BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem dilakukan untuk memperoleh informasi dari permasalahan dan gambaran yang tepat dari apa yang akan dilakukan oleh sistem. Analisis sistem bertujuan untuk mengetahui bagaimana sistem bekerja, sistem yang akan dibangun dan bagaimana sistem yang akan diteliti.

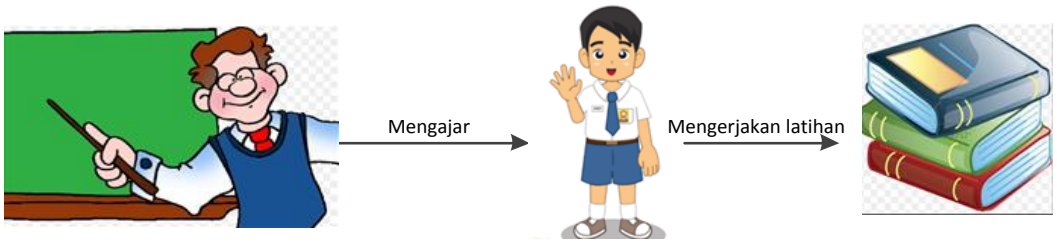
Pada bagian analisis sistem ini akan dibahas mengenai analisa-analisa masalah yang akan dilakukan terhadap aplikasi, analisa pengguna dan kegunaan dari aplikasi yang akan dibangun.

3.1.1 Analisis Masalah

Berdasarkan latar belakang yang telah diuraikan maka masalah yang didapat adalah bagaimana cara membangun sebuah aplikasi pembelajaran dengan menggunakan teknologi *voice recognition* untuk membantu perkembangan dan kemampuan siswa *down syndrome* untuk mengucapkan kata dan juga membantu guru dalam proses belajar dan mengajar.

3.1.2 Analisis Prosedur Yang Sedang Berjalan

Analisis Prosedur atau analisis proses adalah tahapan yang memberi gambaran tentang sistem yang sedang berjalan. Analisis ini bertujuan untuk memberikan gambaran yang lebih detail bagaimana cara kerja dari sistem yang sedang berjalan. Analisa Prosedur pada bisnis proses aplikasi ini pada gambar 3.1



Gambar 3.1 Bisnis Proses Mengajar

Gambar 3.1 adalah alur bisnis proses mengajar Sekolah Khusus YPAC. Berdasarkan hasil penelitian, sistem yang sedang berjalan masih terdapat kendala yaitu, ketika murid sulit menangkap pelajaran yang diajarkan oleh guru, menjawab pertanyaan, mengingat dan menjawab latihan yang ada dibuku. Buku menjadi bahan acuan utama dalam penyampaian materi pembelajaran ataupun latihan yang diberikan kepada murid tanpa didukungnya perangkat multimedia. Berikut ini adalah analisis dari bisnis proses yang sedang berjalan :

1. Guru mengajar murid sesuai dengan jadwal atau bahan ajaran harian.
2. Guru mensimulasikan bahan ajaran dengan menggunakan alat peraga.
3. Murid mengerjakan latihan soal dengan dibimbing oleh guru.

Dari gambaran prosedur tersebut Sekolah Khusus YPAC masih terkendala yakni murid yang masih kesulitan mengingat mengenal kata dan pengucapan disebabkan murid masih belum bisa membayangkan bentuk huruf dan pengucapanya sehingga mengakibatkan murid sulit menghafal kata.

3.1.3 Analisis *Aplikasi* Sejenis

Pada tahap ini dilakukan analisis terhadap *aplikasi* sejenis yang telah dibuat. Tujuannya dari analisis ini yaitu membandingkan dan mendapatkan konsep *voice recognition* dari *aplikasi* yang sudah ada untuk kemudian dikembangkan dan disesuaikan denga kebutuhan. *aplikasi* yang dibandingkan dan akan menjadi acuan adalah *aplikasi pembelajaran* bahasa Inggris *English Kids*.

3.1.3.1 English Pronunciation Testing

English pronunciation testing dibuat oleh Ruiyu team, *english pronunciation testing* adalah *aplikasi* tes pengucapan kata bahasa inggris.



Gambar 3.2 Halaman Awal

1.1.3.1.1 Cara Penggunaan *English Pronunciation Testing*

Didalam menu utama tes pengucapan bahasa inggris ini terdapat 2 pilihan menu yaitu: menu pengembang dan menu *level.*

1. Menu pengembang

   Berisi tentang pengembang dan alamat situs web pengembang



Gambar 3.3 Menu Pengembang

2. Level

   Pada bagian ini terdapat 3 level dari level 1 sampai dengan 3 dengan tingkat kesulitan nya masing-masing.



Gambar 3.4 Tampilan Menu Utama

3. Menu pilihan test

Pada menu pilihan test terdapat 5 pilihan menu utama yaitu  untuk kembali ke menu sebelumnya, menu pilihan  digunakan untuk mendengarkan contoh pengucapan kata, *previous* digunakan untuk mengulang kata sebelumnya, *next* digunakan untuk melanjutkan kekata selanjutnya, *please read* digunakan untuk merekam suara.



Gambar 3.5 Level

1.1.3.1.2 Komponen Aplikasi *English Pronunciation Testing*

Pada aplikasi *english pronunciation testing* terdapat komponen yang dapat digunakan dapat dilihat pada tabel

Tabel 3.1 Kompenen Aplikasi English Pronunciation Testing

No	Komponen	Keterangan
1	Menu	Pada aplikasi *english pronunciation testing* terdapat beberapa menu sebagai berikut: Menu Pengembang Menu utama level 1 Submenu kembali Submenu mendengarkan Submenu *previous* Submenu *please read*

Tabel 3.1 Kompenen Aplikasi English Pronunciation Testing (lanjutan)

No	Komponen	Keterangan
1	Menu	Submenu *next* Submenu *share* Menu utama level 2 Submenu kembali Submenu mendengarkan Submenu *previous* Submenu *please read* Submenu *next* Submenu *share* Menu utama level 3 Submenu kembali Submenu mendengarkan Submenu *previous* Submenu *please read* Submenu *next* Submenu *share*

2	Grafis	Aplikasi ini memiliki grafis 2 dimensi
3	Tombol	Aplikasi *english pronunciation testing* memiliki tombol yang digunakan untuk mengarahkan pengguna sesuai dengan fungsi masing-masing berikut adalah tombol yang digunakan: Tombol pengembang Digunakan untuk mengetahui informasi tentang pengembang Tombol level 1 Digunakan untuk menampilkan test pengucapan di level 1 Tombol level 2 Digunakan untuk menampilkan test pengucapan di level 2 Tombol level 3 Digunakan untuk menampilkan test pengucapan di level 3 Tombol kembali Digunakan untuk kembali kemenu utama Tombol  Digunakan untuk mendengarkan contoh pengucapan kata

Tabel 3.1 Kompenen Aplikasi English Pronunciation Testing (lanjutan)

No	Komponen	Keterangan
3	Tombol	Tombol *previous* Digunakan untuk kembali kekata sebelumnya Tombol please read Digunakan untuk merekam kata yang diucapkan Tombol next Digunakan untuk melanjutkan kekata berikutnya Tombol share Digunakan untuk membagikan hasil test ke media Sosial
4	Fungsi Aplikasi	Fungsi aplikasi *english pronunciation testing* Mengetest kemampuan pengucapan kata Menampilkan hasil dari pengucapan kata

1.1.3.1.3 Hasil Analisis Aplikasi *English Pronunciation Testing*

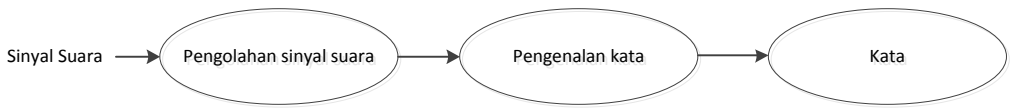
Hasil analisis aplikasi *english pronunciation testing* adalah sebagai berikut :

1. Aplikasi *english pronunciation testing* merupakan aplikai multimedia yang interaktif, aplikasi ini menyajikan materi test pengucapan kata unruk membantu penggunanya mengucapkan kata dengan benar.
2. Aplikasi *english pronunciation testing* harus terhubung dengan jaringan internet.

3.1.4 Deskripsi Umum Sistem

Dari permasalahan-permasalahan yang ada, solusi yang ditawarkan adalah membangun *aplikasi* pembelajaran untuk melatih kemampuan pengucapan kata pada anak sehingga dengan menyediakan aplikasi pembelajaran pengucapan kata ini diharapkan anak dapat menjadi lebih termotivasi belajar. Sistem yang akan dibangun pada sistem ini adalah aplikasi menggunakan teknologi *voice recognition*, sistem ini mampu menangkap suara dan membandingkan dengan data yang disimpan pada *database*.

Deskripsi umum aplikasi pembelajaran pengucapan kata ini adalah dimulai dari tahap pengolahan sinyal suara. Tahap pengolahan sinyal suara adalah proses pengkondisian sinyal suara dimulai dari perekaman suara dan tahap konversi analog ke digital. Setelah itu masuk pada tahap ekstraksi dengan sebelumnya sinyal diolah menggunakan algoritma FFT. Gambar 3.6 menunjukan proses pengenalan kata yang akan dirancang.

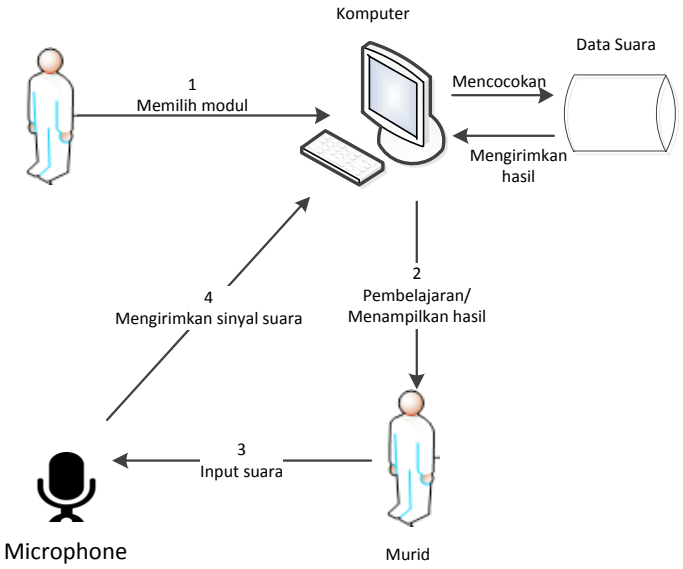


Gambar 3.6 Proses Aplikasi Pengenalan Kata

Analaisis algoritma FFT adalah tahap merubah sinyal suara dalam domain waktu menjadi sinyal suara dalam domain frekuensi. Tahap pengenalan kata adalah tahap pencocokan hasil rekaman dengan basis data, tahap ini melakukan perbandingan hitungan kata satu dengan yang lain berupa paling banyak baris yang sama dengan basis data.

3.1.5 Analisis Aplikasi Yang Akan di Bangun

Dari permasalahan-permasalahan yang ada, solusi yang ditawarkan adalah membangun *aplikasi* pembelajaran untuk melatih kemampuan pengucapan kata pada anak sehingga dengan menyediakan aplikasi pembelajaran pengucapan kata ini diharapkan anak dapat menjadi lebih termotivasi belajar. Sistem yang akan dibangun pada sistem ini adalah aplikasi menggunakan teknologi *voice recognition*, sistem ini mampu menangkap suara dan membandingkan dengan data yang disimpan pada *database*.



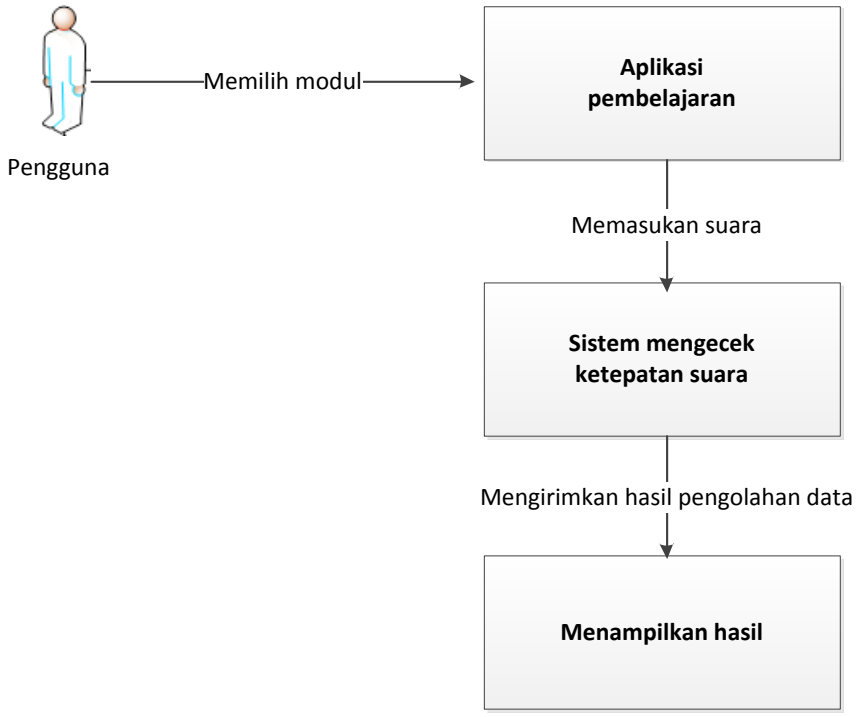
Gambar 3.7 Analisis Sistem yang akan dibangun

Berikut ini adalah alur pada sistem *aplikasi* pembelajaran untuk melatih kemampuan pengucapan kata pada anak yang akan dibangun gambar 3.7:

1. Guru memilih materi yang terdapat didalam aplikasi.
2. Murid memperhatikan materi yang telah dipilih oleh guru, materi tersebut berupa sebuah animasi sehingga materi yang disampaikan lebih interaktif.
3. Setelah mendengarkan materi murid meniru dan memasukan suara sesuai contoh lewat *Mic*.
4. Sistem akan mengecek hasil masukan.
5. kemudian sistem akan menampilkan informasi hasil suara yang dimasukan.

3.1.6 Analisis Arsitektur Sistem

Arsitektur sistem yang akan dibangun terdiri dari beberapa komponen yaitu mengenai sistem aplikasi pembelajaran dan sistem pengenalan ucapan. Arsitektur sistem dalam pengembangan aplikasi pembelajaran pengucapan kata ini ditunjukan pada gambar 3.8 mengenai analisis arsitektur sistem.



Gambar 3.8 Analisis Arsitektur Sistem

Gambar 3.8 menunjukan garis besar proses analisis arsitektur sistem

1. Sistem akan menampilkan modul yang dipilih oleh murid kemudian sistem akan menampilkan contoh pengucapan kata yang dipilih oleh murid
2. Kemudian murid akan menginputkan suara.
3. Sistem akan mengecek ketepatan suara, deskripsi secara umum dalam pengucapan kata dimulai dari tahap pengolahan sinyal suara.setelah itu masuk pada tahap *analog to digital conversion*. Setelah itu sinyal diolah menggunakan analisis Fast Fourier Transform (FFT).
4. Kemudian sistem akan mengirimkan hasil pengolahan data dan kemudain menampilkan hasil dari pengolahan tersebut.

3.1.7.1 Skoring

Pada setiap materi latihan pemain harus menyelesaikan soal yang diberikan secara acak, pemain akan mendapatkan penilaian berupa bintang diakhir latihan. Pada Tabel 3.6 menjelaskan rincian perhitungan untuk skoring jawaban.

Tabel 3.2 Analisis Skoring Permainan

Pengucapan Kata	Jumlah Bintang
1 sampai 3	3
3 sampai 6	2
6 sampai 10	1

3.1.7.2 Deskripsi Konsep Aplikasi Pembelajaran Pengucapan Kata

Aplikasi ini ditujukkan untuk melatih kemampuan pengucapan kata kepada anak. Pada tampilan awal permainan terdapat menu utama yaitu guru dapat memilih materi mana yang akan diberikan kepada murid, setelah memilih materi maka anak dapat mengikuti intruksi yang ada didalam aplikasi. Tugas murid adalah menyelesaikan soal yang terdapat didalam setiap materi hingga selesai.

Tabel 3.3 Deskripsi Konsep Aplikasi

Unsur yang Membangun	Deskripsi
Judul	Aplikasi pembelajaran pengucapan kata pada

	anak penderita down syndrome
Audiens	Siswa *downsyndrome* yang sedang belajar di sekolah YPAC
Gambar	Gambar yang ada dalam aplikasi pembelajaran merupakan gambar yang disesuaikan dengan anak-anak sehingga anak merasa terhibur.
Audio	*Audio* yang digunakan terdiri dari narasi dan *backsound* yang telah diedit dan disesuaikan dengan anak.
Interaktivitas	Interaktivitas dalam aplikasi pembelajaran ini adalah pengucapan kata, tombol lanjut, kembali dan keluar.

Adapun materi yang akan dibahas dalam aplikasi pembelajaran pengucapan kata bisa dilihat pada tabel 3.4

Tabel 3.4 Materi Dan Database

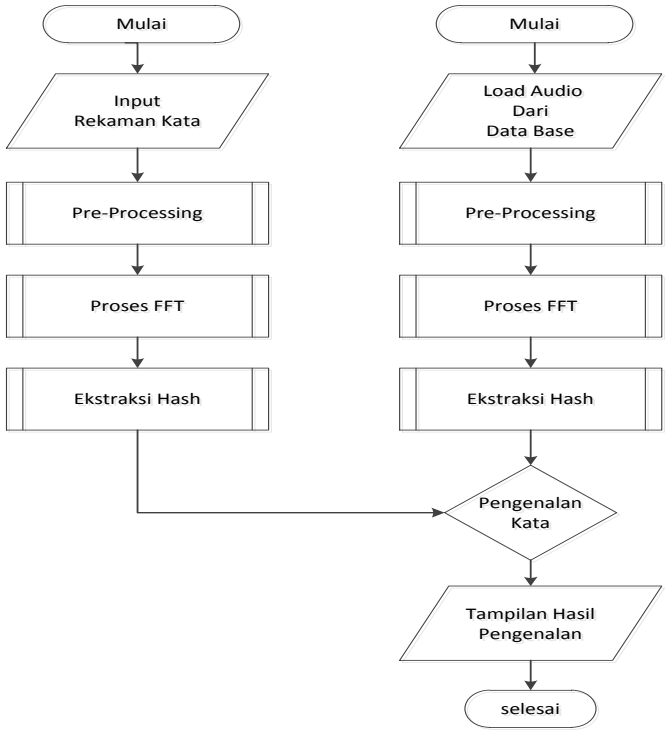
Materi	Database
Warna	Biru, Hijau, Hitam, Kuning, Putih, Ungu, Merah, Jingga, Coklat
Angka	Satu, Dua, Tiga, Empat, Lima, Enam, Tujuh, Delapan, Sembilan
Hewan	Anjing, Babi, Buaya, Kelinci, Kerbau, Kucing, Monyet, Kura-kura, Kambing
Huruf	A, B, C, D, E, F, G, H, I

Tabel 3.4 menerangkan materi yang ada pada aplikasi dan objek sebagai database yang disediakan pada aplikasi pembelajaran pengucapan kata pada penderita *downsyndrome.*

3.2 Analisis Pengolahan Sinyal Suara

Proses pengenalan kata dan simulasi pada sistem harus melalui beberapa tahap proses agar dapat dikenali oleh sistem. Suara yang telah direkam berupa sinyal analog diolah menjadi sinyal digital untuk diubah menjadi data diskrit . data suara hasil rekaman kembali diproses dalam pengolahan sinyal suara, diekstraksi ciri fiturnya untuk mengetahui perbedaan ciri-ciri tiap suara agar memudahkan saat proses pengenalan kata. Setelah itu barulah sistem pengenalan kata akan mengenali ucapan dari pengguna setelah sistem memiliki *template* fitur yang akan di jadikan pembanding.

Dalam analisis pengolahan sinyal suara dibagi menjadi 3 tahap, tahap pertama adalah tahap pre-processing, tahap kedua adalah tahap memproses sinyal berdasarkan frekuensi dengan FFT, lalu data hasil dari proses FFT diambil untuk dijadikan hash. Secara umum peroses pengolahan sinyal suara pada aplikasi mengikuti alur seperti pada gambar 3.9



Gambar 3.9 Alur Pengolahan Sinyal Suara

3.2.1 Analisis *Pre-Processing*

Sinyal suara yang akan diproses bersifat analog sehingga jika dilakukan pengolahan secara digital, sinyal suara itu 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)



Gambar 3.10 Konsep Kerja ADC

1. Sampling

Proses sampling adalah proses konversi dari bentuk gelombang yang masih kontinyu menjadi bentuk gelombang diskrit. Sedangkan proses rekontruksi adalah proses konversi kembali dari gelombang diskrit menjadi gelombang yang kontinyu. Tujuan dari proses sampling adalah mempermudah pemrosesan sinyal analog pada sistem pemrosesan yang diskrit (biasanya melibatkan ADC dan DAC) dan menghemat bandwidth tranmisi sinyal analog, misalnya PCM atau voice.

Data berupa sinyal suara diperoleh dengan cara merekam suara melalui *microphone* yang dihubungkan dengan komputer. Perekaman suara didalam aplikasi menggunakan bantuan *function* dari program dan *accessories windows* yaitu *sound recorder* dengan format .wav dan ferekunsi sampling standar terbaik pada PC adalah 8000Hz, 11025Hz, 22050Hz, dan 44100Hz. Suara dengan format .wav ini bisa menggunakan 8 atau 16 bits/sample dan 1 untuk *channel mono* atau 2 untuk *chanel streo.* Perhitungan pada proses akusisi data (proses mengambil, mengumpulkan dan menyiapkan data, hingga menghasilkan data yang dikehendaki) dilakukan untuk pengambilan sampling, durasi suara yang direkam apabila lebih pendek lebih mudah untuk diambil perbedaan fiturnya.

$$X = FS.dt.\left(\frac{bit}{8}\right) . j$$

Dimana:

X = Jumlah data sampel

FS = Frekuensi Sampling (Hz)

dt = Durasi (detik)

bit = Jumlah bit tiap sampel

j = Channel (mono=1 dan stereo = 2)

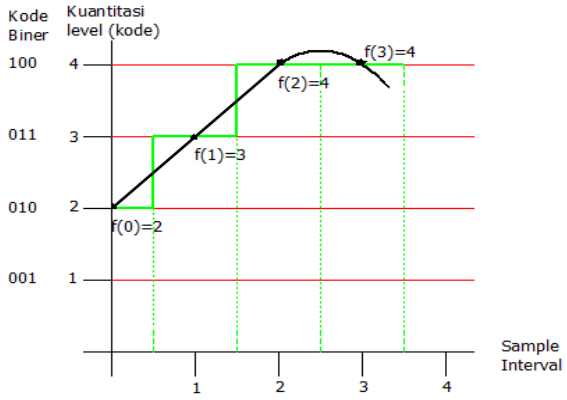
Dalam analisis ini digunakan contoh durasi rekaman yang diambil adalah 0,4 detik.

Tabel 3.5 Implementasi Aplikasi Model

Frekuensi (Hz)	Bits/ sample	Channel	Perhitungan	Jumlah Data Sample (byte)
8000	8	Mono	8000*0,4*(8/8)*1	3200
11025	8	Mono	11025*0,4*(8/8)*1	4410
22050	8	Mono	22050*0,4*(8/8)*1	8820
44100	8	Mono	44100*0,4*(8/8)*1	17640

2. Kuantisasi

Proses ini adalah proses pengkonversian nilai analog kedalam suatu nilai diskrit. Selama proses kuantisasi, ADC mengkonversi setiap nilai analog ke dalam bentuk diskrit. Diambil suatu input sinyal analog yang akan di kuantisasi, terlihat pada gambar 3.10



Gambar 3.10 Input Sinyal Analog Yang Di Kuantisasi

Dari hasil kuantisasi pada gambar 3.9 maka dapat diambil data suara hasil sinyal diskrit dengan diawali sebagai nilai *f(x)* sebagai berikut:

2	3	4	4
f(0)	f(1)	f(2)	f(3)

3. Coding

Pada proses coding, tiap nilai diskrit yang telah didapat selanjutnya dipresentasikan dalam angka biner n-bit. Cara konversinya adalah dengan membagi bilangan desimal dengan bilangan biner dengan memperhatikan sisa pembagian.

Tabel 3.6 Perhitungan Konversi Sinyal Analog

Nilai Diskrit dari Hasil Kuantisasi	Kode Biner (8-bit)
2	00000010
3	00000011
4	00000100
4	00000100

Kode biner hasil *coding* tidak akan dimasukan dalam perhitungan proses selanjutnya, karena kode biner ini hanya dibaca oleh *processor* komputer untuk pembacaan suatu sinyal digital.

3.2.2 Analisis Proses Fast Fourier Transform

Fast Fourier Transform adalah suatu algoritma komputasi optimal yang mengimplemetasikan *Discrete Fourier Transform* (DFT) dengan teknik perhitungan yang cepat serta memanfaatkan sifat periodikal dari transformasi fourier. FFT merupakan operasi matematika yang bertujuan untuk dekomposisi dari suatu sinyal domain waktu ke sinyal domain frekuensi.

DFT dilakukan dengan mengimplementasikan sebuah transformasi denagan panjang vektor N berdasarkan rumus:

$$F(u) = \frac{1}{N}\sum_{x=0}^{x=N-1} f(x) exp[-2\pi ux/N]$$

$$F(u) = \frac{1}{N}\sum_{x=0}^{x=N-1} f(x)(\cos(\frac{2\pi ux}{N}) - j\sin(\frac{2\pi ux}{N}))$$

Misal diambil data suara hasil kuantitasi sinyal diskrit dengan nilai f(x) sebagai berikut:

2	3	4	4
f(0)	f(1)	f(2)	f(3)

Nilai diskrit f(x) sebanyak 4 data, sehingga dapat ditentukan nilai N = 4 (banyak data), perhitungannya adalah:

Tabel 3.7 Contoh Perhitungan DFT Sinyal Suara

N	$f(x)$	Perhitungan DFT $\sum_{x=0}^{N-1}$	Hasil dari DFT / $F(u)$
1	$f(0)$	1/4[2(cos(2*pi*0*0/4) – j sin(2*pi*0*0/4) + 3(cos(2*pi*0*1/4) – j sin(2*pi*0*1/4) + 4(cos(2*pi*0*2/4) – j sin(2*pi*0*2/4) + 4(cos(2*pi*0*3/4) – j sin(2*pi*0*3/4)]	3.25
		1/4 [2(1-0) + 3(1-0) + 4(1-0) + (1-0)]	
2	$f(1)$	1/4[2(1-0) + 3(0- *j*) + 4(-1) + 4 (*j*)]	-0.5 + 0.25*j*
3	$f(2)$	1/4 [2(1-0) + 3(1-2) + 4(1-0) + 4(1-2)]	-0.25
4	$f(3)$	1/4[2(1-0) + 3(*j*) + 4(-1) + 4 (0-*j*)]	-0.5 - 0.25*j*

Perhitungan FFT mengimplementasikan pencerminan transformasi ganda hasil DFT dengan hanya menghitung nilai setengahnya data sinyal sehingga perhitungan akan lebih cepat, lalu nilai setengahnya lagi dihitung dengan cara *conjugate* nilai yang telah dihitung dengan DFT. Untuk membagi data sinyal adalah dengan fungsi (3.4) :

Tabel 3.8 Perhitungan FFT dari hasil DFT

$f(x)$ hasil	Hasil Perhitungan FFT / $F(u)$	
$f(0)$	3.25	
$f(1)$	-0.5 + 0.25	
$f(2)$	-0.25	← nilai tengah (b)
$f(3)$	-0.5 – 0.25j	← hasil *conjugate* $f(1)$
$f(4)$	3.25	← hasil *conjugate* $f(0)$

A
lur
mas
uka
n
suar
a

dimulai dari sinyal masuk, diproseses dengan rumus DFT, lalu setengah data hasil DFT di konjugasi agar menghasilkan nilai FFT.

```
Procedure FFT(u: N:integer, a,b,fx:double, o: c:double)
{ Menghitung nilai Fast Fourier Transform
  IS : Banyaknya elemen data n dan sinyal suara
  FS : c[u] berupa elemen hasil FFT
}

Deklarasi
u, N    : integer
x, j , y : double

Algoritma
for u←0 to N-1 do
      F(u)  ←0
          for x←0 to N-1 do
          a←a+(fx*cos((2*3.14*u*x)/N))
            b←b+(fx*sin((2*3.14*u*x)/N))
     endfor x
  F(u) ←a+b
endfor u

for u← (0+N-1)div2 do
        F[u] ←0
      for x←0 to N-1 do
         a←a+(fx*cos((2*3.14*u*x)/N)
         b←b+(fx*sin((2*3.14*u*x)/N)
       endfor x
    F[u] a+b
endfor u

for u← (0+N)div2+1 to N-1 do
      j←u-((0+N-1)div2)
     y((0+N-1)div2)-j
c[u] ←c[y]
endfor u
```

Gambar 3.11 Algoritma procedure FFT

3.2.3 Analisis Ekstraksi Hash

File suara yang sebelumnya direkam berformat .wav dan selanjutnya diproses kedalam suatu proses sampling dan proses FFT sehingga dapat dilakukan ekstraksi

hash, yaitu sebuah proses yang mengkonversi sinyal suara menjadi beberapa parameter yang dapat diambil untuk proses selanjutnya yaitu identifikasi pola suara.

$$b = (N + 1)div2 \qquad (3.4)$$

Berdasarkan fungsi (3.4) maka didapatkan $b = (4 + 1)div2 = 3$, jadi perhitungan DFT hanya sampai data ke-3 / *f(2)* dan untuk data selanjunya hanya memakai fungsi *conjugate* dari nilai hasil DFT.

Parameter yang memungkinkan dari suatu file suara adalah nilai amplitudo. Karena nilai amplitudo didapatkan berdasarkan domain waktu sehingga untuk pengambilan sampel data akan didapatkan waktu yang mungkin berbeda-beda. Hal inilah yang menyebabkan ekstraksi hash tidak konsisten untuk dijadikan suatu acuan atau masukan pada proses selanjutnya. Untuk itulah parameter dari suatu file suara tersebut harus diubah terlebih dahulu kedalam domain frekuensi dengan menggunakan algoritma *Fast Fourier Transform* karena pada manusia memiliki batas frekuensi tertentu sehingga data sampel bisa lebih konsisten untuk dijadikan suatu masukan pada proses selanjutnya. Proses inilah yang merupakan bagian dari pengolahan sinyal yang akan menghasilkan suatu ekstraksi hash berupa nilai magnitudo terhadap domain frekuensi. Untuk membedakan lagu yang satu dengan yang lain perlu ditentukan beberapa titik-titik kunci dalam lagu. Untuk setiap baris dalam analisis spektrum, aplikasi mengambil titik dengan besarnya magnitudo dari rentang tertentu.

Tabel 3.9 Rentang Frekuensi Magnitudo Sinyal Suara

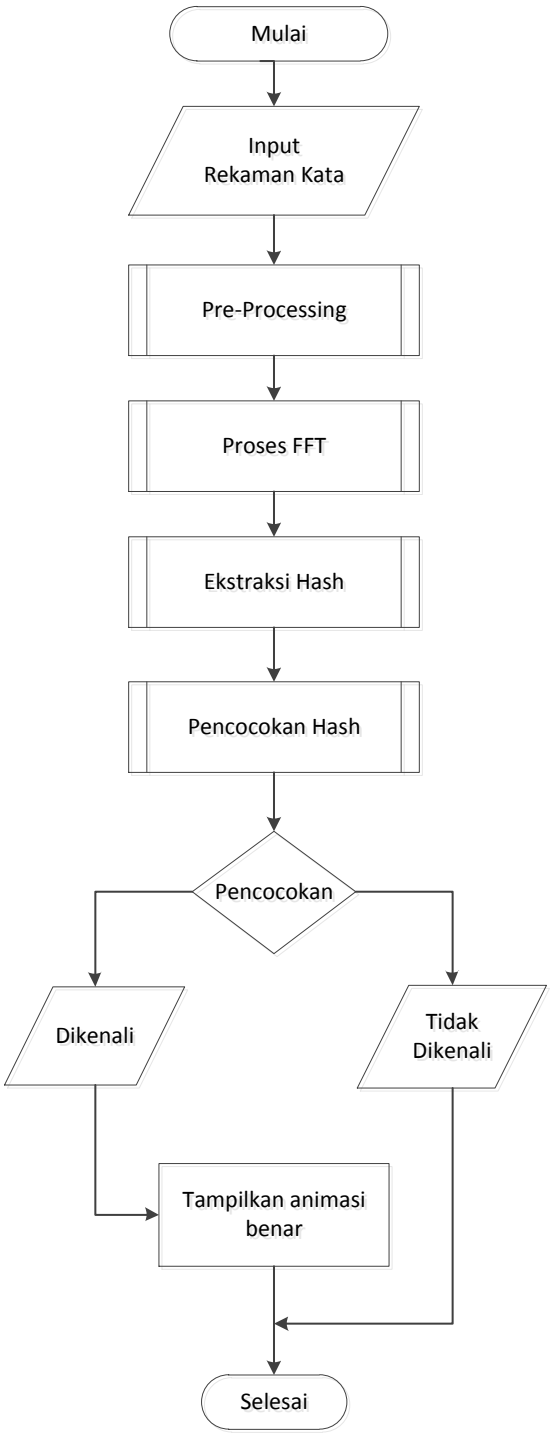
Frekuensi (Hz)	Oktav	Keterangan
16-32	1	Ambang pendengaran manusia, dan nada terendah dari alat musik
32-512	2-5	Frekuensi Rhythm, di mana nada bass bawah dan atas .
512-2048	6-7	Mendefinisikan kejelasan ucapan manusia, memberikan kualitas seperti sirine atau nyaring terdengar.
2048-8192	8-9	Mendefenisikan sebuah pidato.

8192-16384	10	Suara lonceng dan bunyi simbal.

3.2.4 Analisis Pengenalan Kata

Pengenalan kata merupakan proses membandingkan hasil ekstraksi hash yang didapat dari rekaman dengan hash yang ada didalam file indeks (basis data). Teknik yang dipakai dalam membandingkan hash yaitu dengan mengurutkan hash dari terkecil hingga terbesar, kemudian menghitung jumlah nilai skor hash tiap id kata yang cocok dengan hash di file indeks. Id lagu yang memiliki jumlah nilai skor yang lebih dari 5 dan terbanyak dari id kata lain merupakan id kata yang berhasil dikenal oleh aplikasi.

3.2.5 Analisis Alur Pengenalan Kata



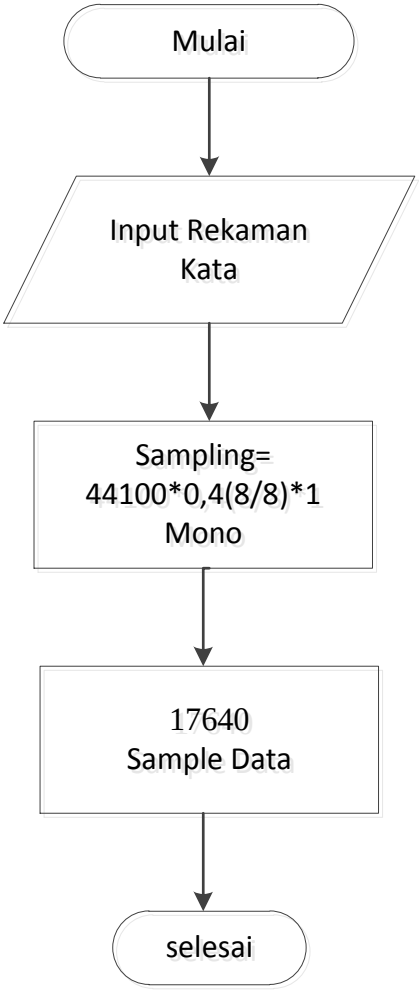
Gambar 3.12 Alur Umum Sistem

Alur umum sistem pada gambar 3.12 :

1. Suara diinputkan kemudian setelah suara dinputkan kemudian akan dilakukan proses
2. *pre-processing* proses ini di gunakan untuk mengubah sinyal suara digital ke analog,
3. kemudaian setelah suara diubah kedigital tahapan selanjutnya adalah FFT(Fast Fourier Transform) bertujuan untuk dekomposisi dari suatu sinyal domain waktu ke sinyal domain frekuensi
4. kemudian dilakukan ekstraksi hash
5. setelah itu dilakukan pencocokan dari kemungkinan pengucapan kata yang benar
6. setelah dilakukan pencocokan bila suara di kenali maka akan muncul tampilan animasi benar dan jika salah harus mengulangi lagi dan mencoba sampai benar.

3.2.5.1 Flochart Sampling Suara

Data berupa sinyal suara diperoleh dengan cara merekam melalui *microphone* dengan frekuensi sampling 44100 Hz, 8 bit dan *channel* mono. *Flowchart* proses sampling suara dapat dilihat pada Gambar 3.13.



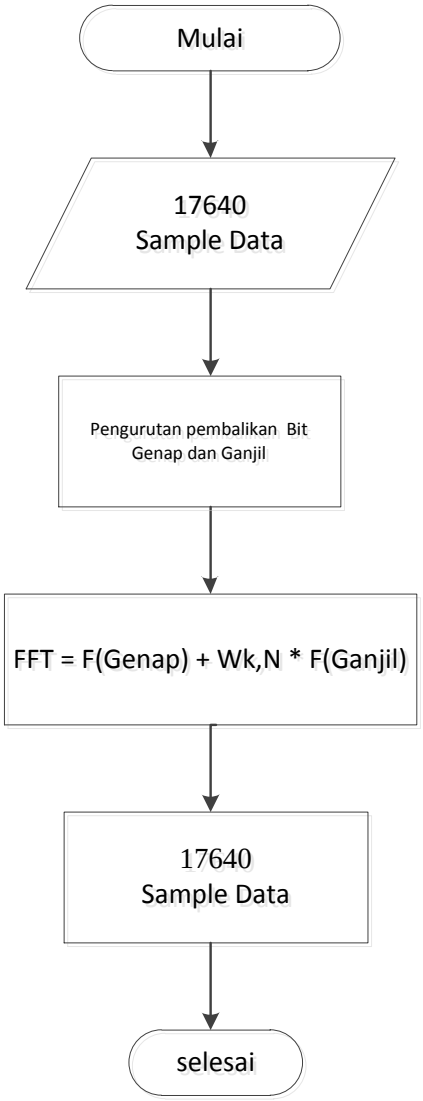
Gambar 3.13 Proses Sampiling Suara

Proses sampling suara pada gambar 3.13

1. Setelah kata diinputkan maka aplikasi akan menyimpan sampel suara dalam bentuk xml
2. Rumus sampel suara X = FS.*dt*.($\frac{bit}{8}$) . j (lebih jelas lihat table 3.5)
3. Di dapat total 17640 sample data per kata

3.2.5.2 Analisis Pengolahan Sinyal Suara

Pengolahan sinyal suara menggunakan algoritma FFT dapat dilihat pada



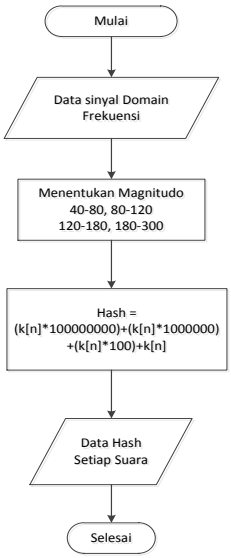
Gambar 3.14 Proses Pengolahan Sinyal Suara

Proses pengolahan sinyal suara pada gambar 3.14

1. Setelah didapat sampling suara yang totalnya 17640 per kata kemudian dilakukan

2. Pengurutan pembalikan ke genap dan ganjil untuk menampilkan data seperti masukan
3. Kemudian data diolah untuk dikembalikan kenilai awal.

3.2.5.3 Analisis Ekstraksi Hash



Gambar 3.15 Ekstrasi Hash

Proses Ekstraksi hash pada gambar 3.15

1. Setelah dilakukan pengolahan suara akan didapat data sinyal domain frekuensi
2. Dengan menentukan mangnitudo dengan menentukan rentang frekuensi mangnitudo, rentang frekuensi mangnitudo pendengaran manusia 40-80, 80-120, 120-180, 180-300, (untuk lebih jelas pada tabel 3.9)
3. Setelah ditentukan titik kunci dari magnitudo, pembuatan hash dilakukan dengan membalikkan urutan dari belakang ke depan dan menggabungkannya. Namun, pada urutan ke-dua terdapat perbedaan jumlah digit, yaitu dua digit dan tiga digit. Hal ini memerlukan teknik yaitu mengalikan tiap rentang sesuai satuan dan menjumlahkan keseluruhan rentang. Berikut rumus dalam pembuatan hash:

   hash=(K[n]*100000000)+(K[n]*100000)+ (K[n]*100)+ K[N]

   dimana:

k = Rentang magnitudo ke-n

n = Urutan 3, 2, 1, 0

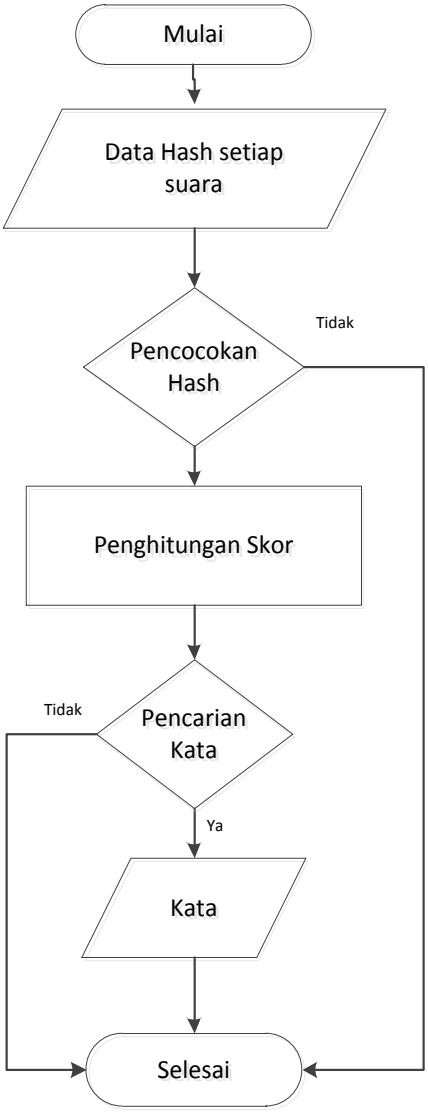
Sebagai contoh didapat titik k = {47, 94, 137, 193}, maka hashnya:

Hash = (193 * 100000000) + (137 * 100000) + (94 * 100) + 47

= 19300000000 + 13700000 + 9400 + 47

= 19313709447

3.2.5.4 Pengenalan Kata



Gambar 3.16 Pengenalan Kata

3.3 Analisis algoritma Fisher-Yates Shuffle

Algoritma Fisher-Yates Shuffle akan diimplementasikan pada aplikasi pembelajaran pengucapan kata pada penderita *downsyndrome* untuk mengacak urutan kata yang akan ditampilkan. Permutasi yang dihasilkan oleh algoritma ini muncul

dengan probabilitas yang sama. Metode dasar yang digunakan untuk menghasilkan permutasi acak dari angka 1 – *N* adalah sebagai berikut :

1. Tuliskan angka dari 1 sampai n (jumlah banyak objek)
2. Isi nilai j dengan bilangan acak lalu kalikan (i+1) lalu bulatkan kebawah
3. Lalu isi temporary dengan nilai array ke-j
4. Lalu isi array ke-j dengan nilai array ke-i
5. Lalu isi array ke-i dengan nilai temporary

Jika diimplementasikan pada aplikasi pembelajaran pengucapan kata yang akan dibangun, maka berikut *Pseudocode* dengan menggunakan bahasa algoritmik :

```
PROGRAM MengacakUrutanBinatang
{ Program untuk mengacak urutan benda yang akan ditampilkan}
DEKLARASI
i, j, temp : integer
arr : array[1..8] of integer {jumlah banyak objek}
ALGORITMA :
i : arr.length {menghitung banyak bilangan di array}
while --i > 0 do
   j ← floor(random * (i+1))
   temp ← arr[j]
   arr[j] ← arr[i]
   arr[i] ← temp
endwhile
```

Gambar 3.17 *Pseudocode* Algoritma Fisher Yates

Gambar 3.17 adalah *Pseudocode* algoritma Fisher Yates yang digunakan untuk mengacak urutan objek. Pada tabel 3.10 dapat dilihat contoh urutan objek yang masih tersusun dari 1-8 yang mana dibutuhkan untuk mengacak urutan objek yang masih tersusun.

Tabel 3.10 Posisi Awal Warna

I	Warna
1	Biru
2	Hijau

3	Hitam
4	Kuning
5	Putih
6	Ungu
7	Merah
8	Jingga

Data yang berada di tabel 3.10 merupakan data yang akan diolah dengan menggunakan algoritma Fisher Yates. Pada Tabel 3.11 merupakan hasil dari pengolahan dengan menggunakan algoritma tersebut.

Tabel 3.11 Posisi Hasil Pengolahan

I	Warna
2	Biru
8	Hijau
4	Hitam
6	Kuning
3	Putih
5	Ungu
7	Merah
1	Jingga

Berdasarkan analisis diatas dipastikan dengan menggunakan algoritma tersebut semua objek pasti muncul tanpa adanya yang terlewat.

3.4 Analisis Kebutuhan Non Fungsional

Analisis non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan *system*. Spesifikasi ini meliputi komponen-komponen yang dibutuhkan untuk membangun *system* dan sampai *system* tersebut diimplementasikan. Analisis kebutuhan non fungsional menentukan spesifikasi *input* yang diperlukan *system*, memproses inputan oleh *system* dan menghasilkan *output* yang telah direncanakan sesuai dengan spesifikasi.Analisis non-fungsional menjelaskan analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras dan analisis pengguna.

3.3.1 Analisis Perangkat Keras

Perangkat keras merupakan aspek penting dalam membangun aplikasi dan implementasi aplikasi, perangkat keras yang memenuhi syarat minimum penggunaan akan memenuhi standar kebutuhan aplikasi, oleh karena itu dibutuhkan perangkat keras yang sesuai dengan kebutuhan aplikasi.

Pada tabel 3.12 terdapat perangkat keras yang dibutuhkan *developer* untuk membangun aplikasi pembelajaran pengucapan kata.

Tabel 3.12 Spesifikasi Perangkat Keras (*developer*)

Nama Perangkat	Spesifikasi
Processor	Dual-core 2.66-GHz
Memory	2 GB
HDD	40GB
VGA	1GB
Mouse	1 Unit
Keyboard	1 Unit
Earphone	1 Unit

Pada tabel 3.13 terdapat perangkat keras yang dibutuhkan user untuk menggunakan aplikasi pembelajaran pengucapan kata.

Tabel 3.13 Spesifikasi Perangkat Keras (*user*)

Nama Perangkat	Spesifikasi
Processor	Dual-core 2.66-GHz
Memory	2 GB
HDD	40GB
VGA	1GB
Mouse	1 Unit
Earphone	1 Unit
USB	USB 2.0

3.3.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak merupakan aspek penting dalam mendukung kinerja sistem yang dibangun. Perangkat lunak yang digunakan dalam sebuah sistem merupakan satu kesatuan dengan perangkat keras. Perangkat lunak bekerja dengan mengirimkan perintah kepada perangkat keras sehingga keduanya dapat saling terhubung. Pada tabel 3.14 terdapat perangkat keras yang dibutuhkan untuk membangun aplikasi pembelajaran pengucapan kata:

Tabel 3.14 Spesifikasi Perangkat Lunak (*developer*)

Jenis	Nama Perangkat Lunak
Sistem Operasi	Windows 7
Engine	Unity 4.6
Framework	.NET Framework 4.0
Framework Graphics	DirectX 11

Pada tabel 3.15 terdapat perangkat lunak yang dibutuhkan pengguna untuk menjalankan aplikasi ini adalah sebagai berikut :

Tabel 3.15 Spesifikasi Perangkat Lunak (*user*)

Jenis	Nama Perangkat Lunak
Sistem Operasi	Windows 7
Framework	.NET Framework 4.0
Framework Graphics	DirectX 11

3.3.3 Analisis Kebutuhan Pengguna

Target pengguna aplikasi yang akan dibangun adalah anak penderita downsyndrome 0% - 75% yaitu yang berada di Sekolah Khusus YPAC. Adapun spesifikasi secara umum pengguna mengerti dan memahami dalam mengoperasikan komputer. Pada analisis pengguna mencakup analisis beberapa parameter calon pengguna dari aplikasi, parameternya adalah sebagai berikut :

1. Pengetahuan dan Pengalaman Pengguna

Pengetahuan dan pengalaman pengguna dalam menggunakan sistem yang serupa merupakan aspek penting dalam pembangunan aplikasi yang akan dibangun. Pada tabel 3.16 terdapat klasifikasi *knowledge and experience from user*.

Tabel 3.16 Analisis Klasifikasi Knowledge and Experience

Nama Klasifikasi	Deskripsi
Educational level	Aplikasi yang akan dibangun ditunjukkan untuk anak penderita downsyndrome 0% - 75% yang berada di sekolah Khusus YPAC
Reading Level	Aplikasi yang akan dibangun ditunjukkan untuk pengguna yang masih mengenal huruf dan angka.
Typing Skills	Aplikasi yang akan dibangun tidak membutuhkan kemampuan mengetik.
Computer Literacy	Aplikasi yang akan dibangun ini ditunjukkan untuk pengguna yang sudah atau belum memiliki kemampuan dalam mengoperasikan komputer yang memadai.
Task Experience	Aplikasi yang akan dibangun ditunjukkan untuk pengguna yang pernah atau belum pernah menggunakan komputer
System Experience	Aplikasi yang akan dibangun membutuhkan pengalaman pengguna dalam mengoperasikan sistem komputer
Application Experience	Aplikasi yang akan dibangun ini ditujukkan untuk pengguna yang memiliki pengalaman dalam penggunaan aplikasi komputer.
Native Laguage	Aplikasi yang akan dibangun ini menggunakan bahasa indonesia.
Use Of Other System	Pengguna yang menggunakan aplikasi diharapkan dapat menghafalkan dan mengucapkan kata dengan baik dan benar dan pengguna lebih termotifasi lagi.

2. Karakter Fisik Pengguna

Keadaan fisik seseorang sangat menentukan keberhasilan interaksi dalam aplikasi pembelajaran ini. Pada tabel 3.17 terdapat karakteristik fisik pengguna yang menjadi acuan agar pengguna dapat maksimal dalam menggunakan aplikasi yang akan dibangun, karakteristiknya.

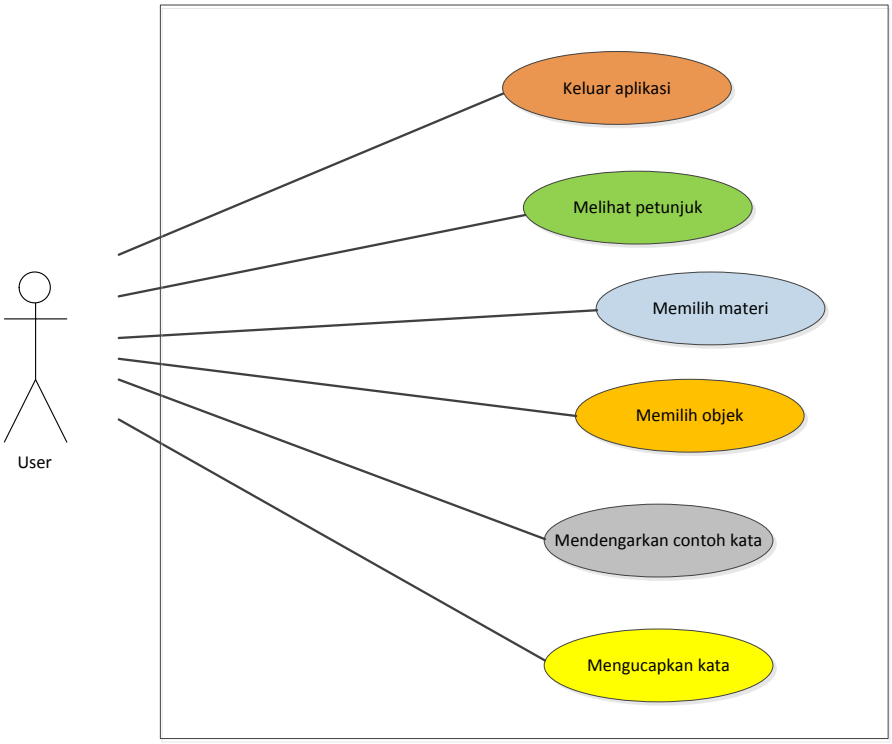
Tabel 3.17 Analisis Users Physical Characteristic

Age	Tidak ada batasan (penderita *downsyndrome*)
Gender	Tidak ada batasan jenis kelamin
Handedness	Bisa mendengar dan bicara
Color Blind	Tidak ada batasan pengguna dalam menggunakan sistem, namun pengguna yang memiliki kekurangan tersebut tidak dapat optimal dalam berinteraksi dengan aplikasi dikarenakan aplikasi ini membutuhkan interaksi antara mendengar dan mengucapkan.

3.5 Analisis Kebutuhan Fungsional

3.5.1 Use Case Diagram

Use Case diagram dari aplikasi pembelajaran pengucapan kata pada siswa memiliki 1 aktor dan 6 *use case. Use case* diagram yang digunakan pada aplikasi pembelajaran pengucapan kata adalah pada gambar 3.10 berikut :



Gambar 3.18 Use Case Diagram

3.5.1.1 Definisi Aktor

Definisi aktor yang ada dalam aplikasi pembelajaran pengucapan kata pada penderita *downsyndrome* dinamai dengan *user*, *user* merupakan orang yang menggunakan atau mengoprasikan aplikasi pembelajan ini.

3.5.1.2 Identifikasi *Use Case*

Identifikasi *Use case* merupakan deskripsi yang ada dalam setiap *use case* aplikasi pembelajaran pengucapan kata pada penderita *downsyndrome* seperti pada tabel 3.18

Tabel 3.18 Identifikasi Use Case

No	Nama *Use Case*	*Deskripsi*
1.	Memilih keluar aplikasi	Merupakan fungsionalitas untuk keluar dari aplikasi.

2.	Melihat petunjuk	Merupakan fungsionalitas untuk bantauan.
3.	Memilih materi	Merupakan fungsionalitas untuk memilih materi yang disediakan
4	Memilih objek	Merupakan fungsionalitas untuk memilih objek yang akan dipelajari
5.	Mendengarkan contoh kata	Merupakan fungsionalitas untuk mendengarkan contoh kata yang akan di ucapkan
6.	Mengucapkan kata	Merupakan fungsionalitas untuk merekam kata yang diucapkan murid dari objek yang dipilih.

3.5.1.3 Sekenario *Use Case*

Skenario *use case* berfungsi untuk menjelaskan lebih detail mengenai modul-modul yang terdapat dalam perangkat lunak. Berikut ini skenario *use case* yang terdapat pada aplikasi pembelajaran pengucapan kata pada penderita *downsyndrome.*

Tabel 3.19 Sekenario *Use Case* Keluar Aplikasi

Nama *Use case*	Keluar aplikasi
Aktor	User
Pre-condition	User berada dihalaman awal
Post-condition	Keluar dari aplikasi
Sekenario	
Aksi Aktor	**Aksi *System***
1. Memilih menu keluar	
	2. Keluar aplikasi.

Tabel 3.20 Sekenario *Use Case* Melihat Bantuan

Nama *Use case*	Melihat bantuan
Aktor	User
Pre-condition	User berada dihalaman objek
Post-condition	Menampilkan bantuan berupa fungsi-fungsi menu
Sekenario	
Aksi Aktor	**Aksi *System***
1. Memilih bantuan	
	2. Menampilkan menu bantuan

Tabel 3.21 Sekenario *Use Case* Memilih Materi

Nama *Use case*	Memilih materi
Aktor	User
Pre-condition	User berada dihalaman materi yang menampilkan pilihan seperti hewan, warna, huruf, dan angka.
Post-condition	Menampilkan pilihan objek
Sekenario	
Aksi Aktor	**Aksi *System***
1. Memilih modul	
	2. Menampilkan objek

Tabel 3.22 Sekenario Use Case Memilih Objek

Nama *Use case*	Memilih objek
Aktor	User
Pre-condition	User berada dihalaman objek
Post-condition	Menampilkan hasil
Sekenario	
Aksi Aktor	**Aksi *System***
1. Memilih objek	
	2. Menampilkan objek pilihan
Alternatif	
a. Murid dapat memilih objek satu persatu dengan memilih tombol kembali b. Murid dapat langsung pengucapkan kata jika jawaban benar	

Tabel 3.23 Sekenario *Use Case* Memilih Mendengar

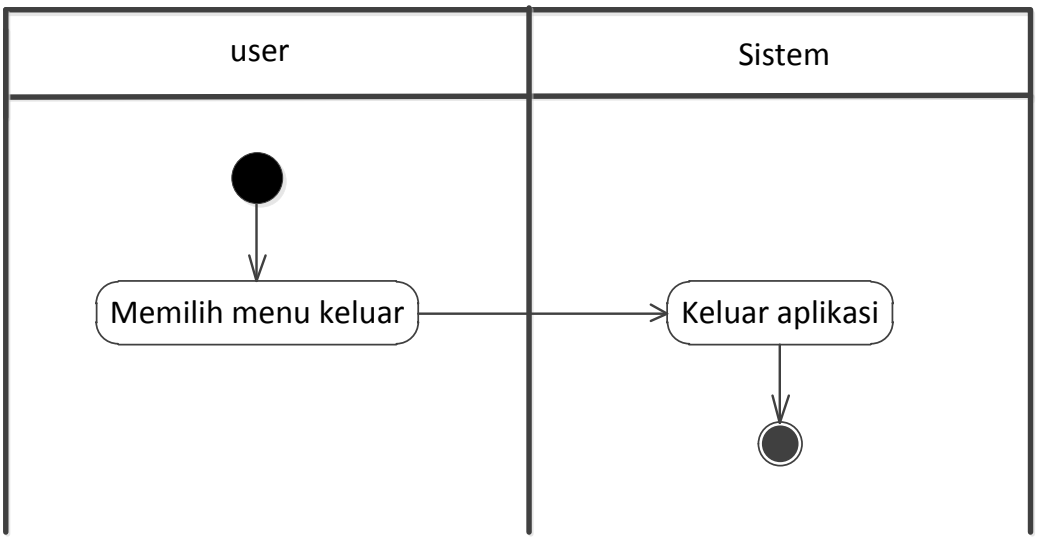
Nama *Use case*	Memilih mendengar
Aktor	User
Pre-condition	User berada dihalaman objek
Post-condition	Menampilkan contoh suara
Sekenario	
Aksi Aktor	**Aksi *System***
1. Memilih pilihan mendengar	
	2. Menampilkan contoh suara

Tabel 3.24 Sekenario *Use Case* Merekam Kata

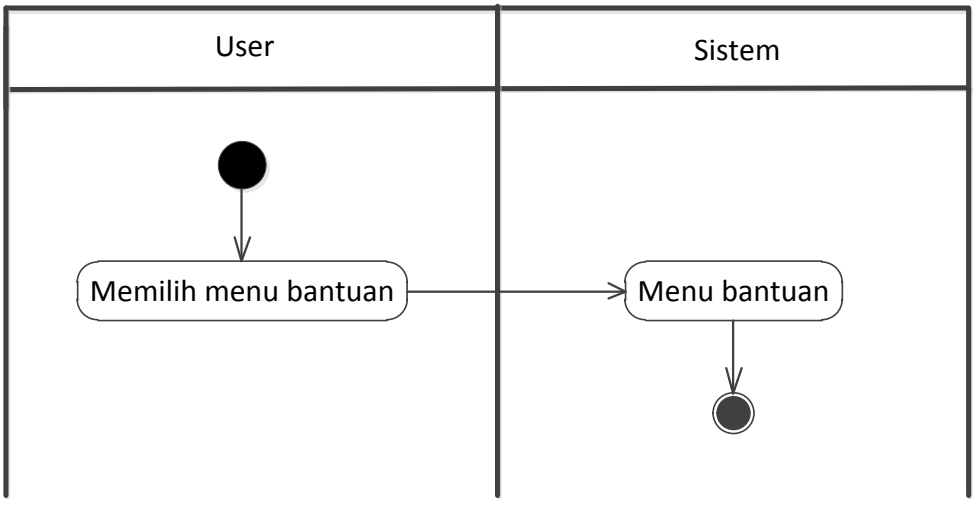
Nama *Use case*	Merekam kata
Aktor	User
Pre-condition	User berada dihalaman objek
Post-condition	Menampilkan hasil
Skenario	
Aksi Aktor	**Aksi *System***
	1. Menampilkan contoh suara
2. Mengucapkan kata	
	3. Membandingkan kata dengan *database*
	4. Menampilkan hasil
Alternatif	
a. Jika suara yang dimasukan sesuai maka akan tampil skor b. Jika salah akan muncul peringatan coba lagi	

3.5.2 *Activity* Diagram

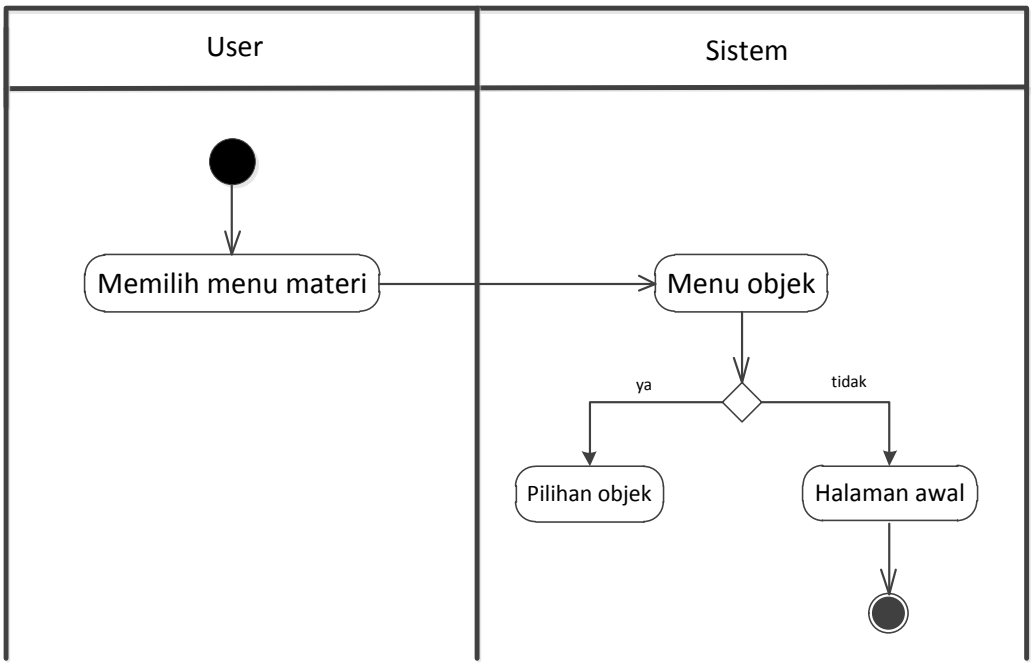
Activity diagam menggambarkan berbagai alir aktivitas dalam perangakat lunak yang sedang dirancang, bagaimana masing-masing alir berawal, pengambilan keputusan yang mungkin terjadi dan bagaimana proses tersebut berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa proses. Berikut *activity diagram* yang terdapat pada aplikasi pembelajaran pengucapan kata pada penderita *downsyndrome* :



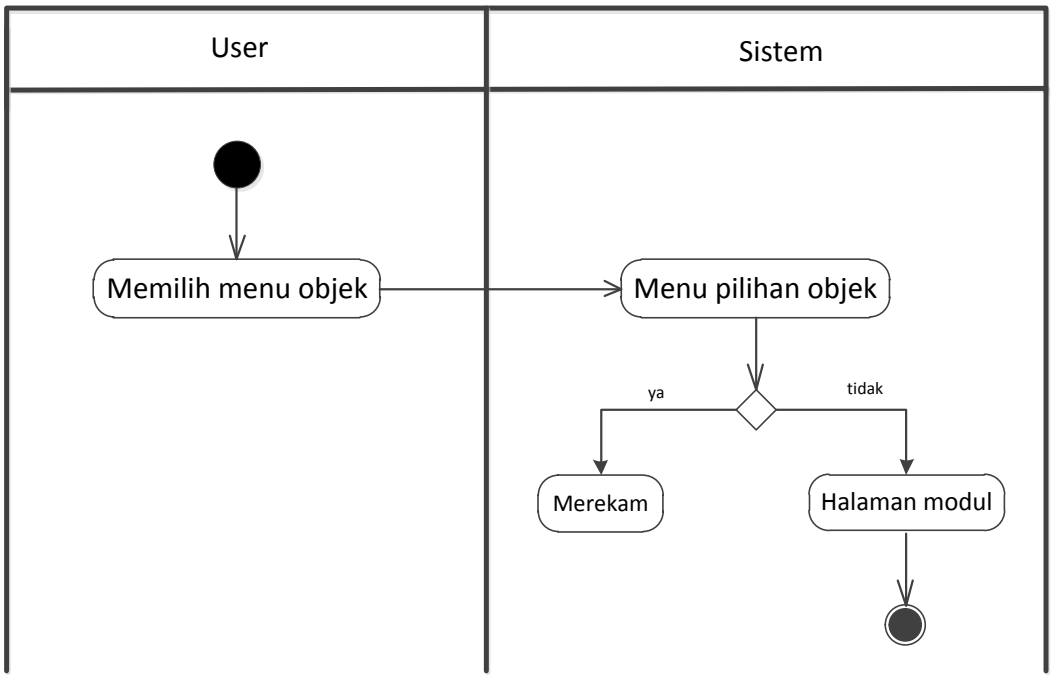
Gambar 3.19 *Activity Diagram* Keluar Aplikasi



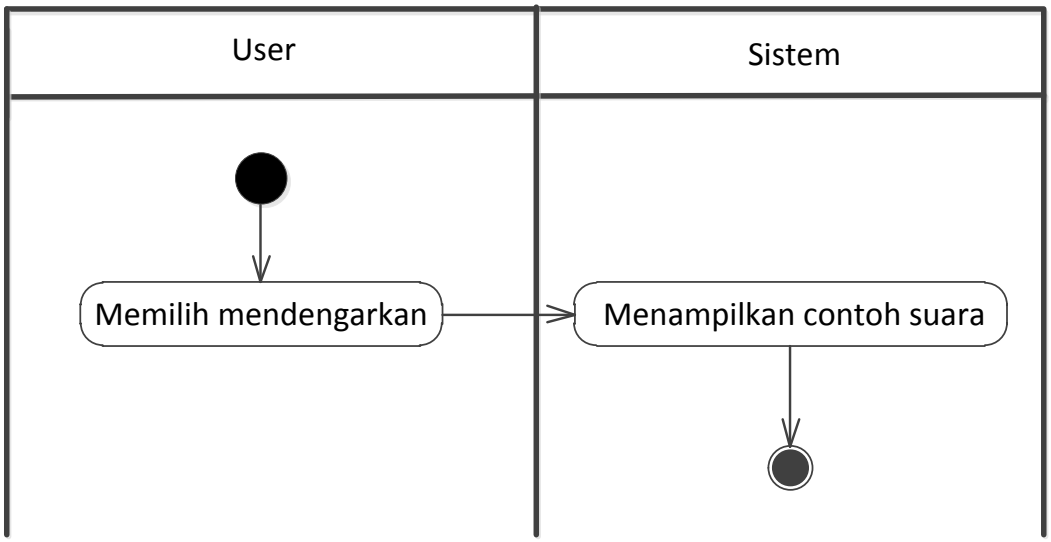
Gambar 3.20 *Activity Diagram* Bantuan



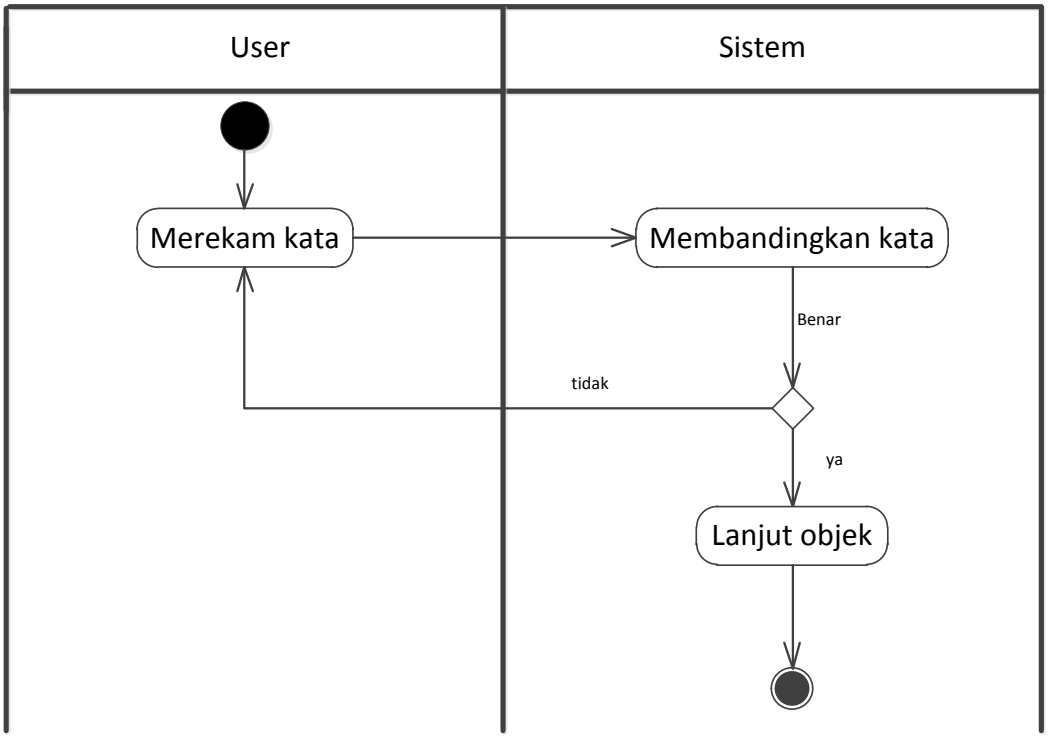
Gambar 3.21 *Activity Diagram* Memilih Materi



Gambar 3.22 *Activity Diagram* Memilih Menu Objek



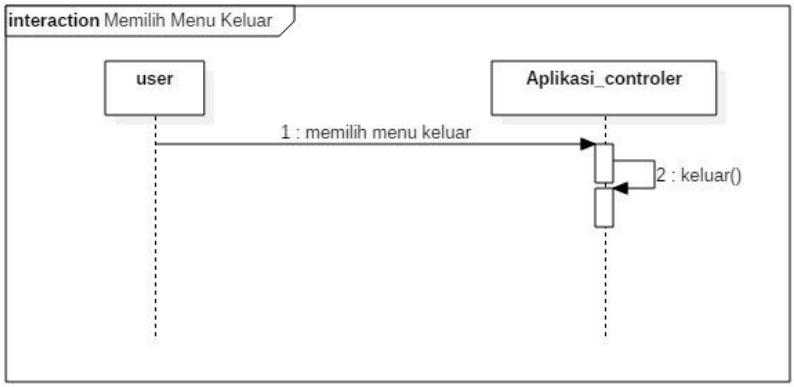
Gambar 3.23 *Activity Diagram* Mendengarkan Contoh Kata



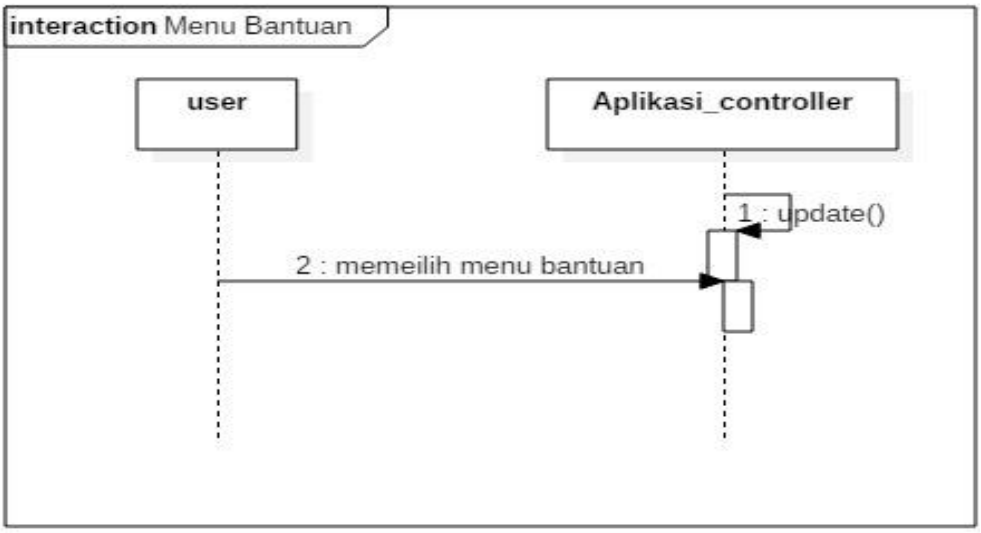
Gambar 3.24 *Activity Diagram* Merekam Kata

3.5.3 Sequence Diagram

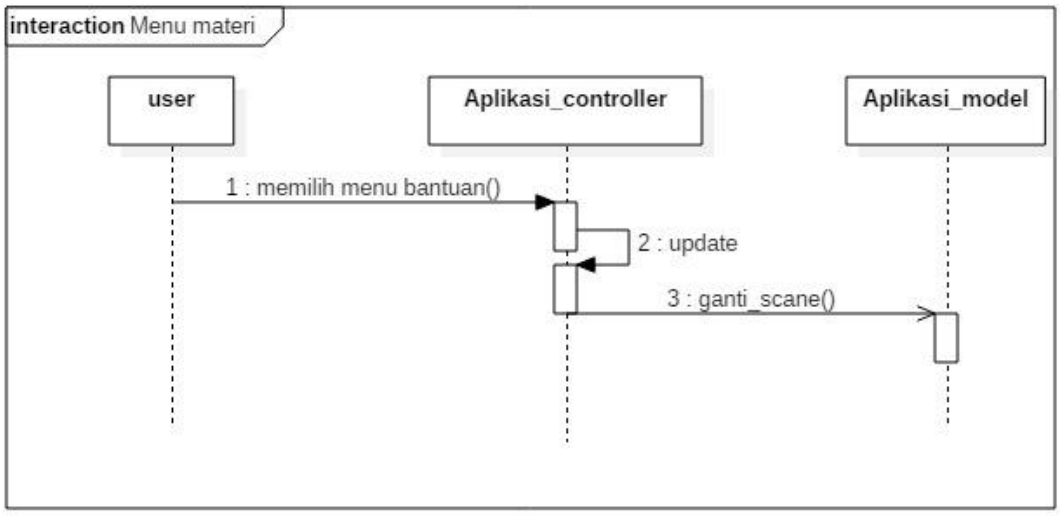
Sequence Diagram menggambarkan interaksi antar objek didalam dan disekitar perangkat lunak termasuk pengguna. Interaksi tersebut berupa pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram pada aplikasi pembelajaran pengucapan kata pada anak *downsyndome* terbagi menjadi beberapa bagian berikut diantaranya :



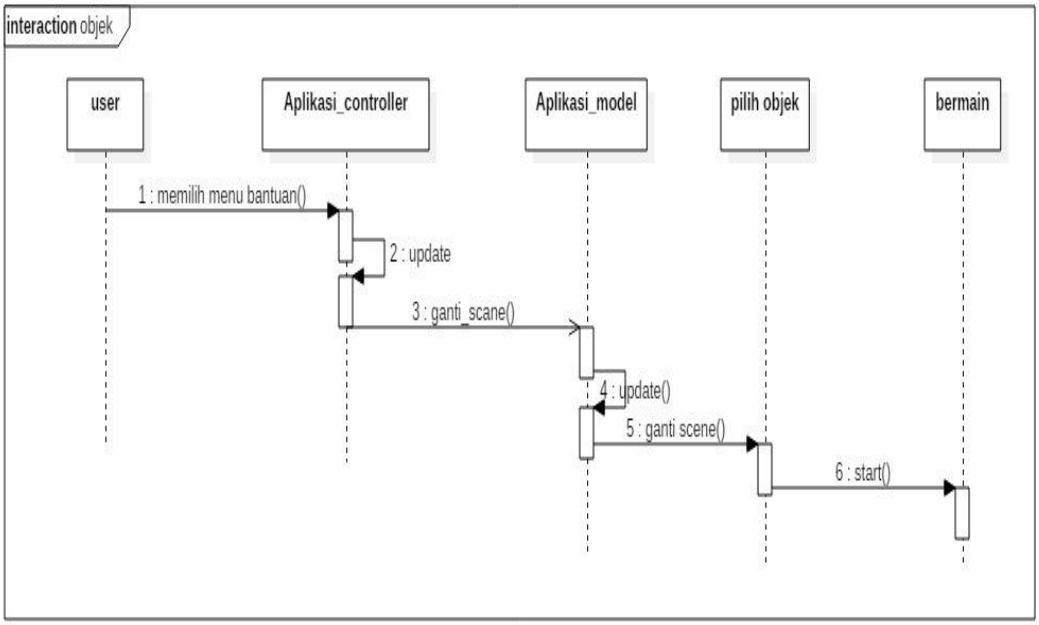
Gambar 3.25 Squence Diagram Memilih Menu Keluar



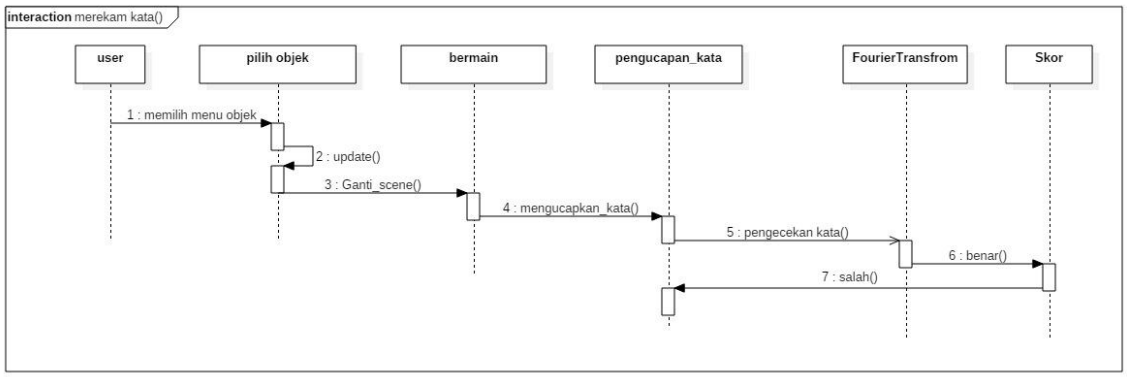
Gambar 3.26 Squence Diagram Memilih Menu Bantuan



Gambar 3.27 Squence Diagram Memilih Menu Materi



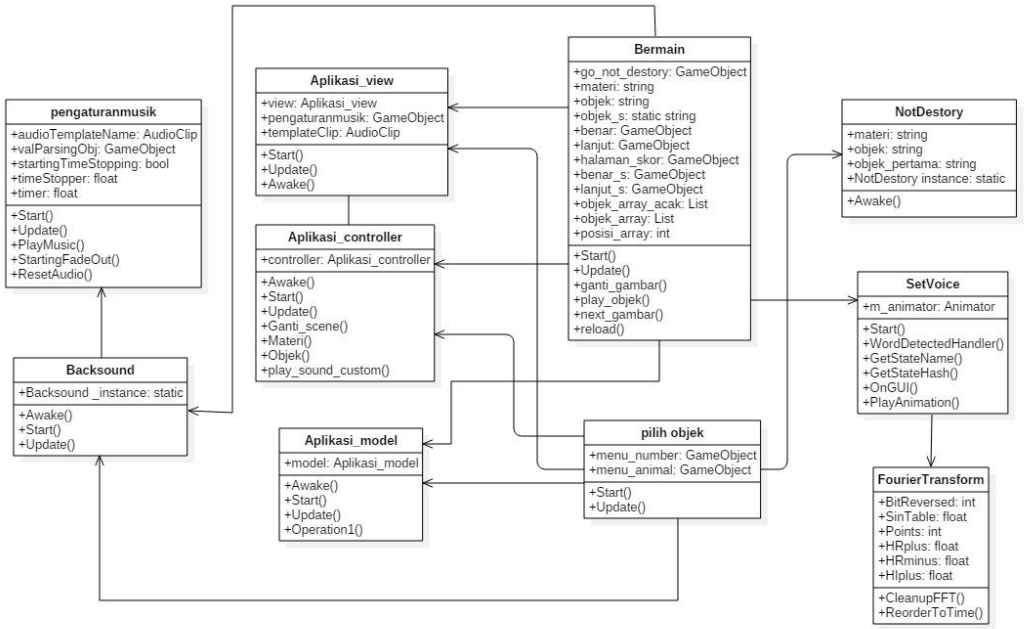
Gambar 3.28 Squence Diagram Memilih Menu Objek



Gambar 3.29 Squence Diagram Merekam Kata

3.5.4 Class Diagram

Class adalah spesifikasi yang jika diinsialisasi akan menjadi sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* meggambarkan keadaan (atribut/properti) suatu perangkat lunak, sekaligus dapat memanipulasi keadaan tersebut (metoda/fungsi). Berikut *class diagram* pada aplikasi pembelajaran pengucapan kata pada anak *downsyndom*, pada gambar 3. :



Gambar 3.30 *Class* Diagram

3.5.5 Perancangan Sistem

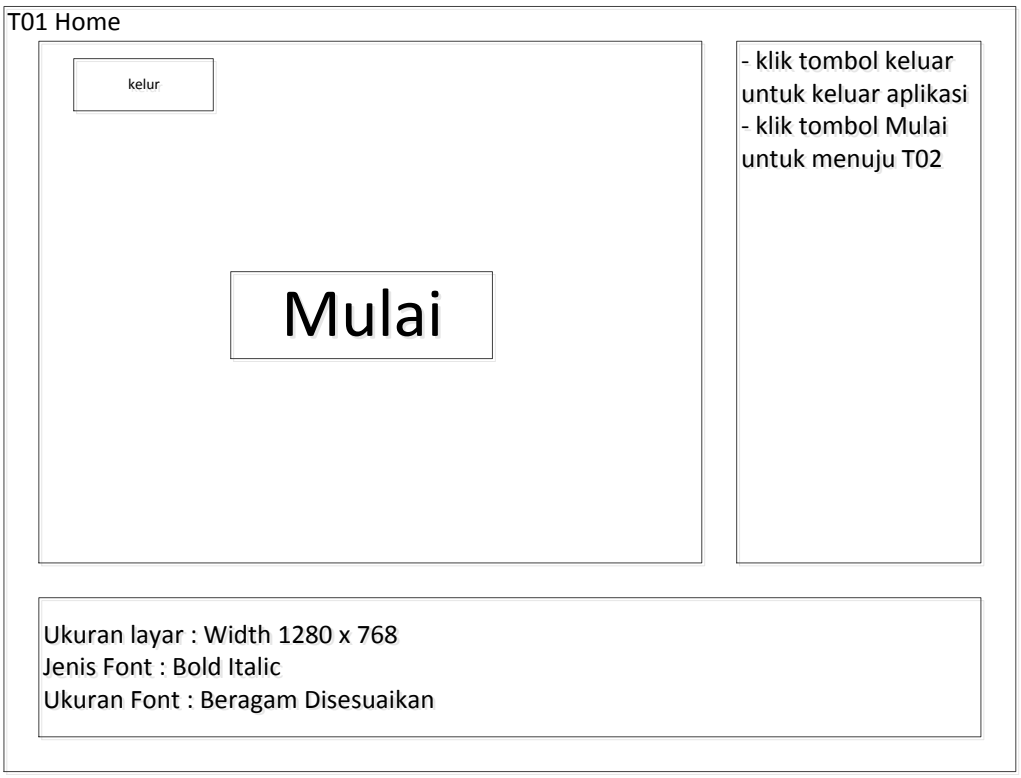
Perancangan adalah suatu bagian dari metodologi pengembangan suatu perangkat lunak yang dilakukan setelah tahapan *requirement gathering* untuk memberikan gambaran mengenai perancangan sistem secara terperinci.

Perancangan sistem merupakan tahapan dari siklus pengembangan sistem yang disesuaikan dengan kebutuhan-kebutuhan fungsional. Persiapan untuk rancang bangun implementasi yang menggambarkan bagaimana suatu sistem dibentuk, yang dapat berupa penggambaran, perancangan, dan pembuatan sketsa dari sistem yang akan dibangun.

3.4.5.1 Perancangan Tampilan

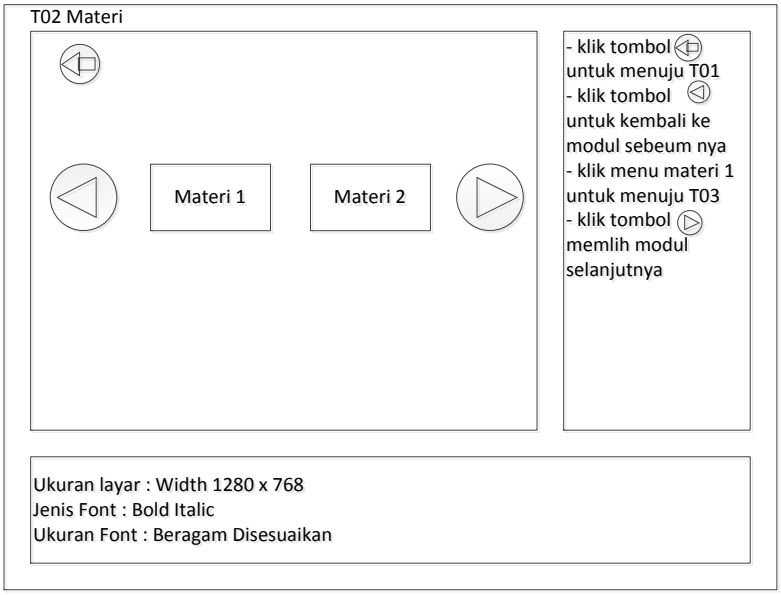
Berikut ini adalah tampilan yang terdapat dalam aplikasi pembelajaran pengucapan kata pada penderita downsyndrome

a. Halaman Awal



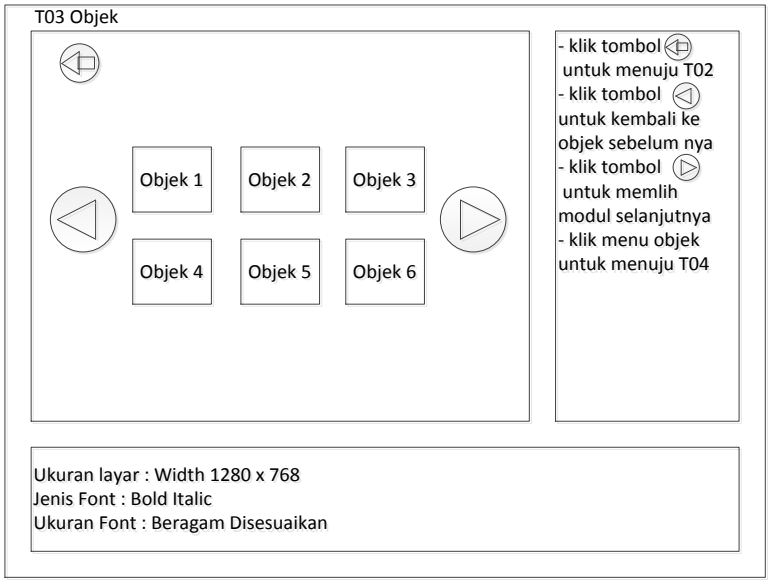
Gambar 3.31 Tampilan Menu Awal

b. Halaman Materi



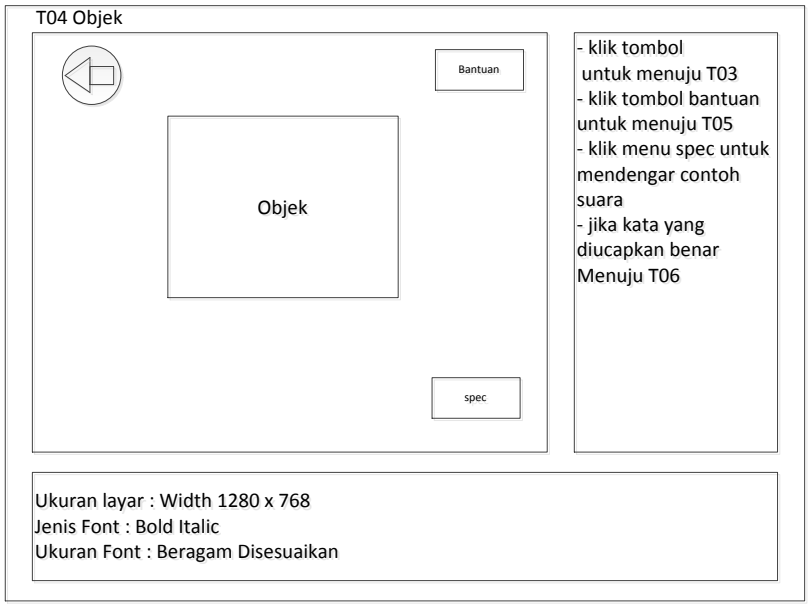
Gambar 3.32 Tampilan Halaman Materi

c. Halaman Pilihan Objek



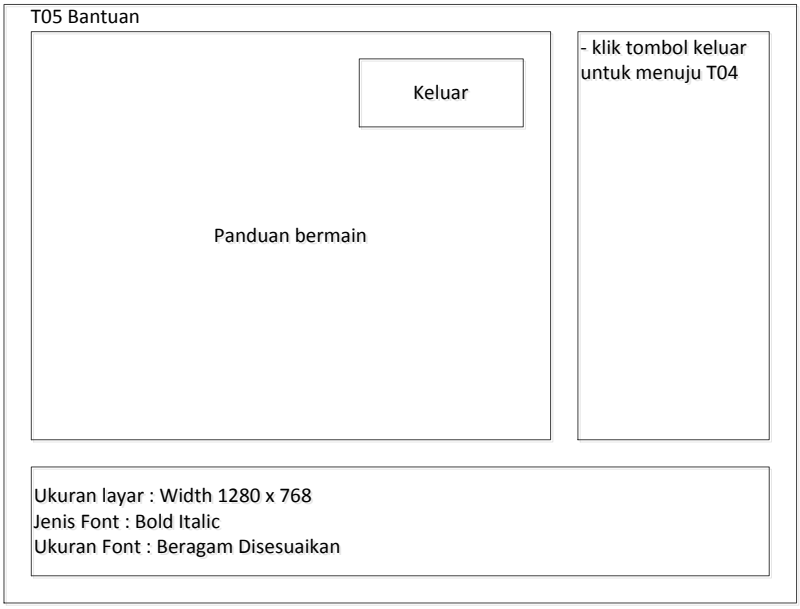
Gambar 3.33 Tampilan Halaman Pilihan Objek

d. Halman Objek



Gambar 3.34 Tampilan Halaman Objek

e. Halaman Bantuan



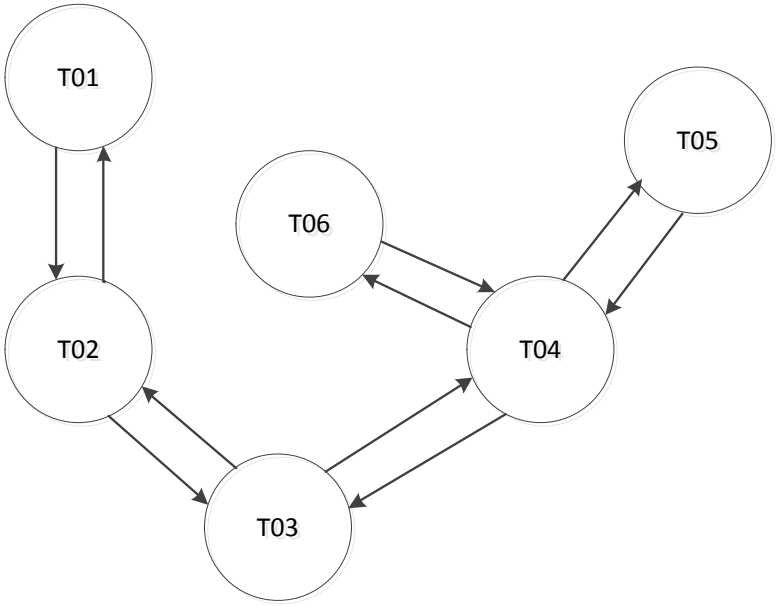
Gambar 3.35 Tampilan Halaman Bantuan

f. Halaman Skor



Gambar 3.36 Tampilan Halaman Skor

3.4.6 Jaringan Semantik



Gambar 3.37 Jaringan Semantik

Keterangan:

T01 : Perancangan antarmuka halaman awal

T02 : Perancangan antarmuka halaman materi

T03 : Perancangan antarmuka halaman pilihan objek

T04 : Perancangan antarmuka halaman objek

T05 : Perancangan antarmuka halaman bantuan

T06 : Perancangan antarmuka halam