

LAPORAN PRAKTIKUM LATIHAN 07



Tolak Ivandi Yudistiawan

3122522020

D3 PSDKU Sumenep

PRODI D3 TEKNIK INFORMATIKA  
DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER  
PENS PSDKU SUMENEP

Tabel kartu\_keluarga

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	no_kk	char(17)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	2	alamat	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3	rt	char(3)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4	rw	char(3)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	5	kode_pos	char(6)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	6	desa_kelurahan	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	7	kecamatan	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	8	kabupaten_kota	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	9	provinsi	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More

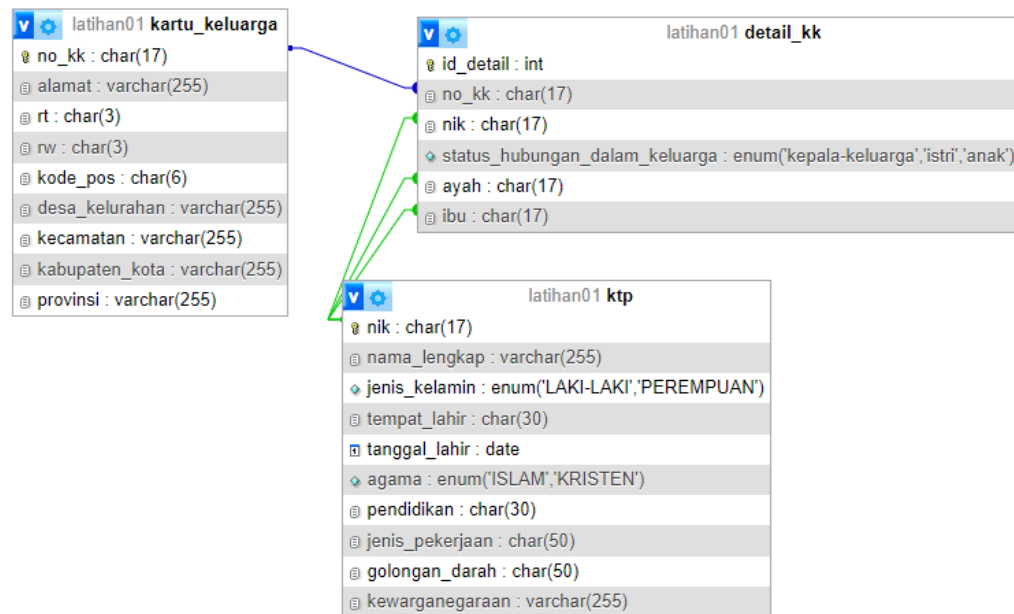
Tabel nik

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	nik	char(17)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	2	nama_lengkap	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3	jenis_kelamin	enum('LAKI-LAKI', 'PEREMPUAN')	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4	tempat_lahir	char(30)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	5	tanggal_lahir	date			No	None			Change  Drop  More
<input type="checkbox"/>	6	agama	enum('ISLAM', 'KRISTEN')	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	7	pendidikan	char(30)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	8	jenis_pekerjaan	char(50)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	9	golongan_darah	char(50)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	10	kewarganegaraan	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More

Tabel detail\_kk

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id_detail	int			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2	no	char(17)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3	nik	char(17)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4	status_hubungan_dalam_keluarga	enum('kepala-keluarga', 'istri', 'anak')	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	5	ayah	char(17)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More
<input type="checkbox"/>	6	ibu	char(17)	utf8mb4_0900_ai_ci		No	None			Change  Drop  More

designer



Di bawah ini adalah kode programnya nik

```
1  const express = require("express");
2  const router = express.Router();
3
4  const { body, validationResult } = require("express-validator");
5
6  const connection = require("../config/db");
7
8  router.get("/", function (req, res) {
9    connection.query("SELECT * FROM ktp ORDER BY nik DESC", function (err, rows) {
10     if (err) {
11       return res.status(500).json({
12         status: false,
13         message: "Server error",
14       });
15     } else {
16       return res.status(200).json({
17         status: true,
18         message: "Data berhasil",
19         data: rows,
20       });
21     }
22   });
23 });
```

```

25 router.post(
26   "/add",
27   [
28     body("nik").notEmpty(),
29     body("nama_lengkap").notEmpty(),
30     body("jenis_kelamin").notEmpty(),
31     body("tempat_lahir").notEmpty(),
32     body("tanggal_lahir").notEmpty(),
33     body("agama").notEmpty(),
34     body("pendidikan").notEmpty(),
35     body("jenis_pekerjaan").notEmpty(),
36     body("golongan_darah").notEmpty(),
37     body("kewarganegaraan").notEmpty(),
38   ],
39   (req, res) => {
40     const errors = validationResult(req);
41     if (!errors.isEmpty()) {
42       return res.status(422).json({
43         errors: errors.array(),
44       });
45     }
46
47     const data = {
48       nik: req.body.nik,
49       nama_lengkap: req.body.nama_lengkap,
50       jenis_kelamin: req.body.jenis_kelamin,
51       tempat_lahir: req.body.tempat_lahir,
52       tanggal_lahir: req.body.tanggal_lahir,
53       agama: req.body.agama,
54       pendidikan: req.body.pendidikan,
55       jenis_pekerjaan: req.body.jenis_pekerjaan,
56       golongan_darah: req.body.golongan_darah,

```

```

56       golongan_darah: req.body.golongan_darah,
57       kewarganegaraan: req.body.kewarganegaraan,
58     };
59
60     connection.query("INSERT INTO ktp SET ?", data, function (err, result) {
61       if (err) {
62         return res.status(500).json({
63           status: false,
64           message: "Server error",
65         });
66       } else {
67         data.id = result.insertId;
68         return res.status(201).json({
69           status: true,
70           message: "Data berhasil ditambahkan",
71           data: data,
72         });
73       }
74     });
75   }
76 );

```

```

78 router.get("/:nik", function (req, res) {
79     const nik = req.params.nik;
80     connection.query(
81         "SELECT * FROM ktp WHERE nik = ?",
82         nik,
83         function (err, rows) {
84             if (err) {
85                 return res.status(500).json({
86                     status: false,
87                     message: "Server error",
88                 });
89             } else if (rows.length === 0) {
90                 return res.status(404).json({
91                     status: false,
92                     message: "Data tidak ditemukan",
93                 });
94             } else {
95                 return res.status(200).json({
96                     status: true,
97                     message: "Data berhasil ditemukan",
98                     data: rows[0], // Hanya satu baris yang diambil karena nik bersifat unik
99                 });
100             }
101         }
102     );
103 });
104

```

```

105 ✓ router.patch("/update/:nik", (req, res) => {
106     const nik = req.params.nik;
107     ✓ const updatedData = {
108         nik: req.body.nik,
109         nama_lengkap: req.body.nama_lengkap,
110         jenis_kelamin: req.body.jenis_kelamin,
111         tempat_lahir: req.body.tempat_lahir,
112         tanggal_lahir: req.body.tanggal_lahir,
113         agama: req.body.agama,
114         pendidikan: req.body.pendidikan,
115         jenis_pekerjaan: req.body.jenis_pekerjaan,
116         golongan_darah: req.body.golongan_darah,
117         kewarganegaraan: req.body.kewarganegaraan,
118     };
119
120 ✓ connection.query(
121     "UPDATE ktp SET ? WHERE nik = ?",
122     [updatedData, nik],
123     ✓ function (err, result) {
124     ✓         if (err) {
125     ✓             return res.status(500).json({
126                 status: false,
127                 message: "Server error",
128             });
129     ✓         } else if (result.affectedRows === 0) {
130     ✓             return res.status(404).json({
131                 status: false,
132                 message: "Data tidak ditemukan",
133             });
134     ✓         } else {

```

```

129     } else if (result.affectedRows === 0) {
130         return res.status(404).json({
131             status: false,
132             message: "Data tidak ditemukan",
133         });
134     } else {
135         return res.status(200).json({
136             status: true,
137             message: "Data berhasil diupdate",
138         });
139     }
140 }
141 );
142 });

```

```

144 ✓ router.delete("/delete/:nik", (req, res) => {
145     const nik = req.params.nik;
146
147     ✓ connection.query(
148         "DELETE FROM ktp WHERE nik = ?",
149         nik,
150     ✓ function (err, result) {
151     ✓         if (err) {
152     ✓             return res.status(500).json({
153                 status: false,
154                 message: "Server error",
155             });
156     ✓         } else if (result.affectedRows === 0) {
157     ✓             return res.status(404).json({
158                 status: false,
159                 message: "Data tidak ditemukan",
160             });
161     ✓         } else {
162     ✓             return res.status(200).json({
163                 status: true,
164                 message: "Data berhasil dihapus",
165             });
166         }
167     }
168 );
169 });
170
171
172 module.exports = router;
173

```

Di bawah ini adalah output posman nya

**POST** ▼ http://localhost:8080/api/ktp/add Send

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

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

	Key	Value	Description	...	Bulk
<input checked="" type="checkbox"/>	nik	1			
<input checked="" type="checkbox"/>	nama_lengkap	tolak ivandi			
<input checked="" type="checkbox"/>	jenis_kelamin	LAKI-LAKI			
<input checked="" type="checkbox"/>	tempat_lahir	sumenep			

**Body** Cookies Headers (7) Test Results 🌐 Status: **201 Created** Time: **175 ms** Size: **548 B** 💾 Save as Example

**Pretty** Raw Preview Visualize JSON ▼ 🔗

```
1
2  "status": true,
3  "message": "Data berhasil ditambahkan",
4  "data": {
5    "nik": "1",
6    "nama_lengkap": "tolak ivandi",
7    "jenis_kelamin": "LAKI-LAKI",
8    "tempat_lahir": "sumenep",
9    "tanggal_lahir": "2003-12-27",
10   }
```

**GET** ▼ http://localhost:8080/api/ktp Send ▼

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings Cookies

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

This request does not have a body

**Body** Cookies Headers (7) Test Results 🌐 Status: **200 OK** Time: **14 ms** Size: **791 B** 💾 Save as Example ⋮

**Pretty** Raw Preview Visualize JSON ▼ 🔗 🔍

```
18  "nik": "1",
19  "nama_lengkap": "tolak ivandi",
20  "jenis_kelamin": "LAKI-LAKI",
21  "tempat_lahir": "sumenep",
22  "tanggal_lahir": "2003-12-26T17:00:00.000Z",
23  "agama": "ISLAM",
24  "pendidikan": "D3",
25  "jenis_pekerjaan": "mahasiswa",
26  "golongan_darah": "A",
```

GET http://localhost:8080/api/ktp/1 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 17 ms Size: 548 B Save as Example

Pretty Raw Preview Visualize JSON

```
1  {
2    "status": true,
3    "message": "Data berhasil ditemukan",
4    "data": {
5      "nik": "1",
6      "nama_lengkap": "tolak ivandi",
7      "jenis_kelamin": "LAKI-LAKI",
8      "tempat_lahir": "sumenep",
9      "tanggal_lahir": "2003-12-26T17:00:00.000Z",
10     "agama": "ISLAM",
11     "pendidikan": "D3",
12     "jenis_pekerjaan": "mahasiswa",
13     "golongan_darah": "A",
14     "kewarganegaraan": "indonesia"
15   }
16 }
```

PATCH http://localhost:8080/api/ktp/update/1234 Send

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

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1  {
2    "nik": "1234",
3    "nama_lengkap": "John CENA",
4    "jenis_kelamin": "Laki-Laki",
5    "tempat_lahir": "Jakarta",
6    "tanggal_lahir": "2000-01-01",
7    "agama": "Islam",
8    "pendidikan": "S1",
9    "jenis_pekerjaan": "Pegawai",
10   "golongan_darah": "A+",
11   "kewarganegaraan": "WNI"
12 }
```

Body Cookies Headers (7) Test Results

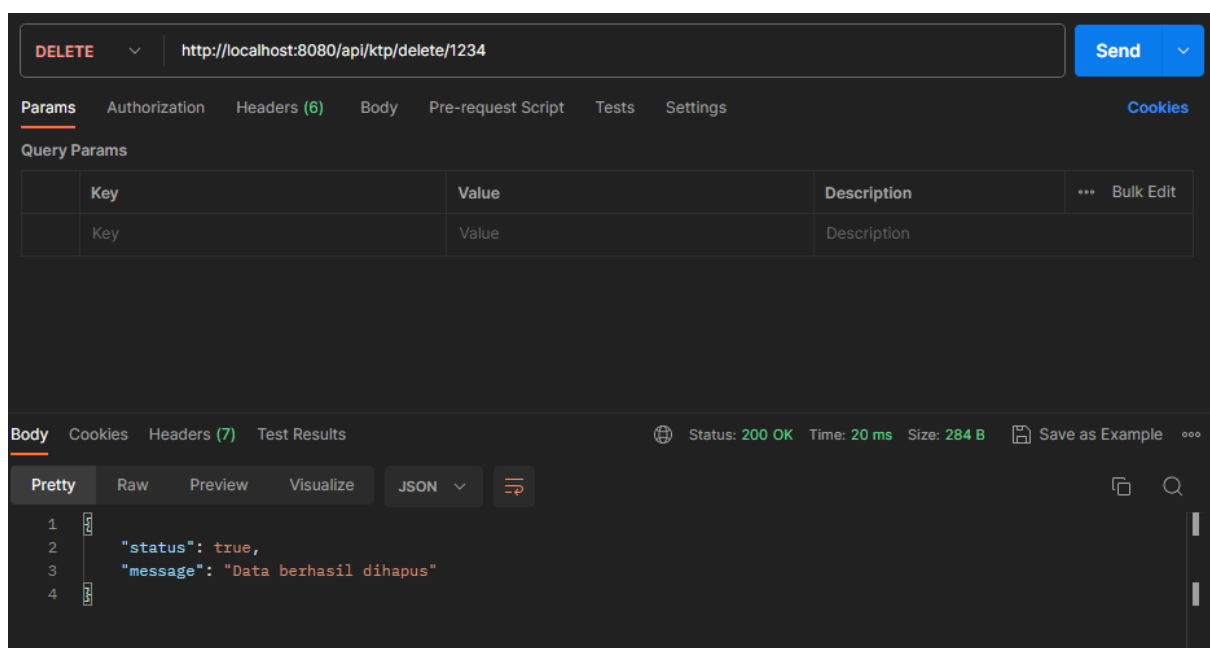
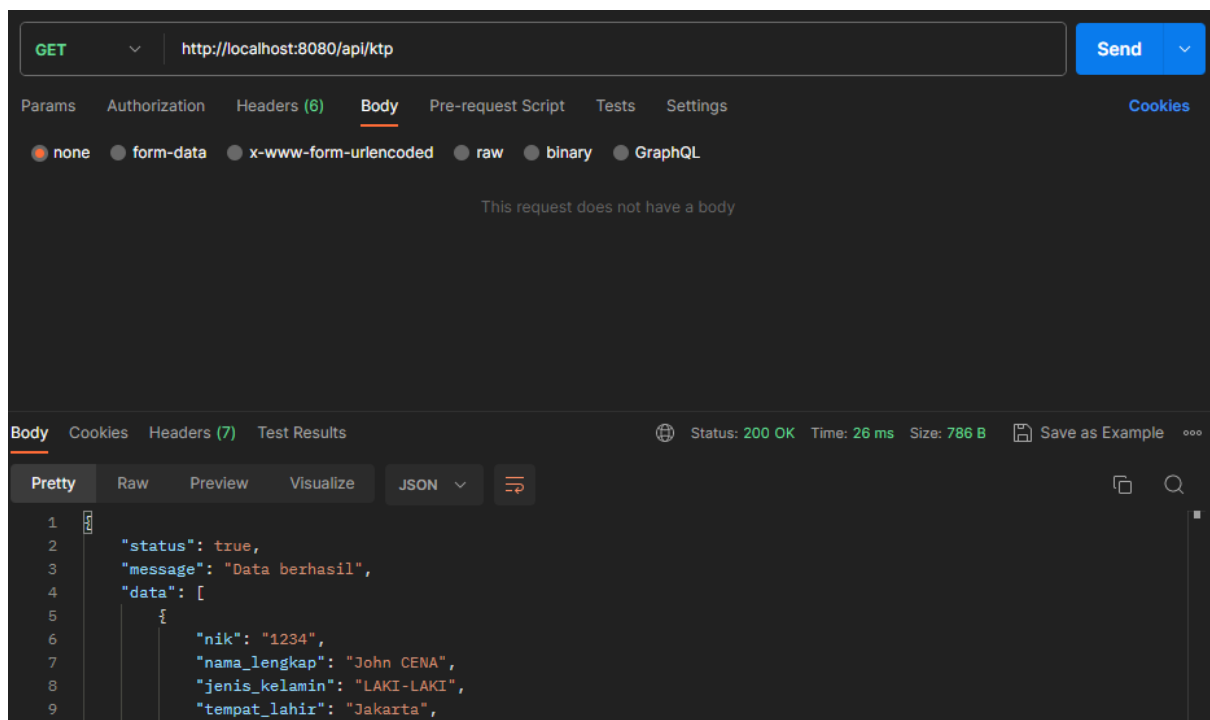
Status: 200 OK Time: 53 ms Size: 285 B Save as Example

Pretty Raw Preview Visualize JSON

```
1  {
2    "status": true,
3    "message": "Data berhasil diupdate"
4  }
```



Setelah di update



Di bawah ini adalah kode programnya ktp

```
1  const express = require("express");
2  const router = express.Router();
3
4  const { body, validationResult } = require("express-validator");
5
6  const connection = require("../config/db");
7
8  router.get("/", function (req, res) {
9    connection.query(
10     ⚡ "SELECT * FROM kartu_keluarga ORDER BY no_kk DESC",
11     function (err, rows) {
12       if (err) {
13         return res.status(500).json({
14           status: false,
15           message: "Server error",
16           error: err,
17         });
18       } else {
19         return res.status(200).json({
20           status: true,
21           message: "Data berhasil",
22           data: rows,
23         });
24       }
25     }
26   );
27 });
28
29 module.exports = router;
```

```

28
29  router.post(
30    "/add",
31    [
32      body("no_kk").notEmpty(),
33      body("alamat").notEmpty(),
34      body("rt").notEmpty(),
35      body("rw").notEmpty(),
36      body("kode_pos").notEmpty(),
37      body("desa_kelurahan").notEmpty(),
38      body("kecamatan").notEmpty(),
39      body("kabupaten_kota").notEmpty(),
40      body("provinsi").notEmpty(),
41    ],
42    (req, res) => {
43      const errors = validationResult(req);
44      if (!errors.isEmpty()) {
45        return res.status(422).json({
46          errors: errors.array(),
47        });
48      }
49
50      const data = {
51        no_kk: req.body.no_kk,
52        alamat: req.body.alamat,
53        rt: req.body.rt,
54        rw: req.body.rw,
55        kode_pos: req.body.kode_pos,
56        desa_kelurahan: req.body.desa_kelurahan,
57        kecamatan: req.body.kecamatan,
58        kabupaten_kota: req.body.kabupaten_kota,
59        provinsi: req.body.provinsi,
60      };

```

```

61
62    connection.query("INSERT INTO kartu_keluarga SET ?", data, function (err, result) {
63      if (err) {
64        return res.status(500).json({
65          status: false,
66          message: "Server error",
67        });
68      } else {
69        data.id = result.insertId;
70        return res.status(201).json({
71          status: true,
72          message: "Data berhasil ditambahkan",
73          data: data,
74        });
75      }
76    });
77  }
78  );
79
80  module.exports = router;

```

```
137 router.get("/:kk", function (req, res) {
138   const kk = req.params.kk;
139   connection.query(
140     "SELECT * FROM kartu_keluarga WHERE no_kk = ?",
141     kk,
142     function (err, rows) {
143       if (err) {
144         return res.status(500).json({
145           status: false,
146           message: "Server error",
147           error: err,
148         });
149       } else if (rows.length === 0) {
150         return res.status(404).json({
151           status: false,
152           message: "Data tidak ditemukan",
153         });
154       } else {
155         return res.status(200).json({
156           status: true,
157           message: "Data berhasil ditemukan",
158           data: rows[0], // Hanya satu baris yang diambil
159         });
160       }
161     }
162   );
163 });
164 module.exports = router;
```

```

192 router.patch("/update/:no_kk", (req, res) => {
193   const no_kk = req.params.no_kk;
194   const updatedData = {
195     no_kk: req.body.no_kk,
196     alamat: req.body.alamat,
197     rt: req.body.rt,
198     rw: req.body.rw,
199     kode_pos: req.body.kode_pos,
200     desa_kelurahan: req.body.desa_kelurahan,
201     kecamatan: req.body.kecamatan,
202     kabupaten_kota: req.body.kabupaten_kota,
203     provinsi: req.body.provinsi,
204   };
205
206   connection.query(
207     "UPDATE kartu_keluarga SET ? WHERE no_kk = ?",
208     [updatedData, no_kk],
209     function (err, result) {
210       if (err) {
211         return res.status(500).json({
212           status: false,
213           message: "Server error",
214         });
215       } else if (result.affectedRows === 0) {
216         return res.status(404).json({
217           status: false,
218           message: "Data tidak ditemukan",
219         });
220       } else {
221         return res.status(200).json({
222           status: true,

```

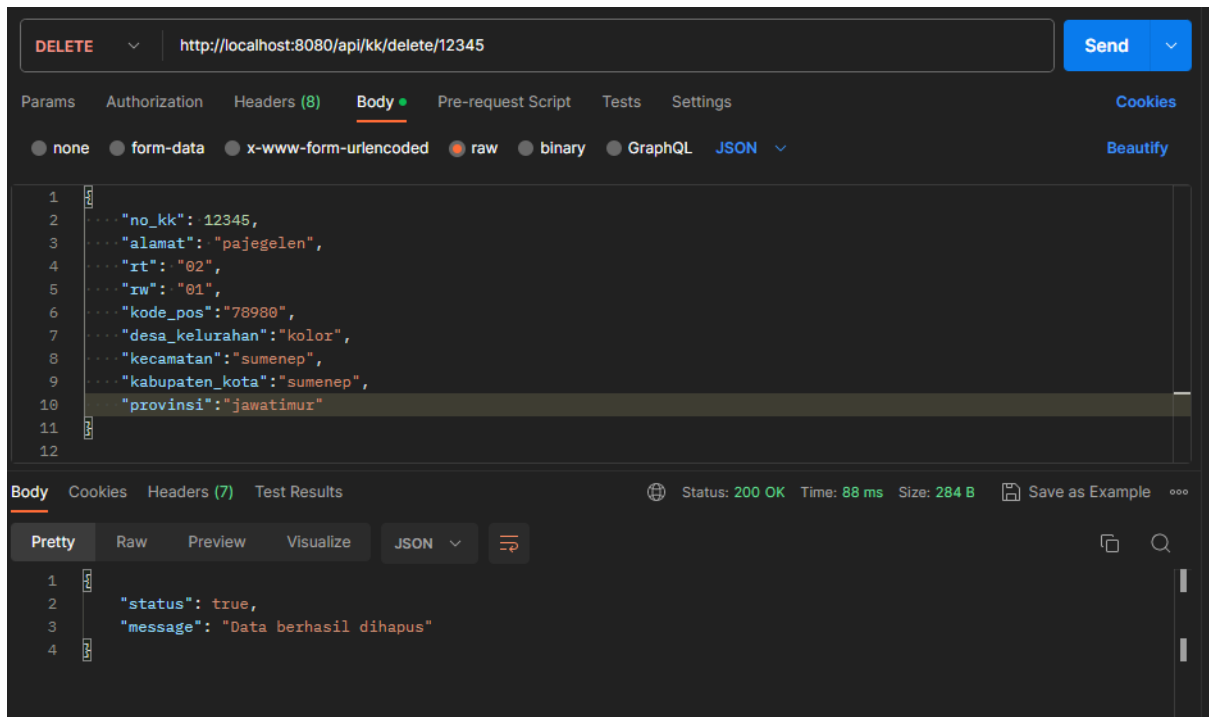
```

220       } else {
221         return res.status(200).json({
222           status: true,
223           message: "Data berhasil diupdate",
224         });
225       }
226     }
227   );
228 });
229 module.exports = router;
230

```

```
230 router.delete("/delete/:no_kk", (req, res) => {
231   const no_kk = req.params.no_kk;
232
233   connection.query(
234     "DELETE FROM kartu_keluarga WHERE no_kk = ?",
235     no_kk,
236     function (err, result) {
237       if (err) {
238         return res.status(500).json({
239           status: false,
240           message: "Server error",
241         });
242       } else if (result.affectedRows === 0) {
243         return res.status(404).json({
244           status: false,
245           message: "Data tidak ditemukan",
246         });
247       } else {
248         return res.status(200).json({
249           status: true,
250           message: "Data berhasil dihapus",
251         });
252       }
253     }
254   );
255 });
256
257 module.exports = router;
258
```

## Hasil OUTPUT dari posman



The image shows a Postman interface for a DELETE request. The URL is `http://localhost:8080/api/kk/delete/12345`. The request body is a JSON object with the following fields: `no_kk`, `alamat`, `rt`, `rw`, `kode_pos`, `desa_kelurahan`, `kecamatan`, `kabupaten_kota`, and `provinsi`. The response status is 200 OK, and the response body is a JSON object with `status` and `message` fields.

```
DELETE http://localhost:8080/api/kk/delete/12345
```

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

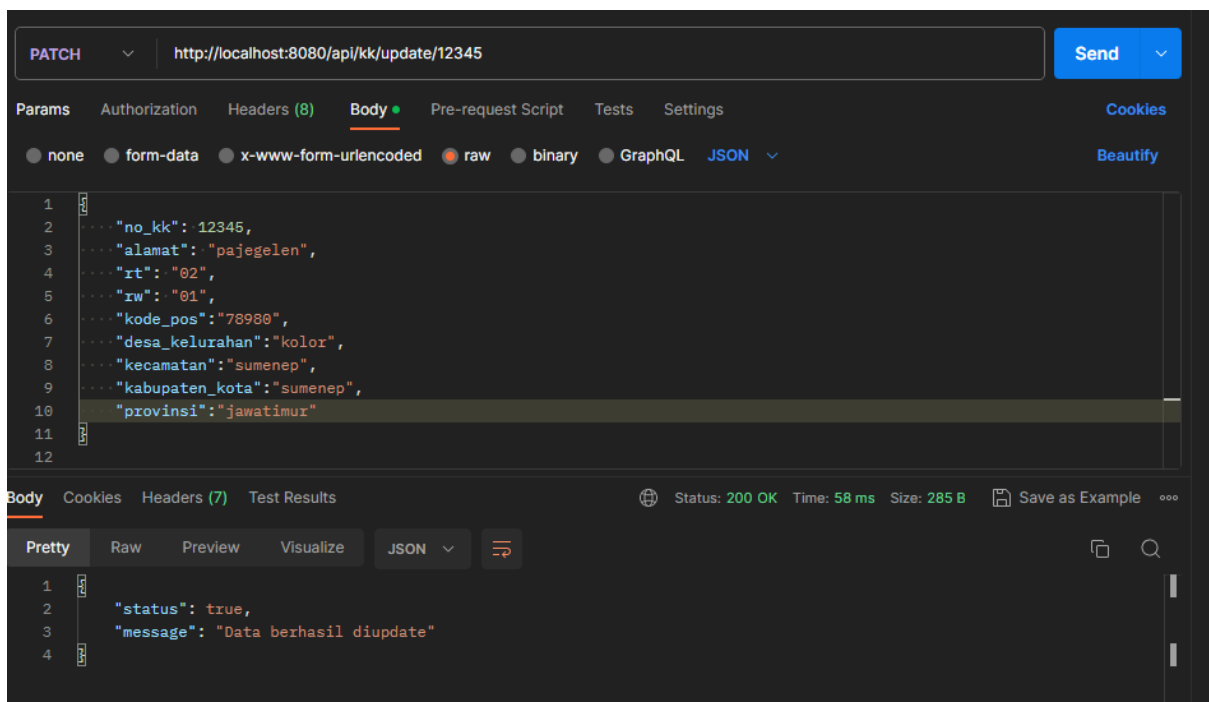
none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "no_kk": 12345,
3   "alamat": "pajegelen",
4   "rt": "02",
5   "rw": "01",
6   "kode_pos": "78980",
7   "desa_kelurahan": "kolor",
8   "kecamatan": "sumenep",
9   "kabupaten_kota": "sumenep",
10  "provinsi": "jawa timur"
11 }
12
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 88 ms Size: 284 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": true,
3   "message": "Data berhasil dihapus"
4 }
```



The image shows a Postman interface for a PATCH request. The URL is `http://localhost:8080/api/kk/update/12345`. The request body is a JSON object with the following fields: `no_kk`, `alamat`, `rt`, `rw`, `kode_pos`, `desa_kelurahan`, `kecamatan`, `kabupaten_kota`, and `provinsi`. The response status is 200 OK, and the response body is a JSON object with `status` and `message` fields.

```
PATCH http://localhost:8080/api/kk/update/12345
```

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

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "no_kk": 12345,
3   "alamat": "pajegelen",
4   "rt": "02",
5   "rw": "01",
6   "kode_pos": "78980",
7   "desa_kelurahan": "kolor",
8   "kecamatan": "sumenep",
9   "kabupaten_kota": "sumenep",
10  "provinsi": "jawa timur"
11 }
12
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 58 ms Size: 285 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": true,
3   "message": "Data berhasil diupdate"
4 }
```

GET

http://localhost:8080/api/kk/12345

Send

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettingsCookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

BodyCookiesHeaders (7)Test Results

Status: 200 OKTime: 8 msSize: 471 BSave as Example

PrettyRawPreviewVisualizeJSON

```
1  {
2    "status": true,
3    "message": "Data berhasil ditemukan",
4    "data": {
5      "no_kk": "12345",
6      "alamat": "karangbudi",
7      "rt": "01",
8      "rw": "02",
9      "kode_pos": "7896",
10     "desa_kelurahan": "karangan",
11     "kecamatan": "gapura",
12     "kabupaten_kota": "sumenep",
13     "provinsi": "jawa timur"
14   }
15 }
```

POST

http://localhost:8080/api/kk/add

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	no_kk	12345			
<input checked="" type="checkbox"/>	alamat	karangbudi			
<input checked="" type="checkbox"/>	rt	01			
<input checked="" type="checkbox"/>	rw	02			
<input checked="" type="checkbox"/>	kode_pos	7896			

BodyCookiesHeaders (7)Test Results

Status: 201 CreatedTime: 105 msSize: 485 BSave as Example

PrettyRawPreviewVisualizeJSON

```
1  {
2    "status": true,
3    "message": "Data berhasil ditambahkan",
4    "data": {
5      "no_kk": "12345",
6      "alamat": "karangbudi",
7      "rt": "01",
8      "rw": "02",
9      "kode_pos": "7896",
10     "desa_kelurahan": "karangan",
11     "kecamatan": "gapura",
12     "kabupaten_kota": "sumenep",
13     "provinsi": "jawa timur"
14   }
15 }
```



LATIHAN 7 / ADD KTP

GET http://localhost:8080/api/kk Send

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

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

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	nik	1			
<input checked="" type="checkbox"/>	nama_lengkap	tolak ivandi			
<input checked="" type="checkbox"/>	jenis_kelamin	LAKI-LAKI			
<input checked="" type="checkbox"/>	tempat_lahir	sumenep			
<input checked="" type="checkbox"/>	tanggal_lahir	2003-12-27			
<input checked="" type="checkbox"/>	agama	ISLAM			

Body Cookies Headers (7) Test Results Status: 200 OK Time: 61 ms Size: 286 B Save as Example

Pretty Raw Preview Visualize JSON Copy Search

```

1
2  "status": true,
3  "message": "Data berhasil",
4  "data": []
5

```

Di bawah ini adalah kode programnya detail\_kk

```

1  const express = require("express");
2  const router = express.Router();
3
4  const { body, validationResult } = require("express-validator");
5
6  const connection = require("../config/db");
7
8  router.get("/", function (req, res) {
9    connection.query(
10     "SELECT * FROM detail_kk ORDER BY id_detail DESC",
11     function (err, rows) {
12       if (err) {
13         return res.status(500).json({
14           status: false,
15           message: "Server error",
16           error: err,
17         });
18       } else {
19         return res.status(200).json({
20           status: true,
21           message: "Data berhasil",
22           data: rows,
23         });
24       }
25     }
26   );
27 });
28
29 module.exports = router;
30

```

```

8 > router.get("/", function (req, res) { ...
27 });
28
29 router.post(
30   "/add",
31   [
32     body("no_kk").notEmpty(),
33     body("nik").notEmpty(),
34     body("status_hubungan_dalam_keluarga").notEmpty(),
35     body("ayah").notEmpty(),
36     body("ibu").notEmpty(),
37   ],
38   (req, res) => {
39     const errors = validationResult(req);
40     if (!errors.isEmpty()) {
41       return res.status(422).json({
42         errors: errors.array(),
43       });
44     }
45
46     const data = {
47       no_kk: req.body.no_kk,
48       nik: req.body.nik,
49       status_hubungan_dalam_keluarga: req.body.status_hubungan_dalam_keluarga,
50       ayah: req.body.ayah,
51       ibu: req.body.ibu,
52     };

```

```

54     connection.query("INSERT INTO detail_kk SET ?", data, function (err, result) {
55       if (err) {
56         return res.status(500).json({
57           status: false,
58           message: "Server error",
59         });
60       } else {
61         data.id = result.insertId;
62         return res.status(201).json({
63           status: true,
64           message: "Data berhasil ditambahkan",
65           data: data,
66         });
67       }
68     });
69   }
70 );

```

```

73 router.patch("/update/:id_detail", (req, res) => {
74   const id_detail = req.params.id_detail;
75   const updatedData = {
76     no_kk: req.body.no_kk,
77     nik: req.body.nik,
78     status_hubungan_dalam_keluarga: req.body.status_hubungan_dalam_keluarga,
79     ayah: req.body.ayah,
80     ibu: req.body.ibu,
81   };
82
83   connection.query(
84     "UPDATE detail_kk SET ? WHERE id_detail = ?",
85     [updatedData, id_detail],
86     function (err, result) {
87       if (err) {
88         return res.status(500).json({
89           status: false,
90           message: "Server error",
91         });
92       } else if (result.affectedRows === 0) {
93         return res.status(404).json({
94           status: false,
95           message: "Data tidak ditemukan",
96         });
97       } else {
98         return res.status(200).json({
99           status: true,
100           message: "Data berhasil diupdate",
101         });
102       }
103     }
104   );
105 });

```

```

107 router.delete("/delete/:id_detail", (req, res) => {
108     const id_detail = req.params.id_detail;
109
110     connection.query(
111         "DELETE FROM detail_kk WHERE id_detail = ?",
112         id_detail,
113         function (err, result) {
114             if (err) {
115                 return res.status(500).json({
116                     status: false,
117                     message: "Server error",
118                 });
119             } else if (result.affectedRows === 0) {
120                 return res.status(404).json({
121                     status: false,
122                     message: "Data tidak ditemukan",
123                 });
124             } else {
125                 return res.status(200).json({
126                     status: true,
127                     message: "Data berhasil dihapus",
128                 });
129             }
130         }
131     );
132 });
133
134
135 module.exports = router;

```

Hasil OUTPUT dari posman

The screenshot shows a Postman interface for a DELETE request. The URL is `http://localhost:8080/api/detail/delete/1`. The request body is set to `x-www-form-urlencoded` with two parameters: `no_kk` with value `1234` and `nilk` with value `1`. The response status is `200 OK` with a time of `83 ms` and a size of `284 B`. The response body is a JSON object: `{\"status\": true, \"message\": \"Data berhasil dihapus\"}`.

Key	Value	Description
<input type="checkbox"/>		
<input checked="" type="checkbox"/> no_kk	1234	
<input checked="" type="checkbox"/> nilk	1	

```

1  {
2    "status": true,
3    "message": "Data berhasil dihapus"
4  }

```

Copy link to request

PATCH  Send

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

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

	Key	Value	Description	...	Bulk Edit
<input type="checkbox"/>					
<input checked="" type="checkbox"/>	no_kk	1234			
<input checked="" type="checkbox"/>	nik	1			

Body Cookies Headers (7) Test Results 🌐 Status: 200 OK Time: 25 ms Size: 285 B Save as Example ...

Pretty Raw Preview Visualize JSON

```
1
2  "status": true,
3  "message": "Data berhasil diupdate"
4
```

HTTP LATIHAN 7 / ADD KTP Save ✎ 💬

GET  Send

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

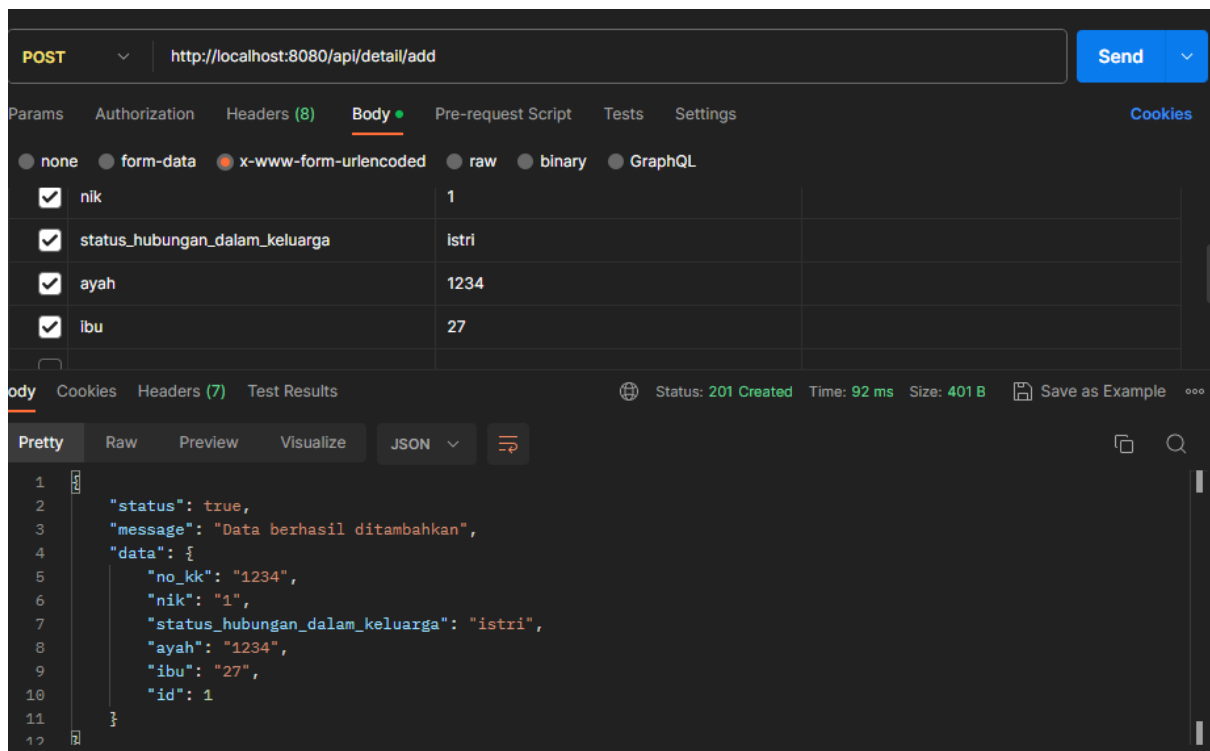
☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	nik	1			
<input checked="" type="checkbox"/>	nama_lengkap	tolak ivandi			
<input checked="" type="checkbox"/>	jenis_kelamin	LAKI-LAKI			

Body Cookies Headers (7) Test Results 🌐 Status: 200 OK Time: 30 ms Size: 286 B Save as Example ...

Pretty Raw Preview Visualize JSON

```
1
2  "status": true,
3  "message": "Data berhasil",
4  "data": []
5
```



<https://github.com/tolakivandi/tugas07>

<https://github.com/tolakivandi/latihan01-06>