


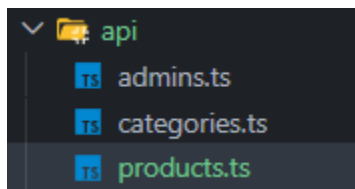
Nama : Prayogi Setiawan  
NPM : 22312009  
Kelas : “yogi: crud products & fix panel layout”

Sebelumnya **layout.tsx** di folder panel menggunakan ‘*use client*’, jadi untuk metadata saya menggunakan tag **Head** dari next. Tapi tidak terdeteksi Ketika dijalankan, jadi saya ubah menggunakan ‘*use server*’, dan saya tambahkan metadata menggunakan **Metadata** dari next.



```
1 export const metadata: Metadata = {  
2   title: "Sokraf Panel",  
3   description: "Panel for sokraf management",  
4   icons: {  
5     icon: "/favicon.ico",  
6     shortcut: "/favicon.ico",  
7   }  
8 };
```

Buat file **products.ts** di dalam folder **api** sebagai controller untuk memproses semua CRUD dari products.



```
▼ api  
  TS admins.ts  
  TS categories.ts  
  TS products.ts
```

Isikan file **products.ts** seperti berikut.

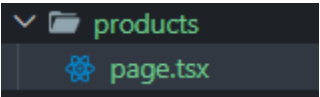
```
1  'use server';
2
3  import { PrismaClient } from '@prisma/client';
4  import slugify from 'slugify';
5  import { v4 as uuidv4 } from 'uuid';
6  import { writeFileSync } from 'fs';
7  import { join } from 'path';
8  import { unlink } from 'fs/promises';
9
10 const prisma = new PrismaClient();
11
12 export async function createProduct(data: {
13   name: string;
14   description: string;
15   price: number;
16   categoryId: number;
17   whatsapp: string;
18   image: File;
19 }) {
20   const uniqueName = `${uuidv4()}-${data.image.name}`;
21   const imagePath = join(process.cwd(), "public", "images", uniqueName);
22   const arrayBuffer = await data.image.arrayBuffer();
23   await writeFileSync(imagePath, Buffer.from(arrayBuffer));
24
25   return prisma.product.create({
26     data: {
27       name: data.name,
28       slug: slugify(data.name, { lower: true }),
29       description: data.description,
30       price: data.price,
31       categoryId: data.categoryId,
32       whatsapp: data.whatsapp,
33       image: `images/${uniqueName}`,
34     },
35   });
36 }
37
38 export async function getAllProducts() {
39   return prisma.product.findMany({
40     orderBy: { id: 'desc' },
41     include: {
42       category: {
43         select: { name: true }
44       }
45     },
46   })
47 }
48
49 export async function getProductById(id: number) {
50   return prisma.product.findUnique({
51     where: { id },
52     include: {
53       category: true
54     },
55   })
56 }
57
58 export async function updateProduct(id: number, data: {
59   name: string,
60   description: string,
61   price: number,
62   categoryId: number,
63   whatsapp: string,
64 }) {
65   const slug = slugify(data.name, { lower: true });
66
67   return prisma.product.update({
68     where: { id },
69     data: {
70       name: data.name,
71       slug,
72       description: data.description,
73       price: data.price,
74       categoryId: data.categoryId,
75       whatsapp: data.whatsapp
76     },
77   });
78 }
79
80 export async function deleteProduct(id: number) {
81   const product = await prisma.product.findUnique({
82     where: { id },
83     select: { image: true }
84   });
85
86   if (!product) {
87     throw new Error('Product not found');
88   }
89
90   const imagePath = join(process.cwd(), 'public', product.image);
91
92   try {
93     await unlink(imagePath);
94
95     await prisma.product.delete({
96       where: { id }
97     })
98   } catch (error) {
99     console.error('Error deleting product or image', error);
100     throw new Error('Failed to delete product and image');
101   }
102 }
```

Di atas saya membuat 5 fungsi untuk melakukan CRUD. **createProduct()** untuk membuat data product baru, **getAllProducts()** untuk mengambil semua data product, **getProductById()** untuk mengambil product berdasarkan id, **updateProduct()** untuk handle mengubah data product, dan **deleteProduct()** untuk menghapus data product.

Install juga library uuid untuk membuat nama image nya menjadi lebih unik.

```
npm install uuid
```

Membuat folder **products** di dalam folder **panel** untuk membuat route `/products`. Di dalamnya buat juga file **page.tsx** untuk membuat tampilan halamannya.



Membuat isi tampilan products.

```
1 'use client';
2
3 import React, { useEffect, useState } from "react";
4 import { deleteProduct, getAllProducts } from "../api/products";
5 import Link from "next/link";
6 import { useRouter } from "next/navigation";
7 import {
8   Breadcrumb,
9   BreadcrumbItem,
10  Button,
11  Table,
12  TableBody,
13  TableCell,
14  TableHead,
15  TableHeadCell,
16  TableRow,
17 } from "flowbite-react";
18 import { HiHome, HiOutlinePencilAlt, HiOutlineTrash, HiPlus } from "react-icons/hi";
```

Menyiapkan semua kebutuhan pengambilan dan wadah data.

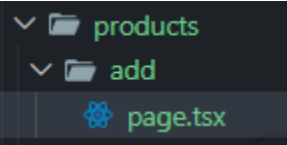
```
1
2 const [getProducts, setProducts] = useState({});
3 const router = useRouter();
4
5 async function fetchData() {
6   setProducts(await getAllProducts());
7 }
8
9 useEffect(() => {
10   fetchData();
11 }, []);
12
13 const handleDelete = async (id: number) => {
14   if (confirm('Are you sure you want to delete this product?')) {
15     await deleteProduct(id);
16     fetchData();
17   }
18 }
19
20 function formatRupiah(angka: number): string {
21   if (isNaN(angka)) {
22     return '';
23   }
24
25   return 'Rp ' + angka.toLocaleString('id-ID');
26 }
```

Buat tampilan table untuk pengelolaannya dan juga tombol action delete nya.

```
1
2  return (
3    <div>
4      <Breadcrumb aria-label="Default breadcrumb example" className="mt-3 mb-5">
5        <BreadcrumbItem href="/panel" icon={HiHome}>
6          <Link href="/panel">
7            Dashboard
8          </Link>
9        </BreadcrumbItem>
10       <BreadcrumbItem>Products</BreadcrumbItem>
11     </Breadcrumb>
12     <div className="flex justify-between items-center mb-4">
13       <h2 className="text-xl font-medium">Products</h2>
14       <Button as={Link} href="/panel/products/add" color="blue">
15         <HiPlus className="mr-2 h-5 w-5" />
16         Add Product
17       </Button>
18     </div>
19     <div className="shadow p-5 rounded mb-10">
20       <div className="overflow-x-auto">
21         <Table>
22           <TableHead>
23             <TableHeadCell>#</TableHeadCell>
24             <TableHeadCell>Product Name</TableHeadCell>
25             <TableHeadCell>Category</TableHeadCell>
26             <TableHeadCell>Price</TableHeadCell>
27             <TableHeadCell>WhatsApp</TableHeadCell>
28             <TableHeadCell className="text-center">Action</TableHeadCell>
29           </TableHead>
30           <TableBody className="divide-y">
31             {Object.values(getProducts)?.map((product: any, index: number) => (
32               <TableRow key={product.id} className="bg-white dark:border-gray-700 dark:bg-gray-800">
33                 <TableCell>{index + 1}</TableCell>
34                 <TableCell className="whitespace-nowrap font-medium text-gray-900 dark:text-white">
35                   {product.name}
36                 </TableCell>
37                 <TableCell>{product.category.name}</TableCell>
38                 <TableCell>{formatRupiah(product.price)}</TableCell>
39                 <TableCell>+62{[product.whatsapp]}</TableCell>
40                 <TableCell>
41                   <div className="flex gap-1 justify-center">
42                     <Button as={Link} href={`"/panel/products/edit/${product.id}`} color="blue" size="xs">
43                       <HiOutlinePencilAlt className="h-5 w-5" />
44                     </Button>
45                     <Button color="failure" size="xs" onClick={() => handleDelete(product.id)}>
46                       <HiOutlineTrash className="h-5 w-5" />
47                     </div>
48                   </TableCell>
49                 </TableRow>
50               </TableRow>
51             )})
52           </TableBody>
53         </Table>
54       </div>
55     </div>
56   </div>
57 )
```

Tambah Product

Buat folder **add** di dalam folder **products**, lalu buat juga **page.tsx** nya.



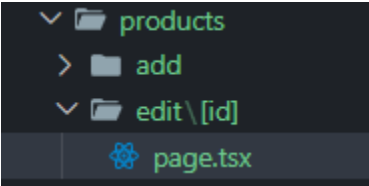
Untuk tampilannya seperti ini.

```
1  'use client';
2
3  import Link from "next/link"
4  import {
5    Breadcrumb,
6    BreadcrumbItem,
7    Button,
8    Label,
9    TextInput,
10   Select,
11   Textarea,
12   FileInput,
13 } from "flowbite-react";
14 import { HiHome, HiArrowSmLeft, HiOutlineSave } from "react-icons/hi";
15 import React, { useEffect, useState } from "react";
16 import { useRouter } from "next/navigation";
17 import { createProduct } from "../api/products";
18 import { getAllCategories } from "../api/categories";
19
20 export default function AddProduct() {
21   const [categories, setCategories] = useState<{ id: number; name: string }[]>([]);
22
23   useEffect(() => {
24     const fetchCategories = async () => {
25       const data = await getAllCategories();
26       setCategories(data);
27     }
28
29     fetchCategories();
30   }, []);
31
32   const [form, setForm] = useState<{
33     name: string,
34     description: string,
35     price: string,
36     categoryId: string,
37     whatsapp: string,
38     image: File | null,
39   }>({
40     name: "",
41     description: "",
42     price: "",
43     categoryId: "",
44     whatsapp: "",
45     image: null,
46   });
47
48   const router = useRouter();
49
50   const handleChange = (e: React.ChangeEvent<HTMLInputElement | HTMLTextAreaElement>) => {
51     const { name, value } = e.target;
52     setForm({ ...form, [name]: value });
53   };
54
55   const handleFileChange = (e: React.ChangeEvent<HTMLInputElement>) => {
56     if (e.target.files?.[0]) {
57       setForm({ ...form, image: e.target.files[0] });
58     }
59   };
60
61   const handleSubmit = async (e: React.FormEvent) => {
62     e.preventDefault();
63
64     if (!form.image) {
65       alert("Please upload an image.");
66       return;
67     }
68
69     const data = {
70       name: form.name,
71       description: form.description,
72       price: parseInt(form.price),
73       categoryId: parseInt(form.categoryId, 10),
74       whatsapp: form.whatsapp,
75       image: form.image,
76     };
77
78     await createProduct(data);
79     router.push('/panel/products');
80   }
```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
return (
  <div>
    <Breadcrumb aria-label="Default breadcrumb example" className="mt-3 mb-5">
      <BreadcrumbItem href="/panel" icon={HiHome}>
        <Link href="/panel">
          Dashboard
        </Link>
      </BreadcrumbItem>
      <BreadcrumbItem href="/panel/products">
        <Link href="/panel/products">
          Products
        </Link>
      </BreadcrumbItem>
      <BreadcrumbItem>Add Product</BreadcrumbItem>
    </Breadcrumb>
    <div className="flex justify-between items-center mb-4">
      <h2 className="text-xl font-medium">Add Product</h2>
      <Button as={Link} href="/panel/products" color="blue">
        <HiArrowSmLeft className="mr-2 h-5 w-5" />
        Back
      </Button>
    </div>
    <div className="grid grid-cols-12 gap-6">
      <div className="col-span-8">
        <div className="shadow p-5 rounded">
          <form onSubmit={handleSubmit} id="form-product">
            <div className="mb-3">
              <div className="mb-2 block">
                <Label value="Product Name" />
              </div>
              <TextInput name="name" type="text" value={form.name} onChange={handleChange} required />
            </div>
            <div className="grid grid-cols-3 gap-4">
              <div className="mb-3">
                <div className="mb-2 block">
                  <Label value="Category" />
                </div>
                <Select name="categoryId" value={form.categoryId} onChange={(e) => setForm({... form, categoryId: e.target.value}} required>
                  <option value="">Select category</option>
                  {categories.map((category) => (
                    <option key={category.id} value={category.id}>{category.name}</option>
                  ))}
                </Select>
              </div>
              <div className="mb-3">
                <div className="mb-2 block">
                  <Label value="Price" />
                </div>
                <TextInput name="price" type="number" addon="Rp" value={form.price} onChange={handleChange} required />
              </div>
              <div className="mb-3">
                <div className="mb-2 block">
                  <Label value="WhatsApp" />
                </div>
                <TextInput name="whatsapp" type="number" addon="+62" value={form.whatsapp} onChange={handleChange} required />
              </div>
            </div>
            <div className="mb-3">
              <div className="mb-2 block">
                <Label value="Description" />
              </div>
              <Textarea name="description" required rows={4} value={form.description} onChange={handleChange} />
            </div>
            <Button type="submit" color="blue">
              <HiOutlineSave className="mr-2 h-5 w-5" />
              Save
            </Button>
          </form>
        </div>
      </div>
      <div className="col-span-4">
        <div className="shadow p-5 rounded">
          <div className="mb-3">
            <div className="mb-2 block">
              <Label value="Image" />
            </div>
            <FileInput name="image" form="form-product" accept="image/*" onChange={handleFileChange} required />
          </div>
        </div>
      </div>
    </div>
  </div>
)
};
```

Edit Product

Buat folder **edit** di dalam folder **products**, lalu folder **[id]** untuk menampung id product nya, dan file **page.tsx** nya.



Buat isi halamannya seperti berikut. Ambil data product berdasarkan id menggunakan fungsi getProductById().

```
1  'use client';
2
3  import Link from "next/link"
4  import {
5    Breadcrumb,
6    BreadcrumbItem,
7    Button,
8    Label,
9    TextInput,
10   Select,
11   Textarea,
12   FileInput,
13 } from "flowbite-react";
14 import { HiHome, HiArrowSmLeft, HiOutlineSave } from "react-icons/hi";
15 import React, { useEffect, useState } from "react";
16 import { useRouter, useParams } from "next/navigation";
17 import { updateProduct, getProductById } from "../api/products";
18 import { getAllCategories } from "../api/categories";
19
20 export default function EditProduct() {
21   const [categories, setCategories] = useState<{ id: number; name: string }[]>([]);
22   const [form, setForm] = useState<{
23     name: string,
24     description: string,
25     price: string,
26     categoryId: string,
27     whatsapp: string,
28     image: string,
29   }>({
30     name: "",
31     description: "",
32     price: "",
33     categoryId: "",
34     whatsapp: "",
35     image: ""
36   });
37
38   const router = useRouter();
39   const { id } = useParams();
40
41   useEffect(() => {
42     const fetchProduct = async () => {
43       const product = await getProductById(Number(id));
44       if (product) {
45         setForm({
46           name: product.name,
47           description: product.description,
48           price: product.price.toString(),
49           categoryId: product.categoryId.toString(),
50           whatsapp: product.whatsapp,
51           image: product.image
52         });
53       }
54     };
55
56     const fetchCategories = async () => {
57       const data = await getAllCategories();
58       setCategories(data);
59     };
60
61     fetchProduct();
62     fetchCategories();
63   }, [id]);
64
65   const handleChange = (e: React.ChangeEvent<HTMLInputElement | HTMLTextAreaElement>) => {
66     const { name, value } = e.target;
67     setForm({ ...form, [name]: value });
68   };
69
70   const handleSubmit = async (e: React.FormEvent) => {
71     e.preventDefault();
72
73     const data = {
74       id,
75       name: form.name,
76       description: form.description,
77       price: parseInt(form.price),
78       categoryId: parseInt(form.categoryId, 10),
79       whatsapp: form.whatsapp,
80     };
81
82     await updateProduct(Number(id), data);
83     router.push('/panel/products');
84   };
85 }
```



```

1
2   return (
3     <div>
4       <Breadcrumbs aria-label="Default breadcrumb example" className="mt-3 mb-5">
5         <BreadcrumbItem href="/panel" icon={HiHome}>
6           <Link href="/panel">
7             Dashboard
8           </Link>
9         </BreadcrumbItem>
10        <BreadcrumbItem href="/panel/products">
11          <Link href="/panel/products">
12            Products
13          </Link>
14        </BreadcrumbItem>
15        <BreadcrumbItem>Edit Product</BreadcrumbItem>
16      </Breadcrumbs>
17      <div className="flex justify-between items-center mb-4">
18        <h2 className="text-xl font-medium">Edit Product</h2>
19        <Button as={Link} href="/panel/products" color="blue">
20          <HiArrowSmLeft className="mr-2 h-5 w-5" />
21          Back
22        </Button>
23      </div>
24      <div className="grid grid-cols-12 gap-6 pb-10">
25        <div className="col-span-8">
26          <div className="shadow p-5 rounded">
27            <form onSubmit={handleSubmit} id="form-product">
28              <div className="mb-3">
29                <div className="mb-2 block">
30                  <Label value="Product Name" />
31                </div>
32                <TextInput name="name" type="text" value={form.name} onChange={handleChange} required />
33              </div>
34              <div className="grid grid-cols-3 gap-4">
35                <div className="mb-3">
36                  <div className="mb-2 block">
37                    <Label value="Category" />
38                  </div>
39                  <Select name="categoryId" value={form.categoryId} onChange={(e) => setForm({ ...form, categoryId: e.target.value })} required>
40                    <option value="">Select category</option>
41                    {categories.map((category) => (
42                      <option key={category.id} value={category.id}>{category.name}</option>
43                    ))}
44                  </Select>
45                </div>
46                <div className="mb-3">
47                  <div className="mb-2 block">
48                    <Label value="Price" />
49                  </div>
50                  <TextInput name="price" type="number" addon="Rp" value={form.price} onChange={handleChange} required />
51                </div>
52                <div className="mb-3">
53                  <div className="mb-2 block">
54                    <Label value="WhatsApp" />
55                  </div>
56                  <TextInput name="whatsapp" type="number" addon="+62" value={form.whatsapp} onChange={handleChange} required />
57                </div>
58              </div>
59              <div className="mb-3">
60                <div className="mb-2 block">
61                  <Label value="Description" />
62                </div>
63                <Textarea name="description" required rows={4} value={form.description} onChange={handleChange} />
64              </div>
65              <Button type="submit" color="blue">
66                <HiOutlineSave className="mr-2 h-5 w-5" />
67                Save
68              </Button>
69            </form>
70          </div>
71        </div>
72        <div className="col-span-4">
73          <div className="shadow p-5 rounded">
74            <img src={`/${form.image}`} alt="" className="w-full"/>
75          </div>
76        </div>
77      </div>
78    </div>
79  );
80 }
81

```