

Nama: Jowna Alynsah

NIM: 51190331

Kelas: T

Dosen: Pak Fikri Fadillah

Mata Kuliah: Web Technology 3

A. Gambaran Umum Proyek

Nama Website: Binbi

Bidang: Akomodasi, Travel

Layanan: Penyewaan tempat akomodasi, penginapan, penyewaan rumah atau apartemen

B. Penjelasan Proses Bisnis Proyek

Situs web Binbi akan menyediakan layanan-layanan yang terinspirasi oleh aplikasi AirBnB yakni layanan yang mengizinkan para pengguna untuk menyewa ataupun menyewakan rumah, apartemen, dan akomodasi lainnya.

Tampilan yang dikembangkan, bersifat prototipe dan masih dalam pengembangan karena pada proyek ini difokuskan pada CRUD dalam sisi *backend*.

CATATAN: Jika ada masalah dengan CORS pada sisi front-end maka silahkan instalasi ekstensi tersebut pada browser Anda:

<https://chrome.google.com/webstore/detail/who-cors/hnlmanpjeomjnpdglldcnpkppmndbp>

Berikut adalah proses bisnis serta rangkaian fitur-fitur dalam tampilan **user**:

1. **Registrasi dan login:** pengguna diperlukan untuk membuat akun melalui halaman registrasi jika belum memiliki akun. Jika sudah memiliki akun, pengguna dapat mengakses *dashboard* dengan memasukkan *username* dan *password* untuk melakukan *login*.
2. **Katalog tempat (Places):** setelah melakukan *login*, pengguna akan masuk ke tampilan *dashboard* yang berisi katalog tempat-tempat yang disewakan serta menambahkan tempat sendiri yang ingin disewa. Disini pengguna dapat melakukan *searching* atau pencarian dengan nama tempat atau lokasi tempat.
3. **Pembayaran (Order Confirmation):** ketika pengguna memilih untuk menyewa sebuah tempat maka pengguna dapat melihat detail transaksi tersebut sebelum melakukan *checkout*. Setelah melakukan *checkout* maka pengguna dapat melihat total biaya yang harus dibayar serta memilih metode pembayaran.

4. **Histori Transaksi (Transactions):** setelah melakukan pembayaran, pengguna dapat melihat histori transaksi dengan mengklik *link* yang tersedia di *navigation bar*. Hanya akan menampilkan transaksi pengguna itu sendiri
5. **Pemesanan (Orders):** pengguna juga dapat menggunakan platform untuk menyewa sehingga jika ada pengguna lain yang menyewa akan masuk dalam tampilan *orders*.
6. **Profil:** pengguna dapat melihat profil diri sendiri seperti *username*, nama depan, nama belakang, dan lain-lain
7. **Logout:** navigasi ke halaman *login*

Berikut adalah rangkaian fitur-fitur untuk tampilan sisi **admin**:

1. **Login:** untuk mengakses tampilan admin maka diperlukan *username* dan *password* sebagai berikut:

Username: jowna
Password: admin123
2. **Katalog tempat (Places):** sama seperti akun pengguna admin dapat melihat katalog tempat namun juga dapat menghapus dan memodifikasinya.
3. **Daftar Pengguna (Users):** admin dapat melihat daftar pengguna yang telah membuat akun
4. **Logout:** navigasi ke halaman *login*

C. Bahasa dan Alat yang digunakan

Frontend:

- Kombinasi Javascript dan HTML
- ReactJS sebagai framework
- Package-package ReactJS, seperti:
 - o React-bootstrap
 - o React-bootstrap-table2
 - o React-router

Backend:

- Bahasa Go (Go Language) sebagai bahasa utama sisi *backend*
- Mariadb & MySQL sebagai basis data
- Docker sebagai platform untuk menjalankan basis data
- DBeaver sebagai *Database Management System (DBMS)*

D. Setup Backend

1. Basis Data (Maria DB)

Berikut adalah instruksi *docker run* yang saya gunakan:

```
docker run --hostname=87a72baea90e --mac-address=02:42:ac:11:00:03 --  
env=MARIADB_USER=example-user --env=MARIADB_PASSWORD=incorrect123 --  
env=MARIADB_ROOT_PASSWORD=incorrect123 --  
env=PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin --  
env=GOSU_VERSION=1.14 --env=MARIADB_VERSION=1:10.9.2+maria~ubu2204 --  
volume=/var/lib/mysql -p 3306:3306 --restart=no --runtime=runc -d mariadb:latest
```

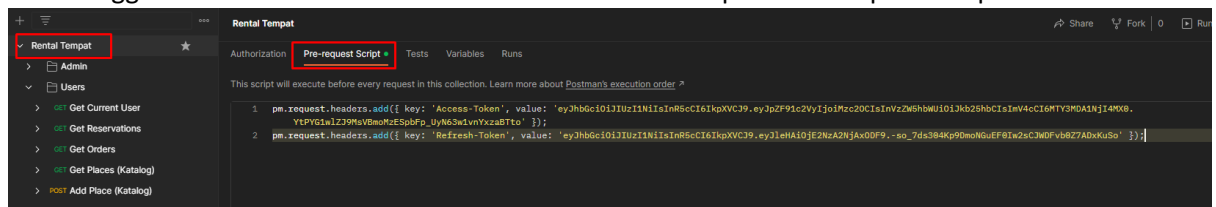
Lakukan *import database* (*restore database*)

2. Jalankan *file backend* dengan membuka folder *backend* lalu menjalankan terminal dan menuliskan perintah

```
go run main.go
```

Lakukan pengecekan pada Postman (telah disediakan contohnya)

Jika menggunakan Postman maka masukan header tersebut pada Pre-request Script



```
pm.request.headers.add({ key: 'Access-Token', value: ACCESS TOKEN ANDA });  
pm.request.headers.add({ key: 'Refresh-Token', value: REFRESH TOKEN ANDA });
```

3. Jalankan *file frontend* dengan membuka folder *frontend* lalu menjalankan terminal dan menuliskan perintah

```
npm start
```

Catatan:

- *backend* menggunakan port :3030
- *frontend* menggunakan port :3000

E. Uraian Basis Data

Basis data bernama: rental_tempat memiliki tabel-tabel sebagai berikut:

1. detail_reservasi – menyimpan uraian dan detail mengenai tempat yang di reservasi
 - id_reservasi (Primary Key)
 - id_tempat (Foreign Key → id_tempat dalam tabel tempat)
 - id_pemilik (Foreign Key → id_user dalam tabel pengguna)
 - tanggal_mulai
 - tanggal_selesai
 - id_penyewa (Foreign Key → id_user dalam tabel pengguna)
2. pengguna – menyimpan data akun pengguna
 - id_user (Primary Key)
 - username
 - password
 - nama_depan
 - nama_belakang
 - no_telp
3. tempat – menyimpan data tempat yang disewakan
 - id_tempat (Primary Key)
 - id_pemilik (Foreign Key → id_user dalam tabel pengguna)
 - nama_tempat
 - lokasi_tempat
 - harga_per malam
 - jumlah_kamar
 - jumlah_kamar_mandi
 - air_panas
 - fasilitas_lain
4. transaksi – menyimpan transaksi yang dibuat para pengguna
 - id_transaksi (Primary Key)
 - id_reservasi (Foreign Key → id_reservasi dalam tabel detail_reservasi)
 - total
 - metode_bayar

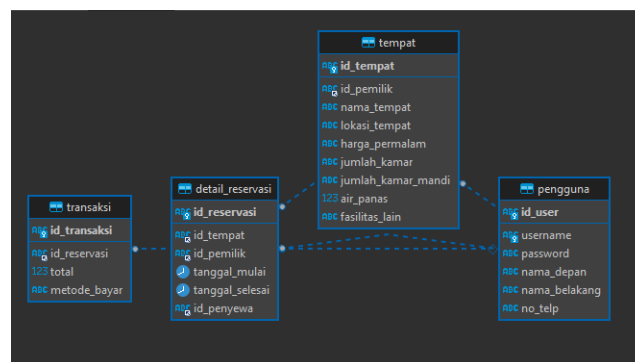


Diagram ERD Basis Data rental_tempat

F. Dokumentasi API

Untuk melihat respon mohon untuk melakukan testing pada aplikasi Postman.

Login:

- <http://localhost:3030/login>
- Perlu mengisi *body* dengan *username* dan *password*:
Contoh *body* (JSON) :

```
{  
  "username": "jowna",  
  "password": "admin123"  
}
```

Registrasi:

- <http://localhost:3030/register>
- Perlu mengisi *body* dengan *id_user*, *username*, *password*, *nama_depan*, *nama_belakang*, *no_telp*:
Contoh *body* (JSON) :

```
{  
  "id_user": "10",  
  "username": "barakobama",  
  "password": "barakobama",  
  "nama_depan": "barak",  
  "nama_belakang": "obama",  
  "no_telp": "12312312"  
}
```

Untuk memasuki sisi admin dan user, perlu memiliki 2 otorisasi JSON yang ada pada *headers* yaitu

“Access-Token”

“Refresh-Token”

Keduanya dapat diperoleh saat melakukan *login*, jika token expired, mohon melakukan *login* ulang.

SISI ADMIN – menggunakan akun admin:

Get Users:

- Metode: GET
- <http://localhost:3030/admin/users>

Get Single User:

- Metode: GET
- <http://localhost:3030/admin/users/:id> contohnya <http://localhost:3030/admin/users/03>

Update Place Catalog:

- Metode: PATCH
- <http://localhost:3030/admin/places/:id> (:id = id_tempat) contohnya <http://localhost:3030/admin/places/T07>
- Perlu mengisi *body* dengan *ide_tempat*, *ide_penyewa*, *nama_tempat*, *lokasi_tempat*, *harga_per malam*, *jumlah_kamar*, *jumlah_kamar_mandi*, *air_panas*, dan *fasilitas_lain*:
Contoh *body* (JSON) :

```
{
  "id_tempat": "T03",
  "id_pemilik": "02",
  "nama_tempat": "Villae2",
  "lokasi_tempat": "Puncak2",
  "harga_per malam": "50000",
  "jumlah_kamar": "6",
  "jumlah_kamar_mandi": "5",
  "air_panas": "true",
  "fasilitas_lain": "kolam"
}
```

Delete Place:

- Metode: GET
- <http://localhost:3030/admin/places/:id> (:id = id _tempat) contohnya <http://localhost:3030/admin/places/T07>

SISI USER – menggunakan Token untuk filtering

Get Current User:

- Metode: GET
- <http://localhost:3030/user/>

Get Reservations:

- Metode: GET
- <http://localhost:3030/user/reservations>

Get Orders:

- Metode: GET
- <http://localhost:3030/user/orders>

Get Places:

- Metode: GET
- <http://localhost:3030/user/places>

Add Place:

- Metode: POST
- <http://localhost:3030/user/places>
- Perlu mengisi *body* dengan *ide_tempat*, *ide_penyewa*, *nama_tempat*, *lokasi_tempat*, *harga_per malam*, *jumlah_kamar*, *jumlah_kamar_mandi*, *air_panas*, dan *fasilitas_lain*:
Contoh *body* (JSON) :

```
{
  "id_tempat": "890",
  "id_pemilik": "3768",
  "nama_tempat": "sunlake",
  "lokasi_tempat": "sunter",
  "harga_per malam": "450000",
  "jumlah_kamar": "1",
  "jumlah_kamar_mandi": "1",
  "air_panas": "true",
  "fasilitas_lain": "kolam"
}
```

Get Transaksi:

- Metode: GET
- <http://localhost:3030/user/transactions/>

Add Transaksi:

- Metode: POST
- <http://localhost:3030/user/transactions/>
- Perlu mengisi *body* dengan *id_transaksi*, *id_reservasi*, *total*, dan *metode_bayar*:
Contoh *body* (JSON) :

```
{
  "id_transaksi": "519",
  "id_reservasi": "R0378",
  "total": 500000000,
  "metode_bayar": "OVO"
}
```

Add Reservation:

- Metode: POST
- <http://localhost:3030/user/reservations/>

- Perlu mengisi *body* dengan *id_reservasi*, *id_tempat*, *id_pemilik*, *tanggal_mulai*, *tanggal_selesai*, dan *ide_penyewa*:
Contoh *body* (JSON) :

```
{
  "id_reservasi": "892",
  "id_tempat": "T890",
  "id_pemilik": "3768",
  "tanggal_mulai": "2022-12-02",
  "tanggal_selesai": "2022-12-26",
  "id_penyewa": "03"
}
```

Catatan: Dalam Frontend, *id_pemilik* akan secara otomatis tertulis

G. Saran Pengembangan

Proyek yang dikembangkan bersifat prototipe dikarenakan fokus dalam sisi *backend* adalah untuk operasi *create*, *read*, *update*, *delete*, dan, *searching*. Sedangkan pada bagian *frontend* yang dimodelkan sebagai gambaran saja sehingga berikut adalah saran untuk mengembangkan proyek lebih lanjut:

Backend

1. Mengimplementasi enkripsi atau *hashing* pada *password* pengguna
2. Mengelola *foreign key* sehingga dapat menghapus atau memodifikasi data secara serentak

Frontend

1. Rekonstruksi ulang tampilan terlihat lebih elegan
2. Melakukan validasi nomor telepon
3. Mengecek apakah ID sudah ada pada tabel dimana ID ditentukan secara acak
4. Melakukan validasi untuk memastikan tanggal selesai terjadi setelah tanggal mulai
5. Melakukan validasi pengguna tidak dapat menyewa tempat sendiri

Dengan ini, saya harap proyek dapat memenuhi kriteria penilaian yang telah ditentukan dan dapat dikembangkan lebih lanjut