

## CARA MENJALANKAN PROGRAM

Pada aplikasi chatbot ini terdapat 2 proyek **yang ada** yaitu *frontend* (*vhiobot-platform*) *source code*: <https://github.com/bagusgandhi/vhiobot-platform.git> dan *backend* (*vhiobot-api*) *source code*: <https://github.com/bagusgandhi/vhiobot-api.git>. Untuk menjalankan program ini lakukan *download source code* tersebut lalu selanjutnya untuk menjalankan diperlukan docker sebagai *environment* yang dapat dijalankan di dalam *container*, database PostgreSQL dan Redis agar aplikasi dapat berjalan.

### 1. Langkah ke 1. Melakukan Instalasi Docker dan PostgreSQL

Pada langkah pertama ini diperlukan melakukan instalasi docker, berikut untuk melakukan instalasi docker. Untuk setup docker dapat menggunakan aplikasi docker desktop dengan cara melakukan *download docker desktop* pada link berikut: <https://docs.docker.com/desktop/>. Selanjutnya buka file tersebut dan jalankan installernya. Setelah docker terinstall selanjutnya, lakukan setup untuk menjalankan instalasi database menggunakan PostgreSQL. Buat sebuah file dengan nama **docker-compose.yml** lalu isikan dengan script berikut:

```
version: '3'
services:
  db:
    container_name: # nama container
    image: postgres
    restart: always
    environment:
      POSTGRES_DB: # nama database
      POSTGRES_USER: # user database
      POSTGRES_PASSWORD: # password database
    ports:
      - "5432:5432"
    volumes:
      - postgres_data:/var/lib/postgresql/data
volumes:
```

```
postgres_data:
```

selanjutnya isikan **container\_name**, **POSTGRES\_DB**, **POSTGRES\_USER**, dan **POSTGRES\_PASSWORD**. Apabila sudah diisi, selanjutnya ketikkan perintah berikut pada terminal

```
docker compose up -d -build
```

Perintah tersebut adalah untuk melakukan pembuatan container postgresQL pada docker. Jika sudah selesai ketikkan perintah berikut untuk melihat container postgresQL untuk memastikan sudah berjalan.

```
docker compose ps
```

Setelah database terinstall lakukan koneksi ke database tersebut untuk memastikan database sudah berjalan dan bisa dilakukan koneksi database. Untuk melakukan koneksi database memerlukan database client seperti DBEAVER yang dapat di download dan install pada URL berikut: <https://dbeaver.io/>

## 2. Langkah ke 2. Setup dialogflow platform dan google cloud

Terkait langkah tersebut sudah dipaparkan pada naskah skripsi di [Bab IV halaman 42](#) yang mana terdapat penjelasan mengenai *setup dialogflow* dan kredensialnya.

## 3. Langkah ke 3. Menjalankan program backend

Pada bagian backend yang perlu dilakukan copy file **.env.example** menjadi **.env** dan melakukan penyesuaian dengan koneksi database yang sudah dibuat. Pada bagian **DB\_NAME**, **DB\_USER**, dan **DB\_PASSWORD** bisa dimasukan kredensial database yang sudah dibuat pada step sebelumnya pada setup database.

```
PORT=3009
DB_HOST=127.0.0.1
```

```

DB_PORT=5432
DB_NAME=
DB_USER=
DB_PASSWORD=
PRIVATE_KEY_FILE=jwt/private.key
PUBLIC_KEY_FILE=jwt/public.key
...

```

Lalu langkah berikutnya yaitu perlu dilakukan pembuatan jwt private dan public key dengan perintah berikut, pastikan untuk file diletakan didalam folder jwt

### Pembuatan private.key

```
openssl genrsa -out private.key 2048
```

### Pembuatan public.key

```
openssl rsa -in private.key -pubout -out public.key
```

jika pada langkah *setup dialogflow platform* dan *google cloud* sudah dilakukan maka selanjutnya adalah melakukan *running program backend* dan *redis* dengan perintah berikut:

```
docker compose up -d -build
```

jika sudah selesai dalam proses *build & run* diatas, selanjutnya ketikkan perintah berikut untuk memastikan program backend dan redis sudah berjalan seperti berikut:

```
docker compose ps
```

Pastikan kedua *container* berikut berstatus *Up*, yang mana berarti program *backend* dan *redis* sudah berjalan (*running*)

```

NAME                IMAGE                COMMAND                SERVICE  CREATED      STATUS      PORTS
vhiobot-api         vhiobot-api-backend "docker-entrypoint.s..." backend  2 hours ago  Up 2 hours  0.0.0.0:80->80
vhiobot-redis       redis:7.4.1-alpine  "docker-entrypoint.s..." redis    2 months ago Up 3 hours  6379
(base) vhiolite@vhiolite-server:~/vhiobot-api$ █

```

#### 4. Langkah ke 4. Menjalankan program frontend

Setelah program backend sudah dipastikan dapat berjalan, langkah selanjutnya adalah menjalankan program *frontend*. Program ini adalah tampilan antarmuka untuk pada aplikasi vhiobot. Lakukan copy file `.env.example` menjadi `.env.local` lalu isikan informasi berikut agar bisa terkoneksi ke backend

```
NEXTAUTH_URL=http://localhost:3000
NEXTAUTH_SECRET=
GOOGLE_SECRET= # google secret
GOOGLE_CLIENT_ID= # google client id
VHIOBOT_API_HOST= # http://localhost:3009/api/v1
NEXT_PUBLIC_SOCKET_URL= # http://localhost:3009/
SECRET_API_KEY= # generate your key
NEXT_PUBLIC_API_URL=http://localhost:3000
NEXT_PUBLIC_BACKEND_URL= # http://localhost:3009/api/v1
```

jika bagian environment sudah dijalankan, selanjutnya ketikkan berikut pada bagian terminal *project frontend*

```
docker compose up -d -build
```

jika sudah selesai dalam proses *build & run* diatas, selanjutnya ketikkan perintah berikut untuk memastikan program *frontend* sudah berjalan seperti berikut:

```
docker compose ps
```

Pastikan *container* berikut berstatus **Up**, yang mana berarti program *frontend* sudah berjalan (*running*)

NAME	IMAGE	COMMAND	SERVICE	CREATED	STATUS
vhiobot-platform	vhiobot-platform-vhiobot-platform	"docker-entrypoint.s..."	vhiobot-platform	4 hours ago	up 3 hours