

## **BAB III**

### **ANALISIS DAN PERANCANGAN**

#### **III.1. Analisa Sistem**

Perancangan *game* mencocokkan gambar ini dibuat agar dapat berjalan pada sistem yang beroperasi pada perangkat komputer, game yang dikembangkan adalah mengenai mencocokkan gambar yang bertujuan untuk menguji ingatan pemain. Konsep permainan ini hanya memilih gambar yang sama, dan apabila sama maka gambar tersebut akan terbuka. Apabila pemain tidak menemukan gambar yang sama, maka gambar tersebut akan tertutup.

Untuk menyempurnakan game ini agar tidak monoton maka penulis membuat game ini dengan teracak, dan penulis menggunakan metode LCM (Linear Congruent Method). Dimana metode LCM ini untuk membangkitkan bilangan acak, dan pada metode ini akan melakukan perulangan pada periode waktu tertentu atau setelah sekian kali pembangkitan. Hal ini adalah salah satu sifat utama daripada metode ini. Dengan menggunakan LCM, suatu solusi pengacakan gambar dapat terjadi tanpa terdapat pengulangan pada acakan yang sebelumnya. Dengan demikian, aplikasi ini diharapkan akan memberikan hasil yang lebih efektif terhadap pengacakan gambar setiap memulai permainan baru. Adapun aktifitas yang dapat diberikan kepada pemain adalah :

1. Pemain dapat memainkan game untuk menguji ingatan pemain terhadap gambar yang telah dipilih.

2. Pemain dapat melihat tentang mengakses beberapa menu yang memberikan layanan ataupun bantuan mengenai game mencocokkan gambar tersebut.

### **III.2. Analisa Kebutuhan Sistem**

Perancangan bertujuan untuk memberikan gambaran logika sistem yang baru serta garis besar kepada pemakai sistem dalam desain sistem tersebut sudah tergambar logika dari seluruh sistem. Implementasi program perancangan dan pembuatan game ini merupakan tahap paling penting dimana sistem yang sudah dirancang, diimplementasikan untuk menghasilkan sistem yang sesuai dengan yang diinginkan dan siap dioperasikan pada keadaan yang sebenarnya.

Pembuatan animasi ini membutuhkan serangkaian peralatan yang dapat mendukung kelancaran proses perancangan dan pembuatan animasi. Berikut ini aspek-aspek yang dibutuhkan dalam pembuatan animasi :

#### **III.2.1. Perangkat Keras (*Hardware*)**

Hardware merupakan komponen yang terlihat secara fisik, yang saling bekerjasama dalam pengolahan data. Perangkat keras (hardware) yang digunakan meliputi :

- a. Laptop 14"
- b. Memori 2 GB
- c. Mouse
- d. Speaker aktif

### III.2.2. Perangkat Lunak (Software)

Software adalah instruksi atau program-program komputer yang dapat digunakan oleh komputer dengan memberikan fungsi serta penampilan yang diinginkan. Dalam hal ini, perangkat lunak yang digunakan untuk aplikasi perancangan adalah :

- a. Sistem Operasi Windows 7
- b. Microsoft Visual Studio 2010

### III.3. Strategi Pemecahan Masalah

*Linear Congruent Method* (LCM) adalah sebuah metode yang membangkitkan bilangan acak yang banyak dipergunakan dalam program komputer. Pada metode ini, perulangan pada periode waktu tertentu atau setelah sekian kali pembangkitan. Hal ini adalah salah satu sifat utama daripada metode ini. Penentuan konstanta pada *Linear Congruent Method* (LCM) sangat menentukan baik tidaknya bilangan acak yang diperoleh dalam arti memperoleh bilangan acak yang seakan-akan tidak akan terjadi pengulangan. Pemakaian metode *Linear Congruent Method* (LCM) dalam kasus ini adalah hanya untuk pengacakan gambar agar setiap permainan baru dapat dilakukan pengacakan dengan efektif.

Dengan menggunakan *Linear Congruent Method* (LCM), suatu solusi pengacakan gambar dapat terjadi tanpa terdapat pengulangan pada acakan yang sebelumnya. Dengan demikian, diharapkan aplikasi yang dirancang ini akan

memberikan hasil yang lebih efektif terhadap pengacakan gambar setiap pemulaian permainan baru.

### III.3.1. Solusi Pengacakan pada Aplikasi *Try Out*

*Linear Congruent Method* (LCM) merupakan metode pembangkitan bilangan acak yang banyak digunakan dalam program komputer. *Linear Congruent Method* (LCM) memanfaatkan model linear untuk membangkitkan bilangan acak.

Ciri khas dari *Linear Congruent Method* (LCM) adalah terjadi pengulangan pada periode waktu tertentu atau setelah sekian kali pembangkitan, hal ini adalah salah satu sifat dari metode ini dan *pseudorandom generator* pada umumnya. Penentuan konstanta *Linear Congruent Method* (LCM) ( $a$ ,  $c$  dan  $m$ ) sangat menentukan baik tidaknya bilangan acak yang diperoleh dalam arti memperoleh bilangan acak yang seakan-akan tidak terjadi perulangan.

Jika terdapat gambar sebanyak 20 buah dan belum diacak, di mana proses pengacakan gambar dapat dilakukan dengan menentukan nilai  $a = 1$ ,  $c = 7$ ,  $m = 20$  dan  $X_0 = 2$  adalah sebagai berikut, dengan rumus :

$$X_i = ( a * x_i + c ) \text{ mod } m$$

Dimana :  $X_i$  adalah bilangan acak ke  $n$

$a$  dan  $c$  adalah konstanta LCM

$m$  adalah batas maksimum bilangan acak

Dapat dilihat dalam konteks contoh seperti berikut bahwa membangkitkan bilangan acak sebanyak 8 kali dengan ketentuan  $a = 4$ ,  $c = 7$ ,  $m = 20$  dan  $X_0 = 2$ ,

agar nilai  $X_i$  tidak menghasilkan 0, maka dalam simulasi pengacakan gambar ini, setiap kali  $X_i$  telah ditambahkan dengan 1, dan rumusnya adalah :

$$X_i + 1$$

Maka dapat diperoleh hasil dari rumus diatas adalah :

$$X_i = ( a * x_i + c ) \text{ mod } m$$

$$1. X_1 = ( 1 (2) + 7 ) \text{ mod } 20$$

$$= 9$$

$$= X_1 + 1$$

$$= 9 + 1 = 10$$

$$2. X_2 = ( 1 (9) + 7 ) \text{ mod } 20$$

$$= 16$$

$$= X_2 + 1$$

$$= 16 + 1 = 17$$

$$3. X_3 = ( 1 (16) + 7 ) \text{ mod } 20$$

$$= 3$$

$$= X_3 + 1$$

$$= 3 + 1 = 4$$

$$4. X_4 = ( 1 (3) + 7 ) \text{ mod } 20$$

$$= 10$$

$$= X_4 + 1$$

$$= 10 + 1 = 11$$

$$5. X_5 = (1(10) + 7) \bmod 20$$

$$= 17$$

$$= X_5 + 1$$

$$= 17 + 1 = 18$$

$$6. X_6 = (1(17) + 7) \bmod 20$$

$$= 4$$

$$= X_6 + 1$$

$$= 4 + 1 = 5$$

$$7. X_7 = (1(4) + 7) \bmod 20$$

$$= 11$$

$$= X_7 + 1$$

$$= 11 + 1 = 12$$

$$8. X_8 = (1(11) + 7) \bmod 20$$

$$= 18$$

$$= X_8 + 1$$

$$= 18 + 1 = 19$$

$$9. X_9 = (1(18) + 7) \bmod 20$$

$$= 5$$

$$= X_9 + 1$$

$$= 5 + 1 = 6$$

$$10. X_{10} = (1(5) + 7) \bmod 20$$

$$= 12$$

$$= X_{10} + 1$$

$$= 12 + 1 = 13$$

$$11. X_{11} = (1(12) + 7) \bmod 20$$

$$= 19$$

$$= X_{11} + 1$$

$$= 19 + 1 = 20$$

$$12. X_{12} = (1(19) + 7) \bmod 20$$

$$= 6$$

$$= X_{12} + 1$$

$$= 6 + 1 = 7$$

$$13. X_{13} = (1(6) + 7) \bmod 20$$

$$= 13$$

$$= X_{13} + 1$$

$$= 13 + 1 = 14$$

$$14. X_{14} = (1(13) + 7) \bmod 20$$

$$= 0$$

$$= X_{14} + 1$$

$$= 0 + 1 = 1$$

$$15. X_{15} = (1(0) + 7) \bmod 20$$

$$= 7$$

$$= X_{15} + 1$$

$$= 7 + 1 = 8$$

$$16. X_{16} = (1(7) + 7) \bmod 20$$

$$= 14$$

$$= X_{16} + 1$$

$$= 14 + 1 = 15$$

$$17. X_{17} = (1(14) + 7) \bmod 20$$

$$= 1$$

$$= X_{17} + 1$$

$$= 1 + 1 = 2$$

$$18. X_{18} = (1(1) + 7) \bmod 20$$

$$= 8$$

$$= X_{18} + 1$$

$$= 8 + 1 = 9$$

$$19. X_{19} = (1(8) + 7) \bmod 20$$

$$= 15$$

$$= X_{19} + 1$$

$$= 15 + 1 = 16$$

$$20. X_{20} = (1(15) + 7) \bmod 20$$

$$= 2$$

$$= X_{20} + 1$$

$$= 2 + 1 = 3$$

Maka, bilangan acak yang dibangkitkan adalah : 10, 17, 4, 11, 18, 5, 12, 19, 6, 13, 20, 7, 14, 1, 8, 15, 2, 9, 16, 3. Dengan keterangan,  $X_1$  merupakan gambar nomor 1 sebelum diacak. Agar nilai  $X(i)$  tidak menghasilkan 0, maka dalam simulasi pengacakan gambar ini, setiap kali  $X(i)$  telah ditambahkan dengan 1. Setelah terjadi pengacakan pada gambar, maka gambar yang akan muncul sebagai gambar nomor 1 adalah menjadi gambar nomor 10, dan untuk gambar nomor 2 menjadi gambar nomor 17 demikian sampai gambar ke-n, hasil dari pembangkitan bilangan acak dengan nilai-nilai tersebut dapat kita lihat pada tabel berikut.

**Tabel III.1. Tabel Pembangkitan Bilangan Acak**

I	$X_i$	$X_{i+1}$				11	12	20
1	2	10				12	19	7
2	9	17				13	6	14
3	16	4				14	13	1
4	3	11				15	0	8
5	10	18				16	7	15
6	17	5				17	14	2
7	4	12				18	1	9
8	11	19				19	8	16
9	18	6				20	15	3
10	5	13						

Dari hasil pembangkitan bilangan acak di atas terjadi pembangkitan bilangan 1 pada periode 20, hal ini menunjukkan perulangan kembali ke awal setelah terjadi pengacakan kepada seluruh bilangan. Setelah nilai-nilai dari variabel pada metode LCM didapat. Maka pengacakan dapat dilakukan. Setiap proses pengacakan berarti menghasilkan satu paket permainan dan jika pengguna menginginkan paket sebanyak yang diinginkan, maka *user* harus melakukan pengacakan sebanyak itu pula.

Algoritma *Linear Congruent Method* (LCM) adalah sebagai berikut :

*Input* : masukkan konstanta ( $a$ ,  $c$ , dan  $m$ )

: batas pembangkitan bilangan acak ( $n$ )

: tentukan pengumpan ( $Z_0$ )

*Output* : bilangan acak hasil pembangkitan ( $Z_i$ )

*Proses* :  $a =$  konstanta

:  $c =$  konstanta

:  $m =$  konstanta

:  $n =$  batas bilangan acak

:  $X_0 =$  pengumpan

$i = 1$

*While*  $i \leq n$

$Z(i) = ( a Z(i-1) + c ) \text{ mod } m$

*If*  $Z(i) = 0$  *then*

$Z(i) = m$

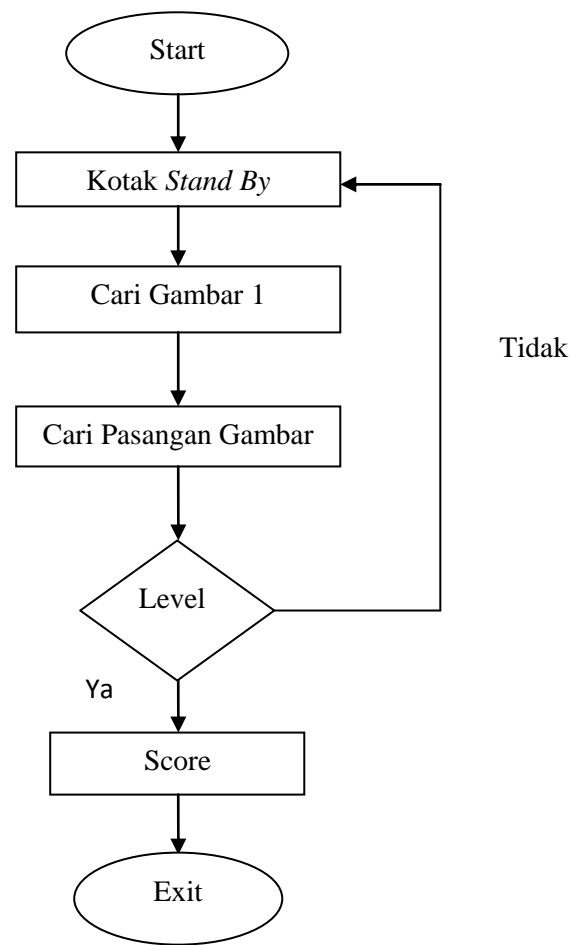
*end if*

$$i = i + 1$$

*End while*

### III.3.2. Flow Chart Linear Congruent Method

Adapun *flow chart Linear Congruent Method* pada aplikasi ini adalah sebagai berikut :



**Gambar III.1. Flow Chart Linear Congruent Method pada Aplikasi Game**

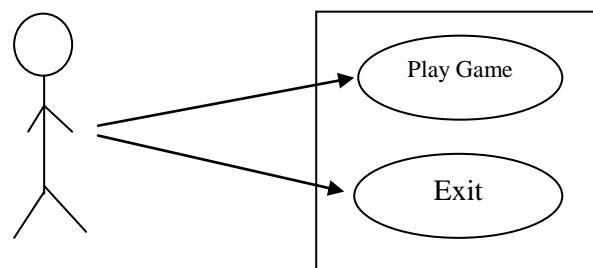
**Mencocokkan Gambar**

### III.4. Perancangan Sistem

Pada perancangan aplikasi game mencocokkan gambar dengan metode LCM (*Linear Congruent Method*) mengenai rancangan game yang akan dikerjakan serta fitur-fitur yang akan dipakai pada game tersebut seperti rancangan layar dan lainnya.

#### III.4.1. Use Case Diagram

*Use Case Diagram* berfungsi untuk menggambarkan kegiatan aktor atau pemain game. Adapun *use case diagram game* yang dirancang dapat dilihat pada gambar III.1. sebagai berikut :



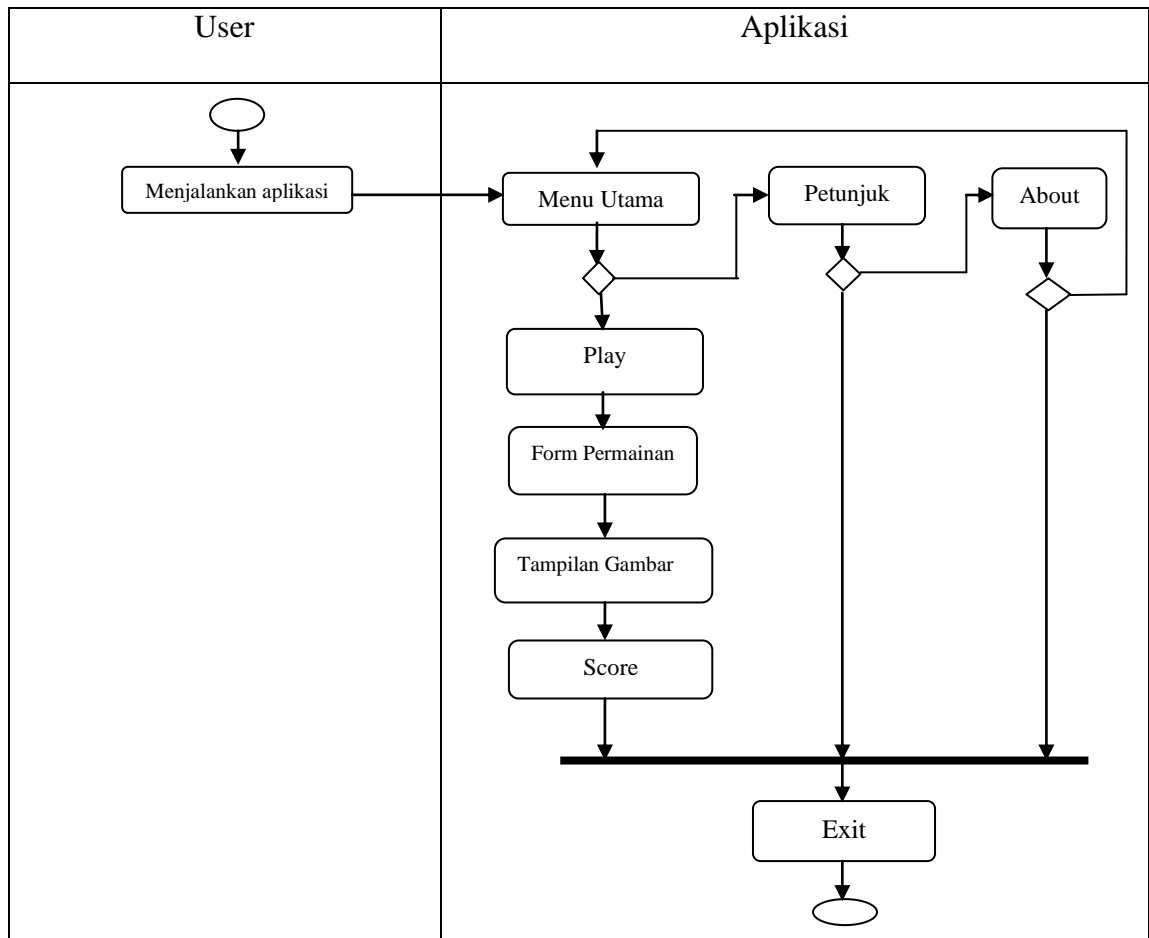
**Gambar III.2. Use Case Diagram Game Mencocokkan Gambar**

Dari gambar use case diagram diatas menjelaskan tentang sistem yang berjalan pada aplikasi game mencocokkan gambar, adapun fungsi use case yaitu :

1. Play Game, merupakan form yang menampilkan arena permainan.
2. Exit, merupakan button untuk mengakhiri game atau keluar dari game.

### III.4.2. Activity Diagram

Pada activity diagram dibawah ini menggambarkan proses yang berjalan pada game yang dimainkan. Setelah menjalankan game, pemain dapat memulai permainan, yang dapat dilihat pada tabel III.2. sebagai berikut :



**Tabel III.2. Activity Diagram Game Mencocokkan Gambar**

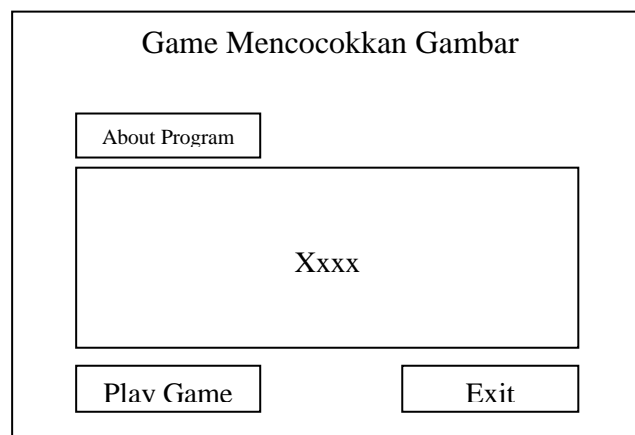
Dari gambar activity diagram diatas, proses permainan merupakan tahapan yang disajikan terhadap permainan yang telah dimainkan dan penilaian merupakan proses yang diperoleh pemain dalam memainkan game mencocokkan gambar tersebut.

### III.5. Perancangan Layar/*Interface*

Game yang dirancang merupakan game yang menggunakan metode LCM (Linear Congruent Method), berikut ini merupakan penjelasan dari rancangan layar game yang dirancang.

#### 1. Tampilan Menu Awal

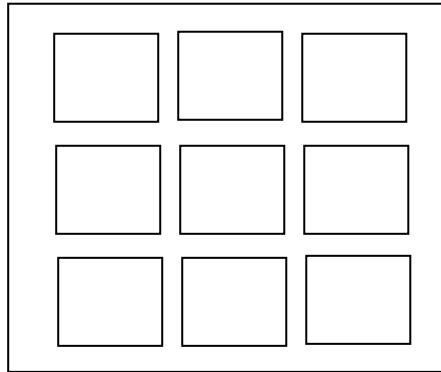
Rancangan layar menu awal merupakan awal pembuka game, terdapat 3 button yang dapat diakses oleh pemain yaitu start untuk memulai permainan, about untuk informasi tentang game, dan close untuk keluar dari game, yang dapat dilihat pada gambar III.3.



**Gambar III.3. Tampilan Menu Awal**

#### 2. Tampilan Form Mulai Permainan

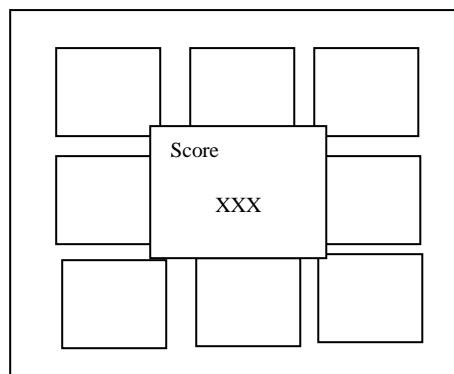
Pada tampilan form ini pemain dapat memulai permainan dengan menjawab setiap gambar yang akan ditampilkan, yang dapat dilihat pada gambar III.4.



**Gambar III.4. Tampilan Form Mulai Permainan**

### **3. Tampilan Form *Score***

Pada tampilan ini merupakan *score* hasil dari permainan mencocokkan gambar, dimana perhitungan *score* ini dirancang dalam pengurangan waktu, maka dalam setiap detik yang berjalan dikurang 2 *score* hingga batas waktu nilai nol.

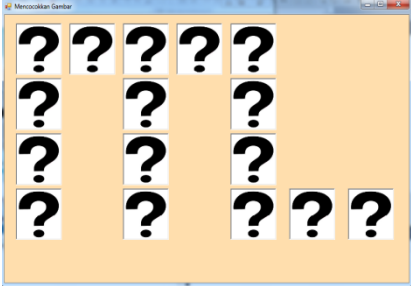

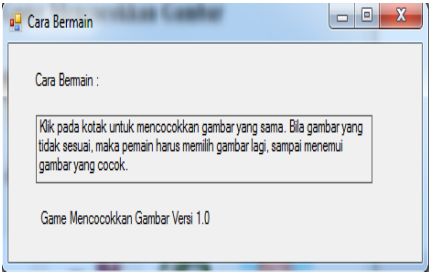


**Gambar III.5. Tampilan Form *Score***

### **III.6. *Storyboard***

Merupakan cerita ataupun alur dari aplikasi yang digunakan, berikut ini adalah tampilan storyboard dari aplikasi game mencocokkan gambar :

Tabel III.3. Tabel *Storyboard*

Tampilan	Penjelasan
	<p>Pengguna harus memilih kotak yang tersedia untuk memilih gambar dan mencari gambar yang sama, sehingga gambar yang sama akan tampil, dan apabila tidak cocok maka kotak tersebut akan tutup kembali.</p>
	<p>Gambar 2 dimensi ini merupakan salah satu gambar yang akan dipilih <i>user</i>, dimana <i>user</i> harus mencari gambar yang sama seperti ini.</p>
	<p>Tampilan ini menjelaskan cara bermain dalam permainan ini.</p>