BAB II
TINJAUAN PUSTAKA

2.1 Aplikasi

Aplikasi adalah suatu program komputer yang bertujuan untuk mengerjakan tugas dari *user* (Habibi, 2020). Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna (Sari, 2021). Aplikasi adalah suatu perangkat lunak (*software*) atau program komputer yang beroperasi pada sistem tertentu yang diciptakan dan dikembangkan untuk melakukan perintah tertentu (Pane dkk, 2020).

Aplikasi adalah sebuah program siap pakai yang bisa dipakai untuk menjalankan sejumlah perintah dari pengguna aplikasi itu sendiri (Habibi dan Karnovi, 2020). Aplikasi merupakan suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Aplikasi dibuat untuk memudahkan pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan, dan penambahan data yang dibutuhkan. (Setyawan dan Munari, 2020).

2.1.1 Pengertian Aplikasi Berbasis Website

Aplikasi berbasis *web* ini menggunakan protokol HTTP, aplikasi di sisi server berkomunikasi dengan *client* melalui *web* server. Aplikasi di sisi *client* umumnya berupa *web browser*. Jadi, aplikasi berbasis *web* (*client/server-side script*) berjalan diatas aplikasi berbasis internet (Setyawan dan Munari, 2020).

2.1.2 Tujuan Aplikasi Berbasis Website

Tujuan aplikasi berbasis *web* menurut (Setyawan dan Munari, 2020) yaitu :

1. Aplikasi berbasis *web* dapat digunakan untuk membantu operasional perusahaan seperti *invoice*, sistem informasi persediaan.
2. Memudahkan dalam penyimpanan data di *database*.
3. Aplikasi berbasis *web* juga dapat bekerja me-*monitoring* sistem dalam hal tampilan, dapat didesain dan disesuaikan untuk berbagai jenis industri

2.1.3 Karakteristik Aplikasi Berbasis Website

Karakteristik aplikasi berbasis *web* menurut (Setyawan dan Munari, 2020) antara lain:

1. Kepadatan Jaringan Aplikasi *web* umumnya berada pada suatu jaringan komputer dan harus bisa melayani kebutuhan komunitas klien yang beragam.
2. Keserempakan Sejumlah besar pengguna mungkin akan mengakses aplikasi secara serempak.
3. Jumlah pengguna yang tidak dapat diprediksi Jumlah pengguna mungkin beragam dari hari ke hari. Pengguna di hari libur mungkin berada di hari kerja.
4. Kinerja Penggunaan aplikasi berbasis *web* akan pergi jika pengguna terlalu lama menunggu aplikasi *web*.
5. Ketersediaan Menyediakan akses layanan 24 jam.
6. Digerakkan oleh data Beberapa aplikasi *web* bergantung pada layanan data, contoh: *online shop*.
7. Peka terhadap isi Kualitas isi dan keindahan tetap merupakan faktor penting kualitas *web*.
8. Evolusi yang berkesinambungan Merupakan hal yang biasa untuk aplikasi *web* jika ada pembaharuan setiap saat.
9. Keamanan Terhubung ke internet, menyebabkan aplikasi *web* rentan serangan dari pihak lain.
10. Estetika Salah satu daya tarik aplikasi *web* adalah tampilan dan nuansanya.
11. Kesegeraan Aplikasi *web* sering memerlukan kedisiplinan waktu untuk merilis produk ke pasar dalam waktu beberapa hari atau beberapa minggu.

2.2 Pasang Baru

Sebagai perusahaan yang menyelenggarakan pelayanan publik PDAM senantiasa memberikan pelayanan terbaiknya kepada para pelanggan. Salah satu layanan adalah memenuhi permintaan pelanggan untuk melakukan penyambungan

pelanggan baru setelah calon pelanggan tersebut memenuhi segala persyaratan administrasi dan keuangan yang ditetapkan (Santoso dkk, 2020).

Pasang baru PDAM adalah istilah yang merujuk pada proses pemasangan jaringan pipa baru oleh PDAM untuk menyediakan pasokan air bersih ke daerah atau bangunan yang sebelumnya belum terhubung dengan jaringan PDAM. Pemasangan ini dapat mencakup pembuatan saluran pipa baru, pemasangan meteran, dan pemasangan peralatan pendukung lainnya (Budi, 2019).

2.3 Website

Website atau biasa disebut *web* adalah kumpulan dari halaman-halaman situs yang terangkum dalam sebuah domain atau *subdomain* yang berada didalam *World Wide Web* (WWW) yang diakses melalui jaringan internet (Roza dkk, 2020). *Website* merupakan sekumpulan halaman pada suatu domain di internet yang dibuat dengan tujuan tertentu dan saling berhubungan serta dapat diakses secara luas melalui halaman depan menggunakan sebuah *browser* menggunakan URL *website* (Mulyani, 2020).

Website adalah halaman informasi yang disediakan melalui jalur internet dan memiliki fasilitas hiperteks untuk menampilkan data berupa teks, gambar, suara, animasi, dan data multimedia lain nya sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. *Website* dapat diartikan sekumpulan halaman situs yang saling berhubungan menampilkan berbagai macam informasi maupun data berupa teks, gambar diam ataupun bergerak, data animasi, suara, video maupun gabungan dari semuanya yang bisa diakses dimana saja kapan saja diseluruh dunia (Putra dan Eka, 2023).

2.3.1 Website Statis

Website Statis adalah *website* dimana pengguna tidak dapat merubah konten dari *web* tersebut secara langsung menggunakan *browser*. Interaksi yang terjadi antara pengguna dan server hanyalah seputar pemrosesan link saja. Halaman – halaman *web* tersebut tidak memiliki *database*, data dan informasi yang ada pada *web* statis tidak berubah – ubah kecuali diubah sintaksnya. Dokumen *web* yang

dikirim *client* akan sama isinya seperti apa yang ada di *web* server (Setijabudi, 2022).

Website statis adalah *website* yang berisi/menampilkan informasi-informasi yang sifatnya statis (tetap). Disebut statis karena pengguna tidak dapat berinteraksi dengan *web* tersebut. Pengguna hanya dapat melihat isi dokumen pada halaman *web* dan apabila diklik akan berpindah kehalaman *web* yang lain. Interaksi pengguna terbatas hanya melihat informasi yang ditampilkan, tetapi tidak bisa mengolah informasi yang dihasilkan. Biasanya merupakan *HTML* yang ditulis pada editor teks dan disimpan dalam bentuk .html atau .htm (Putra dan Eka, 2023).

2.3.2 Website Dinamis

Website dinamis adalah *website* yang menampilkan informasi serta dapat berinterksi dengan pengguna. *Web* yang dinamis memungkinkan pengguna untuk berinteraksi dengan *form* sehingga dapat mengolah informasi yang ditampilkan. *Website* dinamis bersifat teraktif, tidak kaku, dan terlihat lebih indah (Putra dan Eka, 2023).

Dalam *website* dinamis, interaksi yang terjadi antara pengguna dan server sangat kompleks. Pengguna *website* dapat mengubah konten dari halaman tertentu dengan menggunakan *browser*. Permintaan dari pengguna dapat diproses oleh server yang kemudian ditampilkan dalam isi yang berbeda – beda menurut alur programnya. Halaman – halaman *web* tersebut memiliki *database*. *Website* dinamis memiliki data dan informasi yang berbeda – beda tergantung *input* yang disampaikan *client*. Dokumen yang sampai di client akan berbeda dengan dokumen yang ada di *web* server (Setijabudi, 2022).

Perbedaan *web* statis dan *web* dinamis:

a. Interaksi antar pengunjung dan pemilik *web* dalam *web* statis tidak dimungkinkan terjadinya interaksi antar pengunjung dengan pemilik *web*. Sementara dalam *web* dinamis terdapat interaksi antar pengunjung dengan pemilik *web*
b. *Web* statis hanya menggunakan satu bahasa *script* yaitu HTML, atau setidaknya ditambahkan dengan CSS. Sedangkan *web* dinamis

menggunakan bahasa pemrograman *web* yang lebih kompleks seperti PHP, ASP, serta *Javascript*.

c. *Website* statis tidak menggunakan *database* karena tidak ada data untuk disimpan dan diproses. Sedangkan *web* dinamis menggunakan *database* seperti MYSQL Oracle dll. untuk menyimpan dan mengolah data.
d. Konten *web* statis hanya disediakan oleh pemilik situs *web* dan jarang di *update*, sedangkan konten di *web* dinamis dapat berasal dari pengunjung dan lebih sering di *update*. Kemudian konten di *web* dinamis dapat diambil dari *database* sehingga konten mungkin berbeda meskipun kita membuka halaman *web* yang sama.

2.4 HTML (Hypertext Markup Language)

Hypertext Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi dalam sebuah penjelajah *web* internet dan memformat hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi (Prastya, 2022).

Yang bisa dilakukan dengan HTML yaitu:

1. Mengatur tampilan dan halaman *web* dan isinya.
2. Membuat tabel dalam halaman *web*.
3. Mempublikasikan dalam halaman *web* secara *online*.
4. Membuat *form* yang bisa digunakan untuk menangani registrasi dan transaksi via *web*. Contoh: Setiap dokumen HTML diawali dan diakhiri dengan *tag* HTML.

Berikut ini merupakan *syntax* pemrograman sederhana dari HTML, seperti pada Gambar 2.1.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Document</title>
    </head>
    <body>
        <!-- Tulis disini -->

    </body>
</html>
```

Gambar 2.1 Tampilan *Syntax* Dasar HTML

2.5 PHP (Hypertext Preprocessor)

PHP merupakan program *Server Side Scripting* yaitu program yang dapat di kompilasi atau diterjemahkan ke dalam server, sehingga dapat menghasilkan aplikasi *web* dinamis (Adi, 2020). PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs *web* dan bisa digunakan Bersama dengan HTML (Mulyani, 2020). PHP merupakan sebuah bahasa *scripting* (kode untuk membangun suatu program) yang dikombinasikan pada *HyperText Markup Language* (HTML) (Fauzan dan Dwitya, 2019).

Bahasa pemrograman *PHP* (*PHP Hypertext Preprocessor*) adalah bahasa pemrograman yang bekerja dalam sebuah *web* server. *Script-script PHP* harus tersimpan dalam sebuah server dan dieksekusi atau proses dalam server tersebut. Dengan menggunakan program *PHP*, sebuah *website* akan lebih interaktif dan dinamis (Nirsal, Rusmala, 2020). Berikut merupakan logo dari PHP seperti yang ditunjukkan pada Gambar 2.2.

Gambar 2.2 Tampilan Logo PHP

Menurut (Sahi, 2020), Kelebihan dari *PHP*, yaitu:

1. Bahasa pemrograman *PHP* adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunanya.
2. *Web server* yang mendukung *PHP* dapat ditemukan dimana-mana dari mulai *apache, IIS, Lightpd, nginx,* hingga *Xitami* dengan konfigurasi lebih mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu pengembangan.
4. Dalam sisi pemahaman, *PHP* adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. *PHP* adalah bahasa open source yang dapat digunakan dibeberapa mesin (*Linux, Unix, Macintosh, Windows*) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah *system.*

2.6 CSS (Cascading Style Sheet)

CSS (*Cascading Style Sheet*) adalah salah satu bahasa desain *web (Style sheet language)* yang mengontrol format tampilan sebuah halaman *web* yang ditulis dengan menggunakan penanda *markup laguage*. Biasanya CSS digunakan untuk mendesain sebuah halaman HTML dan XHTML, tetapi sekarang CSS bisa diaplikasikan untuk segala dokumen XML, termasuk SVG dan XUL bahkan android (Hakim, 2019).

CSS atau *Cascading Style Sheet* secara sederhana adalah sebuah metode yang digunakan untuk mempersingkat penulisan *tag* HTML seperti *font, color, text* dan tabel menjadi lebih singkat. CSS digunakan untuk mengatur tampilan dokumen. CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda (Rusyana, 2019). CSS mempunyai beberapa kegunaan dan manfaat antara lain:

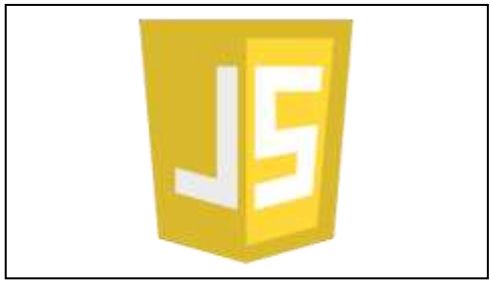
1. Mempersingkat penulisan *tag* HTML. Tidak perlu lagi mendifinisikan setiap *tag* dengan *property* dan nilai yang sama.
2. Mempercepat proses *rendering* atau pembacaan HTML karena tidak terdapat pengulangan penulisan.

3. Mudah dan Cepat dalam *maintenance* dikarenakan *file* CSS yang dibuat secara terpisah. Apabila inign melakukan perubahan cukup merubah satu *file* tanpa merombah seluruh *file* HTML.
4. CSS bisa melakukan apa yang tidak bisa dilakukan oleh HTML, misalnya kita bisa memberikan warna pada *input box* atau *scrollbar.*

2.7 Javascript

Javascript dikembangkan oleh Netscape Communications yang bekerja sama dengan *Sun Microsystem*. Sebenarnya *javascript* dikembangkan dari bahasa *livescript* yang khusus dirancang untuk *netscape navigator*. Oleh karena itu dulunya *javascript* hanya dapat dijalankan pada *browser netscape navigator*, namun dalam perkembangannya Internet Explorer yang dirilis oleh Microsoft juga dapat mendukung *javascript*. *Javascript* memang tidak secanggih *java* (karena hanya merupakan *script*) tetapi untuk pemrogramannya tidak diperlukan pemrogram profesional. *Javascript* dijalankan di *client (client side programing*) sehingga dapat mengurangi beban kerja dari server (Muhamad, 2021).

Dengan *javascript* dapat dibuat halaman *web* yang interaktif dan juga cerdas. Sebagai contoh *javascript* dapat digunakan untuk mengecek sah tidaknya masukan pengguna sebelum masukan dikirim ke server, *javascript* juga dapat melakukan operasi aritmatik (Seperti penjumlahan, pengurangan, dan perkalian), dan bisa juga menampilkan animasi sederhana. *Javascript* bukan bahasa berorientasi *object*, melainkan bahasa berbasis *object*. Bahasa berorientasi *object* harus mendukung tiga konsep dasar, yaitu pengkapsulan (encapsulation), pewarisan (*inheritance*), dan polimorfisme (*polymorphism*) (Muhamad, 2021). Berikut merupakan logo dari *Javascript* seperti yang ditunjukkan pada Gambar 2.3.



Gambar 2.3 Tampilan Logo *Javascript*

2.8 Database

Basis data terdiri dari basis dan data. Basis data dapat diartikan sebagai markas atau gudang, sedangkan basis data sendiri adalah himpunan kelompok data yang saling terhubung dan diorganisasi sedemikian rupa supaya kelak dapat dimanfaatkan kembali secara cepat dan mudah (Rachmadi, 2020). *Database* atau basis data adalah kumpulan informasi yang disimpan secara sistematis dalam komputer sehingga dapat diverifikasi oleh program komputer untuk mengambil informasi dari *database*. Penggunaan utama dari sistem *database* adalah bahwa pengguna dapat membangun tampilan data abstrak. Hal ini bertujuan untuk menyederhanakan interaksi antara pengguna dan sistem dan *database* dapat menyajikan pandangan 15 yang berbeda kepada pengguna *programmer* dan administrator (Apryliyana dan Adeswastoto, 2021).

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. *Database* adalah representasi kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan. *Database* merupakan sekumpulan informasi yang saling berkaitan pada suatu subjek tertentu pada tujuan tertentu pula. *Database* adalah susunan record data operasional lengkap dari suatu organisasi atau perusahaan, yang diorganisir dan disimpan secara terintegrasi dengan menggunakan metode tertentu dalam komputer sehingga mampu memenuhi informasi yang optimal yang dibutuhkan oleh para pengguna (Helmud, 2021).

2.9 PHPMyAdmin

PHPMyadmin adalah sebuah aplikasi open source yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan PHPMyAdmin, kita dapat membuat *database*, tabel, *menginsert*, menghapus dan *mengupdate* data dengan GUI dan terasa lebih mudah tanpa harus mengetikkan perintah SQL secara

manual. PHPMyAdmin adalah *front-end* MySQL berbasis *web*. PHPMyAdmin dibuat dengan menggunakan PHP (Nia, 2022).

PHPMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui world wide *web*. PHPMyAdmin mendukung berbagai operasi MySQL, diantaranya mengolah basis data, tabel-tabel, *fields*, relasi, indeks, *user, permissions*, dan lain-lain (Nirsal, Rusmala, 2020).

PHPMyAdmin adalah sebuah aplikasi atau perangkat berbasis opensource yang bisa kita gunakan secara gratis untuk melakukan pemrograman ataupun administrasi pada *database* MySQL. PHPMyAdmin sendiri menggunakan bahasa PHP untuk pemrogramannya, selain itu PHPMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang *(fields)*, relasi *(relations)*, indeks, pengguna (*users*), perijinan (*permissions*), dan lain lain. Jadi dapat disimpulkan pula bahwa phpMyAdmin berbeda dengan MySQL. Dimana PhpMyAdmin digunakan sebagai alat yang memudahkan dalam pengoperasian *database* MySQL, sedangkan MySQL adalah suatu *database* itu sendiri, dimana *database* berfungsi sebagai penyimpanan data (Ramadhan dan Mukhaiyar, 2020).

2.10 MySQL

MySQL adalah suatu RDBMS (*Relational Data Base Management System*), yaitu aplikasi sistem yang menjalankan fungsi pengolahan data (Sa'ad, 2020). MySQL adalah sistem manajemen *database* SQL yang sifatnya open source (terbuka) dan paling banyak digunakan saat ini. MySQL bisa dikatakan sebagai implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis (Mundzir, 2020).

MySQL adalah salah satu aplikasi DBMS (*Database Management System*) yang sudah banyak digunakan oleh para *programmer* aplikasi *web*. Dalam sistem basis data non-relasional semua informasi disimpan dalam satu bidang besar yang terkadang sulit dan memakan waktu untuk mengakses data yang dikandungnya. Tetapi MySQL adalah sistem *database* relasional sehingga dapat mengelompokkan

informasi ke dalam tabel atau kelompok informasi terkait. Setiap tabel berisi bidang terpisah mewakili setiap bit informasi. MySQL menggunakan indeks untuk mempercepat pencarian baris informasi tertentu. MySQL membutuhkan setidaknya satu indeks per tabel. Biasanya akan menggunakan kunci utama atau pengenal unik untuk membantu melacak data (Setijabudi, 2022).

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang *multithread, multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia (Rusmawan, 2019). MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multithreaded, multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia (Haqi dan Setiawan, 2019). Berikut merupakan logo dari MySQL seperti yang ditunjukkan pada Gambar 2.4.

Gambar 2.4 Tampilan Logo MySQL

2.11 XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTPServer, MySQLdatabase,* dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web* server yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. Untuk mendapatkanya dapat *mendownload* langsung dari *web* resminya (Nurrizki, 2021).

XAMPP yaitu perangkat lunak yang mendukung sistem operasi, merupakan kompilasi dari beberapa program (Harani dan Andri, 2020). XAMPP adalah sebuah

software yang menjalankan peran sebagai local *web server database*. *Software* ini juga digunakan untuk pengujian aplikasi *web* melalui *localhost*, aplikasi *web* yang dikembangkan secara *native*, *framework* maupun CMS bisa diuji menggunakan XAMPP (Jho, 2020). Berikut merupakan tampilan dari kontrol panel XAMPP seperti yang ditunjukkan pada Gambar 2.5.

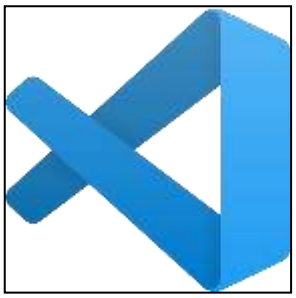
Gambar 2.5 Tampilan Kontrol Panel XAMPP

2.12 Visual Studio Code

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap *(suite)* yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi Windows, ataupun aplikasi *Web*. Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas .NET *Framework*). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi *SilverLight*, aplikasi Windows Mobile (yang berjalan di atas .NET *Compact Framework*) (Hakim, 2021).

Visual Studio kini telah menginjak versi Visual Studio 9.0.21022.08, atau dikenal dengan sebutan Microsoft Visual Studio 2008 yang diluncurkan pada 19

November 2007, yang ditujukan untuk platform Microsoft .NET *Framework* 3.5. Versi sebelumnya, Visual Studio 2005 ditujukan untuk platform .NET *Framework* 2.0 dan 3.0. Visual Studio 2003 ditujukan untuk .NET *Framework* 1.1, dan Visual Studio 2002 ditujukan untuk .NET *Framework* 1.0. Versi-versi tersebut di atas kini dikenal dengan sebutan Visual Studio .NET, karena memang membutuhkan Microsoft .NET *Framework*. Sementara itu, sebelum muncul Visual Studio .NET, terdapat Microsoft Visual Studio 6.0 (Hakim, 2021). Berikut merupakan logo dari aplikasi Visual Studio Code seperti yang ditunjukkan pada Gambar 2.6.



Gambar 2.6 Tampilan Logo Visual Studio Code

2.13 Microsoft Visio

Microsoft Office Visio atau biasa disebut dengan Microsoft Visio, merupakan sebuah *software* komputer yang biasanya digunakan untuk membuat diagram, diagram alir, *brainstorm*, dan skema jaringan. Selain Word, Excel dan PowerPoint, Microsoft Visio juga termasuk dalam paket Microsoft Office. *Software* ini menggunakan grafik vektor untuk membuat diagramnya. Microsoft Visio pertama kali dikenalkan pada tahun 1992, yang mana dibuat oleh Visio Corporation. Namun pada tahun 2000, *software* ini telah diakuisisi oleh Microsoft. Setelah itu, Microsoft Visio dapat dibedakan menjadi tiga edisi, yaitu *Standart*, *Profesional* dan *Online*. Tentu saja dari ketiga edisi tersebut memiliki fitur, kapabilitas, serta harga yang berbeda-beda. Dengan *software* ini dapat membantu pengguna dalam meningkatkan kinerja, mulai dari mempersiapkan penggambaran diagram seperti DFD, ERD, UML, Jaringan, Rancangan *User Interface* dan sejenisnya. Terlebih adanya sejumlah *template* yang disediakan, Dapat memungkinkan pengguna untuk membuat diagram dengan mudah, intuitif serta

profesional (Gustafianto, 2020). Berikut merupakan logo dari aplikasi Microsoft Visio seperti yang ditunjukkan pada Gambar 2.7.

Gambar 2.7 Tampilan Logo Microsoft Visio

2.14 Framework

Framework adalah suatu kumpulan kode berupa pustaka (*library*) dan alat (*tool*) yang dipadukan sedemikian rupa menjadi satu kerangka kerja (*framework*) guna memudahkan dan mempercepat proses pengembangan aplikasi *web*. Jadi, *Framework* adalah kumpulan-kumpulan potongan program yang dipadukan menjadi satu kerja kerja yang digunakan untuk membatu dalam pembuatan sebuah aplikasi (Wibowo, 2020).

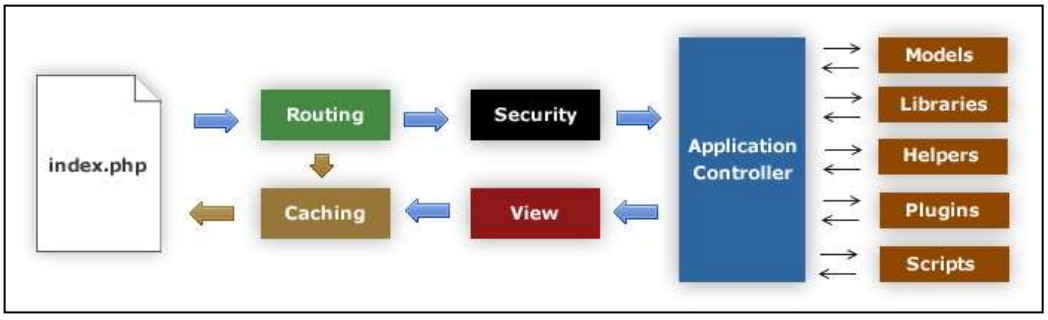
Salah satu alasan mengapa orang menggunakan *framework* terutama dalam membangun sebuah aplikasi adalah kemudahan yang ditawarkan. Didalam sebuah *framework* biasanya sudah tersedia struktur aplikasi yang baik, *standard coding*, *best practice*, *design pattern,* dan *common function*. Dengan menggunakan *framework* kita dapat langsung fokus kepada *business process* yang dihadapi tanpa harus berfikir banyak masalah struktur aplikasi, standar *coding* dan lain-lain. Sehingga, *framework* merupakan koleksi atau kumpulan potongan-potongan program yang disusun atau diorganisasikan sedemikian rupa, sehingga dapat digunakan untuk membantu membuat aplikasi utuh tanpa harus membuat semua kodenya dari awal (Wibowo, 2020).

2.15 CodeIgniter

CodeIgniter merupakan sebuah *framework* PHP yang bersifat *open source* untuk memudahkan *programmer* atau *developer* dalam membangun sebuah

aplikasi berbasis *web* tanpa harus membuatnya dari awal dengan menggunakan metode MVC (*Model, View, Controller*). *CodeIgniter* (CI) memiliki *library* yang lengkap untuk mengerjakan operasi - operasi yang umum dibutuhkan oleh aplikasi berbasis *web* misalnya mengakses *database*, memvalidasi *form* sehingga sistem yang dikembangkan mudah (Sallaby dan Kanedi, 2020).

CodeIgniter adalah sebuah *web application network* yang bersifat *open source* yang digunakan untuk membangun aplikasi php dinamis. *CodeIgniter* menjadi sebuah *framework* PHP dengan model MVC (Model, View dan Control) untuk membangun *website* dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi *web*. Selain ringan dan cepat, *CodeIgniter* juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih *CodeIgniter* sebagai *framework* pilihannya. Karena kelebihan-kelebihan yang dimiliki oleh *CodeIgniter*, pembuat PHP Rasmus Lerdorf memuji *CodeIgniter* di frOSCon dengan mengatakan bahwa dia menyukai *CodeIgniter* karena *"it is faster, lighter and the least like a framework"*. *CodeIgniter* pertama kali dikembangkan pada tahun 2006 oleh Rick Ellis, dengan logo api yang menyala (Wibowo, 2020). Berikut merupakan konsep MVC pada *CodeIgniter* seperti yang ditunjukkan pada Gambar 2.8.



Gambar 2.8 Konsep MVC pada *CodeIgniter*

1. index.php bertugas sebagai *front controller*, menginisialisasi *resource* dasar yang dibutuhkan oleh *codeigniter*.
2. *Router* memeriksa permintaan HTTP untuk menentukan apa yang seharusnya dikerjakan dengan permintaan itu.

3. Jika di dalam *cache* tersedia, maka akan diteruskan kedalam *browser*, tanpa harus melewati arus eksekusi yang normal.
4. *Security*. sebelum *application controler* di jalankan, permintaan HTTP dan semua data *user* yang di *submit* akan di filter untuk keamanan.
5. *Controller* akan menjalankan model, *core libraries*, *helpers* dan *resource* yang lainnya yang dibutuhkan untuk diproses permintaan yang spesifik.
6. Akhirnya *View* akan memproses apa yang akan ditampilkan di *web browser*.
7. Jika *caching* aktif, tampilan akan di simpan di *cache* terlebih dahulu kemudian urutan *request* akan di sediakan untuk *web browser*

Berdasarkan tinjauan pustaka diatas, hal tersebutlah yang mendasari penulis memilih *framework CodeIgniter* dalam membangun aplikasi berbasis *web* dikarenakan *framework CodeIgniter* cenderung mudah dipelajari, dapat dimodifikasi, dan fleksibel. Dalam membangun aplikasi ini, penulis menggunakan *Framework CodeIgniter* 3. Di bawah ini merupakan tampilan *website CodeIgniter* seperti yang tertera pada Gambar 2.9 dan dapat diakses pada alamat https://codeigniter.com/.

Gambar 2.9 Tampilan *Website CodeIgniter*

2.16 AJAX

AJAX adalah singkatan dari *Asynchronous JavaScript* and XML. Pada dasarnya ajax menggunakan XMLHttpRequest object *Javascript* untuk membuat

request ke server secara *asynchronous* atau tanpa melakukan *refresh* halaman *website*. Yang dibutuhkan agar ajax dapat berjalan adalah *javascript* harus di *enable* pada *browser* yang digunakan. Walaupun *javascript* merupakan dasar dari Ajax, dimana *javascript* sangat susah pada implementasi dan *maintenance*, tetapi Ajax memiliki struktur pemrograman yang lebih mudah untuk dipahami. Anda tinggal membuat *object* XMLHttpRequest dan memastikan *object* tersebut terbentuk dengan benar. Kemudian menentukan kemana hasilnya akan ditampilkan atau dikirim (Nugroho, 2020).

AJAX dapat digunakan untuk melakukan banyak hal,seperti *loading* halaman HTML tanpa *refresh* halaman *web*, validasi *form* dan banyak lagi yang bisa dilakukan dengan bahasa pemrograman PHP yang sangat *powerfull*. Ajax bertugas melakukan *request* ke *web* server dan PHP yang berada di server akan melakukan apa yang diminta oleh Ajax, mengirim hasilnya ke *web browser* dan Ajax menampilkannya kepada *user* (Nugroho, 2020).

2.17 jQuery

jQuery adalah pustaka *JavaScript* lintas-platform yang didesain untuk menyederhanakan *client-side scripting* pada HTML. Dewasa ini, jQuery merupakan pustaka *JavaScript* yang paling populer, dengan 65% pemasangan dari 10 juta situs *Web* dengan pengunjung tertinggi. jQuery adalah gratis, dan bersifat kode sumber terbuka dibawah lisensi MIT (Kasim, 2019).

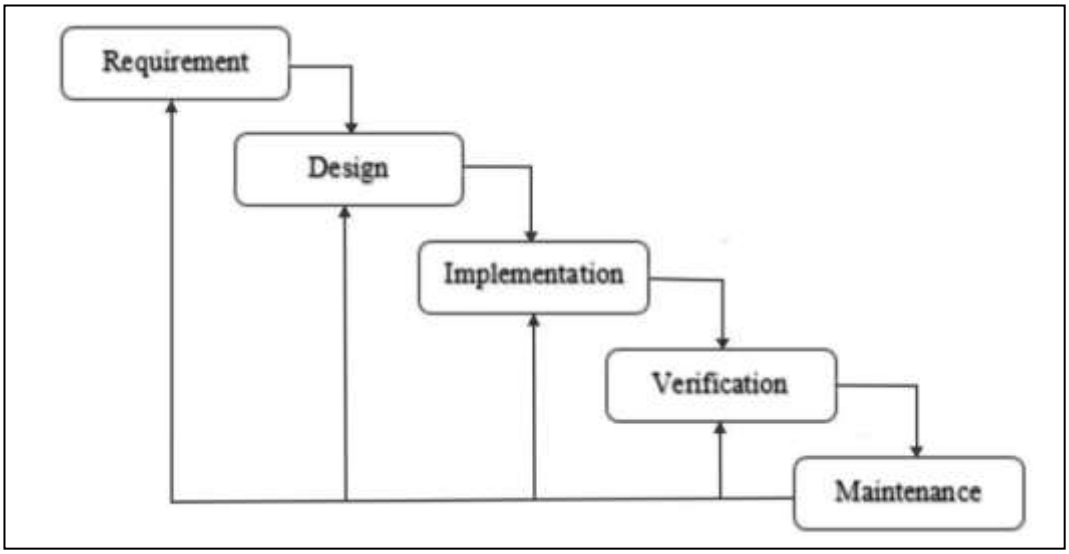
Sintaks pada jQuery didesain untuk memudahkan dalam navigasi sebuah dokumen, pemilihan elemen DOM, pembuatan animasi, penanganan *event*, dan pengembangan aplikasi berbasis Ajax. jQuery juga menyediakan kemampuan bagi para pengembang untuk dapat membuat plug-in pada pustaka *JavaScript* ini. Ini memungkinan mereka untuk membuat abstraksi pada interaksi dan animasi tingkat-rendah, efek lanjutan, serta tampilan *widget* yang dapat dimodifikasi. Pendekatan modular pada jQuery memungkinkan kita dalam pembuatan halaman *Web* yang dinamis dan aplikasi berbasis *Web* (Kasim, 2019).

jQuery pada intinya merupakan pustaka untuk memanipulasi DOM (*Document Object Model*). Dom adalah struktur-pohon representasi dari semua

elemen yang terdapat dalam sebuah halaman *Web*, dan jQuery menyederhanakan *syntax* untuk mencari, menyeleksi, dan memanipulasi elemen elemen DOM tersebut. Sebagai contoh, jQuery dapat digunakan untuk mencari elemen dalam dokumen dengan properti tertentu (mis: semua elemen dengan *tag* h1), mengubah satu atau lebih atribut tersebut (mis: warna, keterlihatan), atau membuatnya merespon sebuah *event* (mis: ketika mouse diklik). jQuery juga menyediakan sebuah paradigma untuk penanganan *event* yang diluar pemilihan dan manipulasi elemen dasar DOM. *Event assignment* dan *event callback function* dapat dilakukan dengan hanya satu langkah atau satu baris kode. jQuery juga bertujuan menggabungkan fungsional *JavaScript* yang serting digunakan (mislkan: *fade in* dan *fade out* ketika menyembunyikan elemen, animasi dengan memanipulasi *property* CSS) (Kasim, 2019).

2.18 Metode Diagram Alir (Waterfall)

Model *Waterfall* merupakan salah satu model SDLC yang sering digunakan dalam pengembangan sistem informasi atau perangkat lunak. Model ini menggunakan pendekatan sistematis dan berurutan. Tahapan dalam model ini dimulai dari tahap perencanaan hingga tahap pengelolaan *(maintenance)* dan dilakukan secara bertahap. Pengembang perlu mengetahui lebih lanjut tentang bagaimana proses pengembangan sistem jika menggunakan model *waterfall* dan juga karakteristik dari model *waterfall* tersebut (Wahid, 2020). Berikut merupakan tahapan pada metode *waterfall* seperti yang ditunjukkan pada Gambar 2.10.



Gambar 2.10 Tahapan Metode *Waterfall*

Berikut merupakan tahapan dari metode *waterfall* :

1. *Requirement*

   Tahap ini pengembang sistem diperlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2. *Design*

   Pada tahap ini, pengembang membuat desain sistem yang dapat membantu menentukan perangkat keras (*hardware*) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3. *Implementation*

   Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing.

4. *Verification*

   Pada tahap ini, sistem dilakukan verifikasi dan pengujian apakah sistem sepenuhnya atau sebagian memenuhi persyaratan sistem, pengujian dapat dikategorikan ke dalam unit *testing* (dilakukan pada modul tertentu kode),

sistem pengujian (untuk melihat bagaimana sistem bereaksi ketika semua modul yang terintegrasi) dan penerimaan pengujian (dilakukan dengan atau nama pelanggan untuk melihat apakah semua kebutuhan pelanggan puas).

5. Maintenance

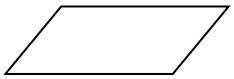
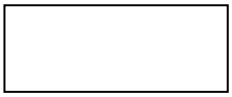
   Ini adalah tahap akhir dari metode *waterfall*. Perangkat lunak yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

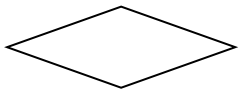
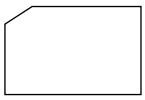
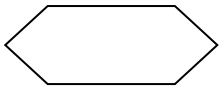
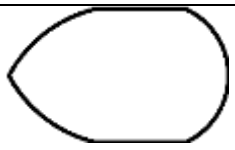
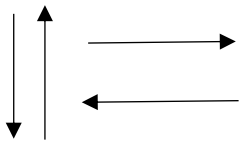
2.19 Flowchart atau Diagram Alir

Diagram Alir (*Flowchart*) adalah bagan (*chart*) yang menunjukkan alir atau arus (*flow*) di dalam program atau prosedur sistem secara logika (Rusmawan, 2019). *Flowchart* adalah algoritma dalam suatu program, yang menyatakan arah alur program tersebut (Qoriah dkk, 2020). *Flowchart* (bagan alir) merupakan sebuah gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program yang menyatakan arah alur program tersebut (Rusmawan, 2019).

Flowchart memiliki simbol tersendiri yang digunakan untuk menggambarkan algoritma dalam bentuk diagram alur dan kegunaan dari simbol-simbol yang bersangkutan dijelaskan pada Tabel 2.1.

Tabel 2.1 Simbol-simbol *Flowchart*

No.	Simbol	Nama	Fungsi
1.		Terminal	Menyatakan permulaan atau akhir suatu program
2.		Input / Output	Menyatakan proses input atau output tanpa tergantung jenis peralatannya
3.		Process	Menyatakan suatu tindakan (proses) yang dilakukan oleh computer.

No.	Simbol	Nama	Fungsi
4.		Decision	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya / tidak
5.		Connector	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
6.		Offline Connector	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
7.		Predefined process	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
8.		Punched Card	Menyatakan input berasal dari kartu atau output ditulis ke kartu
9.		Preparation symbol	Memberi nilai awal suatu besaran
10.		Document	Mencetak keluaran dalam bentuk dokumen (melalui printer)
11.		Display	Menyatakan peralatan output yang digunakan
12.		Flow	Menyatakan jalannya arus suatu proses

2.20 Pengujian Black Box

Pendekatan *Black Box testing* ini melakukan pengujian terhadap fungsi operasional *software*. Pendekatan ini biasanya dilakukan oleh penguji yang tidak ikut serta dalam pengkodean *software* (Rusyana, 2019).

1. *Black Box testing* berfokus pada kebutuhan fungsional pada *software* berdasarkan pada spesifikasi kebutuhan dari *software*.
2. *Black Box testing* bukan teknik alternative daripada *white box testing*.
3. *Black Box testing* melakukan pengujian tanpa pengetahuan detil instruktur internal dari sistem atau komponen yang dites juga disebut sebagai *behavioral testing, specification-based testing, input/output testing* atau *functional testing*.
4. Pada *Black Box testing* terdapat jenis teknik desain tes yang dapat dipilih berdasarkan pada tipe testing yang akan digunakan diantaranya adalah *equivalence class partitioning, boundary value analysis, state transitions testing, cause-effect graphing*.

Kategori *error* yang akan diketahui melalui *Black Box testing* diantaranya adalah fungsi yang hilang atau tak benar, *error* dari antar-muka, *error* dari struktur data atau akses *external database*, *error* dari kinerja atau tingkah laku, *error* inisialisasi dan terminasi pendekatan *Black Box testing*, pendekatan ini melakukan pengujian terhadap fungsi operasional *software* (Rusyana, 2019).