BAB II
TEORI PENUNJANG

2.1 Pengertian Web Server

Web server merupakan perangkat lunak atau perangkat keras yang dikhususkan sebagai pemberi layanan kepada *client*. Fungsi utama dari *web server* adalah menyimpan data, memproses data, dan memberikan layanan *website* kepada *client. Web server* dapat bersifat lokal maupun global, dimana *web server* lokal hanya bisa diakses di area tertentu, misalnya satu sekolah, sedangkan *web server* global dapat diakses dari manapun asalkan *client* memiliki koneksi jaringan Internet. Penghubungan komunikasi antara *web server* dengan *client* menggunakan HTTP dan biasanya layanan berkas HTML yang diberikan kepada *client.*

2.2 Pengertian Jaringan Client-Server

Jaringan *client-server* merupakan konfigurasi jaringan yang memiliki dua ketegori bagian. Bagian pertama yaitu sistem komputer yang digunakan sebagai *server* dan bagian kedua yaitu sistem komputer yang digunakan sebagai *client*. Pembagian kategori ini bertujuan agar memudahkan dalam melakukan sentralisasi jaringan. Dimana *server* difungsikan sebagai pemberi layanan kepada *client*, contohnya memberikan layanan *website*, sedangkan *client* melakukan permintaan layanan kepada *server*, contohnya *client* meminta layanan *website*.

2.3 Pengertian Tunneling

Tunneling merupakan konfigurasi jaringan yang menghubungkan dua jaringan *private* yang memanfaatkan jaringan umum Internet. Penghubungan dua jaringan *private* agar dapat melalui jaringan umum Internet membutuhkan jalur koneksi *virtual*. Dimana jalur koneksi *virtual* ini menggunakan konfigurasi perangkat *tun* dan *tap*. Perangkat *tun* digunakan sebagai *virtual IP point-to-point device*, sedangkan perangkat *tap* digunakan sebagai *virtual ethernet device.*

2.4 Pengujian Beban Web Server

Pengujian beban *web server* dilakukan untuk mengetahui beban kerja yang dapat dikerjakan dalam waktu tertentu oleh *web server*, antara lain meliputi

deviation, throughput, response time, error %, received KB/s, dan *sent KB/s.* Pengujian beban *web server* menggunakan *HTTP request,* dimana pengujian menggunakan metode *POST* dan *GET.* Penggunaan metode *POST* dan *GET* dilakukan untuk melakukan simulasi *client* saat meminta layanan kepada *web server,* dimana metode *POST* yaitu *client* memberikan data kepada *web server* untuk diproses, sedangkan metode *GET* yaitu *client* meminta informasi kepada *web server* untuk ditampilkan. Tujuan pengujian beban *web server* untuk mengetahui ketahanan maksimal server memberikan layanan kepada *client* dalam satu waktu.

2.5 Pengertian Pemrograman Berorientasi Objek

Pemrograman berorientasi objek merupakan salah satu metode perancangan perangkat lunak. Metode perancangan ini menekankan pada objek ketimbang aksi dan data ketimbang logika. Metode ini memusatkan pada objek yang ingin dimanipulasi ketimbang logika yang dibutuhkan untuk memanipulasinya.

2.6 Pengertian UML (Unified Modeling Language)

UML merupakan salah satu bahasa pemodelan perangkat lunak yang dijadikan standar dalam memodelkan objek nyata sebagai langkah awal melakukan perancangan sistem pemrograman berorientasi objek.

2.7 Pengertian Basis Data

Basis data merupakan tempat penampungan informasi organisasi di dalam sistem komputer yang dapat dengan mudah diakses, diolah, dan diperbarui. Data diorganisasi kedalam baris, kolom, dan tabel, dimana terdapat index untuk mempermudah pencarian informasi tertentu, misalnya mencari informasi pada tabel pengguna dengan index satu, maka akan menampilkan seluruh informasi pengguna yang memiliki index satu, bisa berupa nama lengkap, alamat, dan umur. Data pada basis data bersifat *integrated* dan *shared,* dimana sifat *integrated* adalah data pada basis data saling terkait satu sama lain, sedangkan sifat *shared* adalah data pada basis data dapat digunakan pada waktu bersamaan. Tujuan penggunaan basis data untuk mempermudah dalam melakukan organisasi data-data pada sistem rancangan.

2.8 Pengujian Black-box

Pengujian *black-box* merupakan salah satu pengujian perangkat lunak, pengujian *black-box* dilakukan untuk melihat fungsi *input-output* tanpa mempedulikan struktur internal dan cara kerjanya (hanya melihat hasil eksekusi). Dimana kebanyakan pengujian *fungsional* yang dilakukan, walaupun tidak menutup kemungkinan pengujian *non-fungsional* dilakukan. Perancang melakukan uji dengan memberikan *input* kemudian mengamati hasil *output* apakah sudah sesuai dengan rancangan. Tujuan melakukan pengujian *black-box* untuk mengetahui fungsi-fungsi yang kurang atau tidak benar, kesalahan antarmuka, kesalahan akses basis data, kesalahan kinerja, dan kesalahan inisialisasi mau pun terminasi.

2.9 Raspberry Pi 3

Raspberry Pi 3 adalah mini komputer yang dikembangkan oleh Raspberry Pi Foundation di Inggris. Tujuan dari pengembangan ini untuk membantu siswa dasar-dasar ilmu komputer. Produk Raspberry Pi 3 menggunakan prosesor buatan Broadcom ARM dengan frekuensi 1.2 GHz. Penggunaan pengolah grafis buatan Broadcom VideoCore IV dengan frekuensi 400MHz. Dikombinasikan dengan memori berkapasitas 1 GB DDR2 dengan frekuensi 900MHz. Konsumsi daya pada saat *idle* berkisar 1,5 watt dan pada saat *full-load* berkisar 6.7 watt.

2.10 Raspbian

Raspbian adalah sistem operasi berbasis Linux Debian. Sering digunakan pada mini komputer Raspberry. Sejak 2015 dikembangkan oleh Raspberry Pi Foundation sebagai sistem operasi utama produk-produk Raspberry. Tampilan desktop menggunakan *Pi Improved Xwindows Environment, Lightweight (PIXEL).*

2.11 Apache

Apache adalah perangkat lunak *open-source cross-platform web server*. Penggunaan *web server* untuk melayani permintaan halaman *website*. Protokol yang digunakan pada Apache unutk melayani permintaan *website* yaitu HTTP.

2.12 OpenVPN

OpenVPN adalah perangkat lunak *open-source* yang mengimplementasikan teknik *Virtual Private Network (VPN)*. OpenVPN membuat koneksi aman *point-to-point* atau *site-to-site* pada router. Memiliki protokol keamanan khusus berbasis *Secure Sockets Layer (SSL)* atau *Transport Layer Security (TLS)* yang memungkinkan untuk melewati *Network Address Translators (NAT)* dan *firewall*.

2.13 Hostapd

Hostapd (Host access point daemon) adalah perangkat lunak untuk mengkonfigurasikan normal *Network Interface Card (NIC)* menjadi *Access Point (AP)*. Teknologi wi-fi yang diimplementasikan oleh Hostapd yaitu IEEE 802.11.

2.14 Dnsmasq

Dnsmasq adalah perangkat lunak *open-source* yang digunakan untuk melakukan konfigurasi *DNS forwarder* dan *DHCP server*. Dnsmasq sering digunakan pada *router* umum yang sering digunakan pada rumah-rumah. Static DHCP dan dinamik DHCP didukung oleh Dnsmasq. Serta mendukung teknologi modern seperti IPv6 dan DNSSEC.

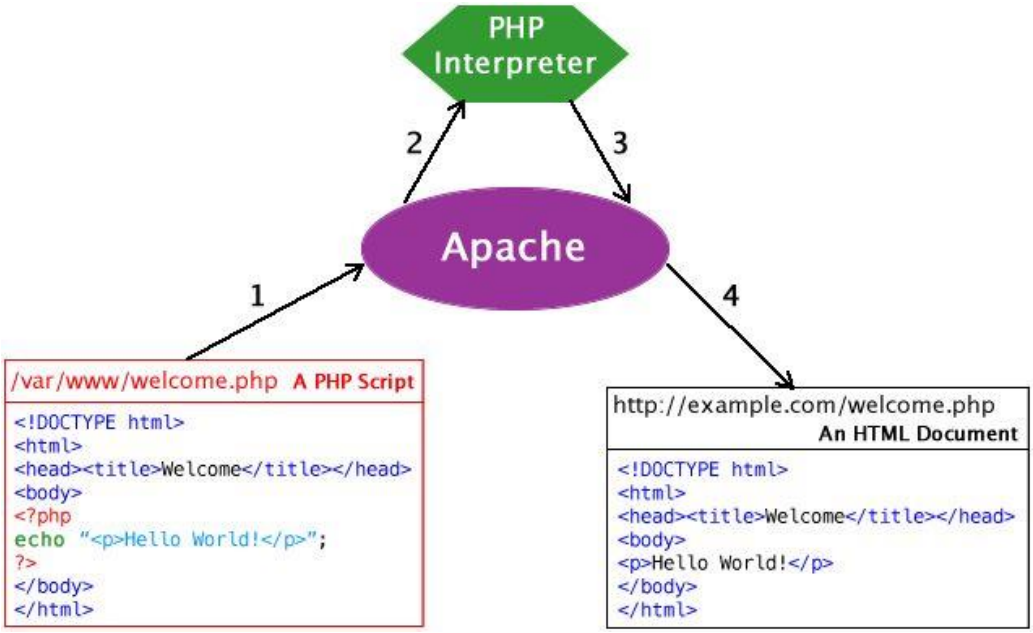
2.15 MySQL

MySQL merupakan salah satu basis data *RDBMS (Relational Database Management System)* barbasis *SQL (Structured Query Language)*. MySQL bekerja secara *virtual* pada sistem operasi Linux, UNIX, dan Windows. MySQL juga sering digunakan sebagai *stack* perancangan *website*, lebih sering disebut *LAMP stack*, dimana *"L"* mewakili *Linux* sebagai sistem operasi, *"A"* mewakili *Apache* sebagai perangkat lunak *web server*, *"M"* mewakili *MySQL* sebagai basis data, dan yang terakhir *"PHP"* sebagai bahasa pemrograman berbasis objek.

2.16 PHP

PHP adalah *server-side scripting* untuk perancangan berkas *website*. Penggunaan PHP bertujuan agar perancangan berkas *website* bersifat dinamis. Pada dasarnya *web server* (Contoh: Apache) hanya dapat menyediakan layanan HTML kepada *client* yang mengakses *website*. *Script* PHP hanya diproses pada server. *Script* PHP dapat diimplementasikan kedalam berkas HTML, hanya

tinggal menambahkan tag awal *"<?php"* dan *tag* akhir *"?>"*. PHP *interpreter* akan memproses kode PHP dan peranagkat lunak *web server* akan menggabungkan hasil dari PHP *interpreter* untuk melakukan eksekusi kode PHP. Lebih jelasnya dapat dilihat pada gambar 2.1 di bawah.



Gambar 2.1 Proses Menyediakan Layanan Website

Pada gambar 2.1 dapat dijelaskan sebagai berikut:

1. File PHP diproses di Apache.
2. PHP *interpreter* memproses kode PHP.
3. Apache menggabungkan hasil PHP *interpreter* dengan kode PHP.
4. Apache menyediakan berkas PHP kepada *browser*.

2.17 HTML

Hypertext Markup Language (HTML) adalah bahasa *markup* untuk perancangan *website*. Bahasa *markup* digunakan untuk memproses, mendefinisikan dan mempresentasikan text. Terdapat kode-kode khusus untuk memisahkan beberapa format text pada satu berkas. Istilah kode ini sering disebut dengan *tag* dalam bahasa pemrograman. Beberapa contoh HTML *tag* dapat dilihat pada tabel 2.1 di bawah.

Tabel 2.1 Beberapa Contoh HTML Tag

HTML tag	Keterangan tag
<!-- … -->	Komentar
…	Paragraf
<audio> … </audio>	Audio
<video> … </video>	Video
<script> … </script>	*Client-side* script

Pada tabel 2.1 dijelaskan beberapa HTML *tag*. Lebih jelasnya akan diberikan contoh tiap HTML *tag* pada tabel 2-1, sebagai berikut:

1. <!-- Text ini berupa komentar dan tidak akan tampil pada *website* -->
2. Text ini memiliki format paragraf
3. <audio><source src=”audio.mp3” type=”audio/mp3”></audio>
4. <video><source src=”video.mp4” type=”video/mp4”></</video>
5. <script>$(#field).val();</script>

2.18 JavaScript

JavaScript adalah bahasa pemrograman untuk HTML. JavaScript salah satu inti perkembangan perancangan *website*. Browser populer seperti *Google Chrome, Mozilla Firefox dan Safari* sudah dapat mendukung JavaScript tanpa perlu menambahkan *plug-in* tambahan. Pada awalnya JavaScript hanya sebagai *client-side script web browser* sedangkan sekarang sudah berkembang pada non-*browser* perangkat lunak PDF dan desktop *widget*.

2.19 CSS

Cascading Style Sheets (CSS) adalah bahasa yang mempresentasikan *style* pada bahasa *markup*. Bahasa markup yang sering menggunakan CSS yaitu HTML. CSS digunakan untuk memodifikasi tampilan *website* menjadi lebih *user-friendly*. CSS memisahkan antara aspek konten *website* dan tampilan *website*, aspek tersebut meliputi tata letak, warna dan *font*. Pemisahan aspek ini menjadikan konten lebih fleksibel, dimana *style* dapat digunakan oleh beberapa konten. Sehingga mengurangi kompleksitas dan pengulangan *script* yang sama

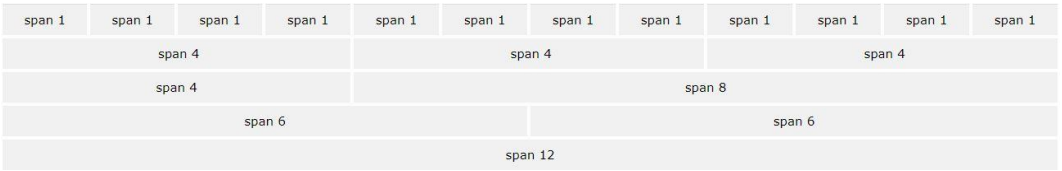
pada beberapa berkas HTML. Pengubahan *style* pun hanya perlu mengubah berkas CSS tanpa perlu mengubah berkas HTML.

2.20 AJAX

Asynchronous JavaScript and XML (AJAX) adalah salah satu teknik dalam transfer data. Teknik AJAX dapat melakukan pembaruan konten *website* tanpa perlu melakukan *reload* halaman *website* secara menyeluruh, dapat pula melakukan permintaan dan penerimaan data server setelah halaman *website* terakses sepenuhnya, serta mengirim data ke server secara *background* tanpa perlu mengganggu tampilan dan kinerja dari halaman *website*.

2.21 Bootstrap

Bootstrap adalah *open-source front-end framework* untuk mempermudah perancangan *website*. Dimana sudah terdapat *template* dasar HTML, CSS, dan JavaScript. Contoh *template build-in* yang sudah disediakan oleh Bootstrap adalah grid, modal dan menu navigasi. Bootstrap CSS dapat menyesuaikan tampilan berdasarkan kategori *desktop, tablet* dan *smartphone*. Maka dari itu Bootstrap dapat digunakan sebagai perancangan berkas *website* responsif. Sistem grid Bootstrap dapat dilihat pada gambar 2.2 di bawah.



Gambar 2.2 Sistem Grid Bootstrap

Pada gambar 2.2 dapat dilihat bahwa grid pada Bootstrap dibagi dua belas kolom. Dimana piksel tiap kolomnya secara otomatis mengikuti ukuran resolusi layar pada perangkat yang mengakses *website*. Contoh: Pada resolusi 1920x1080 setiap kolom terkecil memiliki panjang 160px sedangkan pada resolusi 1366x768 setiap kolom terkecil memiliki kisaran panjang 113px.