

LATIHAN 2 PART 9
REST-API EXPRESS



Shera Zahra Alya Nasywa Hardian
(3122522026)
3 D3 PSDKU Sumenep

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

Membuat tabel dpi beserta atributnya

```
1 CREATE TABLE dpi (  
2     id_dpi INT AUTO_INCREMENT PRIMARY KEY,  
3     nama_dpi VARCHAR(255) NOT NULL,  
4     luas DECIMAL(10, 2)  
5 );  
6 |
```

Membuat tabel pemilik beserta atributnya

```
1 CREATE TABLE pemilik (  
2     id_pemilik INT AUTO_INCREMENT PRIMARY KEY,  
3     nama_pemilik VARCHAR(255) NOT NULL,  
4     alamat VARCHAR(255),  
5     no_hp VARCHAR(15) NOT NULL  
6 );  
7 |
```

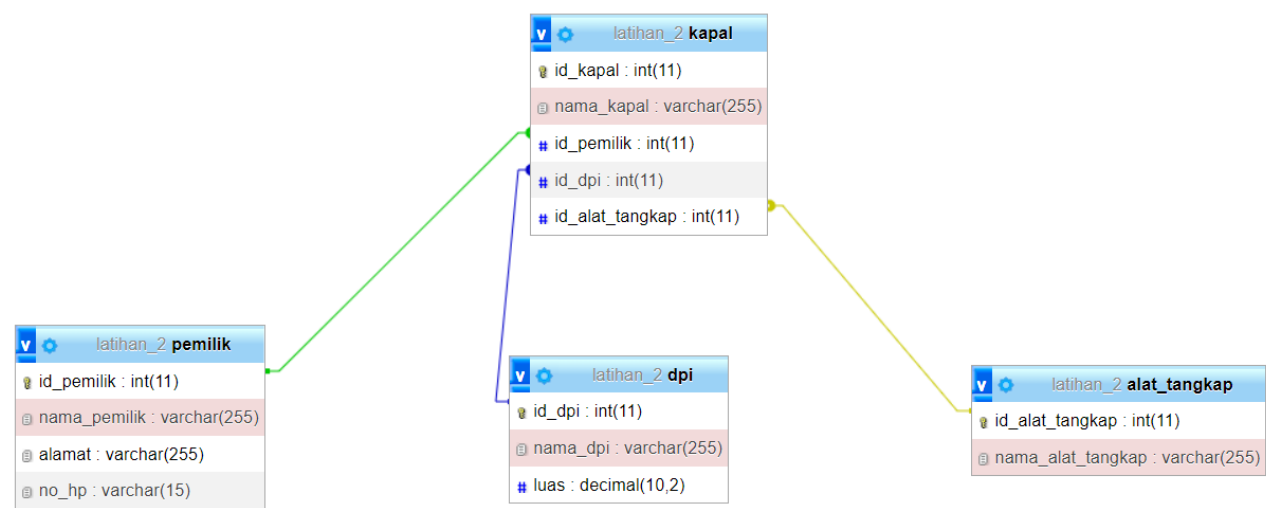
Membuat tabel alat_tangkap beserta atributnya

```
1 CREATE TABLE alat_tangkap (  
2     id_alat_tangkap INT AUTO_INCREMENT PRIMARY KEY,  
3     nama_alat_tangkap VARCHAR(255) NOT NULL  
4 );  
5 |
```

Membuat tabel kapal beserta atribut, lalu menambahkan relasi untuk dapat menghubungkan foreign key antar tabel

```
1 CREATE TABLE kapal (  
2     id_kapal INT PRIMARY KEY,  
3     nama_kapal VARCHAR(255),  
4     id_pemilik INT,  
5     id_dpi INT,  
6     id_alat_tangkap INT,  
7     FOREIGN KEY (id_pemilik) REFERENCES pemilik(id_pemilik),  
8     FOREIGN KEY (id_dpi) REFERENCES dpi(id_dpi),  
9     FOREIGN KEY (id_alat_tangkap) REFERENCES alat_tangkap(id_alat_tangkap)  
10 );  
11
```

Desain relasi yang berhasil dibuat, tabel kapal memiliki relasi dengan semua tabel dengan menjadikan ide dari tiap tabel sebagai foreign key



Membuat koneksi yang akan menghubungkan dengan database yang berisi tabel yang sudah dibuat

```
config > JS db.js > connection > database
1 let mysql = require('mysql');
2
3 let connection = mysql.createConnection({
4   host: 'localhost',
5   user: 'root',
6   password: '',
7   database: 'latihan_2'
8 });
9
10 connection.connect(function (error) {
11   if(!error){
12     console.log(error)
13   }else{
14     console.log('koneksi berhasil');
15   }
16 })
17
18 module.exports = connection;
```

Membuat koneksi agar program dapat dijalankan dan dicek melalui postman dan tentunya localhost

```
JS index.js > ...
1 const express = require('express')
2 const app = express()
3 const port = 3000
4
5 app.get('/', (req, res) => {
6   res.send("Halo SHERA!")
7 });
8
9 const pemilikRouter = require('./routes/pemilik');
10 app.use('/api/pemilik', pemilikRouter);
11
12 const dpiRouter = require('./routes/dpi');
13 app.use('/api/dpi', dpiRouter);
14
15 const alat_tangkapRouter = require('./routes/alat_tangkap');
16 app.use('/api/alat_tangkap', alat_tangkapRouter);
17
18 const kapalRouter = require('./routes/kapal');
19 app.use('/api/kapal', kapalRouter);
20
21 app.listen(port, () => {
22   console.log(`aplikasi berjalan di http://localhost:${port}`)
23 })
24
```

MEMBUAT CRUD DATA PEMILIK

```
routes > JS pemilik.js > ...
1  const express = require('express');
2  const router = express.Router();
3  const {body, validationResult } = require('express-validator');
4
5  const connection = require('../config/db.js');
6
7  > router.get('/', function (req, res){ ...
22  });
23
24  > router.post('/store', [ ...
54  ])
55
56  > router.get('/:id', function (req, res) { ...
79  })
80
81  > router.patch('/update/:id', [ ...
111  ])
112
113  > router.delete('/delete/:id', function(req, res){ ...
128  })
129
130  module.exports = router;
```

Script untuk dapat menghubungkan progra dengan database yang dibuat, memastikan data yang dimasukkan valid, menggunakan method get

```
7  router.get('/', function (req, res){
8      connection.query('select * from pemilik order by id_pemilik desc', function(err, rows){
9          if(err){
10             return res.status(500).json({
11                 status: false,
12                 message: 'Server Failed',
13             })
14         }else{
15             return res.status(200).json({
16                 status: true,
17                 message: 'Data Pemilik',
18                 data: rows
19             })
20         }
21     })
22 });
```

http://localhost:3000/api/pemilik

Save

GET

http://localhost:3000/api/pemilik

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	Key	Value	Description	Bulk Edit
--	-----	-------	-------------	-----------

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

"status": true,


"message": "Data Pemilik",


"data": []



Script untuk melakukan create data, menggunakan method post untuk dapat menambahkan data

```
24 router.post('/store', [  
25   body('nama_pemilik').notEmpty(),  
26   body('alamat').notEmpty(),  
27   body('no_hp').notEmpty()  
28 ],(req, res) => {  
29   const error = validationResult(req);  
30   if(!error.isEmpty()){  
31     return res.status(422).json({  
32       error: error.array()  
33     })  
34   }  
35   let Data = {  
36     nama_pemilik: req.body.nama_pemilik,  
37     alamat: req.body.alamat,  
38     no_hp: req.body.no_hp  
39   }  
40   connection.query('insert into pemilik set ?', Data, function(err, rows){  
41     if(err){  
42       return res.status(500).json({  
43         status: false,  
44         message: 'Server Error',  
45       })  
46     }else{  
47       return res.status(201).json({  
48         satus: true,  
49         message: 'Success..!',  
50         data: rows[0]  
51       })  
52     }  
53   })  
54 })
```

Memasukkan data sesuai dengan atribut dari database, jika tertulis succes maka data berhasil masuk

 http://localhost:3000/api/pemilik/store

 Save

POST

http://localhost:3000/api/pemilik/store

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 Edit
<input checked="" type="checkbox"/>	nama_pemilik	shera			
<input checked="" type="checkbox"/>	alamat	mojokerto			
<input checked="" type="checkbox"/>	no_hp	085732329729			

Body

Cookies

Headers (7)

Test Results


Pretty

Raw

Preview

Visualize

JSON



1

{

2

"satus": true,

3

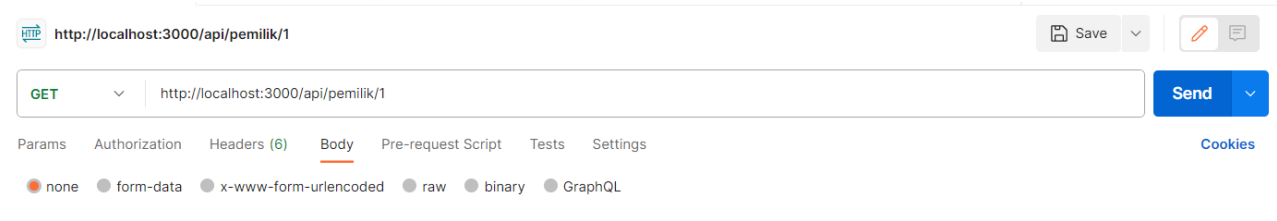
"message": "Success..!"

4

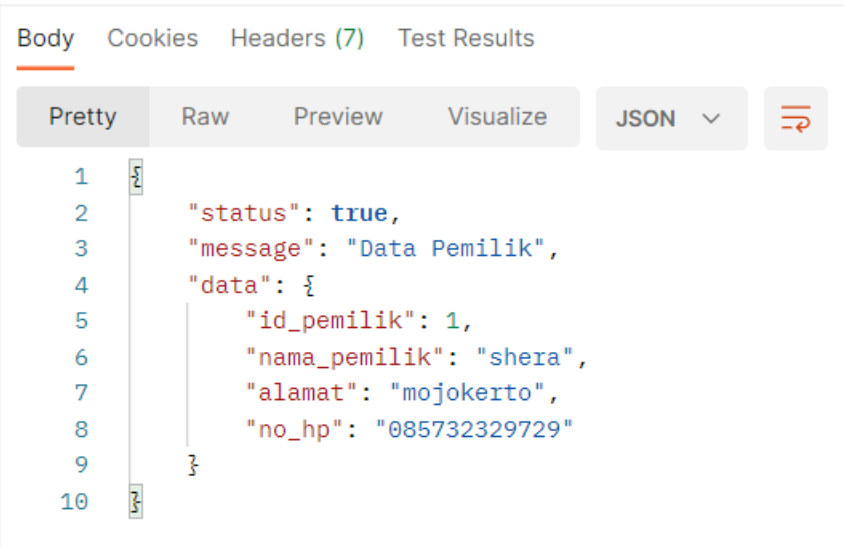
}

Mengecek apakah data sudah masuk dan dapat tertampil dengan menggunakan ide sebagai pemanggil, menggunakan method get

```
56 router.get('/:id)', function (req, res) {
57   let id = req.params.id;
58   connection.query(`select * from pemilik where id_pemilik = ${id}`, function (err, rows) {
59     if(err){
60       return res.status(500).json({
61         status: false,
62         message: 'Server Error',
63       })
64     }
65     if(rows.length <=0){
66       return res.status(404).json({
67         status: false,
68         message: 'Not Found',
69       })
70     }
71     else{
72       return res.status(200).json({
73         status: true,
74         message: 'Data Pemilik',
75         data: rows[0]
76       })
77     }
78   })
79 })
```



Jika data yang dipanggil sesuai idnya sudah muncul dan sesuai maka dinyatakan sudah berhasil



Script untuk dapat melakukan edit atau update data sesuai dengan id data mana yang dimasukkan, menggunakan method patch untuk menyimpan perubahan yang dibuat

```
81 router.patch('/update/:id', [
82   body('nama_pemilik').notEmpty(),
83   body('alamat').notEmpty(),
84   body('no_hp').notEmpty()
85 ], (req,res) => {
86   const error = validationResult(req);
87   if(!error.isEmpty()){
88     return res.status(422).json({
89       error: error.array()
90     });
91   }
92   let id = req.params.id;
93   let Data = {
94     nama_pemilik: req.body.nama_pemilik,
95     alamat: req.body.alamat,
96     no_hp: req.body.no_hp
97   }
98   connection.query(`update pemilik set ? where id_pemilik = ${id}`, Data, function (err, rows) {
99     if(err){
100       return res.status(500).json({
101         status: false,
102         message: 'Server Error',
103       })
104     }else {
105       return res.status(200).json({
106         status: true,
107         message: 'Update Success..!'
108       })
109     }
110   })
111 })
```

Memasukkan perubahan yang akan ditambahkan, jika tertulis succes maka data sudah berhasil terupdate menjadi data yang terbaru setelah di lakukan perubahan

http://localhost:3000/api/pemilik/update/1

Save

Send

PATCH

http://localhost:3000/api/pemilik/update/1

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"/>	nama_pemilik	ajeng			
<input checked="" type="checkbox"/>	alamat	jakarta			
<input checked="" type="checkbox"/>	no_hp	085732329729			

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

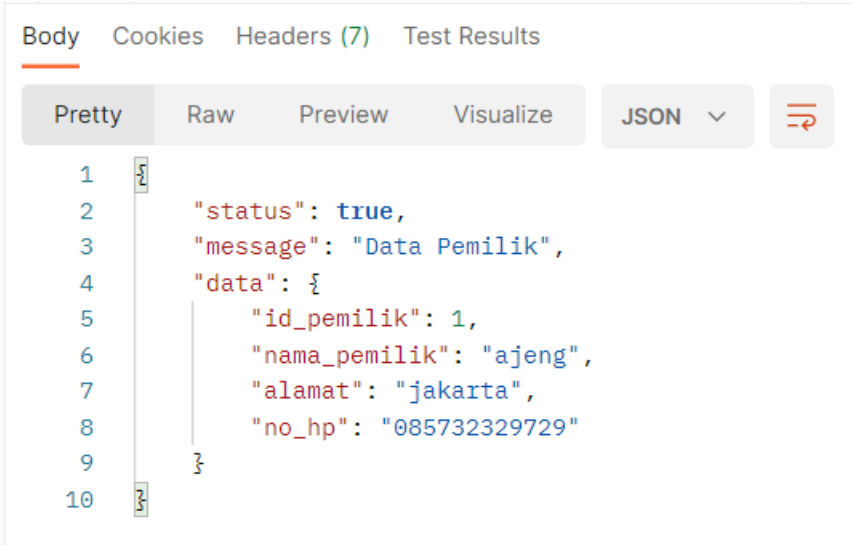
{

"status": true,

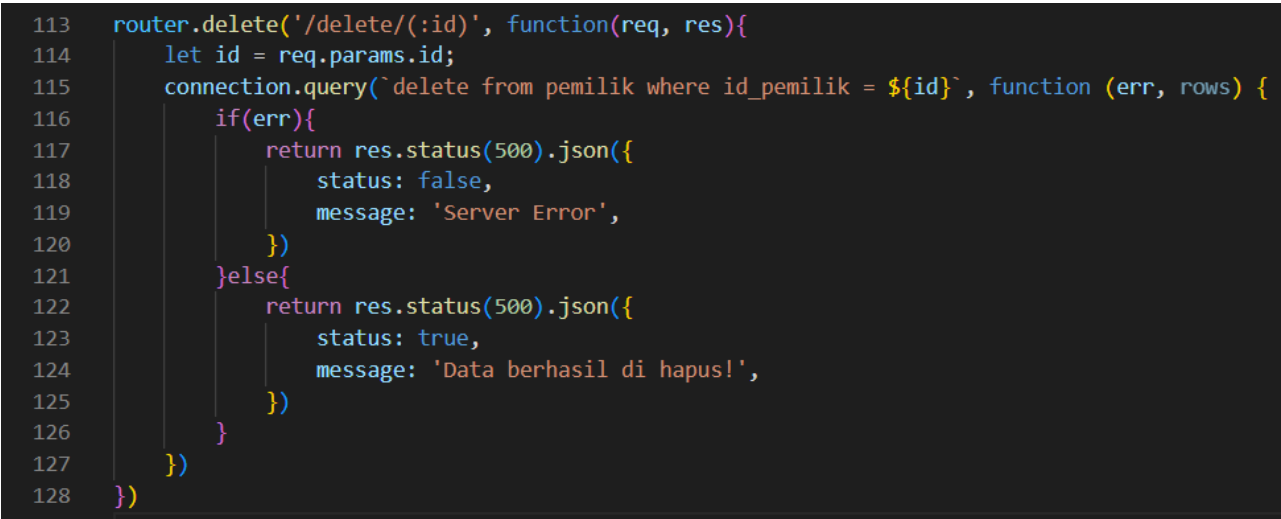
"message": "Update Success..!"

}

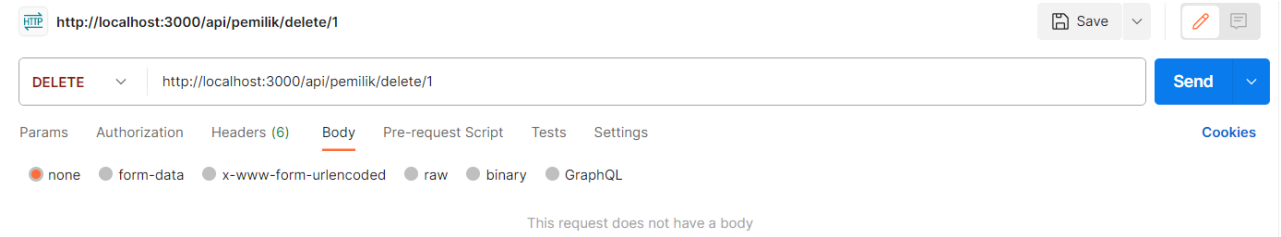
Maka dapat dicek kembali apakah data benar-benaar sudah terganti sesuai update terbaru atau belum



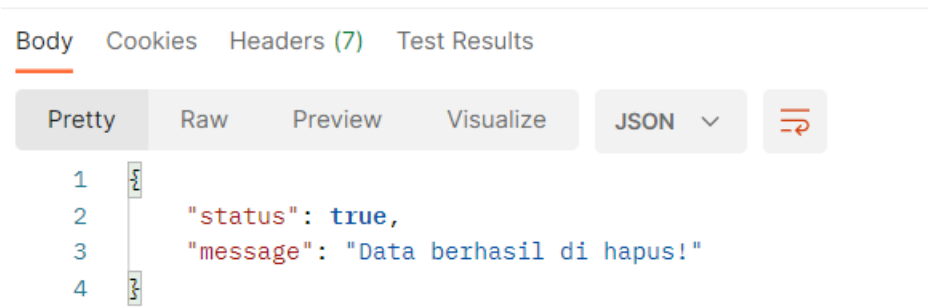
Script untuk melakukan hapus data atau delete sesuai id data yang sudah ada di database



Jangan lupa untuk memasukkan id data mana yang akan dihapus



Jika sudah berhasil menjalankan proses penghapusan maka akan muncul bahwa data sudah berhasil didelete / dihapus



Cek pada database, maka data yang dihapus akan hilang atau kosong

MEMBUAT CRUD DATA DPI

```
routes > JS dpi.js > ...
1  const express = require('express');
2  const router = express.Router();
3  const {body, validationResult } = require('express-validator');
4
5  const connection = require('../config/db.js');
6
7  > router.get('/', function (req, res){ ...
22  });
23
24  > router.post('/store', [ ...
52  })
53
54  > router.get('/:id', function (req, res) { ...
77  })
78
79  > router.patch('/update/:id', [ ...
107  })
108
109  > router.delete('/delete/:id', function(req, res){ ...
124  })
125
126  module.exports = router;
```

Script untuk dapat menghubungkan progra dengan database yang dibuat, memastikan data yang dimasukkan valid, menggunakan method get

```
7  router.get('/', function (req, res){
8      connection.query('select * from dpi order by id_dpi desc', function(err, rows){
9          if(err){
10             return res.status(500).json({
11                 status: false,
12                 message: 'Server Failed',
13             })
14         }else{
15             return res.status(200).json({
16                 status: true,
17                 message: 'Data DPI',
18                 data: rows
19             })
20         }
21     })
22 });
```

http://localhost:3000/api/dpi

Save

GET

http://localhost:3000/api/dpi

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

{

2

"status": true,

3

"message": "Data DPI",

4

"data": []

5

}

Script untuk melakukan create data, menggunakan method post untuk dapat menambahkan data

```
24 router.post('/store', [  
25   body('nama_dpi').notEmpty(),  
26   body('luas').notEmpty()  
27 ],(req, res) => {  
28   const error = validationResult(req);  
29   if(!error.isEmpty()){  
30     return res.status(422).json({  
31       error: error.array()  
32     })  
33   }  
34   let Data = {  
35     nama_dpi: req.body.nama_dpi,  
36     luas: req.body.luas  
37   }  
38   connection.query('insert into dpi set ?', Data, function(err, rows){  
39     if(err){  
40       return res.status(500).json({  
41         status: false,  
42         message: 'Server Error',  
43       })  
44     }else{  
45       return res.status(201).json({  
46         satus: true,  
47         message: 'Success..!',  
48         data: rows[0]  
49       })  
50     }  
51   })  
52 })
```

Memasukkan data sesuai dengan atribut dari database, jika tertulis succes maka data berhasil masuk

http://localhost:3000/api/dpi/store

Save

POST

http://localhost:3000/api/dpi/store

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQL

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	nama_dpi	sumenep			
<input checked="" type="checkbox"/>	luas	100,6			

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

{

2

"satus": true,

3


"message": "Success..!"

4

}

Mengecek apakah data sudah masuk dan dapat tertampil dengan menggunakan ide sebagai pemanggil, menggunakan method get

```
54 router.get('/:id)', function (req, res) {
55   let id = req.params.id;
56   connection.query(`select * from dpi where id_dpi = ${id}`, function (err, rows) {
57     if(err){
58       return res.status(500).json({
59         status: false,
60         message: 'Server Error',
61       })
62     }
63     if(rows.length <=0){
64       return res.status(404).json({
65         status: false,
66         message: 'Not Found',
67       })
68     }
69     else{
70       return res.status(200).json({
71         status: true,
72         message: 'Data DPI',
73         data: rows[0]
74       })
75     }
76   })
77 })
```

 http://localhost:3000/api/dpi/1

GET

http://localhost:3000/api/dpi/1

Send

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Cookies

Jika data yang dipanggil sesuai idnya sudah muncul dan sesuai maka dinyatakan sudah berhasil

BodyCookiesHeaders (7)Test Results

PrettyRawPreviewVisualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

```
1 {
2   "status": true,
3   "message": "Data DPI",
4   "data": {
5     "id_dpi": 1,
6     "nama_dpi": "sumenep",
7     "luas": 100
8   }
9 }
```

Script untuk dapat melakukan edit atau update data sesuai dengan id data mana yang dimasukkan, menggunakan method patch untuk menyimpan perubahan yang dibuat

```
79 router.patch('/update/:id', [
80   body('nama_dpi').notEmpty(),
81   body('luas').notEmpty()
82 ], (req,res) => {
83   const error = validationResult(req);
84   if(!error.isEmpty()){
85     return res.status(422).json({
86       error: error.array()
87     });
88   }
89   let id = req.params.id;
90   let Data = {
91     nama_dpi: req.body.nama_dpi,
92     luas: req.body.luas
93   }
94   connection.query(`update dpi set ? where id_dpi = ${id}`, Data, function (err, rows) {
95     if(err){
96       return res.status(500).json({
97         status: false,
98         message: 'Server Error',
99       })
100     }else {
101       return res.status(200).json({
102         status: true,
103         message: 'Update Success..!'
104       })
105     }
106   })
107 })
```

Memasukkan perubahan yang akan ditambahkan, jika tertulis succes maka data sudah berhasil terupdate menjadi data yang terbaru setelah di lakukan perubahan

http://localhost:3000/api/dpi/update/1

Save

PATCH

http://localhost:3000/api/dpi/update/1

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQL

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	nama_dpi	sumenep			
<input checked="" type="checkbox"/>	luas	50,8			

BodyCookiesHeaders (7)Test Results

PrettyRawPreviewVisualizeJSON

1

2

3

4

"status": true,

"message": "Update Success..!"

Maka dapat dicek kembali apakah data benar-benaar sudah terganti sesuai update terbaru atau belum

BodyCookiesHeaders (7)Test Results

PrettyRawPreviewVisualizeJSON

1

2

3

4

5

6

7

8

9

"status": true,

"message": "Data DPI",

"data": {

"id_dpi": 1,

"nama_dpi": "sumenep",

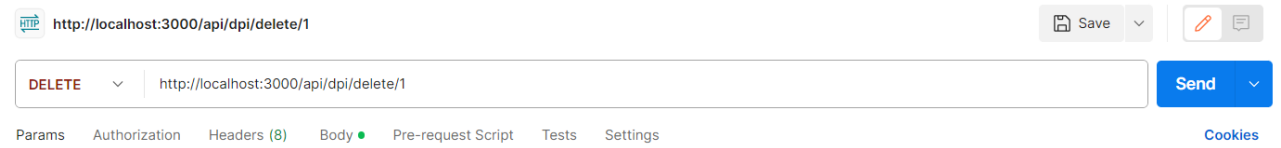
"luas": 50

}

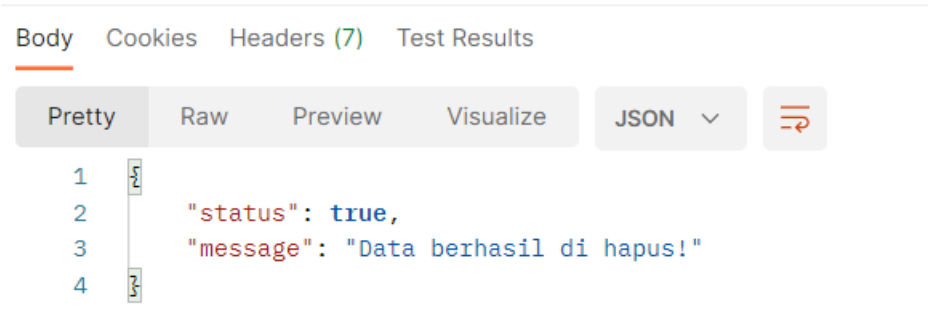
Script untuk melakukan hapus data atau delete sesuai id data yang sudah ada di database

```
109 router.delete('/delete/:id', function(req, res){
110   let id = req.params.id;
111   connection.query(`delete from dpi where id_dpi = ${id}`, function (err, rows) {
112     if(err){
113       return res.status(500).json({
114         status: false,
115         message: 'Server Error',
116       })
117     }else{
118       return res.status(500).json({
119         status: true,
120         message: 'Data berhasil di hapus!',
121       })
122     }
123   })
124 })
125
126 module.exports = router;
```

Jangan lupa untuk memasukkan id data mana yang akan dihapus



Jika sudah berhasil menjalankan proses penghapusan maka akan muncul bahwa data sudah berhasil didelete / dihapus



Cek pada database, maka data yang dihapus akan hilang atau kosong

MEMBUAT CRUD ALAT TANGKAP

```
routes > JS alat_tangkap.js > ...
1  const express = require('express');
2  const router = express.Router();
3  const {body, validationResult } = require('express-validator');
4
5  const connection = require('../config/db.js');
6
7  > router.get('/', function (req, res){...
22 });
23
24 > router.post('/store', [...
50 })
51
52 > router.get('/:id', function (req, res) {...
75 })
76
77 > router.patch('/update/:id', [...
103 })
104
105 > router.delete('/delete/:id', function(req, res){...
120 })
121
122 module.exports = router;
```

Script untuk dapat menghubungkan progra dengan database yang dibuat, memastikan data yang dimasukkan valid, menggunakan method get

```
7 router.get('/', function (req, res){
8   connection.query('select * from alat_tangkap order by id_alat_tangkap desc', function(err, rows){
9     if(err){
10      return res.status(500).json({
11        status: false,
12        message: 'Server Failed',
13      })
14    }else{
15      return res.status(200).json({
16        status: true,
17        message: 'Data alat_tangkap',
18        data: rows
19      })
20    }
21  })
22 });
```

http://localhost:3000/api/alat_tangkap

GET http://localhost:3000/api/alat_tangkap Send

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

Query Params

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

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

Script untuk melakukan create data, menggunakan method post untuk dapat menambahkan data

```
24 router.post('/store', [
25   body('nama_alat_tangkap').notEmpty(),
26 ],(req, res) => {
27   const error = validationResult(req);
28   if(!error.isEmpty()){
29     return res.status(422).json({
30       error: error.array()
31     })
32   }
33   let Data = {
34     nama_alat_tangkap: req.body.nama_alat_tangkap,
35   }
36   connection.query('insert into alat_tangkap set ?', Data, function(err, rows){
37     if(err){
38       return res.status(500).json({
39         status: false,
40         message: 'Server Error',
41       })
42     }else{
43       return res.status(201).json({
44         satus: true,
45         message: 'Success..!',
46         data: rows[0]
47       })
48     }
49   })
50 })
```

Memasukkan data sesuai dengan atribut dari database, jika tertulis succes maka data berhasil masuk

http://localhost:3000/api/alat_tangkap/store

Save

POSThttp://localhost:3000/api/alat_tangkap/storeSend

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

noneform-datax-www-form-urlencodedrawbinaryGraphQL

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	nama_alat_tangkap	jala			

BodyCookiesHeaders (7)Test Results

PrettyRawPreviewVisualizeJSON

1{

2 "satus": true,

3 "message": "Success..!"

4}

Mengecek apakah data sudah masuk dan dapat tertampil dengan menggunakan ide sebagai pemanggil, menggunakan method get

```
52 router.get('/:id', function (req, res) {
53   let id = req.params.id;
54   connection.query(`select * from alat_tangkap where id_alat_tangkap = ${id}`, function (err, rows) {
55     if(err){
56       return res.status(500).json({
57         status: false,
58         message: 'Server Error',
59       })
60     }
61     if(rows.length <=0){
62       return res.status(404).json({
63         status: false,
64         message: 'Not Found',
65       })
66     }
67     else{
68       return res.status(200).json({
69         status: true,
70         message: 'Data alat_tangkap',
71         data: rows[0]
72       })
73     }
74   })
75 })
```

http://localhost:3000/api/alat_tangkap/1

Save

GEThttp://localhost:3000/api/alat_tangkap/1Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

Query Params

	Key	Value	Description	...	Bulk Edit
--	-----	-------	-------------	-----	-----------

Jika data yang dipanggil sesuai idnya sudah muncul dan sesuai maka dinyatakan sudah berhasil

BodyCookiesHeaders (7)Test Results

PrettyRawPreviewVisualizeJSON

1{

2 "status": true,

3 "message": "Data alat_tangkap",

4 "data": {

5 "id_alat_tangkap": 1,

6 "nama_alat_tangkap": "jala"

7 }

8}

Script untuk dapat melakukan edit atau update data sesuai dengan id data mana yang dimasukkan, menggunakan method patch untuk menyimpan perubahan yang dibuat

```
77 router.patch('/update/:id', [
78   body('nama_alat_tangkap').notEmpty()
79 ], (req,res) => {
80   const error = validationResult(req);
81   if(!error.isEmpty()){
82     return res.status(422).json({
83       error: error.array()
84     });
85   }
86   let id = req.params.id;
87   let Data = {
88     nama_alat_tangkap: req.body.nama_alat_tangkap
89   }
90   connection.query(`update alat_tangkap set ? where id_alat_tangkap = ${id}`, Data, function (err, rows) {
91     if(err){
92       return res.status(500).json({
93         status: false,
94         message: 'Server Error',
95       })
96     }else {
97       return res.status(200).json({
98         status: true,
99         message: 'Update Success..!'
100       })
101     }
102   })
103 })
```

Memasukkan perubahan yang akan ditambahkan, jika tertulis succes maka data sudah berhasil terupdate menjadi data yang terbaru setelah di lakukan perubahan

http://localhost:3000/api/alat_tangkap/update/1

Save

PATCH

http://localhost:3000/api/alat_tangkap/update/1

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"/>	nama_alat_tangkap	jaring			

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

{

2

"status": true,

3

"message": "Update Success..!"

4

}

Maka dapat dicek kembali apakah data benar-benaar sudah terganti sesuai update terbaru atau belum

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

{

2

"status": true,

3

"message": "Data alat_tangkap",

4

"data": {

5

"id_alat_tangkap": 1,

6

"nama_alat_tangkap": "jaring"

7

}

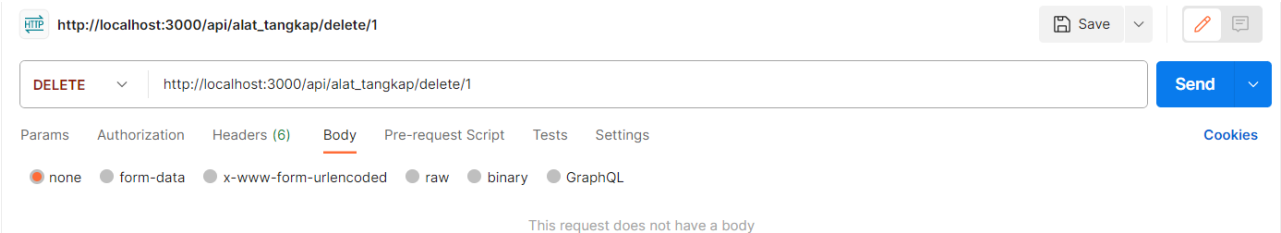
8

}

Script untuk melakukan hapus data atau delete sesuai id data yang sudah ada di database

```
105 router.delete('/delete/:id', function(req, res){
106   let id = req.params.id;
107   connection.query(`delete from alat_tangkap where id_alat_tangkap = ${id}`, function (err, rows) {
108     if(err){
109       return res.status(500).json({
110         status: false,
111         message: 'Server Error',
112       })
113     }else{
114       return res.status(500).json({
115         status: true,
116         message: 'Data berhasil di hapus!',
117       })
118     }
119   })
120 })
121
122 module.exports = router;
```

Jangan lupa untuk memasukkan id data mana yang akan dihapus



Jika sudah berhasil menjalankan proses penghapusan maka akan muncul bahwa data sudah berhasil didelete / dihapus



Cek pada database, maka data yang dihapus akan hilang atau kosong

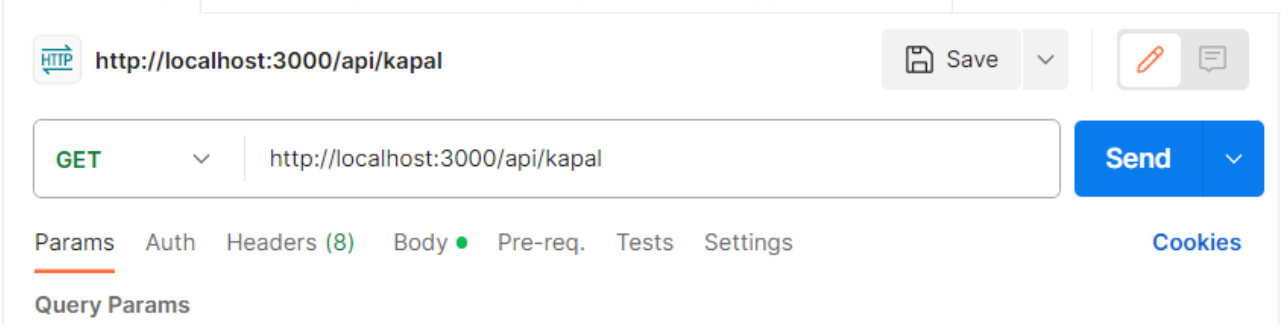
MEMBUAT CRUD KAPAL

```
routes > JS kapal.js > router.get("/") callback
1  const express = require('express');
2  const router = express.Router();
3  const {body, validationResult } = require('express-validator');
4
5  const connection = require('../config/db.js');
6
7  > router.get("/", (req, res) => {
25  });
26
27 > router.post( ...
66  );
67
68 > router.get("/:id", (req, res) => { ...
88  });
89
90 > router.patch( ...
134  );
135
136 > router.delete("/:id", (req, res) => { ...
155  });
156
157 module.exports = router;
```

Dikarenakan tabel kapal merupakan tabel utama yang memiliki relasi dengan ketiga tabel lainnya maka akan ada sedikit perbedaan penulisan scriptnya bukan lagi menggunakan select * from namaun menggunakan inner join yang akan menghubungkan relasi dari tabel lainnya ke tabel kapal

```
7 router.get("/", (req, res) => {
8   connection.query(
9     'SELECT k.nama_kapal, d.luas, p.nama_pemilik, a.nama_alat_tangkap FROM kapal as k INNER JOIN dpi as d ON k.id_dpi = d.id_dpi INNER JOIN
10    pemilik as p ON k.id_pemilik = p.id_pemilik INNER JOIN alat_tangkap as a ON k.id_alat_tangkap = a.id_alat_tangkap ',
11    (err, rows) => {
12      if (err) {
13        return res.status(500).json({
14          status: false,
15          message: "Internal Server Error",
16        });
17      } else {
18        return res.status(200).json({
19          status: true,
20          message: "Data kapal",
21          payload: rows,
22        });
23      }
24    });
25  });
```

Karena pada soal yang ingin ditampilkan adalah nama kapal, nama pemilik dari tabel pemilik, nama alat tangkap dari tabel alat tangkap dan luas dari tabel dpi maka fungsi dari ide hanyalah untuk pemanggil yangakan menghubungkan antar tabel namun yang akan terbaca dan tampil adalah nama dari id tiap tabel yang dipilih



Script untuk dapat menghubungkan program dengan database yang dibuat, memastikan data yang dimasukkan valid, menggunakan method get


```
Body
Pretty Raw Preview Visualize
1 {
2   "status": true,
3   "message": "Data kapal",
4   "payload": []
5 }
```

Script untuk melakukan create data, menggunakan method post untuk dapat menambahkan data



```
27 router.post(  
28   "/store",  
29   [  
30     body("nama_kapal").notEmpty(),  
31     body("id_pemilik").notEmpty(),  
32     body("id_dpi").notEmpty(),  
33     body("id_alat_tangkap").notEmpty(),  
34   ],  
35   (req, res) => {  
36     const error = validationResult(req);  
37  
38     // if validation failed  
39     if (!error.isEmpty()) {  
40       return res.status(422).json({  
41         error: error,  
42       });  
43     }  
44  
45     let data = {  
46       nama_kapal: req.body.nama_kapal,  
47       id_pemilik: req.body.id_pemilik,  
48       id_dpi: req.body.id_dpi,  
49       id_alat_tangkap: req.body.id_alat_tangkap,  
50     };
```

```
51     connection.query("INSERT INTO kapal set ? ", data, (err, rows) => {  
52       if (err) {  
53         return res.status(500).json({  
54           status: false,  
55           message: "Internal Server Error",  
56         });  
57       } else {  
58         return res.status(201).json({  
59           status: true,  
60           message: "Data berhasil ditambahkan",  
61           payload: data,  
62         });  
63       }  
64     });  
65   }  
66 );
```

Memasukkan data sesuai dengan atribut dari database, namun dikarenakan tabel kapal memiliki atribut yang berfungsi sebagai foreign key maka data yang dimasukkan adalah data yang sudah ada pada masing masing tabel yang berelasi dengan cara memasukkan id dari tiap data tabel masing-masing, jika tertulis succes maka data berhasil masuk

 http://localhost:3000/api/kapal/store

Save



POST

http://localhost:3000/api/kapal/store

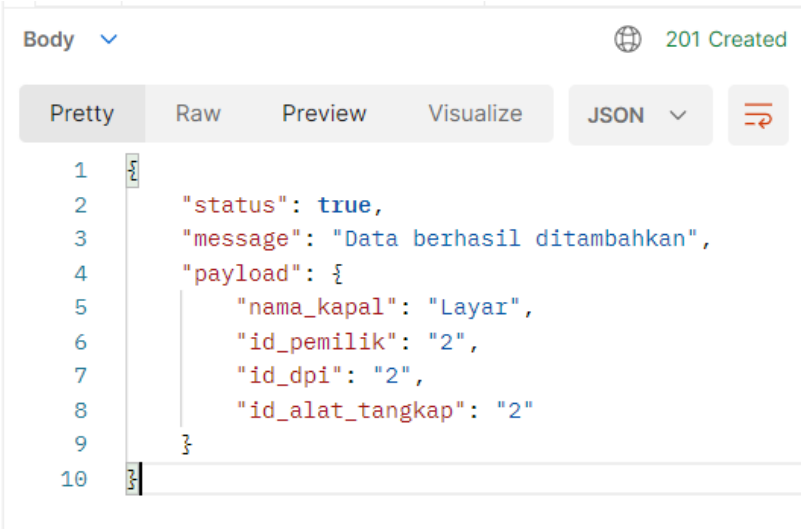
Send

ParamsAuthHeaders (8)BodyPre-req. Tests SettingsCookies

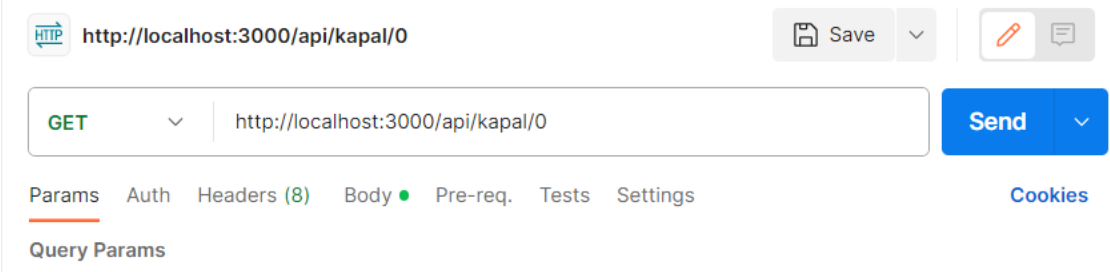
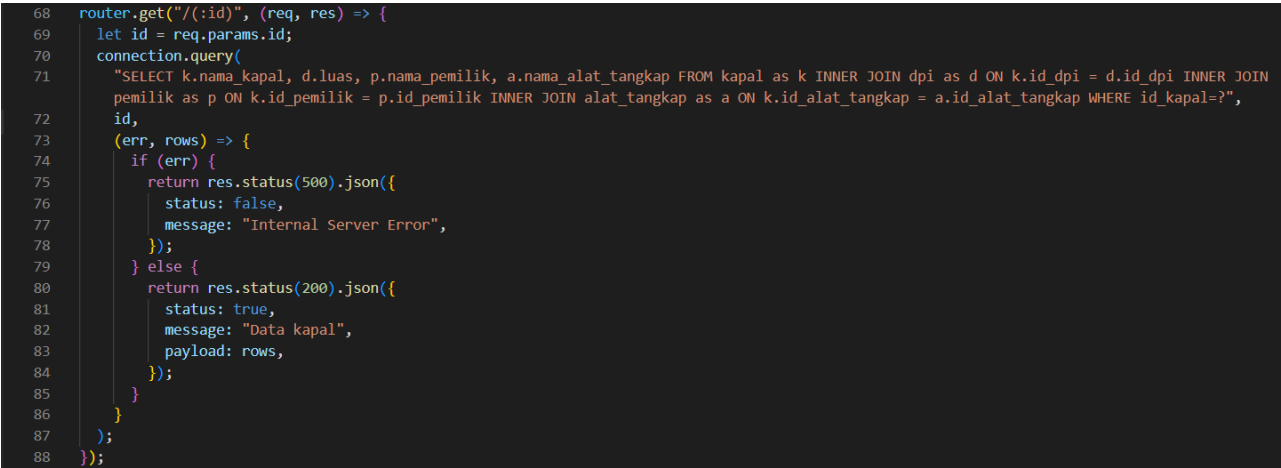
x-www-form-urlencoded

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	nama_kapal	Layar			
<input checked="" type="checkbox"/>	id_pemilik	2			
<input checked="" type="checkbox"/>	id_dpi	2			
<input checked="" type="checkbox"/>	id_alat_tangkap	2			

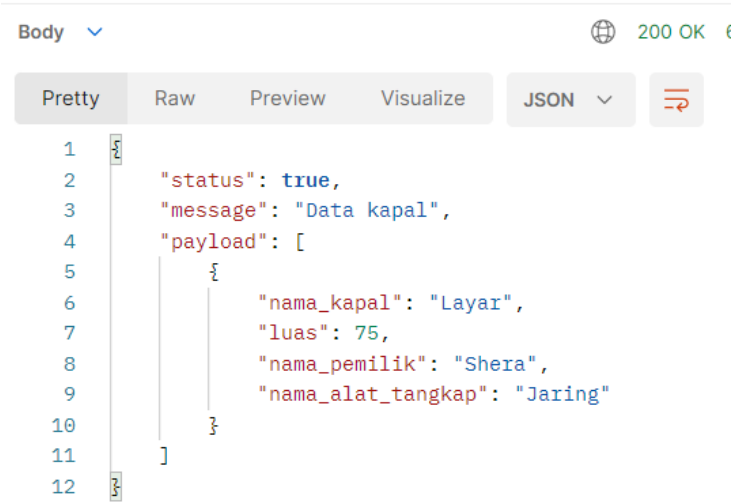
Disini akan langsung muncul data yang baru saja ditambahkan



Mengecek apakah data sudah masuk dan dapat tertampil dengan menggunakan ide sebagai pemanggil, menggunakan method get, disini script yang digunakan masih tetap sama dengan script get diawal namun bedanya karena sudah ada data yang masuk di database maka menambahkan WHERE ide kapal untuk dapat mengecek isi dari data



Jika data yang dipanggil sesuai idnya sudah muncul dan sesuai maka dinyatakan sudah berhasil, disini tampilan sudah berbeda, akan langsung tertampil nama kapal, luas, nama pemilik dan nama alat tangkap dari data yang sudah ada didatabase, tidak lagi menampilkan ide nya saja namun yang ditampilkan adalah namanya.



Script untuk dapat melakukan edit atau update data sesuai dengan id data mana yang dimasukkan, menggunakan method patch untuk menyimpan perubahan yang dibuat

```
90 router.patch(  
91   "/update/:id",  
92   [  
93     body("nama_kapal").notEmpty(),  
94     body("id_pemilik").notEmpty(),  
95     body("id_dpi").notEmpty(),  
96     body("id_alat_tangkap").notEmpty(),  
97   ],  
98   (req, res) => {  
99     const error = validationResult(req);  
100  
101     // if validation failed  
102     if (!error.isEmpty()) {  
103       return res.status(422).json({  
104         error: error,  
105       });  
106     }  
107  
108     let id = req.params.id;  
109     let data = {  
110       nama_kapal: req.body.nama_kapal,  
111       id_pemilik: req.body.id_pemilik,  
112       id_dpi: req.body.id_dpi,  
113       id_alat_tangkap: req.body.id_alat_tangkap,  
114     };  
115  
116     connection.query(  
117       "UPDATE kapal set ? WHERE id_kapal=?",  
118       [data, id],  
119       (err, rows) => {  
120         if (err) {  
121           return res.status(500).json({  
122             status: false,  
123             message: "Internal Server Error",  
124           });  
125         } else {  
126           return res.status(200).json({  
127             status: true,  
128             message: "Data berhasil diupdate",  
129             payload: data,  
130           });  
131         }  
132       });  
133     }  
134   );
```

Memasukkan perubahan yang akan ditambahkan, jika tertulis succes maka data sudah berhasil terupdate menjadi data yang terbaru setelah di lakukan perubahan

HTTP

http://localhost:3000/api/kapal/update/0

Save

PATCH

http://localhost:3000/api/kapal/update/0

Send

ParamsAuthHeaders (8)BodyPre-req. Tests SettingsCookies

x-www-form-urlencoded

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	nama_kapal	Layar			
<input checked="" type="checkbox"/>	id_pemilik	2			
<input checked="" type="checkbox"/>	id_dpi	3			
<input checked="" type="checkbox"/>	id_alat_tangkap	2			

```
Body 200 OK 7
Pretty Raw Preview Visualize JSON
{
  "status": true,
  "message": "Data berhasil diupdate",
  "payload": {
    "nama_kapal": "Layar",
    "id_pemilik": "2",
    "id_dpi": "3",
    "id_alat_tangkap": "2"
  }
}
```

Maka dapat dicek kembali apakah data benar-benaar sudah terganti sesuai update terbaru atau belum

```
Body 200 OK 226
Pretty Raw Preview Visualize JSON
{
  "status": true,
  "message": "Data kapal",
  "payload": [
    {
      "nama_kapal": "Layar",
      "luas": 100,
      "nama_pemilik": "Shera",
      "nama_alat_tangkap": "Jaring"
    }
  ]
}
```

Script untuk melakukan hapus data atau delete sesuai id data yang sudah ada di database

```
136 router.delete("/delete/(:id)", (req, res) => {
137   let id = req.params.id;
138   connection.query(
139     "DELETE FROM kapal WHERE id_kapal=?",
140     id,
141     (err, rows) => {
142       if (err) {
143         return res.status(500).json({
144           status: false,
145           message: "Internal Server Error",
146         });
147       } else {
148         return res.status(200).json({
149           status: true,
150           message: "Data kapal berhasil didelete",
151         });
152       }
153     }
154   );
155 });
```

Jangan lupa untuk memasukkan id data mana yaang akan dihapus

HTTP

http://localhost:3000/api/kapal/delete/0

Save

DELETE

http://localhost:3000/api/kapal/delete/0

Send

Params

Auth

Headers (8)

Body

Pre-req.

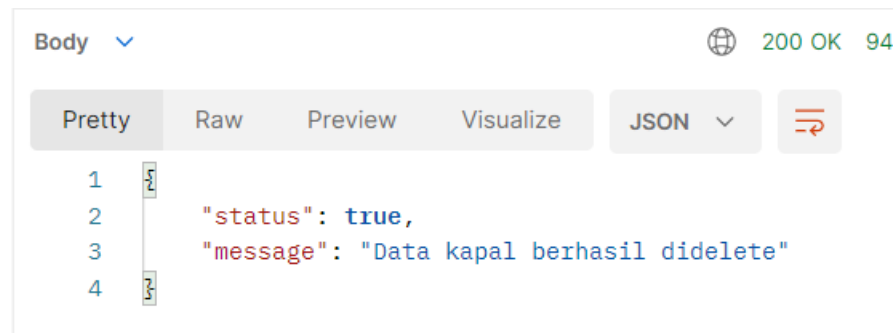
Tests

Settings

Cookies

Query Params

Jika sudah berhasil menjalankan proses penghapusan maka akan muncul bahwa data sudah berhasil didelete / dihapus



KESIMPULAN:

setelah melakukan beberapa percobaan pada tahap ini dapat dipelajari cara membuat CRUD secara express dan mengecek hasil dengan menggunakan beberapa method yang ada pada aplikasi postman. Kita juga dapat membuat tabel dengan memperhatikan ERD yang dibuat dan merepresentasikannya kedalam database yang baik dan benar. Penggunaan relasi juga sangat penting agar data tetap terdata dengan valid dan aplikasi atau program dapat dijalankan secara sempurna.

Link github: <https://github.com/sherazahra/latihan9.git>