BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan penguraian dari satu sistem informasi yang utuh kedalam bagian – bagian komponennya yang dimaksudkan untuk mengidentifikasi dan mengevaluasi segala permasalahan dan hambatan yang terjadi serta kebutuhan yang diharapkan dapat menjadi acuan untuk diusulkannya perbaikan – perbaikan.

3.1.1 Analisis Masalah

Masalah yang diangkat dari penelitian skripsi ini adalah tentang pembangungan *game* cerita rakyat fabel.

3.1.2 Ananlisis Game Sejenis

Analisis *game* sejenis dilakukan untuk memberikan gambaran sistem agar lebih mudah dalam pembangunan dan sebagai pembanding untuk kesempurnaan *game* yang akan dicapai. Beberapa permasalahan yang merupakan hasil dari analisis yang dilakukan dijadikan sebagai latar belakang dalam pembangunan aplikasi *game* cerita rakyat fabel.

3.1.2.1 *Storyline Princess* Repunzel

Dahulu kala hiduplah sepasang suami-istri tanpa anak yang tinggal di sebelah taman yang ditutupin oleh tembok besar yang dimiliki oleh seorang penyihir. Suami istri tersebut hidup sangatlah sederhana, mereka tidak mengingikan harta yang berlimpah atau pun makanan yang enak, mereka hanya menginginkan seorang anak hadir didalam kehidupan keluarga sederhana ini. Setelah penantian yang lama, akhirnya sang istri hamil. Ketika sang istri berjalan jalan di sekitar rumah penyihir, Si istri ini melihat bunga rapunzel yang tumbuh indah di taman rumah si penyihir dan sang istri saat itu sangat menginginkan bunga rapunzel tersebut. Sang suami jelas menuruti permintaan sang istri tersebut.

Maka dari itu ketika malam hari ia manyelinap ke dalam taman si penyihir untuk mengambil bunga rapunzel yang indah tersebut. Malam pertama sang suami pun berhasil mengambil bunga rapunzel tanpa sepengetahuan penyihir, malam kedua pun sang suami berhasil lagi mendapatkan setangkai bunga rapunzel dari

taman penyihir itu. Namun pada malam ketiga, ketika ia baru selesai mengambil setangkai bunga rapunzel dan ingin membawanya pulang kerumah dari taman si penyihir, tiba tiba sang pemilik taman pun datang, si penyihir melihat bunga rapunzel peliharaannya di curi oleh seseorang.

Sang suami memohon belas kasihan kepada sang penyihir untuk memaafkannya, si penyihir setuju untuk mengampuni si suami dengan satu syarat. Syaratnya yaitu anak yang ada didalam kandungan istrinya harus diserahkan kepadanya begitu anak itu lahir. Karena sudah putus asa dan ketakutan, si suami pun setuju dengan syarat yang di ajukan oleh sang penyihir. Pulanglah sang ayah kerumah dan memberikan setangkai bunga rapunzel itu, namun ia tidak menceritakan kejadian yang barusan ia alami saat mengambil bunga rapunzel tersebut

Beberapa hari kemudian ternyata si istri melahirkan seorang bayi perempuan, keluarga kecil itu pun sangat bahagia, namun kebahagiaan itu tak berlangsung lama karena tak beberapa lama bemudian si penyihir pun datang dan langsung mengambil bayi yang baru saja lahir tersebut dan membawanya ke tempat tinggalnya. Sang istri sangat sedih karena anak yang selama ini diharap harap kan diambil begitu saja oleh sang penyihir tersebut, sang suami tak bisa melakukan apa apa.

Setelah sang bayi dibawa kerumah sang penyihir, sang bayi pun dirawat oleh si penyihir. Penyihir itu memberi nama bayi itu rapunzel, sesuai dengan nama bunga yg waktu itu di curi oleh ayah si rapunzel. ketika si rapunzel berumur 18 tahun, Rapunzel tumbuh sebagai gadis yang cantik dan mempunyai rambut berwarna emas yang sangat indah. si penyihir memindahkan dan mengurungnya kedalam sebuah menara ditengah hutan tanpa tangga, tanpa pintu dan amat sangat tinggi, dan hanya terdapat 1 kamar, dan satu jendela di menara itu. Kalau si penyihir ingin naik ke atas menara untuk menemui rapunzel, ia akan berdiri dibawah menara dan memanggil rapunzel lalu berteriak “rapunzel, rapunzel, julurkan rambutmu ke bawah agar aku bisa naik ke atas lewat rambut emas mu itu..

Ketika rapunzel sudah mendengar kalimat itu ia langsung melilitkan rambut emasnya yang panjang ke sebuah pengait yang terletak di sebelah jendela yg sengaja di buat penyihir untuk memudahkan rapunzel mengangkat si penyihir untuk naik ke atas menara. ada si. Biasanya si penyihir akan memanjat ke atas menggunakan rambut rapunzel itu.

Disuatu pagi yang cerah seorang pangeran yang sedang berburu rusa di tengah hutan, ia mendengar suara perempuan yang sedang bernyanyi. Karena sang pangeran tertarik dengan suara merdunya, si pangeran akhirnya mencari dari mana asal suara merdu itu, setelah lama ia berjalan bersama kudanya, ia menemukan sebuah menara yang sangat tinggi, ternyata memang benar suara merdu itu berasal dari atas menara yang sangat tinggi itu, ia ingin sekali naik ke atas menara tinggi itu namun apa daya saat ia mengelilingi menara ia tidak dapat menemukan tangga atau alat semacamnya untuk naik ke atas sana. Namun sang pangeran tidak putus asa. Dan tetap mencari cara bagaimana ia bisa naik ke atas menarang yang tinggi itu.

Hari hari berikutnya sang pangeran lebih sering mendatangi menara tersebut karena ia makin penasaran siapakah gerangan wanita yang ada diatas menara yg berada ditengah hutan yg tak berpenghuni ini, dan pada suatu saat di penyihir datang untuk menemui rapunzel. Sang pangeran pun tau sekarang bagai mana caranya ia bisa masuk ke dalam menara itu.Waktu si penyihir pergi, si pangeran meminta rapunzel untuk menjulurkan rambut emasnya ke bawah, dan kemudian si pangeran memanjat seca ra perlahan ketika sampai dimenara, sang rapunzel terkejut karena bukan penyihir yang datang menemuinya melaikan orang asing. Si penyihir bilang bahwa rapunzel dilarang menerima orang asing masuk kedalam menara itu. Namun sang pangeran menenangkan rapunzel. Mereka akhirnya mulai akrab satu sama lain.

Rapunzel dan pangeran awalnya hanya berteman, namun lama kelamaan mereka saling jatuh cinta, sang pangeran pun akhirnya melamar si rapunzel dan rapunzel bersedia untuk menikah dengannya. Setelah rapunzel mengiyakan lamaran sang pangeran, mereka berdua menyusun rencana untuk bagaimana

caranya iya keluar dari menara itu setiap malam sang pangeran sengaja datang untung membawakan kain sutra sebagai alat untuk rapunzel turun dari menara itu.

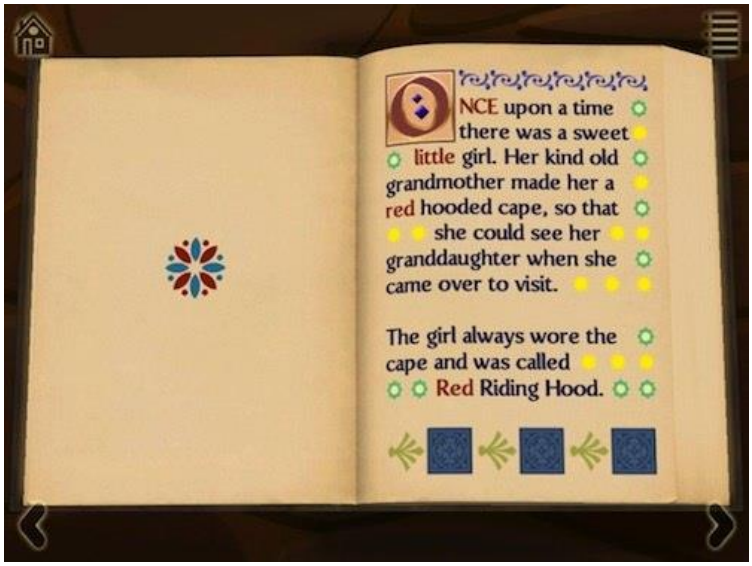
Belum rencana yang selama ini di rancang oleh rapunzel dan pangeran, dengan polosnya rapunzel bertanya ke pada sang penyihir ketika sang penyihir datang untuk menemuinya mengapa gaun-gaun yang ada di lemari semakin sesak saja dibagian perut.nya. sang penyihirpun murka karna tau si rapunzel hamil, penyihir akhirnya menghukum rapunzel, penyihir memotong rambut rapunzel dan membuangnya ke alam liar.

Keesokan harinya pangeran datang untuk menjemput rapunzel untuk tinggal di istana, ktika sang pangeran memanggil rapunzel dan rambut rapunzel diturunkan ketika yang pangeran sampai di atas ia terkejut, ternyata bukan rapunzel yang ada di dalam menara melaikan si penyihir, si penyihir pun lalu mendorong sang pangeran sampai terjatuh di semak semak berduri dan membuat mata sang pangeran buta. Sudah beberapa hari sang pangeran berjalan tanpa tujuan. Di lain tempat rapunzel akhirnya melahirkan 2 anak kembar lelaki dan perempuan, suatu hari ketika si rapunzel sedang menimba air, ia meliahat sang pangeran datang sempoyongan dari arah hutan dan terjatuh, si rapunzel pun menghampirinsi pangeran dan terkejut dangan keadaan sang pangeran, rapunzel pun menangis, dan air mataya itu dapat menyembuhkan mata sang pangeran. Mereka saling berpelukan dan akhirnya sang pangeran memboyong rapunzel dan ke 2 anaknya ke istana dan hidup menjadi keluarga yang bahagia.

3.1.2.2 *Gameplay* Grimm's Repunzel 3D – *Interactive Pop-Up Books*

Grimm Rapunzel ~ 3D *Interaktif Pop - up Book* merupakan aplikasi universal yang menceritakan dongen klasik dari certa grimm yaitu cerita *princess* Repunzel. Dalam game ini ada 3 mode permainan, yaitu mode *autoplay, read to me*, dan *read my self*. Mode *autoplay* adalah mode dimana sistem akan membacakan cerita dan memainkan permainannya sendiri, dan pemain hanya menonton. Sedangkan mode *reat to me* adalah mode dimana sistem akan membacakan ceritanya saja, sedangkan pada saat game dimulai, pemain yang memainkan game tersebut, dan mode *reat my self* adalah mode dimana sistem melepas semua kendali dalam permainan dan cerita hanya dilakukan oleh pemain.

Pada saat bermain, akan ada cerita dari repunzel yang akan dijalankan sesuai dengan mode yang telah dipilih pada menu utama di awal permainan. Setelah cerita pertama dijalankan atau dibacakan, makan akan masuk kedalam *game* yang berbentuk *pop-up book* 3 dimensi. Dalam *game* pemain akan memainkan permainan sesuai dengan cerita yang telah dijalankan atau dibacakan pada saat sebelum memasuki game.



Gambar 3.1 Grimm's Repunzel 3D – Interactive Pop-Up Books

Setiap satu bacaan cerita, makan akan di tampilkan dalam *game* berbentuk *pop up* 3 dimensi setelanya sampai cerita selesai. Kekurangan dari *game* ini adalah cerita yang diceritakan adalah cerita atau dongeng dari luar Indonesia, sehingga kurang tepat jika diunduh atau dimainkan oleh anak – anak Indonesia. Bahasa yang digunakan pun bahsa inggris, sehingga anak – anak yang memainkan game ini tidak akan mengerti alur dari cerita yang akan dimainkan.

Sedangkan kelebihan dari game ini adalah gambar 3 dimensi yang sangat menarik, memang didisain untuk anak – anak dan dapat diputar ke kiri, kanan, atas atau bawah. Setiap *game* yang dimainkan mengacu pada cerita yang dibacakan, sehingga keserasian alur cerita dengan game terbentuk.



Gambar 3.2 Grimm's Repunzel 3D – Interactive Pop-Up Books

Tabel 3.1 Analisis *Game* Sejenis

Nama *Game*	*Genre Game*	Desain Grafis	*Storyline* dan *gameplay*	*multiplayer*	Basis
Grimm's Repunzel 3D – Interactive Pop-Up Books	*Adventure*	Memiliki gambar 3D yang baik.	Alur cerita bersifat fiktif bukan diambil dari hal-hal yang pernah terjadi.	tidak	*Mobile*

3.1.3 Analisis *Game* yang Dibangun

3.1.3.1 *Storyline Game* Kancil dan Buaya

Kancil adalah hewan yang dikenal cerdas, dia adalah hewan yang baik hati, senang menolong teman – teman hewan yang lainnya apabila mereka sedang dalam kesusahan. Pada suatu hari makanan dihutan tempat kancil berada sudah

habis, jika kancil menginginkan makanan yang banyak dan melimpah, kancil harus mencarinya di luar daerah tempat dia berada sekarang. Setelah berjalan sekian lama, kancil pun kehausan, maka kancil mencari sungai untuk dapat meminum airnya. Setelah menemukan sungai, kancil pun minum banyak air dari sungai. Setelah puas meminum air, kancil pun memilih untuk mencari makanan disekitar sungai, agar jika nanti kancil haus, kancil tidak susah untuk mendapatkan air untuk diminum.

Setelah sekian lama berjalan di pinggir sungai, kancil pun melihat banyak sekali makanan di hutan sebrang sungai, maka kancil pun berfikir bagaimana caranya agar dapat menyebrangi sungai. Tidak mudah untuk menyebrangi sungai, karena sungai yang akan diseberangi oleh kancil dijaga oleh para buaya lapar, jika saja kancil tertangkap oleh buaya itu maka kancil akan langsung dimakan. Buaya tersebut berada didalam sungai, dan sesekali muncul ke permukaan untuk melihat - lihat sekitar. Sadar akan bahaya yang dihadapi oleh kancil, maka kancil tidak kehabisan akal, kancil mencari cara bagaimana agar pada saat kancil menyebrang tidak di ganggu oleh buaya – buaya tersebut. Setelah berfikir, kancil pun mencoba melempari buaya yang muncul ke permukaan dengan batu – batu yang ada dipinggir sungai, buaya yang terkena lemparan batu dari kancil pingsan dan tenggelam di sungai. Setelah kancil melempari semua buaya dengan batu, kancil pun berani untuk mencoba menyebrangi sungai, kancil mencoba menyebrangi sungai dengan melompati batu – batu yang ada di sungai.

Pada saat kancil melompati batu, buah – buahan jatuh dari pohon dan masuk ke sungai serta mengalir melewati kancil, kancil pun tergeiur untuk memakan buah – buahan yang jatuh tersebut. Namun pada saat kancil akan mengambil buah yang mengalir terbawa oleh air sungai, tiba - tiba seekor buaya ingin membuat kancil terjatuh ke sungai, kancil pun kaget dan langsung melompati ke batu yang berada didepannya. Pada saat kancil melompati batu untuk sampai ke sebrang sungai, kacil dikagetkan dengan batang pohon yang hampir membuat dia terjatuh, kancil pun menunggu batang kayu tersebut melewati lintasa kancil agar dapat melompati batu berikutnya.

Sadar kancil akan sampai ke sebrang sungai, buaya pun mencoba menjatuhkan kancil yang berada di atas batu dengan menabrakan dirinya ke batu tempat kancil berpijak agar kancil dapat jatuh kesungai. Tetapi sayangnya kancil adalah hewan yang cerdik, pada saat buaya menabrakan dirinya ke batu dimana kancil berada, kancil pun langsung melompat ke batu berikutnya, dan akhirnya sampai ke sebrang sungai. Dengan sumringahnya kancil akhirnya dia dapat menyebrangi sungai yang dipenuhi buaya dan dapat memakan banyak sekali makanan yang ada ditempat tersebut. Karena buaya gagal untuk mendapatkan kancil, maka mereka bersumpah tidak akan melepaskan sang kancil apabila bertemu pada masa yang datang. Dendam buaya tersebut terus membara hingga hari ini, sementara itu sang kancil terus melompoat kegirangan dan teus meninggalkan buaya – buaya tersebut dan menghilang didalam kebun buah – buahan untuk menikmati buah-buahan yang sedang matang.

3.1.3.2 *Gameplay Game* Kancil dan Buaya

Ketika memulai permainan, akan ada 2 pilihan menu, yaitu *play* dan *option*. Jika pemain menekan tombol *play* maka pemain akan langsung menuju permainan, jika pemain menekan tombol *option*, maka akan muncul tampilan *option* yang berisi *music* dan *developer*, serta tombol back untuk kembali ke menu utama. Pada *music* pemain dapat melakukan *on* atau *off music*, dan pada developer hanya akan menunjukkan pembuat game.

Ketika permainan dimulai, akan ada 2 gambar pembuka cerita, gambar yang pertama menceritakan bahwa ditempat si kancil berada sekarang sudah sedikit makanan, sehingga memaksa si kancil untuk keluar hutan tempat dia berada untuk mencari makanan. Dan setelah gambar yang menyatakan bahwa ditempat si kancil berada sekarang sudah sedikit makanan, maka akan ada gambar kedua yang menyatakan bahwa si kancil telah menemukan sungai, dan bertemu dengan buaya, lalu setelah gambar tersebut akan ada game permainan yg akan dimainkan oleh pemain.

a. Stage 1

Si kancil akan menghadapi buaya yang lapar. Buaya tersebut berada didalam sungai. Maka game yang akan dimainkan dalam Stage 1 ini adalah si

kancil akan melempari buaya dengan batu, buaya akan muncul kepermukaan sungai, dan bersembunyi kembali, ketika buaya muncul ke permukaan, maka batu dilemparkan ke arah buaya, jika batu mengenai buaya, maka buaya akan mati, dan akan muncul poin, waktu adalah parameter untuk menetukan lamanya pemain bermaian dalam Stage ini. Dalam Stage 1 ini akan ada 3 tantangan, pada tantangan pertama pemain harus mengumpulkan poin sebanyak 50 point agar dapat menyelesaikan tantangan 1. Untuk menyelesaikan tantangan 2 kancil harus mengumpulkan 120 point dari point awal tantangan 1 yaitu 50 poin. Untuk menyelesaikan Stage 1 ini maka kancil harus mengumpulkan 180 poin.

b. Stage 2

Setelah menyelesaikan Stage 1, maka si kancil akan melanjutkan ke Stage 2. Dalam Stage ini si kancil harus menangkap buah – buahan yang mengalir disungai, dan akan ada batu tempat si kancil berpijak. Tantangannya adalah akan ada buaya yang akan memakan buah tersebut jika si kancil tidak dapat mengambilnya. Kancil akan berpijak di salah satu batu, jika buah tersebut melewati si kancil dan si kancil tidak mengambilnya, maka si kancil harus ke batu yang lebih dekat dengan buah untuk mengambil buah tersebut, atau buah tersebut dimakan oleh buaya.

Jika kancil mendapatkan buah, maka akan ada poin untuk si kancil. Dalam Stage ini pun akan ada 3 tantangan untuk pemain, tantangan 1 pemain akan dengan mudah mendapatkan buah – buahan yang mengalir disungai, dan buaya yang akan menghalangi kancil untuk mendapatkan buah, namun buaya ditantangan pertama ini hanya akan ada 1 ekor buaya, sedangkan di tantangan selanjutnya akan ada 2 ekor buaya. Untuk lolos dari tahap 1 ini, kancil harus mengumpulkan 80 poin, itu artinya kancil harus mendapatkan 8 buah yang mengalir dari sungai agar dapat menuju tantangan ke 2. Dan untuk melanjutkan ke tantangan berikutnya kancil harus mengumpulkan 160 poin.

c. Stage 3

Dalam Stage ini kancil akan melompati batu untuk sampai ke sebrang sungai, namun dalam melompati batu ini tidak akan mudah, akan ada penghalang seperti batang pohon yang menghalangi lompatan si kancil, batu yang mengapung

sehingga jika diinjak oleh si kancil maka akan tenggelam. Jika si kancil jatuh ke air, maka si kancil akan mati.

Dalam Stage 3 ini pun akan ada 3 tantangan, tantangan 1 kancil akan melompati bebatuan. Dalam melompati bebatuan di tantangan 1 ini, kancil akan disusahkan oleh batang pohon yang menghalangi kancil untuk melompoat ke batu yang berikutnya, batang pohon tersebut akan menutup dan membuka jalan untuk kancil, jika si kancil melompat pada saat batang pohon menutupi batu, maka si kancil akan jatuh ke sungai. Apabila kancil terjatuh ke sungai, maka akan ada pilihan bermain lagi atau kembali ke menu utama. Tantangan ke 2, halangan pun akan berfariasi di tantangan ke 2 ini, bukan hanya batang pohon yang akan menghalangi lompatan kancil, tetapi akan ada batu yang mengapung, yang dapat mengelabui si kancil, apabila kancil menginjak batu mengapung maka kancil akan tenggelam, dan kancil akan tenggelam. Apabila kancil tidak terjatuh dan dapat mencapai batu terakhir di tantangan ke 2, maka kancil berhak menuju ke tantangan selanjutnya. Ditantangan ke 3 akan ada buaya, buaya akan menabrakan dirinya ke batu tempat kancil berpijak agar kancil dapat jatuh ke sungai.

3.1.3.3 *Stage Game* Kancil dan Buaya

Dalam *game* cerita rakya fabel ini memiliki 3 stage. Berikut stage dapat dilihat pada table 3.2

Tabel 3.1 Penjelasan Stage Pada *Game*

Tingkatan *Stage*	Deksripsi
Stage 1	Pada *stage* ini pemain hanya akan melemparkan batu ke arah buaya, apabila lemparan batu mengenai buaya, maka buaya akan mati.
Stage 2	Disini pemain akan mulai beradu cepat dengan buaya untuk mendapatkan buah yang mengalir di sungai, apabila jumlah buah yang didapat kancil telah mencapai buah

	dengan point dan waktu yang telah ditentukan, maka kancil dapat melanjutkan permainan, begitu pula sebaliknya.
Stage 3	Disini pemain dituntut untuk lebih berkonsentrasi karena pemain harus melewati batu – batuan sungai, apabila pemain tidak dapat berkonsentrasi maka pemain akan jatuh ke sungai dan dimakan oleh buaya.

3.1.3.4 Analisis Skoring

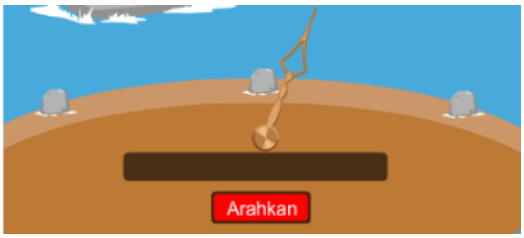
Skoring pada Stage 1, yaitu ketika lemparan batu kancil mengenai buaya, maka akan bertambah 10 point untuk kancil. Apabila poin kancil mencapai 50, maka kancil akan dapat menuju tantangan ke 2. Dalam tantangan ke 2 ini kancil harus mengumpulkan 120 poin untuk dapt melanjutkan ke tantangan ke 3, dan apabila kancil dapat mencapai poin 180 di tantangan 3, maka kancil akan memenangkan Stage 1.

Skoring pada Stage 2, yaitu ketika kancil harus mengambil buah yang hanyut disungai. Kancil akan mendapatkan point 10 apabila kancil dapat mengambil buah. Dalam Stage 2, akan ada 3 tahap, yaitu tahap 1, tahap 2, dan tahap 3. Pada setiap tahap akan berbeda skor yang di butuhkan untuk memenangkan tiap tahapnya. Pada tahap 1, kancil harus mengumpulkan 80 point agar dapat melanjutkan ke tahap 2, pada tahap 2 kancil harus mengumpulkan 80 point untuk dapat melanjutkan ke tahap 3, dan pada tahap 3, kancil harus mengumpulkan 70 point untuk dapat melanjutkan ke Stage 3.

3.1.3.5 Analisis Busur

Pada *stage* 1 pemain harus mengalahkan buaya dengan cara melempari buaya dengan batu menggunakan busur. Adapun cara menggunakannya adalah dengan mengarahkan busur ke arah buaya dan menentukan kekuatan lemparan batu. Berikut analisis lemparan batu pada *stage* 1.

Table 3.2 Analisis Busur

Busur	Deskripsi
	Ini adalah busur yang digunakan pada *stage* 1. Busur ini akan bergerak ke kanan dan ke kiri. Busur ini berbentuk animasi yang terus di ulang dan akan berhenti jika pemain menyentuh tombol arahkan.
	Ketika tombol arahkan di sentuh, maka bar akan mulai bergerak dengan parameter warna sebagai ukuran jaraknya, yaitu hijau untuk dekat, kuning untuk sedang, dan merah untuk jauh. Dan ketika tembak di sentuh maka batu akan keluar dari ujung busur dengan jarak yang ditentukan oleh pemain dari bar.

Dalam pengaplikasiannya didalam program ketika pemain menyentuh tombol tembak, maka bar akan berhenti. Maka warna pada bar akan mentukan jauh atau tidaknya lemparan dengan menggnakan *frame*, dan *frame* akan dikalikan dengan *pixel display* x dan y yang dan dikalikan 100 / 40. Maka setelah perhitungan tadi akan keluar batu dari busur dengan kekuatan yang telah dihitung dari bar.

3.1.3.6 Gambaran Umum *Game* Kancil dan Buaya

Game yang akan dibagun ber*genre adventure*, game ini memiliki grafis 2D (2 Dimensi). Latar cerita yang diambil adalah latar dari cerita fabel "*Si Kancil dan Buaya*". Berikut adalah fitur – fitur yang terdapat dalam game :

a. Grafik *game* adalah 2D (2 dimensi).

b. *Game* memiliki 3 Stage.

c. *Game* ini dibuat dengan *genre Adventure*.

d. *Game* ini memiliki menu *option* untuk mengatur suara dan *mute* music.

3.1.4 Analisis Algoritma *Simplified Memory-Bounded* A* (SMA*)

Untuk masalah tertentu, dimana memori computer terbatas, algoritma A* mungkin tidak mampu menemukan solusi karena sudah tidak tersedia memori untuk menyimpan simpul – simpul yang dibangkitkan. SMA* mengingat *f*-Cost dari setiap iterasi sampai sejumlah simpul yang ada di dalam memori. Karena batasan memori dalam jumlah tertentu, maka kita dapat membatasi pencarian hanya sampai pada simpul-simpul yang dapat dicapai dari *root* sepanjang suatu jalur yang memorinya masih mencukupi. Kemudian mengembalikan suatu rute terbaik diantara rute-rute yang ada dalam batasan jumlah simpul tersebut. Disini yang dimaksud *f*-Cost adalah gabungan biaya sebenarnya dan biaya perkiraan, yang secara matematika dinyatakan sebgai *f*-*C*ost = g(n) + h(n) **[8]**.

Untuk lebih memahami algoritma SMA*, bisa dilihat dari kasus SMA* yang diterapkan dalam game cerita rakyat fable ini. Seperti terlihat pada gambar 3.3. Diasumsikan dalam game, karakter ingin bergerak dari node awal ke tujuan. Untuk memudahkan setiap node diberi warna, warna biru adalah titik mulai, dan warna *orange* titik tujuan, warna abu – abu adalah dinding atau collider, warna biru muda adalah node yang berada di closed list dan warna hijau muda adalah node yang berada di open list , disini diasumsikan hanya tersedia 2 node untuk open list.

23	24	Tujuan 25	26	27	28	29
22	Penghalang	21	Penghalang	20	Penghalang	19
12	13	14	15	16	17	18
11	Penghalang	10	Penghalang	9	Penghalang	8
1	2	3	Node Awal 4	5	6	7

Gambar 3.3 Langkah 1 Pencarian SMA*

Mulai pencarian dari node awal dan tambahkan node awal ke list. Generate suksesor dari node awal, kemudian hitung f(s) nya dengan rumus:

$$f(s) = g(s) + h(s)) \quad (3.1)$$

dengan keterangan :

f(s) = fungsi evaluasi *node* yang dicari

f(n) = fungsi evaluasi *node* parent

g(s) = biaya yang sudah dikeluarkan

h(s) = estimasi biaya untuk sampai pada tujuan

Untuk perhitungan nilai H digunakan fungsi heuristic, metode yang digunakan di dalam contoh ini adalah metode *Euclidian* yang dirumuskan dengan :

$$h(n) = \sqrt{((tujuan.x - n.x)^2 + (tujuan.y - n.y)^2)} \quad (3.2)$$

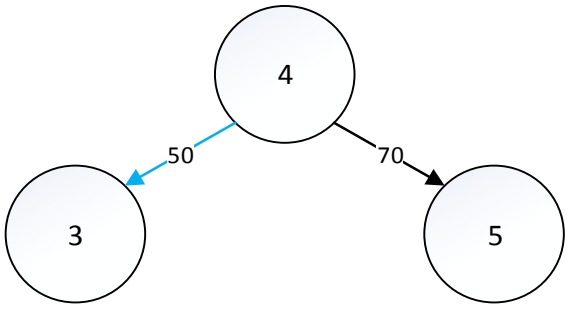
Sedangkan untuk perhitungan nilai g dilakukan dengan ketentuan, untuk pergerakan *horizontal* dan *vertical cost*nya adalah 10.

Dengan f(a) = h(a) = 50, karena g(n) = 0 biaya dari initial state ke initial state = 0, lalu masukkan suksesor ke dalam open list, disini suksesor hanya bisa digenerate 2 node, karena hanya tersedia 2 di open list. Ketika semua suksesor dari node awal sudah dibangkitkan ganti nilai f(A) dengan nilai f-cost terkecil dari nilai-nilai f-cost yang ada pada semua suksesornya, karena nilai f-cost yang terkecil adalah 50 yang sama dengan f(A) jadi tidak perlu diganti, lalu masukkan node awal ke closed list seperti terlihat pada gambar 3.2.

23	24	Tujuan 25	26	27	28	29
22	Penghalang	21	Penghalang	20	Penghalang	19
12	13	14	15	16	17	18
11	Penghalang	10	Penghalang	9	Penghalang	8
1	2	50 3 ← 10 40	50 4	70 5 10 60	6	7

Gambar 3.4 Langkah 2 Pencarian SMA*

Gambar 3.3 merupakan penjelasan langkah pertama pencarian jalur dari algoritma *astar*. Dibawah ini adalah penjelasan langkah pertama algoritma A *star* dengan menggunakan *tree.*

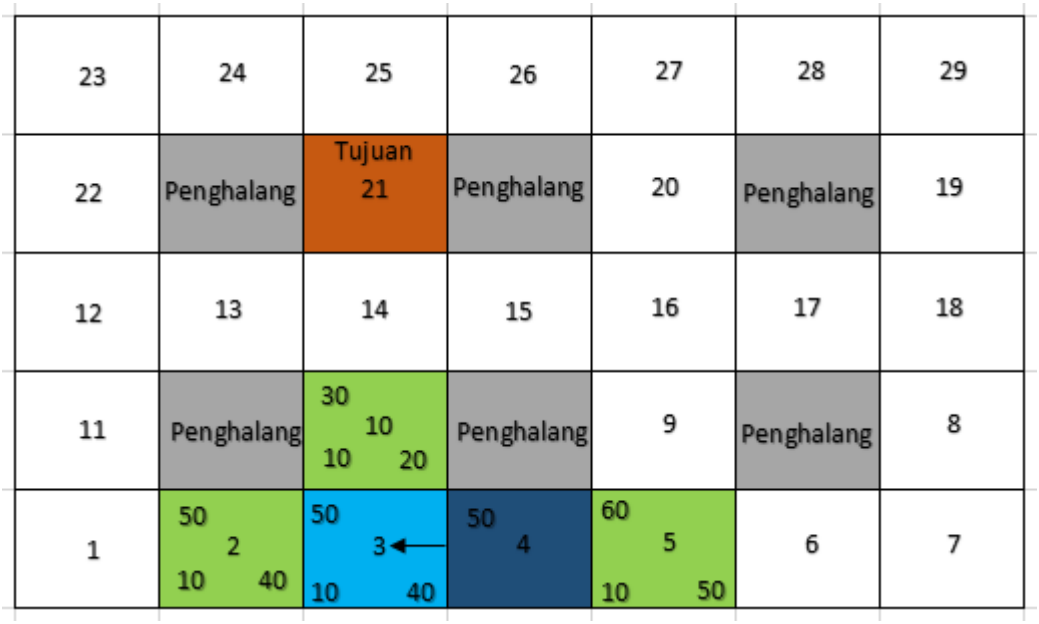


Gambar 3.5 Tree Langkah 2 Pencarian SMA*

Pada awalnya queue berisi initial state yang memiliki f-cost = h(n) karena g(n) = 0 (biaya dari initial state menuju ke nitial state adalah 0). Selantunya dilakukan iterasi sampai goal ditemukan atau sampai queue kosong. Awalnya, initial state akan terpilih sebagai n. Kemudian satu persatu, suksesornya dibangkitkan dan dihitung biayanya yang dilambangkan dengan f(s). Ketika semuanya suksesor dari n sudah dibangkitkan, maka ganti f-cost pada n dengan nilai f-cost terkecil dari nilai – nilai f-cost yang ada pada semua suksesornya. Pengganti ini dilakukan juga untuk semua ancestor-nya n (ayah, kakek, dst

keatas). Tetapi jika ancestor-nya n tersebut memiliki nilai f-cost yang lebih kecil dari pada f-cost baru pada n, maka nilai f-cost pada ancestor tersebut tidak perlu diganti. Selanjutnya, jika semua suksesor dari n sudah dimemori, maka keluarkan n dari queue, tetapi tidak dihapus secara fisik dari memori.

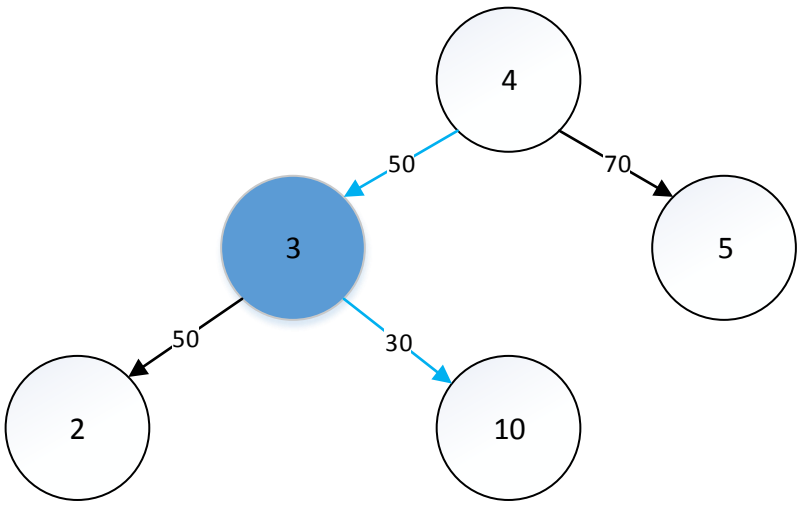
Langkah selanjutnya adalah pilih node dengan f-cost terkecil sebagai node yang sedang di cek, generate suksesor-nya dan hitung f-cost-nya juga, pada gambar 3.3 node yang terkecil adalah dengan nilai 30.



Gambar 3.6 Langkah 3 Pencarian SMA*

Karena dalam game tujuannya bergerak, tidak diam, maka tujuan dari node awal akan bergeser 1 point dari tempat awal.

Gambar 3.5 menunjukkan penjelasan melalui *tree* untuk pencarian jalan terpendek pada langkah kedua.



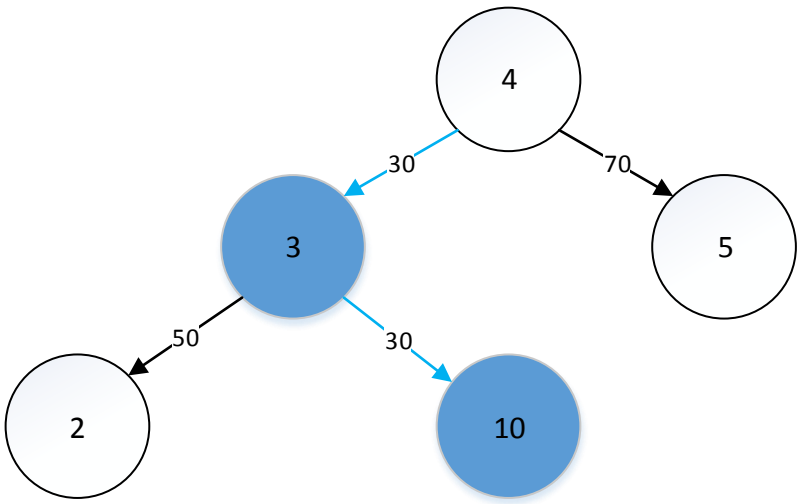
Gambar 3.7 Tree Langkah 3 Pencarian SMA*

Sama seperti langkah sebelumnya dipilih f-cost yang terkecil yaitu 50, lalu generate suksesor-nya di dapatkan 2 suksesor baru, dan f(s) yang terkecil adalah 30 sehingga f(best) dari 50 diganti dengan 30, begitu juga dengan ancestor-nya, seperti terlihat pada gambar 3.6.

23	24	25	26	27	28	29
22	Penghalang	Tujuan 21	Penghalang	20	Penghalang	19
12	13	14	15	16	17	18
11	Penghalang	30 10 10 20	Penghalang	9	Penghalang	8
1	50 2 10 40	30 3 10 40	30 4	50 5 10 40	6	7

Gambar 3.8 Langkah 4 Pencarian SMA*

Gambar 3.7 menunjukkan penjelasan melalui *tree* untuk pencarian jalan terpendek pada langkah ketiga.



Gambar 3.9 Tree Langkah 4 Pencarian SMA*

Node selanjutnya yang terpilih merupakan tujuan, sehingga didapatkan jalur terpendek dari node awal menuju ke tujuan, terlihat pada gambar 3.8.

23	24	25	26	27	28	29
22	Penghalang	21	Penghalang	20	Penghalang	19
12	13	Tujuan 14	15	16	17	18
11	Penghalang	30 10 10 20	Penghalang	9	Penghalang	8
1	50 2 10 40	30 3 10 40	30 4	50 5 10 40	6	7

Gambar 3.9 Langkah 5 Pencarian SMA*

3.1.5 Analisis Kebutuhan Non Fungsional

Analisis *non*-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan

ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan. Pada analisis kebutuhan sistem non-fungsional ini dijelaskan analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras dan analisis pengguna.

3.1.5.1 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan perangkat yang dibutuhkan untuk mengaktifkan dan mengoptimalkan kinerja perangat keras. Perangkat lunak yang dibutuhkan untuk membangun aplikasi *game* cerita rakyat fabel ini adalah sebagai berikut :

1. Sistem Operasi Windows 7.
2. Adobe Flash CS6.
3. Adobe Photoshop CS6
4. Java *Development Kit*
5. Flash Player

Perangkat lunak yang dibutuhkan pengguna untuk dapat menjalankan aplikasi ini di desktop adalah sebagai berikut :

1. Sistem Operasi Windows XP ke atas.
2. Adobe Flash Player.

Perangkat lunak *mobile* yang dibutuhkan pengguna untuk dapat menjalankan aplikasi ini adalah sebagai berikut :

1. Android Version 4.0 Ice Cream Sandwich ke atas
2. Adobe AIR for Android

Berdasarkan hasil analisis diatas dapat disimpulkan bahwa untuk membangun game cerita rakyat fabel membutuhkan beberapa perangkat lunak yang disebutkan diatas. Sedangkan untuk menjalankan game cerita rakyat fabel membutuhkan sistem operasi minimal minimal Windows XP untuk menjalankannya di dekstop. Sedangkan untuk menjalankan gam cerita rakyat fable di *mobile* membutuhkan minimal sistem operasi android versi 4.0 Ice Cream Sandwich dan adobe AIR.

3.1.5.2 Analisis Kebutuhan Perangkat Keras

Komputer terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi-instruksi kepada perangkat keras untuk melakukan suatu tugas tertentu, sehingga dapat menjalankan suatu sistem di dalamnya. Adapun perangkat keras yang digunakan untuk membangun aplikasi *game* ini adalah sebagai berikut :

Perangkat keras yang digunakan oleh pengembang untuk membangun aplikasi ini adalah sebagai berikut :

1. Processor Core 2 duo 2,40 Ghz
2. RAM : 2 GB.
3. VGA : 1 GB.
4. Harddisk 400 GB.
5. Monitor.
6. *Mouse* dan *Keyboard*.
7. *Speaker.*

Perangkat keras minimum yang dapat digunakan pengguna untuk dapat menjalankan aplikasi ini di desktop adalah :

3. Processor Pentium 4 2 Ghz
4. RAM 512
5. VGA On Board 128 Mb
6. Hardisk 40 Gb
7. *Monitor*
8. *Mouse* dan *Keyboard*
9. *Speaker*

Perangkat *mobile* yang dapat digunakan untuk memainkan game ini adalah :

1. Display 480 x 800 Pixels, 4.3 Inches (Multi Touch)
2. Memory Internal 4 Gb dan RAM 768 Gb
3. OS Android v4.0.4 *Ice Cream Sandwich*
4. CPU Dual – Core 1 GHz Cortex-A5GPU
5. GPU PowerVR SGX540

Kesimpulan yang dapat diambil dari analisis kebutuhan perangkat keras ini adalah, game dapat dimainkan di desktop walaupun dengan spesifikasi rendah, sedangkan di*mobile*, *game* ini dapat di mainkan dengan spesifiikasi mobile diatas, namun tidak seresponsif di desktop.

3.1.5.3 Analisis Pengguna Sistem (*User*)

Pada analisis *user* (pemakai) ini akan mencakup analisis beberapa parameter terhadap calon *user* dari aplikasi.

a. User Knowledge Experience

Game cerita rakyat fabel ini diperuntukan kepada anak berusia 7 – 12 tahun. *Game* ini dirancang sederhana agar anak yang memainkan permainan ini dapat dengan mudah memaminkannya, tetapi anak yang pernah memainkan permainan di*mobile* akan lebih mudah lagi memainkan permainan ini. Tabel 3 berikut ini akan menjelaskan analisis klasifikasi *knowledge and experience* dari pengguna aplikasi.

Tabel 3.3 Analisis Klasifikasi *Knowledge and Experience*

Karakteristik *Player*	**Keterangan**
Computer Literacy	Dapat menggunakan *Smartphone*
Age	7 – 12 tahun
Application Experience	Tidak dibutuhkan
Gender	Pria dan wanita
Native Language	Menggunakan bahasa inggris dalam navigasinya

b. User Physical Characteristic

Keadaan fisik seseorang mungkin akan berpengaruh pada penggunaan aplikasi *game* ini. Ada hal-hal yang harus diperhatikan juga terhadap *user* dari karakteristik fisiknya untuk dapat menggunakan aplikasi ini yaitu, *Color Blind*, *Handednes* dan *Gender* seperti terlihat pada tabel 3.3.

Tabel 3.4 Analisis *Users Physical Characteristic*

Usia	7 – 12 tahun
Jenis Kelamin	Pria dan Wanita

Tangan	Kanan dan Kiri
Buta Warna	*User* yang tidak bisa membedakan warna yang satu dengan yang lainnya (buta warna) masih mampu menggunakan aplikasi ini, karena tidak ada indikator warna-warna khusus yang membedakan antara fungsional yang satu dengan fungsional yang lainnya. Akan tetapi penggunaannya tidak akan optimal karena dalam *game* ini terdapat banyak sekali perbedaan warna yang menunjang interaksi dan ketertarikan dalam permainan.

3.1.6 Analisis Kebutuhan Fungsional

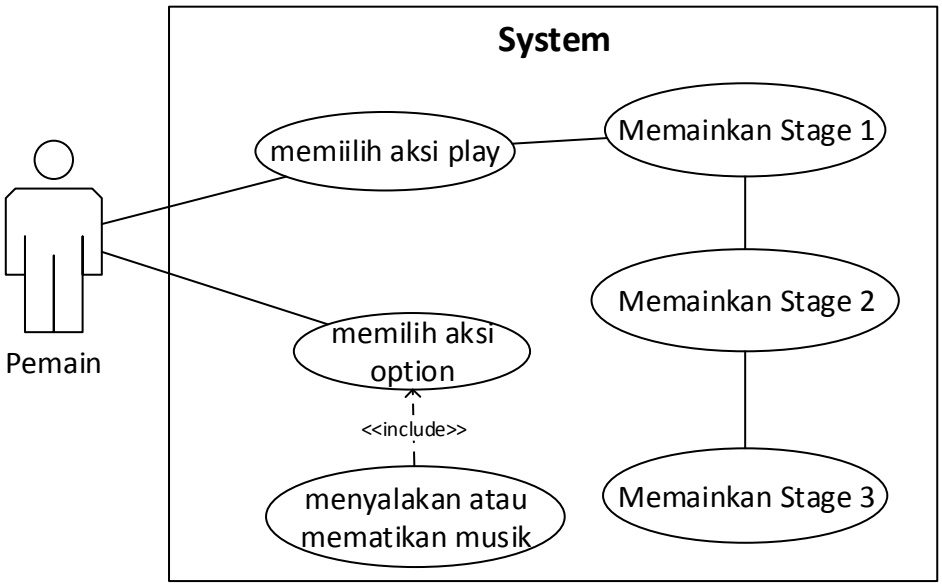
Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik sesuai kebutuhan.

3.1.6.1 Identifikasi Aktor

Tahap pertama yang dilakukan dalam melakukan analisis berorientasi objek menggunakan UML adalah menentukan aktor atau pengguna sistem. Kata aktor dalam konteks UML, menampilkan peran (*roles*) yang pengguna (atau sesuatu di luar sistem yang dikembangkan yang dapat berupa perangkat keras, end *user*, sistem yang lain, dan sebagainya).

3.1.6.2 Use Case Diagram

Use case Diagram merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antar aktor dengan aktivitas yang terdapat pada sistem. Sasaran pemodelan *use case* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan sistem yang akan dibangun. Dari hasil analisis aplikasi yang ada maka *usecase diagram* untuk *game* cerita rakyat fabel dapat dilihat pada gambar 3.2 berikut:



Gambar 3.10 *Use Case Diagram* [18]

3.1.6.3 Definisi *Actor*

Definisi *Actor* berfungsi untuk menjelaskan *Actor* yang terdapat pada *Use Case Diagram*. Definisi *Actor* diterangkan pada Tabel 3.5 berikut :

Tabel 3.5 Definisi *Actor*

Actor	**Deskripsi**
Pemain	Orang yang akan memainkan *Game.*

3.1.6.4 Definisi *Use Case*

Definisi *Use Case* berfungsi untuk menjelaskan proses yang terdapat pada setiap *Use Case*.

Tabel 3.6 Definisi *Use Case*

No	***Use Case***	**Deskripsi**
1	Play	Untuk memainkan permainan.
2	*Stage* 1	Untuk menampilkan permainan *Stage* 1
3	*Stage* 2	Untuk menampilkan permainan *Stage* 2
4	*Stage* 3	Untuk menampilkan permainan *Stage* 3
5	Option	Untuk menampilkan option.

3.1.6.5 *Use Case* Skenario

Skenario *Use Case* menggambarkan alur penggunaan sistem dimana setiap skenario digambarkan dari sudut pandang aktor, seseorang atau piranti yang berinteraksi dengan perangkat lunak dalam berbagai cara **[18]**.

a. Use Case Skenario Play

Tabel 3.7 Skenario Play

Identifikasi
Nama Use Case : *Play*
Deskripsi : Proses untuk menampilkan permainan
Aktor : Pemain
Kondisi Awal : Pemain berada di menu utama permainan
Pemicu : Pemain memilih menu *play* dari menu utama
Skenario Utama : 1. Pemain memulai permainan. 2. Pemain menyentuh tombol play. 3. Sistem menampilkan slide cerita awal kancil dan buaya. 4. Sistem menampilkan petunjuk permainan stage 1. 5. Pemain menyentuh tombol main 6. Sistem menuju ke permainan stage 1.
Skenario Alternatif : Tidak ada

b. Use Case Skenario Stage 1

Tabel 3.8 Skenario *Stage* 1

Identifikasi
Nama Use Case : *Stage* 1
Deskripsi : Proses untuk menampilkan permainan *Stage* 2

Aktor : Pemain
Kondisi Awal : Pemain berada pada petunjuk permainan *Stage* 1
Pemicu : Pemain memilih menu main dari petunjuk permainan *Stage* 1
Skenario Utama : 1. Pemain membaca petunjuk permainan dan menyentuh tombol main. 2. Sistem mengeluarkan *sound*, menginisialisasi posisi buaya, dan parameter waktu dimulai. 3. Pemain menyentuh tombol pause. 4. Sistem merespon dengan memangil fungsi pause dan menampilkan menu pause. 5. Pemain menyentuh tombol off pada musik. 6. Sistem merespon dengan terjadinya aksi *mute*. 7. Pemain menyentuh tombol main menu, maka pemain akan keluar dari permainan stage 1 dan menuju menu utama. 8. Pemain mengarhkan lemparan batu dan menentukan kekuatan lemparan. 9. Pemain menekan tombol tembak. 10. Sistem merespon dengan terjadinya lemparan batu ke arah yang ditentukan oleh pemain dan dengan kekuatan yang telah ditentukan oleh pemain. 11. Lemparan mengenai buaya maka buaya akan langsung mati. 12. Sistem menambahkan point untuk pemain. 13. Apabila poin pemain mencapain poin yang telah ditentukan maka pemain dapat melanjutkan ke *stage* berikutnya.
Skenario Alternatif : 3a. Pemain tidak menyentuh tombol *pause*. 4a. Sistem tidak menampilkan menu *pause*. 5a. Pemain tidak menyentuh tombol off pada musik. 6a. Tidak terjadinya aksi mute.

7a. Pemain tidak menyentuh tombol main menu. 8a. Pemain tidak mengarahkan lemparan batu dan tidak menentukan kekuatan lemparan. 11a. Lemparan tidak mengenai buaya, dan buaya tidak mati. 12a. Sistem tidak menambahkan poin untuk pemain. 13a. Point pemain tidak mencapai poin yang telah ditentukan maka pemain tidak berhak menuju ke *Stage* berikutnya.
Skenario Pengecualian : 1. Game mengalami *Crash*.

c. Use Case Skenario *Stage* 2

Tabel 3.9 Skenario *Stage* 2

Identifikasi
Nama Use Case : *Stage* 2
Deskripsi : Proses untuk menampilkan permainan *Stage* 3
Aktor : Pemain
Kondisi Awal : Pemain berada pada petunjuk permainan *Stage* 2
Pemicu : Pemain memilih menu *next Stage* dari pesan menang di *Stage* 1
Skenario Utama : 1. Pemain membaca petunjuk permainan dan menyentuh tombol main. 2. Sistem mengeluarkan *sound*, kancil berada pada posisi awal, dan parameter waktu dimulai. 3. Pemain menyentuh tombol pause. 4. Sistem merespon dengan memangil fungsi pause dan menampilkan menu pause. 5. Pemain menyentuh tombol off pada musik. 6. Sistem merespon dengan terjadinya aksi *mute*. 7. Pemain menyentuh tombol main menu, maka pemain akan keluar dari permainan stage 1 dan menuju menu utama.

8. Menampilkan buaya. 9. Menampilkan buah. 10. Kancil bergerak ke arah yang ditentukan oleh pemain menggunakan panel. 11. Pemain menyentuh tombol makan dan sistem merespon dengan bertambahnya poin pemain. 12. Apabila poin pemain mencapain poin yang telah ditentukan maka pemain dapat melanjutkan ke *Stage* berikutnya.
Skenario Alternatif : 3a. Pemain tidak menyentuh tombol *pause*. 4a. Sistem tidak menampilkan menu *pause*. 5a. Pemain tidak menyentuh tombol off pada musik. 6a. Tidak terjadinya aksi mute. 7a. Pemain tidak menyentuh tombol main menu 11a. Pemain tidak berhasil memakan buah dan system tidak menambahkan point pemain. 12a. Poin pemain tidak mencapai poin yang telah ditentukan dan pemain tidak berhak menuju ke stage berikutnya.
Skenario Pengecualian : 1. Game mengalami *Crash*.

d. Use Case Skenario *Stage* 3

Tabel 3.10 Skenario *Stage* 3

Identifikasi
Nama Use Case : *Stage* 3
Deskripsi : Proses untuk memenangkan game
Aktor : Pemain
Kondisi Awal : Pemain berada pada petunjuk permainan *Stage* 3
Pemicu : Pemain memilih menu *next Stage* dari pesan menang di *Stage* 2
Skenario Utama : 1. Pemain membaca petunjuk permainan dan menyentuh tombol main. 2. Sistem mengeluarkan *sound*, kancil berada pada posisi awal, dan parameter waktu dimulai. 3. Pemain menyentuh tombol pause. 4. Sistem merespon dengan memangil fungsi pause dan menampilkan menu pause. 5. Pemain menyentuh tombol off pada musik. 6. Sistem merespon dengan terjadinya aksi *mute*. 7. Pemain menyentuh tombol main menu, maka pemain akan keluar dari permainan stage 1 dan menuju menu utama. 8. Menampilkan kancil pada posisi awal. 9. Menampilkan kayu. 10. Kancil bergerak ke arah yang ditentukan oleh pemain menggunakan panel. 11. Apabila kancil menabrak kayu, maka kancil akan terjatuh dan sistem akan menampilkan menu kalah. 12. Apabila kancil menginjak batu apung, maka sistem akan menampilkan menu kalah. 13. Munculnya buaya

14. Buaya mengejar kancil dan membenturkan dirinya ke batu agar kancil jatuh. 15. Kancil berhasil ke sebrang sungai, dan sistem menampikan slide bahwa kancil telah berhasil menyebrangi sungai.
Skenario Alternatif : 3a. Pemain tidak menyentuh tombol *pause*. 4a. Sistem tidak menampilkan menu *pause*. 5a. Pemain tidak menyentuh tombol off pada musik. 6a. Tidak terjadinya aksi mute. 7a. Pemain tidak menyentuh tombol main menu 11a. Kancil tidak terjatuh. 12a. Kancil tidak menginjak batu apung. 14a. Kancil tidak mencapai sebrang dan system menampilkan menu kalah.
Skenario Pengecualian : 1. Game mengalami *Crash*.

e. Use Case Skenario Option

Tabel 3.11 Skenario *Option*

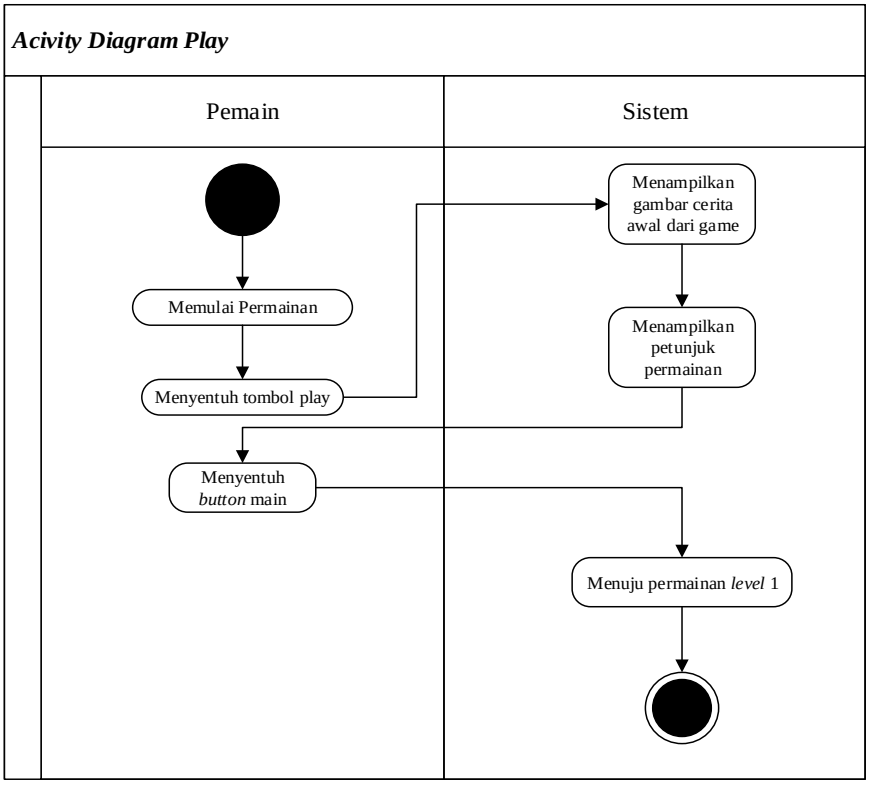
Identifikasi
Nama Use Case : *Option*
Deskripsi : Proses untuk menampilkan menu musik dan menu *developer*.
Aktor : Pemain
Kondisi Awal : Pemain berada pada menu utama.
Pemicu : Pemain memilih menu *option*
Skenario Utama : 1. Pemain menyentuh tombol option. 2. Sistem menampilkan submenu musik dan developer. 3. Pemain menekan *off* pada musik

4. Sistem menampilkan menu *developer* yang ber isi tentang pembuat game.
Skenario Alternatif : 3a. Pemain tidak menyentuh tombol *off* musik pada menu musik.

3.1.6.6 *Activity Diagram*

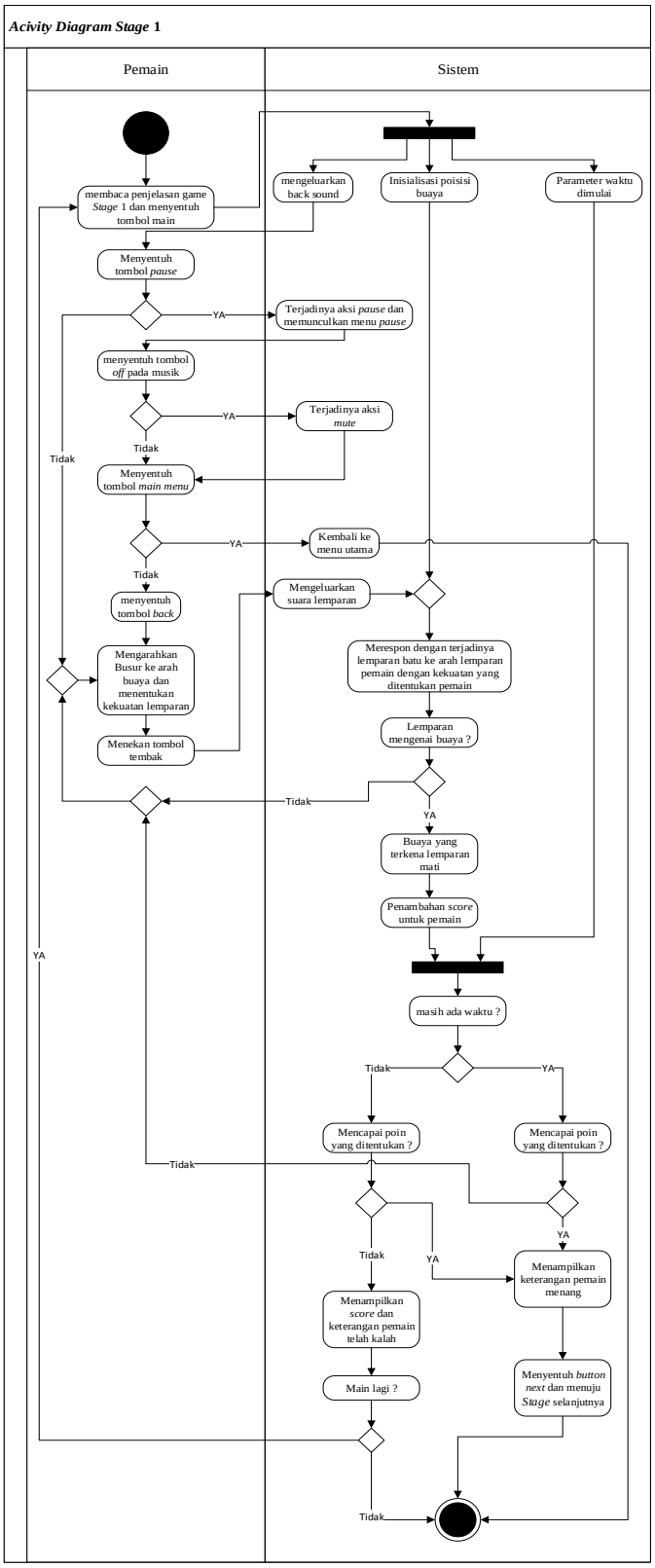
Activity diagram menggambarkan berbagai aliran aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour* internal sebuah sistem dan interaksi antar subsistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktifitas dari *Stage* atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *Use Case* atau lebih. Aktifitas menggambarkan proses yang berjalan, sementara *Use Case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktifitas.

a. Activity Diagram Play



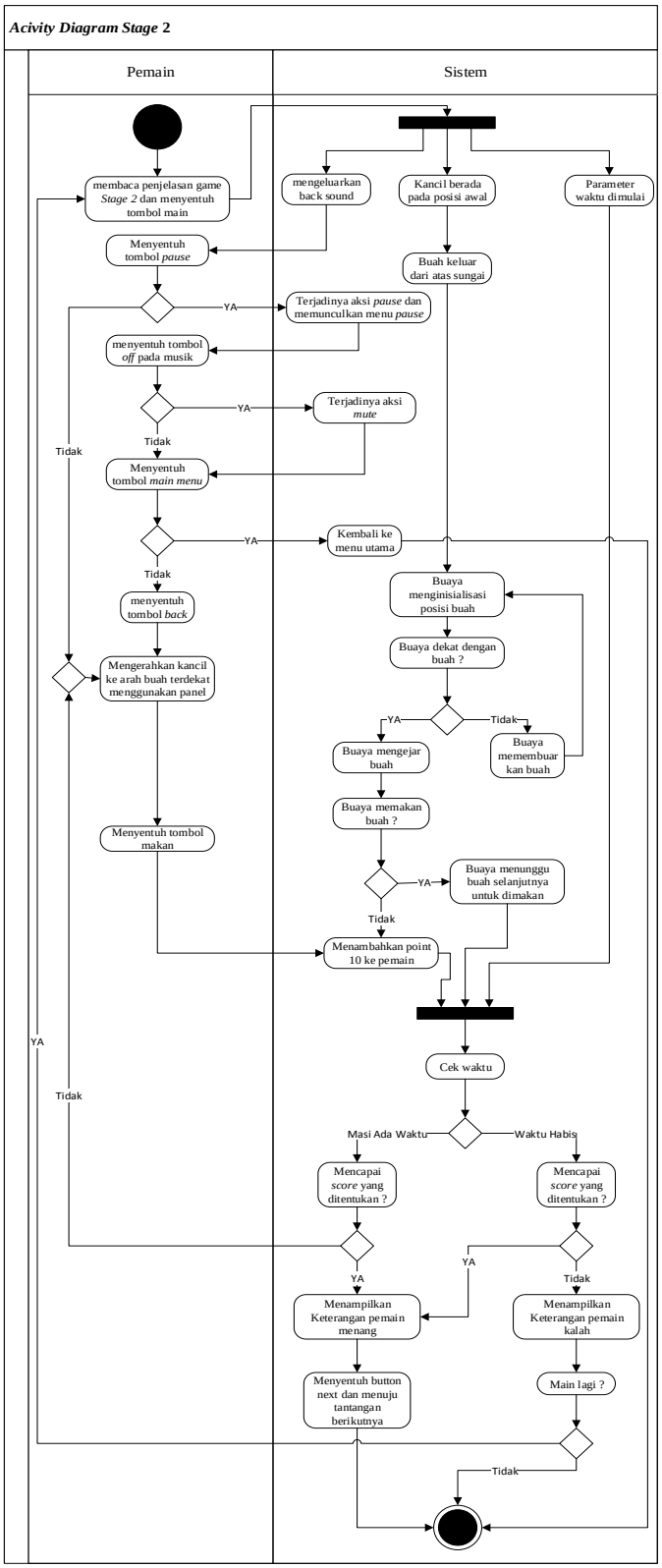
Gambar 3.11 Activity Diagram Play

b. Activity Diagram Stage 1



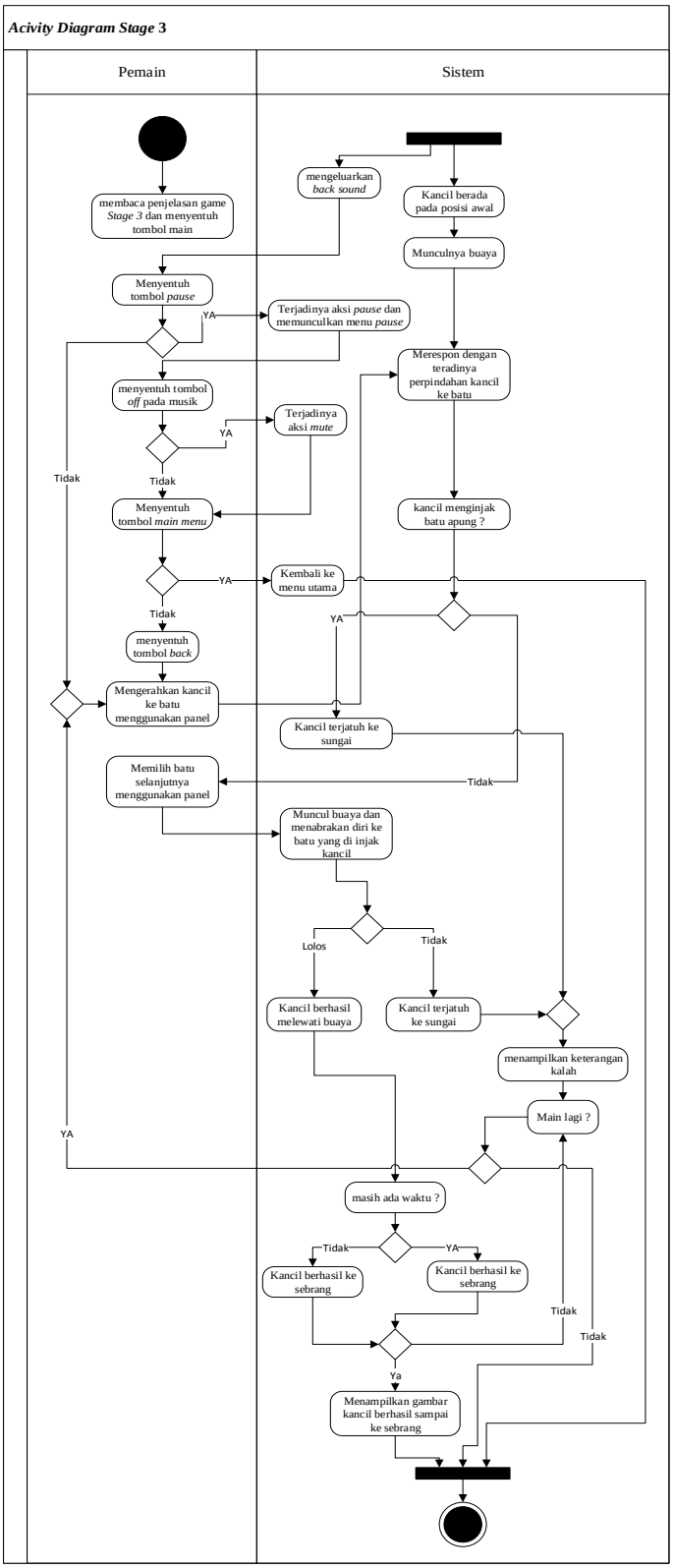
Gambar 3.12 *Activity Diagram Stage* 1

c. Activity Diagram Stage 2



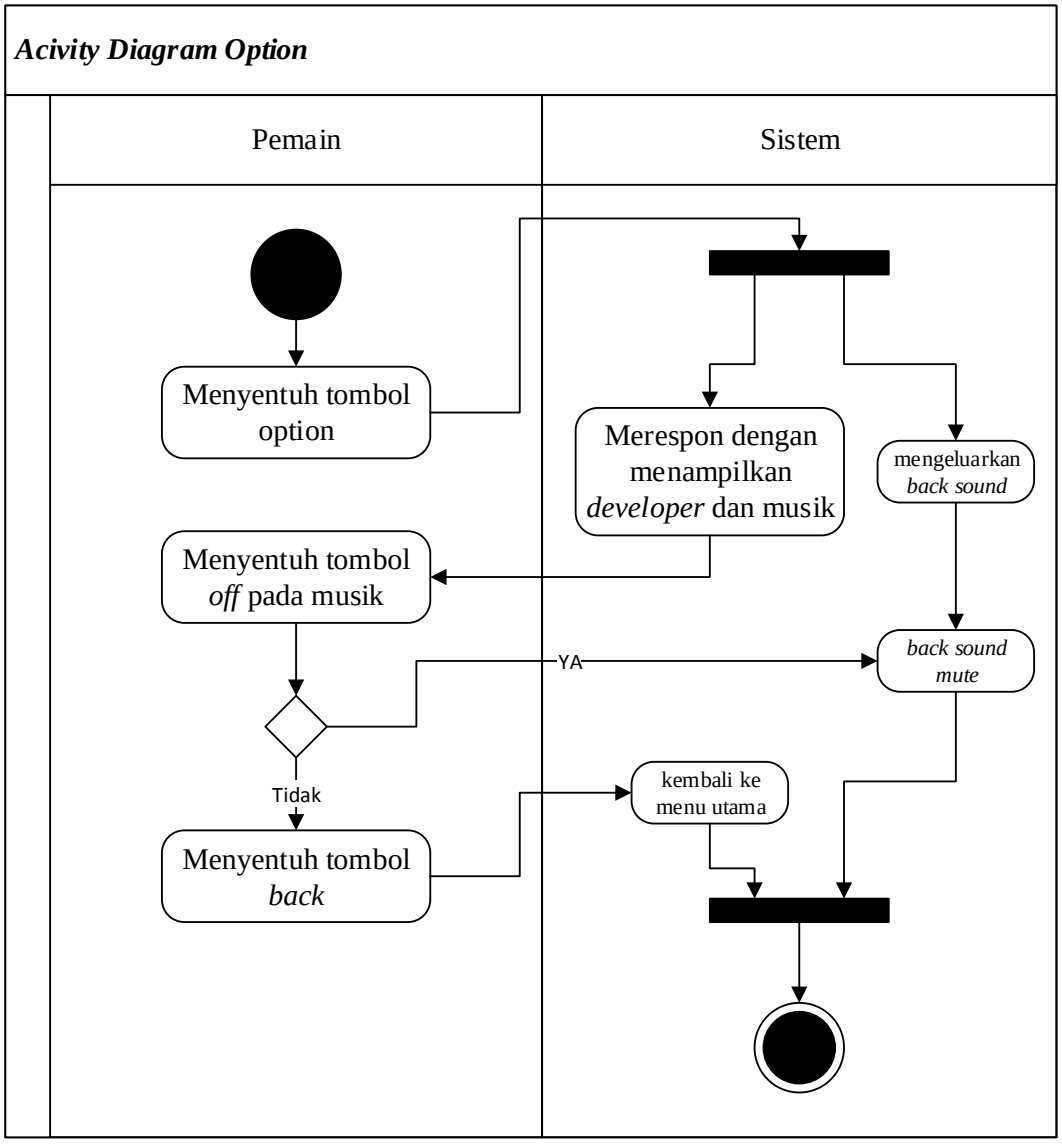
Gambar 3.13 *Activity Diagram Stage* 2

d. Activity Diagram Stage 3



Gambar 3.13 *Activity Diagram Stage* 3

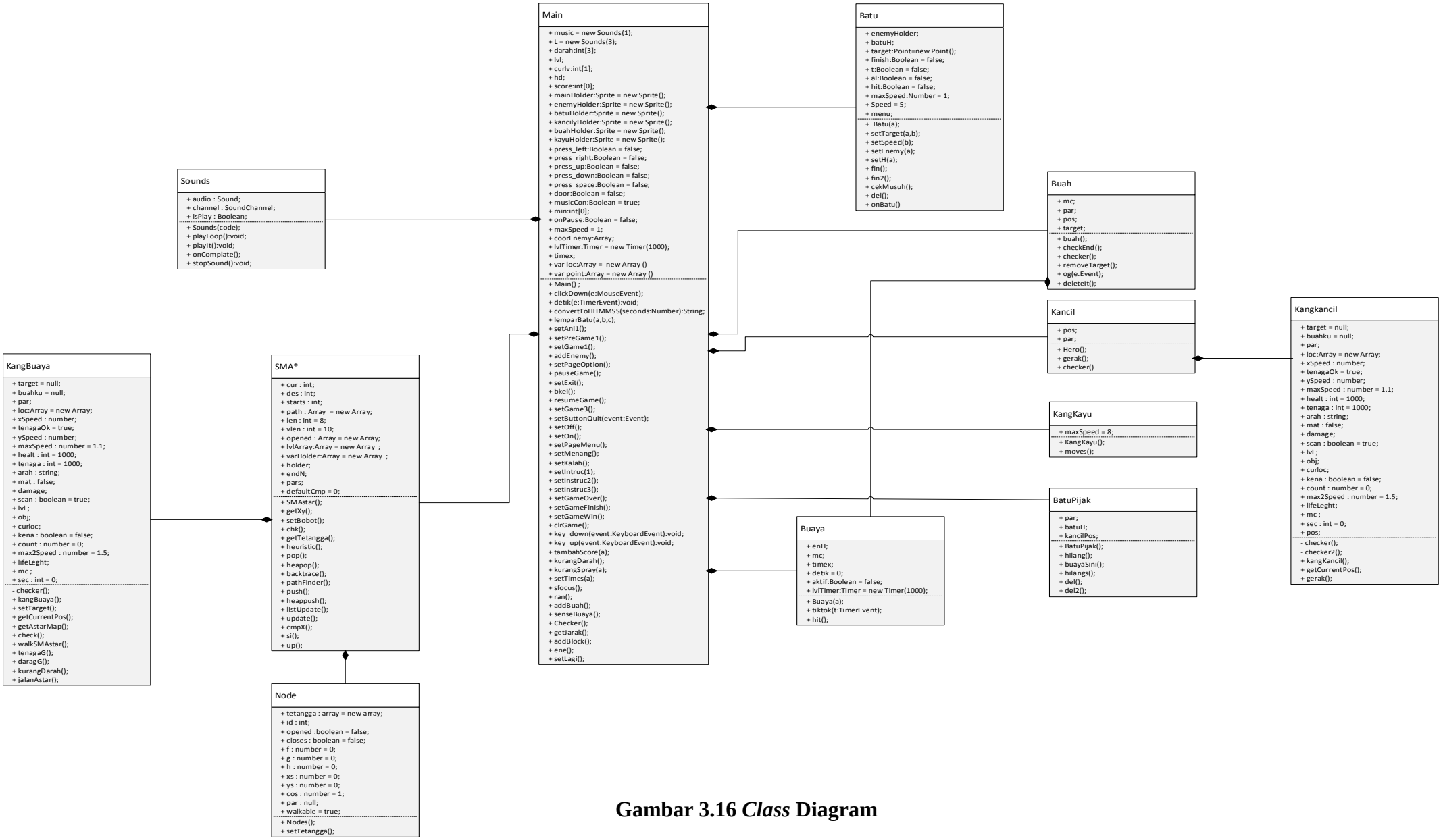
e. Activity Diagram Option



Gambar 3.15 ***Activity Diagram Option***

3.1.6.7 *Class Diagram*

Class Diagram menunjukkan kelas dan bagaimana mereka dapat dipasang bersama-sama. Perancangan class diagram pada game ini dapat dilihat pada gambar 3.13 berikut.



Gambar 3.16 *Class* Diagram

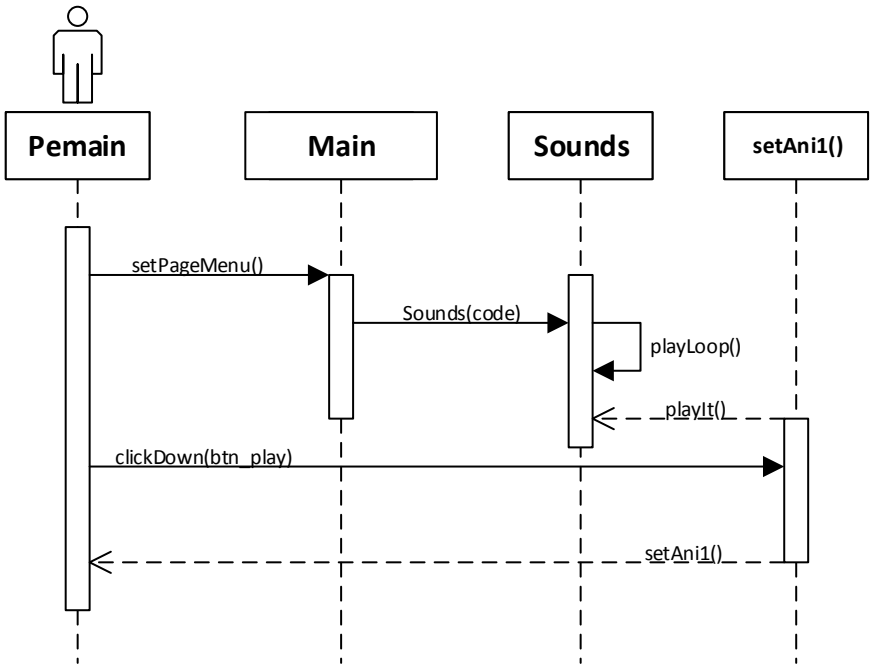
3.1.6.8 Sequence Diagram

Sequence diagram biasa digunakan untuk menggambarakan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Berikut ini beberapa *sequencial diagram* yang terdapat pada *game* cerita rakyat fabel yaitu :

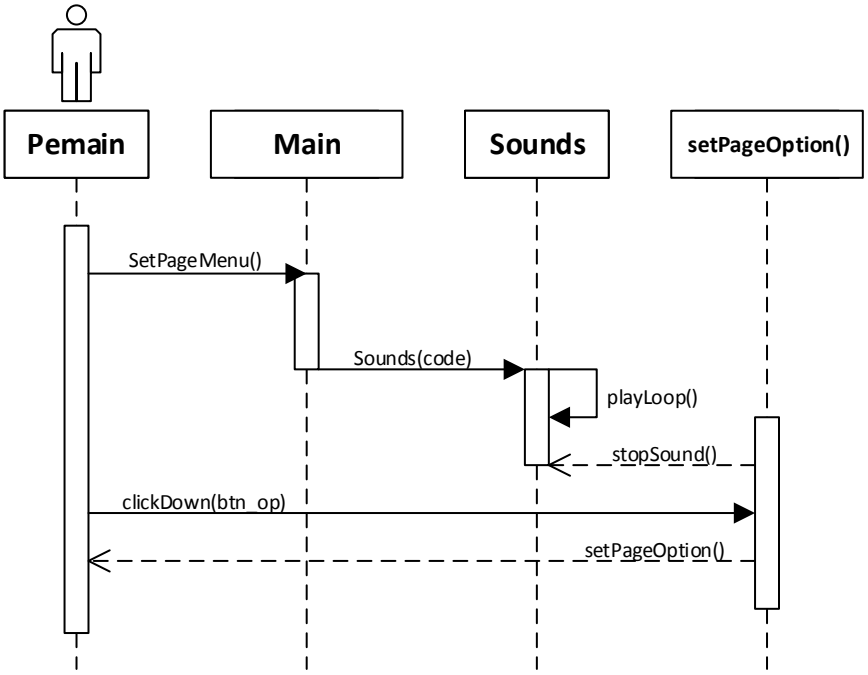
a. Sequence Diagram Play

Diagram ini menjelaskan interaksi antar partisipan di dalam aktivitas *play game* dalam urutan waktu. Untuk lebih jelasnya lihat gambar 3.14.



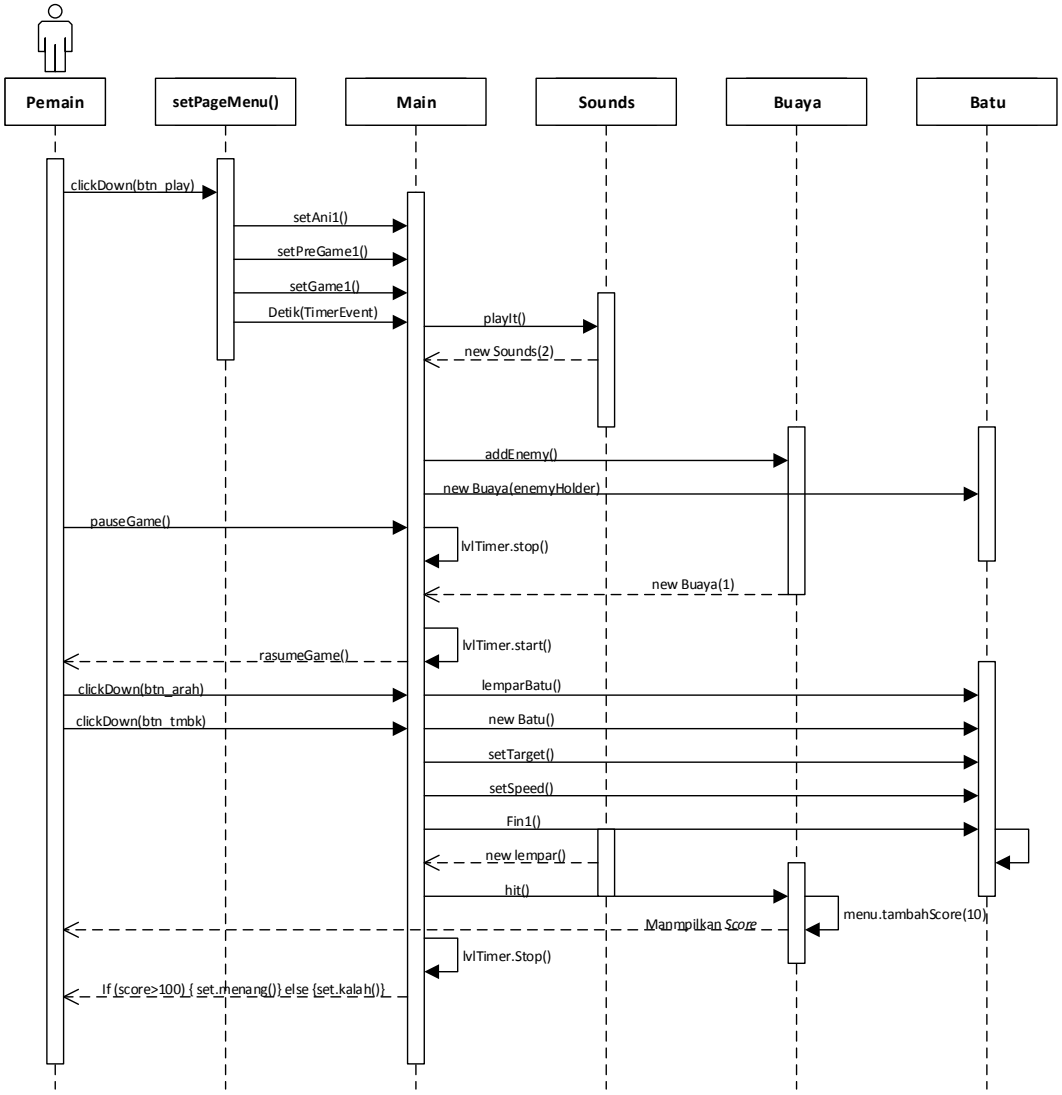
Gambar 3.17 *Sequence Diagram Play*

b. Sequence Diagram Option



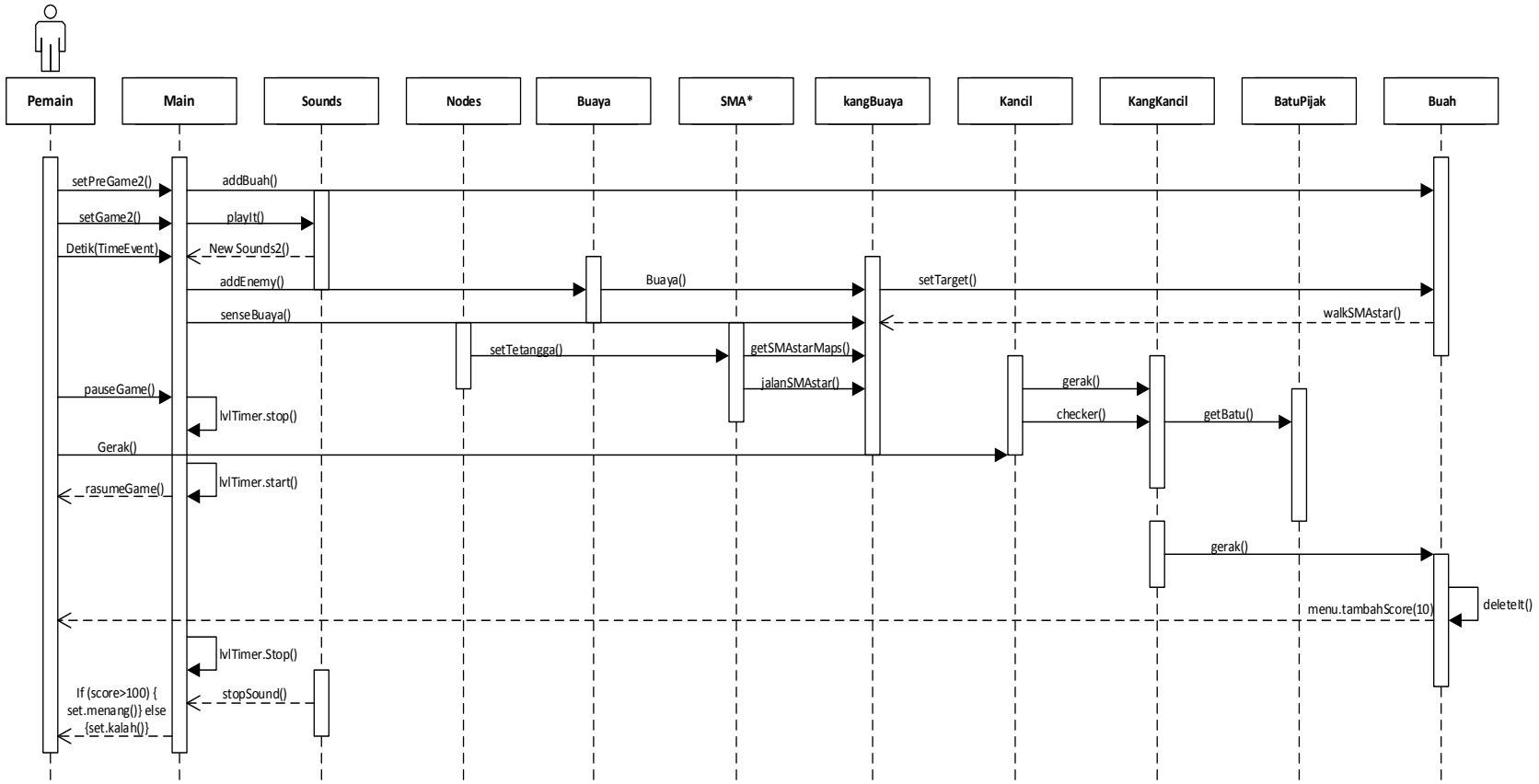
Gambar 3.18 ***Sequence Diagram Option***

c. Sequence Diagram Stage 1



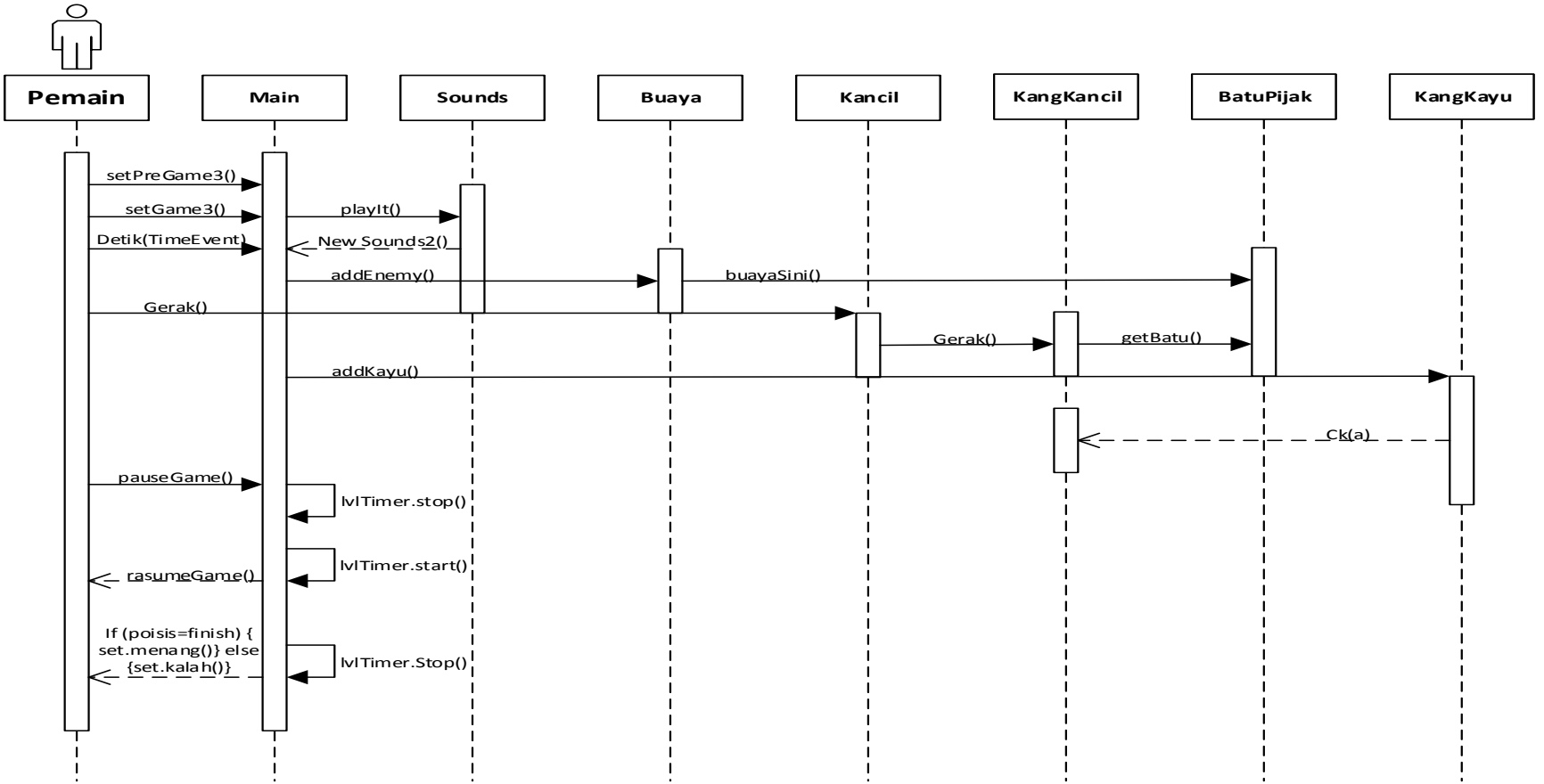
Gambar 3.19 *Sequence Diagram Stage* 1

d. Sequence Diagram Stage 2



Gambar 3.20 *Sequence Diagram Stage* 2

e. Sequence Diagram Stage 3



Gambar 3.21 *Sequence Diagram Stage* 3

3.2 Perancangan Sistem

Perancangan adalah suatu bagian dari metodologi pengembangan pembangunan suatu perangkat lunak yang dilakukan setelah tahapan untuk memberikan gambaran secara terperinci. Berdasarkan uraian diatas perancangan sistem merupakan tahapan dari siklus pengembangan sistem yang didefinisikan dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi yang menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perancangan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satukesatuan yang utuh dan berfungsi, juga menyangkut konfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem.

3.2.1 Perancangan Komponen Permainan

Di dalam bagian ini akan dijelaskan karakter yang terdapat pada game dan storyboard game cerita rakyat fabel.

3.2.1.1 Pengenalan Karakter

Karakter utama dalam *game* ini adalah kancil, kancil merupakan hewan yang dikenal cerdas oleh hewan lainnya dihutan. Dalam permainan ini, karakter kancil akan mulai digunakan di Stage ke 2 dan ke 3. Bentuk karakter dapat dilihat pada gambar 3.19.

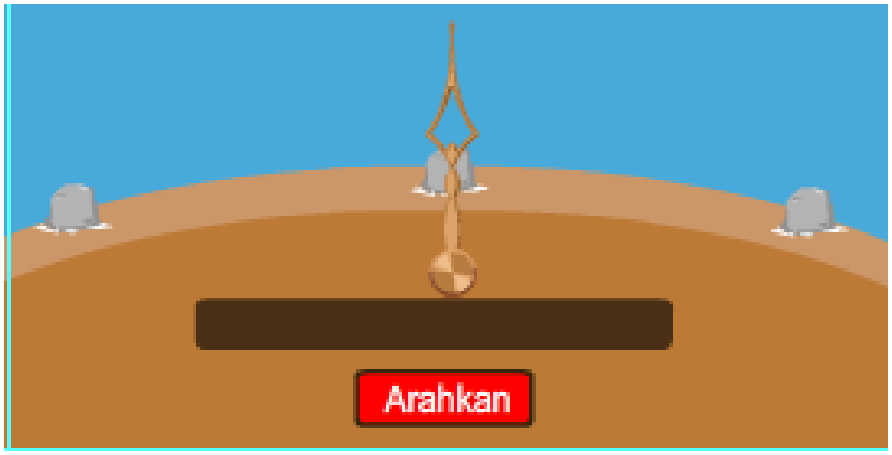


Gambar 3.22 Karakter Utama

Tabel 3.12 Detail Karakter

Nama	Kancil
Kemampuan	melompat, melempar batu, mengambil buah
Senjata	Batu

Untuk melawan buaya pada Stage 1, kancil harus melepari para buaya dengan batu, dalam pelemparannya akan ada jarum sebagai penunjuk arah yang disebut sebagai busur, dapat dilihan pada gambar 3.20.

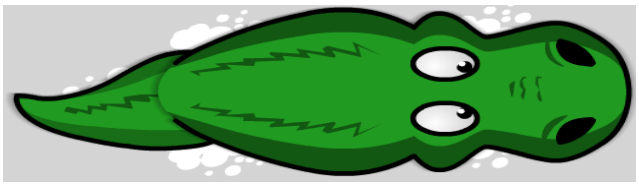


Gambar 3.23 Busur

Tabel 3.13 Detail Busur

Nama	Busur
Serangan	Buaya akan langsung mati apabila terkena lemparan batu.
Peluru	Batu

Musuh yang terdapat pada *game* ini adalah buaya, buaya ini memiliki tingkat kesulitan berbeda-beda untuk dikalahkan. Musuh dapat dilihat pada Gambar 3.21.



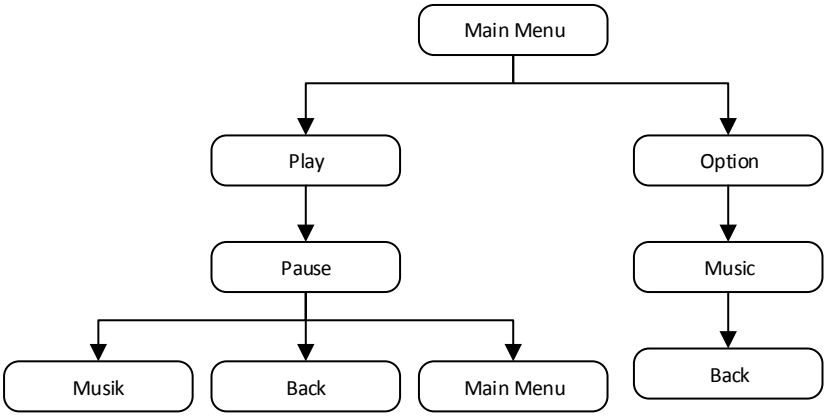
Gambar 3.24 Buaya

3.2.1.2 Perancangan Arsitektur

Setelah melakukan perancangan data pada sistem yang dibangun, maka dilakukanlah perancangan arsitektur. Perancangan arsitektur yang telah dibuat meliputi beberapa perancangan diantaranya struktur menu, perancangan antarmuka.

3.2.1.3 Perancangan Struktur Menu

Perancangan menu dibuat agar pengguna dapat menggunakan aplikasi Game dengan mudah, sehingga pengguna dapat dengan mudah memilih menu aplikasi yang sedang berjalan. Stuktur menu dapat dilihat pada gambar 3.21 berikut.



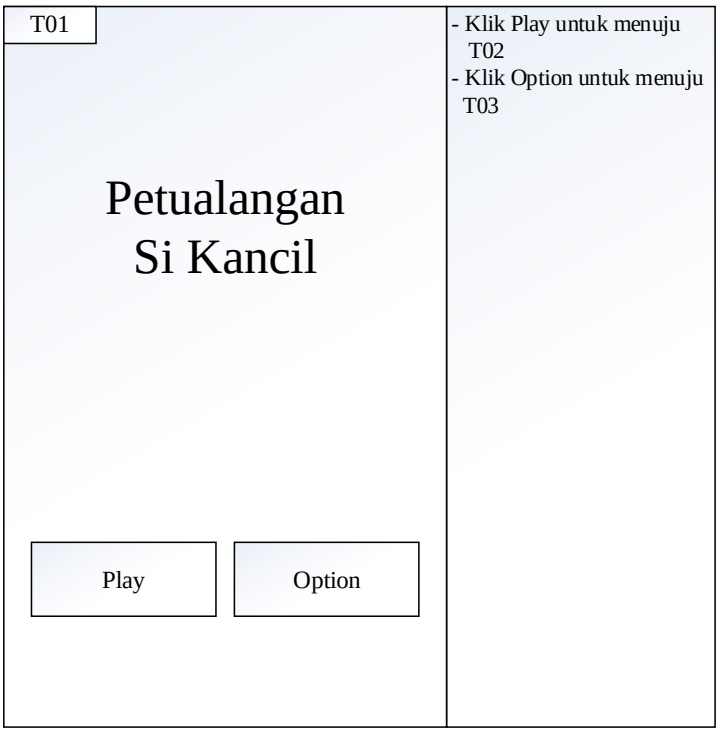
Gambar 3.25 Struktur Menu

3.2.1.4 Perancangan Antar Muka

Setelah melakukan perancangan data pada sistem yang dibangun, maka dilakukanlah perancangan arsitektur. Perancangan arsitektur yang telah dibuat meliputi beberapa perancangan diantaranya struktur menu, perancangan antarmuka.

a. Antarmuka Menu Utama

Perancangan antarmuka menu utama dari *game* cerita rakyat fabel dapat dilihat pada gambar 3.23.

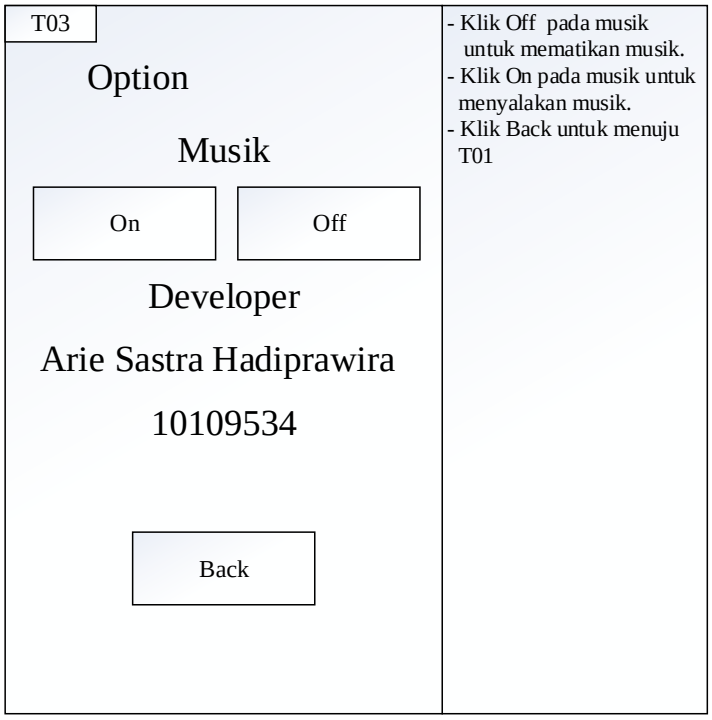


Gambar 3.26 Antarmuka Tampilan Menu Utama

Tabel 3.14 Penggunaan Tombol Menu Utama

No	Jenis Objek	Nama Objek	Keterangan
1	Button	Play	Untuk masuk memulai permainan
		Option	Untuk menampilkan halaman option

b. Antarmuka Option

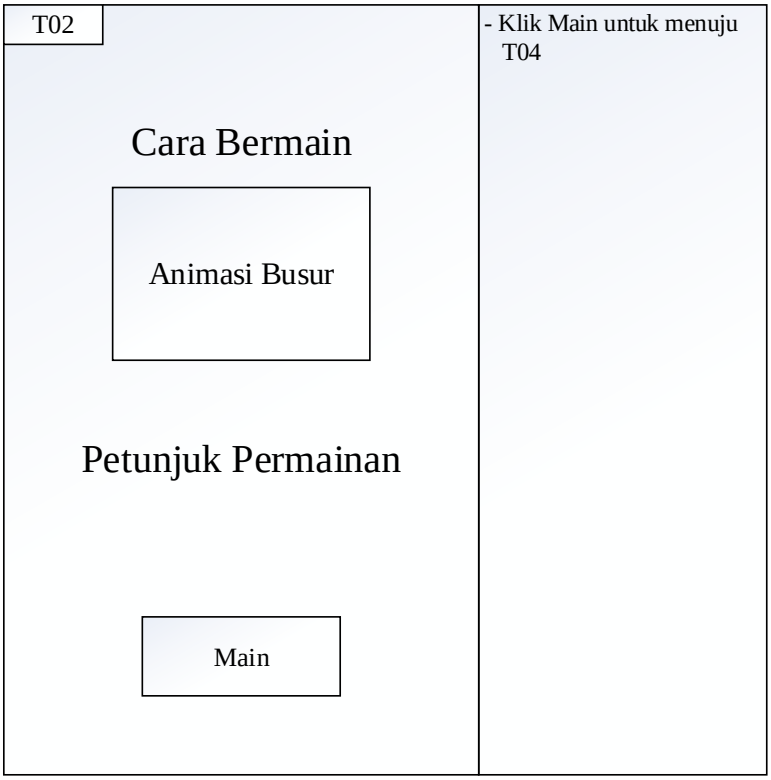


Gambar 3.26 Antarmuka Tampilan Menu Option

Tabel 3.15 Penggunaan Tombol Menu Option

No	Jenis Objek	Nama Objek	Keterangan
1	Button	On	Untuk menyalakan musik
		Off	Untuk mematika musik
		Back	Untuk kembali ke menu utama

c. Antarmuka Petunjuk

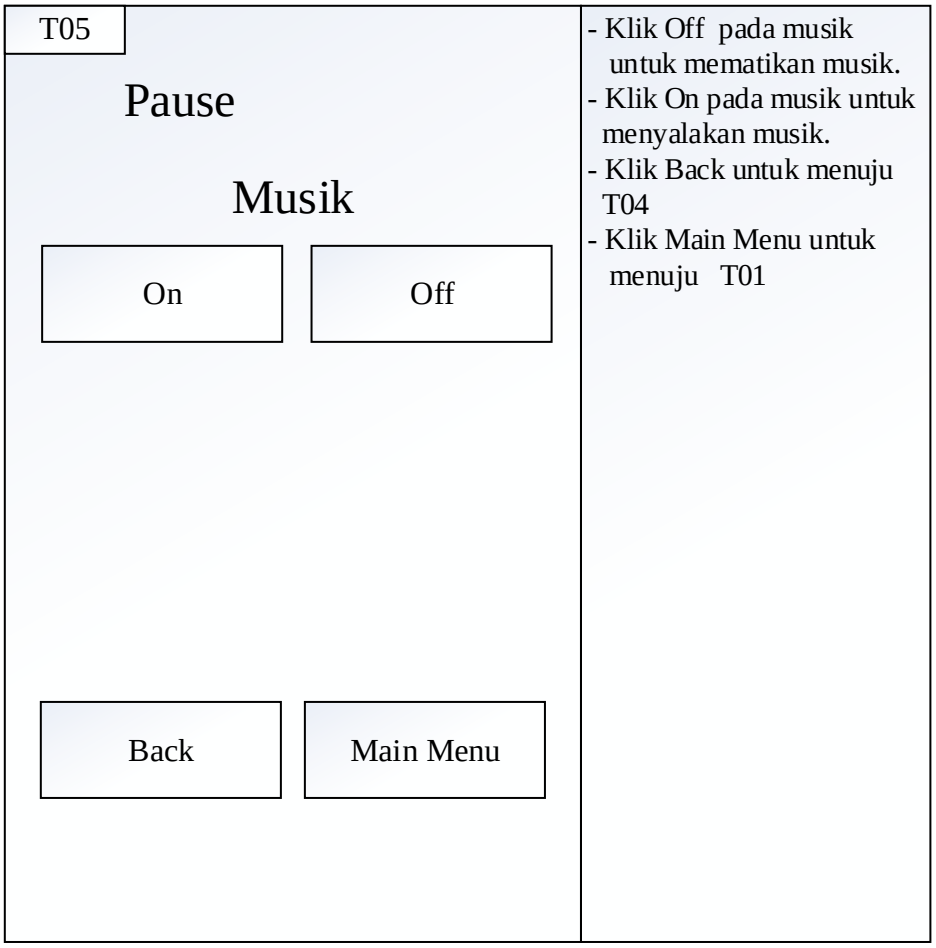


Gambar 3.28 Antarmuka Tampilan Petunjuk *Game*

Tabel 3.16 Penggunaan Tombol Menu Petunjuk

No	Jenis Objek	Nama Objek	Keterangan
1	Button	Main	Untuk menuju ke permainan

d. Antarmuka Pause

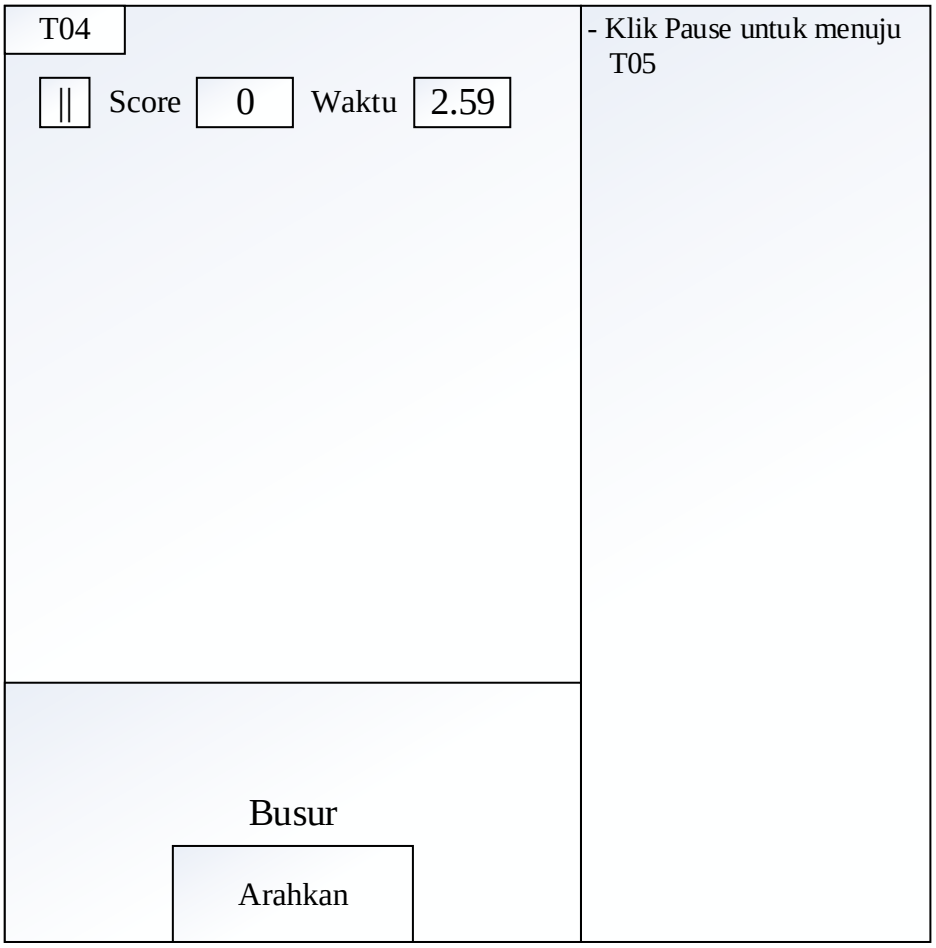


Gambar 3.29 Antarmuka Tampilan Pause

Tabel 3.17 Penggunaan Tombol Menu Pause

No	Jenis Objek	Nama Objek	Keterangan
1	Button	On	Untuk menyalakan musik
		Off	Untuk mematika musik
		Back	Untuk kembali ke permainan
		Main Menu	Untuk kembali ke menu utama

e. Antarmuka Permainan *Stage* 1

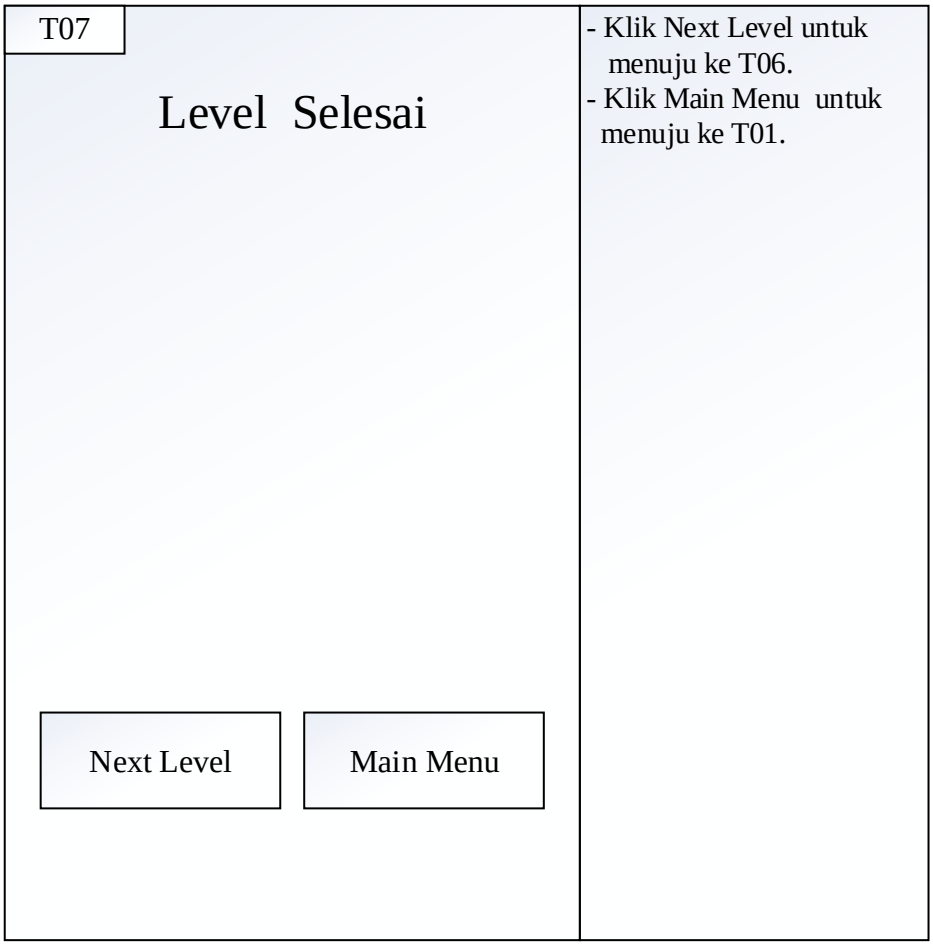


Gambar 3.29 Antarmuka Tampilan *Stage* 1

Tabel 3.18 Penggunaan Tombol Menu *Stage* 1

No	Jenis Objek	Nama Objek	Keterangan
1	Button	||	Untuk mem*pause* permainan
		Arahkan	Untuk mengarahkan lemparan
		Tembak	Untuk menebakkan batu

f. Antarmuka Permainan Keterangan Menang

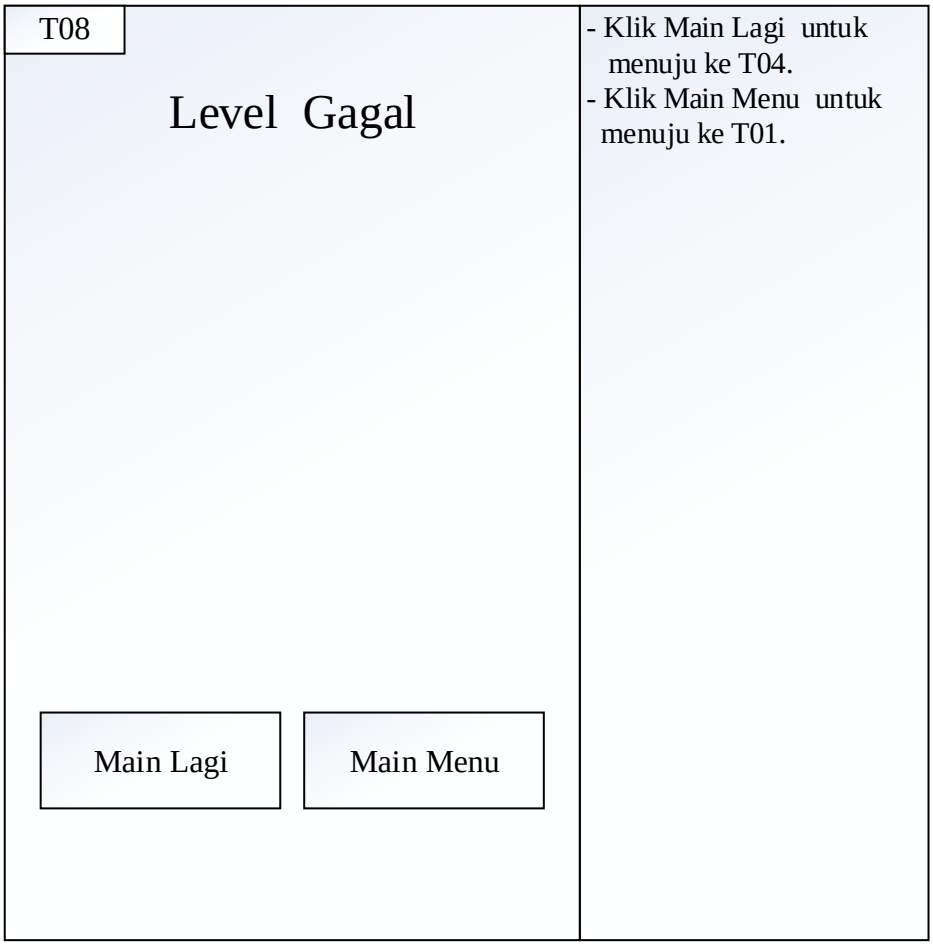


Gambar 3.31 Antarmuka Tampilan Menang

Tabel 3.19 Penggunaan Tombol Menu Menang

No	Jenis Objek	Nama Objek	Keterangan
1	Button	Next Stage	Untuk menuju ke permainan selanjutnya
		Main Menu	Untuk menuju ke menu utama.

g. Antarmuka Permainan Keterangan Kalah

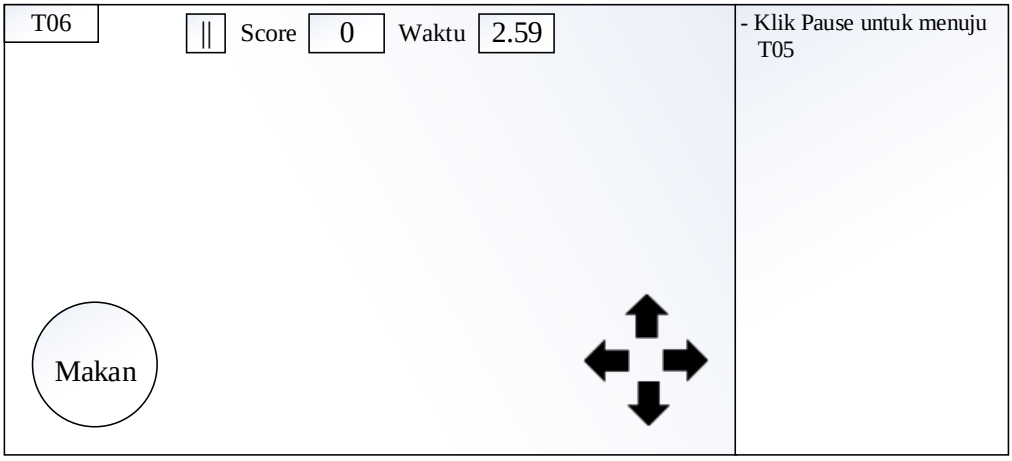


Gambar 3.32 Antarmuka Tampilan Kalah

Tabel 3.20 Penggunaan Tombol Menu Kalah

No	Jenis Objek	Nama Objek	Keterangan
1	Button	Main Lagi	Untuk mengulagi permainan
		Main Menu	Untuk menuju ke menu utama.

h. Antarmuka Permainan Stage 2

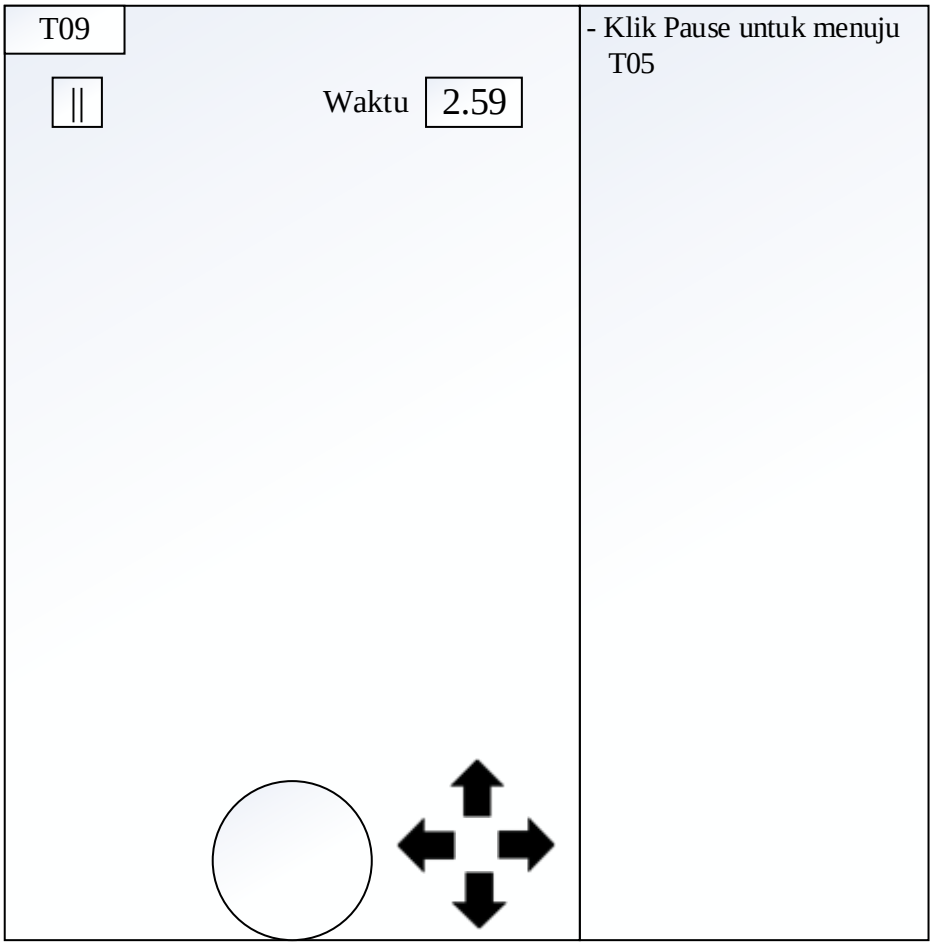


Gambar 3.33 Antarmuka Tampilan *Stage* 2

Tabel 3.21 Penggunaan Tombol Menu *Stage* 2

No	Jenis Objek	Nama Objek	Keterangan
1	Button	||	Untuk mem*pause* permainan
		➡	Untuk mengarahkan karakter ke kanan
		⬇	Untuk mengarahkan karakter ke bawah
		⬅	Untuk mengarahkan karakter ke kiri
		⬆	Untuk mengarahkan karakter ke atas

i. Antarmuka Permainan Stage 3



Gambar 3.33 Antarmuka Tampilan *Stage* 3

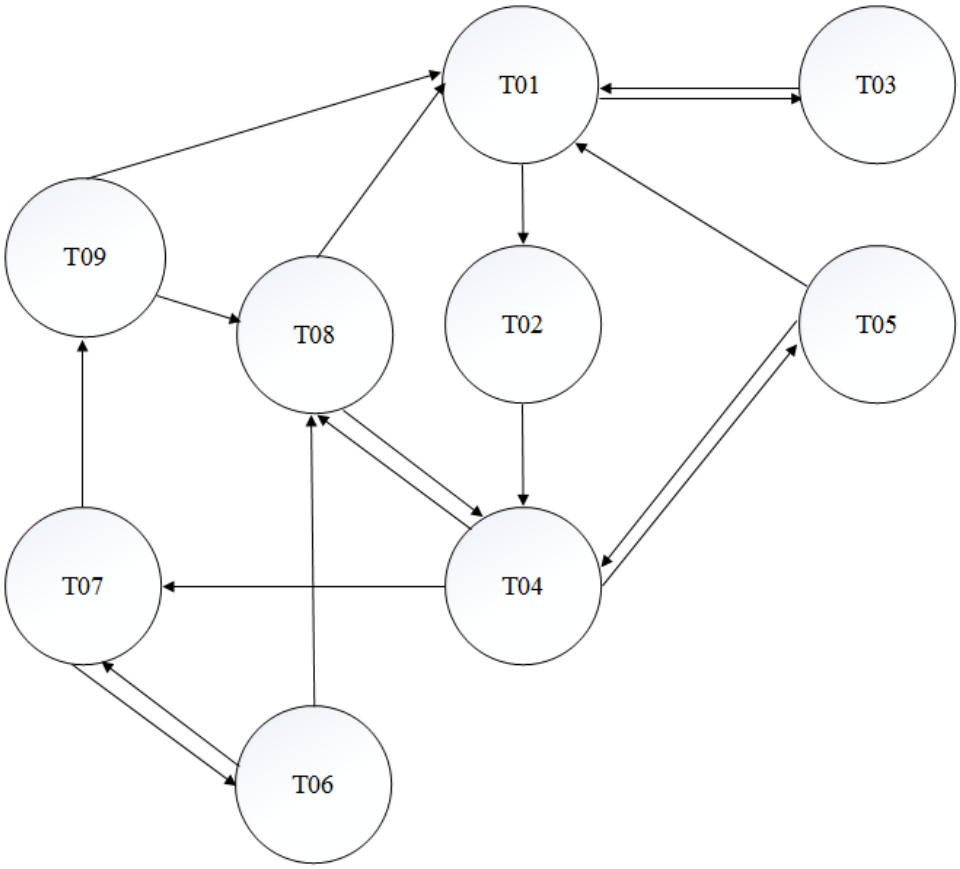
Tabel 3.22 Penggunaan Tombol Menu *Stage* 3

No	Jenis Objek	Nama Objek	Keterangan
1	Button	||	Untuk mem*pause* permainan
		➡	Untuk mengarahkan karakter ke kanan
		⬇	Untuk mengarahkan karakter ke bawah

			Untuk mengarahkan karakter ke kiri
			Untuk mengarahkan karakter ke atas

3.2.2 Jaringan Semantik

Struktur aplikasi ini dibuat secara moduler, yaitu program dipecah menjadi modul-modul kecil yang mudah dibuat, mudah dites, dan mudah dimodifikasi. Dalam pembuatan *game* ini menggunakan alat bantu bagan struktural yang disebut juga jaringan semantik dapat dilihat pada gambar 3.31.



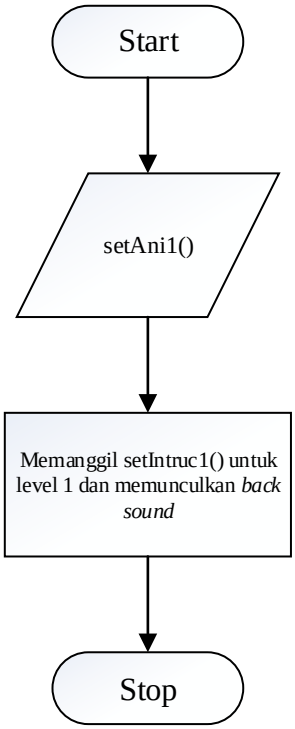
Gambar 3.35 Jaringan Semantik

3.2.3 Perancangan *Method*

Perancangan *method* merupakan perancangan yang berfungsi untuk 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 dalam *game* certia rakyat fabel yang akan dibangun dijelaskan melalui *flowchart* sebagai berikut.

3.2.3.1 Method Play

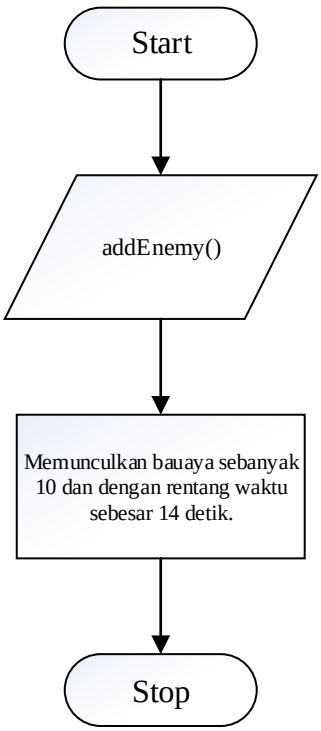
Method ini digunakan untuk proses pemanggilan gambar awal, proses pemanggilan *back sound*, dan proses pemanggilan intruksi permainan yang dapat dilihat pada gambar 3.32.



Gambar 3.36 *Method Play*

3.2.3.2 Method Muncul Buaya

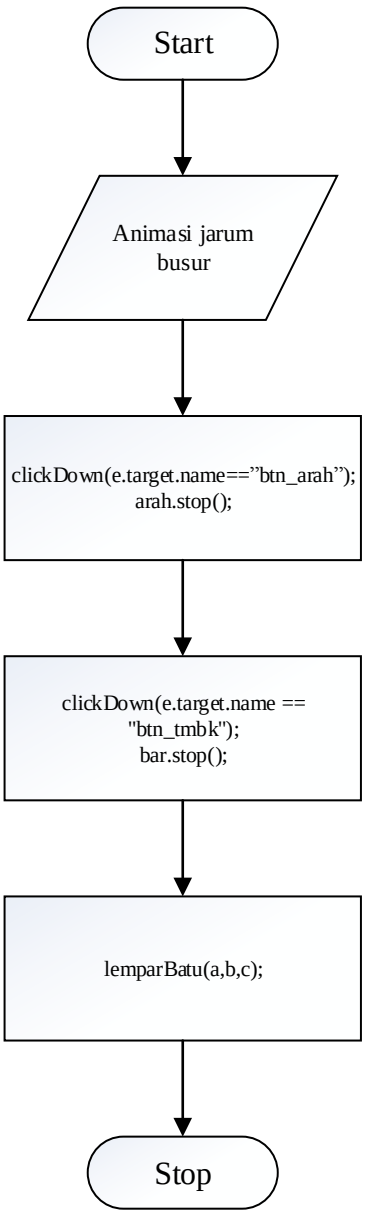
Method ini digunakan untuk proses kemunculan buaya secara random pada Stage 1.



Gambar 3.37 *Method* Muncul Buaya

3.2.3.3 Method Lempar Batu

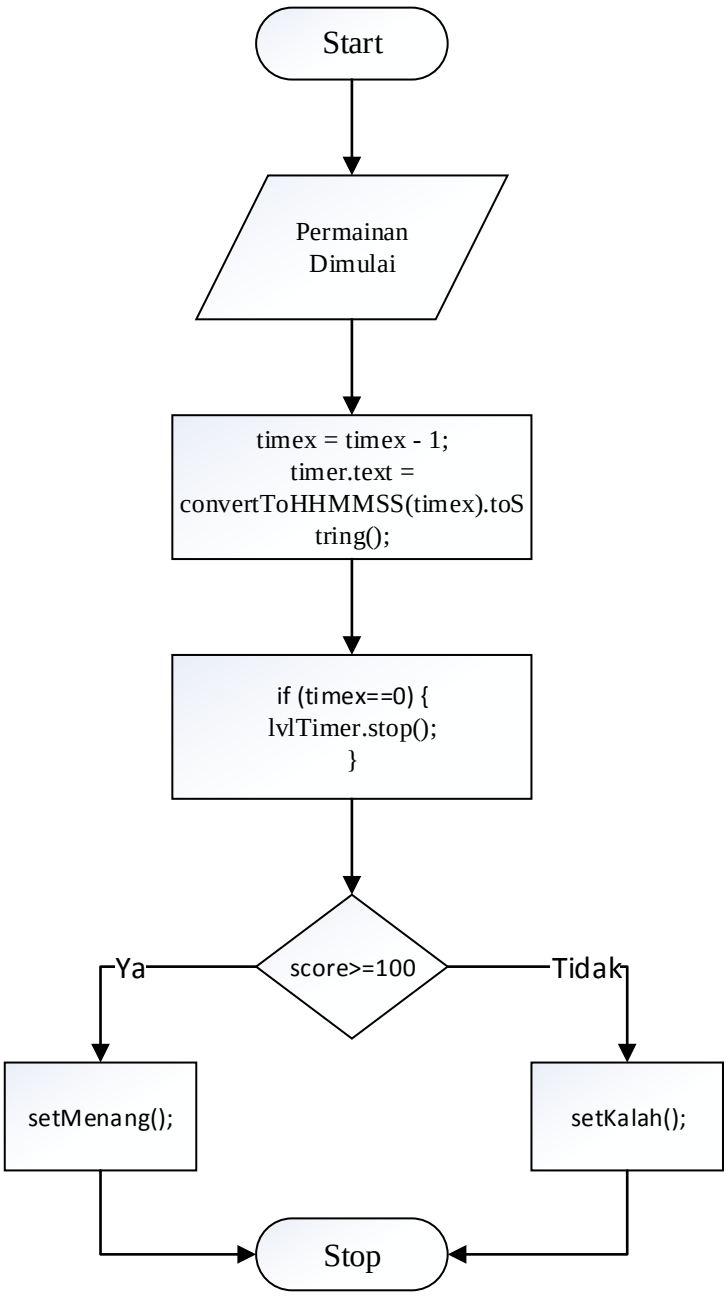
Method ini digunakan dalam proses lempar batu yang terjadi pada Stage 1, untuk melakukan lempar batu ada 2 tahap yang harus dilakukan oleh pemain, yaitu menentukan arah lemparan dan menetukan tenaga lemparan dari batu.



Gambar 3.38 *Method* Tembak

3.2.3.4 Method *Detik*

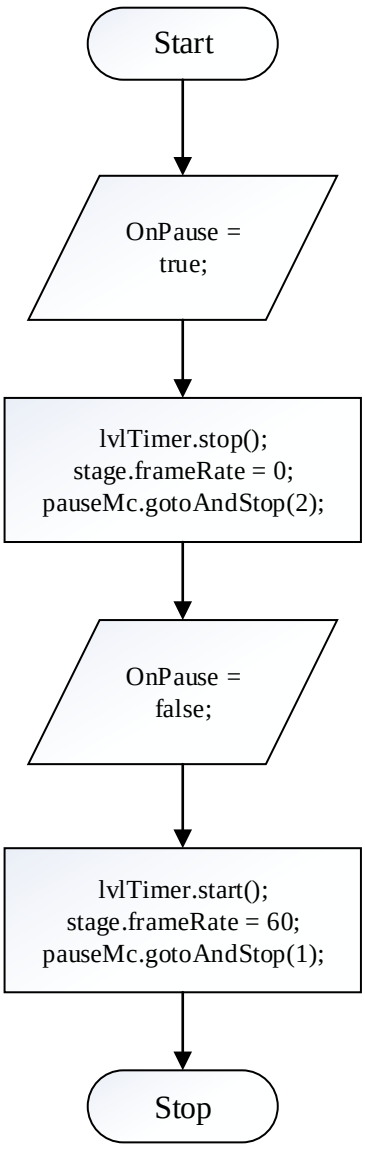
Method ini digunakan dalam proses perhitungan waktu, perhitungan waktu dimulai ketika pemain menekan tombol mulai pada tutorial permainan.



Gambar 3.39 *Method* Detik

3.2.3.5 Method Pause *dan* Rasume

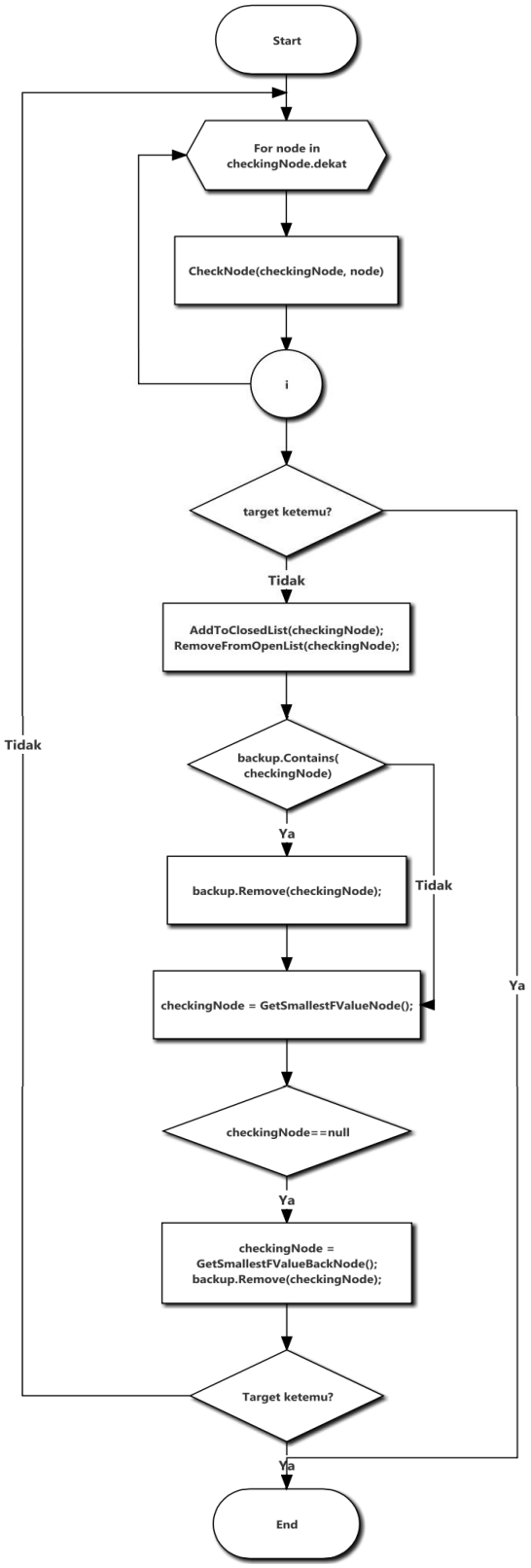
Method ini digunakan dalam proses pause dan rasume game.



Gambar 3.40 ***Method Pause*** **dan** ***Rasume***

3.2.3.6 Method *findpath*

Method ini digunakan ketika buaya mengejar buah di Stage 2.



Gambar 3.41 ***Method findpath***