BAB 3

ANALISIS DAN PERACANGAN SISTEM

Bab ini menjelaskan mengenai analisis dan gambaran dari sistem dan perancangan program aplikasi yang akan dibangun. Analisis merupakan penguraian konsep ke dalam bagian – bagian yang lebih sederhana, sehingga strukturnya menjadi lebih jelas.

3.1 Analisis Sistem

Analisis sistem merupakan penguraian sistem yang utuh ke dalam bagian – bagian komponen nya yang bertujuan untuk mengidentifikasi dan mengevaluasi segala permasalahan dan hambatan yang terjadi serta kebutuhan yang diharapkan, sehingga nantinya dapat menjadi acuan untuk perbaikan – perbaikan pada sistem itu sendiri

3.1.1 Analisis Masalah

Berdasarkan latar belakang masalah yang sudah ada, maka analisis masalah yang didapat adalah sebagai berikut :

1. Tidak semua siswa atau anak – anak mengikuti kegiatan ekstrakurikuler yang mempelajari ilmu pertolongan pertama.
2. Sedikitnya media informasi bersifat edukasi yang menarik yang dapat memberikan informasi dan mengajarkan ilmu pengetahuan di bidang pertolongan pertama kepada siswa atau anak – anak yang berbahasa Indonesia.
3. *Game* edukasi yang sudah ada belum menjelaskan mengenai pengetahuan dasar tentang pertolongan pertama, alat pelindung diri , dan luka.

3.1.2 Analisis *Game* Sejenis

Sebelum merancang *Game* yang akan dibangun, terlebih dahulu dilakukan analisis pada *Game* sejenis. Tujuan dari analisis ini adalah untuk mendapatkan

konsep permainan dan perbandingan dengan *Game* yang telah ada. *Game* sejenis yang bertemakan pertolongan pertama yang akan dijadikan pembanding adalah *Game Life Runner ELS*..

a. Pengenalan *Game Life Runner ELS*

Game ini merupakan *Game online* yang bertemakan tentang pertolongan pertama. Didalam *Game* ini juga terdapat tiga macam mini *Game*s yaitu *Recovery Position, CPR,* dan *Chockin Aid*. Permainan mini *Game* ini bukan lah permainan utama, para pemain dapat memilih memainkan permainan mini *Game* ini satu persatu tanpa harus memenangkan terlebih dahulu permainan yang lainnya. Pada permainan utama menolong korban yang ada pada sebuah mall Pemain dapat memilih tokoh wanita atau tokoh pria. Pada *Game utama* ini hanya terdapat 1 level permainan saja.

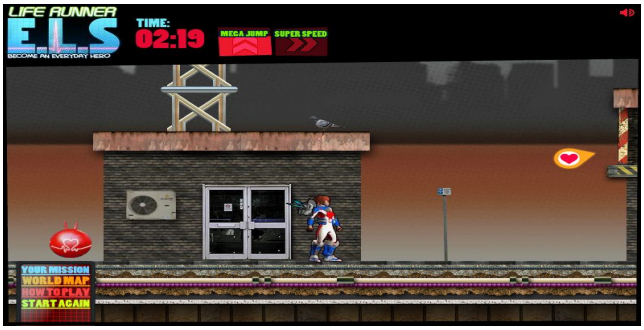


Gambar 3.1 Tampilan awal *Game Life Runner ELS*

b. Gameplay

Misi pada *Game* didalam permainan utama ini adalah menyelamatkan seorang korban yang berada di sebuah mall, korban tersebut membutuhkan tindakan CPR. Rute map yang harus dilewati oleh pemain adalah mulai dari *city – park – mall.* Pemain harus mengikuti tanda hati atau panah untuk menemui korban nya. Dalam *Game* ini terdapat i*con* atau tanda panah ganda *vertical* berwarna biru yang digunakan untuk melompat lebih tinggi dan tanda panah ganda *horizontal* berwarna merah untuk melakukan lari cepat. Kedua *icon* panah ganda tersebut harus dicari atau di dapat sendiri dan waktu pemakaian nya pun terbatas. Pemain dapat melakukan gerakan dengan menggunakan keyboard atas, kiri dan kanan. Dalam permainan ini

pemain akan mengalami keterlambatan dalam berjalan atau berlari jika berpapasan dengan orang – orang yang ada dalam *Game* tersebut.



Gambar 3.2 Tampilan *Game Life Runner ELS* mencari korban.

Seperti yang sudah dijelaskan pada bagian pengenalan, di dalam *Game* ini juga terdapat 3 mini *Game*s yang mempunyai *Game*play dan misi yang berbeda – beda seperti :

1) *Recovery Position* (Posisi Pemulihan)

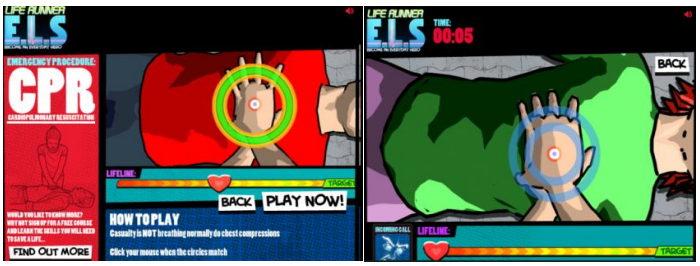
Dimana dalam mini *Game* ini pemain harus melakukan pemindahan atau menempatkan korban yang bernapas normal namun tidak sadar ke dalam posisi pemulihan dengan mengikuti instruksi yang ada. Permainan ini menggunakan mouse.



Gambar 3.3 Tampilan mini *Game Life Runner ELS* "Recovery Position".

2) CPR (*Cardio Pulmonary Resuscitation*)

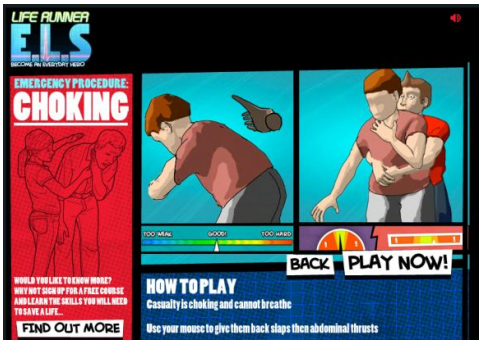
Pada mini *Game* ini pemain harus melakukan tindakan CPR terhadap korban yang bertujuan untuk menyelamatkan nyawa korban.



Gambar 3.4 Tampilan mini *Game Life Runner ELS* "CPR".

3) Choking Aid

Misi pada *Game* ini pemain harus membantu korban yang tersedak dengan cara memukul pudak dan menarik perut korban dari belakang dengan cara sambil memeluk korban.



Gambar 3.5 Tampilan mini *Game Life Runner ELS "Choking Aid"*

c. Kelebihan dan kekurangan *Game*

Didalam *Game* utama pada permainan ini memiliki tingkat kesulitan dalam perjalanan yang harus ditempuh oleh pemain, karna perjalanan tersebut arahnya masih membingungkan atau kurang jelas, serta waktu yang disediakan oleh *Game* untuk menemukan korban terlalu singkat, sehingga menyulitkan pemain untuk segera menolong korban, akan tetapi dengan memainkan *Game* ini pemain dapat mengetahui bagaimana menolong korban dengan cara melakukan CPR, *recovery position* atau posisi pemulihan, dan membantu korban tersedak. Sayangnya pada *Game* utama dan *mini Game* yang telah ada ini belum terdapat informasi mengenai dasar pertolongan pertama, pengenalan alat perlindungan diri, dan cara penanganan terhadap luka yang sering di derita oleh para siswa dan anak – anak serta bahasa yang digunakan menggunakan bahasa

asing,karena gam ini merupakan *Game* buatan luar negri sehingga dimungkinkan banyak anak – anak Indonesia yang terkendala oleh bahasa yang digunakan pada *Game* tersebut.

3.1.3 Analisis *Game* Pertolongan Pertama.

Analisis *Game* pertolongan pertama merupakan penguraian pembahasan mengenai penjelasan – penjelasan yang berkaitan dengan *Game* pertolongan pertama.

3.1.4 Pengenalan Aplikasi

Game yang akan dibuat berjudul *Game* Pertolongan Pertama yang bergenre *Adventure Game. Game* ini dibangun untuk mensosialisasikan ilmu pengetahuan mengenai pertolongan pertama kepada siswa atau anak – anak yang tidak bisa mengikuti kegiatan ekstrakurikuler yang mengajarkan tentang pertolongan pertama di sekolah. Target pengguna atau pemain pada *Game* ini mulai dari 7 – 15 tahun atau pada usia sekolah berada di tingkat SD dan SMP. *Game* pertolongan pertama dibangun dengan menggunakan grafis 2 dimensi (2D) dan menerapkan sistem s*ingle Player* (perorangan). Adapun beberapa kelebihan yang akan dibuat pada *Game* ini adalah :

1. Memberikan pengenalan sekaligus pembelajaran mengenai dasar pertolongan pertama
2. Memberikan pengenalan dan pembelajaran mengenai peralatan dan obat yang dapat dijadikan sebagai alat bantu melakukan pertolongan pertama, dan pengetahuan tentang luka.
3. Aplikasi *Game* ini bersifat bermain sambil belajar.

3.1.5 Story Line

Game pertolongan pertama ini menceritakan tentang petualangan perjalanan seorang anak kecil bernama “Suny” yang ingin menonton pentas teater anak disebuah taman rekreasi tanpa didampingi orang tuanya namun suny sudah mengingat pelajaran yang di informasikan oleh pamannya yang seorang dokter mengenai pertolongan pertama untuk mengantisipasi keadaan jika ia memerlukan

bantuan. Suny selalu membawa tas yang berisi berbagai peralatan pertolongan pertama atau yang biasa disebut dengan tas P3K / PP untuk membantu nya menghadapi beberapa hal dan kemungkinan kecelakaan yang terjadi selama suny melakukan pertualangan. *Game* ini terbagi menjadi 3 stage level. Dimana dalam setiap permainan (levelnya), Suny berada di lokasi yang berbeda – beda.

3.1.6 Analisis Level

Sebelum memasuki level arena permainan akan ditampilkan animasi intro Yang menceritakan tetang keinginan Suny untuk pergi dengan ayahnya ke pentas teater yang diadakan di sebuah taman hiburan, akan tetapi ayah Suny meminta maaf karena tidak dapat menemani Suny dan menyarankan suny untuk mengajak pamannya. Mendengar saran dari ayahnya Suny pun berpikir dan mengingat bahwa pamannya pasti tidak bisa menemaninya pergi karena setiap hari libur pamanya yang seorang dokter itu selalu membuka praktek dirumah nya. Akhirnya Suny pun mengatakan kepada ayahnya bahwa Ia dapat pergi sendirian ke pentas teater yang ada di taman hiburan tersebut.

Level pertama, pada level ini pemain berada didalam sebuah ruangan (rumah), disini pemain terlebih dahulu mengumpulkan kelengkapan peralatan dan obat – obatan yang akan dimasukan ke dalam tas PP dan mengambil koin informasi, selanjutnya pemain akan menemukan dan menolong satu korban saja. Setiap alat dan koin yang diambil oleh pemain akan menampilkan informasi mengenai fungsi dari masing – masing alat dan koin tersebut dan korban yang harus ditolong menderita luka yang harus disembuhkan dengan menggunakan peralatan dan obat yang sudah ada pada tas PP. Pemain tidak dapat melanjutkan permainan ke level berikutnya jika tidak menyelesaikan target permainan pada level ini.

Level kedua pemain berada diluar ruangan, tepatnya berada pada sebuah jalan yang terletak di sisi lapangan yang ditumbuhi pohon, rumput dan terdapat beberapa benda seperti bola, sepeda, gawang bola, dan pagar. Pada level ini terdapat dua korban yang akan ditolong oleh pemain, dimana salah satu dari kedua korban tersebut harus ditolong terlebih dahulu dari pada korban yang

lain. Jika pemain berhasil memainkan keseluruhan level ini maka pemain akan mendapatkan skor. Apabila pemain tidak berhasil menolong korban atau kehabisan waktu, maka pemain tidak dapat melanjutkan ke level berikutnya dan akan mengulang permainan pada level yang terakhir pemain mainkan.

Level ketiga pemain berada pada sebuah taman rekreasi anak. Disini pemain diinformasikan untuk membantu para petugas taman menemukan korban yang pingsan. Dimana untuk menolong korban tersebut pemain terlebih dahulu harus mengumpulkan potongan peta yang terbagi menjadi 4 bagian, dimana peta tersebut diperlukan sebagai penunjuk ambulance mendatangi posisi penolong dan korban. Setelah peta berhasil dikumpulkan pemain harus mencari letak posisi korban. Setelah bertemu korban, pemain harus menghubungi petugas medis, maka ambulance akan menemukan jalur terpendek untuk melakukan pertolongan lebih lanjut terhadap korban yang sudah ditemukannya. Setelah korban berhasil dibawa oleh ambulance maka pemain harus pergi menuju pentas teater sebagai tujuan utama nya, jika pemain berhasil melakukan misi pada *Game* ini maka pemain sudah menyelesaikan semua permainan dan diakhir permainan ini pemain dapat mengisi nama nya untuk ditampilkan pada daftar nilai tertinggi.

Pemain dalam permainan ini sudah dibekali oleh tas PP yang berisi alat dan obat – obatan yang dapat digunakan oleh pemain dalam melakukan tindakan pertolongan. Dalam level ini juga terdapat limit waktu (*timer*) dimana pemain harus menyelesaikan permainan sebelum waktu (*timer*) nya habis. Jika pemain tidak dapat menyelesaikan misi ditiap level karena kehabisan waktu maka permainan akan gagal dan pemain akan kalah.

3.1.7 Analisis Tingkat Kesulitan

Analisis tingkat kesulitan ini menceritakan tentang kesulitan atau tantangan yang akan dihadapi oleh pemain pada setiap levelnya.

1. Level pertama

   Pemain akan dihadapkan dengan satu orang korban yang menderita luka ringan dan pemain harus menghindari rintangan / NPC (tikus dan paku),

serta memperhitungkan waktu/ *timer*, agar darah pemain tidak habis dan tidak kekurangan waktu.

2. Level kedua

   Pemain dihadapkan dengan dua orang korban dengan luka yang berbeda dan harus melakukan penilaian terhadap korban untuk mengetahui siapa korban yang harus terlebih dahulu ditolong. Pada level ini juga akan ada *timer*/ waktu dan NPC (anjing dan duri) yang akan mengurangi darah pemain, jika pemain mengenai NPC tersebut.

3. Pemain diharuskan untuk mengumpulkan peta dan melakukan penolongan terhadap korban, dengan cara menghubungi petugas medis, dan pergi ke pentas teater anak yang menjadi tujuan utama nya. Jika pemain salah menghubungi bantuan maka pemain akan kalah dan mengulang kmbali permainan yang ada pada level ini.

3.1.8 Analisis Scoring

Analisis scoring adalah analisa untuk memberikan penilaian kepada pemain yang menyelesaikan tiap tahapan level yang ada pada *Game*. Analisis scoring yang digunakan pada *Game* ini berupa jumlah angka score. Berikut adalah analisis scoring pemain

Tabel 3.1 Analisis Scoring

Level	Keterangan	Score
Level Pertama	1. Pengumpulan setiap satu buah alat dan obat – obatan	Mendapat score 50
	2. Melakukan pertolongan kepada korban dengan menggunakan setiap satu peralatan dan obat yang ada di tas PP	Mendapat score 100
	3. Pengumpulan setiap satu alat pengecoh	Tidak Mendapat tambahan skor
Level Kedua	1. Menolong korban pertama	Mendapat score 100
	2. Menolong korban kedua	Mendapat score 100

	3. Menjawab quiz dengan benar	Mendapat score 100
Level Ketiga	1. Mengumpulkan peta	Mendapat score 100
	2. Menolong korban	Mendapat score 100

Tabel 3.1 Analisis Scoring

Level	Keterangan	Score
	3. Menghubungi bantuan	Mendapat score 100
	4. Pergi ke pentas teater	Mendapat score 100

3.1.9 Analisis Komponen *Game*

Komponen yang terdapat dalam *Game* pertolongan pertama diantaranya adalah sebagai berikut :

1. Arena permainan

   Pada permainan *Game* pertolongan pertama arena permainannya adalah didalam sebuah ruangan, dan luar ruangan (lapangan dan tempat rekreasi anak). Didalam arena tersebut akan muncul korban – korban yang harus ditolong oleh pemain.

2. Pemain

   Pemain yang bernama “Suny” merupakan aktor utama dalam permainan ini. Suny dapat melakukan kegiatan berjalan, melompat, menolong korban dan menghindari rintangan. Misi dari Suny adalah untuk mengumpulkan alat, menolong korban, mengumpulkan peta, memanggil bantuan, dan mendatangi pentas teater anak.

   Korban yang ada dalam permainan ini berjumlah 4 orang anak. Korban yang muncul menderita luka yang berbeda – beda dimulai dari luka terbuka, luka tertutup, patah tulang dan pingsan.

3. Peralatan tas PP

   Isi dari peralatan tas PP yang ada didalam tas dan akan dikumpulkan oleh pemain adalah:

   a. Alat perlindungan diri (APD) berupa sarung tangan lateks, masker penolong, dan kacamata peindung.

b. Alat – alat bantu berupa kasa steril, pembalut gulung atau perban, gunting, pinset, kapas, dan plester.
c. Obat – obatan berupa cairan *anti septic*, dan *alcohol.*

3.1.10 *Gameplay*

Dalam memainkan permainan ini, tugas utama para pemain yaitu menyelesaikan misi yang tersedia pada setiap level dalam permainan ini. Setiap level mempunya misi yang berbeda – beda, berikut adalah misi – misi permainan yang harus diselesaikan tersebut.

1. Level pertama

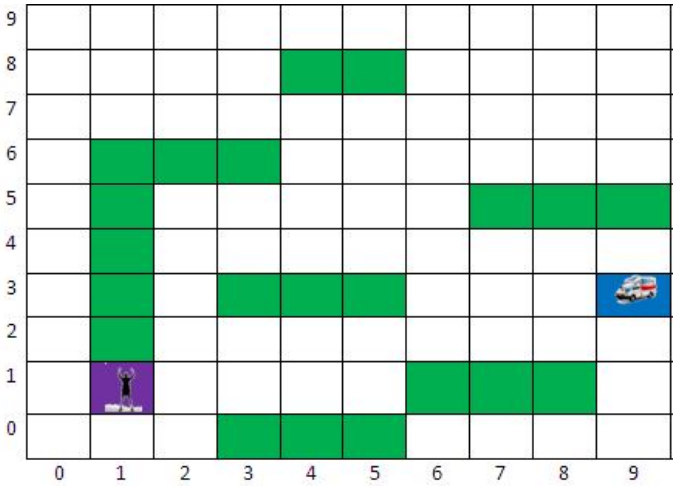
   Pada level ini pemain harus menyelesaikan misi mengumpulkan peralatan yang merupakan alat perlindungan diri (APD) seperti : masker, kacamata penolong, obat luka, dll. Pemain juga harus mengambil koin informasi yang berisi informasi mengenai pertolongan pertama. Setelah berhasil mengumpulkan peralatan APD dan koin informasi maka pemain diharuskan menolong korban, dan menggunakan peralatan yang sudah dikumpulkan tadi untuk menolong korban tersebut. Pemain tidak boleh kehabisan waktu dan kehabisan darah karena mengenai rintangan/NPC.
2. Level kedua

   Misi pada level ini adalah menentukan koban mana yang harus ditolong terlebih dahulu, karena pada level ini terdapat 2 orang korban dimana salah satu korban menderita luka berat dan salah satu korban lainnya hanya menderita luka ringan. Pemain tidak boleh kehabisan waktu dan kehabisan darah karena mengenai rintangan.
3. Level ketiga

   Pada level akhir ini pemain harus mengumpulkan potongan peta, menemukan korban yang harus ditolong dan memangil bantuan dengan dengan benar. Setelah berhasil menolong korban pemain melanjutkan misinya untuk pergi k petas teater anak. Pemain tidak boleh kehabisan waktu. Dan jika waktu pemain habis maka pemain akan gagal melakukan misi, serta dinyatakan kalah bermain.

3.1.11 Analisis Algoritma A* (A Star)

Algoritma A* (A star) pada *Game* ini akan diterapkan pada mobil ambulance yang digunakan untuk menjemput korban yang sudah ditemukan pemain, dimana tujuan penggunaan algoritma A star ini adalah untuk mencari jalur terpendek yang akan ditempuh oleh ambulance menuju korban. Seperti dalam gambar 3.6, yang menggambarkan posisi awalan ambulance, pemain, dan korban.



Gambar 3.6 Penempatan Pemain, Korban dan Ambulance

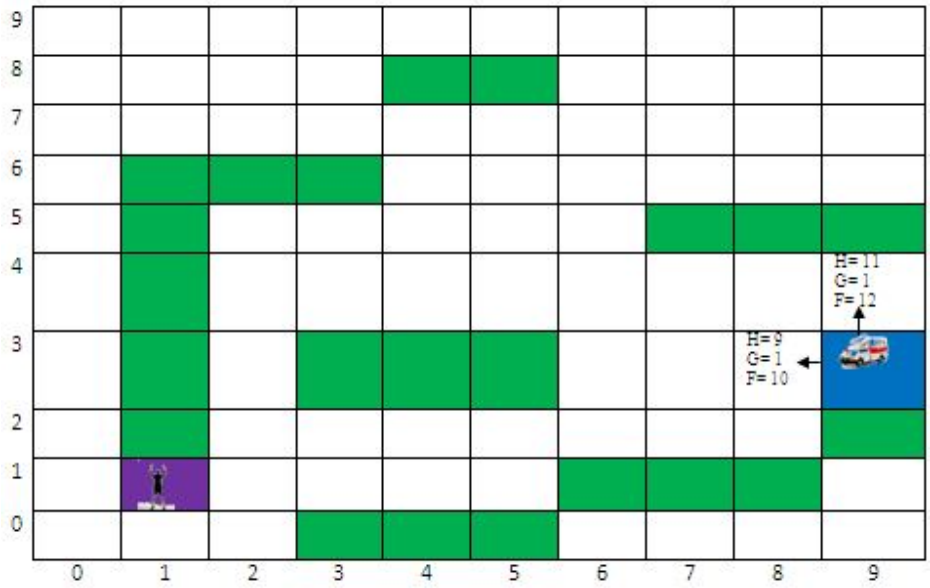
Dengan penjelasan isi gambar 3.6 dalam tabel 3.2 seperti berikut ini.

Tabel 3.2 Keteragan Warna Pada *Node* Ordo

Warna	Keterangan
	Block atau dinding penghalang (collider) ambulance dan pemain yang berada pada koordinat : (1,2),(1,3),(1,4),(1,5),(1,6),(2,6),(3,6),(3,0),(3,3),(4,0),(4,3),(4,8),(5,0),(5,3),(5,8),(6,1),(7,1)(7,5),(8,1),(8,5),(9,5)
	Posisi pemain bersama korban (1,1)
	Posisi awal ambulance & rumah sakit (9,3)
	Rute terpendek hasil perhitungan a*

Berikut adalah keterangan dan gambar yang akan menjelaskan langkah per langkah perhitungan A star dengan menggunakan fungsi heuristic manhattan distance, dimana dalam perhitungan menggunakan fungsi tersebut pencarian jarak / jalur yang diperbolehkan hanya pencarian secara *horizontal* maupun *vertical* dan posisi awalan yang sudah dijelaskan pada gambar 3.7.

Perhitungan Langkah Pertama.



Gambar 3.7 Penjelasan Pergerakan Algoritma A* Langkah Pertama

Penjelasan gambar 3.7 diatas yaitu pada langkah pertama, *node* (9,3) dimasukan kedalam *Open list* dan selanjutnya memeriksa neighbor *node* atau *node* yang berdekatan dengan *node* (9,3) yaitu (9,4) dan (8,3), Setelah diperiksa *node – node* tersebut dimasukan ke dalam *open list*. Kemudian hapus *node* (9,3) dari *Open list* dan masukan *node* (9,3) kedalam *closed list*. Setelah *node – node* yang berada didalam *open list*, maka pilih lah *node* dengan nilai *f(n)* yang paling rendah, yaitu *node* (8,3) dengan nilai *f(n)*= 10

Perhitungan A* akan diilustrasikan dengan mengambil salah satu contoh dari *node* (8,3) degan perhitungan sebagai berikut :

Nilai n = (8,3) : nilai g(n)= (8,3) = 1

Dengan *node* tujuan adalah (1,1) dengan target X= 1 dan target Y = 1

Nilai h (8,3) = (abs(n.X – goal.X) + abs (n.Y – goal.Y))

= (abs(8-1) + abs(3-1)

= (abs(7) + abs(2)

= (7+2)

= 9

Jadi nilai *f(n)* = g(n)+h(n)

= 1+9

= 10`

Selanjutnya cara perhitungan A* pada langkah pertama ini semuanya sama. Berikut dibawah ini adalah tabel 3.3 hasil nilai keseluruhan *node* pada langkah pertama.

Tabel 3.3 Hasil Perhitungan A* langkah Pertama

Node	Nilai *g(n)*	Nilai *h(n)*	Nilai *f(n)*
9,4	1	11	12
8,3	1	9	10

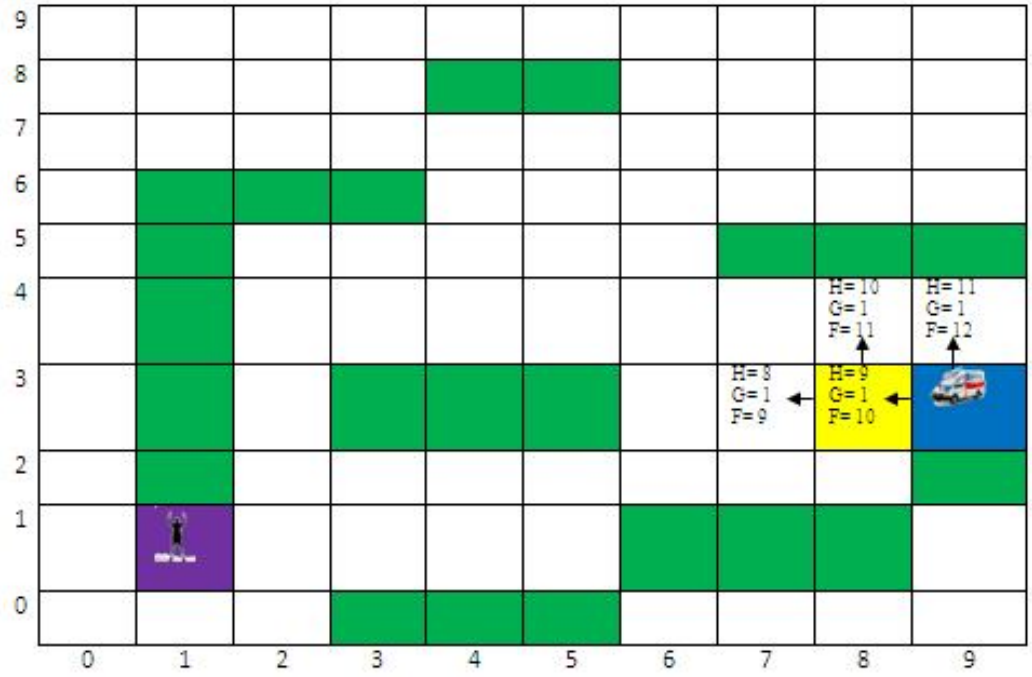
Selanjutnya pada langkah pertama yang menghasilkan *node – node* didalam *Open list* dan *Closed list* dijelaskan dengan tabel 3.4vsebagai berikut :

Tabel 3.4 Tabel Perhitungan Langkah Pertama

Open list	(9,4),(8,3)
Closed list	(9,3),
Parent	(9,3)
Current node	(8,3)

Proses yang telah dijelaskan diatas dilakukan berulang kali atau looping, dimulai dari *open list*, hingga menemukan *node* dengan nila *f(n)* paling rendah.

Berikut ini adalah perhitungan pada langkah kedua yang ada pada koordinat (8,3) yang digunakan pada langkah awal



Gambar 3.8 Penjelasan Pergerakan Algoritma A* Langkah Ke-2

Pada gambar 3.8 diatas terlihat ada 2 *node* yang masuk kedalam *Open list* yaitu *node* (8,4) dan (7,3). Sedangkan *node* (9,3),(9,4), dan (8,3) sudah dimasukan kedalam *closed list*. Dari langkah kedua ini didapat *node* dengan nilai *f(n)* terendah adalah (7,3) dengan nilai *f(n)* = 9. Berikut adalah penjelasan hasil tabel perhitungan langkah kedua :

Tabel 3.5 Hasil Perhitungan A* Langkah ke-2

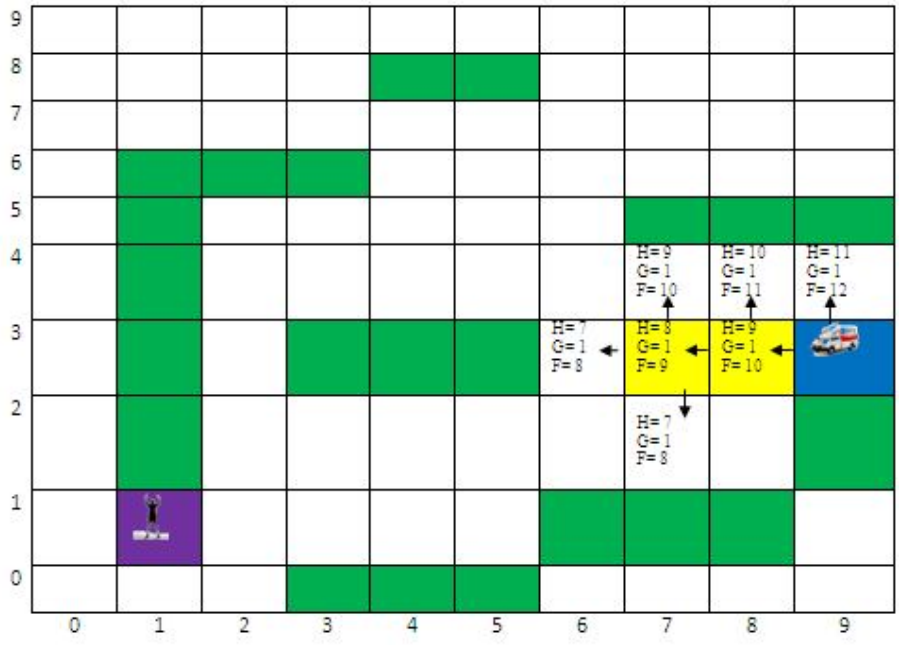
Node	Nilai *g(n)*	Nilai *h(n)*	Nilai *f(n)*
8,4	1	10	11
7,3	1	8	9

Dengan penjelasan perhitungan langkah kedua melalui tabel berikut :

Tabel 3.6 Penjelasan Perhitungan Langkah Kedua

Open list	(8,4) dan (7,3)
Closed list	(9,3),(9,4), dan (8,3)
Parent	(8,3)
Current node	(7,3)

Berikut ini adalah perhitungan langkah ketiga dalam pencarian jalur terpendek A* :



Gambar 3.9 Penjelasan Pergerakan Algoritma A* Langkah Ketiga

Pada gambar 3.9 terlihat ada 3 *node* yang masuk kedalam *Open list* yaitu *node* (7,4),(7,2) dan (6,3). Sedangkan *node* (9,3), (8,3), (8,4) dan (7,3) sudah dimasukan kedalam *closed list*. Dari langkah ketiga ini didapat 2 *node* dengan nilai *f(n)* terendah adalah (7,2),(6,3) dengan nilai *f(n)* = 7. Berikut adalah penjelasan hasil tabel 3.7 perhitungan langkah ketiga :

Tabel 3.7 Hasil Perhitungan A* Langkah Ketiga

Node	Nilai *g(n)*	Nilai *h(n)*	Nilai *f(n)*
7,4	1	8	9
7,2	1	7	8
6,3	1	7	8

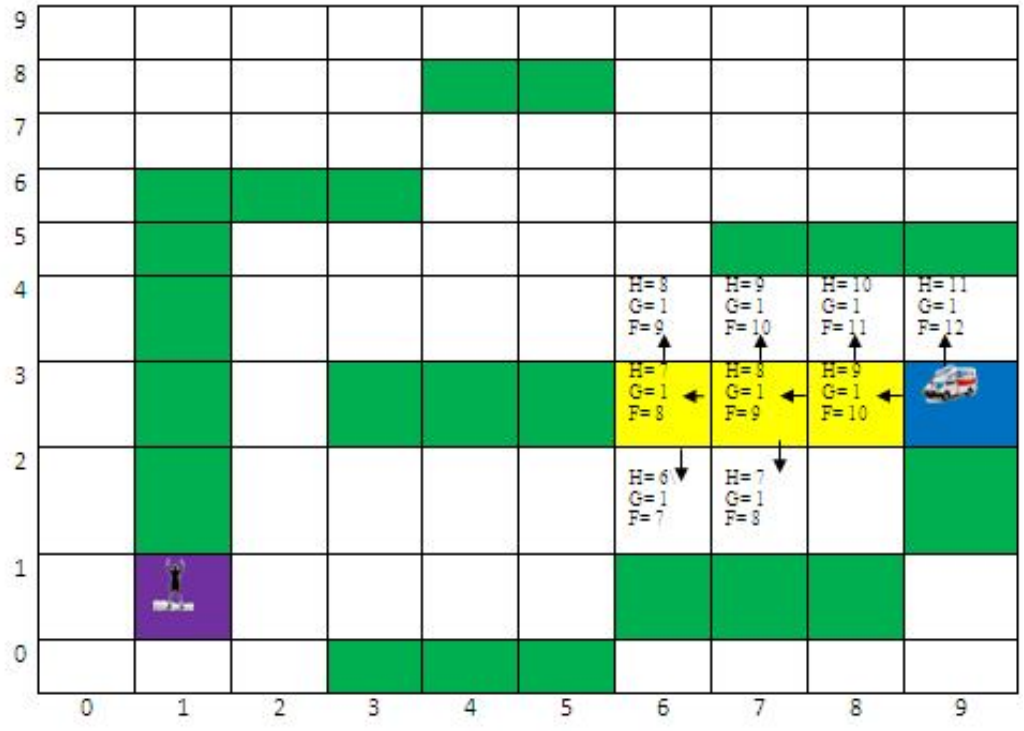
Jika kondisi seperti ini terjadi maka sistem akan memeriksa *node-node* yang masih berada di *open list* yaitu *node* (7,4) untuk dicek apakah nilai dari *node* tersebut tidak lebih kecil dari *node* (7,2) dan (6,3), karena nilai dari *node* (7,4) tidak lebih kecil, maka sistem akan memilih *node* (6,3) sebagai *node* dengan nilai *f* lebih kecil

meskipun nilai *f node* tersebut sama dengan *node* (7,2). hal ini disebabkan sistem akan menentukan nilai *f* pada *node* terakhir yang ditambahkan ke dalam *open list*.

Tabel 3.8 Penjelasan Perhitungan Langkah Ketiga:

Open list	(7,4),(7,2),dan(6,3)
Closed list	(9,3),(9,4), (8,3), dan (7,3)
Parent	(7,3)
Current node	(6,3)

Dalam gambar 3.10 akan dijelaskan menggenai perhitungan langkah keempat menemukan jalur terpendek.



Gambar 3.10 Penjelasan Pergerakan Algoritma A* Langkah Keempat

Pada gambar3.10 diatas terlihat ada 2 *node* yang masuk kedalam *Open list* yaitu *node* (6,4) dan (6,2). Sedangkan *node* (9,3), (9,4), (8,3), (7,3), (7,4), (7,2), dan (6,3)sudah dimasukan kedalam *closed list.* Dari langkah keempat ini didapat *node* dengan nilai *f(n)* terendah adalah (6,2) dengan nilai *f(n)* = 7.

Berikut adalah penjelasan hasil tabel 3.9 perhitungan langkah keempat :

Tabel 3.9 Hasil Perhitungan A* Langkah Keempat

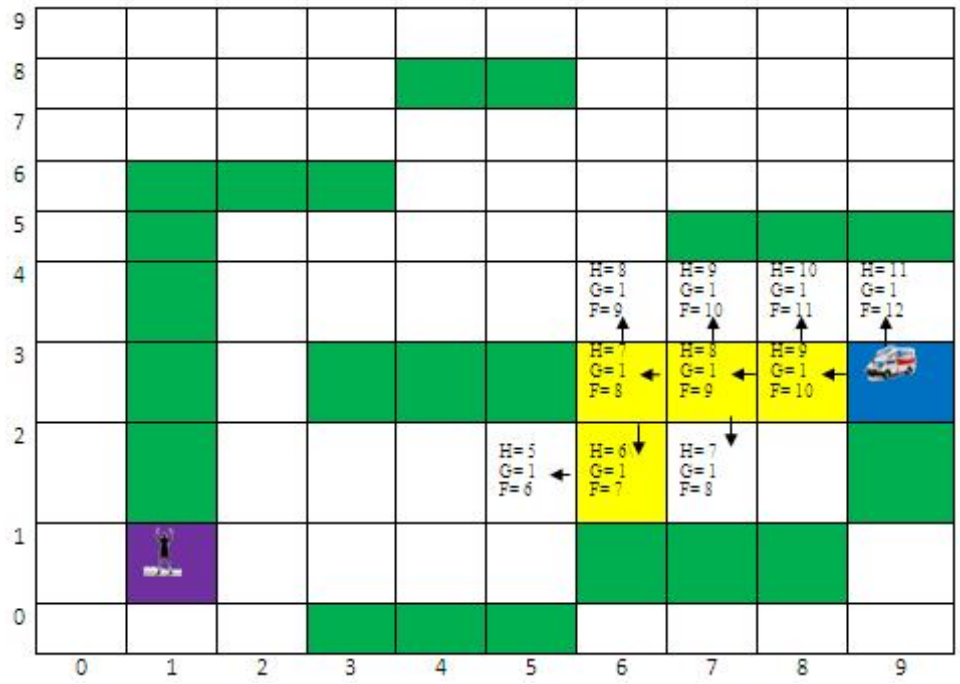
Node	Nilai *g(n)*	Nilai *h(n)*	Nilai *f(n)*
6,4	1	8	9
6,2	1	6	7

Berikut adalah tabel penjelasan *node - node* perhitungan langkah keempat :

Tabel 3.10 Penjelasan Perhitungan Langkah Keempat

Open list	(6,4) dan (6,2)
Closed list	(9,3),(9,4), (8,3), (7,3) ,(7,4),(7,2),(6,3)
Parent	(6,3)
Current node	(6,2)

Dalam gambar 3.11 akan dijelaskan menggenai perhitungan langkah kelima menemukan jalur terpendek.



Gambar 3.11 Penjelasan Pergerakan Algoritma A* Langkah Kelima

Pada gambar 3.11 diatas terlihat ada 2 *node* yang masuk kedalam *Open list* yaitu *node* (5,2). Sedangkan *node* (9,3), (9,4), (8,3), (7,3) ,(7,4), (7,2), (6,3), (6,4), dan (6,2)sudah dimasukan kedalam *closed list*. Dari langkah kelima ini didapat *node* dengan nilai *f(n)* terendah adalah (5,1) dengan nilai *f(n)* = 40. Berikut adalah penjelasan hasil tabel perhitungan langkah kelima :

Tabel 3.11 Hasil Perhitungan A* langkah kelima

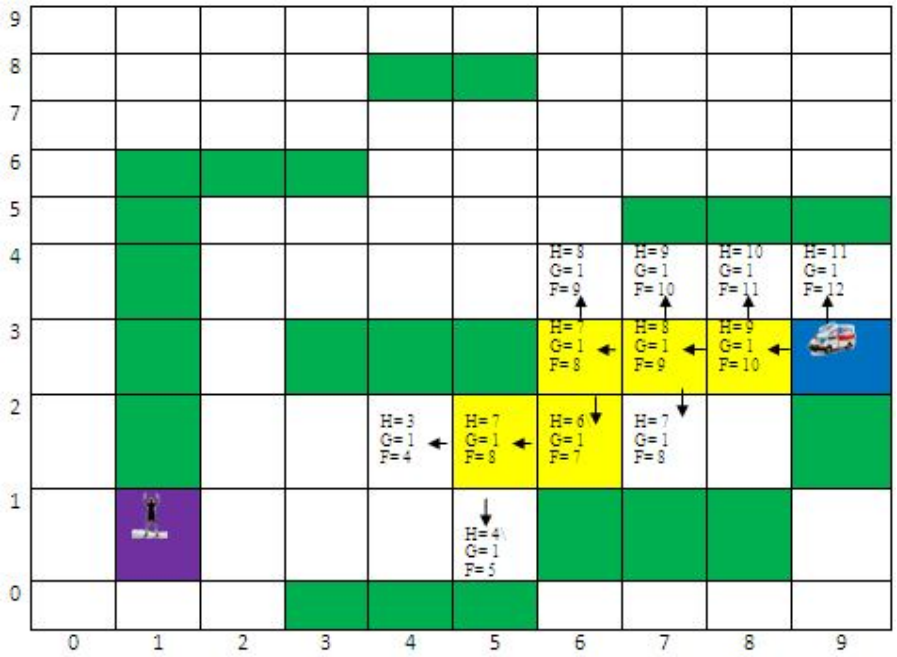
Node	Nilai *g(n)*	Nilai *h(n)*	Nilai *f(n)*
5,2	1	5	6

Berikut adalah tabel yang berisi penjelasan *node* - *node* perhitungan langkah kelima :

Tabel 3.12 Penjelasan Perhitungan Langkah Kelima

Open list	(5,2)
Closed list	(9,3),(9,4), (8,3), (7,3) ,(7,4),(7,2),(6,3),(6,4),(6,2)
Parent	(6,2)
Current node	(5,2)

Sebab *node* belum mencapai *node* tujuan maka dilakukan kembali pencarian dengan mengunakan langkah keenam.



Gambar 3.12 Penjelasan Pergerakan Algoritma A* Langkah Keenam

Pada gambar 3.12 diatas terlihat ada 2 *node* yang masuk kedalam *Open list* yaitu *node* (5,1) dan (4,2). Sedangkan *node* (9,3), (9,4), (8,3), (7,3), (7,4), (7,2), (6,3), (6,4), (6,2), (5,2) sudah dimasukan kedalam *closed list*. Dari langkah keenam ini didapat *node* dengan nilai *f(n)* terendah adalah (4,2) dengan nilai *f(n)* = 4. Berikut adalah penjelasan hasil tabel perhitungan langkah keenam :

Tabel 3.13 Hasil Perhitungan A* langkah keenam

Node	Nilai *g(n)*	Nilai *h(n)*	Nilai *f(n)*
5,1	1	4	5
4,2	1	3	4

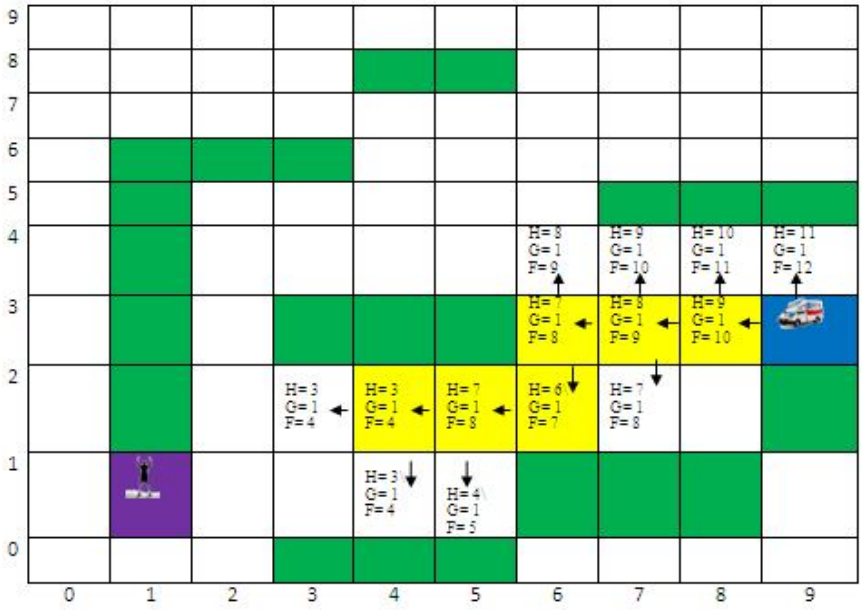
Berikut adalah tabel yang berisi penjelasan *node - node* perhitungan langkah keenam :

Tabel 3.14 penjelasan perhitungan langkah keenam

Open list	(5,1) dan (4,2)
Closed list	(9,3),(9,4), (8,3), (7,3) ,(7,4),(7,2),(6,3),(6,4),(6,2),(5,2)
Parent	(5,2)
Current node	(4,2)

Setelah melakukan pencarian dengan menggunakan langkah keenam, maka *node* tujuan sudah hampir didekati, akan tetapi perhitungan pencarian langkah selanjutnya harus tetap dihitung hingga mendapatkan *node* tujuan akhir.

Berikut adalah gambar pencarian perhitungan langkah ketujuh.



Gambar 3.13 Penjelasan Pergerakan Algoritma A* Langkah Ketujuh

Pada gambar 3.13 diatas terlihat ada 2 *node* yang masuk kedalam *Open list* yaitu *node* (4,1) dan (3,2). Dari langkah ketujuh ini didapat 2 *node* dengan nilai *f(n)* terendah adalah (4,1) dan (3,2) dengan nilai *f(n)* = 4. Seperti langkah keitga jika terdapat 2 nilai *f(n)* yang sama maka yang dipilih adalah nilai akhir yang diinputkan yaitu *node* (3,2) dengan nilai *f(n)* = 4 Berikut adalah penjelasan hasil tabel perhitungan langkah ketujuh.

Tabel 3.15 Hasil Perhitungan Langkah ke Tujuh

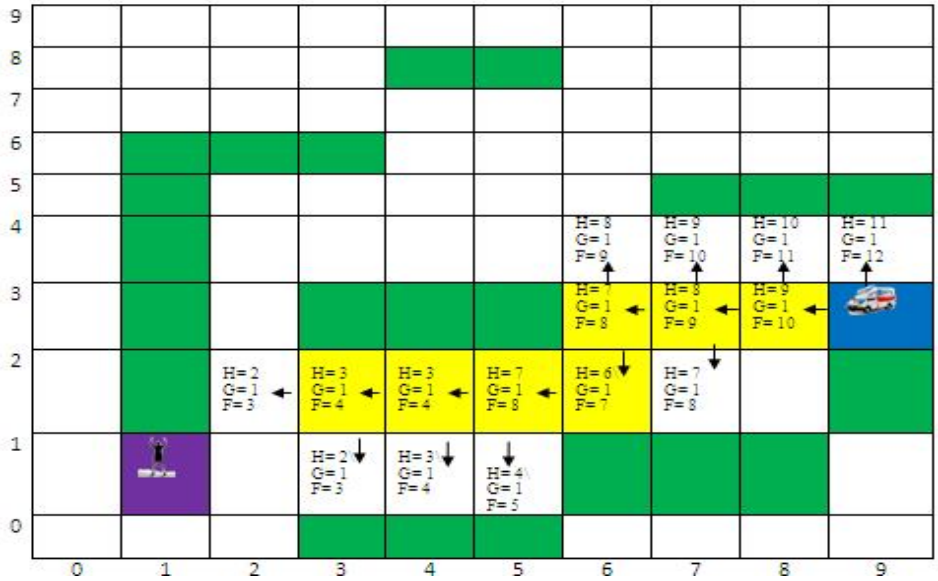
Node	Nilai *g(n)*	Nilai *h(n)*	Nilai *f(n)*
4,1	1	3	4
3,2	1	3	4

Berikut adalah tabel yang berisi penjelasan *node* - *node* perhitungan langkah ketujuh :

Tabel 3.16 Penjelasan Perhitungan Langkah Ketujuh

Open list	(4,1) dan (3,2)
Closed list	(9,3),(9,4), (8,3), (7,3) ,(7,4),(7,2),(6,3),(6,4),(6,2),(5,2),(5,1),(4,2)
Parent	(4,2)
Current node	(3,2)

Berikut adalah gambar pencarian perhitungan langkah kedelapan.



Gambar 3.14 Penjelasan Pergerakan Algoritma A* Langkah Kedelapan

Pada gambar 3.14 diatas terlihat ada 2 *node* yang masuk kedalam *Open list* yaitu *node* (3,1) dan (2,2). Dari langkah ketujuh ini didapat 2 *node* dengan nilai *f(n)* terendah adalah (3,1) dan (2,2) dengan nilai *f(n)* = 3. Seperti langkah keitga jika terdapat 2 nilai *f(n)* yang sama maka yang dipilih adalah nilai akhir yang diinputkan yaitu *node* (2,2) dengan nilai *f(n)* = 3 Berikut adalah penjelasan hasil tabel perhitungan langkah kedelapan.

Tabel 3.17 Hasil Perhitungan Langkah Kedelapan

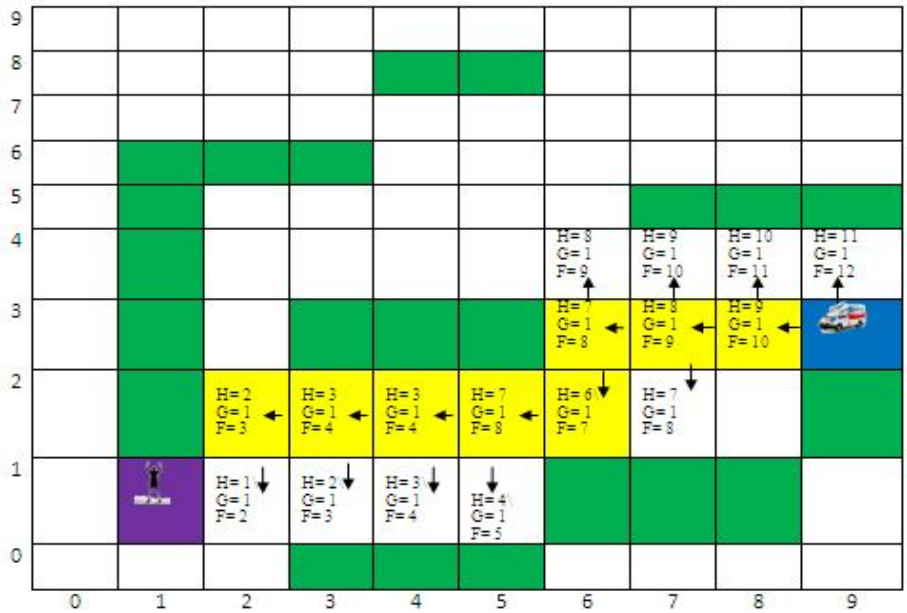
Node	Nilai *g(n)*	Nilai *h(n)*	Nilai *f(n)*
3,1	1	2	3
2,2	1	2	3

Berikut adalah tabel yang berisi penjelasan *node - node* perhitungan langkah kedelapan :

Tabel 3.18 Penjelasan Perhitungan Langkah Kedelapan

Open list	(3,1) dan (2,2)
Closed list	(9,3),(9,4), (8,3), (7,3) ,(7,4),(7,2),(6,3),(6,4),(6,2),(5,2),(5,1),(4,2), (4,1),(3,2)
Parent	(3,2)
Current node	(2,2)

Berikut adalah gambar pencarian perhitungan langkah kesembilan.



Gambar 3.15 Penjelasan Pergerakan Algoritma A* Langkah Kesembilan

Pada gambar 3.15 diatas terlihat ada 1 *node* yang masuk kedalam *Open list* yaitu *node* (2,1) dengan nilai *f(n)* = 2. Berikut adalah penjelasan hasil tabel perhitungan langkah kesembilan.

Tabel 3.19 Hasil Perhitungan Langkah kesembilan

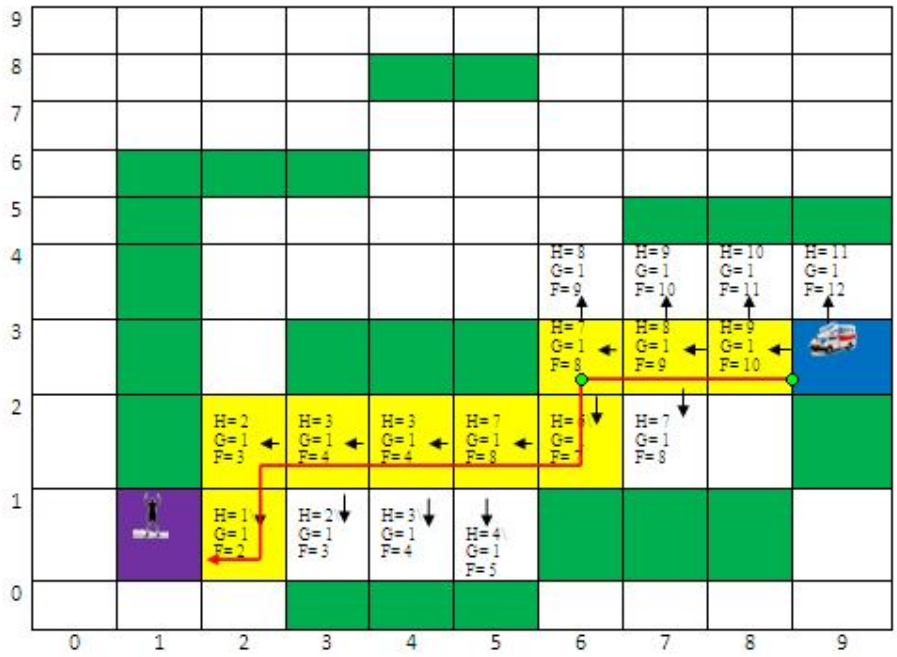
Node	Nilai *g(n)*	Nilai *h(n)*	Nilai *f(n)*
2,1	1	1	2

Berikut adalah tabel yang berisi penjelasan *node - node* perhitungan langkah kesembilan :

Tabel 3.20 Penjelasan Perhitungan Langkah Kesembilaan

Open list	(2,1)
Closed list	(9,3),(9,4), (8,3), (7,3) ,(7,4),(7,2),(6,3),(6,4),(6,2),(5,2),(5,1),(4,2), (4,1),(3,2),(3,1),(3,2),(3,1),(2,2)
Parent	(2,2)
Current node	(2,1)

Untuk menentukan cara menemukan jalan terbaik, langkah yang diperlukan adalah melakukan pemeriksaan dimulai dari *node* target atau *node* terakhir, dimana dalam ilustrasi pada gambar disini menggunakan tanda yang diberi warna unggu. Dengan mengikuti arah *Pointer* yang dapat membawa kembali ke jalan *node* awal, maka akan didapatkan jalan terbaik. Sehingga jika diimplemetasikan kedalam sebuah *Game*, pergerakan pencarian jalur ambulance adalah seperti Gambar 3.16 berikut ini



Gambar 3.16 Penjelasan Pergerakan Algoritma A* Langkah Terakhir

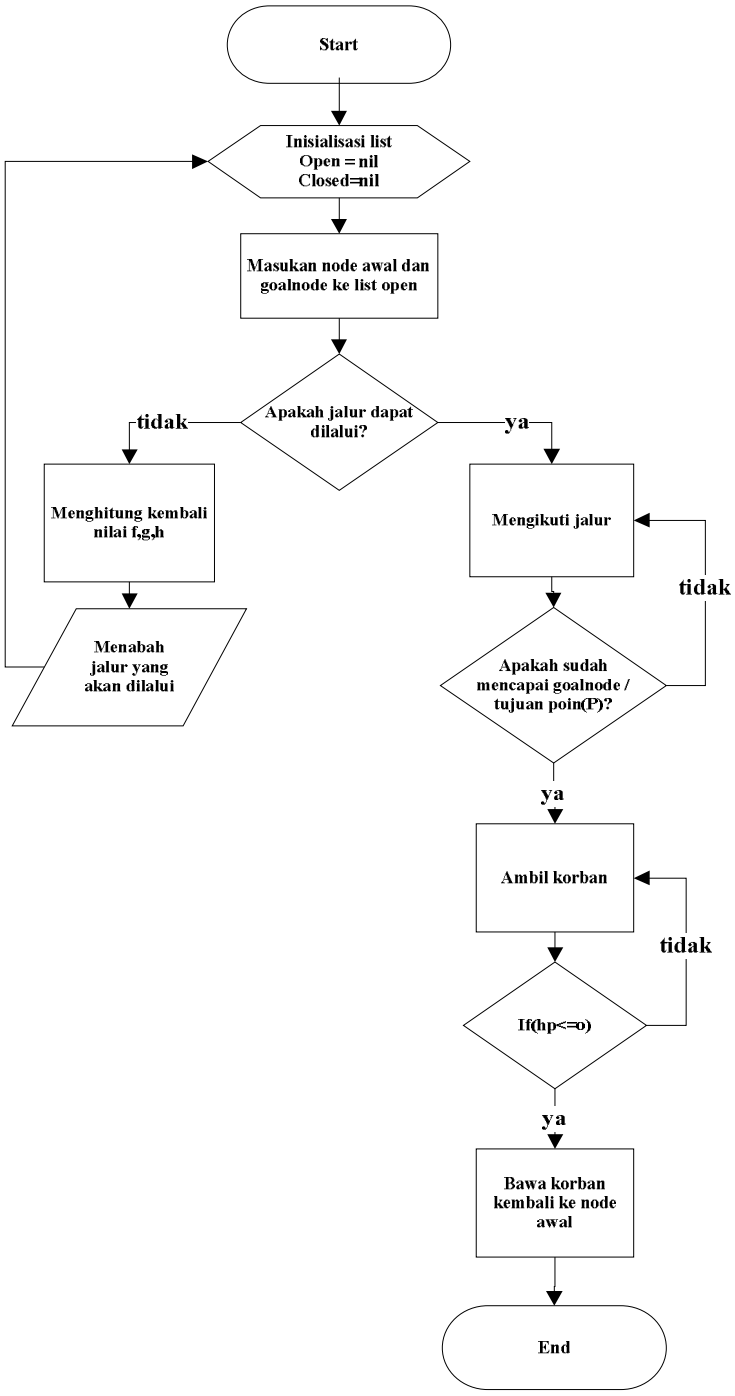
Tabel berikut adalah penjelasan mengenai pencarian jalur terpendek pada langkah terakhir.

Tabel 3.21 Penjelasan Perhitungan Langkah Terakhir

Open list	(1,1)
Closed list	(9,3),(9,4), (8,3), (7,3) ,(7,4),(7,2),(6,3),(6,4),(6,2),(5,2),(5,1),(4,2), (4,1),(3,2),(3,1),(3,2),(3,1),(2,2),(2,1)
Parent	(2,1)
Current node	(1,1)

3.1.11.1 Flowchart Procedure pencarian jalur

Gambar dibawah ini akan menjelaskan alur pencarian jalur terpendek ambulance dalan menemukan korban yang sudah ditemukan oleh pemain..



Gambar 3.17 Flowchart Pencarian Jalur

3.1.12 Analisis Kebutuhan Non fungsional

Analisis kebutuhan non fungsional ini meliputi elemen – elemen atau komponen yang diperlukan untuk membuat aplikasi *Game* Pertolongan Pertama. Kebutuhan non funsional terbagi menjadi tiga bagian yaitu : analisis perangakat keras, analisis perangkat lunak, dan analisis pengguna (*user*) aplikasi pada *Game* ini.

3.1.13 Analisis Perangkat Keras

Perangkat keras atau hardware merupakan salah satu komponen yang penting pada aplikasi ini. Karena tanpa hardware yang memenuhi syarat, program aplikasi yang akan dibuat tidak akan berjalan dengan baik. Berikut spesifikasi standar perangkat keras dan spesifikasi yang dapat dipergunakan untuk membangun aplikasi pada *Game* ini yaitu :

1. *Processor dengan kecepatan 2.0 Ghz*
2. *RAM* 2 GB
3. *Hardisk* dengan ruang kosong minimal 3 GB
4. *Vga Card 512MB*
5. *Monitor* (resolusi 1024 x 768)
6. *Mouse*
7. *Keyboard*
8. *Speaker*

Berikut spesifikasi standar perangkat keras dan spesifikasi yang dapat dipergunakan untuk menjalankan aplikasi pada *Game* ini yaitu :

1. *Processor dengan kecepatan 1.8 Ghz*
2. *RAM 1 GB*
3. *Hardisk* dengan ruang kosong 500 MB
4. *Vga Card 512MB*
5. *Monitor* (resolusi 1024 x 768)
6. *Mouse*
7. *Keyboard*
8. *Speaker*

3.1.14 Analisis Perangkat Lunak

Perangkat lunak atau *software* merupakan hal terpenting dalam mendukung kinerja sebuah sistem. Perangkat lunak digunakan dalam sebuah sistem untuk mengeksekusi atau menjalankan perintah – perintah yang akan diberikan kepada perangkat keras agar dapat saling berinteraksi satu sama lain.

Perangkat lunak yang dibutuhkan pengguna untuk menjalankan apikasi *Game* Pertolongan Pertama ini adalah sebagai berikut :

1. Sistem Operasi Windows (XP dengan service pack 2, windows 7, dan sistem operasi diatasnya)
2. Flash *Player* 10.

3.1.15 Analisis Pengguna (*User*)

Analisis pengguna merupakan analisis beberapa parameter terhadap calon pengguna yang akan menggunakan aplikasi ini, berikut parameter yang digunakan untuk menganalisis **:**

1. *Users knowledge and experience* dari target *user* yang akan mnggunakan aplikasi ini

Tabel 3.22Analisis Klasifikasi *Knowledge and Experience*

Educational level	***Reading Level***	***Typing Skills***
Game ini dapat digunakan oleh siswa- siswi atau anak – anak berumur 10 – 15 tahun atau usia sekolah tingkat sd dan smp	*Game* ini dapat digunakan oleh siswa-siswi kelas atau anak – anak berumur 10 – 15 tahun atau usia sekolah tingkat sd dan smp dengan r*eading level* sedang.	*Game* ini tidak memerlukan skill yang tinggi. Namun diharapkan pengguna sudah terbiasa menggunakan perangkat komputer dan fungsi keyboard
Komputer Literacy	***Task Experience***	***System Experience***
Game ini dapat digunakan oleh pengguna yang memiliki kemampuan komputer yang sedang.	*Game* ini dapat digunakan oleh pengguna yang memiliki pengalaman penggunaan komputer dan *Game* yang mudah.	*Game* ini dapat digunakan oleh pengguna yang memiliki pengalaman penggunaan komputer dan *Game* yang mudah.
Application Experience	***Native Language***	***Use Of Other System***
Game inidapat digunakan dalam sistem operasi Windows xp, 7, dan sistem operasi diatasnya.	*Game* ini menggunakan satu bahasa yaitu bahasa Indonesia.	*Game* ini dapat dijalankan dengan terlebih dahulu menginstalkan sistem operasi windows dan flash *Player* 10

2. *Users Physical Characteristic*

Keadaan fisik seseorang memungkinkan berpengaruh pada penggunaan aplikasi *Game* ini. Ada hal-hal yang harus diperhatikan terhadap *user* dari karakteristik fisiknya untuk dapat menggunakan aplikasi adalah *Handedness, Color Blindness, dan Gender*. *Handedness* adalah kemudahan dalam menggunakan *keyboard* dan input device (tangan kanan, tangan kiri). Color *Blindness* adalah warna pada layar. Gender adalah aplikasi yang dibuat untuk wanita atau pria.

Tabel 3.23 *Users Physical Characteristic*

Age	10 - 15 tahun
Gender	Pria dan Wanita
Handedness	Kanan dan Kiri
Color Blind	Pengguna (*User*) yang tidak dapat membedakan warna yang satu dengan yang lainnya (butawarna) menungkinkan dapat menggunakan aplikasi ini, karena dalam karakter yang dibuat dapat dikenali dengan mudah melalui bentuknya. Akan tetapi penggunaannya tidak akan optimal karena dalam *Game* ini terdapat banyak sekali perbedaan warna yang menunjang interaksi dan ketertarikan.

3.1.16 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan untuk mengetahui gambaran sistem yang akan dibangun. Gambaran tersebut meliputi : pendefinisian fungsi *Game,* penggambaran alur aktivitas pada *Game*, pengambaran interaksi antar objek pada *Game* dan *class diagram.*

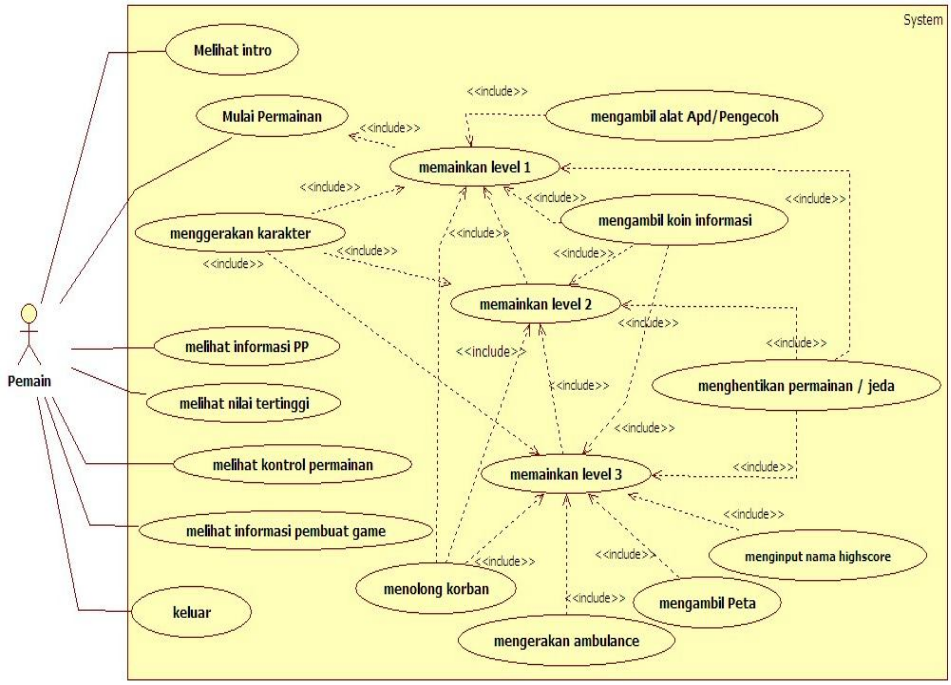
3.1.17 Pendefinisian Fungsi *Game*

Pendefinisian fungsi *Game* digunakan sebagai gambaran pada fungsi – fungsi pada *Game* yang akan dibangun. Pendefinisian pada fungsi *Game* ini digambarkan dengan menggunakan *Use case diagram*, definisi actor, definisi *Use case*, dan skenario *Use case.*

3.1.17.1 *Use case* Diagram

Use case diagram merupakan pengambaran fungsi pada *Game* berdasarkan interaksi antara actor dengan objek yang ada pada sebuah sistem. Berikut dibawah ini adalah *Use case* yang digunakan pada *Game* Pertolongan Pertama.

1. *Use case* diagram *Game* pertolongan pertama



Gambar 3.18 *Use case* Diagram *Game* Pertolongan Pertama

3.1.17.2 Definisi Actor

Definisi actor berfungsi untuk menjelaskan apa saja yang dapat dilakukan actor yang terdapat pada *Use case* diagram.

Tabel 3.24 Definisi Actor

No	Actor	Deskripsi
1	Pemain	Pemain merupakan pengguna / orang / *user* yang memainkan *Game* dan berinteraksi langsung dengan antarmuka *Game*.

3.1.17.3 Definisi *Use case*

Definisi *Use case* berfungsi untuk menjelaskan setiap proses dan objek yang terdapat pada setiap *Use case*.

1. Definisi *use case*

   Berikut adalah tabel 3.25 definisi *use case Game* pertolongan pertama.

Tabel 3.25 Definisi *Use case Game* Pertolongan Pertama

No	*Use case*	Deskripsi
1.	Melihat intro	Proses untuk melihat animasi intro *Game*
2.	Mulai Permainan	Proses untuk memulai permainan
3.	Memainkan level 1	Proses memainkan level 1
4.	Mengambil alat APD/pengecoh	Proses pengambilan alat APD/ pengecoh oleh pemain
5.	Menolong korban	Proses yang digunakan untuk menolong korban
6.	Mengambil koin informasi	Proses yang digunakan untuk mengambil koin dan menampilkan isi informasi dari koin.
7.	Memainkan level 2	Proses memainkan level 2
8.	Memainkan level 3	Proses memainkan level 3
9.	Mengambil peta	Proses mengambil Peta
10.	Mengerakan ambulance	Proses pegerakan ambulance menjemput korban
11.	Menginput nama highscore	Proses menginputkan nama pemain untuk ditampilkan pada menu nilai tertinggi
12.	Pergerakan karakter	Proses menggerakan karakter permainan
13.	Memberhentikan Permainan / jeda	Proses untuk menghentikan permainan /jeda
14.	Melihat informasi PP	Proses yang digunakan untuk melihat informasi mengenai pertolongan pertama.
15.	Melihat informasi kontrol permainan	Proses yang digunakan untuk melihat informasi control permainan
16.	Melihat informasi nilai tertinggi	Proses yang digunakan untuk melihat informasi nilai tertinggi dari semua pemain
17.	Melihat informasi pembuat *Game*	Proses untuk melihat informasi mengenai pembuat *Game*
18.	Keluar	Proses yang digunakan untuk mengakhiri permainan dan keluar dari aplikasi.

3.1.17.4 Skenario *Use case*

Skenario *Use case* menunjukan proses apa saja yang terjadi pada setiap bagian di dalam *Use case* tersebut, dimana pengguna / *user* memberikan perintah pada bagian – bagian *Use case* dan respon apa yang diberikan oleh sistem kepada pengguna / *user* setelah pengguna / *user* memberikan perintahnya.

1. Skenario *Use case* melihat Intro Permainan

Tabel 3.26 *Skenario Use case* Melihat Intro

***Use case* Name : Melihat Intro**
Requirements explored : 1
***Player* (actor)context (role) :** Pemain
Preconditions :
Triggers : Pemain membuka aplikasi
Main course of Action : 1. Sistem akan menampilkan animasi intro 2. Pemain melihat intro sampai akhir 3. Sistem tetap menjalankan intro animasi sampai akhir
Alternative courses of action : 2a. pemain tidak melihat intro sampai akhir / memilih tombol skip 3a. sistem menutup intro dan menampilkan menu utama
Exceptional courses of action : 1a. sistem error / *crash* 2b. pemain mematikan computer

2. Skenario *Use case* Mulai Permainan

Tabel 3.27*Use case* Skenario Mulai Permainan

***Use case* Name : Mulai Permainan**
Requirements explored : 2
***Player* (actor)context (role) :** Pemain
Preconditions : aplikasi *Game* sudah berjalan
Triggers : pemain memilih dan mengklik tombol Mulai Permainan
Main course of Action : 1. Sistem akan menampilkan loading / memuat data 2. Pemain memulai permainan
Alternative courses of action : 1a. sistem tidak menampilkan loading 1b. pemain tetap berada di menu utama
Exceptional courses of action : 1a. sistem error / *crash* 2b. pemain mematikan computer

3. Skenario *Use case* memainkan level 1

Tabel 3.28 Skenario *Use case* Memainkan Level 1

Use case Name : memainkan level 1
Requirements explored : 3
Player (actor)context (role) : Pemain
Preconditions : aplikasi *Game* sudah berjalan
Triggers : waktu permainan dimulai ketika pemain mengklik tombol main
Main course of Action : 1. Sistem menampilkan intruksi permainan level 1 2. Pemain menekan tombol panah (main). 3. Sistem menampilkan permainan level 1 4. Sistem menginisialisasi posisi pemain 5. Sistem mensetting jumlah darah 6. Sistem menginisialisasi posisi alat 7. Sistm menginisialisasi posisi koin informasi. 8. Sistem menginisialisasi posisi musuh / NPC / penghalang 9. Sistem menginisialisasi posisi korban 10. Sistem memulai permainan level 1 11. Pemain memainkan permainan level 1.

Tabel 3.28 Scenario *Use case* Level 1

Main course of Action : 12. Pemain mengendalikan karakter permainan 13. Sistem menggerakkan karakter 14. Pemain berhasil menjalankan misi 15. Sistem menampilkan score pemain.
Alternative courses of action : 1. Pemain mengenai penghalang / musuh / gagal menolong 2. Sistem mengurangi darah pemain 14a Pemain gagal melaksanakan misi 15a. Sistem menampilkan pesan gagal
Exceptional courses of action : 1. Sistem error 2. Pemain menutup paksa aplikasi 3. Pemain mematikan computer

4. Scenario mengambil alat APD

Tabel 3.29 Scenario *Use case* Mengambil Alat APD

***Use case* Name :** mengambil alat APD/pengecoh
Requirements explored : 4
***Player* (actor)context (role) :** Pemain
Preconditions : aplikasi *Game* sudah berjalan
Triggers : Pemain Berada pada permainan level 1
Main course of Action : 1. Pemain menggerakan karakter ke posisi alat APD 2. Sistem menampilkan informasi alat APD yang sudah diambil 3. Sistem mengurangi jumlah alat yang harus dikumpulkan 4. Sistem menambahkan score jika alat yang diambil merupakan alat apd
Alternative courses of action : 1. pemain mengerakan karakter ke alat pengecoh. 2 Ssistem menampilkan informasi alat pengecoh 3. Sistem tidak menambahkan score jika alat yang diambil merupakan alat pengecoh
Exceptional courses of action : 4. Sistem error 5. Pemain menutup paksa aplikasi 6. Pemain mematikan computer

5. Scenario *use case* menolong korban

Tabel 3.30 Scenario *Use case* Menolong Korban

***Use case* Name :** menolong korban
Requirements explored : 5
***Player* (actor)context (role) :** Pemain
Preconditions : aplikasi *Game* sudah berjalan
Triggers : pemain pada permainan level 1
Main course of Action : 1. Pemain menggerakan karakter ke posisi korban 2. Sistem menampilkan stage pengobatan 3. Sistem menampilkan intruksi pengobatan 4. Pemain memilih alat pengobatan 5. Sistem menampilkan hasil pengobatan dan skor
Alternative courses of action : 1. pemain salah memilih alat pengobatan 2. sistem mengurangi darah pemain
Exceptional courses of action : 1. Sistem error 2. Pemain menutup paksa aplikasi 3. Pemain mematikan computer

6. Scenario *use case* mengambil koin informasi

Tabel 3.31 Scenario *Use case* Mengambil Koin Informasi

***Use case* Name :** mengambil koin informasi
Requirements explored : 6
***Player* (actor)context (role) :** Pemain
Preconditions : aplikasi *Game* sudah berjalan
Triggers : pemain berada pada permainan level1
Main course of Action : 1. Pemain menggerakan karakter ke posisi koin 2. Sistem menampilkan informasi koin
Alternative courses of action : -
Exceptional courses of action : -

7. Skenario *Use case* memainkan level 2

Tabel 3.32 *use case* memainkan level 2

***Use case* Name :** memainkan level 2
Requirements explored : 7
***Player* (actor)context (role) :** Pemain
Preconditions : *Game* dimulai
Triggers : waktu permainan dimulai ketika pemain mengklik tombol main
Main course of Action : 1. Sistem menampilkan intruksi permainan level 2 2. Pemain menekan tombol panah (main) 3. Sistem menampilkan permainan level 2 4. Sistem menginisialisasi posisi pemain 5. Sistm mensetting jumlah darah pmain 6. Sistem menginisialisasi posisi koin informasi 7. Sistem menginisialisasi posisi korban 8. Sistem menginisialisasi posisi musuh / npc / penghalang 9. Pemain memainkan permainan level 2 10. Pemain mengendalikan karakter permainan 11. Sistem menggerakkan karakter 12. Pemain berhasil menjalankan misi 13. Sistem menampilkan score pemain
Alternative courses of action : 3. Pemain mengenai penghalang / musuh / gagal menolong 4. Sistem mengurangi darah pemain 5. Pemain gagal melaksanakan misi 6. Sistem menampilkan pesan gagal
Exceptional courses of action : 1. Sistem error 2. Pemain mematikan computer

8. Skenario *Use case* memainkan level 3

Tabel 3.33 *use case* memainkan level 3

Use case Name : memainkan level 3
Requirements explored : 8
Player (actor)context (role) : Pemain
Preconditions : aplikasi *Game* sudah berjalan
Triggers : waktu permainan dimulai ketika pemain mengklik tombol main
Main course of Action : 1. Sistem menampilkan intruksi permainan level 3 2. Pemain menekan tombol panah (main). 3. Sistem menampilkan permainan level 3 4. Sistem menginisialisasi posisi pemain 5. Sistem mensetting jumlah darah 6. Sistm menginisialisasi posisi koin informasi. 7. Sistem menginisialisasi posisi korban 8. Sistem menginisialisasi posisi ambulance 9. Sistem menginisialisasi posisi peta. 10. Pemain memainkan permainan level 3 11. Pemain mengendalikan karakter permainan 12. Sistem menggerakkan karakter 13. Pemain berhasil menjalankan misi 14. Sistem menampilkan score pemain.
Alternative courses of action : 1. Pemain gagal melaksanakan misi 2. Sistem menampilkan pesan gagal
Exceptional courses of action : 1. Sistem error 2. Pemain menutup paksa aplikasi 3. Pemain mematikan computer

9. Scenario *use case* mengambil peta

Tabel 3.34 Scenario *Use case* Mengambil Peta

***Use case* Name :** mengambil peta
Requirements explored : 9
***Player* (actor)context (role) :** Pemain
Preconditions : aplikasi *Game* sudah berjalan
Triggers : pemain berada pada permainan level 3
Main course of Action : 1. Pemain menggerakan karakter ke posisi peta 2. Sistem mengurangi / menghilangkan peta
Alternative courses of action : -
Exceptional courses of action : 1. Sistem error 2. Pemain menutup paksa aplikasi 3. Pemain mematikan computer

10. Skenario *Use case* mengerakan ambulance

Tabel 3.35 Skenario *Use case* Mengerakan Ambulance

***Use case* Name : A**mbulance
Requirements explored : 10
***Player* (actor)context (role) :** Pemain
Preconditions : aplikasi *Game* sudah berjalan
Triggers : pemain berada pada permainan level 3
Main course of Action : 1. Pemain sudah menemukan korban dan memangill bantuan 2. Sistem menjalankan ambulance menjemput korban 3. Sistem menjalankan ambulance kemabli ke rumah sakit
Alternative courses of action : -
Exceptional courses of action : 1. Sistem error 2. Pemain menutup paksa aplikasi 3. Pemain mematikan computer

11. Skenario *Use case* menginput nama highscore

Tabel 3.36 Scenario *Use case* Menginputkan Nama

***Use case* Name :** menginput nama highscore
Requirements explored : 11
***Player* (actor)context (role) :** Pemain
Preconditions : pemain memainkan level 3
Triggers : pemain menyelesaikan level 3
Main course of Action : 1. Sistem menyimpan score pemain 2. Sistem menampilkan *Game* selesai 3. Sistem menampilakan halaman pesan isi nama 4. Pemain menginputkan nama nya dengan memberi instruksi inputan pada keyboard dan klik simpan 5. Sistem menyimpan nama pemain
Alternative courses of action : -
Exceptional courses of action : -

12. Skenario *Use case* pergerakan karakter

Tabel 3.37 Scenario *Use case* Pegerakan Karakter

***Use case* Name :** pergerakan karakter
Requirements explored : 12
***Player* (actor)context (role) :** Pemain
Preconditions : *Game* dimulai
Triggers : pemain berada di arena permainan
Main course of Action : 6. Pemain menekan tombol panah kanan pada keyboard 7. Sistem meggerakan karakter maju kedepan atau arah kanan
Alternative courses of action : 1a. Pemain menekan tombol panah kiri pada keyboard 2a. Sistem meggerakan karakter maju kebelakang atau arah kiri 1b. Pemain menekan tombol panah atas pada keyboard 2b. Sistem meggerakan karakter dengan lompatan / arah atas 1c. Pemain menekan tombol panah bawah pada keyboard 2c. Sistem meggerakan karakter ke arah bawah
Exceptional courses of action : -

13. Skenario *Use case* menghentikan permainan / jeda

Tabel 3.38 Skenario *Use case* Menghentikan Permainan / jeda

***Use case* Name : M**enghentikan permainan / jeda
Requirements explored : 13
***Player* (actor)context (role) :** Pemain
Preconditions : *Game* dimulai
Triggers : pemain berada di arena permainan
Main course of Action : 1. Pemain menekan tombol menu pause 2. Sistem menampilkan menu jeda 3. Pemain memilih menu lanjutkan 4. Sistem akan melanjutkan permainan
Alternative courses of action : 3b. pemain memilih menu keluar 4b. sistem akan menampilkan menu utama Pemain memilih menu keluar
Exceptional courses of action : -

14. Skenario *Use case* melihat informasi PP

Tabel 3.39 Scenario *Use case* Melihat Informasi PP

***Use case* Name :** melihat informasi PP
Requirements explored : 14
***Player* (actor)context (role) :** Pemain
Preconditions : pemain membuka aplikasi
Triggers : pemain berada di menu utama permainan
Main course of Action : 1. Pemain memilih tombol menu informasi PP 2. Sistem menampilkan layar informasi seputar pertolongan pertama
Alternative courses of action : 1a. Pemain tidak memilih menu informasi PP 1b. sistem tetap berada di menu utama permainan dan tidak akan menampilkan informasi PP
Exceptional courses of action :

15.

16. Skenario *Use case* melihat informasi kontrol permainan

Tabel 3.40 Scenario *Use case*Mmelihat Informasi Kontrol Permainan

***Use case* Name :** melihat informasi kontrol permainan
Requirements explored : 15
***Player* (actor)context (role) :** Pemain
Preconditions : pemain membuka aplikasi
Triggers : pemain berada di menu utama permainan
Main course of Action : 1. Pemain memilih tombol menu Kontrol permainan 2. Sistem menampilkan petunjuk cara kontrol permainan
Alternative courses of action : 1a. Pemain tidak memilih menu control permainan 1b. sistem tetap berada di menu utama permainan dan tidak akan menampilkan informasi control permainan
Exceptional courses of action :

17. Skenario *Use case* melihat informasi nilai tertinggi

Tabel 3.41 Scenario *Use case* Melihat Informasi Nilai Tertinggi

***Use case* Name :** melihat informasi nilai tertinggi
Requirements explored : 16
***Player* (actor)context (role) :** Pemain
Preconditions : pemain membuka aplikasi
Triggers : pemain berada di menu utama permainan
Main course of Action : 3. Pemain memilih tombol menu nilai tertinggi 4. Sistem menampilkan informasi nilai tertinggi
Alternative courses of action : 1a. Pemain tidak memilih menu nilai tertinggi 1b. sistem tetap berada di menu utama permainan dan tidak akan menampilkan informasi nilai tertinggi
Exceptional courses of action :

18. Skenario *Use case* melihat informasi pembuat *Game*

Tabel 3.42 Scenario *Use case* Melihat Informasi Pembuat *Game*

***Use case* Name :** melihat informasi pembuat *Game*
Requirements explored : 16
***Player* (actor)context (role) :** Pemain
Preconditions : pemain membuka aplikasi *Game*
Triggers : pemain berada di menu utama permainan
Main course of Action : 5. Pemain memilih tombol menu informasi pembuat *Game* 6. Sistem menampilkan informasi pembuat *Game*
Alternative courses of action : 1a. Pemain tidak memilih menu pembuat *Game* 1b. sistem tetap berada di menu utama permainan dan tidak akan menampilkan informasi pembuat *Game*
Exceptional courses of action :

19. Skenario *Use case* keluar aplikasi

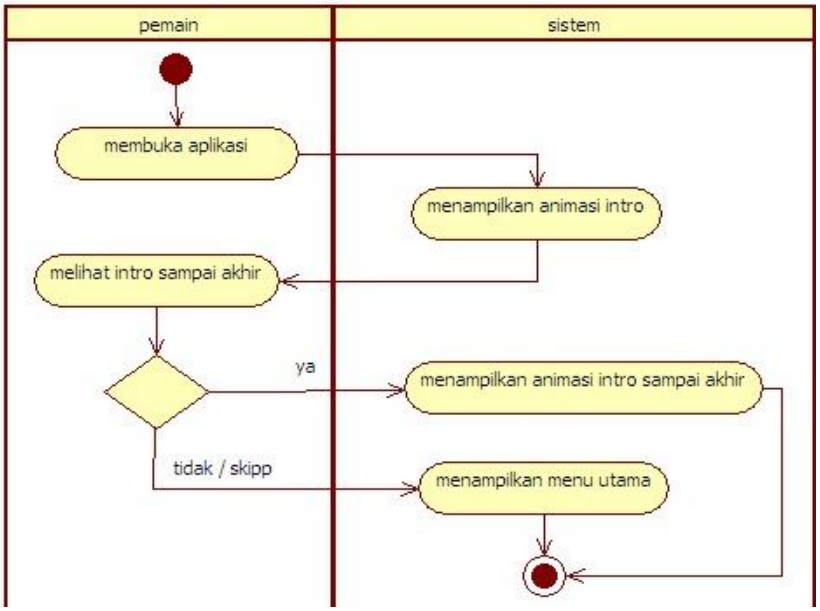
Tabel 3.43 Scenario *Use case* Keluar Aplikasi

***Use case* Name :** Keluar aplikasi
Requirements explored : 18
***Player* (actor)context (role) :** Pemain
Preconditions : pemain membuka aplikasi
Triggers : pemain berada di menu utama permainan
Main course of Action : 1 .Pemain memilih menu tombol keluar 2. Sistem menampilkan pertanyaan yakin keluar atau tidak.
Alternative courses of action : 1a. Pemain tidak memilih menu tombol keluar 1b. sistem tetap berada di menu utama permainan
Exceptional courses of action :

3.1.18 Penggambaran Alur Aktifitas pada *Game*

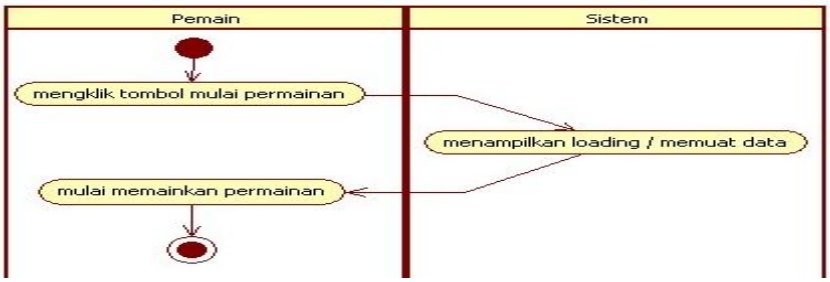
Pengambaran alur aktivitas pada *Game* menggambarkan berbagai alur aktivitas dalam *Game* yang sedang dirancang, bagaimana masing – masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Pengambaran alur aktivitas pada *Game* digambarkan menggunakan *Activity Diagram*. *Activity Diagram* merupakan state diagram khusus, dimana sebagian besar state adalah action dan sbagian besar transisi di *trigger* oleh selesainya state sbelumnya *(internal Processing).* Oleh karena itu *Activity Diagram* tidak menggambarkan behavior internal sebuah sistem dan interaksi antar subsistem secara eksak, tetapi lebih menggambarkan proses – proses dan jalur – jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *Use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *Use case* menggambarkan bagaimana actor menggunakan sistem untuk melakukan aktivitas. Berikut adalah *Activity Diagram* yang digunakan pada *Game* Pertolongan Pertama :

3.1.18.1 *Activity Diagram* Melihat Intro



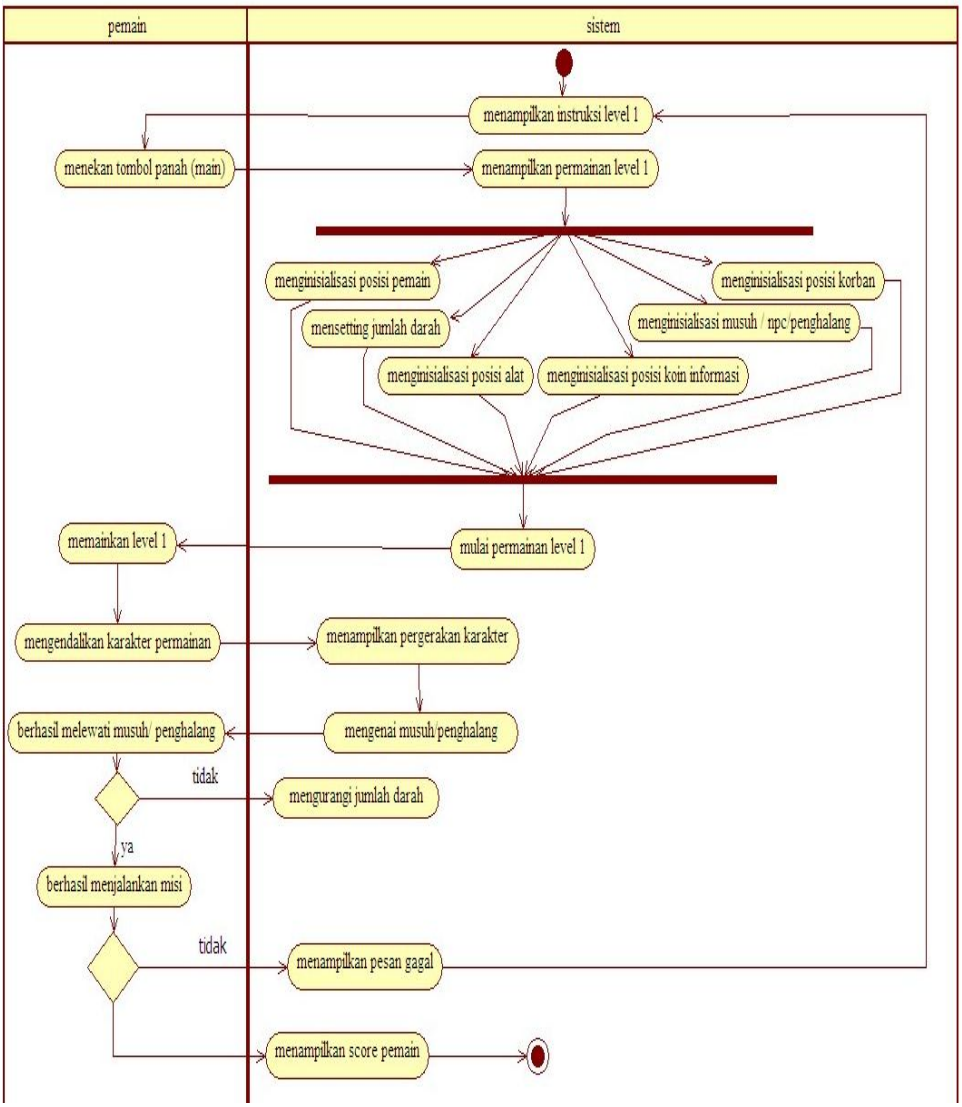
Gambar 3.19 *Activity Diagram* Melihat Intro

3.1.18.2 *Activity Diagram* mulai Permainan



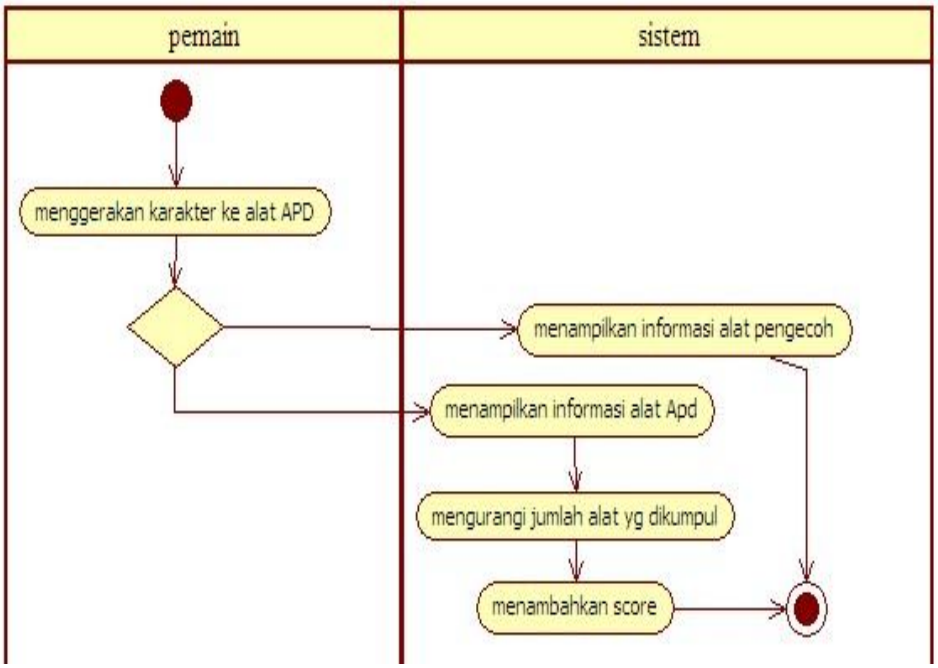
Gambar 3.20 *Activity Ddiagram* Mulai Permainan

3.1.18.3 *Activity Diagram* Memainkan Level 1



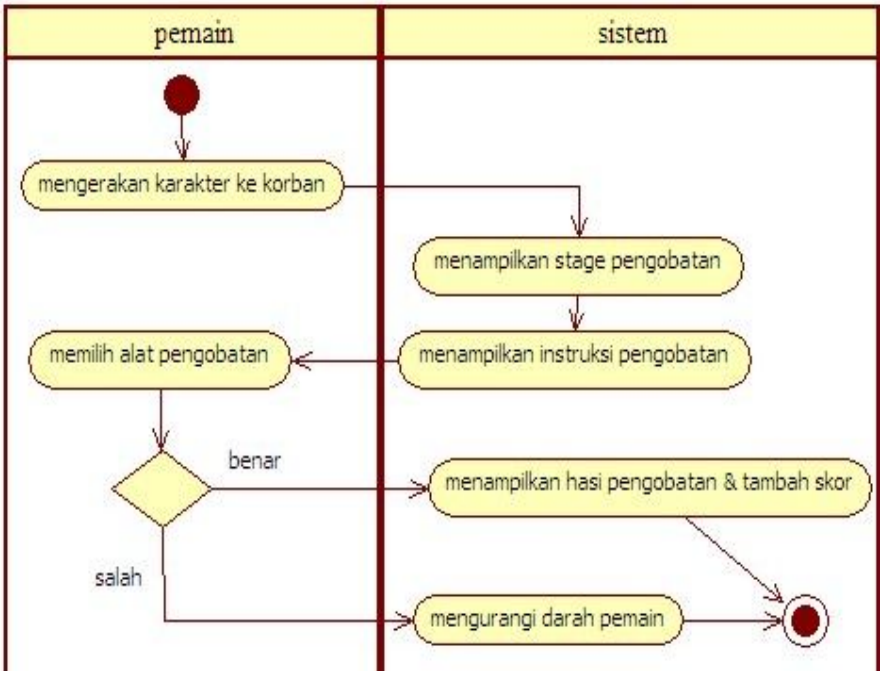
Gambar 3.21 *Activity Diagram* Memainkan Level 1

3.1.18.4 *Activity Diagram* Mengambil Alat



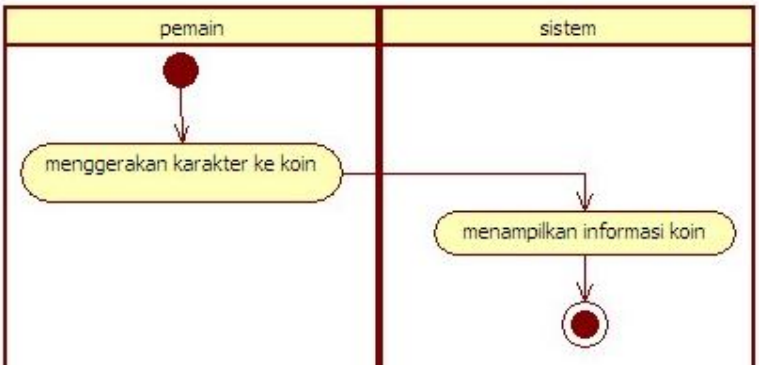
Gambar 3.22 *Activity Diagram* Mengambil Alat

3.1.18.5 *Activity Diagram* Menolong Korban



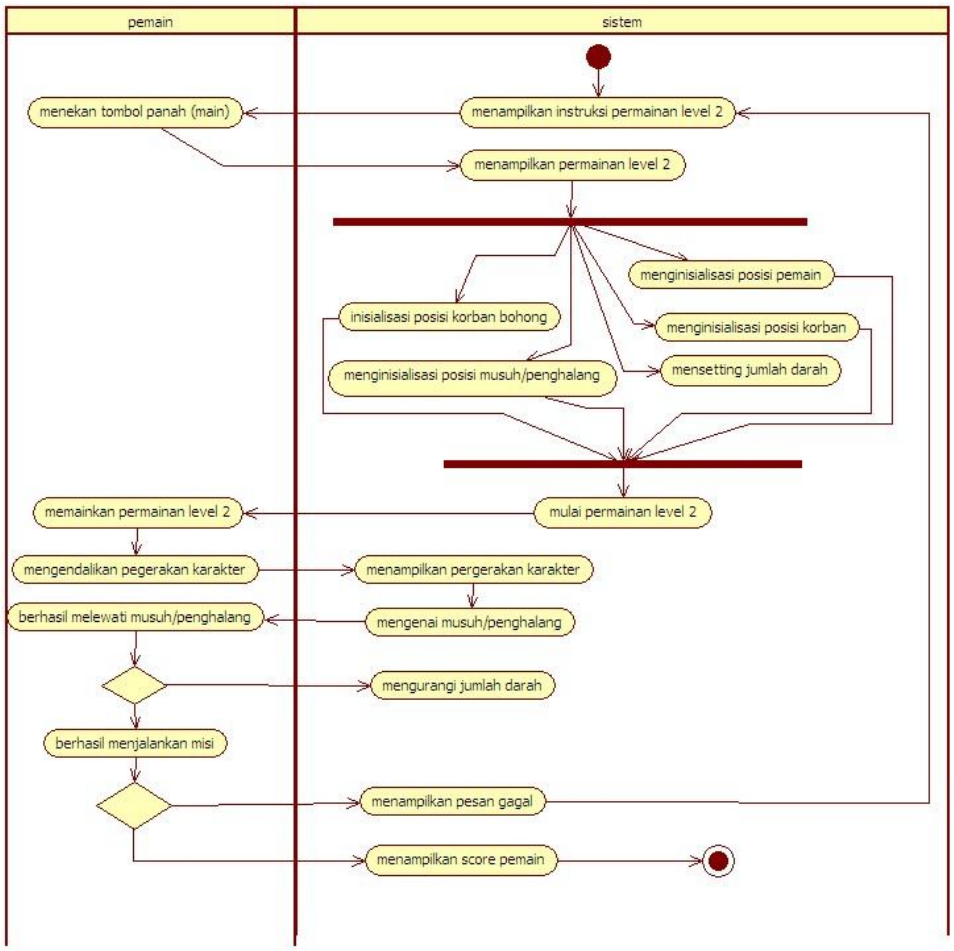
Gambar 3.23 *Activity Diagram* Menolong Korban

3.1.18.6 *Activity Diagram* Mengambil Koin Informasi



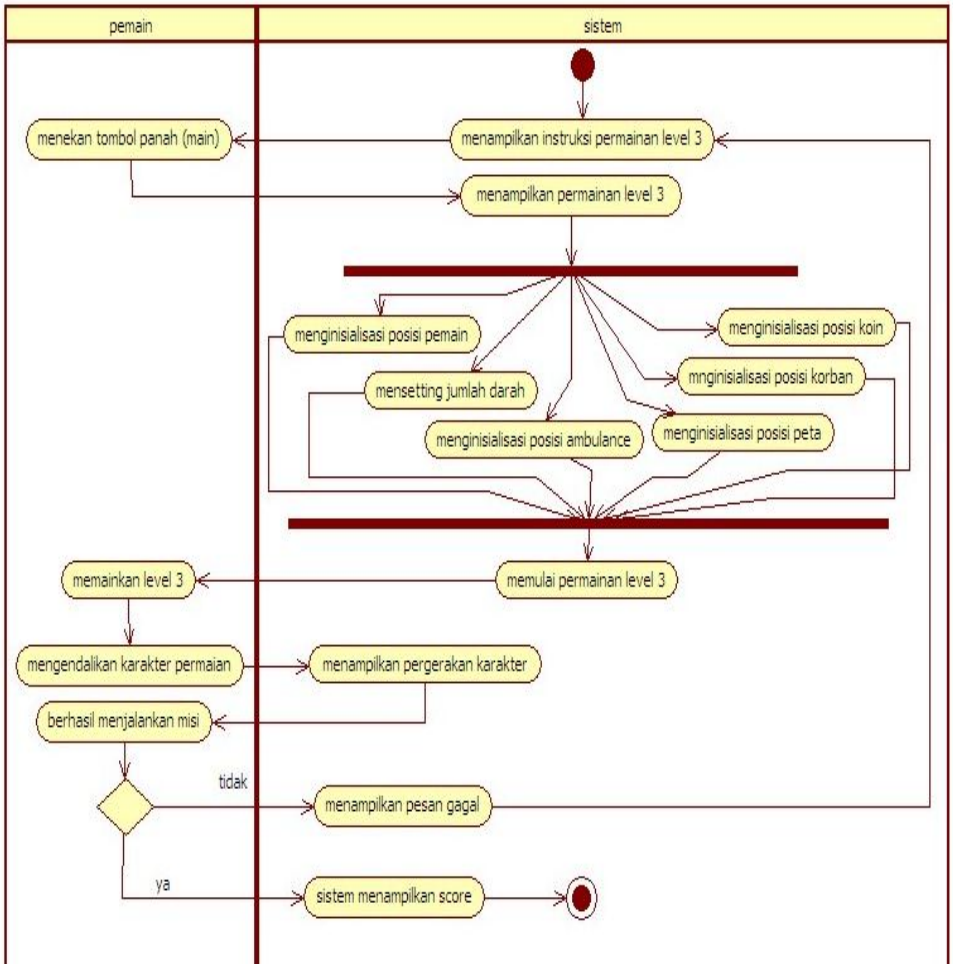
Gambar 3.24 *Activity Diagram* Ambil Koin Informasi

3.1.18.7 *Activity Diagram* Memainkan Level 2



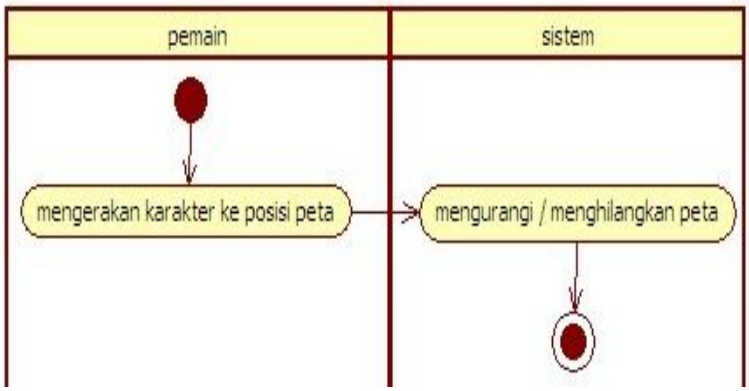
Gambar 3.25 *Activity Diagram* Memainkan Level 2

3.1.18.8 *Activity Diagram* Memainkan Level 3



Gambar 3.26 *Activity Diagram* Level 3

3.1.18.9 *Activity Diagram* Mengambil Peta



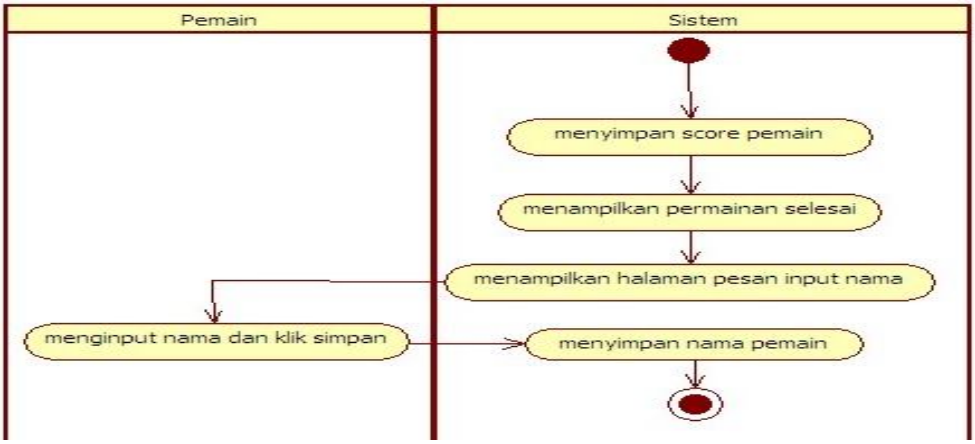
Gambar 3.27 *Activity Diagram* Mengambil Peta

3.1.18.10 *Activity Diagram* Ambulance



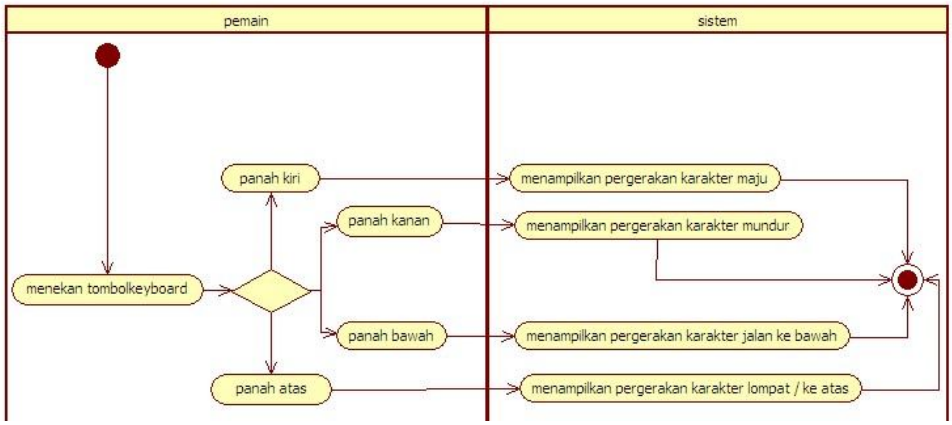
Gambar 3.28 *Activity Diagram* Ambulance

3.1.18.11 *Activity Diagram* Input Nama Highscore



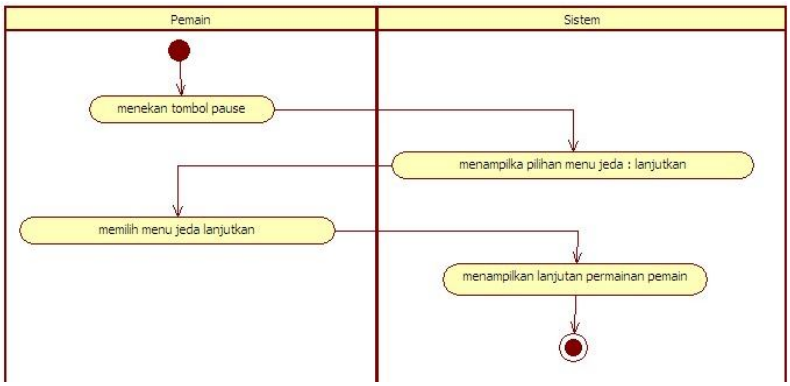
Gambar 3.29 *Activity Diagram* Input Nama Highscore

3.1.18.12 *Activity Diagram* Pergerakan Karakter



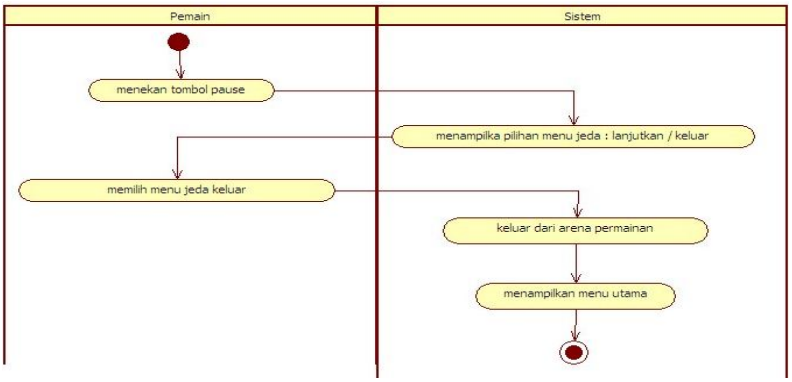
Gambar 3.30 *Activity Diagram* Pergerakan Karakter

3.1.18.13 *Activity Diagram* Menghentikan Permainan/ jeda



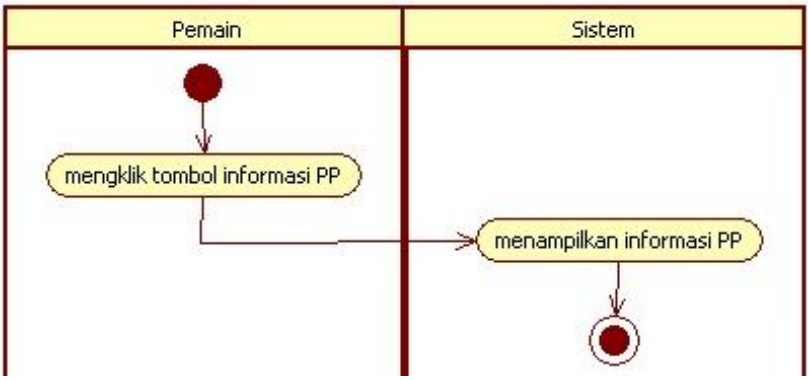
Gambar 3.31 *Activity Diagram* Menghentikan Permainan / jeda dengan pilihan Menu Lanjutan

Alternatife diagram menghentikan permainan / jeda dengan pilihan “keluar”.



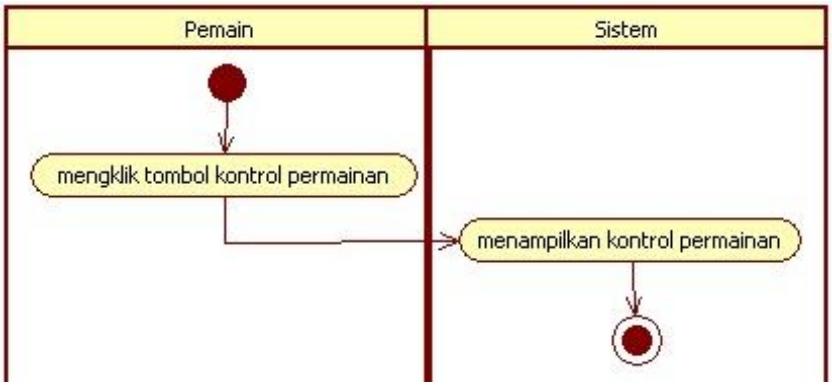
Gambar 3.32 Alternatif *Activity Diagram* Menghentikan Permainan / jeda dengan Pilihan Menu Keluar

3.1.18.14 *Activity Diagram* Melihat Informasi PP



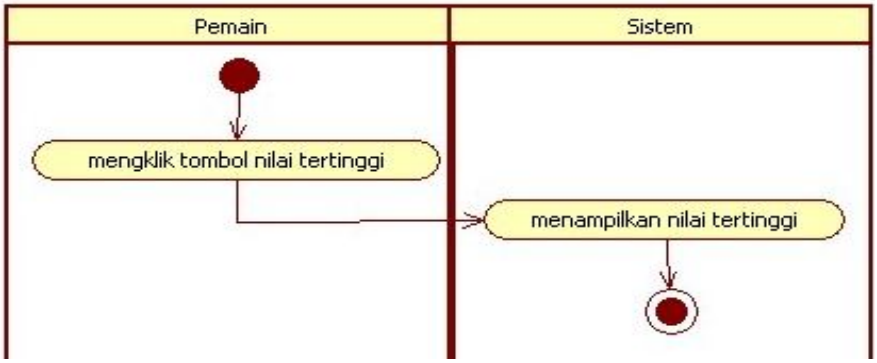
Gambar 3.33 *Activity Diagram* melihat informasi PP

3.1.18.15 *Activity Diagram* Melihat Kontrol Permainan



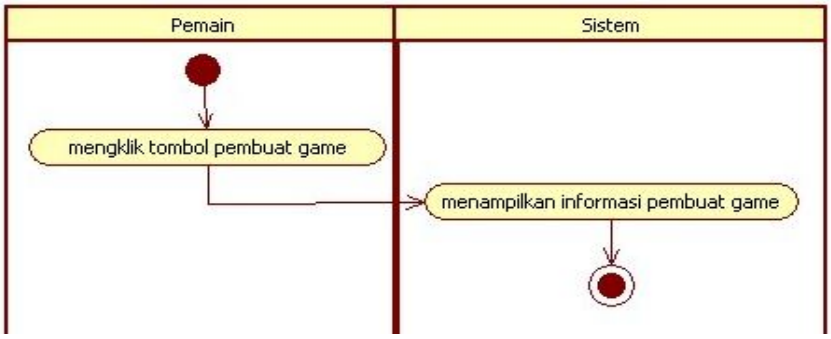
Gambar 3.34 *Activity Diagram* Melihat Kontrol Permainan

3.1.18.16 *Activity Diagram* Informasi Nilai Tertinggi



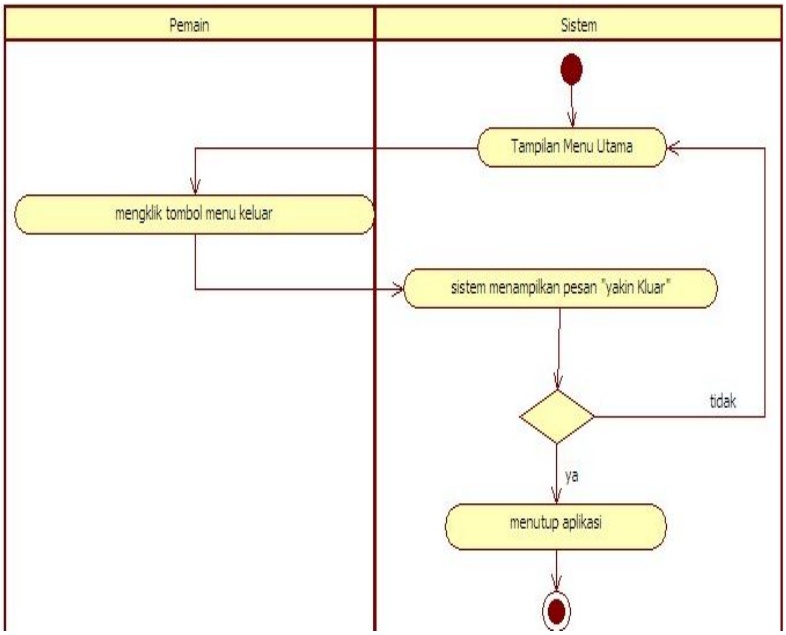
Gambar 3.35 *Activity Diagram* Melihat Nilai Tertinggi

3.1.18.17 *Activity Diagram* Informasi Pembuat *Game*



Gambar 3.36 *Activity Diagram* Melihat Informasi Pembuat *Game*

3.1.18.18 *Activity Diagram* Keluar Aplikasi

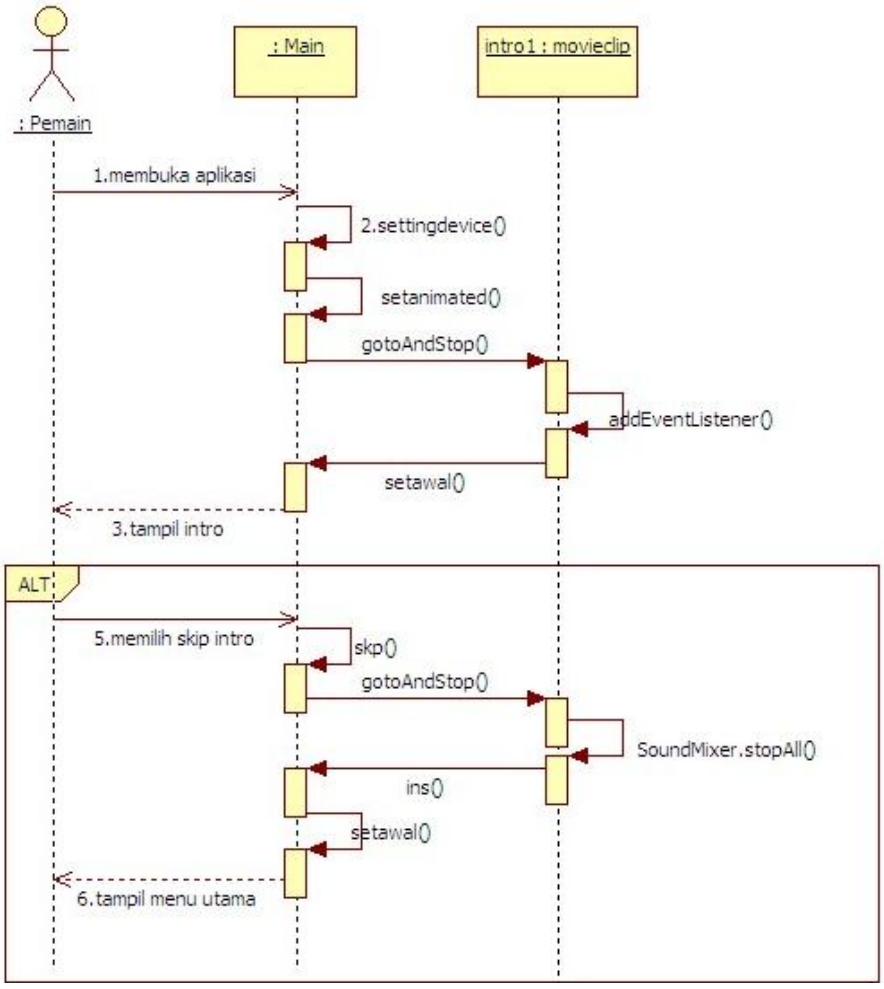


Gambar 3.37 *Activity Diagram* Keluar Apikasi.

3.1.19 Pengambaran Interaksi Antar Objek pada *Game*

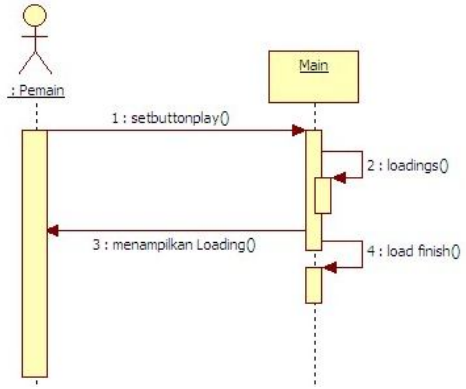
Penggambaran interaksi antar objek menggambarkan interaksi antar objek yang terjadi didalam dan diluar sistem pada *Game* (termasuk *Player*, *vertical*, dan sebagainya). Pengambaran interaksi pada *Game* menggunakan *Sequence Diagram*. *Sequence Diagram* terdiri atas dimensi *vertical* (waktu) dan dimensi *horizontal* (objek – objek ang terkait). Pada *Sequence Diagram* ini menggambarkan scenario atau rangkaian langkah – langkah yang dilakukan pada sistem sebagai respon dari sebuah *event* untuk menghasilkan output tertentu. Berikut adalah gambar dari *Sequence Diagram* yang digunakan pada *Game* Pertolongan Pertama :

3.1.19.1 *Sequence Diagram* melihat intro



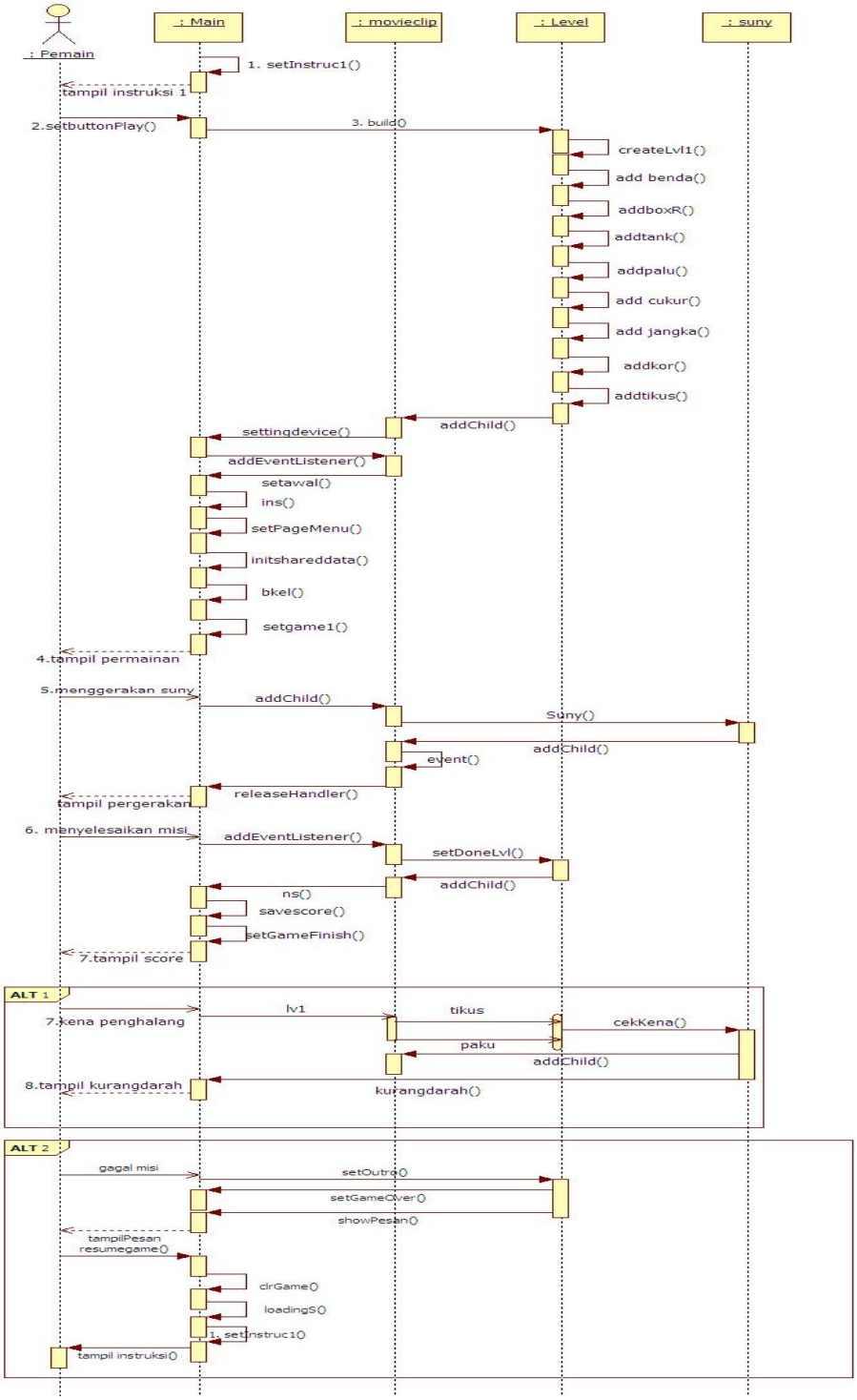
Gambar 3.38 Sequence Melihat Intro

3.1.19.2 *Sequence Diagram* mulai permainan



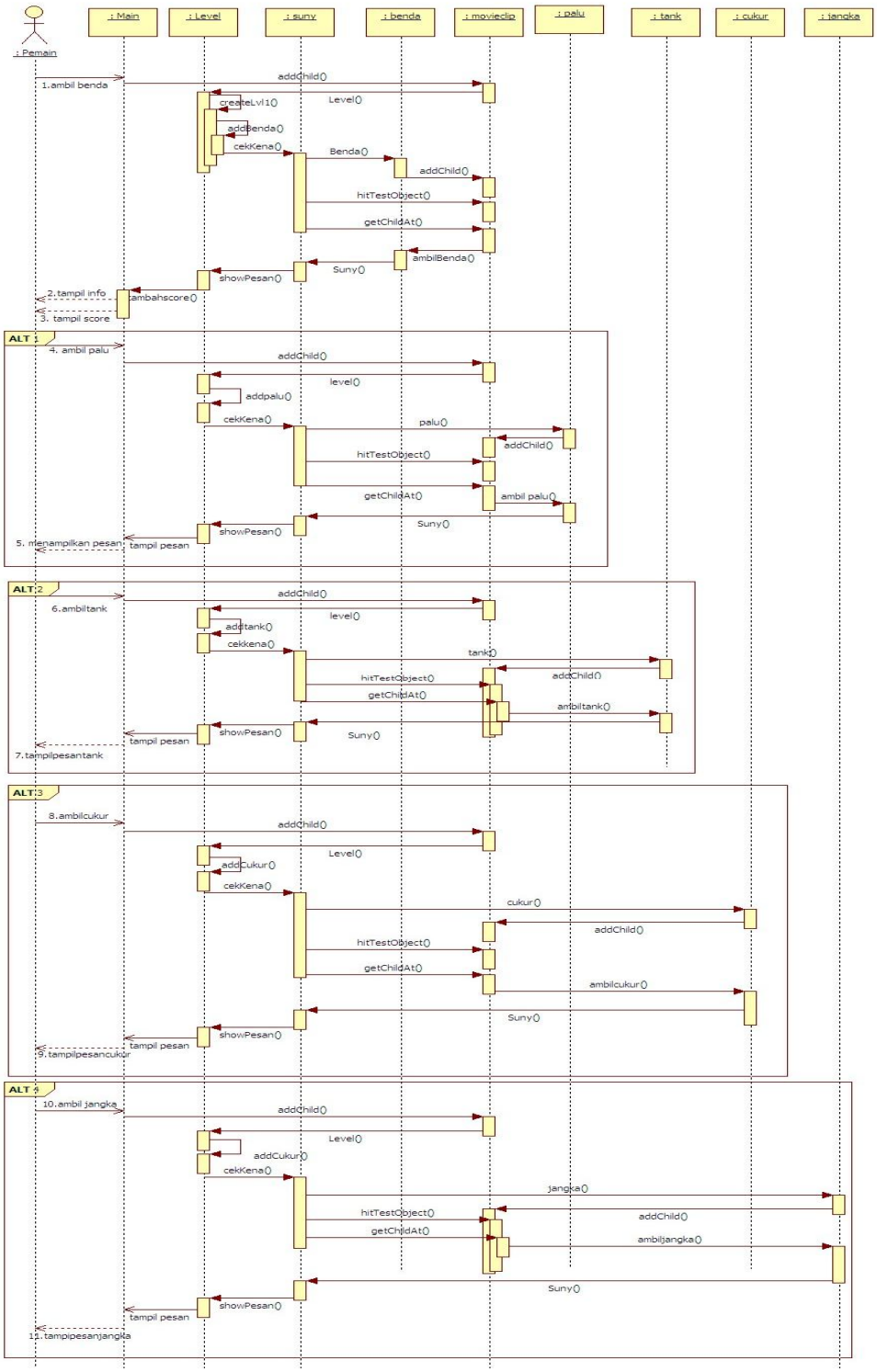
Gambar 3.39 *Sequence Diagram* Mulai Permainan

3.1.19.3 *Sequence Diagram* Level 1



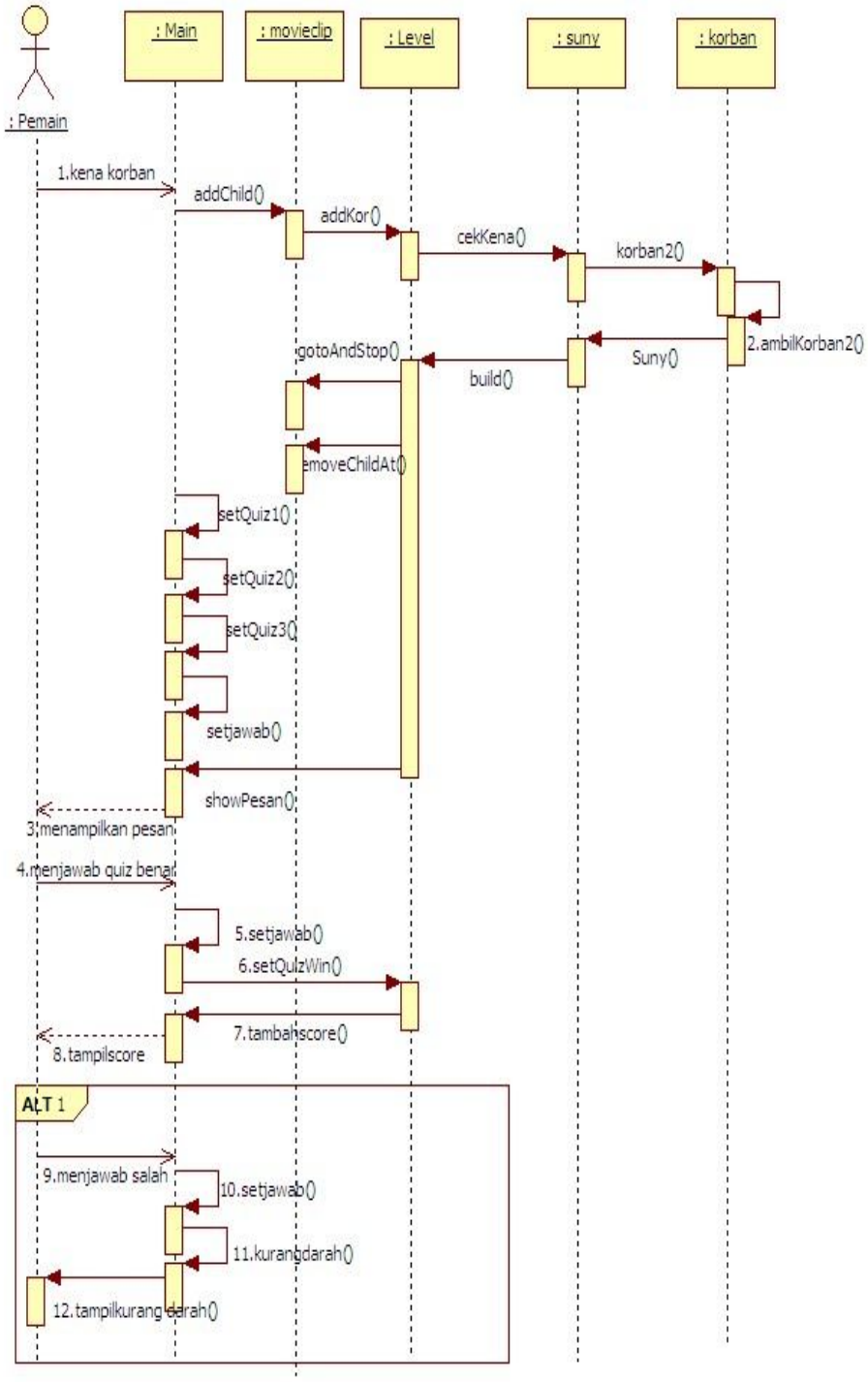
Gambar 3.40 *Sequence Diagram* Level 1

3.1.19.4 *Sequence Diagram* alat Apd



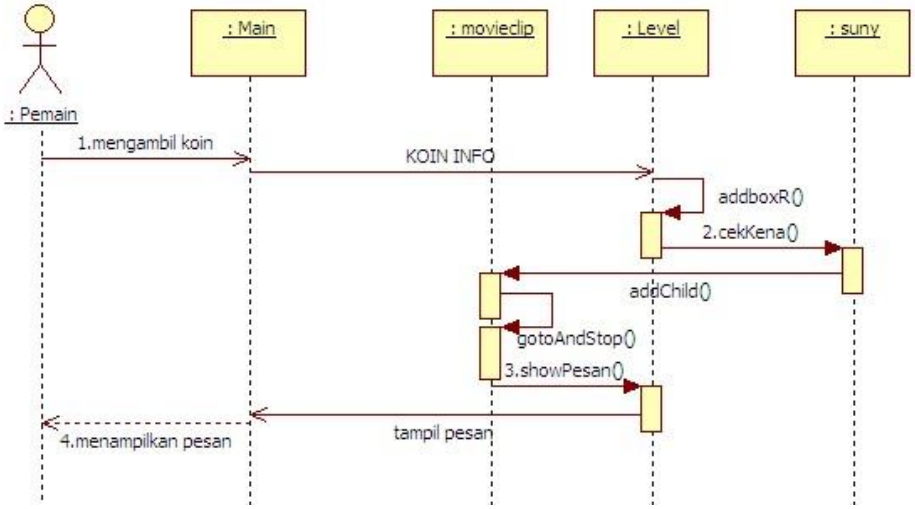
Gambar 3.41 *Sequence Diagram* alat APD

3.1.19.5 *Sequence Diagram* menolong korban



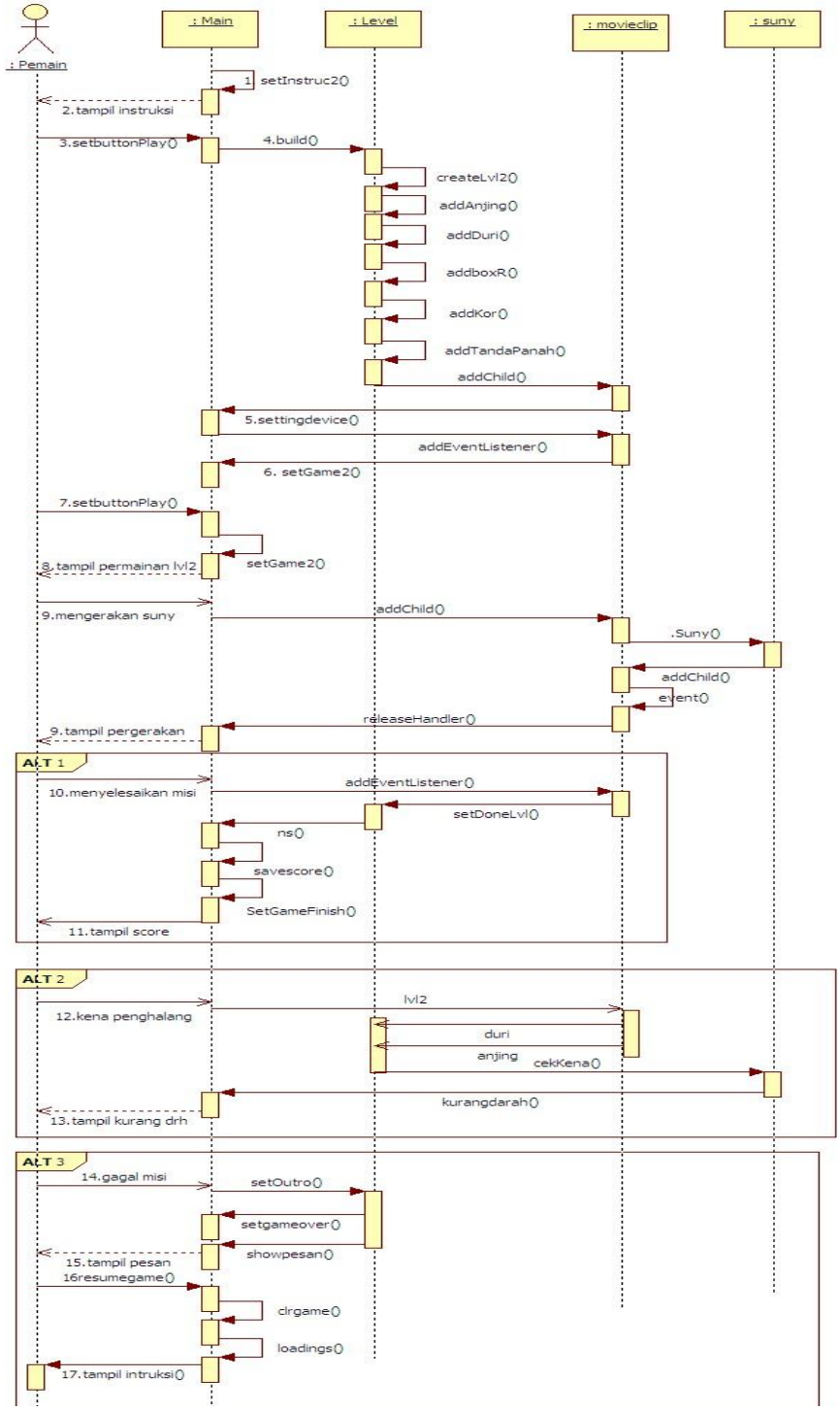
Gambar 3.42 *Sequence Diagram* Menolong Korban

3.1.19.6 *Sequence Diagram* koin informasi



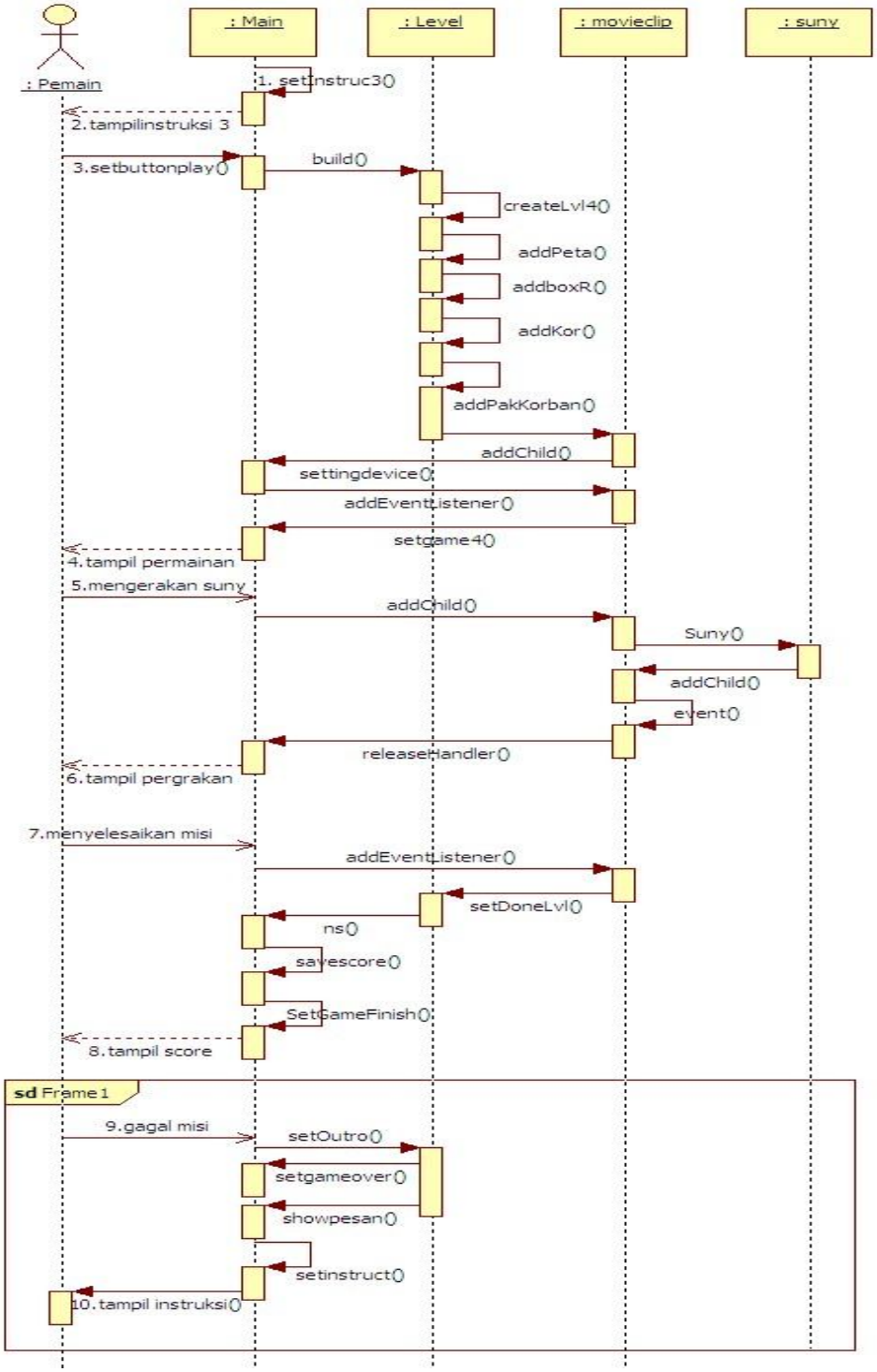
Gambar 3.43 *Sequence Diagram* koin informasi

3.1.19.7 *Sequence Diagram* level 2



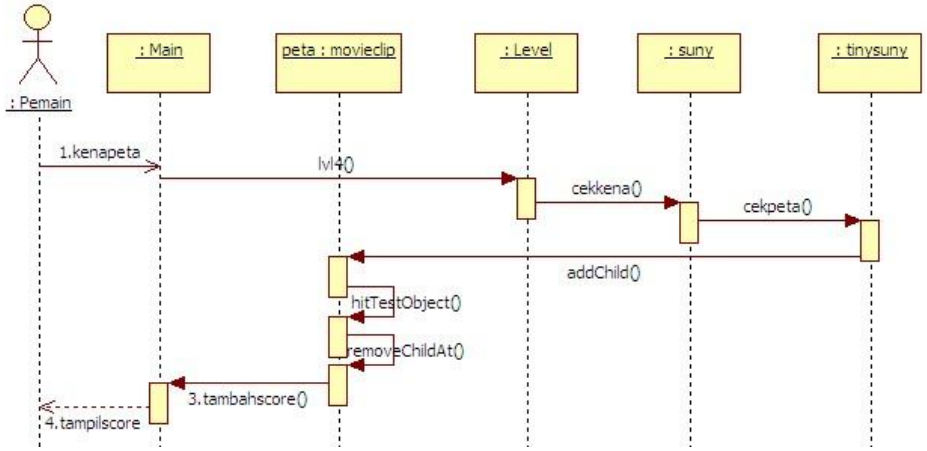
Gambar 3.44 *Sequence Diagram* Level 2

3.1.19.8 *Sequence Diagram* level 3



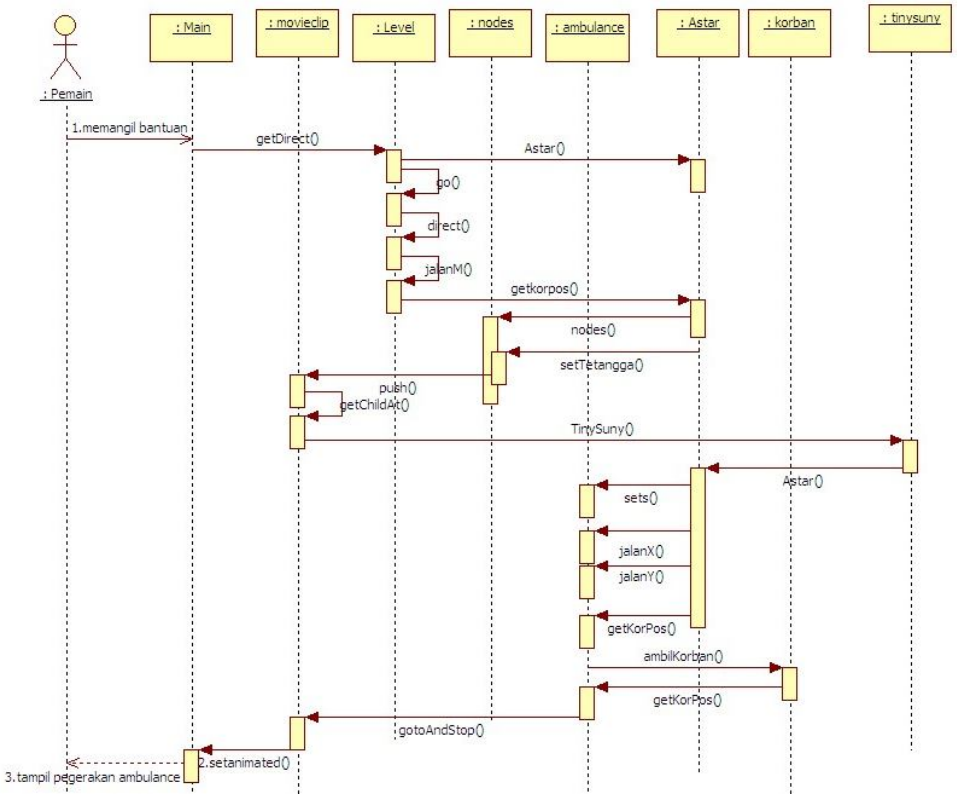
Gambar 3.45 *Sequence Diagram* level 3

3.1.19.9 *Sequence Diagram* Ambil Peta



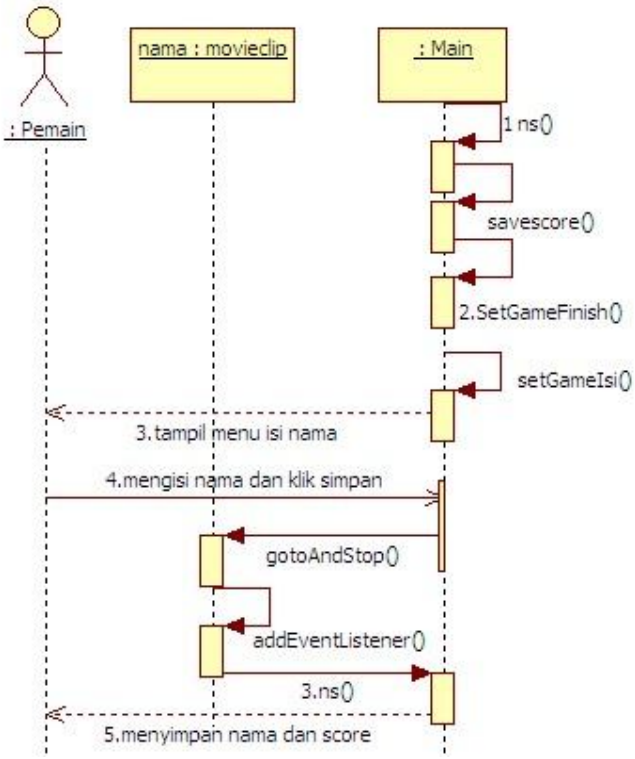
Gambar 3.46 *Sequence Diagram* Mengambil Peta

3.1.19.10 *Sequence Diagram* pegerakan ambulance



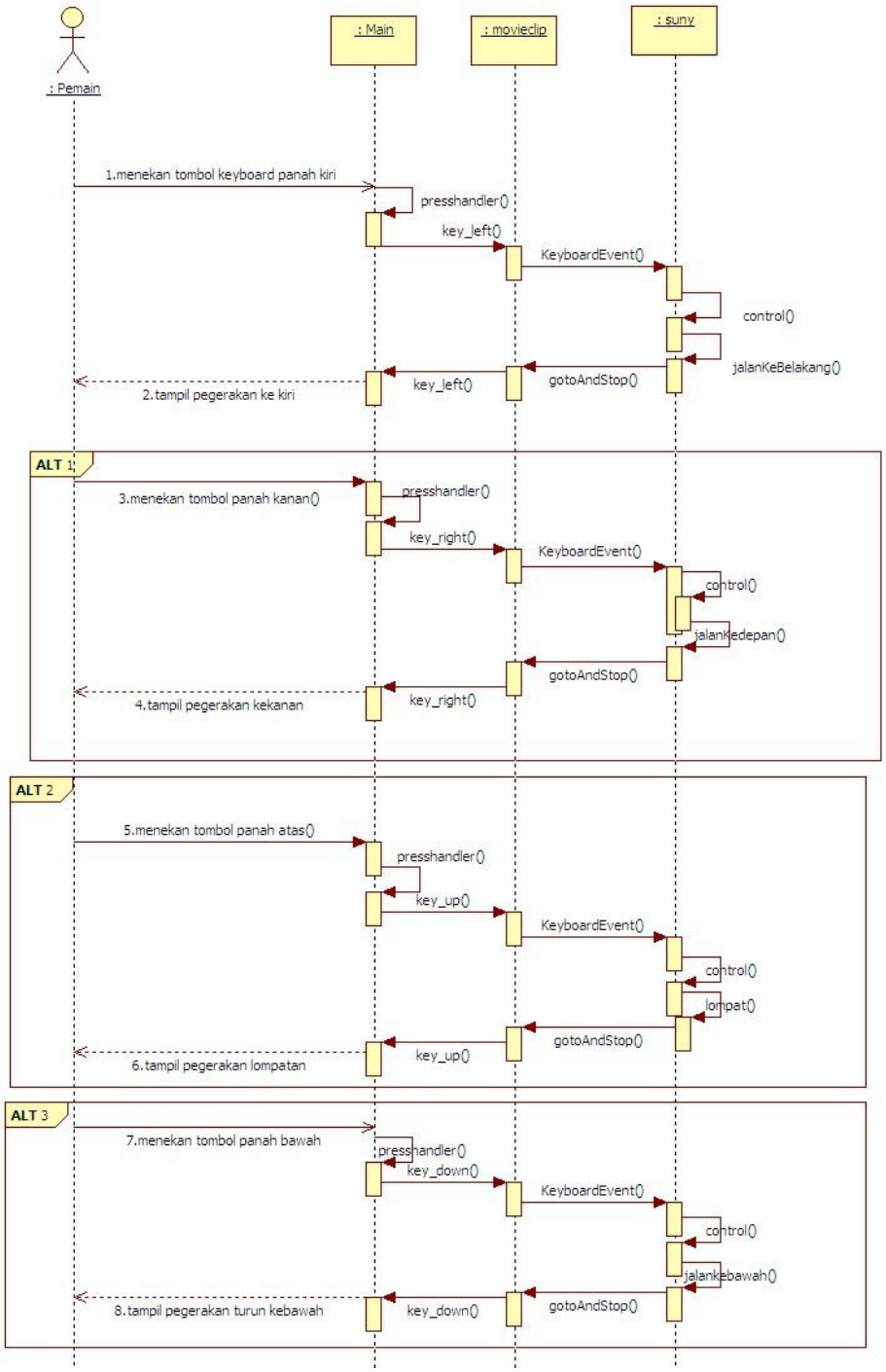
Gambar 3.47 *Sequence Diagram* Pegerakan Ambulance

3.1.19.11 *Sequence Diagram* Menginput Nama Highscore



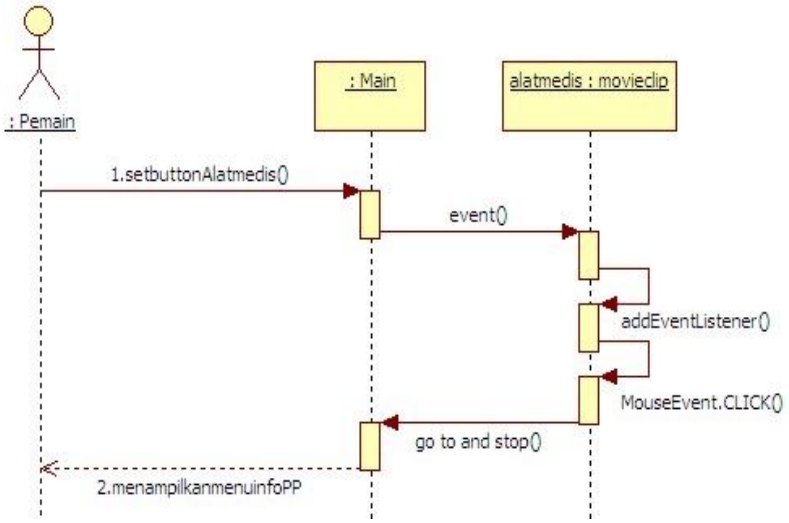
Gambar 3.48 *Sequence Diagram* Input Nama

3.1.19.12 Sequnce Diagram Mengerakan Karakter



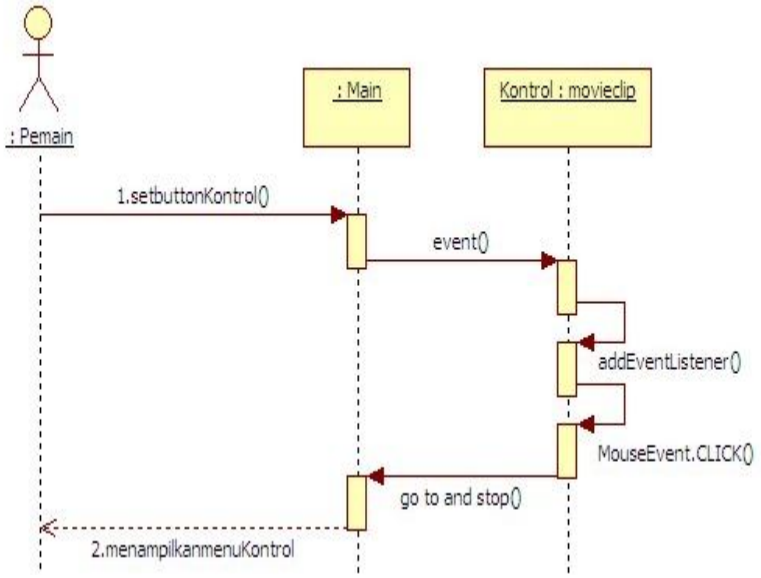
Gambar 3.49 *Sequence Diagram* pegerakan karakter

3.1.19.13 *Sequence Diagram* Melihat Informasi PP



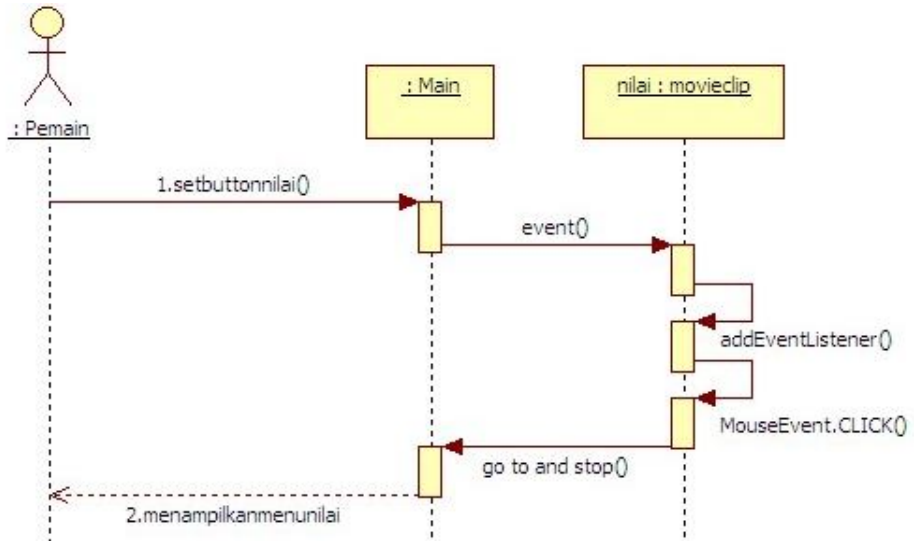
Gambar 3.50 *Sequence Diagram* Melihat Informasi PP

3.1.19.14 *Sequence Diagram* melihat Kontrol Permainan



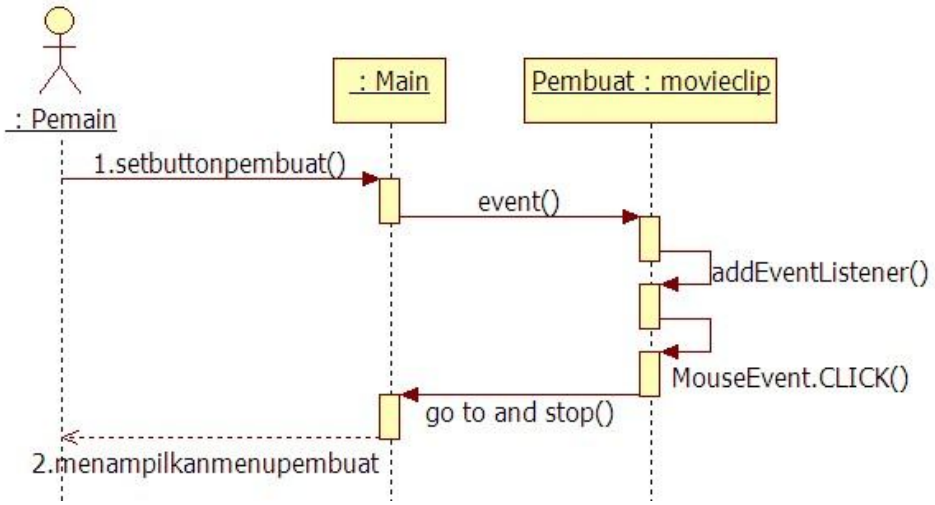
Gambar 3.51 *Sequence Diagram* Melihat Kontrol Permainan

3.1.19.15 *Sequence Diagram* Nilai Tertinggi



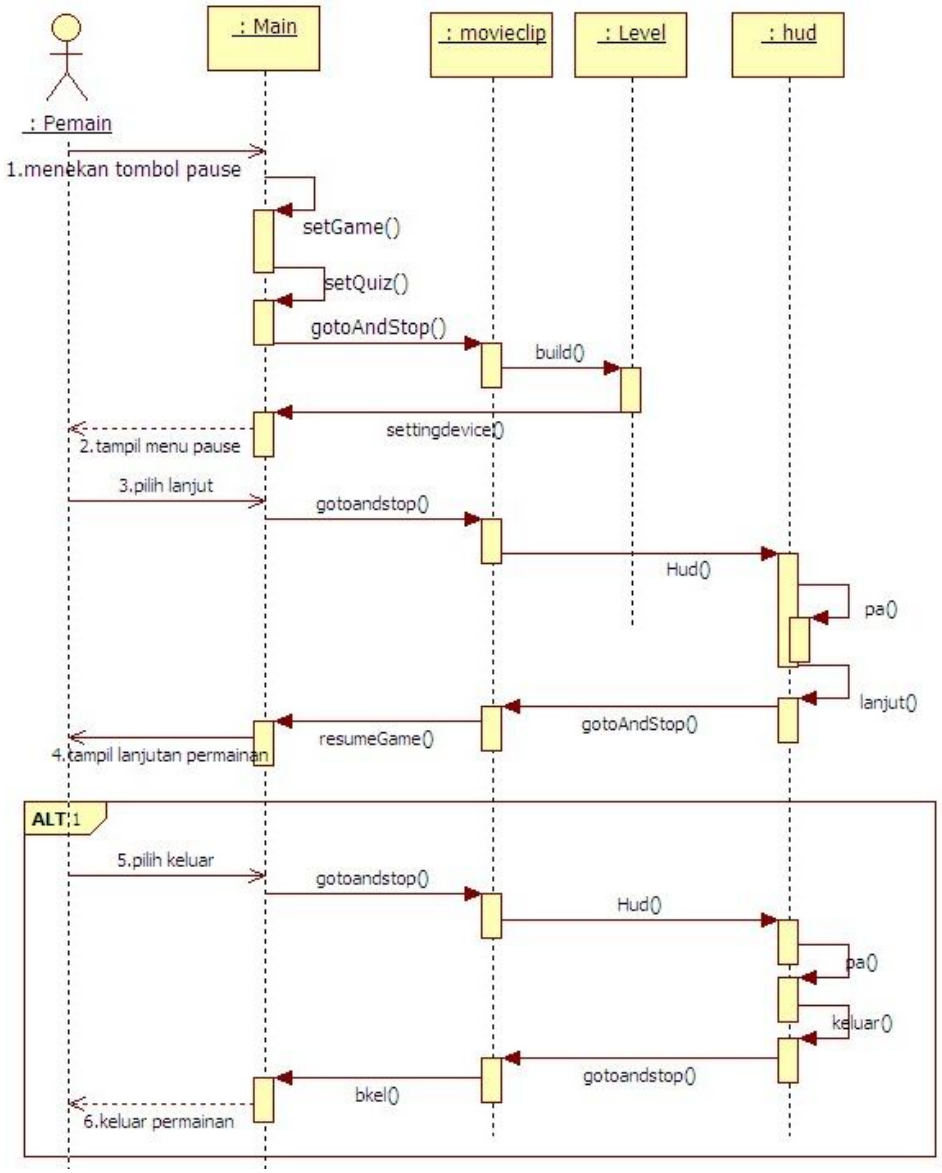
Gambar 3.52 *Sequence Diagram* Nilai Tertinggi

3.1.19.16 *Sequence Diagram* Pembuat *Game*



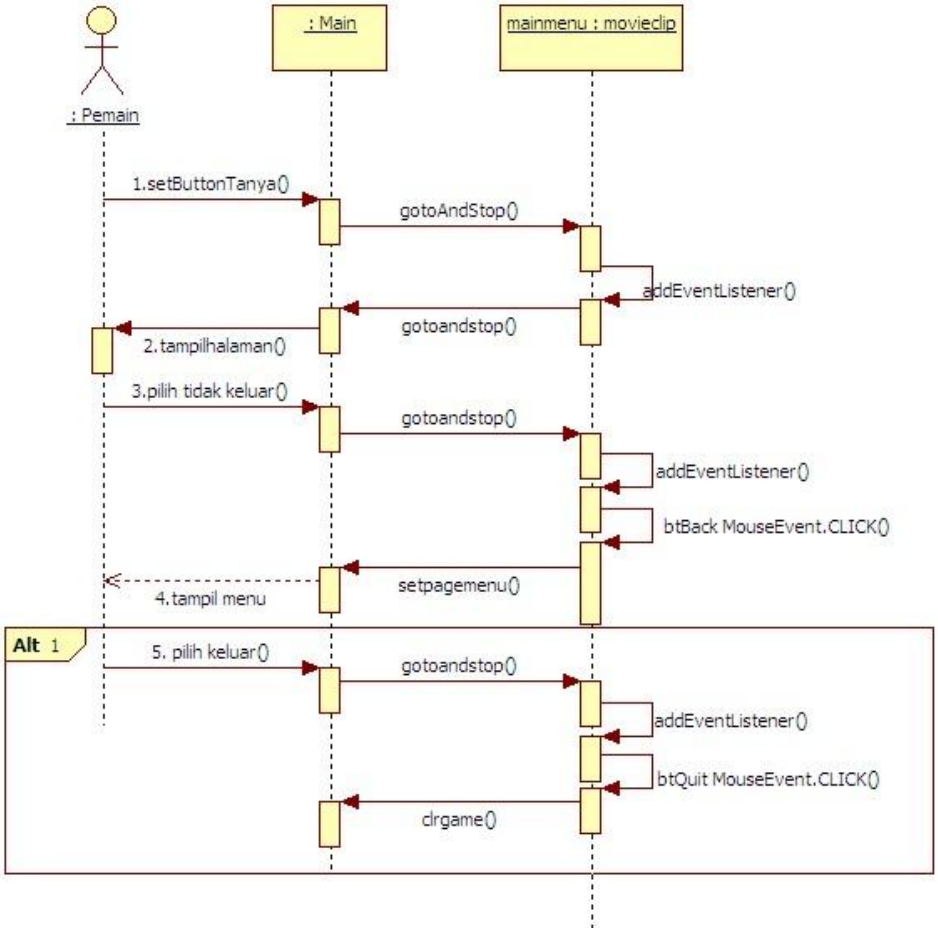
Gambar 3.53 *Sequence* Pembuat *Game*

3.1.19.17 *Sequence Diagram* Menghentikan Permainan / jeda



Gambar 3.54 *Sequence Diagram* Menghentikan Permainan / jeda

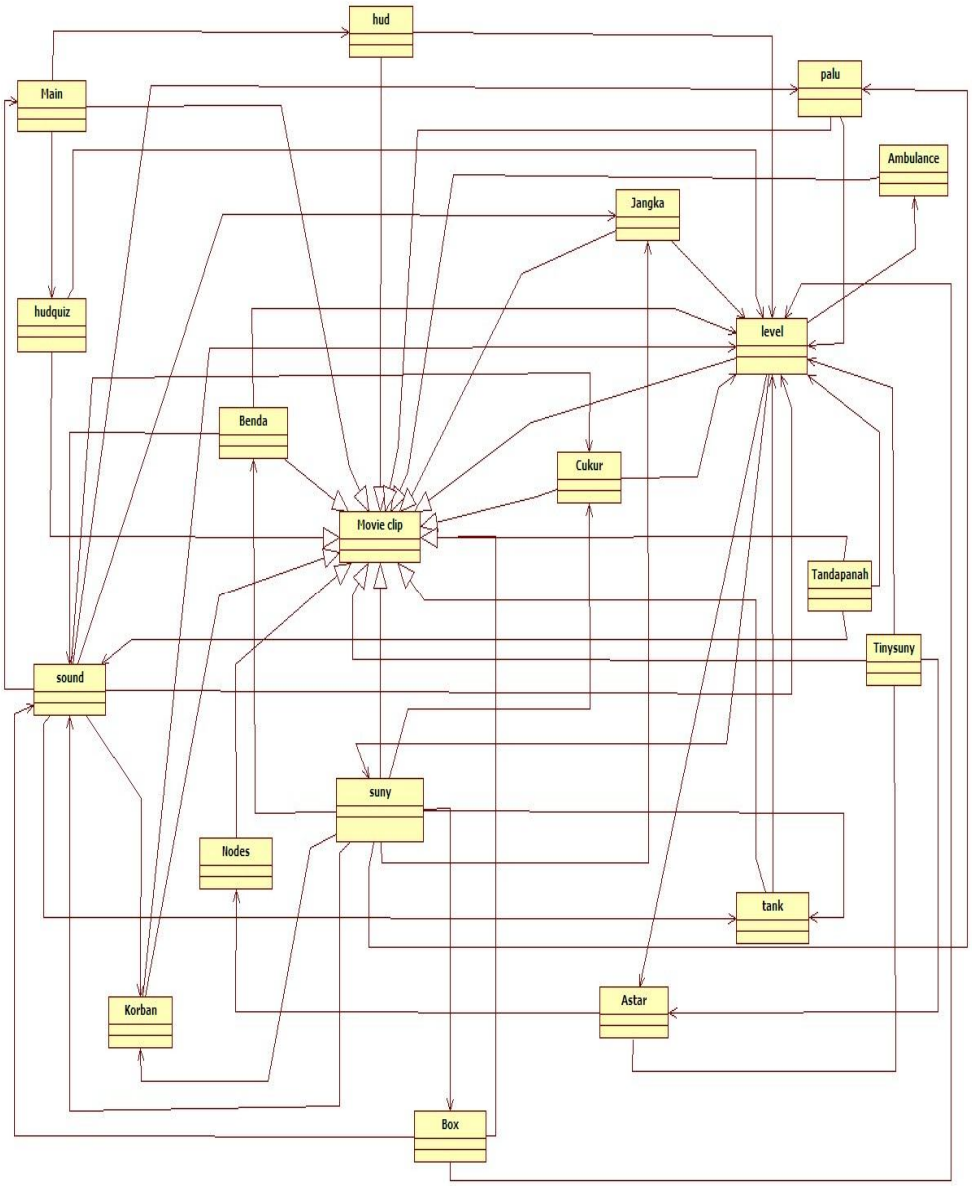
3.1.19.18 *Sequence Diagram* Keluar Aplikasi



Gambar 3.55 *Sequence Diagram* Keluar Aplikasi

3.1.20 *Class Diagram*

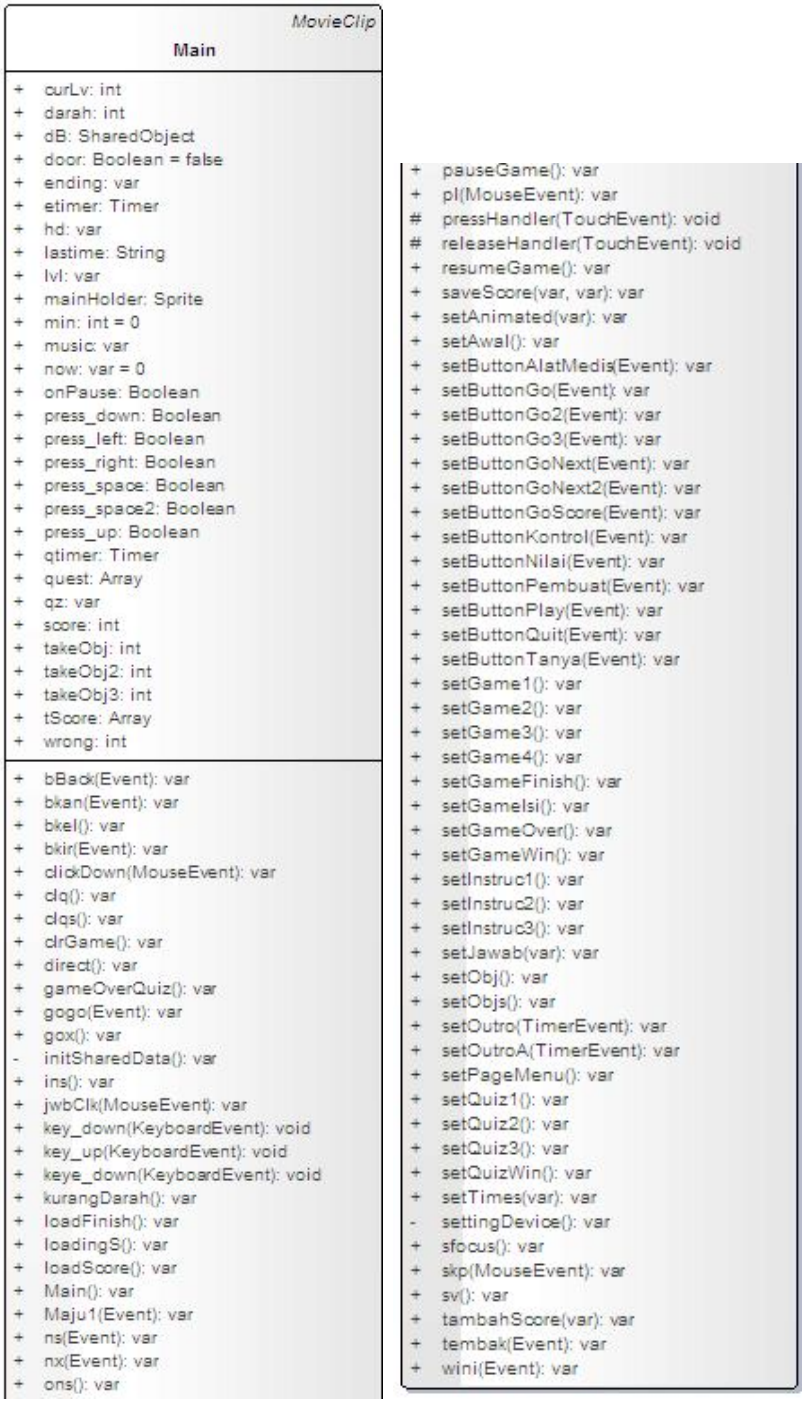
Class diagram berfungsi untuk menunjukan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.



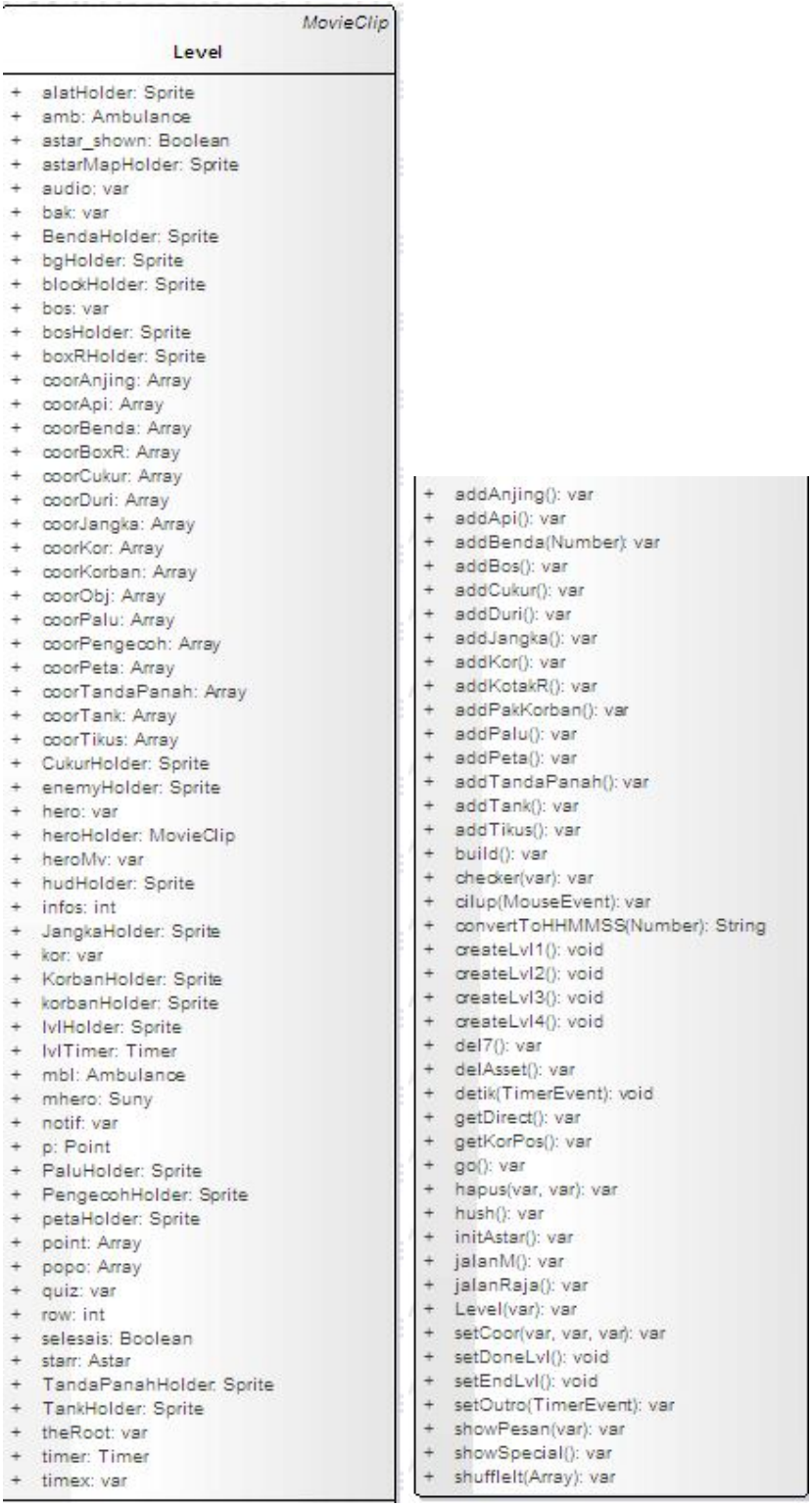
Gambar 3.56 *Class Diagram*

Berikut dibawah ini adalah rincian nama – nama kelas yang ada di dalam *Game* Pertolongan Pertama :

1. Main



2. Level


MovieClip
Level
+ alatHolder: Sprite
+ amb: Ambulance
+ astar_shown: Boolean
+ astarMapHolder: Sprite
+ audio: var
+ bak: var
+ BendaHolder: Sprite
+ bgHolder: Sprite
+ blockHolder: Sprite
+ bos: var
+ bosHolder: Sprite
+ boxRHolder: Sprite
+ coorAnjing: Array
+ coorApi: Array
+ coorBenda: Array
+ coorBoxR: Array
+ coorCukur: Array
+ coorDuri: Array
+ coorJangka: Array
+ coorKor: Array
+ coorKorban: Array
+ coorObj: Array
+ coorPalu: Array
+ coorPengecoh: Array
+ coorPeta: Array
+ coorTandaPanah: Array
+ coorTank: Array
+ coorTikus: Array
+ CukurHolder: Sprite
+ enemyHolder: Sprite
+ hero: var
+ heroHolder: MovieClip
+ heroMv: var
+ hudHolder: Sprite
+ infos: int
+ JangkaHolder: Sprite
+ kor: var
+ KorbanHolder: Sprite
+ korbanHolder: Sprite
+ lvlHolder: Sprite
+ lvlTimer: Timer
+ mbl: Ambulance
+ mhero: Suny
+ notif: var
+ p: Point
+ PaluHolder: Sprite
+ PengecohHolder: Sprite
+ petaHolder: Sprite
+ point: Array
+ popo: Array
+ quiz: var
+ row: int
+ selesais: Boolean
+ starr: Astar
+ TandaPanahHolder: Sprite
+ TankHolder: Sprite
+ theRoot: var
+ timer: Timer
+ timex: var
+ addAnjing(): var
+ addApi(): var
+ addBenda(Number): var
+ addBos(): var
+ addCukur(): var
+ addDuri(): var
+ addJangka(): var
+ addKor(): var
+ addKotakR(): var
+ addPakKorban(): var
+ addPalu(): var
+ addPeta(): var
+ addTandaPanah(): var
+ addTank(): var
+ addTikus(): var
+ build(): var
+ checker(var): var
+ cilup(MouseEvent): var
+ convertToHHMMSS(Number): String
+ createLvl1(): void
+ createLvl2(): void
+ createLvl3(): void
+ createLvl4(): void
+ del7(): var
+ delAsset(): var
+ detik(TimerEvent): void
+ getDirect(): var
+ getKorPos(): var
+ go(): var
+ hapus(var, var): var
+ hush(): var
+ initAstar(): var
+ jalanM(): var
+ jalanRaja(): var
+ Level(var): var
+ setCoor(var, var, var): var
+ setDoneLvl(): void
+ setEndLvl(): void
+ setOutro(TimerEvent): var
+ showPesan(var): var
+ showSpecial(): var
+ shuffleIt(Array): var

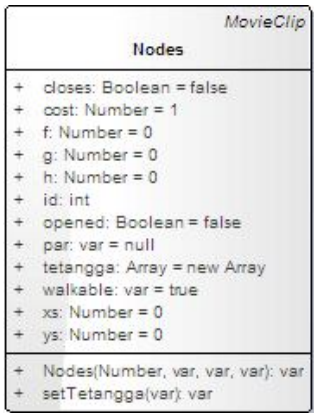
3. Tiny Suny



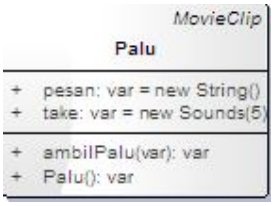
4. Astar



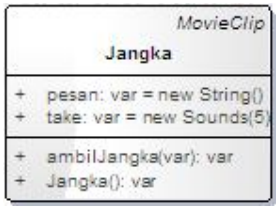
5. *Node*s



6. Palu



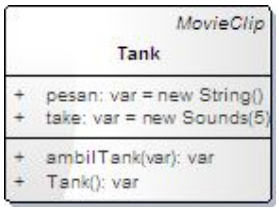
7. Jangka



8. Ambulance



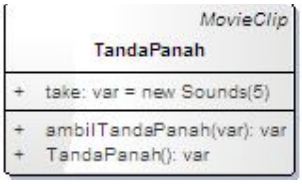
9. Tank



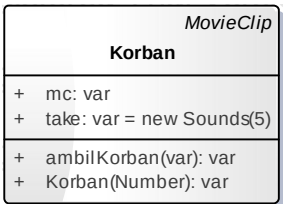
10. Sounds



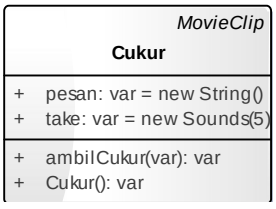
11. Tandapanah



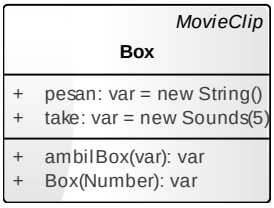
12. Korban



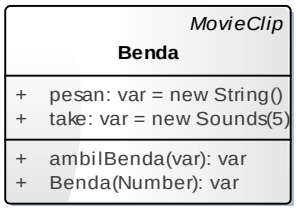
13. Cukur



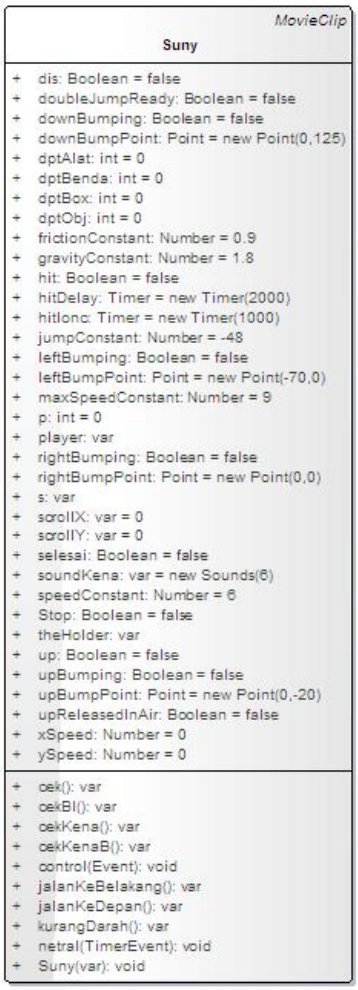
14. Box



15. Benda



16. Suny



17. Hudquiz



18. Hud



19. Movieclip

Movieclip
+ add*Event*Listener() + gotoAndStop() +*event*() + stage.remove*Event*Listener() + gotoAndPlay() + addChild() + remove*Event*Listener() + removeChildAt() + localToGlobal() + Point() + hitTestPoint() + push() + contains() +Math.random() +splice() + hitTestObject() + btBack Mouse*Event*.CLICK +btQuit Mouse*Event*.CLICK + *event*(): Keyboard*Event* + stage.focus() + releaseHandler()

3.2 Perancangan

Setelah melakukan analisis pada sistem maka langkah selanjutnya yaitu melakukan perancangan arsitektur. Perancangan arsitektur yang telah dibuat meliputi beberpa perancangan diantaranya perancangan komponen permainan, struktur menu, perancangan antarmuka.

3.2.1 Perancangan Komponen Permainan

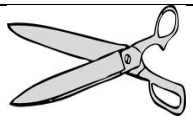
Perancangan komponen meliputi deskripsi karakter dan storyboard. Berikut adalah deskripsi karakter, dan *storyboard* yang diguakan dalam *Game* pertolongan pertama.

3.2.1.1 Karakter *Game*

Dalam aplikasi *Game* pertolongan pertama ini terdapat beberapa tokoh karakter dengan rincian deskripsi dapat dilihat pada tabel dibawah ini :

Tabel 3.44 Deskripsi Karakter Tokoh dan Item *Game*

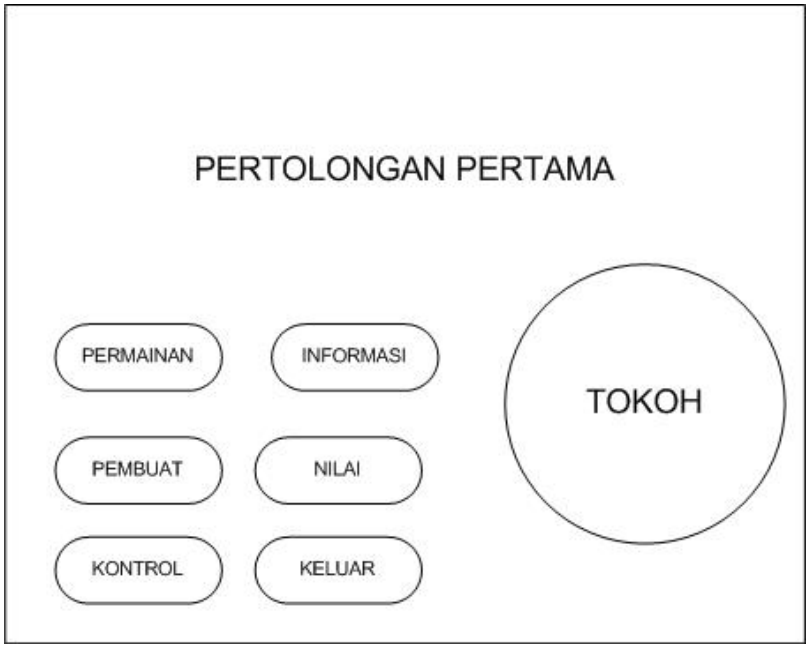
No	Gambar Karakter	Nama Karakter	Deskripsi
1		Suny	Suny merupakan tokoh utama pada *Game* pertolongan pertama ini.
2		herman	Herman merupakan tokoh petugas rekreasi anak yang akan meminta bantuan kepada Suny untuk menemukan korban.
		sara	Sara merupakan tokoh yang akan menjadi korban pada salah satu level yang ada di *Game* ini dan harus ditolong oleh pemain

		Raja	Raja merupakan salah satu tokoh korban yang ada pada level ini dan harus ditolong oleh pemain
		Theo	Theo merupakan salah satu korban yang ada pada *Game* ini dan harus ditolong oleh pemain.
		Malika	Malika merupakan salah satu korban pingsan yang ada pada *Game* ini dan harus di tolong oleh pemain
		Dokter Christ	Dokter Christ adalah tenaga medis yang akan muncul di akhir permainan setelah mendapat panggilan dari tokoh utama
		peta	Peta merupakan bagian benda yang harus dikumpulkan pemain
		Sarung tangan	Sarung tangan merupakan alat yang dibutuhkan oleh pemain
		Pebalut gulung	Pembalut gulung merupakan alat yang dibutuhkan oleh pemain untuk menolong korban
		Gunting	Gunting merupakan alat yang dibutuhkan oleh pemain

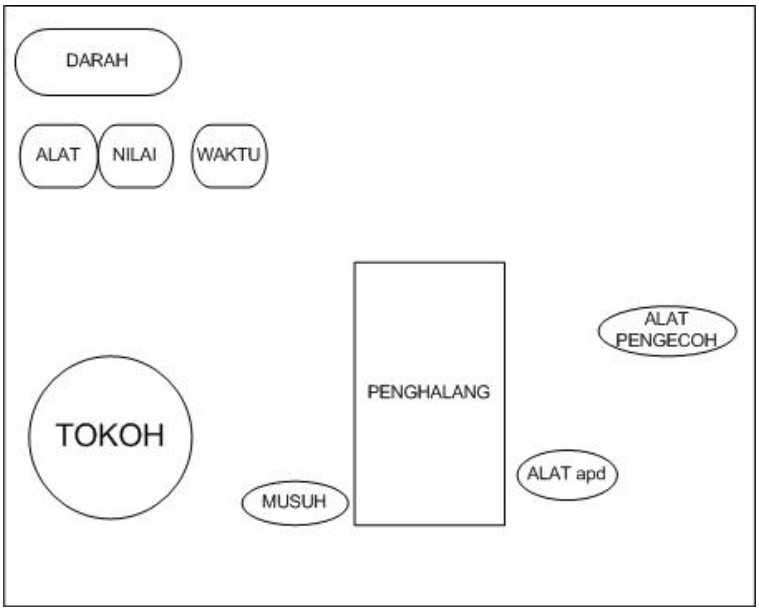
		Pinset	Pinset merupakan alat yang dibutuhkan oleh pemain
		Tas PP	Tas PP adalah perlengkapan yang slalu dibawa oleh tokoh utama yang berisi alat – alat yang dibutuhkan oleh tokoh utama untuk menolong korbannya
		Kacamata pelindung	Kacamata merupakan alat yang dibutuhkan oleh pemain untuk melindungi dirinya
		Kapas	Kapas dibutuhkan oleh pemain utnuk menyerap luka korban
		Plester luka	Plester merupakan benda yang digunakan untuk menutup luka korban
		masker	Merupakan alat perlindungan diri tokoh utama
		Obat Luka	Obat luka digunakan untuk menyembuhkan luka yang terbuka.
		Alkohol	Alcohol adalah caira antiseptic pembersih luka dari kuman – kuman yang menempel pada luka.

3.2.1.2 StoryBoard

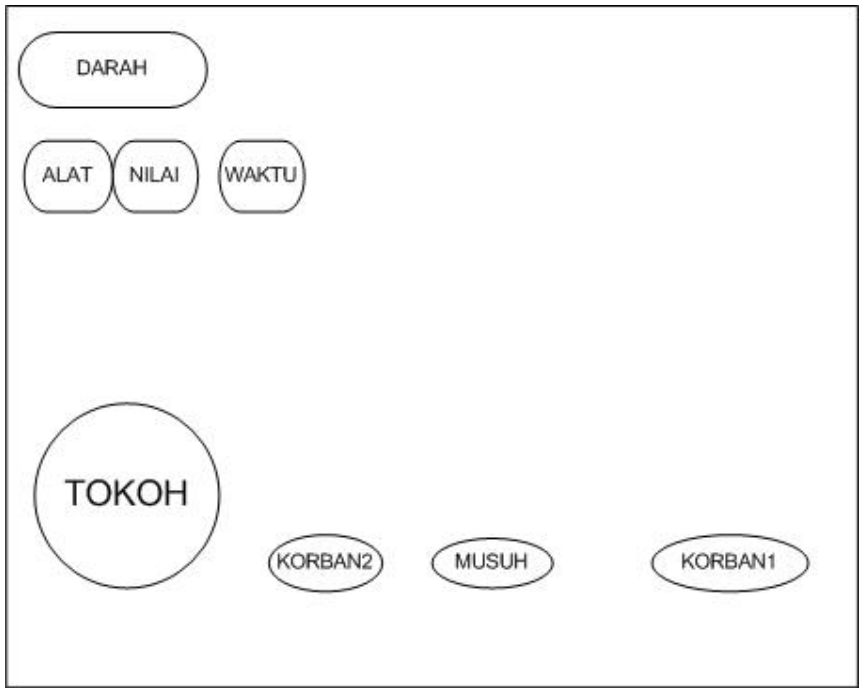
Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan naskah, dengan menggunakan storyboard dapat menyampaikan ide cerita kepada orang lain dengan lebih mudah. Berikut adalah storyboard yang digunakan dalam *Game* pertolongan pertama.



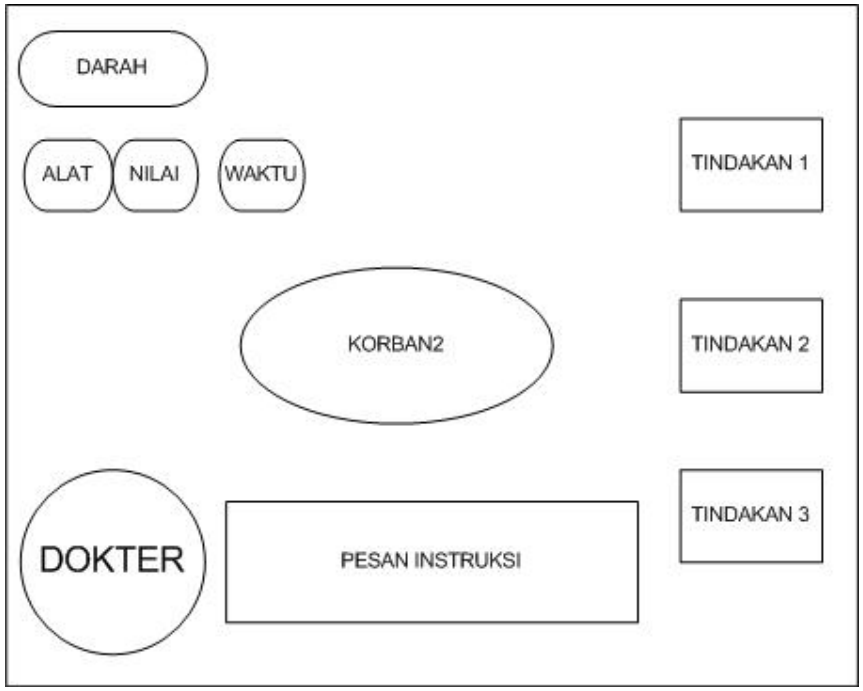
Gambar 3.57 Menu Utama Permainan



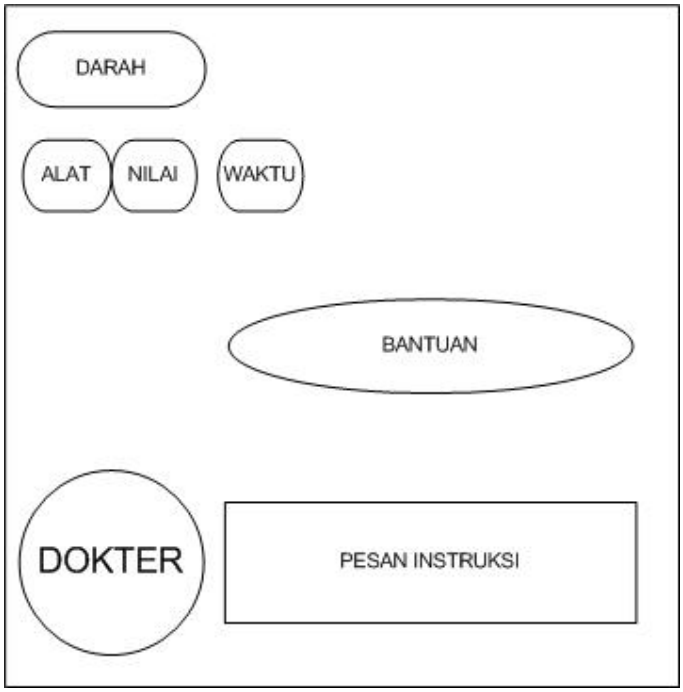
Gambar 3.58 Mengumpulkan Alat



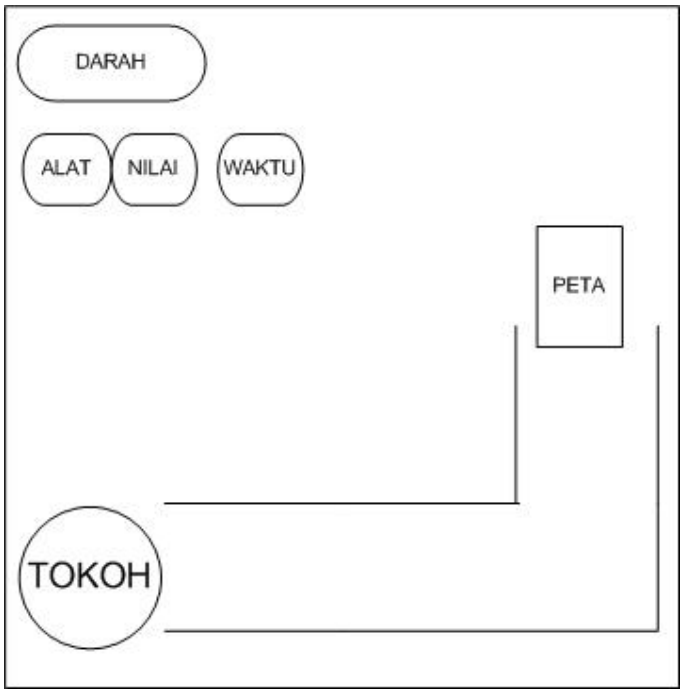
Gambar 3.59 Menemukan Korban



Gambar 3.60 Memberikan Pertolongan



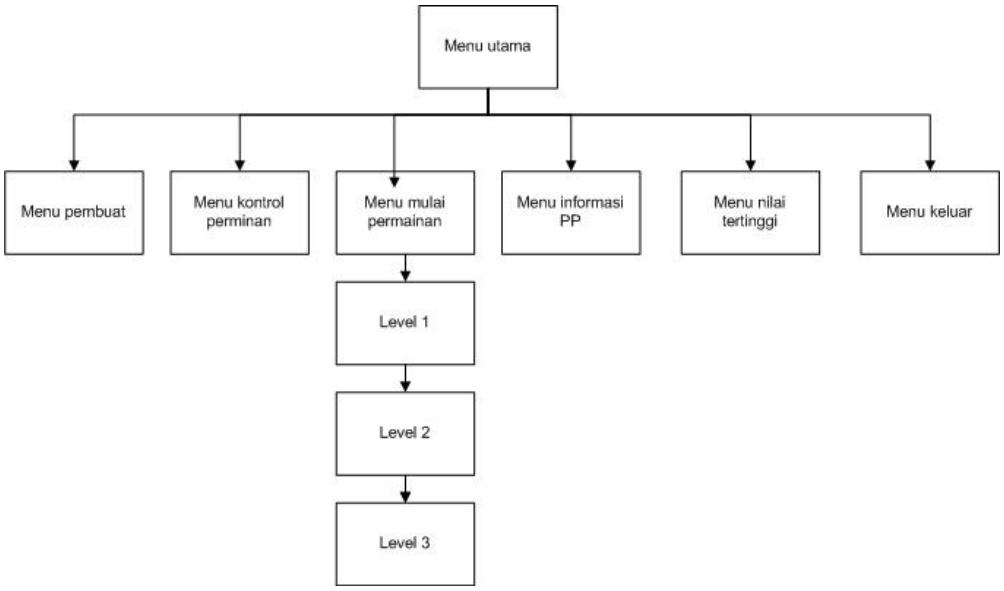
Gambar 3.61 Memanggil Bantuan



Gambar 3.62 Mengumpulkan Peta

3.2.2 Perancangan Struktur Menu

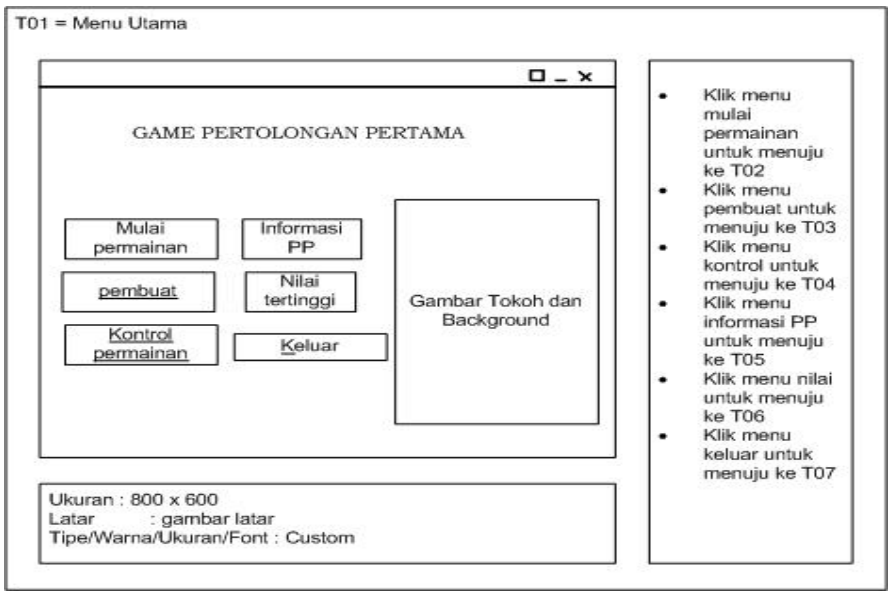
Berikut ini dapat dilihat perancagan dari struktur menu *Game* pertolongan pertama



Gambar 3.63 perancangan Struktur Menu

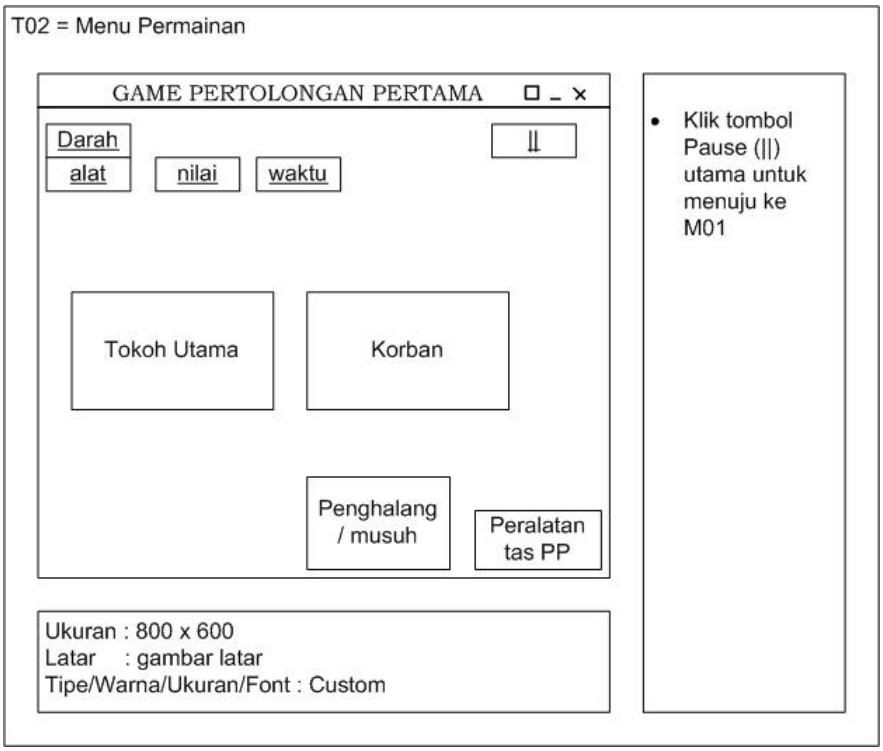
3.2.3 Perancangan Antarmuka

1. Menu Utama



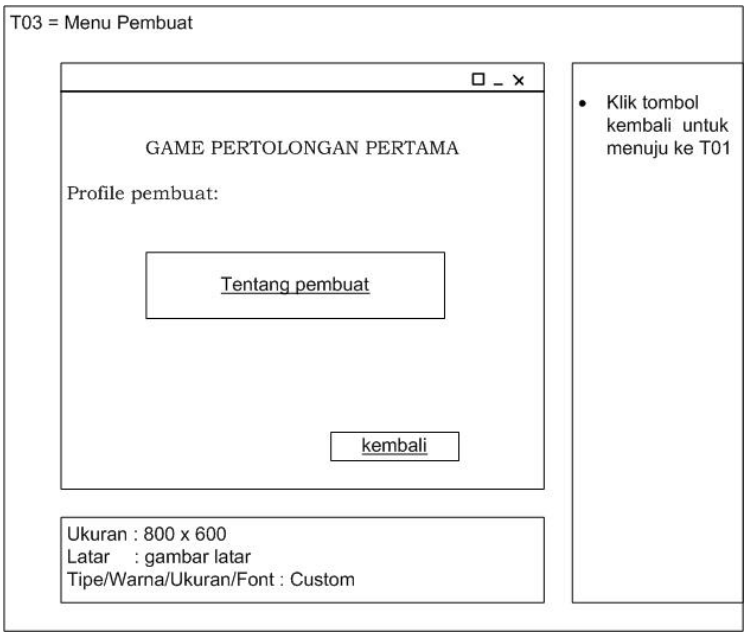
Gambar 3.64 Tampilan menu utama

2. Mulai Permainan



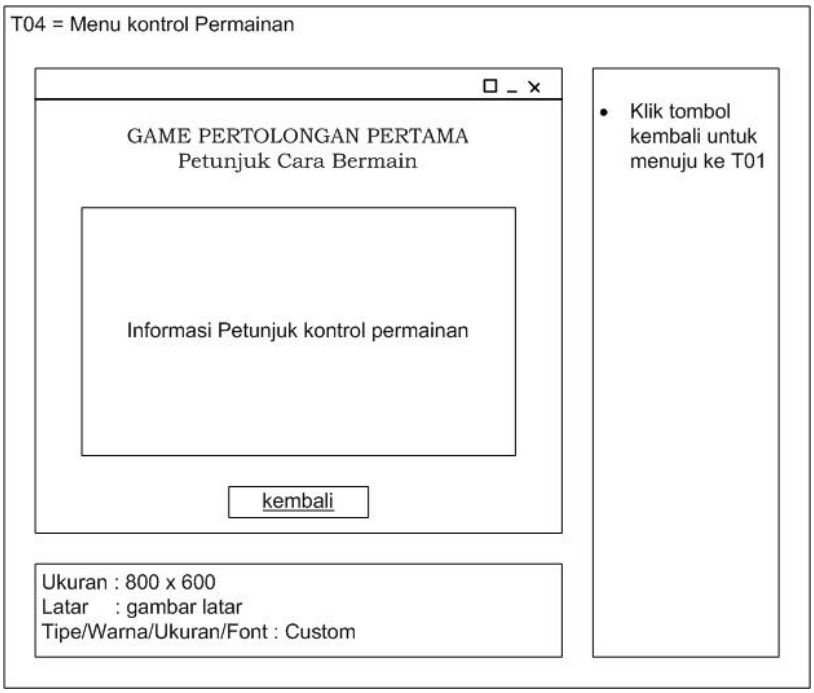
Gambar 3.65 Menu Mulai Permainan

3. Menu Pembuat



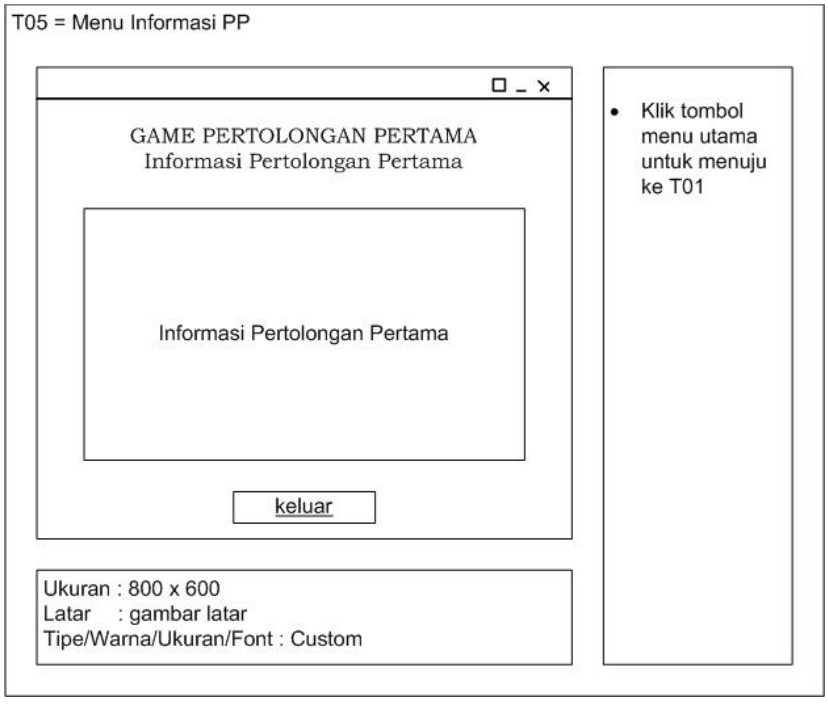
Gambar 3.66 Menu Pembuat

4. Menu Kontrol Permainan



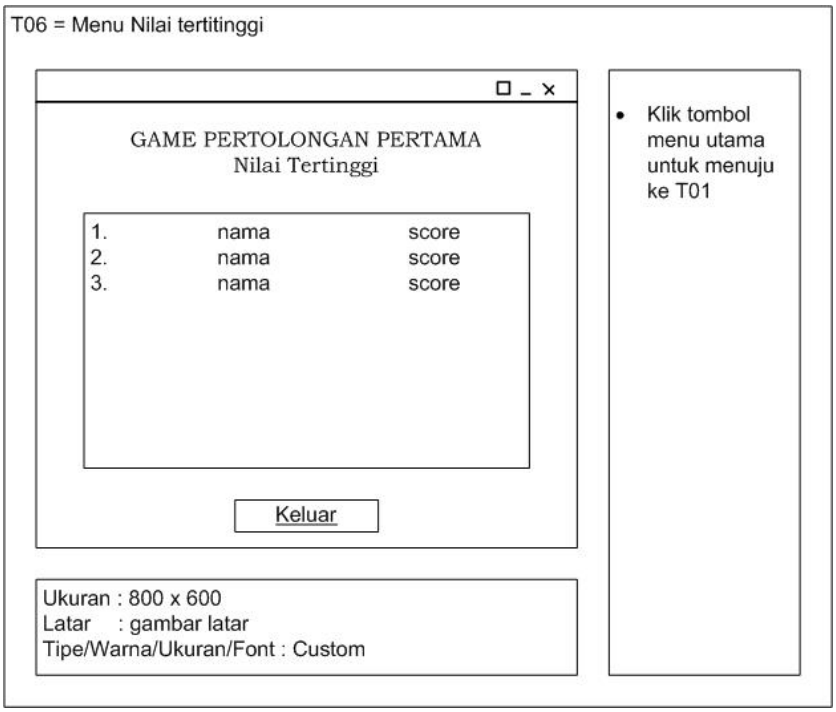
Gambar 3.67 Menu Kontrol Permainan

5. Menu Informasi PP



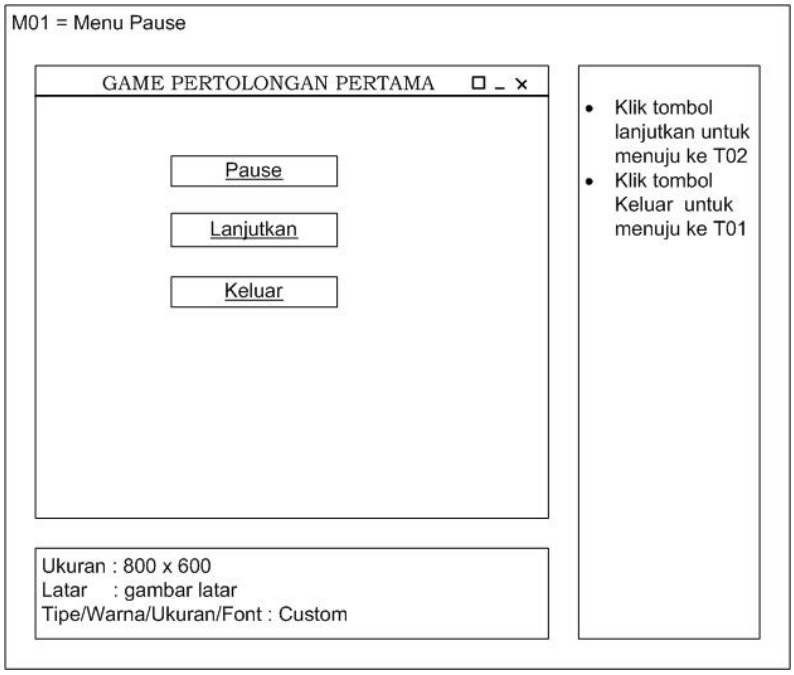
Gambar 3.68 Menu Antarmuka Informasi PP

6. Menu Nilai Tertinggi



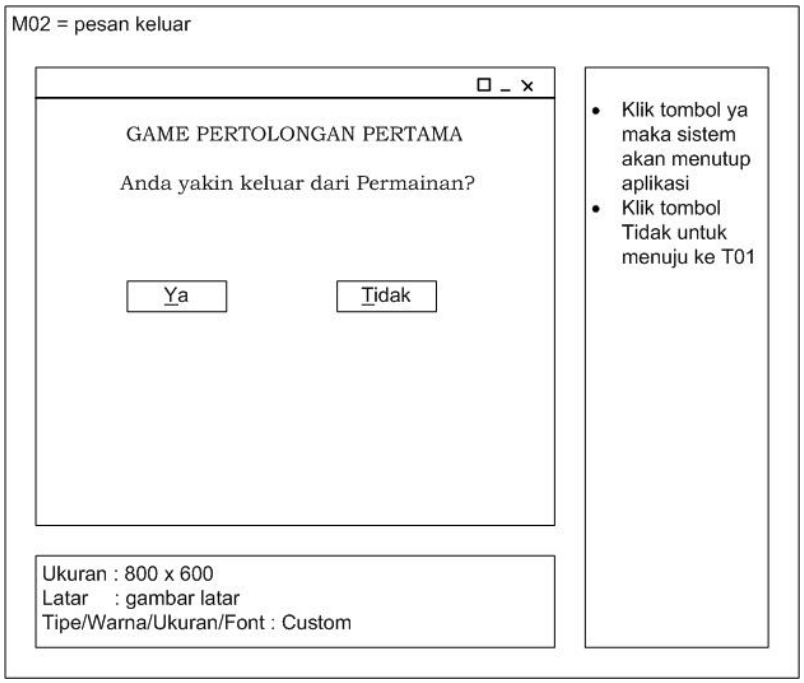
Gambar 3.69Antarmuka Menu Nilai tertinggi

7. Menu Pause



Gambar 3.70 Antarmuka Menu Pause

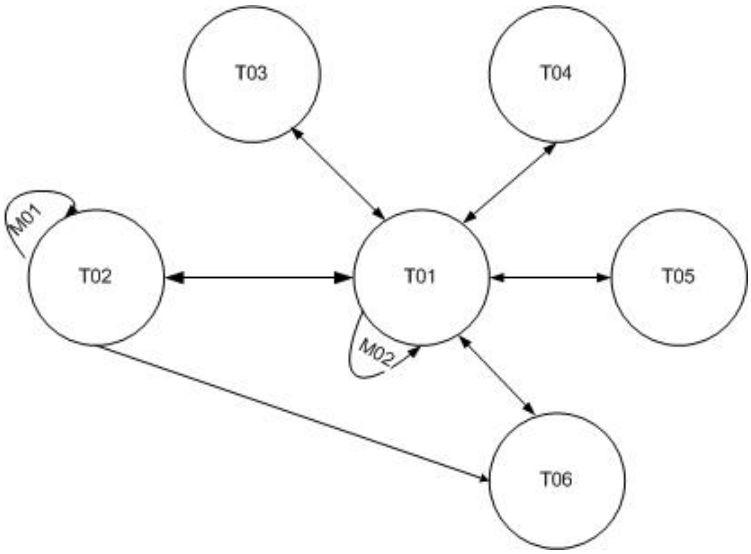
8. Menu Pesan Keluar



Gambar 3.71 Antarmuka Menu Pesan Keluar

3.2.4 Jaringan Semantik

Tahap selanjutnya adalah menggambarkan hubungan anar menu pada aplikasi *Game* Pertolongan Pertama. Penggambaran tersebut diwujudkan melalui jaringan semantic seperti gambar berikut :



Gambar 3.72 Jaringan Semantik Antar Menu