Pelabuhan

(TuT TuT TuT)

# Akses Cloud

Untuk services yang saya gunakan adalah *Microsoft azure*. Pake ini karena dapet packet student yang free. Berikut Langkah-langkah yang perlu dilakukan untuk claim(?)

1. Kunjungi URI ini <https://azure.microsoft.com/id-id/free/students>
2. Klik mulai gratis dan sesuaikan dengan akun klen.
3. Harusnya nanti nunggu konfirmasi dulu dan dapet deh azure student.

# Dockerize

Pertama-tama untuk website yang akan di-docker saya menggunakan website abal-abal yang pernah saya buat pas liburan, yaitu website ini <https://github.com/Indraswara/expense-tracker>. Struktur directory-nya kurang lebih begini

A screenshot of a computer

Description automatically generated

Nanti, ini akan dilakukan sedikit perubahan sehingga menjadi seperti ini.

A screenshot of a computer

Description automatically generated

Abaikan folder temp, karena itu buat docker-compose.yml untuk deploy nanti. Ketika sudah seperti ini kita bikin file “dockerfile” dan “.dockerignore” pada directory client (frontent) dan directory backend.

Dockerfile client

|  |
| --- |
| # client/Dockerfile  #membuat image dari node versi 20.15.0  FROM node:20.15.0  #membuat direktori /app  WORKDIR /app  #copy package.json dan package-lock.json ke direktori /app  COPY package\*.json ./  #install dependensi  RUN npm install  #copy semua file ke direktori /app  COPY . .  #expose port 3000 untuk mengakses aplikasi  EXPOSE 3000  #menjalankan webapp  CMD ["npm", "start"] |

Dockerfile backend

|  |
| --- |
| # backend/Dockerfile  #ini untuk membuat image dari node versi 20.15.0  FROM node:20.15.0  #membuat direktori /app  WORKDIR /app  #copy package.json dan package-lock.json ke direktori /app  COPY package\*.json ./  #install dependensi  RUN npm install  #copy semua file ke direktori /app  COPY . .  #expose port 5000 untuk mengakses aplikasi  EXPOSE 5000  #menjalankan server.js  CMD ["node", "server.js"] |

.Dockerignore untuk client dan backend isinya sama, yaitu hanya untuk mengabaikan directory node\_modules.

.Dockerignore

|  |
| --- |
| node\_modules/ |

Lalu, kita buat docker-compose.yaml pada root directory untuk menjalankan 2 container frontend dan backend secara bersamaan.

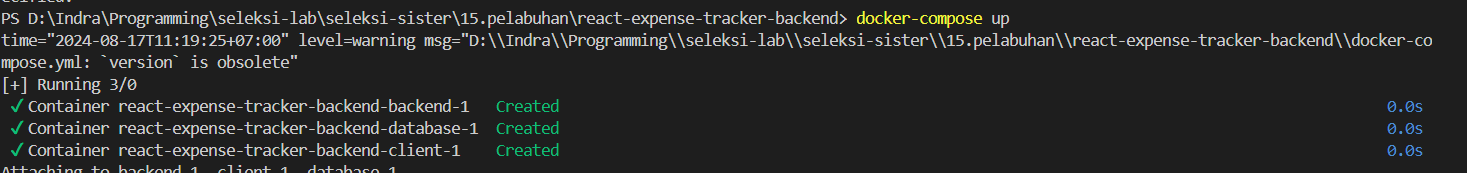
Docker-compose.yml

|  |
| --- |
| version: '3.8'  services:  client:  build:  context: ./client  dockerfile: Dockerfile  ports:  - "3000:3000"  environment:  - REACT\_APP\_BACKEND\_URL=http://backend:5000  depends\_on:  - backend  backend:  build:  context: ./backend  dockerfile: Dockerfile  ports:  - "5000:5000"  environment:  - MONGO\_URI=mongodb+srv://indra:indra@wallet.ui1aiah.mongodb.net/expense\_tracker?retryWrites=true&w=majority&appName=Wallet  database:  image: mongo:latest  ports:  - "27017:27017"  volumes:  - mongo-data:/data/db  volumes:  mongo-data: |

Kayanya docker ini 3 tier deh mungkin bonus? Gatau aku sendiri juga gayakin.

Setelah selesai kita bisa menjalankannya dengan command berikut pada terminal. Kalo di Windows pastikan kalian membuka aplikasi docker desktop terlebih dahulu

* Docker-compose build (untuk build container-nya)
* Docker-compose up (untuk menjalankan container-nya sehingga bisa webapp bisa diakses melalui localhost)



A screenshot of a computer

Description automatically generated

* Docker-compose down (untuk menonaktfikan container yang sedang menyala)

# Deployment

## Access Azure

Deployment disini saya gunakan azure karena dapat azure for students. Buat dapetinnya tinggal kunjungi aja website ini

<https://azure.microsoft.com/id-id/free/students>

dan ikuti prosedurnya, nanti kalau udah selesai di beranda Microsoft azure bagian subscriptions kitab isa melihat subscriptions yang sudah kita dapat

A screenshot of a computer

Description automatically generated

A screenshot of a video game

Description automatically generated

## Step By Step

### Bikin ACR (Azure Container Registry)

Bagian ini kita akan membuat ACR terlebih dahulu yang nantinya isinya adalah image dari webapp yang sudah dibuat, nantinya akan ada tiga, yaitu frontend, backend, dan database. Cara bikinnya

1. Klik Create Resource
2. Pilih Container Registry

A screenshot of a computer

Description automatically generated

1. Sisanya ini bagian Basics aja terus langsung review and create

A screenshot of a computer

Description automatically generated

1. Kalo udah di-deploy harusnya nanti kita bisa memiliki container registry seperti berikut

A screenshot of a computer

Description automatically generated

1. Nanti kita bisa push image kita ke registry ini dengan mengikuti prosedur “push a container image”. Hasil dari push tadi bisa kita lihat di bagian repositories. Pada gambar di bawah terlihat 3 repositories yang masing-masing adalah container image untuk frontend, backend, dan database(mongo).

A screenshot of a computer

Description automatically generated

1. Bagian ini sudah selesai dan kita bisa lanjut untuk deploy webapp-nya

### Deploy Webapp

1. Pilih create resource pada home

A screenshot of a computer

Description automatically generated

1. Pilih categories containers dan pilih “web app for containers”

A screenshot of a computer

Description automatically generated

1. Bagian basics bisa diisi sesuai kebutuhan dan jangan lupa bagian publish pilih container

A screenshot of a computer

Description automatically generated

1. Selanjutnya ke bagian docker, nanti disini pilih docker compose (karena container lebih dari satu). Image Resource pilih ACR, Registrynya pilih registry yang kita pakai untuk menyimpan image di awal tadi. Terakhir, untuk configuration masukkan docker-compose.yml

A screenshot of a computer

Description automatically generated

Bagian docker.compose.yml berbeda dari docker.compose.yml pada dockerize disini berbeda

Berikut codenya,

|  |
| --- |
| version: '3.8'  services:  client:  image: pelabuhan.azurecr.io/client:latest  ports:  - "3000:3000"  environment:  - REACT\_APP\_BACKEND\_URL=http://backend:5000  - DANGEROUSLY\_DISABLE\_HOST\_CHECK=true  depends\_on:  - backend  backend:  image: pelabuhan.azurecr.io/backend:latest  ports:  - "5000:5000"  environment:  - MONGO\_URI=mongodb+srv://indra:indra@wallet.ui1aiah.mongodb.net/expense\_tracker?retryWrites=true&w=majority&appName=Wallet  database:  image: pelabuhan.azurecr.io/mongo:latest  ports:  - "27017:27017"  volumes:  - mongo-data:/data/db  volumes:  mongo-data: |

Bagian image berbeda, yang kita masukkan adalah link untuk ke repositories yang ada di dalam registry.

1. Bagian Container isi sama seperti Docker
2. Terakhir kita bisa langsung review + create dan tunggu hasilnya.
3. Nah, untuk punya saya ini hasil deploy-nya

<https://pelabuhan-sister-b8erdqahcdahb8gw.southeastasia-01.azurewebsites.net/>

A screenshot of a computer

Description automatically generated

1. Dan selesai (gweh habis 5-6 jam buat deploy ini anjim)