

MODUL 7

Docker Compose

7.1. Topik Pembahasan

- a. Docker Compose
- b. Proses Dasar Penggunaan Docker Compose
- c. Docker Compose Command
- d. Kelebihan Docker Compose
- e. Environment Variable

7.2. Tujuan Praktikum

- a. Praktikan Dapat memahami dan mengenali konsep dasar Docker Compose
- b. Praktikan memahami dan dapat membuat Docker Compose dasar
- c. Praktikan dapat menggunakan *command-command* docker compose
- d. Praktikan memahami fungsi dan cara kerja dari environment variable
- e. Praktikan dapat menggunakan environment variable dalam sebuah docker compose

7.3. Alat Dan Bahan

- a. Laptop

7.4. Dasar Teori

7.4.1. Docker Compose

Docker-Compose adalah alat untuk mendefinisikan dan menjalankan satu atau beberapa *container* yang saling terkait dengan sebuah command. Pada implementasinya kita dapat menggunakannya dengan membuat sebuah file berekstensi `yml/yml` yang di dalamnya terdapat konfigurasi-konfigurasi terhadap service aplikasi yang akan dijalankan. Gambaran sederhananya seperti menyatukan semua Dockerfile dari setiap *service* aplikasi ke dalam sebuah file `yml` (`docker-compose file`), dan selanjutnya dengan sebuah command, kita bisa meng-create dan men-start semua service yang telah didefinisikan pada file `yml` tersebut. script konfigurasi nya juga sangat mudah dan mirip dengan argumen-argumen saat kita ingin menjalankan sebuah

Dockerfile. Kita bisa menggunakannya di semua lingkungan kerja : *production*, *staging*, *development*, dan *testing*.

```
services:
  web:
    build: .
    ports:
      - "8000:5000"
    volumes:
      - ./code
      - logvolume01:/var/log
    depends_on:
      - redis
  redis:
    image: redis
volumes:
  logvolume01: {}
```

Gambar 8. 1 Contoh Docker Compose YAML

7.4.2. Proses Dasar Penggunaan Docker-Compose

Secara umum terdapat proses atau Langkah – Langkah dalam membuat sebuah Docker-Compose seperti berikut:

1. Mendefinisikan environment aplikasi dengan membuat Dockerfile sehingga bisa digunakan kembali dimana saja.
2. Mendefinisikan service lainnya (termasuk aplikasi kita) di dalam docker-compose.yml sehingga semuanya bisa berjalan bersamaan dalam environment yang terisolasi.
3. Jalankan command docker compose yaitu : docker-compose build dan docker-compose up melalui command prompt/shell (pastikan anda berada pada path yang sama dengan posisi file docker-compose.yml).

7.4.3. Docker Compose Command

Docker compose Command, adalah command (perintah) yang bisa dijalankan melalui docker-compose untuk mengelola keseluruhan service yang telah didefinisikan. Berikut adalah command docker compose yang biasa digunakan : format : docker-compose [command]

1. **docker-compose build** : digunakan untuk mem-build semua service yang telah didefinisikan pada docker-compose.yml, output dari command ini adalah image per service yang siap untuk dijalankan pada sebuah

container. Default nama image yang dihasilkan adalah [namafolder-posisi-file_docker-compose.yml]_[nama_service] misal jika file docker-compose.yml yang memiliki service: web berada di dalam folder project, maka nama image yang dihasilkan adalah project_web

2. **docker-compose up** : digunakan untuk menjalankan semua image yang telah di-build (docker-compose build), jika tidak terjadi masalah maka image tersebut menjadi running container sedangkan jika terdapat masalah maka biasanya akan terbentuk container yang posisinya berhenti/stop. Default nama container jika tidak didefinisikan adalah sama dengan nama imagenya.³
3. **docker-compose down** : Menghentikan (stop) semua container dan meremovanya (ingat stop tidak otomatis remove) semua container, networks, volumes, dan images yang dihasilkan oleh perintah up
4. **docker-compose logs** : command ini penting jika kita ingin melihat apa yg terjadi pada service tertentu saat dijalankan dalam container. Biasa digunakan untuk mendeteksi error atau melihat kondisi/respon yang terjadi pada service tertentu.
5. **docker-compose config** : digunakan untuk mengecek apakah script yang ada pada file docker-compose.yml yang telah kita buat sudah valid atau sudah sesuai dengan referensi dari docker-compose. Output dari command ini akan memberikan pesan error (jika error) atau jika berhasil hanya menampilkan print out semua script yg ada di docker-compose.yml.

7.4.4. Kelebihan Docker Compose

Docker Compose memiliki sejumlah kelebihan yang menjadikannya alat yang populer dalam dunia software development. Berikut adalah beberapa kelebihan utama Docker Compose.

1. Sintaks yang tersedia mudah dipahami dan digunakan.
2. Antarmuka yang intuitif dan mudah dipahami, memungkinkan pengguna memulai dan mengelola aplikasi yang terdiri dari beberapa kontainer secara cepat.
3. Adanya pengaturan lintas platform sehingga compatible dipakai pada berbagai platform seperti Linux, macOS, dan Windows, dan lainnya.
4. Pengguna bisa membuat replikasi kontainer dengan mudah, sehingga memungkinkan aplikasi untuk berkembang dan menangani beban kerja yang lebih besar.

5. Terintegrasi dengan Docker sehingga memberikan keuntungan penuh dari teknologi kontainerisasi. Hal tersebut termasuk isolasi, portabilitas, dan kemampuan untuk membagikan aplikasi dengan orang lain secara mudah.

7.4.5. Environment Variable

Environment Variable adalah tempat untuk menampung suatu nilai yang bersifat dinamis (bisa berubah) yang nantinya bisa mempermudah kita untuk menjalankan suatu perintah pada script.

Contoh, pada bash biasanya terdapat variable \$USER yang bernilai username saat ini. Dengan demikian kita bisa mudah merubah hak kepemilikan suatu file tanpa harus menulis username yang mungkin lebih panjang.

Penggunaan Environment Variable pun digunakan untuk meningkatkan keamanan suatu program, karena dengan menyimpan suatu informasi penting dalam Environment Variable informasi penting tidak akan secara langsung terekpose dalam suatu program