

TUGAS 6

Source Code:

/src/Pages/Admin/Mahasiswa.jsx

```
Pertemuan-3 > src > Pages > Admin > Mahasiswa.jsx > ...
1 import React, { useState, useEffect } 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 { mahasiswaList } from "@/Data/Dummy";
6 import { useNavigate } from "react-router-dom";
7 import ModalMahasiswa from "./ModalMahasiswa";
8 import TableMahasiswa from "./TableMahasiswa";
9
10 const Mahasiswa = () => {
11   const navigate = useNavigate();
12
13   // Data Mahasiswa
14   const [mahasiswa, setMahasiswa] = useState([]);
15
16   // State Modal dan Form
17   const [isModalOpen, setIsModalOpen] = useState(false);
18   const [isEdit, setIsEdit] = useState(false);
19   const [form, setForm] = useState({ nim: "", nama: "" });
20   const [originalNim, setOriginalNim] = useState(null);
21
22   // useEffect
23   useEffect(() => {
24     setTimeout(() => fetchMahasiswa(), 500); // simulasi Loading
25   }, []);
26
27   const fetchMahasiswa = async () => {
28     // simulasi fetch API
29     setMahasiswa(mahasiswaList);
30   };
31
32   // Handle input form
33   const handleChange = (e) => {
34     setForm({ ...form, [e.target.name]: e.target.value });
35   };
36
37   // Tambah data mahasiswa baru
38   const addMahasiswa = (newData) => {
39     setMahasiswa([...mahasiswa, newData]);
40   };
41
```

Pertemuan-3 > src > Pages > Admin >  Mahasiswa.jsx > ...

```
10  const Mahasiswa = () => {
11
12      // Update data mahasiswa
13      const updateMahasiswa = (nim, newData) => {
14          const updated = mahasiswa.map(mhs) =>
15              mhs.nim === nim ? { ...mhs, ...newData } : mhs
16          );
17          setMahasiswa(updated);
18      };
19
20      // Hapus data mahasiswa
21      const deleteMahasiswa = (nim) => {
22          if (confirm("Yakin ingin hapus data ini?")) {
23              const filtered = mahasiswa.filter((mhs) => mhs.nim !== nim);
24              setMahasiswa(filtered);
25              alert("Data berhasil dihapus!");
26          }
27      };
28
29      // Buka modal untuk tambah
30      const handleAdd = () => {
31          setForm({ nim: "", nama: "" });
32          setIsEdit(false);
33          setIsModalOpen(true);
34      };
35
36      // Buka modal untuk edit
37      const handleEdit = (mhs) => {
38          setForm(mhs);
39          setOriginalNim(mhs.nim);
40          setIsEdit(true);
41          setIsModalOpen(true);
42      };
43
44      // Submit form
45      const handleSubmit = (e) => {
46          e.preventDefault();
47
48          if (!form.nim.trim() || !form.nama.trim()) {
49              alert("NIM dan Nama wajib diisi!");
50              return;
51          }
52      };
53
54      // Tambahkan data baru
55      const handleNew = () => {
56          const newMhs = {
57              nim: form.nim,
58              nama: form.nama
59          };
60
61          const updated = [...mahasiswa, newMhs];
62
63          setMahasiswa(updated);
64          setForm({ nim: "", nama: "" });
65          setIsEdit(false);
66          setIsModalOpen(false);
67      };
68
69      // Simpan perubahan
70      const handleSave = () => {
71          const updated = mahasiswa.map(mhs) =>
72              mhs.nim === originalNim ? { ...mhs, ...form } : mhs
73
74          setMahasiswa(updated);
75          setForm({ nim: "", nama: "" });
76          setIsEdit(false);
77          setIsModalOpen(false);
78      };
79
80      // Klik tombol cancel
81      const handleCancel = () => {
82          setForm({ nim: "", nama: "" });
83          setIsEdit(false);
84          setIsModalOpen(false);
85      };
86
87      // Klik tombol close
88      const handleClose = () => {
89          setForm({ nim: "", nama: "" });
90          setIsEdit(false);
91          setIsModalOpen(false);
92      };
93
94      // Klik tombol ok
95      const handleOk = () => {
96          handleSave();
97          handleClose();
98      };
99
100     // Klik tombol cancel
101     const handleCancelEdit = () => {
102         setForm({ nim: "", nama: "" });
103         setIsEdit(false);
104         setIsModalOpen(false);
105     };
106
107     // Klik tombol close
108     const handleCloseEdit = () => {
109         setForm({ nim: "", nama: "" });
110         setIsEdit(false);
111         setIsModalOpen(false);
112     };
113
114     // Klik tombol ok
115     const handleOkEdit = () => {
116         handleSave();
117         handleCloseEdit();
118     };
119
120     // Klik tombol refresh
121     const handleRefresh = () => {
122         setMahasiswa([]);
123         setForm({ nim: "", nama: "" });
124         setIsEdit(false);
125         setIsModalOpen(false);
126     };
127
128     // Klik tombol search
129     const handleSearch = (e) => {
130         e.preventDefault();
131
132         const query = e.target.value;
133
134         const filtered = mahasiswa.filter((mhs) => mhs.nama.toLowerCase().includes(query));
135
136         setMahasiswa(filtered);
137     };
138
139     // Klik tombol sort
140     const handleSort = (e) => {
141         e.preventDefault();
142
143         const direction = e.target.value;
144
145         const sorted = [...mahasiswa].sort((a, b) => {
146             if (a.nama < b.nama) {
147                 return -1;
148             } else if (a.nama > b.nama) {
149                 return 1;
150             } else {
151                 return 0;
152             }
153         });
154
155         setMahasiswa(sorted);
156     };
157
158     // Klik tombol filter
159     const handleFilter = (e) => {
160         e.preventDefault();
161
162         const checked = e.target.checked;
163
164         const filtered = checked ? [...mahasiswa] : [];
165
166         setMahasiswa(filtered);
167     };
168
169     // Klik tombol clear
170     const handleClear = () => {
171         setMahasiswa([]);
172     };
173
174     // Klik tombol print
175     const handlePrint = () => {
176         window.print();
177     };
178
179     // Klik tombol export
180     const handleExport = () => {
181         const csvContent = `nim,nama\n${mahasiswa.map((mhs) => `${mhs.nim},${mhs.nama}`).join("\n")}`;
182
183         const blob = new Blob([csvContent], { type: "text/csv" });
184
185         const url = URL.createObjectURL(blob);
186
187         const link = document.createElement("a");
188         link.href = url;
189         link.download = "data.csv";
190
191         link.click();
192     };
193
194     // Klik tombol import
195     const handleImport = () => {
196         const fileInput = document.querySelector("#file");
197
198         fileInput.addEventListener("change", (e) => {
199             const file = e.target.files[0];
200
201             const reader = new FileReader();
202
203             reader.onload = (e) => {
204                 const csvContent = e.target.result;
205
206                 const rows = csvContent.split("\n");
207
208                 const data = rows.map((row) => {
209                     const [nim, nama] = row.split(",");
210
211                     return { nim, nama };
212                 });
213
214                 const updated = [...mahasiswa, ...data];
215
216                 setMahasiswa(updated);
217             };
218
219             reader.readAsText(file);
220         });
221     };
222
223     // Klik tombol refresh
224     const handleRefresh = () => {
225         setMahasiswa([]);
226         setForm({ nim: "", nama: "" });
227         setIsEdit(false);
228         setIsModalOpen(false);
229     };
230
231     // Klik tombol search
232     const handleSearch = (e) => {
233         e.preventDefault();
234
235         const query = e.target.value;
236
237         const filtered = mahasiswa.filter((mhs) => mhs.nama.toLowerCase().includes(query));
238
239         setMahasiswa(filtered);
240     };
241
242     // Klik tombol sort
243     const handleSort = (e) => {
244         e.preventDefault();
245
246         const direction = e.target.value;
247
248         const sorted = [...mahasiswa].sort((a, b) => {
249             if (a.nama < b.nama) {
250                 return -1;
251             } else if (a.nama > b.nama) {
252                 return 1;
253             } else {
254                 return 0;
255             }
256         });
257
258         setMahasiswa(sorted);
259     };
260
261     // Klik tombol filter
262     const handleFilter = (e) => {
263         e.preventDefault();
264
265         const checked = e.target.checked;
266
267         const filtered = checked ? [...mahasiswa] : [];
268
269         setMahasiswa(filtered);
270     };
271
272     // Klik tombol clear
273     const handleClear = () => {
274         setMahasiswa([]);
275     };
276
277     // Klik tombol print
278     const handlePrint = () => {
279         window.print();
280     };
281
282     // Klik tombol export
283     const handleExport = () => {
284         const csvContent = `nim,nama\n${mahasiswa.map((mhs) => `${mhs.nim},${mhs.nama}`).join("\n")}`;
285
286         const blob = new Blob([csvContent], { type: "text/csv" });
287
288         const url = URL.createObjectURL(blob);
289
290         const link = document.createElement("a");
291         link.href = url;
292         link.download = "data.csv";
293
294         link.click();
295     };
296
297     // Klik tombol import
298     const handleImport = () => {
299         const fileInput = document.querySelector("#file");
300
301         fileInput.addEventListener("change", (e) => {
302             const file = e.target.files[0];
303
304             const reader = new FileReader();
305
306             reader.onload = (e) => {
307                 const csvContent = e.target.result;
308
309                 const rows = csvContent.split("\n");
310
311                 const data = rows.map((row) => {
312                     const [nim, nama] = row.split(",");
313
314                     return { nim, nama };
315                 });
316
317                 const updated = [...mahasiswa, ...data];
318
319                 setMahasiswa(updated);
320             };
321
322             reader.readAsText(file);
323         });
324     };
325
326     // Klik tombol refresh
327     const handleRefresh = () => {
328         setMahasiswa([]);
329         setForm({ nim: "", nama: "" });
330         setIsEdit(false);
331         setIsModalOpen(false);
332     };
333
334     // Klik tombol search
335     const handleSearch = (e) => {
336         e.preventDefault();
337
338         const query = e.target.value;
339
340         const filtered = mahasiswa.filter((mhs) => mhs.nama.toLowerCase().includes(query));
341
342         setMahasiswa(filtered);
343     };
344
345     // Klik tombol sort
346     const handleSort = (e) => {
347         e.preventDefault();
348
349         const direction = e.target.value;
350
351         const sorted = [...mahasiswa].sort((a, b) => {
352             if (a.nama < b.nama) {
353                 return -1;
354             } else if (a.nama > b.nama) {
355                 return 1;
356             } else {
357                 return 0;
358             }
359         });
360
361         setMahasiswa(sorted);
362     };
363
364     // Klik tombol filter
365     const handleFilter = (e) => {
366         e.preventDefault();
367
368         const checked = e.target.checked;
369
370         const filtered = checked ? [...mahasiswa] : [];
371
372         setMahasiswa(filtered);
373     };
374
375     // Klik tombol clear
376     const handleClear = () => {
377         setMahasiswa([]);
378     };
379
380     // Klik tombol print
381     const handlePrint = () => {
382         window.print();
383     };
384
385     // Klik tombol export
386     const handleExport = () => {
387         const csvContent = `nim,nama\n${mahasiswa.map((mhs) => `${mhs.nim},${mhs.nama}`).join("\n")}`;
388
389         const blob = new Blob([csvContent], { type: "text/csv" });
390
391         const url = URL.createObjectURL(blob);
392
393         const link = document.createElement("a");
394         link.href = url;
395         link.download = "data.csv";
396
397         link.click();
398     };
399
400     // Klik tombol import
401     const handleImport = () => {
402         const fileInput = document.querySelector("#file");
403
404         fileInput.addEventListener("change", (e) => {
405             const file = e.target.files[0];
406
407             const reader = new FileReader();
408
409             reader.onload = (e) => {
410                 const csvContent = e.target.result;
411
412                 const rows = csvContent.split("\n");
413
414                 const data = rows.map((row) => {
415                     const [nim, nama] = row.split(",");
416
417                     return { nim, nama };
418                 });
419
420                 const updated = [...mahasiswa, ...data];
421
422                 setMahasiswa(updated);
423             };
424
425             reader.readAsText(file);
426         });
427     };
428
429     // Klik tombol refresh
430     const handleRefresh = () => {
431         setMahasiswa([]);
432         setForm({ nim: "", nama: "" });
433         setIsEdit(false);
434         setIsModalOpen(false);
435     };
436
437     // Klik tombol search
438     const handleSearch = (e) => {
439         e.preventDefault();
440
441         const query = e.target.value;
442
443         const filtered = mahasiswa.filter((mhs) => mhs.nama.toLowerCase().includes(query));
444
445         setMahasiswa(filtered);
446     };
447
448     // Klik tombol sort
449     const handleSort = (e) => {
450         e.preventDefault();
451
452         const direction = e.target.value;
453
454         const sorted = [...mahasiswa].sort((a, b) => {
455             if (a.nama < b.nama) {
456                 return -1;
457             } else if (a.nama > b.nama) {
458                 return 1;
459             } else {
460                 return 0;
461             }
462         });
463
464         setMahasiswa(sorted);
465     };
466
467     // Klik tombol filter
468     const handleFilter = (e) => {
469         e.preventDefault();
470
471         const checked = e.target.checked;
472
473         const filtered = checked ? [...mahasiswa] : [];
474
475         setMahasiswa(filtered);
476     };
477
478     // Klik tombol clear
479     const handleClear = () => {
480         setMahasiswa([]);
481     };
482
483     // Klik tombol print
484     const handlePrint = () => {
485         window.print();
486     };
487
488     // Klik tombol export
489     const handleExport = () => {
490         const csvContent = `nim,nama\n${mahasiswa.map((mhs) => `${mhs.nim},${mhs.nama}`).join("\n")}`;
491
492         const blob = new Blob([csvContent], { type: "text/csv" });
493
494         const url = URL.createObjectURL(blob);
495
496         const link = document.createElement("a");
497         link.href = url;
498         link.download = "data.csv";
499
500         link.click();
501     };
502
503     // Klik tombol import
504     const handleImport = () => {
505         const fileInput = document.querySelector("#file");
506
507         fileInput.addEventListener("change", (e) => {
508             const file = e.target.files[0];
509
510             const reader = new FileReader();
511
512             reader.onload = (e) => {
513                 const csvContent = e.target.result;
514
515                 const rows = csvContent.split("\n");
516
517                 const data = rows.map((row) => {
518                     const [nim, nama] = row.split(",");
519
520                     return { nim, nama };
521                 });
522
523                 const updated = [...mahasiswa, ...data];
524
525                 setMahasiswa(updated);
526             };
527
528             reader.readAsText(file);
529         });
530     };
531
532     // Klik tombol refresh
533     const handleRefresh = () => {
534         setMahasiswa([]);
535         setForm({ nim: "", nama: "" });
536         setIsEdit(false);
537         setIsModalOpen(false);
538     };
539
540     // Klik tombol search
541     const handleSearch = (e) => {
542         e.preventDefault();
543
544         const query = e.target.value;
545
546         const filtered = mahasiswa.filter((mhs) => mhs.nama.toLowerCase().includes(query));
547
548         setMahasiswa(filtered);
549     };
550
551     // Klik tombol sort
552     const handleSort = (e) => {
553         e.preventDefault();
554
555         const direction = e.target.value;
556
557         const sorted = [...mahasiswa].sort((a, b) => {
558             if (a.nama < b.nama) {
559                 return -1;
560             } else if (a.nama > b.nama) {
561                 return 1;
562             } else {
563                 return 0;
564             }
565         });
566
567         setMahasiswa(sorted);
568     };
569
570     // Klik tombol filter
571     const handleFilter = (e) => {
572         e.preventDefault();
573
574         const checked = e.target.checked;
575
576         const filtered = checked ? [...mahasiswa] : [];
577
578         setMahasiswa(filtered);
579     };
580
581     // Klik tombol clear
582     const handleClear = () => {
583         setMahasiswa([]);
584     };
585
586     // Klik tombol print
587     const handlePrint = () => {
588         window.print();
589     };
590
591     // Klik tombol export
592     const handleExport = () => {
593         const csvContent = `nim,nama\n${mahasiswa.map((mhs) => `${mhs.nim},${mhs.nama}`).join("\n")}`;
594
595         const blob = new Blob([csvContent], { type: "text/csv" });
596
597         const url = URL.createObjectURL(blob);
598
599         const link = document.createElement("a");
600         link.href = url;
601         link.download = "data.csv";
602
603         link.click();
604     };
605
606     // Klik tombol import
607     const handleImport = () => {
608         const fileInput = document.querySelector("#file");
609
610         fileInput.addEventListener("change", (e) => {
611             const file = e.target.files[0];
612
613             const reader = new FileReader();
614
615             reader.onload = (e) => {
616                 const csvContent = e.target.result;
617
618                 const rows = csvContent.split("\n");
619
620                 const data = rows.map((row) => {
621                     const [nim, nama] = row.split(",");
622
623                     return { nim, nama };
624                 });
625
626                 const updated = [...mahasiswa, ...data];
627
628                 setMahasiswa(updated);
629             };
630
631             reader.readAsText(file);
632         });
633     };
634
635     // Klik tombol refresh
636     const handleRefresh = () => {
637         setMahasiswa([]);
638         setForm({ nim: "", nama: "" });
639         setIsEdit(false);
640         setIsModalOpen(false);
641     };
642
643     // Klik tombol search
644     const handleSearch = (e) => {
645         e.preventDefault();
646
647         const query = e.target.value;
648
649         const filtered = mahasiswa.filter((mhs) => mhs.nama.toLowerCase().includes(query));
650
651         setMahasiswa(filtered);
652     };
653
654     // Klik tombol sort
655     const handleSort = (e) => {
656         e.preventDefault();
657
658         const direction = e.target.value;
659
660         const sorted = [...mahasiswa].sort((a, b) => {
661             if (a.nama < b.nama) {
662                 return -1;
663             } else if (a.nama > b.nama) {
664                 return 1;
665             } else {
666                 return 0;
667             }
668         });
669
670         setMahasiswa(sorted);
671     };
672
673     // Klik tombol filter
674     const handleFilter = (e) => {
675         e.preventDefault();
676
677         const checked = e.target.checked;
678
679         const filtered = checked ? [...mahasiswa] : [];
680
681         setMahasiswa(filtered);
682     };
683
684     // Klik tombol clear
685     const handleClear = () => {
686         setMahasiswa([]);
687     };
688
689     // Klik tombol print
690     const handlePrint = () => {
691         window.print();
692     };
693
694     // Klik tombol export
695     const handleExport = () => {
696         const csvContent = `nim,nama\n${mahasiswa.map((mhs) => `${mhs.nim},${mhs.nama}`).join("\n")}`;
697
698         const blob = new Blob([csvContent], { type: "text/csv" });
699
700         const url = URL.createObjectURL(blob);
701
702         const link = document.createElement("a");
703         link.href = url;
704         link.download = "data.csv";
705
706         link.click();
707     };
708
709     // Klik tombol import
710     const handleImport = () => {
711         const fileInput = document.querySelector("#file");
712
713         fileInput.addEventListener("change", (e) => {
714             const file = e.target.files[0];
715
716             const reader = new FileReader();
717
718             reader.onload = (e) => {
719                 const csvContent = e.target.result;
720
721                 const rows = csvContent.split("\n");
722
723                 const data = rows.map((row) => {
724                     const [nim, nama] = row.split(",");
725
726                     return { nim, nama };
727                 });
728
729                 const updated = [...mahasiswa, ...data];
730
731                 setMahasiswa(updated);
732             };
733
734             reader.readAsText(file);
735         });
736     };
737
738     // Klik tombol refresh
739     const handleRefresh = () => {
740         setMahasiswa([]);
741         setForm({ nim: "", nama: "" });
742         setIsEdit(false);
743         setIsModalOpen(false);
744     };
745
746     // Klik tombol search
747     const handleSearch = (e) => {
748         e.preventDefault();
749
750         const query = e.target.value;
751
752         const filtered = mahasiswa.filter((mhs) => mhs.nama.toLowerCase().includes(query));
753
754         setMahasiswa(filtered);
755     };
756
757     // Klik tombol sort
758     const handleSort = (e) => {
759         e.preventDefault();
760
761         const direction = e.target.value;
762
763         const sorted = [...mahasiswa].sort((a, b) => {
764             if (a.nama < b.nama) {
765                 return -1;
766             } else if (a.nama > b.nama) {
767                 return 1;
768             } else {
769                 return 0;
770             }
771         });
772
773         setMahasiswa(sorted);
774     };
775
776     // Klik tombol filter
777     const handleFilter = (e) => {
778         e.preventDefault();
779
780         const checked = e.target.checked;
781
782         const filtered = checked ? [...mahasiswa] : [];
783
784         setMahasiswa(filtered);
785     };
786
787     // Klik tombol clear
788     const handleClear = () => {
789         setMahasiswa([]);
790     };
791
792     // Klik tombol print
793     const handlePrint = () => {
794         window.print();
795     };
796
797     // Klik tombol export
798     const handleExport = () => {
799         const csvContent = `nim,nama\n${mahasiswa.map((mhs) => `${mhs.nim},${mhs.nama}`).join("\n")}`;
800
801         const blob = new Blob([csvContent], { type: "text/csv" });
802
803         const url = URL.createObjectURL(blob);
804
805         const link = document.createElement("a");
806         link.href = url;
807         link.download = "data.csv";
808
809         link.click();
810     };
811
812     // Klik tombol import
813     const handleImport = () => {
814         const fileInput = document.querySelector("#file");
815
816         fileInput.addEventListener("change", (e) => {
817             const file = e.target.files[0];
818
819             const reader = new FileReader();
820
821             reader.onload = (e) => {
822                 const csvContent = e.target.result;
823
824                 const rows = csvContent.split("\n");
825
826                 const data = rows.map((row) => {
827                     const [nim, nama] = row.split(",");
828
829                     return { nim, nama };
830                 });
831
832                 const updated = [...mahasiswa, ...data];
833
834                 setMahasiswa(updated);
835             };
836
837             reader.readAsText(file);
838         });
839     };
840
841     // Klik tombol refresh
842     const handleRefresh = () => {
843         setMahasiswa([]);
844         setForm({ nim: "", nama: "" });
845         setIsEdit(false);
846         setIsModalOpen(false);
847     };
848
849     // Klik tombol search
850     const handleSearch = (e) => {
851         e.preventDefault();
852
853         const query = e.target.value;
854
855         const filtered = mahasiswa.filter((mhs) => mhs.nama.toLowerCase().includes(query));
856
857         setMahasiswa(filtered);
858     };
859
860     // Klik tombol sort
861     const handleSort = (e) => {
862         e.preventDefault();
863
864         const direction = e.target.value;
865
866         const sorted = [...mahasiswa].sort((a, b) => {
867             if (a.nama < b.nama) {
868                 return -1;
869             } else if (a.nama > b.nama) {
870                 return 1;
871             } else {
872                 return 0;
873             }
874         });
875
876         setMahasiswa(sorted);
877     };
878
879     // Klik tombol filter
880     const handleFilter = (e) => {
881         e.preventDefault();
882
883         const checked = e.target.checked;
884
885         const filtered = checked ? [...mahasiswa] : [];
886
887         setMahasiswa(filtered);
888     };
889
890     // Klik tombol clear
891     const handleClear = () => {
892         setMahasiswa([]);
893     };
894
895     // Klik tombol print
896     const handlePrint = () => {
897         window.print();
898     };
899
900     // Klik tombol export
901     const handleExport = () => {
902         const csvContent = `nim,nama\n${mahasiswa.map((mhs) => `${mhs.nim},${mhs.nama}`).join("\n")}`;
903
904         const blob = new Blob([csvContent], { type: "text/csv" });
905
906         const url = URL.createObjectURL(blob);
907
908         const link = document.createElement("a");
909         link.href = url;
910         link.download = "data.csv";
911
912         link.click();
913     };
914
915     // Klik tombol import
916     const handleImport = () => {
917         const fileInput = document.querySelector("#file");
918
919         fileInput.addEventListener("change", (e) => {
920             const file = e.target.files[0];
921
922             const reader = new FileReader();
923
924             reader.onload = (e) => {
925                 const csvContent = e.target.result;
926
927                 const rows = csvContent.split("\n");
928
929                 const data = rows.map((row) => {
930                     const [nim, nama] = row.split(",");
931
932                     return { nim, nama };
933                 });
934
935                 const updated = [...mahasiswa, ...data];
936
937                 setMahasiswa(updated);
938             };
939
940             reader.readAsText(file);
941         });
942     };
943
944     // Klik tombol refresh
945     const handleRefresh = () => {
946         setMahasiswa([]);
947         setForm({ nim: "", nama: "" });
948         setIsEdit(false);
949         setIsModalOpen(false);
950     };
951
952     // Klik tombol search
953     const handleSearch = (e) => {
954         e.preventDefault();
955
956         const query = e.target.value;
957
958         const filtered = mahasiswa.filter((mhs) => mhs.nama.toLowerCase().includes(query));
959
960         setMahasiswa(filtered);
961     };
962
963     // Klik tombol sort
964     const handleSort = (e) => {
965         e.preventDefault();
966
967         const direction = e.target.value;
968
969         const sorted = [...mahasiswa].sort((a, b) => {
970             if (a.nama < b.nama) {
971                 return -1;
972             } else if (a.nama > b.nama) {
973                 return 1;
974             } else {
975                 return 0;
976             }
977         });
978
979         setMahasiswa(sorted);
980     };
981
982     // Klik tombol filter
983     const handleFilter = (e) => {
984         e.preventDefault();
985
986         const checked = e.target.checked;
987
988         const filtered = checked ? [...mahasiswa] : [];
989
990         setMahasiswa(filtered);
991     };
992
993     // Klik tombol clear
994     const handleClear = () => {
995         setMahasiswa([]);
996     };
997
998     // Klik tombol print
999     const handlePrint = () => {
1000        window.print();
1001    };
1002
1003    // Klik tombol export
1004    const handleExport = () => {
1005        const csvContent = `nim,nama\n${mahasiswa.map((mhs) => `${mhs.nim},${mhs.nama}`).join("\n")}`;
1006
1007        const blob = new Blob([csvContent], { type: "text/csv" });
1008
1009        const url = URL.createObjectURL(blob);
1010
1011        const link = document.createElement("a");
1012        link.href = url;
1013        link.download = "data.csv";
1014
1015        link.click();
1016    };
1017
1018    // Klik tombol import
1019    const handleImport = () => {
1020        const fileInput = document.querySelector("#file");
1021
1022        fileInput.addEventListener("change", (e) => {
1023            const file = e.target.files[0];
1024
1025            const reader = new FileReader();
1026
1027            reader.onload = (e) => {
1028                const csvContent = e.target.result;
1029
1030                const rows = csvContent.split("\n");
1031
1032                const data = rows.map((row) => {
1033                    const [nim, nama] = row.split(",");
1034
1035                    return { nim, nama };
1036                });
1037
1038                const updated = [...mahasiswa, ...data];
1039
1040                setMahasiswa(updated);
1041            };
1042
1043            reader.readAsText(file);
1044        });
1045    };
1046
1047    // Klik tombol refresh
1048    const handleRefresh = () => {
1049        setMahasiswa([]);
1050        setForm({ nim: "", nama: "" });
1051        setIsEdit(false);
1052        setIsModalOpen(false);
1053    };
1054
1055    // Klik tombol search
1056    const handleSearch = (e) => {
1057        e.preventDefault();
1058
1059        const query = e.target.value;
1060
1061        const filtered = mahasiswa.filter((mhs) => mhs.nama.toLowerCase().includes(query));
1062
1063        setMahasiswa(filtered);
1064    };
1065
1066    // Klik tombol sort
1067    const handleSort = (e) => {
1068        e.preventDefault();
1069
1070        const direction = e.target.value;
1071
1072        const sorted = [...mahasiswa].sort((a, b) => {
1073            if (a.nama < b.nama) {
1074                return -1;
1075            } else if (a.nama > b.nama) {
1076                return 1;
1077            } else {
1078                return 0;
1079            }
1080        });
1081
1082        setMahasiswa(sorted);
1083    };
1084
1085    // Klik tombol filter
1086    const handleFilter = (e) => {
1087        e.preventDefault();
1088
1089        const checked = e.target.checked;
1090
1091        const filtered = checked ? [...mahasiswa] : [];
1092
1093        setMahasiswa(filtered);
1094    };
1095
1096    // Klik tombol clear
1097    const handleClear = () => {
1098        setMahasiswa([]);
1099    };
1100
1101    // Klik tombol print
1102    const handlePrint = () => {
1103        window.print();
1104    };
1105
1106    // Klik tombol export
1107    const handleExport = () => {
1108        const csvContent = `nim,nama\n${mahasiswa.map((mhs) => `${mhs.nim},${mhs.nama}`).join("\n")}`;
1109
1110        const blob = new Blob([csvContent], { type: "text/csv" });
1111
1112        const url = URL.createObjectURL(blob);
1113
1114        const link = document.createElement("a");
1115        link.href = url;
1116        link.download = "data.csv";
1117
1118        link.click();
1119    };
1120
1121    // Klik tombol import
1122    const handleImport = () => {
1123        const fileInput = document.querySelector("#file");
1124
1125        fileInput.addEventListener("change", (e) => {
1126            const file = e.target.files[0];
1127
1128            const reader = new FileReader();
1129
1130            reader.onload = (e) => {
1131                const csvContent = e.target.result;
1132
1133                const rows = csvContent.split("\n");
1134
1135                const data = rows.map((row) => {
1136                    const [nim, nama] = row.split(",");
1137
1138                    return { nim, nama };
1139                });
1140
1141                const updated = [...mahasiswa, ...data];
1142
1143                setMahasiswa(updated);
1144            };
1145
1146            reader.readAsText(file);
1147        });
1148    };
1149
1150    // Klik tombol refresh
1151    const handleRefresh = () => {
1152        setMahasiswa([]);
1153        setForm({ nim: "", nama: "" });
1154        setIsEdit(false);
1155        setIsModalOpen(false);
1156    };
1157
1158    // Klik tombol search
1159    const handleSearch = (e) => {
1160        e.preventDefault();
1161
1162        const query = e.target.value;
1163
1164        const filtered = mahasiswa.filter((mhs) => mhs.nama.toLowerCase().includes(query));
1165
1166        setMahasiswa(filtered);
1167    };
1168
1169    // Klik tombol sort
1170    const handleSort = (e) => {
1171        e.preventDefault();
1172
1173        const direction = e.target.value;
1174
1175        const sorted = [...mahasiswa].sort((a, b) => {
1176            if (a.nama < b.nama) {
1177                return -1;
1178            } else if (a.nama > b.nama) {
1179                return 1;
1180            } else {
1181                return 0;
1182            }
1183        });
1184
1185        setMahasiswa(sorted);
1186    };
1187
1188    // Klik tombol filter
1189    const handleFilter = (e) => {
1190        e.preventDefault();
1191
1192        const checked = e.target.checked;
1193
1194        const filtered = checked ? [...mahasiswa] : [];
1195
1196        setMahasiswa(filtered);
1197    };
1198
1199    // Klik tombol clear
1200    const handleClear = () => {
1201        setMahasiswa([]);
1202    };
1203
1204    // Klik tombol print
1205    const handlePrint = () => {
1206        window.print();
1207    };
1208
1209    // Klik tombol export
1210    const handleExport = () => {
1211        const csvContent = `nim,nama\n${mahasiswa.map((mhs) => `${mhs.nim},${mhs.nama}`).join("\n")}`;
1212
1213        const blob = new Blob([csvContent], { type: "text/csv" });
1214
1215        const url = URL.createObjectURL(blob);
1216
1217        const link = document.createElement("a");
1218        link.href = url;
1219        link.download = "data.csv";
1220
1221        link.click();
1222    };
1223
1224    // Klik tombol import
1225    const handleImport = () => {
1226        const fileInput = document.querySelector("#file");
1227
1228        fileInput.addEventListener("change", (e)
```

Pertemuan-3 > src > Pages > Admin >  Mahasiswa.jsx > ...

```
10  const Mahasiswa = () => {
75      const handleSubmit = (e) => {
82
83          if (isEdit) {
84              updateMahasiswa(originalNim, form);
85              alert("Data berhasil diperbarui!");
86          } else {
87              const exists = mahasiswa.some((mhs) => mhs.nim === form.nim);
88              if (exists) {
89                  alert("NIM sudah terdaftar!");
90                  return;
91              }
92              addMahasiswa(form);
93              alert("Data berhasil ditambahkan!");
94          }
95
96          setForm({ nim: "", nama: "" });
97          setIsEdit(false);
98          setIsModalOpen(false);
99          setOriginalNim(null);
100     };
101
102     // Modal Tambah/Edit
103     const openAddModal = () => {
104         setIsModalOpen(true);
105         setForm({ nim: "", nama: "" });
106         setIsEdit(false);
107     };
108
109     return (
110         <>
111             <Card>
112                 <div className="flex justify-between items-center mb-4">
113                     <Heading as="h2" className="mb-0 text-left">
114                         Daftar Mahasiswa
115                     </Heading>
116                     <Button onClick={openAddModal}>+ Tambah Mahasiswa</Button>
117                 </div>
118
119             <TableMahasiswa
120                 data={mahasiswa}
121                 onEdit={handleEdit}
```

Pertemuan-3 > src > Pages > Admin > Mahasiswa.jsx > ...

```
10  const Mahasiswa = () => {
118
119      <TableMahasiswa
120          data={mahasiswa}
121          onEdit={handleEdit}
122          onDelete={deleteMahasiswa}
123          onDetail={(nim) => navigate(`/admin/mahasiswa/${nim}`)}
124      />
125      </Card>
126
127      <ModalMahasiswa
128          isOpen={isModalOpen}
129          isEdit={isEdit}
130          form={form}
131          onChange={handleChange}
132          onClose={() => setIsModalOpen(false)}
133          onSubmit={handleSubmit}
134      />
135      </>
136  );
137 };
138
139 export default Mahasiswa;
140
```

/src/Pages/Admin/ModalMahasiswa.jsx

```
Pertemuan-3 > src > Pages > Admin > ModalMahasiswa.jsx > ...
79  import Form from "@/Pages/Layouts/Components/Form";
80  import Input from "@/Pages/Layouts/Components/Input";
81  import Label from "@/Pages/Layouts/Components/Label";
82  import Button from "@/Pages/Layouts/Components/Button";
83
84  const ModalMahasiswa = ({
85    isOpen,
86    isEdit,
87    form,
88    onChange,
89    onClose,
90    onSubmit,
91  }) => {
92    if (!isOpen) return null;
93
94    return (
95      <div className="fixed inset-0 flex items-center justify-center bg-[rgba(0,0,0,0.3)] z-50">
96        <div className="bg-white rounded-lg shadow-lg w-full max-w-md">
97          {/* Header Modal */}
98          <div className="flex justify-between items-center p-4 border-b">
99            <h2 className="text-lg font-semibold">
100              {isEdit ? "Edit Mahasiswa" : "Tambah Mahasiswa"}
101            </h2>
102            <button
103              onClick={onClose}
104              className="text-gray-600 hover:text-red-500 text-xl"
105              >
106              &times;
107            </button>
108          </div>
109
110          {/* Form Input */}
111          <Form onSubmit={onSubmit} className="p-4 space-y-4">
112            <div>
113              <Label htmlFor="nim">NIM</Label>
114              <Input
115                type="text"
116                name="nim"
117                value={form.nim}
118                onChange={onChange}
119                placeholder="Masukkan NIM"
120                required
121              </Input>
122            </div>
123          </Form>
124        </div>
125      </div>
126    );
127  );
128}
```

Pertemuan-3 > src > Pages > Admin >  ModalMahasiswa.jsx > ...

```
84  const ModalMahasiswa = ({  
117    value={form.nim}  
118    onChange={onChange}  
119    placeholder="Masukkan NIM"  
120    required  
121    />  
122  </div>  
123  
124  <div>  
125    <Label htmlFor="nama">Nama</Label>  
126    <Input  
127      type="text"  
128      name="nama"  
129      value={form.nama}  
130      onChange={onChange}  
131      placeholder="Masukkan Nama"  
132      required  
133      />  
134  </div>  
135  
136  <div className="flex justify-end space-x-2 mt-4">  
137    <Button type="button" onClick={onClose} variant="secondary">  
138      Batal  
139    </Button>  
140    <Button type="submit">{isEdit ? "Perbarui" : "Simpan"}</Button>  
141  </div>  
142  </Form>  
143  </div>  
144  </div>  
145);  
146};  
147  
148 export default ModalMahasiswa;  
149
```

/src/Pages/Admin/TableMahasiswa.jsx

```
Pertemuan-3 > src > Pages > Admin > TableMahasiswa.jsx > default
 1  import Button from "@Pages/Layouts/Components/Button";
 2
 3  const TableMahasiswa = ({ data = [], onEdit, onDelete, onDetail }) => {
 4    return (
 5      <table className="w-full text-sm text-gray-700">
 6        <thead className="bg-blue-600 text-white">
 7          <tr>
 8            <th className="py-2 px-4 text-left">NIM</th>
 9            <th className="py-2 px-4 text-left">Nama</th>
10            <th className="py-2 px-4 text-center">Aksi</th>
11          </tr>
12        </thead>
13        <tbody>
14          {data.map((mhs, index) => (
15            <tr key={mhs.nim} className={index % 2 === 0 ? "bg-white" : "bg-gray-100"}>
16              <td className="py-2 px-4">{mhs.nim}</td>
17              <td className="py-2 px-4">{mhs.nama}</td>
18              <td className="py-2 px-4 text-center space-x-2">
19                <Button size="sm" onClick={() => onDetail(mhs.nim)}>Detail</Button>
20                <Button size="sm" variant="warning" onClick={() => onEdit(mhs)}>
21                  Edit
22                </Button>
23                <Button size="sm" variant="danger" onClick={() => onDelete(mhs.nim)}>
24                  Hapus
25                </Button>
26              </td>
27            </tr>
28          )));
29        </tbody>
30      </table>
31    );
32  };
33
34  export default TableMahasiswa;
```

Hasil Tampilan:

The screenshot shows the 'Mahasiswa' section of the Admin Dashboard. The left sidebar has 'Mahasiswa' selected. The main area displays a table titled 'Daftar Mahasiswa' with columns 'NIM' and 'Nama'. Three rows are listed: A11.2022.14655 (Nauval Sutisna), 20211002 (Siti Aminah), and 20211003 (Ahmad Fauzi). Each row has 'Detail', 'Edit', and 'Hapus' buttons. A blue button '+ Tambah Mahasiswa' is at the top right.

NIM	Nama	Aksi
A11.2022.14655	Nauval Sutisna	[Detail] [Edit] [Hapus]
20211002	Siti Aminah	[Detail] [Edit] [Hapus]
20211003	Ahmad Fauzi	[Detail] [Edit] [Hapus]

The modal window is titled 'Tambah Mahasiswa'. It has fields for 'NIM' (A11.2022.14656) and 'Nama' (empty). A validation message 'Harap isi bidang ini.' is shown above the 'Simpan' button. The 'Simpan' button is blue.

NIM
A11.2022.14656

Nama
Masukkan Nama

! Harap isi bidang ini. **Simpan**

The screenshot shows the 'Mahasiswa' section again. The table now includes a fourth row: A11.2022.14656 (Alifqi). The 'Edit' and 'Hapus' buttons for this row are highlighted in yellow.

NIM	Nama	Aksi
A11.2022.14655	Nauval Sutisna	[Detail] [Edit] [Hapus]
20211002	Siti Aminah	[Detail] [Edit] [Hapus]
20211003	Ahmad Fauzi	[Detail] [Edit] [Hapus]
A11.2022.14656	Alifqi	[Detail] [Edit] [Hapus]

Edit Mahasiswa

NIM
20211002

Nama
Siti Aminah

Batal **Perbarui**

Edit Mahasiswa

NIM
A11.2000.14555

Nama
Siti Stabilizer

Batal **Perbarui**

Daftar Mahasiswa

+ Tambah Mahasiswa

NIM	Nama	Aksi
A11.2022.14655	Nauval Sutisna	Detail Edit Hapus
A11.2000.14555	Siti Stabilizer	Detail Edit Hapus
20211003	Ahmad Fauzi	Detail Edit Hapus
A11.2022.14656	Alifqi	Detail Edit Hapus

Menghapus Siti Stabilizer

A screenshot of a web browser displaying a confirmation dialog box. The dialog box is centered and contains the text "localhost:5173 menyatakan" followed by "Yakin ingin hapus data ini?". Below the dialog are two buttons: "Oke" (in green) and "Batal" (in blue). The background shows a dark-themed administrative interface titled "Mahasiswa" under the "Admin" sidebar. The main content area is titled "Daftar Mahasiswa" and lists four student records:

NIM	Nama	Aksi
A11.2022.14655	Nauval Sutisna	[Detail] [Edit] [Hapus]
A11.2000.14555	Siti Stabilizer	[Detail] [Edit] [Hapus]
20211003	Ahmad Fauzi	[Detail] [Edit] [Hapus]
A11.2022.14656	Alifqi	[Detail] [Edit] [Hapus]

A small "+ Tambah Mahasiswa" button is located in the top right corner of the main content area.

A screenshot of the "Daftar Mahasiswa" page. The page title is "Daftar Mahasiswa" and features a "+ Tambah Mahasiswa" button. The main content area displays a table with three student records:

NIM	Nama	Aksi
A11.2022.14655	Nauval Sutisna	[Detail] [Edit] [Hapus]
20211003	Ahmad Fauzi	[Detail] [Edit] [Hapus]
A11.2022.14656	Alifqi	[Detail] [Edit] [Hapus]