

## DAFTAR PUSTAKA

- Adi, L., Akbar, R. J., & Khotimah, W. N. (2017). Platform E-learning untuk Pembelajaran Pemrograman Web Menggunakan Konsep Progressive Web Apps . *Jurnal Teknik ITS*, A579.
- Ala'a Al-Shaikh, A. S. (2017). *Evaluating IndexedDB Performance on Web Browsers*. Department of Computer Science, University of Jordan: International Conference on Information Technologi (ICIT).
- Dawood, R., Qiana, S. F., & Muchallil, S. (2014). Kelayakan Raspberry Pi sebagai Web Server : Perbandingan Kinerja Nginx, Apache, dan Lightppd pada Platform Raspberry Pi. *Jurnal Rekayasa Elektrika*, 11 (1), 25-29.
- Fajrin, R. (2017). Pengembangan Sistem Informasi Geografis Berbasis Node.js untuk Pemetaan Mesin dan Tracking Engineer dengan Pemanfaatan Geolocation pada PT IBM Indonesia. *Jurnal Informatika Vol. 11*, 41.
- Fauzy, A. A. (2019). Analisis Performa dan Perancangan *Website* Progressive Web App dengan Library AngularFire2 dalam Pengembangan Website E-Commerce.
- K, Y. (2018, May 10). *16 Bahasa Pemrograman untuk Membuat Website*. Retrieved from NiagahosterBlog:  
<https://www.niagahoster.co.id/blog/bahasa-pemrograman/>
- Kumar, A., & Kumar, R. S. (2016). Comparative Analysis of AngularJS and ReactJS. *International Journal of Latest Trends in Engineering and Technology*, 226.
- Lavarino, D., & Yustanti, W. (2016). Rancang Bangun E-Voting Berbasis Website di Universitas Negeri Surabaya. *Jurnal Manajemen Informatika*, 73.
- Nainggolan, L. N. (2019). *Analisis Teknologi Offline First App Pasca Bencana pada Rumah Sakit menggunakan React JS*. Universitas Hasanuddin.

- Nawan Tutu Syah Lampah, E. B. (2018). *Aplikasi Asesmen Anak Berkebutuhan Khusus di SLB Rafaha Arjasari menggunakan Progressive Web App*. Universitas Komputer Indonesia: Program Studi Teknik Informatika.
- Permana, T. D. (2014). Sistem Monitoring Menggunakan Mini PC Raspberry Pi. 6-12.
- Putra, A. D., Yahya, W., & Bhawiyuga, A. (2019). Analisis Kinerja dan Konsumsi Sumber Daya Aplikasi Web Server pada platform Raspberry Pi. *Jurnal Pengembangan Teknologi Infromasi dan Ilmu Komputer*, 3516.
- Shenoy, K., Bhokare, P., & Pai, U. (2013). FOG Computing Future of Cloud Computing. *International Journal of Science and Research*, 55-56.
- Sidharta, S. M. (2017, September 16). *Progressive Web Apps*. Retrieved from Binus Malang: <http://binus.ac.id/malang/2017/09/progressive-web-apps/>
- Sidik, B. (2011). *JavaScript*. Bandung: Informatika.
- Suliyanti, W. N. (2019). Studi Literatur Basis Data SQL dan NoSQL. *Jurnal Kilat*, 49-50.
- Tamire, W. T. (2016). HTML5 and Its Capability to Develop Offline Web Applications. 7.
- Tung, K. (2018). Developing a frontend application using ReactJS and Redux. 20-22.

## LAMPIRAN

### LAMPIRAN 1

Hasil wawancara analisis kebutuhan dengan pihak IKM

No.	Pertanyaan	Jawaban
1	Bagaimanakah alur kerja pada sistem yang sedang berjalan pada industri ?	Saat ini, usaha kami ada 3 bagian yaitu, produksi, keuangan, dan pemasaran.  Saya sendiri menangani bagian keuangan, pembuatan bibit dan pada saat panen, kemudian dua orang di bagian pembuatan baglog hingga panen, dan bagian pemasaran ditangani oleh teman saya terkait pembuatan konten promosi.
2	Untuk bagian keuangan, data apa saja yang dicatat ?  Dan bagaimana pencatatan yang dilakukan selama ini ?	Untuk saat ini, kami mencatat data-data pemasukan dan pengeluaran.  Kami membuat pencatatan dalam sebuah buku pencatatan.
3	Untuk bagian pemasukan, apa saja yang dicatat ?	Data produk yang terjual, yaitu jamur, bibit, baglog.

4	Untuk bagian pengeluaran, apa saja yang dicatat ?	Berapa jumlah pengeluaran dan keterangan pengeluaran tersebut untuk apa.
5	Bagaimana jika buku pencatatannya hilang atau rusak ?	Jika buku pencatatan hilang, maka data-data pemasukan dan pengeluaran akan hilang sehingga mempengaruhi pencatatan keuangan selanjutnya dan kita tidak bisa mengetahui keuntungan yang dihasilkan maupun kerugian secara nyata.
6	Sistem seperti apakah yang diinginkan untuk membantu bagian keuangan ?	Tentu sistem yang dapat membantu dalam proses pencatatan pemasukan dan pengeluaran. Dan data-data tersimpan dengan aman.

7	<p>Apakah bersedia jika sistem yang akan dibentuk berupa sistem berbasis <i>website</i> ?</p> <p>Sistem berbasis <i>website</i> lebih fleksibel untuk diperluas dan dikembangkan nantinya.</p> <p>Kemudian sistem ini nantinya akan dirancang sehingga bisa diakses meskipun jaringan tidak bagus. Jadi anda tetap bisa melakukan proses input data meski koneksi jaringan anda tidak stabil.</p>	<p>Kenapa sistem berbasis <i>website</i> ?</p> <p>Jika seperti itu saya setuju.</p>
8	<p>Apakah bersedia memberikan data-data pemasukan dan pengeluaran usaha anda untuk keperluan penelitian dan pembuatan sistem ?</p>	<p>Saya bersedia.</p>

## LAMPIRAN 2

Hasil pengujian beta konsep PWA

1 : <i>Loading Page</i> yang cepat pada saat mengakses web sistem informasi industri kecil dan menengah			
Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
36	3	0	0
$Y = (36+3+0+0)/40 \times 100\% = 97.5\%$ (Sangat Layak)			
2 : web sistem informasi industri kecil dan menengah dapat digunakan dalam keadaan <i>offline</i> .			
Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
32	6	0	0
$Y = (32+6+0+0+)/40 \times 100\% = 95\%$ (Sangat Layak)			
3 : Semua url/menu pada web sistem informasi industri kecil dan menengah dapat diakses secara <i>offline</i>			
Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
32	6	0	0
$Y = (32+6+0+0+)/40 \times 100\% = 95\%$ (Sangat Layak)			
4 : Web Sistem Informasi Industri Kecil dan Menengah memiliki waktu akses yang cepat pada saat <i>online</i> maupun <i>offline</i>			
Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
24	12	0	0

$Y = (24+12+0+0+)/40 \times 100\% = 90\% \text{ (Sangat Layak)}$			
5 : Web Sistem Informasi Industri Kecil dan Menengah mudah untuk ditambahkan pada <i>homescreen</i>			
Sangat Setuju	setuju	Tidak Setuju	Sangat Tidak Setuju
16	15	0	0
$Y = (16+15+0+0)/40 \times 100\% = 77.5\% \text{ (Sangat Layak)}$			
6 : Web Sistem Informasi Industri Kecil dan Menengah tidak perlu melewati proses instalasi layaknya aplikasi.			
Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
36	0	2	0
$Y = (36+0+2+0)/40 \times 100\% = 90\% \text{ (Sangat Layak)}$			
7 : Aplikasi Web Sistem Informasi Industri Kecil dan Menengah dapat diakses dengan cepat dalam keadaan <i>online</i> maupun <i>offline</i>			
Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
36	3	0	0
$Y = (36+3+0+0)/40 \times 100\% = 97.5\% \text{ (Sangat Layak)}$			
8. Web Sistem Informasi Industri Kecil dan Menengah tidak memiliki perbedaan tampilan pada keadaan jaringan apapun			
Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
32	3	2	0
$Y = (32+3+2+0)/40 \times 100\% = 92.5\% \text{ (Sangat Layak)}$			

9 : Data yang diinput saat offline berhasil tersimpan pada saat online			
Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
36	3	0	0
$Y = (36+3+0+0)/40 \times 100\% = 97.5\%$ (Sangat Layak)			
10 : Data pada Web Sistem Informasi Industri Kecil dan Menengah dapat diakses dengan cepat dalam kondisi jaringan apapun (online, kondisi jaringan internet lamabat, maupun offline			
Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
36	0	2	0
$Y = (36+0+2+0)/40 \times 100\% = 90\%$ (Sangat Layak)			



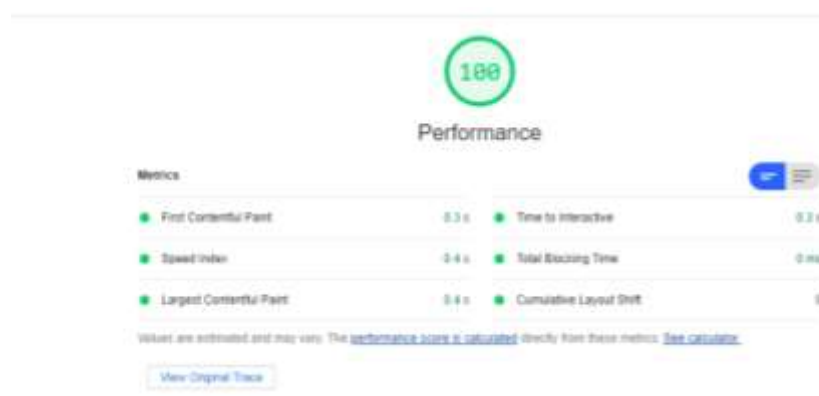
## LAMPIRAN 3

Detail pengujian progressive web apps pada lighthouse kategori performance

- Mode “Clear Storage”



- Tanpa “Clear Storage”



## LAMPIRAN 4

### Source Code Sistem Informasi Industri Kecil dan Menengah

#### App.js

```
class App extends React.Component {
  constructor(props){
    super(props);
    this.state = {
      currentUser: null,
      isAdmin: false,
    };
  }
  componentDidMount(){
    authenticationService.currentUser.subscribe(x => this.setState({
      currentUser: x,
      isAdmin: x && x.role === Role.Admin,
    }));
  }
  logout() {
    authenticationService.logout();
    history.push('/login');
  }
  render() {
    const { currentUser, isAdmin } = this.state;
    return (
      <BrowserRouter history={history}>
        {currentUser &&
          <nav className="navbar navbar-expand-lg bg-dark navbar-dark">
            <Link to="/" className="navbar-brand"><font color="white" size="4">Industri Kecil Menengah
            ||</font></Link>
            <div className="collapse navbar-collapse" id="navbarSupportedContent">
              <ul className="navbar-nav mr-auto">
                {isAdmin ?
                  <React.Fragment>
                    <li className="nav-item">
```

```

        <Link to={'/inputData'} className="nav-link"><font color="white" size="4"> Input
Pemasukan</font></Link>

    </li>

    <li className="nav-item">

        <Link to={'/indexdata'} className="nav-link"><font color="white" size="4"> Data
Pemasukan</font></Link>

    </li>

    <li className="nav-item">

        <Link to={'/inputdata1'} className="nav-link"><font color="white" size="4"> Input
Pengeluaran</font></Link>

    </li>

    <li className="nav-item">

        <Link to={'/indexdata1'} className="nav-link"><font color="white" size="4"> Data
Pengeluaran</font></Link>

    </li>

</React.Fragment> :

<React.Fragment>

    <li className="nav-item">

        <Link to={'indexdata_keuangan'} className="nav-link"><font color="white" size="4"> Data
Pemasukan</font></Link>

    </li>

    <li className="nav-item">

        <Link to={'indexdata1_keuangan'} className="nav-link"><font color="white" size="4"> Data
Pengeluaran</font></Link>

    </li>

    <li className="nav-item">

        <Link to={'summary'} className="nav-link"><font color="white"
size="4">Laporan</font></Link>

    </li>

</React.Fragment>    }

</ul>

<form className="form-inline my-2 my-lg-0">

    <button onClick={this.logout} className="btn btn-danger">Keluar</button>

</form>

</div>

</nav>    }

```

```

<Switch>
  <PrivateRoute exact path="/" component={HomePage} />
  <PrivateRoute path="/inputData" roles={[Role.Admin]} component={InputData} />
  <PrivateRoute path="/indexdata" roles={[Role.Admin]} component={IndexData} />
  <PrivateRoute path="/indexdata1" roles={[Role.Admin]} component={IndexData1} />
  <PrivateRoute path="/inputdata1" roles={[Role.Admin]} component={InputData1} />
  <PrivateRoute      path="/indexdata_keuangan"      roles={[Role.User]}
component={IndexData_Keuangan} />
  <PrivateRoute      path="/indexdata1_keuangan"      roles={[Role.User]}
component={IndexData1_Keuangan} />
  <Route exact path="/data/view/:id" component={(props) => <DataView {...props}
nomor={this.state.nomor}/>}/>
  <Route exact path="/data1/view/:id" component={(props) => <DataView1 {...props}
nomor={this.state.nomor}/>}/>
  <Route exact path="/summary" component={Summary} />
  <Route path="/login" component={LoginPage} />
</Switch>
</BrowserRouter>
)
}
}
export default App;

```

## Login.js

```
import React from 'react';
import { Formik, Field, Form, ErrorMessage } from 'formik';
import * as Yup from 'yup';
import { authenticationService } from '../_services';

class LoginPage extends React.Component {
  constructor(props) {
    super(props);

    // redirect to home if already logged in
    if (authenticationService.currentUserValue) {
      this.props.history.push('/');
    }
  }

  render() {
    return (
      <div className="container">
        <h2>Login</h2>
        <Formik
          initialValues={{
            username: '',
            password: ''
          }}
          validationSchema={Yup.object().shape({
            username: Yup.string().required('Username is required'),
            password: Yup.string().required('Password is required')
          })}
          onSubmit={({ username, password }, { setStatus, setSubmitting }) => {
            setStatus();
            authenticationService.login(username, password)
              .then(
                user => {
```

```

const { from } = this.props.location.state || { from: { pathname: "/" } };

    this.props.history.push(from);

    },

    error => {

        setSubmitting(false);

        setStatus(error);

    }

});

render=(({ errors, status, touched, isSubmitting }) => (

    <Form>

        <div className="form-group">

            <label htmlFor="username">Username</label>

            <Field name="username" type="text" className={'form-control' + (errors.username &&
touched.username ? ' is-invalid' : '')} />

            <ErrorMessage name="username" component="div" className="invalid-feedback" />

        </div>

        <div className="form-group">

            <label htmlFor="password">Password</label>

            <Field name="password" type="password" className={'form-control' + (errors.password
&& touched.password ? ' is-invalid' : '')} />

            <ErrorMessage name="password" component="div" className="invalid-feedback" />

        </div>

        <div className="form-group">

            <button
                type="submit"
                className="btn
                btn-primary"
                disabled={isSubmitting}>Login</button>

            {isSubmitting &&

            </div>

            {status &&

            <div className={'alert alert-danger'}>{status}</div>

            }

        </Form>

    )}

/>

</div> ) }}

```

```
export { LoginPage };
```

## HomePage.js

```
import React from 'react';

import { authenticationService } from '../_services';

class HomePage extends React.Component {
  constructor(props) {
    super(props);

    this.state = {
      currentUser: authenticationService.currentUserValue,
    };
  }

  render() {
    const { currentUser } = this.state;
    return (
      <div className="container">
        <h2>Sistem Informasi Industri Kecil Menengah</h2>
        <p>Kamu Login sebagai: <strong>{currentUser.role}</strong>.</p>
      </div>
    );
  }
}

export { HomePage };
```

## InputData.js

```
class InputData extends React.Component {  
  state = {  
    open: true,  
  };  
  constructor(props) {  
    super(props);  
    this.state = {  
      tanggal: 'tanggal',  
      bulan: 'bulan',  
      tahun: 'tahun',  
      jamur: '',  
      baglog: '',  
      bibit: '',  
      nama_mesin: '',  
      lainnya: '',  
      contoh:[],  
      database:[],  
    };  
  
    render() {  
      const { tanggal, bulan, tahun, jamur, baglog, bibit, lainnya } = this.state;  
      return(  
        <div className="container-fluid">  
          <div className="card">  
            <div className="card-body">  
              <h3 className="card-title">Input Data Pemasukan</h3>  
              <div className="card-body">  
                <br />  
                <form noValidate autoComplete="off">  
                  <div className="row">  
                    <div className="col-sm-6">  
                      <div className="form-group">  
                        <label>Penanggung Jawab:  
                        <input  
                          type="text"
```



```
className="form-control"
name="nama_mesin"
value={ this.state.nama_mesin }
placeholder="Nama Penanggung Jawab"
onChange={(e)=>this.handleMessageChange(e)}
/>
</label>
</div>
<br />
<div className="form-group">
  <label>Tanggal :
  <select className="form-control"
  value={ tanggal }
  onChange={ (e)=>this.handleMessageChange(e)}
  id="outlined-adornment-weight"
  variant="outlined" name="tanggal">
    <option value="">Pilih Tanggal</option>
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
    <option value="6">6</option>
    <option value="7">7</option>
    <option value="8">8</option>
    <option value="9">9</option>
    <option value="10">10</option>
    <option value="11">11</option>
    <option value="12">12</option>
    <option value="13">13</option>
    <option value="14">14</option>
    <option value="15">15</option>
    <option value="16">16</option>
    <option value="17">17</option>
    <option value="18">18</option>
    <option value="19">19</option>
```

```
<option value="20">20</option>
<option value="21">21</option>
<option value="22">22</option>
<option value="23">23</option>
<option value="24">24</option>
<option value="25">25</option>
<option value="26">26</option>
<option value="27">27</option>
<option value="28">28</option>
<option value="29">29</option>
<option value="30">30</option>
<option value="31">31</option>
</select>
</label>
</div>
<br />
<div className="form-group">
  <label>Bulan :
  <select className="form-control"
  value={bulan}
  onChange={(e)=>this.handleMessageChange(e)}
  id="outlined-adornment-weight"
  variant="outlined" name="bulan">
    <option value="">Pilih Bulan</option>
    <option value="Januari">Januari</option>
    <option value="Februari">Februari</option>
    <option value="Maret">Maret</option>
    <option value="April">April</option>
    <option value="Mei">Mei</option>
    <option value="Juni">Juni</option>
    <option value="Juli">Juli</option>
    <option value="Agustus">Agustus</option>
    <option value="September">September</option>
    <option value="Oktober">Oktober</option>
    <option value="November">November</option>
    <option value="Desember">Desember</option>
```

```
</select>
</label>
</div>
<br />
<div className="form-group">
  <label htmlFor="tahun">Tahun :
  <select className="form-control"
    value={tahun}
    onChange={(e)=>this.handleMessageChange(e)}
    id="outlined-adornment-weight"
    variant="outlined" name="tahun">
    <option value="">Pilih Tahun</option>
    <option value="2017">2017</option>
    <option value="2018">2018</option>
    <option value="2019">2019</option>
    <option value="2020">2020</option>
    <option value="2021">2021</option>
  </select>
</label>
</div>
</div>
<div className="col-sm-6">
  <div className="form-group">
    <label>Penjualan Jamur (Rp):
    <input
      type="text"
      className="form-control"
      name="jamur"
      value={jamur}
      placeholder="Penjualan Jamur"
      onChange={(e)=>this.handleMessageChange(e)} />
    </label>
  </div>
  <br />
  <div className="form-group">
    <label>Penjualan Baglog (Rp):
```

```
<input
  type="text"
  className="form-control"
  name="baglog"
  value={ baglog }
  placeholder="Penjualan Baglog"
  onChange={(e)=>this.handleMessageChange(e)}      />
</label>
</div>
<br />
<div className="form-group">
<label>Penjualan Bibit (Rp):
  <input
    type="text"
    className="form-control"
    name="bibit"
    value={ bibit }
    placeholder="Penjualan Bibit"
    onChange={(e)=>this.handleMessageChange(e)}    />
  </label>
</div>
<br />
<div className="form-group">
<label>Pemasukan Lainnya (Rp):
  <input
    type="text"
    className="form-control"
    name="lainnya"
    value={ lainnya }
    placeholder="Lainnya"
    onChange={(e)=>this.handleMessageChange(e)}    />
  </label>
</div>
</div>
<br />
```

```

        </form>

        <center>

        <Button onClick={()=>this.sendMessage()} variant="contained" color="primary" >

        <SaveIcon />

        Save

        </Button>

        </center>

    </div>

</div>

</div>

</div>

); }

handleMessageChange(e) {

    this.setState({[e.target.name]: e.target.value}); }

async sendMessage() {

    var today = new Date();

    var dd = today.getDate();

    var mm = today.getMonth() + 1; //January is 0!

    var yyyy = today.getFullYear();

    if (dd < 10) {

        dd = '0' + dd; }

    if (mm < 10) {

        mm = '0' + mm; }

    const hari = mm + '/' + dd + '/' + yyyy;

    const id = Date.now().toString();

    const { tanggal, bulan, tahun, jamur, baglog, bibit, nama_mesin, lainnya } = this.state

    const tipe = 'Penjualan';

    const date = hari.toString();

    const dataDatabase = {id, tanggal, bulan, tahun, jamur, baglog, bibit, nama_mesin, lainnya, date, tipe};

    await this.db.database.insert(dataDatabase);

    this.setState({ tanggal: ""});

    this.setState({ bulan: ""});

    this.setState({ tahun: ""});

    this.setState({ jamur: ""});

    this.setState({ baglog: ""});

    this.setState({ bibit: ""});

```

```

this.setState({ nama_mesin: ""});
this.setState({lainnya: ""}); }
DeleteMessage = async (id) =>{
  await this.db.database.findOne().where('_id').eq(id)
  .remove(); }
editData = async (id) => {
  //this.setState({newMessage: ""});
  const query = this.db.database.find().where('_id').gt(id);
  // await query.update({
  //   $inc: {
  //     me: 1 // increases age of every found document by 1
  //   }
  // });
  console.log(query);
  //}); }
getData = async (id) => {
  //await this.db.messages.findOne().where('_id').eq(id).exec().then(doc => console.log(doc));
  await this.db.database.findOne(id).exec().then(doc => {
    console.log(doc._data)
    this.setState({ contoh : doc._data })
    setTimeout(()=>{
      console.log(this.state.contoh)
    },2000)
  }); }
sgetData = async (id) =>{
  // await this.db.messages.findOne().where('_id').eq(id).exec().then(doc => console.log(doc));
  await this.db.database.findOne(id).exec().then(doc => console.log(doc._data)) ; }
updateData = async (id) => {
  this.getData(id);
  console.log(id) }}

```

```
export default (InputData);
```

## IndexData.js

```
class IndexData extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      tanggal: 'tanggal',
      bulan: 'bulan',
      tahun: 'tahun',
      jamur: 0,
      baglog: 0,
      bibit: 0,
      nama_mesin: "",
      lainnya: "",
      total: number,
      jumlah: "",
      contoh:[],
      database:[],
    };

    async createDatabase() {
      // password must have at least 8 characters dan ini fungsinya untuk enkripsi data yang masuk ke dalam koleksi data
      const db = await RxDB.create(
        {name: dbName, adapter: 'idb', password: '12345678', ignoreDuplicate: true}
      );
      console.dir(db);
      // create collection
      const databaseCollection = await db.collection({
        name: 'database',
        schema: skema
      });
    }
    render() {
      return (
        <div className="container-fluid">
          <div className="card">
            <div className="card-body">
```

```

    <h3 className="card-title">Data Pemasukan Industri Kecil Menengah</h3>
    <div className="card-body">
      <table>
      <thead>
      <tr>
      <td width="100">Tanggal</td>
      <td width="250" align="center">Penjualan Jamur (Rp)</td>
      <td width="250" align="center">Penjualan Baglog (Rp)</td>
      <td width="250" align="center">Penjualan Bibit (Rp)</td>
      <td width="250" align="center">Pemasukan Lainnya (Rp)</td>
      <td align="center"></td>
      </tr>
      </thead>
      {this.renderMessages()}
    </table>
  </div>
</div>
</div>
</div>
); }
renderMessages = () =>{
  return
  this.state.database.map(({id, jumlah,nama_mesin,tanggal,bulan,jamur,baglog,bibit,lainnya,tahun,date}) => {
    return (
      <tbody className="table table-hover">
      <tr>
      <td width="100">
      {tanggal}
      </td>
      <td width="250" align="center">
      <NumberFormat value={jamur} displayType='text' thousandSeparator={true} />
      </td>
      <td width="250" align="center">
      <NumberFormat value={baglog} displayType='text' thousandSeparator={true} />
      </td>
      <td width="250" align="center">

```



```

<NumberFormat value={bibit} displayType='text' thousandSeparator={true} />
</td>
<td width="250" align="center">
<NumberFormat value={lainnya} displayType='text' thousandSeparator={true} prefix='Rp' />
</td>
<td>
<Link to={"/data/view/${id}`}><Button variant="contained" color="primary">
View
</Button></Link>
</td>
<td>
<Button variant="contained" color="secondary" onClick={()=>{this.DeleteMessage(id)}}>Delete</Button>
</td>
</tr>
</tbody>
);
});
}

handleMessageChange(e) {
  this.setState({[e.target.name]: e.target.value});
}

async addMessage() {
  var today = new Date();
  var dd = today.getDate();
  var mm = today.getMonth() + 1; //January is 0!
  var yyyy = today.getFullYear();
  if (dd < 10) {
    dd = '0' + dd;
  }
  if (mm < 10) {
    mm = '0' + mm;
  }
  const hari = mm + '/' + dd + '/' + yyyy;
  const id = Date.now().toString();
  const { nama_mesin,tanggal,bulan,tahun } = this.state
  const tipe = 'Penjualan';
  const date = hari.toString();
  const dataDatabase = {id,nama_mesin,tanggal,bulan,tahun,date,tipe};

```

```

await this.db.database.insert(dataDatabase);
}
DeleteMessage = async (id) =>{
  await this.db.database.findOne().where('_id').eq(id)
  .remove();
  // this.db.messages.find().exec() // <- find all documents
  // .then(documents => console.log(id));
  // console.log(this.state.messages.map); }
editData = async (id) => {
  //this.setState({ newMessage: ""});
  const query = this.db.database.find().where('_id').gt(id);
  console.log(query);
  //});
}
getData = async (id) => {
  //await this.db.messages.findOne().where('_id').eq(id).exec().then(doc => console.log(doc));
  await this.db.database.findOne(id).tipe('Penjualan').exec().then(doc => {
    console.log(doc._data)
    this.setState({ contoh : doc._data })
    setTimeout(()=>{
      console.log(this.state.contoh)
    },2000)
  }); }
sgetData = async (id) =>{
  // await this.db.messages.findOne().where('_id').eq(id).exec().then(doc => console.log(doc));
  await this.db.database.findOne(id).exec().then(doc => console.log(doc._data)) ;
}
updateData = async (id) => {
  this.getData(id);
  console.log(id) } }
export default (IndexData);

```

## DataView.js

```
class DataView extends React.Component {  
  state = {  
    open: true,  
  };  
  constructor(props) {  
    super(props);  
    this.state = {  
      tanggal: 'tanggal',  
      bulan: 'bulan',  
      tahun: 'tahun',  
      jamur: 0,  
      baglog: 0,  
      bibit: 0,  
      nama_mesin: "",  
      lainnya: "",  
      contoh:[],  
      database:[],  
    };  
  }  
  render() {  
    return (  
      <div>  
        <p>{this.EditData()}</p>  
      </div>  
    );  
  }  
  EditData = () => {  
    const { database } = this.state;  
    return (  
      <main>  
        <div>  
          <div className="card-body" align="left">  
            <h4 className="card-title">Pemasukan {database.tanggal} {database.bulan} {database.tahun}</h4>  
            <dl>  
              <dt>Penjualan Jamur :</dt>  
              <dd>{database.jamur}</dd>  
            </dl>  
          </div>  
        </div>  
      </main>  
    );  
  }  
}
```

```

<dt>Penjualan Baglog :</dt>
<dd>{database.baglog}</dd>
<dt>Penjualan bibit :</dt>
<dd>{database.bibit}</dd>
<dt>Pemasukan lainnya :</dt>
<dd>{database.lainnya}</dd>
<dt>Penanggung jawab :</dt>
<dd>{database.nama_mesin}</dd>
</dl>
</div>
</div>
<div className="container-fluid">
  <div className="card">
    <div className="card-body">
      <h3 className="card-title">Edit Data Pemasukan {database.tanggal} {database.bulan}
{database.tahun}</h3>
      <div className="card-body">
        <br />
        <form noValidate autoComplete="off">
          <div className="row">
            <div className="col-sm-6">
              <div className="form-group">
                <label>Penanggung Jawab:
                <input
                  type="text"
                  className="form-control"
                  name="nama_mesin"
                  value={this.state.nama_mesin}
                  placeholder={database.nama_mesin}
                  onChange={(e)=>this.handleMessageChange(e)}
                  InputProps={{
                    endAdornment: <InputAdornment position="end"></InputAdornment>,
                  }} />
                </label>
              </div>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>

```

```
<div className="form-group">
  <label>Tanggal :
  <select className="form-control"
    value={this.state.tanggal}
    onChange={(e)=>this.handleMessageChange(e)}
    InputProps={{
      endAdornment: <InputAdornment position="end"></InputAdornment>,
    }}
    id="outlined-adornment-weight"
    variant="outlined" name="tanggal">
    <option value="">{database.tanggal}</option>
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
    <option value="6">6</option>
    <option value="7">7</option>
    <option value="8">8</option>
    <option value="9">9</option>
    <option value="10">10</option>
    <option value="11">11</option>
    <option value="12">12</option>
    <option value="13">13</option>
    <option value="14">14</option>
    <option value="15">15</option>
    <option value="16">16</option>
    <option value="17">17</option>
    <option value="18">18</option>
    <option value="19">19</option>
    <option value="20">20</option>
    <option value="21">21</option>
    <option value="22">22</option>
    <option value="23">23</option>
    <option value="24">24</option>
    <option value="25">25</option>
```

```

        <option value="26">26</option>
        <option value="27">27</option>
        <option value="28">28</option>
        <option value="29">29</option>
        <option value="30">30</option>
        <option value="31">31</option>
    </select>
</label>
</div>
<br />
<div className="form-group">
    <label>Bulan :
        <select          className="form-control"          value={this.state.bulan}
onChange={(e)=>this.handleMessageChange(e)} InputProps={{
    endAdornment: <InputAdornment position="end"></InputAdornment>,
}}
id="outlined-adornment-weight"
variant="outlined" name="bulan">
        <option value="">{database.bulan}</option>
        <option value="Januari">Januari</option>
        <option value="Februari">Februari</option>
        <option value="Maret">Maret</option>
        <option value="April">April</option>
        <option value="Mei">Mei</option>
        <option value="Juni">Juni</option>
        <option value="Juli">Juli</option>
        <option value="Agustus">Agustus</option>
        <option value="September">September</option>
        <option value="Oktober">Oktober</option>
        <option value="November">November</option>
        <option value="Desember">Desember</option>
    </select>
</label>
</div>
<br />
<div className="form-group">

```

```

<label htmlFor="tahun">Tahun :
  <select className="form-control" value={this.state.tahun} onChange={(e)=>this.handleMessageChange(e)}
  InputProps={{ endAdornment: <InputAdornment position="end"></InputAdornment>,
    id="outlined-adornment-weight"
    variant="outlined" name="tahun">
      <option value="">{database.tahun}</option>
      <option value="2017">2017</option>
      <option value="2018">2018</option>
      <option value="2019">2019</option>
      <option value="2020">2020</option>
      <option value="2021">2021</option>
    </select>
  </label>
</div>
</div>
<div className="col-sm-6">
  <div className="form-group">
    <label>Penjualan Jamur (Rp):
      <input
        type="text"
        className="form-control"
        name="jamur"
        value={this.state.jamur}
        placeholder={database.jamur}
        onChange={(e)=>this.handleMessageChange(e)}
        InputProps={{
          endAdornment: <InputAdornment position="end"></InputAdornment>,
        }}
      </input>
    </label>
  </div>
  <br />
  <div className="form-group">
    <label>Penjualan Baglog (Rp):
      <input
        type="text"
        className="form-control"

```

```

name="baglog"
value={ this.state.baglog }
placeholder={ database.baglog }
onChange={(e)=>this.handleMessageChange(e)}
InputProps={{
  endAdornment: <InputAdornment position="end"></InputAdornment>,
}}
/>
</label>
</div>
<div className="form-group">
<label>Penjualan Bibit (Rp):
<input
type="text"
className="form-control"
name="bibit"
value={ this.state.bibit }
placeholder={ database.bibit }
onChange={(e)=>this.handleMessageChange(e)}
InputProps={{
  endAdornment: <InputAdornment position="end"></InputAdornment>,
}}
/>
</label>
</div>
<div className="form-group">
<label>Pemasukan Lainnya (Rp):
<input
type="text"
className="form-control"
name="lainnya"
value={ this.state.lainnya }
placeholder={ database.lainnya }
onChange={(e)=>this.handleMessageChange(e)}
InputProps={{
  endAdornment: <InputAdornment position="end"></InputAdornment>,
}}
/>
</label>

```



```

        </div>
    </div>
</div>
</form>

<Button onClick={()=>this.updateData(database.id)} variant="contained" color="primary" >
    Perbaharui
</Button>
</center>
</div>
</div>
</div>
</div>
</main>
) }

    handleMessageChange(e) {
    this.setState({[e.target.name]: e.target.value}); }
async updateData(nomor) {
    const { tanggal,bulan,tahun,jamur,baglog,bibit,nama_mesin,lainnya,date } = this.state
    await this.db.database.find().where('id').eq(nomor).update({
        $set:{
            tanggal : tanggal,
            bulan : bulan,
            tahun : tahun,
            jamur : jamur,
            baglog : baglog,
            bibit : bibit,
            lainnya : lainnya,
            nama_mesin : nama_mesin,
            date : date,
        }
    });
    this.setState({tanggal:"});
    this.setState({bulan:"});
    this.setState({tahun: "});
    this.setState({jamur: "});
    this.setState({baglog: "});
    this.setState({bibit: "});

```

```

this.setState({ nama_mesin: ""});
this.setState({lainnya: ""});
this.setState({ date: ""}); }

DeleteMessage = async (id) =>{
await this.db.database.findOne().where('_id').eq(id)
.remove();
editData = async (id) => {
const query = this.db.database.find().where('_id').gt(id);
console.log(query);
}); }

getData = async (id) => {
await this.db.database.findOne(id).exec().then(doc => {
console.log(doc._data)
this.setState({ contoh : doc._data})
setTimeout(()=>{
console.log(this.state.contoh)
},2000)
}); }

sgetData = async (id) =>{
await this.db.database.findOne(id).exec().then(doc => console.log(doc._data)) ; }
} }


export default (DataView);

```













## KARTU BIMBINGAN SKRIPSI






Prodi S1 Teknik Informatika Universitas Hasanuddin

Stb.	Nama Mahasiswa
D42115012	Charina

Pembimbing.	Nama Pembimbing	Paraf & Tgl. Persetujuan Ujian Akhir
I	Dr. Eng. Zulkifli Tahir, S.T., M.Sc.	
II	A. Ais Prayogi Alimuddin, S.T., M.Eng.,	
No SK Pemb:		

Judul Skripsi:	Perancangan Aplikasi yang Handal Bencana menggunakan <i>Progressive Web App</i> dengan <i>React Js</i> dan <i>CouchDB</i> Studi Kasus : Industri Kecil dan Menengah
----------------	---

No.	Tanggal Bimbingan	Uraian Kegiatan Bimbingan	Paraf Pemb.
1	11/06/2020	Laporan progress pengerjaan sistem ke Pembimbing I	
2	09/09/2020	Konsultasi penulisan skripsi ke Pembimbing I	
3	12/10/2020	Laporan progress ke Pembimbing I	
4	19/10/2020	Konsultasi penulisan skripsi ke Pembimbing II	
5	05/11/2020	Laporan progress penulisan skripsi ke Pembimbing I	
6	12/11/2020	Konsultasi penulisan skripsi ke Pembimbing I	
7	16/11/2020	Laporan progress penulisan skripsi ke Pembimbing II	
8	20/11/2020	Demonstrasi rancangan sistem ke Pembimbing I	
9	24/11/2020	Konsultasi penulisan skripsi ke Pembimbing II	
10	26/11/2020	Laporan progress ke Pembimbing II	
11	27/11/2020	Laporan progress ke Pembimbing I	
12	01/12/2020	Laporan progress ke Pembimbing I	

13	23/01/2020	Bimbingan revisi seminar hasil ke Pembimbing I	
14	01/02/2021	Bimbingan revisi seminar hasil ke Pembimbing II	
15	09/02/2021	Bimbingan revisi seminar hasil ke Pembimbing II	
16	09/02/2021	Bimbingan revisi seminar hasil ke Pembimbing I	
17	26/02/2021	Asistensi skripsi ke pembimbing I	
18	26/02/2021	Asistensi skripsi ke pembimbing II	