

BAB II

KAJIAN PUSTAKA

2.1 Teori Dasar

Logika *fuzzy*, adalah salah satu metode yang digunakan dalam pengambilan keputusan yang belum pasti berbasis aturan yang bertujuan untuk memberikan solusi atau memecahkan masalah, dimana suatu sistem yang sulit untuk dimodelkan atau terdapat ambigu atau ketidakpastian. Metode logika *fuzzy* dapat digunakan dalam pengambilan keputusan untuk menentukan tingkat kebugaran jasmani pada siswa sekolah dasar, yang dapat dibedakan antara bugar dengan yang kurang bugar. Logika *fuzzy* juga termasuk bagian dari kecerdasan buatan atau *Artificial Intelligence* dimana arti dari kecerdasan buatan tersebut akan dijelaskan lebih lanjut pada sub bab berikut ini.

2.1.1 Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan buatan berasal dari bahasa Inggris “*Artificial Intelligence*” atau disingkat *AI*, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan *Artificial* artinya buatan. Kecerdasan buatan yang dimaksud di sini merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan manusia. (Sutojo, dkk, 2011: 1). Alan Turing, ahli matematika berkebangsaan Inggris yang dijuluki bapak

komputer modern dan pembongkar sandi Nazi dalam era Perang Dunia II 1950, menetapkan definisi *Artificial Intelligence* “Jika komputer tidak dapat dibedakan dengan manusia saat berbincang melalui terminal komputer, maka bisa dikatakan komputer itu cerdas, mempunyai kecerdasan” (Sutojo, dkk, 2011:2).

John Mc Carthy dari Stanford mendefinisikan kecerdasan sebagai “kemampuan untuk mencapai sukses dalam menyelesaikan suatu permasalahan” (Sutojo, dkk, 2011: 2).

Ada beberapa definisi yang dapat di artikan oleh beberapa para ahli kecerdasan buatan (*Artificial Intelligence*), berikut adalah beberapa definisi mengenai kecerdasan buatan yang dapat diketahui, yaitu (Sutojo, dkk, 2011: 2-3):

a. Herbert Alexander Simon (*June 15, 1916-February 9, 2001*):

Kecerdasan buatan (*Artificial Intelligence*) merupakan kawasan penelitian, aplikasi, dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas.

b. Rich and Knight (1991):

Kecerdasan buatan (*AI*) merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia.

c. Encyclopedia Britannica:

Kecerdasan buatan (*AI*) merupakan cabang ilmu komputer yang dalam merepresentasi pengetahuan lebih banyak menggunakan bentuk simbol-simbol daripada bilangan dan memproses informasi berdasarkan metode heuristik atau dengan berdasarkan sejumlah aturan.

- d. Menurut Winston dan Prendergast (1984), tujuan kecerdasan buatan adalah:
1. Membuat mesin menjadi lebih pintar (tujuan utama)
 2. Memahami apa itu kecerdasan (tujuan ilmiah)
 3. Membuat mesin lebih bermanfaat (tujuan *entrepreneurial*)

Berdasarkan definisi ini, maka kecerdasan buatan menawarkan media maupun uji teori tentang kecerdasan. Teori-teori ini nantinya dapat dinyatakan dalam bahasa pemrograman dan eksekusinya dapat dibuktikan pada komputer nyata. Dari sini dapat dikatakan bahwa: cerdas adalah memiliki pengetahuan, pengalaman, dan penalaran untuk membuat keputusan dan mengambil tindakan. Jadi, agar mesin bisa cerdas (bertindak seperti manusia) maka harus diberi bekal pengetahuan dan diberi kemampuan untuk menalar (Sutojo, dkk, 2011: 3).

Jika dibandingkan dengan kecerdasan alami (kecerdasan yang dimiliki oleh manusia), kecerdasan buatan memiliki keuntungan komersial, antara lain Turban (1992) dalam Sutojo, dkk (2011: 10-11) mengemukakan:

1. Kecerdasan buatan lebih bersifat permanen.
2. Kecerdasan buatan lebih mudah diduplikasi dan disebar.
3. Kecerdasan buatan lebih murah dibandingkan kecerdasan alami.
4. Kecerdasan buatan lebih bersifat konsisten.
5. Kecerdasan buatan dapat didokumentasi.
6. Kecerdasan buatan dapat mengerjakan pekerjaan lebih cepat dibanding dengan kecerdasan alami.
7. Kecerdasan buatan dapat mengerjakan pekerjaan lebih baik dibanding dengan kecerdasan alami.

Didalam Kecerdasan Buatan (*Artificial Intelligence*) terdapat tiga cabang ilmu yang digunakan untuk mengimplementasikan Kecerdasan Buatan tersebut yaitu:

1. Sistem Pakar (*Expert System*).
2. Jaringan Saraf Tiruan.
3. Logika *Fuzzy* (*Fuzzy Logic*).

Berikut ini sedikit ulasan tentang ketiga cabang ilmu dalam Kecerdasan Buatan (*Artificial Intelligence*).

2.1.2 Sistem Pakar (*Expert System*)

Sistem pakar merupakan cabang dari *Artificial Intelligence* (AI) yang cukup tua karena sistem ini mulai dikembangkan pada pertengahan 1960. Sistem pakar yang muncul pertama kali adalah *General-purpose problem solver* (GPS) yang dikembangkan oleh Newel dan Simon. Sampai saat ini sudah banyak sistem pakar yang dibuat, seperti MYCIN untuk diagnosis penyakit, DENDRAL untuk mengidentifikasi struktur molekul campuran yang tak dikenal, XCON & XSEL untuk membantu konfigurasi sistem komputer besar, SOPHIE untuk analisis sirkuit elektronik, *Prospector* digunakan di bidang geologi untuk membantu mencari dan menemukan deposit, FOLIO digunakan untuk membantu memberikan keputusan bagi seorang manager dalam stok dan investasi, DELTA dipakai untuk pemeliharaan lokomotif listrik disel, dan sebagainya (Sutojo, dkk, 2011: 159).

Istilah sistem pakar berasal dari istilah *knowledge-based expert system*. Istilah ini muncul karena untuk memecahkan masalah, sistem pakar menggunakan pengetahuan seorang pakar yang dimasukkan kedalam komputer. Seseorang yang bukan pakar menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar menggunakan sistem pakar untuk *knowledge assistant* (Sutojo, dkk, 2011: 160). Berikut adalah beberapa pengertian sistem pakar (Sutojo, 2011: 160):

1. Turban (2001, p402). “Sistem pakar adalah sebuah sistem yang menggunakan pengetahuan manusia di mana pengetahuan tersebut dimasukkan ke dalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia”.
2. Jackson (1993, p3). “Sistem pakar adalah program komputer yang mempresentasikan dan melakukan penalaran dengan pengetahuan beberapa pakar untuk memecahkan masalah atau memberikan saran”.
3. Luger dan Stubblefield (1993, p308). “Sistem pakar adalah program yang berbasiskan pengetahuan yang menyediakan solusi kualitas pakar kepada masalah-masalah dalam bidang (*domain*) yang spesifik”.

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikan, diantaranya (Sutojo, dkk, 2011: 160-161):

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat.
2. Membuat seorang awam bekerja seperti layaknya seorang pakar.

3. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Andal. Sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Meningkatkan kapabilitas sistem komputer.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti. Berbeda dengan sistem komputer konvensional. Sistem pakar dapat bekerja dengan informasi yang tidak lengkap. Pengguna dapat merespon dengan “tidak tahu” atau “tidak yakin” pada satu atau lebih pertanyaan selama konsultasi dan sistem pakar tetap akan memberikan jawabannya.
10. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

Selain manfaat, ada juga beberapa kekurangan yang ada pada sistem pakar, diantaranya (Sutojo, dkk, 2011: 161):

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100% benar.

Ciri-ciri dari sistem pakar adalah sebagai berikut (Sutojo, dkk, 2011: 162):

1. Terbatas pada *domain* keahlian tertentu.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami.
4. Bekerja berdasarkan kaedah/*rule* tertentu.
5. Mudah dimodifikasi.
6. Basis pengetahuan dan mekanisme inferensi terpisah.
7. Keluarannya bersifat anjuran.
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna.

Konsep dasar sistem pakar meliputi enam hal berikut ini (Sutojo, dkk, 2011: 163-167):

1. *Kepakaran (Expertise)*. Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca dan pengalaman. Kepakaran inilah yang memungkinkan para ahli dapat mengambil keputusan lebih cepat dan lebih baik daripada seseorang yang bukan pakar (Sutojo, dkk, 2011: 163).
2. *Pakar (Expert)*. Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasihat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal baru yang berkaitan dengan topik permasalahan, jika perlu harus mampu menyusun kembali pengetahuan-

pengetahuan yang didapatkan, dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakarannya (Sutojo, dkk, 2011: 163).

3. Pemandahan Kepakaran (*Transferring Expertise*). Tujuan dari sistem pakar adalah memindahkan kepakaran dari seorang pakar ke dalam komputer, kemudian ditransfer kepada orang lain yang bukan pakar (Sutojo, dkk, 2011: 164).
4. Inferensi (*Inferencing*). Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dalam melakukan penalaran. Inerensi ditampilkan pada suatu komponen yang disebut mesin inferensi yang mencakup prosedur-prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki oleh seorang pakar disimpan pada basis pengetahuan oleh sistem pakar. Tugas mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimilikinya (Sutojo, dkk, 2011: 164).
5. Aturan-aturan (*Rule*). Kebanyakan *software* sistem pakar komersial adalah sistem yang berbasis *rule*, yaitu pengetahuan disimpan terutama dalam bentuk *rule*, sebagai prosedur-prosedur pemecahan masalah (Sutojo, dkk, 2011: 165).
6. Kemampuan menjelaskan (*Explanation Capability*)
Fasilitas lain dari Sistem Pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukan dalam subsistem yang disebut subsistem penjelasan (*Explanation*) (Sutojo, dkk, 2011: 165).

Pada sistem pakar berbasis *rule*, *domain* pengetahuan direpresentasikan dalam sebuah kumpulan *rule* berbentuk *IF-THEN*, sedangkan data direpresentasikan dalam sebuah kumpulan fakta-fakta tentang kejadian saat ini. Mesin inferensi membandingkan masing-masing *rule* yang tersimpan dalam basis pengetahuan dengan fakta-fakta yang terdapat dalam *database*. Jika bagian *IF* (kondisi) dari *rule* cocok dengan fakta maka *rule* dieksekusi dan bagian *THEN* (aksi) diletakkan dalam *database* sebagai fakta baru yang ditambahkan (Sutojo, dkk, 2011: 171).

Terdapat dua teknik inferensi dalam Sistem Pakar, yaitu (Sutojo, dkk, 2011: 171-178):

1. *Forward chaining*. *Forward chaining* adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian *IF* dari *rules IF-THEN*. Bila ada fakta yang cocok dengan bagian *IF*, maka *rule* tersebut dieksekusi. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru (bagian *THEN*) ditambahkan ke dalam *database*. Setiap kali pencocokan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi *rule* yang bisa dieksekusi. Metode pencarian yang digunakan adalah *Depth First Search (DFS)*, *Breath-First Search (BFS)* atau *Best First Search* (Sutojo, dkk, 2011: 171).
2. *Backward chaining*. *Backward chaining* adalah metode inferensi yang bekerja mundur ke arah kondisi awal. Proses diawali dari *Goal* (yang berada di bagian *THEN* dari *rule IF-THEN*), kemudian pencarian mulai dijalankan

untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis di bagian *IF*. Jika cocok, *rule* dieksekusi, kemudian hipotesis di bagian *THEN* ditempatkan di basis data sebagai fakta baru. Jika tidak cocok, simpan premis di bagian *IF* ke dalam *stack* sebagai sub *Goal*. Proses berakhir jika *Goal* ditemukan atau tidak ada *rule* yang bisa membuktikan kebenaran dari sub *Goal* atau *Goal* (Sutojo, dkk, 2011: 178).

2.1.3 Jaringan Saraf Tiruan (JST)

Jaringan Saraf Tiruan adalah paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia. Elemen kunci dan paradigma ini adalah struktur dari sistem pengolahan informasi yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (*neuron*), bekerja serentak untuk menyelesaikan masalah tertentu. Cara kerja JST seperti cara kerja manusia, yaitu belajar melalui contoh. Sebuah JST dikonfigurasi untuk aplikasi tertentu, seperti pengenalan pola atau klasifikasi data, melalui proses pembelajaran. Belajar dalam sistem biologis melibatkan penyesuaian terhadap koneksi *synaptic* yang ada antara *neuron*. Hal ini berlaku juga untuk JST (Sutojo, dkk, 2011: 283).

Kelebihan-kelebihan yang diberikan jaringan saraf tiruan antara lain (Sutojo, dkk, 2011: 284):

1. Belajar *Adaptive*: Kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengamatan awal.

2. *Self-Organisation*: Sebuah JST dapat membuat organisasi sendiri atau representasi dari informasi yang diterimanya selama waktu belajar.
3. *Real Time Operation*: Perhitungan JST dapat dilakukan secara paralel sehingga perangkat keras yang dirancang dan diproduksi secara khusus dapat mengambil keuntungan dari kemampuan ini.

Selain mempunyai kelebihan-kelebihan tersebut, JST juga mempunyai kelemahan-kelemahan berikut (Sutojo, dkk, 2011: 284-285):

1. Tidak efektif jika digunakan untuk melakukan operasi-operasi numerik dengan presisi tinggi.
2. Tidak efisien jika digunakan untuk melakukan operasi algoritma aritmatika, operasi logika, dan simbolis.
3. Untuk beroperasi JST butuh pelatihan sehingga bila jumlah datanya besar, waktu yang digunakan untuk proses pelatihan sangat lama.

Lapisan-lapisan penyusun JST dibagi menjadi tiga, yaitu (Sutojo, dkk, 2011: 292):

1. Lapisan *Input (Input Layer)*. Disebut unit *input* yang bertugas menerima pola *input* dari luar yang menggambarkan suatu permasalahan.
2. Lapisan Tersembunyi (*Hidden Layer*). Unit-unit dalam lapisan tersembunyi disebut unit-unit tersembunyi, yang mana nilai *output*nya tidak dapat diamati secara langsung.
3. Lapisan *Output (Output Layer)*. Unit-unit dalam lapisan *output* disebut unit-unit *output*, yang merupakan solusi JST terhadap suatu permasalahan.

Beberapa arsitektur jaringan yang sering digunakan dalam jaringan saraf tiruan antara lain (Sutojo, dkk, 2011: 292-294):

1. Jaringan Lapisan Tunggal.

Jaringan dengan lapisan tunggal terdiri dari 1 lapisan *input* dan 1 lapisan *output*. Setiap unit dalam lapisan *input* selalu terhubung dengan setiap unit yang terdapat pada lapisan *output*. Jaringan ini menerima *input* kemudian mengolahnya menjadi *output* tanpa melewati lapisan tersembunyi. Contoh JST yang menggunakan lapisan tunggal adalah *ADELIN*, *Hopfield*, *Perceptron*.

2. Jaringan Lapisan Banyak

Jaringan lapisan banyak mempunyai 3 jenis lapisan, yaitu lapisan *input*, lapisan tersembunyi, dan lapisan *output*. Jaringan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan dengan jaringan lapisan tunggal. Contoh JST yang menggunakan jaringan lapisan banyak adalah *MADALINE*, *backpropagation*, dan *Neocognitron*.

3. Jaringan dengan Lapisan Kompetitif

Jaringan ini memiliki bobot yang telah ditentukan dan tidak memiliki proses pelatihan. Jaringan ini digunakan untuk mengetahui *neuron* pemenang dari sejumlah *neuron* yang ada. Akibatnya, pada jaringan ini sekumpulan *neuron* bersaing untuk mendapatkan hak menjadi aktif. Nilai bobot setiap *neuron* untuk dirinya sendiri adalah 1, sedangkan untuk *neuron* lainnya bernilai *random negatif*. Contoh JST yang menggunakan jaringan dengan lapisan kompetitif adalah *LVQ*.

Terdapat dua algoritma pembelajaran pada Jaringan Saraf Tiruan (JST) antara lain algoritma pembelajaran dengan supervisi dan algoritma pembelajaran tanpa supervisi (Jaringan Kohonen). Berikut ini metode pembelajaran dengan supervisi pada Jaringan Saraf Tiruan (Sutojo, dkk, 2011: 307):

1. *Hebb Rule*. Model *neuron* McCulloch-Pitts akan mengalami kesulitan bila berhadapan dengan fungsi-fungsi yang kompleks. Hali ini terjadi karena dalam menentukan bobot w dan nilai ambang Θ harus dilakukan secara analitik atau menggunakan cara coba-coba. Pada 1949, D.O. Hebb memperkenalkan cara menghitung bobot w dan bias secara iteratif dengan memanfaatkan model pembelajaran dengan supervisi sehingga bobot w dan bias dapat dihitung secara otomatis tanpa harus melakukan cara coba-coba. Model Hebb merupakan model jaringan tertua yang menggunakan pembelajaran dengan supervisi. Arsitektur jaringan Hebb sama seperti arsitektur jaringan McCilloh-Pitts, yaitu beberapa unit *input* dihubungkan langsung dengan sebuah unit *output*, ditambah dengan sebuah bias (Sutojo, dkk, 2011: 307).
2. *Perceptron*. Model jaringan perceptron merupakan model yang paling baik pada saat itu. Model ini ditemukan oleh Rosenblatt (1962) dan Minsky – Papert (1969) (Sutojo, dkk, 2011: 326).
3. *Delta Rule*. Selama pelatihan pola, *delta rule* akan mengubah bobot dengan cara meminimalkan *error* antara *output* jaringan y dengan target t (Sutojo, dkk, 2011: 352).

4. *Backpropagation*. *Backpropagation* adalah metode penurunan gradien untuk meminimalkan kuadrat *error* keluaran. Ada tiga tahap yang harus dilakukan dalam pelatihan jaringan, yaitu tahap perambatan maju (*forward propagation*), tahap perambatan balik, dan tahap perumahan bobot dan bias. Arsitektur jaringan ini terdiri dari *input layer*, *hidden layer* dan *output layer* (Sutojo, dkk, 2011: 360).
5. *Heteroassociative Memory*. Jaringan saraf *heteroassociative memory* adalah jaringan yang dapat menyimpan kumpulan pengelompokan pola, dengan cara menentukan bobot-bobotnya sedemikian rupa. Setiap kelompok merupakan pasangan vektor $(s(n), t(n))$ dengan $n=1, 2, \dots, N$. Algoritma pelatihan yang biasa digunakan adalah *Hebb Rule* (Sutojo, dkk, 2011: 374).
6. *Bidirectional Associative Memory (BAM)*. *Bidirectional Associative Memory (BAM)* adalah model jaringan saraf yang memiliki 2 lapisan, yaitu lapisan *input* dan lapisan *output* yang mempunyai hubungan timbal balik antara keduanya. Hubungan ini bersifat *bidirectional*, artinya jika bobot matriks dari sinyal yang dikirim dari lapisan *input* X ke lapisan *output* Y adalah W, maka bobot matriks dari sinyal yang dikirim dari lapisan *output* Y ke lapisan *input* X adalah W^T (Sutojo, dkk, 2011: 380).
7. *Learning Vector Quantization (LVQ)*. *Learning Vector Quantization (LVQ)* adalah suatu metode pelatihan pada lapisan kompetitif terawasi yang akan belajar secara otomatis untuk mengklasifikasikan vektor-vektor *input* ke dalam kelas-kelas tertentu. Kelas-kelas yang dihasilkan tergantung pada jarak antara vektor-vektor *input*. Jika ada 2 vektor *input* yang hampir sama

maka lapisan kompetitif akan mengklasifikasikan kedua vektor *input* tersebut ke dalam kelas yang sama (Sutojo, dkk, 2011: 387).

Sedangkan pembelajaran tanpa supervisi, orang pertama yang memperkenalkan jaringan kohonen ini adalah Prof. Teuvo Kohonen pada tahun 1982. Pada jaringan ini, *neuron-neuron* pada suatu lapisan akan menyusun dirinya sendiri berdasarkan *input* nilai tertentu dalam suatu *cluster*. Dalam proses penyusunan diri, *cluster* yang dipilih sebagai pemenang adalah *cluster* yang mempunyai vektor bobot paling cocok dengan pola *input* (Sutojo, dkk, 2011: 392).

2.1.4 Logika *Fuzzy* (*Fuzzy Logic*)

Logika *Fuzzy* merupakan logika samar yang dapat memecahkan suatu ketidakpastian dalam proses pemecahan masalah suatu data melalui ruang *input* sebagai data masukan yang akan dipecahkan ketidakpastiannya dan ruang *output* sebagai solusi dari hasil ketidakpastian.

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. Dalam banyak hal, logika *fuzzy* digunakan sebagai suatu cara untuk memetakan permasalahan dari *input* menuju ke *output* yang diharapkan. Logika

fuzzy dapat dianggap sebagai kotak hitam yang menghubungkan antar ruang *input* menuju ke ruang *output* (Gelley, 2000 dalam Kusumadewi dan Purnomo, 2010: 2). Kotak hitam tersebut berisi cara atau metode yang saat digunakan untuk mengolah data *input* menjadi *output* dalam bentuk informasi yang baik (Kusumadewi dan Purnomo, 2010: 1-2).

Secara umum, *fuzzy logic* adalah sebuah metodologi “berhitung” dengan variabel kata-kata (*linguistic variable*), sebagai pengganti berhitung dengan bilangan. Kata-kata yang digunakan dalam *fuzzy logic* memang tidak sepresisi bilangan, namun kata-kata jauh lebih dekat dengan intuisi manusia. Manusia bisa langsung “merasakan” nilai dari variabel kata-kata yang sudah dipakainya sehari-hari. Demikianlah *fuzzy logic* memberi ruang dan bahkan mengeksploitasi toleransi terhadap ketidakpresisian. *Fuzzy logic* membutuhkan “ongkos” yang lebih murah dalam memecahkan berbagai masalah yang bersifat *fuzzy*. *Fuzzy logic* telah menjadi area riset yang mengagumkan karena kemampuannya dalam menjembatani bahasa mesin yang serba presisi dengan bahasa manusia yang cenderung tidak presisi, yaitu hanya dengan menekankan pada makna atau arti (Naba, 2009: 1-2).

Ada beberapa alasan mengapa orang menggunakan logika *fuzzy* (Cox, 1994 dalam Kusumadewi dan Purnomo, 2010: 2-3) antara lain:

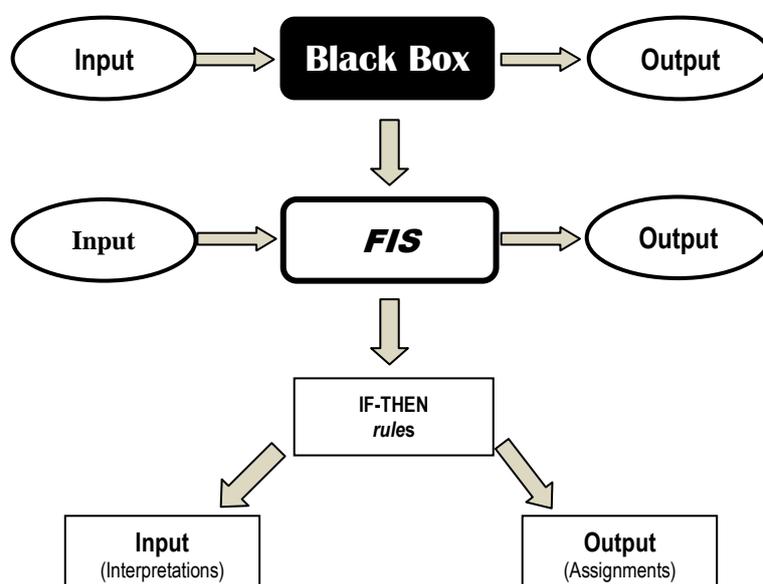
1. Konsep logika *fuzzy* mudah dimengerti. Karena logika *fuzzy* menggunakan dasar teori himpunan, maka konsep matematis yang mendasari penalaran *fuzzy* tersebut cukup mudah untuk dimengerti.

2. Logika *fuzzy* sangat fleksibel, artinya mampu beradaptasi dengan perubahan-perubahan, dan ketidakpastian yang menyertai permasalahan.
3. Logika *fuzzy* memiliki toleransi terhadap data yang tidak tepat. Jika diberikan kelompok 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-fungsi nonlinier yang sangat kompleks.
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan. Dalam hal ini, sering dikenal dengan nama *Fuzzy Expert System*.
6. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional. Hal ini umumnya terjadi pada aplikasi di bidang teknik mesin maupun teknik elektro.
7. Logika *fuzzy* didasarkan pada bahasa alami. Logika *fuzzy* menggunakan bahasa sehari-hari sehingga mudah dimengerti.

2.1.4.1 Konsep *Fuzzy Logic*

Motivasi utama teori *fuzzy logic* adalah memetakan sebuah ruang *input* ke dalam ruang-ruang *output* dengan menggunakan *IF-THEN rules*. Pemetaan dilakukan dalam *Fuzzy Inference System (FIS)*. Urutan *rule* dapat sembarang. *FIS* mengevaluasi semua *rule* secara simultan untuk menghasilkan kesimpulan. Oleh karenanya, semua *rule* harus didefinisikan lebih dahulu sebelum membangun *FIS*

yang akan digunakan untuk menginterpretasikan semua *rule* tersebut. Mekanisme dalam *FIS* dapat dirangkum seperti ini: *FIS* adalah sebuah metode yang menginterpretasikan harga-harga dalam vektor *input*, menarik kesimpulan berdasarkan *IF-THEN rules* yang diberikan, dan kemudian menghasilkan vektor *output* (Naba, 2009: 13-14). Kronologi proses rancang bangun *FIS* diilustrasikan dalam Gambar 2.1.



Gambar 2.1 Konsep Umum Kronologi Proses Pembangunan *FIS*

2.1.4.2 Dasar-Dasar Logika *Fuzzy*

Untuk memahami logika *fuzzy*, harus memperhatikan dahulu tentang konsep himpunan *fuzzy*. Himpunan *fuzzy* memiliki 2 atribut (Sutojo, dkk, 2011: 212-213), yaitu:

1. *Linguistik*, yaitu nama suatu kelompok yang mewakili suatu keadaan tertentu dengan menggunakan bahasa alami, misalnya DINGIN, SEJUK,

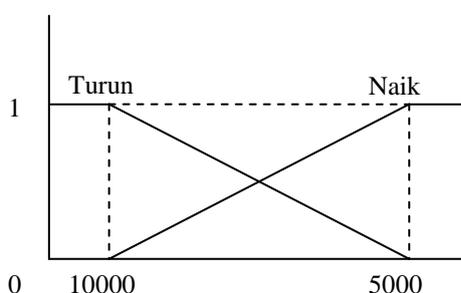
PANAS mewakili temperatur. Contoh lainnya seperti MUDA, PAROBAYA, TUA mewakili variabel umur.

2. *Numeris*, yaitu suatu nilai yang menunjukkan ukuran dari suatu variabel, misalnya 10, 35, 40, dan sebagainya.

Disamping itu, ada beberapa hal yang harus dipahami dalam memahami logika *fuzzy*, yaitu:

1. Variabel *fuzzy*, yaitu variabel yang akan dibahas dalam suatu sistem *fuzzy*.
Contoh: penghasilan, temperatur, permintaan, umur, dan sebagainya.
2. Himpunan *fuzzy*, yaitu suatu kelompok yang mewakili suatu keadaan tertentu dalam suatu variabel *fuzzy*.

Variabel permintaan, terbagi menjadi 2 himpunan, yaitu NAIK dan TURUN.



Gambar 2.2 Variabel Permintaan Terbagi Menjadi 2 Himpunan *Fuzzy*

3. Semesta pembicaraan, yaitu seluruh nilai yang diizinkan untuk dioperasikan dalam suatu variabel *fuzzy*. Contoh:
 - a. Semesta pembicaraan untuk variabel permintaan: $[0+\infty)$
 - b. Semesta pembicaraan untuk variabel temperatur: $[-10\ 90]$
4. *Domain* himpunan *fuzzy*, yaitu seluruh nilai yang diizinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Pada

Gambar 2.2 *domain* untuk himpunan TURUN dan himpunan NAIK masing masing adalah:

- a. *Domain* himpunan TURUN= [0 500]
- b. *Domain* himpunan NAIK = [1000 $+\infty$)

2.1.4.3 Fungsi Keanggotaan

Fungsi keanggotaan adalah grafik yang mewakili besar dari derajat keanggotaan masing-masing variabel *input* yang berada dalam internal antara 0 dan 1. Derajat keanggotaan sebuah variabel x dilambangkan dengan simbol $\mu(x)$. *Rule* menggunakan nilai keanggotaan sebagai faktor bobot untuk menentukan pengaruhnya pada saat melakukan inferensi untuk menarik kesimpulan (Sutojo, dkk, 2011: 213).

Fungsi keanggotaan (*membership function*) 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 dengan 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi (Kusumadewi dan Purnomo, 2010: 8).

Secara umum, beberapa kesimpulan tentang himpunan dan fungsi keanggotaan *fuzzy* diberikan dibawah ini (Naba, 2009: 21):

1. *Fuzzy set* menekankan konsep variabel samar (*vague or fuzzy variable*) seperti variabel hari akhir minggu, suhu panas, pelari cepat, dan lain-lain.

2. *Fuzzy set* mengizinkan keanggotaan parsial dari suatu himpunan seperti hari jumat yang dianggap sebagai hari akhir minggu namun dengan derajat dibawah 1.
3. Derajat keanggotaan *fuzzy* dalam *fuzzy set* berkisar antara 0 sampai 1.
4. Tiap fungsi keanggotaan μ berasosiasi dengan sebuah *fuzzy set* tertentu dan memetakan suatu nilai *input* ke nilai derajat keanggotaan yang sesuai. Misalnya dalam kasus *fuzzy set* orang berbadan “tinggi” mempunyai fungsi keanggotaan sendiri, yaitu μ_{tinggi} , yang berbeda dengan fungsi keanggotaan dari *fuzzy set* orang berbadan “rendah”, yaitu μ_{rendah}

Ada beberapa fungsi yang bisa digunakan (Kusumadewi dan Purnomo, 2010: 1-13) yaitu:

1. Representasi Linear

Pada representasi linear, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk 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 bergerak ke kanan menuju ke nilai *domain* yang memiliki derajat keanggotaan lebih tinggi. 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.

2. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis linear.

3. Representasi Kurva Trapesium

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

4. Representasi Kurva Bentuk Bahu

Grafik keanggotaan kurva “bahu” digunakan untuk mengakhiri variabel suatu daerah *fuzzy* yang nilai derajat keanggotaannya adalah konstan (biasanya 1).

5. Representasi Kurva-S (*Sigmoid*)

Grafik keanggotaan kurva S memiliki bentuk seperti huruf “S” yang mempunyai ukuran yang diletakkan oleh parameter a , b , dan c . Titik b disebut titik infleksi, yaitu titik yang mempunyai derajat keanggotaan 0,5. Ada dua macam kurva-S, yaitu kurva-S pertumbuhan dan kurva-S penyusutan.

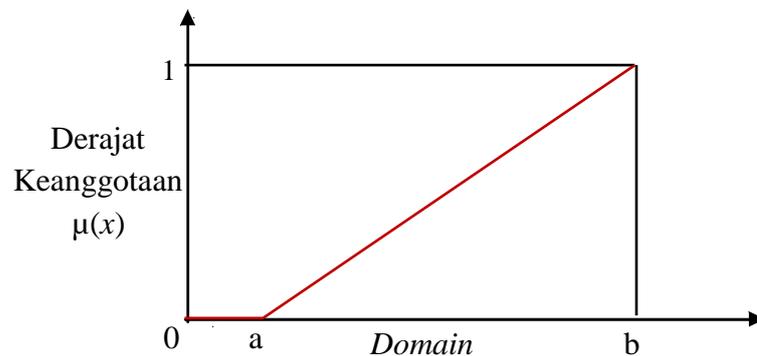
6. Representasi Kurva Bentuk Lonceng (*Bell Curve*)

Selain kurva-kurva di atas, kurva berbentuk lonceng juga bisa digunakan untuk merepresentasikan bilangan *fuzzy*. Kurva ini terbagi menjadi tiga, yaitu kurva P1, kurva beta, dan kurva Gauss. Ketiganya dibedakan oleh gradien yang dibentuknya.

2.1.4.3.1 Representasi Linear

Pada grafik keanggotaan kurva linear, pemetaan *input* ke derajat keanggotaannya dapat digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas (Kusumadewi dan Purnomo, 2010: 1-13). Menurut Kusumadewi dan Purnomo (2010: 9) ada 2 keadaan himpunan *fuzzy* yang linear yaitu:

1. Representasi linear naik, yaitu kenaikan himpunan dimulai dari nilai *domain* yang memiliki nilai keanggotaan nol [0] bergerak ke kanan menuju ke nilai *domain* yang memiliki derajat keanggotaan yang lebih tinggi (Kusumadewi dan Purnomo, 2010: 9).



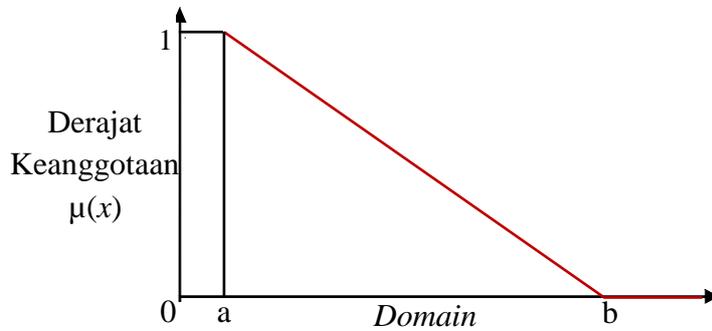
Gambar 2.3 Representasi Linear Naik

Fungsi keanggotaan sebagai berikut :

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

2. Representasi Linear turun, yaitu garis lurus yang dimulai dari nilai *domain* dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak

turun ke nilai *domain* yang memiliki derajat keanggotaan lebih rendah (Kusumadewi dan Purnomo, 2010: 10).



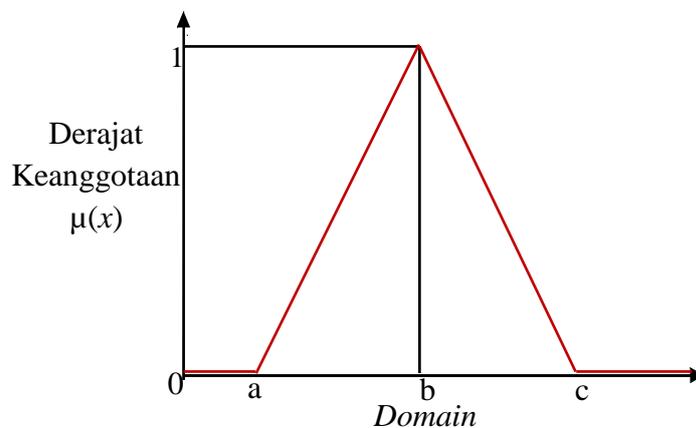
Gambar 2.4 Representasi Linear Turun

Fungsi Keanggotaan:

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

2.1.4.3.2 Representasi Kurva Segitiga

Representasi kurva segitiga, pada dasarnya adalah gabungan antara dua garis linear (Kusumadewi dan Purnomo, 2010: 11).



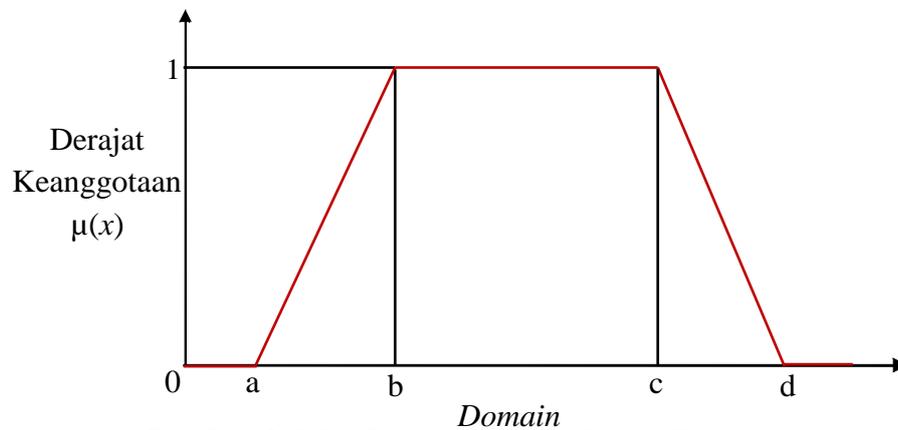
Gambar 2.5 Representasi Kurva Segitiga

Fungsi Keanggotaan:

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

2.1.4.3.3 Representasi Kurva Trapesium

Representasi kurva trapesium pada dasarnya seperti bentuk kurva segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1 (satu) (Kusumadewi dan Purnomo, 2010: 13).



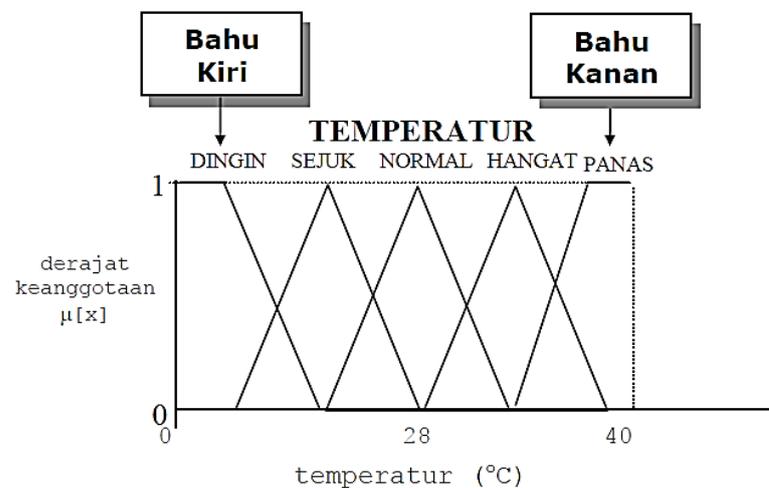
Gambar 2.6 Grafik Keanggotaan Kurva Trapesium

Fungsi Keanggotaan:

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

2.1.4.3.4 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 (misalkan: DINGIN bergerak ke SEJUK bergerak ke HANGAT dan bergerak ke PANAS) (Kusumadewi dan Purnomo, 2010: 14).

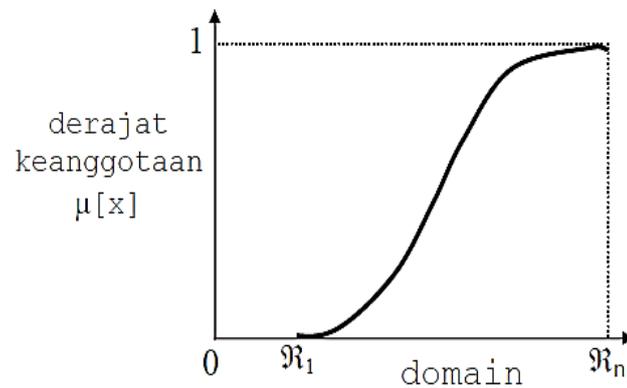


Gambar 2.7 Kurva Bentuk Bahu

2.1.4.3.5 Representasi Kurva-S

Kurva PERTUMBUHAN dan PENYUSUTAN merupakan kurva-S atau sigmoid yang berhubungan dengan kenaikan dan penurunan permukaan secara tak linear (Kusumadewi dan Purnomo, 2010: 15).

Kurva-S untuk PERTUMBUHAN akan bergerak dari sisi paling kiri (nilai keanggotaan = 0) ke sisi paling kanan (nilai keanggotaan = 1). Fungsi keanggotaannya akan tertumpu pada 50% nilai keanggotaannya yang sering disebut dengan titik infleksi (Kusumadewi dan Purnomo, 2010: 15).

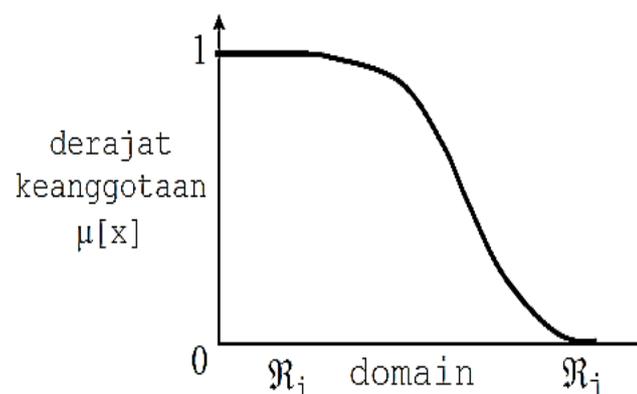


Gambar 2.8 Himpunan *Fuzzy* dengan Kurva-S PERTUMBUHAN

Fungsi Keanggotaan:

$$S(x; \alpha; \beta; \gamma) = \begin{cases} 0 & \rightarrow x \leq \alpha \\ 2((x - \alpha) / (\gamma - \alpha))^2 & \rightarrow \alpha \leq x \leq \beta \\ 1 - 2((\gamma - x) / (\gamma - \alpha))^2 & \rightarrow \beta \leq x \leq \gamma \\ 1 & \rightarrow x \geq \gamma \end{cases} \dots \text{Rumus 2.5}$$

Kurva-S untuk PENYUSUTAN akan bergerak dari sisi paling kanan (nilai keanggotaan = 1) ke sisi paling kiri (nilai keanggotaan = 0) (Kusumadewi dan Purnomo, 2010: 15).



Gambar 2.9 Himpunan *Fuzzy* dengan Kurva-S PENYUSUTAN

Fungsi Keanggotaan:

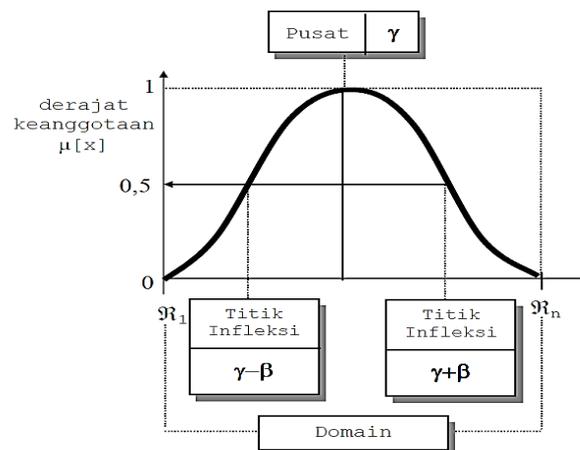
$$S(x; \alpha; \beta; \gamma) = \begin{cases} 1 & \rightarrow x \leq \alpha \\ 1 - 2((x - \alpha) / (\gamma - \alpha))^2 & \rightarrow \alpha \leq x \leq \beta \\ 2((\gamma - x) / (\gamma - \alpha))^2 & \rightarrow \beta \leq x \leq \gamma \\ 0 & \rightarrow x \geq \gamma \end{cases} \quad \dots \text{Rumus 2.6}$$

2.1.4.3.6 Representasi Kurva Bentuk Lonceng

Untuk merepresentasikan bilangan *fuzzy*, biasanya digunakan kurva berbentuk lonceng. Kurva berbentuk lonceng ini terbagi atas 3 kelas, yaitu: himpunan *fuzzy* PI, Beta, dan Gauss. Perbedaan ketiga kurva ini terletak pada gradiennya (Kusumadewi dan Purnomo, 2010: 19).

1. Kurva PI

Kurva PI berbentuk lonceng dengan derajat keanggotaan 1 terletak pada pusat dengan *domain* (γ), dan lebar kurva (β) (Kusumadewi dan Purnomo, 2010: 19).



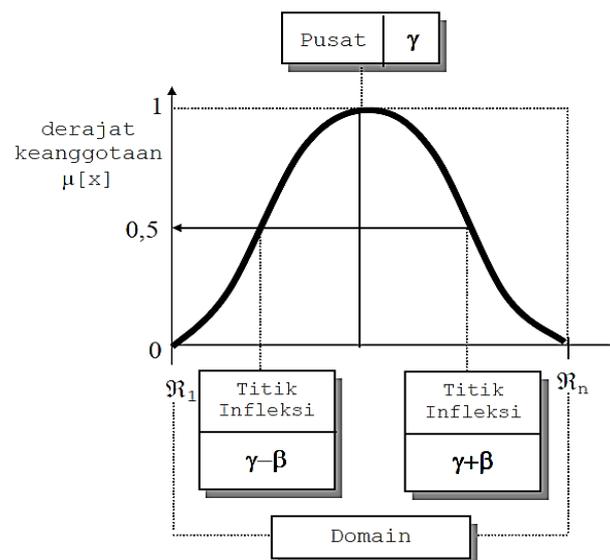
Gambar 2.10 Karakteristik Fungsional Kurva PI

Fungsi Keanggotaan :

$$\Pi(x; \beta; \gamma) = \begin{cases} S \left[x; \gamma - \beta, \gamma - \beta / 2, \gamma \right] \rightarrow x \leq \gamma \\ 1 - S \left[x; \gamma, \gamma + \beta / 2, \gamma + \beta \right] \rightarrow x > \gamma \end{cases} \quad \dots \text{Rumus 2.7}$$

2. Kurva BETA

Seperti halnya kurva PI, kurva BETA juga berbentuk lonceng namun lebih rapat. Kurva ini juga didefinisikan dengan 2 parameter, yaitu nilai pada *domain* yang menunjukkan pusat kurva (γ), dan setengah lebar kurva (β) (Kusumadewi dan Purnomo, 2010: 21).



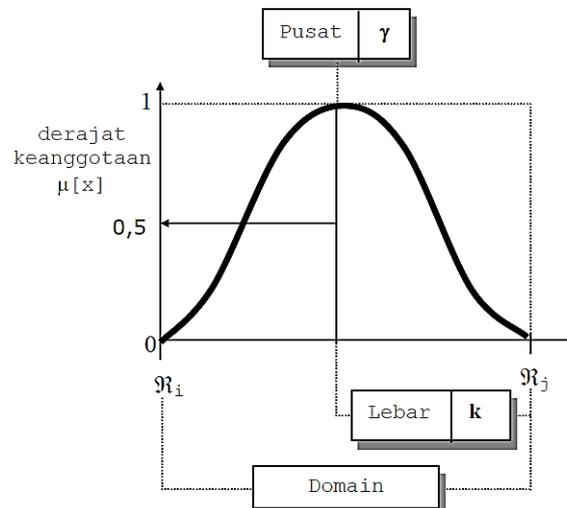
Gambar 2.11 Karakteristik Fungsional Kurva BETA

Fungsi Keanggotaan:

$$B(x; \gamma, \beta) = \frac{1}{1 + \left[\frac{x - \gamma}{\beta} \right]^2} \quad \dots \text{Rumus 2.8}$$

3. Kurva GAUSS

Jika kurva PI dan kurva BETA menggunakan 2 parameter yaitu (γ) dan (β), kurva GAUSS juga menggunakan (γ) untuk menunjukkan nilai *domain* pada pusat kurva, dan (k) yang menunjukkan lebar kurva (Kusumadewi dan Purnomo, 2010: 21).



Gambar 2.12 Karakteristik Fungsional Kurva GAUSS

Fungsi Keanggotaan:

$$G(x; k, \gamma) = e^{-k(\gamma - x)^2} \quad \dots \text{Rumus 2.9}$$

2.1.4.4 Operator Zadeh Untuk Operasi Himpunan *Fuzzy*

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan *fuzzy*. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama *fire strength* atau α -predikat. Menurut Kusumadewi dan Purnomo (2010: 23) ada 3 operator dasar yang diciptakan oleh Zadeh yaitu:

1. Operator *AND*

Operator ini berhubungan dengan operasi interseksi pada himpunan. α -predikat sebagai hasil operasi dengan operator *AND* diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan (Kusumadewi dan Purnomo, 2010: 24).

$$\mu_{A \cap B} = \min(\mu_A[x], \mu_B[y]) \quad \dots \text{Rumus 2.10}$$

2. Operator *OR*

Operator ini berhubungan dengan operasi union pada himpunan. α -predikat sebagai hasil operasi dengan operator *OR* diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan (Kusumadewi dan Purnomo, 2010: 24).

$$\mu_{A \cup B} = \max(\mu_A[x], \mu_B[y]) \quad \dots \text{Rumus 2.11}$$

3. Operator *NOT*

Operator ini berhubungan dengan operasi komplemen pada himpunan. α -predikat sebagai hasil operasi dengan operator *NOT* diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1 (Kusumadewi dan Purnomo, 2010: 25).

$$\mu_{A^c} = 1 - \mu_A[x] \quad \dots \text{Rumus 2.12}$$

2.1.4.5 *Fuzzy Inference System*

Fuzzy Inference System (FIS) dapat dilakukan dengan tiga metode, yaitu dengan metode Mamdani, metode Sugeno dan metode Tsukamoto (Kusumadewi dan Purnomo, 2010: 31-59). Pada penelitian ini, penulis fokus pada satu sistem

inferensi yaitu dengan Metode Mamdani karena mudah di pahami dan cocok pada judul yang di angkat.

2.1.4.5.1 Metode Mamdani

Metode Mamdani sering dikenal dengan nama Metode *Max–Min*. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Menurut Kusumadewi dan Purnomo (2010: 37-42) untuk mendapatkan *output*, diperlukan 4 tahapan yaitu:

1. Pembentukan himpunan *fuzzy*

Pada metode Mamdani, baik variabel *input* maupun variabel *output* dibagi menjadi satu atau lebih himpunan *fuzzy*.

2. Aplikasi fungsi implikasi

Pada metode Mamdani, fungsi implikasi yang digunakan adalah *Min*.

3. Komposisi Aturan

Apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari gabungan antar aturan. Ada tiga metode yang digunakan dalam melakukan inferensi sistem *fuzzy*, yaitu: *max*, *additive* dan *probabilistik OR (probor)*.

- a. Metode *Max (Maximum)*

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikas daerah *fuzzy*, dan mengaplikasikannya ke *output* dengan menggunakan operator *OR (union)*. Jika semua proposisi telah dievaluasi, *Then output* akan berisi suatu himpunan *fuzzy* yang

merefleksikan kontribusi dari tiap-tiap proposisi (Kusumadewi dan Purnomo, 2010: 38). Secara umum dapat dituliskan:

$$\mu_{fs}(x_i) = \max(\mu_{fs}(x_i), \mu_{kf}(x_i)) \quad \dots \text{Rumus 2.13}$$

dengan:

$\mu_{fs}(x_i)$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i

$\mu_{kf}(x_i)$ = nilai keanggotaan konsekuen *fuzzy* aturan ke-i

b. Metode *Additive (Sum)*

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan *bounded-sum* terhadap semua *output* daerah *fuzzy* (Kusumadewi dan Purnomo, 2010: 39). Secara umum dituliskan:

$$\mu_{sf}(x_i) = \min(1, \mu_{sf}(x_i) + \mu_{kf}(x_i)) \quad \dots \text{Rumus 2.14}$$

dengan:

$\mu_{sf}(x_i)$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i

$\mu_{kf}(x_i)$ = nilai keanggotaan konsekuen *fuzzy* aturan ke-i

c. Metode *Probabilistik OR (probor)*

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan *product* terhadap semua *output* daerah *fuzzy*. (Kusumadewi dan Purnomo, 2010: 40) Secara umum dituliskan:

$$\mu_{sf}(x_i) = (\mu_{sf}(x_i) + \mu_{kf}(x_i)) - (\mu_{sf}(x_i) * \mu_{kf}(x_i)) \quad \dots \text{Rumus 2.15}$$

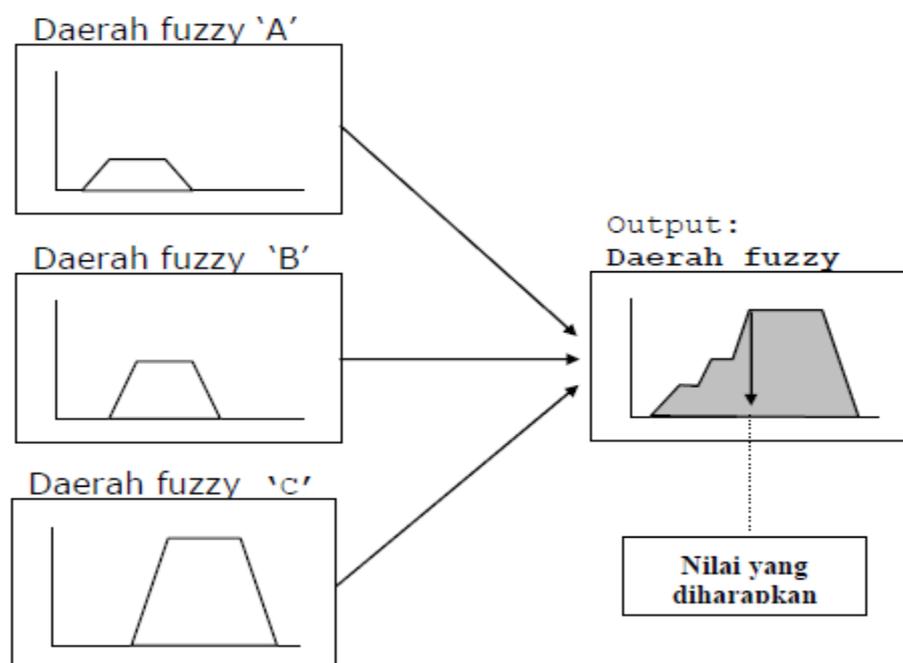
dengan:

$\mu_{sf}(x_i)$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i

$\mu_{kf}(x_i)$ = nilai keanggotaan konsekuen *fuzzy* aturan ke-i

4. Penegasan (*defuzzy*)

Input dari proses *defuzzyfikasi* adalah suatu himpunan *fuzzy* yang diperoleh dari suatu komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai *output*.



Gambar 2.13 Proses *Defuzzyfikasi*

Menurut Kusumadewi dan Purnomo (2010: 41) ada beberapa metode *defuzzyfikasi* pada komposisi aturan MAMDANI, antara lain:

a. Metode Centroid (*Composite Moment*)

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil titik pusat (z^*) daerah *fuzzy* (Kusumadewi dan Purnomo, 2010: 41).

b. Metode Bisektor

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai pada *domain* fuzzy yang memiliki nilai keanggotaan setengah dari jumlah total nilai keanggotaan pada daerah *fuzzy* (Kusumadewi dan Purnomo, 2010: 41).

c. Metode *Mean of Maksimum (MOM)*

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai rata-rata *domain* yang memiliki nilai keanggotaan maksimum (Kusumadewi dan Purnomo, 2010: 41).

d. Metode *Largest of Maximum (LOM)*

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terbesar dari *domain* yang memiliki nilai keanggotaan maksimum (Kusumadewi dan Purnomo, 2010: 41).

e. Metode *Smallest of Maximum (SOM)*

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai terkecil dari *domain* yang memiliki nilai keanggotaan maksimum (Kusumadewi dan Purnomo, 2010: 42).

2.1.4.5.2 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, sehingga metode ini sering juga disebut dengan

Metode TSK. Metode TSK terdiri dari 2 jenis, (Kusumadewi dan Purnomo, 2010:

46) yaitu:

- a. Model *Fuzzy* SUGENO Orde-Nol

Secara umum bentuk model *fuzzy* SUGENO Orde-Nol adalah:

$$\text{IF } (x_1 \text{ is } A_1) \cdot (x_2 \text{ is } A_2) \cdot (x_3 \text{ is } A_3) \cdot \dots \cdot (x_N \text{ is } A_N) \text{ THEN } z=k$$

Dengan A_i adalah himpunan *fuzzy* ke-i sebagai anteseden, dan k adalah suatu konstanta (tegas) sebagai konsekuen.

- b. Model *Fuzzy* SUGENO Orde-Satu

Secara umum bentuk model *fuzzy* SUGENO Orde-Satu adalah:

$$\text{IF } (x_1 \text{ is } A_1) \cdot \dots \cdot (x_N \text{ is } A_N) \text{ THEN } z = p_1 \cdot x_1 + \dots + p_N \cdot x_N + q$$

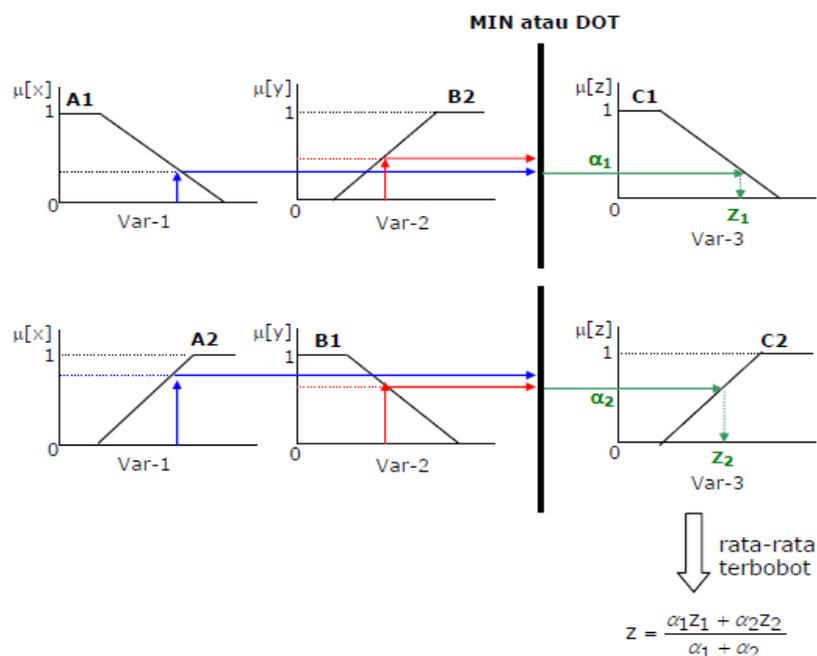
Dengan A_i adalah himpunan *fuzzy* ke-i sebagai anteseden, dan p_i adalah suatu konstanta (tegas) ke-i dan q juga merupakan konstanta dalam konsekuen.

Apabila komposisi aturan menggunakan metode SUGENO, maka defuzzifikasi dilakukan dengan cara mencari nilai rata-ratanya.

2.1.4.5.3 Metode Tsukamoto

Metode Tsukamoto merupakan perluasan dari penalaran monoton, 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 (Kusumadewi dan Purnomo, 2010: 31).



Gambar 2.14 Inferensi dengan Menggunakan Metode Tsukamoto

2.2 Variabel

Variabel adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi tentang hal tersebut, kemudian ditarik kesimpulannya (Sugiyono, 2014: 38).

Secara teoritis variabel dapat didefinisikan sebagai atribut seseorang, atau obyek, yang mempunyai “variasi” antara satu orang dengan yang lain atau obyek dengan obyek yang lain (Hatch dan Farhady, 1981 dalam Sugiyono, 2014: 38). Variabel yang digunakan pada penelitian ini adalah menentukan tingkat kebugaran jasmani pada siswa Sekolah Dasar (SD).

2.2.1 Pengertian Kebugaran Jasmani

Kebugaran Jasmani (*Physical Fitness*) secara umum dapat diartikan sebagai kesanggupan dan kemampuan tubuh untuk menjalankan tugas/kerja yang dibebankan kepadanya tanpa mengalami kelelahan yang berlebihan sehingga ia masih mempunyai cadangan tenaga untuk menjalankan tugas berikutnya (Isnaini dan Sabarini, 2010: 60).

Berbagai cara dilakukan seseorang untuk menjaga kesehatan dan kebugaran fisiknya. Di antaranya dengan melakukan olahraga secara teratur, mengonsumsi makanan dan minuman yang bergizi, ditunjang oleh istirahat yang cukup (Isnaini dan Sabarini, 2010: 44).

2.2.2 Bentuk-bentuk Latihan Kebugaran Jasmani

Guna meningkatkan taraf kebugaran jasmani seseorang diperlukan latihan yang berkenaan dengan kondisi fisik. Mengenai unsur-unsur kebugaran jasmani yang berkaitan dengan keterampilan olahraga, yaitu kecepatan dan unsur kebugaran jasmani yang berkaitan dengan kesehatan yaitu kekuatan, daya tahan, dan kelentukan (Isnaini dan Sabarini, 2010: 44).

1. Kekuatan

Kekuatan adalah tenaga kontraksi otot yang dicapai dalam sekali usaha maksimal. Usaha maksimal ini dilakukan oleh otot untuk mengatasi suatu tahanan. Kekuatan merupakan unsur yang sangat penting dalam aktivitas olahraga karena kekuatan merupakan daya penggerak dan pencegah cedera.

Selain itu, kekuatan memainkan peranan penting dalam komponen-komponen dan kemampuan fisik yang lain misalnya *power*, kelincahan, dan kecepatan. Dengan demikian, kekuatan merupakan faktor utama untuk menciptakan prestasi optimal.

Bentuk-bentuk latihan kekuatan seperti: latihan *Push Up* untuk melatih kekuatan otot lengan dan bahu, latihan *Sit Up* untuk melatih kekuatan otot perut, latihan *Back Lift* untuk melatih kekuatan otot punggung, dan lain-lain (Isnaini dan Sabarini, 2010: 44-45).

2. Kecepatan

Kecepatan dapat diartikan sebagai kemampuan berpindah tempat dalam waktu sesingkat-singkatnya. Sedangkan menurut (Biyakto *dalam* Isnaini dan Sabarini, 2010: 47) mengemukakan kecepatan adalah kemampuan untuk melakukan suatu gerakan dalam periode waktu yang singkat. Kecepatan sangat diperlukan bagi pelari jarak pendek atau bagi seorang pemain sepak bola.

Menurut Isnaini dan Sabarini (2010: 47) berikut ini merupakan faktor-faktor yang mempengaruhi kecepatan.

- a. Keturunan
- b. Waktu reaksi
- c. Kemampuan untuk menahan tahanan luar
- d. Teknik
- e. Konsentrasi dan kemauan
- f. Elastisitas otot

Kecepatan dapat dibagi menjadi tiga macam, yaitu kecepatan *sprint*, kecepatan reaksi, dan kecepatan bergerak (Isnaini dan Sabarini, 2010: 47).

3. Daya Tahan (*Endurance*)

Endurance atau daya tahan adalah kemampuan seseorang melaksanakan gerak dengan seluruh tubuhnya dalam waktu yang cukup lama dan dengan tempo sedang sampai cepat tanpa mengalami rasa sakit dan kelelahan berat. Kemampuan otot untuk melakukan kerja terus-menerus adalah sangat penting dalam aktivitas olahraga karena secara tidak langsung merupakan daya untuk dapat mengatasi kelelahan otot.

Menurut Isnaini dan Sabarini (2010: 48) latihan daya tahan dapat dibagi menjadi dua macam.

- a. Daya tahan otot setempat (*Muscular Endurance*) adalah daya tahan yang menunjukkan kemampuan otot atau sekelompok otot dalam melaksanakan tugasnya dengan waktu yang cukup lama. Contoh: latihan *weight training*/latihan berbeban.
- b. *Cardiorespiratory Endurance* yaitu latihan yang bertujuan meningkatkan kemampuan seluruh tubuh untuk selalu bergerak dalam tempo sedang sampai cepat yang cukup lama. Contoh: berlari, berenang, bersepeda. Daya tahan kardiovaskuler merupakan faktor utama dalam kebugaran jasmani. Pengukuran yang paling objektif dilakukan dengan mengukur kemampuan pengambilan denyut jantung maksimal (VO_2 maksimal).

4. Kelentukan (*Fleksibilitas*)

Kelentukan adalah keleluasaan gerakan terutama pada otot persendian.

Tujuan latihan kelentukan atau fleksibilitas adalah agar otot-otot sendi tidak kaku dan dapat bergerak dengan leluasa tanpa ada gangguan yang berarti.

Bentuk-bentuk latihan kelentukan seperti: latihan kelentukan sendi bahu, latihan kelentukan otot pinggang, latihan kelentukan tongok (Isnaini dan Sabarini, 2010: 49-50).

2.2.3 Denyut Jantung dan Nadi dalam Olahraga

Denyut jantung merupakan jumlah denyutan jantung per satuan waktu, biasanya dihitung per menit dan direpresentasikan sebagai *beats per minute* (*bpm*). Sedangkan denyut nadi merupakan denyutan arteri dari gelombang darah yang mengalir melalui pembuluh darah sebagai akibat dari denyutan jantung. Sehingga keduanya saling berkaitan (<http://zonapelatih.net/2016/01/pentingnya-paham-denyut-jantung-dan-nadi-latihan.html> diakses pada 20 November 2016).

Denyut jantung setiap orang berbeda-beda. Faktor yang mempengaruhi jumlah denyut jantung adalah umur/usia, aktivitas fisik atau kebugaran kardiovaskularnya, waktu perhitungan, suhu udara sekitar, posisi tubuh (berbaring atau berdiri), tingkat emosi seseorang, fluktuasi hormon, asupan kafein, merokok dan obat yang sedang dikonsumsi. Jumlah denyut jantung normal pada orang dewasa yang sehat pada waktu istirahat berada pada kisaran 60-100 denyut per menit (*bpm*). Berikut ini merupakan tabel denyut nadi normal pada manusia berdasarkan usia.

Tabel 2.1 Denyut Nadi Normal Pada Manusia Berdasarkan Usia

UMUR	DENYUT NADI
Bayi 0-1 bulan	70-190 denyut <i>bpm</i> (per menit)
Bayi 1-11 bulan	80-160 denyut <i>bpm</i>
Anak-anak 1-2 tahun	80-130 denyut <i>bpm</i>
Anak-anak 3-4 tahun	80-120 denyut <i>bpm</i>
Anak-anak 5- 6 tahun	75-115 denyut <i>bpm</i>
Anak-anak 7-9 tahun	70-110 denyut <i>bpm</i>
Anak-anak , lebih tua serta orang dewasa	60-100 denyut <i>bpm</i>
Khusus atlet terlatih	40-60 denyut <i>bpm</i>

Sumber: Zonaplatih (2016: 10)

Denyut nadi dalam bidang olahraga dikenal dengan istilah denyut nadi maksimal (*Maximal Heart Rate*), yaitu denyut nadi yang didapat ketika melakukan aktivitas atau latihan maksimal. Denyut nadi maksimal merupakan batas kemampuan seseorang untuk melakukan aktivitas secara normal. Apabila seseorang melakukan suatu aktivitas atau latihan melebihi batas denyut nadi maksimal (*DN Max*) maka segera beristirahat, karena bisa berbahaya bagi jantung dan juga organ tubuh lainnya. Cara untuk mengetahui denyut nadi maksimal adalah dengan menggunakan rumus $(220 - \text{usia})$ (<http://zonaplatih.net/2016/01/pentingnya-paham-denyut-jantung-dan-nadi-latihan.html> diakses pada 20 November 2016).

2.2.4 Indikator Penentuan Tingkat Kebugaran Jasmani Siswa

Berikut ini merupakan indikator atau parameter yang biasa digunakan dalam menentukan tingkat kebugaran jasmani siswa:

1. Baring Duduk (*Sit Up*)

Dengan melakukan tes olahraga baring duduk, siswa melatih kekuatan otot perut yang termasuk kedalam jenis latihan kebugaran jasmani (Kurniadi dan Prapanca, 2010:23).

2. Jongkok Lompat (*Squat Jump*)

Dengan melakukan tes olahraga Jongkok lompat, siswa melatih kekuatan otot paha dan tungkai kaki yang termasuk kedalam jenis latihan kebugaran jasmani (Kurniadi dan Prapanca, 2010: 24).

3. Lari (*Run*)

Dengan melakukan tes olahraga lari yang dilaksanakan dengan mengelilingi lapangan yang ada di sekolah, siswa melatih kekuatan daya tahan dan juga meningkatkan kesegaran jasmani (Kurniadi dan Prapanca, 2010: 21)

2.3 Software Pendukung

Pada penelitian ini menggunakan bantuan MATLAB untuk simulasi grafik dan *rule* karena pada MATLAB menyediakan perkakas untuk membuat *Fuzzy Inference System (FIS)* yaitu *Fuzzy Logic Toolbox* yang didalamnya terdapat *Graphical User Interface (GUI)* untuk merancang *FIS*.

2.3.1 Pengertian MATLAB

MATLAB adalah bahasa pemrograman tingkat tinggi di mana arti perintah dan fungsi-fungsinya bisa dimengerti dengan mudah, meskipun bagi seorang pemula. Hal itu karena di dalam MATLAB, masalah dan solusi bisa diekspresikan dalam notasi-notasi matematis yang biasa dipakai. MATLAB singkatan dari *matrix laboratory*. Dalam dunia akademis, ia telah menjadi alat bantu standar instruksional dalam kuliah-kuliah pengenalan dan tingkat lanjut bidang matematik, teknik dan sains. Spektrum penggunaan MATLAB yang luas ini dimungkinkan karena MATLAB telah melengkapi diri dengan berbagai *toolbox*. Sebuah *toolbox* dalam MATLAB adalah koleksi berbagai fungsi MATLAB (*M-Files*, yaitu *file* berekstensi *.m*), yang merupakan perluasan MATLAB untuk memecahkan masalah-masalah khusus pada bidang tertentu. Oleh karenanya, dengan memakai *toolbox* dalam MATLAB, para pengguna bisa belajar dan menerapkan berbagai *specialized technology*. Beberapa bidang sudah tersedia *toolbox*-nya dalam MATLAB, meliputi *fuzzy logic*, *neural network* (jaringan syaraf tiruan) *control system* (sistem kontrol), *signal processing* (pengolahan sinyal) dan *wavelet* (Naba, 2009: 39).

2.3.2 Fuzzy Logic Toolbox

Fuzzy Logic Toolbox adalah sekumpulan *tool* yang akan membantu peneliti merancang sistem *fuzzy* untuk diaplikasikan dalam berbagai bidang, seperti *automatic control*, *signal processing*, *identification system*, *pattern recognition*,

time series prediction, data mining, dan bahkan financial applications. Dengan *Fuzzy Logic Toolbox*, peneliti bisa membuat atau mengedit *FIS* dalam lingkungan kerja MATLAB. *Fuzzy Logic Toolbox* sangat *user friendly*, memungkinkan pengguna berkreasi dengan bebas dalam rancang bangun *FIS* (Naba, 2009: 79).

Menurut Naba (2009: 79) semua *tool* dalam *Fuzzy Logic Toolbox* dikelompokkan menjadi tiga kategori yaitu:

1. *Command Lines*. Fungsi-fungsi *command lines Fuzzy Logic Toolbox* adalah fungsi-fungsi yang dapat dieksekusi langsung dari MATLAB *Prompt*. Sebagian besar fungsi ini ditulis dalam bentuk *M-Files*.
2. *Graphical User Interface (GUI)*. *GUI* memungkinkan pengguna mengakses banyak fungsi-fungsi yang tersedia dalam *Fuzzy Logic Toolbox*. Sebenarnya *Fuzzy Logic Toolbox* lebih banyak mengandalkan *GUI* dalam membantu penyelesaian kerja dalam rancang bangun *FIS*, meskipun bisa dilakukan dari *command lines*. *GUI* sangat cocok untuk pemula, sementara *command lines* ditujukan untuk pemakai yang sudah berpengalaman.
3. *Simulink Block*. Kategori ketiga adalah *tool* dalam bentuk blok-blok Simulink. Sebenarnya *tool* kategori ketiga ini dirancang khusus untuk aplikasi-aplikasi *FIS* dalam lingkungan Simulink.

2.3.3 Graphical User Interface (GUI)

GUI memungkinkan pengguna mengakses banyak fungsi-fungsi yang tersedia dalam *Fuzzy Logic Toolbox*. Sebenarnya *Fuzzy Logic Toolbox* lebih banyak mengandalkan *GUI* dalam membantu penyelesaian kerja dalam rancang

bangun *FIS*, meskipun bisa dilakukan dari *command lines*. *GUI* sangat cocok untuk pemula, sementara *command lines* ditujukan untuk pemakai yang sudah berpengalaman (Naba, 2009: 80).

Menurut Naba (2009: 80) terdapat 5 *GUI tools* yang dapat digunakan untuk meng-*edit*, mengamati, dan membangun model *fuzzy* yaitu:

1. *Fuzzy Inference System (FIS) Editor*

GUI ini yang berfungsi untuk mengedit model *fuzzy* yang dibuat. *FIS Editor* dapat dipanggil dengan mengetikkan tulisan “*fuzzy*” pada *Command window* (Naba, 2009: 80).

2. *Membership Function Editor (MFE)*

GUI ini yang berfungsi untuk merancang atau membuat fungsi keanggotaan yang akan digunakan dalam model *fuzzy*. Terdapat beberapa fungsi keanggotaan yang dapat digunakan, antara lain fungsi keanggotaan segitiga dan Gauss. *Editor* ini dapat dipanggil dari *FIS* (Naba, 2009: 80).

3. *Rule Editor*

GUI ini yang berfungsi menyusun aturan Jika-Maka berdasarkan pengetahuan maupun aturan-aturan yang kemudian akan digunakan sebagai penalaran *fuzzy* yang merupakan inti dari model *fuzzy*. *Rule Editor* dapat dipanggil dengan cara pilih *view*→ *Edit Rules* (Naba, 2009: 80).

4. *Rule Viewer*

GUI ini yang berfungsi untuk menampilkan penalaran dari model *fuzzy* secara keseluruhan dalam bentuk model 2 dimensi. *Rule Viewer* dapat dipanggil dengan memilih menu *view*→*view rule* (Naba, 2009: 80).

5. *Surface Viewer*

GUI ini yang berfungsi untuk menampilkan penalaran dari model *fuzzy* dalam bentuk 3 dimensi. *Surface Viewer* dapat dipanggil dengan memilih menu *view*→*view Surface* (Naba, 2009: 80).

2.4 Penelitian Terdahulu

Ada beberapa riset yang telah dilakukan yang berkaitan dengan penelitian ini diantaranya adalah:

1. (Suharjana, 2012), Pengukuran Tingkat Kebugaran Jasmani Siswa SD di Kabupaten Banyumas Jawa Tengah Sebagai Dasar Evaluasi Proses Pembelajaran Pendidikan Jasmani. Populasi dalam penelitian ini adalah siswa SD Kelas Atas di Kabupaten Banyumas, Provinsi Jawa Tengah. Sampel yang digunakan adalah sebanyak 353 siswa. Teknik yang digunakan untuk mengambil sampel adalah dengan teknik *multistage random sampling* (Nasir, 2003). Untuk pengambilan data penelitian peneliti dibantu oleh guru penjasorkes SD di wilayah Banyumas, yang berjumlah 20 orang. Untuk mengetahui kebugaran jasmani, digunakan instrument tes baku, yaitu Tes Kebugaran Jasmani Indonesia (TKJI), yang terdiri atas: (1) Tes lari 40 m, (2) Gantung Siku Tekuk, (3) Baring Duduk 30 detik, (4) Loncat Tegak, (5) lari 600 m. Data statistik dianalisis dengan analisis diskriptif persentase dengan mode tabulasi silang. Untuk mengetahui status kebugaran jasmani dilakukan dengan cara data tes untuk setiap butir tes dicocokkan dengan

tabel nilai TKJI, selanjutnya jumlah nilai TKJI tersebut dicocokkan dengan tabel normal TKJI untuk membuat klasifikasi. Nilai dalam bentuk klasifikasi tersebut selanjutnya dianalisis dengan analisis diskriptif.

2. (Nonce, 2014), Survey Tingkat Kesegaran Jasmani Siswa Kelas V Sekolah Dasar Negeri 25 Palu Kecamatan Palu Timur. Penelitian ini dilaksanakan di Sekolah Dasar Negeri 25 Palu Kecamatan Palu Timur. Dalam Penelitian ini yang menjadi subjek penelitian adalah siswa kelas V_B sebanyak 40 orang yang terdiri dari laki-laki 21 orang dan perempuan 19 orang. Data dikumpulkan dengan melaksanakan berbagai macam tes pada siswa. Dalam tahap pelaksanaan ini penulis membagi peserta tes menjadi 5 kelompok yang terdiri dari kelompok 1 (lari 40 meter), kelompok 2 (baring duduk) kelompok 3 (loncat tegak) kelompok 4 (lari 600 meter) dan kelompok 5 (gantung siku tekuk). Setelah seluruh data terkumpul selanjutnya data dianalisis dengan analisis statistik, karena data yang terkumpul berupa angka-angka. Selanjutnya hasil dari penelitian menunjukkan bahwa: 1. Tingkat kesegaran jasmani siswa SDN 25 pali yang masuk dalam kategori baik sekali ada 2 siswa atau sebanyak 5%; 2. Tingkat kesegaran jasmani siswa SDN 25 Palu yang masuk dalam kategori baik ada 16 siswa atau sebanyak 40%; 3. Tingkat kesegaran jasmani siswa SDN 25 Palu yang masuk dalam kategori sedang ada 15 siswa atau sebanyak 37.5%; 4. Tingkat kesegaran jasmani siswa SDN 25 Palu yang masuk dalam kategori kurang ada 4 siswa atau sebanyak 10%; 5. Tingkat kesegaran jasmani siswa SDN

25 pali yang masuk dalam kategori kurang sekali ada 3 siswa atau sebanyak 7,5%.

3. (Sutisna & Basjaruddin, 2015), Sistem Pendukung Keputusan Pemilihan Pekerjaan Menggunakan Metode *Fuzzy Mamdani* Studi Kasus: Amik BSI Tasikmalaya. Pada penelitian ini, digunakan logika *fuzzy mamdani* untuk mendapatkan keluaran berupa keputusan pekerjaan yang dipilih tersebut sesuai atau tidak dan menghasilkan rekomendasi berupa pekerjaan yang paling tepat dari sistem. Sistem pendukung keputusan ini digunakan untuk Mahasiswa AMIK BSI Tasikmalaya dengan menggunakan 3 variabel *input* yaitu IPK, Gaji dan Umur. Dari hasil pengujian Sistem Pendukung Keputusan Pemilihan Pekerjaan dapat diambil kesimpulan bahwa sistem dapat menangani proses pemilihan pekerjaan berdasarkan data yang *diinputkan* dan rekomendasi yang tepat. Sistem Pendukung Keputusan ini menghasilkan rekomendasi pekerjaan yang sesuai dengan data yang dimiliki mahasiswa. Hasil ini lebih baik dibandingkan dengan hanya menebak nebak pekerjaan apa yang tepat dan perkiraan bisa diterima atau tidak pada sebuah perusahaan tersebut.
4. (Sumiati & Shodik, 2013), Sistem Pendukung Keputusan Dalam Menentukan Penilaian Kinerja Dosen dengan Metode *Fuzzy Database Mode Mamdani*. Penelitian ini akan menganalisis penilaian kinerja dosen berbasis sistem komputasi cerdas menggunakan *Fuzzy Inference System* dengan metode Mamdani. Fokus penelitian lebih menitik beratkan pada proses penilaian kinerja dosen. Penilaian ini berdasarkan penilaian kinerja dosen,

yakni tridarma dan aktivitas internal. Sistem penunjang keputusan ini membantu dan memberikan alternatif dalam melakukan penilaian setiap dosen, melakukan perubahan kriteria, Hal ini berguna untuk memudahkan pengambil keputusan yang terkait dengan masalah penilaian kinerja dosen, sehingga akan di dapatkan dosen yang paling layak diberi reward atau penghargaan dalam penilaian kinerja meliputi: data primer, dan data data sekunder. Data Primer terdiri dari ketepatan dalam penilaian, persiapan dalam materi, kemampuan dalam menyajikan materi, memberikan contoh yang berkaitan dengan materi yang diajarkan. Sedangkan data sekunder terdiri dari tridarma perguruan tinggi meliputi pengajaran, penelitian dan pengabdian masyarakat. Sistem Penilaian kinerja dosen tetap Universitas Serang Raya dirancang dan dibangun dengan menerapkan logika *Fuzzy Inference System* dengan metode Mamdani yang disesuaikan dengan kondisi dari ketentuan penilaian kinerja dosen tetap Universitas Serang Raya yang ada. Hasil atau keluaran dari penelitian ini adalah sebuah keputusan dalam penilaian kinerja dosen di Universitas Serang Raya.

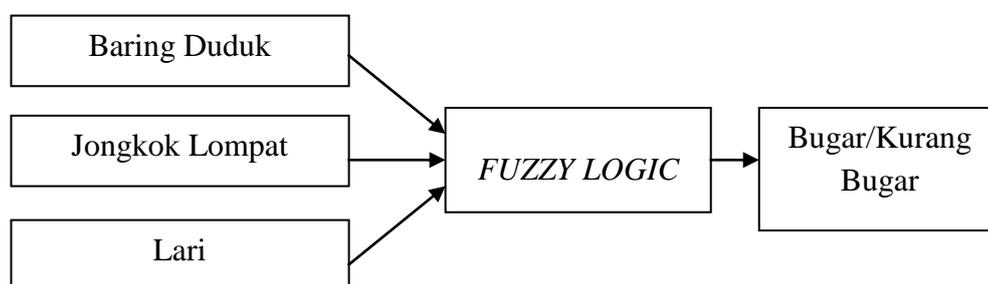
5. (Dzulhaq & Imani, 2015), Sistem Pendukung Keputusan Pemilihan Konsentrasi Jurusan Menggunakan *Fuzzy Inference Sistem* Metode Mamdani. Pemilihan jurusan yang tepat di perguruan tinggi dapat membuat suatu perbedaan besar, karena siswa perlu menemukan jurusan yang cocok dengan ketertarikan, kemampuan dan kecenderungan kemampuan mereka. Siswa mempunyai kemampuan berpikir yang berbeda-beda dan minat yang berbeda-beda pula untuk berbuat sesuatu. Siswa yang ingin mendaftar pada

STMIK Bina Sarana Global memiliki beberapa pilihan jurusan serta konsentrasinya. Nilai TPA dan nilai Minat menjadi bahan pertimbangan dalam memilih jurusan serta konsentrasi yang tepat. Implementasi menggunakan bahasa pemrograman Matlab. Metode penelitian sistem yang digunakan adalah *Fuzzy Inference System* metode mamdani. Metode mamdani paling sesuai dengan naluri manusia, bekerja berdasarkan kaidah linguistik dan memiliki algoritma *fuzzy* yang menyediakan aproksimasi untuk dimasuki analisa matematik. Data yang diolah dalam metode mamdani yaitu data nilai TPA dan nilai Bakat. Data tersebut diproses melalui tahap-tahap perhitungan logika *fuzzy* dan memberikan keluaran dari sistem berupa rekomendasi jurusan yang disarankan untuk diambil oleh siswa yang bersangkutan. Sistem ini menampilkan sebuah hasil keputusan untuk memberikan rekomendasi pemilihan jurusan serta konsentrasi khususnya yang ada pada STMIK Bina Sarana Global. Sistem ini sangat bermanfaat dalam membantu siswa memilih jurusan serta konsentrasi karena hasil yang diperoleh telah melalui perhitungan logika *fuzzy* dengan data-data yang valid. Penelitian ini menghasilkan sebuah sistem yang dapat membantu siswa yang mendaftar pada STMIK Bina Sarana Global dalam memutuskan jurusan serta konsentrasi.

2.5 Kerangka Pemikiran

Kerangka berfikir adalah model konseptual tentang bagaimana teori hubungan dengan berbagai faktor yang telah didefinisikan sebagai masalah yang penting (Sugiyono, 2014: 60).

Penelitian ini melalui tahap-tahap kegiatan yang tertuang dalam kerangka berfikir yang meliputi metode pengumpulan data dari *input* yaitu data-data yang akan di seleksi ke dalam logika *fuzzy* kemudian *outputnya* hasil dari data-data yang telah di *input* yaitu menentukan tingkat kebugaran jasmani pada siswa sekolah dasar tersebut. Kerangka pemikiran pada penelitian ini dapat dilihat pada Gambar 2.15.



Gambar 2.15 Kerangka Berfikir

Untuk menentukan kebugaran jasmani siswa, ditetapkan kriteria-kriteria yang akan digunakan sebagai variabel *input* yang akan di implementasikan kedalam logika *fuzzy*, adapun kriteria yang ditetapkan yaitu baring duduk, jongkok lompat, dan lari. Dari data yang diperoleh dilakukan proses implementasi dengan menggunakan logika *fuzzy* metode Mamdani. Logika *fuzzy* metode Mamdani digunakan untuk mengolah data sehingga dapat menghasilkan keputusan tepat cepat dan efisien dengan *output* keputusan bugar/kurang bugar.