

## BAB II

### Tinjauan Pustaka dan Dasar Teori

#### 2.1. Tinjauan Pustaka

Tinjauan pustaka yang dipakai dalam penelitian ini didapat dari penelitian yang dilakukan oleh Asmorowati (2019) dalam penelitian tersebut membuat sistem informasi pemesanan dan produksi pada restoran mediternea Yogyakarta yang berbasis web dengan bahasa pemrograman *PHP* dan menggunakan database *MySql*. Sistemnya dapat memberikan informasi mengenai menu makanan dan minuman secara *online*, pemesanannya dapat dilakukan secara *online*, dan sistemnya mampu mengalkulasi bahan baku yang dibutuhkan saat memproduksi makanan dan minuman.

Diky (2018) dalam penelitiannya membuat aplikasi pemesanan makanan dan minuman dengan implementasi teknologi web untuk *push notification* pada sebuah restoran memanfaatkan teknologi web, *Java Server Side* dengan *node.js* untuk memberikan sumber daya yang mendukung implementasi *Real-Time System* melalui modul *Socket.io* untuk *push notification*.

Sappi (2018) dalam penelitiannya mengimplementasikan *AngularJs* membuat aplikasi kasir berbasis web pada sebuah kedai kopi. Pengimplementasian *AngularJs* akan ada pada bagian CRUD dan proses transaksi penjualan.

Debbie dan Lestari (2017) pada jurnalnya melakukan penelitian tentang membuat aplikasi pemesanan makanan dan minuman menggunakan *mobile web browser* dengan memanfaatkan wifi yang nantinya digunakan untuk masuk ke

halaman aplikasi. Bahasa pemrograman yang digunakan adalah *PHP* dan menggunakan *MYSQL* sebagai basis datanya.

Abdurrahman (2019) pada jurnalnya melakukan penelitian tentang aplikasi pemesanan makanan dan minuman berbasis *web responsive* dengan bahasa pemrograman *PHP*. Dengan fitur yang memanfaatkan teknologi *AJAX* pada bagian status pemberitahuan yang ditujukan ke bagian dapur saat ada pesanan yang masuk.

Dimas (2020) dalam usulan penelitiannya memiliki kesamaan pada membuat aplikasi untuk pemesanan makanan dan minuman. Perbedaan dengan penelitian yang ada diatas adalah pada penelitian ini lebih menekankan pada proses pengelolaan pemesanan makanan dan minuman menggunakan teknologi *AJAX* yang nantinya pesanan tersebut akan menampilkan pesanan apa saja yang dipesan ke bagian dapur lalu saat pesanan selesai sampai dengan ke proses pembayaran masih menggunakan teknologi *AJAX* berbeda dengan penelitian sebelumnya yang hanya menyampaikan pesanan saja kepada bagian dapur.

**Table 1.1 Perbandingan Tinjauan Pustaka**

| <b>Penulis</b>            | <b>Objek</b>  | <b>Metode/Teknologi</b> | <b>Jenis Aplikasi</b> |
|---------------------------|---|-------------------------|-----------------------|
| Asmorowati (2019)         | Sistem Informasi Pemesanan Dan Produksi pada Restoran Mediternea Yogyakarta | PHP, MySQL              | Web                   |
| Diky Iwan Setiawan (2018) | Rumah Makan   | Javascript, PHP, MySQL  | Web Mobile            |

**Tabel 2.1 (Lanjutan)**

|                           |   |                  |            |
|---------------------------|---|------------------|------------|
| Sappi (2018)              | Implementasi AngularJs pada Aplikasi Kasir                            | PHP, Javascript  | Web        |
| Debbie dan Lestari (2017) | Aplikasi Pemesanan Makanan Dan Minuman Online                         | PHP, MySQL       | Web Mobile |
| Abdurrahman (2019)        | Implementasi Web Responsive Pemesanan Makanan Dan Minuman Di Restoran | PHP, AJAX        | Web        |
| Dimas (2020)              | Aplikasi Pengelolaan Pemesanan Makanan dan Minuman                    | PHP, AJAX, MySQL | Web        |

## 2.2. Dasar Teori

Pada penelitian ini terdapat beberapa tinjauan pustaka sebagai landasan teori dalam pengembangan Implementasi Ajax pada Aplikasi Pengelolaan Pemesanan Makanan dan Minuman (Studi Kasus:Kedai OPSI).

### 1) Coffe Shop (Kedai Kopi )

Menurut Marsum (2001), terdapat *twenty two* (22) jenis tipe restoran, yaitu *a'la carte restaurant, tabel d'hote restaurant, coffe shop atau brasserie, cafeteria/cafe, canteen, continental restaurant, carvery, dining room, dischoteque, fish and chip shop, grill room (rotisserie), inn tavern, night club/super club, pizzeria, pan cake house/creperie, pub, snack bar/cafe/milk bar, specialty*

*restaurant, terrace restaurant, gourmet restaurant, family type restaurant, dan main dining room. Coffee shop* merupakan salah satu dari kedua puluh dua tipe restoran di atas yang menyediakan tempat untuk menikmati kopi dan menikmati makanan ringan sebagai pendamping kopi tersebut, orang-orang yang datang ke *coffee shop* ini memiliki berbagai tujuan seperti berbicara santai bersama teman, rapat, atau bahkan mengerjakan tugas.

## 2) **Pemesanan**

Menurut Melly Soedirman (2012:20), dalam jurnal Andreas Susanto (2013) mendefinisikan : “Pemesanan adalah keseluruhan proses kegiatan dan memiliki *rule* yang berkaitan dengan pengelolaan inventory atau persediaan tempat pendistribusian produk dan catatan keseluruhan transaksi. Dari pendapat ahli tersebut dapat disimpulkan bahwa pemesanan adalah suatu perjanjian pemesanan yang dilakukan oleh *two* (dua) pihak atau lebih yaitu pemilik sebagai pemberi dan pelanggan sebagai pemakai jasa atau barang untuk memenuhi kebutuhannya berupa barang atau jasa tersebut sehingga terbentuk perjanjian pemesanan tersebut berupa atas pemesanan ruangan, tempat duduk, kamar dan lainnya pada periode waktu tertentu dan akan mendapatkan bukti perjanjian berupa catatan transaksi.

## 3) **PHP (*Hypertext Preprocessor*)**

PHP ( PHP :HyperText Preprocessor ), merupakan bahasa web *script* yang bekerja di *server-side* dan bersifat *embedded* script pada *HTML*. Sebagai pemrogram C, Rasmus Lerdorf menciptakan PHP pada tahun 1995. Dengan versi 1.0 yaitu *Personal Home Page Tools* yang dibuat Rasmus dapat digunakan gratis.

*PHP* versi 2 dikeluarkan tahun 1996 dengan diberi nama *PHP/FI*. *PHP* versi 2 ini sudah memiliki dukungan untuk mengkoneksikannya ke database dan sudah terintegrasi dengan *HTML*. Sedangkan Tanggal 6 Juni 1998 *PHP* 3.0 resmi dirilis lalu 2 tahun pada tanggal 22 Mei 2000 kemudian *PHP* versi 4 yang terbit.

Didalam *PHP* terdapat fungsi built in dan beberapa *library* seperti fungsi *support* (pendukung) yang terintegrasi dengan *extensible Markup Language* (XML).

Selain itu, *PHP* dapat digunakan secara gratis dan mendukung banyak database antara lain : *Adabas D, dBase, Empress, FilePro, FrontBase, Hyperwave, IBM DB2, Informix, Ingres, Interbase, MSOL, Direct MS SQL, MySQL, ODBC, Oracle, Ovrimos, PostgreeSQL, Solid, Sybase, Velocis, Unix DBM*. *PHP* bersifat cross platform, artinya dapat berjalan pada sistem operasi yang berbeda seperti pada *Windows 98, UNIX, dan Macintosh*. *PHP* juga mendukung untuk berkomunikasi dengan layanan lain menggunakan protokol *IMAP, SNMP, NNTP, POP3, HTTP*, dan lainnya (Abdul Kadir,2003).

Terdapat *Main Rule* pada penulisan *script* *PHP* antara lain :

- Diawali `<?>` dan ditutup dengan `?>`
- Variabel pada *PHP* memiliki sifat case sensitive, ini akan membuat perbedaan besar pada huruf kapital dan huruf kecil.
- *PHP* bersifat embedded script dalam dokumen *HTML* atau *non embedded script* pada dokumen php terpisah.

#### 4) MySQL

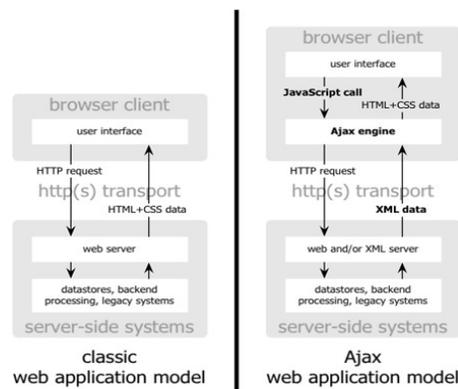
*Software* sistem manajemen database (*Database Management System – DBMS*) yang sangat populer pada kalangan pemrogram web yaitu *MySQL*, terutama dilingkungan *linux* dengan menggunakan script *PHP* dan *Perl*. *Software database* ini kini telah tersedia juga pada platform sistem operasi *windows* (Sianipar, 2015).

#### 5) AJAX

*AJAX* akronim dari *Asynchronous JavaScript and XML*. *AJAX* bukanlah suatu teknologi yang berdiri sendiri, tetapi suatu kombinasi yang mengacu pada penggunaan beberapa teknologi secara bersama – sama. Jadi *AJAX* dapat diartikan sebagai “*Asynchronous JavaScript + CSS + DOM + XMLHttpRequest*” Sebagai contoh dapat dilihat pada *Messenger* milik *Facebook*, mengacu pada text balasan pada pesan dalam *messenger*. Jika ada terdapat pesan dari teman atau akun lain yang baru masuk maka akan muncul text baru dan bertambah pada tanpa *refresh* halaman tersebut. Contoh lain ada pada *Gmail* yang menggunakannya pada bagian email di *inbox* mereka menggunakan teknologi *XMLHttpRequest*. *XMLHttpRequest* Adalah sebuah komponen teknis yang membuat adanya komunikasi secara *asynchronous* yang terjadi pada *client* dan *server*. Terdapat fitur pada *XMLHttpRequest* yang berasal dari *browser - browser* baru tetapi tidak semua *browser* baru yang digunakan untuk melakukan *request HTTP*, *JavaScript* digunakan untuk mengakses fitur tersebut. Halaman pada *gmail* yang memuat jumlah *inbox* akan

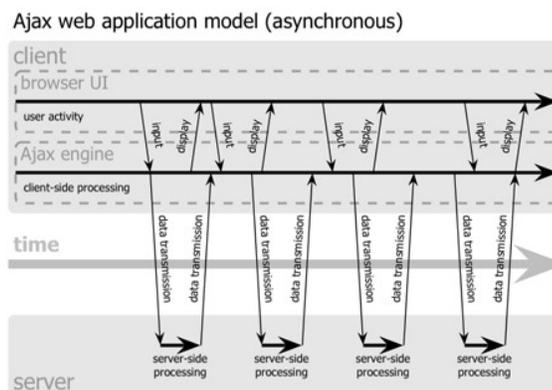
melakukan *request* pada server dan mengupdate nilainya di halaman web dalam waktu tertentu atau sesekali. Maka halaman tersebut tidak melakukan *request* ulang, tapi hanya *request* data yang terjadi di belakang layar. Pada model aplikasi *web* klasik proses kerjanya adalah : *client* mengirimkan *HTTP request* untuk meminta respons dari *server*, *server* melakukan beberapa proses seperti : menerima data, mengolahnya di *database*, dan *server web* merespon dengan mengirimkan sebuah halaman *web* yang baru yang berisi informasi yang dibutuhkan oleh *client*. Proses suatu aplikasi *web* dengan model *AJAX* adalah : ketika aplikasi *di load* yang dilakukan adalah *meload AJAX engine*. *AJAX Engine* dibuat dengan javascript yang fungsinya untuk *render* tampilan untuk *user* dan dan berkomunikasi dengan *server*. Dengan begitu komunikasi ke *server* bisa terjadi secara *asynchronous* (Ballard, Phil. 2006.).

Dengan demikian *AJAX Engine* mengupdate tampilan secara otomatis pada *client* yang sumbernya berasal dari *server*. Untuk melihat perbandingannya antara proses yang menggunakan aplikasi *web* klasik dan proses yang menggunakan *AJAX* ada pada gambar 2.1.

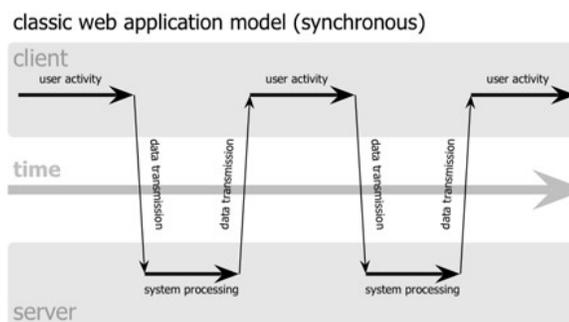


**Gambar 2.1 Perbandingan cara kerja Web Klasik dan Ajax**

Perbedaan antara aplikasi *web* berbasis *AJAX* dengan *web* klasik adalah, *AJAX* mengeliminasi *start-stop-start-stop* pada interaksi *client* dengan *server*. Akan tetapi dengan menggunakan *AJAX engine*, komunikasi yang terjadi antara *client* dan *server* dapat berlangsung terus menerus secara *asynchronous*, dengan dilakukannya hal tersebut maka akan terlihat halaman *web* yang tidak perlu melakukan proses *reload* karena seperti memasukan *layer* baru pada halaman *web*. Pada gambar 2.2 secara *asynchronous* dan gambar 2.3 secara *synchronous* dapat dilihat perbedaan komunikasi antara *client* dan *server* antara *web* klasik dan *web* berbasis *AJAX* :



**Gambar 2.2 Perbandingan Transfer Data Asynchronous**



**Gambar 2 3 Perbandingan Transfer Data Synchronous**