BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Pada bagian analisis akan menjelaskan beberapa macam analisis, diantaranya analisis *game* sejenis, analisi *game* yang akan dibangun sampai dengan analisis berbagai kebutuhan fungsional maupun non-fungsional.

3.1.1 Analisis Masalah

Berdasarkan analisis masalah yang ada pada latar belakang sebelumnya terdapat beberapa permasalahan yang telah di analisis yaitu Menurut Kanti Waluyo Cerita dan tokoh Gatotkaca sekarang ini sudah mulai terlupakan, seiring dengan perkembangan zaman banyak masyarakat terutama remaja kurang mengenal bahkan tidak mengetahui tokoh pewayangan yang satu ini. Dikarenakan masyarakat sekarang lebih menyukai sesuatu yang modern, simpel tetapi berteknologi tinggi. Jika hal ini dibiarkan, dampak yang akan terjadi yaitu salah satu kebudayaan indonesia yang berasal dari jawa tengah yaitu tokoh gatotkaca akan semakin dilupakan dan bukan tidak mungkin anak-anak kita nanti tidak akan tahu lagi siapa itu Gatotkaca.

3.1.2 Analisis Game Sejenis

Analisis *game* sejenis dilakukan sebagai acuan sebagai *game* yang akan dibangun yaitu *game* Petualangan Gatotkaca. *Game* sejenis yang akan jadi acuan adalah *game Andrio's World*.

3.1.2.1 Pengenalan Game Andrio's World

Game ini merupakan salah satu *game* yang menarik, dimana pemain harus melewati rintangan seperti tebing, menghindari musuh, dan mengambil koin disetiap *level* yang di lewatinya.

3.1.2.1.1 Gameplay Andrio's World

Game ini mempunyai karakter yang bernama Andrio, pada *game* ini Andrio bertugas untuk mengumpilkan koin yang berada disetiap *level*-nya. Namun pada saat mengambil koin tidaklah mudah karena disetiap perjalanannya Andrio harus melawan musuh atau menghindari musuh yang dapat mengurangi

nyawa karakter Andrio. Pemain sudah disediakan tombol-tombol yang tersedia untuk menjalankan karakter tersebut yaitu terdapat tombol tanda panah kiri dan kanan berfungsi untuk menggerakan karakter kekiri dan kekanan, kemudian terdapat tombol untuk meloncat dan aksi disebelah kanan.

3.1.2.1.2 Kelebihan dan Kekurangan Game Andrio's World

Adapaun kelebihan dan kekurangan *game Andrio's World* adalah sebagai berikut :

a. Kelebihan *Game Andrio's World*
    1. *Game Andrio's World* memiliki tampilan yang menarik.
    2. *Game* ini tidak terlalu sulit untuk dimainkan.
    3. Game ini tujuannya hanya mengambil koin.
b. Kekurangan *Game Andrio's World*
    1. Karakter pada *game* Andrio tidak mempunyai action yang hanya bisa melompat untuk menghindari musuhnya.
    2. *Game* Andrio hanya diberi nyawa 2, sehingga karakter jika terkena musuh akan langsung mati.
    3. Pemain tidak bisa melawan musuh jadi pemain harus menghidari musuh agar tidak mati.

3.1.2.1.3 Tampilan Game Andrio's World

Tampilan menu utama pada *game Andrio's World* terdapat gambar tokoh karakter pada menu utama dan beberapa menu yaitu diantaranya menu play, setting, dan high score. Lebih jelasnya tampilan menu utama pada *game Andrio's World* dapat dilihat pada gambar 3.1 :



Gambar 3. 1 Tampilan Awal *Game Andrio's World*

Tampilan gameplay permainan *Andrio's World* yaitu bertujuan untuk mengambil koin pada tiap level dan melewati musuh yang ada pada tiap level. Untuk lebih jelasnya tampilan gameplay pada *Andrio's World* bisa dilihat pada gambar 3.2:



Gambar 3. 2 *Gameplay Andrio's World*

Tampilan gameplay permainan *Andrio's World* bisa dilihat pada gambar 3.3 :



Gambar 3. 3 ***Gameplay Andrio's World***

3.1.2.2 Pengenalan Game Perjuangan Gatotkaca

Game ini merupakan *game* berbasis *android*, *game* ini menceritakan tentang cerita pewayangan yaitu pewayangan Gatotkaca.

3.1.2.2.1 Gameplay Game Perjuangan Gatotkaca

Game perjuangan Gatotkaca menceritakan seseorang wanita yang bernama Dewi Pregiwa yang telah diculik oleh seorang raja yang jahat, seorang wanita itu adalah kekasih seorang kesatria yang bernama Gatotkaca. Setelah Gatotkaca tahu kalau kekasihnya diculik, Gatotkaca pun bergegas untuk mencari sang kekasih yang bernama Dewi Pregiwa. Pada permainan ini pemain hanya cukup menekan tombol yang sudah disediakan, tombol tersebut berfungsi untuk menjalankan karakter naik dan turun ketika ditekan tombol tersebut karakter akan naik dan ketika tombol dilepas maka karakter akan turun.

3.1.2.2.2 Kelebihan dan Kekurangan Game Perjuangan Gatotkaca

Adapaun kelebihan dan kekurangan *game* Perjuangan Gatotkaca adalah sebagai berikut :

a. Kelebihan
    1. Mengenalkan sejarah pewayangan Indonesia kepada masyarakat melalu *game*.
    2. Menceritakan cerita pewayangan Gatotkaca.
b. Kekurangan
    1. *User interface* pada *game* ini kurang menarik.
    2. *Game* ini hanya terbang saja dan tidak ada musuh.

3.1.2.2.3 Tampilan Game Perjuangan Gatotkaca

Pada tampilan menu utama game perjuangan gatotkaca terdapat beberapa menu yaitu menu informasi, menu play game, dan menu highscore. Berikut adalah tampilan menu utama pada *game* Perjuangan Gatotkaca bisa dilihat pada gambar 3.4 :



Gambar 3. 4 Tampilan awal *Game* Perjuangan Gatotkaca

Pada tampilan gameplay perjuangan gatotkaca yaitu pemain hanya cukup menekan touch tanda panah untuk menghindari rintangan yang ada pada permainan. Untuk lebih jelasnya berikut adalah tampilan *gameplay* Perjuangan Gatotkaca bisa dilihat pada gambar 3.5 :



Gambar 3. 5 Tampilan *Gameplay* Perjuangan Gatotkaca

3.1.3 Analisis Game Yang Dibangun

Analisis game yang dibangun akan menjelaskan tahap analisis *game* yang akan dibuat, diantaranya pengenalan perancangan, metode dan algoritma yang akan digunakan pada *game* yang akan dibangun.

3.1.3.1 Pengenalan Tokoh Utama

Gatotkaca adalah tokoh cerita pewayangan yang berasal dari jawa, *Game* Gatotkaca ini bisa menjadi pilihan karakter utama karena Gatotkaca merupakan legenda pewayangan Indonesia, khususnya dari pulau Jawa yang sudah sangat dikenal oleh masyarakat sebagai sosok ksatria yang berani tak mengenal takut, teguh, tangguh, cerdik pandai, gagah dan sangat kuat. Ia sangat sakti, sehingga digambarkan sebagai ksatria yang mempunyai "otot kawat balung besi"[5].

3.1.3.2 Jalan Cerita

Game ini menceritakan seorang prajurit yang gagah perkasa bernama Gatotkaca yang sebut sebagai kesatria di Pringgadani yang diutus oleh raja ditugaskan untuk mengambil senjata pusaka bernama KontaWijaya yang diambil oleh seorang raja bernama Adipatikarna, karena senjata itu sangat sakti mandraguna, hanya gatotkaca yang bisa melawan para kurawa dan raja Adipatikarna.

3.1.3.3 Gameplay

Pada permainan ini, tugas utama pemain atau *actor* yaitu melakukan perjalanan untuk mengambil pusaka, melewati rintangan, dan menghadapi musuh. Terdapat 3 *level*, adapun pembagian *level* nya yaitu sebagai berikut :

3.1.3.3.1 Gameplay Level 1

Level ke 1, pada stage ini pemain diharuskan mengambil item pusaka dengan cara melewati beberapa tempat seperti tebing telah disimpan pada tempat-tempat tertentu. Di *level* l ini pemain bertugas untuk mengumpulkan Pusaka bernama Kotang Antakusuma dimana pemain harus mengumpulkan 5 benda pusaka tersebut, di level ini terdapat hewan buas yang akan menghalangi pemain untuk mengambil pusaka tersebut dan terdapat item koin untuk menambah score kepada pemain. Ketika pemain sudah dapat mengumpulkan item baju pusaka

maka akan muncul sebuah portal yang bercahaya dimana pemain harus masuk kedalam portal tersebut untuk melanjutkan *level* berikutnya.

3.1.3.3.2 Gameplay Level 2

Level ke 2, di stage ini pemain diharuskan mengambil benda pusaka yang bernama Caping Basunanda yaitu sebuah pelindung kepala, dimana seperti pada level sebelumnya pemain harus mengumpulkan 5 benda pusaka yang telah disediakan. Pemain akan menghadapi lebih kesulitan dalam upayanya mengambil pusaka. Di stage ini dimunculkan musuh kurawa bernama petruk sehingga pemain selain diharuskan untuk mengambil pusaka pemain juga harus dapat menghindari serangan musuh atau melawan musuh. Jika pemain terkena serangan musuh karna tidak mampu mengindar maka jumlah atau kesempatan nyawa yang dimiliki oleh pemain itu akan berkurang setiap pemain terkena serangan musuh tersebut, namun pemain bisa mengambil item kendi untuk menambah darah sehingga pemain tidak akan mati, dan di level ini terdapat koin yang disediakan untuk menambah score. Seperti halnya di stage awal jika pemain sudah bisa menemukan dan mengumpulkan benda pusaka yang sudah disediakan maka portal selanjutnya akan terbuka dan menandakan bahwa pemain telah berhasil melewati *level* kedua dan mampu melanjutkan ke *level* berikutnya.

3.1.3.3.3 Gameplay Level 3

Level 3, pada *level* ini tingkat kesulitan yang dihadapai pemain akan jauh lebih sulit dari pada *level-level* sebelumnya, karena pada *level* ini akan dimunculkan banyak musuh para kurawa yaitu petruk, hewan buas serigala dan burung, dan seperti pada stage sebelumnya gatotkaca harus mengumpulkan benda pusaka, pada level ini pemain harus mengambil pusaka bernama Pada Kacarma, jika pemain sudah mengumpulkan 5 pusaka maka portal akan terbuka. Dan terdapat koin untuk menambah score pada pemain dan juga item kendi untuk menambah darah pemain jika pemain terkena musuh sehingga darah berkurang. Jika pemain sudah mengumpulkan 5 benda pusaka tersebut maka portal terbuka untuk melanjutkan stage terakhir yaitu melawan raja adipatikarna.

3.1.3.3.5 Gameplay Melawan Raja

Level 4 merupakan *level* terakhir dari permainan dimana Gatotkaca harus melawan raja yang telah mengambil pusaka KontaWijaya nama raja tersebut adalah Adipatikarna, Gatotkaca harus melawan dan menghindari serangan dari raja, Keberhasilan *level* ini ditandai dengan kematian sang raja adipatikarna dan akhirnya senjata kontawijaya yang diambil Adipatikarna akhirnya telah direbut kembali oleh Gatotkaca.

3.1.3.3.4 Game Control

Game ini menggunakan tombol *touchscreen* untuk menggerakan dan memukul kepada lawan atau musuhnya. Berikut adalah tampilan *game control* Perjuangan Gatotkaca bisa dilihat pada gambar 3.6 :



Gambar 3. 6 *Game control* Petualangan Gatotkaca

3.1.4 Analisis Algoritma A* (A Bintang)

Algoritma A Star (A*) adalah algoritma pencarian terbaik dalam mencari jalur terpendek dengan perhitungan terkecil pada jalur dengan simpul awal menuju simpul akhir disebut dengan Algoritma A. Lalu dengan optimasi heuristik, disebut dengan A Star (A*). Rumus perhitungan pada Algoritma A Star (A*) menggunakan persamaan (2.1) dapat dilihat pada gambar 3.7:

$$F(x) = G(x) + H(x)$$

Gambar 3. 7 Rumus Perhitungan A*

Dimana :

a. G(x) adalah nilai pada pergerakan simpul awal menuju simpul berikutnya.

b. H(x) adalah perkiraan nilai pergerakan simpul awal menuju tujuan akhir simpul. Fungsi ini seringkali disebut dengan fungsi heuristik, dinamakan heuristik karena perhitungan tersebut berdasarkan perkiraan (guess).
c. F(x) adalah jumlah nilai dari fungsi G(x) dan H(x). dengan nilai terkecil F(x) adalah jalur terpendek menuju tujuan akhir.

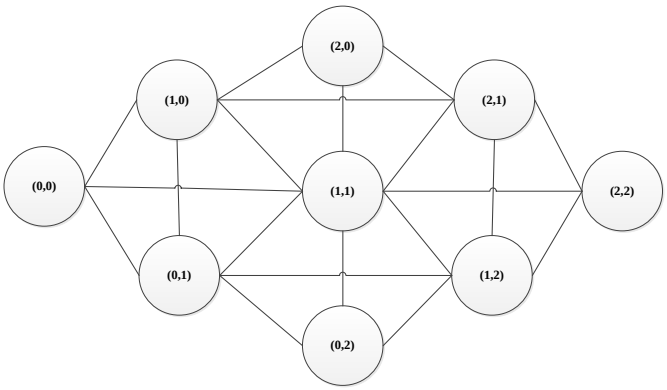
Terdapat ketentuan pada grafik agar algoritma A Star (A*) ini bila diterapkan akan selalu mendapatkan jalan yang terpendek. Ketentuan tersebut yang harus dipenuhi pada grafik yaitu.

1. Setiap simpul (node) dalam grafik memiliki jumlah terbatas pada area pencariannya.
2. Pada pencarian terdapat jalan yang dilalui untuk mencapai tujuan.
3. Fungsi F(x) pada grafik bernilai rendah dari pada fungsi F(x) pada pencarian sebelumnya.

3.1.5 Struktur data Algoritma A*

Struktur data pada algoritma A* bisa diasumsikan misal pada arena permainan dibagi kedalam sebuah tilemap berbentuk array dua dimensi, setiap tile memiliki koordinat X dan Y dan dapat digambarkan pada struktur data graph. Graph adalah suatu struktur data yang berbentuk network jaringan dimana hubungan antar elemen-elemennya adalah *many-to many*. Contoh penerapan struktur data graph pada arena permainan yang terdapat pada petualangan gatotkaca dapat dilihat pada gambar 3.8 :

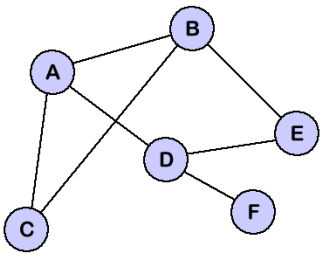
	0	1	2
0	(0,0)	(1,0)	(2,0)
1	(0,1)	(1,1)	(2,1)
2	(0,2)	(1,2)	(2,2)



Gambar 3. 8 Contoh penerapan Stuktur data graph

Secara umum terdapat dua macam representasi dari struktur data graf yang dapat diimplementasikan.

Pertama, disebut *adjacency list*, yaitu list yang diimplementasikan dengan menampilkan masing-masing simpul sebagai sebuah struktur data yang mengandung senarai dari semua simpul yang saling berhubungan.Yang kedua adalah representasi berupa adjacency matrix dimana baris dan kolom dari matriks tersebut merepresentasikan simpul awal dan simpul tujuan dan sebuah entri didalam senarai yang menyatakan apakah terdapat sisi di antara kedua simpul tersebut. Berikut ini adalah contoh adjacency list dapat dilihat pada gambar 3.9 :

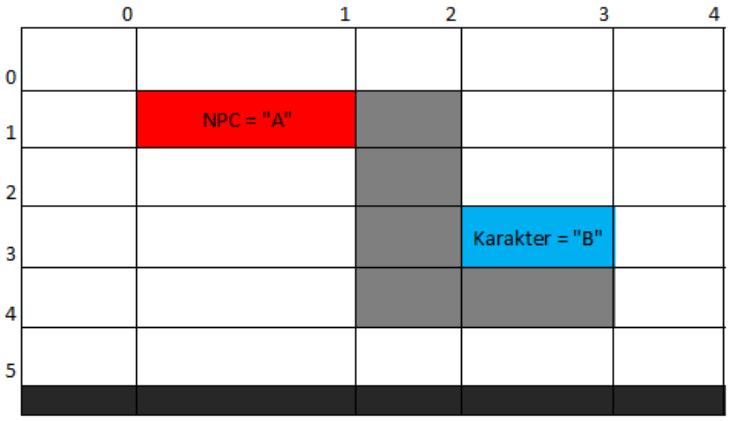


Gambar 3. 9 contoh adjacency list

3.1.6 Penerapan Algoritma A* Pada Game Petualangan Gatotkaca

Penerapan Algortima A* pada *Game* Petualangan Gatotkaca secara ringkas langkah demi langkahnya adalah sebagai berikut. Algoritma A* pada *game* Petualangan Gatotkaca diterapkan pada musuh atau disebut jga NPC (*Non Playable Character*) yaitu dimana algoritma tersebut tidak diterapkan pada karakter. Pada algoritma ini yaitu menerapkan musuh sebagai pencari jalur terpendek atau pathfinding untuk mendekati target yaitu karakter. Musuh atau

NPC sebagai Node “A” dan karakter sebagai Node “B”, contoh penerapan NPC dan karakter pada algoritma A* bisa dilihat pada gambar 3.10 :



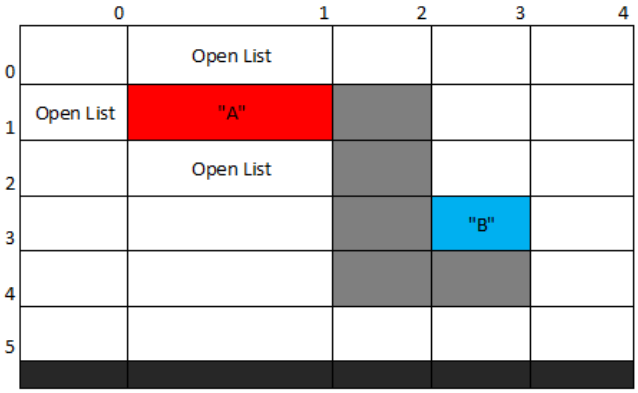
Gambar 3. 10 Penerapan NPC dan Karakter pada algoritma A*

Gambar 3.10 dijelaskan petak yang berwarna merah “A” adalah simpul yang sedang dijalankan dalam algoritma pencarian terpendek, Harga (F) adalah nilai yang diperoleh dari penjumlahan nilai G, jumlah nilai tiap simpul dalam jalur terpendek dari starting point ke A, dan H, jumlah nilai perkiraan dari sebuah simpul ke simpul tujuan. Sedangkan petak-petak yang berwarna putih adalah sebagai representasi dari area pathpinding. Petak berwarna kuning adalah (unwalkable node) disebut juga sebagai penghalang.

1. Diawali dengan menempatkan A pada starting point, kemudian memasukkan seluruh simpul yang bertetangga dan tidak memilik atribut rintangan dengan A ke dalam open list.
2. Kemudian mencari nilai H terkecil dari simpul-simpul dalam open list tersebut.
3. Kemudian memindahkan A ke simpul yang memiliki nilai H terkecil. Simpul sebelum A disimpan sebagai parent dari A dan dimasukkan ke dalam closed list. Jika terdapat simpul lain yang bertetangga dengan A (yang sudah berpindah) namun belum termasuk kedalam anggota open list, maka masukkan simpul-simpul tersebut ke dalam open list.
4. Setelah itu, bandingkan nilai G yang ada dengan nilai G sebelumnya (pada langkah awal, tidak perlu dilakukan perbandingan nilai G). Jika nilai G

sebelumnya lebih kecil maka A kembali ke posisi awal. Simpul yang pernah dicoba dimasukkan ke dalam closed list.

Berikut adalah penerapan Closed List dan Open List bisa dilihat pada gambar 3.11 :



Gambar 3. 11 Penerapan Closed List dan Open List

Pada gambar 3.11 menjelaskan node berwarna merah "A" dimasukan kedalam Closed List, kemudian Jika terdapat simpul lain yang bertetangga dengan A (yang sudah berpindah) namun belum termasuk kedalam anggota open list, maka masukkan simpul-simpul tersebut ke dalam open list.

Langkah berikutnya adalah memilih node yang memiliki nilai f terendah di dalam open list dengan menggunakan rumus (f = g + h), bandingkan nilai G yang ada dengan nilai G sebelumnya (pada langkah awal, tidak perlu dilakukan perbandingan nilai G). Jika nilai G sebelumnya lebih kecil maka A kembali ke posisi awal. Simpul yang pernah dicoba dimasukkan ke dalam closed list. Hal tersebut dilakukan berulang-ulang hingga terdapat solusi atau tidak ada lagi simpul lain yang berada pada open list.

Kinerja algoritma A* bergantung pada fungsi heuristik yang digunakan, game ini menggunakan fungsi heuristic Manhattan Distance, dengan fungsi heuristic yang digunakan menggunakan persamaan (2.2) bias dilihat sebagai berikut:

h_diagonal(n) = min(abs(n.x-goal.x) + abs(n.y-goal.y))

h_orthogonal(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))

h(n) = h_diagonal(n) + (h_orthogonal (n) – (2 * h_diagonal(n)))

Berikut ini langkah perhitungan menggunakan fungsi heuristic Manhattan Distance, yaitu dengan menentukan titik koordinat.

a) Koordinat n(0,0)

Nilai n.x = 0

Nilai n.y = 0

Simpul tujuan(3,3) sehingga goal.x = 3, goal.y = 3

g (0,0) = 0,3

h_orthogonal(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))

h_orthogonal(0,0) = (abs(0 - 3) + abs(0 - 3))

= (abs(-3) + abs(-3))

= 6

h_diagonal(n) = -(abs(n.x-goal.x) + abs(n.y-goal.y))

h_diagonal(0,0) = -(abs(0 - 3)+abs(0 - 3))

= -(abs(-3)+abs(-3))

= -6

h(n) = h_diagonal(n) + (h_orthogonal (n) – (2 * h_diagonal(n)))

h(0,0) = (-6) + (6-(2*(-6)))

= (-6) + 22

= 16

f (0,0) = g (n) + h (n)

= 0,3 + 16

= 16,3

b) Koordinat n(0,1)

Nilai n.x = 0

Nilai n.y = 1

Simpul tujuan (3,3) sehingga goal.x = 3, goal.y = 3

g (n) = 1

h_orthogonal(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))

h_orthogonal(0,1) = (abs(0 - 3) + abs(1 - 3))

= (abs(-3) + abs(-2))

= 5

h_diagonal(n) = min(abs(n.x-goal.x), abs(n.y-goal.y))

h_diagonal(0,1) = min(abs(0 - 3) + abs(1 - 3))

= min(abs(-3) + abs(-2))

= -5

h(n) = h_diagonal(n) + (h_orthogonal (n) – (2 * h_diagonal(n)))

h(0,1) = (-5) + (5-(2*(-5)))

= -5 + 15

= 10

f (0,1) = g (1,0) + h (1,0)

= 1 + 10

= 11

c) Koordinat n(0,2)

Nilai n.x = 0

Nilai n.y = 2

Simpul tujuan (3,3) sehingga goal.x = 3, goal.y = 3

g(n) = 1,3

h_orthogonal(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))

h_orthogonal(0,2) = (abs(0 - 3) + abs(2 - 3))

= (abs(-3) + abs(-1))

= 4

h_diagonal(n) = min(abs(n.x-goal.x) + abs(n.y-goal.y))

h_diagonal(0,2) = min(abs(0 - 3) + abs(2 - 3))

= min(abs(-3) + abs(-1))

= -4

h(n) = h_diagonal(n) + (h_orthogonal (n) – (2 * h_diagonal(n)))

h(0,2) = (-4) + (4-(2*(-4)))

= -4 + 12

= 8

f (0,2) = g (n) + h (n)

= 1,3 + 8

= 9.3

d) Koordinat n(2,1)

Nilai n.x = 2

Nilai n.y = 1

Simpul tujuan (3,3) sehingga goal.x = 3, goal.y = 3

h_orthogonal(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))

h_orthogonal(2,1) = (abs(2 - 3) + abs(1 - 3))

= (abs(-1) + abs(-2))

= 3

h_diagonal(n) = min(abs(n.x-goal.x) + abs(n.y-goal.y))

h_diagonal (2,1) = min(abs(2 - 3) + abs(1 - 3))

= min(abs(-1) + abs(-2))

= -3

h(n) = h_diagonal(n) + (h_orthogonal (n) – (2 * h_diagonal(n)))

h(2,1) = (-3) + (3-(2*(-3)))

= -3 + 9

= 6

f (2,1) = g (2,1) + h (2,1)

= 1 + 6

= 7

e) Koordinat n(1,2)

Nilai n.x = 1

Nilai n.y = 2

Simpul tujuan(3,3) sehingga goal.x = 3, goal.y = 3

h_orthogonal(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))

h_orthogonal(1,2) = (abs(1 - 3) + abs(2 - 3))

= (abs(-2) + abs(-1))

= 3

h_diagonal(n) = min(abs(n.x-goal.x) + abs(n.y-goal.y))

h_diagonal(1,2) = (abs(1 - 3) + abs(2 - 3))

= min(abs(-2) + abs(-1))

= -3

h(n) = h_diagonal(n) + (h_orthogonal (n) – (2 * h_diagonal(n)))

h(1,2) = (-3) + (3-(2*(-3)))

= -3 + 9

= 6

f (1,2) = g (1,2) + h (1,2)

= 1+ 6

= 7

f) Koordinat n(2,2)

Nilai n.x = 2

Nilai n.y = 2

Simpul tujuan(3,3) sehingga goal.x = 3, goal.y = 3

g(n) = 1,3

h_orthogonal(n) = (abs(n.x-goal.x) + abs(n.y-goal.y))

h_orthogonal(2,2) = (abs(2 - 3) + abs(2 - 3))

= (abs(-1) + abs(-1))

= 2

h_diagonal(n) = min(abs(n.x-goal.x) + abs(n.y-goal.y))

h_diagonal(2,2) = -(abs(2 - 3) + abs(2 - 3))

= -(abs(-1) + abs(-1))

= -3

h(n) = h_diagonal(n) + (h_orthogonal (n) – (2 * h_diagonal(n)))

h(2,2) = (-2) + (2-(2*(-2)))

= -2 + 8

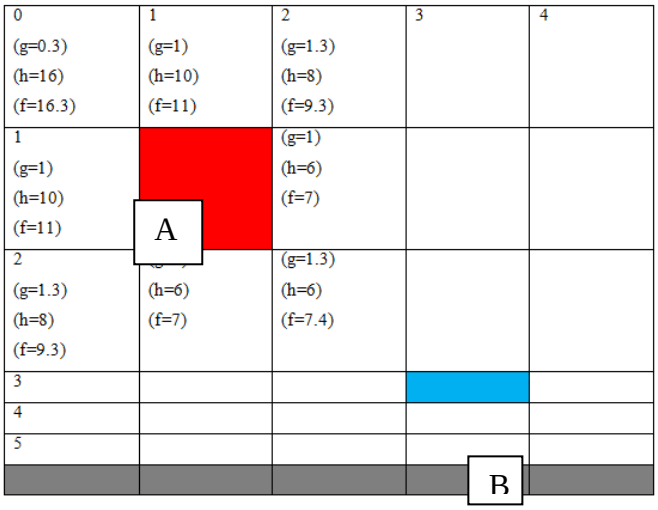
= 6

f (2,2) = g (2,2) + h (2,2)

= 1,3 + 6

= 7,4

Berikut hasil perhitungan fungsi heuristic Non-Manhattan Distance dapat dilihat pada gambar 3.12 :



Gambar 3. 12 Hasil perhitungan heuristic Manhattan Distance

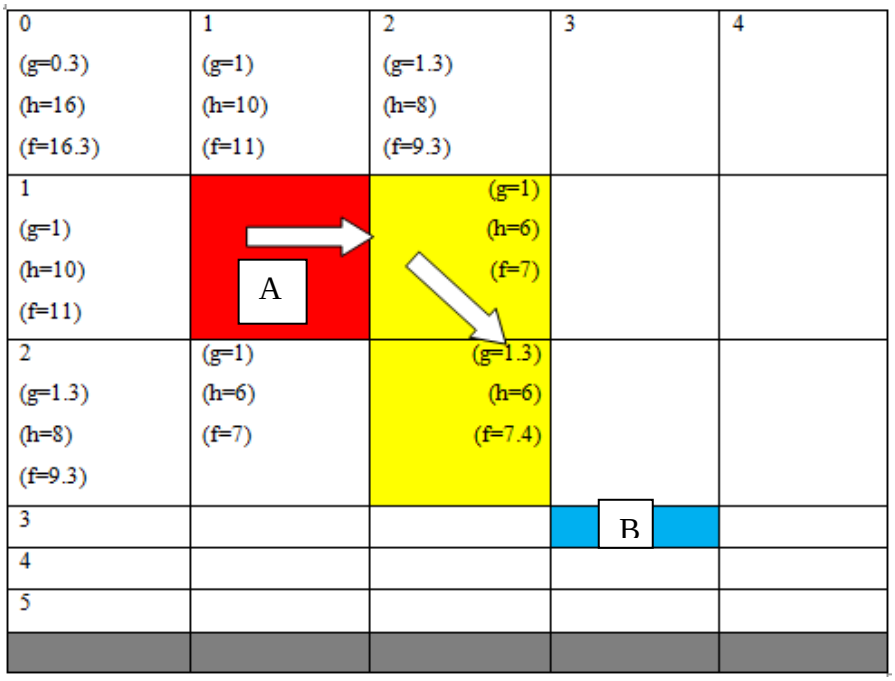
Keterangan :

h : didapat dari rumus manhattan distance

g : didefinisikan pergerakan horizontal dan vertikal= 1, jika diagonal = 1,3

Sebagaimana terlihat pada gambar diatas, simpul yang akan dijadikan sebagai simpul aktif (current node) selanjutnya adalah simpul yang memiliki nilai f paling rendah. Namun karena ada lebih dari satu simpul dengan nilai f sama kecil, maka simpul yang dijadikan simpul aktif adalah simpul dengan nilai f kecil paling pertama yang diambil dari OpenList yaitu simpul (1,2). Selanjutnya simpul tetangga dari simpul (1,2) akan dievaluasi dan dimasukkan ke OpenList. Setelah proses ini selesai, simpul aktif (1,2) dimasukkan ke ClosedList. Lebih lanjut, karena memiliki nilai f kecil dan berada di posisi paling pertama OpenList, simpul (2,2) dijadikan simpul aktif selanjutnya. Simpul tetangga dari simpul (2,2) pun kemudian dievaluasi dan dimasukkan ke OpenList.

Proses ini kemudian terus berulang hingga simpul aktif (current node) sama dengan titik tujuan. Berikut ini gambar hasil akhir pencarian langkah algoritm A Star bisa dilihat pada gambar 3.13 :



Gambar 3. 13 hasil akhir pencarian langkah algoritm A*

3.1.7 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional dilakukan untuk menghasilkan spesifikasi kebutuhan non-fungsional. Spesifikasi kebutuhan non-fungsional adalah spesifikasi yang rinci tentang hal-hal yang akan dilakukan sistem ketika diimplementasikan. Analisis ini diperlukan untuk menentukan keluaran yang akan dihasilkan sistem dan masukan yang diperlukan sistem.

3.1.7.1 Analisis dan Kebutuhan Perangkat Lunak

Analisis perangkat lunak yang digunakan dalam pembangunan aplikasi *game* ini adalah :

1. Sistem Operasi Windows7
2. *Adobe flash CS* 6.0

3.1.7.2 Analisis dan Kebutuhan Perangkat Keras

Perangkat keras atau hardware merupakan salah satu hal yang penting karena tanpa hardware yang memenuhi syarat, program yang akan dibuat tidak akan dapat berjalan. Agar aplikasi dapat berjalan dengan baik, maka dibutuhkan perangkat keras yang sesuai dengan kebutuhan aplikasi. Analisis perangkat

keras yang dibutuhkan dalam pembangunan *game* Petualangan Gatotkaca adalah sebagai berikut :

1. CPU dengan kecepatan minimal 2.0Ghz
2. Ram 1 GB
3. VGA 512 MB
4. Hardisk 20 GB
5. Monitor standar 1024 x 768
6. Mouse, keyboard, dan speaker

Adapun analisis perangkat keras yang dibutuhkan untuk dapat menjalankan Aplikasi *game* Petualangan Gatotkaca adalah :

1. HP berbasis android
2. OS minimal *Ice Cream Sandwitch* 4.0
3. CPU ARM Cortex-A7 Processor dualcore 1.0Ghz
4. Ram 512 MB
5. Screen Resolution minimal 480 x 854

3.1.7.3 Analisis Pengguna

Analisis pengguna untuk dapat memainkan *game* Petualangan Gatotkaca dapat dilihat pada tabel dibawah ini :

Tabel 3. 1 Karakteristik Pengguna

Karakteristik fisik pengguna	
Umur	13 tahun keatas
Gender	Laki-laki dan perempuan
Handedness	Tangan kanan dan kiri
Disabilities	Tidak buta warna total
Pengetahuan dan pengalaman pengguna	
Computer Literacy	Memiliki pengalaman mengoperasikan sebuah *smartphone* android
System Experience	Pernah memainkan aplikasi *game mobile*
Aplication Experience	Pernah memainkan *game* sejenis
Education	Sekolah dasar
Task Experience	Mengetahui cara menggunakan android

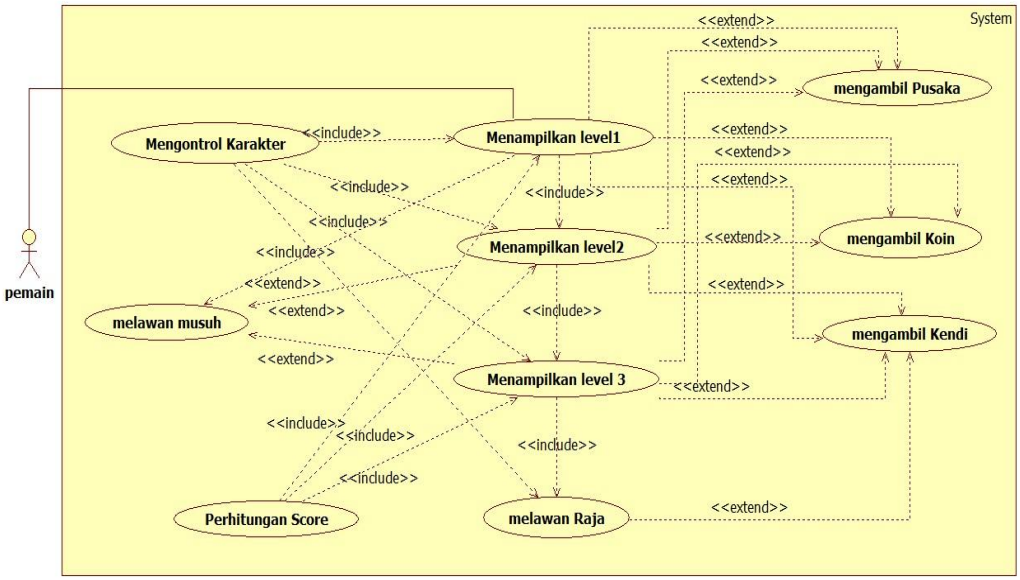
Native language or culture	Bahasa Indonesia
Karakteristik Psikologi pengguna	
Motivation	Mempunyai motivasi yang cukup dalam menyelesaikan permainan
Patience	Mempunyai kesabran dalam menyelesaikan permainan
Stress level	Tidak mudah *stress*
Attitude	*Neutral*

3.1.8 Analisis dan Kebutuhan Fungsional

Kebutuhan fungsional disebut juga kebutuhan operasional, yaitu kebutuhan yang berkaitan dengan fungsi atau proses transformasi yang harus mampu dikerjakan oleh perangkat lunak. Kebutuhan fungsional yang digunakan adalah diagram *usecase*, *activity*, *sequence*, dan *class*.

3.1.8.1 Use Case Diagram

Use Case merupakan gambaran interaksi antara pengguna atau pemain dengan aplikasi *game* Petualangan Gatotkaca dan mendeskripsikan hubungan yang terjadi antara *actor* dengan aktivitas pada sistem yang berjalan. Berikut adalah use case diagram pada *game* petualangan gatotkaca bisa dilihat pada gambar 3.14 :



Gambar 3. 14 *Use Case Diagram Game* Petualangan Gatotkaca

3.1.8.2 Definisi Use Case

Definisi *Use case* yaitu menjelaskan fungsi *use case* yang terdapat pada *use case* diagram. Berikut definisi *Use case* di tunjukan pada tabel 3.2 :

Tabel 3. 2 Definisi *Use Case*

No	*Use case*	Deskripsi
1	Menampilkan *Level* 1	Proses untuk menampilkan permainan *level* 1
2	Menampilkan *Level* 2	Proses untuk menampilkan permainan *level* 2
3	Menampilkan *Level* 3	Proses untuk menampilkan permainan *level* 3
4	Kontrol karakter	Proses untuk mengendalikan karakter Gatotkaca
5	Melawan Musuh	Proses untuk melawan Musuh
6	Mengambil Kendi	Proses untuk mengambil item Kendi
7	Mengambil pusaka	Proses untuk mengambil item Pusaka
8	Mengambil Koin	Proses untuk mengambil item Koin
9	Melawan Raja	Proses Pertarungan antara karakter Gatotkaca dengan Musuh Raja adipatikarna
10	Perhitungan Score	Proses untuk menghitung score

3.1.8.3 Skenario Use case

Scenario use Case menjelaskan urutan langkah-langkah proses apa yang akan terjadi pada setiap bagian yang terdapat pada *use case* diagram, berikut adalah *scenario* masing-masing *use case* yang telah didefinisikan sebelumnya:

1. **Use Case Skenario menampilkan Level 1**

Skenario use case ini menjelaskan interaksi antara aktor yaitu pemain dengan use case menampilkan level 1 yang akan dijelaskan pada tabel 3.3 :

Tabel 3. 3 *Use Case* Skenario Menampilkan *level* 1

Identifikasi	
Nomor	1
Nama	Menampilkan *Level* 1
Tujuan	Menampilkan *Level* 1
Deskripsi	Mengontrol proses yang dilakukan pemain untuk menampilkan permainan *level1* dan memulai permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di menu utama
Aksi Aktor	**Reaksi Sistem**
1. Pemain memilih menu mulai permainan	
	2. Sistem akan merespon dengan menampilkan intro permainan
3. Pemain menekan tombol skip	
	4. Sistem menampilkan arena permainan *level* 1
5. Pemain mengendalikan karakter	
	6. Karakter mengambil kendi
	7. Karakter melawan Musuh

	8. Mengambil Koin
	9. Mengambil Pusaka
	10. Jika pemain kalah maka pemain mengulang permainan
	11. Jika pemain menang maka melanjutkan permainan ke level selanjutnya
Kondisi Akhir	Karakter harus mendekati portal untuk melanjutkan ke *level* selanjutnya.

2. Use Case Skenario menampilkan Level 2

Skenario use case ini menjelaskan interaksi antara aktor yaitu pemain dengan use case menampilkan level 2 yang akan dijelaskan pada tabel 3.4 :

Tabel 3. 4 *Use Case* Skenario Menampilkan *Level* 2

Identifikasi	
Nomor	2
Nama	Menampilkan *Level* 2
Tujuan	Menampilkan *level* 2
Deskripsi	Proses untuk melanjutkan atau menampilkan *level* 2
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Pemain berada di arena Permainan.
Aksi Aktor	**Reaksi Sistem**
1. Mulai Permainan level 2	
	2. Sistem menampilkan arena permainan *level* 2
3. Mengendalikan karakter pemain	
	4. Pemain mengambil kendi
	5. Pemain melawan musuh

	6. Pemain mengambil pusaka
	7. Pemain mengambil koin
	8. Jika pemain kalah maka permainan diulang
	9. Jika pemain menang melanjutkan ke level selanjutnya
Kondisi terakhir	Karakter harus mendekati portal untuk melanjutkan ke *level* selanjutnya.

3. Use Case Skenario menampilkan Level 3

Skenario *use case* ini menjelaskan interaksi antara aktor yaitu pemain dengan use case menampilkan level 3 yang akan dijelaskan pada tabel 3.5 :

Tabel 3. 5 *Use Case* Skenario Menampilkan *Level* 3

Identifikasi	
Nomor	3
Nama	Menampilkan Level 3
Tujuan	Menampilkan *Level* 3
Deskripsi	Proses untuk menampilkan permainan *level* 3
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Berada di arena permainan
Aksi Aktor	**Reaksi Sistem**
1. Mulai permainan level 3	
	2. Sistem menampilkan permainan *level* 3
3. Mengendalikan karakter pemain	
	4. Pemain mengambil kendi
	5. Pemain melawan musuh
	6. Pemain mengambil pusaka
	7. Pemain mengambil koin
	8. Jika pemain kalah maka permainan

	diulang
	9. Jika pemain menang maka pemain melanjutkan level selanjutnya yaitu Melawan raja
Kondisi terakhir	Karakter harus mendekati portal untuk melanjutkan ke *level* selanjutnya.

4. Use Case Skenario Mengontrol karakter

Skenario use case ini menjelaskan interaksi antara aktor yaitu pemain dengan use case Mengontrol Karakter yang akan dijelaskan pada tabel 3.6 :

Tabel 3. 6 *Use Case* Skenario Mengontrol Karakter

Identifikasi	
Nomor	4
Nama	Mengontrol Karakter
Tujuan	Menggerakan karakter Gatotkaca
Deskripsi	Proses untuk menggerakan karakter
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Berada di Memulai Permainan.
Aksi Aktor	**Reaksi Sistem**
1. Pemain menekan tombol arah kanan 	
	2. Sistem akan merespon dan karakter akan berjalan maju
3. pemain menekan tombol arah kiri 	

	4. Sistem akan merespon dan karakter akan berjalan berbalik
5. Pemain menekan tombol lompat 	
	6. Sistem akan merespon dan karakter akan melompat
7. Pemain menekan tombol pukul 	
	8. Sistem akan merespon dan karakter akan memukul.
Kondisi Akhir	Karakter akan bergerak.

5. Use Case Skenario Melawan Musuh

Skenario use case ini menjelaskan interaksi antara aktor yaitu pemain dengan use case Melawan Musuh yang akan dijelaskan pada tabel 3.7 :

Tabel 3. 7 *Use Case* Skenario Melawan Musuh

Identifikasi	
Nomor	5
Nama	Melawan musuh
Tujuan	Mengalahkan musuh
Deskripsi	Proses untuk melawan musuh
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Karakter bergerak
Aksi Aktor	**Reaksi Sistem**
1. Pemain menggerakan	

karakter	
	2. Sistem akan merespon dan karakter bergerak mendekati musuh
3. Pemain memukul musuh	
	4. Sistem akan merespon dan musuh menghilang
	5. Jika karakter terkena pukulan darah berkurang
Kondisi Akhir	Mengalahkan musuh.

6. Use Case Skenario Mengambil Kendi

Skenario use case ini menjelaskan interaksi antara aktor yaitu pemain dengan use case Mengambil Kendi yang akan dijelaskan pada tabel 3.8 :

Tabel 3. 8 *Use Case* Skenario Mengambil Kendi

Identifikasi	
Nomor	6
Nama	Mengambil Kendi
Tujuan	Mengambil Kendi
Deskripsi	Proses untuk mengambil Kendi
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Karakter bergerak
Aksi Aktor	**Reaksi Sistem**
1. Pemain bergerak mendekati Kendi	
	2. Sistem mengecek collision antara pemain dengan kendi
	3. Jika kena, darah pemain bertambah
	4. Jika tidak kena, darah pemain tidak bertambah
Kondisi Akhir	darah bertambah pada saat mengambil Kendi.

7. Use Case SKenario Mengambil Pusaka

Skenario use case ini menjelaskan interaksi antara aktor yaitu pemain dengan use case Mengambil Pusaka yang akan dijelaskan pada tabel 3.9 :

Tabel 3. 9 *Use Case* Skenario Mengambil Pusaka

Identifikasi	
Nomor	7
Nama	Mengambil Pusaka
Tujuan	Mengambil Pusaka
Deskripsi	Proses untuk mengambil Pusaka
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Karakter bergerak
Aksi Aktor	**Reaksi Sistem**
1. Pemain bergerak mendekati Pusaka	
	2. Sistem mengecek collision antara karakter dengan pusaka
	3. Sistem memeriksa jumlah item pusaka
	4. Jika Pusaka sudah terkumpul maka portal terbuka
	5. Jika pusaka belum terkumpul portal tidak terbuka
Kondisi Akhir	Sistem membuka Portal level berikutnya

8. Use Case Skenario Mengambil Koin

Skenario use case ini menjelaskan interaksi antara aktor yaitu pemain dengan use case Mengambil Koin yang akan dijelaskan pada tabel 3.10 :

Tabel 3. 10 Use *Case* Skenario Mengambil Koin

Identifikasi	
Nomor	8
Nama	Mengambil Koin
Tujuan	Mengambil Koin

Deskripsi	Proses untuk mengambil Koin
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Karakter bergerak
Aksi Aktor	**Reaksi Sistem**
1. Pemain bergerak mendekati koin	
	2. Sistem mengecek collision karakter dengan koin
	3. Jika koin terambil score bertambah
	4. Jika koin tidak terambil score tidak bertambah
Kondisi Akhir	score bertambah pada saat mengambil Koin.

9. Use Case Skenario Melawan Raja

Skenario use case ini menjelaskan interaksi antara aktor yaitu pemain dengan use case Melawan Raja yang akan dijelaskan pada tabel 3.11 :

Tabel 3. 11 *Use Case* Skenario Melawan Raja

Identifikasi	
Nomor	9
Nama	Melawan Raja
Tujuan	Mengalahkan Raja
Deskripsi	Proses untuk melawan Raja
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Karakter bergerak
Aksi Aktor	**Reaksi Sistem**
1. Pemain berjalan mendekati Raja	
	2. Sistem akan merespon dan Raja bergerak mendekati karakter
	3. Sistem menjalankan algoritma A*

	4. Memasukan node awal ke list open
	5. mengecek node yang kosong
	6. mengecek rute pergerakan pemain
	7. Jika ditemukan raja akan mengikuti pergerakan pemain
8. Pemain memukul Raja	
	9. Jika serangan pemain mengenai raja darah akan berkurang
	10. Jika raja menyentuh pemain darah pemain akan berkurang
	11. Pemain menghindari raja darah tidak akan berkurang
Kondisi Akhir	Mengalahkan Raja

12. Perhitungan Score

Skenario use case ini menjelaskan interaksi antara aktor yaitu pemain dengan use case Perhitungan Score yang akan dijelaskan pada tabel 3.12 :

Tabel 3. 12 *Use Case* Skenario Perhitungan *Score*

Identifikasi	
Nomor	10
Nama	Perhitungan *Score*
Tujuan	Menghitung *score* di tiap level
Deskripsi	Proses untuk menghitung *score* di tiap *level*
Aktor	Pemain
Skenario Utama	
Kondisi Awal	pemain berada di arena permainan.
Aksi Aktor	**Reaksi Sistem**
1. Pemain mengambil item dan melawan musuh	
	2. Cek collision antara karakter dengan koin

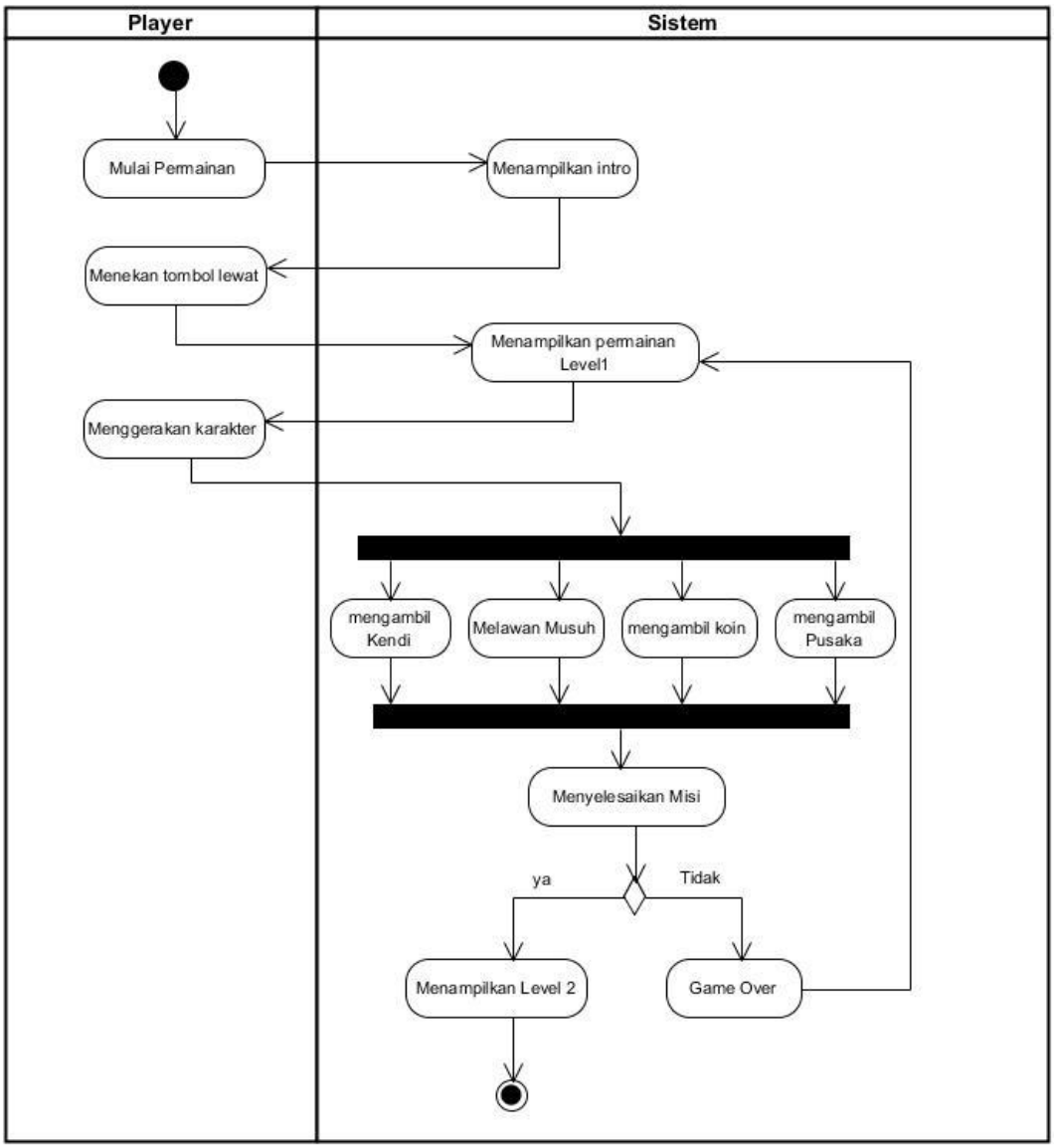
	3. Objek akan menghilang ketika pemain mengambil koin
	4. jika pemain mengambil koin score bertambah 100 point
	5. jika pemain tidak mengambil koin score tidak bertambah
	6. Cek collision antara karakter dengan musuh
	7. Jika pemain mengalahkan musuh *score* bertambah 10 point
	8. Jika pemain tidak melawan musuh *score* tidak akan bertambah
Kondisi Akhir	Score bertambah jika pemain mengambil koin dan melawan musuh

3.1.8.4 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah system atau proses bisnis atau menu yang ada pada perangkat lunak. Pada *activity* diagram ini adalah menggambarkan diagram aktivitas sistem bukan apa yang dilakukan *actor*. Berikut ini *activity* diagram pada aplikasi *game* Petualangan Gatotkaca.

1. Activity Diagram menampilkan Level 1

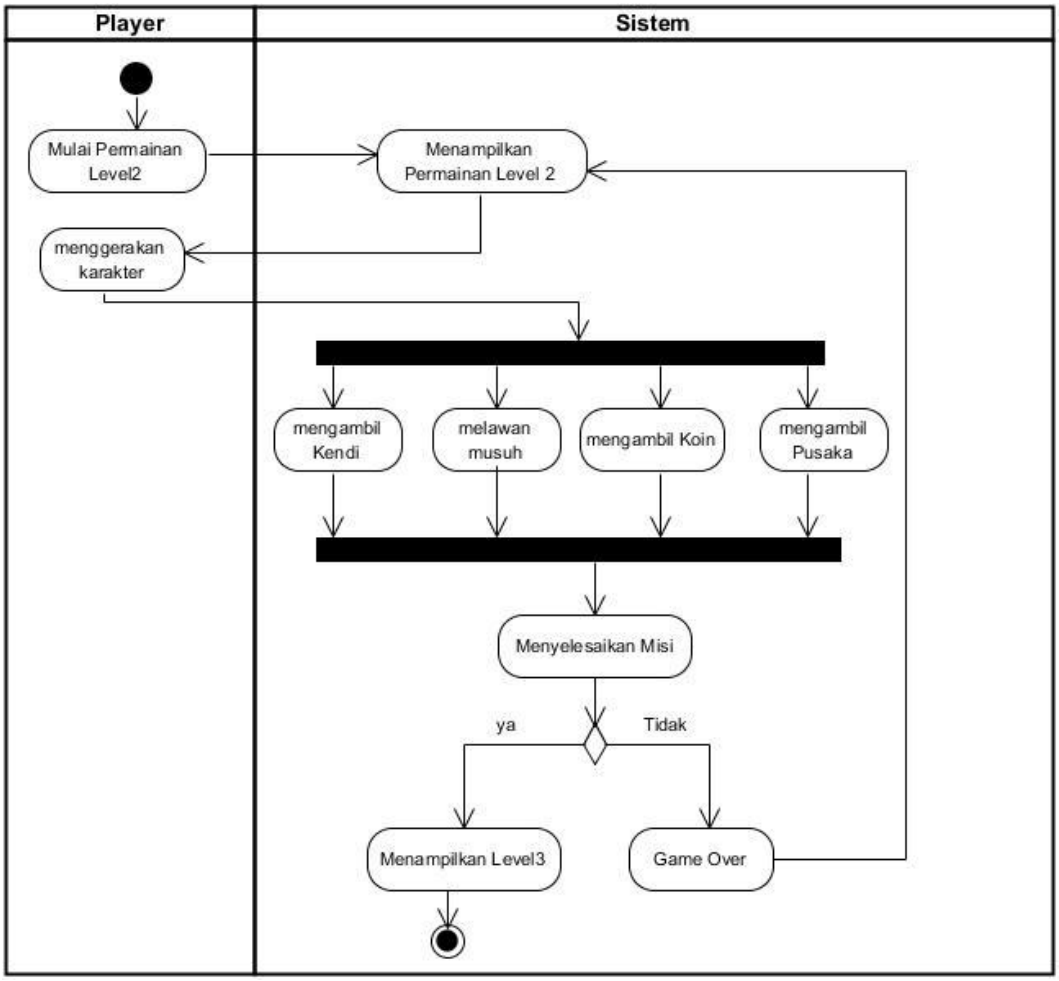
Activity Diagram Menampilkan *level* 1 pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.15 :



Gambar 3. 15 *Activity Diagram* Menampilkan *Level 1*

2. Activity Diagram menampilkan Level 2

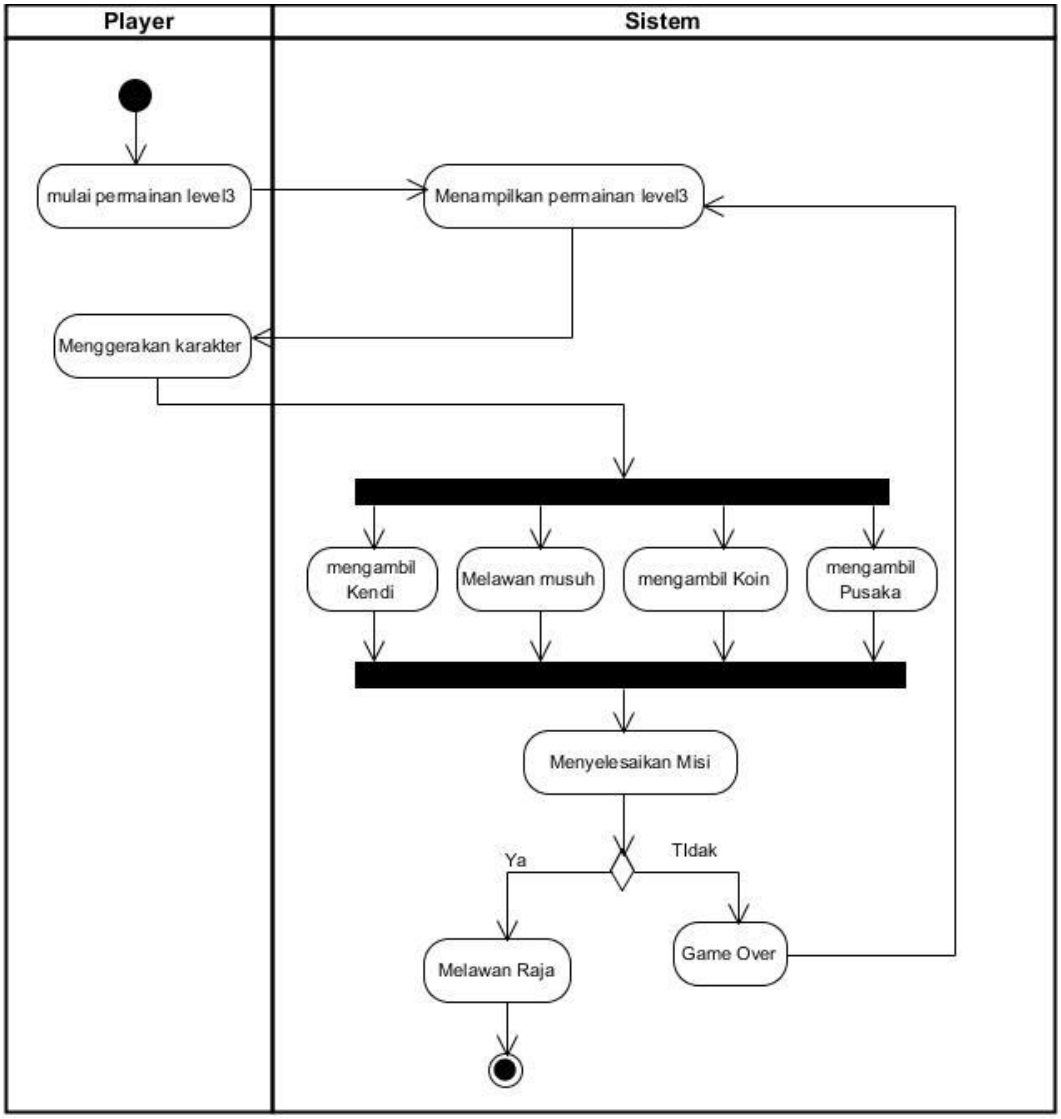
Activity Diagram Menampilkan *level* 2 pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.16:



Gambar 3. 16 *Activity Diagram* Menampilkan *level* 2

3. Activity Diagram menampilkan level 3

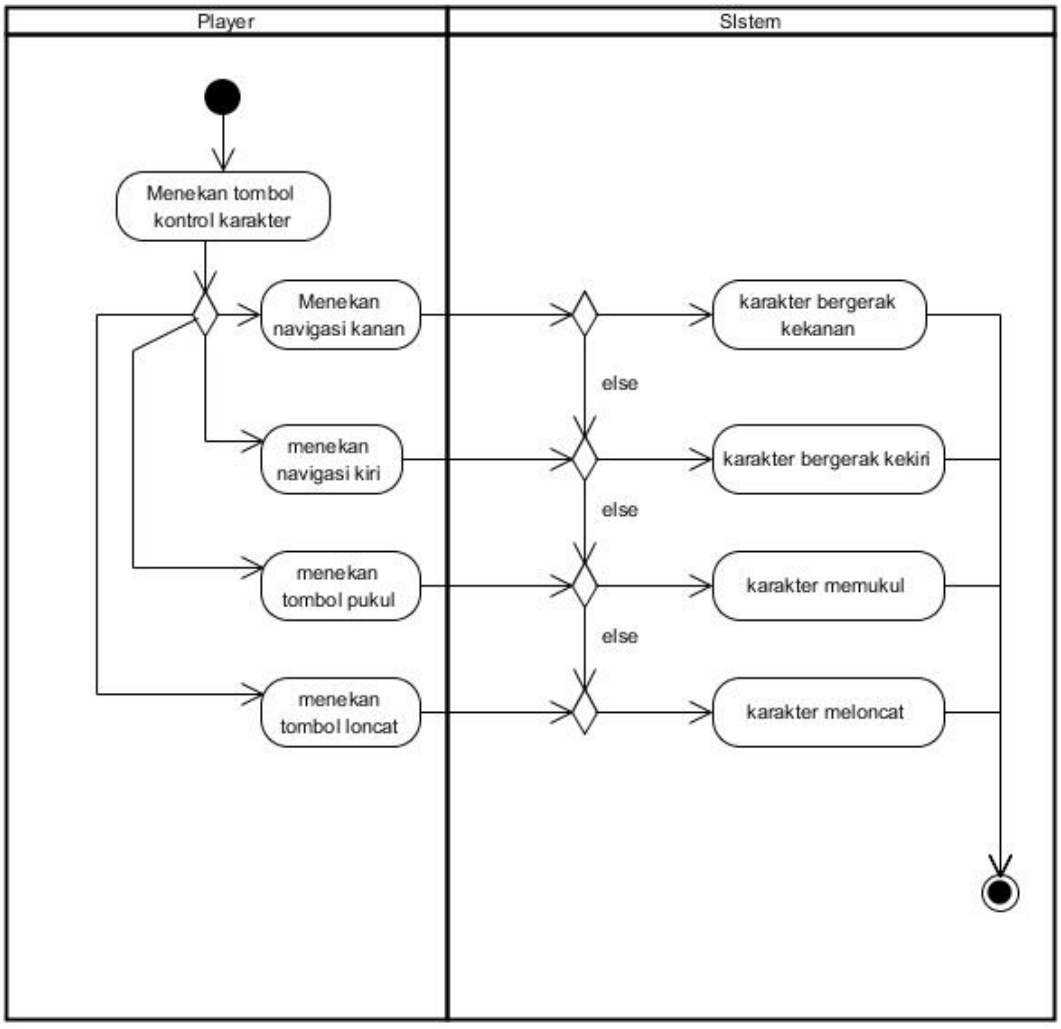
Activity Diagram Menampilkan *level* 3 pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.17 :



Gambar 3. 17 *Activity Diagram* Menampilkan *Level* 3

4. Activity Diagram Kontrol karakter

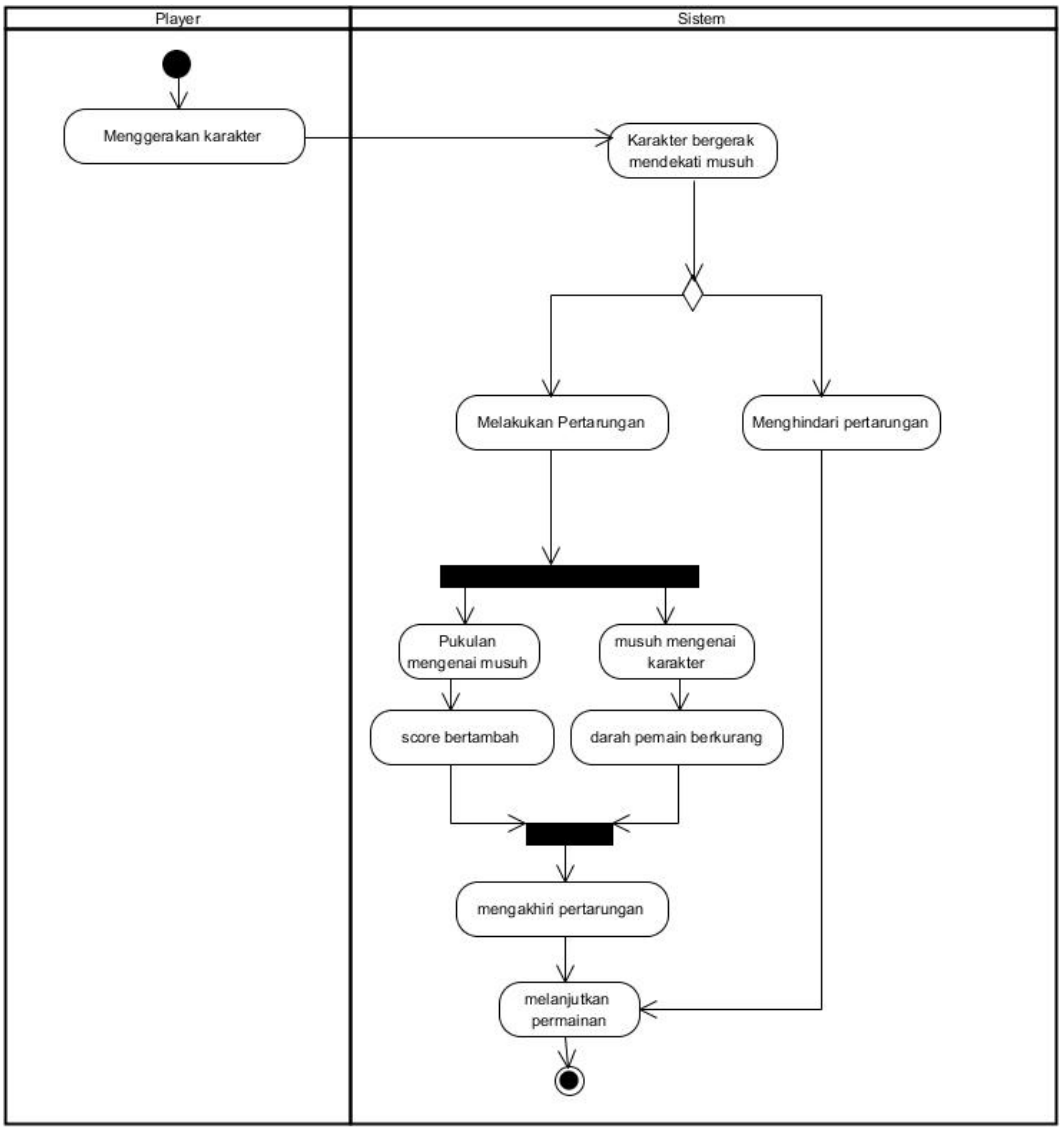
Activity Diagram kontrol karakter pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.18 :



Gambar 3. 18 *Activity Diagram* Kontrol karakter

5. Activity Diagram Melawan Musuh

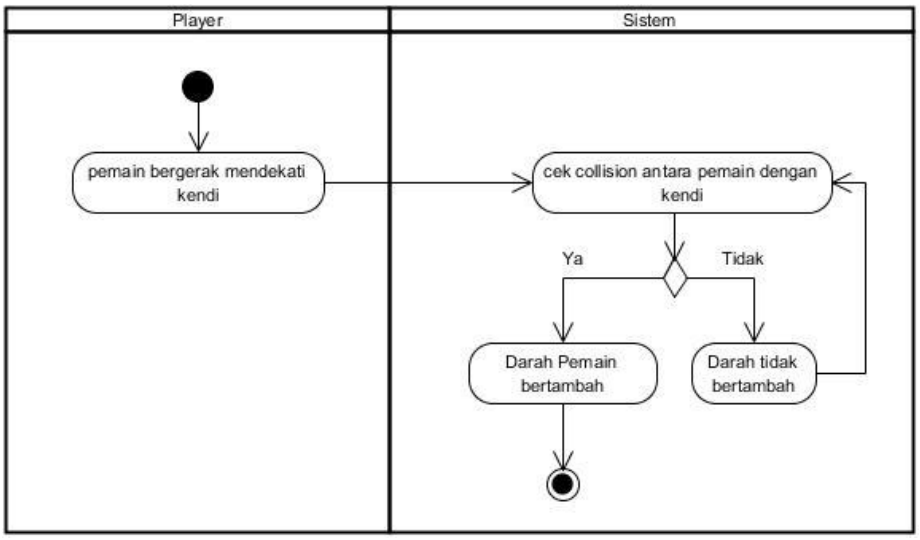
Activity Diagram melawan musuh pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.19 :



Gambar 3. 19 *Activity Diagram* Melawan Musuh

6. Activity Diagram Mengambil Kendi

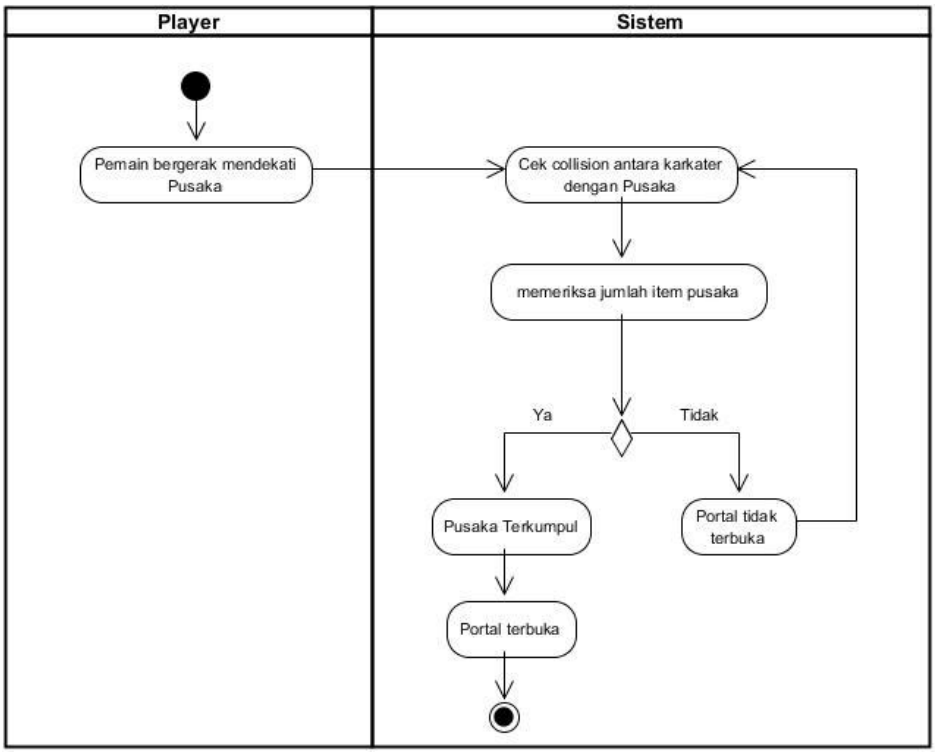
Activity Diagram mengambil Kendi pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.20 :



Gambar 3. 20 *Activity Diagram* Mengambil Kendi

7. Activity Diagram Mengambil Pusaka

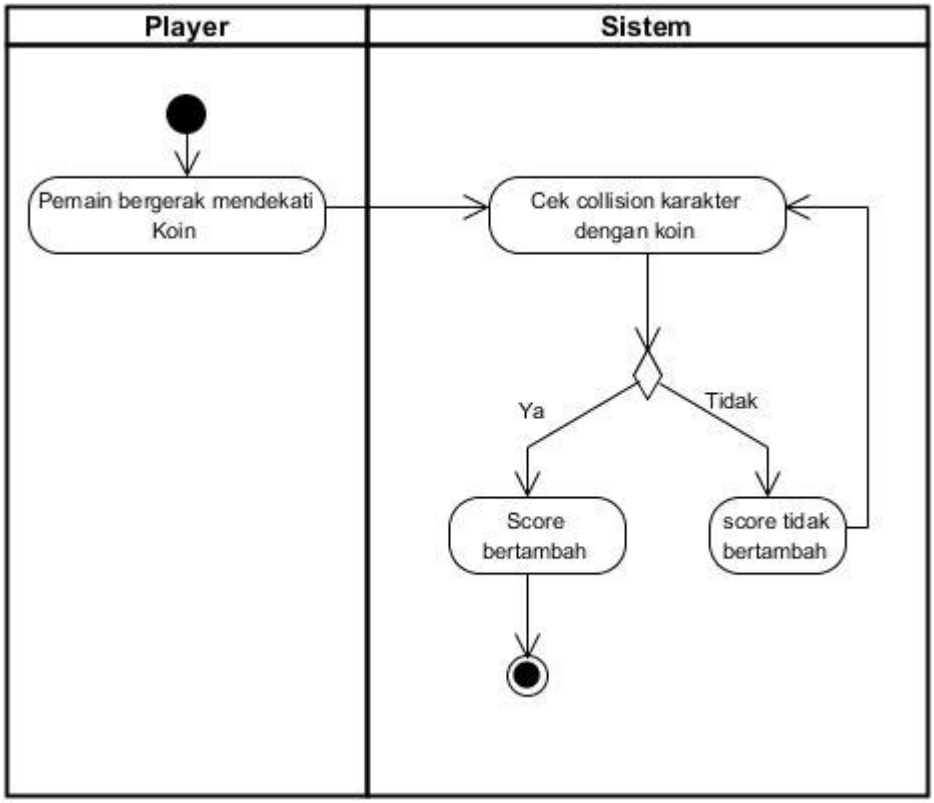
Activity Diagram mengambil Pusaka pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.21 :



Gambar 3. 21 *Activity Diagram* Mengambil Pusaka

8. Activity Diagram Mengambil Koin

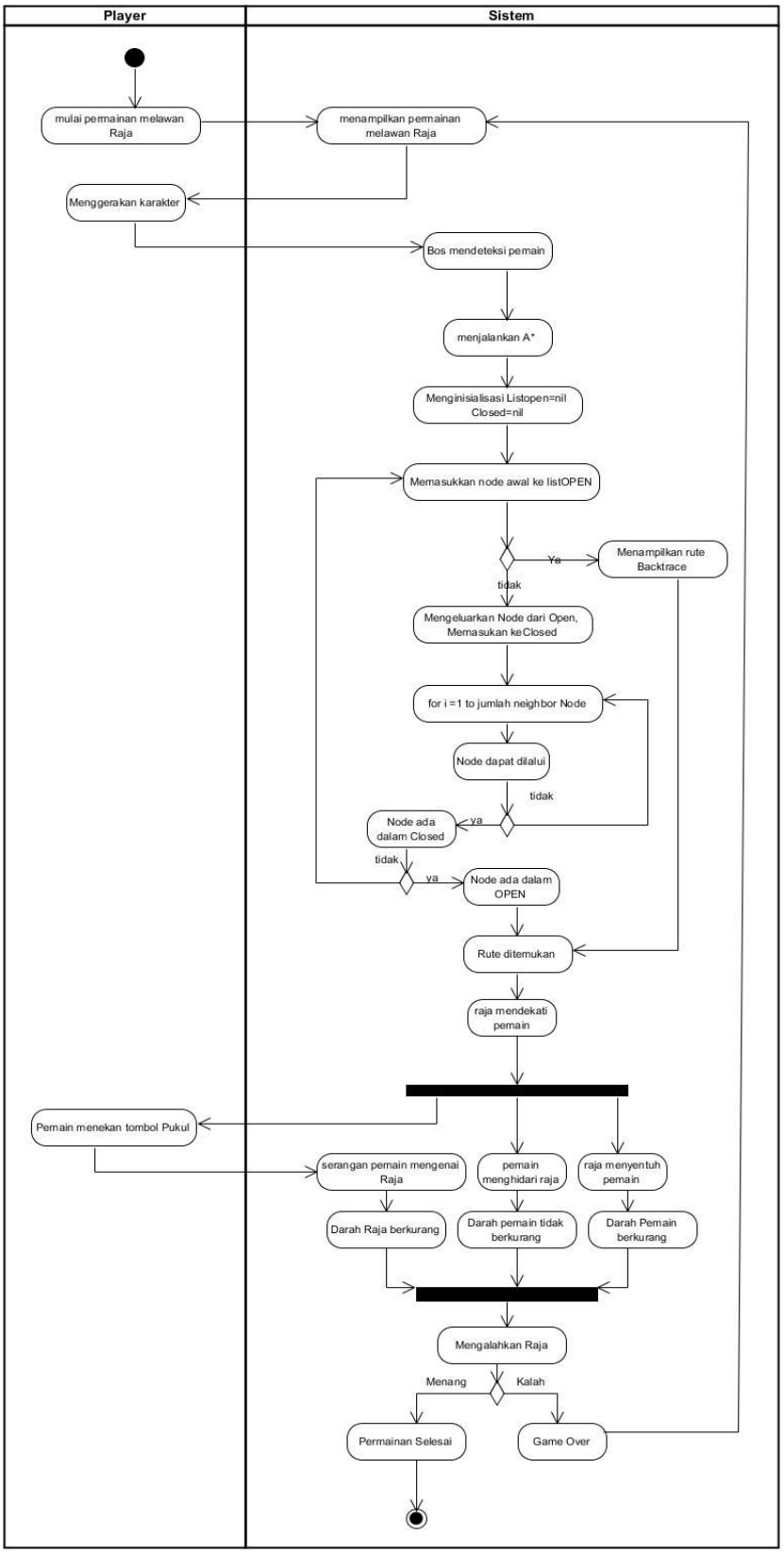
Activity Diagram Mengambil Koin pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.22 :



Gambar 3. 22 *Activity Diagram* Mengambil Koin

9. Activity Diagram Melawan Raja

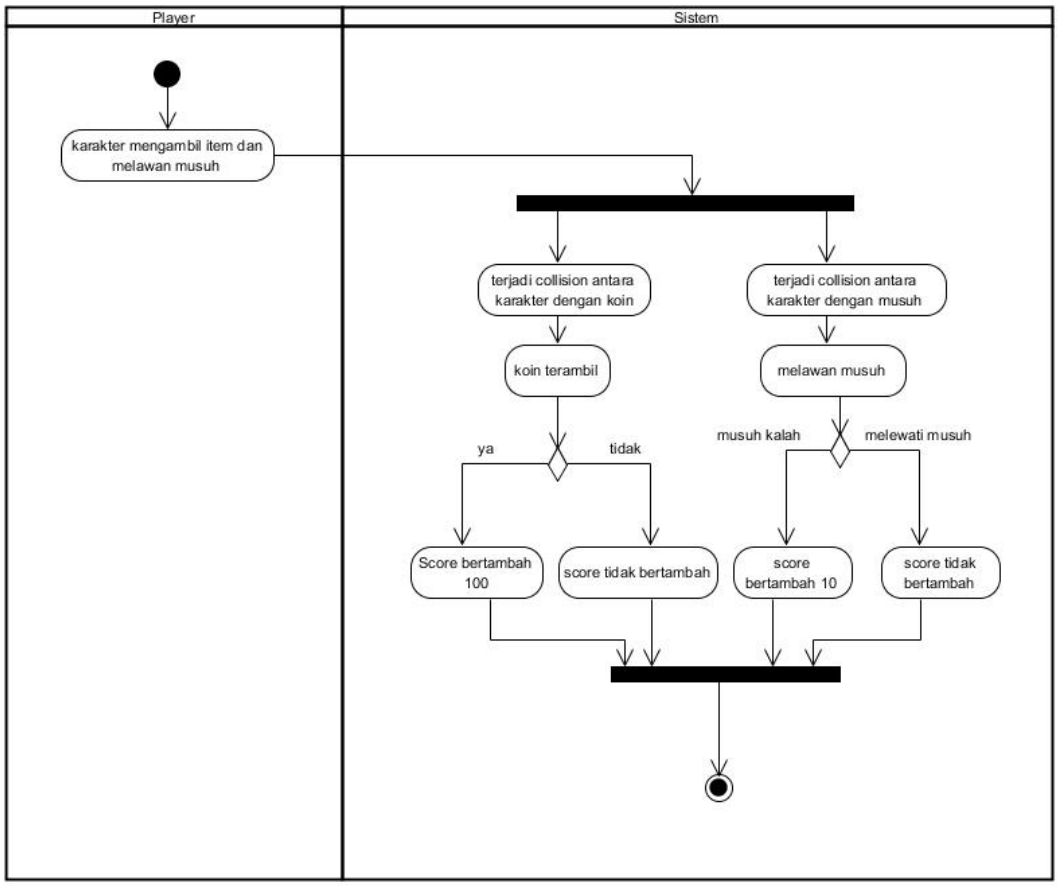
Activity Diagram Melawan Raja pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.23 :



Gambar 3. 23 *Activity Diagram* Melawan Raja

10. Activity Diagram Perhitungan Score

Activity Diagram Perhitungan score pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.24:



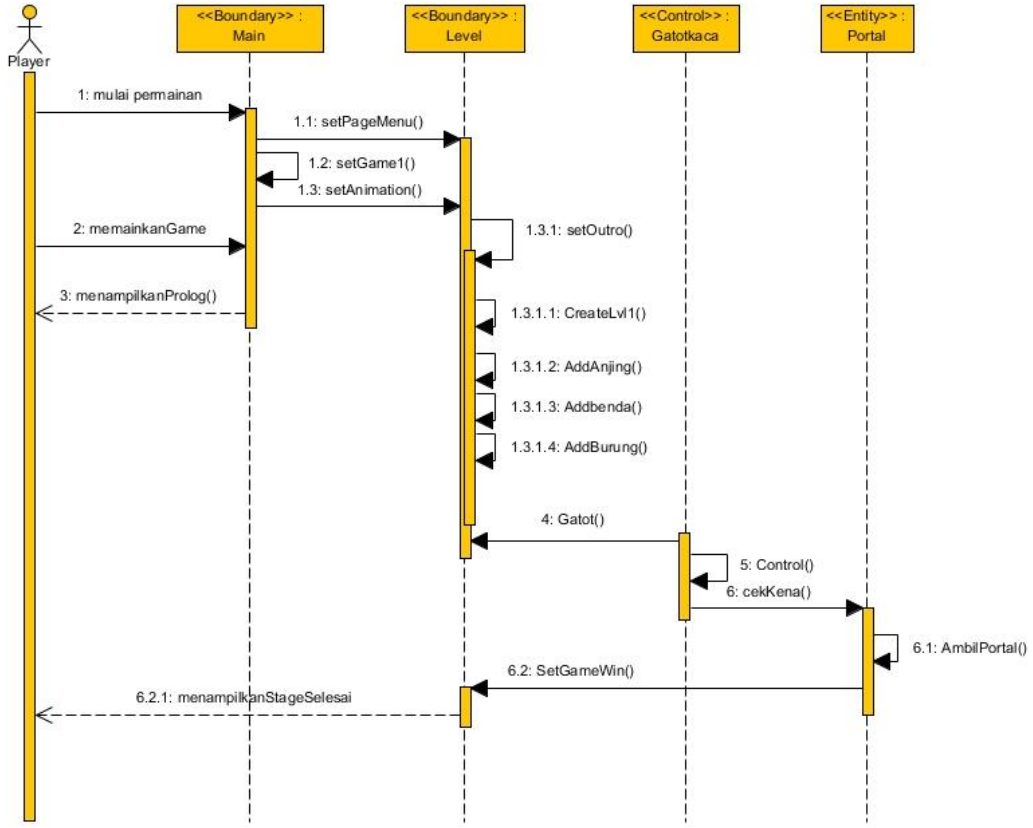
Gambar 3. 24 *Activity Diagram* Perhitungan Score

3.1.8.5 Sequence Diagram

Sequence diagram yaitu menggambarkan interaksi antar masing-masing objek pada setiap *use case*. *Sequence diagram* pada *game* Petualangan Gatotkaca terdiri dari *sequence diagram* menampilkan *sequence diagram* mulai permainan, *sequence diagram* tentang pembuat, *sequence diagram* cara bermain, dan *sequence diagram* keluar. Penjelasan *sequence diagram* Petualangan Gatotkaca dapat dilihat sebagai berikut :

1. Sequence Diagram menampilkan level 1

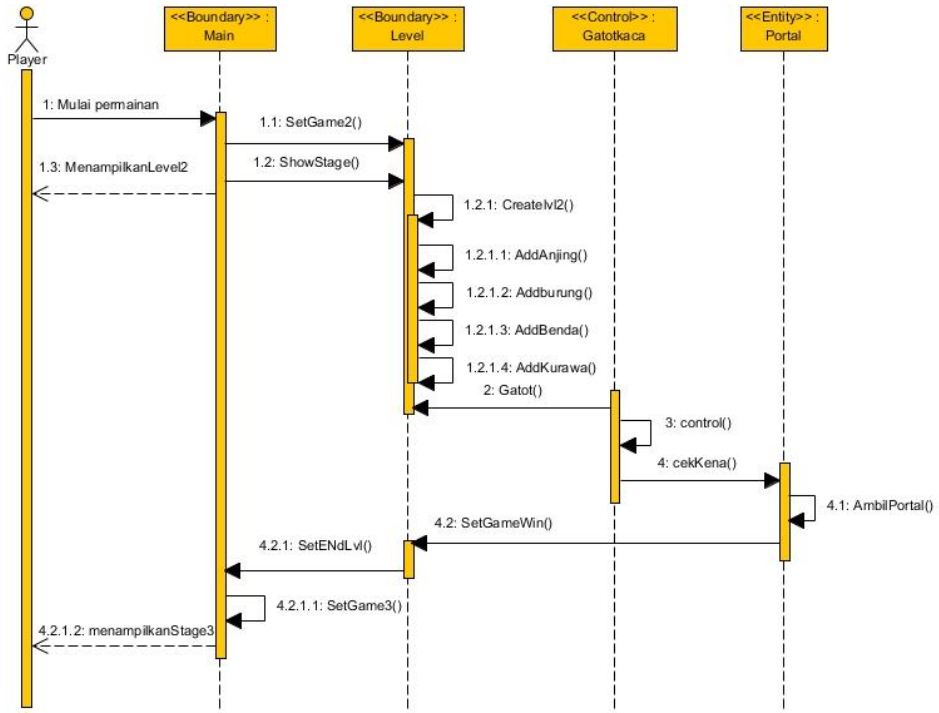
Sequence Diagram menampilkan *level 1* pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.25:



Gambar 3. 25 *Sequence Diagram* Menampilkan *Level* 1

2. Sequence Diagram Menampilkan Level 2

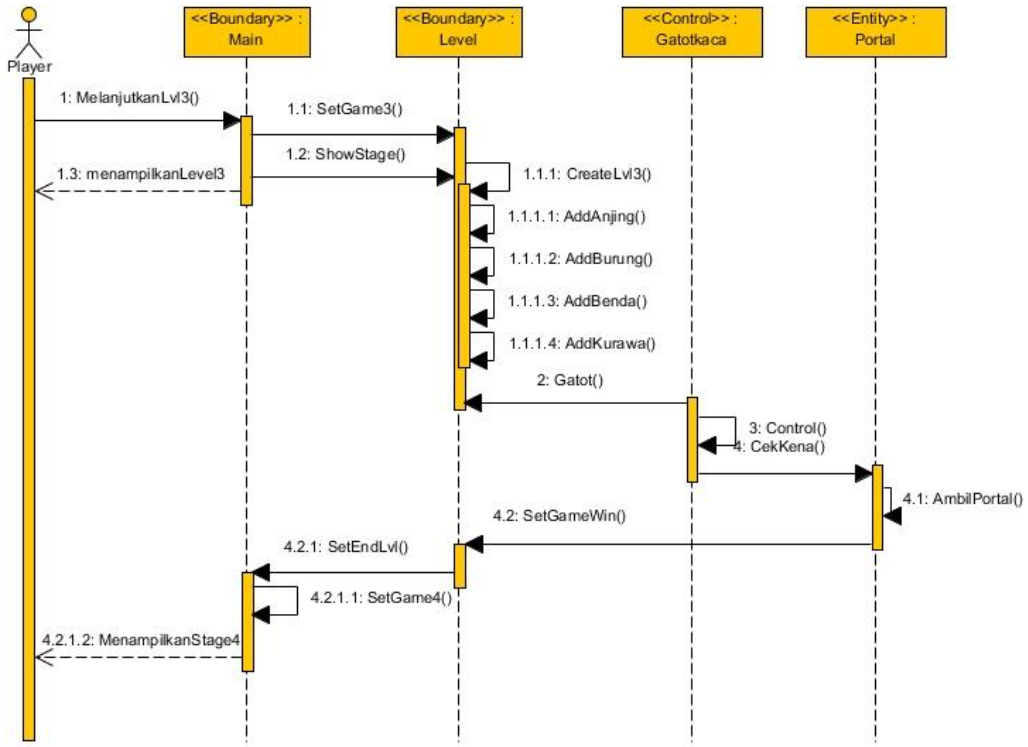
Sequence Diagram Menampilkan *Level* 2 pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.26:



Gambar 3. 26 *Sequence Diagram* Menampilkan *level* 2

3. Sequence Diagram Menampilkan Level 3

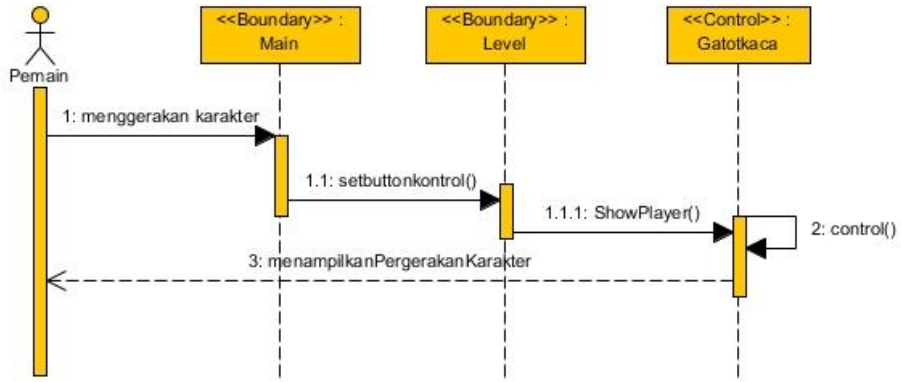
Sequence Diagram Menampilkan *level3* pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.27 :



Gambar 3. 27 *Sequence Diagram* Menampilkan *Level* 3

4. Sequence Diagram Kontrol Karakter

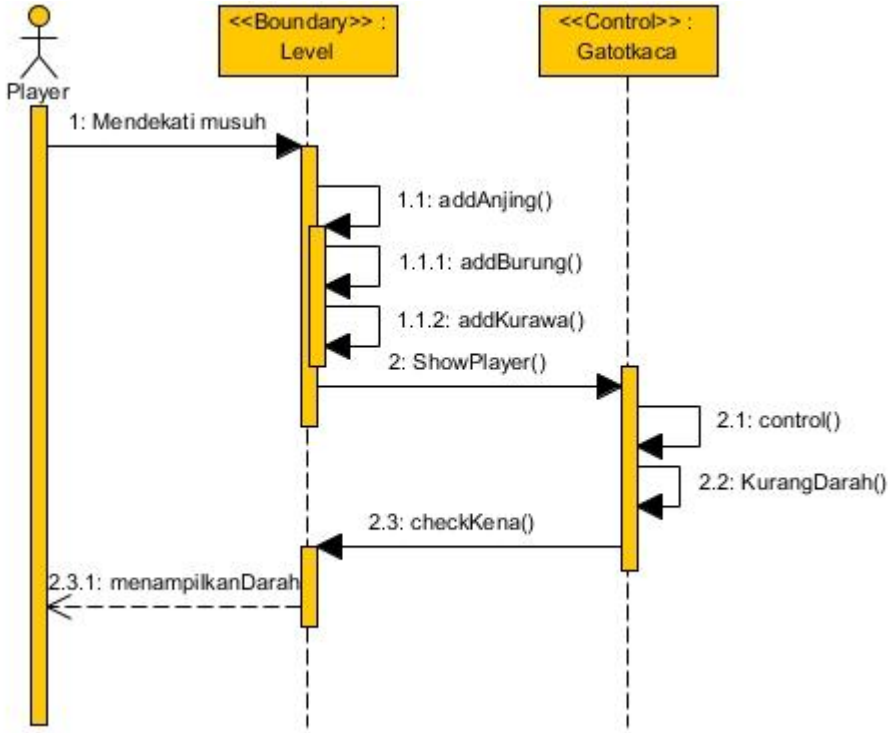
Sequence Diagram kontrol karakter pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.28 :



Gambar 3. 28 *Sequence Diagram* Kontrol Karakter

5. Sequence Diagram Melawan Musuh

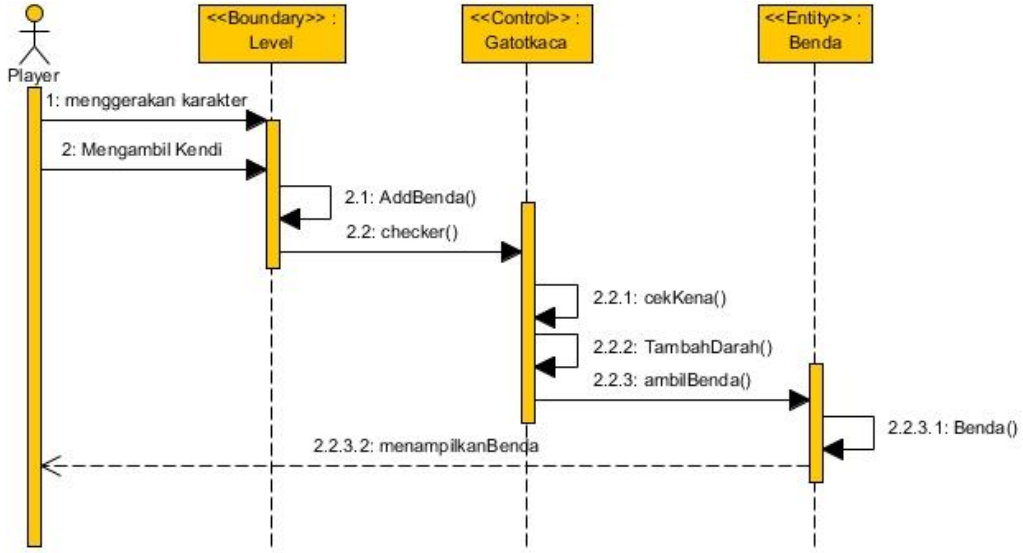
Sequence Diagram Melawan Musuh pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.29 :



Gambar 3. 29 *Sequence Diagram* Melawan Musuh

6. Sequence Diagram Mengambil Kendi

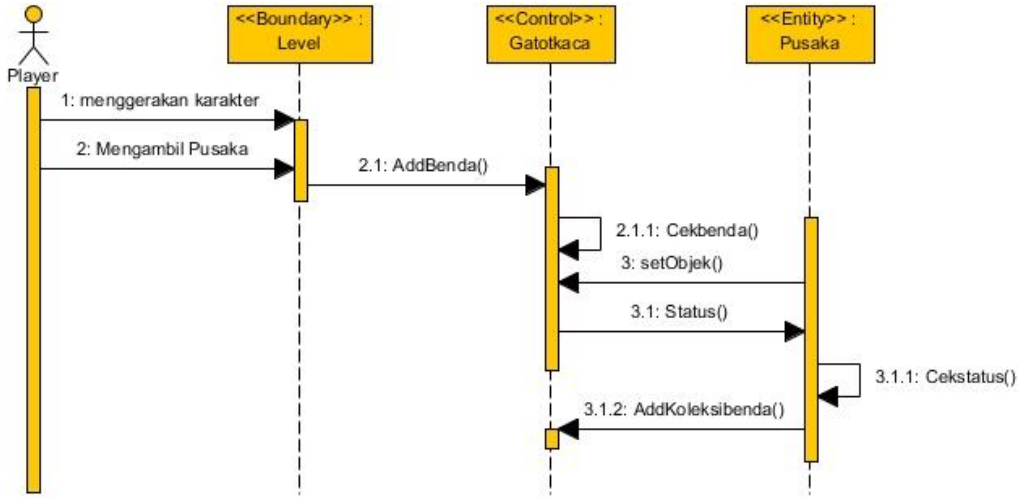
Sequence Diagram Mengambil Kendi pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.30:



Gambar 3. 30 *Sequence Diagram* Mengambil Kendi

7. Sequence Diagram Mengambil Pusaka

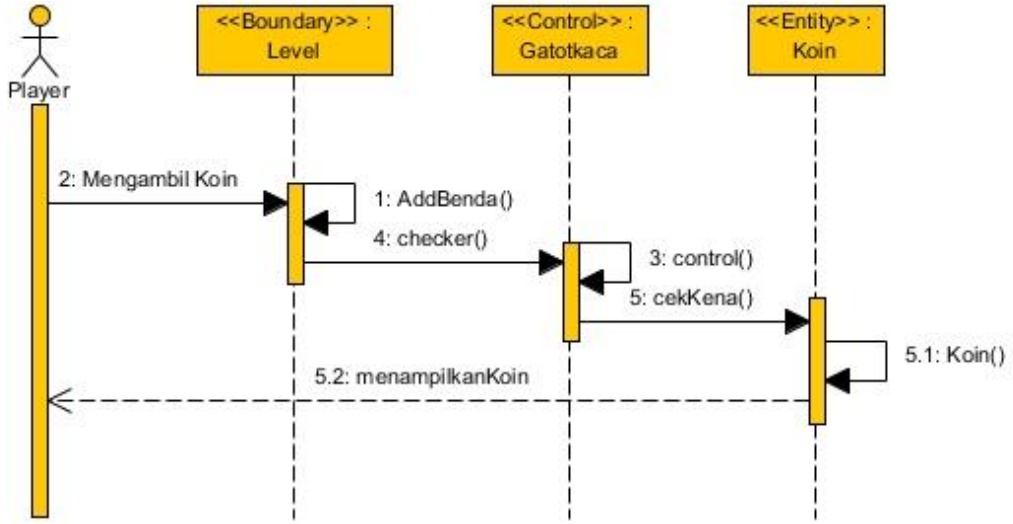
Sequence Diagram Mengambil Pusaka pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.31:



Gambar 3. 31 *Sequence Diagram* Mengambil Pusaka

8. Sequence Diagram Mengambil Koin

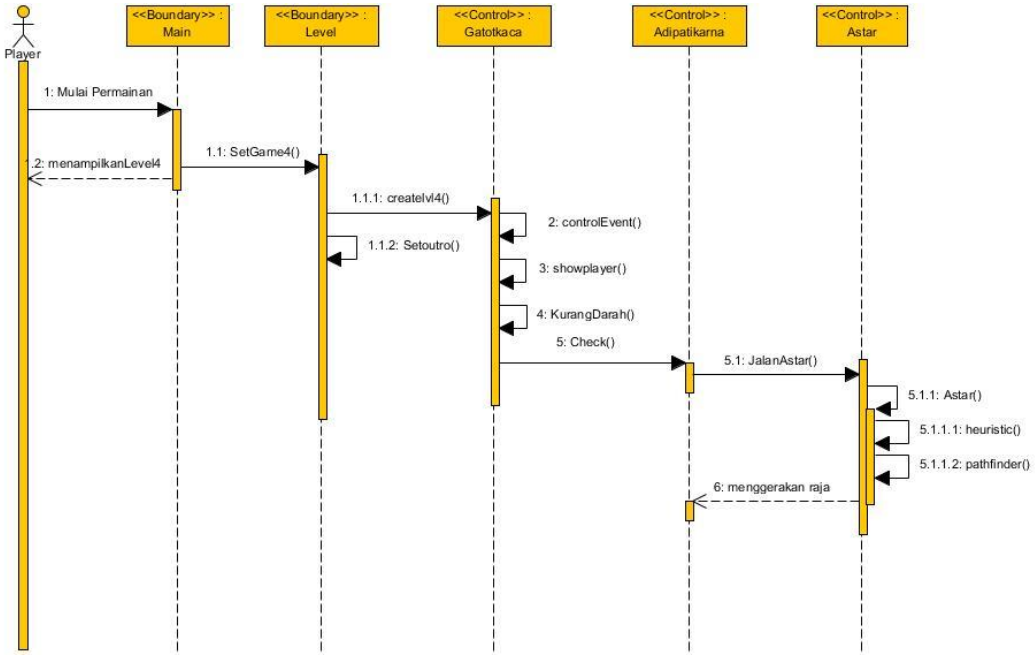
Sequence Diagram Mengambil Koin pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.32:



Gambar 3. 32 *Sequence Diagram* Mengambil Koin

9. Sequence Diagram Melawan Raja

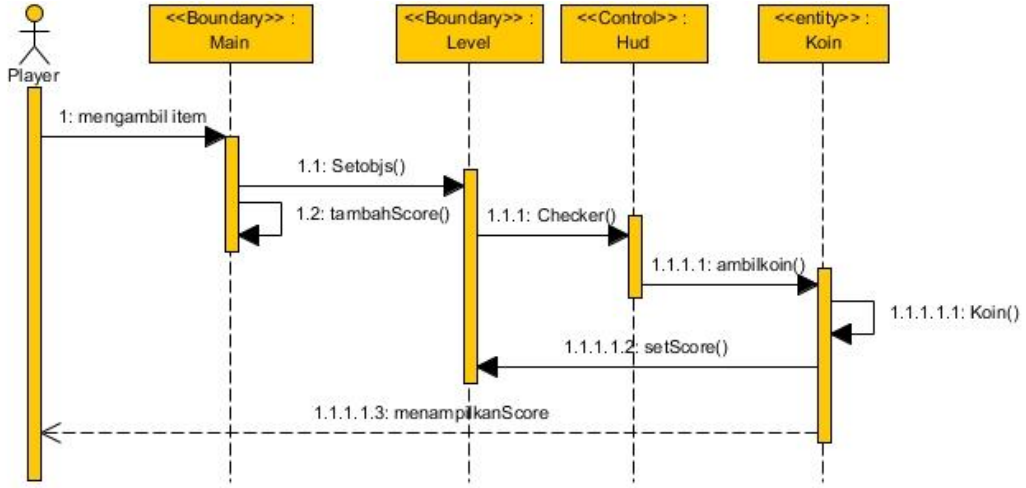
Sequence Diagram Melawan Raja pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.33:



Gambar 3. 33 *Sequence Diagram* Melawan Raja

10. Sequence Diagram Perhitungan Score

Sequence Diagram perhitungan *score* pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.34:



Gambar 3. 34 *Sequence Diagram* Perhitungan Score

3.1.8.6 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas yang dibuat pada *game* Petualangan Gatotkaca, struktur yang ada meliputi *atribut-atribut* dan *method-method* yang ada pada masing-masing *class*.

1. Perancangan Class

Perancangan class menjelaskan class – class yang akan dibuat. Terdapat 3 jenis class yaitu class entity, class boundary dan class control.

A. Perancangan Class Boundary

Class boundary merupakan class yang berisi interface, terdapat 2 buah class boundary pada *game* petualangan gatotkaca. Perancangan class boundary pada *game* petualangan gatot dapat dilihat pada tabel 3.13 :

Tabel 3. 13 Class Boundary

Nama *Class*	*Stereo type*
Main.as	*Boundary*
Level.as	*Boundary*

B. Perancangan Class Entity

Class entity merupakan class yang berisi data, terdapat 5 buah class entity pada *game* petualangan gatotkaca dapat dilihat pada tabel 3.14 :

Tabel 3. 14 Class Entity

Nama *Class*	*Stereo type*
Kendi.as	*Entity*
Benda.as	*Entity*
Portal.as	*Entity*
Pusaka.as	*Entity*
Koin.as	*Entity*

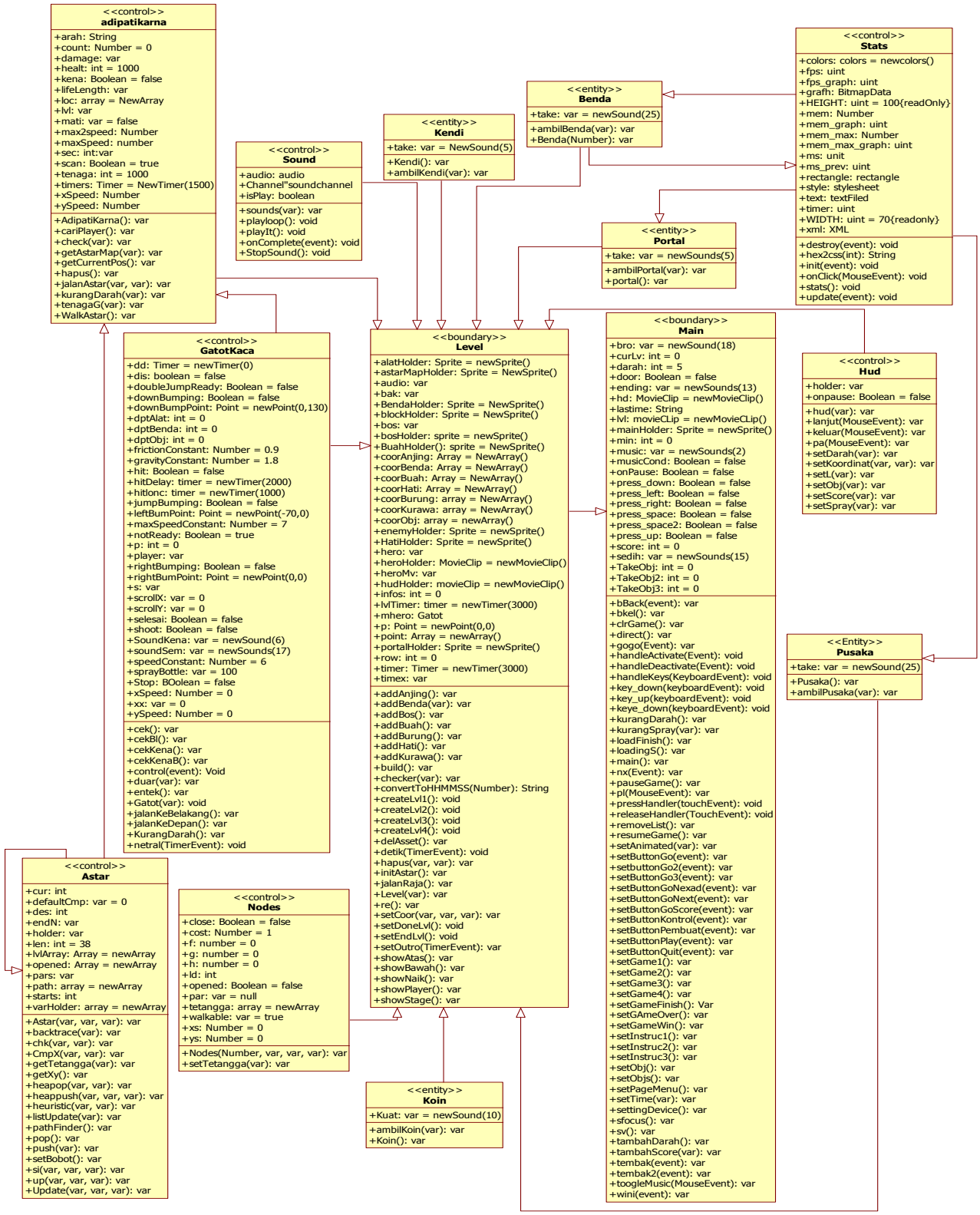
C. Perancangan Class Control

Class Control merupakan class yang mengontrol sistem, terdapat 7 buah class control pada *game* petualangan gatotkaca. Bisa dilihat pada tabel 3.15 :

Tabel 3. 15 Class Control

Nama *Class*	*Stereo type*
Adipatikarna.as	*Control*
Gatotkaca.as	*Control*
Sound.as	*Control*
Astar.as	*Control*
Node.as	*Control*
Hud.as	*Control*
Stats.as	*Control*

Class merupakan deskripsi sekelompok objek dari property(atribut), sifat (operasi), relasi antar objek dan semantik yang umum. *Class Diagram* dari *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.35 :



Gambar 3. 35 Class Diagram Petualangan Gatotkaca

3.1.9 Perancangan Sistem

Perancangan sistem yaitu suatu kegiatan untuk merancang aplikasi yang akan dibangun pada *game* petualangan Gatotkaca.

3.1.9.1 Pengenalan Karakter

Pengenalan karakater pada *game* Petualangan Gatotkaca terdapat beberapa karakter yang diantaranya dapat dilihat pada table 3.16:

Tabel 3. 16 Pengenalan Karakter Petualangan Gatotkaca

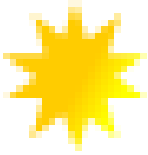
No	Nama	Gambar	Keterangan
1	GatotKaca		GatotKaca merupakan karakter utama pada *game*
2	Serigala hitam		Serigala hitam sebagai hewan yang buas untuk mengejar gatot kaca
3	Elang		Elang musuh yang menyerang diudara jika Gatotkaca terbang elang akan menyerang
4	Kurawa		Kurawa adalah musuh utusan dari raja jahat adipati karna

5	Kuda		terdapat pada level terakhir dimana kuda ini adalah tunggangan dari raja adipati karna
6	Adipati Karna		Adipati Karna adalah raja jahat yang menggunakan kereta kencana

3.1.9.2 Item Game

Item game adalah *item* pendukung yang digunakan pada *game* petualangan Gatotkaca dapat dilihat pada table 3.17:

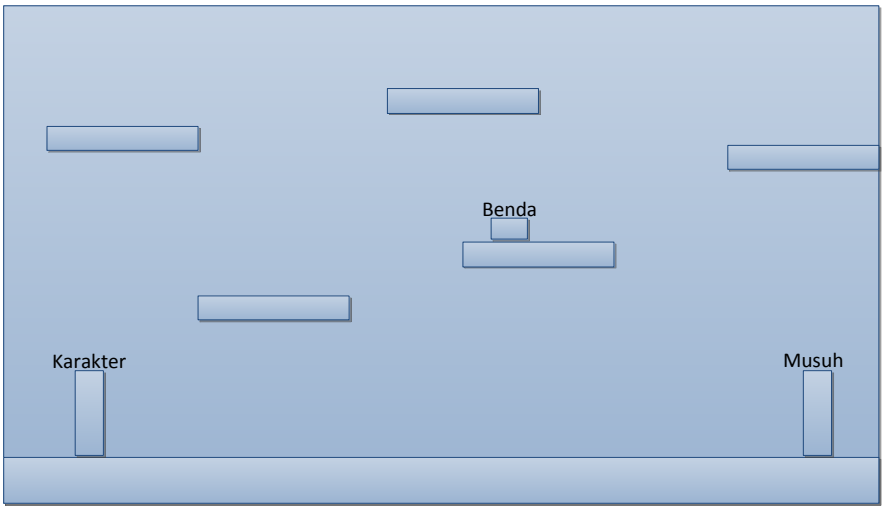
Tabel 3. 17 *Item Game*

No	Nama	Gambar	Keterangan
1	kotang Antakusuma		baju brevet penerbang bernama “kotang Antakusuma” yang membuat Gatotkaca dapat terbang dengan cepat tanpa menimbulkan ledakan supersonic.
2	Portal		Portal adalah tempat untuk melanjutkan permainan ke level selanjutnya
3	Caping Basunanda		Topi gatotkaca bernama Caping Basunanda, yang mempunyai kesaktian apabila panas tidak merasa panas dan hujan tidak menjadi basah.

No	Nama	Gambar	Keterangan
4	Kendi		Air Minum Untuk menambah darah pada Pemain.
5	Pada Kacarma		sepatu “Pada Kacarma” yang mempunyai kesaktian tidak akan kualat walaupun melintasi daerah-daerah angker.
6	Senjata Konta Jaya		Konta Jaya adalah senjata yang sangat Ampuh Namun hanya dapat digunakan Satu Kali saja.

3.1.9.3 StoryBoard

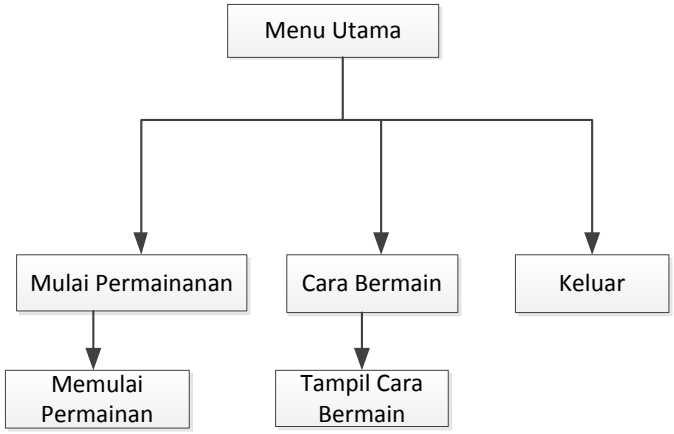
Storyboard merupakan gambaran secara umum tempat atau sketsa *level* permainan pada Petualangan Gatotkaca bisa dilihat pada gambar 3.36 :



Gambar 3. 36 *storyboard level game* Petualangan Gatotkaca

3.1.9.4 Perancangan Stuktur Menu

Perancangan struktur menu pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.37:



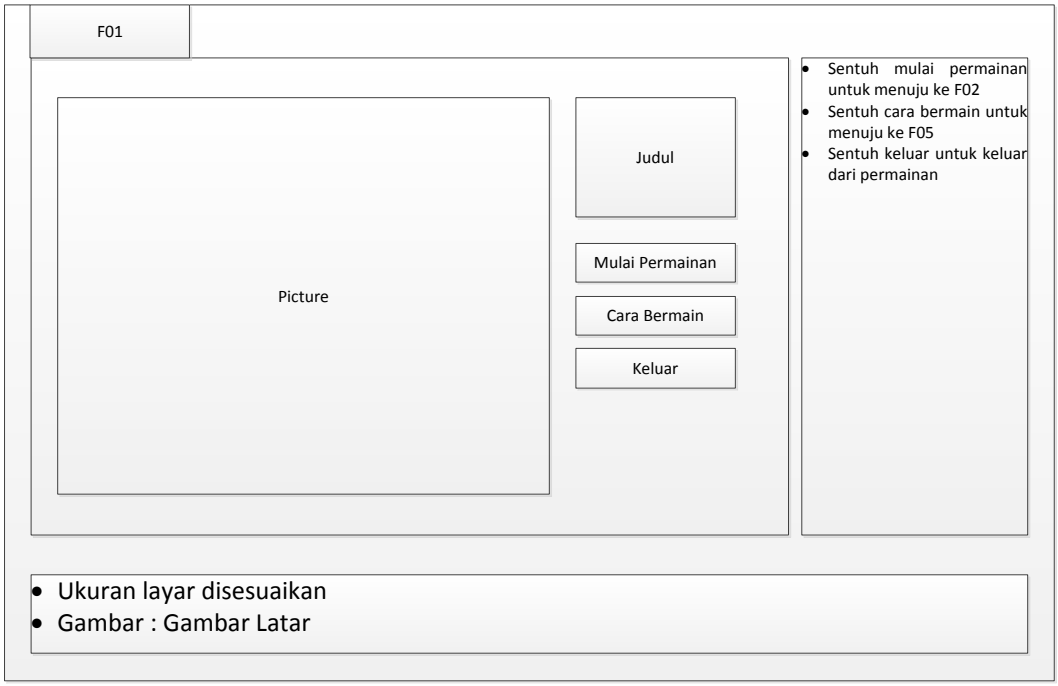
Gambar 3. 37 Struktur menu *game* Petualangan Gatotkaca

3.1.9.5 Perancangan Antarmuka

Perancangan antarmuka dibangun untuk mempermudah dalam mengimplementasikan sistem dan menggambarkan aplikasi yang akan kita bangun. Pada perancangan antar muka ini akan menjelaskan perancangan antar muka menu-menu yang ditampilkan pada *game* Petualangan Gatotkaca.

1. Perancangan Antarmuka Menu Utama

Perancangan antarmuka Menu Utama pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.38 :



Gambar 3. 38 Perancangan Antarmuka Menu Utama

2. Perancangan Antarmuka Level 1

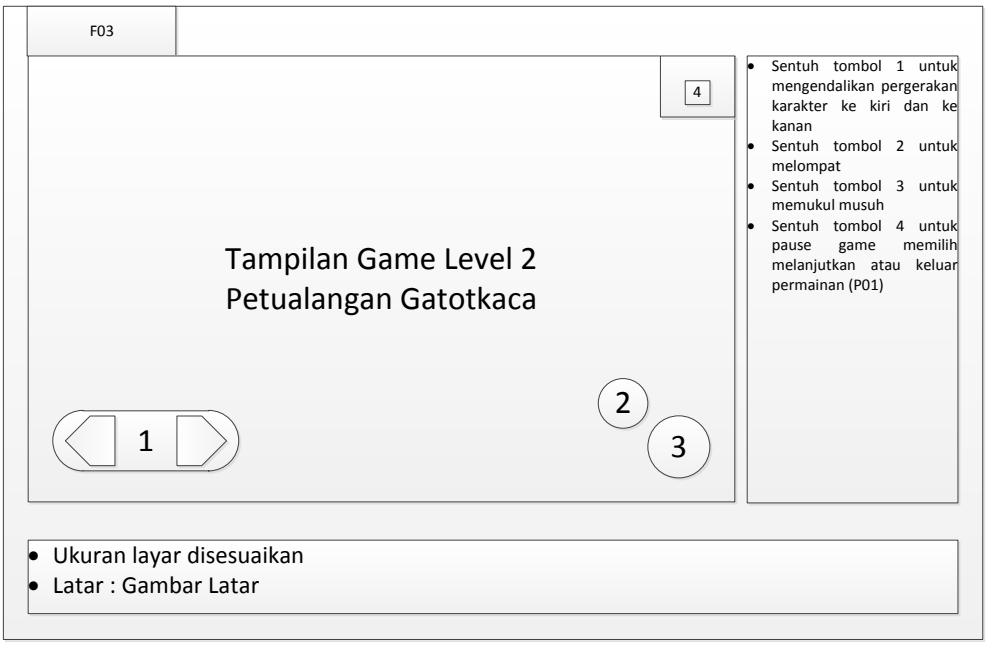
Perancangan antarmuka level 1 pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.39:



Gambar 3. 39 Perancangan Antarmuka Level 1

3. Perancangan Antarmuka Tampilan Level 2

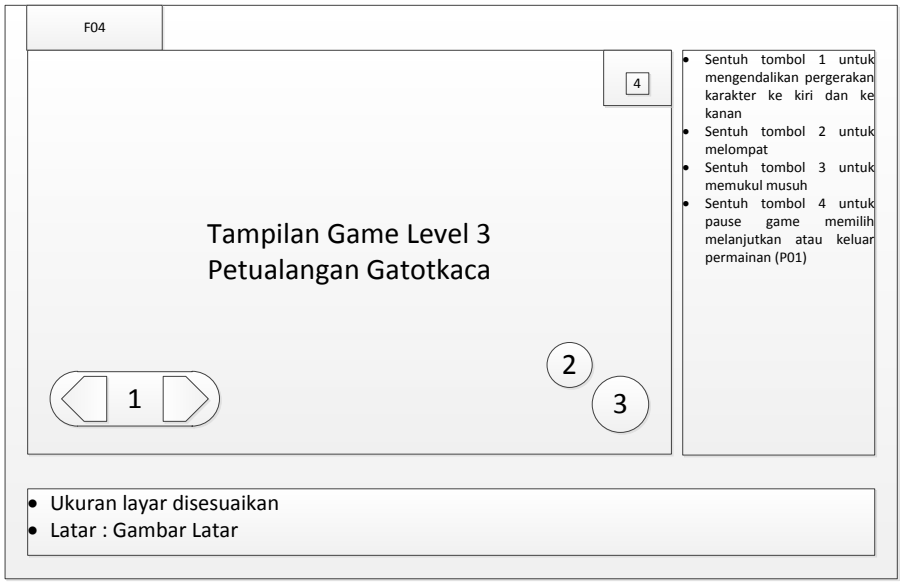
Perancangan antarmuka Tampilan Level 2 pada Petualangan Gatotkaca dapat dilihat pada gambar 3.40 :



Gambar 3. 40 Perancangan Antarmuka Tampilan Level 2

4. Perancangan Antarmuka Tampilan Level 3

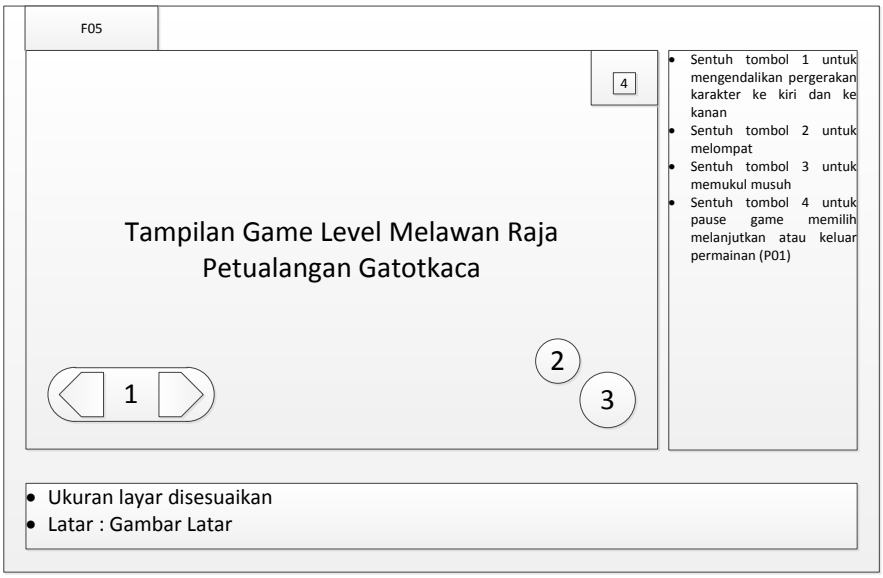
Perancangan antarmuka Tampilan Level 3 pada Petualangan Gatotkaca dapat dilihat pada gambar 3.41 :



Gambar 3. 41 Perancangan Antarmuka Tampilan Level 3

5. Perancangan Antarmuka Tampilan Level Melawan Raja

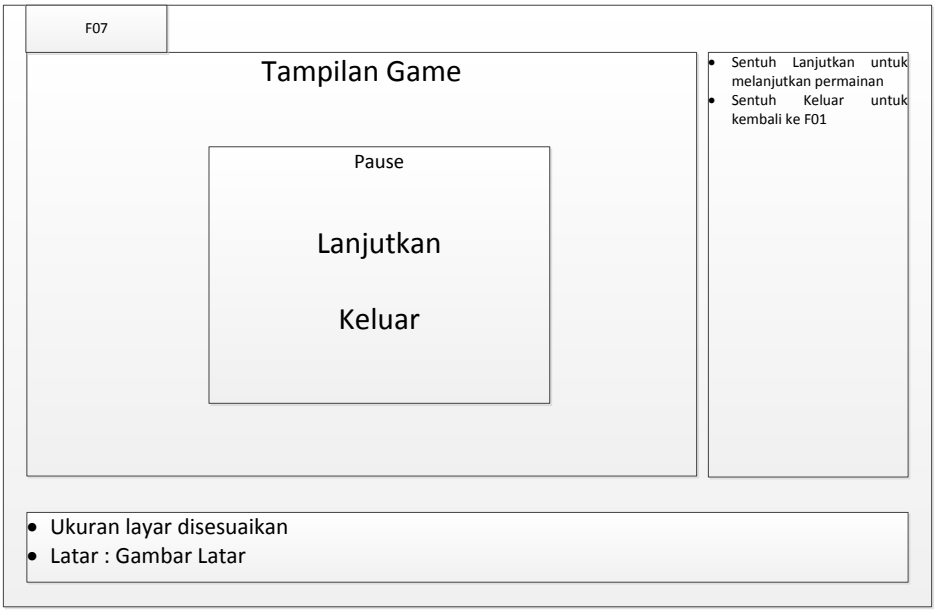
Perancangan antarmuka Tampilan Level Melawan Raja pada Petualangan Gatotkaca dapat dilihat pada gambar 3.42:



Gambar 3. 42 Perancangan Antarmuka Tampilan Level Melawan Raja

6. Perancangan Antarmuka Pause Game

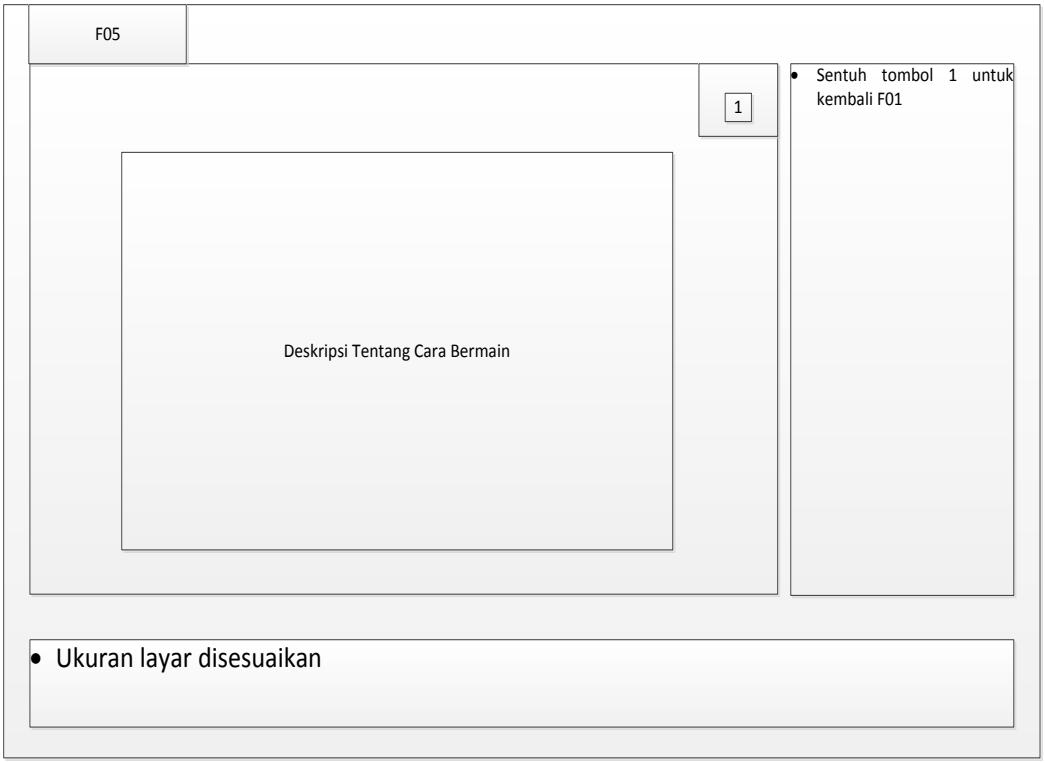
Perancangan antarmuka Pause pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.43:



Gambar 3. 43 Perancangan Antarmuka Pesan Pause Menu

7. Perancangan Antarmuka Cara Bermain

Perancangan antarmuka cara bermain pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.44 :



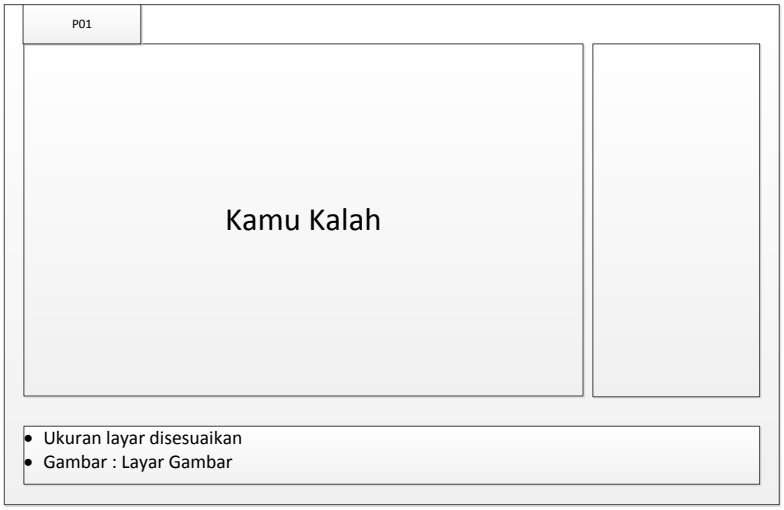
Gambar 3. 44 Perancangan Antarmuka Cara Bermain

3.1.9.6 Perancangan Pesan

Perancangan pesan bertujuan untuk memberikan peringatan jika user melakukan aktivitas tertentu yang akan memicu pesan.

1. Game Over

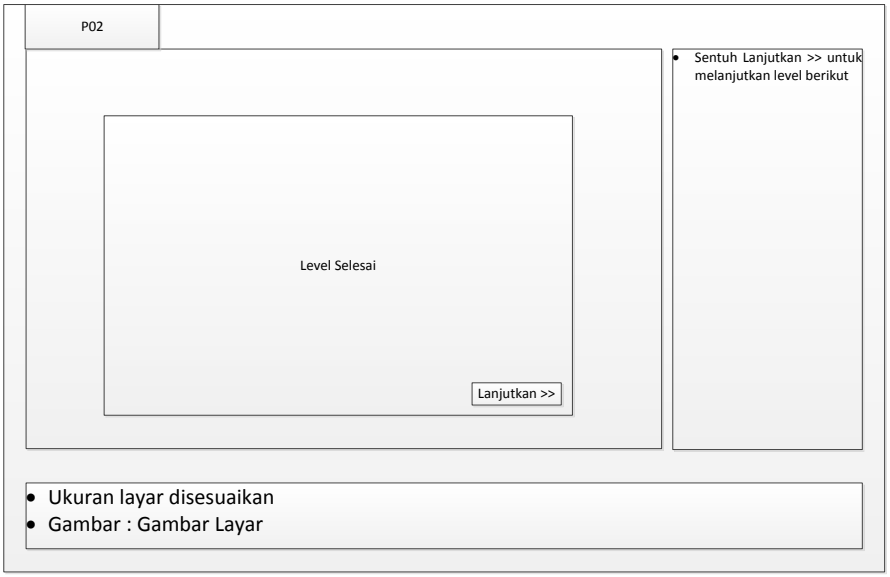
Perancangan antarmuka *GameOver* pada Petualangan Gatotkaca dapat dilihat pada gambar 3.45 :



Gambar 3. 45 Perancangan Pesan *Game Over*

2. Level Selesai

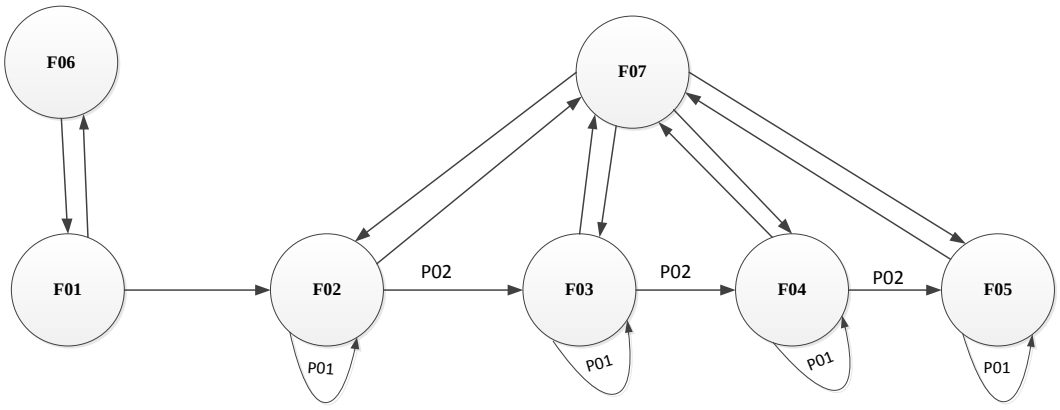
Perancangan antarmuka level selesai pada Petualangan Gatotkaca dapat dilihat pada gambar 3.46 :



Gambar 3. 46 Perancangan Pesan Level selesai

3.1.10 Perancangan Jaringan Simantik

Jaringan semantik merupakan jaringan data yang menunjukan hubungan antar berbagai objek. Jaringan simantik pada *game* petualangan gatotkaca dapat dilihat pada gambar 3.47 :



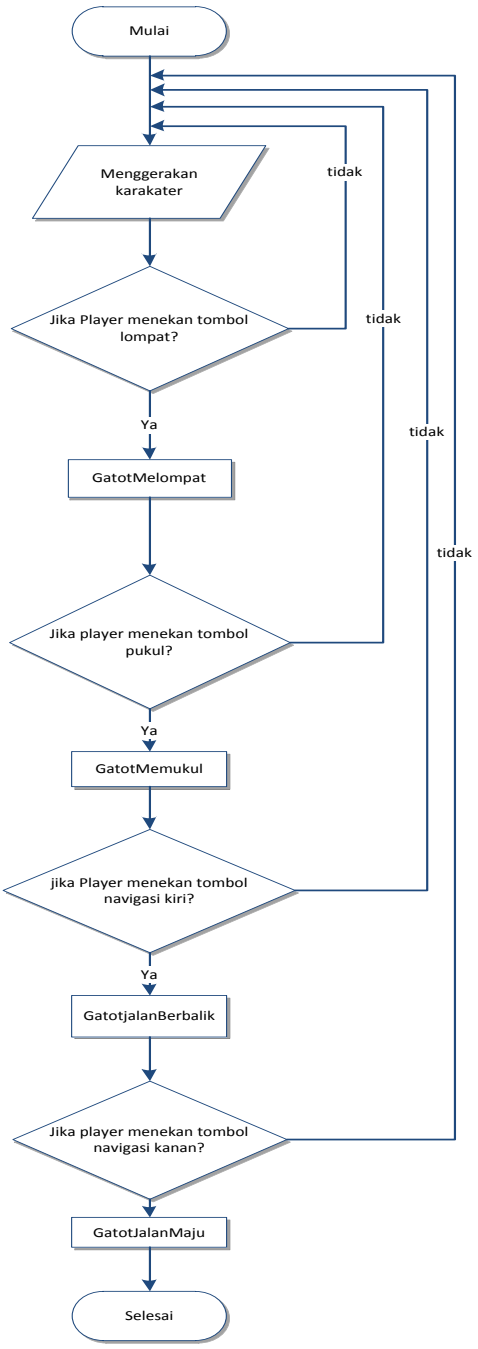
Gambar 3. 47 Jaringan Semantik

3.1.11 Perancangan Method

Perancangan *method* yaitu berfungsi mendeskripsikan *method-method* yang berada di dalam aplikasi. Method dapat dipanggil dengan menyertakan variabel, baik hanya satu variabel, banyak *variabel* atau bahkan tidak ada sama sekali. Adapun *method-method* yang terdapat pada *game* petualangan Gatotkaca adalah sebagai berikut :

1. Perancangan *Class* gatotkaca.*control* ( )

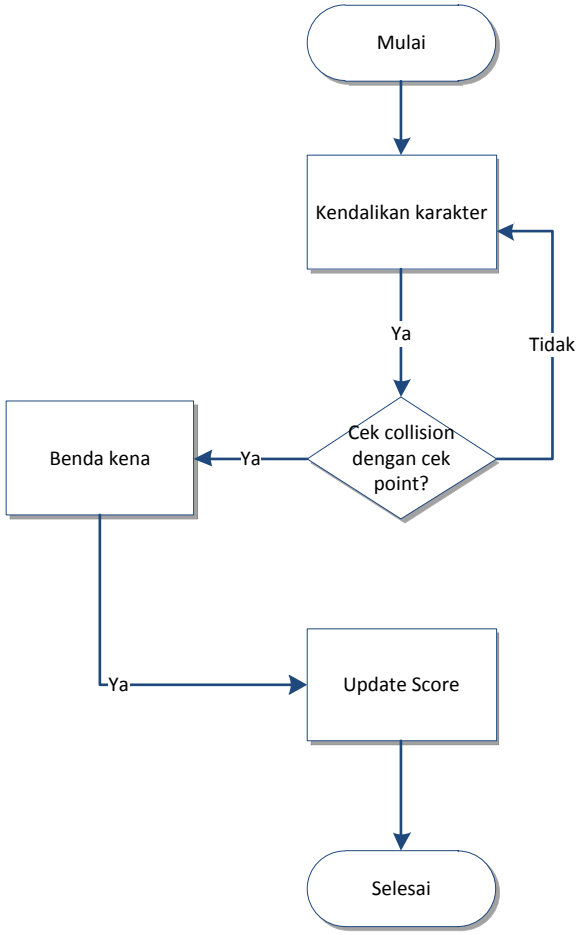
Method bergerak merupakan method yang berfungsi agar aktor melakukan pergerakan meloncat, memukul, berjalan dan berbalik. Method ini bekerja ketika pengguna menekan tombol meloncat, memukul, berjalan dan berbalik pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.48 :



Gambar 3. 48 Perancangan *class* gatot.*control*( )

2. Perancangan *Class* Hud.Set*Score* ( )

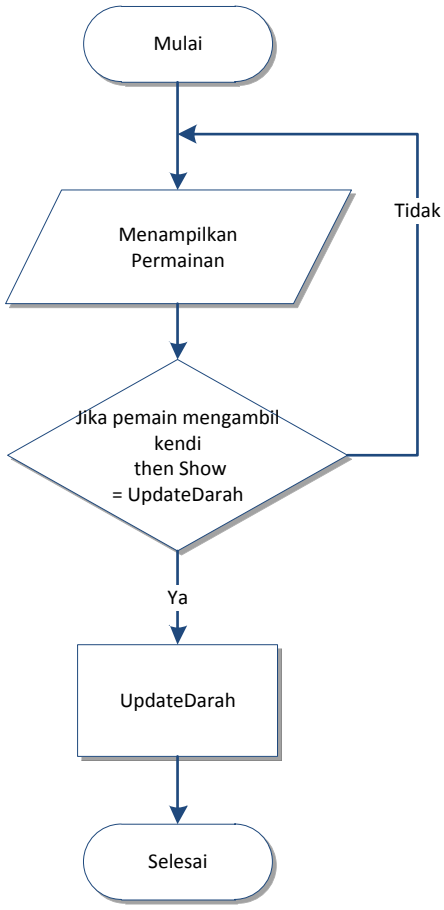
Method score merupakan *method* yang berfungsi agar pemain mendapatkan score. *Method* ini bekerja ketika karakter mengambil koin dan melawan musuh. Perancangan *method score* bisa dilihat pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.49 :



Gambar 3. 49 Perancangan *Class* main.*SetScore* ( )

3. Perancangan *Class* Hud.SetDarah( )

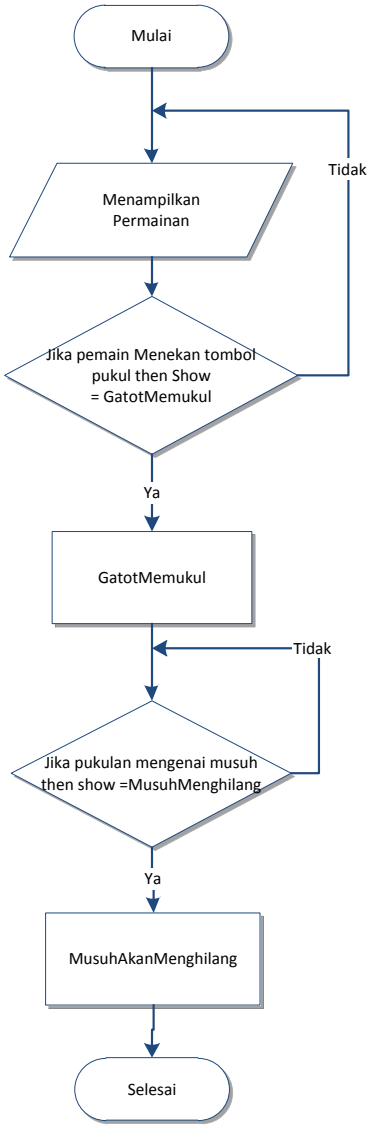
Method SetDarah merupakan method yang berfungsi agar pemain mendapatkan darah. Method ini bekerja ketika karakter mengambil darah. Perancangan *method* SetDarah bisa dilihat pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.50 :



Gambar 3. 50 Perancangan *Class* Hud.SetDarah( )

4. Perancangan *Class* Main.tembak( )

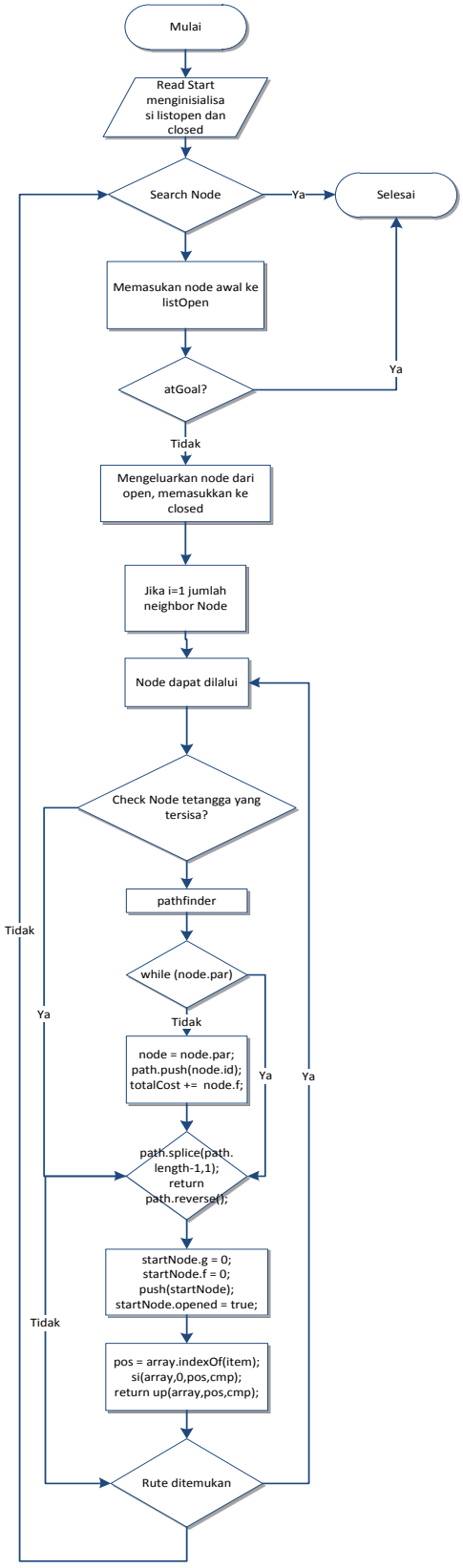
Method tembak merupakan *method* yang berfungsi agar pemain bias memukul musuh. *Method* ini bekerja ketika karakter memukul musuh. Perancangan *method* tembak bisa dilihat pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.51 :



Gambar 3. 51 Perancangan *Class* Main.tembak( )

5. Perancangan Class Astar.Pathfinder

Method pathfinder merupakan method yang berfungsi agar raja bergerak mendekati pemain. Method ini bekerja ketika karakter bergerak. Perancangan method pathfinder bisa dilihat pada *game* Petualangan Gatotkaca dapat dilihat pada gambar 3.52 :



Gambar 3. 52 Perancangan Class Astar.PathFinder