# BAB 2
# LANDASAN TEORI

## 2.1 *Front Office*

Istilah *front office* jika diartikan berarti ‘kantor depan’. Pada sebuah hotel, *front office department* merupakan suatu bagian paling depan dari sebuah perusahaan yang berfungsi sebagai penjual kamar dan memberikan layanan seperti menangani surat, fax, pesan, dan informasi yang berkaitan dengan hotel tersebut [7].

Sebuah kampus memiliki *front office* yang memiliki tugas mirip dengan *front office* pada sebuah hotel. Jika *front office* pada sebuah hotel bertugas untuk memberikan informasi yang terkait dengan hotel tersebut, maka petugas *front office* pada sebuah kampus bertugas untuk memberikan layanan yang baik berupa informasi-informasi yang berkaitan dengan kampus yang bersangkutan.

Universitas Komputer Indonesia, atau lebih dikenal dengan nama UNIKOM memiliki kantor *front office* yang terletak di Jalan Dipatiukur, Bandung. Bagian *front office* di UNIKOM memiliki 3 orang staf utama yang bertugas untuk melayani permintaan informasi dari pengunjung kampus UNIKOM.

Pengunjung yang mengunjungi *front office* ini beragam, baik dari segi umur, maupun status, seperti status sebagai mahasiswa, dosen, orang tua, wali, dan sebagainya. Jumlah pengunjung terhitung rendah pada saat hari-hari biasa, dan akan meningkat pada saat hari-hari tertentu, seperti pada saat penerimaan mahasiswa baru. Puncak dari jumlah pengunjung ada pada saat gelombang terakhir pendaftaran mahasiswa baru, yang mencapai lebih dari 100 pengunjung per hari. Biasanya pada saat-saat tersebut jam kerja pelayanan *front office* akan bertambah karena tidak seimbangnya jumlah staf dengan jumlah pengunjung yang membutuhkan pelayanan.

Pertanyaan-pertanyaan yang diajukan pengunjung diantaranya, mengenai tata cara, waktu, dan biaya untuk *autodebet* atau pembayaran perkuliahan per semester, tata cara, waktu, dan biaya pendaftaran, serta lokasi ruangan.

## 2.2 Tanya Jawab (*Question & Answering*)

Menurut KBBI, arti kata ‘tanya’ adalah ‘permintaan keterangan’, arti kata ‘jawab’ adalah ‘sahut’ atau ‘balas’, dan arti kata ‘proses’ adalah ‘rangkaian tindakan, pembuatan, atau pengolahan yang menghasilkan produk’. Dari keterangan tersebut bisa disimpulkan bahwa ‘proses tanya jawab’ adalah rangkaian tindakan permintaan keterangan yang mendapatkan sahutan atau balasan. Dalam hal ini, sahutan atau balasan yang terjadi tentu saja harus berhubungan dengan permintaan keterangan yang ada.

Contoh mudah proses tanya jawab adalah proses tanya jawab yang dilakukan oleh pengunjung kampus dengan bagian *front office*. Dalam hal ini pengunjung melakukan permintaan keterangan mengenai suatu hal, dan *front office* akan memberikan sahutan atau balasan yang berisi informasi mengenai keterangan yang diminta.

Sistem Tanya Jawab (QA) merupakan sebuah sistem pada komputer yang berisi sebuah cara otomatis untuk menjawab pertanyaan menggunakan bahasa alami yang biasa dipakai manusia. Jawaban ini akan berisi informasi yang berasal dari sebuah sumber basis data, baik terstruktur maupun tidak terstruktur [1]. Dengan kata lain, sistem tanya jawab merupakan implementasi proses tanya jawab pada komputer.

Beberapa contoh Sistem Tanya Jawab yang sudah pernah dibangun diantaranya, mesin pencari di internet (Google, Yahoo, Yandex, Naver), SIRI, Google Now, dan Microsoft Cortana. Penelitian terkait pada penelitian ini adalah Sistem Tanya Jawab yang menggunakan *keyword-matching*, dan Sistem Pengambilan Keputusan mengenai resep obat yang menggunakan metode MCRDR.

Pada penelitian ini, sistem tanya jawab akan menggunakan metode *Multiple Classification Ripple Down System* yang merupakan bagian dari metode *rule-based*.

## 2.3 Penelitian Terkait

Kajian terhadap penelitian sebelumnya merupakan hal yang perlu dilakukan dan dapat dijadikan sebagai data pendukung penelitian. Penelitian mengenai sistem tanya jawab, diantaranya dilakukan oleh Sando Han, Hyosup Shim, Byungsoo Kim, Seonyeon Park, Seonghan Ryu, dan Gary Geunbae Lee (Pohang University of Science and Technology, 2015) dengan judul *Keyword Question Answering System with Report Generation for Linked Data*. Hasil penelitian ini adalah sebuah sistem tanya jawab yang menggunakan *keyword matching* untuk mendapatkan informasi atau jawaban yang relevan dengan *keyword* tersebut [2].

Ivan Karl Bindoff membangun sebuah sistem pengambilan keputusan mengenai resep obat menggunakan metode *rule-based* MCRDR pada penelitiannya yang berjudul *An Intelligent Decision Support System for Automated Medication Review* yang dipublikasikan oleh Universitas Tasmania pada tahun 2005 [3]. Sistem pengambilan keputusan ini memiliki data masukan berupa keluhan-keluhan yang diderita pasien untuk diproses dan selanjutnya sistem akan memberikan resep obat yang sesuai dengan keluhan-keluhan tersebut.

Data masukan dan keluaran dari kedua penelitian sebelumnya adalah kalimat-kalimat atau kata-kata dalam bahasa Inggris. Berdasarkan ketiga penelitian terdahulu tersebut, maka penulis melakukan penelitian mengenai sistem tanya jawab yang dikhususkan untuk menerima kalimat dan kata-kata dalam Bahasa Indonesia.

## 2.4 *Preprocessing*

Dalam pembangunan sebuah sistem tanya jawab atau *question & answering system (QA system)*. Perlu dilakukan beberapa tahapan supaya data masukan bisa diproses oleh sistem. Pada tahap awal, karakter, kata, dan kalimat masukan merupakan sebuah kesatuan data yang tidak terstruktur. Disini *preprocessing* berperan untuk mengubah data tersebut menjadi data yang terstruktur yang hasilnya akan digunakan sebagai masukan proses berikutnya [8]. *Preprocessing* sendiri terdiri dari beberapa tahapan, diantaranya : *Case Folding,* Penghilangan

Emotikon & Simbol, Konversi kata tidak baku, Penghilangan Stopword, dan Tokenisasi.

### 2.4.1 *Case Folding*

*Case Folding* merupakan proses dimana semua huruf pada teks masukan diubah menjadi huruf non-kapital atau kapital [9]. *Case folding* diperlukan supaya tidak ada permasalahan pada sistem saat memproses kata yang sama dengan tipe huruf yang berbeda. Contohnya kata **Makan** dan **makan**, dua kata tersebut sama, tetapi karena **Makan** diawali dengan huruf kapital, sistem akan menganggap kata tersebut berbeda dengan kata **makan**.

**Tabel 2.1 Contoh *Case Folding***

| | | |
|---|---|---|
| **Makan** | ➔ | **makan** |

### 2.4.2 Penghilangan Emotikon & Simbol

Penghilangan Emotikon & Simbol merupakan proses dimana *emoticon* dan simbol-simbol yang tidak diperlukan akan dihilangkan. *Emoticon* merupakan salah satu bentuk komunikasi teks informal yang berisikan simbol-simbol untuk menunjukkan emosi pada sebuah komunikasi [10]. Bentuk-bentuk *emoticon* ini sendiri biasanya mirip dengan raut wajah dan bahasa tubuh dalam komunikasi langsung antar manusia. Beberapa contoh dari *emoticon*, misalnya :

:) :( :| :o ^_^ ^o^ :3

**Tabel 2.2 Contoh Penghilangan Emotikon & Simbol**

| | | |
|---|---|---|
| **Apakah anda sudah makan? :)** | ➔ | **Apakah anda sudah makan** |

### 2.4.3 Konversi Kata Tidak Baku

Konversi kata tidak baku adalah proses dimana sebuah kalimat akan di analisa setiap katanya. Dalam kalimat tersebut, setiap kata yang merupakan kata tidak baku, seperti **klo, bru, kmu, sy,** dan sebagainya akan dikonversi menjadi bentuk bakunya : **kalau, baru, kamu, saya**.

**Tabel 2.3 Contoh Konversi Kata Tidak Baku**

| | | |
|---|---|---|
| **klo** | ➔ | **kalau** |
| **bru** | ➔ | **baru** |
| **kmu** | ➔ | **kamu** |
| **sy** | ➔ | **saya** |

### 2.4.4 Tokenisasi

Tokenisasi merupakan sebuah proses pemecahan sebuah teks menjadi bagian-bagian kata, frasa, simbol, dan elemen elemen lainnya yang disebut *token* [8]. Tujuan dari tokenisasi ini adalah untuk mengambil kata-kata yang akan diproses dari sebuah kalimat. Tokenisasi berbeda-beda untuk setiap bahasa, dikarenakan setiap bahasa memiliki struktur yang berbeda-beda pula. Salah satu contoh tokenisasi bisa dilihat pada contoh berikut.

**Tabel 2.4 Contoh Tokenisasi**

| **Adik berangkat ke sekolah naik bus** |
|---|
| ↓ |

| | | | | | |
|---|---|---|---|---|---|
| **Adik** | **berangkat** | **ke** | **sekolah** | **naik** | **bus** |

## 2.5 Kata Tanya

Kata Tanya merupakan salah satu jenis kata yang berfungsi untuk menanyakan sesuatu. Di dalam bahasa Indonesia, kata tanya utama beserta fungsinya bisa dilihat pada tabel berikut.

**Tabel 2.5 Daftar Kata Tanya Pada Bahasa Indonesia**

| Kata Tanya | Fungsi | Contoh |
|---|---|---|
| apa, apakah | Menanyakan benda. | Apa yang kamu bawa? |
| siapa, siapakah | Menanyakan orang. | Siapa yang menjadi juara? |
| kapan, kapankah | Menanyakan waktu. | Kapan kakek datang? |
| dimana, kemana, darimana, dimanakah, kemanakah, darimanakah | Menanyakan lokasi | Dimana Rumahnya?<br>Kemana saya harus pergi?<br>Darimana dia berasal? |
| berapa, berapakah | Menanyakan Jumlah | Berapa yang harus saya bayar? |
| bagaimana, bagaimanakah | Menanyakan keadaan atau cara melakukan sesuatu | Bagaimana cuaca hari ini? |
| mengapa, kenapa, mengapakah, kenapakah | Menanyakan alasan | Mengapa kamu tidak hadir hari ini? |

## 2.6 Metode *Rule-Based*

*Rule* jika diterjemahkan secara literal berarti aturan. Disekitar kita banyak sekali terdapat aturan-aturan yang ada. Fungsi dari aturan-aturan tersebut adalah untuk membatasi apa saja yang dilakukan oleh manusia. Beberapa contoh aturan yang ada di kehidupan kita misalnya, hukum, aturan lalu lintas, dan aturan penggunaan sebuah peralatan.

Di dalam dunia informatika, *Rule-Based System* akan memberikan sebuah alat yang berguna dalam pembangunan aplikasi-aplikasi tertentu. Menurut Antoni Ligeza (2006), bentuk dasar dari *rule-based system* adalah sebagai berikut.

| *rule : (preconditions) -> (conclusion)* |
|---|

Dimana (*preconditions*) adalah sebuah formula yang mendefinisikan kapan *rule* atau aturan tersebut diaplikasikan, dan (*conclusion*) adalah definisi dari efek dari pengaplikasian *rule* atau aturan tersebut [11].

Banyak sekali penelitian yang menggunakan metode *rule-based* ini, seperti pada penelitian berjudul *Rule-Based Highway Maneuver Intention Recognition* (2015) yang membahas tentang penerapan metode *rule-based* pada *driver assistance system* atau sistem pembantu pengemudi [12], *An Algebraic Approach to Rule-Based Information Extraction* (2008) yang membahas tentang penerapan *rule-based* untuk ekstraksi informasi [13], dan *Rule-Based Sentiment Analysis for Financial News* (2015) yang membahas mengenai metode *rule-based* analisis sentimen pada berita keuangan [14].

Beberapa contoh metode *rule-based* yang banyak digunakan adalah sebagai berikut.

- *Ripple Down Rules*

  *Ripple Down Rules* (RDR) merupakan salah satu jenis metode *rule-based* yang melakukan pendekatan secara sistematis untuk mendapatkan informasi dari *experts* atau seseorang yang berkompeten dalam bidangnya [15].

- *Multiple Classification Ripple Down Rules*

  *Multiple Classification Ripple Down Rules* (MCRDR) merupakan ekstensi atau pengembangan dari RDR. Penjelasan mengenai MCRDR dapat dilihat pada sub bab 2.5.

- *Decision Tree*

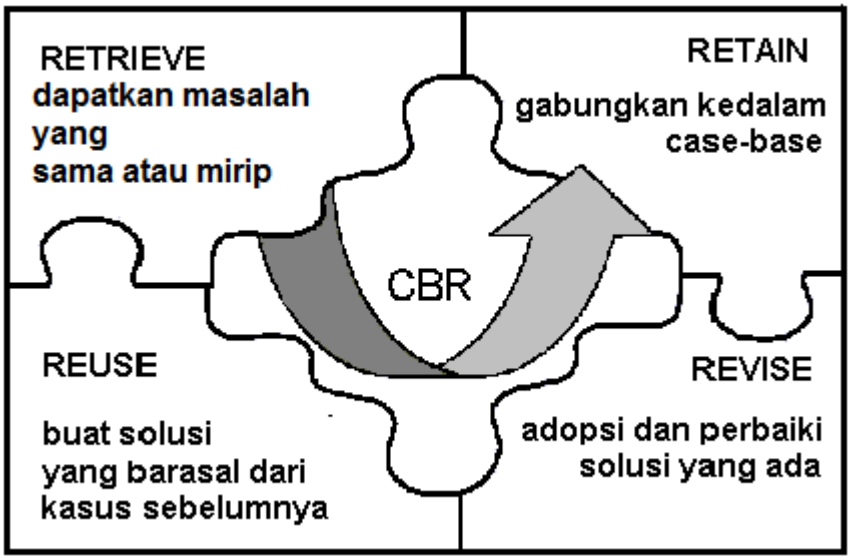
  *Decision Tree* merupakan sebuah model klasifikasi yang berbentuk *tree* atau pohon. Model ini dibangun melalui proses pembelajaran, yang berdasarkan pengumpulan data pada kasus tertentu [16]. Gambar berikut merupakan contoh *decision tree* dalam mengklasifikasikan binatang.




**Gambar 2.1 Contoh *Decision Tree* Klasifikasi Binatang**

- *Case-based reasoning*

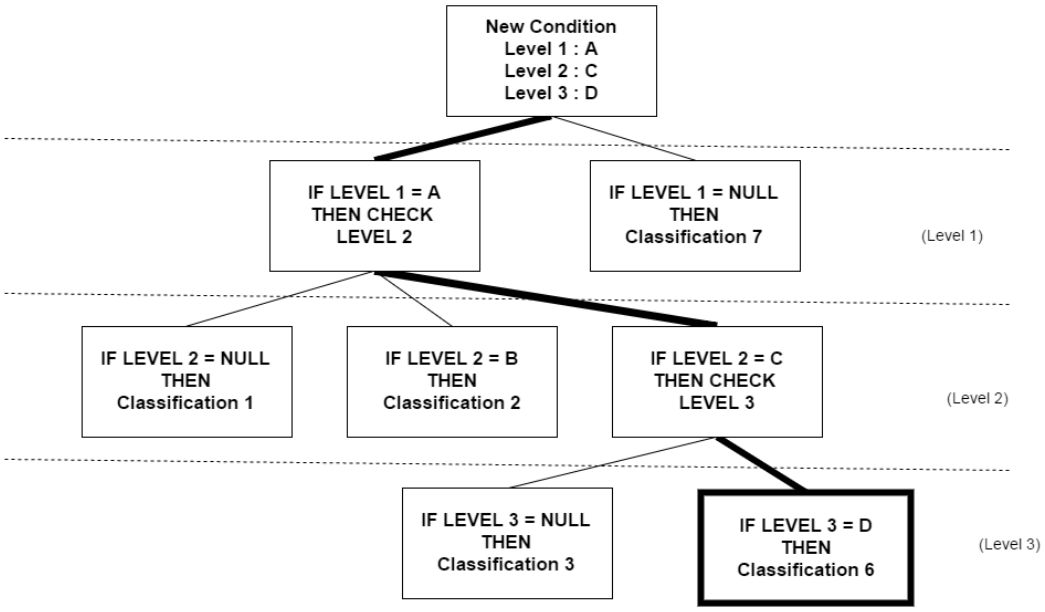
  *Case-based reasoning* adalah suatu model yang menggunakan kasus nyata yang terlah didokumentasikan dengan baik sebagai sarana pembelajaran [17].




**Gambar 2.2 Contoh Model *Case-Based Reasoning* [17]**

## 2.7 Multiple Classification Ripple Down Rules (MCRDR)

Pada penelitian ini metode *rule-based* yang digunakan adalah metode *Multiple Classification Ripple Down Rules* (MCRDR). MCRDR merupakan ekstensi dari RDR, dimana MCRDR adalah sebuah metode akuisisi informasi yang memungkinkan seorang ahli melakukan baik proses akuisisi informasi, ataupun proses *maintenance* pada sebuah sistem pakar [18]. Penggunaan MCRDR ditujukan untuk menutupi kelemahan RDR dimana RDR hanya cocok digunakan untuk masalah yang memerlukan klasifikasi tunggal atau hanya memiliki satu kategori solusi. Pertanyaan yang dimasukkan oleh pengguna akan sangat mungkin memiliki lebih dari satu kategori solusi untuk setiap pertanyaannya. Contohnya pada pertanyaan '**berapa dan kapan'**, pertanyaan tersebut akan memiliki dua kategori jawaban, jawaban untuk kata tanya **berapa** dan jawaban untuk kata tanya **kapan**. Gambar berikut akan menunjukkan skema *tree* dari sebuah MCRDR.




**Gambar 2.3 Skema Tree Dari MCRDR**

Pada gambar di atas, bisa dilihat bahwa untuk sebuah kasus dengan level 1 A, level 2 C, dan level 3 D memiliki klasifikasi nomor 6. Penerapan kasus pada penelitian ini adalah tentang **kata tanya, kata kunci, keterangan**.

## 2.8 *Entity Relationship Diagram* (ERD)

ERD merupakan suatu model jaringan yang menggunakan susunan data yang disimpan pada sistem secara abstrak. ERD juga menggambarkan hubungan antara satu entitas yang memiliki sejumlah atribut dengan entitas lain dalam suatu sistem yang terintegrasi. ERD digunakan oleh perancang sistem untuk memodelkan data yang nantinya akan dikembangkan menjadi basis data. ERD ini juga merupakan model konseptual yang dapat mendeskripsikan hubungan antara file yang digunakan untuk memodelkan struktur data serta hubungan antar data. [19]

ERD terbagi atas tiga komponen, yaitu entitas (*entity*), atribut (*atribute*), dan relasi atau hubungan (*relation*). Secara garis besar entitas merupakan dasar yang

terlibat dalam sistem. Atribut atau field berperan sebagai penjelas dari entitas, dan relasi atau hubungan menunjukkan hubungan yang terjadi antara dua entitas.

1. Entitas (*Entity*)

   Entitas (*entity*) menunjukkan objek-objek dasar yang terkait di dalam sistem. Objek dasar dapat berupa orang, benda atau hal lain yang keterangannya perlu disimpan dalam basis data.

2. Atribut (*Atribute*)

   Atribut sering juga disebut sebagai properti merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan sebagai basis data. Atribut berfungsi sebagai penjelas sebuah entitas.

3. Relasi (*Relation*)

   Relasi atau hubungan adalah kejadian atau transaksi yang terjadi di antara dua entitas yang keterangannya perlu disimpan dalam basis data.

4. Derajat Relasi (Kardinalitas)

   Kardinalitas relasi menunjukkan maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas relasi yang terjadi di antara dua himpunan entitas (misalkan A dan B) dapat berupa satu ke satu (*one to one*), satu ke banyak (*one to many*), banyak ke satu (*many to one*), dan banyak ke banyak (*many to many*). [19]

   1. Satu ke Satu (*One to One*)

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

   2. Satu ke Banyak (*One to Many*)

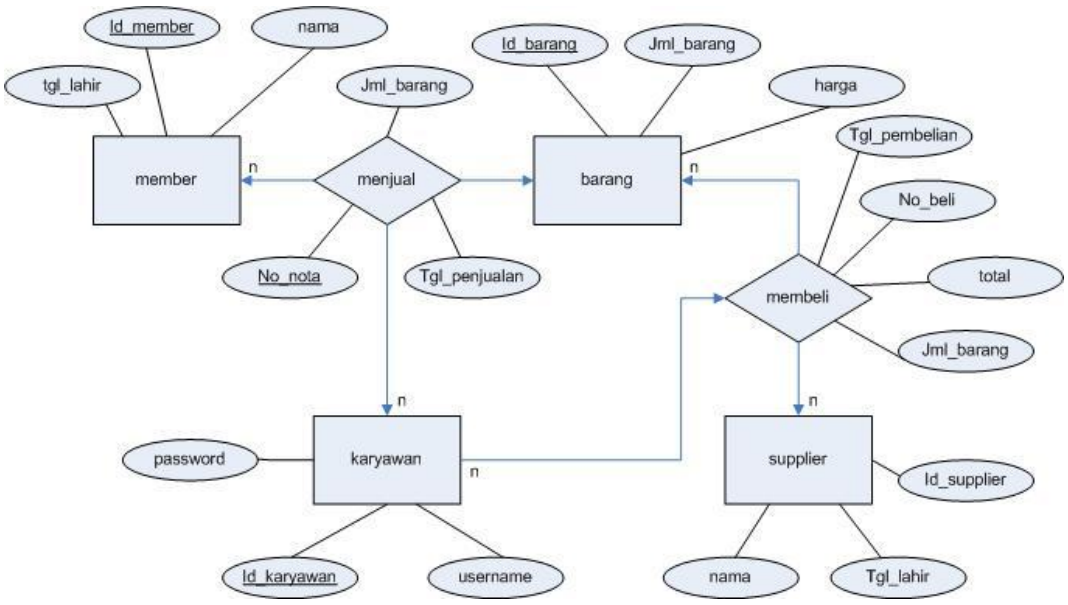
      Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himbungan entitas B, tetapi tidak sebaliknya di mana setiap entitas pada himpunan entitas berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.

   3. Banyak ke Satu (*Many to One*)

Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan b, tetapi tidak sebaliknya, di mana setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B.

4. Banyak ke Banyak (*Many to Many*)

   Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, demikian juga sebaliknya, di mana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.




**Gambar 2.4 Contoh ERD Pada Sistem Informasi Toko**

## 2.9 Diagram Konteks *dan Data Flow Diagram* (DFD)

Diagram konteks merupakan alat pemodelan atau suatu diagram yang menggambarkan sistem berbasis komputer yang dirancang secara global dan merupakan suatu diagram alir data tingkat atas, dimana di dalam diagram konteks ini menggambarkan seluruh jaringan, baik masukan maupun sebuah keluaran sebuah sistem. Diagram konteks ini juga biasa disebut dengan DFD Level 0. [20]

DFD merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi yang dapat digunakan untuk penggambaran analisis maupun rancangan sistem yang mudah dikomunikasikan kepada pemakai maupun pembuat program. [20]

1. Arus Data (*Data Flow*)

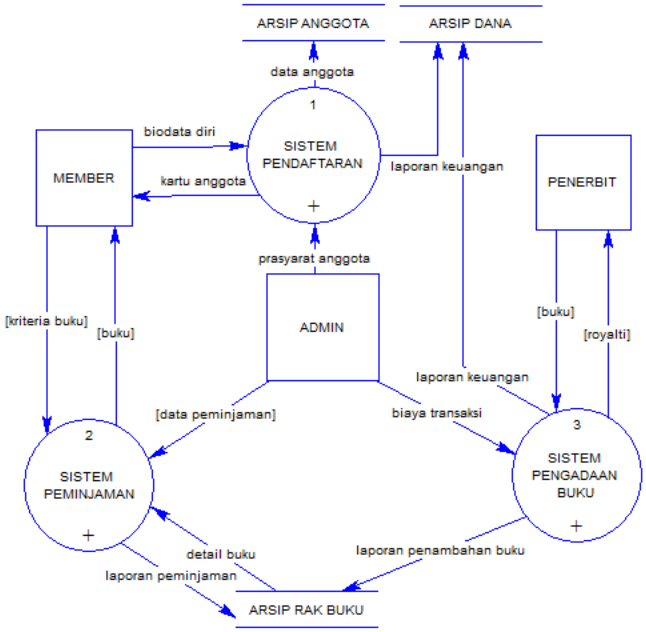
   Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.
2. Proses

   Proses adalah kegiatan yang dilakukan oleh orang, mesin atau komputer dari hasil arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.
3. Kesatuan Luar *( External Entity)*

   Kesatuan luar merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain yang akan memberikan masukan *(input)* atau menerima keluaran *(output)* dari sistem.

4. *File*

Kumpulan data yang disimpan dengan cara tertentu. Data yang mengalir disimpan dalam *file*. Aliran data di-*update* atau ditambahkan ke dalam *file*.




**Gambar 2.5 Contoh DFD Pada Sistem Informasi Perpustakaan**

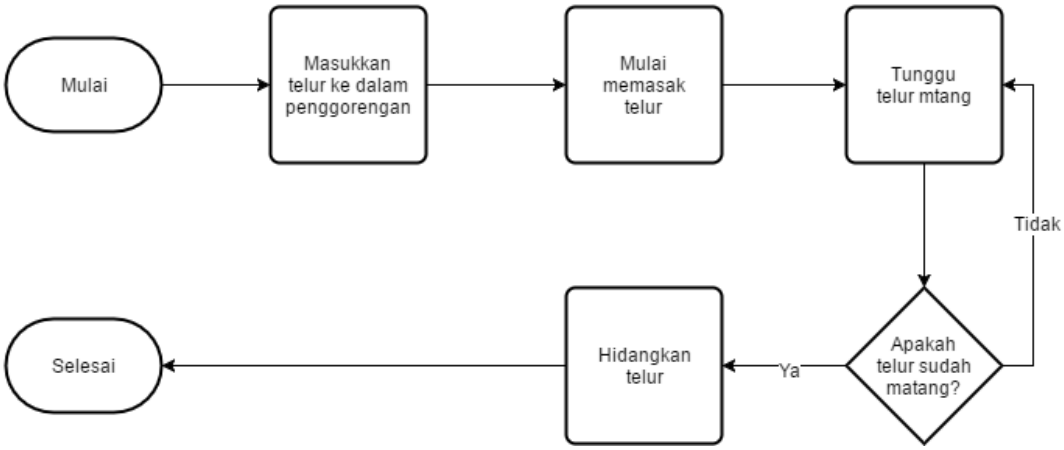
## 2.10 Kamus Data

Kamus data adalah katalog fakta tentang data dan kebutuhan informasi dari suatu sistem informasi. Kamus data merupakan tempat penyimpanan definisi dari aliran-aliran data, file-file dan proses-proses dalam sebuah sistem informasi [20]. Kamus data dapat memberikan informasi mengenai definisi struktur pemakaian masing-masing elemen, dapat mendefinisikan data yang mengalir di sistem dengan lengkap, dapat menghindari duplikasi elemen-elemen dan menghindari konflik antara elemen-elemen. Kamus data berfungsi untuk membantu pelaku sistem untuk mengartikan alokasi secara detail dan mengorganisasikan semua elemen data yang digunakan dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

## 2.11 *Flowchart*

*Flowchart* atau Bagan Alir Program merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program [20]. *Flowchart* dibuat menggunakan simbol-simbol yang terdapat pada daftar simbol sebelumnya. *Flowchart* ini akan menggambarkan logika setiap langkah proses dari sebuah program, dimana logika ini bisa dianalogikan dengan logika yang ada pada kehidupan sehari-hari.

Penggunaan *flowchart* terhadap logika yang ada pada kehidupan sehari-hari bisa di terapkan dalam analogi memasak telur. Logika dasar dalam memasak telur adalah, masukkan telur kedalam penggorengan, mulai memasak, tunggu sampai matang, jika sudah matang hidangkan. Logika dasar tersebut bisa digambarkan dengan *flowchart* pada gambar berikut.




**Gambar 2.6 *Flowchart* Pada Logika Memasak Telur**

## 2.12 Bahasa Pemrograman

Bahasa pemrograman merupakan kumpulan aturan yang disusun sedemikian rupa sehingga memungkinkan pengguna komputer membuat program yang dapat dijalankan dengan aturan tersebut [21]. Beberapa contoh bahasa pemrograman yang banyak dipakai, yaitu C, C++, C#, Java, JavaScript, PHP, SQL, Python, Perl, dan Pascal. Bahasa Pemrograman yang digunakan pada penelitian ini adalah PHP, JavaScript dan SQL.

### 2.12.1 PHP

PHP (*Personal Home Page*) merupakan *script* untuk pemrograman *web server-side*, *script* yang membuat dokumen HTML secara *on the fly*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat

dengan menggunakan teks *editor* atau *editor* HTML. Dengan menggunakan PHP, maintanance suatu *situs web* menjadi lebih mudah. Proses *update* data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan *script* PHP. PHP/FI merupakan nama awal dari PHP, dibuat pertama kali oleh Rasmus Lerdoff. PHP awalnya merupakan program CGI yang dikhususkan untuk menerima *input* melalui *form* yang ditampilkan dalam *browser web*. *Software* ini disebarkan dan dilisensikan sebagai perangkat lunak *Open Source*. Adapun keunggulan yang dimiliki oleh PHP adalah:

1. *Life Cycle* yang sangat singkat, sehingga PHP selalu *up to date* mengikuti perkembangan teknologi internet.
2. *Cross Platform*, yakni PHP dapat dipakai di hampir semua *webserver* yang ada di pasaran (terutama Apache dan Microsoft IIS) dan dijalankan pada berbagai sistem operasi (Linux, Windows, FreeBSD).
3. PHP mendukung koneksi ke banyak *database* baik yang gratis maupun komersil, seperti *MySQL, mSQL, Oracle, Microsoft SQL Server, Interbase*, dan banyak lagi.
4. PHP bersifat *open source* dan gratis.

*Prototype* yang dibangun pada penelitian ini menggunakan PHP versi 5.6.8 yang dirilis pada 16 April 2015.

### 2.12.2 JavaScript (AJAX & jQuery)

HTTP (*Hypertext Transfer protocol*) merupakan protokol utama dibalik WWW (*World Wide Web*). Protokol ini yang menangani semua permintaan halaman website, gambar, link, dan *form* [22]. Dengan adanya HTTP memungkinkan sebuah cara yang terstandarisasi untuk komputer berkomunikasi dengan komputer lainnya. HTTP akan menspesifikasikan bagaimana klien meminta data, dan bagaimana *server* memproses permintaan tersebut.

HTTP memiliki tiga tipe permintaan dasar : POST, GET, dan HEAD. Permintaan HEAD digunakan untuk memverifikasi bahwa sebuah halaman web

yang diminta masih tersedia. Sebuah mesin pencari biasanya menggunakan permintaan HEAD untuk memastikan bahwa hasil pencariannya masih aktif. Jika permintaan HEAD gagal, maka mesin pencari tersebut bisa langsung menghapus hasil pencarian dari database yang dimiliki oleh mesin pencari tersebut. [23]

Permintaan GET merupakan permintaan yang paling sering dipakai. Pada saat sebuah *web browser* meminta sebuah halaman web, permintaan tersebut dilakukan menggunakan GET [23]. Dalam sebuah permintaan GET kita bisa menyertakan informasi tambahan yang dibutuhkan, tetapi informasi tersebut terbatas akan informasi-informasi yang berupa text saja. Informasi tersebut juga akan muncul di alamat dari sebuah halaman web.

Permintaan POST mirip dengan permintaan GET, perbedaannya adalah, dalam permintaan POST, kita bisa menyertakan informasi tambahan yang dibutuhkan yang tidak mampu ditangani oleh GET, seperti *file* [23]. Dalam melakukan permintaan POST, informasi yang disertakan tidak akan muncul pada alamat dari sebuah halaman web, melainkan akan disimpan dan hanya bisa muncul menggunakan metode tertentu pada bahasa pemrograman web yang bersangkutan.

Javascript merupakan salah satu bahasa pemrograman yang pertama kali dikembangkan oleh Netscape yang digunakan dalam sebuah *browser* atau perambah web [24]. Javascript dibuat agar mudah diintegrasikan ke dalam program dan aplikasi berbasiskan web.

AJAX, yang merupakan singkatan dari *Asynchronous Javascript and XML* merupakan bagian dari javascript yang bertugas atau berfungsi untuk memodifikasi sebagian atau semua dari sebuah halaman web secara dinamis pada saat dibutuhkan saja [25]. Dengan menggunakan AJAX kita tidak perlu me-*refresh* atau memuat ulang sebuah halaman web pada saat konten atau isi dari halaman tersebut berubah. AJAX mampu merubah sebagian komponen dari sebuah halaman web tanpa perlu memuat ulang halaman tersebut.

jQuery merupakan sebuah *library* javascript yang didesain untuk memudahkan dalam pembangunan perangkat lunak atau aplikasi berbasiskan web yang membutuhkan javascript [26].

jQuery digunakan dalam penelitian ini untuk memudahkan proses AJAX dalam memproses permintaan POST dan GET pada saat pengguna memasukkan pertanyaan. Dengan menggunakan jQuery AJAX, data pertanyaan dan jawaban bisa langsung ditampilkan tanpa harus memuat ulang sebuah halaman web. jQuery yang dipakai untuk membangun *prototype* pada penelitian ini adalah jQuery versi 1.11.13.

### 2.12.3 SQL (*Structured Query Language*)

Menurut Alan Beaulieu, pada bukunya yang berjudul *Learning SQL* (2009), SQL adalah sebuah bahasa untuk mengambil, menambahkan, dan memanipulasi data dari sebuah basis data relasional [27]. Bahasa SQL sendiri dibagi ke beberapa kategori pernyataan atau *statements*. Kategori pertama adalah pernyataan atau *statements* untuk membuat objek basis data, seperti tabel, *index*, *constraints*, dan sebagainya yang biasa disebut dengan *schema statements*. Kategori kedua merupakan pernyataan atau *statements* yang digunakan untuk membuat, memanipulasi, dan mengambil data yang tersimpan dalam basis data. Kategori kedua ini biasa disebut dengan SQL *data statements*. Biasanya, seseorang yang bekerja sebagai *programmer* atau penulis laporan hanya perlu menggunakan (atau hanya diberi akses) *data statements* saja, karena yang perlu dilakukan hanyalah berhubungan dengan data yang tersimpan pada sebuah basis data, bukan struktur basis data itu sendiri [27].

Contoh bahasa SQL yang banyak digunakan bisa dilihat pada tabel berikut.

**Tabel 2.6 Contoh Contoh Structured Query Language**

| |
|---|
| **USE namadatabase** |
| **SELECT * FROM namatabel** |
| **DELETE FROM namatabel WHERE namakolom = parameter** |

**UPDATE namatabel SET namakolom = parameter_baru**

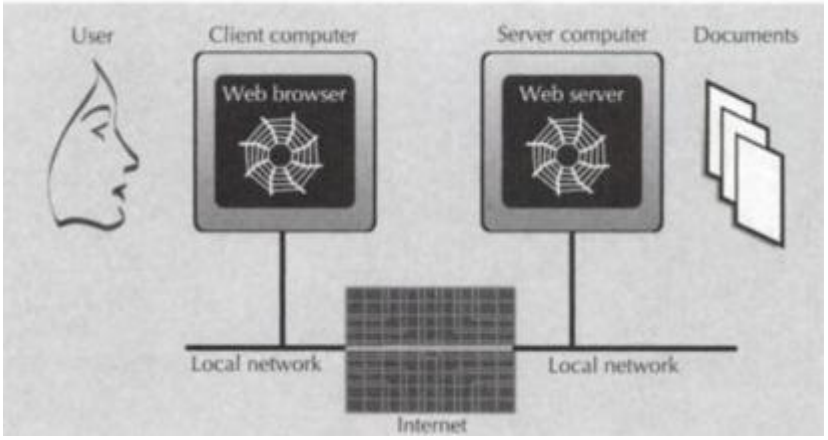
Bahasa SQL banyak diadopsi oleh beberapa aplikasi basis data relational, seperti MySQL, Microsoft SQL Server, PostgreSQL, dan SQLite.

## 2.13 Perangkat Lunak

Perangkat lunak adalah istilah yang digunakan untuk menggambarkan instruksi-instruksi yang memberitahu perangkat keras untuk melakukan suatu tugas sesuai dengan perintah [28]. Perangkat lunak yang digunakan dalam pembangunan sistem pada penelitian ini adalah *Web Server* Apache, *Web Browser* Google Chrome, dan MySQL.

### 2.13.1 *Web Server* Apache

Sebuah *web server* adalah perangkat lunak yang menawarkan pelayanan yang bisa didapatkan oleh seluruh jaringan melalui protokol HTTP. Dengan adanya *web server* memungkinkan sebuah komputer untuk menyediakan dokumen atau halaman web yang bisa diakses melalui *web browser* [29]. Skema permintaan dokumen dari *web browser* ke *web server* bisa dilihat pada gambar berikut.




**Gambar 2.7 Skema Web Server**

Pada saat sekarang ini ada banyak jenis-jenis *web server* yang tersedia, seperti nginx, Litespeed, Apache dan IIS. Salah satu yang banyak digunakan adalah Apache.

Apache merupakan salah satu jenis *web server* yang didistribusikan di bawah lisensi *open source*. Apache mampu berjalan di berbagai macam sistem operasi yang berbasiskan UNIX dan Windows, seperti Linux, Windows XP, Windows 7, Windows 8.1 [30]. Berikut adalah keunggulan dari *web server* Apache :

1. Apache mampu menangani berbagai macam kebutuhan web, mulai dari web pribadi yang kecil sampai web kelas *enterprise*.
2. Apache gratis dan *open source*, untuk bisa menggunakannya, pengguna hanya perlu mengunjungi halaman http://www.apache.org dan mengunduh tipe Apache yang sesuai dengan sistem operasi yang dipakai pengguna.

Komunitas *open source* Apache sangat banyak, sehingga mempercepat waktu perbaikan apabila terjadi *bugs* atau *error* pada aplikasi Apache tersebut. Penelitian ini menggunakan Apache versi 2.4.4.

### 2.13.2 *Web Browser* Google Chrome

*Web browser* adalah perangkat lunak yang dijalankan pada komputer pengguna yang meminta informasi dari sebuah *web server* dan menampilkannya pada komputer pengguna. Banyak sekali *web browser* yang tersedia yang bisa langsung digunakan, seperti Internet Explorer, Mozilla Firefox, Opera, Safari, dan Google Chrome. Penelitian ini menggunakan *web browser* Google Chrome versi 47.

### 2.13.3 MySQL

MySQL merupakan salah satu dari sekian banyak Database Server. Database Server merupakan sebuah perangkat lunak bertugas untuk menyediakan tempat penyimpanan basis data, serta melayani permintaan informasi dari basis data tersebut. Beberapa jenis *database* server lain diantaranya, PostgreSQL, Microsoft SQL Server, Oracle, dan banyak lagi.

MySQL adalah *Relational Database Management System* (RDBMS) atau sistem manajemen basis data relasional yang didistribusikan secara gratis di bawah lisensi GPL (General Public License). Lisensi GPL ini memungkinkan semua orang untuk bebas menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial [14].

Sebagai salah satu *database* server, MySQL harus mempunyai beberapa kemampuan, salah satunya harus menyediakan suatu sistem manajemen basis data yang dapat mengatur bagaimana menyimpan, menambah, mengakses dan melakukan transaksi transaksi pada sebuah basis data.

Perintah-perintah yang dijalankan dalam MySQL menggunakan bentuk SQL (*Structured Query Language*). SQL adalah sebuah bahasa komputer yang digunakan untuk menyimpan, memanipulasi, dan mengambil data yang tersimpan dalam basis data relasional [15]. SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah.

Beberapa keunggulan atau keistimewaaan MySQL diantaranya :

1. Portabilitas

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

2. *Open Source*

   MySQL didistribusikan secara *open source,* di bawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. Multiuser

   MySQL dapat digunakan oleh beberapa pengguna dalam waktu bersamaan tanpa mengalami masalah konflik

4. Performance Tuning

   MySQL mampu memproses lebih banyak *query* SQL per satuan waktu.

5. Jenis Kolom

   MySQL memiliki tipe kolom yang sangat kompleks, seperti integer, float, double, char, text, date, datetime, dan lain lain.

MySQL yang digunakan dalam pembangunan *prototype* pada penelitian ini adalah MySQL versi 5.5.32 (*Community Server*).