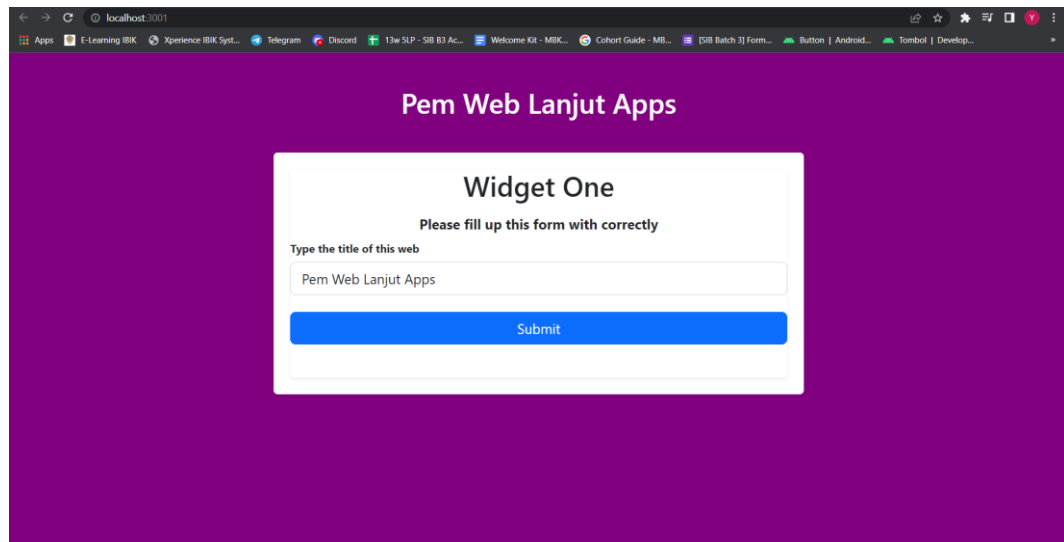


Nama : Yudhistira Kusuma
NPM : 202310067
Kelas : 6 TI-20-PA1
Matkul : Pemrograman Web Lanjut

1. Pada project sebelumnya buatlah sebuah package baru dengan nama latihan-5_1. Ubahlah script pada contoh kasus Gambar 4, dengan menggunakan RCC untuk Widget1.

Hasil Program



Source Code

```
import React from "react";

const Layout = (props) => {
  return (
    <div
      className="d-flex flex-column flex-root vh-100"
      style={{ backgroundColor: "purple" }}
    >
      <div className="d-flex flex-column flex-column-fluid">
        <div className="d-flex flex-center flex-column flex-column-
fluid p-10 pb-lg-20">
          <h1 className="mb-12 text-white text-center my-
5">{props.title}</h1>
          <div className="w-50 bg-body rounded shadow-sm p-4 mx-
auto">
            {props.children}
          </div>
        </div>
      </div>
    </div>
  );
};
```

```
};

export default Layout;
```

```
import React, { Component } from "react";
import Layout from "../Layout";

class Widget1 extends Component {
  constructor(props) {
    super(props);
    this.state = {
      title: "Pem Web Lanjut Apps",
    };
  }

  newTitle = (newTitle) => {
    this.setState({ title: newTitle });
  };

  render() {
    return (
      <Layout title={this.state.title}>
        <div className="bg-body rounded shadow-sm p-10 p-lg-15">
          <form method="post" autoComplete="off" className="form w-
100">
            <div className="text-center mb-10">
              <h1 className="text-dark mb-3">Widget One</h1>
              <div className="text-gray-300 fw-bold fs-5">
                Please fill up this form with correctly
              </div>
            </div>

            <div className="fv-row mb-10 fv-plugins-icon-container">
              <label className="form-label fs-6 fw-bolder text-dark
my-2">
                Type the title of this web
              </label>
              <input
                type="text"
                name="title"
                className="form-control form-control-lg form-control-
solid"
                defaultValue={this.state.title}
                onChange={(e) => this.newTitle(e.target.newTitle)}
              />
            </div>
          </form>
        </div>
      </Layout>
    );
  }
}
```

```

        <div className="text-center">
          <button
            type="button"
            className="btn btn-lg btn-primary w-100 mb-5 my-4"
          >
            Submit
          </button>
        </div>
      </form>
    </div>
  </Layout>
);
}
}

export default Widget1;

```

2. Pada tugas praktikum sebelumnya tentang form biodata anda, ubahlah tugas tersebut sehingga mampu menampung data dalam bentuk multiple.

Hasil Program

The screenshot shows a web browser window with a 'Form Mahasiswa' (Student Form) overlaying a background image of a classroom. The form is titled 'Form Mahasiswa' and contains five input fields: 'NPM (Numeric Only)', 'First Name', 'Middle Name', 'Last Name', and 'Birthdate'. Below the fields is a link that says 'Add new row'. At the bottom right of the form are two buttons: 'Clear' and 'Save'.

localhost:3001

Apps E-Learning IBK Xperience IBK Syst... Telegram Discord 13e SLP - SIB B3 Ac... Welcome Kit - MBK... Cohort Guide - MB... [SIB Batch 3] Form... Button | Android... Tombol | Develop...

Form Mahasiswa

NPM	First Name	Middle Name	Last Name	Birthdate
<input type="text" value="NPM (Numeric Only)"/>	<input type="text" value="First Name"/>	<input type="text" value="Middle Name"/>	<input type="text" value="Last Name"/>	<input type="text" value="2023-03-23"/>
<div>Please fill out this field.</div>				
				<div>Clear Save</div>

localhost:3001

Apps E-Learning IBK Xperience IBK Syst... Telegram Discord 13e SLP - SIB B3 Ac... Welcome Kit - MBK... Cohort Guide - MB... [SIB Batch 3] Form... Button | Android... Tombol | Develop...

Form Mahasiswa

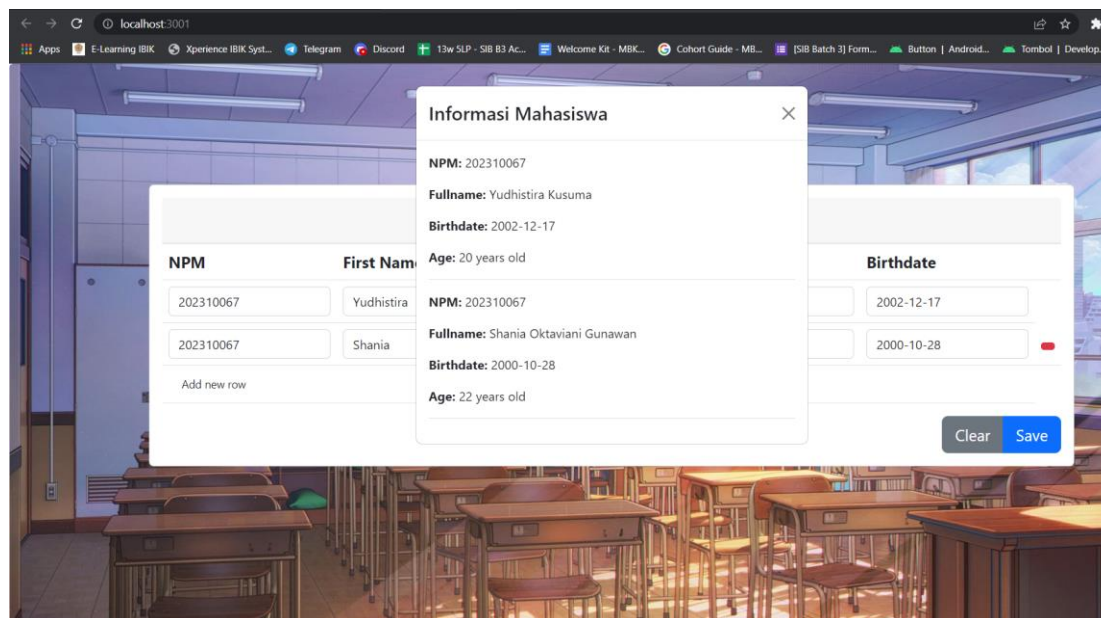
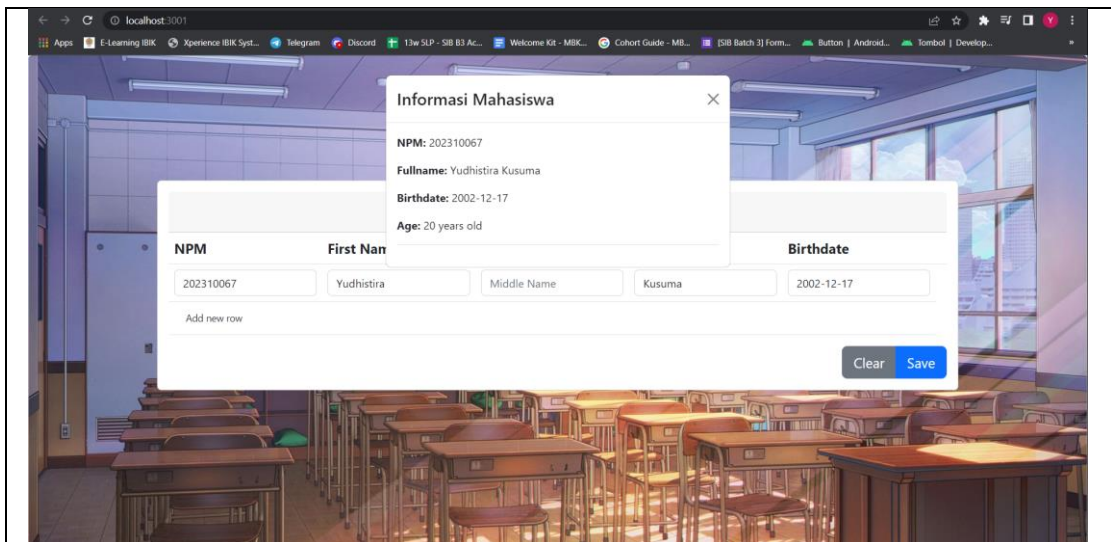
NPM	First Name	Middle Name	Last Name	Birthdate
<input type="text" value="202310067"/>	<input type="text" value="First Name"/>	<input type="text" value="Middle Name"/>	<input type="text" value="Last Name"/>	<input type="text" value="2023-03-23"/>
<div>Add new row</div> <div>Please fill out this field.</div>				
				<div>Clear Save</div>

localhost:3001

Apps E-Learning IBK Xperience IBK Syst... Telegram Discord 13e SLP - SIB B3 Ac... Welcome Kit - MBK... Cohort Guide - MB... [SIB Batch 3] Form... Button | Android... Tombol | Develop...

Form Mahasiswa

NPM	First Name	Middle Name	Last Name	Birthdate
<input type="text" value="202310067"/>	<input type="text" value="Yudhistira"/>	<input type="text" value="Middle Name"/>	<input type="text" value="Last Name"/>	<input type="text" value="2023-03-23"/>
<div>Add new row</div> <div>Please fill out this field.</div>				
				<div>Clear Save</div>



Source Code

```
import React, { useState } from "react";
import { Modal, Card, Form } from "react-bootstrap";
import DatePicker from "react-datepicker";
import "react-datepicker/dist/react-datepicker.css";
import Background from "../../asset/Background.jpg";

export default function FormsMultiple() {
  var today = new Date();
  const currDate = formatDate(today);
  const objProfile = {
    npm: 0,
    fname: "",
    mname: "",
    lname: "",
  };
}
```

```

    birthdate: currDate,
  };
  const [postData, setPostData] = useState([objProfile]);
  const [mymodal, setMymodal] = useState({
    show: false,
    message: "",
  });
  const modalHandleClose = () => {
    setMymodal({ ...mymodal, show: false });
  };
  const [show, setShow] = useState(false);
  const [validated, setValidated] = useState(false);

  const RowItem = ({ data, index }) => {
    return (
      <tr>
        <td>
          <input
            type="text"
            name="npm"
            className="form-control"
            placeholder="NPM (Numeric Only)"
            pattern="^[0-9]*$"
            defaultValue={data.npm ? data.npm : ""}
            onMouseLeave={(e) => {
              setPostData((prev) => {
                const newData = [...prev];
                newData[index].npm = parseInt(e.target.value);
                return newData;
              })
            }}
          />
        </td>
        <td>
          <input
            type="text"
            name="fname"
            className="form-control"
            placeholder="First Name"
            defaultValue={data.fname}
            onMouseLeave={(e) => {
              setPostData((prev) => {
                const newData = [...prev];
                newData[index].fname = e.target.value;
                return newData;
              })
            }}
          />
        </td>
      </tr>
    );
  };

```



```

        required
      />
    </td>
    <td>
      <input
        type="text"
        name="mname"
        className="form-control"
        placeholder="Middle Name"
        defaultValue={data.mname}
        onMouseLeave={(e) =>
          setPostData((prev) => {
            const newData = [...prev];
            newData[index].mname = e.target.value;
            return newData;
          })
        }
      />
    </td>
    <td>
      <input
        type="text"
        name="lname"
        className="form-control"
        placeholder="Last Name"
        defaultValue={data.lname}
        onMouseLeave={(e) =>
          setPostData((prev) => {
            const newData = [...prev];
            newData[index].lname = e.target.value;
            return newData;
          })
        }
      </td>
      <td>
        <DatePicker
          selected={new Date(data.birthdate)}
          onChange={(value) =>
            setPostData((prev) => {
              const newData = [...prev];
              newData[index].birthdate = formatDate(value);
              return newData;
            })
          }
          className="form-control"
          dateFormat="yyyy-MM-dd"
        />
      </td>
    </tr>
  </tbody>
</table>

```



```

        </p>
      </div>
    ) : (
      ""
    )
  })
</div>
);
};

const handlerSubmit = (e) => {
  e.preventDefault();
  let myresult = resultPost(postData);
  setMymodal({ ...mymodal, show: true, message: myresult });
  const form = e.currentTarget;

  if (form.checkValidity() === false) {
    e.preventDefault();
    e.stopPropagation();
  } else {
    e.preventDefault();
    setShow(true);
  }
  setValidated(true);
};

return (
  <Card
    className="d-flex justify-content-center align-items-center vh-
100"
    style={{
      backgroundImage: `url(${Background})`,
      backgroundRepeat: "no-repeat",
      backgroundSize: "cover",
      backgroundPosition: "center",
    }}
  >
    <div id="form-multiple" className="card mb-5 p-3">
      <div className="card-header">
        <div className="card-title text-center">
          <h3>Form Mahasiswa</h3>
        </div>
      </div>
      <div className="card-body p-0">
        <form
          method="post"
          autoComplete="off"
          onSubmit={(e) => handlerSubmit(e)}

```

```

    >
    <table className="table table-flush align-middle table-
row-bordered table-row-solid gy-4 gs-9">
      <thead className="border-gray-200 fs-5 fw-bold bg-
lighten">
        <tr>
          <th>NPM</th>
          <th>First Name</th>
          <th>Middle Name</th>
          <th>Last Name</th>
          <th>Birthdate</th>
          <th></th>
        </tr>
      </thead>
      <tbody>
        {postData.map((v, index) =>
          v ? <RowItem key={index} data={v} index={index} />
: ""
        )}
      </tbody>
      <tfoot>
        <tr>
          <td colspan={5}>
            <button
              className="btn btn-sm btn-light-primary"
              type="button"
              onClick={() =>
                setPostData((prev) => {
                  return [...prev, objProfile];
                })
            >
            <i className="bi bi-plus-circle me-2"></i>
            Add new row
            </button>
          </td>
        </tr>
      </tfoot>
    </table>

    <div className="text-end p-10">
      <div className="btn-group">
        <button className="btn btn-lg btn-secondary"
type="button">
          Clear
        </button>
        <button className="btn btn-lg btn-primary"
type="submit">

```

```

        Save
      </button>
    </div>
  </div>
</form>
</div>
<ModalPopUP
  show={mymodal.show}
  handleClose={modalHandleClose}
  message={mymodal.message}
/>
</div>
</Card>
);
}

const formatDate = (datestring) => {
  const today = new Date(datestring);
  const yyyy = today.getFullYear();
  let mm = today.getMonth() + 1; // Months start at 0!
  let dd = today.getDate();

  if (dd < 10) dd = "0" + dd;
  if (mm < 10) mm = "0" + mm;

  return yyyy + "-" + mm + "-" + dd;
};

const calculateAge = (birthdate) => {
  var today = new Date();
  var birthDate = new Date(birthdate);
  var age = today.getFullYear() - birthDate.getFullYear();
  var m = today.getMonth() - birthDate.getMonth();
  if (m < 0 || (m === 0 && today.getDate() < birthDate.getDate())) {
    age--;
  }
  return age;
};

const ModalPopUP = ({ show, handleClose, message }) => {
  return (
    <Modal show={show} onHide={handleClose}>
      <Modal.Header closeButton>
        <Modal.Title>Informasi Mahasiswa</Modal.Title>
      </Modal.Header>
      <Modal.Body>{message}</Modal.Body>
    </Modal>
  );
};

```

```
};
```