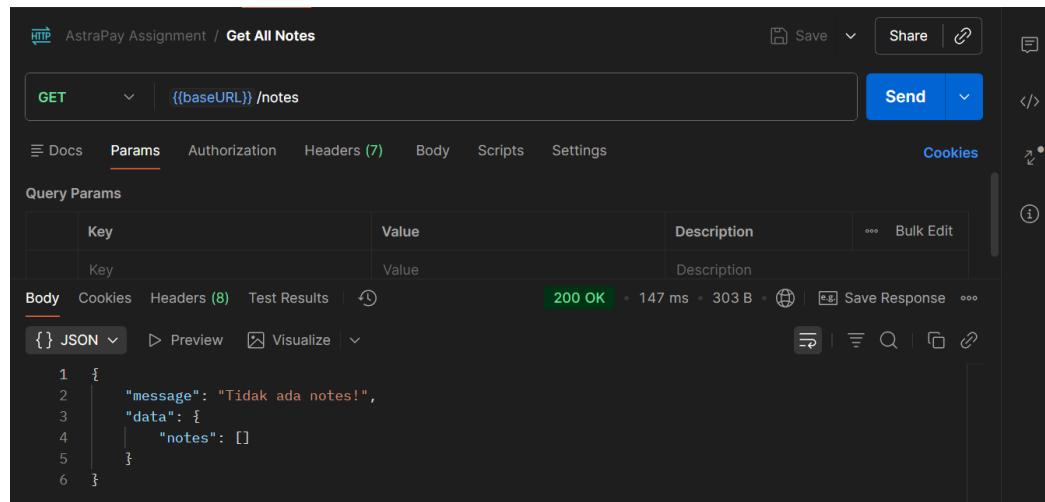


1. Get All Notes

a. No Note

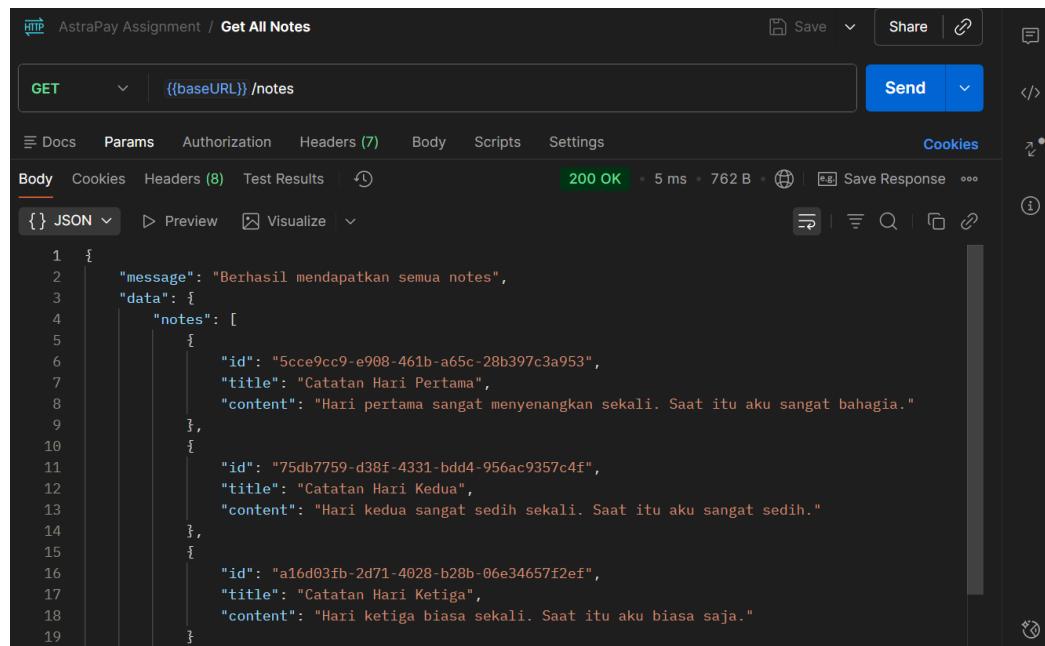


A screenshot of the Postman application interface. The URL bar shows 'AstraPay Assignment / Get All Notes'. The method dropdown is set to 'GET'. The 'Params' tab is selected. The 'Body' tab is selected. The response status is '200 OK' with a response time of 147 ms and a body size of 303 B. The response content is a JSON object:

```
1 {  
2   "message": "Tidak ada notes!",  
3   "data": {  
4     "notes": []  
5   }  
6 }
```

postman request 'localhost:8760/api/notes'

b. Note Exist



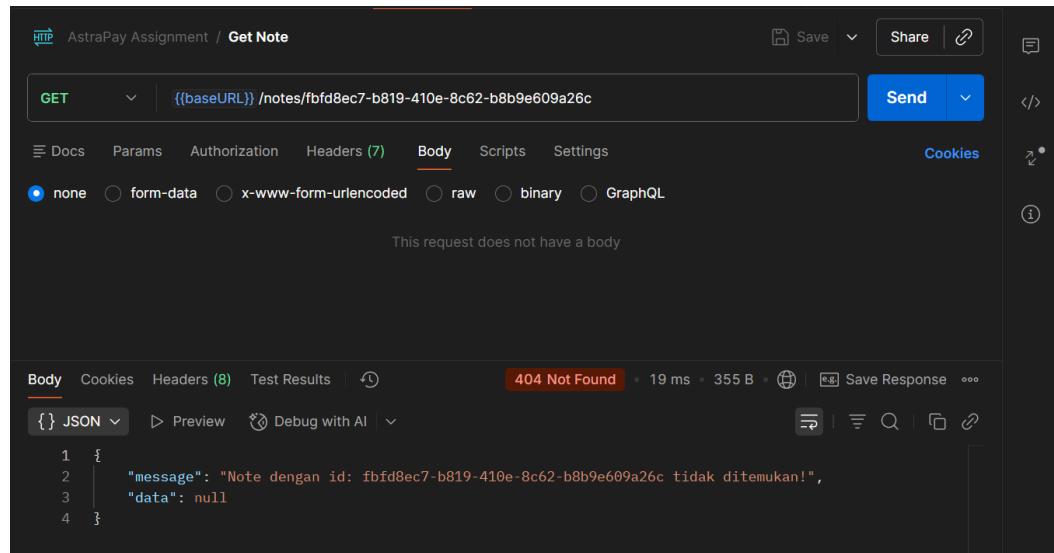
A screenshot of the Postman application interface. The URL bar shows 'AstraPay Assignment / Get All Notes'. The method dropdown is set to 'GET'. The 'Params' tab is selected. The 'Body' tab is selected. The response status is '200 OK' with a response time of 5 ms and a body size of 762 B. The response content is a JSON object:

```
1 {  
2   "message": "Berhasil mendapatkan semua notes",  
3   "data": {  
4     "notes": [  
5       {  
6         "id": "5cce9cc9-e908-461b-a65c-28b397c3a953",  
7         "title": "Catatan Hari Pertama",  
8         "content": "Hari pertama sangat menyenangkan sekali. Saat itu aku sangat bahagia."  
9       },  
10      {  
11        "id": "75db7759-d38f-4331-bdd4-956ac9357c4f",  
12        "title": "Catatan Hari Kedua",  
13        "content": "Hari kedua sangat sedih sekali. Saat itu aku sangat sedih."  
14      },  
15      {  
16        "id": "a16d03fb-2d71-4028-b28b-06e34657f2ef",  
17        "title": "Catatan Hari Ketiga",  
18        "content": "Hari ketiga biasa sekali. Saat itu aku biasa saja."  
19      }  
20    ]  
21  }  
22 }
```

postman request 'localhost:8760/api/notes'

2. Get One Note

a. Not Exist

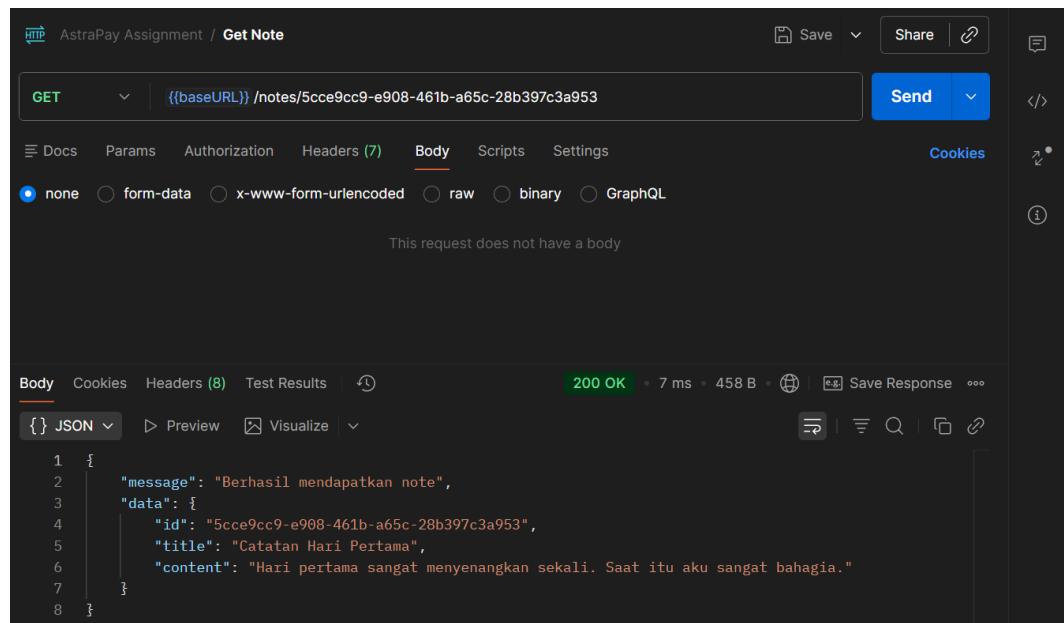


A screenshot of the Postman application interface. The request URL is `GET {{baseURL}} /notes/fbfd8ec7-b819-410e-8c62-b8b9e609a26c`. The response status is **404 Not Found** with a message: "Note dengan id: fbfd8ec7-b819-410e-8c62-b8b9e609a26c tidak ditemukan!", and the data field is null.

postman request

'localhost:8760/api/notes/fbfd8ec7-b819-410e-8c62-b8b9e609a26c'

b. Exist



A screenshot of the Postman application interface. The request URL is `GET {{baseURL}} /notes/5cce9cc9-e908-461b-a65c-28b397c3a953`. The response status is **200 OK** with a message: "Berhasil mendapatkan note", and the data field contains a note object with id, title, and content.

```
1 {  
2   "message": "Berhasil mendapatkan note",  
3   "data": {  
4     "id": "5cce9cc9-e908-461b-a65c-28b397c3a953",  
5     "title": "Catatan Hari Pertama",  
6     "content": "Hari pertama sangat menyenangkan sekali. Saat itu aku sangat bahagia."  
7   }  
8 }
```

postman request

'localhost:8760/api/notes/5cce9cc9-e908-461b-a65c-28b397c3a953'

3. Add New Note

a. Success

The screenshot shows the Postman interface for a 'POST' request to 'localhost:8760/api/notes'. The request body is a JSON object with 'title' and 'content' fields. The response status is '201 Created' with a message indicating success and providing the note's ID and details.

```
1 {
2   "title": "Catatan Hari Pertama",
3   "content": "Hari pertama sangat menyenangkan sekali. Saat itu aku sangat bahagia."
4 }
```

```
1 {
2   "message": "Note berhasil ditambahkan",
3   "data": {
4     "id": "5cce9cc9-e908-461b-a65c-28b397c3a953",
5     "title": "Catatan Hari Pertama",
6     "content": "Hari pertama sangat menyenangkan sekali. Saat itu aku sangat bahagia."
7   }
8 }
```

```
postman request POST 'localhost:8760/api/notes' \
--header 'Content-Type: application/json' \
--body '{
  "title": "Catatan Hari Pertama",
  "content": "Hari pertama sangat menyenangkan sekali. Saat itu aku sangat
bahagia."
}'
```

b. Required

The screenshot shows the Postman interface for a 'POST' request to 'localhost:8760/api/notes'. The request body is a JSON object with only a 'content' field. The response status is '400 Bad Request' with an error message indicating that the 'title' field is required.

```
1 {
2   "content": "Hari ketiga biasa sekali. Saat itu aku biasa
saja."
4 }
```

```
1 {
2   "message": "Title tidak boleh kosong",
3   "data": null
4 }
```

On the right side of the interface, there is a 'Postman CLI' panel showing the command used to run the request:

```
1 postman request POST
  'localhost:8760/api/
  notes' \
2 --header 'Content-Type:
  application/json' \
3 --body '{
4   "content": "Hari ketiga
  biasa sekali. Saat
  itu aku biasa saja."
6 }'
```

HTTP AstraPay Assignment / Add New Note

POST localhost:8760/api/notes

Body (9) Headers (9) Body Scripts Settings Cookies

raw JSON Schema Beautify

```
1 {  
2   "title": null,  
3   "content": "Hari ketiga biasa sekali. Saat itu aku biasa  
4   saja."  
}
```

Body 400 Bad Request 8 ms 283 B

```
1 {  
2   "message": "Title tidak boleh kosong",  
3   "data": null  
4 }
```

Postman CLI

```
1 postman request POST  
  'localhost:8760/api/  
  notes' \  
2 --header 'Content-Type:  
  application/json' \  
3 --body '{  
4   "title": null,  
5   "content": "Hari ketiga  
  biasa sekali. Saat  
  itu aku biasa saja."  
6 }'
```

Install Postman CLI

Use the command below to [install the Postman CLI](#) on macOS, Linux (x64) or Windows Subsystem for Linux (WSL)

```
npm install -g postman-cli
```

Learn how to [install the Postman CLI on Windows](#)

HTTP AstraPay Assignment / Add New Note

POST localhost:8760/api/notes

Body (9) Headers (9) Body Scripts Settings Cookies

raw JSON Schema Beautify

```
1 {  
2   "content": "Hari ketiga biasa sekali. Saat itu aku biasa  
3   saja."  
3 }
```

Body 400 Bad Request 9 ms 283 B

```
1 {  
2   "message": "Title tidak boleh kosong",  
3   "data": null  
4 }
```

Postman CLI

```
1 postman request POST  
  'localhost:8760/api/  
  notes' \  
2 --header 'Content-Type:  
  application/json' \  
3 --body '{  
4   "content": "Hari ketiga  
  biasa sekali. Saat  
  itu aku biasa saja."  
5 }'
```

Install Postman CLI

Use the command below to [install the Postman CLI](#) on macOS, Linux (x64) or Windows Subsystem for Linux (WSL)

```
npm install -g postman-cli
```

Learn how to [install the Postman CLI on Windows](#)

HTTP AstraPay Assignment / Add New Note

POST localhost:8760/api/notes

Body (8) Headers (8) Body Scripts Settings Cookies

raw JSON Schema Beautify

```
1 Ctrl+Alt+P to Ask AI
```

Body 400 Bad Request 8 ms 283 B

```
1 {  
2   "message": "Request body is required",  
3   "data": null  
4 }
```

Postman CLI

```
1 postman request POST  
  'localhost:8760/api/  
  notes' \  
2 --body ''
```

Install Postman CLI

Use the command below to [install the Postman CLI](#) on macOS, Linux (x64) or Windows Subsystem for Linux (WSL)

```
npm install -g postman-cli
```

Learn how to [install the Postman CLI on Windows](#)

4. Update Note

a. Success

The screenshot shows the Postman interface for an 'Update Note' request. The 'Body' tab is selected, containing the following JSON payload:

```
1 {
2   "title": "Update Hari Pertama",
3   "content": "Testing update"
4 }
```

The response status is '200 OK' with a response time of 19 ms and a body size of 401 B. The response JSON is:

```
1
2   "message": "Note berhasil diperbarui",
3   "data": {
4     "id": "fb18eee-8190-455f-b954-fe7f858032c2",
5     "title": "Update Hari Pertama",
6     "content": "Testing update"
7   }
8 }
```

b. Not Found

The screenshot shows the Postman interface for an 'Update Note' request. The 'Body' tab is selected, containing the same JSON payload as the success case.

The response status is '404 Not Found' with a response time of 10 ms and a body size of 355 B. The response JSON is:

```
1
2   "message": "Note dengan id: fb18eee-8190-455f-b254-fe7f858032c2 tidak ditemukan!",
3   "data": null
4 }
```

c. Required

The screenshot shows two separate instances of the Postman application interface, each displaying a request to update a note in an AstraPay Assignment API.

Request 1 (Top):

- Method: PUT
- URL: `PUT {{baseURL}}/notes/fbd18eee-8190-455f-b954-f... (9)`
- Body (JSON):

```
1 {
2     "title": "",
3     "content": "Testing update"
4 }
```

- Response: 400 Bad Request (12 ms, 283 B)
{"message": "Title tidak boleh kosong", "data": null}

Request 2 (Bottom):

- Method: PUT
- URL: `PUT {{baseURL}}/notes/fbd18eee-8190-455f-b954-f... (9)`
- Body (JSON):

```
1 {
2     "title": "Hari Kelima",
3     "content": ""
4 }
```

- Response: 400 Bad Request (8 ms, 285 B)
{"message": "Content tidak boleh kosong", "data": null}

Postman CLI:

```
1 postman request PUT
  'localhost:8760/api/
  notes/
  fbd18eee-8190-455f-b954-f
  e7f858032c2' \
2 --header 'Content-Type:
  application/json' \
3 --body '{
4   "title": "Hari Kelima",
5   "content": ""
6 }'
```

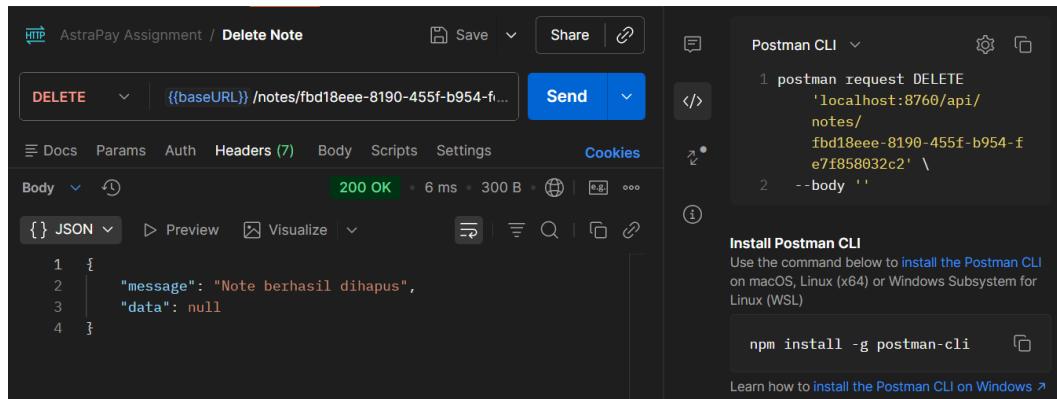
Install Postman CLI:
Use the command below to [install the Postman CLI](#) on macOS, Linux (x64) or Windows Subsystem for Linux (WSL)

```
npm install -g postman-cli
```

Learn how to [install the Postman CLI on Windows](#)

5. Delete Note

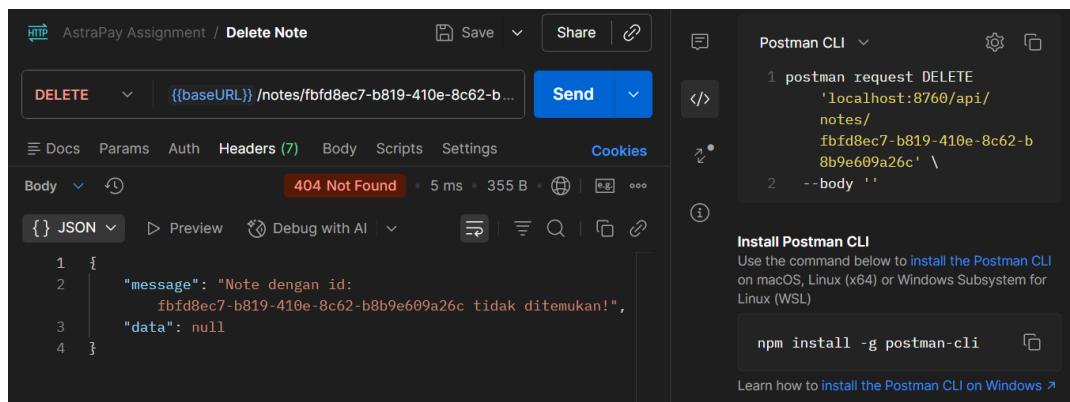
a. Success



A screenshot of the Postman application interface. The request URL is `DELETE {{baseURL}} /notes/fbd18eee-8190-455f-b954-f...` . The response status is `200 OK` with a response time of `6 ms` and a body size of `300 B`. The response body is JSON:

```
1 {
2   "message": "Note berhasil dihapus",
3   "data": null
4 }
```

b. Not Found



A screenshot of the Postman application interface. The request URL is `DELETE {{baseURL}} /notes/fbfd8ec7-b819-410e-8c62-b...` . The response status is `404 Not Found` with a response time of `5 ms` and a body size of `355 B`. The response body is JSON:

```
1 {
2   "message": "Note dengan id: fbfd8ec7-b819-410e-8c62-b8b9e609a26c tidak ditemukan!",
3   "data": null
4 }
```