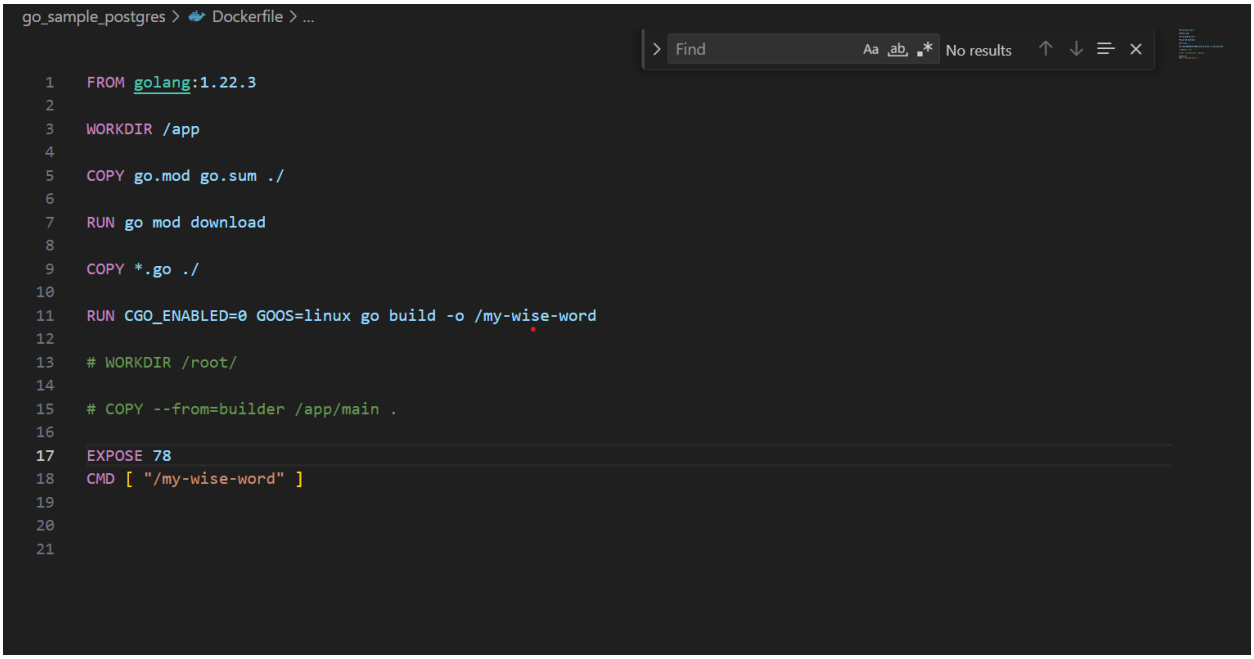
Task

program docker.file



main. go

```
package main

import (
    "database/sql"
    "fmt"
    "net/http"
    "os"

    "github.com/labstack/echo"
    "github.com/labstack/echo/middleware"
    _ "github.com/lib/pq"
)
```

```
func main() {

    // get env
    dbUser := os.Getenv("DB_USER")
    dbPass := os.Getenv("DB_PASS")
    dbHost := os.Getenv("DB_HOST")
    dbPort := os.Getenv("DB_PORT")
    dbName := os.Getenv("DB_NAME")

    //connect to database
    connStr := fmt.Sprintf("postgresql://%s:%s@%s:%s/%s? sslmode
    fmt.Println("connect to :", connStr)
    db, err := sql.Open("postgres", connStr)
    if err != nil {
        fmt.Println("error ")
    }

    //test ping to database
    if err := db.Ping(); err != nil {
        fmt.Printf("connestion to database failed (DB_HOST: %s:
    } else {
        fmt.Println("connestion to database success", db)
    }

    e := echo.New()
    fmt.Println("someone hit me!")

    e.Use(middleware.Logger())
    e.Use(middleware.Recover())

    e.GET("/", func(c echo.Context) error {
        return c.HTML(http.StatusOK, "bebas tapi brutal \n")
    })

    e.GET("/ping", func(c echo.Context) error {
```

```
        return c.String(http.StatusOK, "pong")
    })

    httpPort := os.Getenv("PORT")
    if httpPort == "" {
        httpPort = "78"
    }

    e.Logger.Fatal(e.Start(":" + httpPort))

}
```

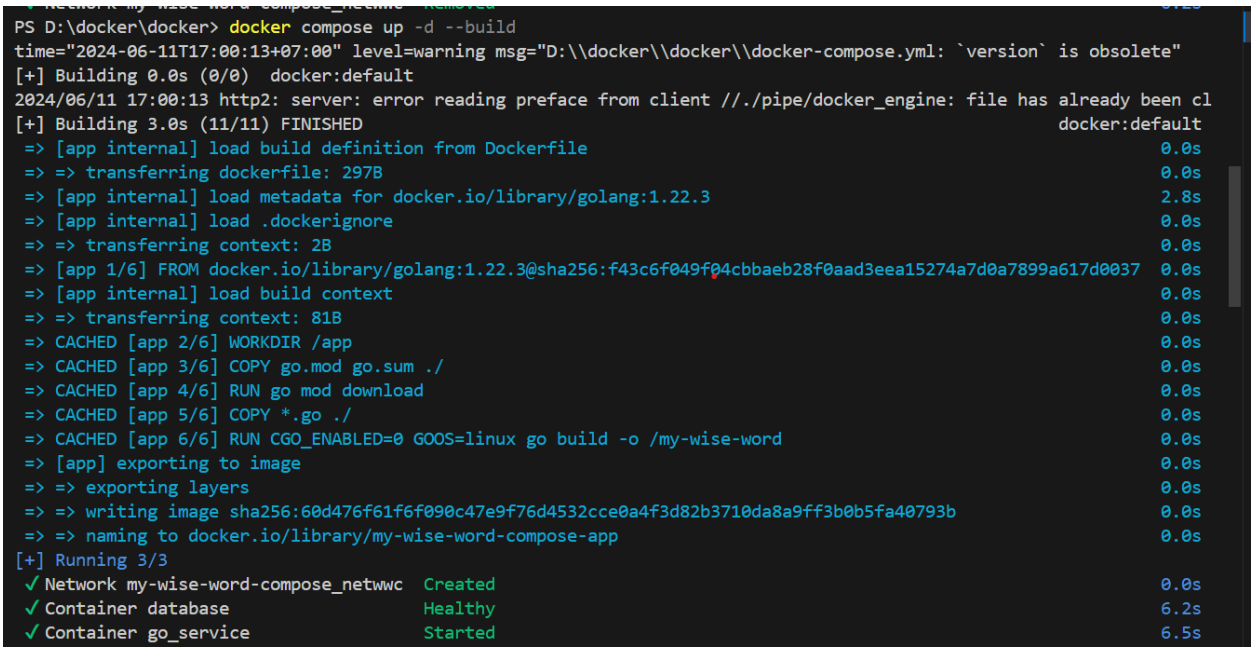
docker compose.yml

```
version: '3.9'

name: my-wise-word-compose

services:
  app:
    container_name: go_service
    environment:
      - DB_USER=${DB_USER}
      - DB_PASS=${DB_PASS}
      - DB_HOST=${DB_HOST}
      - DB_PORT=${DB_PORT}
      - DB_NAME=${DB_NAME}
    env_file:
      - .env
      # - name=value
    build:
```

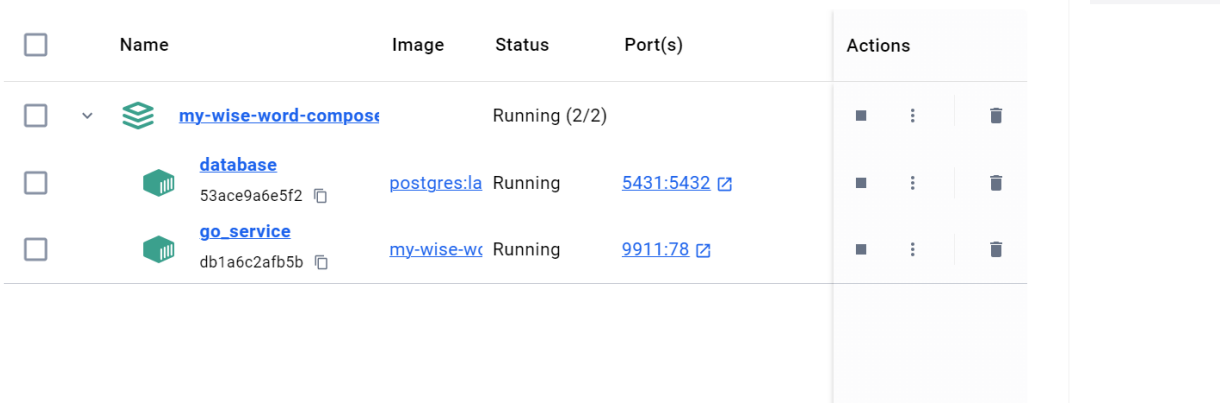
```
      context: ./go_sample_postgres/
      dockerfile: Dockerfile
    depends_on:
      postgresdb:
        condition: service_healthy
    ports:
      - 9911:78
    networks:
      - netwwc

  postgresdb:
    image: postgres:latest
    container_name: database
    environment:
      - POSTGRES_USER=${DB_USER}
      - POSTGRES_PASSWORD=${DB_PASS}
      - POSTGRES_DB=${DB_NAME}
      - DATABASE_HOST=${DB_HOST}
    env_file:
      - .env
    healthcheck:
      test: ["CMD-SHELL", "pg_isready -U postgres"]
      interval: 5s
      timeout: 5s
      retries: 5
    ports:
      - '5431:5432'
    volumes:
      - volwwc:/var/lib/postgresql/data
    networks:
      - netwwc

volumes:
  volwwc:
networks:
```

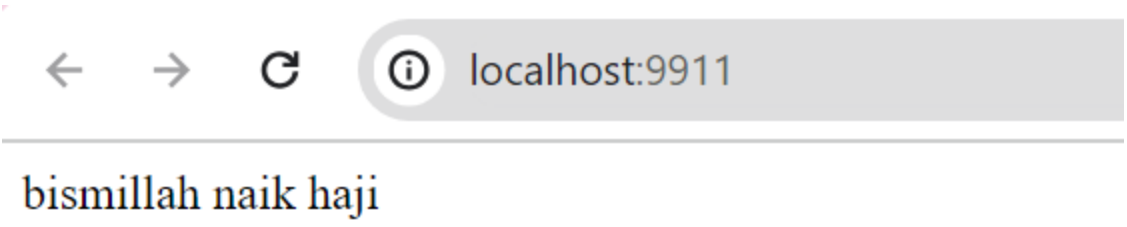
```
  netwwc:
    driver: bridge
```

build docker compose setelah program tidak ada errror

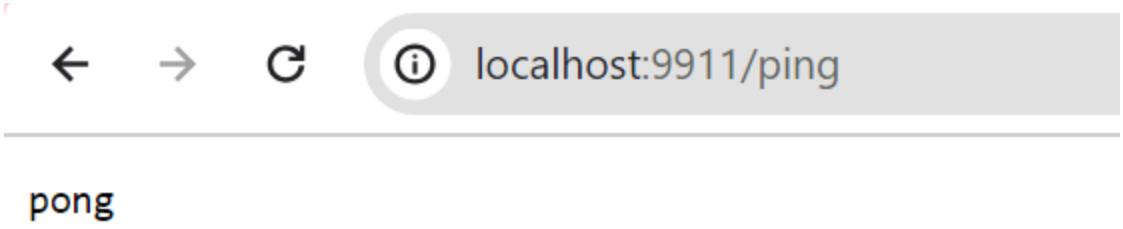


compose dan container

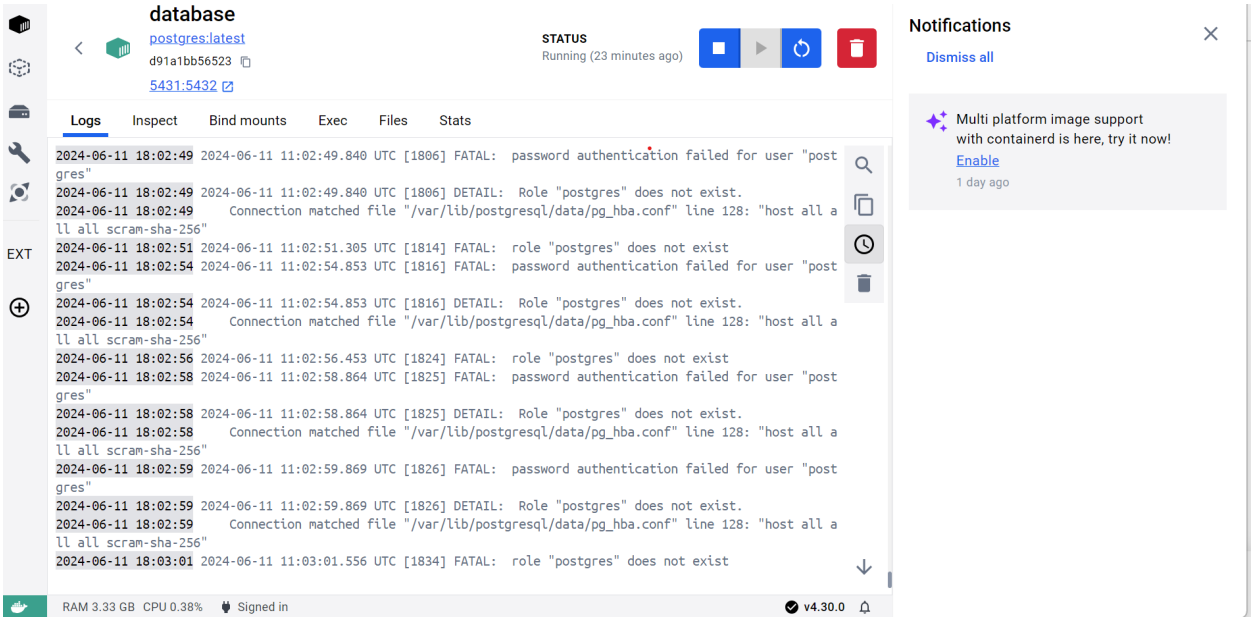




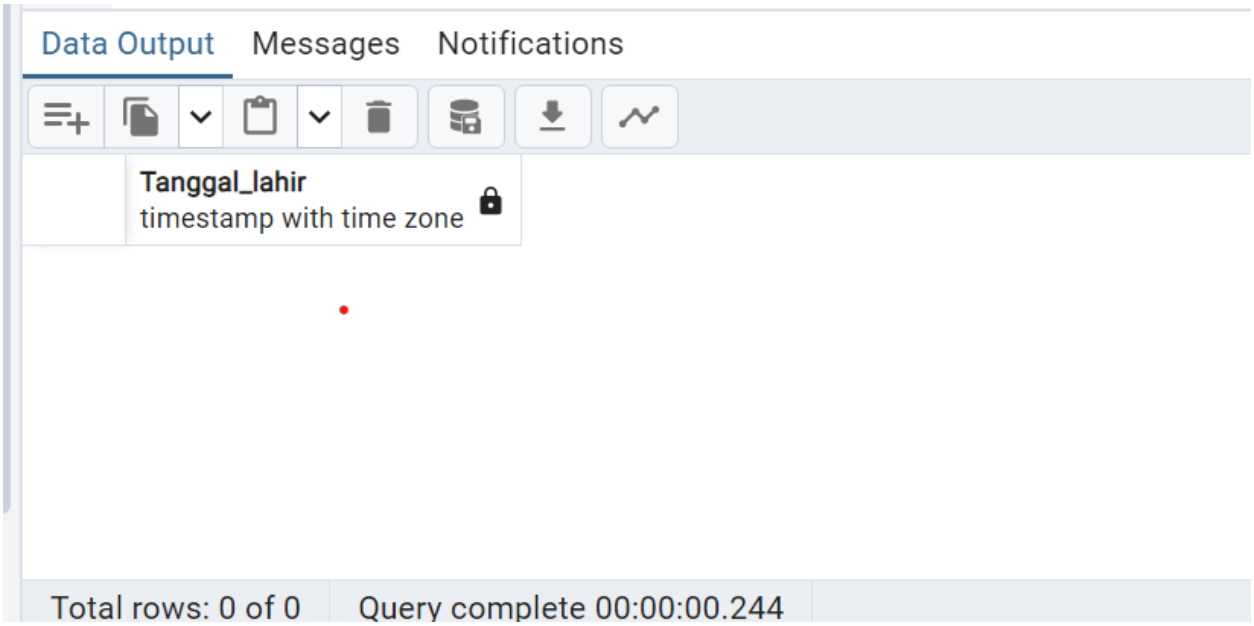
hasil menjalankan main lewat docker des



hasil ping



connect compose to database



database dengan fotmat typr timestamp with time zone