

BAB III

ANALISIS MASALAH DAN RANCANGAN PROGRAM

III.1. Analisis Masalah

Proses analisa sistem merupakan langkah kedua pada pengembangan sistem. Analisa sistem dilakukan untuk memahami informasi-informasi yang didapat dan dikeluarkan oleh sistem itu sendiri. Sistem permainan mencari 7 dragon ball belum begitu banyak diketahui oleh seorang pengguna (*user*), seorang pengguna ingin melakukan analisa terhadap proses permainan ini dengan metode *Depth First Search*. Karena dengan berkembangnya teknologi informasi secara otomatis akan menambah wawasan pribadi.

Permainan mencari 7 dragon ball sebenarnya belum pernah ada yang membuat, permainan mencari 7 dragon ball hampir mirip tetapi tidak sama dengan permainan *letter linker* yang terdapat didalam permainan *game house*. *letter linker* terdapat huruf mulai dari A sampai dengan Z, *user* dalam permainan untuk mencari satu objek huruf sesuai dengan kata yang disusun. Dan permainan *letter linker* yang terdapat pada *game house* ini penulis tidak paham metode apa yang digunakan dan teknologi pembuatan permainan *letter linker* ini juga tidak penulis ketahui. Oleh karena itu penulis membuat permainan dengan mencari 7 dragon ball karena permainan ini filmnya sering ditayangkan di televisi.

Untuk itu, sistem *game* (permainan) yang penulis rancang adalah permainan mencari 7 dragon ball dengan bahasa inggris. Permainan ini dikemas untuk mencari 7 dragon ball.

Dalam permainan mencari 7 dragon ball user harus mampu menggunakan radar dalam pencarian ini. Bentuk radar ini user yang harus menggerakannya ke kiri dan kekanan lalu ke atas dan ke bawah dengan metode DFS yang digunakan. Jadi seorang *user* harus mampu mencari ke 7 dragon ball yang dilambangkan dengan bola berdasarkan bintang dari masing-masing bola tersebut, dengan melakukan pergeseran menuju bola yang dicari berdasarkan radar, pergeseran ini yang menggunakan metode DFS, karena pergeseran ini berputar dengan rotasi dua kolom dan dua baris, jadi tidak sama dengan permainan *letter linker* yang hanya melakukan klik sekali jika ditemukan objek yang sama, jadi tidak ada pergeseran di permainan *letter linker*.

Dengan sistem permainan mencari ke 7 bola dragon ball ini adalah tahap pengembangan permainan yang berjenis puzzle seperti *letter linker*, analisa sistem merupakan hal yang harus dilakukan sebelum proses perancangan sistem.

III.2. Evaluasi Sistem Yang Berjalan

Sistem permainan mencari 7 *dragon ball* saat ini hanya sebatas teori seperti pemberitahuan tentang mencari huruf tersebut. Kelemahan dari sistem ini adalah susah mencari ke 7 dragon ball dalam satu wadah yang sudah ditentukan dalam permainan ini.

Maka solusi yang penulis buat untuk mengatasi masalah tersebut adalah membuat suatu sistem tentang pergeseran ke kiri dan ke kanan berdasarkan metode yang digunakan yaitu DFS.

III.3. Strategi Pemecahan Masalah

Adapun strategi pemecahan masalah dari sistem permainan mencari 7 bola dragon ball yang dirancang adalah sebagai berikut :

1. Teori tentang 7 dragon ball yang di implementasi ke permainan bola dragon ball dikemas melalui media, apabila data tentang mencari 7 dragon ball tersebut dapat digunakan sebagai analisa untuk berkonsentrasi mencari masalah dalam permainan, oleh karena itu perlu dibuat melalui media *adobe flash* sebagai ilustrasinya.
2. Agar mencari 7 dragon ball mudah dipahami dan dimengerti maka perlu dibuat pergeseran dengan metode DFS, agar setiap pengguna memainkan lebih mudah dalam pencarian tersebut.

III.4. Analisa Kebutuhan *Hardware* Dan *Software*

Kebutuhan non fungsional menjabarkan apa-apa saja yang harus dimiliki oleh sistem agar dapat berjalan. Analisis kebutuhan non fungsional bertujuan untuk mengetahui sistem seperti apa yang cocok diterapkan, perangkat keras dan perangkat lunak apa saja yang dibutuhkan serta siapa saja pengguna yang akan menggunakan sistem ini. Jika kebutuhan *hardware* dan *software* tidak dipenuhi maka permainan yang dibuat tidak dapat berjalan dengan semestinya.

1. Aspek Perangkat Keras

Perangkat keras adalah semua bagian fisik komputer dan dibedakan dengan data yang berada di dalamnya atau yang beroperasi di dalamnya, dan dibedakan

dengan perangkat lunak yang menyediakan instruksi untuk perangkat keras dalam menyelesaikan tugasnya.

Adapun kebutuhan *hardware* untuk menciptakan aplikasi atau perangkat lunak permainan dragon ball tersebut terdiri dari *Prosesor Intel Core I3, harddisk, Memory RAM 2 GB, Monitor 15, Keyboard dan Mouse*.

2. Aspek Perangkat Lunak (*software*)

Perangkat lunak adalah program yang digunakan untuk menjalankan perangkat keras. Tanpa adanya perangkat lunak ini komponen perangkat keras tidak dapat berfungsi, adapun aplikasi dan *software* yang digunakan dalam pembuatan game tersebut terdiri dari OS windows 7, adobe flash cs4.

III.5. Penerapan Algoritma DFS

Dalam permainan mencari 7 dragon ball dengan metode DFS. *Depth First Search* membangkitkan simpul berikutnya dari sebuah simpul yang sejauh ini terbaik diantara semua *leaf nodes* yang pernah dibangkitkan. Pertanyaannya bagaimana menentukan sebuah simpul terbaik saat ini. Penentuan simpul terbaik dapat dilakukan menggunakan informasi berupa.

Biaya perkiraan dari suatu simpul menuju ke goal atau gabungan antara biaya sebenarnya dan biaya perkiraan tersebut. Biaya perkiraan dapat diperoleh dengan menggunakan fungsi heuristic. Oleh karena itu algoritma Depth First Search ini digolongkan sebagai algoritma heuristic. Secara informal, algoritma Depth First Search diilustrasikan.

1. *Open* berisi initial state dan *Closed* masih kosong.

2. Ulangi sampai goal ditemukan atau sampai tidak ada nodes didalam *Open*.
 - a. Jika simpul tersebut sama dengan goal maka sukses.
 - b. Jika simpul tersebut sama dengan goal maka sukses.
 - c. Jika tidak, masukkan simpul tersebut sampai *Closed*.
 - d. Bangkitkan semua suksesor dari simpul tersebut.
 - e. Untuk setiap suksesor kerjakan :
 - 1) Jika suksesor tersebut belum pernah dibangkitkan, evaluasi suksesor tersebut, tambahkan ke *Open*, dan catat parent atau orang tuanya.
 - 2) Jika suksesor tersebut sudah pernah dibangkitkan, ubah parent-nya jika jalur melalui parent ini lebih baik daripada jalur melalui parent yang sebelumnya. selanjutnya, perbarui biaya untuk suksesor tersebut dan nodes lain yang berada di *level* bawahnya.

Pada algoritma tersebut diatas, *Open* dan senarai (*list*) yang digunakan untuk menyimpan simpul-simpul yang pernah dibangkitkan dan nilai heuristic-nya telah dihitung tetapi belum terpilih sebagai simpul terbaik (*best node*). Dengan kata lain *Open* berisi simpul-simpul yang masih memiliki peluang (peluangnya hasil terbuka) untuk terpilih sebagai simpul terbaik. sedangkan *Closed* adalah senarai untuk menyimpan simpul-simpul yang sudah pernah dibangkitkan dan sudah pernah terpilih sebagai simpul terbanyak. Artinya *Closed* berisi simpul-simpul yang tidak mungkin terpilih sebagai simpul terbaik (peluang untuk terpilih sudah tertutup).

DFS adalah sebuah algoritma pencarian yang digunakan dalam sebuah struktur pohon, Pada algoritma ini pencarian dilakukan dimulai dari simpul akar, dan kemudian pencarian dilakukan secara menyebar sesuai dengan tingkat dari masingmasing simpul di dalam pohon.

Input : n, l, j

Output : Susunan Gambar

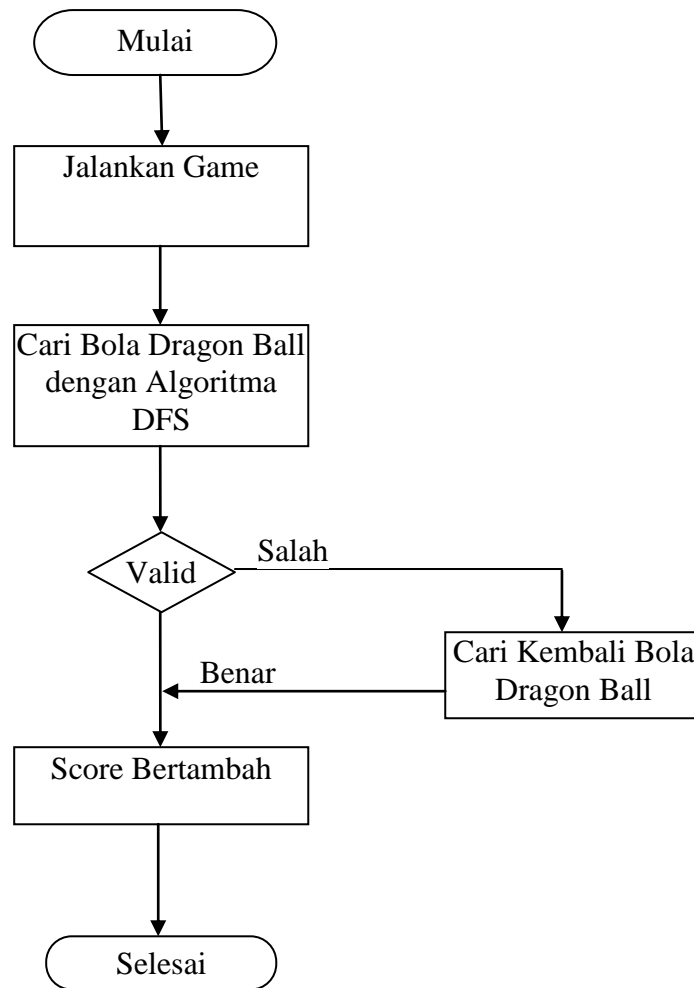
Proses : $For (i=0 < n; i++)$

$For (j=0; J < n; j++)$

$i=0$

III.6. Flowchart Algoritma DFS Dalam Permainan Dragon Ball

Pada perancangan mengenai pembuatan *flowchart* program. Maka, pada perancangan hanya akan dijelaskan tentang *form* permainan saja karena *form* yang lainnya hanyalah sebatas opsional. Pada akhirnya untuk melengkapi rancangan input, *flowchart* dari proses permainan dapat dilihat dibawah ini. Rancangan ini diharapkan dapat membantu untuk memahami proses jalannya program.







Gambar III.1. Flowchart Algoritma DFS

III.7. Storyboard

Storyboard mempermudah penjelasan tentang aplikasi yang dibuat karena dijelaskan setiap objek atau animasi yang terdapat didalam aplikasi tersebut, adapaun storybord tersebut adalah sebagai berikut :

Tabel III.1. *Storyboard*

No	Objek	Keterangan
1.		Objek ini adalah objek tombol play yang digunakan untuk masuk kedalam area permainan dragon ball.
2.		Objek ini adalah objek tombol help yang digunakan untuk masuk kedalam menu yang berisi informasi tentang permainan dragon ball
3.		Objek ini adalah objek tombol dragon ball digunakan untuk masuk kedalam area permainan dragon ball.
4.		Objek ini adalah objek tombol untuk membatalkan masuk kedalam area permainan dragon ball.
5.		Objek Multiplier ini untuk memberitahukan langkah-langkah dalam mencari bola dragon ball.
6.		Objek score ini memberikan informasi nilai dari setiap menyelesaikan mencari ketujuh dragon ball.
7.		Kotak ini digunakan untuk memberikan informasi tentang bola dragon ball
8.		Objek ini digunakan untuk melakukan pergeseran bola dragon ball.

9.		Objek radar ini digunakan untuk mendeteksi bola dragon ball yang asli.
10.		Objek bola ini adalah objek bola dragon ball yang asli.
11.		Objek bola ini adalah objek yang bukan bola dragon ball.
12.		Objek angka ini adalah objek yang menunjukkan timer pada permainan dragon ball.

Tabel III.1. *Storyboard*

III.8. Desain Sistem

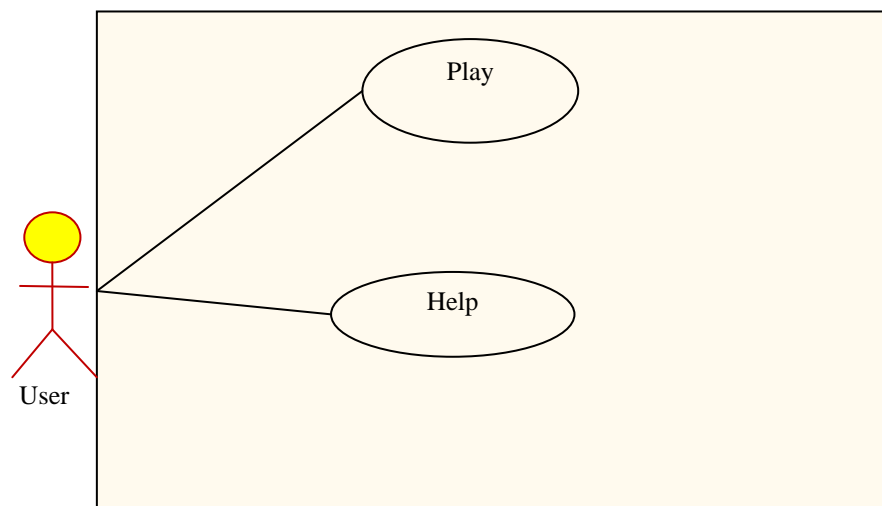
Setelah tahapan analisis sistem, maka selanjutnya dibuat suatu rancangan sistem. Perancangan sistem adalah tahapan yang berguna untuk memperbaiki efisiensi kerja suatu sistem yang telah ada. Pada perancangan sistem ini terdiri dari tahap perancangan yaitu :

1. Perancangan *Use Case Diagram*
2. Perancangan *Sequence Diagram*
3. Perancangan *Activity Diagram*

4. Perancangan *Output* dan *Input*

III.8.1. Diagram *Use Case*

Use case menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu. Sebuah *Use Case* mempresentasikan sebuah interaksi antara aktor dengan sistem dan menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Diagram *Use Case* tersebut dapat dilihat pada gambar III.2.



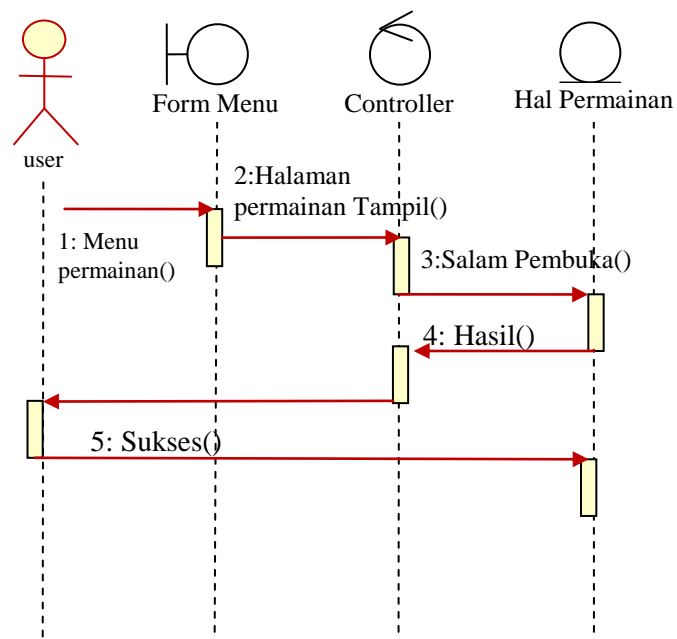
Gambar III.2. Diagram *Use Case*

III.8.2. *Sequence Diagram*

Sequence diagram menunjukkan bagaimana operasi yang dilakukan secara detail. *Sequence* diagram menjelaskan interaksi obyek yang disusun dalam suatu urutan waktu. Urutan waktu yang dimaksud adalah urutan kejadian yang dilakukan oleh seorang *actor* dalam menjalankan sistem, adapun *sequence* yang dilakukan adalah sebagai berikut.

1. *Sequence Dragon Ball*

Permainan mencari 7 dragon ball digunakan halaman pembuka untuk memunculkan permainan dragon ball, yang terdiri dari salam pembuka, untuk lebih jelasnya dapat dilihat pada gambar III.3.

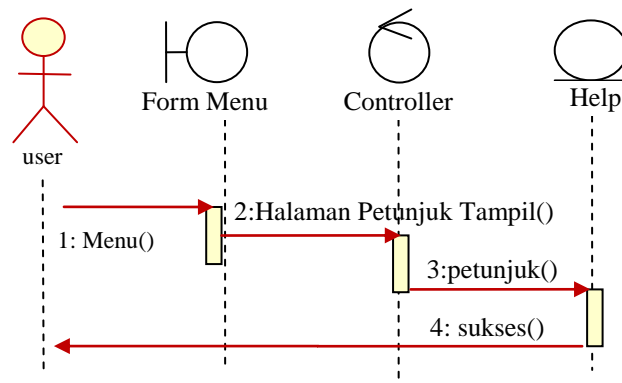


Gambar III.3. *Sequence Permainan Dragon Ball*

Dari gambar III.3. menunjukkan bahwa seorang pengguna jika ingin masuk ke halaman permainan dragon ball harus terlebih dahulu melakukan klik pada menu permainan dragon ball.

2. *Sequence Help*

Sequence terdiri dari tentang bagaimana cara memainkan dragon ball, untuk lebih jelasnya dapat dilihat pada gambar III.4.



Gambar III.4. Sequence Help

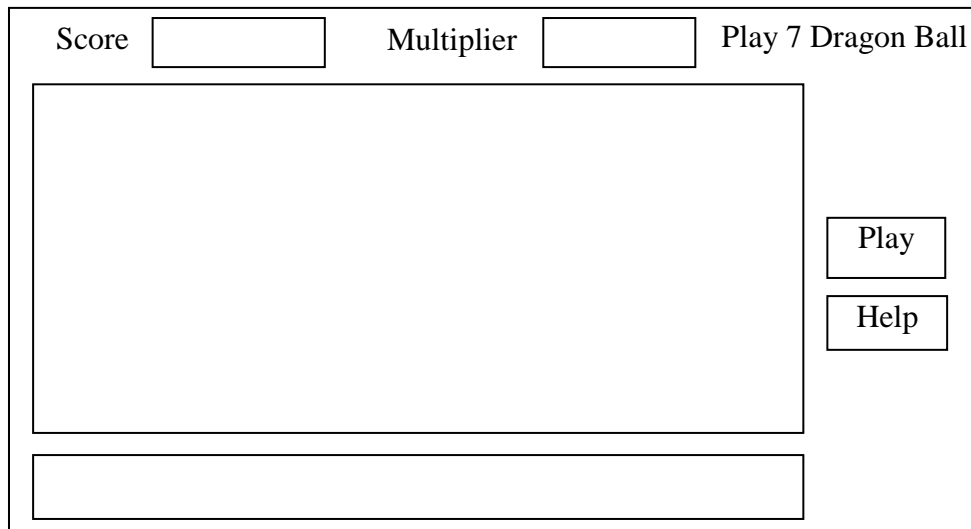
Dari gambar III.4. jika pengguna ingin melihat bagaimana cara memainkan permainan dragon ball, pengguna dapat masuk kedalam halaman petunjuk. Jika tidak ada masalah maka pengguna atau user dapat melihat informasi tentang penggunaan permainan dragon ball ini.

III.9. Rancangan Program

Rancangan program permainan dragon ball, teknis sistem secara fisik yaitu perancangan bentuk fisik atau bagan arsitektur sistem yang diusulkan. Dalam merancang suatu sistem perlu diketahui hal yang akan menunjang sistem, agar dapat mempermudah untuk memainkan permainan dragon ball ini. Permainan mencari 7 dragon ball ini diharapkan dapat mempermudah dalam hal penyajian, pelayanan dan pembuatan berbagai bentuk bola dragon ball yang dibutuhkan.

1. Rancangan *Form* Menu Utama

Form menu utama ini dirancang sebagai pusat dari semua menu pada aplikasi permainan pencarian dragon ball, untuk jelasnya dapat dilihat pada gambar III.5.



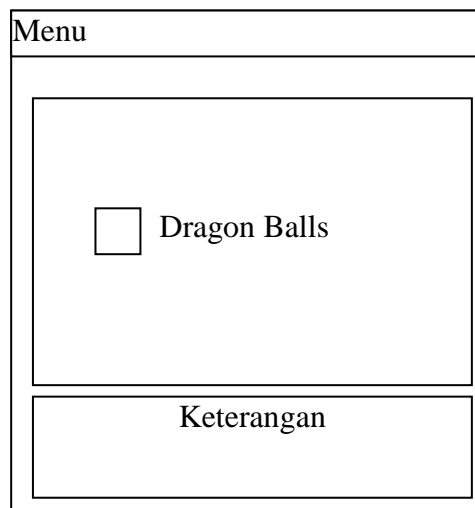
The diagram shows a main menu form with a rectangular border. At the top left, the text 'Score' is followed by a small empty rectangular box. To its right, the text 'Multiplier' is followed by another small empty rectangular box. Further right, the text 'Play 7 Dragon Ball' is displayed. Below these elements is a large, empty rectangular area. On the right side of this large area, there are two vertically stacked rectangular buttons labeled 'Play' and 'Help'. At the bottom of the form, there is a wide, empty rectangular bar.

Gambar III.5. Rancangan *Form* Menu Utama

Rancangan menu utama yang terdapat pada gambar III.5. terdiri dari area dragon ball, help. Dari masing-masing *tool* tersebut tugasnya berbeda-beda.

2. Rancangan *Form* Menu Dragon Ball

Form menu dragon ball digunakan sebagai salam pembuka dari permainan dragon ball, untuk lebih jelasnya dapat dilihat pada gambar III.6.



The diagram shows a 'Menu' form with a rectangular border. At the top left, the text 'Menu' is displayed. Below this, there is a large, empty rectangular area. In the center of this area, there is a small square icon followed by the text 'Dragon Balls'. At the bottom of the form, there is a wide, empty rectangular bar labeled 'Keterangan'.

Gambar III.6. Rancangan *Form* Menu Dragon Ball

Pada gambar III.6. jika pengguna ingin memainkan permainan dragon ball ini, pengguna harus masuk dulu ke dalam menu dragon ball ini, pengguna diminta untuk mengklik game dragon ball, setelah itu pengguna dapat memainkan permainan dragon ball ini.

3. Rancangan *Form* Halaman Permainan Dragon Ball

Dalam perancangan *form* Teori ini terdiri dari beberapa halaman untuk mencari ke 7 bola dragon ball, untuk mencari bola dapat menggunakan radar yang telah disediakan dengan mengklik tombol kiri jika ingin arahnya ke kiri dan jika user ingin arahnya ke kanan dalam memutar user dapat menekan tombol ctrl lalu klik tombol kiri. Untuk lebih jelasnya dapat dilihat pada gambar III.7.

The form layout is as follows:

- Score
- Multiplier
- A grid of 13 rows and 16 columns. A circle is located in the first cell of the second row, which is highlighted with a thick border.
- Play
- Help
- Keterangan Permainan Dragon Ball

Gambar III.7. Rancangan *Form* Halaman Permainan Dragon Ball

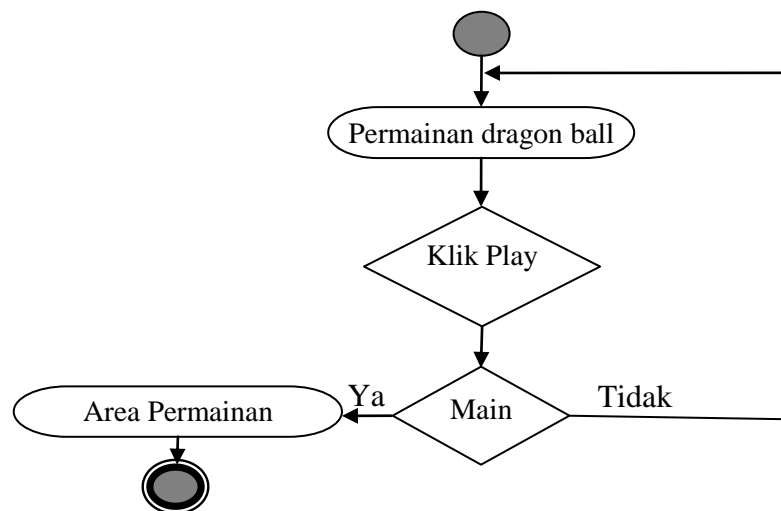
Rancangan *form* halaman permainan dragon ball yang terdapat pada gambar III.7. terdiri dari 16 kolom dan 13 baris, untuk radar berbentuk bulat.

Dalam permainan ini pemain harus dapat mencari ke 7 dragon ball tersebut berdasarkan radar dalam pencarian bola tersebut.

III.10. Activity Diagram

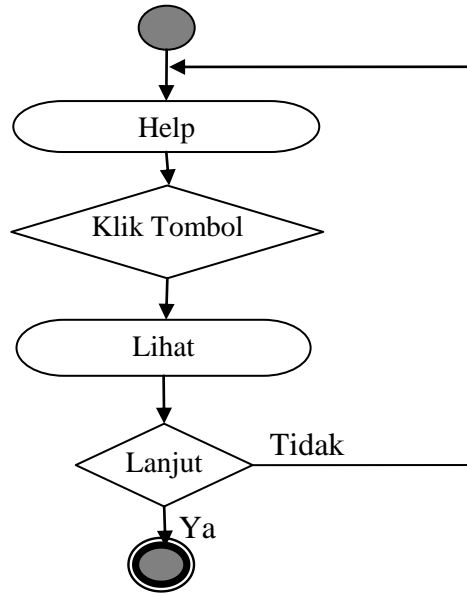
Logika program dari sistem yang diusulkan akan digambarkan dalam sebuah *activity diagram*. *Activity diagram* ini akan menjelaskan setiap kegiatan yang akan dilakukan pengguna pada sistem nantinya. Dengan menggambarkan setiap aktivitas dari sistem diharapkan sistem yang akan dibangun lebih mudah dipahami. Adapun logika program yang digambarkan dengan menggunakan *activity* pada UML. *Activity diagram* permainan dragon ball dapat dilihat pada gambar III.8.

III.10.1. Activity Diagram Permainan Dragon Ball



Gambar III.8. Activity Diagram Permainan Dragon Ball

III.10.2. Activity Diagram Help



Gambar III.9. Activity Diagram Help