

## HTTPS

Hypertext Transfer Protocol Secure (HTTPS) sebenarnya hanyalah protokol HTTP tetapi dengan enkripsi data menggunakan Secure Sockets Layer (SSL) atau yang lebih baru dikenal dengan Transport Layer Security (TLS). Berbeda dengan HTTP, secara default, HTTPS berjalan pada port 443.

Mengapa harus menggunakan HTTPS ?

1. Keamanan – Mengamankan data antara client (browser) dengan server serta menjaga integritas website dari peretas yang dapat menginjeksi malware.
2. Search Engine Optimization (SEO) – Sejak tahun 2014 crawler mesin pencari Google lebih menyukai website yang menggunakan HTTPS.
3. Web Masa Depan – Teknologi web masa depan, mulai dari API, AMP, PWA semuanya diharuskan menggunakan HTTPS.

Pada praktikum kali ini, kita akan mencoba mengkonfigurasi HTTPS pada webserver lokal.

Berikut ini beberapa perangkat lunak yang digunakan :

- OS Linux Ubuntu 18.04 64-bit (boleh menggunakan distro berbeda)
- Apache
- brew
- mkcert
- Browser

Tool yang kita digunakan agar HTTPS dapat berjalan di lingkungan localhost adalah mkcert. Mkcert merupakan sebuah tool untuk membuat sertifikat SSL secara lokal dan dipercaya oleh web browser. Mkcert dapat membuat sertifikat untuk host localhost, custom host, atau IP address seperti 127.0.0.1. Mkcert mendukung sistem operasi Linux, macOS, dan Windows.

### Instalasi Webserver Apache

Apache adalah webserver yang paling banyak digunakan di dunia. Menyediakan banyak fitur canggih, termasuk modul yang dapat dimuat secara dinamis, dukungan media yang kuat, dan integrasi luas dengan perangkat lunak populer lainnya.

Apache tersedia dalam repositori perangkat lunak default Ubuntu, sehingga Anda dapat menginstalnya menggunakan *package management tools* konvensional.

1. Perbarui terlebih dahulu indeks paket lokal Anda  
`$ sudo apt update`
2. Install package Apache2  
`$ sudo apt install apache2`

## Sesuaikan Firewall

Agar webserver dapat diakses dari luar komputer, maka perlu penyesuaian terhadap konfigurasi firewall.

1. Periksa profil aplikasi ufw yang tersedia

```
$ sudo ufw app list
```

### Output

Available applications:

```
Apache
Apache Full
Apache Secure
OpenSSH
```

2. Aktifkan profil "Apache Full" agar firewall mengizinkan lalu lintas di port 80 (lalu lintas web normal, tidak terenkripsi) dan 443

```
$ sudo ufw allow 'Apache Full'
```

3. Aktifkan ufw dengan perintah

```
$ sudo ufw enable
```

4. Cek perubahan firewall dengan perintah

```
$ sudo ufw status
```

### Output

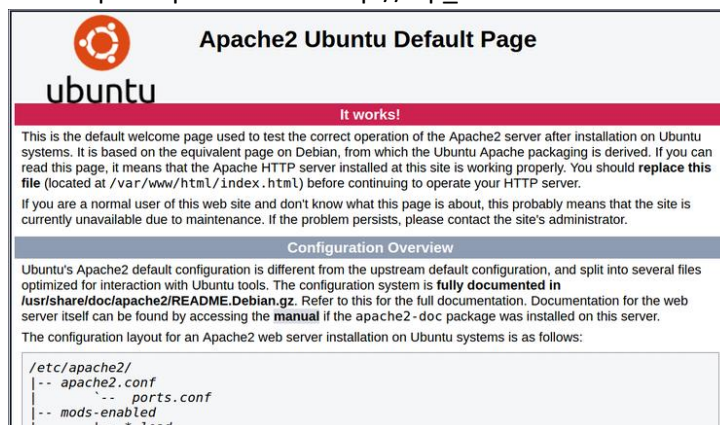
Status: active

To	Action	From
--	-----	----
Apache	ALLOW	Anywhere
Apache Full	ALLOW	Anywhere
Apache (v6)	ALLOW	Anywhere (v6)
Apache Full (v6)	ALLOW	Anywhere (v6)

5. Cek status webserver

```
$ sudo systemctl status apache2
```

6. Apabila apache telah berjalan dengan benar maka Anda dapat mengakses direktori root webserver menggunakan browser di alamat <http://localhost> atau <http://127.0.0.1>. Apabila konfigurasi firewall benar, webserver juga dapat diakses dari komputer lain menggunakan alamat ip komputer server `http://<ip_server>`.



## Instalasi Brew

Mkcert di install menggunakan *brew package manager*. Homebrew (brew) sendiri pada awalnya adalah package manager yang dirancang untuk memudahkan instalasi perangkat lunak di macOS. Brew dikembangkan dalam bahasa pemrograman Ruby dan secara default sudah tersedia di macOS. Homebrew menggunakan lisensi open source sehingga terbuka untuk dikembangkan oleh siapa pun. Ribuan kontributor telah mengembangkan Homebrew agar dapat berjalan di sistem operasi berbasis Linux yang diberi nama Linuxbrew. Berikut ini langkah-langkah instalasi brew di Linux.

1. Install paket dependensi yang dibutuhkan

Debian/Ubuntu

```
$ sudo apt-get install build-essential curl file git
```

Fedora/CentOS/Red Hat

```
$ sudo yum groupinstall 'Development Tools' && sudo yum install curl file git
```

2. Unduh kemudian install brew

```
$ sh -c "$(curl -fsSL https://raw.githubusercontent.com/Linuxbrew/install/master/install.sh)"
```

3. Setelah proses instalasi selesai, jalankan perintah dibawah ini

```
$ test -d ~/.linuxbrew && eval "$(~/.linuxbrew/bin/brew shellenv)
$ test -d /home/linuxbrew/.linuxbrew && eval "$(~/.linuxbrew/bin/brew shellenv)
$ test -r ~/.bash_profile && echo "eval \"\$(brew --prefix)/bin/brew shellenv\"" >> ~/.bash_profile
$ echo "eval \"\$(brew --prefix)/bin/brew shellenv\"" >> ~/.profile
```

## Instalasi mkcert

1. Install paket dependensi

```
$ sudo apt-get install libnss3-tools
```

2. Install mkcert via brew, tunggu hingga proses unduh dan install selesai.

```
$ brew install mkcert
```

3. Install Certificate Authority (CA)

```
$ mkcert -install
```

### Output

```
Created a new local CA at "/home/<nama_user>/.local/share/mkcert" 🦊
```

```
Sudo password:
```

```
The local CA is now installed in the system trust store! ⚡
```

```
The local CA is now installed in the Firefox and/or Chrome/Chromium trust store (requires browser restart)! 🦊
```

4. Selanjutnya buat sertifikat menggunakan mkcert untuk localhost.

```
$ mkcert localhost
```

### Output

```
Using the local CA at "/home/akeo/.local/share/mkcert" 🦊
```

```
Created a new certificate valid for the following names 📄
```

```
- "localhost"
```

```
The certificate is at "./localhost.pem" and the key at "./localhost-key.pem" ✅
```

## Konfigurasi SSL pada Apache

1. Pada langkah sebelumnya kita telah mendapatkan pasangan kunci dan sertifikat. Selanjutnya kita pindahkan keduanya ke direktori ssl.

```
$ sudo mv localhost.pem /etc/ssl/certs  
$ sudo mv localhost-key.pem /etc/ssl/private
```

2. Konfigurasi virtualhost localhost untuk port 443 (HTTPS)

```
$ sudo nano /etc/apache2/sites-available/default-ssl.conf
```

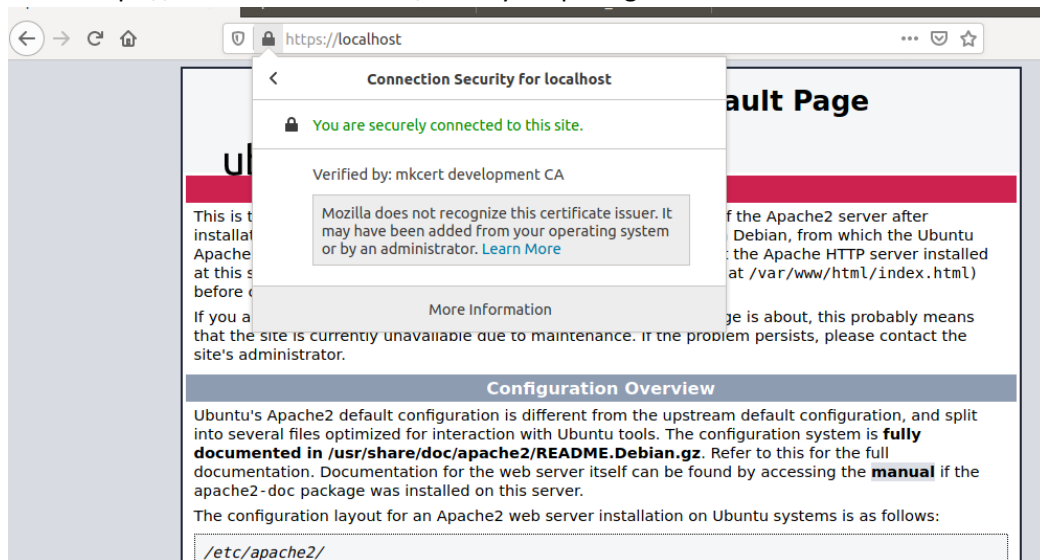
3. Cari opsi sertifikat dan ubah seperti di bawah ini.

```
SSLCertificateFile      /etc/ssl/certs/localhost.pem  
SSLCertificateKeyFile   /etc/ssl/private/localhost-key.pem
```

4. Aktifkan modul SSL untuk Apache dan virtualhost default-ssl.

```
$ sudo a2enmod ssl  
$ sudo a2ensite default-ssl.conf  
$ sudo systemctl restart apache2  
$ sudo systemctl status apache2
```

5. Akses https://localhost di browser, hasilnya seperti gambar di bawah ini.



## Tugas.

Modifikasi virtualhost lokal menjadi domain yang Anda inginkan, kemudian aktifkan HTTPS pada domain tersebut. Baik melalui HTTP atau HTTPS nama domain dapat diakses dari komputer berbeda. Tuliskan langkah-langkah yang Anda lakukan seperti layaknya sebuah tutorial. Lengkapi dengan tangkapan layar atau perintah-perintah yang digunakan. Kumpulkan tugas dalam format pdf.

