LAPORAN PRAKTIKUM 14

PRAKTIKUM MANAJEMEN DATA

PEMBAHASAN UAS & DOCKER

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

Disusun Oleh:

Dailatul Arofah

3324600046

Sains Data Terapan B

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER

PROGRAM STUDI SAINS DATA TERAPAN

PEMBAHASAN UAS

1. Membuat script untuk melakukan pengecekan service ssh secara berkala dengan interval waktu tiap 10 detik dan memberikan notifikasi ke layar jika service ssh mati.
    - Membuat file untuk diisi perintah

      ```
      dellarfh@della:~$ nano 46_monssh.sh
      ```

    - Menambahkan isi seperti di bawah dan disimpan

      ```
        GNU nano 7.2
      #!/bin/bash

      while true; do
              STATUS=$(systemctl is-active ssh)

              if [ "$STATUS" != "active" ]; then
                   echo " [$(date)] SERVICE SSH MATI! "
              else
                   echo " [$(date)] SSH aktif. "
              fi

              sleep 10
      done_
      ```

    - Membuka akses agar bisa diedit atau dijalankan

      ```
      dellarfh@della:~$ chmod +x 46_monssh.sh
      ```

    - Output

      ```
      dellarfh@della:~$ ./46_monssh.sh
       [Wed May 28 12:19:20 PM UTC 2025] SSH aktif.
       [Wed May 28 12:19:30 PM UTC 2025] SSH aktif.
      ```

2. Membuat script backup direktori tertentu dan jalankan backup secara berkala setiap 15 detik dengan backup ke file 1 sd 10 dan kembali lagi menimpa file 1 jika sudah file 10 dijalankan selama 1 jam.
    - Membuat direktori data dan file contoh

      ```
      dellarfh@della:~$ mkdir -p ~/data
      dellarfh@della:~$ echo "Example File" > ~/data/
      ```

    - Membuat direktori backup

      ```
      dellarfh@della:~$ mkdir -p ~/backup
      ```

    - Membuat dan mengedit script backup

      ```
      dellarfh@della:~$ nano 46_backup.sh
      ```

      ```
        GNU nano 7.2                                                    46_
      #!/bin/bash

      src_dir = "$HOME/data"
      backup_dir = "$HOME/backup"
      max_files = 10
      interval = 15
      runtime = 3600 #1jam = 3600 detik

      start_time = $(date + %s)

      while [ $(($(date + %s) - start_time)) -lt $runtime ]; do
              index=$(( (($(date + %s) -start_time) / interval) % max_files + 1 ))
              filename="file${index}.tar.gz"
              tar -czf "$backup_dir/$filename" -C "$src_dir" .
              echo "$(date): Backup disimpan ke $backup_dir/$filename"
              sleep $interval
      done
      ```

- Memberikan izin eksekusi pada script

```
dellarfh@della:~$ chmod +x 46_backup.sh
```

- Menjalankan script

```
dellarfh@della:~$ ./46_backup.sh
./46_backup.sh: line 3: src_dir: command not found
./46_backup.sh: line 4: backup_dir: command not found
./46_backup.sh: line 5: max_files: command not found
./46_backup.sh: line 6: interval: command not found
./46_backup.sh: line 7: runtime: command not found
date: extra operand ‘%s’
Try 'date --help' for more information.
./46_backup.sh: line 9: start_time: command not found
./46_backup.sh: line 12: syntax error near unexpected token `)'
./46_backup.sh: line 12: `       index=$(( ($date + 5s) -start_time) / interval
dellarfh@della:~$
```

- Mengedit crontab untuk menjalankan script secara berkala (setelah script diperbaiki)

```
> crontab -e
```

```
  GNU nano 7.2                                                    /tmp/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 1 * * 0 /tmp/crontab.GMWauV/crontab/46_backup.sh
```

- Melihat entri crontab yang sudah diinstal

```
crontab: installing new crontab
> crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 1 * * 0 /tmp/crontab.GMWauV/crontab/46_backup.sh
>
```

3. Membuat script awk dari suatu file, lalu menghitung rata-rata dan menampilkan nilai maksimum dari file tersebut.
    - Menuliskan sebuah data yang nantinya akan diinputkan ke file data.txt

```
dellarfh@della:~$ cat > data.txt <<EOF
> timestamp suhu kelembapan 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
> EOF
dellarfh@della:~$
```

    - Membuat sebuah file untuk diisi perintah

```
dellarfh@della:~$ nano 46_suhukelembapan.sh
```

    - Menambahkan isi seperti di bawah dan disimpan

```
  GNU nano 7.2                                                    46_
#!/bin/bash

echo -n "Suhu maksimum: "
awk 'NR>1 { if ($2 > max) max=$2 } END { print max }' data.txt

echo -n "Rata-rata kelembapan: "
awk 'NR>1 { total += $3; count++ } END { print total/count }' data.txt
```

    - Output

```
ellarfh@della:~$ ./46_suhukelembapan.sh
uhu maksimum: 27.1
ata-rata kelembapan: 63.5
```

4. Membuat perintah di linux untuk melihat isi file /etc/passwd, lalu melakukan filter yang memiliki directory home dan mengambil nama user nya, dan melakukan filtering lagi.
    - Melihat isi file /etc/passwd

```
dellarfh@della:~$ cat /etc/passwd
```

    - Memfilter baris yang mengandung /home

```
dellarfh@della:~$ grep "/home" /etc/passwd
dellarfh@della:~$ grep "/home" /etc/passwd
dellarfh:x:1000:1000:dellaa,22,0878663509449,0343125678:/home/dellarfh:/bin/bash
paul:x:1001:1001::/home/paul:/bin/sh
jane:x:1002:1001::/home/jane:/bin/sh
alice:x:1003:1002::/home/alice:/bin/sh
derek:x:1004:1002::/home/derek:/bin/sh
```

    - Memfilter baris dan mengekstrak nama user dengan cut

```
dellarfh@della:~$ grep "/home" /etc/passwd | cut -d: -f1
dellarfh
paul
jane
alice
derek
dellarfh@della:~$
```

    - Membuat dan mengedit script shell, lalu disimpan

```
dellarfh@della:~$ nano 46_filter.sh
```

```
  GNU nano 7.2
#!/bin/bash

echo "Daftar user dengan direktori /home: "
awk -F: '/\/home/ { print $1 }' /etc/passwd
```

- Memberikan izin eksekusi pada script

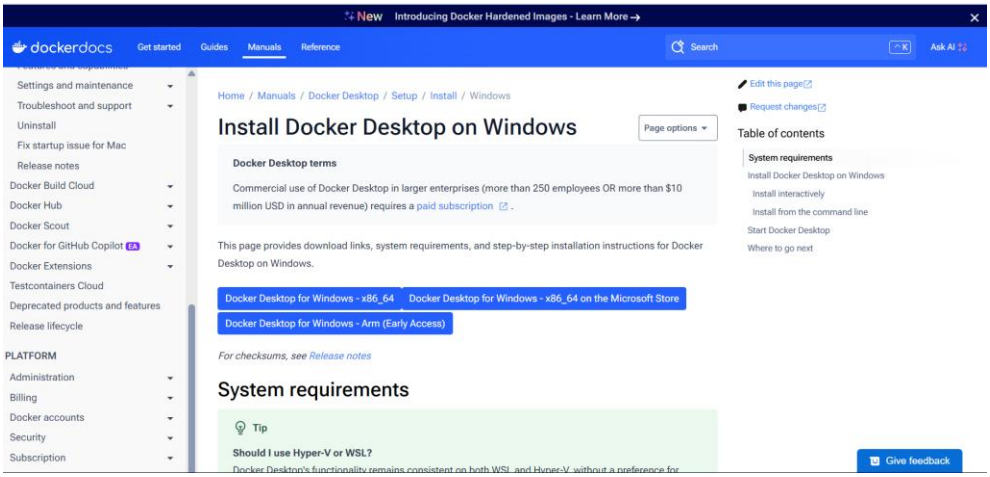
```
dellarfh@della:~$ chmod +x 46_filter.sh
```

- Output

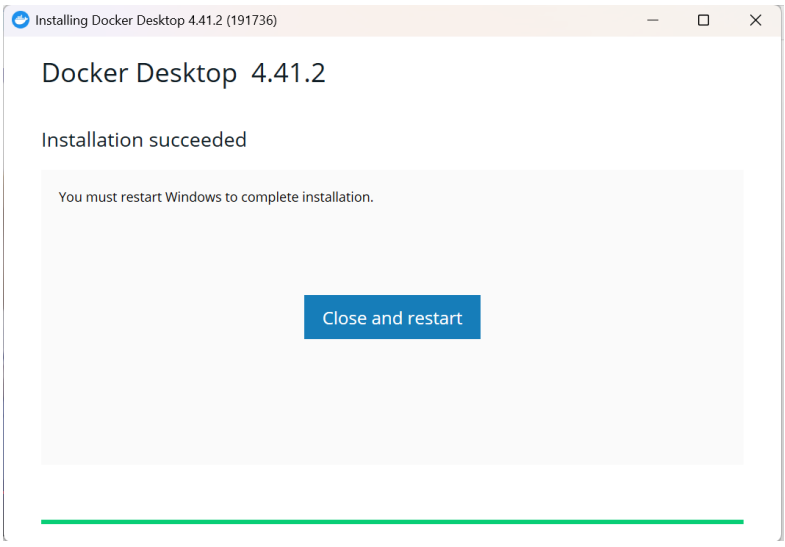
```
dellarfh@della:~$ ./46_filter.sh
Daftar user dengan direktori /home:
dellarfh
paul
jane
alice
derek
dellarfh@della:~$
```

INSTALASI DOCKER DAN MEMBUAT ISI IMAGE

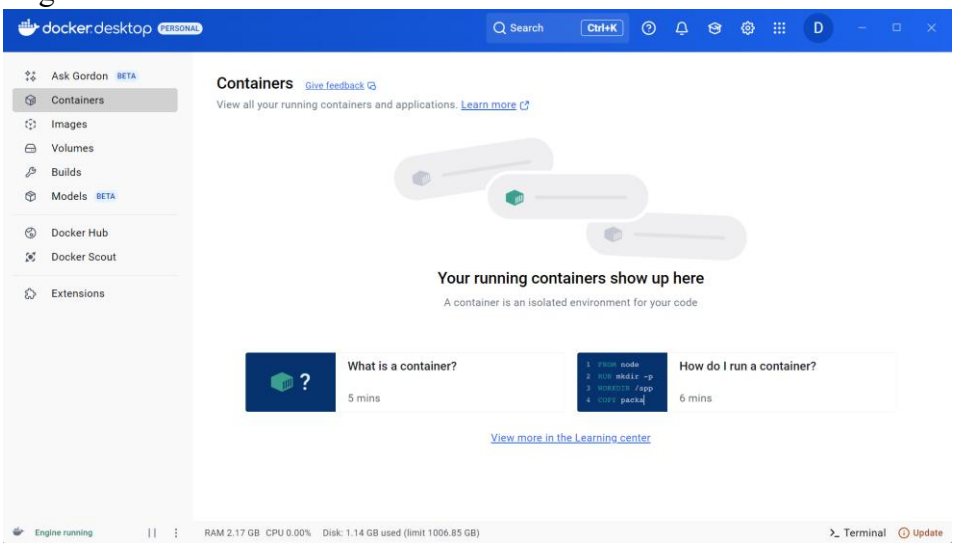
1. Install Docker



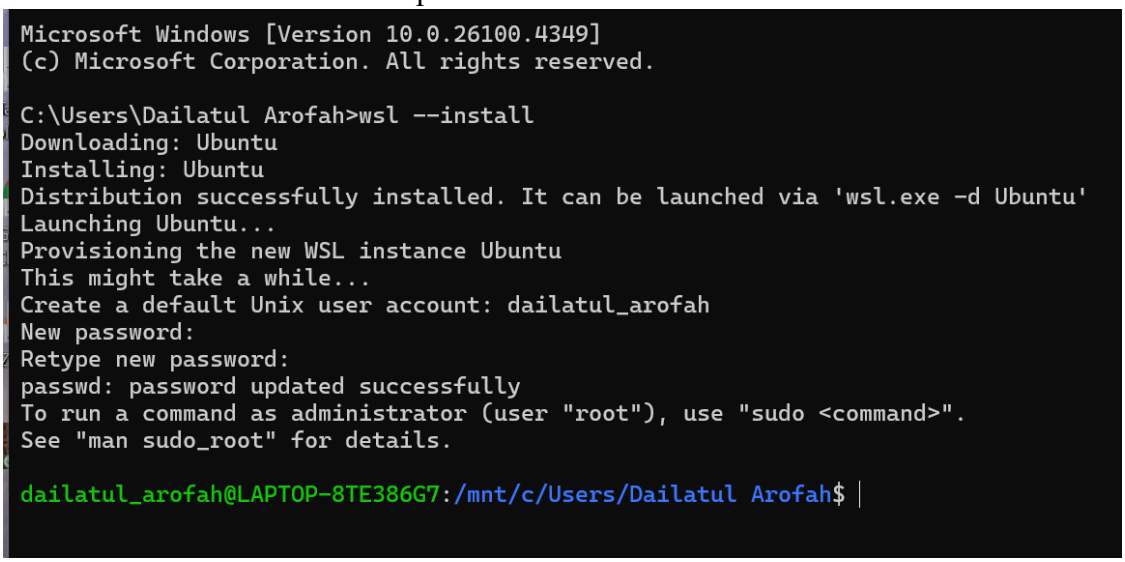
2. Restart



3. Login dan Pastikan Sudah Masuk



4. Install WSL di Command Prompt



5. Ketik ‘docker image ls’

6. Ketik ‘docker imgaes’

7. Mengambil data dari postgres

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

8. Cek kembali images

```
C:\Users\Dailatul Arofah>docker images
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
postgres     latest    6cf6142afacf   4 days ago   621MB
```

9. Cek version docker

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

10. Mengambil data dari mysql

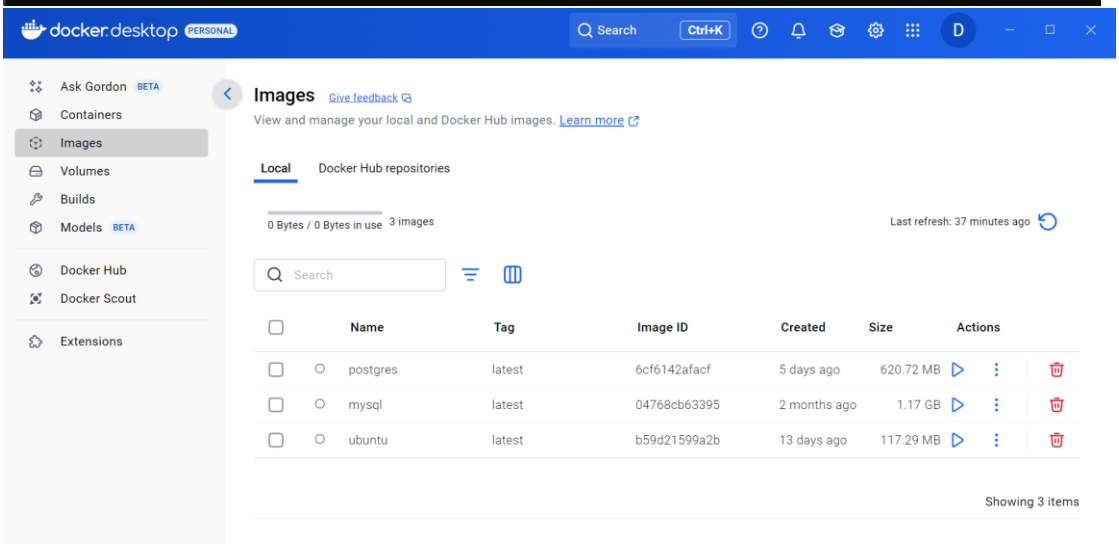
```
C:\Users\Dailatul Arofah>docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
5056ce4ab875: Pull complete
d23320eed97a: Pull complete
7074f55c9a02: Pull complete
7488ffd7127f: Pull complete
72ac912b8a2e: Pull complete
8a50ff4ab30c: Pull complete
b288ccce2510: Pull complete
9845df06f911: Pull complete
b097427f1ebe: Pull complete
4bd1fb59dd90: Pull complete
Digest: sha256:04768cb63395f56140b4e92cad7c8d9f48dfa181075316e955da75aadca8a7cd
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
```

11. Mengambil data dari ubuntu

```
C:\Users\Dailatul Arofah>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
```

12. Mengecek keberhasilan pengambilan

```
C:\Users\Dailatul Arofah>docker image ls
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
postgres     latest    6cf6142afacf   4 days ago     621MB
ubuntu       latest    b59d21599a2b   13 days ago    117MB
mysql        latest    04768cb63395   8 weeks ago    1.17GB
```



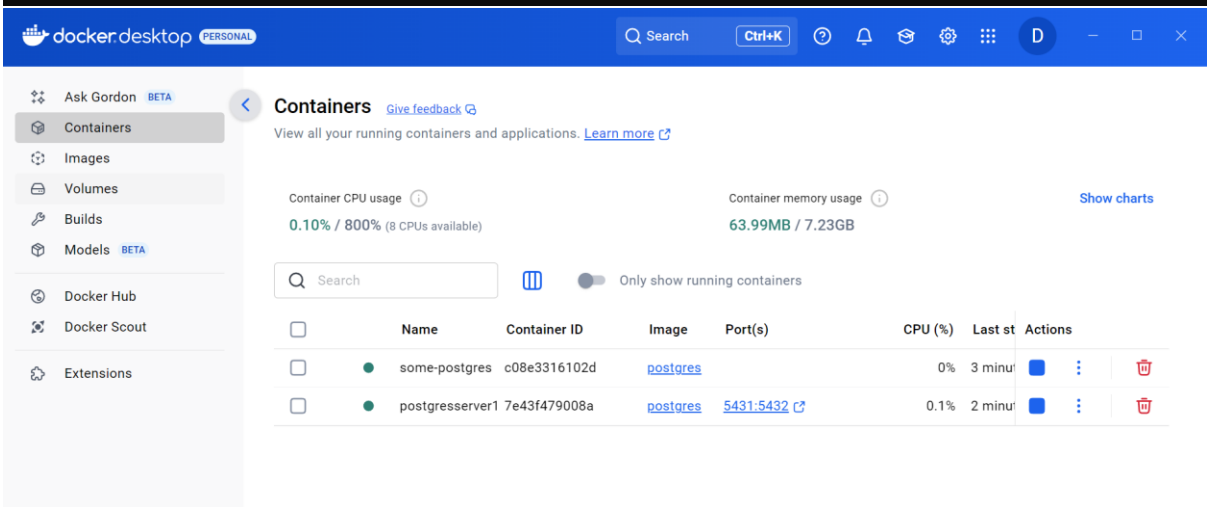
13. Mengecek isi dari docker

```
C:\Users\Dailatul Arofah>docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

14. Menjalankan 2 container

```
C:\Users\Dailatul Arofah>docker run --name some-postgres -e POSTGRES_PASSWORD=123456 -d postgres
c08e3316102dd605a75db723063f2b5e091ff5a27da0f1e22582da505d685f46
```

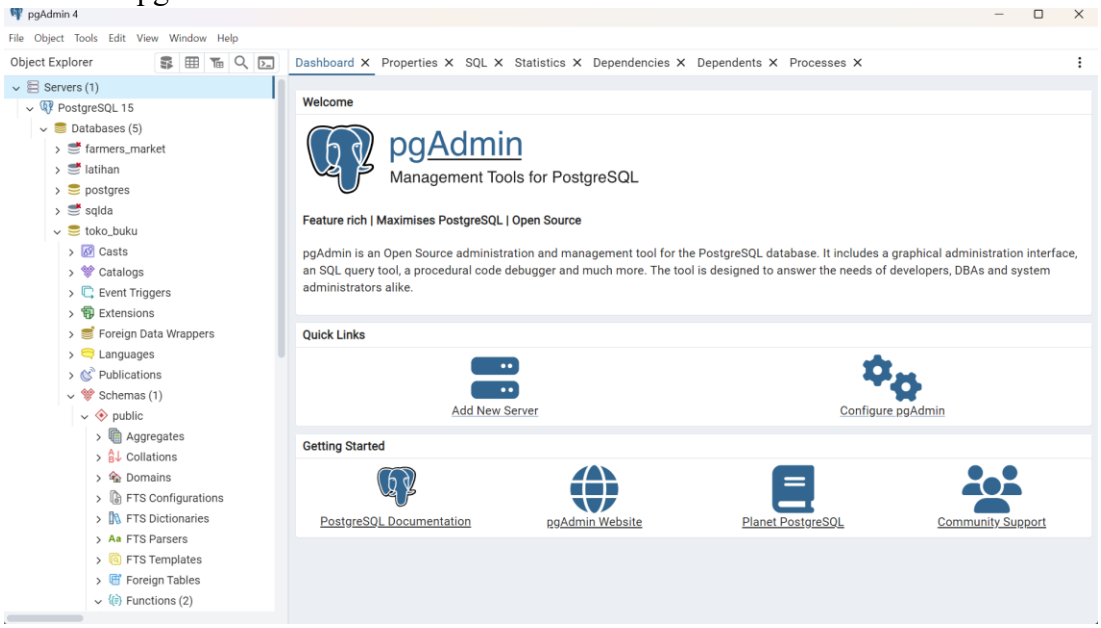
```
C:\Users\Dailatul Arofah>docker run --name postgresserver1 -p 5431:5432 -e POSTGRES_PASSWORD=123456 -d postgres
7e43f479008a50d459daee701e9366b0d420f6b43ce5d5ac2c5bd70df8d2578c
```

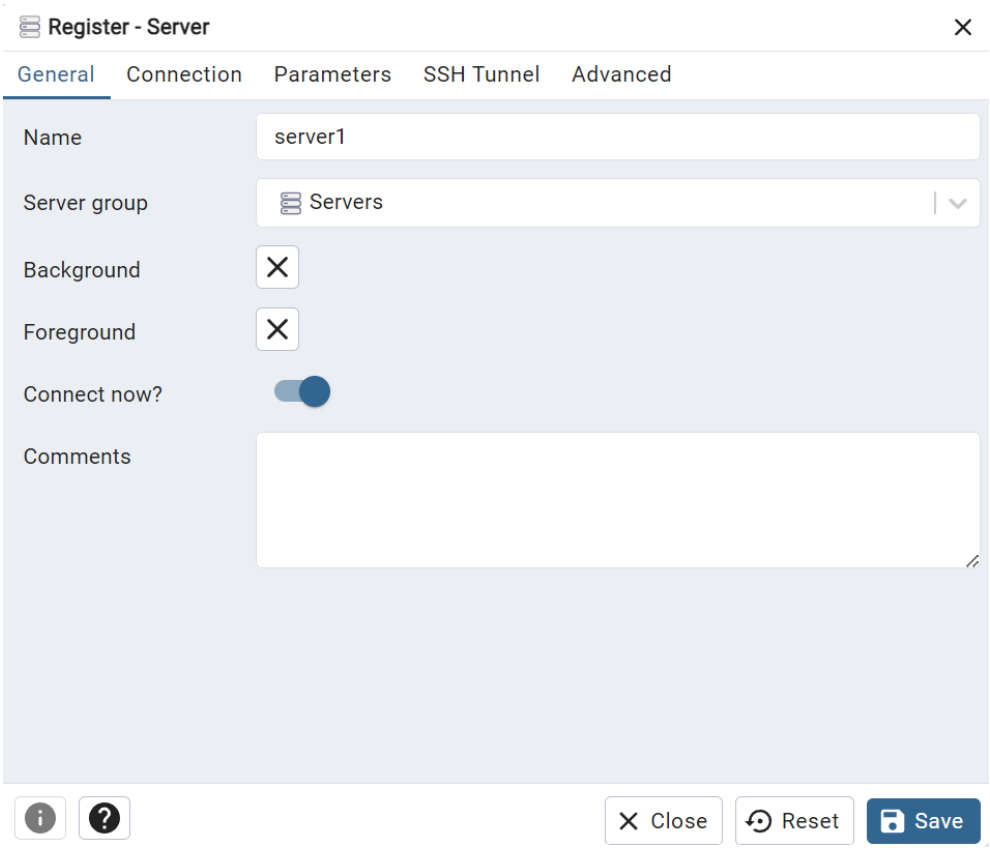


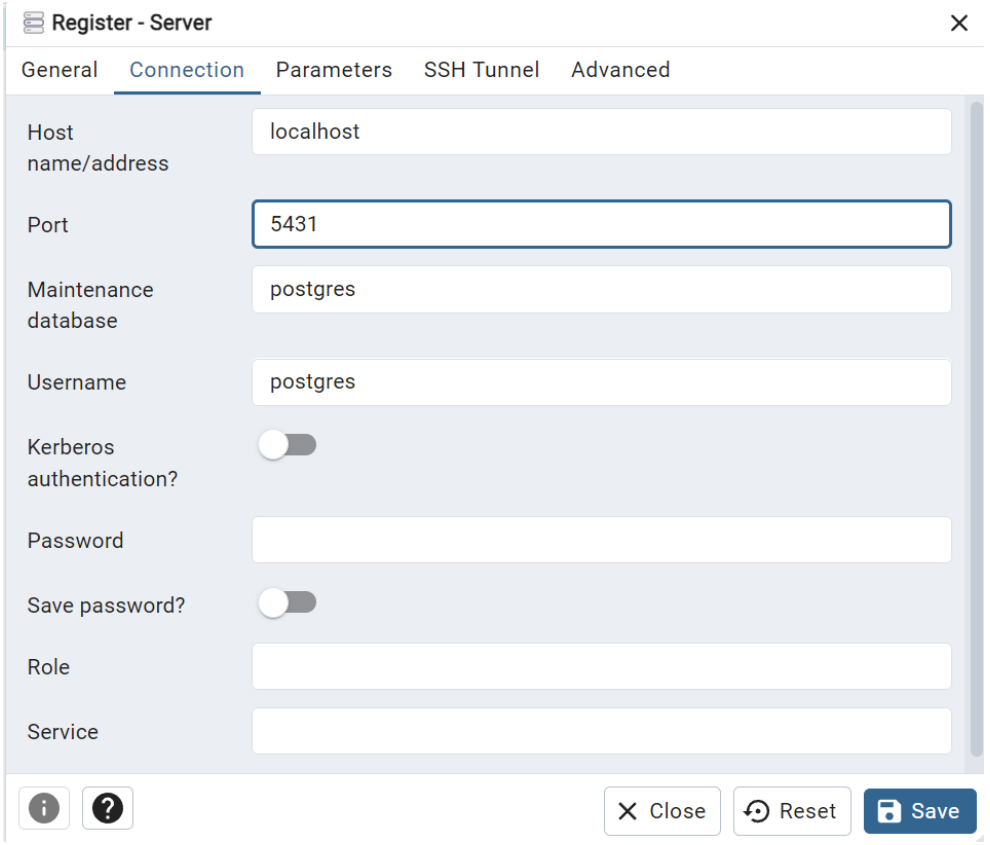
15. Menjalankan image

```
C:\Users\Dailatul Arofah>docker run -it ubuntu
root@ce38e833419b:/# ls
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
root@ce38e833419b:/# pwd
/
root@ce38e833419b:/# cd
root@ce38e833419b:~# pwd
/root
root@ce38e833419b:~#
```

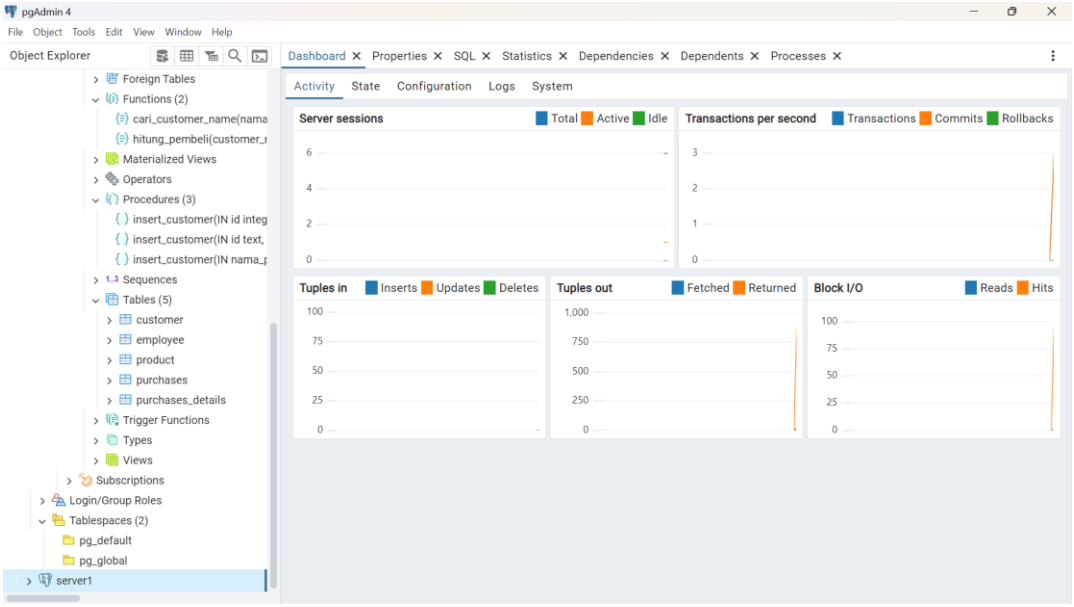
16. Masuk ke pgAdmin – server – dashboard – add new server



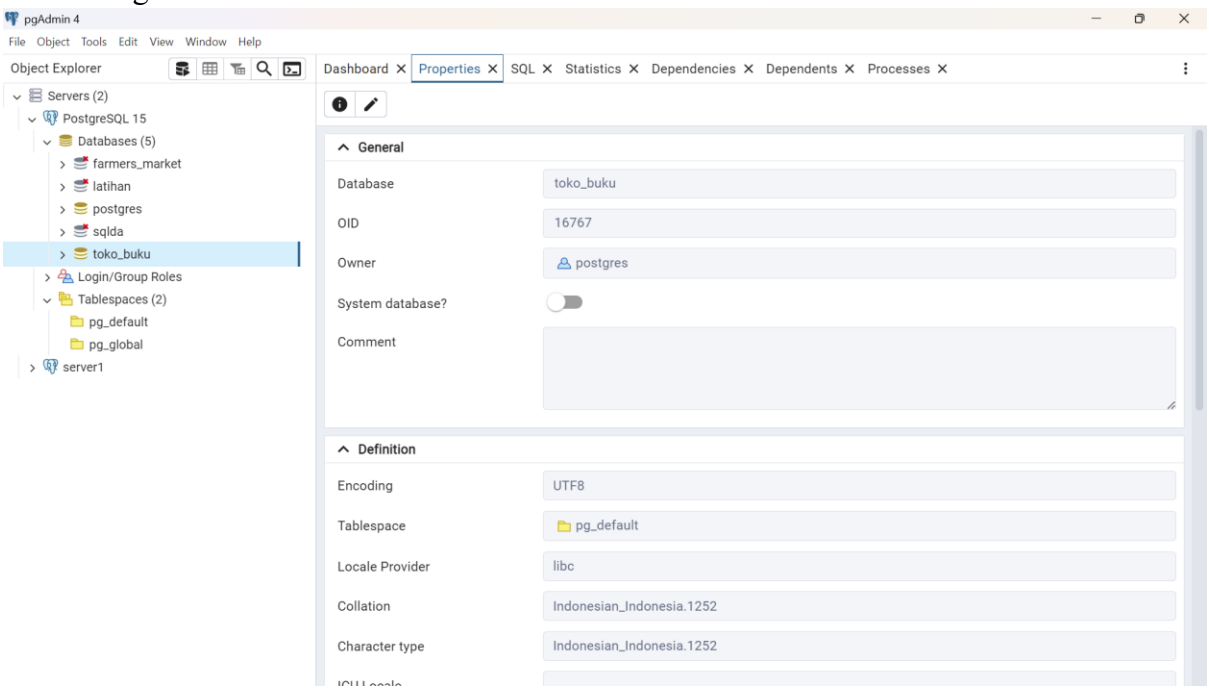

Register - Server
General
Connection
Parameters
SSH Tunnel
Advanced
Name
server1
Server group
Servers
Background
Foreground
Connect now?
Comments
Close
Reset
Save

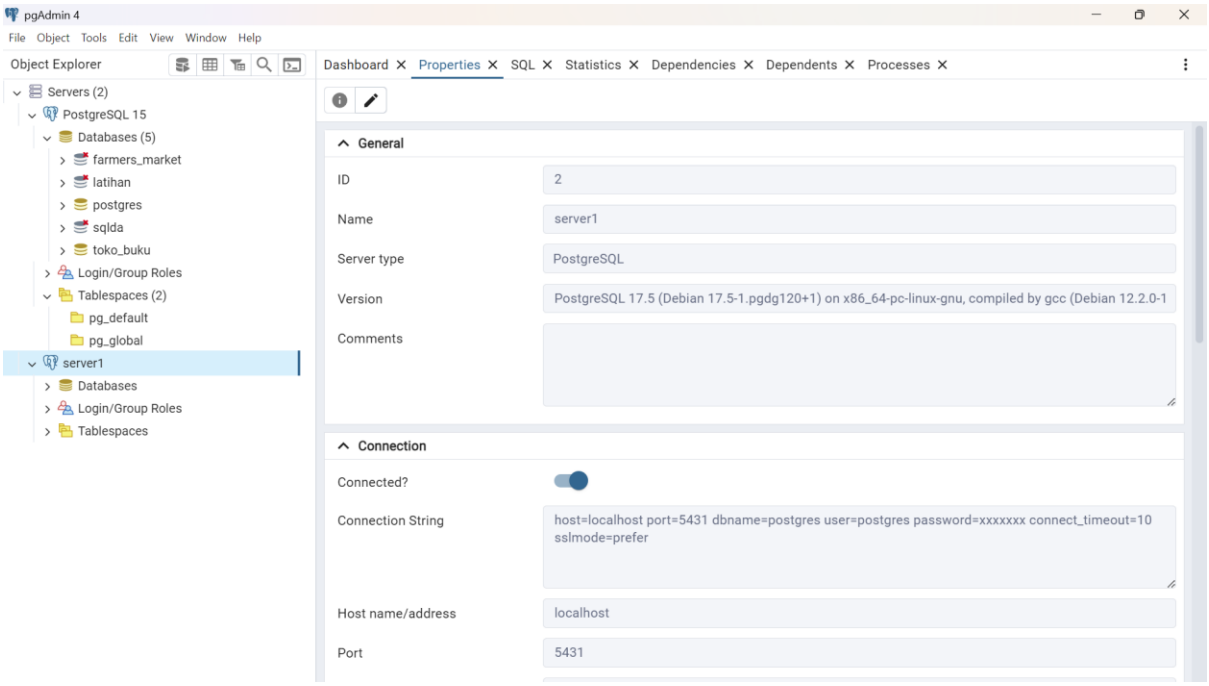

Register - Server
General
Connection
Parameters
SSH Tunnel
Advanced
Host name/address
localhost
Port
5431
Maintenance database
postgres
Username
postgres
Kerberos authentication?
Password
Save password?
Role
Service
Close
Reset
Save

17. Hasil dari server1



18. Perbandingan





19. Install httpd

```
C:\Users\Dailatul Arofah>docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
d0a755bf09a1: Pull complete
d1042d58e186: Pull complete
4f4fb700ef54: Pull complete
be5c5a616c3a: Pull complete
c06cec1379c2: Pull complete
Digest: sha256:f6557a77ee2f16c50a5ccbb2564a3fd56087da311bf69a160
d43f73b23d3af2d
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
```

20. Run httpd

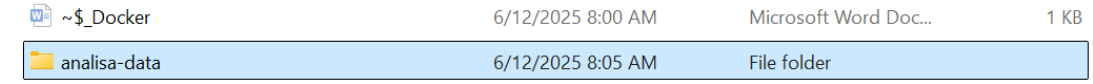
```
C:\Users\Dailatul Arofah>docker run -d -p 80:80 --name my-apache httpd
cb19dca9170f55cd4420eab76b438cb3c2afc80141f58e9022a161ed019169fc
```

21. Mengecek image

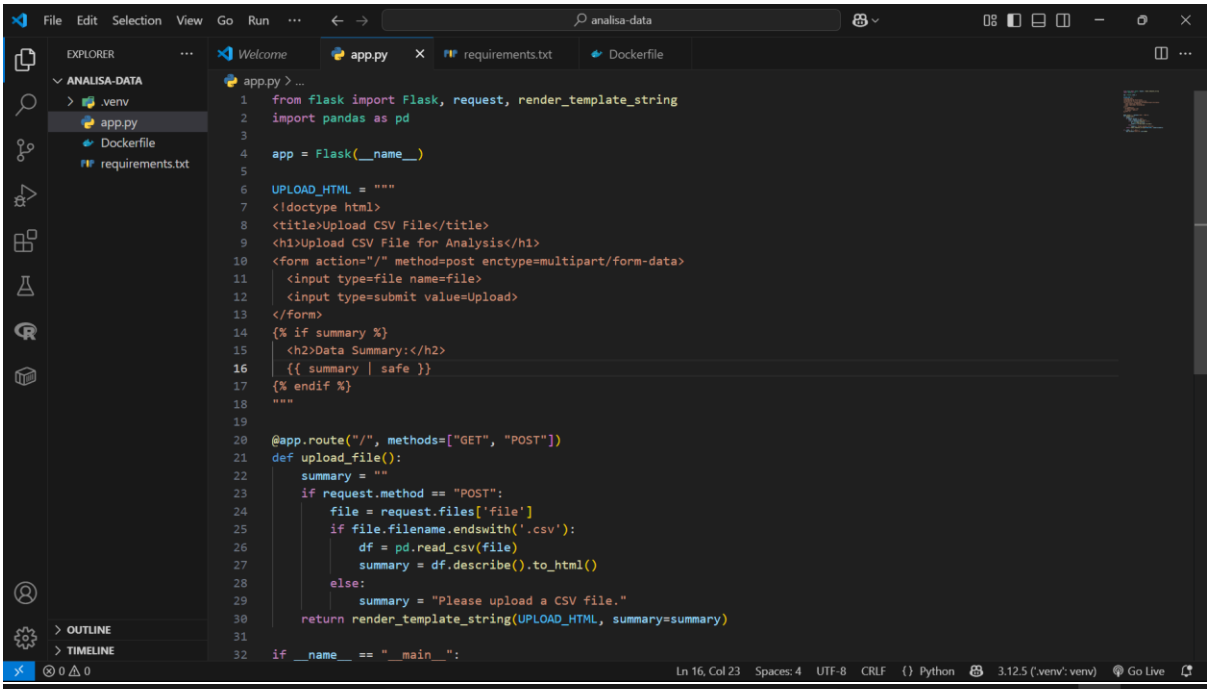
```
C:\Users\Dailatul Arofah>docker image ls
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
postgres     latest    6cf6142afacf   4 days ago     621MB
ubuntu       latest    b59d21599a2b   13 days ago    117MB
mysql        latest    04768cb63395   8 weeks ago    1.17GB
httpd        latest    f6557a77ee2f   4 months ago   221MB
```

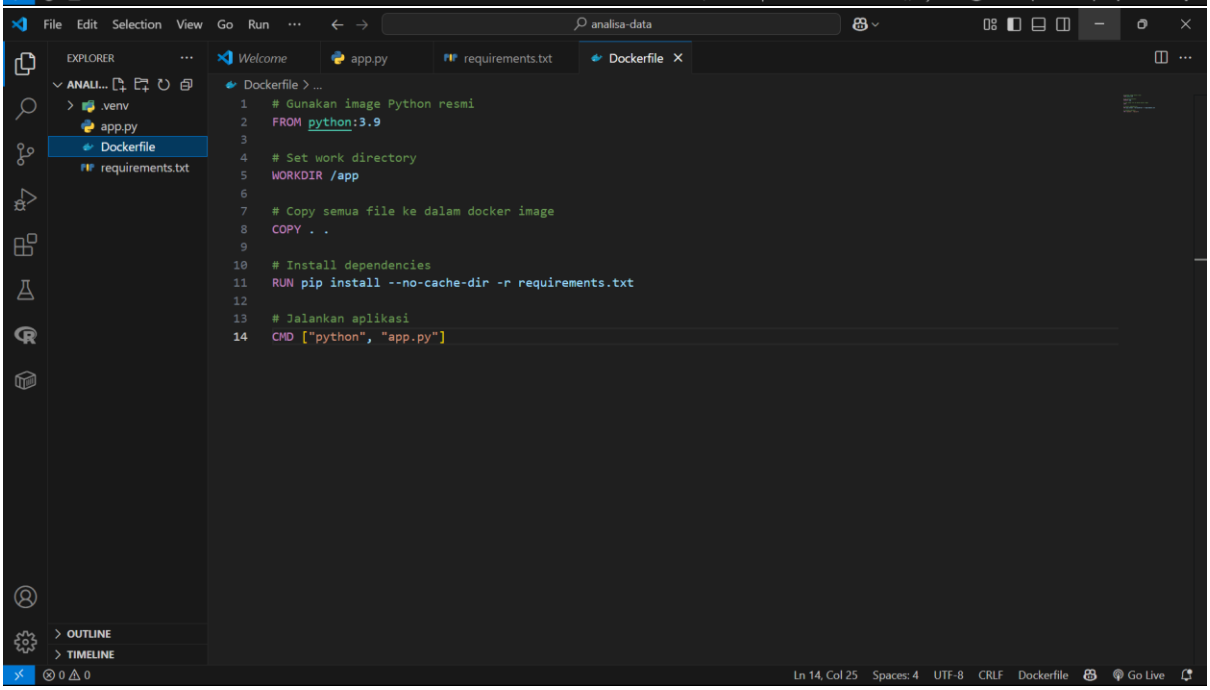
MEMBUAT APLIKASI ANALISA DATA

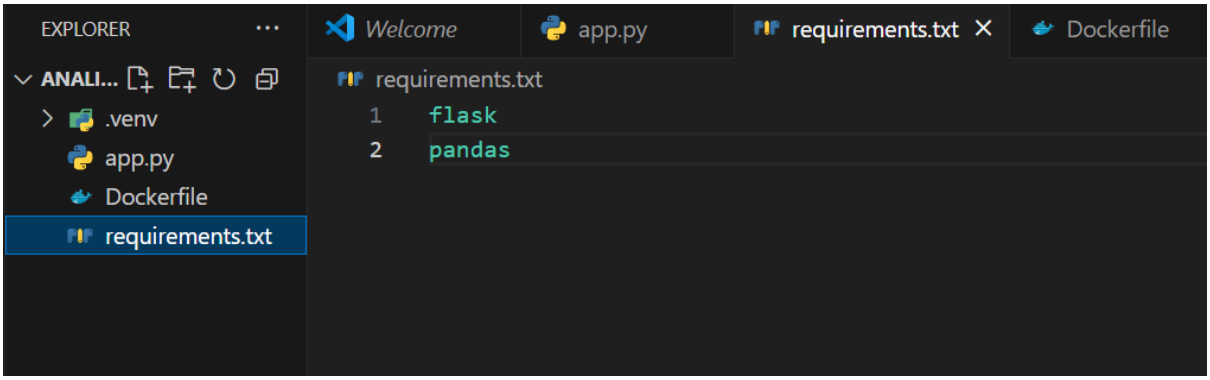
1. Membuat folder ‘analisis-data’



2. Membuat 3 file, yaitu .py, .txt, dan DOckerfile di Virtual Studio Code yang nantinya akan berada di dalam folder ‘analisis-data’.







3. Buka CMD dan masuk ke direktori folder ‘analisa-data’

```
C:\Users\Dailatul Arofah>cd "C:\Users\Dailatul Arofah\OneDrive\D
ocuments\analisa-data"
```

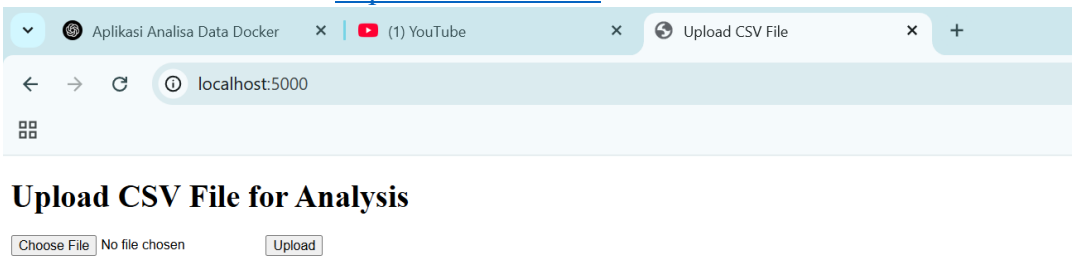
4. Build Docker Image

```
C:\Users\Dailatul Arofah\OneDrive\Documents\analisa-data>docker build -t analisa-data-app .
[+] Building 8.1s (4/9)                                                    docker:desktop-linux
 => [internal] load build definition from Dockerfile                                       0.1s
 => => transferring dockerfile: 302B                                                       0.0s
 => [internal] load metadata for docker.io/library/python:3.9                              4.1s
 => [auth] library/python:pull token for registry-1.docker.io                              0.0s
 => [internal] load .dockerignore                                                          0.1s
 => => transferring context: 2B                                                            0.0s
 => [1/4] FROM docker.io/library/python:3.9@sha256:99bf395a7271e373937b9f4d30f1562a15f6e  3.8s
 => => resolve docker.io/library/python:3.9@sha256:99bf395a7271e373937b9f4d30f1562a15f6e  0.0s
 => => sha256:f57f9892d27393f8c8cea049f776d6cd4eb831a8be16b3e89db6ef64efb93fdb 251B / 251B  0.4s
 => => sha256:a7d2ea3f33a8c566c6e27c0e123d043ab74d29b26c2b22f01fc65131ccb90bf8 5.24MB / 6.16MB  3.2s
 => => sha256:b6754ffd126059cf8be2be19089f13392b63ecd646f6dfb411c015fce35a15d2 1.05MB / 19.85MB  3.1s
 => => sha256:48b8862a18fa961ebfbac8484877dd4894e96ee88177d8c4f1f54d9727262b7d 1.05MB / 211.37MB  3.1s
 => => sha256:b1b8a0660a31403a35d70b276c3c86b1200b8683e83cd77a92ec98744017684a 1.05MB / 64.40MB  2.8s
 => [internal] load build context                                                          3.8s
 => => transferring context: 5.41MB                                                        3.7s
```

5. Menjalankan Docker Container

```
C:\Users\Dailatul Arofah\OneDrive\Documents\analisa-data>docker run -d -p 5000:5000 analisa-data-app
f8339897b29ae66098f3774ef52bb0d5161ef0b8ddb41f5354936961d553ec6d
```

6. Buka browser dan ketik ‘ http://localhost:5000 ‘



7. Masukkan file CSV untuk dianalisis

Upload CSV File for Analysis

Choose File No file chosen Upload

Data Summary:

	New Cases	New Deaths	New Recovered	New Active Cases	Total Cases	Total Deaths	Total Recovered	Total Active Cases	City or Regency	Total Regencies	Total Cities	Total Districts	Total Urban Villages	Total Rural Villages	Area (km2)	Population
count	20816.000000	20816.000000	20816.000000	20816.000000	2.081600e+04	20816.000000	2.081600e+04	20816.000000	0.0	20816.000000	20228.000000	20816.000000	20226.000000	20201.000000	2.081600e+04	2.081600e+04
mean	408.078257	13.784829	393.044341	1.249087	7.788760e+04	2388.498463	6.935489e+04	6144.216132	NaN	24.155073	5.872306	420.381197	508.692277	4489.751101	1.112142e+05	1.547845e+07
std	2119.450324	78.146224	2042.353301	958.146043	3.417859e+05	10758.160910	3.124817e+05	28108.333301	NaN	68.699338	16.469371	1198.862430	1428.926046	12646.177100	3.203338e+05	4.483034e+07
min	0.000000	0.000000	0.000000	-25725.000000	1.000000e+00	0.000000	0.000000e+00	-2305.000000	NaN	1.000000	1.000000	44.000000	35.000000	275.000000	6.640000e+02	6.484070e+05
25%	8.000000	0.000000	4.000000	-17.000000	1.579500e+03	43.000000	8.730000e+02	217.000000	NaN	7.000000	1.000000	103.000000	99.000000	928.000000	1.678700e+04	1.999539e+06
50%	45.000000	1.000000	35.000000	1.000000	9.827000e+03	253.000000	7.739000e+03	1002.000000	NaN	11.000000	2.000000	169.000000	175.000000	1591.000000	4.201300e+04	4.216171e+06
75%	161.000000	5.000000	152.000000	29.000000	3.363625e+04	949.000000	2.918100e+04	2748.250000	NaN	18.000000	4.000000	289.000000	332.000000	2853.000000	7.546800e+04	9.095591e+06
max	56757.000000	2069.000000	48832.000000	36726.000000	4.247320e+06	143519.000000	4.092586e+06	574135.000000	NaN	416.000000	98.000000	7230.000000	8488.000000	74953.000000	1.916907e+06	2.651855e+08