LAPORAN PRAKTIKUM

DOCKER

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

Disusun untuk Memenuhi Tugas

Mata Kuliah:

Manajemen Data

Oleh:

HAFIZH MUHAMMAD ILHAM

NRP. 3324600043

PROGRAM STUDI D4 SAINS DATA TERAPAN

DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

TUGAS 1

1. Script SSH Monitor

Soal:

Buat script untuk melakukan pengecekan service ssh secara berkala dengan interval waktu tiap 10 detik dan memberikan notifikasi ke layar jika service ssh mati.

Command:

nano ssh_monitor.sh

isi file:

```
#!/bin/bash

while true; do
   if systemctl is-active --quiet ssh; then
       echo "$(date): SSH service is running"
   else
       echo "$(date): SSH service is DOWN - Starting SSH service"
       systemctl start ssh
   fi
   sleep 10
done
```

Output:

```
hafizhilham24@ubunt:~$ ./ssh_monitor.sh
Thu Jun 12 04:15:35 AM UTC 2025: SSH service is running
Thu Jun 12 04:15:55 AM UTC 2025: SSH service is running
Thu Jun 12 04:16:05 AM UTC 2025: SSH service is running
Thu Jun 12 04:16:30 AM UTC 2025: SSH service is running
```

2. Script Backup Berkala

Soal:

Buat script backup direktori tertentu dan jalankan backup secara berkala setiap 15 detik dengan backup ke file sd 10 dan kembali lagi menimpa file jika sudah mencapai file ke 10 jalankan selama 1 jam.

Command Jawaban:

bash

nano backup_script.sh

Isi file:

```
#!/bin/bash

while true; do
    TIMESTAMP=$(date +%Y%m%d_%H%M%S)
    BACKUP_FILE="backup_music_$TIMESTAMP.tar.gz"

    # Backup direktori Music ke direktori backup
    tar -czf backup/$BACKUP_FILE Music/
    echo "$(date): Music backup created: $BACKUP_FILE"

    # Hapus backup yang lebih dari 10 file
    cd backup
    ls -t backup_music_*.tar.gz 2>/dev/null | tail -n +11 | xargs -r rm
    cd ..
    echo "$(date): Old music backups cleaned"

    sleep 15
done
```

menjalankan script:

```
hafizhilham24@ubunt:~$ ./backup_script.sh
Thu Jun 12 04:09:39 AM UTC 2025: Music backup created: backup_music_20250612_
040939.tar.gz
Thu Jun 12 04:09:39 AM UTC 2025: Old music backups cleaned
Thu Jun 12 04:09:55 AM UTC 2025: Music backup created: backup_music_20250612_
040955.tar.gz
Thu Jun 12 04:09:55 AM UTC 2025: Old music backups cleaned
^Z
[7]+  Stopped                 ./backup_script.sh
```

3. Script Kelembapan dan Suhu

Soal:

Buat file data.txt dengan isian app:

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 66 900

2025-05-27T08:45 27.1 67 950

- Buat script awk yang membaca file tersebut dan menghasilkan output sbb: Suhu maksimum: 27.1
- Buat script awk yang membaca file tersebut dan menghitung rata kelembaban dengan output sbb: Rata-rata kelembaban: 63.5

Command Jawaban:

Nano data.txt

Isi file:

```
  GNU nano 7.2                          data.txt
2025-05-25T08:00 24.5 60 800
2025-05-27T08;15 25.0 62 850
2025-05-27T08:30 26.2 66 900
2025-05-27T08:45 27.1 67 950
```

Nano max_temp.sh

Isi file:

```
#!/bin/bash
awk '{if(NR==1 || $2>max) max=$2} END {print max}' data.txt
```

Nano avg_humidity.sh

Isi file:

```
#!/bin/bash
awk '{sum+=$3; count++} END {print sum/count}' data.txt
```

Menjalankan script:

```
hafizhilham24@ubunt:~$ ./max_temp.sh
27.1
```

```
hafizhilham24@ubunt:~$ ./avg_humidity.sh
63.75
```

4. Script Filter/ctc/passwd

Soal:

4. Buatkan perintah dilinux untuk meLihat file /etc/passwd, missal isinya sebegai berikut:

```
pulse:x:108:116:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
speech-dispatcher:x:109:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
avahi:x:110:118:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
saned:x:111:119::/var/lib/saned:/usr/sbin/nologin
colord:x:112:120:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
hplip:x:113:7:HPLIP system user,,,:/var/run/hplip:/bin/false
lightdm:x:114:121:Light Display Manager:/var/lib/lightdm:/bin/false
isbat:x:1000:1000:isbat,,,:/home/isbat:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
```

Lakukan Filter yang punya directory home dan ambil nama usernya, missal hasilnya sbb :

```
isbat:x:1000:1000:isbat,,,:/home/isbat:/bin/bash
testing:x:1001:1001:,,,:/home/testing:/bin/bash
test:x:1002:1002:,,,:/home/test:/bin/bash
```

Lakukan filter lagi sbb menjadi sbb :

```
isbat
testing
test
```

Tunjukkan step by step scriptnya dan hasilnya.

Buat dok file, Capture hasil program anda, dan pastikan script jalan capture hasil

1. Melihat file/etc/passwd

```
hafizhilham24@ubunt:~$ 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
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

2. Filter yang punya directory home dan ambil nama usernya

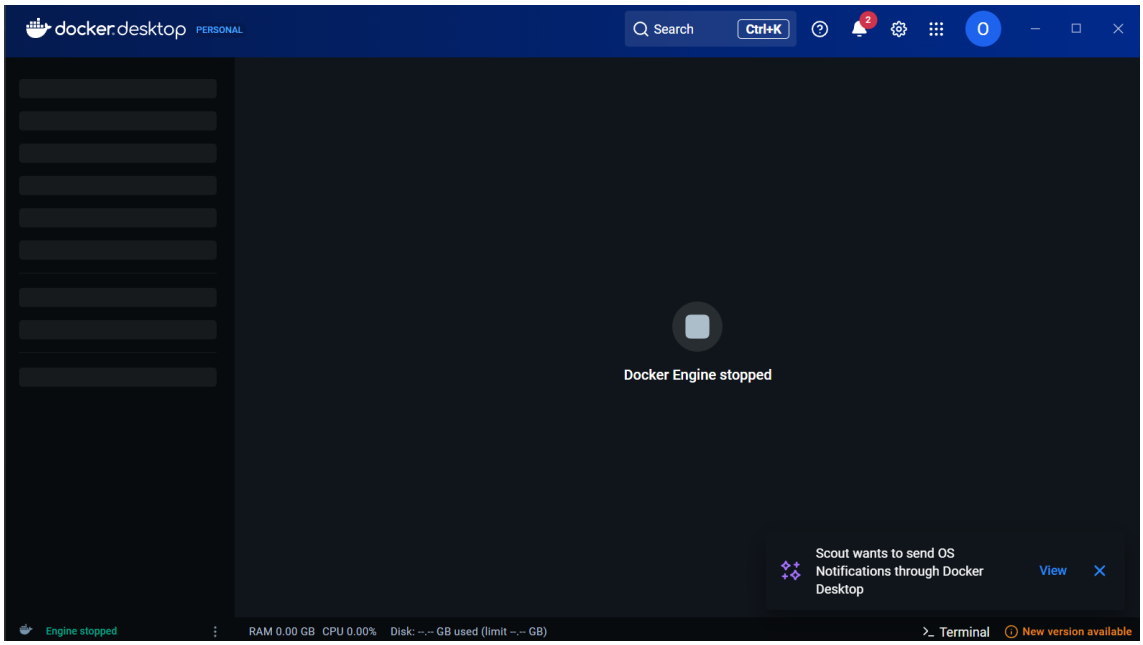
```
hafizhilham24@ubunt:~$ grep "/home/" /etc/passwd | cut -d: -f1
hafizhilham24
```

3. Filter untuk yang jadi sbb (nama user saja)

```
hafizhilham24@ubunt:~$ grep "/home/" /etc/passwd | cut -d: -f1 | head -3
hafizhilham24
```

TUGAS 2

1. Download dan install Docker Desktop



2. Install wsl

```
C:\Users\hafizh>wsl --status
Default Distribution: docker-desktop
Default Version: 2
```

3. Verifikasi instalasi

```
C:\Users\hafizh>docker --version
Docker version 28.1.1, build 4eba377
```

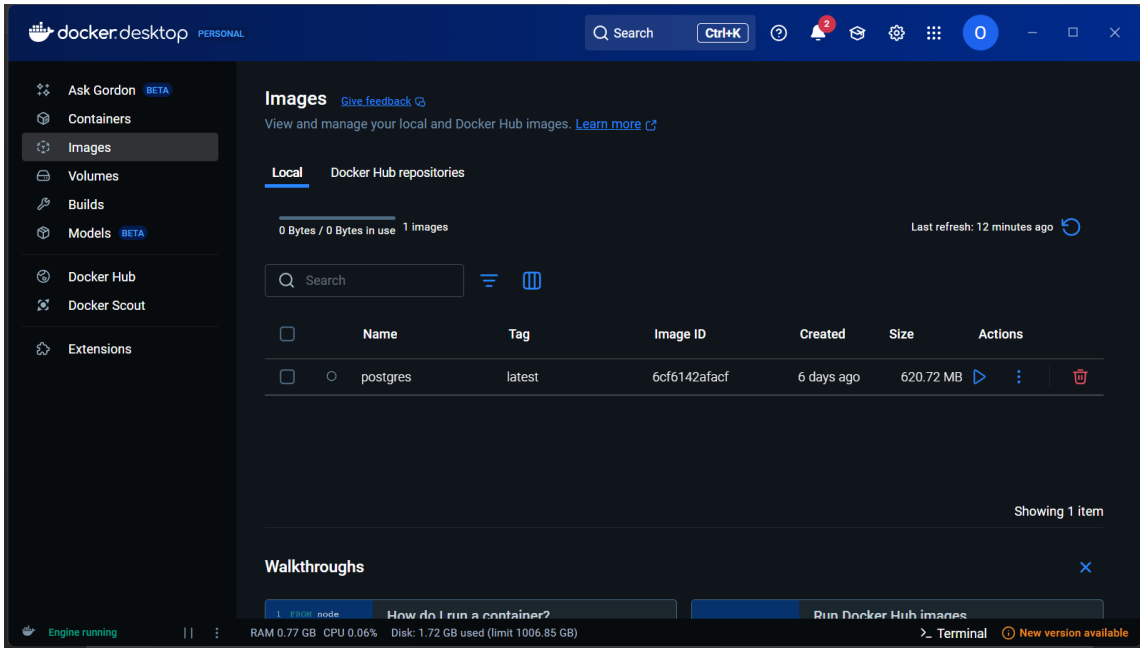
4. Pull image postgres

```
C:\Users\hafizh>docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
dad67da3f26b: Pull complete
c6def2c6e21d: Pull complete
3664068a9b37: Pull complete
64e8f1b2b243: Pull complete
f4ce9941f6e3: Pull complete
db3ab53631e4: Pull complete
603ef9fcdd8e: Pull complete
b47a445a47f0: Pull complete
abfd68ef219e: Pull complete
928d00623a6e: Pull complete
c95f49cc11b3: Pull complete
8a1f652e0c97: Pull complete
05b641b3bdab: Pull complete
eb3a531023c8: Pull complete
Digest: sha256:6cf6142afacfa89fb28b894d6391c7dcbf6523c33178bdc33e782b3b533a9342
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
```

Pengecekan status:

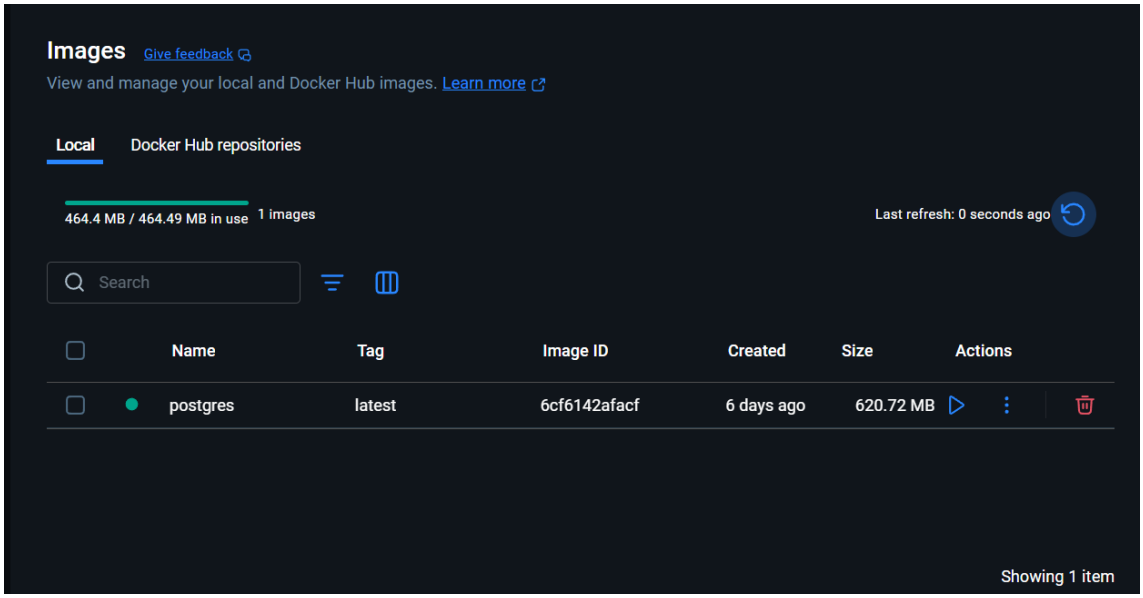
```
C:\Users\hafizh>docker image ls
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
postgres     latest    6cf6142afacf   6 days ago   621MB
```

5. List image di docker



6. Menjalankan container

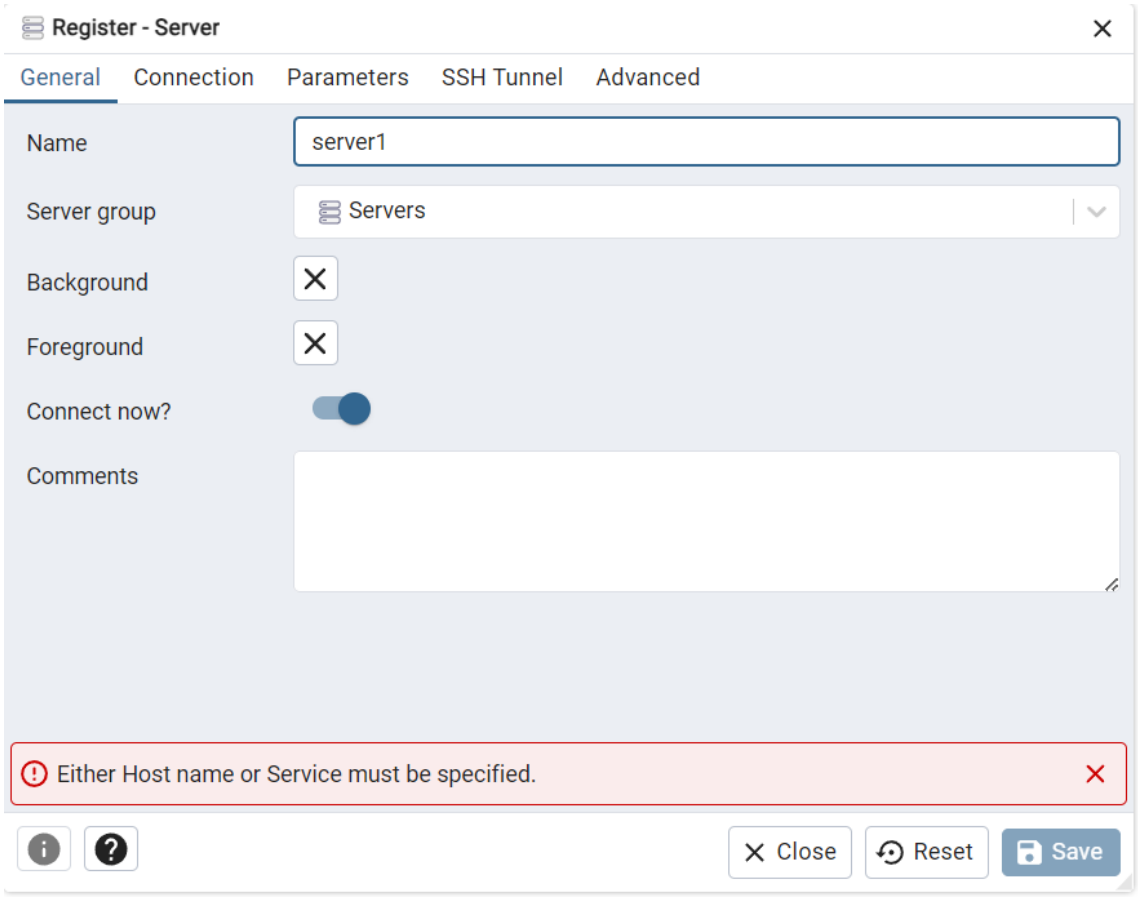
```
C:\Users\hafizh>docker run --name some-postgres -e POSTGRES_PASSWORD=123456 -d postgres
e1a621b0af29479d4c929d439f5b71fe10aa1524987b127a0bfc83180e2c0b6e

C:\Users\hafizh>docker run --name postgresserver1 -p 5431:5432 -e POSTGRES_PASSWORD=123456 -d postgres
e63c881267dc187d407eaee496d87b30023a009b3e5f30a3f5307bf5bba5bca1
```

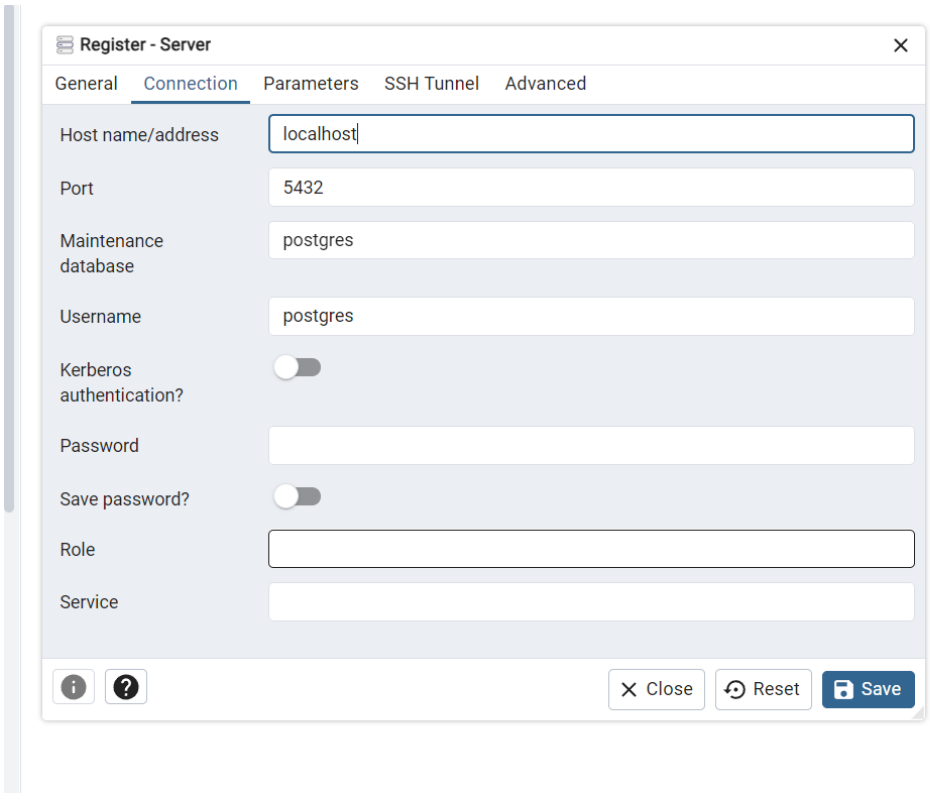


7. Menjalankan image

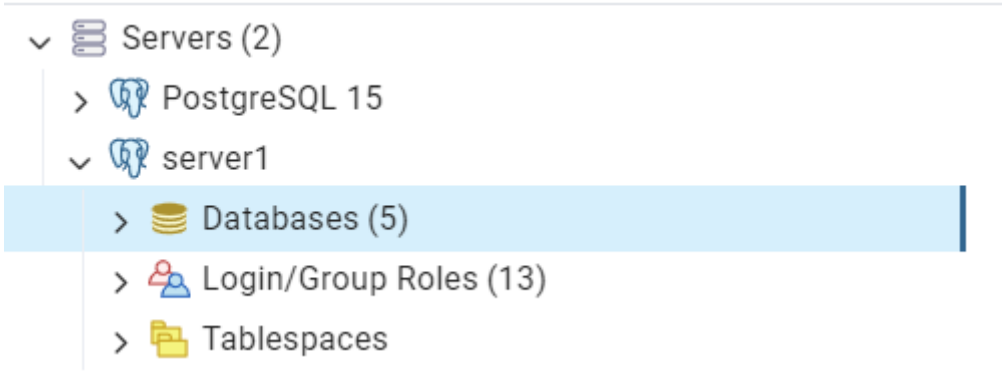
```
C:\Users\hafizh>docker run -it ubuntu
root@8c01170fc543:/# pwd
/
root@8c01170fc543:/# cd
root@8c01170fc543:~# pwd
/root
root@8c01170fc543:~#
root@8c01170fc543:~#
```

8. Koneksi ke postgres


Register - Server
General
Connection
Parameters
SSH Tunnel
Advanced
Name
server1
Server group
Servers
Background
Foreground
Connect now?
Comments
Either Host name or Service must be specified.
Close
Reset
Save


Register - Server
General
Connection
Parameters
SSH Tunnel
Advanced
Host name/address
localhost
Port
5432
Maintenance database
postgres
Username
postgres
Kerberos authentication?
Password
Save password?
Role
Service
Close
Reset
Save

Hasil :



9. Install httpd – run

```
C:\Users\hafizh>docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
c06cec1379c2: Pull complete
d1042d58e186: Pull complete
4f4fb700ef54: Pull complete
be5c5a616c3a: Pull complete
d0a755bf09a1: Pull complete
Digest: sha256:f6557a77ee2f16c50a5ccbb2564a3fd56087da311bf69a160d43f73b23d3af2d
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
```

Run :

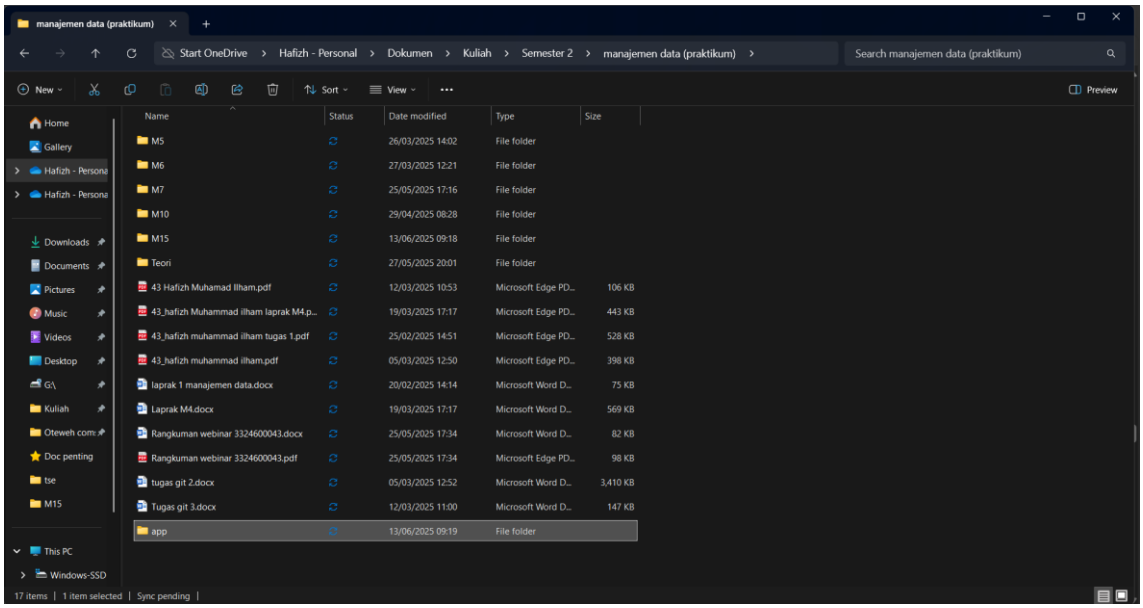
```
C:\Users\hafizh>docker run -d -p 80:80 --name my-apache httpd
cd00b9970ef1f49a90fbfb6d8b29f7f50e0e402c6df45e0a1be24c1c0c747de4
```

Cek:

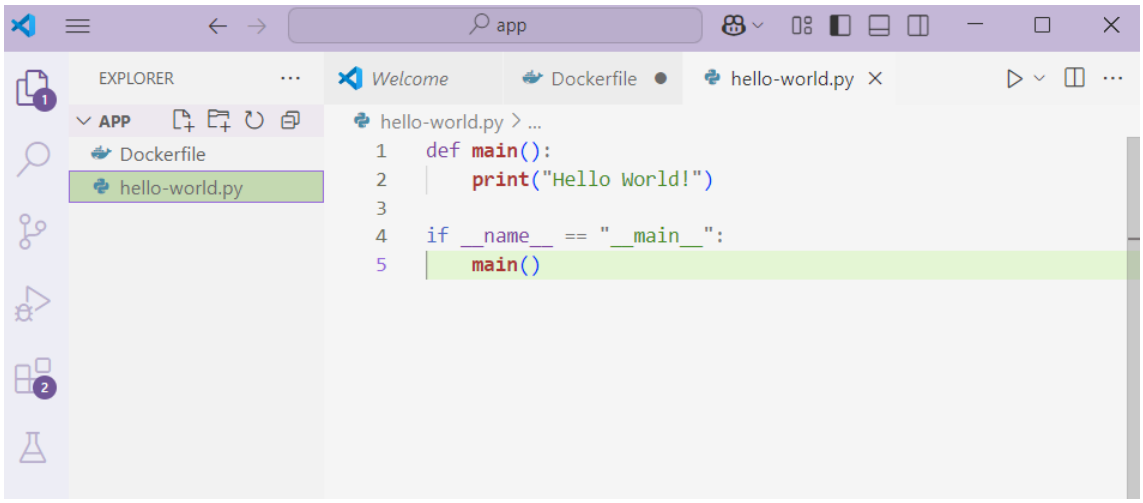
```
C:\Users\hafizh>docker image ls
REPOSITORY   TAG       IMAGE ID       CREATED         SIZE
postgres     latest    6cf6142afacf   6 days ago      621MB
ubuntu       latest    b59d21599a2b   2 weeks ago     117MB
httpd        latest    f6557a77ee2f   4 months ago    221MB
```

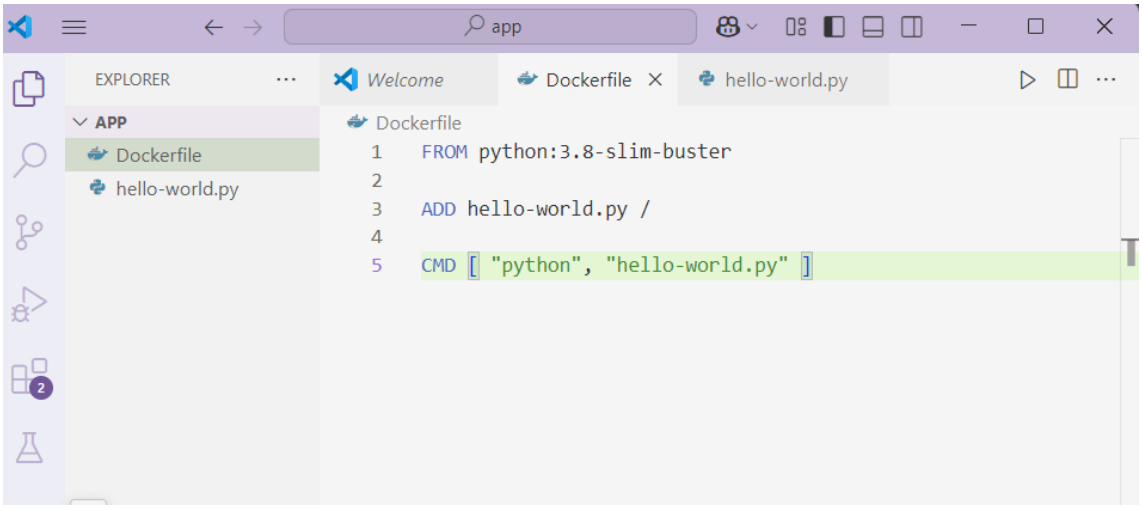
Membangun aplikasi di python

10. Buat directory



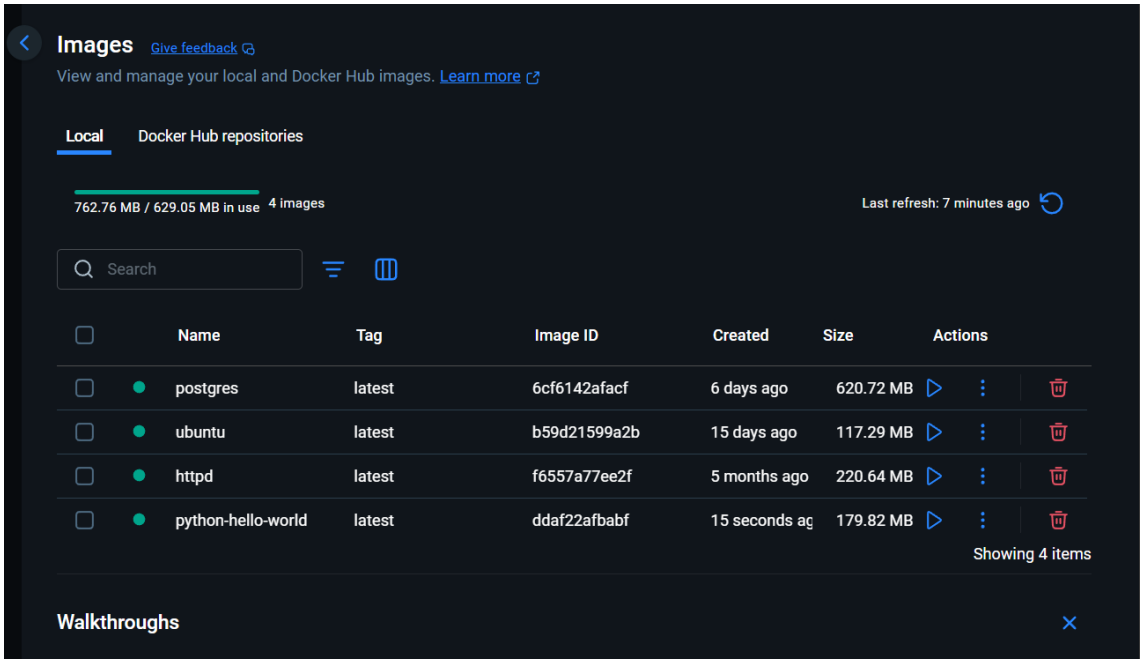
11. Buat file python dan bocker file





12. Membuat docker image

```
C:\Users\hafizh\OneDrive\Dokumen\Kuliah\Semester 2\manajemen data (praktikum)\app>docker build -t python-hello-world .
[+] Building 15.8s (8/8) FINISHED                                                                    docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                0.1s
 => => transferring dockerfile: 126B                                                                                0.0s
 => [internal] load metadata for docker.io/library/python:3.8-slim-buster                                           3.8s
 => [auth] library/python:pull token for registry-1.docker.io                                                       0.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [internal] load build context                                                                                   0.1s
 => => transferring context: 121B                                                                                   0.0s
 => [1/2] FROM docker.io/library/python:3.8-slim-buster@sha256:8799b0564103a9f36cfb8a8e1c562e11a9a6f2e3bb214e2ad   10.8s
 => => resolve docker.io/library/python:3.8-slim-buster@sha256:8799b0564103a9f36cfb8a8e1c562e11a9a6f2e3bb214e2adc   0.0s
 => => sha256:276dfcf5defffff3c5d540a8e0d9a18656a4c03637a8b4f4eec1f4a147799c901 3.14MB / 3.14MB                     0.6s
 => => sha256:cbfea27109a8b1136059a7973ccb8243889faf162ebc173a05909dcb0bec03c9 244B / 244B                          0.7s
 => => sha256:8f777578c172d018077d3dc22d6654911fff600660979943fe8c4697ecf8aac35 12.89MB / 12.89MB                   7.3s
 => => sha256:824416e234237961c9c5d4f41dfe5b295a3c35a671ee52889bfb08d8e257ec4c 2.78MB / 2.78MB                      1.2s
 => => sha256:8b91b88d557765cd8c6802668755a3f6dc4337b6ce15a17e4857139e5fc964f3 27.14MB / 27.14MB                    6.0s
 => => extracting sha256:8b91b88d557765cd8c6802668755a3f6dc4337b6ce15a17e4857139e5fc964f3                           2.4s
 => => extracting sha256:824416e234237961c9c5d4f41dfe5b295a3c35a671ee52889bfb08d8e257ec4c                           0.3s
 => => extracting sha256:8f777578c172d018077d3dc22d6654911fff600660979943fe8c4697ecf8aac35                          0.7s
 => => extracting sha256:cbfea27109a8b1136059a7973ccb8243889faf162ebc173a05909dcb0bec03c9                           0.0s
 => => extracting sha256:276dfcf5defffff3c5d540a8e0d9a18656a4c03637a8b4f4eec1f4a147799c901                          0.5s
 => [2/2] ADD hello-world.py /                                                                                      0.4s
 => exporting to image                                                                                              0.4s
 => => exporting layers                                                                                             0.1s
 => => exporting manifest sha256:242d0890232a517cb4a5f045009a26a5f4aeebb8a53364112a5bdfc61a967adc                   0.0s
 => => exporting config sha256:bfe89e973435fb85f12d2463af4d5038b3fbaea9071bdf29e1ba6e70620ad6e9                     0.0s
 => => exporting attestation manifest sha256:5c354bb62c65bd8231c4f4348f865c4c169bbb3a1b3676e4e8469b16b8011c21      0.0s
 => => exporting manifest list sha256:ddaf22afbabf6cde8361c835e99f6b3362362597854c6e964e19abf2f454cec2             0.0s
 => => naming to docker.io/library/python-hello-world:latest                                                        0.0s
 => => unpacking to docker.io/library/python-hello-world:latest                                                     0.0s
```



13. Menjalankan Program

```
C:\Users\hafizh\OneDrive\Dokumen\Kuliah\Semester 2\manajemen data (praktikum)\app>docker run python-hello-world
Hello World!
```

Membuat Aplikasi Analisa Data Web

1. Buat program python

```python
from flask import Flask

app = Flask(__name__)

# Data sederhana
data = [
    {"nama": "Alice", "nilai": 85},
    {"nama": "Bob", "nilai": 92},
    {"nama": "Charlie", "nilai": 78},
    {"nama": "Diana", "nilai": 88}
]

@app.route('/')
def home():
    return '''
    <h1>Aplikasi Analisa Data Sederhana</h1>
    <p><a href="/data">Lihat Data</a></p>
    <p><a href="/rata">Lihat Rata-rata</a></p>
    '''

@app.route('/data')
def show_data():
    html = "<h2>Data Siswa:</h2><ul>"
    for siswa in data:
        html += f"<li>{siswa['nama']}: {siswa['nilai']}</li>"
    html += "</ul><p><a href='/'>Kembali</a></p>"
    return html

@app.route('/rata')
def rata_rata():
    total = sum(siswa['nilai'] for siswa in data)
    rata = total / len(data)
    return f'''
    <h2>Analisa Data:</h2>
    <p>Jumlah Siswa: {len(data)}</p>
    <p>Total Nilai: {total}</p>
    <p>Rata-rata: {rata:.1f}</p>
```

2. Buat dockerfile dan requirment file

```
Welcome | app.py | requirment.txt | dockerfile ×
dockerfile > ...
1   FROM python:3.9-slim
2
3   WORKDIR /app
4
5   COPY requirements.txt .
6   RUN pip install -r requirements.txt
7
8   COPY app.py .
9
10  EXPOSE 5000
11
12  CMD ["python", "app.py"]
```

```
Welcome | app.py | requirment.txt × | dockerfile
requirment.txt
1   Flask==2.3.3
```

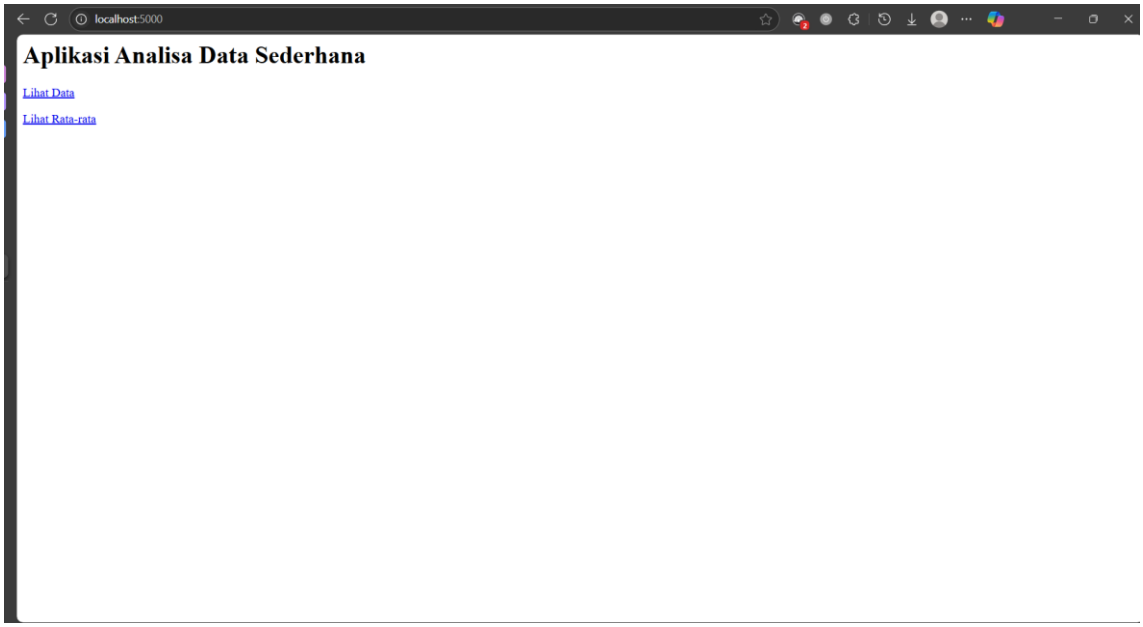
3. Build docler image melalui cmd

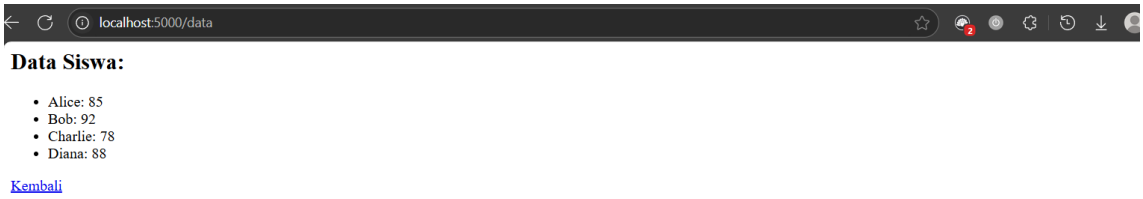
```
C:\Users\hafizh\OneDrive\Dokumen\Kuliah\Semester 2\manajemen data (praktikum)\app_program_data>docker build -t my-app .
[+] Building 15.2s (10/10) FINISHED                                                                    docker:desktop-linux
 => [internal] load build definition from dockerfile                                                                0.0s
 => => transferring dockerfile: 199B                                                                                0.0s
 => [internal] load metadata for docker.io/library/python:3.9-slim                                                  1.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [1/5] FROM docker.io/library/python:3.9-slim@sha256:a40cf9eba2c3ed9226afa9ace504f07ad30fe831343bb1c69f7a6707a   4.8s
 => => resolve docker.io/library/python:3.9-slim@sha256:a40cf9eba2c3ed9226afa9ace504f07ad30fe831343bb1c69f7a6707a   0.0s
 => => sha256:ea13ebdb5390b4e3fa5651d1daf14e6756a134f9a168fbedde44f02b1cee5fa8 250B / 250B                          0.4s
 => => sha256:cd1e1b7e12d38cac4095e7ea4f161334542f130d381d6ef2013fa1ac01b4b6b0 14.94MB / 14.94MB                    2.5s
 => => sha256:20a97c0d8fc11f8337ff080be3f192c7211a0b7d1e6b886d6d2cff6674761652 3.51MB / 3.51MB                      3.0s
 => => extracting sha256:20a97c0d8fc11f8337ff080be3f192c7211a0b7d1e6b886d6d2cff6674761652                           0.3s
 => => extracting sha256:cd1e1b7e12d38cac4095e7ea4f161334542f130d381d6ef2013fa1ac01b4b6b0                           1.3s
 => => extracting sha256:ea13ebdb5390b4e3fa5651d1daf14e6756a134f9a168fbedde44f02b1cee5fa8                           0.0s
 => [internal] load build context                                                                                   0.0s
 => => transferring context: 85B                                                                                    0.0s
 => [2/5] WORKDIR /app                                                                                              0.2s
 => [3/5] COPY requirements.txt .                                                                                   0.1s
 => [4/5] RUN pip install -r requirements.txt                                                                       6.7s
 => [5/5] COPY app.py .                                                                                             0.1s
 => exporting to image                                                                                              2.0s
 => => exporting layers                                                                                             1.1s
 => => exporting manifest sha256:6a53aa500d7cd0d32c2588d8965dcf504c866c55ae629be959433173284ebcc0                   0.0s
 => => exporting config sha256:9799a24408aa9aa0cf114f5c3c3eb69e9b6da62c2963dd950e28c634678a3b5d                     0.0s
 => => exporting attestation manifest sha256:1344db1b7d3ce0e6f81271ba3b25b7fc6d28390071b4b2714aeed24ba63e30d2       0.0s
 => => exporting manifest list sha256:508039be67f21c0f96c9f7e92aec2e8b2512b44d0d0aa6ad6e037aa6ee3486ec              0.0s
 => => naming to docker.io/library/my-app:latest                                                                    0.0s
```

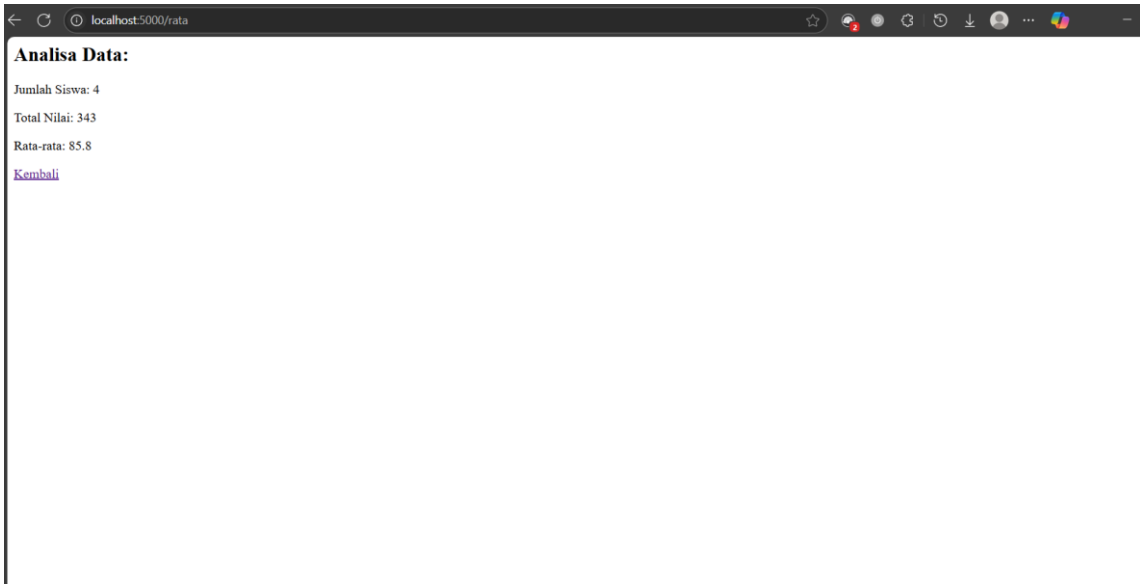
4. Menjalankan container

```
C:\Users\hafizh\OneDrive\Dokumen\Kuliah\Semester 2\manajemen data (praktikum)\app_program_data>docker run -p 5000:5000 m
y-app
 * 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://172.17.0.2:5000
Press CTRL+C to quit
```

5. Membuka program menggunakan browser







Dokumentasi tugas:

1. Youtube : https://youtu.be/FRZvH1vYDz0
2. Gitbub : hafizhilham24/Praktikum-docker