

MAKALAH
APLIKASI WEB SERVICE JSON DAN FOOTBALL API

**Diajukan untuk memenuhi tugas Ujian Akhir Semester (UAS) Mata Kuliah Aplikasi
Teknologi Online**

Dosen: Eko Budi Setiawan, S.Kom., M.T



Disusun Oleh :

Rifqi Pratama Juliansyah

10118322

IF-8

DOMAIN

<https://rifqipratama9.000webhostapp.com/>

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS KOMPUTER INDONESIA**

2020

DAFTAR ISI

DAFTAR ISI.....	i
DAFTAR GAMBAR	iii
DAFTAR TABEL.....	iv
DAFTAR SCRIPT	v
1. Pembahasan Membuat Aplikasi	1
1.1 Pembuatan Web Service.....	1
1.1.1 Membuat Database, Tabel Buku dan Insert Data Buku	1
1.1.2 Membuat Fitur Web Service JSON.....	4
1.1.2.1 Membuat Koneksi ke Database	4
1.1.2.2 Membuat Fitur Tampil Semua Data	6
1.1.2.3 Fitur Tambah Data	10
1.1.2.4 Fitur Tampil Data Berdasarkan ID.....	14
1.1.2.5 Fitur Update Data	16
1.1.2.6 Fitur Delete Data.....	19
1.1.2.7 Fitur Search Data.....	21
1.2 Football API	24
1.2.1 Mendapatkan Token API atau Key API.....	24
1.2.2 Pembuatan Fitur API	28
1.2.2.1 Klasemen Premier League	29
1.2.2.2 Daftar Klub Premier League	33
1.2.2.3 Detail Info Klub.....	36
1.2.2.4 Daftar Pertandingan Premier League.....	40
2. Pembahasan Aplikasi.....	43
3. Pembahasan Fitur Application Programming Interface.....	46
4. Pembahasan Fitur Web Service.....	51
4.1 ListAll.....	51
4.2 listId.....	52
4.3 Tambah Data	55
4.4 Update Data.....	57

4.6	Search Data.....	62
------------	-------------------------	-----------

DAFTAR GAMBAR

Gambar 1. Tabel Buku	3
Gambar 2. Halaman football-data.org.....	24
Gambar 3. Register Akun football-data.org.....	25
Gambar 4. Email API.....	26
Gambar 5. Login football-data.org	26
Gambar 6. Akun football-data.org	28
Gambar 7. Halaman Utama atau Klasemen	46
Gambar 8. Daftar Klub	47
Gambar 9. Klik Info Klub	47
Gambar 10. Klik Info Klub 2	48
Gambar 11. Info Klub	48
Gambar 12. Daftar Pemain	49
Gambar 13. Daftar Pemain 2	49
Gambar 14. Pertandingan.....	50
Gambar 15. Pertandingan 2.....	50
Gambar 16. listAll.....	51
Gambar 17. listAll Postman	52
Gambar 18. listId	53
Gambar 19. listId Postman.....	53
Gambar 20. listId Tidak Ada	54
Gambar 21. listId Tidak Ada Postman.....	54
Gambar 22. Tambah Postman.....	55
Gambar 23. Tambah Data Berhasil	56
Gambar 24. Tambah Data Harus Diisi.....	57
Gambar 25. List Setelah Tambah Data	57
Gambar 26. List Sebelum Update Data	58
Gambar 27. Update Data.....	58
Gambar 28. Update Berhasil.....	59
Gambar 29. Data Setelah Diupdate.....	59
Gambar 30. Delete Data.....	60
Gambar 31. Delete Data Berhasil	61
Gambar 32. List Setelah Dihapus	61
Gambar 33. Search Data	62
Gambar 34. Search Data Tidak Ditemukan	63

DAFTAR TABEL

Tabel 1. Tabel Buku..... 2

DAFTAR SCRIPT

Script 1. Membuat Database	1
Script 2. Membuat Tabel Buku	2
Script 3. Insert Data Buku.....	4
Script 4. Database.php	5
Script 5. buku.php (listAll)	8
Script 6. listAll.php	9
Script 7. Buku.php (tambah)	11
Script 8. tambah.php	13
Script 9. buku.php (listId)	14
Script 10. listId.php.....	16
Script 11. buku.php (update).....	17
Script 12. update.php	18
Script 13. buku.php (delete).....	20
Script 14. delete.php	21
Script 15. buku.php (search)	22
Script 16. search.php.....	23
Script 17. index.php	31
Script 18. klub.php.....	35
Script 19. info-klub.php	39
Script 20. pertandingan.php	42

1. Pembahasan Membuat Aplikasi

1.1 Pembuatan Web Service

Web Service Json yang dibuat adalah mengenai data buku, yang dimana nantinya bisa menampilkan seluruh data buku (listAll.php), menampilkan data buku sesuai id yang dipilih (listId.php), mencari data buku sesuai dengan judul buku (search.php), menambah data buku (tambah.php), mengubah data buku (update.php) dan menghapus data buku (delete.php).

Sebelum membuat fungsi-fungsi atau *method* diatas, kita buat *database*-nya, tabel dan atribut yang dibutuhkan terlebih dahulu.

1.1.1 Membuat Database, Tabel Buku dan Insert Data Buku

Membuat database dengan nama web_service_buku dengan script

```
CREATE DATABASE web_service_buku;
```

Script 1. Membuat Database

Lalu membuat tabel dengan nama buku dan atributnya seperti tabel di bawah ini:

Nama	Tipe Data	Key	Keterangan
Id_buku	Int(11)	Primary Key	Not null, auto_increment
Judul	Varchar(60)		Not null
Penulis	Varchar(60)		Not null
Penerbit	Varchar(50)		Not null
Deskripsi	Varchar(200)		Not null
Bahasa	Varchar(30)		Not null

Genre	Varchar(50)		Not null
Jumlah_halaman	Int(11)		Not null
Tahun_terbit	Varchar(5)		Not null

Tabel 1. Tabel Buku

Untuk pembuatan tabel buku sesuai dengan kriteria di atas kita menggunakan script seperti ini:

```
CREATE TABLE buku (
    id_buku INT NOT NULL auto_increment,
    judul VARCHAR(60) NOT NULL,
    penulis VARCHAR(60) NOT NULL,
    penerbit VARCHAR(50) NOT NULL,
    deskripsi VARCHAR(200) NOT NULL,
    bahasa VARCHAR(30) NOT NULL,
    genre VARCHAR(50) NOT NULL,
    jumlah_halaman INT NOT NULL,
    tahun_terbit VARCHAR(5) NOT NULL,
    PRIMARY KEY (id_buku)
);
```

Script 2. Membuat Tabel Buku

Dan jika sudah diimplementasikan atau dijalankan script di atas maka kira-kira struktur tabel buku yang kita punya akan seperti ini di database.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_buku	int(11)			No	None		AUTO_INCREMENT	Change Drop ▾ More
2	judul	varchar(60)	utf8mb4_general_ci		No	None			Change Drop ▾ More
3	penulis	varchar(60)	utf8mb4_general_ci		No	None			Change Drop ▾ More
4	penerbit	varchar(50)	utf8mb4_general_ci		No	None			Change Drop ▾ More
5	deskripsi	varchar(200)	utf8mb4_general_ci		No	None			Change Drop ▾ More
6	bahasa	varchar(30)	utf8mb4_general_ci		No	None			Change Drop ▾ More
7	genre	varchar(50)	utf8mb4_general_ci		No	None			Change Drop ▾ More
8	jumlah_halaman	int(11)			No	None			Change Drop ▾ More
9	tahun_terbit	varchar(5)	utf8mb4_general_ci		No	None			Change Drop ▾ More

Gambar 1. Tabel Buku

Selanjutnya kita masukkan data buku yang kita punya dengan script seperti ini:

INSERT INTO buku VALUE

- (1, "Aplikasi Internet Menggunakan HTML, CSS dan JavaScript", "Taryana Suryana & Koesheryatin", "Elex Media Komputindo", "Buku ini akan menuntun Anda dalam membuat aplikasi Internet secara lengkap. Sistematika pembahasan dalam buku lebih ditekankan pada segi praktikum", "Indonesia", "Komputer dan Teknologi", 242, "2018"),
- (2, "Islam dan Akal Merdeka", "Mohammad Natsir", "Sega Arsy", "Buku ini berisi tentang kritik atas pemikiran Soekarno tentang Islam Sontoloyo dan seputra pembaruan pemikiran Islam sekitar tahun 1934 - 1940", "Indonesia", "Sejarah Indonesia", 172, "2018"),
- (3, "Mengenal Microsoft Office 2013", "Taryana Suryana & Koesheryatin", "Elex Media Komputindo", "Buku ini adalah buku untuk mengenal tentang Microsoft Office 2013 yang disusun secara singkat dan sistematis menurut kebutuhan dari yang sederhana sampai cukup kompleks", "Indonesia", "Komputer dan Teknologi", 238, "2018"),
- (4, "Perangkat Keras Komputer", "Bobi Kurniawan, ST., M.Kom.", "Elex Media Komputindo", "Buku ini berisi untuk memberi pemahaman secara mendasar kepada pengguna dalam memahami bagian penting dari suatu komputer", "Indonesia", "Komputer dan Teknologi", 164, "2018"),
- (5, "Bumi Manusia", "Pramoedya Ananta Toer", "Lentera Dipantara", "Buku ini adalah seri dari Tetralogi Buru yang mengambil latarbelakang dan cikalbakal nation Indonesia di awal abad ke-20", "Indonesia", "Novel Roman", 551, "2018"),
- (6, "Guns, Germs and Steel", "Jared Diamond", "KPG", "Buku ini mengajak kita melihat riwayat peradaban manusia pada masa tepat sebelum masa sejarah

dimulai dari sekitar tahun 11000 SM dan perkembangannya sampai saat ini", "Indonesia", "Sejarah", 636, "2018"),
(7, "Pemrograman Berorientasi Objek Menggunakan Java", "Adam Mukhairil Bachtiar & Firman Nizammudin Fakhrul", "Penerbit Informatika", "Buku ini ditujukan untuk pengembang perangkat lunak pemula, mahasiswa, dosen, atau siapapun yang ingin mempelajari mengenai pemrograman Berorientasi objek", "Indonesia", "Komputer dan Teknologi", 262, "2018"),
(8, "Pemrograman C dan C++", "Adam Mukhairil Bachtiar", "Penerbit Informatika", "Buku ini ditujukan untuk mahasiswa atau siapapun yang ingin mempelajari lebih dalam mengenai konsep dan fundamental dari bahasa C dan C++", "Indonesia", "Komputer dan Teknologi", 324, "2017"),
(9, "Matematika Diskrit", "Rinaldi Munir", "Penerbit Informatika", "Buku ini disusun sebagai buku teks mahasiswa yang mengambil mata kuliah Matematika Diskrit", "Indonesia", "Komputer dan Teknologi", 548, "2016");

Script 3. Insert Data Buku

Setelah database, tabel sudah dibuat dan data buku sudah dimasukkan ke database, lalu langkah selanjutnya adalah mulai membuat fitur untuk web service json nya.

1.1.2 Membuat Fitur Web Service JSON

1.1.2.1 Membuat Koneksi ke Database

Kita buat folder projek dengan nama api dan di dalamnya kita buat tiga folder kembali dengan nama buku, config, dan models. Folder buku disini diisi dengan fitur-fitur atau bisa disebut dengan *controller*, untuk *config* sendiri isinya merupakan konfigurasi untuk koneksi ke *database*, dan untuk models sendiri merupakan model atau fungsi yang menyambungkan *controller* dengan *database*.

Pertama kita buat terlebih dahulu fungsi untuk menyambungkan aplikasi dengan *database*, buat file di dalam folder **/api/config** dengan nama *database.php*.

Script untuk database.php sendiri adalah seperti di bawah ini:

```
1. <?php
2. class Database {
3.
4.     private $host = "localhost";
5.     private $db = "web_service_buku";
6.     private $username = "root";
7.     private $pass = "";
8.
9.     public $koneksi;
10.
11.    public function getKoneksi() {
12.
13.        $this->koneksi = null;
14.
15.        try {
16.            $this->koneksi = new PDO("mysql:host=" . $this-
>host . ";dbname=" . $this->db, $this->username, $this->pass);
17.            $this->koneksi->exec("set names utf8");
18.        } catch (PDOException $e) {
19.            echo "Koneksi Error : " . $e->getMessage();
20.        }
21.
22.        return $this->koneksi;
23.    }
24.
25.
26. ?>
```

Script 4. Database.php

Pada *file* atau *script* **database.php** diatas terdapat sebuah *class* dengan nama *Database*. Class tersebut mempunyai beberapa variabel atau atribut seperti *host* yang mempunyai hak akses *private* dan berisi nama *host* dari *database*. Selanjutnya ada atribut atau variabel *db* yang mempunyai hak akses *private* juga dan berisi nama *database* yang kita pakai, disini saya menggunakan database yang sebelumnya sudah dibuat yaitu “*web_service_buku*”. Selanjutnya ada atribut atau variabel *username* yang sama-sama mempunyai hak akses *private* dan berisi nama dari *username* dari *database server* kita. Selanjutnya ada variable atau atribut *pass* yang berisi *password* untuk akses

ke *database server* kita. Dan yang terakhir adalah atribut atau variabel koneksi yang fungsinya untuk menampung hasil proses koneksi ke database.

Di dalam *class Database* juga terdapat *method* atau fungsi dengan nama `getKoneksi`. Method ini berfungsi untuk koneksi ke database. Di dalamnya menggunakan PHP PDO (PHP Data Object), alasan mengapa menggunakan PHP PDO adalah karena agar lebih efisien dan efektif, karena PDO sendiri tidak tergantung pada database server tertentu, mungkin suatu saat kita akan mengganti atau bermigrasi ke *database server* lainnya seperti PostgreSQL, kita hanya perlu mengganti cara pemanggilan awal PDO saja tanpa harus menambah ekstensi-ekstensi lain.

Karena PDO sendiri diakses dengan menggunakan objek maka hasil kembalinya (return) pun adalah berupa objek . Di dalam *method* ini pun kita memanggil menangkap (catch) kesalahan ketika melakukan koneksi ke *database*, maka apabila terjadi *error* saat melakukan koneksi ke *database*, method ini akan menampilkan pesan *error*.

Method ini mempunyai hak akses public, karena method ini akan digunakan oleh kelas-kelas lainnya untuk melakukan koneksi ke *database*.

1.1.2.2 Membuat Fitur Tampil Semua Data

Untuk membuat fitur tampil semua data, pertama kita buat file dengan nama `buku.php` pada folder `/api/models`. File `buku.php` ini berfungsi sebagai model atau penyambung antara controller

dengan database. Di dalam file ini terdapat method-method yang dibutuhkan untuk melakukan query ke database, seperti menampilkan data, mencari, mengubah, menghapus dan menambah.

Pertama-tama kita buat sebuah class dengan nama Buku di dalam file buku.php tersebut. Lalu, di dalam class tersebut kita deklarasikan atribut atau variabel apa saja yang kita butuhkan. Setelah itu kita buat konstruktor yang berfungsi untuk melakukan koneksi ke database.

Setelah semua itu selesai, kita buat method dengan nama listAll yang nantinya akan kita gunakan untuk menampilkan data seluruh buku. Method ini juga akan kita panggil pada konstruktor listAll.php pada folder /api/buku/.

Script dari buku.php sendiri adalah sebagai berikut:

```
1. class Buku {
2.
3.     // Atribut / variabel
4.     private $koneksi;
5.     private $table = "buku";
6.
7.     // Atribut / variabel
8.     public $id_buku;
9.     public $judul;
10.    public $penulis;
11.    public $penerbit;
12.    public $deskripsi;
13.    public $bahasa;
14.    public $genre;
15.    public $jumlah_halaman;
16.    public $tahun_terbit;
17.
18.    // Konstruktor koneksi db
19.    public function __construct($db) {
20.        $this->koneksi = $db;
21.    }
22.
23.    // Method atau fungsi untuk menampilkan semua data buku
24.    public function listAll() {
25.
26.        // Query sql
27.        $query = "SELECT * FROM ". $this-
>table ." ORDER BY id_buku ASC";
28.    }
```

```
29.     // Prepare statement
30.     $stmt = $this->koneksi->prepare($query);
31.
32.     // Eksekusi query
33.     $stmt->execute();
34.
35.     // Kembalikan hasil eksekusi query
36.     return $stmt;
37. }
38. }
```

Script 5. buku.php (listAll)

Di dalam *file* atau *script* tersebut kita melakukan deklarasi variabel yang akan diperlukan kedepannya, seperti koneksi, table yang berisi nama tabel yang akan kita tuju pada database, dan atribut-atribut lainnya seperti id_buku, judul, penulis, penerbit dan lainnya.

Di dalam class tersebut juga kita menggunakan konstruktor untuk membangun koneksi ke *database*. Setelah semua yang dibutuhkan sudah di deklarasi maka langkah selanjutnya kita adalah membuat sebuah *method* yaitu *listAll* yang berfungsi untuk menampilkan data seluruh buku dari tabel buku.

Di dalam method *listAll* ini kita melakukan query “SELECT” ke tabel buku pada database untuk mengambil seluruh data untuk ditampilkan dan method ini mengembalikan nilai *stmt* yang nantinya akan ditangkap oleh *controller* dari *listAll.php*

Setelah model sudah siap, langkah selanjutnya adalah membuat controller dengan nama *listAll.php* pada folder /api/buku/

Script dari controller *listAll.php* sendiri adalah sebagai berikut:

```
1. <?php
2.     header("Access-Control-Allow-Origin: *");
3.     header("Content-Type: application/json");
4.
5.     include_once '../config/database.php';
6.     include_once '../models/buku.php';
```

```

7.      // Instance database
8.      $database = new Database();
9.
10.     // Class koneksi db
11.     $db = $database->getKoneksi();
12.
13.     // Instance buku
14.     $buku = new Buku($db);
15.
16.     // query buku
17.     $stmt = $buku->listAll();
18.     $num = $stmt->rowCount();
19.
20.     // Jika ditemukan record lebih dari 0
21.     if($num > 0) {
22.         // Array buku
23.         $buku_arr = array();
24.         $buku_arr["buku"] = array();
25.
26.         // Mendapatkan data dari tabel database
27.         while($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
28.
29.             // Mendapatkan data buku
30.             $buku_item = array(
31.                 "id_buku" => $row['id_buku'],
32.                 "judul" => $row['judul'],
33.                 "penulis" => $row['penerbit'],
34.                 "penerbit" => $row['penulis'],
35.                 "deskripsi" => $row['deskripsi'],
36.                 "bahasa" => $row['bahasa'],
37.                 "genre" => $row['genre'],
38.                 "jumlah_halaman" => $row['jumlah_halaman'],
39.                 "tahun_terbit" => $row['tahun_terbit']
40.             );
41.
42.             // Tambahkan data ke record array
43.             array_push($buku_arr["buku"], $buku_item);
44.         }
45.
46.         // set respons kode 200 = OK
47.         http_response_code(200);
48.
49.         // Tampilkan data json dari array buku
50.         echo json_encode($buku_arr);
51.     } else {
52.         // set respons kode 404 = Not Found
53.         http_response_code(404);
54.
55.         // Tampilkan pesan
56.         echo json_encode(array("message" => "Belum ada data buku"));
57.     }
58. }
59. ?>

```

Script 6. listAll.php

Pada script di atas kita *set header* dengan *content-type* berupa *data json*, kemudian kita juga melakukan *instance* objek dari *class Database* agar dapat menggunakan method *getKoneksi* untuk melakukan koneksi ke *database*. Dan juga kita melakukan *instance* objek dari *class Buku* dengan parameter *db* yang isinya merupakan *method* untuk melakukan koneksi ke database.

Pada **listAll.php** kita memanggil *method* yang ada pada model yang telah dibuat yaitu **listAll**. Lalu kita buat *array* untuk menampung data hasil dari *fetch* dari tabel buku. Setelah mendapatkan data dari hasil *fetch* ke tabel buku, selanjutnya hasil dari *fetch* tersebut ditambahkan ke dalam *array* dengan nama **buku_arr**. Jika data ada / hasil *query* menunjukkan ada *record* lebih dari 0, maka *set http_response_code* dengan **200** yang artinya **OK!**, dan ubah *array_buku* menjadi bentuk *json* dan kemudian tampilkan.

Jika data hasil *query* tidak ada, maka *set http_response_code* dengan **404** yang artinya **not found** kemudian tampilkan pesan dengan format *json* dengan pesan gagal atau “Data buku belum ada”.

1.1.2.3 Fitur Tambah Data

Kita buat method pada model atau pada file *buku.php* yang berada pada folder */api/model/* dengan nama tambah.

Script dari method tambah sendiri adalah sebagai berikut:

```
1. // Method atau fungsi untuk tambah data buku
2. public function tambah() {
3.
4.     // Query sql
```

```

5.      $query = "INSERT INTO ". $this-
6.          >table . " SET judul=:judul, penulis=:penulis, penerbit=:penerbit,
7.                      deskripsi=:deskripsi, bahasa=:bahasa, genre=:genre, jumlah_
8.                      ah_halaman=:jumlah_halaman, tahun_terbit=:tahun_terbit";
9.      // Prepare statement
10.     $stmt = $this->koneksi->prepare($query);
11.     // Sanitize input browser
12.     $this->judul = htmlspecialchars(strip_tags($this->judul));
13.     $this->penulis = htmlspecialchars(strip_tags($this->penulis));
14.     $this->penerbit = htmlspecialchars(strip_tags($this-
15.         >penerbit));
16.     $this->deskripsi = htmlspecialchars(strip_tags($this-
17.         >deskripsi));
18.     $this->bahasa = htmlspecialchars(strip_tags($this->bahasa));
19.     $this->genre = htmlspecialchars(strip_tags($this->genre));
20.     $this->jumlah_halaman = htmlspecialchars(strip_tags($this-
21.         >jumlah_halaman));
22.     $this->tahun_terbit = htmlspecialchars(strip_tags($this-
23.         >tahun_terbit));
24.     // Binding isi
25.     $stmt->bindParam(":judul", $this->judul);
26.     $stmt->bindParam(":penulis", $this->penulis);
27.     $stmt->bindParam(":penerbit", $this->penerbit);
28.     $stmt->bindParam(":deskripsi", $this->deskripsi);
29.     $stmt->bindParam(":bahasa", $this->bahasa);
30.     $stmt->bindParam(":genre", $this->genre);
31.     $stmt->bindParam(":jumlah_halaman", $this->jumlah_halaman);
32.     $stmt->bindParam(":tahun_terbit", $this->tahun_terbit);
33.     // Eksekusi query
34.     if($stmt->execute()) {
35.         return true;
36.     }
37. }

```

Script 7. Buku.php (tambah)

Pada *method* tambah ini kita melakukan *query* ke *database* dengan operasi “*INSERT*” dan dengan *value* yang diambil dari inputan. Pada *method* ini juga dilakukan *sanitize* dari inputan dengan *htmlspecialchars* yang tujuannya adalah untuk menghindari dari serangan XSS (XSS Attack), dan kemudian kita lakukan *binding* pada *value* yang diinput. *Method* ini nantinya akan dipanggil pada *controller* dengan nama **tambah.php**.

Selanjutnya kita buat *file* untuk *controller* di dalam folder /api/buku/ dengan nama **tambah.php**.

Script dari **tambah.php** adalah sebagai berikut:

```
1. <?php
2.
3.     header("Access-Control-Allow-Origin: *");
4.     header("Content-Type: application/json");
5.     header("Access-Control-Allow-Method: POST");
6.
7.     include_once '../config/database.php';
8.     include_once '../models/buku.php';
9.
10.    // Instance objek database
11.    $database = new Database();
12.
13.    // Koneksi database
14.    $db = $database->getKoneksi();
15.
16.    // Instance objek buku
17.    $buku = new Buku($db);
18.
19.    // Mendapatkan input json
20.    $data = json_decode(file_get_contents("php://input"));
21.
22.    // Jika field tidak kosong
23.    if(!empty($data->judul) && !empty($data-
>penulis) && !empty($data->penerbit)
24.        && !empty($data->deskripsi) && !empty($data-
>bahasa) && !empty($data->genre)
25.        && !empty($data->genre) && !empty($data-
>jumlah_halaman) && !empty($data->tahun_terbit)) {
26.
27.        // Isi property buku
28.        $buku->judul = $data->judul;
29.        $buku->penulis = $data->penulis;
30.        $buku->penerbit = $data->penerbit;
31.        $buku->deskripsi = $data->deskripsi;
32.        $buku->bahasa = $data->bahasa;
33.        $buku->genre = $data->genre;
34.        $buku->jumlah_halaman = $data->jumlah_halaman;
35.        $buku->tahun_terbit = $data->tahun_terbit;
36.
37.        // Jika proses tambah berhasil ( true )
38.        if($buku->tambah()) {
39.            // Set respons kode 201 = created
40.            http_response_code(201);
41.
42.            // Tampilkan pesan berhasil
43.            echo json_encode(array("Message" => "Data Buku Berhasil Dita-
mbahkan"));
44.        } else {
45.            // Set respons kode 503 = service unavailable
46.            http_response_code(503);
47.
48.            // Tampilkan pesan gagal
```

```

49.          echo json_encode(array("Message" => "Data Buku Gagal Ditamba
50.          hkan"));
51.      }
52.    } else {
53.      // Jika data / field tidak lengkap
54.      http_response_code(404);
55.    }
56.    // Tampilkan pesan gagal
57.    echo json_encode(array("Message" => "Gagal! Semua Data Harus Di
58.    iisi"));
58. ?>

```

Script 8. tambah.php

Pada script **tambah.php** kita set *header* dengan **Content-Type** berupa *json* dan **Access-Control-Allow-Method** dengan **POST**. Seperti sebelumnya kita juga melakukan *instance* objek dari *class Database* dan juga *class Buku*. Kemudian kita juga melakukan validasi dari hasil inputan, apabila semua *field* tidak diisi atau ada *field* yang kosong maka tampilkan pesan berupa *json* dengan pesan “Gagal! Semua Data Harus Diisi” kemudian set **http_response_code** dengan status **404** atau *not found*.

Jika semua *field* sudah terisi maka isi *property* buku dengan hasil inputan kemudian lakukan proses tambah yang sebelumnya sudah kita buat pada bagian model. Jika proses tambah berhasil, set **http_response_code** dengan **201** yang artinya *created* dan tampilkan pesan berupa *json* dengan pesan “ Data Buku Berhasil Ditambahkan”. Jika proses tambah gagal set **http_response_code** dengan **503** yang artinya *service unavailable* dan kemudian tampilkan pesan berupa *json* dengan pesan “Data Buku Gagal Ditambahkan”.

1.1.2.4 Fitur Tampil Data Berdasarkan ID

Selanjutnya untuk membuat fitur tampil data berdasarkan id, kita tambahkan *method* dengan nama **listId** pada file model **buku.php** yang berada di dalam folder /api/model/.

Kita tambahkan *script* di bawah ini di dalam *class* Buku.

```
1. // Method atau fungsi untuk menampilkan data berdasarkan id ( parameter
)
2.     public function listId() {
3.
4.         // Query sql
5.         $query = "SELECT * FROM " . $this-
>table . " WHERE id_buku = ? LIMIT 0,1";
6.
7.         // Prepare statement
8.         $stmt = $this->koneksi->prepare($query);
9.
10.        // Binding isi / parameter
11.        $stmt->bindParam(1, $this->id_buku);
12.
13.        // Eksekusi query
14.        $stmt->execute();
15.
16.        // Fetch data hasil query
17.        $row = $stmt->fetch(PDO::FETCH_ASSOC);
18.
19.        // Masukkan isi hasil fetch data ke dalam objek
20.        $this->judul = $row['judul'];
21.        $this->penulis = $row['penulis'];
22.        $this->penerbit = $row['penerbit'];
23.        $this->deskripsi = $row['deskripsi'];
24.        $this->bahasa = $row['bahasa'];
25.        $this->genre = $row['genre'];
26.        $this->jumlah_halaman = $row['jumlah_halaman'];
27.        $this->tahun_terbit = $row['tahun_terbit'];
28.    }
```

Script 9. buku.php (listId)

Pada *method* ini kita melakukan query ke *database* dengan operasi “SELECT” dan dengan klausa *id_buku*. Di dalam *method* ini juga kita melakukan binding dari inputan yaitu *id_buku*. Kita melakukan *fetch* data dari tabel buku dan memasukkan *value* atau nilai dari hasil *fetch* tersebut ke dalam

objek atau *property* buku. Method ini akan digunakan pada file atau *controller* **listId.php**.

Selanjutnya kita buat file `listId.php` pada folder `/api/buku/`.

Untuk script `listId.php` adalah sebagai berikut:

```
1. <?php
2.     header("Access-Control-Allow-Origin: *");
3.     header("Content-Type: application/json");
4.
5.     include_once '../config/database.php';
6.     include_once '../models/buku.php';
7.
8.     // Instance database
9.     $database = new Database();
10.
11.    // Class koneksi ke database
12.    $db = $database->getKoneksi();
13.
14.    // Instance buku
15.    $buku = new Buku($db);
16.
17.    // Set id_buku untuk mendapatkan data yang akan ditampilkan
18.    $buku->id_buku = isset($_GET['id']) ? $_GET['id'] : die();
19.
20.    // Method tampilkan data buku berdasarkan id
21.    $buku->listId();
22.
23.    if($buku->judul != null) {
24.
25.        // Array buku
26.        $buku_arr = array(
27.            "id_buku" => $buku->id_buku,
28.            "judul" => $buku->judul,
29.            "penulis" => $buku->penerbit,
30.            "penerbit" => $buku->penulis,
31.            "deskripsi" => $buku->deskripsi,
32.            "bahasa" => $buku->bahasa,
33.            "genre" => $buku->genre,
34.            "jumlah_halaman" => $buku->jumlah_halaman,
35.            "tahun_terbit" => $buku->tahun_terbit
36.        );
37.
38.        // Set response kode 200 = OK!
39.        http_response_code(200);
40.
41.        // Format array menjadi json
42.        echo json_encode($buku_arr);
43.    } else {
44.        // Set response kode 404 = Not Found
45.        http_response_code(404);
46.
47.        // Tampilkan pesan buku tidak ditemukan
48.        echo json_encode(array("Message" => "Data buku Tidak Ditemukan!")
49.    }
}
```

| 50. ?>

Script 10. listId.php

Pada *script* ini kita atur header dengan **Content-Type** berupa *json* dan seperti sebelumnya kita juga melakukan *import file database.php* dan juga **buku.php** agar kita dapat melakukan *instance* objek *Database* dari *class Database* dan juga objek *Buku* dari *class Buku*.

Di dalam *script* ini kita mengatur parameter id yang di dapat dari *browser*. Lalu kita panggil method **listId** pada model yang sebelumnya sudah dibuat. Dan jika proses dijalankan dan data buku ada, maka tambahkan data buku tersebut ke dalam *array* dengan nama *buku_arr*, lalu set **http_response_code** dengan **200** yang artinya **OK!** Dan kita tampilkan dan konversi data *array* tersebut ke dalam bentuk *json*.

Jika buku yang didapat berdasarkan parameter id tidak ada atau tidak ditemukan, maka set **http_response_code** dengan **404** dan tampilkan pesan berupa *array* dengan pesan “Data Buku Tidak Ditemukan”.

1.1.2.5 Fitur Update Data

Buat method dengan nama update di dalam file model *buku.php* yang berada di dalam folder */api/model/*. Tambahkan script di bawah ini di dalam *class Buku* yang sebelumnya sudah dibuat.

```
1. // Method atau fungsi untuk update data buku
2. public function update() {
3.
4.     // Query sql
5.     $query = "UPDATE " . $this->table . "
6.             SET judul=:judul, penulis=:penulis,
```

```

7.          penerbit=:penerbit, deskripsi=:deskripsi,
8.          bahasa=:bahasa, genre=:genre, jumlah_halaman=:jumlah_hal-
9.          man,
10.         tahun_terbit=:tahun_terbit
11.         WHERE id_buku=:id_buku";
12. 
13.     // Prepare statement
14.     $stmt = $this->koneksi->prepare($query);
15. 
16.     // Sanitize input
17.     $this->judul = htmlspecialchars(strip_tags($this->judul));
18.     $this->penulis = htmlspecialchars(strip_tags($this->penulis));
19.     $this->penerbit = htmlspecialchars(strip_tags($this-
20.         >penerbit));
21.     $this->deskripsi = htmlspecialchars(strip_tags($this-
22.         >deskripsi));
23.     $this->bahasa = htmlspecialchars(strip_tags($this->bahasa));
24.     $this->genre = htmlspecialchars(strip_tags($this->genre));
25.     $this->jumlah_halaman = htmlspecialchars(strip_tags($this-
26.         >jumlah_halaman));
27.     $this->tahun_terbit = htmlspecialchars(strip_tags($this-
28.         >tahun_terbit));
29.     $this->id_buku = htmlspecialchars(strip_tags($this->id_buku));
30. 
31.     // Binding isi
32.     $stmt->bindParam(":judul", $this->judul);
33.     $stmt->bindParam(":penulis", $this->penulis);
34.     $stmt->bindParam(":penerbit", $this->penerbit);
35.     $stmt->bindParam(":deskripsi", $this->deskripsi);
36.     $stmt->bindParam(":bahasa", $this->bahasa);
37.     $stmt->bindParam(":genre", $this->genre);
38.     $stmt->bindParam(":jumlah_halaman", $this->jumlah_halaman);
39.     $stmt->bindParam(":tahun_terbit", $this->tahun_terbit);
40.     $stmt->bindParam(":id_buku", $this->id_buku);
41. 
42.     // Eksekusi statement atau query
43.     if($stmt->execute()) {
44.         return true;
45.     }
46. 
47.     return false;
48. }
```

Script 11. buku.php (update)

Pada *method* ini kita melakukan *query* ke *database* dengan operasi “UPDATE” dan dengan isi berupa *field* yang berada pada file buku. Kemudian dari inputan dilakukan *sanitize* untuk menghindari dari serangan XSS (XSS Attack), lalu kita *binding* isi inputan tersebut. Method ini akan digunakan pada file *controller* dengan nama **update.php**

Selanjutnya buat file update.php di dalam folder /api/buku/. File tersebut berisi script seperti di bawah ini

```
1. <?php
2.     header("Access-Control-Allow-Origin: *");
3.     header("Content-Type: application/json");
4.     header("Access-Control-Allow-Methods: POST");
5.
6.     include_once '../config/database.php';
7.     include_once '../models/buku.php';
8.
9.     // Instance objek database
10.    $database = new Database();
11.
12.    // Koneksi database
13.    $db = $database->getKoneksi();
14.
15.    // Instance objek buku
16.    $buku = new Buku($db);
17.
18.    // Mengambil data yang akan di edit
19.    $data = json_decode(file_get_contents("php://input"));
20.
21.    // Mendapat id buku yang akan diedit
22.    $buku->id_buku = $data->id_buku;
23.
24.    // Set nilai yang akan diedit
25.    $buku->judul = $data->judul;
26.    $buku->penulis = $data->penulis;
27.    $buku->penerbit = $data->penerbit;
28.    $buku->deskripsi = $data->deskripsi;
29.    $buku->bahasa = $data->bahasa;
30.    $buku->genre = $data->genre;
31.    $buku->jumlah_halaman = $data->jumlah_halaman;
32.    $buku->tahun_terbit = $data->tahun_terbit;
33.
34.    // Jika update berhasil
35.    if($buku->update()) {
36.
37.        // Set respons kode 200 = OK!
38.        http_response_code(200);
39.
40.        // Tampilkan pesan berhasil
41.        echo json_encode(array("Message" => "Data buku berhasil diupdate
   "));
42.    } else {
43.
44.        // Jika update gagal
45.        // Set respons kode 503 = service unavailable
46.        http_response_code(503);
47.
48.        // Tampilkan pesan gagal
49.        echo json_encode(array("Message" => "Data buku gagal diupdate"))
   ;
50.    }
51. ?>
```

Script 12. update.php

Pada script ini kita tetapkan header dengan **Content-Type** berupa *json* dan **Acess-Control-Allow-Methods** dengan **POST**. Seperti sebelum-sebelumnya disini juga melakukan *instance* objek dari *class Database* dan *class Buku*.

Data yang akan diupdate diambil dari inputan dan lalu dari data tersebut lakukan proses update yang sebelumnya sudah dibuat pada bagian model dengan nama method update. Jika proses *update* berhasil set **http_response_code** dengan **200** dan tampilkan pesan berupa *json* “Data Buku Berhasil diupdate”. Jika proses *update* gagal set **http_response_code** dengan **503** dan tampilkan pesan gagal berupa *json* “Data buku gagal diupdate”.

1.1.2.6 Fitur Delete Data

Buat method dengan nama delete di dalam file buku.php yang berada di dalam folder /api/model/. Script dari method delete tersebut adalah seperti ini:

```
1. // Method atau fungsi untuk delete data buku
2. public function delete() {
3.
4.     // Query sql
5.     $query = "DELETE FROM " . $this->table . " WHERE id_buku = ?";
6.
7.     // Prepare statement
8.     $stmt = $this->koneksi->prepare($query);
9.
10.    // Sanitize isi
11.    $this->id = htmlspecialchars(strip_tags($this->id_buku));
12.
13.    // Binding isi
14.    $stmt->bindParam(1, $this->id);
15.
16.    // Eksekusi query
17.    if($stmt->execute()) {
18.        return true;
19.    }
20.}
```

```
21.         return false;
22.     }
```

Script 13. buku.php (delete)

Di dalam script tersebut kita melakukan *query* dengan operasi “DELETE” dan dengan klausa berupa id_buku. Id buku tersebut didapat dari inputan. Di dalam *method* ini juga kita melakukan *sanitize* dari inputan agar terhindar dari serangan XSS (XSS Attack) dan juga melakukan *binding* dari inputan.

Selanjutnya kita buat file controller dengan nama delete.php di dalam folder /api/buku/. Script dari file tersebut adalah sebagai berikut:

```
1. <?php
2.     header("Access-Control-Allow-Origin: *");
3.     header("Content-Type: application/json");
4.     header("Access-Control-Allow-Methods: POST");
5.
6.     include_once '../config/database.php';
7.     include_once '../models/buku.php';
8.
9.     // Instance database
10.    $database = new Database();
11.
12.    // Koneksi ke database
13.    $db = $database->getKoneksi();
14.
15.    // Instance buku
16.    $buku = new Buku($db);
17.
18.    // Mendapatkan data inputan
19.    $data = json_decode(file_get_contents("php://input"));
20.
21.    $buku->id_buku = $data->id_buku;
22.
23.    // Jika buku berhasil dihapus
24.    if($buku->delete()) {
25.        // set respons kode 200 = OK!
26.        http_response_code(200);
27.
28.        // Tampilkan pesan berhasil
29.        echo json_encode(array("Message" => "Data buku berhasil dihapus"
30.    ));
31.    } else {
32.        // set response kode 503 = service unavailable
33.        http_response_code(503);
34.    }
35.    // Tampilkan pesan gagal
```

```
35.         echo json_encode(array("Message" => "Data buku gagal dihapus"));
36.     }
37. ?>
```

Script 14. delete.php

Di dalam *script delete.php* ini kita mengatur *header* dengan **Content-Type** berupa **json** dan **Access-Control-Allow-Methods** berupa **POST**. Seperti sebelumnya kita juga melakukan *instance* objek dari *class Database* dan *class Buku*.

Lalu kita lakukan proses *delete* yang sebelumnya sudah dibuat pada bagian model dengan nama *method delete*. Jika proses *delete* ini berhasil (true) maka set **http_response_code** dengan **200** dan tampilkan pesan berupa *json* “Data buku berhasil dihapus”. Jika proses delete gagal atau tidak berhasil, maka set **http_response_code** dengan **503** dan tampilkan pesan berupa *json* “Data buku gagal dihapus”

1.1.2.7 Fitur Search Data

Buat *method* dengan nama *search* pada *file buku.php* yang berada di dalam folder */api/model/*. *Script* method *search* sendiri adalah sebagai berikut:

```
1. // Method atau fungsi cari data berdasarkan parameter
2. function search($keywords){
3.
4.     // Query sql
5.     $query = "SELECT * FROM ".$this->table . " WHERE judul LIKE ?";
6.
7.     // Prepare statement
8.     $stmt = $this->koneksi->prepare($query);
9.
10.    // Sanitize isi
11.    $keywords=htmlspecialchars(strip_tags($keywords));
```

```

12.     $keywords = "%{$keywords}%" ;
13.
14.     // Bind isi
15.     $stmt->bindParam(1, $keywords);
16.
17.     // Eksekusi query
18.     $stmt->execute();
19.
20.     // Kembalikan nilai statement atau hasil eksekusi
21.     return $stmt;
22. }
```

Script 15. buku.php (search)

Pada method atau fungsi search ini kita melakukan query dengan operasi “SELECT” dan dengan klausa “LIKE”. Di dalam method ini juga kita mengambil parameter dengan nama keywords.

Selanjutnya kita buat file controller dengan nama search.php di dalam folder /api/buku/. Isi dari file tersebut adalah sebagai berikut:

```

1. <?php
2.
3.     header("Access-Control-Allow-Origin: *");
4.     header("Content-Type: application/json; charset=UTF-8");
5.
6.     include_once '../config/database.php';
7.     include_once '../models/buku.php';
8.
9.     // Instance database
10.    $database = new Database();
11.
12.    // Koneksi database
13.    $db = $database->getKoneksi();
14.
15.    // Instance objek buku
16.    $buku = new Buku($db);
17.
18.    // Mendapatkan keyword untuk data yang akan dicari
19.    $keywords=isset($_GET["s"]) ? $_GET["s"] : "";
20.
21.    // Query
22.    $stmt = $buku->search($keywords);
23.    $num = $stmt->rowCount();
24.
25.    // Jika record data lebih dari 0
26.    if($num>0){
27.
28.        // Buat array buku
```

```

29.     $buku_arr=array();
30.     $buku_arr["buku"]=array();
31.
32.     // Mengambil data dari tabel
33.     while ($row = $stmt->fetch(PDO::FETCH_ASSOC)){
34.
35.         $buku_item = array(
36.             "id_buku" => $row['id_buku'],
37.             "judul" => $row['judul'],
38.             "penulis" => $row['penulis'],
39.             "penerbit" => $row['penerbit'],
40.             "deskripsi" => $row['deskripsi'],
41.             "bahasa" => $row['bahasa'],
42.             "genre" => $row['genre'],
43.             "jumlah_halaman" => $row['jumlah_halaman'],
44.             "tahun_terbit" => $row['tahun_terbit']
45.         );
46.
47.         // Tambahkan data ke array
48.         array_push($buku_arr["buku"], $buku_item);
49.     }
50.
51.     // Set respons kode 200 = OK!
52.     http_response_code(200);
53.
54.     // Format array ke json
55.     echo json_encode($buku_arr);
56. } else {
57.     // Jika data tidak ada
58.     // Set respons kode 404 = not found
59.     http_response_code(404);
60.
61.     // Tampilkan pesan
62.     echo json_encode(array("Message" => "Data buku tidak ditemukan"))
63. }
64. ?>

```

Script 16. search.php

Di dalam *script search.php* kita mengatur *header* dengan **Content-Type** berupa **json**, dan seperti sebelumnya kita juga melakukan *instance* objek dari *class Database* dan *class Buku*. Di dalam script ini juga kita menangkap parameter dengan nama *keywords*, lalu kita lakukan proses *search* dengan parameter *keywors* tersebut.

Jika *record* data yang dicari ada, maka kita buat *array* buku lalu kita lakukan *fetch* data ke *database*, dan tambahkan data hasil

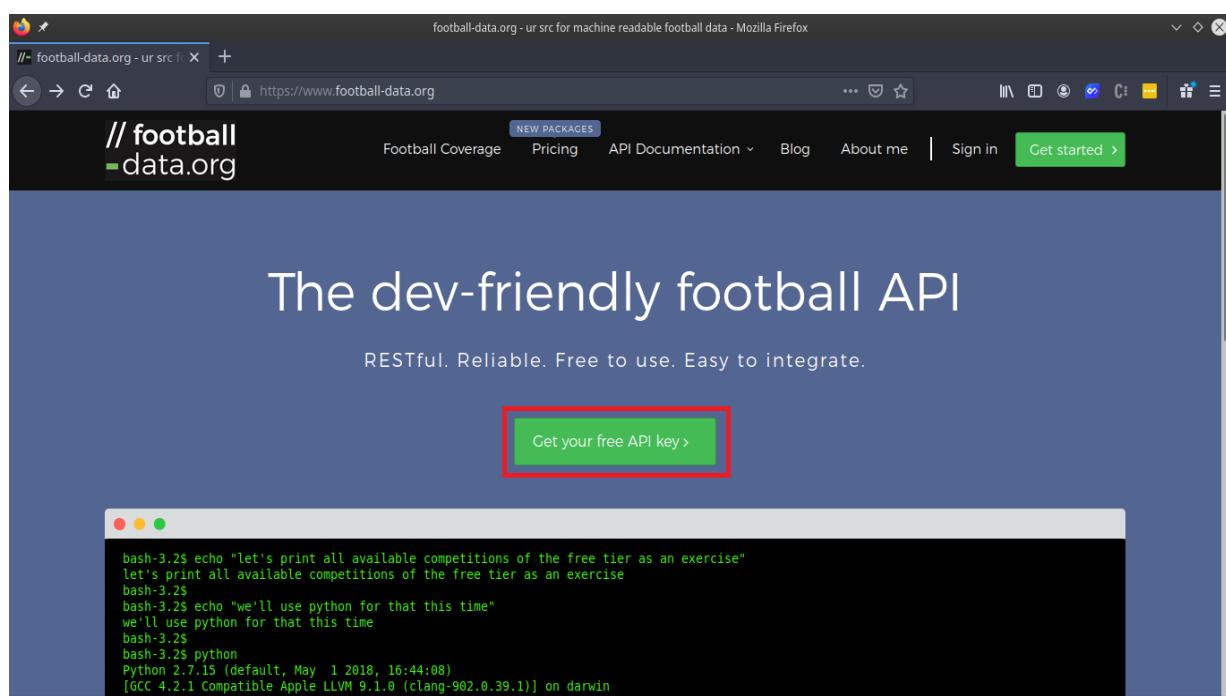
fetch tersebut ke dalam *array* dengan nama **buku_item**, lalu set **http_response_code** dengan **200** dan lakukan format data *array* **buku_arr** ke dalam bentuk *json*. Jika *record* data yang dicari tidak ada, maka set **http_response_code** dengan **404** dan tampilkan pesan berupa *json* “Data buku tidak ditemukan”.

1.2 Football API

Selanjutnya adalah cara menampilkan Football API dari <https://www.football-data.org/> ke dalam website kita. Langkah pertama adalah dengan mendapatkan token API terlebih dahulu.

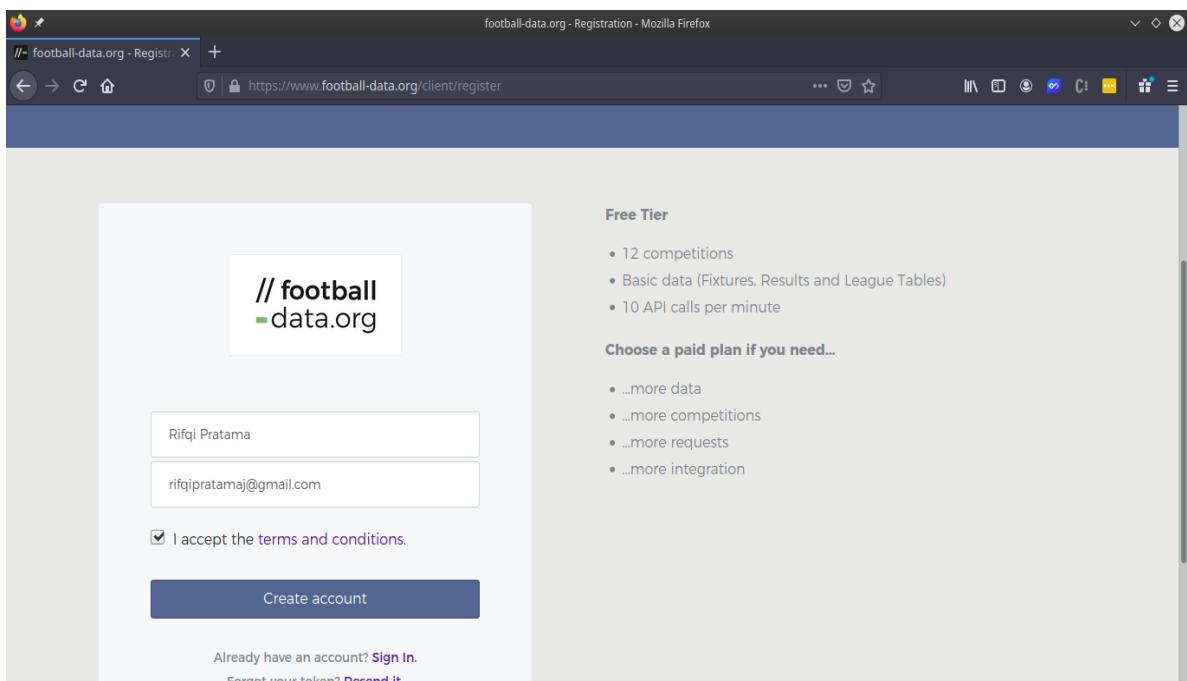
1.2.1 Mendapatkan Token API atau Key API

Pertama, melakukan daftar akun terlebih dahulu. Masuk ke halaman website <https://www.football-data.org/>, klik pada bagian *Get your free API key*.



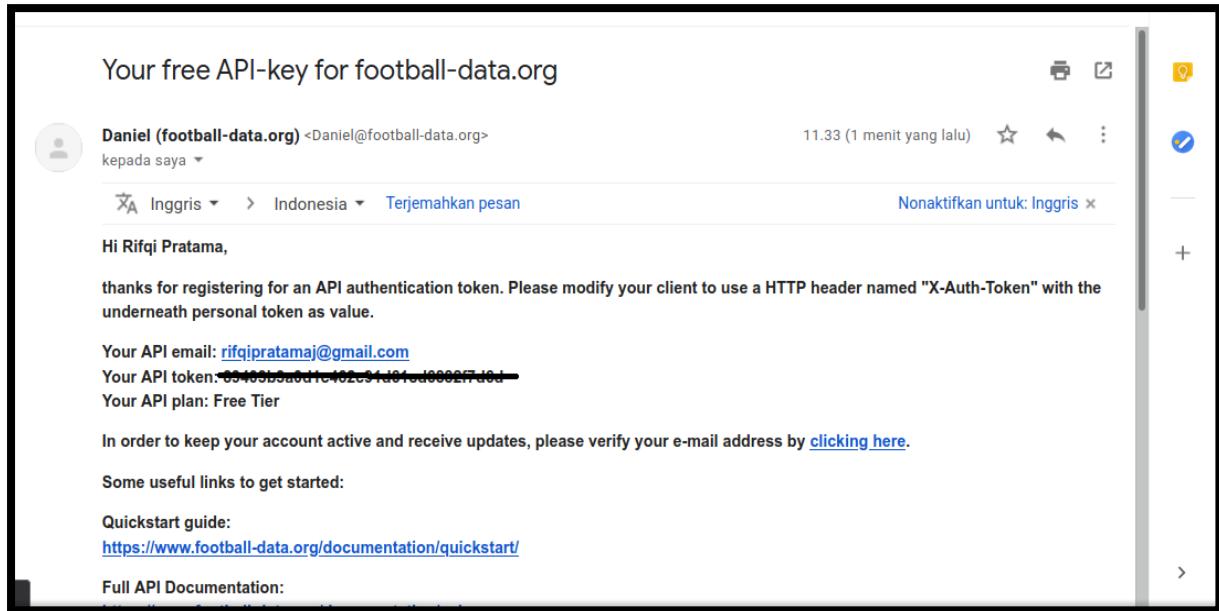
Gambar 2. Halaman football-data.org

Lalu, kita akan ditujukan pada halaman register akun. Jika sudah mempunyai akun bisa langsung melakukan sign in tetapi jika belum kita lakukan register akun terlebih dahulu. Caranya adalah dengan memasukkan nama dan email.



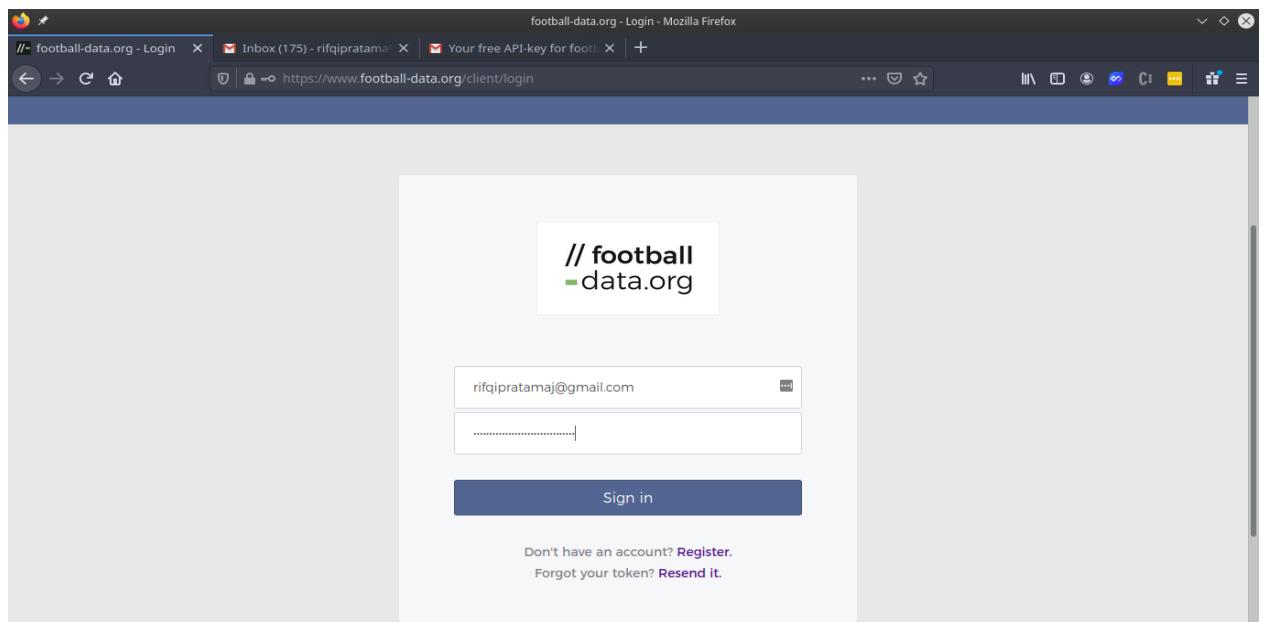
Gambar 3. Register Akun football-data.org

Setelah kita melakukan *register* akun, maka **football-data.org** akan mengirimkan *email* pada *email* yang tadi kita daftarkan. Isi dari *email* tersebut adalah **API email**, **API token**, dan **API plan**. *Email* tersebut juga member tahu untuk melakukan modifikasi saat akan melakukan pemanggilan API, yaitu dengan menambahkan **HTTP header** dengan key “**X-Auth-Token**” dan diisi dengan **token API** yang telah kita dapatkan.



Gambar 4. Email API

Sesudah kita mendapatkan token API tersebut, maka akun yang tadi kita daftarkan sudah aktif, langkah selanjutnya adalah melakukan *login*. Pada saat melakukan *login* kita hanya diminta untuk mengisi *email* dan token API kita sebagai passwordnya.

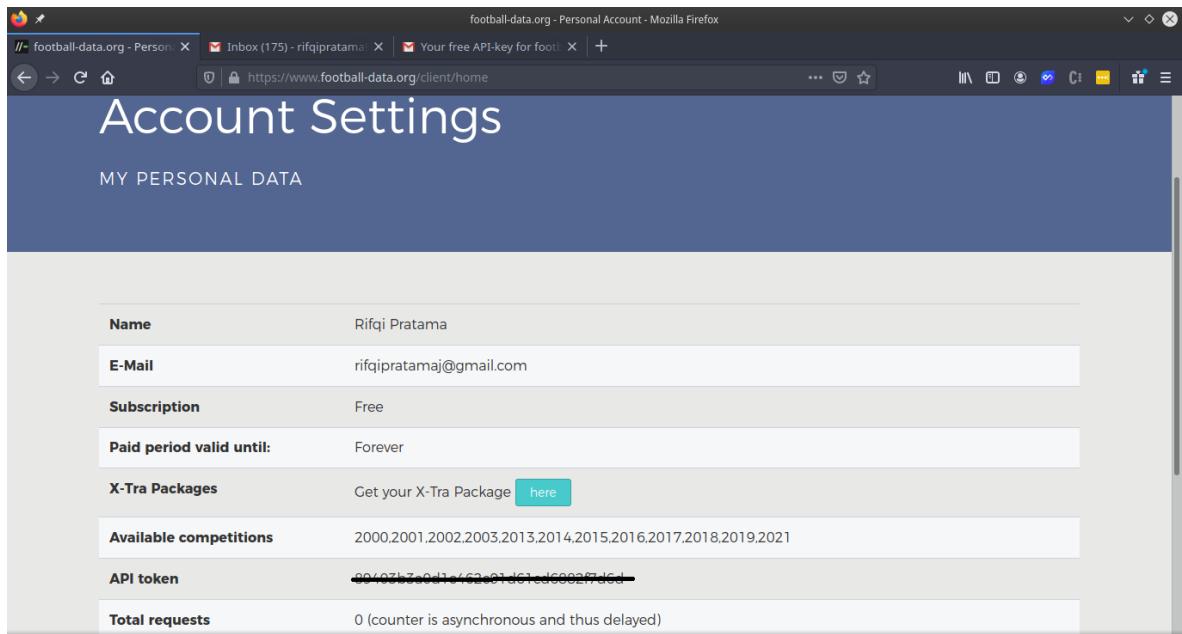


Gambar 5. Login football-data.org

Jika proses *login* berhasil, maka kita akan dilanjutkan pada halaman informasi akun kita. Di dalam halaman akun kita tersebut terdapat beberapa informasi seperti token API kita, total *request* yang sudah dilakukan, *plan API*, dan kode atau id dari kompetisi sepak bola yang ada. Terdapat beberapa id atau kode kompetisi sepak bola yang tersedia untuk versi gratis. Diantaranya adalah:

- 2000 : Fifa World Cup
- 2001 : UEFA Champions League
- 2002 : Liga Jerman
- 2003 : Liga Belanda
- 2013 : Liga Brazil
- 2014 : Liga Spanyol
- 2015 : Liga Perancis
- 2016 : Liga Inggris Champions
- 2017 : Liga Portugal Primeira Liga
- 2018 : Europa Championship
- 2019 : -
- 2021 : Premier League

Disini saya menggunakan kompetisi Premier League dengan kode 2021.



Gambar 6. Akun football-data.org

Untuk dokumentasi API yang lebih lengkap dapat mengunjungi halaman website atau situs dari football-data.org sendiri pada link <https://www.football-data.org/> atau <https://www.football-data.org/documentation/api>

1.2.2 Pembuatan Fitur API

Untuk memudahkan *styling* pada halaman website, disini menggunakan *css framework*, yaitu **materialize-css**. Caranya, pertama kita tambahkan pada folder projek kita, folder dengan nama js dan css. Lalu setelah itu kita *download* **materialize-css** nya pada link <https://materializecss.com/getting-started.html>. Setelah download selesai, ekstrak file **materialize-css** nya lalu pada *folder* bagian js, kita copy file **materialize.min.js** ke dalam folder js projek kita. Dan lakukan juga pada bagian *folder* css **materialize-css**, copy file **materialize.min.css** pada folder css projek kita.

1.2.2.1 Klasemen Premier League

Buat file dengan nama **index.php** pada di dalam folder projek kita. File **index.php** ini akan berfungsi untuk menampilkan data klasemen sementara yang ada pada *premier league*.

Script **index.php** adalah sebagai berikut:

```
1. <?php
2.     $url = 'https://api.football-
3.         data.org/v2/competitions/2021/standings';
4.         $token = '89403b3a0d1c462c91d61cd6882f7d6d';
5. 
6.         // Inisialisasi
7.         $curl = curl_init($url);
8. 
9.         // Set HTTP Header : Token
10.        curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
11.        curl_setopt($curl, CURLOPT_HTTPHEADER, [
12.            'X-Auth-Token: '.$token.''
13.        ]);
14. 
15.        // Mendapatkan respons
16.        $response = json_decode(curl_exec($curl), true);
17. 
18.        // Menutup CURL
19.        curl_close($curl);
20. 
21.        // Mendapatkan array tentang klasemen
22.        $result = $response['standings'];
23. 
24.        // Mendapatkan array tentang kompetisi
25.        $update = $response['competition'];
26. 
27. <!DOCTYPE html>
28. <html lang="en">
29. <head>
30.     <meta charset="UTF-8">
31.     <meta name="viewport" content="width=device-width, initial-
32.         scale=1.0">
33.         <!-- Google Icon Font -->
34.         <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
35.             rel="stylesheet">
36.             <!-- My CSS -->
37.             <link rel="stylesheet" href="css/style.css">
38. 
39.             <!-- Materialize CSS -->
40.             <link type="text/css" rel="stylesheet" href="css/materialize.min.css"
41.                 " media="screen,projection"/>
```

```

42.      <title>Klasemen Premier League</title>
43.  </head>
44.  <body>
45.
46.      <!-- Navigasi -->
47.
48.      <nav class="indigo darken-4">
49.          <div class="nav-wrapper container">
50.              <a href="#" class="brand-logo">Premier League</a>
51.              <a href="#" data-target="mobile-demo" class="sidenav-trigger"><i class="material-icons">menu</i></a>
52.
53.              <ul class="right hide-on-med-and-down">
54.                  <li><a href="index.php">Klasemen</a></li>
55.                  <li><a href="klub.php">Daftar Klub</a></li>
56.                  <li><a href="pertandingan.php">Pertandingan</a></li>
57.              </ul>
58.          </div>
59.      </nav>
60.
61.      <ul class="sidenav" id="mobile-demo">
62.          <li><a href="index.php">Klasemen</a></li>
63.          <li><a href="klub.php">Daftar Klub</a></li>
64.          <li><a href="pertandingan.php">Pertandingan</a></li>
65.      </ul>
66.
67.      <div class="row">
68.          <h2 class="center-align">Klasemen Premier League</h2>
69.          <p class="center-
    align">Terakhir Update: <strong><?= $update['lastUpdated']; ?></strong><
/p>
70.          <table class="klasemen responsive-table" id="tabel-klasemen">
71.              <thead>
72.                  <tr>
73.                      <th class="center-align"> No.</th>
74.                      <th colspan="2" class="center-align"> Klub</th>
75.                      <th class="center-align"> Main</th>
76.                      <th class="center-align"> Menang</th>
77.                      <th class="center-align"> Seri </th>
78.                      <th class="center-align"> Kalah </th>
79.                      <th class="center-align"> GM </th>
80.                      <th class="center-align"> GK</th>
81.                      <th class="center-align"> SG </th>
82.                      <th class="center-align"> Poin </th>
83.                  </tr>
84.
85.              </thead>
86.              <tbody>
87.
88.                  <?php if(!empty($result)) { ?>
89.                      <?php foreach($result as $res) : ?>
90.                          <?php $season = $res['type']; ?>
91.                          <?php $match = $res['table']; ?>
92.
93.                              <?php if($season == "TOTAL") { ?>
94.                                  <?php foreach($match as $m) : ?>
95.                                      <?php $image = $m['team']['crestUrl']; ?>
96.                                          <?php $image = preg_replace("/^http:/i",
    "https:", $image); ?>
97.

```

```

98.                                     <tr>
99.                                         <td >
100.                                         <td align="center-"
101.                                             <td align="center-"
102.                                                 "><?= $team['name']; ?></a>
107.                                         <td align="center-"
108.                                             <td align="center-"
109.                                                 <td align="center-"
110.                                                 <td align="center-"
111.                                                 <td align="center-"
112.                                                 <td align="center-"
113.                                                 <td align="center-"
114.                                                 <td align="center-"
115.                                         </tr>
116.                                         <?php endforeach; ?>
117.                                         <?php } ?>
118.                                         <?php endforeach; ?>
119.                                         <?php } ?>
120.                                         </tbody>
121.                                         </table>
122.                                         </div>
123.
124.                                         <script src="js/materialize.min.js"></script>
125.                                         <script>
126.                                             // Sidenav
127.                                             document.addEventListener('DOMContentLoaded', function()
128.     {
129.         var elems = document.querySelectorAll('.sidenav');
130.         M.Sidenav.init(elems);
131.     });
132.                                         </script>
133.                                         </body>
134.                                         </html>

```

Script 17. index.php

Pada script **index.php** diatas kita melakukan *request API* dengan menggunakan *cURL*. Untuk mendapatkan data klasemen dari kompetisi premier league sendiri adalah melalui <https://api.footballdata.org/v2/competitions/{id}>

Dimana pada bagian {id} kita isi dengan kode atau id dari kompetisi yang disediakan yang telah disebutkan pada bagian sebelumnya. Disini kita menggunakan id : 2021 untuk mendapatkan data kompetisi dari premier league.

Dan untuk mendapatkan data klasemen dari premier league sendiri adalah dengan menambahkan /standings di belakangnya, jadi url lengkapnya adalah seperti ini: <https://api.football-data.org/v2/competitions/2021/standings>.

Di dalam script ini juga kita mengatur **HTTP header** dengan nama atau *key* “**X-Auth-Token**” yang *value*-nya adalah token API kita. Lalu, kita lakukan eksekusi *cURL* untuk mendapatkan data *json* dari API kita. Untuk mendapatkan array tentang klasemen sendiri kita hanya cukup mengambil dari response dengan *index standings*, dan untuk informasi kompetisi kita hanya cukup mengakses dari response dengan *index competition*.

Untuk menampilkan data hasil *response* tersebut ke dalam *html*, karena hasil dari *response* adalah dalam berbentuk *array* maka kita dilakukan dengan *foreach* untuk mendapatkan hasil dari datanya, untuk gambar sendiri atau logo klub kita harus menggantinya dulu dari *http* menjadi *https* dengan menggunakan *preg_replace* milik PHP.

Pada bagian *script* juga kita meninisialisasi *materialize* untuk mengaktifkan *sidenav* ketika website ditampilkan pada mobile.

1.2.2.2 Daftar Klub Premier League

Buat file dengan nama **klub.php** di dalam folder projek kita. File ini nantinya yang akan menampilkan daftar klub apa saja yang berada pada *Premier League*.

Script dari **klub.php** adalah sebagai berikut:

```
1. <?php
2.     // API URL
3.     $url = 'https://api.football-data.org/v2/competitions/2021/teams';
4.
5.     // API Token
6.     $token = '89403b3a0d1c462c91d61cd6882f7d6d';
7.
8.     // Inisialisasi CURL
9.     $curl = curl_init($url);
10.
11.    // Set Header Auth Token
12.    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
13.    curl_setopt($curl, CURLOPT_HTTPHEADER, [
14.        'X-Auth-Token: '.$token.''
15.    ]);
16.
17.    // Mendapatkan respons
18.    $response = json_decode(curl_exec($curl), true);
19.
20.    // Menutup CURL
21.    curl_close($curl);
22.
23.    // Mendapatkan array dari respons
24.    $result = $response['teams'];
25. ?>
26.
27.
28. <!DOCTYPE html>
29. <html lang="en">
30. <head>
31.     <meta charset="UTF-8">
32.     <meta name="viewport" content="width=device-width, initial-
33.         scale=1.0">
34.         <!-- Google Icon Font -->
35.         <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
36.             rel="stylesheet">
37.             <!-- My CSS -->
38.             <link rel="stylesheet" href="css/style.css">
39.             <!-- Materialize CSS -->
40.             <link type="text/css" rel="stylesheet" href="css/materialize.min.css"
41.                 " media="screen,projection"/>
```

```

42.      <title>Daftar Klub - Premier League</title>
43.  </head>
44.  <body>
45.
46.      <!-- Navigasi -->
47.
48.      <nav class="indigo darken-4">
49.          <div class="nav-wrapper container">
50.              <a href="#" class="brand-logo">Premier League</a>
51.              <a href="#" data-target="#mobile-demo" class="sidenav-trigger"><i class="material-icons">menu</i></a>
52.
53.              <ul class="right hide-on-med-and-down">
54.                  <li><a href="index.php">Klasemen</a></li>
55.                  <li><a href="klub.php">Daftar Klub</a></li>
56.                  <li><a href="pertandingan.php">Pertandingan</a></li>
57.              </ul>
58.          </div>
59.      </nav>
60.
61.      <ul class="sidenav" id="mobile-demo">
62.          <li><a href="index.php">Klasemen</a></li>
63.          <li><a href="klub.php">Daftar Klub</a></li>
64.          <li><a href="pertandingan.php">Pertandingan</a></li>
65.      </ul>
66.
67.      <div class="row">
68.          <h2 class="center-align">Daftar Klub</h2>
69.          <div class="col s12 m12">
70.              <?php if(!empty($result)) { ?>
71.                  <?php foreach($result as $res) : { ?>
72.                      <?php $image = $res['crestUrl']; ?>
73.                      <?php $image = preg_replace("/^http:/i", "https:", $image); ?>
74.
75.                      <div class="col s12 m4">
76.                          <div class="card">
77.                              <center><span class="card-title"><strong><?= $res['name']; ?></strong></span></center>
78.
79.                              <div class="card-image">
80.                                  
81.                              </div>
82.
83.                              <div class="card-content">
84.                                  <div class="content">
85.                                      <p class="center-align">Nama Pendek</p>
86.                                      <p class="center-align"><b><?= $res['shortName']; ?></b></p>
87.                                  </div>
88.
89.                              <div class="content">
90.                                  <p class="center-align">Berdiri</p>
91.                                  <p class="center-align"><b><?= $res['founded']; ?></b></p>
92.                              </div>
93.
94.                              <div class="content">

```

```

95.                                     <p class="center-
  align">Stadion</p>
96.                                     <p class="center-
  align"><b><?= $res['venue']; ?></b></p>
97.                                     </div>
98.
99.                                     <div class="content">
100.                                     <p class="center-
  align">Website</p>
101.                                     <p class="center-
  align"><a href=<?= $res['website']; ?>><?= $res['website']; ?></a></p>
102.                                     </div>
103.                                     </div>
104.
105.                                     <div class="card-action center-
  align">
106.                                         <a href="info-
  klub.php?id=<?= $res['id']; ?>" class="waves-effect waves-
  light btn indigo darken-2">Info Klub</a>
107.                                         </div>
108.                                     </div>
109.                                     <?php } endforeach; ?>
110.                                     <?php } ?>
111.                                     </div>
112.                                     </div>
113.                                     </div>
114.
115.                                     <script src="js/materialize.min.js"></script>
116.                                     <script>
117.                                         // Sidenav
118.                                         document.addEventListener('DOMContentLoaded', function()
  {
119.                                             var elems = document.querySelectorAll('.sidenav');
120.                                             M.Sidenav.init(elems);
121.                                         });
122.                                     </script>
123.
124.                                     </body>
125.                                     </html>

```

Script 18. klub.php

Pada *script klub.php* ini cara kerjanya sama seperti pada *script index.php* yang membedakan hanyalah *url* untuk *request API* nya saja. Untuk *request API* data klub adalah dengan *url* <https://api.football-data.org/v2/competitions/2021/teams>

Seperti halnya sebelumnya kita juga menetapkan **HTTP header** dengan nama atau *key* “**X-Auth-Token**” dan dengan *value* token API milik kita. Untuk mendapatkan data team pada

Premier League caranya adalah dengan mengakses hasil *response* dengan index *teams*.

Untuk menampilkan data hasil *request* ke *server* ke *html*, karena data hasil *request* adalah berupa *array*, maka pada bagian *html* kita harus melakukan *foreach* untuk mengakses *array* tersebut. Sama seperti sebelumnya untuk dapat menampilkan gambar atau logo klub kita harus mengganti terlebih dahulu *url* dari *file* gambar dari *http* menjadi *https* dengan menggunakan *preg_replace* milik PHP.

Di dalam daftar klub juga kita bisa melihat lebih detail tentang suatu klub. Seperti misalnya daftar pemain pada klub tersebut, tempat latihan, *website*, *email* dan lainnya. Caranya adalah dengan mengklik tombol info klub, maka otomatis halaman *website* akan menampilkan info detail klub yang kita pilih.

1.2.2.3 Detail Info Klub

Buat file dengan nama **info-klub.php** pada folder projek kita. File ini fungsinya adalah menampilkan tentang informasi detail tentang suatu klub yang dipilih pada halaman daftar klub.

Script dari *info-klub.php* sendiri adalah sebagai berikut:

```
1. <?php
2.
3. // API URL
4. $url = 'https://api.football-data.org/v2/teams';
5.
6. // TOKEN API
7. $token = '89403b3a0d1c462c91d61cd6882f7d6d';
8.
9. // Inisialisasi CURL dengan parameter
10. $curl = curl_init($url . '/' . $_GET['id']);
11.
12. // Set header request dengan Auth Token
13. curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
14. curl_setopt($curl, CURLOPT_HTTPHEADER, [
```

```

15.         'X-Auth-Token: '.$token.''
16.     ]);
17.
18.     // Mendapatkan respons dari server berupa json
19.     $response = json_decode(curl_exec($curl), true);
20.
21.     // Close CURL
22.     curl_close($curl);
23.
24.     // Mendapatkan array dari hasil respons
25.     $result = $response;
26.     $emain = $response['squad'];
27. ?>
28.
29. <!DOCTYPE html>
30. <html lang="en">
31. <head>
32.     <meta charset="UTF-8">
33.     <meta name="viewport" content="width=device-width, initial-
scale=1.0">
34.
35.     <!-- Google Icon Font -->
36.     <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
37.
38.     <!-- My CSS -->
39.     <link rel="stylesheet" href="css/style.css">
40.
41.     <!-- Materialize CSS -->
42.     <link type="text/css" rel="stylesheet" href="css/materialize.min.css"
" media="screen,projection"/>
43.
44.     <title>Info Klub</title>
45. </head>
46. <body>
47.
48.     <!-- Navigasi -->
49.
50.     <nav class="indigo darken-4">
51.         <div class="nav-wrapper container">
52.             <a href="#" class="brand-logo">Premier League</a>
53.             <a href="#" data-target="mobile-demo" class="sidenav-
trigger"><i class="material-icons">menu</i></a>
54.
55.             <ul class="right hide-on-med-and-down">
56.                 <li><a href="index.php">Klasemen</a></li>
57.                 <li><a href="klub.php">Daftar Klub</a></li>
58.                 <li><a href="pertandingan.php">Pertandingan</a></li>
59.             </ul>
60.         </div>
61.     </nav>
62.
63.     <ul class="sidenav" id="mobile-demo">
64.         <li><a href="index.php">Klasemen</a></li>
65.         <li><a href="klub.php">Daftar Klub</a></li>
66.         <li><a href="pertandingan.php">Pertandingan</a></li>
67.     </ul>
68.
69.     <div class="container">
70.         <div class="row">
71.             <h2 class="center-align"> Daftar Klub </h2>

```

```

72.          <div class="container col s12 m12 center-align">
73.              <div class="card">
74.                  <?php $image = $result['crestUrl']; ?>
75.                  <?php $image = preg_replace("/^http:/i", "https:
    ", $image); ?>
76.
77.
78.                  <div class="card-image">
79.                      
80.                  </div>
81.
82.                  <div class="card-content">
83.                      <span class="card-
    title"><strong> <?= $result['name']; ?> </strong></span>
84.
85.                      <table>
86.                          <tr>
87.                              <th> Nama Pendek </th>
88.                              <td> <?= $result['shortName']; ?> </
    td>
89.                          </tr>
90.                          <tr>
91.                              <th> Singkatan </th>
92.                              <td> <?= $result['tla']; ?></td>
93.                          </tr>
94.                          <tr>
95.                              <th> Berdiri </th>
96.                              <td> <?= $result['founded']; ?> </td
    >
97.                          </tr>
98.                          <tr>
99.                              <th> Stadion </th>
100.                             <td> <?= $result['venue']; ?>
    </td>
101.                         </tr>
102.                         <tr>
103.                             <th> Jersey </th>
104.                             <td> <?= $result['clubColors'
    ]; ?> </td>
105.                         </tr>
106.                         <tr>
107.                             <th> Alamat </th>
108.                             <td> <?= $result['address'];
    ?> </td>
109.                         </tr>
110.                         <tr>
111.                             <th> No. Tlp </th>
112.                             <td> <?= $result['phone'];
    </td>
113.                         </tr>
114.                         <tr>
115.                             <th> Website </th>
116.                             <td> <a href="<?= $result['we
    bsite']; ?>" target="_blank"><?= $result['website'];
    ?></a> </td>
117.                         </tr>
118.                         <tr>
119.                             <th> Email </th>
120.                             <td> <?= $result['email'];
    ?>
    </td>
121.                         </tr>

```

```

122.                                     </table>
123.                                 </div>
124.                                 </div>
125.                                 </div>
126.                               </div>
127.
128.                               <h2 class="center-align"> Daftar Pemain </h2>
129.
130.                               <div class="row">
131.                                 <div class="col s12 m12">
132.                                   <?php if(!empty(empty($pemain)) { ?>
133.                                     <?php foreach($pemain as $p) : { ?>
134.
135.                                       <div class="col s12 m6">
136.                                         <div class="card">
137.
138.                                           <div class="card-content">
139.                                             <center><span class="card-
    title"><strong><?= $p['name']; ?></strong></span></center>
140.
141.                                           <p class="center-
    align"><?= $p['position']; ?></p>
142.                                         <h1 class="center-
    align"><?= $p['shirtNumber']; ?></h1>
143.                                           <p class="center-
    align">Asal Negara: <b><?= $p['nationality']; ?></b></p>
144.                                           <p class="center-
    align">Tempat Lahir: <b><?= $p['countryOfBirth']; ?></b> </p>
145.                                           <p class="center-
    align">Tanggal Lahir: <b><?= $p['dateOfBirth']; ?></b></p>
146.                                         </div>
147.                                       </div>
148.                                     </div>
149.                                   <?php } endforeach; ?>
150.                                 <?php } ?>
151.                               </div>
152.                             </div>
153.                           <center><h5>Last Update: <?= $result['lastUpdated']; ?></
    h5></center>
154.                         </div>
155.
156.                         <script src="js/materialize.min.js"></script>
157.                         <script>
158.                           // Sidenav
159.                           document.addEventListener('DOMContentLoaded', function()
  {
160.                             var elems = document.querySelectorAll('.sidenav');
161.                             M.Sidenav.init(elems);
162.                           });
163.                         </script>
164.
165.                       </body>
166.                     </html>

```

Script 19. info-klub.php

Cara *request* data pada **info-klub.php** juga sama seperti sebelumnya, yang membedakan hanyalah dari *url*-nya saja, kita tambahkan parameter id yang didapat ketika kita mengklik info klub pada halaman daftar klub.

Untuk menampilkan daftar pemain kita hanya perlu mengakses dari hasil *response* dengan index *squad*. Dan menampilkannya dengan *html* dan dilakukan dengan *foreach* untuk mengakses data *array*-nya.

1.2.2.4 Daftar Pertandingan Premier League

Buat *file* dengan nama **pertandingan.php** di dalam folder projek kita. File **pertandingan.php** ini nantinya akan menampilkan hasil *request* API tentang histori atau daftar pertandingan yang dilakukan pada *Premier League*.

```
1. <?php
2.     // API URL
3.     $url = 'https://api.football-
4.         data.org/v2/competitions/2021/matches';
5.     // API Token
6.     $token = '89403b3a0d1c462c91d61cd6882f7d6d';
7.
8.     // Inisialisasi CURL
9.     $curl = curl_init($url);
10.
11.    // Set Header Auth Token
12.    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
13.    curl_setopt($curl, CURLOPT_HTTPHEADER, [
14.        'X-Auth-Token: '.$token.''
15.    ]);
16.
17.    // Mendapatkan respons
18.    $response = json_decode(curl_exec($curl), true);
19.
20.    // Menutup CURL
21.    curl_close($curl);
22.
23.    // Mendapatkan array dari respons
24.    $lastUpdate = $response['competition'];
```

```

25.     $result = $response['matches'];
26.
27. ?>
28.
29.
30.
31. <!DOCTYPE html>
32. <html lang="en">
33. <head>
34.     <meta charset="UTF-8">
35.     <meta name="viewport" content="width=device-width, initial-
36.         scale=1.0">
37.         <!-- Google Icon Font -->
38.         <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
39.             rel="stylesheet">
40.         <!-- My CSS -->
41.         <link rel="stylesheet" href="css/style.css">
42.
43.         <!-- Materialize CSS -->
44.         <link type="text/css" rel="stylesheet" href="css/materialize.min.css"
45.             " media="screen,projection"/>
46.     <title>Pertandingan - Premier League</title>
47. </head>
48. <body>
49.
50.     <!-- Navigasi -->
51.
52.     <nav class="indigo darken-4">
53.         <div class="nav-wrapper container">
54.             <a href="#" class="brand-logo">Premier League</a>
55.             <a href="#" data-target="mobile-demo" class="sidenav-
56.                 trigger"><i class="material-icons">menu</i></a>
57.             <ul class="right hide-on-med-and-down">
58.                 <li><a href="index.php">Klasemen</a></li>
59.                 <li><a href="klub.php">Daftar Klub</a></li>
60.                 <li><a href="pertandingan.php">Pertandingan</a></li>
61.             </ul>
62.         </div>
63.     </nav>
64.
65.     <ul class="sidenav" id="mobile-demo">
66.         <li><a href="index.php">Klasemen</a></li>
67.         <li><a href="klub.php">Daftar Klub</a></li>
68.         <li><a href="pertandingan.php">Pertandingan</a></li>
69.     </ul>
70.
71.     <div class="container">
72.         <div class="row">
73.             <h2 class="center-align"> Pertandingan </h2>
74.             <p class="center-
75.                 align"><strong>Terakhir Update: <?= $lastUpdate['lastUpdated']; ?></stro-
76.                 ng></p>
77.                 <div class="col s12 m12">
78.                     <?php if(!empty($result)) { ?>
79.                         <?php foreach($result as $res) : ?>

```

```

80.                     <div class="card-stacked">
81.                         <div class="card-content">
82.                             <div class="col s12 m5">
83.                                 <h2 class="center-
84.                                     align"> <?= $res['score']['fullTime']['homeTeam']; ?></h2>
85.                                         <h4 class="center-
86.                                             align"> <?= $res['homeTeam']['name']; ?></h4>
87.                                         </div>
88.                                         <div class="col s12 m2">
89.                                             <h3 class="center-
90.                                                 align"> VS </h3>
91.                                             </div>
92.                                         <div class="col s12 m5">
93.                                             <h2 class="center-
94.                                                 align"> <?= $res['score']['fullTime']['awayTeam']; ?></h2>
95.                                                 <h4 class="center-
96.                                                     align"><strong> <?= $res['utcDate']; ?> </strong> </h4>
97.                                                     <p class="center-
98.                                                         align"><strong> <?= $res['status']; ?> </strong></p>
99.                                                     </div>
100.                                         <?php endforeach; ?>
101.                                         </div>
102.                                         </div>
103.
104.                                         <script src="js/materialize.min.js"></script>
105.                                         <script>
106.                                             // Sidenav
107.                                             document.addEventListener('DOMContentLoaded', function()
108.     {
109.         var elems = document.querySelectorAll('.sidenav');
110.         M.Sidenav.init(elems);
111.     });
112.                                         </script>
113.                                         </body>
114.                                         </html>

```

Script 20. pertandingan.php

Sama juga seperti sebelum-sebelumnya, cara kerja dari script pertandingan.php diatas juga sama seperti cara kerja script-script sebelumnya, yang membedakan hanyalah url untuk request API nya saja. URL untuk melakukan request data tentang pertandingan adalah dengan link <https://api.football-data.org/v2/competitions/2021/matches>

Cara menampilkan data hasil *request* (*response*) nya pun sama seperti sebelumnya, dimana kita melakukan *foreach* pada bagian html untuk mengakses data *array*-nya.

2. Pembahasan Aplikasi

Aplikasi atau Website yang dibuat memiliki dua fungsi yaitu:

1. Website sebagai Web Service

Website yang dibuat dapat menjadi web service, dimana web service yang diberikan adalah tentang data buku. Web service yang dibuat mempunyai beberapa fitur, seperti listAll atau menampilkan seluruh data buku, listId atau menampilkan data buku berdasarkan id buku, search atau mencari data buku, tambah atau menambah data buku, edit atau mengubah data buku, dan delete atau menghapus data buku.

- Untuk fitur listAll atau menampilkan seluruh data buku bisa diakses atau dilihat melalui link:

<https://rifqipratama9.000webhostapp.com/api/buku/listAll.php>

- Untuk fitur listId atau menampilkan data buku bisa diakses atau dilihat melalui link dibawah dengan mengisi id buku yang ingin diakses:

<https://rifqipratama9.000webhostapp.com/api/buku/listId.php?id={id}>

- Untuk fitur tambah buku, bisa diakses atau dilihat melalui link di bawah ini, dengan mengirim berupa body melalui postman

atau aplikasi yang kita punya. Body ini berisi tentang id buku, judul, penulis, penerbit, deskripsi, bahasa, genre, jumlah halaman dan tahun terbit.

<https://rifqipratama9.000webhostapp.com/api/buku/tambah.php>

- Untuk fitur update atau ubah data pun sama, kita harus mengirim body yang berisi id buku, judul, penulis, penerbit, deskripsi, bahasa, genre, jumlah halaman dan tahun terbit. Link untuk melakukan request atau akses adalah

<https://rifqipratama9.000webhostapp.com/api/buku/update.php>

- Untuk fitur delete atau hapus data, kita harus mengirim body yang berisi hanya id buku yang akan kita hapus saja. Untuk melakukan request atau akses bisa melalui link

<https://rifqipratama9.000webhostapp.com/api/buku/delete.php>

- Untuk fitur search atau hapus data, bisa diakses atau melakukan request melalui link di bawah ini, dengan mengisi parameter s. S adalah query untuk data yang akan kita cari.

<https://rifqipratama9.000webhostapp.com/api/buku/search.php?>
s={s}

2. Website menampilkan atau mengkonsumsi API dari pihak lain (football-data.org)

Pada website ini juga dapat menampilkan API dari pihak ketiga atau pihak lain. API yang dipakai pada website ini adalah API dari football-data.org. Website ini dapat menampilkan klasemen dari Premier League, daftar klub yang ada di Premier League, Info detail tentang suatu klub, dan pertandingan di Premier League.

- Untuk mengakses link untuk menampilkan klasemen bisa dilalui link:

<https://rifqipratama9.000webhostapp.com/>

- Untuk mengakses link untuk menampilkan data klub bisa dilihat atau melalui link:

<https://rifqipratama9.000webhostapp.com/klub.php>

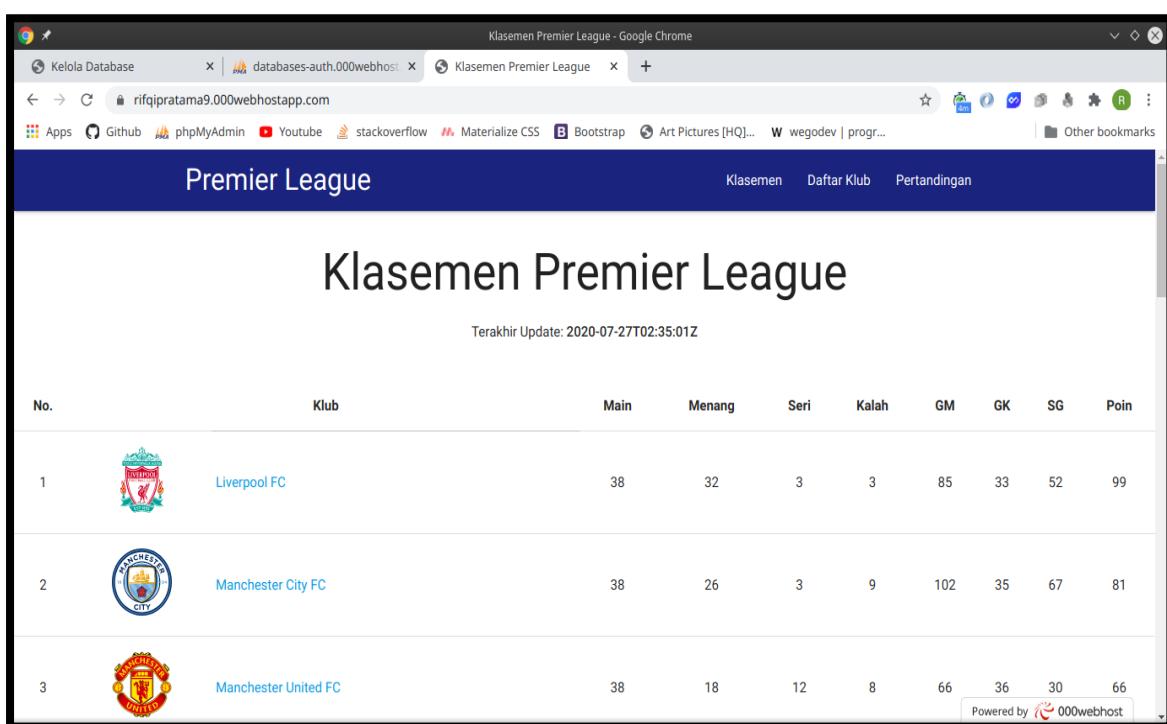
- Untuk menampilkan info detail tentang suatu klub bisa dilihat melalui halaman daftar klub lalu klik pada bagian info klub, maka otomatis website akan mengakses info detail tentang klub yang dipilih.
- Untuk menampilkan info pertandingan, bisa melalui atau dilihat pada link:

<https://rifqipratama9.000webhostapp.com/pertandingan.php>

3. Pembahasan Fitur Application Programming Interface

Fitur-fitur yang dari API yang dipakai adalah dapat menampilkan klasemen Premier League, daftar klub pada Premier League, info detail klub, daftar pemain pada klub tertentu dan pertandingan di Premier League.

Untuk bagian halaman utama atau klasemen, seperti ini tampilannya

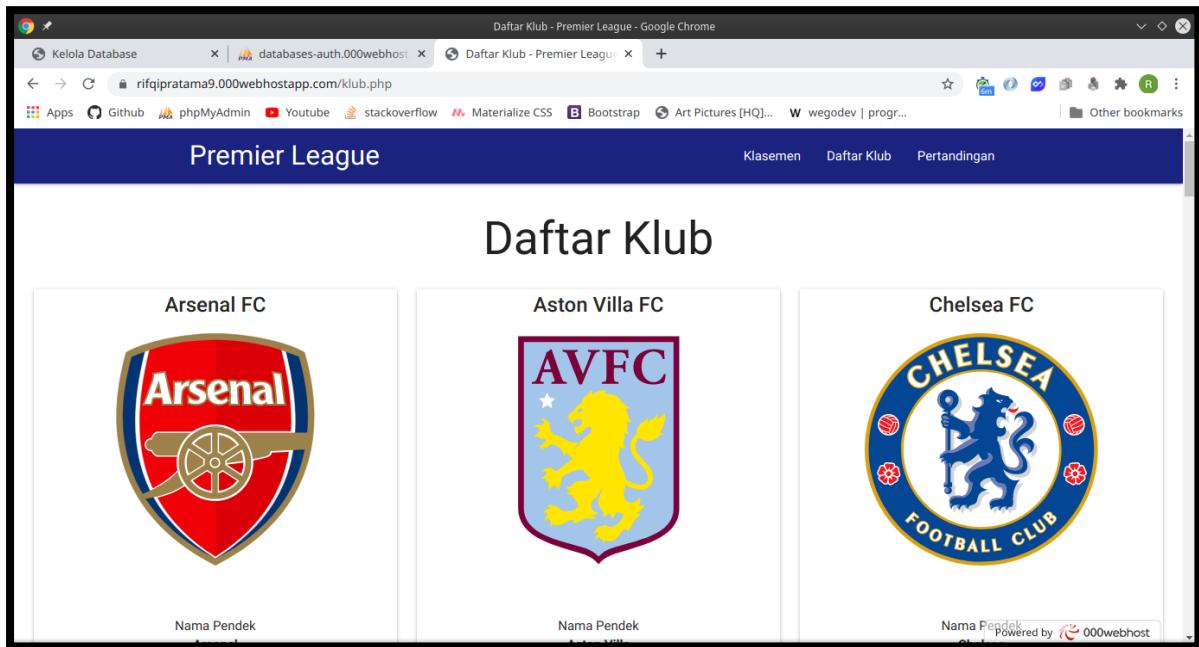


The screenshot shows a Google Chrome browser window with the title 'Klasemen Premier League - Google Chrome'. The address bar shows 'databases-auth.000webhost.com'. The page itself has a dark blue header with 'Premier League' on the left and navigation links 'Klasemen', 'Daftar Klub', and 'Pertandingan' on the right. Below the header is a large heading 'Klasemen Premier League' and a timestamp 'Terakhir Update: 2020-07-27T02:35:01Z'. A table follows, showing the top three teams in the league:

No.	Klub	Main	Menang	Seri	Kalah	GM	GK	SG	Poin
1	Liverpool FC	38	32	3	3	85	33	52	99
2	Manchester City FC	38	26	3	9	102	35	67	81
3	Manchester United FC	38	18	12	8	66	36	30	66

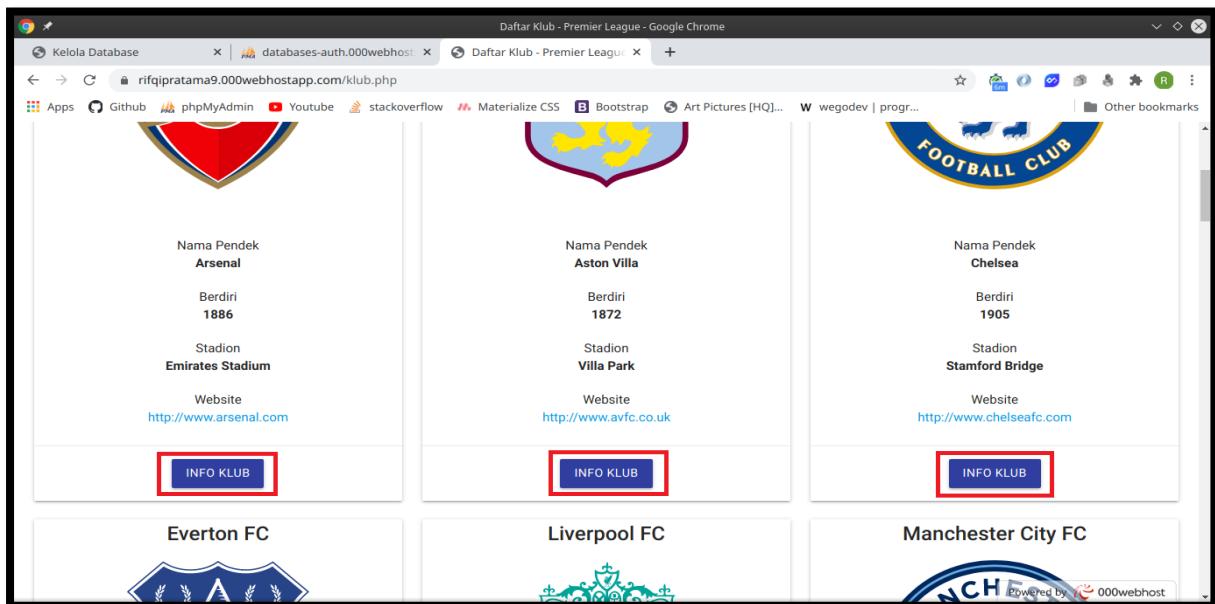
Gambar 7. Halaman Utama atau Klasemen

Untuk mengakses bagian daftar klub kita bisa klik pada bagian navigasi tentang daftar klub. Tampilan untuk daftar klub sendiri adalah seperti ini:



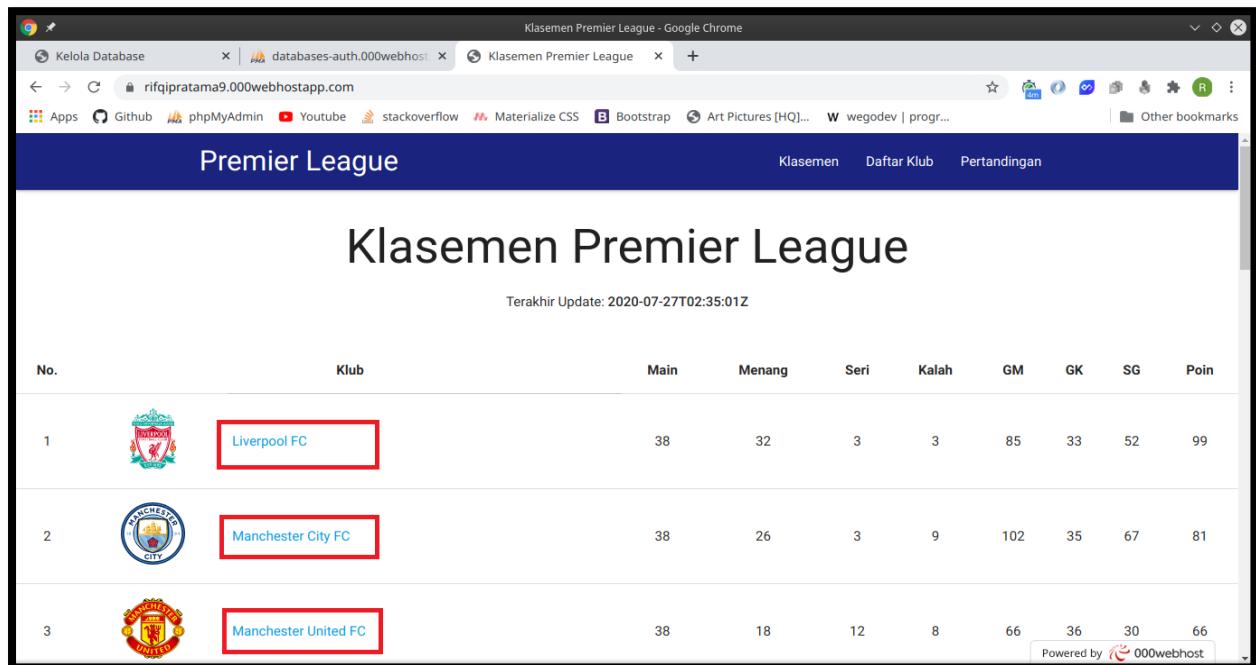
Gambar 8. Daftar Klub

Untuk mengakses informasi tentang detail klub kita bisa klik pada bagian info klub pada bagian bawah daftar klub



Gambar 9. Klik Info Klub

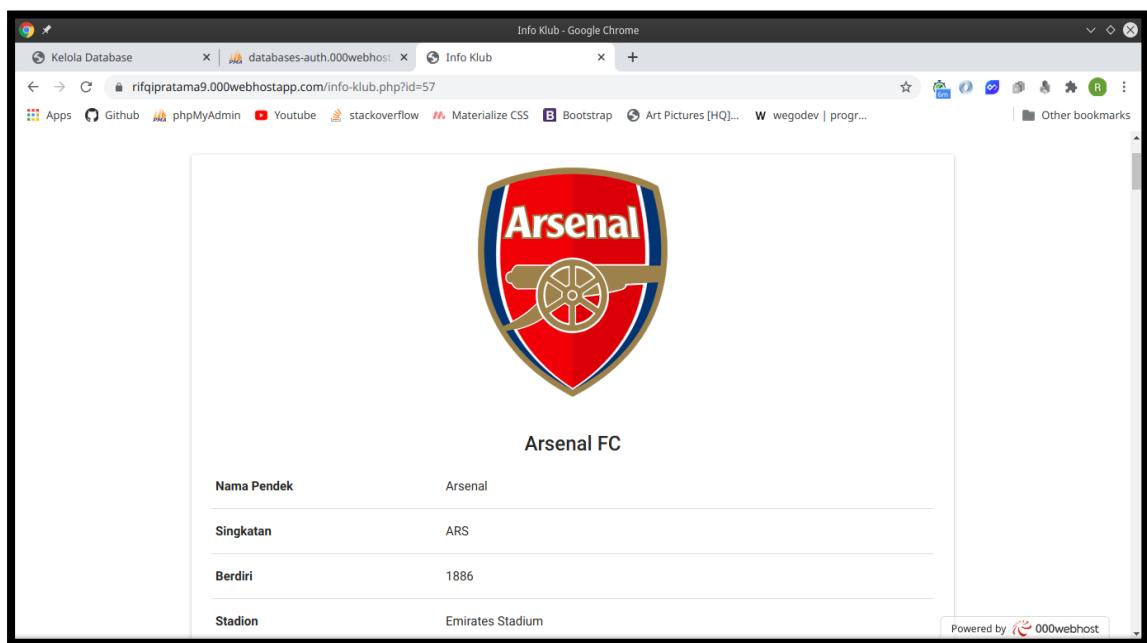
Atau kita juga bisa dengan mengklik pada bagian nama klub di bagian halaman utama atau klasemen.



No.	Klub	Main	Menang	Seri	Kalah	GM	GK	SG	Poin
1	Liverpool FC	38	32	3	3	85	33	52	99
2	Manchester City FC	38	26	3	9	102	35	67	81
3	Manchester United FC	38	18	12	8	66	36	30	66

Gambar 10. Klik Info Klub 2

Dan untuk tampilan bagian informasi detail klub adalah seperti berikut ini:



Gambar 11. Info Klub

Dan di dalam informasi tentang detail klub, kita juga bisa melihat daftar pemain pada klub tersebut.

The screenshot shows a Google Chrome window titled "Info Klub - Google Chrome". The URL is rifqipratama9.000webhostapp.com/info-klub.php?id=57. The page title is "Daftar Pemain". It displays four player cards:

- Emiliano Martínez** (Goalkeeper)
Asal Negara: Argentina
Tempat Lahir: Argentina
Tanggal Lahir: 1992-09-02T00:00:00Z
- Bernd Leno** (Goalkeeper)
Asal Negara: Germany
Tempat Lahir: Germany
Tanggal Lahir: 1992-03-04T00:00:00Z
- Matt Macey** (Goalkeeper)
Number: 33
- Sokratis Papastathopoulos** (Defender)
Number: 5

Powered by 000webhost

Gambar 12. Daftar Pemain

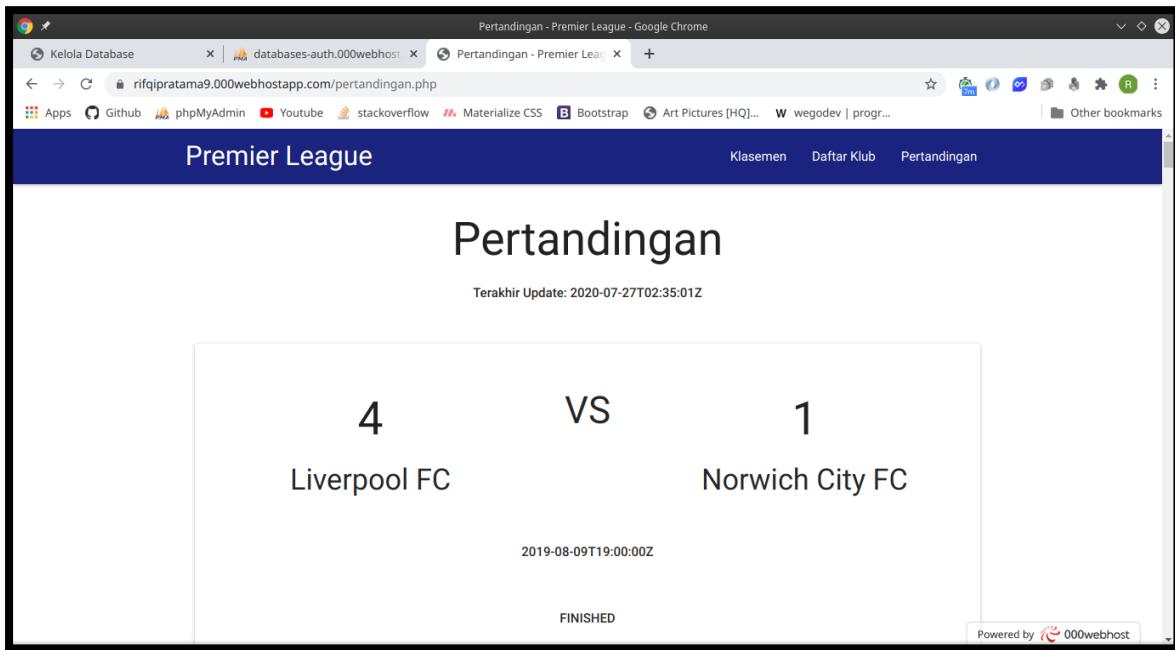
The screenshot shows a Google Chrome window titled "Info Klub - Google Chrome". The URL is rifqipratama9.000webhostapp.com/info-klub.php?id=57. The page title is "Daftar Pemain". It displays four player cards:

- Matt Macey** (Goalkeeper)
Number: 33
Asal Negara: England
Tempat Lahir: England
Tanggal Lahir: 1994-09-09T00:00:00Z
- Sokratis Papastathopoulos** (Defender)
Number: 5
Asal Negara: Greece
Tempat Lahir: Greece
Tanggal Lahir: 1988-06-09T00:00:00Z
- Cédric** (Defender)
Number: 2
Asal Negara: Portugal
Tempat Lahir: Germany
- Pablo Marí** (Defender)
Number: 22
Asal Negara: Spain
Tempat Lahir: Spain

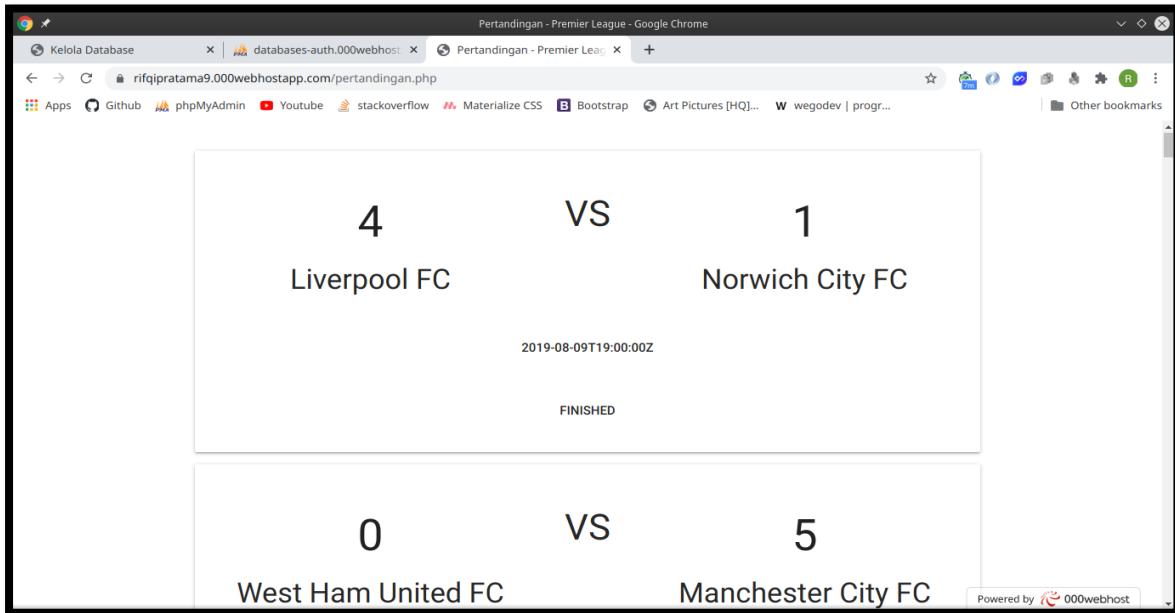
Powered by 000webhost

Gambar 13. Daftar Pemain 2

Dan yang terakhir adalah tentang daftar pertandingan pada Premier League, untuk tampilan daftar pertandingan adalah seperti ini:



Gambar 14. Pertandingan



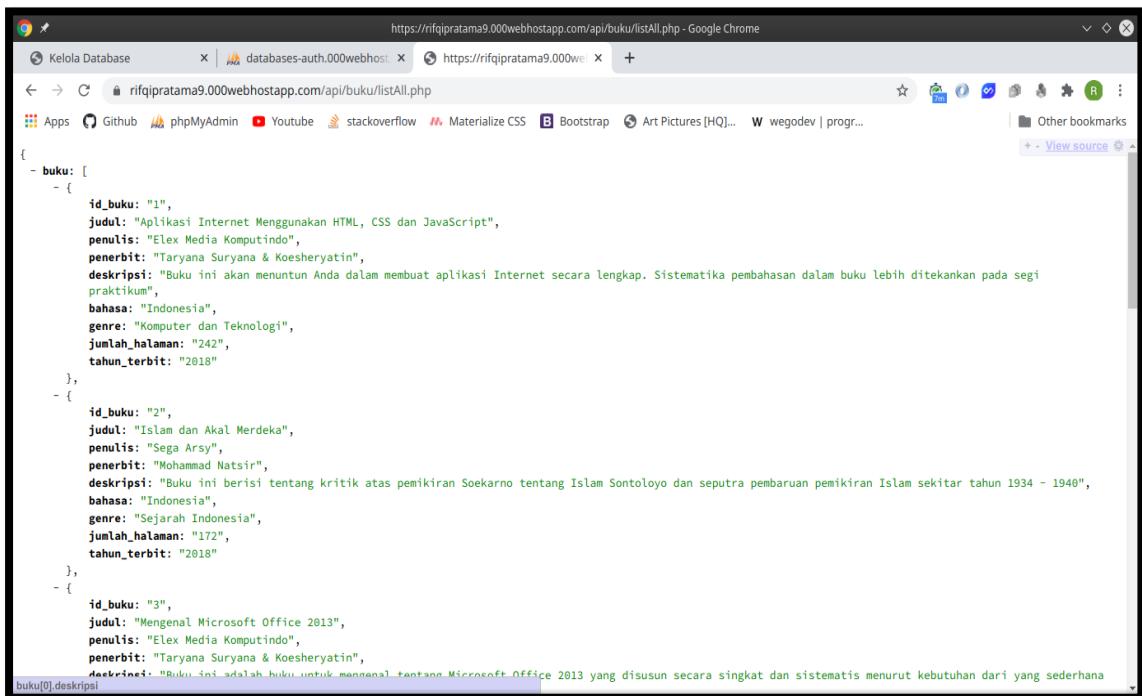
Gambar 15. Pertandingan 2

4. Pembahasan Fitur Web Service

Terdapat beberapa fitur pada web service ini, diantaranya adalah listAll yang dapat menampilkan seluruh data, listId yang dapat menampilkan data berdasarkan id buku, delete untuk menghapus data buku, update untuk mengubah atau mengedit data buku, tambah untuk menambah data buku, dan search untuk mencari data buku berdasarkan judul buku.

4.1 ListAll

Pada fitur listAll ini kita hanya perlu melakukan request pada link <https://rifqipratama9.000webhostapp.com/api/buku/listAll.php> dan dengan menggunakan method GET, maka semua data dengan format json akan muncul.

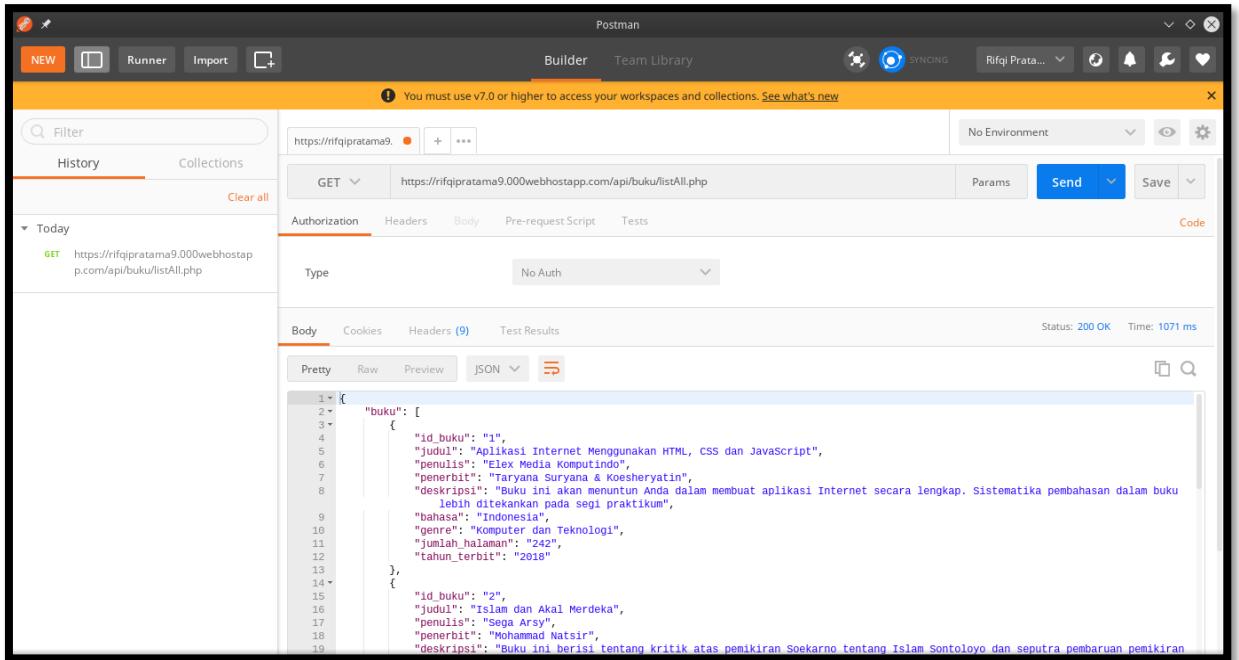


The screenshot shows a Google Chrome browser window with the URL <https://rifqipratama9.000webhostapp.com/api/buku/listAll.php> in the address bar. The page content displays a JSON array of book data:

```
{
  "buku": [
    {
      "id_buku": "1",
      "judul": "Aplikasi Internet Menggunakan HTML, CSS dan JavaScript",
      "penulis": "Elex Media Komputindo",
      "penerbit": "Taryana Suryana & Koesheryatin",
      "deskripsi": "Buku ini akan menunjang Anda dalam membuat aplikasi Internet secara lengkap. Sistematika pembahasan dalam buku lebih ditekankan pada segi praktikum",
      "bahasa": "Indonesia",
      "genre": "Komputer dan Teknologi",
      "jumlah_halaman": "242",
      "tahun_terbit": "2018"
    },
    {
      "id_buku": "2",
      "judul": "Islam dan Akal Merdeka",
      "penulis": "Sega Arsy",
      "penerbit": "Mohammad Natsir",
      "deskripsi": "Buku ini berisi tentang kritik atas pemikiran Soekarno tentang Islam Sontoloyo dan seputar pembaruan pemikiran Islam sekitar tahun 1934 - 1940",
      "bahasa": "Indonesia",
      "genre": "Sejarah Indonesia",
      "jumlah_halaman": "172",
      "tahun_terbit": "2018"
    },
    {
      "id_buku": "3",
      "judul": "Mengenal Microsoft Office 2013",
      "penulis": "Elex Media Komputindo",
      "penerbit": "Taryana Suryana & Koesheryatin",
      "deskripsi": "Buku ini adalah buku untuk mempelajari Microsoft Office 2013 yang disusun secara singkat dan sistematis menurut kebutuhan dari yang sederhana"
    }
  ]
}
```

Gambar 16. listAll

Dan jika kita menggunakan postman akan seperti ini:



The screenshot shows the Postman application interface. In the top bar, there are buttons for NEW, Runner, Import, and a workspace dropdown. The main area has tabs for History, Collections, and a search bar. A message at the top says "You must use v7.0 or higher to access your workspaces and collections. See what's new". Below this, a GET request is selected with the URL "https://rifqipratama9.000webhostapp.com/api/buku/listAll.php". The "Authorization" tab is active, showing "No Auth". The "Body" tab is selected, displaying a JSON response with two book entries. The JSON output is as follows:

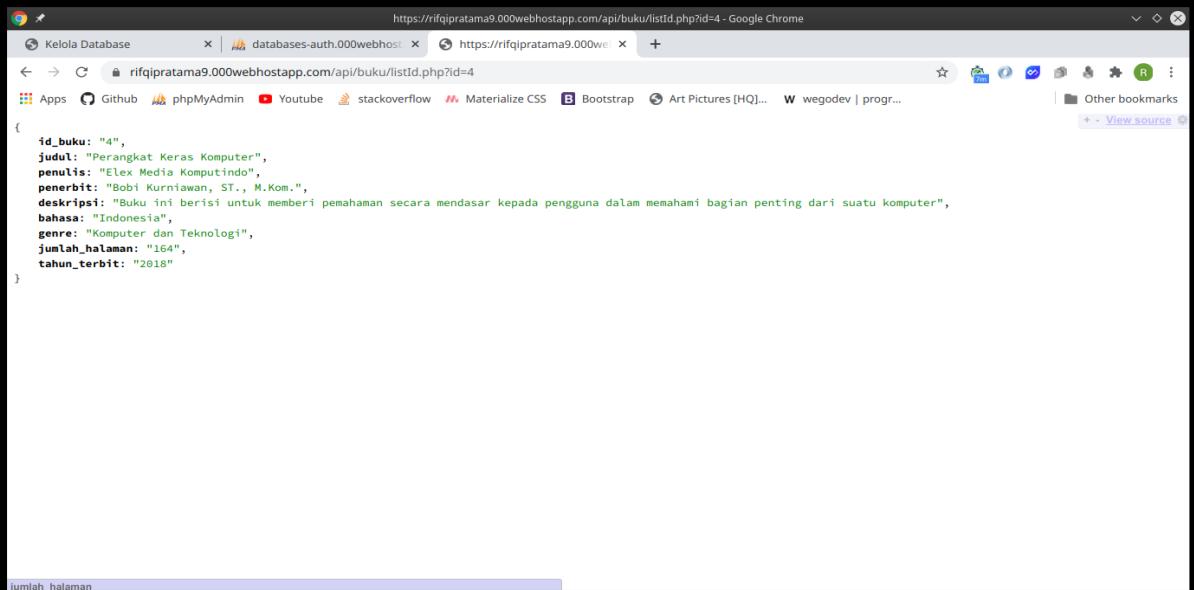
```
1  [
2   {
3     "buku": [
4       {
5         "id_buku": "1",
6         "judul": "Aplikasi Internet Menggunakan HTML, CSS dan Javascript",
7         "penulis": "Ellex Media Komputindo",
8         "penerbit": "Taryana Suryana & Koesheryatin",
9         "deskripsi": "Buku ini akan menuntun Anda dalam membuat aplikasi Internet secara lengkap. Sistematika pembahasan dalam buku lebih ditekankan pada segi praktikum",
10        "bahasa": "Indonesia",
11        "genre": "Komputer dan Teknologi",
12        "jumlah_halaman": "242",
13        "tahun_terbit": "2018"
14      },
15      {
16        "id_buku": "2",
17        "judul": "Islam dan Akal Merdeka",
18        "penulis": "Sega Arsy",
19        "penerbit": "Mohammad Natsir",
20        "deskripsi": "Buku ini berisi tentang kritik atas pemikiran Soekarno tentang Islam Sontoloyo dan seputra pembaruan pemikiran"
21      }
22    ]
23  ]
```

The status bar at the bottom right indicates "Status: 200 OK" and "Time: 1071 ms".

Gambar 17. listAll Postman

4.2 listId

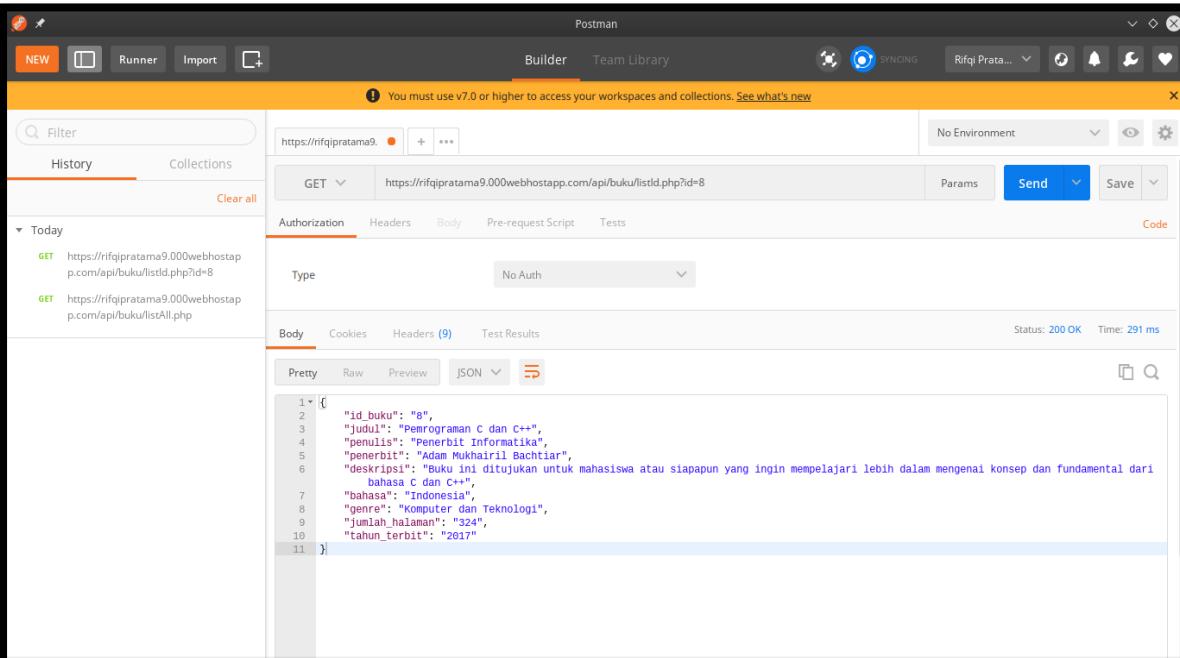
Pada fitur listId ini kita hanya perlu melakukan request pada link <https://rifqipratama9.000webhostapp.com/api/buku/listId.php?id={id}> dan dengan menggunakan method GET dan disertai dengan parameter berupa id buku yang ingin ditampilkan.



A screenshot of a Google Chrome browser window. The address bar shows the URL: <https://rifqipratama9.000webhostapp.com/api/buku/listId.php?id=4>. The page content displays a JSON object representing a book record:

```
{  
    "id_buku": "4",  
    "judul": "Perangkat Keras Komputer",  
    "penulis": "Elex Media Komputindo",  
    "penerbit": "Bobi Kurniawan, ST., M.Kom.",  
    "deskripsi": "Buku ini berisi untuk memberi pemahaman secara mendasar kepada pengguna dalam memahami bagian penting dari suatu komputer",  
    "bahasa": "Indonesia",  
    "genre": "Komputer dan Teknologi",  
    "jumlah_halaman": "164",  
    "tahun_terbit": "2018"  
}
```

Gambar 18. listId

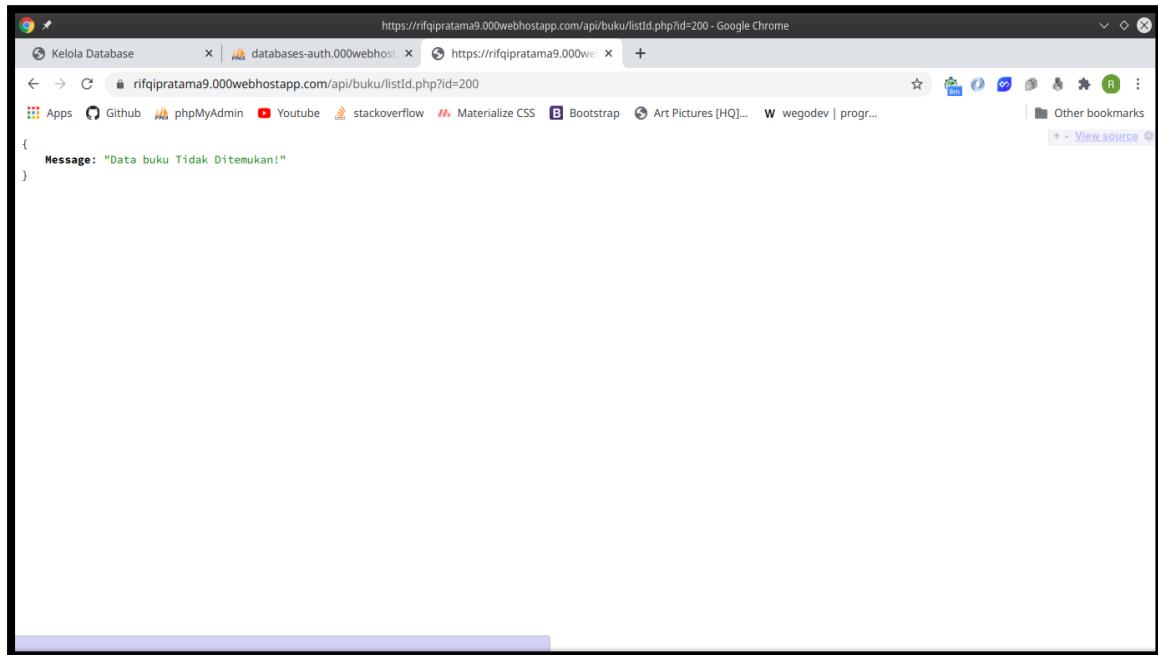


A screenshot of the Postman application interface. The top navigation bar includes 'NEW', 'Runner', 'Import', and 'Builder'. The main area shows a 'History' section with two recent requests: 'GET https://rifqipratama9.000webhostapp.com/api/buku/listId.php?id=8' and 'GET https://rifqipratama9.000webhostapp.com/api/buku/listAll.php'. A 'Builder' tab is active, showing a GET request to <https://rifqipratama9.000webhostapp.com/api/buku/listId.php?id=8>. The 'Body' tab displays the JSON response:

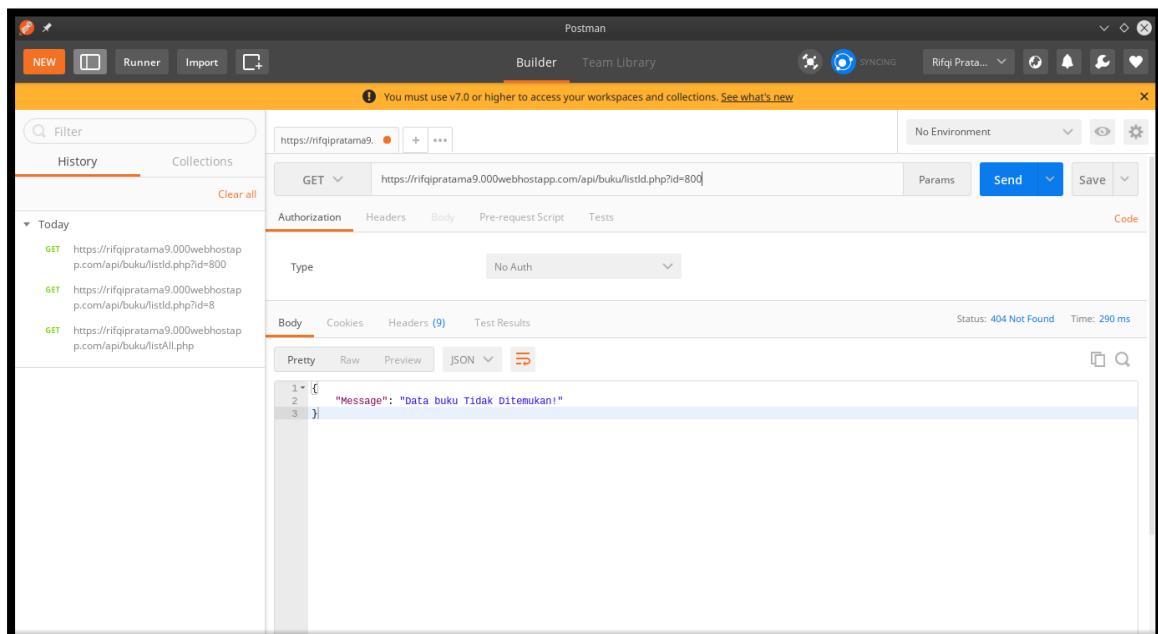
```
1  {  
2      "id_buku": "8",  
3      "judul": "Pemrograman C dan C++",  
4      "penulis": "Penerbit Informatika",  
5      "penerbit": "Adam Mukhairil Bachtiar",  
6      "deskripsi": "Buku ini ditujukan untuk mahasiswa atau siapapun yang ingin mempelajari lebih dalam mengenai konsep dan fundamental dari bahasa C dan C++.",  
7      "bahasa": "Indonesia",  
8      "genre": "Komputer dan Teknologi",  
9      "jumlah_halaman": "324",  
10     "tahun_terbit": "2017"  
11 }
```

Gambar 19. listId Postman

Lalu jika id buku pada parameter tidak ada atau tidak ditemukan, maka akan memunculkan pesan.



Gambar 20. listId Tidak Ada

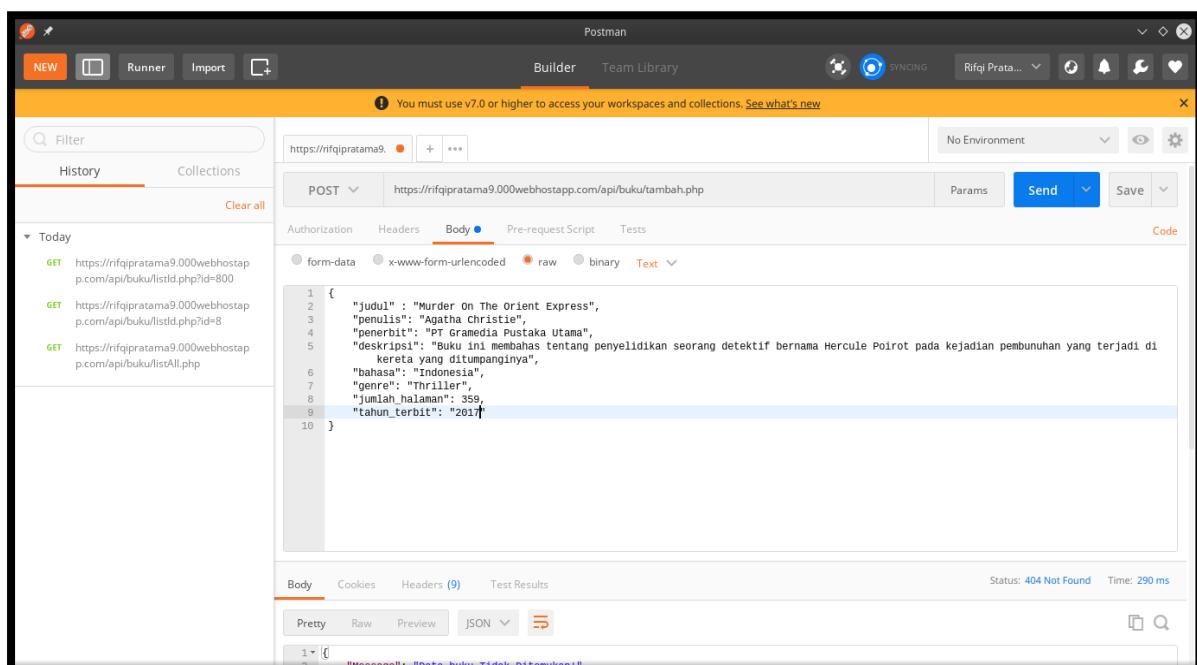


Gambar 21. listId Tidak Ada Postman

4.3 Tambah Data

Untuk melakukan tambah data, kita harus mengirimkan body dengan value judul, penulis, penerbit, deskripsi, bahasa, genre, jumlah halaman, dan tahun terbit. Untuk menambah data sendiri kita gunakan method POST. Untuk link tambah data adalah

<https://rifqipratama9.000webhostapp.com/api/buku/tambah.php>



Gambar 22. Tambah Postman

Jika proses tambah data berhasil, maka akan mengembalikan pesan berhasil

The screenshot shows the Postman application interface. In the center, there is a POST request to the URL `https://rifqipratama9.000webhostapp.com/api/buku/tambah.php`. The request body contains the following JSON data:

```
4 "penerbit": "PT Gramedia Pustaka Utama",
5 "deskripsi": "Buku ini membahas tentang penyelidikan seorang detektif bernama Hercule Poirot pada kejadian pembunuhan yang terjadi di
6 kereta yang ditumpanginya",
7 "bahasa": "Indonesia",
8 "genre": "Thriller",
9 "jumlah_halaman": 359,
10 "tahun_terbit": "2017"
```

Below the request, the response status is `201 Created` and the time taken is `786 ms`. The response body shows a JSON message: `{"Message": "Data Buku Berhasil Ditambahkan"}`.

Gambar 23. Tambah Data Berhasil

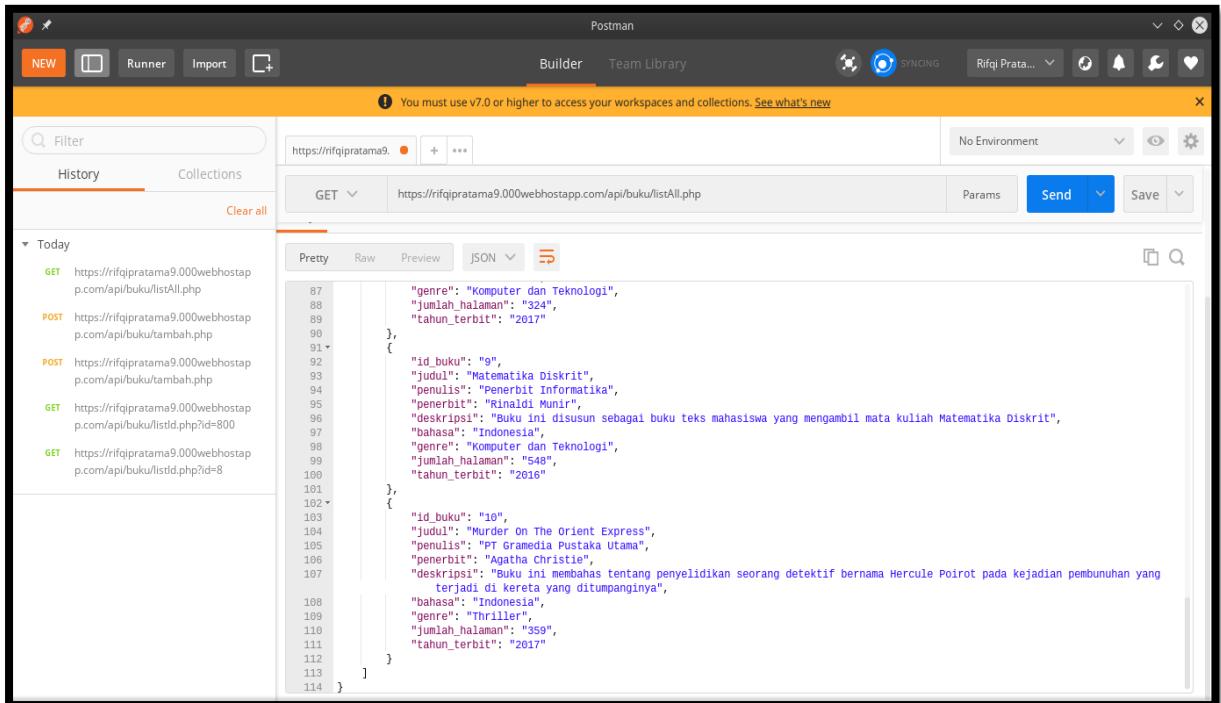
Jika pada saat akan melakukan tambah data terdapat data yang belum diisi atau kosong, maka tambah data akan gagal dan akan mengembalikan pesan

The screenshot shows the Postman application interface. In the center, there is a POST request to the URL `https://rifqipratama9.000webhostapp.com/api/buku/tambah.php`. The request body is empty, indicated by the number `10` at the top of the body area.

Below the request, the response status is `404 Not Found` and the time taken is `289 ms`. The response body shows a JSON message: `{"Message": "Gagal! SEMUA DATA HARUS DIISI!"}`.

Gambar 24. Tambah Data Harus Diisi

List Data setelah penambahan data berhasil.



The screenshot shows the Postman application interface. In the top navigation bar, 'Builder' is selected. The main area displays a GET request to 'https://rifqipratama9.000webhostapp.com/api/buku/listAll.php'. The response body is shown in 'Pretty' format, displaying a JSON array of book records. The JSON data includes fields such as genre, jumlah_halaman, tahun_terbit, id_buku, judul, penulis, penerbit, bahasa, and deskripsi. The last record added is a mystery novel by Agatha Christie titled 'Murder On The Orient Express'.

```
[{"id": 87, "genre": "Komputer dan Teknologi", "jumlah_halaman": "324", "tahun_terbit": "2017"}, {"id": 88, "id_buku": "9", "judul": "Matematika Diskrit", "penulis": "Penerbit Informatika", "penerbit": "Rinaldi Munir", "deskripsi": "Buku ini disusun sebagai buku teks mahasiswa yang mengambil mata kuliah Matematika Diskrit", "bahasa": "Indonesia", "genre": "Komputer dan Teknologi", "jumlah_halaman": "548", "tahun_terbit": "2016"}, {"id": 89, "id_buku": "10", "judul": "Murder On The Orient Express", "penulis": "PT Gramedia Pustaka Utama", "penerbit": "Agatha Christie", "deskripsi": "Buku ini membahas tentang penyelidikan seorang detektif bernama Hercule Poirot pada kejadian pembunuhan yang terjadi di kereta yang ditumpanginya", "bahasa": "Indonesia", "genre": "Thriller", "jumlah_halaman": "359", "tahun_terbit": "2017"}]
```

Gambar 25. List Setelah Tambah Data

4.4 Update Data

Untuk melakukan update data, kita juga harus mengirimkan body dengan isi id buku, judul, penulis, penerbit, deskripsi, bahasa, genre, jumlah halaman, dan tahun terbit. Untuk update data sendiri, method yang digunakan adalah POST. Untuk link melakukan update data adalah <https://rifqipratama9.000webhostapp.com/api/buku/update.php>.

Berikut ini adalah data buku sebelum diupdate atau diubah.

The screenshot shows the Postman interface with a successful API call. The URL is `https://rifqipratama9.000webhostapp.com/api/buku/listAll.php`. The response body is a JSON array of books:

```

1  {
2   "buku": [
3     {
4       "id_buku": "1",
5       "judul": "Islam dan Akal Merdeka",
6       "penulis": "Sega Arsy",
7       "penerbit": "Mohammad Natsir",
8       "deskripsi": "Buku ini adalah buku kritik atas pemikiran Soekarno tentang Islam Sontoloyo dan seputar pembaharuan pemikiran Islam",
9       "bahasa": "Indonesia",
10      "genre": "Sejarah Indonesia",
11      "jumlah_halaman": "172",
12      "tahun_terbit": "2018"
13    },
14    {
15      "id_buku": "2",
16      "judul": "Islam dan Akal Merdeka",
17      "penulis": "Sega Arsy",
18      "penerbit": "Mohammad Natsir",
19      "deskripsi": "Buku ini berisi tentang kritik atas pemikiran Soekarno tentang Islam Sontoloyo dan seputar pembaharuan pemikiran"
}

```

Gambar 26. List Sebelum Update Data

Kita akan melakukan update atau edit data dengan id buku yaitu 1

The screenshot shows the Postman interface with a successful API call. The URL is `https://rifqipratama9.000webhostapp.com/api/buku/update.php`. The request body is a JSON object:

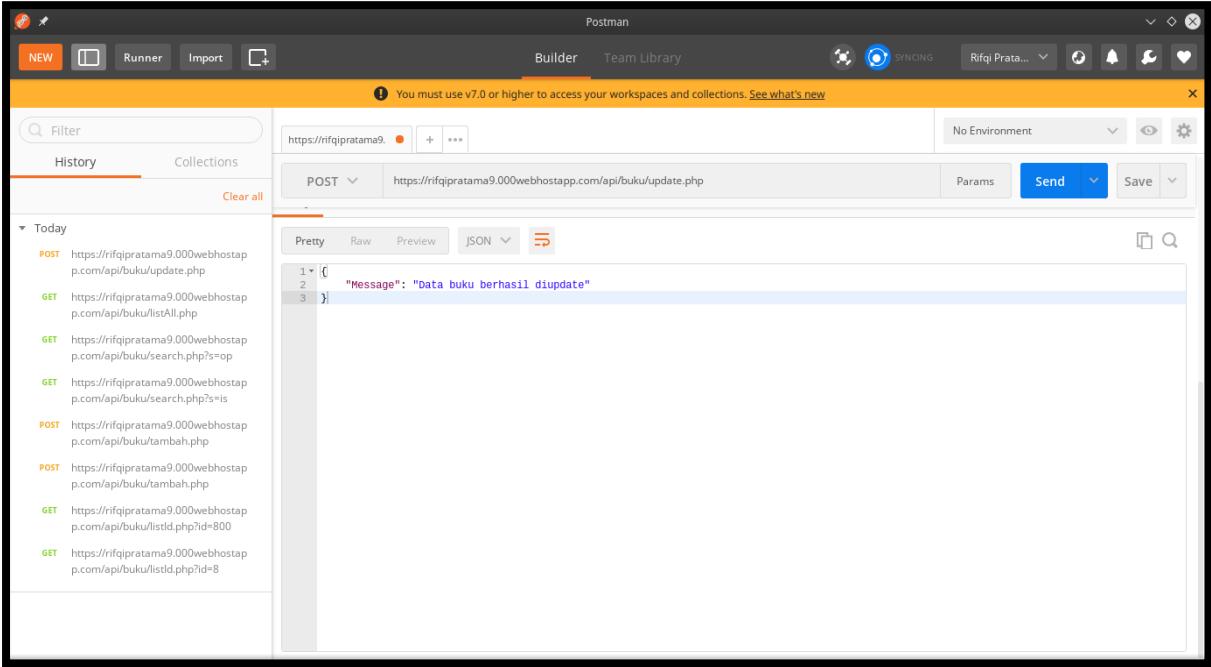
```

1  {
2   "id_buku": 1,
3   "judul": "Islam dan Akal Merdeka",
4   "penulis": "Mohammad Natsir",
5   "penerbit": "Sega Arsy",
6   "deskripsi": "Buku ini adalah buku kritik atas pemikiran Soekarno tentang Islam Sontoloyo dan seputar pembaharuan pemikiran Islam sekitar tahun 1934 - 1940",
7   "bahasa": "Indonesia",
8   "genre": "Sejarah",
9   "jumlah_halaman": 172,
10  "tahun_terbit": "2018"
}

```

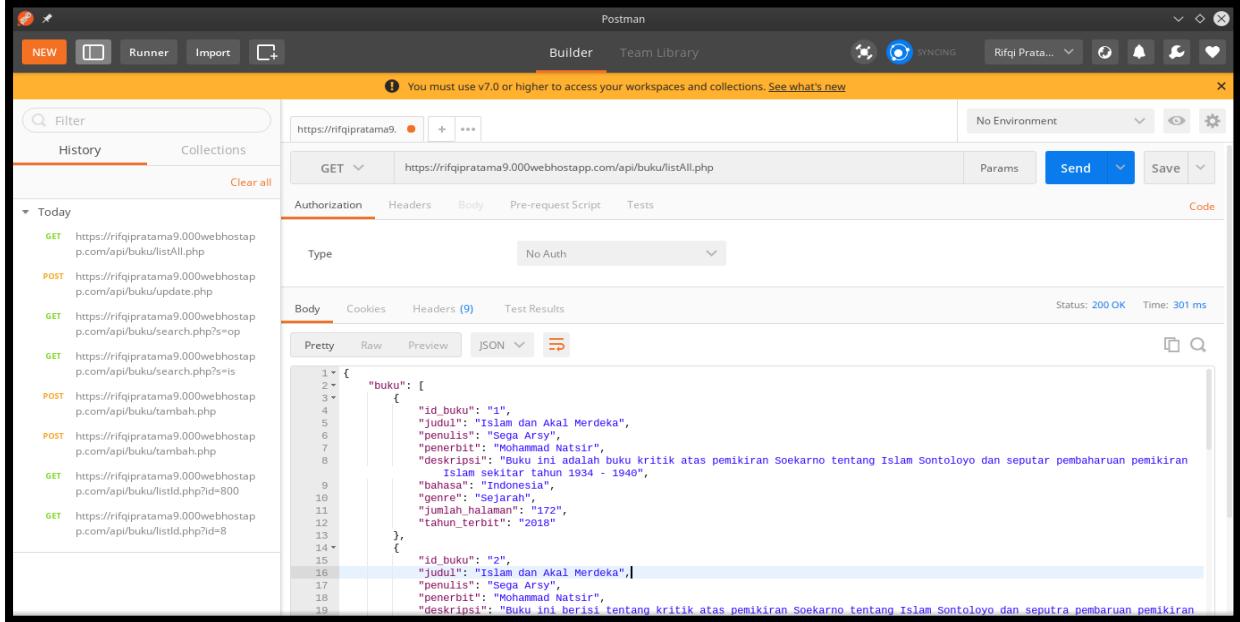
Gambar 27. Update Data

Jika proses update atau edit sudah berhasil, maka akan mengembalikan pesan berhasil



The screenshot shows the Postman application interface. In the center, there is a request builder window for a POST method to the URL <https://rifqipratama9.000webhostapp.com/api/buku/update.php>. The request body is set to JSON and contains the message: "Message": "Data buku berhasil diupdate". Below the request builder, the left sidebar shows a history of API calls made today, including various GET and POST requests to the same endpoint, all of which have been successful.

Gambar 28. Update Berhasil



The screenshot shows the Postman application interface after a successful update. It displays a GET request to the URL <https://rifqipratama9.000webhostapp.com/api/buku/listAll.php>. The response body is shown in JSON format, containing a list of books. One book entry is highlighted, showing its details: "id_buku": "1", "judul": "Islam dan Akal Merdeka", "penulis": "Sega Arsy", "penerbit": "Mohammad Natsir", "deskripsi": "Buku ini adalah buku kritik atas pemikiran Soekarno tentang Islam Sontoloyo dan seputar pembaharuan pemikiran Islam sekitar tahun 1934 - 1940", "bahasa": "Indonesia", "genre": "Sejarah", "jumlah_halaman": "172", "tahun_terbit": "2018". This indicates that the book data has been successfully updated in the database.

Gambar 29. Data Setelah Diupdate

4.5 Delete Data

Untuk menghapus data kita juga butuh body untuk dikirimkan, body yang dikirimkan berisi id buku yang akan kita hapus datanya. Untuk delete data sendiri menggunakan method POST dan untuk mengaksesnya dapat melalui link :

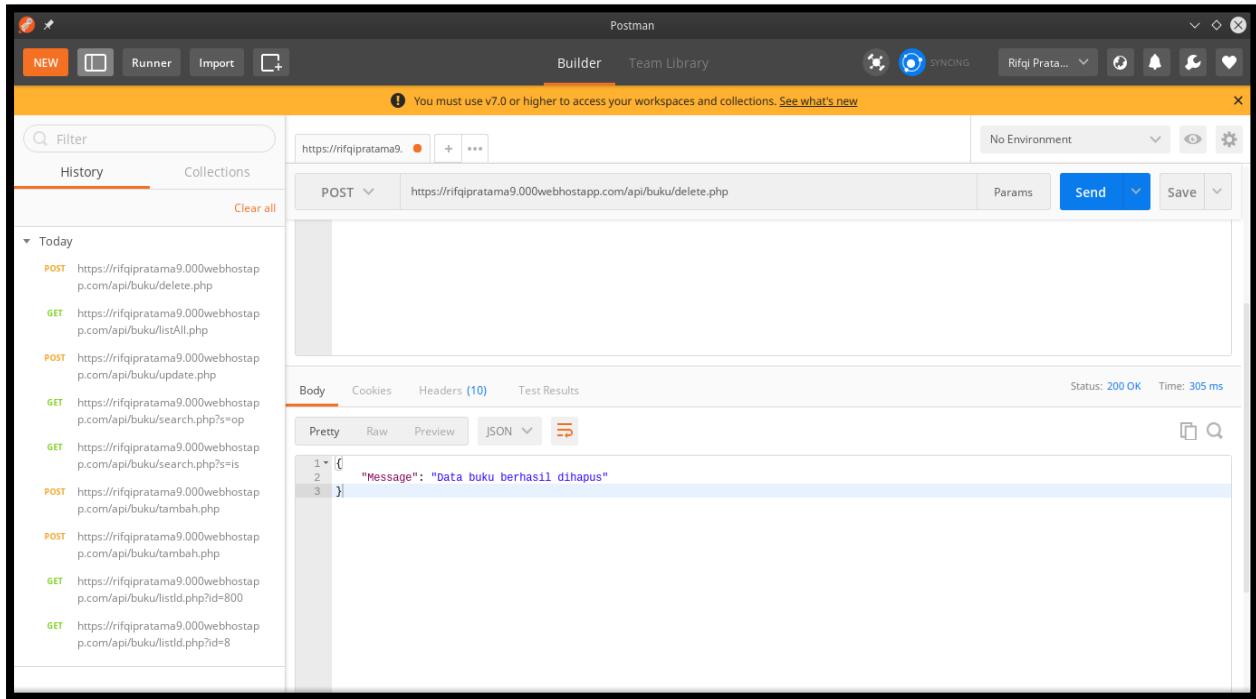
<https://rifqipratama9.000webhostapp.com/api/buku/delete.php>

Dibawah ini kita akan menghapus data dengan id buku : 10

The screenshot shows the Postman application interface. In the top navigation bar, 'Builder' is selected. The URL field contains 'https://rifqipratama9.000webhostapp.com/api/buku/delete.php'. The 'Body' tab is active, showing a JSON payload: { "id_buku": 10 }. The status bar at the bottom right indicates 'Status: 200 OK' and 'Time: 301 ms'. The left sidebar shows a history of API requests.

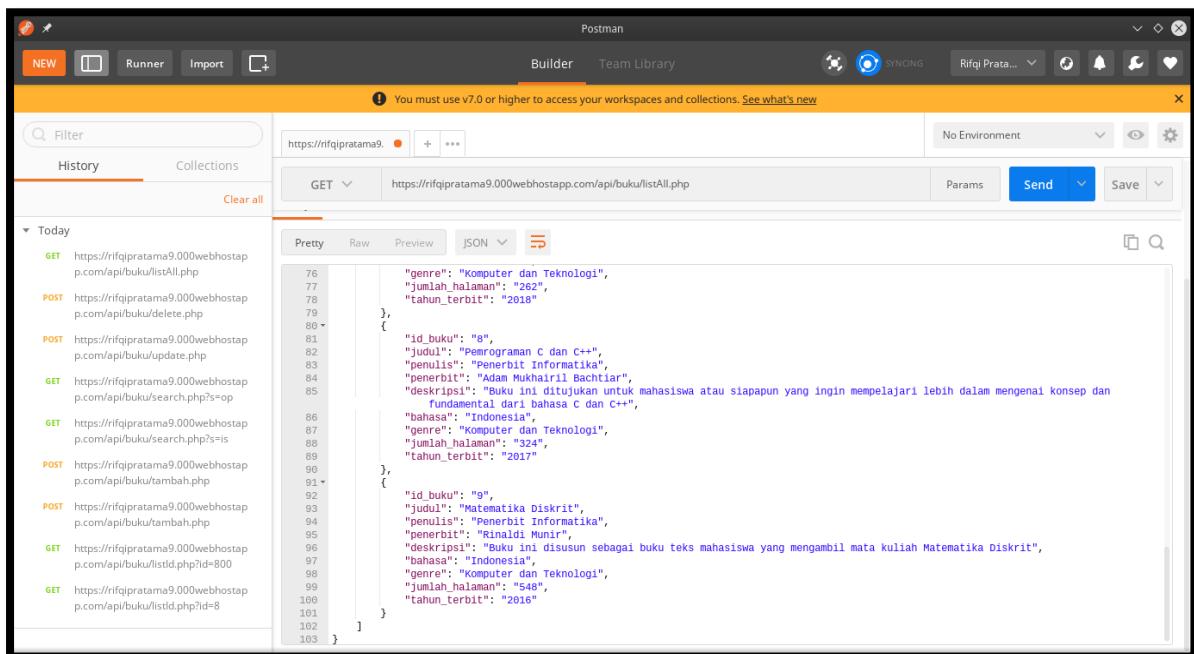
Gambar 30. Delete Data

Jika data berhasil kita hapus, maka akan mengembalikan pesan berhasil seperti ini



Gambar 31. Delete Data Berhasil

Maka data yang tadi dihapus akan hilang dari database

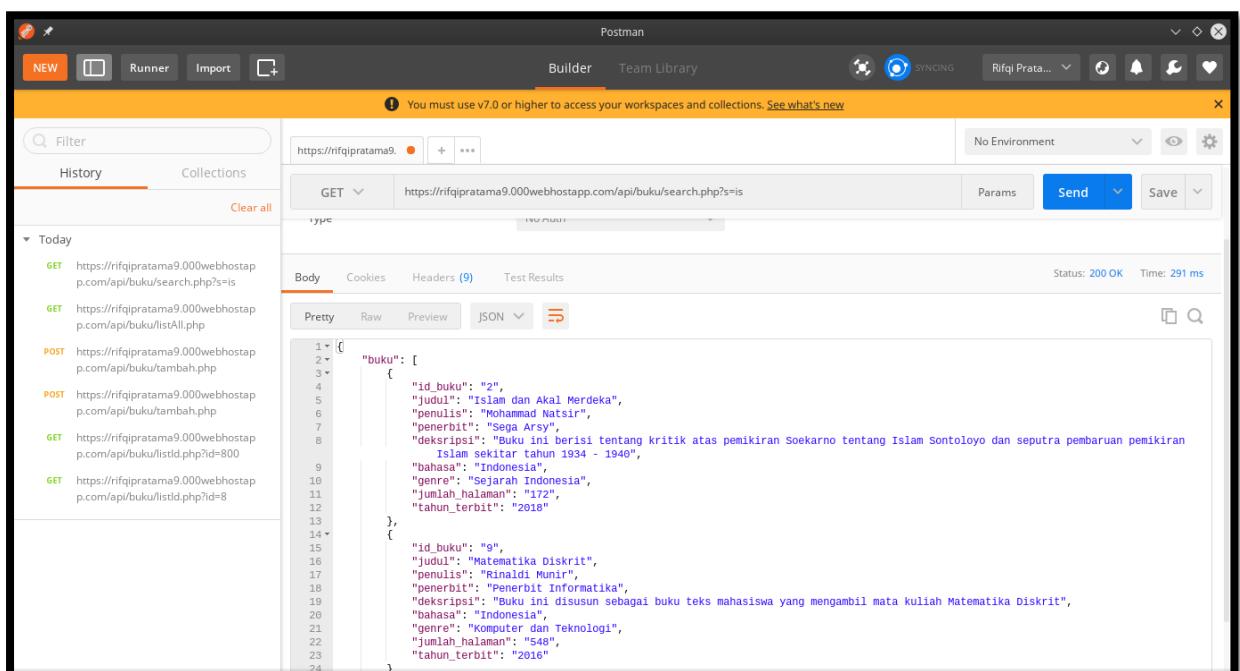


Gambar 32. List Setelah Dihapus

4.6 Search Data

Untuk melakukan proses cari data kita menggunakan method GET, pada link kita perlu menambahkan query untuk menjadi parameter data yang akan dicari. Untuk link search data sendiri adalah <https://rifqipratama9.000webhostapp.com/api/buku/search.php?s={s}>, dengan s sebagai query untuk parameter pencarian.

Dibawah ini kita akan melakukan pencarian dengan query s: is

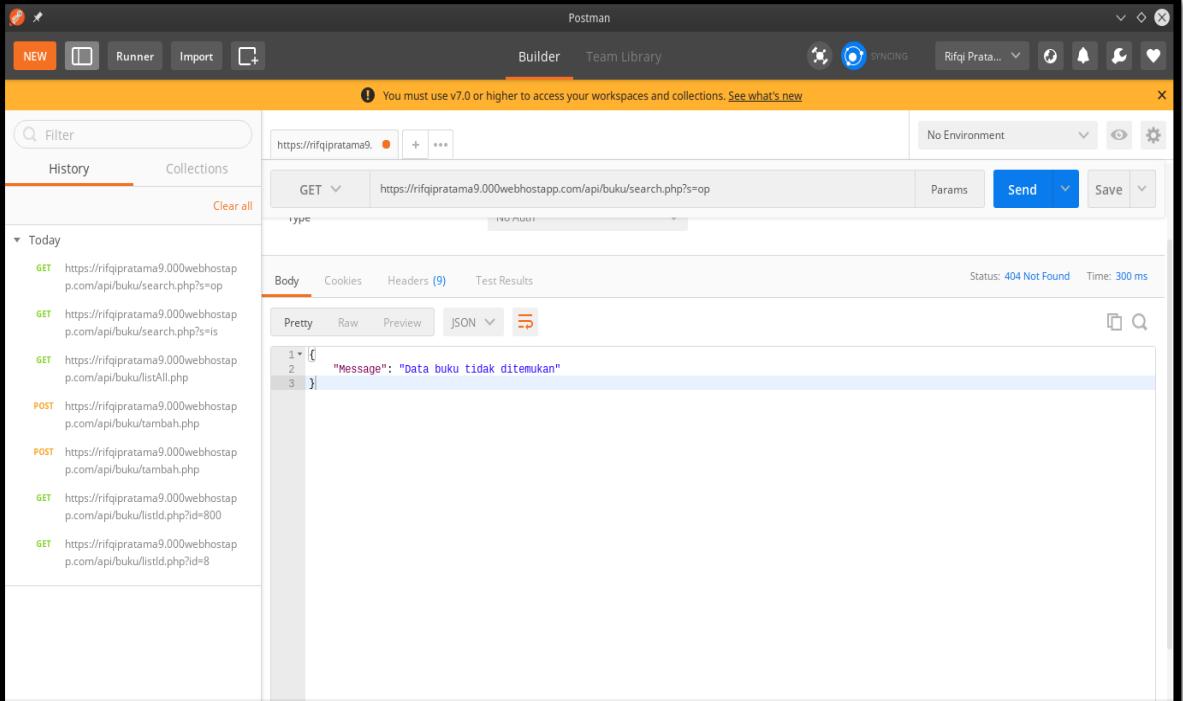


The screenshot shows the Postman application interface. In the top navigation bar, 'Builder' is selected. The URL field contains 'https://rifqipratama9.000webhostapp.com/api/buku/search.php?s=is'. The response body is displayed in JSON format, showing two book entries:

```
[{"buku": [{"id_buku": "2", "judul": "Islam dan Akal Merdeka", "penulis": "Mohammad Natsir", "penerbit": "Sega Arsy", "deskripsi": "Buku ini berisi tentang kritik atas pemikiran Soekarno tentang Islam Sontoloyo dan seputra pembaruan pemikiran Islam sekitar tahun 1934 - 1940.", "bahasa": "Indonesia", "genre": "Sejarah Indonesia", "jumlah_halaman": "172", "tahun_terbit": "2018"}, {"id_buku": "9", "judul": "Matematika Diskrit", "penulis": "Rinaldi Munir", "penerbit": "Penerbit Informatika", "deskripsi": "Buku ini disusun sebagai buku teks mahasiswa yang mengambil mata kuliah Matematika Diskrit", "bahasa": "Indonesia", "genre": "Komputer dan Teknologi", "jumlah_halaman": "548", "tahun_terbit": "2016"}]}
```

Gambar 33. Search Data

Jika data yang kita cari tidak dapat ditemukan, maka akan mengembalikan pesan gagal



The screenshot shows the Postman application interface. In the center, there is a request configuration window with the following details:

- Method: GET
- URL: <https://rifqipratama9.000webhostapp.com/api/buku/search.php?s=op>
- Headers: (9)
- Status: 404 Not Found
- Time: 300 ms

The "Body" tab is selected, showing a JSON response:

```
1 [ {  
2     "Message": "Data buku tidak ditemukan"  
3 } ]
```

Gambar 34. Search Data Tidak Ditemukan