

## DAFTAR PUSTAKA

- Abdul Kadir.(2017).*Pemrograman Arduino Dan Processing*. Jakarta : Penerbit Elex Media Komputindo
- Abdul Kadir.(2018). *Arduino Dan Sensor*. Yogyakarta : Penerbit Andi Publisier
- Handayani Saptaji W.(2015).*Mudah Belajar Mikrokontroller Dengan Arduino* : Penerbit Widya Media
- Ir. Sutarno, M,Sc.(2014).*Instrumentasi Industri Dan Kontrol Proses*. Yogyakarta : Penerbit Graha Ilmu
- Irfan Indra Kurniawan.(2017).*fansElectronics.com*.
- Ivan Brokhotkov.(2015).*ESP 8266 Arduino Core Documentation Release 2.4.0*
- Ajie.(2016).*Saptaji.com*
- M. Fajar Wicaksana, Hidayat.(2017).*Mudah Belajar Mikrokontroler Arduino* : Penerbit Informatika
- Tim Fakultas Teknologi Industri (2019). *Pedoman Tugas Akhir*. Universitas PGRI Adi Buana Surabaya.
- Ratih Listiyani.(2018). *Dasar Listrik Dan Elektronika*. Yogyakarta : Penerbit CV Budi Utama
- Andrianto, Heri dan Aan Darmawan. 2016. *Arduino Belajar Cepat dan Pemograman. Informatika* : Bandung.
- Artanto, Dian. 2009. *Merakit PLC dengan Mikrokontroler*. PT Alex Media Komputindo : Jakarta.
- Budiarto, Widodo. 2011. *Aneka Proyek Mikrokontroler*. Graha Ilmu: Yogyakarta.
- Hidayat, Dewi, D,S. 2008. *Bisnis Reparasi Ponsel Penebar Plus* : Jakarta.
- Hutahaean,J. 2014. *Konsep Sistem Informasi* . Deepublish :Yogyakarta.
- Irwansyah, Edy dan Moniaga J,V. 2014. *Pengantar Teknologi Informasi*. Deepublish: Yogyakarta



# UNIVERSITAS PGRI ADI BUANA SURABAYA

## FAKULTAS TEKNIK


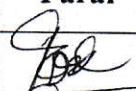







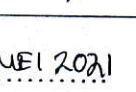
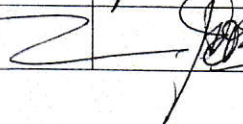
Program Studi : Teknik Lingkungan – Perencanaan Wilayah Kota  
Teknik Industri – Teknik Elektro - PVKK

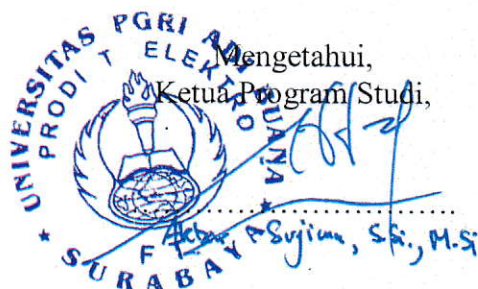
KAMPUS II: Jl. Dukuh Menanggal XII/4 ☎ (031) 8281181 Surabaya 60234

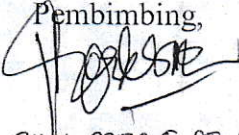
Website : [www.ft.unipasby.ac.id](http://www.ft.unipasby.ac.id) E-mail : [ft@unipasby.ac.id](mailto:ft@unipasby.ac.id)


### BERITA ACARA BIMBINGAN SKRIPSI

Form Skripsi-03

Nama	: FIT RAMDHANU			
NIM	: 193609005			
Program Studi	: TEKNIK ELEKTRO 2019 B			
Pembimbing	: DRS. BUDI PRIJO SENBODO, ST, M.KOM			
Periode Bimbingan	: <del>Gesel</del> /Genap*) Tahun 2020 / 2021			
Judul Skripsi	RANCANGAN SISTEM DETEUSI IDENTITAS GROUND SUPPORT EQUIPMENT (GSE) UNTUK ASES MASUK AIRSIDE DI BANDAR UDARA INTERNASIONAL JUANDA SURABAYA			
<b>KEGIATAN KONSULTASI / BIMBINGAN</b>				
No	Tanggal	Materi pembimbingan	Keterangan	Paraf
1	05 MARET 2021	PENJELASAN SOP PENULISAN	ACC	
2	12 MARET 2021	TUJUAN PENELITIAN	REVISI	
3	26 MARET 2021	TUJUAN PENELITIAN	ACC	
4	29 MARET 2021	MEMBUAT ANALISIS DATA	ACC	
5	04 APRIL 2021	METODE PENGUMPULAN DATA	ACC	
6	10 APRIL 2021	PENYAJIAN DATA	REVISI	
7	24 APRIL 2021	PENYAJIAN DATA	ACC	
8	08 MEI 2021	DEBATAHAN	REVISI	
9	18 MEI 2021	DEBATAHAN	ACC	
Dinyatakan selesai tanggal : ...20 MEI..... 2021.				



Pembimbing,  
  
DRS. BUDI PRIJO, S. ST, M.KOM

Surabaya, 20 MEI 2021  
Mahasiswa,  
  
FIT RAMDHANU





# UNIVERSITAS PGRI ADI BUANA SURABAYA

## FAKULTAS TEKNIK

Program Studi : Teknik Lingkungan – Perencanaan Wilayah Kota  
Teknik Industri – Teknik Elektro - PVKK

KAMPUS II: Jl. Dukuh Menanggal XII/4 ☎ (031) 8281181 Surabaya

Website : [www.ft.unipasby.ac.id](http://www.ft.unipasby.ac.id) E-mail : [ft@unipasby.ac.id](mailto:ft@unipasby.ac.id)

### FORM REVISI SKRIPSI

Nama Mahasiswa : FIT RAMDHANU  
NIM : 193609005  
Fakultas / Progdil : TEKNIK ELEKTRO  
Judul Skripsi : RANCANGAN SISTEM DETEKSI IDENTITAS GROUND SUPPORT  
EQUIPMENT (GSE) UNTUK AKSES MASUK AIRSIDE DI BANDAR  
UDARA INTERNASIONAL JUANDA SURABAYA  
Ujian Tanggal : 2 JUNI 2021

No Bab.	Tanggal	Materi Konsultasi	Keterangan Catatan	Tanda Tangan Penguji
I	9 JUNI 2021	PENULISAN SUMBER BAB II	Ace	
II	9 JUNI 2021	PENULISAN DAFTAR PUSTAKA	Ace	
III	11 JUNI 2021	PENAMAAN GAMBAR / TABEL	Ace	
IV	11 JUNI 2021	PENYAJIAN DATA	Ace	
V				

Disetujui Dosen Penguji  
Pada Tanggal, 11 JUNI 2021.

Penguji I,

Penguji II,

- Penyelesaian Revisi paling lambat 2 minggu dari pelaksanaan Ujian Skripsi.
  - Pengetikan, penjilidan, penandatanganan Skripsi dan mengumpulkan Skripsi paling lambat 2 minggu dari revisi.
- Apabila sampai batas waktu tersebut ( point 1,a dan b ) mahasiswa belum menyelesaikan revisi dan tanda tangan, maka **Ujian dinyatakan Gugur**.
- Foto copy Form Revisi diserahkan ke Program Studi.
  - Skripsi yang sudah direvisi diserahkan ke Fakultas tiga eksemplar untuk dijilid.

## LAMPIRAN 1

Listing program Rancangan Sistem Keamanan *Ground Support Equipment (GSE)* menggunakan *Radio Frequency Identification (RFID)* :

```
import React, { useState, useEffect, useMemo, useCallback } from "react";
import { withApollo } from "../libs/apollo";
import AdminArea, { protectAdminArea } from "../components/AdminArea";
import Header from "../components/Header";
import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";
import Table from "../components/Table";
import { useRouter } from "next/router";
import { useNotification } from "../components/Notification";
import { gql, useQuery, useMutation, useApolloClient } from "@apollo/client";
import { FormModal } from "../components/Modal";
import { showLoadingSpinner, hideLoadingSpinner } from "../components/App";
import FadeImage from "../components/FadeImage";
import appConfig from "../app.json";
import { ShortText } from "../components/form/ShortText";
import { Email } from "../components/form/Email";
import { PhoneNumber } from "../components/form/PhoneNumber";
import { LongText } from "../components/form/LongText";
import { SingleSelect } from "../components/form/SingleSelect";

const ALL_USERS = gql`
  query A {
    allUsers {
      _id
      username
      email
      phone
      roleId
      Role {
        _id
        name
      }
    }
    status
  }
`
```

```

      name
      address
      pictureUrl
    }
  }
`;
```

```

const ACTIVATE = gql`
  mutation B($_id: ID!) {
    activateUser(_id: $_id)
  }
`;
```

```

const DEACTIVATE = gql`
  mutation C($_id: ID!) {
    deactivateUser(_id: $_id)
  }
`;
```

```

const Page = (props) => {
  const notification = useNotification();
  const router = useRouter();
  const { loading, error, data, refetch } = useQuery(ALL_USERS, {});
```

```

  let allUsers = [];
  if (data && data.allUsers) {
    allUsers = data.allUsers;
  }

```

```

const [formModalVisible, setFormModalVisible] = useState(false);
const [resetPasswordModalVisible, setResetPasswordModalVisible] =
  useState(false);
const [userData, setUserData] = useState({});
```

```

const handleInput = (key) => (e) => {
  setUserData({
    ...userData,
```

```

      [key]: e.target.value,
    });
  });

const [activateUser] = useMutation(ACTIVATE);
const [deactivateUser] = useMutation(DEACTIVATE);
const apolloClient = useApolloClient();

// console.log(allUsers);

const columns = useMemo(() => {
  let columnDefinitions = [
    {
      Header: "Username",
      accessor: "username",
    },
    {
      Header: "Nama",
      accessor: "name",
      Cell: (props) => (
        <div
          className={
            "w-full " +
            (props.row.original._deletedAt ? "text-gray-200" : "text-black")
          }
        >
          {props.cell.value || props.row.original.username}
        </div>
      ),
    },
    {
      Header: "Email",
      accessor: "email",
    },
    {
      Header: "Kontak",
      accessor: "phone",
    },
  ];

```

```

    },
    {
      Header: "Role",
      accessor: "_id",
      Cell: (props) => <div>{props.row.original.Role.name}</div>,
    },
    {
      Header: "Status",
      accessor: "status",
      Cell: (props) => (
        <div>
          <div
            className={
              "pb-2 font-bold " +
              (props.cell.value === "Aktif"
                ? "text-green-500"
                : "text-red-500")
            }
          >
            {props.cell.value}
          </div>
          {props.cell.value === "Aktif" ? (
            <button
              className="btn btn-sm text-xs btn-danger"
              onClick={async (e) => {
                if (e) e.preventDefault();
                showLoadingSpinner();
                try {
                  await deactivateUser({
                    variables: {
                      _id: props.row.original._id,
                    },
                  });
                  await refetch();

                  notification.addNotification({
                    title: "Sukses!",

```

```

        message: `User ${props.row.original.username} saat ini telah
berstatus Non Aktif`,
        level: "success",
    });
    } catch (err) {
        notification.handleError(err);
    }
    hideLoadingSpinner();
  }}
>
  <FontAwesomeIcon icon="ban" /> Set Tidak Aktif
</button>
): (
  <button
    className="btn btn-sm text-xs btn-success"
    onClick={async (e) => {
      if (e) e.preventDefault();
      showLoadingSpinner();
      try {
        await activateUser({
          variables: {
            _id: props.row.original._id,
          },
        });
        await refetch();

        notification.addNotification({
          title: "Sukses!",
          message: `User ${props.row.original.username} saat ini telah
berstatus Aktif`,
          level: "success",
        });
      } catch (err) {
        notification.handleError(err);
      }
      hideLoadingSpinner();
    }}
  >

```



```

        <FontAwesomeIcon icon="check-circle" /> Set Aktif
      </button>
    )}
  </div>
),
},
{
  Header: "Utilitas",
  accessor: "address",
  Cell: (props) => (
    <div>
      <div className="py-1">
        <button
          type="button"
          className="btn btn-primary btn-sm"
          onClick={(e) => {
            setUserData({
              ...props.row.original,
            });
            setFormModalVisible(true);
          }}
        >
          <FontAwesomeIcon icon="pencil-alt" /> Update User
        </button>
      </div>
      <div className="py-1">
        <button
          type="button"
          className="btn btn-purple btn-sm"
          onClick={(e) => {
            setUserData({
              ...props.row.original,
              newPassword: "",
              passwordConfirmation: "",
            });
            setResetPasswordModalVisible(true);
          }}
        >

```

```

        >
        <FontAwesomeIcon icon="key" /> Reset Password
    </button>
</div>
</div>
),
},
];

return columnDefinitions;
});

return (
    <AdminArea>
        <Header title="Daftar Users" />

        <FormModal
            title={
                <span>
                    <FontAwesomeIcon icon="user" /> User
                </span>
            }
            visible={formModalVisible}
            onClose={(e) => {
                if (e) e.preventDefault();
                setFormModalVisible(false);
            }}
            onSubmit={async (e) => {
                if (e) e.preventDefault();
                showLoadingSpinner();
                try {
                    if (userData._id) {
                        await apolloClient.mutate({
                            mutation: gql`
                                mutation updateUser(
                                    $_id: ID!
                                    $username: String

```

```

    $email: String
    $phone: String
    $name: String
    $address: String
    $roleId: String
  ) {
    updateUser(
      _id: $_id
      name: $name
      username: $username
      email: $email
      phone: $phone
      address: $address
      roleId: $roleId
    )
  }
`,
  variables: {
    ...userData,
  },
});
await refetch();
notification.addNotification({
  title: "Sukses",
  message: `User ${userData.username} berhasil diperbarui!`,
  level: "success",
});
} else {
  const { password, passwordConfirmation } = userData;
  if (password !== passwordConfirmation) {
    throw {
      message: "Konfirmasi password tidak cocok!",
    };
  }
}

await apolloClient.mutate({
  mutation: gql`

```

```

mutation registerUser(
  $username: String!
  $password: String!
  $roleId: String!
  $name: String!
  $status: String!
  $email: String
  $phone: String
  $address: String
) {
  registerUser(
    username: $username
    password: $password
    roleId: $roleId
    name: $name
    status: $status
    email: $email
    phone: $phone
    address: $address
  ) {
    _id
  }
}
`,
variables: {
  ...userData,
  status: "Aktif",
},
});
await refetch();
notification.addNotification({
  title: "Sukses",
  message: `User ${userData.username} berhasil ditambahkan!`,
  level: "success",
});
}
setFormModalVisible(false);

```

```

    } catch (err) {
      console.log({ err });
      notification.handleError(err);
    }
    hideLoadingSpinner();
  }}
  size="lg"
>
<div className="grid grid-cols-1 sm:grid-cols-2 gap-6">
  <div>
    <ShortText
      label="Username"
      disabled={!userData._id}
      required
      value={userData.username}
      onChange={(e) => {
        if (e) e.preventDefault();
        if (!userData._id) return;
        setUserData({
          ...userData,
          username: e.target.value,
        });
      }}
    />
    <SingleSelect
      label="Role"
      value={userData.roleId}
      options={[
        {
          label: "Admin",
          value: "__ADMIN__",
        },
        {
          label: "User",
          value: "__USER__",
        },
      ]}
    />
  </div>
</div>

```



```

renderValue={(value) => {
  const selectedItem = [
    {
      label: "Admin",
      value: "__ADMIN__",
    },
    {
      label: "User",
      value: "__USER__",
    },
  ].find((item) => item.value === value);
  return selectedItem.label;
}}
onChange={(e) => {
  if (e) e.preventDefault();
  setUserData({
    ...userData,
    roleId: e.target.value,
  });
}}
/>

```

```

{!userData._id && (
  <div>
    <div className="form-group pb-6">
      <label>Password *</label>
      <input
        id="new-password"
        autoComplete="new-password"
        type="password"
        className="form-control"
        value={userData.password}
        onChange={(e) => {
          if (e) e.preventDefault();
          setUserData({
            ...userData,
            password: e.target.value,

```

```

        });
    }}
/>
</div>
<div className="form-group pb-6">
  <label>Konfirmasi Password *</label>
  <input
    id="confirm-password"
    autoComplete="confirm-password"
    type="password"
    className="form-control"
    value={userData.passwordConfirmation}
    onChange={(e) => {
      if (e) e.preventDefault();
      setUserData({
        ...userData,
        passwordConfirmation: e.target.value,
      });
    }}
  />
</div>
</div>
)}
</div>
<div>
  <ShortText
    label="Nama"
    required
    value={userData.name}
    onChange={(e) => {
      if (e) e.preventDefault();
      setUserData({
        ...userData,
        name: e.target.value,
      });
    }}
  />

```

```

<Email
  label="Email"
  required
  value={userData.email}
  onChange={(e) => {
    if (e) e.preventDefault();
    setUserData({
      ...userData,
      email: e.target.value,
    });
  }}
/>
<PhoneNumber
  label="Kontak / HP"
  required
  value={userData.phone}
  onChange={(e) => {
    if (e) e.preventDefault();
    setUserData({
      ...userData,
      phone: e.target.value,
    });
  }}
/>
<LongText
  label="Alamat"
  value={userData.address}
  onChange={(e) => {
    if (e) e.preventDefault();
    setUserData({
      ...userData,
      address: e.target.value,
    });
  }}
/>
</div>
</div>

```