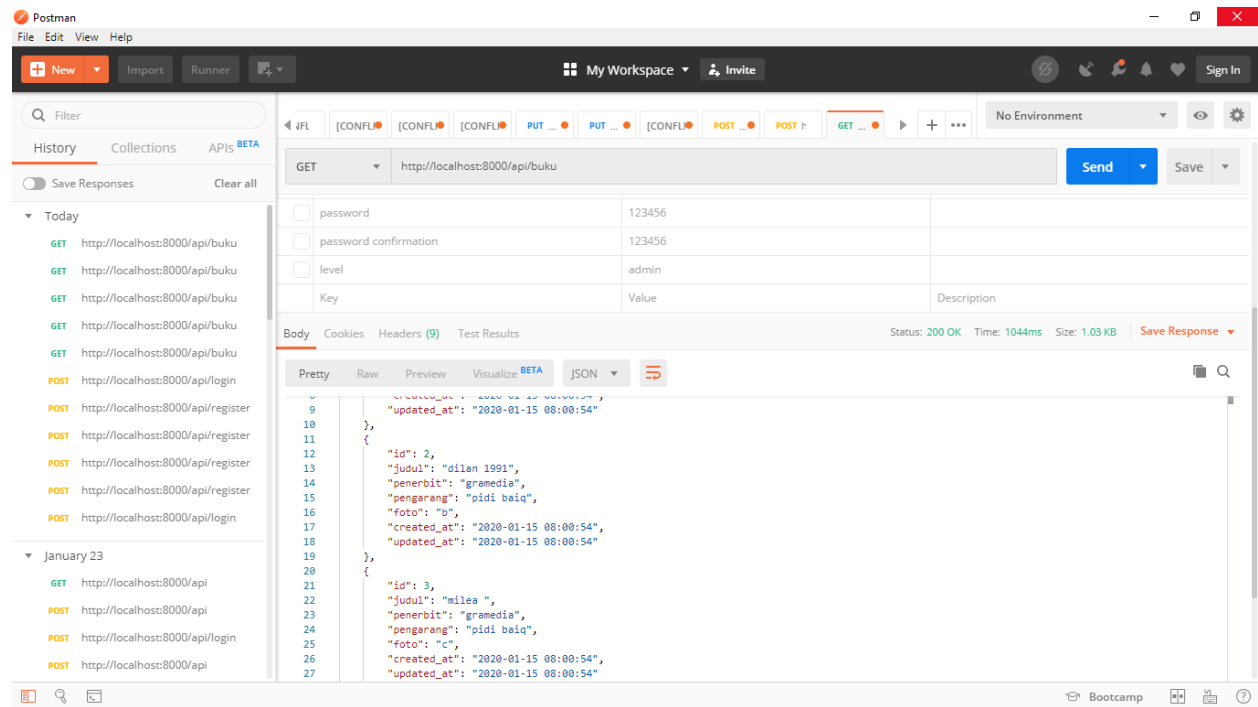


Read buku



Postman interface showing a GET request to `http://localhost:8000/api/buku`. The response is a JSON array of book objects.

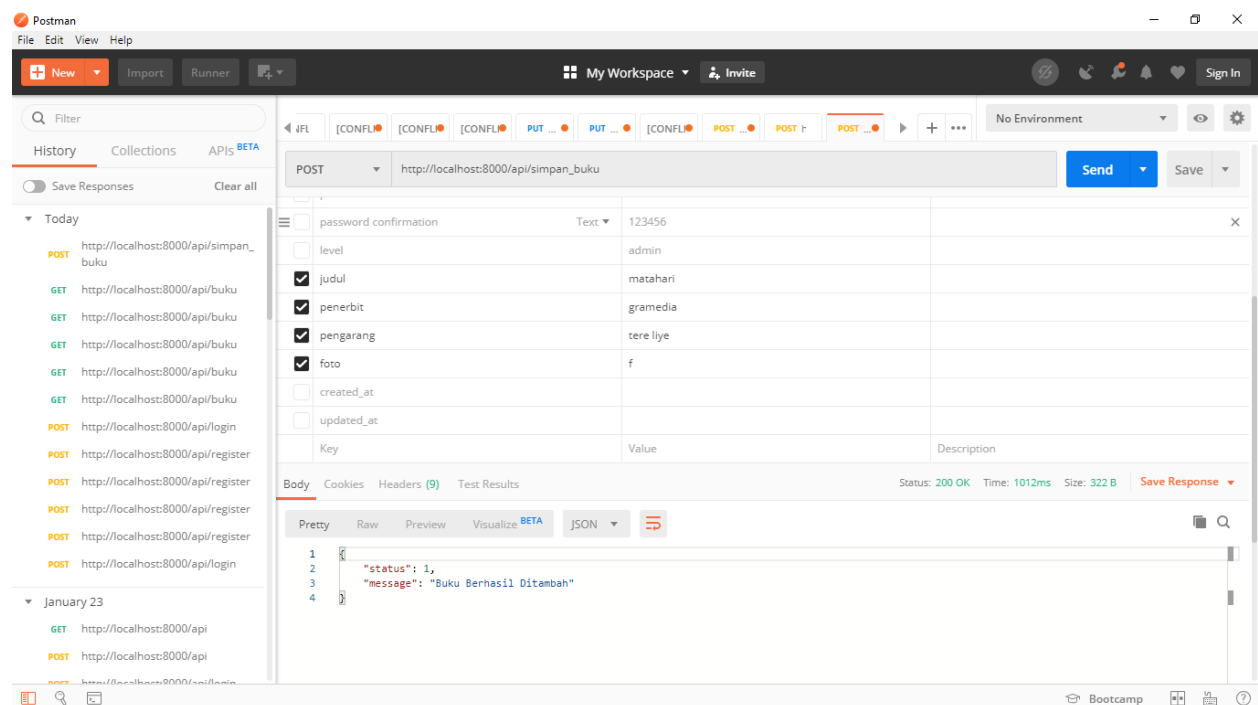
Request: GET `http://localhost:8000/api/buku`

Response: Status: 200 OK, Time: 1044ms, Size: 1.03 KB

Body (JSON):

```
[{"id": 2, "judul": "dilan 1991", "penerbit": "gramedia", "pengarang": "pidi baiq", "foto": "b", "created_at": "2020-01-15 08:00:54", "updated_at": "2020-01-15 08:00:54"}, {"id": 3, "judul": "milea ", "penerbit": "gramedia", "pengarang": "pidi baiq", "foto": "c", "created_at": "2020-01-15 08:00:54", "updated_at": "2020-01-15 08:00:54"}]
```

create buku



Postman interface showing a POST request to `http://localhost:8000/api/simpan_buku`. The response is a JSON object with status and message.

Request: POST `http://localhost:8000/api/simpan_buku`

Body (Form Data):

Key	Value
password confirmation	123456
level	admin
judul	matahari
penerbit	gramedia
pengarang	tere liye
foto	f
created_at	
updated_at	

Response: Status: 200 OK, Time: 1012ms, Size: 322 B

Body (JSON):

```
{"status": 1, "message": "Buku Berhasil Ditambah"}
```

Update buku

The screenshot shows the Postman application with a PUT request configured to update a book. The request is sent to `http://localhost:8000/api/ubah_buku/2`. The body is in `x-www-form-urlencoded` format with the following data:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> judul	dilan 2020	
<input checked="" type="checkbox"/> penerbit	yyyy	
<input checked="" type="checkbox"/> pengarang	pidi baiq	
<input checked="" type="checkbox"/> foto	b	
Key	Value	Description

The response is a 200 OK status with the following JSON body:

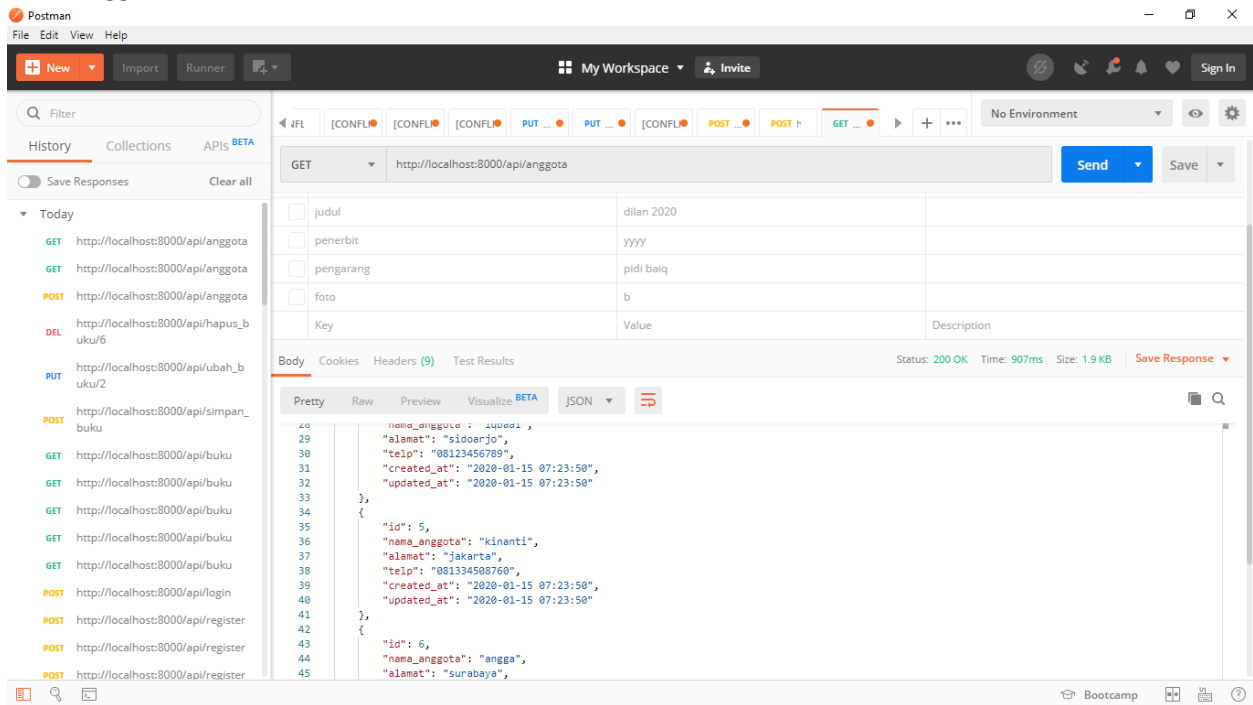
```
{  "status": 1,  "message": "Data Berhasil Diubah"}
```

Hapus buku

The screenshot shows the Postman application with a DELETE request configured to delete a book. The request is sent to `http://localhost:8000/api/hapus_buku/6`. The response is a 200 OK status with the following JSON body:

```
{  "status": 1,  "message": "Buku Berhasil Dihapus"}
```

Read anggota



Postman interface showing a GET request to `http://localhost:8000/api/anggota`. The response is a JSON array of two member objects.

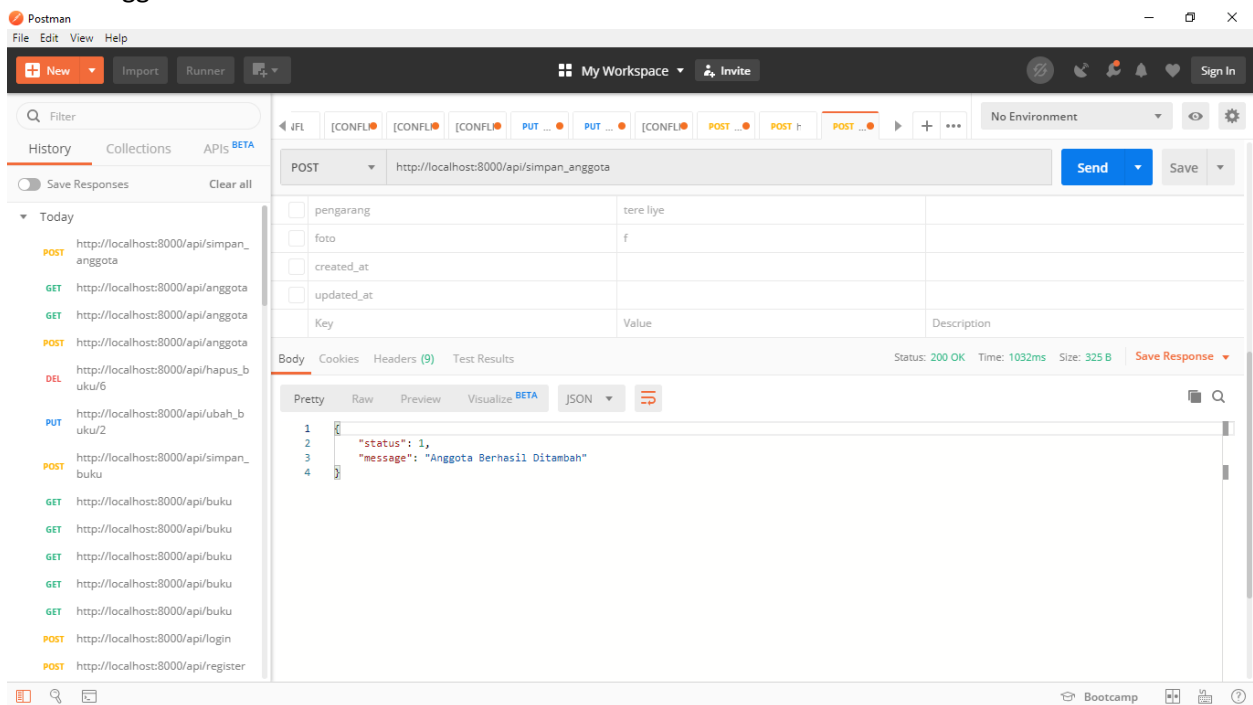
Request: GET `http://localhost:8000/api/anggota`

Response: Status: 200 OK, Time: 907ms, Size: 1.9 KB

Body (JSON):

```
[{"id": 4, "nama_anggota": "Luposa", "alamat": "Sidoarjo", "telp": "08123456789", "created_at": "2020-01-15 07:23:50", "updated_at": "2020-01-15 07:23:50"}, {"id": 5, "nama_anggota": "Kinanti", "alamat": "Jakarta", "telp": "081334508760", "created_at": "2020-01-15 07:23:50", "updated_at": "2020-01-15 07:23:50"}]
```

Create anggota



Postman interface showing a POST request to `http://localhost:8000/api/simpan_anggota`. The response is a JSON object indicating successful creation.

Request: POST `http://localhost:8000/api/simpan_anggota`

Response: Status: 200 OK, Time: 1032ms, Size: 325 B

Body (JSON):

```
{ "status": 1, "message": "Anggota Berhasil Ditambah" }
```

Update anggota

The screenshot shows the Postman application with a PUT request to `http://localhost:8000/api/ubah_anggota/3`. The request body is a JSON object with the following fields:

Key	Value	Description
<input type="checkbox"/> penerbit	yyyy	
<input type="checkbox"/> pengarang	pidi baiq	
<input type="checkbox"/> foto	b	
<input checked="" type="checkbox"/> nama_anggota	dimas	
<input checked="" type="checkbox"/> alamat	sda	
<input checked="" type="checkbox"/> telp	081334508760	

The response is a JSON object:

```
1 {
2   "status": 1,
3   "message": "Data Berhasil Diubah"
4 }
```

Status: 200 OK, Time: 1017ms, Size: 320 B

Hapus anggota

The screenshot shows the Postman application with a DELETE request to `http://localhost:8000/api/hapus_anggota/13`. The request body is a JSON object with the following fields:

Key	Value	Description
<input type="checkbox"/> penerbit	yyyy	
<input type="checkbox"/> pengarang	pidi baiq	
<input type="checkbox"/> foto	b	
<input checked="" type="checkbox"/> nama_anggota	dimas	
<input checked="" type="checkbox"/> alamat	sda	
<input checked="" type="checkbox"/> telp	081334508760	

The response is a JSON object:

```
1 {
2   "status": 1,
3   "message": "Data Berhasil Dihapus"
4 }
```

Status: 200 OK, Time: 941ms, Size: 321 B

anggota controller

```
<?php
```

```
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\anggotaModel;
use Illuminate\Support\Facades\Validator;

class Anggota extends Controller
{
    public function index(){
        $dt_ag=anggotaModel::get();
        return response()->json($dt_ag);
    }

    public function store(Request $request){
        $validator=Validator::make($request->all(),
            [
                'nama_anggota'=>'required',
                'alamat'=>'required',
                'telp'=>'required'
            ]
        );

        if($validator->fails()){
            return Response()->json($validator->errors());
        }

        $simpan=anggotaModel::create([
            'nama_anggota'=>$request->nama_anggota,
            'alamat'=>$request->alamat,
            'telp'=>$request->telp
        ]);
    }
}
```

```

        $status=1;
        $message="Anggota Berhasil Ditambah";
        if($simpan){
            return Response()-
>json(compact('status','message'));
        }else {
            return Response()->json(['status'=>0]);
        }
    }

    public function update($id,Request $request){
        $validator=Validator::make($request->all(),
            [
                'nama_anggota'=>'required',
                'alamat'=>'required',
                'telp'=>'required'
            ]
        );

        if($validator->fails()){
            return Response()->json($validator->errors());
        }

        $ubah=anggotaModel::where('id',$id)->update([
            'nama_anggota'=>$request->nama_anggota,
            'alamat'=>$request->alamat,
            'telp'=>$request->telp
        ]);

        $status=1;
        $message="Data Berhasil Diubah";
    }
}

```

```

        if($ubah){
            return Response()-
>json(compact('status','message'));
        } else {
            return Response()->json(['status'=>0]);
        }
    }

    public function tampil(){
        $data_anggota=anggotaModel::get();
        $count=$data_anggota->count();
        $arr_data=array();
        foreach ($data_anggota as $dt_ag){
            $arr_data[]=array(
                'id' => $dt_ag->id,
                'nama_anggota' => $dt_ag->nama_anggota,
                'alamat' => $dt_ag->alamat,
                'telp' => $dt_ag->telp
            );
        }
        $status=1;
        return Response()-
>json(compact('status','count','arr_data'));
    }

    public function destroy($id){
        $hapus=anggotaModel::where('id',$id)->delete();
        $status=1;
        $message="Data Berhasil Dihapus";
        if($hapus){
            return Response()-
>json(compact('status','message'));

```

```

        }else {
            return Response()->json(['status'=>0]);
        }
    }
}

```

Anggota model

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class anggotaModel extends Model
{
    protected $table="anggota";
    protected $primaryKey="id";
    protected $fillable=[
        "nama_anggota","alamat","telp"
    ];
}

```

Buku controller

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\bukuModel;

```



```
use Illuminate\Support\Facades\Validator;

class Buku extends Controller
{
    public function index(){

        $dt_bk=bukuModel::get();
        return response()->json($dt_bk);

    }

    public function store(Request $req){
        $validator = Validator::make($req->all(),
        [
            'judul' => 'required',
            'penerbit' => 'required',
            'pengarang' => 'required',
            'foto' => 'required',
        ]);
        if($validator->fails()){
            return Response()->json($validator->errors());
        }

        $simpan = bukuModel::create([
            'judul' => $req->judul,
            'penerbit' => $req->penerbit,
            'pengarang' => $req->pengarang,
            'foto' => $req->foto,
        ]);
        $status=1;
        $message="Buku Berhasil Ditambah";
        if($simpan){
```

```

        return Response()-
>json(compact('status','message'));
    } else {
        return Response()->json(['status' => 0]);
    }
}

public function update($id, Request $req){
    $validator = Validator::make($req->all(),
    [
        'judul' => 'required',
        'penerbit' => 'required',
        'pengarang' => 'required',
        'foto' => 'required',
    ]);
    if($validator->fails()){
        return Response()->json($validator->errors());
    }

    $ubah = bukuModel::where('id', $id)->update([
        'judul' => $req->judul,
        'penerbit' => $req->penerbit,
        'pengarang' => $req->pengarang,
        'foto' => $req->foto,
    ]);
    $status=1;
    $message="Data Berhasil Diubah";
    if($ubah){
        return Response()-
>json(compact('status','message'));
    } else {
        return Response()->json(['status' => 0]);
    }
}

```

```

    }
}

public function destroy($id){
    $hapus = bukuModel::where('id', $id)->delete();
    $status=1;
    $message="Buku Berhasil Dihapus";
    if($hapus){
        return Response()-
>json(compact('status','message'));
    } else {
        return Response()->json(['status' => 0]);
    }
}

public function tampil(){
    $data_buku=bukuModel::get();
    $count=$data_buku->count();
    $arr_data=array();
    foreach ($data_buku as $dt_bk){
        $arr_data[]=array(
            'id' => $dt_bk->id,
            'judul' => $dt_bk->judul,
            'penerbit' => $dt_bk->penerbit,
            'pengarang' => $dt_bk->pengarang,
            'foto' => $dt_bk->foto
        );
    }
    $status=1;
    return Response()-
>json(compact('status','count','arr_data'));
}

```

```
}
```

Buku model

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class bukuModel extends Model
{
    protected $table="buku";
    protected $primaryKey="id";
    protected $fillable=[
        "judul","penerbit","pengarang","foto"
    ];
}
```

Api.php

```
// buku
Route::get('/buku','Buku@index');
Route::post('/simpan_buku','Buku@store');
Route::put('/ubah_buku/{id}','Buku@update');
Route::get('/tampil_buku','Buku@tampil');
Route::delete('/hapus_buku/{id}','Buku@destroy');

// anggota
Route::get('/anggota','Anggota@index');
Route::post('/simpan_anggota','Anggota@store');
```

```
Route::put('/ubah_anggota/{id}','Anggota@update');  
Route::get('/tampil_anggota','Anggota@tampil');  
Route::delete('/hapus_anggota/{id}','Anggota@destroy');
```