LEARNING PROGRESS REVIEW

#4

OUR MINI PROJECT, LINUX & UNIX ADMINISTRATION, DOCKER

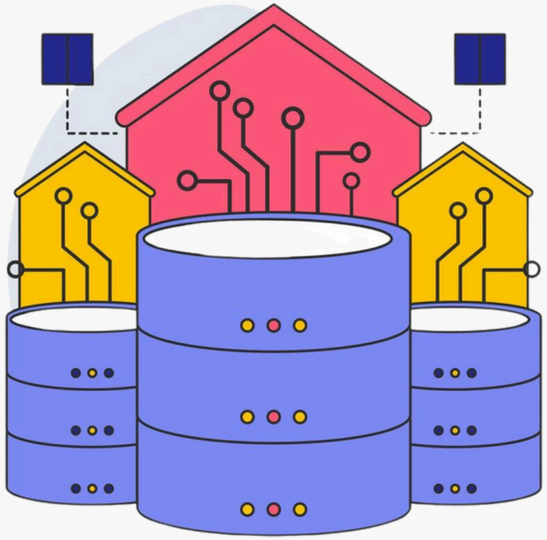
FAST TRACK DATA ENGINEER SCHOLARSHIP
BY
DIGITAL SKOLA

GARAP RENDANG TEAM

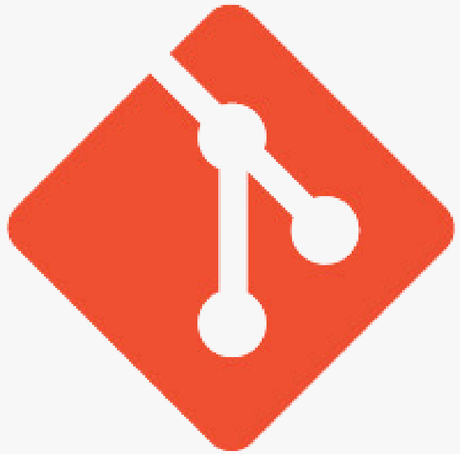
- Suhendar Aditya
- Chandra Trio Pamungkas
- Aryo Putra
- Kurniaman Andreas Zega
- Ahmad Fayyadh

MINI PROJECT

DATA INGESTION FROM DATA SOURCE TO DATA WAREHOUSE



Tools



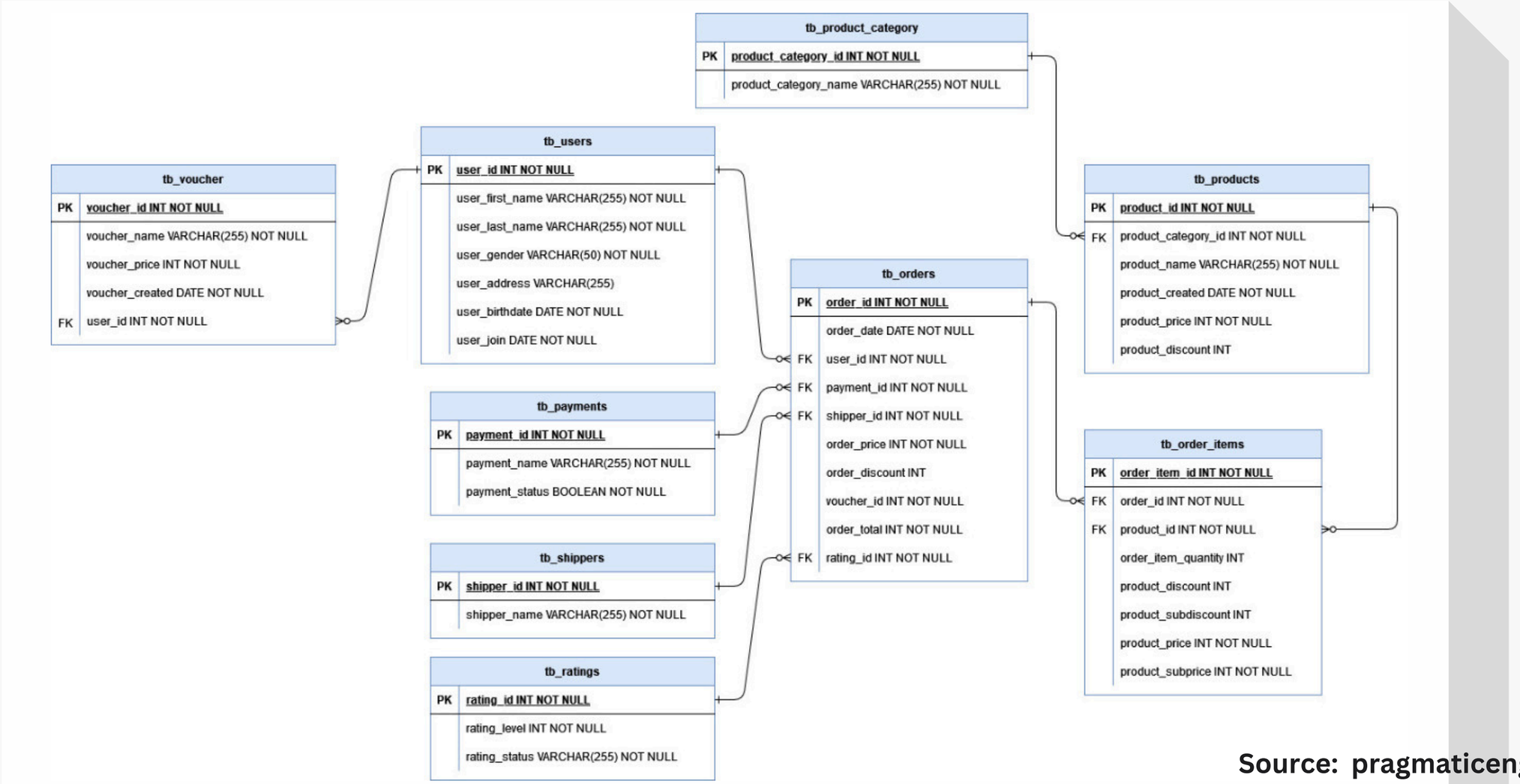
Source: pragmaticengineer.com

Background Project

Tim data analyst membutuhkan tabel untuk membuat dashboard terkait detail order dari data source yang ada di database production marketplace, Anda sebagai data engineer di minta untuk membuat script data migrasi dari tabel source tersebut ke data warehouse, sehingga tim data analyst bisa menggunakan table dari data warehouse tanpa membebankan database production marketplace.

Source: pragmaticengineer.com

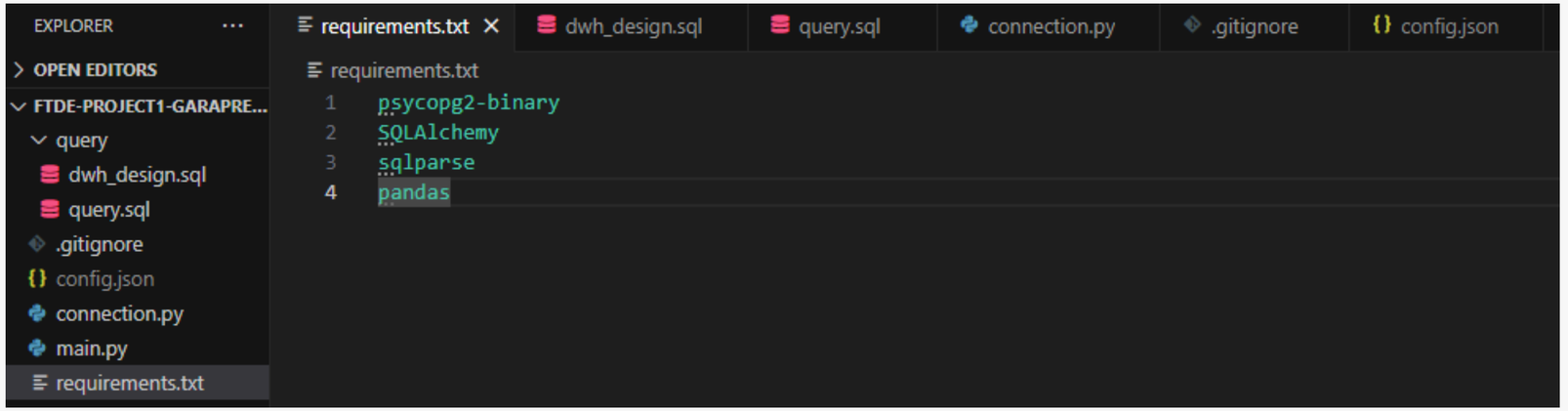
Schema



Source: pragmaticengineer.com

How we do it?

1. Install the requirements

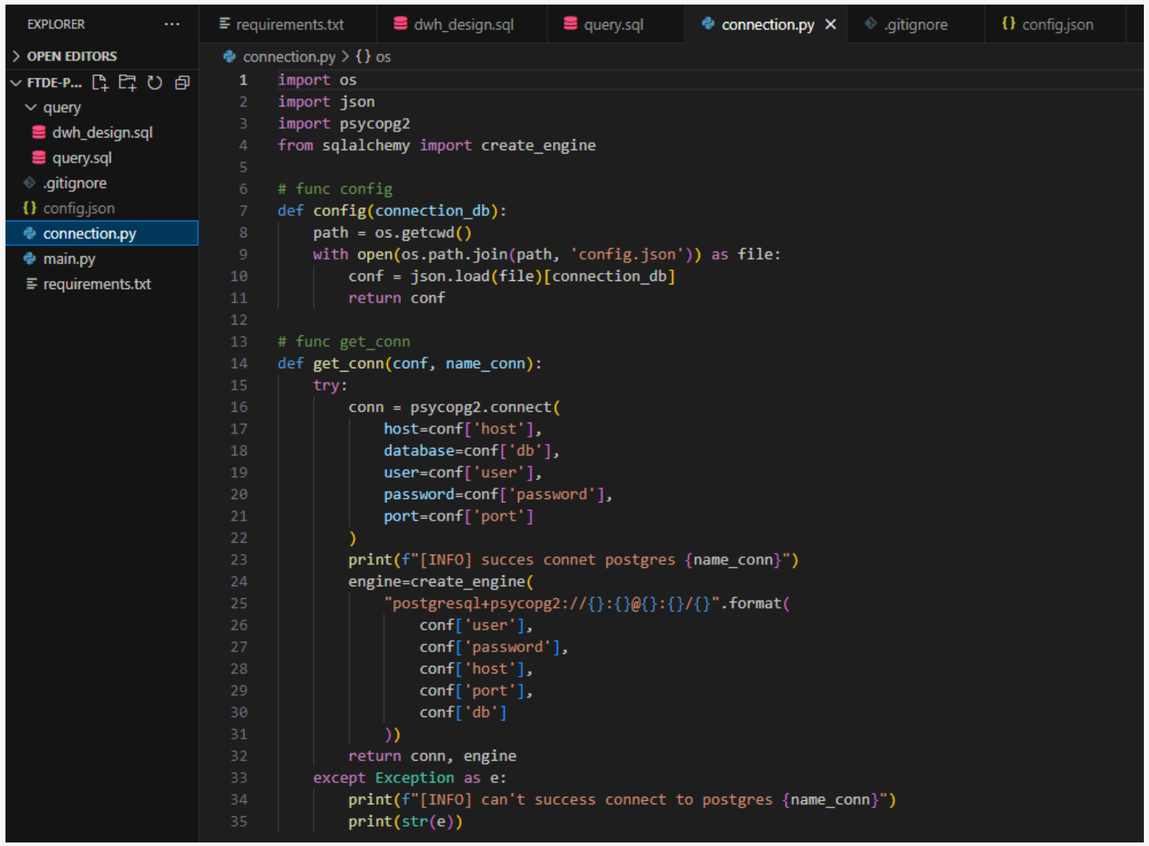


Install menggunakan “pip/conda install -r requirements.txt”. jangan lupa untuk membuat virtual environtment (venv) terlebih dahulu agar lebih mudah dan tertata dengan requirement mendatang di project lainnya.

Source: pragmaticengineer.com

How we do it?

2. Create connection.py

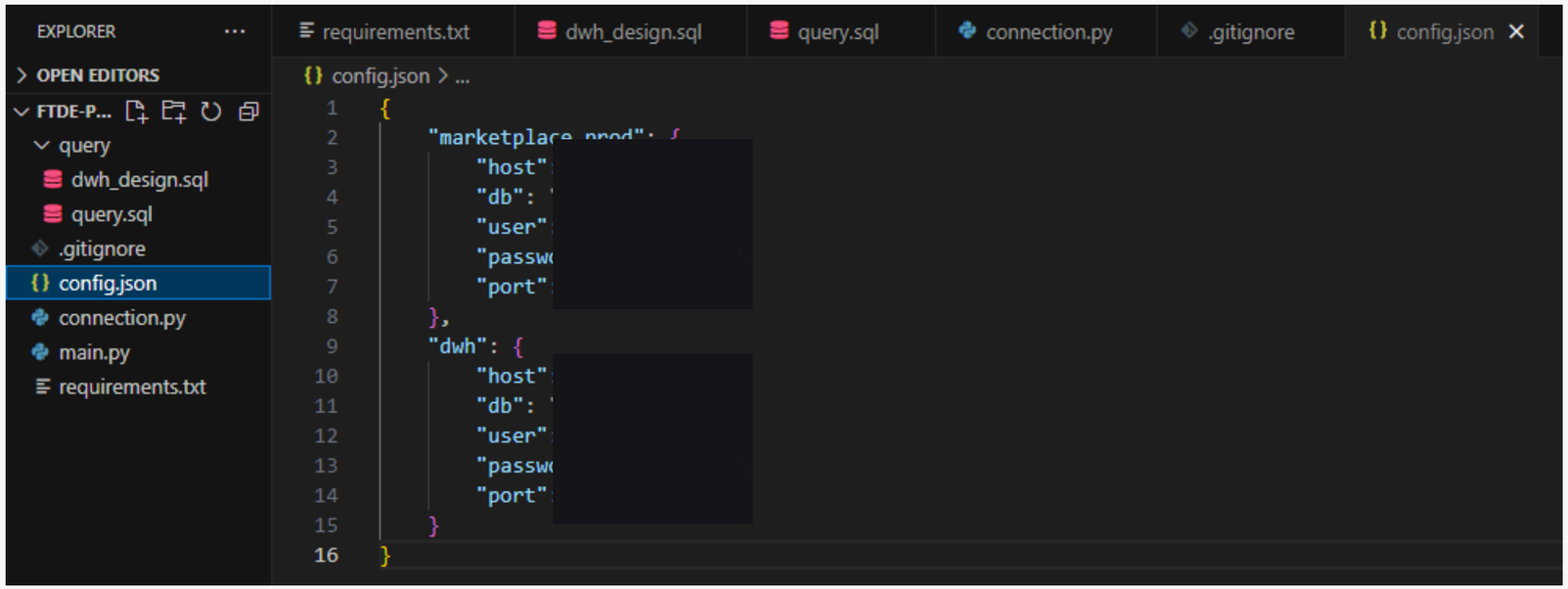


Membuat “connection .py” dengan detail user, password, host, port, dan db yang terdapat pada config.json.

Source: pragmaticengineer.com

How we do it?

3. Create config.json



Berisi detail dari source dan destination, jangan lupa untuk memasukan file ini kedalam .gitignore agar tidak ter push kedalam repository github.

Source: pragmaticengineer.com

How we do it?

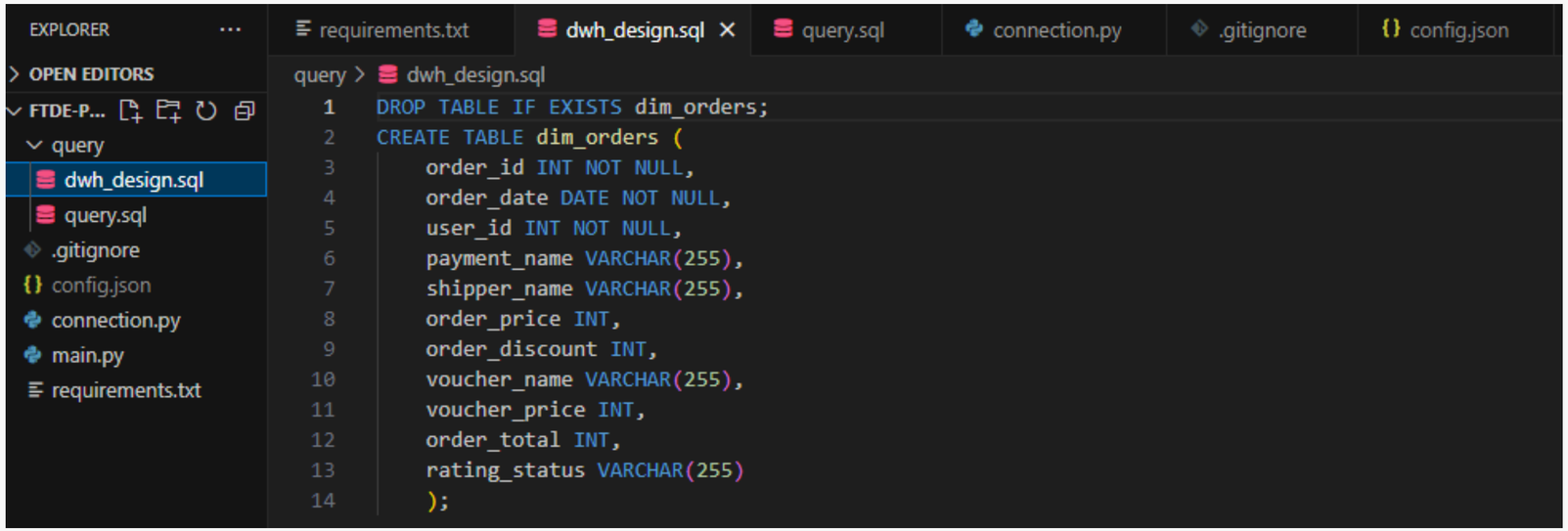
4. Create .gitignore

Lengkapnya dapat dicari dengan keyword “python gitignore”.

Source: pragmaticengineer.com

How we do it?

5. Create query folder and data warehouse design file

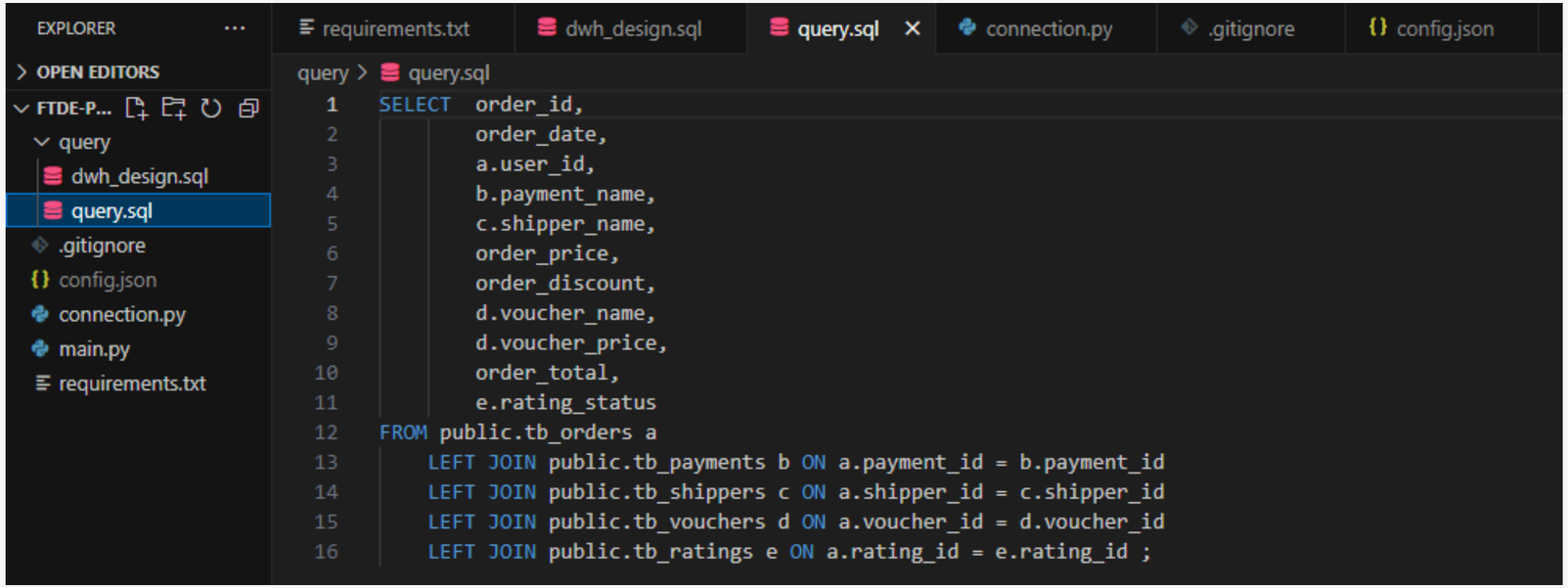


Membuat table pada data warehouse dengan detail column dari table lainnya pada source.

Source: pragmaticengineer.com

How we do it?

6. Create query file

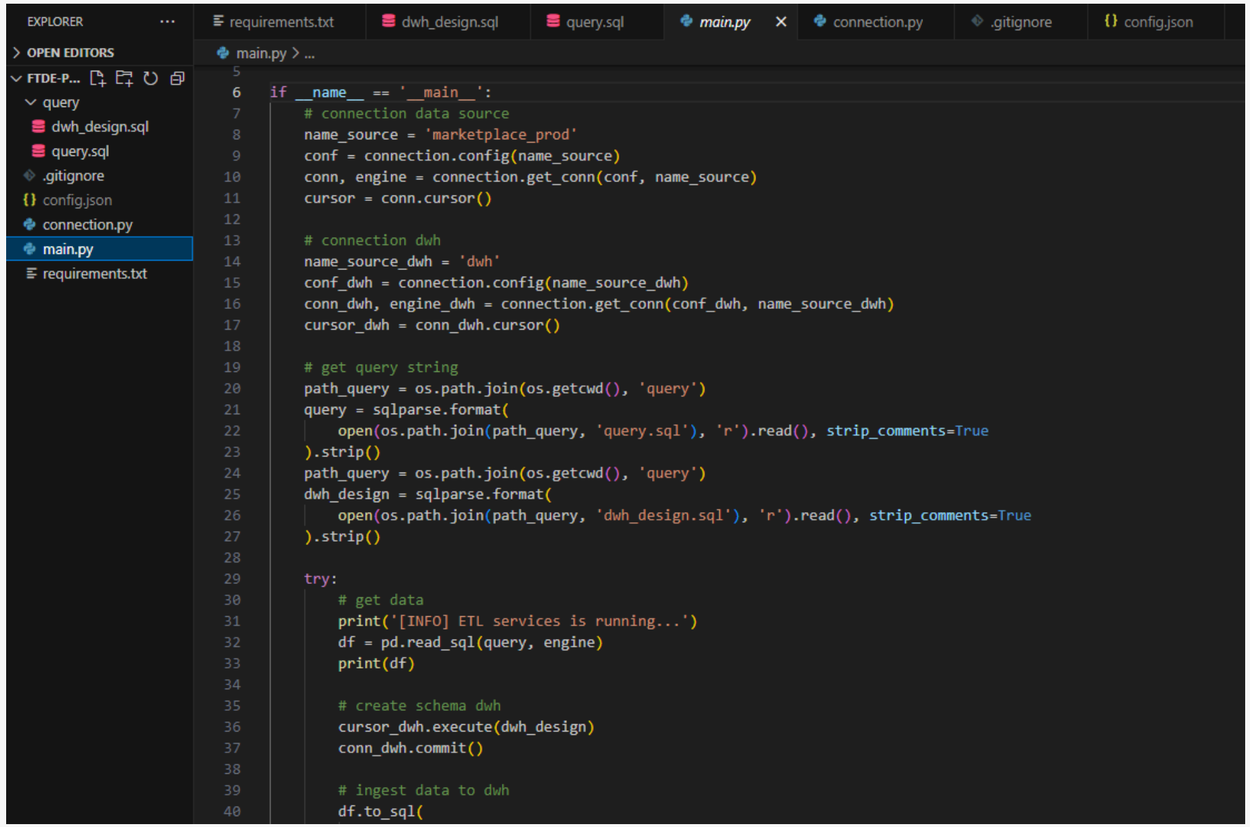


Membuat query untuk mengambil data dari beberapa table berbeda dengan “left join’ untuk diteruskan ke data warehouse.

Source: pragmaticengineer.com

How we do it?

7. Create main.py file

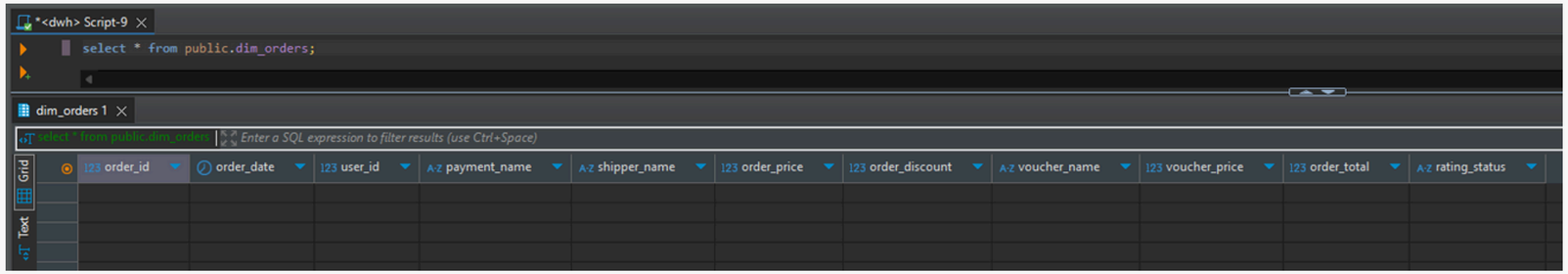


Membuat query utama untuk menjalankan semuanya secara berurutan.

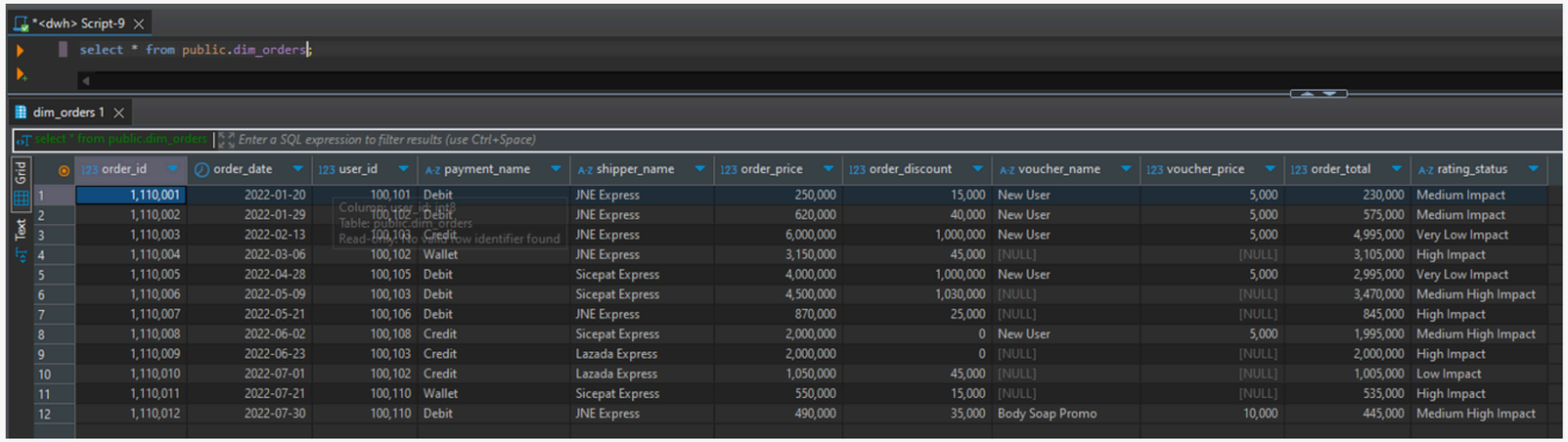
Source: pragmaticengineer.com

Results

From this



To this



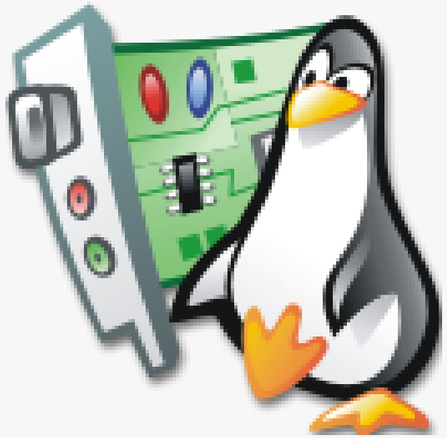
Source: pragmaticengineer.com

LINUX

“Sistem operasi (OS) komputer yang bersifat open source dan gratis”

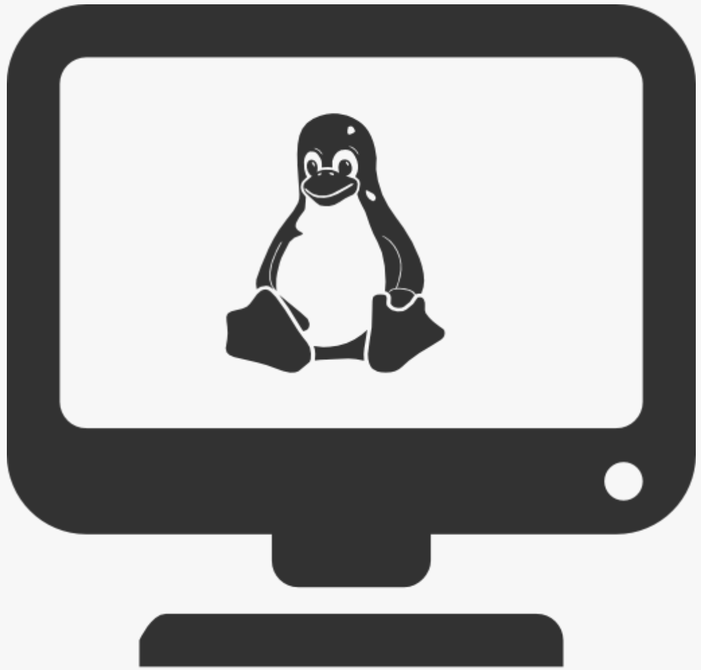
ALASAN MEMPELAJARI LINUX

- **Open Source**: Bebas digunakan dan dimodifikasi.
- **Gratis**: Tidak memerlukan biaya lisensi.
- **Keamanan Tinggi**: Tingkat keamanan lebih baik dibandingkan sistem lainnya.
- **Dominasi Cloud:** 90% sistem cloud berbasis Linux.
- **Kompatibilitas Alat Data Engineering**: Sebagian besar alat untuk data engineering berjalan di Linux.



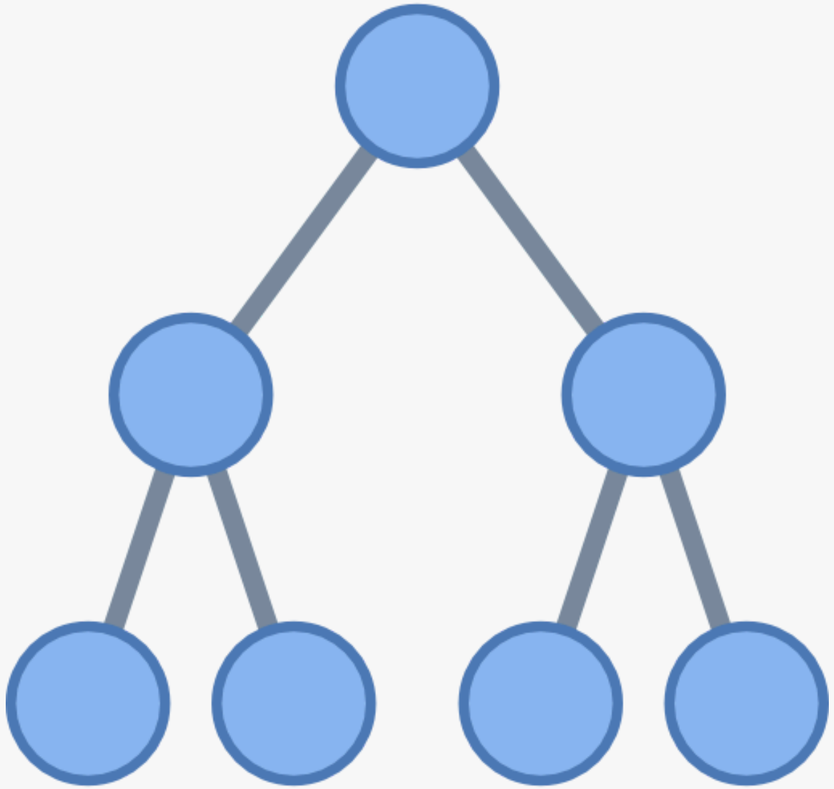
TENTANG LINUX

- **Sejarah:** Diciptakan oleh Linus Torvalds, terinspirasi dari kernel Minix milik Andrew Tanenbaum.
- **Distro Linux:** Sistem operasi berbasis kernel Linux, seperti Ubuntu, Fedora, dll., mencakup aplikasi seperti server web, database, dan desktop environment.



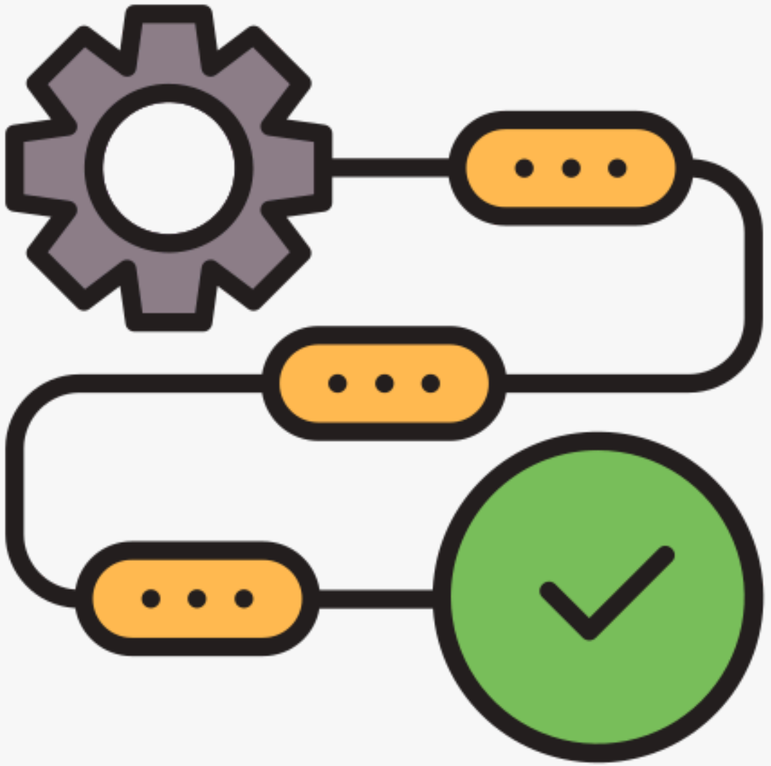
STRUKTUR DIREKTORI LINUX

- **/ (Root):** Direktori utama.
- **/bin:** File biner untuk perintah dasar shell.
- **/etc:** File konfigurasi sistem inti.
- **/home:** Direktori pribadi pengguna.
- **/var:** File runtime sistem, log, dan data lainnya.



PROSES DI LINUX

- **Foreground Process:** Proses interaktif yang dijalankan oleh pengguna.
- **Background Process:** Proses non-interaktif yang berjalan otomatis.
- **Manajemen Proses:** Menggunakan perintah seperti ps dan top untuk memonitor penggunaan sumber daya.



CRONTAB

- **Cron Daemon:** Menjadwalkan tugas otomatis (cronjobs) pada waktu tertentu.
- **Perintah Crontab:**
    - `crontab -e: Membuat atau mengubah cronjobs.`
    - crontab -l: Melihat daftar cronjobs.
    - crontab -r: Menghapus cronjobs

KESIMPULAN

“ **Linux** merupakan sistem operasi yang fleksibel, aman, dan dominan di lingkungan cloud dan data engineering, sehingga penting untuk dipelajari oleh administrator sistem dan profesional teknologi.”

Unix dan **Linux** merupakan dua sistem operasi yang berbeda namun sering dibandingkan karena kemiripannya,

Unix merupakan sistem operasi komersial sedangkan **Linux** sistem operasi open source yang memiliki basis kernel Unix

UNIX®
A Standard of The Open Group®

UNIX SYSTEM ADMINISTRATOR

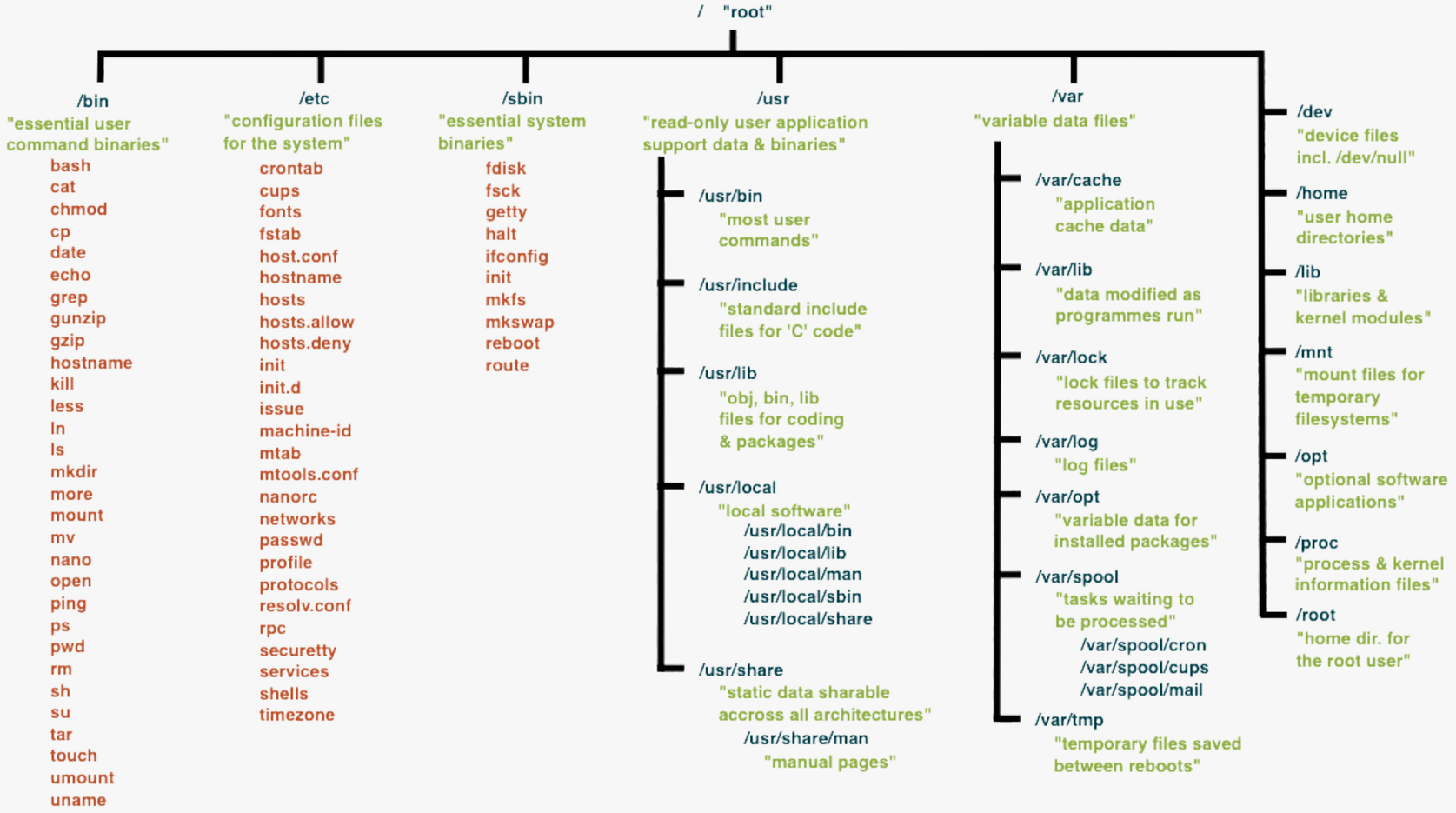
Bertugas mengelola sistem operasi unix, beberapa tugas nya antara lain :

- Mengatur lingkungan kerja (*.env*) yang menggunakan sistem operasi Unix multi-user
- Melakukan tugas manajemen sistem, seperti menyimpan dan melindungi data pengguna, serta menjaga keamanan
- Mengembangkan manual dan proses auto pada panduan teknis untuk perusahaa

UNIX®
A Standard of The Open Group®

GARAP RENDANG
Grup Andalan Rakitan Pipeline Record Engineer Ngolah Data Anti Gagal

UNIX SYSTEM ADMINISTRATOR - FILESYSTEM HEIRARCHY



(RCC-UCHICAGO, 2024)

UNIX®
A Standard of The Open Group®

UNIX SYSTEM ADMINISTRATOR - USER MANAGE

Instruksi	Keterangan	Contoh
useradd	Menambahkan User	useradd [options] [User_name]
userdel	Menghapus entitas user	userdel [options] username
usermod	Memodifikasi data user	sudo usermod -c "[message]" test_user
passwd	Merubah password user	passwd [options] [username]
groupadd	Menambah sebuah group	groupadd [options] group_name

UNIX®
A Standard of The Open Group®

UNIX SYSTEM ADMINISTRATOR - ENVIRONMENT

Instruksi	Keterangan	Contoh
env	Menjalankan program pada lingkungan tertentu	env [OPTION]... [-][NAME=VALUE]... [COMMAND [ARG]...]
printenv	Melihat environment (*.env*) yang ingin dilihat	printenv [name]
set	memasukan nilai ke dalam sebuah environment	set [options] [arguments]
unset	hapus environtment variable	unset var_name_here unset [options] var
export	memungkinkan memperbarui dan menyebarkan nilai variabel lingkungan ke sesi saat ini dan proses turunan yang dihasilkan	export [-f] [-n] [name[=value] ...] or export -p

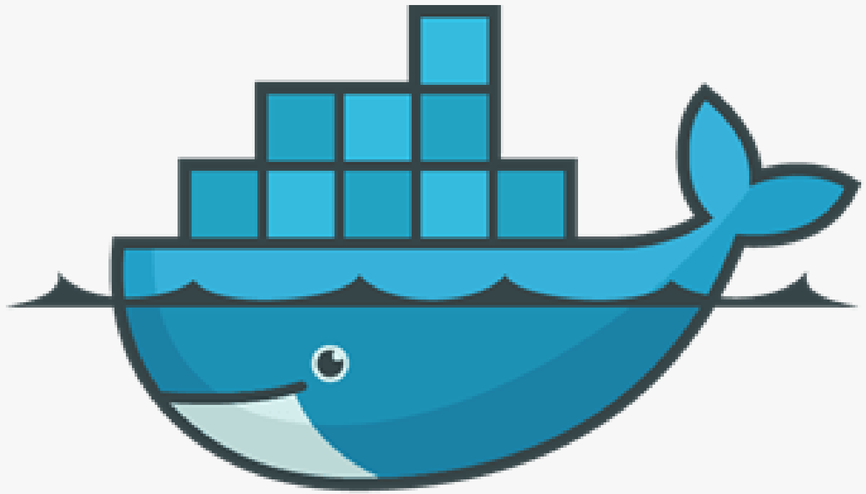
UNIX®
A Standard of The Open Group®

KESIMPULAN

“ perintah **Unix** dapat menjadi alat yang ampuh untuk menavigasi sistem berkas, mengelola berkas, dan membangun sistem informasi”

DOCKER

“Platform untuk mengembangkan, mengirim dan menjalankan aplikasi pada lingkungan yang terisolasi”



ARSITEKTUR DOCKER

- **Docker Client:** Aplikasi yang digunakan untuk berinteraksi dan memberi perintah ke docker daemon.
- **Docker Daemon:** Aplikasi utama yang dilajankan di atas operating system (OS) berfungsi mengelola image, container, volume, dan networks.
- **Docker Registry:** Tempat penyimpan docker images yang bisa disimpan secara privasi atau publik

Source: pragmaticengineer.com

KOMPONEN DOCKER

- **Images:** Sebuah template intruksi untuk membuat container
- **Containers:** Aplikasi yang dijalankan berdasarkan image yang digunakan pada lingkungan yang terisolasi.
- **Volumes:** Tempat penyimpanan yang dikelola oleh docker dan dapat digunakan oleh container
- **Networks:** Jaringan virtual yang dapat menghubungkan satu container dengan container lainnya.

Source: pragmaticengineer.com

DOCKERFILE

File yang berisi perintah-perintah yang dapat dijalankan untuk membuat image.

```
FROM python:3.8-slim-buster
WORKDIR /app
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt
COPY . .
CMD ["python", "app.py"]
```

Source: pragmaticengineer.com

DOCKER COMPOSE

Alat untuk mengatur dan menjalankan beberapa container sekaligus di docker berdasarkan apa yang telah didefinisikan pada file berformat YAML

```
services:
  web:
    image: python:3.8-slim-buster
    container_name: flask_app
    working_dir: /app
    volumes:
      - .:/app
    ports:
      - "5000:5000"
    command: >
      sh -c "pip install -r requirements.txt &&
             python app.py"
```

Source: pragmaticengineer.com

THANK YOU

ANY FEEDBACKS?