Nama : Farhan Dwi Septian
NIM : 105221036

Post Test Praktikum 1 Pemrograman Web

HTTP

Hypertext Transfer Protocol (HTTP) merupakan sebuah protokol jaringan aplikasi yang digunakan untuk mendistribusikan informasi antara server dengan client. Server disini yang dimaksud adalah jenis web server dengan bentuk fisik jaringan komputer yang memiliki kapasitas penyimpanan data berskala besar. Selanjutnya yang berperan sebagai client adalah web browser yang dapat mengakses, menerima hingga menampilkan konten web melalui browser.

Bagaimana cara kerja HTTP?

Sebagai permulaan, HTTP bekerja sepenuhnya melalui pesan HTTP. Ada dua jenis pesan HTTP:

- Request: Ini adalah pesan yang dikirim oleh klien ke server untuk memicu suatu tindakan
- Response: Ini adalah pesan yang dikirim oleh server ke klien sebagai respons terhadap pesan permintaan

Kita dapat melihat bahwa pertukaran pesan-pesan ini dalam siklus request/respons memungkinkan pertukaran data antara klien dan server.

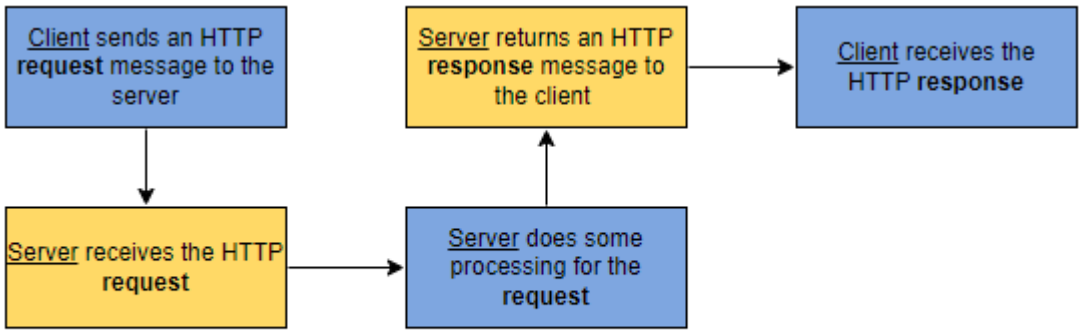

Figure 1: Request/response cycle

Gambar 2 di bawah ini mengilustrasikan garis besar pesan HTTP pada umumnya. Pesan ini dibagi menjadi tiga bagian: baris awal, header, dan isi. Namun, perlu diingat bahwa pesan permintaan dan respons memiliki beberapa perbedaan utama dalam konten yang dimasukkan ke dalam masing-masing bidang ini yang akan dibahas di bagian masing-masing.

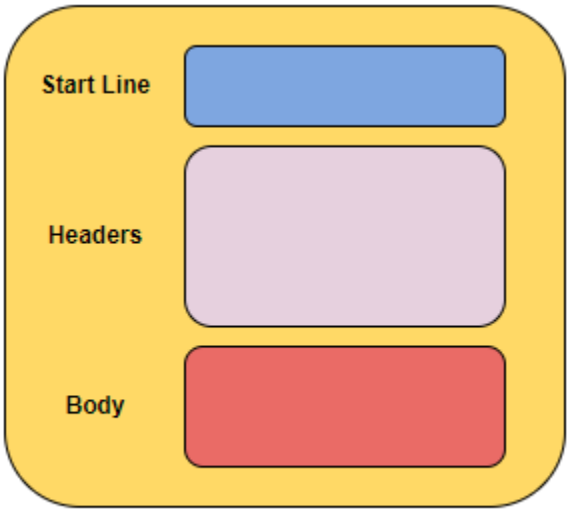

Figure 2: A typical HTTP message

Pesan permintaan HTTP

Gambar 3 menggambarkan struktur pesan permintaan HTTP yang umum dan data yang ada di dalam field-fieldnya.

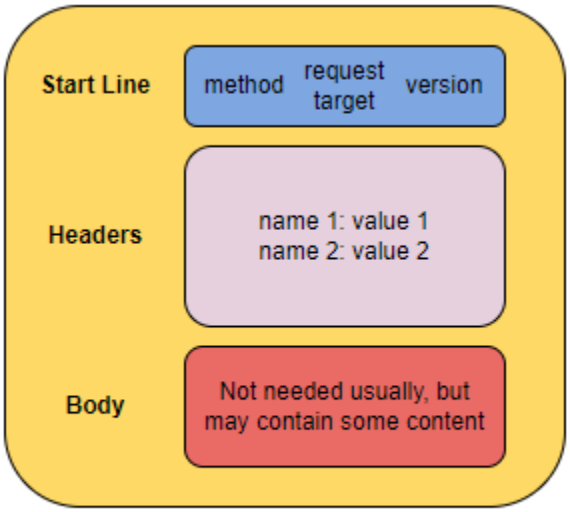

Figure 3: An HTTP request message

Start Line:

Seperti yang ditunjukkan pada Gambar 3, baris awal untuk pesan permintaan berisi hal-hal berikut:

1. HTTP Method: GET, POST, PUT, DELETE
2. Request Target
3. HTTP Version

Header:

Header untuk pesan permintaan dan respons berupa satu atau lebih pasangan nama:nilai. Pasangan ini dapat dibagi menjadi beberapa kelompok:

- General
- Request
- Representation

Body

Badan pesan permintaan sering kali kosong karena sebagian besar metode HTTP (seperti GET dan DELETE) meminta sumber daya dari server. Metode permintaan seperti PUT menyertakan sumber daya yang akan disimpan di server.

Pesan respons HTTP

Gambar 3 menggambarkan struktur pesan respons HTTP yang khas dan data yang dikandungnya di dalam field-fieldnya.

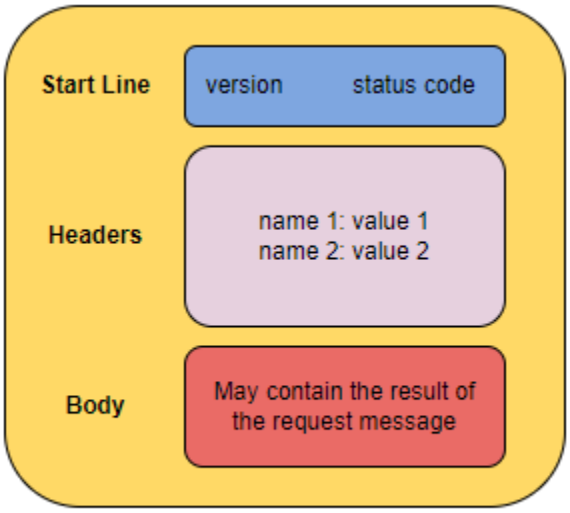

Figure 4: An HTTP response message

Status Code:

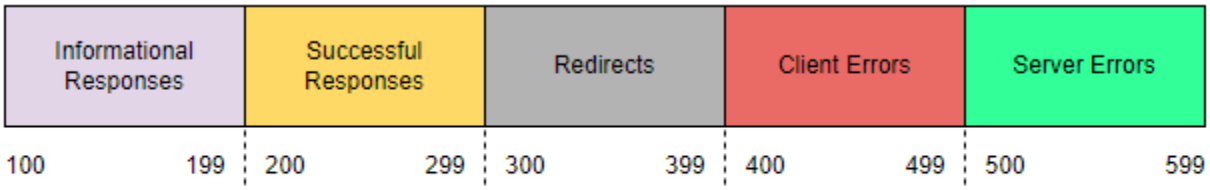

Figure 5: Status codes

HTTPS:

Hypertext transfer protocol secure (HTTPS) adalah versi aman dari HTTP, yang merupakan protokol utama yang digunakan untuk mengirim data antara peramban web dan situs web. HTTPS dienkripsi untuk meningkatkan keamanan transfer data. Hal ini sangat penting terutama ketika pengguna mengirimkan data sensitif, seperti masuk ke rekening bank, layanan email, atau penyedia asuransi kesehatan.

Situs web apa pun, terutama yang memerlukan kredensial login, harus menggunakan HTTPS. Pada peramban web modern seperti Chrome, situs web yang tidak menggunakan HTTPS ditandai dengan cara yang berbeda dari yang menggunakan HTTPS. Carilah gembok di bilah URL untuk menandakan bahwa halaman web tersebut aman. Peramban web menganggap HTTPS sebagai hal yang serius; Google Chrome dan peramban lainnya menandai semua situs web non-HTTPS sebagai tidak aman.

Bagaimana cara kerja HTTPS?

HTTPS menggunakan protokol enkripsi untuk mengenkripsi komunikasi. Protokol ini disebut Transport Layer Security (TLS), meskipun sebelumnya dikenal sebagai Secure Sockets Layer (SSL). Protokol ini mengamankan komunikasi dengan menggunakan apa yang dikenal sebagai infrastruktur kunci publik asimetris. Jenis sistem keamanan ini menggunakan dua kunci yang berbeda untuk mengenkripsi komunikasi antara dua pihak:

- The Private Key - kunci ini dikendalikan oleh pemilik situs web dan dijaga kerahasiaannya, seperti yang mungkin pembaca duga, bersifat pribadi. Kunci ini berada di server web dan digunakan untuk mendekripsi informasi yang dienkripsi oleh kunci publik.
- The Public Key- kunci ini tersedia untuk semua orang yang ingin berinteraksi dengan server dengan cara yang aman. Informasi yang dienkripsi oleh kunci publik hanya dapat didekripsi oleh kunci privat.

Mengapa HTTPS penting? Apa yang terjadi jika situs web tidak memiliki HTTPS?

HTTPS mencegah situs web menyiarkan informasinya dengan cara yang mudah dilihat oleh siapa pun yang mengintip di jaringan. Ketika informasi dikirim melalui HTTP biasa, informasi tersebut dipecah menjadi paket-paket data yang dapat dengan mudah "diendus" menggunakan perangkat lunak gratis. Hal ini membuat komunikasi melalui media yang tidak aman, seperti Wi-Fi publik, sangat rentan terhadap penyadapan. Faktanya, semua komunikasi yang terjadi melalui HTTP terjadi dalam bentuk teks biasa, membuatnya sangat mudah diakses oleh siapa saja dengan alat yang tepat, dan rentan terhadap serangan di jalur.

Dengan HTTPS, lalu lintas dienkripsi sedemikian rupa sehingga meskipun paket-paket tersebut diendus atau dicegat, mereka akan muncul sebagai karakter yang tidak masuk akal. Mari kita lihat sebuah contoh:

Sebelum enkripsi:

Ini adalah serangkaian teks yang sepenuhnya dapat dibaca

Setelah enkripsi:

ITM0IRyiEhVpa6VnKyExMiEgNveroyWBPlgGyfkflYjDaaFf/Kn3bo3OfghBPDWo6AfSHlNtL
8N7ITEwIXc1gU5X73xMsJormzzXlwOyrCs+9XCPk63Y+z0=

Pada situs web tanpa HTTPS, penyedia layanan Internet (ISP) atau perantara lain dapat menyuntikkan konten ke dalam halaman web tanpa persetujuan dari pemilik situs web. Hal ini biasanya terjadi dalam bentuk iklan, di mana ISP yang ingin meningkatkan pendapatan menyuntikkan iklan berbayar ke dalam halaman web pelanggan mereka. Tidak mengherankan, ketika hal ini terjadi, keuntungan dari iklan dan kontrol kualitas dari iklan tersebut sama sekali tidak dibagikan kepada pemilik situs web. HTTPS menghilangkan kemampuan pihak ketiga yang tidak dimoderasi untuk menyuntikkan iklan ke dalam konten web.

DNS

Domain Name System (DNS) adalah salah satu fondasi internet, yang bekerja di latar belakang untuk mencocokkan nama-nama situs web yang diketikkan orang ke dalam kotak pencarian dengan alamat IP yang sesuai, sebuah deretan angka panjang yang tidak mungkin bisa diingat oleh siapa pun.

Masih mungkin bagi seseorang untuk mengetikkan alamat IP ke dalam peramban untuk membuka sebuah situs web, tetapi kebanyakan orang ingin alamat internet terdiri dari kata-kata yang mudah diingat, yang disebut nama domain. (Sebagai contoh, Network World).

Bagaimana cara kerja DNS?

Ketika komputer Anda ingin menemukan alamat IP yang terkait dengan nama domain, komputer pertama-tama membuat permintaan DNS melalui klien DNS, biasanya di browser Web. Permintaan tersebut kemudian diteruskan ke server DNS rekursif, yang juga dikenal sebagai resolver rekursif. Penyelesai rekursif biasanya dioperasikan oleh Internet Service Provider (ISP), seperti AT&T atau Verizon (atau pihak ketiga lainnya), dan mengetahui server DNS mana yang harus ditanyakan untuk menyelesaikan nama situs dengan alamat IP-nya. Server yang benar-benar memiliki informasi yang dibutuhkan disebut server nama otoritatif.

DNS diatur dalam sebuah hirarki. Permintaan DNS awal untuk alamat IP dilakukan ke resolver rekursif. Pencarian ini pertama-tama mengarah ke server root, yang memiliki informasi tentang domain tingkat atas (.com, .net, .org), serta domain negara. Server root berlokasi di seluruh dunia, sehingga sistem DNS merutekan permintaan ke server terdekat.

Setelah permintaan mencapai server root yang benar, permintaan akan diteruskan ke server domain tingkat atas (server nama TLD), yang menyimpan informasi untuk domain tingkat kedua, yaitu kata-kata yang Anda ketik di kotak pencarian. Permintaan kemudian diteruskan ke server nama domain, yang mencari alamat IP dan mengirimkannya kembali ke perangkat klien DNS sehingga dapat mengunjungi situs web yang sesuai. Semua ini hanya membutuhkan waktu beberapa milidetik saja.

Secara Umum Cara Kerja DNS adalah:

1. Permintaan DNS: Ketika Anda memasukkan sebuah URL (Uniform Resource Locator) atau nama domain dalam browser web Anda, komputer Anda perlu tahu alamat IP yang sesuai untuk menghubungi situs web tersebut. Jadi, komputer Anda mengirim permintaan DNS ke server DNS.
2. Caching: Pertama, komputer Anda akan memeriksa apakah sudah ada catatan DNS yang disimpan dalam cache lokalnya. Jika ya, dan catatan DNS tersebut masih valid (belum kadaluwarsa), maka komputer akan menggunakan informasi yang tersimpan dalam cache tanpa perlu melakukan permintaan ke server DNS eksternal. Ini dapat menghemat waktu dan meningkatkan kinerja.
3. Resolusi Hierarki: Jika informasi DNS tidak ada dalam cache lokal atau sudah kadaluwarsa, komputer akan mengirim permintaan DNS ke server DNS tingkat atas, yang dikenal sebagai Root DNS Server. Root DNS Server adalah bagian paling atas dari hierarki DNS.
4. Tingkat TLD (Top-Level Domain): Setelah menerima permintaan dari komputer Anda, Root DNS Server akan merujuk permintaan tersebut ke server DNS tingkat atas yang sesuai dengan TLD yang ditemukan dalam nama domain. Contohnya, jika nama domain adalah www.example.com, server DNS tingkat atas untuk TLD ".com" akan dihubungi.
5. Otoritas DNS: Server DNS tingkat atas TLD akan memberikan alamat IP server otoritatif yang bertanggung jawab atas domain yang diminta. Jika server DNS tingkat atas ini tidak memiliki informasi tentang domain tersebut, maka permintaan akan dikirim ke server DNS tingkat atas yang lebih tinggi dalam hierarki.
6. Server Otoritatif: Server DNS otoritatif untuk domain yang diminta akan merespons dengan alamat IP yang sesuai dengan nama domain. Informasi ini kemudian dikirim kembali ke komputer Anda melalui proses yang sama, melalui server DNS tingkat atas, hingga mencapai komputer Anda.
7. Penyimpanan Cache Lokal: Setelah menerima respon dari server DNS otoritatif, komputer Anda akan menyimpan informasi tersebut dalam cache lokal untuk penggunaan mendatang. Ini membantu mempercepat akses ke situs web yang sama di masa depan.
8. Akses ke Situs Web: Setelah komputer Anda memiliki alamat IP yang sesuai, browser web Anda akan menggunakan alamat ini untuk membuat koneksi dengan server web hosting

situs web yang sesuai. Ini memungkinkan Anda mengakses dan menampilkan konten dari situs web tersebut.

Server

Server adalah program atau perangkat yang menyediakan fungsionalitas untuk klien yang disebut sebagai program atau perangkat lain. Arsitektur ini disebut model klien-server.

Satu komputasi keseluruhan didistribusikan di beberapa proses atau perangkat. Server dapat menyediakan berbagai fungsi yang disebut layanan. Layanan ini termasuk berbagi data atau sumber daya di antara beberapa klien atau melakukan perhitungan untuk klien. Beberapa klien dapat dilayani oleh satu server, dan satu klien dapat menggunakan beberapa server.

Cara Kerja Server

Perangkat perlu disiapkan untuk mendengarkan permintaan klien melalui koneksi jaringan agar dapat menjalankan peran sebagai server. Sistem operasi dapat menyertakan fungsionalitas ini sebagai aplikasi terinstal, peran, atau kombinasi keduanya.

Sistem operasi windows server dari microsoft memiliki kemampuan untuk mendengar dan merespons permintaan klien. Jenis permintaan klien yang dapat ditangani server bertambah dengan adanya role atau layanan tambahan yang terinstal. Ilustrasi lain adalah ketika aplikasi tambahan yang disebut Apache diletakkan di atas sistem operasi untuk menangani permintaan dari browser web. Klien mengirimkan permintaan melalui jaringan setiap kali membutuhkan data atau fungsionalitas dari server. Server menerima permintaan ini dan memberikan informasi yang diperlukan sebagai tanggapan. Ini adalah model permintaan dan respons jaringan klien-server, yang biasa disebut sebagai model panggilan dan respons.

Sebagai bagian dari satu permintaan dan respons, server sering kali menyelesaikan berbagai tugas tambahan, seperti mengonfirmasi identitas pemohon, memastikan klien memiliki izin untuk mengakses data atau sumber daya yang diminta, dan memformat dengan benar atau mengembalikan respons yang diperlukan dengan cara yang diharapkan.

Referensi:

https://www.dicoding.com/blog/perbedaan-http-dan-https/
https://www.educative.io/answers/how-does-http-work
https://www.cloudflare.com/learning/ssl/what-is-https/