

## **BAB II**

### **LANDASAN TEORI**

Pada bab ini menjelaskan teori-teori yang mendukung dalam pembuatan tugas akhir dengan judul sistem pendukung keputusan untuk menilai kinerja keuangan perusahaan menggunakan metode analisa rasio dengan metode *fuzzy*, yang meliputi :

#### **2.1. Laporan Keuangan**

Laporan keuangan merupakan hasil akhir dari suatu proses penentuan yang merupakan suatu ringkasan transaksi-transaksi keuangan yang terjadi selama tahun buku yang dapat dijadikan dasar untuk analisa bisnis dalam ekonomi dan peramalan masa depan.

##### **1. Jenis Laporan Keuangan**

Laporan keuangan disusun berdasarkan standart khusus akuntansi keuangan yang terdiri dari :

##### **a. Neraca**

Dalam hal penyajian neraca secara umum aktiva dan pasiva. Neraca bermanfaat untuk menggambarkan posisi keuangan pada saat tertentu dan memberikan gambaran aktiva, hutang dan modal.

##### **b. Perhitungan laba /rugi**

Laporan yang menggambarkan pendapatan dan biaya kegiatan lainnya dalam satu periode secara rinci.

## 2. Tujuan laporan kegiatan

Adapun tujuan umum dari laporan keuangan, sebagai berikut :

- a. Untuk memberikan informasi keuangan mengenai aktiva dan kewajiban serta modal suatu perusahaan.
- b. Untuk memberikan informasi keuangan yang membantu para pemakai laporan keuangan didalam menaksir potensi perusahaan dalam menghasilkan laba.

## 3. Sifat dan keterbatasan laporan keuangan

Walaupun laporan keuangan merupakan informasi yang sangat berguna bagi pihak-pihak untuk pengambilan keputusan, tetapi laporan keuangan tersebut masih mempunyai sifat dan keterbatasan. Menurut Munawir (1998 : 10) sifat dan keterbatasan laporan keuangan adalah sebagai berikut :

- a. Laporan keuangan bersifat historis. Laporan keuangan tidak dapat dianggap sebagai satu-satunya sumber informasi dalam proses pengambilan keputusan.
- b. Laporan keuangan bersifat umum dan bukan dimaksudkan untuk memenuhi kebutuhan pihak tertentu.
- c. Proses penyusunan laporan keuangan tidak luput dari penggunaan penaksiran berbagai pertimbangan.
- d. Laporan keuangan bersifat konservatif dalam menghadapi ketidakpastian.
- e. Laporan keuangan lebih menekankan pada makna ekonomi suatu peristiwa atau transaksi daripada bentuk hukumnya (formalitas).

#### 4. Pihak-pihak yang berkepentingan dengan laporan keuangan

Pihak-pihak yang berkepentingan dengan posisi keuangan maupun perkembangan suatu perusahaan meliputi :

##### a. Pihak intern perusahaan

###### - Pemilik

Karena dengan laporan keuangan tersebut pemilik perusahaan dapat menilai sukses tidaknya manajer dalam memimpin perusahaan dan kesuksesan seorang manajer biasanya dinilai dengan laba yang diperoleh perusahaan.

###### - Manajer atau pemimpin

Dengan mengetahui posisi keuangan maka dapat menyusun rencana yang baik dan memperbaiki sistem pengawasannya dan juga menentukan kebijakan-kebijakan dengan tepat.

##### b. Pihak ekstern perusahaan

###### - Investor

Mereka berkepentingan terhadap prospek jaminan investasinya serta untuk mengetahui kondisi keuangan jangka pendek perusahaan.

###### - Kreditur dan Banker

Laporan keuangan dibutuhkan mereka sebelum mengambil keputusan untuk memberikan atau menolak permintaan kredit dari suatu perusahaan.

## 2.2. Analisa Rasio Keuangan

Analisa ratio adalah suatu metode analisa untuk mengetahui hubungan dari pos-pos tertentu dalam neraca atau laporan rugi laba secara individual atau kombinasi dari kedua laporan tersebut, (Munawir 1992:37). Atau merupakan bagian kegiatan yang hendak mencoba menilai keadaan perusahaan yang dapat digunakan untuk berbagai tujuan. Sehat tidaknya suatu perusahaan akan dapat ditunjukkan oleh ratio-ratio yang diperoleh dari analisa ini.

Analisa rasio keuangan dapat memberikan indikasi apakah perusahaan memiliki kas yang cukup untuk memenuhi kewajiban finansialnya, besarnya piutang yang cukup rasional, efisien manajemen persediaan, perencanaan pengeluaran investasi yang baik dan struktur modal yang sehat sehingga tujuan memaksimalkan kemakmuran pemegang saham dapat tercapai.

Dalam melakukan analisa rasio keuangan dapat dilakukan dengan membandingkan prestasi satu periode dibandingkan dengan periode sebelumnya sehingga diketahui adanya kecenderungan selama periode tertentu. Selain itu dapat pula dilakukan dengan cara membandingkan dengan perusahaan sejenis dalam industri tersebut sehingga dapat diketahui bagaimana posisi perusahaan dalam industri.

Selain membantu manajer finansial dalam membuat keputusan-keputusan yang penting bagi perusahaan, analisa ratio keuangan berguna juga bagi calon investor atau kreditur sebagai bahan pertimbangan apakah menguntungkan atau tidak menanamkan dana mereka kedalam perusahaan melalui pasar modal dengan cara membeli saham perusahaan yang go-publik.

Penggolongan angka rasio berdasarkan tujuannya, angka-angka rasio keuangan dapat diklasifikasikan dalam lima golongan sebagai berikut :

*a. Liquidity Ratio*

Rasio yang mengukur kemampuan perusahaan untuk memenuhi kewajiban finansial yang berupa hutang-hutang jangka pendek. Yang termasuk liquidity ratio antara lain :

$$\text{- Current Ratio} = \frac{\text{Aktiva Lancar}}{\text{Hutang Lancar}} \times 100\% \dots\dots\dots (1)$$

$$\text{- Cash Ratio} = \frac{\text{Kas + Efek}}{\text{Hutang Lancar}} \times 100\% \dots\dots\dots (2)$$

*b. Solvency / Leverage Ratio*

Rasio yang mengukur kemampuan perusahaan untuk membayar hutang jangka panjang, bila pada suatu saat perusahaan dilikuidasi atau dibubarkan. Dan rasio ini dapat menunjukkan seberapa jauh perusahaan dinilai oleh pihak luar atau kreditor. Yang termasuk leverage rasio adalah :

$$\text{- Debt To Total Assets Ratio} = \frac{\text{Total Hutang}}{\text{Total Aktiva}} \times 100\% \dots\dots\dots (3)$$

$$\text{- Debt To Equity Ratio} = \frac{\text{Total Hutang}}{\text{Total Modal Sendiri}} \times 100\% \dots\dots\dots (4)$$

*c. Asset Management Ratio*

Rasio yang digunakan untuk mengukur efektivitas perusahaan dalam menggunakan sumber-sumber yang tersedia didalam perusahaan. Yang termasuk asset management ratio antara lain :

$$\text{- Perputaran Aktiva Tetap} = \frac{\text{Penjualan}}{\text{Aktiva Tetap}} \dots\dots\dots (5)$$

$$\text{- Perputaran Aktiva} = \frac{\text{Penjualan}}{\text{Total Aktiva}} \dots\dots\dots (6)$$

*d. Profitability Ratio*

Rasio yang digunakan untuk mengukur kemampuan perusahaan dalam memperoleh keuntungan. Yang termasuk profitability ratio antara lain :

$$\text{- Net Profit Margin} = \frac{\text{Laba Bersih setelah Pajak}}{\text{Penjualan}} \times 100\% \dots\dots\dots (7)$$

$$\text{- Return On Assets} = \frac{\text{Laba Bersih sebelum Pajak (EBIT)}}{\text{Total Aktiva}} \times 100\% \dots\dots (8)$$

$$\text{- Return On Equity} = \frac{\text{Laba Bersih setelah Pajak}}{\text{Modal Sendiri}} \times 100\% \dots\dots\dots (9)$$

### 2.3. Analisa Kinerja Keuangan Perusahaan

Analisa keuangan merupakan suatu penilaian terhadap kinerja perusahaan pada waktu yang lalu dan prospek pada masa mendatang. Melalui analisa keuangan tersebut dapat diketahui kekuatan dan kelemahan suatu perusahaan dengan menggunakan informasi yang terdapat dalam laporan keuangan (*financial statement*). Laporan keuangan perusahaan mencerminkan prestasi kerja perusahaan yang menunjukkan berhasil atau tidaknya suatu perusahaan dalam menjalankan operasinya pada periode tertentu.

Keberhasilan kinerja suatu perusahaan merupakan impian dari setiap perusahaan yang terus menerus diusahakan untuk mencapainya. Kinerja adalah ukuran seberapa efisien dan efektif seorang manajer atau sebuah organisasi dalam mencapai tujuannya. Kinerja yang efektif dan efisien akan dapat meningkatkan kemampuan perusahaan dari berbagai segi baik keuangan, manajemen perusahaan maupun performance perusahaan (Husein Umar, 2002:25).

Adapun tujuan dari pengukuran kinerja perusahaan, yaitu :

- a. Untuk mengukur efektifitas manajemen berdasarkan hasil pengembalian yang dihasilkan penjual dan investasi.
- b. Untuk mengukur kemampuan perusahaan dalam mempertahankan pertumbuhan perekonomian dalam industri atau pasar produk tempatnya beroperasi.

#### **2.4. Sistem Pendukung Keputusan**

Pada dasarnya SPK ini merupakan pengembangan lebih lanjut dari Sistem Informasi Manajemen terkomputerisasi (*Computerized Management Information System*), yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakainya. Sifat interaktif ini dimaksudkan untuk memudahkan integrasi antara berbagai komponen dalam proses pengambilan keputusan, seperti prosedur, kebijakan, teknik analisis, serta pengalaman dan wawasan manajerial guna membentuk suatu kerangka keputusan yang bersifat fleksibel.

Sebuah sistem pendukung keputusan merangkai sumber-sumber terpenting yang berasal dari tiap individu dengan kemampuan dari sistem komputer untuk memperbaiki kualitas dari keputusan-keputusan. Hal itu merupakan sebuah *computer based system* untuk memanajemen pembuat keputusan dengan masalah-masalah yang semi-terstruktur.

#### **2.5. Logika Fuzzy**

Penggolongan himpunan (*Set*) yang dinamakan *fuzzy (fuzzy set)* digunakan untuk menghitung gradasi yang tidak terbatas jumlahnya antara benar

dan salah. Tidak seperti logika *boolean* yang menyatakan bahwa suatu pernyataan adalah benar atau salah, *fuzzy logic* dapat membaginya dalam derajat keanggotaan dan derajat kebenaran sehingga suatu pernyataan dapat menjadi sebagian benar dan sebagian salah pada waktu yang sama. (Lotfi Zadeh, 1994)

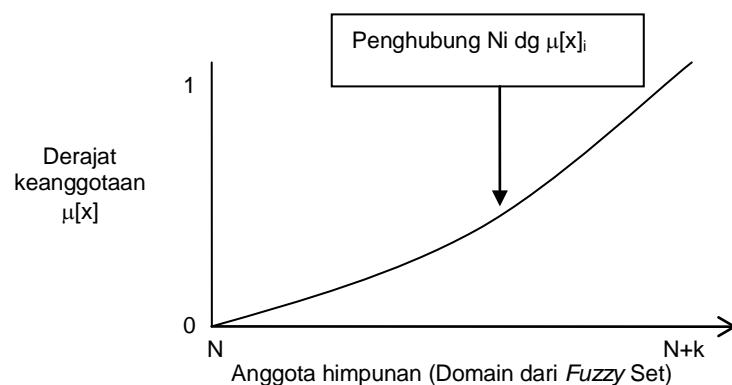
### 2.5.1 Konsep utama fuzzy

#### A. Prinsip ketidakpastian

Beberapa ilmu matematika terkadang sulit untuk dipastikan, seperti teori probabilitas. Hal ini bisa diklasifikasikan berdasar tipe ketidakpastian yang dilakukan. Ada beberapa tipe ketidakpastian, dua diantaranya adalah *Stochastic Uncertainty* dan *Lexical Uncertainty*.

*Stochastic Uncertainty* berhubungan dengan arah ketidakpastian dari kejadian yang pasti. Sedangkan *Lexical Uncertainty* merupakan ketidakpastian yang diungkapkan oleh kata-kata manusia, seperti “orang yang tinggi”, “hari yang panas” dan sebagainya.

#### B. Himpunan fuzzy



Gambar 2.1. Himpunan *Fuzzy* (*fuzzy sets*)



Gambar 2.1 menjelaskan tentang himpunan *fuzzy* yang terdiri atas 3 bagian, dimana sumbu horisontal menunjukkan himpunan anggota, sumbu vertikal menunjukkan derajat dari keanggotaan, dan garis yang menghubungkan masing-masing titik dari anggota dengan derajat keanggotaan yang tepat. Himpunan *fuzzy* akan dibahas lebih lanjut pada bagian 2.5.4.

### C. Fungsi keanggotaan (*Membership Function*)

Derajat dimana angka teknis bernilai sesuai konsep bahasa dari kondisi variabel bahasa (*linguistic*) dinamakan sebagai derajat keanggotaan. Untuk variabel berlanjut (*continous variable*) derajat ini disebut fungsi keanggotaan. Fungsi keanggotaan akan dibahas lebih lanjut pada bagian 2.5.5.

### D. Variabel linguistik

*Fuzzy logic* pada dasarnya menitikberatkan pada pengukuran dan penalaran tentang kekaburan atau bentuk *fuzzy* yang nampak dalam bahasa alami. Dalam *fuzzy* bentuk *fuzzy* dinyatakan sebagai variabel linguistik (disebut juga variabel *fuzzy*).

Variabel linguistik adalah bentuk yang digunakan dalam bahasa alami untuk menggambarkan beberapa konsep yang biasanya mempunyai kekaburan atau nilai *fuzzy*. Sebagai contoh dalam pernyataan “Jack adalah muda” menyatakan bahwa variabel linguistik umur mempunyai nilai linguistik muda.

Seperti halnya variabel aljabar yang berisi angka sebagai nilainya maka variabel linguistik menggunakan kata dan kalimat sebagai nilainya. Misalnya: jika  $T$  variabel linguistik yang berisi himpunan umur, maka isi  $T$  yang juga

merupakan himpunan *fuzzy* adalah:  $T = \{sangat\ tua, tua, setengah\ baya, agak\ muda, muda, sangat\ muda\}$ .

## E. Aturan fuzzy

Aturan dari sistem *fuzzy* (*Fuzzy System*) menggambarkan pengetahuan dari sistem. Mereka menggunakan variabel linguistik sebagai bahasanya, sebagai contoh untuk mengekspresikan strategi pengendalian dari sebuah pengendali pengontrol *fuzzy logic*. Menjelaskan aturan *fuzzy logic* berarti menunjukkan, bagaimana menghitung dengan konsep linguistik.

### 2.5.2 Perhitungan fuzzy

#### A. Fuzzyfikasi

Proses fuzzyfikasi merupakan proses untuk mengubah variabel non *fuzzy* (variabel *numeric*) menjadi variabel *fuzzy* (variabel *linguistic*). Nilai masukan-masukan yang masih dalam bentuk variabel *numeric* yang telah dikwantisasi sebelum diolah oleh pengendali logika *fuzzy* harus diubah terlebih dahulu kedalam variabel *fuzzy*. Melalui fungsi keanggotaan yang telah disusun maka dari nilai-nilai masukan tersebut menjadi informasi *fuzzy* yang berguna nantinya untuk proses pengolahan secara *fuzzy* pula. Proses ini disebut *fuzzyfikasi*.

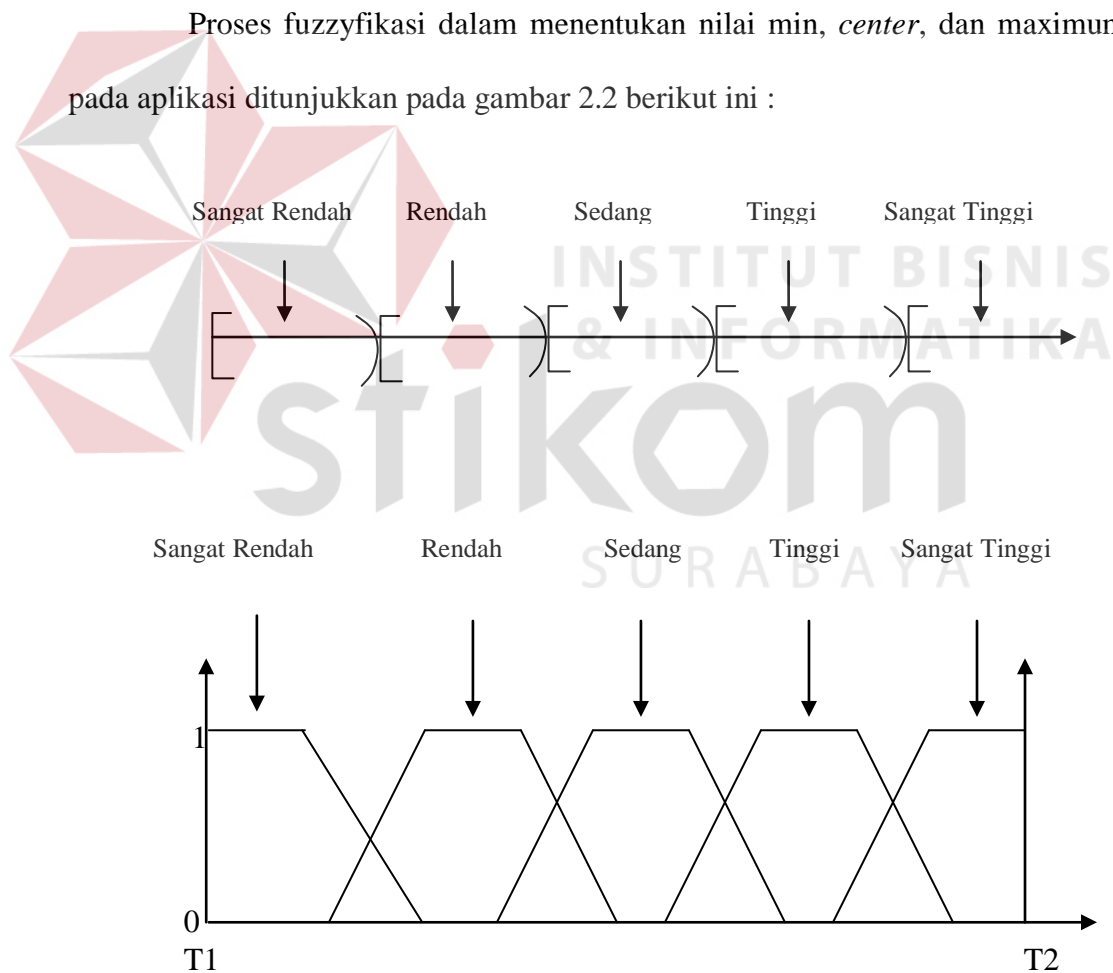
Dengan kata lain fuzzyfikasi merupakan pemetaan titik *numeric* (*crisp points*)  $\underline{x}(x^1, \dots, x^n)^T \in U$  ke himpunan *fuzzy*  $A$  di  $U$ .  $U$  adalah semesta pembicaraan. Paling tidak ada dua kemungkinan pemetaan, yaitu: Fuzzyfikasi *singlenton* :  $A$  adalah *fuzzy singlenton* dengan support  $x$ , artinya,  $\mu_A(x') = 1$  untuk  $x' = x$  dan  $\mu_A(x') = 0$  untuk selain  $x' \in U$  dengan  $x' \neq x$  [1]. Fuzzyfikasi

*nonsingleton* :  $\mu_A(c)=1$  dan  $\mu_A(x')$  menurun dari 1 sebagaimana  $x'$  bergerak menjauh dari  $x$ . sebagai contoh :

$$\mu_A(A') = \exp\left[-\frac{(x'-x)^T(x'-x)}{\sigma^2}\right] \dots\dots\dots (10)$$

dimana  $\sigma'$  adalah parameter yang menentukan bentuk dari  $\mu_A(x')$  [1]. Sejah ini yang banyak digunakan adalah *fuzzyfikasi singleton*, tetapi pemakaian non singleton juga telah dirintis terutama untuk masukan-masukan yang banyak dimasuki oleh derau (*noise*).

Proses fuzzyfikasi dalam menentukan nilai min, *center*, dan maximum pada aplikasi ditunjukkan pada gambar 2.2 berikut ini :



Gambar 2.2 Variabel Fuzzy

## B. Inferensi fuzzy

Dalam inferensi *fuzzy* dilakukan proses yang dinamakan evaluasi *rule*. Tahap ini digunakan untuk mencari derajat kebenaran (*rule strength*) dari masukan *fuzzy* yang nilai keanggotaannya telah ditentukan sebelumnya pada proses fuzzyfikasi. Struktur dasar dari sistem inferensi *fuzzy* terdiri dari basis aturan yang berisi aturan *if-then*, basis data yang mendefinisikan fungsi keanggotaan dari himpunan *fuzzy*.

### 2.5.3 Himpunan fuzzy

Teori himpunan tradisional menggambarkan dunia sebagai hitam dan putih. Ini berarti sebuah obyek berada didalam atau diluar himpunan yang diberikan. Dalam teori himpunan tradisional untuk anggota diberi nilai 1 dan untuk bukan anggota diberi nilai 0; ini disebut himpunan *crisp*. Sebagai contoh anggota himpunan orang muda dapat berisi hanya orang yang berumur kurang dari 10. Penggunaan interpretasi ini pada seseorang yang berulang tahun ke-11, maka orang tersebut bukan anggota himpunan orang muda.

Himpunan *fuzzy* memberikan nilai keanggotaan antara 0 dan 1 yang menggambarkan secara lebih alami sebuah kumpulan anggota dengan himpunan, Sebagai contoh, jika seorang berumur 5 tahun dapat diberikan nilai keanggotaan 0.9 atau jika umurnya 13 tahun nilai keanggotaannya 0.1. Dalam contoh ini “umur” adalah variabel linguistik dan “muda” adalah salah satu himpunan *fuzzy*.

Himpunan *fuzzy* dapat didefinisikan sebagai berikut : misalkan  $X$  semesta pembicaraan, dengan elemen dari  $X$  dinotasikan  $x$ . Sebuah himpunan *fuzzy*  $A$  dari  $X$  dikarakteristikan dengan fungsi keanggotaan  $\mu_A(x) : X \rightarrow [0,1]$ .

Pada *fuzzy*, kejadian atau elemen  $x$  diberikan nilai keanggotaan dengan fungsi keanggotaan  $\mu$ . Nilai ini mempresentasikan derajat keanggotaan elemen  $x$  pada himpunan *fuzzy*  $A$ .  $\mu_A(x) = \text{Degree}(x \in A)$  nilai keanggotaan dari  $x$  berada pada interval :  $0 \leq \mu_A(x) \leq 1$

Himpunan *fuzzy* adalah perluasan dari teori himpunan tradisional. Himpunan *fuzzy* menyamakan konsep keanggotaan dengan menggunakan fungsi keanggotaan  $\mu$  yang menghasilkan nilai antara 0 dan 1 yang mempresentasikan derajat keanggotaan obyek  $x$  pada himpunan  $A$ .

Untuk mempresentasikan himpunan *fuzzy* dalam komputer perlu didefinisikan fungsi keanggotaannya. Sebagai contoh : orang tinggi. Dapat dinyatakan pada setiap individu, pada tingkatan mana bahwa mereka yakin seseorang itu dikatakan tinggi. Setelah mengumpulkan jawaban untuk interval ukuran tinggi, dapat disajikan tingkat rata-rata untuk menghasilkan suatu himpunan *fuzzy* dari orang-orang yang tinggi. Fungsi ini dapat digunakan sebagai suatu keyakinan (nilai keanggotaan). Bagi individu yang menjadi anggota himpunan *fuzzy* dari orang tinggi.

Dengan membentuk *fuzzy* subset untuk berbagai bentuk *fuzzy*, dianggap nilai keanggotaan dari obyek yang diberikan pada setiap himpunan. Pendekatan lain yang sering ditemukan pada praktek untuk membentuk himpunan *fuzzy* sangat berhubungan dengan interpretasi dari seorang ahli. Seperti teknik pengumpulan data, dapat ditanyakan pada pakar untuk kepercayaannya bahwa berbagai obyek merupakan bagian himpunan yang diberikan.

### 2.5.4 Operasi himpunan fuzzy

Terdapat 3 operasi dalam himpunan *fuzzy*, yaitu :

#### A. Irisan (*Intersection*)

Dalam teori himpunan klasik, irisan dari dua himpunan berisi elemen-elemen yang sama dari keduanya. Dalam himpunan *fuzzy*, sebuah elemen mungkin sebagian dalam kedua himpunan. Oleh karena itu ketika mengingat irisan dari kedua himpunan, tidak dapat dikatakan bahwa sebuah elemen adalah lebih mungkin menjadi dalam irisan daripada dalam suatu himpunan asli.

#### B. Gabungan (*Union*)

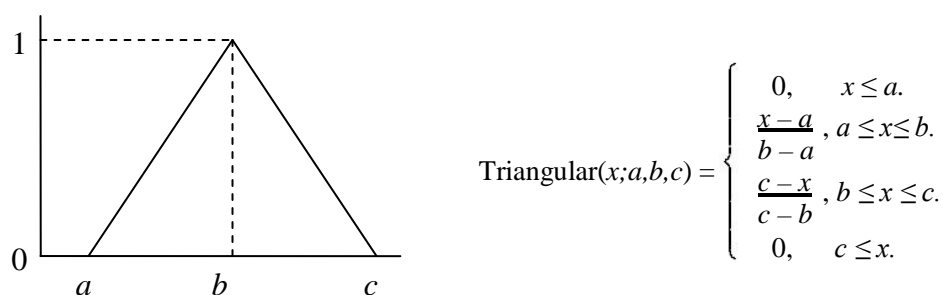
Cara kedua dari penggabungan himpunan *fuzzy* adalah gabungannya. Penggabungan dari dua himpunan adalah terdiri dari dua himpunan adalah terdiri dari elemen-elemen yang menjadi satu atau dua himpunan. Dalam situasi ini anggota dari gabungan tidak dapat mempunyai nilai keanggotaan yang kurang dari nilai keanggotaan yang lain dari himpunan aslinya.

#### C. Komplemen (*Complement*)

Komplemen dari himpunan *fuzzy* A dinotasikan dengan ( $\sim A$ ) dinyatakan dengan persamaan sebagai berikut :  $\mu_{\sim A}(x) = 1 - \mu_A(x)$

### 2.5.5 Fungsi keanggotaan (*Membership Function*)

Fungsi keanggotaan segitiga dijelaskan sebagai berikut :



Gambar 2.3. Fungsi keanggotaan segitiga

Gambar 2.3 menjelaskan tentang fungsi keanggotaan yang digunakan dalam mempresentasikan himpunan *fuzzy*. Dalam *fuzzy* fungsi keanggotaan yang biasa dipakai adalah fungsi keanggotaan segitiga, trapesium, Gaussian, fungsi keanggotaan S, fungsi keanggotaan lonceng dan sebagainya. Dalam sistem ini fungsi keanggotaan digunakan adalah fungsi keanggotaan segitiga :

### 2.5.6 Batasan (*Hedges*)

Dalam pembicaraan normal, manusia mungkin menambahkan kekaburan untuk memberikan pernyataan dengan menggunakan kata keterangan seperti sangat, agak. Kata keterangan adalah sebuah kata yang memodifikasi kata benda, kata sifat, kata keterangan lain, atau keseluruhan kalimat. Sebagai contoh, kata keterangan memodifikasi kata sifat, “orang itu sangat tinggi”. Sebuah *hedges* memodifikasi himpunan *fuzzy* yang sudah ada secara matematis untuk menghitung beberapa kata keterangan yang ditambahkan.

## 2.6. Data Flow Diagram (DFD)

Analisa sistem informasi merupakan tahap penguraian dari sistem informasi yang utuh ke dalam sub sistem yang dimaksud, mengidentifikasi permasalahan yang ada serta kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan. DFD digunakan untuk merepresentasikan seluruh proses-proses data dalam sebuah organisasi. Pendekatan data *flow* menitik beratkan pada logika yang tersirat dari suatu sistem.

### 2.6.1 Pembuatan DFD (Data Flow Diagram)

DFD dapat dan harus digambarkan secara sistematis. Pertama, dibutuhkan sistem analisis untuk mengkonsep data *flow*, dari atas ke bawah seperti dijelaskan sebagai berikut :

1. Membuat sebuah daftar dari kegiatan bisnis dengan menggunakan beberapa variasi, yaitu : (a) *Entity* luar (*external entity*), (b) *Data flow*, (c) Proses, dan (d) *Data store*.
2. Membuat sebuah context diagram dimana ditunjukkan *external entity* dan *data flow* ke dan dari sistem.
3. Menggambar diagram level 0, level selanjutnya.
4. Membuat sebuah *child* diagram untuk tiap-tiap proses pada level 0.
5. Pengecekan *error*.

Untuk memulai sebuah DFD dari suatu sistem biasanya dituangkan dalam sebuah daftar dengan empat kategori yaitu *entity* luar, arus data, proses, dan penyimpanan data. Daftar ini akan membantu menentukan batasan-batasan dari suatu sistem yang akan digambarkan. Pada dasarnya daftar tersebut berisi elemen-elemen data yang dikarang yang terdiri dari :



a. Pembuatan *context* diagram

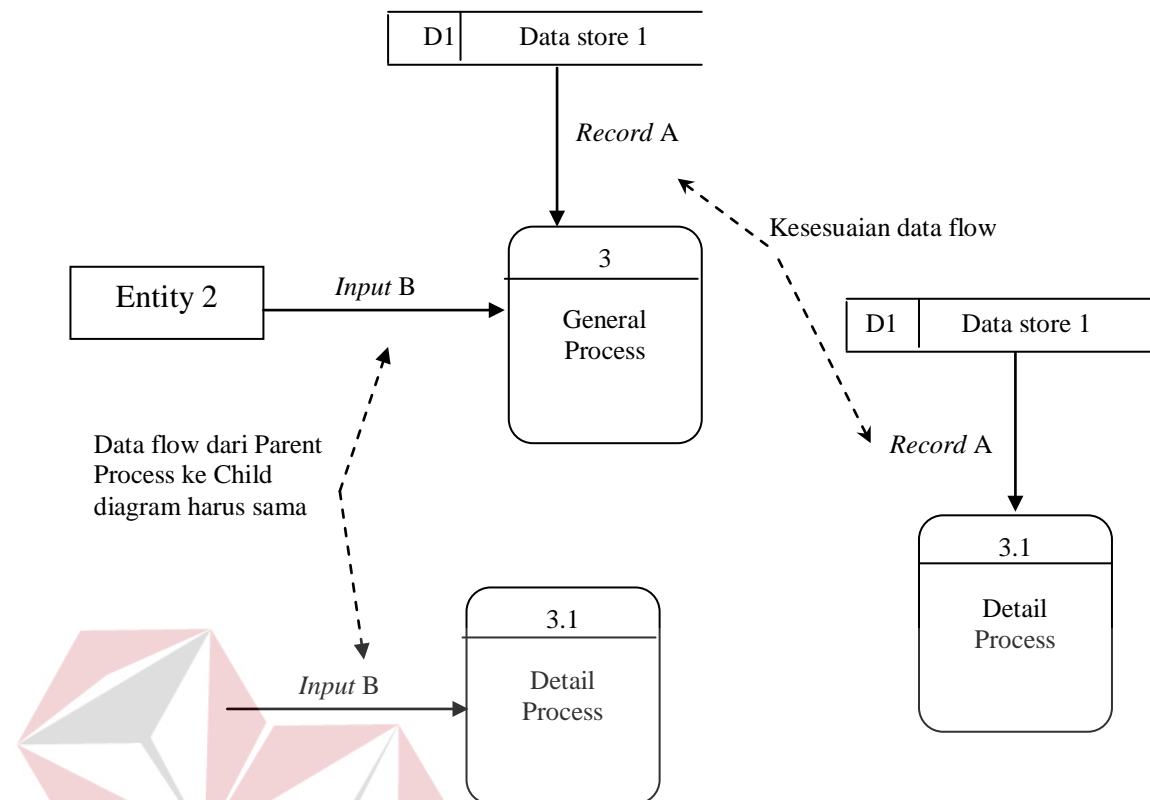
*Context* diagram adalah level yang tertinggi dalam sebuah DFD dan hanya berisi satu proses serta merupakan representasi dari sebuah sistem. Proses dimulai dengan penomoran ke-0 dan untuk seluruh *entity* luar akan ditunjukkan dalam *context* diagram yang sama seperti data awal yang dikirim dari *entity* luar. *Context* diagram tidak berisi penyimpanan data.

b. Pembuatan diagram level 0 serta level berikutnya

Diagram level 0 dihasilkan oleh *context* diagram dan berisi proses-proses. Pengisian proses-proses yang berlebihan pada level ini akan menghasilkan sebuah diagram yang salah, sehingga sulit untuk dimengerti. Masing-masing proses diberikan penomoran dengan sebuah bentuk *integer*. Umumnya dimulai dari kiri atas dan penyelesaiannya di kanan bawah dalam sebuah bentuk diagram.

c. Pembuatan *child* diagram

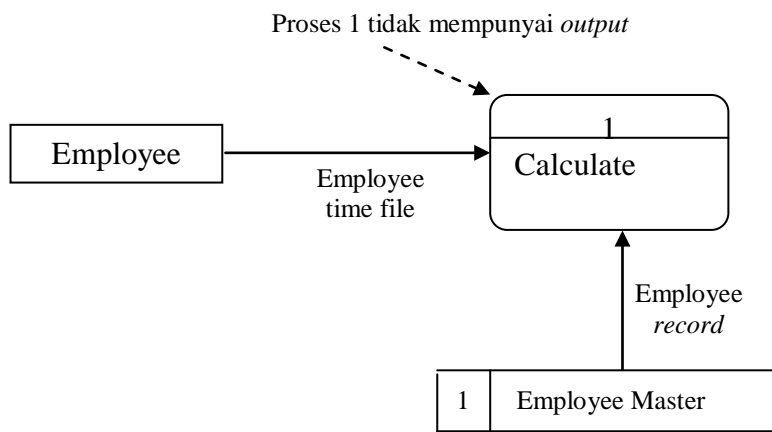
*Child* diagram diberikan nomor yang sama seperti proses di atasnya (parent proses) dalam diagram level 0. Contohnya, proses 3 harus diturunkan ke diagram 3, proses pada *child* diagram menggunakan penomoran unik untuk masing-masing proses dengan mengikuti penomoran proses di atasnya. Contohnya, dalam diagram 3 proses-proses diberikan nomor 3.1, 3.2, 3.3 dan seterusnya. Konversi ini diikuti oleh analisis sistem untuk menelusuri seri-seri dari proses-proses yang dikeluarkan oleh beberapa level, jika pada proses diagram level 0 digambarkan sebagai 1, 2 dan 3 maka *child* diagram-diagramnya adalah 1, 2, dan 3 pada level yang sama. Ilustrasi level detil dengan sebuah *child* DFD dapat ditunjukkan pada gambar 2.4 :



Gambar 2.4. Contoh ilustrasi detail child diagram

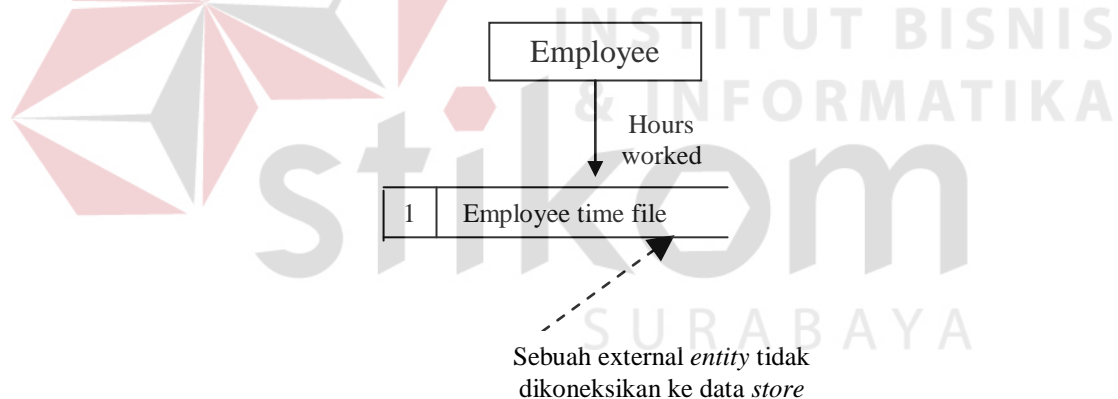
d. Pengecekan kesalahan-kesalahan pada diagram digunakan untuk melihat kesalahan-kesalahan yang terdapat pada sebuah DFD. Beberapa kesalahan-kesalahan yang umum terjadi ketika penggambaran / pembuatan DFD sebagai berikut :

1. Tidak menginputkan sebuah arus data atau arah panah langsung. Sebagai contoh adalah penggambaran proses yang menunjukkan sebuah data *flow* seperti *input* atau seperti *output*. Tiap-tiap proses perubahan data harus menerima *input* dan *output*. Tipe kesalahan ini terjadi ketika sistem analis tidak memasukkan sebuah data *flow* atau meletakkan sebuah arah panah di tempat yang salah seperti ditunjukkan pada gambar 2.5 berikut ini :



Gambar 2.5 Kesalahan proses input dan output

2. Hubungan penyimpanan data dan *entity* luar secara langsung satu sama lain. Data *store* dan *entity* luar harus dikoneksikan melalui sebuah proses seperti ditunjukkan pada gambar 2.6 berikut ini :



Gambar 2.6 Kesalahan penulisan hubungan entity luar dengan data store

3. Kesalahan penamaan (label) pada proses-proses atau data *flow*. Pengecekan DFD untuk memastikan bahwa tiap-tiap obyek atau data *flow* telah diberikan label. Sebuah proses haruslah diindikasikan seperti nama dari sistem atau menggunakan format kata kerja. Tiap data *flow* haruslah dideskripsikan dengan sebuah kata benda.

4. Memasukkan lebih dari sembilan proses dalam sebuah DFD. Memiliki banyak proses akan mengakibatkan kekacauan pada diagram sehingga dapat menyebabkan kebingungan dalam pembacaan sebuah proses dan akan menghalangi tingkat komunikasi. Jika lebih dari sembilan proses dalam sebuah sistem, maka beberapa grup dalam proses dilakukan bersama-sama ke dalam sebuah sub sistem dan meletakkannya dalam sebuah *child* diagram.
5. Menghilangkan suatu arus data. Pengujian dari suatu diagram yang menunjukkan garis / arah (*flow*), dimana untuk setiap proses data *flow* hanya mempunyai *input* data, *output* kecuali dalam kasus dari detil (*child*). Setiap *child* data dari DFD, arah arus data seringkali digambarkan untuk mengidentifikasi bahwa diagram tersebut kehilangan data *flow*.
6. Membuat ketidaksesuaian komposisi dalam *child* diagram , dimana tiap *child* diagram harus mempunyai *input* dan *output* arus data yang sama seperti proses di level atasnya (*parent process*). Pengecualian untuk *rule* ini adalah kurangnya *output*, seperti kesalahan garis yang ada didalam *child* diagram.

### 2.6.2 Keuntungan pembuatan data flow

Data *flow* mempunyai lima keuntungan utama dari penjelasan-penjelasan jalannya data dalam sistem, yaitu :

1. Kebebasan yang berasal dari kepercayaan untuk mengimplementasikan secara benar teknik sistem dari suatu sistem yang baru.
2. Memberikan pengertian dari hubungan sistem-sistem dan subsistem yang ada.
3. Komunikasi mengenai pengetahuan sistem bagi *user* melalui DFD
4. Analisa dari sebuah usulan sistem untuk menentukan jika data dan proses-proses yang ada dapat didefinisikan secara mudah.

5. Penggunaan data *flow* merupakan keuntungan tambahan yang dapat digunakan sebagai latihan bagi sistem analis, kesempatan sistem analis menjadi lebih mengerti tentang hubungan sistem dan subsistem yang ada didalamnya.

## 2.7. Desain Database

Desain *database* merupakan gambaran atau deskripsi dari file-file yang digunakan serta merupakan sebuah pendefinisian normal dan merupakan gambaran dari pusat penyimpanan dari data tertentu yang digunakan dalam beberapa aplikasi yang berbeda. Desain *database* terdiri dari :

### 2.7.1 Database

*Database* bukan hanya merupakan sebuah koleksi dari suatu file-file. Meskipun, sebuah *database* merupakan sebuah pusat sumber data yang disimpan oleh beberapa *user* dari sebuah aplikasi-aplikai yang bervariasi. Inti dari sebuah *database* adalah DBMS (*Database Management Sistem*), dimana diikuti dengan kreasi, modifikasi, dan perubahan (*update*) dari *database*.

### 2.7.2 Bentuk database dan file

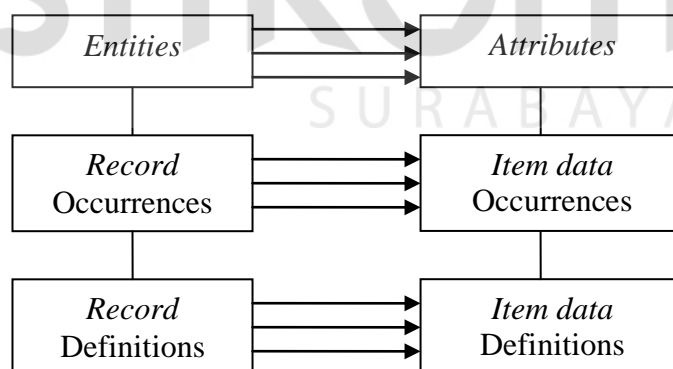
Ada dua bentuk pendekatan pada proses penyimpanan data dalam sebuah sistem komputer. Metode pertama adalah untuk menyimpan file-file tunggal, masing-masing dengan bentuk unik untuk berbagai macam aplikasi. Pendekatan yang kedua adalah untuk menyimpan data dalam sebuah sistem komputer dengan melibatkan pembuatan sebuah *database*.

### 2.7.3 Konsep-konsep data

Konsep-konsep data merupakan hal yang sangat penting untuk di mengerti bagaimana data dipresentasikan kembali sebelum memutuskan penggunaan *file* atau *database*. Konsep-konsep tersebut terdiri dari :

#### A. Realita, data dan metadata

Realita data merupakan gambaran atribut-atribut yang dimiliki oleh sekumpulan data. Data merupakan representasi fakta dunia nyata yang mewakili suatu objek. Metadata merupakan data dimana dijelaskan tentang data *file* atau *database* serta menjelaskan pemberian nama dan menunjukkan panjang dari masing-masing item data. Metadata juga menjelaskan panjang dan komposisi dari tiap-tiap *record*. Gambaran dari hubungan antara realita, data dan meta data yaitu didalamnya terdapat *entity* dan atribut (realita), *record* dan *item data* (data), definisi *record* dan definisi *item data* (meta data). Ditunjukkan pada gambar 2.8 berikut :



Gambar 2.7 Hubungan antara realita, data dan metadata

#### 1. Entity

Merupakan beberapa obyek atau kejadian tentang dimana dapat mencocokkan koleksi data sebagai sebuah *entity*. *Entity* juga dapat berupa sebuah kejadian atau unit dari satu waktu.

## 2. *Relationship*

Merupakan hubungan antara *entity*, adapun tipe-tipe dari *relationship* antara lain *one to one* (1:1), *one to many* (1:M), dan *many to many* (M:N).

## 3. *Attribute*

Merupakan sebuah karakteristik dari sebuah *entity*.

## 4. *Record*

Merupakan suatu kumpulan dari item-item data yang secara umum merupakan penjelasan umum dari *entity*.

## B. Kelompok *file*

Sebuah *file* berisi grup-grup dari *record* yang digunakan untuk melengkapi informasi untuk suatu operasi-operasi, perencanaan, manajemen, dan pembuatan keputusan. Tipe-tipe dari *file* yang digunakan antara lain :

1. *File master* : berisi *record-record* dari sebuah kelompok *entity*.
2. *File tabel* : berisi data yang digunakan.
3. *File-file transaksi* : digunakan untuk mengisi perubahan (*update*) sebuah *file master* dan laporan-laporan.
4. *Work file* : digunakan untuk menjalankan program agar lebih efektif.
5. *File laporan* : Memudahkan untuk menjalankan program (ketika tidak ada printer).

## C. Normalisasi

Normalisasi dibutuhkan untuk mengorganisir data dan menghindari redundansi data (*data double*). Ada tiga bentuk normalisasi, antara lain :

### 1. *First Normal Form* (1NF)

Langkah pertama ini terdapat pada sebuah relasi normalisasi yang digunakan untuk menghilangkan (menghapus) grup-grup yang berulang.

### 2. *Second Normal Form* (2NF)

Pada bentuk normal yang kedua, seluruh attribut yang ada akan difungsikan tergantung pada PK (*Primary Key*).

### 3. *Third Normal Form* (3NF)

Sebuah relasi penormalisasian yang ketiga adalah jika seluruh yang bukan kunci PK (*Primary Key*) dari semua attribut seluruhnya difungsikan bergantung pada PK dan attribut tersebut bukan transitif ketergantungan (tanpa kunci).

## 2.8. Desain Output dan Input

### 2.8.1 Input

*Input* merupakan sebuah informasi yang dikirimkan kepada suatu sistem informasi. Bentuk-bentuk desain secara khusus mungkin dapat digunakan jika suatu analisa sistem dapat disesuaikan dengan bentuk desain secara lengkap dan bermanfaat. Hal itu juga penting untuk mengenalkan secara dini desain yang digunakan, arus data atau bentuk-bentuk yang tidak dibutuhkan pada sumber-sumber suatu organisasi dan hal itu harus dihilangkan. Untuk mendesain *form-form* yang baik dan berguna, ada beberapa hal dari desain *form* yang harus diterapkan antara lain :

1. Membuat *form-form* tersebut mudah dalam pengisiannya.



2. Memastikan *form-form* tersebut sesuai dengan tujuan untuk masing-masing desain.
3. Desain *form* digunakan untuk menjamin kelengkapannya.
4. Mempertahankan *form-form* yang menarik.

### 2.8.2 Output

*Output* merupakan sebuah informasi yang dikirimkan kepada *user* melalui suatu sistem informasi. Beberapa data yang dibutuhkan diproses secara teliti sebelum dinyatakan layak sebagai sebuah *output*; penyimpanan data lain, dan ketika data-data tersebut dibutuhkan kembali, data-data tersebut berupa *ouput* yang membutuhkan proses yang sedikit. *Output* dapat di ambil dari beberapa bentuk yaitu : bentuk laporan yang dihasilkan printer dan tampilan layar komputer, mikrofon (suara). Oleh sebab itu, penggunaan *output* sangat penting guna menjamin pemakaian dan penerimaan dari suatu sistem informasi.