

LAPORAN PRAKTIKUM

QUIZ & DOCKER

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



Disusun untuk Memenuhi Tugas

Mata Kuliah:

MANAJEMEN DATA

Oleh:

Nadia Fitriana Mudlikah

NRP. 3324600036

PROGRAM STUDI D4 SAINS DATA TERAPAN

DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER

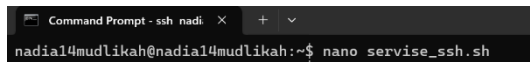
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

QUIZ

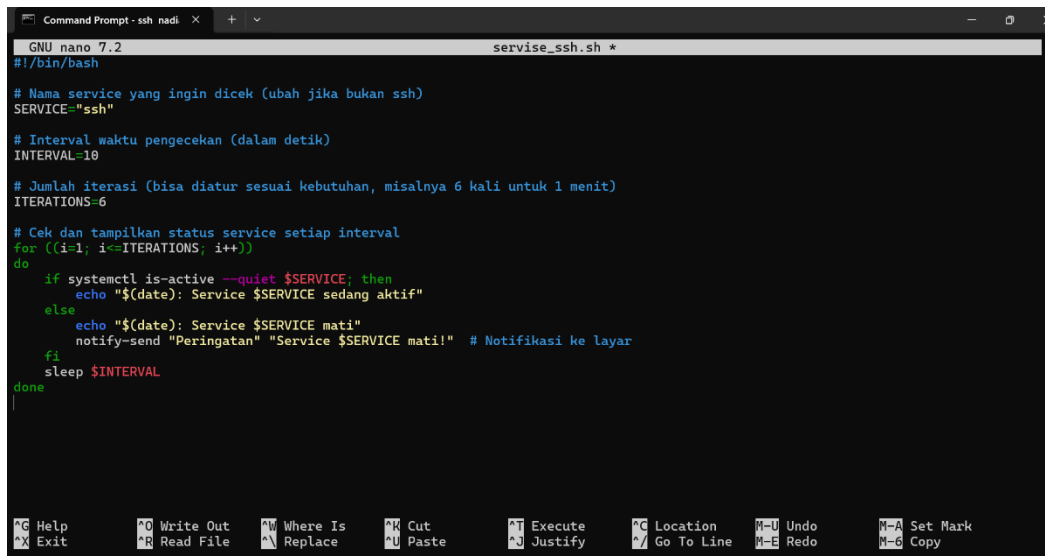
SOAL 1

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

SCRIPT & ANALISIS



Perintah nano berfungsi untuk membuka dan mengedit file teks servise_ssh.sh langsung dari terminal.



```
GNU nano 7.2 servise_ssh.sh *
#!/bin/bash

# Nama service yang ingin dicek (ubah jika bukan ssh)
SERVICE="ssh"

# Interval waktu pengecekan (dalam detik)
INTERVAL=10

# Jumlah iterasi (bisa diatur sesuai kebutuhan, misalnya 6 kali untuk 1 menit)
ITERATIONS=6

# Cek dan tampilkan status service setiap interval
for ((i=1; i<=ITERATIONS; i++))
do
    if systemctl is-active --quiet $SERVICE; then
        echo "$(date): Service $SERVICE sedang aktif"
    else
        echo "$(date): Service $SERVICE mati"
        notify-send "Peringatan" "Service $SERVICE mati!" # Notifikasi ke layar
    fi
    sleep $INTERVAL
done
```

Script servise_ssh.sh digunakan untuk memantau apakah layanan SSH dalam keadaan aktif atau mati dalam periode waktu tertentu. Script ini memakai tiga variabel utama, yaitu SERVICE="ssh" untuk menentukan nama layanan yang ingin dicek, INTERVAL=10 sebagai jeda waktu antar pengecekan (dalam detik), dan ITERATIONS=6 sebagai jumlah pengulangan pemeriksaan. Perulangan dilakukan dengan perintah for, dan status layanan diperiksa menggunakan systemctl is-active --quiet. Jika layanan aktif, perintah echo akan menampilkan waktu dan status layanan. Jika layanan mati, akan muncul pesan peringatan

melalui echo, dan notifikasi ke layar dikirim menggunakan notify-send. Perintah sleep digunakan untuk memberi jeda sebelum pemeriksaan berikutnya.

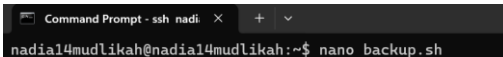
```
nadial4mudlikah@nadial4mudlikah:~$ chmod +x servise_ssh.sh
nadial4mudlikah@nadial4mudlikah:~$ ./servise_ssh.sh
Wed May 28 12:17:03 PM UTC 2025: Service ssh sedang aktif
Wed May 28 12:17:16 PM UTC 2025: Service ssh sedang aktif
Wed May 28 12:17:26 PM UTC 2025: Service ssh sedang aktif
^C
```

Untuk menjalankan skrip, perlu diberikan izin akses eksekusi dengan `chmod +x servise_ssh.sh`, lalu dijalankan menggunakan `./servise_ssh.sh`.

SOAL 2

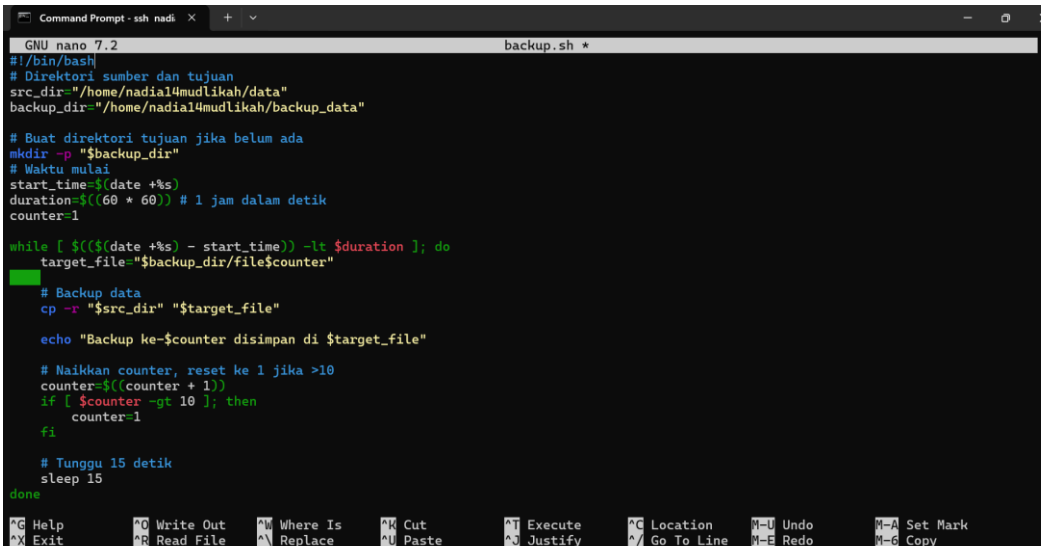
Buat script backup direktori tertentu dan jalankan backup secara berkala setiap 15 detik dengan backup ke file1 sd 10 dan kembali lagi menimpa file1 jika sudah file10 jalankan selama 1 jam. Jalankan backup tersebut secara otomatis dengan crontab pada setiap hari minggu jam 1 malam.

SCRIPT & ANALISIS



```
Command Prompt - ssh nadi
nadial4mudlikah@nadial4mudlikah:~$ nano backup.sh
```

Digunakan untuk membuka dan mengedit file skrip bernama `backup.sh` menggunakan editor teks Nano.



```
GNU nano 7.2 backup.sh *
#!/bin/bash
# Direktori sumber dan tujuan
src_dir="/home/nadial4mudlikah/data"
backup_dir="/home/nadial4mudlikah/backup_data"

# Buat direktori tujuan jika belum ada
mkdir -p "$backup_dir"
# Waktu mulai
start_time=$(date +%s)
duration=$((60 * 60)) # 1 jam dalam detik
counter=1

while [ $((date +%s) - start_time) -lt $duration ]; do
    target_file="$backup_dir/file$counter"

    # Backup data
    cp -r "$src_dir" "$target_file"

    echo "Backup ke-$counter disimpan di $target_file"

    # Naikkan counter, reset ke 1 jika >10
    counter=$((counter + 1))
    if [ $counter -gt 10 ]; then
        counter=1
    fi

    # Tunggu 15 detik
    sleep 15
done
```

Skrip `backup.sh` ini berfungsi untuk melakukan pencadangan otomatis dari folder sumber ke folder tujuan selama satu jam. Pada awal skrip, ditentukan terlebih

dahulu folder sumber data dan folder tujuan untuk hasil pencadangan. Jika folder tujuan belum ada, maka folder tersebut akan dibuat terlebih dahulu menggunakan perintah `mkdir -p`. Skrip ini juga mencatat waktu mulai dan mengatur durasi pencadangan menjadi satu jam. Selain itu, terdapat counter untuk memberikan nomor urut pada setiap hasil pencadangan agar tidak saling menimpa. Proses pencadangan akan terus berjalan selama durasi yang ditentukan, yang akan dicek menggunakan perulangan `while`. Di dalam perulangan, folder sumber akan disalin ke folder tujuan dengan nama file yang disesuaikan dengan nomor urut. Setelah setiap pencadangan selesai, akan muncul pesan yang menandakan bahwa pencadangan berhasil. Jika counter sudah mencapai angka 10, maka counter akan kembali ke angka 1. Skrip akan menunggu selama 15 detik sebelum melanjutkan pencadangan berikutnya.

```
nadia14mudlikah@nadia14mudlikah:~$ chmod +x backup.sh
```

Memberi izin agar file `backup.sh` bisa dieksekusi sebagai program.

```
Command Prompt - ssh nadi X + v
nadia14mudlikah@nadia14mudlikah:~$ chmod +x backup.sh
nadia14mudlikah@nadia14mudlikah:~$ ./backup.sh
cp: cannot stat '/home/nadia14mudlikah/data': No such file or directory
Backup ke-1 disimpan di /home/nadia14mudlikah/backup_data/file1
cp: cannot stat '/home/nadia14mudlikah/data': No such file or directory
Backup ke-2 disimpan di /home/nadia14mudlikah/backup_data/file2
cp: cannot stat '/home/nadia14mudlikah/data': No such file or directory
Backup ke-3 disimpan di /home/nadia14mudlikah/backup_data/file3
^C
```

Menjalankan skrip `backup.sh`.

```
nadia14mudlikah@nadia14mudlikah:~$ mkdir -p /home/nadia14mudlikah/data
```

Karena file direktori `/home/nadia14mudlikah/data` belum tersedia, maka dibuat terlebih dahulu menggunakan perintah `mkdir -p /home/nadia14mudlikah/data`

```
nadia14mudlikah@nadia14mudlikah:~$ ./backup.sh
Backup ke-1 disimpan di /home/nadia14mudlikah/backup_data/file1
Backup ke-2 disimpan di /home/nadia14mudlikah/backup_data/file2
Backup ke-3 disimpan di /home/nadia14mudlikah/backup_data/file3
Backup ke-4 disimpan di /home/nadia14mudlikah/backup_data/file4
Backup ke-5 disimpan di /home/nadia14mudlikah/backup_data/file5
Backup ke-6 disimpan di /home/nadia14mudlikah/backup_data/file6
Backup ke-7 disimpan di /home/nadia14mudlikah/backup_data/file7
Backup ke-8 disimpan di /home/nadia14mudlikah/backup_data/file8
Backup ke-9 disimpan di /home/nadia14mudlikah/backup_data/file9
Backup ke-10 disimpan di /home/nadia14mudlikah/backup_data/file10
Backup ke-1 disimpan di /home/nadia14mudlikah/backup_data/file1
Backup ke-2 disimpan di /home/nadia14mudlikah/backup_data/file2
```

Menjalankan ulang skrip `backup.sh`.

```
nadia14mudlikah@nadia14mudlikah:~$ crontab -e
crontab: installing new crontab
```

Membuka file konfigurasi `cron` untuk menjadwalkan tugas otomatis.

```
Command Prompt - ssh nadi x + v
GNU nano 7.2 /tmp/crontab.JEF8yT/crontab *
#DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.21968 installed in Fri Jan 24 11:30:41 2003)
# (Cron version -- $Id: chap8.xml,v 1.9 2006/09/28 09:42:45 tille Exp $)
32 11 * * * ls -l /proc/self/fd/ > /var/tmp/fdtest.cron
0 1 * * 0 /home/nadial4mudlikah/backup.sh
```

Menjadwalkan agar skrip backup.sh dijalankan otomatis setiap hari Minggu pukul 01:00.

SOAL 3

Buat file data.txt dengan isian sbb : 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

- Buat script awk yang membaca file tersebut dan menghasilkan output sbb : Suhu maksimum: 27.1
- Buat script awk membaca file tersebut dan menghitung rata2 kelembaban dengan output sbb : Rata-rata kelembaban: 63.5

SCRIPT & ANALISIS

```
Command Prompt - ssh nadi x + v
nadial4mudlikah@nadial4mudlikah:~$ nano data.txt
```

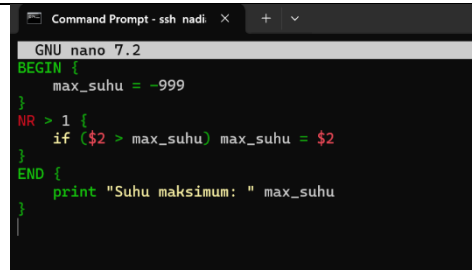
Membuka dan mengedit file data.txt menggunakan editor teks nano di terminal.

```
Command Prompt - ssh nadi x + v
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
```

File berisi data dengan kolom-kolom timestamp, suhu, kelembaban, dan cahaya yang mencatat nilai suhu pada waktu tertentu.

```
nadial4mudlikah@nadial4mudlikah:~$ nano max_suhu.awk
```

Membuka atau membuat skrip AWK di file max_suhu.awk. Skrip ini bertujuan untuk mencari suhu maksimum dari data yang ada pada file data.txt.



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

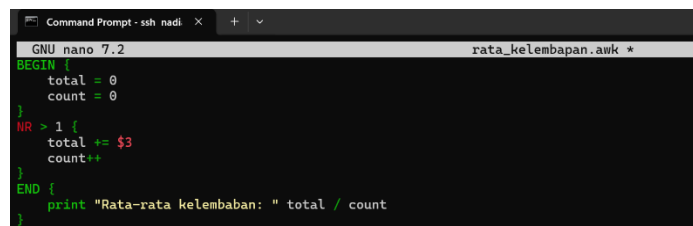
BEGIN { max_suhu = -999 }: Bagian ini inisialisasi variabel max_suhu dengan nilai awal yang sangat kecil (-999) untuk memastikan bahwa nilai suhu pertama akan lebih besar dari nilai ini.

NR > 1 { if (\$2 > max_suhu) max_suhu = \$2 }: Bagian ini mulai memeriksa data setelah baris pertama (yang berisi header). Untuk setiap baris data, jika nilai suhu (kolom kedua, \$2) lebih besar dari max_suhu, maka max_suhu diupdate dengan nilai suhu yang lebih besar tersebut.

END { print "Suhu maksimum: " max_suhu }: Setelah semua baris diproses, bagian ini mencetak suhu maksimum yang ditemukan dalam file.

```
nadial4mudlikah@nadial4mudlikah:~$ nano rata_kelembapan.awk
```

Membuka atau membuat skrip AWK di file rata_kelembapan.awk. Skrip ini bertujuan untuk mencari rata-rata kelembaban dari data yang ada pada file data.txt.



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

Skrip rata_kelembapan.awk digunakan untuk menghitung rata-rata kelembaban dari file data yang memiliki kolom-kolom seperti timestamp, suhu, kelembaban, dan cahaya. Bagian BEGIN menginisialisasi variabel total dan count ke nol. Baris data dimulai dari baris kedua (menghindari header), lalu nilai kelembaban di kolom ketiga ditambahkan ke total dan count bertambah satu untuk setiap baris. Setelah seluruh baris diproses, bagian END mencetak hasil pembagian total kelembaban dengan jumlah data, yaitu rata-rata kelembaban.

```
nadia14mudlikah@nadia14mudlikah:~$ awk -f max_suhu.awk data.txt  
Suhu maksimum: 27.1
```

Menjalankan skrip max_suhu.awk pada file data.txt. Output yang dihasilkan adalah suhu maksimum yang tercatat dalam data tersebut. Pada kasus ini, suhu maksimum yang ditemukan adalah 27.1.

```
nadia14mudlikah@nadia14mudlikah:~$ awk -f rata_kelembapan.awk data.txt  
Rata-rata kelembapan: 63.5
```

Menjalankan skrip data_kelembapan.awk pada file data.txt. Output yang dihasilkan adalah rata-rata kelembapan yang tercatat dalam data tersebut. Pada kasus ini, suhu maksimum yang ditemukan adalah 27.1.

SOAL 4

Buatkan perintah dilinux untuk melihat file /etc/passwd, missal isinya sebagai berikut:

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

Lakukan filter lagi sbb menjadi sbb : Tunjukkan step by step scriptnya dan hasilnya.

SCRIPT & ANALISIS

```
nadia14mudlikah@nadia14mudlikah:~$ cat /etc/passwd
```

```

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:Mailng 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
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon,,,:/usr/lib/dhcpcd:/bin/false
messagebus:x:101:102::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:992:992:systemd Resolver:/:/usr/sbin/nologin
pollinate:x:102:1:/:var/cache/pollinate:/bin/false
polkitd:x:991:991:User for polkitd:/:/usr/sbin/nologin
syslog:x:103:104::/nonexistent:/usr/sbin/nologin
uuuid:x:104:105::run/uuid:/usr/sbin/nologin
tcpdump:x:105:107:/nonexistent:/usr/sbin/nologin
tss:x:106:108:TPM software stack,,,:/var/lib/tpm:/bin/false
landscape:x:107:109:/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:989:989:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin
usbmux:x:108:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
nadia14mudlikah:x:1000:1000:nadia14mudlikah,36,081252440722,-:/home/nadia14mudlikah:/bin/bash
diane:x:1004:100::/home/diane:/bin/sh
sshd:x:109:65534::run/sshd:/usr/sbin/nologin
nadia1:x:1005:1005:/home/nadia1:/bin/sh
nadia2:x:1006:1006:/home/nadia2:/bin/sh
nadia3:x:1007:1007:/home/nadia3:/bin/sh
nadia14mudlikah@nadia14mudlikah:~$

```

cat /etc/passwd digunakan untuk menampilkan isi file sistem yang berisi informasi user di sistem Linux, seperti username, UID, GID, dan direktori home.

```
nadia14mudlikah@nadia14mudlikah:~$ nano filter.sh
```

```

GNU nano 7.2 filter.sh *
#!/bin/bash

# Menampilkan user yang punya direktori home dan mengambil nama usernya
awk -F: '$6 ~ /^\/home\/' { print $1 } /etc/passwd

```

Perintah nano berfungsi untuk membuka dan mengedit file teks filter.sh langsung dari terminal. Isi skrip filter.sh menggunakan AWK untuk menampilkan hanya user yang direktori home-nya berada di /home. AWK membaca file /etc/passwd, memisahkannya berdasarkan tanda : (-F:), lalu mencocokkan field ke-6 (\$6) dengan pola /^\/home\/. Jika cocok, maka mencetak field pertama (\$1), yaitu nama user.

```
nadia14mudlikah@nadia14mudlikah:~$ chmod +x filter.sh
```

Perintah chmod +x filter.sh digunakan untuk memberi hak akses eksekusi ke file filter.sh, agar bisa dijalankan langsung sebagai skrip.

```

nadia14mudlikah@nadia14mudlikah:~$ ./filter.sh
nadia14mudlikah
diane
nadia1
nadia2
nadia3

```

Perintah ./filter.sh menjalankan skrip tersebut, dan hasilnya menampilkan daftar user yang memiliki direktori home di /home, yang berarti mereka adalah user biasa yang pernah dibuat di sistem.



DOCKER

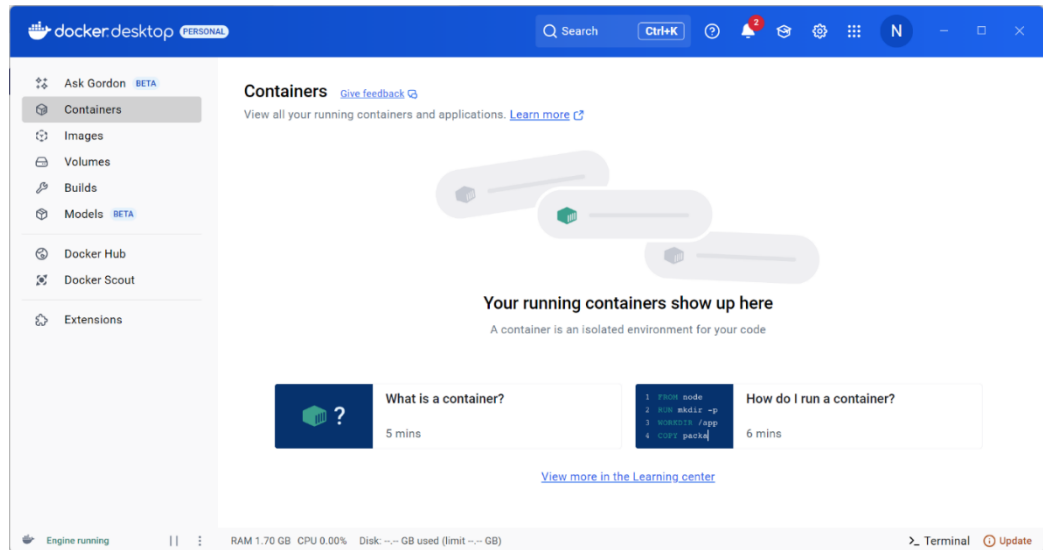
Docker adalah platform open-source yang digunakan untuk mengemas, mendistribusikan, dan menjalankan aplikasi di dalam wadah (container). Adapun beberapa istilah yang ada dalam docker:

- **WSL (Windows Subsystem for Linux)** adalah fitur di Windows yang memungkinkan pengguna menjalankan sistem Linux langsung di dalam Windows tanpa perlu dual-boot.
- **Image** adalah cetakan atau paket aplikasi lengkap berisi kode, pustaka, dan konfigurasi yang dibutuhkan untuk menjalankan aplikasi dalam Docker.
- **Container** adalah hasil dari menjalankan image, yaitu aplikasi yang berjalan dalam lingkungan terisolasi dan ringan.

Berikut Langkah-langkah install docker:

Install Docker Windows

- Download Docker
 - [https:// docs.docker.com/desktop/setup/install/windows-install/](https://docs.docker.com/desktop/setup/install/windows-install/)
- Install.
- Restart Windows.
- Login menggunakan email.



Install wsl

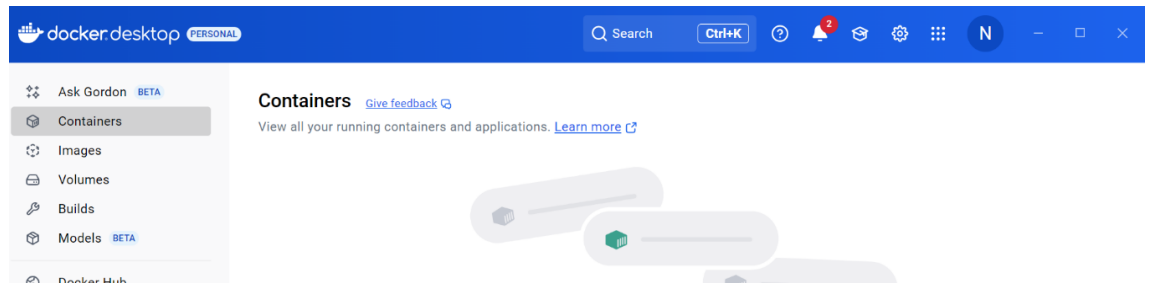
```
PS C:\WINDOWS\system32> wsl --install
Downloading: Ubuntu
Launching Ubuntu...
Provisioning the new WSL instance Ubuntu
This might take a while...
Create a default Unix user account: nadia14mudlikah
New password:
Retype new password:
passwd: password updated successfully
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

nadia14mudlikah@DESKTOP-PFMT2J7: /mnt/c/WINDOWS/system32$
```

Kemungkinan Error

- Jika ada error dicoba restart lagi komputernya dan pastikan docker jalan
- Pastikan sudah running sebagai berikut:

```
PS C:\WINDOWS\system32> docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
```

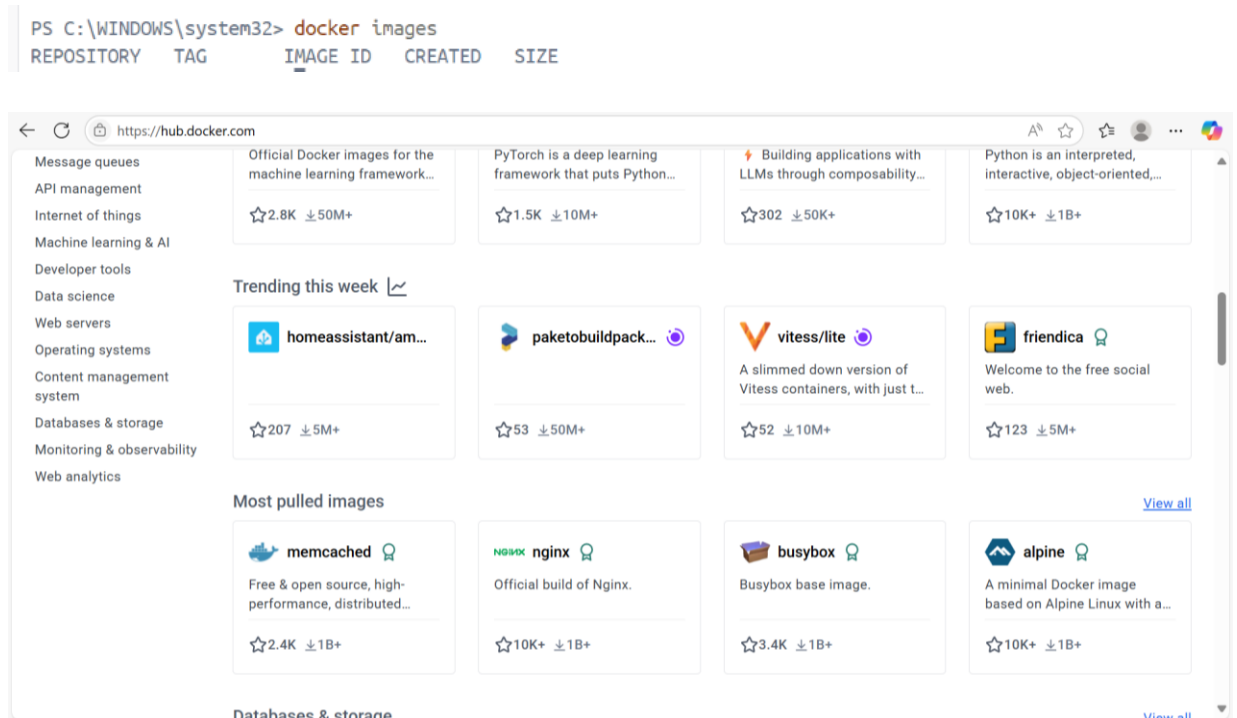


Docker Images

- Bisa dibikin, bisa download images yang sudah ada

- Images yang sudah ready ada di docker hub

<https://hub.docker.com/>



Download Images

- Misal postgres
- Jalankan perintah docker pull postgres

```
PS C:\WINDOWS\system32> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE

PS C:\WINDOWS\system32> docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
f4ce9941f6e3: Pulling fs layer
dad67da3f26b: Pulling fs layer
eb3a531023c8: Pulling fs layer
05b641b3bdab: Pulling fs layer
05b641b3bdab: Downloading 1.049MB/4.534MB
603ef9fcd8e: Downloading 4.194MB/8.066MB
8a1f652e0c97: Download complete
c6def2c6e21d: Download complete
```

- Cek : docker images

```
PS C:\WINDOWS\system32> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
postgres latest 6efd0df010dc 2 weeks ago 621MB
```



- Jalankan perintah pull mysql

```
PS C:\WINDOWS\system32> docker --version
Docker version 28.1.1, build 4eba377
```

```
PS C:\WINDOWS\system32> docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
5056ce4ab875: Pull complete
9845df06f911: Pull complete
72ac912b8a2e: Pull complete
4bd1fb59dd90: Pull complete
d23320eed97a: Pull complete
7074f55c9a02: Pull complete
b288ccce2510: Pull complete
b097427f1ebe: Pull complete
7488ffd7127f: Pull complete
8a50ff4ab30c: Pull complete
```

- Jalankan perintah pull ubuntu

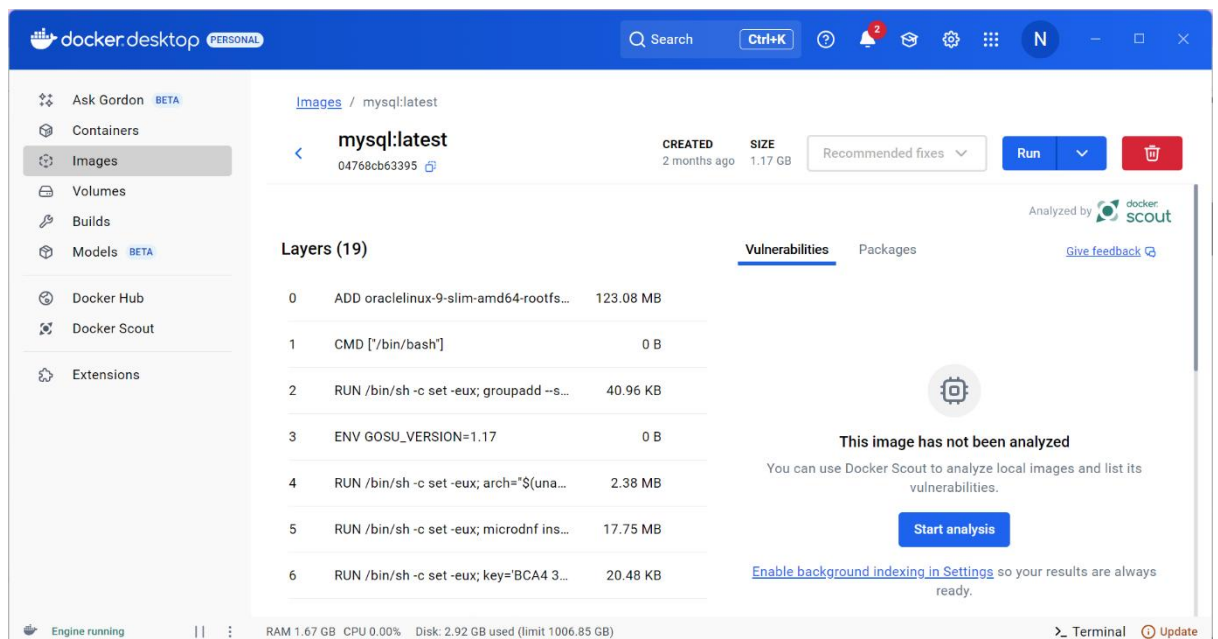
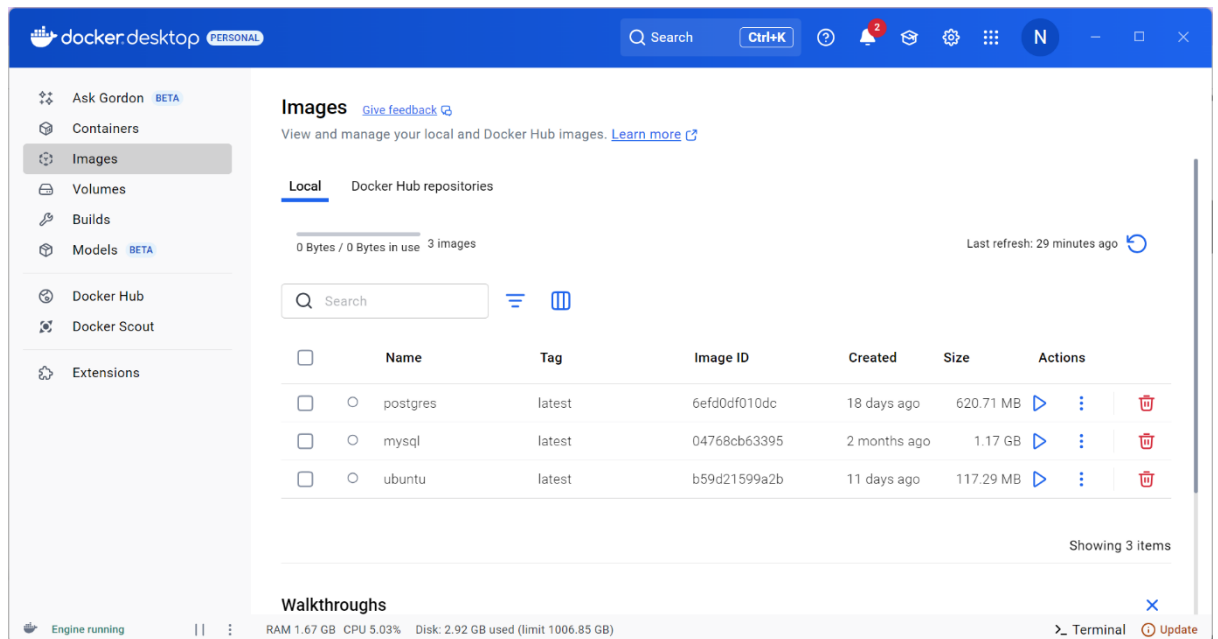
```
PS C:\WINDOWS\system32> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
d9d352c11bbd: Pull complete
Digest: sha256:b59d21599a2b151e23eea5f6602f4af4d7d31c4e236d22bf0b62b86d2e386b8f
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

List Image

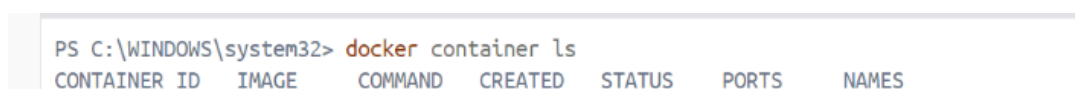
- Docker image ls

```
PS C:\WINDOWS\system32> docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    b59d21599a2b   10 days ago    117MB
postgres      latest    6efd0df010dc   2 weeks ago    621MB
mysql         latest    04768cb63395   7 weeks ago    1.17GB
```

List Image di Docker Desktop

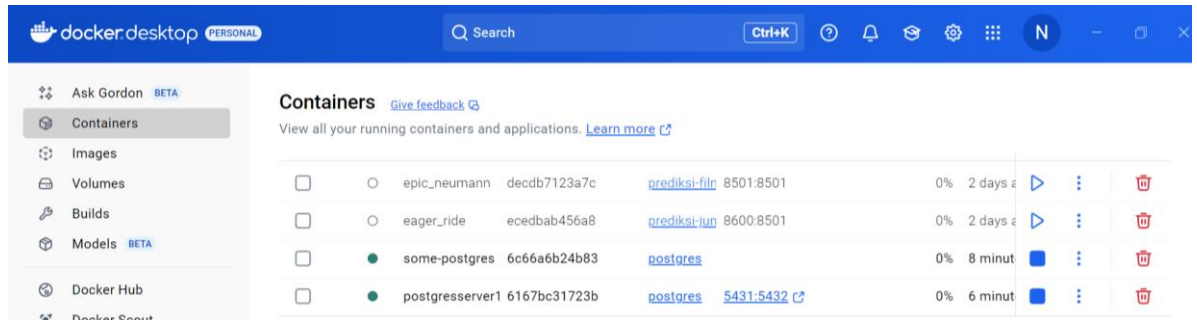


Container



Membuat dan Menjalankan Container

```
PS C:\WINDOWS\system32> docker run --name some-postgres -e POSTGRES_PASSWORD=123456 -d postgres
6c66a6b24b83b887ea8853d0ade2397b9413e42a57e221fa564ca21609e0beeb
PS C:\WINDOWS\system32> docker run --name postgresserver1 -p 5431:5432 -e POSTGRES_PASSWORD=123456 -d postgres
6167bc31723be06ff10d44221b2f8ad3722b9c48c40424c9b79bbb7fd39bcff1
```

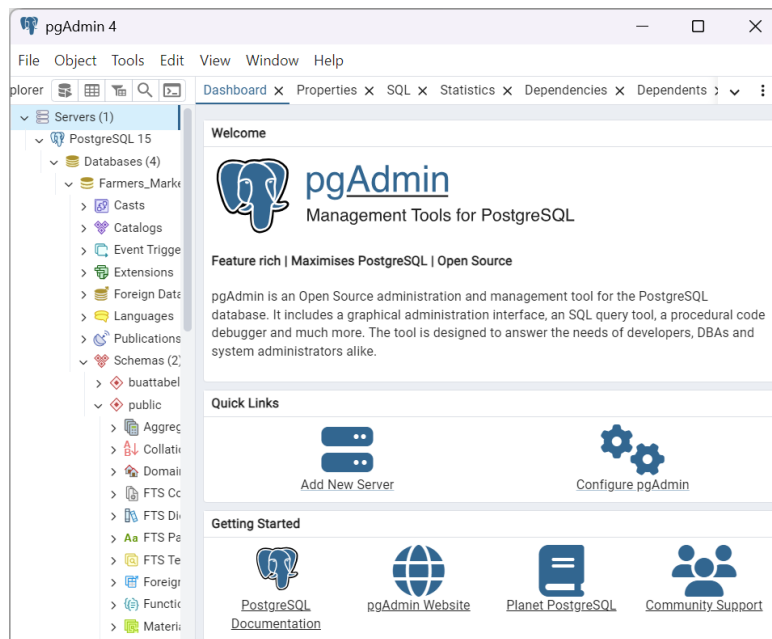


Menjalankan Image

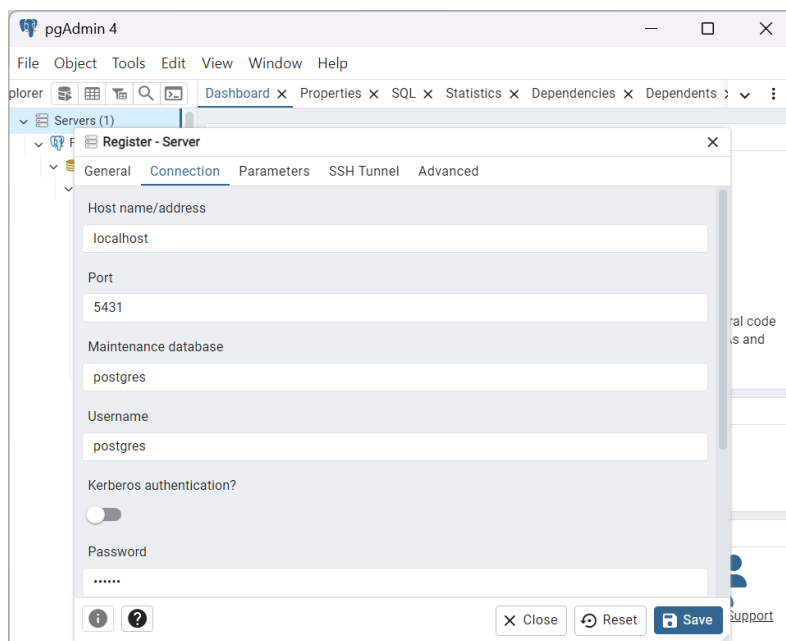
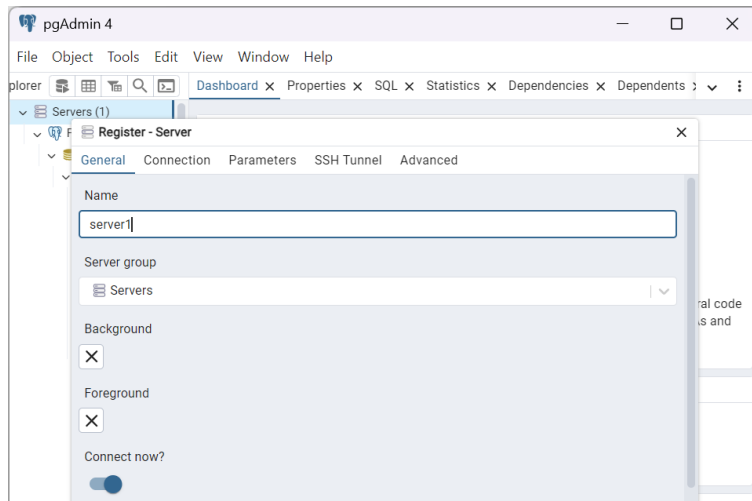
```
PS C:\WINDOWS\system32> docker run -it ubuntu
root@d61b3e933b9f:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@d61b3e933b9f:/# pwd
/
root@d61b3e933b9f:/# cd
root@d61b3e933b9f:/# pwd
/root
root@d61b3e933b9f:/#
```

Koneksi ke pgadmin

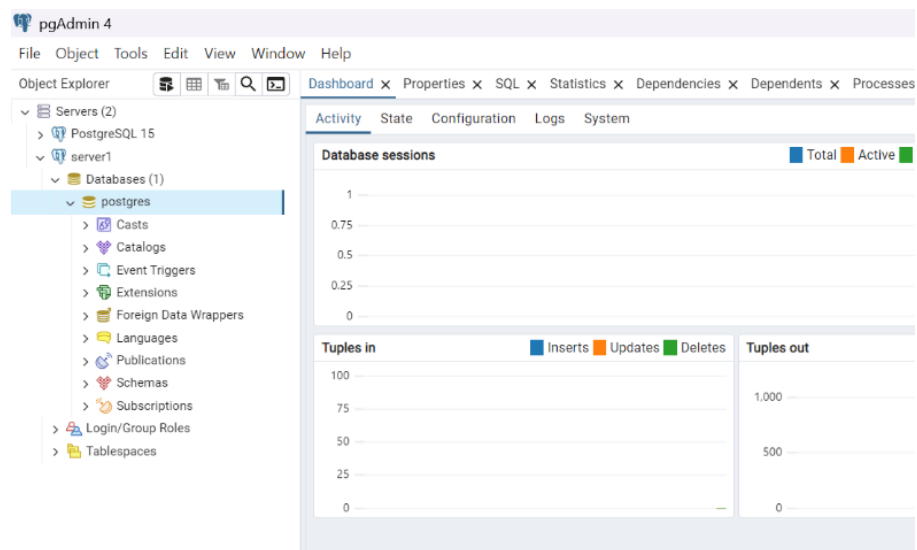
- Buka pgadmin
- Buka Server → dashboard → add new server



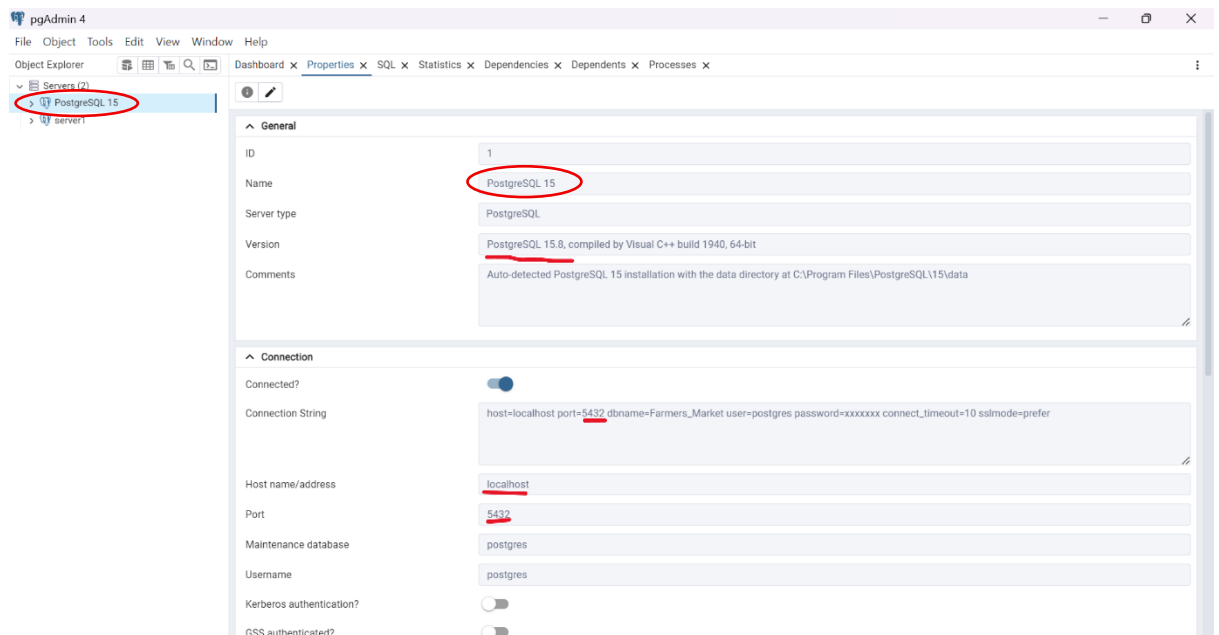
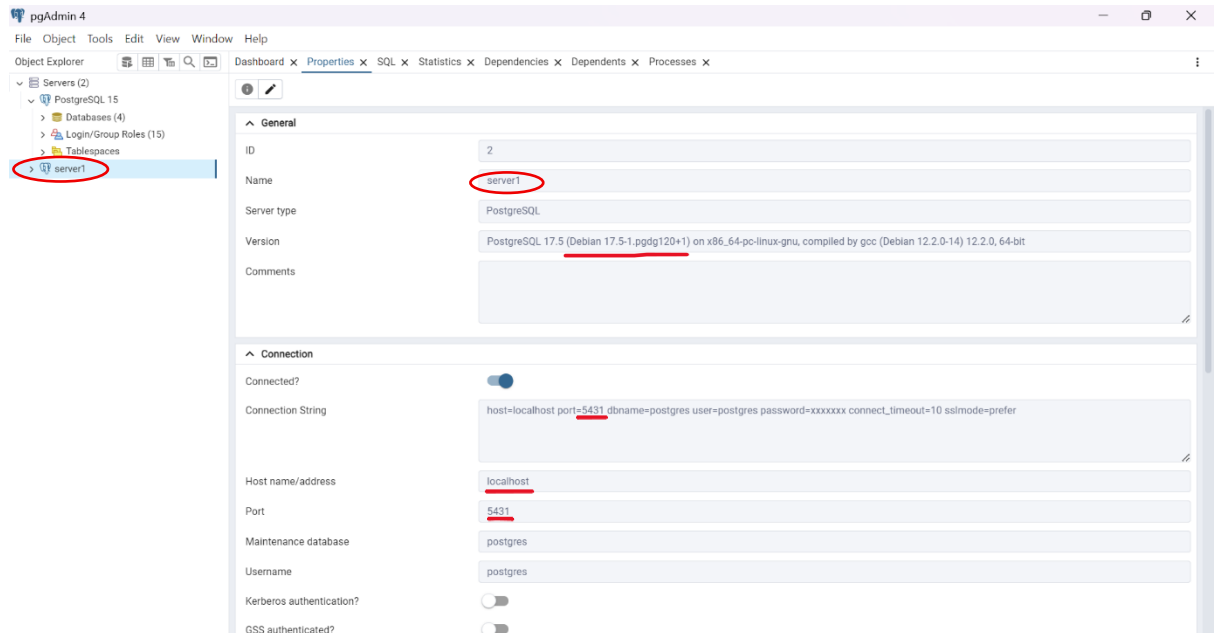
Koneksi ke pgadmin



Hasil:



Perbandingan



Install httpd – Run

Image httpd digunakan untuk menjalankan web server lokal dalam container.

```
PS C:\WINDOWS\system32> docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
4f4fb700ef54: Pull complete
be60498bea0a: Pull complete
8f86928406fd: Pull complete
162ef2c73af1: Pull complete
8dbbd44856ed: Pull complete
Digest: sha256:09cb4b94edaaa796522c545328b62e9a0db60315c7be9f2b4e02204919926405
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
```

```
PS C:\WINDOWS\system32> docker run -d -p 80:80 --name my-apache httpd
d824027d845451e99e028c1dc54136a53851897f5473c7de50f8f094f8cde4fa
PS C:\WINDOWS\system32> docker image ls
```

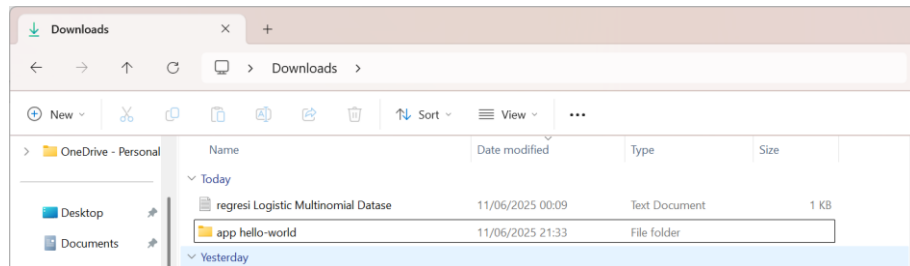
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	b59d21599a2b	10 days ago	117MB
postgres	latest	6efd0df010dc	2 weeks ago	621MB
mysql	latest	04768cb63395	7 weeks ago	1.17GB
httpd	latest	09cb4b94edaa	4 months ago	221MB

← → ↺ ⓘ localhost

It works!

Membangun Aplikasi di python

- Buat Direktori



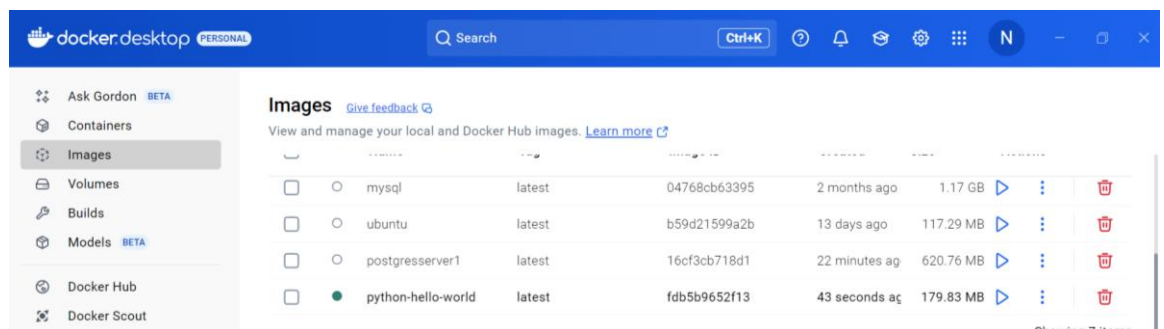
- Pastikan vscode ada atau download dan install jika belum ada
- Buat file hello-world.py



- Buat Dockerfile



- Jalankan lihat docker desktop



- Jalankan docker run python-hello-world

```
C:\Users\ASUS\Downloads\app>docker run hello-python
Hello World!
```

Aplikasi Sederhana untuk Prediksi Penonton Film Jumbo

1. Siapkan dataset simpan dalam bentuk file csv

Sumber data: Instagram @jumbofilm_id

File CSV ini adalah sumber data utama yang digunakan oleh aplikasi. Di dalamnya terdapat dua kolom penting, yaitu kolom tanggal dan jumlah penonton film *Jumbo*. Data ini bersifat historis dan menjadi dasar dari model prediksi yang dibuat. Agar aplikasi bisa berjalan dengan benar, data dalam file ini harus bersih dan terstruktur dengan format tanggal yang konsisten. Tanpa file ini, model tidak bisa dilatih dan aplikasi tidak bisa menampilkan hasil apapun.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Tanggal	Jumlah Penonton												
2	02-05-2025	8.127.009												
3	03-05-2025	8.294.207												
4	04-05-2025	8.489.638												
5	05-05-2025	8.689.676												
6	06-05-2025	8.776.400												
7	07-05-2025	8.844.724												
8	08-05-2025	8.906.224												
9	09-05-2025	8.957.462												
10	10-05-2025	9.014.021												
11	11-05-2025	9.127.865												
12	12-05-2025	9.293.830												
13	13-05-2025	9.474.665												
14	14-05-2025	9.568.290												
15	15-05-2025	9.604.512												
16	16-05-2025	9.630.188												
17	17-05-2025	9.655.610												
18	18-05-2025	9.712.344												
19	19-05-2025	9.777.989												
20	20-05-2025	9.803.918												
21	21-05-2025	9.826.272												
22	22-05-2025	9.844.075												
23	23-05-2025	9.856.163												
24	25-05-2025	9.897.319												
25	26-05-2025	9.929.659												

2. Buat file app-analisis-film-jumbo.py

File ini merupakan inti dari aplikasi karena di dalamnya terdapat seluruh kode yang menjalankan program prediksi. Di dalam file ini, data

dibaca dari file CSV, lalu dilakukan proses pemodelan menggunakan regresi linear dari library scikit-learn, dan hasilnya ditampilkan secara interaktif menggunakan Streamlit. Pengguna bisa melihat hasil prediksi selama 14 hari ke depan dalam bentuk tabel dan grafik, yang membuat data lebih mudah dipahami. File ini juga mengatur bagaimana tampilan web aplikasi, mulai dari judul, tabel, grafik, sampai keterangan di bagian bawah.

```
app-analisis-film-jumbo.py
1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5 from sklearn.linear_model import LinearRegression
6
7 st.set_page_config(page_title="Prediksi Penonton Jumbo", layout="centered")
8 st.title("🎬 Prediksi Penonton Film Jumbo (14 Hari ke Depan)")
9
10 @st.cache_data
11 def load_data():
12     df = pd.read_csv("DATA FILM JUMBO - Sheet1.csv")
13     df = df[['Tanggal', 'Jumlah Penonton']].dropna()
14     df['Tanggal'] = pd.to_datetime(df['Tanggal'], format='%d-%m-%Y')
15     df['Jumlah Penonton'] = df['Jumlah Penonton'].str.replace('.', '', regex=False).astype(int)
16     return df.sort_values('Tanggal').reset_index(drop=True)
17
18 df = load_data()
19
20 # Siapkan data untuk prediksi
21 X = np.array(range(len(df))).reshape(-1, 1)
22 y = df['Jumlah Penonton'].values
23
24 # Latih model regresi
25 model = LinearRegression()
26 model.fit(X, y)
27
28 # Prediksi 14 hari ke depan
29 future_days = 14
30 last_day = X[-1][0]
31 X_future = np.array([last_day + i for i in range(1, future_days + 1)]).reshape(-1, 1)
32 y_pred = model.predict(X_future)
33
34 # Buat dataframe prediksi
35 future_dates = pd.date_range(start=df['Tanggal'].iloc[-1] + pd.Timedelta(days=1), periods=future_days)
36 df_pred = pd.DataFrame({
37     'Tanggal': future_dates,
38     'Prediksi Penonton': y_pred.astype(int)
39 })
40 # Tampilkan tabel prediksi
41 st.subheader("📊 Prediksi 14 Hari ke Depan")
42 st.dataframe(df_pred)
43
44 # Grafik visualisasi
45 st.subheader("📈 Grafik Penonton Film Jumbo")
46 fig, ax = plt.subplots(figsize=(10, 5))
47 ax.plot(df['Tanggal'], df['Jumlah Penonton'], marker='o', label='Data Historis')
48 ax.plot(df_pred['Tanggal'], df_pred['Prediksi Penonton'], marker='x', linestyle='--', color='red', label='F')
49 ax.set_xlabel("Tanggal")
50 ax.set_ylabel("Jumlah Penonton")
51 ax.set_title("Prediksi Penonton Film Jumbo")
52 ax.legend()
53 plt.xticks(rotation=45)
54 plt.grid(True)
55 st.pyplot(fig)
56 st.caption("Aplikasi prediksi sederhana berbasis linear regression")
```

3. Buat file Dockerfile

Dockerfile berfungsi sebagai panduan untuk membungkus seluruh aplikasi ke dalam container Docker. Di dalam file ini ditentukan langkah-langkah seperti menggunakan Python versi tertentu, menyalin file aplikasi ke dalam container, menginstal pustaka yang diperlukan, serta menjalankan aplikasi

Streamlit. Dengan menggunakan Dockerfile, aplikasi bisa dijalankan di lingkungan manapun tanpa harus mengatur ulang konfigurasi, sehingga sangat memudahkan proses deployment.

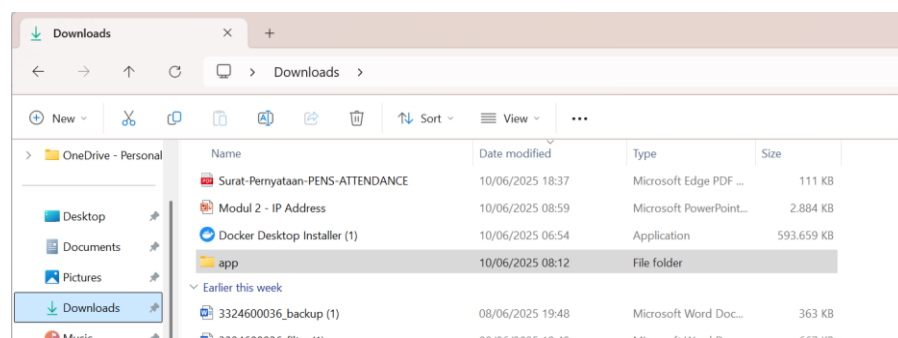
```
Dockerfile
1
2 # Gunakan image dasar Python
3 FROM python:3.10-slim
4
5 # Set direktori kerja di dalam container
6 WORKDIR /app
7
8 # Salin semua isi folder lokal ke container
9 COPY . .
10
11 # Instal semua dependensi
12 RUN pip install --no-cache-dir -r requirements.txt
13
14 # Buka port default Streamlit
15 EXPOSE 8501
16
17 # Jalankan aplikasi
18 CMD ["streamlit", "run", "app-analisis-film-jumbo.py", "--server.port=8501", "--server.address=0.0.0.0"]
19
```

4. Buat file requiremenst.txt

File ini berisi daftar pustaka (library) Python yang dibutuhkan agar aplikasi bisa dijalankan tanpa error. Di dalamnya terdapat nama-nama seperti streamlit, pandas, matplotlib, dan scikit-learn. File ini sangat berguna saat ingin menjalankan aplikasi di komputer lain atau di Docker, karena cukup menggunakan satu perintah untuk menginstal semua kebutuhan. Jadi, file ini memastikan lingkungan pengembangan aplikasi tetap konsisten dan bebas masalah.

```
requirements.txt
1 streamlit
2 pandas
3 numpy
4 matplotlib
5 scikit-learn
```

5. Buat folder baru untuk menyimpan file app-analisis-film-jumbo.py, Dorkfile, requirements.txt, dan dataset DATA FILM JUMBO - Sheet1.csv



6. Buka terminal dari folder app, lalu jalankan kode berikut:

```
PS C:\Users\ASUS\Downloads\app> docker build -t prediksi-film-jumbo .
[+] Building 892.0s (10/10) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile                 0.2s
=> => transferring dockerfile: 465B                                0.1s
=> [internal] load metadata for docker.io/library/python:3.10-slim 7.3s
=> [auth] library/python:pull token for registry-1.docker.io       0.0s
=> [internal] load .dockerignore                                    1.4s
=> => transferring context: 2B                                       1.3s
=> [1/4] FROM docker.io/library/python:3.10-slim@sha256:2ee0fb8794bf82af1b2db168ada5869b4c27d1772 0.2s
=> => resolve docker.io/library/python:3.10-slim@sha256:2ee0fb8794bf82af1b2db168ada5869b4c27d1772 0.2s
=> [internal] load build context                                    0.2s
```

```
=> => transferring context: 2.13kB                                  0.1s
=> CACHED [2/4] WORKDIR /app                                       0.0s
=> [3/4] COPY . .                                                  0.3s
=> [4/4] RUN pip install --no-cache-dir -r requirements.txt       616.4s
=> exporting to image                                              259.7s
=> => exporting layers                                              182.7s
=> => exporting manifest sha256:44c60c709921338abf9ad95451531c50fc2d049c205b9079cf6757cfd3efc95b 0.0s
=> => exporting config sha256:1e92655e29e37b997a9bc2e150b052b785146265ecdb57a2074d3635b0a01549 0.1s
=> => exporting attestation manifest sha256:4c8f124256e26a6ae6e3ce829e33a63378e324b7f75d9ebee0b9 0.1s
=> => exporting manifest list sha256:0717435b560965c8b42b8a875310d2409204305f2ee4702d9f4e1a99bf24 0.2s
=> => naming to docker.io/library/prediksi-film-jumbo:latest      0.1s
```

```
PS C:\Users\ASUS\Downloads\app> docker run -p 8501:8501 prediksi-film-jumbo

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.

You can now view your Streamlit app in your browser.

URL: http://0.0.0.0:8501
```

Cek images

Images Give feedback							
View and manage your local and Docker Hub images. Learn more							
<input type="checkbox"/>	Name	Tag	Image ID	Created	Size	Actions	
<input type="checkbox"/>	prediksi-jumbo	latest	886b4c0ee48f	2 hours ago	1.38 GB	▶	⋮
<input type="checkbox"/>	prediksi-film-jumbo	latest	0717435b5609	5 minutes ago	1.05 GB	▶	⋮

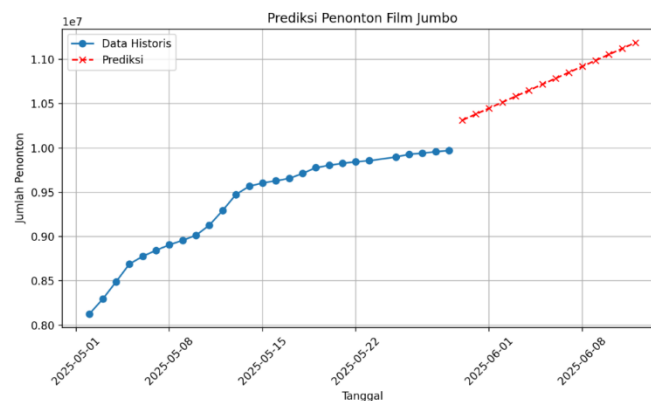
Akses aplikasi di browser: <http://localhost:8501>

Prediksi Penonton Film Jumbo (14 Hari ke Depan)

Prediksi 14 Hari ke Depan

	Tanggal	↓ Prediksi Penonton
13	2025-06-12 00:00:00	11190707
12	2025-06-11 00:00:00	11123249
11	2025-06-10 00:00:00	11055792
10	2025-06-09 00:00:00	10988334
9	2025-06-08 00:00:00	10920877
8	2025-06-07 00:00:00	10853419
7	2025-06-06 00:00:00	10785961
6	2025-06-05 00:00:00	10718504
5	2025-06-04 00:00:00	10651046

Grafik Penonton Film Jumbo



Aplikasi prediksi sederhana berbasis regresi linier

Penjelasan Aplikasi

Aplikasi ini merupakan sebuah alat prediksi sederhana yang dibuat menggunakan bahasa pemrograman Python dan framework Streamlit untuk menampilkan antarmuka pengguna berbasis web. Aplikasi ini memanfaatkan algoritma regresi linear dari pustaka scikit-learn untuk memprediksi jumlah penonton film "Jumbo" selama 14 hari ke depan berdasarkan data historis yang tersedia. Data jumlah penonton diolah dalam bentuk deret waktu, lalu dilatih menggunakan model linear untuk menghasilkan tren prediktif yang divisualisasikan

dalam bentuk grafik dan tabel. Tujuan dari aplikasi ini adalah untuk memberikan gambaran awal tentang kemungkinan jumlah penonton di masa mendatang secara praktis dan mudah dipahami, khususnya untuk keperluan analisis data dasar atau tugas akhir yang berkaitan dengan prediksi berbasis data.

Berdasarkan hasil prediksi, terlihat bahwa jumlah penonton film "Jumbo" cenderung mengalami peningkatan dari hari ke hari. Hal ini dapat dilihat dari grafik garis tren naik yang dihasilkan model, yang mengindikasikan bahwa film tersebut memiliki daya tarik yang kuat dan konsisten. Walaupun model ini bersifat sederhana dan tidak memperhitungkan faktor eksternal seperti promosi, hari libur, atau kompetisi film lain, hasilnya tetap berguna untuk memberikan estimasi awal dan membantu pengambilan keputusan dasar, seperti strategi distribusi atau evaluasi performa film.

Link Video Drive:

https://drive.google.com/file/d/1ISpuy-_UEgaOTz-QcRon3Rl9Oi61fEc0/view?usp=drivesdk