BAB 3

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Pada analisis sistem akan mencakup analisis masalah, analisis kebutuhan non-fungsional dan analisis kebutuhan fungsional.

3.1.1 Analisis Masalah

Menurut Lam [3], mobile *game* merupakan *game* yang tidak hanya dapat dimainkan melalui telepon seluler, namun dapat dikembangkan dalam berbagai macam mobile handset seperti PDA, symbian OS dan Microsoft Smartphone. Mobile *game* dimainkan dengan menggunakan teknologi yang ada di dalam alat itu sendiri.

Mobile *game* yang saat ini mulai berkembang pesat alangkah baiknya dijadikan suatu alat untuk meningkatkan kecintaan anak bangsa terhadap kebudayaanya sendiri yang saat ini mulai ditinggalkan dan terlupakan, mobile *game* yang menarik dan mempunyai tingkat kesulitan yang diharpakan membuat banyak anak muda memakai dan menjadikan pembelajaran yang sangat efektif. *Game* mobile saat ini sangat banyak dipasaran tetapi jarang sekali yang berunsur kebudayaan sendiri.

Modernisasi ini mempengaruhi stabilitas kebudayaan di berbagai daerah, termasuk kebudayaan Sunda. Kebudayaan asli masyarakat Jawa Barat ini semakin lama semakin tergilas. Sungguh disayangkan, padahal kebudayaan Sunda itu banyak mengandung nilai-nilai filosofis dan termasuk salah satu kebudayaan tertua di Indonesia [12]. Sehinggan diangkatnya kebudayaan sunda didalam game ini untuk menumbukan kembali kecintaan terhadap kebudayaan sunda itu sendiri.

3.1.2 Pengenalan dan analisis *game* sejenis "*Game angry birds dan cat and dog"*

Game dengan tipe *gameplay casual* sudah tidak asing lagi dalam dunia *game*, perkembangannya drastis meningkat karena tipe ini tetap saja menarik

untuk dimainkan karena *game*playnya yang simple membuat orang-orang mudah untuk memainkannya, banyak *game* yang memakai sistem *gameplay casual* ini, beberapa diantaranya yaitu *angry birds, crazy pinguin, cat and dog, perjuangan semut.*

Dalam pembahasan, penulis akan membahas *game* dengan jenis *gameplay casual* dan *game* edukasi. *Game* yang akan dibahas dan menjadi acuan disini ialah *game angry birds* dan *game cat and dog,* secara *game*play *game angry birds* serupa dengan *gameplay game* kebudayaan sunda yang akan di bangun oleh penulis. oleh karena itu penulis mencoba menggabungkan tipe *gameplay casual* dengan menerapkan unsur edukasi. Berikut ulasan detail mengenai *game angry birds* dan *game cat and dog.* Dan penerapan metode AI pada kekuatan angin.

The angry birds adalah *game* dengan genre casual. *Game* yang dibuat menggunakan *platform flash* (swf). Dalam *game* ini, karakter utamanya adalah seorang burung dengan berbagai macam bentuk.

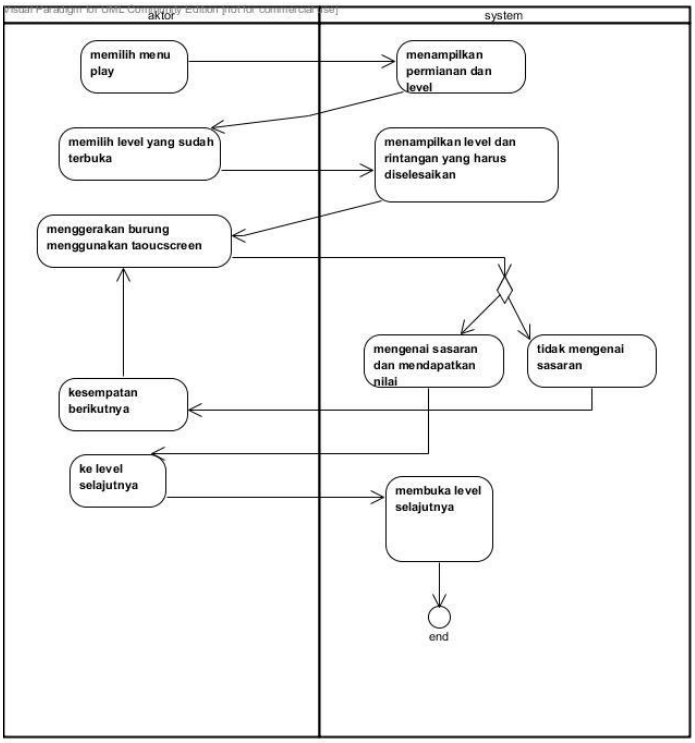
3.1.2.1 *Gameplay Game angry birds*

Beberapa aturan dalam *game angry birds*, adalah sebagai berikut:

1. Karakter adalah seoarng burung yang dilempar menggunakan ketapel, dengan sasaran beberapa ekor babi.
2. Dalam tiap levelnya burung akan bertemu dengan berbagai macam karater musuh(babi) dalam beberapa kesulitan, untuk melawan musuh ini burung mendapatkan bantuan diantaranya beberapa burung bisa bervariasi untuk mengenai sasaran , jika karakter tidak mengenai sasaran makan si user mendapatkan point kecil.
3. Dalam setiap episode akan ada papan petunjuk yang menampilkan informasi yang harus dilakukan oleh karakter.
4. Dalam *game* ini ada point-point bintang jika menyelesaikan semua.

3.1.2.2 *Activity diagram main* pada *game angry bird*

Activity diagram menu pada game angry bird dilihat pada gambar *3.1*



Gambar 3.1 *activity Game The angry birds.*

3.1.2.3 Tampilan *Game The angry birds*.

Tampilan *game cat and dog* dapat dilihat pada gambar 3.2



Gambar 3.2 Tampilan *Game The angry birds.*

3.1.3 Pengenalan *Game cat and dog*

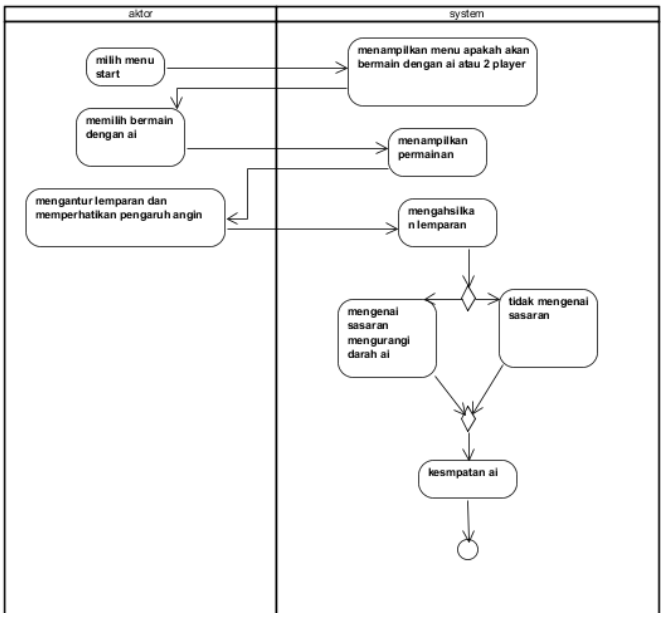
Game cat and dog adalah *game* yang menerapkan unsur kekutan angin dalam dalam tingkat kesuiltan *game*nya.

3.1.3.1 *Gameplay Game* cat and dog

Permainan ini mengisahkan tentang seorang kucing dan anjing yang harus mengasah kemampuan mereka dalam menembak sasaran. Kucing dan anjing tersebut diberikan sebuah benda untuk dikenakan kemasing-masing lawan. Apabila dia berhasil menghabisakan darah salah satu maka di anggap menang *game* ini dalam memaikannya memerlukan prediksi kekuatan angin yang mempengaruhi kekuatan pemain sehingga cukup sulit.

3.1.3.2 *Activity diagram main menu* pada *game cat and dog*

Tampilan *activity cat and dog* dapat dilihat pada gambar 3.3



Gambar 3.3 Tampilan *activity diagram Game cat and dog.*

3.1.3.3 Tampilan *Game Cat and Dog*

Tampilan *game cat and dog* dapat dilihat pada gambar 3.4



Gambar 3.4 Tampilan Permainan *Game* cat and dog

3.1.4 Pengenalan *Game* Edukasi kebudayaan sunda berbasis *mobile*

Game yang akan dibangun berjudul mengenal kebudayaan sunda yang bertipe *casual*. *Game* ini dibangun dengan mengaplikasikan teknologi sebagai jembatan untuk menyajikan materi mengenai kebudayaan sunda secara *interaktif* dan *edukatif*. *Game* ini dibangun dengan grafis 2D (2 Dimensi).

Berikut ini adalah fitur-fitur yang ada pada *game* ini:

1. Sistem *single player*.
2. Mengangkat tema kebudayaan sunda.
3. Grafik *game* 2D.
4. Menerapkan logika fuzzy dalam kekuatan angin.
5. Jenis *game* bertipe *casual* yang disajikan secara interaktif.
6. Aplikasi *game* bersifat bermain dan belajar karena dalam *game* disajikan mengenai informasi-informasi kebudayaan sunda.

3.1.4.1 Storyline / keunikan secara umum

Game edukasi kebudayaan sunda berbasis *mobile* ini mempunyai keunikan yaitu game gameplay sederhana dan mempunyai control karakter yang mudah bagi semua pengguna , acuan anank-anak sebagai target utama akan

membuat anak tertarik memaikannya disamping itu game ini juga member pengetahuan umum tentang kebudayaan sehingga anak diajarkan sejak dini mengenal kebudaayannya, disamping untuk melestarikan budaya sendiri untuk memupuk kecintaan terhadap kebudayaan , ketapel yang di gunakan sebagai control karakter user diambil dari game yang sedang marak digunakan sehinnga hafal betl cara mnggunakannya. menceritakan bagaimana user mencari ilmu dengan melemparkan batu kepada objek sehingga mendapatkan informasi dalam perjalan setiap level user mendapat rintangan berupa angin yang berubah-ubah setiap melempar :

Level 1

Di sini user akan diberikan 6 batu mengenai sasaran. Didalam level 1 materi yang ada adalah seni wayang golek, alat musik angklung, dan tari jaipong. Dan pada setiap user mengenai sasaran akan diberikan informasi bertemakan apa yang didapatkan dan pada level 1.Letak kesulitan berada pada objek sasaran benda berada yang bervariasi disetiap levelnya.

1. Level 2

   Di sini user akan diberikan 5 batu untuk tepat mengenai sasaran. Didalam level 2 materi yang ada adalah seni degung,kecapi suling, pencak silat. Dan pada setiap user mengenai sasaran akan diberikan informasi bertemakan apa yang didapatkan dan pada level 2. Letak kesulitan berada pada objek sasaran benda berada yang bervariasi disetiap levelnya, pada level 2 letak objek agak sulit ditembak.

2. Level 3

   Di sini user akan diberikan 4 batu untuk mengenai sasaran .level 3 materi yang ada adalah kesenian sisingan,kuda lumping , seni Reog. Dan pada setiap user mengenai sasaran akan diberikan informasi bertemakan apa yang didapatkan dan pada level 3. Letak kesulitan berada pada objek sasaran benda berada yang bervariasi disetiap levelnya. Pada level 3 letak objek sasaranya menjadi sulit utuk ditembak.

3.1.4.2 *Game*play

Tugas utama pemain dalam memainkan *game* ini adalah menembak sasaran dan mengumpulkan informasi, pada setiap sasaran akan tampil informasi mengenai kebudayaan sunda, pada tiap tingkatan level mendapatakan kesulitan yang berbeda pengaruh angin yang ada didalam *game* akan mempengaruhi tingkat ketepatan sehingga lebih menarik.

Tipe permainan berupa *game casual* dimana pemain dapat membidik dan memberikan tenaga untuk mengenai target.

3.1.5 Analisis logika fuzzy

3.1.5.1 Angin

Angin terjadi karena adanya perbedaan tekanan udara atau perbedaan suhu udara pada suatu daerah atau wilayah. Hal ini berkaitan dengan besarnya energi panas matahari yang di terima oleh permukaan bumi. Pada suatu wilayah, daerah yang menerima energi panas matahari lebih besar akan mempunyai suhu udara yang lebih panas dan tekanan udara yang cenderung lebih rendah. Sehingga akan terjadi perbedaan suhu dan tekanan udara antara daerah yang menerima energi panas lebih besar dengan daerah lain yang lebih sedikit menerima energi panas, akibatnya akan terjadi aliran udara pada wilayah tersebut.

Kategori angin dapat dilihat pada tabel berikut dengan sekala beafort [5] :

Tabel 3.1 Tabel skala beaufort

Skala Beaufort	**Kategori**	**Satuan dalam km/jam**	**Satuan dalam knots**
0	Udara Tenang	0	0
1~3	Angin Lemah	≤ 19	≤ 10
4	Angin Sedang	20~29	11~16
5	Aangin Segar	30~39	17~21
6	Angin Kuat	40~ 50	22~ 27

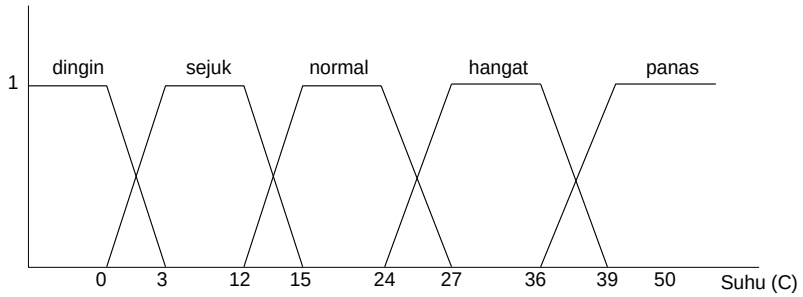
7	Angin Ribut	51~ 62	28 ~33
8	Angin Ribut sedang	63~ 75	34~ 40
9	Angin Ribut Kuat	76~ 87	41~ 47
10	Badai	88~ 102	48~ 55
11	Badai Kuat	103 ~117	56~ 63
12+	Topan	[3]118	[3]64

3.1.5.2 Penerapan logika fuzzy untuk penentuan kecepatan angin

Teori tentang *fuzzy* set dan *fuzzy logic* banyak digunakan untuk membangun system berbasis aturan fuzzy untuk masalah kontrol.

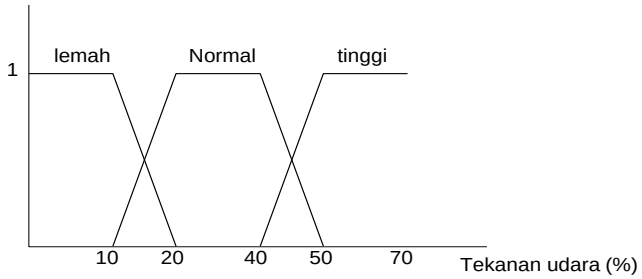
Proses fuzzification

Suhu udara menggunakan fungsi keanggotaan trapesium dengan lima variabel : dingin , sejuk , normal , hangat , panas [1].



Gambar 3.5 Fungsi keanggotan trapezium untuk suhu udara.

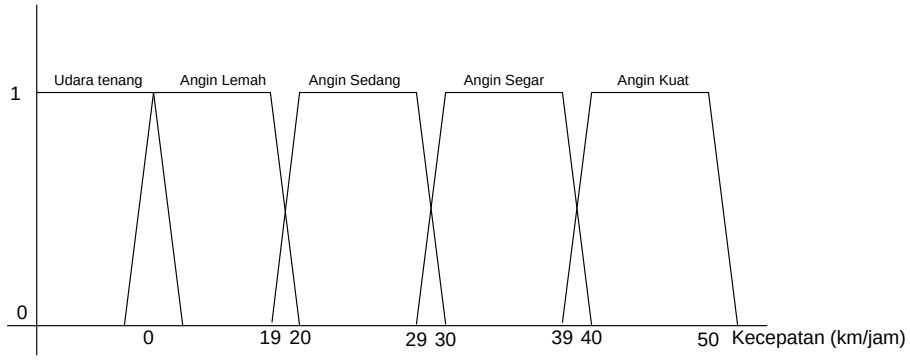
Dan untuk tekanan udara menggunakan fungsi keangotaan trapezium dengan 3 variabel : lemah , normal , tinggi [1].



Gambar 3.6 Fungsi keanggotaan trapezium untuk tekanan udara.

Proses inferensi

Terdapat berbagai macam cara dalam menentukan aturan fuzzy . misalkan, untuk menetukan kecepatan angin, menggunakan fungsi keanggotanaan trapesium dengan lima nilai linguistik : udara tenang , angin lemah , angin sedang, angin segar , angin kuat.



Gambar 3.7 Fungsi keanggotaan trapesium untuk kecepatan angin

Table 3.2 Aturan fuzzy untuk masalah kecepatan angin

Suhu / tekanan	Dingin	Sejuk	Normal	Hangat	Tinggi
Rendah	Udara tenang	Udara tenang	Angin lemah	Angin lemah	Angin lemah
Sedang	Angin sedang	Angin sedang	Angin segar	Angin segar	Angin segar
Tinggi	Angin segar	Angin kuat	Angin kuat	Angin kuat	Angin kuat

Dengan definisi aturan fuzzy di atas kita mempunyai 3 x 5 = 15 aturan fuzzy yaitu:

If *suhu* = dingin AND *tekanan* = Rendah THEN angin = udara tenang

If *suhu* = sejuk AND *tekanan* = Rendah THEN angin = udara tenang

If *suhu* = normal AND *tekanan* = Rendah THEN angin = Angin lemah

If *suhu* = hangat AND *tekanan* = Rendah THEN angin = Angin lemah

If *suhu* = panas AND *tekanan* = Rendah THEN angin = Angin sedang

If *suhu* = dingin AND *tekanan* = sedang THEN angin = Angin sedang

If *suhu* = sejuk AND *tekanan* = sedang THEN angin = Angin sedang

If *suhu* = normal AND *tekanan* = sedang THEN angin = Angin segar

If *suhu* = hangat AND *tekanan* = sedang THEN angin = Angin segar

If *suhu* = panas AND *tekanan* = sedang THEN angin = Angin segar

If *suhu* = dingin AND *tekanan* = tinggi THEN angin = Angin segar

If *suhu* = sejuk AND *tekanan* = tinggi THEN angin = Angin kuat

If *suhu* = normal AND *tekanan* = tinggi THEN angin = Angin kuat

If *suhu* = hangat AND *tekanan* = tinggi THEN angin = Angin kuat

If *suhu* = panas AND *tekanan* = tinggi THEN angin = Angin kuat

Flowchart logika fuzzy



Gambar 3.8 flowchart logika fuzzy

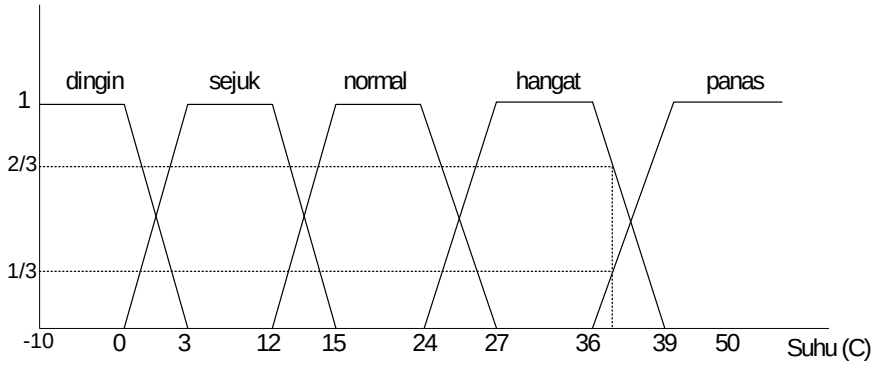
3.1.5.3 Contoh kasus logika fuzzy pada kekuatan angin

Misalkan, nilai crips yang diterima oleh system suhu adalah 37^{O}C dan nilai crips yang diterima system tekanan udara adalah 12% , apakah anginyang di dapatkan?

a. Proses fuzzification

   Misalkan suhu udara menggunakan fungsi keanggotaan trapesium dengan lima variabel lingustik : *dingin*, *sejuk*, *Normal*, *hangat* dan *panas*, dengan fungsi ini , maka crips input suhu 37^{o}C dikonversi ke nilai fuzzy dengan cara :

- Suhu 37^{o}C berada pada nilai lingustik hangat dan panas.
- Semantik atau derajat keanggotaan untuk hangat dihitung menggunakan rumus *-(x-d)/(d-c),c<x≤d* (lihat notasi matematika untuk fungsi trapesium), dimana c=36, dan d=39 sehingga derajat keanggotaan untuk hangat = -(37-36)/(39-36)=2/3
- Derajat keanggotaan untuk panas dihitung menggunakan rumus *(x-a)/(b-a),a<x<b,* dimana a=36 dan b=39. Sehingga derajat keanggotaan untuk panas = (37-36)/(39-36)=1/3.
- µhangat(37)= *-(x-d)/(d-c),c<x≤d*
  = -(37-36)/(39-36)=2/3
- µpanas(37)= *(x-a)/(b-a),a<x<b*
  = (37-36)/(39-36)=1/3.

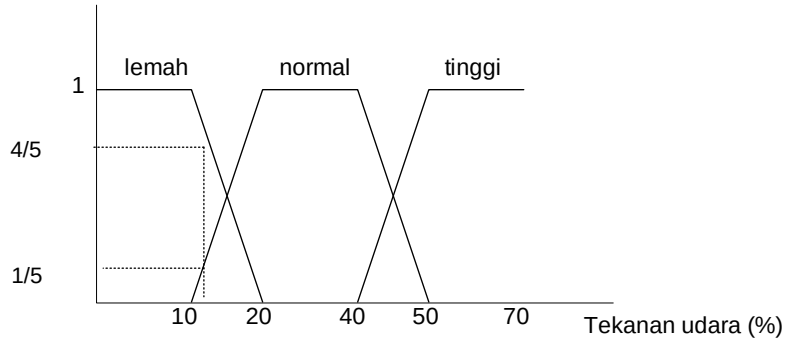


Gamabr 3.9 Fungsi keanggotan trapezium untuk suhu udara.

Misalkan menggunakan fungsi keanggotaan trapesium untuk tekanan udara . dengan fungsi ini, maka crips input tekanan udara 12% dikonversi menjadi nilai fuzzy dengan cara berikut ini:

- Tekanan udara 12% berada pada nilai linguistik lemah dan normal.
- Semantik atau derajat keanggotaan untuk lemah dihitung dengan menggunakan rumus *–(x-d)/(d-c),c<x≤d,* dimana c=10, dan d=20 sehingga derajat keanggotaan untuk lemah adalah –(12-20)/(20-10)=4/5.

- Derajat keanggotaan untuk normal dihitung dengan menggunakan rumus *(x-a)/(b-a),a<x<b*, diamana a=10 dan b 20 . sehingga derajat keanggotaan lemah =(12-10)/(20-10)=1/5.
- μ(lemah) = *–(x-d)/(d-c),c<x≤d*
  = -(12-20)/(20-10)=4/5.
- μ(normal) = *(x-a)/(b-a),a<x<b*
  =(12-10)/(20-10)=1/5.

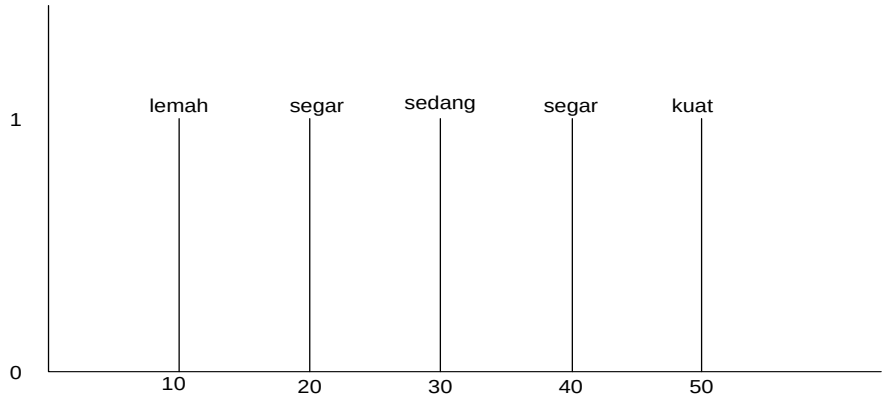


Gambar 3.10 Fungsi keanggotaan trapezium untuk tekanan udara

Jadi proses fuzzification menghasilkan empat fuzzy input : suhu udara = hangat (2/3) dan panas (1/3), dan tekanan udara = lemah (4/5) dan normal (1/5).

b. Proses inferensi menggunakan model sugeno

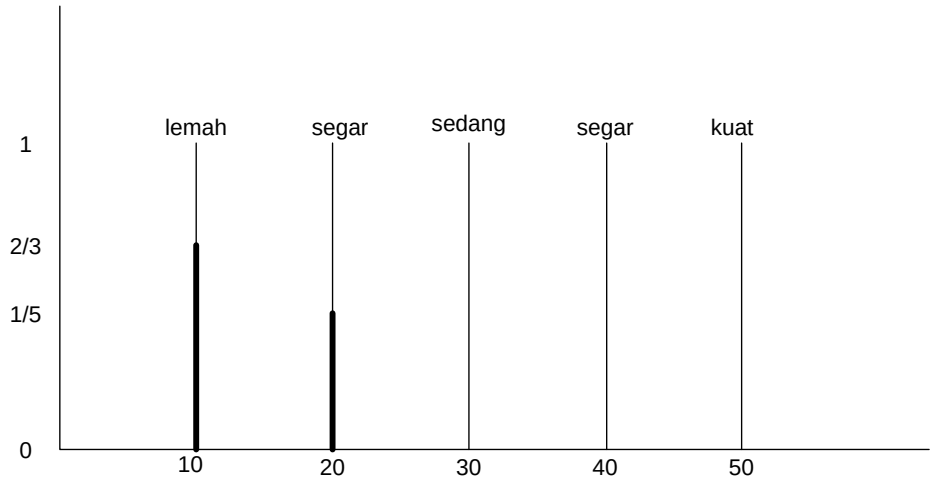
Fungsi keanggotaan singleton yaitu fungsi keanggotaan yang memilik derajat keanggotaan 1 pada suatu nilai crips tunggal dan 0 pada semua nilai crips yang lain. Misal didefinisikan fungsi singleton untuk kecepatan angin :



Gambar 3.11 contoh proses inferensi pada model sugeno

Dari empat data fuzzy input tersebut , hangat (2/3), panas (1/3), lemah (4/5), dan normal (1/5) , maka kita mendaptakan 4 aturan fuzzy (dari 15 aturan) yang diaplikasikan:

- Gunakan aturan *counjuction* (^) untuk memilih derajat keangotaan minimum
  - If *suhu* = hangat (2/3) AND *tekanan* = Rendah (4/5) THEN angin = (2/3) angin lemah.
  - If *suhu* = hangat (2/3) AND *tekanan* = sedang (1/5) THEN angin = (1/5) angin segar.
  - If *suhu* = panas (1/3) AND *tekanan* = Rendah (4/5) THEN angin = (1/3) angin lemah.
  - If *suhu* = panas (1/3) AND *tekanan* = sedang (1/5) THEN angin = (1/5) angin segar .
- Gunakan aturan disjunction (v) dangan memilih derajat keanggotaan maximum dari nilai linguistic yang dihubungkan oleh v ; dari angin lemah (2/3) v angin lemah (1/3) angin sedang (1/3) dihaslikan angin segar (2/3). Sedangkan angin segar (1/5) v angin segar (1/5) dihasilkan angin segar (1/5). Dengan demikian kita memperoleh dua pernyataan : angin lemah (2/3) dan angin segar (1/5)



Gambar 3.12 dua derajat keangotaan yang dihasilkan oleh model sugeno. Angin lemah (2/3) dan angin segar (1/5)

c. Proses *defuzzyfication* menggunakan model sugeno

   Proses defuzzyfication menggunakan proses *hight method* maka dari dua fuzzy set , segar (1/5) dan lemah (2/3) maka dipilih nilai maximumnya yaitu angin lemah (2/3) karena nilai crips sebesar 10 maka kecepatan angin adalah 10 dan disebut angin lemah. kesimpulan dalam sistem adalah sistem mengelurkan kekuatan angin sebesar 10 yaitu angin lemah.

d. Kesimpulan jadi Misalkan, nilai crips yang diterima oleh system suhu adalah $37^{O}C$ dan nilai crips yang diterima system tekanan udara adalah 12% , apakah angin yang di dapatkan adalah “angin lemah” yaitu kekuatannya 10, Nilai 10 ini digunakan sebagai animasi angin.

3.2 Analisis Kebutuhan Non-Fungsional

Analisis non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

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

2.2.1 Analisis Kebutuhan Perangkat Lunak

Game edukasi kebudayaan sunda berbasis android mobile memiliki kebutuhan perangkat lunak sebagai berikut:

1. Sistem operasi Android
2. Minimal processor 1Ghz

Dari hasil pengamatan perangkat lunak, terdapat beberapa hal penting yang harus diperhatikan agar sistem yang akan dirancang dapat berjalan dengan baik dan sesuai dengan yang diharapkan. Pengguna diharapkan dapat memenuhi kebutuhan perangkat lunak yang telah direkomendasikan agar aplikasi dapat berjalan dengan baik.

2.2.2 Analisis Kebutuhan Perangkat Keras

Agar aplikasi dapat berjalan dengan baik, maka dibutuhkan perangkat keras yang sesuai dengan kebutuhan aplikasi. Spesifikasi minimum perangkat keras yang dibutuhkan agar dapat menjalankan *game* Edukasi Kebudayaan Sunda ini adalah sebagai berikut:

1. Handphone dengan software android
2. Memori procesor minimal 1Ghz
3. Kapasitas memory external minimal 256mb

2.2.3 Analisis Pengguna

Setelah melakukan analisis target user dalam menggunakan *game* sibujarat adalah sebagai berikut :

Table 3.3 Analisi pengguna

Karakteristik Fisik Pengguna	
Umur	6 – 12 tahun
Gender	Laki-laki dan perempuan
Handedness	Cenderung menggunakan tangan kanan
Disabilities	tidak buta dan tidak mempunyai cacat fisik
Pengetahuan dan Pengalaman Pengguna	
Computer Literacy	Memiliki pengalaman dalam mengoperasikan sebuah *mobile phone*
System Experience	Memiliki pengetahuan berinteraksi dengan input *mobile phone*
Application Experience	Pernah memainkan *game* sejenis sebelumnya
Education	Sekolah Dasar
skills	Sudah bisa Membaca
Task Experience	Mengetahui cara menggunakan *mobile phone*
Native Language or Culture	Bahasa indonesia
Karakteristik Psikologi Pengguna	
motivation	Mempunyai motivasi yg cukup dalam menyelesaikan permainan
patience	Mempunyai kesabaran dalam menyelesaikan permainan
Stress level	Tidak gampang strees
Attitude	*positive*

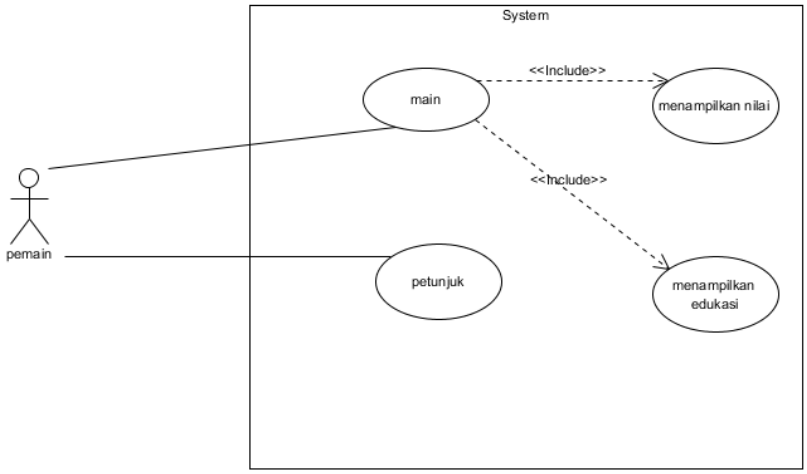
3.3 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik sesuai kebutuhan.

1. *Use Case Diagram*

Use Case atau diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use Case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat [5].

Berikut ini adalah perancangan proses-proses yang terdapat pada *game* edukasi kebuayaan sunda, yang digambarkan dengan *Use Case Diagram* yang dapat dilihat pada Gambar 3.11.



Gambar 3.13 *Use Case Diagram game* Edukasi Kebudayaan Sunda.

Use Case terdiri dari tiga bagian yaitu definisi *actor*, definisi *Use Case* dan skenario *Use Case*.

1.1. *Definisi Actor*

Definisi *Actor* berfungsi untuk menjelaskan *Actor* yang terdapat pada *Use Case Diagram*. Definisi *Actor* diterangkan pada Tabel 3.3.

Tabel 3.4 Definisi *Actor*.

No	*Actor*	Deskripsi
1.	Pemain	Orang yang akan memainkan *game*

1.2. *Definisi Use Case*

Definisi *Use Case* berfungsi untuk menjelaskan fungsi *Use Case* yang terdapat pada *Use Case Diagram*. Definisi *Use Case* diterangkan pada Tabel 3.2.

Tabel 3.5 Definisi *Use Case.*

No	*Use Case*	Deskripsi
1.	Main	Proses untuk memulai permainan
2.	Petunjuk	Proses untuk menampilkan petunjuk panduan bermain
3.	Menampilkan edukasi	Proses untuk menampilkan informasi eduaksi berupa informasi-informasi kebudayaan
4.	Menampilkan Nilai	Proses setelah menyelesaikan *game* mendapatkan nilai

1.1. *Skenario Use Case*

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

1.1.1. *Skenario Use Case Memulai Permainan*

Tabel 3.6 Skenario *Use Case* Memulai Permainan.

Identifikasi	
Nomor	1
Nama	Main
Tujuan	Main
Deskripsi	Proses untuk memulai permainan
Aktor	Pemain
Skenario Utama	

Kondisi awal	Pemain berada di menu utama
Aksi Aktor	Reaksi Sistem
Memilih menu mulai	
	Menampil proses permainan
	Loading screen sebelum masuk kedalam permainan
	Membuka load level 1
	Memulai Permainan level 1
Pemain menggerakan karakter, mengarahkan sasaran dan mengatur tenaga dengan menarik ketapel	
	Menghasilkan tenaga yang diberikan ke benda untuk mengenai sebuah objek dan akan mengurangi jumlah kesempatan batu yang diberikan
	Mengacak angin yang akan muncul
	Merubah tenaga batu menjadi tidak stabil kecenderungan menjadi sulit.
Lemparan Mengani sasaran	
	Mendapatkan informasi angklung , tari jaipong, wayang golek.
Lanjut informasi	
	Setelah mendapatkan informasi kembali kepermainan dan melanjutkan permainan
Tidak mengenai sasaran	
	Game looping sampe habis kesempatan
	Diberikan 6 batu untuk kesempatan

	melempar
Game over	
	Setelah semua lemparan habis maka akan terjadi *game over* , permainan kembali ke menu utama permainan dilanjutkan pada di level 1.
Lanjut level 2	
	Proses Membuka Level 2
	Loading screen sebelum masuk kedalam permainan
	Membuka load level 2
	Memulai Permainan Level 2
Pemain menggerakan karakter, mengarahkan sasaran dan mengatur tenaga dengan menarik ketapel	
	Menghasilkan tenaga yang diberikan ke benda untuk mengenai sebuah objek dan akan mengurangi jumlah kesempatan batu yang diberikan
	Mengacak angin yang akan muncul
	Merubah tenaga batu menjadi tidak stabil kecenderungan menjadi sulit.
Lemparan Mengani sasaran level 2	
	Mendapatkan informasi Degung , kecapi suling, pencak silat.
Lanjut informasi level 2	
	Setelah mendapatkan informasi kembali kepermainan dan

	melanjutkan permainan
Tidak mengenai sasaran	
	Game looping sampe habis kesempatan
	Diberikan 5 batu untuk kesempatan melempar
Game over	
	Setelah semua lemparan habis maka akan terjadi *game over* , permainan kembali ke menu utama permainan dilanjutkan pada di level 1.
Lanjut level 3	
	Proses Membuka Level 3
	Loading screen sebelum masuk kedalam permainan
	Membuka load level 3
	Memulai Permainan Level 3
Pemain menggerakan karakter, mengarahkan sasaran dan mengatur tenaga dengan menarik ketapel	
	Menghasilkan tenaga yang diberikan ke benda untuk mengenai sebuah objek dan akan mengurangi jumlah kesempatan batu yang diberikan
	Mengacak angin yang akan muncul
	Merubah tenaga batu menjadi tidak stabil kecenderungan menjadi sulit.
Lemparan Mengani sasaran level 3	

	Mendapatkan informasi sisingaan , kuda lumping, reog.
Lanjut Informasi level 3	
	Setelah mendapatkan informasi kembali kepermainan dan melanjutkan permainan
Tidak mengenai sasaran	
	Game looping sampe habis kesempatan
	Diberikan 4 batu untuk kesempatan melempar
Game over	
	Setelah semua lemparan habis maka akan terjadi *game over* , permainan kembali ke menu utama permainan dilanjutkan pada di level 1.
Kembali	
	Kembali ke Menu awal permainan
Kondisi akhir	Aplikasi *game* menampilkan untuk memulai permainan dan melanjutkan ke level selanjutnya

1.1.2. *Skenario Use Case Menampilkan Petunjuk*

Tabel 3.7 Skenario *Use Case* Menampilkan Petunjuk.

Identifikasi	
Nomor	2
Nama	Petunjuk
Tujuan	Menampilkan petunjuk panduan bermain
Deskripsi	Proses untuk menampilkan petunjuk panduan bermain

Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di menu utama	
Aksi Aktor		Reaksi Sistem
Memilih menu petunjuk		
		Menampilkan menu yang ada di menu petunjuk (cara bermain, nilai)
Memilih cara bermain		
		Menampilkan informasi mengenai cara memulai permainan, cara bermain *game*.
Kondisi akhir	Aplikasi *game* menampilkan petunjuk panduan bermain	

1.1.3. *Skenario Use Case Menampilkan Edukasi*

Tabel 3.8 Skenario *Use Case* Menampilkan Edukasi.

Identifikasi		
Nomor	3	
Nama	Menampilkan Edukasi	
Tujuan	Menampilkan Edukasi	
Deskripsi	Proses meliputi informasi isi materi	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain melakukan permainan	
Aksi Aktor		Reaksi Sistem
Pemain menggerakan atau membidik sasaran pemain dengan mengarahkan lalu		

mengenai objek tertentu		
		Menampilkan Edukasi
Kondisi akhir	Apliaksi *game* menampilkan informasi Edukasi kebudayaan	

1.1.4. *Skenario Use Case Nilai*

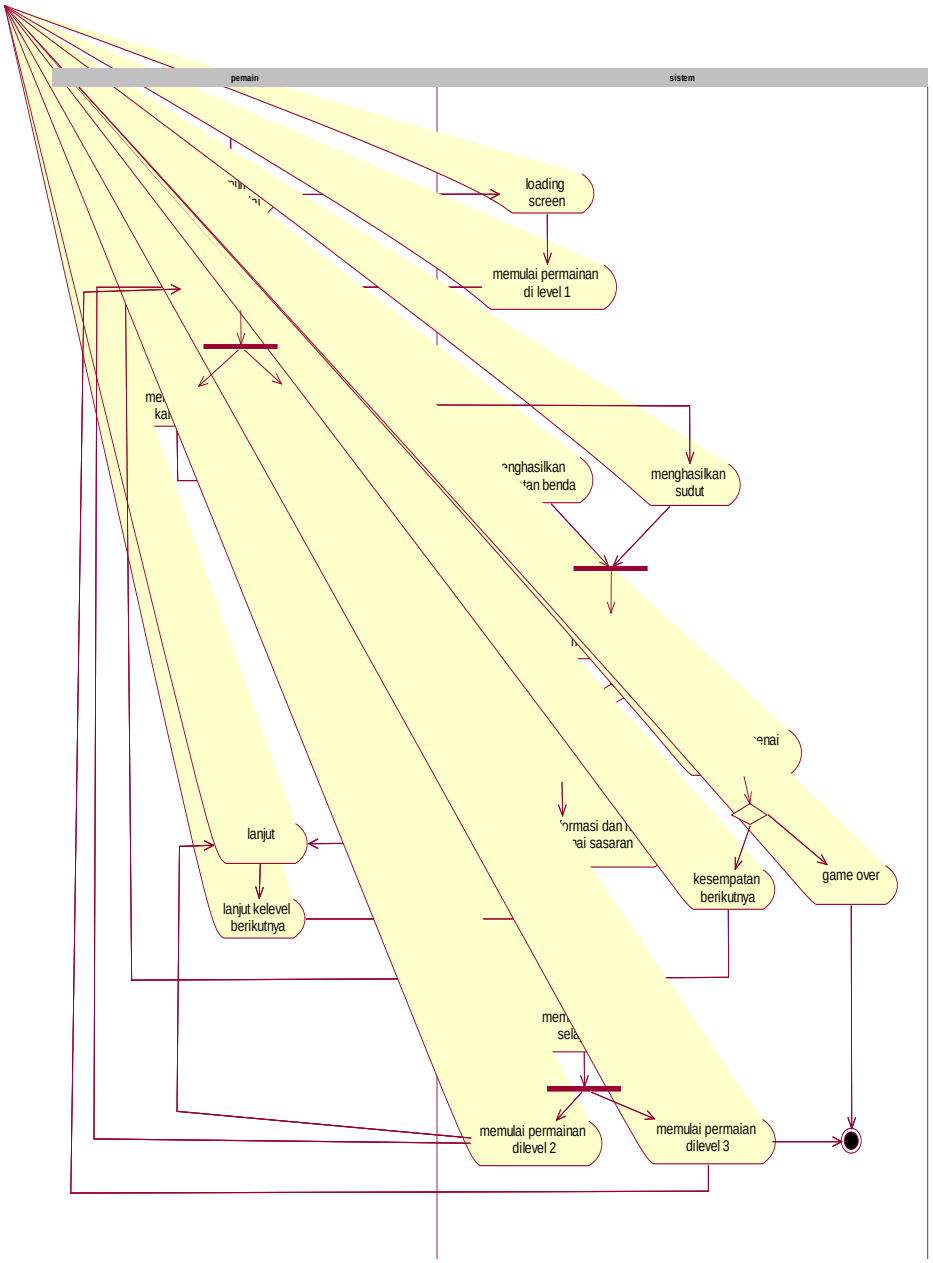
Tabel 3.9 Skenario *Use Case* Nilai.

Identifikasi		
Nomor	4	
Nama	Menampilkan Nilai	
Tujuan	Mendapatkan Score	
Deskripsi	Proses yang didapat ketika menyelesaikan *game*	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain selesai *game*	
Aksi Aktor		Reaksi Sistem
Selesai permainan		
		Menampilkan nilai yang didapatkan setelah menyelesaikan permainan
Kondisi akhir	Aplikasi *game* menampilkan nilai	

2. *Activity Diagram*

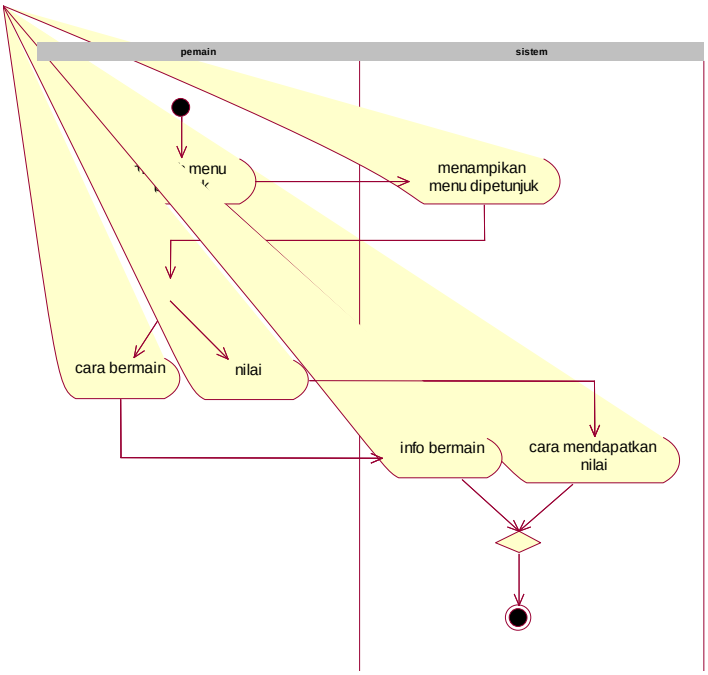
Diagram aktivitas atau activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis [5].

4.1. *Activity Diagram* Main



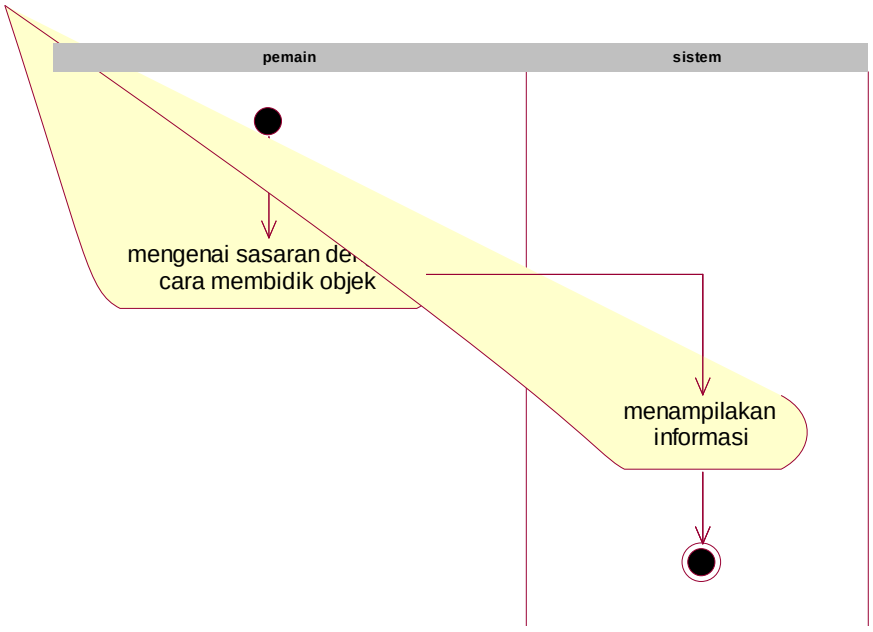
Gambar 3.14 *Activity diagram* Main

4.2 Activity Diagram Petunjuk



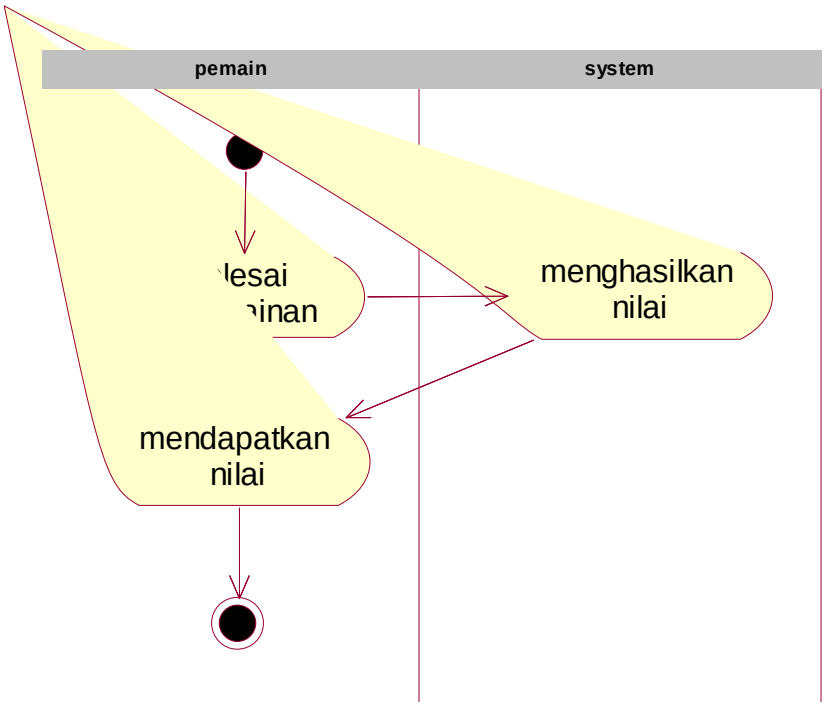
Gambar 3.15 *Activity Diagram* Petunjuk

4.3 Activity Diagram Informasi



Gambar 3.16 *Activity Diagram* Informasi

4.4 Activity Diagram Menampilkan Nilai

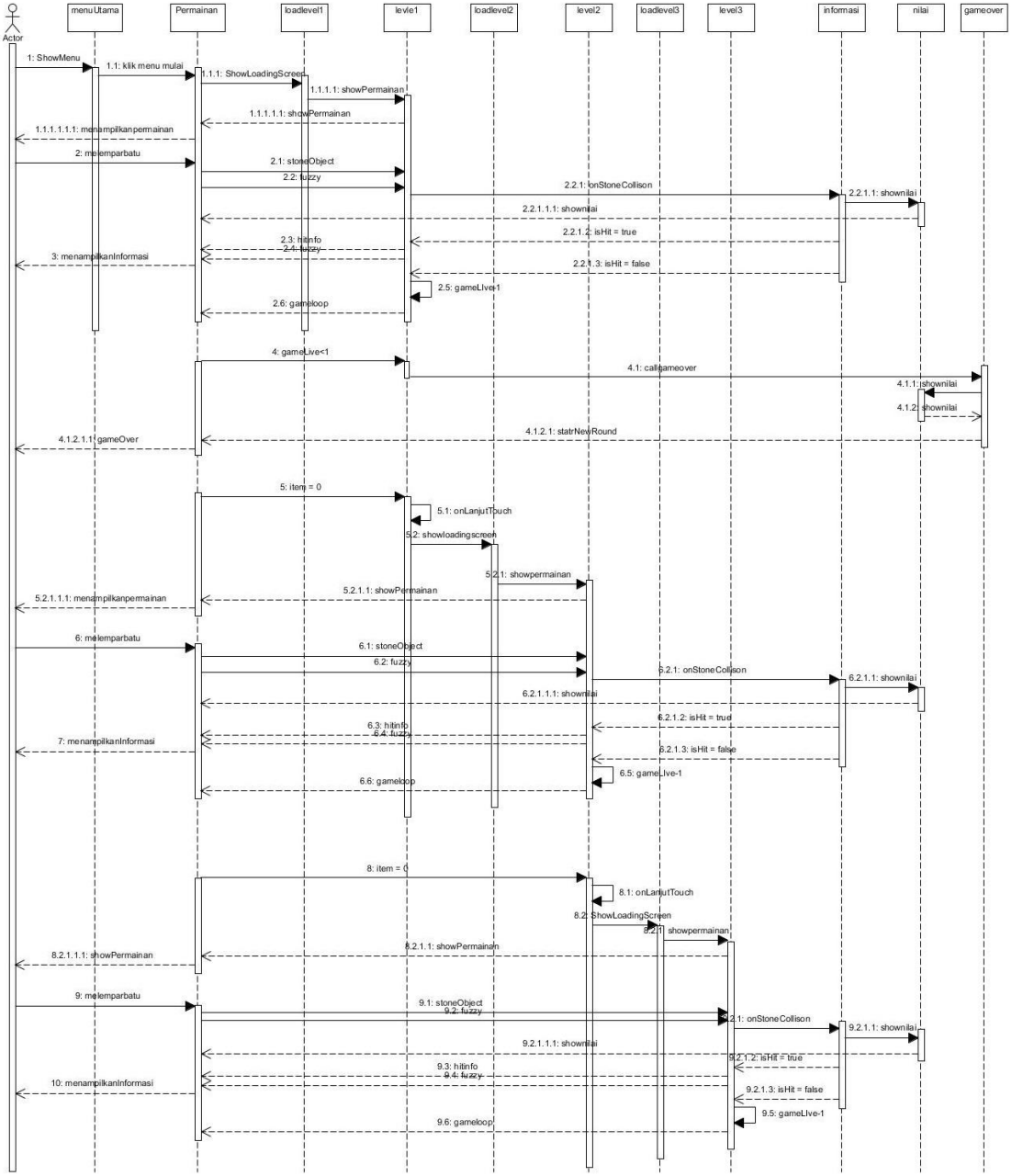


Gambar 3.17 *Activity Diagram* Nilai

3. *Sequence Diagram*

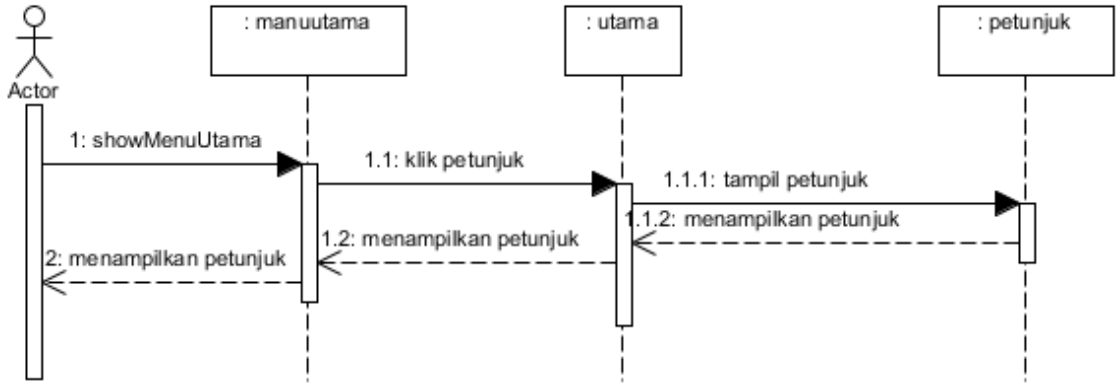
Diagram sequence atau *sequence diagram* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek [5].

3.1 *Sequence Diagram* Main



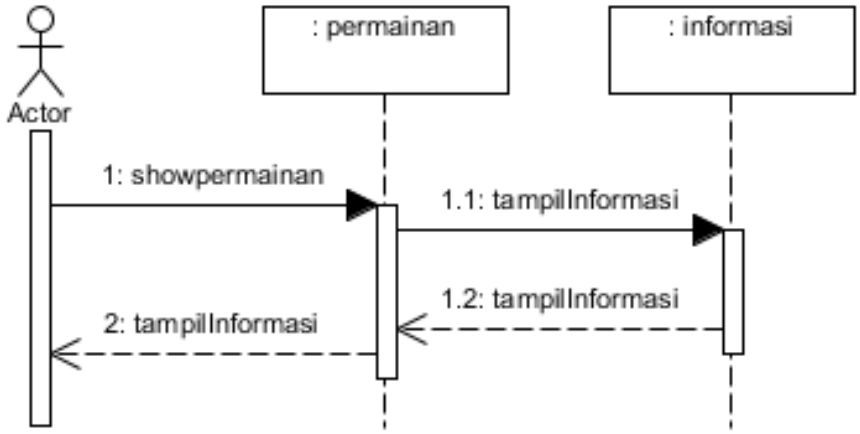
Gambar 3.18 *Sequence Diagram Main*

3.2 *Sequence Diagram* Petunjuk



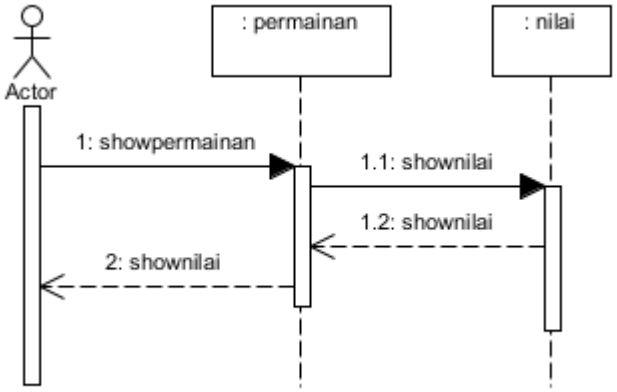
Gambar 3.19 *Sequence Diagram* Petunjuk

3.3 *Sequence Diagram* Menampilkan Edukasi



Gambar 3.20 *Sequence Diagram* Menampilkan Edukasi.

3.4 *Sequence Diagram* Menampilkan Nilai

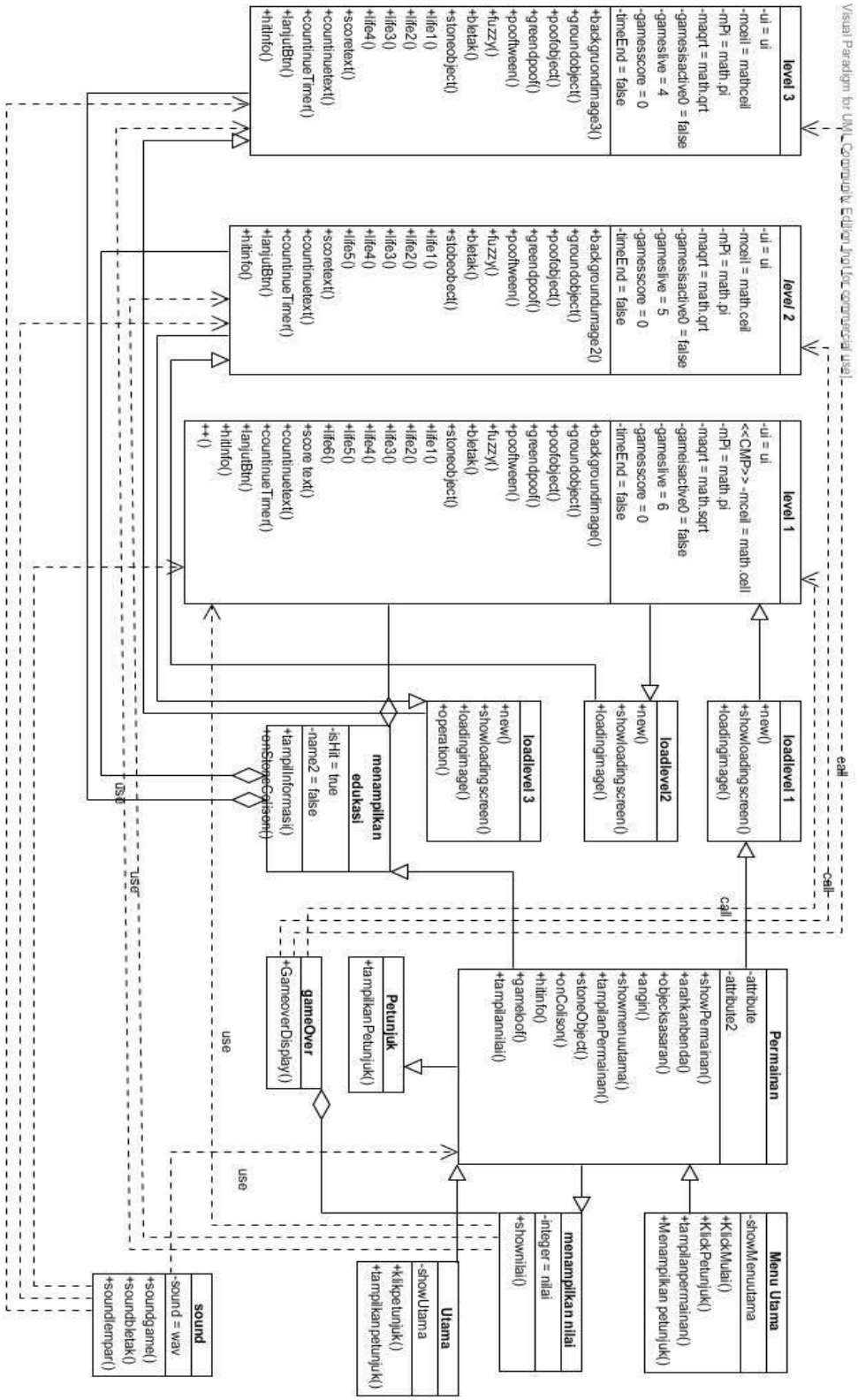


Gambar 3.21 *Sequence Diagram* Menampilkan Nilai.

1. ***Class Diagram***

Diagram kelas atau atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem [5].

Berikut ini adalah perancangan struktur sistem yang terdapat pada *game* edukasi sejarah kemerdekaan Indonesia, yang digambarkan dengan class diagram yang dapat dilihat pada Gambar 3.6.



Gambar 3.22 *Class Diagram Game* Edukasi kebudayaan sunda

3.4 Perancangan Sistem

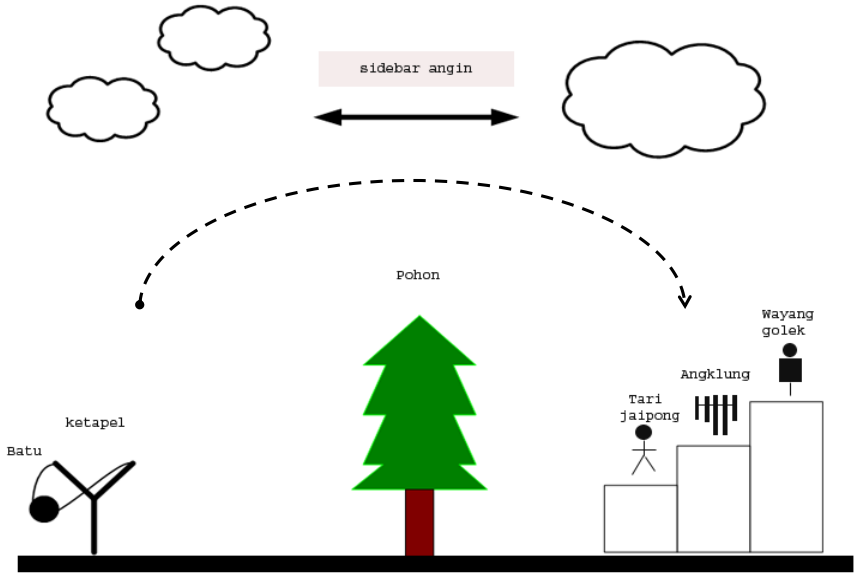
Perancangan adalah suatu bagian dari metodologi pengembangan suatu perangkat lunak yang dilakukan setelah tahapan untuk memberikan gambaran secara terperinci.

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 dan berfungsi, juga menyangkut konfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem.

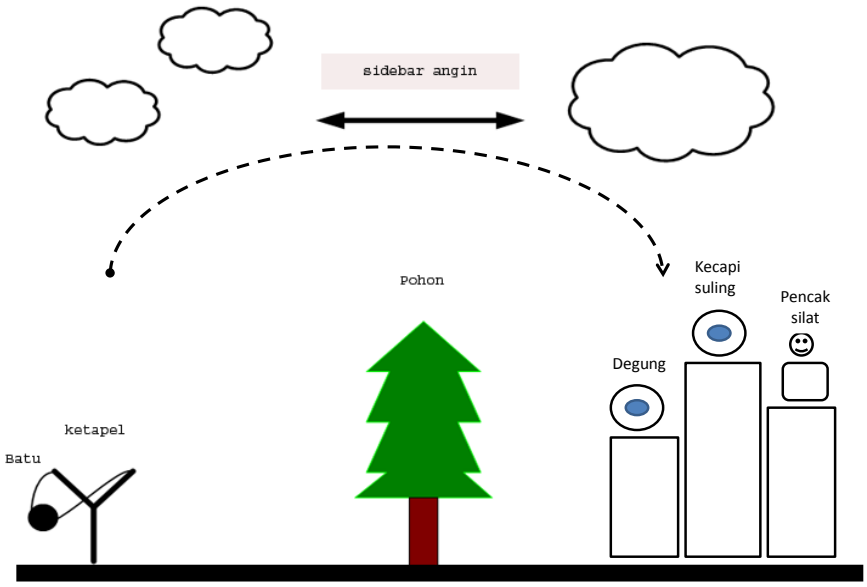
3.4.1 Storyboard

Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan naskah, dengan storyboard kita dapat menyampaikan ide cerita kita kepada orang lain dengan lebih mudah, karena kita dapat menggiring khayalan seseorang mengikuti gambar-gambar yang tersaji, sehingga menghasilkan persepsi yang sama pada ide cerita kita.

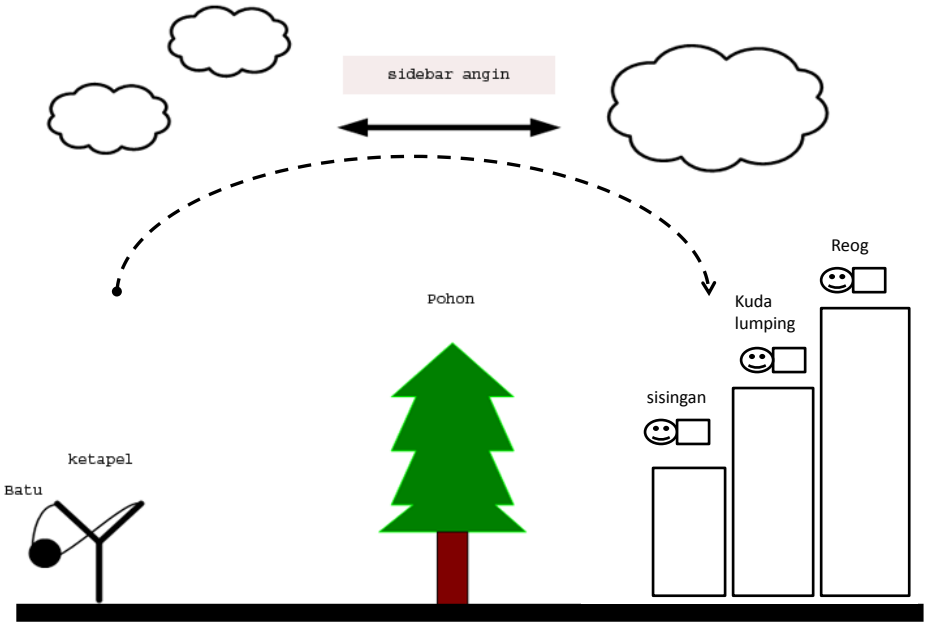
Berikut merupakan storyboard dari *game* edukasi kebudayaan sunda:



Gambar 3.23 *Storyboard* level 1



Gambar 3.24 *Storyboard* level 2.



Gambar 3.25 *Storyboard* level 3.

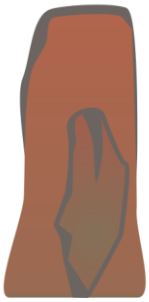
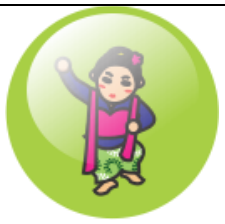
3.4.2 Pengenalan Properti

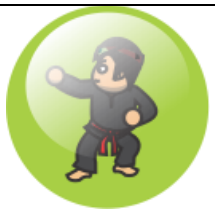
Pengenalan karakter merupakan pembahasan mengenai karakter-karakter dan objek yang ada dan terlibat dalam *game* edukasi kebudayaan sunda.

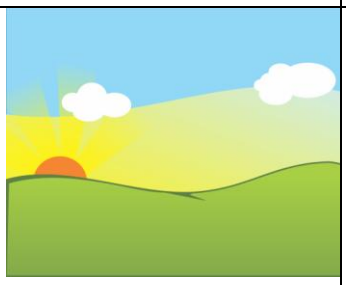
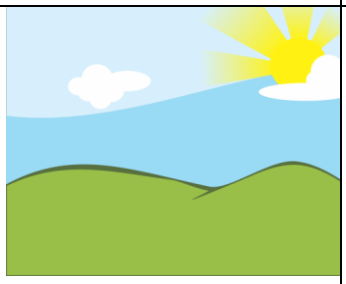
Pengenalan karakter dapat dilihat pada Tabel 3.12.

Tabel 3.10 Pengenalan Karater

No.	Nama	Gambar	Keterangan
1	Ketapel		- Alat yang digunakan user untuk memulai permainan. - Ketapel ini bisa diarahkan dan diatur tenaga yang diinginkan
2	Batu		- Alat yang digunakan untuk mengenai objek berupa batu - Batu ini akan pecah bila

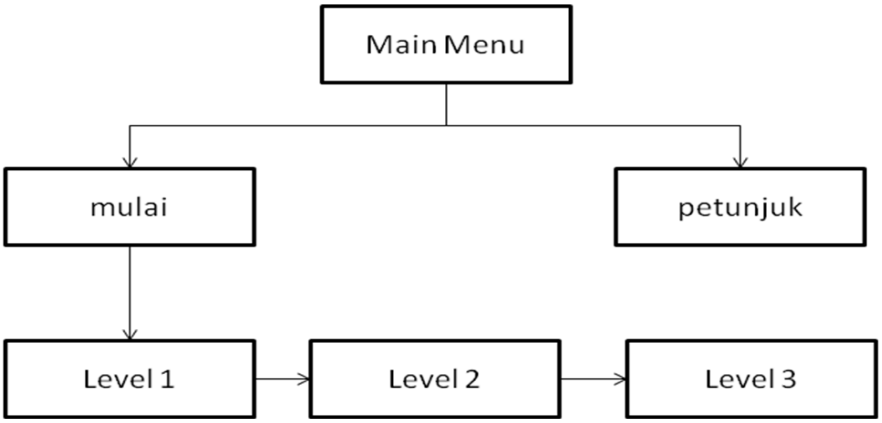
			mengenai sasaran
3	Pohon		- Pohon sebagai penghalang atau rintangan didalam *game* ini
4	Gunung		- Tempat objek sasaran berada - Gunung ini di setiap level berbeda keadaanya
5	Angklung		- Angklung ini berupa icon didalam *game* ini - Jika mengenai sasaran akan mendapat kan informasi berupa angklung tersebut berupa gambar.
6	Tari jaipong		- Tari jaipongan ini berupa icon didalam *game* ini - Jika mengenai sasaran akan mendapat kan informasi berupa angklung tersebut berupa gambar.
7	Wayang golek		- Wayang golek ini berupa icon didalam *game* ini - Jika mengenai sasaran akan mendapat kan informasi berupa angklung tersebut berupa gambar.

8	Degung		- Degung ini berupa icon didalam *game* ini - Jika mengenai sasaran akan mendapat kan informasi berupa angklung tersebut berupa gambar.
9	Kecapi suling		- Kecapi suling ini berupa icon didalam *game* ini - Jika mengenai sasaran akan mendapat kan informasi berupa angklung tersebut berupa gambar.
10	Pencak silat		- Pencak silat ini berupa icon didalam *game* ini - Jika mengenai sasaran akan mendapat kan informasi berupa angklung tersebut berupa gambar.
11	Sisingaan		- sisingan ini berupa icon didalam *game* ini - Jika mengenai sasaran akan mendapat kan informasi berupa angklung tersebut berupa gambar.
12	Kuda lumping		- Kuda lumping ini berupa icon didalam *game* ini - Jika mengenai sasaran akan mendapat kan informasi berupa angklung tersebut berupa gambar.

13	Reog		- Reog ini berupa icon didalam *game* ini - Jika mengenai sasaran akan mendapat kan informasi berupa angklung tersebut berupa gambar.
14	Keadan level1		- Pagi pada saat keadaan di level 1
15	Keadaan level2		- Siang pada saat keadaan di level 2
16	Keadaan level 3		- Malam pada saat keadaan di level 3
17	Angin		- Angin adalah rintangan atau hambatan dalam proses lemparan - Angin terjadi dalam keadan besar ke kiri dan kekanan

3.4.3 Perancangan Arsitektur Menu

Perancangan struktur menu merupakan jalur pemakaian sistem yang mudah dipahami dan mudah digunakan. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini, yaitu :



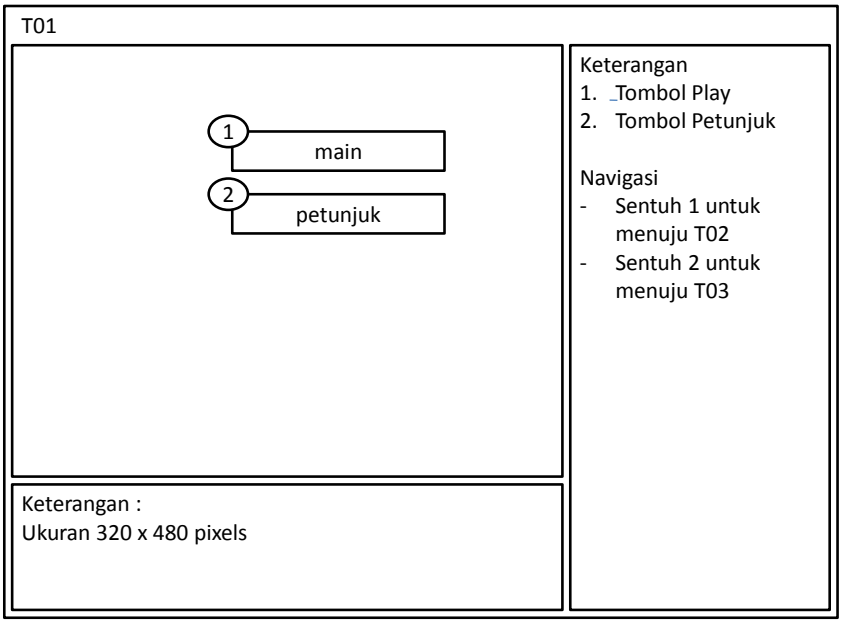
Gambar 3.26 Arsitektur Menu *sibujarat*

1. Menu Mulai : merupakan submenu untuk memulai permainan baru
2. Menu petunjuk : merupakan submenu yang menampilkan informasi petunjuk permainan
3. *Level* 1 : merupakan submenu *level* yang menampilkan dunia *gamelevel* 1
4. *Level* 2 : merupakan submenu *level* yang menampilkan dunia *gamelevel* 2
5. *Level* 3 : merupakan submenu *level* yang menampilkan dunia *gamelevel* 3

3.4.4 Perancangan Antar Muka

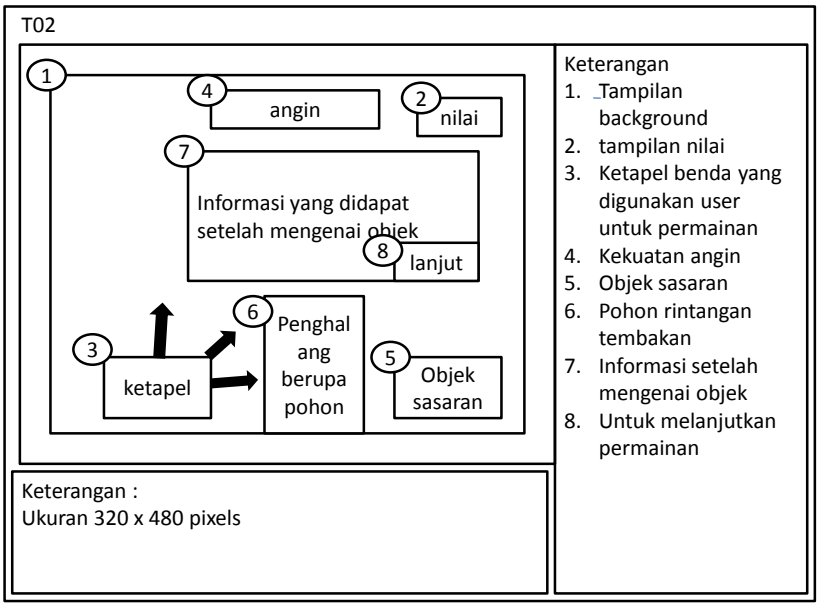
Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan mempermudah dalam mengimplementasikan aplikasi serta akan memudahkan pembuatan aplikasi.

1. Perancangan Antar Muka Menu Utama



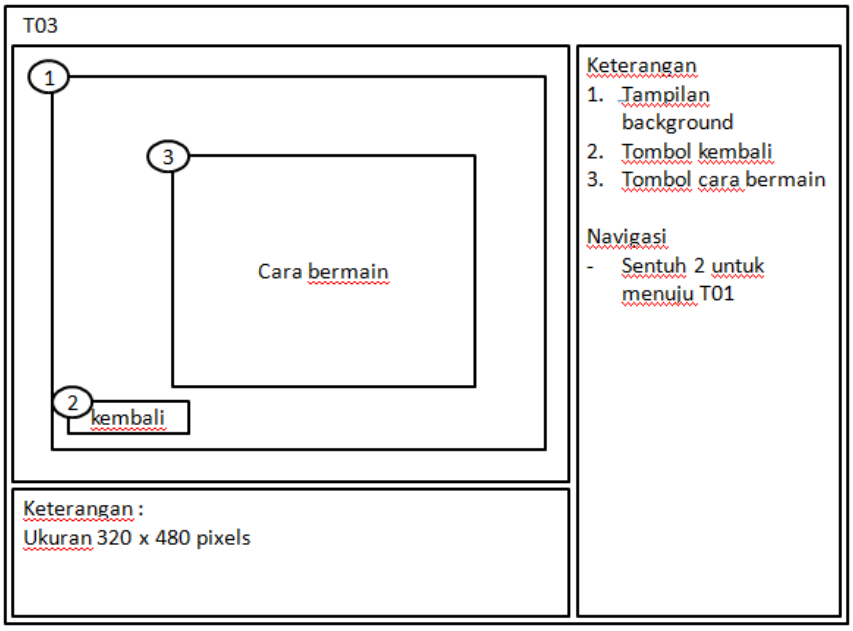
Gambar 3.27 Perancangan Antar Muka Menu Utama.

2. Perancangan Antar Menu Main



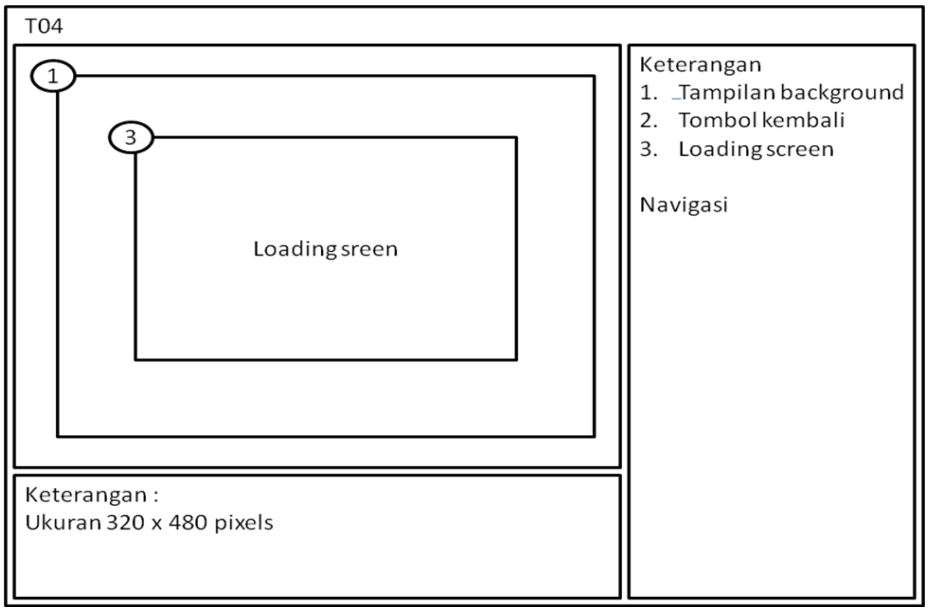
Gambar 3.28 Perancangan Antar Muka Menu Main

3. Perancangan Antar Muka Menu Petunjuk



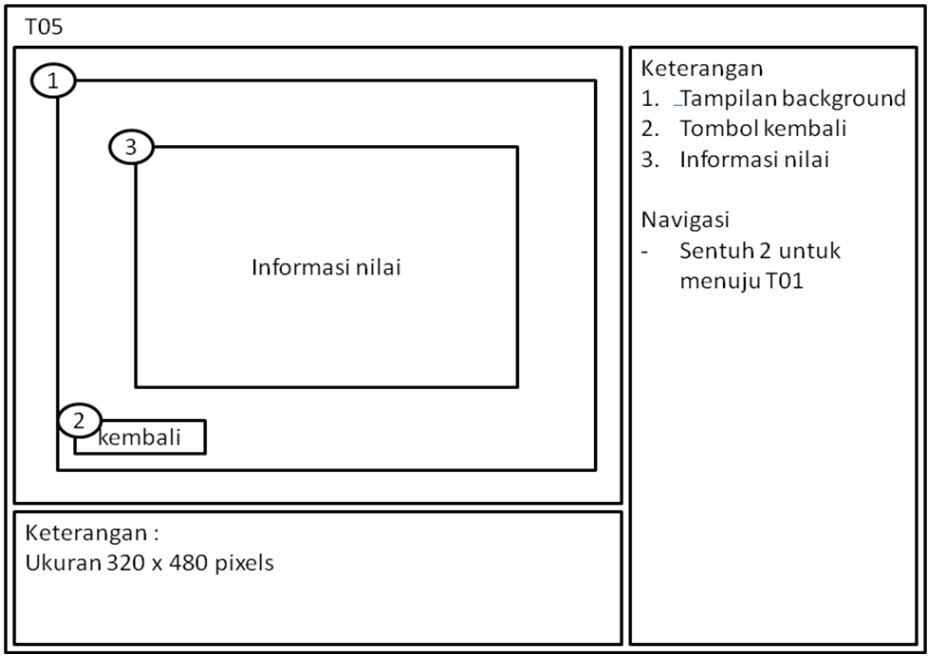
Gambar 3.29 Perancangan Antar Muka petunjuk

4. Perancangan Antar Muka *loading sreen*



Gambar 3.30 Perancangan Antar Muka *loading screen*

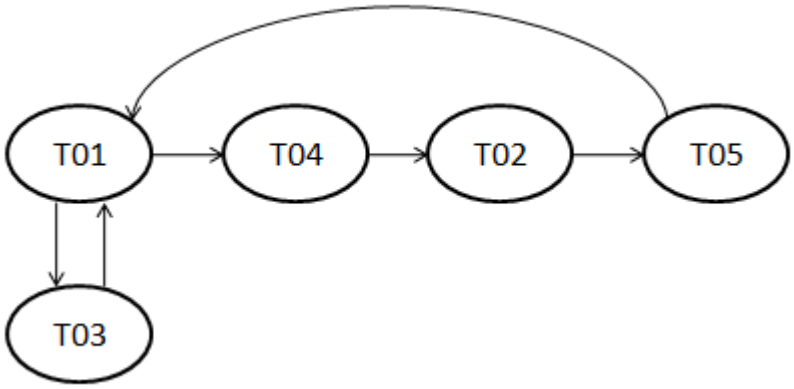
5. Perancangan Antar Muka Nilai



Gambar 3.31 Perancangan Antar Muka Nilai

3.4.5 **Jaringan Semantik**

Jaringan semantik adalah gambaran tentang relasi dari masing- masing halaman yang bisa diakses oleh seorang pengguna. Jaringan semantik ini, digunakan sebagai panduan penggunaan menu-menu dalam sebuah aplikasi. Terdapat 6 tampilan pada aplikasi ini, yaitu T01 untuk tampilan menu utama aplikasi, T02 untuk tampilan Main *level*, T03 untuk tampilan informasi cara bermain, T04 untuk tampilan loading screen, T05 untuk tampilan nilai. Perancangan semantik ini dapat dilihat pada gambar 3.33 berikut :



Gambar 3.32 Jaringan Semantik

3.4.6 **Perancangan*Method***

Perancangan *method* merupakan perncangan yang berfungsi untuk mendeskripsikan *method-method* yang berada di dalam aplikasi. *Method* dapat dipanggil dengan menyertakan variabel, baik hanya satu variabel, banyak variabel atau bahkan tidak adasama sekali.Adapun *method-method* yang terdapat dalam *game* sibujarat yang akan dibangun adalah sebagai berikut :

1. ***Method gameloop***

   Method ini agar permainan kembali setelah melempar objek, untuk jelasnya dapat dilihat dari *algoritma* pada gamabar berikut ini:

```
Algoritma_gameLoop
Deklarasi
        Stoneobject, ishit, dotTimer, newround, gameisactive   : Boolean
        X, timer                                               : integer

Deskripsi
        Gameisaktive  ⟵  false
        dotTimer ⟵  false
        if(gameisactive ⟵  true) then
                if(stoneobject.ishit ⟵false  &&  stoneobject.x >= 610 ) then
                stoneobject.ishit   ⟵ true
                timer.cancel(dotTimer)
        endif
                newround ⟵ false
        if(stoneobject.ishit    ⟵ false  &&  stoneobject.x >= 0) then
                stoneobject.ishit   ⟵ true
                timer.cancel(dotTimer ⟵  true)
        endif
                newround    ⟵ true
```

Gambar 3.33 *algoritma Method gameloop*

2. *Method collison*

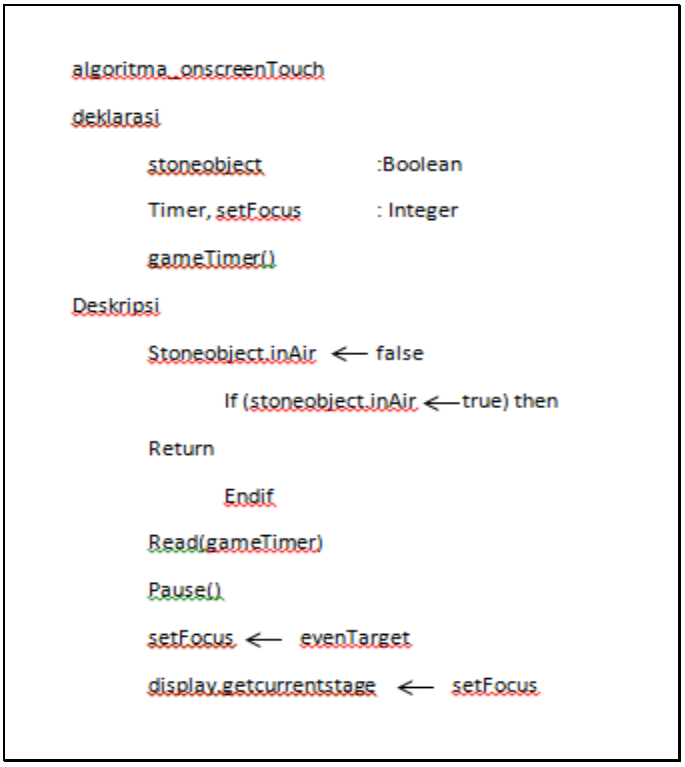
Method ini agar sistem mengecek tumbukan yang terjadi pada dunia *game* dapat dilihat pada gambar berikut ini:

```
algoritma_onstone_collison
deklarasi
        stoneobject, ishit, dotTimer      : boolean
        createStone()
        audio()
        NewScore, timer          : integer
Deskripsi
        Read(creatStoneObject)
        Read(audio)
        If (stoneobject  ⟵  "began") then
                Audio.play (impactsound)
                If (stoneobject.ishit  ⟵  false) then
                        dotTimer  ⟵  timer.cancel
                endif
        newScore ⟵ 500
        setScore (newScore)
        endif
```

Gambar 3.34 *algoritma Method collison*

3. *Method onscreenTouch*

Method ini agar sistem mengentikan permainan atau melanjutkan karena masih ada kesempatan dan membuat setiap memulai permainan harus disentuh dapat dilihat pada gambar berikut ini:



Gambar 3.35 *algoritma onscreenTouch*

4. *Method backgroundimage*

Method ini agar sistem menampilkan background berbeda tiap *level algoritma* pada gambar berikut ini:

```
algoritma_backgroundimage
deklarasi
        backgroundimage       : char
        imageRect()
        gameinsert            : char
deskripsi
        read (imagerect)
        backgroundimage  ←display.newimagerect
        backgroundimage.x  ← display.contentwidth/2
        backgroundimage.y  ← display.contenthight/2
        gameinsert(backgroundimage)
```

Gambar 3.36 *algoritma backgroundimage*

5. *Method ground*

Method ini agar sistem memberikan tanah sebagai pijakan berbeda tiap *level* dapat dilihat pada gambar berikut ini:

```
algoritma_creatground
deklarasi
        groundobject, gamegroup        : char
        imageRect(display)
        serReperencePoint              : integer
deskripsi
        read (imageRect)
        groundobject  ⟵ serReperencePoint
        goundobject.x  ⟵      -60
        goundobject.y  ⟵      350
        gameGroup(insert)
        gameGroup.name ⟵"ground"
```

Gambar 3.37 *algoritma groundobject*

6. *Method poofobject*

Method ini agar sistem memberikan asap setiap mengenai sasaran tanda benda terkena lemparan berbeda tiap *level* dapat dilihat pada gambar berikut ini:

```
algoritma_poofobject
deklarasi
        imageRect (display)
        gameEvent (alpha : integer , visible : Boolean)
        poofobject       :char
deskripsi
        read (imageRect)
        read (gameEvent)
        poofobject.alpha  ⟵ 1.0
        poofobject.isvisible  ⟵ true
```

Gambar 3.38 *algoritma poofobject*

7. *Method greenpoof*

Method ini agar sistem memberikan gambar tanda titik-titik sesudah melempar berbeda tiap *level* dapat dilihat pada gambar berikut ini:

```
algoritma_greenpoof
deklarasi :
        imageRect (display)
        gameEvent (alpha : integer , visible : Boolean)
        greenpoof       :char
deskripsi
        greenpoof.x ⟵ self.x
        greenpoof.y ⟵ self.y
        greenpoof   ⟵ display.newimageRect
        greenpoof.alpha   ⟵  1.0
        greenpoof.isvisible  ⟵  true
```

Gambar 3.39 *Algoritma greenpoof*

9 Method setScore

Method ini agar sistem menghitung berapa akurat score yang didapatkan pada permainan dapat dilihat pada gamabar berikut ini:

```
algoritma_setScore
deklarasi
        gameScore       : integer
        newScore        : integer
        scoreText       : char
        gameRect        ( x : int , y : int)
deskripsi
        read (gameRect)
        if (gameScore < 0) then
                gameScore ⟵ 0
        endif
        scoretext ⟵ gamescore
        scoretext.x ⟵ 30
        scoretext.y ⟵ 20
```

Gambar 3.40 *algoritma setScore*

10 Method btnplay

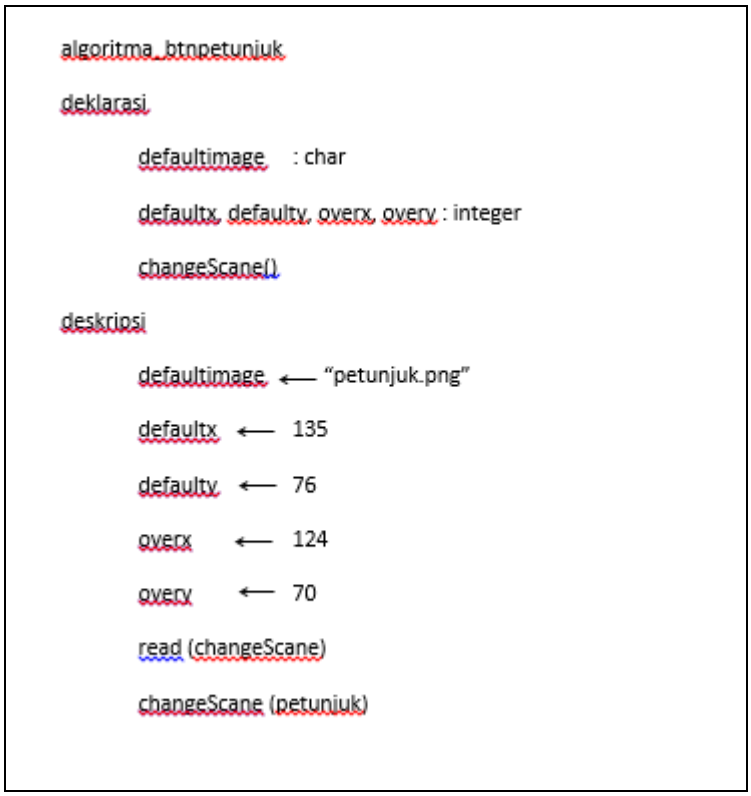
Method ini agar sistem membuat tombol play dan memasukan kedalam permainan pada gamabar berikut ini:

```
algoritma_btnplay
deklarasi
        defaultimage    : char
        defaultx, defaulty, overx, overy : integer
        changeScane()
deskripsi
        defaultimage ← "startbutton.png"
        defaultx ← 135
        defaulty ← 76
        overx ← 124
        overy ← 70
        read (changeScane)
        changeScane ("loadlevel1")
```

Gambar 3.41 *algoritma btnplay*

11. Method btnpetunjuk

Method ini agar sistem tombol petunjuk dan memasukan kedalam dunia petunjuk dapat dilihat pada gambar berikut ini:



Gambar 3.42 *algoritma btnpetunjuk*