

Networking, Security, and Orchestration of Docker

Aplikasi berikut dibangun dengan arsitektur sebagai berikut:

- Frontend: menggunakan Nginx untuk melayani halaman statis
- Backend: API sederhana berbasis Golang Gin
- Database: PostgreSQL untuk menyimpan data aplikasi
- Networking: Backend dan database harus saling terhubung dengan jaringan privat
- Volume: Database PostgreSQL memerlukan penyimpanan data yang persisten
- Github: <https://github.com/adychahyoputro/test-apps>

Langkah-langkah menjalankan aplikasi sebagai berikut:

1. Clone repository di server masing-masing dengan commad berikut:

git clone <https://github.com/adychahyoputro/test-apps.git>

```
root@ubuntu1:/home/ubuntu1/digital_skola# git clone https://github.com/adychahyoputro/test-apps.git
```

2. Lalu pindah folder kedalam project yang sudah di clone:

```
root@ubuntu1:/home/ubuntu1/digital_skola/test-apps# cd backend/  
root@ubuntu1:/home/ubuntu1/digital_skola/test-apps#
```

3. Sebelum menjalankan aplikasi, setting file .env, file Dockerfile untuk backend, file Dockerfile untuk frontend, file docker-compose, file script.js, ganti IP dan PORT yang disesuaikan dengan environment server masing-masing

```
root@ubuntu1:/home/ubuntu1/digital_skola/test-apps# cd backend/  
root@ubuntu1:/home/ubuntu1/digital_skola/test-apps/backend# cat .env  
# ubuntu  
DB_HOST=192.168.157.160  
DB_PORT=5435  
DB_USER=postgres  
DB_PASSWORD=root  
DB_NAME=testdb  
SERVER_ADDRESS=:8082
```

4. Setelah di setting semuanya, lalu simpan dan jalankan aplikasi dengan command berikut:
docker-compose up -d -> ini dijalankan tepat didalam folder project /test-apps

```

1 file changed, 7 user(s) created
root@ubuntu1:/home/ubuntu1/digital_skola/test-apps# docker compose up -d
[+] Running 0/1
[+] Running 0/1end Building
[+] Building 3.4s (15/16)
[+] Building 5.8s (25/25) FINISHED
=> [backend internal] load build definition from Dockerfile
=> => transferring dockerfile: 354B
=> [backend internal] load metadata for docker.io/library/golang:alpine
=> [backend internal] load metadata for docker.io/library/alpine:3.21
=> [backend internal] load .dockerignore
=> => transferring context: 2B
=> [backend builder 1/5] FROM docker.io/library/golang:alpine@sha256:d37127f39271451847bcd91fc53ee014829603c96b91d62ff65ab3a7d1fb3c5e
=> [backend stage-1 1/4] FROM docker.io/library/alpine:3.21@sha256:b97e2a89d0b9e4011bb88c02dd91c544b8c781acf1f4d559e7c8f12f1047ac3
=> [backend internal] load build context
=> => transferring context: 695B
=> CACHED [backend stage-1 2/4] WORKDIR /app
=> CACHED [backend builder 2/5] WORKDIR /build
=> CACHED [backend builder 3/5] COPY . .
=> CACHED [backend builder 4/5] RUN go mod download
=> CACHED [backend builder 5/5] RUN CGO_ENABLED=0 GOOS=linux go build -o /binary
=> CACHED [backend stage-1 3/4] COPY --from=builder /binary /binary
=> CACHED [backend stage-1 4/4] COPY --from=builder /build/.env .
=> [backend] exporting to image
=> => exporting layers
=> => writing image sha256:77c7778f872245b597ebb670f47997496eb0a6f2d4c4cb4d519ca4dd61b35a8
=> => naming to docker.io/library/test-apps-backend
=> [backend] resolving provenance for metadata file
=> [frontend internal] load build definition from Dockerfile
=> => transferring dockerfile: 186B
=> [frontend internal] load metadata for docker.io/library/nginx:alpine-slim
=> [frontend internal] load .dockerignore
=> => transferring context: 2B
=> [frontend 1/3] FROM docker.io/library/nginx:alpine-slim@sha256:a10918f045c90a8304eafdbfcb6ecdee070fe9d27fbbdd53ab6469f26cc8830
=> [frontend internal] load build context
=> => transferring context: 91B
=> CACHED [frontend 2/3] WORKDIR /frontend-build
=> CACHED [frontend 3/3] COPY . /usr/share/nginx/html
=> [frontend] exporting to image
=> => exporting layers
[+] Running 10/10age sha256:2b6f51f9141ae263ad798bcea9161450eac3fe631b445fbf14dc997e910e054c
Service backend Built
Service frontend Built
Network test-apps_net-test-apps Created
Network test-apps_default Created
Volume test-apps_vol-backend Created
Volume test-apps_vol-postgres Created
Volume test-apps_vol-frontend Created
Container test-apps-db-1 Started
Container test-apps-backend-1 Started
Container test-apps-frontend-1 Started
root@ubuntu1:/home/ubuntu1/digital_skola/test-apps# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
test-apps-frontend   latest              2b6f51f9141a       49 minutes ago     11.8MB
backend-image        latest              eeb7b7e51f4a       4 hours ago        20.7MB
test-apps-backend    latest              77c7778f8722       4 hours ago        20.7MB
frontend-image       latest              ca1ab6555113       18 hours ago       17.3MB
postgres             latest              4bc6cc20ca7a       6 weeks ago        435MB

```

- Setelah berhasil dijalankan, cek image dahulu dengan command berikut:
docker images

```

root@ubuntu1:/home/ubuntu1/digital_skola/test-apps# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
test-apps-frontend   latest              2b6f51f9141a       49 minutes ago     11.8MB
backend-image        latest              eeb7b7e51f4a       4 hours ago        20.7MB
test-apps-backend    latest              77c7778f8722       4 hours ago        20.7MB
frontend-image       latest              ca1ab6555113       18 hours ago       17.3MB
postgres             latest              4bc6cc20ca7a       6 weeks ago        435MB
root@ubuntu1:/home/ubuntu1/digital_skola/test-apps# docker container ls -a

```

Terlihat sudah terbentuk 3 image yaitu test-apps-frontend, test-apps-backend dan postgres

- Lalu cek container dengan command berikut:
docker container ls -a

```

root@ubuntu1:/home/ubuntu1/digital_skola/test-apps# docker container ls -a
CONTAINER ID   IMAGE               COMMAND                  CREATED        STATUS        PORTS
be5f528e26bb   test-apps-frontend "/docker-entrypoint. ..." About a minute ago Up 57 seconds 8083/tcp, 0.0.0.0:8083->80/tcp, [::]:8083->80/tcp
p             test-apps-frontend-1
e5220de1bdc4   test-apps-backend  "/binary"                About a minute ago Up 50 seconds 0.0.0.0:8082->8082/tcp, [::]:8082->8082/tcp
9c7b4bdf4ba    postgres           "docker-entrypoint.s ..." About a minute ago Up About a minute 0.0.0.0:5435->5432/tcp, [::]:5435->5432/tcp
20d825a9e191   test-apps-db-1     "/binary"                4 hours ago   Exited (2) 59 minutes ago
ct-backend    test-apps-db-1
10cdd1d5a1b6   frontend-image     "/docker-entrypoint. ..." 18 hours ago   Exited (0) 17 hours ago
8c6894091ece   postgres           "docker-entrypoint.s ..." 3 days ago    Exited (0) 59 minutes ago
ct-postgres01 test-apps-postgres-1
root@ubuntu1:/home/ubuntu1/digital_skola/test-apps# docker volume ls

```

Terlihat sudah terbentuk 3 container yaitu test-apps-frontend-1, test-apps-backend-1, dan test-apps-db-1 dan sudah up langsung

- Lalu cek volume dengan command berikut:
Docker volume ls

```

root@ubuntu1:/home/ubuntu1/digital_skola/test-apps# docker volume ls
DRIVER          VOLUME NAME
local           test-apps_vol-backend
local           test-apps_vol-frontend
local           test-apps_vol-postgres
local           vol-postgres
root@ubuntu1:/home/ubuntu1/digital_skola/test-apps# docker network list
NETWORK ID      NAME

```

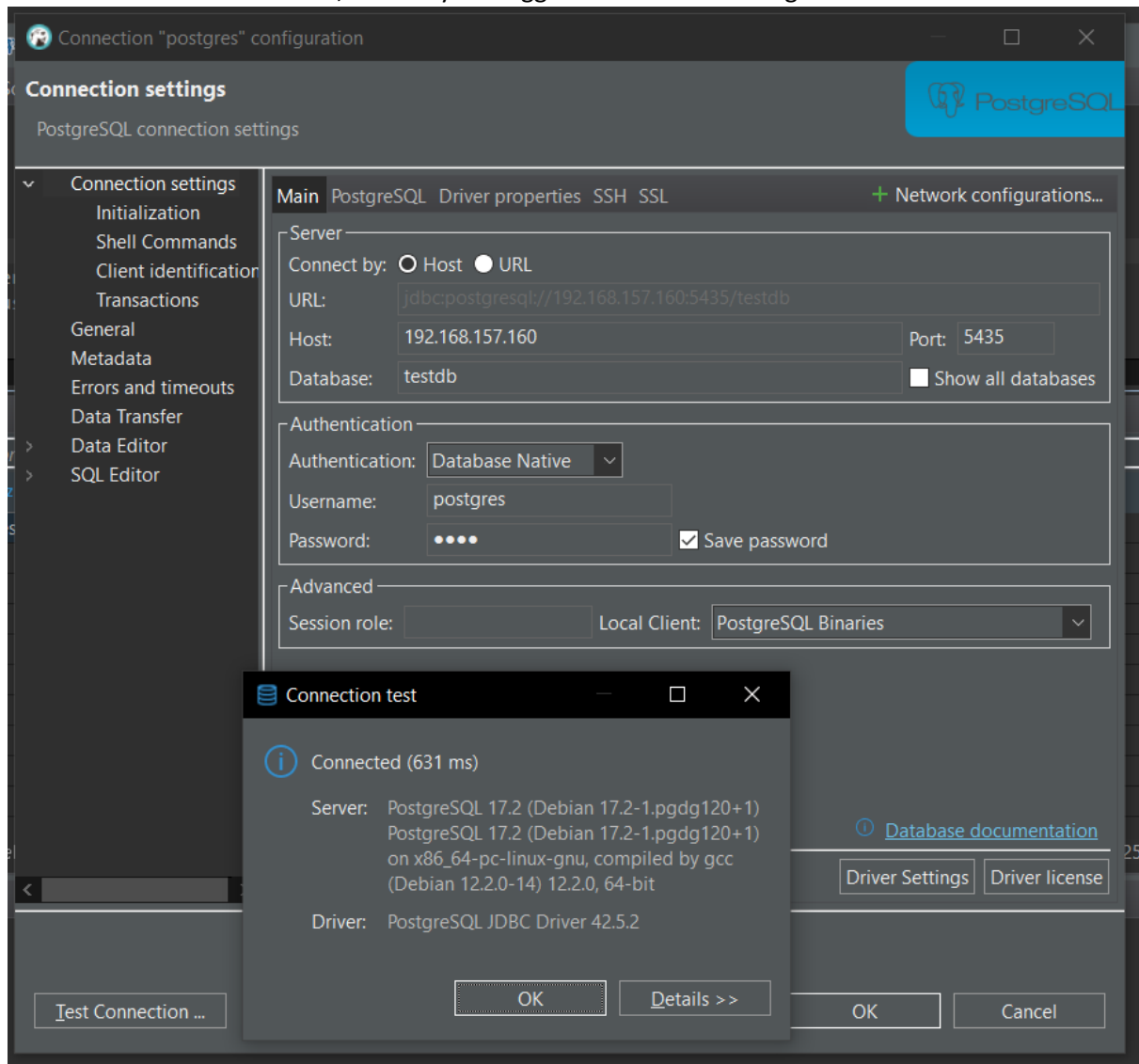
Terlihat sudah terbentuk 3 volume yaitu test-apps_vol-backend, test-apps_vol-frontend dan test-apps_vol-postgres

8. Lalu cek network dengan command berikut:
docker network list

```
root@ubuntu1:/home/ubuntu1/digital_skola/test-apps# docker network list
NETWORK ID          NAME                DRIVER             SCOPE
24b570183174        bridge              bridge              local
e664a189865a        host                host                local
95bed92ab034        none                null                local
7d50b9543040        test-apps_default   bridge              local
f708992e382a        test-apps_net-test-apps bridge              local
root@ubuntu1:/home/ubuntu1/digital_skola/test-apps# cd ..
```

Terlihat sudah terbentuk network yaitu test-apps_net-test-apps dengan driver bridge

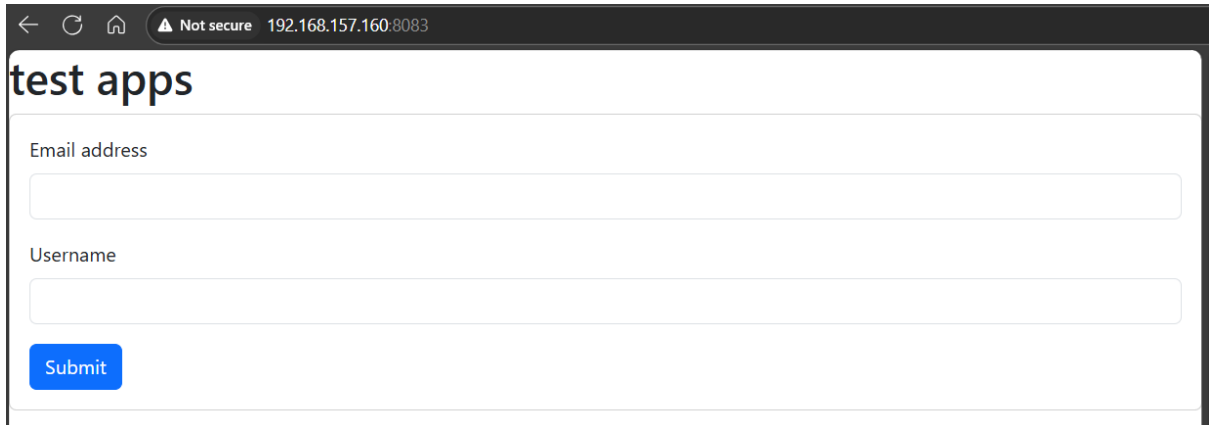
9. Lalu cek connection database, disini saya menggunakan dbeaver sebagai berikut:



Input dengan host dan port masing-masing server, terlihat connection succes

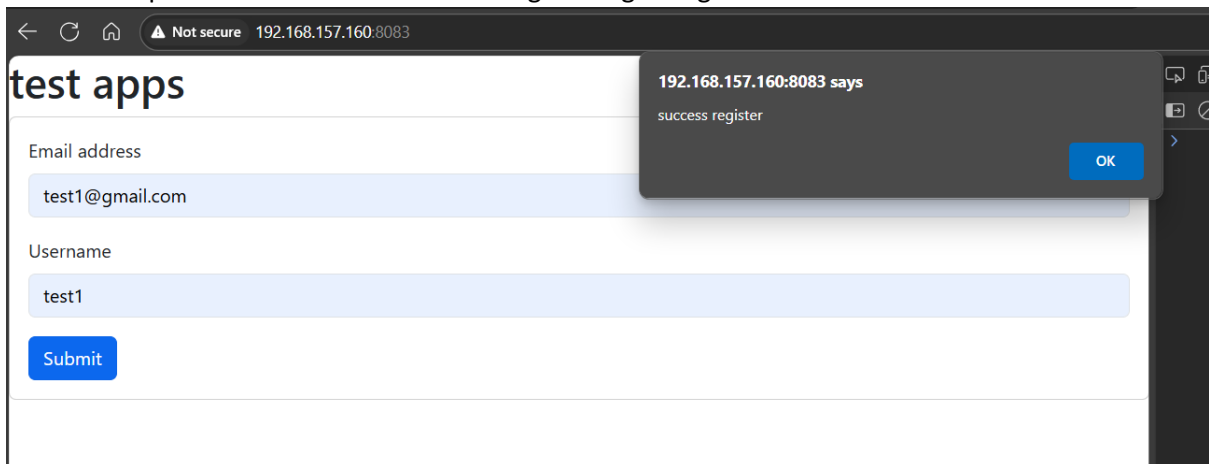
10. Lalu buat dahulu table users di dalam database testdb berikut:
create table users (
id VARCHAR(255),
username VARCHAR(25),
email VARCHAR(25));

11. Lalu cek website dengan url hostname dan port server masing-masing sebagai berikut:



The screenshot shows a web browser window with the address bar displaying '192.168.157.160:8083' and a 'Not secure' warning. The page title is 'test apps'. Below the title, there is a form with two input fields: 'Email address' and 'Username'. A blue 'Submit' button is located below the 'Username' field.

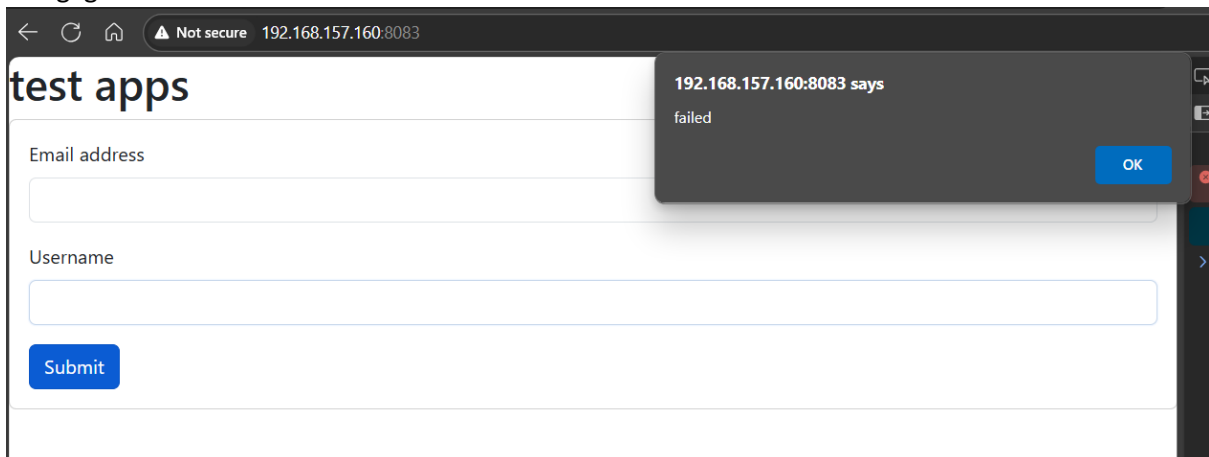
12. Lalu coba input email dan username masing-masing sebagai berikut:



The screenshot shows the 'test apps' registration form with the 'Email address' field containing 'test1@gmail.com' and the 'Username' field containing 'test1'. A blue 'Submit' button is visible. A dark gray notification box in the top right corner displays the message: '192.168.157.160:8083 says success register' with an 'OK' button.

Jika sukses maka akan muncul notifikasi success register

Jika gagal akan muncul notifikasi failed



The screenshot shows the 'test apps' registration form with empty 'Email address' and 'Username' fields. A dark gray notification box in the top right corner displays the message: '192.168.157.160:8083 says failed' with an 'OK' button.

13. Lalu cek di database apakah data sudah masuk atau belum

```
);
select * from users u ;
```

users 1 ×

select * from users u | Enter a SQL expression to filter results (use Ctrl+Space)

	A-z id	A-z username	A-z email	
1	892c79b0-43b9-46b5-ac98-3ea891b33e6b	test1	test1@gmail.com	

Terlihat data sudah berhasil masuk

14. Untuk menghentikan aplikasi gunakan command berikut:
docker compose down

```
root@ubuntu1:/home/ubuntu1/digital_skola/test-apps# docker compose down
[+] Running 5/5
✓ Container test-apps-frontend-1 Removed 1.9s
✓ Container test-apps-backend-1 Removed 0.8s
✓ Container test-apps-db-1 Removed 1.0s
✓ Network test-apps_default Removed 0.3s
✓ Network test-apps_net-test-apps Removed 0.5s
root@ubuntu1:/home/ubuntu1/digital_skola/test-apps# docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS      PORTS          NAMES
2bd825a9e191   backend-image  "/binary"                5 hours ago    Exited (2)  2 hours ago    ct-backend
18cdd1d5a1b6   frontend-image "/docker-entrypoint.s..." 19 hours ago   Exited (0)   18 hours ago    ct-frontend
8c6894091ece   postgres      "docker-entrypoint.s..." 3 days ago     Exited (0)   2 hours ago    ct-postgres01
root@ubuntu1:/home/ubuntu1/digital_skola/test-apps# git pull
```

Sekian Langkah-langkah menjalankan aplikasi sederhana menggunakan docker compose

Terima kasih