**BAB II**

**TINJAUAN PUSTAKA**

* 1. **Tinjauan Studi**

Penelitian tentang pemilihan atau penentuan telah banyak dilakukan oleh penulis terdahulu antara lain:

1. Sulpan Hery Siregar, 2013, Sistem Penunjang Keputusan Untuk Menentukan Pelanggan Terbaik Menggunakan Metode *Fuzzy* Mamdani. Penghargaan yang diberikan terhadap konsumen akan membuat konsumen sulit untuk mengalihkan pandangannya ke perusahaan lain yang sama-sama memasarkan produk yang sama, tentunya selain itu perusahaan juga harus menjaga kualitas produk yang mereka pasarkan. Dalam penelitian ini menggunakan variabel lama berlangganan dengan tiga himpunan *fuzzy* (Ringan, Normal dan Berat), variabel kuantitas belanja dengan tiga himpunan *fuzzy* (Rendah, Sedang dan Tinggi), variabel volume belanja dengan tiga himpunan *fuzzy* (Rendah, Normal dan Tinggi), variabel disiplin membayar dengan tiga himpunan *fuzzy* (Rendah, Normal dan Tinggi). Bahasa pemograman yang digunakan adalah *Visual Basic* 6.0 dengan basis data *MySQL*, Output yang dihasilkan berupa nilai tertinggi terhadap salah satu pelanggan.
2. Akrim Tegu Suseno, 2014, Penerapan Metode *Fuzzy* Mamdani Untuk Sistem Pendukung Keputusan Penerimaan Karyawan PT. Primatexmaco Indonesia. Pada penelitian ini bertujuan untuk penerimaan karyawan dengan penggunaan metode *fuzzy* mamdani, adapun variabel yang digunakan adalah variabel tes nilai psikotest, nilai test fisik, nilai test wawancara, dan nilai test kesehatan. Adapun himpunan *fuzzy* yang digunakan pada setiap variabel *fuzzy* adalah: Rendah (D), Cukup (C), Baik (B) untuk variabel output dan input. Pengujian dilakukan terhadap 30 data pelamar dengan metode *fuzzy* mamdani mendapatkan titik error sebanyak 0-5% dan aplikasi ini dibangun dalam bahasa pemograman Matlab 7.1.
3. Rabiyatul Adawiah dkk, 2013, dengan judul penelitian Sistem Pendukung Keputusan Pemilihan Penerima Beasiswa Berbasis *Fuzzy* Mamdani, Pada penelitian ini telah dibangun sistem penentuan penerima beasiswa bagi mahasiswa STMIK Banjarbaru, guna memberikan rekomendasi diterima atau tidaknya pengajuan beasiswa oleh seorang mahasiswa. Pemasalahan dari penelitian ini yaitu bagaimana membangun sistem penentuan mahasiswa penerima beasiswa dengan mengimplementasikan Logika *Fuzzy* Metode Mamdani. Sistem dibangun menggunakan pendekatan terstruktur dengan *Use Case*, *Sequence*, dan *Activity Diagram* sebagai alat bantu perancangan sisitemnya. Hasil penelitian ini berupa sistem pendukung keputusan yang merekomendasikan maupun tidak merekomendasikan mahasiswa penerima beasiswa yang di urutkan berdasarkan nilai rekomendasi tertinggi.
   1. **Tinjauan Pustaka**
      1. **Definisi Persalinan**

Proses persalinan dan kelahiran merupakan suatu proses yang alami dan hampir dialami oleh setiap wanita. Kodratnya wanita dapat melahirkan secara normal. Persalinan normal adalah persalinan melalui vagina atau jalan lahir biasa (Siswosuhardjo dan Chakrawati, 2013). Apabila wanita tidak dapat melahirkan secara normal maka tenaga medis akan melakukan penatalaksanaan persalinan alternative untuk membantu kelahiran janin (Bopak, Lowdermilk and Jensen 2013).

Persalinan Normal adalah proses pengeluaran janin yang terjadi pada kehamilan cukup bulan (37–42 minggu), lahir normal dengan persentasi belakang kepala yang berlangsung dalam 18 jam, tampa komplikasi baik ibu maupun pada janin (Prawiroharjo, 2013.)

Selanjutnya didefinisikan oleh Siswosuhardjo dan Chakrawati (2013) “Proses persalinan dan kelahiran merupakan suatu proses yang alami dan hampir dialami oleh setiap wanita. Kodratnya wanita dapat melahirkan secara normal. Persalinan normal adalah persalinan melalui vagina atau jalan lahir biasa”.

Menurut Kementerian Kesehatan RI, (2013) mendefinisikan tentang persalinan yaitu “Persalinan merupakan suatu proses alami yang ditandai oleh terbukanya *serviks*, diikuti dengan lahirnya bayi dan plasenta melalui jalan lahir. Penolong persalinan perlu memantau keadaan ibu dan janin untuk mewaspadai secara dini terjadinya komplikasi. Di samping itu, penolong persalinan juga berkewajiban untuk memberikan dukungan moril dan rasa nyaman kepada ibu yang sedang bersalin”.

Selanjutnya Kementerian Kesehatan RI, (2013) mengatakan bahwa tujuan dari persalinan normal adalah menjaga kelangsungan hidup dan memberikan derajat kesehatan yang tinggi bagi ibu dan bayinya, melalui upaya yang terintegrasi dan lengkap tetapi dengan intervensi yang seminimal mungkin agar prinsip keamanan dan kualitas pelayanan dapat terjaga pada tingkat yang diinginkan (optimal). Dengan pendekatan, setiap intervensi yang akan diaplikasikan dalam asuhan persalinan normal harus mempunyai alasan dan bukti ilmiah yang kuat tentang manfaat intervensi tersebut bagi kemajuan dan keberhasilan proses persalinan”.

Sedangkan *Saskio sesarea* adalah kelahiran janin melalui *insisi transabdomen* pada uterus (Bopak, Lowdermilk and Jensen 2013). *Saskio sesarea* dilakukan apabila wanita tidak dapat melahirkan secara normal.

*Document Nasional Center For Healt Statistics* tahun 1992 mencatat bahwa terdapat 14 faktor medis yang dapat dilakukan secara s*askio sesarea*, sebagian besar terkait dengan masalah kesehatan ibu, disamping 15 komplikasi tambahan yang terjadi saat persalinan dan kelahiran (Kaufmann 2014).

* + 1. **Kehamilan**

Masa kehamilan dimulai dari konsepsi sampai lahirnya janin. Lamanya hamil normal adalah 280 hari (40 minggu atau 9 bulan 7 hari) dihitung dari hari pertama haid terakhir. Kehamilan dibagi dalam tiga triwulan yaitu triwulan pertama dimulai dari konsepsi sampai 3 bulan, triwulan ke dua dari bulan ke empat sampai 6 bulan triwulan ketiga dari bulan ke tujuh sampai 9 bulan (Prawirohardjo, 2013).

Kehamilan merupakan proses yang alamiah dan fisiologis, Setiap wanita yang memiliki organ reproduksi sehat, yang telah mengalami menstruasi, melakukan hubungan dengan seorang pria yang organ reproduksinya sehat sangat besar akan mengalami kehamilan (Mandriwati,2014). Kehamilan adalah masa dimana seorang wanita membawa embrio atau fetus didalam tubuhnya dalam kehamilan dapat terjadi banyak gestasi misalnya dalam kasus kembar atau triflet (Wikipidia, 2015).

Kehamilan adalah masa dimana terdapat janin didalam rahim seorang perempuan. Masa kehamilan didahului oleh terjadinya pembuahan yaitu bertemunya sel sperma laki-laki dengan sel telur yang dihasilkan oleh indung telur. Setelah pembuahan, terbentuk kehidupan baru di dalam rahim ibu (Ratna Dewi Pudiastuti, 2014).

* + 1. **Variabel Proses Persalinan Ibu Hamil**

Adapun yang menjadi kriteria untuk membantu prediksi persalinan ibu hamil antara lain:

1. Variabel Umur/Usia : Umur sangat mempengaruhi kesehatan seorang wanita pada saat hamil dan melahirkan, variabel ini terdiri atas dua himpunan *fuzzy* yakni BERESIKO Jika seorang wanita hamil dan melahirkan di usia kurang dari 20 dan diatas 35 tahun. NORMAL jika usia antar 20 – 35 tahun (Kemenkes RI, 2013).

**Tabel 2.1**. Variabel Umur/Usia

|  |  |  |
| --- | --- | --- |
| **No.** | **Himpunan** | **Domain** |
| 1. | Beresiko | < 20 dan > 35 tahun |
| 2. | Normal | 20 – 35 tahun |

1. Variabel Tinggi Badan: Tinggi badan ibu merupakan salah satu indikator penting dalam persalinan. Variabel ini terdiri atas dua himpunan fuzzy yaitu BERESIKO jika tinggi badan ≤ 145cm, dan NORMAL jika > 145 cm (Kemenkes RI, 2013).

**Tabel 2.2**. Variabel Tinggi Badan

|  |  |  |
| --- | --- | --- |
| **No.** | **Himpunan** | **Domain** |
| 1. | Beresiko | ≤ 145cm |
| 2. | Normal | >145cm |

1. Variabel Berat Badan: Berat badan ibu juga menjelang persalinan juga merupakan indikator penting dalam persalinan. Variabel ini terdiri atas dua himpunan fuzzy yaitu NORMAL jika berat badan 45-65 Kg, dan BERAT jika > 65 Kg (Kemenkes RI, 2013).

**Tabel 2.3.** Variabel Berat Badan

|  |  |  |
| --- | --- | --- |
| **No.** | **Himpunan** | **Domain** |
| 1. | Normal | 45-65 Kg |
| 2. | Berat | >65 Kg |

1. Variabel Tekanan Darah: Salah satu penyebab kematian ibu saat persalinan adalah tekanan darah yang tinggi, hal ini akan mengakibatkan eklamsia atau gangguan akibat tekanan darah yang terlalu tinggi. Variabel ini terdiri atas dua himpunan *fuzzy* yaitu BERESIKO jika tekanan darah > 125mm/Hg, dan NORMAL jika < 125mm/Hg (Kemenkes RI, 2013).

**Tabel 2.4.** Variabel Tekanan Darah

|  |  |  |
| --- | --- | --- |
| **No.** | **Himpunan** | **Domain** |
| 1. | Beresiko | > 125mm/Hg |
| 2. | Normal | <125mm/Hg |

1. Variabel Jumlah Hb: variabel ini terdiri atas dua himpunan *fuzzy* yaitu Jumlah Hb (*Hemaglobin*) normal bagi ibu hamil yang akan melahirkan adalah ≥ 11, sebaliknya beresiko jika dibawa 11 Hb (Kemenkes RI, 2013).

**Tabel 2.5.** Variabel Jumlah Hb (*Hemaglobin*)

|  |  |  |
| --- | --- | --- |
| **No.** | **Himpunan** | **Domain** |
| 1. | Beresiko | < 11Hb |
| 2. | Normal | >11Hb |

* 1. **Sistem Pendukung Keputusan**
     1. **Pengambilan Keputusan**

Jenis keputusan menurut Simon dalam buku “*Management Information Systems*” edisi 10 terbagi menjadi 2 yaitu:

1. Keputusan terprogram (*programmed decision*) bersifat repetitif dan rutin, dalam hal prosedur tertentu digunakan untuk menanganinya sehingga keputusan tersebut tidak perlakukan *de novo* (baru) setiap kali terjadi.
2. Keputusan yang tidak terprogram (*nonprogrammed decision*) bersifat baru, tidak terstruktur dan jarang konsekuen. Tidak ada metode yang pasti untuk menangani masalah ini.

Simon menjelaskan konsep keputusan terprogram dan tak terprogram penting karena masing-masing memerlukan teknik yang berbeda.

Tahap-tahap pengambilan keputusan menurut Simon:

1. Kegiatan intelegen, yaitu mengamati lingkungan mencari kondisi-kondisi yang perlu diperbaiki.
2. Kegiatan merancang, yaitu menemukan, mengembangkan, dan menganalisis berbagai alternatif tindakan yang mungkin.
3. Kegiatan memilih, yaitu meilih suatu rangkaian tindakan tertentu dari beberapa yang tersedia.
4. Kegiatan menelaah, yaitu menilai pilihan-pilihan yang lain.
   * 1. **Sistem Pendukung Keputusan (SPK)**

Jogiyanto HM (2005 : 327) mendefinisikan: “Suatu sistem pendukung keputusan (SPK) atau *decision support sytems* (DSS) didefinisikan sebagai suatu sistem informasi untuk membantu manajer level menengah untuk proses pengambilan keputusan setengah terstruktur (*semi structured*) supaya lebih efektif dengan menggunakan model-model analitis dan data yang tersedia”.

Demikian pula didefinisikan oleh penulis lain “Sistem pendukung keputusan merupakan pasangan dari intelektual sumber daya manusia dengan kemampuan dari komputer untuk memperbaiki kualitas dari keputusan, yaitu sistem pendukung keputusan yang terkomputerisasi bagi pembuat keputusan manajemen yang menghadapi masalah semi struktur” (Efraim dkk, dalam buku “Berbagai makalah Sistem Informasi dalam KNSI 2009).

Dari definisi tersebut dapat disimpulkan bahwa sistem pendukung keputusan adalah suatu sistem yang utamanya berbasis komputer untuk membantu para pengambil keputusan untuk memecahkan masalah baik yang bersifat semi terstruktur maupun yang tidak terstruktur melalui suatu model.

Sistem Pendukung Keputusan merupakan suatu penerapan sistem informasi yang ditujukan untuk membantu para pimpinan dalam mengambil keputusan. Hal yang terpenting dari pengertian ini adalah sistem pendukung keputusan merupakan alat pelengkap bagi mereka yang terlibat dalam pengambilan keputusan.

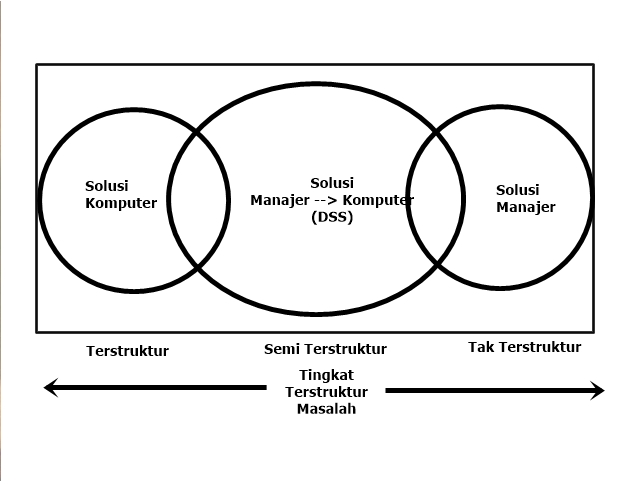
Konsep Sistem Pendukung Keputusan (DSS) dimulai pada akhir tahun 1960-an dengan *timesharing* komputer. Untuk pertama kalinya seseorang dapat berinteraksi langsung dengan komputer tanpa harus melalui spesialis informasi. Baru pada tahun 1971, istilah DSS diciptakan oleh G. Anthony Gorry dan Michael S. Scott Morton, keduanya professor MIT. Mereka merasa perlunya suatu kerangka kerja untuk mengarahkan aplikasi komputer kepada pengambilan keputusan manajemen dan mengembangkan apa yang telah dikenal sebagai Garry & Scott Morton Grid. Matrik (*Grid*) ini didasarkan pada konsep Simon mengenai keputusan terprogram dan tak terprogram serta tingkat-tingkat manajemen Robert N. Anthony.

* + 1. **Tujuan Sistem Pendukung Keputusan**

Dari definisi di atas, maka dapat diketahui tujuan dari SPK adalah sebagai berikut ini:

1. Membantu manajer mengambil keputusan setengah terstruktur yang dihadapi oleh manajer level menengah.
2. Membantu atau mendukung manajemen mengambil keputusan bukan menggantikannya.
3. Meningkatkan efektifitas pengambilan keputusan manajemen.

Tujuan dari SPK ini konsisten dengan yang diutarakan oleh Keen dan Morton (1978). Peter G. Kreen dan Scott Morton adalah pioneer DSS dari MIT menggambarkan sebagai berikut ini. (Jogiyanto. HM, 2010: 328).



**Gambar 2.1.** SPK fokus pada solusi permasalahan *semistructured*

Gambar 2.1 menggambarkan hubungan antara struktur masalah dengan tingkat dukungan yang dapat disediakan oleh komputer. Komputer dapat diterapkan pada bagian masalah yang terstruktur, tetapi manager bertanggung jawab atas bagian yang tak terstruktur menerapkan penilaian atau intuisi dan melakukan analisis. Manajer dan komputer bekerja sama sebagai tim pemecahan masalah yang berada di area semi-terstruktur yang luas.

* 1. **Sisklus Hidup Pengembangan Sistem**

Pengembangan sistem informasi yang berbasis komputer dapat merupakan tugas kompleks yang membutuhkan banyak sumber daya dan dapat memakan waktu berbulan-bulan bahkan bertahun tahun untuk menyelesaikannya. Proses pengembangan sistem melewati beberapa tahapan dari mulai sistem itu direncanakan sampai dengan sistem tersebut diterapkan, dioperasikan dan dipelihara. Bila operasi sistem yang sudah dikembangkan masih timbul kembali permasalahan-permasalahan yang kritis serta tidak dapat diatasi dalam tahap pemeliharaan sistem, maka perlu dikembangkan kembali suatu sistem untuk mengatasinya dan proses ini kembali ke tahap yang pertama, yaitu tahap perencanaan sistem. Siklus ini disebut dengan siklus hidup suatu sistem (*systems life cycle*). Daur atau siklus hidup dari pengembangan sistem merupakan suatu bentuk yang digunakan untuk menggambarkan tahapan utama dan langkah-langkah didalam tahapan tersebut dalam proses pengembangannya.

Siklus hidup pengembangan sistem dengan langkah-langkah utamanya yang akan digunakan adalah sebagai berikut : (Jogiyanto HM, 2010 : 52).

**Kebijakan dan perencanaan sistem**

**Analisis sistem**

**Desain (perancangan) sistem secara umum**

**Seleksi sistem**

**Implementasi ( penerapan) sistem**

**Perawatan sistem**

Awal proyek sistem

Pengembangan sistem

Manajemen sistem

**Gambar 2.2.** Siklus Hidup Pengembangan Sistem

* + 1. **Analisa Sistem**

Jogiyanto HM (2010:129) mendefinisikan analisa sistem sebagai berikut:”Analisa sistem (*systems analysis*) sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya”

Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan di dalam tahap ini akan menyebabkan juga kesalahan ditahap selanjutnya. Tahap analisa sistem mencakup studi kelayakan dan analisis kebutuhan.

Di dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analisis sistem, yaitu sebagai berikut :

1. *Identify,* yaitu mengidentifikasi masalah.

Mengidentifikasi (mengenal) masalah merupakan langkah pertama yang dilakukan dalam tahap analisis sistem. Masalah (*Problems*) dapat didefinisikan sebagai suatu pertanyaan yang diinginkan untuk dipecahkan.

1. *Understand,* yaitu memahami kerja dari sistem yang ada.

Langkah kedua dari tahap analisis sistem adalah memahami kerja dari sistem yang ada. Langkah ini dapat dilakukan dengan mempelajari secara terinci bagaimana sistem yang ada beroperasi. Untuk mempelajari operasi dari sistem ini diperlukan data yang dapat diperoleh dengan cara melakukan penelitian.

1. *Analyze,* yaitu menganalisis sistem tanpa *report*

Langkah ini dilakukan berdasarkan data yang telah diperoleh dari hasil penelitian yang telah dilakukan

1. *Report,* yaitu membuat laporan hasil analisis

Tujuan utama dari pembuatan laporan hasil analisis:

1. Pelaporan bahwa analisis telah selesai dilakukan
2. Meluruskan kesalahan-pengertian mengenai apa yang telah ditemukkan dan dianalisis oleh analisis sistem tetapi tidak sesuai menurut manajemen.
   * 1. **Desain Sistem**

Setelah tahap analisis sistem selesai dilakukan, maka analis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analis sistem untuk memikirkan bagaimana membentuk sistem tersebut. Tahap ini disebut dengan desain sistem (*system design*).

Menurut Robert J.Verzello dan John Reuter, dalam Jogiyanto HM (2005 : 196) desain sistem adalah tahap setelah analisis dari siklus pengembangan sistem; pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi menggambarkan bagaimana suatu sistem dibentuk.

Demikian pula Menurut John Burch dan Gary Grudnitski, dalam Jogiyanto HM (2005:196) desain sistem dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi.

Tahap desain sistem mempunyai dua tujuan utama yaitu :

1. Untuk memenuhi kebutuhan kepada pemakai sistem.
2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknis lainnya.

Desain sistem dapat dibagi dalam dua bagian yaitu desain sistem secara umum (*general systems design*) dan desain sistem secara terinci (*detailed system design*).

1. Desainsistem secara umum(*General System Design*).

Tujuan dari desain sistem secara umum adalah untuk memberikan gambaran secara umum kepada *user* tentang sistem yang baru, yang mana merupakan persiapan dari desain sistem secara rinci. Desain secara umum dilakukan oleh analis sistem untuk mengidentifikasikan komponen-komponen sistem informasi yang akan didesain secara rinci oleh pemrogram komputer dan ahli teknik lainnya.

Pada tahap ini, komponen-komponen sistem informasi dirancang dengan tujuan untuk dikomunikasikan kepada *user*. Komponen sistem informasi yang didesain adalah model, *input*, *database*, *output*, teknologi dan kontrol.

1. Desain sistem secara rinci (*detailed system design*).
2. Desain *input* terinci

Masukan merupakan awal dimulainya proses informasi. Bahan mentah dari informasi adalah data yang terjadi dari transaksi-transaksi yang dilakukanoleh organisasi. Data hasil transaksi merupakan masukan untuk sistem informasi. Hasil dari sistem informasi tidak lepas dari data yang dimasukkan. Desain *input* terinci dimulai dari desain dokumen dasar sebagai penangkap *input* yang pertama kali. Jika dokumen dasar tidak didesain dengan baik, kemungkinan *input* yang tercatat dapat salah bahkan kurang.

Fungsi dokumen dasar dalam penanganan arus data :

* Dapat menunjukkan macam dari data yang harus dikumpulkan.
* Data dapat dicatat dengan jelas, konsisten dan akurat.
* Dapat mendorong lengkapnya data disebabkan data yang dibutuhkan disebutkan satu persatu di dalam dokumen dasarnya.

1. Desain *output* terinci.

Desain *output* terinci dimaksudkan untuk mengetahui bagaimana dan seperti apa bentuk *output*-*output* dari sistem yang baru. Desain *Output* Terinci terbagi atas dua yaitu desain *output* berbentuk laporan dimedia kertas dan desain *output* dalam bentuk dialog pada layar terminal.

1. Desain *Output* Dalam Bentuk laporan

Desain ini dimaksudkan untuk menghasilkan *output* dalam bentuk laporan dimedia kertas. Bentuk laporan yang paling banyak digunakan adalah dalam bentuk tabel dan berbentuk grafik atau bagan.

1. Desain *Output* Dalam Bentuk Dialog Layar Terminal

Desain ini merupakan rancang bangun dari percakapan antara pemakai sistem (*user*) dengan komputer. Percakapan ini dapat terdiri dari proses memasukkan data ke sistem, menampilkan *output* informasi kepada *user* atau keduannya.

Beberapa strategi dalam membuat layar dialog terminal :

* Dialog pertanyaan / jawaban.
* Menu.

Menu banyak digunakan karena merupakan jalur pemakai yang mudah dipahami dan mudah digunakan. Menu berisi beberapa alternatif atau option atau pilihan yang disajikan kepada *user*. Pilihan menu akan lebih baik bila dikelompokkan sesuai fungsinya.

1. Desain *database* terinci.

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan disimpanan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting disistem informasi karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Penerapan *database* dalam sistem informasi disebut *database system*. Sistem basis data (*database system*) 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. Dalam sistem basis data, tiap-tiap orang atau bagian dapat memandang *database* dari beberapa sudut pandang yang bebeda. Pada tahap ini, desain *database* dimaksudkan untuk mendefinisikan isi atau struktur dari tiap-tiap file yang telah diidentifikasikan didesain secara umum.

1. Desain teknologi.

Tahap desain teknologi terbagi atas dua yaitu desain teknologi secara umum dan terinci. Pada tahap ini kita menentukan teknologi yang akan dipergunakan dalam menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknolgi yang dimaksud meliputi :

1. Perangkat Keras (*hardware*), yang terdiri dari alat masukan, alat pemroses, alat *output* dan simpanan luar.
2. Perangkat Lunak (*software*), yang terdiri dari perangkat lunak sistem operasi (*operating system*), perangkat lunak bahasa (*language software*) dan perangkat lunak (*application software*).
3. Sumber Daya Manusia (*brainware*), misalnya operator komputer, pemrogram, spesialis telekomunikasi, sistem analis dan sebagainya.

Desain teknologi sangat diperlukan pada tahap implementasi dan pengujian untuk membuktikan bahwa sistem dapat berjalan secara semestinya.

1. Desain model.

Tahap desain model terbagi menjadi dua yaitu desain model secara umum dan terinci. Tahap desain model secara umum berupa desain sistem secara fisik dan logika. Desain fisik dapat digambarkan dengan bagan alir dokumen.

Desain secara logika digambarkan dengan diagram arus data (DAD). Pada tahap desain model terinci, model akan mendefinisikan secara rinci urutan-urutan langkah dari masing-masing proses yang digambarkan di DAD. Urutan langkah proses ini diwakili oleh suatu program komputer.

**2.4.3 Perancangan Konseptual**

Perancangan konseptual sering kali disebut dengan perancangan logis. Pada perancangan ini kebutuhan pemakai dan pemecahan masalah yang teridentifikasi selama tahap analisis sistem mulai dibuat untuk di implementasikan. Ada tiga langkah penting yang dilakukan dalam perancangan konseptual, yaitu evaluasi alternatif rancangan, penyiapan spesifikasi rancangan dan penyiapan laporan rancangan sistem secara konseptual.

Menurut Romney, Seinbart dan Cushing, 1997 dalam Abdul Kadir (2003 : 407) evaluasi yang dilakukan mengandung hal-hal berikut :

1. Bagaimana alternatif-alternatif tersebut memenuhi sasaran sistem dan organisasi dengan baik?
2. Bagaimana alternatif-alternatif tersebut memenuhi kebutuhan pemakai dengan baik?
3. Apakah alternatif-alternatif tersebut layak secara ekonomi?
4. Apa saja keuntungan dan kerugian masing-masing?

Setelah alternatif rancangan dipilih, tahap selanjutnya adalah penyiapan spesifikasi rancangan yang elemen-elemen sebagai berikut :

* 1. Keluaran

Rancangan laporan mencakup frekuensi laporan (harian, mingguan, dsb), isi laporan, bentuk laporan dan laporan cukup ditampilkan pada layar atau perlu dicetak.

* 1. Penyimpan Data

Dalam hal ini, semua data yang diperlukan untuk membentuk laporan ditentukan lebih detail, termasuk ukuran data dan letaknya dalam berkas.

* 1. Masukan

Rancangan masukan meliputi data yang perlu dimasukkan kedalam sistem.

* 1. Prosedur Pemrosesan dan Operasi

Rancangan ini menjelaskan bagaimana data masukan diproses dan disimpan dalam rangka untuk menghasilkan laporan.

Langkah berikutnya adalah menyiapkan laporan rancangan sistem konseptual. Berdasarkan laporan inilah, perancangan sistem secara fisik dibuat.

**2.4.4 Perancangan Fisik**

Pada perancangan ini, rancangan yang masih bersifat konsep diterjemahkan dalam bentuk fisik sehingga terbentuk spesifikasi lengkap tentang modul sistem dan antarmuka antar modul serta rancangan basis data secara fisik.

Beberapa hasil akhir setelah tahap perancangan fisik berakhir :

1. Rancangan keluaran

Rancangan keluaran berupa bentuk laporan dan rancangan dokumen.

1. Rancangan masukan

Rancangan masukan berupa rancangan layar untuk pemasukan data.

1. Rancangan antarmuka pemakai dan sistem.

Rancangan ini berupa rancangan interaksi antar pemakai dan sistem, misalnya berupa menu, icon dan lain-lain.

1. Rancangan *platform.*

Rancangan ini berupa rancangan yang menentukan *hardware*dan *software*yang akan digunakan.

1. Rancangan basis data.

Rancangan ini berupa rancangan-rancangan berkas dalam basis data termasuk penentuan kapasitas masing-masing.

1. Rancangan modul.

Rancangan ini berupa rancangan program yang dilengkapi dengan algoritma (cara modul / program kerja).

1. Rancangan kontrol.

Rancangan ini berupa rancangan kontrol-kontrol yang digunakan dalam sistem seperti validasi, otorisasi dan audit data.

1. Dokumentasi.

Berupa hasil dokumentasi hingga tahap perancangan fisik.

1. Rencana pengujian.

Berupa rencana yang dipakai untuk menguji sistem.

1. Rencana konversi.

Berupa rencana untuk menerapkan sistem baru terhadap sistem lama.

Bagan Alir sistem merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan alir sistem digambarkan dengan simbol-simbol sebagai berikut :

|  |  |  |  |
| --- | --- | --- | --- |
| Tabel 2.6. Bagan Alir Sistem | | | |
| NO | **NAMA SIMBOL** | **SIMBOL** | KETERANGAN |
| 1. | Simbol Dokumen |  | Menunjukkan dokumen *input* dan *output* baik itu proses manual, mekanik, atau komputer |
| 2. | Simbol Kegiatan Manual |  | Menunjukan pekerjaan manual |
| 3. | Simbol Simpanan Offline |  | Menunjukkan file non-komputer yang diarsip urut angka (*numerical*), huruf (*alphabetical*), atau tanggal (*chronological*) |
| 4. | Simbol Kartu Plong |  | Menunjukkan *input* dan *output* yang menggunakan kartu plong (*punched card*). |
| 5. | Simbol Proses |  | Menunjukkankegiatan proses dari operasi program komputer |
| 6 | Simbol Operasi Luar |  | Menunjukkan operasi yang dilakukan di luar proses operasi komputer |
| 7. | Simbol Pengurutan Offline |  | Menunjukkan proses urut data di luar proses komputer. operasi luar, menunjukkan operasi yang dilakukan di luar proses operasi komputer |
| 8. | Simbol Pita Magnetik |  | Menunjukkan *input* dan *output* menggunakan pita *magnetic*. |
| 9. | Simbol Hard Disk |  | Menunjukkan *input* dan *output*  menggunakan *harddisk* |
| NO | **NAMA SIMBOL** | **SIMBOL** | KETERANGAN |
| 10. | Simbol Diskette |  | Menunjukkan *input* dan *output* menggunakan *diskette* |
| 11. | Simbol Drum Magnetik |  | Menunjukkan *input* dan *output* menggunakan drum magnetic |
| 12. | Simbol Pita Kertas Berlubang |  | Menunjukkan *input* dan *output* menggunakan pita kertas berlubang. |
| 13. | Simbol Keyboard |  | Menunjukkan *input* yang menggunakan *on-line keyboard* |
| 14. | Simbol Display |  | Menunjukkan *output* yang ditampilkan di monitor. |
| 15. | Simbol Pita Kontrol |  | Menunjukkan penggunaan pita kontrol (*control tape*) dalam *batch control* total untuk pencocokan di proses *batch processing*. |
| 16 | Simbol Hubungan Komunikasi |  | Menunjukkan proses transmisi data melalui *channel* komunikasi. |
| 17. | Simbol Garis Alir |  | Menunjukkan arus dari proses |
| 18. | Simbol Penjelasan |  | Menunjukkan penjelasan dari suatu proses |
| 19. | Simbol Penghubung |  | Menunjukkan penghubung ke halaman yang masih sama atau ke halaman yang lain |

Sumber : (Jogiyanto HM, 2005 : 796-799)

Untuk mempermudah penggambaran suatu sistem yang ada atau sistem yang baru yang akan dikembangkan secara logika tanpa memperhatikan lingkungan fisik di mana data tersebut mengalir atau lingkungan fisik di mana data tersebut akan disimpan, maka digunakan Diagram Arus Data (DAD) atau *Data Flow Diagram* (DFD). Dalam menggambarkan sistem perlu dilakukan pembentukan simbol, berikut ini simbol-simbol yang sering digunakan dalam DAD :

1. *External entity* (kesatuan luar) atau *boundary* (batas sistem).

Setiap sistem pasti mempunyai batas sistem *(boundary)* yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya. Kesatuan luar *(external entity)* merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain yang berada di lingkungan luarnya yang akan memberikan *input* serta menerima *output* dari sistem. (Jogiyanto HM, 2005 ;701)

**Gambar 2.3** Notasi Kesatuan Luar

1. *Data flow* (arus data).

Arus data ini menunjukkan arus atau aliran data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem. (Jogiyanto HM, 2005 ;702)

Nama Arus Data

**Gambar 2.4** Notasi Arus Data

1. *Process* (proses).

Suatu proses adalah kegiatan atau kerja yang dilakukan orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. (Jogiyanto HM, 2005 ;705)



**Gambar 2.5** Notasi Proses

1. *Data store* (simpanan data).

Simpanan data pada DFD dapat disimbolkan dengan sepasang garis horisontal paralel yang tertutup disalah satu ujungnya. (Jogiyanto HM, 2005 ;707)

Media Nama Data store

**Gambar 2.6** Notasi Simpanan Data

* + 1. **Implementasi Sistem**

Sistem telah dianalisa dan didesain secara rinci dan teknologi telah diseleksi dan dipilih. Tiba saatnya sekarang sistem untuk diimplementasikan (diterapkan). Tahap implementasi sistem merupakan tahap meletakkan sistem supaya siap untuk dioperasikan. Tahap implementasi sistem dapat terdiri dari langkah-langkah sebagai berikut:

1. Menerapkan rencana implementasi.

Rencana implementasi merupakan kegiatan awal dari tahap implementasi sistem. Rencana implementasi dimaksudkan terutama untuk mengatur biaya dan waktu yang dibutuhkan selama tahap implementasi.

1. Melakukan Kegiatan Implementasi

Kegiatan implementasi dilakukan dengan dasar kegiatan yang telah direncanakan dalam rencana implementasi. Kegiatan-kegiatan yang dapat dilakukan dalam tahap implementasi ini adalah sebagai berikut :

* 1. Persiapan tempat dan instalasi perangkat keras dan perangkat lunak.

Jika peralatan baru akan dimiliki, maka tempat atau ruangan untuk peralatan ini perlu dipersiapkan terlebih dahulu. Keamanan fisik dari tempat ini perlu juga dipertimbangkan. Sistem komputer yang besar membutuhkan tempat dengan lingkungan yang lebih harus diperhitungkan. Langkah selanjutnya setelah persiapan fisik tempat adalah menginstalasi perangkat keras yang sudah dikirim dan menginstalasi perangkat lunak yang sudah ada.

* 1. Pemrograman dan pengetesan sistem.

Pemrograman merupakan kegiatan menulis kode program yang akan dieksekusi oleh komputer. Kode program yang ditulis oleh pemrogram harus berdasarkan dokumentasi yang disediakan oleh analis sistem hasil dari desain sistem secara rinci. Sebelum program diterapkan, maka program harus terlebih dahulu bebas dari kesalahan-kesalahan. Oleh sebab itu, program harus diuji untuk menemukan kesalahan-kesalahan yang mungkin dapat terjadi. Program dites untuk tiap-tiap modul dan dilanjutkan dengan pengetesan untuk semua modul yang telah dirangkai.

* 1. Pengetesan sistem.

Pengetesan sistem biasanya dilakukan setelah pengetesan program. Pengetesan sistem dilakukan untuk memeriksa kekompakan antar komponen sistem yang diimplementasikan. Tujuan utama dari pengetesan sistem ini adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan.

* + 1. **Operasi dan Pemeliharaan**

Setelah masa sistem berjalan sepenuhnya menggantikan sistem lama, sistem memasuki pada tahapan operasi dan pemeliharaan. Zwass (2005) membagi pemeliharaan perangkat lunak menjadi tiga macam,yaitu :

1. Pemeliharaan perfektif.

Pemeliharaan Perfektifditujukan untuk memperbarui sistem lama sebagai tanggapan atas perubahan kebutuhan pemakai dan kebutuhan organisasi, meningkatkan efesiensi sistem, dan memperbaiki dokumentasi.

1. Pemeliharaan adaptif.

Pemeliharaan Adaptif berupa perubahan aplikasi untuk menyesuaikan diri terhadap lingkungan perangkat keras dan perangkat lunakbaru. Sebagai contoh pemeliharaan ini dapat berupa perubahan aplikasi dari *mainframe* ke lingkungan *client/ server*atau mengkonversi dari sistem berbasis berkas ke lingkungan basis data.

1. Pemeliharaan korektif.

Pemeliharaan korektif berpa pembetulan atas kesalahan-kesalahan yang ditemukan pada saat sistem berjalan.

* 1. **Teknik Pengujian Sistem**
     1. ***White Box***

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain dan pengkodean.

Pengujian sistem / perangkat lunak memiliki sejumlah aturan yang berfungsi sebagai sasaran pengujian, diantaranya adalah sebagai berikut :

1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan.
2. *Test case* yang baik adalah *test case* yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya.
3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya.

Pengujian *White Box*, adalah metode pengujian yang menggunakan struktur kontrol desain prosedur untuk memperoleh *test case*. Dengan menggunakan metode *white box*, perekayasa sistem dapat melakukan *test case* yang memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali, menggunakan semua keputusan logis pada sisi *true* dan *false*, mengeksekusi semua *loop* pada batasan mereka dan pada batas operasional mereka, dan menggunakan stuktur data internal untuk menjamin validitasnya. Pengujian *Basis Path* adalah teknik pengujian *white box* yang diusulkan pertama kali oleh Tom McCabe. Metode *basis path* ini memungkinkan desainer *test case* mengukur kompleksitas logis dari desain prosedural dan menggunakannya sebagai pedoman untuk menetapkan basis set dari jalur eksekusi. (Roger S. Pressman, 2002 : 536).

**11**

**9**

**10**

**5**

**4**

**1**

**2**

**3**

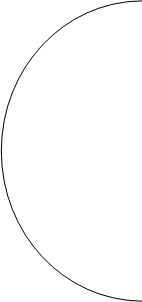
**6**

**7**

**8**

**Gambar 2.7.** Bagan Alir

Bagan alir digunakan untuk menggambarkan struktur kontrol program. Dan untuk menggambarkan grafik alir, harus memperhatikan representasi desain prosedural pada bagan alir. Pada gambar dibawah ini, grafik alir memetakan bagan alir tersebut ke dalam grafik alir yang sesuai (dengan mengasumsikan bahwa tidak ada kondisi senyawa yang diisikan di dalam diamond keputusan dari bagan alir tersebut). Masing-masing lingkaran, yang disebut *simpul* grafik alir, merepresentasikan satu atau lebih statemen prosedural. Urutan kotak proses dan permata keputusan dapat memetakan simpul tunggal. Anak panah tersebut yang disebut *edges* atau *links*, merepresentasikan aliran kontrol dan analog dengan anak panah bagan alir. *Edge* harus berhenti pada suatu simpul, meskipun bila simpul tersebut tidak merepresentasikan statemen prosedural. (Roger S. Pressman, 2002 : 536).

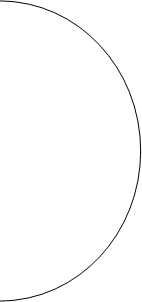


*Edge*

Simpul

Predikat

*Node*



Region

R1

R4

R2

R3

**Gambar 2.8** Grafik Alir

* *Node* adalah lingkaran yang merepresentasikan satu atau lebih statemen prosedural.
* *Edge* adalah anak panah pada grafik alir.
* *Region* adalah area yang membatasi *edge* dan *node*
* Simpul Predikat adalah simpul atau *node* yang berisi kondisi yang ditandai dengan dua atau lebih *edge* yang berasal darinya.

Dari gambar *flowgraph* di atas didapat :

*Path* 1 =1– 11

*Path* 2 =1– 2 – 3 – 4 – 5 – 10– 1–11

*Path* 3 =1– 2 – 3 – 6 – 8 – 9 – 10– 1 – 11

*Path* 4 =1– 2 – 3 – 6 – 7 – 9–10–1–11

*Path* 1,2,3,4 yang telah didefinisikan diatas merupakan *basis set* untuk diagram alir. *Cyclomatic complexity* digunakan untuk mencari jumlah *path* dalam satu *flowgraph*. Dapat dipergunakan rumusan sebagaiberikut :

* 1. Jumlah region grafik alir sesuai dengan *cyclomatic complexity*.
  2. *Cyclomatix complexity*V(G) untuk grafik alir dihitung dengan rumus:

***V(G) =E– N +2*** …………………. (1)

Dimana:

E= jumlah *edge* pada grafik alir

N= jumlah *node* pada grafik alir

Jalur 3 : 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Jalur 4 : 1 – 2 – 3 – 6 – 7 – 9 – 10 – 1 – 11

Jalur 1, 2, 3, dan 4 yang ditentukan di atas terdiri dari sebuah basis set untuk grafik alir pada gambar 2.8 Bagaimana kita tahu banyaknya jalur yang dicari? Komputasi kompleksitas siklomatis memberikan jawaban. Fondasi kompleksitas siklomatis adalah teori grafik, dan memberi kita metriks perangkat lunak yang sangat berguna. Kompleksitas dihitung dalam salah satu dari tiga cara berikut :

1. Jumlah region grafik alir sesuai dengan kompleksitas siklomatis.
2. Kompleksitas siklomatis, *V(G),* untuk grafik alir *G* ditentukan sebagai

*V(G)* = *E – N* + 2 di mana *E* adalah jumlah *edge* grafik alir dan *N* adalah jumlah simpul grafik alir.

1. Kompleksitas siklomatis, *V(G),* untuk grafik alir G juga ditentukan sebagai *V(G)* = P + 1, dimana P adalah jumlah simpul predikat yang diisikan dalam grafik alir *G*.

Pada gambar 2.7 grafik alir, kompleksitas siklomatis dapat dihitung dengan menggunakan masing-masing dari algoritma yang ditulis di atas :

1. Grafik alir mempunyai 4 region
2. *V(G)* = 11 *edge* – 9 simpul + 2 = 4
3. *V(G)* = 3 simpul yang diperkirakan + 1 =4

Dengan demikian, kompleksitas siklomatis dari grafik alir pada gambar 2.8 adalah 4. Yang lebih penting, nilai untuk *V(G)* memberi kita batas atas untuk jumlah jalur independen yang membentuk basis set, dan implikasinya, batas atas jumlah pengujian yang harus didesain dan dieksekusi untuk menjamin semua statemen program.

* + 1. ***Black Box***

Pengujian *Black-Box* berusaha menemukan kesalahan dalam kategori :

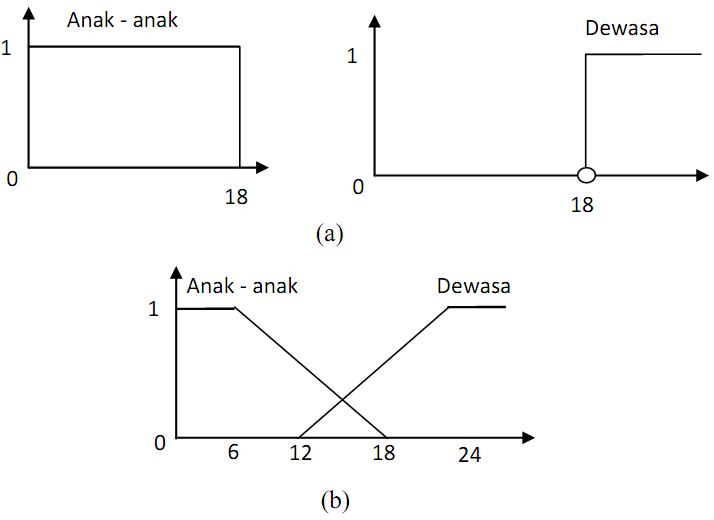
* Fungsi tidak benar atau hilang
* Kesalahan antar muka
* Kesalahan pada struktur data (pengaksesan basis data)
* Kesalahan inisialisasi dan akhir program
* Kesalahan performasi.

Pengujian ini berfokus pada persyaratan fungsional perangkat lunak dan merupakan komplemen dari pengujian *White-Box*. Hal tersebut dapat dicapai melalui :

1. Pengujian *Graph-based* : dimulai dengan membuat grafik sekumpulan *node* yang mempresentasikan objek (misal *New File*, Layar baru dengan atributnya), link (hubungan antar objek), *node-weight* (misal nilai data tertentu seperti atribut layar, perilaku), dan link-weight (karakteristik suatu link, misal menu select).
2. *Equivalence Partitioning* : membagi domain *input* untuk pengujian agar diperoleh kelas-kelas kesalahan (misal kelompok data karakter, atau atribut yang lain).
3. Analisis Nilai Batas : pengujian berdasarkan nilai batas domain *input*.
4. Pengujian Perbandingan : disebut juga pengujian *back-to-back* yang diterapkan pada pada suatu versi perangkat lunak atau perangkat lunak redundan untuk memastikan konsistensinya.
   1. ***Logika Fuzzy***

Konsep logika *fuzzy* pertama kali diperkenalkan oleh Professor Lotfi A.Zadeh dari Universitas California, pada bulan Juni 1965. *Fuzzy* secara bahasa diartikan sebagai kabur atau samar – samar. Menurut Setiadji (2009 : 174), *fuzzy* merupakan suatu nilai yang dapat bernilai benar atau salah secara bersamaan. Namun seberapa besar nilai kebenaran dan kesalahannya tergantung pada derajat keanggotaan yang dimilikinya. Derajat keanggotaan dalam *fuzzy* memiliki rentang nilai 0 (nol) hingga 1(satu). Hal ini berbeda dengan himpunan tegas yang memiliki nilai 1 atau 0 (ya atau tidak). Logika *fuzzy* digunakan untuk menterjemahkan suatu besaran yang diekspresikan menggunakan bahasa (linguistik), misalkan besaran kecepatan laju kendaraan yang diekspresikan dengan pelan, agak cepat, cepat, dan sangat cepat. Dan logika *fuzzy* menunjukkan sejauh mana suatu nilai itu benar dan sejauh mana suatu nilai itu salah. Tidak seperti logika tegas, suatu nilai hanya mempunyai 2 kemungkinan yaitu merupakan suatu anggota himpunan atau tidak. Derajat keanggotaan 0 (nol) artinya nilai bukan merupakan anggota himpunan dan 1 (satu) berarti nilai tersebut adalah anggota himpunan.

Dalam contoh kehidupan seseorang dikatakan dewasa apabila berumur lebih dari 18 tahun, maka seseorang yang kurang dari atau sama dengan 18 tahun di dalam logika tegas akan dikatakan sebagai tidak dewasa atau anak – anak. Sedangkan dalam hal ini pada logika *fuzzy*, seseorang yang berumur sama dengan atau kurang dari 18 tahun dapat dikategorikan dewasa tetapi tidak penuh. Secara grafik dapat digambarkan sebagai berikut:



**Gambar 2.9.** Perbandingan contoh (a) logika tegas dan (b) logika *fuzzy* dalam penentuan golongan umur

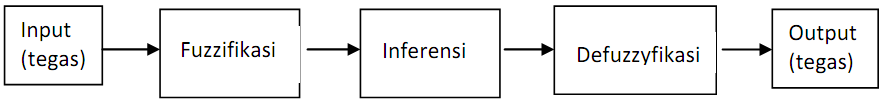
Menurut Kusumadewi (2004 : 2), logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang input kedalam suatu ruang output. *Fuzzy* dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran.

* + 1. **Sistem Inferensi**

Salah satu aplikasi logika *fuzzy* yang telah berkembang amat luas dewasa ini adalah sistem *inferensi fuzzy (Fuzzy Inference System / FIS*), yaitu kerangka komputasi yang didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* berbentuk *IF*-*THEN*, dan penalaran *fuzzy*. Ada tiga metode dalam sistem *inferensi fuzzy* yang sering digunakan, yaitu metode Tsukamoto, metode Mamdani, dan metode Takagi Sugeno

* + 1. **Sistem Berbasis Aturan *Fuzzy***

Pendekatan logika *fuzzy* diimplementasikan dalam tiga tahapan, yakni: *fuzzyfikasi*, evaluasi *rule* (*inferensi*), dan *defuzzifikasi*.



**Gambar 2.10.** Tahapan sistem berbasis aturan *fuzzy*

1. *Fuzzyfikasi*

*Fuzzyfikasi* merupakan fase pertama dari perhitungan *fuzzy*, yaitu mengubah masukan - masukan yang nilai kebenarannya bersifat pasti ke dalam bentuk *fuzzy* input yang berupa tingkat keanggotaan / tingkat kebenaran. Dengan demikian, tahap ini mengambil nilai-nilai *crisp* dan menentukan derajat di mana nilai-nilai tersebut menjadi anggota dari setiap himpunan *fuzzy* yang sesuai.

1. Inferensi

Inferensi adalah melakukan penalaran menggunakan *fuzzy* input dan *fuzzy* *rules* yang telah ditentukan sehingga menghasilkan *fuzzy* output. Secara sintaks, suatu *fuzzy* *rule* (aturan *fuzzy*) dituliskan sebagai berikut:

*IF antecendent THEN consequent*

1. *Defuzzyfikasi*

*Defuzzifikasi* adalah mengubah *fuzzy* output menjadi nilai tegas berdasarkan fungsi keanggotaan yang telah ditentukan. *Defuzzifikasi* merupakan metode yang penting dalam pemodelan sistem *fuzzy*:

Dimana notasi integral melambangkan himpunan semua bersama dengan derajat keanggotaannya pada himpunan *fuzzy* A. Cara ini digunakan pada himpunan fuzzy yang anggotanya bernilai kontinu.

Dimana notasi sigma melambangkan himpunan semua bersama dengan derajat keanggotaannya pada himpunan *fuzzy* A. Cara ini digunakan pada himpunan fuzzy yang anggotanya bernilai diskrit.

Contoh 2.1 Himpunan *Fuzzy* Diskrit

Semesta U adalah himpunan bilangan asli kurang dari 10, dinyatakan dengan U = {1,2,3,4,5,6,7,8,9}. Himpunan *fuzzy* bilangan asli sekitar 5 direpresentasikan seperti gambar 2.11.

1. *Defuzzyfikasi*

*Defuzzifikasi* adalah mengubah *fuzzy* output menjadi nilai tegas berdasarkan fungsi keanggotaan yang telah ditentukan. *Defuzzifikasi* merupakan metode yang penting dalam pemodelan sistem *fuzzy*

* + 1. **Fuzzy Mamdani**

Metode Mamdani sering dikenal dengan nama Metode *Min – Max*. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan output, diperlukan 4 tahapan:

1. Pembentukan himpunan *fuzzy*

Pada Metode Mamdani, baik variabel input maupun variabel output dibagi menjadi satu atau lebih himpunan *fuzzy*.

1. Aplikasi Fungsi Implikasi

Pada metode Mamdani, fungsi implikasi yang digunakan adalah *Min*.

1. Komposisi Aturan

Apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari gabungan antar aturan. Ada tiga metode yang digunakan dalam melakukan *inferensi* sistem *fuzzy*, yaitu: *max, additive* dan *probabilistik OR (probor*).

1. Metode *Max* (*Maximum*)

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah *fuzzy*, dan mengaplikasikannya ke output dengan menggunakan operator OR (*union*). Jika semua proposisi telah dievaluasi, maka output akan berisi suatu himpunan *fuzzy* yang merefleksikan konstribusi dari tiap-tiap proposisi. Secara umum dapat dituliskan:

…(2.2)

Keterangan:

= nilai keanggotaan solusi fuzzy sampai aturan ke-i;

= nilai keanggotaan konsekuen fuzzy aturan ke-i.

1. Metode *Additive* (*Sum*)

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan bounded-sum terhadap semua output daerah *fuzzy*. Secara umum dituliskan:

…...(2.3)

Keterangan:

= nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;

= nilai keanggotaan konsekuen *fuzzy* aturan ke-i.

1. Metode *Probabilistik OR* (*Probor*)

Pada metode ini, solusi himpunan *fuzzy* diperoleh dengan cara melakukan product terhadap semua output daerah *fuzzy*. Secara umum dituliskan:

....(2.4)

Keterangan:

= nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;

= nilai keanggotaan konsekuen *fuzzy* aturan ke-i.

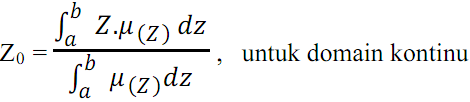
1. Penegasan (*Defuzzyfikasi*)

Input dari proses *defuzzifikasi* adalah suatu himpunan *fuzzy* yang diperoleh dari suatu komposisi aturan – aturan *fuzzy*, sedangkan output yang dihasilkan merupakan suatu bilangan pada himpunan *fuzzy* tersebut. Sehingga jika diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai output.

Menurut Kusumadewi (2004 : 44), ada beberapa metode *defuzzifikasi* pada komposisi aturan Mamdani, antara lain:

1. Metode *Centroid* (*Composite Moment*)

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil titik pusat daerah *fuzzy*. Secara umum dirumuskan:



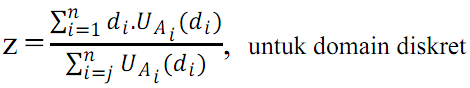
….(2.5)

Keterangan:

Z = nilai domain ke – i,

= derajat keanggotaan titik tersebut,

= nilai hasil penegasan (*defuzzyfikasi*).



…(2.6)

Keterangan:

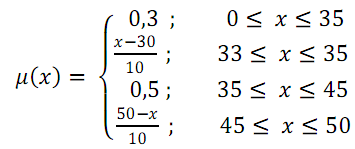
Z = nilai hasil penegasan (*defuzzyfikasi*)

= nilai keluaran pada aturan ke i

= derajat keanggotaan nilai keluaran pada aturan ke –i

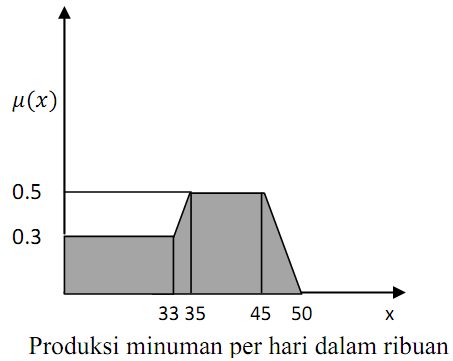
n = banyaknya aturan yang digunakan.

Berikut adalah contoh *defuzzifikasi* dengan metode *centroid* untuk menentukan produksi minuman. Diketahui output dari komposisi aturan pada kasus optimasi produksi minuman adalah sebagai berikut:



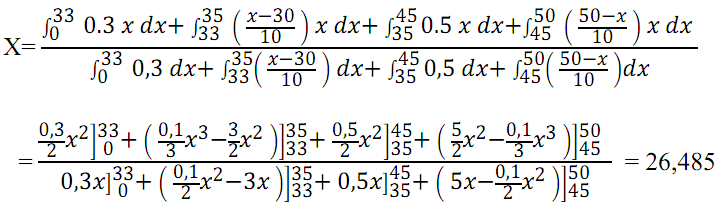
…….(2.7)

Daerah solusi *fuzzy* variabel produksi minuman seperti ditunjukkan pada gambar 2.11.



**Gambar 2.11.** Daerah solusi *fuzzy* variabel produksi minuman

Berdasarkan daerah solusi *fuzzy* tersebut, akan dihitung jumlah produksi minuman setiap hari sebagai keluaran tegas dari proses *defuzzifikasi*.



Jadi jumlah minuman yang harus diproduksi setiap harinya sebanyak 26.485 kemasan.

1. Metode *Bisektor*

Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan setengah dari jumlah total nilai keanggotaan pada daerah *fuzzy*. Secara umum dituliskan:



…..(2.8)

Keterangan :

d = nilai hasil penegasan (*defuzzyfikasi*),

= nilai keluaran pada aturan ke-i,

= derajat keanggotaan nilai keluaran pada aturan ke – i,

n = banyak aturan yang digunakan

1. Metode *Mean Of Maximum* (*MOM*)

Pada metode ini, solusi crisp diperoleh dengan cara mengambil nilai rata– rata *domain* yang memiliki nilai keanggotaan maksimum.

1. Metode *Largest Of Maximum* (*LOM*)

Pada metode ini, solusi crisp diperoleh dengan cara mengambil nilai terbesar dari *domain* yang memiliki nilai keanggotaan maksimum.

1. Metode *Smallest Of Maximum* (*SOM*)

Pada metode ini, solusi crisp diperoleh dengan cara mengambil nilai terkecil dari *domain* yang memiliki nilai keanggotaan maksimum.

* 1. **Contoh Penerapan *Fuzzy* Mamdani**

Andreas Widyantoro (2014) melakukan penelitian tentang penentuan harga kain batik menggunakan *Fuzzy mamdani*. Prosesnya sebagai berikut:

1. Menentukan Himpunan *Fuzzy*
2. Variabel kain dipakai untuk menyatakan harga bahan baku kain sebelum diproses. Variabel kain disini dibagi menjadi tiga sub variabel, sesuai dengan kain yang dipakai oleh produsen batik sesuai dengan tingkatan kualitasnya. Berikut himpunan fuzzy pada kain :

**Tabel 2.7.** Kain Serat Kayu

|  |  |  |
| --- | --- | --- |
| **No.** | **Harga Kain** | **Tingkat Fuzzy** |
| 1 | 10000-12000 | Rendah |
| 2 | 11000-13000 | Sedang |
| 3 | 12000-14000 | Tinggi |

1,4

1,3

1,2

1,5

1

**Gambar 2.12** Himpunan *Fuzzy* Kain Serat Kayu

**Tabel 2.8.** Kain Katun

|  |  |  |
| --- | --- | --- |
| **No.** | **Harga Kain** | **Tingkat Fuzzy** |
| 1 | 9000-10000 | Rendah |
| 2 | 9400-10600 | Sedang |
| 3 | 10000-11000 | Tinggi |

1,1

1,06

1

0,94

0,9

**Gambar 2.13** Himpunan *Fuzzy* Kain Katun

**Tabel 2.9.** Kain Sutra

|  |  |  |
| --- | --- | --- |
| **No.** | **Harga Kain** | **Tingkat Fuzzy** |
| 1 | 10000-12000 | Rendah |
| 2 | 11000-13000 | Sedang |
| 3 | 12000-14000 | Tinggi |

2,29

2,19

2

1,99

1,89

**Gambar 2.14** Himpunan *Fuzzy* Kain Sutra

* 1. Variabel proses pembuatan, digunakan untuk menyatakan macam proses yang digunakan untuk membatik. Kriterianya dibuat berdasarkan upah pengrajin batik. Berikut himpunan *fuzzynya*:

**Tabel 2.10.** Proses Pembuatan

|  |  |  |
| --- | --- | --- |
| **No.** | **Proses Pembuatan** | **Tingkat Fuzzy** |
| 1 | 7500-20000 | Print |
| 2 | 20000-35000 | Cap |
| 3 | 35000-70000 | Tulis |

7

5

2

**Gambar 2.15** Himpunan *Fuzzy* Proses Pembuatan

* 1. Variabel motif, digunakan untuk menyatakan tingkat kerumitan motif batik. Kriterianya dibuat berdasarkan upah pengrajin batik. Berikut ini himpunan fuzzynya:

**Tabel 2.11.** Motif

|  |  |  |
| --- | --- | --- |
| **No.** | **Upah Pengrajin** | **Tingkat Fuzzy** |
| 1 | 25000-32500 | Sederhana |
| 2 | 32500-37500 | Agak Rumit |
| 3 | 37500-50000 | Rumit Sekali |

4,25

3,75

3,25

5

2,5

**Gambar 2.16.** Himpunan *Fuzzy* Motif

* 1. Variabel warna, digunakan untuk menyatakan harga baku pewarna yang dipakai para pengrajin. Kriterianya disesuaikan dengan harga pewarnanya. Berikut himpunan *fuzzynya*:

**Tabel 2.12.** Pewarna

|  |  |  |
| --- | --- | --- |
| **No.** | **Harga Pewarna** | **Tingkat Fuzzy** |
| 1 | 5000-16000 | Sintesis |
| 2 | 40000-200000 | Alam |

2

1,64

0,4

0

**Gambar 2.17.** Himpunan *Fuzzy* Warna

* 1. Variabel Harga, merupakan output yang diharapkan dalam sistem ini. Kriterianya dibuat berdasarkan penelitian yang penulis sudah lakukan, hasil yang diharapkan penulis adalah harga yang lebih murah dibandingkan dengan harga yang diterapkan oleh para pelaku bisnis batik baru. Seringga tidak memberatkan pembeli dan tidak merugikan penjual :

**Tabel 2.13.** Harga

|  |  |  |
| --- | --- | --- |
| **No.** | **Harga Batik** | **Tingkat Fuzzy** |
| 1 | 6500-30000 | Murah |
| 2 | 300500-4000000 | Sedang |
| 3 | 4100000-8000000 | Mahal |

0,9

0,09

6

0,3

**Gambar 2.18.** Himpunan *Fuzzy* Harga

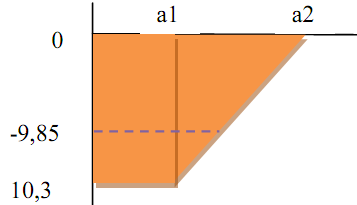
1. Aplikasi Fungsi Implikasi

Setelah aturan dibentuk, maka dilakukan aplikasi fungsi implikasi. Sebagai contoh terdapat sebuah kasus sebagai berikut, berapa jumlah yang harus diproduksi jika:

* Kain serat kayu = 1,15
* Proses = 2
* Warna = 0,6

1. Komposisi Aturan

Dari aturan predikat yang ada, dapat dihasilkan komposisi aturan baru sebagai berikut:



…..(2.9)

1. *Defuzzyfikasi*

Metode *defuzzyfikasi* yang digunakan adalah *centroid*

…..(2.10)

Berdasarkan proses fuzzifikazy yang sudah dilakukan diatas dengan memberikan sampel yang harus dihitung sehingga didapatkan harga jual batik, setelah dilakukan penerapan logika fuzzy akhirnya didapatkan harga jual batik adalah Rp. 154500. Hasilnya masih ada di range harga murah (65000-300000) dan itu berarti didapat sudah sesuai dengan aturan.

* 1. ***Data Management System***

DBMS (*Data Management System*) adalah suatu perangkat lunak yang ditujukan untuk menangani penciptaan, pemeliharaan, dan pengendalian akses data. Dengan menggunakan perangkat lunak ini pengelolaan data menjadi mudah dilakukan. Selain itu perangkat lunak ini juga menyediakan berbagai piranti yang berguna. Misalnya piranti yang memudahkan dalam membuat berbagai bentuk laporan.

* + 1. **Pengertian *Database***

*Database* (basis data) merupakan kumpulan data yang saling berhubungan. Hubungan antar data dapat ditunjukan dengan adanya *field*/ kolom kunci dari tiap file/tabel yang ada. Dalam satu file atau table terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu *record* (umumnya digambarkan sebagai baris data) terdiri dari field yangsaling berhubungan menunjukan bahwa field tersebut dalam satu pengertian yang lengkap dan disimpan dalam satu *record*.

* + 1. **E.R Diagram.**

Model E-R adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas, atribut dan hubungan antar entitas.

**Tabel 2.14.** Simbol‐Simbol ER‐Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NOTASI** | **KET** | **NOTASI** | **KET** |
|  | *Entity*  *Weak Entity*  *Relationship*  *Identifying*  *Relationship*  Atribut |  | Atribut*PrimaryKey*  Atribut  *Multivalue*  Atribut  *Composite*  Atribut  *Derivatif* |

***Entity*** adalah obyek yang dapat dibedakan dalam dunia nyata. *Entity* set adalah kumpulan dari *entity* yang sejenis, *entity* set dapat berupa:

1. Obyek secara fisik: Rumah, Kendaraan, Peralatan
2. Obyek secara konsep : Pekerjaan, Perusahaan, Rencana.

**Atribut** adalah karakteristik dari *entity* atau *relationship*, yang menyediakan penjelasan detail tentang *entity* atau *relationship* tersebut. Nilai Atribut merupakan suatu data actualatau informasi yang disimpan pada suatu atribut didalam suatu *entity* atau *relationship*. Jenis-jenis atribut:

1. *Key*: Atribut yang digunakan untuk menentukan suatu *entity* secara unik.
2. Atribut *Simple*: Atribut yang bernilai tunggal.
3. Atribut *Multivalue*: Atribut yang memiliki sekelompok nilai untuk setiap instan *entity*.
4. Atribut *Composite*: Suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu.
5. Atribut *Derivatif*: Suatu atribut yang dihasilkan dari atribut yang lain.

***Relationship*** adalah hubungan yang terjadi antara satu atau lebih *entity*. *Relationship* seta dalah kumpulan *relationship* yang sejenis, derajat dari *relationship* menjelaskan jumlah *entity* yang berpartisipasi dalam suatu *relationship* :

a. *Unary Degree* (Derajat Satu)

b. *Binary Degree* (DerajatDua)

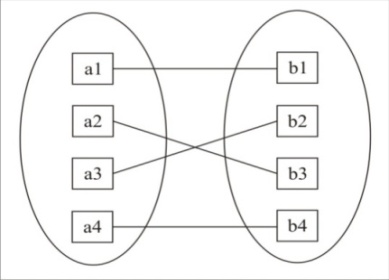
c. *Ternary Degree* (Derajat Tiga)

**Weak *Entity*** adalah suatu *Entity* dimana keberadaan dari *entity* tersebut tergantung dari keberadaan *entity* lain. *Entity* yang merupakan induknya disebut *Identifying Owner* dan *relationship*nya disebut *Identifying Relationship*. Weak *Entity* selalu mempunyai *Total Participation constraint*  dengan *Identifying Owner*.

* + 1. **Hubungan Antar Tabel**

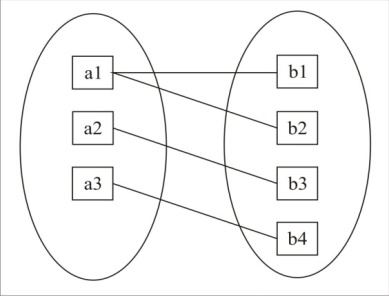
Dalam perancangan Basis Data terdapat hubungan-hubungan yang terjadi antar tabel, hubungan-hubungan antar tabel tersebut adalah:

1. Hubungan *One to One*

**** Hubungan *One to One* merupakan hubungan antara satu tabel induk yang dihubungkan dengan satu tabel anak yang lainnya, yang dihubungkan berdasarkan atribut kunci yang terdapat pada masing-masing tabel.

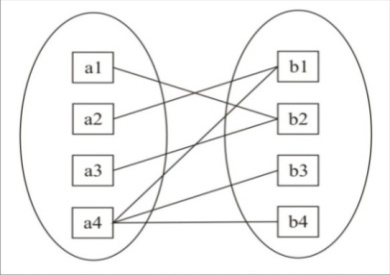
**Gambar 2.19.** Contoh Hubungan *One to One*

1. Hubungan *One to Many*

Hubungan *One to Many* merupakan hubungan dari satu tabel induk yang dihubungkan dengan banyak tabel anak lainnya, dimana hubungan yang terjadi berdasarkan atribut kunci yang ada pada tabel induk.

**Gambar 2.20.** Contoh Hubungan *One to Many*

1. Hubungan *Many to Many*

Hubungan *Many to Many* merupakan hubungan keseluruhan yang berasal dari banyak tabel yang mempunyai hubungan dengan banyak tabel yang lainnya.

**Gambar 2.21.** Contoh Hubungan *Many to Many*

* + 1. **Jenis *Key* (Kunci)**
    2. *Super Key*

*Super Key* adalah satu atribut atau kumpulan atribut yang secara unik mengindentifikasi sebuah tuple atau *record* di dalam relasi atau himpunan dari satu atau lebih entitas yang dapat digunakan untuk mengidentifikasikan secara unik sebuah entitas dalam entitas set.

* + 1. *Candidate Key*

*Candidate Key adalah* satu atribut atau satu set atribut yang mengidentifikasikan secara unik suatu kejadian spesifik dari *entity.* Jika satu*candidate key* berisi lebih dari satu atribut maka disebut sebagai *composite* key (kunci campuran atau kunci gabungan).

* + 1. *Primary Key*

*Primary key* adalah suatu atribut atau satu set atribut yang tidak hanya mengidentifikasikan secara unik suatu kejadian spesifik, tetapi juga dapat mewakili setiap kejadian dari suatu *entity.*

* + 1. *Alternate Key*

*Alternate Key* adalah *candidate key* yang tidak dipakai sebagai *primary key*. *Alternate key* sering dipakai sebagai kunci pengurutan dalam laporan.

* + 1. *Foreign Key*

*Foreign Key* adalah satu atributyang melengkapi satu *relationship* yang menunjukkan ke induknya.

* + 1. **Perangkat Lunak Pendukung**

Berikut ini perangkat lunak pendukung yang digunakan dalam pembuatan aplikasi penentuan harga jual pakaian karawo. PHP merupakan Bahasa pemograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. Sedangkan *MySQL* merupakan software sistem manajemen basis data *SQL* atau *DBMS* yang *multi thread* dan *multi user.* **PHP** dan **MySQL**, seolah pasangan sejati yang tak terpisahkan. Keduanya paling sering disandingkan dalam pembuatan aplikasi berbasis web (web application development). Berikut ini beberapa fungsi PHP yang berhubungan dengan database MySQL:

**Tabel 2.15.** Fungsi PHP yang berhubungan dengan *database* MySQL

|  |  |  |
| --- | --- | --- |
| No. | *Tools* | Keterangan |
| 1. | mysql\_connet | Fungsi ini digunakan untuk mengkoneksikan *script* php dengan *database mysql.* |
| 2. | mysql\_select\_db | Fungsi ini digunakan untuk memilih nama *database* yang akan diakses dengan *script* php. |
| 3. | mysql\_query | Fungsi ini digunakan untuk menjalankan *query* atau perintah sql seperti *select*, *delete*, *update* dan lain-lain. |
| 4. | mysql\_fecth\_array | Fungsi ini digunakan untuk mengeluarkan data-data dari tabel dalam *database* yang dihasilkan dari perintah *mysql\_query* |
| 5. | mysql\_num\_rows | fungsi ini digunakan untuk menghitung jumlah *record* yang dihasilkan dari sebuah perintah *sql.* |
| 6. | mysql\_fetch\_assoc | Fungsi ini menghasilkan *array* *asosiatif* yang mana index-nya sesuai dengan nama kolom yang diseleksi. Mudahnya, tidak tergantung pada urutan kolom, yang penting index-nya sesuai. |

* 1. **Kerangka Pemikiran**

Membangun Sistem Pendukung Keputusan dengan penerapan *Logika Fuzzy* Untuk prediksi Persalinan Pada Rumah Sakit Umum Daerah Prof. Dr. H. Aloei Saboe Kota Gorontalo

***Solusi***



1. Bagaimana merancang suatu penerapan *fuzzy Logic* untuk prediksi persalinan pada Rumah Sakit Umum Daerah Prof. Dr. H. Aloei Saboe Kota Gorontalo?
2. Bagaimana penerapan metode *logika fuzzy* yang diaplikasikan kedalam sistem dapat memberikan solusi pengambilan keputusan dalam menentukan prediksi persalinan pada Rumah Sakit Umum Daerah Prof. Dr. H. Aloei Saboe Kota Gorontalo?

***Masalah***

***Analisis Sistem***



* Sistem Berjalan
* Sistem Yang Diusulkan

***Peluang***

* Pegawai dan Staf Administrasi memiliki pengetahuan yang cukup dalam penggunaan komputer dan aplikasinya
* Dibutuhkannya Sistem Pendukung Keputusan menggunakan logika fuzzy untuk prediksi persalinan.

***Pembangunan Sistem***



* PHP
* *MySQL Server*
* *Dreamweaver*

**Desain *Sistem***



* Desain Model
* Desain *User Interface*
  + Desain *Output*
  + Desain *Input*
  + Desain Menu Utama
* Desain *Database*
* Desain Teknologi
* *White Box*
* *Black Box*

***Pengujian Sistem***



***Implementasi Sistem***



RSUD Prof. Dr. H. Aloei Saboe

Kota Gorontalo

1. Untuk merancang aplikasi sistem pendukung keputusan menggunakan logika *fuzzy* untuk menentukan proses persalinan ibu hamil pada Rumah Sakit Umum Daerah Prof. Dr. H. Aloei Saboe Kota Gorontalo.
2. Untuk menerapkan metode *logika fuzzy* kedalam sistem pendukung keputusan dalam menentukan prediksi persalinan pada Rumah Sakit Umum Daerah Prof. Dr. H. Aloei Saboe KotaGorontalo.

***Tujuan***



**Gambar 2.22.** Kerangka Pemikiran