BAB II

TINJAUAN PUSTAKA

2.1. Landasan Teori

2.1.1. Pengertian Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK), adalah suatu sistem informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur ataupun tidak terstruktur dengan menggunakan data dan model . Tujuan adanya SPK, untuk mendukung pengambil keputusan memilih alternatif hasil pengolahan informasi dengan model-model pengambil keputusan serta untuk menyelesaikan masalah yang bersifat semi terstruktur dan tidak terstruktur [5].

SPK dirancang untuk membantu pengambil keputusan dalam memecahkan masalah. SPK dirancang sedemikian rupa sehingga dapat digunakan atau dioperasikan dengan mudah oleh orang yang tidak memiliki dasar kemampuan pengoperasian komputer yang tinggi dan bersifat alternatif, serta SPK dirancang dengan menekankan pada aspek kemampuan adaptasi yang tinggi.

2.1.2. Manfaat Utama Sistem Pendukung Keputusan

a. Mampu untuk mendukug permasalahan yang kompleks,
b. Memberikan jawaban yang cepat untuk situasi yang tidak diharapkan dari hasil perubahan kondisi,
c. Mampu untuk mencoba beberapa strategi yang berada di bawah konfigurasi yang berbeda, dengan cepat dan obyektif,

d. Wawasan baru dan pengetahuan. Pemakai dapat terbuka dengan wawasan baru melalui komposisi model dan kepekaan yang luas analisis "*what-if*",
e. Memudahkan komunikasi. Kumpulan data dan pelaksanaan pembuatan model dijalankan dengan partisipasi aktif pemakai, sehingga sangat membantu diantara manajer,
f. Memperbaiki kendali manajemen dan memperbaiki performansi organisasi,
g. Keputusan bersifat obyektif. Keputusan yang dihasilkan SPK lebih konsisten dan obyektif daripada keputusan yang dibuat secara intuisi,
h. Memperbaiki keefektifan manajerial. Memperbolehkan manajer untuk melakukan tugas dalam waktu yang singkat,
i. Merperbaiki produktifitas analisis.

Tiga tingkatan teknologi yang digunakan dalam pengembangan SPK terdiri dari :

1. SPK Spesifik (*Spesific DSS*)

   Paket yang terdiri dari pernagkat keras dan lunak digunakan oleh sekelompok pengambil keputusan tertentu untuk menangani permasalahan khusus. SPK spesifik ini dikembangkan dari suatu *tools* SPK dan atau *Generator* SPK. Tujuan dari DSS yaitu ;

   a. Membantu manajer membuat keputusan untuk memecahkan *masalah semi terstruktur*.
   b. Mendukung penilaian manajer bukan mencoba menggantikannya.
   c. Meningkatkan *effektifitas* pengambilan keputusan manajer daripada efisiensinya.

2. Peralatan SPK (*DSS Tools*)

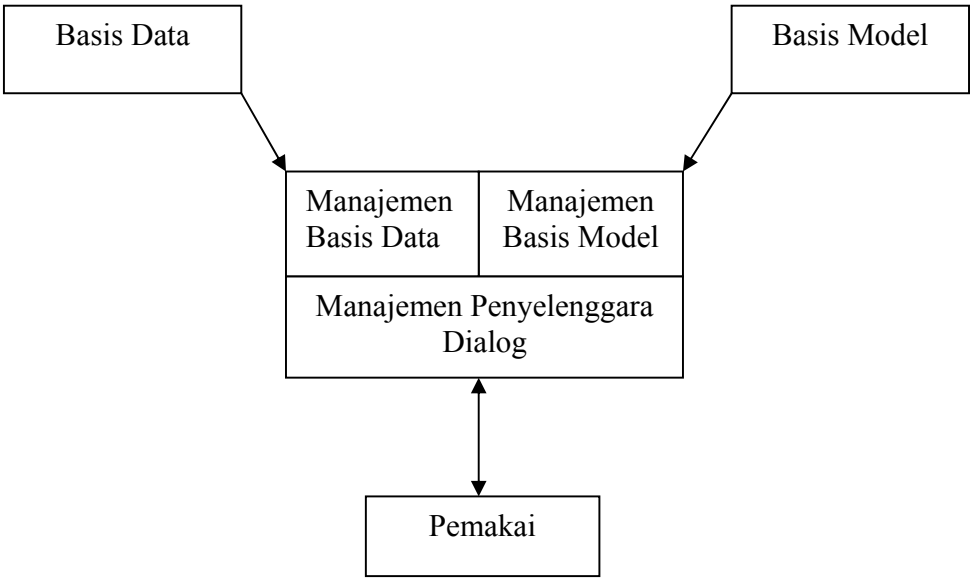
   Merupakan elemen-elemen perangkat keras atau lunak yang dapat dipergunakan untuk mengembangkan SPK spesifik maupun pembangkit SPK. Meskipun peralatan ini mampu membuat SPK spesifik secara langsung, namun mengembangkan SPK spesifik dengan pembangkit SPK jauh lebih mudah dan efisien.

3. Pembangkit SPK (*DSS Generator*)

   Merupakan "paket" dari kumpulan perangkat keras atau lunak yang menyediakan sekumpulan kemampuan untuk membuat SPK spesifik dengan cepat dan mudah.

2.1.3. Komponen-Komponen Sistem Pendukung Keputusan

SPK memiliki tiga sub sistem utama yang menetukan kapabilitas teknis SPK seperti pada gambar :



Gambar 2.1 Komponen Sistem Pendukung Keputusan

Komponen SPK dibagi atas tiga bagian yaitu :

i. Sub Sistem Basis Data

Merupakan komponen SPK penyedia data bagi sistem. Data dimaksud disimpan dalam suatu pangkalan data (*database*) yang diorganisasi yang disebut dengan sistem manajemen pangkalan data (Data Base Management System atau DBMS).

ii. Sub Sistem Basis Dialog

Model adalah suatu peniruan dari alam nyata. Kendala yang sering kali dihadapi dalam merancang suatu model adalah bahwa model yang disusun ternyata tidak mampu mencerminkan seluruh variable alam nyata. Sehingga keputusan yang diambil didasarkan pada model tersebut menjadi tidak akurat dan tidak sesuai dengan kebutuhan. Oleh karena itu, dalam menyimpan barbagai model pada sistem pangkalan model harus tetap dijaga fleksibilitasnya. Artinya harus ada fasilitas yang mampu membantu pengguna untuk memodifikasi atau mneyempurnakan model seiring dengan perkembangan pengetahuan.

iii. Sub Sistem Penyelenggara Dialog (Basis Dialog)

Merupakan alat komunikasi antara pemakai dengan sistem atau dikenal dengan fasilitas antar muka. Bagian ini harus mampu memenuhi keinginan pemakai, bersifat komunikatif.

2.1.4. Tahap Pengambilan Keputusan

Tahap pengambilan keputusan, yaitu :

1. Penelusuran (*Intellegence*)

Merupakan tahap pendefinisian informasi yang dibutuhkan yang berkaitan dengan persoalan yang dihadapi serta keputusan yang akan diambil. Langkah ini sangat menentukan ketepatan keputusan yang akan diambil, karena sebelum suatu tindakan diambil, tentunya persoalan yang dihadapi harus dirumuskan terlebih dahulu secara jelas.

2. Perancangan (*Design*)

Merupakan tahap analisis dalam kaitan mencari atau merumuskan alternatif-alternatif pemecah masalah. Setelah permasalahan dirumuskan dengan baik, maka tahap berikutnya adalah merancang atau membangun model pemecahan masalahnya dan menyusun berbagai alternatif pemecah masalah.

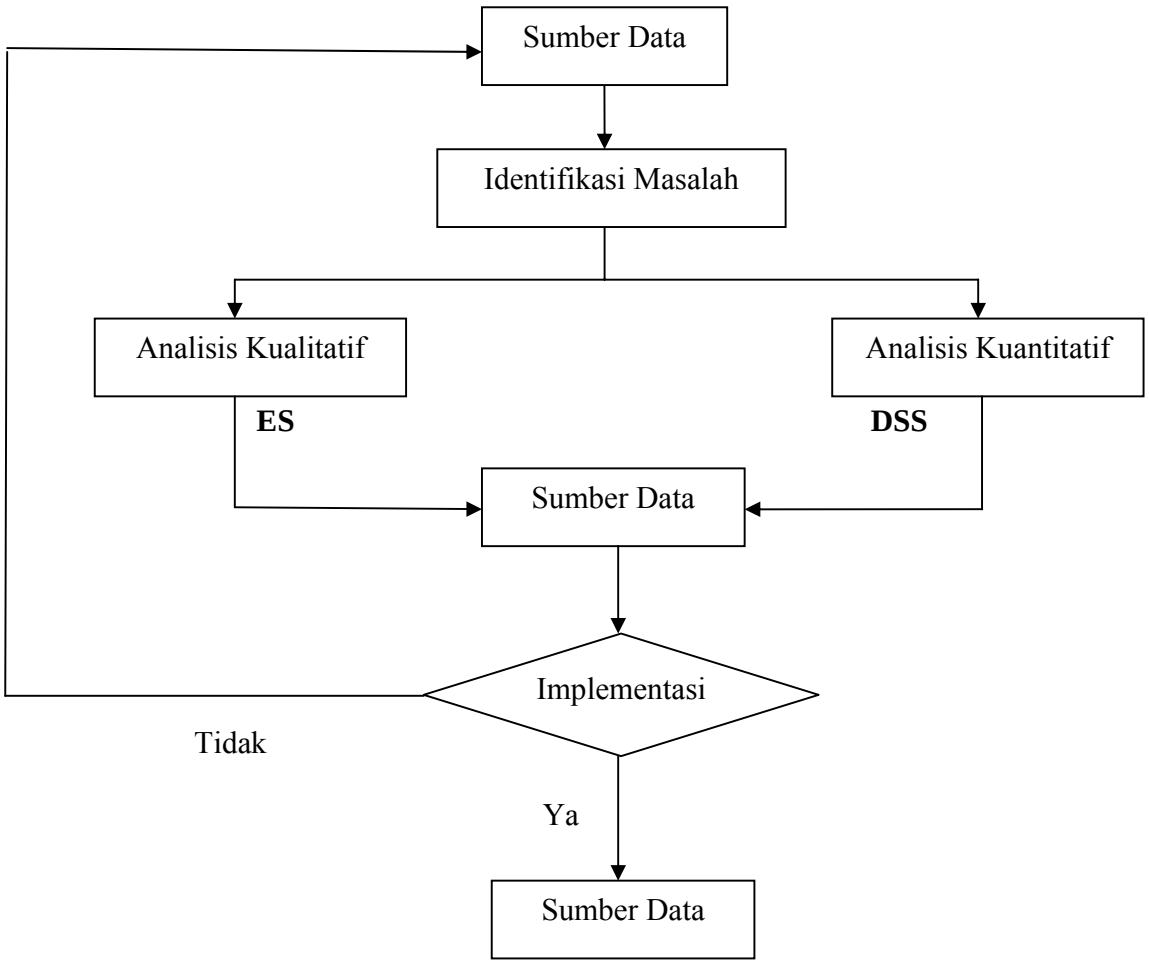
3. Pemilihan (*Choice*)

Dengan mengacu pada rumusan tujuan serta hasil yang diharapkan selanjutnya manajemen memilih alternatif solusi yang diperkirakan paling sesuai. Pemilihan alternatif ini akan mudah dilakukan kalau hasil yang diinginkan terukur atau memiliki nilai kualitas tertentu.

4. Implementasi (*Implementation*)

Merupakan tahap pelaksana dari keputusan yang telah diambil. Pada tahap ini perlu disusun serangkaian tindakan yang terencana, sehingga hasil keputusan dapat dipantau atau diselesaikan apabila diperlukan perbaikan-perbaikan. Dalam

kejadiannya keputusan diterapkan suatu solusi diusulkan, satu Decision Support System memberikan dukungan seperti pada gambar :



Gambar 2.2 Dukungan Komputer Untuk Proses SPK

2.1 5. Logika *Fuzzy*

Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang input kedalam suatu ruang output [4].

1. Himpunan *Fuzzy*

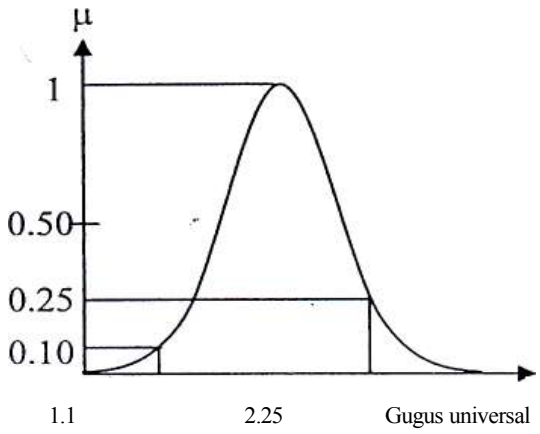
Dalam himpunan biasa *{crisp set)* keanggotaan setiap elemen himpunan *universal* pada suatu himpunan dinyatakan dengan anggota atau bukan anggota himpunan tersebut. Keanggotaan ini diberikan oleh suatu fungsi yang disebut fungsi keanggotaan. Fungsi keanggotaan memberikan nilai 1 untuk menyatakan anggota dan 0 untuk menyatakan bukan anggota [2].

Himpunan *fuzzy* merupakan pengembangan dari himpunan biasa. Fungsi keanggotaannya tidak hanya memberikan nilai 0 dan 1, tapi nilai yang berada pada suatu selang tertentu, biasanya dalam selang [0,1], sehingga suatu elemen dapat memiliki derajat keanggotaan 0, 0.82 atau 1. Nilai yang diberikan oleh fungsi keanggotaan disebut derajat keanggotaan *(degree of membership).*

Apabila U menyatakan himpunan *universal* dan A adalah himpunan *fuzzy* dalam U, maka A adalah himpunan pasangan terurut sebagai berkut:

$A=\{(u, \mu_A(\mu)) \mid u \in U\}$

Dengan $\mu_A(u)$ adalah fungsi keanggotaan yang memberikan nilai derajat kanggotaan u terhadap himpunan *fuzzy* A, yaitu : $\mu_{A : U} \rightarrow [0,1]$. Misalkan fungsi keanggotaan himpunan *fuzzy* A seperti terlihat pada Gambar 2.5. Dari Gambar 2.3 dapat diketahui bahwa $\mu_A(1.1) = 0.10$, dan $\mu_A(2.25) = 0$.



Gambar 2.3 Fungsi keanggotaan himpunan fuzzy

Himpunan *fuzzy* memiliki 2 atribut, yaitu :

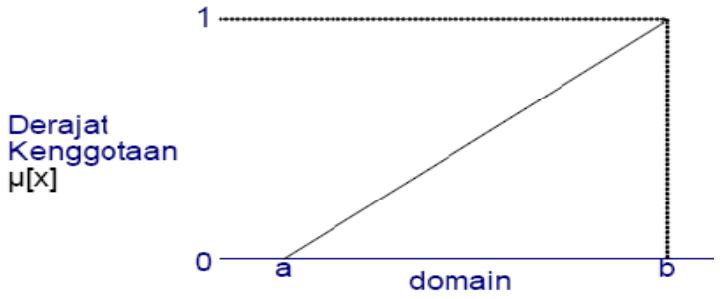
a. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti : BESAR, SEDANG, KECIL.
b. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 12,10,8, dsb.

2. Fungsi Keanggotaan

Fungsi keanggotaan *{membership function)* adalah suatu kurva yang menunjukan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Ada dua cara mendefinisikan keanggotaan himpunan *fuzzy,* yaitu secara numeris dan fungsional. Definisi numeris menyatakan fungsi derajat keanggotaan sebagai vektor jumlah yang tergantung pada tingkat diskretisasi. Misalnya, jumlah elemen diskret dalam semesta pembicaraan. Definisi fungsional menyatakan derajat keanggotaan sebagai

batasan ekspresi analitis yang dapat dihitung. Standar atau ukuran tertentu pada fungsi keanggotaan secara umum berdasar atas semesta X bilangan real.

- Representasi Linear Ada 2 kemungkinan himpunan fuzzy linear yaitu: Kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak kekanan menuju nilai domain yang memiliki derajat keanggotaan lebih tinggi.

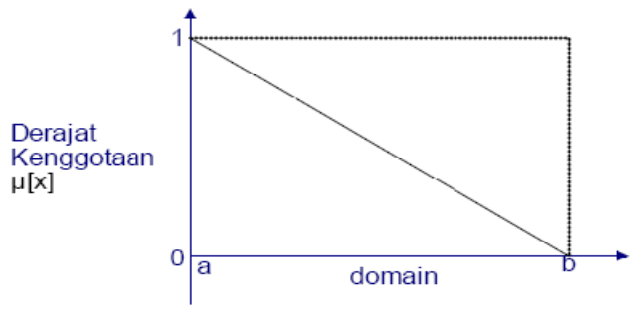


Gambar 2.4 Grafik Fungsi Keanggotaan Linear Naik

Fungsi Keanggotaan :

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

Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun kenilai domain yang memiliki derajat keanggotaan lebih rendah.



Gambar 2.5 Grafik Fungsi Keanggotaan Linear Turun

FungsiKeanggotaan:

$$\mu[x] = \begin{cases} (x-a)/(b-a) ; a \le x \le b \\ 0; \qquad x \ge b \end{cases}$$

1. Basis Data *Fuzzzy* Model Tahani

Selama ini sudah ada beberapa penelitian tentang basis data *fuzzy*. Salah satu diantaranya adalah model Tahani. Basis *data, fuzzy* model Tahani masih tetap menggunakan relasi umum (standar), hanya saja model ini menggunakan teori himpunan *fuzzy* untuk mendapatkan informasi pada *query*-nya [2].

Sebagian besar basis data standar diklarifikasikan berdasarkan bagaimana data tersebut dipandang oleh pengguna. Misalkan data karyawan yang tersimpan pada tabel DT_KARYAWAN dengan *field* NIP, nama, tgl_lahir, gaji_per_bulan seperti pada tabel:

Tabel 2.1 Data Karyawan Mentah

NIP	Nama	Tgl Lahir	Thn Masuk	Gaji / Bln
01	Lia	03-06-1972	1996	750000
02	Iwan	23-09-1954	1985	1500000
03	Sari	12-12-1966	1988	1255000
04	Andi	06-03-1965	1998	1040000
05	Budi	04-12-1960	1990	950000
06	Amir	18-11-1963	1989	1600000
07	Rian	28-05-1965	1997	1250000
08	Kiki	09-07-1971	2001	550000
09	Alda	14-08-1967	1999	735000
10	Yoga	17-09-1977	2000	860000

Kemudian dari tabel DT_KARYAWAN, diperoleh suatu tabel temporer untuk menghitung umur karyawan dan masa kerjanya. Tabel tersebut diberi nama dengan tabel KARYAWAN.

Tabel 2.2 Data Karyawan Setelah diolah

NIP	Nama	Umur (th)	Masa Kerja (th)	Gaji / Bln
01	Lia	30	6	750000
02	Iwan	48	17	1500000
03	Sari	36	14	1255000
04	Andi	37	4	1040000
05	Budi	42	12	950000
06	Amir	39	13	1600000
07	Rian	37	5	1250000
08	Kiki	32	1	550000
09	Alda	35	3	735000
10	Yoga	25	2	860000

Dengan menggunakan basis data standar, maka dapat dicari data-data karyawan dengan spesifikasi tertentu menggunakan *query*. Misal bila ingin mendapatkan informasi tentang nama-nama karyawan yang usianya kurang dari 35 tahun, maka *query-nya* adalah :

```
SELECT NAMA FROM KARYAWAN WHERE (UMUR < 35)
```

Hasil : Lia, Kiki, Yoga.

Query untuk mendapatkan informasi karyawan gajinya lebih dari 1 juta:

```
SELECT NAMA FROM KARYAWAN WHERE (GAJI > 1000000)
```

Hasil : Iwan, Sari, Andi, Amir, Rian.

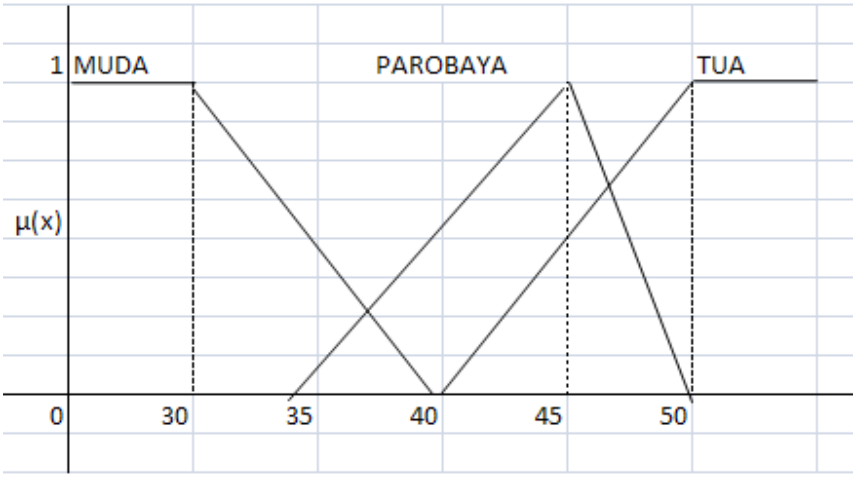
Query untuk mendapatkan informasi karyawan masa kerja kurang dari atau sama dengan 5 tahun tetapi gaji sudah lebih dari 1 juta :

```
SELECT NAMA FROM KARYAWAN WHERE (MASAKERJA <= 5) AND (GAJI
> 1000000)
```

Hasil : Andi, Rian.

Pada kenyataannya seseorang kadang membutuhkan informasi dari data-data yang bersifat *ambiguous,* contoh "mencari data karyawan yang masih muda dan memiliki gaji yang tinggi". Apabila hal ini terjadi, maka digunakanlah basisdata *fuzzy*. Salah satu diantaranya adalah model Tahani. Basisdata *fuzzy* model Tahani masih tetap menggunakan relasi standar, hanya saja model ini

menggunakan teori himpunan *fuzzy* untuk mendapatkan informasi query-nya. Misalkan mengkategorikan usia karyawan diatas kedalam himpunan : MUDA, PAROBAYA, dan TUA



Gambar 2.6 Fungsi Keanggotaan untuk variabel usia

Tabel 2.5 menunjukkan tabel karyawan berdasarkan umur dengan derajat keanggotaannya pada setiap himpunan.

Tabel 2.3 Karyawan Berdasarkan Umur

NIP	Nama	Umur (th)	Derajat Keanggotaan μ[x]		
			MUDA	PAROBAYA	TUA
01	Lia	30	1	0	0
02	Iwan	48	0	0.4	0.8
03	Sari	36	0.4	0.1	0
04	Andi	37	0.3	0.2	0
05	Budi	42	0	0.7	0.2
06	Amir	39	0.1	0.4	0
07	Rian	37	0.3	0.2	0
08	Kiki	32	0.8	0	0
09	Alda	35	0.5	0	0
10	Yoga	25	1	0	0

Fungsi Keanggotaan :

a) $\mu MUDA[x] = \begin{cases} 1 \\ 40 - x/10 \\ 0 \end{cases}$

jika $x \leq 30$
jika $30 < x < 45$
jika $x \geq 40$

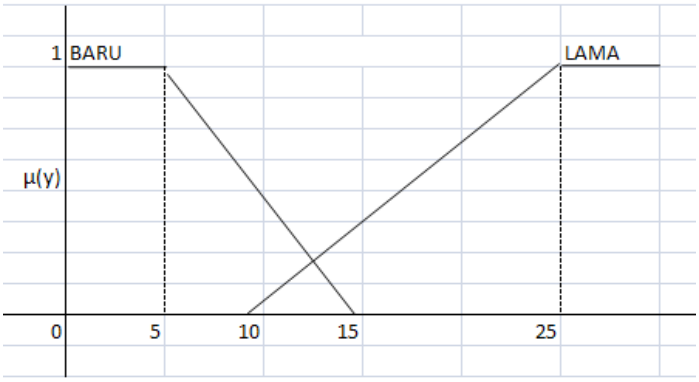
b) $\mu PAROBAYA\ [x] = \begin{cases} 1 \\ (x - 35)/(45 - 35) \\ (50 - x)/(50 - 45) \end{cases}$

jika $x \leq 35$ atau $x \geq 50$
jika $35 < x < 45$
jika $45 \leq x \leq 50$

c) $\mu TUA\ [x] = \begin{cases} 0 \\ (x - 40)/(50 - 40) \\ 1 \end{cases}$

jika $x \leq 40$
jika $40 < x < 50$
jika $x \geq 50$

Variabel masa kerja dikategorikan dalam himpunan : BARU dan LAMA.



Gambar 2.7 Fungsi Keanggotaan untuk masa kerja

2.1.6 Analisis Sistem

Analisis sistem (*system analysis*) dapat didefinisikan sebagai berikut :

"*Penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya*" [8].

Tahap analisis sistem dilakukan setelah tahap perencanaan (*system planning*), dan sebelum tahap desain sistem (*system design*). Tahap analisis merupakan tahap yang sangat kritis dan sangat penting, karena kesalahan pada tahap ini akan menyebabkan kesalahan pada tahap selanjutnya.

Analisis yang dilakukan menggunakan metode analisis terstruktur, dimana metode analisis terstruktur ini banyak memakai *tool-tool* dalam mempresentasikan suatu sistem, sehingga penguraian sistem dapat terstruktur dengan baik, jelas dan lebih mudah dipahami.

2.1.6.1 Diagram Konteks

Diagram konteks merupakan diagram yang mengandung satu proses yang menggambarkan hubungan keterkaitan antara sistem dengan pihak-pihak diluar lingkungan sistem dan posisi sistem didalam lingkungan tersebut. Pihak-pihak tersebut merupakan pihak-pihak yang membutuhkan informasi dan data dari sistem ataupun pihak-pihak yang menjadi sumber informasi dan data bagi sistem. Hubungan

keterkaitannya digambarkan sebagai aliran informasi dan data yang masuk ke dalam sistem dan keluar dari sistem.

2.1.6.2 Data Flow Diagram

DFD adalah sebuah teknik grafik yang menggambarkan aliran data dan transformasi yang digunakan sebagai perjalanan data dari masukan menuju ke keluaran. DFD dapat diartikan juga sebagai model jaringan dari sebuah sistem. DFD dapat menggambarkan proses-proses yang terjadi dan aliran data diantaranya.

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 lingkungan fisik dimana data tersebut disimpan. Secara umum DFD dapat diartikan sebagai salah satu *tools* untuk analisis sistem yang dapat bermanfaat untuk menggambarkan proses, aliran data, *entity* yang terlibat serta *data store* yang digunakan dalam sistem yang dipelajari. Dengan menuangkan hasil analisis ke dalam DFD, seorang analis dapat memahami sistem yang sedang dipelajari dengan mudah dan baik.

Dibawah ini terdapat simbol-simbol untuk DFD yaitu:

1. *Proses*, suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, prosedur atau alat yang digunakan untuk mentransformasikan data.
2. *Data Flow* (Arus Data), data yang mengalir dengan arah tertentu dari asal ke tujuan. Data yang mengalir dapat berupa dokumen, surat atau bentuk lainnya.

3. *Data Store* (Penyimpanan Data), digunakan untuk menyimpan dan mengambil data oleh proses. Data yang disimpan dapat berupa data yang terkomputerisasi maupun tidak terkomputerisasi.
4. *Source* atau *destination* atau dikenal juga dengan *external entity*, berupa orang, organisasi atau sistem lain yang berada diluar batas sistem yang berinteraksi dengan sistem yang sedang dikembangkan.

Dalam DFD terdapat beberapa aturan dasar penyusunan DFD yang dapat membantu untuk mempermudah penggambaran diagram arus data, diantaranya :

1. Setiap data yang dihasilkan atau keluar dari proses harus didasarkan pada data yang masuk ke proses tersebut.
2. Semua aliran data yang dihasilkan harus diberi nama, nama yang diberikan harus mencerminkan aliran data antara proses, *data store* dan *entity* yang ada.
3. Hanya data yang diperlukan untuk melakukan suatu proses saja yang harus digunakan sebagai input suatu proses.
4. Suatu proses harus tergantung pada input dan output yang masuk ke dalam proses itu saja, tidak perlu memperhatikan apa yang terjadi pada proses lainnya.
5. Setiap proses yang ada harus merupakan proses yang berjalan terus menerus, setiap proses harus diasumsikan siap menangani fungsi atau kerja setiap proses.

Agar representasi sistem dalam DFD mudah dipahami, maka DFD disusun dalam bentuk bertingkat (*leveled*) yang merupakan rincian lanjut dari proses pada level yang sebelumnya. Dimulai dari tingkat yang tertinggi dilakukan identifikasi proses apa saja yang ada dan data apa saja yang mengalir antar proses yang ada. Selanjutnya, dilakukan analisis terhadap setiap proses yang ada pada level tersebut, apakah sudah merupakan

proses yang melakukan satu fungsi saja atau masih melakukan beberapa fungsi yang berbeda. Jika proses yang dianalisis masih melakukan lebih dari satu fungsi yang berbeda, maka perlu dilakukan pemecahan terhadap proses tersebut menjadi level lebih tinggi. Demikian seterusnya sampai didapat semua proses yang ada pada level paling tinggi dilakukan satu fungsi saja.

Jika DFD untuk suatu sistem yang dianalisis telah dibuat, sebelum melanjutkan kegiatan lainnya selalu dilakukan evaluasi atau pemeriksaan terhadap DFD tersebut. Pemeriksaan dilakukan terhadap munculnya kesalahan yang dapat diakibatkan oleh salah gambar, tidak konsistennya penggunaan simbol dan analisis terhadap proses yang terjadi.

2.1.6.3 Perancangan sistem

Perancangan sistem adalah suatu fase kreatif dalam siklus pengembangan sistem, dimana sistem yang dirancang harus memenuhi tujuan sistem secara teknis dan operasional. Pada tahap ini perlu dikembangkan spesifikasi untuk keluaran, database, operasi pemrosesan data, masukkan data, pengendalian serta pengamanan.

Perancangan sistem informasi yang dilakukan, bertujuan untuk menghilangkan beberapa kelemahan sistem dan meningkatkan kemampuan sistem dalam menyelesaikan permasalahan yang ada dengan acuan hasil-hasil analisis yang telah dilakukan.

2.1.6.4 Basis data

Pengertian basis data

Basis data adalah kumpulan dari file atau tabel atau arsip yang saling berhubungan disimpan dalam media penyimpanan elektronik (disket atau hardisk). Prinsip utamanya

adalah pengaturan data atau arsip dan tujuan utamanya adalah kemudahan dan kecepatan dalam mengambil kembali data atau arsip.[1]

Sistem pengelola basis data (*Database Management System*/DBMS)

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus atau spesifik. Perangkat lunak ini disebut DBMS, dimana akan menentukan bagaimana data diorganisasi, disimpan, diubah, dan diambil kembali. DBMS juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, keakuratan data dan sebagainya.

Perangkat lunak yang termasuk DBMS seperti Dbase III+, dBase IV, FoxBase, MsAccess, Borland Paradox, Ms SQL Server, Oracle Borland Interbase. Salah satu tujuan DBMS adalah untuk menyediakan fasilitas antarmuka atau interface dalam melihat data (yang lebih ramah atau *user friendly*) pada pemakai.

Bahasa basis data

DBMS merupakan perantara bagi pemakai dengan basis data dalam disk. Cara berinteraksi atau berkomunikasi anatara pemakai dengan basis data tersebut diatur dalam suatu bahasa khusus yang ditetapkan oleh perusahaan pembuat DBMS. Bahasa itu dapat disebut sebagai bahasa basis data yang terdiri atas sejumlah perintah yang diformulasikan dan dapat diberikan ke user dan dikenali atau diproses oleh DBMS untuk melakukan suatu arsip atau pekerjaan tertentu.

Sebuah bahasa basis data dapat dipalah ke dalam 2 bentuk :

a. *Data definition language* (DDL)

DDL adalah struktur atau skema basis data yang menggambarkan desain basis data secara keseluruhan, dispesifikasikan dengan bahasa khusus. Dengan bahasa inilah dapat dibuat tabel baru, membuat index, mengubah tabel, menentukan struktur penyimpanan tabel, dan sebagainya. Hasil dari kompilasi perintah DLL adalah kumpulan tabel yang disimpan dalam file khusus yang disebut kamus data (Data dictionary).

b. *Data manipulation language* (DML)

Sedangkan DML merupakan bentuk bahasa basis data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu data. Manipulasi data dapat berupa :

1. Penyisipan atau penambahan data baru ke suatu basis data.
2. Penghapusan data dari suatu basis data
3. Pengubahan data dari suatu basis data

DML merupakan bahasa yang bertujuan memudahakan pemakai, untuk mengakses data sebagai mana direpresentasikan oleh model baru.

Diagram E-R (*Entity Relationship Diagram*)

ERD Merupakan notasi grafis dalam pemodelan data konseptual, yang mendeskripsikan hubungan antara penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini relatif kompleks. Dengan ERD, model dapat diuji dengan mengabaikan proses yang harus dilakukan. ERD menggunakan

sejumlag notasi dan simbol untuk menggambarkan struktur dan hubungan antar data, pada dasarnya ada 3 macam simbol yang digunakan :

a. *Entity*

Adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat.

b. Atribut

Entity mempunyai elemen yang disebut atribut, dan berfungsi mendeskripsikan karakter entity.

c. Hubungan

Relationship sebagai mana Entity maka dalam hubungan pun harus dibedakan dalam hubungan atau bentuk hubungan antar Entity dengan isi dari hubungan itu sendiri. Relasi antara 2 file atau 2 tabel dapat dikategorikan menjadi 3 macam, yaitu :

1. *One to One*

Yang berarti entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas B, dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.

2. *One to Many Realtionship*

Yang berarti entitas pada himpunan entitas A berhubungan dengan banyak entitas pada satu himpunan entitas B, tetapi tidak sebaliknya setiap entitas pada himpunan entitas B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.

3. *Many to Many Relationship*

   Yang berarti entitas pada himpunan entitas A berhubungan dengan banyak entitas pada banyak himpunan entitas B dan begitu juga sebaliknya setiap entitas pada himpunan entitas B berhubungan dengan banyak entitas pada himpunan A.

Kamus data (*Data dictionary*)

Kamus data merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data dibuat untuk dapat digunakan pada tahap analisis sistem dan saling berhubungan pada tahap perancangan sistem yang didasarkan pada aliran data di DFD.

Pada tahap analisis dan perancangan sistem, kamus data dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem untuk dilakukannya proses perancangan sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan informasi yang dihubungkan oleh pemakai sistem.

2.2 Perangkat Lunak Penunjang

2.2.1 PHP (*PHP Hypertext Preprocessor*)

PHP adalah merupakan bahasa pemrograman yang digunakan untuk membuat halaman web dinamis. Penulisan *script* PHP diawali dengan tag <? dan diakhiri dengan tag ?>, dan setiap baris statement /perintah harus diakhiri dengan menggunakan tanda titik koma (;), umumnya setiap statement dituliskan dalam satu baris serta script PHP menerapkan aturan *case sensitive* yakni ada perbedaan makna antara huruf besar dengan huruf kecil.

Ada dua cara penulisan script PHP yaitu *Embedded Script* dan *Non Embedded Script*. Cara yang pertama *Embedded Script* merupakan penyisipan *script* PHP diantara tag-tag HTML, contoh penulisannya adalah sebagai berikut :

```
<html>
<head>
<title>contoh</title>
</head>
<body>
<?
     echo”Hello World ?”;
?>
</body>
</html>
```

Dan cara yang kedua ialah Non Embedded Script, script PHP pada cara ini digunakan sebagai murni pembuatan program dengan PHP, tag HTML yang dihasilkan untuk membuat dokumen merupakan bagian dari script PHP, contoh penulisannya adalah sebagai berikut :

```
<?
   echo”<html>;
   echo”<head>;
   echo”<title>;
   echo”contoh”;
   echo”</title>;
   echo”</head>;
   echo”<body>;
   echo”Hello World
?”;
   echo”</body>;
   echo”</html>;
?>
```

Kemampuan PHP yang paling diandalkan dan signifikan adalah dukungan kepada banyak database. Membuat halaman web yang menggunakan data dari database dengan sangat mudah dapat dilakukan Berikut adalah daftar database yang didukung oleh PHP :

1. Adabas D
2. dBase
3. Empress
4. FilrPro (read only)
5. FrontBase
6. Hyperwave
7. IBM DB2
8. Informix
9. Ingres
10. Interbase
11. MSQL
12. Direct MS SQL
13. MySQL
14. ODBC
15. Oracle (OC17 dan OC18)
16. Ovrimos
17. PostgreSQL
18. Solid
19. SQLite
20. Sybase
21. Velocis
22. Unix DBM

Penggunaan PHP yang semakin popular dan terus berkembang tidak lepas dari keunggulan yang dimilikinya, yaitu sebagai berikut :

a. **Cepat** , karena dapat ditempelkan (embedded) di dalam kode HTML.
b. **Tidak mahal/gratis**. Pada kenyataannya PHP merupakan program yang gratis dan mudah untuk mendapatnya dengan cara mendownload dan tanpa harus membayarnya.
c. **Mudah untuk digunakan**. PHP berisi beberapa fitur khusus dan fungsi yang dibutuhkan untuk membuat halaman web dinamis.
d. **Berjalan pada beberapa sistem operasi**. PHP berjalan pada system operasi yang beragam seperti Windows, Linux, Mac OS, dan kebanyakan variasi dari Unix.
e. **Dukungan teknis tersedia secara luas**, karena PHP menyediakan dukungan secara gratis via daftar diskusi e-mail.
f. **Aman**. Pengguna tidak melihat kode PHP, karena kode yang ditampilkan pada browser adalah kode HTML
g. **Dirancang untuk mendukung database**. PHP memiliki kemampuan yang dirancang untuk berinteraksi dengan database tertentu.
h. **Customizable**. Lisensi open source sehingga mengijinkan para pemrogram untuk memodifikasi software PHP, menambahkan atau memodifikasi fitur-fitur yang dibutuhkan untuk lingkungan mereka sendiri.

2.2.2 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai

perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.[4]

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali

lebih cepat dibandingkan *Interbase*. Selain itu MySQL juga memiliki beberapa keistimewaan, antara lain :

1. *Portability*

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

2. *Open Source*

   MySQL didistribusikan secara *open source* (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. *Multiuser*

   MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance tuning*

   MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. *Column types*

   MySQL memiliki tipe kolom yang sangat kompleks, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

6. *Command dan functions*

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam *query*.

7. *Security*

   MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password terenkripsi.

8. *Scalability dan limits*

MySQL mampu menangani *database* dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. *Connectivity*

MySQL dapat melakukan koneksi dengan *client* menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).

10. *Localisation*

MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

11. *Interface*

MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. *Clients dan tools*

MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertakan petunjuk online.

13. Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan database lainnya semacam *PostgreSQL* ataupun *Oracle*.