

Laporan Cloud Computing Tugas 4

Ricardo Supriyanto / 5025221218

Link Repository: <https://github.com/Ricardo08S/Task-CloudComputing/tree/main/Task-4>

I. Pendahuluan

Tugas ini bertujuan untuk memperdalam pemahaman peserta dalam mengelola layanan berbasis Docker Compose. Peserta diminta menjalankan empat kasus dari repositori <https://github.com/rm77/cloud2023/tree/master/containers/compose/images>, mendokumentasikan langkah-langkahnya, dan mengembangkan satu kasus tambahan (Case 5) berdasarkan kreativitas. Laporan harus mencakup diagram arsitektur, skrip pendukung, dan tangkapan layar hasil implementasi dengan penjelasan yang detail.

II. Deskripsi Tugas

Tugas ini terdiri dari empat kasus yang harus dijalankan menggunakan Docker. Setiap kasus melibatkan konfigurasi dan pengelolaan container untuk berbagai layanan, termasuk pembuatan Docker Image, serta langkah-langkah untuk menjalankan container dan mendokumentasikan proses implementasinya. Berikut adalah deskripsi setiap kasus:

Case 1: Membuat Docker Image dan menjalankan web server Nginx yang meng-host layanan aplikasi PHP.

Case 2: Membuat Docker Image dan menjalankan aplikasi Linux sederhana pada browser menggunakan VNC.

Case 3: Membuat Docker Image dan menjalankan web server Nginx menggunakan protokol HTTP.

Case 4: Membuat Docker Image dan menjalankan web server Nginx dengan layanan aplikasi PHP yang ditampilkan.

Case 5 (Case Tambahan): Membuat Docker Image dan menjalankan aplikasi konverter mata uang menggunakan Node.js dan Express

III. Arsitektur dan Implementasi Script

Case 1: Menjalankan Layanan Aplikasi PHP dengan Docker

1. Deskripsi:

Menggunakan Dockerfile untuk membuat image **mywebserver:1.0**. Aplikasi PHP akan dijalankan, menampilkan aplikasi sederhana dengan detail dependensi yang digunakan serta tulisan "Hello World".

2. Langkah-langkah:

Jalankan build.sh di direktori /platform untuk membangun Docker image, kemudian jalankan run-mywebserver.sh untuk mengonfigurasi dan menjalankan container.

3. **Penjelasan:**

Aplikasi PHP dapat diakses melalui <http://localhost:9999>, yang menampilkan detail dependensi dan pesan "Hello World".

Case 2: Menjalankan Aplikasi Linux pada Browser Menggunakan VNC

1. **Deskripsi:**

Dockerfile digunakan untuk membuat image **mylinux**. Aplikasi Linux sederhana akan dijalankan dalam browser dengan menggunakan VNC.

2. **Langkah-langkah:**

Jalankan build.sh di direktori /platform untuk membangun Docker image, lalu jalankan run-mylinux.sh untuk mengonfigurasi dan menjalankan container.

3. **Penjelasan:**

Aplikasi Linux dapat diakses melalui <http://localhost:11111>, di mana pengguna akan melihat tampilan desktop Linux sederhana yang memungkinkan akses ke terminal dan browser Firefox.

Case 3: Menjalankan Web Server Nginx dengan Protokol HTTP

1. **Deskripsi:**

Menggunakan Dockerfile untuk membuat image **mywebserver:2.0**, yang menampilkan template-bootstrap yang dapat diakses menggunakan protokol HTTP.

2. **Langkah-langkah:**

Jalankan build.sh untuk membangun Docker image, lalu jalankan run-server.sh untuk mengonfigurasi dan menjalankan container.

3. **Penjelasan:**

Aplikasi dapat diakses di <http://localhost:9999>, menampilkan berbagai template bootstrap yang tersedia untuk diakses melalui HTTP.

Case 4: Menjalankan Web Server Nginx dan Menampilkan Layanan Aplikasi PHP

1. **Deskripsi:**

Menggunakan Dockerfile untuk membuat image **mywebserver:2.1** dan menampilkan berbagai template bootstrap yang dapat diakses menggunakan protokol HTTP, serta aplikasi PHP yang dapat diakses di path tertentu.

2. **Langkah-langkah:**

Jalankan build.sh untuk membangun Docker image, lalu jalankan run-server.sh untuk mengonfigurasi dan menjalankan container.

3. **Penjelasan:**

Aplikasi dapat diakses di <http://localhost:9999>, menampilkan template bootstrap yang dapat diakses melalui HTTP, dan aplikasi PHP yang ditampilkan di <http://localhost:9999/test.php> dengan detail dependensi yang digunakan.

Case 5: Menjalankan Web Server Node.js untuk Konversi Mata Uang (Custom Case)

1. **Deskripsi:**

Menggunakan Dockerfile untuk membuat image **currency-converter-app:1.0**, yang menjalankan aplikasi konverter mata uang menggunakan Node.js dan Express.

Aplikasi ini memungkinkan pengguna untuk konversi antara mata uang USD dan IDR.

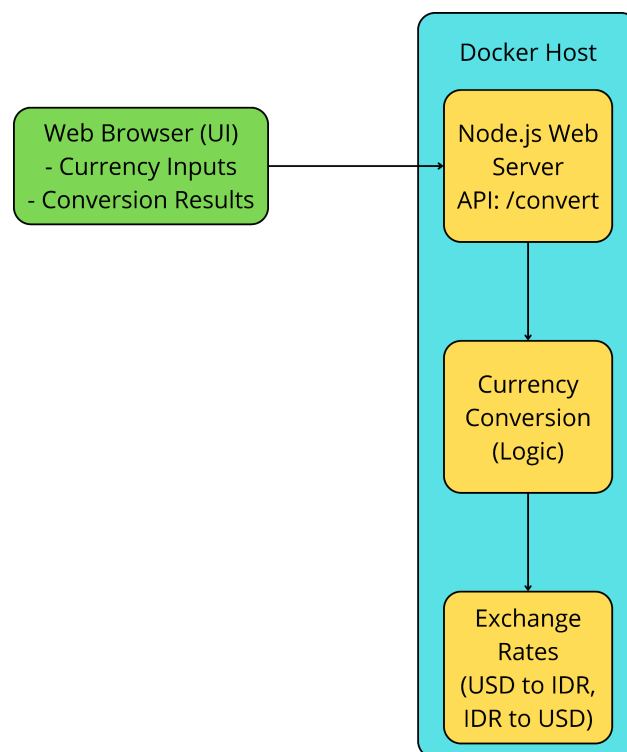
2. **Langkah-langkah:**

Jalankan build.sh di direktori /platform untuk membangun Docker image, lalu jalankan run.sh untuk mengonfigurasi dan menjalankan container. Aplikasi akan berjalan pada port 3000.

3. **Penjelasan:**

Aplikasi dapat diakses di **http://localhost:3000**, yang menyediakan antarmuka pengguna untuk konversi nilai antara USD dan IDR. API backend akan mengembalikan hasil konversi berdasarkan nilai tukar yang telah ditentukan.

Arsitektur **Case 5**:



1. **Web Browser (UI):**

- Pengguna mengakses antarmuka web melalui browser untuk memasukkan jumlah uang yang ingin dikonversi dan memilih mata uang asal serta tujuan.
- Halaman HTML ini diakses melalui port 3000 dari aplikasi yang berjalan dalam Docker.
- Mengirimkan permintaan HTTP GET ke API /convert di server Node.js.

2. **Node.js Web Server:**

- Server Node.js ini berjalan di dalam Docker dan menangani permintaan pengguna yang datang ke endpoint /convert.
- Menggunakan **Express.js** untuk menangani request dan response, serta untuk menyajikan file statis (HTML, CSS) di browser.
- Menerima parameter amount, from, dan to dari pengguna untuk menghitung konversi mata uang.

3. **Currency Conversion Logic:**

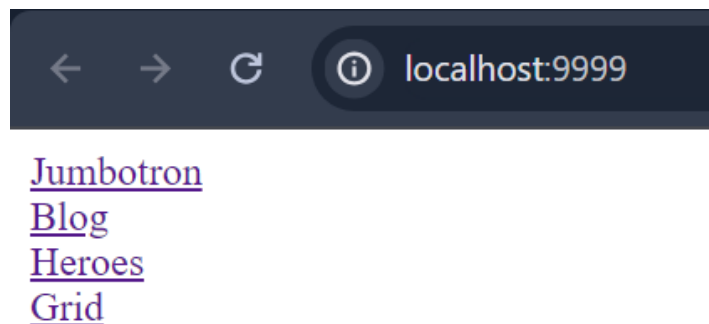
- Bagian ini menghitung konversi mata uang berdasarkan nilai tukar yang telah ditetapkan (misalnya 1 USD = 15,000 IDR).
 - Logika konversi menghitung nilai yang dikonversi dan mengembalikannya dalam format yang sesuai dengan mata uang tujuan.
4. **Exchange Rates:**
- Nilai tukar antara mata uang yang disediakan di-hardcode dalam aplikasi (misalnya, USD ke IDR dan IDR ke USD).
 - Aplikasi menggunakan nilai tukar ini untuk melakukan perhitungan konversi.
5. **Docker Container:**
- Seluruh aplikasi, mulai dari web server, API konversi, hingga logika perhitungan, dijalankan dalam Docker container.
 - Dengan menggunakan Docker, aplikasi ini dapat dijalankan di berbagai sistem tanpa ketergantungan atau konfigurasi tambahan, memastikan konsistensi lingkungan pengembangan dan deployment yang mudah.

Kapan Skenario Ini Cocok Digunakan?

Aplikasi ini cocok digunakan dalam skenario di mana tim pengembang atau individu ingin membangun sistem konversi mata uang yang sederhana dengan pencatatan aktivitas pengguna. Misalnya, dalam aplikasi e-commerce yang memerlukan konversi harga produk antar mata uang atau dalam aplikasi keuangan yang memungkinkan pengguna untuk memantau perubahan nilai tukar dan aktivitas konversi yang telah dilakukan. Dengan fitur logging yang terintegrasi, sistem ini dapat digunakan untuk pelacakan aktivitas dan analisis di masa mendatang.

IV. Screenshot Implementasi dan Penjelasannya

Case 1: Membuat Image dan Menjalankan Layanan Aplikasi PHP



Tampilan browser yang menunjukkan isi halaman web yang diakses melalui `http://localhost:9999`, menampilkan aplikasi PHP dengan pesan "Hello World" dan dependensinya.

Penjelasan:

Gambar pertama menunjukkan terminal yang sedang menjalankan perintah untuk membangun dan menjalankan Docker container. Gambar kedua menunjukkan aplikasi PHP

yang berjalan pada browser, menampilkan pesan "Hello World" dan dependensi yang digunakan.

Case 2: Membuat Image dan Menjalankan Aplikasi Ubuntu Sederhana

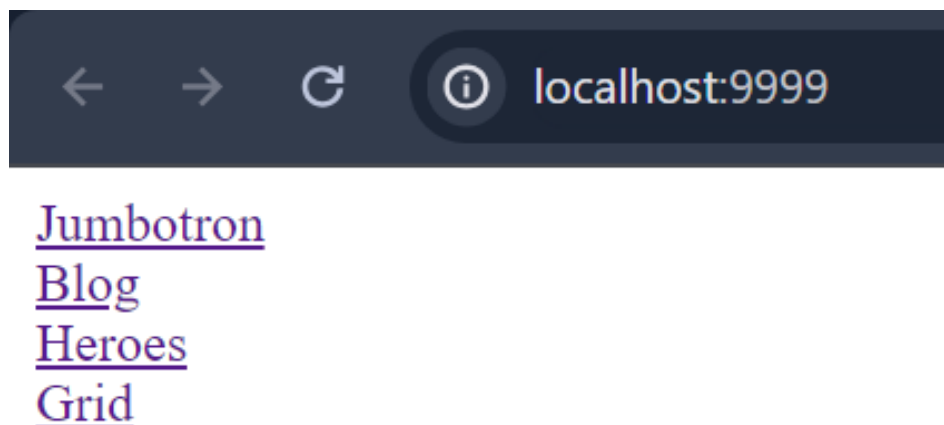


Tampilan browser yang menunjukkan tampilan desktop Linux yang diakses melalui <http://localhost:11111>.

Penjelasan:

Gambar pertama menunjukkan terminal yang sedang menjalankan perintah untuk membangun dan menjalankan container yang menjalankan aplikasi Linux di VNC. Gambar kedua menunjukkan desktop Linux yang berjalan di browser pada <http://localhost:11111>.

Case 3: Membuat Image dan Menjalankan Web Server Nginx

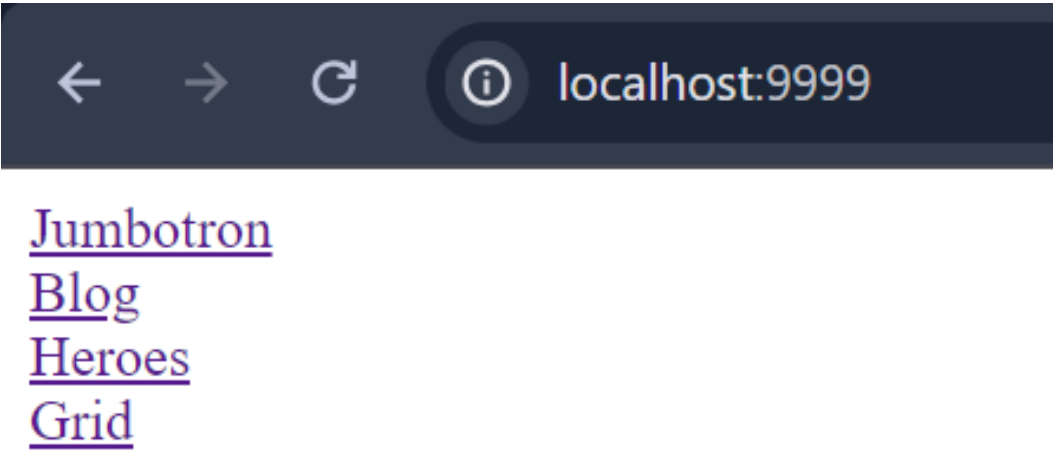


Tampilan browser yang menunjukkan isi halaman dari index.html yang diakses melalui <http://localhost:9999>, menampilkan template bootstrap yang dilayani oleh web server Nginx.

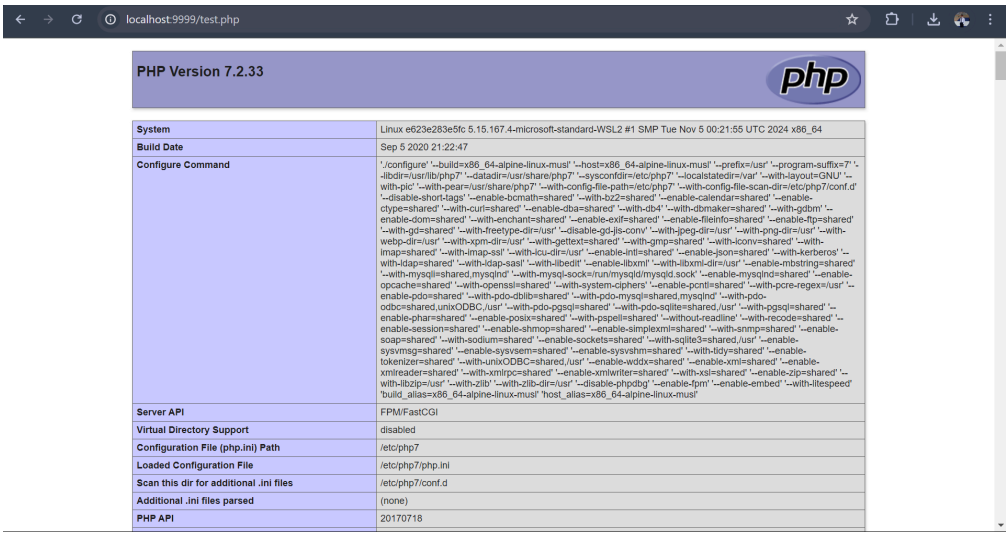
Penjelasan:

Gambar pertama menunjukkan terminal yang sedang menjalankan perintah untuk membangun dan menjalankan Docker container yang meng-host web server Nginx. Gambar kedua menunjukkan halaman web yang menampilkan berbagai template bootstrap yang diakses melalui HTTP pada <http://localhost:9999>.

Case 4: Membuat Image dan Menjalankan Web Server Nginx + Layanan Aplikasi PHP



Tampilan browser yang menunjukkan isi halaman web yang diakses melalui <http://localhost:9999>, menampilkan template bootstrap yang dilayani oleh web server Nginx.

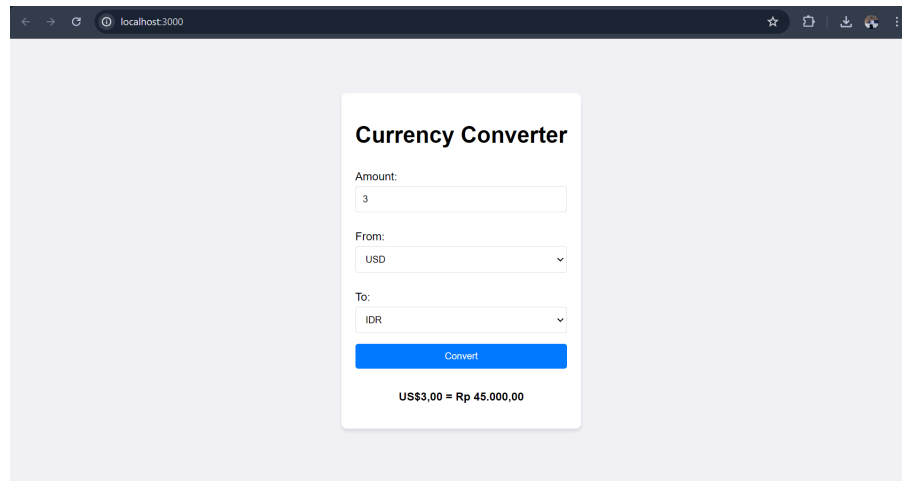


Tampilan browser yang menunjukkan aplikasi PHP dengan dependensinya yang diakses melalui <http://localhost:9999/test.php>.

Penjelasan:

Gambar pertama menunjukkan terminal yang menjalankan perintah untuk membangun dan menjalankan Docker container yang menjalankan Nginx dan aplikasi PHP. Gambar kedua menunjukkan tampilan web yang menampilkan template bootstrap yang diakses melalui Nginx. Gambar ketiga menunjukkan tampilan aplikasi PHP dengan dependensinya yang ditampilkan pada path <http://localhost:9999/test.php>.

Case 5: Membuat Image dan Menjalankan Web Server Node.js untuk Konversi Mata Uang



Tampilan browser yang menunjukkan tampilan aplikasi konverter mata uang yang diakses melalui <http://localhost:3000>, menampilkan form untuk konversi mata uang antara USD dan IDR.

Penjelasan:

Gambar pertama menunjukkan terminal yang sedang menjalankan perintah untuk membangun dan menjalankan Docker container yang menjalankan aplikasi Node.js untuk konversi mata uang. Gambar kedua menunjukkan aplikasi web yang diakses pada <http://localhost:3000>, dimana pengguna dapat memilih jumlah uang dan konversi mata uang antara USD dan IDR.

V. Kesimpulan

Pada laporan ini, empat case dari repository yang diberikan berhasil dijalankan, sementara Case 5 dikembangkan untuk membuat dan menjalankan Docker image secara mandiri dengan fitur konversi mata uang antara USD dan IDR. Pengembangan ini mencakup penjelasan terkait skrip yang digunakan, gambar arsitektur sederhana, serta implementasi antarmuka pengguna untuk melakukan konversi mata uang. Docker mempermudah pengelolaan aplikasi dengan menciptakan lingkungan yang konsisten di berbagai sistem, serta memastikan proses pengembangan dan penerapan aplikasi berjalan secara efisien di berbagai skenario yang melibatkan layanan web dan pengolahan data.