BAB III
ANALISIS DAN PERANCANGAN SISTEM

III.1 Analisis Sistem

Pada bagian analisis sistem ini akan dibahas mengenai analisa masalah kegunaan permainan yang akan dibangun, analisis kebutuhan non-fungsional dan analisis kebutuhan fungsional.

III.1.1 Analisis Masalah

Kebudayaan adalah ciptaan manusia yang indah. Masakan adalah salah satu dari kebudayaan yang indah itu. Negara Indonesia yang sangat luas dengan jumlah sukunya yang banyak telah melahirkan kebudayaan yang sangat beragam termasuk masakan. Perbedaan budaya yang membuat cara memasak dan meracik masakan yang berbeda, justru memperkaya khasanah masakan Indonesia [1] (Danny S, Tamtomo, Sandi, Karina Naftali, 2011). Kemudian timbul masalah sebagai berikut :

1. Bagaimana membangun suatu permainan yang dapat memberikan pembelajaran memasak kepada generasi muda?
2. Bagaimana membangun suatu permainan yang menantang, tidak monoton, interaktif, menarik serta menghibur?

III.1.2 Analisis Permainan

Pada bagian analisis permainan selamatkan masakan nusantara indonesia ini akan dibahas mengenai *storyline*, analisis tingkat kesulitan, *gameplay*, *scoring*, karakter, objek, *storyboard*, dan Analisis Algoritma A*.

III.1.2.1 Storyline

Permainan selamatkan masakan nusantara Indonesia adalah permainan yang bertujuan memberikan edukasi bagaimana cara membuat masakan Indonesia dan memerlukan bahan apa saja dalam membuatnya. Permainan ini menceritakan petualangan Aji, seorang koki yang lahir dari ibu seorang pemain sirkus yang handal dalam melempar pisau dan bapa yang handal dalam hal memasak. Aji memiliki keahlian dalam melempar pisau dan memasak yang diturunkan oleh

orang tuanya. Aji berusaha menghadapi rintangan yang dilewatinya dalam mengumpulkan bahan masakan untuk dijadikan sebuah masakan.

III.1.2.2 Analisis Tingkat Kesulitan

Dalam permainan ini, terdapat 3 level yang masing-masing level dibedakan berdasarkan alur cerita, tingkat kesulitan rintangan yang harus dihadapi, jenis musuh yang bertambah dan bertambahnya bahan yang diperlukan. Di setiap peningkatan level, pemain akan dihadapkan pada musuh dengan tingkat kesulitan yang akan bertambah, mulai dari kecepatan dan kesukaran musuh tersebut dan rintangan yang harus dihadapi.

Tabel III.1 Analisis Tingkat Kesulitan

Tingkat	Jumlah musuh	Total bahan masakan yang tersebar di permainan	Boss
Tingkat 1	1	8	Tidak ada
Tingkat 2	5	10	Ada
Tingkat 3	9	12	Ada

III.1.2.3 Gameplay

Tugas utama pemain dalam permainan ini adalah melewati setiap rintangan, mengumpulkan bahan masakan, dan membuat masakan. Pemain dapat bergerak ke kanan, ke kiri dan melompat. Pemain memiliki 10 poin darah dan akan kehilangan 1 darah apabila pemain mengenai musuh atau mengenai duri.

Pada tingkat 1 permain akan mengumpulkan bahan masakan nasi, oncom, bawang putih, bawang merah, cabe merah, cabe merah keriting, kencur, garam dan minyak goreng untuk membuat masakan nasi tutug oncom yang berasal dari jawa barat.

Pada tingkat 2 pemain akan mengumpulkan bahan masakan udang, lengkuas, kunyit, jahe, ketumbar, tepung, terigu dan minyak goreng untuk membuat masakan udang tepung yang berasal dari jawa timur.

Pada tingkat 3 pemain akan mngumpulkan bahan masakan ayam, bawang putih, bawang merah, kemiri, kayu manis, jeruk nipis, santan, garam, jahe, kecap manis, tomat dan minyak goreng untuk membuat masakan ayam santan yang berasal dari sumatera utara.

III.1.2.4 Scoring

Dalam permainan ini, pemain harus mengumpulkan bahan masakan yang tersebar pada setiap tingkat. Untuk setiap bahan masakan yang terkumpul akan kembali ke keadaan awal apabila pemain telah menghabiskan 10 *Health point* atau meninggalkan permainan. Pemain memiliki nilai 0 dan akan bertambah 20 apabila mengambil bahan masakan dan bertambah 10 apabila melempar pisau dan mengenai objek musuh.

III.1.2.5 Karakter

Berikut ini adalah perancangan karakter yang akan diterapkan di aplikasi game yang akan dibangun dan akan dijelaskan di tabel III.15.

Tabel III.2 Deskripsi karakter permainan

Karakter	Kemampuan	Keterangan
 Aji	Darah : 10 poin Serangan Pisau : 1 poin	Aji merupakan karakter yang dapat dimainkan di dalam pada permainan, memiliki kemampuan untuk jalan ke kiri , jalan ke kanan, melompat serta melempar pisau..
 Kelelawar	Darah : 2 poin Serangan : 1 poin	Kelelawar merupakan karakter yang memiliki kemampuan untuk menyerang, mengejar.
 Tikus	Darah : 1 poin Serangan : 1 poin	Tikus merupakan karakter yang memiliki kemampuan untuk menyerang, mengejar serta patroli.
 Boss Ayam	Darah : 5 poin Serangan : 1 poin	Boss Ayam merupakan karakter yang memiliki kemampuan untuk menyerang serta mengejar.
	Darah : 1 poin Serangan : 1 poin	Kepiting merupakan karakter yang memiliki kemampuan untuk

Karakter	Kemampuan	Keterangan
Kepiting		menyerang serta patroli.
 Boss Burung	Darah : 5 poin Serangan : 1 poin	Boss Burung merupakan karakter yang memiliki kemampuan untuk menyerang serta mengejar.

III.1.2.6 Objek

Berikut ini adalah perancangan objek yang akan diterapkan di aplikasi game yang akan dibangun dan akan dijelaskan di tabel III.16.

Tabel III.3 Deskripsi objek permainan

Objek	Deskripsi
 Kencur	Sebuah objek Kencur yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Nasi putih	Sebuah objek Nasi putih yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Oncom bakar	Sebuah objek Oncom bakar yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Cabe merah	Sebuah objek Cabe merah yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Cabe merah keriting	Sebuah objek Cabe merah keriting yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
	Sebuah objek Ketumbar yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.

Objek	Deskripsi
Ketumbar	
 Gula pasir	Sebuah objek Gula pasir yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Udang	Sebuah objek Udang yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Tepung terigu	Sebuah objek Tepung terigu yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Kunyit	Sebuah objek Kunyit yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Lengkuas	Sebuah objek Lengkuas yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Ayam	Sebuah objek Ayam yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Bawang merah	Sebuah objek Bawang merah yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Bawang putih	Sebuah objek Bawang putih yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.

Objek	Deskripsi
 Garam	Sebuah objek Garam yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Jahe	Sebuah objek Jahe yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Jeruk nipis	Sebuah objek Jeruk nipis yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Kayu manis	Sebuah objek Kayu manis yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Kemiri	Sebuah objek Kemiri yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Minyak goreng	Sebuah objek Minyak goreng yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Tomat	Sebuah objek Tomat yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Bahan halus	Sebuah Objek Bahan halus yang ditampilkan apabila bahan halus mencapai persenan 100.

Objek	Deskripsi
 Bahan potong	Sebuah Objek Bahan potong yang ditampilkan apabila bahan potong mencapai persenan 100.
 Santan	Sebuah objek Santan yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.
 Kecap manis	Sebuah objek Kecap manis yang dapat dikumpulkan untuk menambah nilai sebesar 20 poin.

III.1.2.7 Story Board

Story board adalah sketsa gambar yang disusun seperti halnya sebuah naskah cerita. Dengan *storyboard* kita dapat menyampaikan ide cerita yang kita punya kepada orang lain dengan cara yang lebih mudah, karena dengan *storyboard* kita dapat menggiring imajinasi seseorang menuju persepsi yang kita inginkan melalui gambar-gambar-gambar yang sudah kita buat.



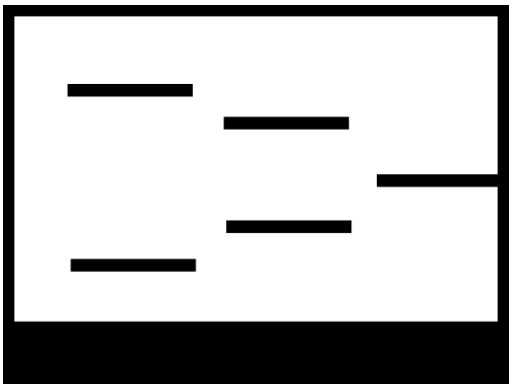
Gambar III.1 Storyboard Komik_Nasi_Tutug_Oncom



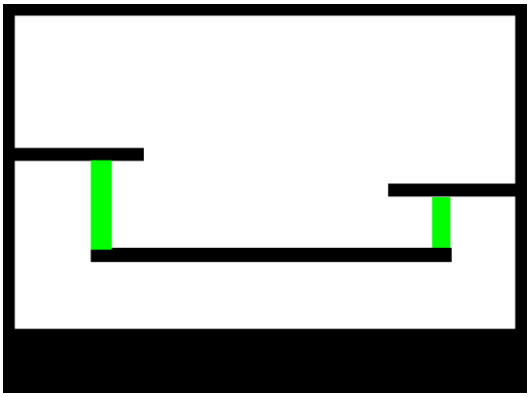
Gambar III.2 Storyboard Tutor_1



Gambar III.3 Storyboard Tutor_2



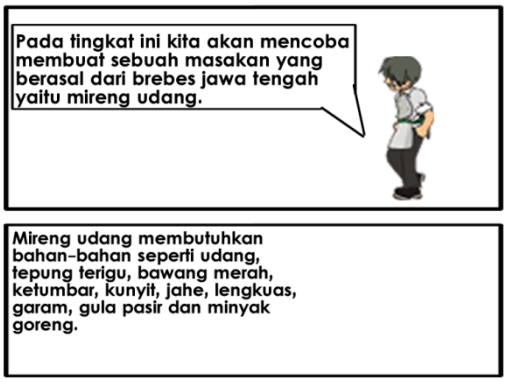
Gambar III.4 Storyboard Tutor_3



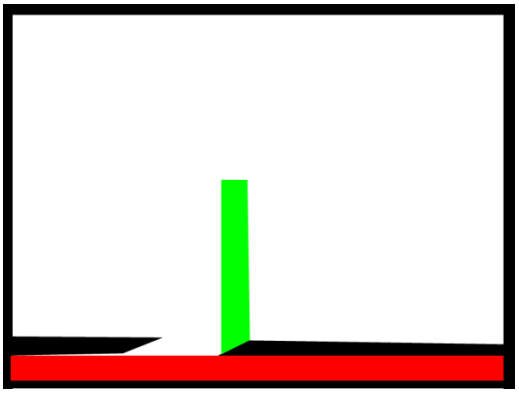
Gambar III.5 Storyboard Tutor_4



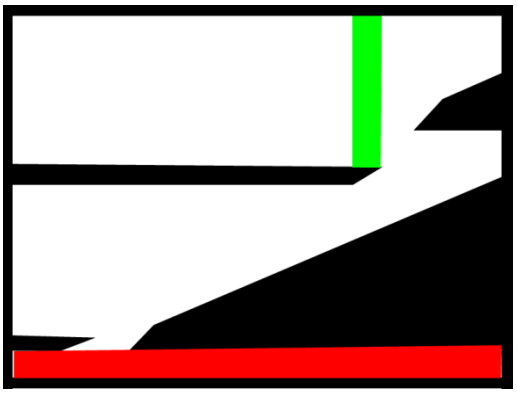
Gambar III.6 Storyboard Masak_Nasi_Tutug_Oncom



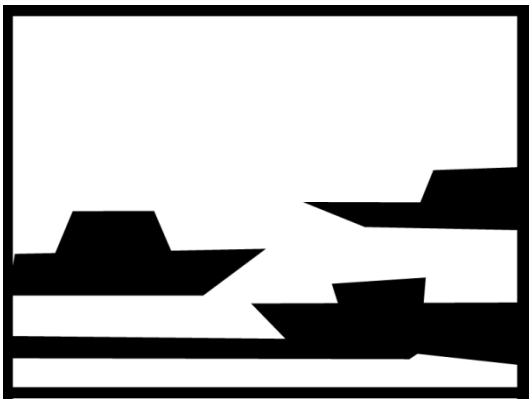
Gambar III.7 Storyboard Komik_Mireng_Udang



Gambar III.8 Storyboard Tingkat_2_1



Gambar III.9 Storyboard Tingkat_2_2



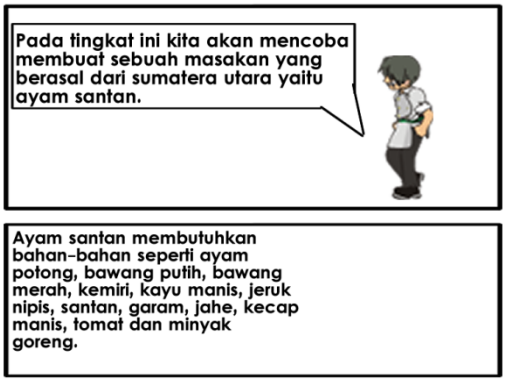
Gambar III.10 Storyboard Tingkat_2_3



Gambar III.11 Storyboard Tingkat_2_Boss



Gambar III.12 Storyboard Masak_Mireng_Udang



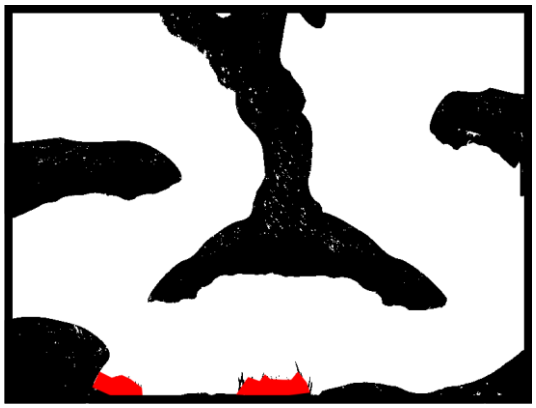
Gambar III.13 Storyboard Komik_Masak_Ayam_Santan



Gambar III.14 Storyboard Tingkat_3_1



Gambar III.15 Storyboard Tingkat_3_2



Gambar III.16 Storyboard Tingkat_3_3



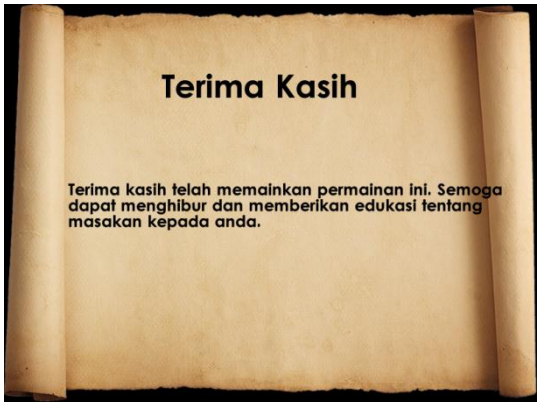
Gambar III.17 Storyboard Tingkat_3_4



Gambar III.18 Storyboard Tingkat_3_Boss



Gambar III.19 Storyboard Masak_Ayam_Santan



Gambar III.20 Storyboard Terima_Kasih

III.1.2.8 Analisis Algoritma A*

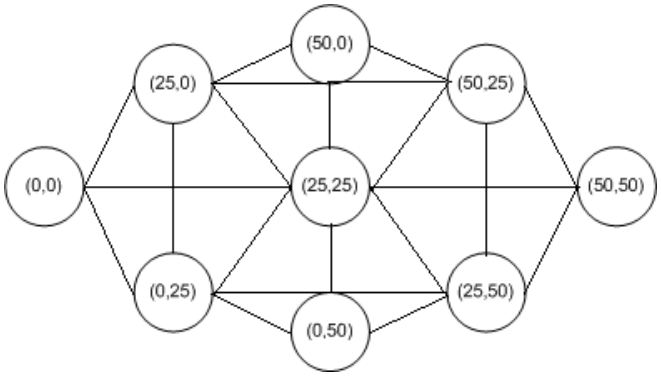
Pada bagian ini akan dibahas analisis algoritma A*, algoritma ini menjadi *complete* dan optimal dikarenakan algoritma ini menggabungkan informasi dari algoritma Djikstra (lebih memilih *node* yang dekat dengan *node* awal) dan informasi yang digunakan algoritma *Best First Search* (lebih memilih *node* yang dekat dengan *node* tujuan). Algoritma A* akan diterapkan pada objek *boss* dalam melakukan pencarian rute terpendek lokasi karakter pemain di arena permainan. Berikut ini adalah istilah-istilah yang akan dibahas:

1. *Open List* adalah *list* yang menyimpan kemungkinan *path* yang akan diperiksa. *Open list* digunakan untuk menentukan secara selektif (berdasarkan nilai f) jalan yang dikira lebih dekat menuju tujuan. *Open list* berisi *node* yang masih memiliki peluang untuk terpilih sebagai *best node*.
2. *Closed list* adalah *list* untuk menyimpan simpul-simpul yang sudah pernah dibangkitkan dan sudah pernah terpilih sebagai *best node*. *Closed list* dan *open list* bertujuan untuk menghindari penulusuran berulang rute agar tidak masuk kembali ke *open list*.
3. Nilai F adalah nilai yang menyatakan seberapa bagus sebuah *node*. Nilai F merupakan hasil dari f(n).
4. Nilai G adalah nilai dari fungsi g(n) adalah banyaknya langkah untuk menuju *path* saat ini.

5. Nilai H adalah nilai estimasi untuk sampai pada suatu tujuan mulai dari n.

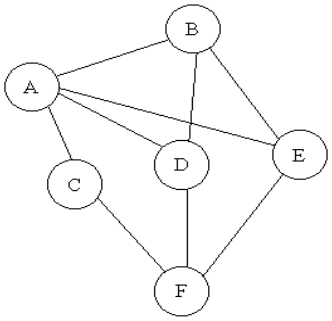
Diasumsikan arena permainan dibagi-bagi kedalam sebuah *tilemap* berbentuk *array* dua dimensi, setiap *tile* memiliki koordinat X dan Y dan dapat digambarkan melalui struktur data *graph*. Struktur data *graph* adalah struktur data yang berbentuk jaringan, hubungan antar elemennya adalah *many- to-many*. Berikut ini adalah contoh arena permainan yang terdapat pada permainan selamatkan masakan nusantara indonesia dapat dilihat pada gambar III.21

	0	1	2
0	(0,0)	(25,0)	(50,0)
1	(0,25)	(25,25)	(50,25)
2	(0,50)	(25,50)	(50,50)



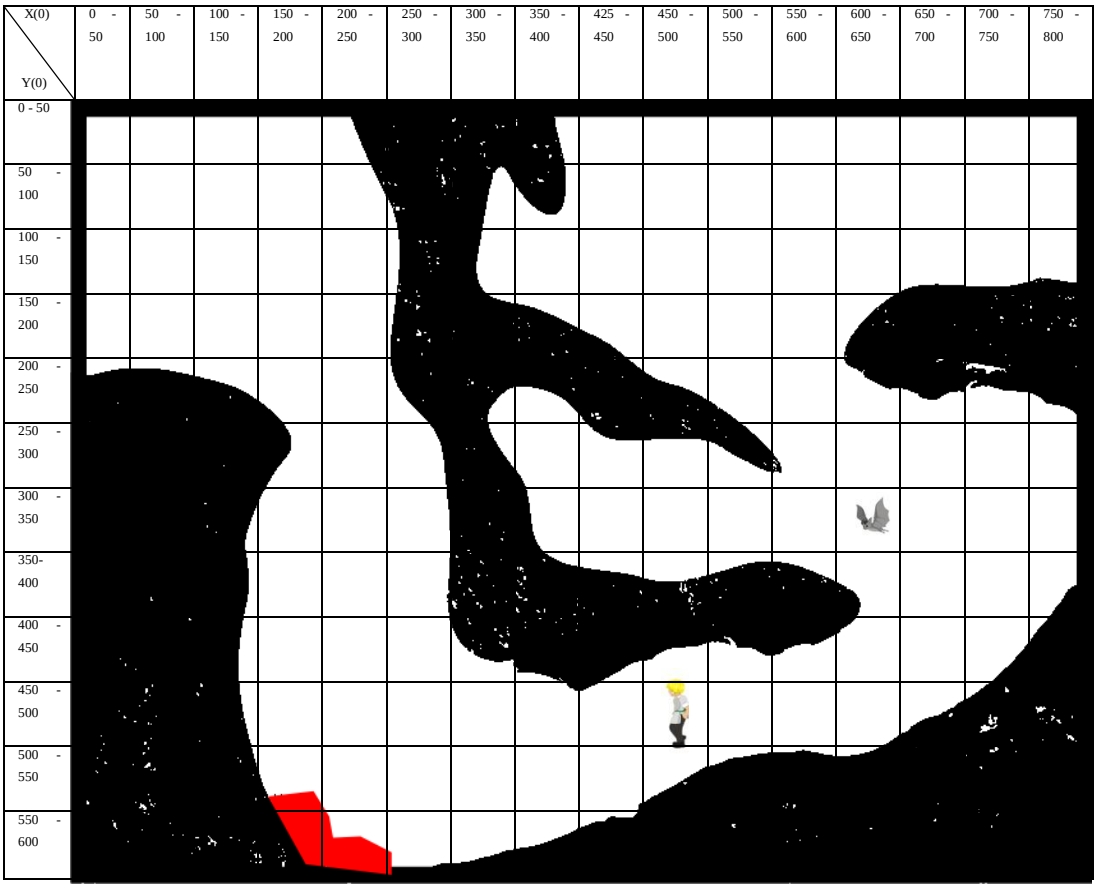
Gambar III.21 Arena Permainan

Secara umum terdapat dua macam representasi dari struktur data graf yang dapat diimplementasikan. Pertama, disebut *adjacency list*, yaitu *list* yang diimplementasikan dengan menampilkan masing-masing simpul sebagai sebuah struktur data yang mengandung senarai dari semua simpul yang saling berhubungan. Yang kedua adalah representasi berupa *adjacency matrix* dimana baris dan kolom dari matriks tersebut merepresentasikan simpul awal dan simpul tujuan dan sebuah entri didalam senarai yang menyatakan apakah terdapat sisi di antara kedua simpul tersebut. Dalam pembangungan *game* Selamatkan Masakan Nusantara Indonesia akan digunakan representasi berupa *adjacency list*. Berikut ini adalah contoh *adjacency list* dapat dilihat pada gambar III.22.



Gambar III.22 Contoh Adjacency List

Pada permainan selamatkan masakan nusantara indonesia, arena permainan dibagi kedalam *tilemap* yang terdiri dari *node* dengan orde X dan Y. tiap *node* memiliki koordinat X dan Y atau berbentuk *array* dua dimensi. Berikut ini adalah contoh kasus algoritma A* pada permainan selamatkan masakan nusantara indonesia dapat dilihat pada gambar III.23.



Gambar III.23 Contoh Kasus Algoritma A* pada permainan

Tile atau *node* direpresentasikan dengan nilai n = node (x,y). *Node* tujuan atau pada kasus ini disebut *node* pemain direpresentasikan dengan karakter pemain dan berada di koordinat (475,475) sedangkan *node* awal atau pada kasus ini disebut musuh direpresentasikan dengan karakter kelelawar dan berada di *node* (625,325), sedangkan *node* berwarna hitam merupakan penghalang atau dinding. Musuh dapat bergerak ke semua *node* yang berwarna putih tetapi musuh tidak dapat bergerak ke arah *node* yang berwarna hitam.

Untuk pencarian nilai G horizontal, G vertical dan G diagonal menggunakan rumus matematika :

1. **G(n) horizontal = abs(n.x - musuh.x)**
2. **G(n) vertical = abs(n.y - musuh.y)**
3. **G(n) diagonal = √ ((n.x - musuh.x) ² + (n.y – musuh.y) ²)**

Contoh untuk pencarian nilai G(n) horizontal :

1. Jika nilai yang akan dicari G(575,325)
2. Node musuh(625,325)
3. Maka G(575,325) horizontal = abs(575 – 625) = 50

Contoh untuk pencarian nilai G(n) vertical :

1. Jika nilai yang akan dicari G(625,275)
2. Node musuh(625,325)
3. Maka G(625,275) vertical = abs(275 – 325) = 50

Contoh untuk pencarian nilai G(n) diagonal :

1. Jika nilai yang akan dicari G(575,275)
2. Node musuh(625,325)
3. Maka G(575,275) diagonal = √((575 - 625) ² + (275 – 325) ²)
4. Maka dihasilkan nilai G(575,275) diagonal = √(2500 + 2500) = 70

Nilai G untuk pergerakan secara horizontal dan vertikal adalah 50, sedangkan untuk nilai untuk pergerakan diagonal adalah 70. Untuk mendapatkan nilai heuristik yang memungkinkan pergerakan ke 8 arah maka digunakan notasi matematika :

H(n) = (abs(n.x-pemain.x) + abs(n.y-pemain.y))

Contoh untuk pencarian nilai H(n) :

1. Jika nilai yang akan dicari H(575,325)
2. Node pemain(475,475)
3. Maka H(575,325) = (abs(575 – 475) + (abs(325 – 475))
4. Maka dihasilkan nilai H(25,75) = (abs(100)) + (abs(-150)) = 250

Nilai F dihasilkan dari perhitungan G dan H. Untuk mendapatkan nilai F maka digunakan notasi matematika :

$$\mathbf{F(n) = G(n) + H(n)}$$

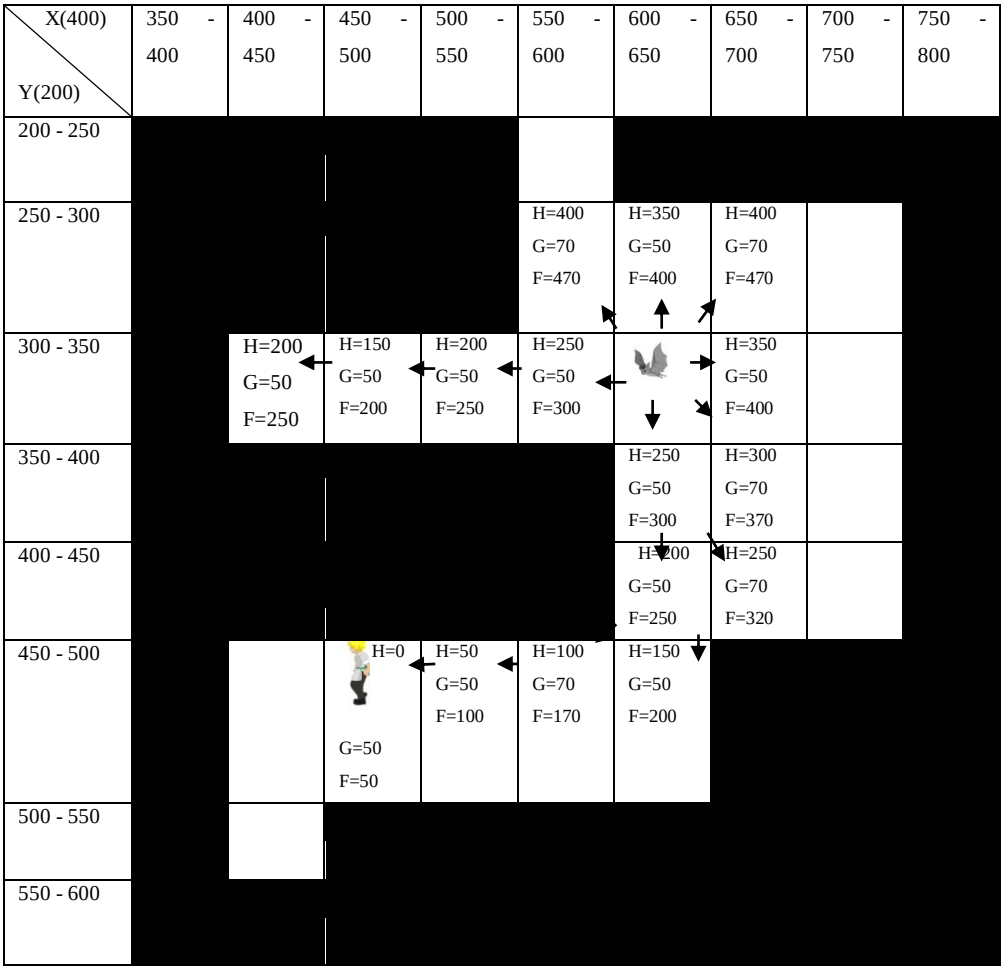
Contoh untuk pencarian nilai F(n) :

1. Jika nilai yang akan dicari F(575,325)
2. Maka F(575,325) = 50 + 250 = 300

Selanjutnya *node* musuh akan melakukan pencarian untuk menemukan jalur terpendek dengan cara memeriksa *node* disekitarnya. Langkah-langkah pencarian jalur terpendek adalah sebagai berikut:

1. *Node* musuh dimasukkan ke dalam *open list*.
2. Periksa semua *node* yang berdekatan dengan *node* musuh apakah *node* tersebut merupakan dinding atau tidak, abaikan *node* yang tidak bisa dilalui, lalu masukkan *node* tersebut kedalam *open list*. Dan untuk setiap *node* baru yang ditambahkan kedalam *open list*, simpan *node* lama sebagai "*parent*" yang akan digunakan untuk menelusuri jalan.
3. *Node* dengan nilai f terkecil akan terpilih sebagai *best node* dan selanjutnya pencarian akan dimulai dari *node* ini.
4. Hapus *node* musuh atau *node* sebelumnya dari *open list* dan masukkan ke dalam *closed list*.

Berikut ini adalah gambar membangkitkan suksesor *node* musuh dapat dilihat di gambar III.24.



Gambar III.24 Pencarian nilai A*

Proses pencarian 1 dengan langkah 1, *openlist* = (625,325). Langkah 2, *openlist* = (575,325), (575,275), (625,275), (675,275), (675,325), (675,375) dan (625,375). *Parent* = (625,325). Langkah 3, *bestnode* = (625,375) dan (575,325). Langkah 4 *closedlist* = (625,325), (575,325), (575,275), (625,275), (675,275), (675,325), (675,375) dan (625,375).

Proses pencarian 2 dengan langkah 1, *openlist* = (625,375) dan (575,325). Langkah 2, *openlist* = (525,325), (675,425) dan (625,425). *Parent* = (625,375) dan (575,325). Langkah 3, *bestnode* = (525,325) dan (625,425). Langkah 4 *closedlist* = (625,375), (575,325), (525,325), (675,425) dan (625,425).

Proses pencarian 3 dengan langkah 1, *openlist* = (525,325) dan (625,425). Langkah 2, *openlist* = (475,325) dan (625,475). *Parent* = (525,325) dan (625,425).

Langkah 3, *bestnode* = (475,325) dan (625,475). Langkah 4 *closedlist* = (525,325), (625,425), (475,325) dan (625,475).

Proses pencarian 4 dengan langkah 1, *openlist* = (475,325) dan (625,475). Langkah 2, *openlist* = (425,325) dan (575,475). *Parent* = (475,325) dan (625,475). Langkah 3, *bestnode* = (575,475). Langkah 4 *closedlist* = (475,325), (625,475), (425,325) dan (575,475).

Proses pencarian 5 dengan langkah 1, *openlist* = (575,475). Langkah 2, *openlist* = (525,475). *Parent* = (575,475). Langkah 3, *bestnode* = (525,475). Langkah 4 *closedlist* = (575,475) dan (525,475).

Proses pencarian 6 dengan langkah 1, *openlist* = (525,475). Langkah 2, *openlist* = (475,475). *Parent* = (525,475). Langkah 3, *bestnode* = (475,475). Langkah 4 *closedlist* = (525,475) dan (475,475).

Pencarian rute terpendek dari *node* musuh menuju *node* pemain menggunakan algoritma A* menghasilkan rute sebagai berikut:

1. Dari *node* musuh bergerak ke koordinat (625,375) dan (575,325) dengan f = 300.
2. Dari *node* dengan f = 300 bergerak ke koordinat (525,325) dan (625,425) dengan f = 250.
3. Dari *node* dengan f = 250 bergerak ke koordinat (475,325) dan (625,475) dengan f = 200.
4. Dari *node* dengan f = 200 bergerak ke koordinat (575,475) dengan f = 170.
5. Dari *node* dengan f = 170 bergerak ke koordinat (525,475) dengan f = 100.
6. Dari *node* dengan f = 100 bergerak ke koordinat (475,475) dan menemukan *goal* (*node* pemain).

III.1.3 Analisis Kebutuhan Non Fungsional

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

menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

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

III.1.3.1 Analisis Kebutuhan Perangkat Keras

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

1. Processor Intel Pentium IV
2. RAM 512 MB
3. VGA 128 MB
4. Monitor dengan resolusi minimal 800 x 600
5. Speaker atau Headset
6. Mouse
7. Keyboard

III.1.3.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan perintah - perintah yang diberikan kepada perangkat keras agar bisa saling berinteraksi diantara keduanya. Perangkat lunak yang dibutuhkan untuk membangun aplikasi selamatkan masakan nusantara Indonesia ini adalah sebagai berikut :

1. Sistem Operasi Windows XP/Vista/7
2. Greenfoot versi 2.2.1
3. jdk-1_5_0_07-windows-i586-p ke atas (JDK 1.5 minimal).
4. CorelDRAW X4
5. Photoshop 3

Adapun perangkat lunak yang dibutuhkan dalam penerapan aplikasi *game* edukasi ini adalah sebagai berikut :

1. Sistem Operasi Windows XP/Vista/7
2. jdk-7u13-windows-i586

III.1.3.3 Analisis Pengguna

Selain dibutuhkannya perangkat lunak dan perangkat keras, pengguna juga sangat dibutuhkan dalam penggunakaan aplikasi permainan ini. Adapun spesifikasi pengguna yang dibutuhkan adalah sebagai berikut :

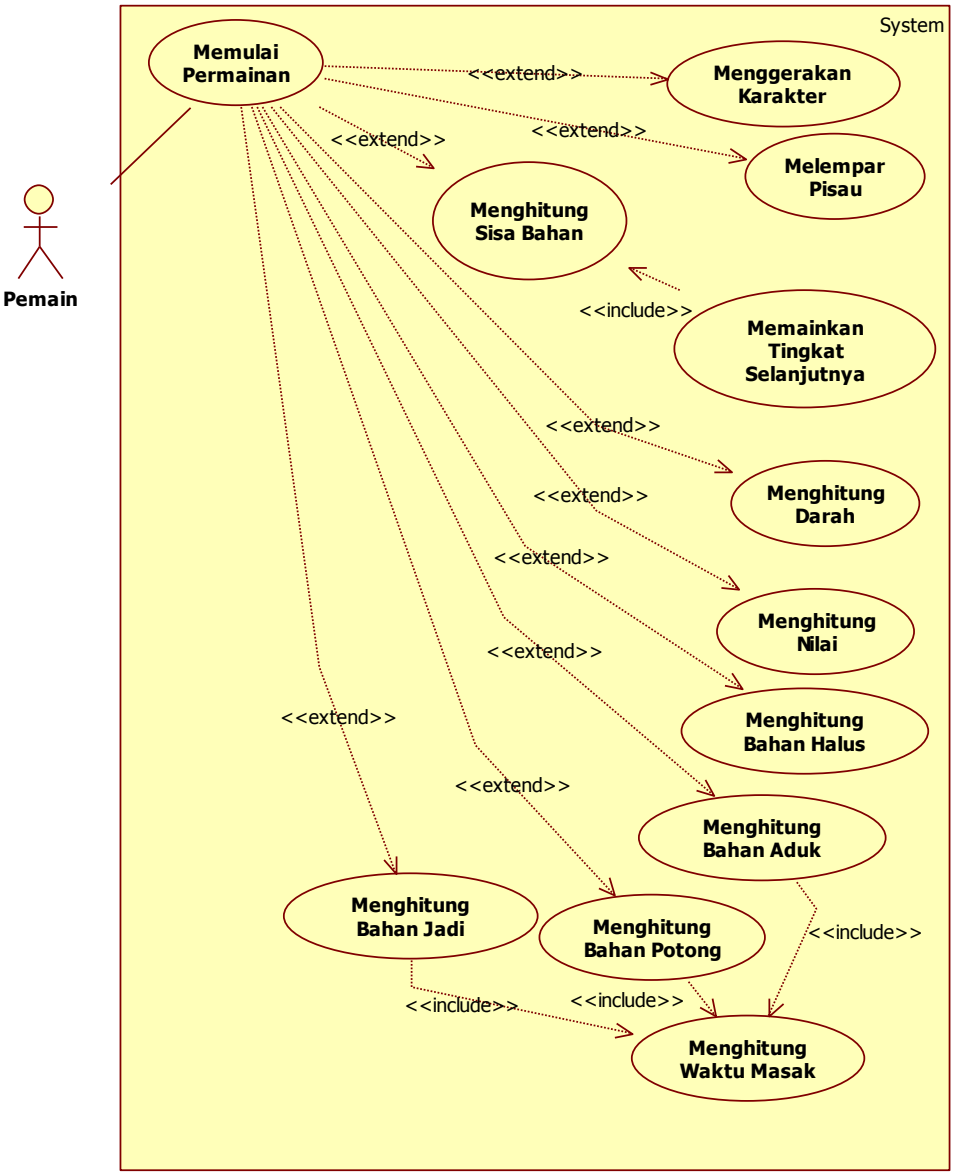
1. Peran : Pemain
2. Tingkat keterampilan : Bisa menggunakan computer
3. Pengalaman pada komputer : Pernah memainkan aplikasi permainan

III.1.4 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.

III.1.4.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk tingkah laku system yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.Berikut ini adalah perancangan *use case* diagram dari *game* Selamatkan Masakan Nusantara Indonesia :



Gambar III.25 Use Case Diagram game

III.1.4.2 Definisi Aktor

Merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat. Dalam permainan ini, hanya terdapat seorang aktor yaitu Pemain.

III.1.4.3 Definisi *Use Case*

Merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antara unit atau aktor.

Tabel III.4 Definisi Use Case

No	Use Case	Deskripsi
1.	Memulai permainan	Proses untuk memulai permainan
2.	Menggerakan karakter	Proses untuk menggerakan karakter
3.	Melempar pisau	Proses untuk melempar pisau
4.	Menghitung sisa bahan	Proses untuk menampilkan informasi sisa bahan masakan pada permainan
5.	Menghitung darah	Proses untuk menampilkan informasi darah pemain
6.	Menghitung nilai	Proses untuk menampilkan informasi nilai yang dihasilkan oleh pemain
7.	Memainkan tingkat selanjutnya	Proses untuk melanjutkan ke tingkat berikutnya
8.	Menghitung bahan halus	Proses untuk menampilkan informasi masak bahan halus
9.	Menghitung bahan aduk	Proses untuk menampilkan informasi masak bahan goreng
10.	Menghitung bahan potong	Proses untuk menampilkan informasi masak bahan potong
11.	Menghitung bahan jadi	Proses untuk menampilkan informasi masak bahan jadi
12.	Menghitung waktu masak	Proses untuk menampilkan informasi waktu masak

III.1.4.4 Skenario *Use Case*

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

1. Skenario *Use Case* Memulai permainan

Tabel III.5 Skenario *Use Case* Memulai permainan

Identifikasi		
Nomor	1	
Nama	Memulai permainan	
Tujuan	Memulai permainan	
Deskripsi	Proses untuk memulai permainan	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di menu utama	
Aksi aktor		Reaksi sistem
1. Memilih menu mulai		
		2. Menampilkan layar permainan
Skenario Alternatif	-	

Kondisi Akhir	Sistem melakukan inisialisasi konten yang dibutuhkan dalam permainan

2. Skenario *Use Case* Menggerakkan karakter

Tabel III.6 Skenario Use Case Menggerakkan karakter

Identifikasi		
Nomor	2	
Nama	Menggerakan karakter	
Tujuan	Menggerakan karakter	
Deskripsi	Proses untuk menggerakan karakter	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di arena permainan	
Aksi aktor		Reaksi sistem
1. Menekan tombol A pada *keyboard*		
		2. Menggerakan karakter pemain ke arah kiri
3. Menekan tombol D pada *keyboard*		
		4. Menggerakan karakter pemain ke arah kanan
5. Menekan tombol Spasi pada *keyboard*		
		6. Karakter pemain akan melompat
Skenario Alternatif	-	
Kondisi Akhir	Sistem melakukan gerakan sesuai dengan tombol yang sudah diberikan perintah pada sistem	

3. Skenario *Use Case* Melempar pisau

Tabel III.7 Skenario *Use Case* Melempar pisau

Identifikasi		
Nomor	3	
Nama	Melempar pisau	
Tujuan	Melempar pisau	
Deskripsi	Proses untuk melempar pisau	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di arena permainan	
Aksi aktor		Reaksi sistem
1. Menekan tombol S pada *keyboard*		
		2. Objek musuh terkena pisau dan akan hilang dari arena permainan
Skenario Alternatif		
1. Menekan tombol S pada *keyboard*		

	2. Pisau hilang pada jarak lebih dari 200 dari pemain atau mengenai tembok
Kondisi Akhir	Sistem memberikan respon yaitu melempar pisau

4. Skenario *Use Case* Menghitung sisa bahan

Tabel III.8 Skenario *Use Case* Melihat menghitung sisa bahan

Identifikasi	
Nomor	4
Nama	Menghitung sisa bahan
Tujuan	Menghitung sisa bahan
Deskripsi	Proses untuk menghitung bahan masakan yang tersisa pada arena permainan
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di arena permainan
Aksi aktor	Reaksi sistem
1. Pemain mengambil bahan masakan	
	2. Menampilkan informasi sisa bahan masakan
Skenario Alternatif	-
Kondisi Akhir	Sistem menampilkan informasi sisa bahan masakan

5. Skenario *Use Case* Menghitung darah

Tabel III.9 Skenario *Use Case* Menghitung darah

Identifikasi	
Nomor	5
Nama	Menghitung darah
Tujuan	Menghitung sisa darah pemain
Deskripsi	Proses untuk melihat informasi darah pemain
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di arena permainan
Aksi aktor	Reaksi sistem
	1. Musuh mengenai pemain
	2. Darah pemain berkurang 1 poin
	3. Update informasi darah
Skenario Alternatif	
1. Pemain menggerakan karater dan mengenai duri	
	2. Darah pemain berkurang 1 poin
	3. Update informasi darah

Kondisi Akhir	Sistem menampilkan informasi darah pemain

6. Skenario *Use Case* Menghitung nilai

Tabel III.10 Skenario *Use Case* Menghitung nilai

Identifikasi		
Nomor	6	
Nama	Menghitung nilai	
Tujuan	Menghitung nilai yang dihasilkan pemain	
Deskripsi	Proses untuk melihat informasi nilai yang dihasilkan pemain	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di arena permainan	
Aksi aktor		Reaksi sistem
1. Membunuh musuh		
		2. Nilai bertambah 10 poin
		3. Update informasi nilai
4. Mengambil bahan masakan		
		5. Nilai bertambah 20 poin
		6. Update informasi nilai
Skenario Alternatif	-	
Kondisi Akhir	Sistem menampilkan informasi nilai	

7. Skenario *Use Case* Memainkan tingkat selanjutnya

Tabel III.11 Skenario *Use Case* Memainkan tingkat selanjutnya

Identifikasi		
Nomor	7	
Nama	Memainkan tingkat selanjutnya	
Tujuan	Pemain melanjutkan permainan pada tingkat selanjutnya	
Deskripsi	Proses untuk melanjutkan permainan pada tingkat selanjutnya	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di arena permainan	
Aksi aktor		Reaksi sistem
1. Mengambil bahan masakan		
		2. Apabila sisa bahan masakan sama dengan 0 maka pintu akan terbuka dan dapat melanjutkan ke level berikutnya
Skenario Alternatif		-
Kondisi Akhir	Sistem menampilkan pintu terbuka	

8. Skenario *Use Case* Menghitung bahan halus

Tabel III.12 Skenario *Use Case* Menghitung bahan halus

Identifikasi		
Nomor	8	
Nama	Menghitung bahan halus	
Tujuan	Menghitung bahan halus	
Deskripsi	Proses untuk menghitung bahan halus	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di arena permainan masak	
Aksi aktor		Reaksi sistem
1. Pemain memasukan bahan untuk bahan halus ke dalam blender		
		2. Update persenan proses bahan halus
Kondisi Akhir	Sistem menampilkan informasi persenan bahan halus	

9. Skenario *Use Case* Menghitung bahan aduk

Tabel III.13 *Use Case* Menghitung bahan aduk

Identifikasi		
Nomor	9	
Nama	Menghitung bahan aduk	
Tujuan	Menghitung bahan aduk	
Deskripsi	Proses untuk menghitung bahan goreng	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di arena permainan masak	
Aksi aktor		Reaksi sistem
1. Pemain memasukan bahan untuk bahan aduk ke dalam mangkok besar		
		2. Update persenan proses bahan aduk
Kondisi Akhir	Sistem menampilkan informasi persenan bahan halus	

10. Skenario *Use Case* Menghitung bahan potong

Tabel III.14 Skenario *Use Case* Menghitung bahan potong

Identifikasi	
Nomor	8
Nama	Menghitung bahan potong
Tujuan	Menghitung bahan potong
Deskripsi	Proses untuk menghitung bahan potong
Aktor	Pemain

Skenario Utama		
Kondisi awal	Pemain berada di arena permainan masak	
Aksi aktor		Reaksi sistem
1. Pemain memasukan bahan untuk bahan potong ke dalam papan potong		
		2. Update persenan proses bahan potong
Kondisi Akhir	Sistem menampilkan informasi bahan potong	

11. Skenario *Use Case* Menghitung bahan jadi

Tabel III.15 Skenario *Use Case* Menghitung bahan jadi

Identifikasi		
Nomor	8	
Nama	Menghitung bahan potong	
Tujuan	Menghitung bahan potong	
Deskripsi	Proses untuk menghitung bahan potong	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di arena permainan masak	
Aksi aktor		Reaksi sistem
1. Pemain memasukan bahan untuk bahan jadi ke dalam mangkok besar untuk masakan nasi tutug oncom dan masukan ke dalam wajan untuk masakan ayam santan dan mireng udang		
		2. Update persenan proses bahan jadi
Kondisi Akhir	Sistem menampilkan informasi bahan jadi	

12. Skenario *Use Case* Menghitung waktu masak

Tabel III.16 Skenario *Use Case* Menghitung waktu masak

Identifikasi		
Nomor	8	
Nama	Menghitung bahan potong	
Tujuan	Menghitung bahan potong	
Deskripsi	Proses untuk menghitung bahan potong	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di arena permainan masak	
Aksi aktor		Reaksi sistem
1. Pemain menekan tombol mulai masak pada layar permainan		

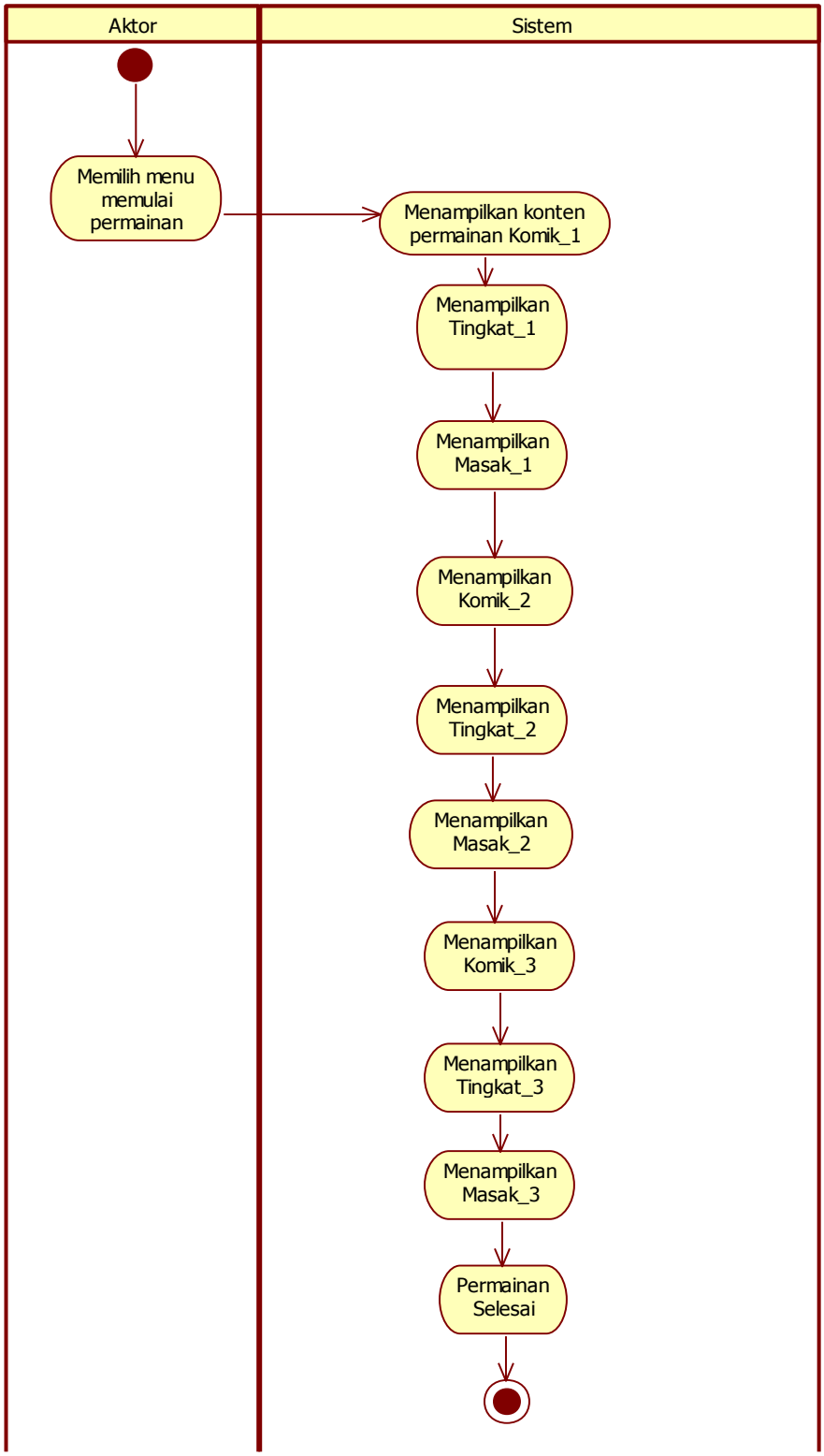
	2. Menampilkan informasi waktu masak berkurang
3. Pemain menekan tombol selesai masak pada layar permainan	
	4. Menampilkan objek bahan masakan yang telah di proses
Skenario Alternatif	
1. Pemain menekan tombol waktu masak pada layar permainan	
	2. Menampilkan informasi waktu masak berkurang
	3. Waktu masak sama dengan 0
	4. Menampilkan permainan berakhir
Kondisi Akhir	Sistem menampilkan informasi waktu masak

III.1.4.5 Activity Diagram

Diagram aktifitas atau *activity diagram* menggambarkan aliran kerja atau aktifitas dari sebuah sistem. Diagram aktifitas berisi penjelasan lebih lengkap mengenai urutan suatu proses yang terjadi antara pelaku dan sistem. Aktifitas yang selalu dimulai dari pelaku, kemudian akan mendapat respon dari sistem yang anntinya akan menampilkan akhir dari proses tersebut.

1. Diagram Aktifitas Memulai permainan

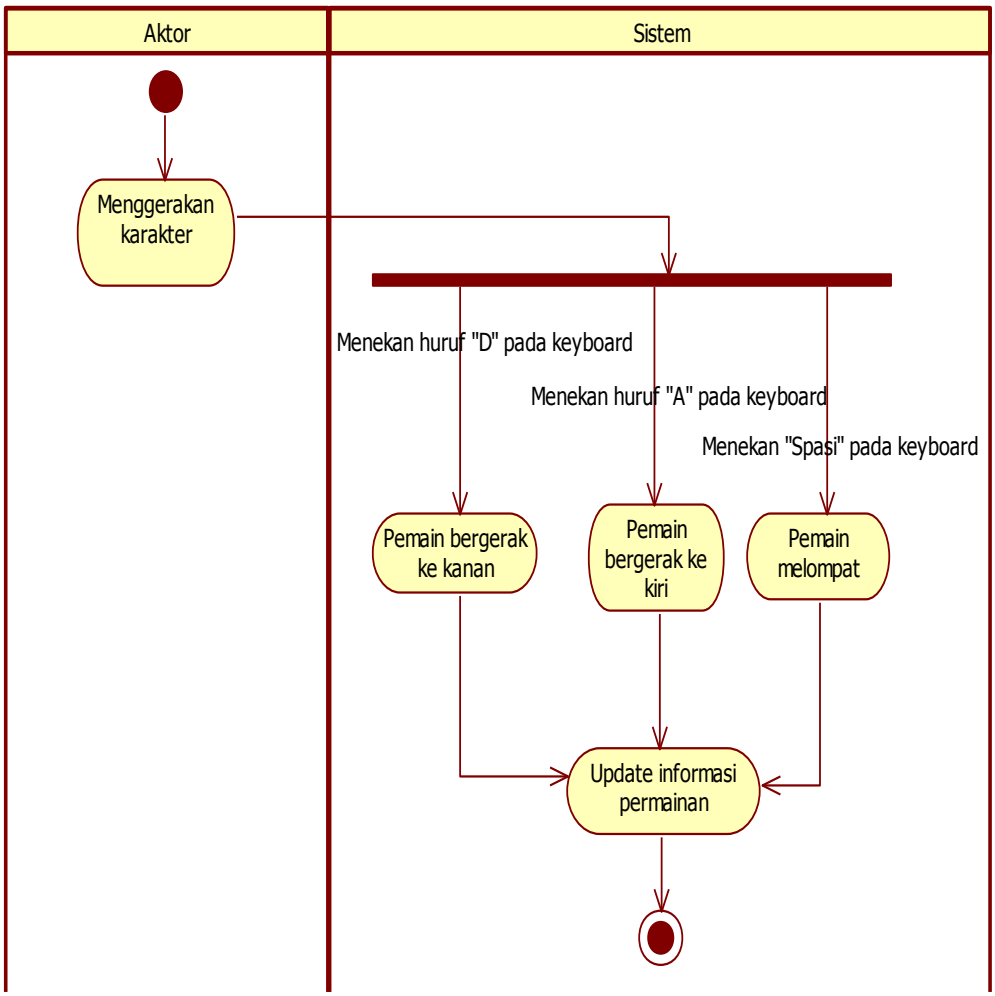
Proses ini terjadi saat pemain memilih menu mulai pada halaman utama permainan Selamatkan Masakan Nusantara Indonesia. Sistem akan menampilkan Komik_1.



Gambar III.26 Diagram Aktifitas Memulai permainan

2. Diagram Aktifitas Menggerakan karakter

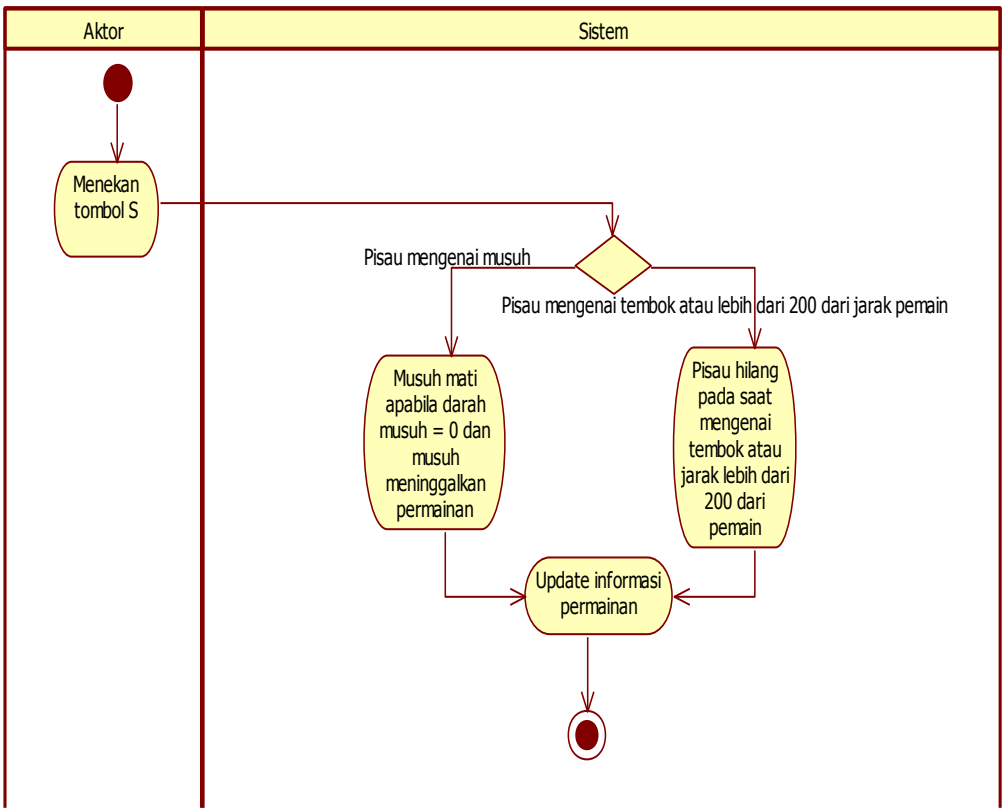
Proses ini terjadi ketika pemain menekan tombol A, D, Spasi pada *keyboard* untuk mengendalikan karakter pada arena permainan.



Gambar III.27 Diagram Aktifitas Menggerakkan karakter

3. Diagram Aktifitas Melempar pisau

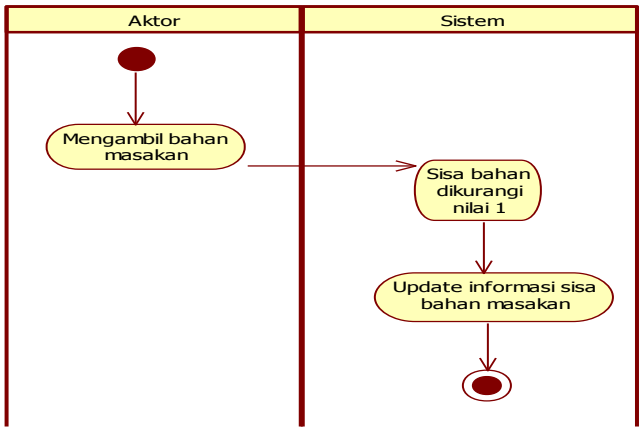
Proses ini terjadi ketika pemain menekal tombol S pada *keyboard* untuk melempar pisau pada musuh.



Gambar III.28 Diagram Melempar pisau

4. Diagram Aktifitas Menghitung sisa bahan

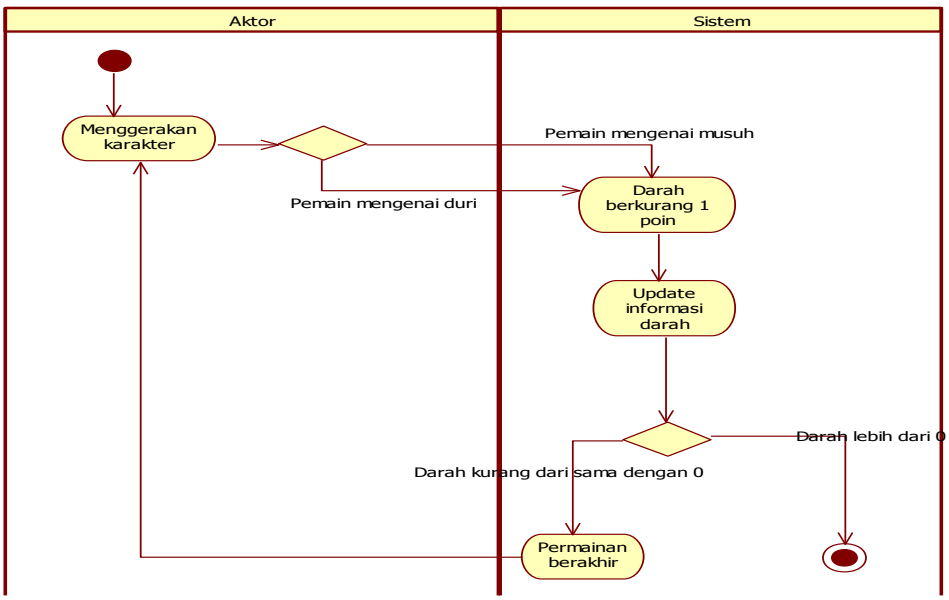
Proses ini akan memberikan informasi kepada pemain tentang sisa bahan masakan pada arena permainan.



Gambar III.29 Diagram Aktifitas Menghitung sisa bahan

5. Diagram Aktifitas Menghitung darah

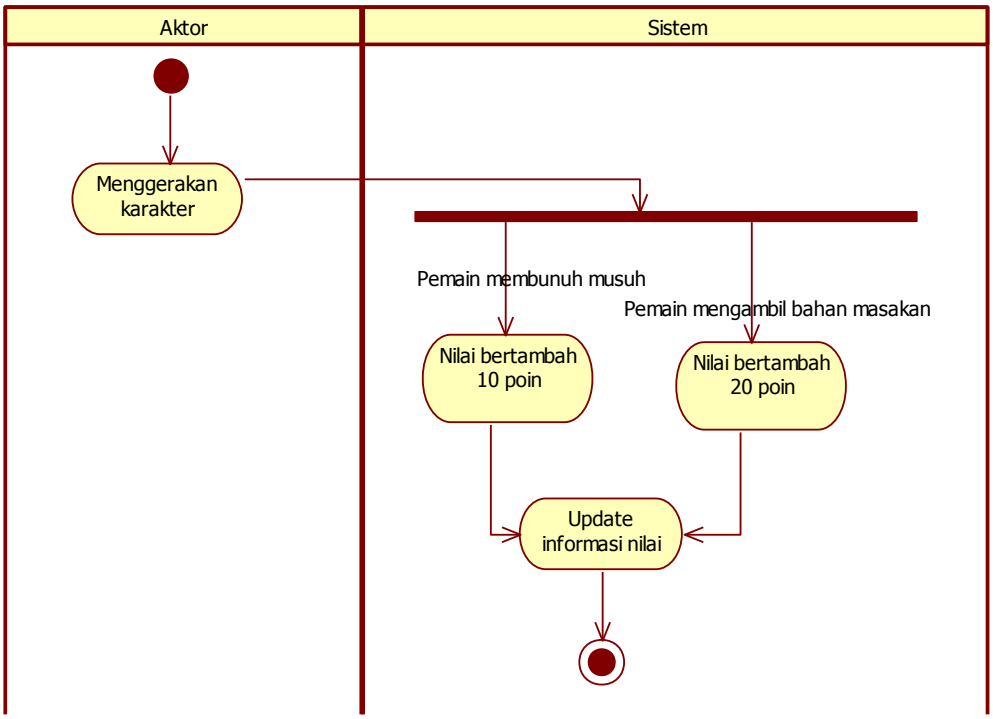
Proses ini akan memberikan informasi kepada pemain tentang sisa darah pemain pada arena permainan.



Gambar III.30 Diagram Aktifitas Menghitung darah

6. Diagram Aktifitas Menghitung Nilai

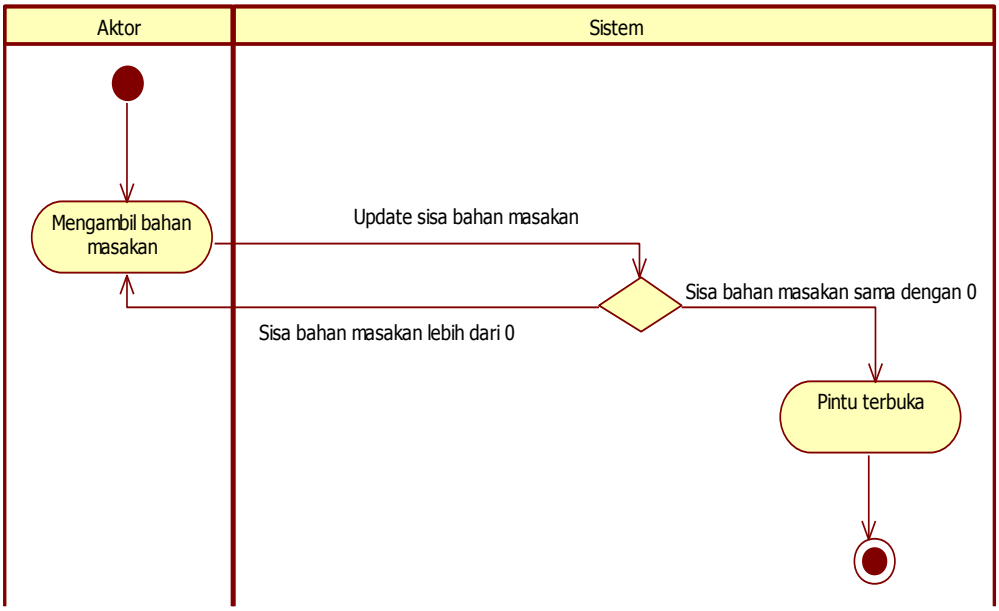
Proses ini akan memberikan informasi kepada pemain tentang nilai yang telah diperoleh pada arena permainan.



Gambar III.31 Diagram Aktifitas Menghitung nilai

7. Diagram Aktifitas Melanjutkan tingkat selanjutnya

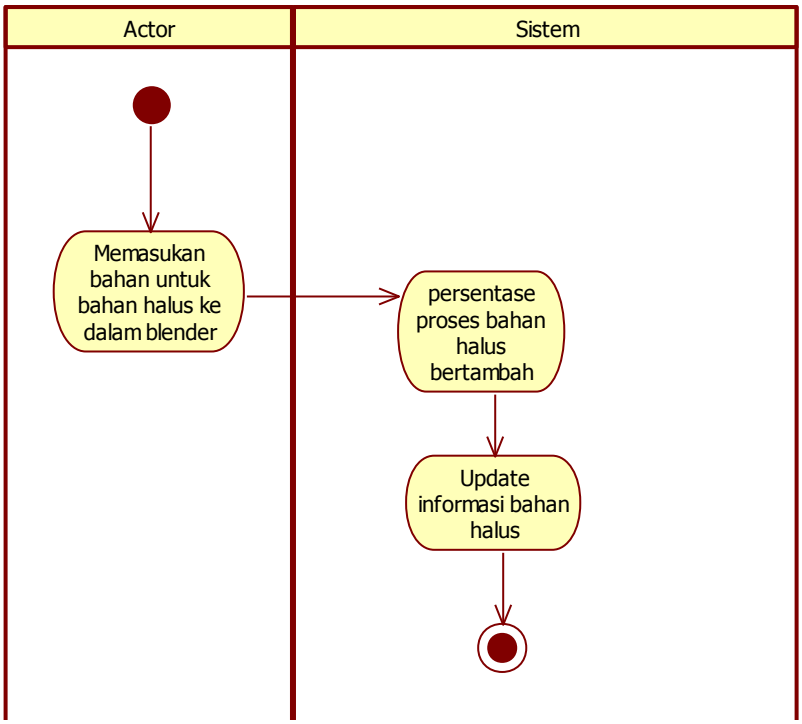
Proses pintu terbuka terjadi apabila pemain telah mengambil semua bahan masakan yang terdapat pada arena permainan dan dapat melanjutkan ke tingkat selanjutnya.



Gambar III.32 Diagram Aktifitas Melanjutkan tingkat selanjutnya

8. Diagram Aktifitas Menghitung bahan halus

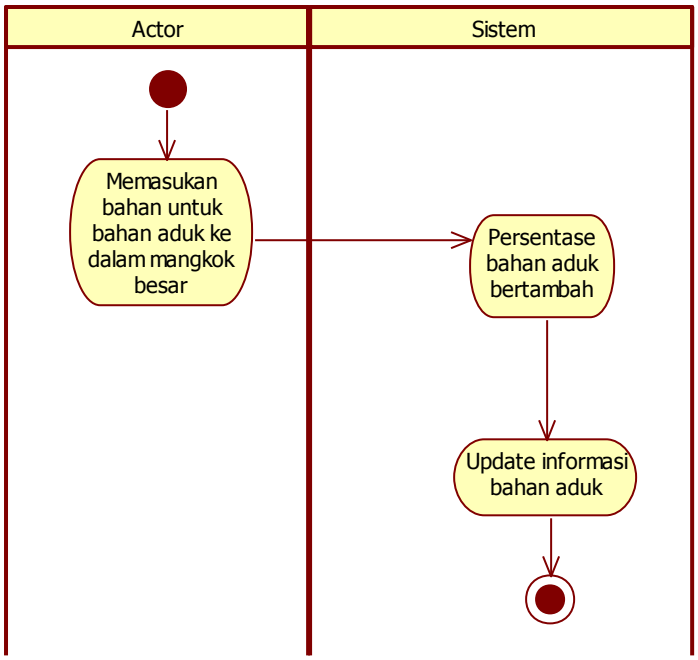
Proses menghitung bahan halus terjadi apabila pemain berada pada tingkat masak dan proses menghitung bahan halus membutuhkan bahan untuk bahan halus.



Gambar III.33 Diagram Aktifitas Menghitung bahan halus

9. Diagram Aktifitas Menghitung bahan aduk

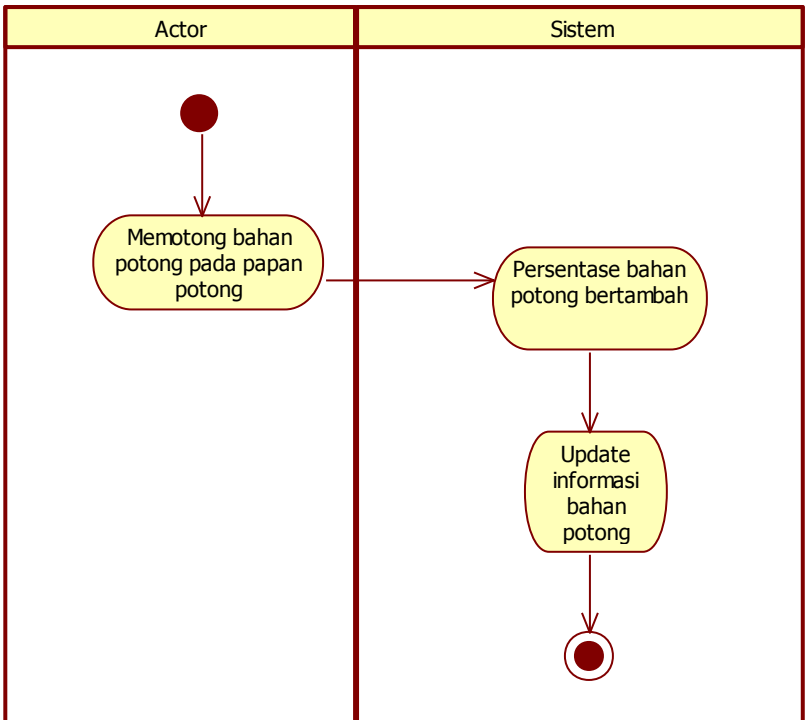
Proses menghitung bahan aduk terjadi apabila pemain berada pada tingkat masak dan proses menghitung bahan aduk membutuhkan bahan untuk bahan aduk.



Gambar III.34 Diagram Aktifitas Menghitung bahan aduk

10. Diagram Aktifitas Menghitung bahan potong

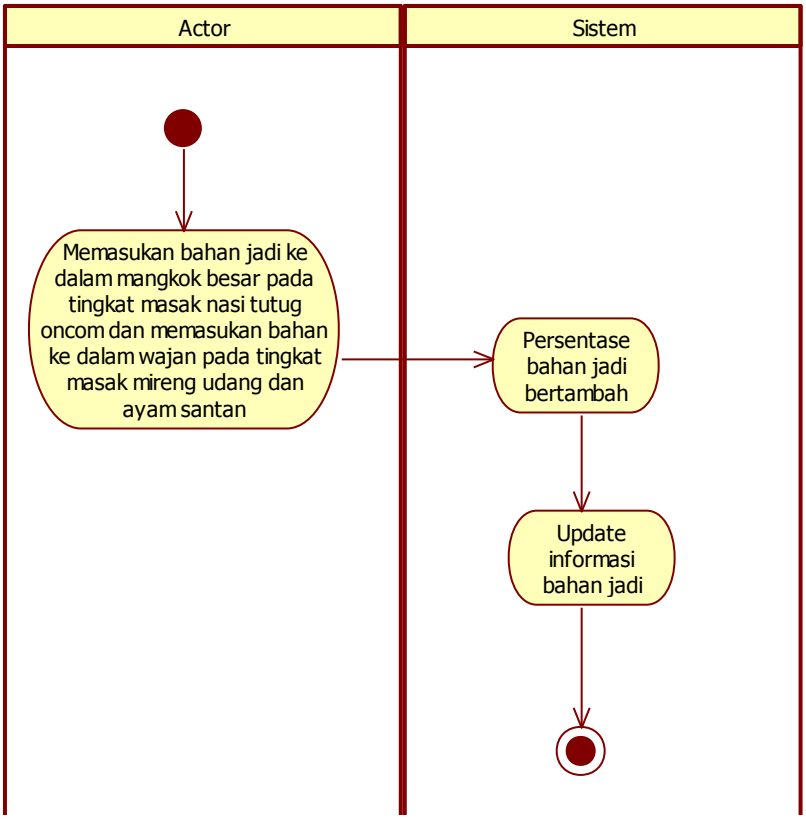
Proses menghitung bahan potong terjadi apabila pemain berada pada tingkat masak dan proses menghitung bahan potong membutuhkan bahan untuk bahan potong.



Gambar III.35 Diagram Aktifitas Menghitung bahan potong

11. Diagram Aktifitas Menghitung bahan jadi

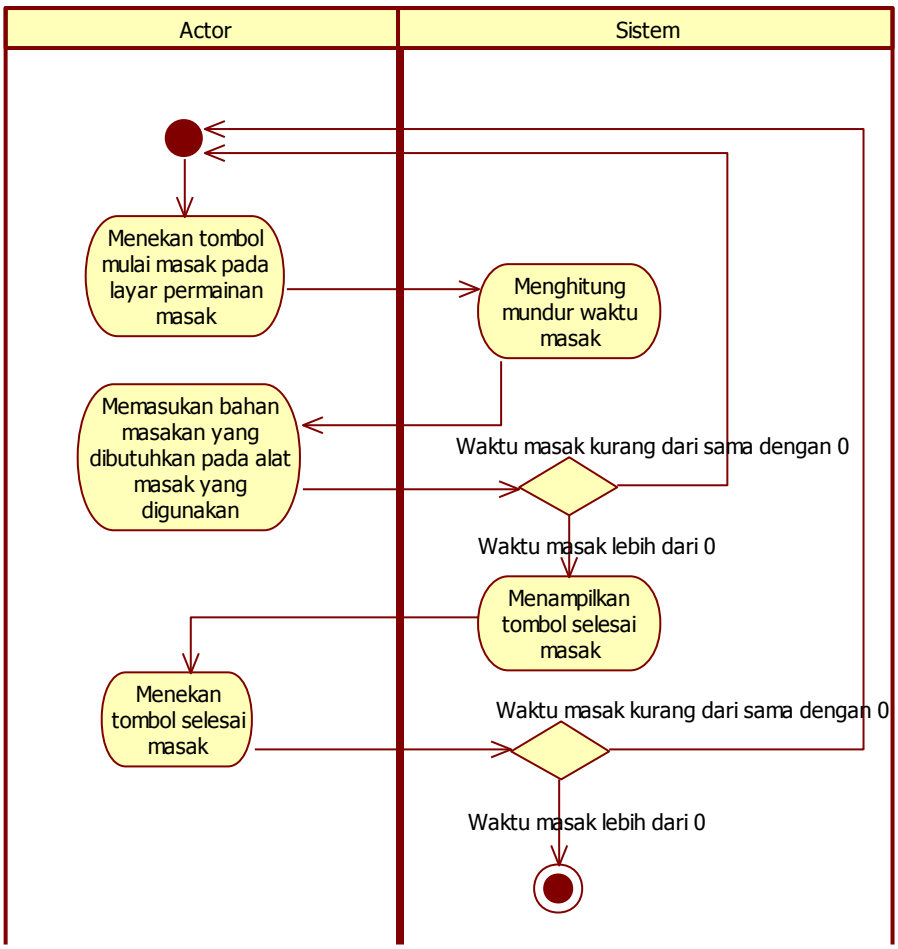
Proses menghitung bahan jadi terjadi apabila pemain berada pada tingkat masak dan proses menghitung bahan jadi membutuhkan bahan untuk bahan jadi.



Gambar III.36 Diagram Aktifitas Menghitung bahan jadi

12. Diagram Aktifitas Menghitung waktu masak

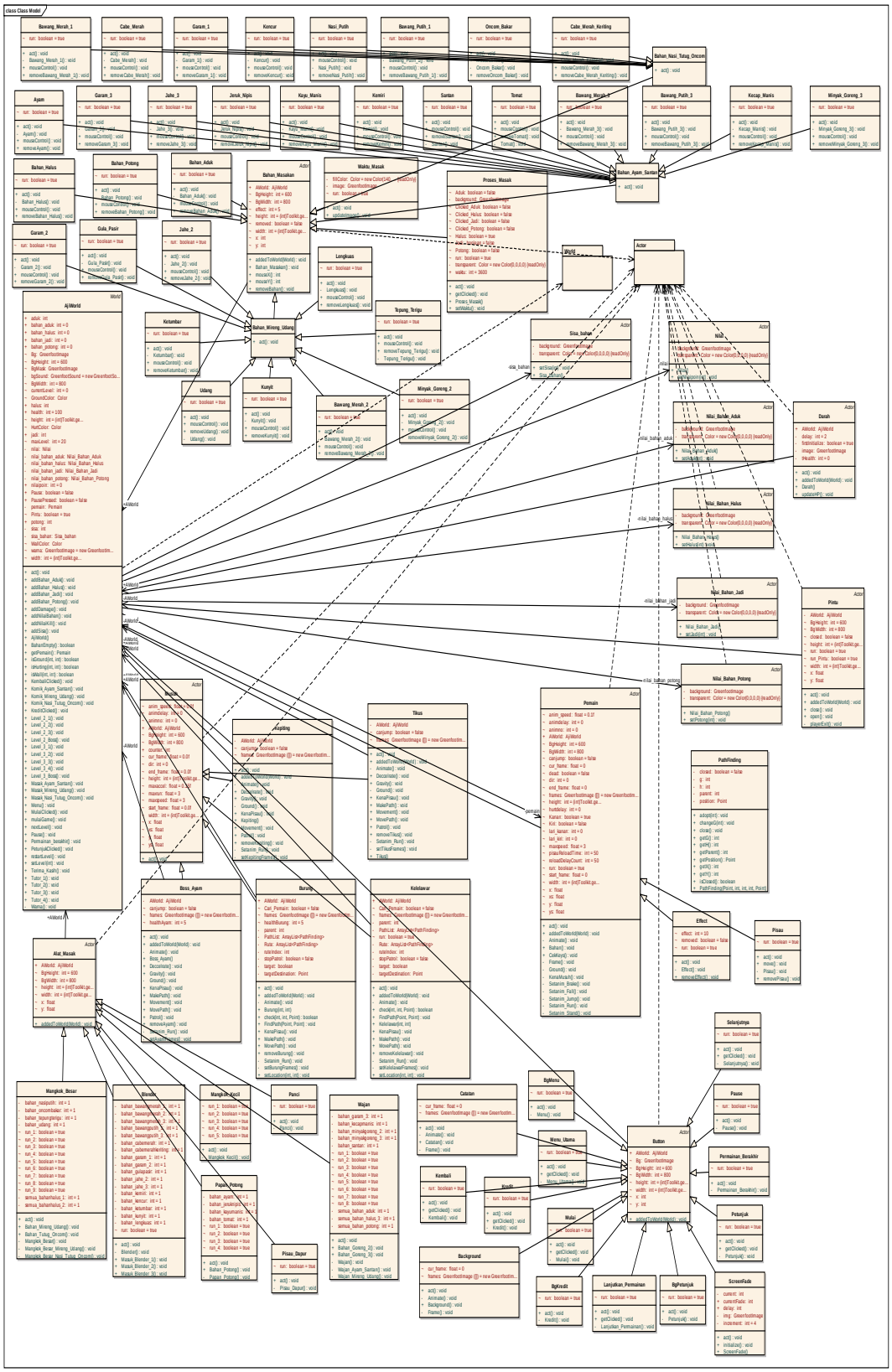
Proses menghitung waktu masak terjadi apabila pemain berada pada tingkat masak.



Gambar III.37 Diagram Aktifitas Menghitung waktu masak

III.1.4.6 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.



Gambar III.38 Kelas Diagram

Penjelasan mengenai kelas-kelas yang terdapat pada kelas diagram akan dijelaskan pada tabel III.14.

Tabel III.17 Definisi *Class* Diagram

Nama *Class*	Jenis *Class*	Deskripsi
World	*Boundary*	Metode world tersedia untuk world.
Actor	*Boundary*	Metode actor tersedia menunkan semua metode actor.
AjiWorld	*Boundary*	Kelas yang mengatur interaksi objek-objek yang terlibat ketika permainan berlangsung.
Pemain	*Control*	Kelas yang bertanggung jawab untuk menggerakkan karakter sesuai interaksi yang dilakukan oleh pengguna.
Effect	*Entity*	Spesialisai dari kelas Pemain yang bertanggung jawab untuk memunculkan objek effect apabila pemain mengenai ranjau atau mengenai kelas musuh dan spesialisasi kelas musuh.
Pisau	*Control*	Spesialisai dari kelas Pemain yang bertanggung jawab untuk memunculkan objek pisau sesuai interaksi yang dilakukan oleh pengguna.
Musuh	*Control*	Kelas yang bertanggung jawab untuk mengelola bertabrakan antara objek dari spesialisasi kelas Musuh dengan kelas Pemain dan spesialisasi kelas Pemain.
Tikus	*Control*	Spesialisasi dari kelas Musuh yang bertanggung jawab untuk menggerakan objek Tikus.
Kelelawar	*Control*	Spesialisasi dari kelas Musuh yang bertanggung

Nama *Class*	Jenis *Class*	Deskripsi
		jawab untuk menggerakan objek Kelelawar.
Burung	*Control*	Spesialisasi dari kelas Musuh yang bertanggung jawab untuk menggerakan objek Burung.
Kepiting	*Control*	Spesialisasi dari kelas Musuh yang bertanggung jawab untuk menggerakan objek Kepiting.
Boss_Ayam	*Control*	Spesialisasi dari kelas Musuh yang bertanggung jawab untuk menggerakan objek Boss_Ayam.
Darah	*Entity*	Kelas yang bertanggung jawab dalam menghitung jumlah darah pada kelas Pemain apabila terjadi bertabrakan antara objek spesialisai kelas Musuh.
Nilai	*Entity*	Kelas yang bertanggung jawab dalam menghitung angka yang diperoleh kelas Pemain.
Pintu	*Entity*	Kelas yang bertanggung jawab untuk memproses permainan ke tingkat berikutnya.
Button	*Control*	Kelas yang bertanggung jawab sebagai interaksi terhadap pengguna pada tampilan menu.
Mulai	*Control*	Spesialisasi dari kelas Button yang bertanggung jawab dalam memproses ke tampilan permainan.
Petunjuk	*Control*	Spesialisasi dari kelas Button yang bertanggung jawab dalam memproses ke tampilan petunjuk permainan.
Kredit	*Control*	Spesialisasi dari kelas Button yang bertanggung jawab dalam memproses ke tampilan kredit.

Nama *Class*	Jenis *Class*	Deskripsi
Kembali	*Control*	Spesialisasi dari kelas Button yang bertanggung jawab dalam memproses kembali pada tampilan menu.
Lanjutkan permainan	*Control*	Spesialisasi dari kelas Button yang bertanggung jawab dalam memproses ke tampilan arena permainan.
Selanjutnya	*Control*	Spesialisasi dari kelas Button yang bertanggung jawab dalam memproses ke tampilan arena permainan.
Bg_Menu	*Boundary*	Spesialisasi dari kelas Button yang bertanggung jawab dalam menampilkan background menu utama pada permainan.
Bg_Petunjuk	*Boundary*	Spesialisasi dari kelas Button yang bertanggung jawab dalam menampilkan background petunjuk pada permainan.
Bg_Kredit	*Boundary*	Spesialisasi dari kelas Button yang bertanggung jawab dalam menampilkan background kredit dalam permainan.
Catatan	*Boundary*	Spesialisasi dari kelas Button yang bertanggung jawab dalam menampilkan catatan pada tingkat masak.
Background	*Boundary*	Spesialisasi dari kelas Button yang bertanggung jawab dalam menampilkan setiap background pada tingkat permainan.
Screenfade	*Boundary*	Spesialisasi dari kelas Button yang bertanggung jawab untuk menampilkan background gelap pada setiap awal tingkat.
Permainan_Berakhir	*Boundary*	Spesialisasi dari kelas Button yang bertanggung jawab untuk menampilkan

Nama *Class*	Jenis *Class*	Deskripsi
		background apbila permainan berakhir.
Pause	*Control*	Spesialisasi dari kelas Button yang bertanggung jawab untuk background apbila pemain menekan "Escape" pada keyboard pada permainan.
Bahan_Masakan	*Control*	Kelas yang bertanggung jawab untuk memproses apabila terjadi bertabrakan antara objek kelas Pemain dan spesialisasi dari kelas Bahan_Masakan.
Bahan_Nasi_Tutug_Oncom	*Boundary*	Spesialisasi dari kelas Bahan_Masakan.
Bahan_Mireng_Udang	*Boundary*	Spesialisasi dari kelas Bahan_Masakan.
Bahan_Ayam_Santan	*Boundary*	Spesialisasi dari kelas Bahan_Masakan.
Bawang_Merah_1	*Boundary*	Spesialisasi dari kelas Bahan_Nasi_Tutug_Oncom yang menampilkan objek Bawang_Merah_1 pada arena permainan.
Kencur	*Boundary*	Spesialisasi dari kelas Bahan_Nasi_Tutug_Oncom yang menampilkan objek Kencur pada arena permainan.
Nasi_Putih	*Boundary*	Spesialisasi dari kelas Bahan_Nasi_Tutug_Oncom yang menampilkan objek Nasi_Putih pada arena permainan.
Garam_1	*Boundary*	Spesialisasi dari kelas Bahan_Nasi_Tutug_Oncom yang menampilkan objek Garam_1 pada arena permainan.
Oncom_Bakar	*Boundary*	Spesialisasi dari kelas Bahan_Nasi_Tutug_Oncom yang menampilkan objek Oncom_Bakar pada arena permainan.

Nama *Class*	Jenis *Class*	Deskripsi
Cabe_Merah	*Boundary*	Spesialisasi dari kelas Bahan_Nasi_Tutug_Onco m yang menampilkan objek Cabe_Merah pada arena permainan.
Cabe_Merah_Keriting	*Boundary*	Spesialisasi dari kelas Bahan_Nasi_Tutug_Onco m yang menampilkan objek Cabe_Merah_Keriting pada arena permainan.
Bawang_Putih_1	*Boundary*	Spesialisasi dari kelas Bahan_Nasi_Tutug_Onco m yang menampilkan objek Bawang_Putih_1 pada arena permainan.
Ketumbar	*Boundary*	Spesialisasi dari kelas Bahan_Mireng_Udang yang menampilkan objek Ketumbar pada arena permainan.
Gula_Pasir	*Boundary*	Spesialisasi dari kelas Bahan_Mireng_Udang yang menampilkan objek Gula_Pasir pada arena permainan.
Bawang_Merah_2	*Boundary*	Spesialisasi dari kelas Bahan_Mireng_Udang yang menampilkan objek Bawang_Merah_2 pada arena permainan.
Minyak_Goreng_2	*Boundary*	Spesialisasi dari kelas Bahan_Mireng_Udang yang menampilkan objek Minyak_Goreng_2 pada arena permainan.
Jahe_2	*Boundary*	Spesialisasi dari kelas Bahan_Mireng_Udang yang menampilkan objek Jahe_2 pada arena permainan.
Udang	*Boundary*	Spesialisasi dari kelas Bahan_Mireng_Udang yang menampilkan objek Udang pada arena permainan.

Nama *Class*	Jenis *Class*	Deskripsi
Garam_2	*Boundary*	Spesialisasi dari kelas Bahan_Mireng_Udang yang menampilkan objek Garam_2 pada arena permainan.
Tepung_Terigu	*Boundary*	Spesialisasi dari kelas Bahan_Mireng_Udang yang menampilkan objek Tepung_Terigu pada arena permainan.
Kunyit	*Boundary*	Spesialisasi dari kelas Bahan_Mireng_Udang yang menampilkan objek Kunyit pada arena permainan.
Lengkuas	*Boundary*	Spesialisasi dari kelas Bahan_Mireng_Udang yang menampilkan objek Lengkuas pada arena permainan.
Ayam	*Boundary*	Spesialisasi dari kelas Bahan_Ayam_Santan yang menampilkan objek Ayam pada arena permainan.
Bawang_Merah_3	*Boundary*	Spesialisasi dari kelas Bahan_Ayam_Santan yang menampilkan objek Bawang_Merah_3 pada arena permainan.
Bawang_Putih_3	*Boundary*	Spesialisasi dari kelas Bahan_Ayam_Santan yang menampilkan objek Bawang_Putih_3 pada arena permainan.
Garam_3	*Boundary*	Spesialisasi dari kelas Bahan_Ayam_Santan yang menampilkan objek Garam_3 pada arena permainan.
Jahe_3	*Boundary*	Spesialisasi dari kelas Bahan_Ayam_Santan yang menampilkan objek Jahe_3 pada arena permainan.
Jeruk_Nipis	*Boundary*	Spesialisasi dari kelas Bahan_Ayam_Santan yang

Nama *Class*	Jenis *Class*	Deskripsi
		menampilkan objek Jeruk_Nipis pada arena permainan.
Kayu_Manis	*Boundary*	Spesialisasi dari kelas Bahan_Ayam_Santan yang menampilkan objek Kayu_Manis pada arena permainan.
Kemiri	*Boundary*	Spesialisasi dari kelas Bahan_Ayam_Santan yang menampilkan objek Kemiri pada arena permainan.
Minyak_Goreng_3	*Boundary*	Spesialisasi dari kelas Bahan_Ayam_Santan yang menampilkan objek Minyak_Goreng_3 pada arena permainan.
Tomat	*Boundary*	Spesialisasi dari kelas Bahan_Ayam_Santan yang menampilkan objek Tomat pada arena permainan.
Santan	*Boundary*	Spesialisasi dari kelas Bahan_Ayam_Santan yang menampilkan objek Santan pada arena permainan.
Kecap_Manis	*Boundary*	Spesialisasi dari kelas Bahan_Ayam_Santan yang menampilkan objek Kecap_Manis pada arena permainan.
Bahan_Halus	*Boundary*	Spesialisasi dari kelas Bahan_Masakan yang menampilkan objek Bahan_Halus pada arena permainan.
Bahan_Potong	*Boundary*	Spesialisasi dari kelas Bahan_Masakan yang menampilkan objek Bahan_Potong pada arena permainan.
Bahan_Aduk	*Boundary*	Spesialisasi dari kelas Bahan_Masakan yang menampilkan objek Bahan_Aduk pada arena permainan.

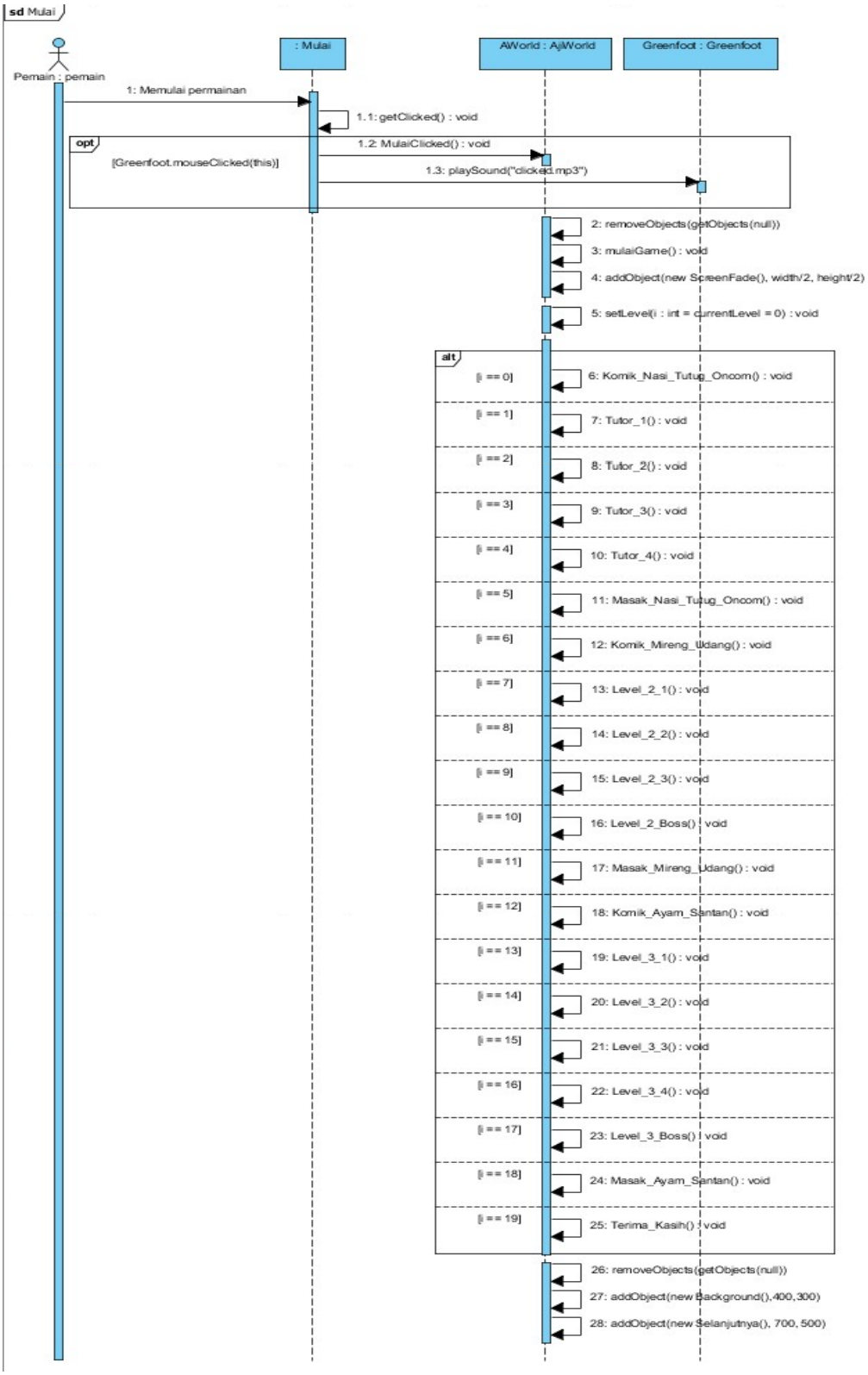
Nama *Class*	Jenis *Class*	Deskripsi
Proses_Masak	*Entity*	Kelas yang bertanggung jawab dalam memproses masak pada tingkat masak.
Waktu_Masak	*Entity*	Kelas yang bertanggung jawab dalam menampilkan waktu masak pada tingkat masak.
Nilai_Bahan_Halus	*Entity*	Kelas yang bertanggung jawab dalam menghitung persentase bahan halus yang diperoleh pada tingkat masak.
Nilai_Bahan_Potong	*Entity*	Kelas yang bertanggung jawab dalam menghitung persentase bahan potong yang diperoleh pada tingkat masak.
Nilai_Bahan_Aduk	*Entity*	Kelas yang bertanggung jawab dalam menghitung persentase bahan aduk yang diperoleh pada tingkat masak.
Nilai_Bahan_Jadi	*Entity*	Kelas yang bertanggung jawab dalam menghitung persentase bahan jadi yang diperoleh pada tingkat masak.
Alat_Masak	*Control*	Kelas yang bertanggung jawab untuk menghubungkan spesialisasi kelas Alat_Masak dengan kelas AjiWorld.
Blender	*Control*	Kelas yang bertanggung jawab untuk memproses apabila terjadi bertabrakan antara objek kelas Bawang_merah, kelas Bawang_putih, kelas Kemiri, kelas Jahe dan merupakan spesialisasi dari kelas Alat_Masak.
Papan_Potong	*Control*	Kelas yang bertanggung jawab untuk memproses apabila terjadi bertabrakan

Nama *Class*	Jenis *Class*	Deskripsi
		antara objek kelas Ayam, kelas Tomat, kelas Kayu_manis, kelas Jeruk_nipis dan merupakan spesialisasi dari kelas Alat_Masak.
Wajan	*Control*	Kelas yang bertanggung jawab untuk memproses apabila terjadi bertabrakan antara objek kelas Bahan_halus, kelas Bahan_potong, kelas Minyak_kelapa, kelas Santan, kelas Garam, kelas Kecap_manis dan merupakan spesialisasi dari kelas Alat_Masak.
Panci	*Control*	Merupakan spesialisasi dari kelas Alat_Masak.
Mangkok_Besar	*Control*	Merupakan spesialisasi dari kelas Alat_Masak.
Mangkok_Kecil	*Control*	Merupakan spesialisasi dari kelas Alat_Masak.
Pisau_Dapur	*Control*	Merupakan spesialisasi dari kelas Alat_Masak.
PathFinding	*Boundary*	Kelas tambahan untuk membantu inisialisasi pada kelas Kelelawar dan kelas Burung.

III.1.4.7 Sequence Diagram

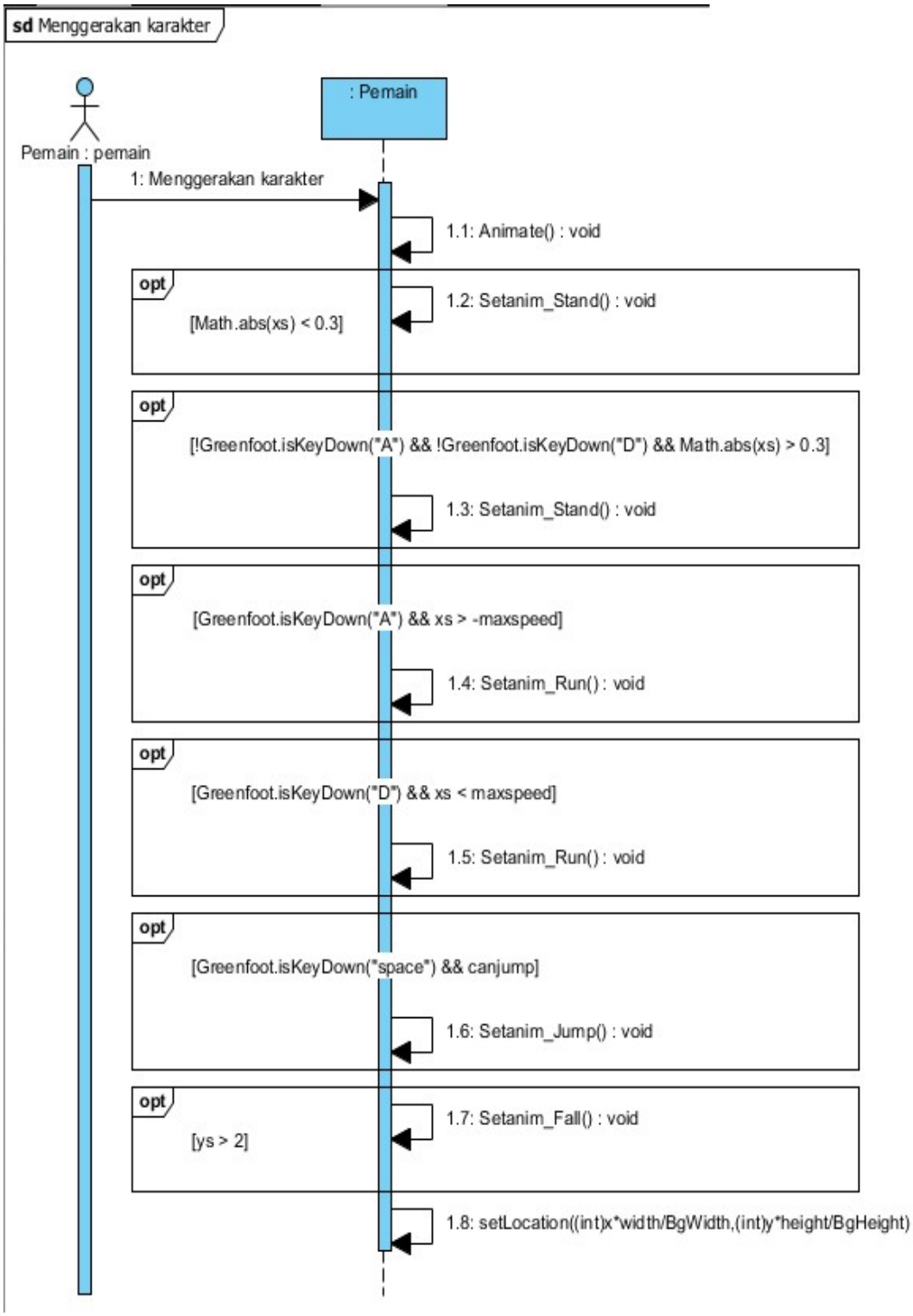
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

1. Diagram Sekuen Memulai permainan



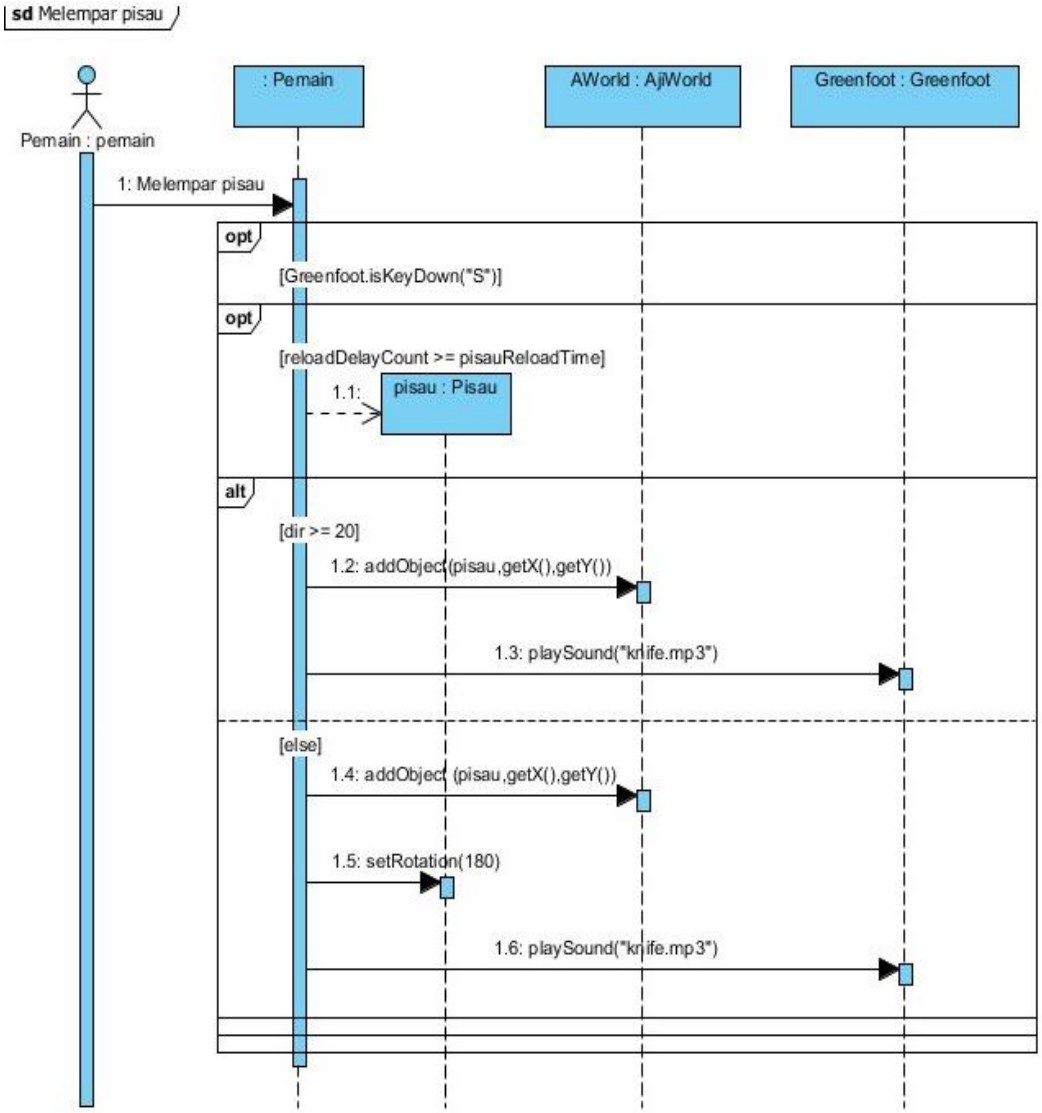
Gambar III.39 Diagram Sekuen memulai permainan

2. Diagram Sekuen Menggerakan karakter



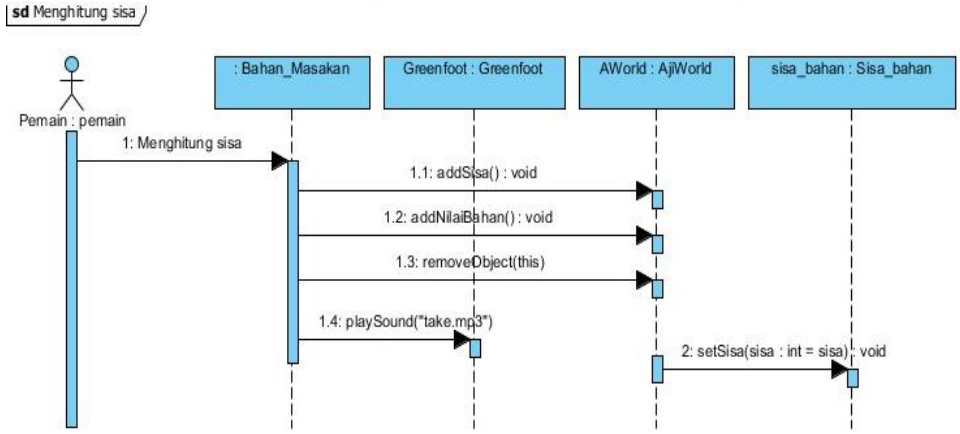
Gambar III.40 Diagram Sekuen Menggerakan karakter

3. Diagram Sekuen Melempar pisau



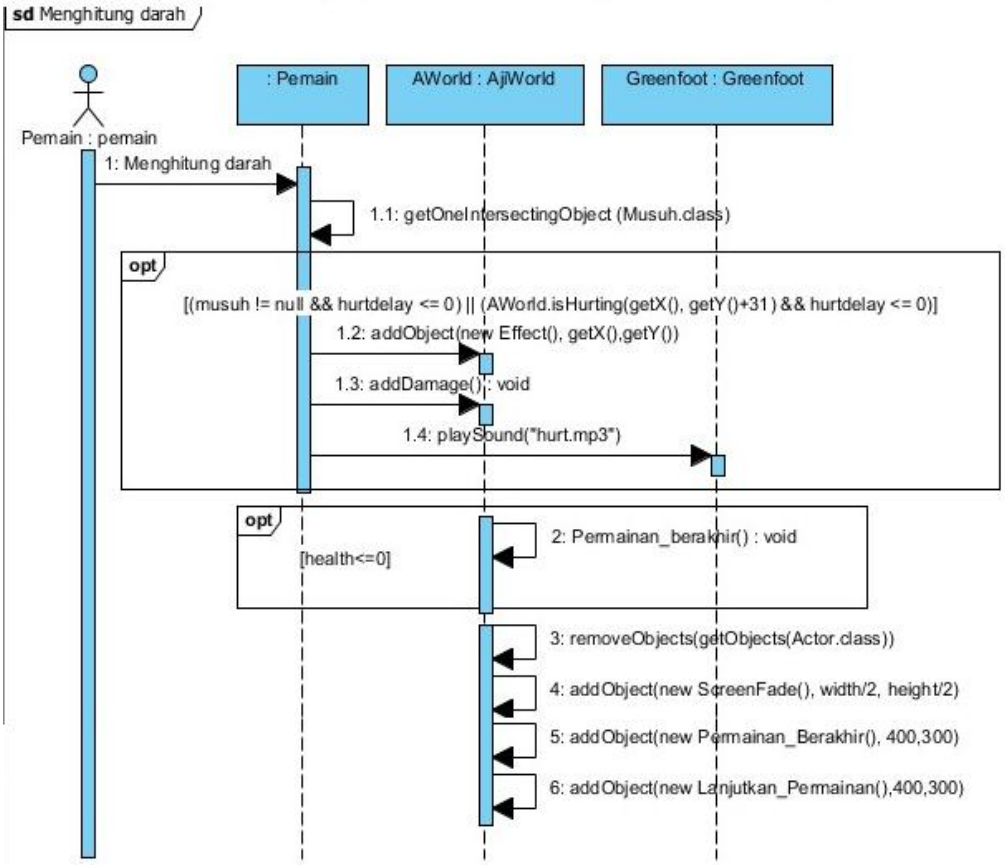
Gambar III.41 Diagram Sekuen Melempar pisau

4. Diagram Sekuen Menghitung sisa bahan



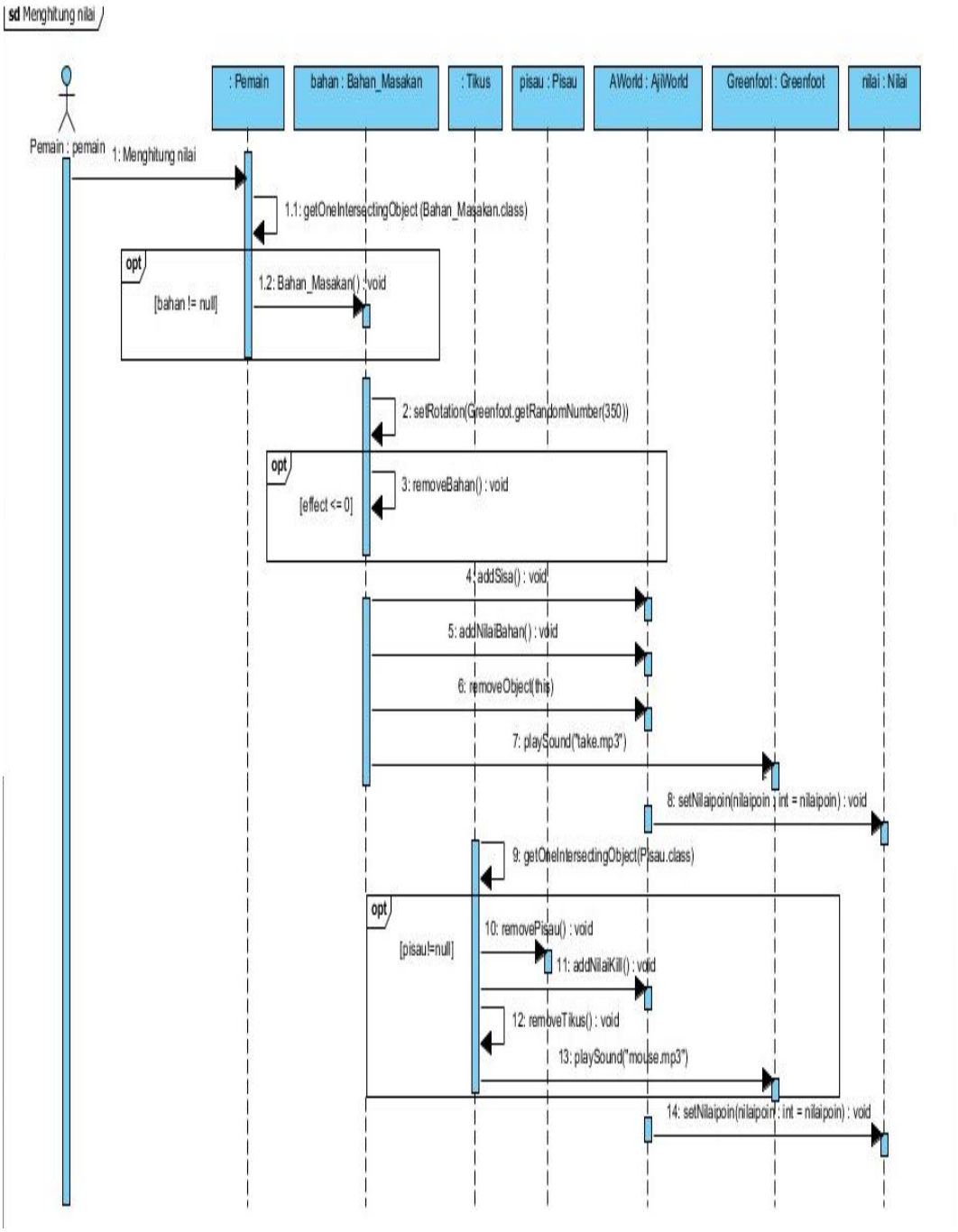
Gambar III.42 Diagram Sekuen Menghitung sisa bahan

5. Diagram Sekuen Menghitung darah



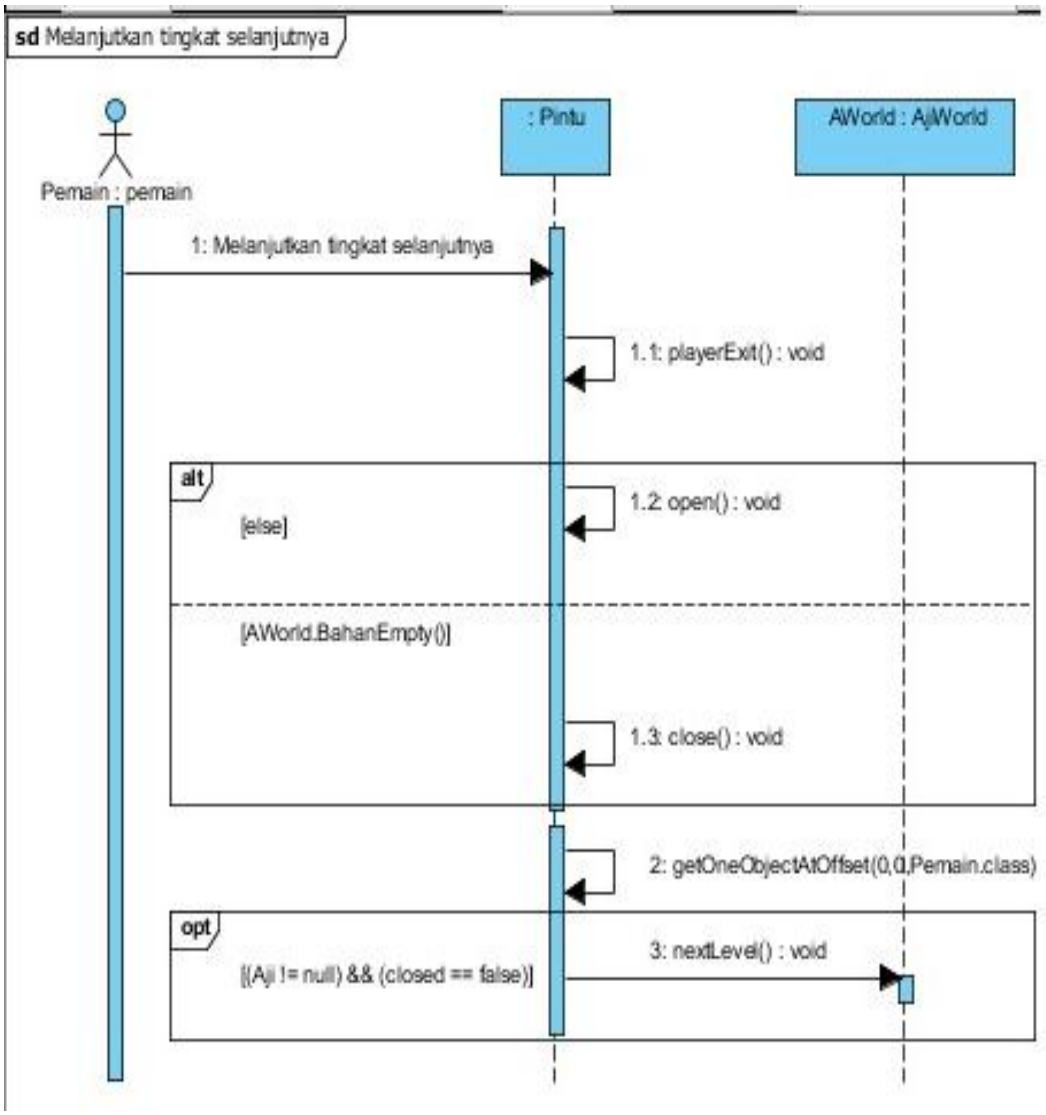
Gambar III.43 Diagram Sekuen Menghitung darah

6. Diagram Sekuen Menghitung nilai



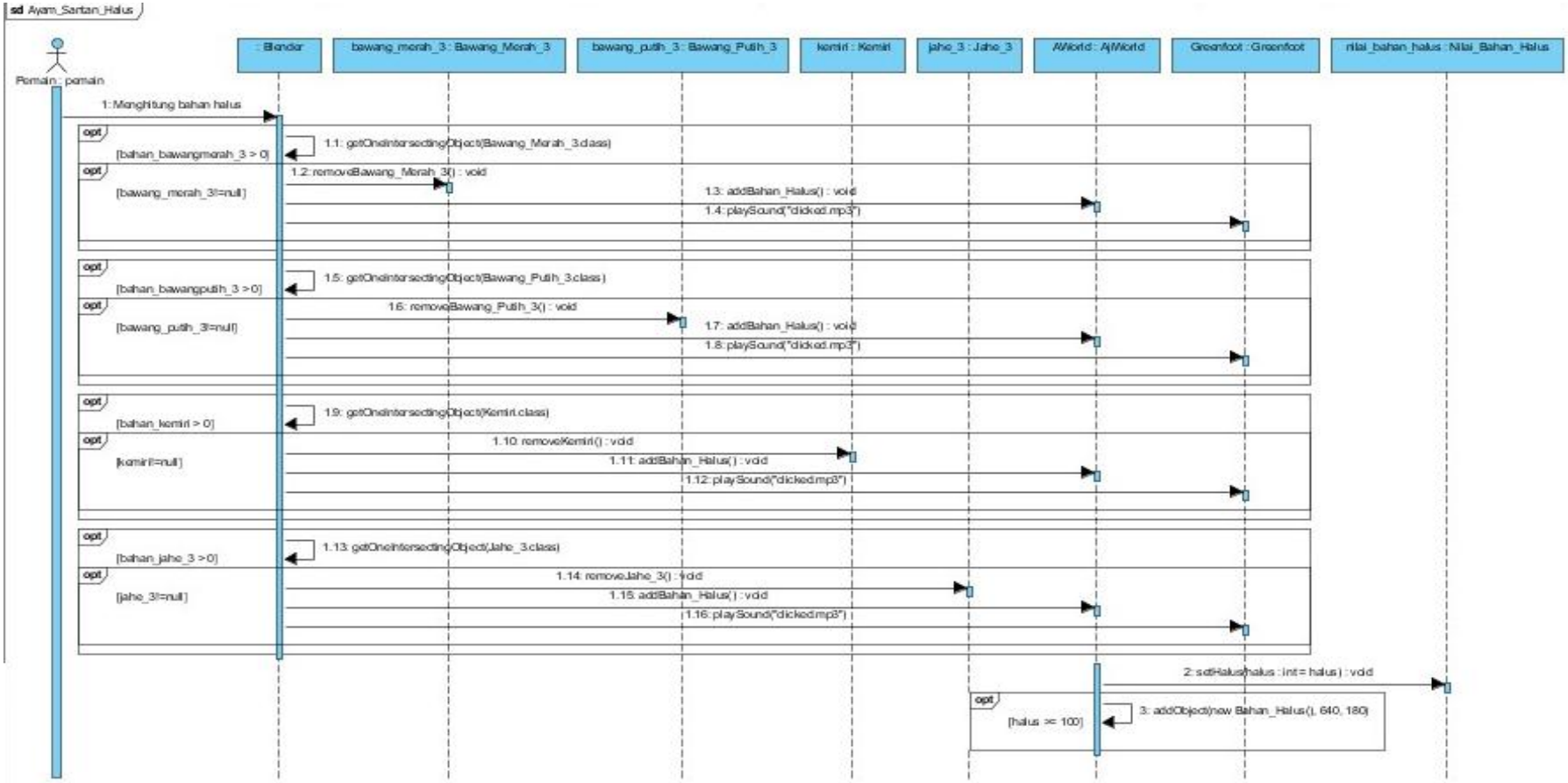
Gambar III.44 Diagram Sekuen Menghitung nilai

7. Diagram Sekuen Memainkan tingkat selanjutnya



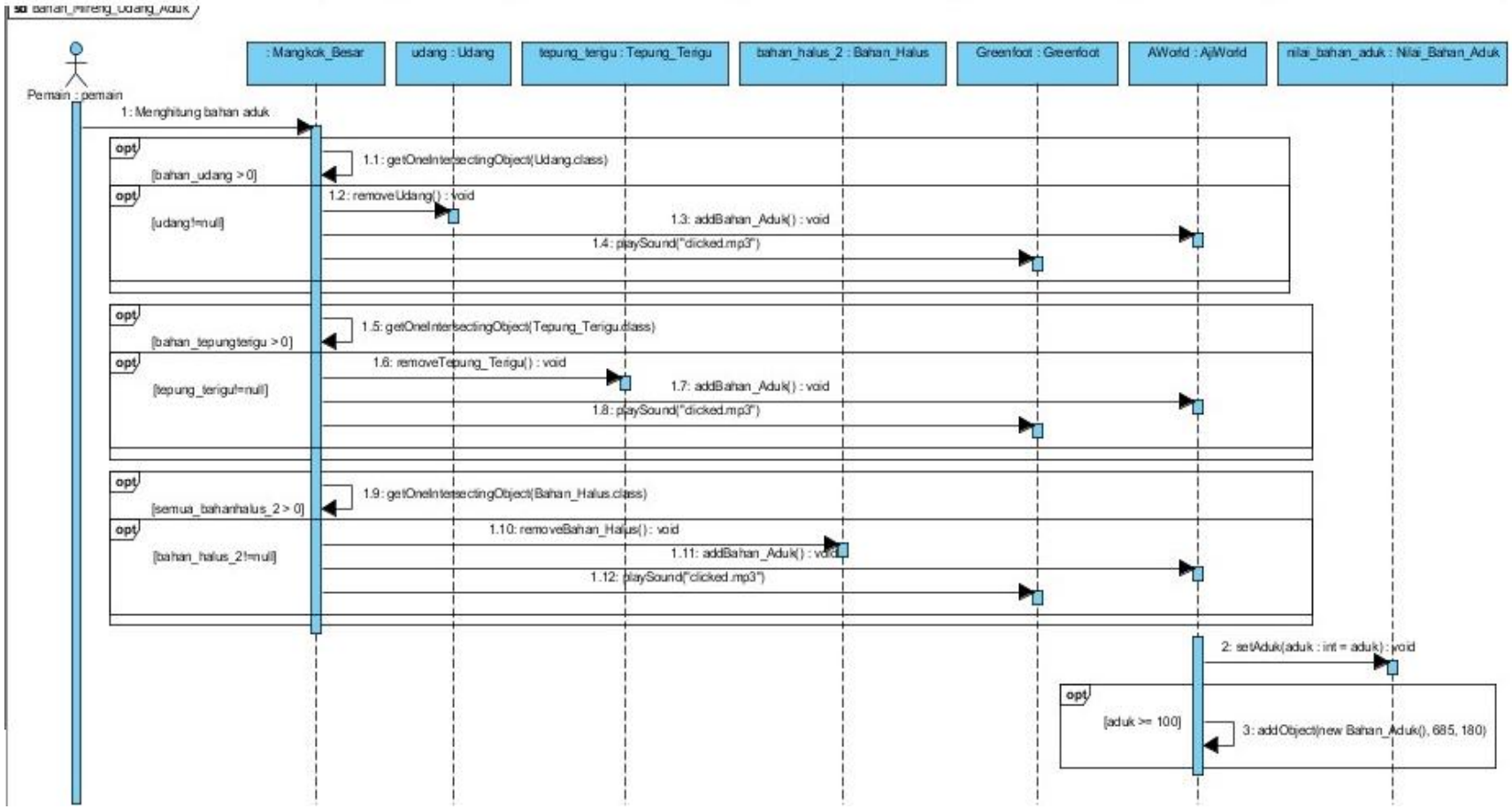
Gambar III.45 Diagram Sekuen Memainkan tingkat selanjutnya

8. Diagram Sekuen Menghitung bahan halus



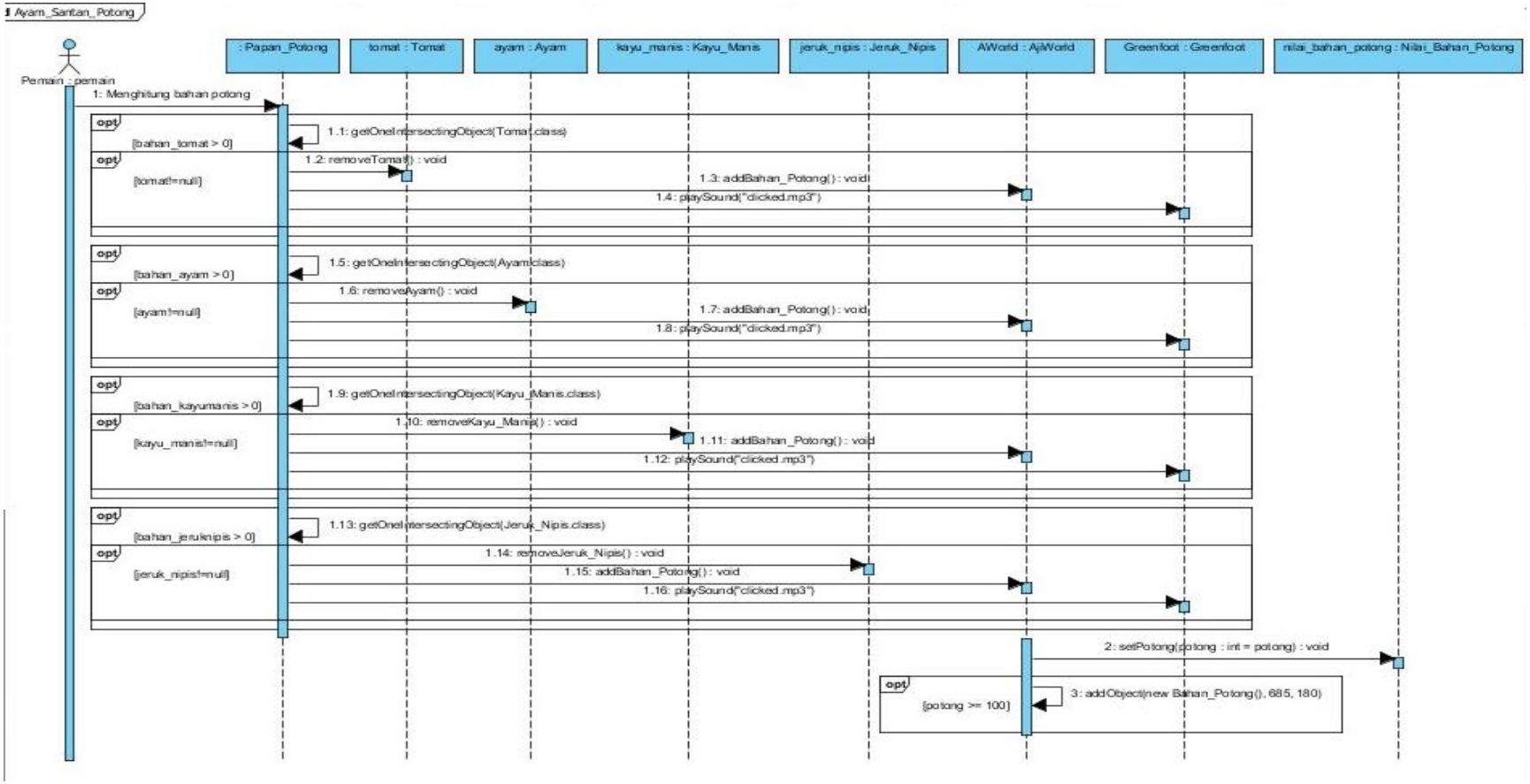
Gambar III.46 Diagram Sekuen Menghitung bahan halus

9. Diagram Sekuen Menghitung bahan aduk



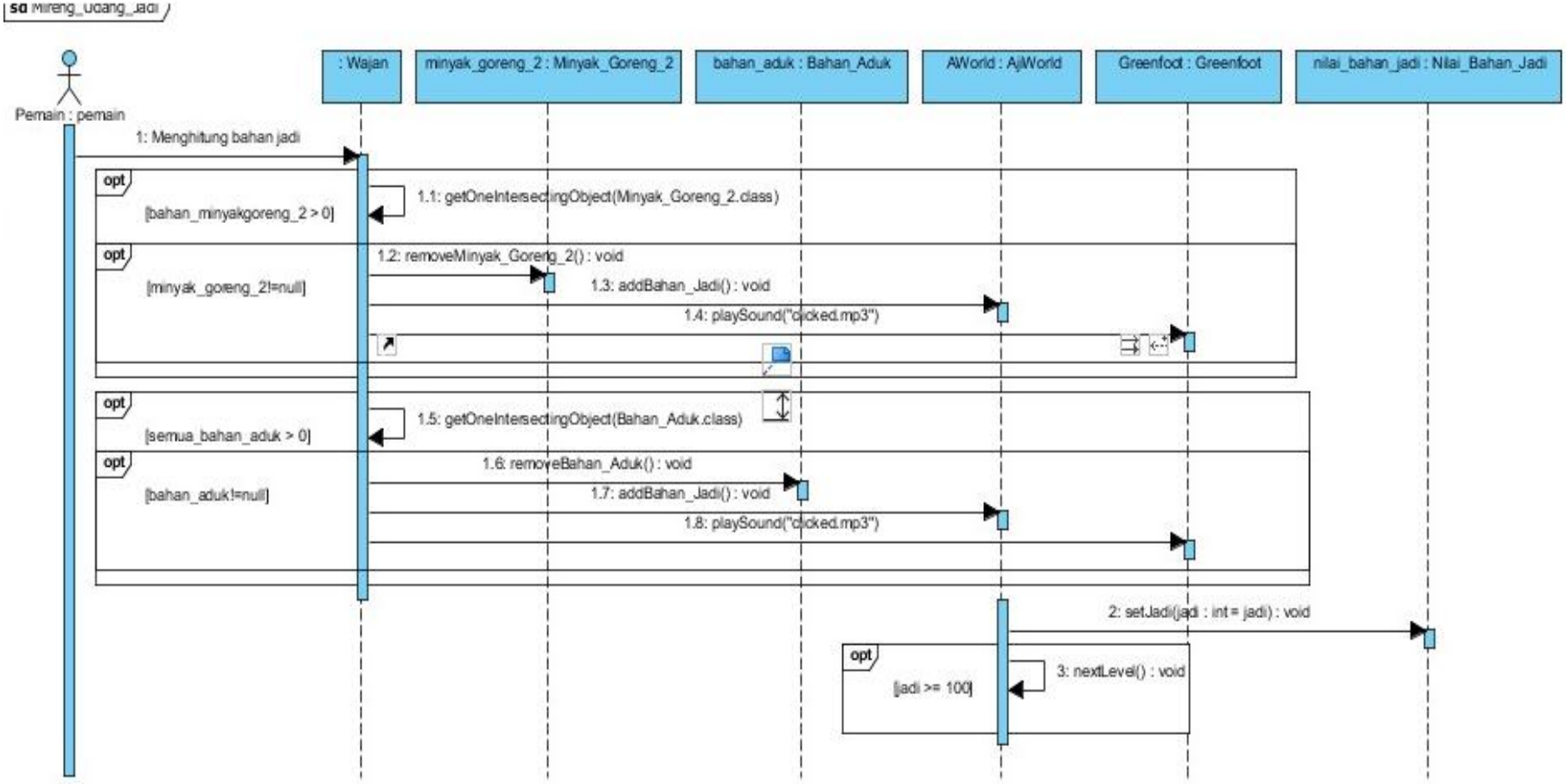
Gambar III.47 Diagram Sekuen Menghitung bahan aduk

10. Diagram Sekuen Menghitung bahan potong



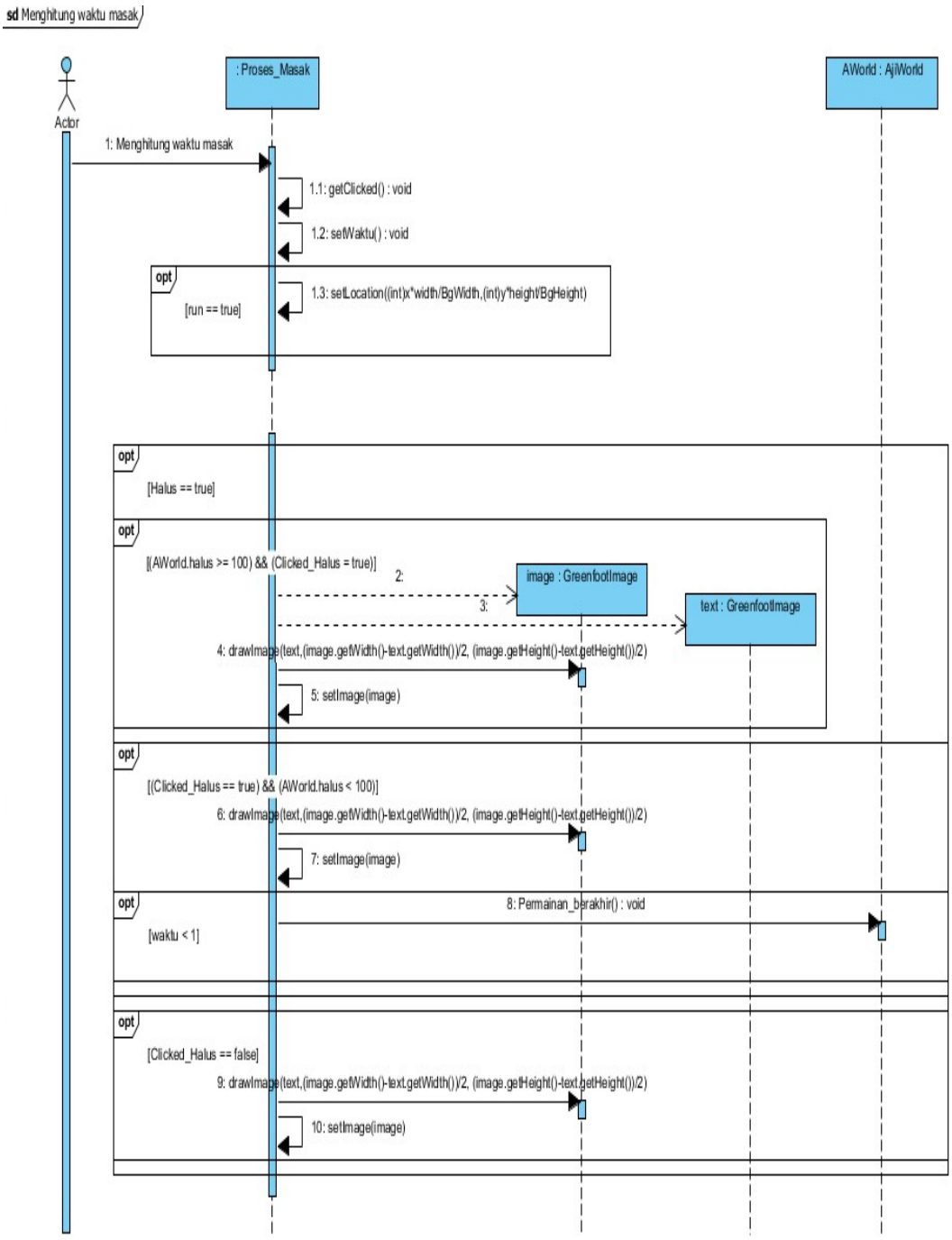
Gambar III.48 Diagram Sekuen Menghitung bahan potong

11. Diagram Sekuen Menghitung bahan jadi



Gambar III.49 Diagram Sekuen Menghitung bahan jadi

12. Diagram Sekuen Menghitung waktu masak



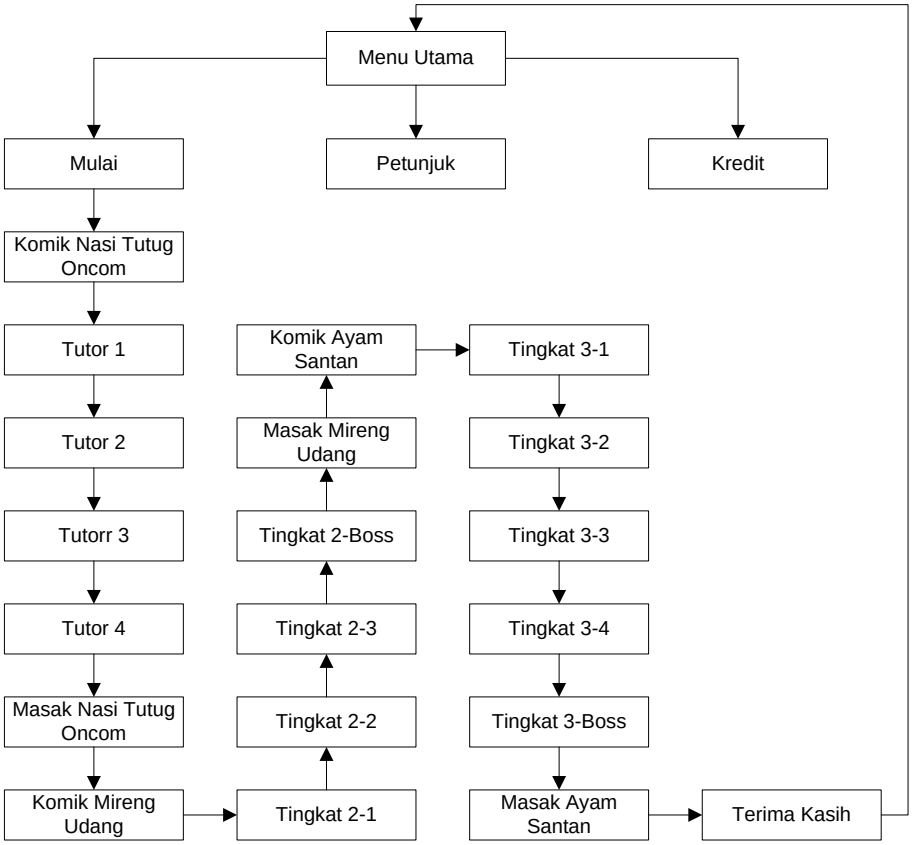
Gambar III.50 Diagram Sekuen Menghitung waktu masak

III.2 Perancangan Sistem

Perancangan sistem adalah suatu bagian dari metodologi pengembangan suatu perangkat lunak yang dilakukan setelah tahapan untuk memberikan gambaran secara terperinci.

III.2.1 Perancangan Struktur Menu

Perancangan struktur menu merupakan fitur-fitur yang disediakan pada sistem yang dirancang, dan untuk mempermudah dalam penggunaan sistem lainnya.

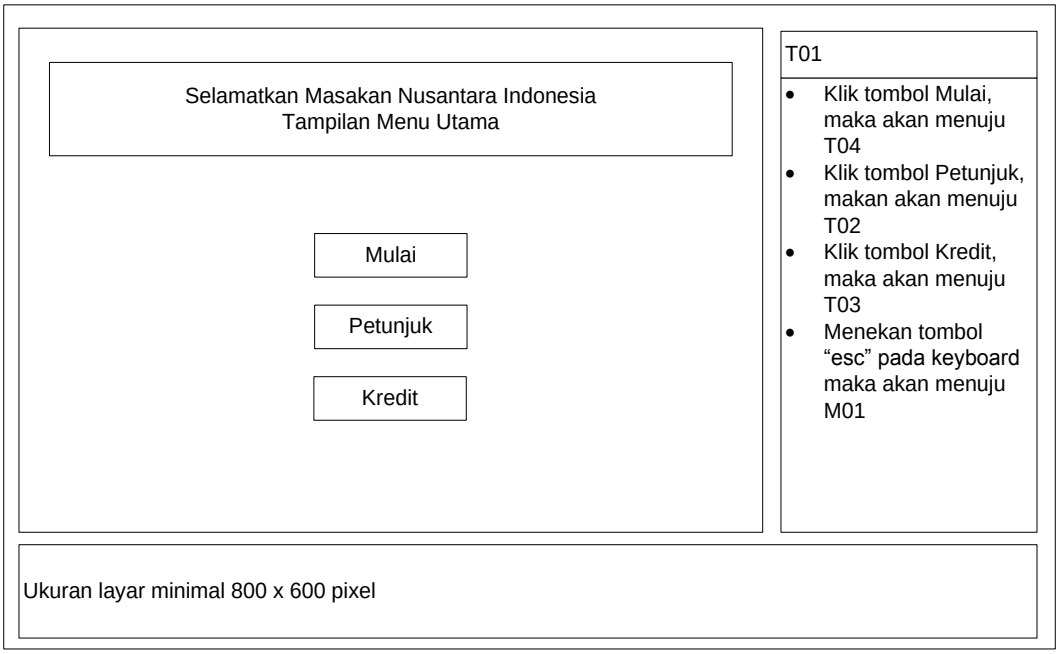


Gambar III.51 Struktur Menu

III.2.2 Perancangan Antar Muka

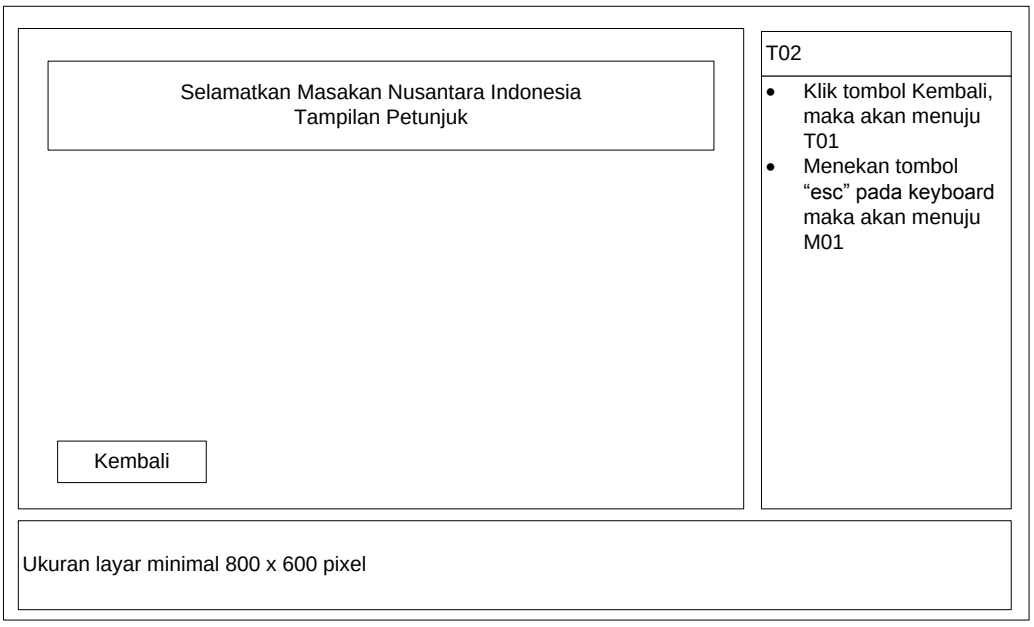
Perancangan antar muka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan memudahkan dalam implementasi aplikasi serta pembuatan aplikasi.

1. Menu Utama



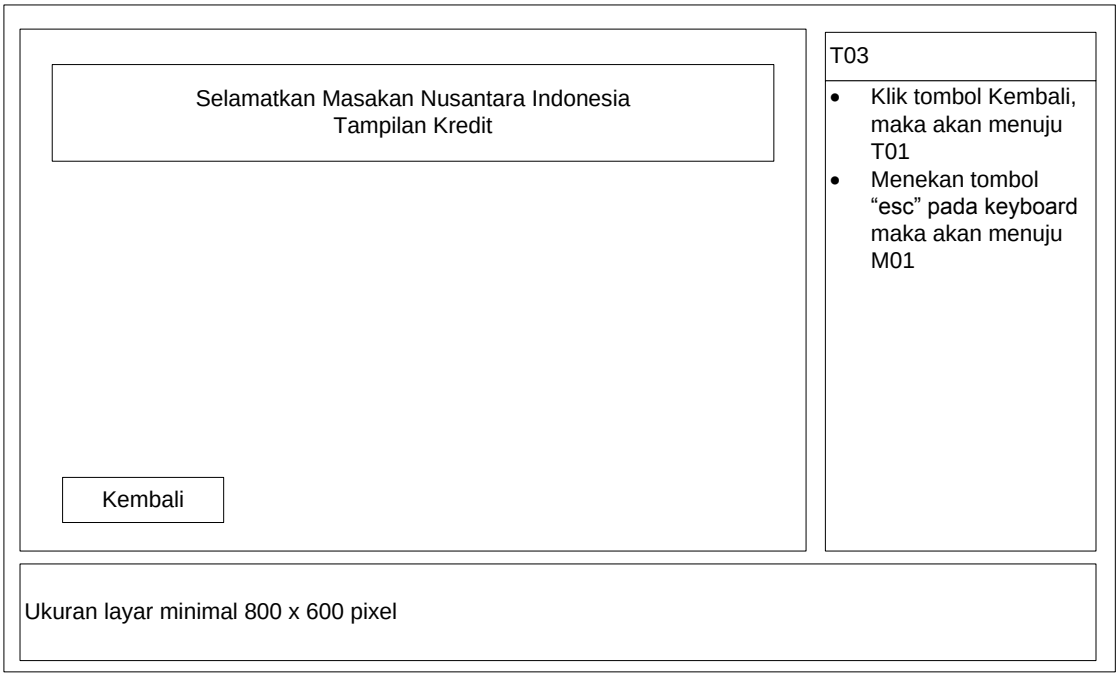
Gambar III.52 Perancangan antar muka menu utama

2. Petunjuk



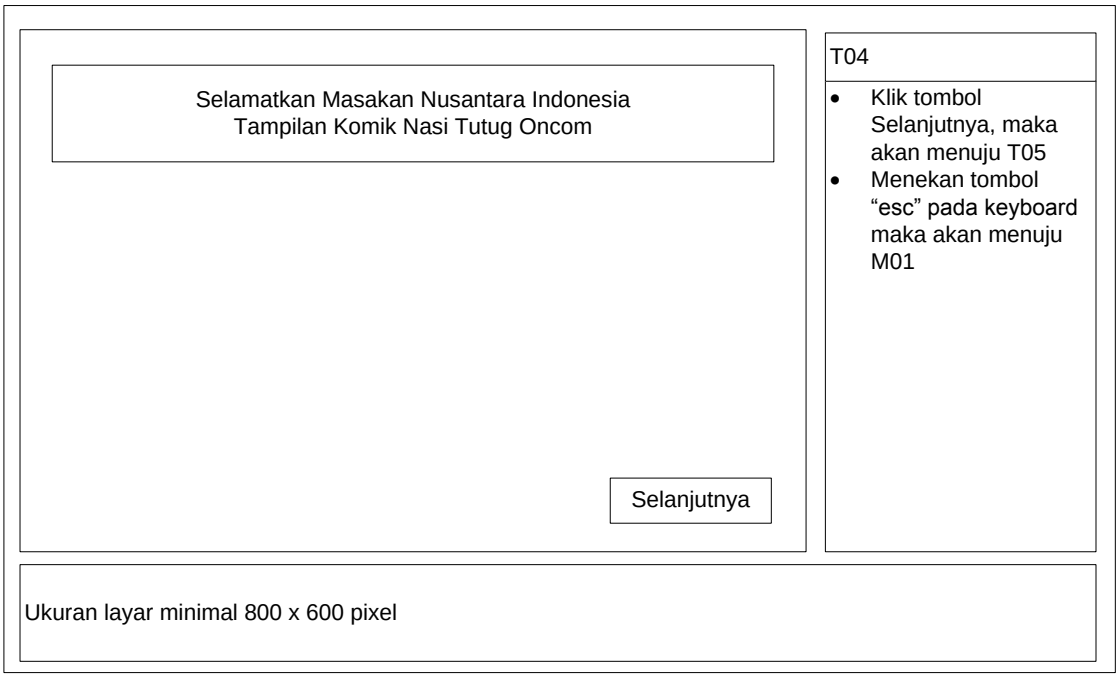
Gambar III.53 Perancangan antar muka Petunjuk

3. Kredit



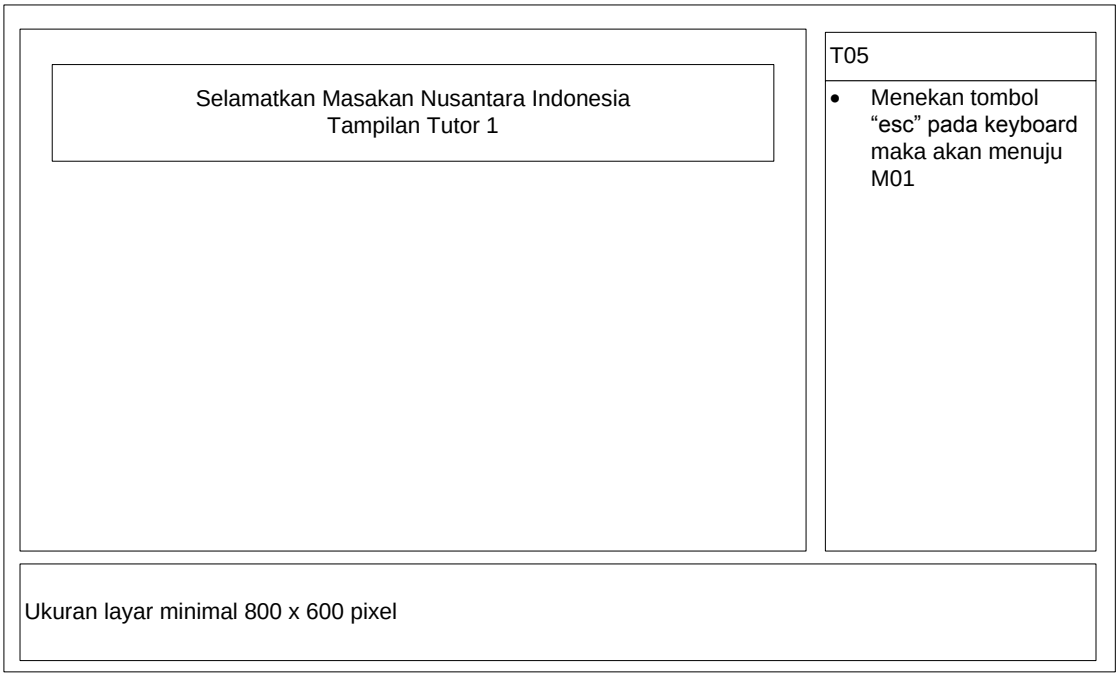
Gambar III.54 Perancangan antar muka Kredit

4. Komik_Nasi_Tutug_Oncom



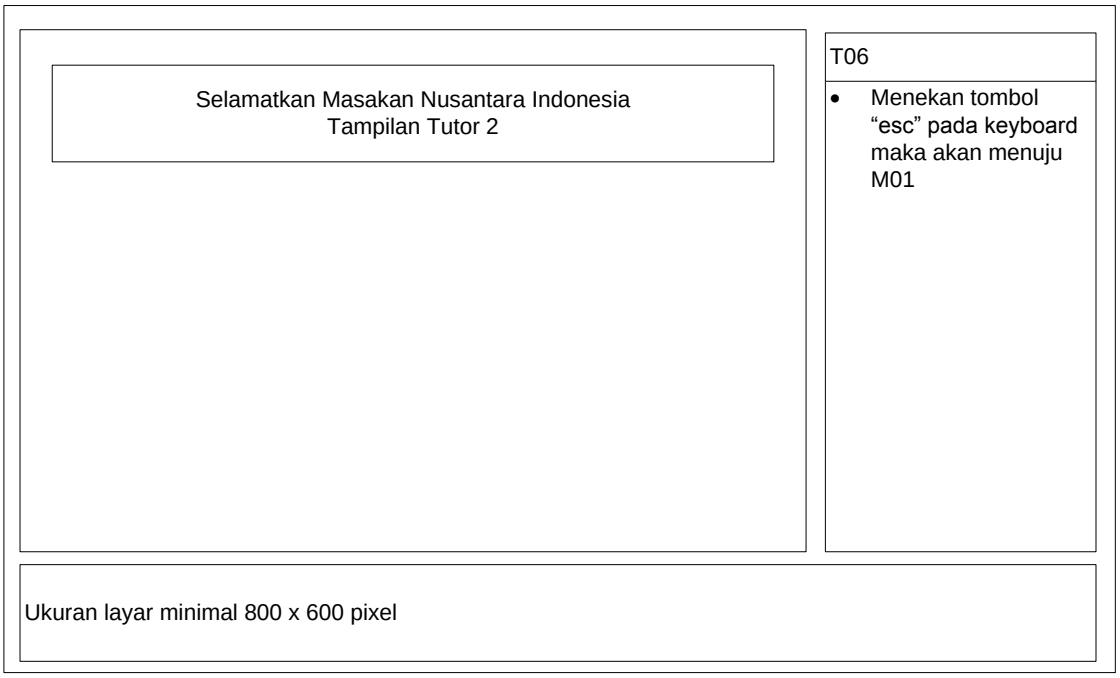
Gambar III.55 Perancangan antar muka Komik_Nasi_Tutug_Oncom

5. Tutor_1



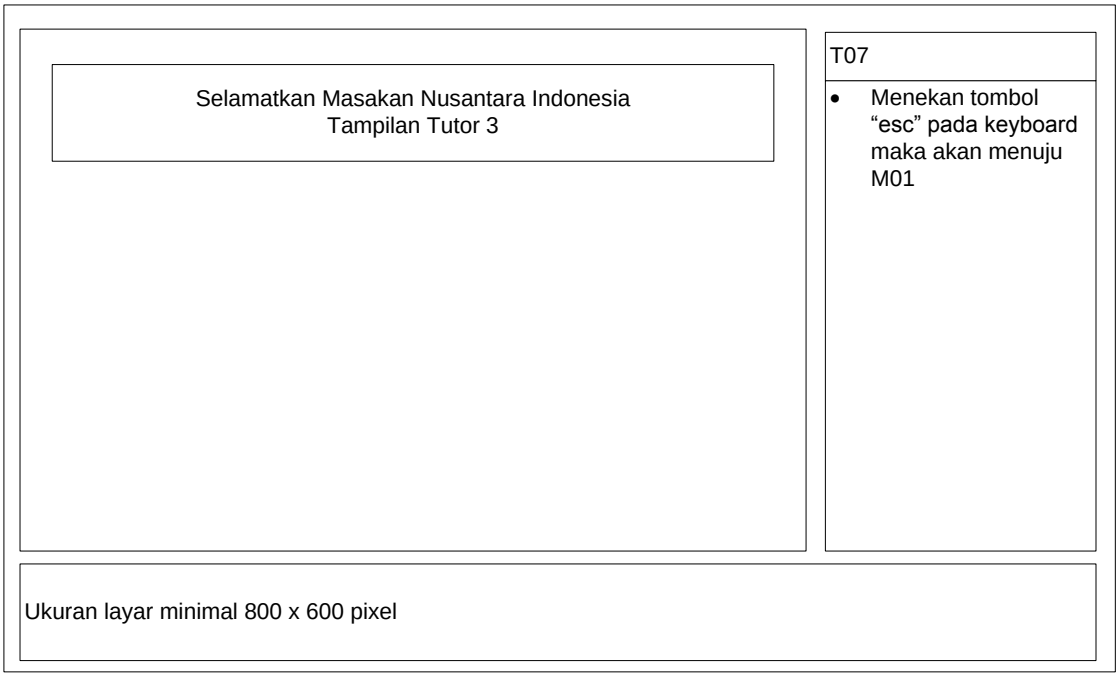
Gambar III.56 Perancangan antar muka Tutor_1

6. Tutor_2



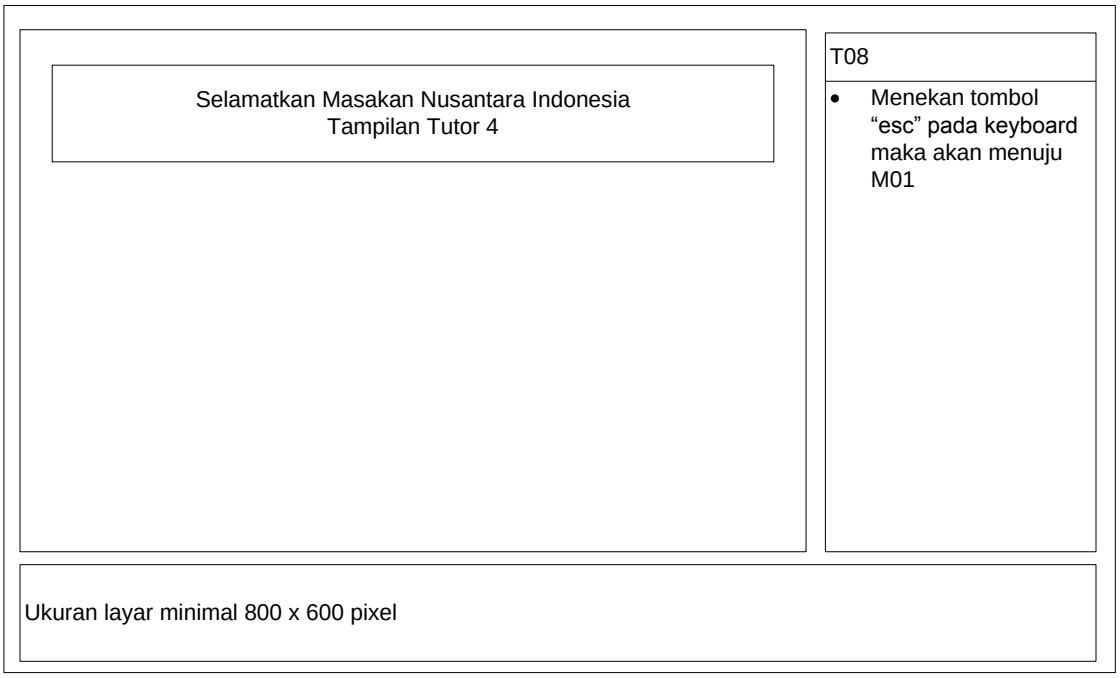
Gambar III.57 Perancangan antar muka Tutor_2

7. Tutor_3



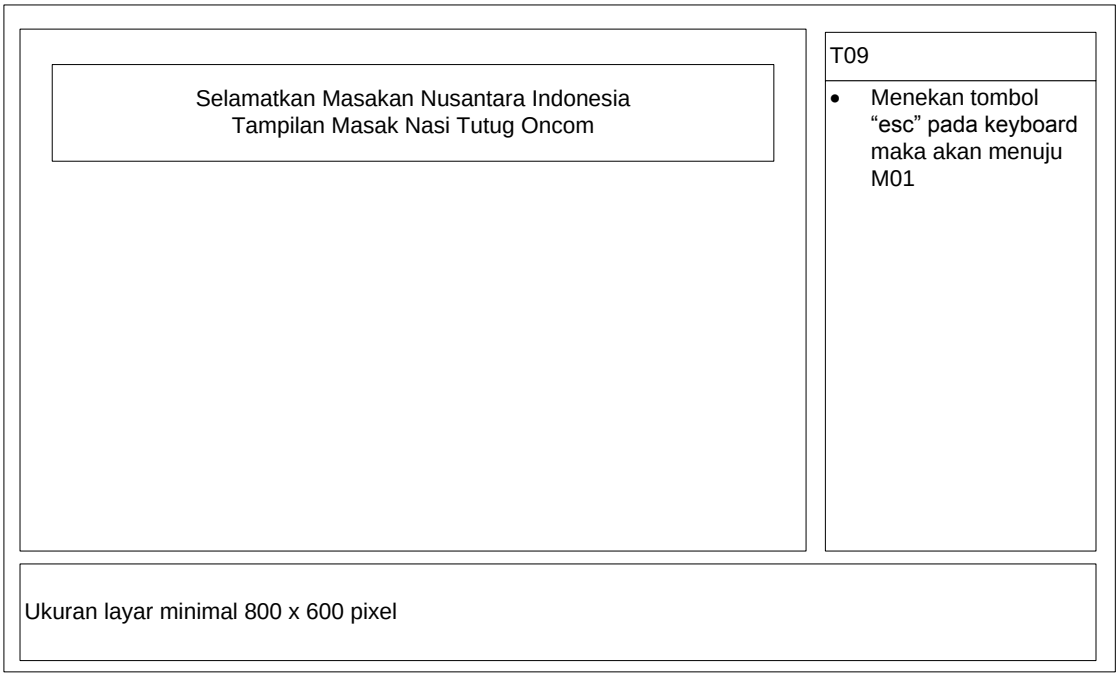
Gambar III.58 Perancangan antar muka Tutor_3

8. Tutor_4



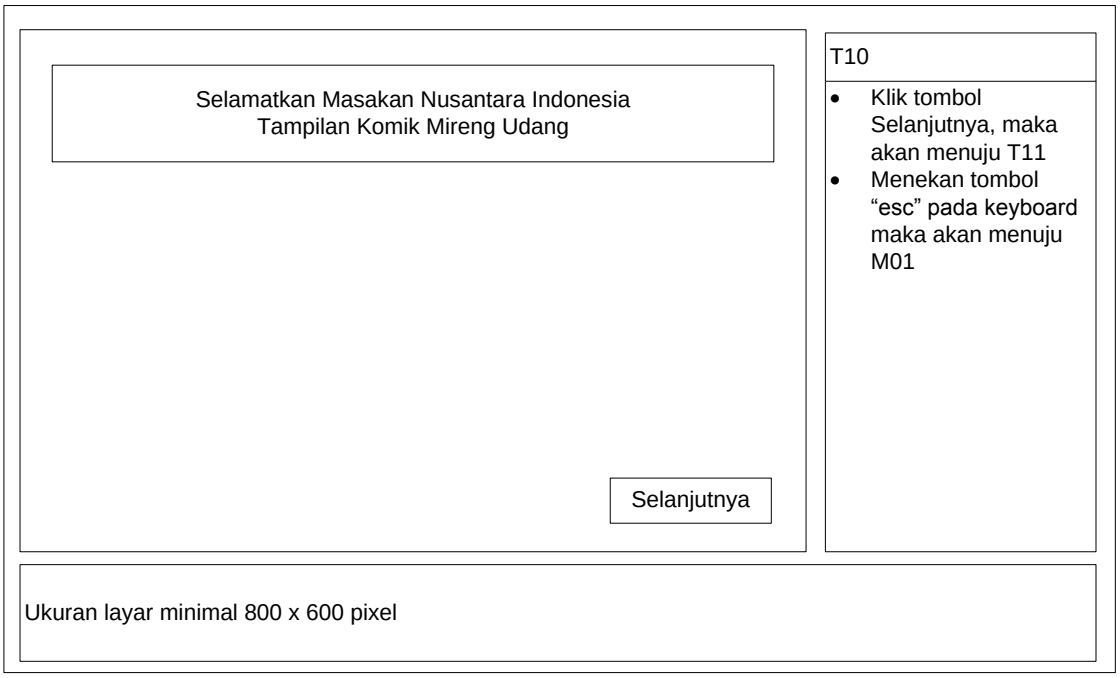
Gambar III.59 Perancangan antar muka Tutor_4

9. Masak_Nasi_Tutug_Oncom



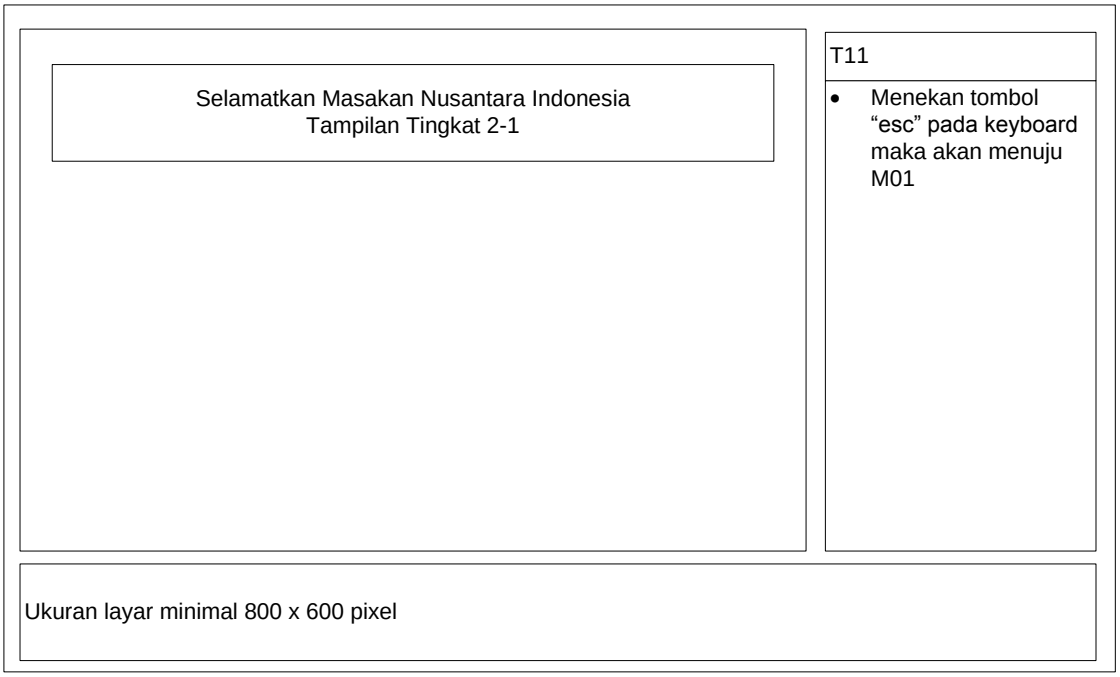
Gambar III.60 Perancangan antar muka Masak_Nasi_Tutug_Oncom

10. Komik_Mireng_Udang



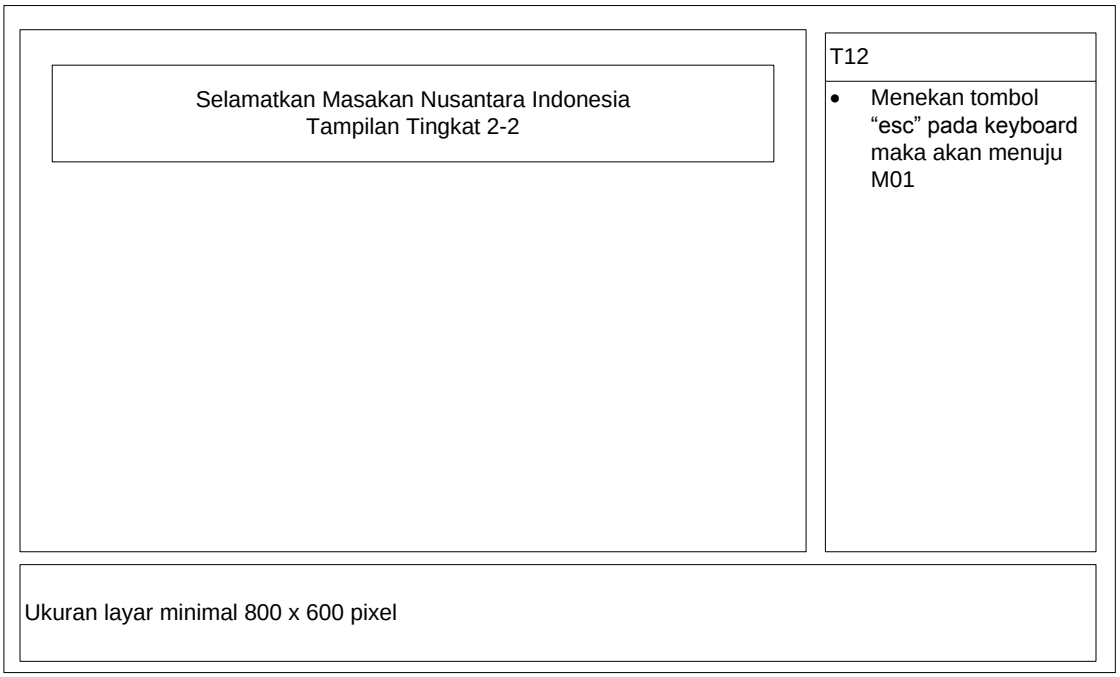
Gambar III.61 Perancangan antar muka Komik_Nasi_Tutug_Oncom

11. Level_2_1



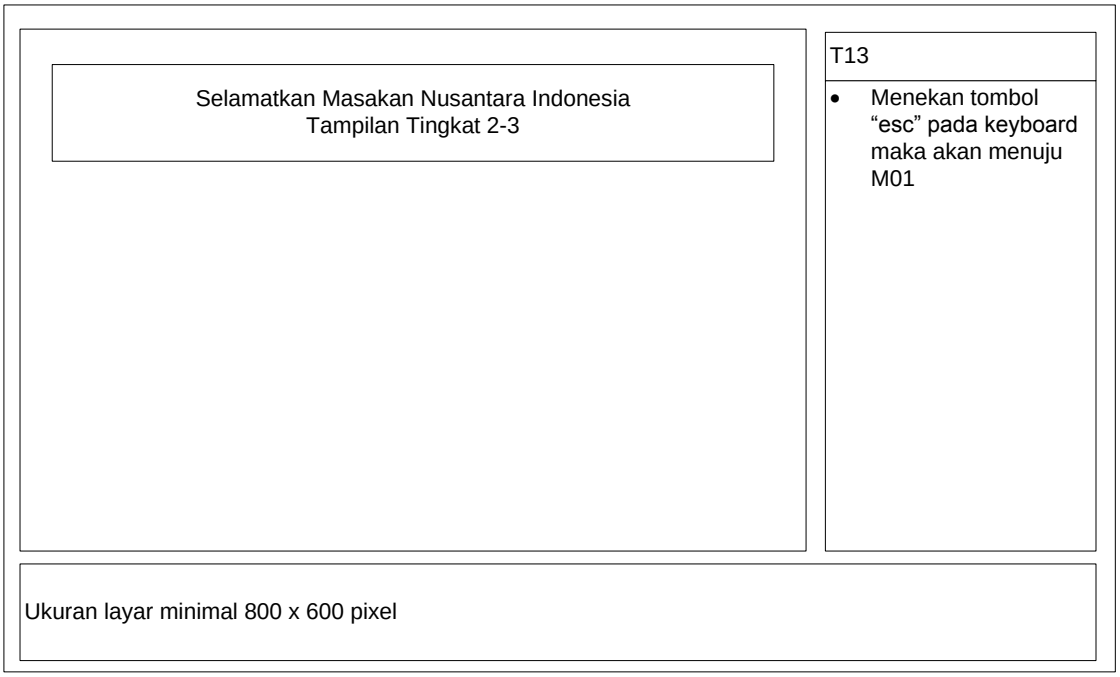
Gambar III.62 Perancangan antar muka Level_2_1

12. Level_2_2



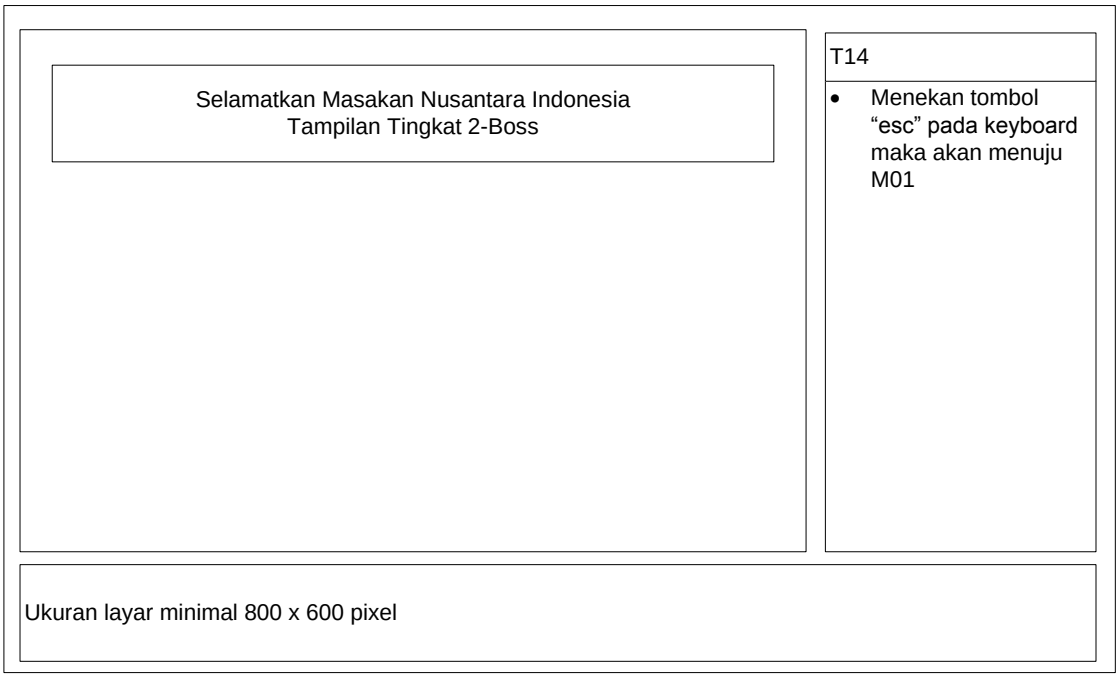
Gambar III.63 Perancangan antar muka Level_2_2

13. Level_2_3



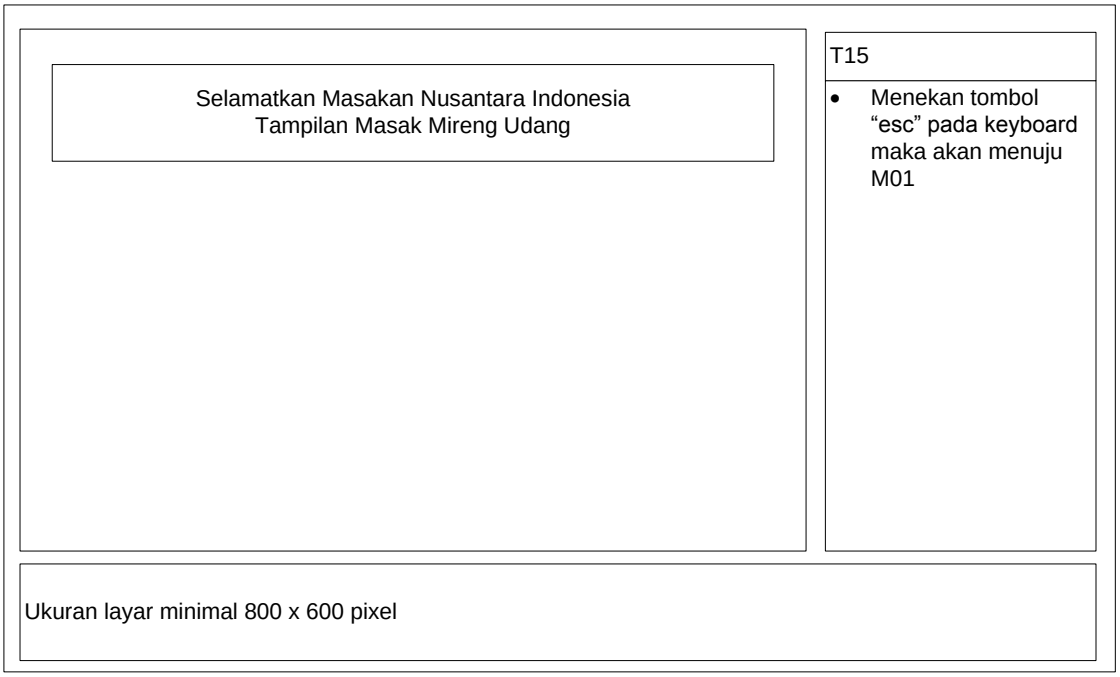
Gambar III.64 Perancangan antar muka Level_2_3

14. Level_2_Boss



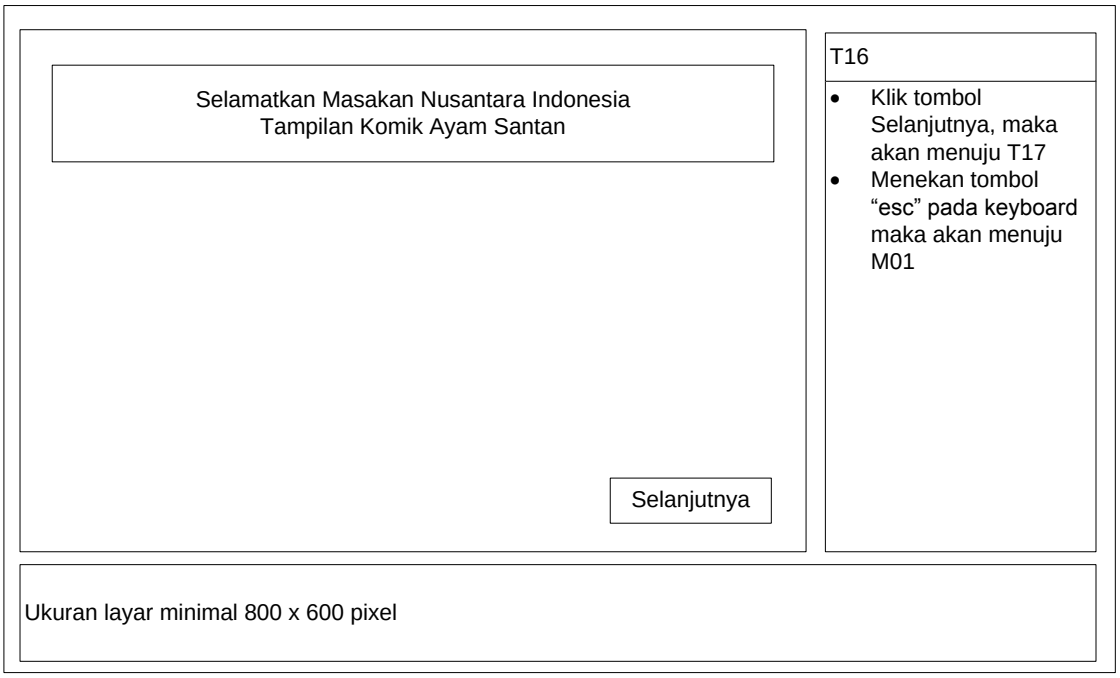
Gambar III.65 Perancangan antar muka Level_2_Boss

15. Masak_Mireng_Udang



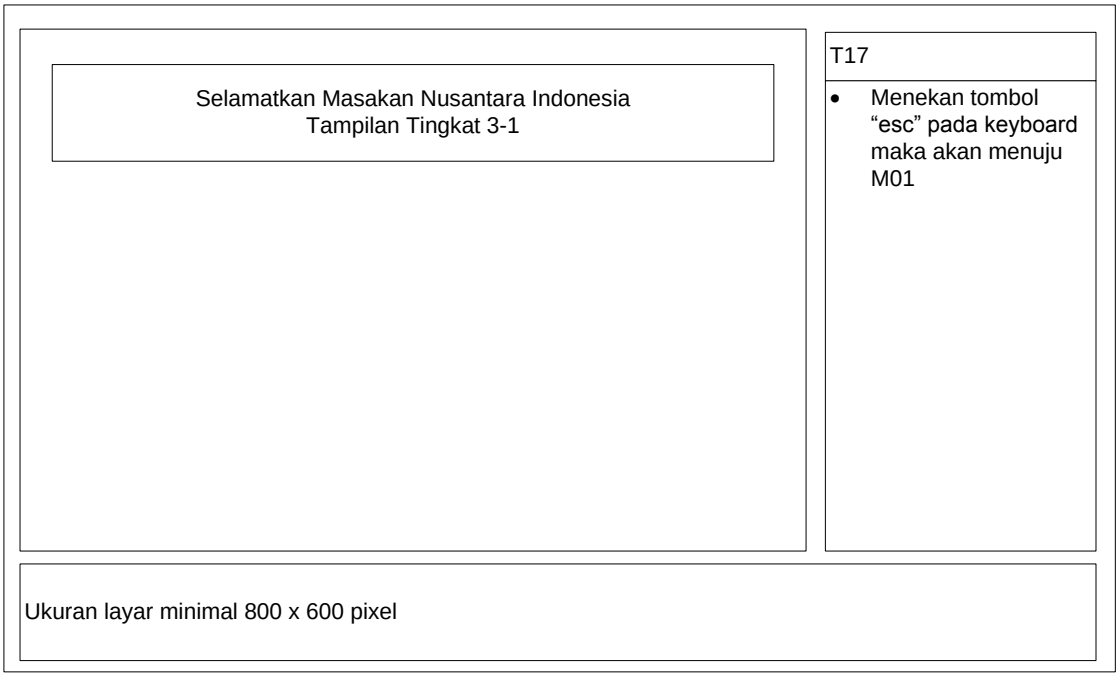
Gambar III.66 Perancangan antar muka Masak_Mireng_Udang

16. Komik_Ayam_Santan



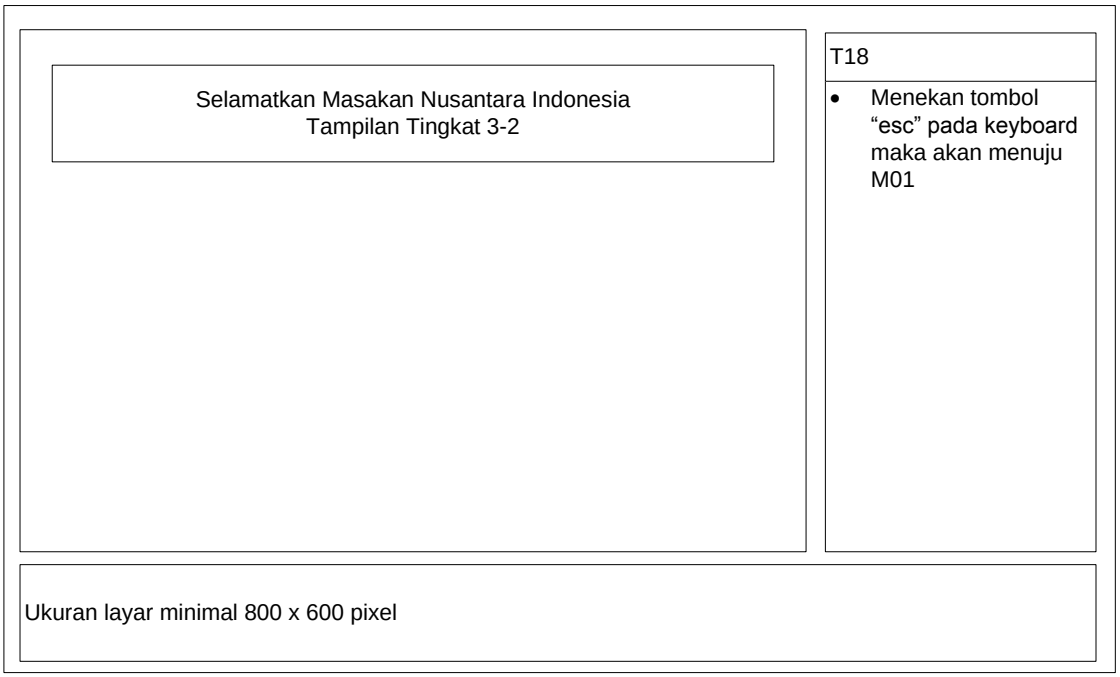
Gambar III.67 Perancangan antar muka Komik_Ayam_Santan

17. Level_3_1



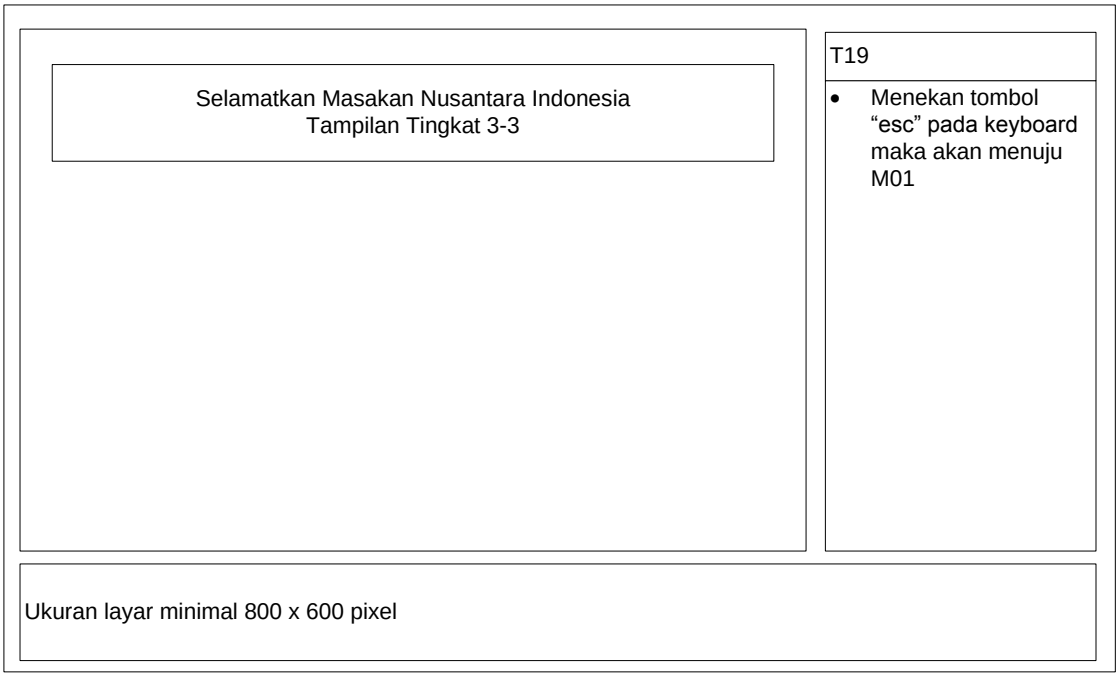
Gambar III.68 Perancangan antar muka Level_3_1

18. Level_3_2



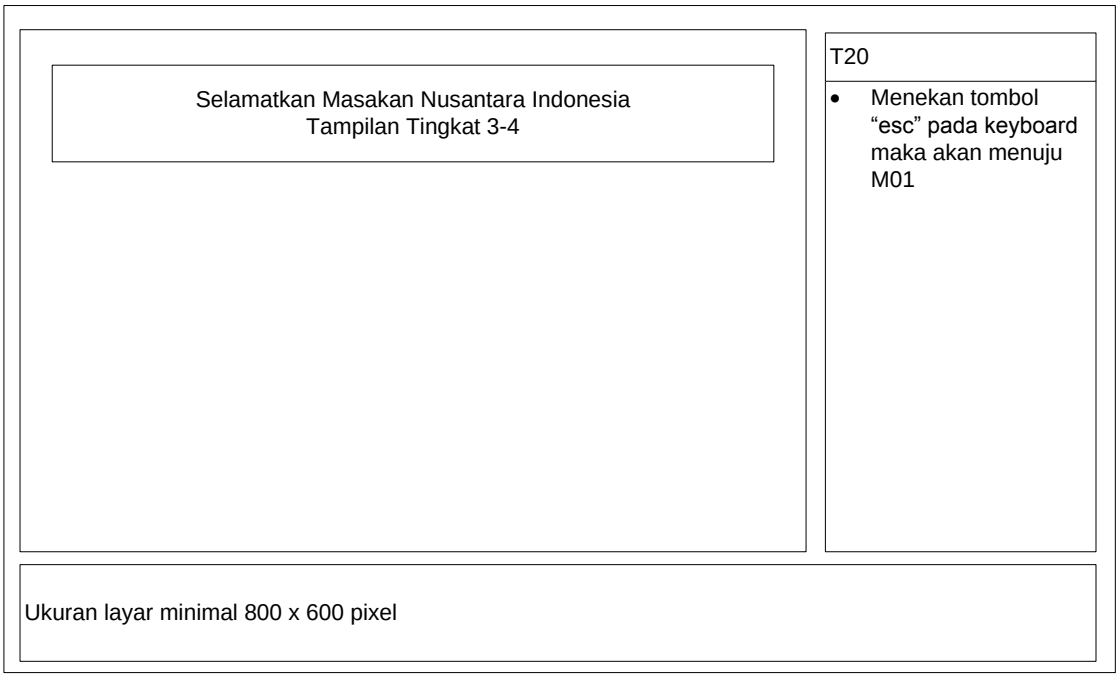
Gambar III.69 Perancangan antar muka Level_3_2

19. Level_3_3



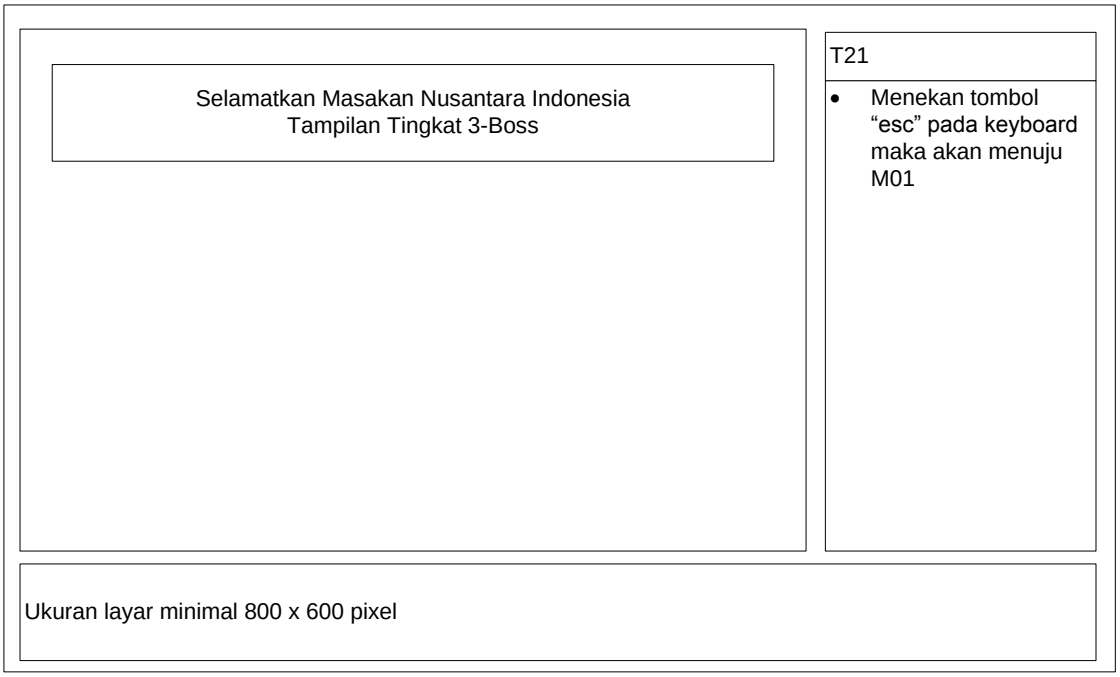
Gambar III.70 Perancangan antar muka Level_3_3

20. Level_3_4



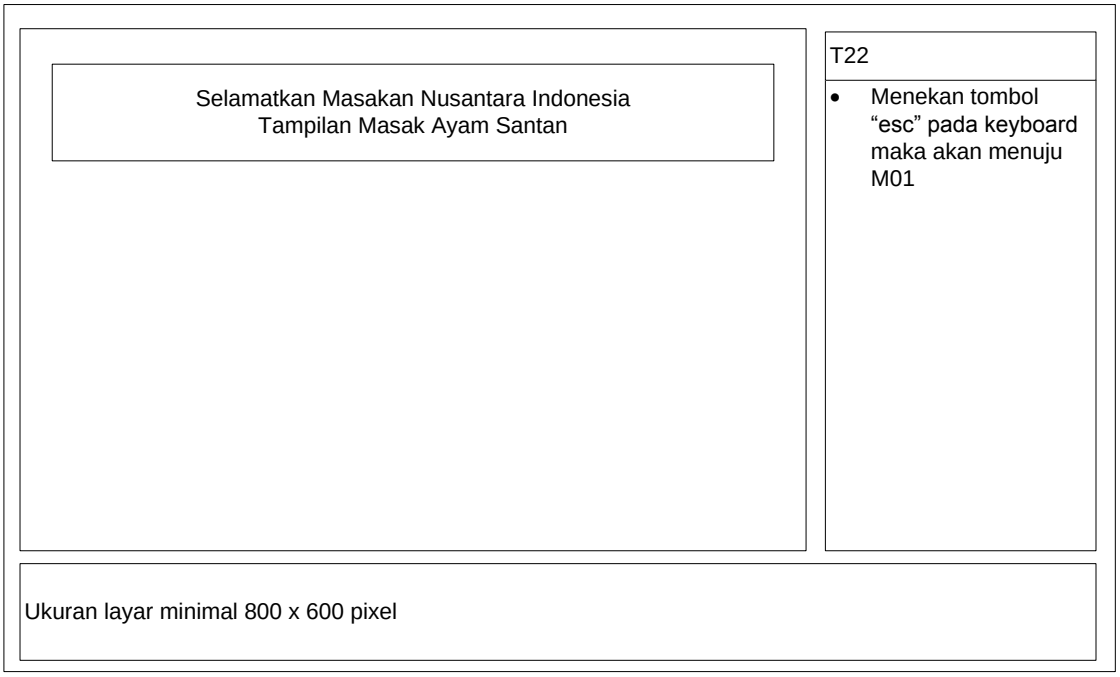
Gambar III.71 Perancangan antar muka Level_3_4

21. Level_3_Boss



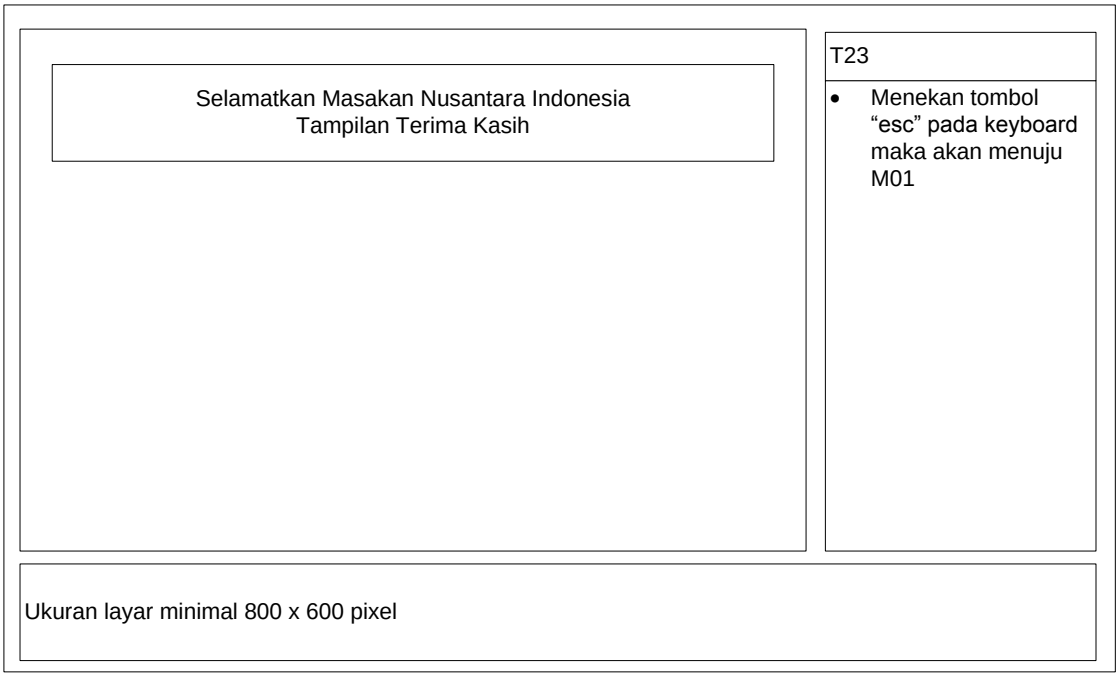
Gambar III.72 Perancangan antar muka Level_3_Boss

22. Masak_Ayam_Santan



Gambar III.73 Perancangan antar muka Masak_Ayam_Santan

23. Terima Kasih

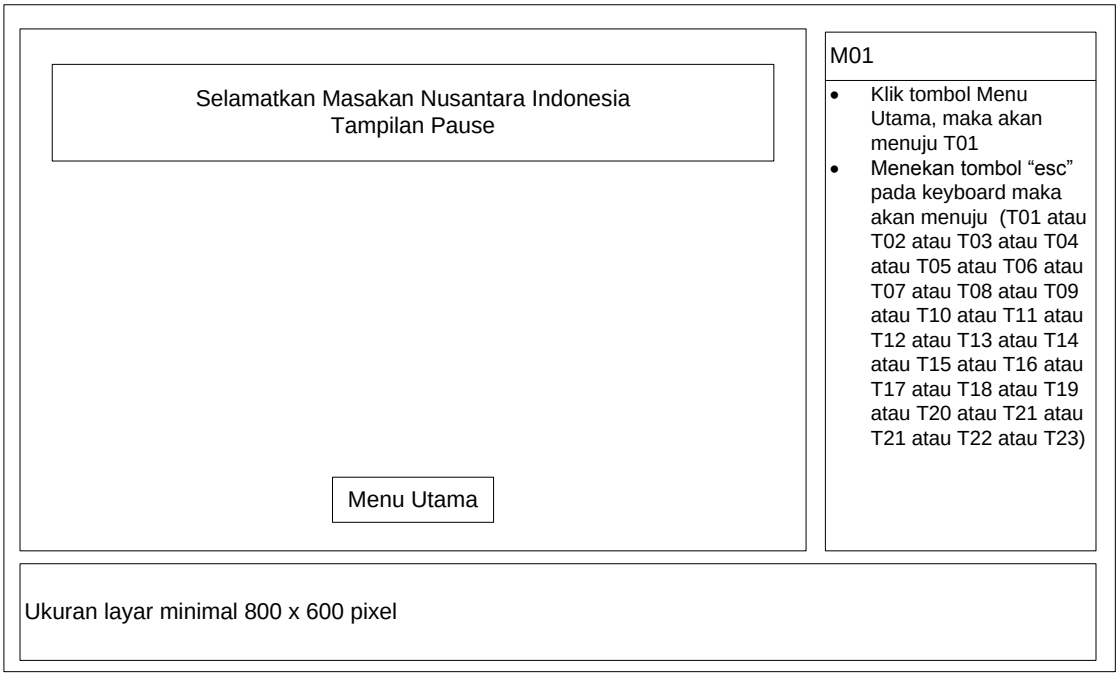


Gambar III.74 Perancangan antar muka Terima Kasih

III.2.3 Perancangan Pesan

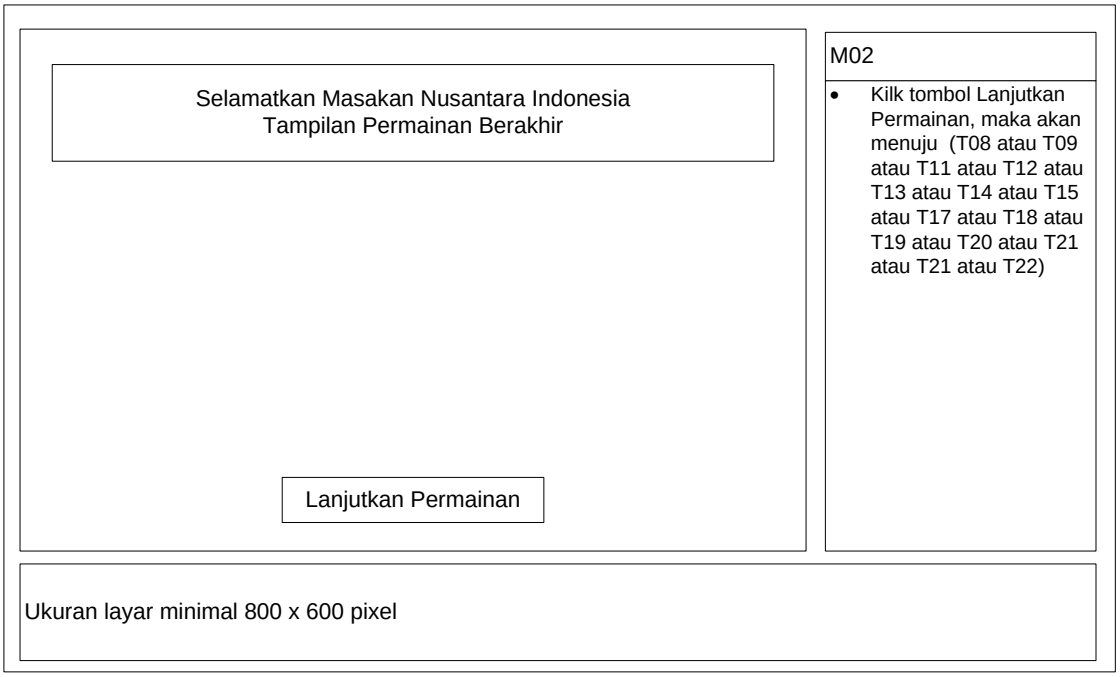
Perancangan pesan bertujuan untuk memberikan peringatan jika user melakukan aktivitas tertentu yang akan memicu pesan.

1. Pause



Gambar III.75 Perancangan pesan Pause

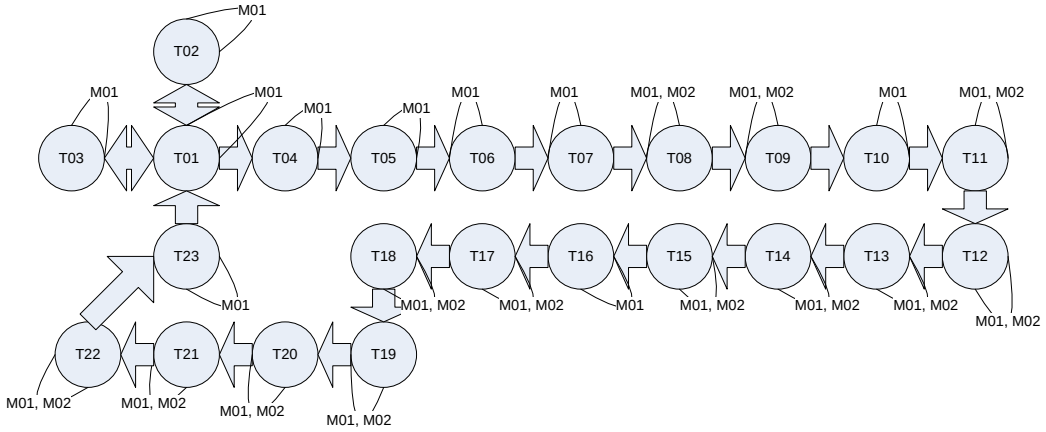
2. Permainan Berakhir



Gambar III.76 Perancangan pesan Permainan Berakhir

III.2.4 Jaringan Semantik

Jaringan semantik bertujuan untuk menggambarkan hubungan antarmuka dan pesan dalam permainan selamatkan masakan nusantara Indonesia.



Gambar III.77 Jaringan Semantik