

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Aplikasi

Menurut Setyawan dan Munari (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.

2.1.1 Pengertian Aplikasi Berbasis Web

Menurut Setyawan dan Munari (2020), 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*.

2.1.2 Tujuan Aplikasi Berbasis Web

Menurut Setyawan dan Munari (2020), tujuan aplikasi berbasis *web* 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 *memonitoring* sistem dalam hal tampilan, dapat didesain dan disesuaikan untuk berbagai jenis industri.

2.1.3 Karakteristik Aplikasi Berbasis Web

Menurut Setyawan dan Munari (2020), karakteristik aplikasi berbasis *web* 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 Pengertian Komunikasi

Pengertian komunikasi menurut KBBI adalah pengiriman dan penerimaan pesan atau berita antara dua orang atau lebih sehingga pesan yang dimaksud dapat dipahami.

2.2.1 Chat

Chatting, yaitu sebuah bentuk komunikasi secara langsung seperti bercakap-cakap atau berkirim pesan secara *real time* menggunakan teks melalui jaringan internal. (Maulana dan Rahmatulloh, 2019)

2.3 Pengertian *Monitoring*

Monitoring didefinisikan sebagai langkah untuk mengkaji apakah kegiatan yang dilaksanakan telah sesuai dengan rencana, mengidentifikasi masalah yang timbul agar langsung dapat diatasi, melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan, mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan. (Sutabri, 2012)

Dengan kata lain, *monitoring* merupakan salah satu proses didalam kegiatan organisasi yang sangat penting yang dapat menentukan terlaksana atau tidaknya sebuah tujuan organisasi. Tujuan dilakukannya *monitoring* adalah untuk memastikan agar tugas pokok organisasi dapat berjalan sesuai dengan rencana yang telah ditentukan. (Aviana, 2012)

2.3.1 Pengertian *Monitoring Jaringan*

Menurut Handri Hermawan (2017), *monitoring* jaringan komputer adalah proses pengumpulan dan melakukan analisis terhadap data-data pada lalu lintas jaringan dengan tujuan memaksimalkan seluruh sumber daya yang dimiliki jaringan komputer. *Monitoring* jaringan komputer dapat dibagi menjadi 2 bagian yaitu :

a. Connection Monitoring

Connection monitoring adalah teknik *monitoring* jaringan yang dapat dilakukan dengan melakukan tes *ping* antara *monitoring station* dan *device* target, sehingga dapat diketahui bila koneksi terputus.

b. Traffic Monitoring

Traffic monitoring adalah teknik *monitoring* jaringan dengan melihat paket aktual dari *traffic* pada jaringan dan menghasilkan laporan berdasarkan *traffic* jaringan.

2.4 Pengertian Jaringan Komputer

Jaringan komputer adalah sekumpulan komputer yang terhubung dan membentuk sebuah jaring-jaring yang dapat saling terhubung satu sama lain. Tidak hanya saling terhubung, tetapi dapat dimanfaatkan untuk berbagai sumber daya (*printer*, CPU), berkomunikasi (pesan, instan, surel), dan dapat mengakses informasi (*browsing web*). Sebuah jaringan komputer minimal terbentuk dari dua komputer yang saling terhubung sehingga masing-masing dapat membagi sumber dayanya (*resource sharing*) atau saling berkomunikasi bertukar data. (Ariawal dan Purbo, 2016)

2.5 Pengertian Website

Website dapat diartikan sebagai kumpulan halaman yang berisi informasi digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia. (Abdulloh, 2018)

2.5.1 Pengertian Pemrograman Web

Pemrograman *web* dapat diartikan sebagai kegiatan pembuatan program atau aplikasi berbasis *web* menggunakan bahasa pemrograman tertentu sehingga dapat memproses data dan menghasilkan informasi sesuai dengan yang dikehendaki oleh pemilik *website*. (Abdulloh, 2018)

2.5.2 Hyper Text Markup Language (HTML)

Dasar HTML dapat mencakup petunjuk untuk memformat dalam bahasa yang disebut *Cascading Style Sheets* (CSS) dan program untuk berinteraksi dalam bahasa yang disebut JavaScript.

HTML adalah bahasa yang mengatur bagaimana tampilan isi dari situs *web*, di dalam html ada *tag-tag* dimana *tag* berfungsi menyediakan informasi berkaitan dengan sifat dan struktur konten serta referensi untuk gambar dan media lainnya. (Winarno, dkk., 2014)

2.5.3 JavaScript

JavaScript adalah bahasa *scripting client side* yang sangat populer. Hampir semua *web programmer* menggunakan *JavaScript* untuk memberi efek pemrograman di halaman. *JavaScript* tidak hanya berdiri sendiri, tapi *JavaScript* juga menjadi dasar yang bisa digunakan untuk teknologi lainnya, seperti AJAX, jQuery dan jQuery *Mobile*. (Winarno, dkk., 2014)

2.5.3.1 JQuery

JQuery merupakan salah satu dari sekian banyak banyak *JavaScript library*, yaitu kumpulan fungsi *JavaScript* yang siap pakai, sehingga mempermudah dan mempercepat dalam membuat kode *JavaScript*. Dengan menggunakan jQuery, *JavaScript* yang panjang dapat disingkat menjadi beberapa baris kode saja. (Abdulloh, 2018)

2.5.4 Hypertext Preprocessor (PHP)

PHP singkatan dari *PHP: Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server (server side HTML embedded scripting)*. PHP adalah *script* yang digunakan untuk membuat halaman *website* dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima oleh *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan. (Anhar, 2010)

2.5.5 Structured Query Language (SQL)

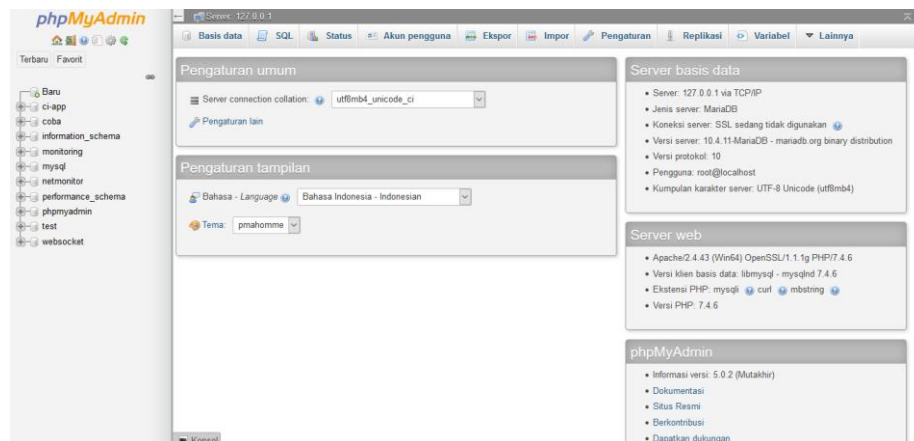
Structured Query Language (SQL) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara *de facto* merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua *server* basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya. (Habibi, dkk., 2020)

2.5.5.1 MySQL

MySQL (*My Structure Query Language*) adalah salah satu *Database Management System* (DBMS) dari sekian banyak DBMS seperti Oracle, MS SQL, PostgreSQL, dan lainnya. MySQL berfungsi untuk mengolah *database* menggunakan bahasa SQL. MySQL bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung/*support* dengan *database* MySQL. (Anhar, 2010)

2.5.5.2 PhpMyAdmin

Untuk mengakses *database* MySQL digunakan *software* MySQL *client*. Ada banyak MySQL *client*, dari mulai yang berbasis *command line*, aplikasi GUI hingga berbasis *web*. PhpMyAdmin adalah MySQL *client* yang berupa aplikasi *web* dan umumnya tersedia di *server* PHP seperti XAMPP maupun *server* komersial lainnya. (Winarno, dkk., 2014)

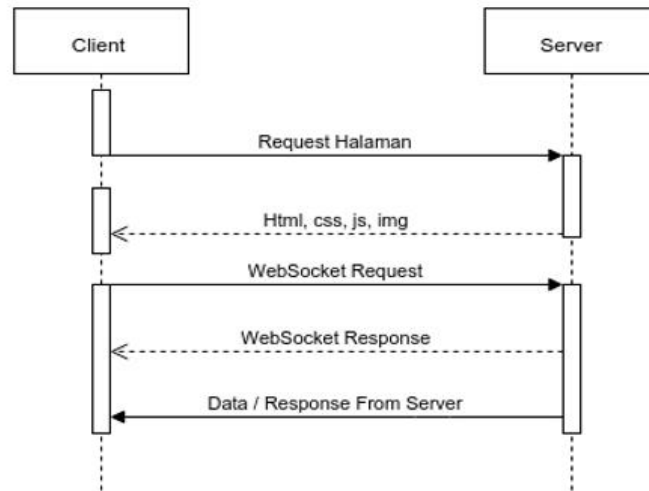


Gambar 2.1 PhpMyAdmin

2.6 WebSocket

Protokol *WebSocket* ditentukan dalam RFC 6455 dan menggunakan HTTP sebagai protokol *transport* yang mendasarinya. Berbeda dengan paradigma permintaan / balasan tradisional, di mana klien mengirim *request* ke server, yang kemudian server membalas dengan pesan respon, koneksi *WebSocket* dapat tetap terbuka untuk waktu yang lama, dan baik server maupun klien dapat mengirim dan menerima pesan (atau *data frames*) di *WebSocket*. (Palala dan Helmich, 2016)

WebSocket



Gambar 2.2 Cara Kerja *WebSocket*

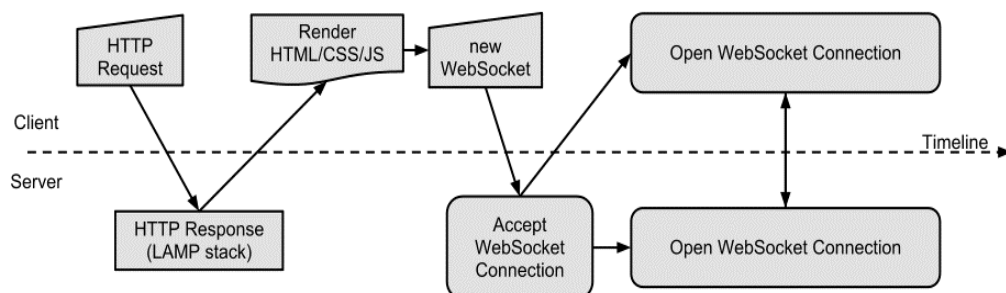
(Sumber: <https://rizkimufrizal.github.io/belajar-membuat-aplikasi-chat-dengan-websocket/>)

WebSocket memungkinkan interaksi antara *web client* (misalnya *browser*) dengan sebuah *web server* berbeban rendah. Dengan *WebSocket*, *data transfer* dan *real-time* dari dan menuju ke *server* menjadi lebih cepat. (Dinata, 2018)

2.6.1 Ratchet

Ratchet adalah *library* PHP yang digabungkan secara longgar yang menyediakan *tools* bagi *developer* untuk membuat aplikasi dua arah yang *real-time* antara klien dan server melalui *WebSockets*. (Anon, <http://socketo.me>)

Berikut ini merupakan gambaran cara kerja dari Ratchet:



Gambar 2.3 Cara Kerja Ratchet

(Sumber: <http://socketo.me/docs/flow>)

2.7 Framework CodeIgniter

2.7.1 Pengertian Framework

Framework merupakan banyak kode, yang disimpan dalam beberapa file yang terpisah, dan memudahkan dalam penggunaan kode yang digunakan secara berulang-ulang. Dengan menggunakan *framework*, tidak diperlukan menulis baris kode yang panjang untuk fungsi tertentu, hanya perlu menggunakan fungsi karena fungsi yang sudah dituliskan implementasinya dalam suatu framework. (Jubilee Enterprise, 2015)

Framework secara sederhana dapat diartikan kumpulan dari fungsi-fungsi/prosedur-prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang pemrograman, tanpa harus membuat fungsi atau *class* dari awal. (Supono dan Putratama, 2018)

2.7.2 CodeIgniter

CodeIgniter adalah aplikasi *open source* berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP. CodeIgniter memudahkan *developer* atau pengembang *web* untuk membuat aplikasi *web* dengan cepat dan mudah dibandingkan dengan membuat dari awal. (Supono dan Putratama, 2018)

2.7.3 Konsep MVC (Model, View, Controller)

MVC adalah sebuah pola pemrograman yang bertujuan memisahkan logika bisnis, logika data dan logika tampilan (*interface*), atau secara sederhana memisahkan antara proses, data dan tampilan. MVC mengatur arsitektur sebuah aplikasi. Umumnya aplikasi yang dibangun menggunakan konsep MVC adalah aplikasi yang cukup besar, karena salah satu keuntungan dari MVC itu adalah kemudahan maintenance, dan pengembangan aplikasi tersebut.

CodeIgniter menggunakan konsep ini, Anda harus memisahkan kode *database* ke folder *model*, kode proses ke folder *controller*, dan kode tampilan ke folder *view*. (Wardana, 2010)

2.8 *Composer*

Composer adalah sebuah *tool* yang bertujuan untuk memudahkan *developer* dalam *me-manage dependency* pada proyek berbasis PHP. *Composer* seperti *npm* pada *NodeJs*, *yum* pada *Redhat*, *bundler* pada *Ruby*, atau *apt* pada *Ubuntu*. *Composer* dapat digunakan untuk meng-*install*, meng-*update* dan menghapus *library* yang kita gunakan dalam proyek. (Iksanudin, 2019)

2.9 Perangkat Lunak yang Dibutuhkan

Berikut ini perangkat lunak yang dibutuhkan untuk pemrograman *web*:

2.9.1 XAMPP

XAMPP adalah salah satu paket *installer* yang berisi *Apache* yang merupakan *web server* tempat penyimpanan *file-file* yang diperlukan *website*, dan *phpmyadmin* sebagai aplikasi yang digunakan untuk perancangan *database MySQL*. (Abdulloh, 2016)



Gambar 2.4 Logo XAMPP

2.9.2 Sublime Text Editor

Sublime *Text Editor* adalah program aplikasi yang digunakan untuk mengedit teks dan skrip kode bahasa pemrograman. Terdapat banyak bahasa pemrograman yang dikenali oleh aplikasi *text editor* ini. Mulai dari bahasa HTML, PHP, *JavaScript*, dan masih banyak lagi. Selain itu, warna koding yang bervariasi dan sangat menarik mampu membuat Anda lebih betah dalam menjalankan proses koding.



Gambar 2.5 Logo Sublime Text Editor




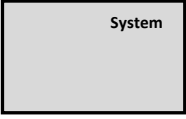
2.10 Pemodelan Sistem dengan UML (*Unified Modeling Language*)

Menurut Prihandoyo (2018), *Unified Modeling Language* (UML) merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek. UML merupakan sebuah standar penulisan atau semacam *blue print* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik.

2.10.1 *Use Case Diagram*

Menurut Prihandoyo (2018), *use case diagram* merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat *actor* yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem.



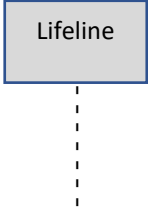
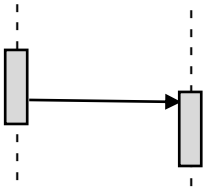
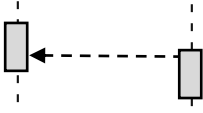

Tabel 2.1 Notasi *Use Case Diagram*

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Pihak yang mengakses <i>use case</i> .
2.		<i>Use Case</i>	Mewakili apa yang sistem bisa lakukan.
3.		<i>Association</i>	Merelasikan aktor dengan <i>use case</i> .
4.		<i>System Boundary</i>	Menggambarkan batasan sistem terhadap lingkungan.

2.10.2 *Sequence Diagram*

Menurut Prihandoyo (2018), *sequence diagram* menggambarkan interaksi antar objek didalam dan di sekitar sistem yang berupa *message* yang digambarkan terhadap waktu.

Tabel 2.2 Notasi *Sequence Diagram*

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi.
2.		<i>Time Active</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi dengan pesan.
3.		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
4.		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
5.		<i>Return Message</i>	Objek/metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
6.		<i>Self Message</i>	Relasi ini menunjukkan bahwa suatu objek hendak memanggil dirinya sendiri.