

BAB II **KAJIAN PUSTAKA**

2.1. Teori Dasar

2.1.1. *Artificial Intelligence*

Kecerdasan Buatan (*Artificial Intelligence*) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia. Aplikasi atau program kecerdasan buatan dapat ditulis dalam semua bahasa komputer, baik dalam bahasa C, Pascal, *Basic*, dan bahasa pemrograman lainnya (Dahria, 2008). Kecerdasan Buatan (*artificial intelligence*) merupakan inovasi baru di bidang ilmu pengetahuan. Mulai ada sejak muncul komputer modern, yakni pada 1940 dan 1950. Kemampuan mesin elektronika baru menyimpan sejumlah besar info, memproses dengan kecepatan sangat tinggi menandingi kemampuan manusia. Ilmu pengetahuan komputer ini khusus ditujukan dalam perancangan otomatisasi tingkah laku cerdas dalam sistem kecerdasan komputer. Pada sistem ini memperlihatkan sifat-sifat khas yang dihubungkan dengan kecerdasan dalam kelakuan yang sepenuhnya dapat menirukan beberapa fungsi otak manusia, seperti pengertian bahasa, pengetahuan, pemikiran, pemecahan, dan masalah.

Pentingnya kecerdasan buatan menjadi nyata bagi negara-negara yang berperan sejak tahun 1970. Para pemimpin negara yang mengakui potensinya

kecerdasan buatan mengharap mendapat persetujuan jangka panjang untuk sumber-sumber yang memerlukan dana intensif. Jepang adalah yang pertama kali melakukan itu. Negara ini mengembangkan program yang sangat berambisi dalam penelitian kecerdasan buatan. Sebagai bidang ilmu pengetahuan komputer, kecerdasan buatan sebenarnya sudah mulai diselidiki pada 1930-an dan 1940-an. Pada saat itu, banyak cendekiawan mengembangkan ide-ide baru mengenai komputasi.

Logika matematika menjadi bidang aktif dari penyelidikan kecerdasan buatan, karena sistem logika deduktif telah berhasil diimplementasikan dalam program-program komputer. Seorang ahli matematika bernama Alan Turing, yang memiliki sumbangan besar dalam pengembangan teori kemampuan penghitungan (*computability*), mengusulkan tes untuk melihat bisa atau tidaknya mesin memberikan respon terhadap seangkaian pertanyaan (agar mesin dapat dikatakan cerdas). Uji yang dilakukan adalah dengan mengukur kinerja (*performance*) mesin cerdas. Uji Alan Turing menjadi dasar bagi banyak strategi yang digunakan dengan menilai program-program kecerdasan buatan. Pada awalnya, kecerdasan buatan hanya ada di universitas-universitas dan laboratorium penelitian, serta hanya sedikit produk yang dihasilkan dan dikembangkan. Menjelang akhir 1970-an dan 1980-an, mulai dikembangkan secara penuh dan hasilnya berangsur-angsur dipublikasikan di khalayak umum. Permasalahan di dalam kecerdasan buatan akan selalu bertambah dan berkembang seiring dengan laju perkembangan zaman menuju arah globalisasi dalam setiap aspek kehidupan manusia, yang membawa persoalan-persoalan yang semakin beragam pula.

Program kecerdasan buatan lebih sederhana dalam pengoperasiannya, sehingga banyak membantu pemakai. Program konvensional dijalankan secara prosedural dan kaku, rangkaian tahap solusinya sudah didefinisikan secara tepat oleh pemrogramnya.

Sebaliknya, pada program kecerdasan buatan untuk mendapatkan solusi yang memuaskan dilakukan pendekatan *trial and error*, mirip seperti apa yang dilakukan oleh manusia.

Kecerdasan buatan atau *Artificial Intelligence* merujuk pada mesin yang mampu berpikir, menimbang tindakan yang diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia (Sutojo, Mulyanto, & Suhartono, 2011).

Menurut John McCarthy, 1956, AI: untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Cerdas, berarti memiliki pengetahuan ditambah pengalaman, penalaran (bagaimana membuat keputusan dan mengambil tindakan) moral yang baik.

Menurut Rich and Knight, 1991: "Kecerdasan buatan (*artificial intelligence*) merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia".

Menurut H.A.Simon, 1987: "Kecerdasan buatan (*artificial intelligence*) merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan hal yang dalam pandangan manusia adalah cerdas".

Menurut Encyclopedia Britannica: “Kecerdasan buatan (*artificial intelligence*) merupakan cabang ilmu komputer yang dalam merepresentasikan pengetahuan lebih banyak menggunakan bentuk simbol-simbol daripada bilangan, dan memproses informasi kecerdasan berdasarkan metode *heuristic* atau berdasarkan sejumlah aturan”(Sutojo et al., 2011).

Tujuan dari kecerdasan buatan menurut Winston dan Prendergast:

- a. Membuat mesin menjadi lebih pintar (tujuan utama)
- b. Memahami apa itu kecerdasan (tujuan ilmiah)
- c. Membuat mesin lebih bermanfaat (tujuan *entrepreneurial*)

Dua bagian utama yang dibutuhkan untuk aplikasi kecerdasan buatan adalah

- a. Basis Pengetahuan (*Knowledge Base*) berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
- b. Motor Inferensi (*Inference Engine*) adalah kemampuan menarik kesimpulan berdasarkan pengalaman.

Sub disiplin ilmu dalam *artificial intelligence* diantaranya meliputi:

1. Sistem Pakar (*Expert System*)

Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah. Sistem pakar akan memberikan pemecahan suatu masalah yang didapat dari dialog dengan pengguna.

2. Pengolahan Bahasa Alami (*Natural Language Processing*)

Pengolahan bahasa alami adalah pembuatan program yang memiliki kemampuan untuk memahami bahasa manusia. Pada prinsipnya bahasa alami

adalah suatu bentuk representasi dari suatu pesan yang ingin dikomunikasikan antarmanusia.

3. Pengenalan Ucapan (*Speech Recognition*)

Pengenalan ucapan adalah suatu pengembangan teknik dan sistem yang memungkinkan komputer untuk menerima masukan berupa kata yang diucapkan.

4. Robotika dan sistem sensor (*Robotics and Sensory Systems*)

Robotika adalah ilmu pengetahuan dan teknologi rekayasa robot dan desain, manufaktur, aplikasi dan disposisi struktural.

5. *Computer Vision*

Computer Vision adalah suatu cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali objek yang diamati atau diobservasi.

6. *Intelligent Computer-Aided Instruction*

Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.

7. *Game Playing*

Permainan dilakukan dengan menggunakan sekumpulan aturan, pencarian ruang, teknik untuk menentukan alternatif dalam menyimak problema ruang merupakan sesuatu yang rumit, teknik tersebut disebut dengan *Heuristic* dan permainan merupakan bidang yang menarik dalam studi *heuristic*.

Unsur-unsur dalam pokok *Soft Computing*, adalah

1. Sistem *Fuzzy* (mengakomodasi ketidaktepatan)
2. Jaringan Syaraf (menggunakan pembelajaran)
3. *Probabilistic Reasoning* (mengakomodasi ketidakpastian)

4. *Evolutionary Computing* (Optimasi)

2.1.2. Sistem Pakar

Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah. Sistem pakar akan memberikan pemecahan suatu masalah yang didapat dari dialog dengan pengguna. Dengan bantuan sistem pakar seseorang yang bukan pakar/ahli dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar (Sutojo et al., 2011).

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikannya, diantaranya:

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang 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. Integrasi sistem pakar dengan sistem komputer lain membuat sistem lebih efektif dan mencakup lebih banyak aplikasi.

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 merespons 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, terdapat beberapa kekurangan yang ada pada sistem pakar, diantaranya:

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

Yang menjadi ciri dari sistem pakar adalah sebagai berikut:

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 kaidah/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, yaitu:

1. Kepakaran (*Expertise*)
2. Pakar (*Expert*)
3. Pemindahan Kepakaran (*Transferring Expertise*)
4. Inferensi (*Inferencing*)
5. Aturan-aturan (*Rule*)
6. Kemampuan menjelaskan (*Explanation Capability*)

2.1.3. Jaringan Saraf Tiruan

Jaringan saraf tiruan adalah paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia. Elemen kunci dari 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 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 et al., 2011).

Kelebihan-kelebihan yang diberikan oleh Jaringan Saraf Tiruan antara lain:

1. Belajar *Adaptive*: kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengalaman awal.
2. *Self-Organization*: sebuah Jaringan Saraf Tiruan dapat membuat organisasi sendiri atau representasi dari informasi yang diterimanya selama waktu belajar.
3. *Real time operation*: perhitungan Jaringan Saraf Tiruan dapat dilakukan secara paralel sehingga perangkat keras yang dirancang dan diproduksi secara khusus dapat mengambil keuntungan dari kemampuan ini.

Selain mempunyai kelebihan, Jaringan Saraf Tiruan juga mempunyai kelemahan-kelemahan berikut:

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

Ada beberapa arsitektur jaringan saraf, antara lain:

- a. Jaringan dengan lapisan tunggal (*single layer net*), yang hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima

input kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi.

- b. Jaringan dengan banyak lapisan (*multilayer net*), memiliki satu atau lebih lapisan yang terletak diantara lapisan *input* dan lapisan *output*.
- c. Jaringan dengan lapisan kompetitif (*competitive layer net*), umumnya hubungan antar neuron pada lapisan kompetitif ini tidak diperlihatkan pada diagram arsitektur.

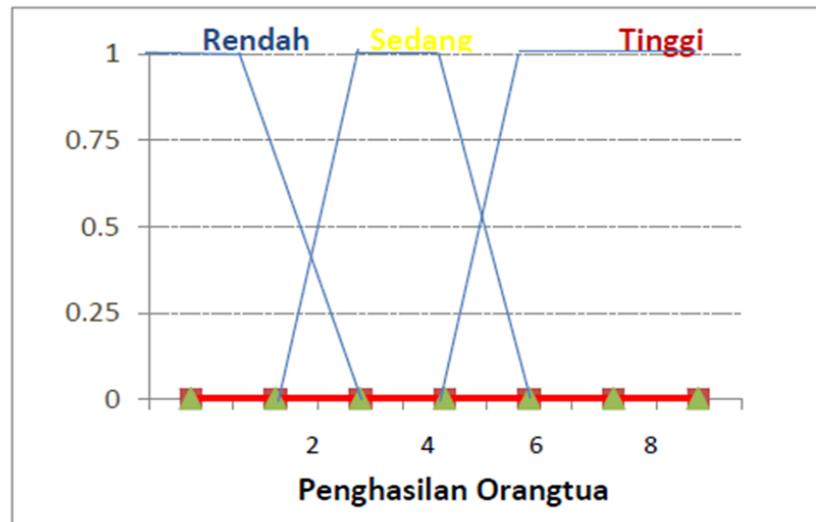
2.1.4. Logika Fuzzy

Logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang cocok untuk diimplementasikan pada sistem, dimulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan PC, multi channel atau *workstation* berbasis akuisisi data, dan sistem control (Setiadji, 2012). Logika *fuzzy* juga merupakan peningkatan dari logika *boolean* yang mengenalkan konsep kebenaran sebagian. Dimana logika klasik menyatakan bahwa segala hal dapat diekspresikan dalam istilah *binary* (0 atau 1, hitam atau putih, ya atau tidak), logika *fuzzy* menggantikan kebenaran *boolean* dengan tingkat kebenaran. Orang yang belum pernah mengenal logika *fuzzy* pasti akan mengira bahwa logika *fuzzy* adalah sesuatu yang amat rumit dan tidak menyenangkan. Kelebihan dari teori logika *fuzzy* adalah kemampuan dalam proses penalaran secara bahasa (*linguistic reasoning*). Sehingga dalam perancangannya tidak memerlukan persamaan matematik dari objek yang akan dikendalikan.

Teori himpunan logika *fuzzy* di kembangkan oleh Professor Lofti A. Zadeh pada tahun 1965. Ia berpendapat bahwa logika benar dan salah dari logika *boolean* konvensional tidak dapat mengatasi masalah gradasi yang berada pada dunia nyata. Untuk mengatasi masalah gradasi yang tidak terhingga tersebut, Zadeh mengembangkan sebuah himpunan *fuzzy*. Tidak seperti logika *boolean*, logika *fuzzy* mempunyai nilai yang berkelanjutan. *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama. Berdasarkan hal tersebut diatas Logika *fuzzy* dapat digunakan untuk memodelkan suatu permasalahan yang matematis, dimana konsep matematis yang mendasari penalaran fuzzy sangat sederhana dan mudah dimengerti (Rizkysari Meimaharani, 2014).

2.1.4.1. Himpunan *Fuzzy*

Dalam teori logika *fuzzy* dikenal himpunan *fuzzy* (pengelompokan sesuatu berdasarkan variabel bahasa dinyatakan dalam fungsi keanggotaan. Didalam semesta pembicaraan (*dicourse*), Fungsi keanggotaan dari suatu himpunan *fuzzy* tersebut bernilai 0 sampai dengan 1. Contoh dari himpunan variabel bahasa antara lain: Himpunan penghasilan orang tua dapat dinyatakan dengan rendah, sedang dan tinggi. Grafik dari himpunan penghasilan orang tua ditunjukkan pada gambar 2.1 ini.



Gambar 2.1 Contoh Himpunan Penghasilan Orang Tua
 Sumber: Sutojo (2011) hal 271

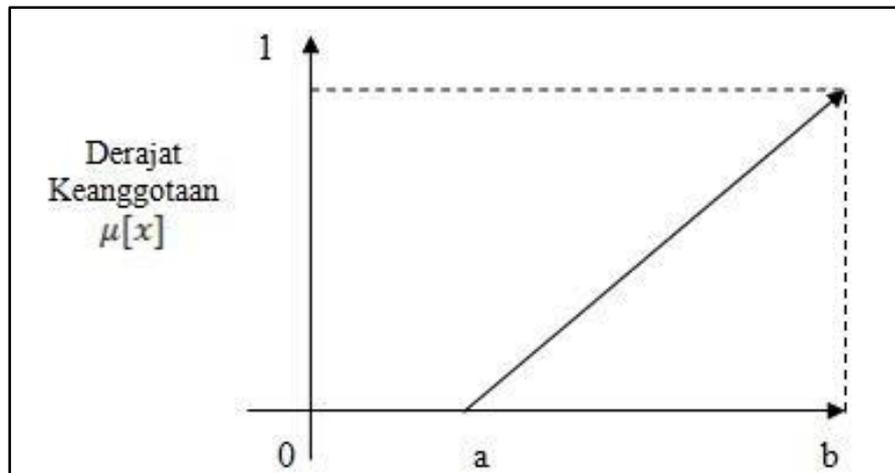
2.1.4.2. Fungsi Keanggotaan (*Membership Function*)

Fungsi keanggotaan merupakan suatu kurva yang menunjukkan pemetaan titik-titik input data kedalam nilai keanggotaannya (disebut juga dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Untuk mendapatkan nilai keanggotaan dapat menggunakan cara pendekatan fungsi. Ada beberapa fungsi keanggotaan yang digunakan dalam teori himpunan *fuzzy* adalah:

a. Representasi *Linier*

Pada representasi *linier*, 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 dua keadaan himpunan *fuzzy* yang linier. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju

ke nilai domain yang memiliki derajat keanggotaan lebih tinggi seperti pada Gambar 2.2.



Gambar 2.2 Representasi Linier Naik

Sumber: Sutojo (2011) hal 214

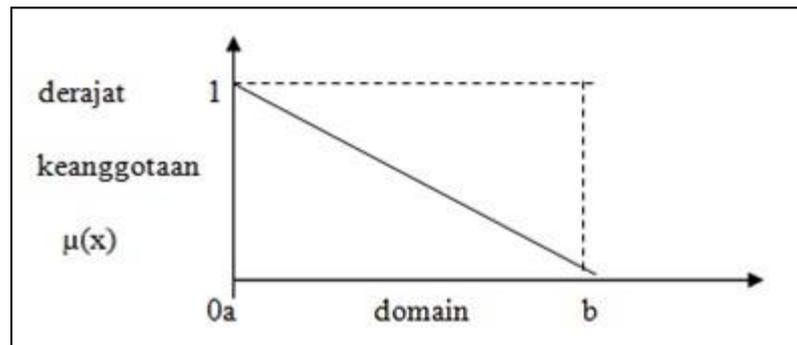
Persamaan fungsi keanggotaan linear naik:

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

Rumus 2.1 Representasi Linier Naik

Sumber: Sutojo (2011) hal 214

Kedua, merupakan kebalikan 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 seperti pada Gambar 2.3.



Gambar 2.3 Representasi Linier Turun

Sumber: Sutojo (2011) hal 215

Persamaan fungsi keanggotaan linier turun :

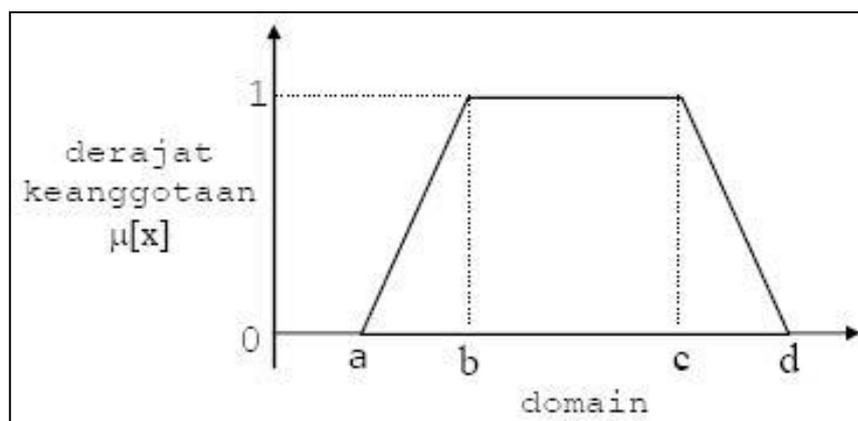
$$\mu(x) = \begin{cases} \frac{b-x}{b-a} & ; a \leq x \leq b \\ 0 & ; x \geq b \end{cases}$$

Rumus 2.2 Representasi Linier Turun

Sumber: Sutojo (2011) hal 216

b. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1 seperti pada Gambar 2.4.



Gambar 2.4 Representasi Kurva Trapesium

Sumber: Sutojo (2011) hal 218

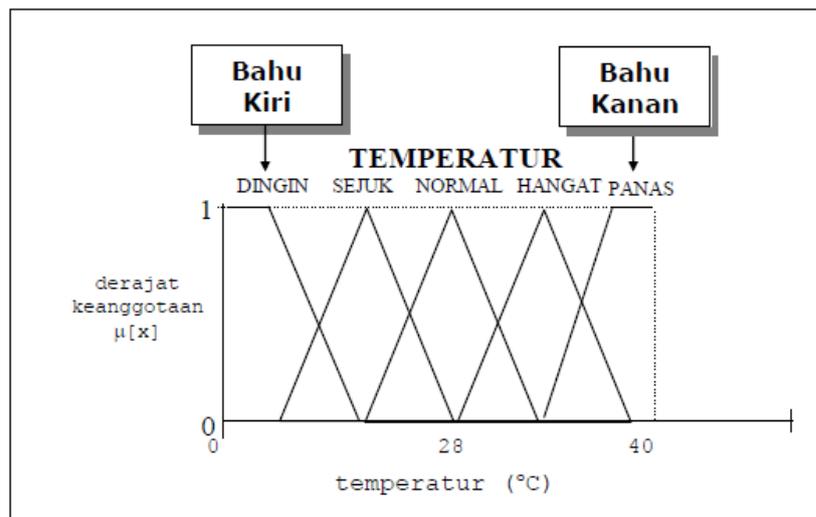
Persamaan fungsi keanggotaan Kurva Trapesium:

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

Rumus 2.3 Kurva Trapesium
Sumber: Sutojo (2011) hal 218

c. Representasi Kurva Bahu

Daerah yang terletak ditengah-tengah suatu variabel yang dipresentasikan dalam bentuk segitiga, pada sisi kanan dan kirinya akan naik dan turun (misalkan: dingin bergerak ke sejuk bergerak ke hangat dan bergerak ke panas). Tetapi terkadang salah satu sisi dari variabel tersebut tidak mengalami perubahan. Sebagai contoh, apabila telah mencapai kondisi panas, kenaikan temperatur akan tetap berada pada kondisi panas. 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. Gambar 2.5 menunjukkan variabel temperatur dengan daerah bahunya.

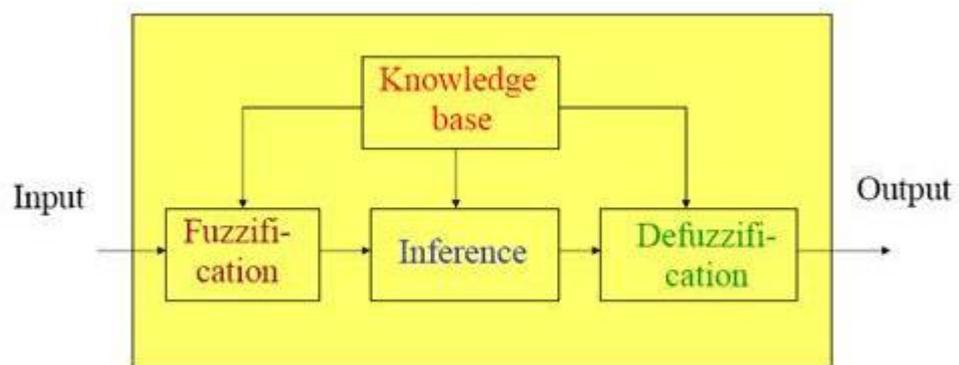


Gambar 2.5 Representasi Kurva Bahu
Sumber: Sutojo (2011) hal 219

2.1.4.3. Metode Sistem Inferensi Fuzzy

Sistem inferensi *fuzzy* akan berfungsi sebagai pengendali proses tertentu dengan menggunakan aturan-aturan inferensi berdasarkan logika *fuzzy*. Sistem inferensi memiliki 4 unit yaitu (Aldyanto, Informatika, & Industri, 2016)

1. Unit fuzzifikasi (*fuzzification unit*)
2. Unit penalaran logika fuzzy (*fuzzy logic reasoning unit*)
3. Unit basis pengetahuan (*knowledge base unit*) yang terdiri dari:
 - a. Basis data (*data base*) yang memuat fungsi-fungsi keanggotaan dari himpunan-himpunan *fuzzy* yang terkait dengan nilai dari variabel-variabel linguistik yang dipakai.
 - b. aturan (*rule base*) yang memuat aturan-aturan berupa implikasi *fuzzy*
4. Unit defuzzifikasi / unit penegasan (*defuzzification unit*). Diagram kontrol fuzzy ditunjukkan pada gambar 2.6



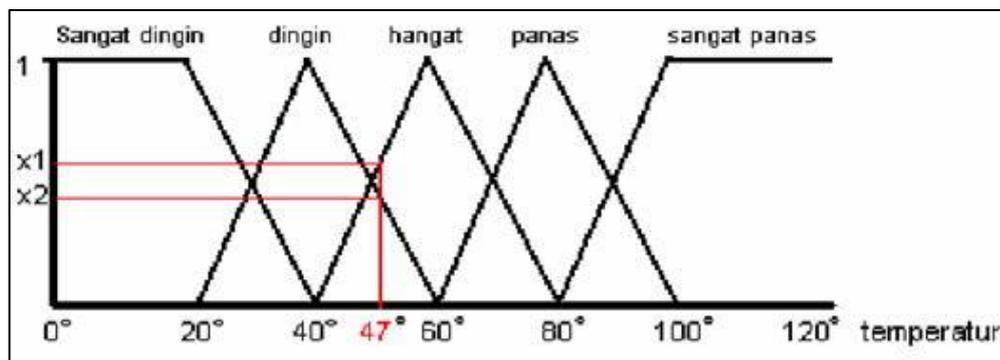
Gambar 2.6 Diagram Kontrol Fuzzy

Sumber: Data Peneliti (2018)

2.1.4.4. Fuzzifikasi

Fuzzifikasi yaitu suatu proses untuk mengubah suatu masukan dari bentuk tegas (*crisp*) menjadi *fuzzy* (variabel linguistik) yang biasanya disajikan dalam bentuk himpunan-himpunan *fuzzy* dengan suatu fungsi keanggotaannya masing-

masing. Contoh dari proses *Fuzzification* adalah seperti yang ditunjukkan digambar 2.7. Sebuah sistem *fuzzy* untuk mengukur suhu mempunyai lima buah *membership function* yang mempunyai label sangat dingin, dingin, hangat, panas, sangat panas. Kemudian *input* yang diperoleh dari *crisp* input adalah 47° maka pengambilan *fuzzy input*-nya adalah seperti pada gambar 2.7.



Gambar 2.7 Nilai Tegas Menjadi Input

Sumber: Aldyanto (2016)

Sehingga didapat dua *fuzzy* input yang masing-masing adalah: dingin (x_2) dan hangat (x_1). Nilai x_1 dan x_2 dapat dicari dengan rumus persamaan garis. Yang menentukan sistem anda sensitif atau tidak adalah *membership function* ini. Jika *membership function*-nya banyak maka sistem anda menjadi sensitif. Yang dimaksud dengan sensitif dalam hal ini adalah jika *input*-nya berubah sedikit saja maka sistem akan cepat merespon dan menghasilkan suatu *output* lain. *Output* dari proses *fuzzification* ini adalah sebuah nilai *input fuzzy* atau yang biasanya dinamakan *fuzzy input*.

2.1.4.5. Aturan Dasar Logika Fuzzy

Aturan *Fuzzy If-Then* (atau disebut juga aturan *fuzzy*, *fuzzy implikasi*, atau pernyataan kondisional *Fuzzy*)(Nasution & Fuzzy, 2012).

“JIKA” $X=A$ dan “JIKA” $Y=B$ “MAKA” $Z=C$

Contoh dari aturan jika-maka ini pada pengendalian suhu ruangan dengan pengaturan kecepatan kipas angin melalui frekuensi variabel adalah sebagai berikut.

1. “JIKA” suhu panas dan
 2. “JIKA” kecepatan kipas sangat lambat
 3. “MAKA” sumber frekuensi dinaikkan sangat tinggi agar kecepatan kipas tinggi
- Jadi aturan dasar kontrol logika *fuzzy* ditentukan dengan bantuan seorang pakar yang mengetahui karakteristik objek yang akan dikendalikan. Aturan dasar tersebut dapat dinyatakan dalam bentuk matrik aturan dasar kontrol logika *fuzzy*. Contoh aturan pengaturan suhu ruangan dapat dilihat pada tabel berikut dimana, $X=$ Suhu, $Y=$ Kecepatan Kipas.

Tabel 2.1 Contoh Aturan Dasar *Fuzzy*

x/y	B	S	K
B	K	K	B
S	K	S	K
K	B	K	B

Sumber: Data Peneliti (2018)

2.1.4.6. Mesin Penalaran Kontrol Logika *Fuzzy*

Mesin penalaran adalah proses implikasi dalam menalar nilai masukan guna penentuan nilai keluar sebagai bentuk pengambil keputusan. Salah satu model penalaran *max-min*. Dalam penalaran *max-min* proses pertama yang dilakukan adalah melakukan operasi operasi *min* sinyal keluaran lapisan fuzzifikasi, yang

diteruskan dengan operasi *max* untuk mencari nilai keluaran yang selanjutnya akan difuzzifikasikan sebagai bentuk keluaran pengontrol. Operasional *max-min* tersebut dinyatakan sebagai berikut.

1. Operasi *min* (irisan)

$$a \cap b = \min(a,b) = a \text{ if } a \leq b$$

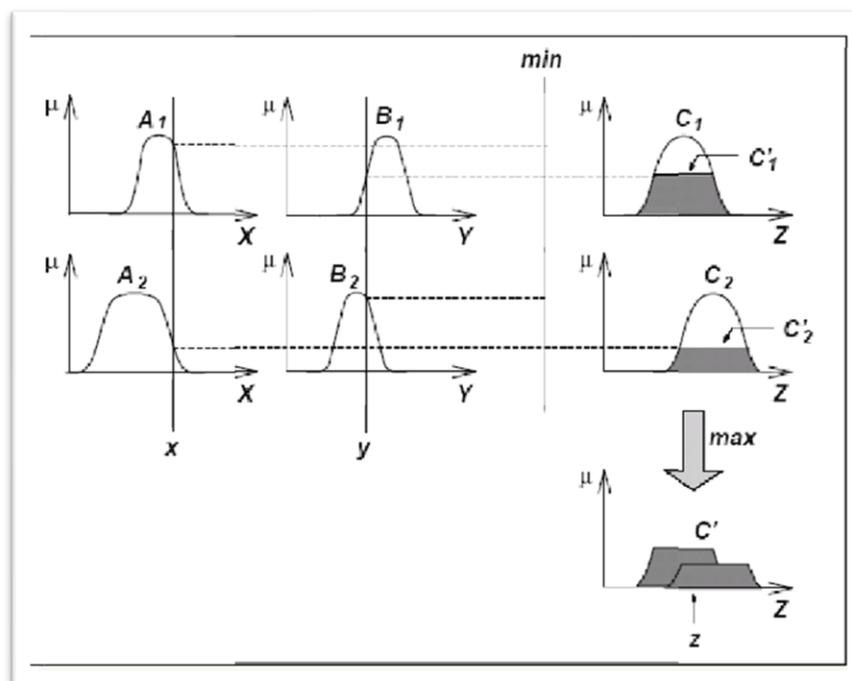
$$= b \text{ if } a > b$$

2. Operasi *max* (union)

$$a \cup b = \max(a,b) = a \text{ if } a \geq b$$

$$= b \text{ if } a < b$$

Proses penalaran *max-min* dapat dijelaskan dengan gambar 2.8



Gambar 2.8 Operasi Min-Max
Sumber: Sri Kusumadewi (2013) hal 189

2.1.4.7. Defuzzifikasi

Defuzzifikasi adalah komponen penting dalam pemodelan sistem samar. *Defuzzifikasi* digunakan untuk menghasilkan nilai variabel solusi yang diinginkan dari suatu daerah konsekuen samar (*fuzzy*) (Setiadi, 2012). Metode dalam *defuzzifikasi* antara lain:

a. Metode *Max*

Solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah *fuzzy*, dan mengaplikasikannya ke *output* dengan menggunakan operator OR (*union*).

Secara umum dapat dituliskan :

$$\mu_{sf} [x_i] = \max(\mu_{sf} [x_i], \mu_{kf}[x_i])$$

b. Metode *Additive (Sum)*

Solusi himpunan *fuzzy* diperoleh dengan cara melakukan *bounded sum* terhadap semua *output* daerah *fuzzy*. Secara umum dapat dituliskan :

$$\mu_{sf} [x_i] = \min(1, \mu_{sf} [x_i] + \mu_{kf}[x_i])$$

c. Metode Probabilistik OR (*probor*)

Solusi himpunan *fuzzy* diperoleh dengan cara melakukan produk terhadap semua *output* daerah *fuzzy*. Secara umum dapat dituliskan:

$$\mu_{sf} [x_i] = (\mu_{sf} [x_i] + \mu_{kf}[x_i]) - (\mu_{sf} [x_i] * \mu_{kf}[x_i])$$

Keterangan:

$\mu_{sf} [x_i]$: nilai keanggotaan solusi *fuzzy* sampai aturan ke-i

$\mu_{kf}[x_i]$: nilai keanggotaan konsekuen aturan ke-i

Metode *defuzzyfikasi* yang akan digunakan adalah Metode *Center of Area* (COA). Dimana pada metode ini, solusi *crisp* diperoleh dengan cara mengambil titik pusat (z^*) daerah *fuzzy*.

2.1.4.8. Metode Sugeno

Fuzzy metode sugeno merupakan metode inferensi *fuzzy* untuk aturan yang direpresentasikan dalam bentuk *IF – THEN*, dimana *output* (konsekuen) tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linier. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. Model Sugeno menggunakan fungsi keanggotaan Singleton yaitu fungsi keanggotaan yang memiliki derajat keanggotaan 1 pada suatu nilai *crisp* tunggal dan 0 pada nilai *crisp* yang lain. Untuk Orde 0 dengan rumus :

$$IF (x_1 \text{ is } a_1) \circ (x_2 \text{ is } A_2) \circ \dots \circ (x_n \text{ is } A_n)$$

$$THEN z = k,$$

dengan A_i adalah himpunan *fuzzy* ke i sebagai antaseden (alasan), \circ adalah operator *fuzzy* (*AND* atau *OR*) dan k merupakan konstanta tegas sebagai konsekuen (kesimpulan). Sedangkan rumus Orde 1 adalah:

$$IF (x_1 \text{ is } a_1) \circ (x_2 \text{ is } A_2) \circ \dots \circ (x_n \text{ is } A_n)$$

$$THEN z = p_1 * x_1 + \dots + p_n * x_n + q,$$

dengan A_i adalah himpunan *fuzzy* ke i sebagai antaseden, \circ adalah operator *fuzzy* (*AND* atau *OR*), p_i adalah konstanta ke i dan q juga merupakan konstanta dalam konsekuen.

2.2. Variabel

Variabel yang digunakan dalam penelitian ini adalah permintaan, bahan baku, dan karyawan.

2.2.1. Variabel Permintaan

Permintaan dalam penelitian ini adalah permintaan kebutuhan mebel dari pelanggan PT. Karya Abadi *Furniture*.

2.2.2. Variabel Bahan Baku

Bahan baku adalah bahan-bahan yang diperlukan dalam pembuatan mebel seperti kayu dan paku. Dalam penelitian ini, bahan baku merupakan variabel yang mempengaruhi hasil penelitian.

2.2.3. Variabel Karyawan

Karyawan adalah pekerja yang mengerjakan produksi mebel di PT. Karya Abadi *Furniture*. Jumlah karyawan merupakan variabel yang mempengaruhi hasil penelitian.

2.3. Software Pendukung

2.3.1. MATLAB

MATLAB (*Matrix Laboratory*) adalah sebuah program untuk analisis dan komputasi numerik dan merupakan suatu bahasa pemrograman matematika lanjutan yang dibentuk dengan dasar pemikiran menggunakan sifat dan bentuk

matriks. Pada awalnya, program ini merupakan *interface* untuk koleksi rutin-rutin numerik dari proyek LINPACK dan EISPACK, dan dikembangkan menggunakan bahasa FORTRAN namun sekarang merupakan produk komersial dari perusahaan Mathworks, Inc. yang dalam perkembangan selanjutnya dikembangkan menggunakan bahasa C++ dan *assembler* (utamanya untuk fungsi-fungsi dasar MATLAB). MATLAB telah berkembang menjadi sebuah *environment* pemrograman yang canggih yang berisi fungsi-fungsi *built-in* untuk melakukan tugas pengolahan sinyal, aljabar linier, dan kalkulasi matematis lainnya. MATLAB juga berisi *toolbox* yang berisi fungsi-fungsi tambahan untuk aplikasi khusus. MATLAB bersifat *extensible*, dalam arti bahwa seorang pengguna dapat menulis fungsi baru untuk ditambahkan pada *library* ketika fungsi-fungsi *built-in* yang tersedia tidak dapat melakukan tugas tertentu. Kemampuan pemrograman yang dibutuhkan tidak terlalu sulit bila telah memiliki pengalaman dalam pemrograman bahasa lain seperti C, PASCAL, atau FORTRAN.

MATLAB (*MATrix LABoratory*) yang merupakan bahasa pemrograman tingkat tinggi berbasis pada matriks sering digunakan untuk teknik komputasi numerik, yang digunakan untuk menyelesaikan masalah-masalah yang melibatkan operasi matematika, elemen, matrik, optimasi, aproksimasi dan lain-lain. Sehingga Matlab banyak digunakan pada:

- a. Matematika dan Komputansi
- b. Pengembangan dan Algoritma
- c. Pemrograman modeling, simulasi, dan pembuatan *prototype*
- d. Analisa data, eksplorasi dan visualisasi

- e. Analisis numerik dan statistik
- f. Pengembangan aplikasi teknik

2.3.2 *Window-window* pada MATLAB

Ada beberapa macam *window* yang tersedia dalam MATLAB, yang dapat dijelaskan sebagai berikut:

a. MATLAB *Command window/editor*

MATLAB *Command window/editor* merupakan *window* yang dibuka pertama kali setiap kali MATLAB dijalankan. *Command windows* juga digunakan untuk memanggil *tool* MATLAB seperti *editor*, *debugger* atau fungsi.

b. MATLAB *Editor/Debugger* (Editor *M-File*/Pencarian Kesalahan)

Window ini merupakan *tool* yang disediakan oleh MATLAB 5 keatas. Berfungsi sebagai *editor script* MATLAB (*M-file*). Walaupun sebenarnya *script* ini untuk pemrograman MATLAB dapat saja menggunakan *editor* yang lain seperti *notepad*, *wordpad* bahkan *word*.

c. *Figure Windows*

Window ini adalah hasil visualisasi dari *script* MATLAB. Namun MATLAB memberi kemudahan bagi *programmer* untuk mengedit *window* ini sekaligus memberikan program khusus untuk itu. Sehingga *window* ini selain berfungsi sebagai visualisasi *output* dapat juga sekaligus menjadi media *input* yang interaktif.

d. MATLAB *help window*

MATLAB menyediakan sistem *help* yang dapat diakses dengan perintah *help*. Misalnya, untuk memperoleh informasi mengenai fungsi *elfun* yaitu fungsi untuk trigonometri, eksponensial, *complex* dan lain-lain, maka hanya perlu mengetikkan perintah berikut: » *help elfun*

2.4. Produksi

Produksi merupakan hasil akhir dalam proses atau aktivitas ekonomi dengan memanfaatkan beberapa masukan atau *input* untuk menghasilkan *output* (Rugian, 2013). Pengertian produksi juga dapat diartikan sebagai usaha untuk menciptakan atau menambah manfaat ekonomi suatu benda dengan tujuan untuk memenuhi kebutuhan manusia. Produksi merupakan suatu kegiatan yang menghasilkan *output* dalam bentuk barang maupun jasa. Contoh: pabrik batre yang memproduksi batu baterai, pabrik mutifa yang memproduksi obat-obatan, dan lain sebagainya.

Dari definisi diatas dapat diketahui bahwa untuk menghasilkan barang atau jasa diperlukan usaha mendayagunakan masukan berupa tenaga kerja, bahan baku, dan peralatan. Perkembangan dari pada proses produksi menghasilkan banyak macam jenis-jenis proses produksi dalam perusahaan.

2.5. Meubel

Meubel dalam bahasa Inggris diterjemahkan menjadi *furniture*. Istilah “mebel” digunakan karena sifat Bergeraknya atau mobilitasnya sebagai barang lepas di dalam *interior* arsitektural. Kata mebel berasal dari bahasa Perancis yaitu *meubel*, atau bahasa Jerman yaitu *mobel*. Pengertian mebel secara umum adalah benda pakai yang dapat dipindahkan, berguna bagi kegiatan hidup manusia, mulai dari duduk, tidur, bekerja, makan, bermain dan sebagainya, yang memberi kenyamanan dan keindahan bagi pemakainya (Yusuf et al., 2017). Mebel juga merupakan salah satu produk kayu olahan yang pertumbuhannya amat pesat dalam beberapa dekade terakhir ini adalah produk mebel. Berawal dari pekerjaan rumah tangga, produk mebel kini telah menjadi industri yang cukup besar dengan tingkat penyerapan tenaga kerja terdidik yang tidak sedikit. Produk jenis ini secara prinsip dibagi dalam dua kategori yaitu mebel untuk taman (*garden*) dan *interior* dalam rumah.

2.6. Penelitian Terdahulu

Pada tinjauan penelitian sebelumnya akan dibahas jurnal dan artikel yang mendukung sebagai dasar pembahasan interpretasi penelitian pada bahan sebelumnya

- 1. Aldyanto Firman, Minarni. Prediksi Jumlah Produksi Roti Menggunakan Metode Logika *Fuzzy* (Studi Kasus: Roti Malabar Bakery). Jurnal Teknoif, Oktober 2016, Vol. 4 No. 2, ISSN: 2338-2724.** Dalam memperkirakan jumlah produksi roti dan mempermudah pengelola perusahaan dalam melakukan penentuan jumlah produksi dapat menggunakan metode tsukamoto dimana hasil prediksi jumlah produksi tidak berbeda jauh dari jumlah produksi yang dilakukan perusahaan.
- 2. Indra. Penerapan Logika *Fuzzy* untuk Menentukan Jumlah Produksi Beras Berdasarkan Data Persediaan dan Jumlah Permintaan (Studi Kasus: UD Siregar Wonomulyo). Jtriste, Oktober 2016, Vol. 3 No. 2, ISSN: 2355-3677.** Dalam menentukan jumlah produksi beras menggunakan logika *fuzzy* dapat membantu perusahaan dalam mengambil keputusan dengan nilai kebenaran mencapai 98% dan dapat membantu admin gudang dalam menentukan jumlah produksi beras.
- 3. Prayogi Agus, Edy Santoso. Sistem Pendukung Keputusan Untuk Penentuan Jumlah Produksi Nanas Menggunakan Metode *Fuzzy* Tsukamoto (Studi Kasus: PT. Great Giant Pineapple). Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Juni 2017, Vol. 2 No. 6, ISSN: 2548-964X.** Sistem pendukung keputusan penentuan jumlah produksi nanas menggunakan metode *fuzzy* Tsukamoto memiliki kinerja sistem yang mampu berjalan sesuai dengan kebutuhan fungsional

dibuktikan dengan hasil pengujian *Black Box* yang memberikan nilai presentase sebesar 100%. Berdasarkan hasil pengujian akurasi diperoleh nilai kesalahan dari hasil peramalan yang kecil yakni 0,0607 %. Hasil yang diberikan oleh metode *Fuzzy* Tsukamoto memiliki kesesuaian dengan hasil data perusahaan dengan nilai kesalahan 0,0607 %.

4. **Farouq Mauladi Kemal. Penentuan Jumlah Produksi Sarung Tenun Tradisional dengan Metode *Fuzzy* Tsukamoto. Jurnal SMATIKA, April 2017, Vol. 7 No. 1, ISSN: 2087-0256.** Dengan menggunakan *Fuzzy* Tsukamoto dapat membantu dalam menentukan jumlah produksi sarung tenun tradisional sehingga tidak terjadi kerugian dalam proses produksi dan menghasilkan keuntungan bagi perusahaan Sarung Tenun.
5. **Oktaviani Laily. Sistem Pendukung Perhitungan Jumlah Produksi Folding Gate Menggunakan *Fuzzy Logic* pada PT. Jihan Jaya. Jurnal Sistem Informasi, 2014, Vol. 1 No. 1, ISSN: 2406-7768.** Dengan adanya sistem ini dapat membantu meningkatkan produksi dengan jumlah yang akurat serta membantu dalam menghitung atau meninjau sejauh mana perusahaan dapat memproduksi *Folding Gate* dalam tempo satu tahun.
6. **Sunarsan Sitohang, Abba Suganda Girsang, Suharjito. Prediction of the Number of Airport Passengers Using Fuzzy C-Means and Adaptive Neuro Fuzzy Inference System. International Review of Automatic Control (I.R.E.A.CO.), 2017, Vol. 10 No. 3, ISSN: 1974-6059.**

Fuzzy C-Means (FCM) adalah metode hirarkis untuk menciptakan komposisi hirarkis dari data objek yang menghasilkan *cluster* bersarang. Pengelompokan non-hierarkis memberikan n jumlah objek dan k yang merupakan jumlah kelompok yang terbentuk dan memproses objek tersebut ke dalam kelompok berdasarkan kriteria pengoptimalan tertentu, di mana masing-masing kelompok merupakan representasi dari sebuah *cluster*.

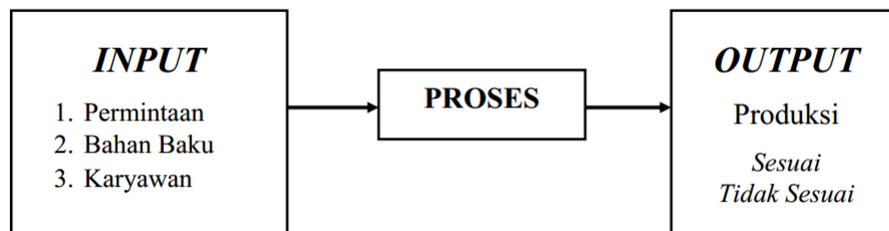
7. **Khushal Khairnar, Kamleshwar Khairnar, Sanketkumar Mane, Rahul Chaudhari.** *Furniture Layout Application Based on Marker Detection and Using Augmented Reality. International Research Journal of Engineering and Technology (IRJET), 2015, Vol.2 No. 7, ISSN: 2395-0056.*

Augmented reality adalah bidang komputer penelitian yang membahas kombinasi kenyataan dengan data terkait komputer. Sistem yang dibuat memungkinkan pengguna untuk membeli *furniture* di rumah tanpa harus datang ke toko. Tujuan utama dari proyek ini adalah untuk mengembangkan aplikasi *windows* dengan cara virtual dan efisiensi waktu beraktivitas.

2.7. Kerangka Pemikiran

Menurut Sugiyono (2012:60), kerangka berfikir merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah

diidentifikasi sebagai masalah yang penting. Kerangka berfikir yang baik akan menjelaskan secara teoritis pertautan antar variabel yang akan diteliti. Jadi kerangka pemikiran merupakan kajian-kajian yang menjelaskan hubungan yang mungkin terdapat antara berbagai faktor yang saling mengait dan membentuk susunan yang dapat dirumuskan. Dalam penelitian ini menjelaskan bagaimana menentukan jumlah produksi mebel menggunakan logika *fuzzy*. Adapun kerangka pemikiran dapat di lihat pada gambar berikut.



Gambar 2.9 Kerangka Berpikir

Sumber: Data peneliti (2018)

Penjelasan Kerangka Berpikir:

1. *Input*
 - a. Permintaan adalah jumlah pemesanan/*order*-an mebel dari pelanggan.
 - b. Bahan baku adalah bahan-bahan yang diperlukan untuk membuat mebel seperti kayu dan paku.
 - c. Karyawan adalah pekerja yang mengerjakan pemesanan dari pelanggan.
2. Proses adalah pengolahan tiap variabel *input* ke dalam sistem.
3. *Output* adalah hasil akhir yang dihasilkan dari pemrosesan.