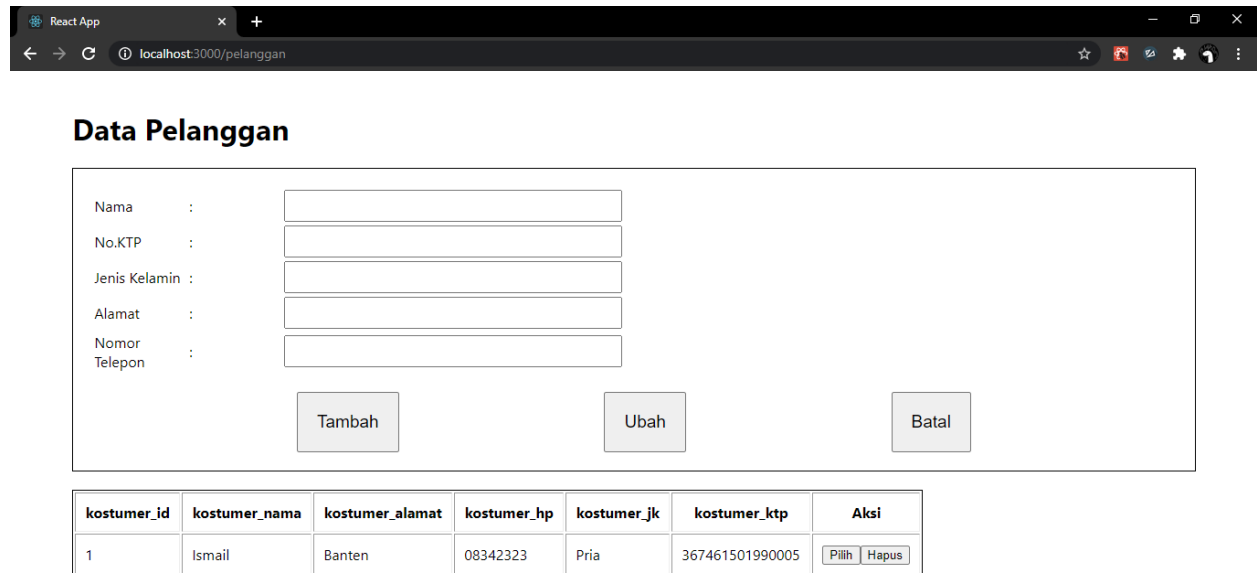


Dokumentasi Page Pelanggan



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/pelanggan'. The page title is 'Data Pelanggan'. Below the title is a form with five input fields labeled 'Nama', 'No.KTP', 'Jenis Kelamin', 'Alamat', and 'Nomor Telepon'. Each field has a corresponding label and a colon. Below the form are three buttons: 'Tambah', 'Ubah', and 'Batal'. Below the buttons is a table with the following data:

kostumer_id	kostumer_nama	kostumer_alamat	kostumer_hp	kostumer_jk	kostumer_ktp	Aksi
1	Ismail	Banten	08342323	Pria	367461501990005	<button>Pilih</button> <button>Hapus</button>

Pada Page Pelanggan Terdapat Form Pelanggan dan Tabel yang berisi data pelanggan, Tiap inputan pada form pelanggan datanya merupakan sebuah state pada react, dengan menggunakan react hooks useState data disimpan pada state tersebut dan data yang ada di state bias di ubah dengan menggunakan setState-nya

```
const [Nama, setNama] = useState('');
const [KTP, setKTP] = useState('');
const [ALamat, setALamat] = useState('');
const [Telp, setTelp] = useState('');
const [JK, setJK] = useState('');
const [id, setId] = useState('');
```

jadi nilai awal pada state tersebut merupakan string kosong yang nanti akan terisi jika diinput atau tombol aksi pilih yang ada ditabel atau bias juga diisi secara manual.

Untuk Tombol Pilih, buttonnya diberi properti onClick agar ketika diklik bisa menjalankan sebuah fungsi berikut adalah fungsinya :

```
const tambah = (e) => {
  e.preventDefault();
  const data = {
    kostumer_hp: Telp,
    kostumer_jk: JK,
```

```
kostumer_ktp: KTP,
kostumer_nama: Nama,
kostumer_alamat: ALamat,
};

axios.post('http://localhost:3001/addKostumer', data);
alert('Berhasil');
setId('');
};
```

Terdapat Object data yang berisi value pelanggan, data ini akan di kirim ke backend menggunakan axios, karena mengirim data maka dari itu diberi Post, lalu ketika berhasil tampilkan alert dan set state id menjadi string kosong, berikut adalah algoritma kode addKostumer pada backend

```
addKostumer: async (req, res) => {
  await table('kostumer').insert(req.body).then(console.log);
  res.send('sukses');
}
```

Menggunakan library knex.js, knex mampu untuk berkomunikasi dengan mysql menggunakan node js, fungsi diatas akan mengirim data dari frontend ke database bisa dilihat table kostumer akan diinsert oleh data req.body, req.body artinya body request atau data yang dari frontend. Untuk Mengubah data juga seperti ini tetapi jika menambah adalah POST kalau ingin update berarti PUT

```
const ubah = (id) => {
  console.log(id);

  const data = {
    kostumer_hp: Telp,
    kostumer_jk: JK,
    kostumer_ktp: KTP,
    kostumer_nama: Nama,
    kostumer_alamat: ALamat,
  };
  axios.put(`http://localhost:3001/putKostumer/${id}`, data);
};
```

Tentu saja kita harus memberi tahu backend data yang mana yang akan di edit dengan mengirim id dari data yang ingin diedit, berikut adalah backend route dan controllernya

```
app.put('/putKostumer/:id', controller.put.putKostumer);
```

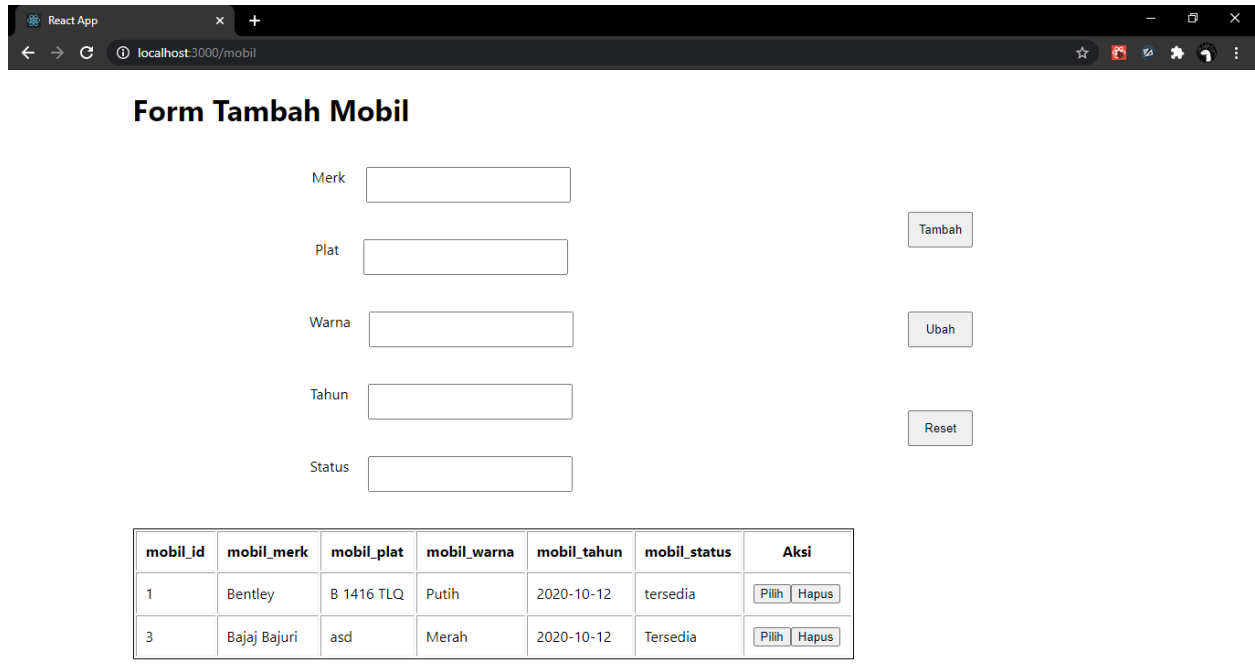
```
putKostumer: async (req, res) => {  
    await table('kostumer').where('kostumer_id', req.params.id).update(req.body);  
},
```

Menggunakan knex.js table kostumer dimana kostumer_id sama dengan req.params.id, artinya adalah request parameter id, jadi id disini adalah apapun yang kita beri pada front end

Black Box Testing Pelangan

No	Skenario Pengujian	Hasil Yang diharapkan	Kesimpulan
1	Memencet Tombol Hapus	Menghapus langsung data yang dipilih	Valid
2	Memencet Tombol Pilih	Meng-auto refill form, dengan data yang dipilih	Valid
3	Memencet Tombol Batal	Mengreset value yang ada di form	Tidak Valid
4	Mengisi Form lalu Mengklik Tombol Tambah	Data masuk ke database lalu table data akan bertambah data yang barusa ditambah	Valid
5	Memilih data lalu mengubah datanya di inputan	Data yang dipilih akan terupdate di table	Valid

Dokumentasi Page Mobil



The screenshot shows a web application running on localhost:3000/mobil. The page is titled 'Form Tambah Mobil'. It contains five input fields for car details: Merk, Plat, Warna, Tahun, and Status. To the right of these fields are three buttons: 'Tambah' (Add), 'Ubah' (Edit), and 'Reset'. Below the form is a table with the following data:

mobil_id	mobil_merk	mobil_plat	mobil_warna	mobil_tahun	mobil_status	Aksi
1	Bentley	B 1416 TLQ	Putih	2020-10-12	tersedia	<button>Pilih</button> <button>Hapus</button>
3	Bajaj Bajuri	asd	Merah	2020-10-12	Tersedia	<button>Pilih</button> <button>Hapus</button>

Page Mobil memiliki kesamaan dengan Pelanggan bedanya hanya saja tabelnya berbeda dan route backendnya juga berbeda. Sama seperti page pelanggan, page mobil juga mempunyai state untuk tiap inputnya, tentu saja juga menggunakan useState react hooks

```
const [mobil_id, setMobil_id] = useState('');
const [mobil_plat, setMobil_plat] = useState('');
const [mobil_warna, setMobil_warna] = useState('');
const [mobil_tahun, setMobil_tahun] = useState('');
const [mobil_status, setMobil_status] = useState('');
const [mobil_merk, setMobil_merk] = useState('');
```

lalu data dari state inilah jika tombol tambah dipencet akan dikirim ke back end lalu backend akan mengirimnya ke database menggunakan knex.js berikut adalah kodenya

```
const tambah = (e) => {
  e.preventDefault();
  const data = {
    mobil_plat: mobil_plat,
    mobil_warna: mobil_warna,
    mobil_tahun: mobil_tahun,
    mobil_status: mobil_status,
    mobil_merk: mobil_merk,
```

```
};

axios.post('http://localhost:3001/addMobil', data);
alert('Berhasil');
setMobil_id('');
};
```

Sama seperti pelanggan tetapi bedanya pada data yang dikirim dan route endpointnya

```
addMobil: async (req, res) => {
  await table('mobil').insert(req.body);
  res.send('sukses');
},
```

Untuk mengedit juga sama dengan pelanggan menggunakan request PUT berikut adalah kode front end dan backendnya

```
const ubah = (id) => {
  const data = {
    mobil_plat: mobil_plat,
    mobil_warna: mobil_warna,
    mobil_tahun: mobil_tahun,
    mobil_status: mobil_status,
    mobil_merk: mobil_merk,
  };
  axios.put(`http://localhost:3001/putMobil/${id}`, data);
};
```

```
app.put('/putMobil/:id', controller.put.putMobil);
```

```
putMobil: async (req, res) => {
  await table('mobil').where('mobil_id', req.params.id).update(req
  .body);
},
```

Menggunakan knex.js table mobil dimana mobil_id sama dengan req.params.id, artinya adalah request parameter id, jadi id disini adalah apapun yang kita beri atau oper pada front end

Black Box Testing Mobil

No	Skenario Pengujian	Hasil Yang diharapkan	Kesimpulan
1	Memencet Tombol Hapus	Menghapus langsung data yang dipilih	Valid
2	Memencet Tombol Pilih	Meng-auto refill form, dengan data yang dipilih	Valid
3	Memencet Tombol Reset	Mengreset value yang ada di form	Valid
4	Mengisi Form lalu Mengklik Tombol Tambah	Data masuk ke database lalu table data akan bertambah data yang barusa ditambah	Valid
5	Memilih data lalu mengubah datanya di inputan, lalu memencet tombol ubah	Data yang dipilih akan terupdate di table	Valid

Dokumentasi Login Page



username

password

Login

Terdapat Input Username dan password yang menerima value string, input pada username bertipe name dan input pada password bertipe password agar ketika user menginput, input box valuenya akan muncul symbol, dan terdapat login button yang ketika inputan dimasukin dengan benar maka akan langsung menjalankan function, function tambah itu berisi

```
e.preventDefault();
const data = {
  username: e.target[0].value,
  password: e.target[1].value,
};
let keterangan = await axios
  .post('http://localhost:3001/getAuth', data)
  .then((res) => res.data);

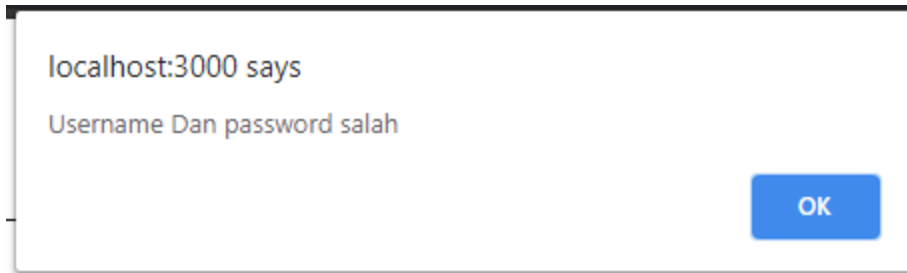
if (keterangan.ket === true) {
  history.push('/mobil');
} else {
  alert('Username Dan password salah');
}
```

```
getAuth: async (req, res) => {
  let { username, password } = req.body;
  const query = table.select('admin_username', 'admin_password').f
  rom('admin');
  const result = await query.then((data) => data);
  if (result[0].admin_username === username && result[0].admin_pas
  sword === password) {
    console.log('kelar');

    res.json({ ket: true });
  } else {
    console.log('Tidak kelar');

    res.json({ ket: false });
  }
},
```

e.preventDefault berfungsi untuk menghilangkan Default browser untuk merefresh browser ketika form disubmit, jadi axios akan mengirim data ke backend, lalu backend berkomunikasi via controller getAuth untuk mendapatkan semua data yang ada di table admin, lalu akan dikomparasi satu satu jika ada yang match maka backend akan mengirim respon object berbentuk JSON , yang berisi keterangan true atau false, jika true langsung diarahkan ke page mobil jika false tampilkan pesan kesalahan

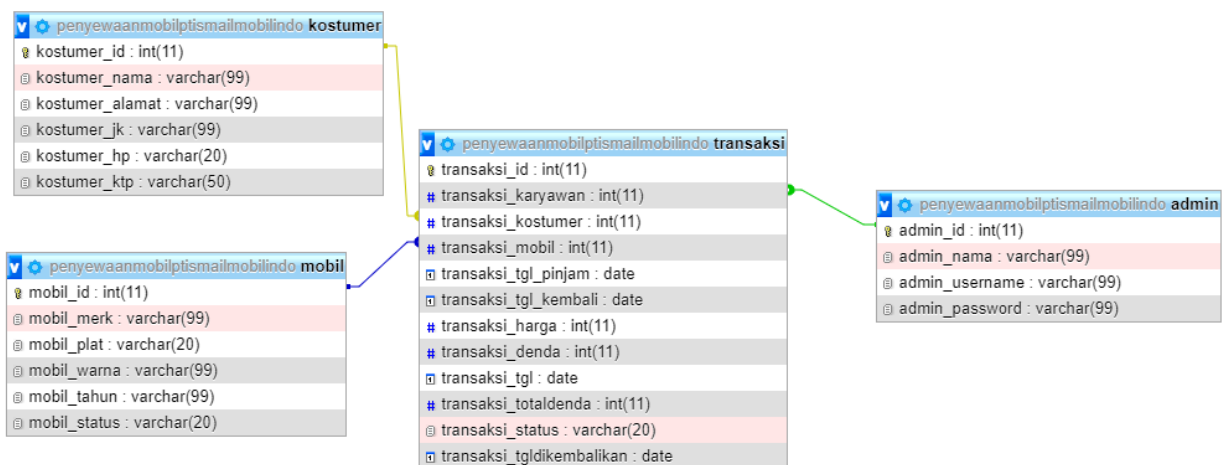


BlackBox testing Login Page

No	Skenario Pengujian	Hasil Yang diharapkan	Kesimpulan
1	Menginput username dan password dengan benar	Langsung mengarahkan user ke page mobil	Valid
2	Menginput username salah dan password yang benar	Muncul pesan kesalahan	valid
3	Menginput username benar dan password yang salah	Muncul pesan kesalahan	valid
4	Memencet Button Tanpa diisi input	Muncul pesan kesalahan	valid

Dokumentasi Database

Relasi pada table bisa dilihat pada gamabr berikut ini



Ismail

171011402364

mobil_id	mobil_merk	mobil_plat	mobil_warna	mobil_tahun	mobil_status
1	Bentley	B 1416 TLQ	Putih	2020-10-12	tersedia
3	Bajaj Bajuri	asd	Merah	2020-10-12	Tersedia

kostumer_id	kostumer_nama	kostumer_alamat	kostumer_jk	kostumer_hp	kostumer_ktp
1	Ismail	Banten	Pria	08342323	367461501990005

admin_id	admin_nama	admin_username	admin_password
1	ismail	admin	admin