

## **BAB III**

### **ANALISIS MASALAH DAN PERANCANGAN PROGRAM**

#### **III.1. Analisis Masalah**

Perkembangan ilmu pengetahuan dan teknologi saat ini telah membawa manusia kepada peradaban yang lebih baik. Banyak manfaat dan kemudahan yang telah dihasilkan dengan adanya perkembangan teknologi terlebih lagi dengan lahirnya komputer, maka kualitas dan efektifitas dalam bekerja semakin meningkat. Fakta menunjukkan bahwa manusia tidak mungkin bisa terlepas dari alat bantu atau aplikasi yang membantu manusia dalam menyelesaikan berbagai pekerjaan mereka, sebab komputer menawarkan kemudahan-kemudahan melalui *software*, misalnya *game* yang membantu manusia menghilangkan kejenuhan akibat dari kegiatan yang padat.

Dengan demikian, penulis berharap *game* ini di harapkan dapat menghilangkan kejenuhan serta melatih kecepatan repleks dan koordinasi tangan dan mata dan kreatifitas dalam berfikir. Dalam tahap analisis ini, penulis membagi kebutuhan sistem kedalam dua jenis yaitu, kebutuhan fungsional dan kebutuhan nonfungsional.

##### **III.1.1. Analisis Kebutuhan Fungsional**

Kebutuhan fungsional adalah jenis kebutuhan yang berisi proses-proses apa saja yang nantinya dilakukan oleh aplikasi. Kebutuhan fungsional juga berisi informasi-informasi apa saja yang harus ada dan dihasilkan oleh aplikasi. Berikut kebutuhan fungsional yang terdapat pada aplikasi yang dirancang.

1. Pada *game* terdapat fasilitas bagi pengguna untuk memilih dan mengatur beberapa komponen pada aplikasi.
2. *Game* harus dapat melakukan pendataan pengguna (*user*) aplikasi.
3. *Game* harus dapat berinteraksi baik dengan pengguna (*user*).

### III.1.2. Analisis Kebutuhan NonFungsional

Kebutuhan ini adalah tipe kebutuhan yang berisikan properti yang dimiliki oleh aplikasi *game*. Berikut adalah kebutuhan nonfungsional yang dimiliki aplikasi.

1. Operasional
  - a. Digunakan pada sistem minimal sistem operasi *Microsoft windows 7*.
  - b. Aplikasi dirancang dengan menggunakan bahasa pemrograman *Microsoft Visual Studio 2008*. Dan aplikasi pendukung untuk desain yaitu *Adobe Photoshop CS3, CorelDRAW X3, Easy GIF Animator Pro*.
  - c. Spesifikasi komputer minimum *Processor dual core 1,33 Ghz, Memory 2 GB*, dan *VGA Card AMD Radeon 1 GB*.
  - d. *Speaker* untuk mendengarkan suara efek dari *game sound nya*.
2. Kenyamanan
  - a. *Game* ini dimainkan dengan menggunakan *mouse* komputer .
  - b. Dilengkapi dengan *menu* bantuan yang memudahkan pengguna dalam memainkannya.

### 3. Kinerja

- a. Waktu yang diperlukan dalam menampilkan *game* ini setelah dieksekusi menjadi *file* .EXE adalah 10 detik, sesuai dengan efek dari halaman pembuka yang merupakan pendukung dari aplikasi.

### III.2. Strategi Pemecahan Masalah

Didalam merancang aplikasi *game* berburu bebek ini, penulis menerapkan strategi pemecahan masalah setelah melihat masalah yang dapat ditimbulkan dari kebutuhan dalam pembuatan *game* dan mempelajarinya, maka diputuskan untuk merancang suatu program aplikasi *game* dan bagaimana menampilkan animasi dan audio *file* dalam sebuah *game*. Agar *game* dapat diterima dan banyak diminati oleh semua kalangan. Tipe audio *file* yang dipilih adalah *file* yang bertipe WAV, karena merupakan tipe *uncompressed audio file* yang mempunyai jumlah bit yang cukup banyak, sehingga suara efek pada *game* yang dirancang akan lebih menarik.

### III.3. Struktur Data Yang Digunakan

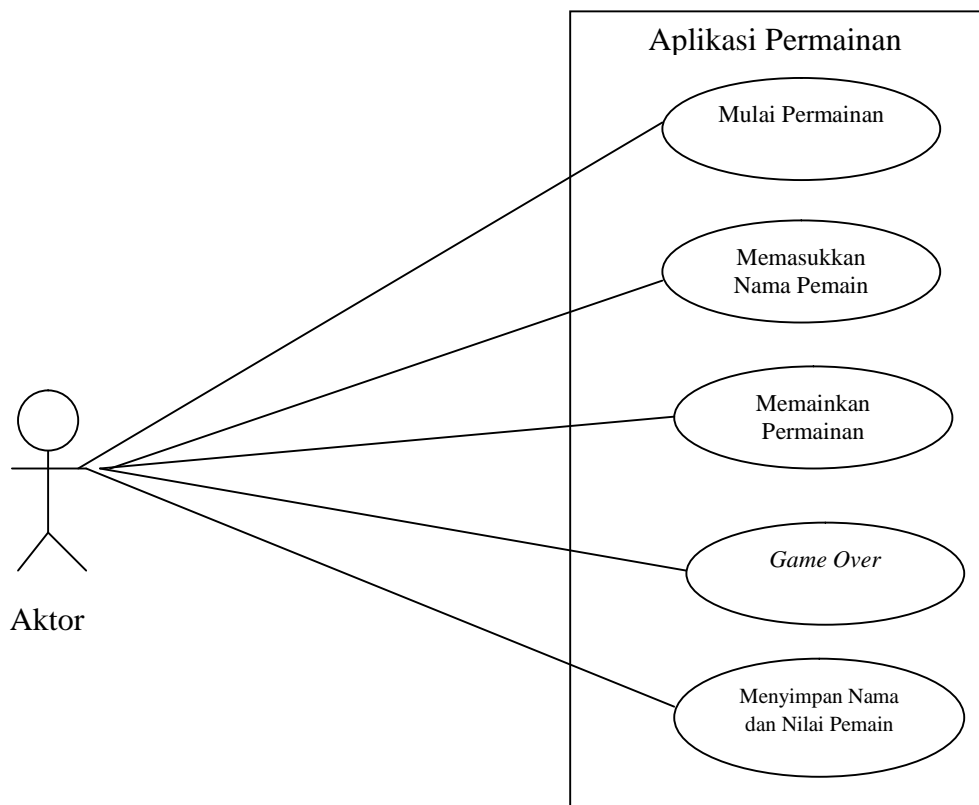
Pada aplikasi *game* ini, digunakan konsep *Object Oriented Programming* (OOP) untuk mengembangkannya oleh karena itu dipilih *Unified Modeling Language* (UML) untuk merancang arsitektur program. Struktur yang digunakan dalam perancangan adalah *use case diagram*, *activity diagram*, *sequence diagram* dan *class diagram*.

### III.4. Perancangan

Sub bab ini berisikan tentang perancangan aplikasi yang akan dirancang, dalam hal ini perancangan terhadap aplikasi permainan berburu bebek yang menggunakan UML.UML yang digunakan meliputi perancangan *use case diagram*, *activity diagram*, dan *sequence diagram*.

#### 1. Use case diagram

*Use case diagram* merupakan sebuah pekerjaan tertentu. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Berikut adalah *use case diagram* yang digunakan dalam merancang program aplikasi permainan berburu bebek.

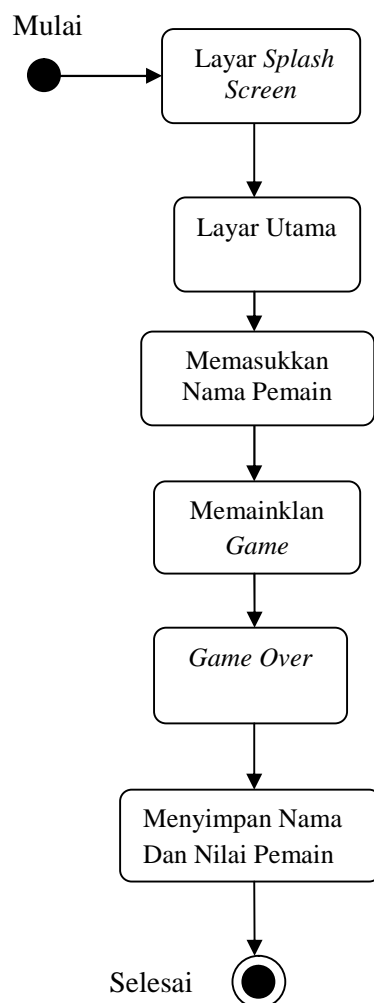


**Gambar III.1. Use Case Diagram Game Berburu Bebek**

## 2. Activity diagram

*Activity diagram* adalah teknik untuk mendiskusikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* banyak mempunyai peran seperti halnya *flowchart*, akan tetapi *flowchart* berbeda dengan *Activity diagram*. *Activity diagram* biasa mendukung perilaku paralel sedangkan *flowchart* tidak bisa.

Berikut adalah *activity diagram* yang digunakan dalam merancang program aplikasi *game* berburu bebek.



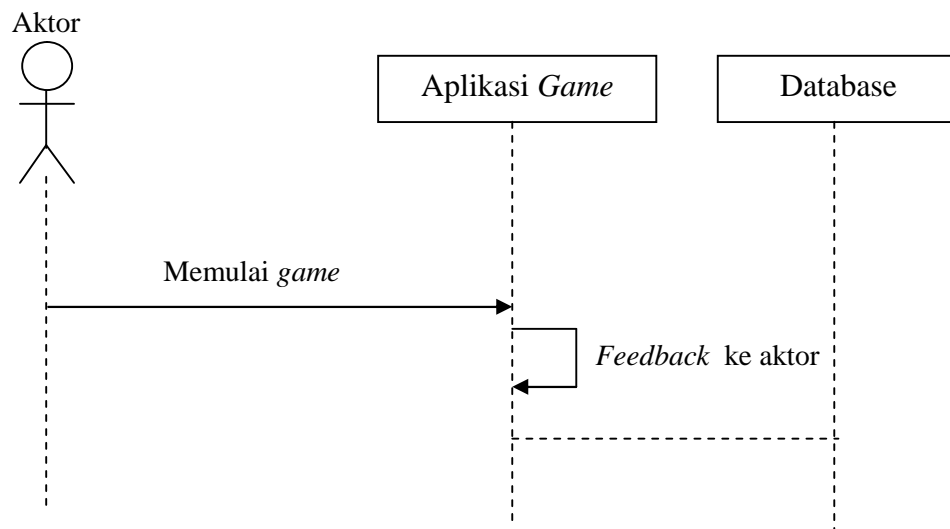
**Gambar III.2. Activity Diagram Game Berburu Bebek**

### 3. *Sequence diagram*

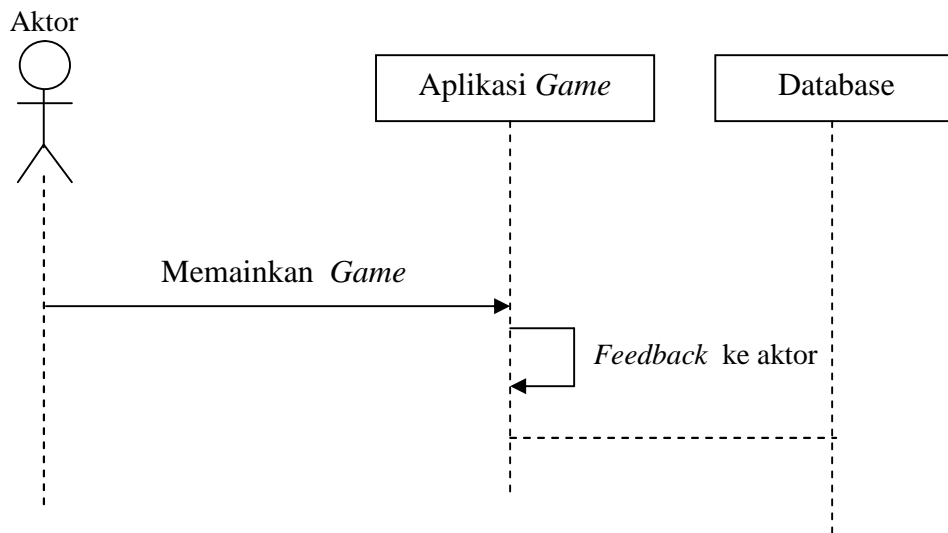
*Sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan *message* (pesan) yang diletakkan diantara objek-objek ini di dalam *use case diagram*.

Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segi empat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertikal*.

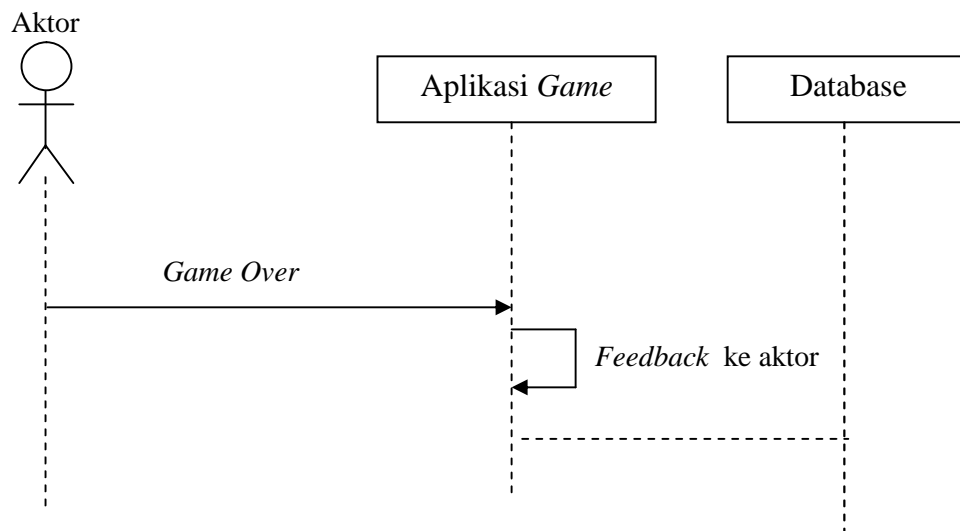
Berikut adalah *sequence diagram* yang digunakan dalam merancang program aplikasi permainan berburu bebek.



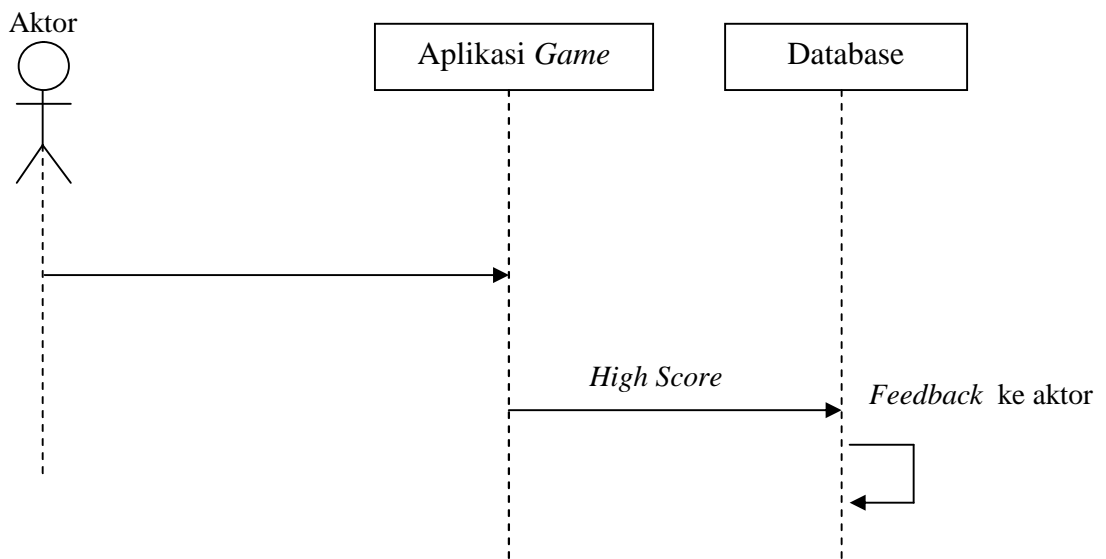
**Gambar III.3. *Sequence Diagram Memulai Game***



**Gambar III.4. Sequence Diagram Memainkan Game**



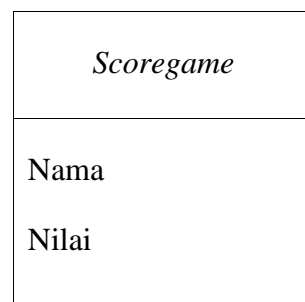
**Gambar III.5. Sequence Diagram Game Over**



**Gambar III.6. Sequence Diagram High Score**

#### 4. Class diagram

*Class* diagram sangat membantu dalam visualisasi struktur kelas dari suatu sistem. Hal ini disebabkan karena *class* adalah deskripsi kelompok obyek-obyek dengan *property*, perilaku (operasi) dan relasi yang sama. Disamping itu *class* diagram bisa memberikan pandangan global atas sebuah sistem.

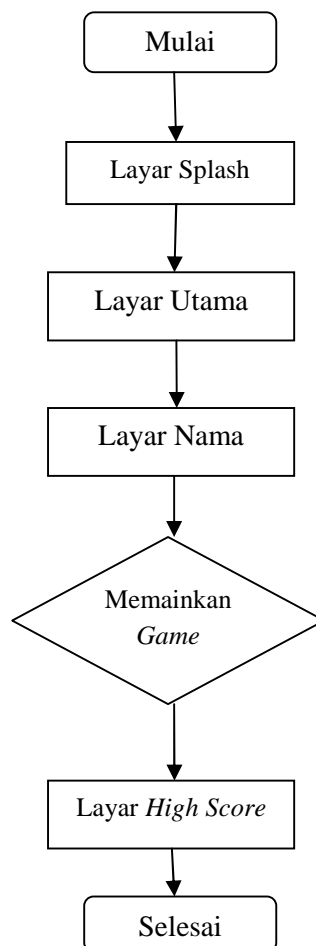


**Gambar III.7. Class Diagram Game Berburu Bebek**

### III.4.1. Flowchart

*Flowchart* merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta pernyataannya. Gambaran ini dinyatakan dengan simbol dan dengan demikian setiap simbol menggambarkan proses tertentu. Hubungan antar proses digambarkan dengan garis penghubung.

Berikut adalah *flowchart* yang digunakan dalam merancang program aplikasi permainan berburu bebek:



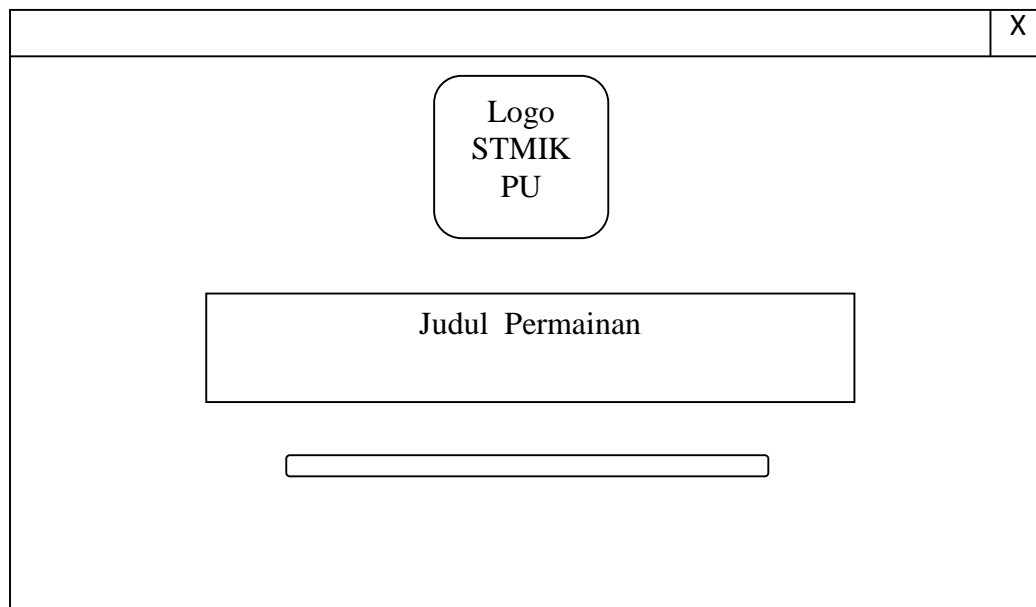
**Gambar III.8. Flowchart Game Berburu Bebek**

### III.4.2. Rancangan Layar

Layar bagi pengguna merupakan tampilan dimana pengguna dapat berinteraksi dengan aplikasi yang di desain. Tujuan dari layar bagi pengguna adalah untuk memungkinkan pengguna menjalankan setiap aplikasi tersebut.

#### 1. Layar *Splash Screen*

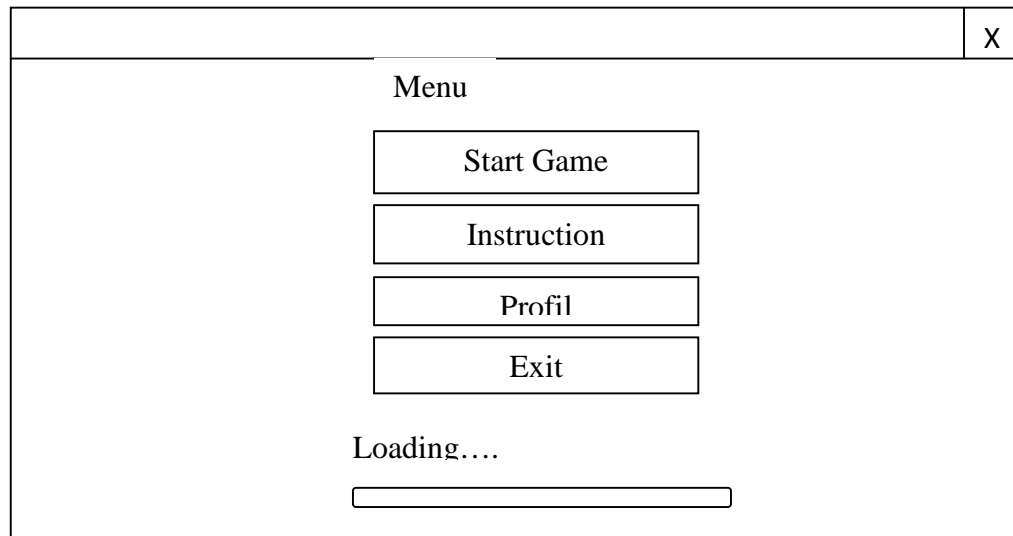
Layar ini adalah semacam layar perkenalan kepada pemain, layar ini yang akan dilihat pemain saat pertama kali menjalankan program ini. Kemudian pemain akan dihadapkan pada layar utama. Berikut adalah rancangan layar *splash screen* dapat dilihat pada gambar III.9.



**Gambar III.9. Layar *Splash Screen***

## 2. Layar Utama

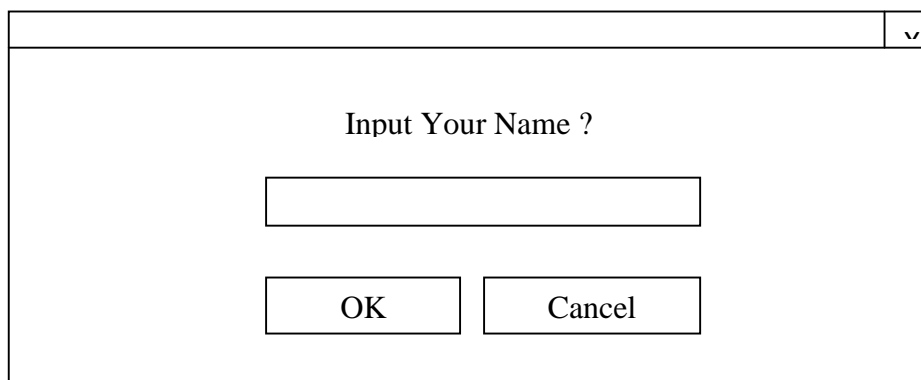
Layar ini akan muncul beberapa saat setelah layar *splash screen*. Layar utama ini berisikan, tombol *start game*, *instruction*, *profil* dan tombol *exit*. Layar ini. Layar utama ini dapat dilihat pada gambar III.10.



**Gambar III.10. Layar Utama**

## 3. Layar Nama

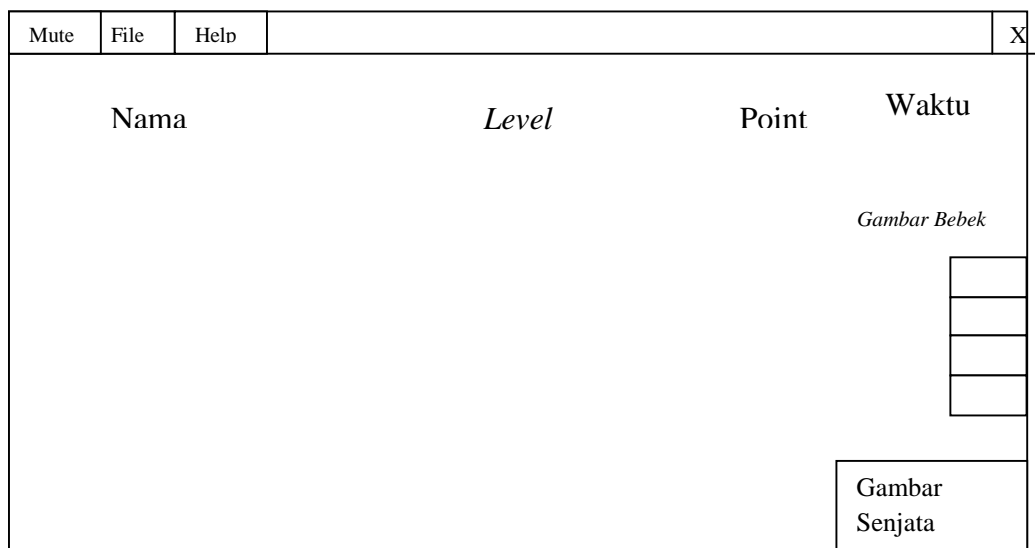
Layar Nama adalah layar yang berfungsi untuk memasukkan nama *user* yang ingin bermain *game*.



**Gambar III.11. Layar Nama**

#### 4. Layar Permainan

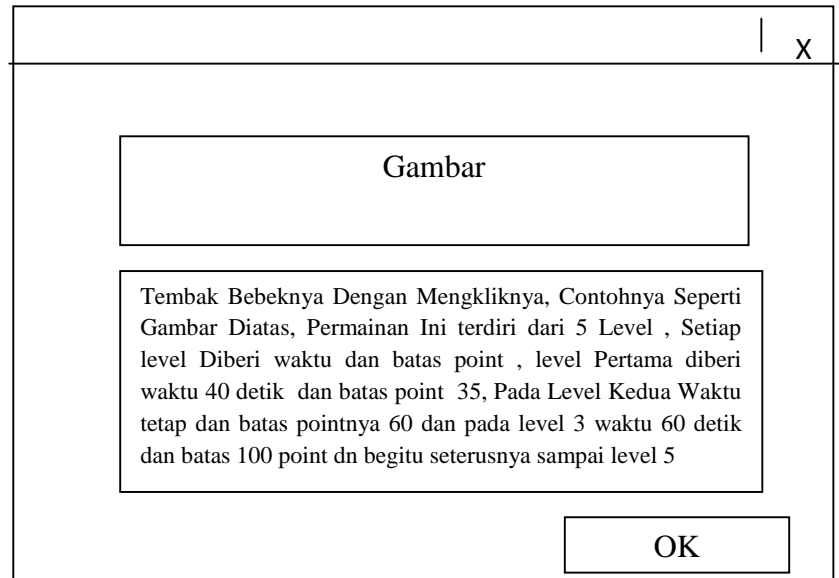
Layar ini adalah layar ini muncul setelah anda menekan tombol *start game* di layar utama. Layar ini berisikan menu bar yang bisa untuk mengakses kesemua layar atau mengatur properti pada *game*. Layar ini juga menampilkan nilai yang didapat oleh pemain serta peluru yang dimiliki. Waktu untuk mengetahui berapa banyak waktu.



**Gambar III.12. Layar Permainan**

#### 5. Layar *Instructions*

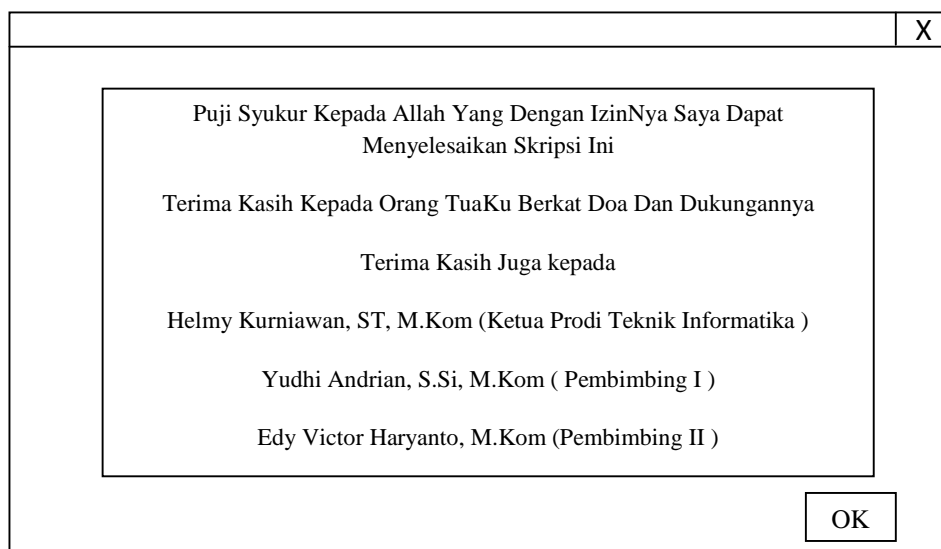
Layar ini adalah layar yang berisikan instruksi atau cara memainkan *game*. Layar *instructions* ini dapat dilihat pada gambar III.13



**Gambar III.13. Layar *Game Instructions***

## 6. Layar Terima Kasih

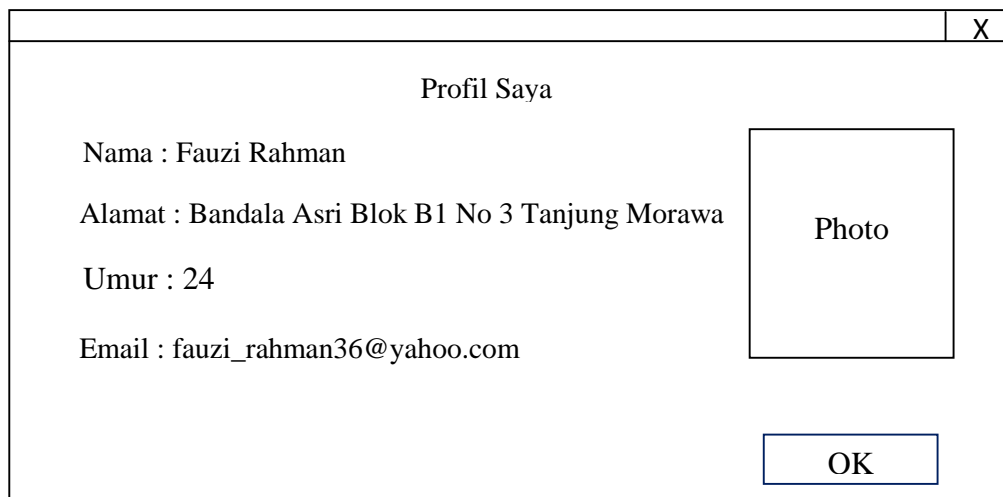
Layar terima kasih adalah layar daftar terima kasih yang disampaikan oleh penulis terhadap orang-orang yang berperan dalam pembuatan skripsi ini yang telah memberikan dukungan baik moril maupun material serta spiritual bagi penulis dalam menyelesaikan laporan skripsi ini. Layar terima kasih ini dapat dilihat pada gambar III.14.



**Gambar III.14. Layar Terima Kasih**

## 7. Layar Profil Saya

Layar profil saya adalah layar yang memberi info tentang yang membuat permainan ini, yang berisikan info pribadi penulis dan foto. Layar profil saya ini dapat dilihat pada gambar III.15.



**Gambar III.15. Layar Profil Saya**

## 8. Layar About

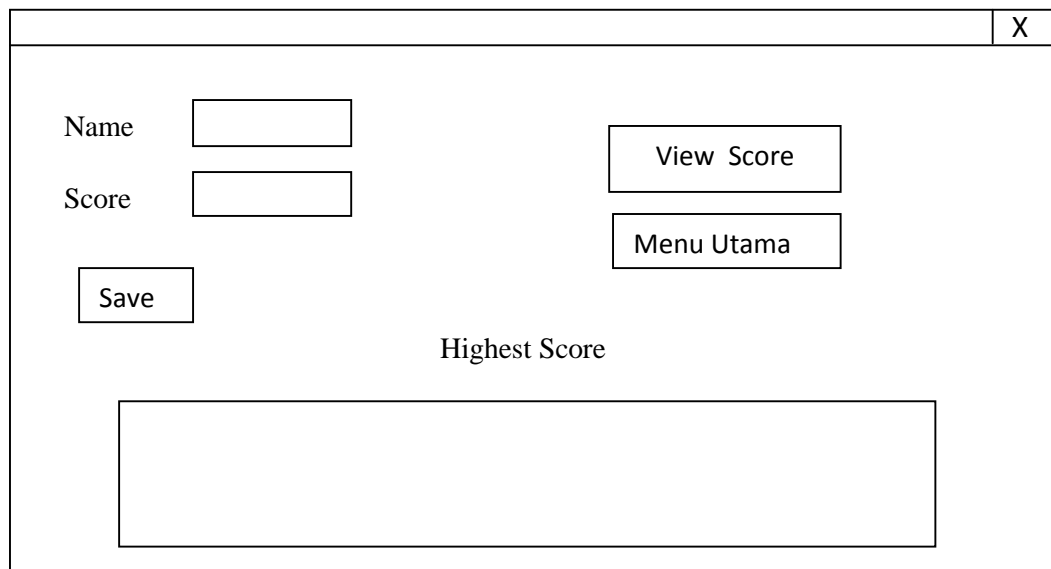
Layar *about* adalah layar yang memberi info tentang aplikasi yang dirancang. Layar *about* ini dapat dilihat pada gambar III.16.



**Gambar III.16. Layar About**

### 9. Layar High Score

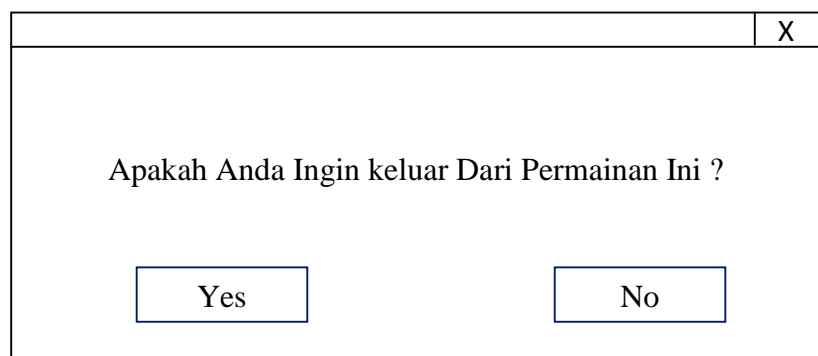
Layar *high score* adalah layar yang memberi info tentang nilai yang didapat oleh *user* setelah selesai bermain.



Gambar III.17. Layar *High Score*

### 10. Layar Konfirmasi

Saat pemain menghendaki untuk mengakhiri permainan, maka pemain harus mengklik *file* lalu keluar (*exit*) maka akan keluar jendela konfirmasi yang memberikan informasi apakah pemain ingin mengakhiri permainan atau tidak. Hal ini agar pemain tidak keluar dari permainan secara sengaja.



Gambar III.18. Layar Konfirmasi

### III.4.3. Algoritma Dari Program

Algoritma merupakan langkah-langkah penyelesaian dari suatu permasalahan. Dapat didefinisikan bahwa algoritma adalah uraian langkah-langkah pemecahan atau penyelesaian suatu tugas dalam bentuk instruksi-instruksi yang terperinci.

Adapun algoritma dari program *game* berburu bebek ini adalah sebagai berikut:

1. Algoritma *game* berburu bebek.
  - a. Pemain atau *user* pertama kali memulai aplikasi *game* berburu bebek pemain dihadapkan dengan layar *splash screen*. Layar *splash screen* adalah semacam layar perkenalan kepada pemain. Layar ini akan hilang dalam waktu beberapa detik yang kemudian akan muncul layar utama *game*.
  - b. Layar utama dirancang menggunakan pilihan tombol menu seperti tombol *start game*, *instruction*, *about* dan tombol *exit*. Sebelum memulai permainan sebaiknya pemain membaca dahulu instruksi permainannya. Tekan tombol *start game* untuk memainkan *game*.
  - c. *Game* akan selesai atau *game over* jika pemain telah waktu yang diberikan telah habis dan target *point* tidak tercapai.