BAB IV

IMPLEMENTASI DAN PENGUJIAN

IV.1 Implementasi Sistem

Pada tahap implementasi sistem, akan diuji program aplikasi yang telah dibuat setelah hasil analisis sistem dan algoritma yang telah diterapkan. Tujuan dari implementasi sistem ini adalah untuk menimplementasikan aplikasi dan mendapat masukan dari pengguna.

IV.1.2 Implementasi Perangkat Keras

Implementasi perangkat keras yang digunakan untuk menjalankan aplikasi ini adalah *smart phone android* dengan spesifikasi sebagai berikut:

Tabel IV.1 Implementasi perangkat keras yang digunakan

Perangkat Keras	Spesifikasi
Processor	Dual Core Cortex A9
RAM	512 GB
Memory	8 GB
Layar	320 x 480 pixels, 3,5 inchi

IV.1.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak yang digunakan untuk menjalankan aplikasi ini adalah sebagai berikut:

Tabel IV.2 Implementasi perangkat lunak yang digunakan

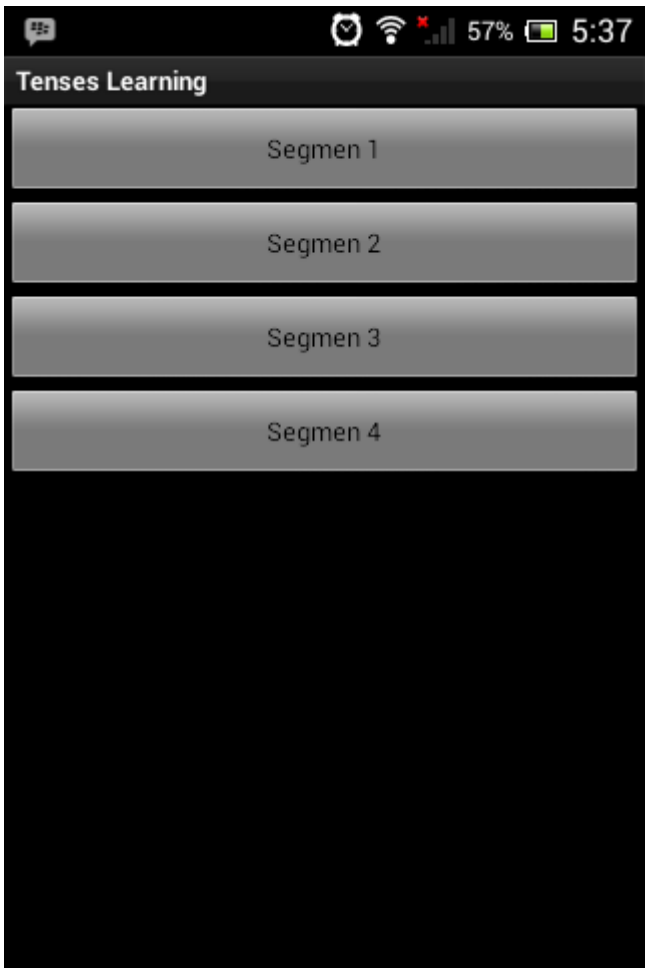
Perangkat Lunak	Spesifikasi
Android OS	Jelly Bean 4.1.2
Eclipse	v21.0.1-543035
Database Manager	SQLite

IV.1.4 Implementasi Antar Muka

Implementasi antar muka merupakan implementasi dari perancangan antar muka yang telah dirancang pada bab sebelumnya. Implementasi antar muka pada aplikasi yang telah dibangun adalah sebagai berikut:

1. Antar muka menu utama

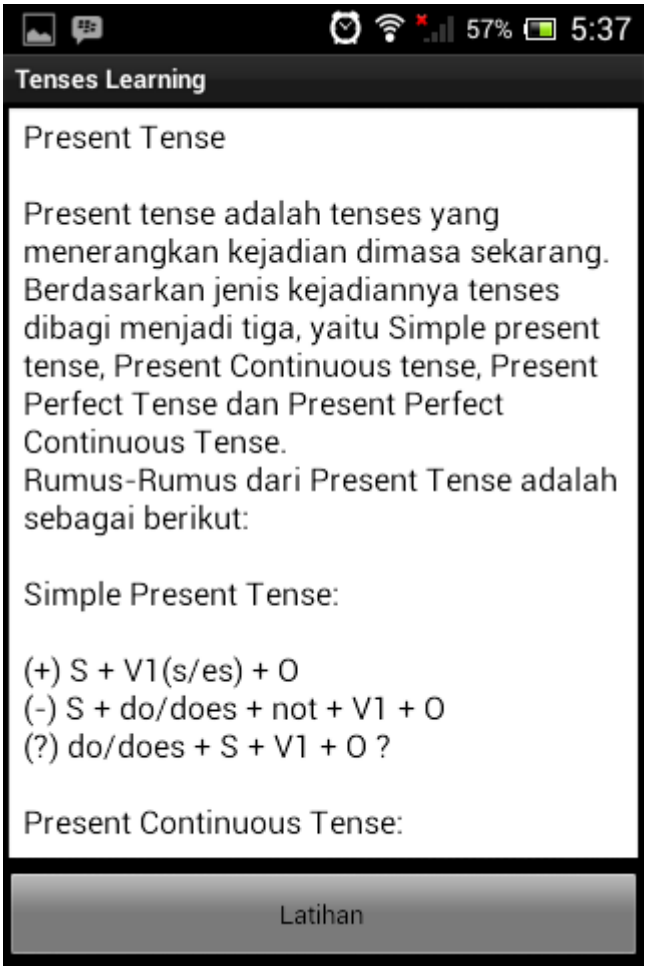
   Antar muka pada menu utama menampilkan tampilan awal yang berisi tombol segmen 1, segmen 2, segmen 3 dan segmen 4. Keempat segmen itu berisi materi, latihan soal dan ujian seperti pada Gambar IV.1.



Gambar IV.1 Antar Muka Menu Utama

2. Antar muka menu materi

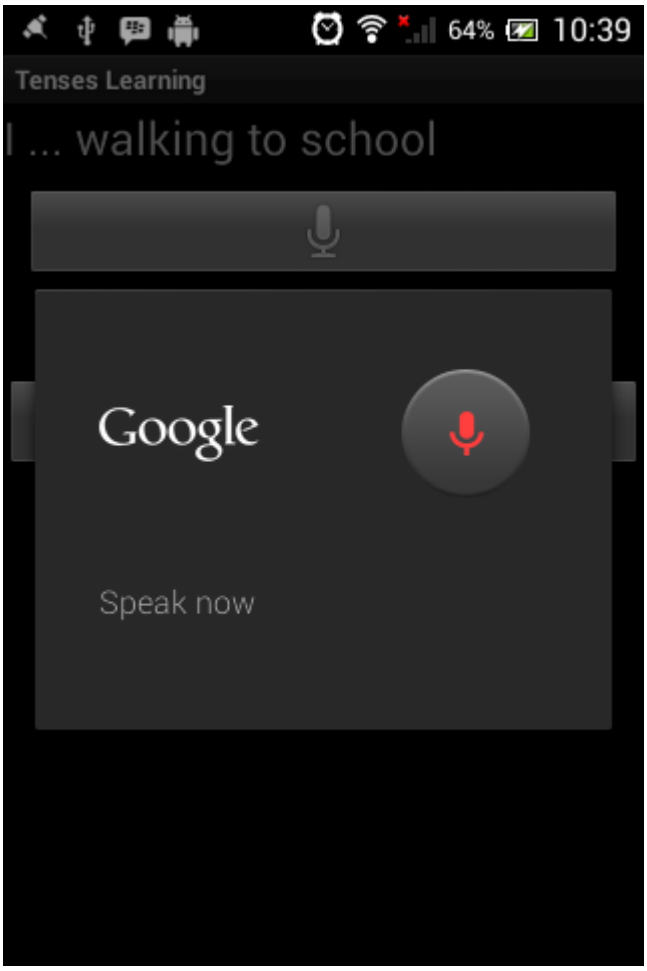
   Pada antar muka menu materi ini berisikan materi formula pada segmen yang telah dipilih sebelumnya. Pada antar muka yang terlihat pada Gambar IV.2 adalah antar muka segmen 1 yaitu *Present Tense*. Pada antar muka ini juga terdapat tombol latihan untuk masuk ke sesi selanjutnya.



Gambar IV.2 Antar Muka Menu Materi

3. Antar muka menu latihan

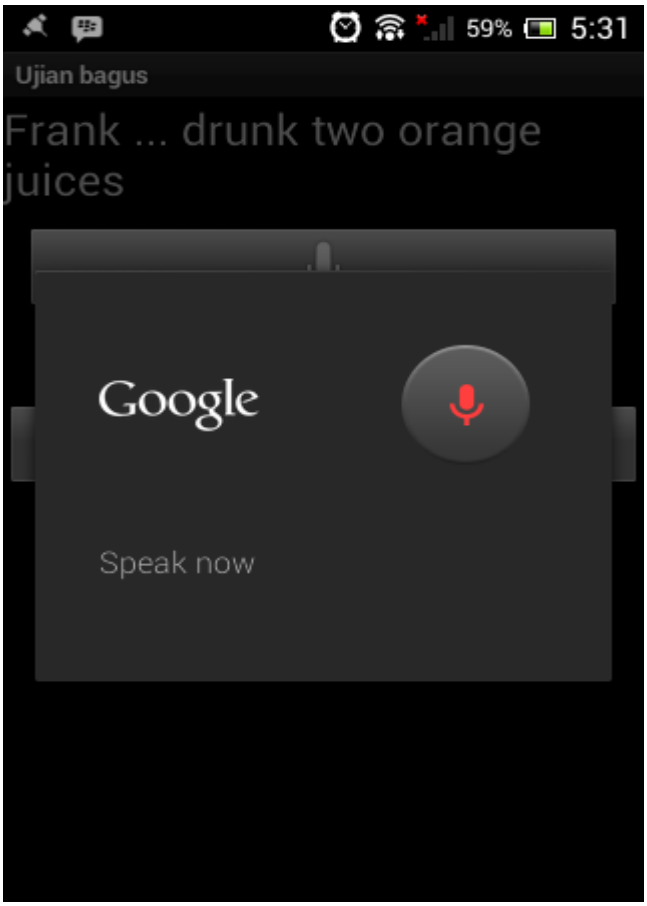
   Antar muka menu latihan adalah antar muka yang menampilkan tampilan pada menu latihan. Menu latihan terdiri dari soal, tombol *speech* untuk menjawab pertanyaan, dan tombol jawab untuk mengkonfirmasi jawaban. Antar muka latihan bias dilihat di Gambar IV.3



Gambar IV.3 Antar Muka Latihan dan Tampilan Google *speech*

4. Antar muka menu ujian

   Setelah pengguna menjawab semua pertanyaan di menu latihan, pengguna akan masuk ke menu ujian seperti terlihat pada Gambar IV.4. Jumlah soal ujian bergantung pada jumlah jawaban benar pengguna sesuai ketentuan yang telah dijelaskan di bab sebelumnya



Gambar IV.4 Antar Muka Ujian

IV.1.5 Implementasi Basis Data

Pembuatan basis data yang dilakukan menggunaka MySql . Implementasi basis data pada MySql dapat dilihat pada tabel IV.3 berikut:

Tabel IV.3 Implementasi Basis Data

No	Nama Tabel	Perintah
1	Segmen1	CREATE TABLE `segmen1` (`id` TEXT, `soal` TEXT, `key` TEXT PRIMARY KEY (id)
2	Segmen2	CREATE TABLE `segmen2` (`id` TEXT, `soal` TEXT, `key` TEXT PRIMARY KEY (id)
3	Segmen3	CREATE TABLE `segmen3` (

		`id` TEXT, `soal` TEXT, `key` TEXT PRIMARY KEY **(**id**)**
4	Segmen4	CREATE TABLE `segmen4` (`id` TEXT, `soal` TEXT, `key` TEXT PRIMARY KEY **(**id**)**

IV.2 Pengujian Sistem

Pengujian sistem dilakukan untuk menguji aplikasi yang telah dibangun. Pengujian sistem menggunakan dua teknik yaitu pengujian *black box* dan *white box*. Pengujian *black box* digunakan untuk menemukan kesalahan pada program, sedangkan pengujian *white box* digunakan untuk mengetahui apakah masih ada *error* pada *source code* didalam program.

IV.2.1 Skenario Pengujian *Black box*

Pada Skenario pengujian ini akan dijelaskan pengujian terhadap sistem yang ada pada aplikasi *game* yang dibangun. Skenario pengujian dari aplikasi yang dibangun dapat dilihat pada Tabel IV.4

Tabel IV.4 Skenario Pengujian

No	Item uji	Detail Pengujian	Jenis Pengujian
1	Menu utama	Menampilkan menu segmen 1	*Black box*
		Menampilkan menu segmen 2	*Black box*
		Menampilkan menu segmen 3	*Black box*
		Menampilkan menu segmen 4	*Black box*
		Menampilkan halaman materi segmen	*Black box*

2	Menu materi	Memilih menu latihan	*Black box*
3	Menu Latihan	Menjawab jawaban	*Black box*
		Menampilkan soal	*Black box*
		Menekan tombol *speech*	*Black box*
		Menekan tombol jawab	*Black box*
4	Menu Ujian	Menjawab jawaban ujian	*Black box*
		Menampilkan soal	*Black box*
		Menekan tombol *speech*	*Black box*
		Menampilkan tombol jawab	*Black box*

IV.2.2 Hasil Pengujian *Black Box*

Tabel IV.5 Hasil Pengujian

No	Kasus/Uji	Skenario Uji	Hasil yang Diharapkan	Hasil Pengujian
1	Menu Utama	Memilih Menu Segmen 1	Masuk ke menu materi	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Segmen 2	Memilih Menu Segmen 2	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Segmen 3	Memilih Menu Segmen 3	[√] Berhasil [ ] Tidak Berhasil

		Memilih Menu Segmen 4	Memilih Menu Segmen 4	[√] Berhasil [ ] Tidak Berhasil
2	Memilih Menu latihan	Memililh Menu Segmen	Menampilkan Materi	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Latihan	Menampilkan Menu Latihan	[√] Berhasil [ ] Tidak Berhasil
3	Menu Latihan	Memilih Menu Latihan	Menampilkan Soal	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu *Speech*	Menampilkan *google speech*	[√] Berhasil [ ] Tidak Berhasil
		Memilih menu jawab	Menampilkan jawaban	[√] Berhasil [ ] Tidak Berhasil
4	Menu Ujian	Memilih Menu Ujian	Menampilkan Soal	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu *Speech*	Menampilkan *google speech*	[√] Berhasil [ ] Tidak Berhasil
		Memilih menu jawab	Menampilkan jawaban	[√] Berhasil [ ] Tidak Berhasil

IV.2.2.3 Pengujian *White Box*

Pengujian *white box* merupakan. pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Pengujian *white box* merupakan cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak. Jika ada modul yang menghasilkan output yang

tidak sesuai dengan proses bisnis yang dilakukan, maka baris-baris program, variabel, dan parameter yang terlibat pada unit tersebut akan dicek satu persatu dan diperbaiki, kemudian di-compile ulang.

1. *Flow Graph* Algoritma Genetika

   Untuk membuat *flow graph*, *source code* PDL (Program Design Language) pada algoritma genetika sesuai nomor untuk setiap node nya. *Source code* pada algoritma genetika adalah sebagai berikut:

```
    public class GA {


          static float[] probc=new float[5];
          static float[] fitness=new float[6];

    public static int[][] total(int batas){
1         int[][] pop=init(batas);
2         fitness=evaluasi(pop,batas);
3         probc=prob(fitness);
4         int[][] popn=RW(batas,pop,probc);

          return popn;
    }

    public static int[][] init(int batas){
5         Random r= new Random();
6         int[][] pop=new int[5][2];
7         for (int i=0;i<5;i++){
8               pop[i][0]=(int)r.nextInt(batas - 1) + 1;
                pop[i][1]=(int)r.nextInt(batas - 1) + 1;
          }

9         return pop;

    }
    public static float[] evaluasi(int[][] populasi,int batas){
10        int ob;
11        for(int i=0;i<5;i++){
12              ob= Math.abs(populasi[i][0]+populasi[i][1]-batas)+1;
13              fitness[i]=(float)1/ob;
          }
14        fitness[5]=fitness[0]+fitness[1]+fitness[2]+fitness[3]+fitn
    ess[4]+fitness[5];
          return fitness;

    }
    public static float[] prob(float[] fitness){
          float[] prob=new float[5];
15        for(int i=0;i<5;i++){
16        prob[i]=fitness[i]/fitness[5];
          }

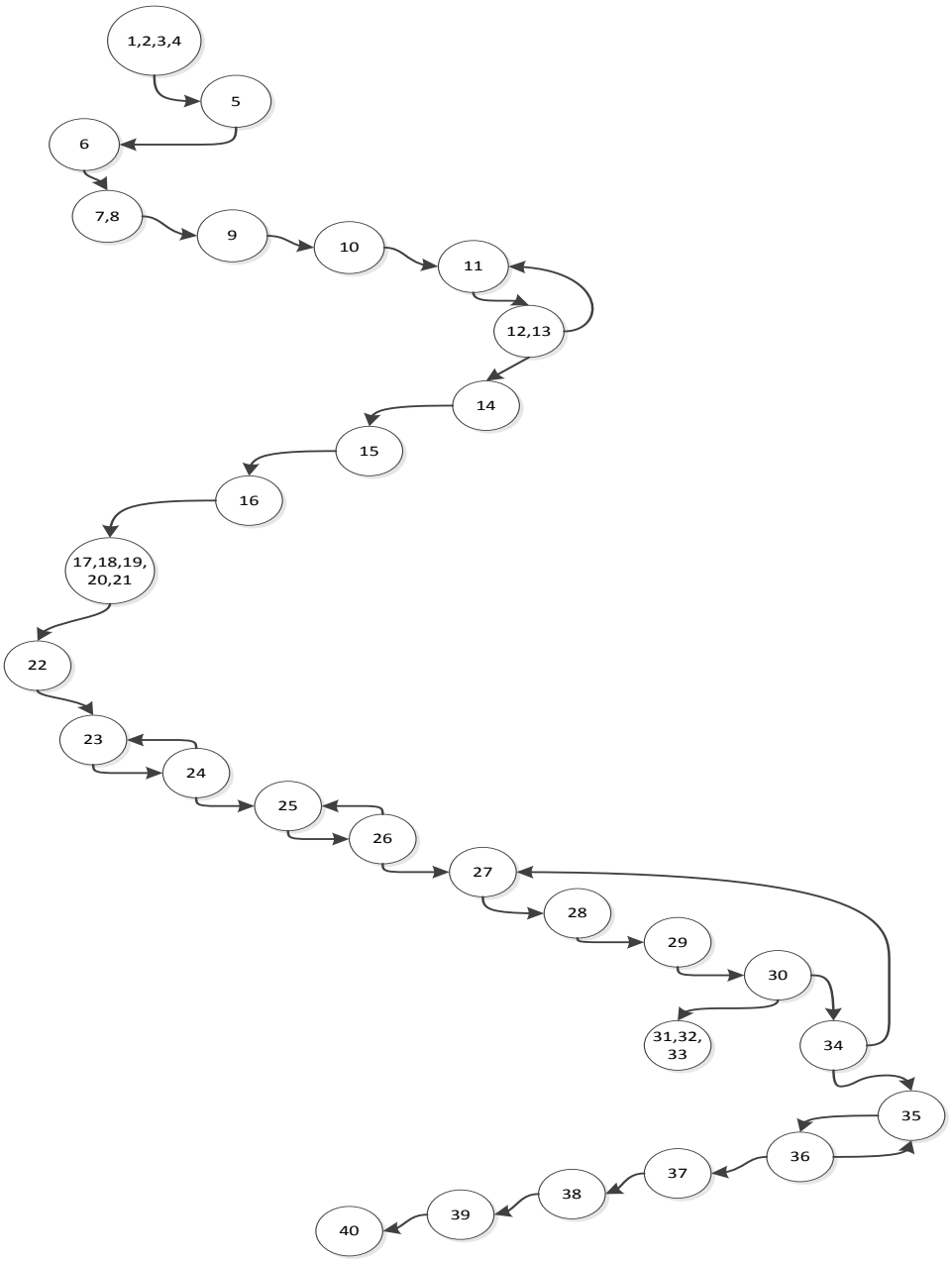
17        probc[0]=prob[0];
18        probc[1]=prob[0]+prob[1];
19        probc[2]=prob[0]+prob[1]+prob[2];
20        probc[3]=prob[0]+prob[1]+prob[2]+prob[3];
21        probc[4]=prob[0]+prob[1]+prob[2]+prob[3]+prob[4];
22
```

```
        return probc;

    }
    public static int[][] RW(int batas,int[][] pop,float[] probc){
        Random r= new Random();


23      float[] ran=new float[5];
24      for (int i=0;i<5;i++){
            ran[i]=(float)r.nextFloat();
        }
25      float[] ranc=new float[5];
26      for (int x=0;x<5;x++){
            ranc[x]=(float)r.nextFloat();
        }
27
28      for(int a=0;a<5;a++){
29          int c=0;
30          while(c<5){
31          if(ran[a]<=probc[c]){
32          pop[a][0]=pop[c][0];
33          pop[a][1]=pop[c][1];
            c=6;
34          }else
                c++;
            }
        }
35
        for(int z=0;z<5;z++){
36
37          if(ran[z]<=0.5){
            if(ran[(z+1)%5]<=0.5 ){
                    pop[z][1]=pop[(z+1)%5][1];
38              }
                if(ran[(z+2)%5]<=0.5 ){
                    pop[z][1]=pop[(z+2)%5][1];
39              }
                if(ran[(z+3)%5]<=0.5 ){
                    pop[z][1]=pop[(z+3)%5][1];
40              }
                if(ran[(z+4)%5]<=0.5 ){
                    pop[z][1]=pop[(z+4)%5][1];
```

Flow graph algoritma genetika dapat dilihat seperti pada gambar IV.5



Gambar IV.5 *Flow Graph* Algortima Genetika

2. *Cyclomatic* Complexity

   Edge = 33, Node = 29

   V(G) = Edge – Node + 2

   V(G) = 33-29 + 2

V(G) = 6

Keterangan :

E : Jumlah busur pada flowgraph

N : Jumlah simpul pada flowgraph

V(G) : Cyclomatic Complexity

3. *Independent* Path

Path 1 : 1,2,3,4 – 5 – 6 – 7,8 – 9 – 10 – 11 – 12,13 – 14 – 15 – 16 - 17,18,19,20,21- 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31,32,33

Path 2 : 1,2,3,4 – 5 – 6 – 7,8 – 9 – 10 – 11 – 12,13 – 14 – 15 – 16 - 17,18,19,20,21- 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 34 – 35 – 36 – 37 – 38 – 39 – 40

4. Setelah dilakukan pengujian hasil yang didapat bisa dilihat pada tabel IV.3 berikut:

Tabel IV.6 Hasil Pengujian *Flow Graph*

Path no	Node (n)	Hasil yang diharapkan	Hasil sesuai uji kasus	keterangan
1	1,2,3,4 – 5 – 6 – 7,8 – 9 – 10 – 11 – 12,13 – 14 – 15 – 16 - 17,18,19,20,21- 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31,32,33	Menghasilkan gen yang sudah melewati proses *Roulette Wheel*	Menghasilkan gen yang sudah di *roulette wheel*	[√] alur terlewati [ ] alur tidak terlewati
2	1,2,3,4 – 5 – 6 – 7,8 – 9 – 10 – 11 – 12,13 – 14	Menghasilkan kromosom yang sudah di	Fungsi objektif hasilnya = 0	[√] alur terlewati

	– 15 – 16 - 17,18,19,20,21- 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 34 – 35 – 36 – 37 – 38 – 39 – 40	*crossover*		[ ] alur tidak terlewati

IV.2.2.2 Kesimpulan Hasil White Box dan Black Box

Hasil pengujian *black-box* dengan pengujian diatas dapat ditarik kesimpulan bahwa pembangunan aplikasi *game* edukasi *tenses* bahasa inggris berbasis *mobile* tidak terdapat kesalahan proses dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

Dari hasil pengamatan dan pengujian *white-box* dan *black-box*. Pengujian *white-box* dapat disimpulkan bahwa pengacakan kata dalam *game* edukasi pintar struktur bahasa sudah sesuai dengan yang diharapkan.

IV.2.3 Pengujian Kuesioner

Pengujian kuesioner adalah media yang digunakan pengguna aplikasi pembelajaran *tenses* bahasa Inggris berbasis *mobile* untuk memberikan penilaian terhadap aplikasi yang telah dibangun. Berdasarkan hasil kuesioner tersebut akan dilakukan perhitungan agar bisa diambil kesimpulan dari penerapan aplikasi yang dibangun. Kuesioner terdiri dari 4 pertanyaan dengan menggunakan skala likert dari skala 1 – 5. Adapun pertanyaan dari kuesioner pengujian beta adalah sebagai berikut:

1. Aplikasi ini membantu anda untuk mendapatkan materi pembelajaran mengenai tenses
   A. Sangat Baik
   B. Baik
   C. Cukup
   D. Ragu-ragu

E. Kurang

2. Aplikasi ini memudahkan anda untuk memahami tenses
   A. Sangat Baik
   B. Baik
   C. Cukup
   D. Ragu-ragu
   E. Kurang
3. Pemanfaat Speech Recognition membantu pembelajaran speaking anda
   A. Sangat Baik
   B. Baik
   C. Cukup
   D. Ragu-ragu
   E. Kurang
4. Jumlah soal di aplikasi ini melatih anda memahami tenses
   A. Sangat Baik
   B. Baik
   C. Cukup
   D. Ragu-ragu
   E. Kurang
5. Platform android memudahkan penggunaan pada aplikasi
   A. Sangat Baik
   B. Baik
   C. Cukup
   D. Ragu-ragu
   E. Kurang
6. Jenis pertanyaan yang ditanyakan membantu anda memahami *to be* dan *verb*
   A. Sangat Baik
   B. Baik
   C. Cukup
   D. Ragu-ragu

E. Kurang

Skor penilaian yang diberikan dengan menggunakan skala likert pada setiap pertanyaannya dapt dilihat di tabel IV.4:

Tabel IV.7 Tabel Skor Penilaian Kuesioner

Skala Jawaban	Keterangan	Skor
SB	Sangat Baik	5
B	Baik	4
C	Cukup	3
R	Ragu-Ragu	2
K	Kurang	1

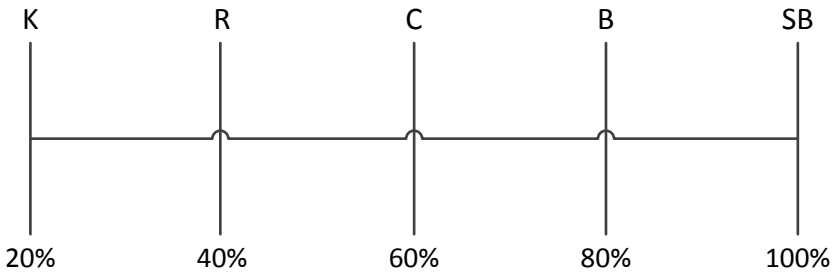
Berdasarkan data hasil kuesioner tersebut, dapat dicari persentase masing-masing jawaban dengan menggunakan rumus:

$$Y = \frac{X}{Skor\ Ideal} \times 100\%$$

Keterangan:

- Y = Nilai persentase yang dicari
- X = Jumlah nilai kategori jawaban dilakukan dengan frekuensi ($\Sigma = N \times R$)
- N = Nilai dari setiap jawaban.
- R = Frekuensi
- Skor Ideal = Nilai tertinggi dikalikan dengan jumlah sampel ($5 \times 30 = 150$).

Setelah diperoleh nilai Y, maka intepretasi nilai atau skor dapt dilihat pada gambar IV.6 berikut:



Gambar IV.6 Gambar Skala Penilaian

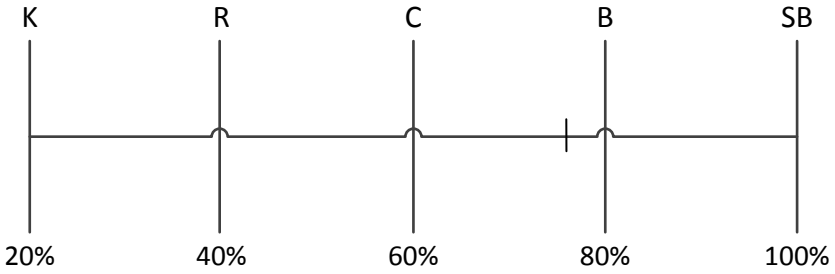
1. Aplikasi ini membantu anda untuk mendapatkan materi pembelajaran mengenai tenses

Tabel IV.8 Tabel Hasil Pertanyaan Kuesioner Nomor 1

Pertanyaan	Keterangan	Skor (N)	Responden (R)	N × R
1	Sangat Baik	5	2	10
	Baik	4	21	84
	Cukup	3	7	21
	Ragu-ragu	2	0	0
	Kurang	1	0	0
Jumlah			**30**	**115**

$$Y = \frac{115}{150} \times 100\% = 76{,}6\%$$

Berdasarkan nilai persentase tersebut, maka dapat disimpulkan bahwa pengguna merasa terbantu dalam mendapatkan materi tentang *tenses*. Berikut skala hasil penilaian nomor 1 yang ditunjukan pada gambar berikut:



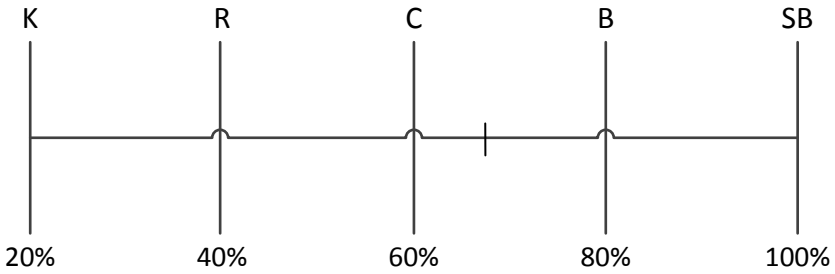
2. Aplikasi ini memudahkan anda untuk memahami tenses

Tabel IV.9 Tabel Hasil Pengujian Pertanyaan Kuesioner Nomor 2

Pertanyaan	Keterangan	Skor (N)	Responden (R)	N × R
2	Sangat Baik	5	1	5
	Baik	4	16	64
	Cukup	3	11	33
	Ragu-ragu	2	0	0
	Kurang	1	0	0
Jumlah			**30**	**102**

$$Y = \frac{102}{150} \times 100\% = 68\%$$

Berdasarkan nilai persentase tersebut, maka dapat disimpulkan bahwa pengguna merasa mendapat kemudahan dalam memahami materi tentang *tenses*. Berikut skala hasil penilaian nomor 2 yang ditunjukan pada gambar berikut:



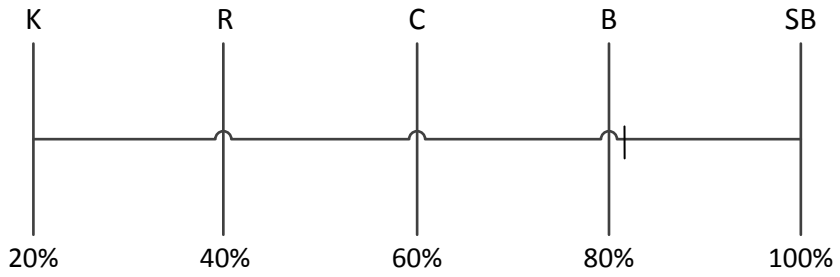
3. Pemanfaat Speech Recognition membantu pembelajaran speaking anda

Tabel IV.10 Tabel Hasil Pengujian Pertanyaan Kuesioner Nomor 3

Pertanyaan	Keterangan	Skor (N)	Responden (R)	N × R
3	Sangat Baik	5	5	25
	Baik	4	21	84
	Cukup	3	4	12
	Ragu-ragu	2	0	0
	Kurang	1	0	0
Jumlah			**30**	**121**

$$Y = \frac{121}{150} \times 100\% = 80{,}6\%$$

Berdasarkan nilai persentase tersebut, maka dapat disimpulkan bahwa pengguna merasa terbantu dalam pembelajaran *speaking* dengan adanya fitur *speech recognition*. Berikut skala hasil penilaian nomor 3 yang ditunjukan pada gambar berikut:



4. Jumlah soal di aplikasi ini melatih anda memahami tenses

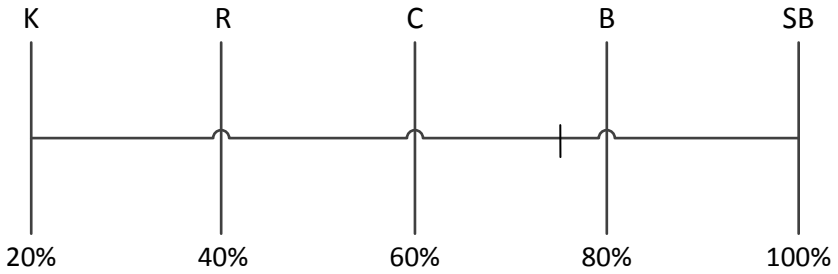
Tabel IV.11 Tabel Hasil Pengujian Pertanyaan Kuesioner Nomor 4

Pertanyaan	Keterangan	Skor (N)	Responden (R)	N × R
4	Sangat Baik	5	2	10
	Baik	4	17	68

	Cukup	3	12	36
	Ragu-ragu	2	0	0
	Kurang	1	0	0
Jumlah			**30**	**110**

$$Y = \frac{114}{150} \times 100\% = 76\%$$

Berdasarkan nilai persentase tersebut, maka dapat disimpulkan bahwa pengguna merasa terbantu dengan jumlah soal untuk memahami *tenses* bahasa Inggris. Berikut skala hasil penilaian nomor 4 yang ditunjukan pada gambar berikut:



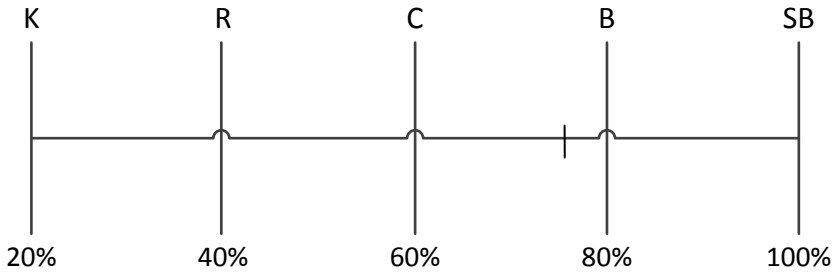
5. Platform android memudahkan penggunaan pada aplikasi

Tabel IV.12 Tabel Hasil Pengujian Pertanyaan Kuesioner Nomor 5

Pertanyaan	**Keterangan**	**Skor (N)**	**Responden (R)**	**N × R**
5	Sangat Baik	5	0	0
	Baik	4	21	84
	Cukup	3	9	27
	Ragu-ragu	2	0	0
	Kurang	1	0	0
Jumlah			**30**	**111**

$$Y = \frac{111}{150} \times 100\% = 74\%$$

Berdasarkan nilai persentase tersebut, maka dapat disimpulkan bahwa pengguna merasa terbantu penerapan aplikasi ini pada platform android. Berikut skala hasil penilaian nomor 5 yang ditunjukan pada gambar berikut:



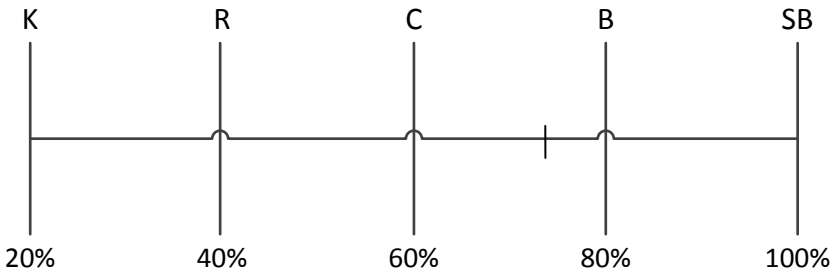
6. Jenis pertanyaan yang ditanyakan membantu anda memahami *to be* dan *verb*

Tabel IV.13 Tabel Hasil Pengujian Pertanyaan Kuesioner Nomor 6

Pertanyaan	Keterangan	Skor (N)	Responden (R)	N × R
6	Sangat Baik	5	0	0
	Baik	4	25	100
	Cukup	3	5	15
	Ragu-ragu	2	0	0
	Kurang	1	0	0
Jumlah			**30**	**115**

$$Y = \frac{115}{150} \times 100\% = 76{,}6\%$$

Berdasarkan nilai persentase tersebut, maka dapat disimpulkan bahwa pengguna merasa terbantu dalam memahami *tenses* dengan jenis pertanyaan *to be* dan *verb*. Berikut skala hasil penilaian nomor 6 yang ditunjukan pada gambar berikut:



IV.2.3.1 Kesimpulan Pengujian Kuesioner

Berdasarkan hasil pengujian beta yang telah dilakukan, pengguna aplikasi pembelajaran edukasi *tenses* bahasa Inggris bahwa pembangunan aplikasi pembelajaran edukasi *tenses* bahasa Inggris berbasis *mobile* ini telah sesuai dengan tujuan yang ingin dicapai.