

## /src/Pages/Layouts/Utils/Apis/MahasiswaApi.js

```
Pertemuan-3 > src > Pages > Layouts > Utils > Apis > JS MahasiswaApi.js > ...
1 import axios from "@/Pages/Layouts/Utils/AxiosInstance";
2
3 // Ambil semua mahasiswa
4 export const getAllMahasiswa = () => axios.get("/mahasiswa");
5
6 // Ambil 1 mahasiswa
7 export const getMahasiswa = (id) => axios.get(`/mahasiswa/${id}`);
8
9 // Tambah mahasiswa
10 export const storeMahasiswa = (data) => axios.post("/mahasiswa", data);
11
12 // Update mahasiswa
13 export const updateMahasiswa = (id, data) => axios.put(`/mahasiswa/${id}`, data);
14
15 // Hapus mahasiswa
16 export const deleteMahasiswa = (id) => axios.delete(`/mahasiswa/${id}`);
```

## /src/Pages/Layouts/Utils/Apis/useMahasiswa.js

```
Pertemuan-3 > src > Pages > Layouts > Utils > Apis > JS useMahasiswa.js > ...
1 import { useQuery, useMutation, useQueryClient } from "@tanstack/react-query";
2 import {
3   getAllMahasiswa,
4   storeMahasiswa,
5   updateMahasiswa,
6   deleteMahasiswa,
7 } from "@/Pages/Layouts/Utils/Apis/MahasiswaApi";
8
9 /* ===== GET ===== */
10 export const useMahasiswa = (query = {}) =>
11   useQuery({
12     queryKey: ["mahasiswa", query],
13     queryFn: () => getAllMahasiswa(query),
14     select: (res) => ({
15       data: res.data || [],
16       total: parseInt(res.headers["x-total-count"] || "0", 10),
17     }),
18     keepPreviousData: true,
19   });
20
21 /* ===== CREATE ===== */
22 export const useStoreMahasiswa = () => {
23   const queryClient = useQueryClient();
24   return useMutation({
25     mutationFn: storeMahasiswa,
26     onSuccess: () => {
27       queryClient.invalidateQueries(["mahasiswa"]);
28     },
29   });
30 };
31
32 /* ===== UPDATE ===== */
33 export const useUpdateMahasiswa = () => {
34   const queryClient = useQueryClient();
35   return useMutation({
36     mutationFn: ({ id, data }) => updateMahasiswa(id, data),
37     onSuccess: () => {
38       queryClient.invalidateQueries(["mahasiswa"]);
39     },
40   });
41 };
42
43 /* ===== DELETE ===== */
44 export const useDeleteMahasiswa = () => {
45   const queryClient = useQueryClient();
46   return useMutation({
47     mutationFn: deleteMahasiswa,
48     onSuccess: () => {
49       queryClient.invalidateQueries(["mahasiswa"]);
50     },
51   });
52 };
53
```

## /src/Pages/Admin/Mahasiswa.jsx

```
Pertemuan-3 > src > Pages > Admin > Mahasiswa.jsx > [o] Mahasiswa
  1 import React, { useEffect, useState } from "react";
  2 import Card from "@/Pages/Layouts/Components/Card";
  3 import Heading from "@/Pages/Layouts/Components/Heading";
  4 import Button from "@/Pages/Layouts/Components/Button";
  5 import { useNavigate } from "react-router-dom";
  6 import { useAuthStateContext } from "@/Pages/Layouts/Utils/Context/AuthContext";
  7
  8 import TableMahasiswa from "./TableMahasiswa";
  9 import ModalMahasiswa from "./ModalMahasiswa";
 10
 11 import { toastError } from "@/Pages/Layouts/Utils/Helpers/ToastHelpers";
 12 import { confirmDelete, confirmUpdate } from "@/Pages/Layouts/Utils/Helpers/SwalHelpers";
 13
 14 import {
 15   useMahasiswa,
 16   useStoreMahasiswa,
 17   useUpdateMahasiswa,
 18   useDeleteMahasiswa,
 19 } from "@/Pages/Layouts/Utils/Hooks/useMahasiswa";
 20
 21 import { getAllKelas } from "@/Pages/Layouts/Utils/apis/KelasApi";
 22 import { getAllMataKuliah } from "@/Pages/Layouts/Utils/apis/MataKuliahApi";
 23
 24 const Mahasiswa = () => {
 25   const navigate = useNavigate();
 26   const { user } = useAuthStateContext();
 27
 28   // ===== STATE PAGINATION & FILTER =====
 29   const [page, setPage] = useState(1);
 30   const [perPage] = useState(6);
 31   const [sort, setSort] = useState("name");
 32   const [order, setOrder] = useState("asc");
 33   const [search, setSearch] = useState("");
 34 }
```

```
// ===== FETCH MAHASISWA =====
const {
  data: mahasiswa = [],
  isLoading: isLoadingMahasiswa,
} = useMahasiswa();

const sortedMahasiswa = [...mahasiswa].sort((a, b) => {
  const valA = a[sort];
  const valB = b[sort];

  // === KHUSUS NIM (M001, M021, dll) ===
  if (sort === "nim") {
    const numA = parseInt(valA.replace(/\D/g, ""), 10);
    const numB = parseInt(valB.replace(/\D/g, ""), 10);

    return order === "asc" ? numA - numB : numB - numA;
  }

  // === STRING NORMAL (name) ===
  if (typeof valA === "string" && typeof valB === "string") {
    return order === "asc"
      ? valA.localeCompare(valB)
      : valB.localeCompare(valA);
  }

  // === NUMBER (max_sks) ===
  if (typeof valA === "number" && typeof valB === "number") {
    return order === "asc"
      ? valA - valB
      : valB - valA;
  }

  return 0;
});

const totalPages = Math.ceil(mahasiswa.length / perPage);
const startIndex = (page - 1) * perPage;
const endIndex = startIndex + perPage;
const paginatedMahasiswa = sortedMahasiswa.slice(startIndex, endIndex);
```

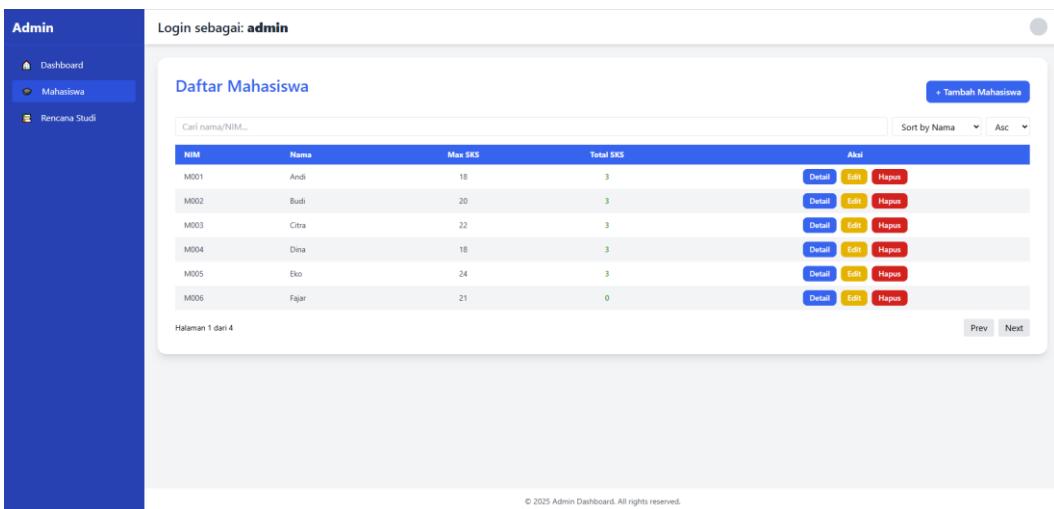
```
205     {/* SEARCH & SORT */}
206     <div className="flex flex-wrap gap-2 mb-4">
207         <input
208             type="text"
209             placeholder="Cari nama/NIM..."
210             className="border px-3 py-1 rounded flex-grow"
211             value={search}
212             onChange={(e) => {
213                 setSearch(e.target.value);
214                 setPage(1);
215             }}
216         />
217
218         <select
219             value={sort}
220             onChange={(e) => {
221                 setSort(e.target.value);
222                 setPage(1);
223             }}
224             className="border px-3 py-1 rounded"
225         >
226             <option value="name">Sort by Nama</option>
227             <option value="nim">Sort by NIM</option>
228             <option value="max_sks">Sort by Max SKS</option>
229         </select>
230
231         <select
232             value={order}
233             onChange={(e) => {
234                 setOrder(e.target.value);
235                 setPage(1);
236             }}
237             className="border px-3 py-1 rounded"
238         >
239             <option value="asc">Asc</option>
240             <option value="desc">Desc</option>
241         </select>
242     </div>
```

```
189 // ----- PAGINATION -----
190 const handlePrev = () => setPage((p) => Math.max(p - 1, 1));
191 const handleNext = () => setPage((p) => Math.min(p + 1, totalPages));
192
```

```
255     {/* PAGINATION */}
256     <div className="flex justify-between items-center mt-4">
257         <p className="text-sm">
258             Halaman {page} dari {totalPages}
259         </p>
260
261         <div className="flex gap-2">
262             <button
263                 className="px-3 py-1 bg-gray-200 rounded"
264                 onClick={handlePrev}
265                 disabled={page === 1}
266             >
267                 Prev
268             </button>
269
270             <button
271                 className="px-3 py-1 bg-gray-200 rounded"
272                 onClick={handleNext}
273                 disabled={page === totalPages}
274             >
275                 Next
276             </button>
277         </div>
278     </div>
```

Tampilan :

Sort by nama, asc

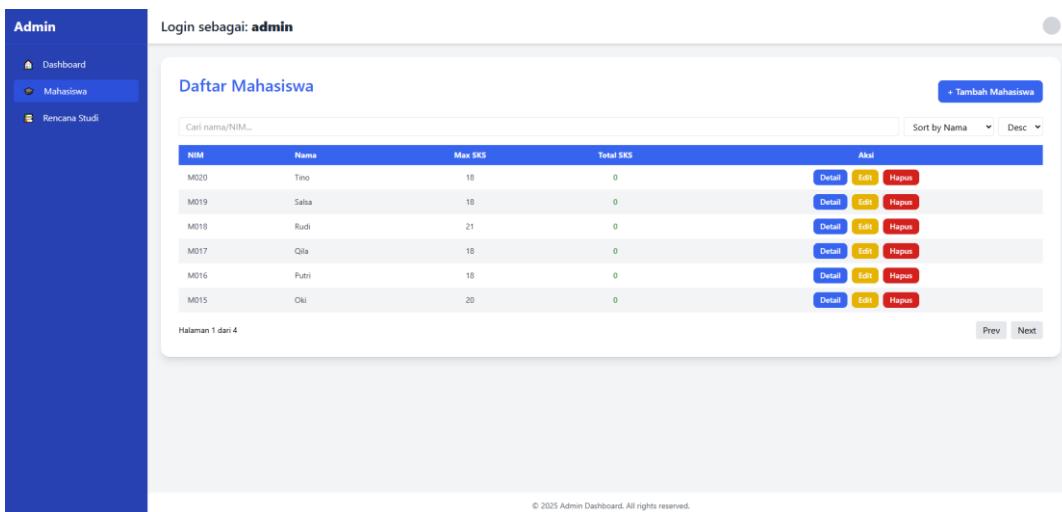


The screenshot shows the 'Daftar Mahasiswa' (Student List) page. The sidebar on the left is titled 'Admin' and includes 'Dashboard', 'Mahasiswa' (which is highlighted in blue), and 'Rencana Studi'. The main content area has a header 'Login sebagai: admin' and a search bar 'Cari nama/NIM...'. A sorting dropdown 'Sort by Nama' is set to 'Asc'. The table lists student data:

NIM	Nama	Max SKS	Total SKS	Aksi
M001	Andi	18	3	[Detail] [Edit] [Hapus]
M002	Budi	20	3	[Detail] [Edit] [Hapus]
M003	Citra	22	3	[Detail] [Edit] [Hapus]
M004	Dina	18	3	[Detail] [Edit] [Hapus]
M005	Eko	24	3	[Detail] [Edit] [Hapus]
M006	Fajar	21	0	[Detail] [Edit] [Hapus]

At the bottom, it says 'Halaman 1 dari 4' and has 'Prev' and 'Next' buttons.

Sort by nama , desc

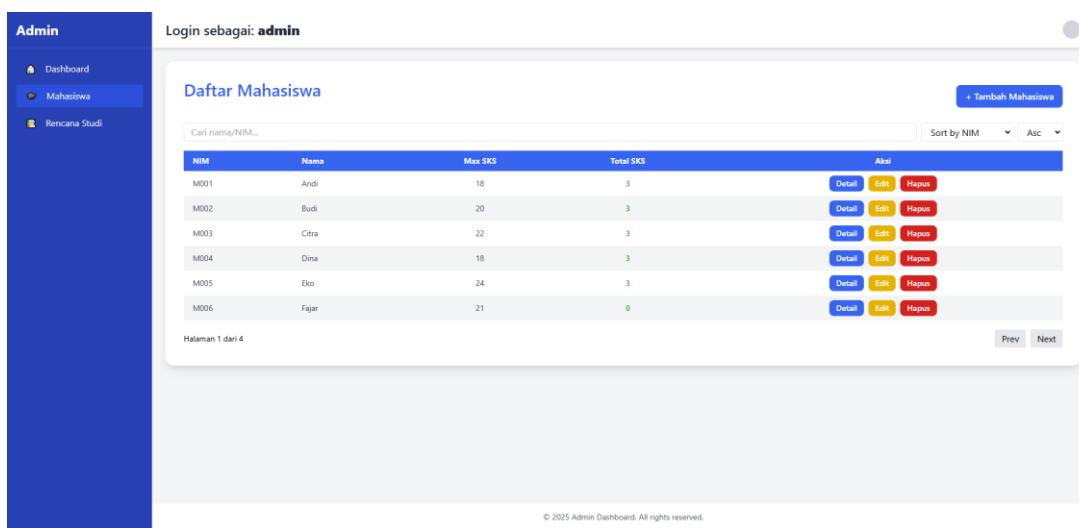


The screenshot shows the 'Daftar Mahasiswa' (Student List) page. The sidebar on the left is titled 'Admin' and includes 'Dashboard', 'Mahasiswa' (highlighted in blue), and 'Rencana Studi'. The main content area has a header 'Login sebagai: admin' and a search bar 'Cari nama/NIM...'. A sorting dropdown 'Sort by Nama' is set to 'Desc'. The table lists student data:

NIM	Nama	Max SKS	Total SKS	Aksi
M020	Tino	18	0	[Detail] [Edit] [Hapus]
M019	Salsa	18	0	[Detail] [Edit] [Hapus]
M018	Rudi	21	0	[Detail] [Edit] [Hapus]
M017	Qila	18	0	[Detail] [Edit] [Hapus]
M016	Putri	18	0	[Detail] [Edit] [Hapus]
M015	Oki	20	0	[Detail] [Edit] [Hapus]

At the bottom, it says 'Halaman 1 dari 4' and has 'Prev' and 'Next' buttons.

Sort by nim, asc



The screenshot shows the 'Daftar Mahasiswa' (Student List) page. The sidebar on the left is titled 'Admin' and includes 'Dashboard', 'Mahasiswa' (highlighted in blue), and 'Rencana Studi'. The main content area has a header 'Login sebagai: admin' and a search bar 'Cari nama/NIM...'. A sorting dropdown 'Sort by NIM' is set to 'Asc'. The table lists student data:

NIM	Nama	Max SKS	Total SKS	Aksi
M001	Andi	18	3	[Detail] [Edit] [Hapus]
M002	Budi	20	3	[Detail] [Edit] [Hapus]
M003	Citra	22	3	[Detail] [Edit] [Hapus]
M004	Dina	18	3	[Detail] [Edit] [Hapus]
M005	Eko	24	3	[Detail] [Edit] [Hapus]
M006	Fajar	21	0	[Detail] [Edit] [Hapus]

At the bottom, it says 'Halaman 1 dari 4' and has 'Prev' and 'Next' buttons.

Sort by nim, desc

The screenshot shows the Admin Dashboard with a sidebar on the left containing 'Dashboard', 'Mahasiswa' (which is highlighted in blue), and 'Rencana Studi'. The main area is titled 'Daftar Mahasiswa' and shows a table of student data. The table has columns: NIM, Nama, Max SKS, Total SKS, and Aksi. The data is sorted by NIM in descending order, with rows M020 (Tino), M019 (Salsa), M018 (Rudi), M017 (Qila), M016 (Putri), and M015 (Oki). Each row includes 'Detail', 'Edit', and 'Hapus' buttons. A search bar at the top says 'Cari nama/NIM...'. A dropdown menu at the top right says 'Sort by NIM' with 'Desc' selected. Navigation buttons 'Prev' and 'Next' are at the bottom.

NIM	Nama	Max SKS	Total SKS	Aksi
M020	Tino	18	0	[Detail] [Edit] [Hapus]
M019	Salsa	18	0	[Detail] [Edit] [Hapus]
M018	Rudi	21	0	[Detail] [Edit] [Hapus]
M017	Qila	18	0	[Detail] [Edit] [Hapus]
M016	Putri	18	0	[Detail] [Edit] [Hapus]
M015	Oki	20	0	[Detail] [Edit] [Hapus]

Sort by max sks, asc

The screenshot shows the Admin Dashboard with a sidebar on the left containing 'Dashboard', 'Mahasiswa' (highlighted in blue), and 'Rencana Studi'. The main area is titled 'Daftar Mahasiswa' and shows a table of student data. The table has columns: NIM, Nama, Max SKS, Total SKS, and Aksi. The data is sorted by Max SKS in ascending order, with rows M001 (Andi), M004 (Dina), M008 (Hani), M010 (Joko), M013 (Maya), and M016 (Putri). Each row includes 'Detail', 'Edit', and 'Hapus' buttons. A search bar at the top says 'Cari nama/NIM...'. A dropdown menu at the top right says 'Sort by Max SKS' with 'Asc' selected. Navigation buttons 'Prev' and 'Next' are at the bottom.

NIM	Nama	Max SKS	Total SKS	Aksi
M001	Andi	18	3	[Detail] [Edit] [Hapus]
M004	Dina	18	3	[Detail] [Edit] [Hapus]
M008	Hani	18	0	[Detail] [Edit] [Hapus]
M010	Joko	18	0	[Detail] [Edit] [Hapus]
M013	Maya	18	0	[Detail] [Edit] [Hapus]
M016	Putri	18	0	[Detail] [Edit] [Hapus]

Sort by max sks, desc

The screenshot shows the Admin Dashboard with a sidebar on the left containing 'Dashboard', 'Mahasiswa' (highlighted in blue), and 'Rencana Studi'. The main area is titled 'Daftar Mahasiswa' and shows a table of student data. The table has columns: NIM, Nama, Max SKS, Total SKS, and Aksi. The data is sorted by Max SKS in descending order, with rows M005 (Elo), M011 (Kiki), M003 (Citra), M014 (Nina), M006 (Fajar), and M018 (Rudi). Each row includes 'Detail', 'Edit', and 'Hapus' buttons. A search bar at the top says 'Cari nama/NIM...'. A dropdown menu at the top right says 'Sort by Max SKS' with 'Desc' selected. Navigation buttons 'Prev' and 'Next' are at the bottom.

NIM	Nama	Max SKS	Total SKS	Aksi
M005	Elo	24	3	[Detail] [Edit] [Hapus]
M011	Kiki	24	0	[Detail] [Edit] [Hapus]
M003	Citra	22	3	[Detail] [Edit] [Hapus]
M014	Nina	22	0	[Detail] [Edit] [Hapus]
M006	Fajar	21	0	[Detail] [Edit] [Hapus]
M018	Rudi	21	0	[Detail] [Edit] [Hapus]