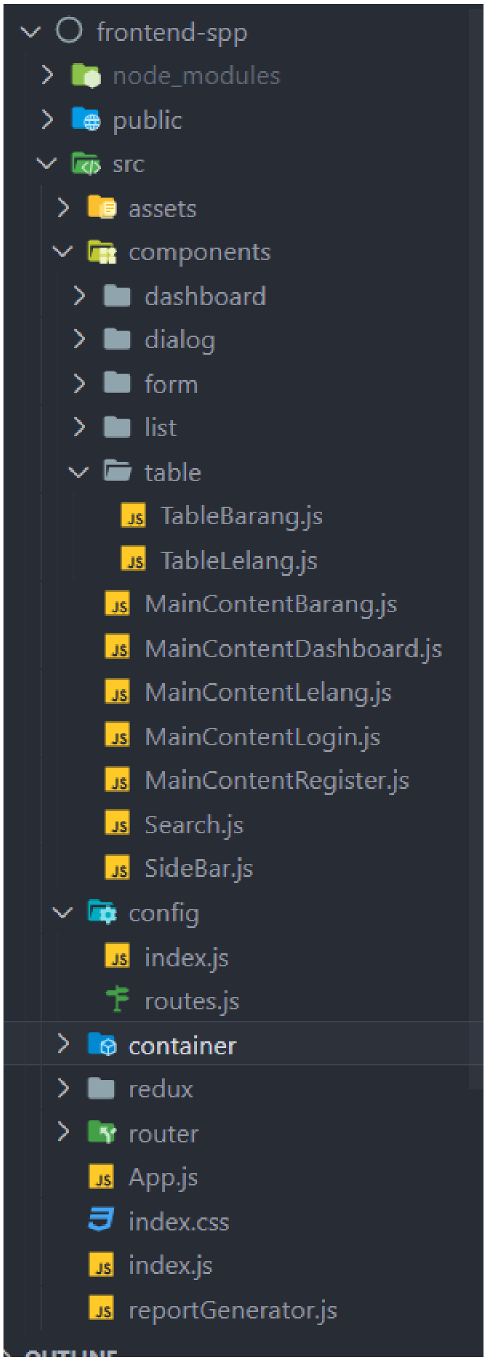
STRUKTUR FOLDER



```
⌄ frontend-spp
  › node_modules
  › public
  ⌄ src
    › assets
    ⌄ components
      › dashboard
      › dialog
      › form
      › list
      ⌄ table
          TableBarang.js
          TableLelang.js
        MainContentBarang.js
        MainContentDashboard.js
        MainContentLelang.js
        MainContentLogin.js
        MainContentRegister.js
        Search.js
        SideBar.js
    ⌄ config
        index.js
        routes.js
    › container
    › redux
    › router
      App.js
      index.css
      index.js
      reportGenerator.js
› OUTLINE
```

App.js

```
import React, { Component } from "react";
import { BrowserRouter as Router, Switch } from "react-router-dom";

import PublicRoute from "./router/publicRouter";
import PrivateRoute from "./router/privateRouter";

import Dashboard from "./container/page/Dashboard";
import Login from "./container/page/Login";
import NotFound from "./container/page/NotFound";
import Barang from "./container/page/Barang";
import Register from "./container/page/Register";
import Lelang from "./container/page/Lelang";
import DateAdapter from "@mui/lab/AdapterDateFns";

import { LocalizationProvider } from "@mui/lab";

import ROUTES from "./config/routes";
class App extends Component {
  render() {
    return (
      <Router>
        <Switch>
          <LocalizationProvider dateAdapter={DateAdapter}>
            <PrivateRoute
              restricted={true}
              path={ROUTES.DASHBOARD}
              component={Dashboard}
            />
            <PrivateRoute
              restricted={true}
              path={ROUTES.BARANG}
              component={Barang}
            />
            <PrivateRoute
              restricted={true}
              path={ROUTES.LELANG}
              component={Lelang}
            />
            <PublicRoute
              exact
              restricted={true}
              path={ROUTES.LOGIN}
              component={Login}
            />
            <PublicRoute
              exact
              restricted={true}
              path={ROUTES.REGISTER}
              component={Register}
            />
            <PublicRoute
              exact
              path={ROUTES.PAGE_NOT_FOUND}
              component={NotFound}
            />
          </LocalizationProvider>
        </Switch>
      </Router>
    );
  }
}

export default App;
```

Index.js

```
import "bootstrap/dist/css/bootstrap.min.css";

import "./index.css";
import App from "./App";
import { Provider } from "react-redux";
import React from "react";
import ReactDOM from "react-dom";
import { createStore } from "redux";
import rootReducer from "./redux/globalRedux";

// ! Store
const storeRedux = createStore(rootReducer);

ReactDOM.render(
  <Provider store={storeRedux}>
    <App />
  </Provider>,
  document.getElementById("root")
);
```

Router/privateRoute.js

```
import React from 'react';
import { Route, Redirect } from 'react-router-dom';

const privateRoute = ({ component: Component, ...rest }) => {
  return (
    <Route
      {...rest}
      render={(props) => {
        if (JSON.parse(localStorage.getItem('userLogin'))) {
          return (
            <Component {...props} history={props.history}
match={props.match} />
          );
        }
        return <Redirect to={'/'} />;
      }}
    />
  );
};

export default privateRoute;
```

Router/publicRoute.js

```
import React from 'react';
import { Route, Redirect } from 'react-router-dom';

const publicRoute = ({ component: Component, restricted, ...rest }) => {
  return (
    <Route
      {...rest}
      render={(props) => {
        if (localStorage.getItem('userLogin') && restricted) {
          return <Redirect to={'/dashboard'} />;
        }
        return <Component {...props} />;
      }}
    />
  );
};

export default publicRoute;
```

Component/Sidebar.js

```
import React, { Component } from "react";
import { Drawer, Box, List } from "@mui/material";
import EvaIcon from "react-eva-icons";

import { styled } from "@mui/styles";

const LayoutIcon = styled("div")(({ theme }) => ({
  height: 56,
  display: "flex",
  justifyContent: "center",
  margin: "8px 0px",
  alignItems: "center",
  cursor: "pointer",
  borderLeft: "3px solid #ffff",
}));

const LayoutIconActive = styled("div")(({ theme }) => ({
  height: 56,
  display: "flex",
  justifyContent: "center",
  margin: "8px 0px",
  alignItems: "center",
  cursor: "pointer",
  color: "#353535",
  borderLeft: "4px solid #FE2E12",
  backgroundColor: "#FFFBFA",
}));

class SideBar extends Component {
  state = {
    active: "Dashboard",
  };

  handleSideBar = (item) => {
    const currentPath = this.props.history.location.pathname;
    const pageDestination = item;

    if (pageDestination !== currentPath) {
      return this.props.history.push(pageDestination);
    }
  };

  handleLogout = () => {
    localStorage.removeItem("token");
    localStorage.removeItem("role");
    localStorage.removeItem("userLogin");
    return this.props.history.push("/");
  };

  render() {
    const drawerWidth = 98;
    const currentPath = this.props.history.location.pathname;

    return (
      <Drawer
        variant="permanent"
        sx={{
          width: drawerWidth,
          flexShrink: 0,
          [`& .MuiDrawer-paper`]: {
            width: drawerWidth,
            boxSizing: "border-box",
            backgroundColor: "#ffff",
            borderRight: "1px solid #EBEBEB",
            // justifyContent: 'center',
          },
        }}
      >
        <Box sx={{ overflow: "auto", mt: 10 }}>
          <List>
            {currentPath === "/dashboard" ? (
              <LayoutIconActive
                onClick={() => this.handleSideBar("/dashboard")}
              >
```

```
                <EvaIcon name="grid" size="large" fill="#18125C" />
              </LayoutIconActive>
            ) : (
              <LayoutIcon onClick={() => this.handleSideBar("/dashboard")}>
                <EvaIcon name="grid" size="large" fill="#A5A5A5" />
              </LayoutIcon>
            )}

            {currentPath === "/barang" ? (
              <LayoutIconActive onClick={() =>
this.handleSideBar("/barang")}>
                <EvaIcon name="shopping-bag" size="large" fill="#18125C" />
              </LayoutIconActive>
            ) : (
              <LayoutIcon onClick={() => this.handleSideBar("/barang")}>
                <EvaIcon name="shopping-bag" size="large" fill="#A5A5A5" />
              </LayoutIcon>
            )}

            {currentPath === "/lelang" ? (
              <LayoutIconActive onClick={() =>
this.handleSideBar("/lelang")}>
                <EvaIcon name="pricetags" size="large" fill="#18125C" />
              </LayoutIconActive>
            ) : (
              <LayoutIcon onClick={() => this.handleSideBar("/lelang")}>
                <EvaIcon name="pricetags" size="large" fill="#A5A5A5" />
              </LayoutIcon>
            )}

            {currentPath === "/admin" ? (
              <LayoutIconActive onClick={() =>
this.handleSideBar("/admin")}>
                <EvaIcon name="person" size="large" fill="#18125C" />
              </LayoutIconActive>
            ) : (
              <LayoutIcon onClick={() => this.handleSideBar("/admin")}>
                <EvaIcon name="person" size="large" fill="#A5A5A5" />
              </LayoutIcon>
            )}

            {currentPath === "/transaksi" ? (
              <LayoutIconActive
                onClick={() => this.handleSideBar("/transaksi")}
              >
                <EvaIcon name="swap" size="large" fill="#18125C" />
              </LayoutIconActive>
            ) : (
              <LayoutIcon onClick={() => this.handleSideBar("/transaksi")}>
                <EvaIcon name="swap" size="large" fill="#A5A5A5" />
              </LayoutIcon>
            )}
          </List>
          <List>
            <LayoutIcon onClick={() => this.handleLogout()}>
              <EvaIcon name="power" size="large" fill="#A5A5A5" />
            </LayoutIcon>
          </List>
        </Box>
      </Drawer>
    );
  }
}

export default SideBar;
```

Container/template/Dashboard.js

```
import React, { Component } from "react";
import { Button } from "react-bootstrap";
import { connect } from "react-redux";

import { CssBaseline, Box, AppBar, Toolbar, Typography } from
"@mui/material";
import SideBar from "../../components/SideBar";

import MainContentDashboard from "../../components/MainContentDashboard";
import MainContentBarang from "../../components/MainContentBarang";
import MainContentLelang from "../../components/MainContentLelang";
class Dashboard extends Component {
  state = {
    login: true,
    role: "",
  };

  handleLogout = () => {
    localStorage.removeItem("userLogin");
    localStorage.removeItem("token");
    localStorage.removeItem("role");

    return this.props.history.push("/");
  };

  getRole = () => {
    const role = localStorage.getItem("role");
    this.setState({
      role: role,
    });
  };

  componentDidMount() {
    this.getRole();
  }
  render() {
    return (
      <>
        {/* if else admin / masyarakat */}
        {this.state.role === "admin" || this.state.role === "petugas" ? (
          <Box sx={{ display: "flex", backgroundColor: "#232323" }}>
            <CssBaseline />
            <AppBar
              position="fixed"
              sx={{
                width: `calc(100% - 98px)`,
                padding: "12px 0px",
                ml: `98px`,
                boxShadow: "none",
                backgroundColor: "transparent",
                borderBottom: "1px solid #EBEBEB",
              }}
            >
              <Toolbar>
                <Typography sx={{ pl: 4 }} variant="h6" noWrap
color="#3D3D3D">
                  {this.props.title}
                </Typography>
              </Toolbar>
            </AppBar>

            <SideBar history={this.props.history} />

            {this.props.title === "Dashboard" && <MainContentDashboard />}
            {this.props.title === "Barang" && <MainContentBarang />}
            {this.props.title === "Lelang" && <MainContentLelang />}
          </Box>
        ) : (
          <div>
            <div>page masyrakat</div>
            <Button onClick={this.handleLogout}>Logout</Button>
          </div>
        )}
      </>
    );
  }
}

// ! Panggil globalState
const mapStateToProps = (state, props) => {
  return {
    statusUser: state.statusUser,
  };
};

export default connect(mapStateToProps)(Dashboard);
```

Container/templates/login.js

```
import { Grid } from "@mui/material";
import React, { Component } from "react";

import OrganismLogin from "../../components/MainContentLogin";

class Login extends Component {
  render() {
    return (
      <Grid container sx={{ height: "100vh" }} justifyContent="center">
        <Grid maxWidth="100%" width="50%">
          <OrganismLogin history={this.props.history} />
        </Grid>
      </Grid>
    );
  }
}

export default Login;
```

Component/form/Register.js

```
import React, { useState } from "react";
import { useFormik } from "formik";
import * as yup from "yup";

import {
  Checkbox,
  FormControlLabel,
  Button,
  Typography,
  CircularProgress,
  Snackbar,
  Alert,
} from "@mui/material";

import { URL_API } from "../../config/index";

import axios from "axios";
import { styled } from "@mui/styles";

const InputStyle = styled("input")(({ theme }) => ({
  margin: 8,
  border: "none",
  borderRadius: 6,
  paddingLeft: 16,
  paddingRight: 16,
  fontSize: 14,
  fontWeight: 300,
  width: "100%",
  letterSpacing: 0.3,
  height: 40,
  backgroundColor: "#eaeef2",
  "&:focus": {
    border: "none",
    outline: "none",
  },
}));

const TypoStyle = styled(Typography)(({ theme }) => ({
  color: "#353535",
}));

const TypoError = styled(Typography)(({ theme }) => ({
  fontSize: 12,
  marginLeft: 8,
  opacity: 0.7,
  color: "#CE1C1C",
}));

const DivStyle = styled("div")(({ theme }) => ({
  width: 400,
  margin: "6px 32px",
}));

const Register = (props) => {
  const [showPassword, setShowPassword] = useState(true);
  const [alert, setAlert] = useState({
    show: false,
    message: null,
  });
  const [snack, setSnack] = useState(false);
  const [RegisterProccess, setRegisterProccess] = useState(false);

  const fetchRegister = async (values, resetForm) => {
    setRegisterProccess(true);
    try {
      const response = await axios.post(
        `${URL_API}/masyarakat/register`,
        values
      );
      if (response) {
        setSnack(true);
        setRegisterProccess(false);
        resetForm();
      }
    } catch (error) {
      setAlert({
        show: true,
        message: "Username is already exist",
      });
      setRegisterProccess(false);
      resetForm();
    }
  };

  const handleShowPassword = () => {
    setShowPassword(!showPassword);
  };

  const validationSchema = yup.object({
    nama: yup.string("Enter your name").required("Name is required"),
    telp: yup
      .string()
      .required("Telepon is required")
      .max(12, "Number Phone be of maximum 12 number length")
      .matches(/^[0-9.]+$/, `Only numbers (1-9)`),
    username: yup
      .string("Enter your username")
      .required("Username is required"),
    password: yup
      .string("Enter your password")
      .min(6, "Password should be of minimum 6 characters length")
      .required("Password is required"),
  });

  const formik = useFormik({
    enableReinitialize: true,
    initialValues: {
      nama: "",
      telp: "",
      username: "",
      password: "",
    },
    validationSchema: validationSchema,
    onSubmit: (values, { resetForm }) => {
      fetchRegister(values, resetForm);
    },
  });

  return (
    <>
      <form onSubmit={formik.handleSubmit}>
        <Snackbar
          open={snack}
          autoHideDuration={5000}
          onClose={() => setSnack(false)}
          anchorOrigin={{ vertical: "top", horizontal: "center" }}
        >
          <Alert
            severity="success"
            onClose={() => setSnack(false)}
```

```
            sx={{ width: "100%" }}
            style={{ minWidth: 500 }}
          >
            <b>Success. </b> Your account successfully register
          </Alert>
        </Snackbar>

        <DivStyle>
          <TypoStyle sx={{ fontWeight: 600, fontSize: 14, ml: 1, mr: 1 }}>
            Nama
          </TypoStyle>
          <InputStyle
            id="nama"
            name="nama"
            autoComplete="current-nama"
            placeholder="supriyadiiii"
            value={formik.values.nama || ""}
            onChange={formik.handleChange}
          />
          {formik.touched.nama && Boolean(formik.errors.nama) ? (
            <TypoError>{formik.touched.nama &&
formik.errors.nama}</TypoError>
          ) : null}
        </DivStyle>

        <DivStyle>
          <TypoStyle sx={{ fontWeight: 600, fontSize: 14, ml: 1, mr: 1 }}>
            Telepon
          </TypoStyle>
          <InputStyle
            id="telp"
            name="telp"
            autoComplete="current-telp"
            placeholder="081364018392"
            value={formik.values.telp || ""}
            onChange={formik.handleChange}
          />
          {formik.touched.telp && Boolean(formik.errors.telp) ? (
            <TypoError>{formik.touched.telp &&
formik.errors.telp}</TypoError>
          ) : null}
        </DivStyle>

        <DivStyle>
          <TypoStyle sx={{ fontWeight: 600, fontSize: 14, ml: 1, mr: 1 }}>
            Username
          </TypoStyle>
          <InputStyle
            id="username"
            name="username"
            autoComplete="current-username"
            placeholder="username"
            value={formik.values.username || ""}
            onChange={formik.handleChange}
          />
          {(formik.touched.username && Boolean(formik.errors.username)) ||
          alert.show ? (
            <TypoError>
              {(formik.touched.username && formik.errors.username) ||
                alert.message}
            </TypoError>
          ) : null}
        </DivStyle>

        <DivStyle>
          <TypoStyle sx={{ fontWeight: 600, fontSize: 14, ml: 1, mr: 1 }}>
            Password
          </TypoStyle>
          <InputStyle
            id="password"
            name="password"
            autoComplete="current-password"
            placeholder="password"
            type={showPassword ? "password" : "text"}
            value={formik.values.password || ""}
            onChange={formik.handleChange}
          />
          {formik.touched.password && Boolean(formik.errors.password) ? (
            <TypoError>
              {formik.touched.password && formik.errors.password}
            </TypoError>
          ) : null}
        </DivStyle>

        <DivStyle style={{ width: 400, margin: "0px 32px" }}>
          <div style={{ margin: "0px 8px" }}>
            <FormControlLabel
              onClick={() => handleShowPassword()}
              control={
                <Checkbox
                  size="small"
                  sx={{
                    "&:hover": {
                      bgcolor: "transparent",
                    },
                  }}
                />
              }
              label={
                <TypoStyle sx={{ fontSize: 14, fontWeight: 500 }}>
                  Show Password
                </TypoStyle>
              }
            />
          </div>
        </DivStyle>
        <DivStyle style={{ width: 400, margin: "6px 32px" }}>
          <Button
            type="submit"
            fullWidth
            variant="contained"
            disabled={RegisterProccess}
            sx={{
              m: 1,
              textTransform: "capitalize",
              backgroundColor: "#FE2E12",
              boxShadow: "none",
              padding: "8px 64px",
              letterSpacing: 1,
              "&:hover": {
                boxShadow: "none",
                backgroundColor: "#FE2E12",
              },
            }}
          >
            {RegisterProccess && <CircularProgress size={24} />}

            {!RegisterProccess && <Typography>Register</Typography>}
          </Button>
        </DivStyle>
      </form>
    </>
  );
};

export default Register;
```

Component/form/login.js

```
import React, { useState } from "react";
import { useFormik } from "formik";
import * as yup from "yup";

import {
  Checkbox,
  FormControlLabel,
  Button,
  Typography,
  CircularProgress,
} from "@mui/material";

import { URL_API } from "../../config/index";

import axios from "axios";
import { useHistory } from "react-router";
import { styled } from "@mui/styles";

const InputStyle = styled("input")(({ theme }) => ({
  margin: 8,
  border: "none",
  borderRadius: 6,
  paddingLeft: 16,
  paddingRight: 16,
  fontSize: 14,
  fontWeight: 300,
  width: "100%",
  letterSpacing: 0.3,
  height: 40,
  backgroundColor: "#eaeef2",
  "&:focus": {
    border: "none",
    outline: "none",
  },
}));

const TypoError = styled(Typography)(({ theme }) => ({
  fontSize: 12,
  marginLeft: 8,
  opacity: 0.7,
  color: "#CE1C1C",
}));

const DivStyle = styled("div")(({ theme }) => ({
  width: 400,
  margin: "6px 32px",
}));

const Login = (props) => {
  const history = useHistory();
  const [showPassword, setShowPassword] = useState(true);
  const [alert, setAlert] = useState({
    show: false,
    message: null,
  });
  const [loginProcess, setLoginProcess] = useState(false);

  const submitLogin = async (values) => {
    setLoginProcess(true);
    try {
      const responseLogin = await axios.post(`${URL_API}/login`, values);
      if (!responseLogin.data.logged) {
        setAlert({
          show: true,
          message: "Invalid username or password",
        });
        setLoginProcess(false);
      } else if (responseLogin.data.logged) {
        setAlert({
```

```
        show: false,
        message: null,
      });
      localStorage.setItem("token", responseLogin.data.token);
      localStorage.setItem(
        "userLogin",
        JSON.stringify(responseLogin.data.data)
      );
      localStorage.setItem("role", responseLogin.data.data.level);

      setLoginProcess(false);
      return history.push("/dashboard");
    }
  } catch (error) {
    setLoginProcess(false);
    console.log(error);
  }
};

const handleShowPassword = () => {
  setShowPassword(!showPassword);
};

const validationSchema = yup.object({
  username: yup
    .string("Enter your username")
    .required("Username is required"),
  password: yup
    .string("Enter your password")
    .min(6, "Password should be of minimum 6 characters length")
    .required("Password is required"),
});

const formik = useFormik({
  enableReinitialize: true,
  initialValues: {
    username: "",
    password: "",
  },
  validationSchema: validationSchema,
  onSubmit: (values) => {
    submitLogin(values);
  },
});
return (
  <form onSubmit={formik.handleSubmit}>
    <DivStyle>
      <Typography sx={{ fontWeight: 600, fontSize: 14, ml: 1, mr: 1 }}>
        Username
      </Typography>
      <InputStyle
        id="username"
        name="username"
        autoComplete="current-username"
        placeholder="username"
        value={formik.values.username || ""}
        onChange={formik.handleChange}
      />
      {(formik.touched.username && Boolean(formik.errors.username)) ||
      alert.show ? (
        <TypoError>
          {(formik.touched.username && formik.errors.username) ||
            alert.message}
        </TypoError>
      ) : null}
    </DivStyle>

    <DivStyle>
      <Typography sx={{ fontWeight: 600, fontSize: 14, ml: 1, mr: 1 }}>
        Password
      </Typography>
      <InputStyle
        id="password"
        name="password"
        autoComplete="current-password"
        placeholder="password"
        type={showPassword ? "password" : "text"}
        value={formik.values.password || ""}
        onChange={formik.handleChange}
      />
      {formik.touched.password && Boolean(formik.errors.password) ? (
        <TypoError>
          {formik.touched.password && formik.errors.password}
        </TypoError>
      ) : null}
    </DivStyle>

    <div style={{ marginLeft: 40 }}>
      <FormControlLabel
        onClick={() => handleShowPassword()}
        control={
          <Checkbox
            checked={!showPassword}
            size="small"
            sx={{
              "&:hover": {
                bgcolor: "transparent",
              },
            }}
          />
        }
        label={
          <Typography sx={{ fontSize: 14, fontWeight: 500 }}>
            Show Password
          </Typography>
        }
```

```
        />
      </div>

      <DivStyle style={{ width: 400, margin: "6px 32px" }}>
        <Button
          type="submit"
          fullWidth
          variant="contained"
          disabled={loginProcess}
          sx={{
            m: 1,
            textTransform: "capitalize",
            backgroundColor: "#FE2E12",
            boxShadow: "none",
            padding: "8px 64px",
            letterSpacing: 1,
            "&:hover": {
              boxShadow: "none",
              backgroundColor: "#FE2E12",
            },
          }}
        >
          {loginProcess && <CircularProgress size={24} />}

          {!loginProcess && <Typography>Sign in</Typography>}
        </Button>
      </DivStyle>
    </form>
  );
};

export default Login;
```

Component/mainContentRegister.js

```
import React from "react";
import { Typography, Stack } from "@mui/material";

import FormRegister from "./form/Register";

const Register = (props) => {
  const handleRegister = () => {
    return props.history.push("/");
  };
  return (
    <Stack
      justifyContent="center"
      alignItems="center"
      sx={{ height: "100vh", width: "100%" }}
    >
```

```
      <div>
        <Typography
          align="center"
          sx={{ fontWeight: 700, fontSize: 32, mb: 5 }}
        >
          Wellcome to Buakags
        </Typography>
      </div>

      {/*Form Register*/}
      <FormRegister />

      <div className="textMovePageLogin">
        <Typography>Already have an account?</Typography>
        <Typography
          onClick={() => handleRegister()}
          sx={{
            fontWeight: 600,
            ml: 1,
            color: "#FE2E12",
            cursor: "pointer",
          }}
        >
          Sign in
        </Typography>
      </div>
    </Stack>
  );
};

export default Register;
```

Component/MainContentLogin.js

```
import React from "react";
import { Typography, Stack } from "@mui/material";

import FormLogin from "./form/Login";

const Login = (props) => {
  const handleRegister = () => {
    return props.history.push("/register");
  };
  return (
    <Stack
      justifyContent="center"
      alignItems="center"
      sx={{ height: "100vh", width: "100%" }}
    >
      <div>
```

```
        <Typography
          align="center"
          sx={{ fontWeight: 700, fontSize: 32, mb: 5 }}
        >
          Sign in to Lelang
        </Typography>
      </div>

      <FormLogin />

      <div className="textMovePageLogin">
        <Typography>Don’t have an account?</Typography>
        <Typography
          onClick={() => handleRegister()}
          sx={{
            fontWeight: 600,
            ml: 1,
            color: "#FE2E12",
            cursor: "pointer",
          }}
        >
          Create Account
        </Typography>
      </div>
    </Stack>
  );
};

export default Login;
```

Component/MainContentDashboard.js

```
import { Container, Grid } from "@mui/material";
import React from "react";

//component nested
import CardCountData from "./dashboard/CardCountData";
import CardUser from "./dashboard/CardUser";
import History from "./dashboard/History";

const MainContentDashboard = (props) => {
  return (
    <main
      style={{
        flexGrow: 1,
        minHeight: "100vh",
        overflow: "auto",
        background: "#FAFAFA",
      }}
    >
```

```
      <Container sx={{ pt: 10 }}>
        <Grid container sx={{ pt: 5 }}>
          <div
            className="left-dashboard"
            style={{
              minWidth: "70%",
            }}
          >
            <CardCountData />
            <History />
          </div>

          <div style={{ minWidth: "28%", padding: "24px 0px" }}>
            <div style={{ height: 700 }}>
              <CardUser />
            </div>
          </div>
        </Grid>
      </Container>
    </main>
  );
};

export default MainContentDashboard;
```

Component/maintContentLelang.js

```
import { Container, Grid, Button } from "@mui/material";
import React, { useState } from "react";
import { styled } from "@mui/material/styles";
import AddRoundedIcon from "@mui/icons-material/AddRounded";

import FormSearch from "./Search";
import Table from "./table/TableLelang";
import AddLelang from "./dialog/AddLelang";

//component nested

const CardStyle = styled("div")(({ theme }) => ({
  padding: 20,
  borderRadius: 16,
  backgroundColor: "#ffff",
}));

const MainContentLelang = (props) => {
  const [valueSearch, setValueSearch] = useState("");
  const [openDialog, setOpenDialog] = useState(false);
  const [reloadAll, setReloadAll] = useState(false);

  const handleSearch = (value) => {
    setValueSearch(value);
  };

  const handleOpenDialog = () => {
    setOpenDialog(true);
  };

  const handleCloseDialog = () => {
    setOpenDialog(false);
  };

  const handleReload = () => {
    setReloadAll(true);
    setTimeout(() => {
      setReloadAll(false);
    }, 500);
  };

  return (
    <main
      style={{
        flexGrow: 1,
        minHeight: "100vh",
        overflow: "auto",
        background: "#FAFAFA",
      }}
    >
      <Container sx={{ pt: 10 }}>
        <Grid container sx={{ pt: 5, mt: 5 }}>
          <AddLelang
            open={openDialog}
            closeDialog={() => handleCloseDialog()}
            processAdd={() => handleReload()}
          />
          <div
            style={{
              minWidth: "100%",
            }}
          >
            <CardStyle
              style={{
                maxWidth: "100%",
                height: 80,
                display: "flex",
                justifyContent: "space-between",
              }}
            >
              <div>
                <FormSearch
                  onKeywordSearchChange={(value) => handleSearch(value)}
                />
              </div>
              <div>
```

```
                <Button
                  sx={{
                    width: 150,
                    backgroundColor: "#FE2E12",
                    height: "100%",
                    textTransform: "capitalize",
                    borderRadius: 2,
                    color: "#ffff",

                    "&:hover": {
                      backgroundColor: "#FE2E12",
                      color: "#ffff",
                    },
                  }}
                  startIcon={<AddRoundedIcon />}
                  onClick={() => handleOpenDialog()}
                >
                  Tambah Lelang
                </Button>
              </div>
            </CardStyle>

            {/* Table */}
            <CardStyle style={{ marginTop: 32 }}>
              <Table keywordChange={valueSearch} reload={reloadAll} />
            </CardStyle>
          </div>
        </Grid>
      </Container>
    </main>
  );
};

export default MainContentLelang;
```

Component/MainContentBarang.js

```
import { Container, Grid, Button } from "@mui/material";
import React, { useEffect, useState } from "react";
import { styled } from "@mui/material/styles";
import AddRoundedIcon from "@mui/icons-material/AddRounded";

import FormSearch from "./Search";
import Table from "./table/TableBarang";
import AddBarang from "./dialog/AddBarang";

import generatePDF from "../reportGenerator";
import axios from "axios";
import { URL_API } from "../config";
```

```
//component nested

const CardStyle = styled("div")(({ theme }) => ({
  padding: 20,
  borderRadius: 16,
  backgroundColor: "#ffff",
}));

const MainContentSiswa = (props) => {
  const [valueSearch, setValueSearch] = useState("");
  const [openDialog, setOpenDialog] = useState(false);
  const [reloadAll, setReloadAll] = useState(false);

  const [dataBarang, setDataBarang] = useState([]);

  const getAllData = async () => {
    try {
      const response = await axios.get(`${URL_API}/barang`, {
        headers: {
          "Content-Type": "application/json",
          Authorization: `Bearer ${localStorage.getItem("token")}`,
        },
      });
      if (response.data.code) {
        const allData = response.data.data;
        setDataBarang(allData);
      }
    } catch (error) {
      console.log(error);
    }
  };

  const handleSearch = (value) => {
    setValueSearch(value);
  };

  const handleOpenDialog = () => {
    setOpenDialog(true);
  };

  const handleCloseDialog = () => {
    setOpenDialog(false);
  };

  const handleReload = () => {
    setReloadAll(true);
    setTimeout(() => {
      setReloadAll(false);
    }, 500);
  };

  useEffect(() => {
    getAllData();
  }, []);
  return (
    <main
      style={{
        flexGrow: 1,
        minHeight: "100vh",
        overflow: "auto",
        background: "#FAFAFA",
      }}
    >
      <Container sx={{ pt: 10 }}>
        <Grid container sx={{ pt: 5, mt: 5 }}>
          <AddBarang
            open={openDialog}
            closeDialog={() => handleCloseDialog()}
            processAdd={() => handleReload()}
          />
          <div
            style={{
              minWidth: "100%",
            }}
          >
            <CardStyle
              style={{
                maxWidth: "100%",
                height: 80,
                display: "flex",
                justifyContent: "space-between",
              }}
            >
              <div>
                <FormSearch
                  onKeywordSearchChange={(value) => handleSearch(value)}
                />
              </div>

              {/* <div>
                <Button onClick={() => generatePDF(dataBarang)}>
                  Generate Laportan
                </Button>
              </div> */}
              <div>
                <Button
                  sx={{
                    width: 150,
                    backgroundColor: "#FE2E12",
                    height: "100%",
                    textTransform: "capitalize",
                    borderRadius: 2,
                    color: "#ffff",

                    "&:hover": {
                      backgroundColor: "#FE2E12",
                      color: "#ffff",
                    },
                  }}
                  startIcon={<AddRoundedIcon />}
                  onClick={() => handleOpenDialog()}
                >
                  Tambah Barang
                </Button>
              </div>
            </CardStyle>

            {/* Table */}
            <CardStyle style={{ marginTop: 32 }}>
              <Table keywordChange={valueSearch} reload={reloadAll} />
            </CardStyle>
          </div>
        </Grid>
      </Container>
    </main>
  );
};

export default MainContentSiswa;
```

Component/dashboard/CardCountData.js

```
import { Typography } from "@mui/material";
import React from "react";
import { styled } from "@mui/styles";
import EvaIcon from "react-eva-icons";
import { motion } from "framer-motion";

const CardStyle = styled("div")(({ theme }) => ({
  margin: "0px 12px",
  padding: 24,
  minHeight: 200,
  borderRadius: 16,
}));

const cardRedVariant = {
  hidden: {
    x: "-100vw",
    opacity: 0,
  },
  visible: {
    x: 0,
    opacity: 1,
    transition: {
      type: "spring",
      stiffness: 50,
    },
  },
};

const cardWhiteVariant = {
  hidden: {
    opacity: 0,
  },
  visible: {
    opacity: 1,
    transition: {
      delay: 0.9,
      type: "spring",
      stiffness: 50,
    },
  },
};

const CardCountData = () => {
  return (
    <div
      style={{
        display: "flex",
        padding: "24px 0px",
      }}
    >
      <motion.div variants={cardRedVariant} initial="hidden"
animate="visible">
        <CardStyle
          style={{
            width: 400,
            backgroundColor: "#FE2E12",
            boxShadow: "0px 10px 20px #FE2E1240",
          }}
        >
          <EvaIcon name="shopping-cart-outline" size="xlarge" fill="#ffff"
/>
          <Typography color="#ffff" sx={{ marginTop: 2, fontWeight: 500 }}>
            Barang yang sedang dilelang
          </Typography>
          <div
            style={{
              display: "flex",
              width: "100%",
              justifyContent: "space-between",
              alignItems: "flex-end",
            }}
          >
            <Typography
              color="#ffff"
              sx={{
                marginTop: 3,
                fontWeight: 500,
              }}
              variant="h3"
            >
              9861
            </Typography>
            <EvaIcon name="arrow-forward" size="large" fill="#ffff" />
          </div>
        </CardStyle>
      </motion.div>

      <motion.div
        variants={cardWhiteVariant}
        initial="hidden"
        animate="visible"
      >
        <CardStyle
          style={{
            width: 152,
            backgroundColor: "#ffff",
            boxShadow: "0px 10px 20px #5E5E5E0A",
          }}
        >
          <EvaIcon name="swap" size="xlarge" fill="#18125C" />
          <Typography
            color="#18125C"
            sx={{ marginTop: 1, fontWeight: 500, opacity: 0.5 }}
          >
            Jumlah Transaksi
          </Typography>
          <div
            style={{
              display: "flex",
              width: "100%",
              justifyContent: "space-between",
              alignItems: "flex-end",
            }}
          >
            <Typography
              color="#18125C"
              sx={{ marginTop: 3, fontWeight: 500 }}
              variant="h4"
            >
              981
            </Typography>
            <EvaIcon name="arrow-forward" size="medium" fill="#18125C" />
          </div>
        </CardStyle>
      </motion.div>

      <motion.div
        variants={cardWhiteVariant}
        initial="hidden"
        animate="visible"
```

```
      >
        <CardStyle
          style={{
            width: 152,
            backgroundColor: "#ffff",
            boxShadow: "0px 10px 20px #5E5E5E0A",
          }}
        >
          <EvaIcon name="shopping-bag-outline" size="xlarge" fill="#18125C"
/>
          <Typography
            color="#18125C"
            sx={{ marginTop: 1, fontWeight: 500, opacity: 0.5 }}
          >
            Jumlah Barang
          </Typography>
          <div
            style={{
              display: "flex",
              width: "100%",
              justifyContent: "space-between",
              alignItems: "flex-end",
            }}
          >
            <Typography
              color="#18125C"
              sx={{ marginTop: 3, fontWeight: 500 }}
              variant="h4"
            >
              67
            </Typography>
            <EvaIcon name="arrow-forward" size="medium" fill="#18125C" />
          </div>
        </CardStyle>
      </motion.div>
    </div>
  );
};

export default CardCountData;
```

Component/dashboard/history.js

```
import React from "react";
import { Typography } from "@mui/material";
import img1 from "../../assets/imageProfile/profile1.jpg";

const History = () => {
  return (
    <>
```

```
      <div style={{ padding: "0px 12px", margin: "24px 0px" }}>
        <div>
          <Typography variant="h5" sx={{ fontWeight: 600 }} color="#3d3d3d">
            History
          </Typography>
          <Typography color="#A5A5A5">Transaction of 2 months</Typography>
        </div>

        <div style={{ marginTop: 8 }}>
          <div className="historyCard">
            <div>
              <img
                src={img1}
                width={40}
                style={{ borderRadius: "50%" }}
                alt=""
              />
            </div>
            <div className="historyCol" id="name">
              Satmika Antargata
            </div>
            <div className="historyCol" id="date">
              16 Januari 2021
            </div>
            <div className="historyCol" id="price">
              IDR 500.000
            </div>
            <div className="historyCol" id="completed">
              Completed
            </div>
          </div>

          <div className="historyCard">
            <div>
              <img
                src={img1}
                width={40}
                style={{ borderRadius: "50%" }}
                alt=""
              />
            </div>
            <div className="historyCol" id="name">
              Juned Supri
            </div>
            <div className="historyCol" id="date">
              16 Januari 2021
            </div>
            <div className="historyCol" id="price">
              IDR 500.000
            </div>
            <div className="historyCol" id="completed">
              Completed
            </div>
          </div>
        </div>
      </div>
    </>
  );
};

export default History;
```

Component/dashboard/CardUser.js

```
import React from "react";
import { Grid, Typography } from "@mui/material";
import img1 from "../../assets/imageProfile/profile1.jpg";
import { motion } from "framer-motion";

import { styled } from "@mui/styles";

const CardStyle = styled("div")(({ theme }) => ({
  padding: 24,
  minHeight: 200,
  borderRadius: 16,
}));

const infoVariant = {
  hidden: {
    opacity: 0,
  },
  visible: {
    opacity: 1,
    transition: {
      delay: 0.9,
      type: "spring",
    },
  },
};

const CardUser = () => {
  const dataLogin = JSON.parse(localStorage.getItem("userLogin"));
  return (
    <>
      <CardStyle
        style={{
          backgroundColor: "#ffff",
          boxShadow: "0px 10px 20px #5E5E5E0A",
          alignItems: "center",
          display: "flex",
          flexDirection: "column",
          justifyContent: "space-between",
        }}
      >
        <div
          style={{
            width: "100%",
            display: "flex",
            justifyContent: "center",
          }}
        >
          <img
            src={img1}
            width={80}
            style={{ borderRadius: "50%" }}
            alt="img"
          />
        </div>
        <Grid
          container
          justifyContent={"space-between"}
          sx={{ marginTop: "auto" }}
        >
          <motion.div
            variants={infoVariant}
            initial="hidden"
            animate="visible"
            className="admin-username"
          >
            <Typography id="title-user">Username</Typography>
            <Typography id="text-user">{dataLogin.username}</Typography>
          </motion.div>
          <motion.div
            className="admin-role"
            variants={infoVariant}
            initial="hidden"
            animate="visible"
          >
            <Typography id="title-user">Level</Typography>
            <Typography id="text-user">{dataLogin.level}</Typography>
          </motion.div>
        </Grid>
      </CardStyle>
    </>
  );
};

export default CardUser;
```

Dialog/AddLelang.js

```
import React, { useEffect, useState } from "react";
import { useFormik } from "formik";
import * as yup from "yup";
import { URL_API } from "../../config/index";
import axios from "axios";
import {
  Dialog,
  DialogTitle,
  DialogContent,
  TextField,
  DialogActions,
  Button,
  Autocomplete,
  FormControl,
  InputLabel,
  Select,
  MenuItem,
} from "@mui/material";

import format from "date-fns/format";
import DateTimePicker from "@mui/lab/DateTimePicker";

const AddLelang = (props) => {
  const [status, setStatus] = useState("");
  const [open, setOpen] = useState(false);

  const [listBarang, setListBarang] = useState([]);

  const getListBarang = async () => {
    try {
      const response = await axios.get(`${URL_API}/barang`, {
        headers: {
          "Content-Type": "application/json",
          Authorization: `Bearer ${localStorage.getItem("token")}`,
        },
      });
      if (response.data.code === 200) {
        const allData = response.data.data;
        setListBarang(allData);
      }
    } catch (error) {
      console.log(error);
    }
  };

  const handleClose = () => {
    setOpen(false);
  };

  const handleOpen = () => {
    setOpen(true);
  };

  const fetchAddLelang = async (values, resetForm) => {
    const payload = {
      ...values,
      endTime: values.endTime + ".000Z",
    };
    try {
      const response = await axios.post(`${URL_API}/lelang`, payload, {
        headers: {
          "Content-Type": "application/json",
          Authorization: `Bearer ${localStorage.getItem("token")}`,
        },
      });

      console.log(response);
      if (response.data.code === 200 || response.data.code === 201) {
        props.processAdd();
        props.closeDialog();
      }
    } catch (error) {
      console.log(error);
    }

    resetForm();
  };

  const dateNow = format(new Date(Date.now()), "yyyy-MM-
dd'T'HH:mm:ss.SSSxxx");

  const validationSchema = yup.object({
    idBarang: yup
      .string("Enter your idBarang")
      .required("Pilih salah satu barang"),
    status: yup.string("Enter your status").required("Pilih status"),
    endTime: yup.string("Enter your date"),
  });

  const formik = useFormik({
    enableReinitialize: true,
    initialValues: {
      idBarang: "",
      idPetugas: JSON.parse(localStorage.getItem("userLogin")).id,
      status: "",
      endTime: "",
    },
    validationSchema: validationSchema,
    onSubmit: (values, { resetForm }) => {
      fetchAddLelang(values, resetForm);
    },
  });

  useEffect(() => {
    getListBarang();
  }, []);

  return (
    <>
      <div>
        <Dialog open={props.open} onClose={props.closeDialog} maxWidth="xs">
          <form onSubmit={formik.handleSubmit}>
            <DialogTitle>Tambah Barang Lelang</DialogTitle>
            <DialogContent sx={{ paddingTop: 4 }}>
              <div className="layout-input-barang" style={{ marginTop: 8 }}>
                <Autocomplete
                  id="combo-box-demo"
                  options={listBarang}
                  getOptionLabel={(options) => options.nama}
                  onChange={(event, value) => {
                    if (value) {
                      formik.setFieldValue("idBarang", value.id);
                    } else if (value === null) {
                      formik.setFieldValue("idBarang", "");
                    }
                  }}
                  renderInput={(params) => (
                    <TextField
                      {...params}
                      label="Barang"
                      variant="outlined"
                      size="small"
                      error={
                        formik.touched.idBarang &&
                        Boolean(formik.errors.idBarang)
                      }
                      helperText={
                        formik.touched.idBarang && formik.errors.idBarang
                      }
                    />
                  )}
                />
              </div>

              <div className="layout-input-barang" style={{ marginTop: 12
}}>
                <FormControl sx={{ minWidth: 120 }} size="small" fullWidth>
                  <InputLabel id="demo-controlled-open-select-label">
                    Status
                  </InputLabel>
                  <Select
                    labelId="demo-controlled-open-select-label"
                    id="demo-controlled-open-select"
                    open={open}
                    onClose={handleClose}
                    onOpen={handleOpen}
                    value={formik.values.status}
                    label="Status"
                    onChange={(event) => {
                      formik.setFieldValue("status", event.target.value);
                      setStatus(event.target.value);
          }}
         >
          <MenuItem value="Ditutup">Ditutup</MenuItem>
          <MenuItem value="Dibuka">Dibuka</MenuItem>
         </Select>
        </FormControl>
       </div>

       {status === "Dibuka" && (
        <div className="layout-input-barang" style={{ marginTop: 12
}}>
         <DateTimePicker
          ampm
          label="Tanggal ditutup lelang"
          onChange={(newValue) => {
           if (newValue) {
            formik.setFieldValue(
             "endTime",
             format(newValue, "yyyy-MM-dd'T'HH:mm:ss")
            );
           } else if (newValue === null) {
            formik.setFieldValue("endTime", "");
           }
          }}
          value={new Date(formik.values.endTime)}
          renderInput={(params) => (
           <TextField
            {...params}
            fullWidth
            size="small"
            error={
             formik.touched.endTime &&
             Boolean(formik.errors.endTime)
            }
            helperText={
             formik.touched.endTime && formik.errors.endTime
            }
           />
          )}
         />
        </div>
       )}
      </DialogContent>
      <DialogActions>
       <Button onClick={props.closeDialog}>Cancel</Button>
       <Button type="submit">Tambah</Button>
      </DialogActions>
     </form>
    </Dialog>
   </div>
  </>
 );
};
```

```
export default AddLelang;
```

Dialog/AddBarang.js

```
import React, { useState } from "react";
import { useFormik } from "formik";
import { styled } from "@mui/material/styles";
import * as yup from "yup";
import { URL_API } from "../../config/index";
import axios from "axios";
import {
  Dialog,
  DialogTitle,
  DialogContent,
  TextField,
  DialogActions,
  Button,
} from "@mui/material";

import UploadFileRoundedIcon from "@mui/icons-material/UploadFileRounded";

const Input = styled("input")({
  display: "none",
});

const AddBarang = (props) => {
  const [alert, setAlert] = useState({
    show: false,
    message: null,
  });

  const fetchAddBarang = async (values, resetForm) => {
    let bodyFormData = new FormData();

    bodyFormData.append("nama", values.nama);
    bodyFormData.append("hargaAwal", values.hargaAwal);
    bodyFormData.append("deskripsi", values.deskripsi);
    bodyFormData.append("image", values.image);
    try {
      const response = await axios.post(`${URL_API}/barang`, bodyFormData, {
        headers: {
          "Content-Type": "multipart/form-data",
          Authorization: `Bearer ${localStorage.getItem("token")}`,
        },
      });
      if (response.data.code === 200) {
        props.processAdd();
        props.closeDialog();
      }
    } catch (error) {
      console.log(error);
      console.log("ERROR");
    }

    resetForm();
  };

  const validationSchema = yup.object({
    nama: yup.string("Enter your username").required("Username is
required"),
    hargaAwal: yup
      .string("Enter your password")
      .matches(/^[0-9.]+$/, `Only numbers (1-9)`)
      .required("Password is required"),
    deskripsi: yup
      .string("Enter your deskripsi")
      .required("Deskripsi is required"),
    image: yup.string("Enter your Image").required("Image is required"),
  });

  const formik = useFormik({
    enableReinitialize: true,
    initialValues: {
      nama: "",
      hargaAwal: "",
      deskripsi: "",
      image: "",
    },
    validationSchema: validationSchema,
    onSubmit: (values, { resetForm }) => {
      fetchAddBarang(values, resetForm);
    },
  });

  return (
    <>
      <div>
        <Dialog
          open={props.open}
          onClose={props.closeDialog}
          width="xs"
          maxWidth={"xs"}
        >
          <form onSubmit={formik.handleSubmit}>
            <DialogTitle>Tambah Barang</DialogTitle>
            <DialogContent>
              <div className="layout-input-barang">
                <TextField
                  type="text"
                  autoFocus
                  margin="dense"
                  variant="outlined"
                  fullWidth
                  id="nama"
                  label="Nama Barang"
                  name="nama"
                  size="small"
                  value={formik.values.nama || ""}
                  onChange={formik.handleChange}
                  error={formik.touched.nama && Boolean(formik.errors.nama)}
                  helperText={formik.touched.nama && formik.errors.nama}
                />
              </div>

              <div className="layout-input-barang">
                <TextField
                  type="text"
                  margin="dense"
                  variant="outlined"
                  fullWidth
                  id="hargaAwal"
                  label="Harga Awal"
                  name="hargaAwal"
                  size="small"
                  value={formik.values.hargaAwal || ""}
                  onChange={formik.handleChange}
                  error={
                    formik.touched.hargaAwal &&
Boolean(formik.errors.hargaAwal)
                  }
                  helperText={
                    formik.touched.hargaAwal && formik.errors.hargaAwal
                  }
                />
              </div>

              <div className="layout-input-barang">
                <TextField
                  multiline
                  type="text"
                  margin="dense"
                  variant="outlined"
                  fullWidth
                  rows={4}
                  id="deskripsi"
                  label="Deskripsi"
                  name="deskripsi"
                  value={formik.values.deskripsi || ""}
                  onChange={formik.handleChange}
                  error={
                    formik.touched.deskripsi &&
Boolean(formik.errors.deskripsi)
                  }
                  helperText={
                    formik.touched.deskripsi && formik.errors.deskripsi
                  }
                />
              </div>

              <div className="layout-input-barang" style={{ marginTop: 8 }}>
                <label htmlFor="contained-button-file">
                  <Input
                    accept="image/*"
                    id="contained-button-file"
                    multiple
                    type="file"
                    onChange={(event) => {
                      formik.setFieldValue("image", event.target.files[0]);
                    }}
                  />
                  <Button
                    variant="contained"
                    component="span"
                    startIcon={<UploadFileRoundedIcon />}
                  >
                    {formik.values.image === "" && "Upload File"}
                    {typeof formik.values.image === "object" &&
                      formik.values.image.name}
                  </Button>
                </label>
              </div>
            </DialogContent>
            <DialogActions>
              <Button onClick={props.closeDialog}>Cancel</Button>
              <Button type="submit">Tambah</Button>
            </DialogActions>
          </form>
        </Dialog>
      </div>
    </>
  );
};

export default AddBarang;
```

Dialog/EditBarang.js

```
import React from "react";
import { useFormik } from "formik";
import * as yup from "yup";
import { styled } from "@mui/material/styles";

import {
  Dialog,
  DialogTitle,
  DialogContent,
  TextField,
  DialogActions,
  Button,
} from "@mui/material";

import UploadFileRoundedIcon from "@mui/icons-material/UploadFileRounded";

const Input = styled("input")({
  display: "none",
});

const EditBarang = (props) => {
  let dataBarang = props.data;

  const fetchEditBarang = async (values) => {
    let bodyFormData = new FormData();

    bodyFormData.append("id", values.id);
    bodyFormData.append("nama", values.nama);
    bodyFormData.append("tgl", values.tgl);
    bodyFormData.append("hargaAwal", values.hargaAwal);
    bodyFormData.append("deskripsi", values.deskripsi);
    bodyFormData.append("image", values.image);

    props.processEdit(bodyFormData);
  };

  const validationSchema = yup.object({
    nama: yup.string("Enter your username").required("Username is
required"),
    hargaAwal: yup
      .string("Enter your password")
      .matches(/^[0-9.]+$/, `Only numbers (1-9)`)
      .required("Password is required"),
    deskripsi: yup
      .string("Enter your deskripsi")
      .required("Deskripsi is required"),
    image: yup.string("Enter your deskripsi").required("Deskripsi is
required"),
  });

  const formik = useFormik({
    enableReinitialize: true,
    initialValues: {
      id: dataBarang.id,
      nama: dataBarang.nama,
      hargaAwal: dataBarang.hargaAwal,
      tgl: dataBarang.tgl,
      deskripsi: dataBarang.deskripsi,
      image: dataBarang.image,
    },
    validationSchema: validationSchema,
    onSubmit: (values) => {
      fetchEditBarang(values);
    },
  });

  return (
    <>
      <Dialog open={props.open} onClose={props.closeDialog} maxWidth="xs">
        <form onSubmit={formik.handleSubmit}>
          <DialogTitle>Tambah Barang</DialogTitle>
          <DialogContent>
            <div className="layout-input-barang">
              <TextField
                type="text"
                autoFocus
                margin="dense"
                variant="outlined"
                fullWidth
                id="nama"
                label="Nama Barang"
                name="nama"
                size="small"
                value={formik.values.nama || ""}
                onChange={formik.handleChange}
                error={formik.touched.nama && Boolean(formik.errors.nama)}
                helperText={formik.touched.nama && formik.errors.nama}
              />
            </div>

            <div className="layout-input-barang">
              <TextField
                type="text"
                margin="dense"
                variant="outlined"
                fullWidth
                id="hargaAwal"
                label="Harga Awal"
                name="hargaAwal"
                size="small"
                value={formik.values.hargaAwal || ""}
                onChange={formik.handleChange}
                error={
                  formik.touched.hargaAwal &&
Boolean(formik.errors.hargaAwal)
                }
                helperText={formik.touched.hargaAwal &&
formik.errors.hargaAwal}
              />
            </div>

            <div className="layout-input-barang">
              <TextField
                type="text"
                margin="dense"
                variant="outlined"
                fullWidth
                InputProps={{ readOnly: true }}
                className="readOnly"
                id="tgl"
                label="Tanggal Penambahan"
                name="tgl"
                size="small"
                value={formik.values.tgl || ""}
                onChange={formik.handleChange}
                error={formik.touched.tgl && Boolean(formik.errors.tgl)}
                helperText={formik.touched.tgl && formik.errors.tgl}
              />
            </div>

            <div className="layout-input-barang">
              <TextField
                multiline
                type="text"
                margin="dense"
                variant="outlined"
                fullWidth
                rows={4}
                id="deskripsi"
                label="Deskripsi"
                name="deskripsi"
                value={formik.values.deskripsi || ""}
                onChange={formik.handleChange}
                error={
                  formik.touched.deskripsi &&
Boolean(formik.errors.deskripsi)
                }
                helperText={formik.touched.deskripsi &&
formik.errors.deskripsi}
              />
            </div>

            <div className="layout-input-barang" style={{ marginTop: 8 }}>
              <label htmlFor="contained-button-file">
                <Input
                  accept="image/*"
                  id="contained-button-file"
                  multiple
                  type="file"
                  onChange={(event) => {
                    formik.setFieldValue("image", event.target.files[0]);
                  }}
                />
                <Button
                  variant="contained"
                  component="span"
                  startIcon={<UploadFileRoundedIcon />}
```

```
                >
                  {typeof formik.values.image === "string" &&
                    formik.values.image}
                  {typeof formik.values.image === "undefined" && "Upload
File"}
                  {typeof formik.values.image === "object" &&
                    formik.values.image.name}
                </Button>
              </label>
            </div>
          </DialogContent>
          <DialogActions>
            <Button onClick={props.closeDialog}>Cancel</Button>
            <Button type="submit">Edit</Button>
          </DialogActions>
        </form>
      </Dialog>
    </>
  );
};

export default EditBarang;
```

Dialog/DeleteBarang.js

```
import React from "react";
import {
  Button,
  Dialog,
  DialogActions,
  DialogContent,
  DialogContentText,
  DialogTitle,
} from "@mui/material";

const DeleteBarang = (props) => {
  const { open, closeDialog, processDelete } = props;

  return (
    <>
      <Dialog
        open={open}
        onClose={closeDialog}
        aria-labelledby="alert-dialog-title"
        aria-describedby="alert-dialog-description"
      >
        <DialogTitle id="alert-dialog-title">Hapus Barang</DialogTitle>
        <DialogContent>
          <DialogContentText id="alert-dialog-description">
            Are you sure want to delete this list? This action can not be
            undone.
          </DialogContentText>
        </DialogContent>
        <DialogActions
          sx={{ background: "#F1F5F5", padding: "12px", paddingRight: "20px"
}}
        >
          <Button onClick={closeDialog}>Cancel</Button>
          <Button
            color="error"
            variant="contained"
            onClick={processDelete}
            sx={{ width: "110px", boxShadow: "none", letterSpacing: "0.5px"
}}
          >
            Delete
          </Button>
        </DialogActions>
      </Dialog>
    </>
  );
};

export default DeleteBarang;
```

Component/search.js

```
import React from "react";
import { styled } from "@mui/material/styles";
import SearchIcon from "@mui/icons-material/Search";
import InputBase from "@mui/material/InputBase";

const Search = styled("div")(({ theme }) => ({
  position: "relative",
  borderRadius: 8,
  backgroundColor: "#F8F8F8",
  "&:hover": {
    backgroundColor: "#F8F8F8",
  },
  marginLeft: 0,
  width: "100%",
  [theme.breakpoints.up("sm")]: {
    marginLeft: theme.spacing(1),
    width: "auto",
  },
}));

const SearchIconWrapper = styled("div")(({ theme }) => ({
  padding: theme.spacing(0, 2),
  height: "100%",
  position: "absolute",
  pointerEvents: "none",
  display: "flex",
  alignItems: "center",
  justifyContent: "center",
}));

const StyledInputBase = styled(InputBase)(({ theme }) => ({
  color: "inherit",
  "& .MuiInputBase-input": {
    padding: theme.spacing(1, 1, 1, 0),
    // vertical padding + font size from searchIcon
    paddingLeft: `calc(1em + ${theme.spacing(4)})`,
    transition: theme.transitions.create("width"),
    width: "100%",
    [theme.breakpoints.up("sm")]: {
      width: "32ch",
      "&:focus": {
        width: "56ch",
      },
    },
  },
}));

const FormSearch = (props) => {
  return (
    <>
      <Search>
        <SearchIconWrapper>
          <SearchIcon />
        </SearchIconWrapper>
        <StyledInputBase
          onChange={(e) => {
            props.onKeywordSearchChange(e.target.value);
          }}
          placeholder="Search…"
          inputProps={{ "aria-label": "search" }}
        />
      </Search>
    </>
  );
};

export default FormSearch;
```

Components/table/tableLelang.js

```
import React, { useEffect, useState } from "react";
import {
  Paper,
  TableContainer,
  Table,
  TableHead,
  TableRow,
  TableCell,
  TableBody,
  TableFooter,
} from "@mui/material";

import axios from "axios";
import { URL_API } from "../../config";
import DataRowTable from "../list/RowLelang";

const TableData = (props) => {
  const [dataLelang, setDataLelang] = useState([]);

  let keywordChange = props.keywordChange;

  const getAllData = async () => {
    try {
      const response = await axios.get(`${URL_API}/lelang`, {
        headers: {
          "Content-Type": "application/json",
          Authorization: `Bearer ${localStorage.getItem("token")}`,
        },
      });
      if (response.data.code) {
        const allData = response.data.data;
        // allData.forEach((data) => (data.id += "") && (data.hargaAwal +=
""));

        setDataLelang(allData);
      }
    } catch (error) {
      console.log(error);
    }
  };

  const filterSearch = dataLelang.filter((data) => {
    return Object.keys(data).some((key) =>

data["barang"].nama.toLowerCase().includes(keywordChange.toLowerCase())
    );
  });

  useEffect(() => {
    getAllData();
  }, [props.reload]);

  return (
    <>
```

```
      <Paper sx={{ width: "100%", overflow: "hidden", boxShadow: "none" }}>
        <TableContainer sx={{ maxHeight: "55vh" }}>
          <Table aria-label="collapsible table">
            <TableHead>
              <TableRow>
                <TableCell />
                <TableCell>Nama Barang</TableCell>
                <TableCell>Harga Akhir</TableCell>
                <TableCell align="center">Tanggal Lelang</TableCell>
                <TableCell align="center">Status</TableCell>
                <TableCell align="center">Tanggal Ditutup</TableCell>
                <TableCell />
              </TableRow>
            </TableHead>
            <TableBody>
              {filterSearch
                .sort(function (a, b) {
                  var textA = a.barang.nama.toUpperCase();
                  var textB = b.barang.nama.toUpperCase();
                  return textA < textB ? -1 : textA > textB ? 1 : 0;
                })
                .map((data, i) => (
                  <DataRowTable
                    data={data}
                    key={i}
                    index={i}
                    reloadChanges={() => getAllData()}
                  />
                ))}
            </TableBody>
            <TableFooter></TableFooter>
          </Table>
        </TableContainer>
      </Paper>
    </>
  );
};

export default TableData;
```

Components/table/TableBarang.js

```
import React, { useEffect, useState } from "react";
import {
  Paper,
  TableContainer,
  Table,
  TableHead,
  TableRow,
  TableCell,
  TableBody,
  TableFooter,
} from "@mui/material";

import axios from "axios";
import { URL_API } from "../../config";
import DataRowTable from "../list/RowBarang";

const TableData = (props) => {
  const [dataBarang, setDataBarang] = useState([]);

  let keywordChange = props.keywordChange;

  const getAllData = async () => {
    try {
      const response = await axios.get(`${URL_API}/barang`, {
        headers: {
          "Content-Type": "application/json",
          Authorization: `Bearer ${localStorage.getItem("token")}`,
        },
      });
      if (response.data.code) {
        const allData = response.data.data;
        allData.forEach((data) => (data.id += "") && (data.hargaAwal +=
""));

        setDataBarang(allData);
      }
    } catch (error) {
      console.log(error);
    }
  };

  const filterSearch = dataBarang.filter((data) => {
    return Object.keys(data).some((key) =>
      data["nama"].toLowerCase().includes(keywordChange.toLowerCase())
    );
  });

  useEffect(() => {
    getAllData();
  }, [props.reload]);

  return (
    <>
      <Paper sx={{ width: "100%", overflow: "hidden", boxShadow: "none" }}>
        <TableContainer sx={{ maxHeight: "55vh" }}>
          <Table aria-label="collapsible table">
            <TableHead>
              <TableRow>
                <TableCell />
                <TableCell>Nama Barang</TableCell>
                <TableCell>Harga Awal</TableCell>
                <TableCell align="center">Tanggal Penambahan</TableCell>
                <TableCell align="center">Status</TableCell>
                <TableCell />
              </TableRow>
            </TableHead>
            <TableBody>
              {filterSearch
                .sort(function (a, b) {
                  var textA = a.nama.toUpperCase();
                  var textB = b.nama.toUpperCase();
                  return textA < textB ? -1 : textA > textB ? 1 : 0;
                })
                .map((data, i) => (
                  <DataRowTable
                    data={data}
                    key={i}
                    index={i}
                    reloadChanges={() => getAllData()}
                  />
                ))}
            </TableBody>
            <TableFooter></TableFooter>
          </Table>
        </TableContainer>
      </Paper>
    </>
  );
};

export default TableData;
```

Component/list/rowLelang.js

```
import React, { useState } from "react";

import { Typography, TableRow, TableCell } from "@mui/material";
import EvaIcon from "react-eva-icons";

import { tableCellClasses } from "@mui/material/TableCell";

import Badge from "react-bootstrap/Badge";
import axios from "axios";
import { URL_API } from "../../config";
import DeleteBarang from "../dialog/DeleteBarang";
import EditBarang from "../dialog/EditBarang";

const RowTableBody = (props) => {
  const { data } = props;
  const [openDelete, setOpenDelete] = useState(false);
  const [openEdit, setOpenEdit] = useState(false);
  const [itemSelect, setItemSelect] = useState("");

  const handleCloseDelete = () => {
    setOpenDelete(false);
  };

  const handleCloseEdit = () => {
    setOpenEdit(false);
  };

  const handleDeleteBarang = (id) => {
    setOpenDelete(true);
    setItemSelect(id);
  };

  const handleEditBarang = (data) => {
    setOpenEdit(true);
    setItemSelect(data);
  };

  const fetchDeleteBarang = async () => {
    try {
      const response = await axios.delete(`${URL_API}/barang/${itemSelect}`,
{
        headers: {
          "Content-Type": "multipart/form-data",
          Authorization: `Bearer ${localStorage.getItem("token")}`,
        },
      });

      if (response.data.code === 200) {
        setItemSelect("");
        props.reloadChanges();
        handleCloseDelete();
      }
    } catch (error) {
      console.log(error);
    }
  };

  const fetchEditBarang = async (payload) => {
    try {
      const response = await axios.put(`${URL_API}/barang`, payload, {
        headers: {
          "Content-Type": "multipart/form-data",
          Authorization: `Bearer ${localStorage.getItem("token")}`,
        },
      });

      if (response.data.code === 200) {
        setItemSelect("");
        props.reloadChanges();
        handleCloseEdit();
      }
    } catch (error) {
      console.log(error);
    }
  };

  return (
    <>
      <EditBarang
        open={openEdit}
        data={itemSelect}
        closeDialog={() => handleCloseEdit()}
        processEdit={(payload) => fetchEditBarang(payload)}
      />
      <DeleteBarang
        open={openDelete}
        closeDialog={() => handleCloseDelete()}
        processDelete={() => fetchDeleteBarang()}
      />
      <TableRow
        sx={{
          [`& .${tableCellClasses.root}`]: {
            borderBottom: "none",
          },
        }}
      >
        <TableCell>
          <Typography sx={{ fontWeight: 500 }}>{props.index +
1}</Typography>
        </TableCell>
        <TableCell component="th" scope="row">
          <Typography sx={{ fontWeight: 500
}}>{data.barang.nama}</Typography>
        </TableCell>
        <TableCell>
          <Typography
            sx={{ fontWeight: 500 }}
          >{`Rp.${data.hargaAkhir}`}</Typography>
        </TableCell>
        <TableCell align="center">
          <Typography sx={{ fontWeight: 500 }}>
{data.tglLelang}</Typography>
        </TableCell>
        <TableCell align="center">
          {data.status === "Ditutup" ? (
            <Badge bg="light" text="danger">
              Ditutup
            </Badge>
          ) : (
            <Badge bg="light" text="warning">
              Dibuka
            </Badge>
          )}
        </TableCell>
        <TableCell align="center">
          <Typography sx={{ fontWeight: 500 }}>
            {data.endTime === null ? "-- -- -- --" : data.endTime}
          </Typography>
        </TableCell>
        <TableCell align="center">
          <div style={{ display: "flex", alignItems: "center" }}>
            <div className="btn-act" onClick={() => handleEditBarang(data)}>
              <EvaIcon name="edit-2-outline" size="normal" fill="#FE2E12" />
            </div>
            <div
              className="btn-act"
              onClick={() => handleDeleteBarang(data.id)}
            >
              <EvaIcon name="trash-2-outline" size="normal" fill="#FE2E12"
/>
            </div>
          </div>
        </TableCell>
      </TableRow>
    </>
  );
};

export default RowTableBody;
```

Component/list/rowBarang.js

```
import React, { useEffect, useState } from "react";

import { Typography, TableRow, TableCell } from "@mui/material";
import EvaIcon from "react-eva-icons";

import { tableCellClasses } from "@mui/material/TableCell";

import Badge from "react-bootstrap/Badge";
import axios from "axios";
import { URL_API } from "../../config";
import DeleteBarang from "../dialog/DeleteBarang";
import EditBarang from "../dialog/EditBarang";

const RowTableBody = (props) => {
  const { data } = props;
  const [openDelete, setOpenDelete] = useState(false);
  const [openEdit, setOpenEdit] = useState(false);
  const [itemSelect, setItemSelect] = useState("");
  const [onLelang, setOnLelang] = useState(false);

  const getStatus = async () => {
    try {
      const response = await axios.get(
        `${URL_API}/lelang/idBarang/${data.id}`,
        {
          headers: {
            "Content-Type": "application/json",
            Authorization: `Bearer ${localStorage.getItem("token")}`,
          },
        }
      );
      if (response.data.data) {
        setOnLelang(true);
      } else {
        setOnLelang(false);
      }
    } catch (error) {
      setOnLelang(false);
      console.log(error);
    }
  };

  const handleCloseDelete = () => {
    setOpenDelete(false);
  };

  const handleCloseEdit = () => {
    setOpenEdit(false);
  };

  const handleDeleteBarang = (id) => {
    setOpenDelete(true);
    setItemSelect(id);
  };

  const handleEditBarang = (data) => {
    setOpenEdit(true);
    setItemSelect(data);
  };

  const fetchDeleteBarang = async () => {
    try {
      const response = await axios.delete(`${URL_API}/barang/${itemSelect}`,
{
        headers: {
          "Content-Type": "multipart/form-data",
          Authorization: `Bearer ${localStorage.getItem("token")}`,
        },
      });

      if (response.data.code === 200) {
        setItemSelect("");
        props.reloadChanges();
        handleCloseDelete();
      }
    } catch (error) {
      console.log(error);
    }
  };

  const fetchEditBarang = async (payload) => {
    try {
      const response = await axios.put(`${URL_API}/barang`, payload, {
        headers: {
          "Content-Type": "multipart/form-data",
          Authorization: `Bearer ${localStorage.getItem("token")}`,
        },
      });

      if (response.data.code === 200) {
        setItemSelect("");
        props.reloadChanges();
        handleCloseEdit();
      }
    } catch (error) {
      console.log(error);
    }
  };

  useEffect(() => {
    getStatus();
  });

  return (
    <>
      <EditBarang
        open={openEdit}
        data={itemSelect}
        closeDialog={() => handleCloseEdit()}
        processEdit={(payload) => fetchEditBarang(payload)}
      />
      <DeleteBarang
        open={openDelete}
        closeDialog={() => handleCloseDelete()}
        processDelete={() => fetchDeleteBarang()}
      />
      <TableRow
        sx={{
          [`& .${tableCellClasses.root}`]: {
            borderBottom: "none",
          },
        }}
      >
        <TableCell>
          <Typography sx={{ fontWeight: 500 }}>{props.index +
1}</Typography>
        </TableCell>
        <TableCell component="th" scope="row">
          <Typography sx={{ fontWeight: 500 }}>{data.nama}</Typography>
        </TableCell>
        <TableCell>
          <Typography
            sx={{ fontWeight: 500 }}
          >{`Rp.${data.hargaAwal}`}</Typography>
        </TableCell>
        <TableCell align="center">
          <Typography sx={{ fontWeight: 500 }}> {data.tgl}</Typography>
        </TableCell>
        <TableCell align="center">
          {!onLelang ? (
            <Badge bg="light" text="success">
              Tidak dilelang
            </Badge>
          ) : (
            <Badge bg="light" text="warning">
              Sedang dilelang
            </Badge>
          )}
        </TableCell>
        <TableCell align="center">
          <div style={{ display: "flex", alignItems: "center" }}>
            <div className="btn-act" onClick={() => handleEditBarang(data)}>
              <EvaIcon name="edit-2-outline" size="normal" fill="#FE2E12" />
            </div>
            <div
              className="btn-act"
              onClick={() => handleDeleteBarang(data.id)}
            >
              <EvaIcon name="trash-2-outline" size="normal" fill="#FE2E12"
/>
            </div>
          </div>
        </TableCell>
      </TableRow>
    </>
  );
};

export default RowTableBody;
```

Index.css

```
.historyCard {
  border-radius: 8px;
  display: flex;
  flex-wrap: wrap;
  align-items: center;
  padding: 12px;
  transition: 0.7s all ease;
}

.historyCard:hover {
  background-color: white;
}

.historyCol {
  color: grey;
  font-size: 18px;
  font-weight: 500;
  padding: 8px;
}

#name {
  width: 200px;
  margin-left: 24px;
}

#date {
  width: 170px;
}

#price {
  width: 150px;
  text-align: "center";
}

#completed {
  width: 170px;
  text-align: "center";
}

#title-user {
  color: rgb(187, 187, 187);
  font-size: 15px;
  font-weight: 500;
}

#text-user {
  font-size: 20px;
  font-weight: 600;
  text-transform: capitalize;
}

.textError {
  font-size: 12px;
  margin-left: 8px;
  opacity: 0.7;
}

.textMovePageLogin {
```

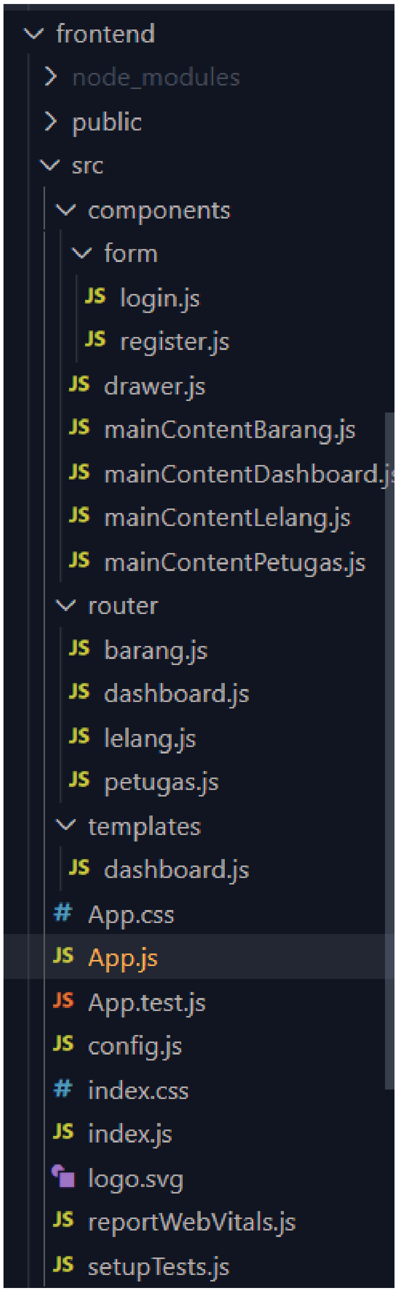
```css
  display: flex;
  margin-top: 16px;
  justify-content: center;
  font-size: 14px;
}

.btn-act {
  margin: 0px 8px;
  cursor: pointer;
}

.layout-input-barang {
  width: 552px;
  max-width: 100%;
}

.readOnly {
  background-color: #f0f0f0;
}
```

VERSI MIKA



```
frontend
  node_modules
  public
  src
    components
      form
        login.js
        register.js
      drawer.js
      mainContentBarang.js
      mainContentDashboard.j
      mainContentLelang.js
      mainContentPetugas.js
    router
      barang.js
      dashboard.js
      lelang.js
      petugas.js
    templates
      dashboard.js
    App.css
    App.js
    App.test.js
    config.js
    index.css
    index.js
    logo.svg
    reportWebVitals.js
    setupTests.js
```

Drawer.js

```
import React from "react";
import { Link } from "react-router-dom";
import {
  Drawer,
  ListItem,
  ListItemText,
  Divider,
  List,
  Box,
  ListItemIcon,
  Typography,
} from "@mui/material";
import HomeIcon from "@mui/icons-material/Home";
import Inventory2Icon from "@mui/icons-material/Inventory2";
import SellIcon from "@mui/icons-material/Sell";
import SupervisorAccountIcon from "@mui/icons-material/SupervisorAccount";
import LogoutIcon from "@mui/icons-material/Logout";
import { styled } from "@mui/styles";

const DivStyle = styled("div")(({ theme }) => ({
  width: "322px",
  height: "49px",
  display: "flex",
  backgroundColor: "#ffffff",
  borderRadius: "10px",
  alignItems: "center",
  marginBottom: "16px",
  paddingLeft: "20px",
  cursor: "pointer",
  fontSize: "18px",
  "&:hover": {
    backgroundColor: "#f7f7f7f7",
  },
}));

class drawer extends React.Component {
  handleSideBar = (item) => {
    const currentPath = this.props.history.location.pathname;
    const pageDestination = item;

    if (pageDestination !== currentPath) {
      return this.props.history.push(pageDestination);
    }
  };
  render() {
    const side = [
      {
        icon: <HomeIcon sx={{ color: "#000000" }} />,
        name: "Dashboard",
        url: "/dashboard",
      },
      {
        icon: <Inventory2Icon sx={{ color: "#000000" }} />,
        name: "Barang",
        url: "/barang",
      },
      {
        icon: <SellIcon sx={{ color: "#000000" }} />,
        name: "Lelang",
        url: "/lelang",
      },
      {
        icon: <SupervisorAccountIcon sx={{ color: "#000000" }} />,
        name: "Petugas",
        url: "/petugas",
      },
      {
        icon: <LogoutIcon sx={{ color: "#000000" }} />,
        name: "Logout",
        url: "/",
      },
    ];
    return (
      <Drawer
        sx={{
          width: "250px",
          flexShrink: 0,
          "& .MuiDrawer-paper": {
            width: "250px",
            boxSizing: "border-box",
          },
        }}
        variant="permanent"
        anchor="left"
      >
        <h1
          style={{
            color: "#125C13",
            fontFamily: "poppins",
            justifyContent: "center",
            display: "flex",
          }}
        >
          LELANG
        </h1>
        <List>
          {side.map((text, index) => (
            <ListItem
              key={text.name}
              onClick={() => this.handleSideBar(text.url)}
              sx={{ backgroundColor: "#ffffff" }}
            >
              <DivStyle>
                <ListItemIcon>{text.icon}</ListItemIcon>
                <ListItemText>
                  {" "}
                  <Typography
                    sx={{
                      fontSize: "14px",
                      color: "#000000",
                      fontFamily: "poppins",
                    }}
                  >
                    {text.name}
                  </Typography>
                </ListItemText>
              </DivStyle>
            </ListItem>
          ))}
        </List>
      </Drawer>
    );
  }
}

export default drawer;
{
  /* <ListItem onClick={() => this.handleSideBar('/dashboard')}
button><ListItemText primary={"Dashboard"}/></ListItem>
                    <ListItem onClick={() => this.handleSideBar('/barang')}
button>Barang</ListItem>
                    <ListItem onClick={() => this.handleSideBar('/Lelang')}
button>Lelang</ListItem>
                    <ListItem onClick={() => this.handleSideBar('/Petugas')}
button>Petugas</ListItem> */
}
```

MainContentDashboard.js

```
import React from "react";
import { Box, Container } from "@mui/material";

class mainContentDashboard extends React.Component {
  render() {
    return (
      <main
        style={{
          background: "#f7f7f7f7",
          minHeight: "100vh",
          flexGrow: 1,
          overflow: "auto",
        }}
      >
        <div
          style={{
            position: "absolute",
            height: "610px",
            width: "1190px",
            left: "300px",
            top: "50px",
            backgroundColor: "#cfe8fc",
          }}
        >
          <h1
```

```
            style={{
              marginLeft: "24px",
              marginTop: "24px",
              fontFamily: "poppins",
            }}
          >
            Welcome Mika Ozora
          </h1>
        </div>
      </main>
    );
  }
}
export default mainContentDashboard;
```

MainContentBarang

```
import React from "react";

class mainContentBarang extends React.Component{
    render(){
        return(
            <main
        style={{
          background: "#f7f7f7f7",
          minHeight: "100vh",
          flexGrow: 1,
          overflow: "auto",
        }}
      >
        <div
          style={{
            position: "absolute",
            height: "610px",
            width: "1190px",
            left: "300px",
            top: "50px",
            backgroundColor: "#cfe8fc",
          }}
        >
          <h1
            style={{
              marginLeft: "24px",
              marginTop: "24px",
              fontFamily: "poppins",
            }}
          >
            Welcome Barang
          </h1>
        </div>
      </main>
        )
    }
}
export default mainContentBarang
```

MainContentLelang

```
import React from "react";

class mainContentLelang extends React.Component{
    render(){
        return(
            <main
        style={{
          background: "#f7f7f7f7",
          minHeight: "100vh",
          flexGrow: 1,
          overflow: "auto",
        }}
      >
        <div
          style={{
            position: "absolute",
            height: "610px",
            width: "1190px",
            left: "300px",
            top: "50px",
            backgroundColor: "#cfe8fc",
          }}
        >
          <h1
            style={{
              marginLeft: "24px",
              marginTop: "24px",
              fontFamily: "poppins",
            }}
          >
            Welcome Lelang
          </h1>
        </div>
      </main>
        )
    }
}
export default mainContentLelang
```

MainContentPetugas

```
import React from "react";

class mainContentPetugas extends React.Component{
    render(){
        return(
            <main
        style={{
          background: "#f7f7f7f7",
          minHeight: "100vh",
          flexGrow: 1,
          overflow: "auto",
        }}
      >
        <div
          style={{
            position: "absolute",
            height: "610px",
            width: "1190px",
            left: "300px",
            top: "50px",
            backgroundColor: "#cfe8fc",
          }}
        >
          <h1
            style={{
              marginLeft: "24px",
              marginTop: "24px",
              fontFamily: "poppins",
            }}
          >
            Welcome Petugas
          </h1>
        </div>
      </main>
        )
    }
}
export default mainContentPetugas
```

Folder Form

Login.js

```
import React, { useState } from "react";
import { Card, Typography, Button, Stack, Snackbar } from "@mui/material";
import { styled } from "@mui/styles";
import { useFormik } from "formik";
import * as yup from "yup";
import axios from "axios";
import { useHistory } from "react-router";
import { url } from "../../config";

const InputStyle = styled("input")(({ theme }) => ({
  width: "300px",
  height: "40px",
  backgroundColor: "#f7f7f7",
  borderRadius: "5px",
  margin: "0px 0px 8px 40px",
  border: "none",
  paddingLeft: 16,
  fontFamily: "poppins",
  fontWeight: 400,
  "&:focus": {
    border: "none",
    outline: "none",
  },
}));

const Login = (props) => {
  const handlerRegister = () => {
    return props.history.push("/register");
  };
  const history = useHistory();
  const [snack, setSnack] = useState({
    open: false,
    vertical: "top",
    horizpntal: "center",
  });
  const { vertical, horizontal, open } = snack;

  const handleClick = (newState) => () => {
    setSnack({ open: true, ...newState });
  };

  const handleClose = () => {
    setSnack({ ...snack, open: false });
  };

  const [alert, setAlert] = useState(false)

  const processLogin = async (values) => {
    try {
      const response = await axios.post(`${url}/login`, values);
      console.log(response);
      if (!response.data.logged) {
        setAlert(true)
        return history.push("/");
      }
      let user = response.data.data;
      let token = response.data.token;
      localStorage.setItem("userLogin", JSON.stringify(user));
      localStorage.setItem("token", token);
      return history.push("/dashboard");
    } catch(err) {
        console.log(err);
    }
  };

  const validation = yup.object({
    username: yup.string().required("Username is required"),
    password: yup.string().required("Password is required"),
  });
  const formik = useFormik({
    enableReinitialize: true,
    initialValues: {
      username: "",
      password: "",
    },
    validationSchema: validation,
    onSubmit: (values) => {
      processLogin(values);
      console.log(values);
    },
  });
  return (
    <form onSubmit={formik.handleSubmit}>
      <Stack
        justifyContent="center"
        alignItems="center"
        sx={{
          height: "100vh",
          width: "100%",
          backgroundColor: "#f7f7f7",
        }}
      >
        <Card
          sx={{
            width: "400px",
            height: "400px",
          }}
        >
          <Typography
            align="center"
            variant="h5"
            sx={{
              fontFamily: "poppins",
              fontWeight: "bold",
              margin: "40px 0px 16px 0px",
            }}
          >
            Login
          </Typography>
          <Typography
            sx={{
              fontFamily: "poppins",
              fontSize: 14,
              fontWeight: 500,
              margin: "0px 0px 10px 40px",
            }}
          >
            Username
          </Typography>
          <InputStyle
            name="username"
            onChange={formik.handleChange}
            values={formik.values.username}
            autoComplete="off"
          />
          <Typography
            sx={{
              fontFamily: "poppins",
              fontSize: 14,
              fontWeight: 500,
              margin: "0px 0px 10px 40px",
            }}
          >
            Password
          </Typography>
          <InputStyle
            type="password"
            name="password"
            onChange={formik.handleChange}
            values={formik.values.password}
            autoComplete="off"
          />
          <Button
            variant="contained"
            type="submit"
            sx={{
              backgroundColor: "#3E7C17",
              width: "317px",
              margin: "24px 0px 8px 40px",
              boxShadow: "none",
              fontFamily: "poppins",
              "&:hover": {
                backgroundColor: "#3E7C17",
              },
            }}
            onClick={handleClick({ vertical: "top", horizontal: "center" })}
          >
            Login
          </Button>
          {alert === true ? (<Snackbar
            anchorOrigin={{ vertical: 'top', horizontal: 'center' }}
            open={open}
            onClose={handleClose}
            message="Invalid Username or Password"
            key={vertical + horizontal}
```

```
            autoHideDuration={6000}
          />) : null}

          <div
            style={{
              display: "flex",
            }}
          >
            <Typography
              sx={{
                fontSize: "12px",
                fontFamily: "poppins",
                fontWeight: 400,
                margin: "0px 2px 8px 40px",
              }}
            >
              Don't Have Account?
            </Typography>
            <div onClick={() => handlerRegister()}>
              <Typography
                sx={{
                  fontSize: "12px",
                  fontFamily: "poppins",
                  fontWeight: 600,
                  color: "#3E7C17",
                  cursor: "pointer",
                }}
              >
                Register
              </Typography>
            </div>
          </div>
        </Card>
      </Stack>
    </form>
  );
};

export default Login;
```

Register.js

```
import React, { useState } from "react";
import {
  Card,
  Typography,
  Button,
  Snackbar,
  Alert,
  Stack,
} from "@mui/material";
import { styled } from "@mui/styles";
import { useFormik } from "formik";
import * as yup from "yup";
import axios from "axios";
import { useHistory } from "react-router";
import { url } from "../../config";

const InputStyle = styled("input")(({ theme }) => ({
  width: "300px",
  height: "40px",
  backgroundColor: "#f7f7f7",
  borderRadius: "5px",
  margin: "0px 0px 8px 40px",
  border: "none",
  paddingLeft: 16,
  fontFamily: "poppins",
  fontWeight: 400,
  "&:focus": {
    border: "none",
    outline: "none",
  },
}));

const Register = (props) => {
  const handlerLogin = () => {
    return props.history.push("/");
  };
  const [snack, setSnack] = useState({
    open: false,
    vertical: "top",
    horizpntal: "center",
  });
  const { vertical, horizontal, open } = snack;

  const handleClick = (newState) => () => {
    setSnack({ open: true, ...newState });
  };

  const handleClose = () => {
    setSnack({ ...snack, open: false });
  };

  const [alert, setAlert] = useState(false);
  const processRegister = async (values, resetForm) => {
    try {
      const response = await axios.post(`${url}/masyarakat/register`,
values);
      console.log("response");
      if (response) {
        setAlert(true);
        resetForm();
      }
    } catch (err) {
      console.log(err);
    }
  };
  const validation = yup.object({
    nama: yup.string().required("nama is required"),
    telp: yup.string().required("telp is required"),
    username: yup.string().required("Username is required"),
    password: yup.string().required("Password is required"),
  });
  const formik = useFormik({
    enableReinitialize: true,
    initialValues: {
      nama: "",
      telp: "",
      username: "",
      password: "",
    },
    validationSchema: validation,
    onSubmit: (values, { resetForm }) => {
      processRegister(values, resetForm);
      console.log(values);
    },
  });

  return (
    <form onSubmit={formik.handleSubmit}>
      <Stack
        justifyContent="center"
        alignItems="center"
        sx={{
          height: "100vh",
          width: "100%",
          backgroundColor: "#f7f7f7",
        }}
      >
        <Card
          sx={{
            width: "400px",
            //   height: "400px",
            paddingBottom: "24px",
          }}
        >
          <Typography
            align="center"
            variant="h5"
            sx={{
              fontFamily: "poppins",
              fontWeight: "bold",
              margin: "40px 0px 16px 0px",
            }}
          >
            Register
          </Typography>
          <Typography
            sx={{
              fontFamily: "poppins",
              fontSize: 14,
              fontWeight: 500,
              margin: "0px 0px 10px 40px",
            }}
          >
            Nama
          </Typography>
          <InputStyle
            name="nama"
            onChange={formik.handleChange}
            values={formik.values.nama}
            autoComplete="off"
          />
          <Typography
            sx={{
              fontFamily: "poppins",
              fontSize: 14,
              fontWeight: 500,
              margin: "0px 0px 10px 40px",
            }}
          >
            Telepon
          </Typography>
          <InputStyle
            name="telp"
            onChange={formik.handleChange}
            values={formik.values.telp}
            autoComplete="off"
          />
          <Typography
            sx={{
              fontFamily: "poppins",
              fontSize: 14,
              fontWeight: 500,
              margin: "0px 0px 10px 40px",
            }}
          >
            Username
          </Typography>
          <InputStyle
            name="username"
            onChange={formik.handleChange}
            values={formik.values.username}
            autoComplete="off"
          />
          <Typography
            sx={{
              fontFamily: "poppins",
              fontSize: 14,
              fontWeight: 500,
              margin: "0px 0px 10px 40px",
            }}
          >
            Password
          </Typography>
          <InputStyle
            type="password"
            name="password"
            onChange={formik.handleChange}
            values={formik.values.password}
            autoComplete="off"
          />
          <Button
            variant="contained"
            type="submit"
            sx={{
              backgroundColor: "#3E7C17",
              width: "317px",
              margin: "24px 0px 8px 40px",
              boxShadow: "none",
              fontFamily: "poppins",
              "&:hover": {
                backgroundColor: "#3E7C17",
              },
            }}
            onClick={handleClick({ vertical: "top", horizontal: "center" })}
          >
            Register
          </Button>
          {alert === true ? (
            <Snackbar
              open={open}
              autoHideDuration={6000}
              onClose={handleClose}
              anchorOrigin={{ vertical: "top", horizontal: "center" }}
            >
              <Alert
                onClose={handleClose}
                severity="success"
                sx={{ width: "100%" }}
              >
                Berhasil Mendaftar
              </Alert>
            </Snackbar>
          ) : (
            <Snackbar
              open={open}
              autoHideDuration={6000}
              onClose={handleClose}
              anchorOrigin={{ vertical: "top", horizontal: "center" }}
            >
              <Alert
```

```
                onClose={handleClose}
                severity="error"
                sx={{ width: "100%" }}
              >
                Gagal Mendaftar
              </Alert>
            </Snackbar>
          )}
          <div
            style={{
              display: "flex",
            }}
          >
            <Typography
              sx={{
                fontSize: "12px",
                fontFamily: "poppins",
                fontWeight: 400,
                margin: "0px 2px 8px 40px",
              }}
            >
              Already Have Account?
            </Typography>
            <div onClick={() => handlerLogin()}>
              <Typography
                sx={{
                  fontSize: "12px",
                  fontFamily: "poppins",
                  fontWeight: 600,
                  color: "#3E7C17",
                  cursor: "pointer",
                }}
              >
                Login
              </Typography>
            </div>
          </div>
        </Card>
      </Stack>
    </form>
  );
};

export default Register;
```

Router folder

Dashboard.js

```
import React  from 'react'
import Dashboard from '../templates/dashboard'
```

```
class dashboard extends React.Component {
    render(){
        return(
            <div>
                <Dashboard {...this.props} title="dashboard" />
            </div>
        )
    }
}

export default dashboard
```

Barang.js

```
import React  from 'react'
import Dashboard from '../templates/dashboard'

class barang extends React.Component {
    render(){
        return(
            <div>
                <Dashboard {...this.props} title="barang" />
            </div>
        )
    }
}

export default barang
```

Lelang.js

```
import React  from 'react'
import Dashboard from '../templates/dashboard'

class lelang extends React.Component {
    render(){
        return(
            <div>
                <Dashboard {...this.props} title="lelang" />
            </div>
        )
    }
}

export default lelang
```

Petugas.js

```
import React  from 'react'
import Dashboard from '../templates/dashboard'

class petugas extends React.Component {
    render(){
        return(
            <div>
                <Dashboard {...this.props} title="petugas" />
            </div>
        )
    }
}

export default petugas
```

Templates folder

Dashboard.js

```
import React from "react";
import Drawer from '../components/drawer'
import MainContentDashboard from '../components/mainContentDashboard'
import MainContentBarang from '../components/mainContentBarang'
import MainContentLelang from '../components/mainContentLelang'
import MainContentPetugas from '../components/mainContentPetugas'

class dashboard extends React.Component {
    render(){

        return(
            <div>
                <Drawer history={this.props.history} />
                {this.props.title === "dashboard" && <MainContentDashboard
/>}
                {this.props.title === "barang" && <MainContentBarang />}
                {this.props.title === "lelang" && <MainContentLelang />}
                {this.props.title === "petugas" && <MainContentPetugas />}
            </div>
        )
    }
}
export default dashboard
```

App.js

```
import React from "react";
import { Switch, Route } from 'react-router-dom'
import barang  from './router/barang'
import lelang  from './router/lelang'
import petugas  from './router/petugas'
import dashboard  from './router/dashboard'
import Login from './components/form/login'
import Register from './components/form/register'
```

```
function App() {
  return (
    <Switch>
      <Route exact path='/' component={Login}></Route>
      <Route exact path='/register' component={Register}></Route>
      <Route exact path='/dashboard' component={dashboard}></Route>
      <Route exact path='/barang' component={barang}></Route>
      <Route exact path='/lelang' component={lelang}></Route>
      <Route exact path='/petugas' component={petugas}></Route>
    </Switch>
  );
}

export default App;
```

Index.js diubah

```
import {BrowserRouter} from 'react-router-dom'

ReactDOM.render(
  <BrowserRouter>
  <App />
  </BrowserRouter>,
  document.getElementById('root')
);
```