

Nama : Rifki Abdullah

NPM : 22312080

DOKUMENTASI Pengerjaan

Membuat Komponen Form

Tanggal : 07 Januari 2025

Waktu : 18.54

- Menginstall komponen **form** dengan perintah **npx shadcn@latest add form** dari Pustaka **shadcn/ui** kedalam proyek, dengan dokumentasi terminal berikut ini :

```
PS D:\ecommerce-visionvortex> npx shadcn@latest add form
✓ Checking registry.
  Installing dependencies.

It looks like you are using React 19.
Some packages may fail to install due to peer dependency issues in npm (see https://ui.shadcn.com/react-19).

✓ How would you like to proceed? » Use --legacy-peer-deps
✓ Installing dependencies.
✓ The file button.tsx already exists. Would you like to overwrite? ... yes
✓ Created 2 files:
  - components\ui\form.tsx
  - components\ui\label.tsx
i Updated 1 file:
  - components\ui\button.tsx

npm notice
npm notice New major version of npm available! 10.9.0 -> 11.0.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.0.0
npm notice To update run: npm install -g npm@11.0.0
npm notice
PS D:\ecommerce-visionvortex>
```

- Berikut ini hasil yang sudah terinstall dengan menghasilkan file baru berupa **form.tsx** dan **label.tsx** pada folder **ui** didalam **components** :



Tanggal : 07 Januari 2025

Waktu : 19.21

- Menambahkan komponen **input** dengan perintah **npx shadcn@latest add input** dari Pustaka **shadcn/ui** kedalam proaiek, dengan dokumentasi terminal berikut ini :

```
PS D:\ecommerce-visionvortex> npx shadcn@latest add input
✓ Checking registry.
✓ Created 1 file:
  - components\ui\input.tsx
```

- Berikut ini hasil yang sudah terinstall dengan menghasilkan file baru berupa **input.tsx** pada pada folder **ui** didalam **components** :



Tanggal : 08 Januari 2025

Waktu : 00.02

Melanjutkan membuat form komponen

Pada file **store-modal.tsx**

```
3 import * as z from "zod";
4 import { useForm } from "react-hook-form";
5 import { zodResolver } from "@hookform/resolvers/zod";
```

Menambahkan import seperti diatas seperti zod untuk membuat dan memvalidasi skema data, lalu import useForm yaitu hook dari Pustaka react hook Form, lalu menambahkan import zodResolver supaya zod dengan React Hook Form terintegrasikan.

```

11 import {
12     Form,
13     FormControl,
14     FormField,
15     FormItem,
16     FormLabel,
17     FormMessage
18 } from "@components/ui/form";
19 import { Input } from "@components/ui/input";
20 import { Button } from "@components/ui/button";

```

Menambahkan import berbagai komponen UI terkait (Form, FormControl, FormField, FormItem, FormLabel, FormMessage). Mengimport komponen (Input), dan mengimport komponen tombol (Button).

```

22 const formSchema = z.object({
23     name: z.string().min(1),
24 });

```

Menambahkan const formSchema untuk mendefinisikan skema validasi formulir menggunakan Zod dan `name` adalah field teks wajib dengan minimal 1 karakter

```

const form = useForm<z.infer<typeof formSchema>>({
    resolver: zodResolver(formSchema),
    defaultValues: {
        name: "",
    },
});

```

Menambahkan const baru untuk menginisialisasi hook `useForm` dengan resolver Zod untuk validasi skema dengan `defaultValues` menetapkan nilai default untuk input formulir, di sini `name` diinisialisasi sebagai string kosong.

```

const onSubmit = async (values: z.infer<typeof formSchema>) => {
    console.log(values);
    // TODO: Create Store
}

```

Menambahkan const baru untuk mendefinisikan fungsi `onSubmit`, yang akan dipanggil saat formulir dikirimkan.

```

44     <Modal
45       title="create store"
46       description="Add a new store to manage products and categories"
47       isOpen={storeModal.isOpen}
48       onClose={storeModal.onClose}
49     >
50       <div>
51         <div className="space-y-4 py-2 pb-4">
52           <Form {...form}>
53             <form onSubmit={form.handleSubmit(onSubmit)}>
54               <FormField
55                 control={form.control}
56                 name="name"
57                 render={({ field }) => (
58                   <FormItem>
59                     <FormLabel>Name</FormLabel>
60                     <FormControl>
61                       <Input placeholder="E-Commerce" {...field}/>
62                     </FormControl>
63                     <FormMessage />
64                   </FormItem>
65                 )}
66               />
67             <div className="pt-6 space-x-2 flex items-center justify-end w-full">
68               <Button
69                 variant="outline"
70                 onClick={storeModal.onClose}>Cancel
71               </Button>
72               <Button type="submit">Continue</Button>
73             </div>
74           </form>
75         </Form>
76       </div>
77     </div>
78   </Modal>

```

Menambahkan isi didalam Form yang terdapat :

- Membungkus formulir HTML dengan komponen `Form` untuk integrasi dengan React Hook Form dan `handleSubmit` menangani validasi dan pemrosesan data sebelum memanggil `onSubmit`.
- Komponen `FormField` agar terhubung dengan input form
- Menambahkan `render` supaya mengonfigurasi elemen input dan menampilkan pesan error jika ada.
- Dan menambahkan Input dengan isi placeholder "E-Commerce" dan dihubungkan ke field "name".
- Menambahkan tombol Button pada form yang terdiri dari Cancel dan Continue
- **Hasil form yang ditambahkan diatas**

create store



Add a new store to manage products and categories

Name

Rifki Abdullah

Cancel

Continue

- Jika table tidak terisikan

create store



Add a new store to manage products and categories

Name

E-Commerce

String must contain at least 1 character(s)

Cancel

Continue

Tanggal : 09 Januari 2025

Waktu : 14.34 s/d ...

Membuat HomePage pada Admin

- Membuat folder (dashboard) dan lalu didalamnya terdapat folder [storeId] lalu membuat file layout.tsx



- Mengerjakan isi layout.tsx

```
1 import db from "@lib/db";
2 import { auth } from "@clerk/nextjs/server";
3 import { redirect } from "next/navigation";
4 import React from "react";
```

Mengimpor modul db, mengimpor fungsi auth dari clerk, mengimpor fungsi redirect dari next.js, mengimpor modul react.

```
export default async function DashboardLayout({
  children,
  params,
}): {
  children: React.ReactNode;
  params: {storeId: string}
}) {
```

Membuat komponen layout untuk dashboard. Menambahkan komponen children, params. Untuk memastikan hanya pengguna yang login.

```
}) {
  const {userId} = await auth(); ←
  if (!userId) {
    redirect("sign-in")
  }
  const store = await db.store.findFirst({
    where: {
      id: params.storeId,
      userId
    }
  })
}
```

Mendapatkan data autentikasi

```

    }) {
      const {userId} = await auth();
      if (!userId) {
        redirect("sign-in")
      }
      const store = await db.store.findFirst({
        where: {
          id: params.storeId,
          userId
        }
      })
    }
  })

```

Membuat supaya pengguna jika blm login diarahkan ke halaman sign-in

```

    }) {
      const {userId} = await auth();
      if (!userId) {
        redirect("sign-in")
      }
      const store = await db.store.findFirst({
        where: {
          id: params.storeId,
          userId
        }
      })
    }
  })

```

Membuat query ke database untuk mencari data store pertama yang memenuhi kriteria.

```

    if (!store) {
      redirect("/")
    }
    return (
      <>
      <div>This is Navbar</div>
      {children}
    </>
  )

```

Melakukan pengecekan jika store tidak ada

- Dokumentasi hasil perubahan dari [storeId]



This is Navbar

This is Dashboard

Dia mendapatkan Id store di Alamat halaman diatas.

- Membuat Folder baru didalam [storeId] dengan nama (routes) dan lalu membuat file dengan nama page.tsx
- Isi page.tsx

```
const DashboardPage = () => {  
  return (  
    <div>  
      This is Dashboard  
    </div>  
  );  
}  
  
export default DashboardPage;
```

- Membuat file baru didalam folder (root) yaitu file layout.tsx



- Membuat isi nya :

```

app > (root) > layout.tsx > SetupLayout
1 import db from "@lib/db";
2 import { auth } from "@clerk/nextjs/server";
3 import { redirect } from "next/navigation";
4 import React from "react";
5
6 export default async function SetupLayout({
7   children,
8 }: {
9   children: React.ReactNode;
10 }) {
11   const {userId} = await auth()
12   if (!userId) {
13     redirect("sign-in")
14   }
15
16   const store = await db.store.findFirst({
17     where: {
18       userId
19     }
20   })
21
22   if (store) {
23     redirect(`/${store.id}`)
24   }
25
26   return (
27     <>
28       {children}
29     </>
30   )
31 }
32

```

Pada isinya hamper sama dengan file layout.tsx pada (dashboard)/[storeId] hanya saja menambahkan isi yang ditandai untuk memastikan store yang diminta oleh pengguna benar-benar ada dan dapat diakses.

- Membuat folder baru didalam folder (root) dengan nama (routes) dan memindahkan file page.tsx yang awalnya di (root) pindah ke (routes).



- Melakukan **npx prisma migrate reset** untuk mereset database ke keadaan awal berdasarkan migrasi prisma.

```

PS D:\Universitas Teknokrat Indonesia\Semester 5\WEB 2\ecommerce-visionvortex> npx prisma generate
Environment variables loaded from .env

```

- Pada bash tersebut melakukan perintah tersebut untuk menghasilkan client prisma yang akan digunakan.

```

PS D:\Universitas Teknokrat Indonesia\Semester 5\WEB 2\ecommerce-visionvortex> npx prisma db push

```

Push skema Prisma langsung ke database

```
const onSubmit = async (values: z.infer<typeof formSchema>) => {
  try {
    setLoading(true)
    const response = await axios.post('/api/stores', values)
    console.log(response.data);
    toast.success("Berhasil Membuat Toko");
    window.location.assign(`/${response.data.id}`)
  } catch (error) {
    toast.error("Gagal Membuat Toko")
  } finally{
    setLoading(false)
  }
}
```

- Menambahkan kode untuk mengarahkan pengguna ke URL tertentu berdasarkan data yang diterima dari respons server.
- Memodifikasi pada file page.tsx pada folder (dashboard)/[storeId]/(routes)

```
2
3 interface DashboardPageProps {
4   params: {storeId: string}
5 }
```

Menambahkan interface tersebut untuk mendeskripsikan tipe properti yang diterima oleh komponen.

```
7 const DashboardPage = async ({params} : DashboardPageProps) => {
8
```

Menambahkan properti params (dari tipe DashboardPageProps)

```
8
9 const store = await db.store.findFirst({
10   where: {
11     id: params.storeId
12   }
13 })
```

Menambahkan const baru untuk mencari data di tabel store. Query ini mencari satu entri dalam tabel store yang memiliki id sesuai dengan nilai params.storeId.

```
return (
  <div>
    Active Store: {store?.name}
  </div>
);
```

Menambahkan untuk menampilkan keterangan berdasarkan nama storenya.

Hasil penampilannya:



Tanggal : 10 Januari 2025

Waktu : 07.23 s/d ...

Membuat Navbar pada Admin



Merubah isi tampilan pada file layout.tsx pada (dashboard)/[storeId]



Lalu membuat file baru pada folder components dengan nama file navbar.tsx dan serta memberikan isinya untuk dipanggil Navbar pada layout.tsx sebelumnya.

```
layout.tsx ...[storeId]  navbar.tsx M X  store-modal.tsx  page.tsx  components > navbar.tsx > default
1  import { UserButton } from "@clerk/nextjs";
2
3  const Navbar = () => {
4    return (
5      <div className="border-b">
6        <div className="flex h-16 items-center px-4">
7          <div>Store Switcher</div>
8          <div>main nav</div>
9          <div className="ml-auto flex items-center space-x-4">
10             <UserButton afterSignOutUrl="/" />
11          </div>
12        </div>
13      </div>
14    );
15  }
16
17  export default Navbar;
```

Membuat isi navbar berupa bordernya, mengatur ukurannya, membuat userbutton-nya.

`<MainNav />`

Mengubah main nav sebelumnya menjadi seperti itu untuk memanggil hasil dari main-nav.tsx

```
layout.tsx ...[storeId]  navbar.tsx M  main-nav.tsx U X  sto components > main-nav.tsx > MainNav
1  'use client'
2
3  export function MainNav({
4    className,
5
6  }: React.HTMLAttributes<HTMLElement>) {
7    return (
8      <nav>
9        This is Main Nav
10      </nav>
11    )
12  }
```

Mendefinisikan MainNav untuk dipanggil dan di render.



Hasilnya.

```
const pathname = usePathname();
const params = useParams();

const routes = [
  {
    href: `/${params.storeId}/settings`,
    label: 'settings',
    active: pathname === `/${params.storeId}/settings`,
  }
]
```

Menambahkan code pada main-nav.tsx untuk mendefinisikan const pathname dan const params. Dan membuat const routes.

```
{routes.map((route) => (
  <Link
    key={route.href}
    href={route.href}
    className={cn(
      "text-sm font-medium transition-colors hover:text-primary",
      route.active ? "text-black dark:text-white" : "text-muted-foreground"
    )}
  >
    {route.label}
  </Link>
)}
```

Merender navigasi dinamis menggunakan data dari array routes pada main-nav.tsx

```
<MainNav className="mx-6"/>
```

Dan memberikan jarak nya.



Active Store: Toko01

-

Dan ini hasilnya.

- `PS D:\Universitas Teknokrat Indonesia\Semester 5\WEB 2\ecommerce-visionvortex> npx shadcn@latest add popover`

Menginstall popover

- `PS D:\Universitas Teknokrat Indonesia\Semester 5\WEB 2\ecommerce-visionvortex> npx shadcn@latest add command`

Menginstall command

```
import db from "@/lib/db";

const Navbar = async () => {
  const {userId} = await auth();

  if (!userId) {
    redirect("/sign-in");
  }

  const stores = await db.store.findMany({
    where: {
      userId,
    }
  })

  return (
    <div className="border-b">
      <div className="flex h-16 items-center px-4">
        <StoreSwitcher items={stores} />
      </div>
    </div>
  )
}
```

-

Menambahkan const baru untuk mengaitkan StoreSwitcher dan menambahkan const baru

```
components > store-switcher.tsx > StoreSwitcher > [currentStore]
1  'use client';
2
3  import { Check, ChevronsUpDown, PlusCircle, StethoscopeIcon, Store as StoreIcon } from "lucide-react";
4  import { useState } from "react";
5  import { Store } from "@prisma/client";
6  import { useParams, useRouter } from "next/navigation";
7
8  import { Popover, PopoverContent, PopoverTrigger } from "@components/ui/popover";
9  import { Button } from "@components/ui/button";
10 import { useStoreModal } from "@hooks/use-store-modal";
11 import { cn } from "@lib/Utils";
12 import { Command, CommandEmpty, CommandGroup, CommandInput, CommandItem, CommandList } from "cmdk";
13
14 type PopoverTriggerProps = React.ComponentPropsWithRef<typeof PopoverTrigger>
15
16 interface StoreSwitcherProps extends PopoverTriggerProps {
17   items: Store[];
18 };
19
20 export default function StoreSwitcher({
21   className,
22   items = []
23 }: StoreSwitcherProps) {
24   const storeModal = useStoreModal();
25   const params = useParams();
26   const router = useRouter();
27
28   const formattedItems = items.map((item) => ({
29     label: item.name,
30     value: item.id
31   }));
32
33   const currentStore = formattedItems.find((item) => item.value === params.storeId);
34
35   const [open, setOpen] = useState(false);
36
37   const onStoreSelect = (store: { value: string, label: string }) => {
```

Mendefinisikan StoreSwitcher didalamnya membuat button took berupa searcing took, nama toko, tambah toko.

Tanggal : 12 Januari 2025

Waktu : 14.44 s/d ...


```
settings-form.tsx M × layout.tsx use-origin.tsx U ×
hooks > use-origin.tsx > useOrigin
1 import { useState, useEffect } from "react";
2
3 export const useOrigin = () => {
4   const [mounted, setMounted] = useState(false);
5   const origin = typeof window !== "undefined" && window.location.origin ? window.location.origin : '';
6
7   useEffect(() => {
8     setMounted(true);
9   }, []);
10
11   if (!mounted) {
12     return '';
13   }
14
15   return origin;
16 }
```

Menambahkan custom React hook bernama useOrigin. Untuk tampilan settings pada admin.

```
27 import { useOrigin } from "@/hooks/use-origin";
28
29
30
31 interface SettingsFormProps {
32   initialData: Store;
33 }
34
35 const formSchema = z.object({
36   name: z.string().min(1)
37 });
38 type SettingsFormValues = z.infer<typeof formSchema>;
39
40 export const SettingsForm: React.FC<SettingsFormProps> =
41   ({ initialData }) => {
42     const params = useParams();
43     const router = useRouter();
44     const origin = useOrigin();
```

Menambahkan const origin dan mengimport dari useOrigin yang sebelumnya dibuat.

```
schema.prisma M X settings-form.tsx
prisma > schema.prisma > Billboard
15 }
16
17 model Store{
18   id String @id @default(uuid())
19   name String
20   userId String
21   billboards Billboard[] @relation("StoreToBillboard")
22   createdAt DateTime @default(now())
23   updatedAt DateTime @updatedAt
24 }
25
26 model Billboard {
27   id String @id @default(uuid())
28   storeId String
29   store Store @relation("StoreToBillboard", fields: [storeId], references: [id])
30   label String
31   imageUrl String
32   createdAt DateTime @default(now())
33   updatedAt DateTime @updatedAt
34
35   @@index([storeId])
36 }
```

• Membuat model baru Billboard dan mendefinisikannya