

# UAS Basis Data

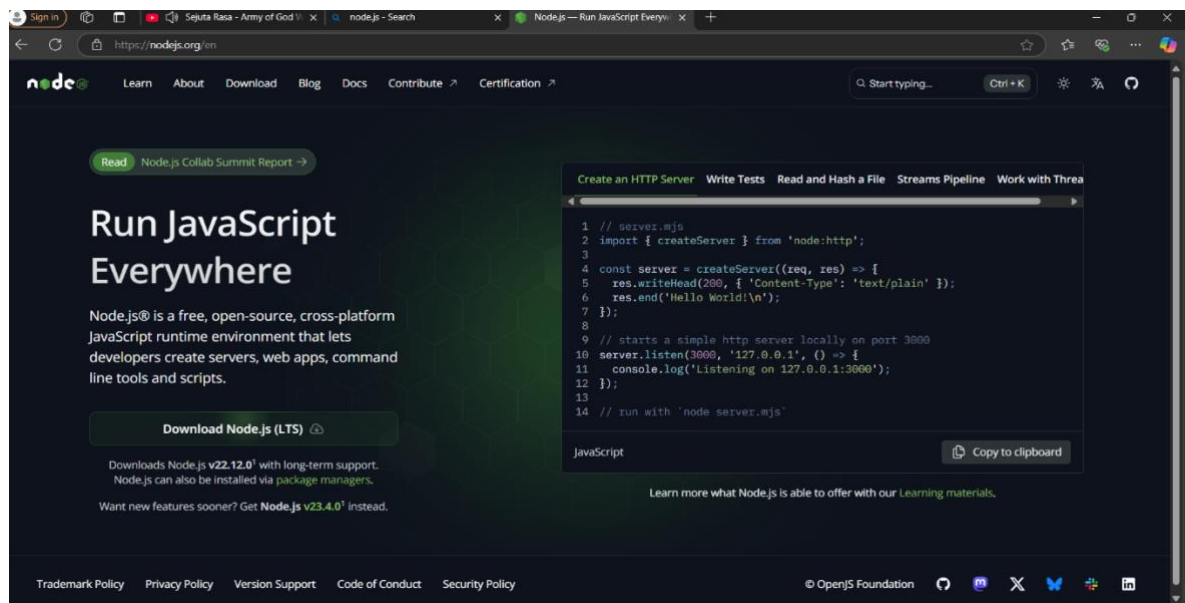
Nama : Kimberly Putri Natalie

Nim : 243026012

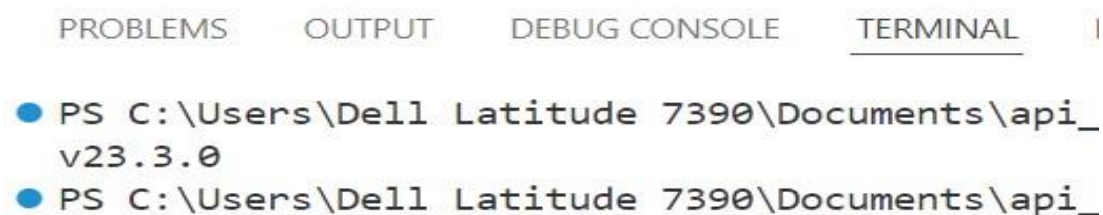
Prodi : S1 Sistem Informasi

## LANGKAH – LANGKAH MEMBUAT API INFORMASI FILM.

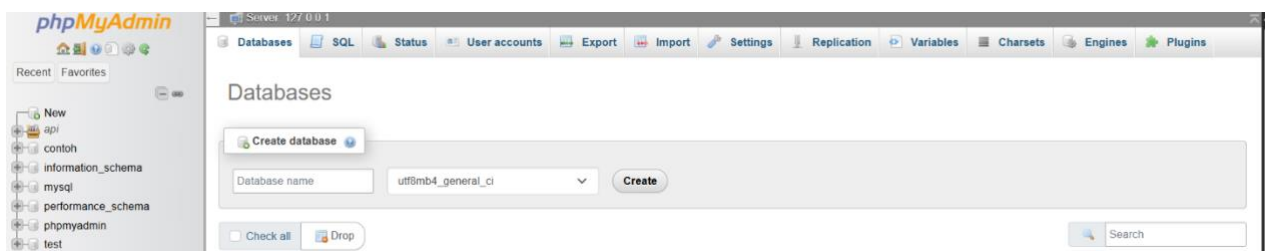
1. Pertama, perlu mengunduh aplikasi node.js melalui situs resmi yaitu Node.js (<https://nodejs.org/>)
2. Lalu menjalankan installer dan mengikuti petunjuk yang ada pada layar device.



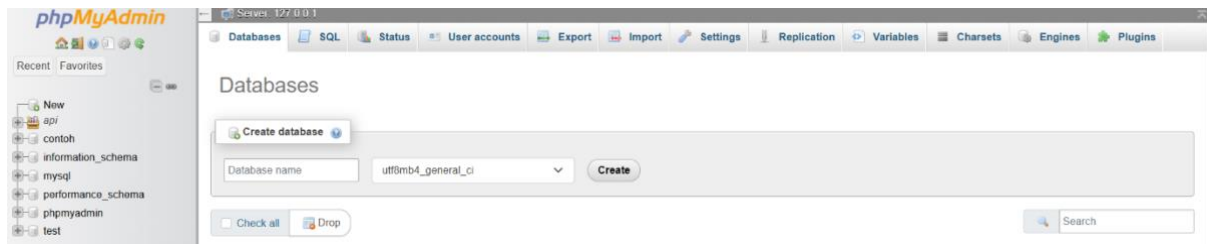
3. Lalu kita membuat folder API untuk proyek yang telah kita pilih untuk buat.
4. Sebelum melanjutkan, kita bisa cek terlebih dahulu apakah node & npm sudah terinstall dengan menggunakan node -v & npm -v di terminal



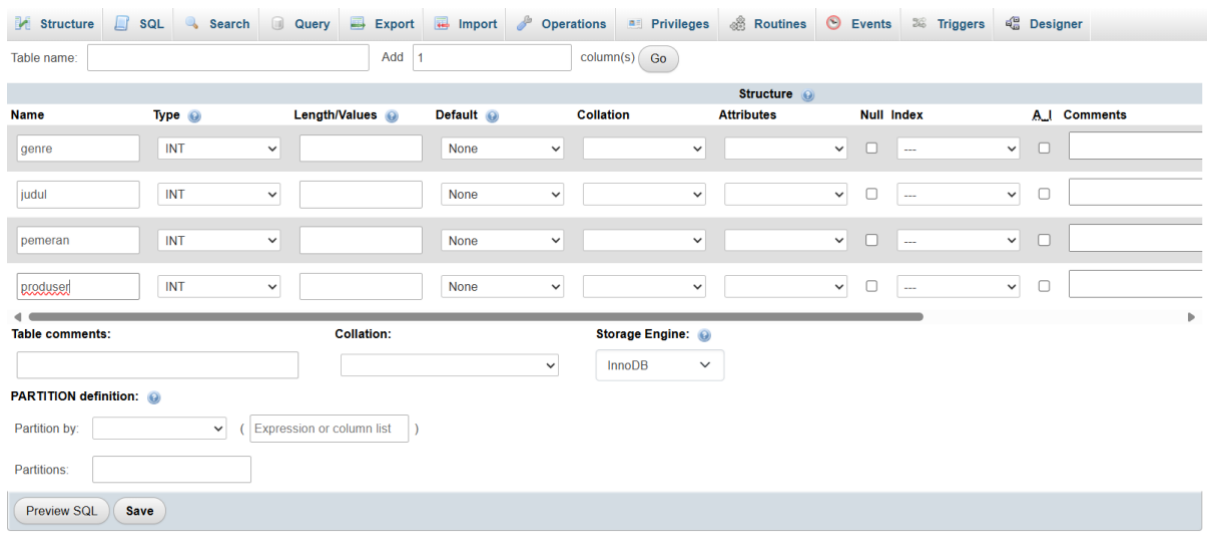
5. Jika sudah, buatlah database dan table pada phpMyAdmin. Disini saya membuat database dengan judul api\_natalie



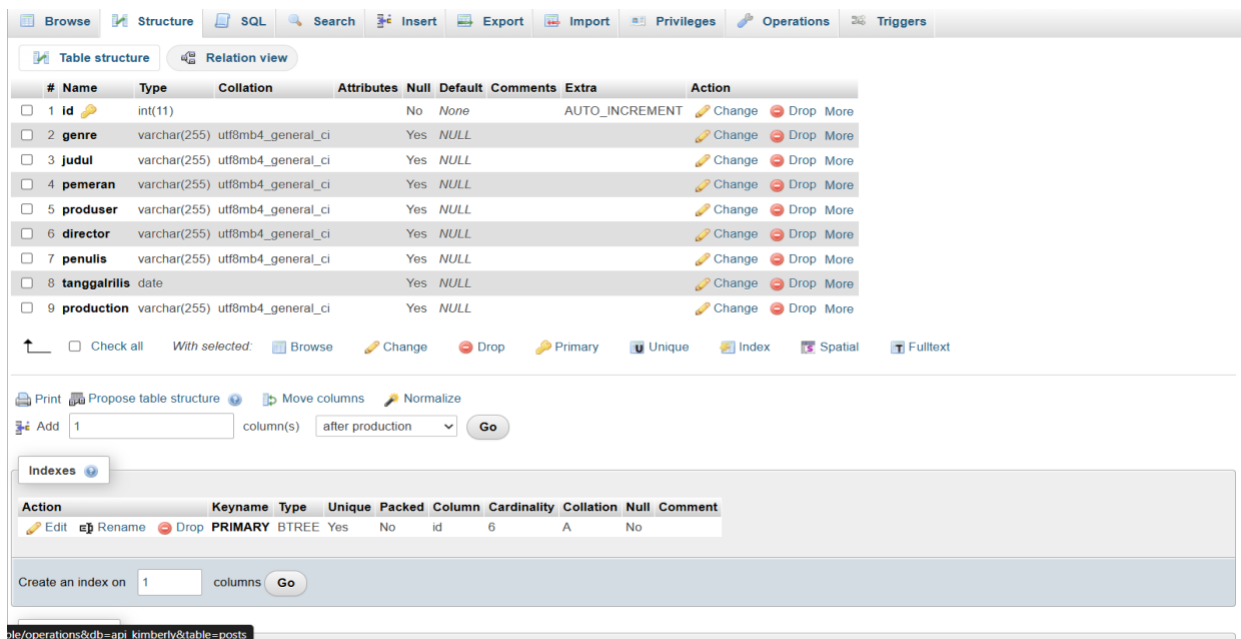
Lalu buatlah table & kolom



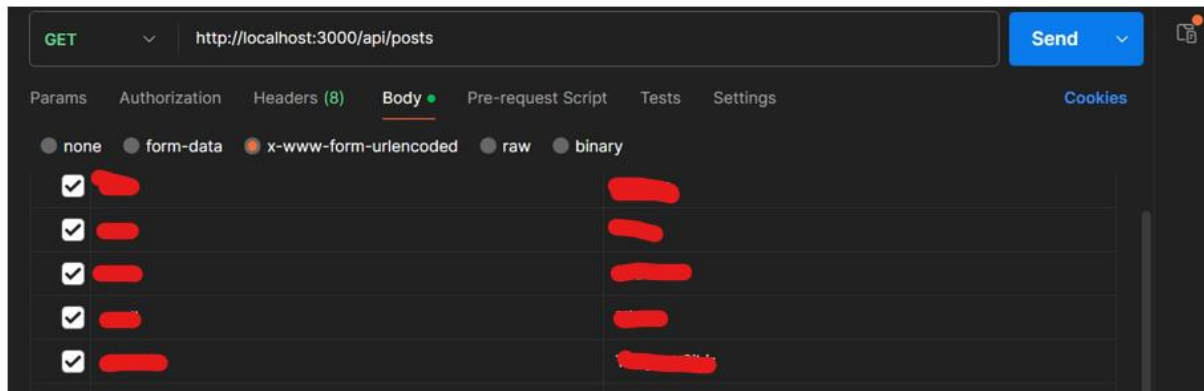
Kemudian isilah Name, Type, Length/Values, Default, Null, dan A\_I di setiap kolom



Jika sudah, maka tampilannya akan seperti ini :



Lalu inputlah data melalui Postman satu persatu

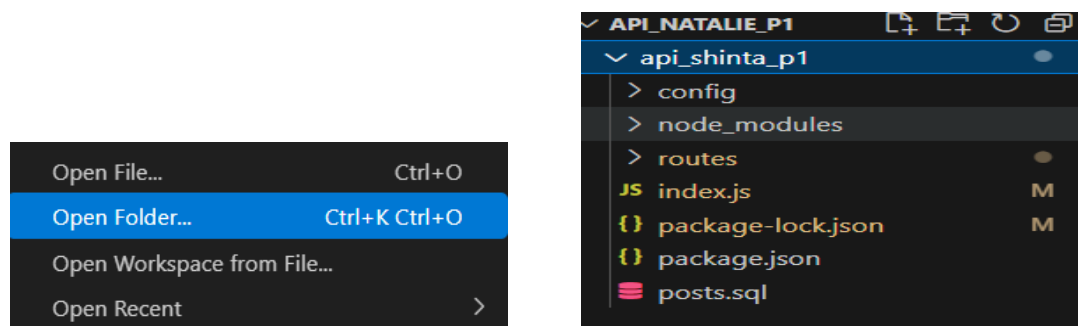


Setelah selesai menginput dan terinput, maka akan muncul tampilan seperti ini di phpMyAdmin

The screenshot shows the phpMyAdmin interface for a database. The 'posts' table is selected, showing 7 rows of data. The table structure is as follows:

	id	genre	judul	pemeran	produser	director	penulis	tanggairilis	production
<input type="checkbox"/>	1	Animasi Musikal	Moana	Auli Cravalho, Dwayne Jhonson, David Fane, Gerald...	Christina Chen, Yvett Merino	David G, Derrick Jr.	Jared Bush, Dana Ledoux Miller	0000-00-00	Walt Disney
<input type="checkbox"/>	2	Fantasy Musikal	Wicked	Ariana Grande, Cynthia Erivo, Jonathan Bailey, Mar...	Marc Platt, David Stone	Jon M, Chu	Winnie Holzman, Dana Fox	0000-00-00	Universal Pictures
<input type="checkbox"/>	3	Animasi	The Lord Of The Rings: The War Of The Rohirrim	Brian Cox, Gaia Wise, Luke Pasqualino, Miranda Ott...	Philippa Boyens, Jason DeMarco, Joseph Chou	Kenji Kamiyama	Jeffrey Addiss, Will Matthews, Phoebe Gittins, Arty...	0000-00-00	Warner Bros Pictures
<input type="checkbox"/>	4	Action Adventures Sci-Fi	Krave The Hunter	Aaron Taylor, Russell Crowe, Fred Hechinger, Arian...	Avi Arad, David B Householter, Matt Tolmach	J.C Chandor	Richard Wenk, Art Marcum, Matt Holloway	0000-00-00	Columbia Pictures
<input type="checkbox"/>	5	Comedy	Konco Konco Edan	Sahila Hisyam, Dimas Kusnadi, Sepriadi, Prit Timo...	Ismail Taufik	Odi Harahap	Erwin Wu	0000-00-00	MPS Pictures
<input type="checkbox"/>	6	Drama Keluarga	Bila Esok Ibu Tiada	Christine Hakim, Slamet Rahardjo, Adinia Wirasti...	Agung Saputro, Nunu Datau	Rudi Sudjarwo	Oka Aurora	0000-00-00	Leo Pictures
<input type="checkbox"/>	7	Drama Romantis	Sampai Nanti, Hannal	Febby Rastanty, Juan Bio One, Ibrahim Rasyid, Anja...	Dewi Umayra Rachman	Agung Sentausa	Swastika Nohara	0000-00-00	Pic(k)lock Films, Azoo Projects, Fortius Corporati...

6. Setelah itu bukalah folder API yang sudah dibuat sebelumnya di VSCode



7. Jika sudah, kita harus mengunduh terlebih dahulu package.json yang berisi body-parser, express, expressvalidator, mysql, nodemon dari terminal vscode agar coding tidak terjadi error

The screenshot shows the VS Code interface. On the left, the Explorer sidebar shows a project named 'API\_NATALIE\_P1' with a subfolder 'api\_shinta\_p1'. Inside this folder, there are files: 'config', 'node\_modules', 'routes', 'index.js', 'package-lock.json', 'package.json', and 'posts.sql'. The main editor area shows the 'package-lock.json' file. The content of the file is as follows:

```
1 {
2   "name": "api_natalie_p1",
3   "version": "1.0.0",
4   "lockfileVersion": 3,
5   "requires": true,
6   "packages": {
7     "": {
8       "name": "api_natalie_p1",
9       "version": "1.0.0",
10      "license": "ISC",
11      "dependencies": {
12        "api_natalie_p1": "file:",
13        "body-parser": "^1.20.3",
14        "express": "^4.21.2",
15        "express-validator": "^7.2.0",
16        "mysql": "^2.18.1"
```

8. Selanjutnya buatlah file index.js di VSCode

The screenshot shows the VS Code Explorer sidebar with the file 'index.js' selected under the 'api\_shinta\_p1' folder.

9. Pada tahap selanjutnya, membuat folder yang bernama config yang berisi file database.js

The screenshot shows the VS Code Explorer sidebar with a new folder named 'config' created. Inside the 'config' folder, there is a file named 'database.js'.

10. Kemudian buatlah folder dan berilah nama routes yang berisi file posts.js

The screenshot shows the VS Code Explorer sidebar with a new folder named 'routes' created. Inside the 'routes' folder, there is a file named 'posts.js'.

11. Ketikkan code file index.js untuk menampilkan halaman awal

The screenshot shows the VS Code editor with the 'index.js' file open. The code is as follows:

```
1 const express = require('express');
2 const app = express();
3 const port = 3000;
4 const bodyparser = require('body-parser');
5
6
7 app.get('/', (req, res) => {
8   res.send('hallo natalie');
9 });
10
11 app.use(bodyparser.urlencoded({extended:false}));
12 app.use(bodyparser.json());
13
14 // import route posts
15 const postsRouter = require('./routes/posts');
16 app.use('/api/posts', postsRouter);
17
18 // ini untuk akses halaman website
19 app.listen(port, () => {
20   console.log(`this application running in http://localhost:${port}`);
21 });
```

12. Ketikkan code file database.js untuk menyambungkan dengan mysql

```
JS posts.js M JS index.js M JS database.js X {} package-lock.json M
api_shinta_p1 > config > JS database.js > ...
You, 5 days ago | 2 authors (yashinta06 and one other)
1 let mysql = require('mysql') 271.9k (gzipped: 88.7k)
2
3 let connect = mysql.createConnection({
4   host: 'localhost',
5   user: 'root',
6   password: '',
7   database: 'api_kimberly'
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;
```

13. Ketikkan code file posts.js untuk menampilkan data, menambahkan data, mengupdate data, dan menghapus data :

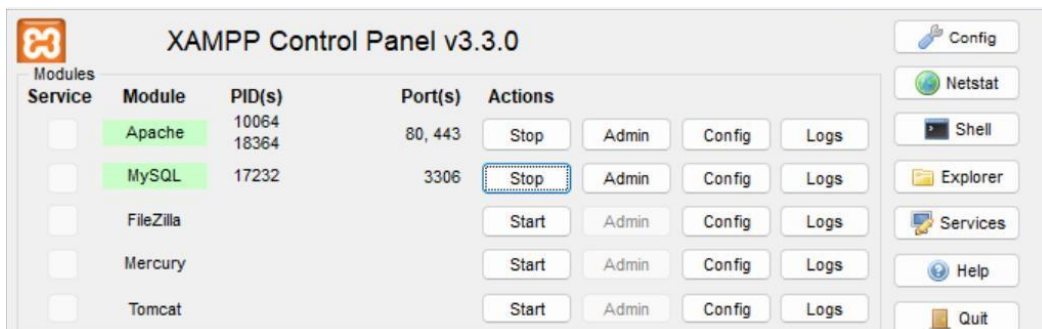
```
50 let formData = {
51 }
52
53 //memasukan query
54 connect.query('INSERT INTO posts SET?', formData, function(err, rows){
55   //untuk menampilkan error
56   if(err){
57     return res.status(500).json({
58       status: false,
59       message: 'Error, Try again later'
60     })
61   }else{
62     return res.status(201).json({
63       status: true,
64       message: 'Data input Successfully'
65     })
66   }
67 })
68
69 //detail data
70 router.get('/:id', function(req, res){
71   let id = req.params.id
72 }
```

14. Lalu buatlah array sesuai table yang telah dibuat di phpMYAdmin

```
110 router.patch('/update/:id',[
111   body('genre').notEmpty(),
112   body('judul').notEmpty(),
113   body('pemeran').notEmpty(),
114   body('produser').notEmpty(),
115   body('director').notEmpty(),
116   body('penulis').notEmpty(),
117   body('tanggalrilis').notEmpty(),
118   body('production').notEmpty(),
119 ],(req,res)=>{
120   const errors = validationResult(req)
121   if(!errors.isEmpty()){
122     return res.status(422).json({
123       errors:errors.array()
124     })
125   }
126
127   let id = req.params.id
128
129   let formData={
130     genre: req.body.genre,
131     judul: req.body.judul,
132     pemeran: req.body.pemeran,
133     produser: req.body.produser,
134     director: req.body.director,
135     tanggalrilis: req.body.tanggalrilis,
136     production: req.body.production,
137   }
138
139 }
```

```
108 // update / edit data
109
110 router.patch('/update/:id',[
111   body('genre').notEmpty(),
112   body('judul').notEmpty(),
113   body('pemeran').notEmpty(),
114   body('produser').notEmpty(),
115   body('director').notEmpty(),
116   body('penulis').notEmpty(),
117   body('tanggalrilis').notEmpty(),
118   body('production').notEmpty(),
119 ],(req,res)=>{
120   const errors = validationResult(req)
121   if(!errors.isEmpty()){
122     return res.status(422).json({
123       errors:errors.array()
124     })
125   }
126
127   let id = req.params.id
128
129   let formData = {
```

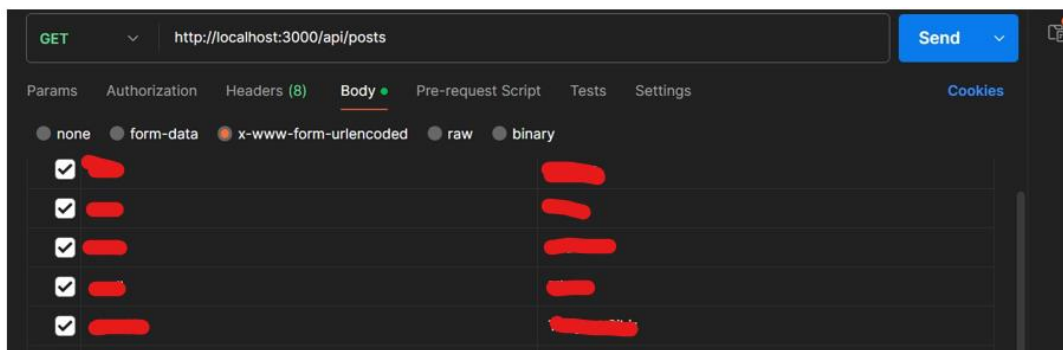
15. Wajib menyalakan Xampp dengan kondisi Apache & MySQL menyala



16. Selanjutnya kita mencoba run di terminal dengan command 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 running in http://localhost:3000
Connection Success
```

17. Selanjutnya membuka aplikasi postman mencoba menampilkan, menambah, mengupdate, dan menghapus data di postman





```

40
47 --
48 -- Dumping data for table `posts`
49 --
50
51 INSERT INTO `posts` (`id`,`judul`,`genre`,`sutradara`,`tahun_rilis`,`pengarang`,`durasi`,`tokoh`,`peran`,`watak`,`negara`,`produser`,`bahasa`,`perusa
52 (8, 'Sekawan Limo', 'Horror Komedi', 'Bayu Skak', '2024', 'Nona Ica', '1 jam 52 menit', 'Dono Pradana', 'Deri', 'Baik', 'Indonesia', 'Chand Parwez Servia', 'Jawa
53 (9, 'Miracle in cell No.7', 'Drama komedi', 'Hanung Bramantyo', '2022', 'Alim Sudio', '2 jam 25 menit', 'Vino G.Bastian', 'Dodo Rozak', 'Baik dan penyabar', 'inde
54 (10, 'Yowis Ben', 'Komedi', 'Bayu Skak', '2018', 'Bayu Skak', '1 jam 39 menit', 'Brandon Salim', 'Nando', 'Baik', 'Indonesia', 'Fiaz Servia', 'Indonesia,jawa', '
55 (11, 'Titanic', 'Romance', 'James Cameron', '1997', 'James Cameron', '3 jam 14 menit', 'Kate Winslet', 'Rose', 'Licik', 'America Serikat', 'James Cameron', 'Inggr
56 (12, 'The Lord of the rings', 'Fantasi', 'Peter Jackson', '1997', 'Fran Walsh', '9 jam 18 menit', 'Frodo Baggins', 'Elijah Wood', 'Tertutup', 'Britania Raya', 'Pe
57
58 --
59 -- Indexes for dumped tables
60 --
61
62 --
63 -- Indexes for table `posts`
64 --
65 ALTER TABLE `posts`
66 ADD PRIMARY KEY (`id`);
67
68 --
69 -- AUTO_INCREMENT for dumped tables
70 --
71
72 --
73 -- AUTO_INCREMENT for table `posts`
74 --
75 ALTER TABLE `posts`
76 MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=13;
77 COMMIT;

```

HTTP <http://localhost:3000/api/posts> Save </>

**POST** <http://localhost:3000/api/posts/store> Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary

☒ ☒

**Body** Cookies Headers (7) Test Results 201 Created 35 ms 291 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "status": true,
3   "message": "Data input Successfully"
4 }

```

HTTP <http://localhost:3000/api/posts/delete/22> Save

**DELETE** <http://localhost:3000/api/posts/delete/22> Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary

Key	Value

**Body** Cookies Headers (7) Test Results 200 OK 20 ms 284 B Save Response

Pretty Raw Preview Visualize JSON

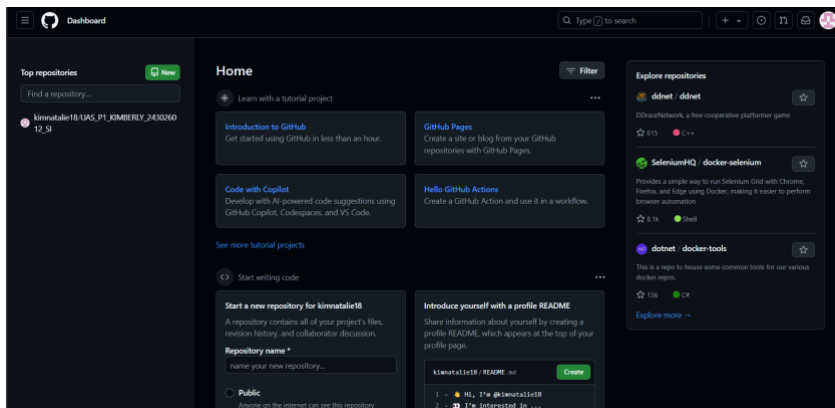
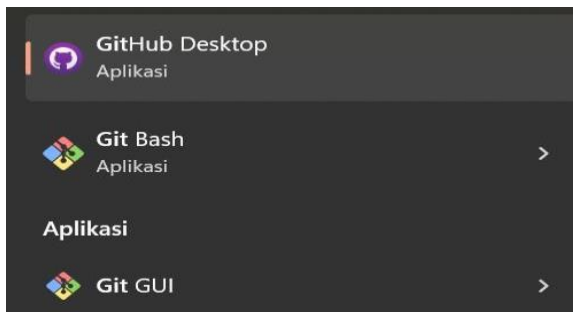
```

1 {
2   "status": true,
3   "message": "Data has been deleted"
4 }

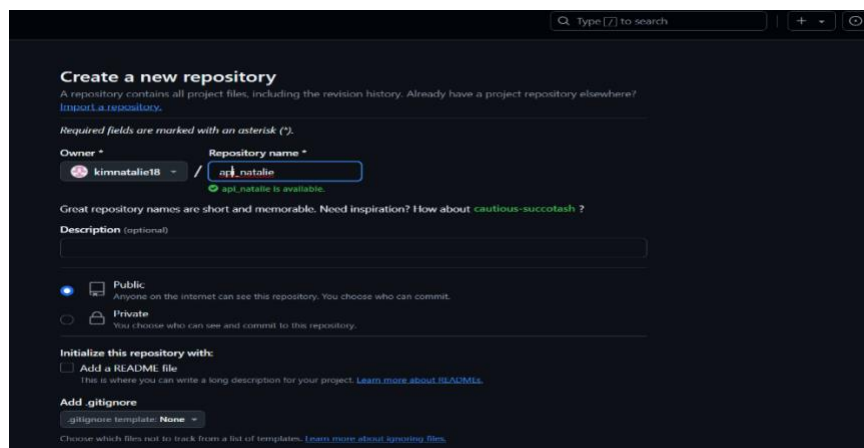
```

18. Selanjutnya kita mengunduh git dan login ke github

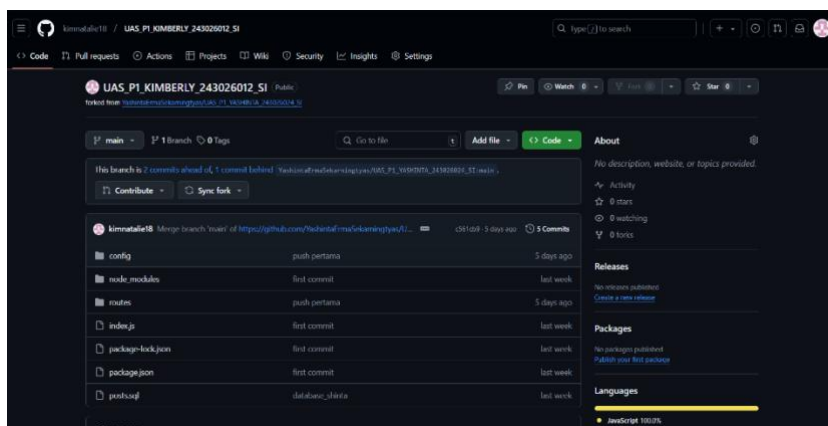




19. Jika sudah, membuat repositori baru



20. Selanjutnya melakukan commit lalu melakukan push file ke github



21. Terakhir mengecek codingan yang sudah di push ke github apakah ada terjadi error atau tidak