

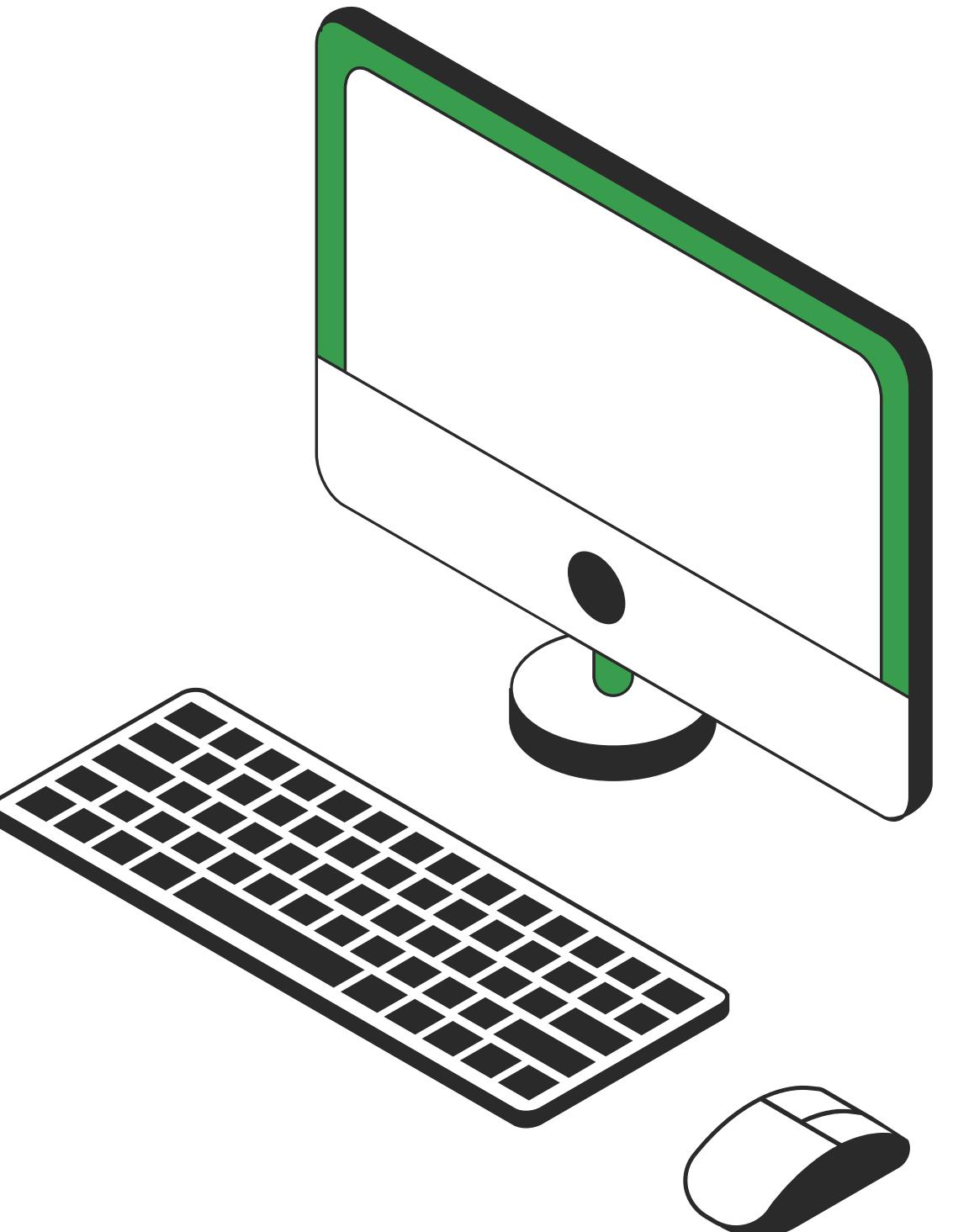
Kelompok 15 SBD

Web Digilab using React and NodeJs

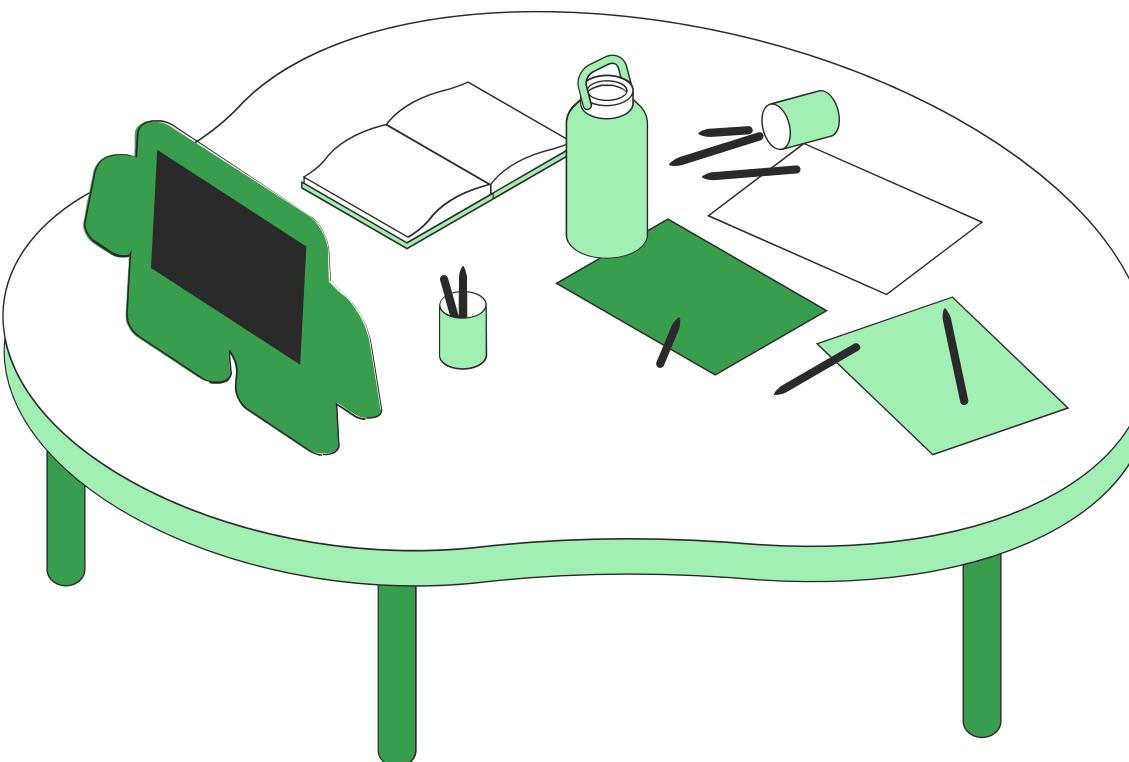


Anggota Kelompok

1. Evandita Wiratama Putra - 2206059572
2. Raihan Muhammad Ihsan - 2206028232
3. Muhammad Daffa Rizkyandri - 2206829194



Daftar Isi



Latar Belakang

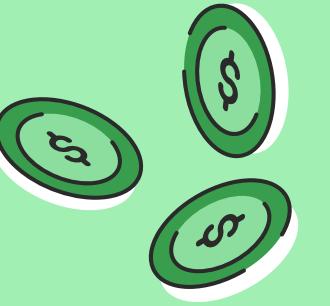
Front-end Aplikasi

Back-end Aplikasi

Implementasi CRUD

Kesimpulan

Latar Belakang



Untuk saat ini, kegiatan praktikum yang diselenggarakan oleh Digital Laboratory masih membutuhkan jasa dari Website EMAS milik Universitas Indonesia dalam mengunggah soal serta materi praktikum.

Web Digilab diciptakan agar kegiatan praktikum tidak sepenuhnya mengandalkan website EMAS lagi. Hal ini bertujuan agar tampilan EMAS praktikan diharapkan hanya berisikan course yang disediakan dosen saja, sedangkan course Asisten akan diunggah pada Web Digilab secara tersendiri.



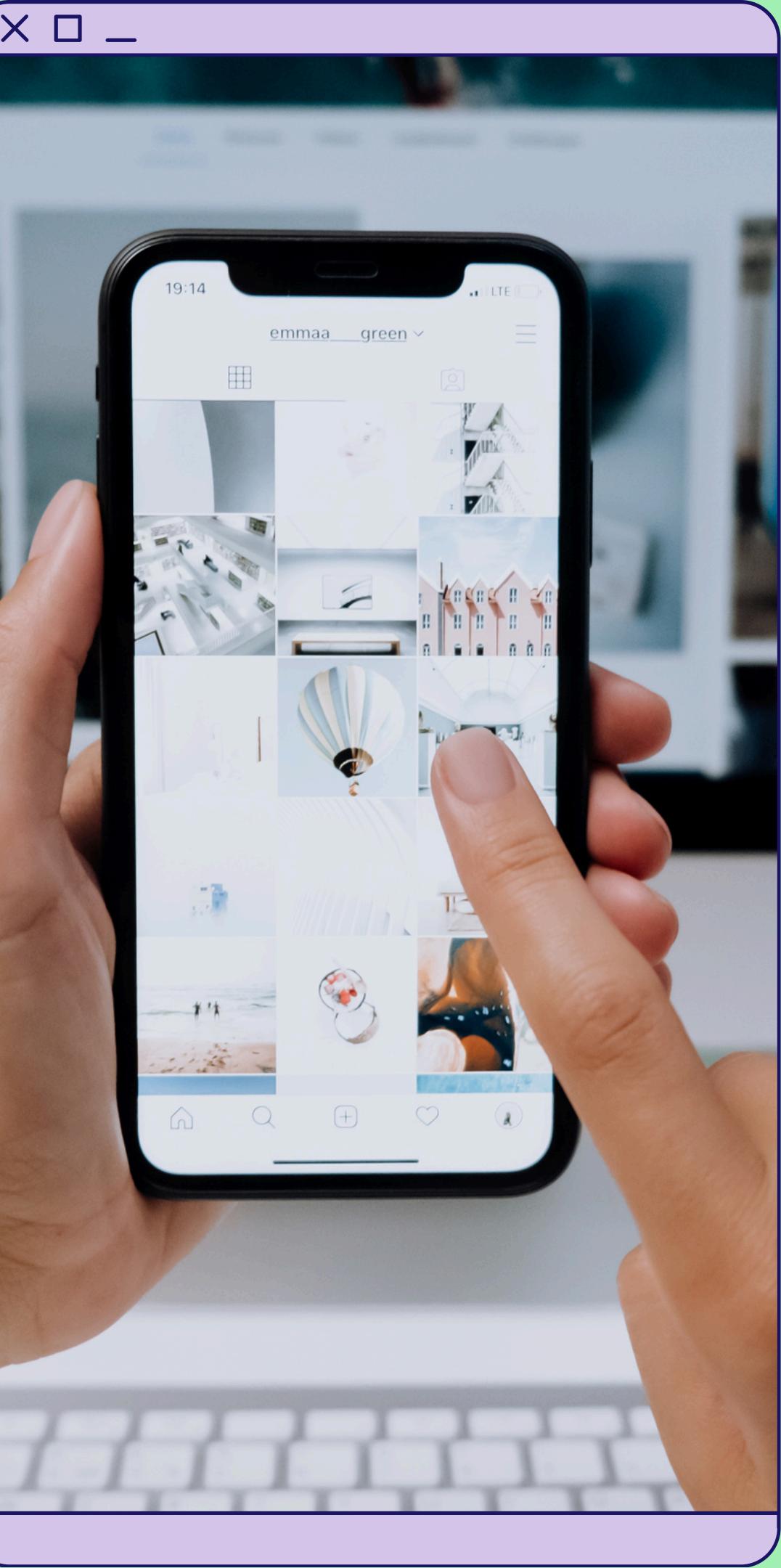


Web Digilab juga merupakan salah satu wujud agar kegiatan praktikum Laboratory Digital dapat berkembangkan kedepannya dengan memanfaatkan Websitenya sendiri.





Front-end



AccountSetting.jsx

Bagian dari aplikasi web untuk mengelola pengaturan akun pengguna. Pengguna dapat memperbarui info akun dan praktikan/Aslab dapat memperbarui info masing-masing. Ada validasi formulir dan opsi untuk menghapus akun. Program ini menggunakan cookie untuk menyimpan info pengguna.

App.jsx

Bagian dari aplikasi web menggunakan React Router untuk navigasi antar halaman. Komponen utama App menggunakan <BrowserRouter> untuk routing. Setiap <Route> menentukan respons aplikasi terhadap URL yang diakses pengguna. Dengan React Router, navigasi antar halaman berdasarkan URL pengguna dapat dilakukan dengan mudah.

Course.jsx

Halaman untuk memilih bab dalam kursus berdasarkan ID. Pengguna dapat melihat daftar bab, memilihnya, dan konten bab akan ditampilkan. Tombol toggle untuk menyembunyikan/menampilkan daftar bab disediakan.

ChapterAdd.jsx

Bagian aplikasi web untuk menambahkan bab baru ke kursus. Pengguna memasukkan info bab melalui formulir, yang dikirim ke server dengan permintaan POST. Validasi dilakukan sebelum pengiriman data. Halaman akan dimuat ulang setelah penambahan berhasil. Selain itu, pengguna dapat memilih kursus dari daftar yang tersedia.

CoursesEdit.jsx

Halaman pengeditan kursus dalam aplikasi DigiLab. Pengguna dapat menambah kursus baru, bab baru, atau mengedit kursus yang ada. Menggunakan React Router dan react-cookie untuk navigasi dan autentikasi. Tampilan antarmuka yang ramah pengguna memudahkan pengguna terautentikasi.

CourseAdd.jsx

Form untuk membuat kursus baru. Pengguna memilih mata kuliah dan mengisi detail kursus. Setelah mengisi semua detail, pengguna dapat membuat kursus baru dengan mengirimkan permintaan POST. Pesan peringatan akan muncul jika ada bidang yang kosong.

CourseEdit.jsx

Komponen untuk mengedit kursus yang ada. Pengguna dapat memperbarui detail kursus dan bab yang terkait. Formulir akan menampilkan input untuk mengedit detail bab jika dipilih. Pengguna juga dapat menghapus kursus atau bab. Menggunakan Tailwind CSS untuk tata letaknya.

Courses.jsx

Menampilkan daftar kursus yang tersedia dalam aplikasi DigiLab. Menggunakan React Router dan react-cookie untuk navigasi dan autentikasi. Pengguna dapat memilih kursus dan melihat detailnya. Menyediakan antarmuka ramah pengguna dan mudah diakses.

GetContact.jsx

Menampilkan kontak - kontak para aslab digilab beserta informasi lainnya tentang aslab yang bersangkutan. Ditampilkan secara statis pada frontend sehingga jika diinginkan adanya update pada page GetContact, harus dilakukan penulisan line code baru.

Quizzes.jsx

Menampilkan quiz yang ada. Semua quiz ditampilkan dengan filter berupa Course dan Chapter dari Course tersebut yang dapat dipilih pada drop down. Pengguna dapat mengeklik salah satu quiz untuk dapat berpindah dan mengakses kuis tujuan.

Quiz.jsx

Halaman untuk praktikan melakukan kuis. Praktikan akan diberikan kolom untuk menjawab dengan isian singkat, dan jawaban akan langsung diperiksa untuk kemudian hasilnya dimasukkan ke database. Hasil akan diperiksa secara otomatis oleh sistem dan akan langsung ditampilkan pada praktikan. Gambar pada soal dapat diklik oleh praktikan untuk view yang lebih besar. Fitur ini memanfaatkan implementasi react modal.

QuizAdd.jsx

Halaman untuk Aslab membuat kuis baru. Asisten lab dapat membuat kuis baru dengan memasukkan detail kuis dan pertanyaan pada halaman ini. Quiz dapat dibuat jika semua pertanyaan telah ditambahkan.

index.css

Aturan dasar dari Tailwind CSS untuk mengimpor konfigurasi bawaan ke dalam proyek. Memungkinkan penggunaan fitur dan gaya yang telah ditetapkan oleh Tailwind CSS melalui kelas-kelas yang telah ditetapkan.

Home.jsx

Halaman utama dalam aplikasi DigiLab. Menyambut pengguna dengan pesan selamat datang yang dipersonalisasi. Pengguna disajikan dengan opsi untuk menjelajahi kursus atau mengedit kursus jika sebagai aslab. Menggunakan React Router dan react-cookie untuk navigasi dan autentikasi. Tampilan antarmuka yang ramah pengguna memudahkan pengguna terautentikasi.

main.jsx

Bagian dari file index.js dalam proyek React. ReactDOM digunakan untuk me-render komponen App ke dalam elemen HTML dengan id 'root'. Metode createRoot digunakan untuk membuat root React baru, dan kemudian metode render digunakan untuk me-render komponen App ke dalam root tersebut. Komponen App dibungkus dalam <React.StrictMode> untuk memastikan aplikasi dijalankan dalam mode ketat. Juga disertakan import untuk file index.css untuk styling aplikasi.

Login.jsx

Komponen untuk proses login pengguna ke dalam platform DigiLab. Menggunakan state untuk menyimpan email dan password pengguna. Menggunakan fetch API untuk mengirim permintaan HTTP ke server untuk login. Hasilnya digunakan untuk menyimpan informasi pengguna dan status menggunakan cookies. Jika login berhasil, pengguna diarahkan ke halaman beranda.

Navigation.jsx

Komponen navigasi dalam aplikasi React. Menampilkan navigasi untuk pengguna yang sudah login dan yang belum login. Bagian untuk pengguna yang sudah login menampilkan dropdown dengan info pengguna dan opsi untuk mengatur akun atau logout. Bagian untuk pengguna yang belum login menampilkan tautan menuju halaman utama, kursus, kontak, dan tentang. Menggunakan useEffect untuk menangani logika logout dan menutup dropdown. Mengambil info pengguna dari cookie untuk menampilkan foto profil dan bio.

Register.jsx

Komponen untuk formulir pendaftaran akun pengguna. Pengguna diminta mengisi info dan memilih jenis akun (aslab/praktikan). Setelah pendaftaran, pengguna akan diberitahu dan diarahkan kembali ke halaman login. Menggunakan efek Spline untuk latar belakang dinamis saat formulir ditampilkan.



Back-end



repository.account.js

Pada skrip ini, PostgreSQL digunakan sebagai basis data untuk menyimpan data pengguna dan akun. Modul bcrypt digunakan untuk mengenkripsi kata sandi sebelum disimpan, dengan fungsi **hashPassword** dan **validateUser**. Fungsi **createAccountAslab** dan **createAccountPraktikan** digunakan untuk membuat akun baru dengan pemeriksaan untuk menghindari duplikasi NPM atau email. Fungsi login seperti **loginAccountAslab** dan **loginAccountPraktikan** memeriksa kecocokan email dan kata sandi saat login. Ada juga fungsi untuk memperbarui atau menghapus akun pengguna yang ada dalam basis data dengan penanganan kesalahan yang sesuai.

repository.content.js

Modul ini menyediakan fungsi-fungsi untuk berinteraksi dengan basis data PostgreSQL, termasuk pembuatan, pengambilan, pembaruan, dan penghapusan data. Fungsi-fungsi tersebut termasuk pembuatan mata kuliah, pembuatan kursus, pengambilan semua kursus, pengambilan kursus berdasarkan ID mata kuliah, pengambilan semua mata kuliah, pembuatan bab kursus, pengambilan semua bab kursus berdasarkan ID kursus, pembaruan kursus, penghapusan kursus, penghapusan bab kursus, dan pembaruan bab kursus. Setiap fungsi menangani kesalahan dan memberikan respons HTTP yang sesuai.

index.js

Kode ini menginisialisasi server Express yang mendengarkan pada port yang ditentukan dalam variabel lingkungan. Middleware Cors dan express.urlencoded digunakan untuk menangani permintaan HTTP dan memungkinkan CORS. Ada juga serangkaian endpoint yang ditentukan untuk mengelola akun dan konten, menggunakan fungsi-fungsi yang didefinisikan di modul repository.account dan repository.content. Saat server berjalan, pesan akan dicetak ke konsol yang menunjukkan bahwa server berjalan dan mendengarkan pada port yang ditentukan.

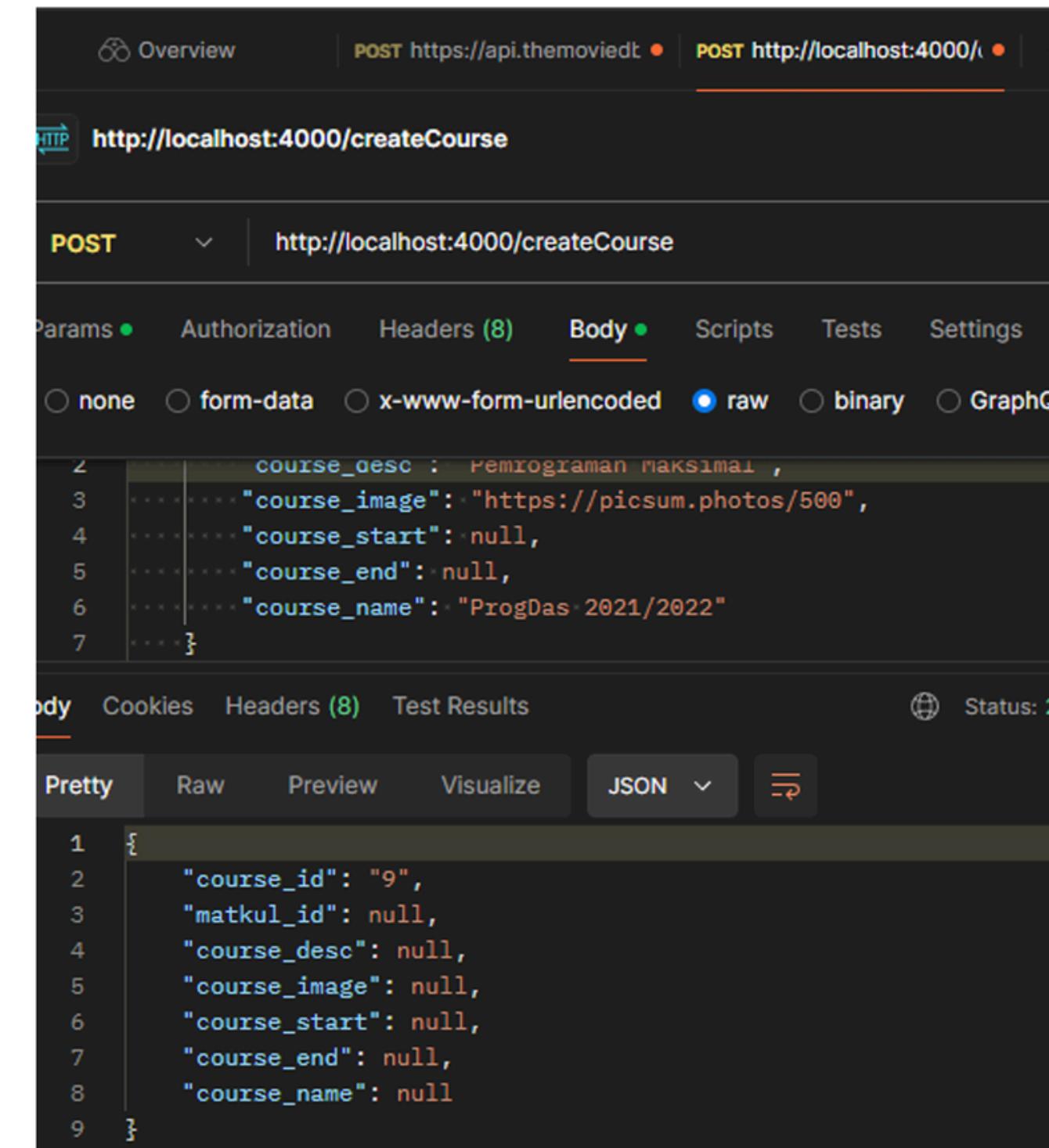


Implementasi CRUD

Implementasi CRUD pada Web Digilab

CREATE

Proses penambahan data baru dengan mengirimkan permintaan POST ke endpoint "createCourse" dengan skrip PHP untuk menambahkan entri baru ke dalam basis data. Pesan "Insert data berhasil" akan ditampilkan ketika data yang dikirim telah berhasil ditambahkan ke dalam basis data.



The screenshot shows a Postman interface for a POST request to `http://localhost:4000/createCourse`. The 'Body' tab is selected, showing raw JSON data:

```
2 |     "course_desc": "Pemrograman maksimal",
3 |     "course_image": "https://picsum.photos/500",
4 |     "course_start": null,
5 |     "course_end": null,
6 |     "course_name": "ProgDas 2021/2022"
7 | }
```

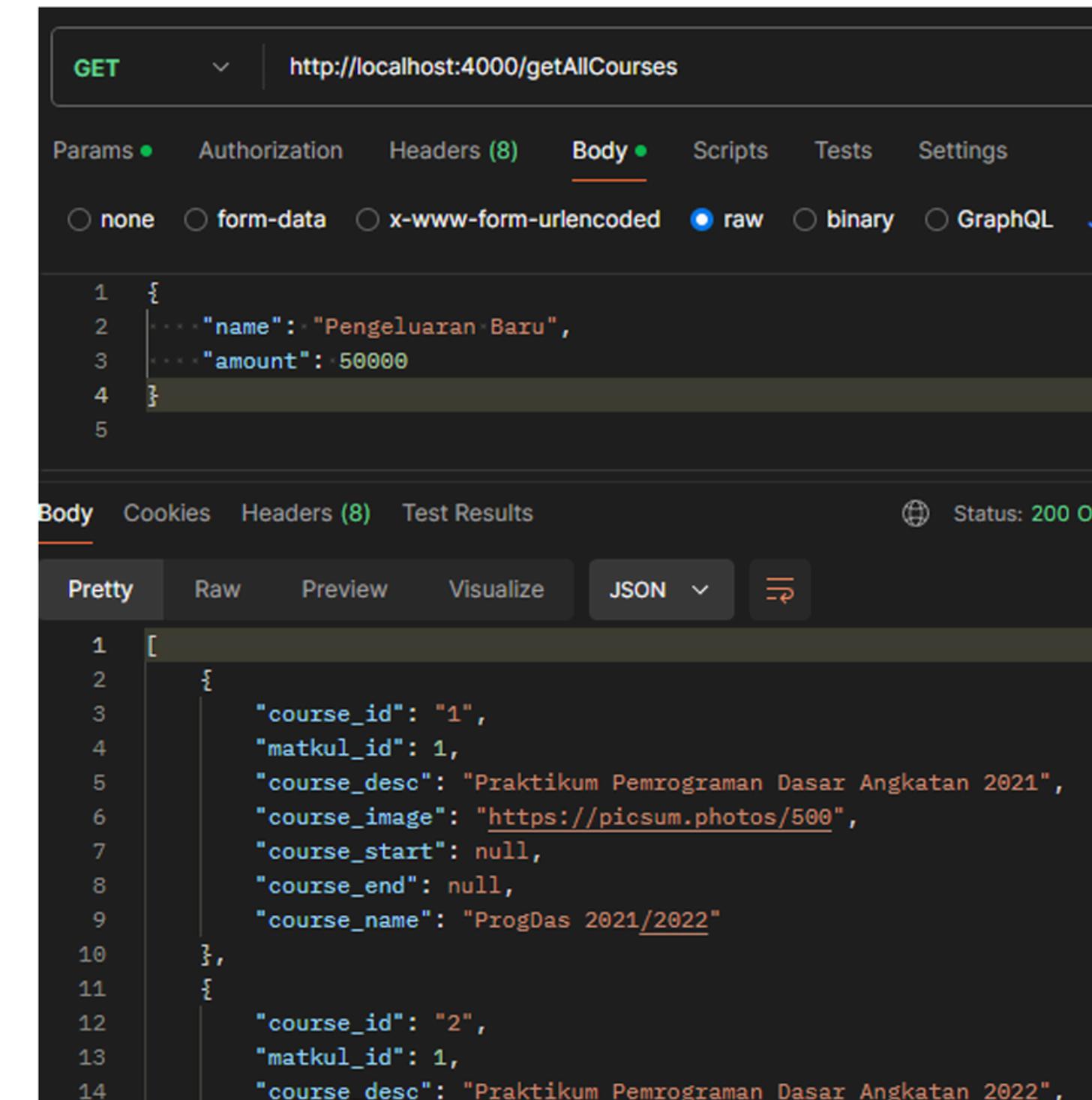
Below the body, the response is shown in 'Pretty' format:

```
1 {
2     "course_id": "9",
3     "matkul_id": null,
4     "course_desc": null,
5     "course_image": null,
6     "course_start": null,
7     "course_end": null,
8     "course_name": null
9 }
```

Implementasi CRUD pada Web Digilab

READ

Entri data yang terkirim dan data yang sudah ada di dalam basis data akan diakses dengan mengirimkan permintaan GET ke endpoint "getAllCourses" untuk menampilkan data dari tabel yang ditentukan dalam basis data, dan respons dari server akan berisi data dari semua course dalam format JSON.



The screenshot shows a Postman interface with the following details:

- Method: GET
- URL: <http://localhost:4000/getAllCourses>
- Body tab selected, showing raw JSON input:

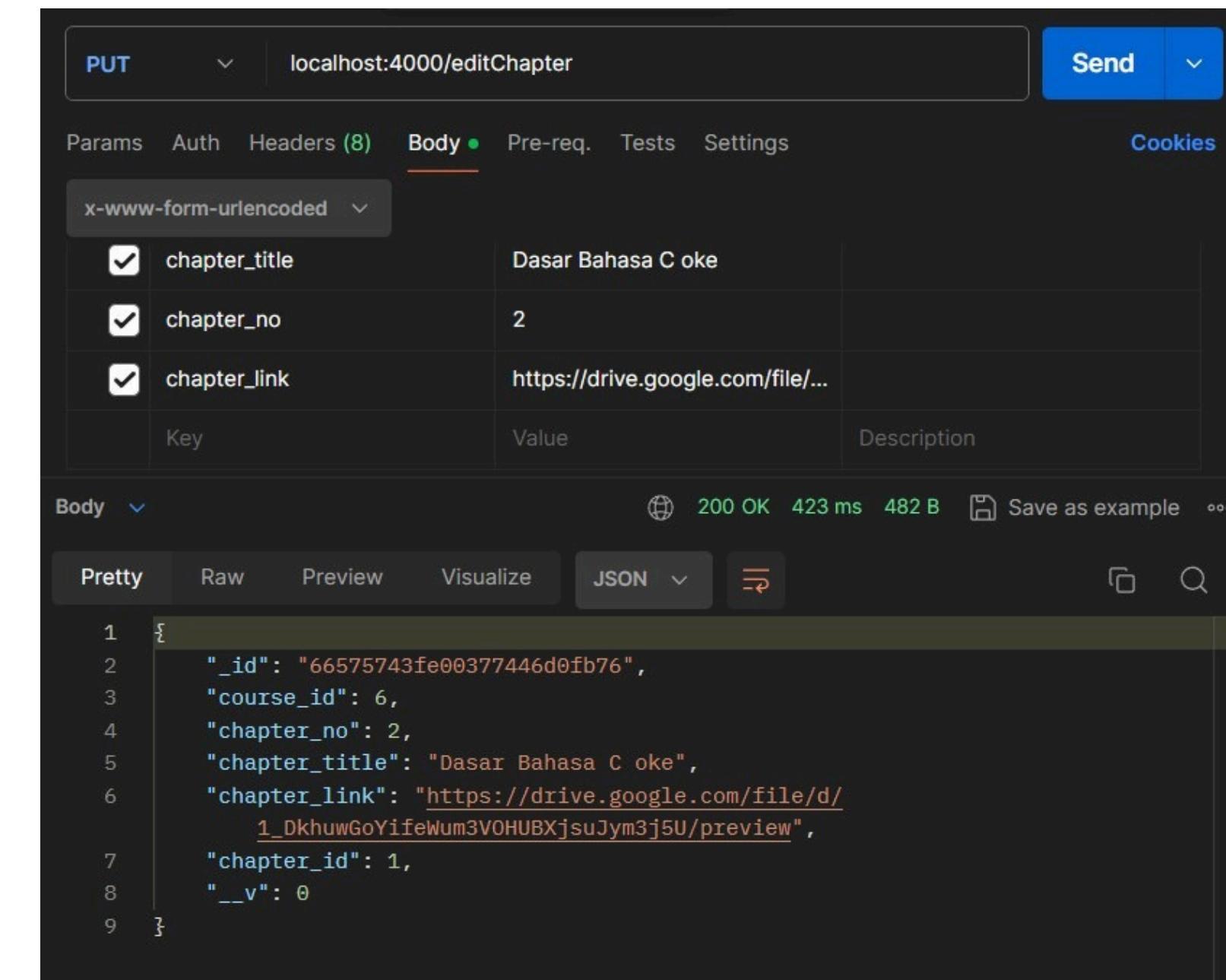
```
1 {  
2   ... "name": "Pengeluaran Baru",  
3   ... "amount": 50000  
4 }  
5
```
- Status: 200 OK
- Body tab selected, showing JSON response:

```
1 [  
2   {  
3     "course_id": "1",  
4     "matkul_id": 1,  
5     "course_desc": "Praktikum Pemrograman Dasar Angkatan 2021",  
6     "course_image": "https://picsum.photos/500",  
7     "course_start": null,  
8     "course_end": null,  
9     "course_name": "ProgDas 2021/2022"  
10    },  
11    {  
12      "course_id": "2",  
13      "matkul_id": 1,  
14      "course_desc": "Praktikum Pemrograman Dasar Angkatan 2022",  
15      "course_image": "https://picsum.photos/500",  
16      "course_start": null,  
17      "course_end": null,  
18      "course_name": "ProgDas 2022/2023"  
19    }  
20 ]
```

Implementasi CRUD pada Web Digilab

UPDATE

Proses memperbarui sebuah data yang terdapat pada database dilakukan dengan permintaan PUT ke endpoint "editChapter" untuk memperbarui data dari tabel yang ditentukan dalam database, dan respons dari server akan berisi data baru dari chapter yang diedit dalam format JSON.



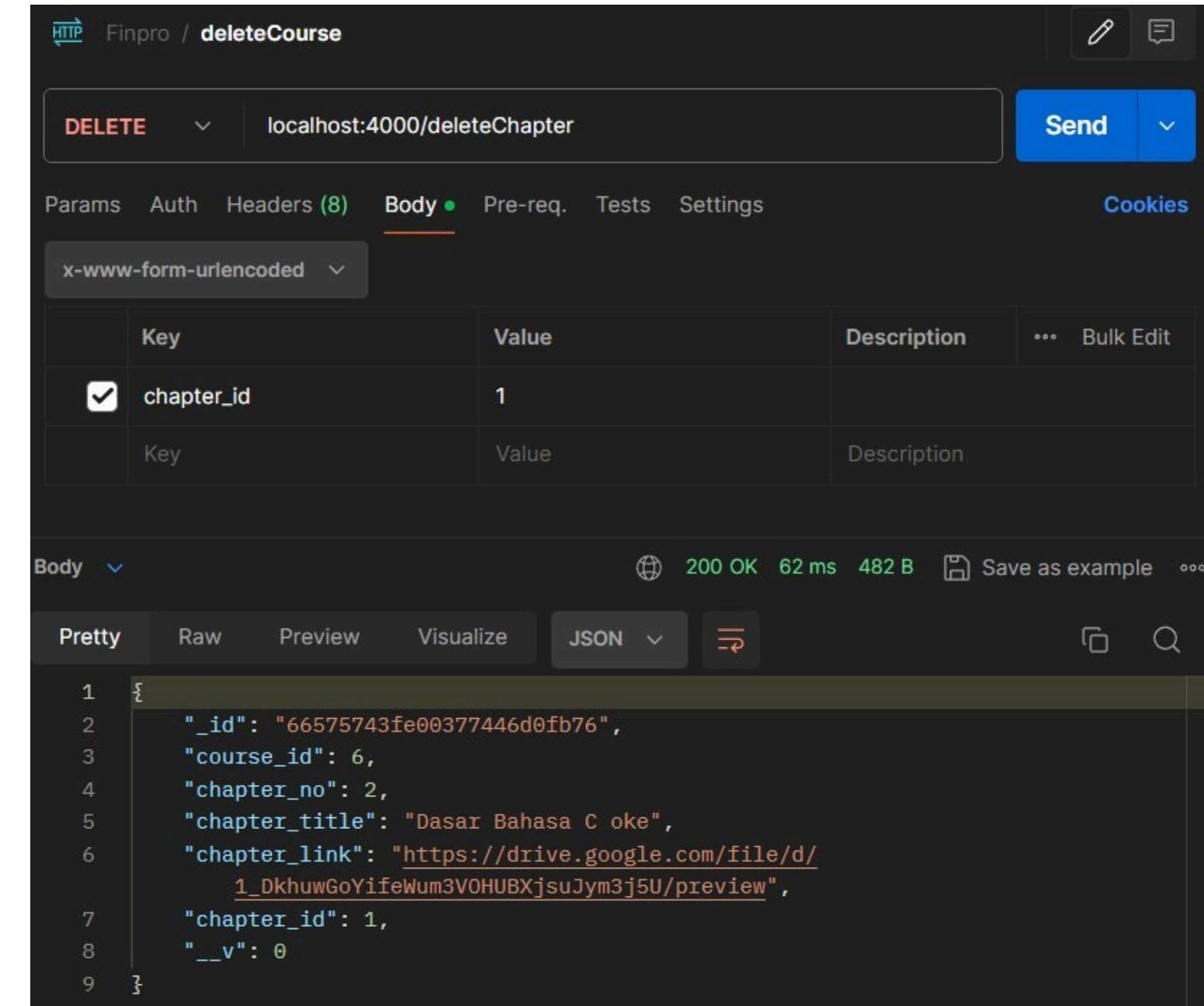
The screenshot shows a POSTMAN interface with a PUT request to `localhost:4000/editChapter`. The request body is set to `x-www-form-urlencoded` and contains three fields: `chapter_title` (value: `Dasar Bahasa C oke`), `chapter_no` (value: `2`), and `chapter_link` (value: `https://drive.google.com/file/d/1_DkhuvGoYifeWum3VOHUBXjsuJym3j5U/preview`). The response is a `200 OK` status with a `423 ms` duration and a `482 B` size. The JSON response body is:

```
1 {  
2   "_id": "66575743fe00377446d0fb76",  
3   "course_id": 6,  
4   "chapter_no": 2,  
5   "chapter_title": "Dasar Bahasa C oke",  
6   "chapter_link": "https://drive.google.com/file/d/  
7     1_DkhuvGoYifeWum3VOHUBXjsuJym3j5U/preview",  
8   "chapter_id": 1,  
9   "__v": 0  
}
```

Implementasi CRUD pada Web Digilab

DELETE

Proses menghapus sebuah data yang terdapat pada database dilakukan dengan permintaan DELETE ke endpoint "deleteChapter" untuk menghapus data dari tabel yang ditentukan dalam database, dan respons dari server akan berisi data dari chapter yang dihapus dalam format JSON



The screenshot shows a Postman API request for the endpoint `localhost:4000/deleteChapter`. The method is set to `DELETE`. In the `Body` section, there is a table with one row containing a key-value pair: `chapter_id: 1`. The response tab shows a `200 OK` status with a response body in JSON format:

```
1 {  
2   "_id": "66575743fe00377446d0fb76",  
3   "course_id": 6,  
4   "chapter_no": 2,  
5   "chapter_title": "Dasar Bahasa C oke",  
6   "chapter_link": "https://drive.google.com/file/d/  
    1_OkhuwGoYifeWum3VOHUBXjsuJym3j5U/preview",  
7   "chapter_id": 1,  
8   "__v": 0  
9 }
```

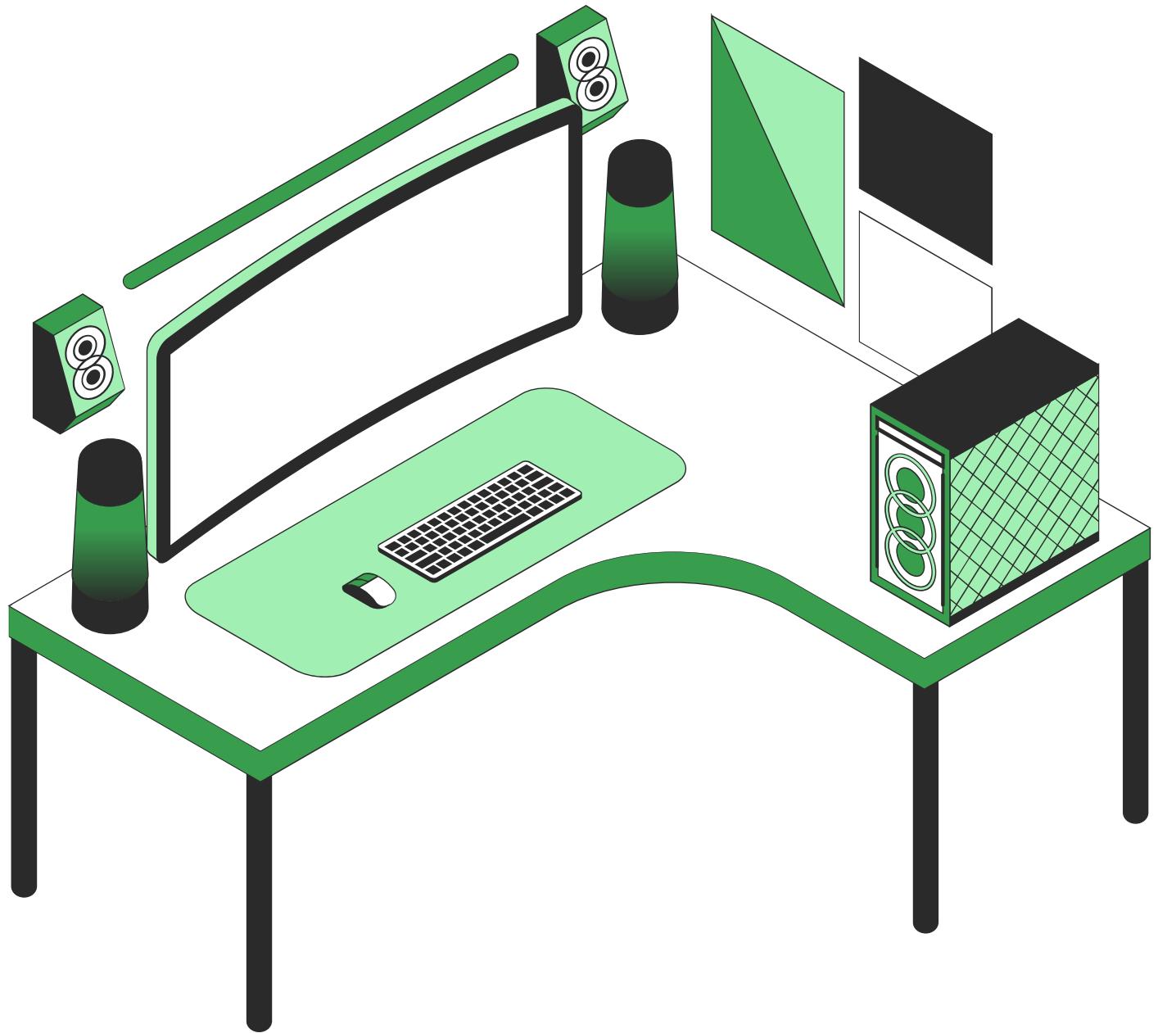


Conclusion





Source Code Web Digilab akan dibagi menjadi 2 bagian, yakni program Front End dan program Back End. Framework React Js, CSS Tailwind, dan Spline adalah framework utama yang dimanfaatkan untuk mengembangkan bagian Front End, sedangkan framework Node Js adalah framework utama yang dimanfaatkan untuk mengembangkan bagian Back End. Selainnya, database PostgreSQL dimanfaatkan sebagai media DBMS untuk Web Digilab yang ingin dikembangkan. Karakteristik Relational dari PostgreSQL menyebabkan database yang dikelola dapat dinormalisasi sehingga mencegah terjadinya redundansi data. Hal tersebut memberikan performa database yang lebih baik serta efisien apabila ingin melakukan operasi CRUD (Create, Read, Update, Delete) oleh Client.



Thank You
For Staying
With Us

Kelompok SBD 15