BAB III
ANALISIS DAN KEBUTUHAN ALGORITMA

3.1 Analisis Masalah

Logika Fuzzy adalah suatu proses pengambilan keputusan berbasis aturan yang bertujuan untuk memecahkan masalah, dimana sistem tersebut sulit untuk dimodelkan atau terdapat ambiguitas dan ketidakjelasan yang berlimpah. Sistem logika fuzzy terdiri dari himpunan fuzzy dan aturan fuzzy. Logika fuzzy dalam game biasanya digunakan dalam pergerakan *NPC* atau musuh dalam sebuah permainan yang terdiri dari beberapa kondisi agar *NPC* itu sendiri dapat melakukan aksi atau tingkah laku yang akan mendukung permainan tersebut.

Game peranan jaksa penuntut memiliki latar belakang proses persidangan. Dalam pembangunan game ditemukan masalah bagaiamana hakim bisa mengambil keputusan dalam menentkan hukuman kepada terdakwa. Sehingga untuk menjadikan game ini menjadi suatu rangkaian dalam sebuah persidangan perlu dibutuhkan suatu algoritma yang dapat di implemantasikan pada *NPC* hakim.

3.2 Analisis *Game*

Pada game persdiangan ini pemain berperan sebagai seoarang jaksa penuntut umum. Hakim sangat berperan dalam mewujudkan peradilan. Hakim atau pengadilan pada prinsipnya adalah pasif, artinya hakim tidak mencari-cari perkara, tapi melayani pihak yang mengadu karena merasa haknya diragukan oleh pihak lain. Sehingga penerapan algoritma Fuzzy Logic bisa diterapkan di *NPC* hakim agar mampu berfikir dan bertindak selayaknya hakim dan yang sesungguhnya.

Gameplay

Berikut adalah *gameplay* pada *game* yang akan dibangun:

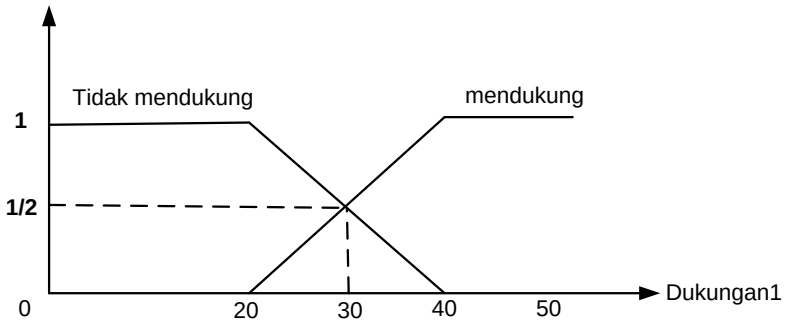
Dalam memulai permainan game ini pengguna akan dihadapkan pada kasus yang telah di sediakan oleh permainan. Setelah memilih kasus, pemain harus memilih

pasal-pasal yang sesuai dengan kasus yang diambilnya. Pasal-pasal yang terpilih ini nantinya akan digunakan oleh pemain untuk menuntut terdakwa dalam proses persidangan berlangsung. Pada tahap persidangan akan dilakukan proses mendengarkan pernyataan baik dari saksi pihak jaksa maupun saksi dari pihak terdakwa. Dalam proses mendengarkan kesaksian ini pemain dapat menganalisa poin – poin yang di utarakan oleh saksi dari pihak terdakwa. Pemain harus memilih poin – poin yang sesuai dengan kasus terdakwa.. Apabila pemain tidak dapat menangkap pernyataan-pernyataan yang salah tersebut maka pemain gagal dalam memberikan hukuman yang berat kepada terpidana. Pada akhir proses persidangan hakim akan memberikan keputusan hukuman kepada terpidana. Keputusan ini dipengaruhi oleh kemampuan pemain dalam menentukan pasal-pasal yang sesuai yang dapat memberatkan terpidana, dan keputusan dalam menangkap pernyataan-pernyataan yang tidak sesuai dengan fakta yang diucapkan oleh saksi.

3.3 Analisis Masukan

Terdapat 4 variabel pada *game* ini, dimana variabel ini akan menentukan hasil *output* pada *game* . Adapun variabel yang digunakan adalah:

1. Kesesuaian saksi dari jaksa penuntut umum (Dukungan1) dilakukan oleh pemain. Dengan range nilai variabel dukungan1 diantara 0 – 50. Nilai ini didasarkan pada tantangan dalam menilai kesaksian saksi yang berjumalah 5 saksi. Jika benar dalam menilai kesakisan, maka akan mendapatkan nilai 10 sehngga untuk mencapai nilai optimal maka pemain harus berhasil menilai semua kesaksian saksi dengan benar. Jika pemain salah semua, maka nilai Dukungan1 adalah 0. Dengan penjelasan sebagai berikut:
   a) Tidak Mendukung = 0 – 40 → artinya kesaksian yang diberikan tidak menguatkan bahwa terdakwa bersalah.
   b) Mendukung = 20 -50 → artinya kesaksian yang diberikan menguatkan bahwa terdakwa bersalah.

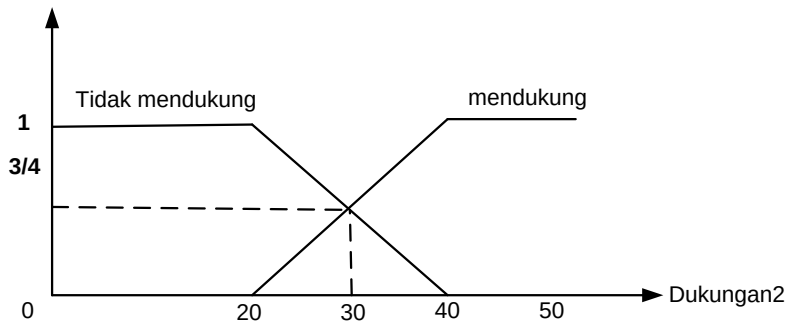


Gambar 3.1 Fungsi keanggotaan trapesium untuk Dukungan1

Fungsi Keanggotaan pada grafik derajat keanggotaan Dukungan1

a. Dukungan1 dengan nilai 20 termasuk dalam himpunan Tidak Mendukung dengan μTidak Mendukung[20] =1.
b. Dukungan1 dengan nilai 30 termasuk dalam himpunan Tidak Mendukung dan mendukung dengan μTidak Mendukung[30] dan μMendukung[30] dihitung menggunakan rumus persamaan (2.4).
c. Dukungan1 dengan nilai 40 termasuk dalam himpunan Mendukung dengan μMendukung[40] =1.
d. Dukungan1 dengan nilai 50 termasuk dalam himpunan Mendukung dengan μMendukung[50] =1.

2. Dukungan bukti saksi dari pihak terdakwa (Dukungan2) yang berperan sebagai musuh. Dengan range nilai variabel Dukungan2 diantara 0 – 50. Nilai ini didasarkan pada tantangan dalam proses interupsi, yaitu menilai kesaksian saksi yang berjumalah 5 saksi dari pihak terdakwa. Jika benar dalam menilai kesakisan, maka akan mendapatkan nilai 10 sehngga untuk mencapai nilai optimal maka pemain harus berhasil menilai semua kesaksian saksi dengan benar. Jika pemain salah semua, maka nilai Dukungan1 adalah 0. Dengan penjelasan sebagai berikut:
   a) Tidak Mendukung = 0 - 20 → artinya kesaksian yang diberikan menguatkan bahwa terdakwa tidak bersalah.
   b) Mendukung = 20 – 50 → artinya kesaksian yang menguatkan bahwa terdakwa bersalah.



Gambar 3.2 Fungsi keanggotaan trapesium untuk Dukungan2

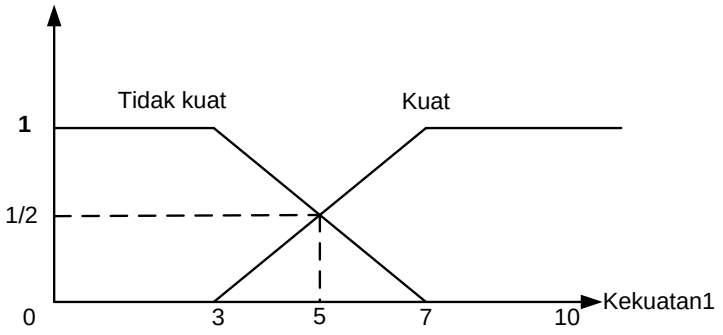
Fungsi Keanggotaan pada grafik derajat keanggotaan Dukungan2

a. Dukungan1 dengan nilai 20 termasuk dalam himpunan Tidak Mendukung dengan μTidak Mendukung[20] =1.
b. Dukungan1 dengan nilai 30 termasuk dalam himpunan Tidak Mendukung dan mendukung dengan μTidak Mendukung[30] dan μMendukung[30] dihitung menggunakan rumus persamaan (2.4).
c. Dukungan1 dengan nilai 40 termasuk dalam himpunan Mendukung dengan μMendukung[40] =1.
d. Dukungan1 dengan nilai 50 termasuk dalam himpunan Mendukung dengan μMendukung[50] =1.

3. Kekuatan pasal dari pemain untuk kasus yang di pilih (Kekuatan1) dilakukan oleh pemain. Dengan range nilai variabel kekuatan1 diantara 0 – 10. Nilai ini didasarkan pada tantangan dalam pemilihan pasal yang sesuai dengan kasus terdakwa . Dari 10 pasal yang disediakan tersebut, terdapat 2 pasal yang memiliki kekuatan yang sangat sesuai dengan kasus terdakwa diberikan nilai +3,5, dan 3 pasal yang hanya berhubungan dengan kasus narkoba saja diberikan nilai +1. Semenatar 5 pasal sisanya merupakan pasal yang tidak berhubungan sama sekali dengan kasus narkoba dan merupakan sebagai pengecoh untuk pemain diberikan nilai -1. Sehingga jika pemain dapat memilih pasal yang sesuai maka akan mendapatkan nilai optimal, tetapi jika pemain salah semua maka pemain mendapatkan nilai 0.

   Dengan penjelasan sebagai berikut:

a) Tidak Kuat = 0 – 7 → artinya pasal – pasal yang dipilih oleh jaksa PU tidak sesuai dengan kasus terdakwa.
b) Kuat = 3 - 10 → artinya pasal – pasal yang di pilih oleh jaksa PU sesuai dengan kasus terdakwa.



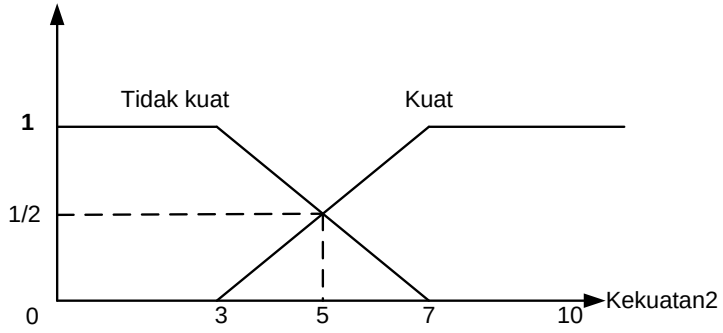
***Gambar* 3.3 Fungsi keanggotaan trapesium untuk Kekuatan1**

Fungsi Keanggotaan pada grafik derajat keanggotaan Kekuatan1

a. Kekuatan1 dengan nilai 3 termasuk dalam himpunan Tidak Kuat dengan μTidak Kuat[3] =1.
b. Kekuatan1 dengan nilai 5 termasuk dalam himpunan Tidak Kuat dan Kuat dengan μTidak Kuat[5] dan μKuat[5] dihitung menggunakan rumus persamaan (2.4).
c. Kekuatan1 dengan nilai 7 termasuk dalam himpunan Kuat dengan μKuat[7] =1.
d. Kekuatan1 dengan nilai 10 termasuk dalam himpunan Kuat dengan μKuat[10] =1.

4. Kekuatan pasal dari pengacara (Kekuatan2) berperan sebagai musuh. Kekuatan2 ini diperoleh dari pemilihan pasal yang sesuai dengan kasus terdakwa yang di pilih oleh pengacara terdakwa yang dilakukan secara random. Dengan range nilai variabel Kekuatan2 diantara 0 – 10 dengan penjelasan sebagai berikut:

a) Tidak Kuat = 0 - 7 → artinya pasal – pasal yang dipilih oleh pengacara tidak sesuai dengan kasus terdakwa.
b) Kuat = 3 – 10 → artinya pasal – pasal yang di pilih oleh pengacara terdakwa sesuai dengan kasus terdakwa.



***Gambar* 3.4 Fungsi keanggotaan trapesium untuk Kekuatan2**

Fungsi Keanggotaan pada grafik derajat keanggotaan Kekuatan2

a. Kekuatan1 dengan nilai 3 termasuk dalam himpunan Tidak Kuat dengan μTidak Kuat[3] =1.
b. Kekuatan1 dengan nilai 5 termasuk dalam himpunan Tidak Kuat dan Kuat dengan μTidak Kuat[5] dan μKuat[5] dihitung menggunakan rumus persamaan (2.4).
c. Kekuatan1 dengan nilai 7 termasuk dalam himpunan Kuat dengan μKuat[7] =1.
d. Kekuatan1 dengan nilai 10 termasuk dalam himpunan Kuat dengan μKuat[10] =1.

3.4 Analisis Algoritma Fuzzy Logic

Kecerdasan buatan yang akan diterapkan dalam pembangunan *game* ini adalah algoritma Fuzzy Logic. Algoritma ini nantinya dapat diterapkan pada *NPC* hakim dalam memberikan keputusan. Berikut adalah ilustrasi logika fuzzy yang diperoleh dari 4 variabel masukan dapat dilihat pada tabel Input dan Output Algoritma logika fuzzy.

Tabel 3.1 Input dan Output Algoritma Logika fuzzy

No	Saksi dari jaksa PU	Saksi dari terdakwa	Pasal yang di pilih pemain	Pasal yang di pilih pengacara	Hasil
1	Mendukung	Mendukung	Kuat	Kuat	Ditunda
2	Mendukung	Mendukung	Kuat	Tidak Kuat	Bersalah
3	Mendukung	Mendukung	Tidak Kuat	Kuat	Tidak bersalah
4	Mendukung	Mendukung	Tidak Kuat	Tidak Kuat	Ditunda
5	Mendukung	Tidak mendukung	Kuat	Kuat	Bersalah
6	Mendukung	Tidak mendukung	Kuat	Tidak Kuat	Bersalah
7	Mendukung	Tidak mendukung	Tidak Kuat	Kuat	Ditunda
8	Mendukung	Tidak mendukung	Tidak Kuat	Tidak Kuat	Bersalah
9	Tidak mendukung	Mendukung	Kuat	Kuat	Tidak bersalah
10	Tidak mendukung	Mendukung	Kuat	Tidak Kuat	Ditunda
11	Tidak mendukung	Mendukung	Tidak Kuat	Kuat	Tidak bersalah
12	Tidak mendukung	Mendukung	Tidak Kuat	Tidak Kuat	Tidak bersalah
13	Tidak mendukung	Tidak mendukung	Kuat	Kuat	Ditunda
14	Tidak mendukung	Tidak mendukung	Kuat	Tidak Kuat	Bersalah
15	Tidak mendukung	Tidak mendukung	Tidak Kuat	Kuat	Tidak Bersalah
16	Tidak mendukung	Tidak mendukung	Tidak Kuat	Tidak Kuat	Ditunda

Pada tabel input dan output digunakan untuk menentukan keputusan yang dilakukan oleh hakim. Berikut penjelasan dari masing-masing *input dan output* tersebut. Adapun himpunan fuzzy pada *game* ini adalah nya adalah:

1. Dukungan1
2. Dukungan2

3. Kekuatan1
4. Kekuatan2

Rule Logika fuzzy pada *game* peranan jaksa penuntut 4 x 2 = 16

1. **IF** Dukungan1 is mendukung **AND** Dukungan2 is mendukung **AND** Kekuatan1 is kuat **AND** Kekuatan2 is kuat then keputusan is ditunda.
2. **IF** Dukungan1 is mendukung **AND** Dukungan2 is mendukung **AND** Kekuatan1 is kuat **AND** Kekuatan2 is tidak kuat then keputusan is bersalah.
3. **IF** Dukungan1 is mendukung **AND** Dukungan2 is mendukung **AND** Kekuatan1 is tidak kuat **AND** Kekuatan2 is kuat then keputusan is tidak bersalah.
4. **IF** Dukungan1 is mendukung **AND** Dukungan2 is mendukung **AND** Kekuatan1 is tidak kuat **AND** Kekuatan2 is tidak kuat then keputusan is ditunda.
5. **IF** Dukungan1 is mendukung **AND** Dukungan2 is tidak mendukung **AND** Kekuatan1 is kuat **AND** Kekuatan2 is kuat then keputusan is bersalah.
6. **IF** Dukungan1 is mendukung **AND** Dukungan2 is tidak mendukung **AND** Kekuatan1 is kuat **AND** Kekuatan2 is tidak kuat then keputusan is bersalah.
7. **IF** Dukungan1 is mendukung **AND** Dukungan2 is tidak mendukung **AND** Kekuatan1 is tidak kuat **AND** Kekuatan2 is kuat then keputusan is ditunda.
8. **IF** Dukungan1 is mendukung **AND** Dukungan2 is tidak mendukung **AND** Kekuatan1 is tidak kuat **AND** Kekuatan2 is tidak kuat then keputusan is bersalah.
9. **IF** Dukungan1 is tidak mendukung **AND** Dukungan2 is mendukung **AND** Kekuatan1 is kuat **AND** Kekuatan2 is kuat then keputusan is tidak bersalah.

10. **IF** Dukungan1 is tidak mendukung **AND** Dukungan2 is mendukung **AND** Kekuatan1 is tidak kuat **AND** Kekuatan2 is kuat then keputusan is ditunda.
11. **IF** Dukungan1 is tidak mendukung **AND** Dukungan2 is mendukung **AND** Kekuatan1 is kuat **AND** Kekuatan2 is tidak kuat then keputusan is tidak bersalah.
12. **IF** Dukungan1 is tidak mendukung **AND** Dukungan2 is mendukung **AND** Kekuatan1 is tidak kuat **AND** Kekuatan2 is tidak kuat then keputusan is tidak bersalah.
13. **IF** Dukungan1 is tidak mendukung **AND** Dukungan2 is tidak mendukung **AND** Kekuatan1 is kuat **AND** Kekuatan2 is kuat then keputusan is ditunda.
14. **IF** Dukungan1 is tidak mendukung **AND** Dukungan2 is tidak mendukung **AND** Kekuatan1 is kuat **AND** Kekuatan2 is tidak kuat then keputusan is bersalah.
15. **IF** Dukungan1 is tidak mendukung **AND** Dukungan2 is tidak mendukung **AND** Kekuatan1 is tidak kuat **AND** Kekuatan2 is kuat then keputusan is tidak bersalah.
16. **IF** Dukungan1 is tidak mendukung **AND** Dukungan2 is tidak mendukung **AND** Kekuatan1 is tidak kuat **AND** Kekuatan2 is tidak kuat then keputusan is ditunda.

Kronoligis Kasus

Berdasarkan laporan dari warga sekitar, yang mencurigai tingkah laku Aji, maka pihak kepolisian dan warga melakukan penggerbekan di kediaman Aji. Saat penggerebekan tersebut ditemukan Aji sedang mengkonsumsi narkoba. Dalam penggerebekan teresebut ditemukan aji bersama 2 orang teman. Sementara 2 orang temannya nya mengaku tidak memakai narkoba. Setelah proses pemeriksaan, 2 orang temannya teresebut benar tidak menggunakan narkoba. Sedangkan Aji setelah proses pemeriksaan dia positif menggunakan narkoba. Dalam TKP ditemukan barang bukti ¼ gr ganja kering, 1 buah handphone nokia,

uang 1 juta, alat hisap dan pisau.

Setelah melakukan penangkapan pihak kepolisisan memanggil 5 orang untuk dijadikan saksi dalam proses persidangan yang merupakan ikut dalam penggerebakan. 3 orang merupakan tetangganya yaitu deni, abdul dan engkus, dan 2 orang merupakan teman tersangka yaitu ruli dan wahyu. Setelah msuk ke persidangan, pada tanggal 12 mei 2014 dilakukan sidang pada kasus Aji. Namun dalam persidangan tersebut, hakim menunda dikarenakan terjadi keributan yang dilakukan oleh pihak keluarga yang tidak rela anaknya di hukum. Sehingga proses mendengarkan kesaksian ditunda.

Berdasarkan penjelasan landasan teori tahapan yang dilakukan dengan algoritma fuzzy logic adalah:

a. Fuzzificatiaon
b. Inference
c. Defuzzyfication

Proses Fuzzification

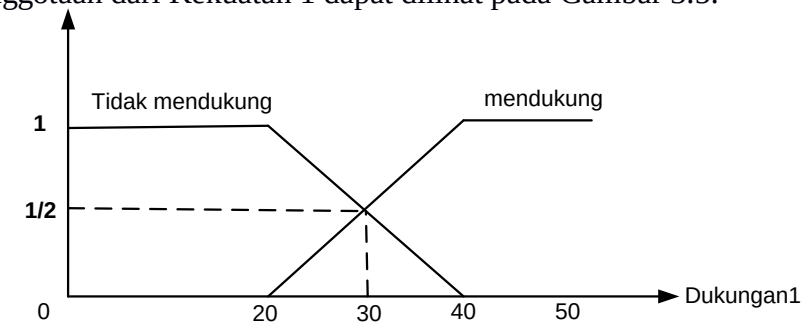
1. Dukungan saksi dari jaksa penuntut umum (Dukungan1)

   Penjelasan Dukungan1 merupakan nilai yang didapatkan dari proses mendengarkan pernyataan saksi, saksi yang di hadirkan berjumlah 5 orang. Kemudian dari pernyataan itu, pemain harus menilai mana saksi yang memberikan pernyataan yang sesuai dengan kronologis penangkapan dan mana yang tidak sesuai. Setiap penilaian yang benar maka mendapatkan nilai Dukungan1 = 10.

   Contoh kasus.

   Misalkan didapatkan nilai Dukungan1 sebesar 30. Pada proses ini maka nilai crips input Dukungan1 30 dikonversi ke nilai fuzzy dengan cara, Dukungan1 30 berada pada nilai linguistik mendukung dan tidak mendukung. Derajat keanggotaan untuk tidak mendukung dihitung menggunakan rumus (Persamaan 2.4) dimana c = 20 dan d = 40 sehingga

derajat keanggotaan tidak mendukung = - (30 – 40) / (40 – 20) = 10/20= ½. Derajat keanggotaan untuk mendukung menggunakan rumus (Persamaan 2.4) dimana a=20 dan b= 40 sehingga derajat keanggotaan mendukung =(30-20) / (40-20) = 10/20 =½. Maka akan didapat grafik derajat keanggotaan dari Kekuatan 1 dapat dilihat pada Gambar 3.5.



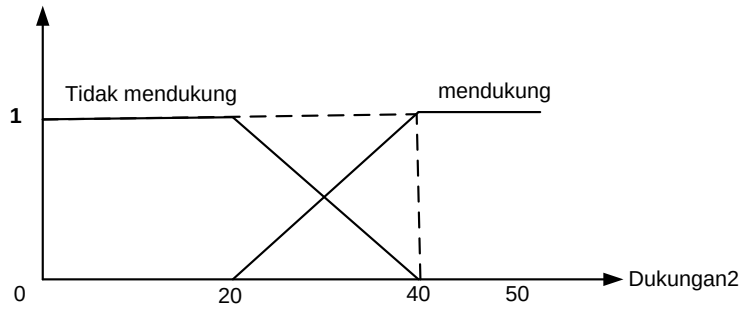
Gambar 3.5 Fungsi keanggotaan trapesium untuk contoh kasus Dukungan1

2. Dukungan saksi dari terdakwa (Dukungan2)

Penjelasan Dukungan2 merupakan nilai yang dihasilkan dari proses mendengarkan pernyataan saksi dari pihak terdakwa. Kemudian pemain harus bisa melakukan proses interupsi jika pernyataan saksi tidak sesuai dengan kasus terdakwa. Jumlah interupsi yang di sediakan sebanyak 5 interupsi. Setiap interupsi yang sesuai mendapatkan nilai Dukungan2 +10.

Contoh kasus:

Misalkan didapatkan nilai Dukungan2 sebesar 40. Pada proses ini maka nilai crips input Dukungan2 40 dikonversi ke nilai fuzzy dengan cara, Dukungan2 40 berada pada nilai linguistik mendukung. Dengan rumus (Persamaan 2.4) maka derajat keanggotaan Dukungan2 = 1. Maka akan didapat grafik derajat keanggotaan dari Kekuatan 1 dapat dilihat pada Gambar 3.6.



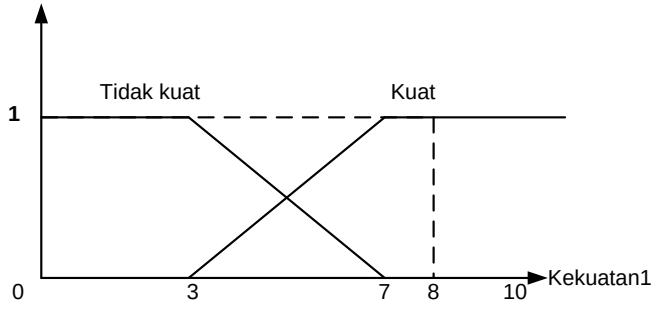
Gambar 3.6 Fungsi keanggotaan trapesium untuk contoh kasus Dukungan2

3. Kekuatan pasal dari pemain (Kekuatan1)

   Penjelasan Kekuatan1 merupakan nilai yang didapatkan dari pemilihan pasal pada kasus terdakwa. Dari 10 pasal yang disediakan tersebut, terdapat 2 pasal yang memiliki kekuatan yang sangat sesuai dengan kasus terdakwa, dan 3 pasal yang hanya berhubungan dengan kasus narkoba saja. Semenatar 5 pasal sisanya merupakan pasal yang tidak berhubungan sama sekali dengan kasus narkoba dan merupakan sebagai pengecoh untuk pemain. Untuk 2 pasal yang sangat sesuai dengan kasus diberikan nilai +3,5 untuk nilai Kekuatan1 dan 3 pasal yang hanya berhubungan dengan kasus narkoba diberikan nilai +1.

   Contoh kasus:

   Misalkan didapatkan nilai Kekuatan1 = 8. Pada proses ini maka nilai crips input Kekuatan1 = 8 dikonversi ke nilai fuzzy dengan cara, Kekuatan1= 8 berada pada nilai linguistik tidak kuat dengan rumus (Persamaan 2.4) didapatkan derajat keanggotaan Kekuatan1 = 1. Maka akan didapat grafik derajat keanggotaan dari Kekuatan1 dapat dilihat pada Gambar 3.7.



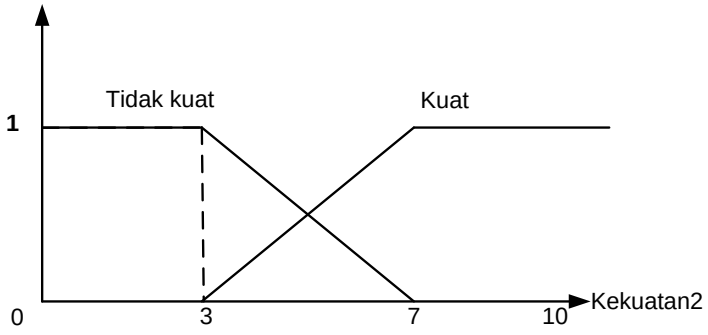
Gambar 3.7 Fungsi keanggotaan trapesium untuk contoh kasus Kekuatan1

4. Kekuatan pasal dari pengacara (Kekuatan2)

   Penjelasan Kekuatan2 merupakan nilai yang dihasilkan dari nilai yang telah di random oleh *NPC* pengacara.

   Contoh kasus:

   Misalkan didaptatkan nilai Kekuatan2 = 3. Pada proses ini maka nilai crips input Kesesuaian2 = 3 dikonversi ke nilai fuzzy dengan cara, Kekuatan2 = 3 berada pada nilai linguistik kuat dengan rumus (Gambar 2.1) derajat keanggotaan Kekuatan2 = 1. Maka akan didapat grafik derajat keanggotaan dari Dukungan1 dapat dilihat pada Gambar 3.8.



Gambar 3.8 Fungsi keanggotaan trapesium untuk contoh kasus Kekuatan2

Pada fuzzyfication menghasilkan 5 fuzzy input:

a. Dukungan1 (saksi penuntut jaksa) = tidak mendukung (1/2) dan mendukung (1/2).
b. Dukungan2 (saksi terdakwa) = mendukung (1)
c. Kekuatan1 (pasal dari pemain) = tidak kuat (1)
d. Kekuatan2 (pasal dari pengacara) = tidak kuat (1)

Dari 5 fuzzy input tersebut maka di dapatkan dua aturan dari 16 aturan yang dapat di aplikasikan.

a. **IF** Dukungan1 is tidak mendukung **AND** dukungan2 is mendukung **AND** Kekuatan1 is kuat **AND** Kekuatan2 is tidak kuat then keputusan is tidak menunda.

b. **IF** Dukungan1 is mendukung **AND** dukungan2 is mendukung **AND** Kekuatan1 is kuat **AND** Kekuatan2 is tidak kuat then keputusan is bersalah.

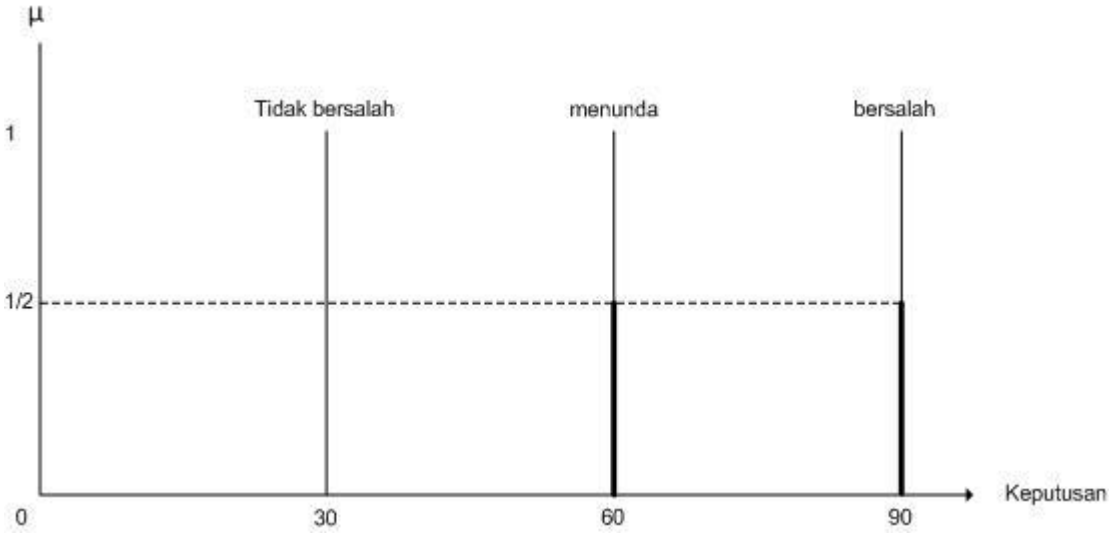
Proses Inference

Dari 2 aturan *fuzzy* dan 5 *fuzzy infut* tersebut maka proses inferece yang terjadi adalah menggunakan aturan Conjunction (∧) dengan memilih derajat keanggotaan minimum sehingga di peroleh:

a. **IF** Dukungan1 is tidak mendukung (1/2) **AND** dukungan2 is mendukung (1) **AND** Kekuatan1 is kuat (1) **AND** Kekuatan2 is tidak kuat (1) then keputusan is menunda (1/2).
b. **IF** Dukungan1 is mendukung (1/2) **AND** dukungan2 is mendukung (1) **AND** Kekuatan1 is kuat (1) **AND** Kekuatan2 is tidak kuat (1) then keputusan is bersalah (1/2).

Dari hasil diatas maka akan dilanjutkan menggunakan Aturan Disjunction (v) dengan memilih derajat keangotaan maksimum . Sehingga dihasikan keputusan is menunda (1/2) ∨ bersalah is (1/2).

Maka akan di dapat grafik keanggotaan model sugeno pada Gambar 3.9



Gambar 3.9 Grafik Derajat Keanggotaan Model Sugeno

Proses Defuzzification

Dilanjutkan dengan proses defuzzyfication menggunakan model sugeno. Dengan menggunakan *Weightd Average* untuk proses *defuzzyfication* maka mengacu pada persamaan 2.8 di atas.

$$y^* = \Sigma \frac{\mu(y)y}{\mu(y)}$$

Maka didapatakan hasil perhitungan sebagai berikut :

$$Y* = \frac{1/2(60)+1/2(90)}{1/2+1/2} = 75$$

Nilai yang didaptkan dari proses defuzyfication tersebut selanjutna dugunakan untuk penentuan keputusan dengan aturan sebagai berikut.

1. Tidak bersalah 0 – 30
2. Menunda 31- 60
3. Bersalah 61 – 90

Dari hasil proses defuzification didapatkan nilai 75. Berdasarkan derajat keanggotaan nilai 75 berada pada keputusan bersalah. Sehingga keptusan nilai 75 adalah beralah.

3.5 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar bisa saling berinteraksi.

Pengguna

Spesifikasi minimum perangkat lunak yang dibutuhkan pengguna untuk memainkan *game* ini dapat dilihat pada tabel 3.3

Tabel 3 2. Spesifikasi Minimum Perangkat Lunak Pengguna

Nama Perangkat Lunak	Spesifikasi
Sistem Operasi	Microsoft Windows XP SP2
Tools	Unity

3.5.1 Analisis Kebutuhan Perangkat Keras

Komputer terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi-instruksi kepada perangkat keras untuk melakukan suatu tugas tertentu, sehingga dapat menjalankan suatu sistem di dalamnya.

Pengguna

Berdasarkan *system requirement* yang ada pada website Unity3D spesifikasi minimum perangkat keras bagi pengguna untuk dapat memainkan *game* ini dengan baik dapat dilihat pada **Error! Reference source not found.**.

Tabel 3.3 Spesifikasi Minimum Perangkat Keras Pengguna

Nama Perangkat	Spesifikasi
Processor	1.80 GHz
RAM	2 GB
VGA Card	250 MB
Harddisk	Tersedia 100 MB
Monitor	Resolusi 800 x 600
Mouse dan *keyboard*	Standar

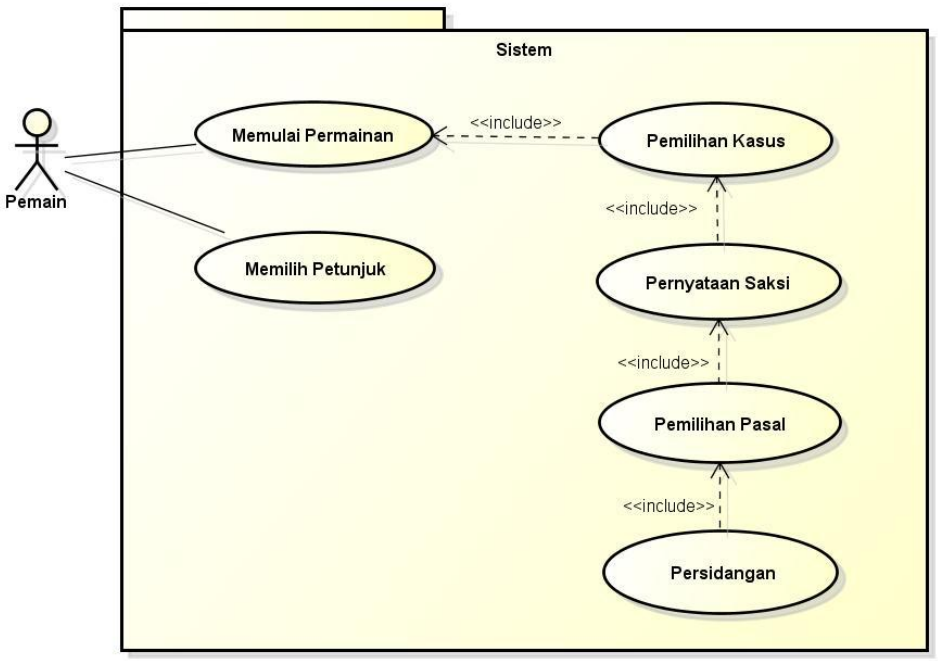
Dengan spesifikasi yang lebih rendah maka pengguna tidak akan memainkan aplikasi *game* dengan maksimal seperti mengalami ketersendatan atau gambar yang kabur.

3.5.2 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. Analisis kebutuhan fungsional dimodelkan dengan menggunakan UML *(Unified Modeling Language)*.

3.5.2.1 Use Case Diagram

Diagram *use case* menggambarkan hubungan antara aktor dan sistem yang dibangun. Implementasi proses-proses yang terdapat pada *game* yang akan dibangun akan digambarkan dengan diagram *use case.*



Gambar 3.10 *Use Case Diagram*

1. **Definisi Aktor**

Definisi aktor berfungsi untuk menjelaskan aktor yang terdapat pada diagram *use case.* Definisi aktor diterangkan pada Tabel 3.8.

Tabel 3.4 Definisi Aktor

No.	Aktor	Deskripsi
1.	Pemain	Pengguna atau orang yang memainkan *game*.

2. **Definisi *Use Case***

Definisi *use case* berfungsi untuk menjelaskan fungsi *use case* yang terdapat pada diagram *use case*. Definisi *use case* diterangkan pada 5.

Tabel 3.5.

Tabel 3.5 Definisi *Use Case*

No.	*Use Case*	Deskripsi
1	Memulai permainan	Proses untuk memulai permainan atau *game*.
2	Melihat petunjuk	Proses untuk menampilkan petunjuk atau panduan cara bermain.
3	Memainkan permainan	Proses memainkan permainan
4	Pemilihan kasus	Proses memilih kasus
5	Pemilihan Pasal	Proses pemilihan pasal yang sesuai dengan kasus terdakwa.
6	Pernyataan saksi	Proses memilih pernyataan saksi dari pihak Jaksa Penuntut.
7	Persidangan	Proses persidangan.

3. Skenario *Use Case*

Skenario setiap bagian pada *use case* menunjukkan proses apa yang terjadi pada setiap bagian didalam *use case* tersebut, dimana pengguna memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh sistem kepada pengguna setelah pengguna memberikan perintah pada setiap bagian - bagian *use case*.

Tabel 3.6 Skenario *Use Case* Memulai Permainan

Identifikasi	
No	1
Nama	Memulai permainan
Tujuan	Untuk mmulai permainan
Deskripsi	Proses untuk memulai permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Aktor berada di dalam permainan
Aksi Aktor	**Reaksi Sistem**
1. Memulai aplikasi	
	2. Menampilkan layar menu utama
3. Memilih menu mulai	
	4. Masuk ke proses permainan.
Kondisi Akhir	Permainan di mainkan

Tabel 3.7 Skenario *Use Case* Melihat Petunjuk

Identifikasi	
No	2
Nama	Memilih Petunjuk
Tujuan	Untuk melihat petunjuk permainan
Deskripsi	Proses untuk melihat petunjuk permainan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Aktor berada di menu utama
Aksi Aktor	**Reaksi Sistem**
Skenario Normal	
1. Memilih menu petunjuk	
	2. Menampilkan petunjuk bermain
3. Menutup tampilan petunjuk	
	4. Menampilkan menu utama
Skenario Alternatif	
1. Memilih tombol kembali	
	2. Menampilkan layar menu utama
Kondisi Akhir	Pemain berada di menu petunjuk

Tabel 3.8 Skenario *Use Case* Pemilihan Kasus

Identifikasi	
No	3
Nama	Memilih Kasus
Tujuan	Untuk memilih kasus
Deskripsi	Proses untuk melihat memilih kasus
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Aktor berada di dalam permainan
Aksi Aktor	**Reaksi Sistem**
Skenario Normal	
	1. Masuk ke proses permainan
	2. Menampilkan layar pilihan kasus
3. Memilih kasus	

	4. Menampilkan layar deskripsi kasus
5. Memilih tombol lanjutkan	
	6. Masuk ke pernyataan saksi
Skenario Alternatif	
1. Memilih tombol kembali	
	2. Menampilkan layar menu utama
Kondisi Akhir	Pemain memilih kasus

Tabel 3.9 Skenario *Use Case* Pernyataan Saksi

Identifikasi	
No	4
Nama	Pernyataan saksi
Tujuan	Untuk mendengarkan pernyataan saksi
Deskripsi	Proses untuk mendengarkan pernyataan saksi
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Aktor berada di dalam permainan
Aksi Aktor	**Reaksi Sistem**
	1. Masuk ke proses pernyataan saksi
	2. Menampilkan layar permainan pernyataan saksi
	3. Memunculkan dialog pernyataan saksi 1
4. Menekan tombol keyboard (spasi)	
	5. Memunculkan dialog pernyataan saksi berikutnya
	6. Menampilkan dialog pernyataan saksi selesai
	7. Menampilkan tombol lanjut.
	8. Menampilkan layar pemilihan saksi
9. Memilih saksi yang akan di pilih	
	10. Meyimpan pilihan saksi
	11. Menampilkan informasi saksi yang terpilih
	12. Menampilkan informasi waktu pemilihan saksi habis
	13. Menampilkan tombol lanjut
14. Menekan tombol lanjut	

	15. Masuk ke layar permainan pemilih pasal
Kondisi Akhir	Pemain memilih saksi

Tabel 3.10 Skenario *Use Case* Pemilihan Pasal

Indentifikasi	
No	4
Nama	Proses pemilihan pasal
Tujuan	Untuk melakukan pemilihan pasal yang sesuai dengan kasus terdakwa
Deskripsi	Proses untuk pemilihan pasal
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Aktor berada di dalam permainan
Aksi Aktor	**Reaksi Sistem**
	1. Masuk ke layar permainan pemilihan pasal
	2. Menampilkan layar permainan pemilihan pasal
3. Menekan tombol next	
	4. Menampilkan pasal berikutnya
5. Menekan tombol prev	
	6. Menampilkan pasal sebelumnya
7. Menekan tombol pilih	
	8. Menyimpan pasal yang telah di pilih
9. Menekan tombol hapus	
	10. Menghapus pasal yang telah di pilih
11. Menekan tombol lanjut	
	12. Masuk ke layar persidangan
Kondisi Akhir	Pemain memilih pasal

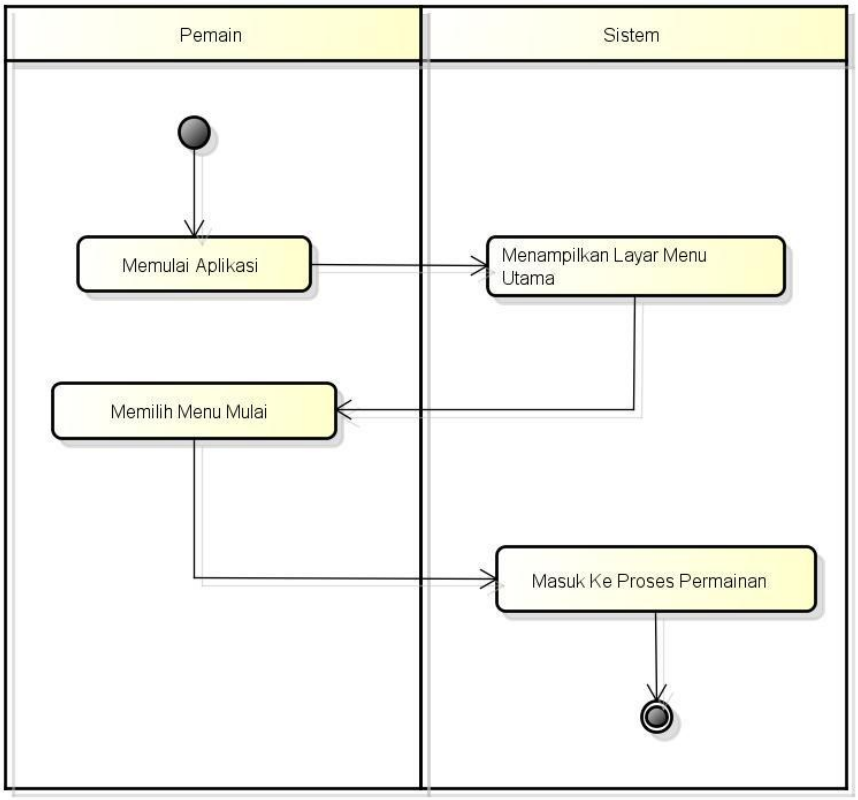
Tabel 3.11 Skenario *Use Case* Proses Persidangan

Indentifikasi	
No	5
Nama	Proses persidangan
Tujuan	Untuk melakukan proses persidangan
Deskripsi	Proses untuk memulai persidangan
Aktor	Pemain
Skenario Utama	
Kondisi Awal	Aktor berada di dalam permainan
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan layar sidang
	2. Menampilkan informasi dialog berita acara 1
	3. Menampilkan dialog berita acara 1
4. Menekan tombol keyboard (spasi)	
	5. Menampilkan dialog berita acara berikutnya
	6. Menampilkan informasi dialog berita acara 2
	7. Menampilkan dialog berita acara 2
8. Menekan tombol keyboard (spasi)	
	9. Menampilkan pilihan interupsi
10. Menekan tombol interupsi	
	11. Menyimpan Pilihan Interupsi
	12. Menampilkan informasi dialog berita acara 3
13. Menekan tombol keyboard (sapasi)	
	14. Menampilkan dialog keputusan akhir
	15. Proses Fuzzy
	16. Menampilkan pesan game selesai
Kondisi Akhir	Permainan selesai

3.5.2.2 *Activity Diagram*

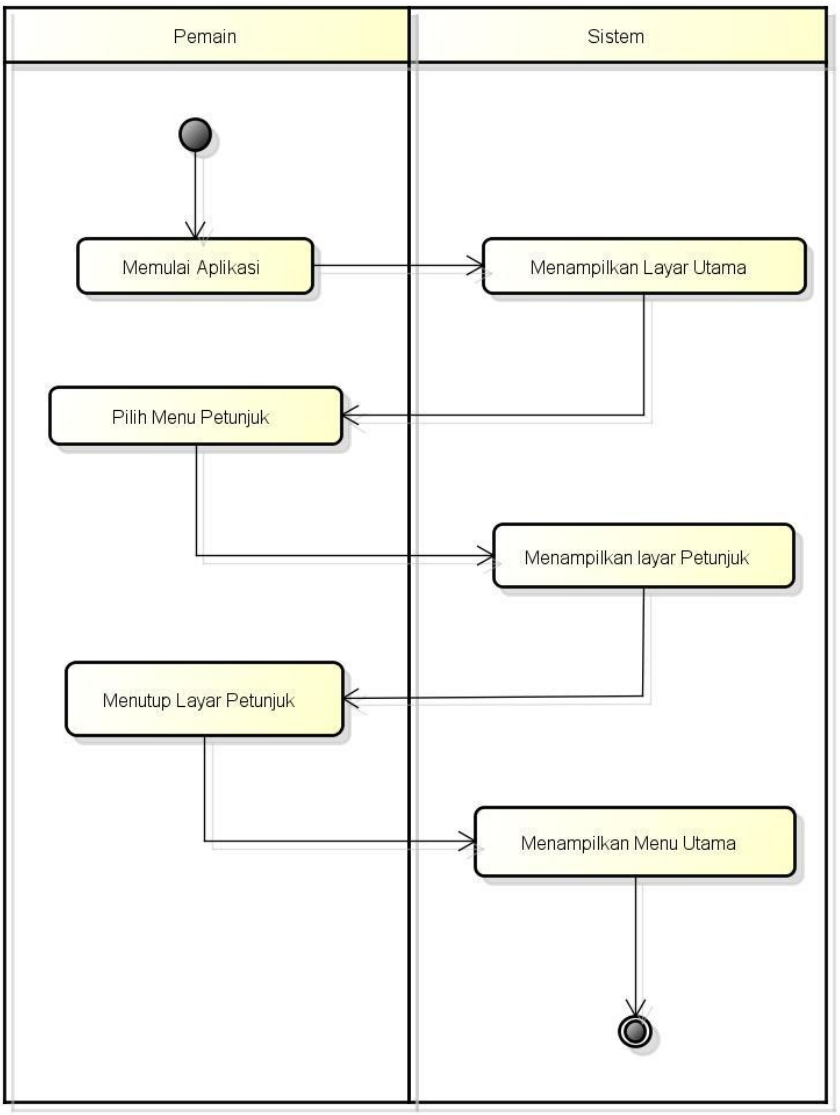
Diagram *activity* menggambarkan proses-proses yang terjadi mulai aktivitas dimulai sampai aktivitas berhenti pada sistem yang sedang dirancang. Bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

1. ***Activity Diagram* Memulai Permainan**



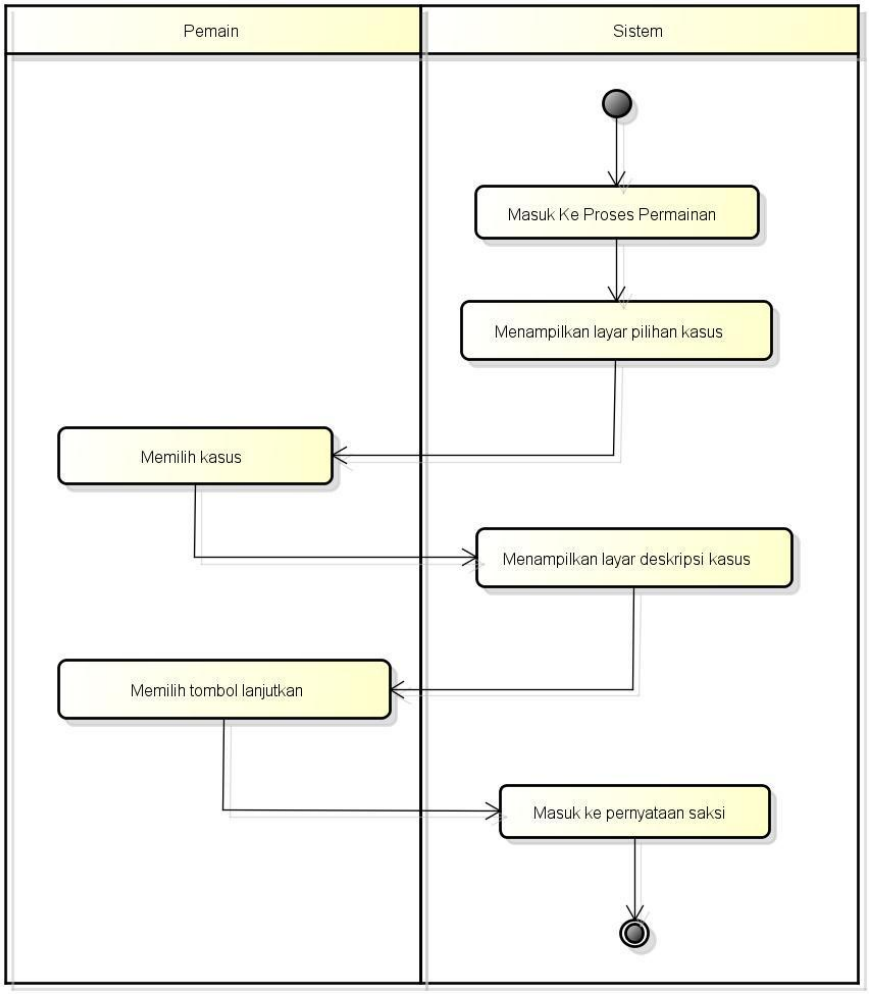
Gambar 3.11 *Activity Diagram* Memulai Permainan

2. *Activity Diagram* Melihat Petunjuk



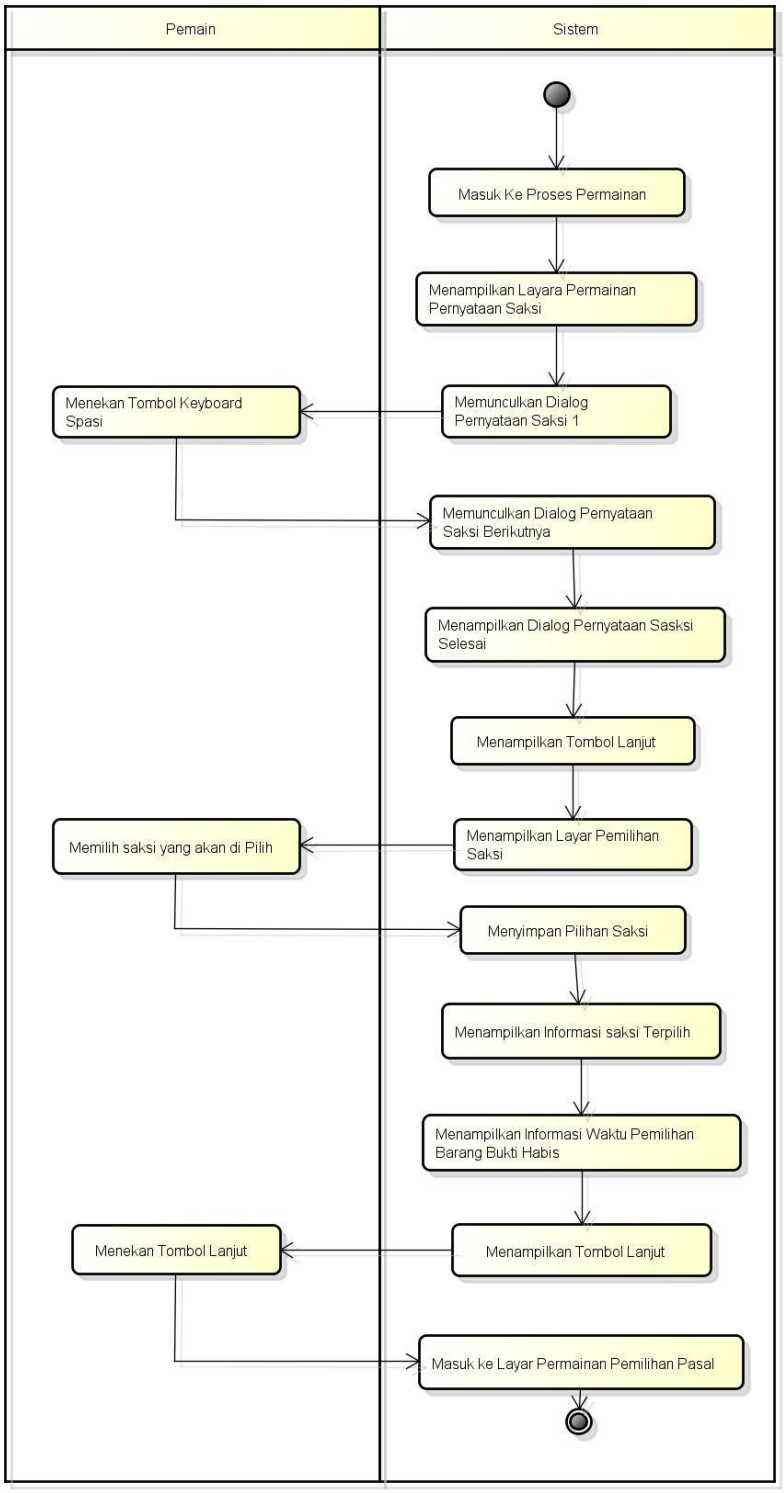
Gambar 3.12 *Activity Diagram* Melihat Petunjuk

3. *Activity Diagram* Pemilihan Kasus



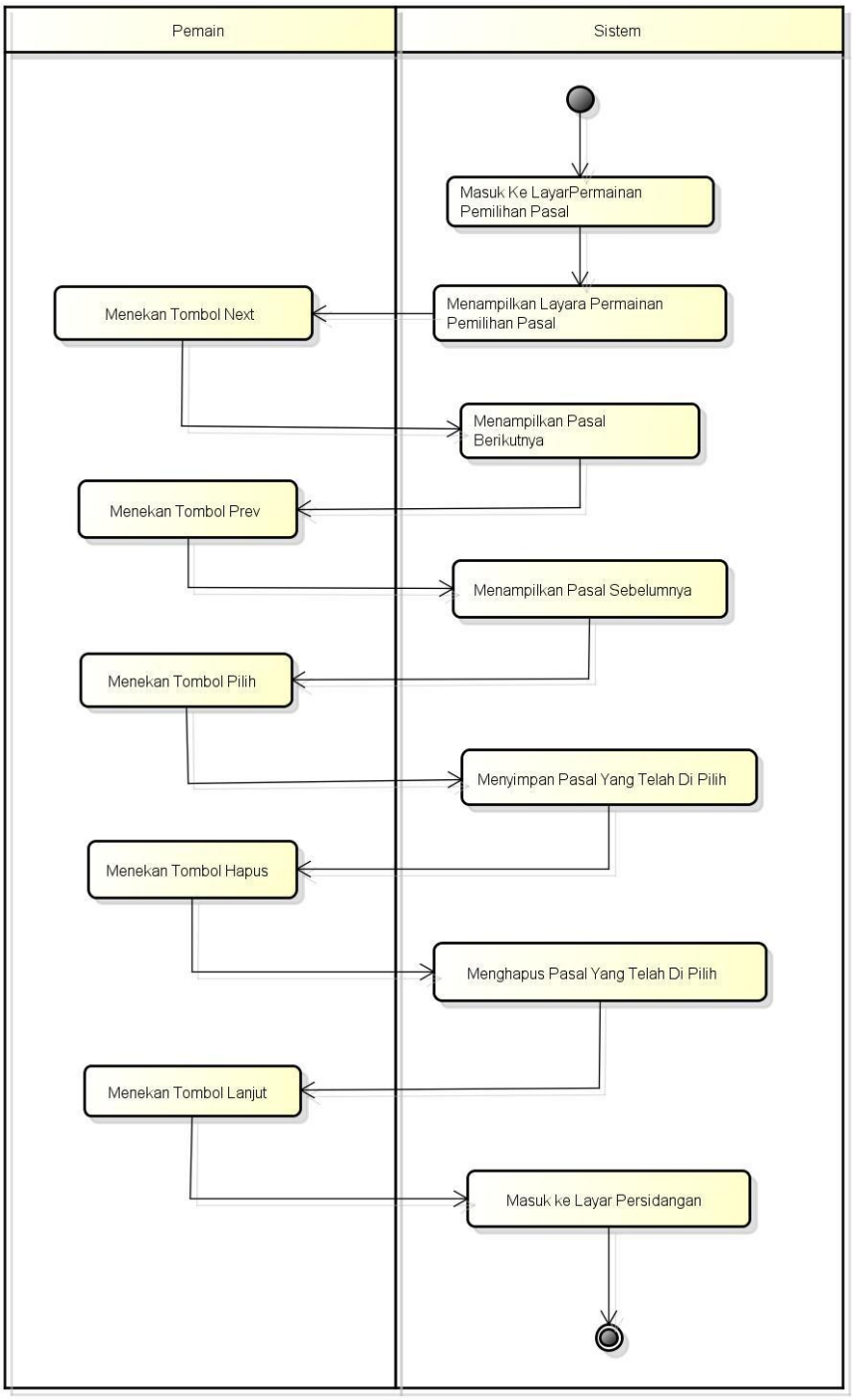
Gambar 3.13 *Activity Diagram* Pemilihan Kasus

4. *Activity Diagram* Pernyataan Saksi



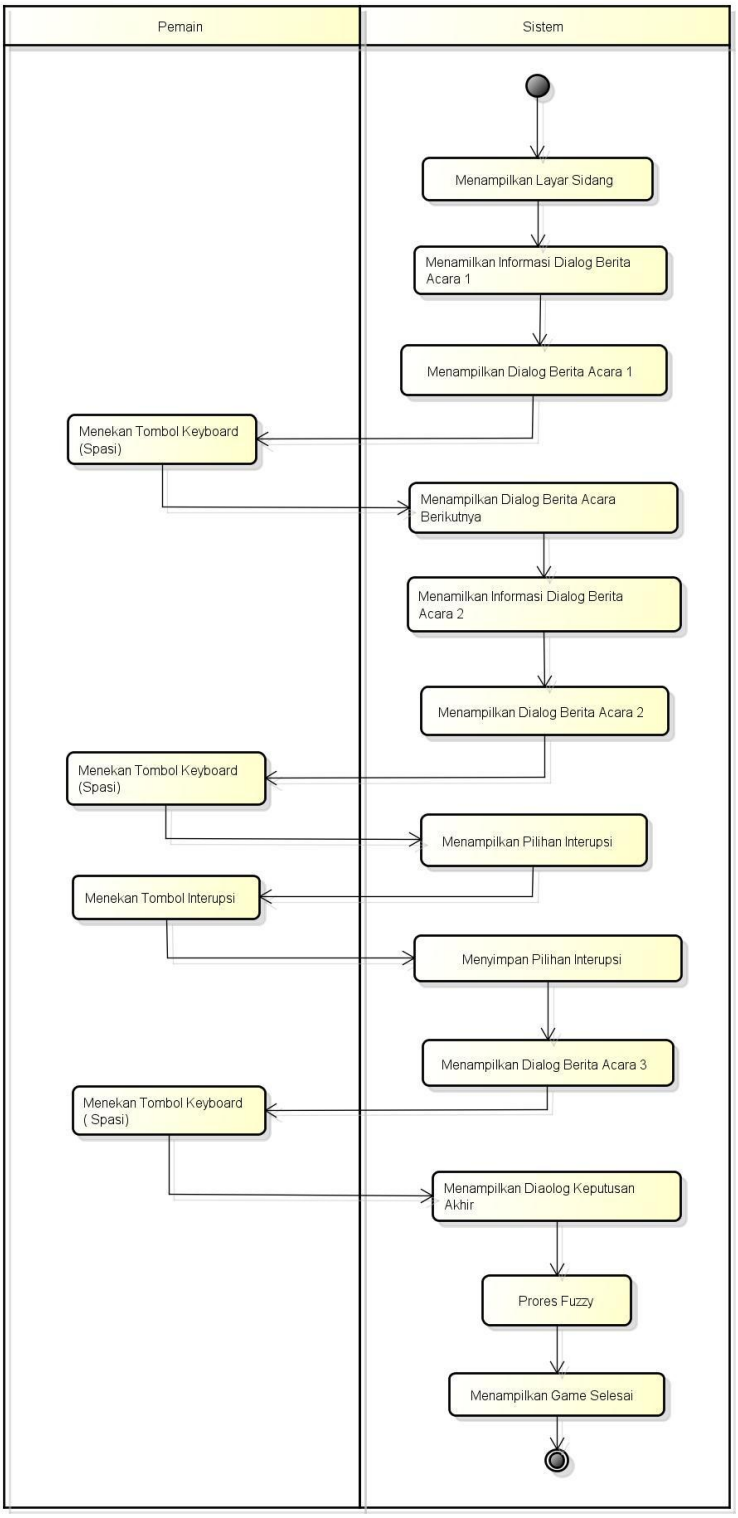
Gambar 3.14 *Activity Diagram* Pernyataan Saksi

5. *Activity Diagram* Pemilihan Pasal



Gambar 3.15 *Activity Diagram* Pemilihan Pasal

6. *Activity Diagram* Proses Persidangan

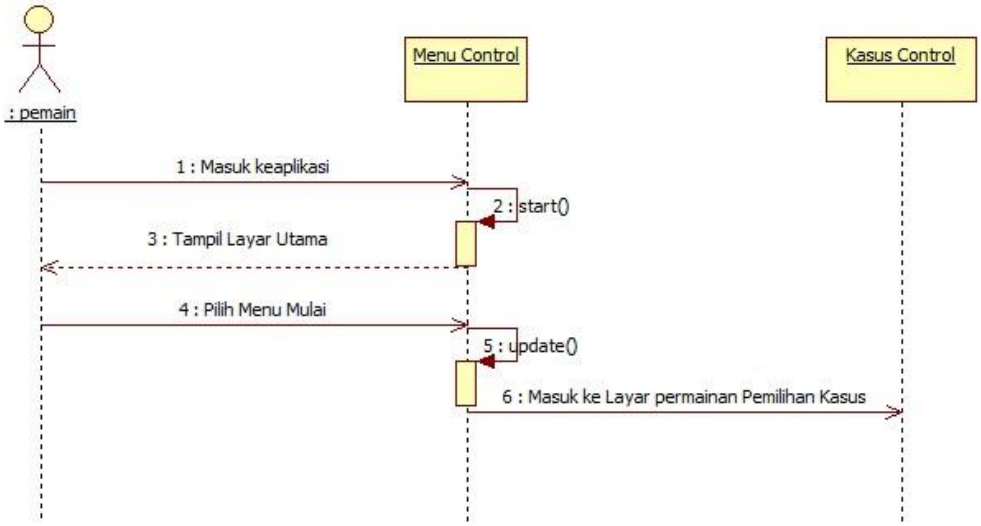


Gambar 3.16 *Activity Diagram* Proses Persidangan

3.5.2.3 Sequence Diagram

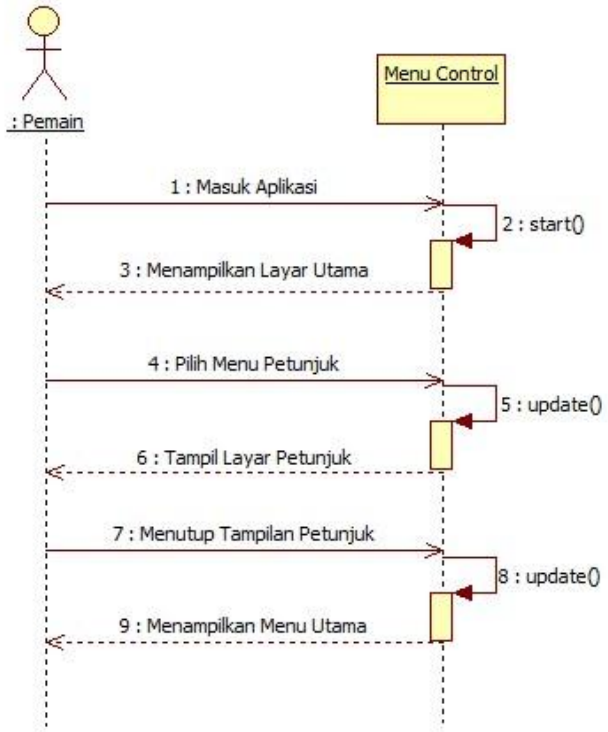
Sequence diagram merupakan penggambaran dari interaksi antar objek di dalam dan sekitar sistem (termasuk pengguna, *display*, dan sebagainya). *Sequence diagram* terdiri atas dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait).

1. ***Sequence Diagram*** **Memulai Permain**



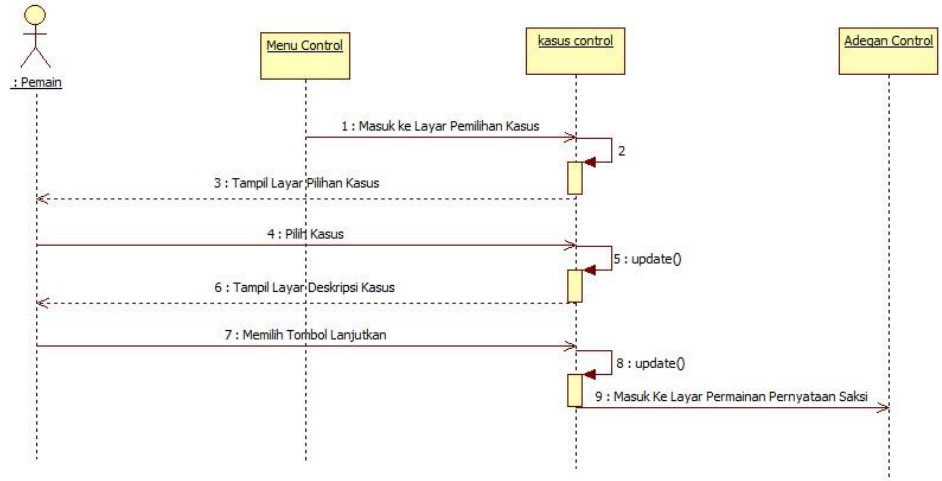
Gambar 3.17 ***Sequence Diagram*** **Memulai Permainan**

2. *Sequence Diagram* Melihat Petunjuk



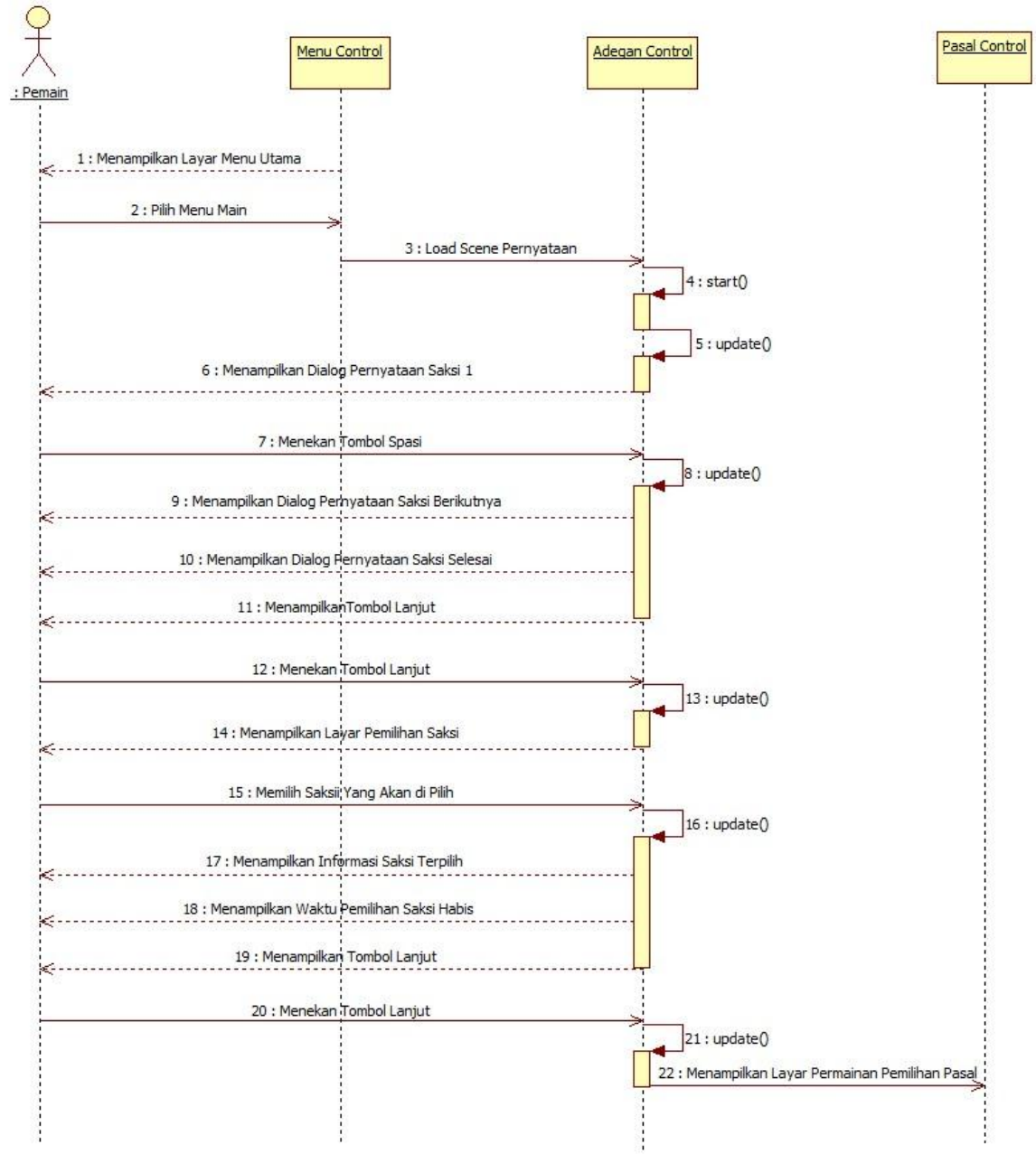
Gambar 3. 18 *Sequence Diagram* Melihat Petunjuk

3. *Sequence Diagram* Pemilihan Kasus



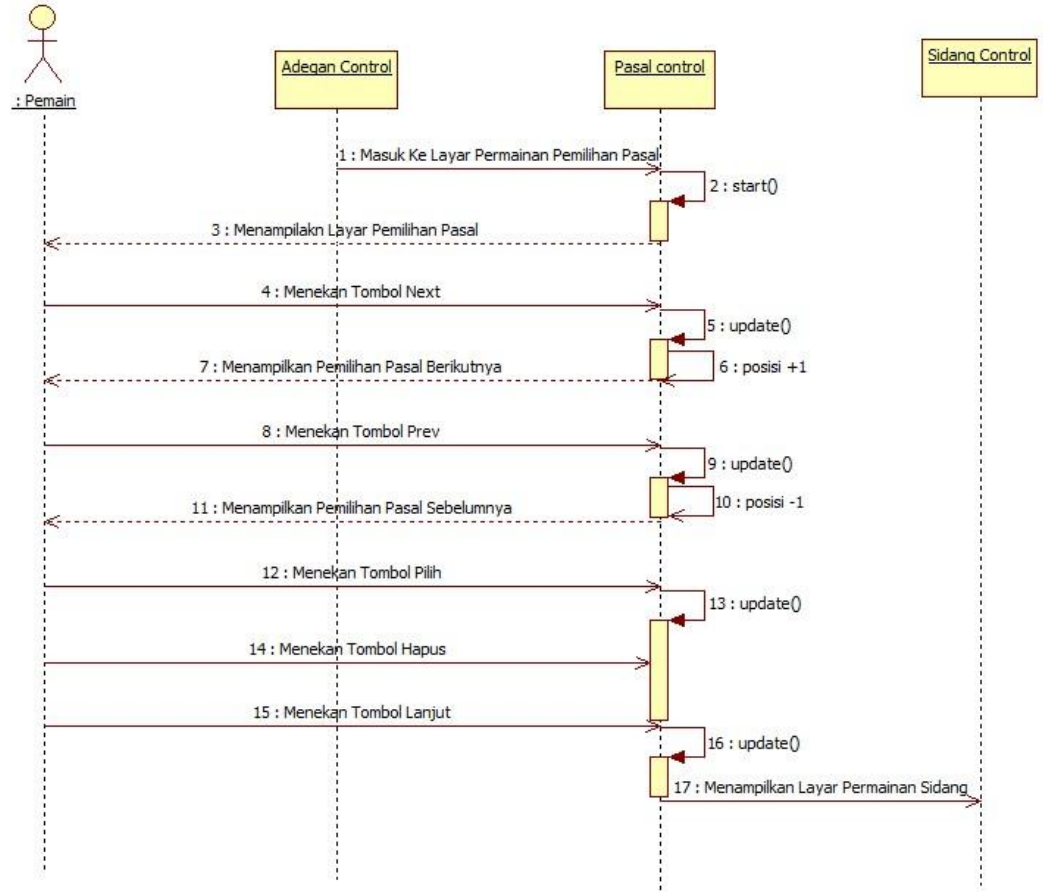
Gambar 3. 19 *Sequence Diagram* Pemilihan Kasus

4. Sequence Diagram Pernyataan Saksi



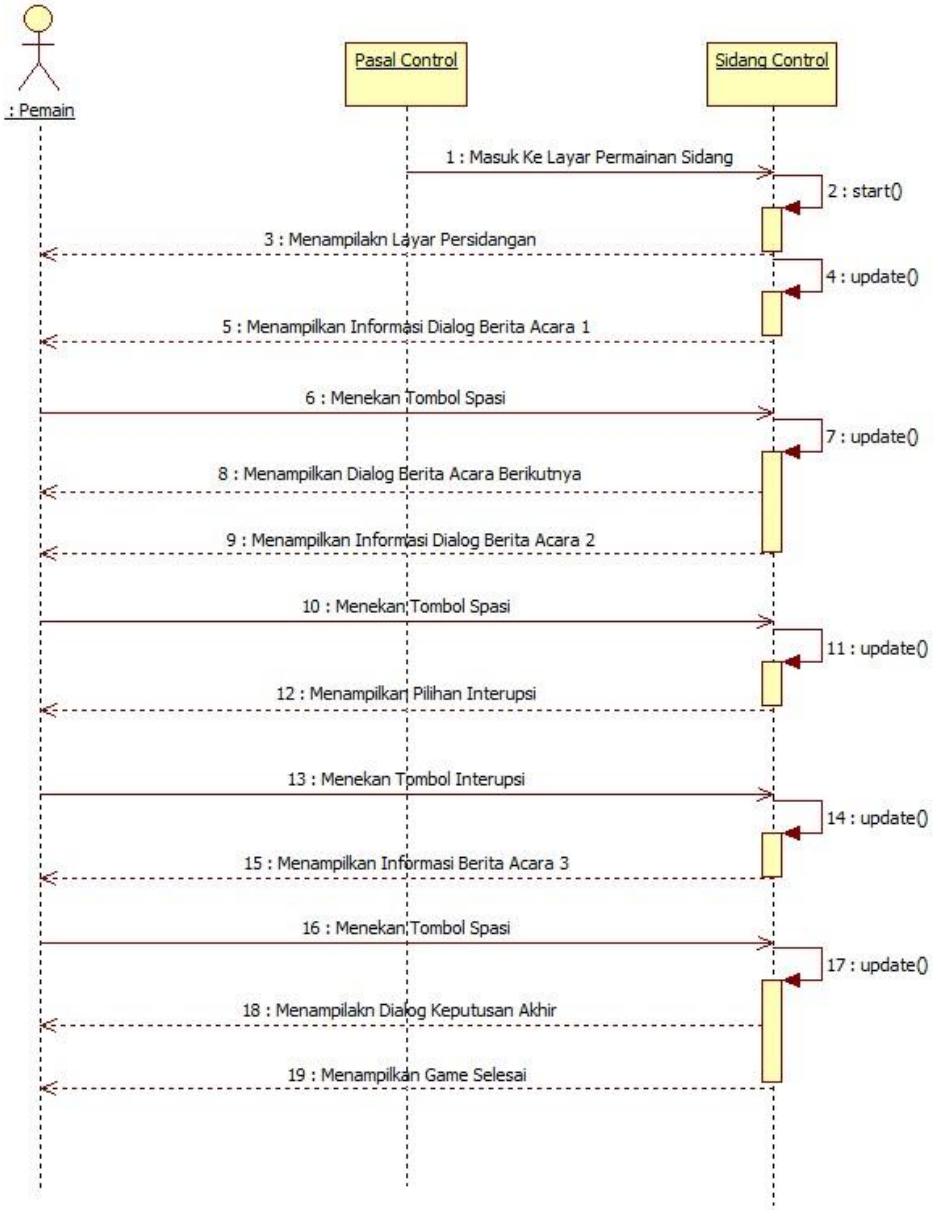
Gambar 3.20 *Sequence Diagram* Pernyataan Saksi

5. Sequence Diagram Pemilihan Pasal



Gambar 3.21 *Sequence Diagram* Pemilihan Pasal

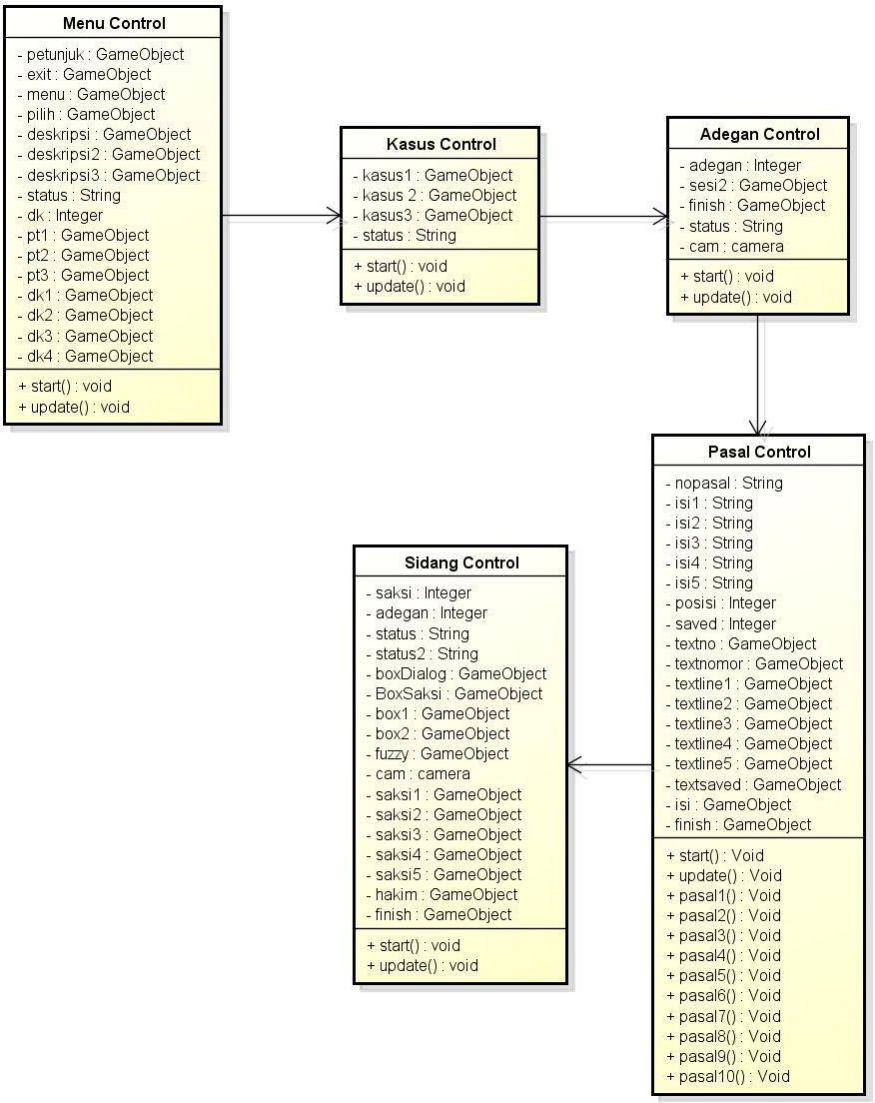
6. Sequence Diagram Persidangan



Gambar 3.22 *Sequence Diagram* Persidangan

3.5.2.4 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas.



Gambar 3.23 *Class Diagram* Game Peranan Jaksa Penuntut

3.5.2.5 Perancangan Class

Merupakan kumpulan class yang ada pada sistem yang dirancang, dan berfungsi untuk membangun sistem. Dan berikut ini adalah penjelasan nya.

Table 3.12 Deskripsi Class

Class	Jenis Class	Deskripsi
menu	*Control*	Class yang berisiakan method memulai permainan
kasus	*Control*	Class yang berisiakan method pemilihan kasus
adegan	*Control*	Class yang berisikan method untuk percakapan saksi dengan jaks
pasal	*Control*	Class yang berisikan method pemilihan pasal
sidang	*Control*	Class yang berisikan method persidangan

Dalam *Class* diagram terdapat beberapa atribut di dalam nya. Adapaun penjelasan nya adalah sebagai berikut.

Table 3.13 Perancangan Class

1. *Class* Menu

Nama Atribut	Tipe Data	Visibility
petunjuk	*GameObjek*	*public*
exit	*GameObjek*	*public*
menu	*GameObjek*	*public*
pilih	*GameObjek*	*public*
status	*String*	*public*
deskripsi	*GameObjek*	*public*
deskripsi2	*GameObjek*	*public*
pt	*Integer*	*public*
dk	*Integer*	*public*
pt1	*GameObjek*	*public*
pt2	*GameObjek*	*public*
pt3	*GameObjek*	*public*
dk1	*GameObjek*	*public*
dk2	*GameObjek*	*public*
dk3	*GameObjek*	*public*

2. *Class* Kasus

Nama Atribut	Tipe Data	Visibility
kasus1	*GameObjek*	*public*
kasus1	*GameObjek*	*public*
kasus1	*GameObjek*	*public*
status	*String*	*public*

3. *Class* Adegan

Nama Atribut	Tipe Data	Visibility
adegan	*Integer*	*public*
sesi2	*GameObjek*	*public*
finish	*GameObjek*	*public*
status	*String*	*public*
cam	*camera*	*public*

4. *Class* Pasal

Nama Atribut	Tipe Data	Visibility
noPasal	*String*	*public*
isi1	*String*	*public*
isi2	*String*	*public*
isi3	*String*	*public*
isi4	*String*	*public*
isi5	*String*	*public*
posisi	*Integer*	*public*
saved	*Integer*	*public*
textno	*GameObjek*	*public*
textnomor	*GameObjek*	*public*
textline1	*GameObjek*	*public*
textline1	*GameObjek*	*public*
textline1	*GameObjek*	*public*
textline1	*GameObjek*	*public*
textline1	*GameObjek*	*public*
textsaved	*GameObjek*	*public*
isi	*GameObjek*	*public*
finish	*GameObjek*	*public*

5. *Class* Sidang

Nama Atribut	Tipe Data	Visibility
saksi	*Integer*	*public*
adegan	*Integer*	*public*
status	*String*	*public*
status2	*String*	*public*
boxDialog	*GameObjek*	*public*
boxSaksi	*GameObjek*	*public*
cam	*camera*	*public*
saksi1	*GameObjek*	*public*
saksi2	*GameObjek*	*public*
saksi3	*GameObjek*	*public*
saksi4	*GameObjek*	*public*
saksi5	*GameObjek*	*public*
hakim	*GameObjek*	*public*
finish	*GameObjek*	*public*

3.5.2.6 Perancangan Method

Merupakan kumpulan method yang ada pada sistem yang dirancang, dan berfungsi untuk menjalankan sistem. Berikut ini adalah penjelasan nya.

Tabel 3.14 Deskripsi Method

Method	Class	Deskripsi
star()	menu	Method yang berfungsi untuk memulai proses sistem di class menu
update()	menu	Method yang berfungsi untuk merubah tampilan sistem di class menu
star()	kasus	Method yang berfungsi untuk memulai proses sistem di class kasus
update()	kasus	Method yang berfungsi untuk merubah tampilan sistem di class kasus
star()	adegan	Method yang berfungsi untuk memulai proses sistem di class adegan
update()	adegan	Method yang berfungsi untuk merubah tampilan sistem

		di class adegan
star()	pasal	Method yang berfungsi untuk memulai proses sistem di class pasal
update()	pasal	Method yang berfungsi untuk merubah tampilan sistem di class pasal
pasal1()	pasal	Method yang berfungsi merubah pasal ke 1 pada proses pemilihan pasal
pasal2()	pasal	Method yang berfungsi merubah pasal ke 2 pada proses pemilihan pasal
pasal3()	pasal	Method yang berfungsi merubah pasal ke 3 pada proses pemilihan pasal
pasal4()	pasal	Method yang berfungsi merubah pasal ke 4 pada proses pemilihan pasal
pasal5()	pasal	Method yang berfungsi merubah pasal ke 5 pada proses pemilihan pasal
pasal6()	pasal	Method yang berfungsi merubah pasal ke 6 pada proses pemilihan pasal
pasal7()	pasal	Method yang berfungsi merubah pasal ke 7 pada proses pemilihan pasal
pasal8()	pasal	Method yang berfungsi merubah pasal ke 8 pada proses pemilihan pasal
pasal9()	pasal	Method yang berfungsi merubah pasal ke 9 pada proses pemilihan pasal
pasal10()	pasal	Method yang berfungsi merubah pasal ke 10 pada proses pemilihan pasal
star()	sidang	Method yang berfungsi untuk memulai proses sistem di class sidang
update()	sidang	Method yang berfungsi untuk merubah tampilan sistem di class sidang

Tabel 3.15 Perancangan Method

NO	Method	Deskripsi
1	start()	Merupakan method untuk memulai proses sistem.
2	update()	Merupakan method untuk merubah tampilan sistem.
3	pasal1()	Merupakan method untuk merubah pasal ke 1
4	pasal2()	Merupakan method untuk merubah pasal ke 2
5	pasal3()	Merupakan method untuk merubah pasal ke 3
6	pasal4()	Merupakan method untuk merubah pasal ke 4
7	pasal5()	Merupakan method untuk merubah pasal ke 5
8	pasal6()	Merupakan method untuk merubah pasal ke 6
9	pasal7()	Merupakan method untuk merubah pasal ke 7
10	pasal8()	Merupakan method untuk merubah pasal ke 8
11	pasal9()	Merupakan method untuk merubah pasal ke 9
12	pasal10()	Merupakan method untuk merubah pasal ke 10