

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem Pendukung Keputusan**

Pada dasarnya SPK merupakan pengembangan lebih lanjut dari sistem Informasi Manajemen terkomputerisasi yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakaiannya. Interaktif dengan tujuan untuk memudahkan integrasi antara berbagai komponen dalam proses pengambilan keputusan seperti prosedur, kebijakan, analisa, pengalaman dan wawasan manajer untuk mengambil keputusan yang lebih baik.

SPK adalah sistem yang dibangun untuk menyelesaikan berbagai masalah yang bersifat manajerial atau organisasi perusahaan yang dirancang untuk mengembangkan efektivitas dan produktivitas para menejer untuk menyelesaikan masalah dengan bantuan teknologi komputer. Hal lainnya yang perlu dipahami adalah bahwa SPK bukan untuk menggantikan tugas menejer akan tetapi hanya sebagai bahan pertimbangan bagi menejer untuk menentukan keputusan akhir.

Dalam menentukan suatu keputusan banayak faktor yang mempengaruhi pengambilan keputusan seorang pengambil keputusan, sehingga dipandang perlu untuk mengidentifikasi berbagai faktor yang penting dan mempertimbangkan tingkat pengaruh suatu faktor dengan faktor yang lainnya sebelum mengambil keputusan akhir, oleh karena itu secara spesifik penulis akan membahas salah satu permasalahan untuk menentukan warga miskin dengan langkah demi langkah

dengan menggunakan SPK untuk menghasilkan keputusan akhir yang disebut solusi dari suatu masalah.

### **II.1.1. Konsep Dasar Sistem Pendukung Keputusan**

Menurut Efraim Turban, Jay E. Aronson, dan Ting-Peng Liang. (2005 : 19) Konsep SPK pertama kali diperkenalkan pada awal tahun 1970-an oleh Scott Morton. Scott Morton mendefinisikan SPK sebagai "Sistem berbasis komputer interaktif, yang membantu para pengambil keputusan untuk menggunakan data dan berbagai model untuk memecahkan masalah-masalah tidak terstruktur." SPK dirancang untuk menunjang seluruh tahapan pembuatan keputusan yang dimulai dari tahap mengidentifikasi masalah, memilih data yang relevan, menentukan pendekatan yang digunakan dalam proses pembuatan keputusan, sampai pada kegiatan mengevaluasi pemilihan alternatif

### **II.1.2. Pengertian Sistem Pendukung Keputusan**

Menurut Efraim Turban, Jay E. Aronson, dan Ting-Peng Liang. (2005 : 136) Sistem Pendukung Keputusan ( Decision Support System) atau lebih dikenal dengan nama SPK adalah sebuah sistem yang dimaksudkan untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan semiterstruktur . DSS dimaksudkan untuk menjadi alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas mereka, namun tidak untuk menggantikan penilaian mereka. DSS ditujukan untuk keputusan – keputusan yang memerlukan penilaian atau pada keputusan yang sama sekali tidak dapat didukung oleh algoritma.

## II.2. *Database (Basis Data)*

*Database*, atau sering juga dieja basis data adalah kumpulan informasi yang disimpan didalam komputer secara sistematis. *Database* merupakan sekumpulan informasi yang saling berkaitan pada suatu subjek tertentu dengan tujuan tertentu. *Database* adalah susunan record data operasional lengkap dari suatu organisasi atau perusahaan, yang diorganisir dan disimpan secara terintegrasi dengan menggunakan metode tertentu dalam komputer sehingga mampu memenuhi informasi yang optimal yang dibutuhkan oleh para pengguna.

Peranan database dalam sistem informasi disebut dengan database sistem. Sistem basis data (*Database System*) ini adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi.

Menurut Ade Rahmat dan Yono Suryadi (2008 : 1) *Database managemen system* (DBMS) adalah sebuah program komputer yang memungkinkan pengguna untuk membuat dan memelihara database sehingga pengguna dapat memasukkan, mengubah, menghapus, memanipulasi dan memperoleh kembali data tersebut dengan mudah.

## II.3. *Visual Studio .Net*

Menurut Suryanto Suharli (2005 : 13) *Visual Studio .Net* merupakan *core* dari pembuatan aplikasi berbasis .NET. merupakan lingkungan pemrograman yang mempermudah tahapan desain, *development*, *debuging*, dan *deployment* dari

aplikasi berbasis . NET dan XML Web service, serta meningkatkan efisiensi *developer* dengan menyediakan lingkungan pemrograman yang sudah biasa digunakan dan bisa di share.

Sebagai development tool, Visual Studio . NET menyediakan dukungan untuk berbagai bahasa pemrograman, tools untuk membuat web, windows dan XML web service, data akses, dan error handling, termasuk debugging lokal, remote dan tracing, bahkan crosslanguage debugging.

#### **II.4. SQL Server 2005**

*SQL Server* merupakan sebuah aplikasi yang digunakan untuk merancang, mengolah dan menyimpan *database*. *Database* adalah kumpulan fakta-fakta sebagai representasi dari dunia nyata yang berhubungan dan mempunyai arti tertentu.

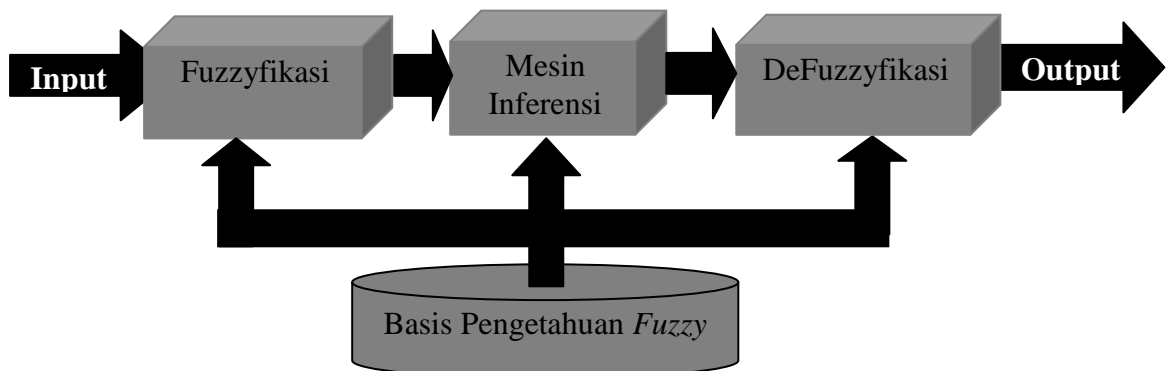
Menurut Emma Utami dan Sukrisno. (2008 : 01) *SQL (Structured Query Language)* pada dasarnya adalah bahasa komputer standar yang ditetapkan untuk mengakses dan memanipulasi sistem database. Sebuah database berisi satu tabel atau lebih dan memiliki nama yang berbeda untuk masing – masing tabel. Masing – masing tabel memiliki satu kolom (*field*) atau lebih dan memiliki baris (*record*). Query digunakan untuk mengakses dan mengolah database.

## II.5. Logika *Fuzzy*

Menurut T. Sutojo, Edi Mulyanto, dan Dr. Vincent Suhartono (2011 : 211) Logika *Fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang cocok diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, embedded system, jaringan PC, multi-channel atau workstation berbasis akuisisi data, dan sistem kontrol.

### II.5.1. Cara Kerja Logika *Fuzzy*

Menurut T. Sutojo, Edi Mulyanto, dan Dr. Vincent Suhartono (2011 : 232) Untuk memahami cara kerja logika *fuzzy*, perhatikan struktur elemen dasar sistem inferensi *fuzzy* berikut:



**Gambar II. 1. Struktur Sistem Inferensi *Fuzzy***

(Sumber : T. Sutojo, Edi Mulyanto, dan Dr. Vincent Suhartono, 2010 : 232)

Keterangan::

1. Basis pengetahuan *Fuzzy*: kumpulan *rule-rule Fuzzy* dalam bentuk pernyataan IF...THEN.

2. *Fuzzyfikasi*: proses untuk mengubah *input* sistem yang mempunyai nilai tegas menjadi variabel linguistik menggunakan fungsi keanggotaan yang disimpan basis pengetahuan *fuzzy*.
3. Mesin Inferensi: proses mengubah *input fuzzy* menjadi *output fuzzy* dengan cara mengikuti aturan-aturan (*IF-THEN Rules*) yang telah ditetapkan pada basis pengetahuan *fuzzy*.
4. DeFuzzyfikasi: mengubah *output fuzzy* yang diperoleh dari mesin inferensi menjadi nilai tegas menggunakan fungsi keanggotaan yang sesuai dengan saat dilakukan fuzzyfikasi.

Cara kerja logika *Fuzzy* meliputi beberapa tahapan sebagai berikut:

1. Fuzzyfikasi
2. Pembentukan basis pengetahuan *Fuzzy* (*Rule* dalam bentuk *IF...THEN*)
3. Mesin inferensi( Fungsi aplikasi *Max-Min* atau *Dot-Product*)
4. DeFuzzyfikasi

Banyak cara melakukan DeFuzzyfikasi, di antaranya metode berikut.

- a. Metode Rata-Rata (*Average*)

$$z^* = \frac{\sum \mu_i z_i}{\sum \mu_i}$$

- b. Metode Titik Tengah (*Center Of Area*)

$$z^* = \frac{\int \mu(z)zdz}{\int \mu(z)dz}$$

### II.5.2. Metode Tsukamoto

Menurut T. Sutojo, Edi Mulyanto, dan Dr. Vincent Suhartono (2011 : 233)

Secara umum bentuk model *fuzzy* Tsukamoto adalah:

*If (X IS A) and (Y IS B) Then (Z IS C)*

Diamana A, B, dan C adalah himpunan *Fuzzy*.

Misalkan diketahui 2 *rule* sebagai berikut:

IF (x is A<sub>1</sub>) AND (y is B<sub>1</sub>) THEN (z is C<sub>1</sub>)

IF (x is A<sub>2</sub>) AND (y is B<sub>2</sub>) THEN (z is C<sub>2</sub>)

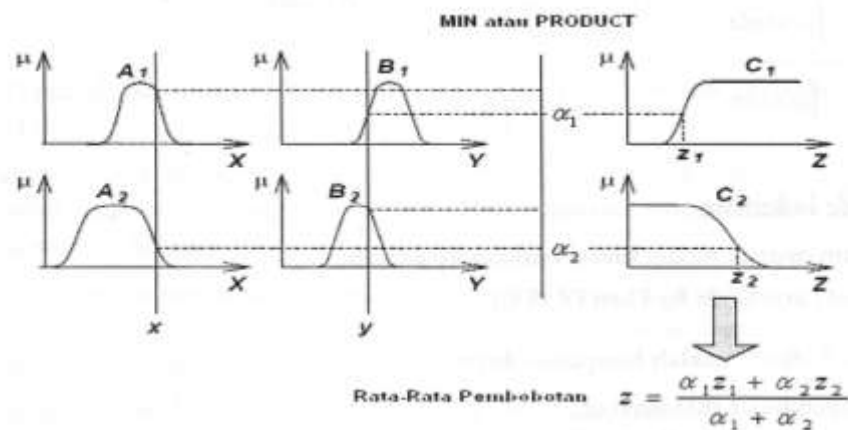
Dalam inferensinya, metode Tsukamoto menggunakan tahapan sebagai berikut:

1. Fuzzyfikasi
2. Pembentukan basis pengetahuan *Fuzzy* (*Rule* dalam bentuk *IF...THEN*)
3. Mesin inferensi( Fungsi aplikasi *Max-Min* atau *Dot-Product*)
4. DeFuzzyfikasi

Menggunakan metode Rata-Rata (*Average*)

$$z^* = \frac{\sum \mu_i z_i}{\sum \mu_i}$$

Gambar II.2. menunjukkan skema penalaran fungsi implikasi MIN dan proses DeFuzzyfikasi dilakukan dengan cara mencari nilai rata-ratanya.



**Gambar II. 2. Skema Penalaran Fungsi Implikasi Min atau Product Dan  
Proses DeFuzzyfikasi**

(Sumber : T. Sutojo, Edi Mulyanto, dan Dr. Vincent Suhartono, 2010 : 234)

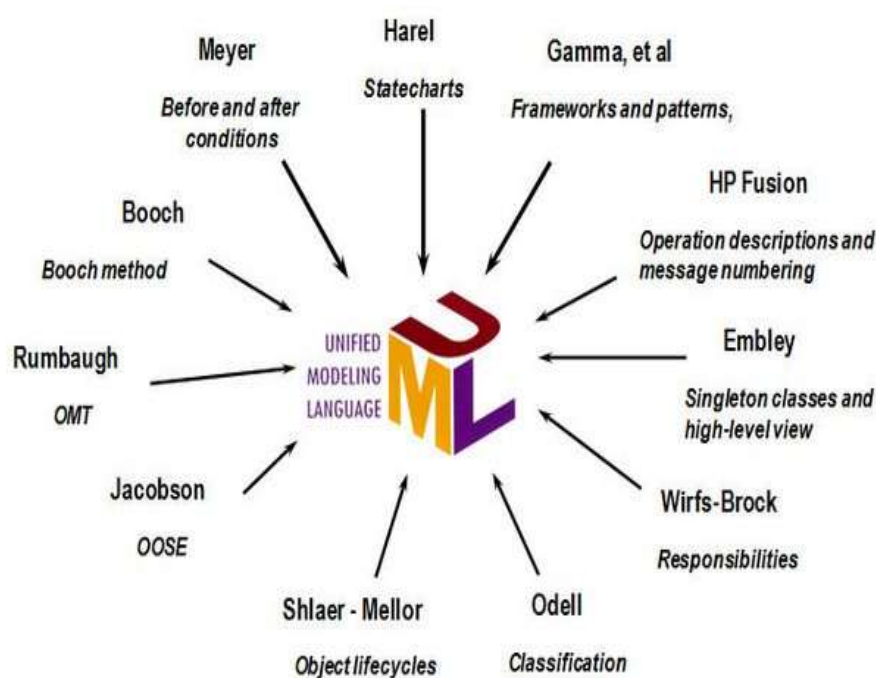
## II.6. UML (*Unified Modeling Language*)

Menurut Munawar ( 2005 : 17 ) UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*Sharing*) dan mengkomunikasikan rancangan dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modelling Technique* (OMT) dan *Object-Oriented Software Engineering* (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisa dan desain ke dalam empat tahapan efektif, yaitu: identifikasi kelas-kelas dan obyek-obyek, identifikasi semantik dari hubungan obyek dan kelas tersebut, perincian interface dan implementasi. Keunggulan metode Booch adalah pada detil dan kayanya dengan notasi dan elemen. Pemodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada analisa terstruktur dan pemodelan *entity-relationship*. Tahapan utama dalam metodologi ini adalah analisa, design obyek dan implementasi. Keunggulan metode ini adalah dalam penotasian yang mendukung semua konsep OO. Metode OOSE dari Jacobson lebih memberi penekanan pada *use case*. OOSE memiliki tiga tahapan yaitu membuat model requirement dan analisis, design dan implementasi, dan model pengujian (*test model*). Keunggulan

metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

Dengan UM, metode Booch, OMT dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambah dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam daripada metode lainnya. Gambar berikut adalah unsur-unsur yang membentuk UML.



**Gambar II.3. Unsur – unsur pembentuk UML**

(Sumber : Munawar, 2005 : 18)

### II.6.1. Tujuan Pemanfaatan UML

Berikut tujuan utama dalam desain UML adalah :

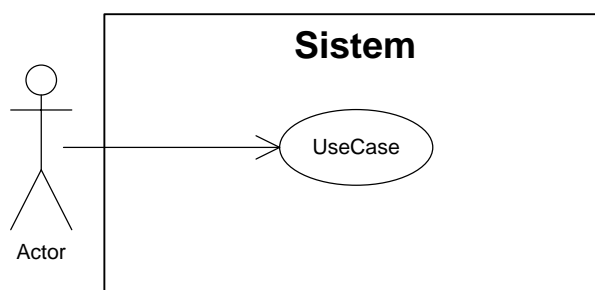
1. Memberikan model yang siap pakai bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek – praktek yang terbaik yang terdapat dalam pemodelan. ([http://www.cetus-links/org-oo\\_uml.html](http://www.cetus-links/org-oo_uml.html))

### II.6.2. Stuktur Diagram dalam Unified Modeling Language (UML)

Ada 5 (lima) macam diagram dalam Unified Modeling Language (UML), yaitu :

a. *Use Case Diagram*

Menurut Munawar (2005:64) Diagram use case menunjukkan 3 aspek dari system yaitu actor, use case dan system/sub system boundary. Actor mewakili peran orang, system yang lain atau ketika berkomunikasi dengan use case.



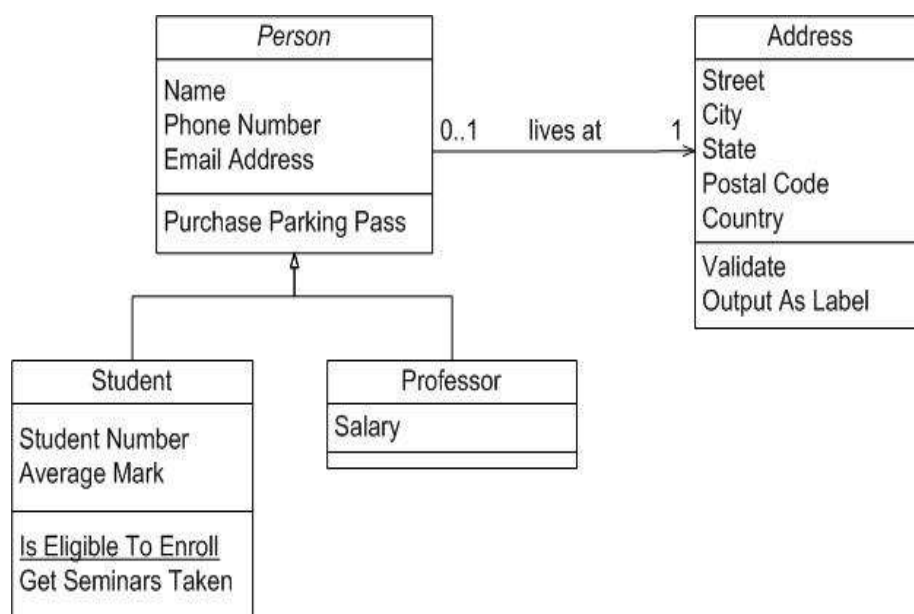
**Gambar II.4. Use Case Diagram**

**(Sumber : Munawar, 2005 : 64)**

b. *Class Diagram*

Menurut Munawar (2005:219). Class

diagram sangat membantu dalam visualisasi struktur kelas dari suatu sistem. Hal ini disebabkan karena class adalah deskripsi kelompok obyek – obyek dengan property, perilaku (operasi) dan relasi yang sama.



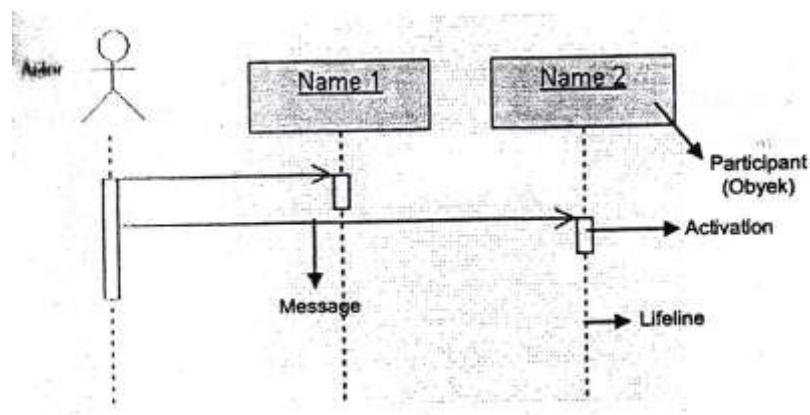
**Gambar II.5. Class Diagram**

**(Sumber: Munawar, 2005 : 220)**

c. *Sequence Diagram*

Menurut Munawar (2005:87) Diagram ini digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesen) yang diletakkan diantara obyek-obyek use case. Komponen utama *sequence diagram* terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*.

Berikut adalah Simbol-simbol yang ada pada *Sequence Diagram* yaitu:



**Gambar II.6.** Simbol –simbol yang ada pada *Sequence Diagram*

(Sumber: Munawar,2005 : 89)

d. *State Chart Diagram*

Menurut Munawar ( 2005:73 ). State diagram menyediakan variasi symbol dan sejumlah ide untuk pemodelan. Simbol UML untuk State transition diagram adalah segi empat yang tiap pojoknya dibuat *rounded* . Titik awalnya menggunakan lingkaran solid yang diarsir dan diakhiri dengan mata.

Berikut adalah simbol UML untuk *State Chart Diagram*:





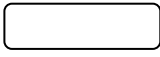
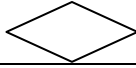


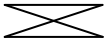
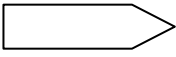
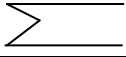

**Gambar II.7.** *State Chart Diagram*

(Sumber: Munawar, 2005 : 74)

e. *Activity Diagram*

Menurut Munawar (2005:109) Activity diagram adalah teknik mendeskripsikan logika procedural, proses bisnis dan aliran kerja dalam kaskus, activity diagram mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah activity diagram bisa mendukung perilaku paralel sedangkan flowchart tidak. Berikut simbol-simbol yang sering digunakan pada saat pembuatan *activity diagram* yaitu:

**Tabel II.1. Simbol Activity Diagram**

Notasi	Keterangan
	Titik Awal
	Titik Akhir
	<i>Activity</i>
	Pilihan untuk pengambilan keputusan
	<i>Fork</i> digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Rake</i> menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran Akhir ( <i>Flow Final</i> )

(Sumber : Munawar, 2005:110)