BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Proses implementasi merupakan tahapan dimana aplikasi yang sudah dibuat dan di rancang lalu di uji kelayakannya. Selanjutnya dioperasikan sebagaimana mestinya sesuai fungsi dan kelayakannya untuk digunakan.

4.1.1 Perangkat Keras Yang Digunakan

Dalam proses pembuatan aplikasi ini, tentunya membutuhkan perangkat keras (*Hardware*) dan perangkat lunak (*software*). Berikut merupakan penjelasan dari perangkat keras dan perangkat lunak yang digunakan.

4.1.1.1 Perangkat Keras (Hardware)

Perangkat keras yang digunakan selama pembangunan aplikasi ini memiliki spesifikasi sebagai berikut:

1. *CPU* : Intel Dual Core 2,8 GHz
2. Hardisk : 250 GB
3. *RAM* : DDR II 2 GB
4. Resolusi Layar : 1024 x 768
5. *Keyboard* : 1 buah
6. *USB Mouse* : 4 Unit

4.1.1.2 Perangkat Lunak (*Software*)

Spesifikasi perangkat lunak yang digunakan untuk menjalankan game edukasi simulasi perlombaan 17 Agustus ini adalah sebagai berikut:

1. Sistem Operasi : Microsoft Windows 7 (Seven)
2. Framework Sistem Operasi : .Net Framework 3.5
3. Microsoft Visual Studio 2010
4. Microsoft Multipoint SDK 1.5.1

4.2 Implementasi Aplikasi

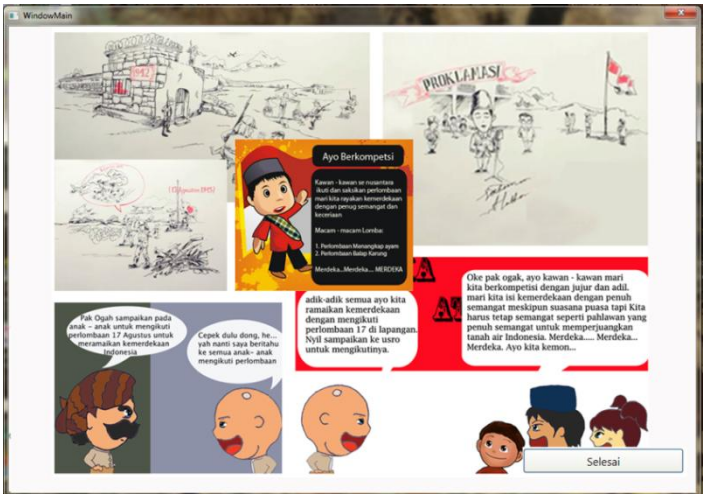
File yang ada pada game edukasi simulasi perlombaaan 17 Agustus ini, dibungkus menjadi satu dalam bentuk *Game* 17 Agustus.zip, berikut langkah-langkah untuk menjalannkan *game* edukasi perlombaan 17 Agustus ini:

1. *Extract* Game17Agustus.rar
2. Akan ada *folder bin/debug*
3. *Double* Klik pada file AgustusGame.exe yang ada di *folder debug*
4. Akan tampil *window game* perlombaan 17 Agustus, dan itu menandakan bahwa *game* sukses dijalankan.

4.3 Implementasi Antar Muka

4.3.1 Tampilan Cerita Game

Layar ini menampilkan pada awal permainan yang menceritakan kejadian kemerdekaan dan kampung di Indonesia yang merayakan perlombaan 17 Agustus.



Gambar 4. 1 Tampilan Intro Game.

4.3.2 Tampilan Utama

Layar ini berfungsi menampilkan Tampilan menu utama atau tampilan awal game dimana didalamnya tersidia empat pilihan, yaitu pilihan petunjuk, main game, tentang dan keluar.



Gambar 4. 2 Tampilan Menu Utama.

4.3.3 Tampilan Petunjuk

Layar ini berfungsi menampilkan informasi mengenai petunjuk cara bermain dalam game Perlombaan Menangkap Ayam dan Balap Karung.



Gambar 4. 3 Tampilan Petunjuk Perlombaan Menangkap Ayam



Gambar 4. 4 Tampilan Petunjuk Perlombaan Balap Karung

4.3.4 Tampilan Pemilihan Main Game

Layar ini berfungsi menampilkan pemilihan perlombaan apa yang dipilih oleh pemain, pilihan bermain terdaapat dua pilihan yaitu game Perlombaan Menangkap Ayam dan Perlombaan Balap Karung.



Gambar 4. 5 Tampilan Pemilihan Main Game

4.3.5 Tampilan Perlombaan Menangkap Ayam

Layar ini berfungsi menampilkan Perlombaan menangkap ayam, masing-masing pemain berkompetisi berebut jawaban yang terdapat pada ayam.



Gambar 4. 6 Tampilan Perlombaan Menangkap Ayam

4.3.6 Tampil Perlombaan Balap Karung

Layar ini berfungsi menampilkan Perlombaan balap karung, masing-masing pemain berkompetisi menjawab pertanyaan untuk menjadi yang terdepan.



Gambar 4. 7 Tampilan Perlombaan Balap Karung.

4.3.7 Tampil Nilai Tertinggi

Layar ini berfungsi menampilkan nilai Tertinggi yang didapatkan pemain setelah menyelesaikan perlombaan .



Gambar 4. 8 Tampilan Nilai Tertinggi

4.3.8 Tampilan Tentang

Layar ini berfungsi menampilkan informasi mengenai ucapan terima kasih dari pembangun aplikasi.



Gambar 4. 9 Tampilan Tentang.

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 Alpha

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

4.4.2 Skenario Pengujian Aplikasi

Skenario pengujian yang akan dilakukan pada aplikasi ini selengkapnya terlihat pada Tabel 4.1

Tabel 4.1 Skenario Pengujian Aplikasi

No	Komponen Yang Diuji	Metode Pengujian
1	Menu Utama	*Blackbox*
2	Petunjuk Permainan	*Blackbox*
3	Menu Main Game	*Blackbox*
4	Perlombaan Menangkap Ayam	*Blackbox*
5	Perlombaan Balap Karung	*Blackbox*
6	Algoritma Fisher-Yates	*White Box*

1. Kasus dan Hasil Pengujian

Berikut ini adalah kasus untuk menguji perangkat lunak yang sudah dibangun menggunakan metode *Black Box* berdasarkan Tabel 4.2 Rencana pengujian yang telah dibuat sebelumnya.

Tabel 4.2 Kasus dan Hasil Uji

No	Kasus yang diuji	Skenario Uji	Hasil Yang Diharapkan	Hasil Pengujian
1.	Menu Utama	Memilih Tombol Petunjuk	Menampilkan Petunjuk yang dipilih	[√] Berhasil [ ] TidakBerhasil
		Memilih Tombol Main Game	Menampilkan Perlombaan yang dipilih	[√] Berhasil [ ] TidakBerhasil
		Memilih Tombol Nilai Tertinggi	Menampilkan Nilai Tertinggi	[√] Berhasil [ ] TidakBerhasil
		Tombol Keluar	Keluar Aplikasi Permainan	[√] Berhasil [ ] TidakBerhasil
2.	Petunjuk Permainan	Memilih Tombol Petunjuk Menangkap Ayam	Menampilkan Petunjuk Menangkap Ayam	[√] Berhasil [ ] TidakBerhasil
		Memilih Tombol Petunjuk Balap Karung	Menampilkan Petunjuk Balap Karung	[√] Berhasil [ ] TidakBerhasil
3.	Menu Main Game	Memilih Tombol Menangkap Ayam	Menampilkan game perlombaan Menangkap Ayam	[√] Berhasil [ ] TidakBerhasil
		Memilih Tombol Balap Karung	Menampilkan game perlombaan Balap Karung	[√] Berhasil [ ] TidakBerhasil
4.	Perlombaan Menangkap Ayam	Sistem melakukan proses inisialisasi mouse yang ada dan menampilkan karakter pada masing-masing pemain	Menampilan banyak pemain dalam game	[√] Berhasil [ ] TidakBerhasil
		Sistem memangil jawaban dari database dan menyisipkan pada setiap ayam	Menampilkan jawaban pada masing-masing ayam	[√] Berhasil [ ] TidakBerhasil
		Sistem memangil soal dari database	Menampilkan soal dari database	[√] Berhasil [ ] TidakBerhasil
		Sistem menghitung waktu mundur	Menampilkan waktu mundur	[√] Berhasil [ ] TidakBerhasil
		Menghitung skor masing - masing pemain	Menampilkan skor masing-masing pemain	[√] Berhasil [ ] TidakBerhasil

5.	Perlombaan Balap Karung	Sistem melakukan proses inisialisasi mouse yang ada dan menampilkan karakter pada masing-masing pemain	Menampilkan banyak pemain pada game	[√] Berhasil [ ] TidakBerhasil
		Jawaban Benar	Menampilkan animasi balap karung maju satu langkah	[√] Berhasil [ ] TidakBerhasil
		Jawaban Salah	Menampilkan animasi balap karung terjatuh	[ ] Berhasil [ ] TidakBerhasil
		Sistem memangil soal dan jawaban dari database	Menampilkan soal dan jawaban masing - masing pemain	[√] Berhasil [ ] TidakBerhasil
		Menghitung skor masing - masing pemain	Menampilkan skor masing-masing pemain	[√] Berhasil [ ] TidakBerhasil

6. Pengujian *White-box* Algoritma *Fisher-Yates*

Pengujian *white-box* atau *white-box testing* merupakan metode perancangan *test case* yang menggunakan struktur kontrol dari perancangan prosedural dalam mendapatkan *test case*. Adapun metode yang digunakan dalam pengujian *white-box* ini 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 .

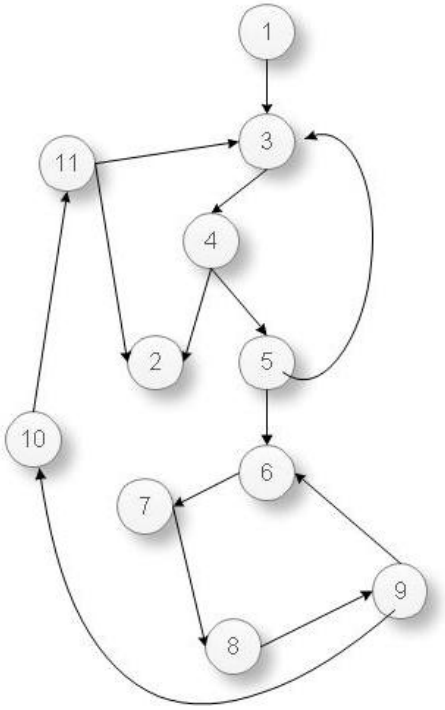
Pengujian *White-box* acak soal dengan algoritma Fisher-Yates

a. Berikut ini *pseudocode* algoritma Fisher – Yates

```
1:  {Input: masukan bilangan 1... n : integer}
2:  {Output: menampilkan variabel acak 1..n}
  Kamus:
  i , j,k, temp,a : integer
3:        for i from 0 to n − 1 do
```

```
4:          a[i] := i
5:            if (n > 1) then
6:              for k from n − 1 down to 1 do
7:                  j := rand(0..k)
8:                   a ← temp(a, r, k)
9:                return(a)
10:           end
```

b. Flow Graph algoritma Fisher Yates



Gambar 4.10 Flow graph algoritma fisher yates

c. Dari flow graph, maka dihitung *cyclomatic complexity*, yakni:

V(G) = E – N + 2

V(G) = 14 – 11 + 2

V(G) =3 + 2

V(G) = 5

Tabel 4.3. Keterangan *flow graph*

Simbol Flow Graph	**Keterangan**
◯	*Node (N)*
→	*Busur (E)*

d. Dari hasil perhitungan cyclomatic complexity terdapat 4 independent path yaitu:

Path 1 = 1 – 3– 4 – 5 – 3

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

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

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

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

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.

e. Adapun graph matrik pada flowgraph Fisher Yates adalah sebagai berikut:

Tabel 4.4. *Graph* matriks.

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

Kompleksitas Siklomatik	Sum(E)+1=	4+1=5

Jadi, pada teknik *flow chart*, jumlah array, *Independent Path* yang dihasilkan bernilai sama.

4.4.3 Pengujian Beta

4.4.3.1 Evaluasi dari Pengguna

Pengujian beta merupakan pengujian yang dilakukan secara objektif, dimana dilakukan pengujian secara langsung terhadap pengguna dengan menggunakan kuisioner mengenai kepuasan pengguna atas aplikasi yang telah dibangun. Adapun metode penilaian pengujian yang digunakan adalah metode kuantitatif berdasarkan data dari pengguna. Pengujian dilakukan pada 10 siswa untuk *game* perlombaan menangkap ayam untuk siswa kelas 1 s.d 4 SD dan 10 siswa untuk *game* perlombaan balap karung untuk siswa kelas 7 s.d 8 SMP diberikan 4 sampel

4.4.3.2 Pertanyaan Pengguna

1. Menurut adik-adik, apakah game edukasi ini dapat memberikan wawasan dan mampu mengenalkan tentang permainan perlombaan 17 Agustus yang ada di Indonesia ?
    a. Sangat setuju
    b. Setuju
    c. Cukup
    d. Kurang Setuju

e. Sangat Kurang Setuju

2. Menurut adik-adik, apakah game edukasi perlombaan 17 Agustus ini dapat meningkatkan kreativitas?
   a. Sangat Setuju
   b. Setuju
   c. Cukup
   d. Kurang Setuju
   e. Sangat Kurang Setuju
3. Menurut adik-adik, apakah game edukasi perlombaan 17 Agustus ini sangat menghibur?
   a. Sangat Kurang
   b. Menghibur
   c. Cukup
   d. Kurang
   e. Sangat Kurang
4. Menurut adik-adik, apakah game edukasi perlombaan 17 Agustus ini sudah pantas untuk dijadikan sebagai media kompetisi secara jujur dan sportif ?
   a. Sangat Setuju
   b. Setuju
   c. Cukup
   d. Kurang

e. Sangat Kurang

4.4.3.3 Rumus Pengujian Beta

Berdasarkan data hasil kuisioner tersebut, dicari prosentase masing-masingjawaban dengan menggunakan rumus R=P/Q*100%, dimana :

P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

R = Nilai prosentase

4.4.3.4 Hasil Kuesioner

1. Menurut adik-adik, apakah game edukasi ini dapat memberikan wawasan dan mampu mengenalkan tentang permainan perlombaan 17 Agustus yang ada di Indonesia ?

Tabel 4.5 Hasil Kuisioner Memberikan wawasan

Kategori Jawaban	Sangat Kurang Setuju	Kurang Setuju	Cukup	Setuju	Sangat setuju
Frekuensi Jawaban	-	-	-	10	10
Persentase Nilai	-	-	-	50%	50%

2. Menurut adik-adik, apakah game edukasi perlombaan 17 Agustus ini dapat meningkatkan kreativitas?

Tabel 4.6 Hasil Kuisioner Meningkatkan Kreatifitas

Kategori Jawaban	Sangat Kurang	Kurang	Cukup	Bermanfaat	Sangat Setuju
Frekuensi Jawaban	-	-	2	12	6
Persentase Nilai	-	-	10 %	60 %	30

3. Menurut adik-adik, apakah game edukasi perlombaan 17 Agustus ini sangat menghibur?

Tabel 4.7 Hasil Kuisioner Menghibur

Kategori Jawaban	Sangat Kurang	Kurang	Cukup	Menghibur	Sangat setuju
Frekuensi Jawaban	-	-	1	4	15
Persentase Nilai	-	-	5 %	20 %	75%

4. Menurut adik-adik, apakah game edukasi perlombaan 17 Agustus ini sudah pantas untuk dijadikan sebagai media kompetisi secara jujur dan sportif ?

Tabel 4.8 Hasil Kuisioner Media Kompetisi

Kategori Jawaban	Sangat Kurang	Kurang	Cukup	Setuju	Sangat setuju
Frekuensi Jawaban	-	-	-	5	15
Persentase Nilai	-	-	-	20 %	75%

Berdasarkan hasil dari pengujian beta dapat disimpulkan bahwa game yang telah dibangun dapat mengenalkan perlombaan 17 Agustus sebagai media untuk berkompetisi secara sportif dan jujur dengan memanfaatkan satu komputer desktop saja.