BAB 2

TINJAUAN PUSTAKA

2.1 Kebangkrutan Perusahaan

Kebangkrutan adalah suatu kondisi disaat perusahaan mengalami ketidakcukupan dana untuk menjalankan usahanya. Menurut Undang-Undang Kepailitan No. 4 Tahun 1998, debitur yang mempunyai dua atau lebih kreditur dan tidak membayar sedikitnya satu utang yang telah jatuh waktu dan dapat ditagih, dinyatakan pailit dengan putusan pengadilan yang berwenang, baik atas permohonannya sendiri, maupun atas permintaan seorang atau lebih krediturnya (Yani dan Widjaja, 2004:153).

Emiten atau perusahaan publik yang gagal atau tidak mampu menghindari kegagalan untuk membayar kewajibannya terhadap pemberi pinjaman yang tidak terafiliasi, maka emiten atau perusahaan publik wajib menyampaikan laporan mengenai pinjaman termasuk jumlah pokok dan bunga, jangka waktu pinjaman, nama pemberi pinjaman, penggunaan pinjaman dan alasan kegagalan atau ketidakmampuan menghindari kegagalan kepada Badan Pengawas Pasar Modal (BAPEPAM) dan Bursa Efek di mana efek emiten atau perusahaan publik tercatat secepat mungkin, paling lambat akhir hari kedua sejak emiten mengalami kegagalan atau mengetahui ketidakmampuan untuk menghindari kegagalan dimaksud (Yani dan Widjaja, 2004:14).

Kegagalan dapat didefinisikan dalam beberapa cara, dan kegagalan tidak harus menyebabkan keruntuhan atau pembubaran perusahaan. Kegagalan ekonomis berarti bahwa pendapatan perusahaan tidak mampu menutup biayanya sendiri. Sedangkan kegagalan keuangan berarti jika perusahaan tidak dapat memenuhi kewajiban pada waktunya harus dipenuhi, walaupun harta totalnya melebihi kewajiban totalnya (Weston dan Brigham, 2000:145).

2.1.1 Penyebab Kegagalan Perusahaan

Menurut R. Agus Sartono (1994:87), ada tiga jenis kegagalan perusahaan yaitu:

1. Perusahaan yang menghadapi *technically insolvent*, jika perusahaan tidak dapat memenuhi kewajibannya yang segera jatuh tempo tetapi asset perusahaan nilainya lebih tinggi daripada utangnya.
2. Perusahaan yang menghadapi *legally insolvent*, jika nilai asset perusahaan lebih rendah daripada nilai utang perusahaan.
3. Perusahaan yang menghadapi kebangkrutan yaitu jika tidak dapat membayar utangnya dan oleh pengadilan dinyatakan pailit.

Sebab utama kegagalan sebuah perusahaan adalah manajemen perusahaan yang kurang kompeten. Sementara menurut Bambang Riyanto (2001:315) faktor-faktor yang merupakan penyebab kegagalan suatu perusahaan pada prinsipnya dapat digolongkan menjadi dua yaitu:

1. Sebab internal adalah sebab-sebab yang timbul dari dalam perusahaan itu sendiri, yang meliputi sebab finansial maupun non finansial.
    a. Sebab-sebab yang menyangkut bidang finansial.
        1) Adanya utang yang terlalu besar sehingga memberikan beban tetap yang berat bagi perusahaan.
        2) Adanya liablitias lancar (*current liabilities*) yang terlalu besar diatas aset lancar (*current assets*).
        3) Lambatnya pengumpulan piutang atau banyaknya piutang tak tertagih (*bad-debts*).
        4) Kesalahan dalam kebijakan dividen (*dividend policy*).
        5) Tidak cukupnya dana-dana penyusutan.
    b. Sebab-sebab yang menyangkut bidang non finansial.
        1) Adanya kesalahan pada para pendiri perusahaan, yaitu antara lain:
            a) Kesalahan dalam pemilihan tempat kedudukan perusahaan.
            b) Kesalahan dalam penentuan produk yang dihasilkan.
            c) Kesalahan dalam penentuan besarnya perusahaan.
        2) Kurang baiknya struktur organisasi perusahaan.
        3) Kesalahan dalam pemilihan pimpinan perusahaan.
        4) Keterbatasan kemampuan manajemen dalam menjalankan kegiatan bisnis perusahaan (*managerial incompetence*).
            a) Kesalahan dalam kebijakan pembelian.

b) Kesalahan dalam kebijakan produksi.
c) Kesalahan dalam kebijakan marketing.
d) Adanya ekspansi yang berlebih-lebihan.

2. Sebab eksternal adalah sebab-sebab yang timbul atau berasal dari luar perusahaan dan yang berada diluar kekuasaan atau kontrol dari pimpinan perusahaan atau badan usaha, yaitu antara lain:
   a. Adanya persaingan yang hebat.
   b. Berkurangnya permintaan terhadap produk yang dihasilkannya.
   c. Turunnya harga-harga, dan lain sebagainya.

2.1.2 Analisis Laporan Keuangan

Laporan keuangan pada dasarnya adalah hasil dari proses akuntansi yang dapat digunakan sebagai alat untuk berkomunikasi antara data keuangan atau aktivitas suatu perusahaan dengan pihak-pihak yang berkepentingan dengan data atau aktivitas perusahaan tersebut (Munawir 2006:2).

2.1.2.1 Laporan Keuangan

Pada sub bab ini akan dibahas mengenai pengertian laporan keuangan, tujuan laporan keuangan, pengguna laporan keuangan, serta komponen dan unsur laporan keuangan.

2.1.2.1.1 Pengertian Laporan Keuangan

Pengertian laporan keuangan menurut Ikatan Akuntan Indonesia (IAI) pada Pernyataan Standar Akuntansi Keuangan No. 1 (2009:1.5) adalah suatu penyajian terstruktur dari posisi keuangan dan kinerja keuangan suatu entitas.

Laporan keuangan menurut Riyanto (2004:70) merupakan bagian dari proses pelaporan keuangan, yang meliputi laporan neraca, laporan laba rugi, laporan perubahan posisi keuangan dan laporan lain serta materi penjelasan yang merupakan bagian integral dari laporan keuangan.

2.1.2.1.2 Tujuan Laporan Keuangan

Tujuan laporan keuangan menurut Kieso *et al*. (2012:5) adalah
"*Provide financial information about the reporting entity that is useful to present and potential equity investors, lenders, and other creditors in decisions about providing resources to the entity*."

Tujuan dari laporan keuangan adalah untuk memberikan informasi keuangan tentang entitas pelapor yang berguna untuk menyajikan dan calon investor ekuitas pemberi pinjaman, dan kreditur lainnya dalam pengambilan keputusan tentang penyediaan sumber daya untuk entitas

Adapun tujuan laporan keuangan menurut IAI yang tertuang dalam PSAK No.1 (2009) adalah memberikan informasi mengenai posisi keuangan, kinerja keuangan, dan arus kas entitas yang bermanfaat bagi sebagian besar kalangan pengguna laporan keuangan dalam pembuatan keputusan ekonomi. Laporan keuangan juga menunjukkan hasil pertanggungjawaban manajemen atas penggunaan sumber daya yang dipercayakan kepada mereka.

2.1.2.1.3 Pengguna Laporan Keuangan

Secara umum, pengguna laporan keuangan menurut Stice *et al*. (2004:10) dapat dikategorikan pada dua kategori, yaitu:

1. Pengguna internal, yaitu pengguna laporan keuangan yang menggunakan laporan keuangan untuk pengambilan keputusan secara langsung dan mempunyai pengaruh terhadap kegiatan internal perusahaan.
2. Pengguna eksternal, yaitu pengguna laporan keuangan yang menggunakan laporan untuk pengambilan keputusan berkaitan dengan hubungan mereka dengan perusahaan.

Menurut Harahap (2007:120) Pengguna Laporan Keuangan sebagai berikut: "Pemegang saham, Investor, Analis pasar modal, Manajer, Karyawan dan serikat pekerja, Instansi pajak, Pemberi dana (kreditur), Supplier, Pemerintah atau lembaga pengatur resmi, Langganan atau lembaga konsumen, Lembaga swadaya masyarakat, Peneliti/Akademis/Lembaga peringkat."

Berdasarkan penjelasan di atas dapat disimpulkan bahwa para pengguna laporan keuangan memiliki kepentingannya masing-masing. Namun, secara umum para pengguna laporan keuangan menggunakan laporan keuangan untuk pengambilan keputusan

2.2.3.1.4 Komponen dan Unsur Laporan Keuangan

Berdasakan PSAK No.1 (IAI, 2009:1.6) laporan keuangan yang lengkap terdiri dari komponen-komponen berikut ini:

1. Laporan posisi keuangan (neraca) pada akhir periode;

2. Laporan laba rugi komprehensif selama periode;
3. Laporan perubahan ekuitas selama periode;
4. Laporan arus kas selama periode;
5. Catatan atas laporan keuangan, berisi ringkasan kebijakan akuntansi penting dan informasi penjelasan lain; dan
6. Laporan posisi keuangan pada awal periode komparatif yang disajikan ketika entitas menerapkan suatu kebijakan akuntansi secara retrospektif atau membuat penyajian kembali pos-pos laporan keuangan, atau ketika entitas mereklasifikasi pos-pos dalam laporan keuangannya.

2.1.3 Analisis Rasio Keuangan

Rasio menggambarkan suatu hubungan atau perimbangan antara suatu jumlah tertentu dengan jumlah yang lain, dan dengan menggunakan alat analisis berupa rasio akan dapat memberikan gambaran kepada penganalisis tentang baik atau buruknya keadaan tentang posisi keuangan suatu perusahaan terutama apabila dibandingkan dengan angka rasio pembanding yang digunakan sebagai standar (Munawir, 2006:64).

Menurut Riyanto (2001:329) penganalisis keuangan dalam mengadakan analisis rasio keuangan pada dasarnya dapat melakukan dengan dua macam cara pembandingan yaitu:

1. Membandingkan rasio sekarang (*present ratio*) dengan rasio-rasio dari waktu-waktu yang lalu (*historical ratio*) atau dengan rasio-rasio yang diperkirakan untuk waktu- waktu yang akan datang dari perusahaan yang sama.
2. Membandingkan rasio-rasio dari suatu perusahaan (*company ratio*) dengan rasio-rasio semacam dari perusahaan lain yang sejenis atau industri untuk waktu yang sama.

2.1.3.1 Penggolongan Rasio Laporan Keuangan

Menurut Riyanto (2001:330) berdasarkan sumber datanya angka rasio digolongkan sebagai berikut:

1. Rasio neraca (*balance sheet ratios*) yaitu semua rasio yang datanya diambil atau bersumber pada neraca, misalnya *current ratio, acid-test ratio, cash ratio*, dan sebagainya.

2. Rasio laporan laba rugi (*income statement ratios*) yaitu semua rasio yang datanya diambil atau bersumber dari laporan laba rugi misalnya *gross profit margin*, *net operating margin*, *operating ratio* dan sebagainya.
3. Rasio antar laporan (*interestatement ratios*) yaitu semua rasio yang datanya diambil atau bersumber dari neraca dan data lainnya dari laporan laba rugi, misalnya tingkat perputaran persediaan (*inventory turnover*), tingkat perputaran piutang (*account receivable turnover*), *total assets turnover* dan sebagainya.

2.1.3.2 Jenis – Jenis Rasio Keuangan

Pada sub bab ini akan dibahas mengenai rasio likuiditas, rasio aktivitas, rasio *leverage* atau solvabilitas, dan rasio rentabilitias atau profitabilitas.

2.1.3.2.1 Rasio Likuiditas

Rasio likuiditas digunakan untuk mengukur kemampuan perusahaan untuk memenuhi kewajiban keuangan jangka pendek yang harus segera dipenuhi, atau kemampuan perusahaan untuk memenuhi kewajiban keuangan pada saat ditagih. Yang termasuk dalam rasio likuiditas yaitu (Riyanto, 2001:332) :

1. Rasio lancar (*current ratio*) digunakan untuk mengukur kemampuan perusahaan dalam memenuhi utang jangka pendeknya dengan menggunakan aktiva lancar.

$$\text{Rasio Lancar} = \text{Aktiva Lancar / Utang Lancar} \quad (1)$$

2. Rasio cepat (*quick ratio*) digunakan untuk mengukur kemampuan perusahaan dalam memenuhi utang jangka pendeknya dengan menggunakan aktiva lancarnya yang likuid, yaitu aktiva lancar diluar persediaan.

$$\text{Rasio Cepat} = \frac{\text{(Aktiva Lancar – Persediaan)}}{\text{Utang Lancar}} \quad (2)$$

3. Rasio kas (*cash ratio*) digunakan untuk mengukur kemampuan perusahaan dalam memenuhi utang jangka pendeknya dengan menggunakan aktiva lancarnya yang paling likuid.

$$\text{Rasio Kas} = \text{(Kas + Efek) / Utang Lancar} \quad (3)$$

4. Rasio modal kerja terhadap total aktiva (*working capital to total assets ratio*) menunjukkan potensi cadangan kas yang ada akibat selisih yang terjadi antara aktiva lancar dengan utang lancar.

$$\text{Rasio Modal Kerja Terhadap Total Aktiva} = \frac{\text{(Aktiva Lancar – Utang Lancar)}}{\text{Total Aktiva}} \quad (4)$$

2.1.3.2.2 Rasio Aktivitas

Rasio aktivitas digunakan untuk mengukur seberapa efektif perusahaan menggunakan sumber daya yang dimiliki, atau dengan kata lain sejauh mana efektivitas penggunaan asset dengan melihat tingkat aktivitas asset. Yang termasuk dalam rasio aktivitas diantaranya (Riyanto, 2001:333) :

1. Rasio tingkat perputaran persediaan menunjukkan tingkat efektivitas manajemen persediaan, yaitu menunjukkan lamanya dana tertanam dalam persediaan.

$$\text{Rasio Tingkat Perputaran Persediaan} = \frac{\text{Harga Pokok Penjualan}}{\text{Rata-Rata Persediaan}} \quad (5)$$

2. Rasio tingkat perputaran aktiva tetap menunjukkan sejauh mana efektivitas perusahaan menggunakan aktiva tetapnya. Semakin tinggi rasio berarti semakin efektif penggunaan aktiva tetapnya.

$$\text{Rasio Tingkat Perputaran Aktiva Tetap} = \frac{\text{Penjualan}}{\text{Aktiva Tetap Bersih}} \quad (6)$$

3. Rasio tingkat perputaran total aktiva manunjukkan efektivitas penggunaan total aktiva.

$$\text{Rasio Tingkat Perputaran Total Aktiva} = \frac{\text{Penjualan}}{\text{Total Aktiva}} \quad (7)$$

2.1.3.2.3 Rasio *Leverage* atau Solvabilitas

Rasio *leverage* atau solvabilitas digunakan untuk mengukur kemampuan perusahaan dalam memenuhi kewajiban-kewajiban jangka panjangnya. Yang termasuk dalam rasio *leverage* atau solvabilitas diantaranya (Riyanto, 2001:334):

1. Rasio utang (*debt ratio*) mengukur sejauh mana kemampuan perusahaan memenuhi kewajiban jangka panjangnya.

$$\text{Rasio Utang} = \text{Total Utang / Total Aktiva} \quad (8)$$

2. Rasio kewajiban terhadap modal (*debt to equity ratio*) menunjukkan kemampuan perusahaan untuk memenuhi semua total kewajibannya dengan menggunakan modal sendiri.

$$\text{Rasio Kewajiban Terhadap Modal} = \frac{\text{Total Utang}}{\text{Total Modal Sendiri}} \quad (9)$$

3. *Time interest earned ratio* mengukur kemampuan perusahaan membayar bunga utang dengan laba sebelum bunga dan pajak atau dengan kata lain seberapa besar laba sebelum bunga dan pajak yang tersedia untuk menutup beban bunga.

$$\textit{Time Interest Earned Ratio} = \text{Laba Operasi / Beban Bunga (per thn)} \quad (10)$$

4. Rasio kewajiban lancar terhadap total aktiva mengukur berapa besar total aktiva perusahaan yang dibiayai dengan kewajiban lancar.

$$\text{Rasio Kewajiban Lancar Terhadap Total Aktiva} = \frac{\text{Utang Lancar}}{\text{Total Aktiva}} \quad (11)$$

5. Rasio kewajiban tidak lancar terhadap total aktiva mengukur berapa besar total aktiva perusahaan yang dibiayai dengan kewajiban bukan lancar.

$$\text{Rasio Kewajiban Tidak Lancar Terhadap Total Aktiva} = \frac{\text{Utang Tidak Lancar}}{\text{Total Aktiva}} \quad (12)$$

2.1.3.2.4 Rasio Rentabilitias atau Profitabilitas

Rasio rentabilitas atau profitabilitas menunjukkan kemampuan perusahaan untuk menghasilkan laba selama periode tertentu. Yang termasuk dalam rasio rentabilitas atau profitabilitas diantaranya (Riyanto, 2001:335):

1. Marjin laba kotor mencerminkan *mark-up* terhadap harga pokok penjualan selain mencerminkan kemampuan manajemen untuk meminimalisasi harga pokok penjualan dalam hubungannya dengan penjualan yang dilakukan perusahaan.

   Marjin Laba Kotor = Laba Kotor / Penjualan Bersih (13)

2. Marjin laba usaha mencerminkan kemampuan manajemen untuk menghasilkan laba setelah beban operasi atau usaha dan harga pokok penjualan dalam hubungannya dengan penjualan yang dilakukan.

   Marjin Laba Usaha = Laba Usaha / Penjualan Bersih (14)

3. Marjin laba bersih mencerminkan kemampuan manajemen untuk menghasilkan laba setelah harga pokok penjualan, beban operasi atau usaha, beban lain-lainnya dan pajak dalam hubungannya dengan penjualan.

   Marjin Laba Bersih = Laba Bersih / Penjualan Bersih (15)

4. *Return on Investment* (ROI) mencerminkan kemampuan manajemen dalam mengatur aktiva-aktivanya seoptimal mungkin sehingga dicapai laba bersih yang diinginkan.

   Return on Investment = (EBIT) Laba Usaha / Total Aktiva (16)

2.2 Konsep Logika Fuzzy

Logika fuzzy merupakan salah satu komponen pembentuk *soft computing*. Logika fuzzy perama 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 penalran dengan logika fuzzy tersebut. (Kusumadewi & Purnomo, 2010:2).

Dalam banyak hal, logika fuzzy digunakan sebagai suatu cara untuk memetakan permasalahan dari masukan menuju ke keluaran yang diharapkan. Beberapa contoh yang dapat diambil antara lain menurut Kusumadewi & Purnomo, 2010:4):

1. Manajer pergudangan mengatakan pada manajer produksi seberapa banyak persediaan barang pada akhir minggu ini, kemudian manajer produksi akan menetapkan jumlah barang yang harus diproduksi esok hari.
2. Seorang pegawai melakukan tugasnya dengan kinerja yang sangat baik, kemudian atasan akan memberikan *reward* yang sesuai dengan kinerja pegawai tersebut.

Ada beberapa alasan mengapa orang menggunakan logika fuzzy (Kusumadewi & Purnomo, 2010:6) antara lain:

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

2.2.1 Himpunan Fuzzy

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item x dalam suatu himpunan A, yang sering ditulis dengan μA[x], memiliki 2 kemungkinan (Kusumadewi & Purnomo, 2010:10) yaitu:

1. Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
2. Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Terkadang kemiripan antara keanggotaan dengan probabilitas menimbulkan kerancuan. Keduanya memiliki nilai pada interval [0,1], namun interprestasi nilainya sangat berbeda antara kedua kasus tersebut. Keanggotaan fuzzy memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang. Misalnya, jika nilai keanggotaan bernilai suatu himpunan fuzzy USIA adalah 0,9; maka tidak perlu dipermasalahkan berapa seringnya nilai itu diulang secara individual untuk mengharapkan suatu hasil yang hampir pasti muda. Di lain pihak, nilai probabilitas 0,9 usia berarti 10% dari himpunan tersebut diharapkan tidak muda (Kusumadewi & Purnomo, 2010:11).

Menurut Sri Kusumadewi dan Hari Purnomo (2010:13) himpunan fuzzy memiliki 2 atribut, yaitu:

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti: MUDA, PAROBAYA, TUA
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variable seperti: 40, 25, 50, dsb.

Ada beberapa hal yang perlu diketahui dalam memahami sistem fuzzy (Kusumadewi & Purnomo,2010:14) yaitu:

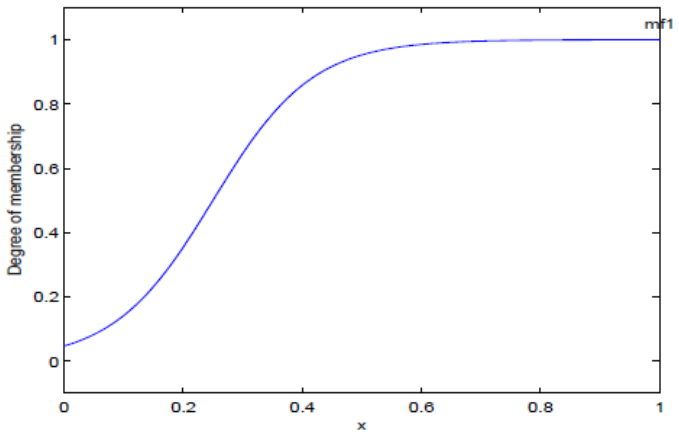
1. Variabel fuzzy

   Variabel fuzzy merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy. Contoh: umur, temperature, permintaan, dsb.
2. Himpunan Fuzzy

   Himpunan fuzzy merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel fuzzy.

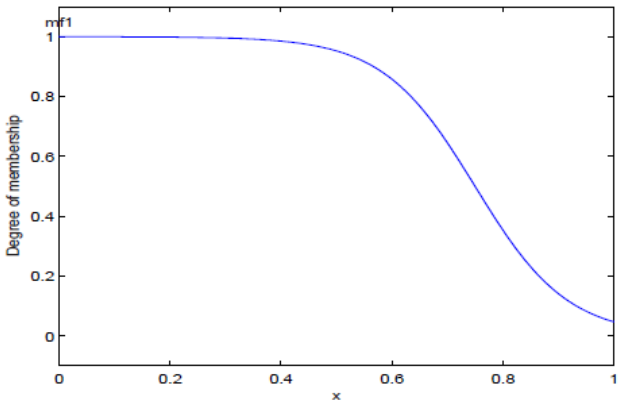
2.2.2 Fungsi Keanggotaan Sigmoid

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik input data kedalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Apabila U menyatakan himpunan universal dan A adalah himpunan fungsi fuzzy dalam U, maka A dapat dinyatakan sebagai pasangan terurut (Wang, 1997 dari Wulandari, 2005). Fungsi Sigmoid membuka ke kanan dengan parameter: sigmoid (x;12,0.25) ditunjukkan dalam Gambar 2.1:



Gambar 2.1 Fungsi Keanggotaan sigmoid membuka ke kanan (Jang *et al.*, 1997)

Sedangkan fungsi Sigmoid membuka ke kiri dengan parameter: sigmoid (x;-12,0.75) ditunjukkan dalam Gambar 2.2 berikut ini:



Gambar 2.2 Fungsi Keanggotaan sigmoid membuka ke kiri (Jang *et al.*, 1997)

2.2.3 Sistem Fuzzy

Seperti yang telah diketahui bahwa pendekatan logika fuzzy didasarkan pada bahasa alami, maka selain atribut numerik (*numeric variable*) logika fuzzy juga memiliki atribut bahasa (*linguistic variable*). Atribut bahasa ini merupakan konsep bahasa yang sering digunakan dalam kehidupan sehari-hari. Atribut bahasa ini biasanya terdiri dua bagian, antara lain (Kusumadewi & Purnomo, 2010):

1. Predikat fuzzy (*primary term*): panas, dingin, adem, dan lain-lain.
2. *Fuzzy modifier*: sangat, sedikit, hampir, sekali, dan lain-lain.

Modifier biasa digunakan untuk merubah makna dari predikat dan dikelompokkan lagi menjadi dua kelas seperti berikut (Kusumadewi & Purnomo, 2010):

1. *Fuzzy truth qualifier*: cukup benar, sangat benar, agak benar, agak salah, hampir salah, dan lain-lain.
2. *Fuzzy quantifier*: banyak, sedikit, hampir semua, semua, sering, jarang, dan lain-lain.

Dikarenakan oleh atribut himpunan fuzzy terdiri dari dua macam, maka proses pada sistem fuzzy dilakukan pada dua level yang berbeda, yaitu level linguistik dan level numerik. Proses pada kedua level ini dibagi menjadi tiga tahapan, antara lain (Kusumadewi & Purnomo, 2010):

1. Fuzzifikasi

   Pada tahap ini akan ditentukan derajat keanggotaan dari masukan yang bernilai numerik.
2. Inferensi fuzzy

   Pada tahap ini dilakukan evaluasi aturan fuzzy. Masukan pada tahap ini adalah derajat keanggotaan yang dihasilkan pada tahap fuzzifikasi, sedangkan keluaran berupa nilai kebenaran sebagai konsekuensi dari fungsi keanggotaan. Jika diperlukan dapat digunakan operator fuzzy untuk mendapatkan hasil evaluasi aturan fuzzy yang berupa nilai tunggal.
3. Defuzzifikasi

   Pada tahap ini dilakukan agregasi himpunan keluaran fuzzy tahap sebelumnya menjadi sebuah nilai numerik tunggal. Agregasi merupakan proses unifikasi dari semua keluaran pada tahap inferensi.

2.2.4 Metode Inferensi Mamdani

Inferensi fuzzy merupakan proses memetakan ruang masukan yang diberikan kedalam ruang keluaran menggunakan logika fuzzy. Dari proses pemetaan ini dapat dilihat dasar dari keputusan yang dibuat atau dapat dilihat pola yang ada. Metode inferensi fuzzy Mamdani merupakan metodologi fuzzy yang paling mudah dipahami. Metode Mamdani merupakan sistem pengaturan pertama yang menggunakan teori himpunan fuzzy. Metode ini pertama kali diusulkan di tahun 1975 oleh Ebrahim Mamdani. Pembuatan metode ini berdasarkan karya ilmiah dari Lotfi Zadeh (tahun 1973) tentang algoritma fuzzy untuk sistem yang kompleks dan digunakan dalam proses pengambilan keputusan (Kusumadewi & Purnomo, 2010).

Metode Mamdani adalah satu jenis inferensi fuzzy dimana himpunan fuzzy yang merupakan konsekuensi dari setiap aturan dikombinasikan menggunakan operator agregasi dan menghasilkan himpunan fuzzy yang kemudian didefuzzifikasikan untuk menghasilkan keluaran tertentu dari suatu sistem. Metode ini menggunakan operator minimum Rc sebagai implikasi fuzzy dan operator maksmin untuk komposisi. Metode Mamdani dapat digunakan untuk data masukan yang *singleton* (hanya terdiri dari satu masukan), maupun data masukan yang merupakan himpunan fuzzy (Kusumadewi & Purnomo, 2010).

2.3 Konsep Basis Data

Pada sub bab ini akan dibahas mengenai sistem basis data, *Structured Query Language* (SQL), dan MySQL

2.3.1 Sistem Basis Data

Basis data adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data dimaksudkan untuk permasalahan pada sistem yang memakai pendekatan berbasis berkas. Basis data adalah kumpulan *file* yang mempunyai kaitan antara satu file dengan file lain sehingga membentuk suatu bangunan data untuk menginformasikan suatu perusahaan atau instansi dalam batasan tertentu (Kadir, 2008).

1. Entitas

   Entititas adalah orang, tempat, kejadian atau konsep yang informasinya direkam.

2. *Field*

   Setiap entity mempunyai atribut atau sebutan untuk mewakili suatu entity.

3. *Record*

   Record adalah kumpulan isi elemen data (atribut) yang saling berhubungan menginformasikan tentang suatu entity secara lengkap.

4. *Data Value*

   Merupakan data aktual atau infomasi yang disimpan ditiap data elemen.

5. Kunci Elemen Data (*Key Element Data*)

   Tanda pengenal yang secara unik mengidentifikasikan entitas dari suatu kumpulan entitas.

Untuk mengelola basis data diperlukan perangkat lunak yang disebut DBMS. DBMS adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol, dan mengakses basis data dengan cara praktis dan efisien. Tujuan utama dalam pengolahan data dalam sebuah basis data adalah agar kita dapat memperoleh data yang kita cari dengan mudah dan cepat. Pemanfaatan basis data dilakukan dengan tujuan yaitu (Kadir, 2008) :

1. Kecepatan dan kemudahan (*speed*)
2. Efisien ruang penyimpanan (*space*)
3. Keakuratan (*accuracy*)
4. Keamanan (*security*)
5. Terpeliharanya keselarasan data (*consistant*)
6. Data dapat dipakai secara bersama (*shared*)
7. Dapat diterapkan standarisasi (*standardization*)

Menurut Abdul Kadir (2008) ada beberapa keunggulan dari DBMS diantaranya:

1. Mengendalikan / mengurangi duplikasi data
2. Menjaga konsistensi dan integritas data

3. Memudahkan pemerolehan informasi yang lebih banyak dari data yang sama disebabkan data dari berbagai bagian dalam organisasi dikumpulkan menjadi satu
4. Meningkatkan keamanan data dari orang yang tidak berwenang
5. Memaksakan penerapan standar
6. Dapat menghemat biaya karena data dapat dipakai oleh banyak departemen
7. Menanggulangi konflik kebutuhan antar pemakai karena basis data dibawah kontrol administrator basis data
8. Meningkatkan produktivitas pemrogram
9. Meningkatkan pemeliharaan melalui independesi data
10. Meningkatkan konkurensi (pemakai data oleh sejumlah data) tanpa menimbulkan masalah kehilangan informasi atau integritas
11. Meningkatkan layanan *backup* dan *recorvery*

Sementara itu kelemahan DBMS menurut Kadir (2008) diantaranya adalah sebagai berikut:

1. Kompleksitas yang tinggi membuat administrator dan pemakai akhir harus benar-benar memahami fungsi-fungsi dalam DBMS agar dapat diperoleh manfaat yang optimal. Kegagalan memahami DBMS dapat mengakibatkan keputusan rancangan yang salah, yang akan memberikan dampak serius bagi perusahaan
2. Ukuran penyimpanan yang dibutuhkan DBMS sangat besar dan membutuhkan memori yang besar agar bisa bekerja secara efisien
3. Rata-rata harga DBMS yang handal sangat mahal
4. Terkadang DBMS meminta kebutuhan perangkat keras dengan spesifikasi tertentu sehingga diperlukan biaya tambahan
5. Dampak kegagalan menjadi lebih tinggi karena semua pemakai sangat bergantung pada ketersediaan DBMS. Akibatnya, kalau terjadi kegagalan dalam komponen lingkungan DBMS akan membuat operasi dalam organisasi tersendat atau bahkan terhenti.

2.3.2 *Structured Query Language* (SQL)

SQL (*Structure Query Language*) adalah bahasa yang digunakan untuk mengakses basis data yang tergolong relasional. *Structure Query Language* (SQL) merupakan komponen bahasa *relational database system*. SQL merupakan bahasa baku (ANSI/SQL), *non procedural*, dan berorientasi himpunan (*set-oriented language*). SQL dapat digunakan baik secara interaktif atau ditempelkan (*embedded*) pada sebuah program aplikasi (Kadir, 2008). Berikut merupakan komponen-komponen SQL (Kadir, 2008):

1. *Data Definition Language* (DDL)

   Digunakan untuk mendefinisikan data dengan menggunakan perintah :

   a. *create*, berfungsi untuk membuat *tabel*, *view* dan *index*
   b. *drop*, berfungsi untuk menghapus *table*, *view* dan *index*.
   c. *alter*, berfungsi untuk merubah atribut pada suatu tabel

2. *Data Manipulation Language* (DML)

   Digunakan untuk memanipulasi data dengan menggunakan perintah :

   a. *select*, berfungsi untuk menampilkan baris (*record*)
   b. *insert*, berfungsi untuk menambah (*record*) baru
   c. *update*, berfungsi untuk merubah *record*
   d. *delete*, berfungsi untuk menghapus *record*.

3. *Data Manipulation Language* merupakan bagian terpadu bahasa SQL. Perintah-perintahnya dapat dibuat secara interaktif atau ditempelkan pada sebuah program aplikasi. Pemakai hanya perlu menentukan 'APA' yang diinginkan, DBMS menentukan 'BAGAIMANA' cara mendapatkannya.

4. *Data Control Language* (DCL)

   Digunakan untuk mengontrol hak para pemakai data dengan perintah :

   a. *grant*, digunakan untuk memberikan ijin akses kepada user
   b. *revoke*, digunakan untuk mencabut ijin akses kepada user

2.3.3 MySQL

MySQL (*My Structure Query Language*) adalah sebuah perangkat lunak system manajemen basis data SQL (*Structure Query Language*) dari sekian banyak DBMS (*Database Management System*), seperti Oracle, MS SQL, Postagre SQL

dan lain-lain. MySQL merupakan DBMS yang *multithread* atau *multi user* yang bersifat gratis di bawah lisensi GNU *General Public Licensce* (GPL) (Kadir, 2008).

Beberapa kelebihan MySQL menurut Abdul Kadir (2008), antara lain :

1. MySQL dapat berjalan stabil pada berbagai sistem oprasi, seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris dan lain-lain.
2. Bersifat *Open Source*
3. Bersifat multiuser, MySQL dapat di gunakan oleh beberpa *user* dalam waktu yang bersamaan tanpa mengalami masalah
4. MySQL memiliki kecepatan yang baik dalam menangani *query*. Dengan kata lain, dapat memproses lebih banyak SQL per satuan waktu.
5. Dari segi *security* atau keamanan data, MySQL memiliki babarapa lapisan sekuriti, seperti *level subnetmask*, nama *host* dan izin akses user dengan sistem perizinan yang mendetail serta password yang ter-*enkripsi*.

2.4 Alat Bantu Pemodelan Terstruktur

Pada sub bab ini akan dibahas mengenai diagram konteks, data flow diagram, kamus data, dan *entity – relationship diagram* (ERD)

2.4.1 Diagram Konteks

Diagram konteks adalah model atau gambar yang menggambarkan hubungan sistem dengan lingkungan sistem. Diagram Konteks ini menggambarkan hubungan input/output antara sistem dengan dunia luarnya (Whitten & Bentley, 2007).

Untuk menggambarkan diagram konteks, kita deskripsikan data apa saja yang dibutuhkan oleh sistem dan dari mana sumbernya, serta informasi apa saja yang akan dihasilkan oleh sistem tersebut dan kemana informasi tersebut akan diberikan. Suatu diagram konteks selalu mengandung satu dan hanya satu proses saja. Proses ini mewakili proses dari seluruh sistem. Dapat dikatakan, bahwa sebuah diagram konteks adalah kasus khusus dari DFD atau bagian dari DFD yang berfungsi memetakan modul lingkungan yang dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem (Whitten & Bentley, 2007).

2.4.2 Data Flow Diagram

DFD atau Diagram Arus Data adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan (Whitten & Bentley, 2007).

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Kelebihan utama pendekatan aliran data, yaitu (Whitten & Bentley, 2007) :

1. Kebebasan dari menjalankan implementasi teknis sistem.
2. Pemahaman lebih jauh mengenai keterkaitan satu sama lain dalam sistem dan subsistem.
3. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram aliran data.
4. Menganalisis sistem yang diajukan untuk menentukan apakah data-data dan proses yang diperlukan sudah ditetapkan.

Disamping itu terdapat kelebihan tambahan, yaitu (Whitten & Bentley, 2007) :

1. Dapat digunakan sebagai latihan yang bermanfaat bagi penganalisis, sehingga bisa memahami dengan lebih baik keterkaitan satu sama lain dalam sistem dan subsistem.
2. Membedakan sistem dari lingkungannya dengan menempatkan batas-batasnya.
3. Dapat digunakan sebagai suatu perangkat untuk berinteraksi dengan pengguna.

Memungkinkan penganalisis menggambarkan setiap komponen yang digunakan dalam diagram.

2.4.3 Kamus Data

Kamus Data (KD) tidak menggunakan notasi grafis sebagaimana halnya DFD, tetapi porsinya dalam memodelkan sistem tidak perlu diragukan lagi (sebuah model tidak lengkap tanpa KD). KD juga mempunyai fungsi yang sama dalam pemodelan sistem. Selain itu KD berfungsi membantu pelaku sistem untuk

mengerti aplikasi secara detil, kamus data mereorganisasi semua elemen data yang digunakan dalam sistem dengan presisi yang sedemikan rupa sehingga pemakai dan penganalisis sistem memiliki dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses (Whitten & Bentley, 2007).

Kamus Data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data selain digunakan untuk dokumentasi dan mengurangi redundansi, juga dapat digunakan untuk (Whitten & Bentley, 2007):

1. Memvalidasi diagram aliran data dalam hal kelengkapan dan keakuratan
2. Menyediakan suatu titik awal untuk mengembangkan layar dan laporan-laporan
3. Menentukan muatan data yang disimpan dalam file-file
4. Mengembangkan logika untuk proses-proses diagram aliran data

KD dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Pada tahap analisis sistem, KD dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. Pada tahap perancangan sistem, KD digunakan untuk merancang input, merancang laporan-laporan dan database. Kamus data dibuat berdasarkan arus data yang ada di DFD (Whitten & Bentley, 2007).

Pada kebanyakan sistem dalam dunia nyata, kadang-kadang elemen data terlalu kompleks utuk didefinisikan. Kekompleksan tersebut seharusnya diuraikan melalalui sejumah elemen data yang lebih sederhana dan didefinisikan. Pendefinisian tersebut menggunakan notasi yang umumnya digunakan dalam menganalisis sistem dengan menggunakan sejumlah simbol yaitu:

Tabel 2.1 Simbol – Simbol Kamus Data

No	Simbol	Keterangan
1.	=	Terdiri dari, mendefinisikan, diuraikan menjadi
2.	+	Dan
3.	( )	Menunjukkan suatu elemen yang bersifat pilihan (opsional). Elemen-elemen yang bersifat pilihan ini bisa dikosongkan pada layar masukan atau bisa juga dengan memuat spasi atau nol untuk field-field numeric pada struktur file.
4.	{ }	Menunjukkan elemen-elemen repetitive, juga disebut kelompok berulang atau tabel-tabel. Kemungkinan bisa ada satu atau beberapa elemen berluang di dalam kelompok tersebut. Kelompok berulang bisa mengandung keadaan-keadaan tertentu, seperti misalnya, jumlah pengulangan yang pasti atau batas tertinggi dan batas terendah uuntuk jumlah pengulangan.
5.	[]	Menunjukkan salah satu dari dua situasi tertentu. Satu elemen bisa ada sedangkan elemen lainnya juga ada, tetapi tidak bisa kedua-duanya ada secara bersamaan. Elemen-elemen yang ada di dalam tanda kurung ini saling terpisah satu sama lain. (dengan kata lain, memilih salah satu dari sejumlah alternatif, seleksi)
6.	|	Pemisah sejumlah alternatif pilihan antara simbol [ ]
7.	@	Identifikasi atribut kunci
8.	**	Komentar

Dalam kamus data juga dapat dicantumkan kode-kode karakter yang dapat membantu analis dalam menggambarkan bentuk dari data ataupun dibuat untuk merancang suatu kerluaran.

Tabel 2.2 Kode Karakter Kamus Data

Karakter Pemformatan	Keterangan
X	Bisa memasukkan atau menampilkan/mencetak suatu karakter
9	Hanya memsukkan atau menampilkan angka-angka
Z	Menampilkan nol-nol yang memimpin sebagai spasi
‘	Menyisipkan koma ke dalam suatu tampilan numerik
.	Menyisipkan suatu periode ke dalam suatu tampilan numerik
/	Menyisipkan slash (/) ke dalam suatu tampilan numerik
-	Menyisipkan suatu tanda penghubung ke dalam suatu tampilan numerik
V	Menunjukkan suatu posisi desimal (bila titik desimal tidak dimasukkan)

2.4.4 *Entity-Relationship Diagram* (ERD)

Entity Relationship Diagram atau ERD adalah suatu model jaringan yang menggambarkan *layout* (susunan) penyimpanan data dari sebuah sistem ER-

Diagram yang menggambarkan data-data dalam keadaan diam (data yang disimpan). Komponen - komponen ERD (Kadir, 2008) :

1. *Entity* adalah segala sesuatu yang dapat dijelaskan dengan data kelompok benda atau obyek diberi nama dengan kata benda.
2. *Relationship* merupaka suatu assosiasi antar satu atau beberapa *entity*, diberi nama dengan kata benda.
3. *Atribute* merupakan *property* atau karekteristik suatu *entity relationship*.

Jenis-jenis *relationship* dengan kardinalitas relasi dan atribut-atributnya (Kadir, 2008) :

1. *One to one* (satu ke satu)

   One to one adalah hubungan yang menggambarkan bahwa *key* suatu entitas hanya memiliki satu atribut yang berhungan dengan satu atribut yang lain pada entity tersebut
2. *One to Many* (satu ke banyak)

   One to Many adalah hubungan yang menggambarkan bahwa *key* suatu entitas memiliki banyak kegiatan entitas lain
3. *Many to Many* (banyak ke banyak)

   Many to Many adalah hubungan yang menggambarkan bahwa *key* suatu entitas memiliki banyak kegiatan entitas lain.

2.5 Konsep Dasar Web

2.5.1 HTML

HTML (*Hyper Text Markup Language*) adalah suatu sistem penulisan perintah dan formating hyper text sederhana yang di tulis ke dalam aplikasi pengolah kata dan disimpan ke dalam format ASCII (*American Standard Code for Information Interchange*) normal sehingga menjadi home page dengan tambahan perintah-perintah HTML. Dengan menggunakan perintah-perintah HTML memungkinkan user untuk melakukan fungsi-fungsi berikut ini (Wahana, 2001):

1. Menentukan ukuran dan alur text
2. Mengintegrasikan gambar dengan text (*in-line*)
3. Membuat links
4. Mengintegrasikan file audio dan video
5. Membuat form interaktif

Struktur penulisan pemrograman HTML yang baik pada dokumen HTML adalah penulisan tag-tag yang ditempatkan pada dokumen yang terbagi ke dalam area dan fungsional (Wahana, 2001). Berikut ini adalah format dasar penulisan program yang baik pada dokumen HTML yang dapat dilihat pada gambar 2.3 :

```
<HEAD>
       Untuk informasi header
</HEAD>
<BODY>
       Untuk informasi body
</BODY>
```

Gambar 2.3 Format Dasar Penulisan Kode Pada HTML

Informasi yang di tempatkan di antara tag <HEAD> di gunakan untuk memberikan keterangan kepada browser dan tidak secara langsung mempengaruhi user. Sedangkan informasi yang di tempatkan di antara tag <BODY> adalah informasi dokumen aktual dan semua tag-tag untuk format tampilan informasi tersebut (Wahana, 2001). Sebagai contoh penulisan program HTML sederhana dapat dilihat pada gambar 2.4:

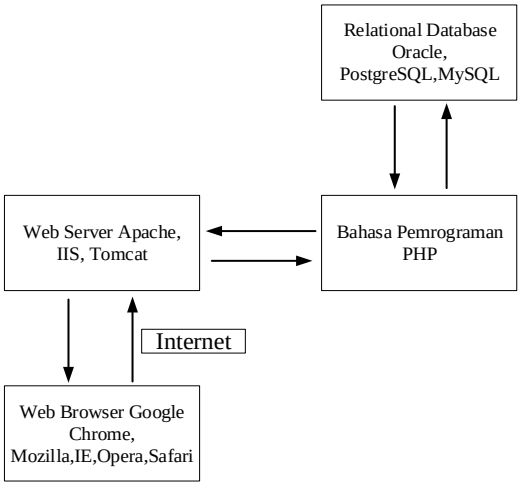
```
<HEAD>
       <title> Halaman web pertama </title>
</HEAD>
<BODY>
       <H1> Ini adalah halaman web pertama saya</H1>
       <HR>
       Hallo dunia!
</BODY>
```

Gambar 2.4 Contoh Penulisan Kode Pada HTML

2.5.2 PHP

PHP (*Pemrograman Hypertext Preprocessor*) yaitu bahasa pemrograman web *server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan beberapa pada server. PHP adalah *script* yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang di terima *client* selalu yang terbaru atau up to date. Semua *script* PHP dieksekusi pada *server* di mana *script* tersebut di jalankan (Wahana, 2001).

Berikut merupakan bagan dari aplikasi web yang dapat dilihat pada Gambar 2.5:



Gambar 2.5 Bagan dari Aplikasi Web

Dari gambar diatas terlihat adanya hubungan antara *web browser*, *web server*, bahasa pemrograman PHP, dan Database. Secara teknis, Database menyimpan data yang diolah oleh bahasa pemrograman PHP yang ada di *web server* kemudian data tersebut ditampilkan dengan *web browser* dengan cara menuliskan alamat atau URL sesuai nama file yang terdapat pada *web server* (Anhar, 2010).

PHP merupakan sebuah bahasa pemrograman web yang memiliki sintak atau aturan dalam menuliskan script atau kode-kodenya. Untuk menjelaskan cara penulisan kode PHP, bisa kita lihat pada empat macam cara penulisan kode PHP, yaitu (Anhar, 2010):

```
1. <? echo ("ini adalah script PHP\n"); ?>

2. <?php echo ("ini juga script PHP\n"); ?>

3. <script language="php">
      Echo ("latihan menulis script PHP");
   </script>

4. <% echo ("kalau ini mirip dengan ASP"); %>
```

Gambar 2.6 Format Dasar Penulisan Kode Pada PHP

Kita dapat memilih salah satu dari keempat contoh penulisan script PHP di atas, tetapi yang paling popular dan sering di gunakan adalah pada contoh nomor 1 dan 2. Perhatikan bahwa kode-kode PHP memiliki tata aturan, yaitu diawali dengan tanda <?php dan diakhiri dengan tanda ?>. Tiap akhir baris harus selalu di beri tanda

titik koma (;). PHP bersifat *CASE SENSITIVE*, artinya penulisan huruf besar dan kecil pada kode PHP sangat berpengaruh (Anhar, 2010).

2.5.3 Javascript

JavaScript merupakan bahasa yang berbentuk kumpulan skrip yang berfungsi untuk memberikan tampilan yang tampak lebih interaktif pada dokumen web. Dengan kata lain, bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan ke dalam bahasa pemrograman HTML dengan mengijinkan pengeksekusian perintah-perintah pada sisi *client*, dan bukan sisi server dokumen web. Pada hakikatnya, bahasa pemrograman *JavaScript* berisi skrip yang pemasangannya terselip di sebuah dokumen HTML. Sehingga bahasa *JavaScript* ini tidaklah memerlukan sebuah *compiler* atau penerjemah khusus untuk mengeksekusinya. Hal tersebut juga bergantung pada navigator yang terdapat di setiap browser (Wahana, 2009).

Untuk mulai menerapkan program *JavaScript* dalam web, ada beberapa hal yang harus diketahui oleh seorang perancang web, yaitu (Wahana, 2009):

1. Seorang perancang harus mengetahui cara menggunakan HTML dan mengedit dokumen HTML.
2. Seorang perancang harus menggunakan browser yang sudah mendukung pemrograman *JavaScript*, misalnya IE (*internet explorer*) versi 4 ke atas dan browser lain yang mendukung *JavaScript*.

Dengan *JavaScript*, sebuah halaman web akan menjadi lebih dinamis dan interaktif terhadap user karena halaman web mampu berfungsi sebagai sebuah program aplikasi yang dapat memproses masukan yang diberikan *user* dan memberikan hasil sesuai dengan yang telah diprogramkan. Halaman web yang merupakan jawaban dari suatu permintaan user tidak harus di *download* terlebih dulu dari *web server*, melainkan dihasilkan sendiri oleh *JavaScript* yang berbeda pada browser (Wahana, 2009).

Adapun kelebihan dari penggunaan *JavaScript* (Wahana, 2009) adalah sebagai berikut:

1. Lebih praktis dan mudah sebab bahasa pemrograman *JavaScript* memiliki sedikit *sintaks*.

2. Koneksi cepat sebab peletakan program terdapat disisi *client,* berukuran file sangat kecil, dan dapat langsung dijalankan di browser.

Sedangkan kelemahan dari bahasa pemrograman *JavaScript* antara lain:

1. Pengelolaan objek dalam *JavaScript* sangat terbatas.
2. Penggunaan *script* dapat di-*copy* langsung melalui sebuah web browser, sehingga setiap orang dapat menggunakan program *JavaScript* yang telah dibuat.

2.5.4 CSS

CSS (*Cascading Style Sheet*) merupakan bahasa pemrograman web yang di desain khusus untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web menjadi lebih rapih, terstruktur dan seragam. Pada tahun 1996, tepatnya pada tanggal 17 Desember 1996, W3C (*World Wide Web Consortium*) menjadikan CSS sebagai bahasa pemrograman standar untuk memformat dokumen web (Saputra, 2013). Tujuannya untuk mengurangi pembuatan tag-tag. Pada saat itu CSS masih versi 1, yang memiliki dukungan terhadap :

1. *Font* (Jenis, Ketebalan, dll)
2. Warna teks, latar belakang, dll
3. Teks atribut, misalnya spasi antar baris, kata, dan huruf

Akhirnya pada tahun 1998, W3C menghadirkan CSS versi 2, yang menjadi standar hingga saat ini. CSS dari tahun ke tahun terus di kembangkan dan hingga saat ini CSS telah mencapai versi 3 (CSS3) (Saputra, 2013).

2.6 Teori Pengujian

Menurut Nidhra dan Dondeti (2012), pengujian software adalah teknik yang sering digunakan untuk verifikasi dan validasi kualitas suatu perangkat lunak. Pengujian perangkat lunak adalah prosedur untuk eksekusi sebuah program atau sistem dengan tujuan untuk menemukan kesalahan.

Menurut Black (2009), *Structural test* (*white-box tests*) menemukan *bug* dalam elemen struktur tingkat rendah seperti yang terjadi di tingkatan kode, skema basis data, *chips, subassemblies* dan *interfaces*. Pengujian struktural ini didasarkan pada bagaimana suatu sistem beroperasi. Contohnya, pengujian struktural akan

mengungkapkan tempat penyimpanan basis data masih memiliki ruang penyimpanan *username* sebanyak 80 karakter, tetapi pada kenyataannya hanya memungkinkan menyimpan 40 karakter.

Pengujian struktural melibatkan pengetahuan teknis terperinci dari sistem. Untuk menguji perangkat lunak, *tester* membuat pengujian yang paling struktural dengan melihat kode dan struktur data itu sendiri. Untuk pengujian perangkat keras, *tester* membuat pengujian struktural untuk membandingkan spesifikasi *chip* untuk pembacaan *oscilloscopes* atau meter tegangan.

Menurut Black (2009), *tester* menggunakan *behavioral test* (*black-box tests*), sering digunakan untuk menemukan *bug* dalam *high level operations*, pada tingkatan fitur, profil operasional dan skenario pelanggan. Tester dapat membuat pengujian fungsional *black-box* berdasarkan pada apa yang harus sistem lakukan. *Behavioral testing* melibatkan pemahaman rinci mengenai domain aplikasi, masalah bisnis yang dipecahkan oleh sistem dan misi yang dilakukan sistem.

Behavioral test paling baik dilakukan oleh penguji yang memahami desain sistem, setidaknya pada tingkat yang tinggi sehingga mereka dapat secara efektif menemukan *bug* umum untuk jenis desain. Menurut Nidhra dan Dondeti (2012), *black box testing* juga disebut *functional testing*, sebuah teknik pengujian fungsional yang merancang *test case* berdasarkan informasi dari spesifikasi.