API DOCUMENTATION

Base/Root URL : http://localhost:8000/api

1. Create Book

membuat/menambahkan data buku ke dalam database firebase.

Endpoint :

POST /api/books

Request Body :

{ "title": "String", "author": "String", "published_at": "YYYY-MM-DD" }

- title (required) : nama buku

- author (required) : nama penulis buku

- published_at (required) : tanggal buku dipublikasikan

Response :

- Status Code : 201 Created

{ "message": "Buku berhasil ditambahkan", "data": { "id": "string", "title": "String", "author": "String", "published_at": "YYYY-MM-DD", "created_at": "ISO date", "updated_at": "ISO date" } }

- Status Code : 400 Bad Request

{ "message": "Semua data wajib diisi" }

2. Read Books

Mengambil dan menampilkan daftar semua buku yang ada di database firebase.

Endpoint :

GET /api/books

Response :

- **Status Code**: 200 OK

{ "data": [{ "id": "string", "title": "String", "author": "String", "published_at": "YYYY-MM-DD" }, ...] }

3. Update Books

Memperbarui informasi buku berdasarkan ID.

Endpoint :

PUT /api/books/:id

Request Body :

{ "title": "String", "author": "String", "published_at": "YYYY-MM-DD" }

- title (optional) : nama buku

- author (optional) : nama penulis buku

- published_at (optional) : tanggal buku dipublikasikan

Response :

- Status Code : 200 OK

{ "message": "Buku berhasil diupdate", "data": { "id": "string", "title": "String", "author": "String", "published_at": "YYYY-MM-DD", "updated_at": "ISO date" } }

- Status Code : 404 Not Found

{ "message": "Buku tidak ditemukan" }

4. **Delete Books**

Menghapus buku berdasarkan ID.

Endpoint :

DELETE /api/books/:id

Response :

- Status Code : 200 OK

{ "message": "Buku berhasil dihapus" }

- Status Code : 404 Not Found

{ "message": "Buku tidak ditemukan" }

5. Error Handling

Jika endpoint yang dimasukkan tidak ditemukan, maka API akan memberikan respons :

Response :

Status Code : 404 Not Found

{ "message": "Endpoint tidak ditemukan" }