemen parameter HMM

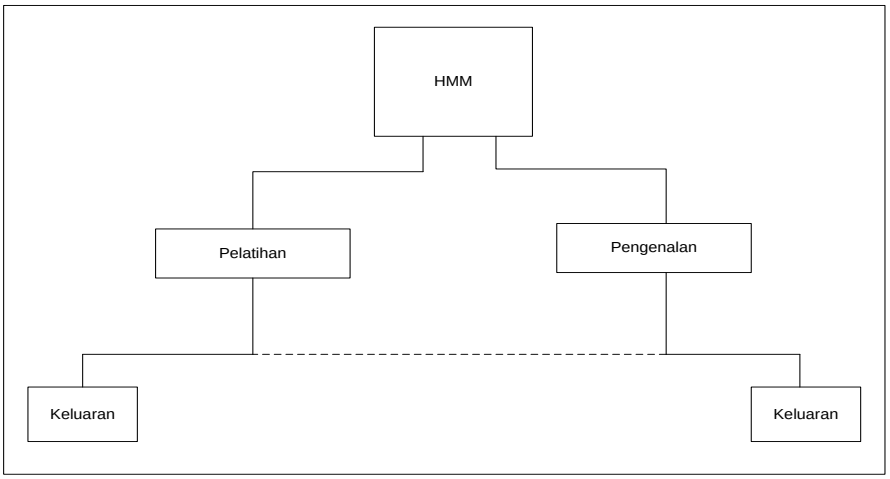
e. Penyimpanan Parameter HMM dengan tujuan untuk mendapatkan parameter yang lebih baik dan penyimpanan.



Gambar 3.6. Bagan Pemodelan Parameter HMM

A. Arsitektur Metode HMM

Arsitektur sistem digambarkan secara global, komponen-komponen yang membanguna system dan fungsi atau peranannya dalam sistem, serta hubungan anatar komponen-komponen tersebut.



Gambar 3.7 Arsitektur HMM

Penjelasan Gambar diatas akan diperjelas sebagai berikut :

1. Pelatihan berfungsi untuk pengestimasian ulang parameter HMM dari suatu data berdasarkan data latih yang diajukan pengguna, sehingga hasil *estimasi* memilki kualitas yang baik. Proses pelatihan dengan proses inisialisasi awal untuk menentukan nilai parameter-parameter dalam model HMM. Dikarenakan nilai awal tersebut masih berupa *first guess* maka harus direestimasi secara berulang untuk mendapatkan model HMM yang optimal.
2. Pengenalan berfungsi melakukan perhitungan *probabilitas* kemiripan pola dari tiap-tiap model HMM yang mempunyai *probabilitas* kemiripan yang tertinggi.

B. Proses Kerja Metode HMM

Prinsip kerja sistem pengenalan ucapan adalah dengan membandingkan informasi ucapan yang ada pada referensi dengan informasi ucapan yang menjadi masukan sistem pengenal ucapan tersebut. Blok pengenalan ucapan dengan HMM dapat dibagi menjadi tiga tahap yaitu :

a. Tahap yang pertama dilakukan pemfilteran sinyal suara dan mengubah sinyal suara analog ke digital. Dimana pemfilterannya adalah sinyal suara yang dilatih pada sistem HMM sebelumnya telah disimpan terlebih dahulu ke dalam database berbentuk bit digital. Adapun faktor yang mempengaruhi kualitas dari suara digital yaitu sample rate adalah banyaknya sample yang diambil, dan ukuran sample adalah jumlah informasi tentang sample.

b. Tahap *feture extraction* adalah untuk mendapatkan parameter-parameter yang dapat mempresentasikan sinyal suara tersebut dan dilakukan analisis serta

kuantisasi vektor. Parameter-parameter tersebut antara lain parameter *Linear Predective Coding* (LPC), parameter *pitch* (Fo) dan parameter *energy* (Eo) .

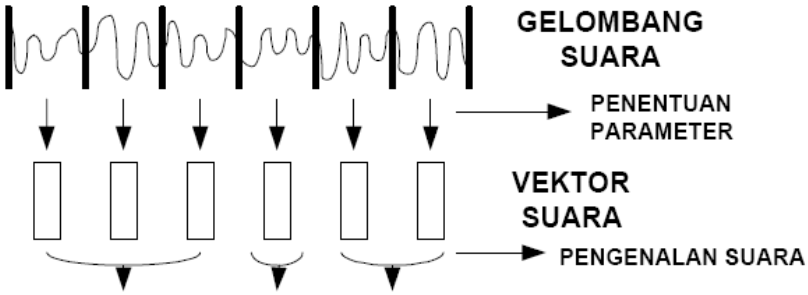
c. Tahap ketiga, dapat dibagi menjadi dua tugas yaitu tugas pemodelan dan tugas pengenalan. Untuk tugas pemodelan dibuatkan suatu model HMM dari data-data yang berupa sampel ucapan dari sebuah kata. Untuk tugas pengenalan mencari nilai probabilitas yang paling mendekati dengan menggunakan model HMM dari data-data berupa suara yang sudah melewati tahap pelatihan terlebih dahulu.

3.2.2.4. Pengidentifikasian Kata

Sebelum masuk ke pelatihan akan masuk ketahap penentuan parameter terlebih dahulu. Cara merepresentasi sinyal suara agar dapat dianalisa lebih lanjut dapatdilakukan dengan mempresentasikan sinyal suara ke dalam suatu bentuk. Suatu sinyal suara pada intinya dapat dipresentasikan ke dalam bentuk:

a. representasi sinyal suara dalam bentuk gelombang
b. representasi sinyal suara dalam bentuk parameter

Untuk representasi suara dalam bentuk gelombang dengan menampilkan pola-pola gelombang suara yang ada. Sedangkan representasi sinyal suara dalambentuk parameter merupakan cara representasi sinyal yang cukup rumit karena dari sinyal yang ada akan dihitung secara matematis parameter sinyal yangmengandung informasi sinyal. Representasi sinyal suara ke dalam parameter memberikan hasil yang lebih baik daripada representasi sinyal suara dalam bentuk gelombang. Penentuan parameter HMM secara umum dapat dilihat pada Gambar 3.8.



Gambar 3.8. Penentuan Parameter HMM Secara Umum

Diasumsikan sinyal suara merupakan pesan yang dienkoding dengan urutan satu atau lebih simbol. Urutan tersebut adalah parameter vektor suara yang diasumsikan merupakan representasi gelombang suara dengan periode tertentu.Sebuah gelombang suara dibagi-bagi dengan lama periode tertentu misalnya 10ms, jadi untuk frekuensi sampling 8000 Hz akan mempunyai 80 titik per periode10 ms. Representasi gelombang suara menjadi vektor suara menggunakan fiturLPC (*Linear Predictive Coding*).Pada penentuan parameter HMM, diasumsikan bahwa urutan vektor observasi setiap kata merupakan model Markov. Pada gambar terlihat gelombang suara direperesentasikan ke bentuk parameter vektor suara yang terdiri dari 6 observasi dengan masing-masing observasi mempunyai panjang urutan tertentu.

Misal suatu ucapan kata didefinisikan sebagai vektor-vektor suara berurutan atau pengarnatan O, yaitu:

$$O= o_1, o_2. o_T \quad (1)$$

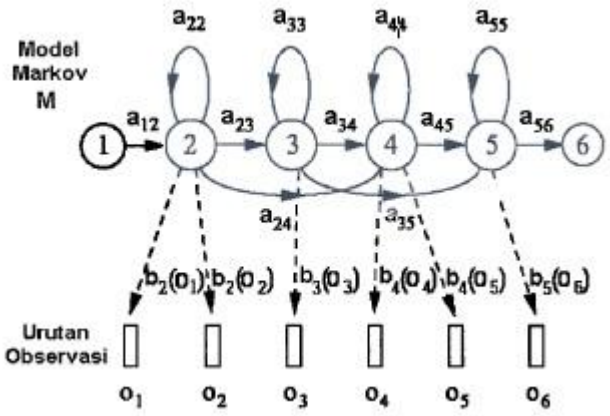
Dengan o_T adalah vektor suara yang diamati pada waktu T. Masalah pengidentifikasian kata dapat dilihat sebagai suatu perhitungan probabilitas posterior :

$$\arg\max \{P(\omega_i|O)\} \quad (2)$$

dengan ω_i adalah kosa kata ke-I. Probabilitas ini tidak dihitung langsung tetapi melalui aturan Bayes :

$$P(\omega_i|O)= P(O|\ \omega_i)P(\omega_i)/P(O) \quad (3)$$

Jadi untuk suatu probabilitas $P(\omega_i)$, pe;uang kata yang diucapkan tergantung hanya pada probabilitas keserupaan *(likelihood)* $P(\omega_i|O)$. Masalah untuk memperkirakan kondisi kelas kerapatan pengamatan $P(\omega_i|O)$ diganti menjadi masalah lebih sederhana untuk mengestimasi parameter-parameter model markov.



Gambar 3.9. Model Pembangkit Markov

Dalam pengenalan suara berbasis HMM, diumpamakan vektor-vektor suara yang diamati berhubungan dengan setiap kata yang dibangkitkan oleh model Merkov seperti gambar 3.9. disini diperlihatkan proses dimana enam model keadaan bergerak melalui suatu urutan keadaan X=1,2,3,4,5,6,7 dalam rangka membangkitkan urutan 0_1 sampai 0_6. Maka urutan keadaan X adalah :

$$P(O,X|M)=a_{12}b_2(o_1)+a_{22}b_2(o_2)+A_{33}b_3(o_3)+\ldots \quad (4)$$

Dalam Prakteknya, hanya urutan pengamatan O yang diketahui dan urutan keadaan X bersifat tersembunyi. Oleh karena itu mengapa metode ini disebut sebagai Model Markov Tersembunyi.

Jika X tidak diketahui, keserupaan yang dibutuhkan dapat dihitung dengan menjumlahkan seluruh kemungkinan urutan keadaan X=x(1),x(2),x(3),….,x(T), yaitu :

$$P(O \mid M) = \sum_{X} a_{x(0)\,x(1)} \prod_{t=1}^{T} b_{x(t)}(o_t)\, a_{x(t)\,x(t+1)} \tag{5}$$

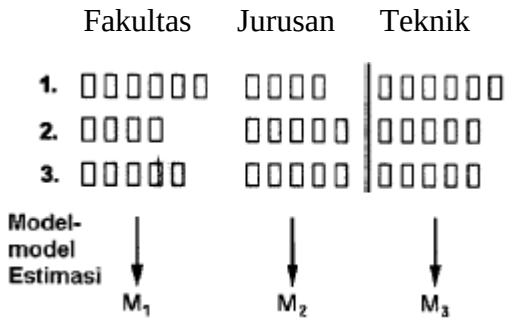
Dengan x(0) dibatas sebagai model keadaan jalan masuk sedangkanx(T+1) dibatasi sebagai model keadaan jalan keluar. Probabilitas kemiripan dapat didekati dengan hanya menggunakan urutan keadaan yang paling serupa :

$$\hat{P}(O \mid M) = \max_{x} \left\{ a_{x(0)\,x(1)} \prod_{t=1}^{T} b_{x(t)}(o_t)\, a_{x(t)\,x(t+1)} \right\} \tag{6}$$

Misal suatu himpunan model M_i berhubungan dengan kata-kata ω_i, persamaan (1) dapat diselesaikan dengan menggunakan persamaan (2) dan menganggap bahwa :

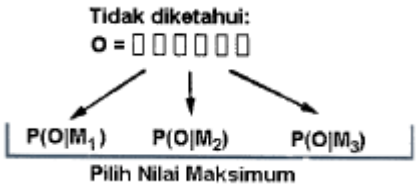
$$P(O|\, \omega_{i)} = P(O|M_i) \tag{7}$$

Dengan menganggap parameter-parameter {aij} dan{bj(ot)} diketahui untuk setiap model M_i dan dengan suatu pelatihan untuk setiap kata, maka probabilitas kemiripan untuk setiap model yang membangkitkan kata tersebut dihitung dan model yang paling mirip akan mengidentifikasi kata tersebut.



Gambar 3.10. Contoh Pelatihan

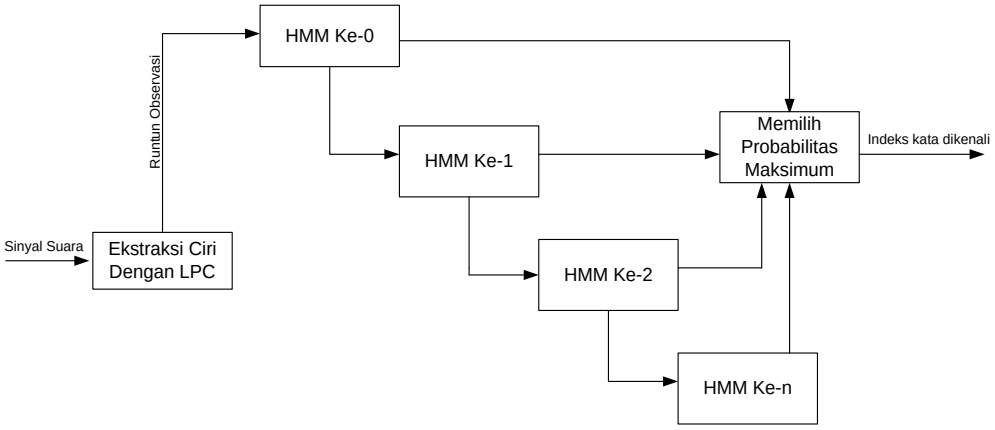
Gambar 3.10. memperlihatkan contoh pelatihan HMM menggunakan kata : “fakultas”,”jurusan”,”teknik”. Dari pelatihan ini diperoleh model-model estimasi M_1, M_2, M_3. Kemudian untuk melakukan pengenalan dari beberapa kata yang tidak dikenal, probabilitas kemiripan akan membangkitkan setiap model dari kata tersebut dan melakukan perhitungan sehingga model yang paling mirip akan mengidentifikasi kata tersebut (Gambar 3.11).



Gambar 3.11. Pengenalan menggunakan HMM

Pada model konsep pengenalan suara adalah jelas bahwa pengembangan untuk pengenalan suara pembicaraan yang kontinyu adalah dengan menghubungkan beberapa HMM secara bersama-sama secara berurutan. Setiap model dalam urutan berhubungan langsung dengan perumpamaan simbol yang mendasari yaitu fonem-fonem untuk pengenalan suara kontinyu. Data pelatihan untuk pengenalan suara kontinyu harus terdiri dari ucapan-ucapan yang kontinyu dan umumnya batas-batas yang membagi segmen-segmen suara yang

berhubungan dengan setiap model fonem dalam urutan ternyata tidak diketahui. Dalam praktek biasanya batas-batas pada sejumlah kecil data pelatihan diberi tanda secara manual dengan tangan. Seluruh segmen yang berhubungan dengan model tersebut dapat dikenal dan pengidentifikasian kata hasil pelatihan yang digambarkan di atas dapat digunakan. Biasanya jumlah data yang diperoleh dengan cara ini sangat terbatas dan model-model yang dihasilkan akan miskin dengan perkiraan-perkiraan. Walaupun menggunakan jumlah data yang lebih besar, batasbatas yang dibuat dengan tangan tidak akan optimal selama terkait dengan HMM.

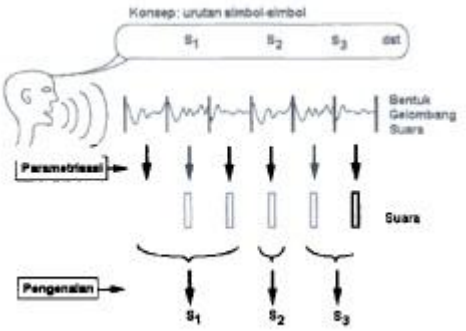


Gambar 3.12. Diagram Blok Pengidentifikasian Kata

Sinyal suara s(n) dimasukkan ke dalam ekstraksi ciri dengan menggunakan LPC sehingga didapatkan vektor runtun observasi O. Kemudian dihitung probabilitas dari runtun observasi terhadap model HMM untuk tiap-tiap. Hasil perhitungan dipilih probabilitas yang paling maksimum untuk kemudian ditetapkan sebagai kata terkenali dengan keluaran berupa teks.

3.2.3. Tahap Pengkodean

Proses pengkodean ciri-ciri vektor sinyal kedalam bentuk simbol (bilangan binary n-bit) HMM. Kumpulan kode-kode terdiri dari hasil proses ekstraksi 28 simbol yang berbeda yang digunkan untuk memetakan parameter yang ada dalam sebuah symbol. Biasanya 28 buah simbol (bilangan binary n-bit) cukup untuk mempresentasikan gelombang asli. Pengkodean suara digital dapat diklasifikasikan kedalam 2 buah kategori, yaitu bentuk gelombang *coders* dan *vocoders*. Gelombang *coders* menggunakan algoritma unutk mengkodeakan sinyal suara sehingga menghasilkan output sistem dalam bentuk yanga dapat diperkirakan. *Vocoders* mengkodekan sinyal suara dengan mengumpulkan parameter-parameter yang digitalkan dan ditransmisikan ke dalam *receiver*.



Gambar 3.13. Pengkodean Suara

Model Markov Tersembunyi (HMM) adalah suatu proses deterministik atau stokastik yang memodelkan fenomena acak yang bervariasi terhadap waktu atau struktur temporal dari urutan sinyal terhadap waktu farik. HMM akan mencocokkan estimasi probabilistik dengan model vektor-vektor berurutan keluaran dari suatu proses acak yang menjadi bagian dari rantai Markov. Rantai Markov adalah suatu struktur yang terdiri dari entitas-entitas stationer yang

disebut keadaan (state). Transisi antara atau di dalam suatu keadaan adalah suatu probabilistik. HMM mempunyai dua komponen utama: rantai markov keadaan hingga (finite state) dan himpunan hingga dari distribusi probabilitas keluaran.

Sistem pengenalan suara menganggap sinyal suara sebagai urutan satu atau lebih simbol-simbol atau segmen-segmen suara (fonem) yang dikodekan (Gambar 3.13). Dalam proses parametrisasi, suatu bentuk gelombang suara kontinyu diubah menjadi vektor-vektor parameter suara diskrit. Sedangkan proses pengenalan melakukan pemetaan antara vektor-vektor suara berurutan dan urutan simbol-simbol yang diinginkan. Ada dua masalah yang menyebabkan pengenalan seperti ini cukup sulit. Pertama, pemetaan antara simbol-simbol ke suara bukan pemetaan satu-ke-satu, karena urutan simbol-simbol berbeda dapat memberikan suara yang mirip. Selain itu ada variasi gelombang suara cukup besar disebabkan oleh misalnya gaya bicara, emosi dan kondisi lingkungan. Kedua, batas-batas antara simbol dalam bentuk gelombang tidak dapat diidentifikasi secara jelas. Permasalahan ini dikenal sebagai masalah lokasi batas-batas kata pada sinyal suara.

3.3. Kebutuhan Non Fungsional

Analisis kebutuhan sistem dilakukan untuk mencari kebutuhan apa saja yang diperlukan untuk membuat sistem yang akan dibangun. Kebutuhan sistem yang diperlukan antara lain :

3.3.1. Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan unutk implementsi system yang akan dibuat. Adapun perangkat keras dibutuhkan anatara lain :

1. Processor yang digunakan AMD Turion X2 64X2 Dual-Core
2. Memory yang digunakan 3GB 8000MHz
3. Harddisk, sebagai media storage yang digunakan 160 GB
4. Microphone
5. Speaker
6. Keyboard

3.3.2. Kebutuhan Perangkat Lunak

Pada bagian ini dijelaskan mengenai kebutuhan perangkat lunak dari palikasi sistem identifikasi kata. Adapun perangkat lunak yang dibtuhkan antara lain :

1. Sistem Operasi

Sistem Operasi yang digunakan adalah Windows 7 *(Seven)*. Dipilih karena sistem operasi ini *user friendly* dengan aplikasi apapun, walaupun belum semua aplikasi *compatible* dengan Sistem Operasi ini.

2. Database :

Database yang digunakan untuk menyimpan data-data user dan pelatihan HMM menggunakan Microsoft Office Access yang nantinya digunakan untuk aplikasi ini.

3. Borland Delphi

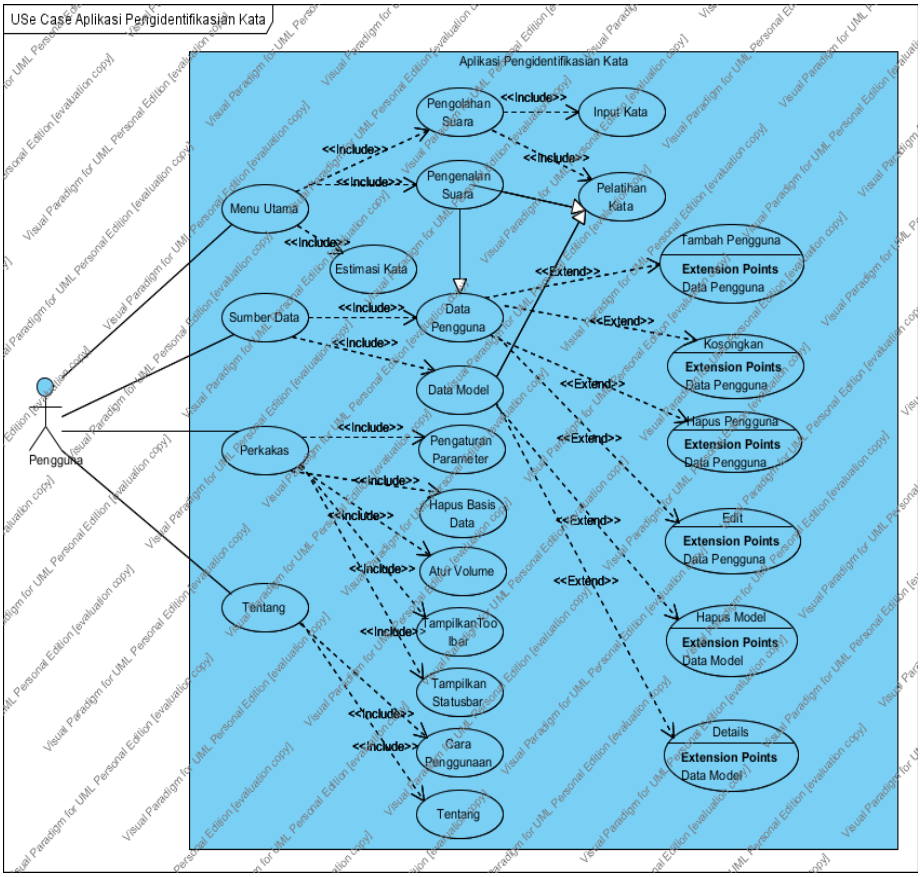
Merupakan tools development yang digunakan dalam proses coding aplikasi identifikasi kata ini.

3.4. Kebutuhanan Fungsional

Kebutuhan fungsional dari aplikasi pengidentifikasian kata diantaranya adalah :

3.4.1. Pemodelan Berbasis Object Oriented

Pemodelan yang digunakan pada aplikasi yang kami buat memakai pemodelan UML dengan tool Use Case Diagram dengan mengggunakan Visual Paradigm sebagai perangkat lunak yang digunakan.



Gambar 3.14. Use Case Diagram aplikasi identifikasi kata

1. Menu Utama

a. Skenario

Identifikasi		
Nomor	1	
Nama	Menu Utama	
Tujuan	Masuk ke Aplikasi	
Deskripsi	Menu List Untuk Masuk Ke Pengolahan suara, pengenalan suara, dan Estimasi Kata	
Tipe		
Aktor	Pengguna	
Skenario Utama		
Kondisi awal	Aplikasi belum dijalankan	
Aksi Aktor		**Reaksi Sistem**
Pengguna memilih pilihan List Menu Utama		Aplikasi mulai berjalan
Skenario Alternatif – Autentikasi Gagal		
Aksi Aktor		**Reaksi Sistem**
-		-
Kondisi akhir	Aplikasi sudah mulai dijalankan	

2. Pengolahan Suara

a. Skenario

Identifikasi		
Nomor	2	
Nama	Pengolahan Suara	
Tujuan	Masuk Ke Aplikasi	
Deskripsi	Menu List Ini Untuk Masuk Ke Menu Input Kata, Dan Pelatihan Kata	
Tipe		
Aktor	Pengguna	
Skenario Utama		
Kondisi awal	aplikasi mulai dijalankan	
Aksi Aktor		**Reaksi Sistem**
Pengguna Memilih Menu Pengolahan Suara		aplikasi mulai dijalankan
Skenario Alternatif – Autentikasi Gagal		
Aksi Aktor		**Reaksi Sistem**

-	-
Kondisi akhir	Aplikasi Sudah Mulai Dijalankan

3. Input Kata

a. Skenario

Identifikasi	
Nomor	3
Nama	Input Kata
Tujuan	Menampilkan form dan melakukan Input Kata
Deskripsi	Proses ini adalah untuk memasukan Input Kata dan juga memlih Pengguna sample
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi baru mulai dijalankan
Aksi Aktor	**Reaksi Sistem**
Telah memasukan pengguna sample dan input kata	Sistem akan Menyimpan Input Kata
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	Input Kata diterima

4. Pelatihan Kata

a. Skenario

Identifikasi	
Nomor	4
Nama	Pelatihan Kata
Tujuan	Menampilkan form dan melakukan pelatihan
Deskripsi	Proses ini adalah suatu proses pelatihan yang sebelumnya telah disetting pada proses input kata
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi mulai dijalankan
Aksi Aktor	**Reaksi Sistem**
Pengguna memilih pilihan pelatihan	Aplikasi menjalankan pelatihan
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
Pengguna Tidak memilih Nama Pengguna Sample dan Data Model	Sistem Akan memunculkan Pesan Error
Kondisi akhir	Masuk Ke form pelatihan

5. Pengenalan Suara

a. Skenario

Identifikasi		
Nomor	5	
Nama	Pengenalan Suara	
Tujuan	Melakukan Pengenalan Suara	
Deskripsi	Proses ini adalah proses pengenalan suara yang sebelumnya telah diadakan pelatihan pada tahap pelatihan setiap katanya	
Tipe		
Aktor	Pengguna	
Skenario Utama		
Kondisi awal	aplikasi sudah dijalankan	
Aksi Aktor		**Reaksi Sistem**
Masuk Menu Pengenalan Suara		Pengenalan Suara teridentifikasi dengan kata yang ditentukan
Skenario Alternatif – Autentikasi Gagal		
Aksi Aktor		**Reaksi Sistem**
-		-
Kondisi akhir	Pengenalan suara teridentifikasi dengan kata yang telah dilatih sebelumnya	

6. Estimasi Kata

a. Skenario

Identifikasi		
Nomor	6	
Nama	Estimasi Kata	
Tujuan	Melakukan Estimasi Kata	
Deskripsi	Proses ini adalah suatu proses Estimasi yang didapat dari pelatihan kata	
Tipe		
Aktor	Pengguna	
Skenario Utama		
Kondisi awal	aplikasi sudah dijalankan	
Aksi Aktor		**Reaksi Sistem**
Memilih Menu List Estimasi Kata		Menerima Pilihan Pengguna
Skenario Alternatif – Autentikasi Gagal		
Aksi Aktor		**Reaksi Sistem**
-		-
Kondisi akhir	Estimasi Kata Berhasil dilakukan	

7. Sumber Data

a. Skenario

Identifikasi		
Nomor	7	
Nama	Sumber Data	
Tujuan	Memilih Menu Sumber Data	
Deskripsi	Menu List Untuk Masuk Ke Sumber Data yang didalamnya Data Pengguna, dan Data Model	
Tipe		
Aktor	Pengguna	
Skenario Utama		
Kondisi awal	aplikasi mulai dijalankan	
Aksi Aktor		**Reaksi Sistem**
Pengguna memilih pilihan List Sumber Data		Aplikasi mulai berjalan
Skenario Alternatif – Autentikasi Gagal		
Aksi Aktor		**Reaksi Sistem**
-		-
Kondisi akhir	Pengguna memilih menu list sumber data	

8. Data Pengguna

a. Skenario

Identifikasi		
Nomor	8	
Nama	Data Pengguna	
Tujuan	Memilih Menu List Data Pengguna	
Deskripsi	Menu List Untuk Menambah Data Pengguna Berupa Tambah, Kosongkan, Hapus, dan Edit	
Tipe		
Aktor	Pengguna	
Skenario Utama		
Kondisi awal	aplikasi sudah dijalankan	
Aksi Aktor		**Reaksi Sistem**
Pengguna Memilih Menu List Data Pengguna		Aplikasi Sudah Berjalan
Skenario Alternatif – Autentikasi Gagal		
Aksi Aktor		**Reaksi Sistem**
-		-
Kondisi akhir	Pengguna Memilih Menu List Data Pengguna	

9. Tambah Pengguna

a. Skenario

Identifikasi		
Nomor	9	
Nama	Tambah Pengguna	
Tujuan	Menambahkan Nama Pengguna Sample	
Deskripsi	Menu List Untuk Menambah Data Pengguna Berupa Nama Pengguna Sample	
Tipe		
Aktor	Pengguna	
Skenario Utama		
Kondisi awal	Aplikasi sudah dijalankan	
Aksi Aktor		**Reaksi Sistem**
Pengguna memasukan nama pengguna sample		Aplikasi sistem akan menyimpan data pengguna sample
Skenario Alternatif – Autentikasi Gagal		
Aksi Aktor		**Reaksi Sistem**
-		-
Kondisi akhir	Tambah Pengguna Berhasil Dilakukan	

10. Kosongkan

a. Skenario

Identifikasi		
Nomor	10	
Nama	Kosongkan	
Tujuan	Mengkosongkan kembali Nama Pengguna	
Deskripsi	Menu List Untuk Menambah Data Pengguna Berupa Nama Pengguna Sample Untuk Mengkosongkan Nama Pengguna	
Tipe		
Aktor	Pengguna	
Skenario Utama		
Kondisi awal	aplikasi sudah dijalankan	
Aksi Aktor		**Reaksi Sistem**
Pengguna Memilih Tombol Kosongkan Untuk Mengkosongkan Nama Pengguna		Sistem Menerima Nama Pengguna Pada Data Pengguna Dikosongkan
Skenario Alternatif – Autentikasi Gagal		
Aksi Aktor		**Reaksi Sistem**
-		-
Kondisi akhir	Kosongkan Nama Pengguna Pada Data Pengguna Berhasil Dikosongkan	

11. Hapus

a. Skenario

Identifikasi	
Nomor	11
Nama	Hapus
Tujuan	Menghapus Nama Pengguna
Deskripsi	Menu List Untuk Menambah Data Pengguna Berupa Nama Pengguna Sample Untuk Menghapus Nama Pengguna
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi sudah dijalankan
Aksi Aktor	**Reaksi Sistem**
Pengguna Memilih Tombol Hapus Untuk Menghapus Nama Pengguna	Sistem Menerima Nama Pengguna Pada Data Pengguna Dihapus
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	Hapus Nama Pengguna Pada Data Pengguna Berhasil Dihapus

12. Edit

a. Skenario

Identifikasi	
Nomor	12
Nama	Edit
Tujuan	Mengedit Nama Pengguna Yang Sudah Ada
Deskripsi	Menu List Untuk Menambah Data Pengguna Berupa Nama Pengguna Sample Untuk Mengedit Nama Pengguna
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi sudah dijalankan
Aksi Aktor	**Reaksi Sistem**
Pengguna Memilih Tombol Edit Untuk Mengedit Nama Pengguna	Sistem Menerima Nama Pengguna Pada Data Pengguna Diedit
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	Edit Nama Pengguna Pada Data Pengguna Berhasil Diedit

13. Data Model

a. Skenario

Identifikasi

Nomor	13
Nama	Data Model
Tujuan	Masuk Ke Form Data Model
Deskripsi	Menu List Untuk Melihat Data Model Yang Tersimpan Mempunyai Fungsi Hapus, Dan Detail Untuk Melihat Berapa Kali Peltihan Yang Sudah Dilakukan.
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi sudah dijalankan
Aksi Aktor	**Reaksi Sistem**
Pengguna Memilih Menu List Data Model	Sistem Aplikasi Sudah Berjalan
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	Pengguna Memilih Menu List Data Model

14. Hapus

a. Skenario

Identifikasi	
Nomor	14
Nama	Hapus
Tujuan	Menghapus Data Model
Deskripsi	Menu List Pada Data Model Untuk Dapat Menghapus Data Model
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi sudah dijalankan
Aksi Aktor	**Reaksi Sistem**
Pengguna Memilih Tombol Hapus Untuk Menghapus Data Model	Sistem Menerima Data Model Dihapus
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	Hapus Data Model Berhasil Dihapus

15. Details

a. Skenario

Identifikasi

Nomor	15
Nama	Details
Tujuan	Melihat Detail Dari Pelatihan
Deskripsi	Menu List Untuk Melihat Details Data Model Berupa Data Pelatihan
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi sudah dijalankan
Aksi Aktor	**Reaksi Sistem**
Pengguna Memilih Tombol Details untuk Melihat Details	Sistem Menerima Data Model Melihat Details
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	Details Berhasil Dilihat

16. Perkakas

a. Skenario

Identifikasi	
Nomor	16
Nama	Perkakas
Tujuan	Memilih Menu Perkakas
Deskripsi	Menu List Untuk Masuk Ke Perkakas yang didalamnya Atur Volume, Pengaturan Parameter, Kosongkan Keterangan, Tampilkan toolbar, dan Tampilkan Statusbar
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi mulai dijalankan
Aksi Aktor	**Reaksi Sistem**
Pengguna Memilih Menu Perkakas	Aplikasi Sudah Berjalan
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	Pengguna Memilih Menu Perkakas

17. Pengaturan Parameter

a. Skenario

Identifikasi

Nomor	17
Nama	Pengaturan Parameter
Tujuan	Memilih Menu List Pengaturan Parameter
Deskripsi	Menu List Pengaturan Parameter Berfungsi Untuk Mengatur Berapa Jumlah Min Pelatihan
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi mulai dijalankan
Aksi Aktor	**Reaksi Sistem**
Pengguna memilih Menu List Pengaturan Paramter	Sistem Sudah Berjalan
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	Pengguna Memilih Menu List Pengaturan Paramter

18. Hapus Basis Data

a. Skenario

Identifikasi	
Nomor	18
Nama	Hapus Basis Data
Tujuan	Memilih Menu List Hapus Basis Data
Deskripsi	Menu List Untuk Menghapus Basis Data
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi mulai dijalankan
Aksi Aktor	**Reaksi Sistem**
Pengguna Memilih Menu List Hapus Basis Data	Aplikasi mulai berjalan
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	Pengguna Memilih Menu List Hapus Basis Data

19. Atur Volume

a. Skenario

Identifikasi

Nomor	19
Nama	Atur Volume
Tujuan	Memilih Menu List Atur Volume
Deskripsi	Menu List Atur Volume Untuk Mengatur Volume Dari *Microphone*
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi mulai dijalankan
Aksi Aktor	**Reaksi Sistem**
Pengguna Memilih Menu List Atur Volume	Sistem Sudah Berjalan
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	Pengguna Memilih Menu List Atur Volume dan Menyesuaikan Sesuai Kebutuhan

20. Tampilkan Toolbar

a. Skenario

Identifikasi	
Nomor	20
Nama	Tampilkan Toolbar
Tujuan	Memilih Menu List Tampilkan Toolbar
Deskripsi	Menu List Untuk Menampilkan dan Menyembunyikan Toolbar Pada fungsi-fungsi Utama
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi mulai dijalankan
Aksi Aktor	**Reaksi Sistem**
Pengguna Memilih Menu List Tampilkan Toolbar	Aplikasi mulai berjalan
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	Pengguna Memilih Menu List Tampilkan Toolbar

21. Tampilkan Statusbar

a. Skenario

Identifikasi

Nomor	21
Nama	Tampilkan Statusbar
Tujuan	Memilih Menu List Tampilkan Statusbar
Deskripsi	Menu List Untuk Menampilkan dan Menyembunyikan Statusbar
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi mulai dijalankan
Aksi Aktor	**Reaksi Sistem**
Pengguna Memilih Menu List Tampilkan Statusbar	Aplikasi mulai berjalan
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	Pengguna Memilih Menu List Tampilkan Statusbar

22. Bantuan

a. Skenario

Identifikasi	
Nomor	22
Nama	Bantuan
Tujuan	Memilih Menu Bantuan
Deskripsi	Menu List Untuk Masuk Ke Bantuan yang didalamnya Cara Penggunaan, dan Tentang
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi mulai dijalankan
Aksi Aktor	**Reaksi Sistem**
Pengguna Memilih Menu Bantuan	Aplikasi Sudah Berjalan
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	Pengguna Memilih Menu Bantuan

23. Cara Penggunaan

Skenario

Identifikasi	
Nomor	23

Nama	Cara Penggunaan
Tujuan	Memilih Menu Perkakas
Deskripsi	Menu List Untuk Mengetahui Bagaimana Cara Penggunaan Dari Aplikasi Ini
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi mulai dijalankan
Aksi Aktor	**Reaksi Sistem**
Pengguna Memilih Menu List Cara Penggunaan	Aplikasi Sudah Berjalan
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	Pengguna Memilih Menu List Cara Pengunaan

24. Tentang

Skenario

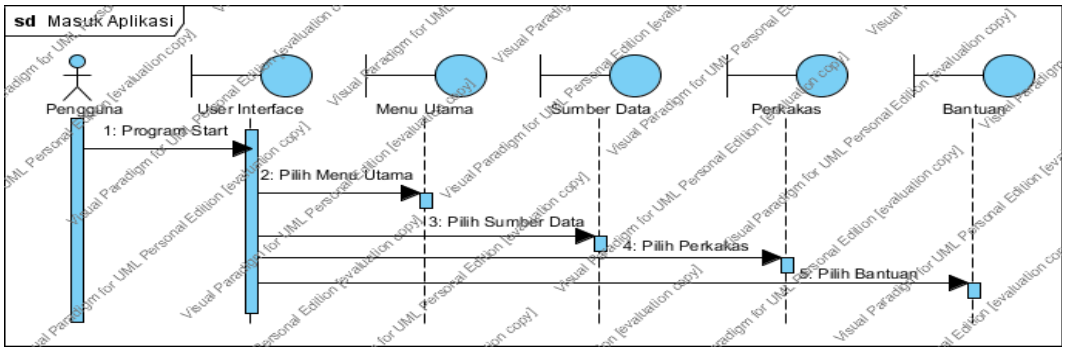
Identifikasi	
Nomor	24
Nama	Tentang
Tujuan	Memilih Menu List Tentang
Deskripsi	Menu List Untuk Mengetahui Informsai Yang Ada Dalam Tentang
Tipe	
Aktor	Pengguna
Skenario Utama	
Kondisi awal	aplikasi mulai dijalankan
Aksi Aktor	**Reaksi Sistem**
Pengguna Memilih Menu List Tentang	Aplikasi Sudah Berjalan
Skenario Alternatif – Autentikasi Gagal	
Aksi Aktor	**Reaksi Sistem**
-	-
Kondisi akhir	Pengguna Memilih Menu List Tentang

3.4.2. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan disekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu.

1. Sequence Diagram Masuk Aplikasi

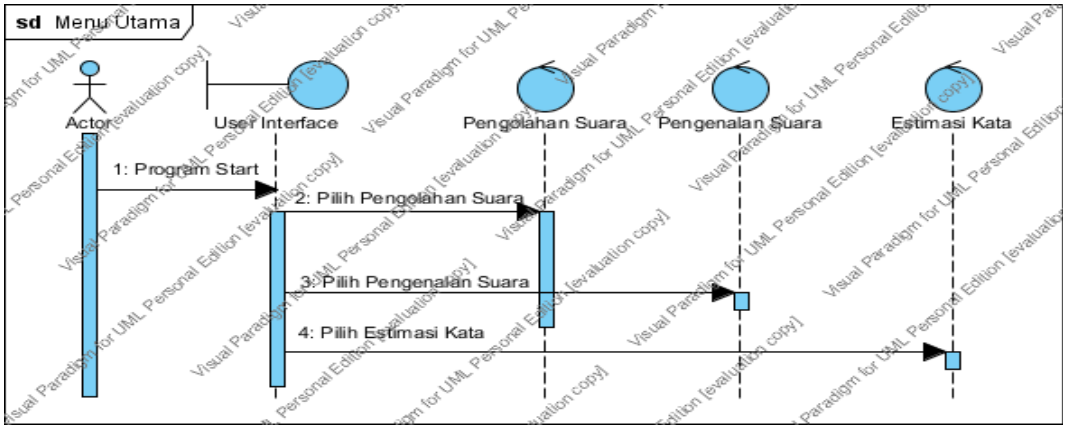
Sequence diagram masuk aplikasi ini terdiri dari menu utama, sumber data, perkakas dan bantuan.



Gambar 3.15. Sequence Diagram Masuk Aplikasi

2. Sequence Diagram Menu Utama

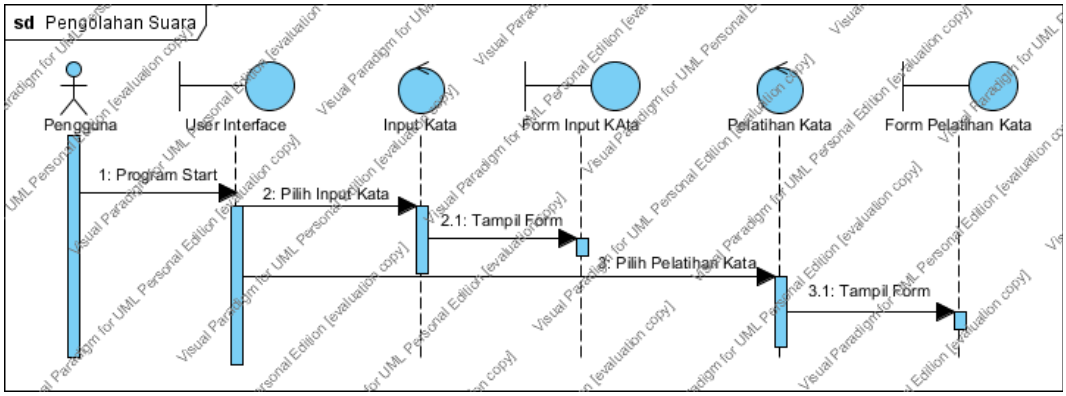
Sequence diagram menu utama ini terdiri dari proses pengolahan suara, proses pengenalan suara, dan proses estimasi kata.



Gambar 3.16. Sequence Diagram Menu Utama

3. Sequence Diagram Pengolahan Suara

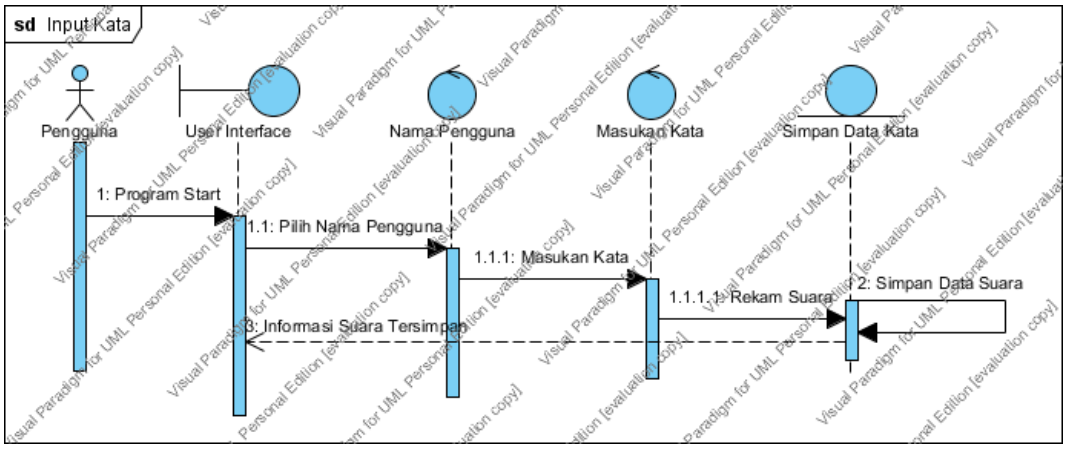
Sequence diagram pengolahan suara terdiri dari proses input kata, tampilan form input kata, proses pelatihan kata, dan tampilan form pelatihan kata.



Gambar 3.17. Sequence Diagram Pengolahan Suara

4. Sequence Diagram Input Kata

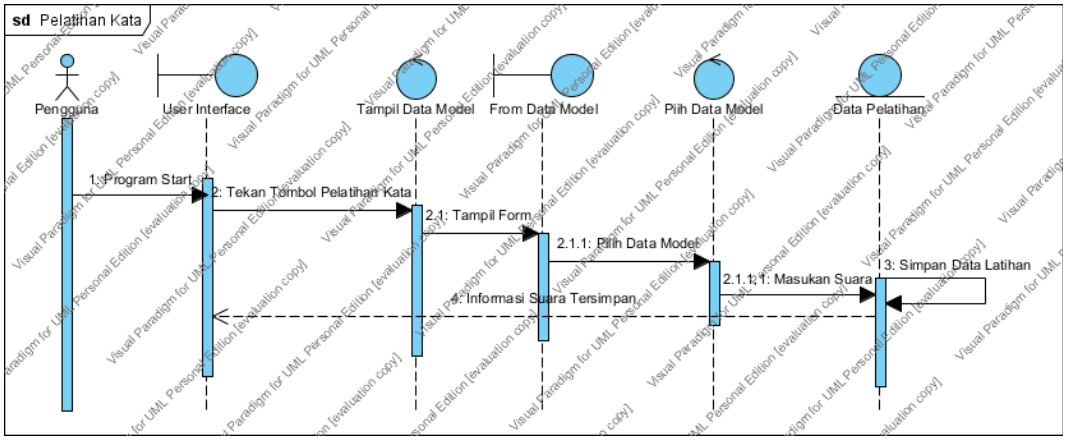
Sequence diagram input kata terdiri dari proses memilih nama pengguna, proses masukan kata, dan simpan data kata.



Gambar 3.18. Sequence Diagram Input Kata

5. Sequence Diagram Pelatihan Kata

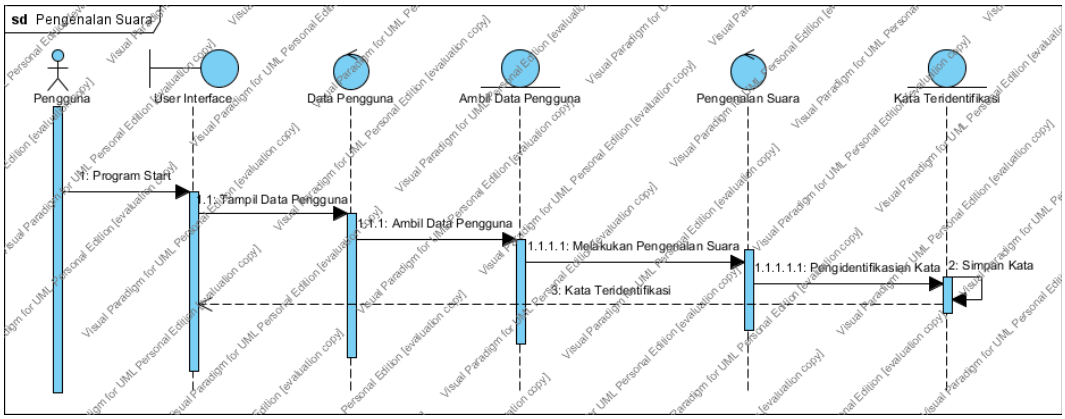
Sequence diagram peltihan kata terdiri dari proses tampil data model, form data model, pilih data model, dan simpan data pelatihan.



Gambar 3.19 Sequence Diagram Pelatihan Kata

6. Sequence Diagram Pengenalan Suara

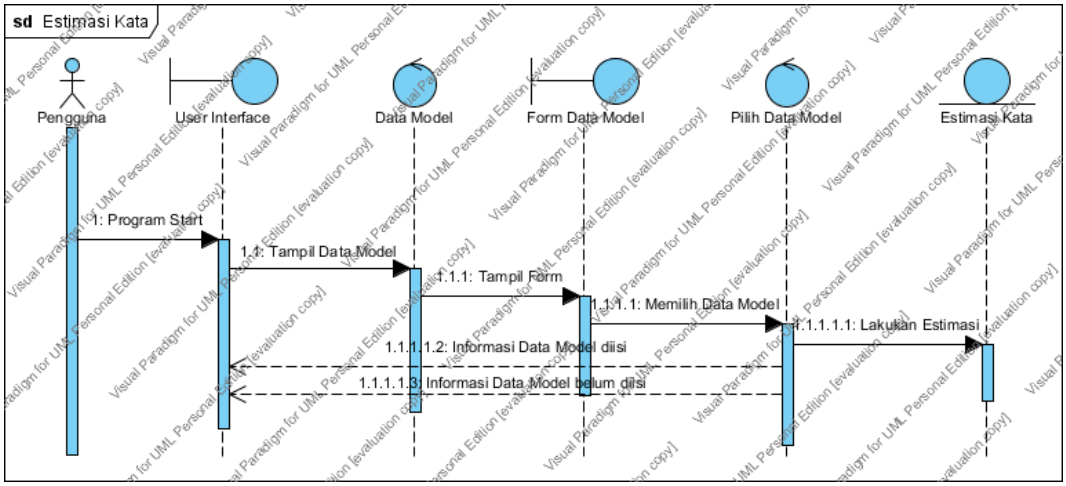
Sequence diagram pengenalan suara terdiri dari tampil data pengguna, ambil data pengguna, proses pengenalan suara, simpan kata teridentifikasi.



Gambar 3.20. Sequence Diagram Pengenalan Suara

7. Sequence Diagram Estimasi Kata

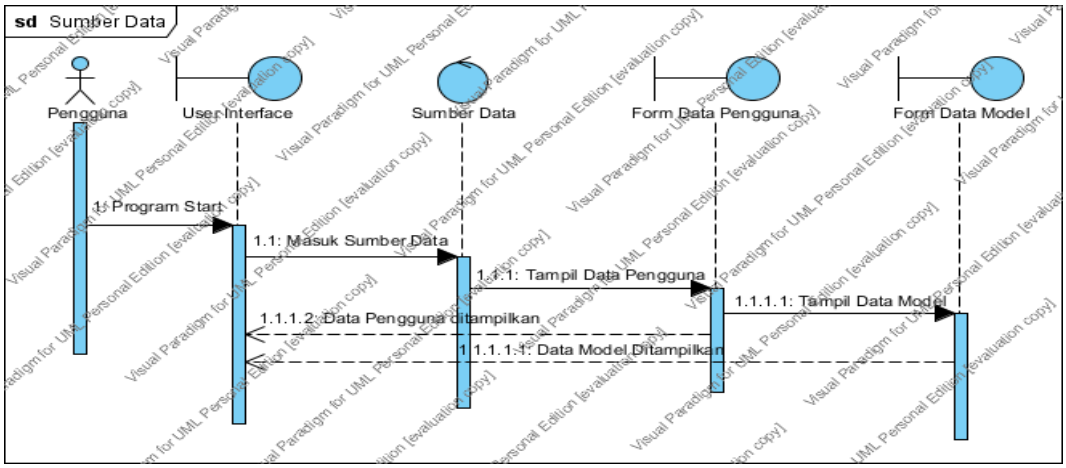
Sequence diagram estimasi kata terdiri dari tampil data model, tampil form data model, memilih data model, dan melakukan estimasi kata.



Gambar 3.21. Sequence Diagram Estimasi Kata

8. Sequence Diagram Sumber Data

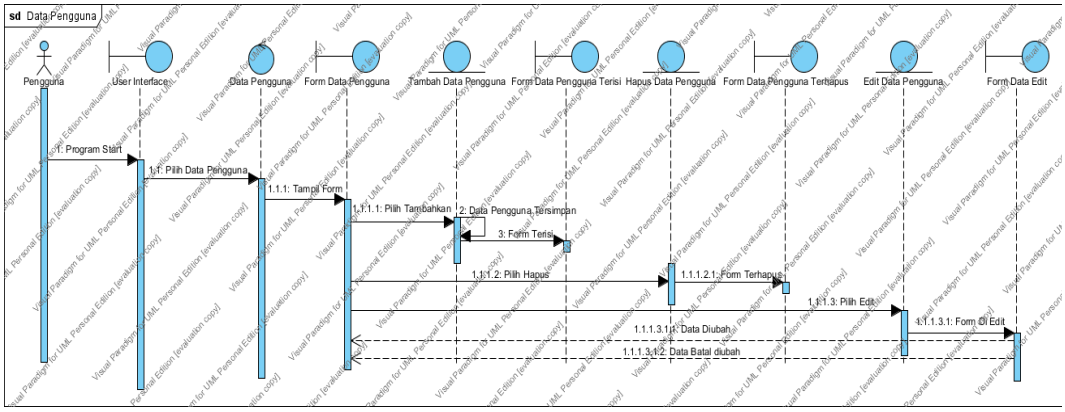
Sequence diagram sumber data terdiri dari masuk sumber data, tampil data pengguna, dan tampil data model.



Gambar 3.22. Sequence Diagram Sumber Data

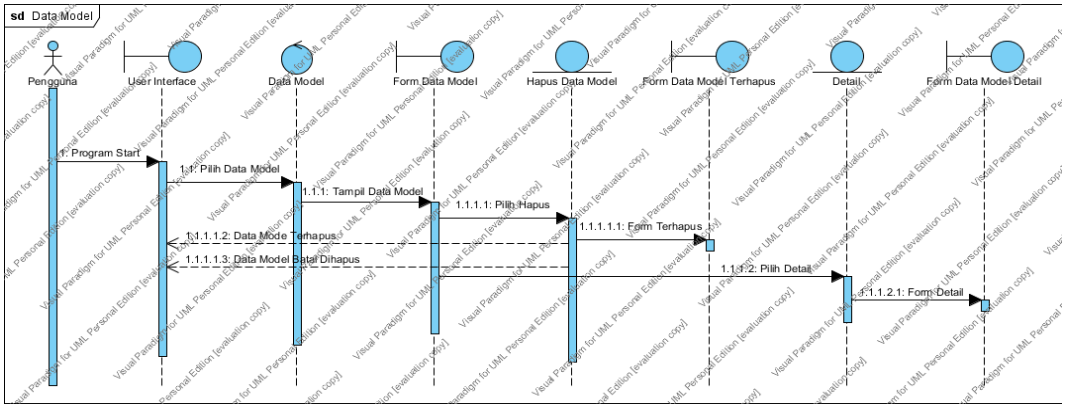
9. Sequence Diagram Data Pengguna

Sequence diagram data pengguna terdiri dari pilih data pengguna, tampila form data pengguna, pilih tambah data pengguna, form data pengguna terisi, hapus data pengguna, form data pengguna terhapus, edit data pengguna, dan form data edit.



Gambar 3.23. Sequence Diagram Data Pengguna

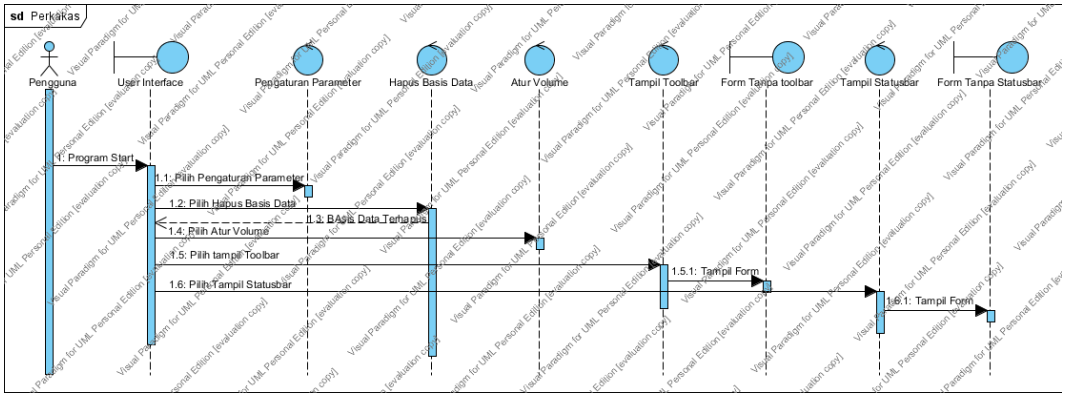
10. Sequence Diagram Data Model



Gambar 3.24. Sequence Diagram Data Model

11. Sequence Diagram Perkakas

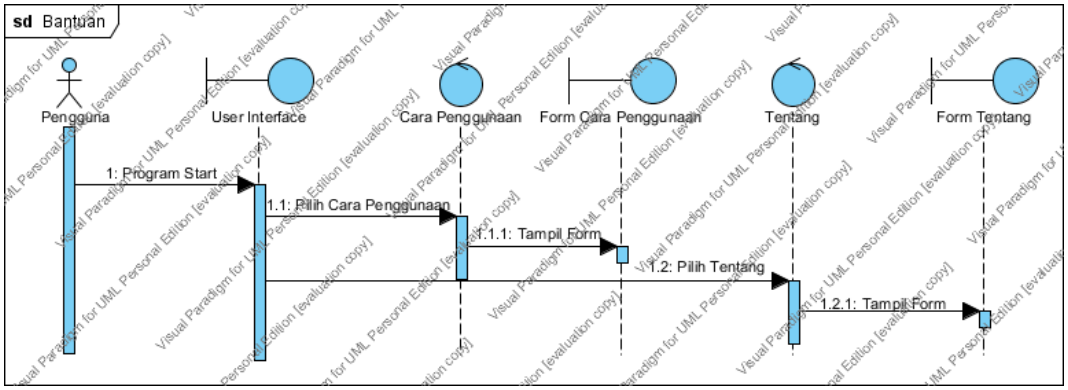
Sequence diagram perkakas terdiri dari pengaturan parameter, hapus basis data, atur volume, tampil toolbar, form tampil toolbar, tampil statusbar, dan form tampil statusbar



Gambar 3.25. Sequence Diagram Perkakas

12. Sequence Diagram Bantuan

Sequence diagram batuan terdiri dari pilih cara penggunaan, tampil form cara penggunaan, pilih tentang, dan tampil form tentang.

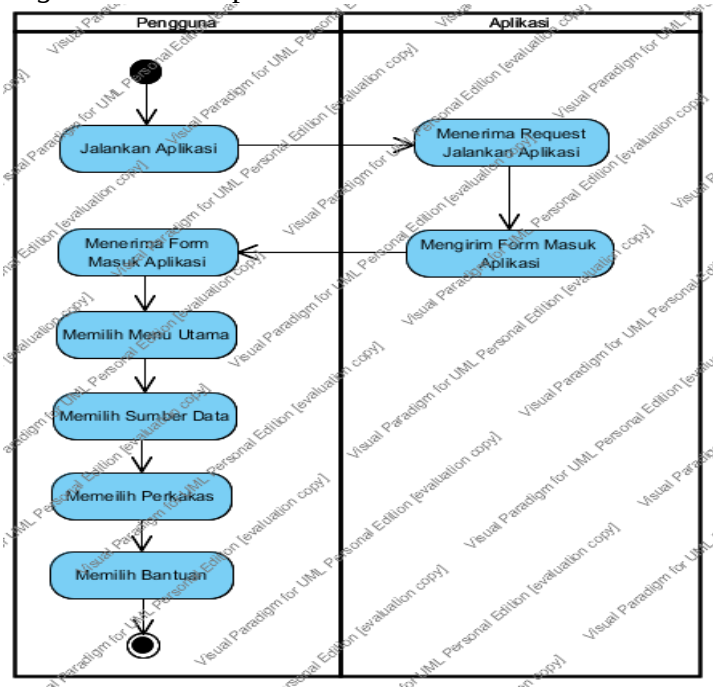


Gambar 3.26. Sequence Diagram Bantuan

3.4.3. *Activity Diagram*

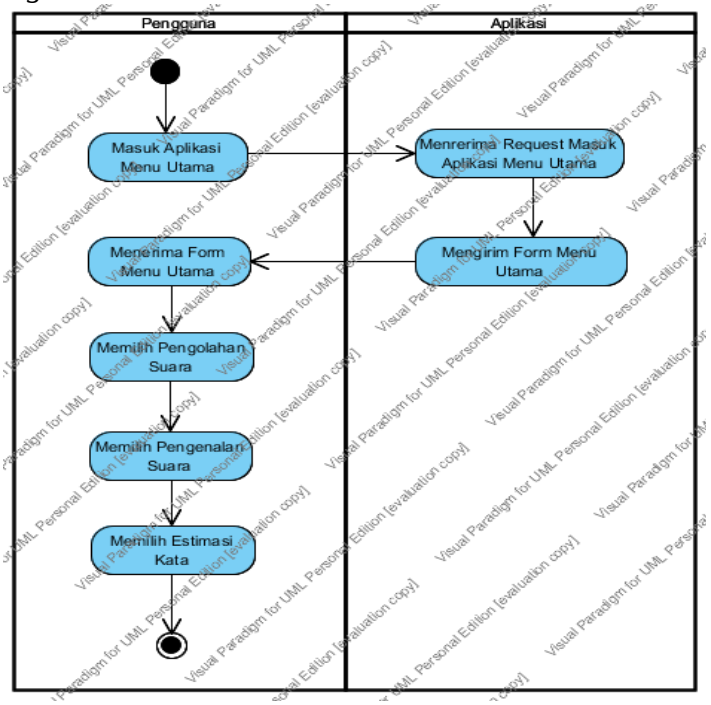
Activity diagram atau diagram aktifitas digunakan untuk menggambarkan aliran kejadian dalam *use case* sistem dengan tujuan untuk memudahkan mengkomunikasikan langkah-langkah dalam aliran kejadian. *Activity diagram* tidak perlu dibuat untuk setiap aliran kejadian, tetapi akan sangat berguna untuk aliran kejadian yang komplek dan melebar.

1. *Activity Diagram* Masuk Aplikasi



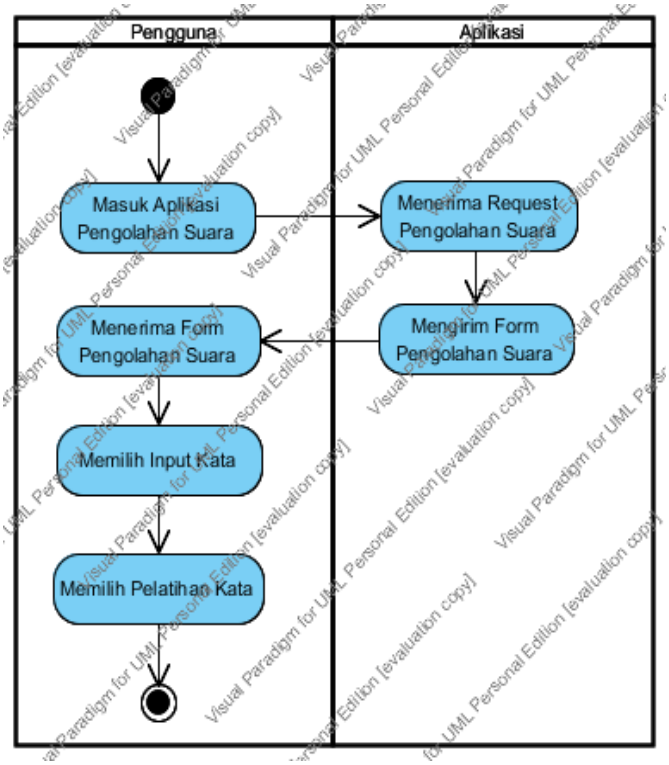
Gambar 3.27. Activity Diagram Masuk Aplikasi

2. *Activity Diagram* Menu Utama



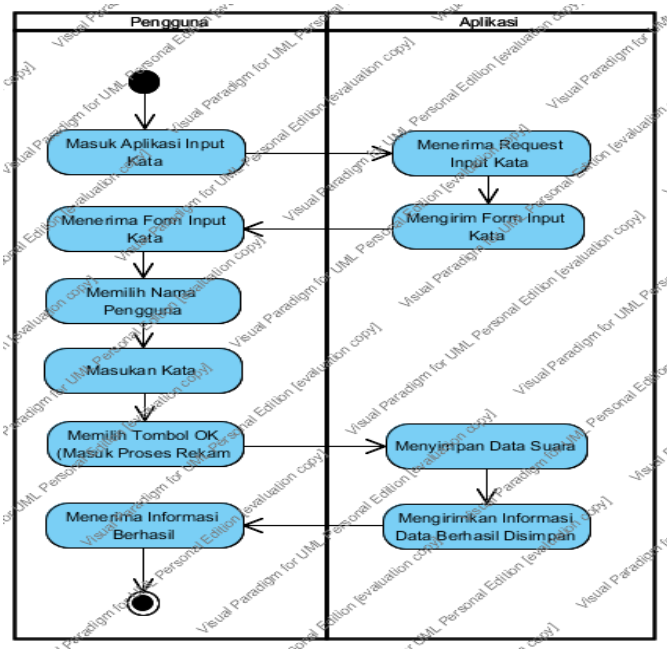
Gambar 3.28. Activity Diagram Menu Utama

3. *Activity Diagram* Pengolahan Suara



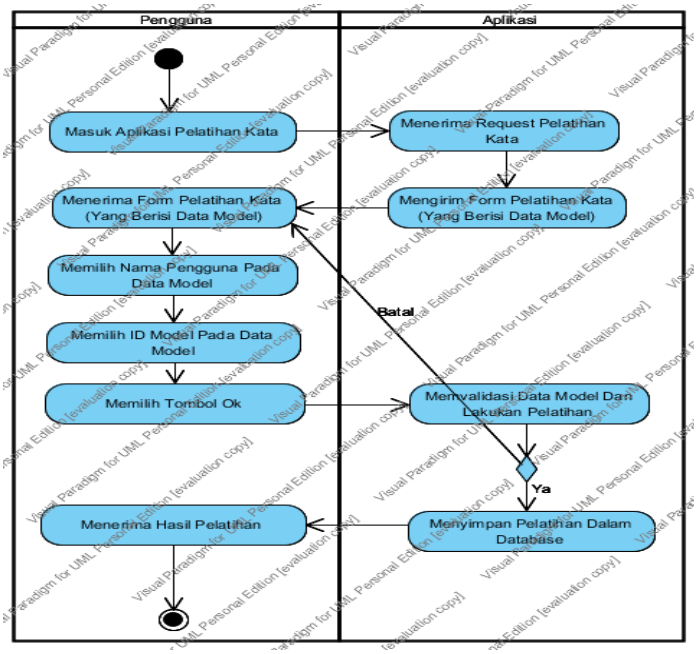
Gambar 3.29. Activity Diagram Pengolahan Suara

4. *Activity Diagram* Input Kata



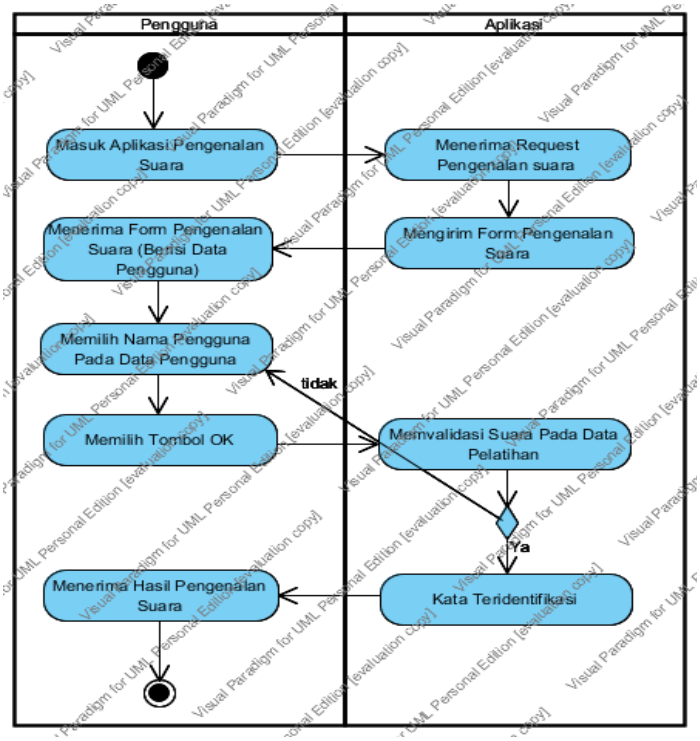
Gambar 3.30. Activity Diagram Input Kata

5. *Activity Diagram* Pelatihan Kata



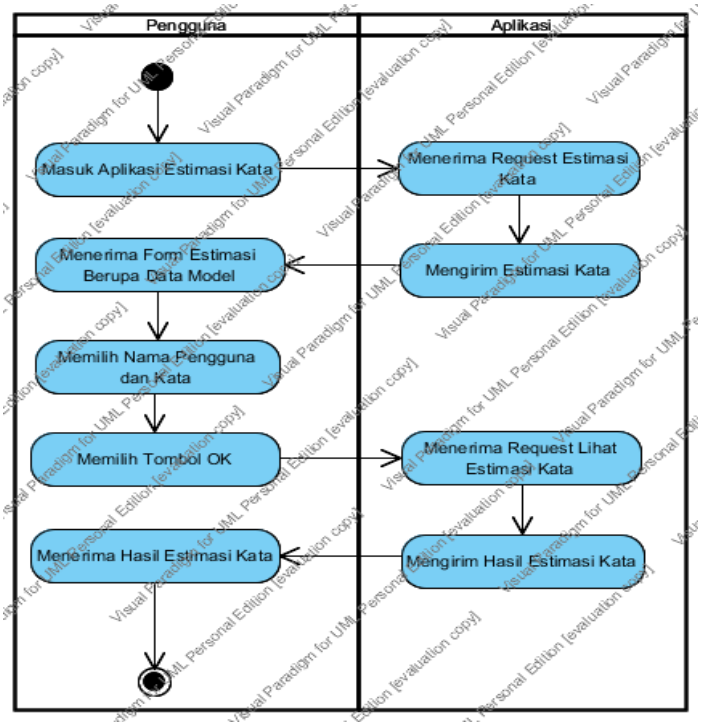
Gambar 3.31. Activity Diagram Pelatihan Kata

6. *Activity Diagram* Pengenalan Suara



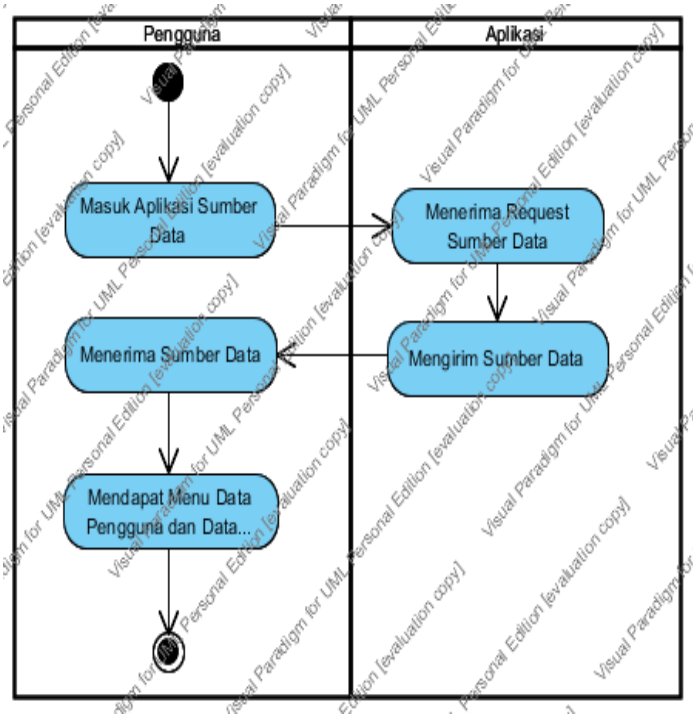
Gambar 3.32. Activity Diagram Pengenalan Suara

7. *Activity Diagram* Estimasi Kata



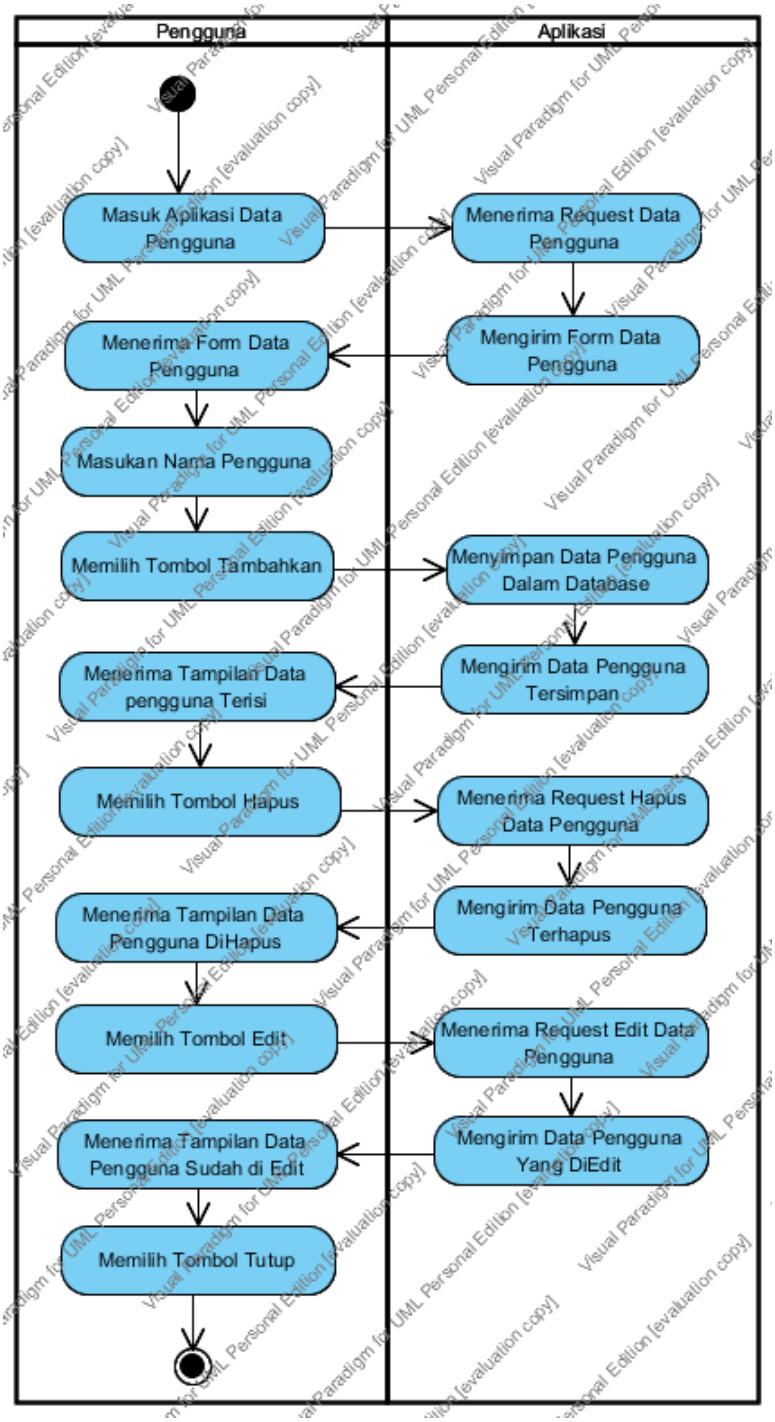
Gambar 3.33 Activity Diagram Estimasi Kata

8. *Activity Diagram* Sumber Data



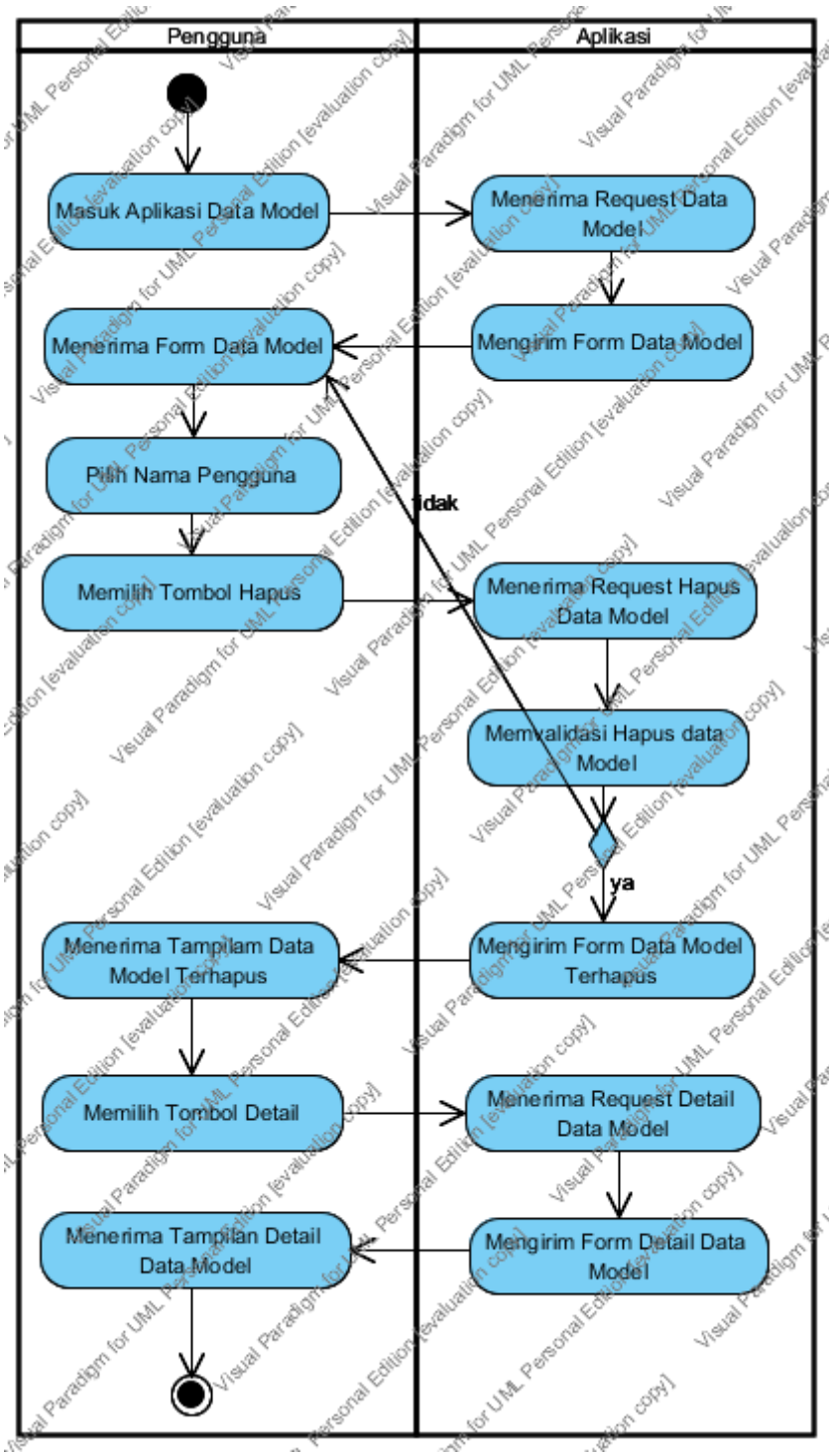
Gambar 3.34. Activity Diagram Sumber Data

9. *Activity Diagram* Data Pengguna



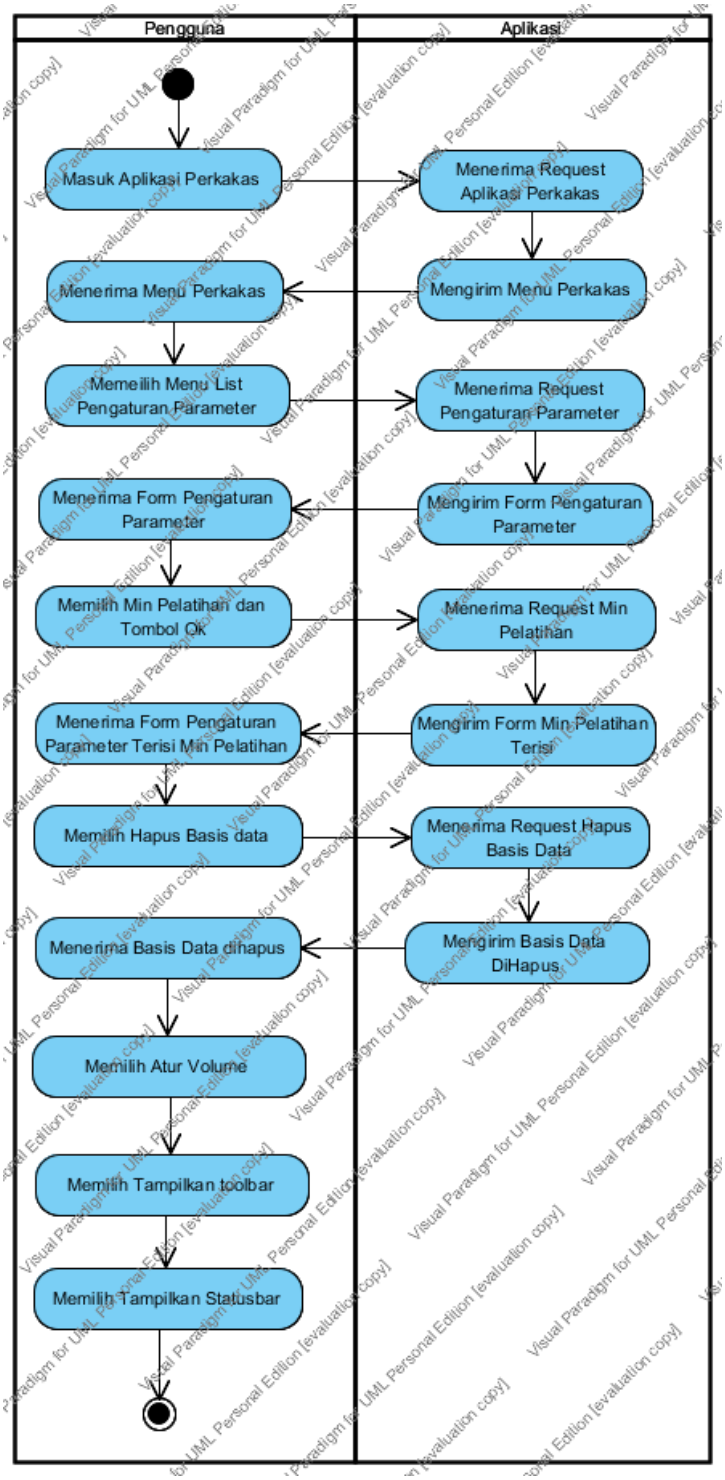
Gambar 3.35. Activity Diagram Data Pengguna

10. *Activity Diagram* Data Model



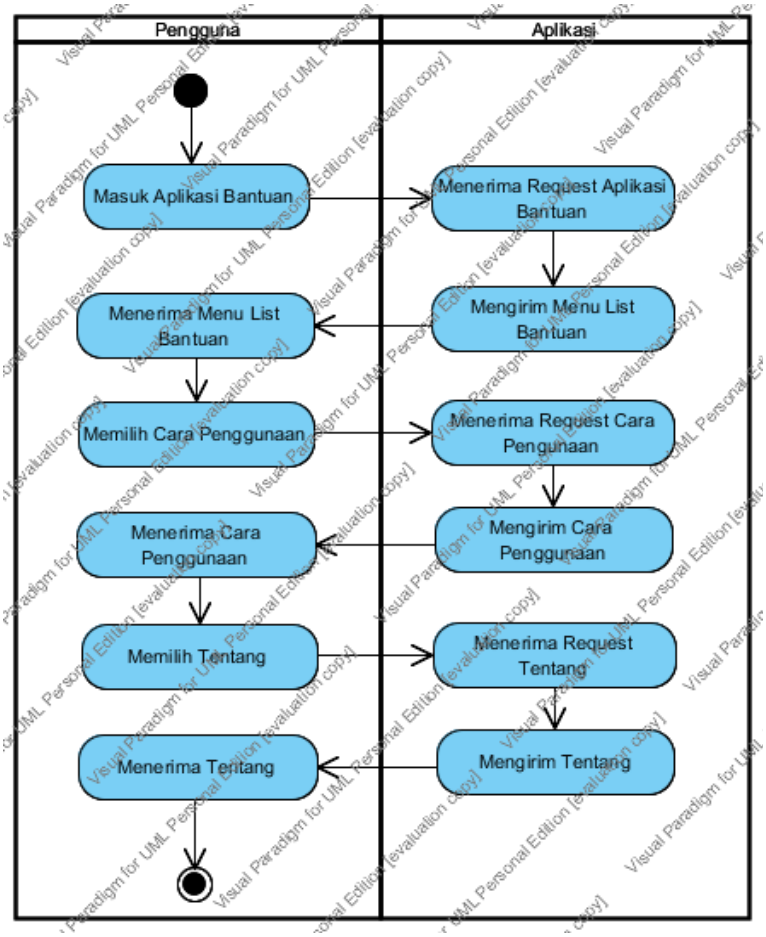
Gambar 3.36. Activity Diagram Data Model

11. *Activity Diagram* Perkakas



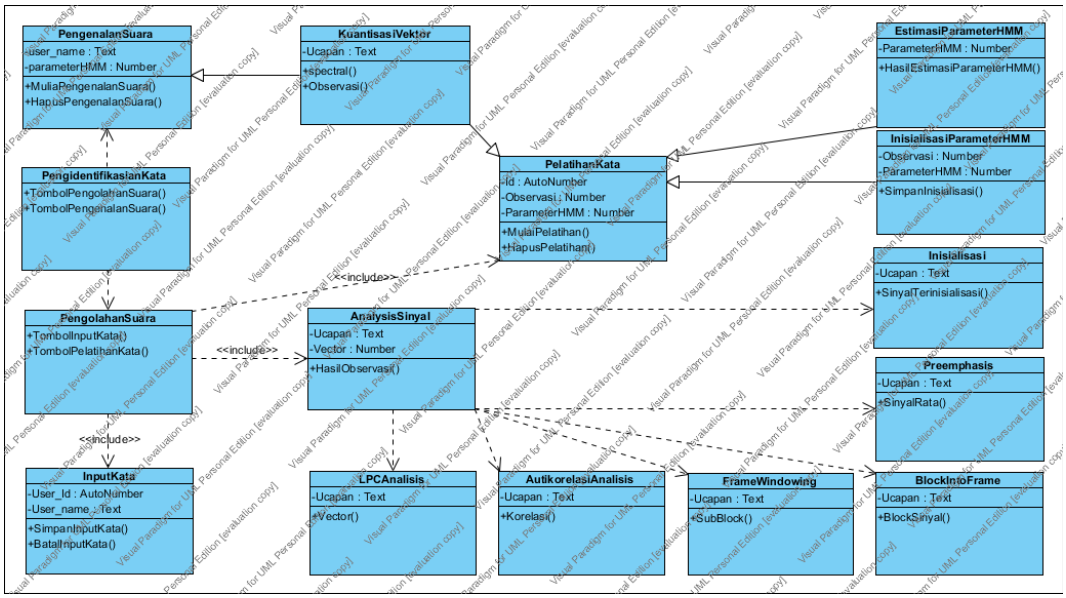
Gambar 3.37. Activity Diagram Perkakas

12. *Activity Diagram* Bantuan



Gambar 3.38. Activity Diagram Bantuan

3.4.4. *Class Diagram*

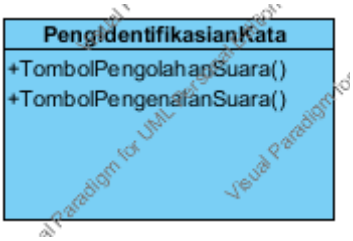


Gambar 3.39. Class Diagram Pengidentifikasian Kata

3.4.4.1.Spesifikasi Kelas

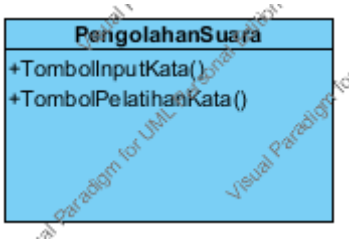
A. Kelas PengidentifikasianKata

Kelas PengidentifikasianKata adalah kelas yang digunkan untuk masuk kedalam kelas PengolahanSuara dan PengenalanSuara.



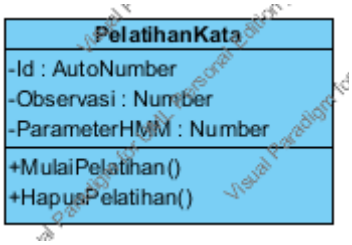
B. Kelas PengolahanSuara

Kelas PengolahanSuara adalah kelas yang digunakan untuk masuk InputKatadan Pelatihan.



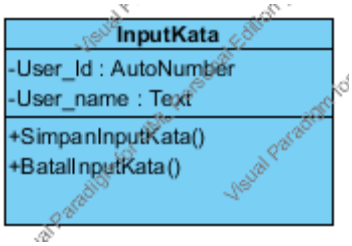
C. Kelas Pelatihan

Kelas Pelatihan adalah kelas yang digunakan untuk masuk MulaiPelatihan dan HapusPelatihan.



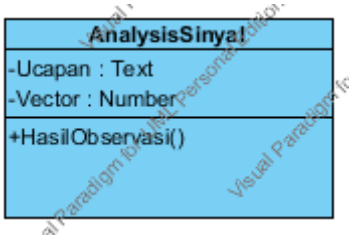
D. Kelas InputKata

Kelas InputKata adalah kelas yang digunakan untuk Simpan InputKata dan Batal InputKata.



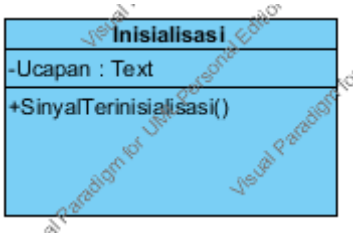
E. Kelas AnalysisSinyal

Kelas AnalysisSinyal adalah kelas yang digunakan untuk mengetahui HasilObservasi.



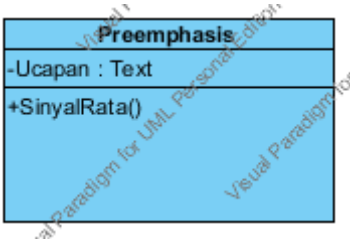
F. Kelas Inisialisasi

Kelas Inisialisasi adalah kelas yang digunakan untuk mengetahui ucapan sudah menjadi sinyal yang terinisialisasi.



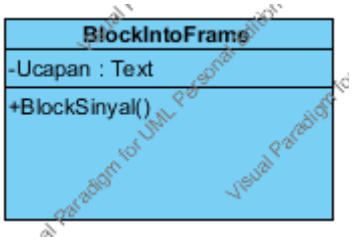
G. Kelas Preemphasis

Kelas Preemphasis adalah kelas yang digunakan untuk mengetahui ucapan sudah menjadi sinyal rata.



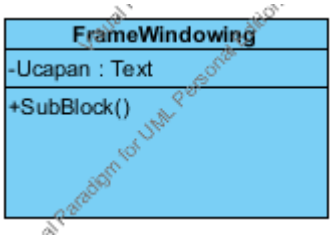
H. Kelas BlockingIntoFrame

Kelas BlockingIntoFrame adalah kelas yang digunakan untuk mengetahui ucapan sudah menjadi block sinyal.



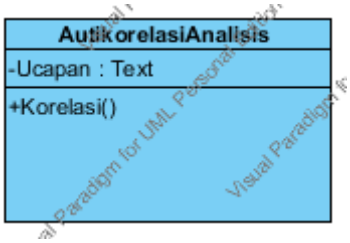
I. Kelas FrameWindowing

Kelas FrameWindowing adalah kelas yang digunakan untuk mengetahui ucapan sudah menjadi sub block.



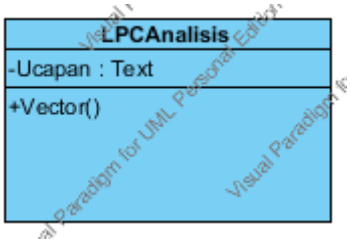
J. Kelas AutoKorelasiAnalysis

Kelas AutoKorelasiAnalysis adalah kelas yang digunakan untuk mengetahui ucapan sudah menjadi korelasi.



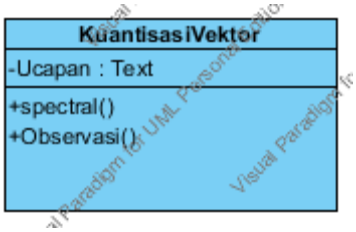
K. Kelas LPCAnalysis

Kelas LPCAnalysis adalah kelas yang digunakan untuk mengetahui ucapan sudah menjadi vektor.



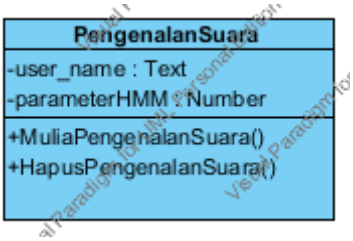
L. Kelas KuantisasiVektor

Kelas KuantisasiVektor adalah kelas yang digunakan untuk mengetahui ucapan sudah menjadi sebuah spectral dan observasi yang akan dipaki pada kelas pelatihan dan pengenalan.



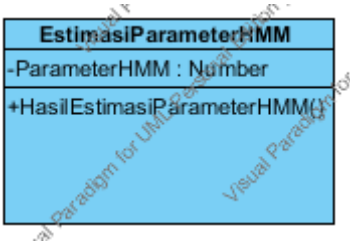
M. Kelas PengenalanSuara

Kelas PengenalanSuara adalah kelas yang digunakan untuk mulai PengenalanSuara dan Hapus Pengenalansuara.



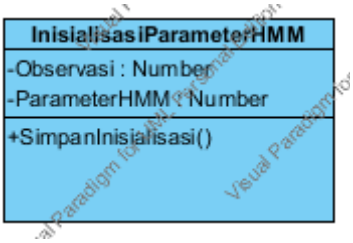
N. Kelas EstimasiParameterHMM

Kelas EstimasiParameterHMM adalah kelas yang digunakan untuk menggeneralisasi kelas PelatihanKata.



O. Kelas InisialisasiParemeterHMM

Kelas InisialisasiParameterHMM adalah kelas yang digunakan untuk menggeneralisasi kelas PelatihanKata.



3.4.4.2. Deskripsi Atribut

A. Deskripsi Atribut PengidentifikasianKata

Nama	Tipe	Panjang	Range	Nilai Tetap
-	--	-	-	-

Deskripsi Layanan :

13. Method TombolPengolahaSuara() digunakan untuk masuk ke proses pengolahan suara.

14. Method TombolPengenalasSuara() digunkan untuk masuk ke proses pengenalan suara.

B. Deskripsi Atribut PengolahanSuara

Nama	Tipe	Panjang	Range	Nilai Tetap
-	--	-	-	-

1. Method TombolInputKata() digunakan untuk masuk ke proses input kata.
2. Method TombolPelatihan() digunakan untuk masuk ke proses pelatihan.

C. Deskripsi Atribut Pelatihan

Nama	Tipe	Panjang	Range	Nilai Tetap
Id	Number	Auto Increament	‘‘0’..’9’	-
Observasi	Number	Auto Increament	‘0’..’9’	
ParameterHMM	Number	Auto Increament	‘0’..’9’	

1. Method MulaiPelatihan() digunakan untuk masuk ke proses Pelatihan
2. Method HapusPelatihan() digunakan untuk menghentikan proses Pelatihan.

D. Deskripsi Atribut InputKata

Nama	Tipe	Panjang	Range	Nilai Tetap
User_Id	AutoNumber	Auto Increament	‘0’..’9’	-
User_Name	Text	50	‘A’..’Z’	

1. Method SimpanInputKata() digunakan untuk menyimpan data pada imput kata..
2. Method BatalInputKata() digunakan untuk tidak menyimpan data pada input kata.

E. Deskripsi Atribut AnalysisSinyal

Nama	Tipe	Panjang	Range	Nilai Tetap
Ucapan	Text	50	‘A’..’Z’	-
Vektor	Number		‘0’..’9’	

1. Method HasilObservasi() digunakan untuk masuk ke proses peltihan dan pengenalan suara.

F. Deskripsi Atribut Insialisasi

Nama	Tipe	Panjang	Range	Nilai Tetap
Ucapan	Text	50	‘A’..’Z’	-

1. Method SinyalTerinisialisasi() digunakan untuk menuju proses preemphasis

G. Deskripsi Atribut Preemphasis

Nama	Tipe	Panjang	Range	Nilai Tetap
Ucapan	Text	50	‘A’..’Z’	-

1. Method SinyalRata() digunakan untuk menuju proses BlockingIntoFrame.

H. Deskripsi Atribut BlockingIntoFrame

Nama	Tipe	Panjang	Range	Nilai Tetap
Ucapan	Text	50	‘A’..’Z’	-

1. Method BlockSinyal() digunakan untuk menuju proses WindowingFrame

I. Deskripsi Atribut WindowingFrame

Nama	Tipe	Panjang	Range	Nilai Tetap
Ucapan	Text	50	‘A’..’Z’	-

1. Method SubBlock() digunakan untuk menuju proses AutoKorelasiAnalysis

J. Deskripsi Atribut AutoKorelasiAnalysis

Nama	Tipe	Panjang	Range	Nilai Tetap
Ucapan	Text	50	‘A’..’Z’	-

1. Method Korelasi() digunakan untuk menuju proses LPCAnalysis

K. Deskripsi Atribut LPCAnalysis

Nama	Tipe	Panjang	Range	Nilai Tetap
Ucapan	Text	50	‘A’..’Z’	-

1. Method Vektor() digunakan untuk menuju proses KuantisasiVektor

L. Deskripsi Atribut KuantisasiVektor

Nama	Tipe	Panjang	Range	Nilai Tetap
Ucapan	Text	50	‘A’..’Z’	-

1. Method Spectral() digunakan untuk menyimpan semua data yang akan disimpan dalm codebook.

2. Method Observasi() digunakan untuk menuju proses pelatihan dan pengenalan.

M. Deskripsi Atribut PengenalanSuara

Nama	Tipe	Panjang	Range	Nilai Tetap
User_Name	Text	50	‘A’..’Z’	-
ParameterHMM	Number	-	‘0’..’9’	

1. Method MulaiPengenalanSuara() digunakan untuk masuk ke proses PengenalanSuara.

2. Method HapusPengenalanSuara () digunakan untuk menghapus proses PengenalanSuara.

N. Deskripsi Atribut EstimasiParameterHMM

Nama	Tipe	Panjang	Range	Nilai Tetap
ParameterHMM	Number	50	‘0’..’9’	-

1. Method HasilEstimasiParameterHMM() digunakan untuk mengetahui hasil estimasi parameter.

O. Deskripsi Atribut InisialisasiParameterHMM

Nama	Tipe	Panjang	Range	Nilai Tetap
Observasi	Number	50	‘A’..’Z’	-
ParameterHMM	Number	-	‘0’..’9’	

1. Method SimpanInisialisasi() digunakan untuk menyimpan proses inisialisasi pada pelatihan.

3.5. Perancangan Sistem

Perancangan merupakan tahap lanjutan dari analisis sistem dimana pada perancangan sistem digambarkan rancangan sistem yang akan dibangun sebelum dilakuka pengkodean kedalam suatu bahasa pemrograman. Dalam perancangan system tidak lepas dari hasil analisis, karena dari hasil analisi sistem baru dapat dibuat suatu perancangan sistem.

Desain umum yang akan diaplikasikan bertujuan untuk memberikan gambaran secara umum kepada penggunaan tentang sistem yang akan dibangun.

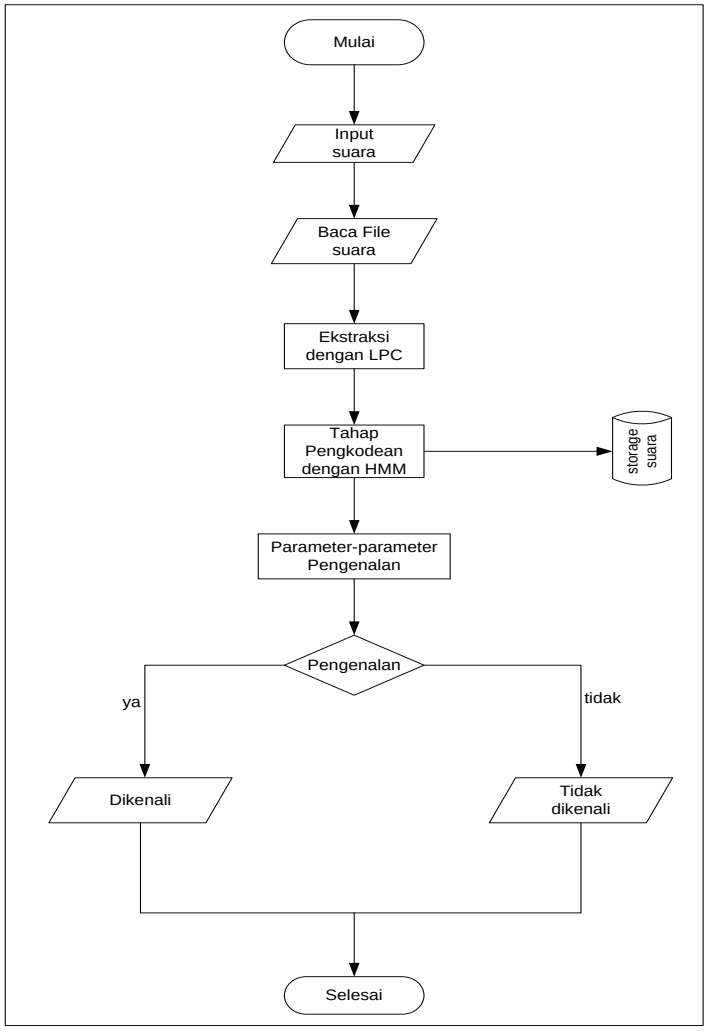
3.5.1. Bagan Alir Sistem

Bagan alir sistem *(System Flowchart)* merupakan bagan yang menunjukan atas pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan prosedur-prosedur yang ada dalam sistem. Bagan alir sistem menunjukan apa yang dikerjakan oleh sistem.

3.5.2. Diagram Alir Pengenalan Suara

Sinyal suara manusia yang diterima dengan menggunkan microphone (sinyal analog) dicuplik/disampling sehingga menjadi sinyal digital. Proses pencuplikan sinyal ini dilakukan dengan menggunkan sound card yang terdapat pada personal computer. Pencuplikan dilakukan pada kecepatan 800 Hz dengan resolusi 8 bit(1 byte) sehingga didapat data sebanyak 8000 byte tiap detik. Kecepatan pencuplikan tersebut dilakukan dengan didasarkan asumsi. Sinyal

digital hasil cuplikan ini terlebih dahulu dinormalisasi kemuddian diproses dengan preprocessing signal yang menggunkan metode LPC sehingga didapat beberapa koefisien LPC. Kemudian Koefisien LPC tersebut dirubah unutk dikodekan dengan metode model HMM tujuan agar perbedaaan antar pola kata yang satu dengan yang lain terlihat lebih jelas sehingga ekstraksi parameter sinyal memberikan hasil yang lebih baik. Proses pengenalan denga membandingkan cirri dan parameter pengenalan dangan hasil data yang sudah dilatihkan terlebih dahulu untuk dijadikan acuan dalam pengenalan ini.



Gambar 3.40. Diagram alir pengenalan

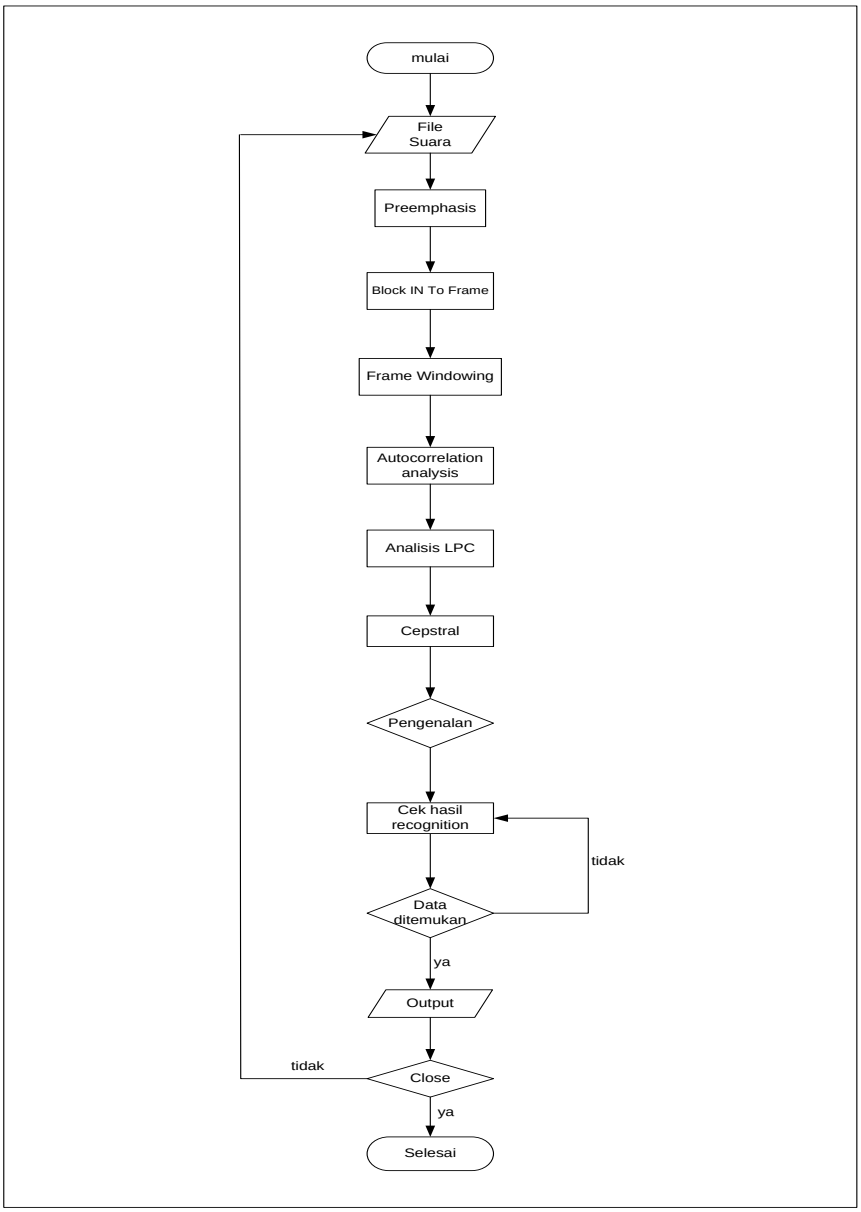
Sinyal ucapan direkam oleh komputer, yang asalnya masih berupa sinyal analog kemudian dirubah menjadi sinyal digital untuk mempermudah proses berikutnya. Hasil rekaman sinyal ini dianalisis oleh proses Ekstraksi sinyal dengan menggunakan LPC. Hasil dari ekstraksi tersebut dirubah menjadi parameter-parameter atau dijadikan ciri untuk selanjutnya disesuaikan atau dibandingkan dengan data referensi yang telah disiapkan sebelumnya pada Training Data pada saat pelatihan. Hasil proses tadi kemudian dibandingakan pada proses pencarian dengan data dari training data/data pelatihan dengan menerapkan beberapa model, baik hanya satu model atau gabungan beberapa model dari Acustic, Lexical, Language Model. Proses perbandingan ini ditunjukan untuk mencari kemiripan tertinggi dari suatu data suara yang diinputkan pada proses pengenaalan dibandingkan dengan hasil data pada saat pelatihan yang dilakukan pada pertama kali. Keseluruhan proses tersebut akan diulang unutk mengolah ucapan berikutnya untuk dijadikan acuan pada tahap pengenalan ini apakah suara akan dikenali atau tidak dikenal oleh aplikasi ini.

Untuk membangun aplikasi identifikasi kata menggunkan speech recognition. Diperlukan sebuah *Engine Speech Recognition*. *Engine* yang dibangun menggunakan metode *Hidden Markov Model* (HMM).

3.5.3. Ekstraksi Sinyal dengan LPC

Linear Predictive Coding (LPC) merupakan salah satu teknik analisis sinyal percakapan yang paling *powerful* dan menyediakan ekstraksi fitur yang berkualitas baik dan efisien untuk digunakan dalam perhitungan. LPC pertama kali digunakan pada tahun 1978 untuk membuat alat sintesis sinyal percakapan.

LPC melakukan analisis dengan cara memperkirakan *formant*, memisahkan *formant* dari sinyal, yang dinamakan proses *inverse filtering*, lalu mengestimasi intensitas dan frekuensi dari sinyal percakapan yang tersisa, yang disebut *residue.16* Karena sinyal percakapan bervariasi seiring waktu, estimasi tersebut dilakukan untuk setiap potongan kecil dari sinyal, yang dinamakan *frame.*



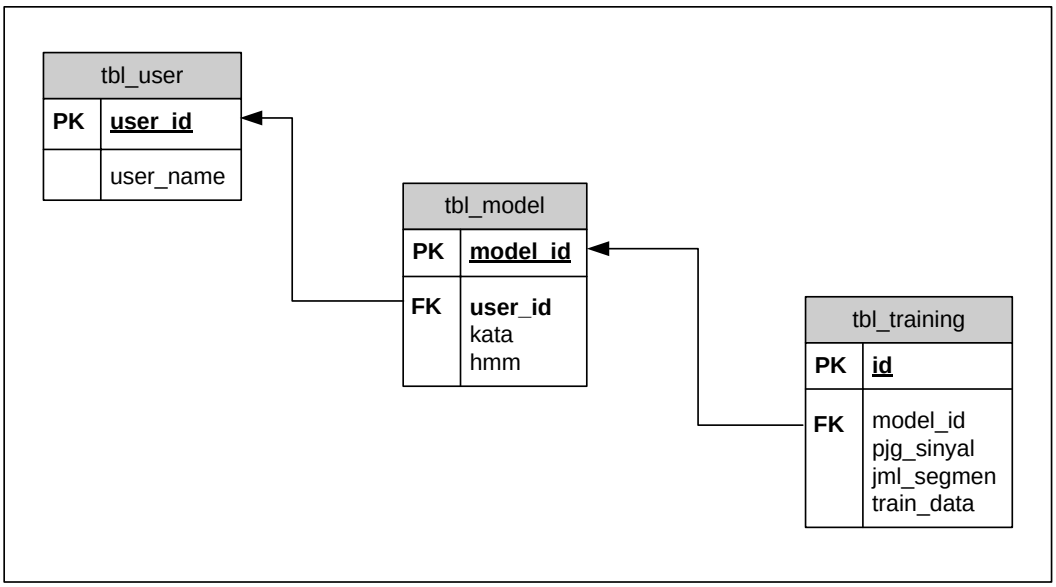
Gambar 3.41. Diagram alir proses eketraksi siinyal pada LPC

3.5.4. Perancangan *Database*

Database merupakan tempat penyimpanan data atau media penyimpanan data yang tersimpan diluar komputer yang dapat digunkan dangan aplikasi atau perangkat lunak yang akan dibuat untuk memanipuloasinya seperti merubah atau menghapus pada isi *database* tersebut. Didalam perancangan database ini penyusun menggunkan aplikasi ini dengan menggunakan Microsoft Office Access.

3.5.4.1.Perancangan Relasi Database

Model basis data relasional menunjukan suatu cara atau mekanisme yang digunakan untuk mengelola data secara fisik dalam dlam memori skunder yang akan berdampak pula pada bagaimana kita mengkelompokkan dan membentuk keseluruhan data yang terkait dalam sistem yang akan kita tinjau pada diagram ini akan digambarkan basis data relasional yang terlihat dengan speech recognition.



Gambar 3.42. Relasi Table Aplikasi Pengidentifikasian Kata

Relasi diatas dapat dijelaskan sebagai berikut :

1. Pada tbl_user berelasi dengan tbl_model dimana user_id sebagai Primary key pada tbl_user mempunyai entitas satu ke banyak terhadap user_id di tbl_model

2. Pada tbl_model berelasi dengan tbl_training dimana model_id sebagai Primary key pada tbl_model mempunyai entitas satu ke banyak terhadap model_id pada tbl_training

3.5.4.2.Struktur Tabel

Struktur tabel merupakan urutan tampilan atau data yang digunakan dalam aplikasi. Struktur table dalam palikasi identifikasi kata menggunakan teknologi speech recognition, dapat dilihat pada tabel-tabel sebagai berikut :

1. Struktur Tabel Pengguna

Untuk menyimpan data-data Pengguna yang dimasukan oleh pengguna.

Nama File : SpeechRecognition.mdb

Nama Tabel: tbl_user

Tabel 3.1. Struktur tabel pengguna

Nama Field	Type	Length	Keterangan
User_id	AutoNumber	-	Primary key
User_Name	Text	-	

2. Sturktur Tabel Model HMM

Untuk menyimpan data-data hasil analisis sinyal pengenalan suara yang telah dirubah kedalam bentuk symbol *Hidden Markov Model* (HMM).

Nama File : SpeechRecognition.mdb

Nama Tabel : tbl_model

Tabel 3.2. Struktur tabel model HMM

Nama Field	Type	Length	Keterangan
model_id	AutoNumber	-	Primary Key
user_id	Number	-	
kata	Text		
HMM	Memo		

3. Struktur Tabel Training

Untuk menyimpan data-data dari hasil pelatihan dari pengguna yang telah dirubah kedalam betuk simbol *Hidden Markov Model* (HMM).

Nama File : SpeechRecognition.mdb

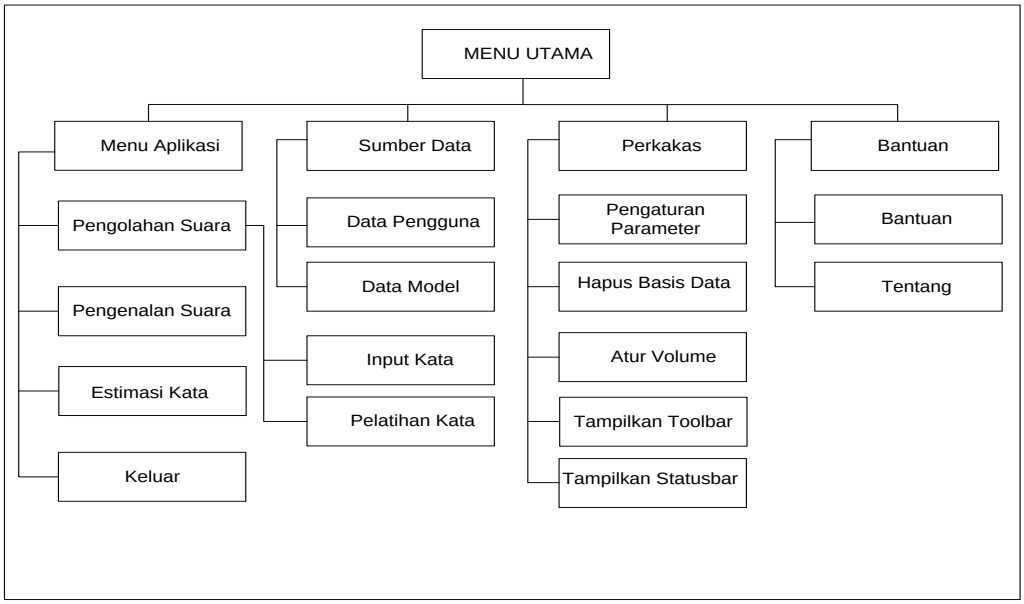
Nama Tabel : tbl_training

Tabel 3.3. Struktur tabel HMM Training

Nama Field	Type	Length	Keterangan
id	AutoNumber	-	Primary Key
Model_Id	Number	-	
Pjg_Sinyal	Number		
Jml_segmen	Number		
Train_data	Memo		

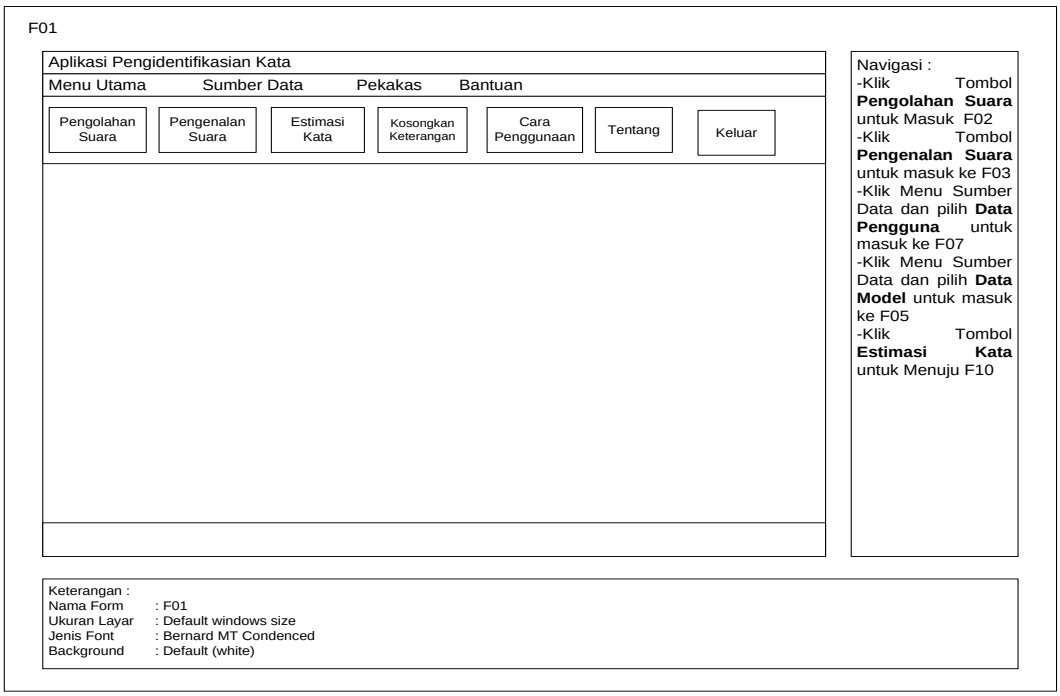
3.5.5. Perancangan Struktur Program

Perancangan struktur program aplikasi identifikasi kata yang akan dibangun.

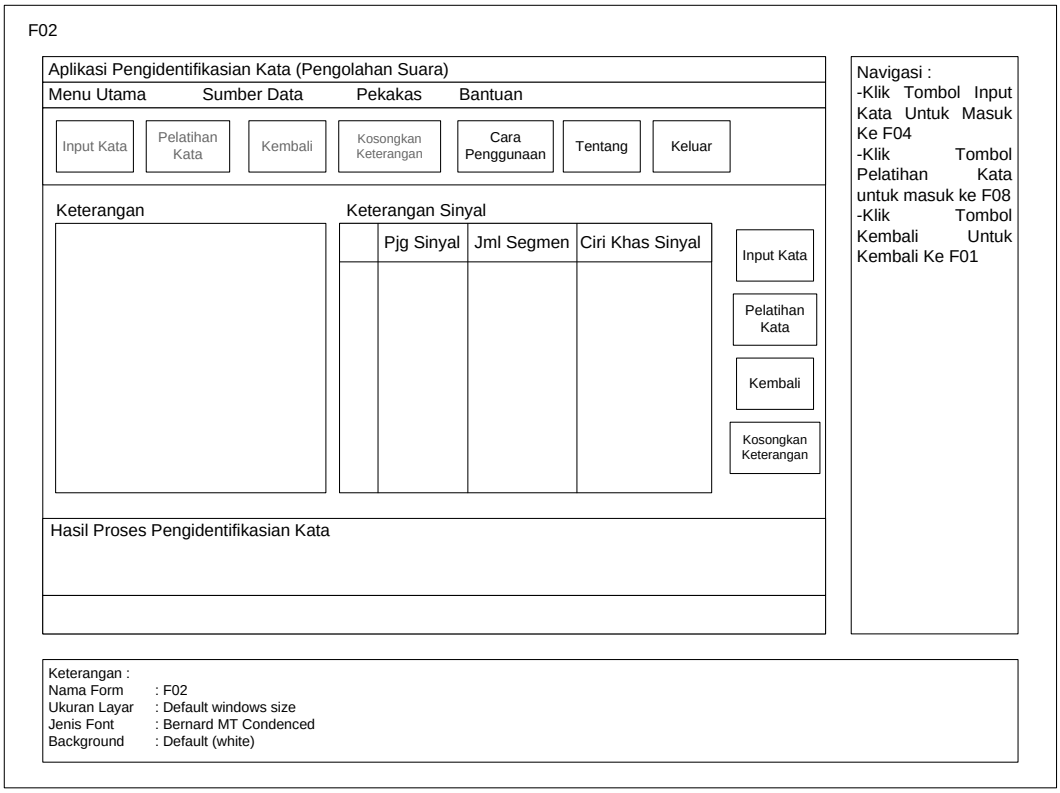


Gambar 3.43. Perancangan Strutur Program

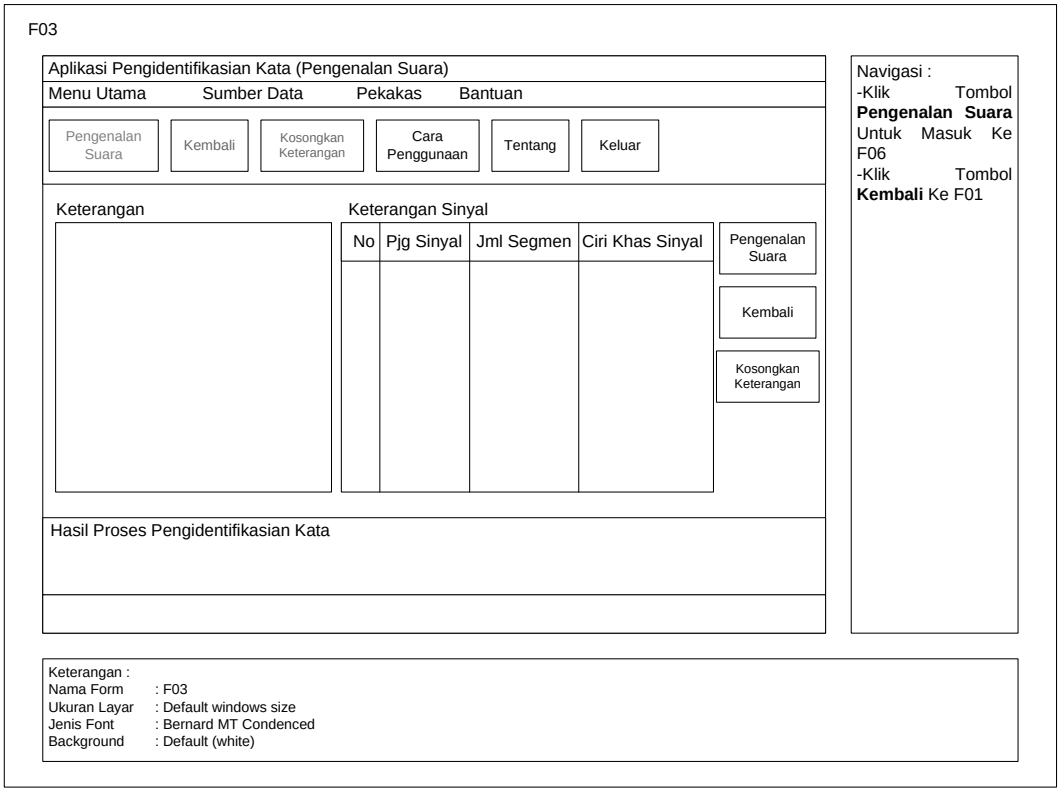
3.5.6. Perancangan Antarmuka



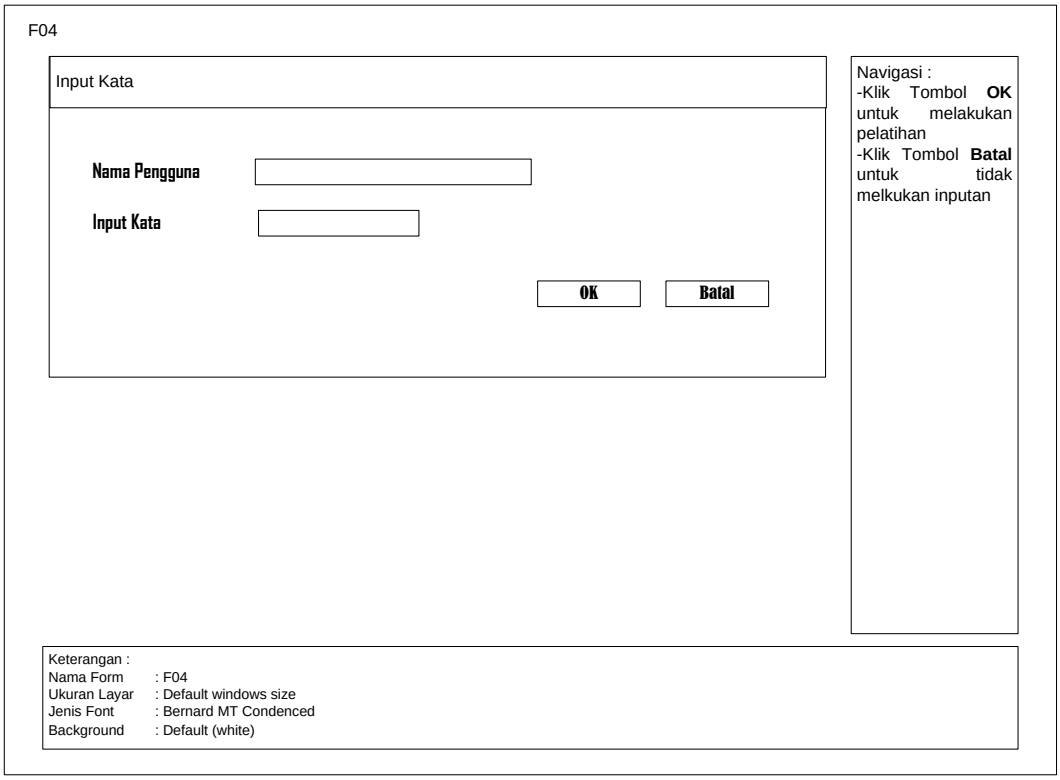
Gambar 3.44. Perancangan Antarmuka Menu Utama



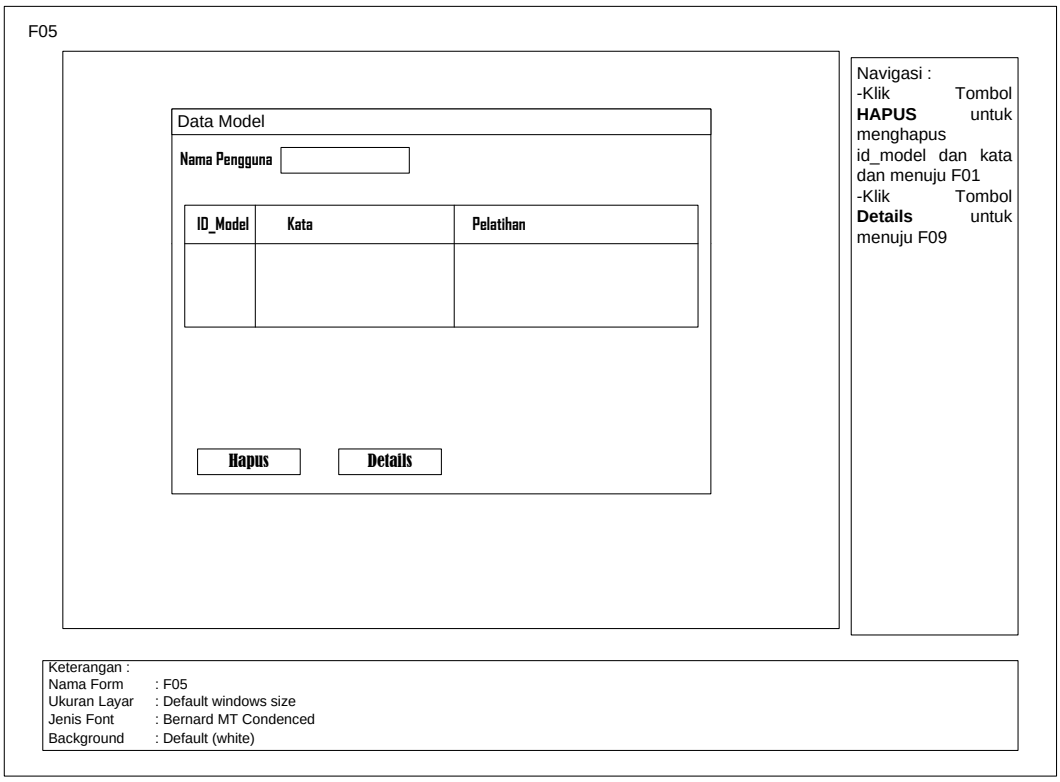
Gambar 3.45. Perancangan Antarmuka Pengolahan Suara



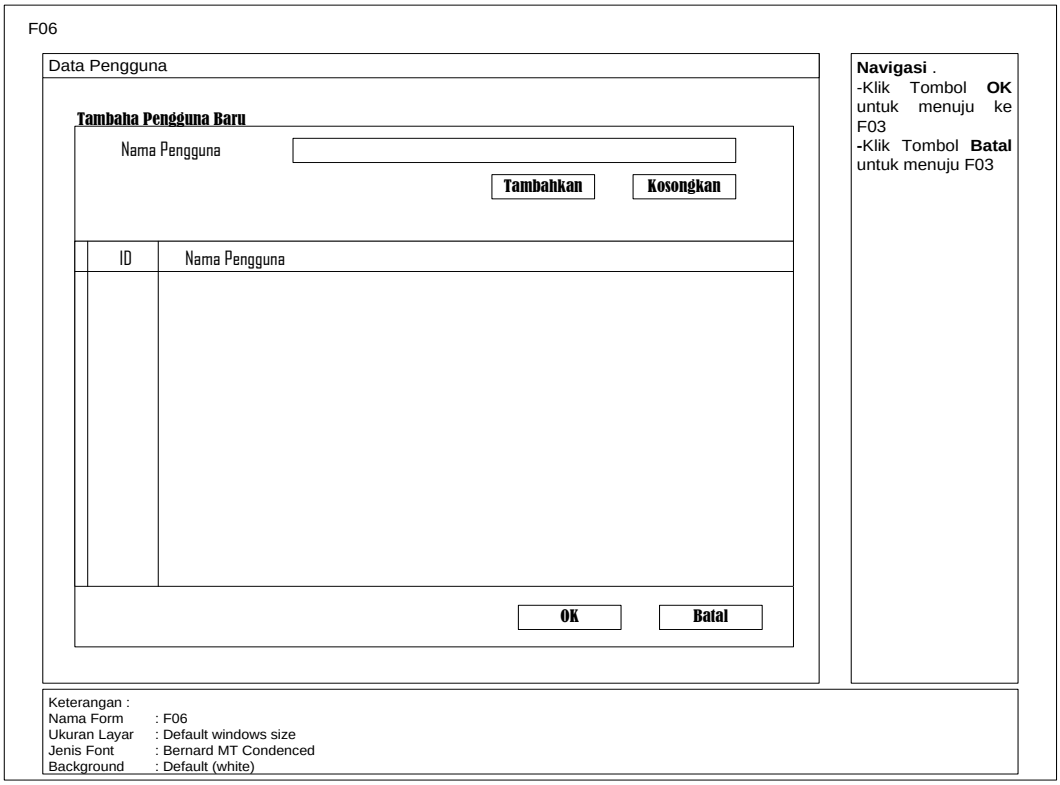
Gambar 3.46. Perancangan Antarmuka Pengenalan Suara



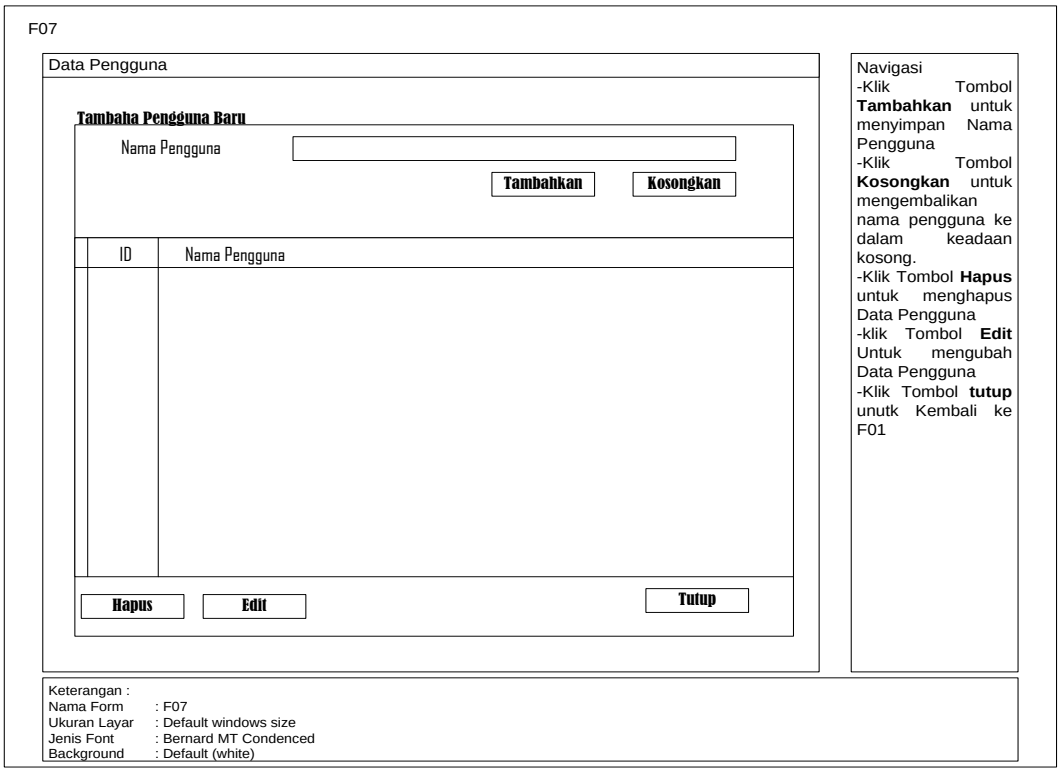
Gambar 3.47. Perancangan Antarmuka Input Kata



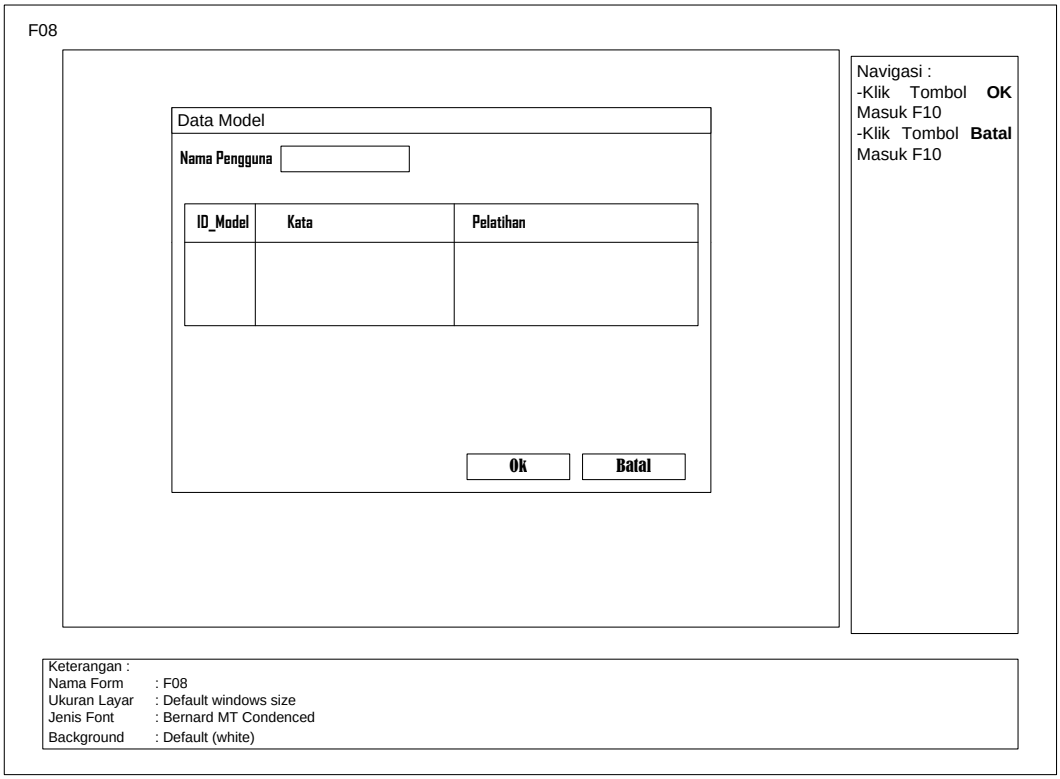
Gambar 3.48. Perancangan Antarmuka Data Model



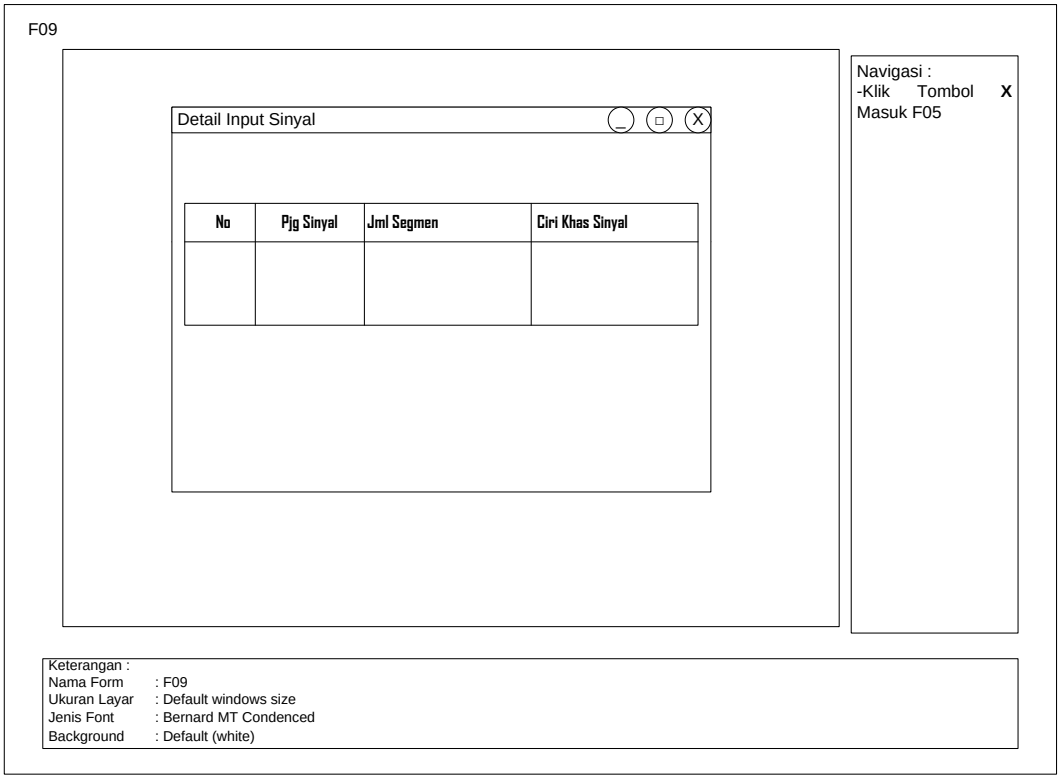
Gambar 3.49 Perancangan Antarmuka Data Pengguna



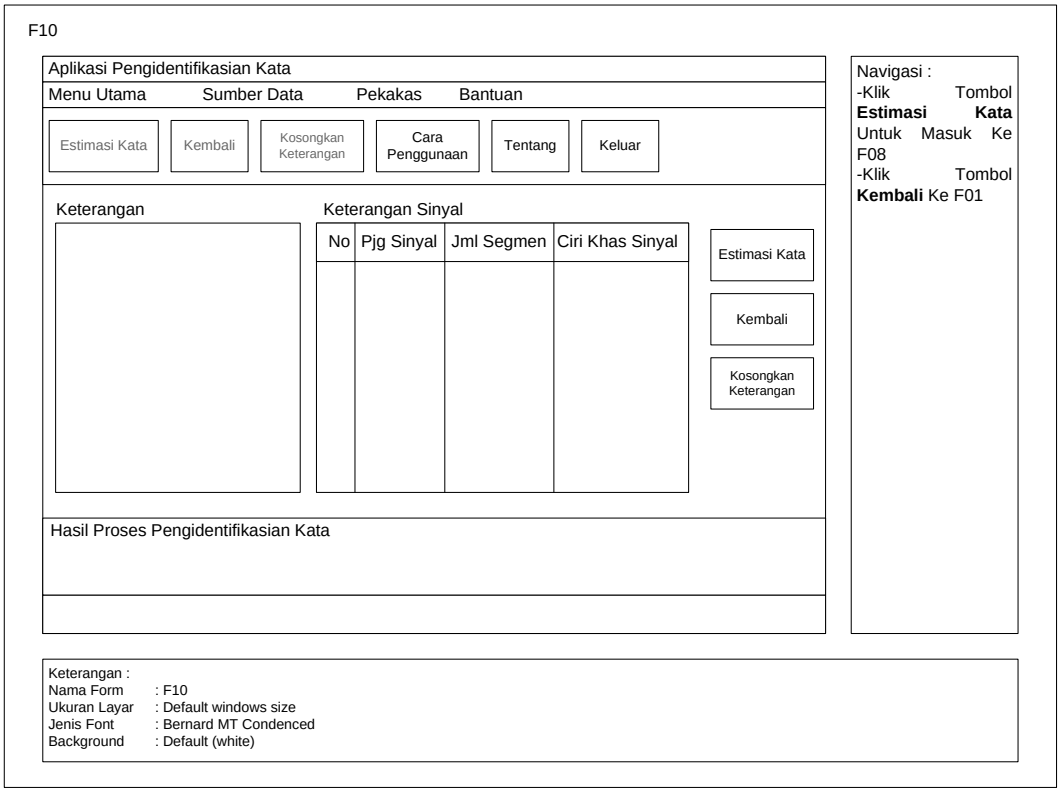
Gambar 3.50. Perancangan Antarmuka Data Tambah Pengguna



Gambar 3.51. Perancangan Antarmuka Data Model Terisi

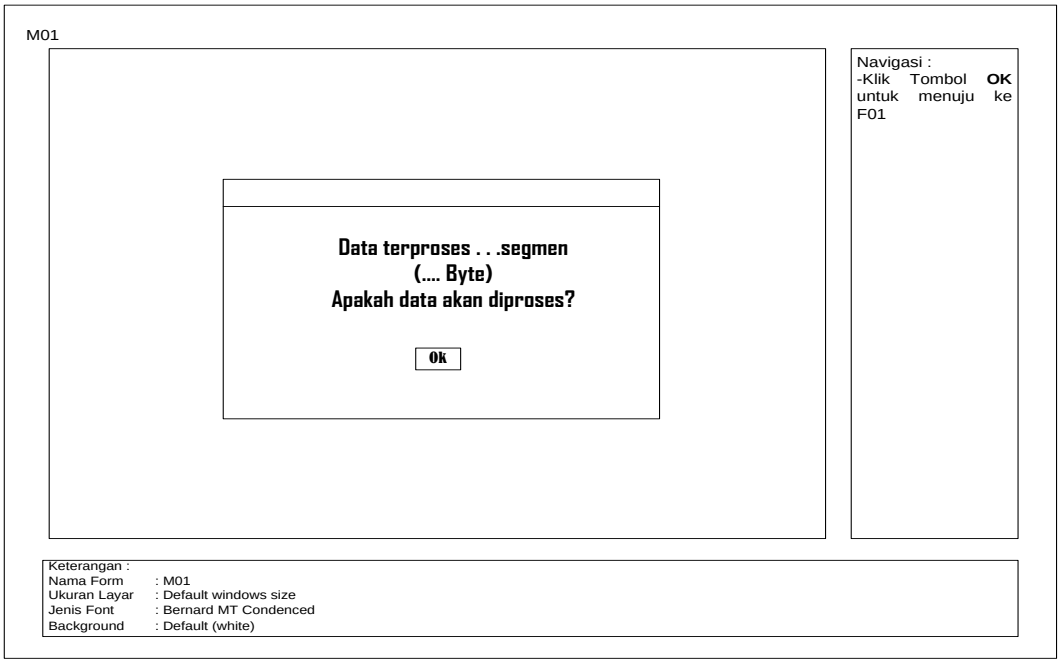


Gambar 3.52. Perancangan Antarmuka Detail Input Sinyal

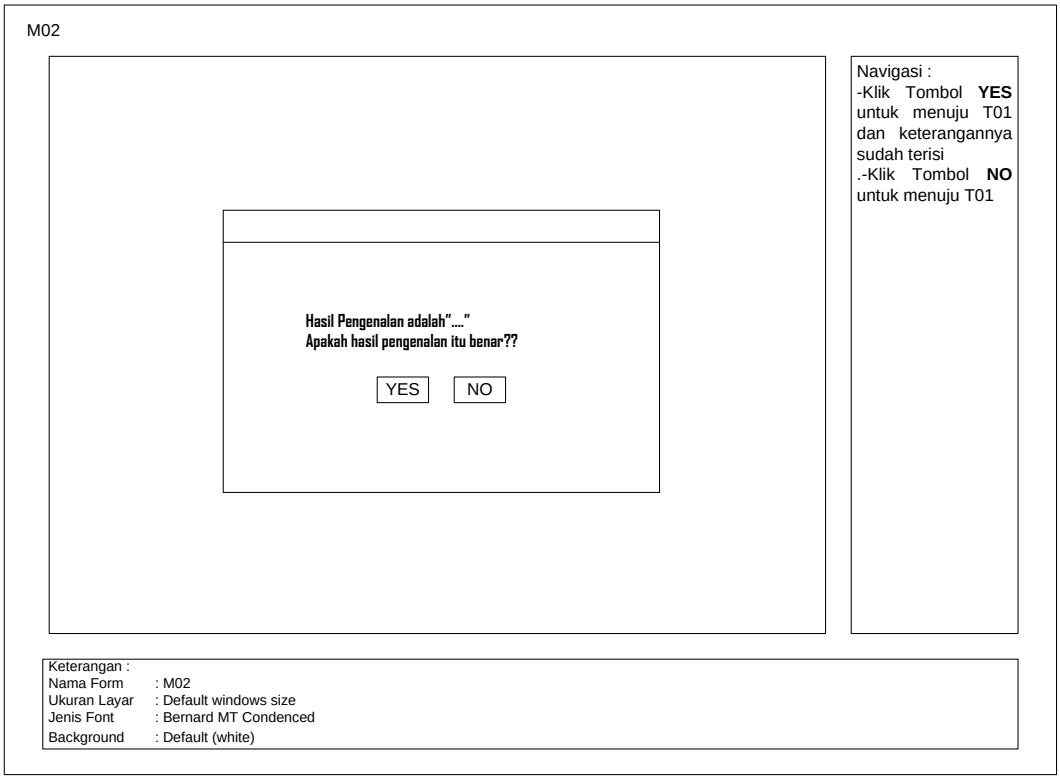


Gambar 3.53. Perancangan Antarmuka Estimasi Kata

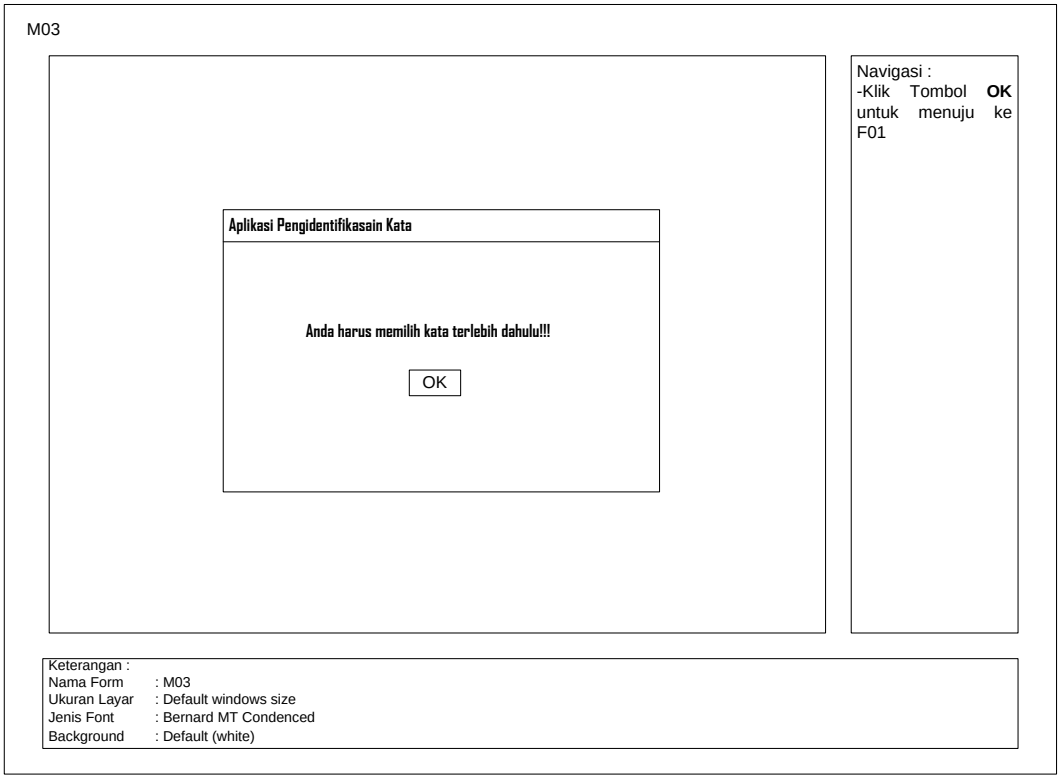
3.5.7. Perancangan Pesan



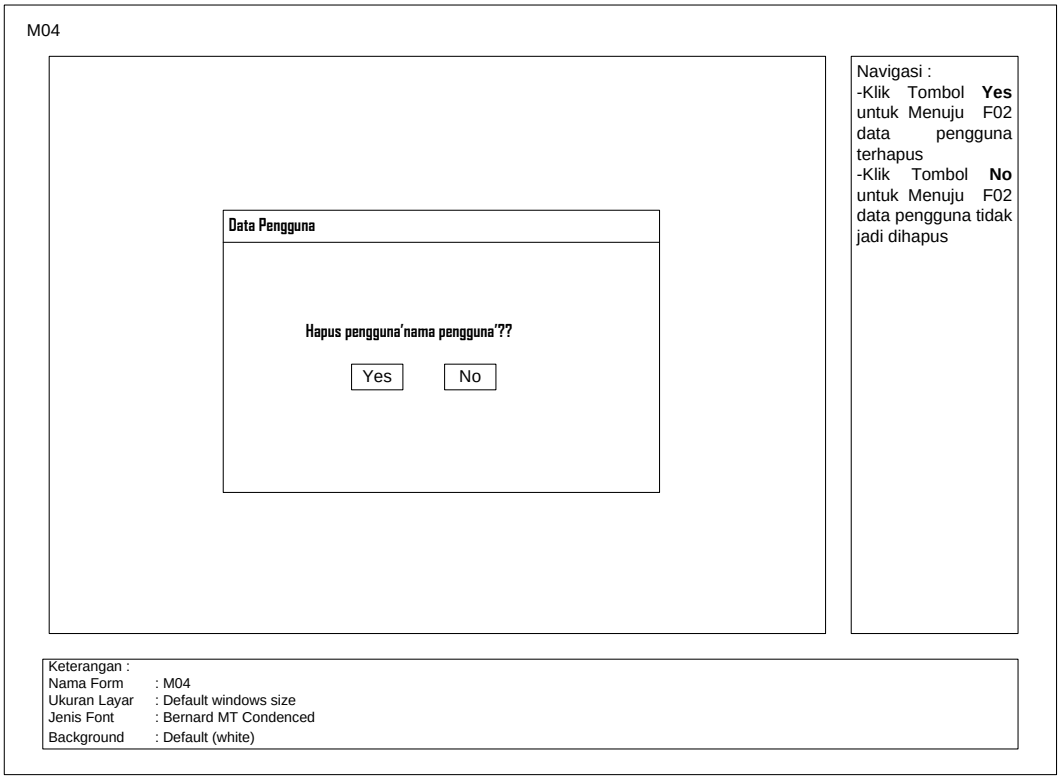
Gambar 3.54. Perancangan Pesan Suara Berhasil Terproses



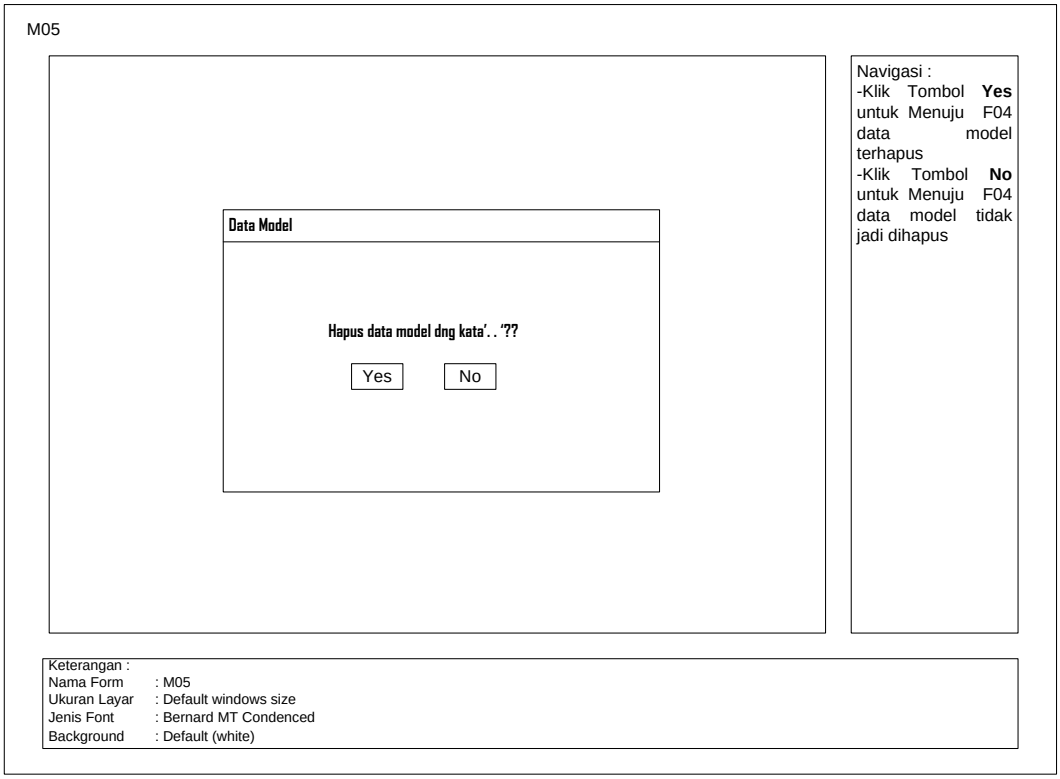
Gambar 3.55. Perancangan Pesan Suara Benar Atau Salah



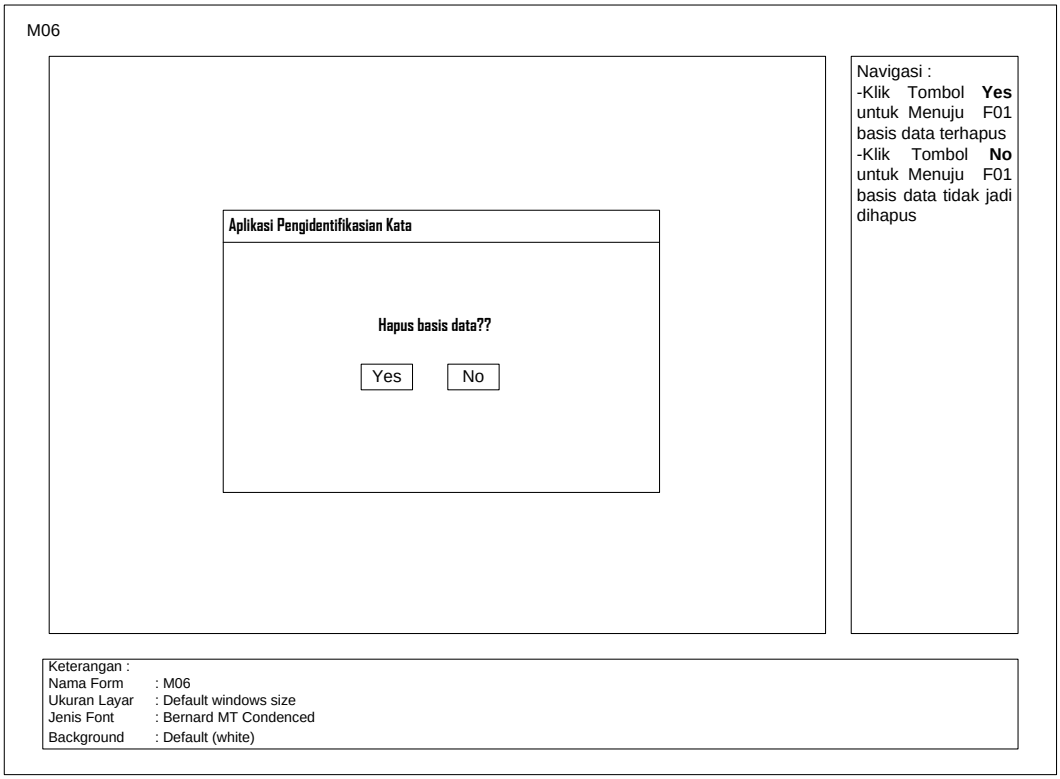
Gambar 3.56. Perancangan Pesan Memilih Kata Terlebih Dahulu



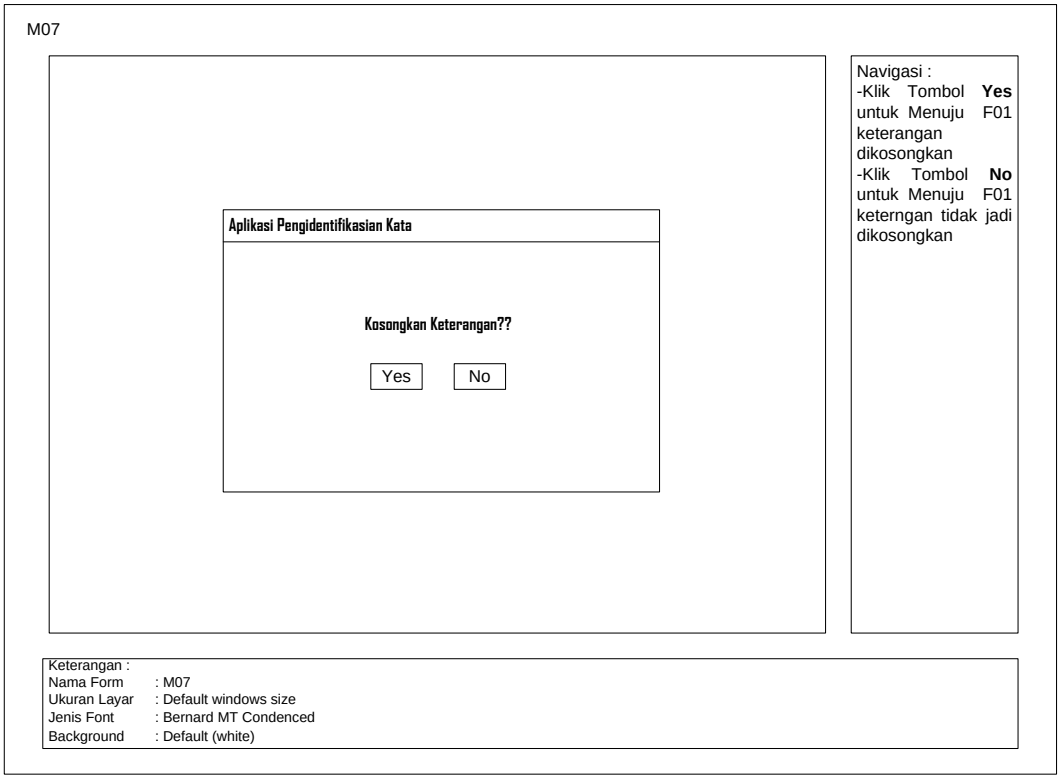
Gambar 3.57. Perancangan Pesan Hapus Nama Pengguna



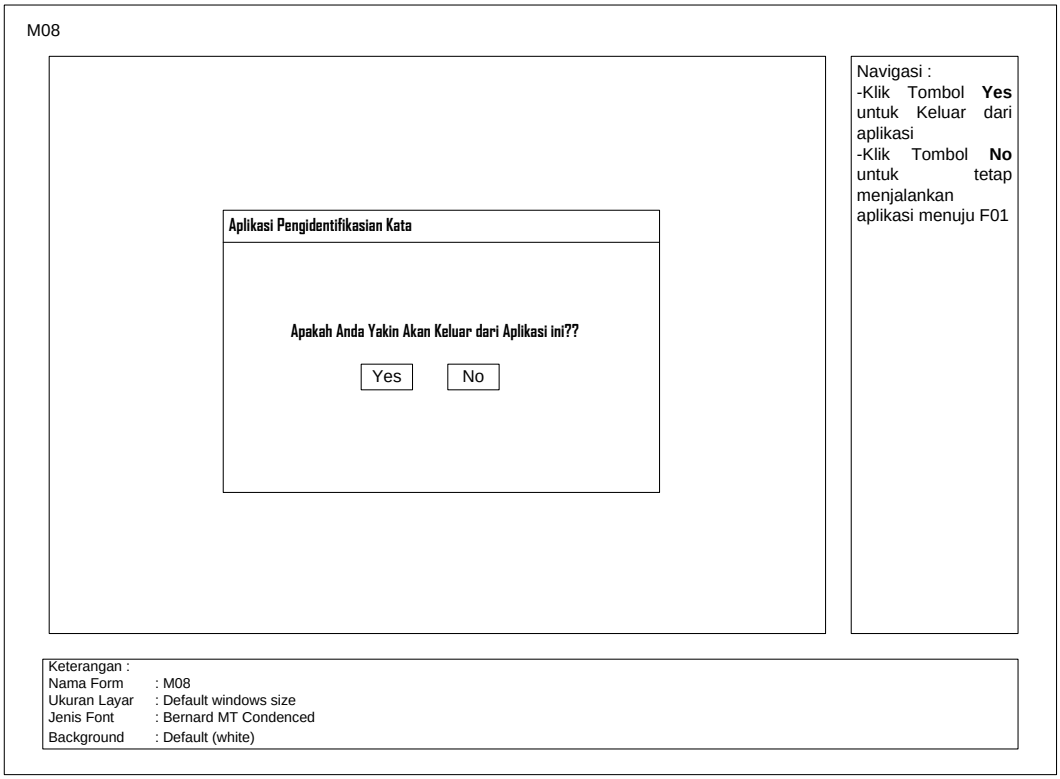
Gambar 3.58. Perancangan Pesan Hapus Data Model



Gambar 3.59. Perancangan Pesan Hapus Basis Data

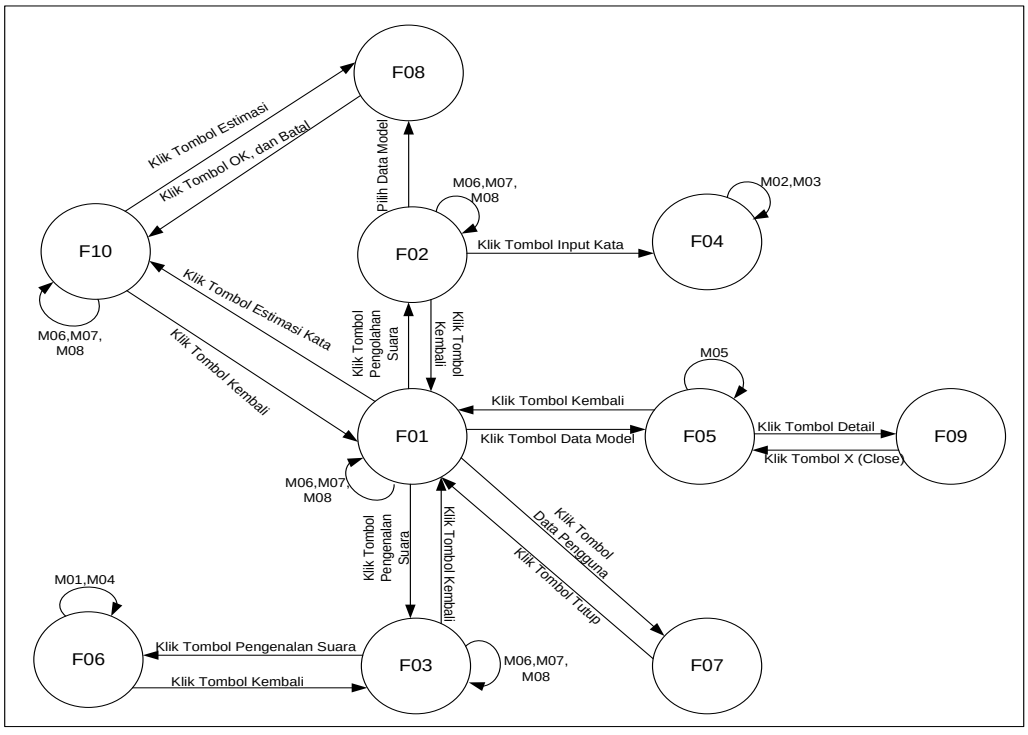


Gambar 3.60. Perancangan Pesan Kosongkan Ketrangan



Gambar 3.61. Perancangan Pesan Keluar Aplikasi

3.5.8. Jaringan Semantik



Gambar 3.62. Jaringan Semantik