

GitHub: <https://github.com/agiftsanyazhar/TTG.git>

Postman: <https://www.postman.com/gold-eclipse-674635/workspace/ttg>

Soal 2 – Back-End API Pengguna

1. Deskripsi Proyek

Proyek ini merupakan aplikasi backend API Node.js untuk sistem pengelolaan user sederhana menggunakan framework Express. Sistem ini menyediakan fitur CRUD (Create, Read, Update, Delete) untuk data user (Admin dan Pegawai). Namun, fitur CRUD hanya tersedia untuk data user. Data master hanya bisa diakses melalui fitur Read. Selain itu, ada juga fitur register untuk user baru. Fitur Utama:

1. Pengelolaan User:

- Create: Menambahkan user baru ke dalam sistem.
- Read: Melihat daftar user yang ada atau informasi detail user berdasarkan ID.
- Update: Memperbarui data user, seperti nama, email, dan password.
- Delete: Menghapus data user berdasarkan ID.

2. Data Master (Role dan Jabatan):

- Role dan Jabatan hanya memiliki fitur Read. Artinya, data pada tabel ini hanya bisa di-read, tidak bisa di-update atau di-delete melalui API.
- Role: Menyimpan informasi mengenai role yang ada dalam sistem (misalnya: Admin, Pegawai).
- Jabatan: Menyimpan informasi mengenai jabatan yang ada dalam sistem (misalnya: Kepala Bengkel, Teknisi, dll).

2. Teknologi yang Digunakan

- Backend: Node.js dan Express.js (framework web yang digunakan untuk develop API RESTful).
- Database: Sequelize ORM (Object-Relational Mapping) untuk connect dengan database (menggunakan MySQL).
- Testing: Pengujian API dilakukan menggunakan Postman untuk memastikan semua endpoint berfungsi dengan baik.

3. Database dan Penjelasan Tabel

1. Tabel Roles (roles): Menyimpan informasi mengenai role yang ada dalam sistem. Tabel ini hanya bisa di-read, tidak ada fitur CRUD.
2. Tabel Jabatan (jabatans): Menyimpan informasi mengenai jabatan yang ada dalam sistem. Tabel ini hanya bisa di-read, tidak ada fitur CRUD.
3. Tabel Users (users): Menyimpan data seluruh user dalam sistem, baik admin maupun pegawai.
4. Tabel Admins (admins): Menyimpan data khusus untuk user dengan role admin.
5. Tabel Pegawais (pegawais): Menyimpan data khusus untuk user dengan role pegawai.

4. Proses CRUD

1. Register: Proses register user baru (baik admin maupun pegawai) melalui endpoint /register. Data user yang baru akan disimpan di tabel users, dan jika user tersebut adalah pegawai, maka akan ada data tambahan di tabel pegawais.
2. Login: User dapat melakukan login melalui endpoint /login. Sistem akan memverifikasi data user dan memberikan token untuk sesi user.
3. Roles: User yang login dapat melakukan read melalui endpoint yang ada di /role. Data ini disimpan di tabel roles.

4. Jabatan: User yang login dapat melakukan read jabatan melalui endpoint yang ada di /jabatan. Data jabatan disimpan di tabel jabatans.
5. CRUD untuk User: User yang login dapat mengelola data user (admin dan pegawai) melalui endpoint yang ada di /user. Data ini disimpan di tabel users, dan jika user adalah pegawai, maka akan ada data tambahan di tabel pegawais.

5. Struktur Folder

1. models/: Berisi file untuk me-define model Sequelize seperti UserModel.js, AdminModel.js, PegawaiModel.js, RoleModel.js, dan JabatanModel.js.
2. controllers/: Menyediakan fungsi untuk menangani logika bisnis dan interaksi dengan database.
3. routes/: Menangani routing API, termasuk route untuk mengakses data user, jabatan, roles, dan autentikasi.
4. middleware/: Menyediakan middleware untuk memverifikasi token autentikasi pada setiap request yang membutuhkan proteksi.
5. seeders/: Menyediakan file untuk mengisi data awal ke dalam tabel database seperti roles, jabatan, users, admins, dan pegawais.

6. Testing

Setelah semua API endpoint selesai, testing dilakukan menggunakan Postman untuk memastikan bahwa semua fitur berfungsi dengan baik. Endpoint-endpoint yang diuji meliputi:

- Register dan login user.
 - Read untuk data role dan jabatan.
 - CRUD untuk user (admin dan pegawai).
 - Verifikasi bahwa API mengembalikan data sesuai dengan yang diharapkan, dan menangani error dengan benar.
1. Menambahkan, memperbarui, dan menghapus user.
 2. Menampilkan data user berdasarkan ID.
 3. Menampilkan data jabatan dan roles yang ada.

7. Cara menjalankan Program

1. Persiapan Environment

- Node.js: Pastikan sudah menginstal Node.js versi minimal v20.16.0.
- Database: Proyek ini menggunakan MySQL atau MariaDB sebagai database. Pastikan sudah menginstal salah satu dari keduanya, atau menggunakan web server seperti XAMPP atau Laragon untuk menjalankan database secara lokal.

2. Download atau Clone Proyek

Download di [GitHub](https://github.com/agiftsanyazhar/TTG) atau clone menggunakan Git dengan perintah:

```
git clone https://github.com/agiftsanyazhar/TTG.git
```

3. Instalasi Dependensi

Setelah memiliki proyek di komputer, buka terminal atau command prompt dan arahkan ke folder proyek. Setelah itu, jalankan perintah berikut untuk menginstal semua dependensi yang diperlukan:

```
npm install
```

Perintah ini akan menginstal semua package yang tercantum dalam file package.json, seperti Express, Sequelize, dan lainnya.

4. Konfigurasi Database

Atur database yang akan digunakan oleh aplikasi. Buat database dan mengonfigurasi koneksi di file konfigurasi aplikasi.

- Buka file config/config.json dan config/database.js.
- Sesuaikan host, username, password, dan nama database sesuai dengan pengaturan database yang kamu gunakan.

5. Menjalankan Migrasi dan Seeder (Opsional)

Jalankan migrasi untuk membuat tabel di database. Jalankan perintah berikut:

```
node index.js
```

Jalankan seeder dengan perintah ini.

```
npm run seed
```

6. Menjalankan Aplikasi

Setelah semua konfigurasi dan dependensi selesai, jalankan aplikasi dengan perintah berikut:

```
npm start
```

Perintah ini akan menjalankan server Express dan aplikasi akan berjalan di port default (port 5000). Jika ingin menentukan port lain, bisa mengubahnya di file index.js di baris yang mengatur port.

7. Mengakses API

Setelah aplikasi berjalan, dapat mengakses API di browser atau menggunakan Postman. Jika aplikasi berjalan di `http://localhost:5000`, bisa mengakses berbagai endpoint seperti:

- GET /users: Untuk melihat daftar user.
- POST /register: Untuk melakukan register user baru.
- GET /roles: Untuk melihat daftar roles yang tersedia.
- GET /jabatan: Untuk melihat daftar jabatan yang tersedia.

8. Pengujian API dengan Postman

Untuk menguji API, kamu bisa menggunakan Postman.

- Jalankan Postman dan buat request ke endpoint yang sesuai (misalnya, GET /users untuk melihat data user).
- Bisa mengirimkan data JSON untuk register user dengan endpoint POST /register.
- Atur Authorization karena aplikasi menggunakan JWT Token.

8. Kesimpulan

Proyek ini merupakan sistem pengelolaan user dengan API RESTful menggunakan Node.js dan Express.js. Aplikasi ini memfasilitasi pengelolaan user (Admin dan Pegawai), termasuk register dan akses data master seperti jabatan dan roles. Semua fitur telah diuji menggunakan Postman untuk memastikan API berfungsi dengan baik.

Soal 3 – Problem Solving: Cari Angka yang Hilang

Buatlah sebuah function untuk mencari nomor yang hilang dari dalam Array. Constraint:

- Nomor dalam input array pasti berurutan.
- Input array bisa sampai ribuan, tidak selalu dimulai dari nol.

Contoh:

Input: [3, 0, 2, 4]

Output: 1

Contoh:

Input: [3106, 3102, 3104, 3105, 3107]

Output: 3103

Penjelasan:

Melakukan iterasi terhadap array, lalu mencari angka yang hilang dengan cara:

1. Mengurutkan array terlebih dahulu.

```
1 <?php
2 function findMissingNumber($array)
3 {
4     sort($array);
5
6     for ($i = 0; $i < count($array) - 1; $i++) {
7         if ($array[$i + 1] - $array[$i] > 1) {
8             return $array[$i] + 1;
9         }
10    }
11
12    return null;
13 }
14
15 $input1 = [3, 0, 2, 4];
16 echo "Angka yang Hilang: " . findMissingNumber($input1) . "\n";
17
18 $input2 = [3106, 3102, 3104, 3105, 3107];
19 echo "Angka yang Hilang: " . findMissingNumber($input2) . "\n";
20
```

2. Melakukan pengecekan angka yang hilang dengan membandingkan elemen saat ini dengan elemen berikutnya. Jika ada selisih lebih dari 1, maka angka yang hilang berada di antara dua angka tersebut.

```
1 [Running] php "c:\nodejs\ttg\2. Soal 3\cari-angka.php"
2 Angka yang Hilang: 1
3 Angka yang Hilang: 3103
4
5 [Done] exited with code=0 in 0.083 seconds
6
7
```

Soal 4 – Problem Solving: Formula Perhitungan

Diberikan daftar angka dan target angka, buatlah sebuah function untuk menampilkan cara menghitung menggunakan operator (+, -, *) untuk mencapai target yang diinginkan.

Constraint

- Operator yang digunakan: +, -, *.
- Semua angka wajib digunakan.

Contoh: **Berhasil**

Input: [1, 4, 5, 6]

Target: 16

Output: 1 + 4 + 5 + 6

Contoh: **Gagal**

Input: [1, 4, 5, 6]

Target: 18

Output: (1 + 5) * 4 - 6

Contoh: **Gagal**

Input: [1, 4, 5, 6]

Target: 50

Output: (4 + 6) * 5

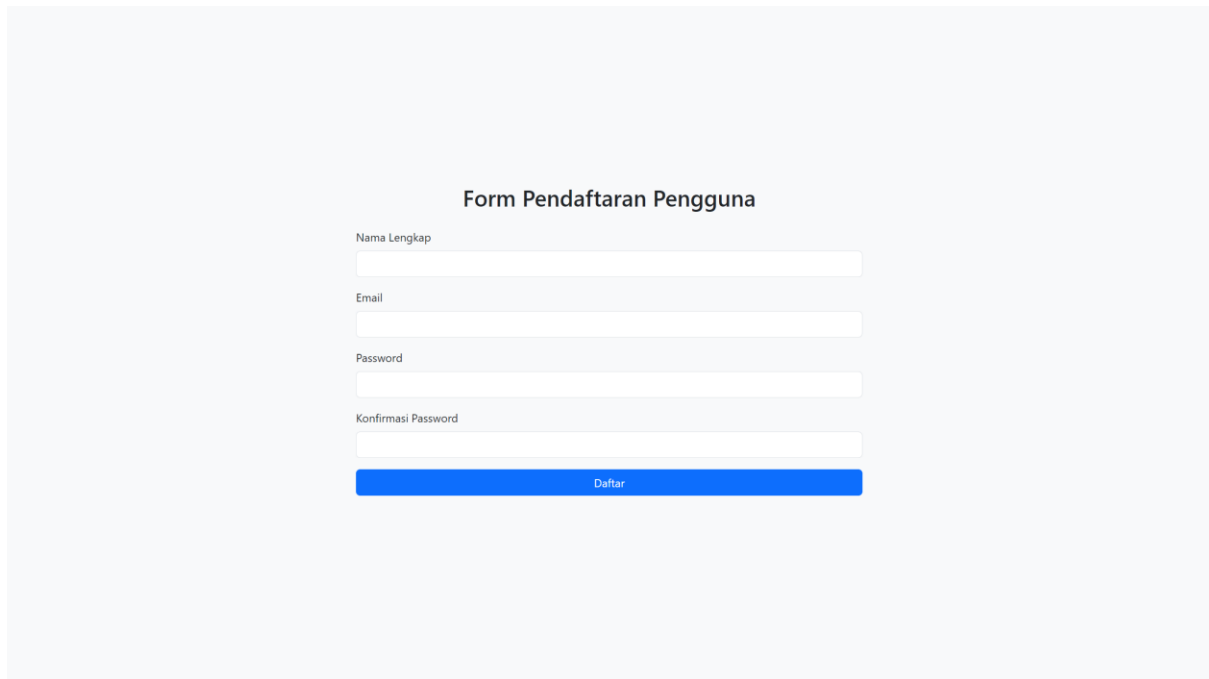


```
1 [Running] php "c:\nodejs\ttg\3. Soal 4\formula-perhitungan.php"
2 Input: 1, 4, 5, 6
3 Target: 16
4 Output: 1 + 4 + 5 + 6
5
6 Input: 1, 4, 5, 6
7 Target: 18
8 Output: Tidak ada kombinasi yang menghasilkan target.
9
10 Input: 1, 4, 5, 6
11 Target: 50
12 Output: Tidak ada kombinasi yang menghasilkan target.
13
14
15 [Done] exited with code=0 in 0.081 seconds
16
17
```

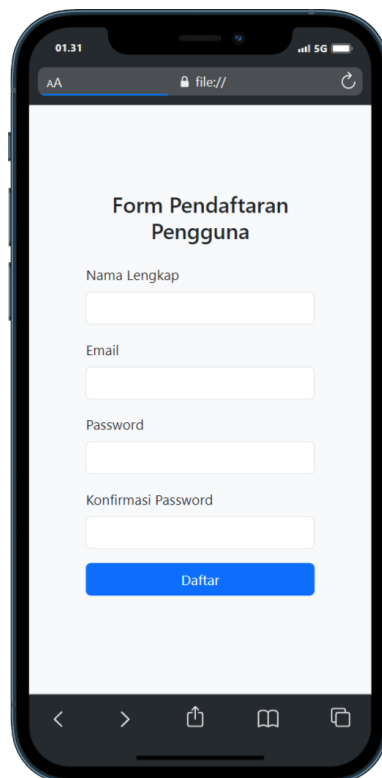
Soal 1 – Front-End: Membuat Halaman Web Sederhana

Struktur file:

1. index.html: Berisi struktur HTML dari form pendaftaran, link ke Bootstrap dan file CSS serta JavaScript eksternal.
2. style.css: Berisi styling tambahan untuk mempercantik tampilan form.
3. script.js: Berisi logika JavaScript untuk validasi form dan penanganan event submit.



The image shows a desktop browser window displaying a registration form titled "Form Pendaftaran Pengguna". The form is centered on a light gray background. It contains four input fields stacked vertically, each with a label to its left: "Nama Lengkap", "Email", "Password", and "Konfirmasi Password". Below the input fields is a solid blue button with the white text "Daftar".



The image shows a mobile phone screen displaying the same registration form. The form is titled "Form Pendaftaran Pengguna" and is centered. It features the same four input fields and the "Daftar" button as the desktop version. The phone's status bar at the top shows the time "01.31", signal strength, and "5G". The bottom of the screen shows a standard Android navigation bar with back, forward, and other icons.