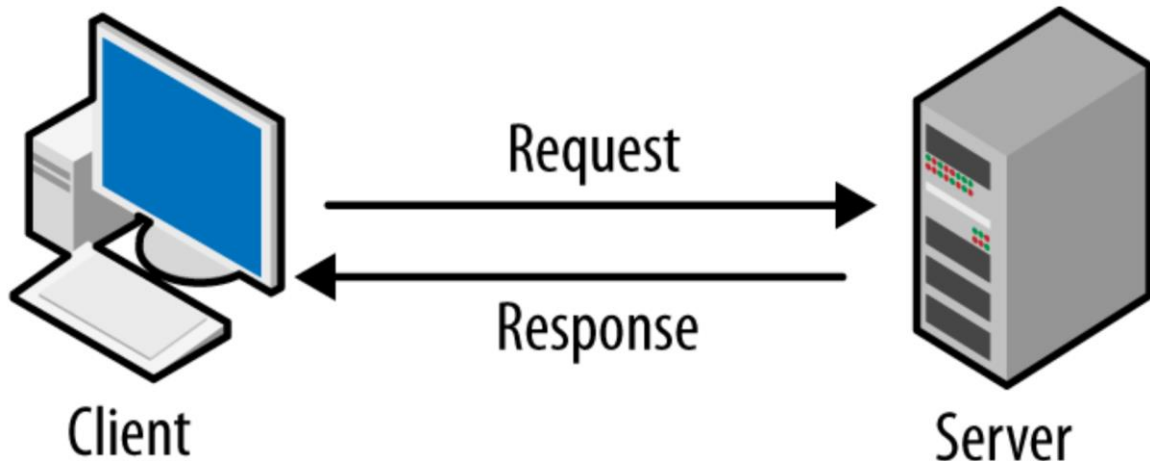


I. Apa Itu Client Server?



Client-server adalah model arsitektur jaringan yang digunakan untuk mengorganisir interaksi antara perangkat dan aplikasi dalam jaringan komputer. Pada model ini, ada dua jenis perangkat lunak yang berbeda, yaitu client dan server.

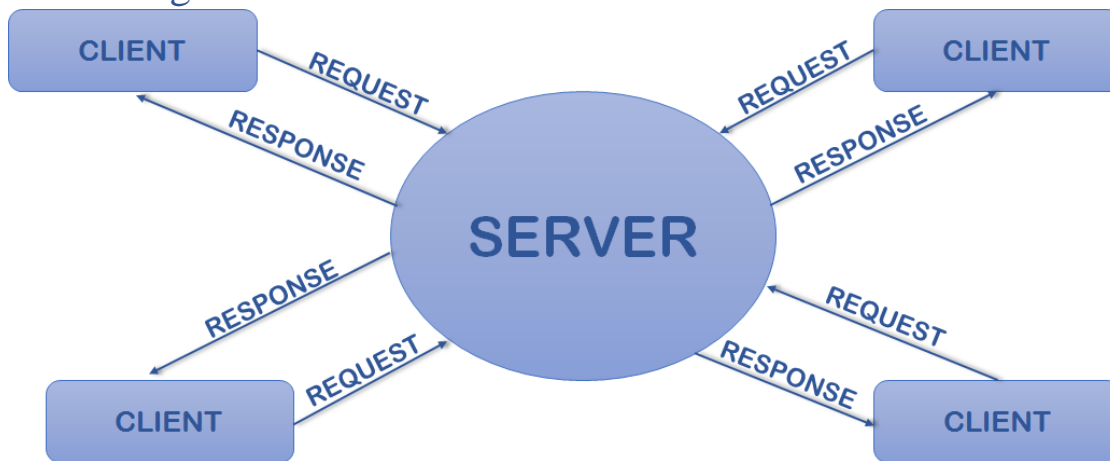
Server adalah komputer atau perangkat lunak yang menyediakan layanan atau sumber daya ke dalam jaringan. Contohnya seperti web server yang menyediakan situs web, file server yang menyimpan dan membagikan file, database server yang menyimpan dan mengakses data, dan lain sebagainya.

Client adalah komputer atau perangkat lunak yang mengakses dan menggunakan layanan atau sumber daya yang disediakan oleh server. Contohnya seperti browser web yang digunakan untuk mengakses situs web di server, aplikasi pengolah kata yang digunakan untuk mengakses file yang disimpan di file server, atau aplikasi mobile yang mengambil data dari server database.

Komunikasi antara client dan server biasanya dilakukan melalui protokol komunikasi tertentu, seperti protokol **HTTP** untuk komunikasi antara browser web dan web server, atau protokol **FTP** untuk transfer file antara client dan file server.

Keuntungan dari model client-server adalah bahwa sumber daya dan layanan dapat dikonsolidasikan dan dikelola secara efisien, dan data dapat diakses dan digunakan dari mana saja dalam jaringan. Model ini juga memungkinkan pengembangan dan pemeliharaan aplikasi menjadi lebih terpusat dan dapat diakses dengan mudah oleh pengguna yang memerlukan layanan tersebut.

II. Fungsi Client Server



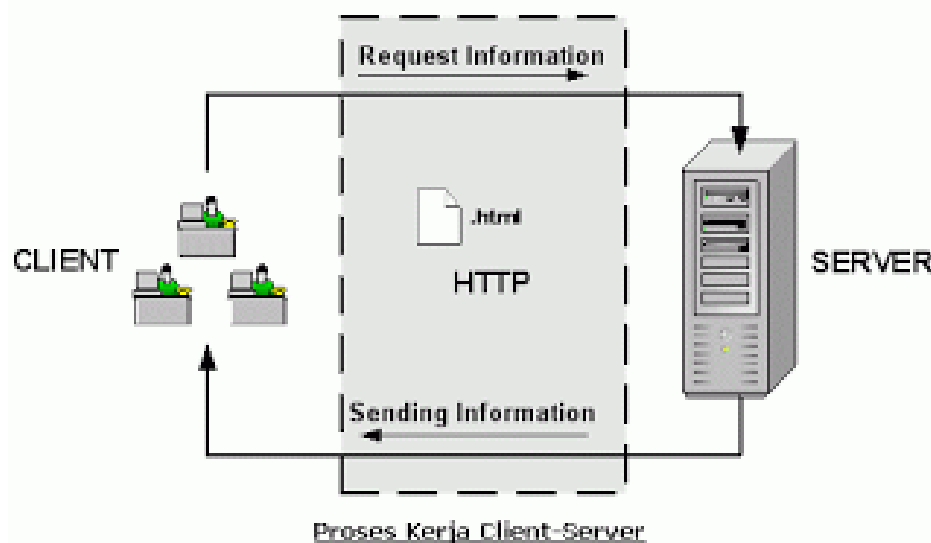
Client server mempunyai instrumen yang berfungsi untuk efisiensi bandwidth serta meningkatkan efektivitas kinerja server yang menyimpan seluruh data client.

Software yang berperan sebagai client hanya memperoleh data atau informasi sesuai dengan permintaan yang dibuatnya. Data yang dikirimkan dari server menuju client juga dapat dipastikan 99% aman karena sudah ada teknologi enkripsi.

Namun fungsi yang paling utama dari client server yaitu pengguna dapat menjalankan bisnis miliknya sendiri melalui website yang telah dirancang sebelumnya. Melalui website, client bisa membagikan informasi produknya ke seluruh dunia secara cepat.

Pengguna lainnya kemudian akan memperoleh informasi mengenai produk atau jasa dari browser yang mengakses website tersebut.

III. Cara Kerjanya



Berikut adalah cara kerja sistem client server yang dijelaskan secara lengkap seperti berikut:

1. Dilihat dari database, client dapat membangun sebuah halaman web menggunakan berbagai hardware dan software dan menghasilkan tampilan atau user interface yang menarik.
2. User interface dipengaruhi oleh peran server. Web server menerima permintaan dan menyimpan UI website tersebut ke dalam bentuk dokumen HTML.
3. Jika diminta oleh pengguna atau client, server akan dengan cepat mengirimkan dokumen HTML tersebut agar dapat ditampilkan di browser dan dilihat oleh pengguna.
4. Ketika client sudah mendapatkan permintaan dari pengguna, client akan melakukan pemeriksaan sintak atau kode program. Proses ini akan menghasilkan basis data atau database yang diperlukan. Umumnya menggunakan Structured Query Language (SQL) atau bahasa lain.
5. Proses ini akan diteruskan ke server sambil menunggu server mendapatkan respons dari pengguna akhir. Ketika pengguna telah memberikan respons, database request dibuat untuk client dan hasilnya akan ditayangkan di layar.

IV. Keuntungan dari Jaringan Client Server

Jaringan komputer yang dibuat menggunakan arsitektur client server akan memberikan banyak keuntungan seperti berikut.

1. Lokasi yang ideal untuk menyimpan big data karena client dapat mengirim serta meminta data yang berukuran sangat besar melalui arsitektur ini.
2. Meningkatkan kinerja karena seluruh data dan informasi disimpan di satu server yang memiliki kemampuan multitasking tingkat tinggi. Sehingga mampu memberikan respons kepada banyak client sekaligus.
3. Lebih hemat biaya untuk pengadaan hardware karena pengguna tidak perlu menyimpan semua data di hard drive atau penyimpanan lokal. Jadi, pengguna tidak perlu membeli perangkat penyimpanan secara terus menerus.
4. Menekan biaya komunikasi untuk menghubungkan dengan sesama pengguna atau user.
5. Melalui sistem client server, satu data dapat diakses oleh banyak client sekaligus dalam waktu yang sama. Dengan begitu, konsistensi data pun akan terjaga dan cenderung meningkat.
6. Hanya server yang membutuhkan ruang penyimpanan besar dengan performa yang tinggi. Perangkat penyimpanan yang memadai ini mampu menampung data dalam jumlah besar dan dapat diakses dalam waktu yang sangat singkat.

V. Pembuatan Aplikasi Client-Server Sederhana

Beberapa hal yang harus diperhatikan dalam pembuatan aplikasi Client-Server Sederhana (Single Thread) menggunakan Python antara lain:

- Membuat objek socket pada Server dan Client
- Mengikat objek socket pada Server ke alamat tertentu

- Mengikat objek socket pada Client ke alamat Server
- Menerima koneksi dari Client pada Server
- Mengirim dan menerima data antara Server dan Client menggunakan socket
- Menutup koneksi antara Server dan Client

Untuk membuat implementasi aplikasi Client-Server Sederhana (Single Thread) pada Python, dapat menggunakan modul socket dan threading yang sudah disediakan oleh Python.

Server :

```
import socket

server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

HOST = 'localhost'
PORT = 12345

server_socket.bind((HOST, PORT))

server_socket.listen(1)
print("Waiting...")
client_socket, client_address = server_socket.accept()

data = client_socket.recv(1024)

angka = int(data.decode())
print("Request dari client :", angka, "IP client :", client_address)

if angka % 2 == 0:
    response = "angka " + str(angka) + " merupakan genap"
else:
    response = "angka " + str(angka) + " merupakan ganjil"

client_socket.sendall(response.encode())

client_socket.close()
server_socket.close()
```

Client :

```
import socket

client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

HOST = 'localhost'
PORT = 12345
client_socket.connect((HOST, PORT))
```

```
pesan = input("Masukkan angka :")
client_socket.sendall(pesan.encode())

data = client_socket.recv(1024)
print(data.decode())

client_socket.close()
```

Pada contoh di atas, server akan menunggu koneksi dari client dan membaca data yang dikirimkan oleh client. Kemudian, server akan mengirim balasan ke client dan menutup koneksi. Client akan mengirimkan pesan ke server dan menunggu balasan dari server. Setelah menerima balasan dari server, koneksi akan ditutup.

Program di atas merupakan contoh sederhana dari aplikasi client-server dengan model single-threaded, di mana server hanya mampu melayani satu koneksi pada suatu waktu. Jika terdapat lebih dari satu koneksi yang masuk, maka koneksi tersebut akan diantre dan dilayani secara berurutan.

VI. Tugas dan Latihan

1. Membuat laporan percobaan praktikum dan beri Analisa Hasil Percobaan tadi yang sudah dibuat Pembuatan Aplikasi Client-Server Sederhana (Single Thread)
2. Membuat sebuah program server yang dapat menerima koneksi dari klien menggunakan protokol TCP. Server ini akan menerima pesan dari klien dan mengirimkan pesan balasan berisi jumlah karakter pada pesan tersebut. Gunakan port 12345 untuk server. Membuat analisa dari hasil program tersebut
3. Membuat sebuah program klien yang dapat terhubung ke server yang telah dibuat pada soal nomor 1. Klien ini akan mengirimkan pesan ke server berupa inputan dari pengguna dan menampilkan pesan balasan jumlah karakter yang diterima dari server. Membuat analisa dari hasil program tersebut