

BAB II

LANDASAN TEORI

II.1 TINJAUAN PUSTAKA

Berdasarkan penelitian terdahulu yang dilakukan oleh Krisanti Nadya Kamanancy terkait Pembangunan Perangkat Lunak Bantu Klasterisasi dengan Metode Fuzzy C-Means menunjukkan bahwa metode fuzzy dapat memberikan solusi permasalahan tentang Clusterisasi. Selain itu Novita Nadia Gultom juga meneliti tentang Pengembangan Sistem Pendukung Keputusan Pembelian Laptop dengan metode Fuzzy Tahani. Hal ini menunjukkan bahwa penerapan metode fuzzy dapat digunakan untuk menyelesaikan banyak permasalahan yang tidak dapat diketahui nilainya secara pasti.

Prinsip logika fuzzy juga digunakan sebagai metode yang mencakup metode untuk membantu pengambilan keputusan terhadap beberapa alternatif keputusan untuk mendapatkan suatu keputusan yang akurat dan optimal. Salah satu metodenya adalah metode Fuzzy Multi-Criteria Decision Making (FMCDM). Dalam jurnal berjudul Fuzzy Multi-Criteria Decision Making, Sri Kusumadewi dan Idham Guswaludin (2005) telah berhasil menerapkan metode yang ada untuk menentukan lokasi pemancar televisi di Yogyakarta pada tiga alternatif lokasi dan lima kriteria.

Penelitian yang dilakukan oleh Wahyudi (2005) dengan judul Implementasi Fuzzy Logic Controller Pada Sistem Pengereman Kereta Api menyimpulkan bahwa Pengereman roda prototip kereta api dengan menggunakan kontrol logika fuzzy lebih lembut (soft), jika dibandingkan tanpa controller (konvensional). Selain itu, terdapat tulisan dari Eberhart dan Shi (2007) mengenai pengembangan tipe fuzzy controller

dengan menggunakan metode Mamdani untuk menyelesaikan masalah pengontrolan kecepatan dari kereta. Proses yang terjadi dideskripsikan dalam lima tahap, yaitu memutuskan nilai keanggotaan fuzzy, memutuskan rule yang akan digunakan dalam rule base matrix, menggabungkan nilai keanggotaan dengan menggunakan operator AND, menggabungkan nilai keanggotaan yang aktif untuk variabel output dan menggunakan centroid defuzzification untuk memutuskan variabel outputnya.

Selain itu, penelitian yang dilakukan oleh Resman, Hany, Thian dan Agus (1999) juga terkait Fuzzy Logic Controller dengan judul Implementasi Fuzzy Logic Pada Mikrokontroler Untuk Kendali Putaran Motor DC. Konsep Fuzzy Logic diterapkan dalam pembuatan alat pembuat gerabah yang direncanakan diputar oleh motor DC Servo yang dikontrol oleh mikrokontroler 8031 dan metode pengaturan yang digunakan adalah fuzzy logic. Fuzzy logic yang dirancang memiliki 2 input (Err & DErr) dan 1 output (Doutput). Masing-masing membership function memiliki 5 label. Disini digunakan 25 fuzzy if-then rule yang terdiri atas 9 rule pokok/utama, 10 rule tambahan dan 6 rule pelengkap. Sedangkan proses fuzzy logic terdiri dari fuzzifikasi, evaluasi rule dan yang terakhir defuzzifikasi.

Hal ini menambah keyakinan peneliti untuk melakukan penelitian tentang penggunaan logika fuzzy controller untuk pengereman Kereta Api untuk memberikan solusi berupa kekuatan pengereman yang lebih lembut agar masalah ketidaknyamanan yang dirasakan oleh penumpang Kereta Api dapat dikurangi. Aplikasi pengereman Kereta Api yang peneliti kembangkan merupakan sebuah aplikasi yang dikhususkan untuk menghasilkan output berupa kekuatan pengereman seperti kecil, sedang atau penuh yang akan diproses lagi untuk menghasilkan output berupa nilai *crisp* yang akan

digunakan sebagai acuan untuk melakukan pengereman. Inputnya berupa kecepatan kereta sebelum dilakukan pengereman dan jarak pengereman yang dengan stasiun sehingga sistem ini diharapkan dapat mengurangi ketidaknyaman penumpang terhadap hentakan dan bunyi pengereman yang mengganggu pendengaran. Selain itu pengembangan aplikasi ini juga didukung dengan adanya simulasi pengereman Kereta Api sehingga dapat dilihat hasil dari aplikasi ini secara representatif.



II.2 Logika Fuzzy (Fuzzy Logic)

II.2.1 Pengertian Logika Fuzzy, Himpunan *Crisp* dan Himpunan Fuzzy

Kata Fuzzy merupakan kata sifat yang berarti kabur, tidak jelas. *Fuzziness* atau kekaburan atau ketidakjelasan atau ketidakpastian selalu meliputi keseharian manusia. Logika fuzzy dikatakan sebagai logika baru yang lama, sebab ilmu tentang logika fuzzy modern dan metodis baru ditemukan beberapa tahun yang lalu, padahal sebenarnya konsep tentang logika fuzzy itu sendiri sudah ada sejak lama (Kusumadewi, 2003).

Logika fuzzy adalah suatu cara yang tepat untuk memetakan ruang input ke dalam suatu ruang output. Konsep ini diperkenalkan dan dipublikasikan pertama kali oleh Lotfi A. Zadeh, seorang profesor dari University of California di Berkeley pada tahun 1965. Logika fuzzy menggunakan ungkapan bahasa untuk menggambarkan nilai variabel. Logika fuzzy bekerja dengan menggunakan derajat keanggotaan dari sebuah nilai yang kemudian digunakan untuk menentukan hasil yang ingin dihasilkan berdasarkan atas spesifikasi yang telah ditentukan. Telah disebutkan sebelumnya bahwa logika fuzzy memetakan ruang input ke ruang output. Antara input dan output ada suatu kotak hitam yang harus memetakan input ke output yang sesuai.

Himpunan *crisp* a didefinisikan oleh item-item yang ada pada himpunan itu. Jika $a \in A$, maka nilai yang berhubungan dengan a adalah 1. Namun, jika $a \notin A$, maka nilai yang berhubungan dengan a adalah 0. Notasi $A = \{x | P(x)\}$ menunjukkan bahwa A berisi item x dengan P(x) benar. Jika X_A merupakan fungsi karakteristik A dan *property* P, maka dapat dikatakan bahwa P(x) benar, jika dan hanya jika $X_A(x)=1$.

Himpunan fuzzy didasarkan pada gagasan untuk memperluas jangkauan fungsi karakteristik sedemikian hingga fungsi tersebut akan mencakup bilangan real pada interval $[0,1]$. Nilai keanggotaannya menunjukkan bahwa suatu item dalam semesta pembicaraan tidak hanya berada pada 0 atau 1, namun juga nilai yang terletak diantaranya. Dengan kata lain, nilai kebenaran suatu item tidak hanya bernilai benar atau salah. Nilai 0 menunjukkan salah, nilai 1 menunjukkan benar, dan masih ada nilai-nilai yang terletak antara benar dan salah.

II.2.2 Alasan Penggunaan Logika Fuzzy

Alasan mengapa orang menggunakan logika fuzzy, yaitu (Kusumadewi, 2003):

1. Konsep logika fuzzy mudah dimengerti. Konsep matematis yang mendasari penalaran fuzzy sangat sederhana dan mudah dimengerti.
2. Logika fuzzy sangat fleksibel.
3. Logika fuzzy memiliki toleransi terhadap data-data yang tidak tepat.
4. Logika fuzzy mampu memodelkan fungsi-fungsi nonlinier yang sangat kompleks.
5. Logika fuzzy dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika fuzzy dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. Logika fuzzy didasarkan pada bahasa alami.

II.2.3 Fungsi Keanggotaan (Membership Function)

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (derajat keanggotaan) yang

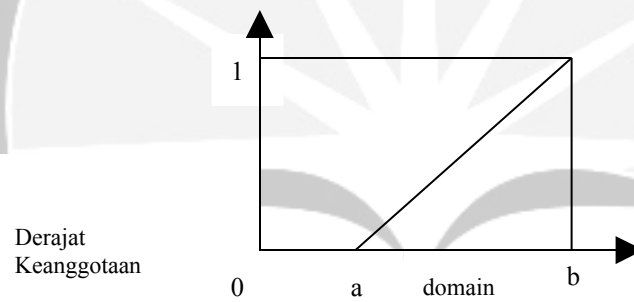
memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi.

Ada beberapa fungsi keanggotaan yang bisa digunakan, yaitu :

a. Representasi Linier

Pada representasi linier, pemetaan input ke derajat keanggotaannya digambarkan sebagai garis lurus. Ada 2 keadaan himpunan fuzzy yang linier.

1. Kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju nilai domain yang memiliki derajat keanggotaan lebih tinggi.

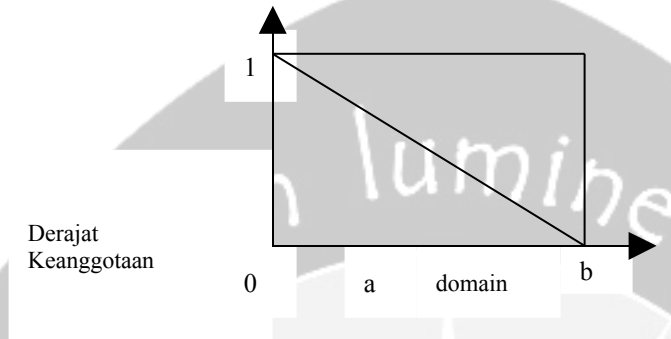


Gambar 2.1 Representasi Linier Naik

Fungsi Keanggotaan:

$$\mu [x] = \begin{cases} 0 & x \leq a \\ (x - a)/(b - a); & a \leq x \leq b \\ 1 & x \geq b \end{cases} \quad (2.1)$$

2. 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.



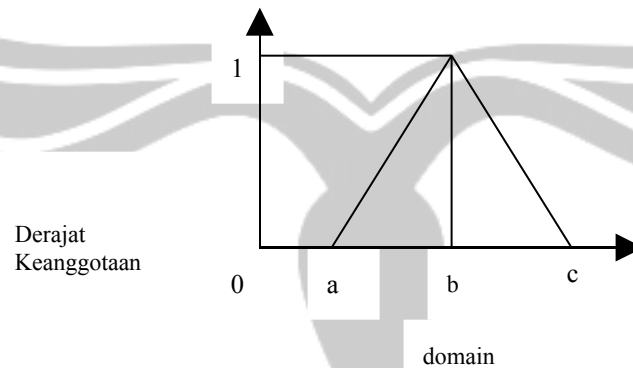
Gambar 2.2 Representasi Linier Turun

Fungsi Keanggotaan:

$$\mu[x] = \begin{cases} (b-x)/(b-a); & a \leq x \leq b \\ 0 & x \geq b \end{cases} \quad (2.2)$$

b. Representasi Kurva Segitiga

Kurva Segitiga pada dasarnya merupakan gabungan antara 2 garis (linier).



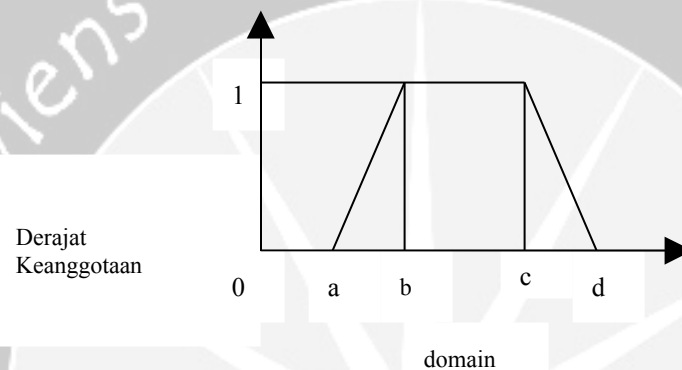
Gambar 2.3 Kurva Segitiga

Fungsi Keanggotaan:

$$\mu [x] = \begin{cases} 0 & x \leq a \text{ atau } x \geq c \\ (x - a)/(b - a); & a \leq x \leq b \\ (c - x)/(c - b); & b \leq x \leq c \end{cases} \quad (2.3)$$

c. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti kurva segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1.



Gambar 2.4 Kurva Trapesium

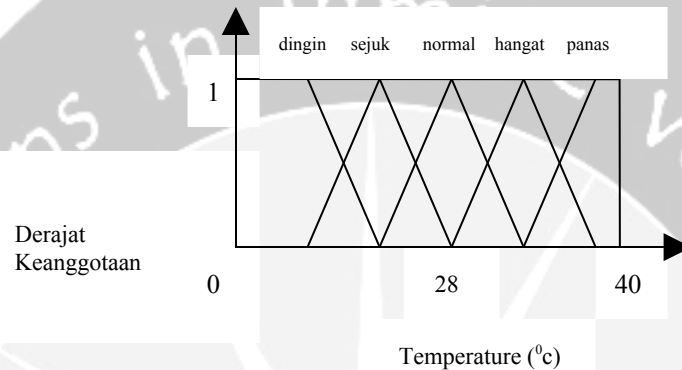
Fungsi Keanggotaan:

$$\mu [x] = \begin{cases} 0 & x \leq a \text{ atau } x \geq d \\ (x - a)/(b - a); & a \leq x \leq b \\ 1 & b \leq x \leq c \\ (d - x)/(d - c); & c \leq x \leq d \end{cases} \quad (2.4)$$

d. Representasi Kurva Bentuk Bahu

Daerah yang terletak di tengah-tengah suatu variabel yang direpresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun. Tetapi terkadang salah satu sisi dari variabel tersebut tidak mengalami perubahan. Himpunan fuzzy

‘bahu’, bukan segitiga, digunakan untuk mengakhiri variabel suatu daerah fuzzy. Bahu kiri bergerak dari benar ke salah, demikian juga bahu kanan bergerak dari salah ke benar. Sebagai contoh, himpunan fuzzy pada variabel TEMPERATUR dengan daerah bahunya.



Gambar 2.5 Kurva Bentuk Bahu

II.2.4 Operator Dasar Logika Fuzzy

Ada 3 operator dasar yang diciptakan oleh Zadeh, yaitu:

a. Operator AND (*Interseksi*)

Operator ini berhubungan dengan operasi *interseksi* pada himpunan. a-predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cap B} = \min(\mu_A[x], \mu_B[y]) \quad (2.5)$$

b. Operator OR (*Union*)

Operator ini berhubungan dengan operasi *union* pada himpunan. a-predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cup B} = \max(\mu_A[x], \mu_B[y]) \quad (2.6)$$

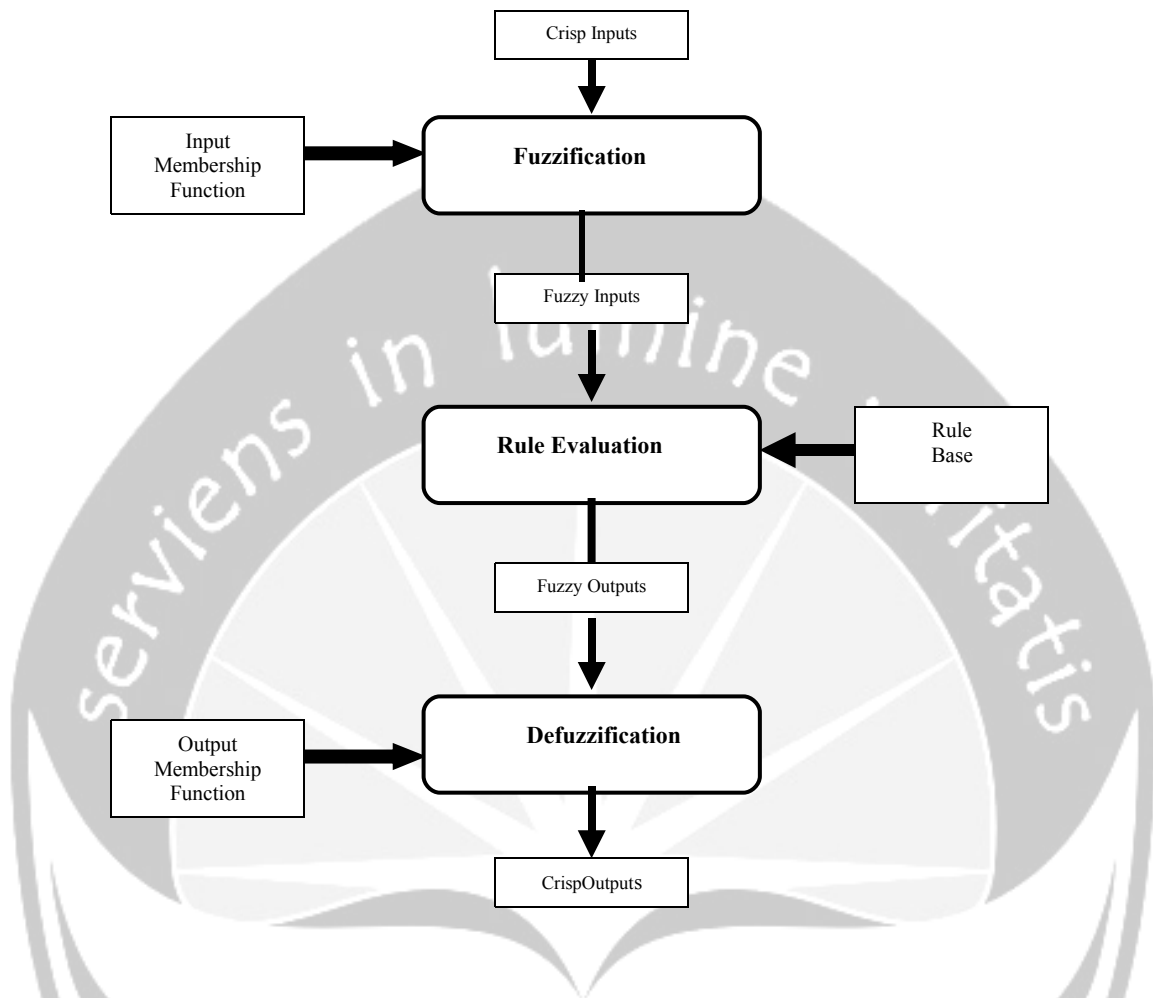
c. Operator NOT (*Komplemen*)

Operator ini berhubungan dengan operasi *komplemen* pada himpunan. a-predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

$$\mu_{A'} = 1 - \mu_A[x] \quad (2.7)$$

II.2.5 Pengendali Logika Fuzzy (Fuzzy Logic Controller)

Prinsip dasar pengendali logika fuzzy sebenarnya sangat sederhana. Bila pada pengontrol konvensional, sistem yang dikontrol dimodelkan secara analitis oleh sejumlah persamaan *diferensial*, yang solusinya menentukan aksi kontrol yang harus diberikan pada sistem, maka pengendali logika fuzzy didasarkan pada suatu model logika yang merepresentasikan proses berfikir seorang operator ketika sedang mengontrol suatu sistem. Di sini terjadi suatu pergeseran dari pemodelan sistem yang dikontrol menjadi pemodelan cara berfikir operator (Sumardi, 2000).



Gambar 2.6 Diagram Blok Pengendali Logika Fuzzy (Wahyudi,2005)

a. Fuzzifikasi

Fuzzifikasi merupakan suatu proses untuk mengubah suatu peubah masukan dari bentuk tegas (*crisp*) menjadi peubah fuzzy (*variable linguistik*) yang biasanya disajikan dalam bentuk himpunan-himpunan fuzzy (fuzzy input) dengan fungsi keanggotaannya masing-masing.

Pada dasarnya *fuzzifikasi* memiliki fungsi sebagai berikut:

1. Mengukur nilai variabel masukan.

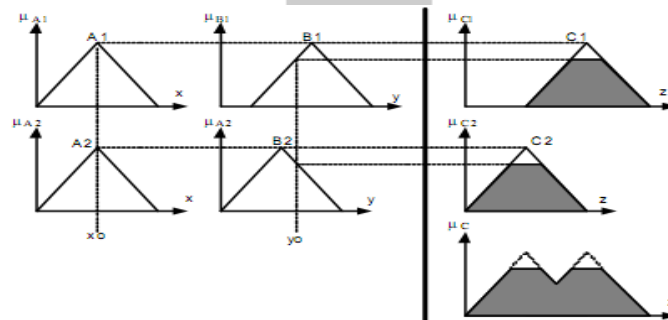
2. Melakukan pemetaan berskala yang merubah jangkauan dari nilai variabel masukan ke dalam semesta pembicaraan yang bersangkutan
3. Merumuskan fungsi *fuzzifikasi* yang merubah data masukan ke dalam nilai linguistik yang sesuai, yang akan digunakan sebagai label dari himpunan fuzzy.

b. Rule Evaluation dan Rules Base

Evaluasi aturan (*rule evaluation*) merupakan proses pengambilan keputusan (*inference*) yang berdasarkan aturan-aturan yang ditetapkan pada basis aturan (*rules base*) untuk menghubungkan antar peubah-peubah fuzzy masukan dan peubah fuzzy keluaran. Aturan-aturan ini berbentuk jika ... maka (IF ... THEN). Output yang dihasilkan pada bagian ini adalah fuzzy output. Teknik pengambilan keputusan yang digunakan adalah metode *max-min*. Pada metode *max-min*, pengambilan keputusan didasarkan pada aturan operasi menurut *Mamdani*.

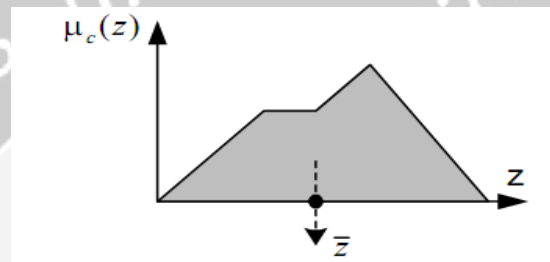
c. Defuzzifikasi

Defuzzifikasi merupakan proses pengubahan besaran fuzzy yang disajikan dalam bentuk himpunan-himpunan fuzzy keluaran dengan fungsi keanggotaannya untuk mendapatkan kembali bentuk tegasnya.



Gambar 2.7 Metode max -min

Hal ini diperlukan karena yang dikenal sebagai besaran sebenarnya untuk regulasi proses adalah nilai tegas. Metode *defuzzifikasi* yang digunakan adalah metode *centroid*. Metode *centroid* ini juga dikenal sebagai metode COA (*Center of Area*) atau metode *Center of Gravity*. Pada metode ini nilai tegas keluarannya diperoleh berdasarkan titik berat dari kurva hasil proses pengambilan keputusan yang dapat dilukiskan pada gambar di bawah ini.



Gambar 2.8 Metode Centroid

Persamaan matematis untuk metode *centroid*:

$$z = \frac{\int_z z \mu(z) dz}{\int_z \mu(z) dz} \text{ atau } z = \frac{\sum_{j=1}^n z_j \mu(z_j)}{\sum_{j=1}^n \mu(z_j)} \quad (2.8)$$

dimana:

$\mu(z)$ = agregasi output dari fungsi keanggotaan

z = nilai *center of area*

Ada dua keuntungan menggunakan metode *centroid*, yaitu:

1. Nilai *defuzzy* akan bergerak secara halus sehingga perubahan dari suatu topologi himpunan fuzzy ke topologi berikutnya juga akan berjalan dengan halus.
2. Mudah dihitung

II.3 Simulasi

II.3.1 Definisi Simulasi

Secara umum simulasi adalah peniruan atau penyerupaan dari proses atau keadaan yang nyata. Namun secara khusus simulasi adalah gerakan, gambaran atau tindakan tiruan yang dilakukan untuk menggambarkan sesuatu keadaan agar peristiwa atau proses yang sebenarnya akan terjadi bisa terlihat dengan jelas (Ensiklopedia Nasional Indonesia).

Selain definisi di atas, masih ada beberapa definisi tentang simulasi, yaitu:

1. Imitasi proses fisik atau objek oleh program yang menyebabkan komputer menanggapi data dan kondisi yang berubah seolah-olah ia merupakan proses objek itu sendiri (Kamus Komputer, 1996).
2. Suatu model pengambilan keputusan dengan mencontoh atau mempergunakan gambaran sebenarnya dari suatu sistem kehidupan dunia nyata tanpa harus mengalaminya pada keadaan sesungguhnya (Hasan, 2002).
3. Suatu peniruan sesuatu yang nyata, keadaan sekelilingnya (*state of affairs*), atau proses. Aksi melakukan simulasi sesuatu secara umum mewakili suatu karakteristik kunci atau kelakuan dari sistem-sistem fisik atau abstrak.

(<http://id.wikipedia.org/wiki/Simulasi>).

II.3.2 Jenis Simulasi

Ada beberapa jenis simulasi, yaitu (Kakiay, 2004):

a. Simulasi Identitas

Simulasi ini hanya memberikan sedikit kontrol atau bahkan tidak sama sekali terhadap situasi atau keadaan untuk mendapatkan jawaban yang efektif.

b. Simulasi Identitas Semu

Simulasi ini sudah selangkah lebih maju dibanding Simulasi Identitas. Simulasi Identitas Semu ini memodelkan berbagai aspek yang terkait dari sistem yang sebenarnya dan dapat mengeluarkan unsur-unsur yang dapat membuat setiap Simulasi Identitas dapat berfungsi dengan baik.

c. Simulasi Laboratorium

Simulasi ini lebih layak daripada Simulasi Identitas dan Simulasi Identitas Semu karena dapat memberikan jawaban yang lebih esensial pada masa yang akan datang. Biasanya Simulasi Laboratorium memerlukan berbagai komponen, seperti operator, software dan hardware, komputer, prosedur operasional, dan lain-lain.

d. Simulasi Komputer

Simulasi ini hanya menggunakan komputer untuk memecahkan masalah sesuai kebutuhan yang kemudian diprogramkan ke dalam komputer.

Dalam program simulasi yang akan dibuat dikategorikan dalam jenis simulasi komputer karena simulasinya menggunakan komputer sebagai media untuk pembuatan simulasi.

II.3.3 Kelebihan Simulasi

Keunggulan simulasi antara lain (Kakiay, 2004) :

1. Mampu mengakomodasi sistem yang kompleks.
2. Fleksibel dan dapat memodelkan berbagai tipe sistem.
3. Dapat melihat performansi sistem setiap saat, bahkan pada kondisi lain.
4. Tidak merusak sistem nyatanya.
5. Memvisualisasikan realitas sistem.
6. Menunjang detail sebuah desain.
7. Hasilnya dapat menjadi masukan perbaikan sistem

II.3.4 Alasan Penggunaan Simulasi

Alasan mengapa digunakan simulasi, adalah :

(<http://www.scribd.com/doc/13917155/Simulasi?autodown=pdf>)

1. Masalah sistem sangat kompleks tidak terstruktur sehingga sulit diselesaikan secara analitik.
2. Biaya sangat mahal apabila digunakan sistem aslinya (dalam percobaan).
3. Berbahaya bila digunakan “sistem aslinya” (dalam percobaan).
4. Waktu percobaan mungkin sangat panjang bila digunakan sistem aslinya.

II.4 Kereta Api

II.4.1 Pengertian Kereta Api

Kereta api adalah sarana transportasi berupa kendaraan dengan tenaga gerak, baik berjalan sendiri maupun dirangkaikan dengan kendaraan lainnya, yang akan ataupun

sedang bergerak di rel. Kereta api merupakan alat transportasi massal yang umumnya terdiri dari lokomotif (kendaraan dengan tenaga gerak yang berjalan sendiri) dan rangkaian kereta atau gerbong (dirangkai dengan kendaraan lainnya). Rangkaian kereta atau gerbong tersebut berukuran relatif luas sehingga mampu memuat penumpang maupun barang dalam skala besar. Karena sifatnya sebagai angkutan massal efektif, beberapa negara berusaha memanfaatkannya secara maksimal sebagai alat transportasi utama angkutan darat baik di dalam kota, antarkota, maupun antar negara (http://id.wikipedia.org/wiki/Kereta_api).

II.4.2 Kinematika Gerak Translasi

Gerak translasi yaitu perpindahan posisi benda dari suatu titik ke titik lain dengan arah gerak selalu berimpit dengan arah geraknya. Kecepatan (v) adalah perubahan posisi (Ds) suatu benda dalam selang waktu tertentu (Dt), dari pengertian kecepatan di atas dapat ditulis dengan persamaan berikut:

$$v = \frac{\Delta s}{\Delta t} \quad (2.9)$$

Percepatan atau perlambatan (a) adalah perubahan kecepatan (Dv) dalam selang waktu tertentu (Dt), dari pengertian percepatan atau perlambatan dapat ditulis dengan persamaan:

$$a = \frac{\Delta v}{\Delta t} \quad (2.10)$$

Tiga Hukum Newton yang berhubungan dengan masalah gerak yaitu:

1. Hukum I Newton

Hukum I Newton menyatakan bahwa, “Jika jumlah gaya yang bekerja pada benda sama dengan nol, maka benda yang sedang diam akan tetap diam dan benda yang sedang bergerak lurus dengan kecepatan tetap akan tetap bergerak lurus dengan kecepatan tetap”. Secara matematis Hukum I Newton dapat ditulis sebagai berikut :

$$\sum F = 0; \quad a = 0 \quad (2.11)$$

2. Hukum II Newton

Hukum II Newton menyatakan bahwa, “Percepatan (a) yang ditimbulkan oleh gaya yang beraksi pada sebuah benda dengan massa (m) berbanding lurus dengan besarnya gaya (F) dan berbanding terbalik dengan massa benda”. Secara matematis Hukum II Newton dapat ditulis sebagai berikut :

$$\sum F = m \cdot a \quad (2.12)$$

dimana:

$\sum F$ = resultan gaya (N)

a = percepatan benda (m/s²)

m = massa benda (kg)

3. Hukum III Newton

Hukum III Newton menyatakan bahwa, “Apabila suatu benda mengerjakan gaya pada benda lain, maka benda kedua akan mengerjakan gaya pada benda pertama yang sama besar dan berlawanan arah”. Secara matematis dapat ditulis sebagai berikut.

$$F_{aksi} = - F_{reaksi} \quad (2.13)$$

II.4.3 Faktor Yang Berpengaruh Pada Jarak Pengereman

Faktor Yang Berpengaruh Pada Jarak Pengereman adalah (Hartono, 2001) :

1. Kecepatan Kereta Api
2. Semakin tinggi kecepatan kereta api maka semakin panjang jarak pengereman.
3. Kemiringan / lereng (gradient) jalan rel.
4. Kemiringan jalan rel berpengaruh terhadap jarak pengereman dengan 2 kemungkinan yaitu menambah jarak pengereman jika lereng menurun atau mengurangi jarak pengereman jika lereng menaik.
5. Prosentase Gaya Pengereman. Prosentase gaya pengereman merupakan besaran gaya pengereman yang bekerja dibandingkan dengan berat kereta api yang akan dilakukan pengereman dikalikan dengan 100%. Semakin kecil besaran gaya pengereman maka akan semakin panjang jarak pengereman.

II.4.4 Mekanisme Pengereman Kereta Api

Pipa utama pengereman pada sistem pengereman udara tekan harus tersambung dari lokomotif hingga kereta atau gerbong terakhir. Sambungan saluran pengereman antara kendaraan dilakukan dengan alat penyambung yang terdiri dari selang karet dan penyambung dari logam seperti pada foto disamping. Pada bagian hulu dari selang karet dilengkapi dengan kran yang digunakan untuk menutup saluran udara pada ujung rangkaian. Perlu diperhatikan bahwa posisi kran selain pada ujung rangkaian harus terbuka. Kran pada bagian rangkaian yang bukan ujung yang tidak terbuka akan menyebabkan saluran udara dalam rangkaian tersumbat dan rem tidak bekerja dengan sempurna.

Sambungan ini harus kedap udara, karena tekanan udara 5 atm pada saluran utama tidak boleh bocor. Jika terjadi kebocoran dan produksi udara tekan dari lokomotif tidak bisa mengimbangi jumlah kebocoran, maka secara otomatis kereta api akan berhenti.

Pengereman di stasiun biasa disebut dengan pengereman penuh (full brake) pada rangkaian kereta api yang dilengkapi peralatan pengereman udara tekan (Westinghouse). Cara melakukan pengereman adalah menurunkan tekanan udara pada pipa utama sebesar 1,4 – 1,6 kg/cm² (1,4 – 1,6 atm) melalui tuas pengereman yang dilakukan masinis di lokomotif yang menyebabkan tekanan maksimum pada silinder pengereman kereta / gerbong mencapai 3,8 kg/cm² (3,8 atm) pada masing-masing kereta /gerbong.

Jarak pengereman dihitung dalam meter (m) sangat penting pengaruhnya pada kereta api sebagai bahan acuan bagi masinis kapan saatnya harus menarik tuas rem dan memulai pengereman untuk dapat berhenti pada waktu dan tempat yang ditentukan harus berhenti.

Dalam keadaan normal dimana kereta api yang berjalan dalam kecepatan penuh dan masinis menyadari bahwa kereta apinya harus berhenti di depan suatu sinyal karena tertahan oleh semboyan 7 (sinyal tidak boleh dilalui) maka masinis harus memperkirakan jarak pengereman dimana harus mulai menarik tuas rem sampai dengan kereta api harus dapat berhenti di muka sinyal tersebut.

II.5 Macromedia Flash MX 2004

II.5.1 Pengertian Macromedia Flash MX 2004

Macromedia Flash MX 2004 adalah sebuah software animasi yang sekarang menjadi software favorit para *web designer* untuk membuat webnya terlihat dinamis dan lebih atraktif. Bahkan sekarang Flash digunakan untuk berbagai keperluan, diantaranya untuk presentasi, proposal modern, e-card dan game.

Keunggulan program Macromedia Flash dibanding program lain yang sejenis adalah (Madcoms, 2004) :

1. Dapat membuat tombol interaktif dengan sebuah *movie* atau object yang lain.
2. Dapat membuat transparansi warna dalam *movie*.
3. Membuat perubahan animasi dari satu bentuk ke bentuk lain.
4. Dapat membuat gerakan animasi dengan mengikuti alur yang telah ditetapkan.
5. Dapat dikonversi dan dipublikasikan (*publish*) ke dalam beberapa tipe, diantaranya *.swf*, *.html*, *.gif*, *.png*, *.exe* dan *.mov*

II.5.2 Komponen Flash MX 2004

Di bawah ini merupakan komponen dari Flash MX yang digunakan dalam pekerjaan pembuatan animasi, yaitu:

1. *Timeline* : digunakan untuk mengatur dan mengontrol isi keseluruhan.
2. *Stage* : merupakan tempat membuat sebuah animasi.
3. *Tools Box* : berisi alat-alat yang digunakan untuk menggambar objek pada *stage*.

Tools Box memiliki komponen-komponen berikut :

- a. *Arrow Tool*

Tool ini digunakan untuk memilih suatu objek atau untuk memindahkannya.

b. *Subselection Tool*

Tool ini digunakan untuk merubah suatu objek dengan edit points.

c. *Line Tool*

Tool ini digunakan untuk membuat suatu garis di *stage*.

d. *Lasso Tool*

Tool ini digunakan untuk memilih daerah di objek yang akan diedit.

e. *Pen Tool*

Tool yang digunakan untuk menggambar dan merubah bentuk suatu objek dengan menggunakan edit points (lebih teliti & akurat).

f. *Text Tool*

Tool ini digunakan untuk menuliskan kalimat atau kata-kata.

g. *Oval Tool*

Tool yang digunakan untuk menggambar sebuah lingkaran.

h. *Rectangle Tool*

Tool yang digunakan untuk menggambar sebuah segiempat.

i. *Pencil Tool*

Tool ini digunakan untuk menggambar sebuah objek sesuai dengan yang disukai. Tetapi setiap bentuk yang dibuat akan diformat oleh Flash MX menjadi bentuk sempurna.

j. *Brush Tool*

Tool ini sering digunakan untuk memberi warna pada objek bebas.

k. *Free Transform Tool*

Tool ini digunakan untuk memutar (*rotate*) objek yang dibuat atau mengubah bentuk objek menjadi bentuk lain.

1. *Fill Transform Tool*

Tools ini digunakan untuk memutar suatu objek yang diimport dari luar lingkungan Flash MX, serta untuk mengatur efek warna.

m. *Ink Bottle Tool*

Tool ini digunakan untuk mengisi warna pada objek yang bordernya telah hilang (tidak ada).

n. *Paint Bucket Tool*

Tool ini digunakan untuk mengisi warna pada objek yang dipilih.

o. *Eraser Tool*

Tool ini digunakan untuk menghapus objek yang Anda bentuk.

4. *Color Window* : merupakan window yang digunakan untuk mengatur warna pada objek yang dibuat.

Color Window terdiri dari:

- a. *Color Mixer*, digunakan untuk mengatur warna pada objek sesuai dengan keinginan Anda. Ada 5 pilihan tipe warna, yaitu: None, Solid, Linear, Radial, Bitmap.
 - b. *Color Swatches*, digunakan untuk memberi warna pada objek yang Anda buat sesuai dengan yang warna pada window.
5. *Actions – Frame* : merupakan window yang digunakan untuk menuliskan Action Script untuk Flash MX. Biasanya Action Script digunakan untuk mengendalikan objek yang dibuat sesuai dengan keinginan.

6. *Properties* : merupakan window yang digunakan untuk mengatur *property* dari objek yang dibuat.
7. *Components* : digunakan untuk menambahkan objek untuk *web application* yang nantinya di publish ke internet.

II.5.3 Action Script 2.0

Action Script merupakan bahasa pemrograman yang ada dalam aplikasi flash dan digunakan untuk menciptakan interaktivitas pada *movie flash* agar dapat berinteraksi dengan pengguna (Chandra, 2006). Action Script 2.0 dirancang untuk memudahkan programmer untuk membuat aplikasi berbasis flash, keuntungannya antara lain adalah untuk sistem navigasi pada suatu situs atau presentasi, menghemat ukuran file dan membuat hal-hal yang bersifat interaktif.

Dalam hal penulisan, *script* terbagi menjadi dua yaitu (Chandra, 2006) :

- a. *Frame Script* : merupakan *script* yang diberikan pada frame.
- b. *Object Script* : merupakan *script* yang diberikan pada objek tombol atau *movie klip*.

Beberapa istilah dari Action Script (Chandra, 2006):

1. *Panel Actions*

Panel ini digunakan untuk membuat atau mengedit pemrograman Action Script pada *movie Flash*.

Elemen pada *panel action* :

- a. *Action Toolbox* berisi kumpulan *action* atau *script* yang disediakan oleh Flash.

- b. *Script Pane* adalah tempat untuk menulis dan mengedit *script*.
- c. *Script Assist* untuk mempermudah penulisan *script* hanya dengan beberapa klik mouse.

2. *Script Stop* Pada Frame

Script yang diberikan pada frame disebut *framescript*. Sedangkan *script stop* berfungsi untuk menghentikan jalannya animasi.

3. *Script play* Pada Frame

Script yang diberikan pada objek tombol disebut *buttonscript*. Sedangkan *script play* berguna untuk menjalankan *movie*.

4. *nextFrame* dan *prevFrame*

Script nextFrame berguna untuk maju satu frame berikutnya, sedangkan *prevFrame* berguna untuk mundur satu frame berikutnya.

5. *gotoAndStop* dan *gotoAndPlay*

Script gotoAndStop berguna untuk lompat ke frame tertentu kemudian berhenti pada frame tersebut, sedangkan *gotoAndPlay* berguna untuk lompat ke frame tertentu kemudian memainkan *movie* selanjutnya.

II.6 Object Oriented Programming (OOP)

II.6.1 Pengertian Object Oriented Programming

Konsep utama OOP yaitu melakukan permodelan objek dari kehidupan nyata ke dalam tipe data abstrak. OOP merupakan konsep pemrograman untuk memodelkan objek yang kita gunakan dalam kehidupan sehari-hari, dan konsep ini membawa perubahan yang mendasar dalam konsep pemrograman terstruktur.

II.6.2 Komponen Object Oriented Programming

Beberapa komponen penting yang sering digunakan dalam OOP antara lain ([http://www.IlmuKomputer.com/Tutorial belajar C# - Pengenalan C#-Bab 6- Pemrograman Berorientasi Objek.html](http://www.IlmuKomputer.com/Tutorial%20belajar%20C%20-%20Pengenalan%20C%20-Bab%206-%20Pemrograman%20Berorientasi%20Objek.html)):

- a. **Objek** : merupakan semua hal yang ada dalam dunia baik konkrit maupun nyata yang menjadi dasar intetitas runtime dalam suatu sistem berorientasi objek.
- b. **Class** : merupakan cetak biru (blueprint) atau template dari objek dengan kata lain class merupakan representasi abstrak sedangkan objek merupakan representasi nyata. Suatu class dapat berisi property, field, method dan event dari suatu objek yang disebut members dari suatu class. Dalam .net semua class dasar dikelompokkan kedalam *namespace*.
- c. **Method** : merupakan suatu tindakan yang dilakukan oleh objek
- d. **Event** : pemberitahuan yang diterima oleh objek
- e. **Field** : merupakan informasi atau atribut yang terdapat didalam suatu objek. Bentuk field mirip dengan variabel yaitu dapat dibaca dan diset secara langsung.
- f. **Property** : merupakan informasi (atribut) yang dapat disimpan dalam objek. Property dapat dideklarasikan dengan *Public Property* dan *Method Property* (dapat mengontrol operasi property). Terdapat 2 bagian dalam property yaitu *Get* dan *Set*. *Get* hanya memperbolehkan akses data sedangkan *Set* memperbolehkan mengubah nilai data.

II.6.3 Karakteristik Object Oriented Programming

Beberapa karakteristik OOP antara lain (Tessy, 2007) :

1. Encapsulation (pembungkusan)

- a. Variabel dan method dalam suatu obyek dibungkus agar terlindungi.
- b. Untuk mengakses, variabel dan method yang sudah dibungkus tadi perlu interface.
- c. Setelah variabel dan method dibungkus, hak akses terhadapnya bisa ditentukan.
- d. Konsep pembungkusan ini pada dasarnya merupakan perluasan dari tipe data struktur.

2. Inheritance (pewarisan)

Sebuah class bisa mewariskan atribut dan method-nya ke class yang lain. Class yang mewarisi disebut superclass. Class yang diberi warisan disebut subclass.

Sebuah subclass bisa mewariskan atau berlaku sebagai superclass bagi class yang lain => disebut multilevel inheritance.

3. Polymorphism (polimorfisme – perbedaan bentuk)

Polimorfisme artinya penyamaran dimana suatu bentuk dapat memiliki lebih dari satu bentuk.

II.7 NET Framework

II.7.1 Pengertian .NET Framework

.NET Framework merupakan platform yang memungkinkan kita untuk membangun aplikasi dan library yang disebut dengan "managed applications". .NET Framework menyediakan *compiler* dan tools agar kita dapat membangun, debug dan mengeksekusi *managed applications*.

II.7.2 Kelebihan .NET Framework

Beberapa kelebihan .NET Framework adalah :

(<http://www.site.uottawa.ca:4321/oose/framework.html>)

- a. Bisa simple atau kompleks
- b. Bisa ditulis menggunakan berbagai bahasa pemrograman
- c. Memungkinkan penggunaan kembali rancangan dan kode
- d. Menyediakan Fasilitas yang umum yang bisa diaplikasikan ke program yang berbeda
- e. Merepresentasikan struktur dari keseluruhan aplikasi atau subsistem

II.8 Bahasa Pemrograman C#

II.8.1 Pengertian C#

C# merupakan bahasa pemrograman dari Microsoft yang merupakan salah satu jenis bahasa pemrograman yang didukung .NET Framework.

II.8.2 Keuntungan Penggunaan C#

Keuntungan penggunaan C# adalah :

1. Sederhana (*Simple*)

C# bersifat sederhana karena bahasa ini didasarkan kepada bahasa C dan C++ yang sederhana.

2. *Object Oriented Language*

C# memenuhi syarat-syarat sebagai sebuah bahasa pemrograman yang bersifat *Object Oriented*, yaitu adanya *encapsulation*, *inheritance* dan *polymorphism*.

3. *Powerfull* dan *Fleksibel*

C# bisa digunakan untuk membuat berbagai macam aplikasi, seperti aplikasi pengolah kata, grafik, spreadsheets, atau bahkan membuat compiler untuk sebuah bahasa pemrograman.

4. *Efisien*

C# tidak memiliki terlalu banyak keyword, sehingga dapat mengurangi kerumitan.

5. *Modular*

Kode C# ditulis dengan pembagian masing-masing Class-Class (*classes*) yang terdiri dari beberapa *routines* yang disebut sebagai *member methods*. Class-Class dan metode-metode ini dapat digunakan kembali oleh program atau aplikasi lain. Hanya dengan memberikan informasi yang dibutuhkan oleh Class dan metode yang dimaksud, maka kita akan dapat membuat suatu kode yang dapat digunakan oleh satu atau beberapa aplikasi dan program (*reusable code*).