

BAB III

ANALISA DAN RANCANGAN *SOFTWARE*

3.1 Analisa Kebutuhan *Software*

Dalam kebutuhan *software* pertama yang harus dilakukan adalah menganalisa berbagai kebutuhan yang diperlukan dalam rancangan *software* untuk dijadikan sebagai bahan masukan atau acuan dalam pengembangan *software* tersebut. Kebutuhan-kebutuhan ini diperlukan sebagai input yang diperlukan, output yang diharapkan dan pembuatan *design user interface* yang mudah dipahami dan mudah digunakan oleh pengguna atau *user*.

3.1.1. Identifikasi Masalah

Permasalahan yang akan dibahas pada penyusunan skripsi ini adalah pembuatan aplikasi mobile dengan sistem operasi android untuk aplikasi pembelajaran *housekeeping*.

3.1.2. Analisa Kebutuhan

Kebutuhan yang akan di analisa mencakup perangkat lunak dan perangkat keras yang digunakan.

1. Kebutuhan perangkat lunak yang digunakan penulis.

Perangkat lunak yang digunakan oleh penulis untuk pengembangan aplikasi ini, sebagai berikut :

MIT App Inventor 2

App Inventor adalah sistem perangkat lunak untuk membuat aplikasi pada perangkat Android. Uniknya, App Inventor dibuat tidak seperti sistem pengembangan aplikasi biasa, di mana seorang programmer harus menuliskan baris - baris kode program, melainkan dengan interaksi visual berbasis grafis. Dalam kaitan ini, inventor dapat disebut sebagai sistem terpadu untuk pengembangan aplikasi berbasis blok - blok grafis (dalam istilah asing: *blocks language*).

2. Kebutuhan perangkat keras yang digunakan penulis.

a. Laptop, dengan spesifikasi :

- | | |
|---------------------|---|
| 1. <i>Processor</i> | : Intel(R) Core(TM) i3-2348M
CPU @ 2.30GHz (4 CPUs), ~2.3GHz |
| 2. <i>Ram</i> | : 6 GB |
| 3. <i>Graphic</i> | : AMD Radeon HD 7610M |
| 4. <i>Harddisk</i> | : 500 GB |
| 5. <i>OS</i> | : Windows 7 64-Bit |

b. *Smart Phone*, dengan spesifikasi :

- | | |
|---------------------|--|
| 1. <i>Processor</i> | : Quad-core 1.5 GHz Cortex-A53 &
Quad-core 1.2 GHz Cortex-A53 |
|---------------------|--|

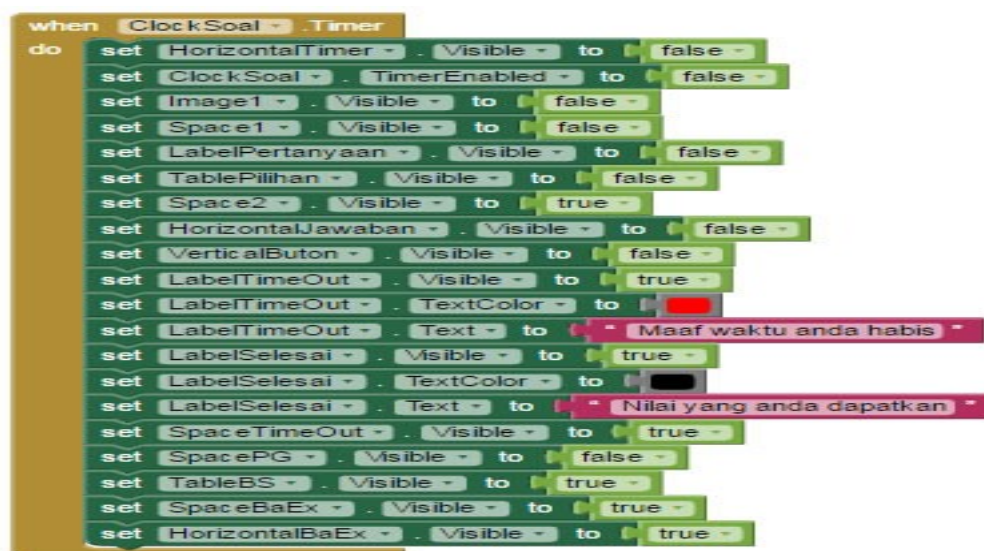
- 2. *Memory* : 3GB
- 3. *Dimensi* : 139.3 x 69.6 x 8.5 mm
- 4. *OS* : Android OS, v5.1.1 (Lollipop)

3.2 Desain

3.2.1. Rancangan Algoritma Pada kasus

Algoritma Boolean Pada Halaman Quiz

1. Hasil Nilai

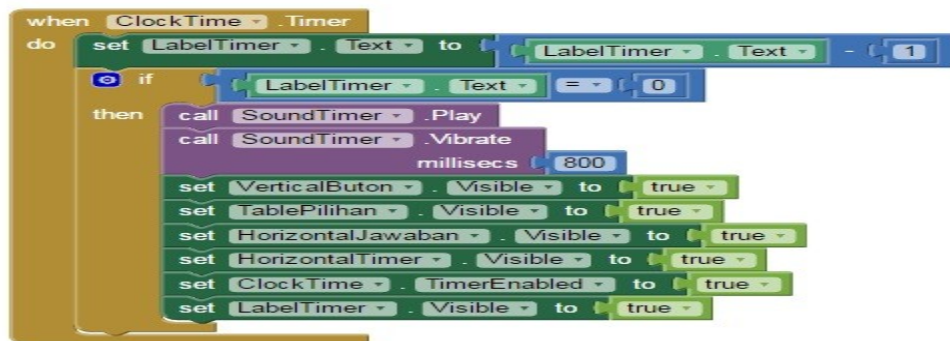


Sumber : Pribadi

Gambar III.1 Blok ClockSoal

- a. Blok ClockSoal adalah perintah waktu yang digunakan untuk menunjukkan atau menampilkan hasil nilai dengan disertakan teks "Maaf waktu anda habis" dan "Nilai yang anda dapatkan".

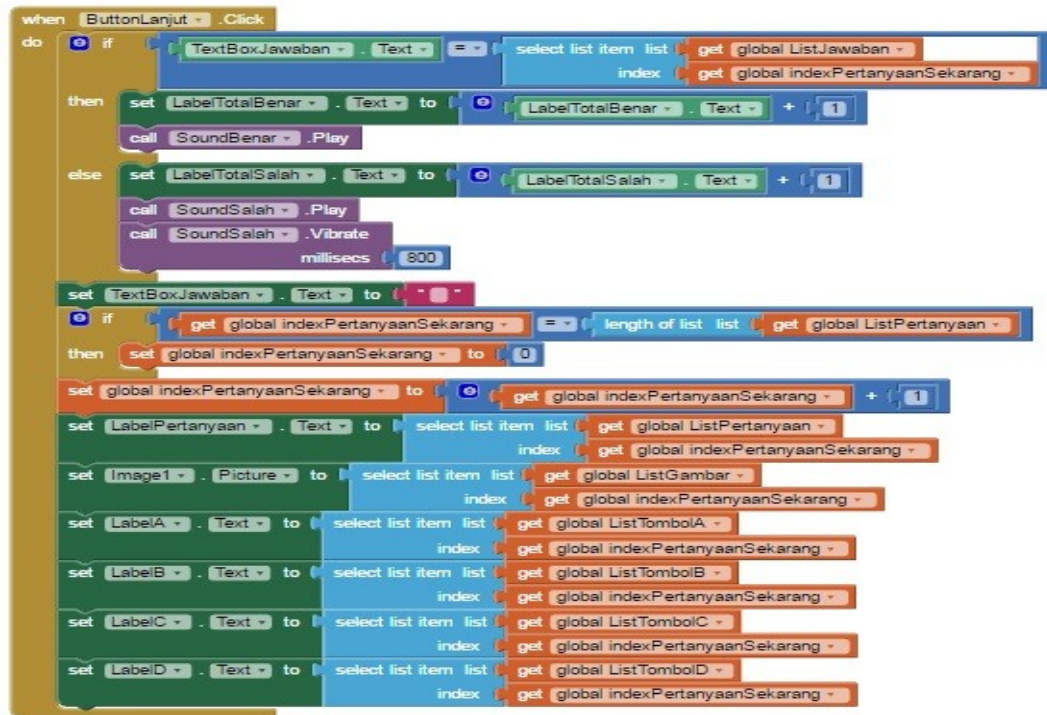
- b. Blok yang disertai dengan *visible* lalu diperintahkan dengan blok *false*, maka akan disembunyikan atau tidak terlihat. Begitupun dengan sebaliknya, blok yang disertai dengan *visible* diperintahkan dengan blok *true* maka akan ditampilkan.



Sumber : Pribadi

Gambar III.2 Blok ClokTime

- a. Blok ClockTime adalah perintah waktu untuk menghitung mundur LabelTimer, dan memutar suara yang menandakan waktu habis.
- b. Blok yang disertai dengan *visible* lalu diperintahkan dengan blok *false*, maka akan disembunyikan atau tidak terlihat. Begitupun dengan sebaliknya, blok yang disertai dengan *visible* diperintahkan dengan blok *true* maka akan ditampilkan.

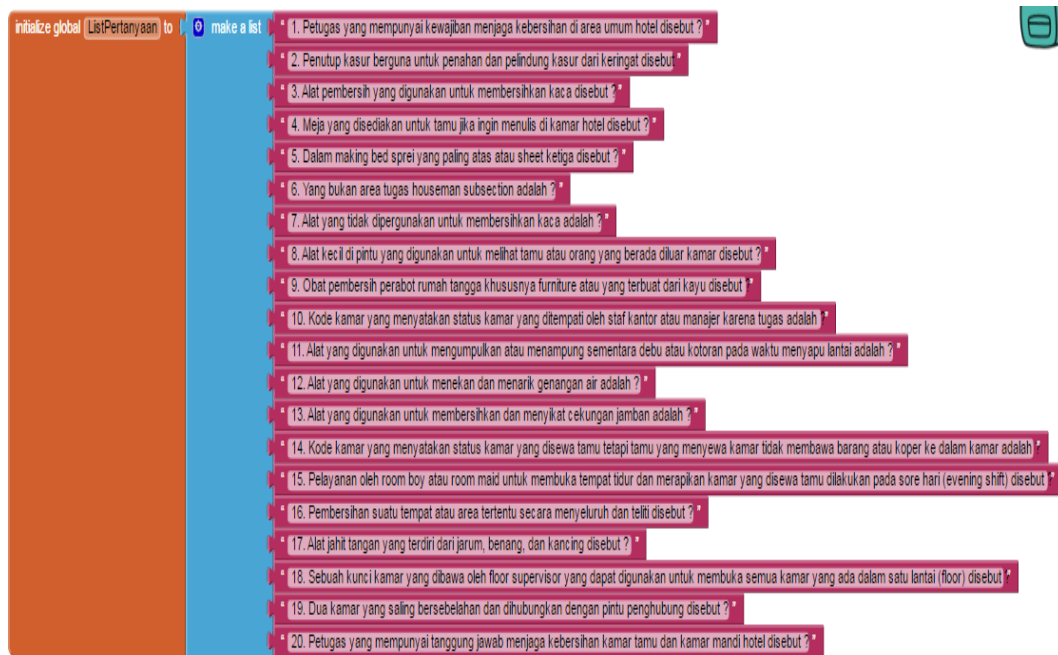


Sumber : Pribadi

Gambar III.3 Blok ButtonLanjut

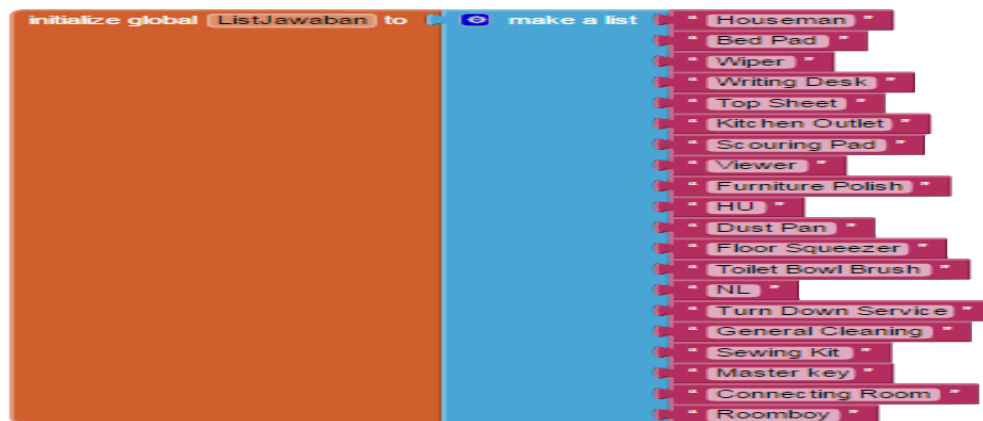
- a. Blok ButtonLanjut adalah perintah yang berfungsi untuk melakukan perhitungan nilai yang didapat user, memutar suara jika salah atau benar, melanjutkan pertanyaan dan jawaban berikutnya.

2. Pertanyaan



Gambar III.4 Blok initialize global ListPertanyaan

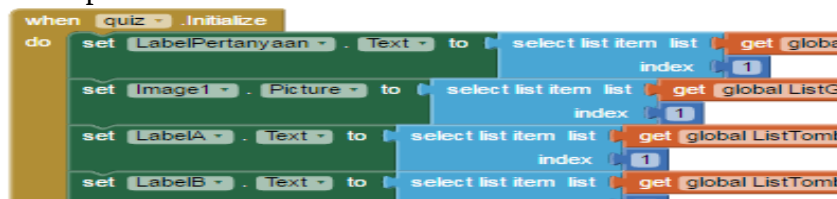
- a. Blok initialize global ListPertanyaan adalah blok yang berisikan *list* sebagai pertanyaan yang dapat diperintahkan menampilkan *text* secara terurut.



Gambar III.5 Blok initialize global ListJawaban

- a. Blok initialize global ListJawaban adalah blok yang berisikan *list* sebagai jawaban yang benar secara terurut.

3. Tampilan



Sumber : Pribadi

Gambar III.6
Blok Quiz

- Blok *quiz* adalah blok perintah untuk menampilkan dan menjalankan LabelPertanyaan, Image1, LabelA, LabelB, LabelC, LabelD.
- Blok ClockSoal diperintahkan untuk tetap berjalan dengan perintah TimerEnabled (*True*). Sedangkan blok TableBS, HorizontalBaEx, dan LabelTimeOut tidak diperlihatkan.

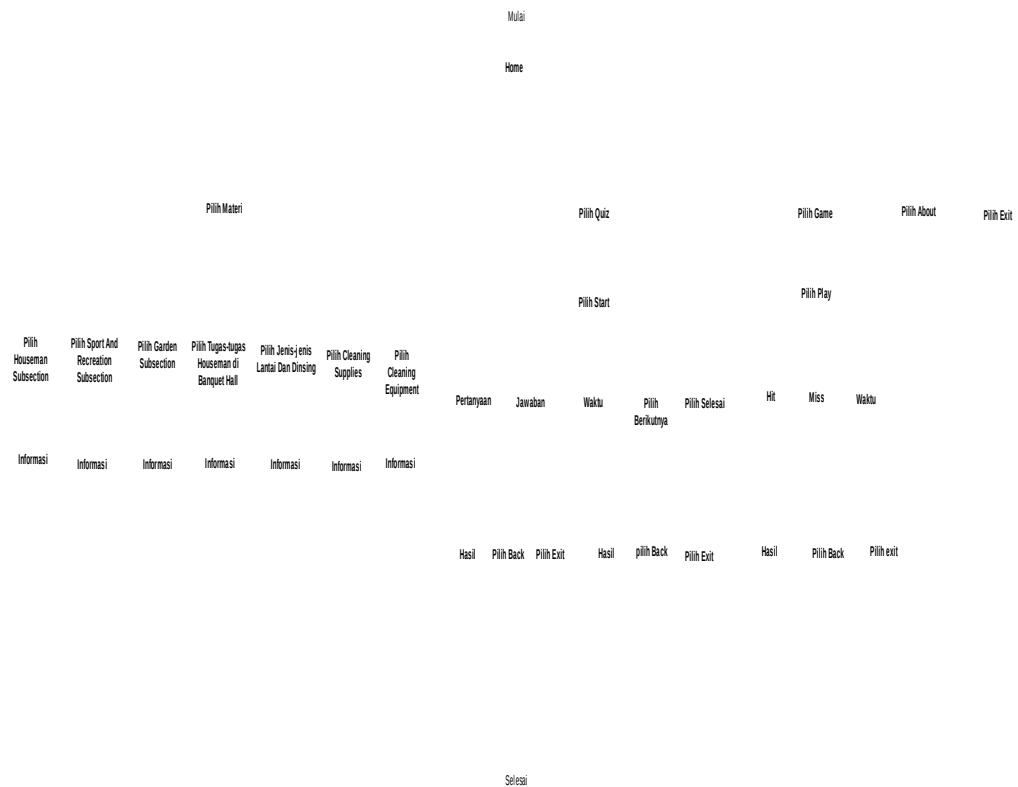
3.2.2. SoftwareArchitecture

1. Pseudocode

Mempermudah manusia dalam pemahaman dibandingkan menggunakan bahasa pemrograman yang umum digunakan, terlebih aspeknya yang ringkas serta tidak bergantung pada suatu sistem tertentu merupakan prinsip utama dalam suatu algoritma.

- Diagram *Use Case*

act Activity Diag...



Sumber : Pribadi

Gambar III.8 Diagram Activity Belajar Housekeeping

c. Sequence Diagram Materi Housekeeping

d. *Sequence Diagram Quiz*

sd Use Case Mo...



Sumber : Pribadi

Gambar III.10 *Sequence Diagram Quiz*

e. *Sequence Diagram Game*

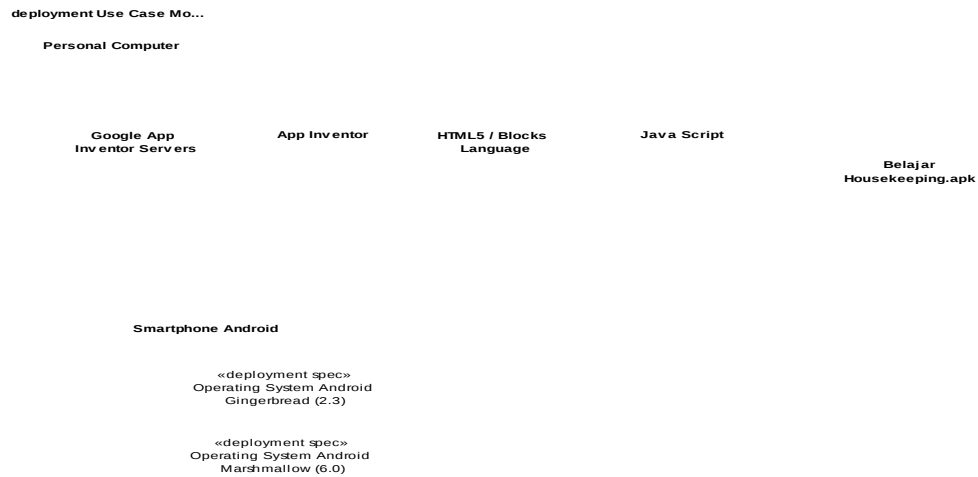
sd Use Case Mo...



Sumber : Pribadi

Gambar III.11 Sequence Diagram Game

f. Deployment Diagram

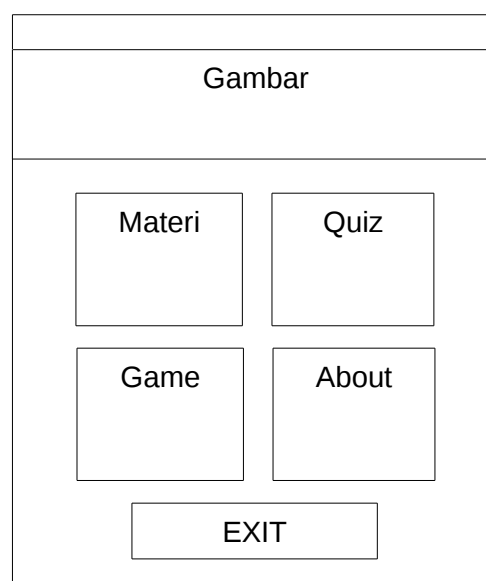


Sumber : Pribadi

Gambar III.12 Deployment Diagram Belajar Housekeeping

3.2.3. User Interface

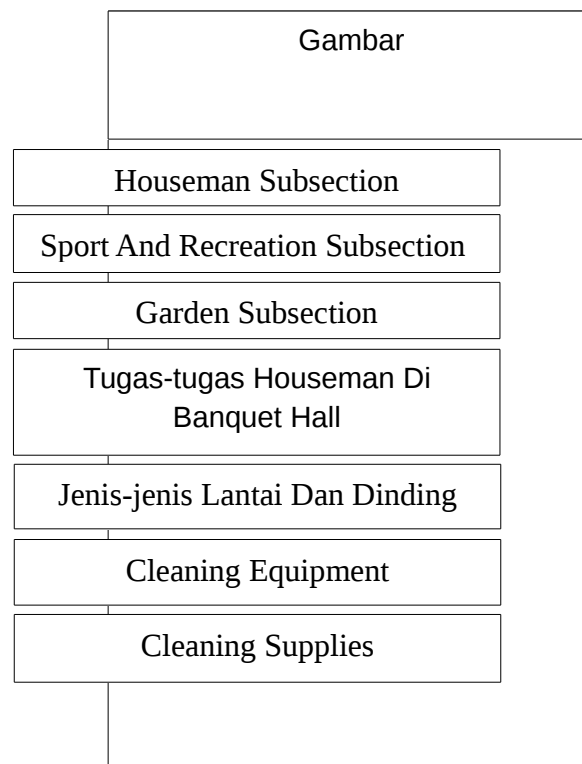
Desain *user interface* pada aplikasi ini meliputi, antara lain :



Sumber : Pribadi

Gambar III.13 User Interface Menu Utama

- a. Jika dipilih tombol materi maka akan menampilkan menu informasi *housekeeping*.
- b. Jika dipilih tombol *quiz* akan menampilkan halaman depan *quiz*.
- c. Jika dipilih tombol *game* akan menampilkan halaman depan *game*.
- d. Jika dipilih tombol *about* akan menampilkan informasi tentang pembuat aplikasi ini.
- e. Jika di pilih tombol *exit* akan keluar dari aplikasi.



Sumber : Pribadi

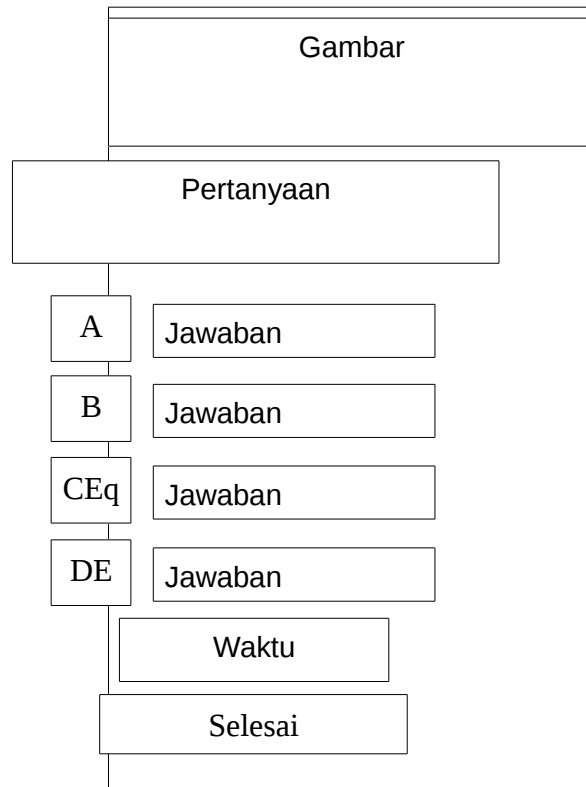
Gambar III.14 User Interface Pada Halaman Materi

- a. Jika dipilih tombol *houseman subsection* maka akan menampilkan informasi tentang *houseman subsection*.
- b. Jika dipilih tombol *sport and recreation subsection* maka akan menampilkan informasi tentang *sport and recreation subsection*.
- c. Jika dipilih tombol *garden subsection* maka akan menampilkan informasi tentang *garden subsection*.
- d. Jika dipilih tombol tugas-tugas *houseman* di *banquet hall* maka akan menampilkan informasi tentang tugas-tugas *houseman* di *banquet hall*.
- e. Jika dipilih tombol jenis-jenis lantai dan dinding maka akan menampilkan informasi tentang jenis-jenis lantai dan dinding.
- f. Jika dipilih tombol *cleaning equipment* maka akan menampilkan informasi tentang *cleaning equipment*.
- g. Jika dipilih tombol *cleaning supplies* maka akan menampilkan informasi tentang *cleaning* *supplies*.

Gambar
Ketentuan Quiz
START

Gambar III.15 User Interface Pada Halaman Depan Quiz

- Halaman depan *quiz* berisi beberapa ketentuan menjawab *quiz*.
- Jika dipilih tombol *START* maka akan menampilkan halaman *quiz*.
- Ketentuan *quiz* adalah ketentuan untuk menjawab pertanyaan.



Gambar III.16 user interface Pada Halaman Quiz

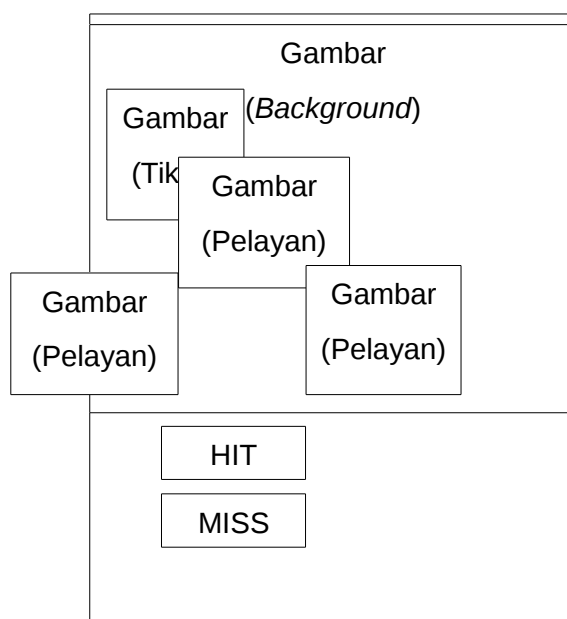
- Jika dipilih tombol A atau B atau C atau D akan menampilkan ke pertanyaan dan jawaban berikutnya.
- Jika dipilih tombol selesai maka akan mengakhiri pertanyaan dari *quiz* dan menampilkan Hasil Nilai yang didapat *user*.
- Jika waktunya habis maka akan mengakhiri pertanyaan dari *quiz* dan menampilkan Hasil Nilai yang didapat *user*.



Sumber : Pribadi

Gambar III.17 User Interface Pada Halaman Depan Game

- Halaman depan *game* berisi beberapa ketentuan bermain *game*.
- Jika dipilih tombol *PLAY* maka akan menampilkan halaman *game*.
- Ketentuan *game* adalah ketentuan untuk bermain *hit mouse*.



Waktu

Sumber : Pribadi

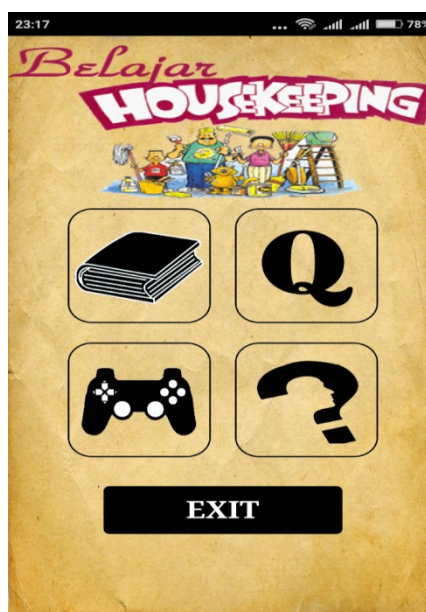
Gambar III.18 User Interface Pada Halaman Game

- a. Jika Gambar (Pelayan) ditekan maka akan menambah *point MISS* sebanyak 2.
- b. Jika Gambar (*Background*) ditekan maka akan menambah *point MISS* sebanyak 1.
- c. Jika Gambar (Tikus) ditekan maka akan menambah *point HIT* sebanyak 1.

3.3. Implementasi

Implementasi merupakan salah satu tahap dalam pengembangan sebuah perangkat lunak. Implementasi aplikasi *Belajar Housekeeping* pada aplikasi mobile android menggunakan bahasa pemrograman *Block Language* pada *compile MIT App Inventor*.

1. Tampilan Menu Utama



Sumber : Pribadi

Gambar III.19 Tampilan Menu Utama

2. Tampilan Materi



Sumber : Pribadi

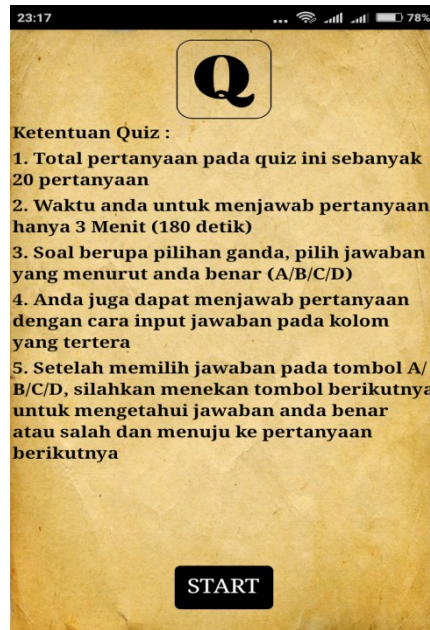
Gambar III.20 Tampilan Materi



Sumber : Pribadi

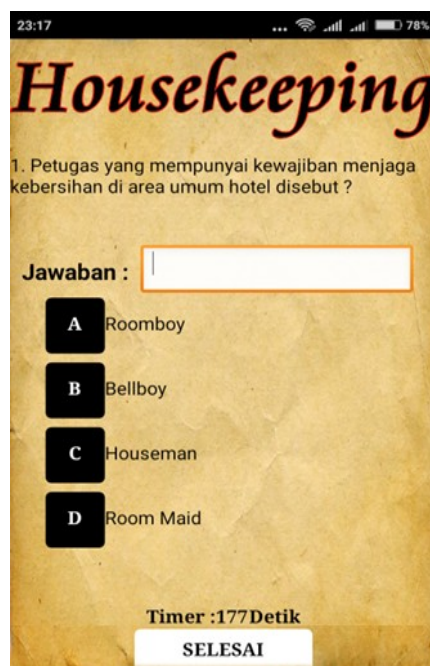
Gambar III.21 Tampilan Informasi Yang Terdapat Pada Menu Materi

3. Tampilan Quiz



Sumber : Pribadi

Gambar III.22 Tampilan Halaman Depan Quiz



Sumber : Pribadi

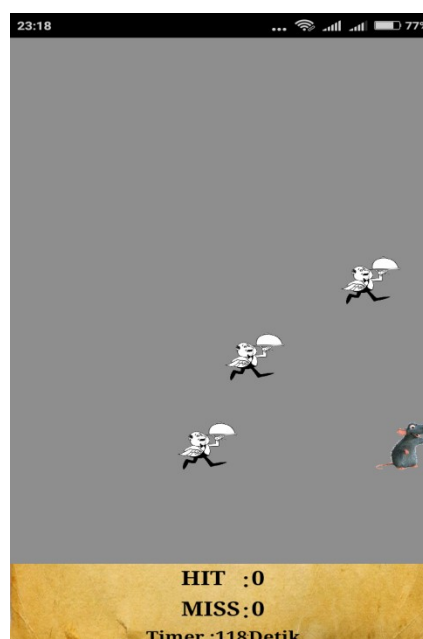
Gambar III.23 Tampilan Halaman Quiz

4. Tampilan *Game*



Sumber : Pribadi

Gambar III.24 Tampilan Halaman Depan *Game*



Sumber : Pribadi

Gambar III.25 Tampilan Halaman *Game*

5. Tampilan *About*



Sumber : Pribadi

Gambar III.26 Tampilan *About*

3.4 *Testing*

Testing adalah proses pengujian suatu program untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan (*error* atau *bugs*) untuk mengetahui kesiapan sebuah program dalam melakukan *input* data, proses pengolahan data dan *output* dari data yang diharapkan. Pengujian *testing* disini penulis menggunakan metode *black box testing* dan *white box testing*.

3.4.1. Pengujian *Black Box*

Black Box Testing adalah metode pengujian perangkat lunak untuk menunjukkan cara beroperasinya, walaupun dirancang untuk menemukan kesalahan, pengujian *black box* digunakan untuk mendemonstrasikan fungsi *software* yang di operasikan apakah *input* data, proses data, dan *output* data sesuai apa yang diharapkan.

Langkah selanjutnya adalah melakukan pengecekan terhadap fungsi logika pada aplikasi ini, dalam hal ini diambil *sample* fungsi logika untuk aplikasi pembelajaran *housekeeping*. Untuk mengecek apakah terdapat kesalahan logika dalam aplikasi ini, maka dibuat *table* percobaan sebagai berikut :

Table III.1 *Black Box* Testing Menu Utama

Data Masukan	Data yang di Harapkan	Pengamatan	Kesimpulan
Klik tombol materi	Menampilkan halaman menu materi	Halaman menu materi tampil	Valid
Klik tombol <i>quiz</i>	Menampilkan halaman depan <i>quiz</i>	Halaman depan <i>quiz</i> tampil	Valid
Klik tombol <i>game</i>	Menampilkan halaman depan <i>game</i>	Halaman depan <i>gametampil</i>	Valid
Klik tombol <i>about</i>	Menampilkan halaman <i>about</i>	Halaman <i>about</i> tampil	Valid
Klik tombol <i>exit</i>	Keluar dari aplikasi	Aplikasi berhasil keluar	Valid

Sumber : Pribadi

Table III.2 *Black Box Testing* Menu Materi

Data Masukan	Data yang di Harapkan	Pengamatan	Kesimpulan
Klik tombol <i>houseman</i> <i>subsection</i>	Menampilkan informasi <i>houseman</i> <i>subsection</i>	Informasi <i>houseman</i> <i>subsection</i> tampil	Valid
Klik tombol <i>sport</i> <i>and recreation</i> <i>subsection</i>	Menampilkan informasi <i>sport and</i> <i>recreation subsection</i>	Informasi <i>sport and</i> <i>recreation subsection</i> tampil	Valid
Klik tombol <i>garden subsection</i>	Menampilkan informasi <i>garden</i> <i>subsection</i>	Informasi <i>garden</i> <i>subsection</i> tampil	Valid
Klik tombol tugas- tugas <i>houseman</i> di <i>banquet hall</i>	Menampilkan informasi tombol tugas-tugas <i>houseman</i> di <i>banquet hall</i>	Informasi tombol tugas-tugas <i>houseman</i> di <i>banquet hall</i> tampil	Valid
Klik tombol jenis- jenis lantai dan dinding	Menampilkan informasi jenis-jenis lantai dan dinding	Informasi jenis-jenis lantai dan dinding tampil	Valid
Klik tombol cleaning equipment	Menampilkan <i>cleaning equipment</i> informasi	Informasi <i>cleaning</i> <i>equipment</i> tampil	Valid
Klik tombol	Menampilkan <i>cleaning</i>	Informasi <i>cleaning</i> <i>supplies</i> tampil	Valid

cleaning supplies *suppliesinformasi*

Sumber : Pribadi

Table III.3 *Black Box TestingQuiz*

Data Masukan	Data yang di Harapkan	Pengamatan	Kesimpulan
Klik tombol <i>start</i>	Tampil <i>layout quiz</i>	<i>Layout quiz</i> tampil	<i>Valid</i>
Klik tombol jawaban (A/B/C/D)	Tampil jawaban ke kolom jawaban	Jawaban ke kolom jawaban tampil	<i>Valid</i>
Klik tombol berikutnya	Tampil soal dan jawaban berikutnya	Soal dan jawaban berikutnya tampil	<i>Valid</i>
Klik tombol selesai	Tampil hasil nilai	Hasil nilai tampil	<i>Valid</i>

Sumber : Pribadi

Table III.4*Black Box TestingGame*

Data Masukan	Data yang di Harapkan	Pengamatan	Kesimpulan
Klik tombol <i>play</i>	Tampil <i>layout game</i>	<i>Layout game</i> tampil	<i>Valid</i>
Klik gambar (tikus)	Menambah 1 point untuk <i>hit</i>	Menambah 1 point untuk <i>hit</i> tampil	<i>Valid</i>
Klik gambar (background)	Menambah 1 point untuk <i>miss</i>	Menambah 1 point untuk <i>miss</i> tampil	<i>Valid</i>
Klik gambar (pelayan)	Menambah 2 point untuk <i>miss</i>	Menambah 2 point untuk <i>miss</i> tampil	<i>Valid</i>

Sumber : Pribadi

Pada *table* di atas menunjukkan hasil pada aplikasi pembelajaran *housekeeping* ini sesuai dengan hasil yang diharapkan. Maka dari itu dapat dipastikan jika aplikasi pembelajaran *housekeeping* ini tidak ditemukan *logic error* atau kesalahan logika. Jika ada unit yang sesuai dengan *outputnya* maka untuk menyelesaikannya, diteruskan pada pengujian kedua, yaitu *white box testing*.

3.4.2. Pengujian *White Box Testing*

White Box Testing berarti pengujian terhadap perhitungan yang dihasilkan oleh aplikasi yang sudah dibuat didasarkan pada pemeriksaan detail *procedural*, apakah terjadi kesalahan atau kekurangan dalam perhitungan. Hal ini dapat dijadikan bahan evaluasi untuk memperbaiki kesalahan–kesalahan pada perangkat lunak yang sudah dibangun. Testing menggunakan *white box*.

Cyclomatic Complexity

Cyclomatic Complexity merupakan suatu sistem pengukuran yang menyediakan ukuran kuantitatif dari kompleksitas logika suatu program. Pada Basis *PathTesting*, hasil dari cyclomatic complexity digunakan untuk menentukan banyaknya *independentpaths*. *Independentpath* adalah sebuah kondisi pada program yang menghubungkan *node* awal dengan *node* akhir.

Terdapat 2 persamaan yang digunakan, yaitu :

$$V(G) = E - N + 2 \text{ atau } V(G) = P + 1$$

Keterangan :

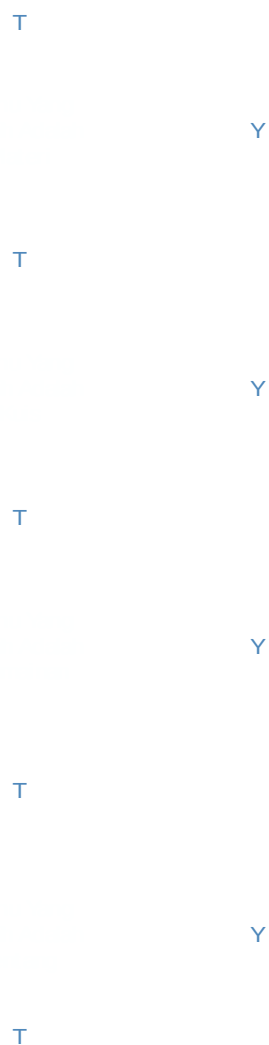
$$V(G) = \text{Cyclomatic Complexity}$$

E = Jumlah *Edge* (panah)

N = Jumlah *Node* (lingkaran)

P = Jumlah *Predicate Node*

1. Menu Utama



Sumber : Pribadi

Gambar III.27 Flowchart Bagan Alir Menu Utama



Sumber : Pribadi

Gambar III.28 Flowchart Grafik Alir Menu Utama

Dari gambar III.27 dan III.28 dapat dihitung cyclomatic complexity sebagai berikut :

$$V(G) = E - N + 2$$

$$V(G) = 14 - 11 + 2$$

$$V(G) = 3 + 2$$

$$V(G) = 5$$

Jadi, cyclomatic complexity untuk gambar III.27 dan III.28 adalah 5. Berdasarkan cyclomatic complexity tersebut, maka terdapat 5 path yang terdiri dari :

Path 1 : 1,2,3,4,5,6,7

Path 2 : 1,2,3,8,2,3,4,5,6,7

Path 3 : 1,2,3,4,9,2,3,4,5,6,7

Path 4 : 1,2,3,4,5,10,2,3,4,5,6,7

Path 5 : 1,2,3,4,5,6,11,2,3,4,5,6,7

2. Menu Materi



Sumber : Pribadi

Gambar III.29 Flowchart Bagan Menu Materi



Sumber : Pribadi

Gambar III.30 Flowchart Grafik Alir Menu Materi

Dari gambar III.29 dan III.30 dapat dihitung cyclomatic complexity sebagai berikut :

$$V(G) = E - N + 2$$

$$V(G) = 23 - 17 + 2$$

$$V(G) = 6 + 2$$

$$V(G) = 8$$

Jadi, cyclomatic complexity untuk gambar III.29 dan III.30 adalah 8. Berdasarkan cyclomatic complexity tersebut, maka terdapat 8 path yang terdiri dari :

Path 1 : 1,2,3,4,5,6,7,8,9,10

Path 2 : 1,2,3,11,2,3,4,5,6,7,8,9,10

Path 3 : 1,2,3,4,12,2,3,4,5,6,7,8,9,10

Path 4 : 1,2,3,4,5,13,2,3,4,5,6,7,8,9,10

Path 5 : 1,2,3,4,5,6,14,2,3,4,5,6,7,8,9,10

Path 6 : 1,2,3,4,5,6,7,15,2,3,4,5,6,7,8,9,10

Path 7 : 1,2,3,4,5,6,7,8,16,2,3,4,5,6,7,8,9,10

Path 8 : 1,2,3,4,5,6,7,8,9,17,2,3,4,5,6,7,8,9,10

3. Quiz

Y

T

Sumber : Pribadi

Gambar III.31 Flowchart Bagan Quiz

Sumber : Pribadi

Gambar III.32 Flowchart Grafik Alir Quiz

Dari gambar III.31 dan III.32 dapat dihitung cyclomatic complexity sebagai berikut :

$$V(G) = E - N + 2$$

$$V(G) = 7 - 7 + 2$$

$$V(G) = 2$$

Jadi, cyclomatic complexity untuk gambar III.31 dan III.32 adalah 3. Berdasarkan cyclomatic complexity tersebut, maka terdapat 3 path yang terdiri dari :

Path 1 : 1,2,3,4,5,6

Path 2 : 1,2,3,4,5,7,2,3,4,5,6

4. *Game*



Sumber : Pribadi

Gambar III.33 Flowchart Bagan *Game*



Sumber : Pribadi

Gambar III.34 Flowchart Grafik Alir *Game*

Dari gambar III.33 dan III.34 dapat dihitung cyclomatic complexity sebagai berikut :

$$V(G) = E - N + 2$$

$$V(G) = 13 - 11 + 2$$

$$V(G) = 4$$

Jadi, cyclomatic complexity untuk gambar III.33 dan III.34 adalah 4. Berdasarkan cyclomatic complexity tersebut, maka terdapat 4 path yang terdiri dari :

Path 1 : 1,2,3,4,5,6,7

Path 2 : 1,2,3,4,8,11,2,3,4,5,6,7

Path 3 : 1,2,3,4,5,9,11,2,3,4,5,6,7

Path 4 : 1,2,3,4,5,6,10,11,2,3,4,5,6,7

3.5. Support

Dalam bagian ini merupakan perangkat yang digunakan untuk mendukung pemasangan aplikasi yang telah diujikan. Perangkat pendukung yang digunakan pada aplikasi ini menggunakan sebuah smartphone berbasis sistem operasi android.

Tabel III.2 Support

	Kebutuhan	Keterangan
Tipe		Xiaomi Redmi 3 Pro
Sistem Operasi		Android Lollipop (5.1)
Processor		Quad-core 1.5 GHz Cortex-A53 &
		Quad-core 1.2 GHz Cortex-A53
RAM		3 GB