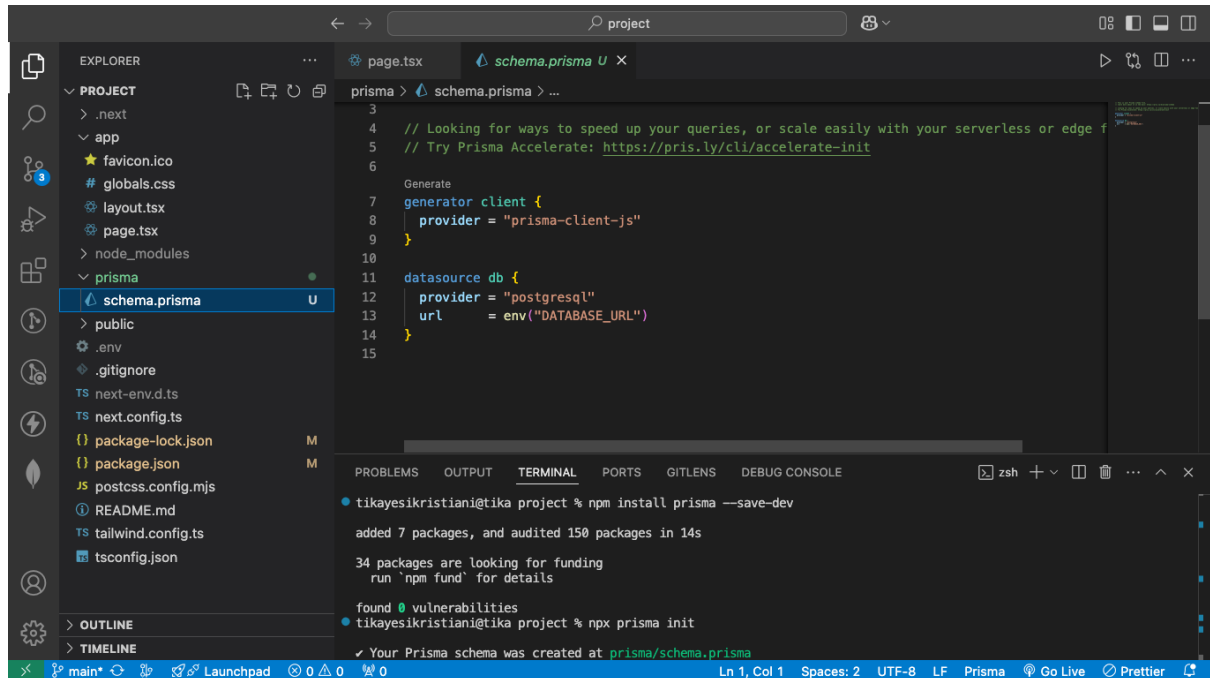


Nama : Deo Cahyo Anggoro

NPM : 22312168

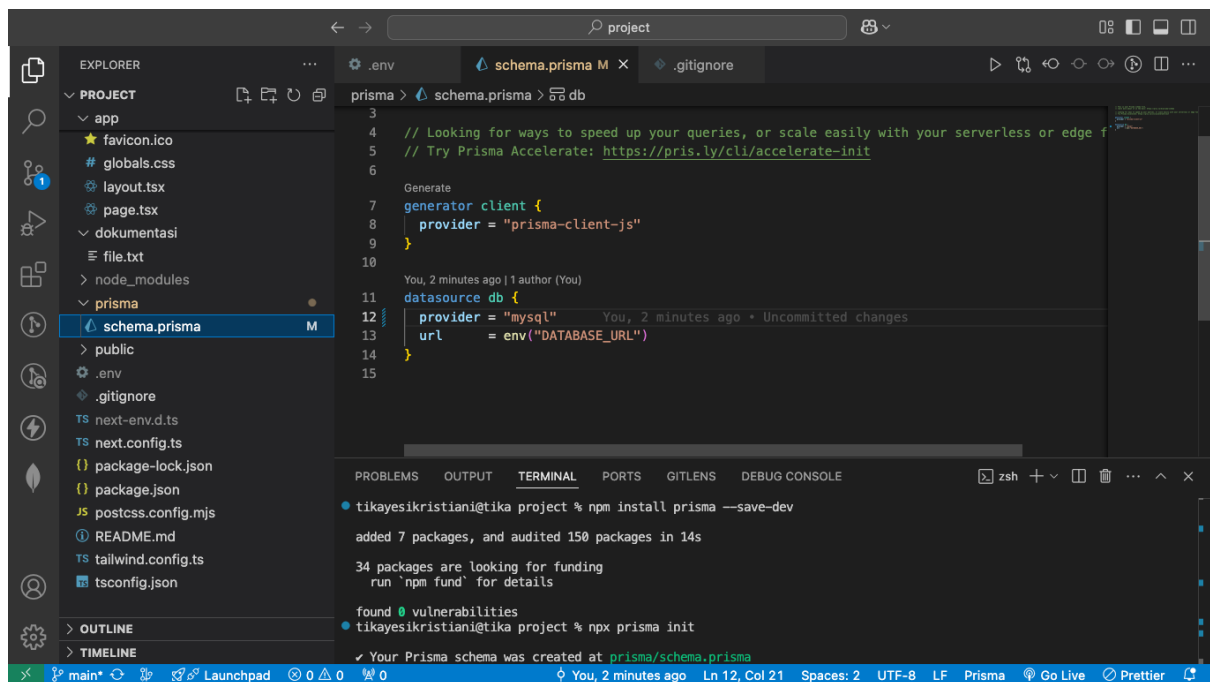
Dokumentasi Project



```
prisma > schema.prisma > ...
3
4 // Looking for ways to speed up your queries, or scale easily with your serverless or edge f
5 // Try Prisma Accelerate: https://pris.ly/cli/accelerate-init
6
7 Generate
8 generator client {
9   provider = "prisma-client-js"
10 }
11
12 datasource db {
13   provider = "postgresql"
14   url      = env("DATABASE_URL")
15 }
```

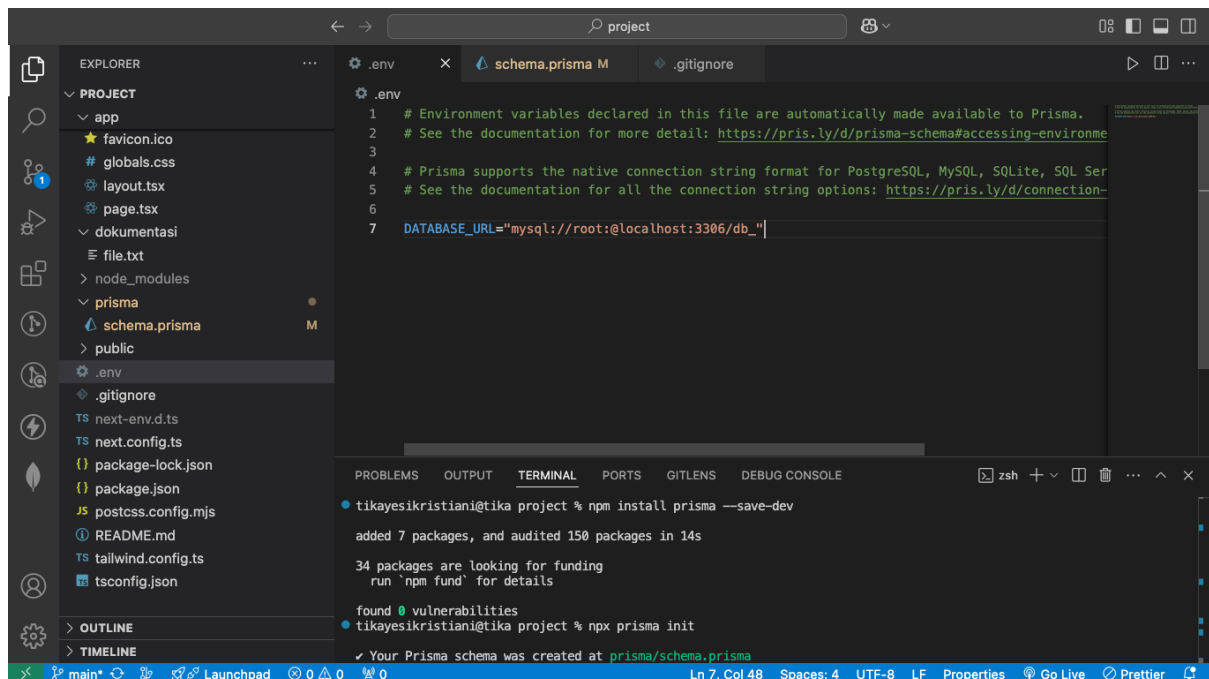
```
tikayesikristiani@tika project % npm install prisma --save-dev
added 7 packages, and audited 150 packages in 14s
34 packages are looking for funding
run 'npm fund' for details
found 0 vulnerabilities
tikayesikristiani@tika project % npx prisma init
✓ Your Prisma schema was created at prisma/schema.prisma
```

Menginstall prisma dengan cara mengetik npm install prisma --save-dev di terminal dan npx prisma init unntuk membuat file schema.prisma dan .env

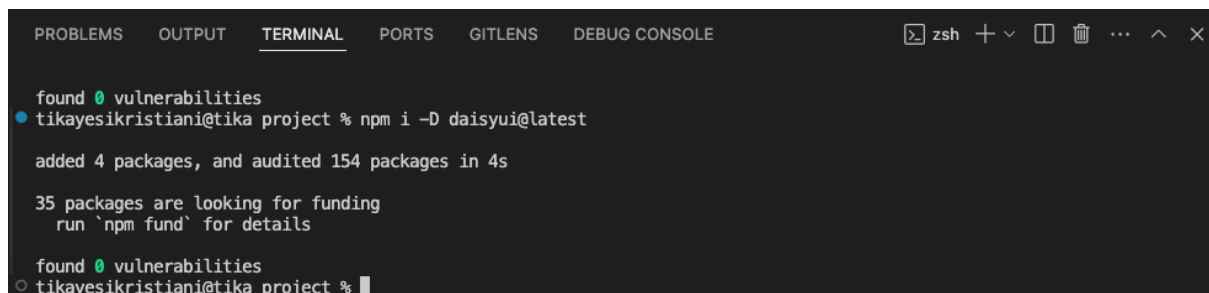


```
prisma > schema.prisma > db
3
4 // Looking for ways to speed up your queries, or scale easily with your serverless or edge f
5 // Try Prisma Accelerate: https://pris.ly/cli/accelerate-init
6
7 Generate
8 generator client {
9   provider = "prisma-client-js"
10 }
11
12 You, 2 minutes ago | 1 author (You)
13 datasource db {
14   provider = "mysql"
15   url      = env("DATABASE_URL")
16 }
```

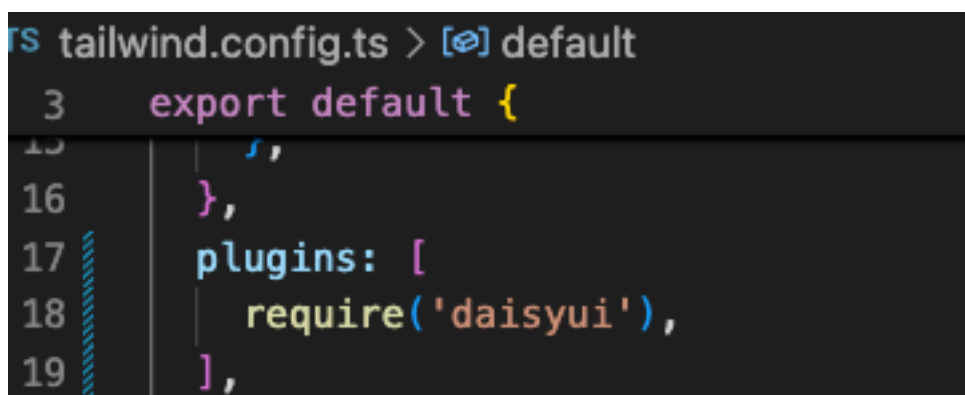
```
tikayesikristiani@tika project % npm install prisma --save-dev
added 7 packages, and audited 150 packages in 14s
34 packages are looking for funding
run 'npm fund' for details
found 0 vulnerabilities
tikayesikristiani@tika project % npx prisma init
✓ Your Prisma schema was created at prisma/schema.prisma
```



Karena Saya menggunakan mysql ganti postgresql ke mysql di file schema.prisma dan .env untuk menghubungkan ke database mysql
 Penjelasan di file .env root adalah usernya dan localhost:3306 adalah portnya



Menginstall DaisyUI dengan cara mengetik npm i -D daisyui@latest di terminal, @latest maksudnya adalah versi terbaru dari daisyui



Lalu masukan require('daisyui'), kedalam plungins di file tailwind.config.ts untuk menambahkan daisyui

```
layout.tsx M •
app > layout.tsx > RootLayout
6 export default function RootLayout({
11   return (
12     <html lang="en" data-theme="lemonade">
13       <body>
14         { /* Area Header */ }
15         <header></header>
16
17         { /* Area Conten */ }
18         <section>
19           {children}
20         </section>
21
22         { /* Area Footer */ }
23         <footer></footer>
24       </body>
25     </html>
26   )
27 }
```

Menambahkan Header, Section, dan Footer bertujuan untuk memisahkan area header, konten, dan footer untuk menciptakan struktur yang lebih terorganisir dan semantic

```
{ /* Area Footer */ }
<footer>
Copyright &copy; 2025 – Manusia Biasa Team
</footer>
```

Menambahkan text copyright di footer, © adalah logo copyright

```
> layout.tsx > ...
You, 1 minute ago | 3 authors (You and others)
// import file "globals.css"
import "../globals.css" |
```

Mengimport globals.css di layout.tsx

```

    { /* Area Footer */ }
    <footer className="flex justify-center bg-blue-700 text-white py-2.5">
    Copyright &copy; 2025 - Manusia Biasa Team
    </footer>

```

Copyright © 2025 - Manusia Biasa Team

Memperbagus tampilan footer

`flex justify-center`

agar textnya berada ditengah

`bg-blue-700`

membuat background bewarna biru

`text-white`

membuat text bewarna putih

`py-2.5`

memberikan padding sebesar 10 px

```

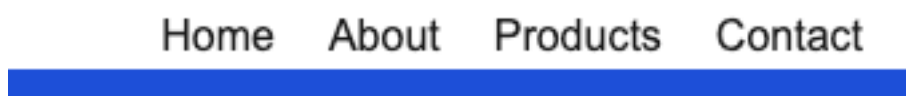
<body>
  { /* Area Header */ }
  <header>
    <Link href="/">Ecommerce</Link>
    <nav>
      <Link href="/">Home</Link>
      <Link href="/">About</Link>
      <Link href="/">Products</Link>
      <Link href="/">Contact</Link>
    </nav>
  </header>

```

Menambahkan Link yang berguna untuk tombol navigasi

```
    { /* Area Header */ }
    <header>
      <Link href={"/"}>Ecommerce</Link>
      <nav className="flex justify-end">
        <Link href={"/"} className="mr-2.5">Home</Link>
        <Link href={"/"} className="mx-2.5">About</Link>
        <Link href={"/"} className="mx-2.5">Products</Link>
        <Link href={"/"} className="ml-2.5 mr-5">Contact</Link>
      </nav>
    </header>
```

Menambahkan flex justify-end agar tombol berada di sebelah kanan



Dan menambahkan margin agar tombolnya memiliki jarak
mr untuk margin kanan
ml untuk margin kiri
dan 2.5 adalah ukuran marginya sebesar 10px

```
    { /* Area Header */ }
    <header className="flex justify-between items-center px-5 py-3">
      <Link href={"/"}>Ecommerce</Link>
      <nav className="flex justify-end">
        <Link href={"/"} className="mr-2.5">Home</Link>
        <Link href={"/"} className="mx-2.5">About</Link>
        <Link href={"/"} className="mx-2.5">Products</Link>
        <Link href={"/"} className="ml-2.5">Contact</Link>
      </nav>
    </header>
```

flex justify-between items-center px-5 py-3

Berguna agar tata letaknya sejajar
flex justify-between Mengatur jarak antar flex items secara horizontal sehingga item pertama berada di sisi kiri, item terakhir di sisi kanan
items-center Menyelaraskan (align) flex items secara vertikal di tengah-tengah flex container
px-5 untuk mengatur padding kiri dan kanan sebesar 20px
py-3 untuk mengatur padding atas dan bawah sebesar 12px

```

model tb_barang{
  id Int @id @default(autoincrement())
  nama_barang String @db.VarChar(50)
  deskripsi String @db.VarChar(100)
  harga Float @db.Double
  kategori String @db.VarChar(20)
  status status
}

enum status {
  Y
  N
}

```

Menambah tabel barang di schema.prisma

Int adalah tipe data integer untuk bilangan bulat

@id menandai bahwa itu adalah primary key

@default(autoincrement()) untuk membuat nilai secara otomatis

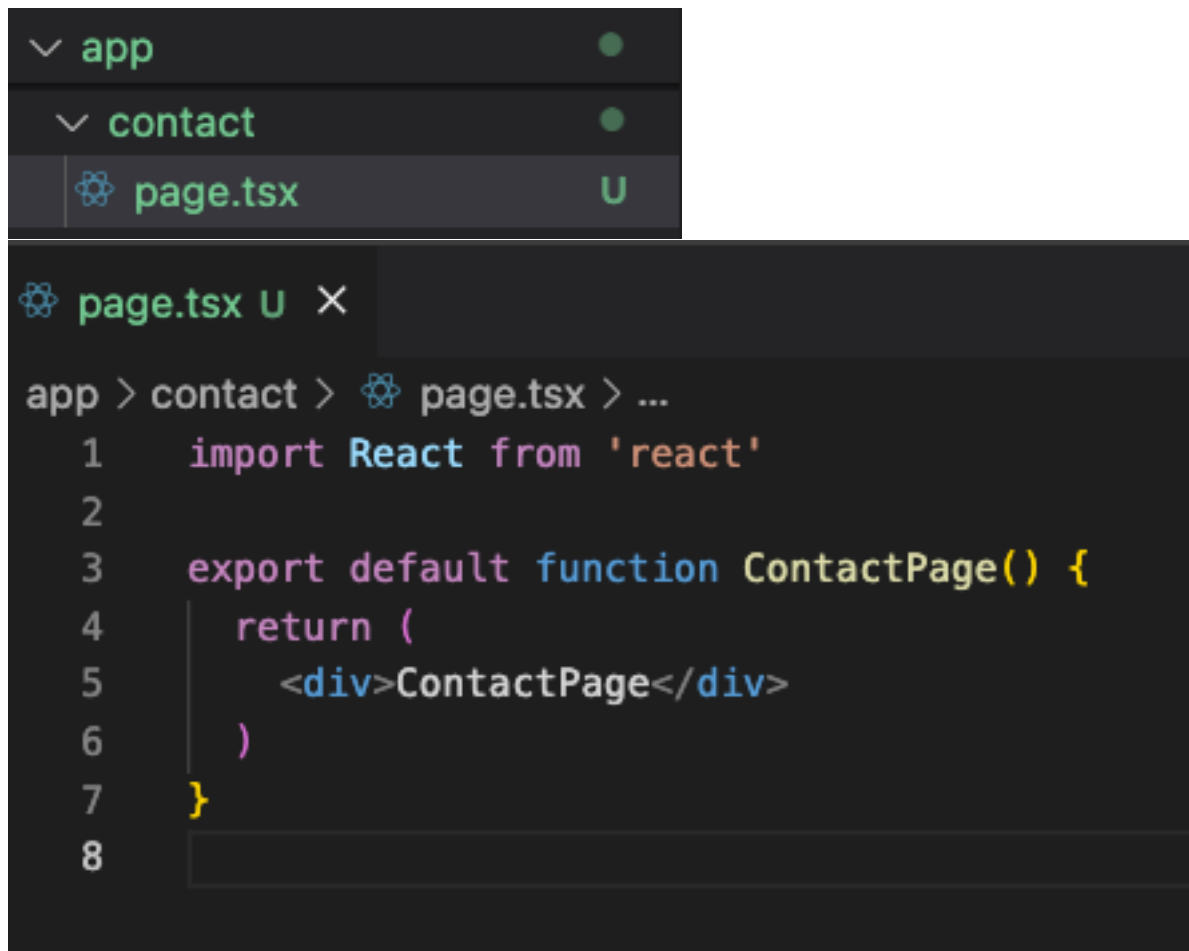
String adalah tipe data untuk teks

@db.VarChar(50) untuk tipe data ini di database adalah varchar dan untuk nilainya adalah panjang maksimum karakter

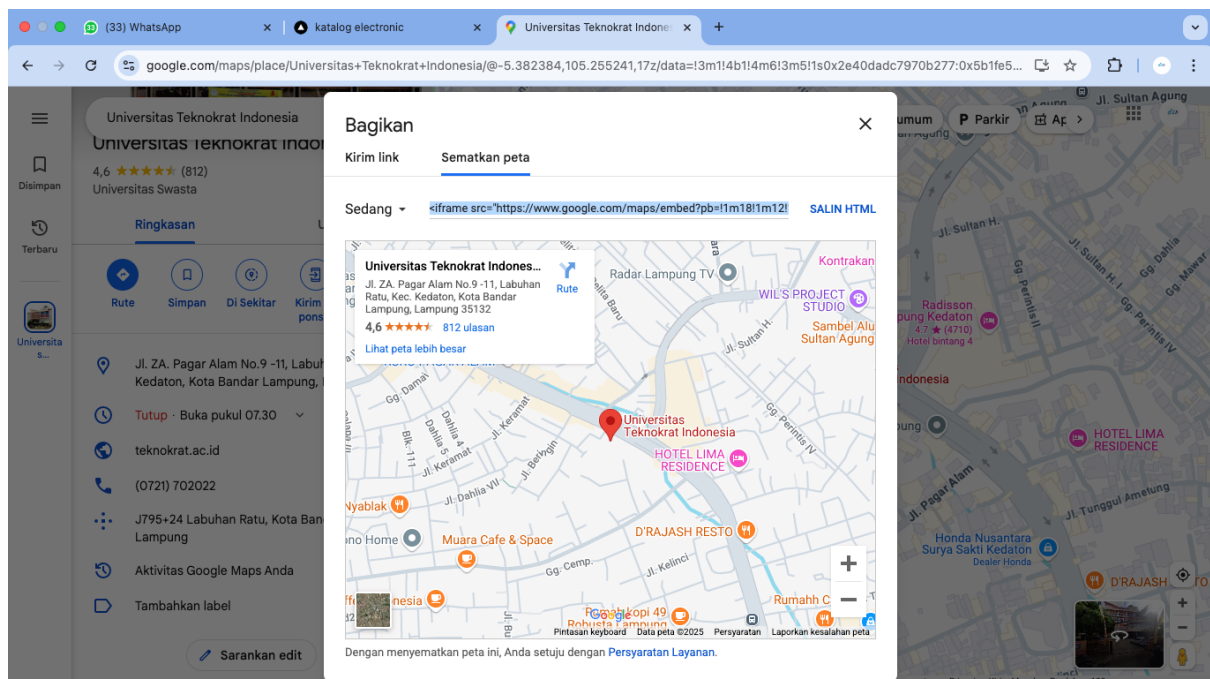
Float adalah tipe data untuk bilangan pecahan

@db.Double untuk tipe data di database adalah double

Untuk melakukan migrate ketikkan `npx prisma migrate dev` di terminal lalu beri nama lalu prisma akan membuat migration file

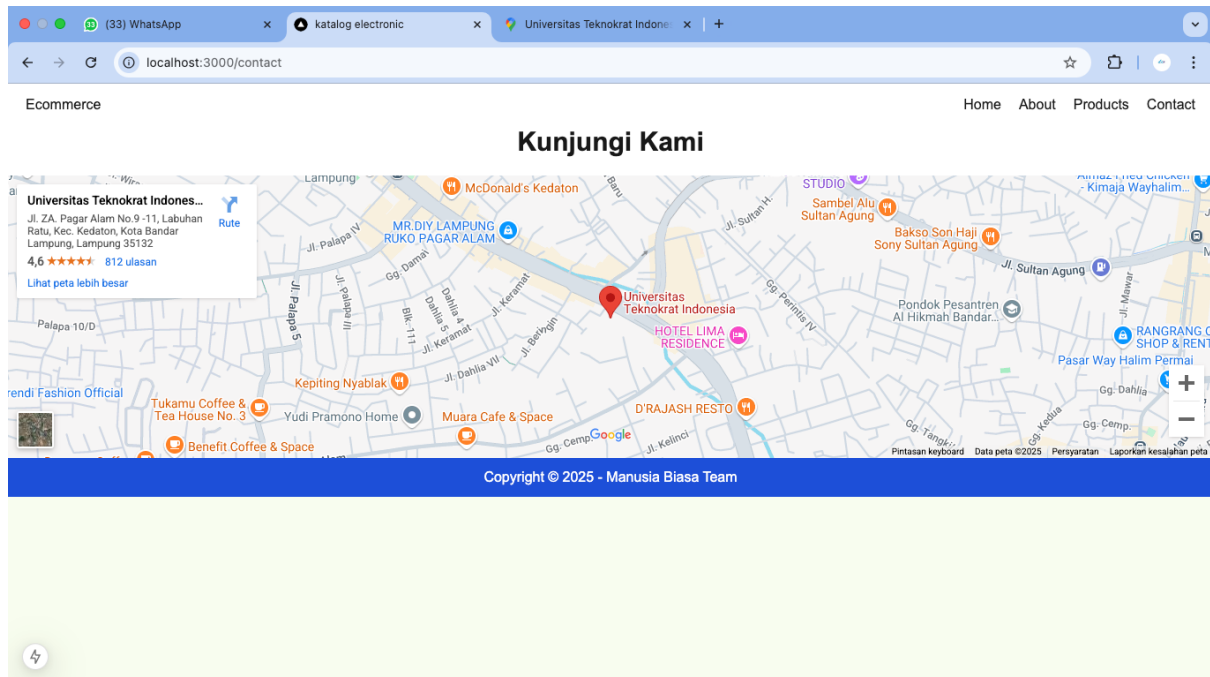


Buat folder baru didalam folder app dengan nama contact lalu buat file dengan nama page.tsx
Lalu ketikkan rfc lalu enter dan ganti page menjadi ContactPage



Untuk menambah map dapat langsung mengcopynya dari google map

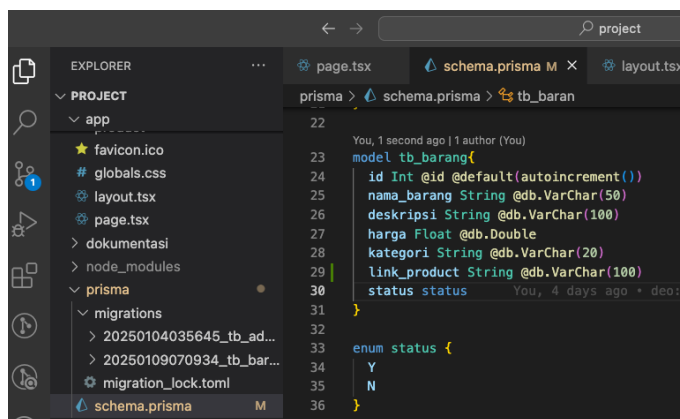
Lalu atur ukuran mapnya menggunakan tailwind dengan cara mengganti defaultnya width dan height dengan className=""



Lalu menambah text judul menggunakan <h1></h1> dan diatur menggunakan tailwind

```
<h1 className="text-3xl font-bold mb-5 text-center">Kunjungi Kami</h1>
```

text-3xl untuk mengatur font size sebesar 30px dan line height sebesar 36px
font bold untuk menebalkan textnya
mb-5 untuk memberikan margin dibawah sebesar 20px
text center agar text berada ditengah



Untuk menambah field di prisma pertama masuk ke file schema.prisma
lalu tambahkan field ke dalam tabel

Disini saya menambahkan link_product dengan tipe data string varchar sebanyak dengan max 100 huruf


```
History restored
tikayesikristiani@tika project % brew services start mysql
=> Successfully started 'mysql' (label: homebrew.mxcl.mysql)
tikayesikristiani@tika project % npx prisma migrate dev
```

Setelah itu lakukan migrate dengan cara mengetikan `npx prisma migrate dev`

```
tikayesikristiani@tika project % brew services start mysql
=> Successfully started 'mysql' (label: homebrew.mxcl.mysql)
tikayesikristiani@tika project % npx prisma migrate dev
Environment variables loaded from .env
Prisma schema loaded from prisma/schema.prisma
Datasource "db": MySQL database "db_katalog" at "localhost:3306"
? Enter a name for the new migration: > tb_barang_rev02
```

Lalu beri nama migration datanya, disini saya menamainya `tb_barang_rev02` karena ini adalah revisi ke 2

```
PROBLEMS OUTPUT TERMINAL PORTS GITLENS DEBUG CONSOLE

The following migration(s) have been created and applied from new schema changes:
migrations/
├─ 20250113052038_tb_barang_rev02/
└─ migration.sql

Your database is now in sync with your schema.
✓ Generated Prisma Client (v6.1.0) to ./node_modules/@prisma/client in 271ms

tikayesikristiani@tika project %
```

File	Line	Content
migrations	28	kategori Str
> 20250104035645_tb_admin_rev01	29	link_product
> 20250109070934_tb_barang_rev01	30	status status
> 20250113052038_tb_barang_rev02	31	}
migration_lock.toml	32	
schema.prisma	33	enum status {
	34	Y
	35	N

Setelah selesai maka akan muncul file migration yang dibuat tadi dan fieldnya akan masuk ke database

```
page.tsx .../contact schema.prisma page.tsx .../product X
PROJECT
├─ app
├─ models
├─ product
└─ page.tsx
favicon.ico
globals.css
layout.tsx


app > product > page.tsx > ProductPage
You, 1 hour ago | 1 author (You)
1 import React from 'react'
2
3 export default function ProductPage() {
4   return (
5     <div>ProductPage</div>
6   )
7 }
8
```

Buat folder baru dengan nama `product` dan file `page.tsx` untuk nantinya dijadikan halaman `product`

Lalu ketikkan `rfc` untuk mempermudah membuat `export default function`

```
<Link href="/" className="mr-2.5">Home</Link>
<nav className="flex justify-end">
  <Link href="/" className="mr-2.5">Home</Link>
  <Link href="/about" className="mx-2.5">About</Link>
  <Link href="/product" className="mx-2.5">Products</Link>
  <Link href="/contact" className="ml-2.5">Contact</Link>
</nav>
</header>
```

Lalu tambahkan /product di rootlayout agar navigasinya mengarah ke file page yang ada di dalam folder product

```
app > models > TS barang.ts >  getData
You, 4 minutes ago | 2 authors (maechel11 and one other)
1  "use server";
2
3  import { PrismaClient } from "@prisma/client";
4
5  //buat variabel "prisma"
6  const prisma = new PrismaClient ();
7
8  // buat fungsi untuk ambil data barang
9  export async function getData() {}
```

Buka file barang.ts lalu membuat export async function getData(){}
Export digunakan agar file ini dapat di impor di file lain
Async menunjukkan bahwa fungsi ini bersifat asynchronous
function getData ini digunakan untuk mengambil data dari database

```

app > models > TS barang.ts > getData > barang
You, 4 minutes ago | 2 authors (You and one other)
1  "use server";
2
3  import { PrismaClient } from "@prisma/client";
4
5  //buat variabel "prisma"
6  const prisma = new PrismaClient ();
7
8  // buat fungsi untuk ambil data barang
9  export async function getData() {
10     // buat variable "barang"
11     const barang = await prisma.tb_barang.findMany({});
12 }

```


Membuat variable barang `const barang = await prisma.tb_barang.findMany({});` digunakan untuk mengambil semua data dari tabel `tb_barang` dalam database menggunakan Prisma

```

app > models > TS barang.ts > getData > barang > where
You, 12 minutes ago | 2 authors (You and one other)
1  "use server";
2
3  import { PrismaClient } from "@prisma/client";
4
5  //buat variabel "prisma"
6  const prisma = new PrismaClient ();
7
8  // buat fungsi untuk ambil data barang
9  export async function getData() {
10     // buat variable "barang"
11     const barang = await prisma.tb_barang.findMany({
12         where: {
13             status: "Y",
14         },
15     });
16 }

```

Where: {status:"Y"},} ini maksudnya data yang akan ditampilkan hanya data yang enum statusnya adalah Y

```
app > models > TS barang.ts >  getData
// buat variabel prisma
6   const prisma = new PrismaClient ();
7
8   // buat fungsi untuk ambil data barang
9   export async function getData() {
10    // buat variable "barang"
11    const barang = await prisma.tb_barang.findMany({
12      where: {
13        status: "Y",
14      },
15    });
16
17    return barang;
18  }
```

Return barang; digunakan untuk mengembalikan data yang telah diambil dari database
Dan kenapa harus dikembalikan? Karena data ini nantinya akan dibuat untuk ditampilkan