

LAMPIRAN CODINGAN

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

import { Link } from "react-router-dom";

import { useFormik } from "formik";

import * as Yup from "yup";

import { connect } from "react-redux";

import { FormattedMessage, injectIntl } from "react-intl";

import * as auth from "../_redux/authRedux";

import { login } from "../_redux/authCrud";

import Swal from "sweetalert2";

/*

INTL (i18n) docs:

https://github.com/formatjs/react-

intl/blob/master/docs/Components.md#formattedmessage

*/

/*

Formik+YUP:

https://jaredpalmer.com/formik/docs/tutorial#getfieldprops

*/
```

```
const initialValues = {  
  email: "",  
  password: "",  
};  
  
if (process.env.NODE_ENV === "development") {  
  initialValues.email = "admin@mail.com";  
  initialValues.password = "password";  
}  
  
function Login(props) {  
  const { intl } = props;  
  const [loading, setLoading] = useState(false);  
  const LoginSchema = Yup.object().shape({  
    email: Yup.string()  
      .email("Wrong email format")  
      .min(3, "Minimum 3 symbols")  
      .max(50, "Maximum 50 symbols")  
      .required(  
        intl.formatMessage({  
          id: "AUTH.VALIDATION.REQUIRED_FIELD",  
        })  
      ),  
    password: Yup.string()
```

```

        .min(3, "Minimum 3 symbols")

        .max(50, "Maximum 50 symbols")

        .required(

            intl.formatMessage({

                id: "AUTH.VALIDATION.REQUIRED_FIELD",

            })

        ),

    });

const enableLoading = () => {
    setLoading(true);
};

const disableLoading = () => {
    setLoading(false);
};

const getInputClasses = (fieldname) => {
    if (formik.touched[fieldname] && formik.errors[fieldname]) {
        return "is-invalid";
    }

    if (formik.touched[fieldname] && !formik.errors[fieldname]) {
        return "is-valid";
    }
}

```

```
}

return "";

};

const formik = useFormik({

  initialValues,

  validationSchema: LoginSchema,

  onSubmit: (values, { setStatus, setSubmitting }) => {

    enableLoading();

    setTimeout(() => {

      login(values.email, values.password)

        .then((response) => {

          console.log(response, "response");

          if (response.status === 200) {

            console.log("masuk sukses");

            // disableLoading();

            const token = response.data.data.token;

            console.log(token, "masuk token");

            props.login(token);

            setSubmitting(true);

          } else {
```

```
console.log("masuk else");

Swal.fire({

  title: "Error!",

  text: `${response.data.error}`,

  icon: "error",

  heightAuto: false,

  confirmButtonText: "Ok",

}).then((result) => {

  disableLoading();

  setSubmitting(false);

});

}

})

.catch((error) => {

  console.log(error, "error");

  setStatus(

    intl.formatMessage({

      id: "AUTH.VALIDATION.INVALID_LOGIN",

    })

  );

})

  .finally(() => {});

}, 1000);

},
```

```
});
```

```
return (
```

```
<div className="login-form login-signin" id="kt_login_signin_form">
```

```
  { /* begin::Head */ }
```

```
<div className="text-center mb-10 mb-lg-20">
```

```
<h3 className="font-size-h1">Login</h3>
```

```
<p className="text-muted font-weight-bold">
```

```
  Enter your Email and Password
```

```
</p>
```

```
</div>
```

```
  { /* end::Head */ }
```

```
  { /* begin::Form */ }
```

```
<form
```

```
  onSubmit={formik.handleSubmit}
```

```
  className="form fv-plugins-bootstrap fv-plugins-framework"
```

```
>
```

```
<div className="form-group fv-plugins-icon-container">
```

```
<input
```

```
  placeholder="Email"
```

```
  type="email"
```

```
  className={`form-control form-control-solid h-auto py-5 px-6`}
```

```
  ${getInputClasses(
```

```

    "email"

  }}` }

  name="email"

  {...formik.getFieldProps("email")}

/>

{formik.touched.email && formik.errors.email ? (

  <div className="fv-plugins-message-container">

    <div className="fv-help-block">{formik.errors.email}</div>

  </div>

) : null}

</div>

<div className="form-group fv-plugins-icon-container">

  <input

    placeholder="Password"

    type="password"

    className={` form-control form-control-solid h-auto py-5 px-6

    ${getInputClasses(

      "password"

    )}}` }

    name="password"

    {...formik.getFieldProps("password")}

  />

  {formik.touched.password && formik.errors.password ? (

    <div className="fv-plugins-message-container">

```

```

    <div className="fv-help-block">{formik.errors.password}</div>

  </div>

  ): null}

</div>

<div className="form-group d-flex flex-wrap justify-content-center align-
items-center">

  <button

    id="kt_login_signin_submit"

    type="submit"

    disabled={formik.isSubmitting}

    className={`btn btn-primary font-weight-bold px-9 py-4 my-3`}

  >

    <span>Sign In</span>

    {loading && <span className="ml-3 spinner spinner-white"></span>}

  </button>

</div>

</form>

{/*end::Form*/}

</div>

);

}

```

```

export default injectIntl(connect(null, auth.actions)(Login));

```