

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem**

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai tujuan. Sebagai gambaran, jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem. (Abdul Kadir; 2014; 61).

Sistem merupakan kumpulan unsur atau elemen – elemen yang saling berkaitan / berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu. (Asbon Hendra; 2012; 157 )

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Teori sistem secara umum pertama kali di uraikan oleh Kenneth Boulding, terutama menekankan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem. Kecenderungan manusia yang mendapat tugas untuk memimpin organisasi adalah dia terlalu memusatkan perhatiannya pada salah satu komponen sistem organisasi. Teori sistem mengadakan bahwa unsur pembentuk organisasi itu penting dan harus mendapat perhatian yang utuh supaya manajer dapat bertindak lebih efektif. Yang dimaksud unsur atau komponen pembentuk organisasi disini bukan hanya bagian-bagian yang tampak secara fisik, tetapi juga hal-hal yang mungkin bersifat

abstrak atau konseptual, seperti misi, pekerjaan, kegiatan, kelompok informal, dan lain sebagainya. (Tata Sutabri; 2012; 3)

Dari pengertian sistem di atas dapat disimpulkan bahwa sistem adalah himpunan komponen atau bagian yang saling berkaitan yang bersama – sama untuk mencapai suatu tujuan tertentu.

### **II.1.1. Elemen Sistem**

Ada beberapa elemen yang membentuk sebuah sistem yaitu:

#### **1. Tujuan**

Setiap sistem memiliki tujuan (*goal*) entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem tanpa tujuan, sistem menjadi tidak terarah dan tidak terkendali.

#### **2. Masukan**

Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud (tampak secara fisik) maupun yang tidak tampak.

#### **3. Proses**

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan limbah.

#### **4. Keluaran**

Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bisa berupa suatu informasi, saran, cetakan laporan, dan sebagainya.

## 5. Mekanisme pengendalian dan umpan balik

Mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feed back*), yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah mengatur agar sistem berjalan sesuai tujuan. (Abdul Kadir; 2014; 62-64).

### II.1.2. Klasifikasi Sistem

#### a. Sistem Abstrak (*Abstract System*)

Sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sebagai contoh, sistem teologia yang merupakan suatu sistem yang menggambarkan hubungan Tuhan dengan manusia.

#### b. Sistem Fisik (*Physical System*)

Merupakan sistem yang ada secara fisik sehingga setiap makhluk dapat melihatnya. Contohnya, sistem komputer, sistem akuntansi, sistem produksi, dan lain-lain.

#### c. Sistem Alamiah (*Natural System*)

Sistem yang terjadi melalui proses alam, dalam artian tidak dibuat oleh manusia, seperti sistem tata surya, sistem galaxy, sistem reproduksi, dan lain-lain.

#### d. Sistem buatan manusia (*Human Made System*)

Sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *human machine system*, contohnya sistem informasi.

- e. Sistem tertentu ( *deterministic System* )  
Beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat didekteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan. Contohnya sistem komputer.
- f. Sistem Tak Tentu ( *Probabilistic System* )  
Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas. Contohnya sistem manusia.
- g. Sistem tertutup ( *Closed System* )  
Sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teori, sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanya ada *relatively closed system*( secara relatif tertutup, tidak benar-benar tertutup).
- h. Sistem terbuka(*open system*)  
Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan sistem terotomatis, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern. (Asbon Hendra; 2012; 61-62 ).

### II.1.3. Sistem Terotomatis

Sistem terotomatis terbagi dalam sejumlah katagori, yaitu :

a. *Online system*

*Sistem online* adalah sistem yang menerima langsung *input* pada area dimana *input* tersebut direkam dan menghasilkan output yang dapat berupa hasil komputasi pada area dimana mereka dibutuhkan. Area sendiri dapat dipisah-pisah dalam skala, misalnya ratusan kilometer. Biasanya digunakan bagi resevasi angkutan udara, resevasi kreta api, perbankan, dan lain sebagainya.

b. *Real- time system*

*Sistem real – time* adalah mekanisme pengontrolan, perekaman data, dan pemrosesan yang sangat cepat sehingga output yang diterima dalam waktu yang relatif sama. Perbedaan dengan sistem *online* adalah satuan waktu yang digunakan real-time biasanya seperseratus atau seperseribu detik sedangkan *online* masih dalam skala detik atau kadang beberapa menit. Perbedaan lainnya, *online* hanya berinteraksi denga pemakai, sedangkan *real-time* berinteraksi langsung dengan pemakai dan lingkungan yang dipetakan.

c. *Decision support system and strategic planing system.*

Sistem ini memproses transaksi organisasi secara harian atau membantu para manajer mengambil keputusan, mengevaluasi, dan menganalisa tujuan organisasi. Bisanya digunakan untuk sistem penggajian, sistem pemesanan, sistem akuntansi, dan sistem produksi. Biasanya berbentuk

paket statistik, paket pemasaran, dan lain – lain. Sistem ini tidak hanya merekam dan menampilkan data, tetapi juga fungsi matematik, data analisa statistik, dan menampilkan informasi dalam bentuk grafik (tabel, chart) sebagai laporan konvensional.

*d. Knowledge-based system*

Program komputer ini dibuat mendekati kemampuan dan pengetahuan seorang pakar. Umumnya menggunakan perangkat keras dan perangkat lunak khusus seperti *LISP* dan *PROLOG*. (Asbon Hendra;2012;162-163).

#### **II.1.4. Metode Sistem**

*a. Blackbox Approach*

Suatu sistem dimana input dan output-nya dapat didefinisikan tetapi prosesnya tidak dapat diketahui atau tidak terdefinisi. Metode ini hanya dapat dimengerti oleh pihak dalam ( yang menangani ) sehingga pihak luar hanya mengetahui masukan dan hasilnya.

Contoh : bagian percetak uang, proses pencernaan.

*b. Analytic System*

Suatu metode yang mencoba untuk meliha hubungan seluruh masalah yang menyelidiki kesistematiskan tujuan dari sistem yang tidak efektif dan evaluasi pilihan dalam bentuk ketidak efektifan dan biaya. Dalam metode ini, beberapa langkah diberikan seperti dibawah ini :

1. Menentukan identitas dari sistem.
  - a. Sistem apa yang diterapkan.
  - b. Batasanya.

- c. Apa yang dilaksanakan sistem tersebut.
2. Menentukan tujuan dari sistem.
  - a. Output yang dihasilkan dari isi sistem.
  - b. Fungsi dan tujuan yang diminta untuk mencoba menanggulangi lingkungan.
3. Bagian apa saja yang terdapat dalam sistem dan apa tujuan dari masing – masing bagian tersebut.
  - a. Tujuan masing – masing bagian sistem harus jelas.
  - b. Cara apa yang digunakan subsistem untuk berhubungan dengan subsitem lainnya.
4. Bagaimana bagian – bagian yang ada dalam sistem itu saling berhubungan menjadi satu kesatuan. (Asbon Hendra; 2012;166 -167 ).

## **II.2. Informasi**

Informasi merupakan salah satu sumber daya penting dalam manajemen modern. Banyak keputusan strategis yang bergantung kepada informasi. Sebagaimana diketahui, sumber daya 4M+1I yang mencakup manusia (Sumber Daya Manusia atau SDM), material (termasuk didalamnya energi), mesin, modal, dan informasi merupakan sumber daya vital bagi kelangsungan organisasi bisnis. (Abdul Kadir; 2014; 41).

Informasi merupakan data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi, ada suatu proses transformasi data menjadi suatu informasi = *input – proses – output*. (Asbon Hendra; 2012; 167 ).

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Bila tidak ada pilihan atau keputusan maka informasi tidak diperlukan. (Tata Sutabri; 2012; 22)

### **II.3. Sistem Informasi**

Pengertian Sistem Informasi menurut Alter “Sistem Informasi adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.

Pengertian Sistem Informasi menurut Gelinas, Oram, dan Wiggins Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada para pemakai. (Abdul Kadir; 2014; 9).

“suatu sistem terintegritasi yang mampu menyediakan informasi yang bermanfaat bag penggunanya”. “sebuah sistem terintegritas atau sistem manusia-mesin, untuk menyediakan informasi untuk mendukung operasi, manajemen dalam suatu organisasi”. “sekumpulan prosedur manua atau terkomputerisasi yang mengumpulkan / mengambil, mengelola, menyimpan, dan menyebarkan informasi dalam mendukung pengambilan dan kendali keputusan”. “sekelompok orang, prosedu,input,output, dan pengelolahanya secara bersama-sama menghasilkan informasi yang akurat, tepat waktu, dan relevan bagi penggunanya. (Asbon Hendra; 2012; 168-169).

Sistem informasi adalah suatu sistem di dalamnya suatu organisasi yang mempertemukan kebutuhan pengelolaan harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan – laporan yang diperlukan oleh pihak luar tertentu. (Tata Sutabri; 2012; 38 )

#### **II.4. Data**

Data dapat dianalogikan dengan sejumlah blok yang biasa digunakan anak-anak untuk membentuk berbagai struktur sesuai dengan imajinasi mereka. Secara konsep, data adalah deskripsi tentang benda, kejadian, aktifitas, dan transaksi, yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai. (Abdul Kadir; 2014; 43).

Data merupakan *raw material* untuk suatu informasi. Perbedaan informasi dan data sangat relatif, tergantung pada nilai gunanya bagi management yang memerlukan. Suatu informasi bagi level manajemen tertentu bisa menjadi data bagi manajemen level di atasnya, atau sebaliknya. (Asbon Hendra; 2012; 167).

#### **II.5. Sistem Pendukung Keputusan (SPK)**

Sistem pendukung keputusan (SPK) atau dikenal dengan *Decision Support System (DSS)* pada tahun 1970-an sebagai pengganti istilah *management information systems (MIS)*. Tetapi pada dasarnya SPK merupakan pengembangan lebih lanjut dari MIS yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakainya, maksud dan tujuan dari adanya sistem pendukung keputusan yaitu untuk mendukung pengambil keputusan memilih *alternative* keputusan yang

merupakan hasil pengolahan informasi-informasi yang diperoleh dengan menggunakan model-model pengambilan keputusan serta menyelesaikan masalah-masalah bersifat terstruktur, semi terstruktur dan tidak terstruktur. (Jurnal Informasi dan teknologi ilmiah (inti), Vol:II, Nomor :1, Februari 2014;12)

Sistem penunjang atau pendukung keputusan (SPK) didefinisikan sebagai suatu sistem informasi untuk membantu manajer proses pengambilan setengah terstruktur (*semi structured*) supaya lebih efektif dengan menggunakan model-model analisi dan data yang tersedia.

Dari defenisi diatas, maka dapat diketahui tujuan SPK adalah sebagai berikut :

1. Membantu manager mengambil keputusan setengah terstruktur yang dihadapi oleh manager level menengah.
2. Membantu atau mendukung manajemen mengambil keputusan bukan menggantikannya.
3. Meningkatkan efektifitas pengambilan keputusan manajemen bukan untuk meningkatkan efesiensi. Walaupun waktu manager penting (efesiensi), tetapi efektifitas merupakan tujuan utama pengguna SPK. (Jurnal Pelita Informatika Budi Darma, Volume : VI , Nomor: 3, April 2014;49-50)

## **II.6. Metode Sugeno**

Penalaran dengan metode *output* (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. Michio Sugeno mengusulkan penggunaan singleton sebagai fungsi keanggotaan dari konsekuen.

Singleton adalah sebuah himpunan *fuzzy* dengan fungsi keanggotaan pada titik tertentu mempunyai sebuah nilai dan 0 di luar titik tersebut. Pada metode Sugeno dua bagian pertama dari proses penarikan kesimpulan *fuzzy*, fuzzifikasi *input* dan menerapkan operator *fuzzy* semua sama dengan metode Mamdani. Perbedaan utama antara metode Mamdani dan Sugeno adalah *output* membership function dari metode Sugeno berbentuk linier atau konstan.

Aturan pada model *fuzzy* Sugeno mempunyai bentuk :

**If *Input 1* =  $x$  and *Input 2* =  $y$  then *Output* is  $z = ax + by + c$**

Untuk model Sugeno orde-Nol, *Output* level  $z$  adalah konstan ( $a=b=0$ ). *Output* level  $z_i$  dari setiap aturan merupakan berat dari aturan  $w_i$  (firing strength). Sebagai contoh, untuk aturan AND dengan *Input 1* =  $x$  dan *Input 2* =  $y$ , maka firing strength adalah :  $w_i = \text{AndMethod}(F_1(X), F_2(Y))$  dimana  $F_{1,2}(\cdot)$  adalah membership function untuk *Input 1* dan 2.

#### **Keuntungan metode Sugeno :**

1. Komputasinya lebih efisien .
2. Bekerja paling baik untuk teknik linear (kontrol PID, dll) .
3. Bekerja paling baik untuk teknik optimasi dan adaptif .
4. Menjamin kontinuitas permukaan *output* .
5. Lebih cocok untuk analisis secara matematis.

**Ada 2 model *fuzzy* dengan metode Sugeno yaitu sebagai berikut:**

1. Model *Fuzzy* Sugeno Orde-Nol

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

**IF (x1 is A1) • (x2 is A2) • (x3 is A3) • ..... • (xN is AN) THEN z=k** dengan  $A_i$  adalah himpunan *fuzzy* ke-i sebagai anteseden, dan k adalah suatu konstanta (tegas) sebagai konsekuen.

## 2. Model *Fuzzy Sugeno* Orde-Satu

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

**IF (x1 is A1) •..... • (xN is AN) THEN  $z = p_1 \cdot x_1 + \dots + p_N \cdot x_N + q$**  Dengan  $A_i$  adalah himpunan *fuzzy* ke-i sebagai anteseden, dan  $p_i$  adalah suatu konstanta (tegas) ke-i dan q juga merupakan konstanta dalam konsekuen. Apabila komposisi aturan menggunakan metode *SUGENO*, maka defuzzifikasi dilakukan dengan cara mencari nilai rata-ratanya . Sumber : (Jurnal Ilmiah Teknologi dan Informasi ASIA Vol. 7 No.1, Februari 2013; 6-7)

Sistem *fuzzy Sugeno* memperbaiki kelemahan yang dimiliki oleh sistem *fuzzy* murni untuk menambah suatu perhitungan matematika sederhana sebagai bagian *THEN*. Pada perubahan ini, sistem *fuzzy* memiliki suatu nilai rata-rata tertimbang (*Weighted Average Values*) di dalam bagian aturan *fuzzy IF-THEN*.

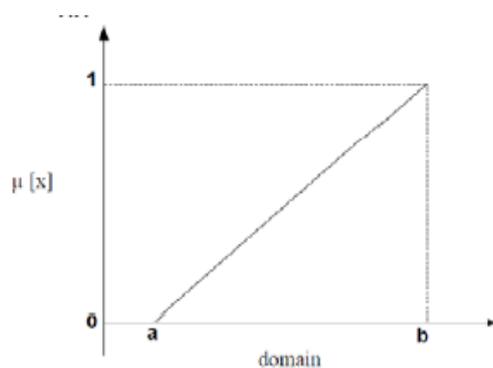
Sistem *fuzzy Sugeno* juga memiliki kelemahan terutama pada bagian *THEN*, yaitu dengan adanya perhitungan matematika sehingga tidak dapat menyediakan kerangka alami untuk merepresentasikan pengetahuan manusia dengan sebenarnya. Permasalahan kedua adalah tidak adanya kebebasan untuk menggunakan prinsip yang berbeda dalam logika *fuzzy*, sehingga ketidakpastian dari sistem *fuzzy* tidak dapat direpresentasikan secara baik dalam kerangka ini

## Pembentukan Fungsi Keanggotaan

Fungsi Keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam 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. Ada beberapa fungsi yang bisa digunakan antara lain :

### 1. Representasi Linear

Pada representasi linear, pemetaan *input* ke derajat keanggotaan yang digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Ada 2 keadaan himpunan *fuzzy* yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi Gambar II.1 dan Gambar II.2.

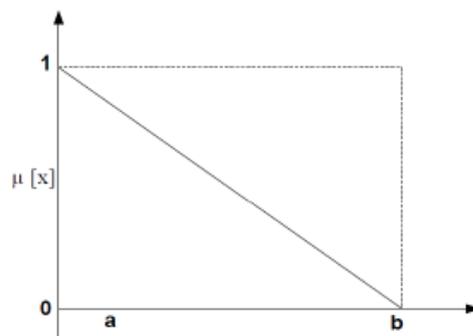


Gambar II.1. Kurva Linier Naik

Sumber : (Jurnal Ilmiah Teknologi dan Informasi ASIA Vol. 7 No. 1,  
Februari 2013; 7)

Fungsi Keanggotaan:

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



**Gambar II.2. Kurvva Turun**

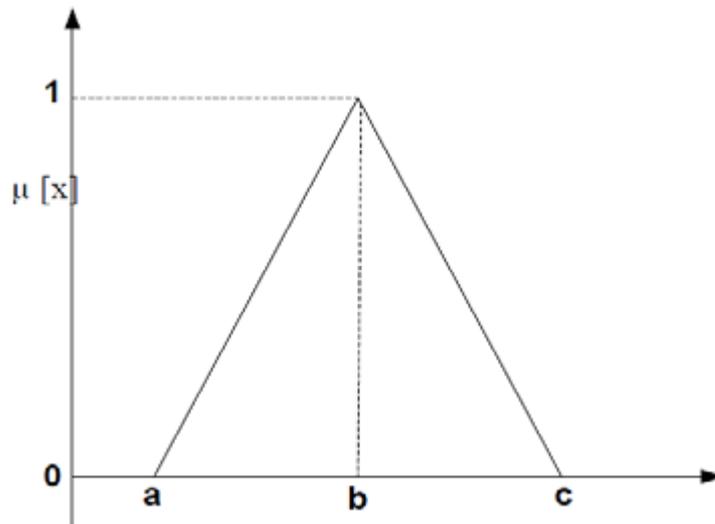
**Sumber : (Jurnal Ilmiah Teknologi dan Informasi ASIA Vol. 7 No. 1,  
Februari 2013; 8)**

Fungsi Keanggotaan:

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

## 2. Representasi Kurva Segitiga

Kurva Segitiga pada dasarnya merupakan gabungan antara 2 garis (linear) yang ditunjukkan dalam Gambar II.3.



**Gambar II.3. Kurva Segitiga**

**Sumber : (Jurnal Ilmiah Teknologi dan Informasi ASIA Vol. 7 No. 1,  
Februari 2013; 8)**

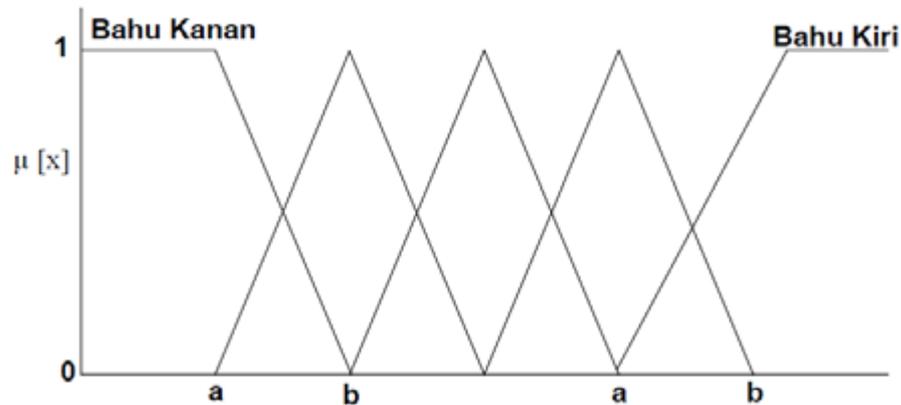
Fungsi Keanggotaan:

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

**Sumber : (Jurnal Ilmiah Teknologi dan Informasi ASIA Vol. 7 No.1,  
Februari 2013; 8)**

### **3. Representasi Kurva Bahu**

Representasi kurva bahu digunakan untuk mengakhiri variabel suatu daerah variable *fuzzy*. Untuk bahu kiri bergerak dari pernyataan benar benar ke pernyataan salah, demikian juga bahu kanan bergerak dari pernyataan salah ke pernyataan benar. Untuk lebih jelasnya dapat dilihat pada Gambar II.4.



**Gambar II.4. Kurva Bahu**

**Sumber : (Jurnal Ilmiah Teknologi dan Informasi ASIA Vol. 7 No. 1,  
Februari 2013; 8)**

Fungsi Keanggotaan bahu kiri:

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

Fungsi Keanggotaan bahu kanan:

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

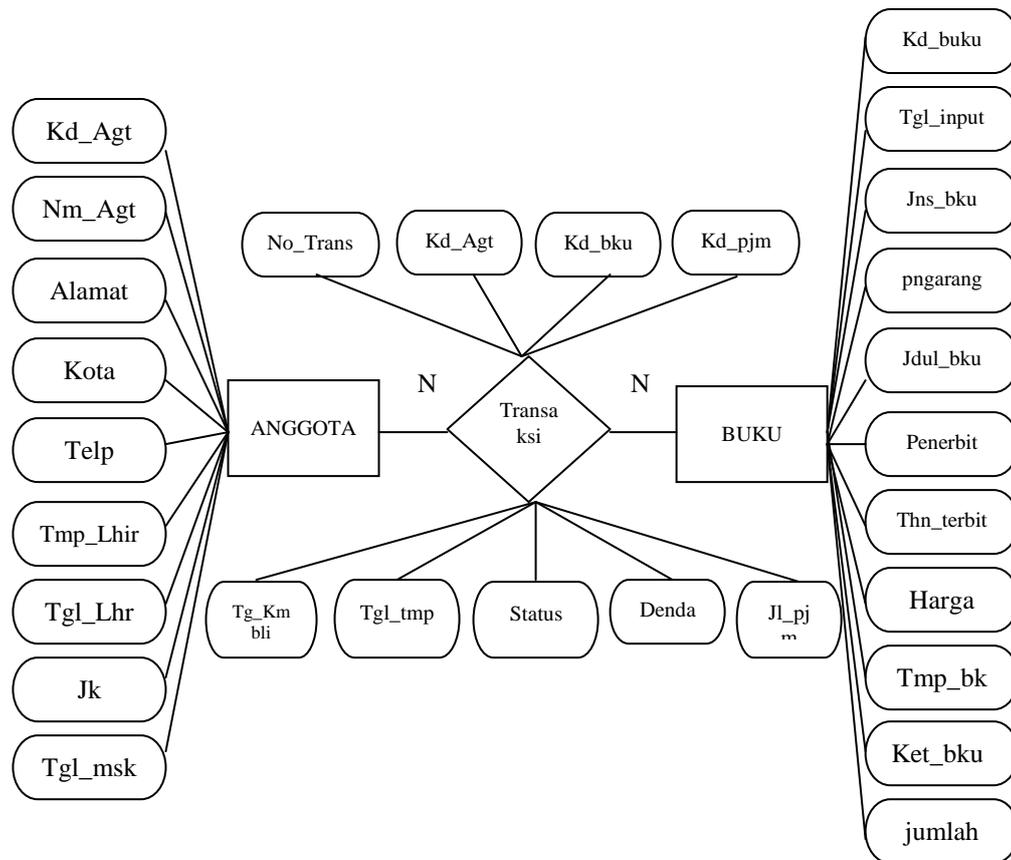
**(Jurnal Ilmiah Teknologi dan Informasi ASIA Vol. 7 No. 1, Februari 2013;**

**6-8)**

## **II.7. Entity Realitionship Diagram (ERD)**

ERD (*Entity Realitionship Diagram*) adalah sebuah diagram yang secara konseptua memetakan hubungan antar penyimpanan pada diagram DFD. ERD ini digunakan untuk melakukan pemodelan terhadap struktur data dan hubunganny. Penggunaan ERD ini dilakukn untuk mengurangi tingkt kerumitan penyusunan sebuah *database* yang baik.

*Entity* dapat berarti sebuah obyek yang daat dibedakan dengan obyek lainnya. Obyek tersebut dapat memiliki komponen-komponen data (atribut atau *field*) yang membuatnya dapat dibedakan dari obyek yang lain. Hal ini berarti sebuah entity memiliki himpunan yang diperlukan sebuah primary key untuk membedakan anggota-anggota dalam himpunan tersebut. Berikut ini contoh pengambaran entity.



**Gambar : II.5. Gambar ERD**

**Sumber : (Wahana Komputer; 2010; 31)**

ERD di atas adalah contoh ERD sistem informasi perpustakaan. Huruf N pada jalur-jalur hubungan antara anggota dan transaksi serta buku ada derajat dari simbol relasi. Ada beberapa derajat yang terjadi, yaitu :

- One to one* , menggambarkan bahwa antara 1 anggota entity A hanya dapat berhubungan dengan 1 anggota entity B. Biasanya derajat relasi ini digambarkan dengan simbol 1-1.
- One to many* , menggambarkan bahwa antara 1 anggota entity A hanya dapat memiliki hubungan dengan lebih dari 1 anggota *entity* B. Biasanya derajat relasi ini digambarkan dengan simbol 1-N.

- c. *Many to many* , menggambarkan bahwa antara satu anggota A dapat memiliki hubungan dengan lebih dari 1 anggota entity B. Biasanya derajat relasi ini digambarkan dengan simbol N-N. (Wahana Komputer; 2010; 30-31)

## II.8. Normalisasi

Normalisasi adalah proses dimana tabel – tabel pada *database* diletakkan dalam hal kesalingtergantungan di antara *field – field* pada sebuah tabel. Misalnya jika pada sebuah tabel terdapat ketergantungan terhadap lebih dari satu *field* dalam tabel tersebut, maka tabel tersebut harus dipecah menjadi banyak tabel. Setiap tabel hanya boleh memiliki sebuah field kunci yang menjadi ketergantungan dari *field* lainnya dalam tabel tersebut.

Ada beberapa langkah dalam normalisasi tabel yaitu :

### 1. Dekomposisi (*Decomposition*)

Dekomposisi adalah proses mengubah bentuk tabel supaya memenuhi syarat tertentu sebagai tabel yang baik. Dekomposisi dapat dikatakan berhasil jika tabel yang dikenai dekomposisi bila digabungkan kembali dapat menjadi tabel awal sebelum di dekomposisi. Dekomposisi akan sering dilakukan dalam proses normalisasi untuk memenuhi syarat – syaratnya.

### 2. Bentuk Tidak Normal

Pada bentuk ini semua data yang ada pada tiap *entity* (diambil atributnya) masih ditampung dalam satu tabel besar. Data yang ada pada tabel ini masih ada yang redundansi dan ada juga yang kosong. Semuanya masih tidak tertata rapi.

### 3. Normal *Form* Pertama (*1st Normal Form*)

Pada tahapan ini tabel di-dekomposisi dari tabel bentuk tidak normal yang kemudian dipisahkan menjadi tabel – tabel kecil yang memiliki kriteria tidak memiliki atribut yang bernilai ganda dan komposit. Semua atribut harus bersifat atomik.

### 4. Normal *Form* Kedua (*2nd Normal Form*)

Pada tahapan ini tabel dianggap memenuhi normal kedua jika pada tabel tersebut semua atribut yang bukan kunci primer bergantung penuh terhadap kunci primer tabel tersebut.

### 5. Normal *Form* Ketiga (*3rd Normal Form*)

.Setiap atribut pada tabel selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama. Bentuk normal ketiga biasanya digunakan bila masih ada tabel yang belum efisien. (Wahana Komputer, 2010;32-35).

## II.9. Basis Data (*Database*)

*Database* atau basis data adalah sekumpulan data yang memiliki hubungan sara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer. Data itu sendiri adalah data representasi dari semua fakta yang ada pada dunia nyata yang ada. *Database* sering digunakan untuk melakukan proses terhadap data-data tersebut untuk menghasilkan informasi tertentu. Misalnya dari data nama siswa dan tanggal lahir siswa anda bisa mendapatkan informasi nama siswa yang berulang tahun pada hari ini. Tentu saja informasi tersebut akan anda dapatkan dari *software* pemrosesan *database* dengan cara anda

memberikan perintah dalam bahasa tertentu yaitu *SQL* ( *Structured Query Language*).

Dalam *database* ada sebutan-sebutan untuk satuan data yaitu :

- a. Karakter, ini adalah satuan data terkecil. Data terdiri atas susunan karakter yang pada akhirnya mewakili data yang memiliki arti dari sebuah fakta.
- b. *Field* , adalah kumpulan dari karakter yang mewakili fakta tertentu misalnya seperti nama mahasiswa, tanggal lahir, dan lain-lain. Dalam dunia perancangan database, *field* juga disebut atribut. Bila dipandang dari sudut pemrograman berorientasi obyek aka sebuah *field* akan memiliki dua properti utama yaitu properti name dan *property type* Properti name atau nama adalah properti dari *field* yang berisi nama *field* yang mewakili data sejenis yang disimpan. Sedangkan *property type* adalah properti yang mengatur tipe data dari data yang akan ditampungnya. Misalnya nama *field*nya adalah nama siswa maka tipe datanya adalah char, bila nama *field*nya adalah tanggal lahir maka tipe datanya adalah *date*. *Field* dilihat seperti kolom.
- c. *Record*, adalah kumpulan dari field. Pada *record* anda dapat menemukan banyak sekali informasi penting dengan cara mengkombinasikan *field-field* yang ada.
- d. Tabel, adalah sekumpulan dari *record-record* yang memiliki kesamaan entity dalam dunia nyata. Kumpulan dari tabel adalah *database*, wujud fisik dalam *database* dalam sebuah komputer adalah sebuah file yang didalamnya terdapat berbagai tingkatan data yang telah disebutkan diatas.

- e. *File*, adalah bentuk fisik dari penyimpanan data file. *Database* berisi semua data yang telah disusun dan diorganisasikan sedemikian rupa sehingga memudahkan pemberian informasi.

Sistem *database* adalah sebuah kumpulan dari komponen-komponen *database-database* yang meliputi :

1. *Database*
2. *Database server*
3. Komponen *client software*
4. Aplikasi *database*

Aplikasi *database* adalah sebuah *software* khusus yang di desain dan digunakan oleh user atau pihak lainya seperti penyedia jasa pemrograman atau konsultan. Sedangkan *client server* adalah salah satu komponen yang termasuk dalam sistem *database* yang memungkinkan *software* aplikasi *database* mengakses data secara *remote* pada sebuah *server database*. (Wahana Komputer, 2010;24-25).

## **II.10. Unified Modeling Language (UML)**

*Unified Modelling Language* (UML) adalah sebuah alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek.

Sejarah UML sendiri terbagi dalam dua fase ; sebelum dan sesudahnya munculnya UML. Dalam fase sebelum, UML sebenarnya sudah mulai

diperkenalkan sejak tahun 1990-an namun notasi yang dikembangkan oleh para ahli analisi dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi.

Fase kedua; dilandasi dengan pemikiran untuk mempersatukan metode tersebut dan dimotori oleh *Object Management Group* (OMG) maka pengembangan UML dimulai pada akhir tahun 1994 ketika *Grady Booch* dengan metode OOD (*Object-Oriented Design*), *Jim Rumbaugh* dengan metode OMT (*Object Modelling Technique*) mereka ini bekerja pada *Rational Software Corporation* dan *Ivar Jacobson* dengan metode OOSE (*Object-Oriented Software Engineering*) yang bekerja pada perusahaan *Objectory Rational*.

Sebagai pencetus metode-metode tersebut mereka bertiga berinisiatif untuk menciptakan bahasa pemodelan terpadu sehingga pada tahun 1996 mereka berhasil merilis UML versi 0.9 dan 0.91 melalui *Request for Proposal* (RFP) yang dikeluarkan oleh *OMG* (*Braun, et.al. 2001*).

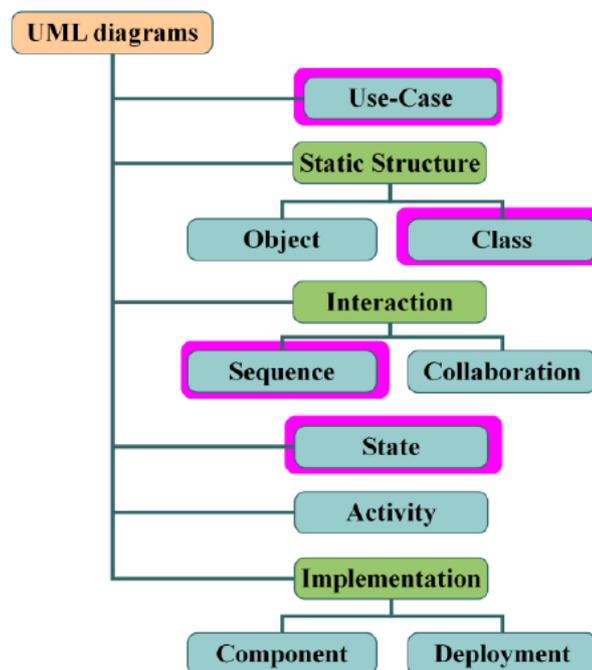
Kemudian pada Januari 1997 *IBM*, *ObjectTime*, *Platinum Technology*, *Ptech*, *Taskon*, *Reich Technologies* dan *Softeam* juga menanggapi *Request for Proposal* (RFP) yang dikeluarkan oleh *OMG* tersebut dan menyatakan kesediaan untuk bergabung.

Perusahaan-perusahaan ini menyumbangkan ide-ide mereka, dan bersama para mitra menghasilkan UML revisi 1.1. Fokus dari UML versi rilis 1.1 ini adalah untuk meningkatkan kejelasan UML Semantik versi rilis 1.0. Hingga saat ini UML versi terbaru adalah versi 2.0 (<http://www.uml.org/>).

Saat ini sebagian besar para perancang sistem informasi dalam menggambarkan informasi dengan memanfaatkan UML diagram dengan tujuan utama untuk membantu tim proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program.

Secara filosofi UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik. (Haviluddin , Jurnal Informatika Muawarman, VOL :6 , No.1 , Februari : 2011, 1-2).

Berikut gambar dari diagram UML



**Gambar II.6. Diagram UML**

**Sumber : (Jurnal Informatika Muawarman, VOL :6 , No.1 , Februari : 2011;2)**

Pemodelan adalah gambaran dari realita yang simpel dan dituangkan dalam bentuk pemetaan dengan aturan tertentu. Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). *UML* hanya bergungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek. (Rosa A.S, M. Shalahuddin; 2011: 116-118).

Diagram *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori yaitu :

#### *1. Structure Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Yang termasuk dalam structure diagram adalah sebagai berikut:

- a. Class diagram*
- b. Object diagram*
- c. Component diagram*
- d. Composite diagram*
- e. Package diagram*
- f. Deployment diagram*

## 2. *Behavior Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Yang termasuk dalam *behavior* diagram adalah sebagai berikut :

- a. *Use case*
- b. *Activity diagram*
- c. *State machine diagram*

## 3. *Interaction Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi antar sub sistem pada suatu sistem. Yang termasuk dalam *interaction* diagram adalah sebagai berikut :

- a. *Sequence diagram*
- b. *Communication diagram*
- c. *Timing diagram*
- d. *Interaction overview diagram* (Rosa A.S, M. Shalahuddin; 2011; 120-121)

## II.11. *Activity Diagram*

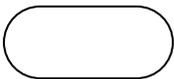
*Activity Diagram* (diagram aktivitas) merupakan diagram *flowchart* yang disempurnakan. Diagram aktivitas menggambarkan operasi pada suatu Obyek atau proses pada sebuah organisasi. Kelebihan diagram aktivitas dibandingkan dengan diagram *flowchart* adalah adanya dukungan konkurensi (pelaksanaan aktivitas secara bersamaan), pengiriman pesan dan *swimlane* (pelaku/penanggung jawab aktivitas).

*Diagram* aktivitas sangat berguna ketika ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai *use case* berinteraksi.

Diagram aktivitas diawali dengan lingkaran hitam, dan diakhiri dengan lingkaran hitam bertepi putih. Aktivitas digambarkan dengan kotak persegi panjang bersudut lengkung. Setiap aktivitas dihubungkan dengan panah dari awal hingga akhir diagram aktivitas.

Sama halnya dengan diagram *flowchart*, diagram aktivitas pun memiliki simbol yang sama untuk menggambarkan keputusan. Keputusan digambarkan dengan intan, namun di akhir kondisi yang menyertai keputusan diletakkan diluar simbol intan. Diagram aktivitas dapat menggambarkan konkurensi, yaitu satu atau lebih aktivitas yang berjalan secara bersamaan. Konkurensi diawali dengan sebuah garis tebal horizontal yang menjadi tempat keluarnya garis aktivitas. Konkurensi juga diakhiri dengan garis tebal horizontal.

**Tabel II.1. Simbol-Simbol Diagram Aktivitas**

<b>Simbol</b>	<b>Arti</b>	<b>Keterangan</b>
	Status Awal	Status awal aktivitas sistem sebuah diagram aktivitas memiliki sebuah status awal
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / <i>Decision</i>	Decision, atau pilihan untuk mengambil keputusan.
	Penggabungan / <i>Join</i>	Arah tanda panah alur proses.

	Status Akhir	Titik Akhir atau akhir dari aktivitas.
---	--------------	--

**Sumber: Rosa A.S, M. Shalahuddin; 2011: 134**

## II.12. Use case

Dalam bukunya Rekayasa Perangkat Lunak Rosa A.S – M. Shalahuddin mengatakan *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan apa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut actor dan *use case*.

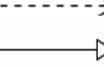
1. Aktor merupakan prang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, tapi *actor* belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. (Rosa A.S – M. Shalahuddin; Rekayasa Perangkat Lunak; 130-131)

Untuk menggambarannya dalam *use case* model biasanya digunakan *association relationship* yang memiliki *stereotype include*, *extend* atau

*generalization relationship*. Hubungan *include* menggambarkan bahwa suatu *use case* seluruhnya meliputi fungsionalitas dari *use case* lainnya.

Hubungan *extend* antar *use case* berarti bahwa satu *use case* merupakan tambahan fungsionalitas dari *use case* yang lain jika kondisi atau syarat tertentu terpenuhi, berikut adalah simbol *use case*.

**Tabel II.2. Simbol Use case**

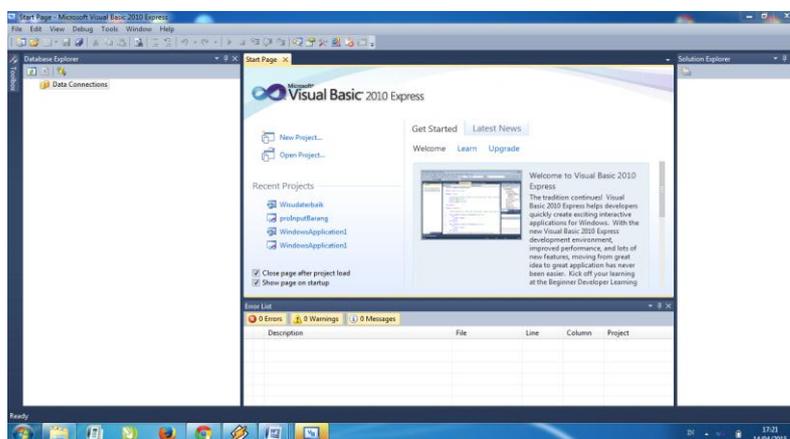
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i>
2		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
3		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum dan khusus) antar dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
4		<i>Include / uses</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankannya <i>use case</i> ini.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

**Sumber: Rosa A.S, M. Shalahuddin; 2011: 131-132**

### II.13. Bahasa Pemrograman *Microsoft Visual Basic 2010*

*Visual Basic 2010* merupakan salah satu bagian dari pemrograman terbaru yang dikeluarkan oleh *Microsoft Visual Studio 2010*. Sebagai pengembangan terintegrasi atau *IDE* andalan yang dikeluarkan oleh *Microsoft*, *Visual Studio 2010* menambahkan perbaikan –perbaikan fitur dan fitur baru yang lebih lengkap dibandingkan versi *Visual Studio* sebelumnya, yaitu *Microsoft Visual Studio 2008*. *Visual studio* berisi beberapa jenis *IDE* pemrograman seperti *Visual Basic*, *Visual C++* , *Visual Web Developer*, *Visual C#* dan *Visual F#*. (Wahana Komputer, 2010,2)

Untuk melihat tampilan *visual studio 2010* dapat dilihat pada gambar II.11. sebagai berikut :

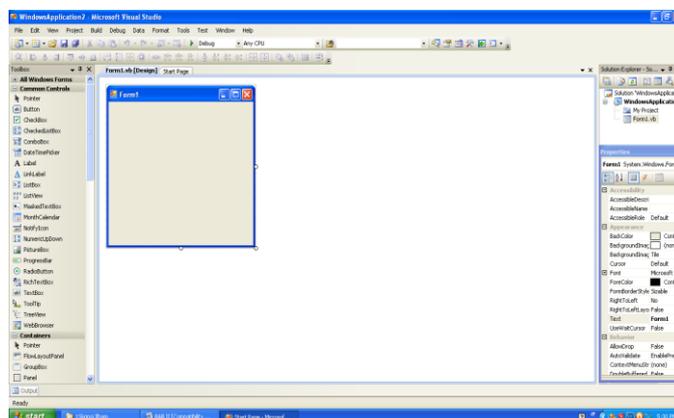


**Gambar II.7. Tampilan Utama Visual Studio 2010**

Sumber : (Wahana Komputer;2010;12)

Di dalam *Visual Studio.NET* menyediakan tampilan *Interface* yang sangat mudah untuk para pengguna merancang dan memodifikasi bentuk atau

*Interface* dari program yang akan dibuat, dimana pada tampilan ini difasilitasi dengan *Tool* dan fasilitas pendukung lainnya untuk mempermudah pengerjaannya.



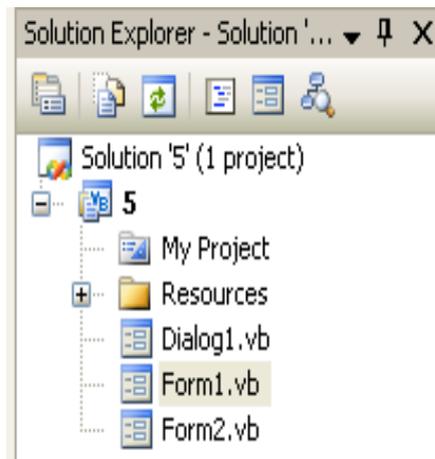
**Gambar II.8. Tampilan Area Kerja Visual Studio.NET**

**Sumber: (Wahana Komputer; 2010; 10)**

Dalam buku Belajar Pemrograman *Visual Basic* terbitan Wahana Komputer dmenjelaskan tampilan Microsoft *Visual Studio.NET*. Ada banyak hal penting yang harus diketahui oleh seorang programmer, diantaranya adalah:

a. *Solution Explorer*

Merupakan fitur yang terletak di sebelah kanan atas di bawah menu aplikasi yang digunakan untuk menampilkan daftar desain *form* dengan *struktur tree* dari project yang sedang dibuka. Dengan adanya fitur ini, memudahkan untuk berpindah antardesain *form* yang sudah anda buat secara cepat dan mudah.

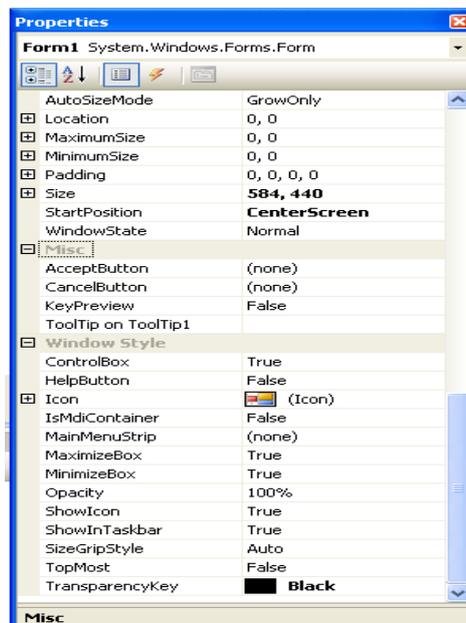


**Gambar II.9. Solution Explorer**

**Sumber: (Wahana Komputer; 2010;14)**

b. *Properties*

Merupakan fitur yang digunakan untuk melakukan pengaturan properti dari objek-objek yang digunakan dalam desain *form* yang akan dibuat, seperti pengaturan *text*, *visible*, *font*, *name* dan lain sebagainya.

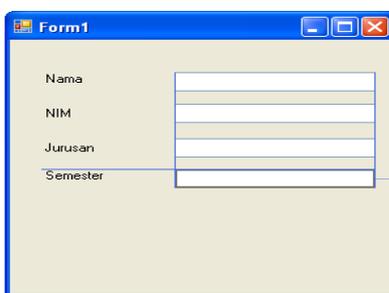


**Gambar II.10. Jendela Properties**

**Sumber: (Wahana Komputer; 2010; 15)**

c. *Form Designer*

Merupakan fitur yang digunakan untuk membuat desain antarmuka atau *interface* dari aplikasi yang akan dikembangkan. Dengan mengadopsi fitur *click and drop*, proses pemanbahan komponen pada *form* menjadi semakin dinamis dan mudah. Selain itu tersedia pula fitur *guidelines* yang memudahkan anda dalam menata komponen yang terdapat pada desain *form* yang sedang anda kerjakan.

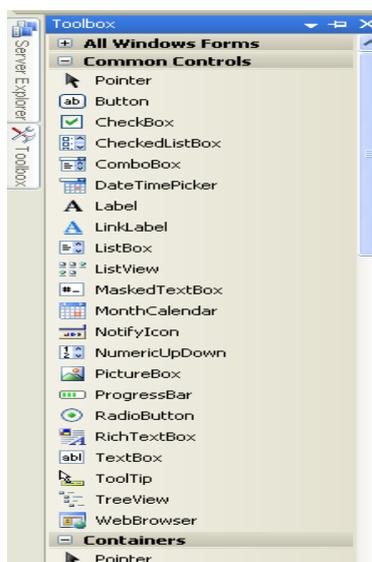


**Gambar II.11. *Form Designer***

**Sumber: (Wahana Komputer; 2010; 15)**

d. *Component Toolbox*

Merupakan fitur *Visual Basic.NET* yang digunakan untuk menampilkan daftar dari komponen baik visual maupun non visual yang bias ditambahkan ke dalam desain *form* yang akan dibuat. Dalam komponen *toolbox*, terdapat berbagai macam kategori sesuai dengan kelompok komponen yang diakses sehingga akan memudahkan dalam mencari komponen yang akan ditambahkan kedalam desain *form*.

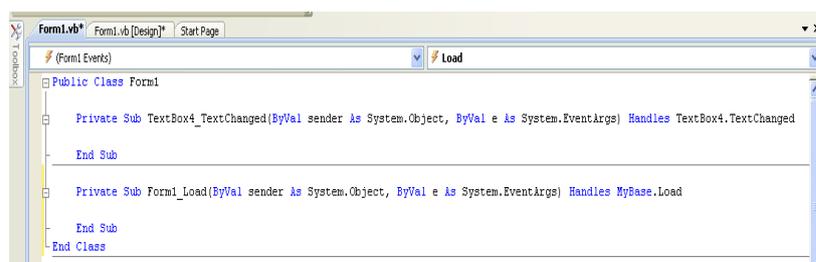


**Gambar II.12. Component Toolbox**

**Sumber: (Wahana Komputer; 2010: 15)**

e. *Code Editor*

Merupakan fitur yang digunakan untuk menambahkan kode program dari aplikasi atau *project* yang akan dikerjakan.



**Gambar II.13. Code Editor**

**Sumber: (Wahana Komputer; 2010; 16)**

## II.14. *Sql Server 2008*

*Sql Server 2008* merupakan sebuah terobosan terbaru dari *Microsoft* dalam bidang *database*. *Sql Server* adalah sebuah *DBMS (Database Management System)* yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan

dunia pengolahan data menyusul pendahuluannya seperti *IBM* dan *Oracle*. *SQL Server* 2008 dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server* 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data. (Wahana Komputer,;2010; 2)