

Nama : Nurhidayat

NPM : 22312099

Dokumentasi Frontend

Instalasi Shadcn ui

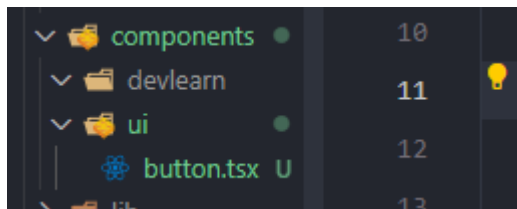
```
PS D:\FOLDER KULIAH\SEMESTER 6\Pemrograman Berorientasi Service\KBizer\frontend> npx shadcn@latest init
>>
Need to install the following packages:
shadcn@2.5.0
Ok to proceed? (y) y
npm WARN deprecated node-domexception@1.0.0: Use your platform's native DOMException instead
✓ Preflight checks.
✓ Verifying framework. Found Next.js.
✓ Validating Tailwind CSS config. Found v4.
✓ Validating import alias.
✓ Which color would you like to use as the base color? » Neutral
✓ Writing components.json.
✓ Checking registry.
✓ Updating CSS variables in app\globals.css
  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).

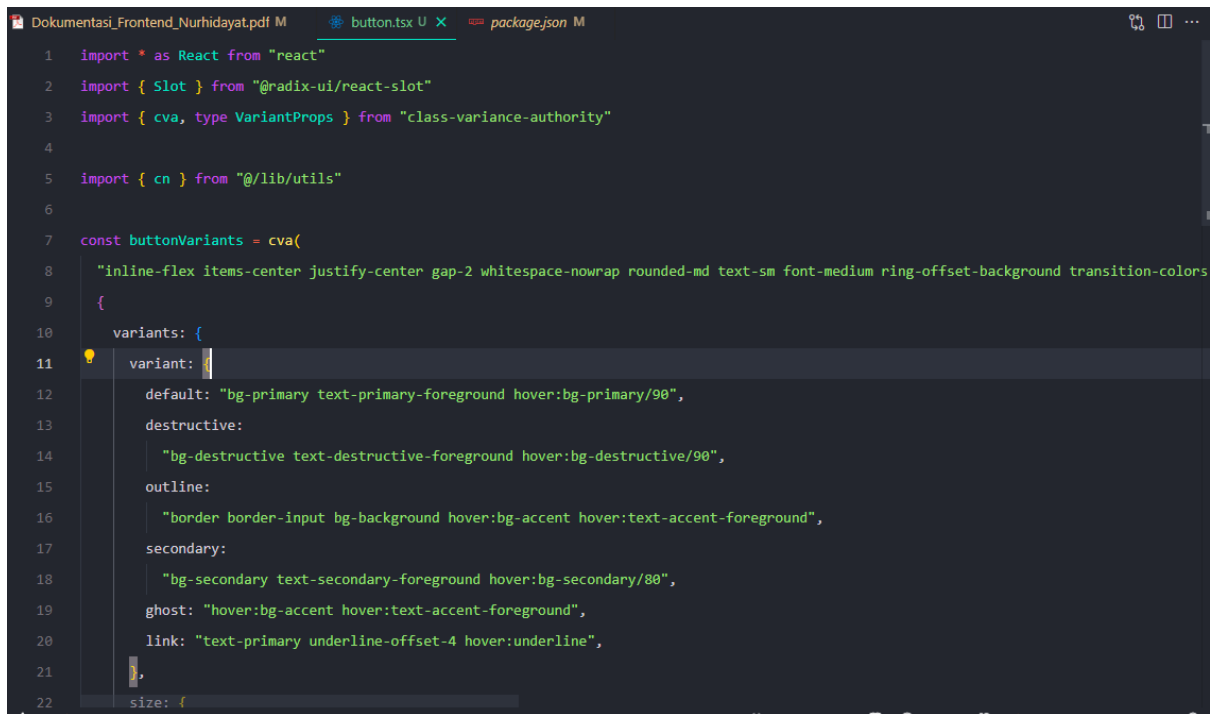
✓ How would you like to proceed? » Use --force
✓ Installing dependencies.
✓ Created 1 file:
  - lib\utils.ts

Success! Project initialization completed.
```

Menambahkan file baru pada folder ui yg ada pada folder components yaitu file button.tsx

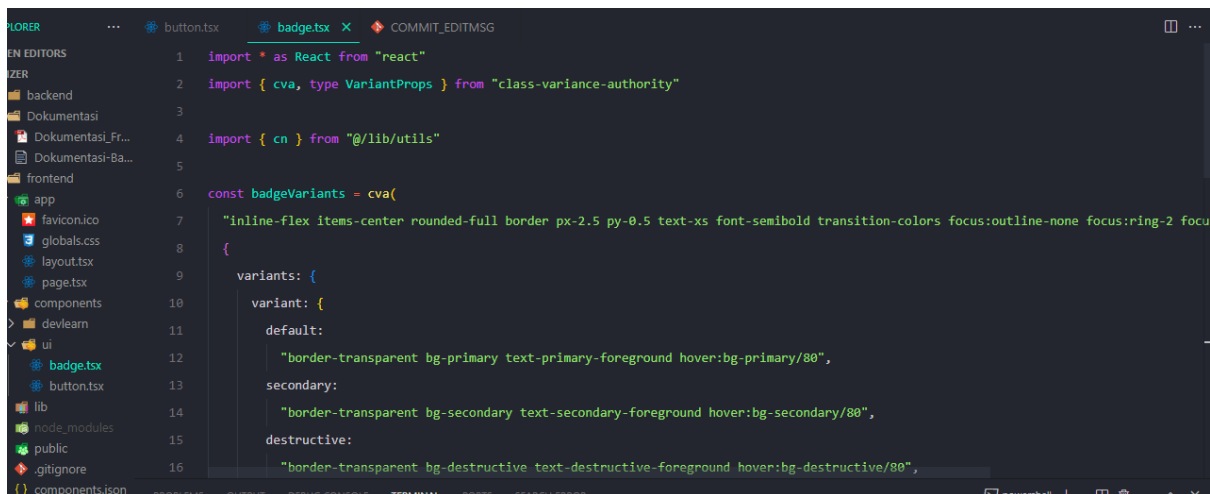


Berikut adalah isi button.tsx



```
1 import * as React from "react"
2 import { Slot } from "@radix-ui/react-slot"
3 import { cva, type VariantProps } from "class-variance-authority"
4
5 import { cn } from "@lib/Utils"
6
7 const buttonVariants = cva(
8   "inline-flex items-center justify-center gap-2 whitespace-nowrap rounded-md text-sm font-medium ring-offset-background transition-colors",
9   {
10     variants: {
11       variant: {
12         default: "bg-primary text-primary-foreground hover:bg-primary/90",
13         destructive: "bg-destructive text-destructive-foreground hover:bg-destructive/90",
14         outline: "border border-input bg-background hover:bg-accent hover:text-accent-foreground",
15         secondary: "bg-secondary text-secondary-foreground hover:bg-secondary/80",
16         ghost: "hover:bg-accent hover:text-accent-foreground",
17         link: "text-primary underline-offset-4 hover:underline",
18       },
19     },
20   },
21 )
22
23 export interface ButtonProps
24   extends React.ComponentPropsWithoutRef,
25     VariantProps {
26   as?: React.ElementType
27 }
28
29 export function Button({
30   className,
31   variant,
32   size,
33   asChild,
34   ...props
35 }: ButtonProps) {
36   const Component = asChild ? Slot : React.createElement
37   return (
38     
```

Menambahkan file badge.tsx



```
1 import * as React from "react"
2 import { cva, type VariantProps } from "class-variance-authority"
3
4 import { cn } from "@lib/Utils"
5
6 const badgeVariants = cva(
7   "inline-flex items-center rounded-full border px-2.5 py-0.5 text-xs font-semibold transition-colors focus:outline-none focus:ring-2 focus:ring-offset-2",
8   {
9     variants: {
10       variant: {
11         default: "border-transparent bg-primary text-primary-foreground hover:bg-primary/80",
12         secondary: "border-transparent bg-secondary text-secondary-foreground hover:bg-secondary/80",
13         destructive: "border-transparent bg-destructive text-destructive-foreground hover:bg-destructive/80",
14       },
15     },
16   },
17 )
18
19 export interface BadgeProps
20   extends React.ComponentPropsWithoutRef,
21     VariantProps {
22   as?: React.ElementType
23 }
24
25 export function Badge({
26   className,
27   variant,
28   asChild,
29   ...props
30 }: BadgeProps) {
31   const Component = asChild ? Slot : React.createElement
32   return (
33     
```

Saya membuat komponen Badge pakai React untuk menampilkan label kecil seperti tag atau status. Supaya bisa dipakai di banyak kondisi, saya tambahkan beberapa pilihan gaya (seperti default, secondary, destructive, dan outline) dengan bantuan cva (class-variance-authority). Jadi kita bisa pilih varian tanpa harus nulis class satu-satu.

Saya juga pakai helper cn buat gabungin class name biar lebih rapi. Komponen ini simpel, fleksibel, dan bisa dikustomisasi lewat props kalau dibutuhkan.

Menambahkan judul pada layout dan mengubah port ke 3002

```
Debug
"scripts": {
  "dev": "next dev --turbo pack -p 3002",
  "build": "next build",
  "start": "next start",
  "lint": "next lint"
```

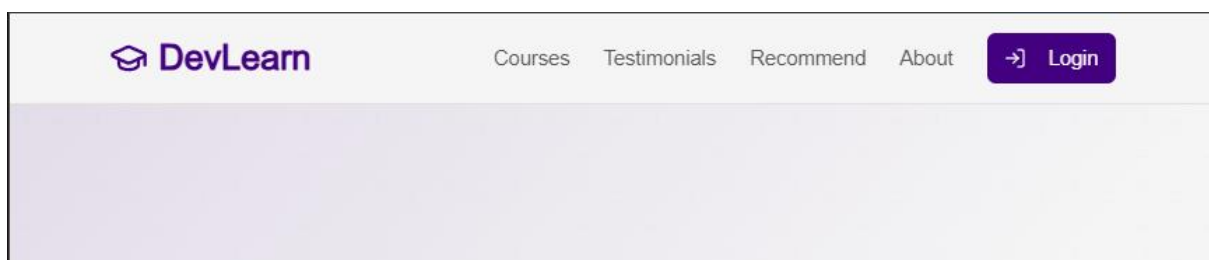
```
14
15 export const metadata: Metadata = {
16   title: "DevLearn - Interactive Learning Platform",
17   description: "Your journey to mastering programming languages starts here. Interactive, AI-powered learning",
18 };
19
```

Menambahkan file header.tsx pada folder devlearn

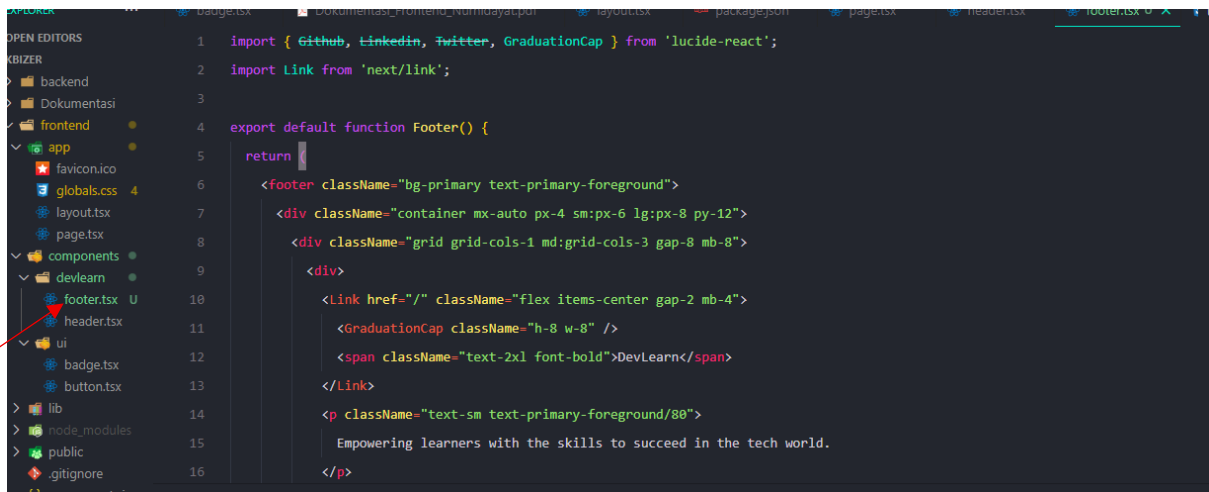
```
OPEN EDITORS
1 import { GraduationCap, Login } from 'lucide-react';
2 import Link from 'next/link';
3 import { Button } from "@components/ui/button";
4
5 export default function Header() {
6   return (
7     <header className="sticky top-0 z-50 w-full border-b bg-background/95 backdrop-blur supports-[backdrop-filter]:bg-background/60">
8       <div className="container mx-auto flex h-16 items-center justify-between px-4 sm:px-6 lg:px-8">
9         <Link href="/" className="flex items-center gap-2">
10           <GraduationCap className="h-7 w-7 text-primary" />
11           <span className="text-2xl font-bold text-primary">DevLearn</span>
12         </Link>
13         <nav className="flex items-center space-x-6">
14           <Link href="#courses-preview" className="text-sm font-medium text-foreground/70 transition-colors hover:text-primary">
15             Courses
16           </Link>
```

Komponen Header ini adalah navigasi utama untuk aplikasi **DevLearn**. Saya buat responsif dan selalu tampil di atas halaman. Bagian kiri berisi logo dan nama aplikasi, sedangkan bagian kanan berisi menu navigasi ke bagian-bagian halaman (Courses, Testimonials, Recommend, About) serta tombol Login. Tampilan dibuat bersih dan modern, dengan ikon dari lucide-react dan tombol dari UI kit internal (Button).

Dan dibawah ini tampilan headernya



Membuat file baru yaitu footer.tsx pada folder devlearn



```
1 import { Github, LinkedIn, Twitter, GraduationCap } from 'lucide-react';
2 import Link from 'next/link';
3
4 export default function Footer() {
5   return (
6     <footer className="bg-primary text-primary-foreground">
7       <div className="container mx-auto px-4 sm:px-6 lg:px-8 py-12">
8         <div className="grid grid-cols-1 md:grid-cols-3 gap-8 mb-8">
9           <div>
10             <Link href="/" className="flex items-center gap-2 mb-4">
11               <GraduationCap className="h-8 w-8" />
12               <span className="text-2xl font-bold">DevLearn</span>
13             </Link>
14             <p className="text-sm text-primary-foreground/80">
15               Empowering learners with the skills to succeed in the tech world.
16             </p>
17           </div>
18         </div>
19       </div>
20     </footer>
21   );
22 }
```

Saya membuat komponen Footer ini sebagai penutup halaman di aplikasi **DevLearn**.
Komponen ini terdiri dari tiga bagian utama:

1. Logo dan Deskripsi

Di bagian kiri, saya tampilkan logo dan nama aplikasi, dilengkapi deskripsi singkat tentang tujuan DevLearn.

2. Quick Links

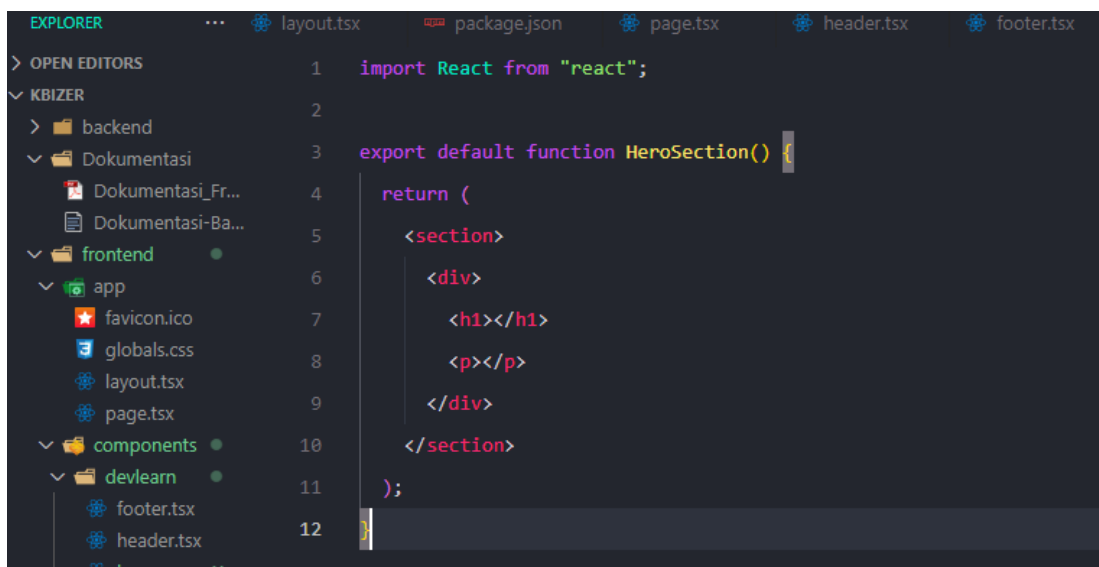
Bagian tengah berisi link navigasi cepat ke halaman penting seperti Courses, About, AI Recommender, dan Contact. Tujuannya biar user lebih mudah akses bagian-bagian utama dari halaman.

3. Sosial Media

Di bagian kanan, saya tambahkan ikon media sosial (Twitter, GitHub, LinkedIn) supaya user bisa terhubung lebih lanjut.

Terakhir, saya tambahkan copyright otomatis berdasarkan tahun saat ini, lengkap dengan pesan personal sebagai penutup.

Membuat struktur dasar komponen pada hero-section



```
1 import React from "react";
2
3 export default function HeroSection() {
4   return (
5     <section>
6       <div>
7         <h1></h1>
8         <p></p>
9       </div>
10     </section>
11   );
12 }
```

Pada tahap ini, saya membuat struktur dasar komponen HeroSection menggunakan React. Komponen ini saya siapkan sebagai bagian utama dari tampilan homepage yang nantinya akan menampilkan judul, deskripsi, tombol, dan ilustrasi. Untuk sementara, saya hanya mendefinisikan elemen `<section>` dan satu `<div>` utama sebagai kerangka awal.

Menambahkan layout dan style background pada hero-section.tsx

```
3 export default function HeroSection() {
4   return (
5     <section className="relative py-20 md:py-32 bg-gradient-to-br from-primary/10 via-background to-background">
6       <div className="absolute inset-0 opacity-5" style={{ backgroundImage: "url('/circuit-board.svg')" }}>
7         <h1></h1>
8         <p></p>
9       </div>
```

Selanjutnya, saya menambahkan styling dasar untuk layout HeroSection. Saya menggunakan kombinasi gradasi warna (`from-primary/10`, `via-background`, dan `to-background`) agar tampilan lebih menarik secara visual. Selain itu, saya tambahkan juga background bergambar `circuit-board (/circuit-board.svg)` dengan opacity rendah agar terlihat seperti watermark bertema teknologi.

Menambahkan heading dan deskripsi pada hero-section.tsx

```
4   return (
5     <section className="relative py-20 md:py-32 bg-gradient-to-br from-primary/10 via-background to-background">
6       <div className="absolute inset-0 opacity-5" style={{ backgroundImage: "url('/circuit-board.svg')" }}>
7         <h1 className="text-4xl sm:text-5xl md:text-6xl font-extrabold tracking-tight text-primary"> Unlock Your <span className="text-accent">Coding Potential</span>
8         <p className="mt-6 max-w-2xl mx-auto text-lg sm:text-xl text-foreground/80"> DevLearn is your interactive gateway to mastering programming languages...</p>
9       </div>
10    </section>
11  );
```

Pada tahap ini, saya menambahkan heading utama (`<h1>`) yang berisi tagline “Unlock Your Coding Potential” dengan penekanan pada kata *Coding Potential* menggunakan warna aksen. Di bawahnya, saya tambahkan paragraf deskripsi singkat mengenai platform DevLearn untuk memberi gambaran singkat kepada pengunjung mengenai manfaat dan fitur yang ditawarkan.

Menambahkan tombol dengan link dan icons pada hero-section.tsx

```
    <Button asChild size="lg" className="bg-accent hover:bg-accent/90 ...">
      <Link href="/courses">
        Explore Courses
      <ArrowRight className="ml-2 h-5 w-5" />
    </Link>
  </Button>

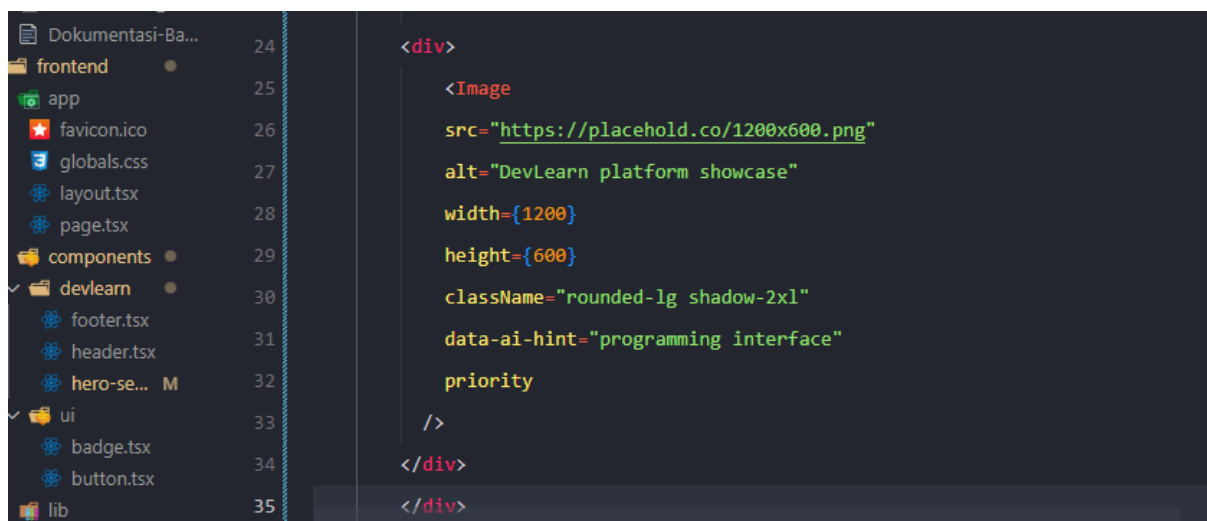
  <Button asChild size="lg" variant="outline" className="...">
    <Link href="#recommendations">Get Recommendations</Link>
  </Button>
</section>
```

Saya menambahkan dua tombol aksi utama untuk mendorong pengguna melakukan eksplorasi lebih lanjut:

- Tombol Explore Courses akan mengarahkan pengguna ke halaman daftar kursus.
- Tombol Get Recommendations akan menggulir halaman ke bagian rekomendasi.

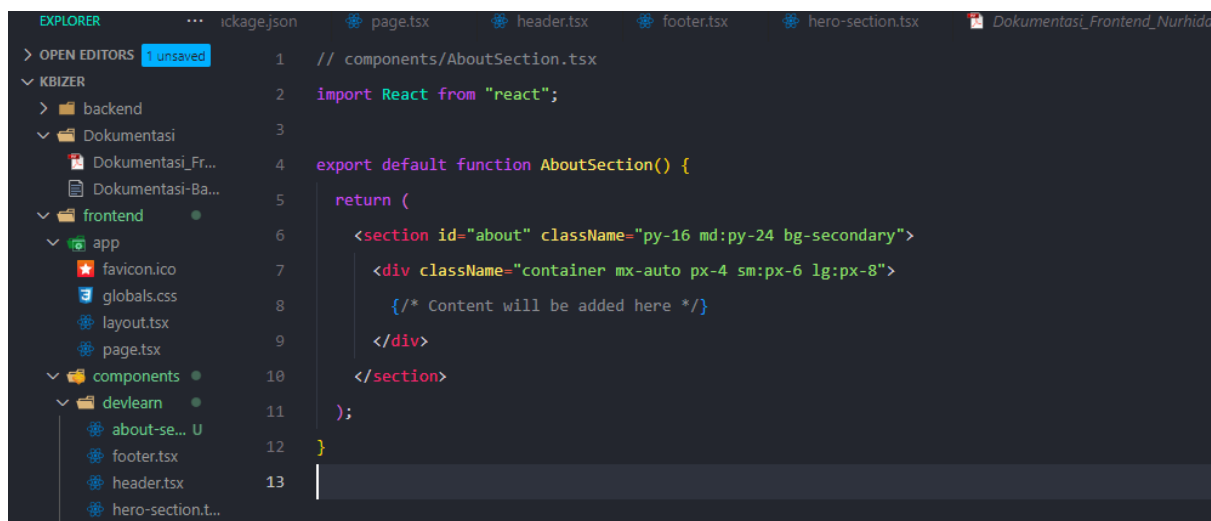
Untuk tombol pertama, saya juga menyertakan ikon ArrowRight dari lucide-react untuk memperkuat arah tindakan. Kedua tombol sudah dilengkapi efek hover dan transisi agar terasa lebih interaktif.

Menambahkan Image showcase di hero-section.tsx



Di bagian bawah HeroSection, saya menambahkan ilustrasi visual dalam bentuk gambar menggunakan komponen Image dari Next.js. Saat ini saya gunakan placeholder dari placeholder.co dengan ukuran 1200x600. Tujuannya adalah untuk menampilkan preview visual dari platform DevLearn sekaligus mempercantik tampilan halaman. Saya juga tambahkan efek shadow-2xl dan rounded-lg untuk hasil akhir yang lebih profesional.

Membuat struktur awal komponen AboutSection



Pada tahap awal ini, saya mulai dengan membuat komponen AboutSection sebagai bagian dari halaman utama website. Saya hanya menambahkan struktur section dasar menggunakan elemen `<section>` dan `<div>` container. Tujuan commit ini adalah untuk menyiapkan kerangka awal tanpa ada konten terlebih dahulu, sehingga memudahkan pengembangan bertahap di commit-commit selanjutnya.

Menambahkan heading dan deskripsi pada Aboutsection

```
<div className="container mx-auto px-4 sm:px-6 lg:px-8">
  { /* heading */ }
  <div className="text-center mb-12">
    <h2 className="text-3xl md:text-4xl font-bold text-primary">About DevLearn</h2>
    <p className="mt-3 text-lg text-foreground/70 max-w-xl mx-auto">
      Empowering the next generation of developers through innovative and accessible learning experiences.
    </p>
  </div>
</div>
```

Setelah membuat strukturnya, saya menambahkan judul utama “About DevLearn” dan deskripsi singkat di bagian atas section. Saya buat agar tampilannya berada di tengah halaman dengan ukuran font yang cukup besar dan warna yang konsisten dengan tema utama. Penambahan ini bertujuan untuk menjelaskan secara ringkas misi umum dari platform DevLearn kepada pengunjung.

Menambahkan gambar dan grid layout dua kolom

```
14      </p>
15    </div>
16    { /* Gambar */ }
17    <div className="grid grid-cols-1 md:grid-cols-2 gap-12 items-center">
18      <div>
19        <Image
20          src="https://placeholder.co/600x400.png"
21          alt="Team collaboration or learning environment"
22          width={600}
23          height={400}
24          className="rounded-lg shadow-xl"
25          data-ai-hint="education technology team"
26        />
27      </div>
28      <div className="space-y-8">
29        { /* Items will be added here */ }
30      </div>
31    </div>
```

Pada tahap ini saya menambahkan grid layout dua kolom, di mana kolom kiri berisi ilustrasi visual menggunakan Image dari Next.js. Saya gunakan gambar placeholder sementara agar nantinya bisa diganti dengan gambar representatif yang menunjukkan suasana belajar atau kolaborasi. Layout ini saya buat responsif agar tetap rapi di berbagai ukuran layar.

Menambahkan blok pertama dengan icon target

```
    </div>
    <div className="space-y-8">
      <div className="flex items-start gap-4">
        <div className="flex-shrink-0 bg-accent text-accent-foreground p-3 rounded-full">
          <Target className="h-6 w-6" />
        </div>
        <div>
          <h3 className="text-xl font-semibold text-primary mb-1">Our Mission</h3>
          <p className="text-foreground/80">
            To provide high-quality, engaging, and personalized programming education to learners worldwide, irre
          </p>
        </div>
      </div>
    </div>
```

Saya menambahkan blok pertama dalam bentuk teks dengan ikon yaitu "Our Mission". Di sini saya pakai ikon Target dari lucide-react untuk memperkuat identitas misi DevLearn. Misinya saya tuliskan dengan pendekatan inklusif, yaitu menyediakan pendidikan pemrograman yang menarik dan mudah diakses oleh semua kalangan.

Menambahkan blok kedua dengan icon lightbulb

```
    <div className="flex items-start gap-4">
      <div className="flex-shrink-0 bg-accent text-accent-foreground p-3 rounded-full">
        <Lightbulb className="h-6 w-6" />
      </div>
      <div>
        <h3 className="text-xl font-semibold text-primary mb-1">Our Vision</h3>
        <p className="text-foreground/80">
          To be the leading platform for interactive coding education, fostering a global community of skilled an
        </p>
      </div>
    </div>
```

Berikutnya saya tambahkan blok kedua yaitu "Our Vision", menggunakan ikon Lightbulb sebagai simbol inovasi dan ide. Visi ini menjelaskan tujuan jangka panjang DevLearn sebagai platform pembelajaran interaktif yang membentuk komunitas global developer yang percaya diri dan berkompeten.

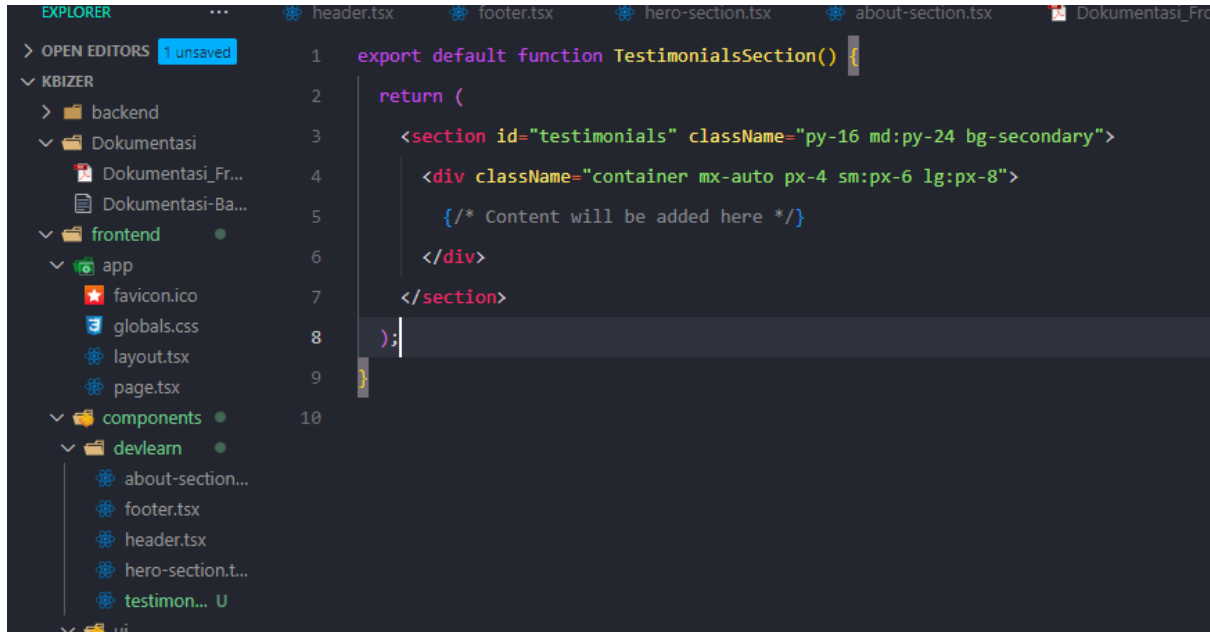
Menambahkan blok terakhir dengan icon Gem

```
    <div className="flex-shrink-0 bg-accent text-accent-foreground p-3 rounded-full">
      <Gem className="h-6 w-6" />
    </div>
    <div>
      <h3 className="text-xl font-semibold text-primary mb-1">Our Values</h3>
      <p className="text-foreground/80">
        Innovation, Accessibility, Community, and a Passion for Continuous Learning. We believe in empowering i
      </p>
    </div>
  </div>
```

Komitmen nilai-nilai inti dari DevLearn saya tuangkan di bagian "Our Values", lengkap dengan ikon Gem sebagai simbol kualitas dan nilai. Saya uraikan nilai-nilai seperti inovasi,

aksesibilitas, komunitas, dan semangat belajar sepanjang hayat. Ini merepresentasikan budaya belajar yang saya ingin tanamkan melalui platform.

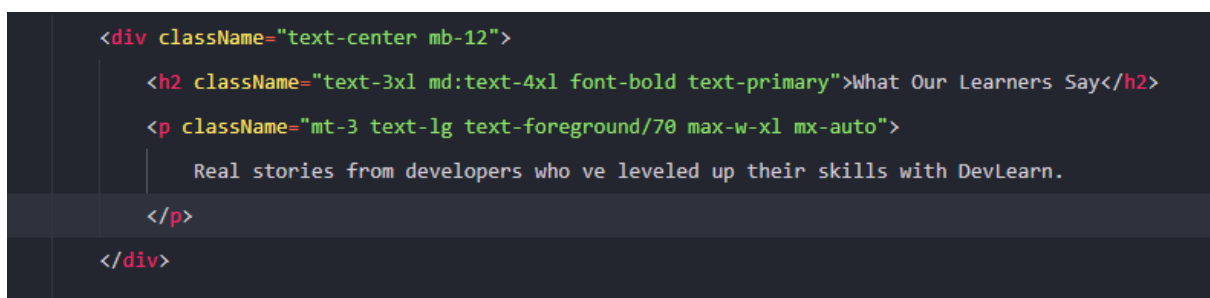
Membuat struktur awal komponen TestimonialsSection

A screenshot of the Visual Studio Code editor. The Explorer sidebar on the left shows a project structure with folders like 'backend', 'Dokumentasi', 'frontend', 'components', and 'devlearn'. The 'TestimonialsSection' component is highlighted. The main editor area shows the code for 'TestimonialsSection.tsx'. The code defines an export default function 'TestimonialsSection()' which returns a JSX element. The JSX element consists of a 'section' with id 'testimonials' and className 'py-16 md:py-24 bg-secondary'. Inside the section is a 'div' with className 'container mx-auto px-4 sm:px-6 lg:px-8'. Inside this div is a comment '/* Content will be added here */'. The code is as follows:

```
1 export default function TestimonialsSection() {
2   return (
3     <section id="testimonials" className="py-16 md:py-24 bg-secondary">
4       <div className="container mx-auto px-4 sm:px-6 lg:px-8">
5         /* Content will be added here */
6       </div>
7     </section>
8   );
9 }
```

Pada tahap awal ini, saya mulai membangun komponen TestimonialsSection yang nantinya akan menampilkan testimoni pengguna. Saya menyiapkan struktur dasar menggunakan elemen `<section>` dan container dengan padding yang responsif. Bagian ini masih kosong dan akan dikembangkan secara bertahap di commit selanjutnya.

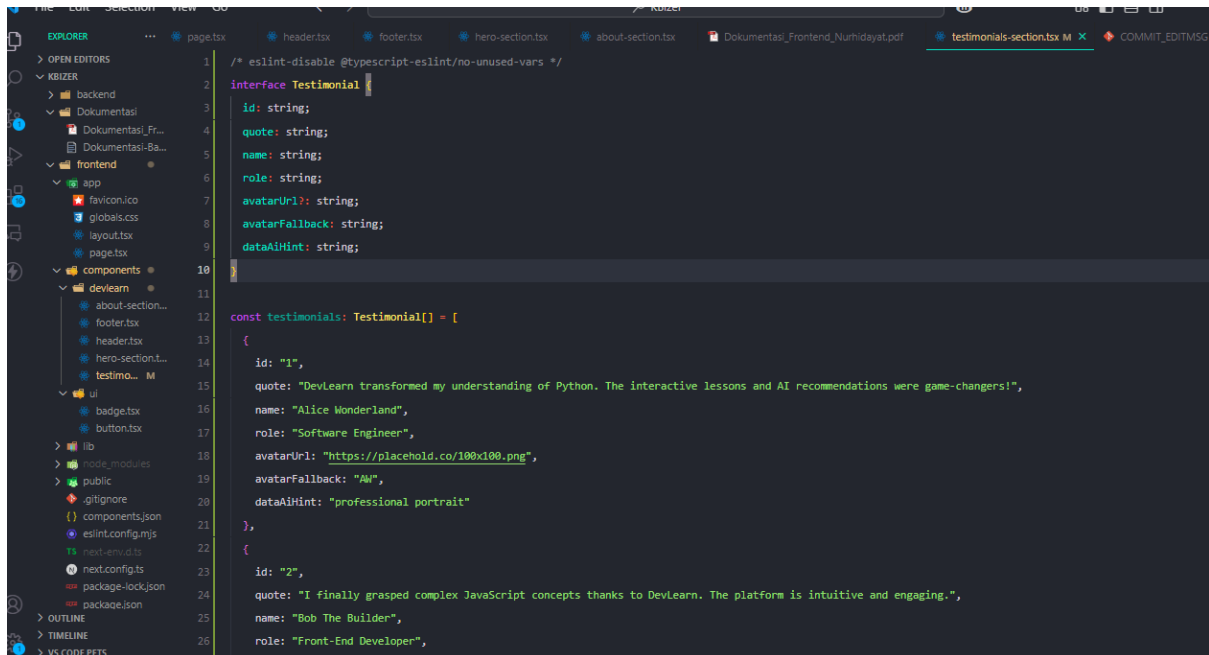
Menambahkan judul dan deskripsi TestimonialsSection

A screenshot of the Visual Studio Code editor showing the updated code for 'TestimonialsSection.tsx'. The code now includes a heading and a paragraph inside the container div. The code is as follows:

```
<div className="text-center mb-12">
  <h2 className="text-3xl md:text-4xl font-bold text-primary">What Our Learners Say</h2>
  <p className="mt-3 text-lg text-foreground/70 max-w-xl mx-auto">
    Real stories from developers who ve leveled up their skills with DevLearn.
  </p>
</div>
```

Di langkah kedua ini, saya menambahkan heading utama dan deskripsi singkat untuk menjelaskan isi dari section ini. Tujuannya agar pengguna langsung memahami bahwa bagian ini berisi testimoni dari para pengguna DevLearn. Saya atur agar tampilannya berada di tengah halaman dan tetap terbaca dengan nyaman di berbagai ukuran layar.

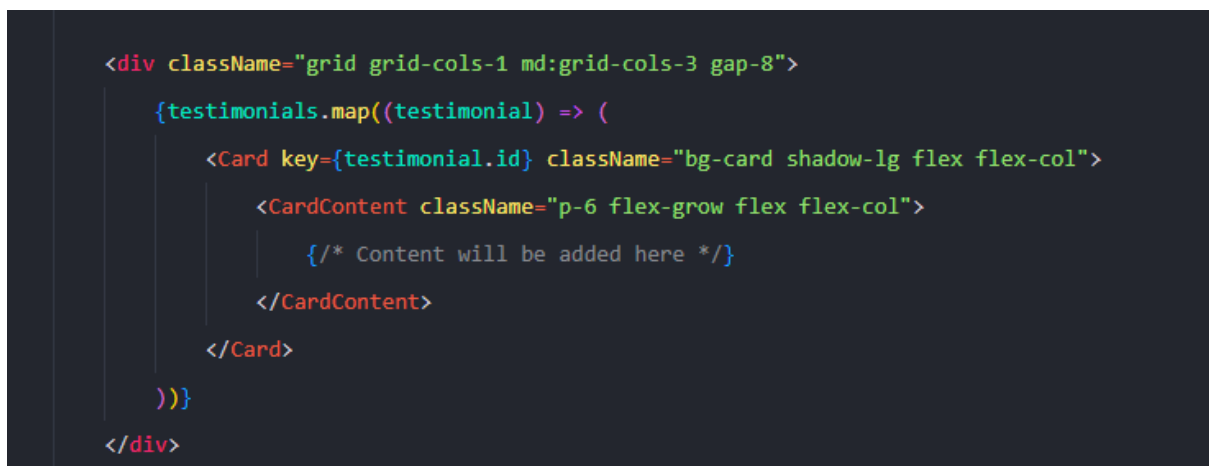
Menambahkan data testimoni dalam bentuk array

A screenshot of a VS Code editor window. The Explorer sidebar on the left shows a project structure with folders like 'backend', 'frontend', and 'components'. The main editor area shows a TypeScript file named 'testimonials-section.tsx'. It contains an interface 'Testimonial' with fields: 'id: string;', 'quote: string;', 'name: string;', 'role: string;', 'avatarUrl?: string;', 'avatarFallback: string;', and 'dataAiHint: string;'. Below the interface, there is a constant 'testimonials: Testimonial[]' initialized with an array of two objects. The first object has id '1', a quote about DevLearn, name 'Alice Wonderland', role 'Software Engineer', and an avatar URL. The second object has id '2', a quote about JavaScript, name 'Bob The Builder', role 'Front-End Developer', and no avatar URL.

```
1 /* eslint-disable @typescript-eslint/no-unused-vars */
2 interface Testimonial {
3   id: string;
4   quote: string;
5   name: string;
6   role: string;
7   avatarUrl?: string;
8   avatarFallback: string;
9   dataAiHint: string;
10 }
11
12 const testimonials: Testimonial[] = [
13   {
14     id: "1",
15     quote: "DevLearn transformed my understanding of Python. The interactive lessons and AI recommendations were game-changers!",
16     name: "Alice Wonderland",
17     role: "Software Engineer",
18     avatarUrl: "https://placeholder.co/100x100.png",
19     avatarFallback: "AI",
20     dataAiHint: "Professional portrait"
21   },
22   {
23     id: "2",
24     quote: "I finally grasped complex JavaScript concepts thanks to DevLearn. The platform is intuitive and engaging.",
25     name: "Bob The Builder",
26     role: "Front-End Developer",
```

Supaya lebih mudah mengelola data testimoni, saya membuat array berisi objek-objek testimonial lengkap dengan nama, kutipan, peran pengguna, dan gambar avatar (atau fallback-nya). Ini juga memungkinkan komponen untuk di-render secara dinamis menggunakan fungsi `.map()` nantinya.

Menambahkan layout kartu testimonial menggunakan grid

A screenshot of a VS Code editor window showing JSX code. The code is a single line of JSX using the `testimonials.map()` function to render a grid of cards. The `div` has a `className` of 'grid grid-cols-1 md:grid-cols-3 gap-8'. The `map` function takes each testimonial and renders a `Card` component. The `Card` has a `key` of `{testimonial.id}` and a `className` of 'bg-card shadow-lg flex flex-col'. Inside the `Card`, there is a `CardContent` component with a `className` of 'p-6 flex-grow flex flex-col' and a comment '/* Content will be added here */'.

```
<div className="grid grid-cols-1 md:grid-cols-3 gap-8">
  {testimonials.map((testimonial) => (
    <Card key={testimonial.id} className="bg-card shadow-lg flex flex-col">
      <CardContent className="p-6 flex-grow flex flex-col">
        /* Content will be added here */
      </CardContent>
    </Card>
  ))}
</div>
```

Saya mulai merender data testimonial ke dalam UI menggunakan komponen Card dari ShadCN. Masing-masing testimonial ditampilkan dalam layout grid 3 kolom agar tampilannya rapi dan responsif. Ini menjadi dasar visual dari setiap kartu testimonial.

Menampilkan isi kartu testimonial (quote, avatar, dan nama)

```
<div className="grid grid-cols-1 md:grid-cols-3 gap-8">
  {testimonials.map((testimonial) => {
    <Card key={testimonial.id} className="bg-card shadow-lg flex flex-col">
      <CardContent className="p-6 flex-grow flex flex-col">
        <Quote className="h-8 w-8 text-accent mb-4" />
        <p className="text-foreground/80 italic mb-6 flex-grow">{testimonial.quote}</p>
        <div className="flex items-center mt-auto">
          <Avatar className="h-12 w-12 mr-4">
            <AvatarImage src={testimonial.avatarUrl} alt={testimonial.name} data-ai-hint={testimonial.dataAiH} />
            <AvatarFallback>{testimonial.avatarFallback}</AvatarFallback>
          </Avatar>
          <div>
            <p className="font-semibold text-primary">{testimonial.name}</p>
            <p className="text-sm text-muted-foreground">{testimonial.role}</p>
          </div>
        </div>
      </CardContent>
    </Card>
  })}
```

Saya menyempurnakan tampilan isi setiap kartu dengan menambahkan kutipan testimonial, ikon quote dari lucide-react, serta bagian avatar pengguna. Saya juga tambahkan nama dan peran masing-masing pengguna agar terlihat lebih personal dan meyakinkan bagi pengunjung lain.

Membuat card.tsx pada folder ui

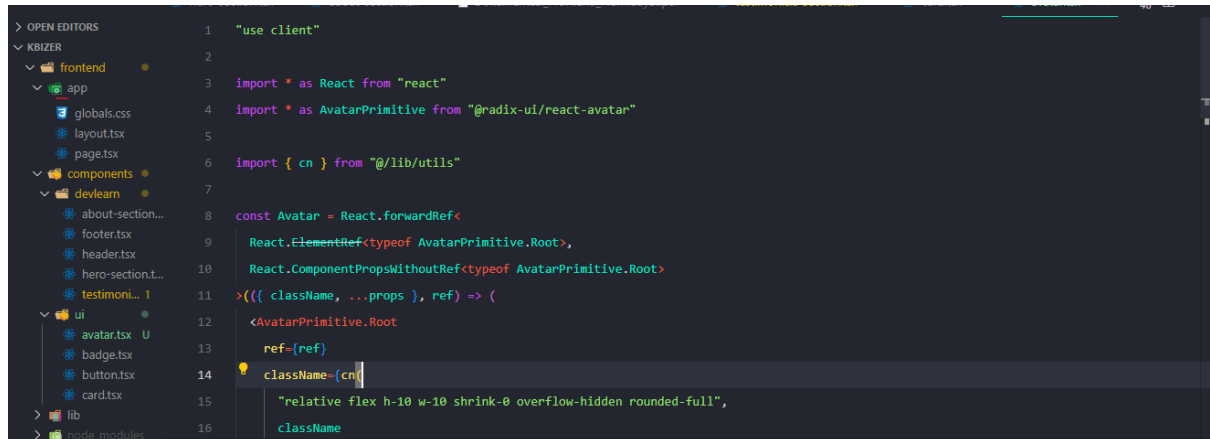
```
EXPLORER    ...    footer.tsx    hero-section.tsx    about-section.tsx    Dokumentasi_Frontend_Nurhidayat.pdf    testimonials-se
> OPEN EDITORS
v KBIZER
  v frontend
    v app
      globals.css
      layout.tsx
      page.tsx
    v components
      v devlearn
        about-section...
        footer.tsx
        header.tsx
        hero-section.t...
        testim... 2, M
      v ui
        badge.tsx
        button.tsx
        card.tsx U
      v lib
      v node_modules
      v public

1  import * as React from "react"
2
3  import { cn } from "@lib/utils"
4
5  const Card = React.forwardRef<
6    HTMLDivElement,
7    React.HTMLAttributes<HTMLDivElement>
8  >(({ className, ...props }, ref) => {
9    <div
10      ref={ref}
11      className={cn(
12        "rounded-lg border bg-card text-card-foreground shadow-sm",
13        className
14      )}
15      {...props}
16    />
```

File card.tsx ini adalah **komponen UI reusable** (bisa dipakai ulang) yang saya buat untuk membangun tampilan **kartu (card)** di aplikasi. Kartu ini biasanya digunakan untuk

membungkus konten seperti gambar, teks, testimonial, artikel, dsb. Desainnya konsisten dan sudah siap pakai.

Membuat file avatar.tsx pada folder ui

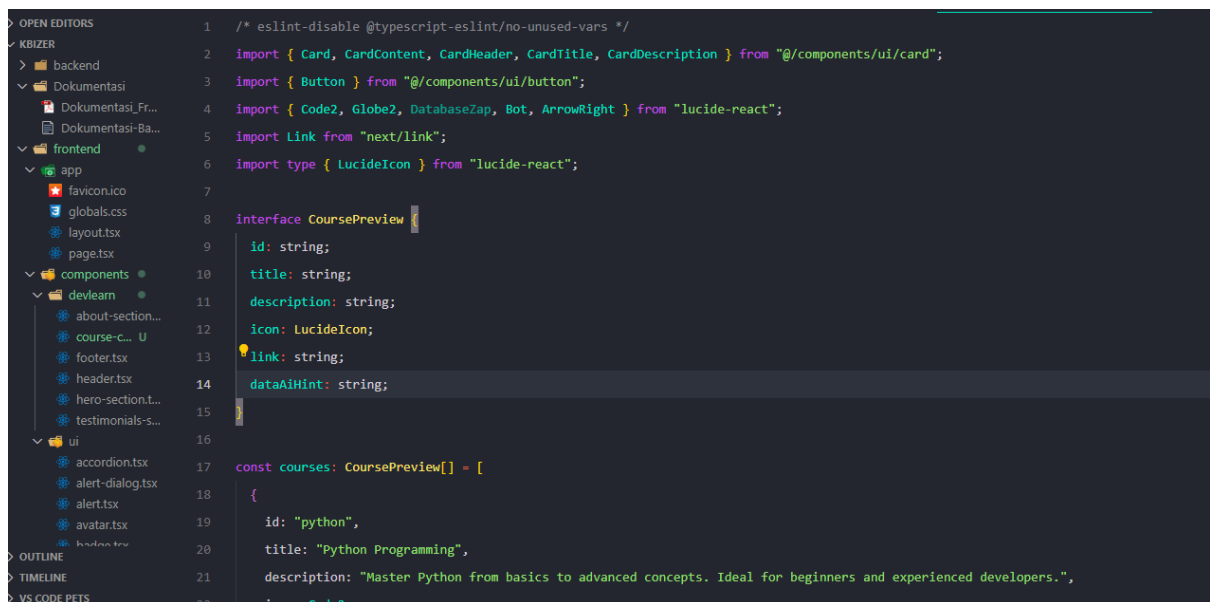


```
1 "use client"
2
3 import * as React from "react"
4 import * as AvatarPrimitive from "@radix-ui/react-avatar"
5
6 import { cn } from "@lib/utils"
7
8 const Avatar = React.forwardRef<
9   React.ElementRef<typeof AvatarPrimitive.Root>,
10   React.ComponentPropsWithoutRef<typeof AvatarPrimitive.Root>
11 >((( className, ...props ), ref) => (
12   <AvatarPrimitive.Root
13     ref={ref}
14     className={cn(
15       "relative flex h-10 w-10 shrink-0 overflow-hidden rounded-full",
16       className
```

Komponen ini saya buat untuk menyederhanakan penggunaan **avatar** (foto profil) dalam berbagai bagian aplikasi, seperti di halaman testimoni, komentar, atau user profile.

Komponen ini dibangun di atas library **Radix UI Avatar** dan sudah saya sesuaikan dengan style Tailwind agar konsisten dengan tema desain proyek.

Membuat tampilan katalog kursus



```
1 /* eslint-disable @typescript-eslint/no-unused-vars */
2 import { Card, CardContent, CardHeader, CardTitle, CardDescription } from "@components/ui/card";
3 import { Button } from "@components/ui/button";
4 import { Code2, Globe2, DatabaseZap, Bot, ArrowRight } from "lucide-react";
5 import Link from "next/link";
6 import type { LucideIcon } from "lucide-react";
7
8 interface CoursePreview {
9   id: string;
10   title: string;
11   description: string;
12   icon: LucideIcon;
13   link: string;
14   dataAiHint: string;
15 }
16
17 const courses: CoursePreview[] = [
18   {
19     id: "python",
20     title: "Python Programming",
21     description: "Master Python from basics to advanced concepts. Ideal for beginners and experienced developers.",
22     icon: Code2
```

Saya membuat komponen **CourseCatalogPreview** untuk menampilkan daftar kursus di aplikasi **DevLearn**.

Setiap kursus ditampilkan dalam bentuk **Card** yang berisi:

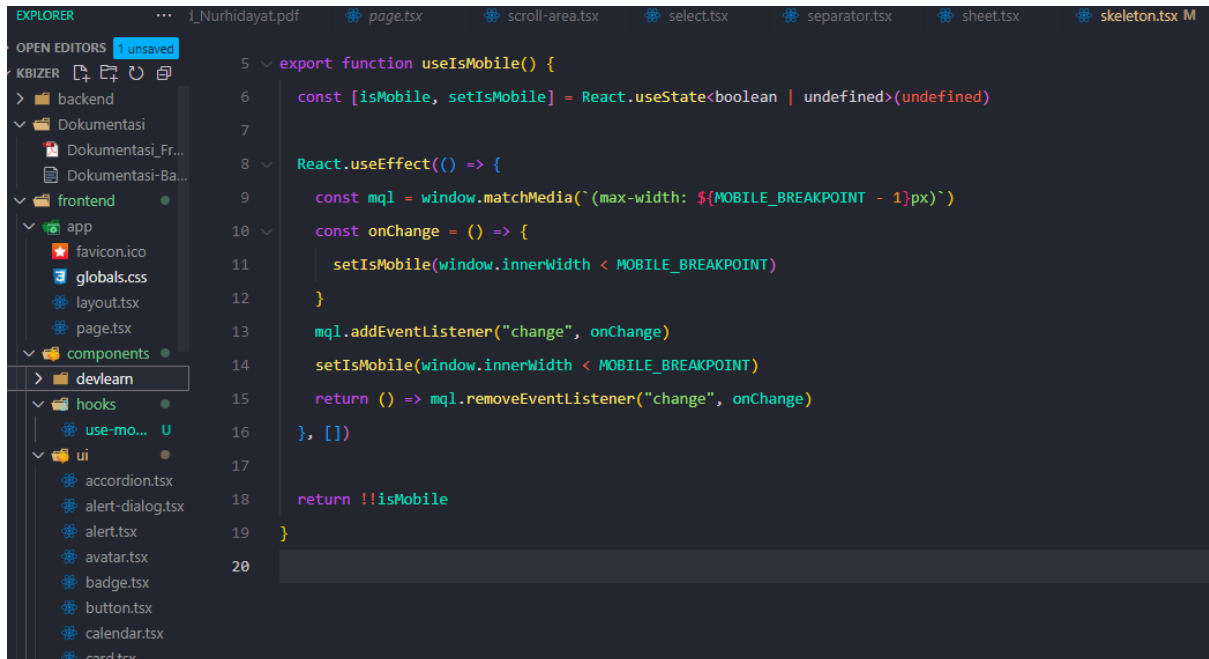
- Icon kursus
- Judul dan deskripsi singkat

- Tombol "Learn More" yang mengarah ke halaman kursus

Tampilannya responsif dan ada animasi saat card disentuh (hover).

Tujuan saya membuat komponen ini adalah agar user bisa melihat dan memilih kursus dengan cepat dan mudah.

Membuat folder hooks dan menambahkan file use-mobile.tsx didalamnya



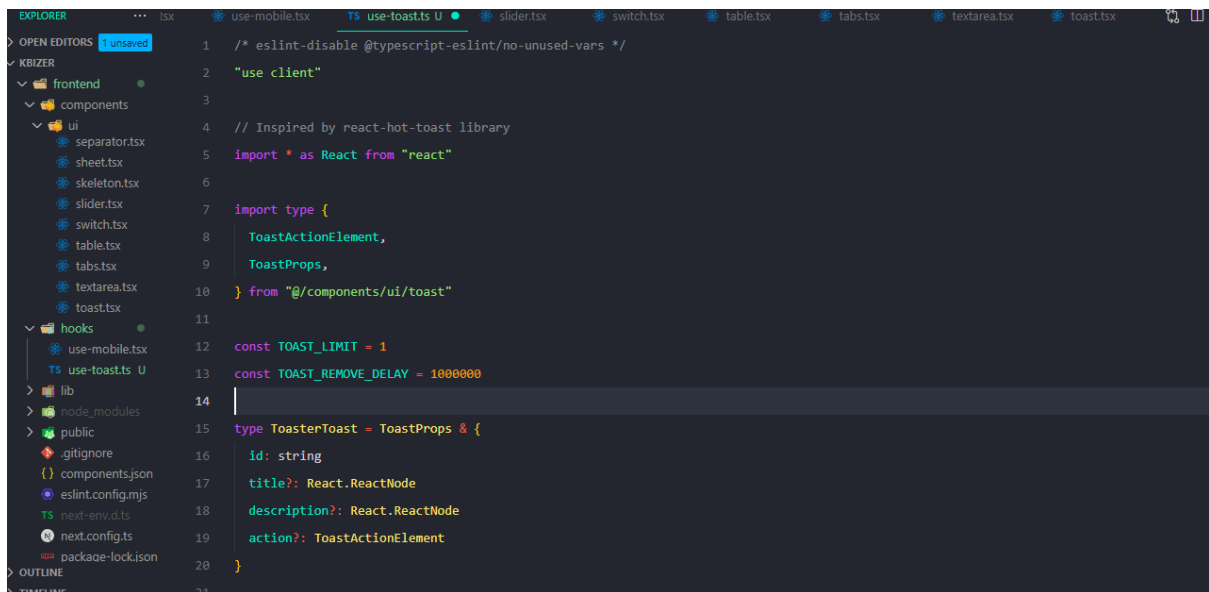
The screenshot shows the VS Code interface. On the left, the Explorer sidebar shows a project structure with folders like 'backend', 'Dokumentasi', 'frontend', 'app', 'components', and 'devlearn'. Inside 'devlearn', there is a 'hooks' folder which is currently selected. The main editor area shows the content of 'use-mobile.tsx'. The code defines a custom hook 'useIsMobile' that uses 'useState' to manage a boolean state 'isMobile' and 'useEffect' to listen for window resize events. It returns '!!isMobile'.

```
5 export function useIsMobile() {
6   const [isMobile, setIsMobile] = React.useState<boolean | undefined>(undefined)
7
8   React.useEffect(() => {
9     const mql = window.matchMedia(`(max-width: ${MOBILE_BREAKPOINT - 1}px)`)
10    const onChange = () => {
11      setIsMobile(window.innerWidth < MOBILE_BREAKPOINT)
12    }
13    mql.addEventListener("change", onChange)
14    setIsMobile(window.innerWidth < MOBILE_BREAKPOINT)
15    return () => mql.removeEventListener("change", onChange)
16  }, [])
17
18  return !!isMobile
19 }
20
```

Folder **hooks** saya buat untuk menyimpan custom hook yang berisi fungsi-fungsi reusable agar bisa digunakan di banyak komponen tanpa menulis ulang logika yang sama.

Pada file use-mobile.tsx, saya membuat custom hook untuk mengecek apakah user sedang memakai perangkat mobile atau bukan. Saya tetapkan ukuran mobile maksimal di **768px**. Hook ini memantau lebar layar secara otomatis dan akan memberikan hasil true jika layar lebih kecil dari 768px. Dengan hook ini, saya bisa dengan mudah menyesuaikan tampilan sesuai ukuran perangkat tanpa perlu cek layar di setiap komponen.

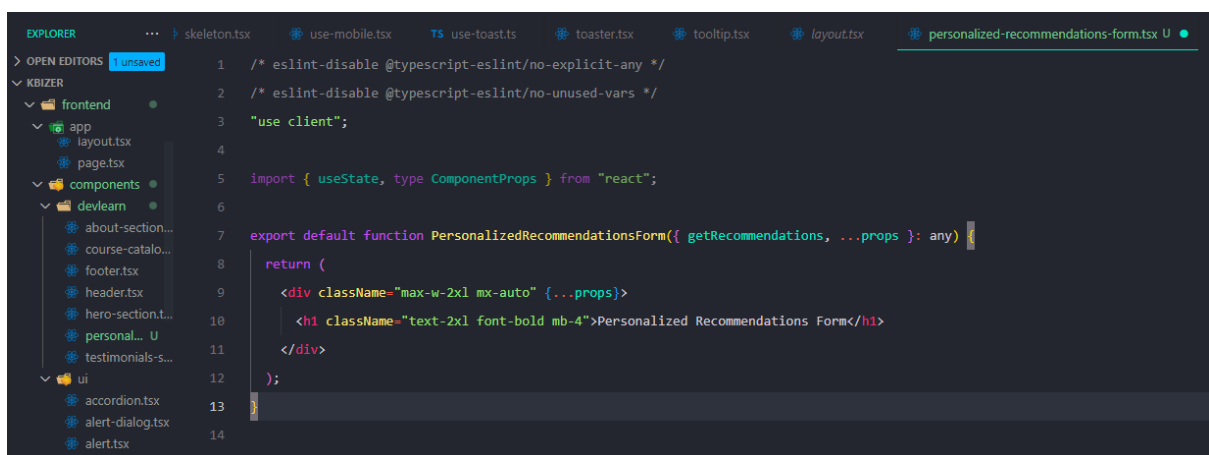
Membuat file use-toast.tsx pada folder hooks



```
1  /* eslint-disable @typescript-eslint/no-unused-vars */
2  "use client"
3
4  // Inspired by react-hot-toast library
5  import * as React from "react"
6
7  import type {
8    ToastActionElement,
9    ToastProps,
10  } from "@components/ui/toast"
11
12  const TOAST_LIMIT = 1
13  const TOAST_REMOVE_DELAY = 1000000
14
15  type ToasterToast = ToastProps & {
16    id: string
17    title?: React.ReactNode
18    description?: React.ReactNode
19    action?: ToastActionElement
20  }
```

Pada file use-toast.tsx, saya membuat custom hook untuk menampilkan notifikasi toast di aplikasi. Fungsinya adalah untuk menambah, mengupdate, dan menghapus toast. Saya batasi supaya hanya **satu toast** yang bisa muncul di layar dalam satu waktu. Toast akan otomatis hilang setelah beberapa waktu. Saya juga buat sistem sederhana agar toast bisa dipakai di mana saja dalam aplikasi tanpa setting tambahan. Dengan hook ini, saya bisa menampilkan notifikasi dengan cara yang mudah dan cepat.

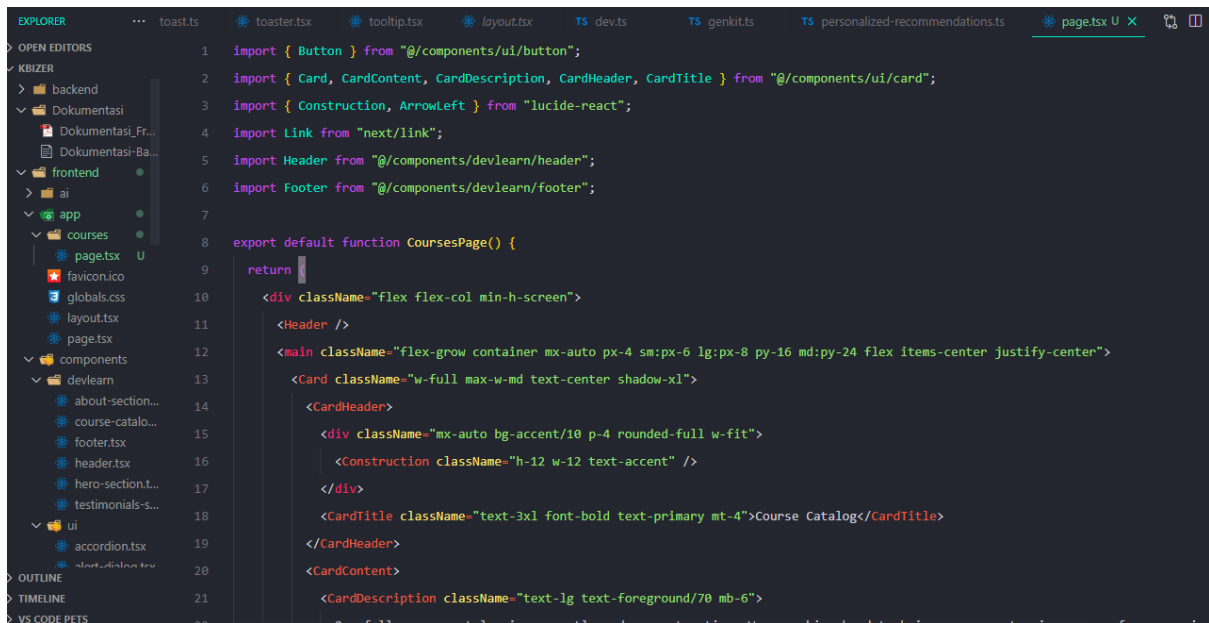
Setup awal komponen form rekomendasi



```
1  /* eslint-disable @typescript-eslint/no-explicit-any */
2  /* eslint-disable @typescript-eslint/no-unused-vars */
3  "use client";
4
5  import { useState, type ComponentProps } from "react";
6
7  export default function PersonalizedRecommendationsForm({ getRecommendations, ...props }: any) {
8    return (
9      <div className="max-w-2xl mx-auto" {...props}>
10        <h1 className="text-2xl font-bold mb-4">Personalized Recommendations Form</h1>
11      </div>
12    );
13  }
```

Saya membuat komponen dasar untuk form rekomendasi. Di sini saya hanya menyiapkan struktur komponen kosong yang nanti akan saya isi dengan form input.

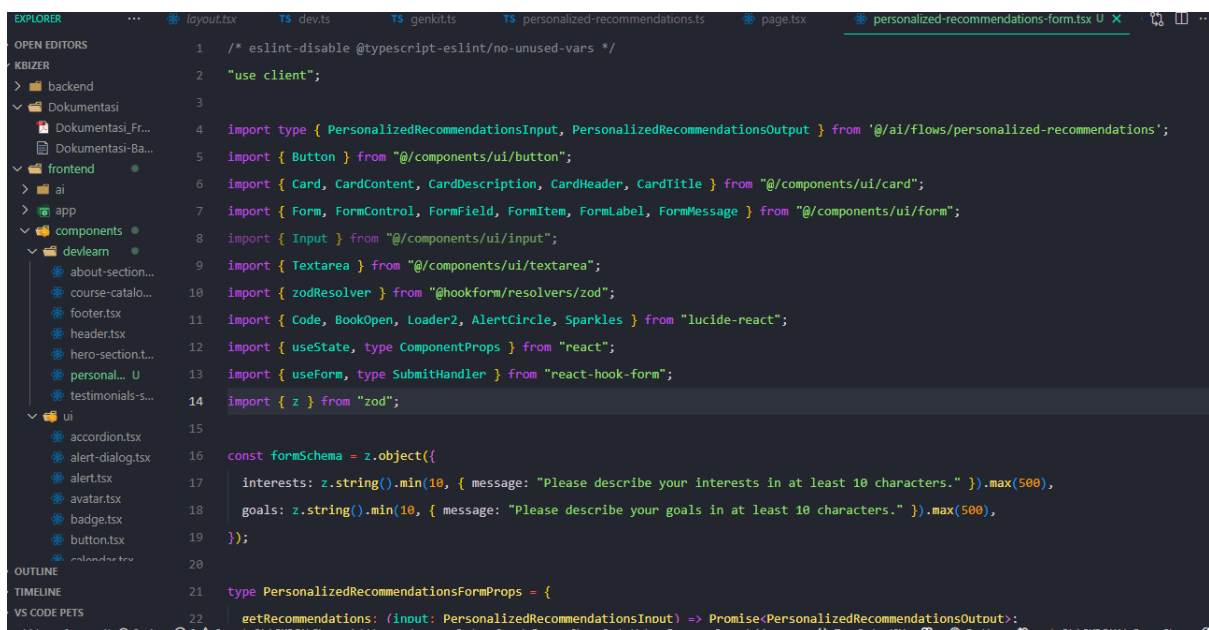
Membuat folder baru courses dan page.tsx didalamnya



```
1 import { Button } from "@components/ui/button";
2 import { Card, CardContent, CardDescription, CardHeader, CardTitle } from "@components/ui/card";
3 import { Construction, ArrowLeft } from "lucide-react";
4 import Link from "next/link";
5 import Header from "@components/devlearn/header";
6 import Footer from "@components/devlearn/footer";
7
8 export default function CoursesPage() {
9   return (
10     <div className="flex flex-col min-h-screen">
11       <Header />
12       <main className="flex-grow container mx-auto px-4 sm:px-6 lg:px-8 py-16 md:py-24 flex items-center justify-center">
13         <Card className="w-full max-w-md text-center shadow-xl">
14           <CardHeader>
15             <div className="mx-auto bg-accent/10 p-4 rounded-full w-fit">
16               <Construction className="h-12 w-12 text-accent" />
17             </div>
18             <CardTitle className="text-3xl font-bold text-primary mt-4">Course Catalog</CardTitle>
19           </CardHeader>
20           <CardContent>
21             <CardDescription className="text-lg text-foreground/70 mb-6">
```

Di halaman ini saya membuat tampilan **Course Catalog** yang sementara masih dalam tahap pembangunan. Saya menampilkan pesan bahwa daftar kursus sedang disiapkan dan belum bisa diakses saat ini. Pada halaman ini saya juga menambahkan tombol **Go Back Home** agar user bisa dengan mudah kembali ke halaman utama. Saya menggunakan komponen Header dan Footer agar tampilannya tetap konsisten dengan halaman lainnya. Tampilan halaman ini saya buat dengan **Card** dan **ikon under construction** supaya user paham kalau fitur ini memang belum tersedia dan sedang dalam proses pengembangan.

Membuat Form Rekomendasi Belajar

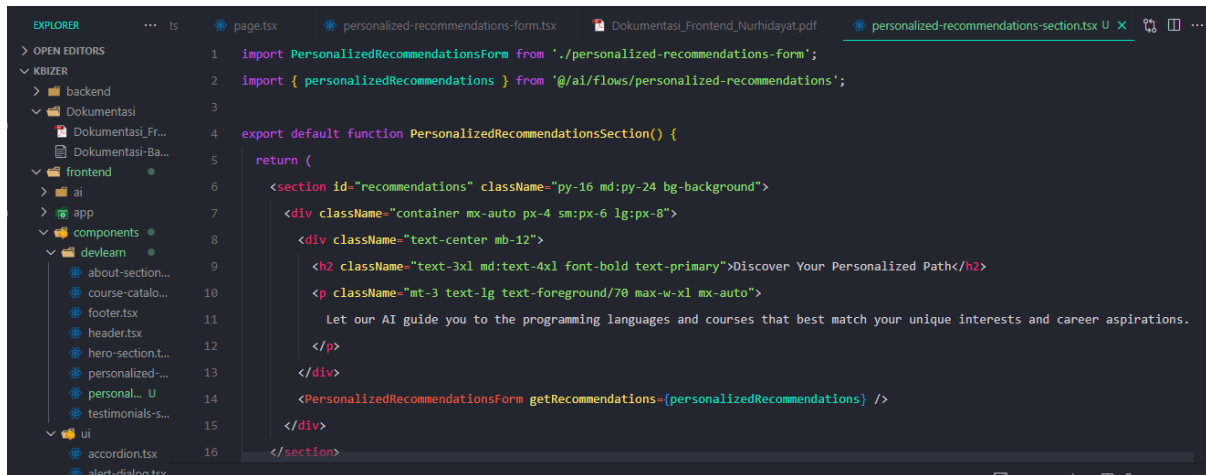


```
1 /* eslint-disable @typescript-eslint/no-unused-vars */
2 "use client";
3
4 import type { PersonalizedRecommendationsInput, PersonalizedRecommendationsOutput } from '@ai/flows/personalized-recommendations';
5 import { Button } from "@components/ui/button";
6 import { Card, CardContent, CardDescription, CardHeader, CardTitle } from "@components/ui/card";
7 import { Form, FormControl, FormField, FormItem, FormLabel, FormMessage } from "@components/ui/form";
8 import { Input } from "@components/ui/input";
9 import { Textarea } from "@components/ui/textarea";
10 import { zodResolver } from "@hookform/resolvers/zod";
11 import { Code, BookOpen, Loader2, AlertCircle, Sparkles } from "lucide-react";
12 import { useState, type ComponentProps } from "react";
13 import { useForm, type SubmitHandler } from "react-hook-form";
14 import { z } from "zod";
15
16 const formSchema = z.object({
17   interests: z.string().min(10, { message: "Please describe your interests in at least 10 characters." }).max(500),
18   goals: z.string().min(10, { message: "Please describe your goals in at least 10 characters." }).max(500),
19 });
20
21 type PersonalizedRecommendationsFormProps = {
22   getRecommendations: (input: PersonalizedRecommendationsInput) => Promise<PersonalizedRecommendationsOutput>;
23 }
```

Saya membuat form rekomendasi belajar yang memungkinkan user mengisi minat dan tujuan mereka. Form ini sudah divalidasi agar input minimal 10 karakter supaya data yang dikirim

cukup jelas. Setelah user mengisi dan menekan tombol, sistem akan memproses dan menampilkan rekomendasi bahasa pemrograman dan kursus yang cocok. Jika proses gagal, akan muncul pesan error agar user tahu harus mencoba lagi. Saat form sedang dikirim, tombol akan berubah menjadi loading agar user tahu proses berjalan. Saya menggunakan React, React Hook Form, Zod untuk validasi, dan Lucide untuk ikon. Fitur ini saya buat agar user lebih mudah menemukan jalur belajar yang sesuai dengan kebutuhan mereka.

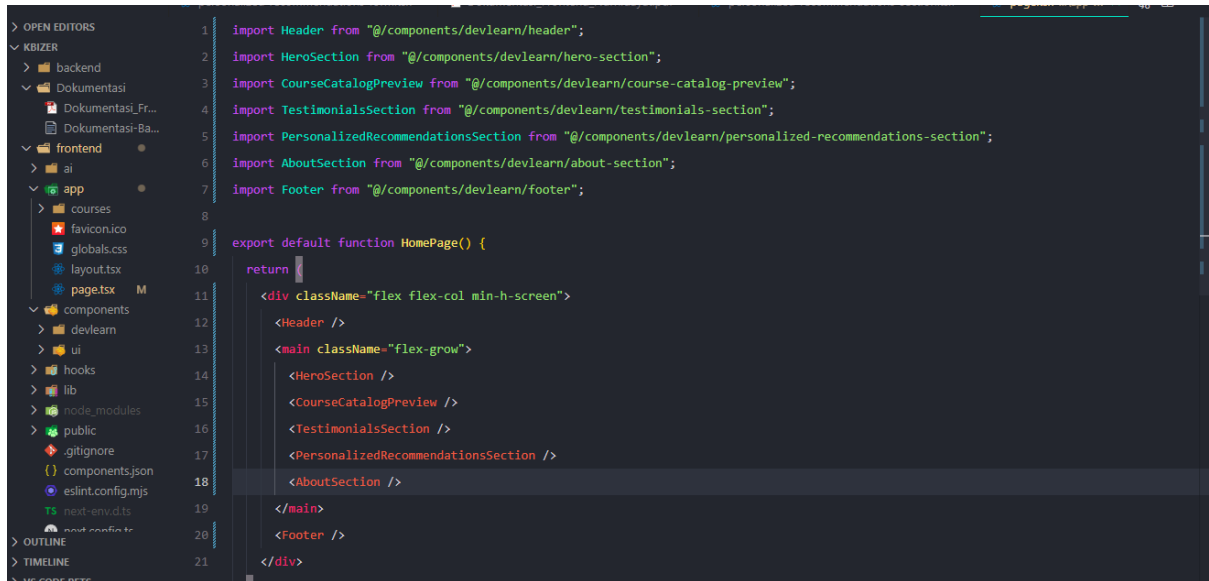
Membuat Section Rekomendasi Belajar pada Folder Devlern



```
1 import PersonalizedRecommendationsForm from './personalized-recommendations-form';
2 import { personalizedRecommendations } from '@ai/flows/personalized-recommendations';
3
4 export default function PersonalizedRecommendationsSection() {
5   return (
6     <section id="recommendations" className="py-16 md:py-24 bg-background">
7       <div className="container mx-auto px-4 sm:px-6 lg:px-8">
8         <div className="text-center mb-12">
9           <h2 className="text-3xl md:text-4xl font-bold text-primary">Discover Your Personalized Path</h2>
10          <p className="mt-3 text-lg text-foreground/70 max-w-xl mx-auto">
11            Let our AI guide you to the programming languages and courses that best match your unique interests and career aspirations.
12          </p>
13        </div>
14        <PersonalizedRecommendationsForm getRecommendations={personalizedRecommendations} />
15      </div>
16    </section>
17  )
18 }
```

Saya membuat komponen **PersonalizedRecommendationsSection** untuk menampilkan bagian rekomendasi belajar di halaman website. Di bagian ini, user bisa mendapatkan rekomendasi bahasa dan kursus yang sesuai dengan minat dan tujuan mereka. Saya menampilkan judul, deskripsi singkat, dan memanggil form yang sudah saya buat sebelumnya. Form ini langsung terhubung dengan AI yang akan memberikan rekomendasi secara otomatis. Fitur ini saya buat supaya user mudah menemukan jalur belajar yang paling cocok.

Memanggil semua komponen yang telah dibuat ke HomePage

The image is a screenshot of a Visual Studio Code editor window. On the left, the Explorer sidebar shows a project structure with folders like 'backend', 'frontend', 'app', 'components', and 'ui'. The 'app' folder is expanded, showing 'pages' and 'components'. The 'pages' folder is further expanded, showing 'index.tsx'. The main editor area displays the code for 'index.tsx'. The code imports several components from '@components/devlearn' and then renders them inside a 'HomePage' function. The components imported are Header, HeroSection, CourseCatalogPreview, TestimonialsSection, PersonalizedRecommendationsSection, AboutSection, and Footer. The rendering uses a flex container with 'flex-grow' for the main content area to ensure responsiveness. The code is as follows:

```
1 import Header from "@components/devlearn/header";
2 import HeroSection from "@components/devlearn/hero-section";
3 import CourseCatalogPreview from "@components/devlearn/course-catalog-preview";
4 import TestimonialsSection from "@components/devlearn/testimonials-section";
5 import PersonalizedRecommendationsSection from "@components/devlearn/personalized-recommendations-section";
6 import AboutSection from "@components/devlearn/about-section";
7 import Footer from "@components/devlearn/footer";
8
9 export default function HomePage() {
10   return (
11     <div className="flex flex-col min-h-screen">
12       <Header />
13       <main className="flex-grow">
14         <HeroSection />
15         <CourseCatalogPreview />
16         <TestimonialsSection />
17         <PersonalizedRecommendationsSection />
18         <AboutSection />
19       </main>
20       <Footer />
21     </div>
22   );
23 }
```

Saya membuat komponen **HomePage** sebagai halaman utama website. Di halaman ini saya menampilkan beberapa bagian penting seperti **Header**, **Hero Section**, **Preview Kursus**, **Testimoni**, **Rekomendasi Belajar**, **About**, dan **Footer**. Strukturnya saya susun secara berurutan mulai dari header di atas, lalu isi utama, dan diakhiri dengan footer. Bagian utamanya saya buat fleksibel dengan class flex-grow agar tampilannya responsif. Halaman ini saya buat supaya user bisa melihat semua informasi penting tentang kursus, rekomendasi belajar, dan detail website dengan tampilan yang rapi dan terstruktur.