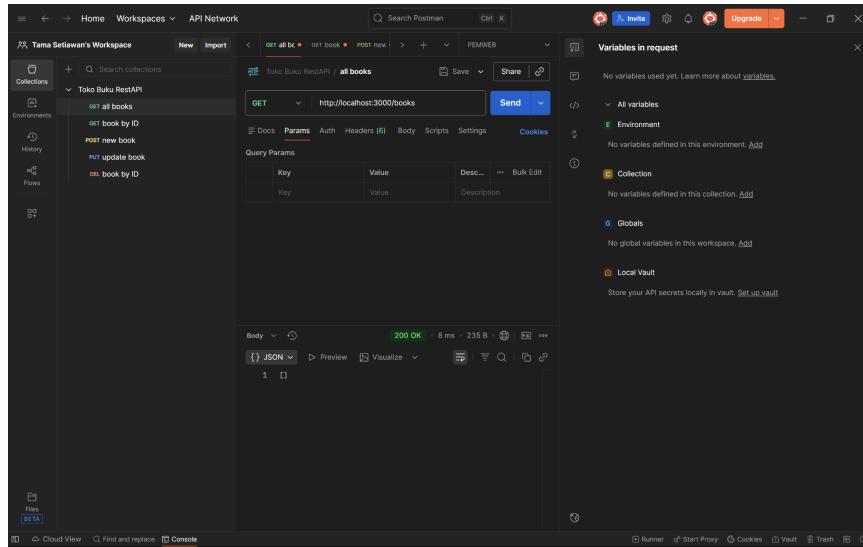


Nama: Muh. Rahmatullah Setiawan  
NIM: D121231076

## Dokumentasi POSTMAN

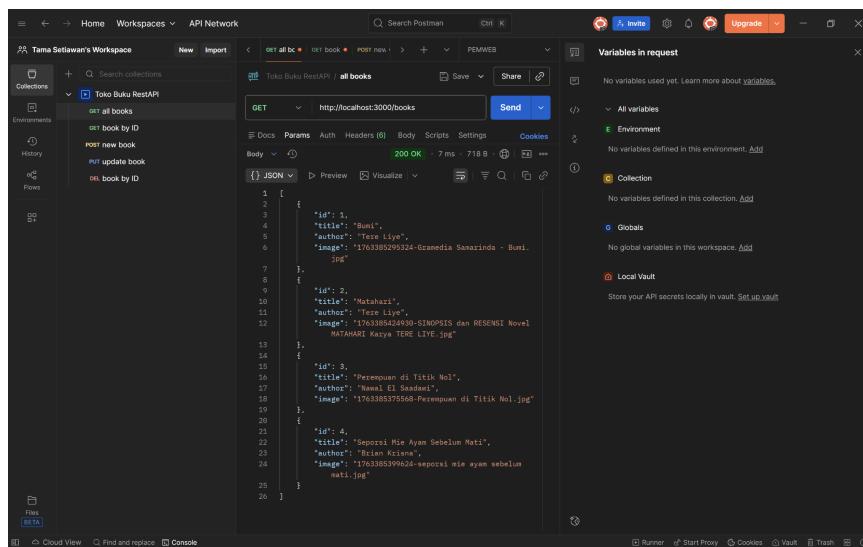
### 1. GET all books <http://localhost:3000/books>

- Request saat database kosong:



The screenshot shows the Postman interface with a collection named "Toko Buku RestAPI". A GET request is made to "http://localhost:3000/books". The response status is 200 OK, and the body is an empty JSON array: `[ ]`.

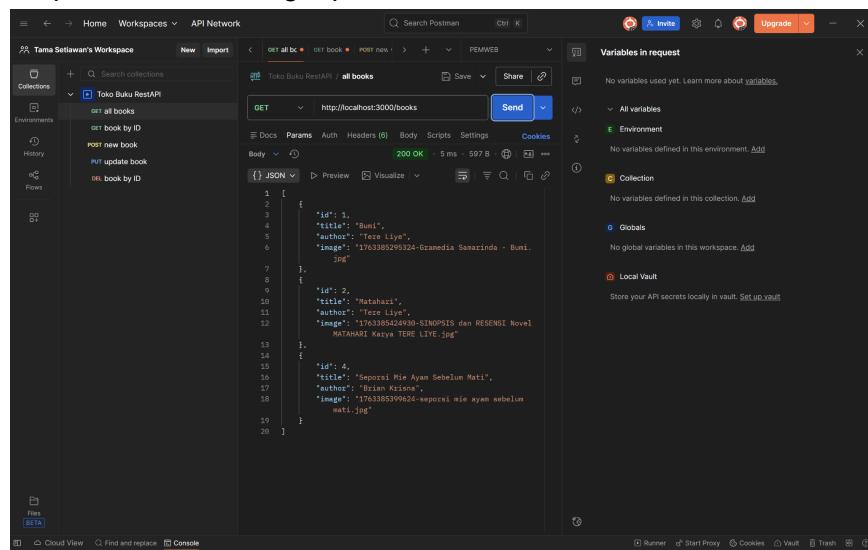
- Request setelah database terisi:



The screenshot shows the Postman interface with a collection named "Toko Buku RestAPI". A GET request is made to "http://localhost:3000/books". The response status is 200 OK, and the body is a JSON array containing four book records:

```
1 [  
2   {  
3     "id": 1,  
4     "title": "Bumi",  
5     "author": "Tere Liye",  
6     "image": "1763380205324-Gramedia Sarazinda - Bumi.jpg"  
7   },  
8   {  
9     "id": 2,  
10    "title": "Mahazir",  
11    "author": "Tere Liye",  
12    "image": "176338524980-SINOPSIS dan RESENSI Novel Mahazir Nasya TERE LIYE.jpg"  
13  },  
14  {  
15    "id": 3,  
16    "title": "Perempuan di Titik Nol",  
17    "author": "Nawal El Saadawi",  
18    "image": "1763385375568-Perempuan di Titik Nol.jpg"  
19  },  
20  {  
21    "id": 4,  
22    "title": "Seporsi Mie Ayam Sebelum Mati",  
23    "author": "Brian Kriana",  
24    "image": "1763385399624-seporpsi mie ayam sebelum mati.jpg"  
25  }  
26]
```

- Request setelah menghapus buku:

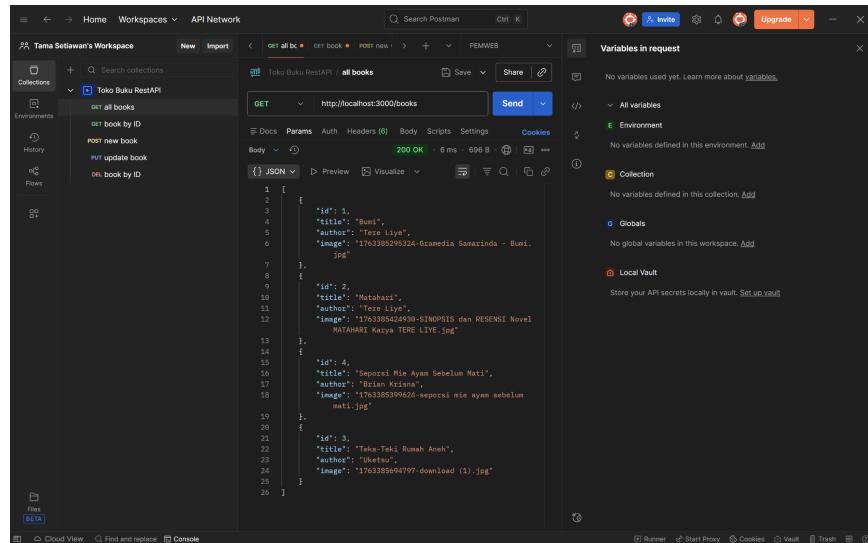


```

[{"id": 1, "title": "Bumi", "author": "Tere Liye", "image": "1763385295324-Gramedia Samarinda - Bumi.jpg"}, {"id": 2, "title": "Mathazai", "author": "Tere Liye", "image": "1763385295324-SINOPSIS dan RESENSI Novel MATAHARI Karya TERE LIYE.jpg"}, {"id": 4, "title": "Seporsi Mie Ayam Sebelum Mati", "author": "Brian Krasni", "image": "1763385399624-seporpsi mie ayam sebelum mati.jpg"}]

```

- Request setelah menambahkan buku baru (mengisi id = 3 yang kosong):



```

[{"id": 1, "title": "Bumi", "author": "Tere Liye", "image": "1763385295324-Gramedia Samarinda - Bumi.jpg"}, {"id": 2, "title": "Mathazai", "author": "Tere Liye", "image": "1763385295324-SINOPSIS dan RESENSI Novel MATAHARI Karya TERE LIYE.jpg"}, {"id": 3, "title": "", "author": "", "image": ""}, {"id": 4, "title": "Seporsi Mie Ayam Sebelum Mati", "author": "Brian Krasni", "image": "1763385399624-seporpsi mie ayam sebelum mati.jpg"}]

```

## 2. GET by ID <http://localhost:3000/books/:id>

- Request saat database kosong:

The screenshot shows the Postman interface with a collection named "Toko Buku RestAPI". A GET request is made to `http://localhost:3000/books/1`. The response status is 404 Not Found, with a message: "Book not found".

- Request book ID = 1:

The screenshot shows the Postman interface with a collection named "Toko Buku RestAPI". A GET request is made to `http://localhost:3000/books/1`. The response status is 200 OK, returning a JSON object with book details: { "id": 1, "title": "Bumi", "author": "Eric Llye", "image": "2763385295324-Gramedia Samarinda - Bumi.jpg" }.

- Request book ID = 2:

The screenshot shows the Postman application interface. In the left sidebar, under 'Collections', there is a section for 'Toko Buku RestAPI' containing several requests: 'get all books', 'get book by ID', 'POST new book', 'PUT update book', and 'DELETE book by ID'. The 'get book by ID' request is selected. The main panel displays a GET request to 'http://localhost:3000/books/2'. The response status is '200 OK' with a response time of 6 ms and a body size of 364 B. The response body is a JSON object:

```
{ "id": 2, "title": "Mataharis", "author": "Rebecca Lobo", "image": "176338642490-SINOPSIS dan RESENSI Novel MATAHARIS Karya TERRI LIEY.jpg" }
```

- Request book ID = 3 (Old):

The screenshot shows the Postman application interface. In the left sidebar, under 'Collections', there is a section for 'Toko Buku RestAPI' containing several requests: 'get all books', 'get book by ID', 'POST new book', 'PUT update book', and 'DELETE book by ID'. The 'get book by ID' request is selected. The main panel displays a GET request to 'http://localhost:3000/books/3'. The response status is '200 OK' with a response time of 3 ms and a body size of 356 B. The response body is a JSON object:

```
{ "id": 3, "title": "Perempuan di Titik Nol", "author": "Nasai El Saadai", "image": "176338642390-Perempuan di Titik Nol.jpg" }
```

- Request ID = 4:

The screenshot shows the Postman application interface. In the left sidebar, under 'Collections', there is a section for 'Toko Buku RestAPI' containing several requests: 'get all books', 'GET book by ID', 'POST new book', 'PUT update book', and 'DELETE book by ID'. The 'GET book by ID' request is selected. The 'Body' tab shows a JSON response for book ID 4, which includes fields like id, title, author, and image. The status bar at the bottom indicates a 200 OK response.

```
{} JSON
1 {
2   "id": 4,
3   "title": "Seporsi Mie Ayam Sebelum Mati",
4   "author": "Bryan Krishna",
5   "image": "1763385399024-seporosi mie ayam sebelum mati.jpg"
6 }
```

- Request ID = 3 (New):

The screenshot shows the Postman application interface. Under the same 'Toko Buku RestAPI' collection, the 'GET book by ID' request is selected. The 'Body' tab shows a JSON response for book ID 3, which includes fields like id, title, author, and image. The status bar at the bottom indicates a 200 OK response.

```
{} JSON
1 {
2   "id": 3,
3   "title": "Perempuan di Titik Nol",
4   "author": "Nasril El Seadewi",
5   "image": "176338453396-Perempuan di Titik Nol.jpg"
6 }
```

### 3. POST new book <http://localhost:3000/books>

- Request saat tidak mengisi seluruh field:

The screenshot shows the Postman interface with a collection named "Toko Buku RestAPI". A POST request is selected with the URL `http://localhost:3000/books`. The request body is set to "form-data" and contains one field named "title" with a value of "Text". When the "Send" button is clicked, the response status is 400 Bad Request, and the JSON response is: 

```
{ "message": "Request body must include 'title' and 'author'" }
```

- Request saat tidak mengisi salah satu main field:

The screenshot shows the Postman interface with the same collection and request setup as the previous screenshot. The request body now includes three fields: "title" (Text), "author" (Text), and "cover" (File). The "cover" field has a placeholder "Select files". The response status is 400 Bad Request, and the JSON response is: 

```
{ "message": "Request body must include 'title' and 'author'" }
```

- Request menambahkan buku baru:

The image contains two side-by-side screenshots of the Postman application interface, both titled "Toko Buku RestAPI / new book".

**Screenshot 1 (Top):**

- Request Details:** POST http://localhost:3000/books
- Body (form-data):**

Key	Type	Value	Description
title	Text	Seporsi Mie Ayam Sebel...	
author	Text	Brian Krisna	
cover	File	seporasi mie ay...	
- Response:** 201 Created

```
{
  "id": 2,
  "title": "Seporsi Mie Ayam Sebelum Mati",
  "author": "Brian Krisna",
  "image": "176338399024-seporasi mie ayam sebelum mati.jpg"
}
```

**Screenshot 2 (Bottom):**

- Request Details:** POST http://localhost:3000/books
- Body (form-data):**

Key	Type	Value	Description
title	Text	Perempuan di Titik Nol	
author	Text	Nawal El Saadawi	
cover	File	Perempuan di ...	
- Response:** 201 Created

```
{
  "id": 3,
  "title": "Perempuan di Titik Nol",
  "author": "Nawal El Saadawi",
  "image": "1763385375568-Perempuan di Titik Nol.jpg"
}
```

Variables in request

Key	Value	Description
<input checked="" type="checkbox"/> title	Text	Matahari
<input checked="" type="checkbox"/> author	Text	Tere Liye
<input checked="" type="checkbox"/> cover	File	<input type="file" value="Gramedia Sam..."/>

Body

```
{
  "id": 2,
  "title": "Matahari",
  "author": "Tere Liye",
  "image": "176338532783-Gramedia Sam... - Bumi.jpg"
}
```

Variables in request

Key	Value	Description
<input checked="" type="checkbox"/> title	Text	Bumi
<input checked="" type="checkbox"/> author	Text	Tere Liye
<input checked="" type="checkbox"/> cover	File	<input type="file" value="Gramedia Sam..."/>

Body

```
{
  "id": 1,
  "title": "Bumi",
  "author": "Tere Liye",
  "image": "1763385295324-Gramedia Sam... - Bumi.jpg"
}
```

## - Request menambahkan buku baru setelah request DEL:

Variables in request

Key	Value	Description
<input checked="" type="checkbox"/> title	Text	Teka-Teki Rumah Aneh
<input checked="" type="checkbox"/> author	Text	Uketsu
<input checked="" type="checkbox"/> cover	File	<input type="file" value="download (1).jpg"/>

Body

```
{
  "id": 3,
  "title": "Teka-Teki Rumah Aneh",
  "author": "Uketsu",
  "image": "2763385694797-download (1).jpg"
}
```

#### 4. PUT update book <http://localhost:3000/books/:id>

- Request saat database kosong:

The screenshot shows the Postman interface with a collection named "Toko Buku RestAPI". A PUT request is made to <http://localhost:3000/books/3>. The request body is empty. The response status is 404 Not Found, and the response body contains the JSON object: { "message": "Book not found" }.

- Request update informasi buku:

The screenshot shows the Postman interface with the same collection. A PUT request is made to <http://localhost:3000/books/2>. The request body contains the following JSON: { "id": 2, "title": "Masabir", "author": "Tere Liye", "image": "5753852d4930-530-SINOPSIS dan RESENSI Novel MATAHARI Karya TERE LIYE.jpg" }. The response status is 200 OK, and the response body contains the updated book details: { "id": 2, "title": "Masabir", "author": "Tere Liye", "image": "5753852d4930-530-SINOPSIS dan RESENSI Novel MATAHARI Karya TERE LIYE.jpg" }.

## 5. DEL book by ID <http://localhost:3000/books/:id>

- Request saat database kosong:

The screenshot shows the Postman interface with a collection named "Toko Buku RestAPI". A DELETE request is made to `http://localhost:3000/books/36`. The response body is a JSON object with a single key-value pair: `{"message": "Book not found"}`.

- Request menghapus buku:

The screenshot shows the Postman interface with a collection named "Toko Buku RestAPI". A DELETE request is made to `http://localhost:3000/books/3`. The response status is `200 OK` with a response time of 14 ms and a size of 403 B. The response body is a JSON object with two keys: `"message"` and `"data"`. The `"message"` field contains the value `"Book deleted successfully"`. The `"data"` field contains a detailed book object with fields: `id`, `title`, `author`, and `image`. The `id` field has a value of 3. The `title` field has a value of `"Perempuan di Titik Nol"`. The `author` field has a value of `"Nawal El Saadawi"`. The `image` field has a value of `"1763389375680-Perempuan di Titik Nol.jpg"`.

## Dokumentasi Log:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (e.g. text, !exclude, \esca...    
```

```
D:\College App\Nodejs\node.exe .\index.js
Server running at http://localhost:3000
[1763385111] GET /books                                         index.js:29
[1763385114] GET /books                                         logger.js:3
[1763385115] GET /books                                         logger.js:3
[1763385190] GET /books                                         logger.js:3
[1763385191] GET /books                                         logger.js:3
[1763385204] DELETE /books/1                                    logger.js:3
[1763385205] DELETE /books/2                                    logger.js:3
[1763385241] GET /books                                         logger.js:3
[1763385259] GET /books/1                                       logger.js:3
[1763385295] POST /books                                        logger.js:3
[1763385327] POST /books                                        logger.js:3
[1763385375] POST /books                                        logger.js:3
[1763385399] POST /books                                        logger.js:3
[1763385424] PUT /books/2                                       logger.js:3
[1763385437] GET /books                                         logger.js:3
[1763385470] DELETE /books/3                                     logger.js:3
Deleted file: 1763385375568-Perempuan di Titik Nol.jpg          bookController.js:87
[1763385489] GET /books                                         logger.js:3
[1763385508] GET /books/4                                       logger.js:3
[1763385694] POST /books                                        logger.js:3
[1763385706] GET /books                                         logger.js:3
[1763386340] GET /books/1                                       logger.js:3
[1763386352] GET /books/2                                       logger.js:3
[1763386360] GET /books/3                                       logger.js:3
[1763386425] PUT /books/2                                       logger.js:3
[1763386437] PUT /books/2                                       logger.js:3
[1763386453] PUT /books/3                                       logger.js:3
[1763386458] GET /books/3                                       logger.js:3
[1763387070] POST /books                                        logger.js:3
[1763387188] POST /books                                        logger.js:3
[1763387519] PUT /books/6                                       logger.js:3
```

Dokumentasi Struktur File:

```
✓ REST-API TOKO BUKU
  ✓ controllers
    JS bookController.js
  ✓ data
    {} books.json M
  ✓ images
    📸 1763385295324-Gramedia Samarinda... U
    📸 1763385399624-seporsi mie ayam se... U
    📸 1763385424930-SINOPSIS dan RESEN... U
    📸 1763385694797-download (1).jpg U
  ✓ middleware
    JS logger.js
    JS upload.js
    JS validateBook.js
  > node_modules
  ✓ routes
    JS bookRoutes.js
    JS index.js
    {} package-lock.json
    {} package.json
    ⓘ README.md
```