



## PRAKTIKUM 1 PENGANTAR WEB

### 1.1. TUJUAN PRAKTIKUM

#### Tujuan Umum

Mahasiswa memahami cara kerja web, mengetahui kode *scripting* pada web, dan mengetahui beberapa *tools* pengembangan web

#### Tujuan Khusus

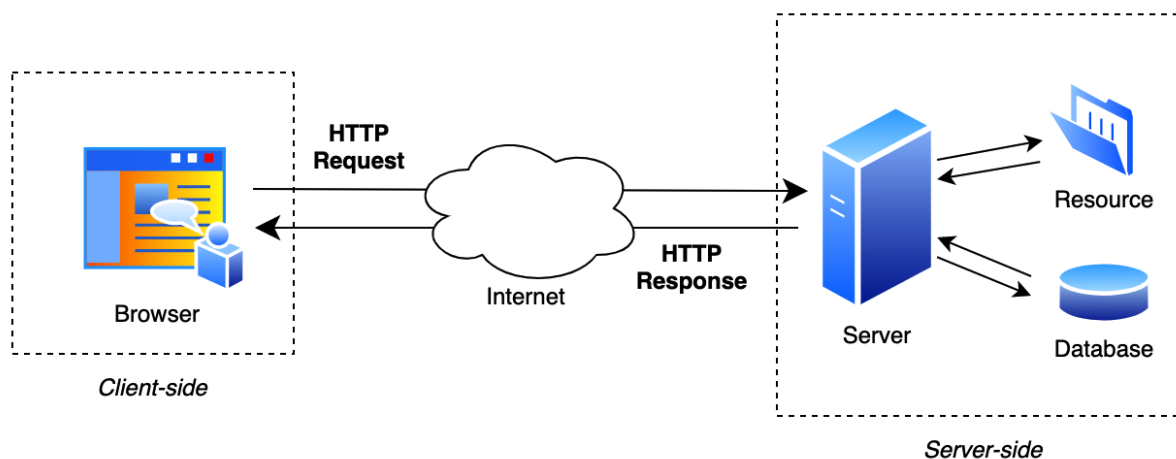
Mahasiswa dapat

1. Menjelaskan cara kerja web
2. Mengetahui kode *scripting* pada web
3. Mengetahui *tools* pengembangan web

### 1.2. TEORI SINGKAT

#### Gambaran Umum Cara Kerja Web

Web (berasal dari *World Wide Web* atau “www”) adalah salah satu cara untuk berbagi informasi melalui internet. Web bekerja dengan menggunakan protokol *HTTP* (*HyperText Transfer Protocol*). *HTTP* adalah protokol jaringan di lapisan aplikasi (*application layer*) yang dikembangkan untuk membantu proses transfer antar komputer, misalnya dokumen, file, gambar, dan video. Komunikasi tersebut berlangsung antara *web server* dengan komputer *client* atau sebaliknya. Perhatikan ilustrasi pada Gambar 1 berikut.

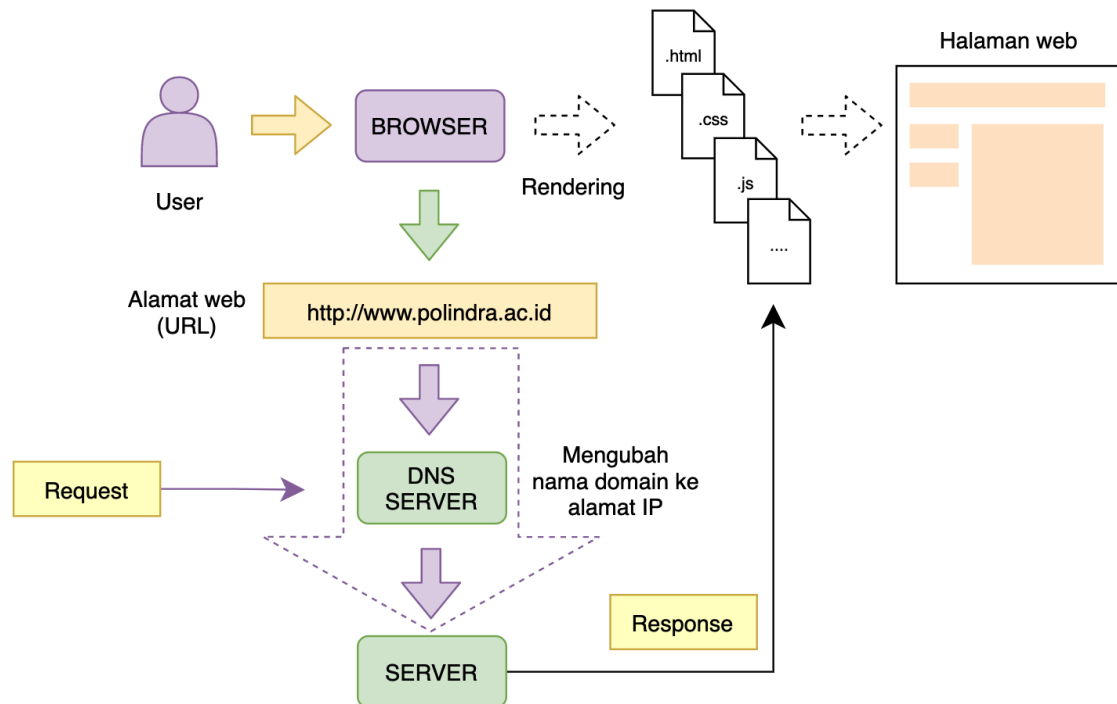


**Gambar 1.** Ilustrasi Gambaran Umum Cara Kerja Web

Pada Gambar 1 dijelaskan bahwa web bekerja melalui inisiasi *user* yang melakukan permintaan kepada *server* menggunakan *HTTP Request* dari aplikasi *browser* di lingkungan *client* (*client-side*). Selanjutnya, *server* menangani permintaan *user* dengan melakukan pengelolaan suatu *resource* atau *database* yang ada di lingkungan *server* (*server-side*). Jika proses pengelolaan yang dilakukan *server* telah selesai, maka permintaan *user* ditanggapi melalui *HTTP Response*. *Browser* yang menerima tanggapan dari *server* selanjutnya melakukan proses *rendering* agar dapat menampilkan hasil yang diberikan oleh *server*.

Komunikasi antar *client-side* dan *server-side* umumnya terhubung ke jaringan internet, namun komunikasi ini bisa saja dilakukan pada jaringan lokal (intranet).

### Rincian Cara Kerja Web

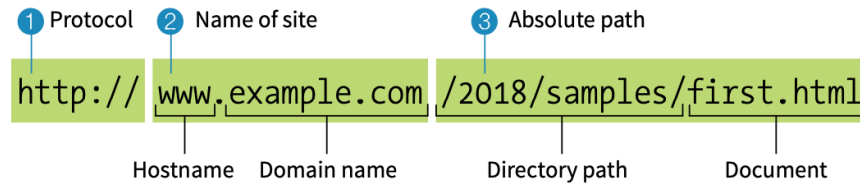


**Gambar 2.** Ilustrasi Rincian Cara Kerja Web

Adapun rincian tahapan cara kerja web yang terjadi (lihat Gambar 2) saat *user* memasukkan suatu alamat web yang lengkap (dikenal dengan istilah *URL* atau *Uniform Resource Locator* misalnya “http://www.polindra.ac.id”) atau memasukkan nama domain (alamat/situs web misalnya “polindra.ac.id”) pada *browser* yaitu:

1. Penyelesaian *URL*

*Resource* pada suatu situs web tidak tersimpan pada lingkungan *client (client-side)*, melainkan tersimpan pada *web server (server-side)* yang ada di suatu *hosting* (tempat untuk menyimpan semua *file* situs web). Situs web pada dasarnya hanya dapat diakses melalui alamat *IP (Internet Protocol)*, misalnya 172.16.254.1. Namun, penggunaan susunan kode alamat *IP* cukup sulit untuk diingat *user*. Untuk memudahkan *user* dalam mengakses suatu situs web pada *server*, digunakanlah nama domain (misalnya “polindra.ac.id”). Nama domain ini tersimpan pada suatu *DNS (Domain Name System) server*. *DNS server* adalah *server* yang berfungsi khusus dalam menerjemahkan nama domain ke alamat *IP*. Dengan begitu, pengaksesan situs web melalui penulisan *URL* menjadi lebih sederhana. Suatu nama domain pada dasarnya terdiri atas dua bagian, yaitu nama web (misalnya “polindra”) dan ekstensi domainnya (misalnya “ac.id”). Adapun bagian-bagian dari suatu *URL* dapat dilihat pada Gambar 3.



**Gambar 3.** Bagian-bagian *URL*

- **Protokol**  
Bagian awal *URL* menunjukkan protokol yang digunakan. Penulisan “*http*” ditujukan untuk menginformasikan kepada *server* agar masuk ke “mode web”. Adapun *URL* yang diawali dengan “*https*” itu ditujukan untuk keamanan (*secure*), artinya protokol *http* telah dimodifikasi dengan cara mengenkripsi bentuk informasi yang dikirim dari *client-side* ke *server-side*.
  - **Nama situs**  
Bagian *URL* yang kedua setelah informasi protokol adalah informasi situs *web* dengan nama domainnya. Penulisan nama *host* (“*www*”) di bagian awal adalah suatu konvensi, bukan sebagai aturan, jadi dapat dihilangkan atau tidak ditulis. Suatu nama domain dapat berisi lebih dari satu situs web (disebut *sub-domain*), misalnya “*ti.polindra.ac.id*” atau “*docs.google.com*”.
  - **Absolute path**  
Bagian *URL* yang ketiga adalah jalur absolut atau *absolute path* untuk menuju suatu *file* pada direktori *server*, misalnya *first.html*. Kata-kata yang dipisahkan dengan garis miring (*slash*) adalah nama direktori, dimulai dengan direktori *root* dari suatu *host*.
2. Pengiriman *request* ke *server*  
*Browser* selanjutnya melakukan permintaan ke *server* melalui alamat IP tersebut dengan menggabungkan beberapa data atau informasi yang mendefinisikan seperti apa *request* dan *response* harus terlihat, data mana yang dapat disertakan dan dalam bentuk apa, serta bagaimana *request* akan diajukan. Setelah itu *server* kemudian menangani *request* dengan memberikan tanggapan yang berisi data atau metadata melalui *response*.
  3. Penguraian (*parsing*) *response* dari *server*  
Sampai di tahap ini *browser* hanya menerima tanggapan dari *server* saja tanpa menampilkan apapun di layar. *Browser* memeriksa data dan metadata yang disertakan dalam *response* yang umumnya berjenis teks atau *HTML (HyperText Markup Language)*.
  4. *Rendering* (menyajikan) halaman  
Ketika semua data maupun metadata telah diterima dari *server* melalui *response*, maka selanjutnya *browser* menyajikannya ke dalam suatu tampilan. Struktur halaman diatur melalui *tag HTML*, sedangkan tampilan diatur melalui *rule CSS (Cascading Style Sheet)*, dan interaksi diatur melalui bahasa pemrograman *JavaScript*.



### Kode Scripting dan Tools Pengembangan Aplikasi Web

Halaman web yang ditampilkan oleh *browser* pada dasarnya adalah hasil *rendering* dari beberapa *file* misalnya *HTML*, *CSS*, dan *JavaScript*. Bahasa kode *scripting* dapat terbagi menjadi dua berdasarkan lingkungannya, yaitu *client-side* dan *server-side*. Adapun rincian bahasa tersebut beserta *tools* yang mendukungnya disajikan pada Tabel 1.1.

**Tabel 1.1.** Kode *scripting* dan *tools* berdasarkan lingkungan

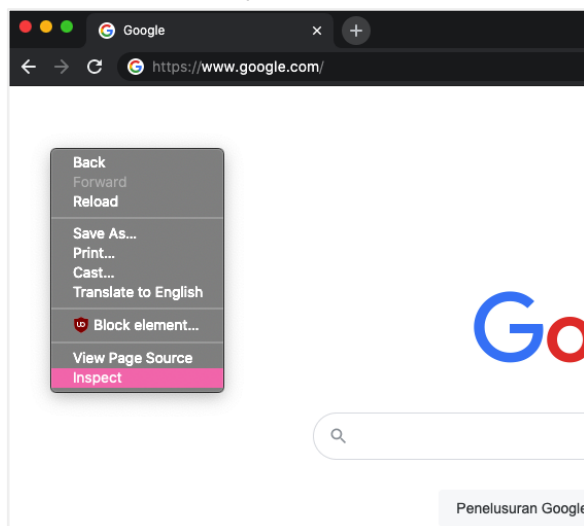
Kategori	Client-side	Server-side
Kode <i>scripting</i>	HTML, CSS, JavaScript	PHP, Java, Python, GoLang, JavaScript
Tools	<u>Browser:</u> Chrome, Firefox, Safari, Opera	<u>Web Server:</u> Apache HTTP Server, NginX, Tomcat  <u>Database Management System (DBMS):</u> MySQL, Firebase, PostgreSQL, MongoDB  <u>Paket Web Server:</u> XAMPP, LAMP, WAMP, MAMP
Text editor	VisualStudio Code (VSCode), SublimeText, Atom, Bracket, Notepad++	

Dalam proses pembuatan kode *scripting*, pada dasarnya dapat dibuat menggunakan suatu *text editor* sederhana misalnya *Notepad* atau *Notes*. Namun, akan diperlukan usaha yang besar untuk mengelolanya, terlebih apabila kode yang dibuat menjadi lebih banyak dan kompleks. Oleh karena itu, dibutuhkan suatu *text editor* yang khusus untuk mengelola kode *scripting*, misalnya VisualStudio Code (VSCode), SublimeText, Atom, Bracket, Notepad++.

### 1.3. PELAKSANAAN PRAKTIKUM

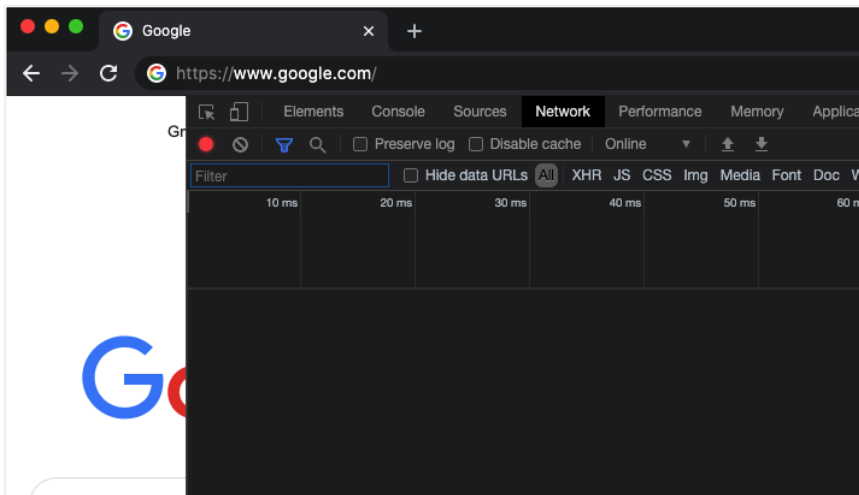
#### Langkah-langkah praktikum cara kerja web

1. Buka aplikasi *browser* (diasumsikan sudah terinstal), misalnya *Google Chrome*
2. Klik kanan pada layar *browser* > *inspect* atau *inspect element*

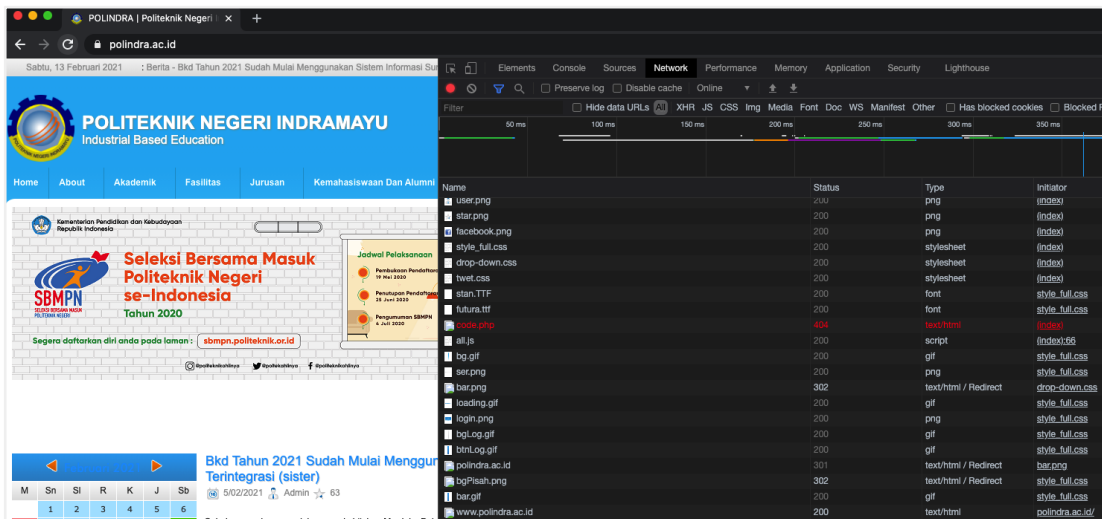




3. Pilih tab *Network*

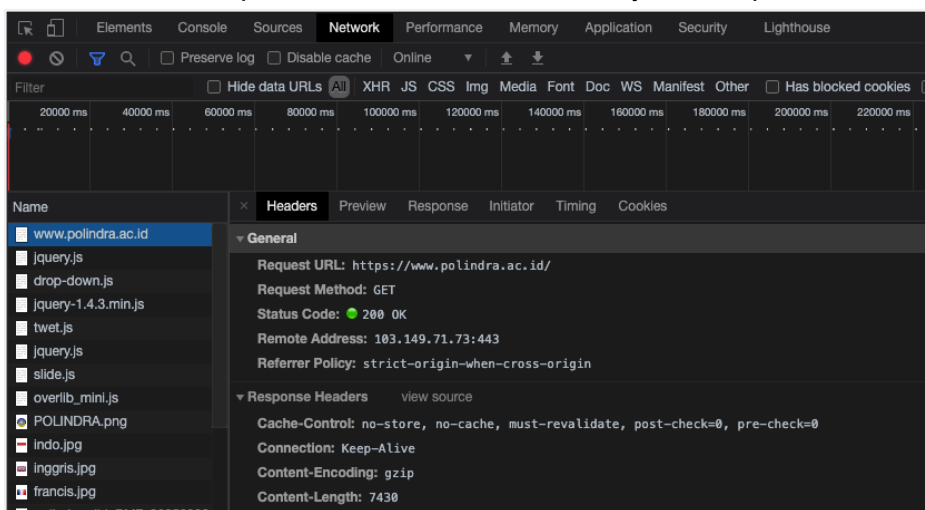


4. Masukkan suatu *URL* atau nama domain tertentu, misalnya *polindra.ac.id*



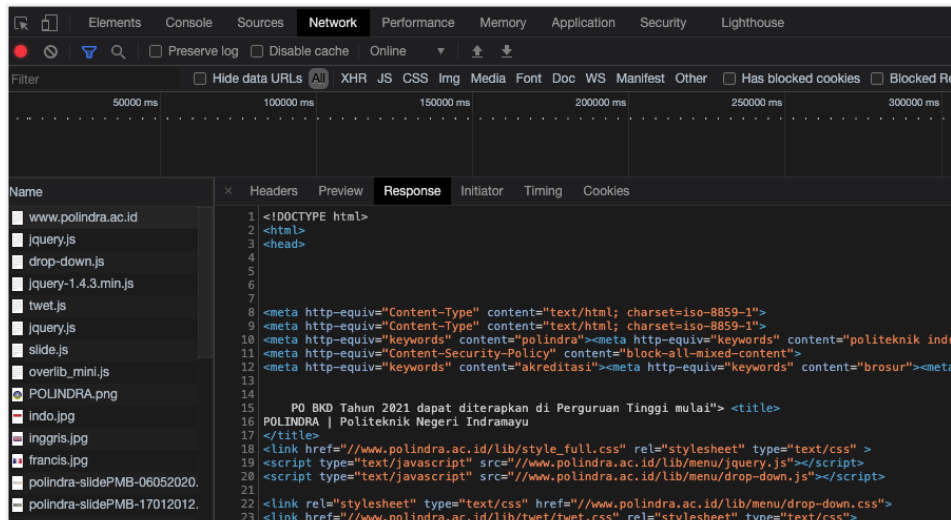
5. Semua *file* yang diterima *browser* dari *server* (*response*) akan terlihat

6. Pilih salah satu *request* dari kolom *name*, misalnya “*www.polindra.ac.id*”



7. *Metadata* dapat dilihat pada tab *Headers*, yang berisi *General*, *Response Headers*, dan *Request Headers*.

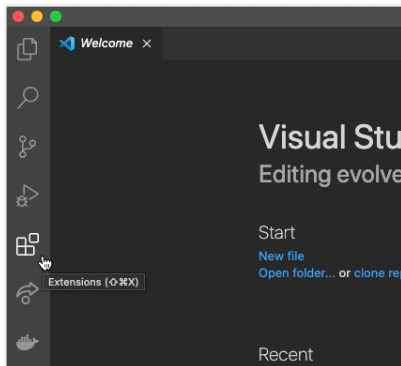
8. Adapun untuk melihat *Response* secara lebih rinci (*kode scripting*) dari setiap *file*, pilih tab *Response*



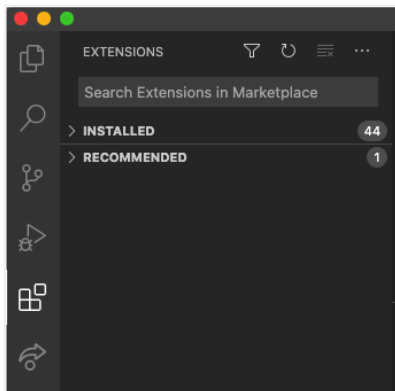
9. Selesai

### Langkah-langkah praktikum instalasi dan penggunaan *text editor*

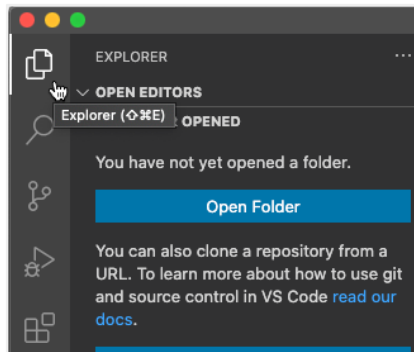
1. Unduh *installer text editor VisualStudio Code (VSCode)* yang terbaru dan sesuai dengan sistem operasi yang digunakan
2. Pilih *icon Extension*



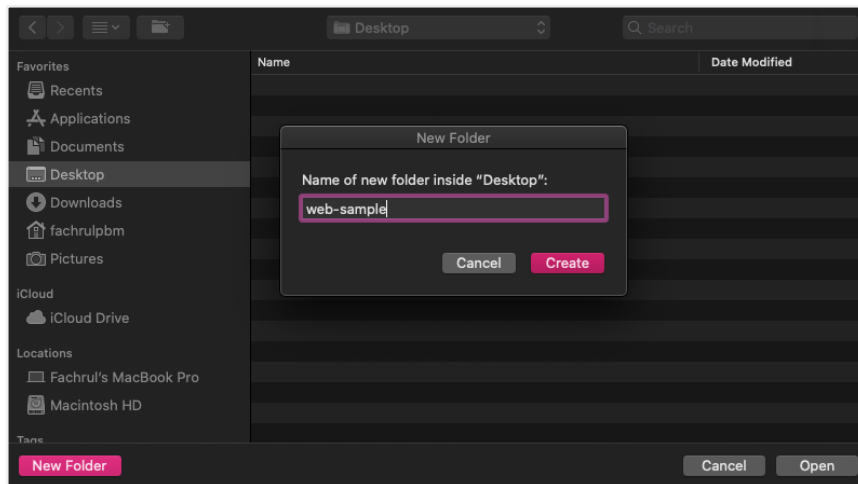
3. *Install beberapa extension* pada VSCode antara lain Live Server, Beautify, Auto Close Tag, ESLint, JavaScript (ES6) Code Snippets, VSCode-icons, Prettier, Bracket Pair Colorizer 2, Auto Rename Tag, dan Indent-Rainbow pada kolom pencarian *extension*



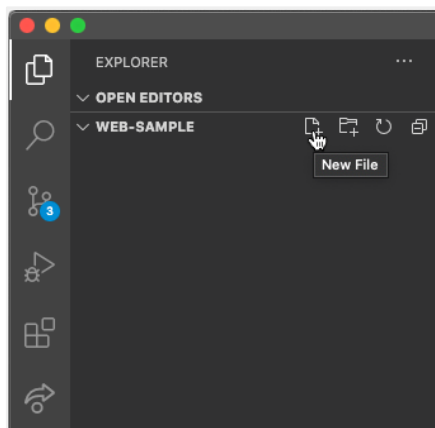
4. Pilih menu *icon explorer*



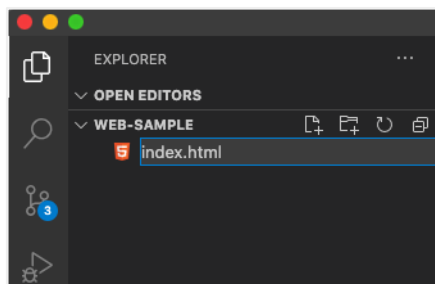
5. Pilih *Open Folder*
6. Buatlah sebuah *folder* baru beri nama “web-sample”, klik Open



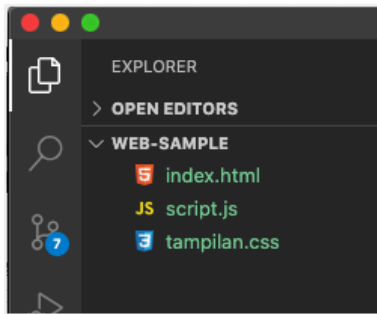
7. Buatlah *file* baru dengan klik *icon New File*



8. Beri nama *index.html*



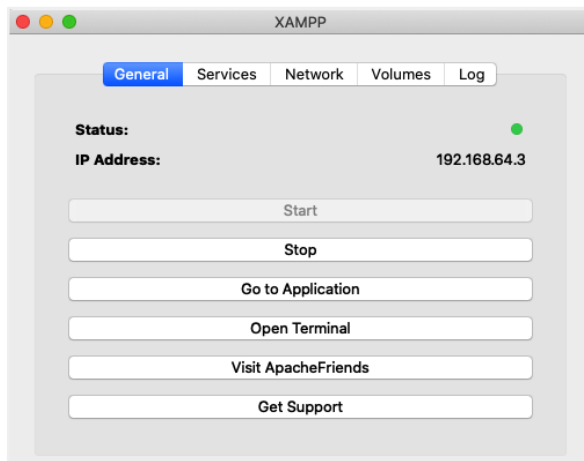
9. Buatlah *file* lainnya, yaitu *tampilan.css* dan *script.js*



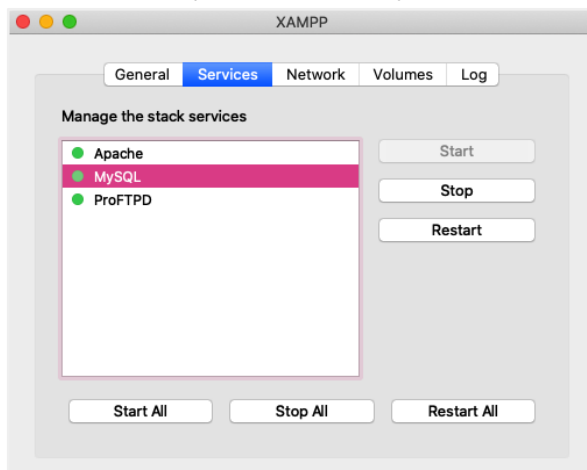
10. Selesai

### Langkah-langkah praktikum instalasi **XAMPP** dan penggunaannya

1. Unduh *installer* paket web server **XAMPP** yang terbaru dan sesuai dengan sistem operasi yang digunakan
2. Klik tombol *Start* pada tab *General* untuk mengaktifkan web server Apache HTTP Server

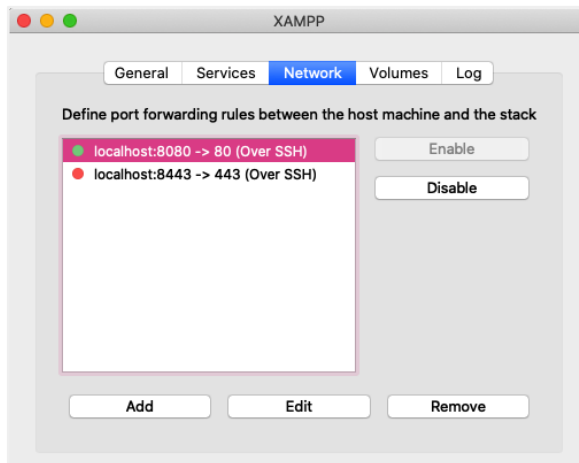


3. Klik tombol *Start All* pada tab *Service* untuk mengaktifkan layanan web server Apache dan layanan DBMS MySQL

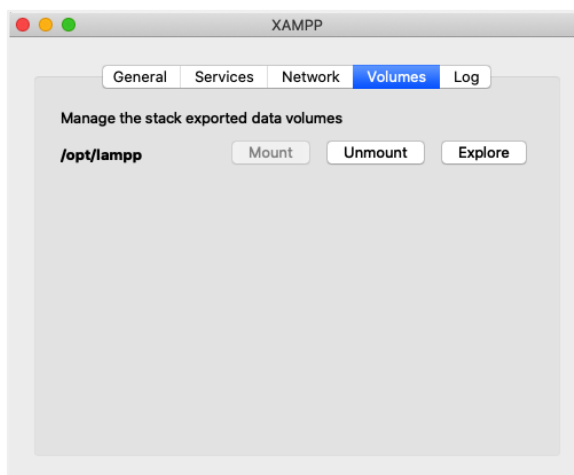


4. Klik tombol *Enable* di salah satu nama situs lokal (*localhost*), misalnya dengan port 8080, pada tab *Network*

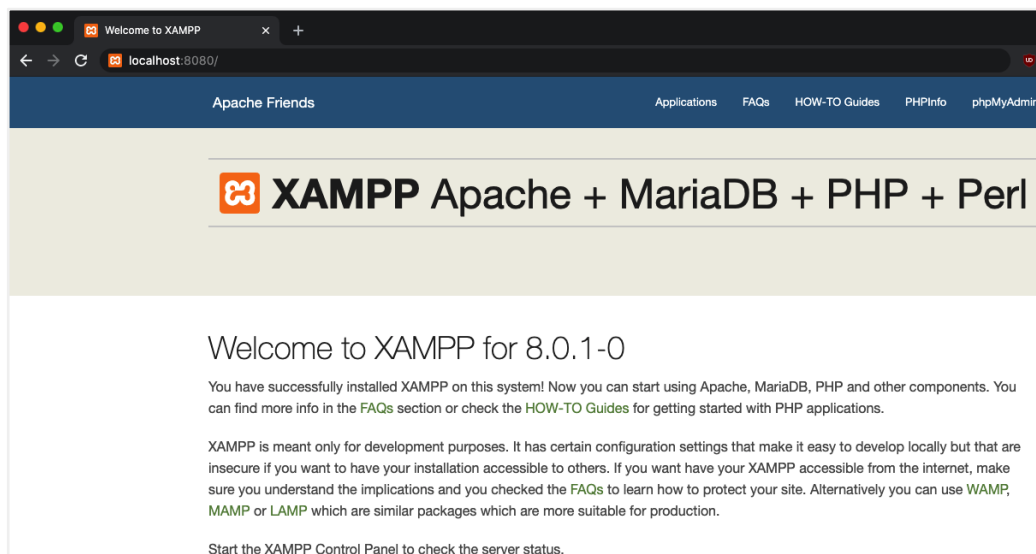




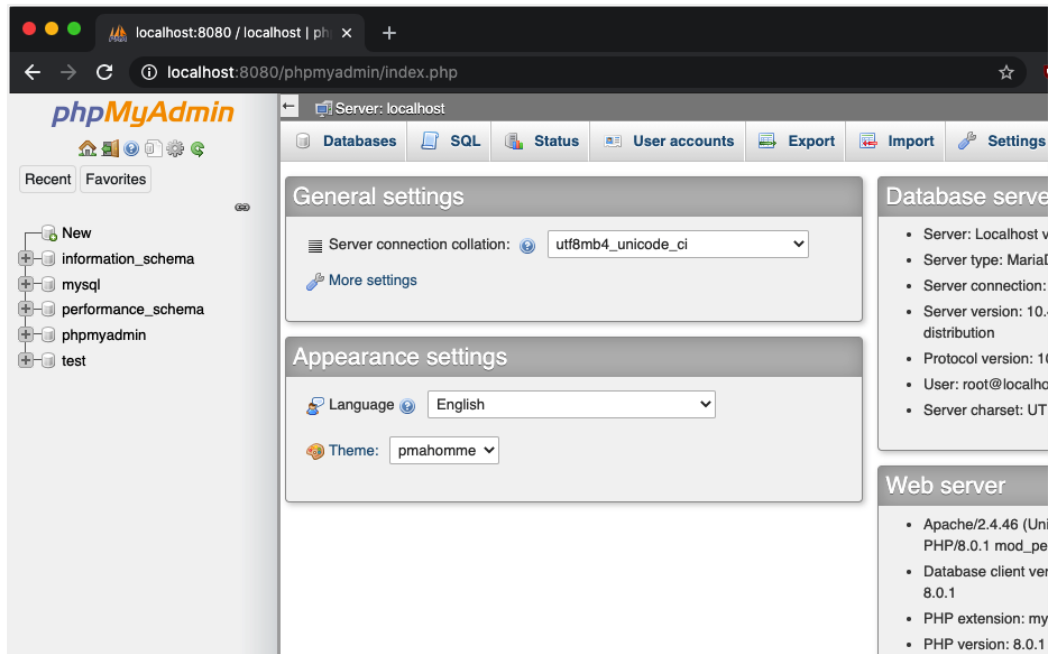
5. Klik tombol *Mount* pada Tab *Volumes* untuk dapat mengakses *file* pada *web server* lokal



6. Pada sistem operasi Windows, lokasi *file web server* berada di C:\xampp\htdocs  
7. Buka *browser*, masukkan alamat situs *localhost:8080* untuk menampilkan halaman dasar *XAMPP*



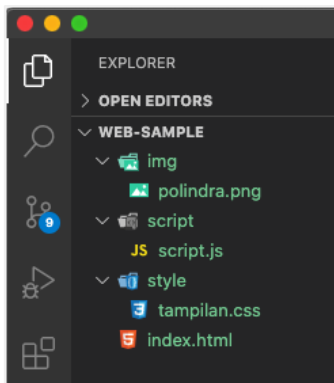
8. Modifikasi alamat *localhost* dengan menambahkan *directory path phpmyadmin*, menjadi *localhost:8080/phpmyadmin*, untuk menampilkan halaman pengelolaan basis data pada *DBMS MySQL*



9. Selesai

## 1.4. LATIHAN

Buatlah susunan *folder* yang berisi beberapa *file* sesuai dengan struktur *folder* berikut, serta jelaskan langkah-langkahnya!



## 1.5. TUGAS INDIVIDU

1. Jelaskan dan berikan contoh macam-macam ekstensi domain! (minimal 10)
2. Jelaskan bagaimana cara situs *web* di-*hosting* sehingga dapat diakses secara publik!

## 1.6. REFERENSI

- Jennifer Niederst Robbins. "Learning Web Design - A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics 5th Edition". O'Reilly Media Inc. Canada. 2018.
- Jon Duckett. "HTML & CSS - Design and Build Websites". John Wiley & Sons, Inc. Canada. 2011.
- Yasin, K. "Pengertian HTTP Beserta Fungsi dan Cara Kerjanya". 2019.  
<https://www.niagahoster.co.id/blog/pengertian-http/>. Diakses tanggal 11 Februari 2021.



- Maximilian Schwarzmuller. "How The Web Works". 2019.  
<https://academind.com/tutorials/how-the-web-works/>. Diakses tanggal 11 Februari 2021.
- Imas Indra. "Apa itu URL? Pengertian URL dan Fungsinya". 2019.  
<https://www.niagahoster.co.id/blog/apa-itu-url/>. Diakses tanggal 11 Februari 2021.
- Alya Iqlima, "Apa itu Domain". 2020. <https://www.niagahoster.co.id/blog/apa-itu-domain/>. Diakses tanggal 12 Februari 2021.
- Zane L. "How-to View Response Headers". 2021.  
<https://support.stackpath.com/hc/en-us/articles/360001368263-How-To-View-Response-Headers>. Diakses tanggal 12 Februari 2021.