

BAB III

ANALISA MASALAH DAN RANCANGAN PROGRAM

III.1. Analisa

Pada bab ini akan dibahas mengenai analisis permainan, yaitu konsep aturan dan cara bermain pada *game* yang berhubungan dengan program yang sedang dibangun yaitu *game Color Memory* dan metode yang digunakan untuk pemecahan solusinya yaitu algoritma *Brute Force*.

Penerapan konsep kecerdasan buatan sangat membantu dalam berbagai masalah pengacakan. Salah satu contoh penerapan kecerdasan buatan untuk masalah pengacakan adalah dalam permainan *Color Memory*. Dalam permainan *Color Memory*, akan tampil urutan warna yang harus diingat oleh pemain berupa kedipan (*blink*) pada papan warna secara bergantian, pemain mengklik papan warna sesuai urutan kedipan yang ditampilkan. Tampilan warna akan diacak kedipannya dan jumlah kedipan juga akan meningkat.

Game ini menggunakan algoritma *Brute Force* sebagai metode optimasi untuk pengacakan pada pola kedipan, pola yang dibandingkan adalah pola sebelumnya dan pola sesudahnya. Pada proses pengacakan ini, akan didapat hasil berupa pola dengan kedipan yang sama, dan bila pemain benar mengingat urutan warna maka jumlah kedipan warna akan bertambah satu kedipan untuk setiap pola permainan selanjutnya.

Algoritma *Brute Force* merupakan algoritma dengan sebuah pendekatan yang lempang (*straight forward*) untuk memecahkan suatu masalah, biasanya

didasarkan pada pernyataan masalah (*problem statement*) dan definisi konsep yang dilibatkan. Algoritma *Brute Force* memecahkan masalah dengan sangat sederhana, langsung dan dengan cara yang jelas (*obvious way*). Adapun perancangan Aplikasi *Game Color Memory* ini menggunakan bahasa pemrograman *visual basic* dengan interface penulisan coding Visual Studio 2010.

III.1.1. Analisa Algoritma Optimasi Metode *Brute Force*

Adapun Metode *Brute Force* dalam optimasi pengacakan adalah sebagai berikut :

1. Mula-mula *pattern* dicocokkan pada awal pola.
2. Dengan bergerak dari kiri ke kanan, bandingkan setiap urutan di dalam pola dengan urutan yang bersesuaian sebelumnya sampai :
 - a. semua pola yang dibandingkan cocok atau sama (pengacakan urutan pola berhasil),
 - b. dijumpai sebuah ketidakcocokan pola (pengacakan urutan pola belum berhasil)
3. Bila *pattern* belum ditemukan kecocokannya dan pola belum habis, geser *pattern* satu pola ke kanan serta penambahan satu pola dan ulangi langkah 2.

Contoh penelusuran pola :

Pattern: 001011

Pola: 10010101**001011**110101010001

10010101**001011**110101010001

1 001011
 2 001011
 3 001011
 4 001011
 5 001011
 6 001011
 7 001011
 8 001011
 9 **001011**

III.1.2. Penerapan *Brute Force* pada Optimasi Pengacakan

Pada kasus pencarian di *game* ini, langkah-langkah yang dilakukan untuk mencari kecocokan pola dengan menggunakan metode *naïve search* atau *brute force* adalah sebagai berikut:

a) Pola pengacakan :

1. Buat peubah i, j, k , dan m . Inisiasikan semuanya dengan nilai 0.
2. Jika belum ada, buat sebuah *array* asosiatif untuk menampung indeks hasil, H . Dimana masing-masing elemen H adalah peubah *array* yang merepresentasikan arah hasil pencarian, yaitu: “horizontal”, “vertikal”, “diagonal_turun”, dan “diagonal_naik”.
3. Bandingkan kecocokan antara array P ke- $[m][k+i]$ dengan array p ke- i ($P[m][k+i] = p[i]$?).
4. Apabila terjadi kecocokan:

- a. Tambahkan nilai i dengan 1.
 - b. Apabila nilai i belum sama dengan jumlah array p dan nilai $k+i$ belum melebihi jumlah kolom pada array $P - 1$, ulangi langkah 3.
 - c. Apabila nilai i sama dengan jumlah array $p-1$ dan nilai $k+i$ belum melebihi jumlah kolom pada array $P - 1$, maka pencarian dikatakan berhasil. Simpan nilai variabel k dan m ke dalam array H indeks ke- j . Kemudian tambahkan variabel j dengan 1. Tambahkan nilai k dengan 1. Nilai i menjadi 0. Kemudian ulangi langkah 3.
 - d. Apabila nilai $k+i$ melebihi jumlah kolom pada array $P - 1$, maka pencarian di baris m dikatakan gagal. Hentikan pencarian. Tambahkan nilai m dengan 1. Nilai k dan i masing-masing menjadi 0.
 - e. Apabila nilai m belum melebihi jumlah baris pada array $P - 1$, ulangi langkah 3.
 - f. Apabila nilai m melebihi jumlah baris pada array $P - 1$, maka pencarian secara keseluruhan dikatakan gagal.
5. Apabila terjadi ketidakcocokan :
- a. Tambahkan nilai k dengan 1. Nilai i menjadi 0.
 - b. Apabila nilai k belum melebihi jumlah kolom pada array $P-1$, ulangi langkah 3.
 - c. Apabila nilai k melebihi jumlah kolom pada array $P-1$, maka pencarian di baris m dikatakan gagal. Hentikan pencarian. Tambahkan nilai m dengan 1. Nilai k dan i masing-masing menjadi 0.

- d. Apabila nilai m belum melebihi jumlah baris pada *array* $P - 1$, ulangi langkah 3.
- e. Apabila nilai m melebihi jumlah baris pada *array* $P - 1$, maka pencarian secara keseluruhan dikatakan gagal.

Berikut ini merupakan *pseudocode* dari metode *brute force* :

Deklarasi

$i, j, k, m : integer$

$H : Array$

Algoritma:

$i, j, k, m \rightarrow 0$

while ($m \leq P.length-1$) *do*

if ($P[m][k+i] = p[i]$) *then*

if ($i = p.length-1$) *and* ($k+i \leq P[m].length-1$) *then*

$H[j][0] \rightarrow k$

$H[j][1] \rightarrow m$

$j \rightarrow j+1$

$k \rightarrow k+1$

$i \rightarrow 0$

else if ($k+i > P[m].length-1$) *then*

$m \rightarrow m+1$

$k \rightarrow 0$

$i \rightarrow 0$

```

        if (m > P.length-1) then
            break
        endif
    else
        i → i+1
    endif
else
    k → k+1
    i → 0
    if (k > P[m].length-1) then
        m → m+1
        k → 0
        i → 0
        if (m > P.length-1) then
            break
        endif
    endif
endif
endwhile

```

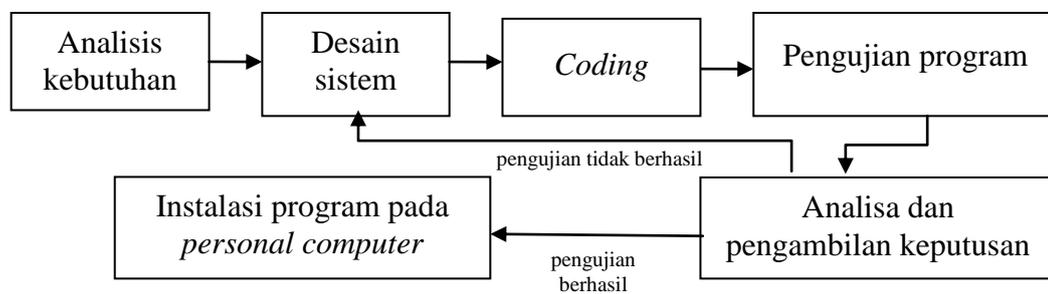
III.2. Strategi Pemecahan Masalah

Aplikasi *game Color Memory* yang akan dirancang merupakan aplikasi permainan mengingat warna yang sederhana, dimana aplikasi tersebut dapat

dijalankan di semua *platform*. Aplikasi ini merupakan aplikasi PC, dimana aplikasi ini akan dijalankan pada *personal computer*. Dalam pembuatan aplikasi *game* ini menggunakan bahasa pemrograman Visual Basic 2010.

Dalam pembuatan aplikasi *Game Color Memory*, penulis merancang sebuah menu awal dan desain media permainan saja. Pada menu awal akan ditampilkan keterangan pembuat program dan cara bermain serta tombol Mulai Bermain untuk memulai permainan.

Uraian tahapan perancangan dalam pembuatan Aplikasi Game Color Memory tersebut dijelaskan pada gambar di bawah ini.



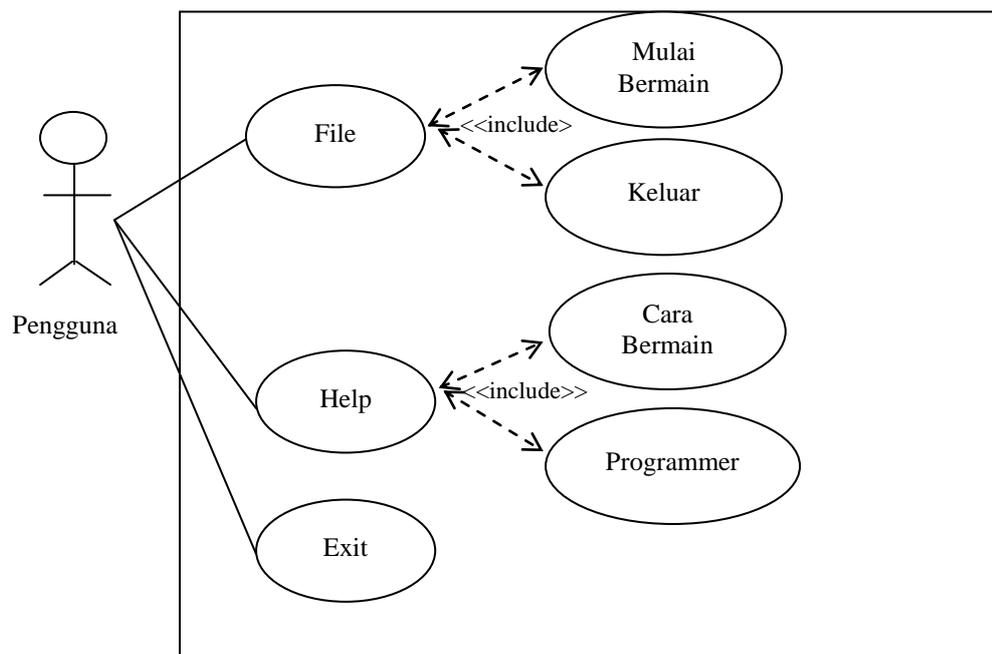
Gambar III.1. Tahapan Pembuatan Aplikasi Game Color Memory

III.3. Perancangan

Perancangan aplikasi *Game Color Memory* ini meliputi rancangan menu utama, yang di dalamnya terdapat menu pilihan berupa *File*, *Help* dan *Exit*. Dalam menu *File* terdapat sub menu Mulai Bermain dan Keluar, sedangkan dalam *Help* terdapat sub menu Cara Bermain dan *Programmer*.

III.3.1. Use Case Diagram

Use case diagram adalah penggambaran sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case* lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Adapun *use case* dari aplikasi yang dirancang dapat dilihat pada gambar III.2.



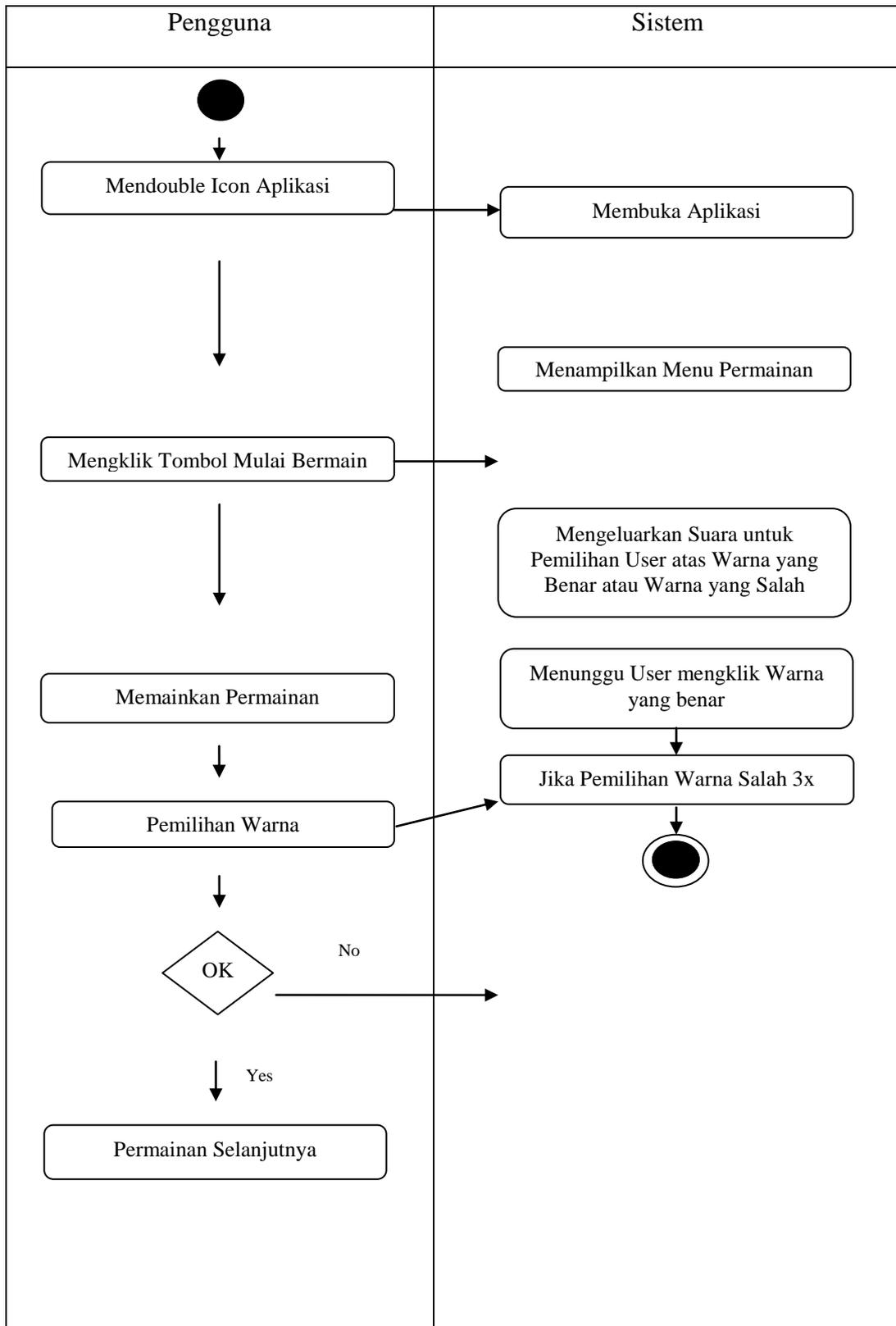
Gambar III.2. *Use Case Diagram* Aplikasi *Game Color Memory*

Dari gambar di atas diketahui bahwasannya pengguna mengklik menu *File*, lalu mengklik sub menu *Mulai Bermain* untuk memulai permainan dan mengklik tombol keluar untuk keluar dari aplikasi. Pada menu *Help*, pengguna dapat mengetahui keterangan untuk memainkan permainan dengan menekan tombol *Cara Bermain*, pengguna juga dapat melihat keterangan pembuat aplikasi

game dengan menekan tombol *Programmer*. Menu *Exit* digunakan untuk keluar dari aplikasi.

III.3.2. Activity Diagram

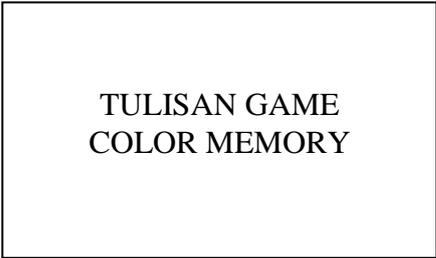
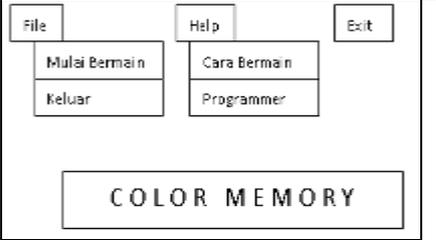
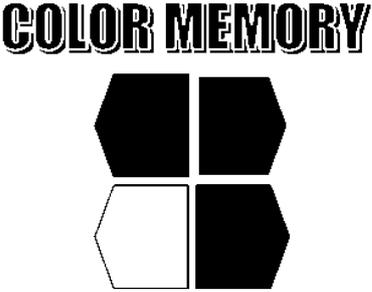
Berdasarkan *use case diagram*, maka mulailah dibuat *activity diagram*. Activity diagram juga sebagai teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. *Activity diagram* dapat dilihat pada gambar III.3.



Gambar III.3. Activity Diagram pada Aplikasi Game Color Memory

III.3.3. Story Board

Story board atau papan cerita dibuat untuk menjelaskan skenario secara lebih detail dari detik demi detik. Berikut *story board* perancangan aplikasi *game Color Memory* yang dijabarkan dalam bentuk tabel berikut ini.

NO	DESKRIPSI	VISUAL	AUDIO
1	<p>Pembuka (Splash Screen):</p> <p>Halaman pertama yang akan dirancang akan menampilkan tulisan GAME COLOR</p>		
2	<p>Form Menu Utama :</p> <p>Rancangan menu utama menampilkan 3 (tiga) pilihan menu yaitu <i>File</i>, <i>Help</i> dan <i>Exit</i>. <i>User</i> memulai bermain dengan mengklik <i>File</i>, lalu <i>Mulai Bermain</i>. Menu <i>Help</i></p>		
3	<p>Form Permainan :</p> <p>Pada menu permainan, pemain harus mengklik tombol urutan warna yang sesuai dengan kedipan yang ditampilkan oleh aplikasi. Pemain akan kalah bila</p>		<p>Suara pemilihan Warna</p>

4	<p>Cara Bermain</p> <p>Berisikan keterangan tentang cara bermain.</p>	<p>Cara Bermain :</p> <p>Anda harus mengklik tombol yang sama sesuai dengan urutan kedipan yang ditampilkan</p>	
5	<p>Programmer</p> <p>Berisikan informasi tentang pembuat program.</p>	<p>Programmer :</p> <p>Berisikan informasi biodata pembuat aplikasi permainan.</p>	

III.3.4. Perancangan Tampilan

Rancangan model tampilan dibuat untuk menggambarkan bentuk dari aplikasi yang akan dirancang. Aplikasi yang akan dirancang dapat dilihat sebagai berikut :

1. Rancangan *Form Splash Screen*

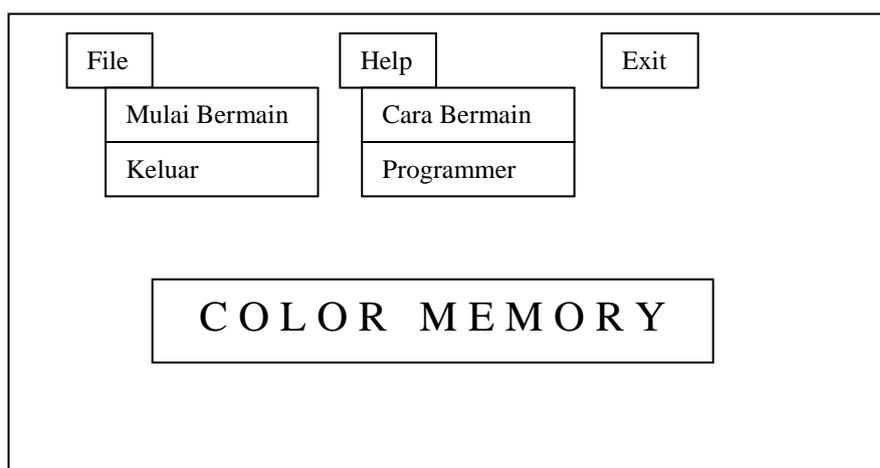
Rancangan *Splash Screen* berisikan nama dari aplikasi *game* yang dirancang. Tampilan ini keluar pada saat aplikasi *game* dieksekusi, tampil lalu hilang, setelah itu akan tampil menu utama. Tampilan *Splash Screen* dapat dilihat pada gambar III.4.



Gambar III.4. Rancangan *Form Splash Screen*

2. Rancangan *Form* Menu Utama

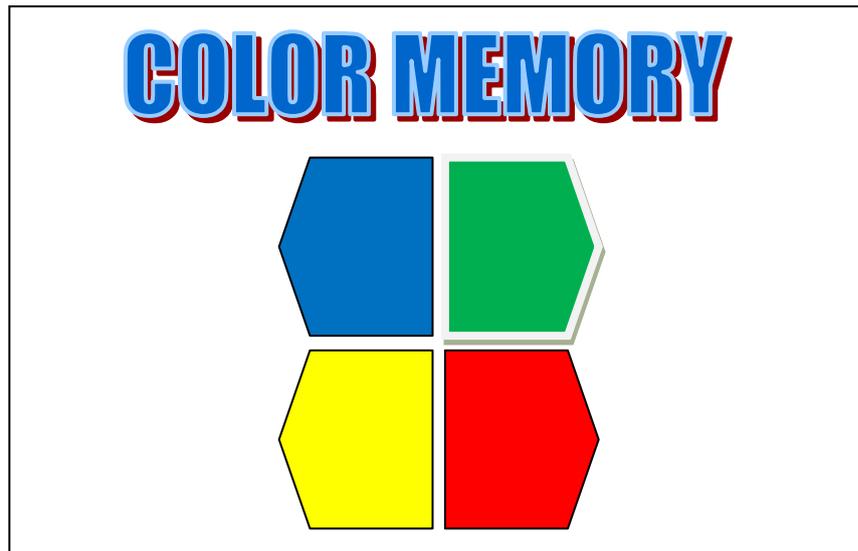
Rancangan menu utama menampilkan 3 (tiga) pilihan menu yaitu *File*, *Help* dan *Exit*. *User* memulai bermain dengan mengklik *File*, lalu Mulai Bermain. Menu *Help* berisikan keterangan cara bermain dan pembuat program, sedangkan menu *Exit* digunakan untuk keluar dari permainan. Rancangan *Form* Menu Utama dapat dilihat pada Gambar III.5.



Gambar III.5. Rancangan *Form* Menu Utama

3. Rancangan *Form* Permainan

Pada menu permainan, pemain harus mengklik tombol urutan warna yang sesuai dengan kedipan yang ditampilkan oleh aplikasi. Pemain akan kalah bila melakukan kesalahan tiga kali. Setiap klik pada kedipan yang benar akan ditandai dengan suara petunjuk, begitu juga sebaliknya. Adapun tampilan menu permainan adalah sebagai berikut :



4. Rancangan *Form* Cara Bermain

Pada menu ini berisikan keterangan cara memainkan permainan. Adapun tampilan menu Cara Bermain dapat dilihat pada gambar III.7. berikut ini.

Cara Bermain :

Anda harus mengklik tombol yang sama sesuai dengan urutan kedipan yang ditampilkan

Anda akan kalah bila mengklik 3 urutan kedipan yang salah

Gambar III.7. *Form* Permainan

5. Rancangan *Form Programmer*

Pada menu ini berisikan keterangan pembuat program. Adapun tampilan menu Cara Bermain dapat dilihat pada gambar III.8.

Programmer :

Berisikan informasi biodata pembuat aplikasi permainan.

Gambar III.8. *Form Programmer*