

Laporan Praktek Topik Khusus 6

Docker File



SEMESTER VI

**Muhammad Abel Al-fahrezi
2211083034**

**PROGRAM STUDI D4 TEKNOLOGI REKAYASA
PERANGKAT LUNAK
JURUSAN TEKNOLOGI INFORMASI**

A. Landasan Teori

1. Windows Subsystem for Linux (WSL) :

WSL adalah fitur dari sistem operasi Windows yang memungkinkan pengguna untuk menjalankan lingkungan GNU/Linux secara langsung di Windows, tanpa harus menggunakan mesin virtual (Virtual Machine) secara penuh. Versi terbaru, **WSL2**, menggunakan kernel Linux asli yang mendukung sistem file yang lebih cepat dan kompatibilitas lebih baik dengan tool berbasis Linux seperti Docker. Dengan WSL2, pengguna Windows dapat menjalankan container Docker **secara native**, tanpa harus bergantung pada teknologi virtualisasi berat seperti Hyper-V atau VirtualBox.

2. Docker dan Container :

Nginx Docker adalah platform open-source yang dirancang untuk mengembangkan, mengirimkan, dan menjalankan aplikasi dalam unit terisolasi yang disebut container. Container berisi semua dependensi aplikasi, seperti kode, pustaka, dan konfigurasi, sehingga menjamin konsistensi aplikasi saat dipindahkan antar lingkungan. Docker bekerja sangat baik di lingkungan Linux karena container dibangun di atas fitur kernel Linux seperti cgroups dan namespaces. Oleh karena itu, menjalankan Docker melalui WSL2 menciptakan kondisi ideal untuk penggunaan container di sistem operasi Windows.

3. Dockerfile

Dockerfile adalah sebuah file teks yang berfungsi sebagai skrip instruksi untuk membangun image Docker. File ini mendeskripsikan secara berurutan perintah-perintah yang harus dijalankan Docker untuk membentuk sebuah lingkungan aplikasi yang dapat dijalankan secara terisolasi dalam container. Instruksi-instruksi dalam Dockerfile mencakup pemilihan image dasar (FROM), penyalinan file dari sistem lokal ke dalam image (COPY atau ADD), pengaturan direktori kerja (WORKDIR), instalasi dependensi (RUN), deklarasi variabel lingkungan (ENV), hingga perintah yang akan dijalankan saat container aktif (CMD atau ENTRYPOINT). Dengan menggunakan Dockerfile, proses pembuatan dan

pengaturan lingkungan aplikasi menjadi lebih otomatis, konsisten, dan portabel, karena image yang dihasilkan dapat dijalankan di berbagai sistem tanpa perlu konfigurasi ulang. Hal ini sangat berguna dalam pengembangan perangkat lunak modern, terutama pada pendekatan DevOps dan microservices, di mana konsistensi dan kecepatan deployment menjadi sangat penting.

B. Tools

- Ubuntu (melalui WSL2)
- VSCode

C. Langkah Kerja

1. RUN Instruction

- Instruksi RUN digunakan untuk mengeksekusi perintah shell selama proses build image.
- Instruksi ini umumnya digunakan untuk menginstal paket atau melakukan modifikasi pada image selama proses build berlangsung.
 - *Docker run -d --name my-nginx:1.0*

Untuk mengeksekusi dockerfile

```
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/run$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
nginx        alpine    6769dc3a703c   6 weeks ago   48.2MB
nginx        latest    be69f2940aaf   6 weeks ago   192MB
<none>       <none>    739f5132fa89   6 weeks ago   48.2MB
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/run$ docker build -t my-nginx:1.0 -q .
sha256:4b2acb5add148232b6d7b6fa92d8cdc3fd994afda5cbdf995a0eaf9b53600f01
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/run$ docker run -d --name my-nginx my-nginx:1.0
6b5ac53c98040c89d3c10858b2957645376e1c665bc2ebb7a65cf8430475abca
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/run$
```

2. Label Instruction

- Instruksi LABEL digunakan untuk menambahkan metadata atau informasi tambahan ke dalam sebuah image.
- Label didefinisikan menggunakan format pasangan *key-value* (kunci-nilai).
- Label dapat berisi informasi seperti versi, pembuat (author), tujuan penggunaan, atau informasi khusus lainnya.
- melihat label dari sebuah image dengan perintah:
 - *Docker inspect my-nginx*

```
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/label$ docker container rm my-nginx
my-nginx
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/label$ docker build -t my-nginx:latest -q .
sha256:d11071b23075a1d3c43ba82034f5652a57dbb370980eb9be452a2d5ac729b2a1
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/label$ docker run -d --name my-nginx my-nginx:latest
ceda7dc94f4da1eb65e929aae188420bf8c6786f91947339bfe6d728dc50c166
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/label$ docker inspect my-nginx
[
  {
    "Id": "ceda7dc94f4da1eb65e929aae188420bf8c6786f91947339bfe6d728dc50c166",
    "Created": "2025-05-30T14:55:07.6387186Z",
    "Path": "/docker-entrypoint.sh",
    "Args": [
      "nginx",
      "-g",
      "daemon off;"
    ],
    "State": {
```

3. Add Instruction

- ADD digunakan untuk menyalin file, folder, atau URL dari jarak jauh ke dalam image Docker.
- Sintaksnya adalah: *ADD <src> <dest>*
- Jika file sumber berupa arsip terkompresi (seperti .tar, .gz), Docker akan mengekstraknya secara otomatis.

```

rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/add$ docker
r build -t my-nginx:latest -q .
sha256:f584c66c5794eb1102f96fa0cb34d96832325d9efa9520195990a2799dc8dc19
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/add$ docker
r run -d --name my-nginx my-nginx:latest
e3841a0740e60a1d7b7a34bc22ef0207f95939ac8be808188fcaa6180dfa6c38
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/add$ docker
r exec -it my-nginx /bin/sh
/ # cd /usr/share/nginx/html
/usr/share/nginx/html # ls -l
total 12
-rw-r--r-- 1 root root 497 Apr 16 12:55 50x.html
-rwxrwxrwx 1 root root 125 May 30 14:25 about.html
-rw-r--r-- 1 root root 615 Apr 16 12:55 index.html
/usr/share/nginx/html # ls
50x.html about.html index.html
/usr/share/nginx/html #

```

4. COPY Instruction

- COPY memiliki fungsi yang sama dengan ADD, yaitu menyalin file atau folder ke dalam image.
- Namun, berbeda dengan ADD, COPY tidak mendukung ekstraksi otomatis file arsip atau pengunduhan dari URL.
- COPY lebih direkomendasikan dibandingkan ADD karena lebih sederhana dan cenderung menghasilkan lebih sedikit kesalahan.

```

rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker$ cd copy
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/copy$ dock
er stop my-nginx
my-nginx
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/copy$ dock
er container rm my-nginx
my-nginx
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/copy$ dock
er build -t my-nginx:latest -q .
sha256:ce5fcf01a40cda6594532751a7f6bece5aa761a29f65df5c1927198ffcd54647
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/copy$ dock
er exec -it my-nginx /bin/sh
Error response from daemon: No such container: my-nginx
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/copy$ dock
er run -d --name my-nginx my-nginx:latest
6ce3e5198e248e9d59b52e67d6a3a7ec99bce5de29c8760bc8906e4c44437b7e
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/copy$ dock
er exec -it my-nginx /bin/sh
/ # cd /usr/share/nginx/html
/usr/share/nginx/html # ls
50x.html about.html contactus.html index.html
/usr/share/nginx/html #

```

5. .dockerignore File

- dockerignore digunakan untuk mengecualikan file atau folder tertentu saat menggunakan perintah COPY atau ADD dalam Dockerfile.
- Fungsinya mirip dengan .gitignore: memberi tahu Docker file atau folder apa saja yang harus diabaikan saat membangun image.
- Hal ini membantu untuk:
 - Mengurangi ukuran image
 - Menghindari penyalinan file yang tidak diperlukan atau bersifat sensitif

```
command.sh label Dockerfile add Dockerfile dockerignore .dockerignore x command.sh add
dockerignore > .dockerignore
1 | ignore.html

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/dockerignore$ docker exec -it my-n
ginx /bin/sh
/# cd /usr/share/nginx/html
/usr/share/nginx/html # ls -l
total 16
-rw-r--r-- 1 root root 497 Apr 16 12:55 50x.html
-rwxrwxrwx 1 root root 125 May 30 14:25 about.html
-rwxrwxrwx 1 root root 109 May 30 14:25 contactus.html
-rw-r--r-- 1 root root 615 Apr 16 12:55 index.html
/usr/share/nginx/html # ls
50x.html about.html contactus.html index.html
/usr/share/nginx/html #
```

6. EXPOSE Instrurction

- Instruksi EXPOSE digunakan untuk menunjukkan bahwa image akan menggunakan port tertentu saat dijalankan sebagai container.
- Kamu juga dapat menentukan protokol yang digunakan (TCP/UDP). Jika tidak disebutkan, Docker secara default akan menganggapnya sebagai TCP.
- Untuk memeriksa port yang diekspos dalam sebuah image, kamu dapat menggunakan perintah:

○ *Docker inspect my-nginx*

```
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker$ cd expose
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/expose$ docker stop my-nginx
ker container rm my-nginxmy-nginx
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/expose$ docker container rm my-nginx
my-nginx
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/expose$ docker build -t my-nginx:late
atest -q .
sha256:fc55dba35bFF9Fcf3ab3e005c2e01ae44da5d47108a5b5e02f237c948a6a102
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/expose$ docker run -d --name my-ngi
inx my-nginx:latest
c0a9cf49129d7ae14af9e9b32eecd0004e5c287afa5c5cb8802d0b6ddf0e7f2d
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/expose$ docker inspect my-nginx
[
  {
    "Id": "c0a9cf49129d7ae14af9e9b32eecd0004e5c287afa5c5cb8802d0b6ddf0e7f2d",
    "Created": "2025-05-30T15:42:09.6309757Z",
    "Path": "/docker-entrypoint.sh",
    "Args": [
      "nginx",
      "-g",
      "daemon off;"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 2545,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2025-05-30T15:42:10.0889974Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    },
    "Image": "sha256:fc55dba35bFF9Fcf3ab3e005c2e01ae44da5d47108a5b5e02f237c948a6a102",
```

7. ENV Instruction

- Instruksi ENV digunakan untuk mendefinisikan variabel lingkungan (environment variables) yang akan tersedia saat container dijalankan.
- Variabel lingkungan digunakan untuk membuat konfigurasi aplikasi menjadi lebih dinamis.
- Kamu dapat menggunakan kembali variabel lingkungan dengan format: \$(NAMA_ENV)

```

rzigms@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker$ cd env
rzigms@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/env$ docker
r stop my-nginx
ker container rm my-nginxmy-nginx
rzigms@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/env$ docker
r container rm my-nginx
my-nginx
rzigms@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/env$ docker
r build -t my-nginx:latest -q .
sha256:2c549ec5ba27018e31603d0f1ff9c3c3d52adc4fb04e5f8681ce98a1e97f875e
rzigms@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/env$ docker
r run -d --name my-nginx my-nginx:latest
a9b2637c5018c162bc596115e70bef03061d6aba33e17b0896fbc6e61acaad08
rzigms@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/env$ docker
r exec -it my-nginx /bin/sh
/ # env
HOSTNAME=a9b2637c5018
DB_PORT=5432
SHLVL=1
HOME=/root
PKG_RELEASE=1
DYNPKG_RELEASE=1
TERM=xterm
NGINX_VERSION=1.27.5
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
NJS_VERSION=0.8.10
NJS_RELEASE=1
DB_PASSWORD=secret
PWD=/
DB_HOST=localhost
DB_USER=admin
/ #

```

8. ARG Instruction

- ARG bekerja mirip dengan ENV, tetapi hanya tersedia selama proses build image.
- Gunakan ARG saat kamu perlu mengatur nilai statis yang tidak berubah saat container dijalankan (runtime).

```

rzigms@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker$ docker ru
n -d --name my-nginx my-nginx:latest
7709d4a2016c138f99ea585ebcc50dddc96f828012eb3989b24c092713c66c2e
rzigms@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker$ docker ex
ec -it my-nginx /bin/sh
/ # more version.txt
1.0
/ #

```

9. WORKDIR Instruction

- WORKDIR (working directory) adalah folder kerja di dalam image.
- Jika folder tersebut belum ada, maka akan dibuat secara otomatis.
- WORKDIR dapat didefinisikan menggunakan path relatif maupun absolut.

```

rzigms@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/workdir$ d
ocker build -t my-nginx:latest -q .
sha256:626508407317e931b317a9480d8c8106522bf1261b5c9dcca3ed1cf4afe7aa2
rzigms@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/workdir$ d
ocker run -d --name my-nginx my-nginx:latest
2d4bcbdb8a91189a0cd7c54e92c0c8b78d5d74dd9ff3f7f0a39bc3e020324d919
rzigms@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/workdir$ d
ocker exec -it my-nginx /bin/sh
/app/public # ls
about.html
/app/public #

```

10. CMD Instruction

- CMD mendefinisikan perintah default yang akan dijalankan saat sebuah container dimulai.
- Hanya satu perintah CMD yang diperbolehkan dalam sebuah Dockerfile — perintah

terakhir akan menggantikan yang sebelumnya.

- CMD dapat didefinisikan dalam dua bentuk:
 - Bentuk Exec (direkomendasikan): CMD ["executable", "param1", "param2"]
 - Bentuk Shell: CMD command param1 param2

```
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker$ cd cmd
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/cmd$ docker
r stop my-nginx
ker container rm my-nginxmy-nginx
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/cmd$ docker
r container rm my-nginx
my-nginx
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/cmd$ docker
r build -t my-nginx:latest -q .
sha256:739f5132fa89fb815547ae21336e52c04e1035afb17f2f3bb08ac0a31b2ba852
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/cmd$ docker
r run --rm my-nginx:latest
Hello from Docker
rzigns@WIN-KL41ST0GR0L:/mnt/f/Abel/Topik Khusus/Latihan/6_DockerFile/belajar-docker/cmd$
```

D. Kesimpulan

Praktik pembuatan Dockerfile memberikan pemahaman yang mendalam tentang cara otomatisasi pembuatan image Docker yang konsisten dan portabel. Dengan menggunakan instruksi seperti FROM, COPY, RUN, WORKDIR, dan CMD, kita dapat membangun lingkungan aplikasi secara terstruktur dan dapat direproduksi di berbagai sistem tanpa perlu konfigurasi manual ulang. Proses build yang otomatis juga memudahkan instalasi dependensi dan pengaturan aplikasi secara efisien. Selain itu, pemahaman tentang instruksi tambahan seperti ENV, ARG, dan EXPOSE membantu mengelola konfigurasi dan komunikasi container dengan lebih baik. Dengan demikian, Dockerfile menjadi alat penting dalam pengembangan aplikasi modern, khususnya dalam praktik DevOps dan deployment berbasis container.