BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Proses implementasi merupakan terjemahan perancangan yang berdasarkan hasil analisis pada bab sebelumnya kedalam bahasa pemrograman yang dapat dimengerti oleh komputer. Bahasa pemrograman dalam sistem ini menggunakan bahasa *Java* dan *Corel Draw X6* sebagai *tools* untuk desain gambar.

4.2 Kebutuhan Perangkat Lunak

Untuk memastikan *game* Edukasi Mengenal dan Membaca Bahasa Arab dapat dimainkan, maka langkah-langkah yang harus dilakukan adalah sebagai berikut :

1. Download JDK 1.5.0 version atau versi diatasnya (Jika Anda belum mempunyai *Software Java Development Kit*).
2. Install JDK 1.5.0.
3. Download Greenfoot-windows-201 (Jika Anda belum mempunyai Greenfoot-windows-201).
4. Install Greenfoot-windows-201.

5. Klik *file* jar (exe) *game* Edukasi Mengenal dan Membaca Bahasa Arab.
6. *Game* Edukasi Mengenal dan Membaca Bahasa Arab dapat dimainkan.

4.3 Implementasi Aplikasi

4.3.1 Sisi *User*

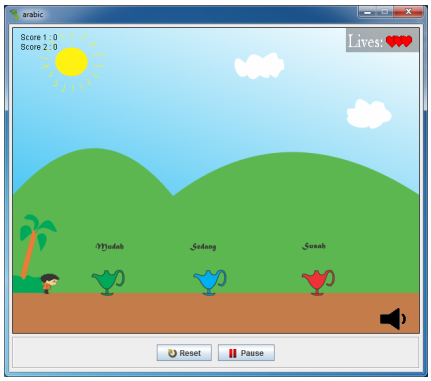
1. Tampilan Utama



Gambar 4.1 Tampilan Utama

Layar ini berfungsi menampilkan tampilan utama atau tampilan awal *game* dimana didalamnya tersedia tiga pilihan yaitu pilihan mulai, *history* dan petunjuk. Untuk memulai permainan pemain dapat memilih pilihan mulai.

2. Tampilan Pilihan Tingkatan Kesulitan



Gambar 4.2 Tampilan Pilihan Tingkat Kesulitan

Layar ini berfungsi menampilkan pilihan tingkatan kesulitan yang terdapat pada *game* yaitu lampu ajaib hijau menunjukkan pilihan tingkat kesulitan mudah, lampu ajaib biru menunjukkan pilihan tingkat kesulitan sedang dan lampu ajaib merah menunjukkan pilihan tingkat kesulitan susah.

3. Tampilan Pilihan Jenis Permainan



Gambar 4.3 Tampilan Pilihan Jenis Permainan

Layar ini berfungsi menampilkan pilihan jenis permainan.

4. Tampilan Jenis Permainan Belajar Bahasa Arab angka dan warna



Gambar 4.4 Tampilan Jenis Permainan Belajar Bahasa Arab Angka dan Warna 1

Layar ini berfungsi untuk menampilkan permainan Belajar Bahasa Arab Angka dan Warna dengan *single player*



Gambar 4.5 Tampilan Jenis Permainan Belajar Bahasa Arab Angka dan Warna 2

Layar ini berfungsi untuk menampilkan menampilkan permainan Belajar Bahasa Arab Angka dan Warna dengan dengan *multiplayer*

5. Tampilan Jenis Permainan Bermain dan Belajar Bahasa Arab



Gambar 4.6 Tampilan Jenis Permainan Bermain dan Belajar Bahasa Arab

Layar ini berfungsi untuk menampilkan permainan Bermain dan Belajar Bahasa Arab dengan *single player*



Gambar 4.7 Tampilan Jenis Permainan Bermain dan Belajar Bahasa Arab 2

Layar ini berfungsi untuk menampilkan permainan Bermain dan Belajar Bahasa Arab dengan *multiplayer*

6. Tampilan Jenis Permainan Petualangan



Gambar 4.8 Tampilan Jenis Permainan Petualangan 1

Layar ini berfungsi untuk menampilkan permainan Petualangan dengan *single player*



Gambar 4.9 Tampilan Jenis Permainan Petualangan 2

Layar ini berfungsi untuk menampilkan permainan Petualangan dengan *multiplayer*

7. Tampilan Tamat

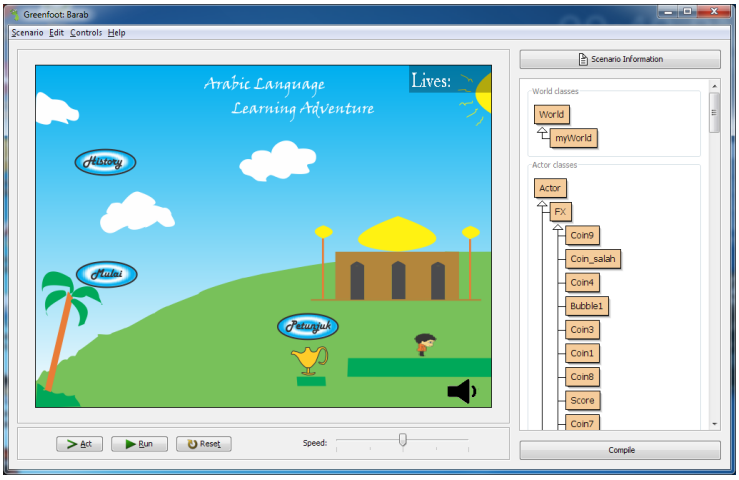


Gambar 4.10 Tampilan Tamat

Layar ini berfungsi untuk menampilkan tampilan tamat atau permainan selesai

4.3.2 Sisi Pembuat

Berikut dibawah ini tampilan dari pengembangan *game* Edukasi Mengenal dan Membaca Bahasa Arab yaitu :



Gambar 4.11 Tampilan Pengembangan *Game*

Layar menampilkan IDE Greenfoot yang digunakan untuk membuat *Game* Edukasi Mengenal dan Membaca Bahasa Arab

4.4 Pengujian Sistem

Pengujian sistem merupakan hal terpenting yang dilakukan untuk menemukan kekurangan atau kesalahan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

4.4.1 Pengujian *White Box*

Pengujian *white box* atau *white box testing* merupakan metode perancangan *test case* yang menggunakan struktur *control* dari perancangan *procedural* dalam mendapatkan *test case*. Adapun metode yang digunakan dalam pengujian *white box* adalah metode Basis *Path*. Metode Basis *Path* mengijinkan pendesain kasus uji untuk membuat perkiraan lojik yang kompleks dari desain prosedural dan menggunakan perkiraan ini untuk mendefinisikan aliran eksekusi.

4.4.1.1 Pengujian *White Box* Algoritma BFS

Berikut ini adalah *pseudocode* dari algoritma BFS :

```
1: baca node awal dan node akhir
   simpan node awal pada record
2: Do while record masih ada
        baca record
3:      if (record = node akhir)
4:              return record;
                break;
5:      else
        simpan record child (left, top, right, bottom) pada child node
6:              Do while child node masih ada
7:                      proses child node
                        simpan child node pada record
8:              endDo
```

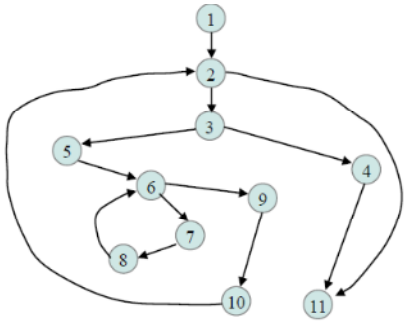
9: endif

10: endDo

return null

11: end

Berdasarkan *pseudocode* di atas maka diagram alirnya adalah sebagai berikut :



Gambar 4.12 Diagram alir algoritma BFS

Dari diagram alir di atas dapat dihitung *cyclomatic complexity* yaitu :

1. *Flowgraph* mempunyai 4 *region*.
2. V(G) = 13 edge – 11 node + 2 = 4
3. V(G) = 3 predicate node + 1 = 4

Dari hasil perhitungan *cyclomatic complexity* terdapat 4 *independent path* yaitu :

Path 1 = 1 – 2 – 11

Path 2 = 1 – 2 – 3 – 4 – 11

Path 3 = 1 – 2 – 3 – 5 – 6 – 9 – 10 – 2 – 11

Path 4 = 1 – 2 – 3 – 5 – 6 – 7 – 8 – 6 – 9 – 10 – 2 – 11

Untuk melakukan uji coba basis *path* di atas digunakan *graph* matrik. *Graph* matrik merupakan matrik empat persegi yang mempunyai ukuran yang sama dengan jumlah *node* pada *flowgraph.* Adapun *graph* matrik pada *flowgraph* BFS di atas adalah sebagai berikut :

	1	2	3	4	5	6	7	8	9	10	11	n(E)-1
1		1										1-1 = 0
2			1								1	2-1 = 1
3				1	1							2-1 = 1
4											1	1-1 = 0
5						1						1-1 = 0
6							1		1			2-1 = 1
7								1				1-1 = 0
8						1						1-1 = 0
9										1		1-1 = 0
10		1										1-1 = 0
11												0
											Sum(E)+1	3+1=4

Gambar 4.13 *Graph* Matrik BFS

Keterangan :

1. Baris dan kolom merepresentasikan simpul.
2. Nilai 1 merepresentasikan adanya keterhubungan antar simpul.

4.4.1.2 Pengujian *White Box* Algoritma DFS

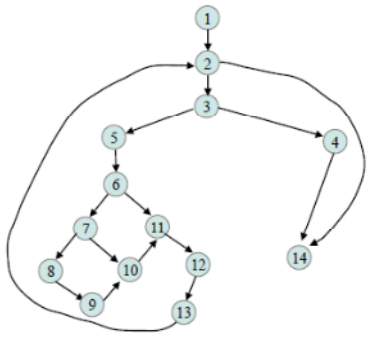
Berikut ini adalah *pseudocode* dari algoritma DFS :

```
1: baca node awal dan node akhir
   node = node awal
2: Do while (node != null)
        baca node
3:      if (node = node akhir)
4:              return node
                break
5:      else
                hasil = proses node child (left)
6:              if (hasil = null)
                        hasil = proses node child (top)
7:                      if (hasil = null)
                                hasil = proses node child (right)
8:                              if (hasil = null)
                                        hasil = proses node child (bottom)
9:                              endif
10:                     endif
11:             endif
                node = hasil
12:     endif
13: endDo
```

return null

14: end

Berdasarkan *pseudocode* di atas maka diagram alirnya adalah sebagai berikut :



Gambar 4.14 Diagram alir algoritma DFS

Dari diagram alir di atas dapat dihitung *cyclomatic complexity* yaitu :

1. *Flowgraph* mempunyai 5 *region*.
2. V(G) = 17 edge – 14 node + 2 = 5
3. V(G) = 4 predicate node + 1 = 5

Dari hasil perhitungan *cyclomatic complexity* terdapat 5 *independent path* yaitu :

Path 1 = 1 – 2 – 14

Path 2 = 1 – 2 – 3 – 4 – 14

Path 3 = 1 – 2 – 3 – 5 – 6 – 11 – 12 – 13 – 2 – 14

Path 4 = 1 – 2 – 3 – 5 – 6 – 7 – 10 – 11 – 12 – 13 – 2 – 14

Path 5 = 1 – 2 – 3 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 2 – 14

Untuk melakukan uji coba basis *path* di atas digunakan *graph* matrik. *Graph* matrik merupakan matrik empat persegi yang mempunyai ukuran yang sama dengan junlah *node* pada *flowgraph*. Adapun *graph* matrik pada *flowgraph* DFS di atas adalah sebagai berikut :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	n(E)-1
1		1													1-1 = 0
2			1											1	2-1 = 1
3				1	1										2-1 = 1
4														1	1-1 = 0
5						1									1-1 = 0
6							1				1				2-1 = 1
7								1		1					2-1 = 1
8									1						1-1 = 0
9										1					1-1 = 0
10											1				1-1 = 0
11												1			1-1 = 0
12													1		1-1 = 0
13		1													1-1 = 0
14															0
														Sum(E)+1	4+1 = 5

Gambar 4.15 Graph Matrik DFS

Keterangan :

1. Baris dan kolom merepresentasikan simpul.
2. Nilai 1 merepresentasikan adanya keterhubungan antar simpul.

4.4.2 Hasil Pengujian Algoritma BFS dan Algoritma DFS

Berdasarkan pengujian maka berikut hasil pengujian dari algoritma BFS dan DFS yaitu :

a. Hasil pengujian Algoritma BFS pada objek musuh

Tabel 4.1 Hasil Pengujian Algoritma BFS

Kasus dan Hasil Uji				
No. *Path*	Node (n), input kondisi	Yang Diharapkan	Pengamatan	Kesimpulan
1.	1(*void*) 2(*false*) 11(*void*)	Rute pemain ditemukan	Rute pemain ditemukan	*Path* dikerjakan
2.	1(*void*) 2(*false*) 3(*true*) 4,11(*void*)	Rute pemain ditelusuri	Rute pemain ditemukan	*Path* dikerjakan
3.	1(*void*) 2(*false*) 3,5,6(*true*) 9(*true*) 10,11(*void*)	Rute Pemain ditelusuri	Rute pemain ditemukan	*Path* dikerjakan
4.	1(*void*) 2(*false*) 3,5,6(*false*)	Rute Tidak dapat ditelusuri	Rute pemain tidak ditemukan	*Path* dikerjakan

	7,8,9(*true*) 10,11(*void*)			

b. Hasil pengujian Algoritma DFS pada objek kayu jatuh

Tabel 4.2 Hasil Pengujian Algoritma DFS

Kasus dan Hasil Uji				
No. *Path*	Node (n), input kondisi	Yang Diharapkan	Pengamatan	Kesimpulan
1.	1(*void*) 2(*false*) 14(*void*)	Pergerakan aktor Ditemukan	Pergerakan aktor ditemukan	*Path* dikerjakan
2.	1(*void*) 2(*false*) 3,4(*true*) 14(*void*)	Pergerakan aktor ditelusuri	Pergreakan aktor ditelusuri	*Path* dikerjakan
3.	1(*void*) 2(*false*) 3,5,6(*true*) 11,12(*true*) 13,14(*void*)	Pergerakan kayu jatuh	Pergerakan kayu Jatuh	*Path* dikerjakan
4.	1(*void*) 2(*false*) 3,5,6(*false*)	Pergerakan kayu mengenai aktor	Pergerakan kayu mengenai aktor	*Path* dikerjakan

	7,10,11(*true*) 12 (true) 13,14(*void*)			
5	1(*void*) 2(*false*) 3,5,6(*false*) 7,8,9(*true*) 10,11,12(*true)* 13,14 (*void*)	Pergerakan aktor tidak dapat ditelusuri	Pergerakan aktor tidak dapat diteludur	*Path* dikerjakan

4.5 Pengujian Alpha (Fungsional)

Pengujian alpha merupakan pengujian fungsional yang digunakan untuk menguji sistem yang baru. Pengujian alpha berfokus pada persyaratan fungsional perangkat lunak.

a. Rencana Pengujian

Renacan pengujian yang akan dilakukan pada aplikasi ini selengkapnya terlihat pada tabel dibawah ini.

Tabel 4.3 Rencana Pengujian Aplikasi

Kelas Uji	Detail Pengujian	Jenis Uji
Mulai	Memulai permainan	*Black Box*
History	Menampilkan tampilan *history*	*Black Box*
Petunjuk	Menampilkan tampilan petunjuk	*Black Box*

Karakter Bergerak	Pergerakan karakter menggunakan tombol-tombol dalam aplikasi	*Black Box*
Enemy	Pergerakan enemy dalam aplikasi	*Black Box*
Soal	Menampilkan tampilan soal	*Black Box*
Edukasi	Menampilkan edukasi bahasa Arab dalam aplikasi	*Black Box*
Suara	Menampilkan suara dalam aplikasi	*Black Box*
Algoritma BFS	Implementasi algoritma BFS terhadap objek musuh	*White Box*
Algoritma DFS	Implementasi algortima DFS terhadap objek kayu jatuh	*White Box*

b. Kasus dan hasil pengujian

Berdasarkan hasil pengujian yang telah disusun, maka dapat dilakukan pengujian sebagai berikut :

1. Pengujian Mulai

Pengujian ini dilakukan untuk memulai aplikasi *game* yang telah dibuat

Tabel 4.4 Pengujian Menu Mulai

Kasus dan hasil uji (data normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih Menu "Mulai"	Memulai permainan kemudian	Permainan dimulai	[√] Tampil [ ] Tidak

	memilih tingkat kesulitan dan pemilihan *level*		

2. Pengujian *History*

   Pengujian ini dilakukan untuk menampilkan informasi *history* yang ada di dalam *game*

Tabel 4.5 Pengujian Menu *History*

Kasus dan hasil uji (data normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih Menu "*History*"	Melihat informasi *history* yang telah dimainkan	*History* permainan	[√] Tampil [ ] Tidak

3. Pengujian Petunjuk

   Pengujian ini dilakukan untuk melihat informasi dan petunjuk yang ada di dalam *game.*

Tabel 4.6 Pengujian Menu Petunjuk

Kasus dan hasil uji (data normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih Menu "Petunjuk"	Melihat informasi dan petunjuk untuk memainkan	Petunjuk permainan	[√] Tampil [ ] Tidak

	permainan		

4. Pengujian Karakter Bergerak

   Pengujian ini dilakukan untuk menampilkan informasi pergerakan karakter yang terdapat dalam *game*.

Tabel 4.7 Pengujian Karakter Bergerak

Kasus dan hasil uji (data normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Tombol →	Abu bergerak ke arah kanan	Pergerakan Abu	[√] Bergerak [ ] Tidak
Tombol ↓	Abu masuk ke dalam lampu ajaib	Pergerakan Abu	[√] Masuk [ ] Tidak
Tombol ←	Abu bergerak ke arah kiri	Pergerakan Abu	[√] Bergerak [ ] Tidak
Tombol ↑	Abu lompat	Pergerakan Abu	[√] Bergerak [ ] Tidak
Tombol ↑ →	Abu lompat ke arah kanan	Pergerakan Abu	[√] Bergerak [ ] Tidak
Tombol ←↑	Abu kompat ke arah kiri	Pergerakan Abu	[√] Bergerak [ ] Tidak
Tombol D	Sholeh bergerak ke arah kanan	Pergerakan Sholeh	[√] Bergerak [ ] Tidak
Tombol A	Sholeh bergerak	Pergerakan Sholeh	[√] Bergerak

	ke arah kiri		[ ] Tidak
Tombol W	Sholeh lompat	Pergerakan Sholeh	[√] Bergerak [ ] Tidak
Tombol S	Sholeh masuk ke dalam lampu ajaib	Pergerakan Sholeh	[√] Masuk [ ] Tidak
Karakter jatuh	Mereset permainan secara otomatis	Pergerakan jatuh karkater	[√] Reset [ ] Tidak

5. Pengujian *Enemy*

Pengujian ini dilakukan untuk menampilkan pergerakan *enemy* yang terdapat dalam *game*.

Tabel 4.8 Pengujian *Enemy*

Kasus dan hasil uji (data normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Karakter mengenai *enemy*	*Lives* karakter berkurang	Pergerakan *enemy*	[√] Berkurang [ ] Tidak

6. Pengujian Soal

Pengujian ini dilakukan untuk menampilkan informasi soal yang ada didalam *game*

Tabel 4.9 Pengujian Soal

Kasus dan hasil uji (data normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Karakter menyundul *block* dengan jawaban benar	Menampilkan tulisan benar dan mendapatkan *score* 100	Tampil tulisan benar dan dapat *score* 100	[√] Tampil [ ] Tidak
Karakter menyundul *block* dengan jawaban salah	Menampilkan tulisan salah dan mendapatkan *score* 1	Tampil tulisan salah dan dapat *score* 1	[√] Tampil [ ] Tidak

7. Pengujian Edukasi

   Pengujian ini dilakukan untuk menampilkan informasi edukasi yang ada didalam *game*

Tabel 4.10 Pengujian Edukasi

Kasus dan hasil uji (data normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih Permainan Belajar Bahasa Arab Angka dan Warna	Menampilkan edukasi pengenalan dan membaca angka 1-10 dan warna	Tampil edukasi pengenalan dan membaca angka1-10 dan warna dalam	[√] Tampil [ ] Tidak

	dalam bahasa Arab	bahasa Arab	
Pilih Permainan Bermaian dan Belajar Bahasa Arab	Menampilkan pertanyaan tentang berhitung	Tampil pertanyaan tentang berhitung	[√] Tampil [ ] Tidak
Pilih Permainan Petualangan	Menampilkan permainan petualangan	Tampil permainan petualangan	[√] Tampil [ ] Tidak

8. Pengujian Suara

   Pengujian ini dilakukan untuk menampilkan bunyi suara atau musik yang terdapat dalam *game*

Tabel 4.11 Pengujian Suara

Kasus dan hasil uji (data normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengklik gambar *speaker* pertama kali	Menampilkan bunyi suara atau musik ke 1	Tampil bunyi suara atau musik ke 1	[√] Tampil [ ] Tidak
Mengklik gambar *speake*r kedua kali	Menampilkan bunyi suara atau musik ke 2	Tampil bunyi suara atau musik ke 2	[√] Tampil [ ] Tidak
Mengklik	Menonaktifkan	Bunyi suara atau	

gambar *speaker* ketiga kali	bunyi suara atau music	musik nonaktif	[ ] Aktif [√] Nonaktif

4.5.1 Hasil Pengujian Alpha

Hasil pengujan dari pengujian alpha yang telah dilakukan, menunjukkan bahwa aplikasi yang dibangun sudah memenuhi persyaratan fungsional. Akan tetapi pada prosesnya masih memungkinkan untuk terjadi kesalahan. Secara fungsional sistem yang telah dibangun sudah dapat menghasilkan keluaran yang diharapkan.

4.6 Pengujian Beta

Pengujian beta merupakan pengujian langsung kepada pengguna untuk mencoba aplikasi yang dibuat. Pengujian yang dilakukan adalah dengan cara membuat kuisioner yang diberikan kepada 30 responden pengguna aplikasi dengan *range* usia yang sesuai dengan sasaran *user*. Berikut adalah rumus yang digunakan dalam perhitungan jawaban pengguna pada kuisioner :

$$Y = P/Q * 100\%$$

Keterangan :

P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

Y = Nilai prosentase

4.6.1 Kuisioner Pengguna

Adapun pertanyaan yang diajukan kepada pengguna dalam kuisioner untuk aplikasi game Edukasi Mengenal dan Membaca Bahasa Arab adalah sebagai berikut :

1. Apakah *game* Edukasi Mengenal dan Membaca Bahasa Arab ini menarik bagi kamu?

   A. Sangat Menarik B. Menarik C. Kurang Menarik D.Tidak Menarik
2. Apakah kamu suka dengan gambar-gambar yang ada didalam *game* ini ?

   A. Sangat Suka B. Suka C. Kurang Suka D. Tidak Suka
3. Apakah kamu berminat untuk mengenal dan membaca bahasa Arab setelah memainkan *game* ini ?

   A. Sangat Berminat B. Berminat C. Kurang Berminat D.TidakBerminat
4. Apakah dengan *game* ini dapat membuat pola belajar kamu dalam mempelajari bahasa Arab lebih menarik ?

   A. Sangat Menarik B. Menarik C. Kurang Menarik D.Tidak Menarik
5. Apakah kamu mengerti dengan materi pembelajaran bahasa Arab yang ada dalam *game* ini ?

   A. Sangat Mengerti B. Mengerti C. Kurang Mengerti D. Tidak Mengerti

4.6.2 Pengolahan Data Hasil Kuisioner

Berikut ini adalah hasil pengolahan data dari kuisioner yang telah dibagikan kepada pengguna dengan *range* usia yang sesuai dengan sasaran *user* yaitu :

1. Apakah *game* Edukasi Mengenal dan Membaca Bahasa Arab ini menarik bagi kamu?

Tabel 4.12 Hasil Pengujian Kuisioner Soal Nomor 1

Pertanyaan	**No.**	**Keterangan**	**Responden**	**Prosentase (%)**
1	1	Sangat Menarik	2	6,67
	2	Menarik	27	90
	3	Kurang Menarik	1	3,33
	4	Tidak Menarik	0	0

2. Apakah kamu suka dengan gambar-gambar yang ada didalam *game* ini ?

Tabel 4.13 Hasil Pengujian Kuisioner Soal Nomor 2

Pertanyaan	**No.**	**Keterangan**	**Responden**	**Prosentase (%)**
2	1	Sangat Suka	5	16,67
	2	Suka	24	80
	3	Kurang Suka	1	3,33
	4	Tidak Suka	0	0

3. Apakah kamu berminat untuk mengenal dan membaca bahasa Arab setelah memainkan *game* ini ?

Tabel 4.14 Hasil Pengujian Kuisioner Soal Nomor 3

Pertanyaan	**No.**	**Keterangan**	**Responden**	**Prosentase (%)**
3	1	Sangat Berminat	1	3,33
	2	Berminat	24	80
	3	Kurang Berminat	5	16,67
	4	Tidak Berminat	0	0

4. Apakah dengan *game* ini dapat membuat pola belajar kamu dalam mempelajari bahasa Arab lebih menarik ?

Tabel 4.15 Hasil Pengujian Kuisioner Soal Nomor 4

Pertanyaan	**No.**	**Keterangan**	**Responden**	**Prosentase (%)**
4	1	Sangat Menarik	1	3,33
	2	Menarik	26	86,67
	3	Kurang Menarik	3	10

	4	Tidak Menarik	0	0

5. Apakah kamu mengerti dengan materi pembelajaran bahasa Arab yang ada dalam *game* ini ?

Tabel 4.16 Hasil Pengujian Kuisioner Soal Nomor 5

Pertanyaan	**No.**	**Keterangan**	**Responden**	**Prosentase (%)**
5	1	Sangat Mengerti	1	3,33
	2	Mengerti	27	90
	3	Kurang Mengerti	2	6,67
	4	Tidak Mengerti	0	0

4.6.3 Hasil Pengujian Beta

Berdasarkan hasil jawaban dari kuisioner yang telah diberikan kepada para responden, maka dapat diambil beberapa kesimpulan yaitu :

1. *Game* Edukasi Mengenal dan Membaca Bahasa Arab ini cukup membuat pengguna tertarik untuk mempelajari bahasa Arab.
2. Gambar yang terdapat dalam *game* disukai oleh pengguna.

3. Materi yang pembelajaran bahasa Arab yang terdapat dalam *game* cukup dimengerti oleh pengguna.
4. Membuat pola belajar pengguna dalam mempelajari bahasa Arab cukup menarik.