**LAPORAN PRAKTIKUM**

**Docker Flask Web Peminjaman**



Disusun Oleh:

Nama : 1. Muhammad Ajra Zemima Muda

2. L Hafidl Alkhair

Kelas : TRKJ 2.C

Jurusan : Teknologi Informasi dan Komputer

Program Studi : Teknologi Rekayasa Komputer Jaringan

Dosen Pembimbing : Indrawati, SST., M.T.



**JURUSAN TEKNOLOGI INFORMASI KOMPUTER**

**PRODI TEKNOLOGI REKAYASA KOMPUTER JARINGAN**

**POLITEKNIK NEGERI LHOKSEUMAWE**

**TAHUN AJARAN 2024-202**

# LEMBAR PENGESAHAN

No Praktikum : 02/TIK/TRKJ-2C/Komputasi Awan

Judul : Laporan Praktikum Docker Flask Web Peminjaman

Nama :1. Muhammad Ajra Zemima Muda

2. L Hafidl Alkhair

Kelas : TRKJ 2.C

Jurusan : Teknologi Informasi dan Komputer

Prodi : Teknologi Rekayasa Komputer Jaringan

Tanggal Praktikum : Jum’at 15 Mei

Tanggal Penyerahan : Jum’at, 21 Juni 2025

|  |
| --- |
| Buketrata, 21 Juni 2025 |
| Dosen Pembimbing, |
|  |
|  |
|  |
| **Indrawati, SST., M.T.** |
| NIP. 19740815 200112 2 001 |

# DAFTAR ISI

[LEMBAR PENGESAHAN i](#_Toc201355986)

[DAFTAR ISI ii](#_Toc201355987)

[DAFTAR GAMBAR iii](#_Toc201355988)

[BAB I 1](#_Toc201355989)

[A. Dasar Teori 1](#_Toc201355990)

[B. Alat dan bahan 2](#_Toc201355991)

[BAB II 3](#_Toc201355992)

[A. Clone Program 3](#_Toc201355993)

[1. Buka github *https://github.com/hafidalkhair/app-peminjaman* 3](#_Toc201355994)

[2. Salin github 3](#_Toc201355995)

[3. Buka cmd 5](#_Toc201355996)

[4. cd .\app-peminjaman\ 5](#_Toc201355997)

[B. Folder proyek app peminjaman. 5](#_Toc201355998)

[C. Membuat database 6](#_Toc201355999)

[E. Membangun docker dalam program app peminjaman 7](#_Toc201356000)

[1. Buat Dockerfile 7](#_Toc201356001)

[2. Membuat requirements.txt 7](#_Toc201356002)

[3. Kode program yang terdapat pada file Dockerfile. 8](#_Toc201356003)

[4. Penjelasan kode program 8](#_Toc201356004)

[5. Build container 10](#_Toc201356005)

[6. Jalankan docker dan ngrok 12](#_Toc201356006)

[BAB III 16](#_Toc201356007)

[A. Kesimpulan 16](#_Toc201356008)

# DAFTAR GAMBAR

[Gambar 1 1 tampilan github 3](#_Toc201355951)

[Gambar 1 1 tampilan github 3](#_Toc201355952)

[Gambar 1 2 copy url github 3](#_Toc201355953)

[Gambar 1 2 copy url github 3](#_Toc201355954)

[Gambar 1 3 git clone git hub di terminal 5](#_Toc201355955)

[Gambar 1 3 git clone git hub di terminal 5](#_Toc201355956)

[Gambar 1 4 masuk direktori app-peminjaman 5](#_Toc201355957)

[Gambar 1 5 sturktur folder 5](#_Toc201355958)

[Gambar 1 5 sturktur folder 5](#_Toc201355959)

[Gambar 1 6 databases 6](#_Toc201355960)

[Gambar 1 6 databases 6](#_Toc201355961)

[Gambar 1 7 Membuat Dockerfile didalam Program app-peminjaman 7](#_Toc201355962)

[Gambar 1 7 Membuat Dockerfile didalam Program app-peminjaman 7](#_Toc201355963)

[Gambar 1 8 tampilan docker container yang kosong 10](#_Toc201355964)

[Gambar 1 8 tampilan docker container yang kosong 10](#_Toc201355965)

[Gambar 1 9 docker berhasil di buat 10](#_Toc201355966)

[Gambar 1 9 docker berhasil di buat 10](#_Toc201355967)

[Gambar 1 10 tampilan docker images yang telah dibuat 10](#_Toc201355968)

[Gambar 1 10 tampilan docker images yang telah dibuat 10](#_Toc201355969)

[Gambar 1 11 container id untuk container yang sedang berjalan. 11](#_Toc201355970)

[Gambar 1 11 container id untuk container yang sedang berjalan. 11](#_Toc201355971)

[Gambar 1 12 tampilan app setelah docker dijalan kan 12](#_Toc201355972)

[Gambar 1 12 tampilan app setelah docker dijalan kan 12](#_Toc201355973)

[Gambar 1 13 intall ngrok 12](#_Toc201355974)

[Gambar 1 13 intall ngrok 12](#_Toc201355975)

[Gambar 1 14 auth token ngrok 13](#_Toc201355976)

[Gambar 1 14 auth token ngrok 13](#_Toc201355977)

[Gambar 1 15 ip ngrok 13](#_Toc201355978)

[Gambar 1 15 ip ngrok 13](#_Toc201355979)

[Gambar 1 16 ngrok run 13](#_Toc201355980)

[Gambar 1 16 ngrok run 13](#_Toc201355981)

[Gambar 1 17 link dibagikan ngrok 14](#_Toc201355982)

[Gambar 1 17 link dibagikan ngrok 14](#_Toc201355983)

[Gambar 1 18 tampilan app peminjaman 15](#_Toc201355984)

[Gambar 1 18 tampilan app peminjaman 15](#_Toc201355985)

# BAB I

**PENDAHULUAN**

1. **Dasar Teori**

Docker adalah sebuah platform open-source yang digunakan untuk mengembangkan, mengirim, dan menjalankan aplikasi dalam sebuah container. Container adalah unit standar perangkat lunak yang mengemas kode dan semua dependensinya sehingga aplikasi dapat berjalan secara konsisten di berbagai lingkungan.

Menurut dokumentasi resmi Docker:

"*Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly*." (Docker Docs, 2024)

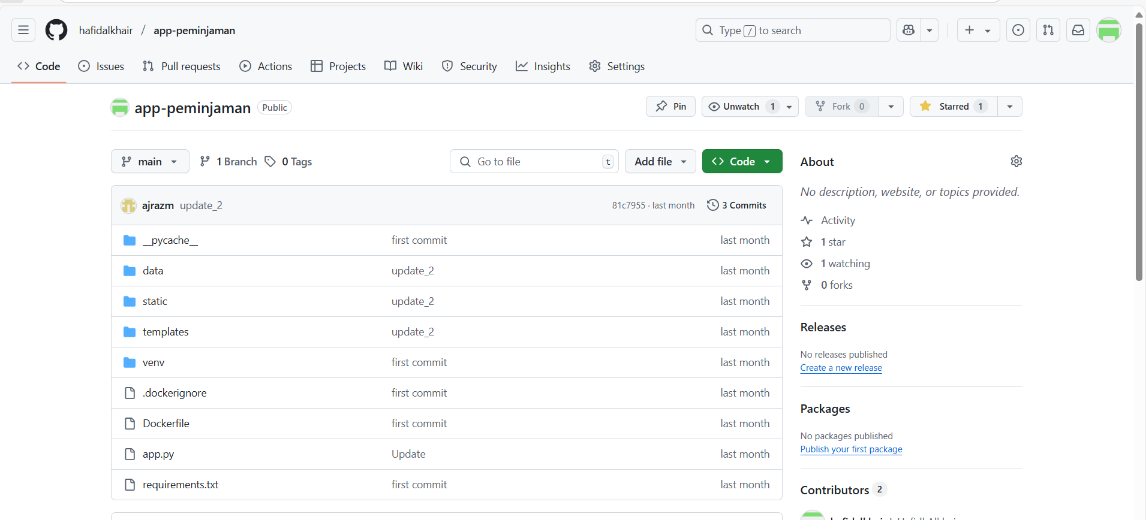
“Docker adalah platform terbuka untuk mengembangkan, mendistribusikan, dan menjalankan aplikasi. Docker memungkinkan Anda untuk memisahkan aplikasi Anda dari infrastruktur sehingga Anda dapat mengirimkan perangkat lunak dengan lebih cepat."

Aplikasi peminjaman barang merupakan sistem informasi berbasis web yang dirancang untuk mempermudah proses pencatatan dan pengelolaan peminjaman barang. Dengan menggunakan bahasa pemrograman Python dan framework Flask, aplikasi ini memungkinkan pengguna untuk melakukan login, registrasi, dan mengajukan peminjaman barang secara daring. Selain itu, Docker digunakan untuk mempermudah proses deployment dengan lingkungan yang terisolasi dan konsisten.

1. **Alat dan bahan**
2. Bahasa Pemrograman: Python 3.x
3. Framework: Flask
4. Database: SQLite
5. Docker
6. Editor Kode: Visual Studio Code
7. Browser: Google Chrome atau Mozilla Firefox
8. Sistem Operasi: Windows/Linux

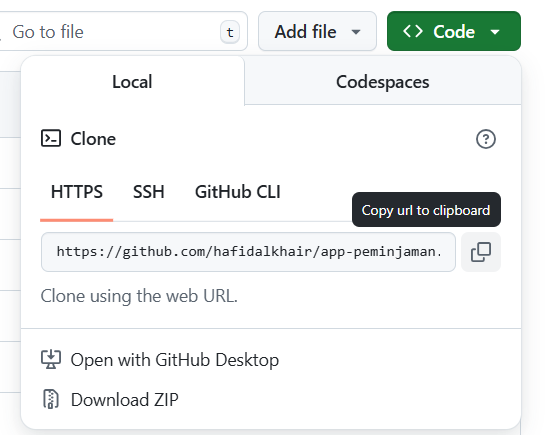
# BAB II

**LANGKAH PRAKTIKUM**

1. **Clone Program**
2. **Buka github** [*https://github.com/hafidalkhair/app-peminjaman*](https://github.com/hafidalkhair/app-peminjaman)

Gambar 1 1 tampilan github

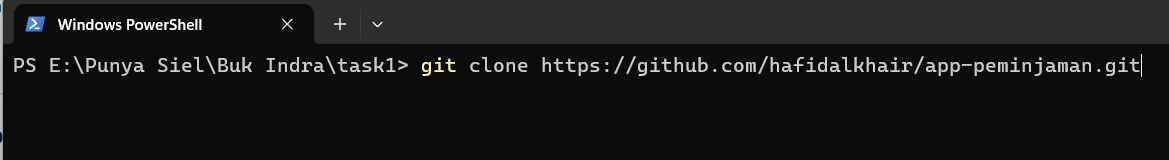
Gambar 1 1 tampilan github

1. **Salin github**

Gambar 1 2 copy url github

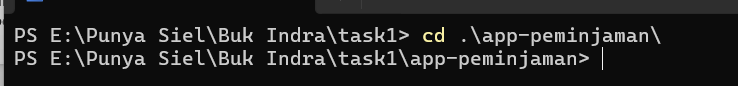
Gambar 1 2 copy url github

1. **Buka cmd**

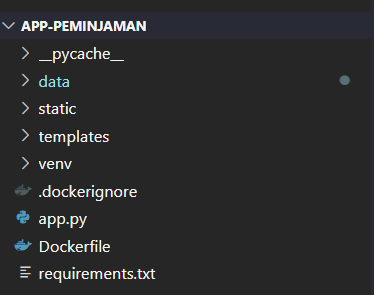
****

Gambar 1 3 git clone git hub di terminal

Gambar 1 3 git clone git hub di terminal

1. **cd .\app-peminjaman\**

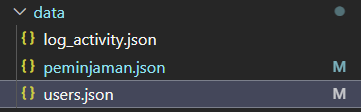
Gambar 1 4 masuk direktori app-peminjaman

1. **Folder proyek app peminjaman.**

Gambar 1 5 sturktur folder

Gambar 1 5 sturktur folder

1. **Membuat database**

Untuk database nya itu sudah dibuat otomatis oleh app.py nya sendiri dan akan disimpan dalam data

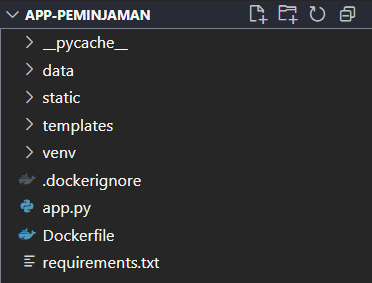
Gambar 1 6 databases

Gambar 1 6 databases

1. **Membangun docker dalam program app peminjaman**

Pada kasus ini, kami menggunakan situs web peminjaman yang telah kami buat bersama. Untuk memasukkan program ke dalam container Docker, ikuti langkah-langkah berikut:

1. Buat Dockerfile

Buat dockerfile didalam program web peminjaman

Gambar 1 7 Membuat Dockerfile didalam Program app-peminjaman

Gambar 1 7 Membuat Dockerfile didalam Program app-peminjaman

1. Membuat requirements.txt

Flask

Flask-Login

Werkzeug

1. Kode program yang terdapat pada file Dockerfile.

# Gunakan image dasar Python

FROM python:3.9-slim-buster

# Set working directory di dalam container

WORKDIR /app

# Salin requirements.txt dan instal dependensi

COPY requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt

# Salin seluruh kode aplikasi

COPY . .

# Set variabel lingkungan

ENV FLASK\_APP=app.py

ENV FLASK\_ENV=production

# Port yang diekspos

EXPOSE 5000

# Perintah untuk menjalankan aplikasi

CMD ["flask", "run", "--host=0.0.0.0"]

1. Penjelasan kode program
2. *FROM python:3.9-slim-buster*

Menggunakan image dasar Python 3.9 yang ringan (slim) sebagai fondasi untuk aplikasi.

1. *WORKDIR /app*

Menentukan direktori kerja di dalam container. Semua perintah selanjutnya akan dijalankan di dalam direktori ini.

1. *COPY requirements.txt .*
2. *RUN pip install --no-cache-dir -r requirements.txt*

Menyalin file requirements.txt ke dalam container dan menginstal semua dependensi yang diperlukan untuk aplikasi Flask.

1. *COPY . .*

Menyalin seluruh kode aplikasi dari direktori lokal ke dalam container.

1. *ENV FLASK\_APP=app.py*
2. *ENV FLASK\_ENV=production*

Mengatur variabel lingkungan yang diperlukan oleh Flask. FLASK\_APP menunjukkan file utama aplikasi, sedangkan FLASK\_ENV diatur ke production untuk mode produksi.

1. *EXPOSE 5000*

Mengekspos port 5000, yang merupakan port default untuk aplikasi Flask.

1. *CMD ["flask", "run", "--host=0.0.0.0"]*

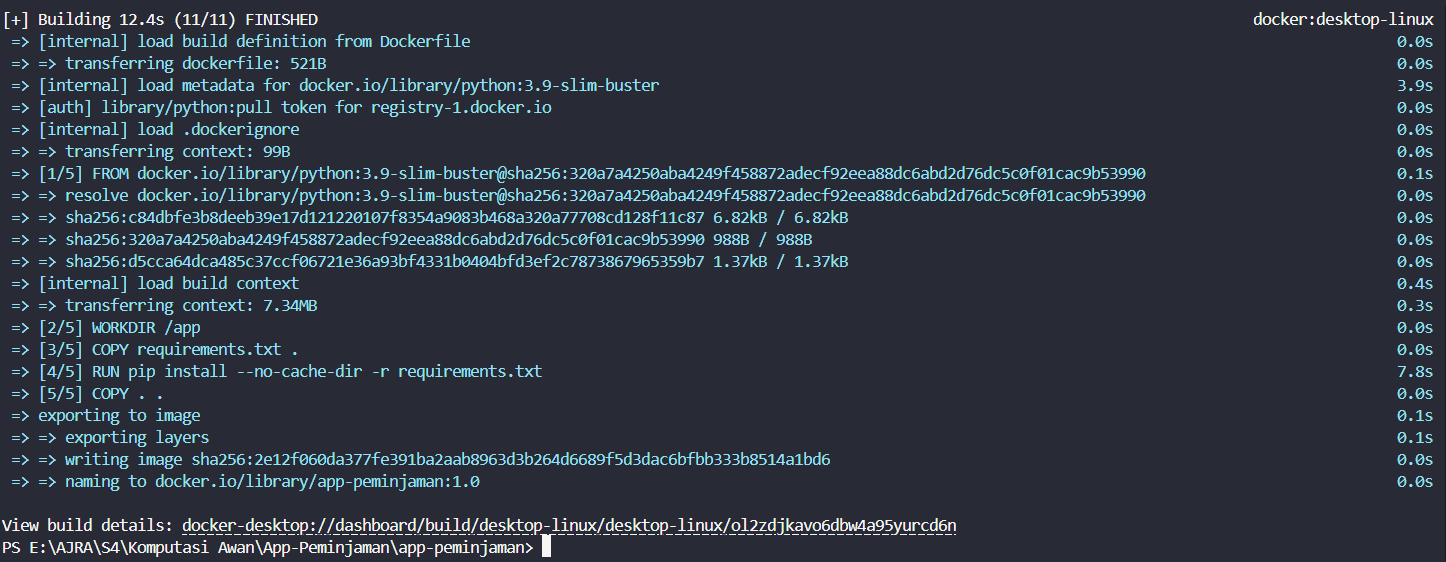
Perintah ini menjalankan aplikasi Flask ketika container dimulai, membuatnya tersedia untuk menerima permintaan dari luar.

1. Build container
2. docker ps

Untuk melihat daftar container Docker yang sedang berjalan.

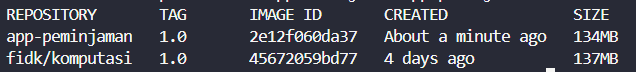
Gambar 1 8 tampilan docker container yang kosong

Gambar 1 8 tampilan docker container yang kosong

1. docker build -t app-peminjaman:1.0 .

Gambar 1 9 docker berhasil di buat

Gambar 1 9 docker berhasil di buat

1. docker images

Gambar 1 10 tampilan docker images yang telah dibuat

Gambar 1 10 tampilan docker images yang telah dibuat

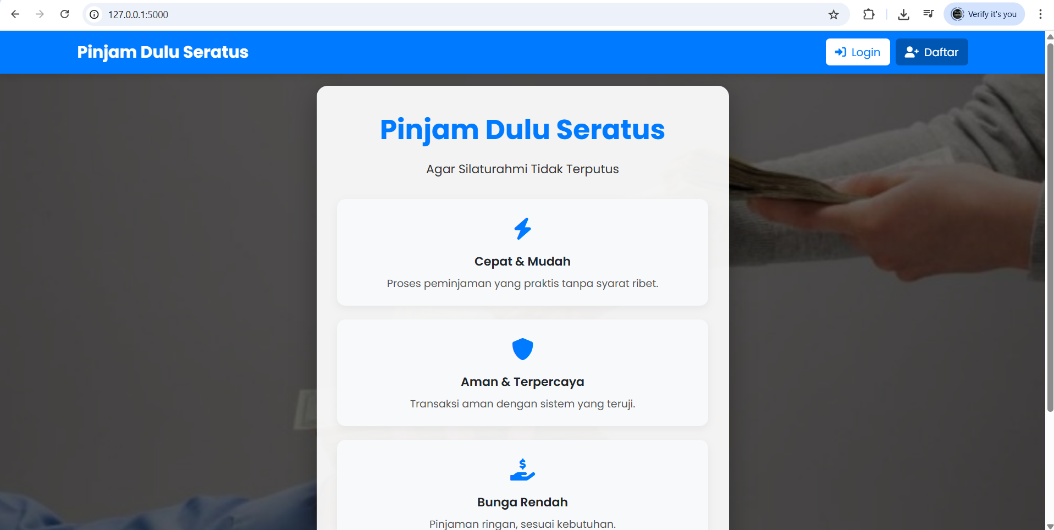
1. docker run -d -p 5000:5000 app-peminjaman:1.0



Gambar 1 11 container id untuk container yang sedang berjalan.

Gambar 1 11 container id untuk container yang sedang berjalan.

1. tampilan app untuk 127.0.0.1:5000



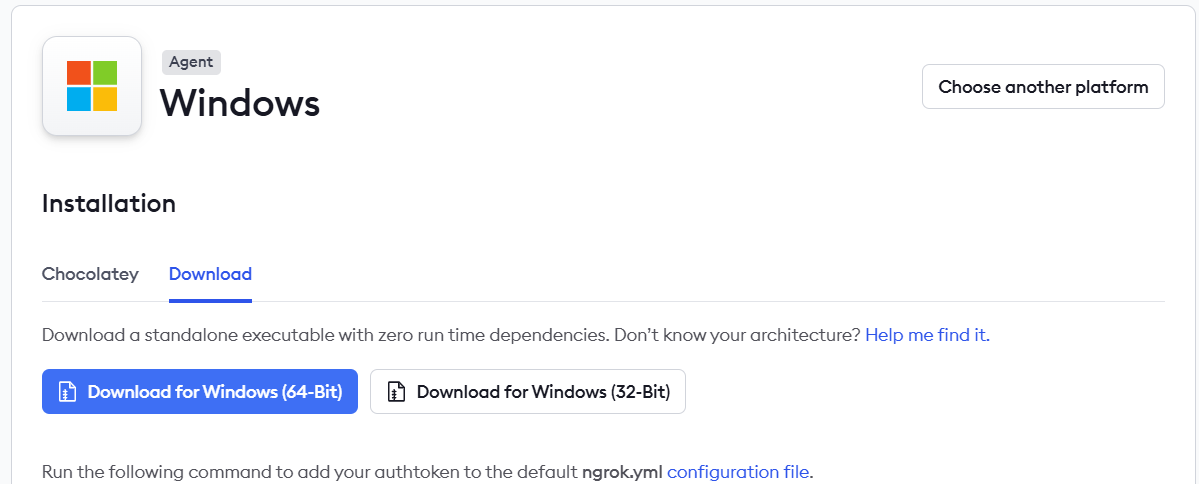
Gambar 1 12 tampilan app setelah docker dijalan kan

Gambar 1 12 tampilan app setelah docker dijalan kan

1. Jalankan docker dan ngrok

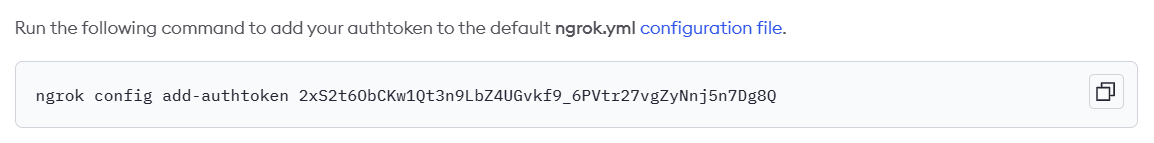
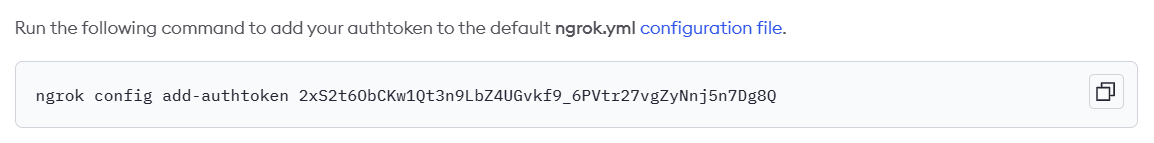
Setelah docker dijalankan, install terlebih dahulu ngrok nya

1. Install ngrok windows/linux

Pergi ke web resmi ngrok kemudian login dan ikuti documentasi untuk download nya

Gambar 1 13 intall ngrok

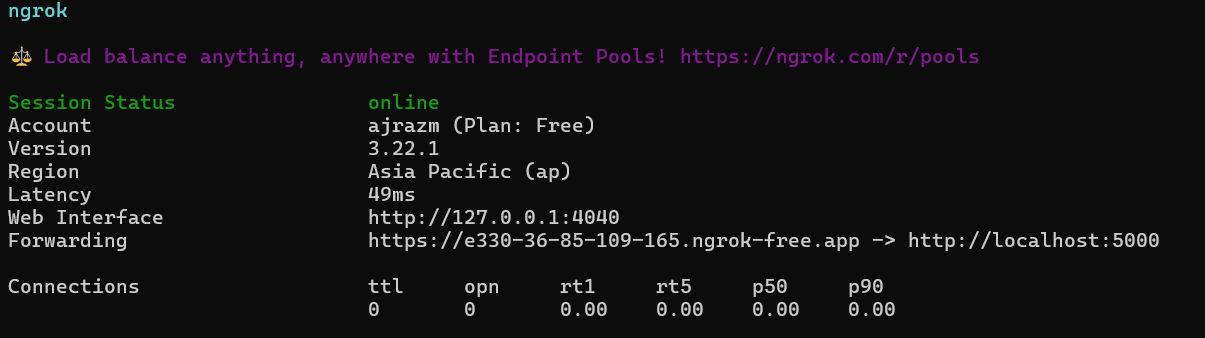
Gambar 1 13 intall ngrok

1. Setelah install
2. jalankan ngrok

Gambar 1 14 auth token ngrok

Gambar 1 14 auth token ngrok



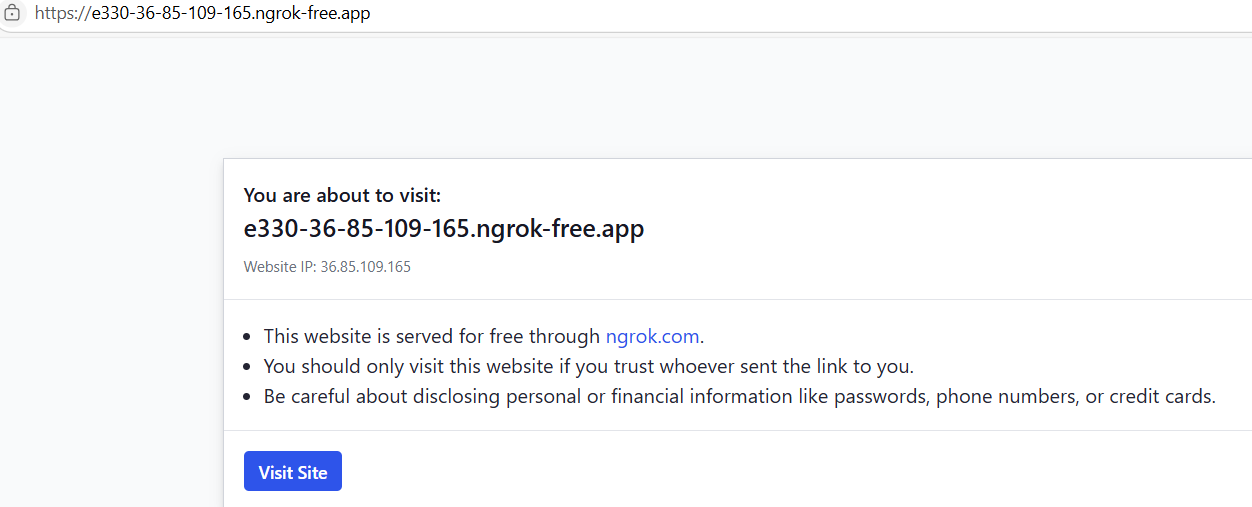
untuk portnya harus di sesuaikan dengan docker setelah itu maka akan diberikan ip

Gambar 1 15 ip ngrok

Gambar 1 15 ip ngrok

Gambar 1 16 ngrok run

Gambar 1 16 ngrok run

klik ip tersebut nanti akan di arahkan kesebuah link

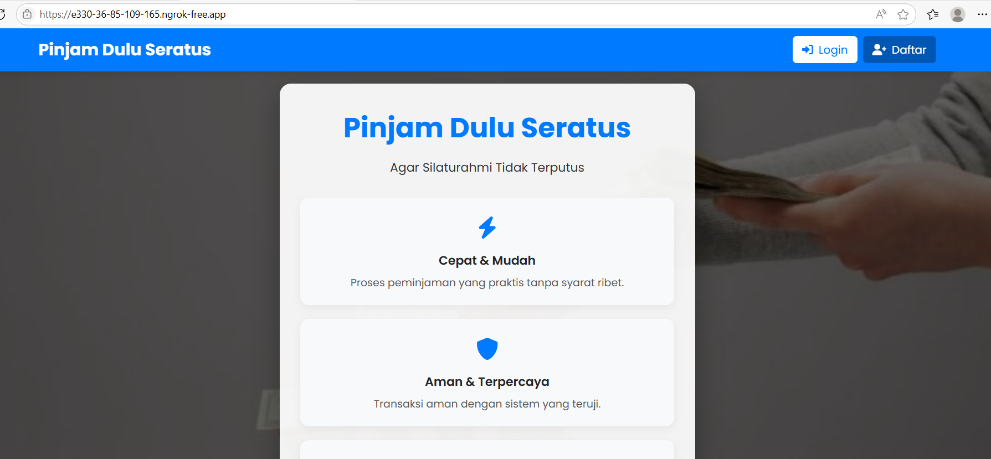
Gambar 1 17 link dibagikan ngrok

Gambar 1 17 link dibagikan ngrok

visit site kemudian akan tampil app-peminjaman yang telah dibuat

Gambar 1 18 tampilan app peminjaman

Gambar 1 18 tampilan app peminjaman



link tesebut dapat dibagikan dan digunakan juga oleh pemilik link tersebut.

# BAB III

**PENUTUP**

1. **Kesimpulan**

Praktikum pembuatan aplikasi peminjaman berbasis web dengan Flask dan Docker ini memberikan pemahaman mendalam tentang bagaimana mengembangkan dan mengelola aplikasi dengan lingkungan yang terisolasi serta mudah untuk didistribusikan. Aplikasi ini berhasil dibangun dengan fitur lengkap, seperti login, registrasi, verifikasi biodata, form peminjaman, serta dashboard untuk pengguna dan admin.

Dengan memanfaatkan Docker, aplikasi dapat dijalankan secara konsisten di berbagai sistem operasi tanpa harus mengatur lingkungan secara manual. Praktikum ini juga memberikan pengalaman nyata dalam pengelolaan data menggunakan file JSON dan penggunaan template HTML untuk antarmuka pengguna.

Secara keseluruhan, praktikum ini memperkuat keterampilan teknis dalam pengembangan web dan DevOps dasar, serta menumbuhkan pemahaman tentang arsitektur aplikasi yang modular dan efisien.