BAB 3

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Pada analisis sistem akan menguraikan suatu sistem secara utuh beserta komponen-komponennya mencakup identifikasi dan evaluasi dari permasalahan – permasalahan juga kebutuhan kebutuhan yang diperlukan sehingga hubungan unsur – unsur itu menjadi jelas serta dapat diusulkan perbaikan – perbaikannya. Bagian ini meliputi analisis masalah, deskripsi sistem, analisis non fungsional dan analisis fungsional.

3.1.1 Analisis Masalah

Perencanaan dalam perjalanan pendakian gunung merupakan hal yang mendasar namun masih banyak pendaki yang tidak tahu bahkan menggabaikan hal – hal yang menjadi penunjang kegiatan ini. Hal tersebut yang menjadi salah satu faktor meningkatnya korban kecelakaan – kecelakaan digunung, walaupun seberapa pintar dan hebatnya kemampuan seorang pendaki, tetap saja selalu ada faktor lain diluar kuasa manusia. Yang menjadi pembahasan dalam masalah ini adalah minimnya akses informasi yang dimiliki para pendaki pemula untuk mempelajari informasi gunung serta langkah – langkah apa yang harus dilakukan selama masa persiapan kegiatan dilakukan. Disisi lain, masalah sumber informasi yang informatif, mudah dipahami dan visualisasi yang atraktif juga dibutuhkan oleh para pendaki pemula.

Sejauh ini informasi - informasi kegiatan pendakian gunung dan berbagai disiplin ilmu yang mendukung kegiatan tersebut bisa didapatkan hanya melalui organisasi-organisasi pendakian gunung, juga sebagian informasi-informasi bagaimana melakukan pendakian dapat ditemui diberbagai situs internet, tetapi hal ini justru membingungkan pembacanya karena informasi yang disampaikan tidak jelas sumbernya.

3.1.2 Analisis *Game* Sejenis

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

1. **Pengenalan *Game* The Mountaineer**

   Game The Mountaineer adalah sebuah *game* berbasis web yang diluncurkan oleh abroy.com. The Mountaineer bergenre adventure sederhana dan menampilkan grafis 2D flash. *Game* ini mengangkat tema tentang perjalanan seorang pendaki gunung yang melewati berbagai rintangan dalam *level* pendakiannya. Tampilan dari *game* tersebut dapat dilihat pada gambar dibawah ini.



Gambar 3. 1 Tampilan awal *game* The Mountaineer



Gambar 3. 2 Tampilan *level* 2 *game* The Mountaineer

2. *Gameplay* & *Storyline Game* The Mountaineer

Adapun *gameplay* & *storyline game* The Mountaineer adalah sebagai berikut :

a. *Game* ini memiliki 8 *level* dengan tingkat kesulitan masing – masing. Mulai dari kaki gunung hingga berakhir di puncak gunung
b. *Player* harus berpikir mencari cara menyelesaikan misi disetiap *level*, seperti :

Tabel 3. 1 *Storyline Game* Sejenis

Level	Misi *Storyline*
1	Menjatuhkan pohon tumbang untuk menyebrang
2	Menyusun kayu – kayu untuk meyebrangi sungai
3	Memperdayai kambing hutan
4	Menghindari longsoran batu
5	Memperdayai beruang gua
6	Menyusun puzzle
7	Memanjat dinding es
8	Menyusun marker bendera

c. Karakter bergerak mengambil dan melakukan aksi dengan cara mengklik tiap item.

Berikut merupakan aspek elemen-elemen dasar yang disorot dari sebuah *game* yang terdapat pada *game* sejenis tersebut seperti yang terlihat pada tabel dibawah ini.

Tabel 3. 2 Elemen yang terdapat pada *game* sejenis

Elemen	*The Mountaineer*
Karakter	The Mountaineer
Rules	Tidak dijelaskan
Level	Terdapat 8 *level*
Nyawa	Tidak ada
Score	Tidak ada
Boss	Tidak ada
Evaluasi / *Victory Conditions*	Tidak ada
Penyajian Materi	Tidak ada
Perspective	*Side Scrolling*
Kontroller	*Mouse Click*

Berdasarkan analisis yang telah dilakukan, maka ditarik kesimpulan bahwa *game* the Mountaineer memiliki kelebihan dan kekurangan sebagai berikut:

Tabel 3. 3 Kelebihan dan kekurangan *game* sejenis

Kelebihan	Kekurangan
a. Tiap *level* memiliki rules berpikir yang berbeda	a. Hanya mengangkat sisi trik permainan saja, tidak melibatkan pengetahuan pendakian
b. Tampilan sederhana namun cukup menarik dan unik	b. Keterbatasan gerak karakter, karakter hanya dapat bergerak (otomatis) ketika mengklik item

	c. Tidak ada petunjuk atau perintah yang harus dilakukan oleh *player* pada tiap *level*nya.

Untuk itu, maka dibutuhkan suatu *game* bertemakan pendakian gunung yang lebih banyak memuat materi persiapan perencanaan perjalanan dan materi dasar pendakian gunung khususnya untuk para pendaki pemula.

3.1.3 Deskripsi Sistem

Game yang dibangun berjudul *Jejak Pendaki* ini memiliki bergenre *adventure*. *Game* ini mengangkat tema pendakian gunung mulai dari perencanaan hingga siap mendaki. Misi dalam *game* ini awal mulanya *player* harus dapat mempersiapkan perencanaan perjalanan yang akan dia tempuh sehingga dapat mencapai ke puncak gunung melewati berbagai variasi halangan dan rintangan yang harus dihadapi. Semua dikemas dalam bentuk yang menarik dan lebih sederhana dari permasalahan yang sebenarnya dengan harapan dapat menghibur pemain sekaligus memberikan informasi tentang pengetahuan pendakian gunung. Berikut point – point yang ada pada *game* ini :

1. Dimainkan secara *single player* pada *platform desktop*
2. Visual grafis yang ditampilkan 2D
3. Terdiri dari 4 *Level* berbeda dengan tingkat kesulitan disetiap *level*nya
4. Mengangkat informasi persiapan pendakian gunung dan informasi beberapa gunung di Indonesia
5. Kecerdasan buatan yang diterapkan pada rintangan musuh

3.1.1.1 Story Line

Ketika dikota, jimbrow dan teman-temannya berencana merayakan HUT RI dengan mendaki gunung pada tanggal 17 agustus nanti. Namun ia belum pernah mendaki gunung sama sekali, ia adalah pendaki pemula. Oleh sebab itu, sebelum mengikuti pendakian 17 agustusnya ia harus menyusun rencana, mempersiapkan segala perlengkapan apa saja yang dibutuhkan dan mengetahui data dan informasi gunung yang akan didakinya serta melakukan simulasi pendakian digunung sangarra untuk melatih dirinya. (detail per *level* lihat dibagian *levelling*)

Sebulan sebelum 17 Agustus-an, ia akan berlatih simulasi pendakian di Gunung Sangarra. Ia harus mempersiapkan perlengkapan pendakiannya sesuai dengan kebutuhan dan menyusunnya kedalam ransel sedemikian rupa. Di gunung, jalur yang ia lewati tidak begitu mudah, ia harus melewati berbagai halangan dan rintangan serta menjaga kondisi tubuh untuk tetap prima dalam berkegiatan karena jika tidak, semuanya akan berakhir.

3.1.1.2 *Gameplay*

Misi didalam *game* jejak pendaki ini adalah tentang bagaimana *player* harus dapat mempersiapkan perjalanan pendakian gunung dan dapat melakukan pendakian dengan aman.

Mempersiapkan perjalanan pendakian gunung mulai dari menyusun rencana pendakian dan mendapatkan data dan informasi gunung yang akan didaki. Selain itu juga dapat memilih, mengatur, mengetahui kegunaan serta mem*packing* perlengkapan dan perbekalan yang dibutuhkan secara efisien. Melist perlengkapan sesuai dengan kebutuhan berdasarkan kategori perlengkapan perjalanan, perlengkapan tidur, perlengkapan masak, perlengkapan navigasi serta perlengkapan tambahan. Mempraktekkan cara *packing* ransel yang baik dari segi penempatan benda-benda yang berat sampai dengan perlengkapan akesoris yang ringan sekalipun, ini bergantung kepada cara kita menumpukkan berat beban kepada tubuh sedemikian rupa. Setelah selesai mem*packing* ransel, *player* juga

harus dapat memilih perlengkapan jalan yang sesuai dengan medan yang dia tempuh (gunung hutan) yang dipakai dibadan. Semua hal tersebut akan dinilai dari benar tidaknya penempatan perlengkapan yang dipilih dan juga waktu tercepat.

Sesudah semua perencanaan perjalanan dilakukan, sebagai pendaki pemula ia akan melakukan latihan simulasi pendakian di gunung sangarra lebih dulu. Pendakian dilakukan dalam 2hari, *start* dimulai dari kaki gunung dan *finish* dipuncak gunung. Tapi didalam pendakian kali ini tidak semudah itu, *player* harus melewati berbagai variasi rintangan, mulai dari melewati lubang dan jurang, juga mengalahkan ular dan babi hutan. Namun hal yang terpenting adalah mengatur stamina bar agar tetap dalam kondisi prima. Semakin banyak aktivitas yang dilakukan *player* maka stamina bar akan terus menurun, namun stamina bar bisa meningkat dengan mengambil beberapa buah yang ada disekitar jalur pendakian. Tetapi jika stamina bar terus menurun dan habis atau jatuh ke jurang maka itulah tanda *game* over dari pendakian ini.

3.1.1.3 *Levelling* dan *Scoring*

1. Skenario : *Level* 1 *Packing*

   Setting : Kamar

   Intro :

   Sebelum memulai perjalanan pendakian, *player* harus mempersiapkan perencanaan dan perlengkapan yang dibutuhkan.

   Misi :

   - Menyiapkan perlengkapan perjalanan yang akan dibawa
   - Mem*packing* perlengkapan kedalam ransel dengan benar

   Apa yang harus dilakukan :

   - Memilih perlengkapan yang perlu dibawa
   - Praktek *packing* yang baik

Outro :

Setelah semua perlengkapan telah di*packing* kedalam ransel, *player* menyiapkan diri untuk memakai perlengkapan jalan, tetapi jika perlengkapan belum ter*packing* semua *player* belum bisa melanjutkan ke*level* selanjutnya.

2. Skenario : *Level* 2 Berpakaian

Setting : Kamar

Intro :

Di*level* ini *player* bersiap untuk segera berangkat dan berpakaian sesuai dengan medan gunung hutan yang ditempuh

Misi :

- Memakai perlengkapan dengan benar
- Selesaikan sebelum waktu habis

Apa yang harus dilakukan :

Memilih dan memakai semua perlengkapan dengan benar secepat mungkin

3. Skenario : *Level* 3 Pendakian

Setting : Gunung Sangarra - H01.

Intro :

Sebelum kegunung di 17 agustusan nanti, *player* akan berlatih simulasi pendakian digunung ini lebih dulu.

Misi :

- Mencapai puncak
- Camp di pos 3

Rintangan :

- Jurang
- Ular
- Babi Hutan

Apa yang harus dilakukan :

- Menghindari jurang, ular dan babi hutan,
- Mendapatkan makanan untuk menambah stamina
- Mencapai camp dipos 3

Scoring :

Setelah sampai di camp 3, *player* akan disuguhkan result yang berisi *Score*. Di pos 3 ini, *player* menginap beristirahat semalam untuk pendakian esok harinya.

4. Skenario : *Level* 4 Pendakian

Setting : Gunung Sangarra – H02.

Intro :

Player melanjutkan perjalanan dari pos 3.

Misi :

- Mencapai puncak dengan selamat

Rintangan :

- Ular
- Babi Hutan
- Jurang

Apa yang harus dilakukan :

- Menghindari rintangan jurang, ular dan babi hutan,
- Mendapatkan makanan untuk menambah stamina
- Mencapai Puncak

Scoring :

Setelah sampai di Puncak, *player* akan disuguhkan result yang berisi *Score*.

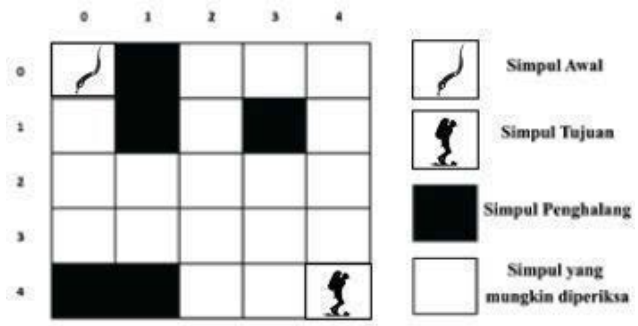
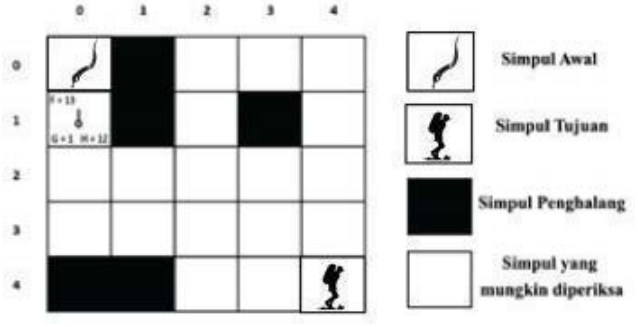
3.1.4 Analisis Algoritma Yang Diterapkan

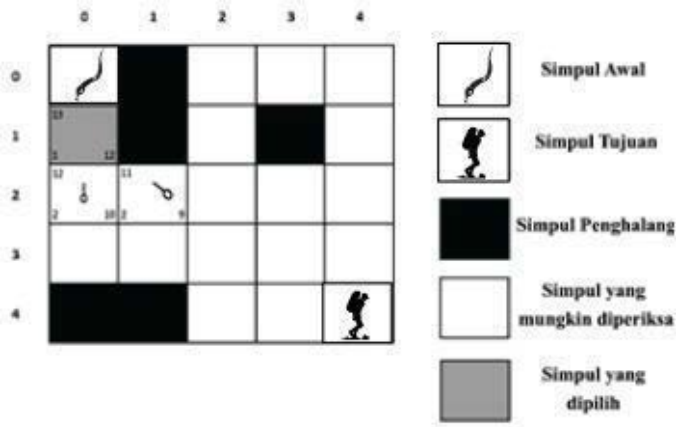
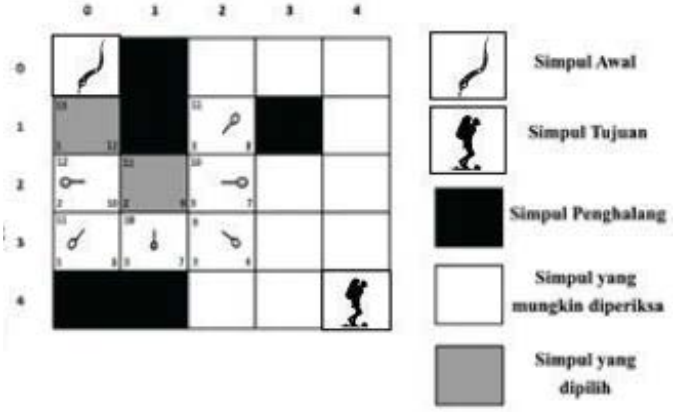
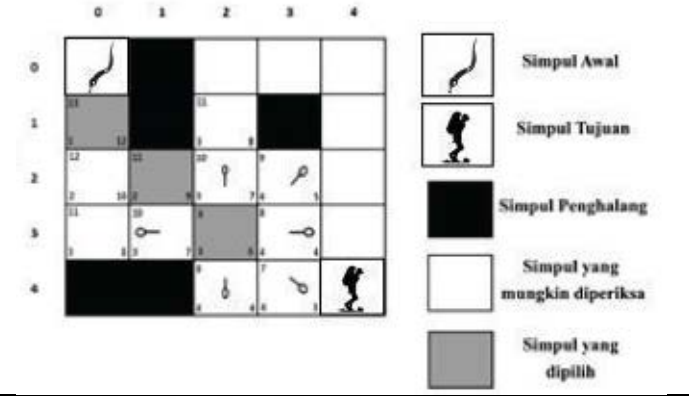
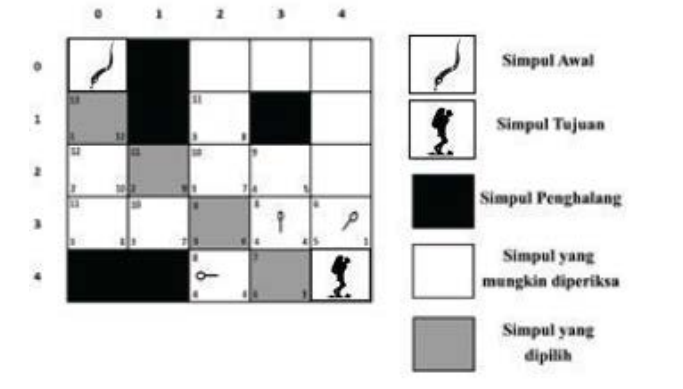
Adapun Algoritma yang diterapkan pada *game* ini adalah algoritma A* yang akan diterapkan pada musuh. Algoritma A* akan melakukan penelusuran pada semua cabang yang dibuka dari simpul induknya dan membandingkan cabang – cabang tersebut berdasarkan nilai terkecil yang akan dipilih.

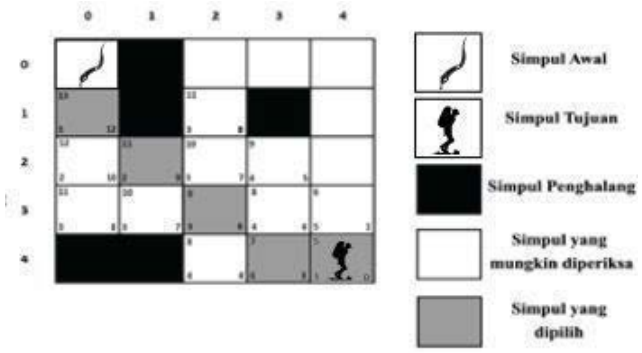
Perhitungan bobot nilai terkecil berdasarkan, g(n) merepresentasikan harga jalan dari simpul awal ke simpul n, dan h(n) merepresentasikan perkiraan harga dari simpul n ke simpul tujuan. Selama iterasi, algoritma A* mencari simpul n yang memiliki nilai f(n) = g(n) + h(n) yang paling kecil.

Penggunaan algoritma A* akan disimulasikan dengan menggunakan array 2 dimensi matriks ordo 5x5. Pencarian jalur terpendek musuh babi ke *player* dengan menggunakan algoritma A* dari simpul awal menuju simpul tujuan akan diperlihatkan pada Tabel di bawah ini.

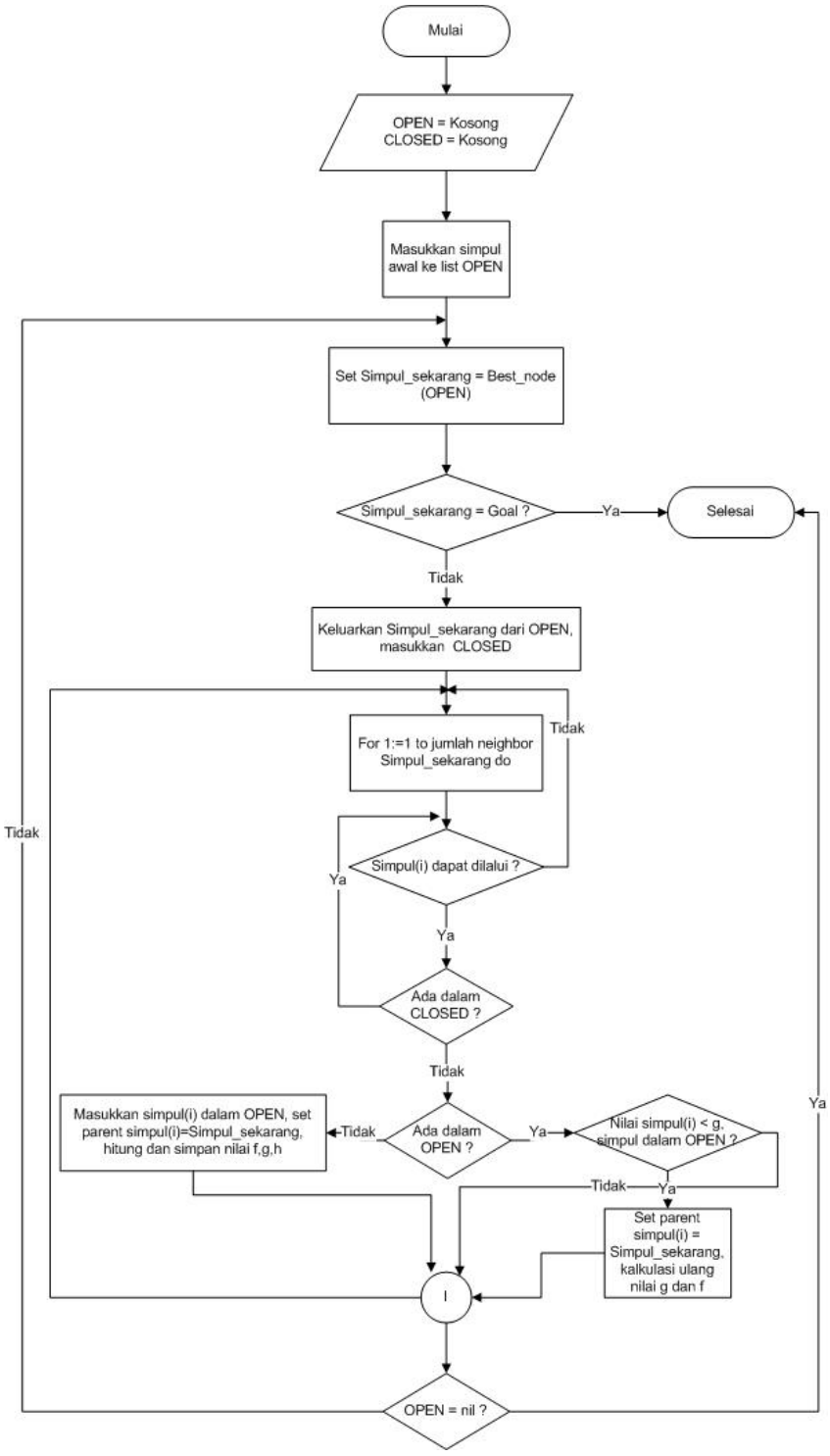
Tabel 3. 4 Skenario Langkah Pencarian Algoritma A*

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 cabang dari (0,0), lalu hitung semua nilai F dari simpul tersebut. Karena dua cabang (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 cabang 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 cabang 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 cabang dari simpul (1,2), karena semua cabang belum pernah ada di *CLOSED*, maka cabang tersebut dimasukkan ke dalam *OPEN*. Lalu diperiksa dan dibandingkan semua nilai F dari cabang tersebut. Simpul (2,3) memiliki nilai F paling kecil dari semua cabang, maka simpul (2,3) dipilih sebagai *BestNode* dan dimasukkan ke dalam *CLOSED*. Kemudian *parent* diubah dari (1,2) menjadi (2,3).
	Bangkitkan semua cabang dari simpul (2,3). Kemudian masukkan semua cabang ke dalam *OPEN*. Lalu diperiksa dan dibandingkan nilai F dari cabang - cabang 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 cabang dari simpul (3,4). Masukkan semua cabang ke dalam *OPEN*. Periksa dan bandingkan nilai F dari semua cabang tersebut. Setelah diperiksa, simpul (4,4) memiliki nilai F paling kecil diantara semua cabang. 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 dihentikan dan solusi ditemukan. Langkah terpendek untuk mencapai tersebut dapat dilihat pada simulasi disamping.

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



Gambar 3. 3 Diagram *Flowchart* Algoritma A*

Secara detail pseudo code algoritma A* untuk musuh babi mencari *player* adalah sebagai berikut:

```
Algoritma A*
 {mencari node dengan nilai f terkecil}
Deklarasi
    level :Array
    openList, closedList :Dictionary
    node :PathNode
    curNode :PathNode
    Finished :Boolean

Deskripsi
     {inisialisasi list}
     openList ← null
     closedList ← null
     {masukkan simpul awal openList}
         openList ← node
     {menentukan perhitungan A* (A star)}
         curNode.g ← 0
         curNode.h ←Pathfinder.manhattanHeuristic (curNode,Finished)
         curF ← node.g + node.h
     while curNode != Finished do
         Search Level()
         for i ← 1 to level.length do
             openList ← level[row][col]
         if openList = curNode & level.obstacle == false then
                   g ← node.g + travelCost
                   h ← Pathfinder.manhattanHeuristic(level, Finished)
                   f ← g + h
         end if
         if Pathfinder.isOpen(level, openList) &&
             Pathfinder.isClosed( level, closedList) then
              if curF > f then
                        curF ← f
                        node.g ← g
                        node.h ← h
                        level.parentNode ← curNode
                        openList.parentNode (level)
              end if
         else
                        level.f ← f
                        level.g ← g
                        level.h ← h
                        level.parentNode ← curNode
                        openList.push(level)
         end if
         end for
       if openList.length = 0 then
             return null
       end if
      openList.sortOn('f', Array.NUMERIC)
      curNode ← openList.push()
     end while
     return Pathfinder.RetracePath(node)
```

Gambar 3. 4 Pseudecode Algoritma A*

3.2 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan sistem yang menitik-beratkan pada perilaku yang dimiliki oleh sistem sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan, diantaranya:

1. Analisis kebutuhan perangkat keras
2. Analisis kebutuhan perangkat lunak
3. Analisis pengguna sistem (*User*)

3.2.1 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.

Hardware (perangkat keras) yang dipakai dalam membangun aplikasi *game* Jejak Pendaki dengan spesifikasi yang digunakan sebagai berikut:

a) Processor 1.83 GHz
b) Memori 4 GB RAM DDR2
c) Hard disk 500 GB
d) VGA 256 MB

Adapun kebutuhan minimal *hardware* (perangkat keras) digunakan untuk menjalankan aplikasi ini sebagai berikut:

1. Processor 1.3 Ghz
2. RAM 512 MB
3. Space Harddisk 75 MB
4. Kartu grafik dengan DirectX 9 compatible sound card
5. CRT monitor dengan resolusi 1024 x 768

3.2.2 Analisis Kebutuhan Perangkat Lunak

Selain kebutuhan *hardware*, dibutuhkan juga kebutuhan *software* untuk melakukan membangun aplikasi ini. Kebutuhan *software* yang diperlukan sebagai berikut:

1. Sistem Operasi Windows 7
   Sistem operasi yang digunakan dalam pembuatan tugas akhir ini.
2. Adobe Flash Pro CS5.5
   Tools pengembang bahasa pemrograman Actionscipt.
3. Adobe Photoshop CS5.5
   Tools untuk editing gambar bitmap.
4. Adobe Illustrator CS5.1
   Tools untuk editing gambar vektor.

Sedangkan perangkat lunak yang dibutuhkan pengguna untuk menjalankan aplikasi *game* ini adalah *flash Player 9.0 atau versi yang terbaru*

3.2.3 Analisis Pengguna Sistem (*User*)

Analisis pengguna dimaksudkan untuk mengetahui siapa saja pengguna yang terlibat dalam proses penggunaan *game Jejak Pendaki* sehingga dapat diketahui hal –hal yang berkaitan dengan pengguna.

Pengguna sistem aplikasi ini yaitu masyarakat yang ingin mengetahui informasi mengenai pendakian gunung. Dalam menggunakan sistem ini, pengguna diharuskan mengunakan *platform desktop* yang terinstall *flash player* untuk bisa menggunakan aplikasi ini. Berikut ini beberapa parameter terhadap calon user dari aplikasi :

1. *User Knowledge and Experience* dari target user yang akan menggunakan *game*.

Tabel 3. 5 Analisis *User Knowledge and Experience*

Educational Level	*Reading Level*	*Typing Skills*
Siswa pelajar tingkat atas (SMA sederajat) ke atas	Lancar baca-tulis	Tidak memerlukan *typing skill* yang tinggi

Computer Literacy	*Task Experience*	*System Experience*
Pemain memiliki kemampuan komputer yang *moderate* (menengah)	Pemain memiliki pengalaman penggunaan komputer dan *game* sedang	Pemain dapat menjalankan sistem sesuai dengan pengalaman komputer dengan *level* sedang
Application Experience	*Native Language*	*Use Of Other Software*
Pemain dapat menggunakan sistem operasi Windows xp/ 7	Pemain mengerti bahasa Indonesia	Pemain memiliki aplikasi pendukung flash *player*

2. *User Physical Characteristic*

   Keadaan fisik seseorang akan berpengaruh pada penggunaan aplikasi *game* ini. Ada hal – hal yang harus diperhatikan terhadap karakteristik fisik user yaitu, umur, buta warna, jenis kelamin dan penggunaan tangan.

Tabel 3. 6 Analisis *User Physical Characteristic*

Umur	15 tahun ke atas
Jenis Kelamin	Laki – laki dan perempuan
Penggunaan Tangan	Kanan dan kiri
Buta Warna	Pemain dapat membedakan warna yang satu dengan lainnya, karena pada *game* ini terdapat warna – warna yang fungsinya untuk membedakan antar fungsional

3.3 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 serta sesuai dengan kebutuhan. Analisis kebutuhan fungsional ini meliputi analisis kebutuhan sistem, analisis kebutuhan konten, dan pemodelan sistem.

3.3.1 Analisis Kebutuhan Sistem

Dalam pembangunan aplikasi perancangan sistem, sebelumnya dilakukan analisis kebutuhan sistem agar sistem yang dibangun sesuai dengan yang diharapkan atau dibutuhkan pengguna. Adapun kebutuhan sistem yang diperlukan antara lain:

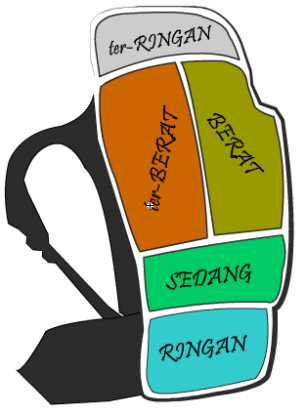
1. Menampilkan informasi tentang perencanaan dan aktivitas pendakian gunung yang di visualisasikan dalam bentuk *game*
2. Informasi mengenai beberapa gunung di Indonesia yang divisualisasikan dalam bentuk *game*.

3.3.2 Analisis Kebutuhan Konten

Analisis ini bertujuan untuk menganalisis konten yang diperlukan agar memudahkan perancangan informasi setelah sistem yang dibutuhkan telah diketahui. Adapun konten yang ditampilkan berupa konten teks, foto, dan animasi yang berisi karakter – karakter dan storyboard.

Tabel 3. 7 Analisis Kebutuhan Konten

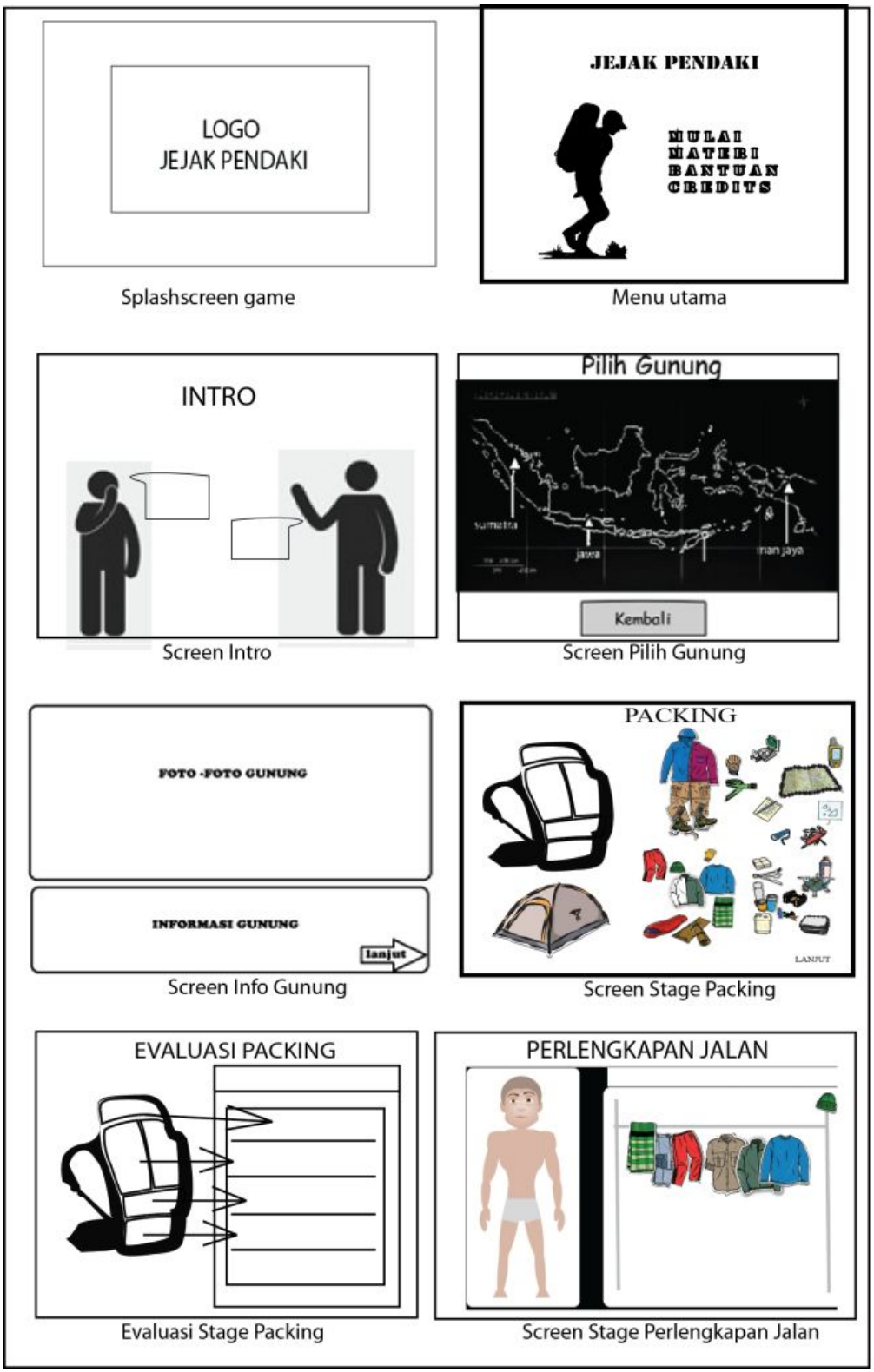
No.	Nama	Gambar	Keterangan
1	Jimbrow		Jimbrow merupakan tokoh utama dan karakter yang dapat dikontrol oleh pemain dengan keyboard kanan, kiri, atas, bawah untuk berjalan dan keyboard A untuk melempar pisau. Jimbrow mempunyai 1000 stamina bar dan akan berkurang apabila terus bergerak dan terkena binatang.

2.	Ito		Ito merupakan karakter NPC yang akan menjadi teman jimbrow dalam merencanakan pendakian.
3.	Ransel / Carrier		Konten ini harus disi oleh pemain dengan mendrag barang – barang ke dalam target ransel
4.	Body target		Konten ini harus disi oleh pemain dengan mendrag pakaian – pakaian ke body target yang benar
5.	Ular		Ular merupakan karakter jahat yang menjadi rintangan dalam menyelesaikan stage pendakian dan stamina bar jimbrow akan berkurang jika terkena ular

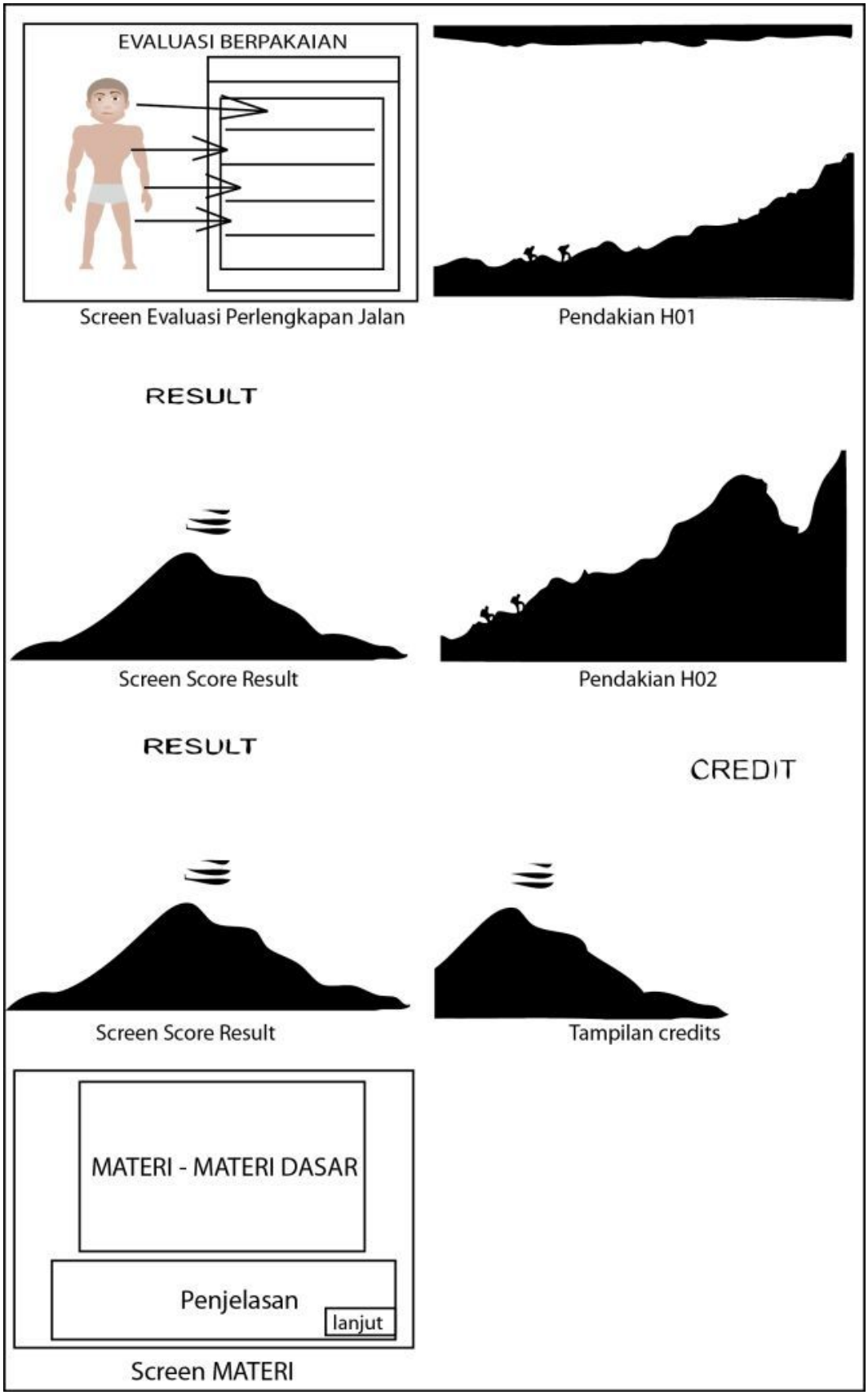
6.	Babi		Babi merupakan karakter jahat yang menjadi rintangan dalam menyelesaikan stage pendakian dan stamina bar jimbrow akan berkurang 50 jika terkena babi

Storyboard

Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan alur skenario. *Storyboard* dapat menyampaikan ide cerita kepada orang lain dengan lebih mudah, sehingga menghasilkan persepsi yang sama mengenai ide cerita kepada seseorang hanya dengan melihat *storyboard.* Berikut merupakan *storyboard* dari *game* Jejak Pendaki :



Gambar 3. 5 *Storyboard* 1 *game* Jejak Pendaki



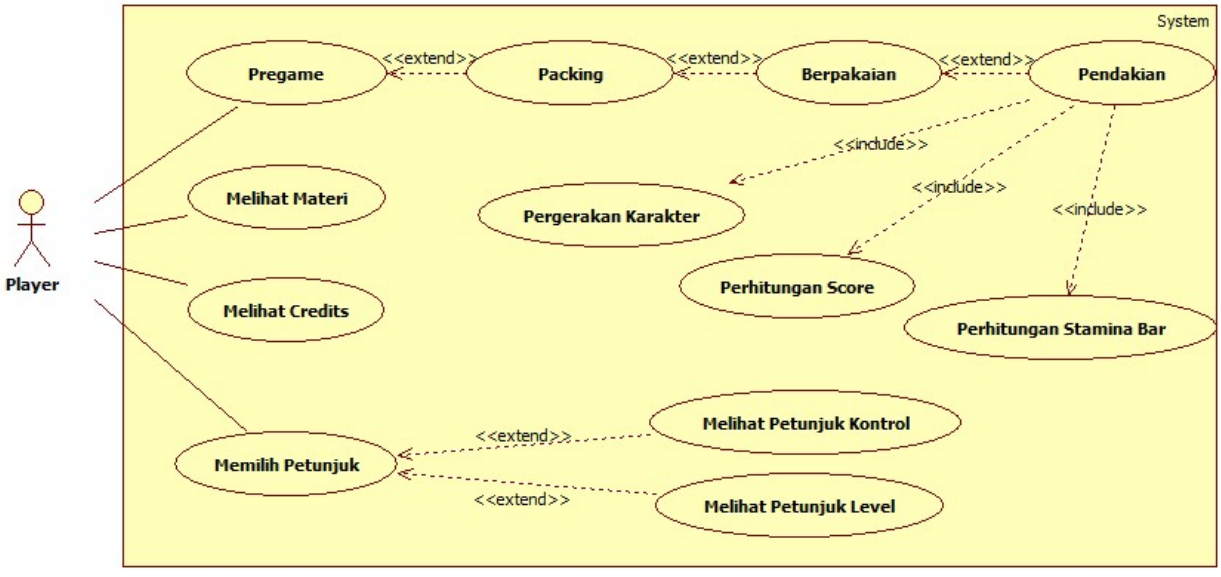
Gambar 3. 6 ***Storyboard*** **2** ***game*** **Jejak Pendaki**

3.3.3 Pemodelan Sistem

Pemodelan sistem berfungsi untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem pada aplikasi serta menentukan kelas yang dibutuhkan untuk realisasi fungsi-fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram.

1. **Diagram Use Case**

   Digunakan untuk memodelkan atau menggambarkan batasan sistem dan fungsi-fungsi utamanya. Mendiskripsikan fungsi dari sebuah sistem dari perspektif pengguna, *use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara pengguna sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Berikut gambaran diagram use case pada *game* jejak pendaki :



Gambar 3. 7 Diagram Use Case *Game* Jejak Pendaki

1.1 Definisi Aktor

Definisi Aktor berfungsi untuk menjelaskan peran dari tiap aktor yang terlibat dalam sistem. Berikut definisi aktor dibawah ini :

Tabel 3. 8 Tabel definisi aktor

No.	Aktor	Deskripsi
1	*Player*	Aktor dengan *role* ini memliki wewenang untuk menampilkan *mainscreen* dengan menjalankan aplikasi

1.2 Definisi Use Case

Definisi *use case* berfungsi untuk menjelaskan proses yang terdapat pada setiap *use case*. Berikut definisi *use case* di tunjukan pada tabel dibawah ini :

Tabel 3. 9 Tabel definisi usecase

No.	Use Case	Deskripsi
1	Pendakian	*Player* memulai permainan pada stage Pendakian
2	Pre*game*	*Player* dapat memilih dan melihat info gunung yang akan didaki
3	*Packing*	*Player* memulai permainan pada stage *Packing*
4	Berpakaian	*Player* memulai permainan pada stage Berpakaian
5	Perhitungan Stamina Bar	Untuk fungsi menghitung Stamina Bar
6	Perhitungan *Score*	Untuk fungsi menghitung *Score*
7	Pergerakan Karakter	Untuk fungsi pergerakan karakter
8	Melihat Materi	*Player* memilih menu Materi untuk melihat materi – materi
9	Melihat *Credits*	*Player* memilih menu *Credits* untuk melihat *credits*
10	Memiilih Petunjuk	*Player* memilih menu Petunjuk
11.	Melihat Petunjuk Kontrol	*Player* memililh menu Petunjuk Kontrol untuk melihat Petunjuk Kontrol
12.	Melihat Petunjuk *Level*	*Player* memilih menu Petunjuk *Level* untuk melihat Petunjuk *Level*

1.3 Skenario Use Case

Skenario Use case merupakan sekenario dari setiap bagian pada *use case* menunjukkan proses apa yang terjadi pada setiap bagian didalam *use case* tersebut, dimana *user* memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh sistem kepada *user* setelah *user* memberikan perintah pada setiap bagian – bagian *use case*. Berikut ini use case scenario pada tiap usecase :

Tabel 3. 10 Tabel Skenario Usecase Pendakian

Identifikasi		
Nama	Pendakian	
Tujuan	Memulai permainan pada stage pendakian	
Aktor	*Player*	
Skenario utama		
Kondisi awal	Tampilan stage pendakian	
Aksi aktor		Reaksi Sistem
1. Memilih siap pendakian		
		2. Memasukkan karakter *player* ke *stage* Pendakian
		3. Memunculkan musuh
		4. Apabila *player* mati, maka menampilkan screen *game over*
5. Jika menekan tombol navigasi kiri, kanan, atas dan bawah		
		6. Menggerakkan karakter
7. Jika menekan tombol *keyboard* A		
		8. Melempar pisau
		9. Apabila musuh terkena pisau dan mati, maka skor ditambahkan
10. Mencapai camp atau puncak gunung		
		11. Menghapus musuh
		12. Menampilkan *win screen*
		13. Memasukkan *player* ke *level* selanjutnya
Kondisi akhir	Menampilkan *win screen*	

Tabel 3. 11 Tabel Skenario Usecase *Pregame*

Identifikasi		
Nama	*Pregame*	
Tujuan	Menampilkan info – info gunung	
Aktor	*Player*	
Skenario utama : Info Gunung dan lanjut		
Kondisi awal	Tampilan pilihan gunung	
Aksi aktor		Reaksi Sistem
1. Memilih gunung yang akan didaki		
		2. Menampilkan info-info Gunung yang dipilih
		3. Menampilkan pesan untuk lanjut ke stage *packing*
4. Memilih masuk ke stage *packing*		
Kondisi akhir	Tampilan stage *packing*	
Skenario alternatif : Info Gunung dan kembali		
Kondisi awal	Tampilan pilihan gunung	
Aksi aktor		Reaksi Sistem
1. Memilih gunung yang akan didaki		
		2. Menampilkan info-info Gunung yang dipilih
		3. Menampilkan pesan untuk lanjut ke stage *packing*
4. Memilih kembali ke pilihan gunug		
Kondisi akhir	Tampilan stage *packing*	

Tabel 3. 12 Tabel Skenario Usecase *Packing*

Identifikasi		
Nama	*Packing*	
Tujuan	Melakukan permainan di stage *packing*	
Aktor	*Player*	
Skenario utama : Objek benar		
Kondisi awal	*Player* memulai permainan	
Aksi aktor		Reaksi Sistem
1. Memindahkan objek – objek (drag drop)		
		2. Mengecek target objek yang dimaksud
		3. Objek berpindah ketempat target
		4. *Score* bertambah
Kondisi akhir	Tampilan screen evaluasi *packing*	
Skenario alternatif : Objek salah		
Kondisi awal	*Player* memulai permainan	
Aksi aktor		Reaksi Sistem
1. Memindahkan objek – objek (drag drop)		
		2. Mengecek target objek yang dimaksud
		3. Objek kembali ke tempat semula
		4. *Score* tetap
Kondisi akhir	*Player* masih distage *packing*	

Tabel 3. 13 Tabel Skenario Usecase Berpakaian

Identifikasi		
Nama	Berpakaian	
Tujuan	Melakukan permainan di stage berpakaian	
Deskripsi	Proses melakukan permainan di stage berpakaian	
Aktor	*Player*	
Skenario utama : Objek benar		
Kondisi awal	*Player* memulai permainan	
Aksi aktor		Reaksi Sistem
1. Memindahkan objek – objek (drag drop)		
		2. Mengecek target objek yang dimaksud
		3. Objek berpindah ketempat target
		4. *Score* bertambah

Kondisi akhir	Tampilan screen evaluasi berpakaian
Skenario alternatif : Objek salah	
Kondisi awal	*Player* memulai permainan
Aksi aktor	Reaksi Sistem
1. Memindahkan objek – objek (drag drop)	
	2. Mengecek target objek yang dimaksud
	3. Objek kembali ke tempat semula
	4. *Score* tetap
Kondisi akhir	*Player* masih distage berpakaian

Tabel 3. 14 Tabel Skenario Usecase *Stamina Bar*

Identifikasi	
Nama	Perhitungan Stamina Bar
Tujuan	Untuk dapat menghitung HP karakter
Aktor	*Player*
Skenario Utama	
Kondisi awal	Permainan dimulai
Aksi Aktor	Reaksi Sistem
	1. Menentukan darah karakter sebesar 1000
2. Menebas	
	3. Apabila mengenai musuh, musuh mati dan menghilang
4. Menyentuh musuh	
	5. Darah pemain berkurang 50
Kondisi akhir	Hasil perhitungan Stamina Bar

Tabel 3. 15 Tabel Skenario Usecase Perhitungan *Score*

Identifikasi	
Nama	Perhitungan *Score*
Tujuan	Untuk dapat menghitung *Score* yang didapatkan *player*
Aktor	*Player*
Skenario Utama	
Kondisi awal	Permainan dimulai
Aksi Aktor	Reaksi Sistem
	1. Menentukan skor awal sebesar 0
2. Membunuh musuh	
	3. Menambah skor sebesar 10 pada

	skor permainan
4. Meletakkan *packing* barang dengan benar	
	5. Menambah skor sebesar 100 pada skor
6. Memakai item pakaian yang tepat	
	7. Menambah skor sebesar 100 pada pada skor
	8. Pada setiap akhir *level* permainan menjumlahkan semua skor yang didapat dari skor permainan
Kondisi akhir	Hasil perhitungan skor

Tabel 3. 16 Tabel Skenario Usecase Pergerakan Karakter

Identifikasi	
Nama	Pergerakan Karakter
Tujuan	Untuk melakukan pergerakan karakter
Aktor	*Player*
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 bergerak keatas
7. Jika pemain menekan tombol bawah pada *keyboard*	
	8. Karakter bergerak kebawah
9. Jika pemain menekan tombol “A” pada *keyboard*	
	10. Karakter melempar pisau
Kondisi akhir	Pemain melakukan pergerakan karakter

Tabel 3. 17 Tabel Skenario Usecase Memilih Petunjuk

Identifikasi		
Nama	Memilih Petunjuk	
Tujuan	Menampilkan menu petunjuk	
Aktor	*Player*	
Skenario utama		
Kondisi awal	Tampilan *Mainscreen*	
Aksi aktor		Reaksi Sistem
1. Memilih menu Petunjuk		
		2. Menampilkan menu yang ada di menu petunjuk (petunjuk kontrol dan petunjuk *level*)
Kondisi akhir	Menampilkan menu petunjuk	

Tabel 3. 18 Tabel Skenario Usecase Melihat Materi

Identifikasi		
Nama	Melihat Materi	
Tujuan	Tampilkan materi – materi	
Aktor	*Player*	
Skenario utama		
Kondisi awal	Tampilan menu utama	
Aksi aktor		Reaksi Sistem
1. Memilih menu materi		
		2. Menampilkan materi – materi dasar
Kondisi akhir	Tampilan materi – materi	

Tabel 3. 19 Tabel Skenario Usecase Melihat *Credits*

Identifikasi	
Nama	Melihat *Credits*
Tujuan	Menampilkan *credits game*
Aktor	*Player*
Skenario utama	
Kondisi awal	Tampilan menu utama

Aksi aktor		Reaksi Sistem
1. Memilih menu *credits*		
		2. Menampilkan *credits game*
Kondisi akhir	Tampilan *credits game*	

Tabel 3. 20 Tabel Skenario Usecase Petunjuk Kontrol

Identifikasi		
Nama	Melihat Petunjuk Kontrol	
Tujuan	Menampilkan Petunjuk Kontrol	
Aktor	*Player*	
Skenario utama		
Kondisi awal	Tampilan menu petunjuk	
Aksi aktor		Reaksi Sistem
1. *Player* memilih menu petunjuk control		
		2. Menampilkan petunjuk kontrol
Kondisi akhir	Tampilan petunjuk kontrol	

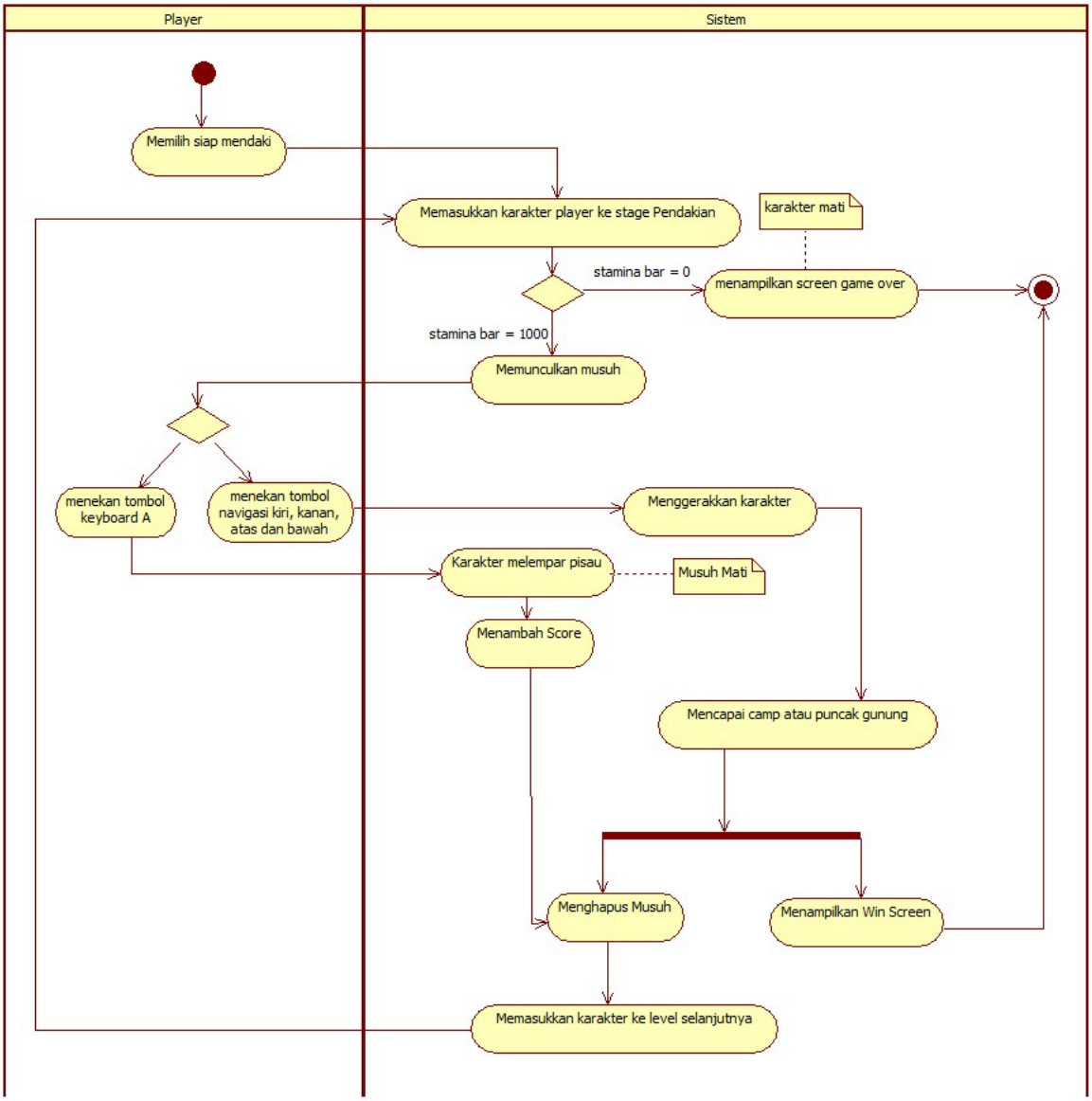
Tabel 3. 21 Tabel Skenario Usecase Petunjuk *Level*

Identifikasi		
Nama	Melihat Petunjuk *Level*	
Tujuan	Menampilkan petunjuk *level*	
Aktor	*Player*	
Skenario utama		
Kondisi awal	Tampilan menu petunjuk	
Aksi aktor		Reaksi Sistem
1. *Player* memilih menu petunjuk *level*		
		2. Menampilkan petunjuk *level*
Kondisi akhir	Tampilan petunjuk *level*	

2. Diagram Aktifitas

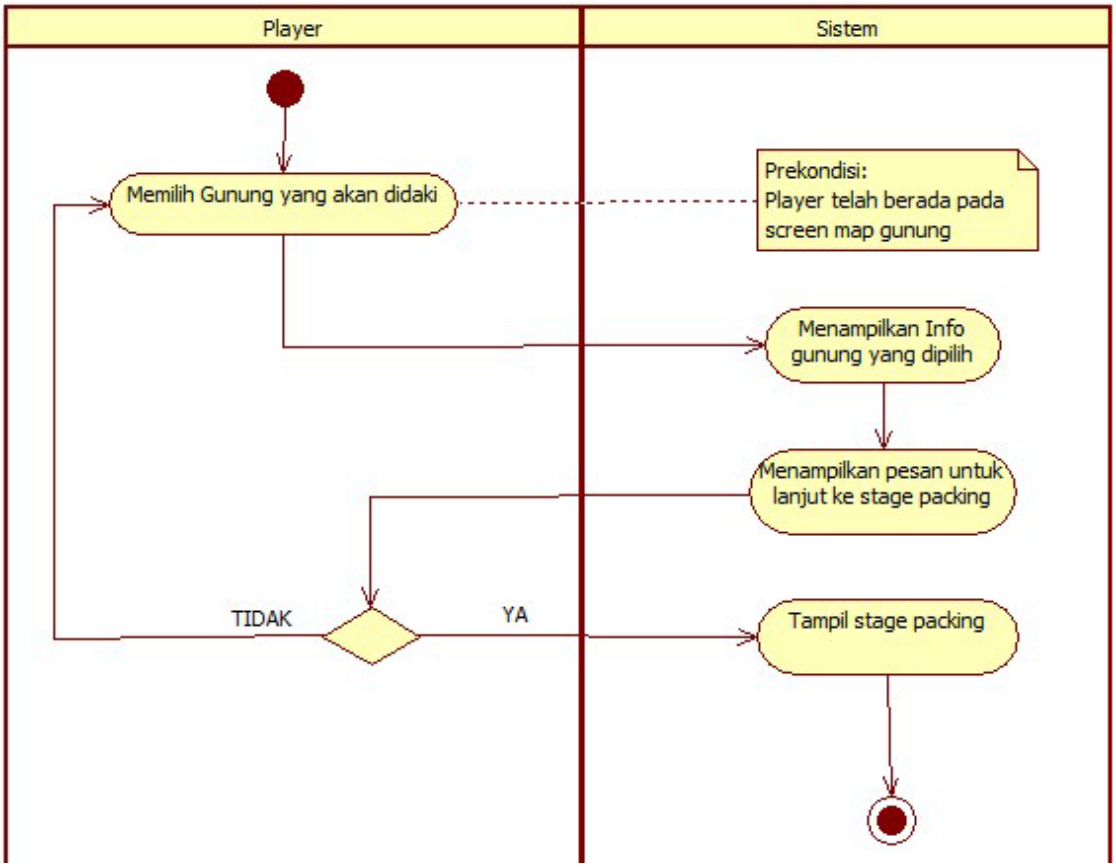
Diagram aktifitas lebih memfokuskan diri pada eksekusi dan alur sistem daripada bagaimana sistem itu dirakit. Diagram ini menunjukkan aktivitas sistem dalam bentuk kumpulan aksi – aksi. Berikut gambaran diagram aktifitas pada *game* jejak pendaki :

2.1. Diagram Aktifitas Pendakian



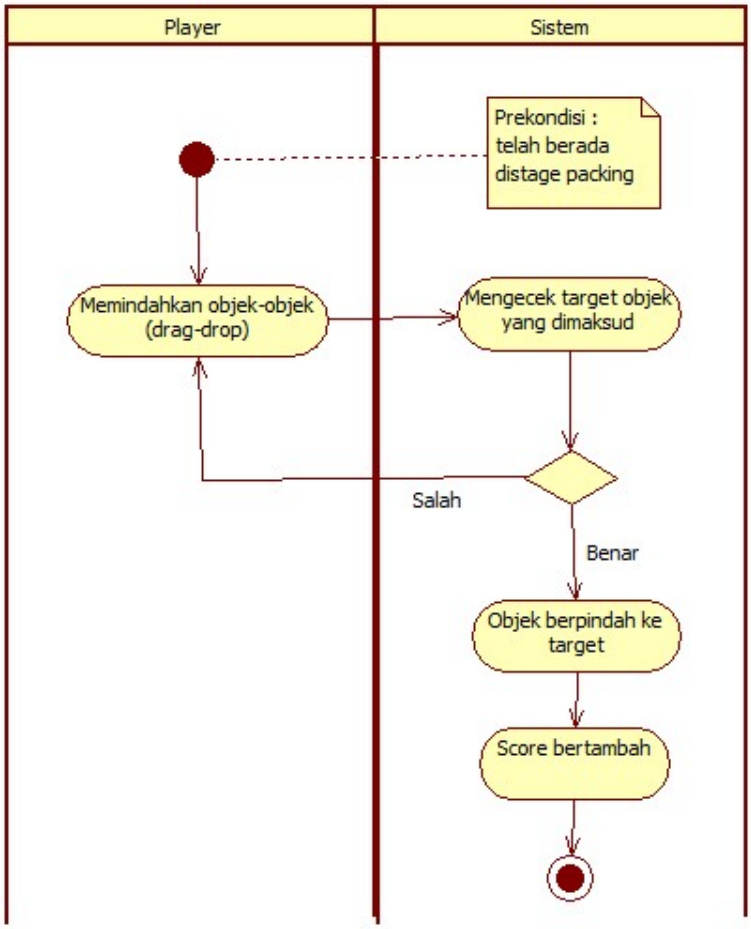
Gambar 3. 8 Diagram Aktifitas Pendakian

2.2. Diagram Aktifitas Pre*game*



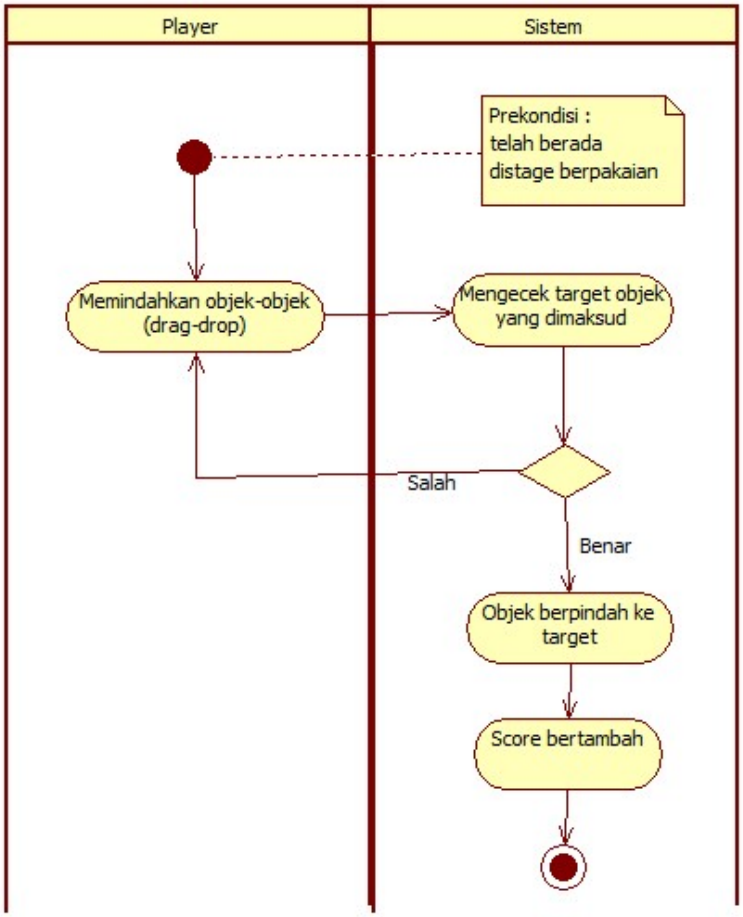
Gambar 3. 9 Diagram Aktifitas Pre*game*

2.3. Diagram Aktifitas *Packing*



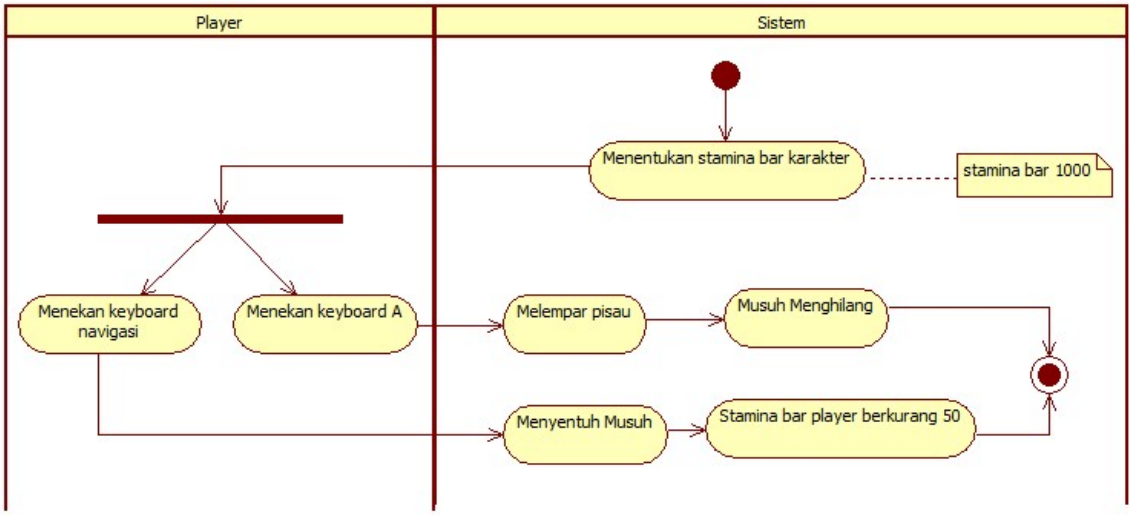
Gambar 3. 10 Diagram Aktifitas *Packing*

2.4. Diagram Aktifitas Berpakaian



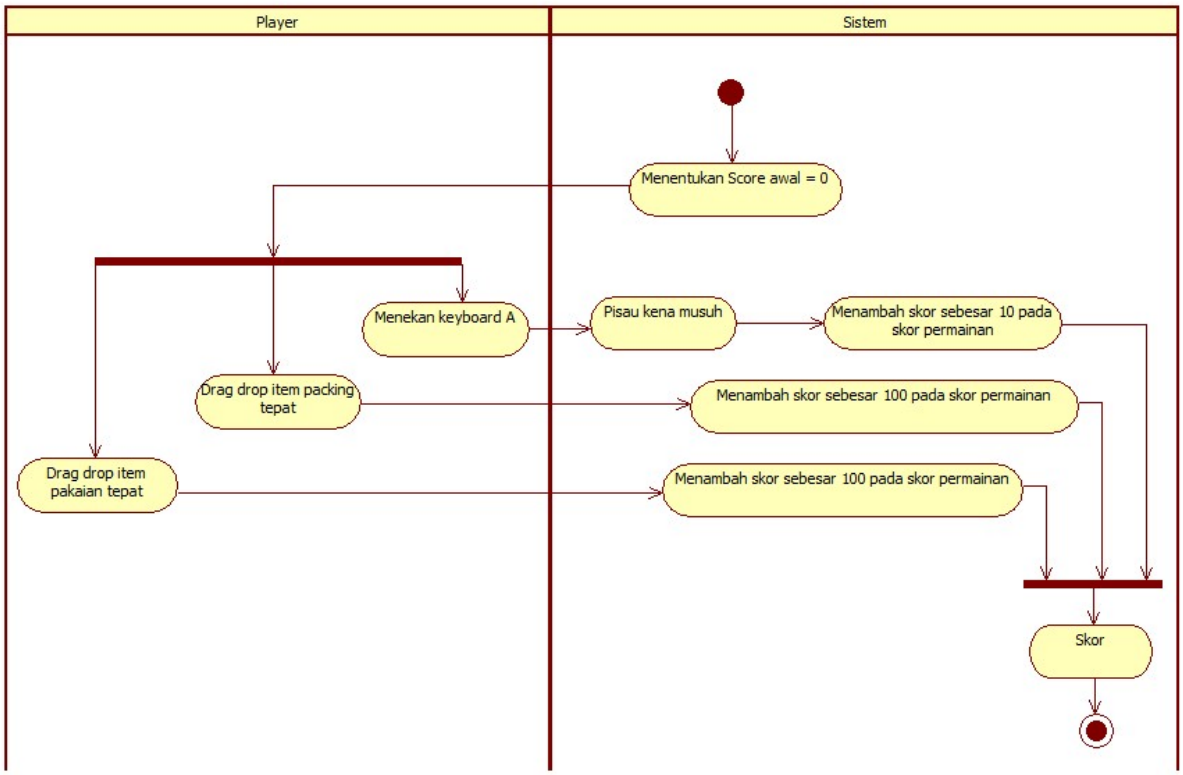
Gambar 3. 11 Diagram Aktifitas Berpakaian

2.5. Diagram Aktifitas Perhitungan *Stamina Bar*



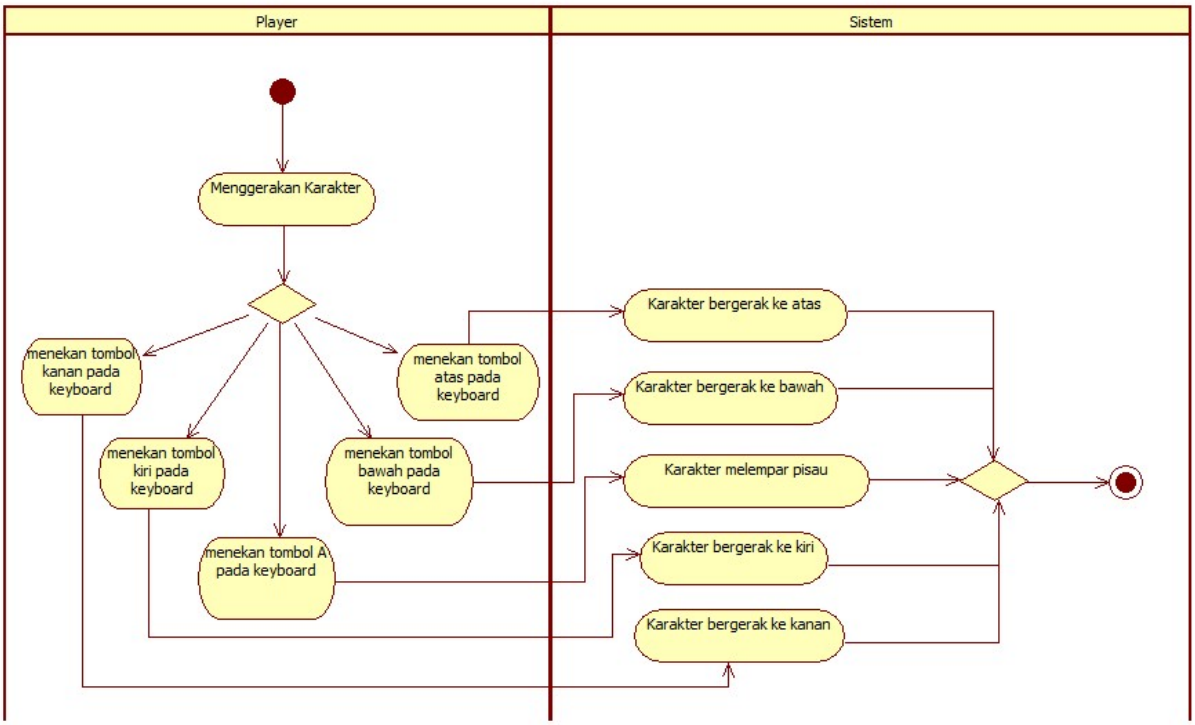
Gambar 3. 12 Diagram Aktifitas Perhitungan Stamina Bar

2.6. Diagram Aktifitas Perhitungan *Score*



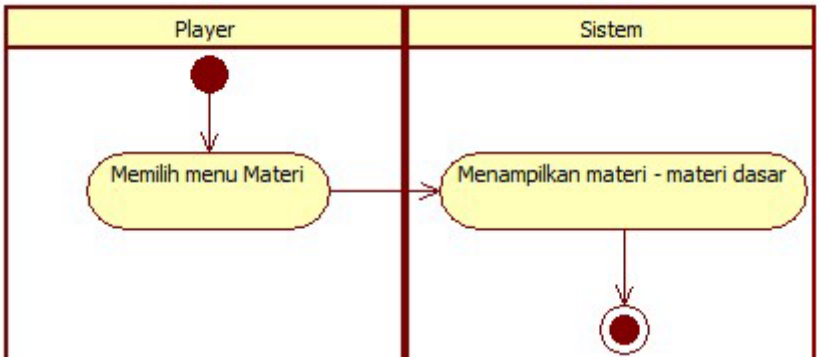
Gambar 3. 13 Diagram Aktifitas Perhitungan *Score*

2.7. Diagram Aktifitas Pergerakan Karakter



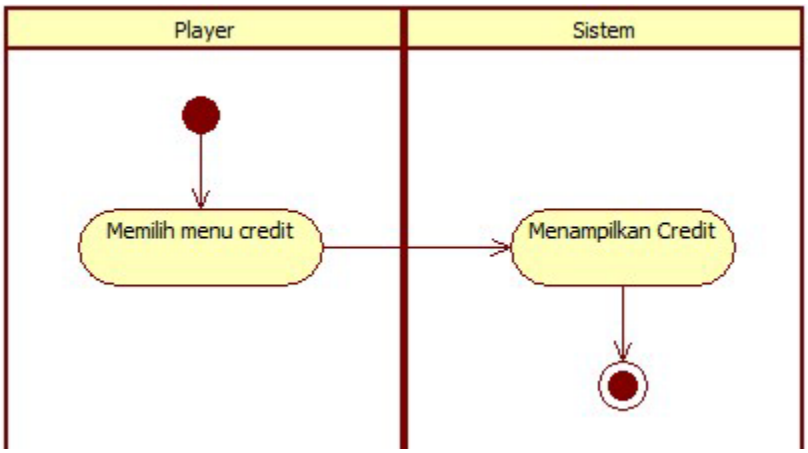
Gambar 3. 14 Diagram Aktifitas Pergerakan Karakter

2.8. Diagram Aktifitas Melihat Materi



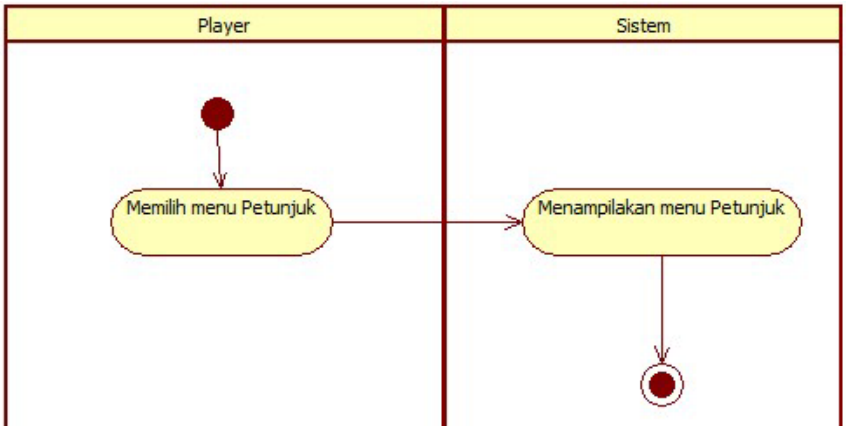
Gambar 3. 15 Diagram Aktifitas Melihat Materi

2.9. Diagram Aktifitas Melihat *Credits*



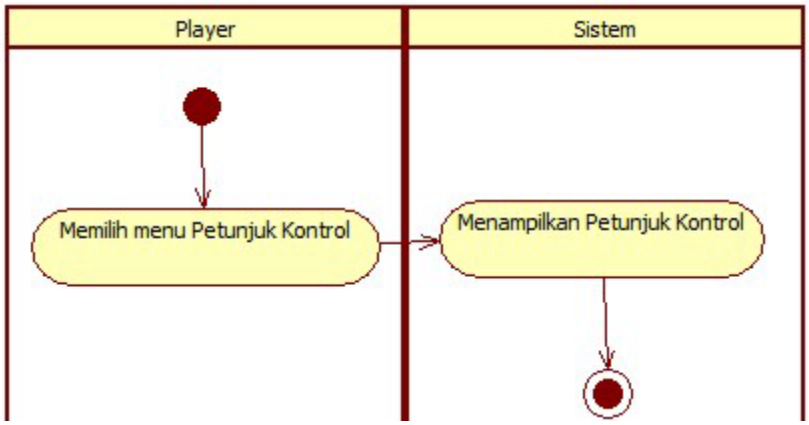
Gambar 3. 16 Diagram Aktifitas Melihat *Credits*

2.10. Diagram Aktifitas Memilih Petunjuk



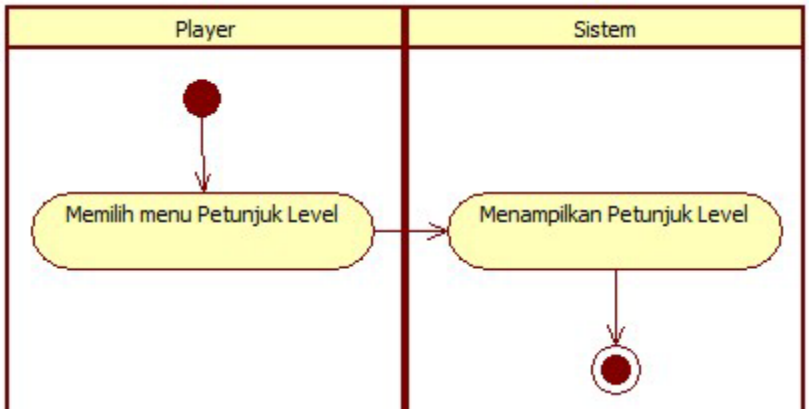
Gambar 3. 17 Diagram Aktifitas Memilih Petunjuk

2.11. Diagram Aktifitas Melihat Petunjuk Kontrol



Gambar 3. 18 Diagram Aktifitas Melihat Petunjuk Kontrol

2.12. Diagram Aktifitas Melihat Petunjuk *Level*

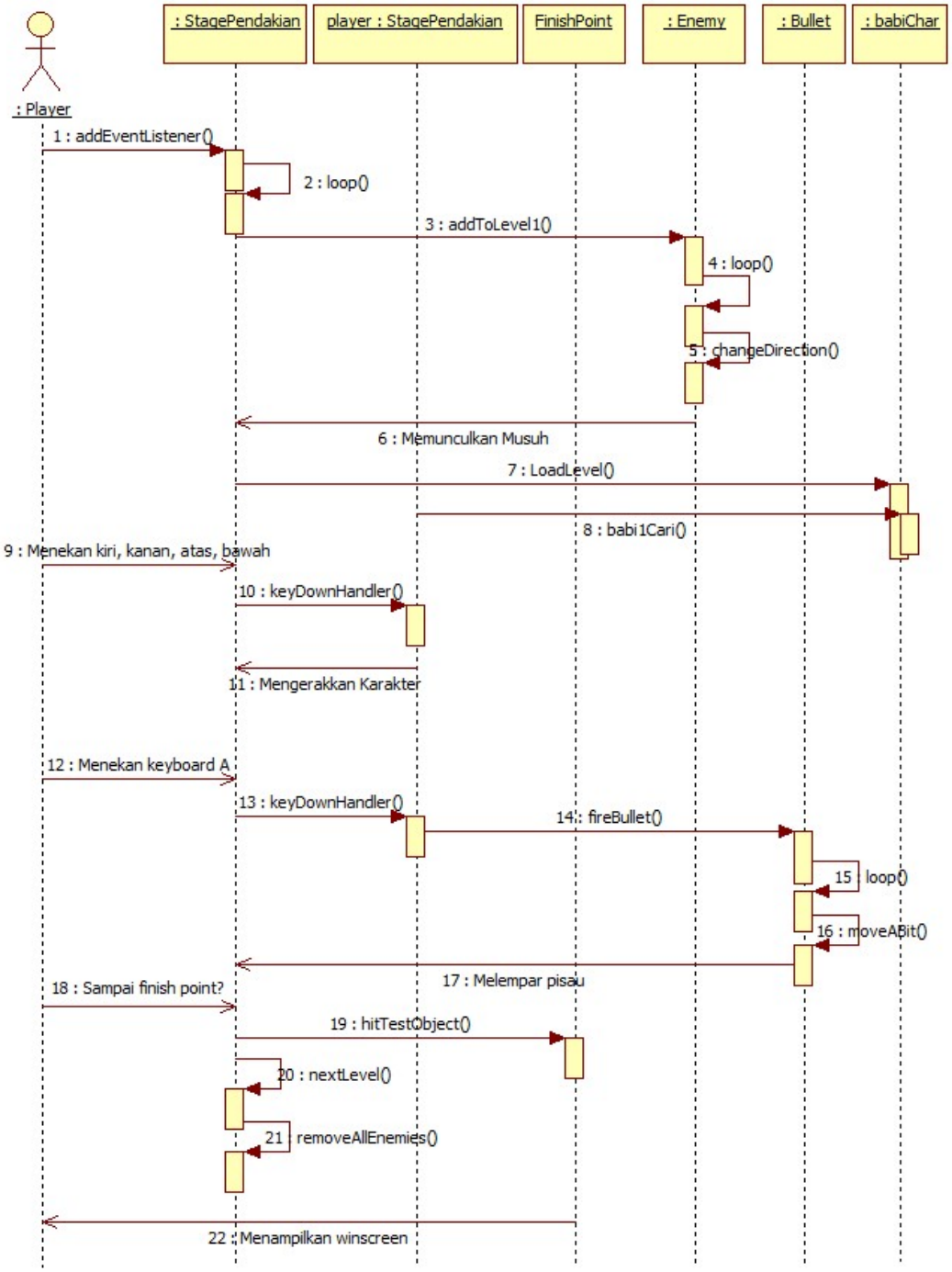


Gambar 3. 19 Diagram Aktifitas Melihat Petunjuk *Level*

3. Diagram *Sequence*

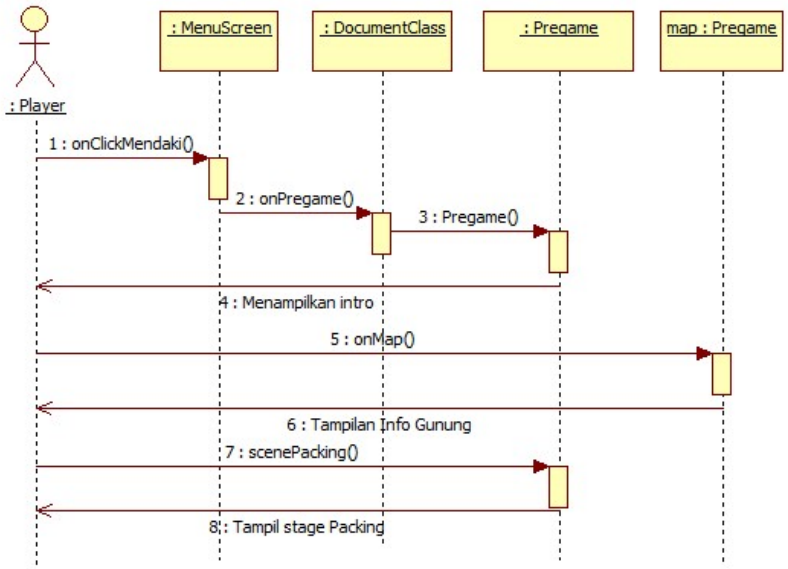
Diagram *sequence* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Diagram *Sequence* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Berikut ini beberapa diagram *sequence* pada *game* jejak pendaki :

3.1. Diagram *Sequence* Pendakian



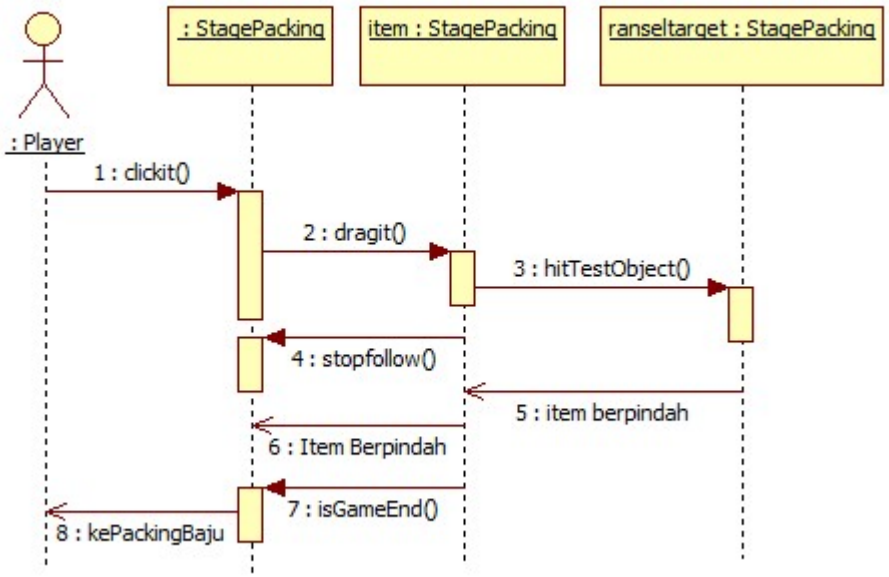
Gambar 3. 20 Diagram *Sequence* Pendakian

3.2. Diagram *Sequence* Pre*game*



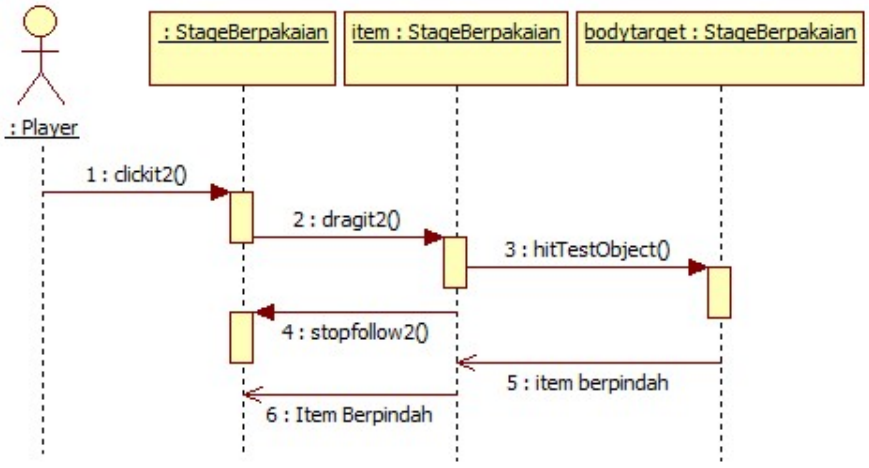
Gambar 3. 21 Diagram *Sequence Pregame*

3.3. Diagram *Sequence Packing*



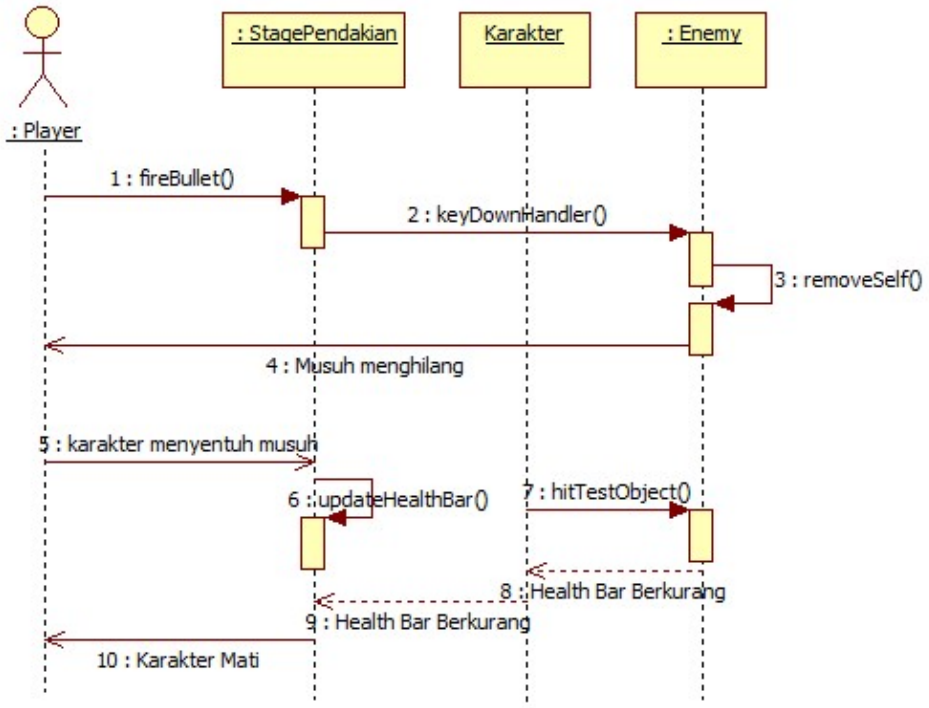
Gambar 3. 22 Diagram *Sequence Packing*

3.4. Diagram *Sequence* Berpakaian



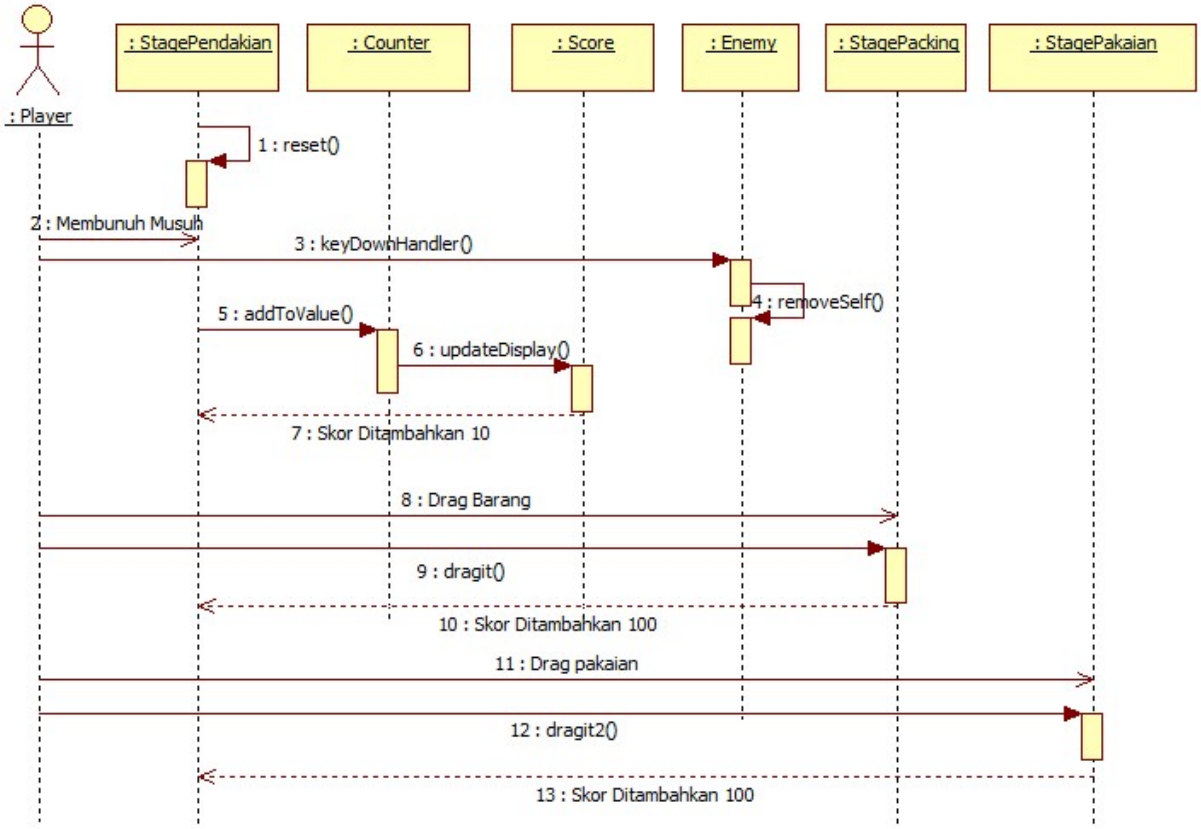
Gambar 3. 23 Diagram *Sequence* Berpakaian

3.5. Diagram *Sequence* Perhitungan Stamina Bar



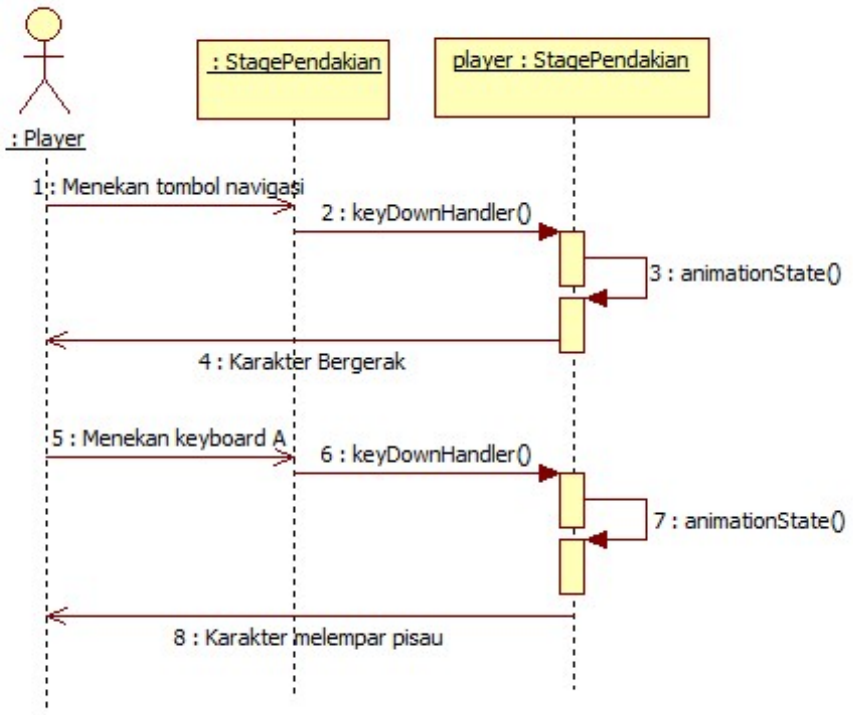
Gambar 3. 24 Diagram *Sequence* Perhitungan Stamina Bar

3.6. Diagram *Sequence* Perhitungan *Score*



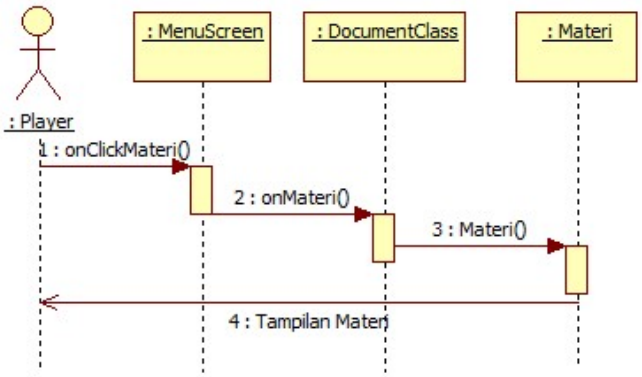
Gambar 3. 25 Diagram *Sequence* Perhitungan Score

3.7. Diagram *Sequence* Pergerakkan Karakter



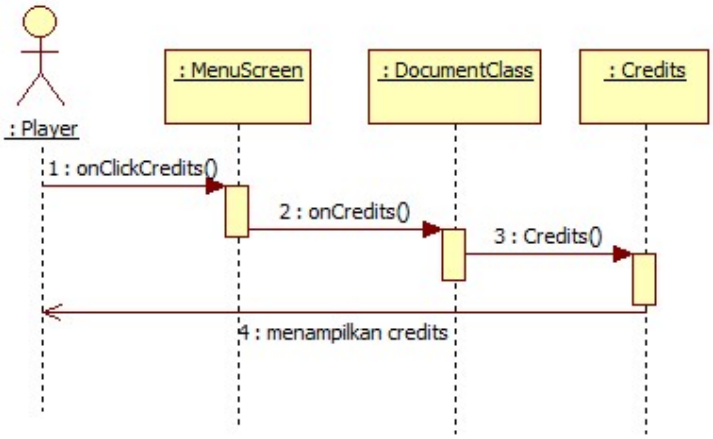
Gambar 3. 26 Diagram *Sequence* Pergerakkan Karakter

3.8. Diagram *Sequence* Melihat Materi



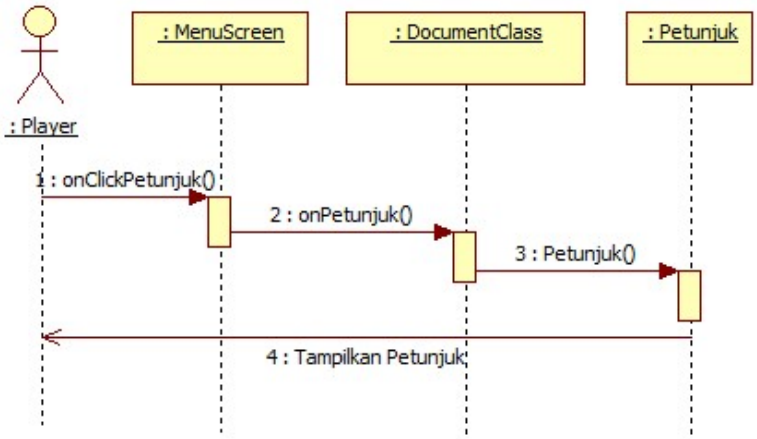
Gambar 3. 27 Diagram *Sequence* Melihat Materi

3.9. Diagram *Sequence* Melihat *Credits*



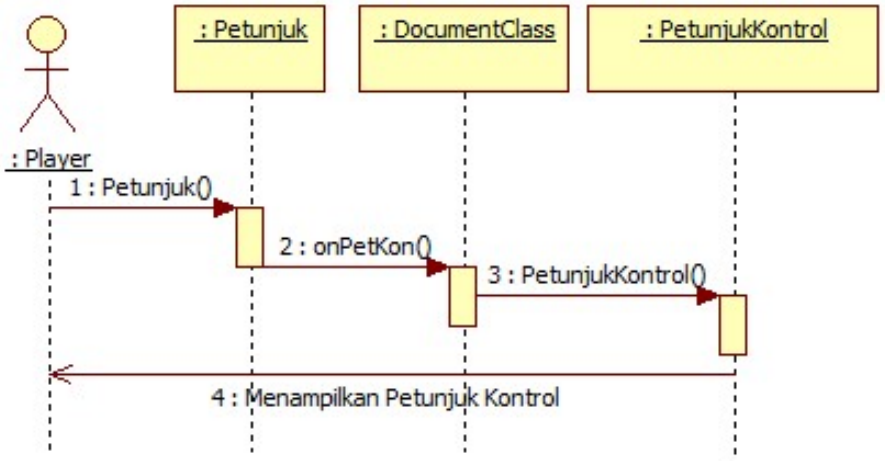
Gambar 3. 28 Diagram *Sequence* Melihat *Credits*

3.10. Diagram *Sequence* Memilih Petunjuk



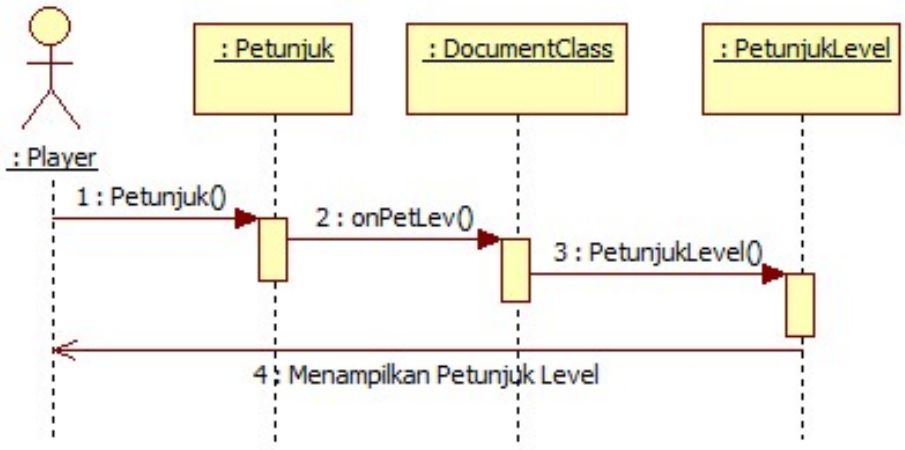
Gambar 3. 29 Diagram *Sequence* Memilih Petunjuk

3.11. Diagram *Sequence* Melihat Petunjuk Kontrol



Gambar 3. 30 Diagram *Sequence* Melihat Petunjuk Kontrol

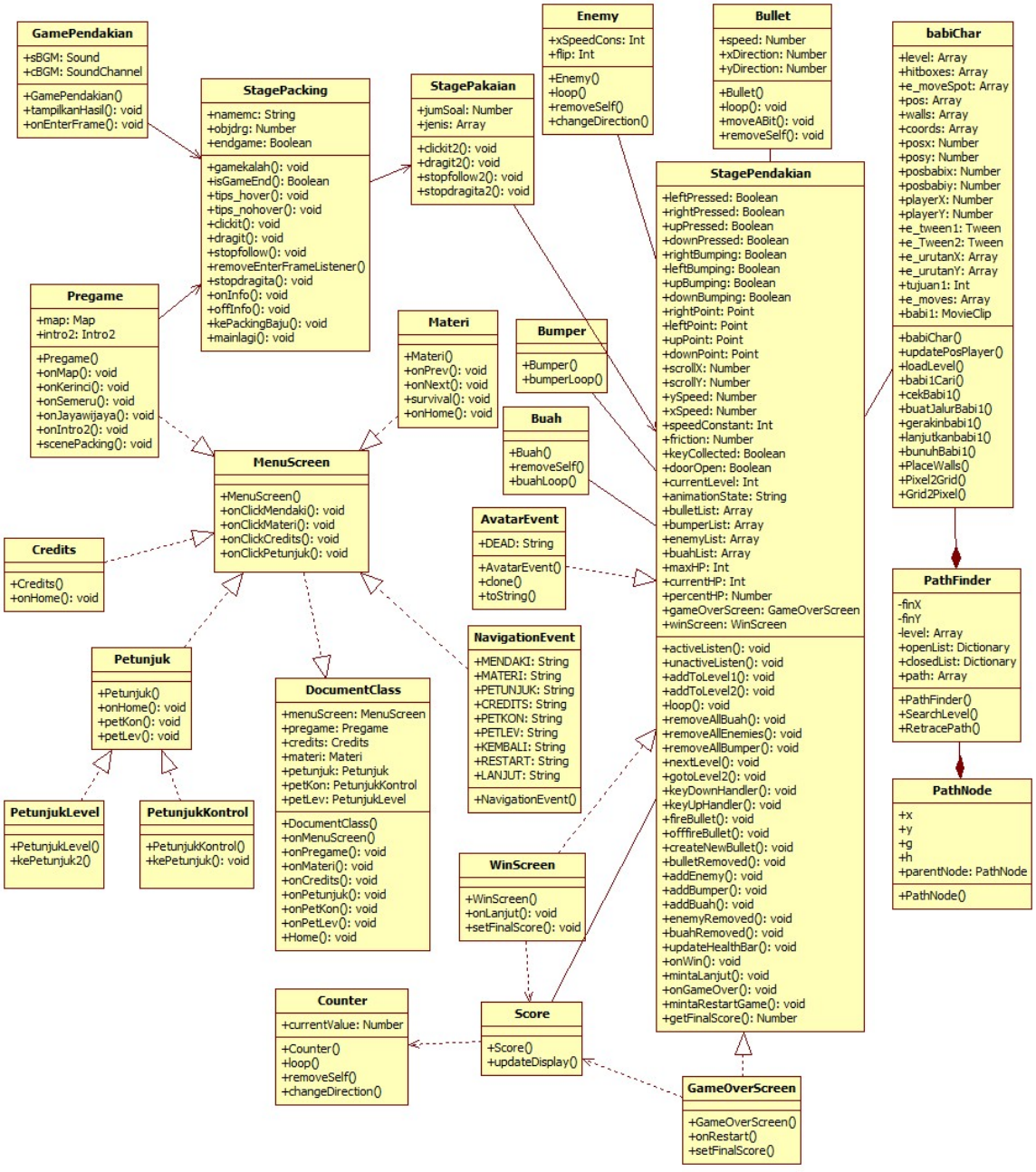
3.12. Diagram *Sequence* Melihat Petunjuk *Level*



Gambar 3. 31 Diagram *Sequence* Melihat Petunjuk *Level*

4. Diagram Kelas

Diagram Kelas menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem. Struktur itu meliputi atribut-atribut dan method-method yang ada pada masing-masing kelas, sedangkan hubungnnya meliputi pewarisan asosiasi, generalilasi. Berikut ini diagram kelas dari *game Jejak Pendaki* :



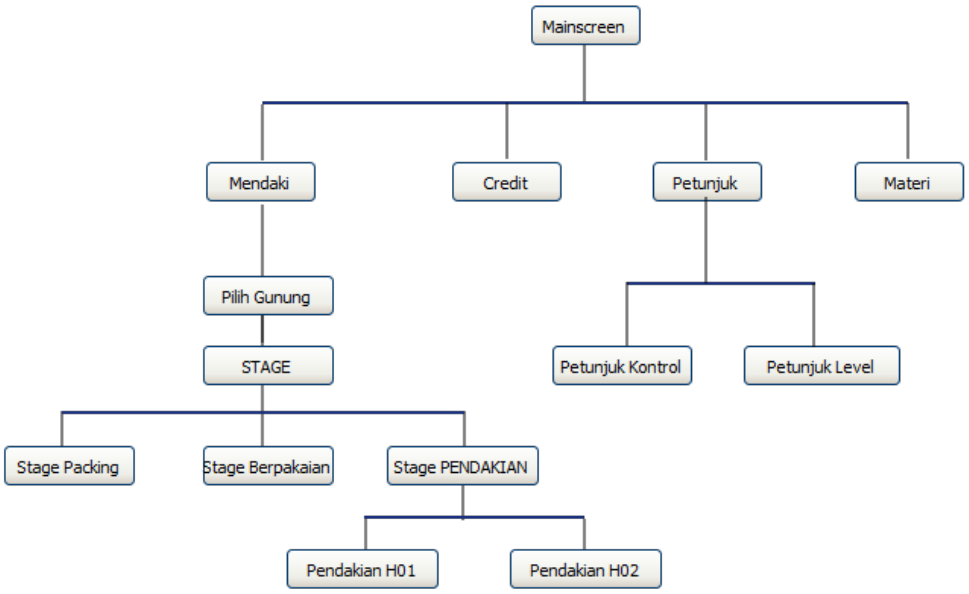
Gambar 3. 32 Diagram Kelas *Game* Jejak Pendaki

3.4 Perancangan Sistem

Pada tahap ini digambarkan rancangan sistem yang akan dibangun sebelum dilakukan pengkodean ke dalam suatu bahasa pemrograman.

3.4.1 Perancangan Menu

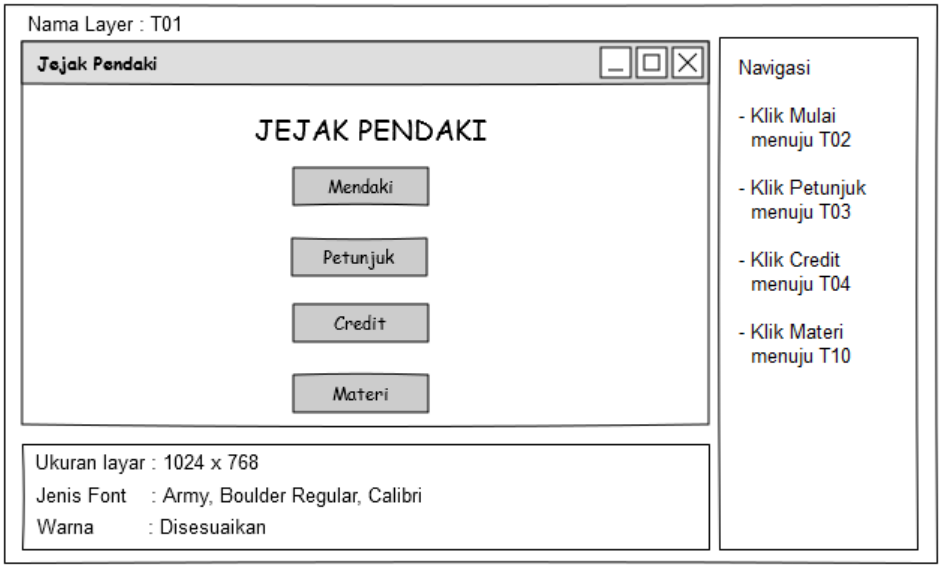
Dalam perancangan sebuah aplikasi dibutuhkan struktur menu yang berisikan menu dan submenu yang berfungsi untuk memudahkan user dalam menggunakan aplikasi tersebut. Berikut ini digambarkan mengenai struktur menu dalam *game* Jejak Pendaki :



Gambar 3. 33 Perancangan Menu

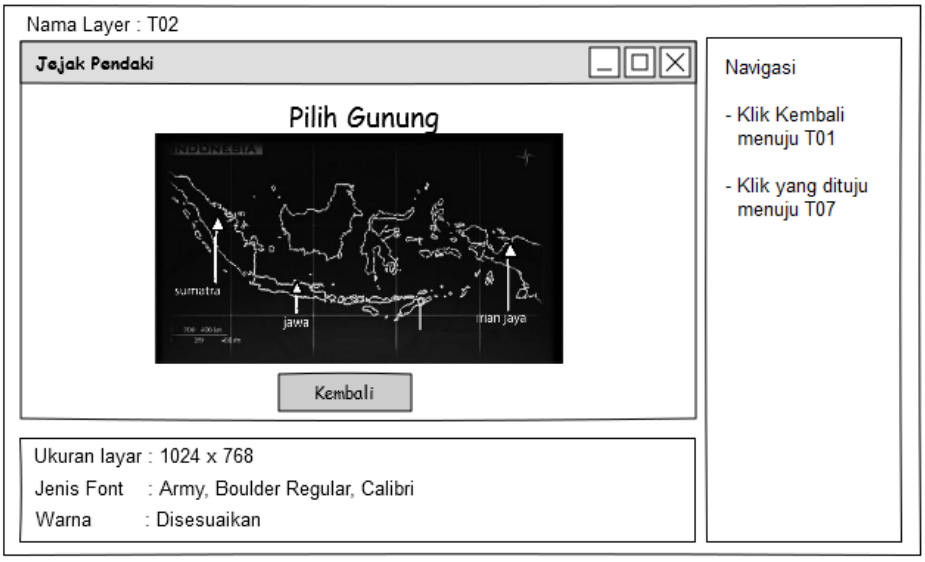
3.4.2 Perancangan Antarmuka

3.4.2.1 Perancangan Antarmuka Mainscreen



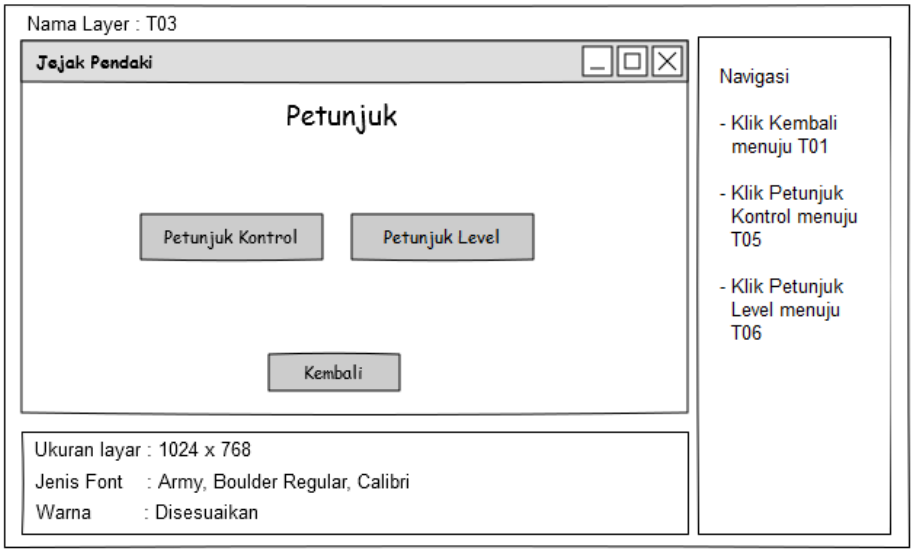
Gambar 3. 34 Perancangan Antarmuka Mainscreen

3.4.2.2 Perancangan Antarmuka Pilih Gunung



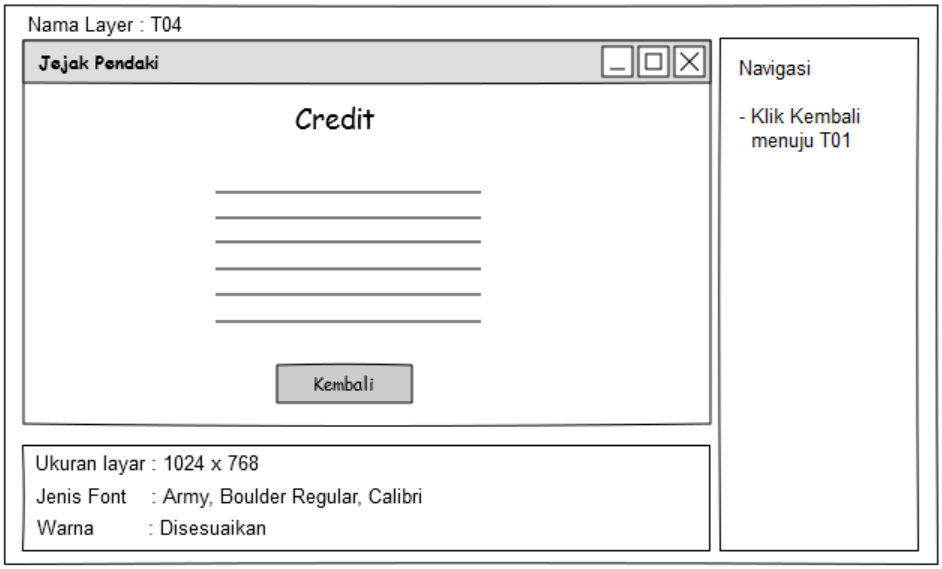
Gambar 3. 35 Perancangan Antarmuka Pilih Gunung

3.4.2.3 Perancangan Antarmuka Petunjuk



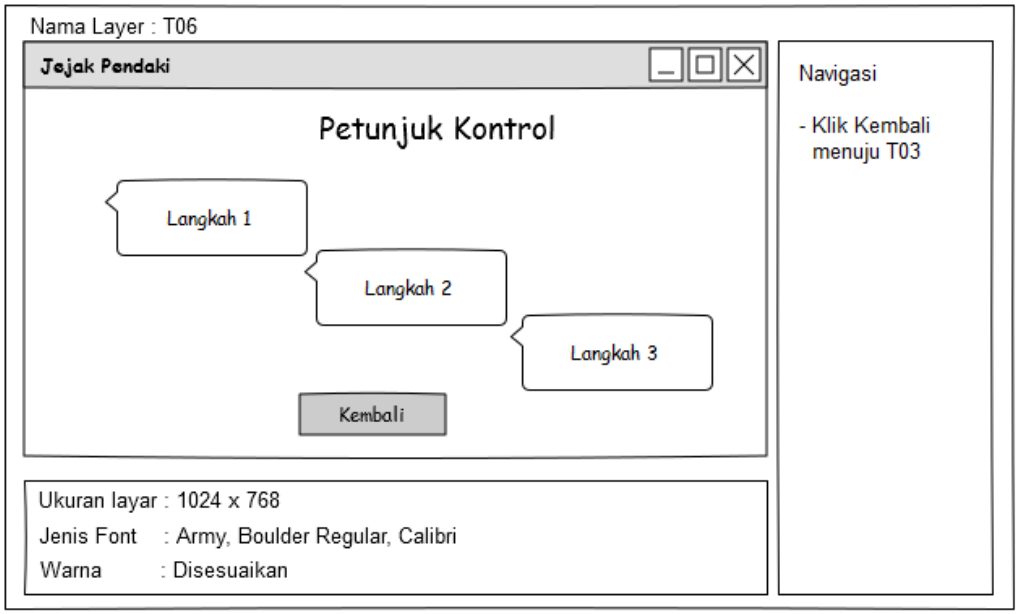
Gambar 3. 36 Perancangan Antamuka Petunjuk

3.4.2.4 Perancangan Antarmuka *Credits*



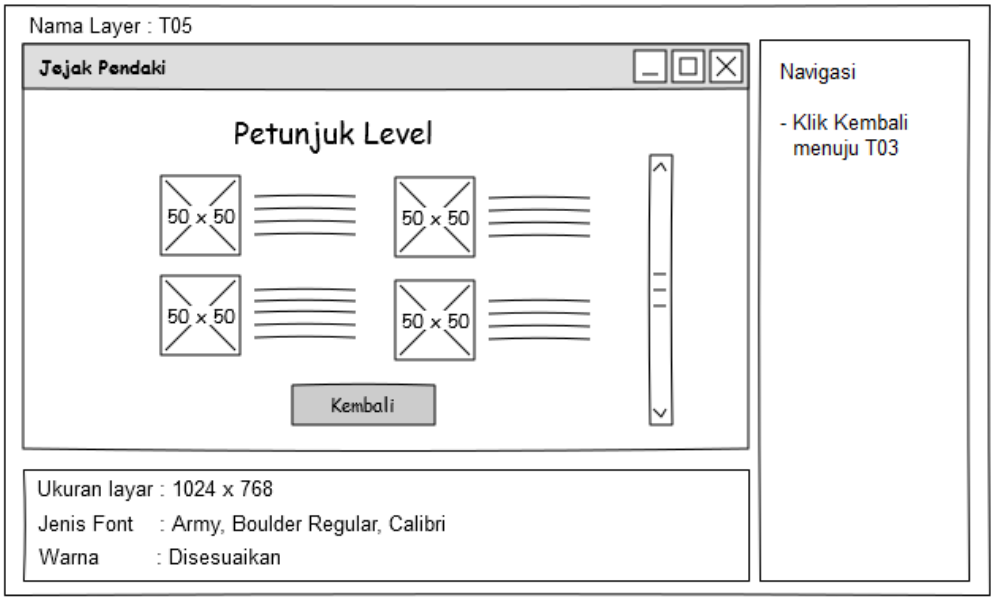
Gambar 3. 37 Perancangan Antarmuka *Credits*

3.4.2.5 Perancangan Antarmuka Petunjuk Kontrol



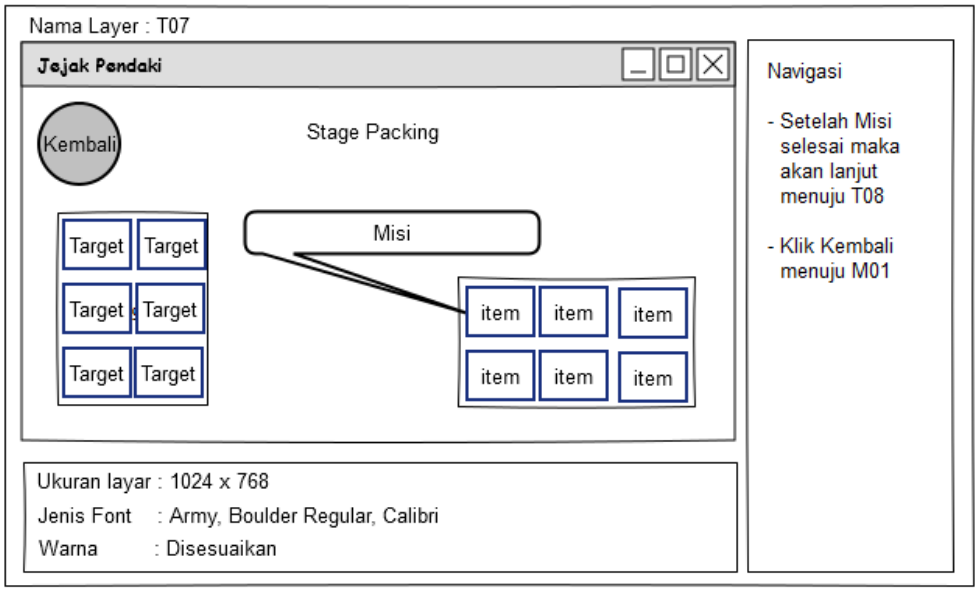
Gambar 3. 38 Perancangan Antarmuka Petunjuk Kontrol

3.4.2.6 Perancangan Antarmuka Petunjuk *Level*

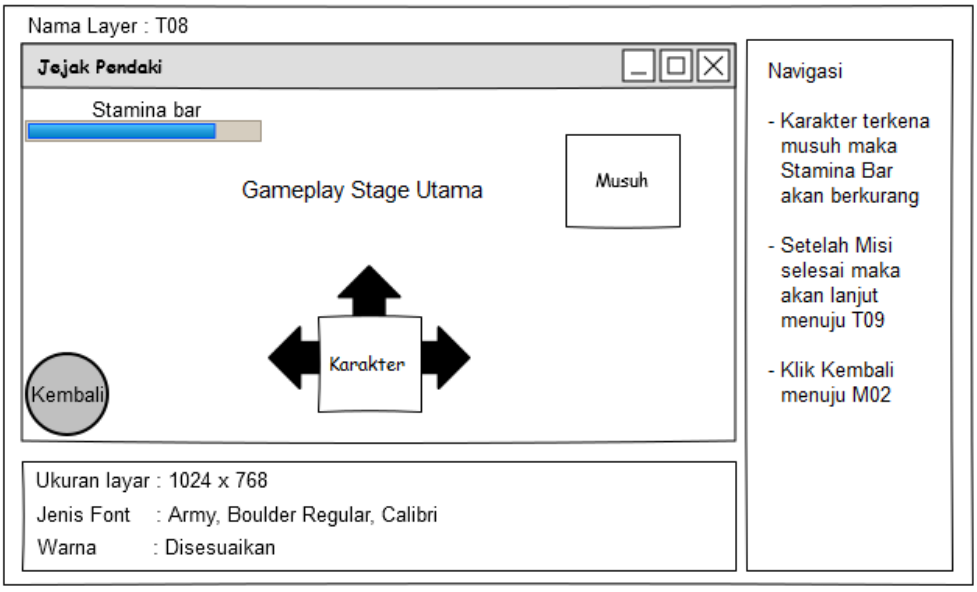


Gambar 3. 39 Perancangan Antarmuka Petunjuk *Level*

3.4.2.7 Perancangan Antarmuka Stage

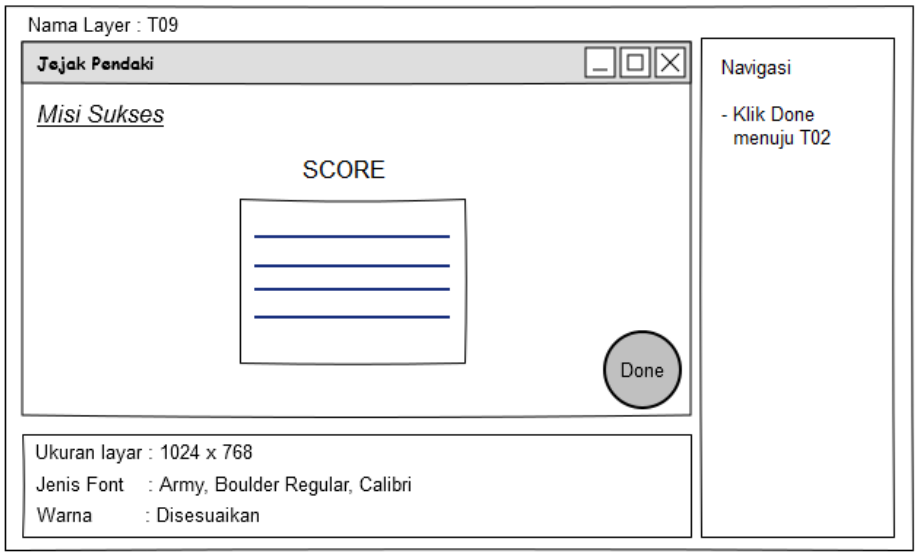


Gambar 3. 40 Perancangan Antarmuka Stage *Packing*



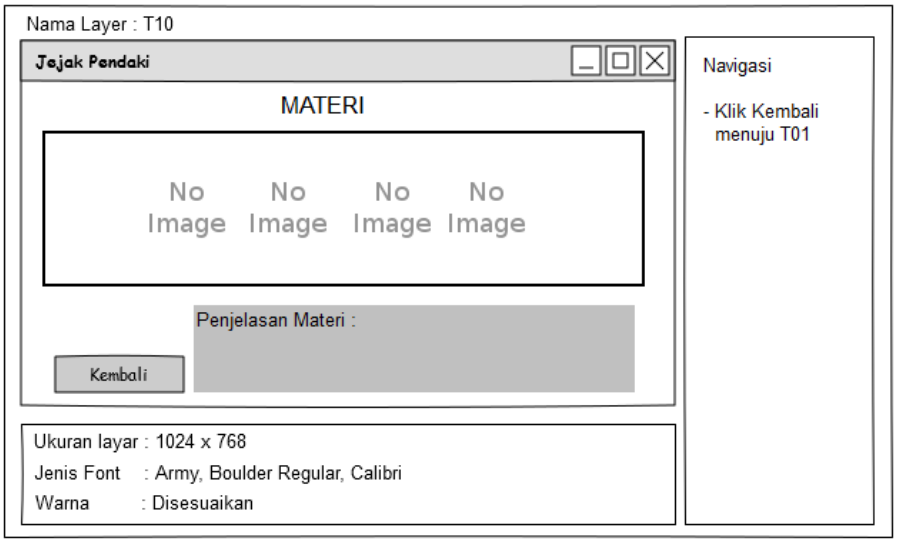
Gambar 3. 41 Perancangan Antarmuka Stage Pendakian

3.4.2.8 Perancangan Antarmuka Scoring



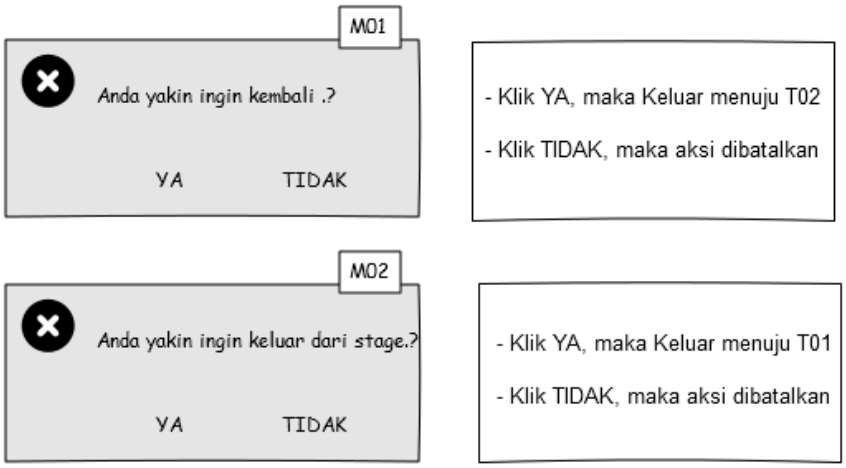
Gambar 3. 42 Perancangan Antarmuka Scoring

3.4.2.9 Perancangan Antarmuka Materi



Gambar 3. 43 Perancangan Antarmuka Materi

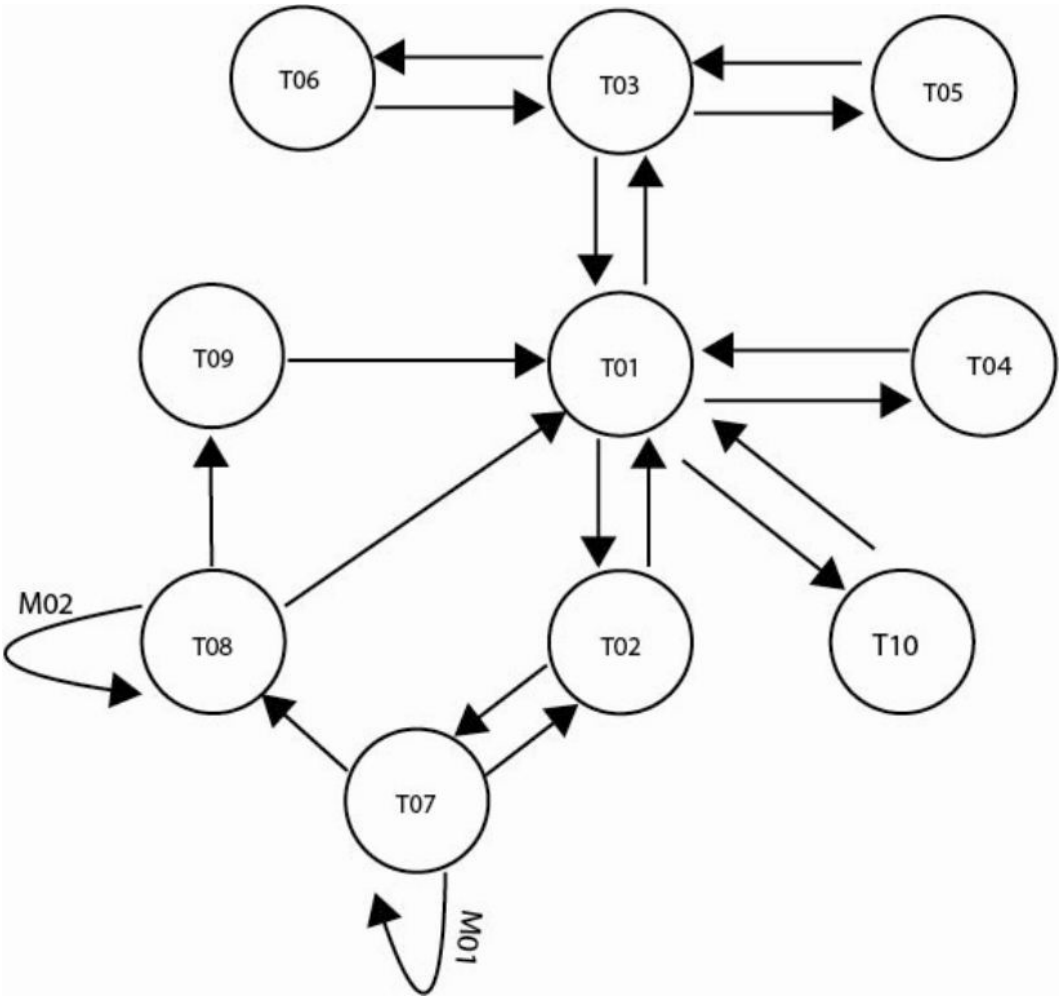
3.4.2.10 Perancangan Antarmuka Pesan



Gambar 3. 44 Perancangan Antarmuka Pesan

3.4.3 Perancangan Jaringan Semantik

Jaringan semantik digunakan untuk menghubungkan setiap tampilan (*interface*) dan pesan pada sebuah aplikasi yang dirancang. Untuk lebih jelasnya lihat gambar berikut :



Gambar 3. 45 Perancangan Jaringan Semantik