BAB III
ANALISIS DAN PERANCANGAN

3.1 Analisis

Pada analisis akan mencakup analisis masalah, analisis kebutuhan non-fungsional dan analisis kebutuhan fungsional pada game yang akan dibangun.

3.1.1 Analisis Masalah

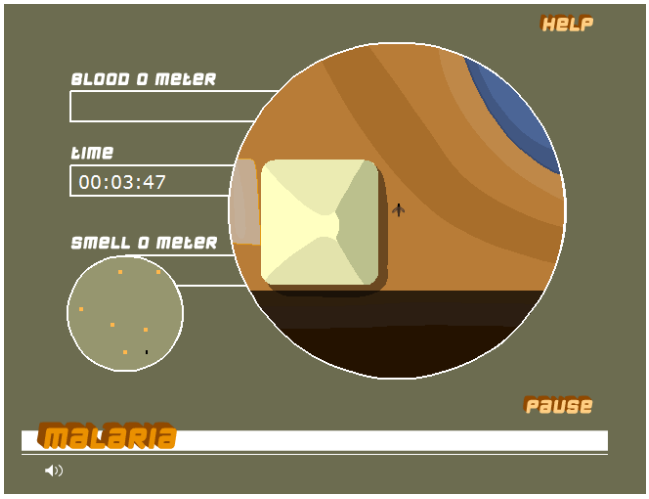
Game dengan tipe *gameplay action* sudah tidak asing lagi dalam dunia game, bahkan *game action* merupakan salah satu gameplay yang lebih banyak dimainkan dibanding genre lainnya [4]. *Game action* sangat banyak diminati karena pengguna merasa permainan tersebut seru, mudah, banyak gerakan, efek visual yang menarik dan penuh tantangan, beberapa diantaranya yaitu seperti *Call of Dutty*, *Halo*, *Brutal Street*, *Batman Arkham City*, dan lain-lain.

Saat ini *game* edukasi merupakan alat bantu yang efektif untuk penyampaian pengetahuan dan pembelajaran, beberapa diantaranya seperti *Democracy, Math Blaster, Mosquito and Parasite Game, Stop Malaria Now, The Blood Typing,* dan lain-lain. Namun pada kenyataanya *game* edukasi memang termasuk salah satu game yang jarang dimainkan oleh pengguna [4].

Berdasarkan hasil observasi yang dilakukan terhadap dua *game* yang sejenis, baik berdasarkan tema dan genre diketahui bahwa terdapat beberapa kesamaan terhadap *game* yang akan dibangun. *Game* sejenis yang akan dianalisis adalah sebagai berikut:

1. *Mosquito and Parasite Games*

Mosquito and Parasite Game adalah *game* bertemakan malaria yang dibangun menggunakan flash. *Game* ini bergenre edukasi dibuat pada tahun 1902 dan mendapatkan *Nobel Prize* di *Physiology or Medicine*. Tampilan dari game tersebut dapat dilihat pada Gambar 3.1.



Gambar 3.1 Tampilan *Mosquito and Parasite Games*

Pada *game* ini pemain diberi pengetahuan tentang bagaimana malaria ditularkan kepada manusia, apa jenis nyamuk dan parasit dari penyebaran malaria, dengan obat kimia apa nyamuk malaria dapat dibunuh, dan cara yang murah dan sederhana untuk mencegah nyamuk menggigit manusia. Skenario permainan pada *Mosquito and Parasite Game* dapat disusun oleh beberapa hal sebagai berikut :

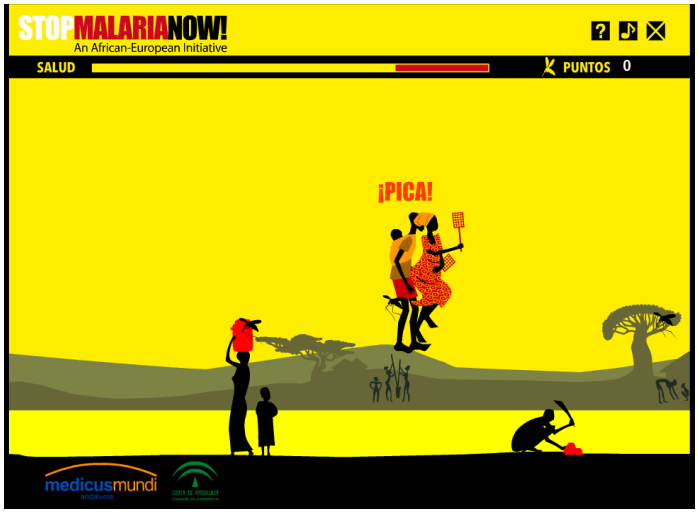
a. Misi Permainan

Tabel 3.1 Misi Permainan Pada *Mosquito and Parasite Game*

Misi Permainan
Pemain mengendalikan nyamuk atau parasit dan mencoba untuk menemukan manusia untuk menggigit, mengisap darahnya dan menginfeksikan virus malaria kepada manusia.

2. *Stop Malaria Now*

Stop Malaria Now adalah game online bertemakan malaria yang merupakan bentuk pengembangan dari *Member Medicusmundi*. Tujuannya untuk memberikan pelajaran lebih lanjut tentang nyamuk malaria dan pencegahannya. Permainan ini berbahasa spanyol, sulit untuk mengerti apa yang disampaikan tentang pengetahuan malaria akan tetapi mudah bagi semua orang di dunia untuk mengikuti langkah-langkah dari permainan ini. Tampilan dari *game* tersebut dapat dilihat pada Gambar 3.2.



Gambar 3.2 Tampilan *Game Stop Malaria Now*

Game ini bercerita tentang sebuah keluarga di Afrika yang ingin menuju ke pusat kesehatan untuk meminta bantuan karena terkena wabah dan infeksi malaria. Cara memainkan *game* ini dengan mengendalikan keluarga tersebut yang terdiri dari ayah dan ibu, kemudian memukul nyamuk-nyamuk yang ada saat melewati jalan menggunakan raket (pemukul nyamuk). Skenario permainan pada *Stop Malaria Now* dapat disusun oleh beberapa hal sebagai berikut :

a. Misi Permainan

Tabel 3.2 Misi Permainan Pada *Stop Malaria Now*

Misi Permainan
Pemain mengendalikan *African's Family* dan mencoba untuk menuju ke pusat kesehatan dan membrantas nyamuk malaria menggunakan raket (alat pemukul nyamuk).

Setelah menganalisis *game* dengan genre yang sejenis, lalu akan dibandingkan elemen-elemen dasar dari sebuah *game* yang terdapat pada *game* sejenis tersebut seperti yang terlihat pada tabel 3.3.

Tabel 3.3 Elemen yang Terdapat Pada *Game* Sejenis

Elemen	*Mosquito and Parasite*	*Stop Malaria Now*
Karakter	Nyamuk Anopheles	Keluarga Afrika

Rules	Memenuhi *rules game edukasi* dengan kesederhanaan aturan dalam permainan	Memenuhi *rules game edukasi* dengan kesederhanaan aturan dalam permainan
Level	Terdapat 5 level	Terdapat 6 level
Nyawa	Ada	Ada
Score	-	Ada
Boss	-	-
Penyajian Materi	Ada	Ada
Exit and Level Warps	Ada	Ada
Victory Conditions	Ada	Ada
Interaction Models	Memenuhi	Memenuhi
Perspective	*Top Down*	*Side Scrolling*
User Interface Design	Memenuhi KISS (*Keep It Simple, Stupid*), memiliki UI yang sederhana	Memenuhi KISS (*Keep It Simple, Stupid*), memiliki UI yang sederhana

Berdasarkan analisis yang telah dilakukan, maka ditarik kesimpulan bahwa *game* edukasi bertemakan malaria mulai digunakan sebagai alat pendidikan dalam bidang kesehatan khususnya dalam membrantas malaria. Namun *game* yang telah ada kurang detail atau kurang begitu nyata dalam mensimulasikan pembrantasan dan pencegahan malaria. Belum adanya simulasi program 3M yaitu menguras, menutup dan mengubur yang menjadi program dalam pencegahan malaria juga menjadi kekurangan dua *game* tersebut. *Game* yang ada hanya memberikan pengetahuan malaria, bagaimana membrantas malaria secara sederhana, tanpa memberikan pengetahuan bagaimana mencegah

perkembangbiakan malaria yang terdapat dalam program 3M. Untuk itu, maka penulis merasa tertarik untuk membangun *game* edukasi malaria dengan memperjelas pembrantasan dan pencegahan malaria menggunakan program 3M.

3.1.2 Analisis *Game Save Family From Malaria*

Game edukasi yang akan dibangun berjudul *Save Family From Malaria* yang bergenre *education-action*. *Game* ini dibangun dengan mengaplikasikan teknologi sebagai jembatan untuk menyajikan materi mengenai bahaya dan cara pembrantasan malaria secara interaktif dan edukatif, khususnya ditujukan kepada anak-anak usia 9-10 tahun. *Game* ini dibangun dengan grafis 2 dimensi dan menggunakan sistem *single player*.

3.1.3 *Storyline*

Game edukasi "*Save Family From Malaria*" menceritakan keteladanan seorang anak SD bernama Bambang. Keluarganya terkena penyakit malaria, ibu dan satu adiknya dilarikan ke rumah sakit karena demam yang sangat tinggi. Sementara itu, Bambang diberitahu oleh ayah dan dokter tentang penyakit yang diderita ibu dan adiknya tersebut. Penyebab penyakit tersebut yaitu disebabkan gigitan nyamuk malaria (*Anopheles*). Setelah itu, Bambang membaca pamflet yang ditujukan dokter, kemudian berimajinasi tentang apa yang ada pada pamflet tersebut seperti bahaya malaria, dan pencegahan menggunakan kelambu. program 3M, dan DDT.

Dalam imajinasinya dia berjalan menyusuri tempat sarang nyamuk dan membrantas mereka dengan menyemprotkan obat nyamuk. Dia juga melakukan apa yang ada pada program 3M (menguras, menutup, mengubur), memasang kelambu dan diakhir cerita melawan ratu nyamuk yang memiliki parasit paling berbahaya dan mematikan menggunakan semprotan DDT.

Berikut topik materi yang akan disampaikan pada game edukasi "*Save Family From Malaria* " :

a. Level 1 "Memasang Kelambu"

Pada level ini akan di sampaikan materi mengenai penanganan terhadap nyamuk *Anopheles* menggunakan semprotan obat nyamuk yang masyarakat

umum biasa gunakan dan bagaimana pencegahan dengan memasang kelambu. Skenario permainan pada level ini adalah sebagai berikut :

1. Kontrol Permainan

Tabel 3.4 Kontrol Permainan pada Level 1

Kontrol Permainan
1. *Keyboard* dan *mouse* merupakan alat input untuk mengontrol karakternya. 2. Pemain dapat menyemprot nyamuk-nyamuk dengan menekan tombol *mouse* sebelah kiri. 3. Pemain dapat mengontrol karakter ke kiri, kanan, dengan menggunakan panah kiri, kanan, dan atas untuk *jump* yang terdapat pada *keyboard*. 4. Pemain dapat menggunakan *mouse* untuk memasang kelambu ke ke tempat tidur.

2. Misi Permainan

Tabel 3.5 Misi Permainan pada Level 1

Misi Permainan
Pemain mengendalikan Bambang dan mencoba untuk membrantas nyamuk malaria menggunakan semprotan, memasang kelambu di tempat tidur dan menuju pintu hutan.

3. *Gameplay* Level 1

Tabel 3.6 *Gameplay* Level 1

Gameplay
1. Pemain berada di sisi kiri layar . 2. Pemain harus mengendalikan Bambang dan membrantas nyamuk. 3. Pemain harus menyemprot nyamuk sebanyak mungkin untuk

mendapatkan *score.* 4. Pemain harus menghindari gigitan nyamuk, jika terkena nyamuk tersebut maka darah pemain akan berkurang. 5. Jika darah pemain habis maka pemain dianggap kalah. 6. Pemain harus menemukan kelambu dan menyentuhnya. 7. Pemain harus memasangkan semua kelambu yang sesuai dengan tempat tidur. 8. Jika tidak menyelesaikan memasang kelambu, pemain tidak dapat membuka pintu hutan dan dianggap kalah. 9. Pemain harus menuju ke pintu hutan dan membuka pintu. 10. Jika berhasil masuk pintu hutan, maka pemain dianggap menang dan mendapatkan *score* sesuai dengan nyamuk yang dibrantas dan kelambu yang dipasang.

b. Level 2 “Menguras Tempat Air”

Pada level ini akan di sampaikan salah satu materi dari 3M yaitu menguras. Menguras adalah penanganan terhadap perkembangbiakan nyamuk *Anopheles* yang biasanya menaruh telurnya di genangan air yang kotor dan lama. Skenario permainan pada level ini adalah sebagai berikut :

1. Kontrol Permainan

Tabel 3.7 Kontrol Permainan pada Level 2

Kontrol Permainan
1. *Keyboard* dan *Mouse* merupakan alat input untuk mengontrol karakternya. 2. Pemain dapat menyemprot nyamuk dengan menekan tombol mouse sebelah kiri. 3. Pemain dapat mengambil sikat pembersih dengan menyentuh sikat.

4. Pemain dapat mengontrol karakter ke kiri, kanan, dengan menggunakan panah kiri, kanan, dan atas untuk *jump* yang terdapat pada *keyboard*. 5. Pemain dapat menggunakan mouse untuk membuka tutup tempat air, menyikat kotoran, dan menutup keran.

2. Misi Permainan

Tabel 3.8 Misi Permainan pada Level 2

Misi Permainan
Pemain mengendalikan Bambang dan mencoba untuk membrantas nyamuk, mengambil sikat pembersih, dan menguras tempat air.

3. *Gameplay* Level 2

Tabel 3.9 *Gameplay* Level 2

Gameplay
1. Pemain berada di tepi kiri layar . 2. Pemain harus mengendalikan Bambang untuk membrantas nyamuk dan menguras tempat air diujung level *game*. 3. Pemain harus menyemprot nyamuk untuk mendapatkan *score*. 4. Pemain harus menghindari gigitan nyamuk, jika terkena nyamuk tersebut maka darah pemain akan berkurang. 5. Jika darah pemain habis maka pemain dianggap kalah. 6. Pemain harus mengambil sikat terlebih dahulu sebelum menyentuh tempat air untuk dapat mengurasnya, jika tidak maka tidak dapat menguras. 7. Jika pemain dapat menguras tempat air, maka pemain dapat menuju ke pintu hutan.

8. Pemain harus menuju ke pintu hutan dan membuka pintu. 9. Jika berhasil masuk pintu hutan, maka pemain dianggap memenangkan level ini dan mendapatkan *score* sesuai dengan nyamuk yang dibrantas dan tempat air yang dikuras.

c. Level 3 “Menutup Tempat Air”

Pada level ini akan di sampaikan salah satu materi dari 3M yaitu menutup. Menutup adalah penanganan terhadap perkembangbiakan nyamuk *Anopheles* yang biasanya menaruh telurnya di genangan air yang kotor dan lama. Skenario permainan pada level ini adalah sebagai berikut :

1. Kontrol Permainan

Tabel 3.10 Kontrol Permainan pada Level 3

Kontrol Permainan
1. *Keyboard* dan *Mouse* merupakan alat input untuk mengontrol karakternya. 2. Pemain dapat menyemprot nyamuk dengan menekan mouse sebelah kiri. 3. Pemain dapat mengambil penutup tempat air dengan menyentuhnya. 4. Pemain dapat mengontrol karakter ke kiri, kanan, dengan menggunakan panah kiri, kanan, dan juga atas untuk *jump* yang terdapat pada *keyboard*. 5. Pemain dapat menutup tempat air yang sesuai dengan menggunakan *mouse*.

2. Misi Permainan

Tabel 3.11 Misi Permainan pada Level 3

Misi Permainan
Pemain mengendalikan Bambang dan mencoba untuk membrantas nyamuk, mengambil penutup, dan menutup tempat air.

3. *Gameplay* Level 3

Tabel 3.12 *Gameplay* Level 3

Gameplay
1. Pemain berada di tepi kiri layar .
2. Pemain harus dapat mengendalikan Bambang membrantas nyamuk dan menutup tempat air.
3. Pemain harus menyemprot nyamuk untuk mendapatkan *score*.
4. Pemain harus menghindari gigitan nyamuk, jika terkena nyamuk tersebut maka darah pemain akan berkurang.
5. Jika darah pemain habis maka pemain dianggap kalah.
6. Pemain harus mengambil penutup terlebih dahulu sebelum menyentuh tempat air untuk dapat menutupnya, jika tidak maka tidak dapat menutup tempat air.
7. Pemain harus menutup semua tempat air untuk mendapatkan *score* dan membuka pintu hutan, jika tidak maka tidak dapat membuka pintu hutan.
8. Pemain harus menuju ke pintu hutan dan membuka pintu.
9. Jika berhasil masuk pintu hutan, maka pemain dianggap memenangkan level ini dan mendapatkan *score* sesuai dengan nyamuk yang dibrantas dan tempat air yang ditutup.

d. Level 4 “Mengubur Wadah Kosong”

Pada level ini akan di sampaikan salah satu materi dari 3M yaitu mengubur. Mengubur adalah penanganan terhadap perkembang biakan nyamuk *Anopheles* yang biasanya menaruh telurnya di wadah kosong yang basah atau berair. Skenario permainan pada level ini adalah sebagai berikut :

1. Kontrol Permainan

Tabel 3.13 Kontrol Permainan pada Level 4

Kontrol Permainan
1. *Keyboard* dan *Mouse* merupakan alat input untuk kontrol karakternya. 2. Pemain dapat menyemprot nyamuk menekan *mouse* sebelah kiri. 3. Pemain dapat mengambil wadah kosong dengan menyentuh wadah. 4. Pemain dapat mengontrol karakter ke kiri, kanan, dengan menggunakan panah kiri, kanan, dan juga atas untuk *jump* yang terdapat pada *keyboard*. 5. Pemain dapat meletakan dan mengubur wadah kosong dengan menggunakan *mouse*.

2. Misi Permainan

Tabel 3.14 Misi Permainan pada Level 4

Misi Permainan
Pemain mengendalikan Bambang dan mencoba untuk membrantas nyamuk malaria, mengambil, meletakan, dan mengubur wadah kosong.

3. *Gameplay* Level 4

Tabel 3.15 *Gameplay* Level 4

Gameplay
1. Pemain berada di kiri layar . 2. Pemain harus dapat mengendalikan Bambang dan membrantas nyamuk untuk mendapatkan *score*. 3. Pemain harus menghindari gigitan nyamuk, jika terkena nyamuk tersebut maka darah pemain akan berkurang. 4. Jika darah pemain habis maka pemain dianggap kalah.

5. Pemain harus mengambil wadah terlebih dahulu sebelum menyentuh lubang untuk dapat menguburnya, jika tidak maka tidak dapat mengubur. 6. Pemain harus mengubur semua wadah untuk mendapatkan *score* dan membuka pintu hutan, jika tidak maka tidak dapat membuka pintu hutan. 7. Pemain harus menuju ke pintu hutan dan membuka pintu. 8. Jika berhasil masuk pintu hutan, maka pemain dianggap memenangkan level ini dan mendapatkan *score* sesuai dengan nyamuk yang dibrantas dan wadah yang dikubur.

e. Level 5 “Semprot dengan DDT”

Pada level ini akan di sampaikan materi mengenai pembrantasan nyamuk *Anopheles* yang telah mewabah di sebuah daerah, dengan cara menyemprotkan DDT oleh petugas khusus ke rumah-rumah warga. Pada gambaran ceritanya Bambang menyemprotkan DDT ke nyamuk dan melawan ratu nyamuk malaria di akhir level. Skenario permainan pada level ini adalah sebagai berikut :

1. Kontrol Permainan

Tabel 3.16 Kontrol Permainan pada Level 5

Kontrol Permainan
1. *Keyboard* dan *Mouse* merupakan alat input untuk mengontrol karakternya. 2. Pemain dapat menyemprotkan obat nyamuk dan DDT menggunakan tombol *mouse* sebelah kiri. 3. Pemain dapat mengontrol karakter ke kiri, kanan, menggunakan panah kiri, kanan, dan juga atas untuk *jump* yang terdapat pada *keyboard*.

2. Misi Permainan

Tabel 3.17 Misi Permainan pada Level 5

Misi Permainan
Pemain mengendalikan Bambang dan mencoba untuk membrantas perkembanganbiakan nyamuk malaria dan Ratu dengan menyemprotkan obat nyamuk dan DDT.

3. *Gameplay* Level 5

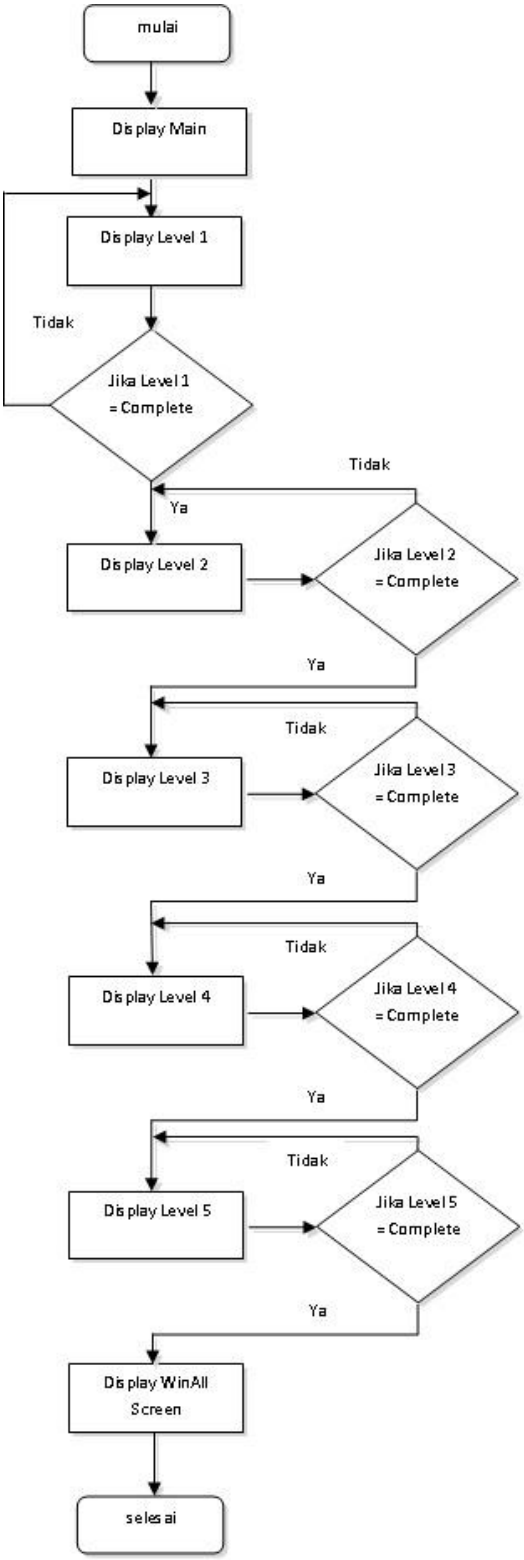
Tabel 3.18 *Gameplay* Level 5

Gameplay
1. Pemain berada di tepi kiri layar. 2. Pemain menyemprotkan obat nyamuk dan DDT untuk membrantas nyamuk dan Ratu. 3. Pemain harus membrantas semua nyamuk dan mengalahkan Ratu nyamuk, jika tidak maka dianggap kalah. 4. Pemain harus menghindari dan membrantas nyamuk dan Ratu, jika tidak maka darah akan berkurang. 5. Jika darah telah habis maka pemain dianggap kalah. 6. Jika pemain menyelesaikan level ini maka mendapat *score* sebanyak nyamuk yang dibunuh ditambah *score* Ratu dan dianggap menang.

3.1.4 *Gameplay*

Tugas utama pemain dalam memainkan *game* ini yaitu melalui level-level secara bertahap. Setiap bagian level menyediakan bentuk permainan dan *stage* yang berbeda-beda yang harus di selesaikan oleh pemain.

Pemain memulai permainan dengan level 1, dimana nantinya pemain harus menyelesaikan level tersebut terlebih dahulu untuk kemudian melanjutkan permainan ke level selanjutnya. Begitu juga dengan level ke 2, 3, 4 dan 5. Hal ini bertujuan untuk memberikan penjelasan detail tentang kronologi dari cerita game dan urutan penanggulan malaria. Untuk lebih jelas dapat dilihat pada *flowchart gameplay* dibawah ini.



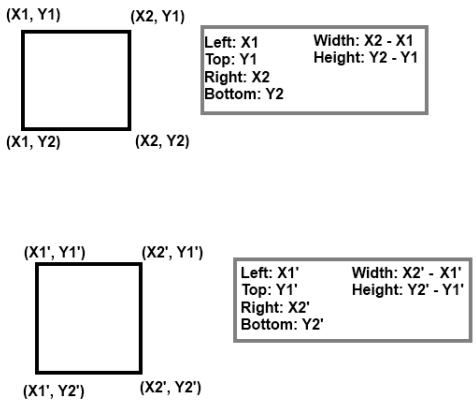
Gambar 3.3 *Flowchart Gameplay*

3.1.5 Analisis Pengunaan *Collision Detection*

Dalam pembuatan *game Save Family From Malaria* ini diperlukan adanya penggunaan *collision detection*. Penggunaan diantaranya untuk memicu event pada pergerakan player, NPC dan perpindahan *state*.

Teknik yang digunakan disini adalah *Rectangles Collision Detection*. Teknik ini merupakan pendeteksian tabrakan menggunakan bound berbentuk persegi, dimana setiap sprite atau objek yang dibuat dalam game memiliki bound masing-masing. Cara ini cukup mudah karena hanya perlu membandingkan apakah ada *intersection* (irisan) pada kedua objek pada kordinat tertentu.

Cara mengetahui bagaimana tabrakan terjadi, pertama adalah menentukan titik koordinat dari masing-masing objek. Contoh, ada dua buah persegi panjang dengan nama R1 dan R2. R1 mempunyai koordinat: (X1,Y1), (X1,Y2), (X2,Y1), (X2,Y2). Sedangkan koordinat R2: (X1', Y1'), (X1', Y2'), (X2',Y1'), (X2',Y2').

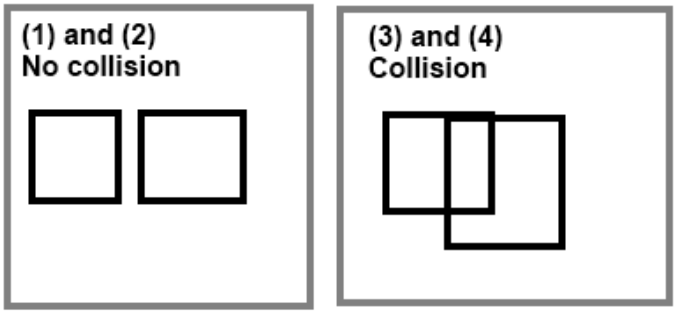


Gambar 3.4 Koordinat Persegi Panjang

Pertama, memakai skenario *Horizontal Collision*. Seperti di bawah ini:

1. Memeriksa apakah tepi kiri R2 ada diantara tepi kiri dan kanan R1. Artinya, jika X1 <X1'<X2 maka terjadi tabrakan. X1 <X1', tapi X1'> X2 maka tidak ada tabrakan.

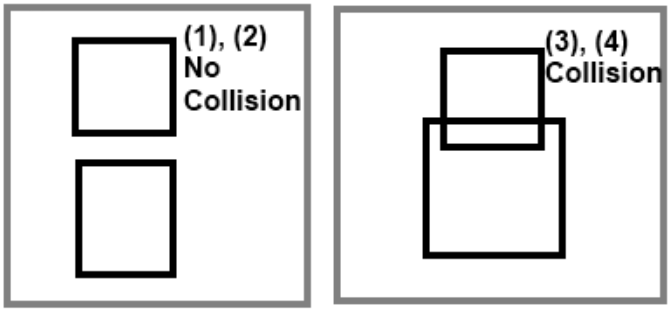
2. Memeriksa apakah tepi kanan R2 ada diantara tepi kiri dan kanan R1. Artinya, jika X1 <X2'<X2 maka terjadi tabrakan.
3. (1), tepi ada tabrakan karena X1 <X1'<X2.
4. (2), tepi ada tabrakan karena X1 <X2'<X2.



Gambar 3.5 *Horizontal Collision*

Selanjutnya adalah *Vertical Collision*, skenarionya sebagai berikut:

1. Memeriksa tepi atas R2, tabrakan jika Y1> Y1' > Y2. Tidak ada tabrakan dalam hal ini karena Y1' > Y2.
2. Memeriksa tepi bawah R2, tabrakan jika Y1> Y2' > Y2. Tidak ada tabrakan dalam hal ini karena Y2' < Y1.
3. Mirip dengan no. (1), tepi kali ini Y1> Y1' > Y2 sehingga ada tabrakan.
4. Seperti no.(2), tepi Y1> Y2' > Y2 sehingga ada tabrakan.



Gambar 3.6 *Vertical Collision*

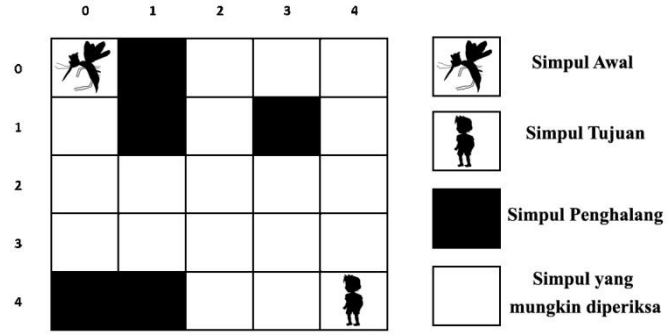
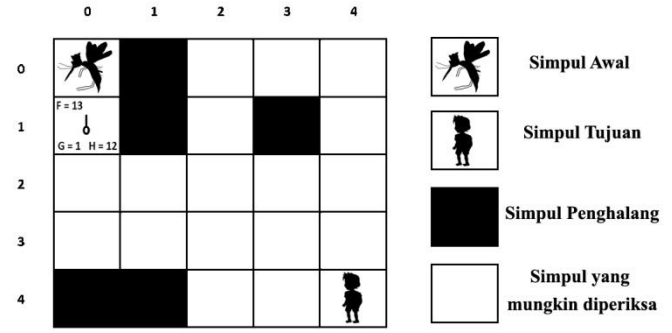
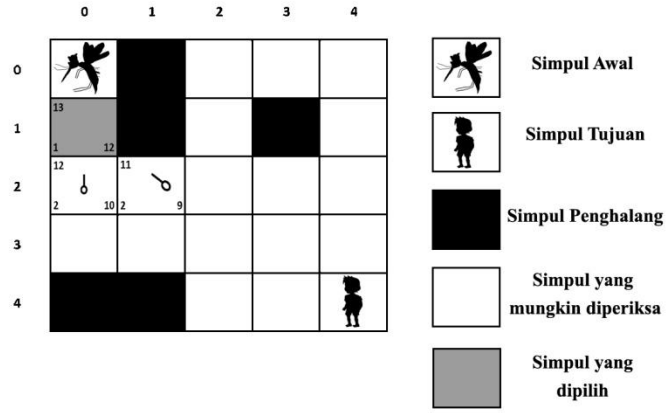
3.1.6 Analisis Penerapan Algoritma A*

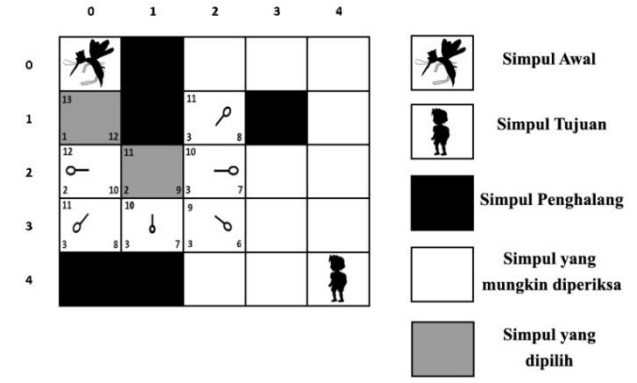
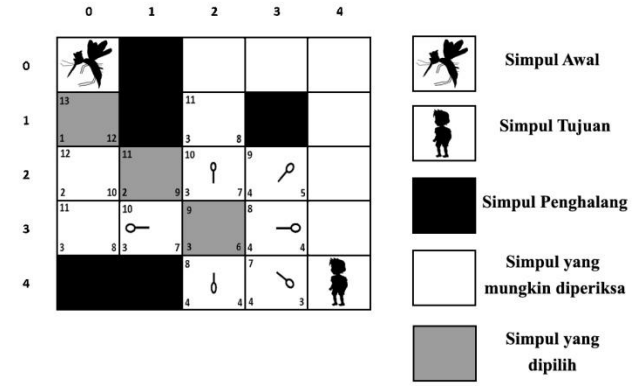
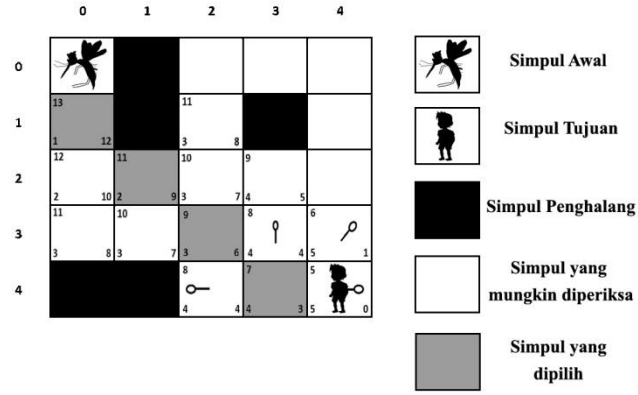
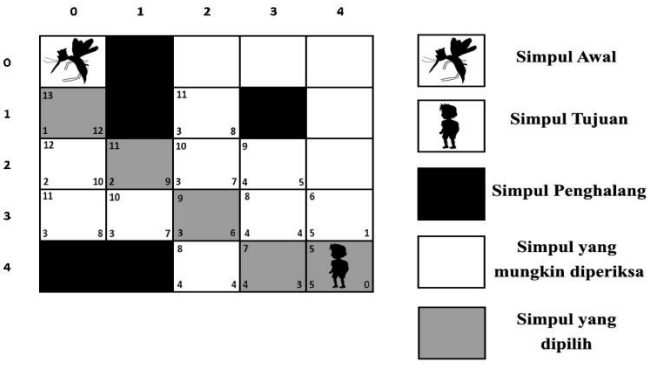
Pada *game Save Family From Malaria*, algoritma A* diterapkan pada ratu nyamuk dengan beberapa *node* pada peta. Struktur data yang digunakan adalah *array* dua dimensi dengan menentukan bobot sesuai koordinat setiap *node*. Pada penentuan bobot setiap *node* akan diberikan nilai sesuai dengan jarak

terdekat ke tujuan. *Node* terjauh dari tujuan akan diberikan bobot yang kecil, sedangkan *node* terdekat dengan tujuan akan diberikan bobot yang lebih besar berdasarkan koordinat dari tiap *node*.

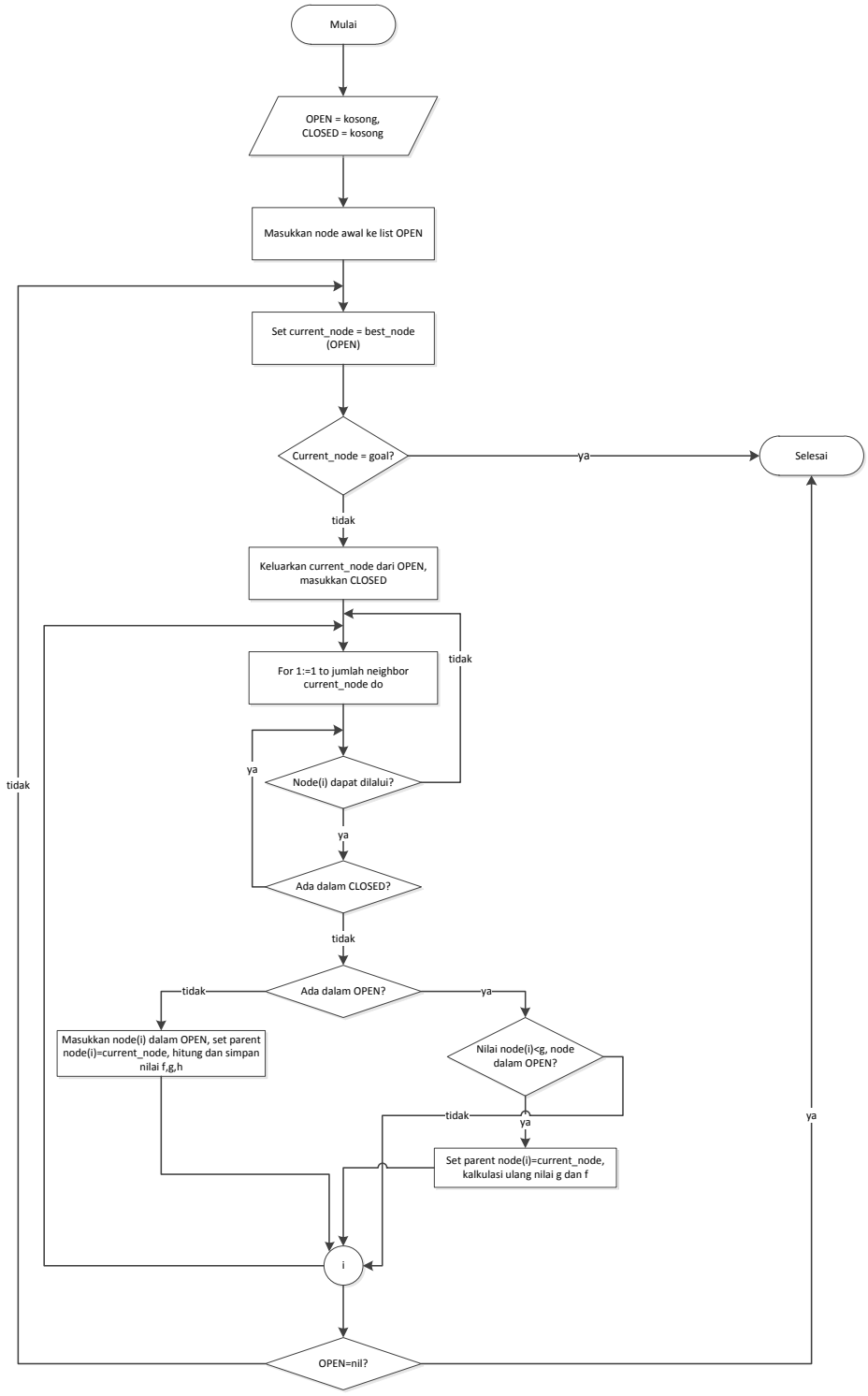
Penggunaan algoritma A* pada *Save Family From Malaria* akan disimulasikan dengan menggunakan matriks dengan ordo 5x5. Pencarian jalur terpendek dengan menggunakan algoritma A* dari simpul awal menuju simpul tujuan akan diperlihatkan pada Tabel 3.5 di bawah ini.

Tabel 3.19 Skenario Langkah Pencarian Algoritma A*

Simulasi	Keterangan
	Di *OPEN* hanya terdapat satu simpul yaitu simpul awal (0,0), maka simpul tersebut dipilih sebagai *Best Node* dan dipindahkan ke *CLOSED*.
	Bangkitkan semua suksesor dari (0,0), lalu hitung semua nilai F dari simpul tersebut. Karena dua seksesor (1,0) dan (1,1) merupakan penghalang maka diabaikan. Tersisa hanya satu *OPEN* yaitu (0,1) kemudian dipilih menjadi *Best Node*. Oleh karena itu *parent* diubah dari (0,0) menjadi (0,1) dan simpul (0,0) dimasukkan ke dalam *CLOSED*.
	Bangkitkan semua suksesor dari (0,1) yaitu (0,2) dan (1,2), karena keduanya belum pernah di *OPEN* maupun *CLOSED*, maka kedua simpul dimasukkan ke dalam *OPEN*. Kemudian diperiksa dan dibandingkan nilai F dari kedua suksesor tersebut. Simpul (1,2) memiliki nilai F lebih kecil, maka simpul (1,2) dipilih sebagai *BestNode* dan dimasukkan ke dalam *CLOSED*. Kemudian ubah *parent* menjadi simpul (1,2).

	Bangkitkan semua suksesor dari simpul (1,2), karena semua suksesor belum pernah ada di *CLOSED*, maka suksesor tersebut dimasukkan ke dalam *OPEN*. Lalu diperiksa dan dibandingkan semua nilai F dari suksesor tersebut. Simpul (2,3) memiliki nilai F paling kecil dari semua suksesor, maka simpul (2,3) dipilih sebagai *BestNode* dan dimasukkan ke dalam *CLOSED*. Kemudian *parent* diubah dari (1,2) menjadi (2,3).
	Bangkitkan semua suksesor dari simpul (2,3). Kemudian masukkan semua suksesor ke dalam *OPEN*. Lalu diperiksa dan dibandingkan nilai F dari suksesor-suksesor tersebut. Setelah diperiksa, simpul (3,4) memiliki nilai F paling kecil, maka simpul tersebut dipilih sebagai *BestNode* dan dimasukkan ke dalam *CLOSED*. Kemudian *parent* diubah dari simpul (2,3) menjadi (3,4).
	Bangkitkan semua suksesor dari simpul (3,4). Kemudian masukkan semua suksesor ke dalam *OPEN*. Kemudian diperiksa dan dibandingkan nilai F dari semua suksesor tersebut. Setelah diperiksa, simpul (4,4) memiliki nilai F paling kecil diantara semua suksesor. Kemudian simpul (4,4) dijadikan sebagai *BestNode* dan dimasukkan ke dalam *CLOSED*. Kemudian *parent* diubah dari simpul (3,4) menjadi (4,4).
	Simpul (4,4) teridentifikasi sebagai simpul tujuan, maka langkah pencarian dihenti- kan dan solusi ditemukan. Langkah terpendek untuk mencapai tersebut dapat dilihat pada simulasi di- samping.

Untuk lebih jelas mengenai pencarian dengan menggunakan algoritma A* dapat dilihat dalam diagram *flowchart* pada Gambar 3.7.



Gambar 3.7 Diagram *Flowchart* Algoritma A*

3.1.7 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem [17]. Spesifikasi ini meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut dapat diimplementasikan.

Pada analisis kebutuhan sistem non fungsional ini dijelaskan analisis kebutuhan perangkat keras, analisis kebutuhan perangkat lunak, dan analisis pengguna.

3.1.7.1 Analisis Kebutuhan Perangkat Keras

Perangkat keras yang digunakan untuk membangun aplikasi *game Save family From Malaria* adalah sebagai berikut :

1. *Processor* 3.20 Ghz
2. *Harddisk* 500GB
3. *RAM* 2 GB
4. *VGA* 2 GB
5. *LCD Monitor* dengan resolusi 1366 x 768

Sedangkan perangkat keras minimum yang dibutuhkan untuk menjalankan aplikasi *game* ini adalah sebagai berikut :

1. *Processor* 1.3 *Ghz*
2. *RAM* 512 MB
3. *Harddisk* 75 MB
4. Kartu grafik dengan DirectX 9 *compatible sound card*
5. *CRT monitor* dengan resolusi 800 x 600

3.1.7.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan untuk membangun aplikasi *game Save Family From Malaria* ini adalah sebagai berikut :

1. Sistem operasi *Window* 7 *Ultimate 32-bit*
2. *Adobe Flash CS3 Professional*
3. *Flash Player 9.0*

Sedangkan perangkat lunak yang dibutuhkan pengguna untuk menjalankan aplikasi *game* ini adalah sebagai berikut :

1. *Flash Player 9.0 atau versi yang terbaru*

3.1.7.3 Analisis Pengguna

Game yang akan dibuat ditujukan khusus untuk siswa siswi sekolah dasar usia 9-10 tahun. Analisis pengguna menggunakan observasi berdasarkan analisis *game* sejenis bergenre *action* yang diminati oleh kebanyakan pemain. Maka analisis penggunanya berdasarkan elemen yang ada adalah sebagai berikut :

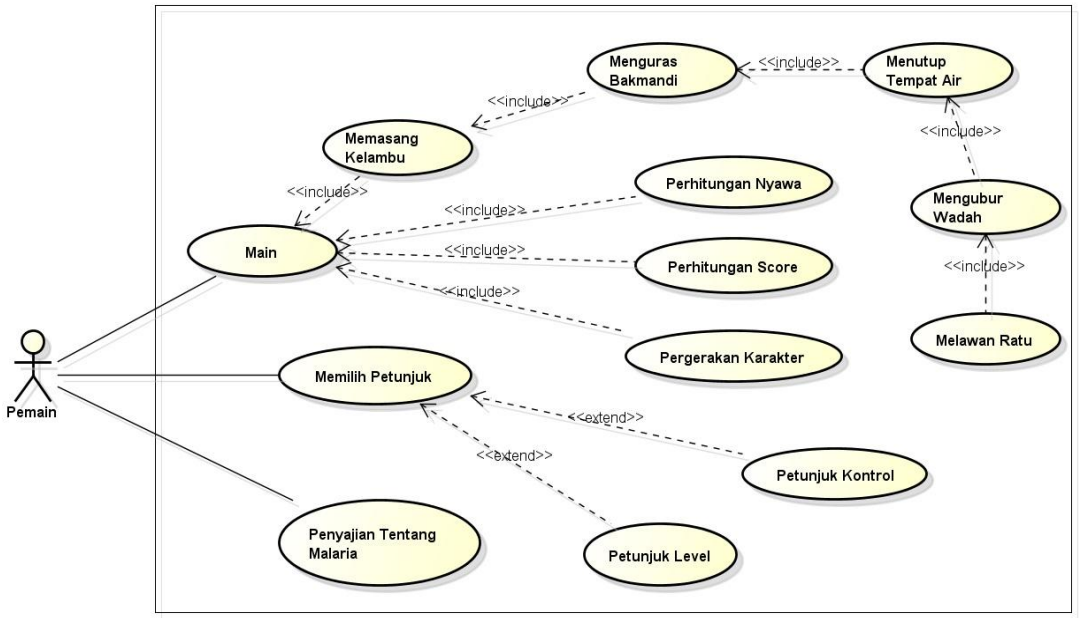
1. Pemain menyukai *game action* yang mempunyai beberapa level dalam permainannya.
2. Pemain menyukai permainan yang mempunyai *checkpoints*. Hal ini dikarenakan pemain akan merasa bosan apabila harus mengulang level dari awal ketika mati.
3. Pemain menyukai karakter yang mempunyai darah pada permainan. Artinya, ketika karakter pemain terkena sekali serangan musuh, permainan tidak akan langsung berakhir.
4. Pemain menyukai permainan yang memiliki *scores*. Hal ini berhubungan dengan *achieve* bagi pemain untuk memainkan permainannya
5. Pemain menyukai permainan yang memiliki bos atau tantangan baru pada setiap akhir dari levelnya, karena dengan adanya hal tersebut akan menambah kesulitan dan tantangan dari permainan.
6. *Dungeon exit and level warps*, pemain menyukai permainan dengan adanya sebuah konten untuk fase perpindahan level, karena hal ini menyebabkan misi dari setiap level menjadi jelas.
7. Pemain menginginkan *victory conditions* pada permainan. Elemen ini dibutuhkan untuk menyelesaikan cerita dari permainan yang sedang dimainkan.
8. Pemain menginginkan *user interface* yang sederhana dan mudah dimengerti, tetapi memenuhi kebutuhan pemain.

3.1.8 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.

3.1.8.1 *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Perancangan proses-proses dapat dijelaskan pada gambar 3.8.



Gambar 3.8 *Use Case Diagram Game Save Family From Malaria*

3.1.8.2 Definisi *Actor*

Definisi *actor* berfungsi untuk menjelaskan aktor-aktor yang terdapat pada *use case diagram*. Terdapat satu *actor* dalam diagram *use case game Save Family From Malaria* yaitu pemain sebagai orang yang memainkan *game*.

3.1.8.3 Definisi *Use Case*

Definisi *use case* berfungsi untuk menjelaskan fungsi *use case* yang terdapat pada *use case diagram*. Definisi *use case* diterangkan pada Tabel 3.20.

Tabel 3.20 Definisi *Use Case*

No	*Use case*	Deskripsi
1	Main	Fungsi untuk memainkan game
2	Memilih Petunjuk	Fungsi untuk memilih petunjuk
3	Penyajian Tentang Malaria	Fungsi untuk menampilkan pengetahuan tentang malaria.
4	Perhitungan Nyawa	Fungsi untuk menghitung nyawa
5	Perhitungan *Score*	Fungsi untuk menghitung *score*
6	Pergerakan Karakter	Fungsi untuk menggerakkan karakter
7	Petunjuk Kontrol	Fungsi untuk mengetahui petunjuk kontrol dari keyboard dan mouse dalam permainan
8	Petunjuk Level	Fungsi untuk mengetahui petunjuk atau aturan dari setiap level permainan
9	Memasang Kelambu	Fungsi untuk memainkan materi memasang kelambu
10	Menguras Bak Mandi	Fungsi untuk memainkan materi menguras bak mandi
11	Menutup Tempat Air	Fungsi untuk memainkan materi menutup tempat air
12	Mengubur Wadah	Fungsi untuk memainkan materi mengubur wadah kosong
13	Melawan Ratu	Fungsi untuk memainkan materi melawan ratu yang memiliki parasit yang terbahaya

3.1.8.4 Skenario *Use Case*

Skenario pada diagram *use case* berisi mengenai alur kejadian yang di dalamnya menjelaskan urutan interaksi antara *actor* dengan *use case* tersebut.

1. Skenario *Use Case* Main

Skenario *use case* ini menjelaskan interaksi antara *actor* yaitu pemain, dengan *use case* main yang dijelaskan pada Tabel 3.21 di bawah ini.

Tabel 3.21 Skenario *Use Case* Main

Identifikasi	
No	1
Nama	Main
Tujuan	Untuk dapat memulai permainan
Actor	Pemain
Skenario Utama	
Kondisi Awal	Menu Utama ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Memilih mulai permainan	
	2. Memasukkan pemain ke *scene* permainan
	3. Memunculkan nyamuk
	4. Apabila pemain mati, maka menampilkan screen *game over*
5. Jika menekan tombol kiri, kanan, atas dan bawah	
	6. Menggerakkan karakter
7. Jika menekan *mouse* kiri	
	8. Mengeluarkan asap obat nyamuk
	9. Apabila musuh terkena tembakan dan mati, maka skor ditambahkan
10. Mengambil item syarat game materi	
	11. Menghilangkan item syarat 12. Membuka *game* materi yang terkunci
13. Menyentuh item game materi	
	14. Memunculkan permainan yang berkaitan dengan materi
15. Menyelesaikan permainan sesuai materi	
	16. Apabila permainan materi yang berkaitan dapat diselesaikan mendapat bonus skor 17. Pintu hutan dapat dibuka
18. Mencapai pintu hutan	
	19. Menghapus nyamuk
	20. Menampilkan *win screen*
	21. Memasukkan pemain ke level selanjutnya
Kondisi Akhir	Menampilkan permainan

2. Skenario *Use Case* Memilih Petunjuk

Skenario *use case* ini menjelaskan interaksi antara *actor* yaitu pemain, dengan *use case* memilih petunjuk yang dijelaskan pada Tabel 3.22 di bawah ini.

Tabel 3.22 Skenario *Use Case* Memilih Petunjuk

Identifikasi	
No	2
Nama	Memilih Petunjuk
Tujuan	Menampilkan petunjuk permainan
Actor	Pemain
Skenario Utama	
Kondisi Awal	Menu Utama ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu petunjuk	
	2. Menampilkan pilihan petunjuk
3. Jika Memilih petunjuk kontrol	
	4. Menampilkan petunjuk kontrol
5. Jika Memilih petunjuk *level*	
	6. Menampilkan petunjuk level
Kondisi Akhir	Menampilkan menu pilihan petunjuk

3. Skenario *Use Case* PenyajianTentang Malaria

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* penyajian tentang malaria yang dijelaskan pada Tabel 3.23 di bawah ini.

Tabel 3.23 Skenario *Use Case* Penyajian Tentang Malaria

Identifikasi	
No	3
Nama	Penyajian Tentang Malaria
Tujuan	Untuk mengatahui tentag malaria
Actor	Pemain
Skenario Utama	
Kondisi Awal	Menu Utama ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu penyajian tentang malaria	
	2. Menampilkan tentang malaria

Kondisi Akhir	Menampilkan menu tentang malaria

4. Skenario *Use case* Perhitungan Nyawa

Skenario *use case* ini akan menjelaskan interaksi antara *actor* yaitu pemain, dengan *use case* perhitungan nyawa yang akan dijelaskan pada Tabel 3.24 di bawah ini.

Tabel 3.24 Skenario *Use case* Perhitungan Nyawa

Identifikasi	
No	4
Nama	Perhitungan nyawa
Tujuan	Untuk dapat menghitung nyawa karakter
Actor	Pemain
Skenario Utama	
Kondisi Awal	Permainan dimulai
Aksi Aktor	**Reaksi Sistem**
	1. Menentukan darah ratu sebesar 200
	2. Menentukan darah karakter sebesar 100
3. Menembak	
	4. Apabila mengenai nyamuk, nyamuk mati dan menghilang
	5. Apabila mengenai ratu, darah ratu -10. Jika darah ratu setelah terkena ≤ 0 maka memainkan animasi ratu mati
6. Menyentuh nyamuk	
	7. Darah pemain berkurang 3
8. Menyentuh ratu	
	9. Darah pemain berkurang 5
Kondisi Akhir	Hasil perhitungan nyawa

5. Skenario *Use case* Perhitungan *Score*

Skenario *use case* ini menjelaskan interaksi antara *actor* yaitu pemain, dengan *use case* perhitungan *score* yang akan dijelaskan pada Tabel 3.25 di bawah ini.

Tabel 3.25 Skenario *Use Case* Perhitungan *Score*

Identifikasi	
No	5
Nama	Perhitungan *Score*

Tujuan	Untuk dapat menghitung *score* yang diraih pemain	
Actor	Pemain	
Skenario Utama		
Kondisi Awal	Permainan dimulai	
Aksi Aktor		**Reaksi Sistem**
		1. Menentukan skor awal sebesar 0
2. Membunuh nyamuk		
		3. Menambah skor sebesar 10 pada skor permainan
4. Menyelesaikan *game* materi		
		5. Menambah skor sebesar 100 pada bonus
6. Membunuh ratu		
		7. Menambah skor sebesar 150 pada pada bonus
		8. Pada setiap akhir level pemainan menjumlahkan semua skor yang didapat dari skor permainan dan bonus
Kondisi Akhir	Hasil perhitungan skor	

6. Skenario *Use case* Pergerakan Karakter

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* pergerakan karakter yang akan dijelaskan pada Tabel 3.26 di bawah ini.

Tabel 3.26 Skenario *Use Case* Pergerakan Karakter

Identifikasi		
No	6	
Nama	Pergerakan Karakter	
Tujuan	Untuk melakukan pergerakan karakter	
Actor	Pemain	
Skenario Utama		
Kondisi Awal	Karakter diam	
Aksi Aktor		**Reaksi Sistem**
1. Jika pemain menekan tombol kanan pada *keyboard*		
		2. Karakter bergerak ke kanan
3. Jika pemain menekan tombol kiri pada *keyboard*		
		4. Karakter bergerak ke kiri
5. Jika pemain menekan tombol atas pada *keyboard*		
		6. Karakter loncat
7. Jika pemain menekan tombol		

bawah pada *keyboard*	
	8. Karakter masuk pintu hutan
9. Jika pemain menekan tombol sebelah kiri pada *mouse*	
	10. Karakter menyemprotkan obat nyamuk
Kondisi Akhir	Pemain melakukan pergerakan karakter

7. Skenario *Use case* Petunjuk Kontrol

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* petunjuk kontrol yang dijelaskan pada Tabel 3.27 di bawah ini.

Tabel 3.27 Skenario *Use Case* Petunjuk Kontrol

Identifikasi	
No	7
Nama	Petunjuk Kontrol
Tujuan	Untuk mengatahui pentunjuk kontrol karakter
Actor	Pemain
Skenario Utama	
Kondisi Awal	Menu memlih petunjuk
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu petunjuk kontrol	
	2. Menampilkan petunjuk kontrol
Kondisi Akhir	Menampilkan menu petunjuk kontrol

8. Skenario *Use case* Petunjuk *Level*

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* petunjuk level yang dijelaskan pada Tabel 3.28 di bawah ini.

Tabel 3.28 Skenario *Use Case* Petunjuk *Level*

Identifikasi	
No	8
Nama	Petunjuk *Level*
Tujuan	Untuk mengatahui pentunjuk *level* ditiap permainan
Actor	Pemain
Skenario Utama	
Kondisi Awal	Menu memilih petunjuk
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu petunjuk l*evel*	
	2. Menampilkan petunjuk *level*

Kondisi Akhir	Menampilkan menu petunjuk *level*

9. Skenario *Use case* Memasang Kelambu

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* memasang kelambu yang dijelaskan pada Tabel 3.29 di bawah ini.

Tabel 3.29 Skenario *Use Case* Memasang Kelambu

Identifikasi	
No	9
Nama	Memasang Kelambu
Tujuan	Untuk mengetahui memasang kelambu
Actor	Pemain
Skenario Utama	
Kondisi Awal	Menyentuh item kelambu
Aksi Aktor	**Reaksi Sistem**
1. Menyentuh item kelambu	
	2. Menampilkan kelambu dan tempat tidur
3. Meletakan kelambu di tempat tidur	
	4. Kelambu terpasang di tempat tidur
	5. Jika tidak sesuai ukuran dan warna kelambu dengan tempat tidur, maka kelambu kembali ke posisi awal
	6. Menampilkan tombol lanjut
7. Menekan tombol lanjut	
	8. Menutup penyajian materi memasang kelambu
	9. Membuka pintu hutan yang terkunci
Kondisi Akhir	Menampilkan *game* materi memasang kelambu

10. Skenario *Use case* Menguras Bak Mandi

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* menguras bak mandi yang dijelaskan pada Tabel 3.30 di bawah ini.

Tabel 3.30 Skenario *Use Case* Menguras Bak Mandi

Identifikasi	
No	10
Nama	Menguras Bak Mandi
Tujuan	Untuk mengetahui menguras bak mandi
Actor	Pemain
Skenario Utama	

Kondisi Awal	Menyentuh item bak mandi
Aksi Aktor	**Reaksi Sistem**
1. Menyentuh item bak mandi	
	2. Menampilkan bakmandi yang kotor
3. Menekan tutup bak mandi	
	4. Tutup terbuka dan air keluar
	5. Menampilkan notifikasi air telah keluar
6. Menekan notifikasi	
	7. Menampilkan sikat
8. Menyikat kotoran pada bak mandi	
	9. Menghilangkan kotoran yang disentuh sikat
	10. Menampilkan pengisian air pada bak mandi
11. Menekan keran air	
	12. Menampilan bak mandi telah terisi air dan bersih
	13. Menampilkan tombol lanjut
14. Menekan tombol lanjut	
	15. Menutup *game* materi menguras
	16. Membuka pintu hutan yang terkunci
Kondisi Akhir	Menampilkan *game* materi menguras bak mandi

11. Skenario *Use case* Menutup Tempat Air

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* menutup tempat air yang dijelaskan pada Tabel 3.31 di bawah ini.

Tabel 3.31 Skenario *Use Case* Menutup Tempat Air

Identifikasi	
No	11
Nama	Menutup Tempat Air
Tujuan	Untuk mengetahui menutup tempat air
Actor	Pemain
Skenario Utama	
Kondisi Awal	Menyentuh item tempat air
Aksi Aktor	**Reaksi Sistem**
1. Menyentuh item tempat air	
	2. Menampilkan tutup dan tempat air
3. Meletakan tutup	
	4. tutup terpasang di tempat air
	5. Jika tidak sesuai ukuran dan warna tutup dengan tempat air, maka tutup kembali ke

	posisi awal
	6. Menampilkan tombol lanjut
7. Menekan tombol lanjut	
	8. Menutup *game* materi menutup tempat air
	9. Membuka pintu hutan yang terkunci
Kondisi Akhir	Menampilkan penyajian materi menutup tempat air

12. Skenario *Use case* Mengubur Wadah

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* mengubur wadah yang dijelaskan pada Tabel 3.32 di bawah ini.

Tabel 3.32 Skenario *Use Case* Mengubur Wadah

Identifikasi	
No	12
Nama	Mengubur Wadah
Tujuan	Untuk mengetahui mengubur wadah kosong
Actor	Pemain
Skenario Utama	
Kondisi Awal	Menyentuh item lubang
Aksi Aktor	**Reaksi Sistem**
1. Menyentuh item lubang	
	2. Menampilkan lubang dan wadah
3. Meletakan wadah di lubang	
	4. Wadah masuk di lubang
	5. Jika wadah tidak sesuai posisi lubang, maka wadah kembali ke posisi awal
	6. Menampilkan *scope*
7. Menekan *scope*	
	8. Mengubur lubang
	9. Menampilkan tombol lanjut
10. Menekan tombol lanjut	
	11. Menutup *game* materi mengubur wadah
	12. Membuka pintu hutan yang terkunci
Kondisi Akhir	Menampilkan *game* materi mengubur wadah

13. Skenario *Use case* Melawan Ratu

Skenario *use case* ini menjelaskan interaksi antara *actor* dengan *use case* melawan ratu yang dijelaskan pada Tabel 3.33 di bawah ini.

Tabel 3.33 Skenario *Use Case* Melawan Ratu

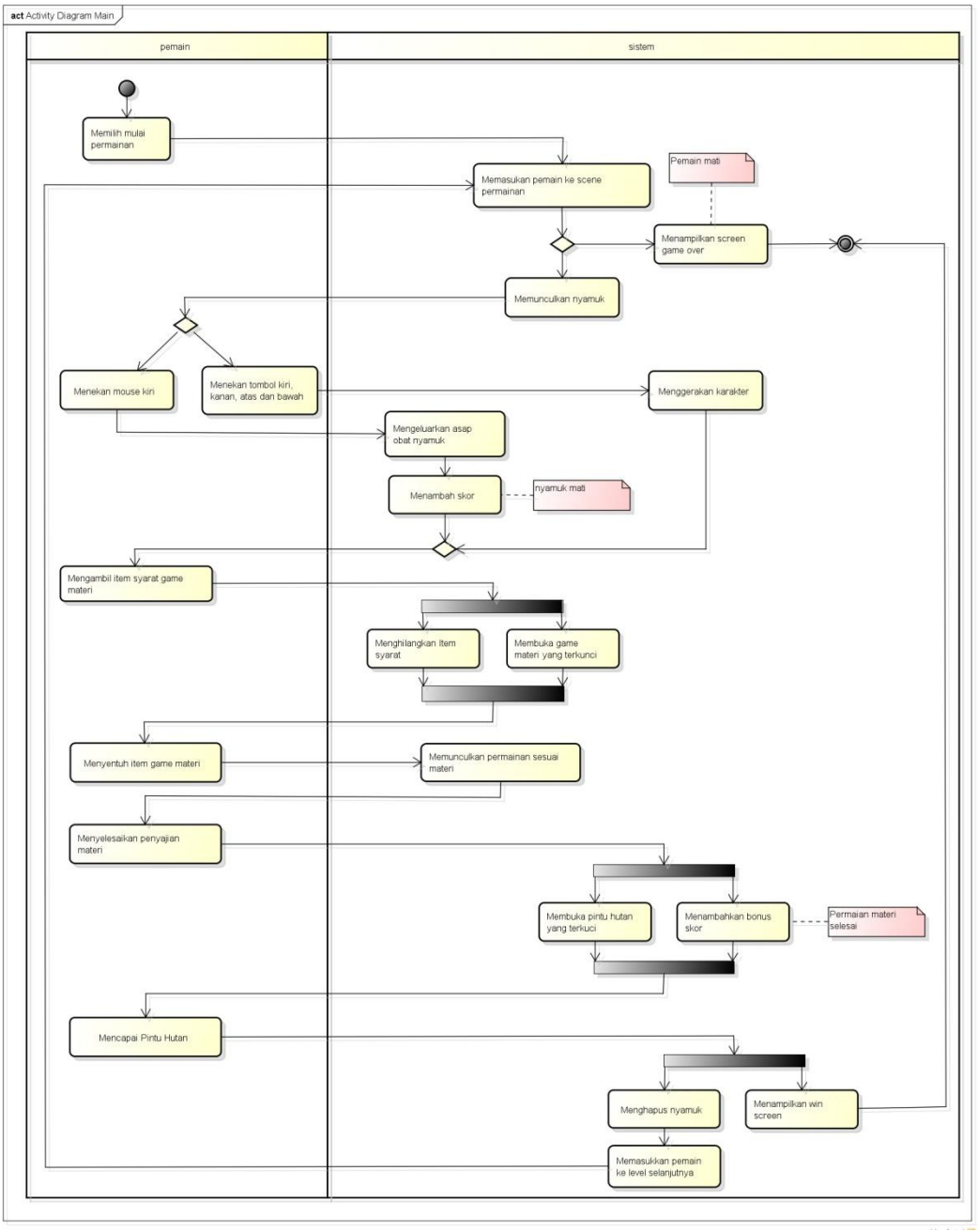
Identifikasi	
No	13
Nama	Melawan Ratu
Tujuan	Untuk dapat melawan ratu
Actor	Pemain
Skenario Utama	
Kondisi Awal	Memunculkan melawan ratu
Aksi Aktor	**Reaksi Sistem**
1. Menuju ke rumah ratu	
	2. Memunculkan ratu
	3. Menggerakan ratu
	4. Apabila pemain mati, maka menampilkan menu *game over*
5. Jika Menekan tombol Kiri, kanan, Atas dan bawah	
	6. Menggerakkan karakter
7. Jika Menekan klik kiri	
	8. Mengeluarkan DDT
	9. Darah ratu berkurang
	10. Jika darah ratu habis, maka ratu mati
	11. Menampilkan *winscreen*
Kondisi Akhir	Menampilkan *game* melawan ratu

3.1.8.5 *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan aliran kerja (*work flow*) atau aktivitas dari sebuah sistem atau proses bisnis.

1. *Activity Diagram* Main

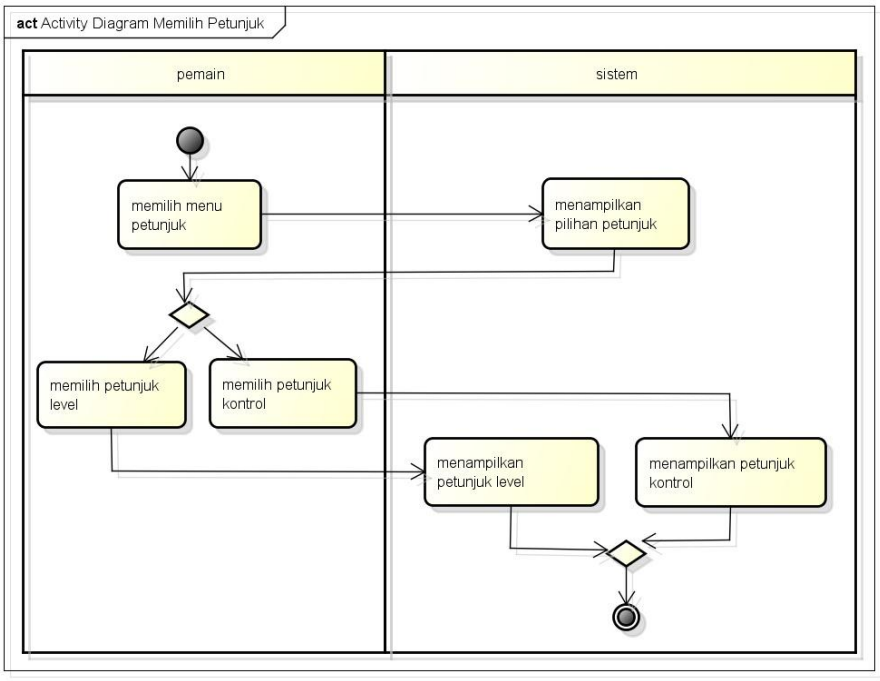
Gambar 3.9 memperlihatkan *activity diagram* main pada aplikasi *game Save Family From Malaria.*



Gambar 3.9 *Activity Diagram* Main

2. *Activity Diagram* Memilih Petunjuk

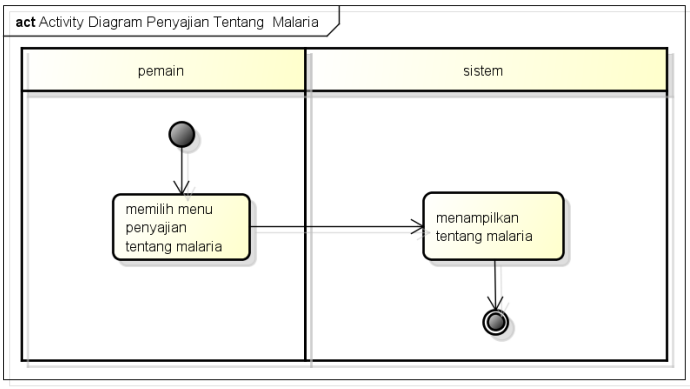
Gambar 3.10 memperlihatkan *activity diagram memilih* petunjuk pada aplikasi *game Save Family From malaria.*



Gambar 3.10 *Activity Diagram* Memilih Petunjuk

3. *Activity Diagram* Penyajian Tentang Malaria

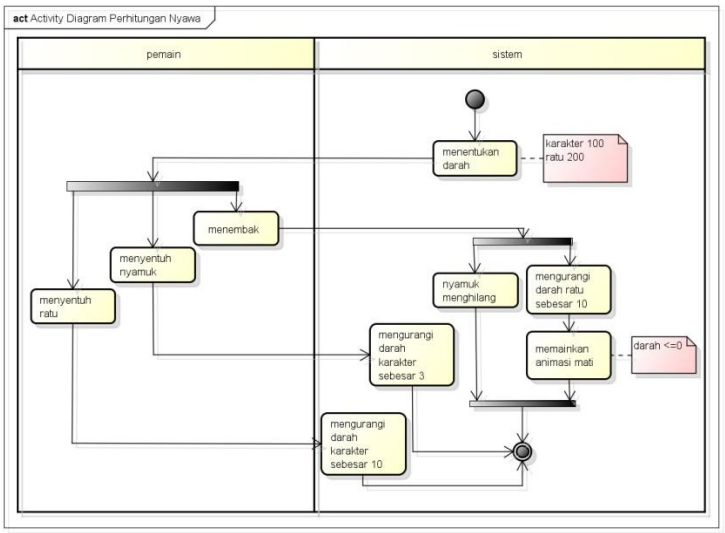
Gambar 3.11 memperlihatkan *diagram activity* penyajian tentang malaria pada aplikasi *game Save Family From Malaria.*



Gambar 3.11 *Activity Diagram* Penyajian Tentang Malaria

4. *Activity Diagram* Perhitungan Nyawa

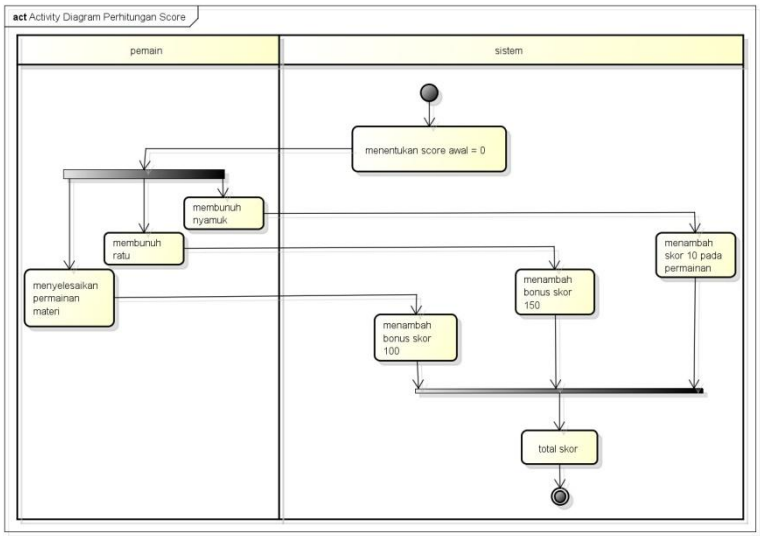
Gambar 3.12 memperlihatkan *diagram activity* perhitungan nyawa pada aplikasi *game Save Family From Malaria.*



Gambar 3.12 *Activity Diagram* Perhitungan Nyawa

5. *Activity Diagram* Perhitungan *Score*

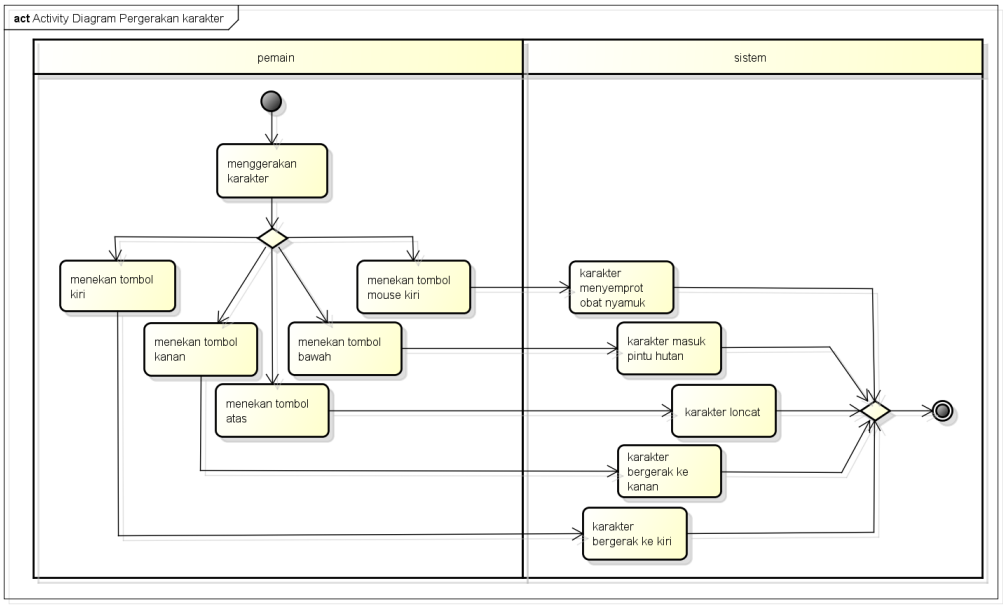
Gambar 3.13 memperlihatkan *diagram activity* perhitungan *score* pada aplikasi *game Save Family From Malaria.*



Gambar 3.13 *Activity Diagram* Perhitungan *Score*

6. *Activity Diagram* Pergerakan Karakter

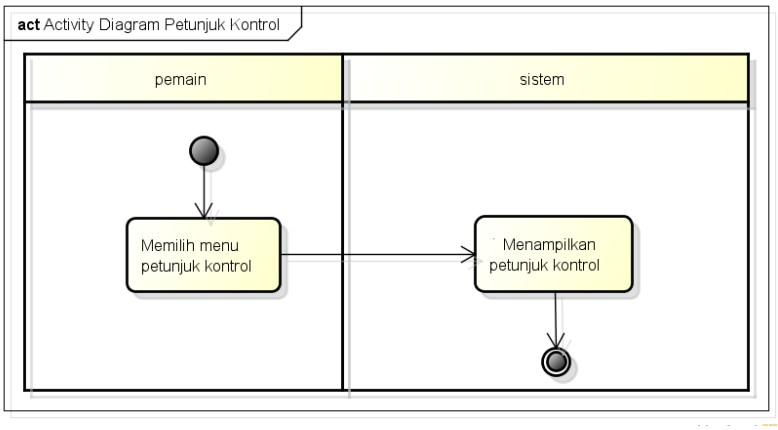
Gambar 3.14 memperlihatkan *activity diagram* pergerakan karakter pada aplikasi *game Save Family From Malaria.*



Gambar 3.14 *Activity Diagram* Pergerakan Karakter

7. *Activity Diagram* Petunjuk Kontrol

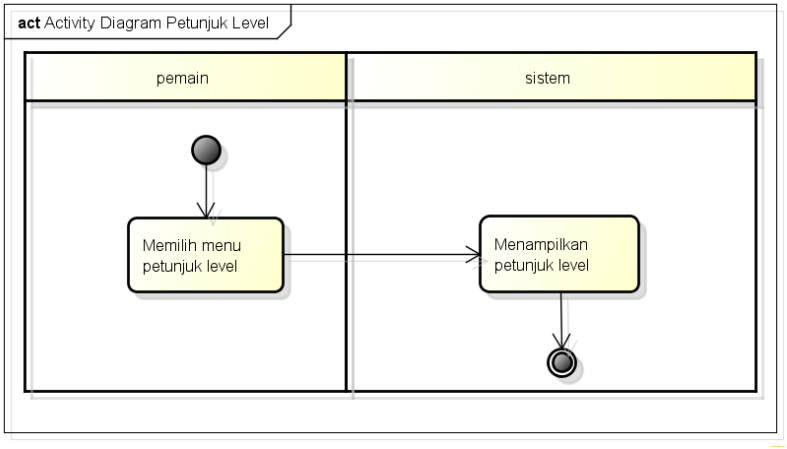
Gambar 3.15 memperlihatkan *activity diagram* petunjuk kontrol pada aplikasi *game Save Family From Malaria.*



Gambar 3.15 *Activity Diagram* Petunjuk Kontrol

8. *Activity Diagram* Petunjuk *Level*

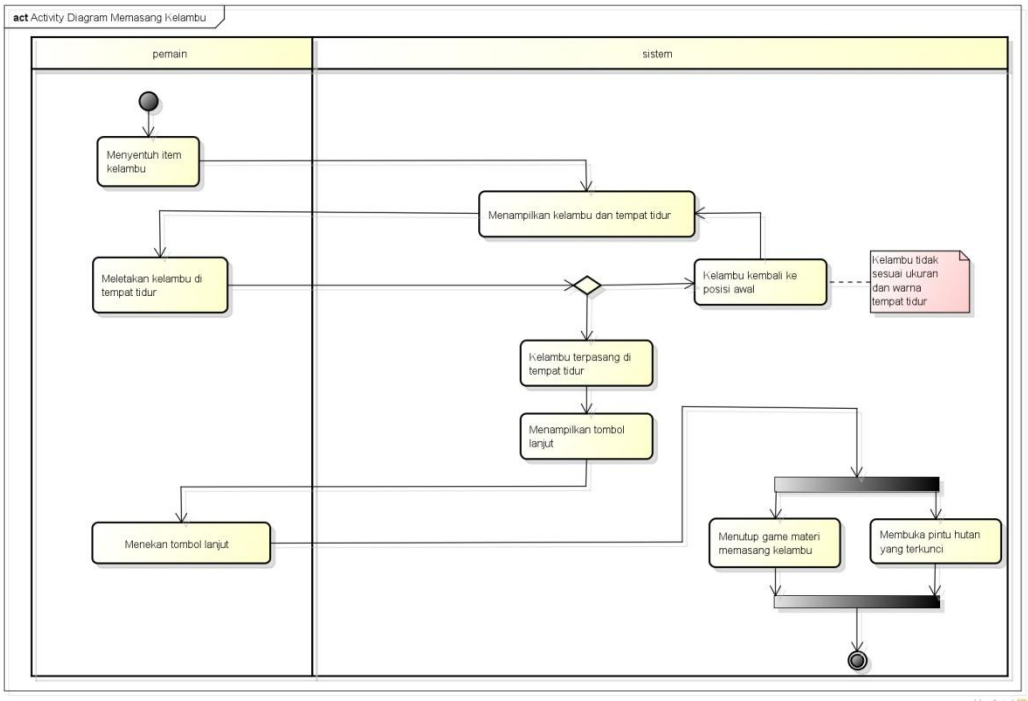
Gambar 3.16 memperlihatkan *activity diagram* petunjuk *level* pada aplikasi *game Save Family From Malaria.*



Gambar 3.16 *Activity Diagram* Petunjuk *Level*

9. *Activity Diagram* Memasang Kelambu

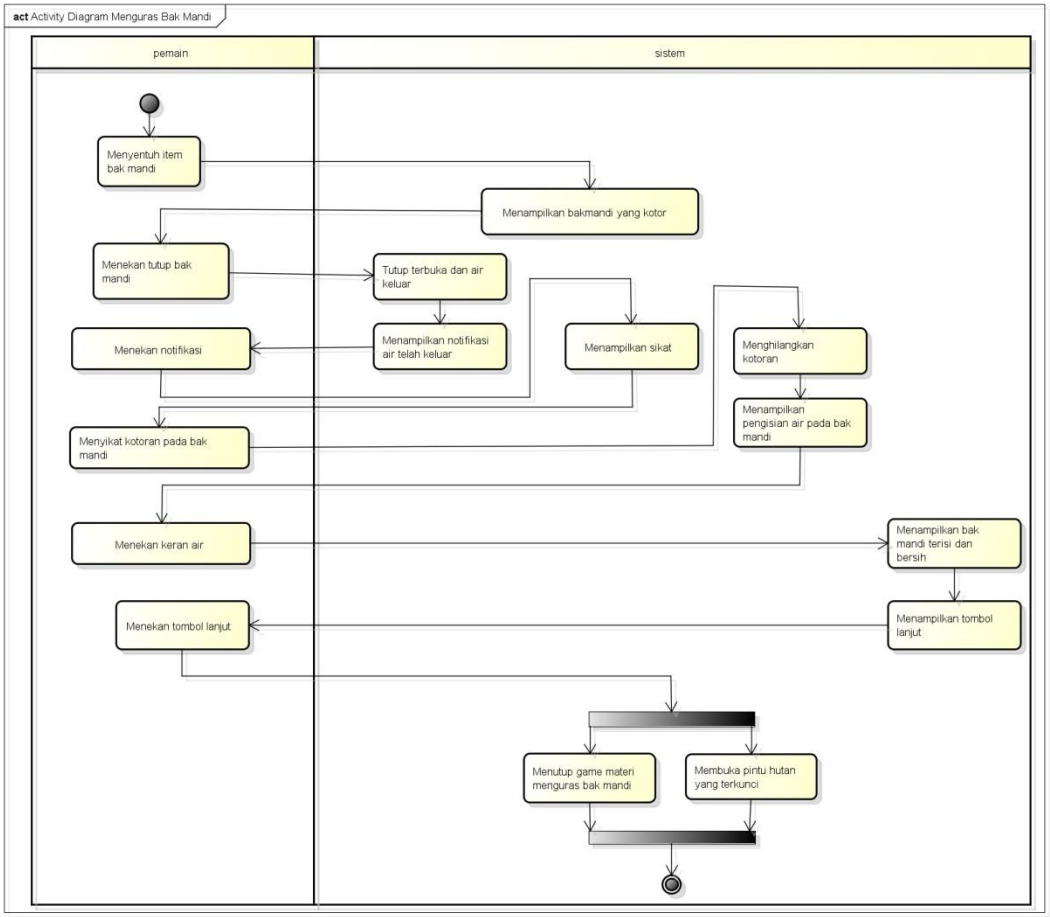
Gambar 3.17 memperlihatkan *activity diagram* memasang kelambu pada aplikasi *game Save Family From Malaria.*



Gambar 3.17 *Activity Diagram* Memasang Kelambu

10. *Activity Diagram* Menguras Bak Mandi

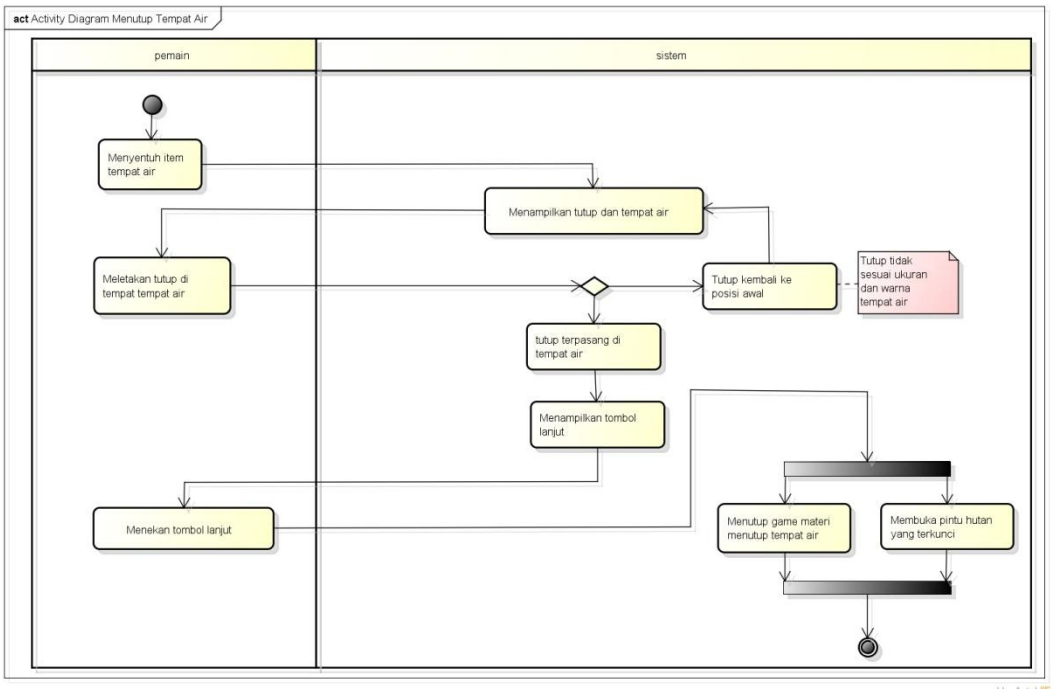
Gambar 3.18 memperlihatkan *activity diagram* meguras bak mandi pada aplikasi *game Save Family From Malaria.*



Gambar 3.18 *Activity Diagram* Menguras Bak Mandi

11. *Activity Diagram* Menutup Tempat Air

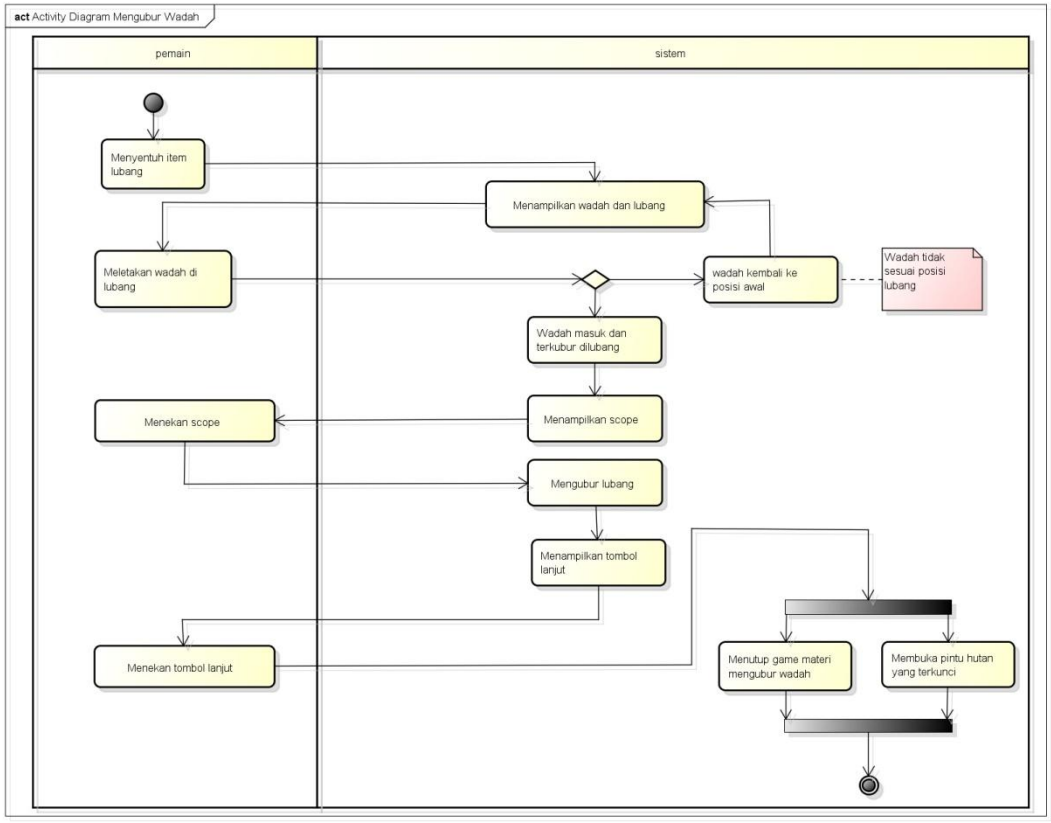
Gambar 3.19 memperlihatkan *activity diagram* menutup tempat air pada aplikasi *game Save Family From Malaria.*



Gambar 3.19 *Activity Diagram* Menutup Tempat Air

12. *Activity Diagram* Mengubur Wadah

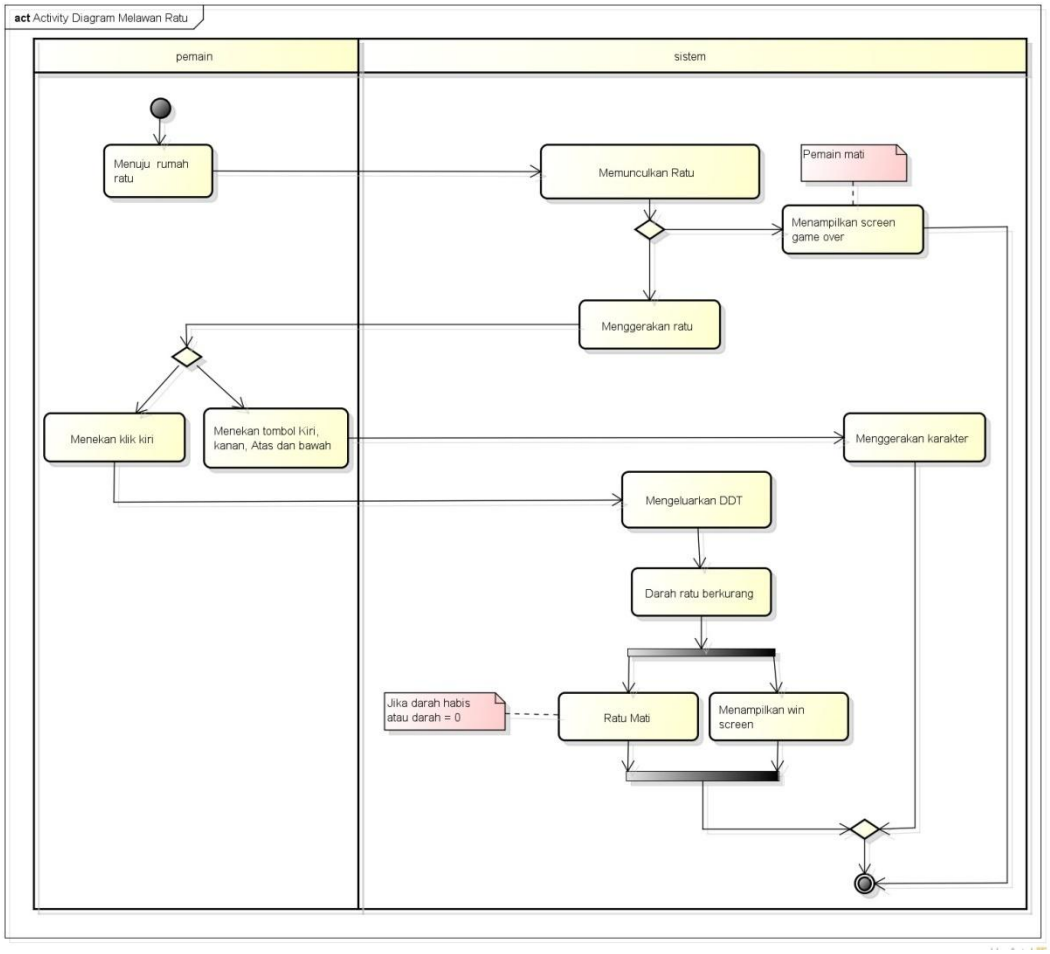
Gambar 3.20 memperlihatkan *activity diagram* mengubur wadah pada aplikasi *game Save Family From Malaria.*



Gambar 3.20 *Activity Diagram* Mengubur Wadah

13. *Activity Diagram* Melawan Ratu

Gambar 3.21 memperlihatkan *activity diagram* melawan ratu pada aplikasi *game Save Family From Malaria*.



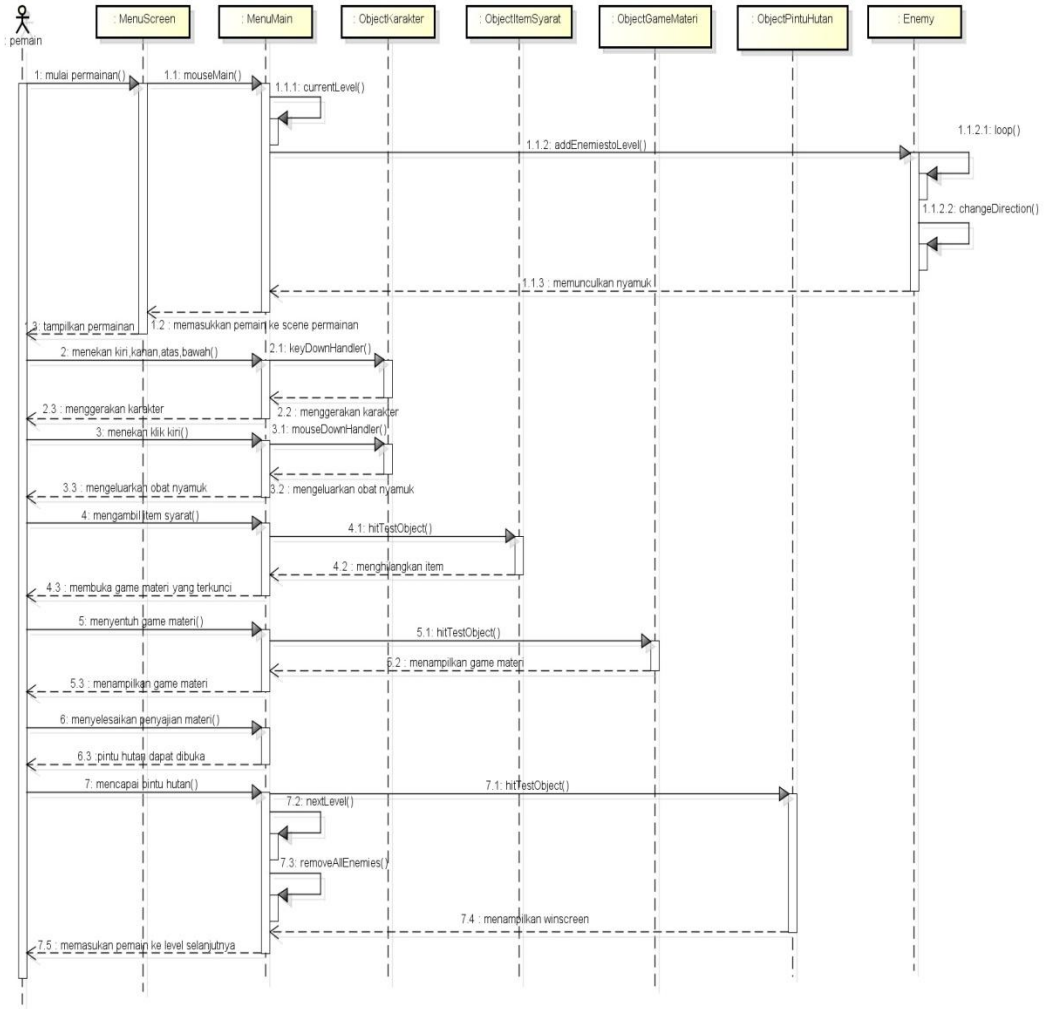
Gambar 3.21 *Activity Diagram* Melawan Ratu

3.1.8.6 *Sequence Diagram*

Sequence diagram menggambarkan perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek [16]. *Sequence diagram* pada *game Save Family From Malaria* adalah sebagai berikut:

1. *Sequence Diagram* Main

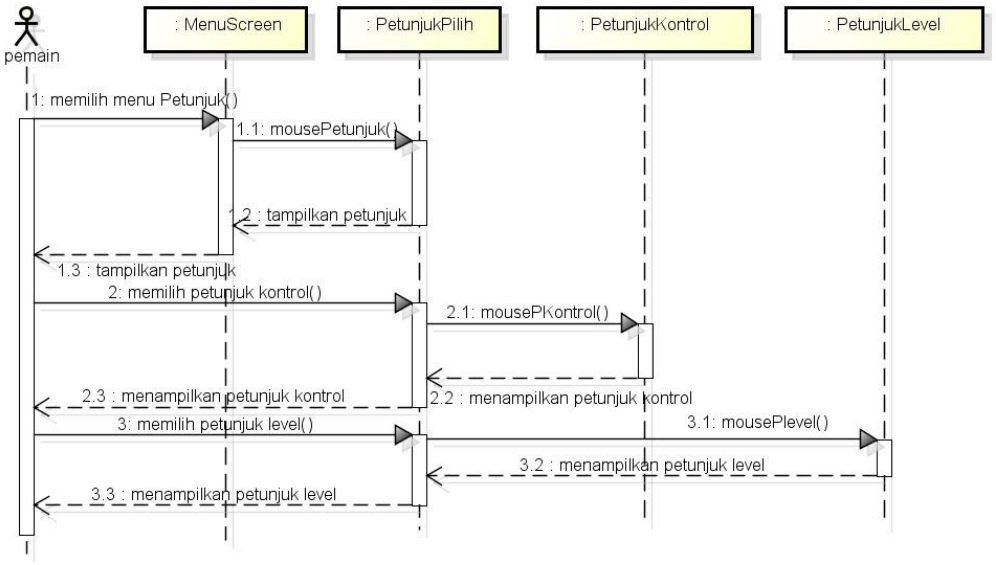
   Sequence diagram main pada aplikasi ini dapat dilihat pada Gambar 3.22.



Gambar 3.22 *Sequence Diagram* Main

2. *Sequence Diagram* Memilih Petunjuk

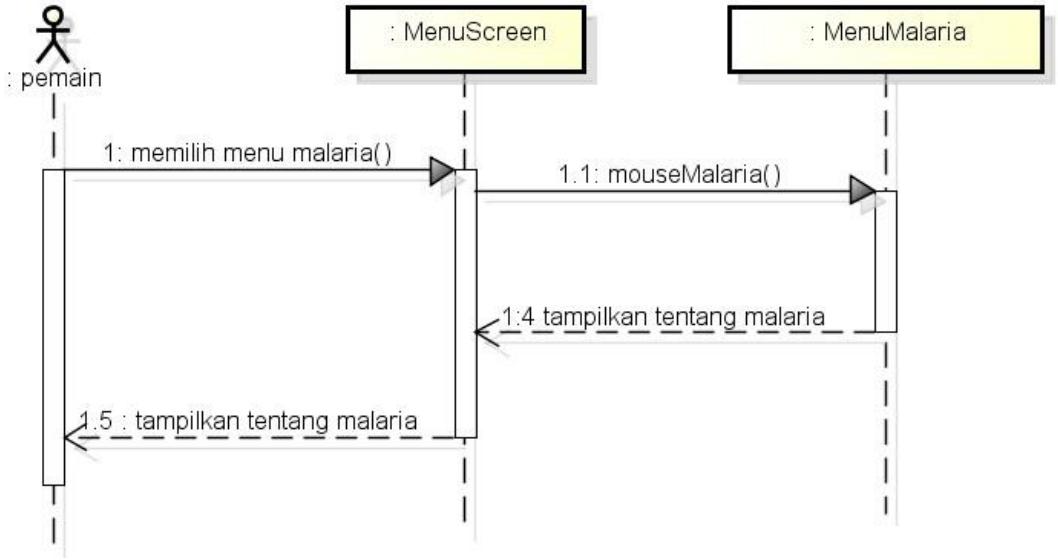
Sequence diagram memilih petunjuk pada aplikasi ini dapat dilihat pada Gambar 3.23.



Gambar 3.23 *Sequence Diagram* Memilih Petunjuk

3. *Sequence Diagram* Penyajian Tentang Malaria

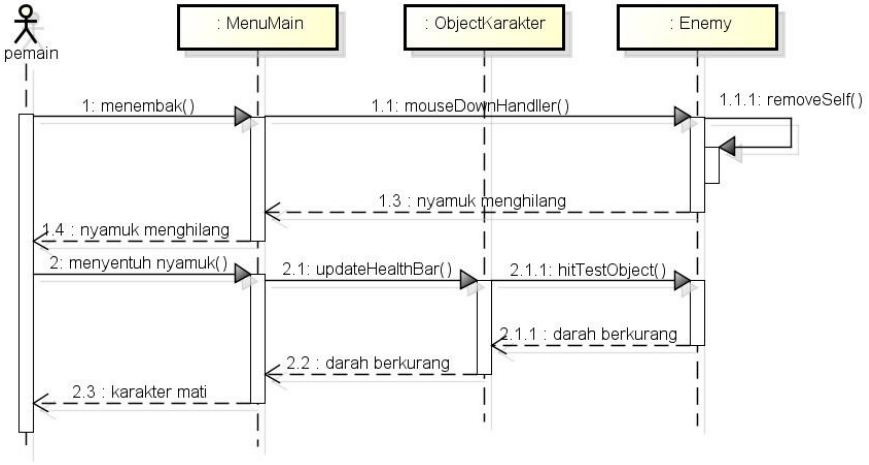
Sequence diagram penyajian tentang malaria pada aplikasi ini dapat dilihat pada Gambar 3.24.



Gambar 3.24 *Sequence Diagram* Penyajian Tentang Malaria

4. *Sequence Diagram* Perhitungan Nyawa

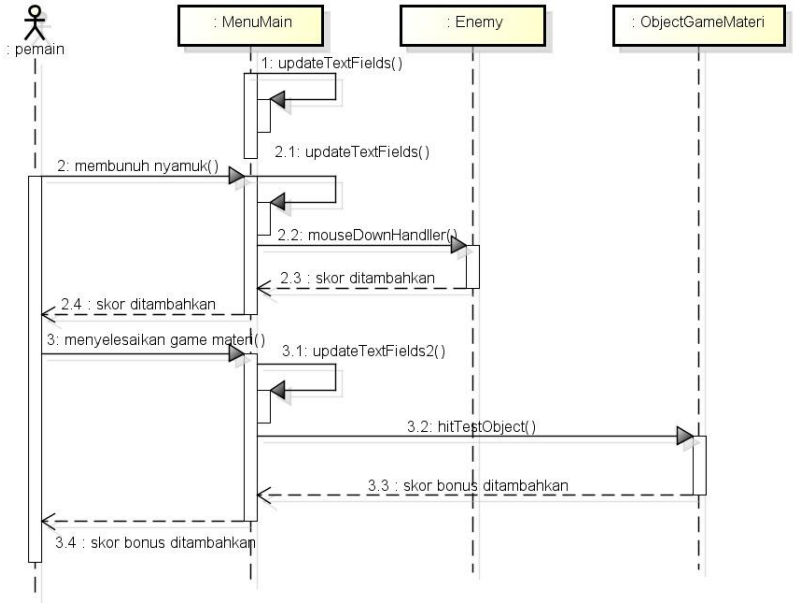
Sequence diagram pergerakan karakter pada aplikasi ini dapat dilihat pada Gambar 3.25.



Gambar 3.25 *Sequence Diagram* Perhitungan Nyawa

5. *Sequence Diagram* Perhitungan *Score*

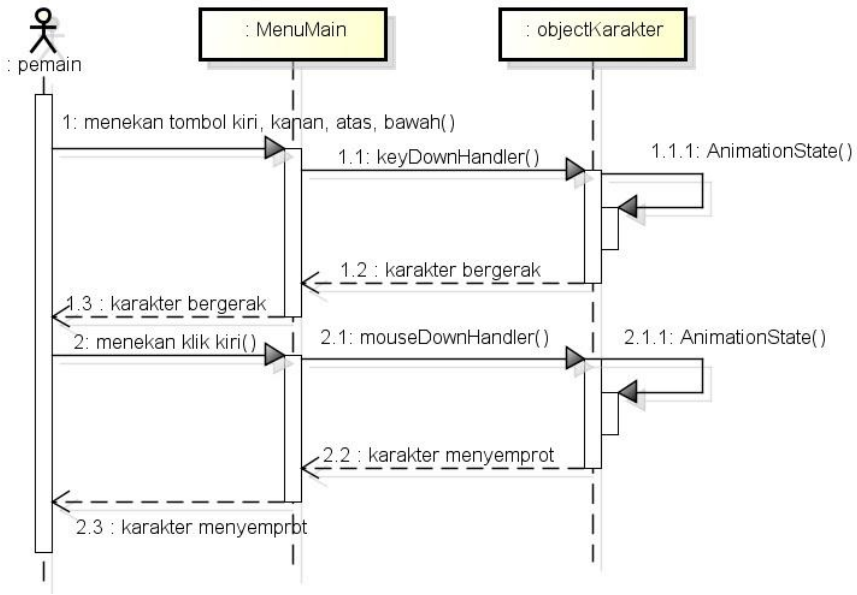
Sequence diagram pergerakan karakter pada aplikasi ini dapat dilihat pada Gambar 3.26.



Gambar 3.26 *Sequence Diagram* Perhitungan *Score*

6. *Sequence Diagram* Pergerakan Karakter

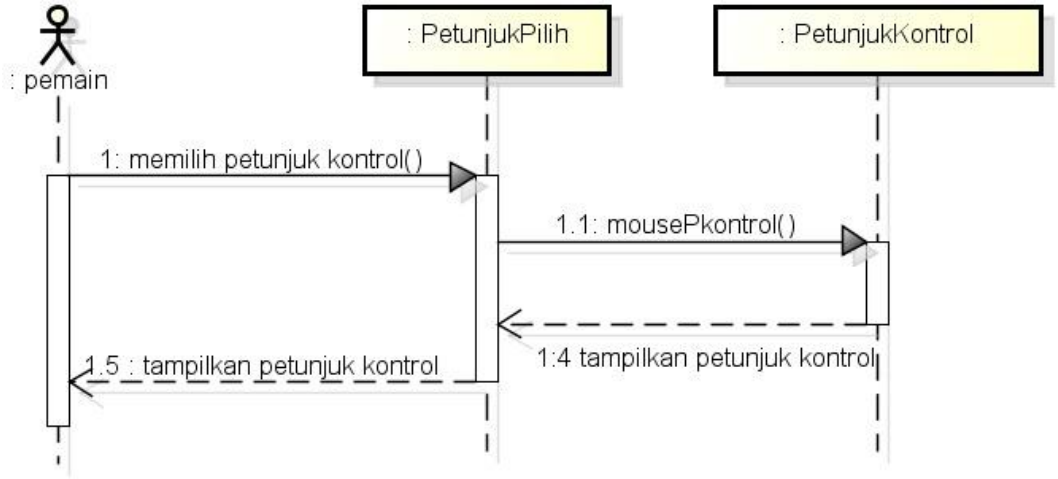
Sequence diagram pergerakan karakter pada aplikasi ini dapat dilihat pada Gambar 3.27.



Gambar 3.27 *Sequence Diagram* Pergerakan karakter

7. *Sequence Diagram* Petunjuk Kontrol

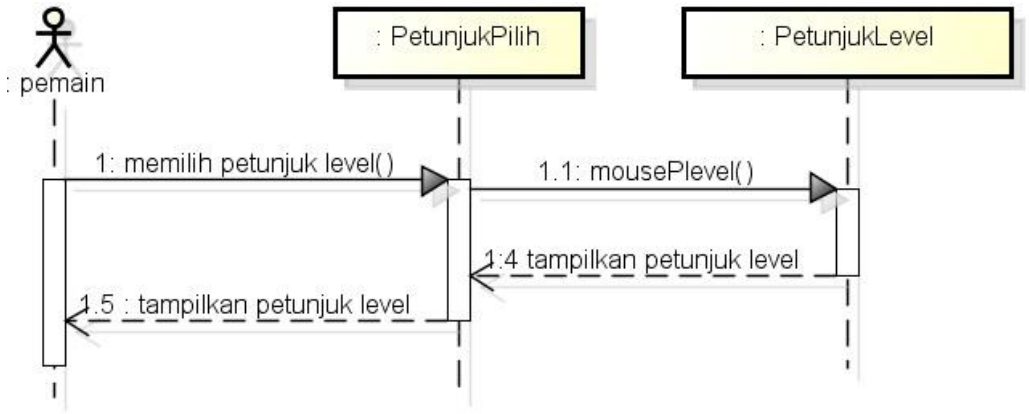
Sequence diagram petunjuk kontrol pada aplikasi ini dapat dilihat pada Gambar 3.28.



Gambar 3.28 *Sequence Diagram* Petunjuk Kontrol

8. *Sequence Diagram* Petunjuk *Level*

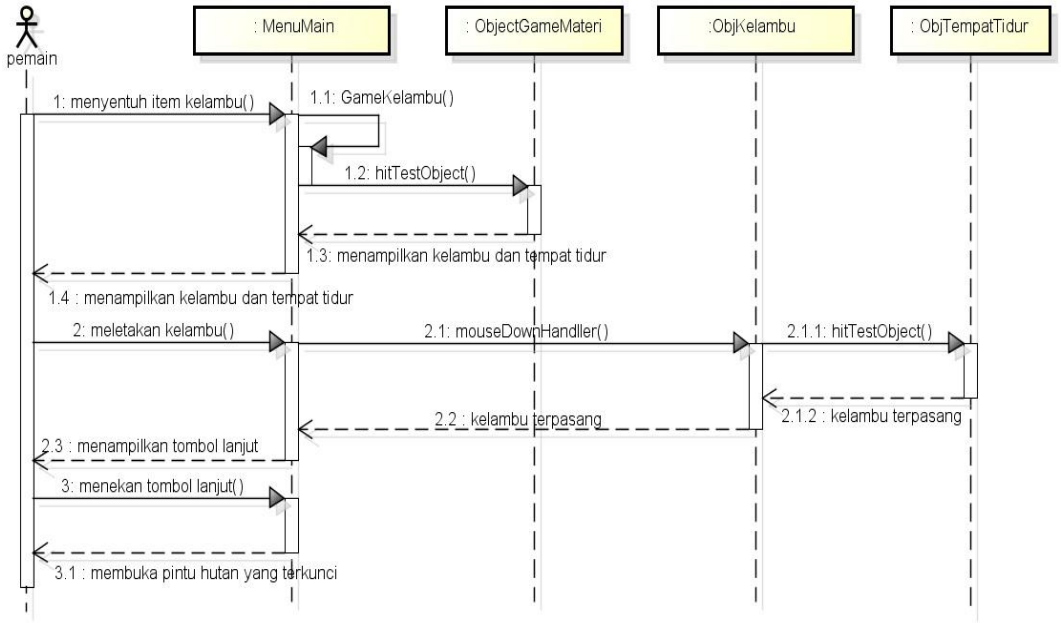
Sequence diagram petunjuk *level* pada aplikasi ini dapat dilihat pada Gambar 3.29.



Gambar 3.29 *Sequence Diagram* Petunjuk *Level*

9. *Sequence Diagram* Memasang Kelambu

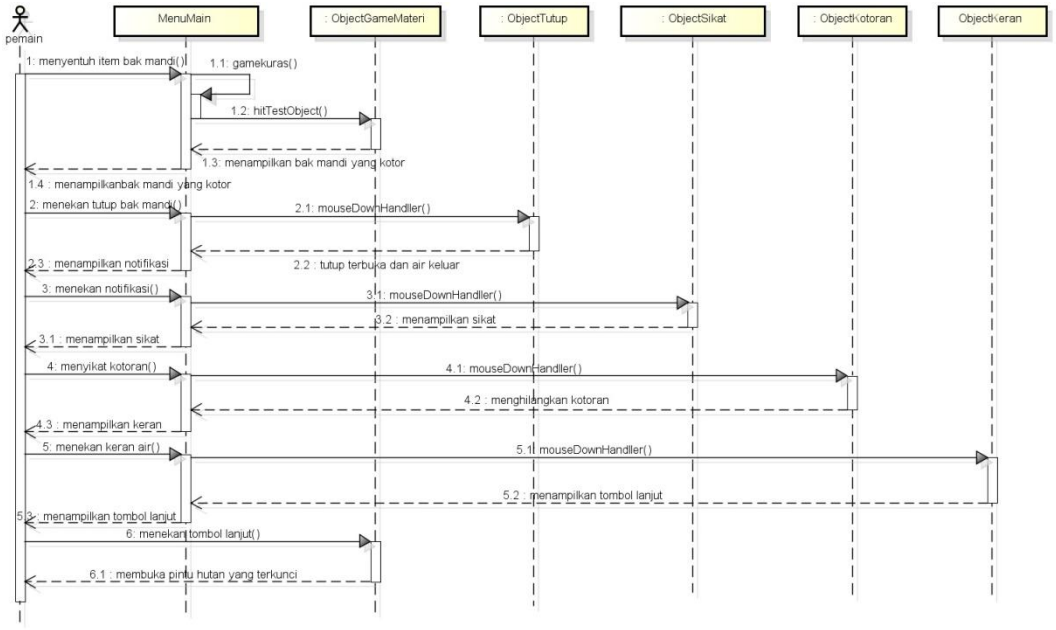
Sequence diagram memasang kelambu pada aplikasi ini dapat dilihat pada Gambar 3.30.



Gambar 3.30 *Sequence Diagram* Memasang Kelambu

10. *Sequence Diagram* Menguras Bak Mandi

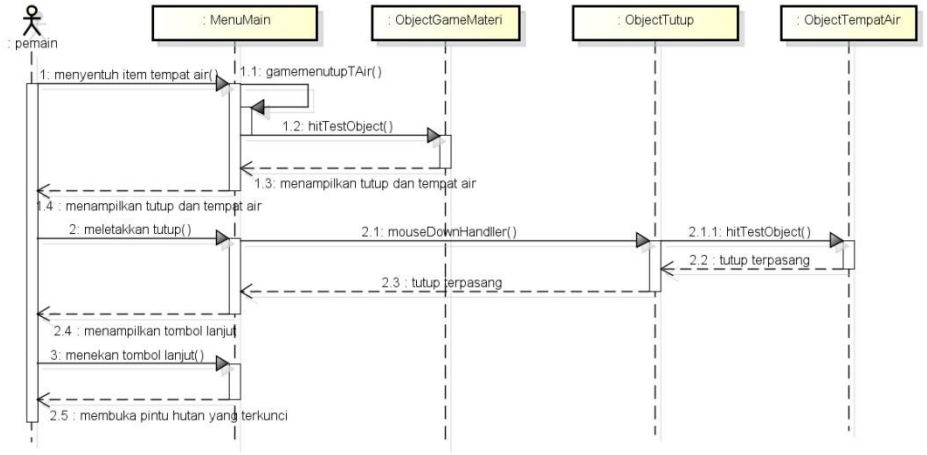
Sequence diagram menguras bak mandi pada aplikasi ini dapat dilihat pada Gambar 3.31.



Gambar 3.31 *Sequence Diagram* Menguras Bak Mandi

11. *Sequence Diagram* Menutup Tempat Air

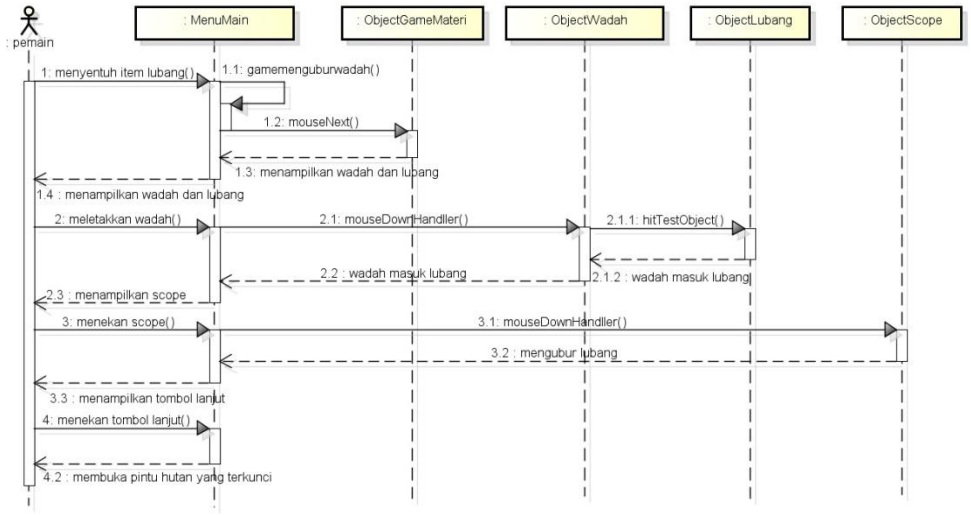
Sequence diagram menutup tempat air pada aplikasi ini dapat dilihat pada Gambar 3.32.



Gambar 3.32 *Sequence Diagram* Menutup Tempat Air

12. *Sequence Diagram* Mengubur Wadah

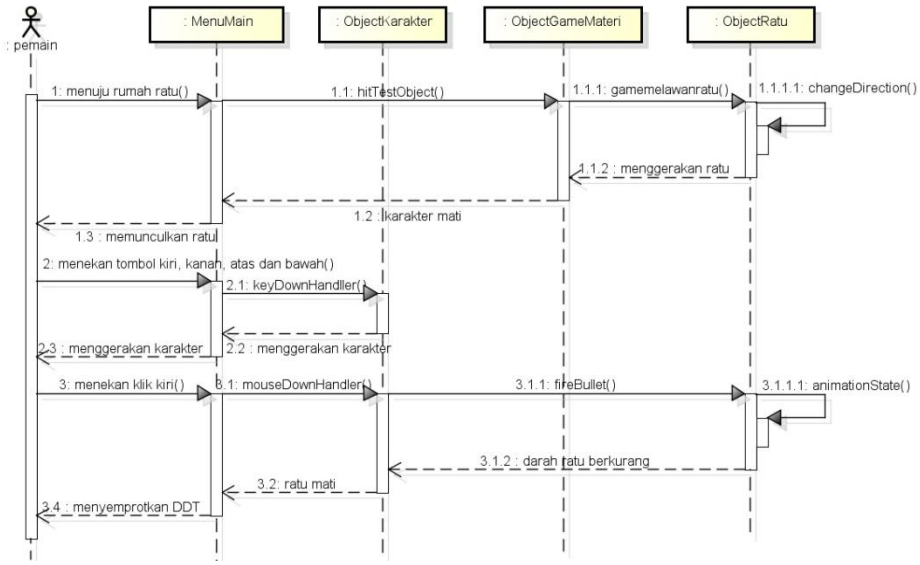
Sequence diagram mengubur wadah pada aplikasi ini dapat dilihat pada Gambar 3.33.



Gambar 3.33 *Sequence Diagram* Mengubur Wadah

13. *Sequence Diagram* Melawan Ratu

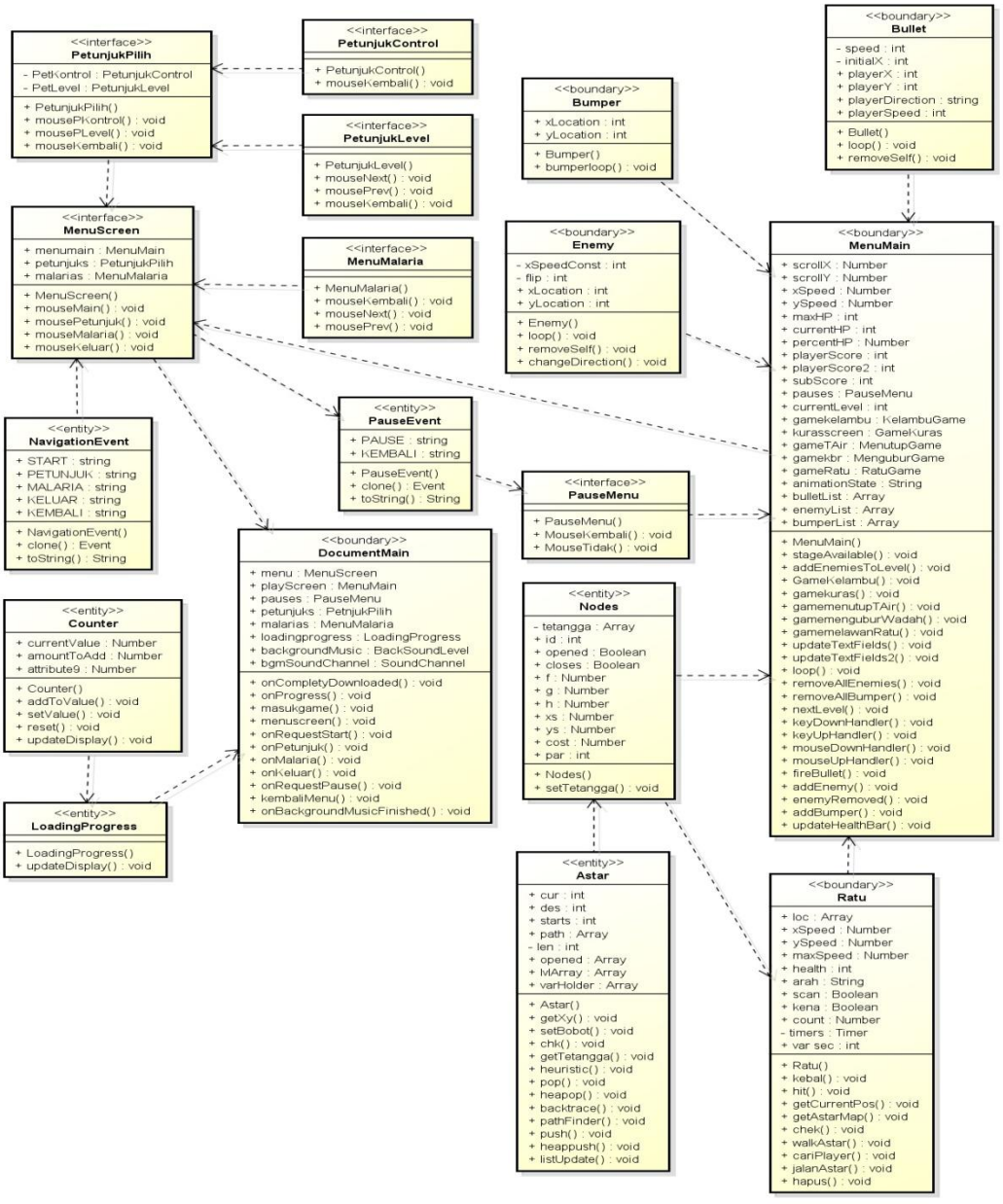
Sequence diagram melawan ratu pada aplikasi ini dapat dilihat pada Gambar 3.34.



Gambar 3.34 *Sequence Diagram* Melawan Ratu

3.1.8.7 *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem. Perancangan struktur sistem yang terdapat pada *game save family from malaria* dapat dilihat pada Gambar 3.35.



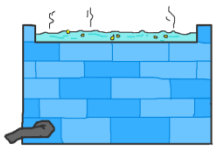
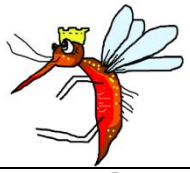
Gambar 3.35 *Class Diagram Game Save Family From Malaria*

3.2 Perancangan

3.2.1 Konten-Konten *Game*

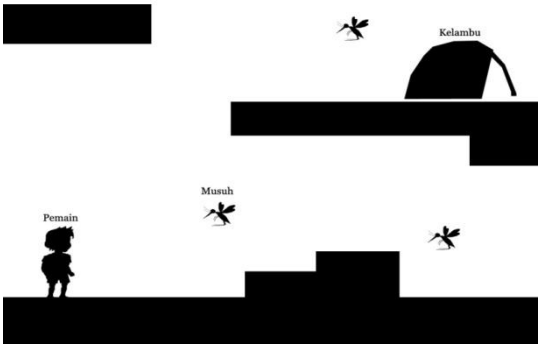
Konten-konten yang akan digunakan dalam pembangunan aplikasi *game Save Family From Malaria* dapat dilihat pada Tabel 3.34.

Tabel 3.34 Konten-Konten Pada *Game Save Family From Malaria*

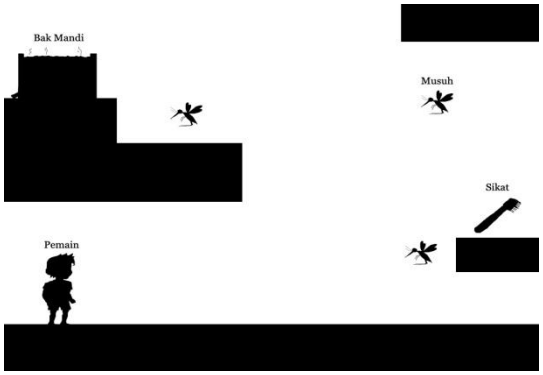
No	Nama	Gambar	Keterangan
1	Bambang		Bambang merupakan tokoh utama dan karakter yang dapat dikontrol oleh pemain dengan tombol kanan, kiri, atas pada *keyboard* untuk berjalan dan klik kiri pada *mouse* untuk menembak. Bambang mempunyai 50 *health* dan akan berkurang apabila musuh menyerangnya.
2	Nyamuk		Nyamuk merupakan karakter jahat yang akan menjadi hambatan pemain dalam menyelesaikan permainannya dan Bambang akan mati apabila terkena serangan sebanyak 5 kali.
3	Sikat Pembersih		Konten ini harus diambil pemain, sebagai syarat untuk menguras bak mandi. Jika tidak mendapatkan konten ini maka tidak dapat menguras bak mandi. Terdapat di level 2 saja.
4	Bak Mandi		Konten ini harus dikuras oleh pemain agar bisa melanjutkan ke level 3. pemain menguras bak mandi dengan syarat mendapatkan sikat pembersih terlebih dahulu. Menguras menekan tombol spasi dan terapat pada *level* 2 saja.
5	Ratu		Ratu berada pada akhir level 5 dan akan melakukan serangan terhadap pemain. Apabila serangan bos mengenai pemain, maka darah pemain akan dikurangi sebesar 20. Kecepatan dari ratu dalam menyerang sangat cepat.
6	Tutup		Konten ini berfungsi untuk menutup tempat air. Pemain harus mengumpulkan tutup ini sebagai syarat menutup tempat air. Terdapat pada level 3 saja.
7	Tempat Air		Konten ini harus ditutup oleh pemain menggunakan tutup dan terdapat pada level 3. Pemain harus menemukan tutup terlebih dahulu. Jika semua tempat air tidak tertutup maka pemain dianggap kalah.
8	Wadah Kosong		Konten ini harus di kumpulkan, dan dikubur masing-masing dalam lubang yang tersedia padal level 4. Jika pemain tidak mengubur semua wadah kosong maka dianggap kalah.

3.2.2 *Story Board*

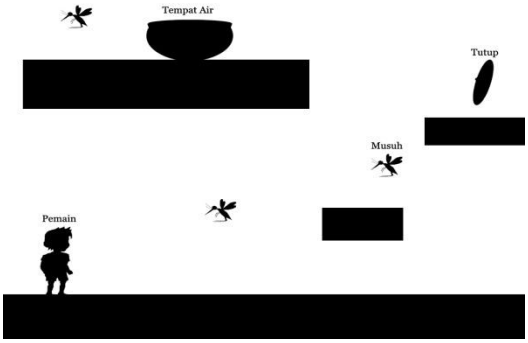
Story board adalah sketsa gambar yang disusun berurutan sesuai dengan naskah. *Story board* dapat menyampaikan ide cerita kepada orang lain dengan lebih mudah, sehingga menghasilkan persepsi yang sama mengenai ide cerita kepada seseorang hanya dengan melihat *story board.* Berikut merupakan *story board* dari *game Save Family From Malaria* :



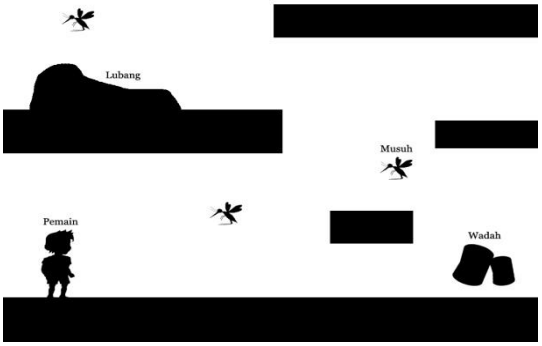
Gambar 3.36 *Story Board* Level 1



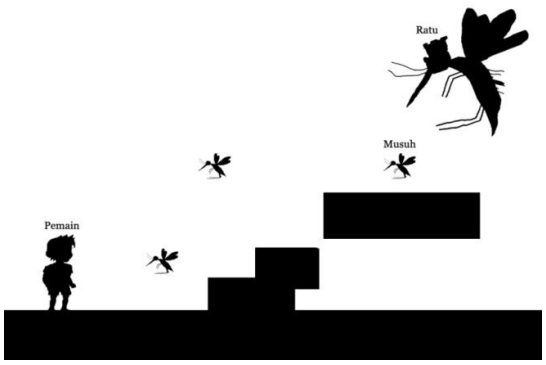
Gambar 3.37 *Story Board* Level 2



Gambar 3.38 *Story Board* Level 3



Gambar 3.39 *Story Board* Level 4



Gambar 3.40 *Story Board* Level 5

1. *Story Board Level* 1

Story board untuk *level* 1 pada aplikasi ini dapat dilihat pada Gambar 3.36. Pemain dapat bergerak ke arah kiri, kanan dalam peta, yang ditandai dengan jalan berwarna hitam. Musuh dapat melakukan serangan fisik terhadap pemain. Untuk dapat melanjutkan ke *level* 2, pemain harus memenangkan misi.

2. *Story Board Level* 2

Story board untuk *level* 2 pada aplikasi ini dapat dilihat pada Gambar 3.37. Pemain dapat bergerak ke arah kiri, kanan dalam peta yang ditandai dengan jalan berwarna hitam. Pada *level* 2 ini pemain harus melawan nyamuk dan menguras tempat air atau bak mandi. Pemain harus menyelesaikan misi untuk dapat melanjutkan ke level selanjutnya.

3. *Story Board Level* 3

Story board untuk *level* 3 pada aplikasi ini dapat dilihat pada Gambar 3.38. Pemain dapat bergerak ke arah kiri dan kanan dalam peta yang ditandai dengan jalan berwarna hitam. Pemain harus menutup semua tempat air dan membrantas nyamuk. Pemain harus menyelesaikan misi untuk melanjutkan level selanjutnya.

4. *Story Board Level* 4

Story board untuk *level 4* pada aplikasi ini dapat dilihat pada Gambar 3.39. Pemain dapat bergerak ke arah kakan dan kiri dalam peta yang ditandai dengan jalan berwarna hitam. Pada *level 4* ini pemain harus mengubur wadah kosong dan membrantas nyamuk. Pemain harus menyelesaikan misi untuk dapat melanjutkan ke level selanjutnya.

5. *Story Board Level 5*

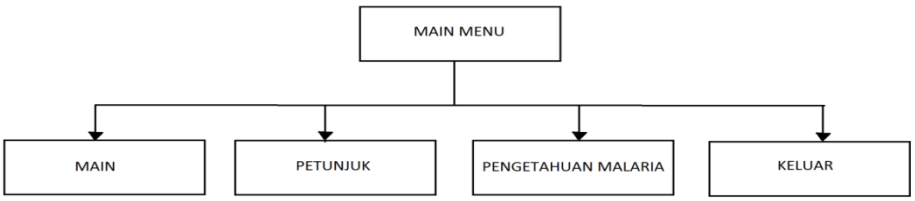
Story board untuk *level 5* pada aplikasi ini dapat dilihat pada Gambar 3.40. Pemain dapat bergerak ke arah kiri dan kanan sesuai dengan area jalan berwarna hitam. Pemain harus membrantas semua nyamuk dan mengalahkan ratu. Pemain harus menyelesaikan misi untuk memenangkan keseluruhan game.

3.2.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak merupakan kegiatan untuk merancang aplikasi yang akan dibangun, tahapannya mulai dari perancangan arsitektur menu, kemudian berlanjut ke perancangan antarmuka.

3.2.3.1 Perancangan Arsitektur Menu

Perancangan struktur menu merupakan jalur pemakaian sistem yang mudah dipahami dan mudah digunakan. Untuk lebih jelasnya dapat dilihat pada Gambar 3.41 di bawah ini.



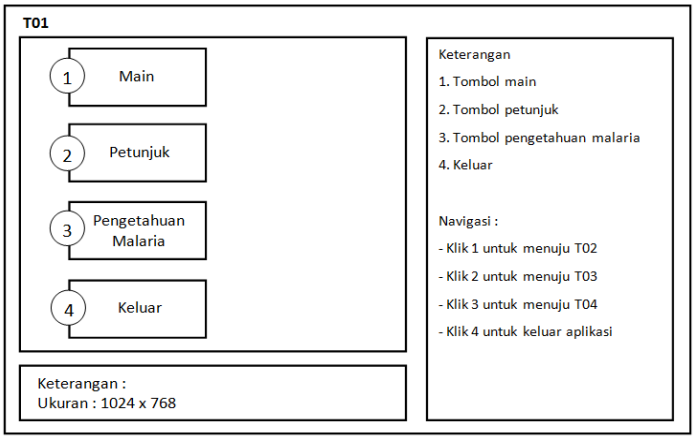
Gambar 3.41 Arsitektur Menu *Game Save Family From Malaria*

1. Menu Main : merupakan submenu untuk memulai permainan baru
2. Menu Petunjuk : merupakan submenu untuk menampilkan petunjuk permainan
3. Menu Pengetahuan Malaria : merupakan submenu untuk menampilkan pengetahuan malaria
4. Menu Keluar: merupakan submenu untuk keluar dari aplikasi

3.2.3.2 Perancangan Antarmuka

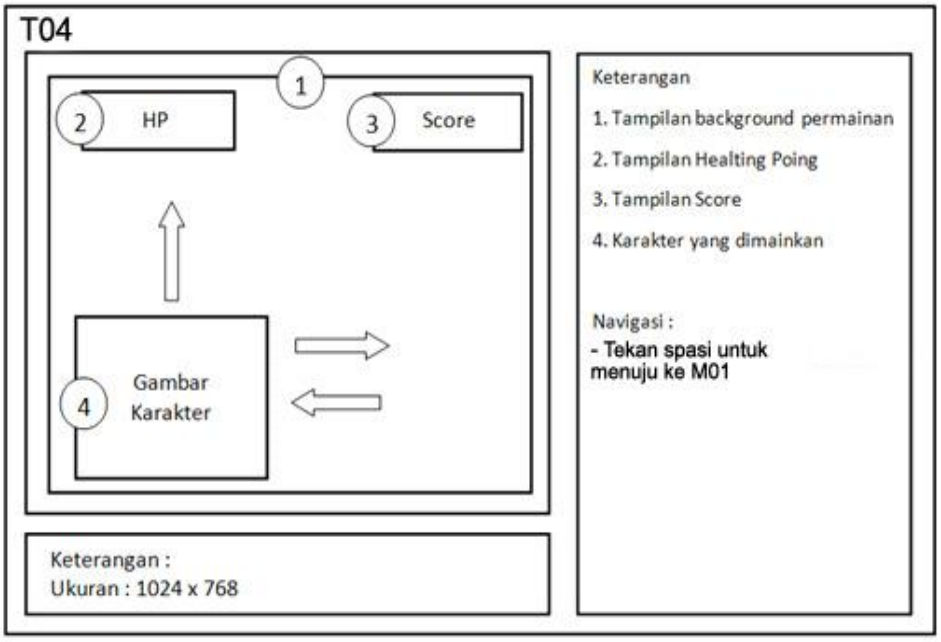
Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga mempermudah dalam meng-implementasikan aplikasi serta memudahkan pembuatan aplikasi. Berikut ini merupakan perancangan antarmuka pada *game Save Family From Malaria*:

1. Perancangan Antarmuka Menu Utama



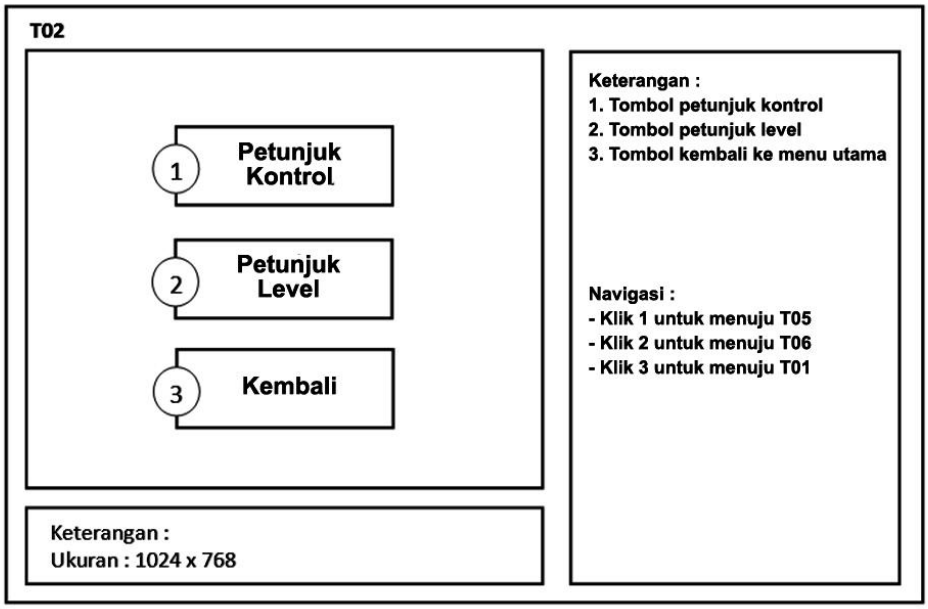
Gambar 3.42 Perancangan Antarmuka Menu Utama

2. Perancangan Antarmuka Menu Main



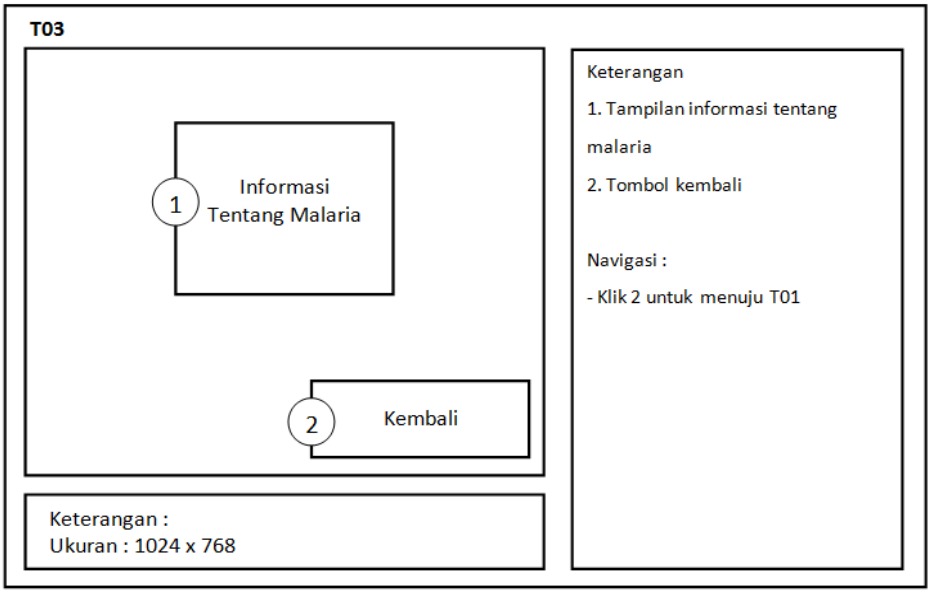
Gambar 3.43 Perancangan Antarmuka Menu Main

3. Perancangan Antarmuka Menu Petunjuk



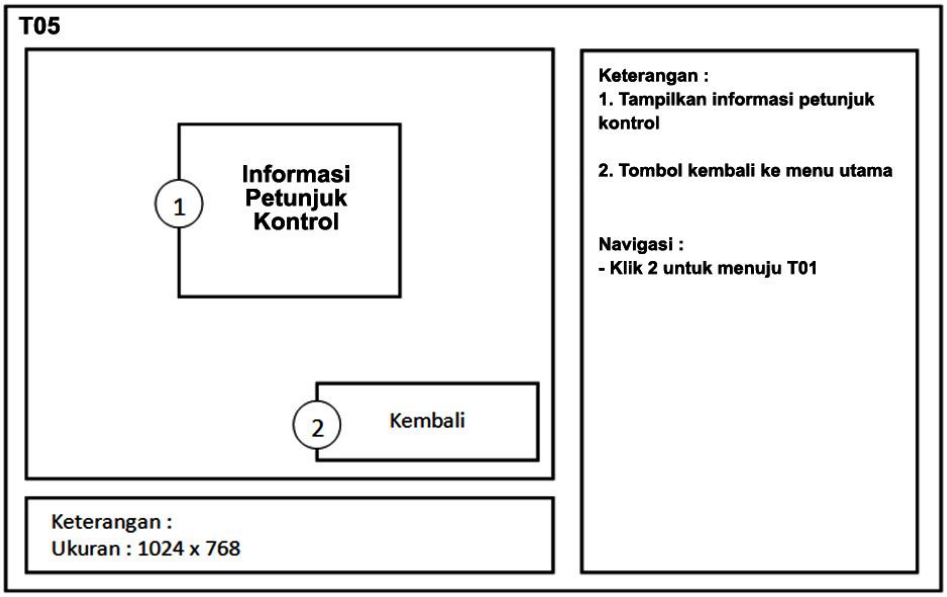
Gambar 3.44 Perancangan Antarmuka Petunjuk

4. Perancangan Antarmuka Pengetahuan Malaria



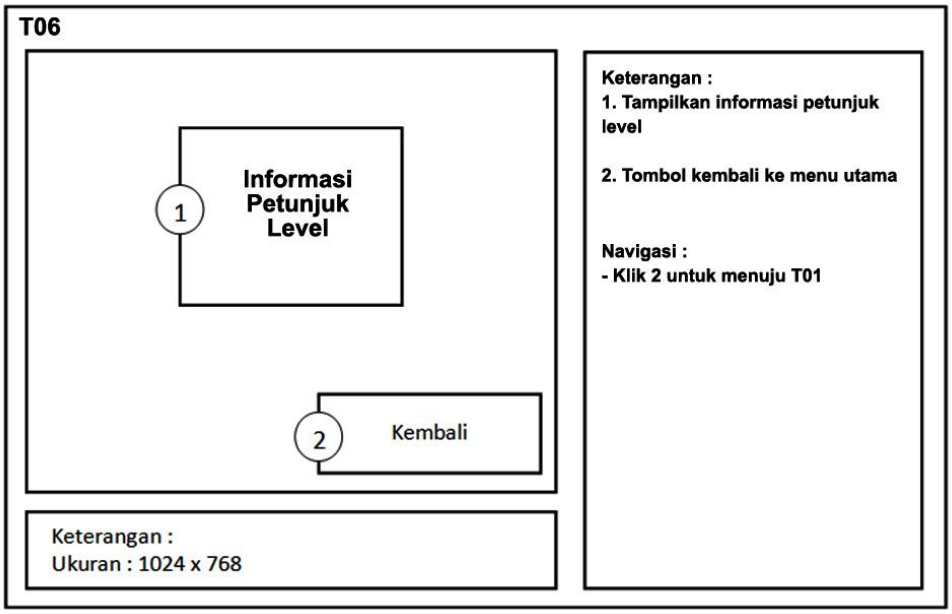
Gambar 3.45 Perancangan Antarmuka Pengetahuan Malaria

5. Perancangan Antarmuka Petunjuk Kontrol



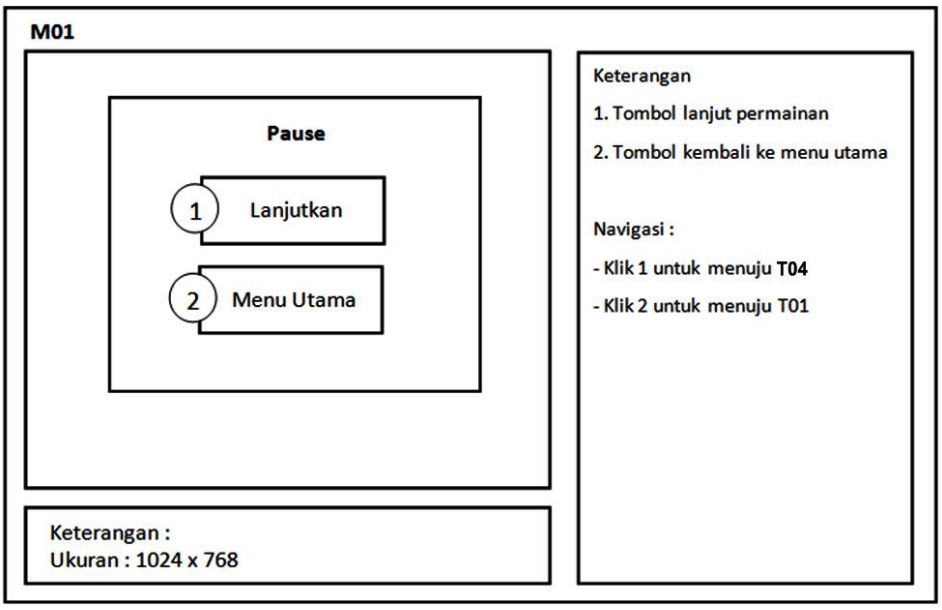
Gambar 3.46 Perancangan Antarmuka Petunjuk Kontrol

6. Perancangan Antarmuka Petunjuk Level



Gambar 3.47 Perancangan Antarmuka Petunjuk Level

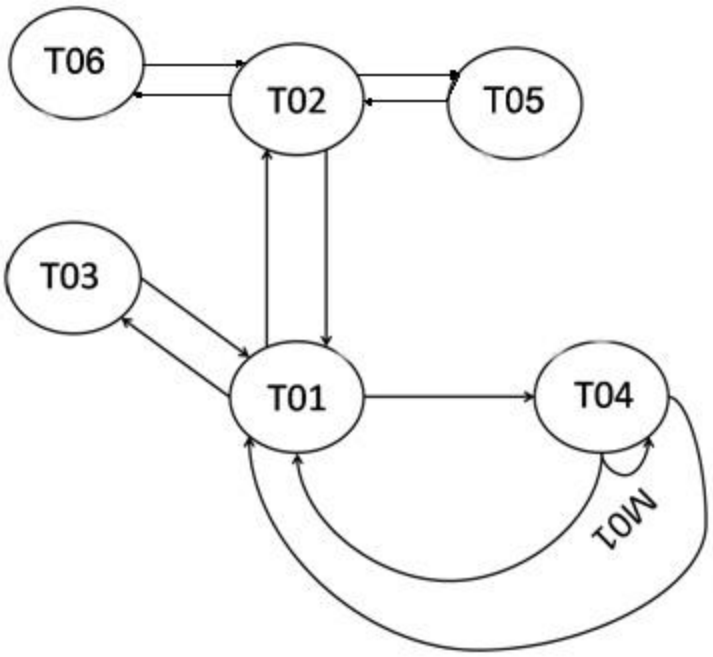
7. Perancangan Antarmuka *Pause*



Gambar 3.48 Perancangan Antarmuka *Pause*

3.2.3.3 Jaringan Semantik

Jaringan semantik adalah gambaran tentang relasi dari masing-masing halaman yang dapat diakses oleh seorang pengguna. Jaringan semantik ini digunakan sebagai panduan penggunaan menu-menu dalam sebuah aplikasi. Terdapat 4 tampilan pada aplikasi ini, yaitu T01 untuk tampilan menu utama aplikasi, T02 untuk tampilan menu petunjuk, T03 untuk tampilan pengetahuan malaria dan T04 untuk main. Perancangan semantik ini dapat dilihat pada Gambar 3.49.



Gambar 3.49 Jaringan Semantik

3.3 Perancangan Metode

Perancangan metode merupakan perancangan yang berfungsi untuk mendeskripsikan metode-metode yang berada di dalam aplikasi. Adapun metode-metode yang terdapat dalam *game Save Family From Malaria* yang akan dibangun adalah sebagai berikut :

1. Metode *keyDownHandler*

Metode ini digunakan untuk menggerakan karakter yang terdapat dalam game menggunakan tombol pada *keyboard*. Untuk lebih jelasnya dapat dilihat dari *pseudocode* pada gambar berikut ini :

```
if (e.keyCode == 37){  // Jika player menekan tombol kiri
       leftPressed = true;  // maka karakter akan begerak ke kiri
} else if (e.keyCode == 39){ // Jika player menekan tombol kanan
       rightPressed = true;  // maka karakter akan begerak ke kanan
}else if (e.keyCode == 38){ // Jika player menekan tombol atas
       upPressed = true;  // maka karakter akan bergerak loncat
} else if (e.keyCode == 32){ // Jika player menekan tombol spasi
       // maka akan menghentikan permainan sejenak atau pause
       dispatchEvent( new PauseEvent( PauseEvent.PAUSE) );
} else if(e.keyCode == 40){ //Jika player menekan tombol bawah
       downPressed = true;
       //Jika player menyentuh pintu hutan yang tertutup
       if(doorOpen &&
player.tar.hitTestObject(back.other.lockeddoor)){
              nextLevel(); //maka lanjut level selanjutnya
       }
}
```

Gambar 3.50 *Pseudocode* Metode *keyDownHandler*

2. Metode PetunjukPilih

Metode ini digunakan untuk menampilkan menu Pilih Petunjuk yang terdapat dalam aplikasi. Untuk lebih jelasnya dapat dilihat dari *pseudocode* pada gambar berikut ini :

```
// inisialisasi fungsi tombol yang akan dipanggil, seperti tombol
petunjuk kontrol, petunjuk level, dan kembali
pkontrol.addEventListener(MouseEvent.MOUSE_DOWN, mouseC1);
plevel.addEventListener(MouseEvent.MOUSE_DOWN, mouseC2);
pkembali.addEventListener(MouseEvent.MOUSE_DOWN, mouseC3);

// Membuat fungsi untuk tombol petunjuk kontrol
public function mouseC1(event:MouseEvent):void{
       addChild(PetKontrol); //Memanggil menu petunjuk kontrol
       kontrolAktif = true;
       PetKontrol.addEventListener( NavigationEvent.KEMBALIPETUNJUK,
       kembalipetunjuk); // Inisialisasi atau aktifasi tombol kembali
}
// Membuat fungsi untuk tombol petunjuk level
public function mouseC2(event:MouseEvent):void{
       addChild(PetLevel); //Memanggil menu petunjuk level
       levelAktif = true;
       PetLevel.addEventListener( NavigationEvent.KEMBALIPETUNJUK,
       kembalipetunjuk); // Inisialisasi atau aktifasi tombol kembali

}
// Membuat fungsi untuk tombol kembali
public function mouseC3(event:MouseEvent):void{
       // Jika tombol kembali ditekan, maka kembali ke menu utama
       dispatchEvent( new NavigationEvent( NavigationEvent.KEMBALI2)
);
}
```

Gambar 3.51 *Pseudocode* Metode PetunjukPilih

3. Metode MenuMalaria

Metode ini digunakan untuk menampilkan menu malaria yang terdapat dalam aplikasi. Untuk lebih jelasnya dapat dilihat dari *pseudocode* pada gambar berikut ini :

```
// inisialisasi fungsi tombol yang akan dipanggil, seperti tombol
kembali, selanjutnya dan sebelumnya
kembali.addEventListener(MouseEvent.MOUSE_DOWN, mouseC1);
slnj.addEventListener(MouseEvent.MOUSE_DOWN, mouseC2);
sblm.addEventListener(MouseEvent.MOUSE_DOWN, mouseC3);

// Membuat fungsi tombol kembali
public function mouseC1(event:MouseEvent):void{
       // Jika tombol kembali ditekan, maka akan kembali ke menu utama
       dispatchEvent(  new  NavigationEvent(  NavigationEvent.KEMBALI2)
);
}
// Membuat fungsi tombol selanjutnya untuk materi malaria
public function mouseC2(event:MouseEvent):void{
       nextFrame(); // Maka akan ke frame selanjutnya
}
// Membuat fungsi tombol sebelumnya untuk materi malaria
public function mouseC3(event:MouseEvent):void{
       prevFrame(); // Maka akan ke frame sebelumnya
}
```

Gambar 3.52 *Pseudocode* Metode MenuMalaria

4. Metode *hit*

Metode ini digunakan ketika ratu terkena peluru DDT. Untuk lebih jelasnya dapat dilihat dari *pseudocode* pada gambar berikut ini :

```
// inisialisasi terhadap movieclip ratu, yang terdiri dari animasi
frame yang diambil, posisi ratu akan kembali, pengurahan darah ratu
this.mc.gotoAndStop(2); // animasi frame yang diambil
getAstarMap(38); // posisi ratu akan kembali
health -=  20; // Nilai pengurangan darah ratu
var l =(153*(health*100/200))/100; //Perhitungan pengurangan darah ratu
this.mc.bar.bar.width = l;
if (health>0){ // Jika darah masih ada atau lebih dari nol
       // Waktu muncul atau pergerakan ratu dan kondisi ratu kebal atau
belum mati
       timers.start();
       timers.addEventListener(TimerEvent.TIMER,kebal);
}else{ // Jika tidak atau darah sama dengan nol
       hapus(); // hapus ratu dan rumah ratu
       // Animasi frame yang diambil
       par.back.collision.sarangnyamuk.gotoAndStop(2);
       par.playerScore2 +=1000; // Menambahkan score menjadi 1000
       par.updateTextFields2(); // Update ke bonus score
}
```

Gambar 3.53 *Pseudocode* Metode *hit*

5. Metode *updateHealthBar*

Metode ini memberikan efek darah yang dimiliki karakter saat berada dalam *game*. Untuk lebih jelasnya dapat dilihat dari *pseudocode* gambar berikut ini :

```
// Perhitungan kondisi darah pada karakter
percentHP = currentHP / maxHP; // Darah sama dengan darah yang ada
dibagi darah maksimal
health.bar.scaleX = percentHP; // Movieclip darah yang ada disamakan
dengan prhitungan darah, dan dimunculkan
```

Gambar 3.54 *Pseudocode* Metode *updateHealthBar*

6. Metode *fireBullet*

Metode ini membuat karakter mengeluarkan peluru atau semprotan obat nyamuk saat berada dalam *game*. Untuk lebih jelasnya dapat dilihat dari *pseudocode* pada gambar berikut ini :

```
// Inisialisasi terhadap posisi karakter atau arah karakter dan
movieclip bullet
var playerDirection:String;
var bullet:Bullet;

if(player.scaleX < 0){ //Jika scaleX dari karakter kurang dari 0
      playerDirection = "left"; //maka semprot peluru akan menghadap
ke kiri
}else if(player.scaleX > 0){ //Jika scaleX dari karakter lebi dari 0
      playerDirection = "right"; //maka semprot peluru akan menghadap
ke kanan
}

back.addChild(bullet); // Panggil movieclip bullet
semprotefek = semprotplayer.play(); // Animasi frame karakter semprot
// Inisialisasi atau aktifasi hapus peluru semprot sesuai arah yang
ditentukan
bullet.addEventListener(Event.REMOVED, bulletRemoved);
bulletList.push(bullet);
```

Gambar 3.55 *Pseudocode* Metode *fireBullet*