BAB 2
LANDASAN TEORI

2.1 Algoritma Logika Fuzzy

Logika *fuzzy* merupakan salah satu komponen pembentuk *soft computing*. Logika fuzzy pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. Dasar logika *fuzzy* adalah teori himpunan *fuzzy*. Pada teori himpunan *fuzzy*, peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting. Nilai keanggotaan atau derajat keanggotaan atau *membership function* menjadi ciri utama dari penalaran dengan logika *fuzzy* tersebut. Menurut Cox (1994) ada beberapa alasan mengapa orang menggunakan logika *fuzzy* antara lain [1].

1. Konsep logika *fuzzy* mudah dimengerti, karena logika *fuzzy* menggunakan dasar teori himpunan, maka konsep matematis yang mendasari penalaran fuzzy tersebut cukup mudah untuk dimengerti
2. Logika *fuzzy* sangat fleksibel, artinya mampu beradaptasi dengan perubahan-perubahan, dan ketidakpastian yang menyertai permasalahannya.
3. Logika *fuzzy* memiliki toleransi terhadap data yang tidak tepat. Jika diberikan sekelompok data yang cukup homogen, dan kemudian ada beberapa data yang "eksklusif", maka logika *fuzzy* memiliki kemampuan untuk menangani data eksklusif tersebut.
4. Logika *fuzzy* mampu memodelkan fungsi nonlinear yang sangat kompleks.
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman pakar secara langsung tanpa harus melalui proses pelatihan. Dalam hal ini, sering dikenal dengan nama *fuzzy expert system* menjadi bagian terpenting
6. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional. Hal ini umumnya terjadi pada aplikasi-aplikasi di bidang teknik mesin maupun teknik elektro.

7. Logika *fuzzy* didasarkan pada bahasa alami. Logika *fuzzy* menggunakan bahasa sehari-hari sehingga mudah dimengerti

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy*, yaitu :

a. Variabel *fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy. Contoh yang diterapkan pada penelitian ini adalah kondisi nafsu makan, Kesehatan, dan Kebersihan, dan lain-lain.
b. Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A, yang sering ditulis dengan $\mu_A(x)$, memiliki dua kemungkinan, yaitu:
    1. Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
    2. Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

    Contoh, Jika diketahui:

    S = {1, 2, 3, 4, 5, 6} adalah semesta pembicaraa

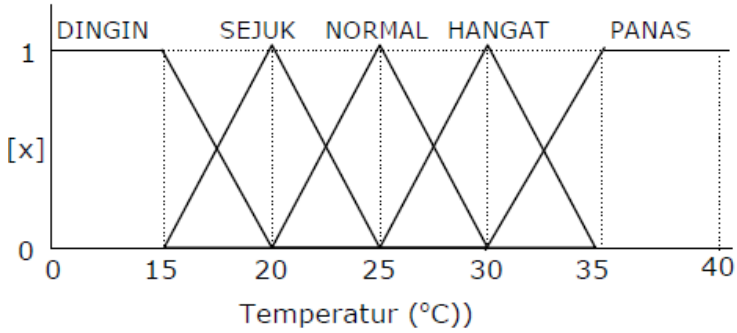
    A = {1, 2, 3} dan B = {3, 4, 5}

    Bisa dikatakan bahwa:

    a. Nilai keanggotaan 2 pada himpunan A, μA[2]= 1, karena 2 ∈ A.
    b. Nilai keanggotaan 3 pada himpunan A, μA[3]= 1, karena 3 ∈ A.
    c. Nilai keanggotaan 4 pada himpunan A, μA[4]= 0, karena 4 ∉ A.
    d. Nilai keanggotaan 2 pada himpunan B, μA[2]= 0, karena 2 ∉ B.
    e. Nilai keanggotaan 3 pada himpunan B, μA[3]= 1, karena 3 ∈ A.

    Contoh:

    a. Variabel Lapar, terbagi menjadi dua himpunan *fuzzy*, yaitu: Lapar (rendah), dan kenyang (tinggi).
    b. Variabel temperatur, terbagi menjadi 5 himpunan *fuzzy*, yaitu: Dingin, sejuk, normal, hangat dan panas. Bisa dilihat pada gambar 2.1.



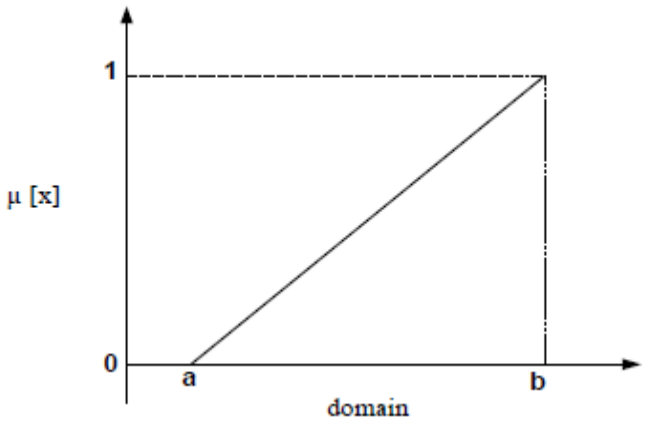
Gambar 2.1 Himpunan *Fuzzy* Pada Variabel Temperatur [1]

c. Semesta Pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan *real* yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya. Contoh semesta pembicara pada penerapan penelitian ini adalah parameter kalori dengan semesta pembicara 0-100.

d. Fungsi Keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik input data kedalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang digunakan, diantaranya adalah:

1. Representasi Linear

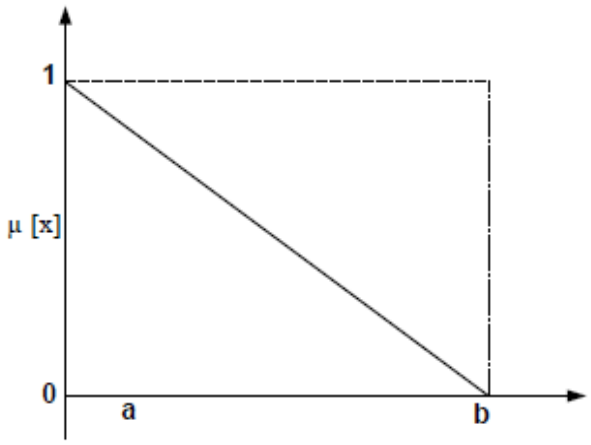
Pada representasi linear, pemetaan *input* derajat keanggotaan yang digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Terdapat dua keadaan himpunan *fuzzy* yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat nol (0) bergerak ke kanan menuju ke nilai dimain yang memiliki derajat keanggotaan lebih tinggi. Dapat dilihat pada gambar 2.2. dan gambar 2.3.



Gambar 2.2. Kurva Linear Naik [1]

Fungsi keanggotaan:

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



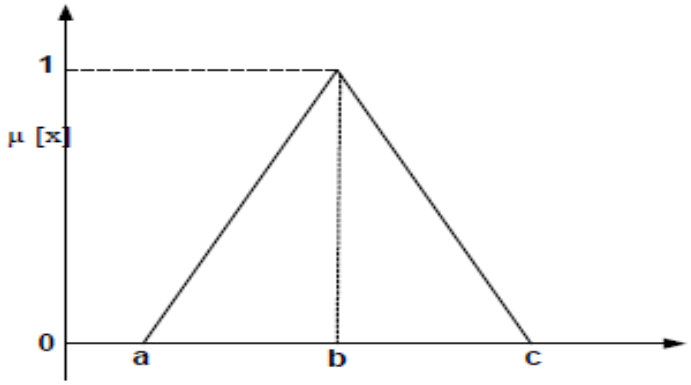
Gambar 2.3. Kurva Linear Turun [1]

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} \frac{b-x}{b-a}; a \leq x \leq b \\ 0; x \geq b \end{cases}$$

2. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara dua garis linear. Seperti dapat dilihat pada gambar 2.4.



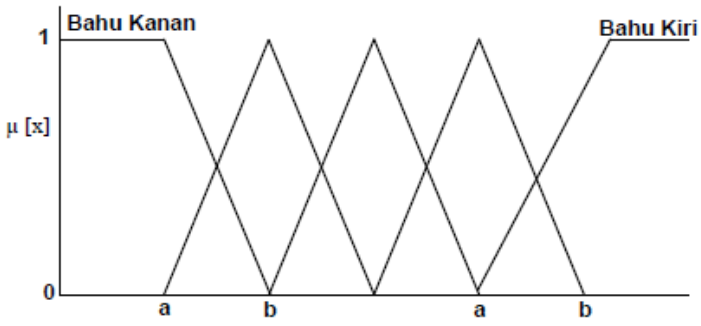
Gambar 2.4. Kurva Segitiga [1]

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; x \leq a \; atau \; x \geq c \\ \frac{x-a}{b-a}; \quad a \leq x \leq b \\ \frac{b-x}{c-b}; \; b \leq x \leq c \end{cases}$$

3. Representasi Kurva Bahu

Representasi kurva bahu digunakan untuk mengakhiri variabel suatu daerah variabel fuzzy. untuk lebih jelasnya dapat dilihat pada Gambar 2.5.



Gambar 2.5. Kurva Bahu [1]

Fungsi keanggotaan Kurva Bahu Kiri:

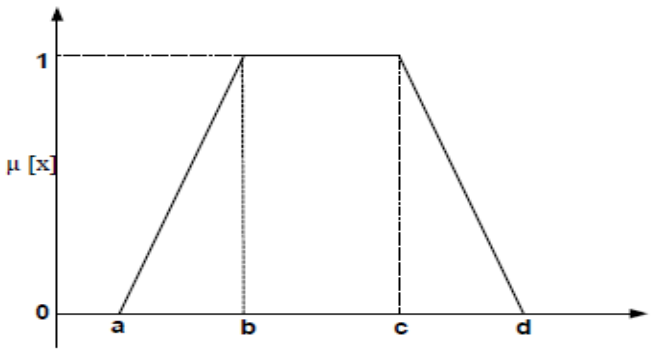
$$\mu[x] = \begin{cases} 1; x \leq a \\ \frac{(b-x)}{(b-a)}; \quad a \leq x \leq b \\ 0; x \geq b \end{cases}$$

Fungsi keanggotaan Kurva Bahu Kanan:

$$\mu[x] = \begin{cases} 0; x \leq a \\ \frac{(x-a)}{(b-a)}; \quad a \leq x \leq b \\ 1; x \geq b \end{cases}$$

4. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai kenggotaan 1. Dapat dlihat pada gambar 2.6.



Gambar 2.6. Kurva Trapesium [1]

FungsiKeanggotaan

$$\mu[x] = \begin{cases} 0; x \leq a \; atau \; x \geq d \\ \frac{x-a}{b-a} \\ 1; b \leq x \leq c \qquad ; a \leq x \leq b \\ \frac{d-x}{d-c}; x \geq d \end{cases}$$

e. Sistem Inferensi Fuzzy

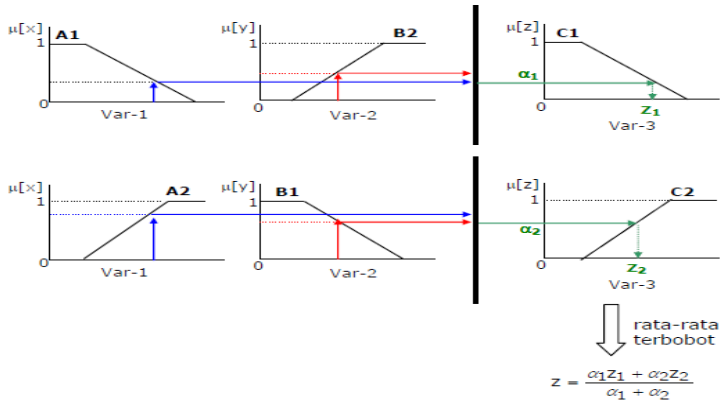
Inferensi *fuzzy* merupakan kerangka komputasi yang didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* berbentuk IF-THEN, dan penalaran *fuzzy*. Inferensi *fuzzy* telah berhasil diterapkan di bidang-bidang seperti kontrol otomatis, klasifikasi data, analisis keputusan, dan sistem pakar. Sehingga dari penerapan yang ada dikenal beberapa istilah lain dalam inferensi *fuzzy* yaitu *fuzzy rule based.*

Dalam inferensi *fuzzy* ada beberapa komponen utama yang dibutuhkan. Komponen tersebut meliputi data variabel *input*, data variabel *output*, dan data aturan. Untuk mengolah data variabel *input* dibutuhkab beberapa fungsi meliputi fungsi *fuzzifikasi* yang terbagi dua, yaitu fungsi untuk menentukan nilai jenis keanggotaan suatu himpunan dan fungsi penggunaan operator.

Sistem inferensi *fuzzy* dari sumber buku yang ditulis oleh Kusumadewi, terdapat tiga metode, yaitu:

1. Metode Tsukamoto

Pada Metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk IF-Then harus direpresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, output hasil inferensi dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan α-predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot. Dapat dilihat pada gambar 2.7.



Gamar 2.7. Inferensi Menggunakan Metode Tsukamoto [1]

Agregasi untuk menentukan keluaran semua *rules* dan dikombinasikan menjadi sebuah *fuzzy set* tunggal. *Defuzzifikasi* untuk menghasilkan nilai kesehatan menggunakan rata-rata terbobot.

$$z = \frac{\alpha 1 z 1 + \alpha 2 z 2 \ldots + \alpha n z n}{\alpha 1 + \alpha 2 \ldots + \alpha n}$$

1. z = nilai crisp
2. $\alpha 1$ = derajat keanggotaan masing-masing variabel
3. $z1$ = nilai domain pada variabel linguistik z.

2. Metode Mamdani

Metode Mamdani sering juga dikenal dengan nama Metode Max-Min. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan *output*, diperlukan 4 tahapan, yaitu Pembentukan himpunan fuzzy, Aplikasi fungsi implikasi (aturan), Komposisi aturan, dan Penegasan (*deffuzy*).

3. Metode Sugeno

Penalaran dengan metode Sugeno hampir sama dengan penalaran MAMDANI, hanya saja output (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985.

a. Model *fuzzy* Sugeno Orde-Nol

Secara umum bentuk Model *fuzzy* Sugeno Orde-Nol adalah : **IF (X_1 is A_1) ● (X_2 is A_2) ● (X_3 is A_3) ●....● (X_N is A_N) THEN z=k**

Dengan A_1 adalah himpunan *fuzzy* ke-i sebagai antaseden, dan k adalah sebagai konstanta (tegas) sebagai konsekuen.

b. Model *fuzzy* Sugeno Orde-Satu

Secara umum bentuk Model *fuzzy* Sugeno Orde-Satu adalah : **IF (X_1 is A_1) ● ... ● (X_N is A_N) THEN z = p_1 * x_1 +...+ p_N*p_N + q**

Dengan A_1 adalah himpunan *fuzzy* ke-i sebagai antaseden, dan p_1 adalah sebagai konstanta (tegas) ke-i dan q juga merupakan konstanta sebagai konsekuen.

2.2 Kecerdasan Buatan

Kecerdasan Buatan atau *Artificial Intelligence* merupakan salah satu bagian dari ilmu komputer yang membuat agar mesin komputer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Makin pesatnya perkembangan teknologi menyebabkan adanya perkembangan dan perluasan lingkup yang membutuhkan kehadiran kecerdasan buatan. Adanya irisan penggunaan kecerdasan buatan di berbagai disiplin ilmu tersebut menyebabkan cukup rumitnya untuk mengklasifikasikan kecerdasan buatan menurut disiplin ilmu yang menggunakannya, untuk memudahkan hal tersebut maka untuk pengklasifikasian lingkup kecerdasan buatan didasarkan pada *output* yang diberikan yaitu [3]:

1. Sistem pakar (*expert system*) disini computer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian komputer akan memiliki keahlian untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar.
2. Pengolahan Bahasa Alami (*Natural Language Processing*) dengan pengolahan bahasa alami ini diharapkan user dapat berkomubikasi dengan computer dengan menggunakan bahasa sehari-hari.
3. Pengenalan ucapan (*Speech Recognition*) melalui pengenalan ucapan diharapkan manusia dapat berkomunikasi dengan computer dengan menggunakan suara.
4. Robotika dan sistem sensor (*Robotics & Sencory Systems*)
5. *Computer Vision*, mencoba untuk dapat mengintrepertasikan gambar atau objek-objek tampak melalui komputer.
6. *Intelligent Computer-aided Instruction Computer* dapat digunakan sebagai tutor yang dapat melatih dan mengajar.
7. *Game playing*.

2.3 Game

Game atau permainan merupakan media hiburan yang sudah dikenal sejak dahulu. *Game* dapat dimainkan berbagai umur tua maupun muda. *Game* juga sudah berkembang dengan pesat sesuai dengan perkembangan teknologi, dari *game* sederhana sampai *game* modern saat ini. Hal ini terbukti dengan adanya perkembangan jenis, produk, serta alat yang digunakan.

2.3.1 Pengertian Game

Game merupakan sebuah bentuk seni dimana penggunanya disebut dengan pemain (*player*), diharuskan membuat keputusan-keputusan dengan tujuan mengelola sumber daya yang diperoleh dari kesempatan-kesempatan bermain (*token*) untuk mencapai tujuan tertentu. *Video game* adalah bentuk *game* yang interaksi umumnya melibatkan media *video* dan *audio*.

Menurut Andang Ismail terdapat dua pengertian *game* (permainan). Pertama, *game* (permainan) adalah sebuah aktifitas bermain yang murni mencari kesenangan tanpa mencari menang atau kalah. Kedua, permainan diartikan sebagai aktifitas bermain yang dilakukan dalam rangka mencari kesenangan dan kepuasan, namun ditandai pencarian menang-kalah.

Berdasarkan representasinya, *game* dapat dibedakan menjadi 2 jenis yaitu *game* 2 dimensi (2D) dan 3 dimensi (3D). *Game* 2D adalah *game* yang secara matematis hanya melibatkan 2 elemen koordinat kartesius yaitu x dan y, sehingga konsep kamera pada *game* 2D hanya menentukan gambar pada *game* yang dapat dilihat oleh pemain. Sedangkan *game* 3D adalah *game* yang selain melibatkan elemen x dan y juga melibatkan elemen z pada perhitungannya sehingga konsep kamera pada *game* 3D benar-benar menyerupai konsep kamera pada kehidupan nyata [7].

2.3.2 Simulsi Game

Simulasi *game* adalah *game* yang mengendalikan suatu objek yang bergerak/tidak bergerak baik secara langsung maupun tidak langsung dengan tujuan melakukan hal-hal tertentu. *Game* ini sangat menarik karena bisa membuat pemain berfantasi seakan objek yang berada dalam *game* tersebut adalah nyata dan milik pemain [7]. Beberapa jenis *game simulation* yaitu:

1. *Assets Simulation*

   Game simulasi yang menuntut kita untuk membuat / membangun / mengelola suatu objek / asset / bangunan / instalasi yang dikendalikan secara personal untuk mencapai tujuan tertentu. Contoh *game* : *game* SIM / Tycoon series

2. *Action Simulation*

   Game simulasi plus action yang menuntut kita untuk mengendalikan suatu objek tertentu secara langsung untuk melakukan suatu hal secara aktif, misalnya mengendalikan kendaraan perang untuk bertempur. Contoh *Game* : *Game* Pesawat Tempur, dan sebagainya.

3. *Vehicle Simulation*

   Game simulasi realitas yang tata cara permainan mengambil dari objek aslinya, dengan tujuan mengajari player bagaimana cara mengoprasikan suatu kendaraan yang tidak biasa dikendarai orang awam dan rumit untuk dioperasikan Contoh *Game* : Seri Flight Simulator, Rail Simulator & Ship Simulator.

4. *Music simulation*

   Game simulasi yang menuntut kita mengendalikan suatu objek secara tidak langsung untuk melakukan hal-hal yang berhubungan dengan musik. Contoh *game* : Seri Guitar Hero, seri Dance-Dance Revolution

5. *Lovesims/visual novel*

   Game ini hanya berupa suatu percakapan antar tokoh disertai dengan gambar CG (*Computer Graphic*) Anime yang indah. Jalan ceritanya berjalan sendiri tanpa bisa kita kendalikan, kita hanya bisa mengatur jalan cerita melalui pilihan langkah sang tokoh di *game* tersebut untuk mencapai akhir

yang buruk atau bahagia. *Game* ini hanya dibuat di negara asal anime yaitu jepang dan lumayan populer di sana namun tidak begitu di negara lainnya. *Game* ini kurang mendapat perhatian karena gameplaynya yang dinilai kurang berbobot, selain itu game tipe ini yang beredar di Indonesia kebanyakan *game* adult yang tidak baik untuk *gamer* usia dibawah umur.

2.4 Tools yang Digunakan

Tools yang digunakan meliputi perancangan berbasis objek, UML, dan Unity.

2.4.1 Perancangan Berbasis Objek

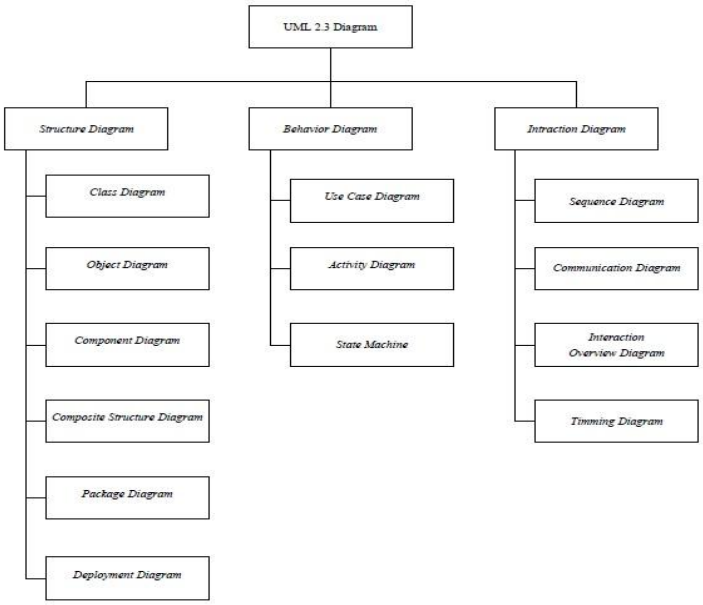
Perancangan berorientasi objek adalah strategi perancangan di mana perancangan sistem memikirkan „benda‟ dan bukan operasi atau fungsi. Proses perancangan berorientasi objek berarti bahwa kita mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilaku. Beberapa cara untuk menentukan karakteristik dalam pendekatan berorientasi objek secara umum mencakup empat hal, yaitu identifikasi, klasifikasi polimorfisme, dan pewarisan. Keuntungan menggunakan perancangan berorientasi objek diantaranya yaitu alami (Natural), dapat diandalkan (*Reliable*), dapat digunakan kembali (*Reusable*), mudah untuk dalam perawatan (*Maintainable*), dapat diperluas (*Extendable)*, dan efisiensi waktu [11].

2.4.2 UML (Unified ModelingLanguage)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. Pada UML terdapat sepuluh macam diagram untuk memodelkan aplikasi dengan berorientasi objek, yaitu [12] :

2.4.2.1 Diagram UML

Menggunakan berbagai macam diagram dengan fungsi masing-masing untuk menggambarkan setiap proses dari sistem berorientasi objek. Pada UML 2.8 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar Gambar 2.8.



Gambar 2.8. Diagram UML [12]

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkain perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsistem pada suatu sistem.

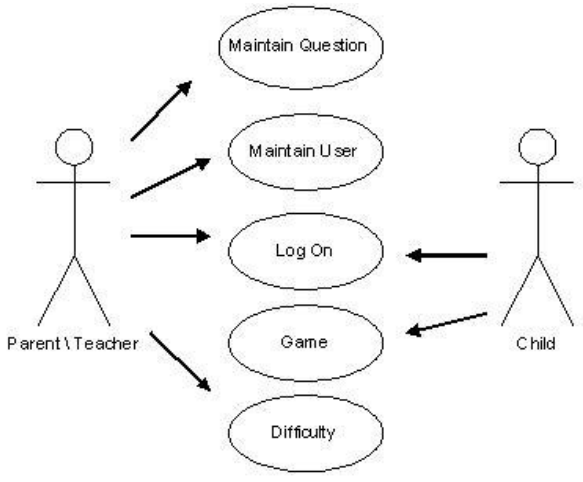
2.4.2.2 Use Case Diagram

Use Case atau diagram *use case* merupakan pemodelan yang digunakan untuk menggambarkan kelakuan (*behavior)* dari sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat diluar sistem yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesar antarunit atau aktor.

Contoh dari *use case diagram* dapat dilihat pada Gambar 2.9.



Gambar 2.9. Contoh Dari *Use Case Diagram*

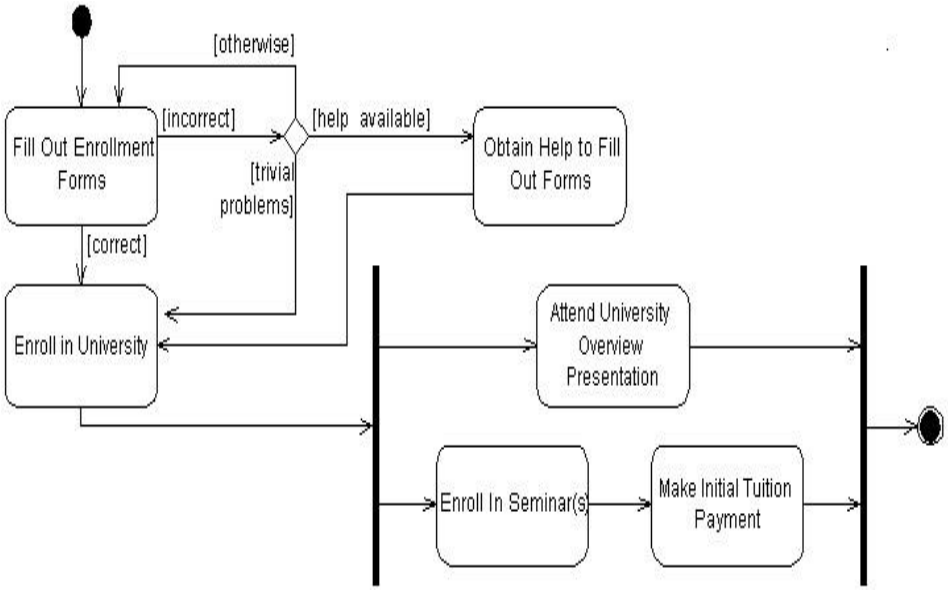
2.4.2.3 Activity Diagram

Diagram aktivitas atau *activity diagram* adalah sebuah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Dalam diagram aktivitas yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem, bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* di mana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengunjian di mana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Contoh dari *activity diagram* dapat dilihat pada gambar 2.10.



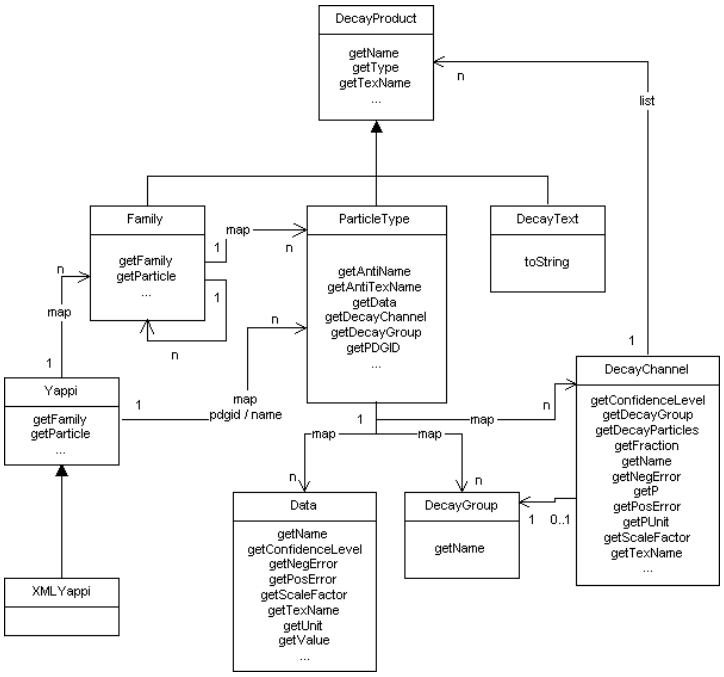
Gambar 2.10. Contoh Dari *Activity Diagram*

2.4.2.4 Class Diagram

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

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Contoh dari *class diagram* dapat dilihat pada gambar 2.11.



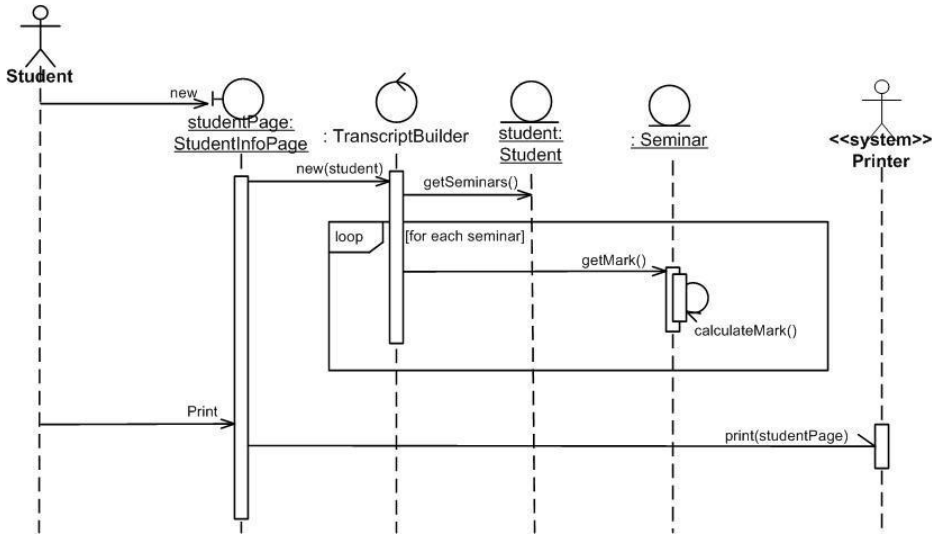
Gambar 2.11. Contoh Dari *Class Diagram*

2.4.2.5 Sequence Diagram

Diagram sekuen atau *sequence diagram* adalah diagram yang menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antarobjek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sekuen yang harus digambarkan adalah sebanyak pendefinisian *use case* yang memiliki prose situ sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Contoh dari *sequence diagram* dapat dilihat pada gambar 2.12.



Gambar 2.12. Contoh Dari *Sequence Diagram*

2.5 Unity

Unity adalah *software* atau *game engine* yang digunakan untuk membuat *video game* berbasis dua atau tiga dimensi dan dapat digunakan secara gratis. Selain untuk membuat *game*, *unity* juga dapat digunakan untuk membuat konten yang interaktif lainnya seperti, visual arsitektur dan *real-time* 3D animasi.

Unity adalah sebuah *tool* yang terintegrasi untuk membuat *game,* arsitektur bangunan dan simulasi. *Unity* bisa digunakan untuk *game*s PC dan *games online*. Untuk *games online* diperlukan sebuah plugin, yaitu *Unity Web Player*, yang sama halnya dengan *flash player* pada *browser*. Bahasa pemrograman yang digunakan bermacam-macam, mulai dari *javascript*, C#, dan *boo* [8]. *Unity* tidak bisa melakukan desain atau *modelling,* dikarenakan *unity* bukan merupakan *tools* untuk mendesain. Banyak hal yang bisa di lakukan di unity, ada fitur *audio reverb*

zone , particle effect , sky box untuk menambahkan langit, dan masih banyak lagi, dan juga bisa langsung *edit texture* dari *editor* seperti *photoshop* dan lain-lain.

Features (Scripting) di dalam *unity* adalah sebagai berikut [9]:

1. Mendukung 3 bahasa pemrograman, *JavaScript*, C#, dan *Boo*. *Flexible and EasyMoving*, *rotating*, dan *scaling objects* hanya perlu sebaris kode. Begitu juga dengan *duplicating*, *removing*, dan *changing properties.*
2. *Multi Platform Game* bisa di deploy di PC, Mac, Wii, iPhone, iPad dan *browser*, *android*.
3. *Visual Properties Variables* yang di definisikan dengan *scripts* ditampilkan pada editor. Bisa digeser, di *drag* and *drop*, bisa memilih warna dengan *color picker*.
4. Berbasis .NET, penjalanan program dilakukan dengan *Open Source* .NET *platform, Mono.*