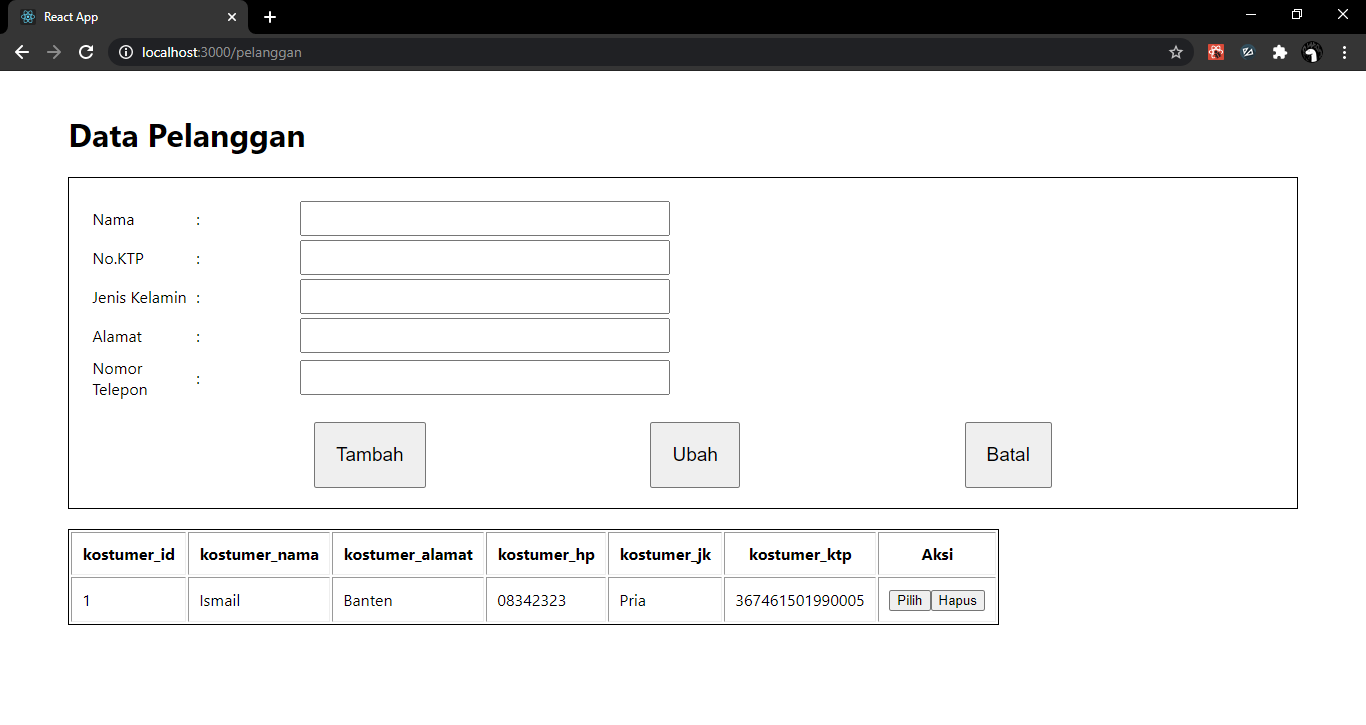
# Dokumentasi Page Pelanggan



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 mengguakan 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 ditable 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);

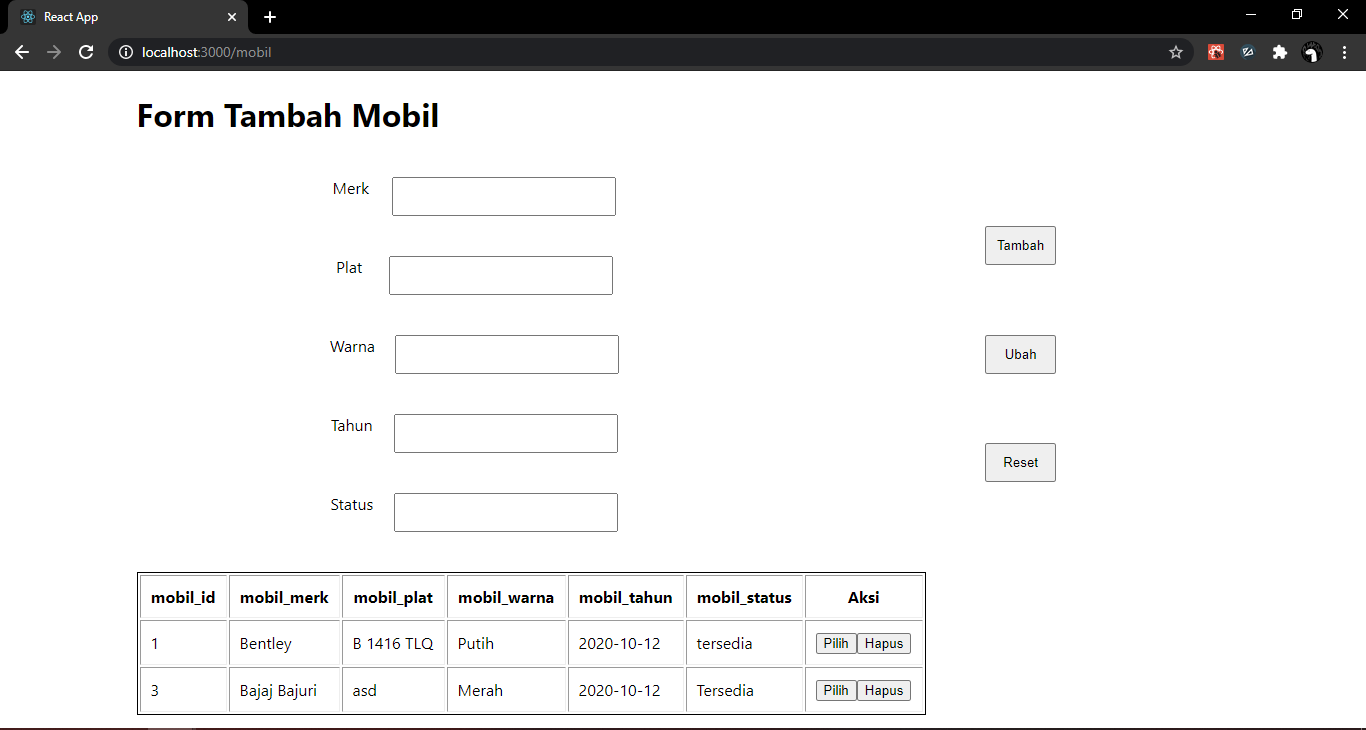
    },

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



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);

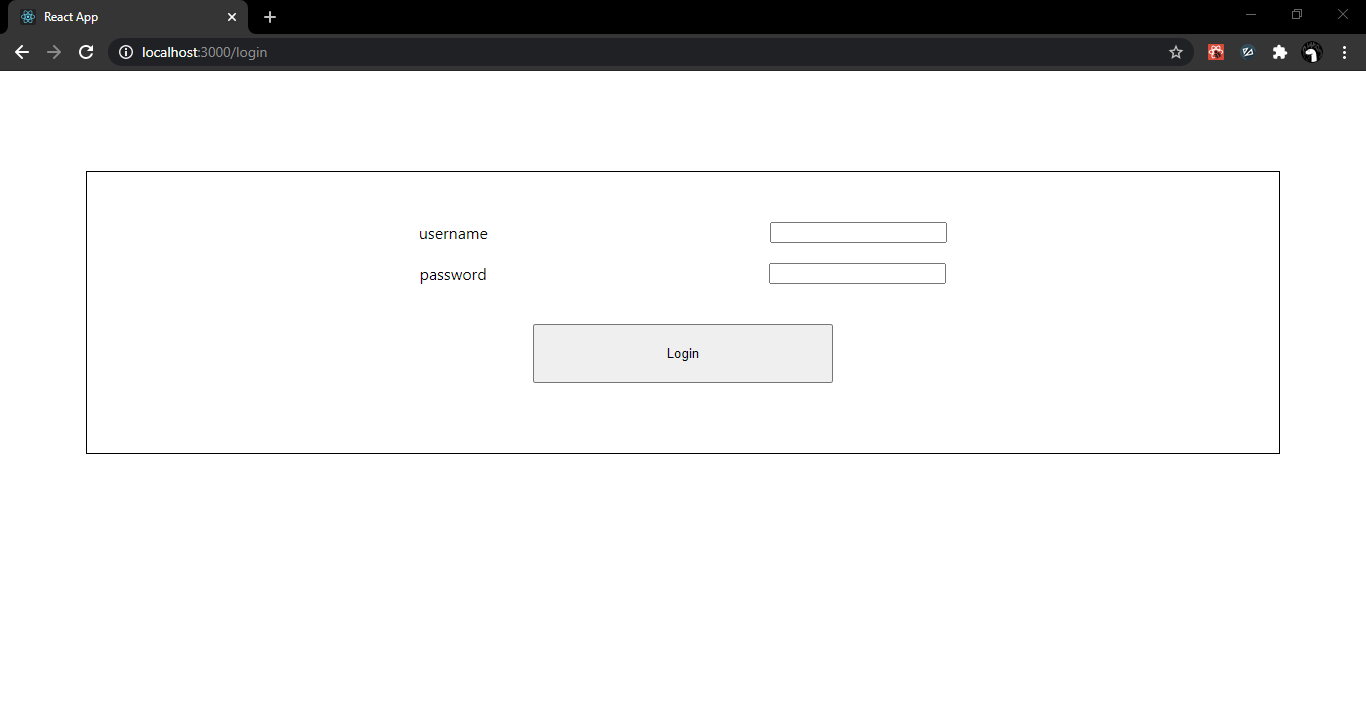
    },

Menggunakan knex.js table mobil dimana molbil\_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



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').from('admin');

      const result = await query.then((data) => data);

      if (result[0].admin\_username === username && result[0].admin\_password === password) {

        console.log('kelar');

        res.json({ ket: true });

      } else {

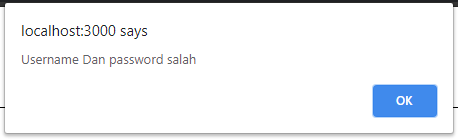
        console.log('Tidak kelar');

        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

