PRAKTIKUM

MANAJEMEN DATA

Dosen Pengampu : Isbat Uzzin Nadhori, S.Kom., M.T.

Disusun Oleh:

Zerlina Ardelia Risdafari Sajida

(3324600014)

Sains Data Terapan A

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER

PROGRAM STUDI SAINS DATA TERAPAN

TUGAS 1

- Buat Kemball tugas dari Quiz terakhir pembuatan script (soal 1 s.d 4)

1. Buat script untuk melakukan pengecekan service ssh secara berkala dengan interval waktu tiap 10 detik dan memberikan notifikasi ke layar jika service ssh mati. Buat dok file, Capture hasil program anda, dan pastikan script jalan dengan melakukan stop terhadap service ssh dan capture hasil script yang dijalankan. Kumpulkan di link yang disediakan.

```
vboxuser@desktop:~$ nano monssh.sh
```

```
 GNU nano 7.2                          monssh.sh
#!/bin/bash

# Nama file log hasil
LOG_FILE="hasil_ssh.log"

# Interval cek dalam detik
INTERVAL=10
 Help
echo "Mulai memonitor service ssh..." > $LOG_FILE

while true; do
    TIMESTAMP=$(date '+%Y-%m-%d %H:%M:%S')
    SERVICE_STATUS=$(systemctl is-active ssh)

    echo "$TIMESTAMP - Status SSH: $SERVICE_STATUS" >> $LOG_FILE

    if [ "$SERVICE_STATUS" != "active" ]; then
        echo "[$TIMESTAMP] ⚠️ SSH Service tidak aktif!" | tee -a $LOG_FILE
    fi

    sleep $INTERVAL
done
```

```
vboxuser@desktop:~$ chmod +x monssh.sh
vboxuser@desktop:~$ ./monssh.sh
[2025-06-08 07:51:13] ⚠️ SSH Service tidak aktif!
[2025-06-08 07:51:24] ⚠️ SSH Service tidak aktif!
[2025-06-08 07:51:34] ⚠️ SSH Service tidak aktif!
```

2. Buat script backup direktori tertentu dan jalankan backup secara berkala setiap 15 detik, dengan backup disimpan ke file1 sampai file10, lalu kembali lagi menimpa file1 jika sudah sampai file10. Jalankan backup tersebut selama 1 jam. Jalankan backup ini secara otomatis dengan crontab setiap hari Minggu jam 1 malam.

```
vboxuser@desktop:~$ mkdir -p ~/data
vboxuser@desktop:~$ echo "file percobaan" > ~/data/file1.txt
vboxuser@desktop:~$ mkdir -p ~/backup
vboxuser@desktop:~$ nano ~/backup/backup.sh
```

```
 GNU nano 7.2                  /home/vboxuser/backup/backup.sh *
#!/bin/bash

SOURCE_DIR="$HOME/data"
BACKUP_DIR="$HOME/backup"
MAX_FILES=10
INTERVAL=15  # detik
DURATION=$((60*60))  # 1 jam = 3600 detik
COUNT=0

end=$((SECONDS+DURATION))

while [ $SECONDS -lt $end ]; do
    FILE_INDEX=$(( (COUNT % MAX_FILES) + 1 ))
    TIMESTAMP=$(date +"%Y%m%d_%H%M%S")
    tar -czf "$BACKUP_DIR/backup$FILE_INDEX.tar.gz" -C "$SOURCE_DIR" .
    echo "Backup ke-$FILE_INDEX selesai pada $TIMESTAMP"
    COUNT=$((COUNT+1))
    sleep $INTERVAL
done
```

```
vboxuser@desktop:~$ chmod +x ~/backup/backup.sh
vboxuser@desktop:~$ ~/backup/backup.sh
Backup ke-1 selesai pada 20250608_075709
Backup ke-2 selesai pada 20250608_075724
Backup ke-3 selesai pada 20250608_075739
Backup ke-4 selesai pada 20250608_075754
```

3.

```
vboxuser@desktop:~$ nano data.txt
```

```
 GNU nano 7.2                                  data.txt *
timestamp          suhu  kelembaban  cahaya
2025-05-27T08:00   24.5     60         800
2025-05-27T08:15   25.0     62         850
2025-05-27T08:30   26.2     65         900
2025-05-27T08:45   27.1     67         950
```

```
vboxuser@desktop:~$ nano suhu_maks.awk
```

```
 GNU nano 7.2                                suhu_maks.awk *
BEGIN { max = -999 }
NR > 1 { if ($2 > max) max = $2 }
END { print "Suhu maksimum: " max }
```

```
vboxuser@desktop:~$ awk -f suhu_maks.awk data.txt
Suhu maksimum: 27.1
```

```
vboxuser@desktop:~$ nano rata_kelembaban.awk
```

```
 GNU nano 7.2                              rata_kelembaban.av
BEGIN { total = 0; count = 0 }
NR > 1 { total += $3; count++ }
END { print "Rata-rata kelembaban: " total / count }
```

```
vboxuser@desktop:~$ awk -f rata_kelembaban.awk data.txt
Rata-rata kelembaban: 63.5
```

4.

```
vboxuser@desktop:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
```

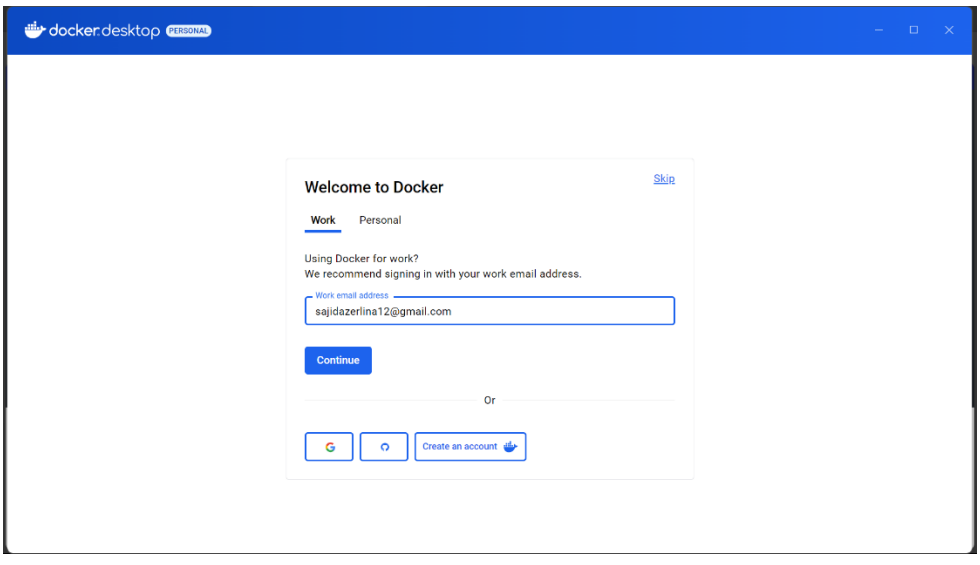
```
vboxuser@desktop:~$ grep "/home/" /etc/passwd
vboxuser:x:1000:1000:vboxuser:/home/vboxuser:/bin/bash
vboxuser@desktop:~$ grep "/home/" /etc/passwd | cut -d: -f1
vboxuser
```

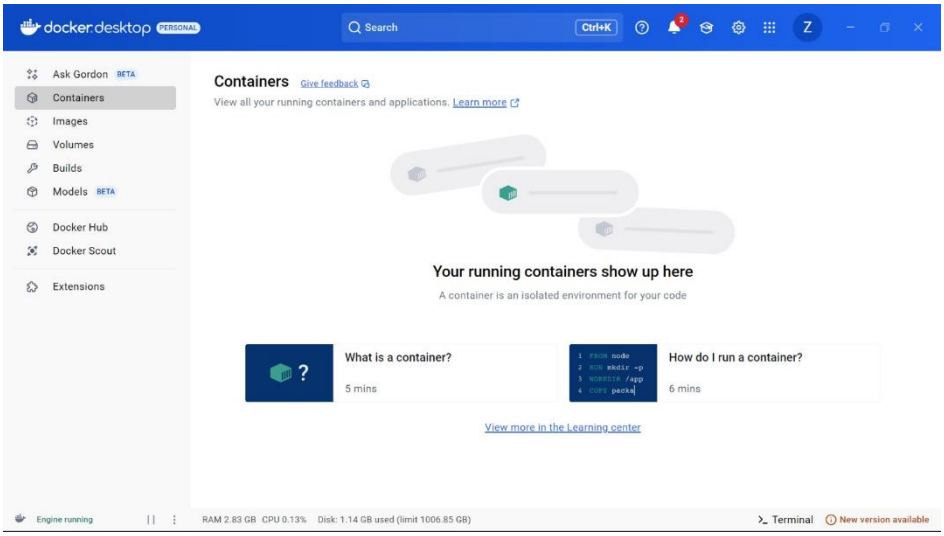
TUGAS 2

1. Download Docker

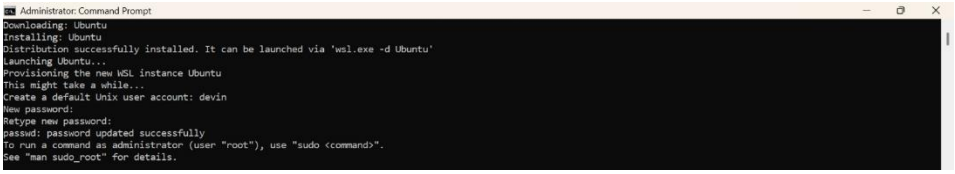
Restart

Log in





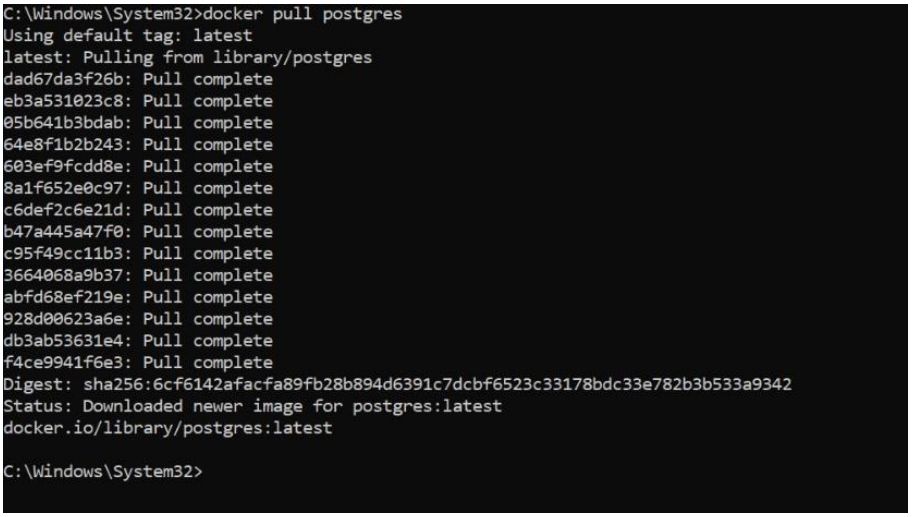
2. Install WSL



3. Docker Images
   - Bisa dibikin, bisa download images yang sudah ada

4. Download Images

- Jalankan perintah docker pull posgtres



- Cek : docker images

5. Download Image
- Pull

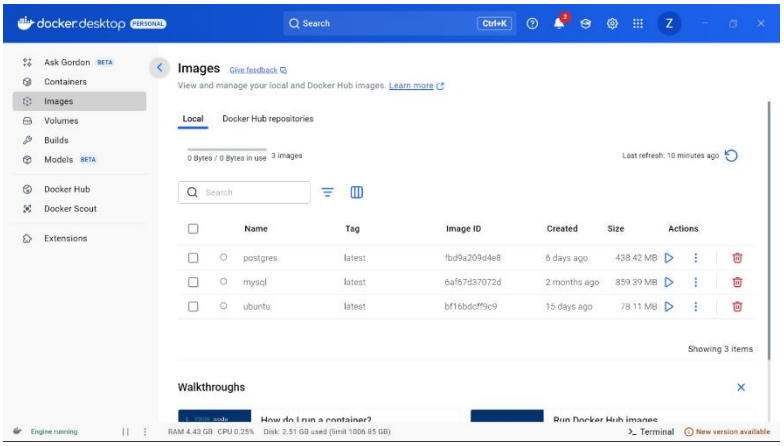
```
C:\Windows\System32>docker --version
Docker version 28.1.1, build 4eba377

C:\Windows\System32>docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
43c486e74c6d: Pull complete
86362e2a75e4: Pull complete
ba155de89fc7: Pull complete
1770ba94c103: Pull complete
d89ba43c350c: Pull complete
55f4efd3f008: Pull complete
e7b97b02f10e: Pull complete
2f44111e3dd4: Pull complete
545a2bff0604: Pull complete
958d7f5cf224: Pull complete
Digest: sha256:072f96c2f1ebb13f712fd88d0ef98f2ef9a52ad4163ae67b550ed6720b6d642e
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
```

```
C:\Windows\System32>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
d9d352c11bbd: Pull complete
Digest: sha256:b59d21599a2b151e23eea5f6602f4af4d7d31c4e236d22bf0b62b86d2e386b8f
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

6. List Image
- Docker image ls

```
C:\Windows\System32>docker image ls
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
postgres     latest    fbd9a209d4e8   6 days ago    438MB
ubuntu       latest    bf16bdcff9c9   2 weeks ago   78.1MB
mysql        latest    6af67d37072d   8 weeks ago   859MB
```

7. List Image di Docker Desktop



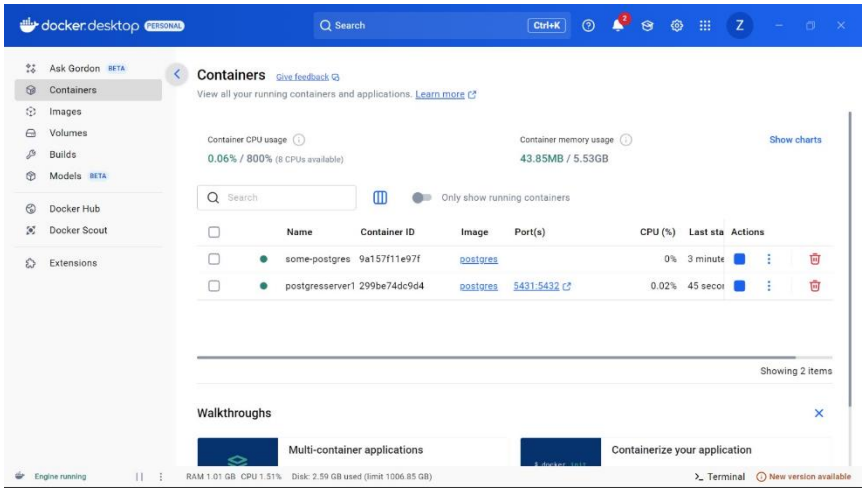
8. Container
- Docker image diistilahkan resep masakan
- Sedangkancontainer adalah makananjadi yang dibuatdari resep tsb
- Biasadisebut instance dari image

```
C:\Windows\System32>docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

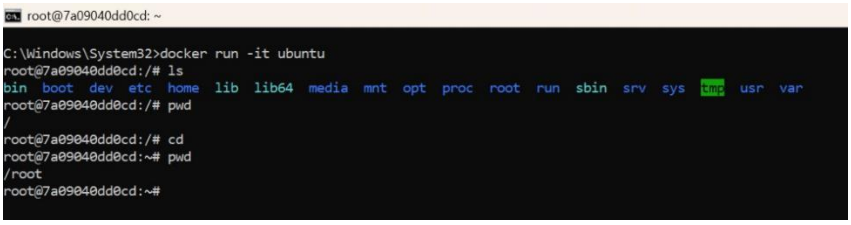
9. Membuatdan MenjalankanContainer

- Menjalankan dua container

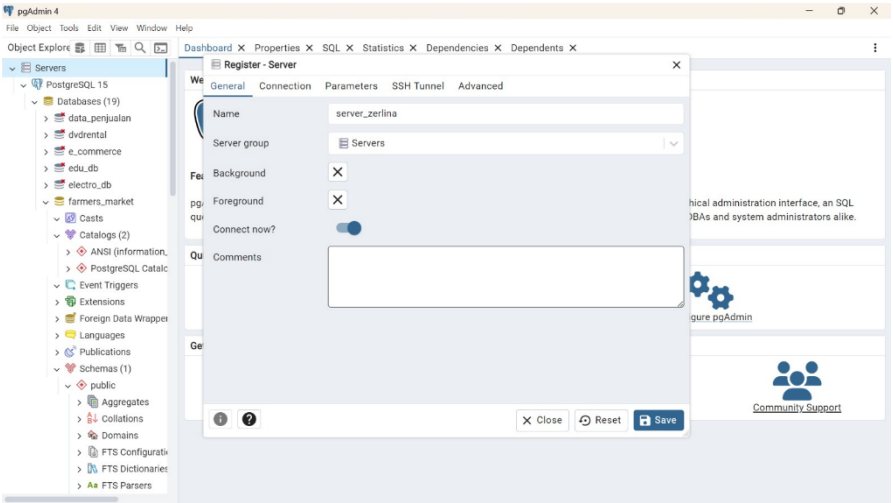
```
C:\Windows\System32>docker run --name some-postgres -e POSTGRES_PASSWORD=123456 -d postgres
9a157f11e97fa14d2ce16357ca0bdca563f1a13b9e18ea157c68aa898c0c86e9

C:\Windows\System32>docker run --name postgresserver1 -p 5431:5432 -e POSTGRES_PASSWORD=123456 -d postgres
299be74dc9d4e56567ec96655078bf7c27e6f8caf4b6ec975a05a78d97c99251
```

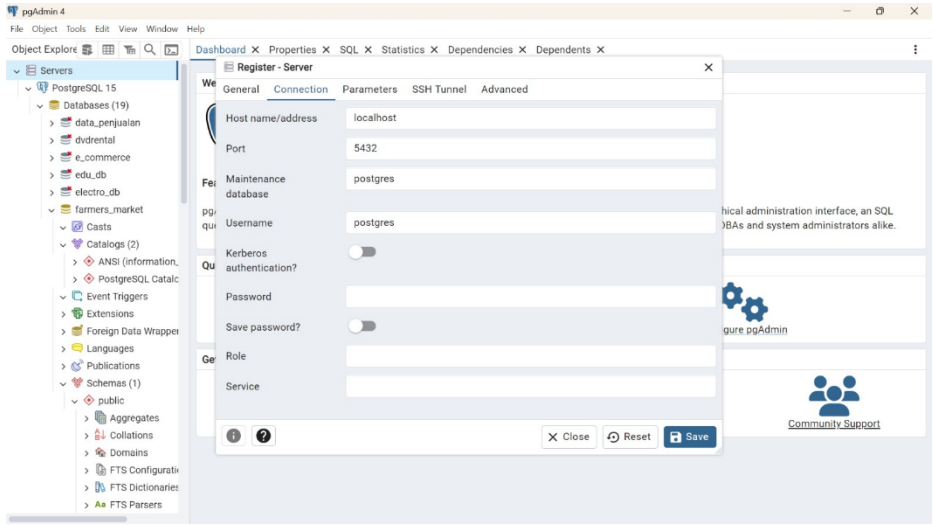


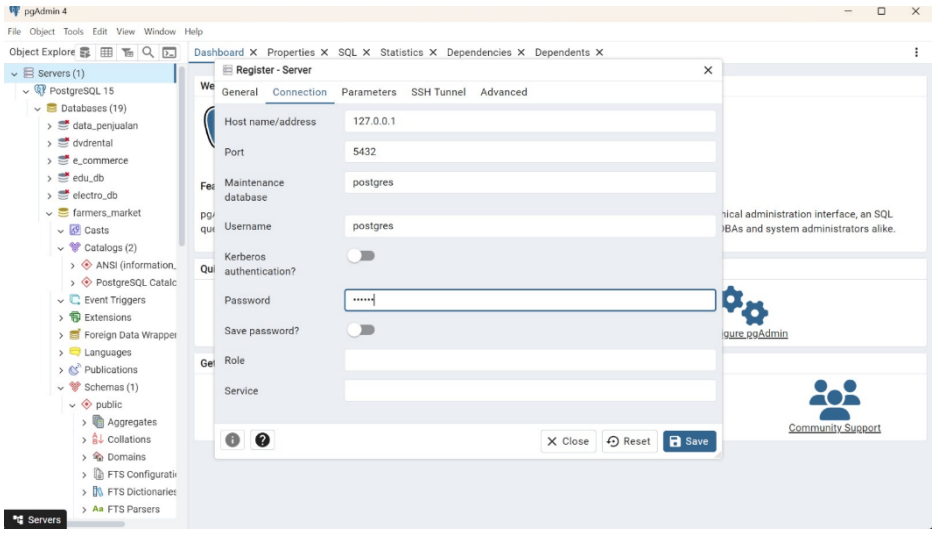
10. Jalankan Image



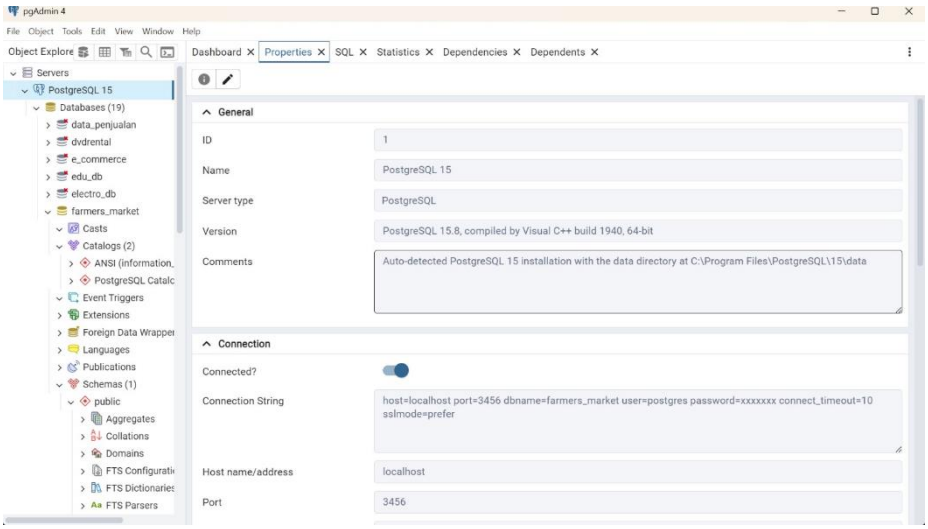
11. Koneksi ke pgadmin
- Buka pgadmin
- Buka Server → dashboard →add new server



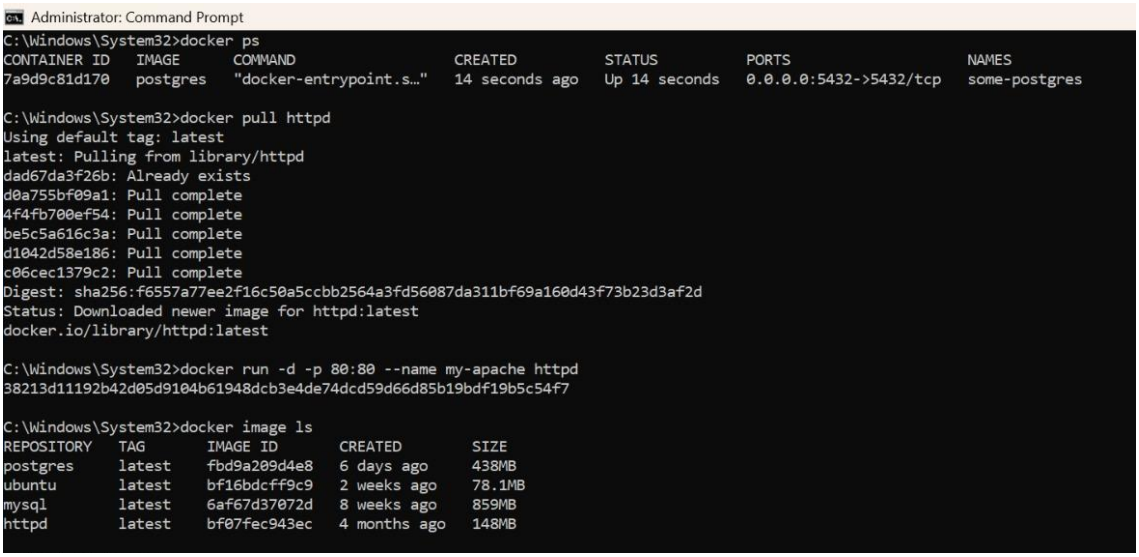


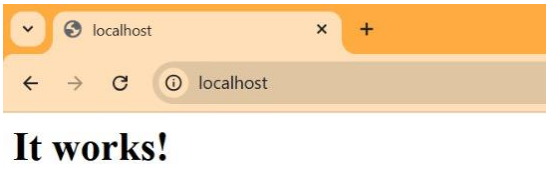


- Perbandingan

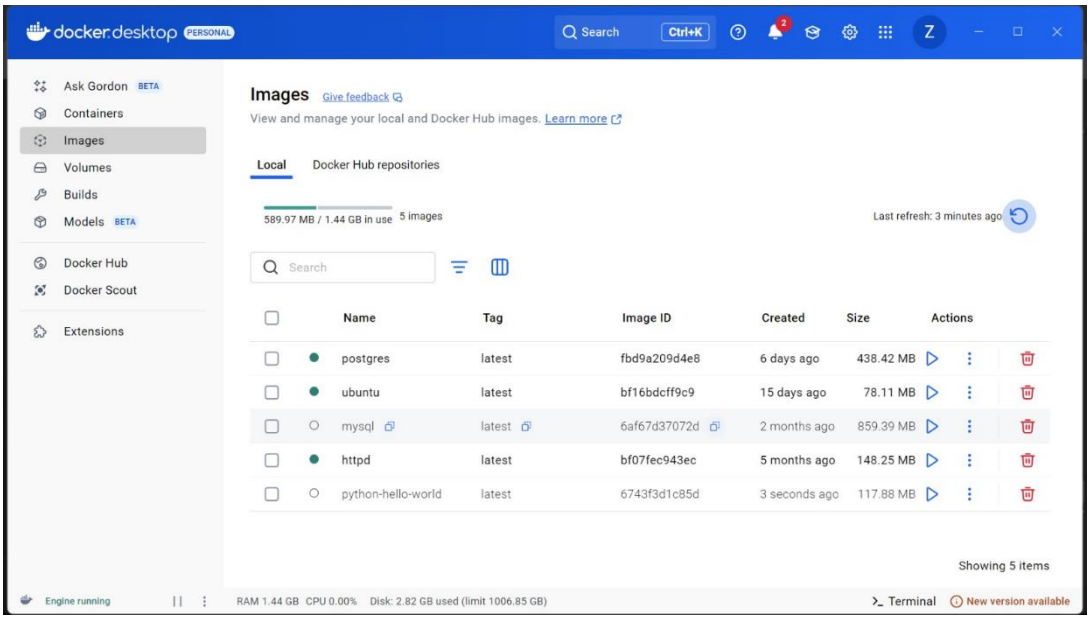


12. Install httpd – Run



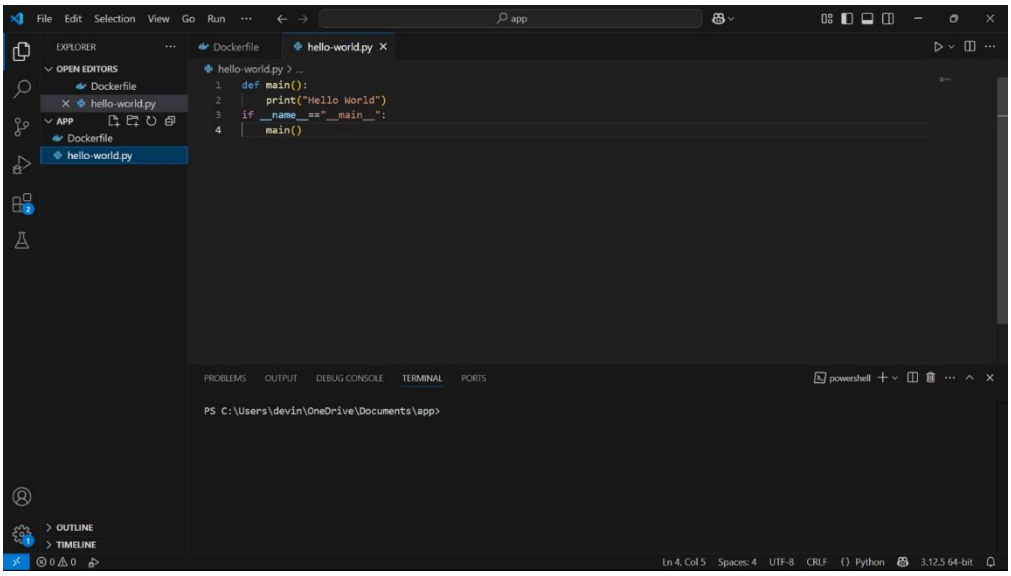


It works!

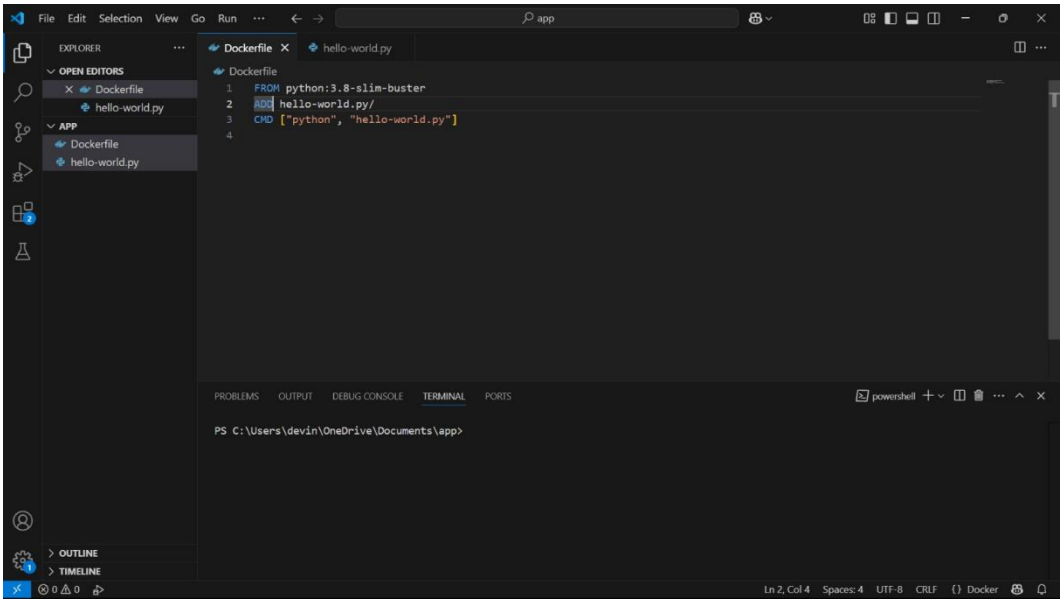


13. Membangun aplikasi di python

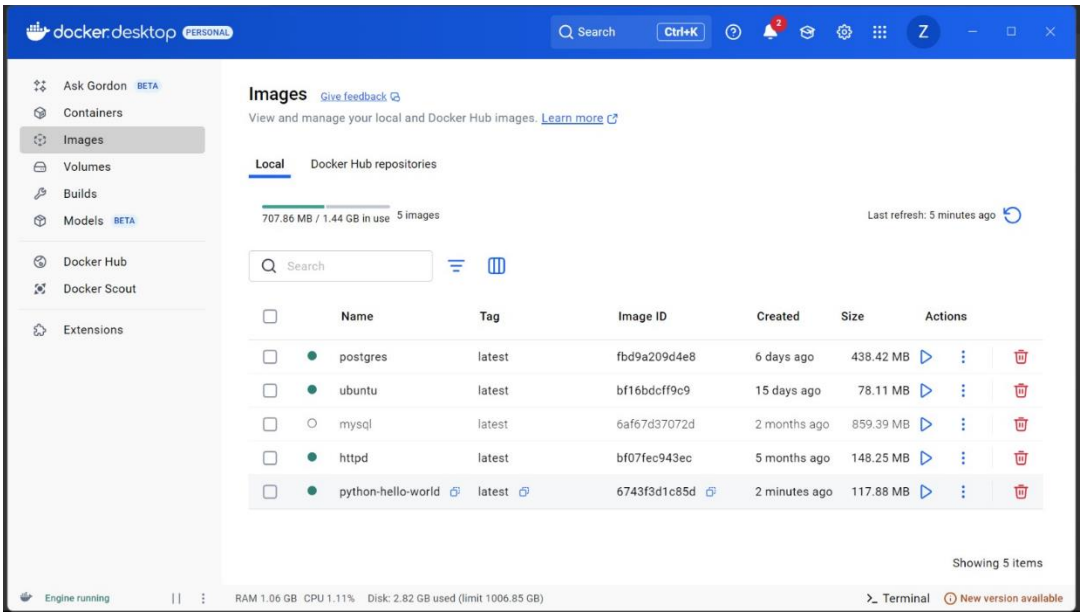
- Pastikan vscode ada ataudownload dan install jika belum ada
- Buatfile hello-world.p



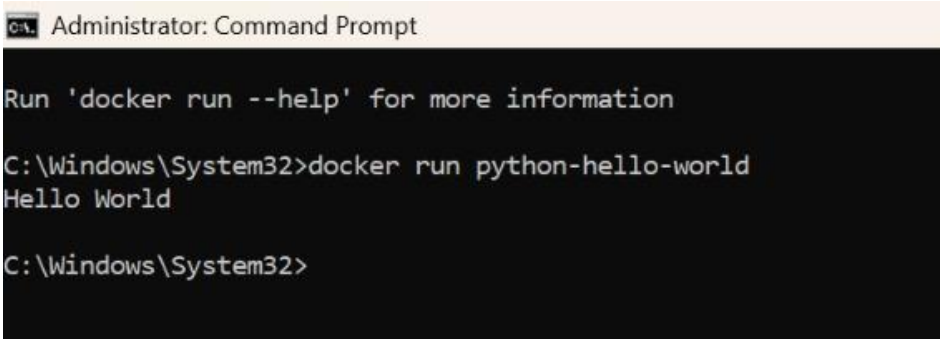
- Buat Dockerfile



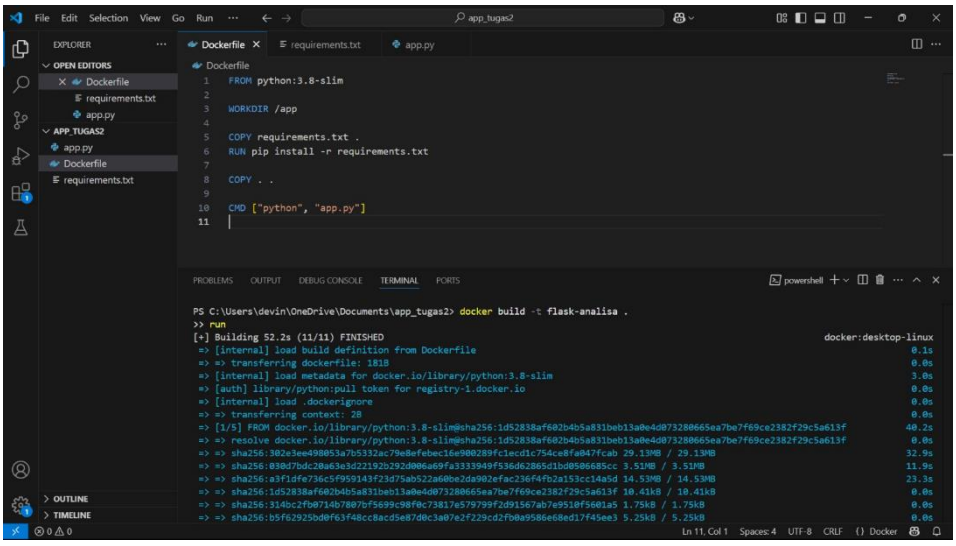
- Jalankan lihat docker desktop

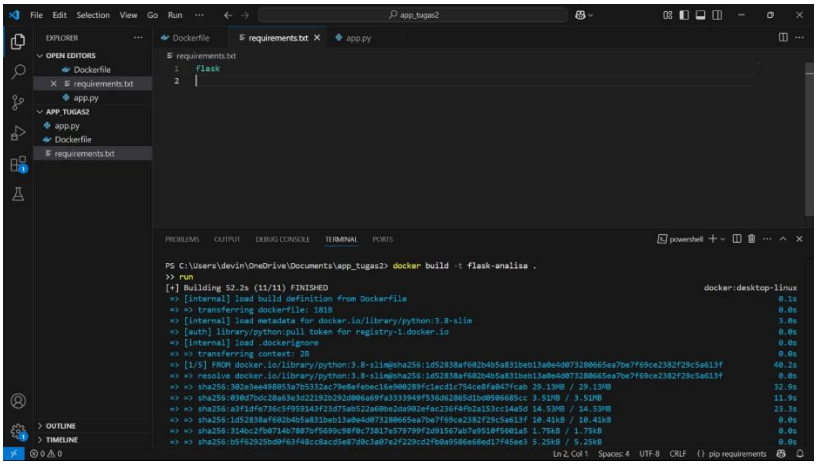


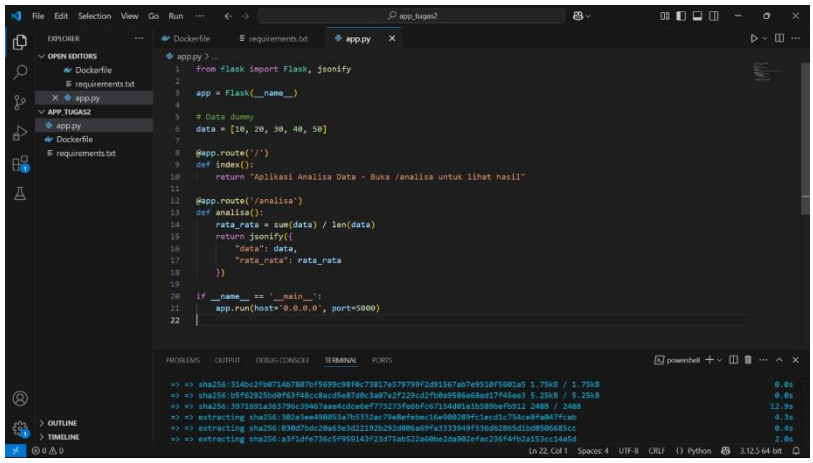
- Jalankan docker run python

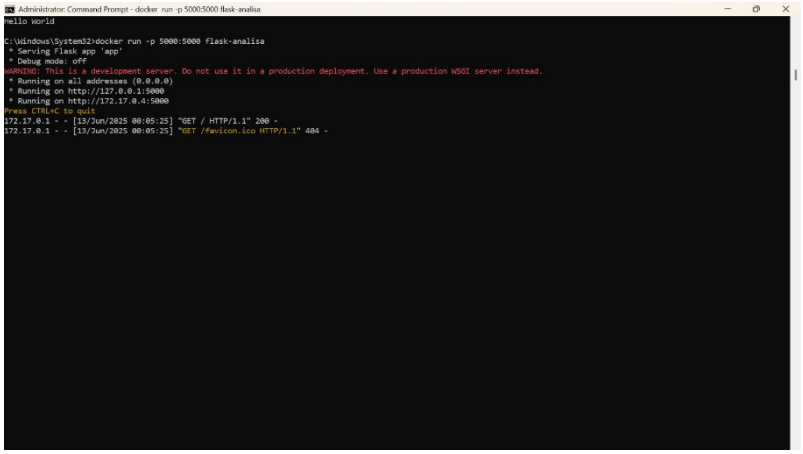


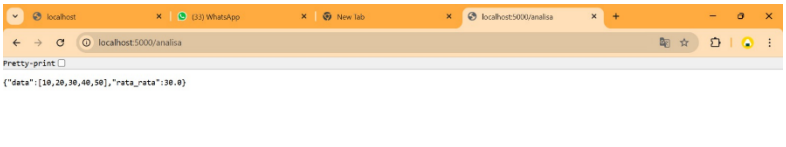
➢ Buat sebuah aplikasi Analisa data sederhana python di docker yang bisa dijalankan via http dan bisa diakses hasilnya via browser ((aplikasi yang dibuat bebas))




File Edit Selection View Go Run
app_tugas2
EXPLORER
OPEN EDITORS
Dockerfile
requirements.txt
app.py
APP_TUGAS2
app.py
Dockerfile
requirements.txt
Dockerfile
requirements.txt
app.py
requirements.txt
1 flask
2
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
PS C:\Users\devin\OneDrive\Documents\app_tugas2> docker build -t flask-analisa .
>> run
[+] Building 52.2s (11/11) FINISHED
docker:desktop-linux
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 181B
=> [internal] load metadata for docker.io/library/python:3.8-slim
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/python:3.8-slim@sha256:1d52838af602b4b5a831beb13a0e4d073280665ea7be7f69ce2382f29c5a613f
=> => resolve docker.io/library/python:3.8-slim@sha256:1d52838af602b4b5a831beb13a0e4d073280665ea7be7f69ce2382f29c5a613f
=> => sha256:302e3ee498053a7b5332ac79e8efebec16e900289fc1ecd1c754ce8fa047fcab 29.13MB / 29.13MB
=> => sha256:030d7bdc20a63e3d22192b292d006a69fa3333949f536d62865d11bd0506685cc 1.51MB / 3.51MB
=> => sha256:a3f1dfe736c5f959143f23d75ab522a60be2da902efac236f4fb2a153cc14a5d 14.53MB / 14.53MB
=> => sha256:1d52838af602b4b5a831beb13a0e4d073280665ea7be7f69ce2382f29c5a613f 10.41kB / 10.41kB
=> => sha256:314bc2fb0714b7807bf5699c98f0c73817e579799f2d91567ab7e9510f5601a5 1.75kB / 1.75kB
=> => sha256:b5f62925bd0f63f48cc8acd5e87d0c3a07e2f229cd2fb0a9586e68ed17f45ee3 5.25kB / 5.25kB
OUTLINE
TIMELINE
Ln 2, Col 1 Spaces: 4 UTF-8 CRLF {} pip requirements


File Edit Selection View Go Run
app_tugas2
EXPLORER
OPEN EDITORS
Dockerfile
requirements.txt
app.py
APP_TUGAS2
app.py
Dockerfile
requirements.txt
Dockerfile
requirements.txt
app.py
app.py > ...
1 from flask import Flask, jsonify
2
3 app = Flask(__name__)
4
5 # Data dummy
6 data = [10, 20, 30, 40, 50]
7
8 @app.route('/')
9 def index():
10 return "Aplikasi Analisa Data - Buka /analisa untuk lihat hasil"
11
12 @app.route('/analisa')
13 def analisa():
14 rata_rata = sum(data) / len(data)
15 return jsonify({
16 "data": data,
17 "rata_rata": rata_rata
18 })
19
20 if __name__ == '__main__':
21 app.run(host='0.0.0.0', port=5000)
22
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
=> => sha256:314bc2fb0714b7807bf5699c98f0c73817e579799f2d91567ab7e9510f5601a5 1.75kB / 1.75kB
=> => sha256:b5f62925bd0f63f48cc8acd5e87d0c3a07e2f229cd2fb0a9586e68ed17f45ee3 5.25kB / 5.25kB
=> => sha256:997169la363796c39467aae4cdce6ef771273fe6bfc6715 4d01e1b5809ef b912 2489 / 2489
=> => extracting sha256:302e3ee498053a7b5332ac79e8efebec16e900289fc1ecd1c754ce8fa047fcab
=> => extracting sha256:030d7bdc20a63e3d22192b292d006a69fa3333949f536d62865d11bd0506685cc
=> => extracting sha256:a3f1dfe736c5f959143f23d75ab522a60be2da902efac236f4fb2a153cc14a5d
OUTLINE
TIMELINE
Ln 22, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.5 64-bit


Administrator: Command Prompt - docker run -p 5000:5000 flask-analisa
Hello World
C:\Windows\System32>docker run -p 5000:5000 flask-analisa
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.4:5000
Press CTRL+C to quit
172.17.0.1 - - [13/Jun/2025 00:05:25] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [13/Jun/2025 00:05:25] "GET /favicon.ico HTTP/1.1" 404 -


localhost
(33) WhatsApp
New tab
localhost:5000/analisa
localhost:5000/analisa
Pretty-print
{"data":[10,20,30,40,50],"rata_rata":30.0}

Link GDrive :

https://drive.google.com/drive/folders/1fG-bfUehc9zl8Fi-78yKeTlFXxX_Odjm?usp=drive_link