

✓ TUGAS10



> build

✓ db

{ } dosen.json

{ } kelas.json

{ } mahasiswa.json

{ } matakuliah.json

{ } user.json

> node_modules

> public

✓ src

✓ Components

JS Button.js

JS Card.js

JS Footer.js

JS Form.js

JS FormDosen.js

JS FormKelas.js

JS FormMahasiswa.js

JS FormMataKuliah.js

JS FormRencanaStudi.js

JS Header.js

JS Input.js

JS Label.js

JS Modal.js

JS ProfileMenu.js

JS Select.js

JS Sidebar.js

✓ Context

JS AuthContext.js

✓ Layouts

JS AdminLayout.js

JS AuthLayout.js

✓ TUGAS10



✓ src

✓ Layouts

✓ `AuthLayout.js`

✓ `JS AuthLayout.js`

✓ Pages

✓ Admin

✓ `JS Dashboard.js`

✓ Auth

✓ `JS Login.js`

✓ `JS Register.js`

✓ Dosen

✓ `JS Dosen.js`

✓ `JS DosenDetail.js`

✓ `JS DosenModal.js`

✓ `JS DosenTable.js`

✓ Mahasiswa

✓ `JS Mahasiswa.js`

✓ `JS MahasiswaModal.js`

✓ `JS MahasiswaTable.js`

✓ MataKuliah

✓ `JS MataKuliah.js`

✓ `JS MataKuliahDetail.js`

✓ `JS MataKuliahModal.js`

✓ `JS MataKuliahTable.js`

✓ RencanaStudi

✓ `JS RencanaStudi.js`

✓ `JS RencanaStudiModal.js`

✓ `JS RencanaStudiTable.js`

✓ `JS MahasiswaDetail.js`

✓ Routes

✓ `JS ProtectedRoute.js`

✓ Utils/Helpers

✓	Utils/Helpers	
✓	Apis	
JS	AuthApi.js	U
JS	DosenApi.js	U
JS	KelasApi.js	U
JS	MahasiswaApi.js	U
JS	MataKuliahApi.js	U
JS	AxiosInstance.js	U
JS	SwalHelper.js	U
JS	ToastHelper.js	U
JS	App.js	M
#	index.css	M
JS	index.js	M
❖	.gitignore	
{}	db.json	U
JS	merge-json.cjs	U
{}	package-lock.json	M
{}	package.json	M
ⓘ	README.md	
JS	tailwind.config.js	U
JS	updaterole.cjs	U


```
[  
  
{  
    "id": "1",  
    "nip": "DSN01",  
    "nama": "Slamet",  
    "email":  
    "slamet@gmail.com",  
},  
{  
    "id": "2",  
    "nip": "DSN02",  
    "nama": "Tini",  
    "email": "tini@gmail.com",  
    "maxSKS": 10 },  
{  
    "id": "3",  
    "nip": "DSN03",  
    "nama": "Udin",  
    "email": "udin@gmail.com",  
    "maxSKS": 14 },  
{  
    "id": "4",  
    "nip": "DSN04",  
    "nama": "Vina",  
    "email": "vina@gmail.com",  
    "maxSKS": 16 },  
{  
    "id": "5",  
    "nip": "DSN05",  
    "nama": "Wawan",  
    "email": "wawan@gmail.com",  
    "maxSKS": 8 }]  
]
```



```
[  
 {  
   "id": "1",  
   "mata_kuliah_id": "1",  
   "dosen_id": "2",  
   "mahasiswa_ids": ["3", "5",  
 "6"]  
 }]
```

```
[  
  {  
    "id": "1",  
    "nim": "M001",  
    "nama": "Andi",  
    "email": "andi@gmail.com",  
    "jurusan": "Informatika",  
    "maxSKS": 18  
  },  
  {  
    "id": "2",  
    "nim": "M002",  
    "nama": "Budi",  
    "email": "budi@gmail.com",  
    "jurusan": "Sistem  
Info"maxSKS": 20  
  },  
  {  
    "id": "3",  
    "nim": "M003",  
    "nama": "Citra",  
    "email": "citra@gmail.com",  
    "jurusan": "Informatika",  
    "maxSKS": 22  
  },  
  {  
    "id": "4",  
    "nim": "M004",  
    "nama": "Dina",  
    "email": "dina@gmail.com",  
    "jurusan": "Sistem  
Info"maxSKS": 18  
  },  
  {  
    "id": "5",  
    "nim": "M005",  
    "nama": "Eko",  
    "email": "eko@gmail.com",  
    "jurusan": "Informatika",  
    "maxSKS": 24  
  },  
  {  
    "id": "6",  
    "nim": "M006",  
    "nama": "Fajar",  
    "email": "fajar@gmail.com",  
    "jurusan": "Teknik Komputer",  
    "maxSKS": 21  
  },  
  {  
    "id": "7",  
    "nim": "M007",  
    "nama": "Gita",  
    "email": "gita@gmail.com",  
    "jurusan": "Sistem  
Info"maxSKS": 20  
  },  
  {  
    "id": "8",  
    "nim": "M008",  
    "nama": "Hani",  
    "email": "hani@gmail.com",  
    "jurusan": "Informatika",  
    "maxSKS": 18  
  },  
  {  
    "id": "9",  
    "nim": "M009",  
    "nama": "Iwan",  
    "email": "iwan@gmail.com",  
    "jurusan": "Teknik Komputer",  
    "maxSKS": 20  
  },  
  {  
    "id": "10",  
    "nim": "M010",  
    "nama": "Joko",  
    "email": "joko@gmail.com",  
    "jurusan": "Informatika",  
    "maxSKS": 18  
  },  
  {  
    "id": "11",  
    "nim": "M011",  
    "nama": "Kiki",  
    "email": "kiki@gmail.com",  
    "jurusan": "Sistem  
Info"maxSKS": 24  
  },  
  {  
    "id": "12",  
    "nim": "M012",  
    "nama": "Lia",  
    "email": "lia@gmail.com",  
    "jurusan": "Teknik Komputer",  
    "maxSKS": 20  
  },  
  {  
    "id": "13",  
    "nim": "M013",  
    "nama": "Maya",  
    "email": "maya@gmail.com",  
    "jurusan": "Informatika",  
    "maxSKS": 18  
  },  
  {  
    "id": "14",  
    "nim": "M014",  
    "nama": "Nina",  
    "email": "nina@gmail.com",  
    "jurusan": "Sistem  
Info"maxSKS": 22  
  },  
  {  
    "id": "15",  
    "nim": "M015",  
    "nama": "Oki",  
    "email": "oki@gmail.com",  
    "jurusan": "Teknik Komputer",  
    "maxSKS": 20  
  },  
  {  
    "id": "16",  
    "nim": "M016",  
    "nama": "Putri",  
    "email": "putri@gmail.com",  
    "jurusan": "Informatika",  
    "maxSKS": 18  
  },  
  {  
    "id": "17",  
    "nim": "M017",  
    "nama": "Qila",  
    "email": "qila@gmail.com",  
    "jurusan": "Sistem  
Info"maxSKS": 18  
  },  
  {  
    "id": "18",  
    "nim": "M018",  
    "nama": "Rudi",  
    "email": "rudi@gmail.com",  
    "jurusan": "Informatika",  
    "maxSKS": 21  
  },  
  {  
    "id": "19",  
    "nim": "M019",  
    "nama": "Salsa",  
    "email": "salsa@gmail.com",  
    "jurusan": "Sistem  
Info"maxSKS": 18  
  },  
  {  
    "id": "20",  
    "nim": "M020",  
    "nama": "Tino",  
    "email": "tino@gmail.com",  
    "jurusan": "Teknik Komputer",  
    "maxSKS": 18  
  }  
]
```

```
[  
  {  
    "id": "1",  
    "kode": "MK01",  
    "nama": "Algoritma dan  
Pemrograman3",  
  },  
  {  
    "id": "2",  
    "kode": "MK02",  
    "nama": "Basis Data",  
    "sks": 3  
  },  
  {  
    "id": "3",  
    "kode": "MK03",  
    "nama": "Jaringan Komputer",  
    "sks": 2  
  },  
  {  
    "id": "4",  
    "kode": "MK04",  
    "nama": "Pemrograman Sisi Klien",  
    "sks": 3  
  },  
  {  
    "id": "5",  
    "kode": "MK05",  
    "nama": "Pemrograman Sisi Server",  
    "sks": 3  
  },  
  {  
    "id": "6",  
    "kode": "MK06",  
    "nama": "Kalkulus",  
    "sks": 3  
  },  
  {  
    "id": "7",  
    "kode": "MK07",  
    "nama": "Rekayasa Perangkat Lunak",  
    "sks": 3  
  },  
  {  
    "id": "8",  
    "kode": "MK08",  
    "nama": "Kecerdasan Buatan",  
    "sks": 3  
  }  
]
```

```
[  
  {  
    "id": 1,  
    "nama": "Admin 1",  
    "email": "admin@gmail.com",  
    "password": "admin123",  
    "role": "admin",  
    "permissions": [  
      "rencanastudi.page",  
      "rencanastudi.read",  
      "rencanastudi.create",  
      "rencanastudi.update",  
      "rencanastudi.delete",  
      "dashboard.page",  
      "mahasiswa.page",  
      "mahasiswa.read",  
      "mahasiswa.create",  
      "mahasiswa.update",  
      "mahasiswa.delete",  
      "dosen.page",  
      "dosen.read",  
      "dosen.create",  
      "dosen.update",  
      "dosen.delete",  
      "matakuliah.page",  
      "matakuliah.read",  
      "matakuliah.create",  
      "matakuliah.update",  
      "matakuliah.delete",  
      "kelas.page",  
      "kelas.read",  
      "kelas.create",  
      "kelas.update",  
      "kelas.delete"  
    ]  
  },  
  {  
    "id": 2,  
    "nama": "Mahasiswa 1",  
    "email":  
    "mahasiswa@gmail.com",  
    "password": "mahasiswa123",  
    "role": "mahasiswa",  
    "permissions": [  
  
      "rencanastudi.page",  
      "rencanastudi.read",  
      "rencanastudi.create",  
      "rencanastudi.update",  
  
      "rencanastudi.delete",  
      "kelas.create",  
      "kelas.read",  
      "kelas.update",  
      "kelas.delete"  
    ]  
  }  
]
```

```
● ● ●
```

```
import React, { useEffect, useState } from "react";
import Label from "../Components/Label";
import Input from "../Components/Input";
import Button from "../Components/Button";
import PropTypes from "prop-types";
import AxiosInstance from "../Utils/Helpers/AxiosInstance";

const FormKelas = ({ onSubmit, editData, onClose }) => {
  const [formData, setFormData] = useState({
    nama: "",
    mataKuliahId: "",
    dosenId: "",
    mahasiswaIds: [],
  });

  const [mataKuliahs, setMataKuliahs] = useState([]);
  const [dosenList, setDosenList] = useState([]);
  const [mahasiswaList, setMahasiswaList] = useState([]);
  const [sksMatkul, setSksMatkul] = useState(0);

  // Fetch data awal
  useEffect(() => {
    const fetchData = async () => {
      try {
        const [mkRes, dosenRes, mhsRes] = await Promise.all([
          AxiosInstance.get("/matakuliah"),
          AxiosInstance.get("/dosen"),
          AxiosInstance.get("/mahasiswa"),
        ]);

        // Normalisasi: konversi id mahasiswa ke string
        const mhsFormatted = mhsRes.data.map((m) => ({
          ...m,
          id: String(m.id),
        }));

        setMataKuliahs(mkRes.data);
        setDosenList(dosenRes.data);
        setMahasiswaList(mhsFormatted);
      } catch (err) {
        console.error("Gagal mengambil data:", err);
      }
    };
    fetchData();
  }, []);

  // Set formData ketika editData tersedia
  useEffect(() => {
    if (editData) {
      setFormData({
        nama: editData.nama || "",
        mataKuliahId: editData.mataKuliahId?.toString() ||
          editData.mataKuliah?.id?.toString() ||
          "",
        dosenId: editData.dosenId?.toString() ||
          editData.dosen?.id?.toString() ||
          "",
        mahasiswaIds: (editData.mahasiswaIds || editData.mahasiswa || []).map(
          (m) => (typeof m === "object" ? String(m.id) : String(m))
        ),
      });
    }
  }, [editData]);

  // Update SKS berdasarkan mata kuliah
  useEffect(() => {
    const matkul = mataKuliahs.find(
      (mk) => mk.id === Number(formData.mataKuliahId)
    );
    setSksMatkul(matkul ? matkul.sks : 0);
  }, [formData.mataKuliahId, mataKuliahs]);

  const isMahasiswaEligible = (m) =>
    m && m.totalSKS + sksMatkul <= m.maxSKS;

  const isDosenEligible = (d) =>
    d && d.totalSKS + sksMatkul <= d.maxSKS;

  const handleChange = (e) => {
    const { id, value } = e.target;
    setFormData((prev) => ({
      ...prev,
      [id]: value,
    }));
  };

  const handleCheckboxChange = (idStr) => {
    setFormData((prev) => {
      const exists = prev.mahasiswaIds.includes(idStr);
      return {
        ...prev,
        mahasiswaIds: exists
          ? prev.mahasiswaIds.filter((mid) => mid !== idStr)
          : [...prev.mahasiswaIds, idStr],
      };
    });
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    const { nama, mataKuliahId, dosenId, mahasiswaIds } = formData;

    if (!nama || !mataKuliahId || !dosenId || mahasiswaIds.length === 0) {
      alert("Harap lengkapi semua field dan pilih minimal satu mahasiswa.");
      return;
    }

    const dosen = dosenList.find((d) => String(d.id) === dosenId);
    const matkul = mataKuliahs.find((mk) => String(mk.id) === mataKuliahId);

    if (!dosen || !matkul) {
      alert("Dosen atau mata kuliah tidak ditemukan.");
      return;
    }

    if (!isDosenEligible(dosen)) {
      alert("Dosen melebihi batas SKS.");
      return;
    }

    const mahasiswaInvalid = mahasiswaIds
      .map((id) => mahasiswaList.find((m) => m.id === id))
      .filter((m) => !m || !isMahasiswaEligible(m));

    if (mahasiswaInvalid.length > 0) {
      alert(`Mahasiswa melebihi SKS: ${mahasiswaInvalid
        .map((m) => (m ? m.nama : "(Mahasiswa tidak ditemukan)"))
        .join(", ")}`);
      return;
    }

    const payload = {
      nama,
      matakuliah: { id: Number(matkul.id), nama: matkul.nama },
      dosen: { id: Number(dosen.id), nama: dosen.nama },
      mahasiswaIds: mahasiswaIds.map((id) => Number(id)),
      mahasiswa: mahasiswaIds.map((idStr) => {
        const m = mahasiswaList.find((m) => m.id === idStr);
        return { id: Number(m.id), nama: m.nama };
      }),
    };

    onSubmit(payload);
  };
}

const handleCancel = () => {
  if (onClose) onClose();
};

return (
  <form onSubmit={handleSubmit} className="p-4 space-y-4">
    <div>
      <Label htmlFor="nama">Nama Kelas</Label>
      <Input
        id="nama"
        type="text"
        value={formData.nama}
        onChange={handleChange}
        required
      />
    </div>
    <div>
      <Label htmlFor="mataKuliahId">Mata Kuliah</Label>
      <select
        id="mataKuliahId"
        value={formData.mataKuliahId}
        onChange={handleChange}
        className="w-full p-2 border rounded"
        required
      >
        <option value="">-- Pilih Mata Kuliah --</option>
        {mataKuliahs.map((mk) => (
          <option key={mk.id} value={mk.id}>
            {mk.nama} ({mk.sks} SKS)
          </option>
        ))}
      </select>
    </div>
    <div>
      <Label htmlFor="dosenId">Dosen</Label>
      <select
        id="dosenId"
        value={formData.dosenId}
        onChange={handleChange}
        className="w-full p-2 border rounded"
        required
      >
        <option value="">-- Pilih Dosen --</option>
        {dosenList.map((d) => (
          <option key={d.id} value={d.id} disabled={!isDosenEligible(d)}>
            {d.nama} ({d.totalSKS}/{d.maxSKS} SKS)
          </option>
        )));
      </select>
    </div>
    <div>
      <Label>Mahasiswa</Label>
      <div className="max-h-40 overflow-y-auto border p-2 rounded space-y-1">
        {mahasiswaList.map((m) => {
          const idStr = String(m.id);
          const disabled = !isMahasiswaEligible(m);
          return (
            <div key={idStr}>
              <label className="inline-flex items-center space-x-2">
                <input
                  type="checkbox"
                  value={idStr}
                  checked={formData.mahasiswaIds.includes(idStr)}
                  onChange={() => handleCheckboxChange(idStr)}
                  disabled={disabled}
                />
                <span className={disabled ? "text-red-500" : ""}>
                  {m.nama} ({m.totalSKS}/{m.maxSKS} SKS)
                </span>
              </label>
            </div>
          );
        })}
      </div>
    </div>
    <div className="flex justify-end space-x-2 mt-4">
      <Button type="button" className="bg-gray-400" onClick={handleCancel}>
        Batalkan
      </Button>
      <Button type="submit" className="bg-blue-600 hover:bg-blue-700">
        {editData ? "Simpan Perubahan" : "Simpan"}
      </Button>
    </div>
  </form>
);
};

FormKelas.propTypes = {
  onSubmit: PropTypes.func.isRequired,
  onClose: PropTypes.func,
  editData: PropTypes.object,
};

export default FormKelas;
```



```
// src/Components>Select.js
import React from "react";

const base = "border rounded w-full focus:outline-none focus:ring-2 focus:ring-blue-500";
sizeVariants = {
  sm: "px-2 py-1 text-sm",
  md: "px-3 py-2 text-sm",
  lg: "px-4 py-2 text-base"
};

const Select = ({
  children,
  className = "",
  size = "md",
  isMulti = false,
  options = [],
  ...props
}) => {
  return (
    <select
      className={`${base} ${sizeVariants[size]} ${className}`}
      multiple={isMulti}
      {...props}
    >
      {options.length > 0
        ? options.map((opt) => (
          <option key={opt.value} value={opt.value}>
            {opt.label}
          </option>
        ))
        : children}
    </select>
  );
};

export default Select;
```



```
import axios from "../../../../../Utils/Helpers/AxiosInstance";

export const getAllDosen = () => axios.get("/dosen");

export const getDosen = (id) => axios.get(`/dosen/${id}`);

export const storeDosen = (data) => axios.post("/dosen", data);

export const updateDosen = (id, data) => axios.put(`/dosen/${id}`, data);
export const deleteDosen = (id) => axios.delete(`/dosen/${id}`);
```



```
import axios from "../../../../../Utils/Helpers/AxiosInstance";

// Ambil semua kelas
export const getAllKelas = () => axios.get("/kelas");

// Ambil 1 kelas
export const getKelas = (id) => axios.get(`/kelas/${id}`);

// Tambah kelas baru
export const storeKelas = (data) => axios.post("/kelas", data);

// Update kelas
export const updateKelas = (id, data) => axios.put(`/kelas/${id}`, data);
// Hapus kelas
export const deleteKelas = (id) => axios.delete(`/kelas/${id}`);
```



```
import axios from "../../../../../Utils/Helpers/AxiosInstance";

export const getAllMataKuliah = () => axios.get("/matakuliah");

export const getMataKuliah = (id) => axios.get(`/matakuliah/${id}`);

export const storeMataKuliah= (data) => axios.post("/matakuliah", data);

export const updateMataKuliah= (id, data) => axios.put(`/matakuliah/${id}`, data);

export const deleteMataKuliah = (id) => axios.delete(`/matakuliah/${id}`);
```

```
import React from "react";
import { NavLink } from "react-router-dom";
import { useAuthStateContext } from "../Context/AuthContext";

const Sidebar = () => {
  const { user } = useAuthStateContext();

  if (!user || !user.permissions) return null;

  const menuItems = [
    { to: "/dashboard", label: "Dashboard", icon: "🏠", permission: "dashboard.page" },
    { to: "/dashboard/mahasiswa", label: "Mahasiswa", icon: "👤", permission: "mahasiswa.page" },
    { to: "/dashboard/dosen", label: "Dosen", icon: "👨‍🎓", permission: "dosen.page" },
    { to: "/dashboard/matakuliah", label: "Mata Kuliah", icon: "💻", permission: "matakuliah.page" },
    { to: "/dashboard/rencanastudi", label: "Rencana Studi", icon: "📅", permission: "rencanastudi.page" },
  ];

  return (
    <aside className="w-64 bg-blue-800 text-white p-4 min-h-screen">
      <h2 className="text-2xl font-bold mb-8 text-white">Admin Panel</h2>
      <nav className="flex flex-col gap-2">
        {menuItems.map((item) =>
          user.permissions.includes(item.permission) ? (
            <NavLink
              key={item.to}
              to={item.to}
              className={({ isActive }) =>
                isActive
                  ? "flex items-center gap-2 bg-blue-600 text-white px-4 py-2 rounded"
                  : "flex items-center gap-2 hover:bg-blue-500 hover:text-white px-4 py-2 rounded"
              }
            >
              <span>{item.icon}</span>
              <span>{item.label}</span>
            </NavLink>
          ) : null
        )}
      </nav>
    </aside>
  );
};

export default Sidebar;
```

```
● ● ●
import React, { useState, useEffect, useCallback } from "react";
import toast from "react-hot-toast";
import Swal from "sweetalert2";
import Button from "../../Components/Button";
import ModalRencanaStudi from "./RencanaStudiModal";
import TableRencanaStudi from "./RencanaStudiTable";
import {
  getAllKelas,
  storeKelas,
  updateKelas,
  deleteKelas,
} from "../../Utils/Helpers/Apis/KelasApi";
import { getAllMahasiswa } from "../../Utils/Helpers/Apis/MahasiswaApi";
import { getAllDosen } from "../../Utils/Helpers/Apis/DosenApi";
import { getAllMataKuliah } from "../../Utils/Helpers/Apis/MataKuliahApi";

const RencanaStudi = () => {
  const [kelas, setKelas] = useState([]);
  const [mahasiswa, setMahasiswa] = useState([]);
  const [dosen, setDosen] = useState([]);
  const [mataKuliah, setMataKuliah] = useState([]);
  const [form, setForm] = useState({ mata_kuliah_id: "", dosen_id: "" });
  const [isModalOpen, setIsModalOpen] = useState(false);
  const [selectedMhs, setSelectedMhs] = useState({});
  const [selectedDsn, setSelectedDsn] = useState({});
  const [selectedMatkul, setSelectedMatkul] = useState({});
  const [searchTerm, setSearchTerm] = useState("");

  const showCustomToast = (message, type = "success") => {
    toast.custom(
      () => (
        <div
          style={{ background: type === "success" ? "#4ade80" : type === "error" ? "#f87171" : "#fc34d", color: "#1e3a8a", fontWeight: "bold", padding: "16px", borderRadius: "12px", fontSize: "16px", boxShadow: "0 4px 14px rgba(0,0,0,0.15)" }}
        >
          <span>{type === "success" ? "✓" : type === "error" ? "✗" : "▲"}</span>
          <span>{message}</span>
        </div>
      ),
      {
        position: "top-center",
        duration: 3000,
      }
    );
  };

  const fetchData = useCallback(async () => {
    try {
      const [resKelas, resDosen, resMahasiswa, resMataKuliah] =
        await Promise.all([
          getAllKelas(),
          getAllDosen(),
          getAllMahasiswa(),
          getAllMataKuliah(),
        ]);
      setKelas(resKelas.data);
      setDosen(resDosen.data);
      setMahasiswa(resMahasiswa.data);
      setMataKuliah(resMataKuliah.data);
    } catch (err) {
      showCustomToast("Gagal mengambil data.", "error");
    }
  }, [fetchData]);

  const mataKuliahSudahDipakai = kelas.map((k) => k.mata_kuliah_id);
  const mataKuliahBelumAdaKelas = mataKuliah.filter(
    (m) => !mataKuliahSudahDipakai.includes(m.id)
  );

  const getMaxSks = (id) => mahasiswa.find((m) => m.id === id)?.maxSKS || 0;
  const getDosenMaxSks = (id) => dosen.find((d) => d.id === id)?.maxSKS || 0;

  const handleAddMahasiswa = async (kelasItem, mhsId) => {
    if (!mhsId)
      return showCustomToast("Silakan pilih mahasiswa terlebih dahulu.", "error");

    const matkul = mataKuliah.find((m) => m.id === kelasItem.mata_kuliah_id);
    const sks = matkul?.sks || 0;

    const totalSks = kelas
      .filter((k) => (k.mahasiswa_ids || []).includes(mhsId))
      .map(
        (k) => mataKuliah.find((m) => m.id === k.mata_kuliah_id)?.sks || 0
      )
      .reduce((a, b) => a + b, 0);

    const maxSks = getMaxSks(mhsId);
    if (totalSks + sks > maxSks)
      return showCustomToast(`SKS melebihi batas maksimal ${maxSks}`, "error");

    if ((kelasItem.mahasiswa_ids || []).includes(mhsId))
      return showCustomToast("Mahasiswa sudah terdaftar", "error");

    await updateKelas(kelasItem.id, {
      ...kelasItem,
      mahasiswa_ids: [...(kelasItem.mahasiswa_ids || []), mhsId],
    });

    showCustomToast("Mahasiswa ditambahkan", "success");
    setSelectedMhs((prev) => ({ ...prev, [kelasItem.id]: "" }));
    fetchData();
  };

  const handleDeleteMahasiswa = async (kelasItem, mhsId) => {
    const updated = {
      ...kelasItem,
      mahasiswa_ids: kelasItem.mahasiswa_ids.filter((id) => id !== mhsId),
    };
    await updateKelas(kelasItem.id, updated);
    showCustomToast("Mahasiswa dihapus", "success");
    fetchData();
  };

  const handleChangeDosen = async (kelasItem) => {
    const dsnId = selectedDsn[kelasItem.id];
    if (!dsnId) return;

    if (dsnId === kelasItem.dosen_id)
      return showCustomToast("Dosen tidak berubah", "warning");

    const totalSksDosen = kelas
      .filter((k) => k.dosen_id === dsnId && k.id !== kelasItem.id)
      .map((k) => mataKuliah.find((m) => m.id === k.mata_kuliah_id)?.sks || 0)
      .reduce((a, b) => a + b, 0);

    const kelasSks =
      mataKuliah.find((m) => m.id === kelasItem.mata_kuliah_id)?.sks || 0;
    const maxSks = getDosenMaxSks(dsnId);

    if (totalSksDosen + kelasSks > maxSks)
      return showCustomToast(`Dosen melebihi batas maksimal SKS ${maxSks}`, "error");

    await updateKelas(kelasItem.id, { ...kelasItem, dosen_id: dsnId });
    showCustomToast("Dosen diperbarui", "success");
    fetchData();
  };

  const handleChangeMatkul = async (kelasItem) => {
    const newMatkulId = selectedMatkul[kelasItem.id];
    if (!newMatkulId)
      return showCustomToast("Pilih mata kuliah terlebih dahulu", "error");

    if (newMatkulId === kelasItem.mata_kuliah_id)
      return showCustomToast("Mata kuliah tidak berubah", "warning");

    const matkulDipakai = kelas.some(
      (k) => k.id !== kelasItem.id && k.mata_kuliah_id === newMatkulId
    );

    if (matkulDipakai)
      return showCustomToast("Mata kuliah ini sudah diajar oleh dosen lain!", "error");

    await updateKelas(kelasItem.id, {
      ...kelasItem,
      mata_kuliah_id: newMatkulId,
    });

    showCustomToast("Mata kuliah berhasil diperbarui", "success");
    fetchData();
  };

  const handleDeleteKelas = async (kelasId) => {
    const kls = kelas.find((k) => k.id === kelasId);
    const matkul = mataKuliah.find((m) => m.id === kls?.mata_kuliah_id);
    const namaMatkul = matkul?.nama || "kelas ini";

    const result = await Swal.fire({
      title: "Konfirmasi Hapus",
      text: `Yakin ingin menghapus Kelas ${namaMatkul}?`,
      icon: "warning",
      showCancelButton: true,
      confirmButtonColor: "#e3342f",
      cancelButtonColor: "#6c757d",
      confirmButtonText: "Ya, Hapus!",
      cancelButtonText: "Batal",
    });

    if (result.isConfirmed) {
      await deleteKelas(kelasId);
      showCustomToast("Kelas berhasil dihapus", "success");
      fetchData();
    }
  };

  const openAddModal = () => {
    setForm({ mata_kuliah_id: "", dosen_id: "" });
    setIsModalOpen(true);
  };

  const handleChange = (e) => {
    setForm({ ...form, [e.target.name]: e.target.value });
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    if (!form.mata_kuliah_id || !form.dosen_id)
      return showCustomToast("Form tidak lengkap", "error");

    const matkulDipakai = kelas.some(
      (k) => k.mata_kuliah_id === form.mata_kuliah_id
    );
    if (matkulDipakai)
      return showCustomToast("Mata kuliah sudah memiliki dosen pengampu!", "error");

    await storeKelas({ ...form, mahasiswa_ids: [] });
    showCustomToast("Kelas ditambahkan", "success");
    setIsModalOpen(false);
    fetchData();
  };

  const filteredKelas = kelas.filter((cls) => {
    const matkul =
      mataKuliah.find((m) => m.id === cls.mata_kuliah_id)?.nama?.toLowerCase() ||
      "";
    const dsn = dosen.find((d) => d.id === cls.dosen_id)?.nama?.toLowerCase() || "";
    return (
      matkul.includes(searchTerm.toLowerCase()) ||
      dsn.includes(searchTerm.toLowerCase())
    );
  });

  return (
    <div className="bg-white p-6 rounded shadow">
      <div className="flex flex-col md:flex-row justify-between items-center mb-4 gap-4">
        <h2 className="text-xl font-bold text-gray-800">Daftar Rencana Studi</h2>
        <div className="flex gap-2 w-full md:w-auto">
          <input type="text" placeholder="Cari nama atau kode..." value={searchTerm} onChange={(e) => setSearchTerm(e.target.value)} className="w-full md:w-64 border border-gray-300 rounded-lg px-4 py-2 focus:outline-none focus:ring-2 focus:ring-blue-500" />
          <Button className="bg-blue-600 hover:bg-blue-700" onClick={openAddModal}>+ Tambah Kelas</Button>
        </div>
      </div>
      <TableRencanaStudi
        kelas={filteredKelas}
        mahasiswa={mahasiswa}
        dosen={dosen}
        mataKuliah={mataKuliah}
        selectedMhs={selectedMhs}
        setSelectedMhs={setSelectedMhs}
        selectedDsn={selectedDsn}
        setSelectedDsn={setSelectedDsn}
        handleAddMahasiswa={handleAddMahasiswa}
        handleDeleteMahasiswa={handleDeleteMahasiswa}
        handleChangeDosen={handleChangeDosen}
        handleDeleteKelas={handleDeleteKelas}
        selectedMatkul={selectedMatkul}
        setSelectedMatkul={setSelectedMatkul}
        handleChangeMatkul={handleChangeMatkul}
      />
      <ModalRencanaStudi
        isOpen={isModalOpen}
        onClose={() => setIsModalOpen(false)}
        onChange={handleChange}
        onSubmit={handleSubmit}
        form={form}
        dosen={dosen}
        mataKuliah={mataKuliahBelumAdaKelas}
        kelas={kelas}
      />
    </div>
  );
};

export default RencanaStudi;
```

```
import Button from "../../Components/Button";

const ModalRencanaStudi = ({  
    isOpen,  
    onClose,  
    onSubmit,  
    onChange,  
    form,  
    dosen,  
    mataKuliah,  
    kelas,  
}) => {  
    if (!isOpen) return null;  
  
    // Ambil ID dari mata kuliah yang sudah digunakan dalam kelas  
    const mataKuliahSudahDipakai = kelas.map((k) => k.mata_kuliah_id);  
  
    return (  
        <div className="fixed inset-0 flex items-center justify-center bg-[rgba(0,0,0,0.3)] z-50"> <div className="bg-white rounded-lg shadow-lg w-full max-w-md">  
            <div className="flex justify-between items-center p-4 border-b">  
                <h2 className="text-lg font-semibold">Tambah Kelas Baru</h2>  
                <button  
                    onClick={onClose}  
                    className="text-gray-600 hover:text-red-500 text-xl"  
                >  
                    &times;  
                </button>  
            </div>  
  
            {/* Form Input */}  
            <form onSubmit={onSubmit} className="p-4 space-y-4">  
                {/* Mata Kuliah */}  
                <div>  
                    <label  
                        htmlFor="mata_kuliah_id"  
                        className="block font-medium text-sm text-gray-700"  
                    >  
                        Mata Kuliah  
                    </label>  
                    <select  
                        name="mata_kuliah_id"  
                        value={form.mata_kuliah_id}  
                        onChange={onChange}  
                        className="w-full border px-3 py-2 rounded"  
                        required  
                    >  
                        <option value="">-- Pilih Mata Kuliah --</option>  
                        {mataKuliah.map((m) => {  
                            const sudahDipakai = mataKuliahSudahDipakai.includes(m.id);  
                            return (  
                                <option key={m.id} value={m.id} disabled={sudahDipakai}>  
                                    {m.nama} {sudahDipakai ? "(Sudah memiliki dosen)" : ""}  
                                </option>  
                            );  
                        })}  
                    </select>  
                </div>  
  
                {/* Dosen */}  
                <div>  
                    <label  
                        htmlFor="dosen_id"  
                        className="block font-medium text-sm text-gray-700"  
                    >  
                        Dosen Pengampu  
                    </label>  
                    <select  
                        name="dosen_id"  
                        value={form.dosen_id}  
                        onChange={onChange}  
                        className="w-full border px-3 py-2 rounded"  
                        required  
                    >  
                        <option value="">-- Pilih Dosen --</option>  
                        {dosen.map((d) => (  
                            <option key={d.id} value={d.id}>  
                                {d.nama}  
                            </option>  
                        ))}  
                    </select>  
                </div>  
  
                {/* Tombol Aksi */}  
                <div className="flex justify-end gap-2">  
                    <Button  
                        type="button"  
                        onClick={onClose}  
                        className="bg-gray-500 hover:bg-gray-600"  
                    >  
                        Batal  
                    </Button>  
                    <Button type="submit" className="bg-blue-600 hover:bg-blue-700">  
                        Simpan  
                    </Button>  
                </div>  
            </form>  
        </div>  
    );  
  
    export default ModalRencanaStudi;
```

```
import React from "react";

export default function TableRencanaStudi({
  kelas,
  mahasiswa,
  dosen,
  mataKuliah,
  selectedMhs,
  setSelectedMhs,
  selectedDsn,
  setSelectedDsn,
  selectedMatkul,
  setSelectedMatkul,
  handleAddMahasiswa,
  handleDeleteMahasiswa,
  handleChangeDosen,
  handleChangeMatkul,
  handleDeleteKelas,
}) {
  return (
    <div className="space-y-6">
      {kelas.map((cls) => {
        const matkul = mataKuliah.find(
          (m) => m.id.toString() === cls.mata_kuliah_id?.toString()
        );
        const dosenPengampu = dosen.find(
          (d) => d.id.toString() === cls.dosen_id?.toString()
        );
        const mhsInClass = (cls.mahasiswa_ids || cls.mahasiswaIds || []).map((id) =>
          mahasiswa.find((m) => m.id.toString() === id.toString())
        );
        .filter(Boolean);

        return (
          <div key={cls.id} className="border rounded-lg shadow-md bg-white transition hover:shadow-lg">
            {/* Header */}
            <div className="px-6 py-4 border-b bg-gray-50">
              <div className="flex flex-col md:flex-row justify-between items-start md:items-center gap-4">
                <div>
                  <p>Mata Kuliah</p>
                  <h3> {matkul?.nama || "-"} </h3>
                  <p>Dosen Pengampu</p>
                  <h3> {dosenPengampu?.nama || "-"} </h3>
                </div>
                {mhsInClass.length === 0 && (
                  <button onClick={() => handleDeleteKelas(cls.id)} className="bg-red-600 hover:bg-red-700 text-white px-4 py-1.5 rounded text-sm">
                    Hapus Kelas
                  </button>
                )}
                </div>
                <div className="mt-4 flex flex-col lg:flex-row flex-wrap gap-3">
                  /* Mata Kuliah */
                  <div className="flex flex-col sm:flex-row gap-2">
                    <select value={selectedMatkul[cls.id] || ""}>
                      onChange={(e) => setSelectedMatkul({
                        ...selectedMatkul,
                        [cls.id]: e.target.value,
                      })}
                    </select>
                    <option value="">-- Ganti Mata Kuliah --</option>
                    {mataKuliah.map((m) => (
                      <option key={m.id} value={m.id}>
                        {m.nama}
                      </option>
                    )));
                    </select>
                    <button onClick={() => handleChangeMatkul(cls)} className="bg-indigo-600 hover:bg-indigo-700 text-white px-4 py-1.5 rounded text-sm">
                      Simpan Mata Kuliah
                    </button>
                  </div>
                  /* Dosen */
                  <div className="flex flex-col sm:flex-row gap-2">
                    <select value={selectedDsn[cls.id] || ""}>
                      onChange={(e) => setSelectedDsn({
                        ...selectedDsn,
                        [cls.id]: e.target.value,
                      })}
                    </select>
                    <option value="">-- Ganti Dosen --</option>
                    {dosen.map((d) => (
                      <option key={d.id} value={d.id}>
                        {d.nama}
                      </option>
                    )));
                    </select>
                    <button onClick={() => handleChangeDosen(cls)} className="bg-blue-600 hover:bg-blue-700 text-white px-4 py-1.5 rounded text-sm">
                      Simpan Dosen
                    </button>
                  </div>
                </div>
              </div>
            {/* Tabel Mahasiswa */}
            <div className="overflow-x-auto">
              <table className="w-full text-sm">
                <thead className="bg-blue-700 text-white">
                  <tr>
                    <th>No</th>
                    <th>Nama</th>
                    <th>NIM</th>
                    <th>Total SKS</th>
                    <th>Aksi</th>
                  </tr>
                </thead>
                <tbody>
                  {mhsInClass.length > 0 ? (
                    mhsInClass.map((m, i) => {
                      const totalSks = kelas
                        .filter((k) =>
                          (k.mahasiswa_ids || []).includes(m.id)
                        )
                        .map(
                          (k) =>
                            mataKuliah.find(
                              (mk) =>
                                mk.id.toString() ===
                                  k.mata_kuliah_id?.toString()
                            )?.sks || 0
                        )
                        .reduce((a, b) => a + b, 0);
                      return (
                        <tr key={m.id}>
                          {i % 2 === 0 ? "bg-white" : "bg-gray-50"}>
                            <td>{i + 1}</td>
                            <td>{m.nama}</td>
                            <td>{m.nim}</td>
                            <td>{totalSks}</td>
                            <td>
                              <button onClick={() => handleDeleteMahasiswa(kls, m.id)}>
                                Hapus
                              </button>
                            </td>
                          </tr>
                      );
                    ) : (
                      <tr>
                        <td colSpan="5" className="py-3 px-4 text-center italic text-gray-500">
                          Belum ada mahasiswa.
                        </td>
                      </tr>
                    );
                  ) : (
                    <tr>
                      <td>
                        <div>
                          <select value={selectedMhs[cls.id] || ""}>
                            onChange={(e) => setSelectedMhs({
                              ...selectedMhs,
                              [cls.id]: e.target.value,
                            })}
                          </select>
                          <option value="">-- Pilih Mahasiswa --</option>
                          {mahasiswa.map((m) => {
                            const totalSksMahasiswa = kelas
                              .filter((k) =>
                                (k.mahasiswa_ids || []).includes(m.id)
                              )
                              .map(
                                (k) =>
                                  mataKuliah.find(
                                    (mk) =>
                                      mk.id.toString() ===
                                        k.mata_kuliah_id?.toString()
                                    )?.sks || 0
                              )
                              .reduce((a, b) => a + b, 0);
                            const maxSks = m.maxSKS || 0;
                            return (
                              <option key={m.id} value={m.id}>
                                {m.nama} ({m.nim}) - {totalSksMahasiswa} / {maxSks} SKS
                              </option>
                            );
                          })}
                        </select>
                        <button onClick={() => handleAddMahasiswa(kls, selectedMhs[cls.id])}>
                          Tambah Mahasiswa
                        </button>
                      </div>
                    </td>
                  ) : (
                    <td>
                      <div>
                        <select value={selectedMhs[cls.id] || ""}>
                          onChange={(e) => setSelectedMhs({
                            ...selectedMhs,
                            [cls.id]: e.target.value,
                          })}
                        </select>
                        <option value="">-- Pilih Mahasiswa --</option>
                        {mahasiswa.map((m) => {
                          const totalSksMahasiswa = kelas
                            .filter((k) =>
                              (k.mahasiswa_ids || []).includes(m.id)
                            )
                            .map(
                              (k) =>
                                mataKuliah.find(
                                  (mk) =>
                                    mk.id.toString() ===
                                      k.mata_kuliah_id?.toString()
                                )?.sks || 0
                              )
                              .reduce((a, b) => a + b, 0);
                          const maxSks = m.maxSKS || 0;
                          return (
                            <option key={m.id} value={m.id}>
                              {m.nama} ({m.nim}) - {totalSksMahasiswa} / {maxSks} SKS
                            </option>
                          );
                        })}
                      </select>
                      <button onClick={() => handleAddMahasiswa(kls, selectedMhs[cls.id])}>
                        Tambah Mahasiswa
                      </button>
                    </div>
                  );
                ) : (
                  <td>
                    <div>
                      <select value={selectedMhs[cls.id] || ""}>
                        onChange={(e) => setSelectedMhs({
                          ...selectedMhs,
                          [cls.id]: e.target.value,
                        })}
                      </select>
                      <option value="">-- Pilih Mahasiswa --</option>
                      {mahasiswa.map((m) => {
                        const totalSksMahasiswa = kelas
                          .filter((k) =>
                            (k.mahasiswa_ids || []).includes(m.id)
                          )
                          .map(
                            (k) =>
                              mataKuliah.find(
                                (mk) =>
                                  mk.id.toString() ===
                                    k.mata_kuliah_id?.toString()
                              )?.sks || 0
                            )
                            .reduce((a, b) => a + b, 0);
                        const maxSks = m.maxSKS || 0;
                        return (
                          <option key={m.id} value={m.id}>
                            {m.nama} ({m.nim}) - {totalSksMahasiswa} / {maxSks} SKS
                          </option>
                        );
                      })}
                    </select>
                    <button onClick={() => handleAddMahasiswa(kls, selectedMhs[cls.id])}>
                      Tambah Mahasiswa
                    </button>
                  </div>
                );
              </td>
            );
          </tbody>
        );
      ) : (
        <tr>
          <td>
            <div>
              <select value={selectedMhs[cls.id] || ""}>
                onChange={(e) => setSelectedMhs({
                  ...selectedMhs,
                  [cls.id]: e.target.value,
                })}
              </select>
              <option value="">-- Pilih Mahasiswa --</option>
              {mahasiswa.map((m) => {
                const totalSksMahasiswa = kelas
                  .filter((k) =>
                    (k.mahasiswa_ids || []).includes(m.id)
                  )
                  .map(
                    (k) =>
                      mataKuliah.find(
                        (mk) =>
                          mk.id.toString() ===
                            k.mata_kuliah_id?.toString()
                      )?.sks || 0
                    )
                    .reduce((a, b) => a + b, 0);
                const maxSks = m.maxSKS || 0;
                return (
                  <option key={m.id} value={m.id}>
                    {m.nama} ({m.nim}) - {totalSksMahasiswa} / {maxSks} SKS
                  </option>
                );
              })}
            </select>
            <button onClick={() => handleAddMahasiswa(kls, selectedMhs[cls.id])}>
              Tambah Mahasiswa
            </button>
          </div>
        );
      );
    );
  );
}
```

```
● ○ ●
import React, { useState, useEffect, useCallback } from "react";
import { useNavigate } from "react-router-dom";
import toast from "react-hot-toast";
import Swal from "sweetalert2";
import AxiosInstance from "../../Utils/Helpers/AxiosInstance";
import Button from "../../Components/Button";
import Modal from "../../Components/Modal";
import FormMahasiswa from "../../Components/FormMahasiswa";
import { useAuthStateContext } from "../../Context/AuthContext";

const Mahasiswa = () => {
  const [isModalOpen, setIsModalOpen] = useState(false);
  const [mahasiswaList, setMahasiswaList] = useState([]);
  const [selectedMahasiswa, setSelectedMahasiswa] = useState(null);
  const [searchTerm, setSearchTerm] = useState("");
  const [currentPage, setCurrentPage] = useState(1);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);
  const itemsPerPage = 5;

  const navigate = useNavigate();
  const { user } = useAuthStateContext();

  const showCustomToast = (message, type = "success") => {
    toast.custom(
      () => (
        <div
          style={{background: type === "success" ? "#4ade80" : type === "error" ? "#f87171" : "#fcd3d4", color: "#1e3a8a", fontWeight: "bold", padding: "16px", borderRadius: "12px", fontSize: "16px", boxShadow: "0 4px 14px rgba(0,0,0,0.15)"}}
        >
          <span>{type === "success" ? "✓" : type === "error" ? "✗" : "⚠"}</span>
          <span>{message}</span>
        </div>
      ),
      {
        position: "top-center",
        duration: 3000,
      }
    );
  };

  // ✅ Sinkron total SKS dari kelas dan matakuliah
  const fetchMahasiswa = useCallback(async () => {
    setLoading(true);
    setError(null);
    try {
      const [resKelas, resMatakuliah] = await Promise.all([
        AxiosInstance.get("/mahasiswa"),
        AxiosInstance.get("/kelas"),
        AxiosInstance.get("/matakuliah"),
      ]);

      const mahasiswaWithSKS = resMatakuliah.data.map((mhs) => {
        const totalSKS = resKelas.data.filter((kls) => kls.mahasiswa_ids || []).includes(mhs.id)
          .map((kls) => resMatakuliah.data.find((mk) => mk.id === kls.mata_kuliah_id)).reduce((a, b) => a + b, 0);

        return { ...mhs, totalSKS };
      });

      setMahasiswaList(mahasiswaWithSKS);
    } catch (err) {
      setError(err);
      showCustomToast("Gagal memuat data mahasiswa.", "error");
      console.error(`Error fetching mahasiswa: ${err}`);
    } finally {
      setLoading(false);
    }
  }, []);

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

  const storeMahasiswa = async (newData) => {
    if (!newData.nim || !newData.nama) {
      showCustomToast("NIM dan Nama harus diisi!", "error");
      return;
    }

    try {
      const checkResponse = await AxiosInstance.get(`/mahasiswa?nim=${newData.nim}`);
      if (checkResponse.data.length > 0) {
        showCustomToast("NIM sudah digunakan!", "error");
        return;
      }

      await AxiosInstance.post("/mahasiswa", newData);
      setIsModalOpen(false);
      showCustomToast("Data mahasiswa berhasil ditambahkan!", "success");
      fetchMahasiswa();
    } catch (err) {
      showCustomToast("Gagal menambahkan data mahasiswa.", "error");
      console.error(`Error storing mahasiswa: ${err}`);
    }
  };

  const updateMahasiswa = async (updatedData) => {
    if (!updatedData.nim || !updatedData.nama) {
      showCustomToast("NIM dan Nama harus diisi!", "error");
      return;
    }

    if (selectedMahasiswa?.nim !== updatedData.nim) {
      const checkResponse = await AxiosInstance.get(`/mahasiswa?nim=${updatedData.nim}`);
      if (checkResponse.data.length > 0) {
        showCustomToast("NIM sudah digunakan oleh mahasiswa lain!", "error");
        return;
      }
    }

    const result = await Swal.fire({
      title: "Konfirmasi Update",
      text: `Yakin ingin mengupdate data mahasiswa NIM ${selectedMahasiswa?.nim}?`,
      icon: "question",
      showCancelButton: true,
      confirmButtonColor: "#3088d6",
      cancelButtonColor: "#d33",
      confirmButtonText: "Ya, Update!",
      cancelButtonText: "Batal",
    });

    if (result.isConfirmed) {
      try {
        await AxiosInstance.put(`/mahasiswa/${selectedMahasiswa.id}`, updatedData);
        setIsModalOpen(false);
        setSelectedMahasiswa(null);
        showCustomToast("Data mahasiswa berhasil diupdate!", "success");
        fetchMahasiswa();
      } catch (err) {
        showCustomToast("Gagal mengupdate data mahasiswa.", "error");
        console.error(`Error updating mahasiswa: ${err}`);
      }
    }
  };

  const deleteMahasiswa = async (id) => {
    const mhs = mahasiswaList.find((m) => m.id === id);

    const result = await Swal.fire({
      title: "Konfirmasi Hapus",
      text: `Yakin ingin menghapus data mahasiswa "${mhs?.nama}" (NIM: ${mhs?.nim})?`,
      icon: "warning",
      showCancelButton: true,
      confirmButtonColor: "#e3342f",
      cancelButtonColor: "#6c757d",
      confirmButtonText: "Ya, Hapus!",
      cancelButtonText: "Batal",
    });

    if (result.isConfirmed) {
      try {
        await AxiosInstance.delete(`/mahasiswa/${id}`);
        showCustomToast("Data mahasiswa berhasil dihapus!", "success");
        fetchMahasiswa();
      } catch (err) {
        showCustomToast("Gagal menghapus data mahasiswa.", "error");
        console.error(`Error deleting mahasiswa: ${err}`);
      }
    }
  };

  const openAddModal = () => {
    setSelectedMahasiswa(null);
    setIsModalOpen(true);
  };

  const openEditModal = async (mhs) => {
    const result = await Swal.fire({
      title: "Konfirmasi Edit",
      text: `Edit data mahasiswa "${mhs?.nama}"?`,
      icon: "question",
      showCancelButton: true,
      confirmButtonColor: "#ff5900",
      cancelButtonColor: "#6c757d",
      confirmButtonText: "Ya, Edit!",
      cancelButtonText: "Batal",
    });

    if (result.isConfirmed) {
      setSelectedMahasiswa(mhs);
      setIsModalOpen(true);
    }
  };

  const handleSubmit = (data) => {
    if (selectedMahasiswa) {
      updateMahasiswa({ ...data, id: selectedMahasiswa.id });
    } else {
      storeMahasiswa(data);
    }
  };

  const filteredMahasiswa = mahasiswaList.filter((mhs) =>
    mhs.nama.toLowerCase().includes(searchTerm.toLowerCase()) ||
    mhs.nim.toLowerCase().includes(searchTerm.toLowerCase())
  );

  const indexOfLastItem = currentPage * itemsPerPage;
  const indexOfFirstItem = indexOfLastItem - itemsPerPage;
  const currentMahasiswa = filteredMahasiswa.slice(indexOfFirstItem, indexOfLastItem);
  const totalPages = Math.ceil(filteredMahasiswa.length / itemsPerPage);

  const handlePrevious = () => {
    if (currentPage > 1) setCurrentPage(currentPage - 1);
  };

  const handleNext = () => {
    if (currentPage < totalPages) setCurrentPage(currentPage + 1);
  };

  if (!user?.permissions?.includes("mahasiswa.page")) {
    return (
      <div className="flex justify-center items-center h-64">
        <p> Anda tidak memiliki izin untuk melihat halaman ini. </p>
      </div>
    );
  }

  if (loading) {
    return (
      <div className="flex justify-center items-center h-64">
        <p> Memuat data mahasiswa...</p>
      </div>
    );
  }

  return (
    <div className="bg-white shadow rounded-lg p-6">
      <div className="flex flex-col md:flex-row justify-between items-center mb-4 gap-4">
        <h2>Daftar Mahasiswa</h2>
        <div className="flex gap-2 w-full md:w-auto">
          <input type="text" placeholder="Cari nama atau NIM..." value={searchTerm} onChange={(e) => setSearchTerm(e.target.value)}>
          <button> Tambah Mahasiswa </button>
        </div>
      </div>
      <div className="overflow-x-auto">
        <table className="w-full text-sm text-gray-700">
          <thead className="bg-blue-600 text-white">
            <tr>
              <th>NIM</th>
              <th>Nama</th>
              <th>Email</th>
              <th>Jurusan</th>
              <th>Maksimal SKS</th>
              <th>Total SKS</th>
              <th>Aksi</th>
            </tr>
          </thead>
          <tbody>
            {currentMahasiswa.length === 0 ? (
              <tr>
                <td colspan="7" className="text-center py-6 text-gray-500"> Tidak ada data mahasiswa yang sesuai </td>
              </tr>
            ) : (
              currentMahasiswa.map((mhs) => (
                <tr key={mhs.id} className="even:bg-gray-100 odd:bg-white">
                  <td className="py-2 px-4 font-medium">{mhs.nim}</td>
                  <td className="py-2 px-4">{mhs.name || "N/A"}</td>
                  <td className="py-2 px-4">{mhs.email}</td>
                  <td className="py-2 px-4">{mhs.jurusan}</td>
                  <td className="py-2 px-4">{mhs.maxSKS}</td>
                  <td className="py-2 px-4 text-center space-x-2">
                    <button> Detail </button>
                    {user.permissions.includes("mahasiswa.update") && (
                      <button> Edit </button>
                      <button> Hapus </button>
                    )}
                  </td>
                </tr>
              ))
            )}
          </tbody>
        </table>
      </div>
      <div className="flex flex-col items-center gap-2">
        <button> Previous </button>
        <button> Next </button>
        <span> Halaman {currentPage} dari {totalPages} </span>
      </div>
    </div>
  );
}

if (error) {
  return (
    <div className="flex justify-center items-center h-64">
      <p> Error: Gagal memuat data. Silakan coba lagi. ({error.message}) </p>
    </div>
  );
}

return (
  <div className="bg-white shadow rounded-lg p-6">
    <div className="flex flex-col md:flex-row justify-between items-center mb-4 gap-4">
      <h2>Daftar Mahasiswa</h2>
      <div className="flex gap-2 w-full md:w-auto">
        <input type="text" placeholder="Cari nama atau NIM..." value={searchTerm} onChange={(e) => setSearchTerm(e.target.value)}>
        <button> Tambah Mahasiswa </button>
      </div>
    </div>
    <div className="overflow-x-auto">
      <table className="w-full text-sm text-gray-700">
        <thead className="bg-blue-600 text-white">
          <tr>
            <th>NIM</th>
            <th>Nama</th>
            <th>Email</th>
            <th>Jurusan</th>
            <th>Maksimal SKS</th>
            <th>Total SKS</th>
            <th>Aksi</th>
          </tr>
        </thead>
        <tbody>
          {currentMahasiswa.length === 0 ? (
            <tr>
              <td colspan="7" className="text-center py-6 text-gray-500"> Tidak ada data mahasiswa yang sesuai </td>
            </tr>
          ) : (
            currentMahasiswa.map((mhs) => (
              <tr key={mhs.id} className="even:bg-gray-100 odd:bg-white">
                <td className="py-2 px-4 font-medium">{mhs.nim}</td>
                <td className="py-2 px-4">{mhs.name || "N/A"}</td>
                <td className="py-2 px-4">{mhs.email}</td>
                <td className="py-2 px-4">{mhs.jurusan}</td>
                <td className="py-2 px-4">{mhs.maxSKS}</td>
                <td className="py-2 px-4 text-center space-x-2">
                  <button> Detail </button>
                  {user.permissions.includes("mahasiswa.update") && (
                    <button> Edit </button>
                    <button> Hapus </button>
                  )}
                </td>
              </tr>
            ))
          )}
        </tbody>
      </table>
    </div>
    <div className="flex flex-col items-center gap-2">
      <button> Previous </button>
      <button> Next </button>
      <span> Halaman {currentPage} dari {totalPages} </span>
    </div>
  </div>
);

if (isModalOpen) onClose={() => setIsModalOpen(false)};
<Modal isOpen={isModalOpen} onOpen={handleAddModal} onSubmit={handleSubmit} editData={selectedMahasiswa} />
</Modal>
</div>
);

export default Mahasiswa;
```

```
import React from "react";
import { useAuth } from "../../Context/AuthContext";

const MahasiswaTable = ({ mahasiswa, openEditModal, onDelete, onDetail }) => {
  const { user } = useAuth();

  return (
    <div className="overflow-x-auto">
      <table className="w-full text-sm text-gray-700">
        <thead className="bg-blue-600 text-white">
          <tr>
            <th className="py-3 px-4 text-left">ID</th>
            <th className="py-3 px-4 text-left">NIM</th>
            <th className="py-3 px-4 text-left">Nama</th>
            <th className="py-3 px-4 text-left">Email</th>
            <th className="py-3 px-4 text-left">Jurusan</th>
            <th className="py-3 px-4 text-left">Maksimal SKS</th>
            <th className="py-3 px-4 text-left">Total SKS</th>
            <th className="py-3 px-4 text-center">Aksi</th>
          </tr>
        </thead>
        <tbody>
          {mahasiswa.length === 0 ? (
            <tr>
              <td colSpan="7" className="text-center py-6 text-gray-500">
                Tidak ada data mahasiswa.
              </td>
            </tr>
          ) : (
            mahasiswa.map((mhs) => (
              <tr key={mhs.id} className="even:bg-gray-100 odd:bg-white">
                <td className="py-2 px-4 font-medium">{mhs.id}</td>
                <td className="py-2 px-4 font-medium">{mhs.nim}</td>
                <td className="py-2 px-4">{mhs.nama}</td>
                <td className="py-2 px-4">{mhs.email}</td>
                <td className="py-2 px-4">{mhs.jurusan || 'N/A'}</td>
                <td className="py-2 px-4">{mhs.maxSKS}</td>
                <td className="py-2 px-4">{mhs.totalSKS}</td>
                <td className="py-2 px-4 text-center space-x-2">
                  {onDetail && (
                    <button
                      onClick={() => onDetail(mhs.id)}
                      className="bg-blue-500 text-white px-3 py-1 rounded hover:bg-blue-600"
                    >
                      Detail
                    </button>
                  )}
                  {user?.permission.includes("mahasiswa.update") && (
                    <button
                      onClick={() => openEditModal(mhs)}
                      className="bg-yellow-500 text-white px-3 py-1 rounded hover:bg-yellow-600"
                    >
                      Edit
                    </button>
                  )}
                  {user?.permission.includes("mahasiswa.delete") && (
                    <button
                      onClick={() => onDelete(mhs.id)}
                      className="bg-red-500 text-white px-3 py-1 rounded hover:bg-red-600"
                    >
                      Hapus
                    </button>
                  )}
                </td>
              </tr>
            )))
          )}
        </tbody>
      </table>
    </div>
  );
};

export default MahasiswaTable;
```

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

const MahasiswaModal = ({  
    isModalOpen,  
    onClose,  
    onSubmit,  
    selectedMahasiswa,  
}) => {  
    const [form, setForm] = useState({  
        nim: "",  
        nama: "",  
        email: "",  
        jurusan: "",  
        maxSKS: 0,  
        totalSKS: 0,  
    });  
    const [errors, setErrors] = useState({});  
  
    useEffect(() => {  
        if (selectedMahasiswa) {  
            setForm(selectedMahasiswa);  
        } else {  
            setForm({ nim: "", nama: "", email: "", jurusan: "", maxSKS: 0, totalSKS: 0 });  
        }  
    }, [selectedMahasiswa, isModalOpen]);  
  
    const handleChange = (e) => {  
        const { name, value } = e.target;  
        setForm({ ...form, [name]: value });  
    };  
  
    const validate = () => {  
        let tempErrors = {};  
        if (!form.nim) tempErrors.nim = "NIM wajib diisi";  
        if (!form.nama) tempErrors.nama = "Nama wajib diisi";  
        if (!form.email) tempErrors.email = "Email wajib diisi";  
        if (!form.jurusan) tempErrors.jurusan = "Jurusan wajib diisi";  
        if (!form.maxSKS) tempErrors.maxSKS = "Maksimal SKS wajib diisi";  
        if (!form.totalSKS) tempErrors.totalSKS = "Total SKS wajib diisi";  
        setErrors(tempErrors);  
        return Object.keys(tempErrors).length === 0;  
    };  
  
    const handleSubmit = (e) => {  
        e.preventDefault();  
        if (validate()) {  
            onSubmit(form);  
            onClose();  
        }  
    };  
  
    if (!isModalOpen) return null;  
  
    return (  
        <div className="fixed inset-0 bg-black bg-opacity-50 flex justify-center items-center z-50">  
            <div className="bg-white p-6 rounded-xl shadow-lg w-full max-w-md">  
                {selectedMahasiswa ? "Edit Mahasiswa" : "Tambah Mahasiswa"}  
                </h2>  
                <form onSubmit={handleSubmit}>  
                    <div className="mb-4">  
                        <label className="block mb-1">NIM</label>  
                        <input  
                            type="text"  
                            name="nim"  
                            value={form.nim}  
                            onChange={handleChange}  
                            className="w-full border p-2 rounded"  
                            disabled={!selectedMahasiswa}>  
                    />  
                    {errors.nim && (  
                        <p className="text-red-500 text-sm mt-1">{errors.nim}</p>  
                    )}  
                </div>  
                <div className="mb-4">  
                    <label className="block mb-1">Nama</label>  
                    <input  
                        type="text"  
                        name="nama"  
                        value={form.nama}  
                        onChange={handleChange}  
                        className="w-full border p-2 rounded">  
                />  
                {errors.nama && (  
                    <p className="text-red-500 text-sm mt-1">{errors.nama}</p>  
                )}  
            </div>  
            <div className="mb-4">  
                <label className="block mb-1">Email</label>  
                <input  
                    type="email"  
                    name="email"  
                    value={form.email}  
                    onChange={handleChange}  
                    className="w-full border p-2 rounded">  
            />  
            {errors.email && (  
                <p className="text-red-500 text-sm mt-1">{errors.email}</p>  
            )}  
        </div>  
        <div className="mb-4">  
            <label className="block mb-1">Jurusan</label>  
            <input  
                type="text"  
                name="jurusan"  
                value={form.jurusan}  
                onChange={handleChange}  
                className="w-full border p-2 rounded">  
        />  
        {errors.jurusan && (  
            <p className="text-red-500 text-sm mt-1">{errors.jurusan}</p>  
        )}  
    </div>  
    <div className="mb-4">  
        <label className="block mb-1">Maksimal SKS</label>  
        <input  
            type="number"  
            name="maxSKS"  
            value={form.maxSKS}  
            onChange={handleChange}  
            className="w-full border p-2 rounded">  
    />  
    {errors.maxSKS && (  
        <p className="text-red-500 text-sm mt-1">{errors.maxSKS}</p>  
    )}  
    </div>  
    <div className="mb-4">  
        <label className="block mb-1">Total SKS</label>  
        <input  
            type="number"  
            name="totalSKS"  
            value={form.totalSKS}  
            onChange={handleChange}  
            className="w-full border p-2 rounded">  
    />  
    {errors.totalSKS && (  
        <p className="text-red-500 text-sm mt-1">{errors.totalSKS}</p>  
    )}  
    </div>  
    <div className="flex justify-end space-x-2">  
        <button  
            type="button"  
            onClick={onClose}  
            className="px-4 py-2 bg-gray-300 rounded hover:bg-gray-400">  
            Batal  
        </button>  
        <button  
            type="submit"  
            className="px-4 py-2 bg-blue-500 text-white rounded hover:bg-blue-600">  
            Simpan  
        </button>  
    </div>  
    </form>  
    </div>  
    </div>  
);  
  
export default MahasiswaModal;
```

```
// src/App.js
import React from "react";
import { BrowserRouter as Router, Routes, Route, Navigate } from "react-router-dom";
import { Toaster } from "react-hot-toast";

import { AuthProvider, useAuthStateContext } from "./Context/AuthContext";
import Login from "./Pages/Auth/Login";
import Register from "./Pages/Auth/Register";
import Dashboard from "./Pages/Admin/Dashboard";
import AuthLayout from "./Layouts/AuthLayout";
import MainLayout from "./Layouts/AdminLayout";
import ProtectedRoute from "./Routes/ProtectedRoute";

import Mahasiswa from "./Pages/Mahasiswa/Mahasiswa";
import MahasiswaDetail from "./Pages/MahasiswaDetail";
import Dosen from "./Pages/Dosen/Dosen";
import DosenDetail from "./Pages/Dosen/DosenDetail";
import MataKuliah from "./Pages/MataKuliah/MataKuliah";
import MataKuliahDetail from "./Pages/MataKuliah/MataKuliahDetail";
import RencanaStudi from "./Pages/RencanaStudi/RencanaStudi";

const AppRoutes = () => {
  const { isLoading } = useAuthStateContext();

  if (isLoading) {
    return (
      <div className="h-screen flex items-center justify-center">
        <p className="text-lg font-medium text-gray-600">Memuat aplikasi...</p>
      </div>
    );
  }

  return (
    <Routes>
      <Route path="/" element={<Navigate to="/login" />} />

      <Route element={<AuthLayout />}>
        <Route path="/login" element={<Login />} />
        <Route path="/register" element={<Register />} />
      </Route>

      <Route element={<ProtectedRoute requiredRole={[ "admin", "dosen", "mahasiswa" ]} />}>
        <Route path="/dashboard" element={<MainLayout />}>
          <Route index element={<Dashboard />} />

          <Route path="mahasiswa" element={<Mahasiswa />} />
          <Route path="mahasiswa/:id" element={<MahasiswaDetail />} />

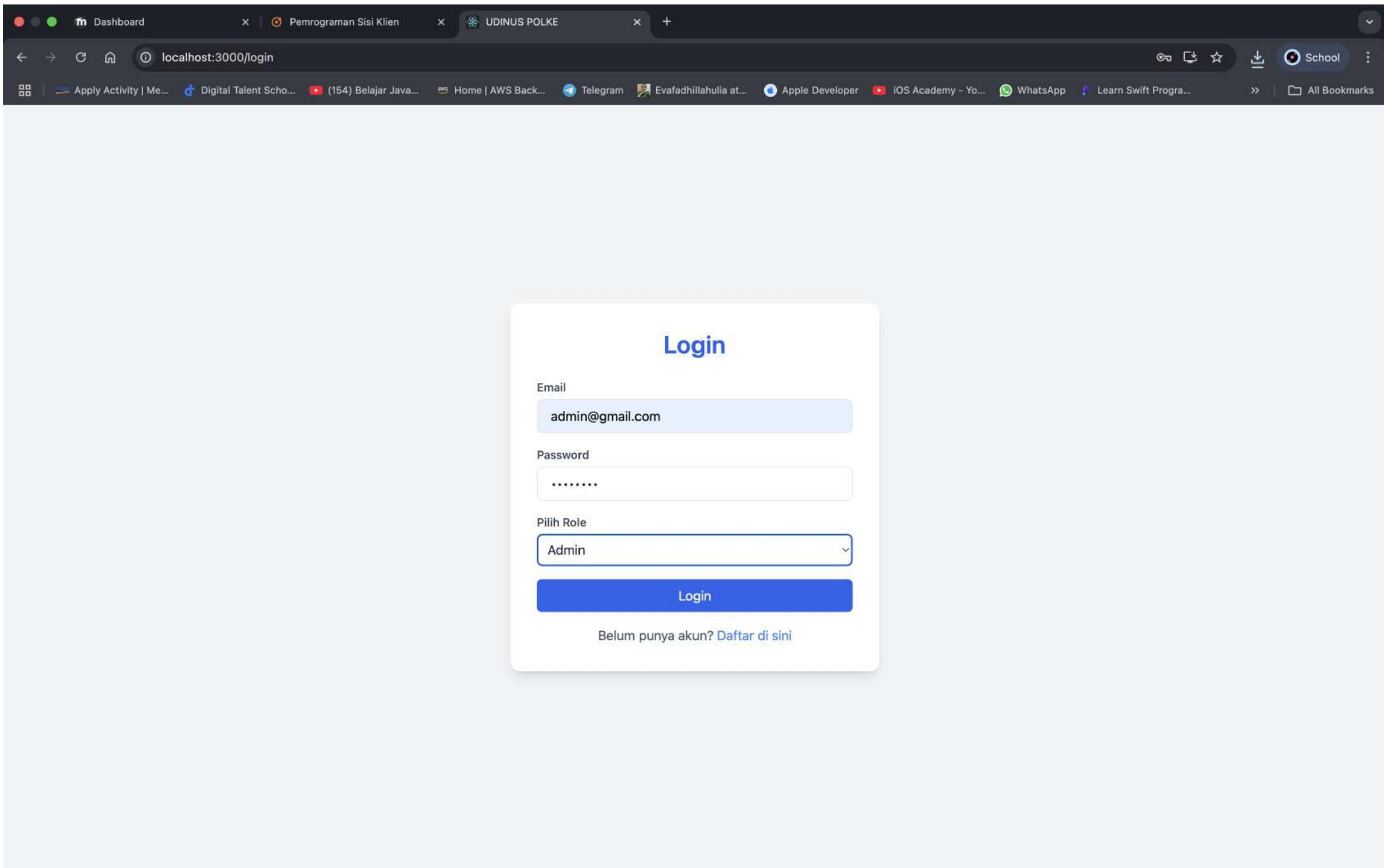
          <Route path="dosen" element={<Dosen />} />
          <Route path="dosen/:id" element={<DosenDetail />} />

          <Route path="matakuliah" element={<MataKuliah />} />
          <Route path="matakuliah/:id" element={<MataKuliahDetail />} />
          <Route path="rencanastudi" element={<RencanaStudi />} />
        </Route>
      </Route>

      <Route path="*" element={<Navigate to="/login" />} />
    </Routes>
  );
};

const App = () => {
  return (
    <AuthProvider>
      <Router>
        <Toaster
          position="top-right"
          toastOptions={{
            duration: 3000,
            style: {
              background: "#fff",
              color: "#333",
              fontWeight: "500",
              borderRadius: "8px",
              boxShadow: "0 4px 12px rgba(0, 0, 0, 0.1)",
            },
          }}
        />
        <AppRoutes />
      </Router>
    </AuthProvider>
  );
};

export default App;
```



Admin Panel

Dashboard

Mahasiswa

Dosen

Mata Kuliah

Rencana Studi

UDINUS HOME

Welcome, Admin 1 (admin)

✓ Login berhasil!



Dashboard Admin

Selamat datang, Admin 1! Berikut informasi untuk role: admin.



Jumlah Mahasiswa

20



Jumlah Dosen

5



Jumlah Mata Kuliah

8



Admin Panel

Dashboard

Mahasiswa

Dosen

Mata Kuliah

Rencana Studi

UDINUS HOME

Welcome, Admin 1 (admin)



Daftar Mahasiswa

Cari nama atau NIM...

+ Tambah Mahasiswa

NIM	Nama	Email	Jurusan	Maksimal SKS	Total SKS	Aksi
M001	Andi	andi@gmail.com	Informatika	18	0	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
M002	Budi	budi@gmail.com	Sistem Informasi	20	0	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
M003	Citra	citra@gmail.com	Informatika	22	3	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
M004	Dina	dina@gmail.com	Sistem Informasi	18	0	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
M005	Eko	eko@gmail.com	Informatika	24	3	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

Menampilkan 1 - 5 dari 20 mahasiswa

Previous

Next

Halaman 1 dari 4

Admin Panel



UDINUS HOME

Welcome, Admin 1 (admin)

[Dashboard](#)[Mahasiswa](#)[Dosen](#)[Mata Kuliah](#)[Rencana Studi](#)

Daftar Dosen

Cari nama, NIP, atau email...

+ Tambah Dosen

NIP	Nama	Email	Maksimal SKS	Total SKS	Aksi
DSN01	Slamet	slamet@gmail.com	12	0	Detail Edit Hapus
DSN02	Tini	tini@gmail.com	10	3	Detail Edit Hapus
DSN03	Udin	udin@gmail.com	14	0	Detail Edit Hapus
DSN04	Vina	vina@gmail.com	16	0	Detail Edit Hapus
DSN05	Wawan	wawan@gmail.com	8	0	Detail Edit Hapus

Menampilkan 1 - 5 dari 5 dosen

[Sebelumnya](#)[Selanjutnya](#)

Admin Panel

Dashboard

Mahasiswa

Dosen

Mata Kuliah

Rencana Studi

UDINUS HOME

Welcome, Admin 1 (admin)



Daftar Mata Kuliah

Cari nama atau kode...

+ Tambah Mata Kuliah

Kode	Nama	SKS	Aksi
MK01	Algoritma dan Pemrograman	3	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
MK02	Basis Data	3	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
MK03	Jaringan Komputer	2	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
MK04	Pemrograman Sisi Klien	3	<button>Detail</button> <button>Edit</button> <button>Hapus</button>
MK05	Pemrograman Sisi Server	3	<button>Detail</button> <button>Edit</button> <button>Hapus</button>

Menampilkan 1 - 5 dari 8 dosen

Previous

Next

Halaman 1 dari 2

Admin Panel

- Dashboard
- Mahasiswa
- Dosen
- Mata Kuliah
- Rencana Studi

UDINUS HOME

Welcome, Admin 1 (admin)



Daftar Rencana Studi

Cari nama atau kode...

+ Tambah Kelas

Mata Kuliah

Algoritma dan Pemrograman

Dosen Pengampu

Tini

-- Ganti Mata Kuliah --

Simpan Mata Kuliah

-- Ganti Dosen --

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
1	Citra	M003	3	<button>Hapus</button>
2	Eko	M005	3	<button>Hapus</button>
3	Fajar	M006	3	<button>Hapus</button>

-- Pilih Mahasiswa --

Tambah Mahasiswa

Admin Panel



UDINUS HOME

Welcome, Admin 1 (admin)

Dashboard

Mahasiswa

Dosen

Mata Kuliah

Rencana Studi

Daftar Rencana Studi

Cari nama atau kode...

+ Tambah Kelas

Mata Kuliah

Algoritma dan Pemrograman

Dosen Pengampu

Tini

-- Ganti Mata Kuliah --

Tambah Kelas Baru

Mata Kuliah

-- Pilih Mata Kuliah --

No

Nama

1 Citra

2 Eko

3 Fajar

Dosen Pengampu

-- Pilih Dosen --

-- Pilih Mahasiswa --

Batal

Simpan

Simpan Dosen

Total SKS

Aksi

3

Hapus

3

Hapus

3

Hapus

[Apply Activity | Me...](#)[Digital Talent Scho...](#)[\(154\) Belajar Java...](#)[Home | AWS Back...](#)[Telegram](#)[Evafadhillahulia at...](#)[Apple Developer](#)[iOS Academy - Yo...](#)[WhatsApp](#)[Learn Swift Progra...](#)[»](#)[All Bookmarks](#)

Admin Panel

[Dashboard](#)[Mahasiswa](#)[Dosen](#)[Mata Kuliah](#)[Rencana Studi](#)

UDINUS HOME

Welcome, Admin 1 (admin)



Daftar Rencana Studi

[+ Tambah Kelas](#)

Mata Kuliah

Algoritma dan Pemrograman

Dosen Pengampu

Tini

[-- Ganti Mata Kuliah --](#)

Tambah Kelas Baru

No	Nama
1	Citra
2	Eko
3	Fajar

Mata Kuliah

Basis Data

Dosen Pengampu

Slamet

[Batal](#)[Simpan](#)[-- Pilih Mahasiswa --](#)

Total SKS	Aksi
3	Hapus
3	Hapus
3	Hapus

3

Hapus

3

Hapus

3

Hapus

Admin Panel

Dashboard

Mahasiswa

Dosen

Mata Kuliah

Rencana Studi

UDINUS HOME

Welcome, Admin 1 (admin)

Kelas ditambahkan

Rencana dan Pengembangan

Dosen Pengampu

Tini

-- Ganti Mata Kuliah --

Simpan Mata Kuliah

-- Ganti Dosen --

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
1	Citra	M003	3	<button>Hapus</button>
2	Eko	M005	3	<button>Hapus</button>
3	Fajar	M006	3	<button>Hapus</button>

-- Pilih Mahasiswa --

Tambah Mahasiswa

Hapus

Hapus

Hapus

Mata Kuliah

Basis Data

Dosen Pengampu

Hapus Kelas

Slamet

-- Ganti Mata Kuliah --

Simpan Mata Kuliah

-- Ganti Dosen --

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
Belum ada mahasiswa.				

-- Pilih Mahasiswa --

Tambah Mahasiswa

Admin Panel

Dashboard

Mahasiswa

Dosen

Mata Kuliah

Rencana Studi

UDINUS HOME

Welcome, Admin 1 (admin)

Mata kuliah berhasil diperbarui

Rencana dan Pengembangan

Dosen Pengampu

Tini

-- Ganti Mata Kuliah --

Simpan Mata Kuliah

-- Ganti Dosen --

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
1	Citra	M003	3	<button>Hapus</button>
2	Eko	M005	3	<button>Hapus</button>
3	Fajar	M006	3	<button>Hapus</button>

-- Pilih Mahasiswa --

Tambah Mahasiswa

Hapus Kelas

Mata Kuliah

Jaringan Komputer

Dosen Pengampu

Slamet

Jaringan Komputer

Simpan Mata Kuliah

-- Ganti Dosen --

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
Belum ada mahasiswa.				
	-- Pilih Mahasiswa --		Tambah Mahasiswa	

Admin Panel

Dashboard

Mahasiswa

Dosen

Mata Kuliah

Rencana Studi

UDINUS HOME

Welcome, Admin 1 (admin)

Dosen diperbarui

Rencana dan Pengembangan

Dosen Pengampu

Tini

-- Ganti Mata Kuliah --

Simpan Mata Kuliah

-- Ganti Dosen --

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
1	Citra	M003	3	<button>Hapus</button>
2	Eko	M005	3	<button>Hapus</button>
3	Fajar	M006	3	<button>Hapus</button>

-- Pilih Mahasiswa --

Tambah Mahasiswa

Mata Kuliah

Jaringan Komputer

Hapus Kelas

Dosen Pengampu

Wawan

Jaringan Komputer

Simpan Mata Kuliah

Wawan

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
Belum ada mahasiswa.				
-- Pilih Mahasiswa --	Tambah Mahasiswa			

Admin Panel

Dashboard

Mahasiswa

Dosen

Mata Kuliah

Rencana Studi

UDINUS HOME

Welcome, Admin 1 (admin)

Mahasiswa ditambahkan

Rencana dan Pengembangan

Dosen Pengampu

Tini

-- Ganti Mata Kuliah --

Simpan Mata Kuliah

-- Ganti Dosen --

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
1	Citra	M003	3	<button>Hapus</button>
2	Eko	M005	3	<button>Hapus</button>
3	Fajar	M006	3	<button>Hapus</button>

-- Pilih Mahasiswa --

Tambah Mahasiswa

Mata Kuliah

Jaringan Komputer

Dosen Pengampu

Wawan

Jaringan Komputer

Simpan Mata Kuliah

Wawan

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
1	Andi	M001	2	<button>Hapus</button>

Admin Panel

Dashboard

Mahasiswa

Dosen

Mata Kuliah

Rencana Studi

UDINUS HOME

Welcome, Admin 1 (admin)

Mahasiswa ditambahkan

Rencana dan Pengembangan

Dosen Pengampu

Tini

-- Ganti Mata Kuliah --

Simpan Mata Kuliah

-- Ganti Dosen --

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
1	Citra	M003	3	<button>Hapus</button>
2	Eko	M005	3	<button>Hapus</button>
3	Fajar	M006	3	<button>Hapus</button>

-- Pilih Mahasiswa --

Tambah Mahasiswa

Mata Kuliah

Jaringan Komputer

Dosen Pengampu

Wawan

Jaringan Komputer

Simpan Mata Kuliah

Wawan

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
1	Andi	M001	2	<button>Hapus</button>
2	Budi	M002	2	<button>Hapus</button>

-- Pilih Mahasiswa --

Tambah Mahasiswa

Admin Panel

Dashboard

Mahasiswa

Dosen

Mata Kuliah

Rencana Studi

UDINUS HOME

Welcome, Admin 1 (admin)

Mahasiswa ditambahkan

Tini

-- Ganti Mata Kuliah --

Simpan Mata Kuliah

-- Ganti Dosen --

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
1	Citra	M003	5	<button>Hapus</button>
2	Eko	M005	3	<button>Hapus</button>
3	Fajar	M006	3	<button>Hapus</button>

-- Pilih Mahasiswa --

Tambah Mahasiswa

Mata Kuliah

Jaringan Komputer

Dosen Pengampu

Wawan

Jaringan Komputer

Simpan Mata Kuliah

Wawan

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
1	Andi	M001	2	<button>Hapus</button>
2	Budi	M002	2	<button>Hapus</button>
3	Citra	M003	5	<button>Hapus</button>

-- Pilih Mahasiswa --

Tambah Mahasiswa

Admin Panel

Dashboard

Mahasiswa

Dosen

Mata Kuliah

Rencana Studi

UDINUS HOME

Welcome, Admin 1 (admin)

Mahasiswa dihapus

Rencana dan Pengembangan

Dosen Pengampu

Tini

-- Ganti Mata Kuliah --

Simpan Mata Kuliah

-- Ganti Dosen --

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
1	Citra	M003	3	<button>Hapus</button>
2	Eko	M005	3	<button>Hapus</button>
3	Fajar	M006	3	<button>Hapus</button>

-- Pilih Mahasiswa --

Tambah Mahasiswa

Mata Kuliah

Jaringan Komputer

Hapus Kelas

Dosen Pengampu

Wawan

Jaringan Komputer

Simpan Mata Kuliah

Wawan

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
Belum ada mahasiswa.				
-- Pilih Mahasiswa --	Tambah Mahasiswa			

Dashboard Pemrograman Sisi Klien UDINUS POLKE localhost:3000/dashboard/rencanastudi School

Apply Activity | Me... Digital Talent Scho... (154) Belajar Java... Home | AWS Back... Telegram Evafadhillahulia at... Apple Developer iOS Academy - Yo... WhatsApp Learn Swift Progra... All Bookmarks

Admin Panel

UDINUS HOME

Welcome, Admin 1 (admin)

Dosen Pengampu
Tini

-- Ganti Mata Kuliah -- Simpan Mata Kuliah -- Ganti Dosen -- Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
1	Citra	M003	3	Hapus
2	Eko		3	Hapus
3	Fajar		3	Hapus

-- Pilih Mahasiswa --

Mata Kuliah
Jaringan Komputer

Dosen Pengampu
Wawan

Jaringan Komputer Simpan Mata Kuliah Wawan Simpan Dosen

Konfirmasi Hapus

Yakin ingin menghapus Kelas Jaringan Komputer?

Ya, Hapus! Batal Hapus Kelas

Belum ada mahasiswa.

-- Pilih Mahasiswa -- Tambah Mahasiswa

© 2025 Admin Dashboard. All rights reserved.

Admin Panel

Dashboard

Mahasiswa

Dosen

Mata Kuliah

Rencana Studi

UDINUS HOME

Welcome, Admin 1 (admin)

Kelas berhasil dihapus

Daftar Rencana Studi

Cari nama atau kode...

+ Tambah Kelas

Mata Kuliah

Algoritma dan Pemrograman

Dosen Pengampu

Tini

-- Ganti Mata Kuliah --

Simpan Mata Kuliah

-- Ganti Dosen --

Simpan Dosen

No	Nama	NIM	Total SKS	Aksi
1	Citra	M003	3	<button>Hapus</button>
2	Eko	M005	3	<button>Hapus</button>
3	Fajar	M006	3	<button>Hapus</button>

-- Pilih Mahasiswa --

Tambah Mahasiswa