BAB 2
LANDASAN TEORI

2.1 Kecerdasan Buatan / AI (*Artificial Intelligence*)

Kecerdasan Buatan adalah salah satu cabang Ilmu pengetahuan berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi. Hal Ini biasanya dilakukan dengan mengikuti karakteristik dan analogi berpikir dari kecerdasan/Inteligensia manusia, dan menerapkannya sebagai algoritma yang dikenal oleh komputer. Dengan suatu pendekatan yang kurang lebih fleksibel dan efisien dapat diambil tergantung dari keperluan, yang mempengaruhi bagaimana wujud dari perilaku kecerdasan buatan. AI biasanya dihubungkan dengan Ilmu Komputer, akan tetapi juga terkait erat dengan bidang lain seperti Matematika, Psikologi, Pengamatan, Biologi, Filosofi, dan yang lainnya. Kemampuan untuk mengkombinasikan pengetahuan dari semua bidang ini pada akhirnya akan bermanfaat bagi kemajuan dalam upaya menciptakan suatu kecerdasan buatan.

Pengertian lain dari kecerdasan buatan adalah bagian ilmu komputer yang membuat agar mesin komputer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia. Pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan jaman, maka peran komputer semakin mendominasi kehidupan manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu, komputer diharapkan untuk dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia.

Manusia bisa menjadi pandai dalam menyelesaikan segala permasalahan di dunia ini karena manusia mempunyai pengetahuan dan pengalaman yang diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki oleh seseorang tentu saja diharapkan akan lebih mampu dalam menyelesaikan

permasalahan. Namu bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang mereka miliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Demikian juga dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik. Agar komputer bisa bertindak seperti dan sebaik manusia, maka komputer juga harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar. Untuk itu AI akan mencoba untuk memberikan beberapa metoda untuk membekali komputer dengan kedua komponen tersebut agar komputer bisa menjadi mesin pintar[4].

2.1.1 Lingkup utama kecerdasan buatan

Lingkup utama kecerdasan buatan antara lain[4] :

1. Sistem pakar. Komputer digunakan sebagai saran untuk menyimpan pengetahuan para pakar. Dengan demikian komputer akan memiliki keahlian untuk menyelesaikan masalah dengan meniru keahlian yang dimiliki para pakar
2. Pengolahan bahasa alami. Dengan pengolahan bahasa alami ini diharapkan user mampu berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.
3. Pengenalan ucapan. Melalui pengenalan ucapan diharapkan manusia mampu berkomunikasi dengan komputer dengan menggunakan suara.
4. Robotika dan Sistem sensor.
5. *Computer vision*, mencoba untuk dapat mengintrepetasikan gambar atau objek-objek tampak melalui komputer.
6. *Intelligent Computer aid Instruction*. Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.

2.1.2 Keuntungan Kecerdasan Buatan

Keuntungan kecerdasan buatan antara lain[4] :

1. Kecerdasan buatan lebih bersifat permanen. Kecerdasan alami akan cepat mengalami perubahan. Hal ini dimungkinkan karena sifat manusia yang pelupa. Kecerdasan buatan tidak akan berubah sepanjang sistem komputer dan program tidak mengubahnya.
2. Kecerdasan buatan lebih mudah diduplikasi dan disebarkan. Mentransfer pengetahuan manusia dari satu orang ke orang lain butuh proses dan waktu lama. Disamping itu suatu keahlian tidak akan pernah bisa diduplikasi secara lengkap. Sedangkan jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebuat dapat ditransfer atau disalin dengan mudah dan cepat dari satu komputer ke komputer lain
3. Kecerdasan buatan lebih murah dibanding dengan kecerdasan alami. Menyediakan layanan komputer akan lebih mudah dan lebih murah dibanding dengan harus mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.
4. Kecerdasan buatan bersifat konsisten. Hal ini disebabkan karena kecerdasan busatan adalah bagian dari teknologi komputer. Sedangkan kecerdasan alami senantiasa berubah-ubah.
5. Kecerdasan buatan dapat didokumentasikan. Keputusan yang dibuat komputer dapat didokumentasikan dengan mudah dengan melacak setiap aktivitas dari sistem tersebut. Kecerdasan alami sangat sulit untuk direproduksi.
6. Kecerdasan buatan dapat mengerjakan pekerjaan lebih cepat dibanding dengan kecerdasan alami
7. Kecerdasan buatan dapat mengerjakan pekerjaan lebih baik dibanding dengan kecerdasan alami.

2.1.3 Implementasi kecerdasan buatan dapat diterapkan pada bidang

Inmpelentasi kecerdasan buatan dapat diterapkan pada bidang[4] :

a. Visualisasi komputer

   Kecerdasan buatan pada bidang visualisasi komputer ini memungkinkan sebuah sistem komputer mengenali gambar sebagai *input*. Contohnya mengenali sebuah pola pada suatu gambar.

b. Pengenalan Suara

   Kecerdasan buatan pada pengenalan suara ini dapat mengenali suara manusia. Cara mengenali suara ini dengan mencocokannya pada acuan yang telah diprogramkan terlebih dahulu. Contohnya perintah komputer dengan menggunakan suara *user*.

c. Sistem Pakar

   Kecerdasan buatan pada Sistem Pakar ini memungkinkan sebuah sistem komputer memiliki cara berpikir dan penalaran seorang ahli dalam mengambil keputusan, untuk memecahkan masalah yang ada pada saat itu. Contohnya program komputer yang dapat mendiagnosa penyakit dengan memasukan gejala-gejala yang dialami pasien.

d. Permainan

   Kecerdasan buatan pada permainan ini memungkinkan sebuah sistem komputer untuk memiliki cara berpikir manusia dalam bermain. Contohnya permainan yang memiliki fasilitas orang melawan komputer. Komputer sudah di program sedemikian rupa agar memiliki cara bermain seperti seorang manusia bahkan bisa melebihi seorang manusia .

2.1.4 Metode Pencarian (*Searching*)

Pada dasarnya teknik pencarian (*searching*) dapat dibagi menjadi 2 kelompok besar, yaitu pencarian buta (*blind search*) dan pencarian terbimbing (*heuristic search*). Untuk mengukur performa metode pencarian, terdapat empat kriteria yang dapat dilakukan yaitu[9] :

a. *Completeness* : apakah metode tersebut menjamin adanya solusi jika solusinya ada?
b. *Time Complexity* : berapa lama waktu yang diberikan untuk menemukan solusi tersebut.
c. *Space complexity* : berapa banyak memori yang dibutuhkan untuk menemukan solusi tersebut.
d. *Optimaly* : apakah metode tersebut menjamin menemukan solusi terbaik jika terdapat beberapa solusi yang berbeda.

2.1.5 AI yang digunakan

AI atau kecerdasan buatan adalah faktor penting yang dapat membuat *Game* menjadi lebih menarik.

2.1.5.1 Logika Fuzzy

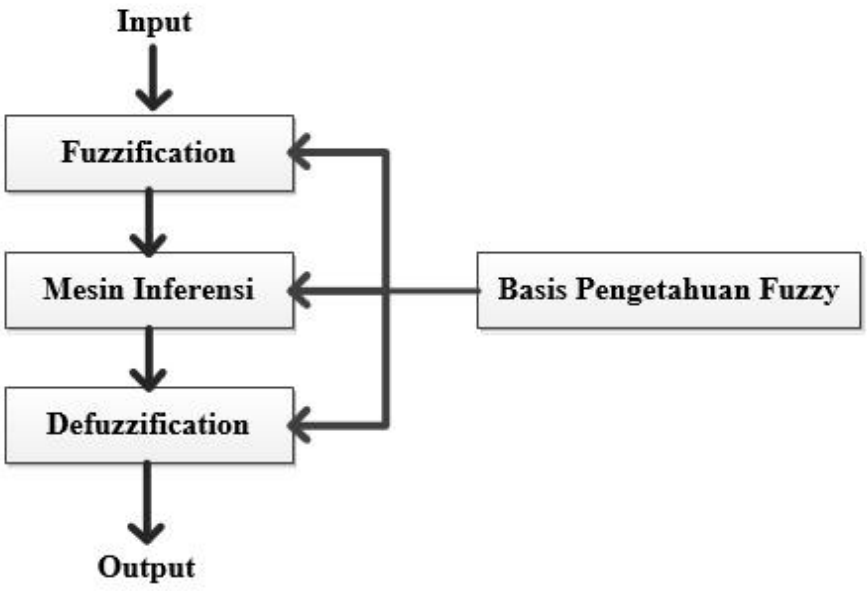
Fuzzy Logic Konsep tentang logika fuzzy diperkenalkan oleh Prof. Lotfi Astor Zadeh pada 1962. Logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah yang cocok diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan PC, *multi-channel* atau *workstation* berbasis akurasi data, dan sistem kontrol. Metodologi ini dapat diterapkan pada perangkat keras, perangkat lunak, atau kombinasi keduanya. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah hanya mempunyai dua kemungkinan, “Ya atau Tidak”, “Benar atau Salah”, “Baik atau Buruk” dan lain-lain. Oleh karena itu, semua ini dapat mempunyai nilai keanggotaan 0 dan 1. Akan tetapi, dalam logika *fuzzy* memungkinakan nilai keanggotaan berada diantara 0 dan 1. Artinya, bisa saja suatu keadaan mempunyai dua nilai “Ya dan Tidak”, “Benar dan Salah”, “Baik dan Buruk” secara bersamaan, namun besar nilainya tergantung pada bobot

keanggotaan yang dimilikinya. Logika *fuzzy* dapat digunakan diberbagai bidang, seperti pada sistem diagnosis penyakit (dalam bidang kedokteran), pemodelan sistem pemasaran, riset operasi (dalam bidang ekonomi), kendali kualitas air, prediksi adanya gempa bumi, klasifikasi dan pencocokan pola (dalam bidang teknik) [4].

Konsep himpunan *fuzzy* memiliki 2 atribut, yaitu :

1. *Linguistik*, yaitu nama suatu kelompok yang mewakili suatu keadaan tertentu dengan menggunakan bahasa alami, misalnya dingin, sejuk, panas mewakili variable temperatur.

2. *Numeris*, yaitu suatu nilai yang menunjukan ukuran dari suatu variabel, misalnya 10, 35, 40 dan sebagainya. Cara kerja logika *fuzzy* dapat digambarkan sebagai struktur elemen dasar sistem inferensi *fuzzy* sebagai berikut:



Gambar 2. 1 Struktur sistem inferensi fuzzy[4].

Keterangan :

1. Basis Pengetahuan Fuzzy merupakan kumpulan *rule-rule fuzzy* dalam bentuk pernyataan IF…THEN.

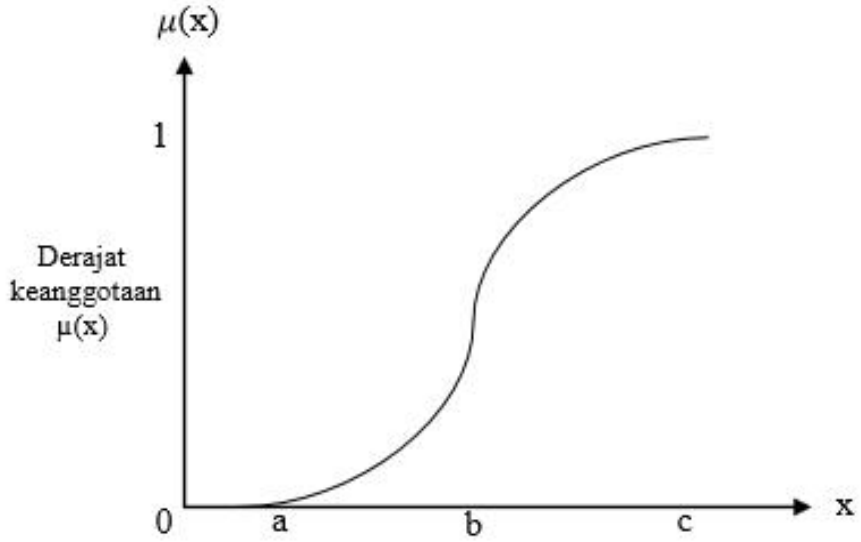
2. Fuzzyfikasi adalah proses untuk mengubah input sistem yang mempunyai nilai tegas menjadi variabel *linguistic* menggunakan fungsi keanggotaan yang disimpan dalam basis pengetahuan *fuzzy*.
3. Mesin Inferensi merupakan proses untuk mengubah *input fuzzy* dengan cara mengikuti aturan-aturan (IF-THEN *Rules*) yang telah ditetapkan pada basis pengetahuan fuzzy.
4. DeFuzzyfikasi merupakan proses mengubah *output fuzzy* yang diperoleh dari mesin inferensi menjadi nilai tegas menggunakan fungsi keanggotaan yang sesuai dengan saat dilakukan fuzzyfikasi.

2.1.5.2 Fungsi Keanggotaan (Member function)

Pada *fuzzy* system, fungsi keanggotaan memainkan peran yang sangat penting untuk merepresentasikan masalah dan menghasilkan keputusan yang akurat. Terdapat banyak sekali fungsi keanggotaan yang dapat digunakan, diantaranya adalah[4] :

1. Fungsi sigmoid

Sesuai dengan namanya, fungsi ini berbentuk kurva sigmoidala seperti huruf S. Setiap nilai x (anggota crips set) dipetakan ke dalam interval [0,1]. Grafik dan notasi matematika untuk fungsi ini adalah sebagai berikut :



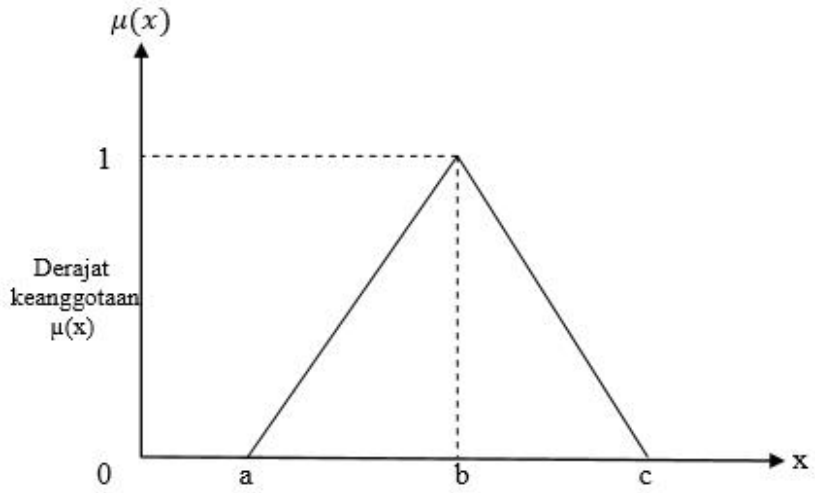
Gambar 2. 2 Grafik fungsi sigmoid[10].

Fungsi keanggotaan :

$$sigmoid(x,a,b,c) = \begin{cases} 0; x \le a \\ \frac{2(x-a)}{(c-a)^2}; a \le x \le b \\ 1 - 2\left(\frac{c-x}{c-a}\right)^2; b \le x \le c \\ 1; x \ge c \end{cases} \quad \text{..... 2.1}$$

2. Fungsi Segitiga

Fungsi segitiga hanya terdapat satu nilai x yang memiliki derajat kenanggotaan sama dengan 1, yaitu ketika x = b. Tetapi, nilai-nilai disekitar b memiliki derajat keanggotaan yang turun cukup tajam (menjauhi 1). Grafik dan notasi matematika dari fungsi segitiga adalah sebagai berikut.



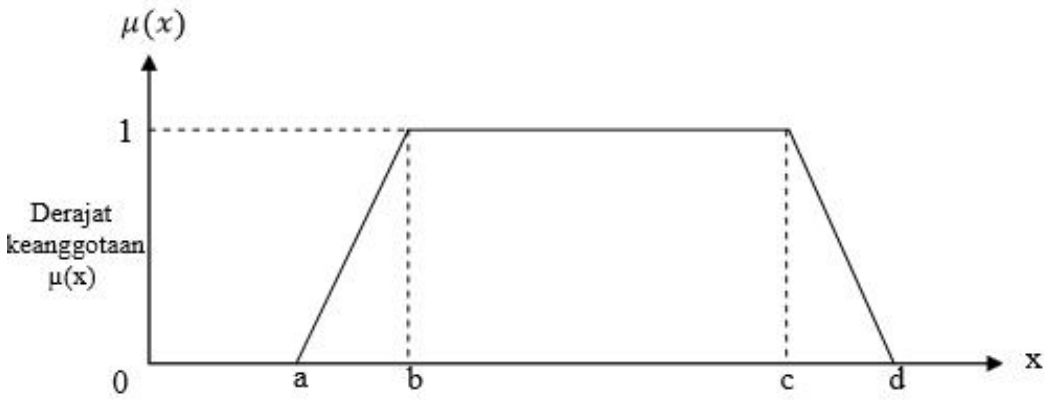
Gambar 2. 3 Graik fungsi segitiga[4].

Fungsi keanggotaan :

$$\mu(x) = \begin{cases} 0; x \leq a \; atau \; x \geq c \\ \frac{x-a}{b-a}; a \leq x \geq b \\ \frac{b-x}{c-b}; b \leq x \geq c \end{cases} \quad \text{..... 2.2}$$

3. Fungsi Trapesium

Berbeda dengan fungsi segitiga, pada fungsi ini terdapat beberapa nilai x yang memiliki derajat keanggotaan sama dengan 1, yaitu ketika b ≤ x ≤ c. Tetapi, derajat keanggotaan untuk c ≤ x ≤ d memiliki karakteristik yang sama dengan fungsi segitiga. Grafik dan notasi matematika dari fungsi Trapesium adalah sebagai berikut[10] :



Gambar 2. 4 Graik fungsi trapesium[4].

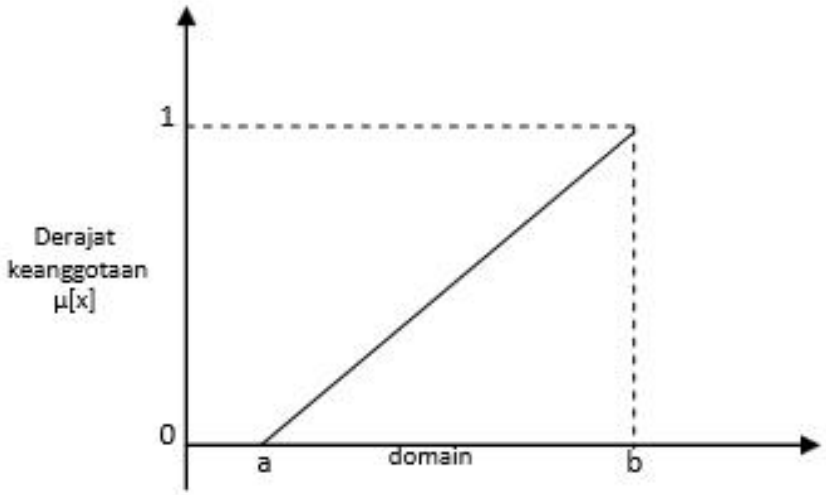
Fungsi Keanggotaan :

$$\mu[X] = \begin{cases} 0 & ; x \leq a \; atau \; x \geq d \\ \frac{x-a}{b-a} & ; a \leq x \leq b \\ 1 & ; b \leq x \leq c \\ \frac{d-x}{d-c} & ; c \leq x \leq d \end{cases} \quad \text{..... 2.3}$$

4. Fungsi Linear

Pada fungsi linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk kurva ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Ada 2 keadaan himpunan fuzzy yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi

Fungsi keanggotaan[4] :

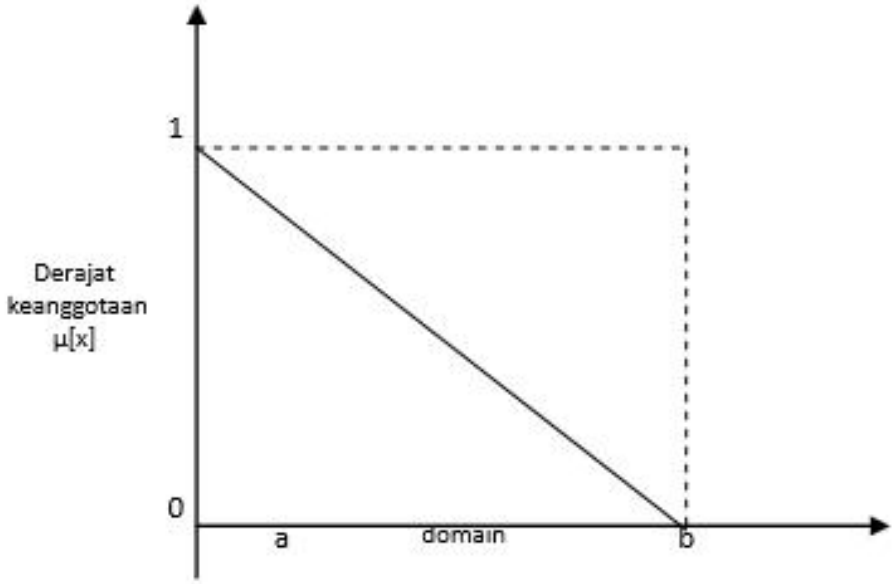


Gambar 2. 5 Representasi Linear Naik[4].

Fungsi Keanggotaan :

$$\mu[X] = \begin{cases} 0 ; x < a \\ \frac{x-a}{b-a} ; a < x \leq b \\ 1 ; x \geq b \end{cases} \quad \text{..... 2.4}$$

Kedua, merupakan kebalikan dari yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah (gambar II.8).



Gambar 2. 6 Representasi Linear Turun[4].

Fungsi Keanggotaan :

$$\mu[X] = \begin{cases} \frac{b-x}{b-a} ; a \leq x < b \\ 0 ; x \geq b \end{cases} \quad \text{..... 2.5}$$

2.1.5.3 Metode Tsukamoto

Pada dasarnya, metode tsukamoto mengaplikasikan penalaran monoton pada setiap aturannya. Kalau pada penalaran monoton, sistem hanya memiliki satu aturan, pada metode tsukamoto, sistem terdiri atas beberapa aturan. Karena menggunakan konsep dasar penalaran monoton, pada metode tsukamoto, setiap konsekuen pada aturan yang berbentuk IF-THEN harus direpresentasikan dengan suatu himpunan fuzzy dengan fungsi keanggotaan yang monoton. Output hasil inferensi dari tiap-tiap aturan diberikan secara tegas (crisp) bedasarkan α-predikat (fire strength). Proses agregasi antar aturan dilakukan, dan hasil akhirnya

diperoleh dengan menggunakan defuzzy dengan konsep rata-rata terbobot. Misalkan ada variabel input, yaitu x dan y, serta satu variabel output yaitu z. Variabel x terbagi atas 2 himpunan yaitu A1 dan A2, variabel y terbagi atas 2 himpunan juga, yaitu B1 dan B2, sedangkan variabel output Z terbagi atas 2 himpunan yaitu C1 dan C2. Tentu saja himpunan C1 dan C2 harus merupakan himpunan yang bersifat monoton. Diberikan 2 aturan sebagai berikut:

1. IF x is A1 and y is B2 THEN z is C1
2. IF x is A2 and y is B2 THEN z is C1

2.1.5.4 Metode Mamdani

Salah satu metode yang memiliki struktur yang sederhana , metode ini paling sering digunakan dalam aplikasi aplikasi. Metode ini menggunakan Opeasi MIN dan MAX atau MAX PRODUCT. Dalam pembuatan aplikasi yang menggunakan Metode Mamdani ini, dapat menggunakan 4 langkah tahapan untuk mendapatkan output yang diharapkan. Tahapan – Tahapan Tersebut diantaranya :

1. Fuzzyfikasi
2. Pembentukan basis Pengetahuan Fuzzy (Rule dalam bentuk IF—-THEN)
3. Aplikasi fungsi implikasi menggunakn fungsi MIN dan Komposisi antar-rule menggunakan fungsi MAX *menghasilkan himpunan Fuzzy baru)
4. Deffuzykasi menggunakan “metode Centroid”.
   Input dari proses *defuzzyfikasi* adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy*tersebut.

2.1.5.5 Metode Sugeno

Logika fuzzy yang pertama kali diperkenalkan oleh Lotfi A. Zadeh, memiliki derajat keanggotaan dalam rentang 0(nol) hingga 1(satu), berbeda dengan logika digital yang hanya memiliki dua nilai yaitu 1(satu) atau 0(nol). Proses *fuzzy inference* dapat dibagi dalam lima bagian [8], yaitu :

1. Fuzzyfikasi Input : FIS mengambil masukan- masukan dan menentukan derajat keanggotaannya dalam semua *fuzzy set*.
2. Operasi logika *fuzzy* :Hasil akhir dari operasi ini adalah derajat kebenaran antecedent yang berupa bilangan tunggal.
3. Implikasi :Merupakan proses mendapatkan *consequent* atau keluaran sebuah IF-THEN *rule* berdasarkan derajat kebenaran *antacedent*. Proses ini menggunakan mengambil nilai MIN/terkecil dari dua bilangan : Hasil operasi fuzzy logic OR dan fuzzy set banyak.
4. Agregasi :Yaitu proses mengkombinasikan semua keluar IF-THEN rule menjadi sebuah *fuzzy set* tunggal dengan menggunakan fungsi MAX.
5. Defuzzyfikasi : Keluaran dari defuzzyfikasi adalah sebuah bilangan tunggal, cara mendapatkannya ada beberapa versi, yaitu *centroid, bisector, middle of maximum, largest of maximum dan smallest of maximum.*

Secara umum model *fuzzy* SUGENO terdiri dari dua jenis :

1. Model fuzzy SUGENO orde-nol : IF input1 = x dan input2 = y, THEN Outputnya adalah z = k.

2. Model fuzzy SUGENO orde-satu : IF input1 = x dan input2 = y, THEN Outputnya adalah z = ax+by+c

Defuzzifikasi dilakukan dengan cara mencari rata-ratanya (weight average/wtaver) :

Final Output $= \frac{\sum_{i=1}^{n} WiZi}{\sum_{i=1}^{n} Wi}$ 2.6

Dimana :

N = Jumlah fuzzy rule

w = Bobot hasil implikasi

z = Output perilaku musuh

2.2 Permainan (*Game*)

Game merupakan kata yang berasal dari bahasa inggris yang berarti permainan. Game atau permainan ini pun tidak sembarang dimana di dalamnya terdapat berbagai aturan yang harus dipahami oleh penggunanya. Dalam game juga perlu adanya sekenario agar alur permainan pun jelas dan terarah. Sekenario

di sini bisa meliputi setting map, level, alur cerita, bahkan efek yang ada dalam game tersebut [3].

2.2.1 Permainan Video (*Video Game*)

Permainan video (*video game*) adalah permainan yang menggunakan interaksi antarmuka pengguna melalui gambar yang dihasilkan oleh piranti video. Permainan video umumnya menyediakan sistem penghargaan, misalnya *score* yang dihitung berdasarkan tingkat keberhasilan yang dicapai dalam menyelesaikan tugas-tugas yang ada di dalam permainan video tersebut. Sistem elektronik yang digunakan untuk menjalankan permainan video disebut *platform*, contohnya adalah komputer pribadi (*personal computer*) ataupun konsol permainan (*game console*)[3].

2.2.2 Permainan komputer (*Computer Game*)

Permainan komputer adalah permainan video yang di mainkan pada komputer pribadi, dan bukan pada konsol permainan. Permainan komputer telah berevolusi dari sistem grafis sederhana menjadi sangat kompleks dan mutakhir. Permainan *onlin*e (*online game*) adalah jenis permainan video atau permainan komputer dengan menggunakan jaringan internet sebagai media interaksinya. Permainan *online* terdiri dari dua unsur utama, yaitu *server* dan *client*. *Server* adalah penyedia layanan gaming yang merupakan basis agar *client –client* yang terhubung dapat memainkan permainan dapat melakukan komunikasi dengan baik. Suatu *server* pada prinsipnya hanya melakukan administrasi permainan dan menghubungkan *client-client*. Sedangkan *client* adalah pengguna permainan dan memakai kemampuan *server*. Contoh permainan online adalah *Ragnarok Online, Risk Your Life, Xian*.

2.2.3 Konsol Permainan (*Game Console*)

Konsol permainan adalah sebuah mesin elektronik yang dirancang khusus untuk memainkan permainan video. Perangkat keluarannya biasa berupa monitor komputer atau televisi dan alat masukan utamanya berupa sebuah pengendali. Konsol permainan yang pertama kali dibuat adalah *Atari*, kemudian dilanjutkan dengan *Nintendo* yang sukses pada tahun 1985-1989. Konsol permainan modern

sekarang ini antara lain adalah *Playstation* buatan perusahaan *Sony*. Untuk *Console* yang berbentuk kecil dan dapat dibawah kemana-mana biasa disebut *Portable Console* .

2.2.4 Klasifikasi Permainan

Sejarah *game* tidak lepas dengan namanya *platform*, dari yang awalnya hanya mesin *Arcade* (Mesin ding dong) hingga sampai sekarang ini yang sudah mencapai era *Mobile Games*. Untuk lebih jelasnya, berikut ini adalah klasifikasi *game* berdasarkan jenis *Platform* yang di gunakan :

1. *Arcade Games* yaitu yang sering disebut ding-dong di Indonesia, biasanya berada di daerah / tempat khusus dan memiliki box atau mesin yang memang khusus didesain untuk jenis video games tertentu dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya lebih merasa "masuk" dan menikmati. Seperti pistol, kursi khusus, sensor gerakan, sensor injakkan dan stir mobil (beserta transmisinya).
2. *PC Games* (*personal computer*) yaitu video game yang dimainkan menggunakan *Personal Computers*.
3. *Console Games* yaitu video game yang dimainkan menggunakan *console* tertentu seperti *playstation XBOX, Nitendo,* dan lan-lain.
4. *Handheld Game* yaitu *game* yang dimainkan di console khusus video game yang dapat dibawa kemana-mana, contoh *nitendo DS dan Sony PSP*.
5. *Mobile Games* yaitu game yang biasa dimainkan khusus untuk *handphone atau PDA.*

Klasifikasi permainan berdasarkan pembatasan usia dari pemainnya, menurut ***ESRB*** (***Entertainment Software Rating Board***) adalah sebagai berikut [3]:

1. *Early Childhood (3+)* memiliki isi yang mungkin cocok untuk usia 3 keatas. Tidak mengandung materi yang tidak pantas seperti adegan kekerasan.
2. *Everyone (6+)* memiliki konten yang mungkin cocok untuk usia 6 keatas. Judul-judul dalam kategori ini mungkin berisi sedikit kartun, fantasi atau kekerasan ringan dan tidak jarang menggunakan bahasa yang ringan.

3. *Everyone (10+)* memiliki konten yang mungkin cocok untuk usia 10 keatas. Judul-judul dalam kategori ini mungkin berisi lebih banyak kartun, fantasi atau kekerasan ringan, bahasa ringan dan juga minimal tema sugestif.
4. *Teen (13+)* memiliki konten yang mungkin cocok untuk usia 13 keatas. Judul-judul dalam kategori ini mungkin mengandung kekerasan, tema sugestif, humor kasar, sedikit darah, simulasi perjudian, dan tidak jarang menggunakan bahasa yang kuat.
5. *Mature (17+)* memiliki konten yang mungkin cocok untuk orang usia 17 dan lebih tua. Judul-judul dalam kategori ini mungkin berisi intens kekerasan, darah, konten seksual dan bahasa yang kuat. *Mature* (17+) memiliki konten yang mungkin cocok untuk orang usia 17 dan lebih tua. Judul judul dalam kategori ini mungkin berisi intens kekerasan, darah, konten seksual dan bahasa yang kuat.
6. *Adults Only (18+)* memiliki konten yang hanya boleh dimainkan oleh orang-orang 18 tahun keatas. Judul-judul dalam kategori ini mungkin termasuk adegan yang intens, kekerasan dan juga mengandung konten seksual.
7. *RP* (*Rating Pending*) telah disampaikan kepada ESRB dan sedang menunggu penilaian akhir. (Simbol ini hanya muncul dalam iklan sebelum permainan dibebaskan).

2.2.5 Sudut Pandang Permainan (*Game View Point*)

Sebuah *game* biasanya mempunyai sudut pandang permainan tersendiri disesuaikan berdasarkan *genre game* yang diambil. Berikut beberapa macam sudut pandang permainan yang biasa digunakan [3]:

1. *Side Scrolling*

Adalah sudut pandang permainan yang terlihat dari samping dan memungkinkan karakter utama untuk bergerak dari kiri ke kanan serta memungkinkan background pada *game* seolah-olah bergeser mengikuti pergerakan karakter utama.

2. *Top Down*

Adalah sudut pandang permainan yang memungkinkan karakter utama bermanuver ke empat arah namun cara permainannya sendiri bergeser dari bawah ke atas, dan biasanya *game* yang menggunakan sudut pandang permainan jenis ini adalah *shooter game*.

3. *Isometric*

Adalah sudut pandang permainan yang memungkinkan permainan terlihat diantara sisi *Side Scrolling* dan juga *Top Down*, dan biasanya diterapkan pada *game* dengan *genre RTS* (*Real Time Strategy*).

2.2.6 Jenis Permainan (*Genre Game*)

Game dari tahun-tahun semakin banyak saja jenisnya, teknologi yang dipakaipun semakin canggih. Dari ratusan atau bahkan ribuan *game* yang telah bermunculan di dunia, *game-game* tersebut dikelompokkan menjadi beberapa *genre*. Pada dasarnya, *genre-genre game* yang beredar di pasaran dapat dikelompokkan sebagai berikut [3]:

a. *Advergames*

Permainan yang dikembangkan untuk tujuan periklanan.

b. *Action Game*

Adalah tipe *game* dengan fitur utama berupa banyaknya aksi di mana pemain harus memiliki keterampilan reaksi yang cepat untuk menghindari musuh atau menghindari rintangan. Pengembang game tipe ini perlu memastikan *game* yang dibuat dioptimasi sehingga pemain memiliki pengalaman bermain yang baik, yang tidak terganggu oleh *delay* proses yang lama, contohnya : *Metal Gear Solid*.

c. *Adventure Game*

Adalah tipe *game* yang umumnya membuat pemain harus berjalan mengelilingi suatu tempat yang terkondisi, seperti sebuah istana, gua yang berkelok, dan planet yang jauh. Pemain melakukan navigasi suatu area mencari pesan-pesan rahasia, memperoleh obyek yang memiliki kemampuan yang bervariasi, bertempur dengan musuh, dan lain-lain.

Untuk membuat *game* ini, diperlukan perencanaan yang akurat sehingga memiliki alur cerita yang menarik bagi pemain, contohnya : *Tomb Raider, Prince of Persia, Hitman*, dan lain-lain.

d. *Fighting Game*

Adalah tipe *game* yang intinya harus menjatuhkan lawan tandingnya, entah itu dengan pukulan, tendangan, combo, maupun dengan jurus special, contohnya : *Tekken, Mortal Kombat, Naruto*, dan lain-lain.

e. *FPS (First Person Shooter)*

Adalah tipe *game* yang mengunakan sudut pandang orang pertama untuk membidik atau membunuh musuh, sehingga kita hanya melihat tangannya saja dan tidak melihat tubuh karakter yang dimainkan, contohnya : *Call of Duty, Half Life, Crysis*, dan lain-lain.

f. *RPG (Role Playing Game)*

Adalah tipe *game* yang seringkali berupa multi karakter *game* di mana setiap pemain memiliki karakter dengan kemampuan, kekuatan, dan kelemahan yang spesifik. Para pemain saling berkompetisi, berinteraksi, dan bertempur satu sama lain. Tampilan grafis yang khas untuk setiap karakter pemain ditambah dengan *storyline* yang mendebarkan akan sangat menarik dan memberikan pengalaman yang berbeda di dalam bermain, contohnya : *Final Fantasy, Warcraft*, dan lain-lain.

g. *Platform Game*

Adalah tipe *game* yang mengharuskan pemain mengarahkan suatu obyek dengan melalui berbagai tahap atau tingkatan area untuk menyerang musuh dan menghindar terhadap serangan. Tipe *game* ini sedikit serupa dengan *action game*, tetapi aksinya tidak secepat *action game*. Teknik *collision detection* sangat sering dimanfaatkan pada tipe *game* ini, contohnya : *Sonic The Hedgehog, Mario Bros, Metal Slug*.

h. *Puzzle Game*

Adalah tipe *game* yang umumnya membuat pemain menggunakan kemampuan berpikirnya sebagai pengganti keterampilan reaksi yang cepat karena terdapat rahasia yang perlu dipecahkan. *Game* ini lebih bersifat statis dibanding *action game*. Pembuatan *game* tipe ini seringkali ditunjang dengan AI (*Artificial Intelligence*), contohnya : *7 Wonders of The Ancient World.*

i. *Racing Game*

Adalah tipe *game* yang tujuannya adalah mencapai garis finish dari suatu *race*, dalam *game* ini biasanya pemain dapat memilih dan membeli kendaraan, mendandani, melakukan upgrade mesin , contohnya : *Need For Speed.*

j. *Simulation Game*

Adalah tipe *game* yang memberikan pengalaman atau interaksi sedekat mungkin dengan keadaan yang aslinya, meskipun terkadang keadaan tersebut masih eksperimen atau bahkan fiktif, tapi ada penekanan khusus

pada detail dan pengalaman realistik menggunakan keadaan tersebut, contohnya : *Gran Turismo.*

k. *Sandbox Game*

Adalah tipe *game* yang umumnya ditujukan untuk menjelajahi suatu kota atau tempat dan bebas berinteraksi dengan objek disekitarnya. Mungkin *game* ini bisa di bilang mirip *RPG* tapi *Sandbox* sangat berbeda dengan *RPGgame*, perbedaannya adalah tidak adanya peningkatan level karakternya, contohnya : *GTA dan Bully*.

l. *RTS (Real Time Strategy)*

Adalah tipe *game* yang mengharuskan pemainnya menggunakan taktik dan strategi untuk jeli dalam melihat setiap peluang, kelemahan musuh dan bijaksana dalam menggunakan sumber daya yang ada, contohnya : *Command and Conquer*.

m. *Sport Game*

Adalah tipe *game* yang berupa kompetisi antara dua pemain atau lebih, di mana pemain dapat berupa individual atau tim. Contoh *game* tipe ini antara lain sepakbola, bola basket, tenis, dan bilyard. Tergantung seberapa cepat permainan yang terjadi, aplikasi *game* perlu dioptimalkan, contohnya : *Pro Evolution Soccer.*

2.3 Tools yang digunakan

Tools adalah unsur penting dalam pembangunan sebuah aplikasi.

2.3.1 OOP (Object Oriented Programming)

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melaluipendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi onjek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek.

Pada saat ini, metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekatan terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan saat ini sangat beragam (aplikasi bisnis, real-time, utility, dan sebagainya) dengan *platform* yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut[5].

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut[5]:

a. Meningkatkan produktivitas

   Karena kelas dan objek yang ditemukan dalam suatu masalahmasih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reusable).*

b. Kecepatan pengembangan

   Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.

c. Kemudahan pemeliharaan

   Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering diubah0ubah.

d. Adanya konsistensi

   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

e. Meningkatkan kualitas perangkat lunak

   Karena adanya pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengambangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

Berikut beberapa contoh bahasa pemrograman yang mendukung pemrograman berorientasi objek[5] :

a. Smalltalk

   Smalltalk merupakan salah satu bahasa pemrogramanyang dikembangkan untuk mendukung pemrograman berorientasi objek mulai tahun 1978.

b. Bahasa Pemrograman Eiffel

   Eiffel merupakan bahasa pemrograman yang dikembangkan untuk mendukung pemrograman berorientasi objek mulai tahun 1985 oleh Bertrand Meyer dan *compiler* Eiffel selesai pada tahun 1987.

c. Bahasa Pemrograman C++

   C++ merupakan pengembangan lebih lanjut bahasa pemrograman C untuk mendukung pemrograman berorientasi objek.

d. Bahasa Pemrograman (web) PHP

   PHP dibuat pertama kali oleh seorang perekayasa perangkat (*software engineering)* yang bernama Rasmus Lerdoff.

e. Bahasa Pemrograman Java

   Java dikembangkan oleh perusahaan Sun Microsystem. Java menurut definisi dari Sun Microsystem adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan.

2.3.1.1 Konsep Dasar Berorientasi Objek

Berikut adalah konsep dasar Permrograman Berorientasi Objek[5] :

a. Objek (*Object*)

   Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur,status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi(status) dan mempunyai operasi yang dapat diterapkan atau dapat berpengaruh pada status objeknya.

b. Kelas (*class*)

   Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dalam kelas tersebut .

c. Pembungkusan (*Encapsulation*)

   Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

d. Pewarisan (*Inheritance*) dan Generalisasi/Spesialisasi

   Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dirinya .

e. Metode

   Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural .

f. *Polimorfisme*

   Kemampuan suatu objek untuk digunakan dibanyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program .

2.3.2 UML (Unified Modeling Language)

Unified Modelling Language (UML) adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG. UML terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu *Diagram Interchange Specification, UML, Infrastructure, UML Superstructure, dan Object Constraint Language (OCL).* Pada UML 2.3 terdisi 13 macam diagram yang dikelompokkan pada 3 kategori, yaitu :

A. Structure Diagram, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

1. Diagram Kelas

   Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut attribut dan metode atau operasi.

2. Diagram Objek

   Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem.

3. Diagram Komponen

   Diagram komponen dibuat untuk menunjukan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem.

4. *Composite Structure Diagram*

   Composite structure diagram baru mulai ada pada UML versi 2.0. Diagram ini dapat digunakan untuk menggambarkan struktur dari bagian-bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan (*runtime*).

5. *Package Diagram*

   Package diagram menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML. Hampir semua diagram dalam UML dapat dikelompokkan menggunakan *package diagram.*

6. *Deployment Diagram*

   Deployment menunjukan konfigurasi komponen dalam proses eksekusi aplikasi.

B. *Behavior Diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

1. *Use Case Diagram*

   Use case diagram 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.

2. *Activity Diagram*

   Activity diagram menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

3. *State Machine Diagram*

   State machine diagram digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek.

C. *Interactions Diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi antar subsistem pada suatu sistem.

1. *Sequence Diagram*

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

2. *Communication Diagram*

   Communication Diagram menggambarkan interaksi antar objek/bagian dalam bentuk urutan pengiriman pesan. Diagram komunikasi merepresentasikan informasi yang diperoleh dari diagram kelas, diagram sekuen, dan diagram *use case* untuk mendeskripsikan gabungan antara struktur statis dan tingkah laku dinamis dari suatu sistem.

3. *Timing Diagram*

   Timing diagram merupakan diagram yang fokus pada penggambaran terkait batasan waktu.

4. *Interaction Overview Diagram*

   Interaction overview diagram mirip dengan diagram aktivitas yang berfungsi untuk menggambarkan sekumpulan urutan aktivitas, diagram ini adalah bentuk aktivitas diagram yang setiap titik merepresentasikan diagram interaksi.

2.3.3 Adobe Flash CS 6

Adobe Flash merupakan sebuah program yang didesain khusus oleh Adobe dan program aplikasi standar *authoring tool professional* yang digunakan untuk membuat animasi dan bitmap yang sangat menarik untuk keperluan pembangunan situs web yang interaktif dan dinamis. Flash didesain dengan kemampuan untuk membuat animasi 2 dimensi yang handal dan ringan sehingga flash banyak digunakan untuk membangun dan memberikan efek animasi pada website, CD Interaktif dan yang lainnya. Selain itu aplikasi ini juga dapat digunakan untuk membuat animasi logo, *movie*, *game*, pembuatan navigasi pada situs web, tombol animasi, banner, menu interaktif, interaktif form isian, *e-card*, *screen saver* dan pembuatan aplikasi-aplikasi web lainnya. Dalam Flash, terdapat teknik-teknik membuat animasi, fasilitas *action*
script, *filter*, *custom easing* dan dapat memasukkan *video* lengkap dengan fasilitas *playback FLV*. Keunggulan yang dimiliki oleh Flash ini adalah ia mampu diberikan sedikit code pemograman baik yang berjalan sendiri untuk mengatur animasi yang ada didalamnya atau digunakan untuk berkomunikasi dengan program lain seperti HTML, PHP, dan Database dengan pendekatan XML, dapat dikolaborasikan dengan web, karena mempunyai keunggulan antara lain kecil
dalam ukuran file outputnya. Movie-movie Flash memiliki ukuran *file* yang kecil dan dapat ditampilkan dengan ukuran layar yang dapat disesuaikan dengan keingginan. Aplikasi Flash merupakan sebuah standar aplikasi industri perancangan animasi web dengan peningkatan pengaturan dan perluasan kemampuan integrasi yang lebih baik. Banyak fitur-fitur baru dalam Flash yang dapat meningkatkan kreativitas dalam pembuatan isi media yang kaya dengan memanfaatkan kemampuan aplikasi tersebut secara maksimal. Fitur-fitur baru ini

membantu peneliti menjadi lebih memusatkan perhatian pada desain yang dibuat secara cepat, bukannya memusatkan pada cara kerja dan penggunaan aplikasi tersebut. Flash juga dapat digunakan untuk mengembangkan secara cepat aplikasiaplikasi

website yang kaya dengan pembuatan *script* tingkat lanjut. Di dalam aplikasinya juga tersedia sebuah alat untuk men-*debug script*, dengan menggunakan *code hint* untuk mempermudah dan mempercepat pembuatan dan pengembangan isi *Action Script* secara otomatis.

2.3.4 Action Script 3.0

ActionScript merupakan bahasa *scripting* yang terdapat di gunakan Flash yang bertujuan untuk mempermudah pembangunan suatu aplikasi atau animasi. Biasanya semakin kompleks animasi pada Flash, maka akan semakin banyak memakan frame. Dengan *ActionScript*, penggunaan frame tersebut dapat dikurangi, bahkan dapat membuat animasi yang kompleks hanya dengan satu frame saja.

ActionScript awalnya berasal dari *ActionScript* 1.0 yang dirilis pada tahun 2000 di Macromedia Flash 5. Bahasa scripting ini berisi semua kode dan perintah lainnya yang berbasis web pengembang bahasa, namun kecepatan dan kekuatannya sangat pendek. Selanjutnya *ActionScript* 2.0, versi ini digunakan pada Macromedia Flash MX 2004 hingga Macromedia Flash 8. Kelebihannya ialah memiliki kemampuan *compile time checking*, *strict-typing*, dan *class-based syntax*. *ActionScript* 3.0 baru mulai digunakan pada Adobe Flash CS3 yang merupakan restrukturisasi fundamental dari model pemrograman sebelumnya. Banyak kelebihan dari AS3 seperti penggunaannya yang luas terutama dalam pengembangan *Rich Internet Application* (RIA) dan mendukung pemograman berorientasi objek.

Berikut merupakan pembaharuan dari *ActionScrpit* 3.0, yaitu :

1. Dapat memeriksa jenis-jenis informasi yang ada di *compile time* dan *runtime* prototipe.

2. Peningkatan kinerja dari sistem pewarisan berbasis kelas yang terpisah dari sistem pewarisan berbasis prototipe.
3. Dukungan untuk paket-paket, *namespaces*, dan ekspresi reguler.
4. Mengkompilasi ke tipe baru sepenuhnya dari *bytecode*, tidak kompatibel dengan *Actionscript* 1.0 dan 2.0 *bytecode*.
5. Revisi *Flash Player* API disusun menjadi paket.
6. Integrasi *ECMAScript* untuk XML (E4X) untuk keperluan pengolahan XML.

Dapat dibilang AS3 merupakan Full OOP sedangkan AS2 masih *procedural* & struktural biasa. AS3 ini dirancang untuk beberapa tujuan, yaitu sebagai berikut:

1. Keamanan, bahasa ini mendukung keamanan pengetikan sehingga para pengembang dapat menulis jelas (tidak ambigu) mudah *maintainable code*.
2. Kesederhanaan, bahasa ini cukup intuitif bagi pengembang untuk bisa membaca dan menulis program tanpa terus-menerus berkonsultasi dengan referensi manual.
3. Kinerja, bahasa memungkinkan pengembang untuk menulis program yang kompleks dengan efisien dan responsif.
4. Kompatibilitas, bahasa yang pendek menyediakan kompatibilitas ke belakang dan ke depan jalan dan tumpang tindih yang signifikan dengan standar industri.

2.3.5 Adobe Photoshop

Adobe Photoshop adalah *software* yang dibuat oleh perusahaan *Adobe System*, yang di khususkan untuk pengeditan foto atau gambar dan pembuatan effect. Perangkat lunak ini banyak digunakan oleh Fotografer Digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (*market leader*) untuk perangkat lunak pengolah gambar. Meskipun pada awalnya *Photoshop* dirancang untuk menyunting gambar untuk cetakan berbasis-kertas, *Photoshop* yang ada saat ini juga dapat digunakan untuk memproduksi gambar untuk *World*

Wide Web. Beberapa versi terakhir juga menyertakan aplikasi tambahan, *Adobe ImageReady*, untuk keperluan tersebut.

Photoshop juga memiliki hubungan erat dengan beberapa perangkat lunak penyunting media, animasi, dan *authoring* buatan-*Adobe* lainnya. File format asli *Photoshop*, PSD, dapat diekspor ke dan dari *Adobe ImageReady*. *Adobe Illustrator, Adobe Premiere Pro, After Effects dan Adobe Encore DVD* untuk membuat DVD profesional, menyediakan penyuntingan gambar non-linear dan layanan special *effect* seperti *background*, tekstur, dan lain-lain untuk keperluan televisi, film, dan situs web [6].