BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

Analisis merupakan suatu proses penguraian konsep ke dalam bagian-bagian yang lebih sederhana, sehingga struktur logisnya menjadi jelas Metode untuk menguji, menilai, dan memahami sistem dengan memecahkannya ke dalam unsur-unsur yang lebih sederhana. Pada sub bab ini akan membahas mengenai sistem atau prosedur yang sedang berjalan pada aplikasi *game Puzzle and The Solver.*

3.1.1. Analisis Masalah

Pembelajaran dan pengenalan kebudayaan melalui permainan merupakan proses pembelajaran yang lebih menekankan pada penyajian bentuk permainan dengan muatan pentingnya mengenali dan mempelajari kebudayaan Indonesia.

Game yang mengangkat tema kebudayaan Indonesia dengan *genre Adventure Game* dan *subgenre word puzzle* akan sangat menarik karena selain dapat meningkatkan intelektual anak juga dapat memperkenalkan kepada anak kebudayaan tradisional yang ada di Indonesia. Jenis *game* ini masih sangat jarang dalam perkembangannya, namun masih tetap menarik untuk dimainkan khususnya bagi anak usia 6 sampai 11 tahun, karena jenis permainan ini terbilang mudah untuk dimainkan.

Berdasarkan uraian diatas yang menjadi pokok permasalahan adalah bagaimana membangun aplikasi *game* yang mengangkat tema kebudayaan Indonesia dengan *genre Adventure game* dan *subgenre word puzzle game*?

3.1.2. Analisis *Game* Sejenis

Analisis *game* sejenis dilakukan untuk memberikan gambaran sistem agar lebih mudah dalam pembangunan dan sebagai pembanding untuk kesempurnaan *game* yang akan dicapai. *Game* yang akan dibahas dan menjadi acuan disini ialah *game* Petualangan Aksara di Jawa.

3.1.2.1. *Game* Petualangan Aksara di Jawa

Game Petualangan Aksara di Jawa ini merupakan salah satu game edukasi yang mengusung tema sejarah dan budaya nasional dari Tanah Air yang dipadukan dengan konten yang memiliki unsur pendidikan kepada para pemainnya. Game ini merupakan game yang dibuat oleh Aksara Studio.

a. Pengenalan *Game* Petualangan Aksara di Jawa

Petualangan Aksara di Jawa yang juga dikenal dengan PAJ adalah sebuah permainan yang simpel tentang seorang anak bernama Aksara yang ingin belajar tentang salah satu budaya Indonesia, Aksara Jawa. Namun, ia tidak memiliki kemampuan Aksara Jawa yang baik sehingga ia membutuhkan bantuan pemain yang memainkan *game* Petualangan Aksara di Jawa. Tampilan *game* Petualangan Aksara di Jawa ini dapat dilihat pada gambar 3.1.



Gambar 3. 1 Petualangan Aksara di Jawa

b. *Gameplay* Petualangan Aksara di Jawa

Pada *game* Petualangan Aksara di Jawa ini memiliki beberapa *level* dimana tiap *level* memiliki tingkat kesulitan bermacam-macam, game ini juga memberikan macam-macam huruf jawa sehingga kita dapat mengetahui macam huruf-huruf jawa serta sistem *scoring* yang membuat pemain berusaha mencapai *score* tertinggi.

c. Skenario *Game* Petualangan Aksara di Jawa

Skenario permainan dalam *game* Petualangan Aksara di jawa adalah nantinya akan ada tombol-tombol yang berisi huruf jawa diatasnya ada sejumlah kata sebagai soal, tugas pemain adalah mengeja kata-kata tersebut dengan menekan huruf-huruf jawa pada tombol sehingga menjadi kata-kata jawa, pemain diberi kesempatan buat mengeja kata-kata selama waktu masih ada, setiap kata-kata yang benar maka akan menambah *score* pemain tetapi bila salah akan mengurangi *score* pemain.

Kelebihan yang terdapat pada game Petualangan Aksara di Jawa, yaitu :

a. Pada *game* Petualangan Aksara di Jawa ini terdapat bermacam-macam huruf Jawa.
b. Mempunyai animasi-animasi yang menarik.
c. Tampilan grafis 2D yang bagus dan menggambarkan kebudayaan Jawa.
d. Terdapat sistem *scoring*.

Sementara itu keterbatasan yang terdapat pada game Petualangan Aksara di Jawa yaitu, pada *game* Petualangan Aksara di Jawa ini setiap kata yang terbentuk dari huruf-huruf jawa tidak di artikan dalam pengertian bahasa Indonesia.

3.1.3. Analisis *Game Puzzle and The Solver*

Puzzle and The Solver adalah aplikasi *game dekstop* yang berfokus pada edukasi pengetahuan sosial khususnya dalam bidang sejarah kebudayaan Indonesia pembelajaran dan permainan anak-anak tingkat sekolah dasar. Permainan *Puzzle and The Solver* menyuguhkan suatu permainan *Adventure game*

dengan *subgenre word puzzle game* dan grafis 3D yang menarik, menyenangkan, dan edukatif.

3.1.3.1. *Storyline*

Pada *game* ini terdapat 3 *stage* yang harus dilalui oleh pemain. Disetiap *stage* pemain mendapati dirinya terkunci didalam sebuah labirin besar. Didalam labirin tersebut terdapat musuh yang akan mengganggu pemain menyelesaikan misi. Untuk dapat keluar dari labirin pemain harus mendapatkan tiga buah bintang yang hanya bisa didapatkan dengan menjawab pertanyaan pada peti *puzzle.*

Game ini terbagi atas 3 *stage* permainan berupa *game* tebak kata atau *word puzzle*, yaitu :

a. *Stage* 1 adalah tahapan Sumatera Barat, disini terdapat soal tebak kata yang berhubungan dengan kebudayaan yang terdapat di provinsi Sumatera Barat.
b. *Stage* 2 adalah tahapan Jawa Barat, disini terdapat soal tebak kata yang berhubungan dengan kebudayaan yang terdapat di provinsi Jawa Barat.
c. *Stage* 3 adalah tahapan Papua, disini terdapat soal tebak kata yang berhubungan dengan kebudayaan yang terdapat di provinsi Papua.

3.1.3.2. Gameplay

Game ini dimainkan secara perseorangan dengan bentuk permainan berupa *puzzle* labirin dan tebak kata. Pada awal permainan, pemain akan mendapati dirinya berada didalam sebuah labirin besar, dan pintu untuk keluar dari labirin dalam keadaan terkunci. Untuk membuka pintu tersebut maka pemain diharuskan untuk mencari peti kuis tebak kata dan menjawab tiap kata dari quis tebak kata yang terdapat didalam labirin tempat dia berada. Jika pemain berhasil menebak setiap kata tersebut maka pemain akan diberikan *bintang* untuk membuka pintu keluar labirin. Jika pintu sudah terbuka maka pemain bisa melanjutkan ke permainan ke tahap selanjutnya. Pada *game* ini pemain akan memiliki kemampuan berupa tembakan sihir untuk mengalahkan musuh-musuh yang terdapat didalam labirin.

3.1.3.3. Analisis Algoritma *Greedy*

Algoritma yang digunakan untuk membangun kuis tebak kata pada aplikasi *game Puzzle and The Solver* ini adalah Algoritma *greedy*. Berikut ini adalah penjelasan dari pengimplementasian algoritma *greedy* pada aplikasi *game Puzzle and The Solver*.

Elemen-elemen yang digunakan dalam penerapan algoritma *greedy* antara lain:

1. Himpunan Kandidat = C

   C ={A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,X,Y,Z}

2. Himpunan Solusi = S

   Himpunan Solusi (S) adalah jawaban dari pertanyaan yang ditampilkan.

3. Fungsi Seleksi

   Mengambil sebuah kandidat dari Himpunan Kandidat (C).

4. Fungsi Kelayakan

   Memeriksa apakah ada berapa huruf berdasarkan kandidat yang dipilih ada dalam barisan huruf dari jawaban. Dilakukan dengan cara mencocokkan pertanyaan yang ditampilkan dengan pertanyaan dari daftar soal, jika sudah ketemu bandingkan kandidat dengan jawaban tersebut. Jika ada tampilkan semua kandidat yang ada dalam jawaban, jika tidak buang kandidat tersebut dan jangan dipilih lagi.

5. Fungsi Objektif

   Batas maksimal kesalahan dalam memilih kandidat adalah tiga(3) kali.

Algoritma *greedy* dapat dituliskan sebagai berikut :

1. Inisialisasi S dengan kosong.

2. Pilih sebuah kandidat dari C dengan fungsi seleksi.

3. Kurangi C dengan kandidat yang telah terpilih di atas.
4. Periksa apakah kandidat yang dipilih tersebut bersama, membentuk solusi atau tidak. Jika ya masukkan kandidat ke S, jika tidak buang kandidat tersebut dan tidak usah dihiraukan lagi serta tambahkan satu untuk variabel kesalahan.
5. Periksa apakah S yang sudah terbentuk telah memberikan solusi yang lengkap. Jika ya, berhenti. Jika tidak, cek kesalahan memilih kandidat apakah sudah tiga(3) kali sebelum mengulangi dari langkah dua (2).

Contoh pertanyaan : Alat musik getar tradisional Jawa Barat yang terbuat dari bambu atau batang aren adalah?

Berikut adalah langkah untuk menyelesaikan pertanyaan di atas dengan menggunakan algoritma *greedy* :

1. Menentukan himpunan kandidat (C).
   C ={A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,X,Y,Z}
2. Menentukan himpunan solusi (S), untuk mendapatkan solusi kita harus membandingkan pertanyaan dengan pertanyaan yang ada didaftar soal, setelah ditemukan hitung panjang huruf dari jawaban untuk pertanyaan tersebut. Sediakan kotak sebanyak panjang huruf dari jawaban.

   S =

*	*	*	*	*	*	*	*	*

3. Berikan nilai nol (0) untuk vaiabel kesalahan (fungsi objektif).
   kesalahan = 0
4. Lakukan langkah no dua (2), ambil sebuah kandidat dari C.
   kandidat_yang_diambil = A
5. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.
   C ={ B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,X,Y,Z}

6. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

S =

*	A	*	*	*	*	*	*	*

kesalahan = 0

7. Ulangi langkah no dua (2), ambil sebuah kandidat dari C.
   kandidat_yang_diambil = R
8. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.
   C ={ B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q, S,T,U,V,X,Y,Z}
9. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

S =

*	A	R	*	*	*	*	*	*

kesalahan = 0

10. Ulangi langkah no dua (2), ambil sebuah kandidat dari C.
    kandidat_yang_diambil = B
11. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.
    C ={ C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,S,T,U,V,X,Y,Z}
12. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

S=

*	A	R	*	*	*	*	*	*

kesalahan = 1

13. Ulangi langkah no dua (2), ambil sebuah kandidat dari C.
    kandidat_yang_diambil = N

14. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.

    C ={ C,D,E,F,G,H,I,J,K,L,M, O,P,Q, S,T,U,V,X,Y,Z}

15. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

S =

*	A	R	*	N	*	*	N	*

kesalahan = 1

16. Ulangi langkah no dua (2), ambil sebuah kandidat dari C.
    kandidat_yang_diambil = I

17. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.

    C ={ C,D,E,F,G,H, J,K,L,M, O,P,Q, S,T,U,V,X,Y,Z}

18. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

S =

*	A	R	I	N	*	I	N	*

kesalahan = 1

19. Ulangi langkah no dua (2), ambil sebuah kandidat dari C.
    kandidat_yang_diambil = G
20. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.
    C ={ C,D,E,F,H,I,J,K,L,M, O,P,Q, S,T,U,V,X,Y,Z}
21. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

    S =

*	A	R	I	N	*	I	N	G

    kesalahan = 1
22. Ulangi langkah no dua (2), ambil sebuah kandidat dari C.
    kandidat_yang_diambil = D
23. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.
    C ={ C,E,F,H,I,J,K,L,M, O,P,Q, S,T,U,V,X,Y,Z}
24. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

    S=

*	A	R	I	N	D	I	N	G

    kesalahan = 1
25. Ulangi langkah no dua (2), ambil sebuah kandidat dari C.
    kandidat_yang_diambil = V
26. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.

C ={ C, E,F,G,H,I,J,K,L,M, O,P,Q, S,T,U, X,Y,Z}

27. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

S =

*	A	R	I	N	D	I	N	G

kesalahan = 2

28. Ulangi langkah no dua (2), ambil sebuah kandidat dari C.
kandidat_yang_diambil = K

29. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.

C ={ C,D,E,F,G,H,I,J, L,M, O,P,Q, S,T,U, X,Y,Z}

30. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

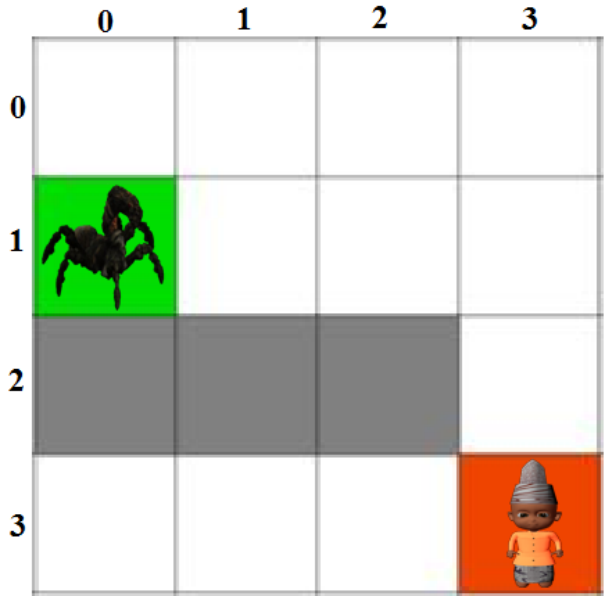
S=

K	A	R	I	N	D	I	N	G

kesalahan = 2

3.1.3.4. Analisis Metode *Path Finding*

Path finding digunakan untuk menentukan arah pergerakan suatu objek dari satu tempat ke tempat lain berdasarkan keadaan peta dan objek lainnya. Pada aplikasi *game puzzle and the solver* ini metode *path finding* diterapkan pada karakter musuh untuk mencari jalur terpendek untuk mendekati dan menyerang karakter utama (pemain). Dalam pemecahan masalah untuk mencari jalur terpendek pada metode *path finding*, algoritma yang akan digunakan adalah algoritma A *star*. Analisis algoritma A *star* pada metode *path finding* akan dimodelkan dalam sebuah kotak yang berbentuk *puzzle* yang ber-ordo 4 x 4, dapat dilihat pada gambar 3.2.



Gambar 3. 2 Contoh kondisi pemodelan ordo 4x4

Seperti terlihat pada gambar 3.2, kotak berwarna abu-abu adalah penghalang atau disebut *unwalkable* yang tidak bisa dilewati, kotak berwarna hijau merupakan titik *node* awal dimana algoritma A *star* diterapkan, sedangkan kotak berwarna merah adalah *node* tujuan yang merupakan koordinat karakter utama(pemain).

Langkah-langkah pencarian jarak terdekat dengan menggunakan algoritma A *star* adalah sebagai berikut :

1. Masukan *node* awal kedalam *open list.*
2. Periksa semua *node* yang berdekatan dengan *node* A apakah *walkable* atau tidak, abaikan *node* dengan penghalang lalu tambahkan semua *node* tersebut ke dalam *open list*. Untuk setiap *node* yang baru ditambahkan, simpan *node* A sebagai "*parent*".
3. Hapus *node* A dari *open list,* dan tambahkan ke dalam *closed list.*
4. Setiap *node* yang berada pada *open list* akan dihitung nilai g, dan h-nya maka dapat diperoleh nilai f-nya. Nilai g diperoleh dengan cara menghitung nilai (*cost*) jarak dari *open list* terhadap *closed list*. Jika pergerakan nilai g secara horizontal atau vertikal, maka nilai g nya didefinisikan sebesar 10, dan jika pergerakan nilai g nya secara diagonal, maka nilainya didefinisikan sebesar 14. Nilai f didapat dari pertambahan nilai g dan h.
5. Menambahkan *node* terdekat dari *node* awal kedalam *open list*, dari gambar 3.2 *node* terdekat dari *node* awal adalah (0,0),(1,0), dan (1,1). Contoh penghitungan *node* terdekat menggunakan rumus *Manhattan distance*:

   h(n) = D * (abs(n.x-goal.x) + abs(n.y-goal.y))

   D = pergerakan nilai g (horizontal = vertikal = 10)

   abs = *Absence* (ketiadaan)

   Hasil penghitungan

   h(0,0) = 10 * (abs(0-3) + abs(0-3)

   h(0,0) = 10 * (abs(-3)+abs(-3))

   h(0,0) = 10 * ((3)+(3))

   h(0,0) = 10 * 6

   h(0,0) = 60

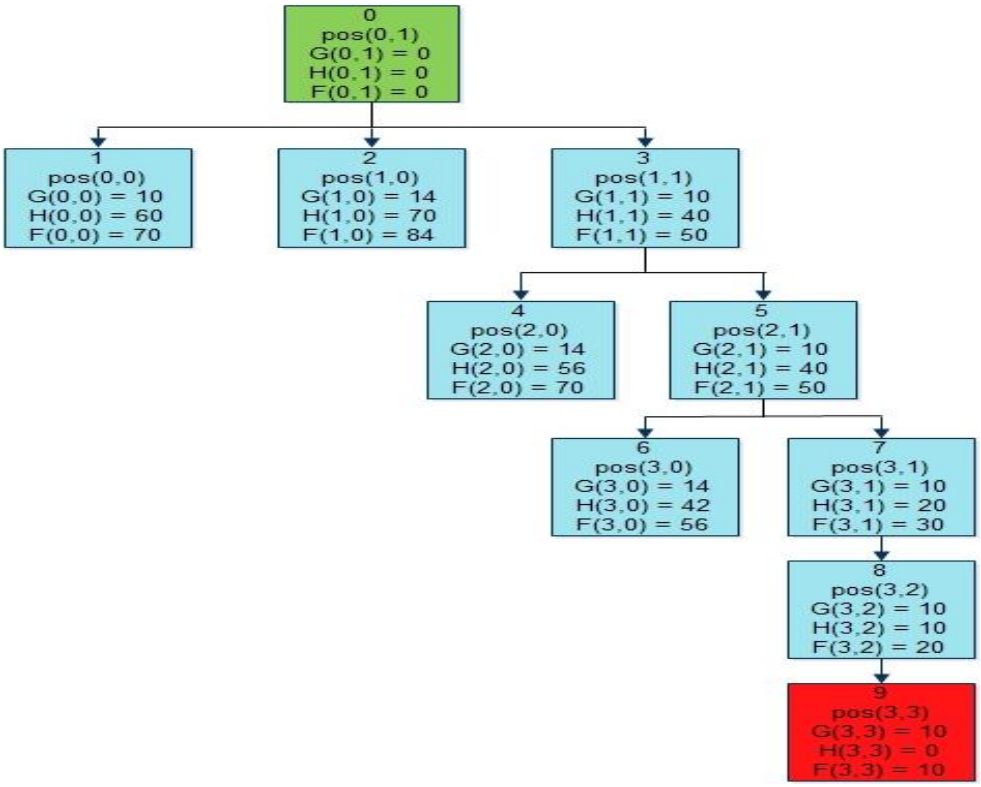
   Penghitungan f(n)

   f(n) = g(n) + h(n)

   f(0,0) = 10 + 60

   f(0,0) = 70

Gambar 3.3 adalah pohon hasil penghitungan algoritma A *star* contoh kondisi pemodelan ordo 4x4 pada gambar 3.3.



Gambar 3. 3 Pohon Solusi Algoritma A *star* ordo 4x4

3.1.4. Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan *non*-fungsional menggambarkan kebutuhan system yang menitik beratkan pada properti prilaku yang dimiliki oleh sistem, diantaranya kebutuhan perangkat keras, perangkat lunak, serta user sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.1.4.1. Analisis Kebutuhan Perangkat Lunak

1. Spesifikasi perangkat lunak bagi pengembang yang digunakan dalam membangun aplikasi *game* edukasi petualangan pulau pintar adalah :
    a. Sistem Operasi Windows 7.
    b. Unity 3D 3.5.6.
    c. Google SketchUp 8.

d. Adobe Photoshop CS5.
e. Autodesk Maya 2011.

2. Perangkat lunak yang digunakan oleh pemain untuk menjalankan *game* edukasi petualangan pulau pintar adalah sistem operasi windows (Windows XP, Vista, Seven).

3.1.4.2. Analisis Kebutuhan Perangkat Keras

1. Spesifikasi *minimum* perangkat keras yang dibutuhkan oleh pengembang.
    a. Prosesor dengan kecepatan 1.8 Ghz.
    b. RAM 2 Gb.
    c. Hardisk 20 Gb.
    d. VGA *Card* 256 Mb.
    e. Monitor.
    f. *Mouse* dan *Keyboard*.
    g. *Speaker*.
2. Spesifikasi *minimum* perangkat keras yang dibutuhkan oleh pemain.
    a. Prosesor dengan kecepatan 1.8 Ghz ke atas.
    b. RAM 1 Gb.
    c. Hardisk 500 Mb.
    d. VGA *Card* 256 Mb.
    e. Monitor.
    f. *Mouse* dan *Keyboard*.
    g. *Speaker*.

3.1.4.3. Analisis User

Analisis *user*(pemain) ini akan mencakup analisis beberapa parameter terhadap calon *user* dari applikasi.

a. Pengetahuan dan Kemampuan pengguna(*user*)

Tingkat Pendidikan	**Kemampuan Membaca**	**Kemampuan mengetik**
Digunakan oleh anak tingkat sekolah dasar, atau usia 6 sampai 11 tahun	Bisa digunakan oleh pengguna dengan *reading level* yang sedang	Tidak memerlukan kemampuan mengetik yang tinggi.
Pengetahuan Komputer	**Pengalaman Bermain game**	**Bahasa yang digunakan**
Tingkat menengah	Bisa digunakan oleh pengguna dengan pengalaman penggunaan komputer dan *game* yang sedang	Bahasa Indonesia

b. Karakteristik keadaan fisik pengguna(*user*)

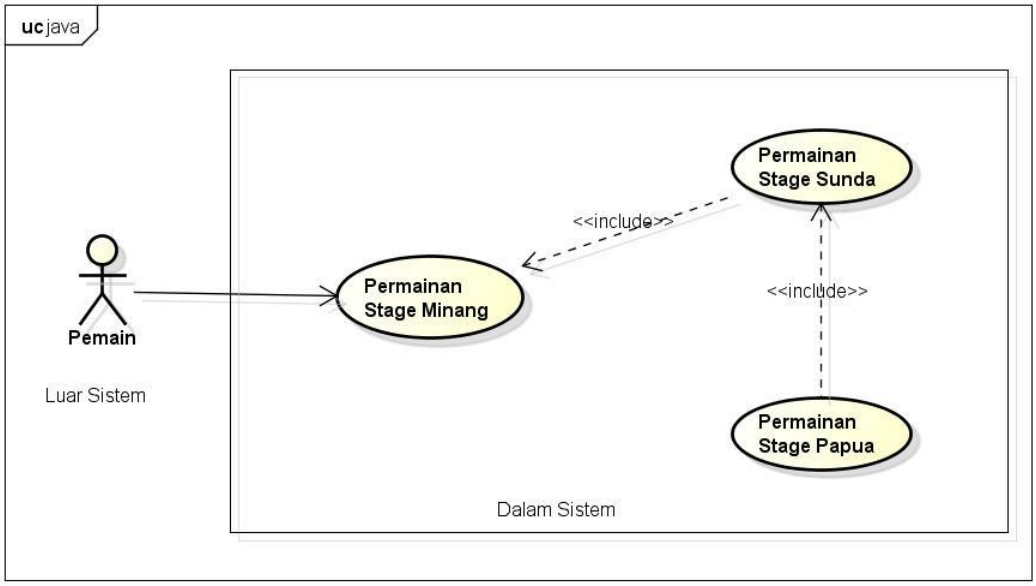
Usia	6 sampai 11 tahun
Jenis kelamin	Pria dan Wanita
Buta warna	User yang tidak bisa membedakan warna yang satu dengan yang lainnya (buta warna) masih mampu menggunakan aplikasi ini, karena tidak ada indicator warna-warna khusus yang membedakan antara fungsional yang satu dengan fungsional yang lainnya. Akan tetapi penggunaannya tidak akan optimal karena dalam *game* ini terdapat banyak sekali perbedaan warna yang menunjang interaksi dan ketertarikan dalam permainan.

3.1.5. 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.

3.1.5.1. Usecase Diagram

Usecase Diagram merupakan konstruksi untuk mendeskripsikan hubungan-hubungan yang terjadi antar aktor dengan aktivitas yang terdapat pada sistem. Sasaran pemodelan *usecase* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan sistem yang akan dibangun. Dari hasil analisis aplikasi yang ada maka *usecase diagram* untuk aplikasi *game puzzle and the solver* dapat dilihat pada gambar 3.4.



Gambar 3. 4 *Usecase Diagram* aplikasi *Puzzle and The Solver*

1. **Definisi *Actor***

Definisi *Actor* berfungsi untuk menjelaskan *Actor* yang terdapat pada *Use Case Diagram*.

Tabel 3. 1 Definisi *Actor*

No	*Actor*	Deskripsi
1	Pemain	Aktor utama yang mengakses aplikasi.

2. Definisi *Usecase*

Tabel 3. 2 Definisi *Usecase*

No	Usecase	Deskripsi
1	Permainan *Stage* Minang	Merupakan *stage* pertama area permainan.
2	Permainan *Stage* Sunda	Merupakan *stage* kedua area permainan.
3	Permainan *Stage* Papua	Merupakan *stage* terakhir area permainan.

3.1.5.2. Skenario *Usecase*

Skenario *Usecase* mendeskripsikan urutan langkah-langkah dalam proses bisnis, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor.

1. Skenario *Usecase* Permainan *stage* Minang

Tabel 3. 3 Skenario *Usecase* Permainan *stage* Minang

Identifikasi	
Nama	Permainan *stage* minang
Tujuan	Memainkan *stage* minang dalam permainan
Deskripsi	*Stage* pertama area permainan
Aktor	*Player* (pemain)
Kondisi awal	Menampilkan permainan *stage* minang
Kondisi Akhir	Pemain dinyatakan berhasil atau gagal
Skenario Normal	
Aksi Aktor	**Reaksi Sistem**
	1.Menampilkan permainan *stage* minang
2.Memainkan permainan *stage* minang 3. Menggerakan karakter minang	
	4.Mengecek trigger peti puzzle 5.Jika pemain berada dalam lingkungan trigger, maka sistem akan menampilkan *form* kuis tebak kata.
6.Menjawab pertanyaan kuis tebak kata	
	7.Jika jawaban benar, maka objek bintang yang berada didalam objek peti dapat diambil.

2. Skenario *Usecase* Permainan *stage* Sunda

Tabel 3. 4 Skenario *Usecase* permainan *stage* Sunda

Identifikasi	
Nama	Permainan *Stage* Sunda
Tujuan	Memainkan *stage* Sunda dalam permainan
Deskripsi	*Stage* kedua area permainan
Aktor	*Player* (pemain)
Kondisi awal	Menampilkan permainan *stage* Sunda
Kondisi Akhir	Pemain dinyatakan berhasil atau gagal
Skenario Normal	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan permainan *stage* Sunda
2.Memainkan permainan *stage* Sunda. 3. Menggerakan karakter sunda	
	4.Mengecek trigger peti puzzle 5.Jika pemain berada dalam lingkungan trigger, maka sistem akan menampilkan *form* kuis tebak kata.
6.Menjawab pertanyaan kuis tebak kata	
	7.Jika jawaban benar, maka objek bintang yang berada didalam objek peti dapat diambil.

3. Skenario *Usecase* Permainan *stage* Papua

Tabel 3. 5 Skenario *Usecase* Permainan *stage* Papua

Identifikasi	
Nama	Permainan *Stage* Papua
Tujuan	Memainkan *stage* Papua dalam permainan
Deskripsi	*Stage* terakhir area permainan
Aktor	*Player* (pemain)
Kondisi awal	Menampilkan permainan *stage* Papua
Kondisi Akhir	Pemain dinyatakan berhasil atau gagal
Skenario Normal	
Aksi Aktor	**Reaksi Sistem**
	1.Menampilkan permainan *stage* papua
2.Memainkan permainan *stage*	

papua. 3.Menggerakan karakter papua	
	4.Mengecek trigger peti puzzle 5.Jika pemain berada dalam lingkungan trigger, maka sistem akan menampilkan *form* kuis tebak kata.
6.Menjawab pertanyaan kuis tebak kata	
	7.Jika jawaban benar, maka objek bintang yang berada didalam objek peti dapat diambil.

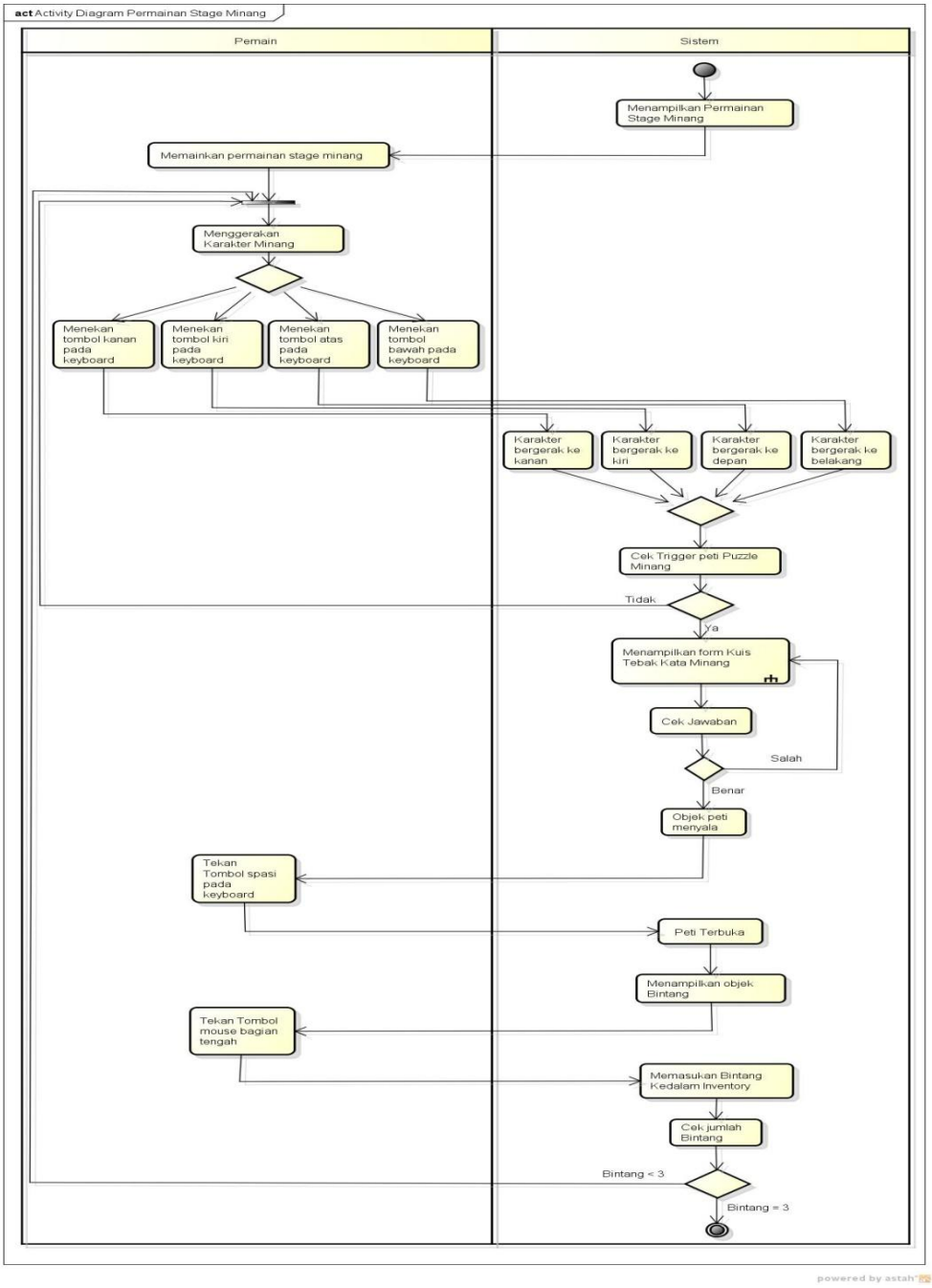
3.1.5.3. Activity Diagram

Activity diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis dan jalur kerja. dalam beberapa hal, *diagram* ini memainkan peran mirip sebuah *diagram* alir, tetapi perbedaan prinsip antara *diagram* ini dan notasi *diagram* alir adalah *diagram* ini mendukung *behavior pararel*.

Activity diagram menggambarkan (*workflow*) berbagai alir aktifitas dalam sistem yang sedang di rancang, bagaimana masing-masing alir berawal, *desition* yang mungkin terjadi dan bagaimana alir berakhir.

1. *Activity Diagram* Permainan *stage* Minang

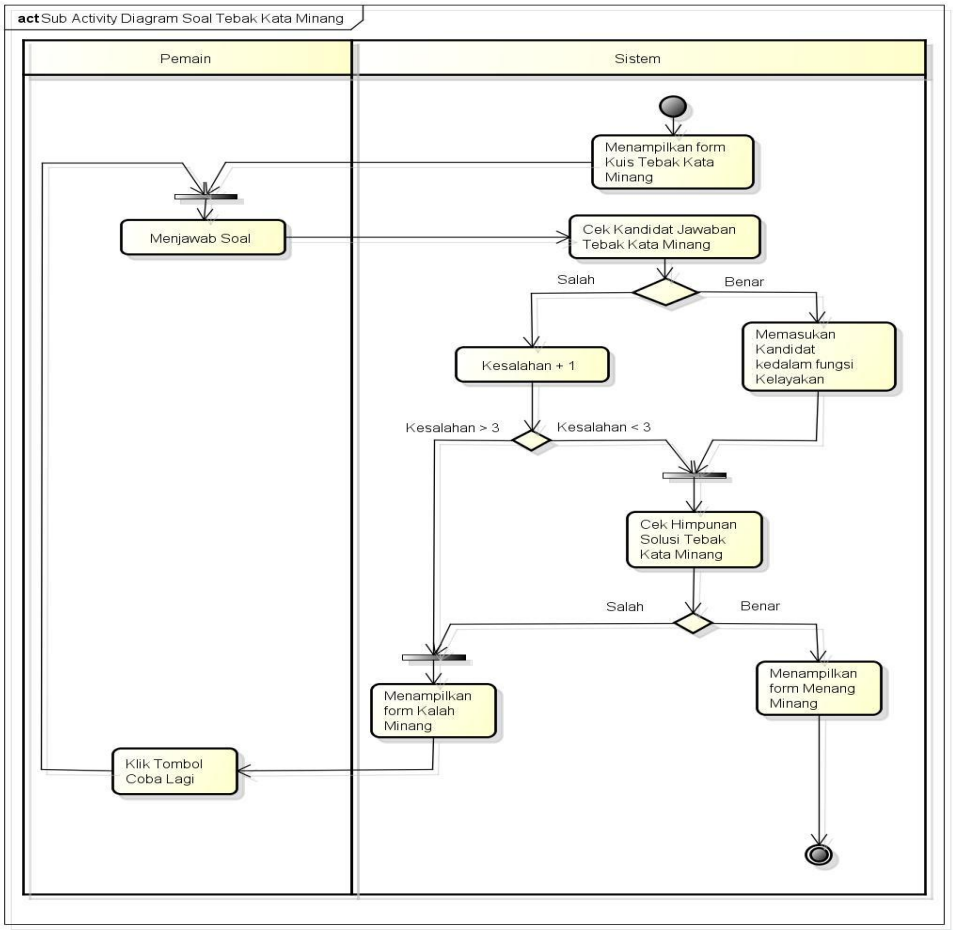
Activity diagram stage Minang adalah aktifitas permainan pada *stage* pertama, dapat dilihat pada gambar 3.5.



Gambar 3. 5 *Activity Diagram* Permainan *stage* Minang

2. *Sub Activity Diagram* Permainan *stage* Minang

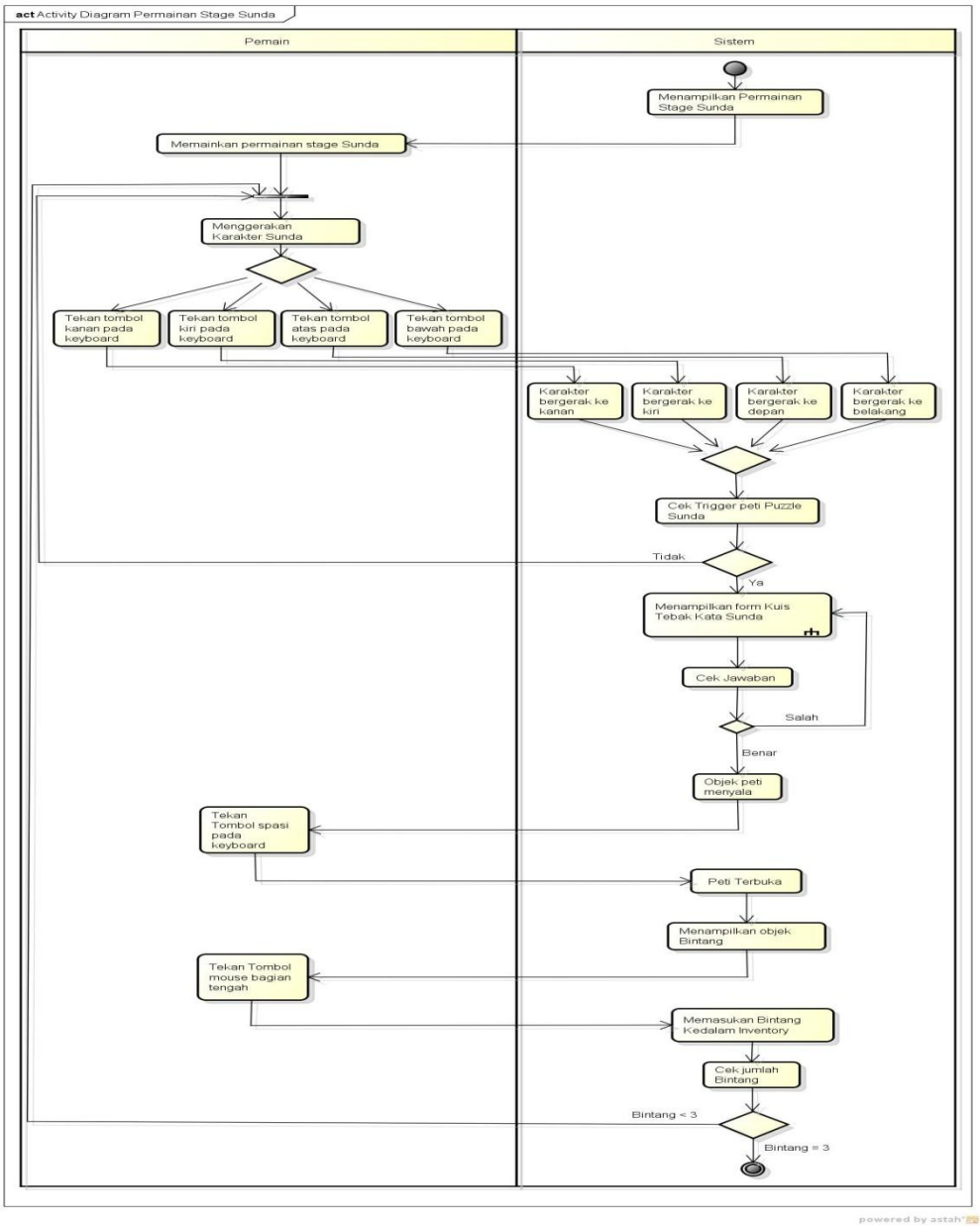
Mengambarkan rangkaian aliran dari *sub activity diagram* tebak kata *stage* Minang. Aktivitas ini merupakan bagian dari *activity diagram stage* Minang, digunakan untuk aktivitas menjawab tebak kata *stage* Minang. Dapat dilihat pada gambar 3.6.



Gambar 3. 6 *Sub Activity Diagram* Permainan *stage* Minang

3. *Activity Diagram* Permainan *stage* Sunda

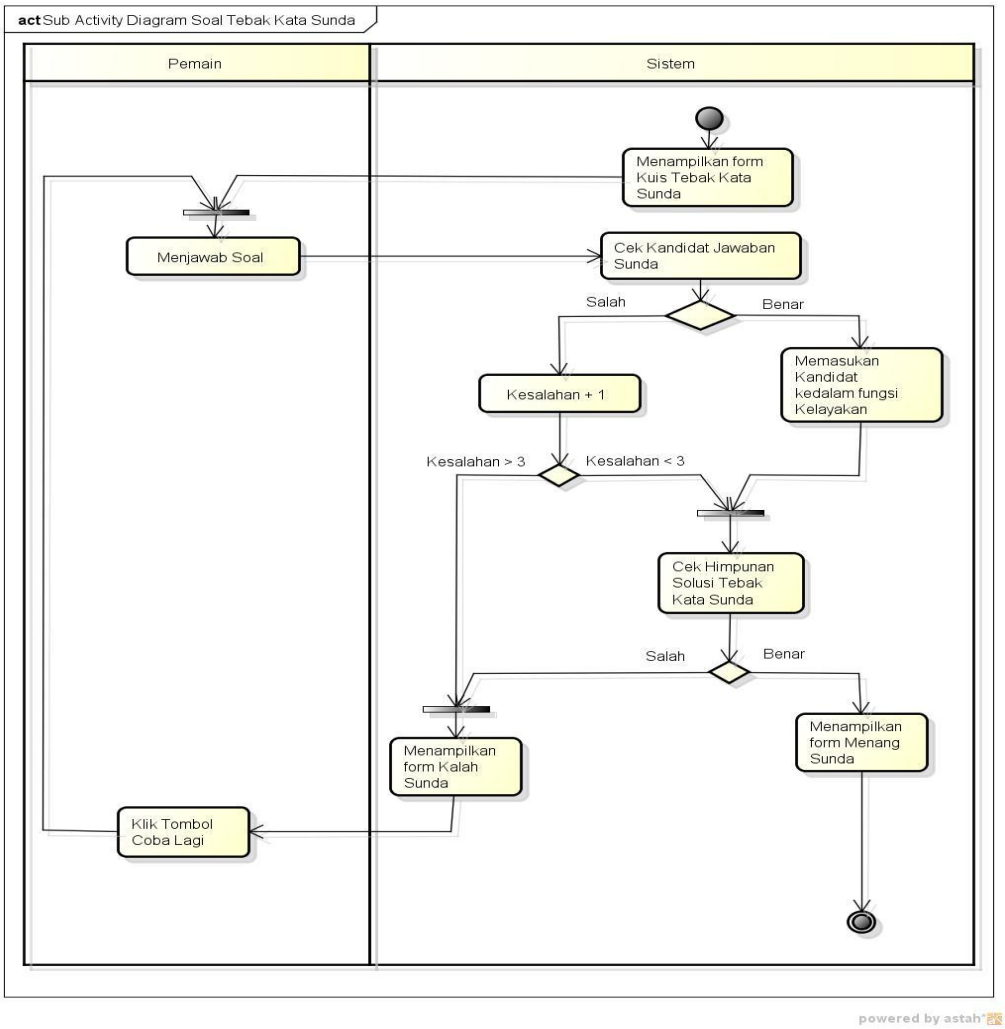
Activity diagram stage sunda adalah aktifitas permainan pada *stage* kedua, dapat dilihat pada gambar 3.7.



Gambar 3. 7 *Activity Diagram* Permainan *stage* Sunda

4. *Sub Activity Diagram* Permainan *stage* Sunda

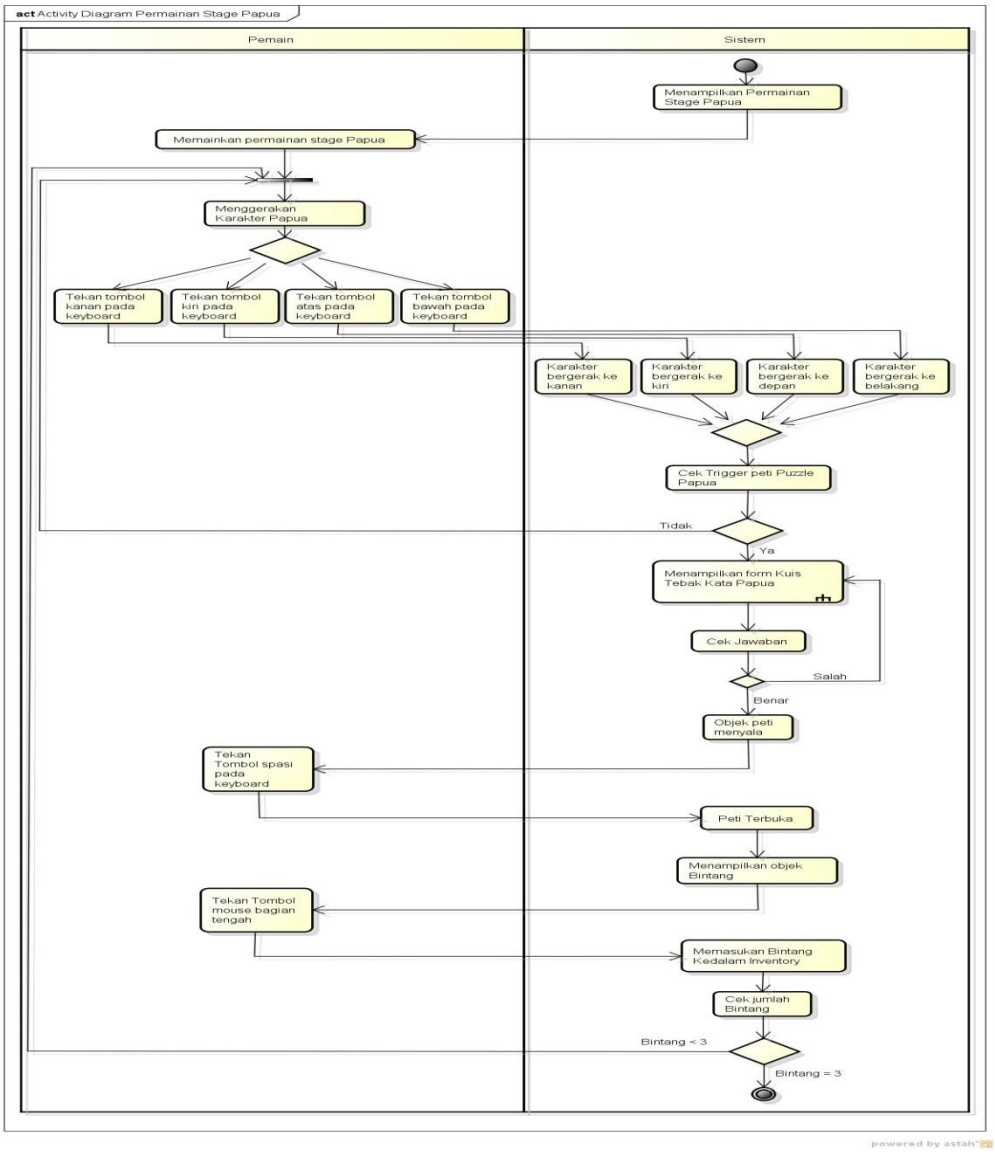
Mengambarkan rangkaian aliran dari *sub activity diagram* tebak kata *stage* Sunda. Aktivitas ini merupakan bagian dari *activity diagram stage* Sunda, digunakan untuk aktivitas menjawab tebak kata *stage* Sunda. Dapat dilihat pada gambar 3.8.



Gambar 3. 8 *Sub Activity Diagram* Permainan *stage* Sunda

5. *Activity Diagram* Permainan *stage* Papua

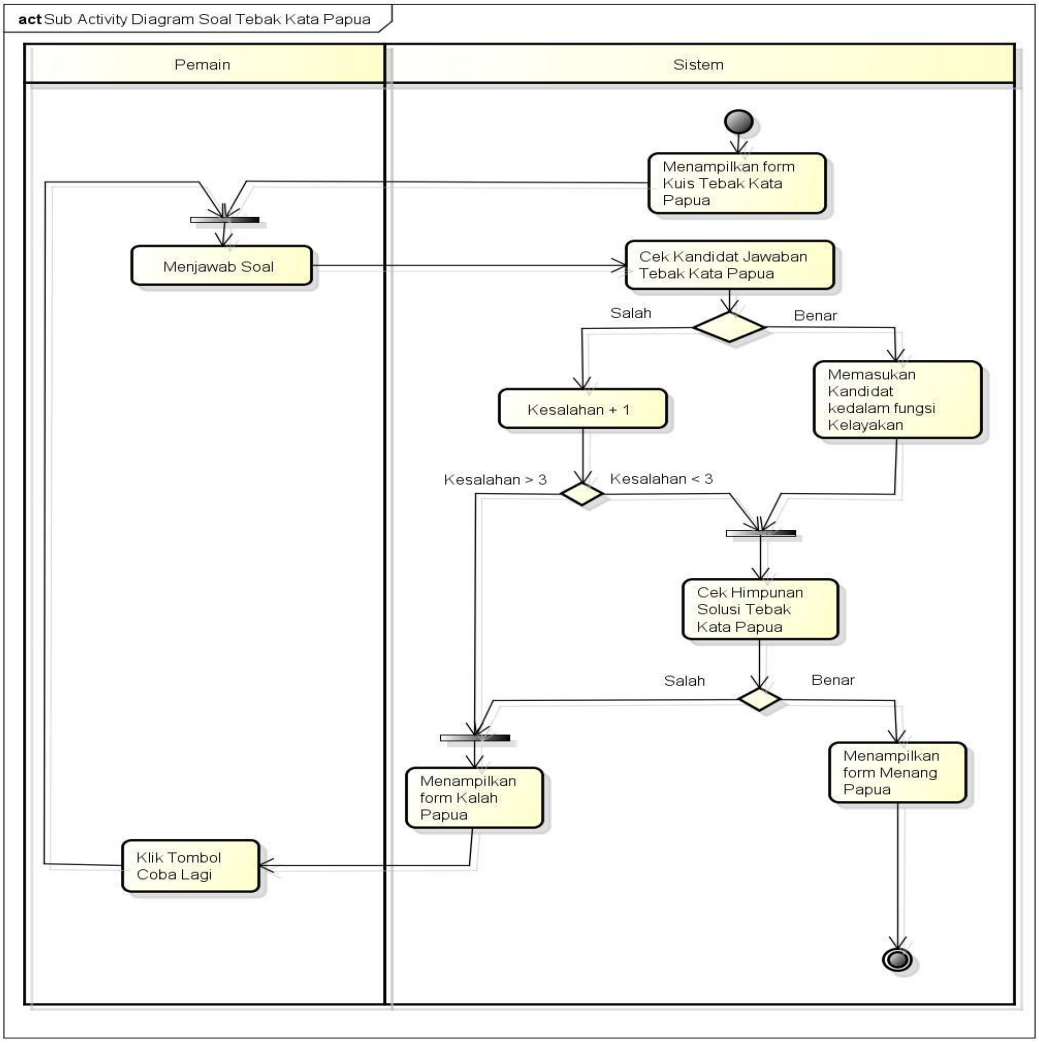
Activity diagram stage papua adalah aktifitas permainan pada *stage* terakhir, dapat dilihat pada gambar 3.9.



Gambar 3. 9 *Activity Diagram* Permainan *stage* Papua

6. *Sub Activity Diagram* Permainan *stage* Papua

Mengambarkan rangkaian aliran dari *sub activity diagram* tebak kata *stage* Papua. Aktivitas ini merupakan bagian dari *activity diagram stage* Papua, digunakan untuk aktivitas menjawab tebak kata *stage* Papua. Dapat dilihat pada gambar 3.10.



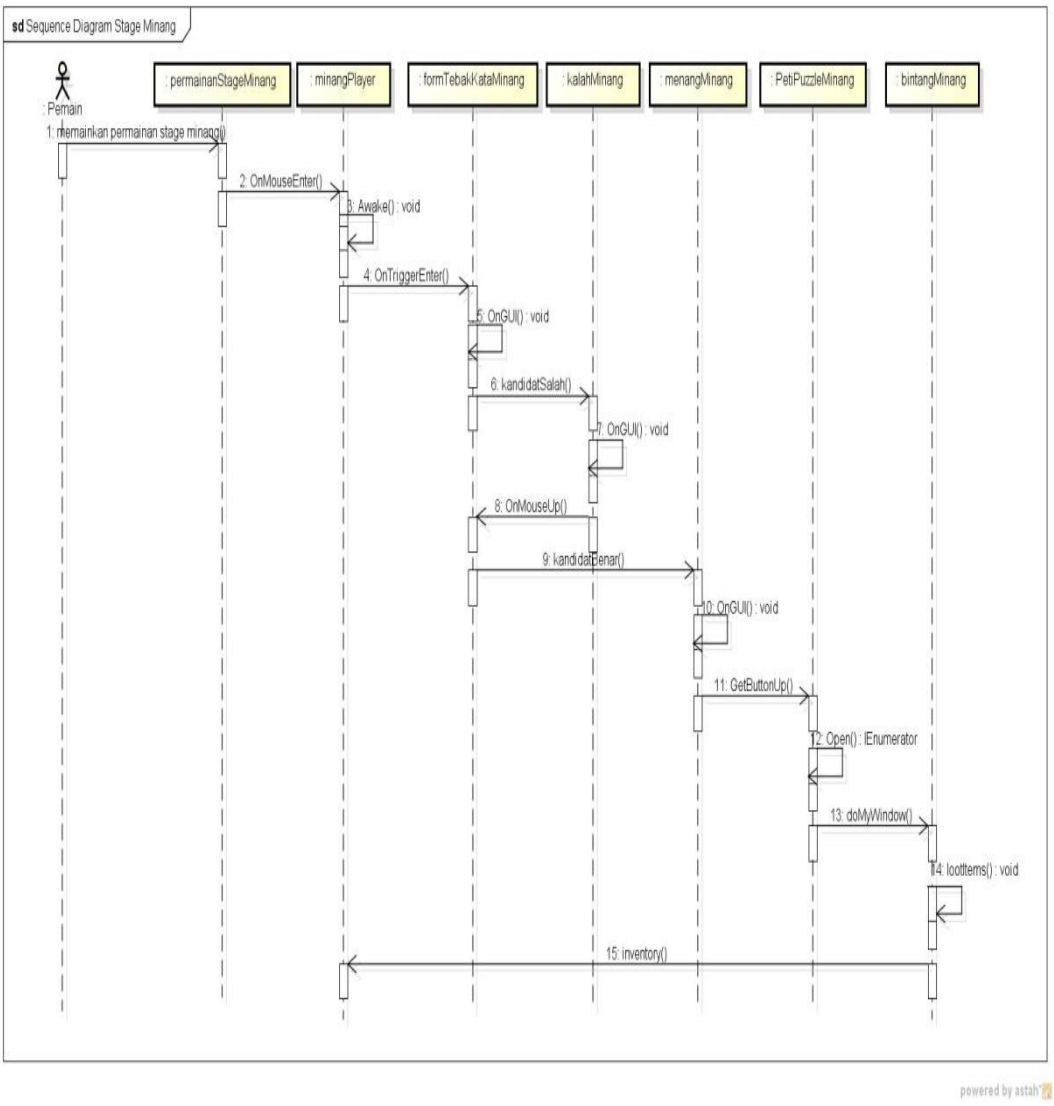
Gambar 3. 10 *Sub Activity Diagram* Permainan *stage* Papua

3.1.5.4. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

1. *Sequence Diagram* Permainan *stage* Minang

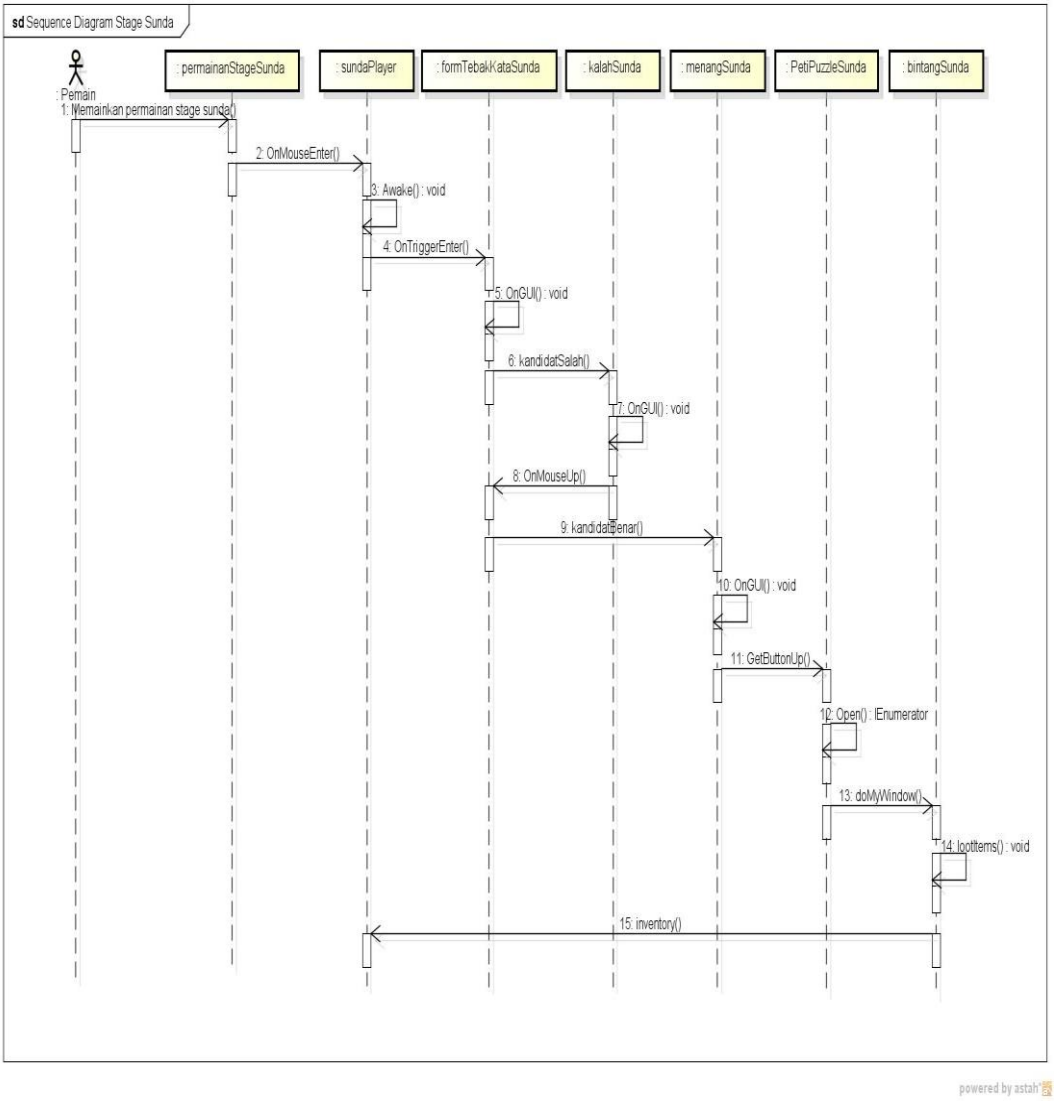
Sequence diagram untuk *Usecase Stage* Minang menggambarkan interaksi antara objek dari *class stage* Minang dan objek yang berkaitan dengan *class stage* Minang. Keseluruhan *sequence stage* Minang dapat dilihat pada Gambar 3.11.



Gambar 3. 11 *Sequence Diagram* Permainan *stage* Minang

2. *Sequence Diagram* Permainan *stage* Sunda

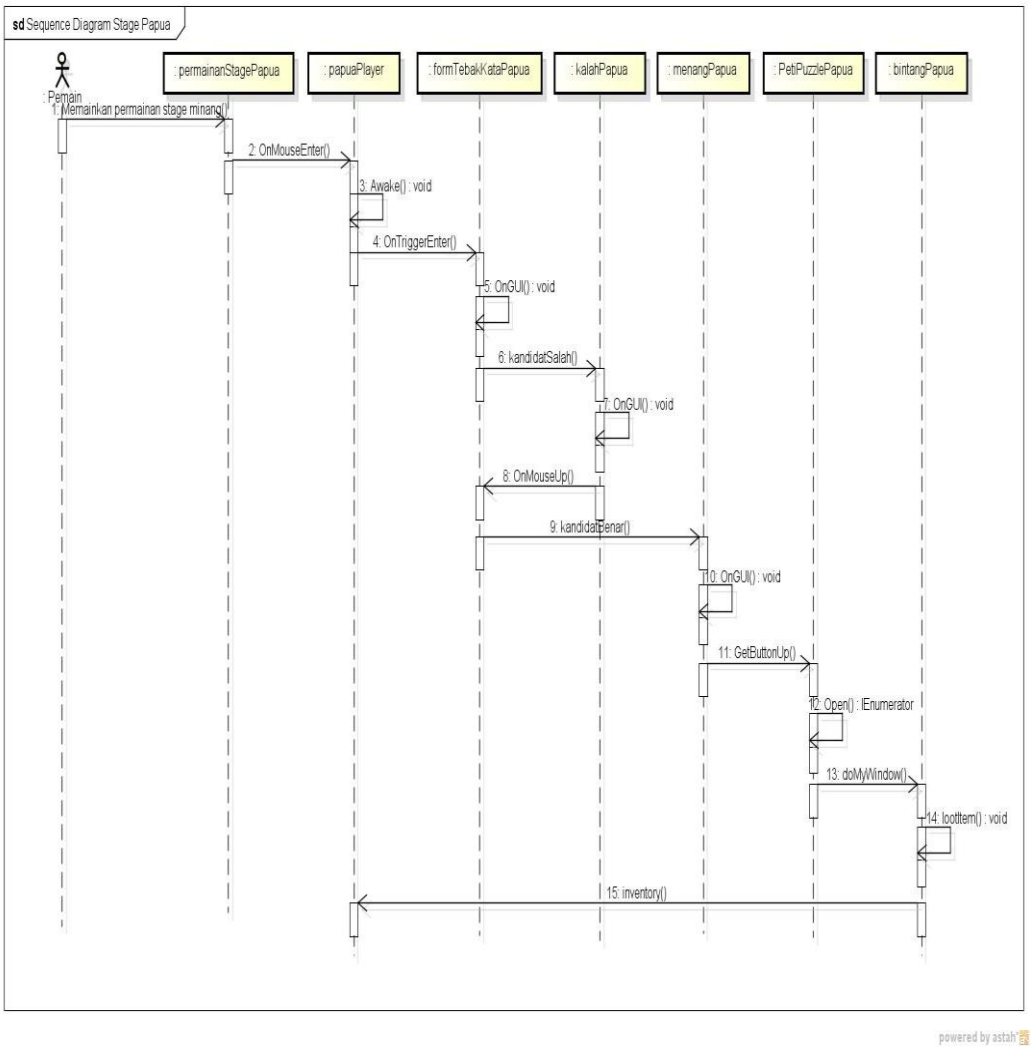
Sequence diagram untuk *Usecase Stage* Sunda menggambarkan interaksi antara objek dari *class stage* Sunda dan objek yang berkaitan dengan *class stage* Sunda. Keseluruhan *sequence stage* Sunda dapat dilihat pada Gambar 3.12.



Gambar 3. 12 *Sequence Diagram* Permainan *stage* Sunda

3. *Sequence Diagram* Permianan *stage* Papua

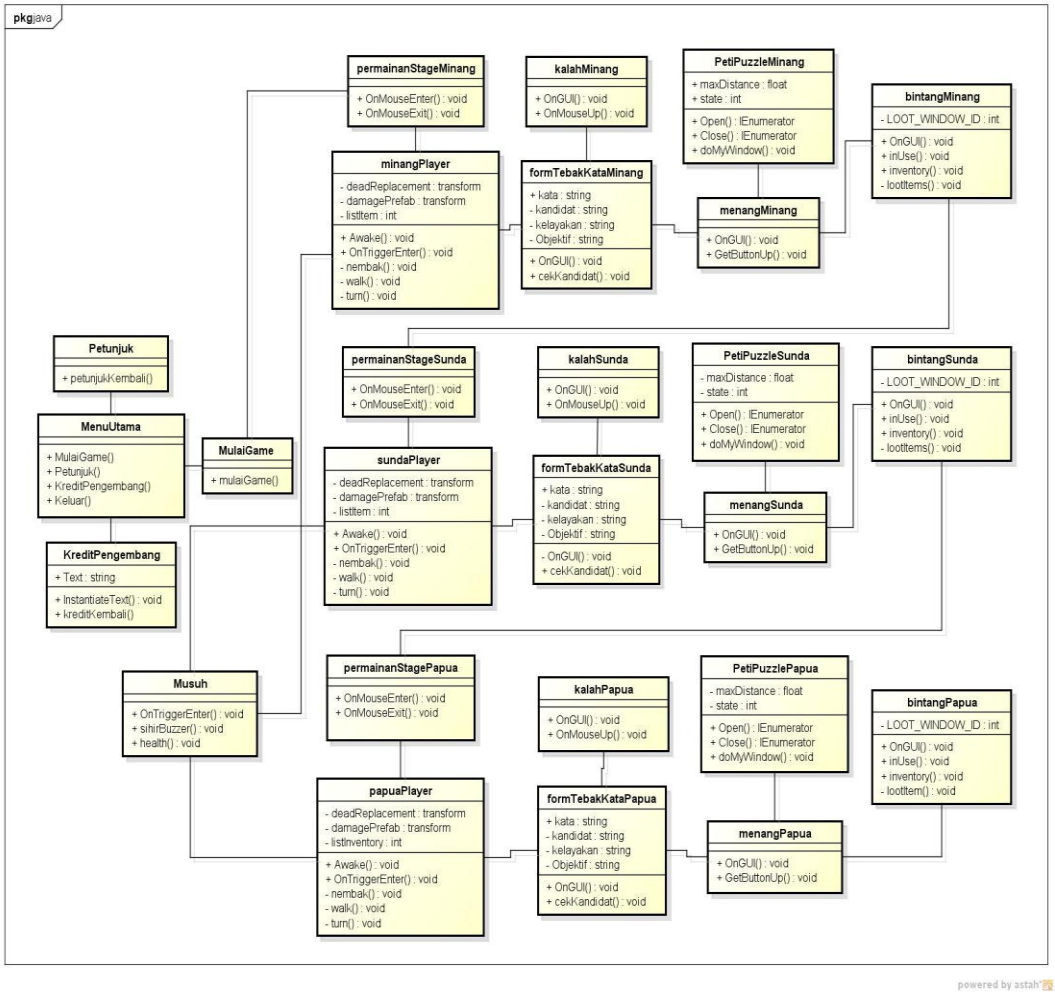
Sequence diagram untuk *Usecase stage* Papua menggambarkan interaksi antara objek dari *class stage* Papua dan objek yang berkaitan dengan *class stage* Papua. Keseluruhan *sequence stage* Papua dapat dilihat pada Gambar 3.13.



Gambar 3. 13 *Sequence Diagram* Permainan *stage* Papua

3.1.5.5. Class Diagram

Diagram kelas atau atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem. Berikut ini adalah perancangan struktur sistem yang terdapat pada *game Puzzle and the Solver* yang digambarkan dalam *class diagram* yang dapat dilihat pada gambar 3.14.



Gambar 3. 14 ***Class Diagram game Puzzle and The Solver***

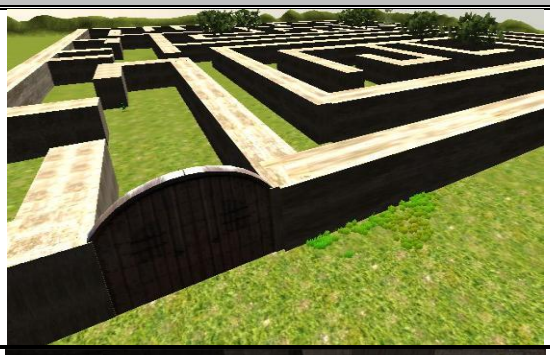
3.2. Perancangan Sistem

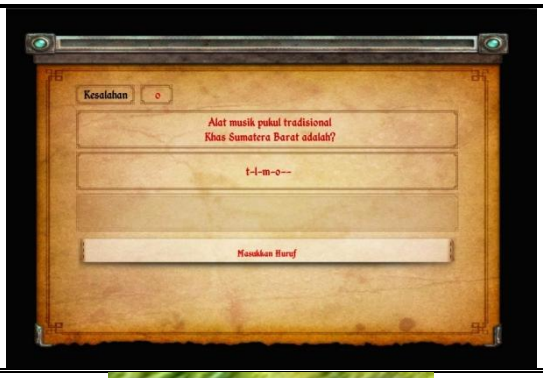
Perancangan adalah suatu bagian dari metodologi pengembangan pembangunan suatu perangkat lunak yang dilakukan untuk memberikan gambaran secara terperinci. Berdasarkan uraian diatas perancangan sistem merupakan tahapan dari siklus pengembangan sistem yang didefinisikan dari kebutuhan-kebutuhan fungsional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

3.2.1. *Storyboard*

Storyboard adalah cerita dalam bentuk naskah dituangkan menjadi bentuk animasi gambar nyata. *Storyboard* merupakan konsep komunikasi dan ungkapan kreatif, teknik dan media untuk menyampaikan pesan gagasan secara visual audio, video dan animasi gerak pada permainan yang akan di bangun.

Tabel 3. 6 *Storyboard*

No	Gambar	Keterangan
1		*Stage* minangkabau adalah *stage* awal permainan. Pada awal permainan pemain akan terkurung didalam sebuah labirin besar.
2	 	Untuk dapat keluar dari labirin, pemain diwajibkan mencari 3 buah Bintang yang berguna untuk membuka pintu labirin. Bintang tersebut bisa didapatkan pemain dari sebuah peti, namun untuk dapat membuka peti tersebut pemain harus menjawab pertanyaan dengan benar.

3		Merupakan *form* kuis tebak kata yang harus dijawab pemain untuk bias mendekati peti Bintang.
4	 	Jika pemain berhasil menjawab dengan benar pertanyaan maka pemain dapat membuka peti bintang dan kumpulkan 3 buah bintang untuk membuka pintu labirin.

a. Pengenalan Karakter

Pengenalan karakter merupakan pembahasan mengenai karakter yang terlibat dalam *game Puzzle and The Solver*.

Tabel 3. 7 Pengenalan Karakter

No	Nama	Gambar	Keterangan
1	Malaka		Pada *stage* pertama pemain akan memainkan karakter malaka, dimana malaka adalah seorang anak keturunan Minangkabau yang sangat menyukai permainan puzzle.

2	Djuanda		Pada *stage* kedua pemain akan memainkan karakter Djuanda, dimana Djuanda adalah seorang anak keturunan Sunda yang sangat menyukai permainan puzzle.
3	Kaisiepo		Pada *stage* ketiga pemain akan memainkan karakter Kaisiepo, dimana Kaisiepo adalah seorang anak keturunan Papua yang sangat menyukai permainan puzzle.
4	Togog		Togog adalah monster penjaga peti bintang. Togog akan sangat marah jika ada yang mengambil bintang Indonesia didalam peti tersebut.
5	Laba-laba		Laba-laba adalah monster yang menjaga labirin agar tidak ada yang berhasil keluar dari labirin tersebut.

b. Pengenalan Objek

Pengenalan objek merupakan pembahasan mengenai objek-objek yang terdapat dalam *game Puzzle and The Solver.*

Tabel 3. 8 Pengenalan Objek

No	Nama	Gambar	Keterangan
1	Bintang Indonesia		Bintang Indonesia berguna untuk membuka pintu labirin.
2	Peti Bintang		Merupakan tempat untuk menyimpan Bintang Indonesia.

3.2.2. Perancangan kuis Tebak Kata *Game Puzzle and The Solver*

Kuis Tebak Kata merupakan misi yang harus diselesaikan pemain untuk bisa keluar dari labirin. Berikut ini adalah perancangan kuis tebak kata pada aplikasi *game puzzle and the solver*.

3.2.2.1. Perancangan Soal dan Jawaban kuis Tebak Kata

Pada aplikasi game *Puzzle and The Solver* ini terdapat 3 *stage* permainan, dimana di tiap *stage* tersebut terdapat tiga kebudayaan berbeda. Kuis tebak kata yang terdapat di setiap *stage* juga berhubungan dengan kebudayaan yang terdapat pada *stage* tersebut. Adapun struktur perancangan soal dan jawaban di setiap *stage* tersebut adalah :

1. *Stage* pertama adalah tahapan yang berhubungan dengan kebudayaan Sumatera Barat. Perancangan soal dan jawaban kuis tebak kata yang terdapat pada *stage* pertama adalah :

Tabel 3. 9 Perancangan Soal dan Jawaban kuis tebak kata Sumatera Barat

No	Soal	Jawaban
1	Alat musik tiup tradisional Sumatera Barat yang terbuat dari bambu adalah?	S A L U A N G
2	Alat musik tiup tradisional Sumatera Barat yang terbuat dari batang padi adalah?	P U P U I K
3	Alat musik tiup tradisional Sumatera Barat yang berbentuk pendek dan terdiri dari 7 lubang suara adalah?	B A N S I
4	Tarian tradisional Sumatera Barat Untuk menyambut tamu kehormatan adalah tari?	P A S A M B A H A N
5	Tarian tradisional sumatera barat dengan menggunakan piring sebagai aksesoris adalah tari?	P I R I A N G
6	Alat musik pukul khas sumatera barat yang biasanya terbuat dari kuningan adalah?	T A L E M P O N G
7	Alat musik pukul khas sumatera barat yang adalah?	G A N D A N G
8	Alat musik gesek khas sumatera barat adalah?	R A B A B
9	Alat musik tiup khas sumatera barat yang ter buat dari dua potong bambu adalah?	S A R U N A I

2. *Stage* kedua adalah tahapan yang berhubungan dengan kebudayaan Jawa Barat. Perancangan soal dan jawaban kuis tebak kata yang terdapat pada *stage* kedua adalah :

Tabel 3. 10 Perancangan Soal dan Jawaban kuis tebak kata Jawa Barat

No	Soal	Jawaban
1	Alat musik getar tradisional Jawa Barat yang terbuat dari bambu adalah?	A N G K L U N G
2	Alat musik getar tradisional Jawa Barat yang terbuat dari bambu atau batang aren adalah?	K A R I N D I N G
3	Alat musik petik tradisional Jawa Barat yang menyerupai gitar adalah?	K A C A P I
4	Sebuah jenis tari pergaulan tradisional masyarakat sunda adalah?	J A I P O N G A N

5	Permainan dan tari tradisional Jawa Barat yang menampilkan barong atau binatang adalah?	B A N G B A R O N G A N
6	Alat musik pukul tradisional Khas Jawa Barat yang terbuat dari tembaga dan dicat keemasan adalah?	G O N G
7	Alat musik gesek tradisional Khas Jawa Barat berbentuk persegi panjang dan lebih sering digunakan sebagai pengiring di acara pencak silat adalah?	T A R A W A N G S A
8	Tarian tradisional Khas Jawa Barat yang dibawakan serta dikembangkan di Cikeruh adalah tari?	C I K E R U H
9	Alat musik gesek tradisional Khas Jawa Barat yang biasanya dimainkan beriringan dengan suara suling dan sinden adalah?	R E B A B

3. *Stage* ketiga adalah tahapan yang berhubungan dengan kebudayaan Papua. Perancangan soal dan jawaban kuis tebak kata yang terdapat pada *stage* ketiga adalah :

Tabel 3. 11 Perancangan Soal dan Jawaban kuis tebak kata Papua

No	Soal	Jawaban
1	Alat musik pukul tradisional Papua yang mirip dengan gendang adalah?	T I F A
2	Alat musik tiup tradisional Papua adalah?	T R I T O N
3	Alat musik tradisi-onal Papua yang berasal dari kata pikonane adalah?	P I K O N
4	Rumah adat khas Papua yang dihuni oleh suku Dani adalah?	H O N A I
5	Binatang yang menjadi ciri Khas Papua adalah?	C E N D R A W A S I H
6	Salah satu suku di papua yang terkenal akan ukiran kayunya yang unik adalah suku?	A S M A T
7	Nama tas tradisional perempuan papua adalah?	N O K E N

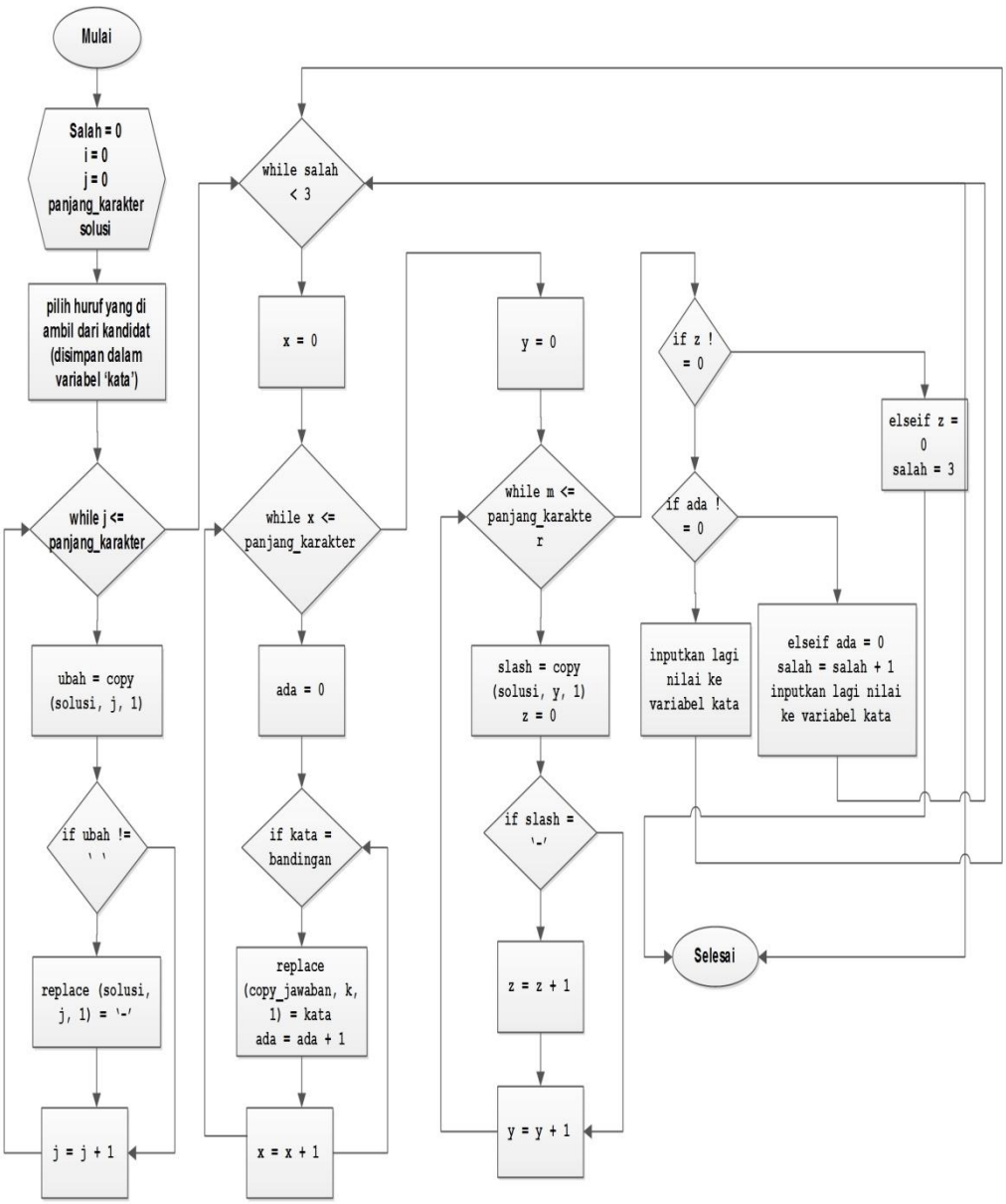
8	Tarian khas papua yang juga diambil dari nama hewan khas provinsi papua\n adalah tari?	C E N D R A W A S I H

3.2.2.2. Perancangan *Flowchart* kuis Tebak Kata

Gambar 3.21 menggambarkan diagram alir algoritma *greedy* yang diimplementasikan pada kuis tebak kata aplikasi *puzzle and the solver*. Penjelasan diagram alir yang sedang berjalan di kuis tebak kata pada aplikasi *puzzle and the solver* adalah sebagai berikut :

1. Mulai
2. Inisialisasi variabel Salah untuk batas kesalahan, variabel i dan variabel j untuk pencacah, variabel panjang_karakter untuk menghitung pangjang karakter jawaban, variabel solusi adalah jawaban dari pertanyaan yang di tampilkan.
3. Proses untuk memilih huruf yang diambil dari kandidat dan disimpan dalam variabel kata.
4. Lakukan perulangan selama variabel j kecil sama dengan panjang karakter.
5. Beri nilai pada variabel ubah yang diambil dari karakter ke j dari variabel solusi.
6. Jika variabel ubah sama dengan ' ' maka tambahkan variabel j.
7. Jika variabel ubah tidak sama dengan ' ' maka ganti dengan karakter '-'.
8. Ulangi langkah 4 jika perulangan masih memenuhi.
9. Selama variabel salah kurang dari 3 beri nilai variabel x sama dengan 0.
10. Selama variabel x kurang sama dengan panjang karakter lakukan perulangan.
11. Beri nilai variabel ada dengan 0.
12. Jika variabel kata sama dengan bandingan maka ganti variabel k dengan variabel kata dan variabel ada sama dengan variabel ada di tambah 1.
13. Proses variabel x sama dengan variabel x ditambah 1.
14. Proses variabel y sama dengan 0.

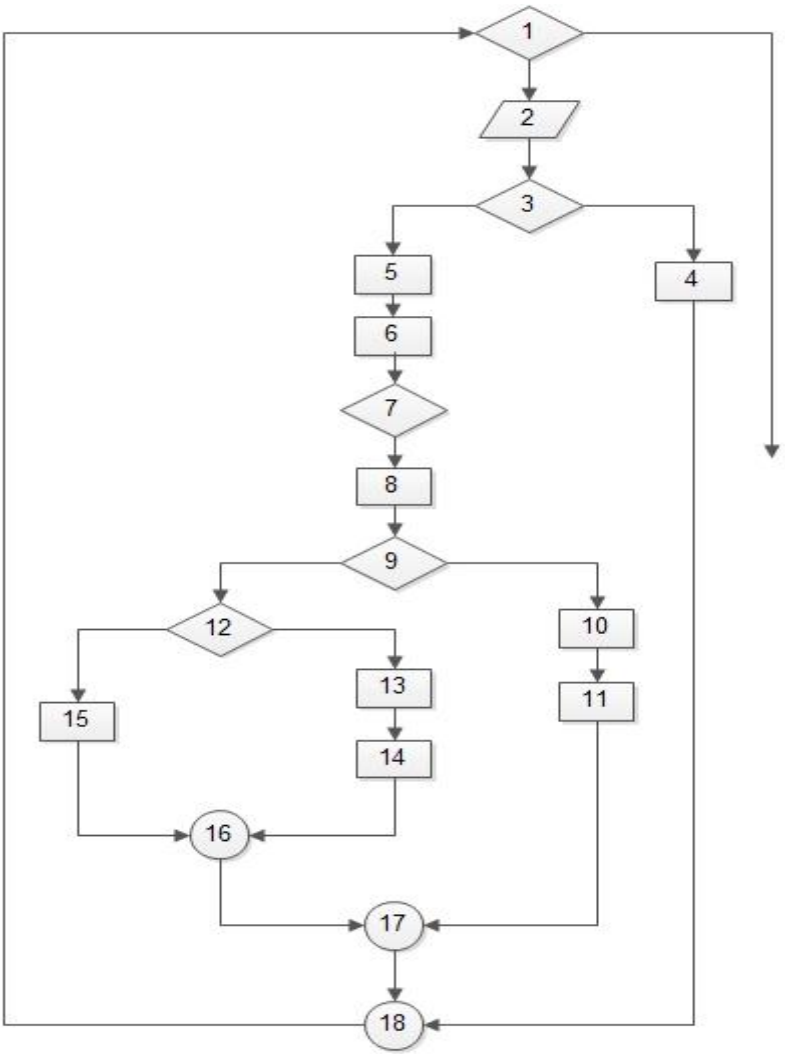
15. Selama variabel m kecil sama dengan panjang karakter lakukan perulangan.
16. Jika slash sama dengan '-' maka variabel z sama dengan variabel z ditambah 1.
17. Proses y sama dengan variabel y ditambah 1.
18. Jika variabel z tidak sama dengan 0 dan jika variabel ada tidak sama dengan 0 maka inputkan lagi nilai ke variabel kata.
19. Jika ada sama dengan 0 maka variabel salah sama dengan salah ditambah satu dan jika variabel z sama dengan 0 maka salah sama dengan 3.



Gambar 3. 15 *Flowchart* Algoritma *greedy*

3.2.2.3. Perancangan *Flowchart* metode *path finding*

Gambar 3.16 menggambarkan diagram alir algoritma A *star* yang diimplementasikan pada musuh dalam aplikasi *puzzle and the solver*. Penjelasan diagram alir metode *path finding* yang sedang berjalan pada aplikasi *puzzle and the solver* dapat dilihat pada tabel 3.15.



Gambar 3. 16 *Flowchart* metode *path finding*

Tabel 3. 12 Keterangan *Flowchart* metode *path finding*

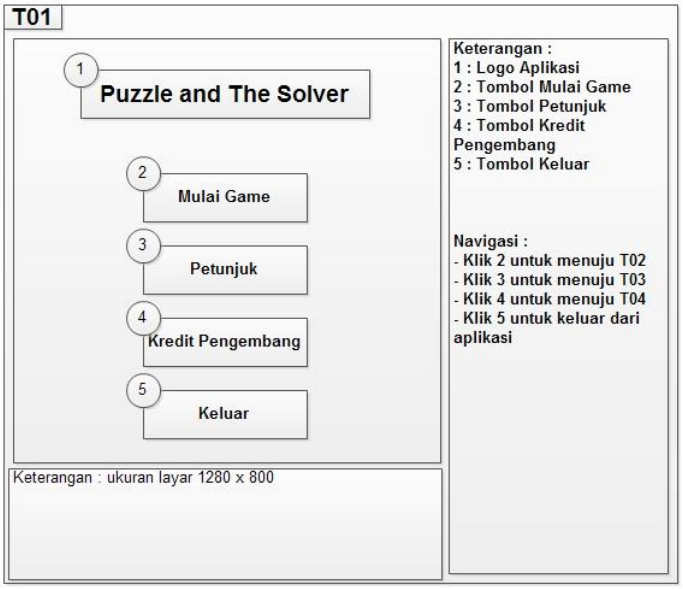
1. Lakukan perulangan selama *open list* tidak kosong
2. Ambil *node* n dari *open list* dengan f(n) terkecil.
3. Jika *node* n sama dengan *node* tujuan, maka target ditemukan
4. Jika tidak masukan *node* n ke dalam *close list* dan mempertimbangkan *node* sebelahnya (n').
5. Untuk setiap node n'
6. Tentukan *parent* dari n' ke n (tentukan nilai h(n'), tentukan nilai g(n'), tentukan nilai f(n'))
7. Jika n' ada di *close list* dengan nilai g(n') lebih kecil
8. Maka *update* n' dengan nilai g(n') terendah yang baru, ubah *parent* dari n' ke *node* sekarang.
9. Jika n' ada di *open list* dengan nilai g(n') lebih kecil.
10. Maka *update* n' dengan nilai g(n') terendah yang baru, ubah *parent* dari n' ke node sekarang.
11. Jika tidak kedua-duanya maka *node* ini tidak ada dalam *close list* dan *open list* (masukan *node* ini ke dalam *open list* dan tentukan nilai g(n)).
12. Ulangi langkah nomor 6.
13. Jika *open list* masih ada ulangi langkah nomor 3.

3.2.3. Perancangan Antarmuka

Antarmuka (*interface*) merupakan bentuk tampilan dari program yang tampil pada layar komputer yang bertujuan memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan memudahkan pengimplementasikan aplikasi sesuai ukuran layar dan memudahkan dalam pembuatan aplikasi.

3.2.3.1 Perancangan Antarmuka Menu Utama

Masuk ke Menu Utama menampilkan 4 tombol pada layar yang akan dipilih oleh pemain sebelum bermain *game*. Antarmuka Menu utama yang dapat dilihat pada Gambar 3.17.



Gambar 3. 17 Perancangan Antarmuka Menu Utama

3.2.3.2 Perancangan Antarmuka Mulai *Game*

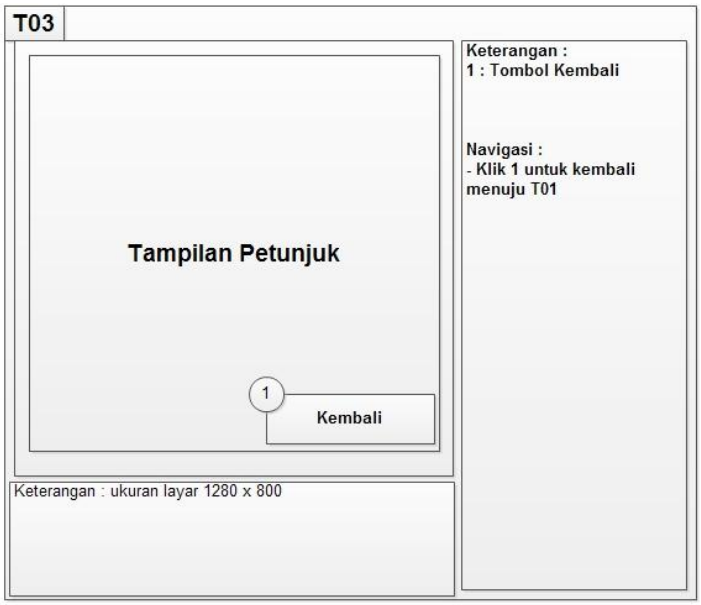
Masuk ke menu Mulai *game* menampilkan area permainan dan tombol kembali ke Menu Utama. Antarmuka menu Mulai *game* dapat dilihat pada Gambar 3.18.



Gambar 3. 18 Perancangan Antarmuka Mulai *Game*

3.2.3.3 Perancangan Antarmuka Petunjuk

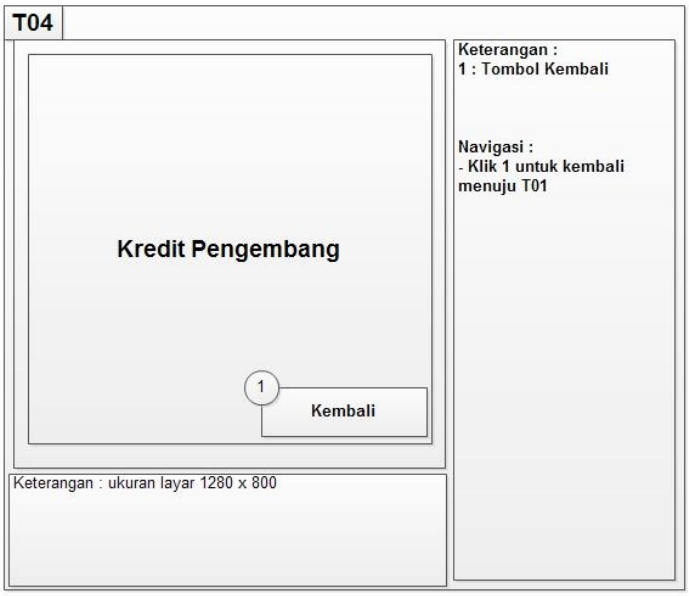
Masuk ke menu Petunjuk akan menampilkan petunjuk permainan yang dapat dilihat pada Gambar 3.19.



Gambar 3. 19 Perancangan Antarmuka Petunjuk

3.2.3.4 Perancangan Antarmuka Menu Kredit Pengembang

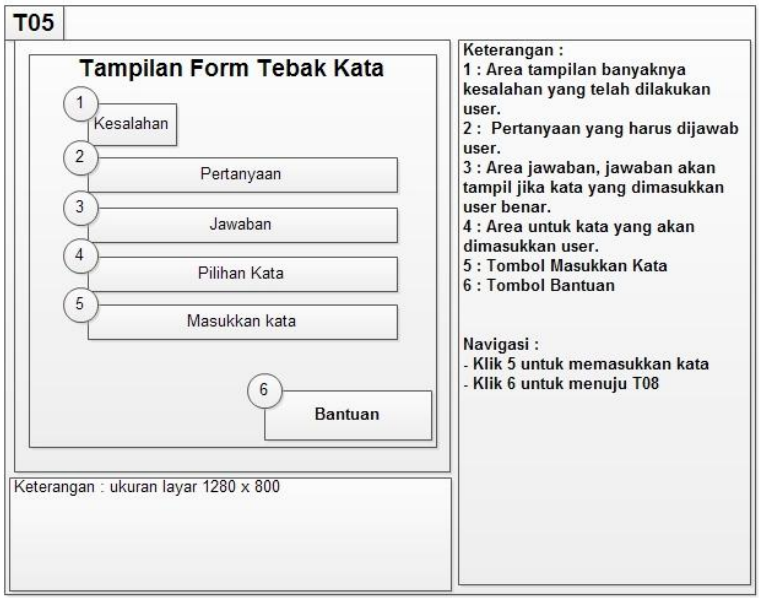
Masuk ke menu Kredit Pengembang akan menampilkan informasi pengembang aplikasi yang dapat dilihat pada Gambar 3.20.



Gambar 3. 20 Perancangan Antarmuka Kredit Pengembang

3.2.3.5 Perancangan Antarmuka Form Tebak Kata

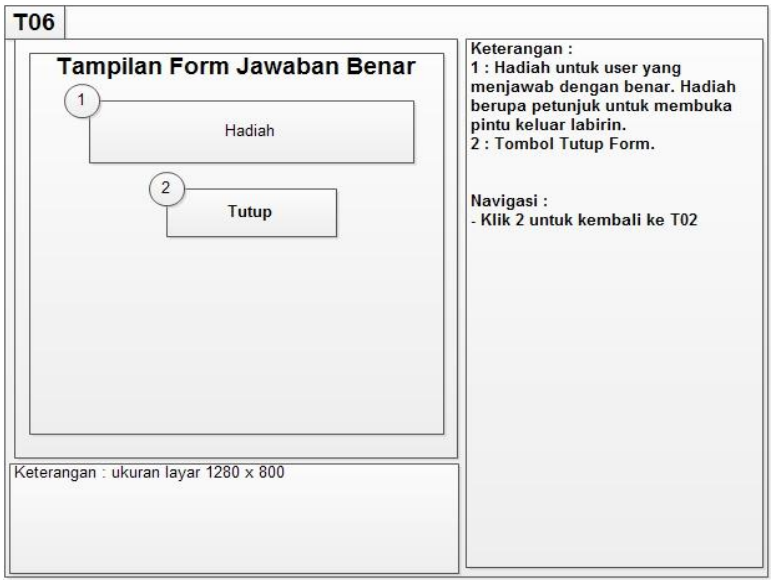
Masuk ke area permainan saat pemain berhasil menemukan peti *puzzle* akan menampilkan form tebak kata yang dapat dilihat pada Gambar 3.21.



Gambar 3. 21 Perancangan Antarmuka *form* Tebak Kata

3.2.3.6 Perancangan Antarmuka *form* Jawaban Benar

Didalam permainan tebak kata jika pemain dapat menjawab dengan benar maka akan keluar *form* kemenangan, dapat dilihat pada gambar 3.22.



Gambar 3. 22 Perancangan Antarmuka *form* Jawaban Benar

3.2.3.7 Perancangan Antarmuka *form* Jawaban Salah

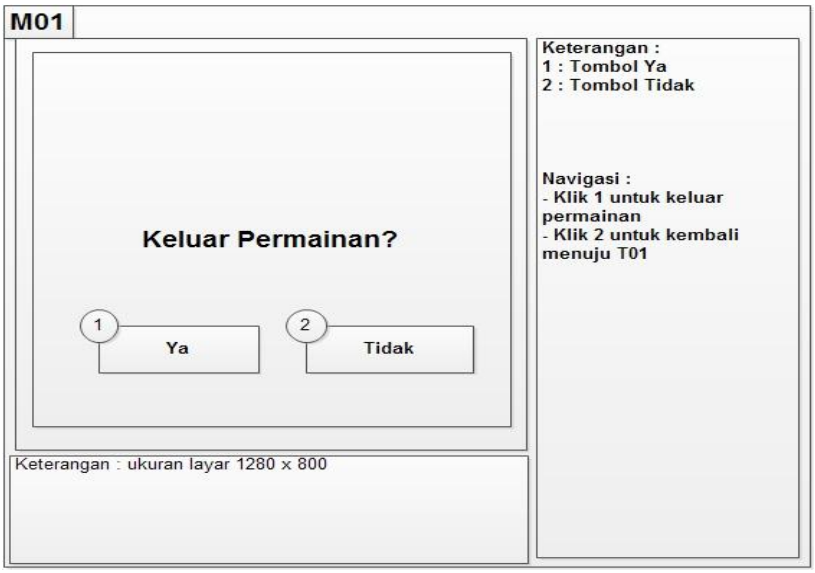
Didalam permainan tebak kata jika pemain tidak dapat menjawab dengan benar maka akan keluar *form* jawaban salah, dapat dilihat pada gambar 3.23.



Gambar 3. 23 Perancangan Antarmuka *form* Jawaban Salah

3.2.3.8 Perancangan Pesan Keluar

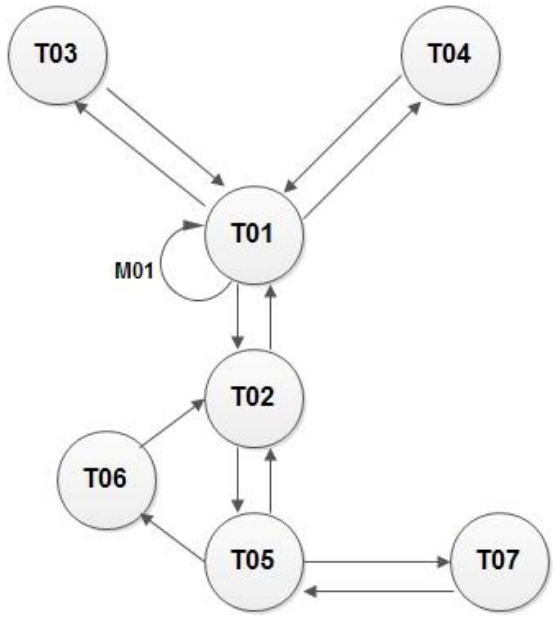
Perancangan pesan saat akan keluar dari aplikasi dapat dilihat pada gambar 3.24.



Gambar 3. 24 Perancangan Pesan Keluar

3.2.4. *Jaringan Semantik*

Jaringan semantik menggambarkan hubungan antara tampilan satu dengan tampilan lainnya. Jaringan semantik dapat dilihat pada Tabel 3.13. sebagai referensi kode tampilan dan Gambar 3.25.



Gambar 3. 25 Jaringan Semantik *Puzzle and The Solver*

Tabel pengkodean tampilan.

Tabel 3. 13 Pengkodean Tampilan

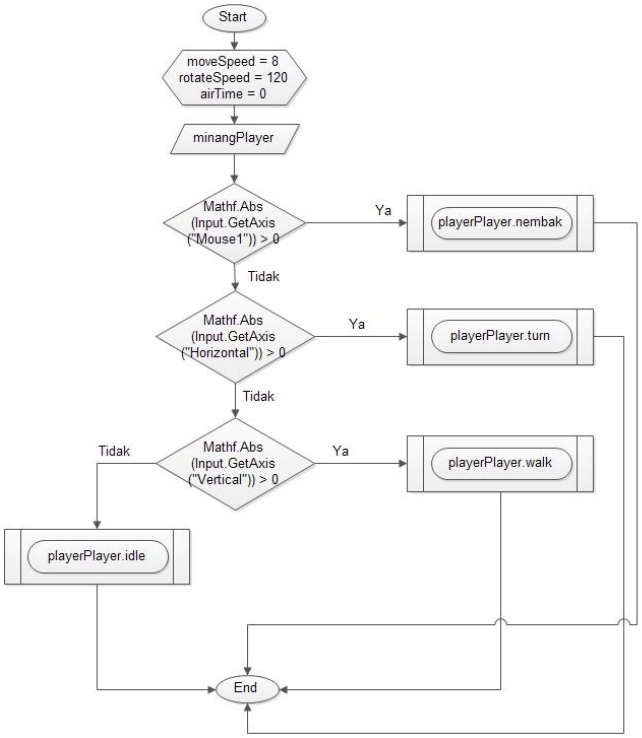
Kode Tampilan	Nama Tampilan
T01	Perancangan antarmuka Menu Utama
T02	Perancangan antarmuka Mulai *Game*
T03	Perancangan antarmuka Petunjuk
T04	Perancangan antarmuka Menu Kredit Pengembang
T05	Perancangan antarmuka Form Tebak Kata
T06	Perancangan antarmuka Form Jawaban Benar
T07	Perancangan antarmuka Form Jawaban Salah
M01	Perancangan pesan Keluar Permainan

3.2.5. Perancangan *Method*

Prosedur perlu dirancang untuk mengetahui alur proses dari program, yang mewakili prosedur di dalam program yang dibangun. Berikut ini adalah prosedur perancangan yang dilakukan untuk membangun aplikasi *game puzzle and the solver*.

3.2.5.1. Perancangan *Method* minangPlayer

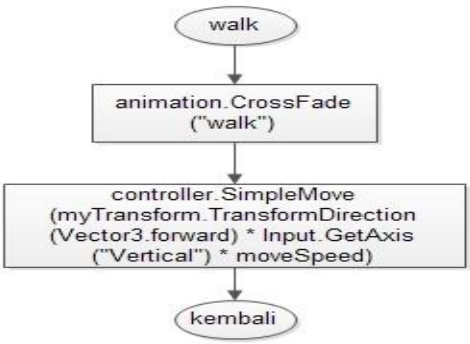
Alur proses perancangan *method* minangPlayer dapat dilihat dapat gambar 3.26.



Gambar 3. 26 Perancangan *Method* minangPlayer

3.2.5.2. Perancangan *Method* minangPlayer.walk

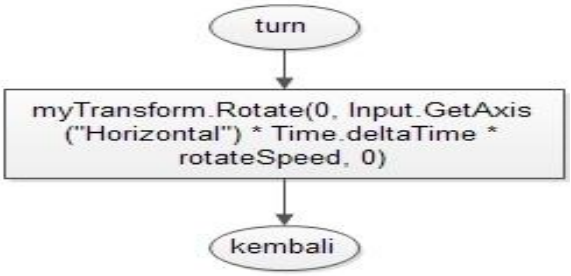
Method ini digunakan untuk menjalankan animasi *walk* pada player, dapat dilihat pada gambar 3.27.



Gambar 3. 27 Perancangan *Method* minangPlayer.walk

3.2.5.3. Perancangan *Method* minangPlayer.turn

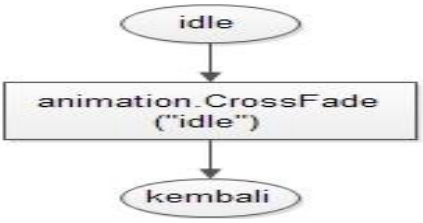
Method ini digunakan untuk menjalankan animasi *turn* pada player, dapat dilihat pada gambar 3.28.



Gambar 3. 28 Perancangan *Method* minangPlayer.turn

3.2.5.4. Perancangan *Method* minangPlayer.idle

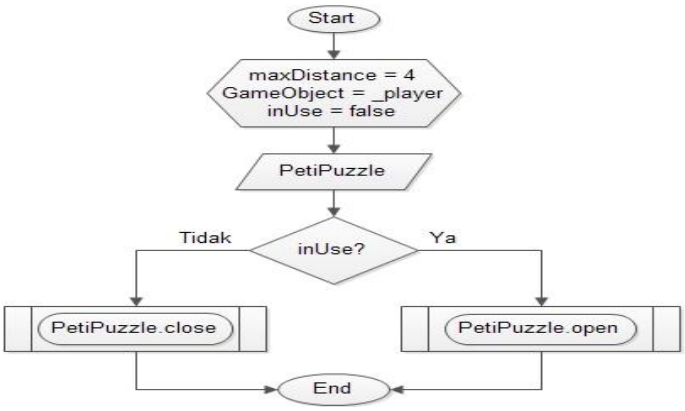
Method ini digunakan untuk menjalankan animasi *idle* pada player, dapat dilihat pada gambar 3.29.



Gambar 3. 29 Perancangan *Method* minangPlayer.idle

3.2.5.5. Perancangan *Method* PetiPuzzle

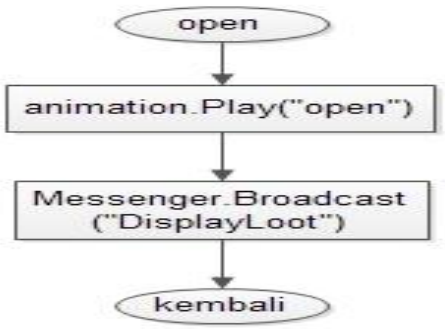
Alur proses perancangan *class* PetiPuzzle yang terdapat pada *stage* minang, sunda dan papua dapat dilihat pada gambar 3.30.



Gambar 3. 30 Perancangan *Method* PetiPuzzle

3.2.5.6. Perancangan *Method* PetiPuzzle.open

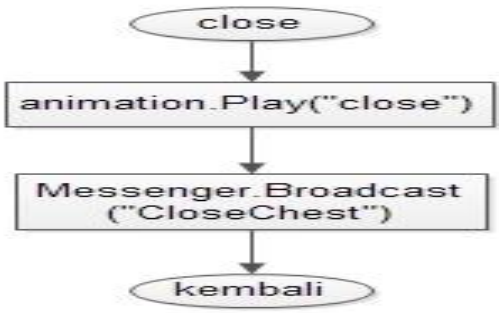
Method ini digunakan untuk memainkan animasi open pada peti puzzle, dapat dilihat pada gambar 3.31.



Gambar 3. 31 Perancangan *Method* PetiPuzzle.open

3.2.5.7. Perancangan *Method* PetiPuzzle.close

Method ini digunakan untuk memainkan animasi open pada peti puzzle, dapat dilihat pada gambar 3.32.



Gambar 3. 32 Perancangan *method* PetiPuzzle.close