



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Umum**

##### **2.1.1 Komputer**

Kadir (2017:2), Komputer adalah peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia.

Januarti (dalam Ikhsan dan Kurniawan, 2015:13), Komputer adalah sebuah mesin hitung elektronik yang secara cepat menerima informasi masukan digital dan mengolah informasi tersebut menurut seperangkat instruksi yang tersimpan dalam komputer tersebut dan menghasilkan keluaran informasi yang dihasilkan setelah diolah.

Jadi, komputer adalah sebuah perangkat elektronik yang secara cepat menerima dan mengolah informasi untuk memudahkan pekerjaan manusia.

##### **2.1.2 Data**

Rusmawan (2019:34), Data adalah catatan atas kumpulan fakta. Sedangkan menurut Iswandy (dalam Heriyanto, 2018:66), Data adalah sesuatu yang belum mempunyai arti bagi penerimanya dan masih memerlukan adanya suatu pengolahan. Data bisa berwujud suatu keadaan, gambar, suara, huruf, angka, matematika, bahasa ataupun simbol-simbol lainnya yang bisa kita gunakan sebagai bahan untuk melihat lingkungan, obyek, kejadian ataupun suatu konsep.

Berdasarkan definisi diatas dapat disimpulkan bahwa data adalah suatu kumpulan fakta dapat berupa gambar, symbol dll yang masih membutuhkan sebuah pengolahan.

##### **2.1.3 Basis Data**

Connolly dan Begg (dalam Fitria dkk, 2019:2), Basis data adalah kumpulan data yang saling berhubungan secara logikal serta deskripsi dari data tersebut, yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi. Basis Data adalah sebuah penyimpanan data yang besar yang

---



bisa digunakan oleh banyak pengguna dan departemen. Seluruh departemen pada perusahaan harus memegang basis data. Basis Data itu sendiri tidak hanya memegang data operasional organisasi tetapi juga penggambaran dari data tersebut.

Sedangkan menurut Abdulloh (2018:103), Basis data adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi.

Jadi, dari pengertian diatas basis data adalah kumpulan data atau informasi yang disimpan dalam komputer serta dapat menyimpan data yang besar sehingga dapat digunakan oleh banyak pengguna.

## **2.2 Teori Judul**

### **2.2.1 Sistem Informasi**

Menurut Davis (dalam Fitria dkk, 2019:2), Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan data harian, penunjang kegiatan dalam penyimpanan data, dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Sedangkan menurut Susanto (dalam Rosyadi dan Wibiantoro, 2019:24), Sistem informasi adalah kumpulan sub-sub sistem baik fisik maupun non fisik yang saling berhubungan satu sama dan bekerja sama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi yang berguna.

Jadi, sistem informasi adalah kumpulan dari sub-sub sistem yang saling berhubungan sehingga menjadi suatu sistem yang dapat digunakan untuk mengolah data menjadi suatu informasi.

### **2.2.2 Harga**

Menurut Kamus Besar Bahasa Indonesia *Online* (2016), Harga adalah nilai barang yang ditentukan atau dirupakan dengan uang.

Menurut Kotler dan Armstrong (dalam Rosyadi dan Wibiantoro, 2019:24), Harga merupakan sejumlah uang yang dibebankan atas suatu barang atau jasa atau

---



jumlah dari nilai uang yang ditukar konsumen atas manfaat-manfaat karena memiliki atau menggunakan produk atau jasa tersebut.

Dari definisi diatas, harga adalah sejumlah nilai uang untuk menjadi alat tukar suatu barang atau jasa.

### 2.2.3 Bahan Pokok

Menurut keputusan Menteri Industri dan Perdagangan NO. 115/MPP/KEP/2/1998 tanggal 27 Februari 1998 (dalam Juansyah, 2018:88), Berikut daftar nama bahan pokok yaitu antara lain :

- a. Beras, Jagung dan sagu
- b. sayur-sayuran dan buah-buahan
- c. Daging (sapi, ikan dan ayam)
- d. Susu
- e. Telur
- f. Gula pasir
- g. Garam yang mengandung yodium/iodium
- h. Minyak dan margarin
- i. Minyak tanah atau gas elpiji.

### 2.2.4 Fuzzy Logic

Kusumadewi (dalam Permatasari dkk, 2015:33), Teori himpunan logika samar dikembangkan oleh Prof. Lofti Zadeh pada tahun 1965. Zadeh berpendapat bahwa logika benar dan salah dalam logika konvensional tidak dapat mengatasi masalah gradasi berada pada dunia nyata. Untuk mengatasi masalah gradasi tidak terhingga tersebut, Zadeh mengembangkan sebuah himpunan *fuzzy*. Tidak seperti logika boolean, logika *fuzzy* mempunyai nilai kontinue. Samar dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu sama.

Kusumadewi dan Purnomo (dalam Juliansyah, 2015:131), Logika *fuzzy* merupakan salah satu komponen pembentuk soft computing. Dasar logika *fuzzy* adalah teori himpunan *fuzzy*. Pada teori himpunan *fuzzy*, peranan derajat

---



keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting. Nilai keanggotaan atau derajat keanggotaan atau member *ship function* menjadi ciri utama dari penalaran dengan logika *fuzzy* tersebut.

Permatasari dkk (2015:33), Logika *Fuzzy* merupakan suatu logika memiliki nilai kekaburan atau kesamaran (*fuzzyness*) antara benar atau salah. Dalam teori logika *fuzzy* suatu nilai bisa bernilai benar atau salah secara bersama. Namun berapa besar keberadaan dan kesalahan suatu tergantung pada bobot keanggotaan dimilikinya. Logika *fuzzy* memiliki derajat keanggotaan dalam rentang 0 hingga 1. Berbeda dengan logika digital hanya memiliki dua nilai 1 atau 0. Logika *fuzzy* digunakan untuk menterjemahkan suatu besaran diekspresikan menggunakan bahasa (*linguistic*), misalkan besaran kecepatan laju kendaraan 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 klasik tegas (*crisp*), 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.

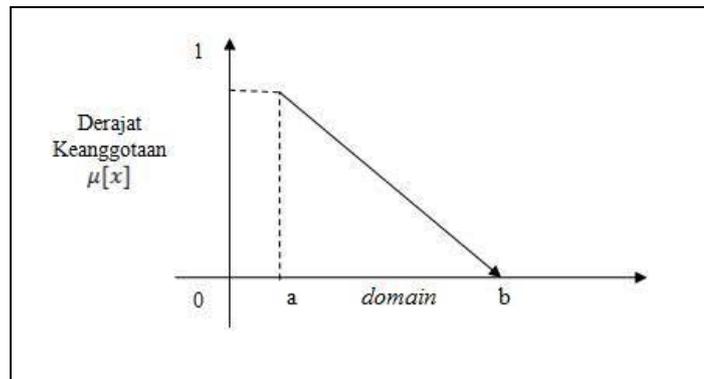
#### 2.2.4.1 Fungsi Keanggotaan *Fuzzy*

Menurut Kusumadewi dan Purnomo (dalam Nugroho dkk, 2019:2582-2583), Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik input pada data ke dalam keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1.

#### 2.2.4.2 Fungsi Keanggotaan Linear

##### 1. Linear Turun

Pada linear turun, terlihat sebaliknya. Dimulai dari domain yang memiliki derajat keanggotaan lebih tinggi lalu kebawah ke arah kanan menuju domain yang memiliki derajat keanggotaan yang lebih rendah. Grafik representasi linear turun ditunjukkan pada gambar 2.1.



**Gambar 2.1** Representasi Linear Turun

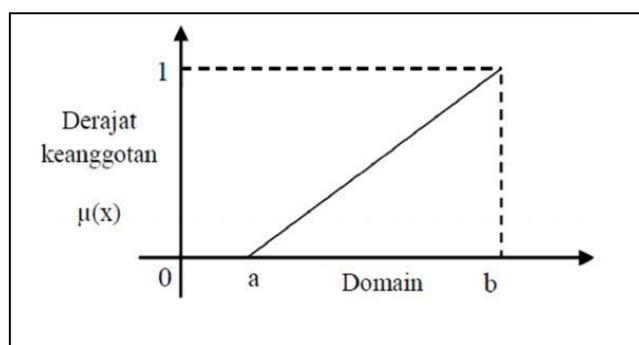
Fungsi keanggotaan dari grafik representasi linear turun pada Gambar 2.1 ditunjukkan pada gambar persamaan 2.2.

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

**Gambar 2.2** Persamaan Linear Turun

## 2. Linear Naik

Pada linear naik, titik domain awal dimulai dari derajat keanggotaan 0 (nol) terlihat naik ke arah kanan menuju domain dengan derajat keanggotaan yang lebih. Grafik representasi linear naik ditunjukkan pada gambar 2.3.



**Gambar 2.3** Representasi Linear Naik

Fungsi keanggotaan dari grafik representasi linear naik pada gambar 2.3 ditunjukkan pada gambar persamaan 2.4.



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

**Gambar 2.4** Persamaan Linear Naik

### 2.2.4.3 Metode Tsukamoto

Nugroho dkk (2019:2583-2584), Metode Tsukamoto mengaplikasikan penalaran monoton pada setiap aturannya. Dimana setiap konsekuen pada aturan yang berbentuk *IF-THEN* harus direpresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Output hasil inferensi dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan  $\alpha$ -predikat (*fire strength*). Proses agregasi antar aturan dilakukan, dan hasil akhirnya diperoleh dengan menggunakan defuzzy dengan konsep rata-rata terbobot. Berikut ini tahapan – tahapan perhitungan *fuzzy* tsukamoto:

1. Tahap pertama dalam perhitungan *Fuzzy* adalah Fuzzifikasi yaitu mengubah nilai tegas (*crisp*) ke nilai *fuzzy*.

$$x = \text{fuzzifier}(x_0)$$

**Gambar 2.5** Rumus Fuzzifikasi

Dimana:

- $x$  = definisi dari variabel dari *vector* himpunan *fuzzy*,
- fuzzifier* = definisi dari mengubah nilai tegas (*crisp*) ke himpunan *fuzzy*,
- $x_0$  = sebuah *vector* nilai tegas dari suatu variabel masukan

2. Tahap kedua yaitu Sistem Inferensi *Fuzzy*. Sistem Inferensi *Fuzzy* adalah penarikan kesimpulan dari aturan atau kaidah *fuzzy* yang didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* yang berbentuk *IF-THEN*, dan penalaran yang memiliki masukan dan keluaran berupa *crisp value*.
3. Tahap ketiga adalah Defuzzifikasi. Defuzzifikasi yaitu proses mengubah *output fuzzy* menjadi nilai tegas (*crisp*) sesuai dengan fungsi keanggotaan yang ditentukan.



$$z = \frac{\sum \alpha_i z_i}{\sum \alpha_i}$$

**Gambar 2.6** Rumus Defuzzifikasi

Dimana:

Z = Defuzzifikasi

$\alpha_i$  = Alpha predikat

z = *Output* inferensi

### 2.2.5 Sistem Informasi Penentuan Harga Bahan Pokok dengan Menggunakan Metode *Fuzzy Logic* pada Dinas Perindustrian dan Perdagangan Kabupaten Penukal Abab Lematang Ilir

Sistem Informasi Penentuan Harga Bahan Pokok dengan Menggunakan Metode *Fuzzy Logic* pada Dinas Perindustrian dan Perdagangan Kabupaten Penukal Abab Lematang Ilir adalah Suatu sistem informasi yang digunakan untuk membantu dalam memprediksi harga bahan pokok dan menginformasikan kepada masyarakat dengan menggunakan metode *Fuzzy Logic*.

## 2.3 Teori Khusus

### 2.3.1 Kamus Data

Sukamto dan Shalahuddin (2018:73-74) Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data biasanya berisi :

- a. Nama-nama dari data.
- b. Digunakan pada proses-proses yang terkait data.
- c. Deskripsi data.
- d. Informasi tambahan seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan



sebagai berikut :

**Tabel 2.1** Simbol Kamus Data

No	Simbol	Keterangan
1	=	disusun atau terdiri dari
2	+	Dan
3	[   ]	baik...atau...
4	{ } <sup>n</sup>	n kali diulang/bernilai banyak
5	( )	data opsional
6	*...*	batas komentar

Sumber : Sukamto dan Shalahuddin (2018:74)

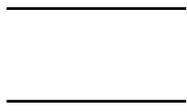
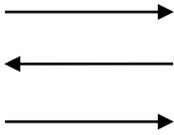
### 2.3.2 Data Flow Diagram (DFD)

Menurut Sukamto dan Shalahuddin (2018:70-72), *Data Flow Diagram* (DFD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang di aplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*). sukamto dan Shalahuddin menjelaskan notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut :

**Tabel 2.2** Simbol *Data Flow Diagram*

No	Notasi	Keterangan
1		Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program. Catatan : Nama yang diberikan pada sebuah proses biasanya berupa kata kerja
		<i>File</i> atau basis data atau penyimpanan ( <i>storage</i> ), pada pemodelan perangkat lunak yang akan

Lanjutan Tabel 2.2 Simbol *Data Flow Diagram*

No	Notasi	Keterangan
2		<p>diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity Relationship Diagram</i> (ERD), <i>Conceptual Data Model</i> (CDM), <i>Physical Data Model</i> (PDM)).</p> <p>Catatan: Nama yang di berikan pada sebuah penyimpanan biasanya kata benda Nama yang diberikan pada sebuah proses biasanya berupa kata kerja.</p>
3		<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang di modelkan atau sistem lain yang terkait dengan aliran data dari sistem yang di modelkan.</p> <p>Catatan : Nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) biasanya berupa kata benda.</p>
4		<p>Aliran data merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan (<i>input</i>) atau keluaran (<i>output</i>).</p> <p>Catatan : Nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “data siswa” atau tanpa kata data misalnya “siswa”</p>

Sumber : Sukamto dan Shalahuddin (2018:71-72)



Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD.

1. Membuat DFD Level 0 atau sering disebut juga *Context Diagram* DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.
2. Membuat DFD Level 1 DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan di kembangkan . DFD level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah di buat.
3. Membuat DFD Level 2 Modul-Modul Pada DFD Level 1 (satu) dapat di-*breakdown* menjadi DFD Level 2 (dua). Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada tingkat ke detailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di-*breakdown*.
4. Membuat DFD Level 3 dan seterusnya DFD Level 3, 4, 5, dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdown* pada level 3,4,5, dan seterusnya aturannya sam persis dengan DFD Level 1 atau Level 2.

### 2.3.3 Entity Relationship Diagram (ERD)

Rusmawan (2019:64), ERD merupakan gambaran grafis dari suatu model data yang menyertakan deskripsi detail dari seluruh entitas (*entity*), hubungan (*relationship*), dan batasan (*constraint*) untuk memenuhi kebutuhan sistem analisis dalam menyelesaikan pengembangan sebuah sistem.

**Tabel 2.3** Simbol-Simbol *Entity Relationship Diagram* (ERD)

No	Simbol	Keterangan
1		Entitas mendeskripsikan tabel.

**Lanjutan Tabel 2.3** Simbol-Simbol *Entity Relationship Diagram* (ERD)

No	Simbol	Keterangan
2		Atribut mendeskripsikan field dalam tabel.
3		Relasi mendeskripsikan hubungan antar tabel.
4		Garis mendeskripsikan penghubung antar himpunan relasi.

Sumber : Rusmawan (2019:64)

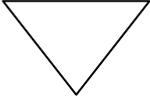
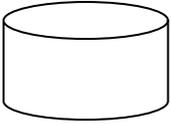
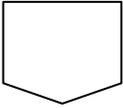
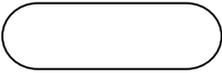
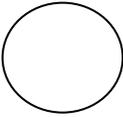
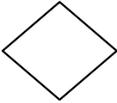
### 2.3.4 Blockchart

Kristanto (2018:75-77), *Blockchart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Adapun simbol-simbol yang sering digunakan dalam *blockchart* dapat dilihat pada tabel berikut ini:

**Tabel 2.4** Simbol *Blockchart*

No	Simbol	Keterangan
1		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku / bendel / berkas atau cetakan
2		Multi dokumen
3		Proses manual

Lanjutan Tabel 2.4 Simbol *Blockchart*

No	Simbol	Keterangan
4		Proses yang dilakukan oleh computer
5		Menandakan dokumen yang diarsipkan (arsip manual)
6		Data penyimpanan ( <i>data storage</i> )
7		Proses apa saja yang tidak terdefinisi termasuk aktivitas fisik
8		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain
9		Terminasi yang menandakan awal dan akhir dari suatu aliran
10		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama
11		Pengambilan keputusan ( <i>decision</i> )
12		Layar peraga ( <i>monitor</i> )

Lanjutan Tabel 2.4 Simbol *Blockchart*

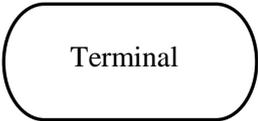
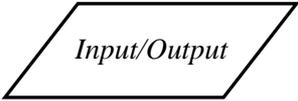
No	Simbol	Keterangan
13		Pemasukan data secara manual

Sumber : Kristanto (2018:75-77)

### 2.3.5 Flowchart

Menurut Harwikarya, dkk (2017:21), *Flowchart* atau diagram alir merupakan salah satu cara mempresentasikan langkah logis pemecahan masalah. Diagram alir terdiri dari beberapa lambing yang telah disepakati di dunia pemrograman. Berikut ini merupakan lambang-lambang diagram alir.

Tabel 2.5 Lambang *Flowchart*

No	Simbol	Keterangan
1		Terminal merupakan lambang untuk mengawali dan menutup satu proses. Ketika Anda akan membuat diagram alir langkah logis satu penyelesaian masalah maka terminal akan mengawali dan menutup langkah-langkah logis tersebut.
2		<i>Input-output</i> berfungsi untuk membaca <i>input</i> dan menampilkan <i>output</i> . Contoh <i>input</i> ketika membaca tinggi dan alas segitiga, <i>output</i> ketika menampilkan luas segitiga tersebut.

Lanjutan Tabel 2.5 Lambang *Flowchart*

No	Simbol	Keterangan
3		Proses merupakan perhitungan yang diperlukan program contoh pada perhitungan luas segitiga, maka proses akan menghitung luas segitiga, yaitu $\text{luas} = \text{alas} * 0,5 * \text{tinggi}$ .
4		<i>Decission</i> merupakan tempat pengujian untuk mengambil keputusan langkah logis selanjutnya, contoh memeriksa apakah nilai mahasiswa $> 60$ jika iya, maka lulus, jika tidak, maka gagal.
5		<i>Connector</i> akan menggabungkan proses jika dalam pembuatan diagram alir ternyata harus pindah ke lain halaman, maka langkah logis akan disambung oleh <i>connector</i> .

Sumber : Harwikarya, dkk (2017:21)

## 2.4 Teori Program

### 2.4.1 Hypertext Processor (PHP)

Menurut Abdulloh (2018:127), PHP merupakan kependekan dari PHP *Hypertext Preprocessor* yaitu bahasa pemrograman web yang dapat disisipkan dalam skrip HTML dan bekerja di sisi server. Tujuan dari bahasa ini adalah membantu para pengembangan web untuk membuat web dinamis dengan cepat.

#### 2.4.1.1 Sintaks Dasar PHP

Menurut Abdulloh (2018:128), Skrip PHP dituliskan di antara tanda `<?php` dan `?>` yang memisahkan skrip PHP dengan skrip lainnya. Satu file PHP dapat berisi full skrip PHP atau dapat disisipkan diantara skrip lain seperti HTML,



CSS maupun JavaScript. Setiap baris skrip PHP harus di akhiri dengan tanda semicolon (;). Jika tidak, maka akan menampilkan pesan error.

Berikut contoh penulisan skrip PHP di dalam skrip HTML.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title> Latihan PHP </title>
  </head>
  <body>
    <?php
      Echo "Latihan PHP";
    ?>
  </body>
</html>
```

#### 2.4.1.2 Tipe Data PHP

Abdulloh (2018:133), menjelaskan bahwa variabel pada PHP dapat menyimpan berbagai macam tipe data. Beda tipe data dapat melakukan hal yang berbeda pula. Berikut tipe data yang di dukung oleh PHP.

1. **String**, yaitu tipe data berupa teks atau angka yang ditulis di antara petik ganda.
2. **Integer**, yaitu tipe data numerik non desimal antara -2,147,483,648 hingga 2,147,483,647.
3. **Float**, yaitu tipe data numerik berupa angka desimal. Tanda koma pada angka desimal menggunakan tanda titik, missal : 23.25.
4. **Boolean**, yaitu tipe data yang hanya dapat bernilai *true* atau *false*.
5. **Array**, yaitu tipe data yang menyimpan banyak nilai dalam satu variabel.
6. **Object**, yaitu tipe data yang menyimpan data beserta informasi bagaimana data di proses.
7. **Null**, yaitu tipe data yang hanya dapat bernilai NULL.



Untuk dapat menampilkan tipe data beserta nilainya pada PHP dapat menggunakan perintah **var\_dump()** contohnya seperti berikut:

```
<? Php
    $kata = "Selamat pagi kawan";
    $angka = 210;
    Var dump ($kata);
    Var dump ($angka);
?>
```



**Gambar 2.7** Logo PHP

#### 2.4.2 MySQL

Rusmawan (2019:97), Berpendapat bahwa MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Manajemen System*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia.

Menurut Amin (dalam Bahrudin dkk, 2019:75), MySQL merupakan *software* yang tergolong *database server* dan bersifat *Open Source*. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* langsung dalam sistem operasi, dan bisa diperoleh dengan cara mengunduh di internet secara gratis. Hal menarik lainnya adalah MySQL juga bisa bersifat *flatfrom*, mysql dapat dijalankan pada beberapa sistem operasi.



Jadi, pengertian MySQL yang dapat disimpulkan adalah sebuah perangkat lunak manajemen basis data atau *database server* yang bersifat flatfrom sehingga bisa dijalankan di beberapa sistem informasi.



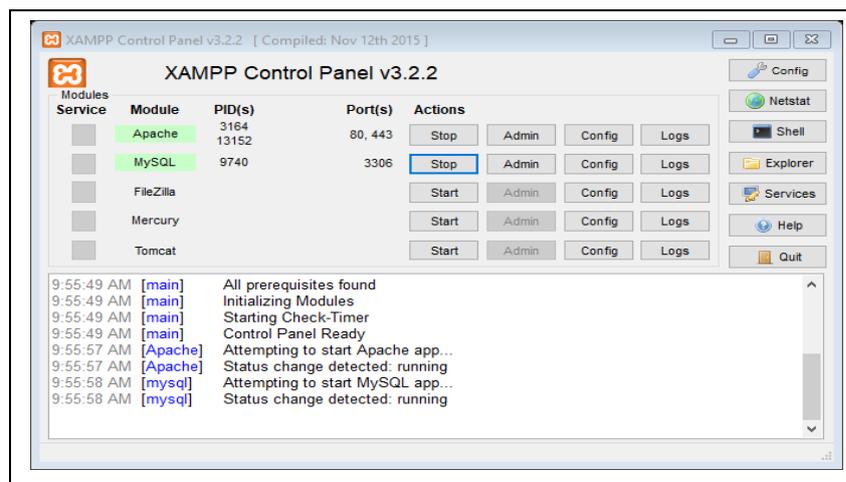
**Gambar 2.8** Logo MySQL

### 2.4.3 XAMPP

Noor (2016:78), XAMPP adalah *software* web server apache yang di dalamnya tertanam server MySQL yang didukung dengan bahasa pemrograman PHP untuk membuat *website* yang dinamis. XAMPP mendukung dua sistem operasi yaitu windows dan Linux.

Menurut Madcoms (dalam Ayu dan Permatasari, 2018:19), XAMPP adalah sebuah paket kumpulan *software* yang terdiri dari Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla, dan lain. XAMPP berfungsi untuk memudahkan instalasi lingkungan PHP, di mana biasanya lingkungan pengembangan web memerlukan PHP, Apache, MySQL dan PhpMyAdmin.

Jadi, XAMPP adalah *software* Apache yang didalamnya terdapat server MySQL dan PhpMyAdmin yang berfungsi untuk melakukan instalasi lingkungan PHP.



Gambar 2.9 XAMPP

#### 2.4.4 Notepad++

Noor (2016:78), *Notepad++* adalah program aplikasi pengembang yang berguna untuk mengedit teks dan skrip kode pemrograman. *Notepad++* bisa mengenal *tag* dan kode dalam berbagai bahasa pemrograman. Fitur pencarian tingkat lanjut dan pengeditan teks yang tersedia juga cukup ampuh, sangat membantu tugas seorang *programmer* atau developer dalam menyelesaikan *skrip* kode programnya. Program *Notepad++* banyak diaplikasikan dan digunakan oleh kalangan pengguna komputer di bidang pemrograman aplikasi desktop dan web.

Menurut Madcoms (dalam Ayu dan Permatasari, 2018:21), *Notepad++* adalah sebuah teks editor yang sangat berguna dalam membuat program. *Notepad++* menggunakan komponen Scintilla untuk menampilkan teks dan berkas kode sumber berbagai bahasa pemrograman yang berjalan diatas sistem operasi M. Windows.

Jadi, pengertian *Notepad++* yang dapat disimpulkan adalah sebuah program teks editor pemrograman yang dapat membantu para *programmer* dalam membuat program atau penyelesaian *skrip* kode programnya



```

54 <body class="skin-blue">
55 <!-- header logo: style can be found in header.less -->
56 <header class="header">
57 <a href="index.php" class="logo">
58 | 
59 </a>
60 <!-- Header Navbar: style can be found in header.less -->
61 <nav class="navbar navbar-static-top" role="navigation">
62 <!-- Sidebar toggle button-->
63 <a href="#" class="navbar-btn sidebar-toggle" data-toggle="offcanvas" role="button">
64 <span class="sr-only">Toggle navigation</span>
65 <span class="icon-bar"></span>
66 <span class="icon-bar"></span>
67 <span class="icon-bar"></span>
68 </a>
69 <div class="navbar-right">
70 <ul class="nav navbar-nav">
71
72 <!-- User Account: style can be found in dropdown.less -->
73 <li class="dropdown user user-menu">
74 <a href="#" class="dropdown-toggle" data-toggle="dropdown">
75 <i class="fa fa-user"></i>
76 <span><?php echo "Admin Daskrimti dan Perpustakaan"; ?> <i class="caret"></i></span>
77 </a>
78 <ul class="dropdown-menu dropdown-custom dropdown-menu-right">
79
80 <li>
81 | <a href=".." />
82 | <a href=".." />
83 | <a href=".." />
84 | <a href=".." />
85 | <a href=".." />
86 </li>
</ul>
</div>

```

Gambar 2.10 Tampilan Notepad++

## 2.5 Metode Pengembangan *Rapid Application Development* (RAD)

Kojongian dkk (2017:3), RAD merupakan salah satu metode pengembangan suatu sistem informasi dengan waktu yang *relative* singkat. RAD adalah sebuah strategi pengembangan sistem dimana menekan kecepatan pengembangan melalui keterlibatan *user* ekstensif dalam kecepatan, *iterative* (berulang) dan *incremental construction* dari serangkaian fungsi dari prototype sebuah sistem yang pada akhirnya akan mengalami perubahan secara bertahap menuju sistem akhir.

Menurut Mediana dan Nurhidayat (2018:77), *Rapid Application Development* adalah salah satu model proses pengembangan perangkat lunak *sequential* linear yang menekankan pada siklus pengembangan yang sangat pendek. Jika kebutuhan dapat dipahami dengan baik tidak menutup kemungkinan tim pengembang akan menciptakan sistem yang sempurna secara fungsional dalam waktu kira kira 60-90 hari. Metode RAD umumnya dipakai untuk pembuatan aplikasi sistem konstruksi.

### 2.5.1 Tahapan Aktifitas Pemodelan *Rapid Application Development* (RAD)

Berikut adalah tahapan aktifitas pemodelan RAD menurut Yurinda (dalam Mediana dan Nurhidayat (2018:78) :



### 1. *Bussiness Modelling*

Tahap ini bertujuan untuk mendefinisikan semua kebutuhan informasi dalam sebuah sistem seperti perangkat fisik, manusia, anggaran, dan lain sebagainya. Biasanya pada tahap diajukan pertanyaan-pertanyaan seperti informasi yang mengendalikan proses bisnis, informasi yang sering dimunculkan, dan pihak yang memroses.

### 2. *Data Modelling*

Pada tahap ini aliran informasi yang sudah didapatkan pada fase sebelumnya disaring ulang ke dalam objek data yang dibutuhkan untuk menopang bisnis tersebut. Karakteristik masing-masing objek diidentifikasi dan didefinisikan hubungannya.

### 3. *Process Modelling*

Aliran informasi yang didefinisikan pada fase sebelumnya ditransformasikan untuk membentuk sebuah aliran informasi yang dibutuhkan untuk implementasi sebuah bisnis.

### 4. *Application Generation*

Pada fase ini aplikasi mulai dibangun. RAD lebih banyak memroses sebuah sistem untuk digunakan kembali komponen-komponen program yang ada.

### 5. *Testing and turnover*

RAD menekankan pada pemakaian kembali komponen-komponen program yang ada sehingga mengurangi waktu keseluruhan pengujian. Tetapi, pada komponen yang baru tetap dilakukan pengujian secara keseluruhan.

## **2.5.2 Tahapan Model *Rapid Application Development* (RAD)**

Aswati & Siagia (dalam Mediana dan Nurhidayat (2018:78), Model RAD memiliki tiga tahapan yaitu requirement planning, design system, dan implementation. Berikut adalah tiga tahapan pada model Rapid Application Development (RAD):

---



### 1. Requirement Planning

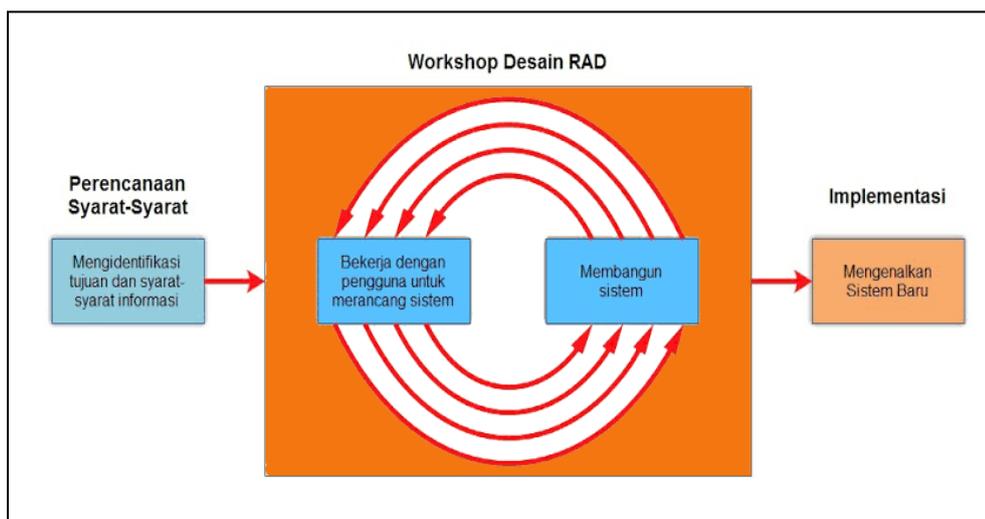
Pengguna dan tim pembangun melakukan pertemuan untuk mengidentifikasi tujuan dan kebutuhan informasi pada sistem yang akan dibuat. Tahap ini merupakan tahap terpenting karena harus melibatkan antara pengguna dengan tim analisis aplikasi.

### 2. Design System

Pada tahap ini peran pengguna yang terlibat sangat dibutuhkan. Hal tersebut terjadi karena pada proses ini dilakukan proses desain dan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara pengguna dan tim analisis. Seorang pengguna dapat langsung memberikan komentar jika terdapat ketidaksesuaian pada desain dan rancangan sistem. Keluaran dari tahapan ini adalah spesifikasi sistem yang meliputi organisasi sistem secara umum, dan struktur data.

### 3. Implementation

Pada Tahap ini adalah desain dan alur sistem yang sudah disetujui oleh pengguna dengan tim analisis akan mulai dibangun oleh programmer. Sistem akan dilakukan proses pengujian atau testing baik secara *black box* atau *white box* sebelum diaplikasikan pada sebuah organisasi. Pada tahap ini biasanya sistem yang sudah dibuat akan ditanggapi dan disetujui oleh pengguna.



**Gambar 2.11** Rapid Application Development (RAD)



## 2.6 Referensi Penelitian Sebelumnya

1. Penelitian yang telah dilakukan oleh Amelia (2013), masalah pada jurnal ini adalah Tidak stabilnya harga barang dipasaran setiap saat menyebabkan perusahaan harus terus memantau fluktuasi harga barang dan menentukan harga barang tersebut dengan tepat. Penggunaan metode Logika *Fuzzy* Tsukamoto dapat membantu pimpinan perusahaan dalam menentukan harga barang suatu produk tersebut.
2. Penelitian yang telah dilakukan oleh Nugroho dkk (2019), penelitian ini menerapkan metode Logika *Fuzzy* Tsukamoto untuk memecahkan masalah yaitu kesulitan dalam menentukan harga sewa hotel karena mereka hanya mengira-ngira harga hotel yang akan disewakan berdasarkan musim yang berlangsung. Hal ini menimbulkan keuntungan yang kurang stabil.
3. Penelitian yang telah dilakukan oleh Sari dkk (2017), penelitian ini bertujuan untuk menentukan harga final yang akan didapatkan oleh *developer* pembuat aplikasi dengan menertapkan metode Logika *Fuzzy* Tsukamoto.
4. Penelitian yang telah dilakukan oleh Boki dkk (2016), penelitian ini dilakukan untuk membantu pihak sekolah dalam menentukan Guru berprestasi secara efisien dan terkomputerisasi dikarenakan belum adanya sistem pendukung keputusan. Pemilihan tersebut tentu memiliki banyak kekurangan terutama dari segi objektivitas
5. Penelitian yang telah dilakukan oleh Novita (2016), penelitian ini dilakukan untuk menentukan jumlah dan jenis beasiswa yang akan didapat oleh mahasiswa berdasarkan IP dan Jumlah pendapatan orang tua dengan menerapkan metode Logika *Fuzzy* Tsukamoto.