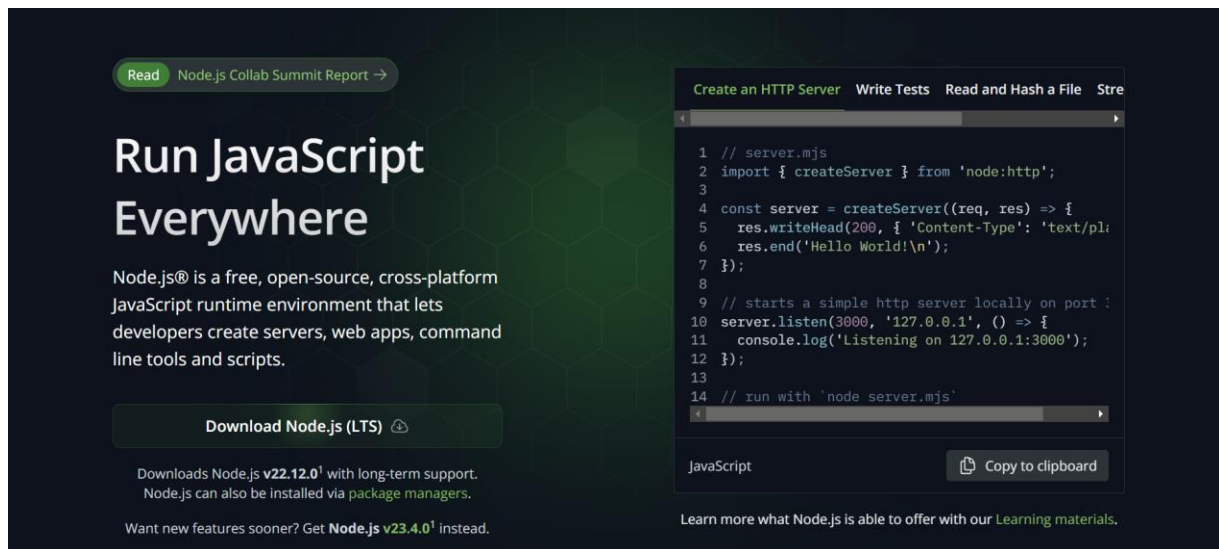
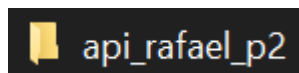


LANGKAH LANGKAH MEMBUAT API NARUTO

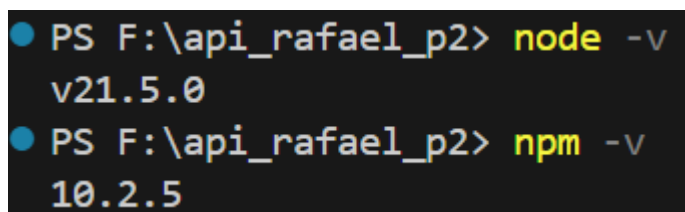
1. Mendownload node js di google



2. Membuat Folder API



3. Mengecek apakah node & npm sudah terinstall dengan menggunakan node -v & npm -v

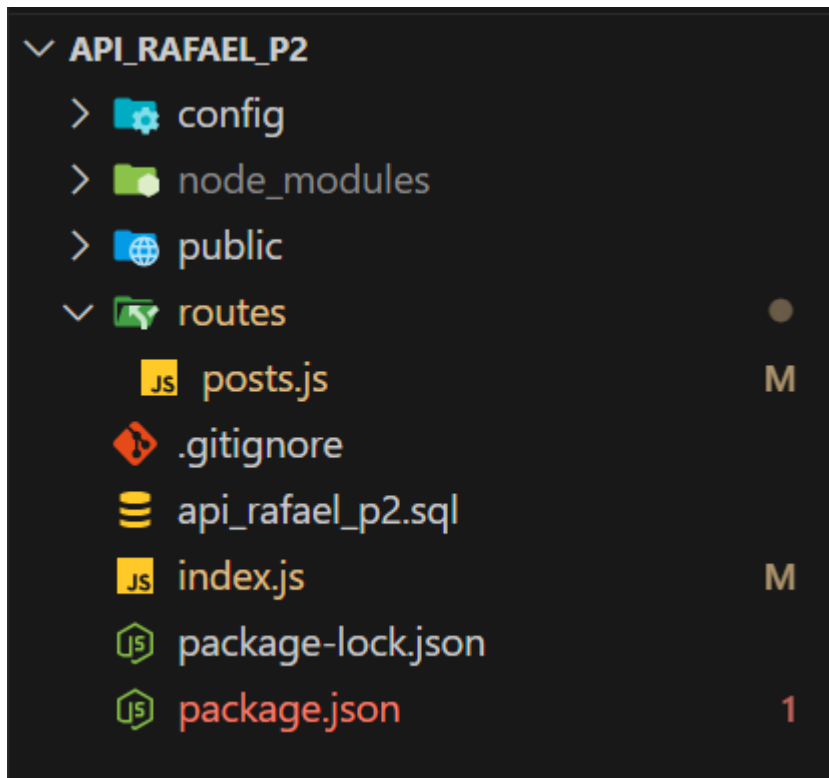


4. Membuat database dan table

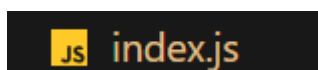
(LANGKAH – LANGKAH MEMBUAT DATABASE & TABLE AKAN DIJELASKAN DIBAWAH)

5. membuka folder api yang sudah dibuat sebelumnya di vs code

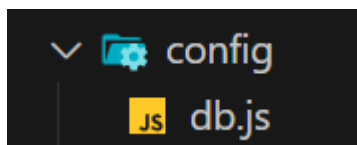
6. mendownload package.json yang berisi body-parser,express,express-validator,mysql,nodemon dari terminal vscode



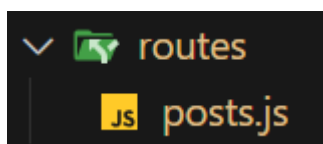
7. selanjutnya membuat file index.js



8. Membuat folder yang Bernama config yang berisi file database.js



9. membuat folder yang Bernama routes yang berisi file posts.js



10. mengcode file index.js untuk menampilkan halaman awal

```
index.js M X
index.js > ...
You, 43 seconds ago | 1 author (You)
1  const express = require('express');
2  const app = express();
3  const port = 3000;
4  const bodyParser = require('body-parser');
5  const path = require('path');
6
7  app.use(express.static(path.join(__dirname, 'public')));
8  app.use(bodyParser.urlencoded({ extended: false }));
9  app.use(bodyParser.json());
10 const postsRouter = require('./routes/posts');
11 app.use('/api/posts', postsRouter);
12
13 app.get('/', (req, res) => {
14   const filePath = path.join(__dirname, 'public', 'design.html');
15   console.log('Berhasil Update: ${filePath}');
16   res.sendFile(filePath, (err) => {
17     if (err) {
18       console.error('Error sending file:', err);
19       res.status(err.status).end();
20     }
21   });
22 });
23
24 app.listen(port, () => {
25   console.log('This application is running at http://localhost:${port}');
26 });
```

11. Mengcode file database.js untuk menyambungkan dengan mysql

```
db.js X
config > db.js > ...
You, last week | 1 author (You)
1  let mysql = require('mysql');
2
3  let connect = mysql.createConnection({
4    host: 'localhost',
5    user: 'root',
6    password: '',
7    database: 'api_rafael_p2'
8  });
9
10 connect.connect(function(error){
11   if(!error){
12     console.log(error)
13   }else{
14     console.log('Connection Success')
15   }
16 })
17
18 module.exports = connect;
```

12. mengcode file posts.js untuk menampilkan data,menambahkan data,mengupdate data,dan menghapus data

Query untuk Menambahkan data :

```
// Query database untuk menambahkan data baru
connect.query('INSERT INTO posts SET?', formData, function (err, rows) {
  if (err) {
    // Jika terjadi error, kirim status 500 dan pesan error
    return res.status(500).json({
      status: false,
      message: 'Error, Try again later'
    })
  } else {
    // Jika data berhasil disimpan, kirim status 201 dan pesan sukses
    return res.status(201).json({
      status: true,
      message: 'Data input Successfully',
    })
  }
})
})
```

Query untuk Menampilkan Data:

```
// Query database untuk menampilkan data berdasarkan ID
connect.query(`SELECT * FROM posts WHERE ID=${id}`,
  function (error, rows) {
    if (error) {
      // Jika terjadi error, kirim status 500 dan pesan error
      return res.status(500).json({
        status: false,
        message: 'Server Error!!'
      })
    }
    // Jika data tidak ditemukan, kirim status 404 dan pesan error
    if (rows.length <= 0) {
      return res.status(404).json({
        status: false,
        message: 'Data is missing'
      })
    } else {
      // Jika data berhasil diambil, kirim status 200 dan data
      return res.status(200).json({
        status: true,
        message: 'Display data according to id',
        data: rows[0]
      })
    }
  }
)
})
```

Query untuk Mengupdate Data:

```
//update query
connect.query(`UPDATE posts set ? WHERE id=${id}`, formData,
  function(err,rows){
    if(err){
      return res.status(500).json({
        status: false,
        message: 'Interval server error'
      })
    }else{
      return res.status(200).json({
        status: true,
        message: 'Update data success'
      })
    }
  }
)
})
```

Query untuk Menghapus Data :

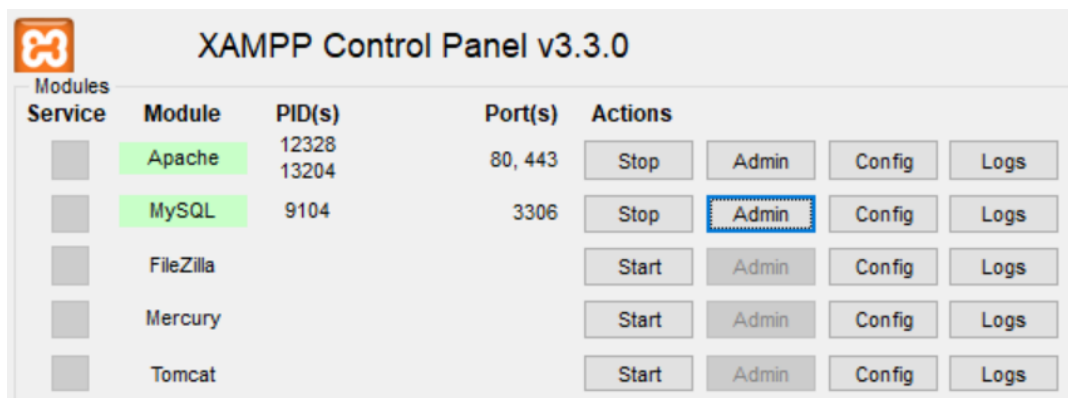
```
//Query database untuk megapus data berdasarkan id
router.delete('/delete/(id)',
  function(req,res){
    let id = req.params.id
    connect.query(`DELETE FROM posts WHERE id = ${id}`,
      function(error,rows){
        if(error){
          return res.status(500).json({
            status:false,
            message: 'Server Error!!'
          })
        }else {
          return res.status(200).json({
            status: true,
            message: 'Data has been deleted',
          })
        }
      }
    )
  }
)
})
```

13. jangan lupa membuat array sesuai table yang telah dibuat di database

```
router.post('/store', [
  // Validasi data input
  body('nama').notEmpty(),
  body('gender').notEmpty(),
  body('sword').notEmpty(),
  body('village').notEmpty(),
  body('jutsu').notEmpty(),
  body('abilities').notEmpty(),
  body('team').notEmpty(),
  body('description').notEmpty(),
], (req, res) => {
  // Cek validasi data
  const errors = validationResult(req)
  if (!errors.isEmpty()) {
    // Jika terjadi error validasi, kirim status 422 dan pesan error
    return res.status(422).json({
      errors: errors.array()
    })
  }

  // Definisikan data yang akan disimpan
  let formData = {
    nama: req.body.nama,
    gender: req.body.gender,
    sword: req.body.sword,
    village: req.body.village,
    jutsu: req.body.jutsu,
    abilities: req.body.abilities,
    team: req.body.team,
    description: req.body.description,
  }
})
```

14. Menyalakan Xampp dengan kondisi apache & MySql Menyala



15. Melakukan Run pada terminal dengan kode nodemon npm start

```
PS F:\api_rafael_p2> nodemon npm start
[nodemon] 3.1.7
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node npm start index.js`
This application is running at http://localhost:3000
Connection Success
```

14. selanjutnya membuka aplikasi postman

15. mencoba menampilkan,menambah,mengupdate,dan menghapus data di postman

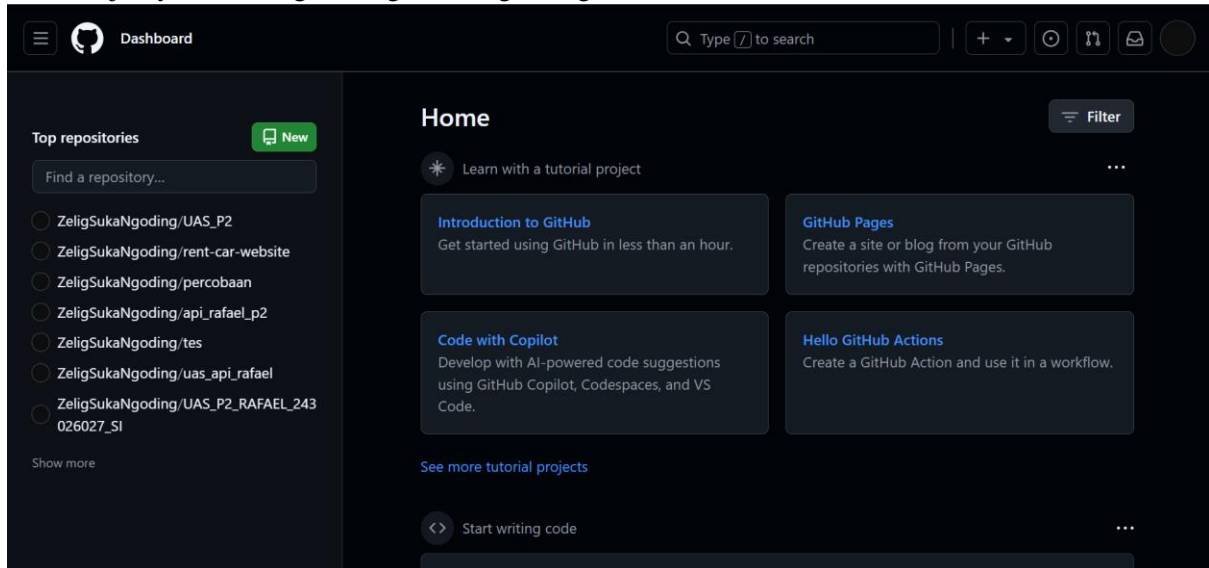
The screenshot shows the Postman interface with a GET request to `http://localhost:3000/api/posts`. The 'Body' tab is selected, and the 'x-www-form-urlencoded' format is chosen. The body contains the following data:

Key	Value
nama	Mangetsu Hōzuki
gender	Male
sword	Hiramekarei
village	Kirigakure
jutsu	Hydrification Technique
abilities	Water Release
team	Seven Ninja Swordsmen of the Mist
description	Mangetsu Hōzuki, reputed as the Second Coming of the De...

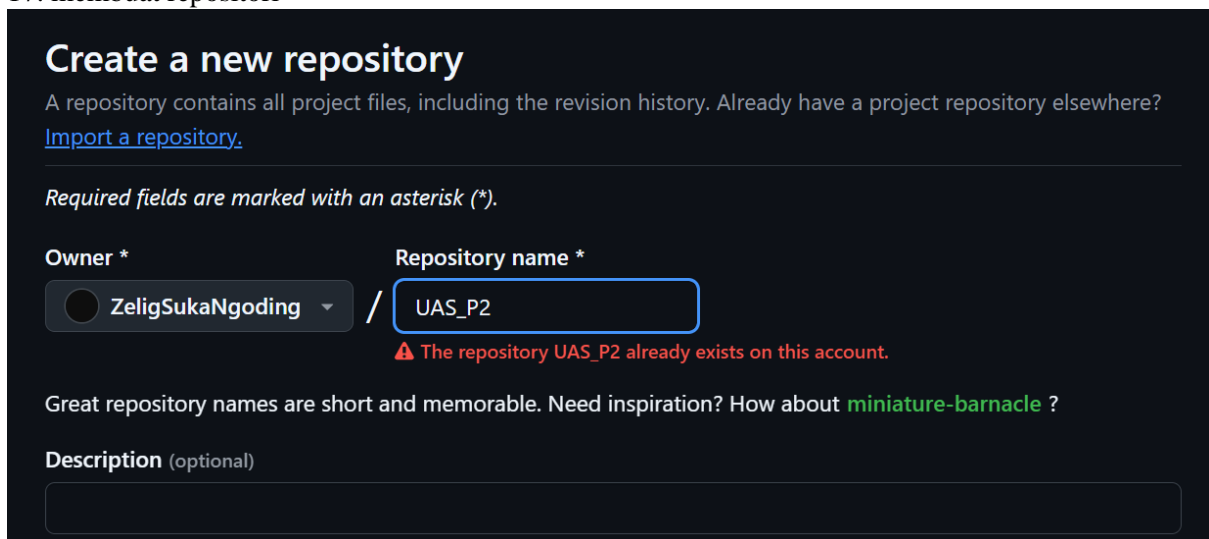
The screenshot shows the Postman interface with the same GET request. The 'Body' tab is selected, and the response is displayed in JSON format. The response status is 200 OK, with a response time of 55 ms and a size of 2.92 KB. The JSON response is as follows:

```
{
  "id": 12,
  "nama": "Mangetsu Hōzuki",
  "gender": "Male",
  "sword": "Hiramekarei",
  "village": "Kirigakure",
  "jutsu": "Hydrification Technique",
  "abilities": "Water Release",
  "team": "Seven Ninja Swordsmen of the Mist",
  "description": "Mangetsu Hōzuki, reputed as the Second Coming of the Demon was a shinobi from Kirigakure's Hōzuki clan, and a member of the previous generation of the Seven Ninja Swordsmen of the Mist."
}
```

16. selanjutnya kita menginstall git dan login ke github



17. membuat repository





18. selanjutnya kita melakukan commit lalu melakukan push file ke github



Langkah – Langkah Membuat Database api_rafael_p2

Disini saya membuat database melalui phpMyAdmin. Berikut Step By Step proses pembuatan database dibawah:

Membuat Database :

 **Create database** 

Membuat Table & Column :


 **Create new table**

Table name

Number of columns

Membuat Nama,Type,Length,Default,Collation,Atributes.Null,A_I pada setaip column :

Name	Type	Length/Values	Default	Collation	Attributes	Null	A_I
id	INT	11	None			<input type="checkbox"/>	<input checked="" type="checkbox"/>
nama	VARCHAR	255	NULL	utf8mb4_general_		<input checked="" type="checkbox"/>	<input type="checkbox"/>
gender	VARCHAR	255	NULL	utf8mb4_general_		<input checked="" type="checkbox"/>	<input type="checkbox"/>
sword	VARCHAR	255	NULL	utf8mb4_general_		<input checked="" type="checkbox"/>	<input type="checkbox"/>
village	VARCHAR	255	NULL	utf8mb4_general_		<input checked="" type="checkbox"/>	<input type="checkbox"/>
jutsu	VARCHAR	255	NULL	utf8mb4_general_		<input checked="" type="checkbox"/>	<input type="checkbox"/>
abilities	VARCHAR	255	NULL	utf8mb4_general_		<input checked="" type="checkbox"/>	<input type="checkbox"/>
team	VARCHAR	255	NULL	utf8mb4_general_		<input checked="" type="checkbox"/>	<input type="checkbox"/>
description	VARCHAR	255	NULL	utf8mb4_general_		<input checked="" type="checkbox"/>	<input type="checkbox"/>

Jika Sudah Maka tampilannya akan seperti ini:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	nama	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 3	gender	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 4	sword	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 5	village	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 6	jutsu	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 7	abilities	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 8	team	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 9	description	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Jika sudah maka coba mmenginput data lewat postman , jika sudah terinput maka akan muncul tampilan seperti ini :

		id	nama	gender	sword	village	jutsu	abilities	team	description
<input type="checkbox"/>	Edit Copy Delete	6	Jinin Akebino	Male	Kabutowari.	Kirigakure	Water Style	Silent Killing	Seven Ninja Swordsmen of the Mist	Jinin Akebino was a jōnin from Kirigakure, and a m...
<input type="checkbox"/>	Edit Copy Delete	7	Jūzō Biwa	Male	Kubikiribōchō.	Kirigakure	Water Style	Water Clone	Seven Ninja Swordsmen of the Mist	Jūzō Biwa was a jōnin from Kirigakure and a member...
<input type="checkbox"/>	Edit Copy Delete	8	Kisame Hoshigaki	Male	Samehada	Kirigakure	Water Release	Water Prison Shark Dance Technique	Seven Ninja Swordsmen of the Mist	Feared as the Monster of the Hidden Mist, was a sh...
<input type="checkbox"/>	Edit Copy Delete	9	Kushimaru Kuriarare	Male	Nuibari	Kirigakure	Longsword Ninja Art — Earth Spider Sewing	Silent Killing	Seven Ninja Swordsmen of the Mist	Kushimaru Kuriarare was a Kirigakure jōnin and a m...
<input type="checkbox"/>	Edit Copy Delete	10	Raiga Kurosuki	Male	Kiba, Twin Swords	Kirigakure	Lightning Release	Lightning Burial: Banquet of Lightning	Seven Ninja Swordsmen of the Mist	Raiga Kurosuki renowned as the Thunder of the Hidd...
<input type="checkbox"/>	Edit Copy Delete	11	Jinpachi Munashi	Male	Shibuki	Kirigakure	Blastsword Technique: Blasting Bridle Repeating De...	Explosive Tag	Seven Ninja Swordsmen of the Mist	Jinpachi Munashi was a jōnin from Kirigakure and a...