

# **Panduan Membuat Website**



**Linear**KU

## I. Set Up

### A. Kebutuhan

Memahami Basic :

HTML



HTML adalah singkatan dari Hypertext Markup Language, HTML merupakan salah satu bahasa pengkodean atau pemograman yang digunakan untuk membuat halaman website yang ditampilkan pada web browser

CSS



CSS merupakan singkatan dari Cascading Style Sheets, CSS digunakan untuk menentukan gaya dari tampilan website yang akan kamu buat seperti tata letak halaman, warna , dan font. Semuanya dapat ditentukan oleh CSS, CSS bertujuan agar website terlihat lebih menarik.

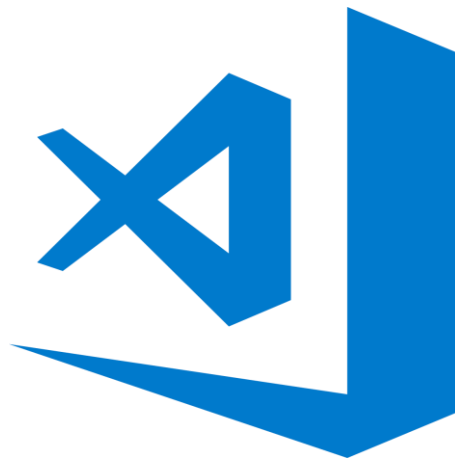
## Java Script



JavaScript adalah bahasa pemrograman yang digunakan untuk membuat halaman web interaktif dan dinamis. JavaScript merupakan salah satu bahasa pemrograman yang paling umum digunakan, setelah HTML dan Cascading Style Sheet (CSS).

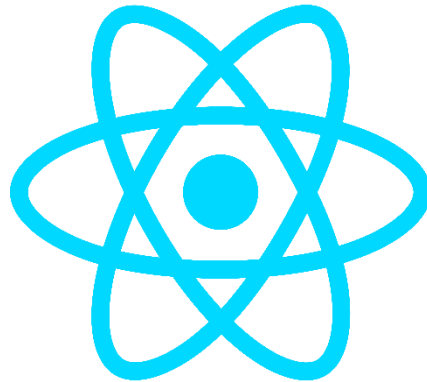
### B. Tools Yang Digunakan

#### 1. Visual Studio Code



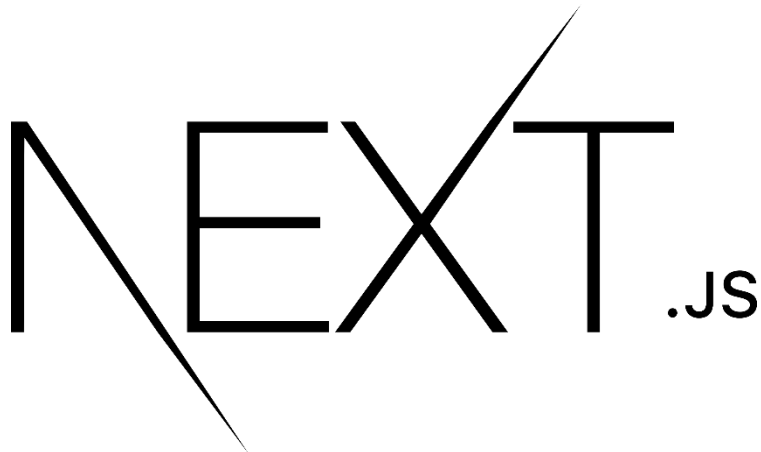
Visual Studio Code adalah editor source code yang ringan, gratis, dan open source, yang dikembangkan oleh Microsoft. Alat ini biasanya digunakan untuk pengembangan web, tetapi mendukung banyak bahasa pemrograman dan memiliki berbagai extension yang tersedia.

## 2. React JS



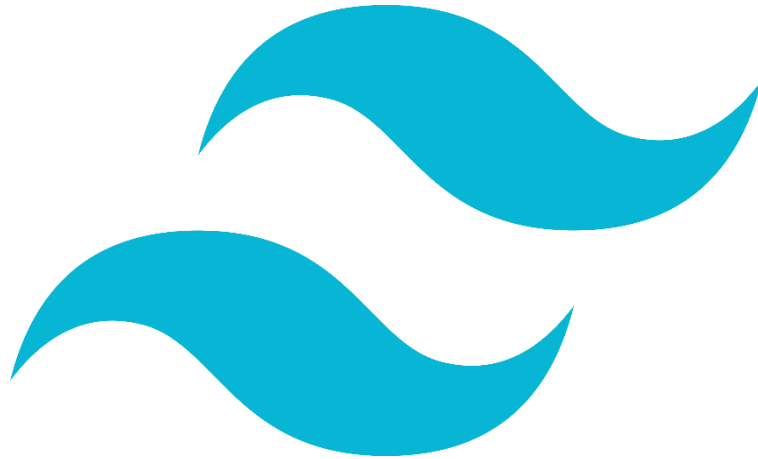
React JS adalah pustaka JavaScript yang digunakan untuk membangun antarmuka pengguna (user interface) yang interaktif untuk situs web, aplikasi desktop, dan aplikasi seluler

## 3. Next JS



Next.js adalah kerangka kerja JavaScript yang digunakan untuk membangun aplikasi web modern berbasis React:

#### 4. Tailwind CSS



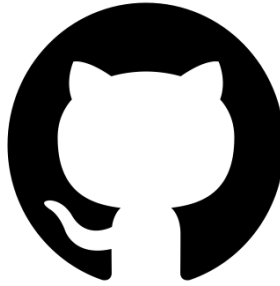
Tailwind CSS merupakan sebuah framework Cascading Style Sheet yang digunakan untuk mengustom atau mendesain user interface pada sebuah web. Framework ini berbasis utility yang hanya terdiri dari utility class dan tanpa utility komponen lainnya.

#### 5. Node JS



Node.js adalah perangkat lunak yang digunakan untuk mengembangkan aplikasi berbasis web. Node.js merupakan runtime environment yang memungkinkan pengembang untuk menjalankan kode JavaScript di luar lingkungan browser

## 6. Github



GitHub adalah platform berbasis cloud tempat Anda dapat menyimpan, berbagi, dan bekerja sama dengan orang lain untuk menulis kode . Menyimpan kode Anda dalam "repository" di GitHub memungkinkan Anda untuk: Memamerkan atau berbagi hasil kerja Anda. Melacak dan mengelola perubahan pada kode Anda dari waktu ke waktu. Membiarkan orang lain meninjau kode Anda, dan memberikan saran untuk memperbaikinya.

## 7. Vercel



Vercel adalah platform berbasis cloud yang digunakan untuk menghosting situs web dan aplikasi web. Vercel didesain untuk mempermudah proses pengembangan dan pemindahan proyek web, sehingga pengembang dapat fokus pada pengembangan kode tanpa perlu khawatir dengan infrastruktur hosting.

## II. Pembuatan Aplikasi

### A. SetUp NextJS

```
D:\Coding  
> npx create-next-app@latest|
```

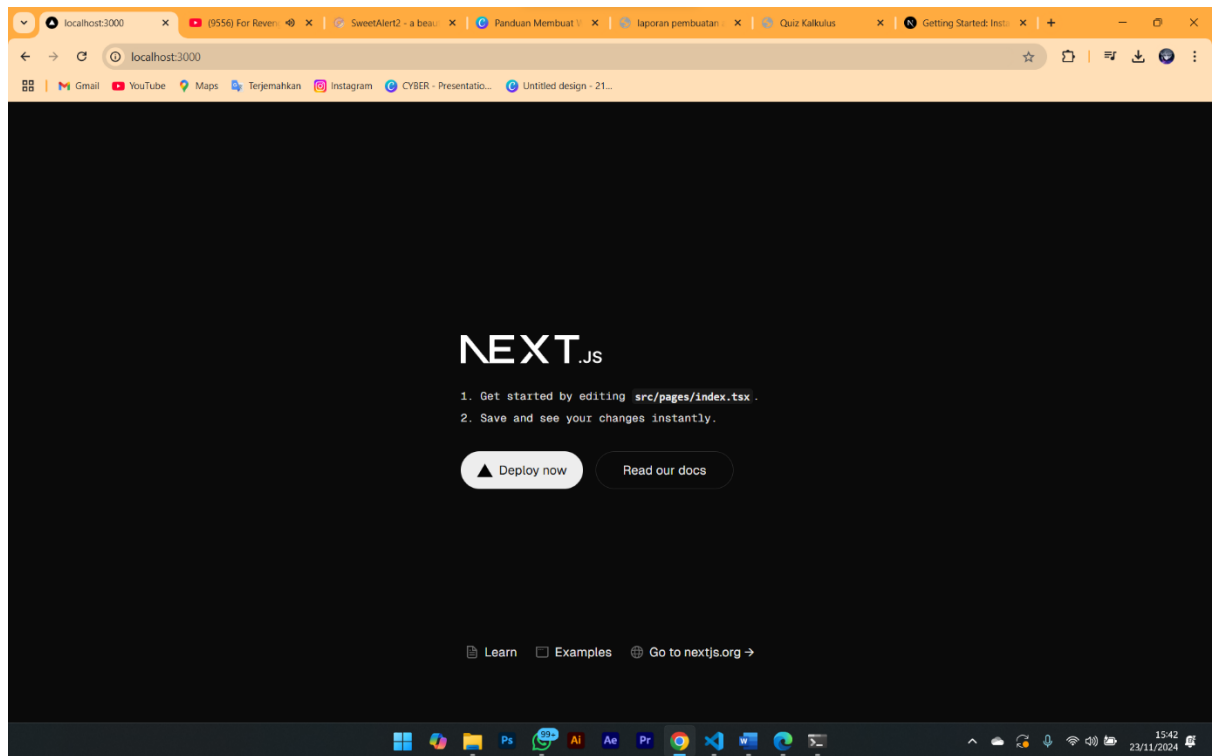
Memulai dengan menginstall next-js dengan cara mengetikan *npx create-next-app@latest*

```
Need to install the following packages:  
create-next-app@15.0.3  
Ok to proceed? (y) y  
✓ What is your project named? ... linearku  
✓ Would you like to use TypeScript? ... No / Yes  
✓ Would you like to use ESLint? ... No / Yes  
✓ Would you like to use Tailwind CSS? ... No / Yes  
✓ Would you like your code inside a `src/` directory? ... No / Yes  
✓ Would you like to use App Router? (recommended) ... No / Yes  
✓ Would you like to use Turbopack for next dev? ... No / Yes  
✓ Would you like to customize the import alias (@/* by default)? ... No / Yes  
✓ What import alias would you like configured? ... @/*  
Creating a new Next.js app in D:\Coding\linearku.
```

Atur project yang ingin kalian gunakan disini, Ketika sudah success, bukalah project yang sudah dibuat tadi didalam software visual studio code

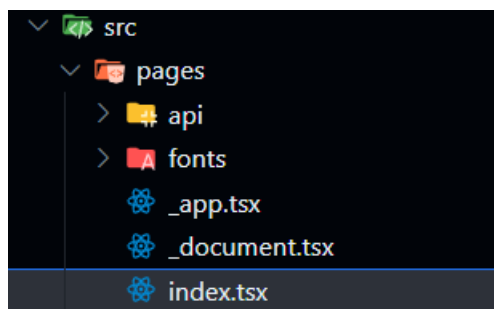
```
linearku on  master via v20.9.0  
○ > npm run dev  
  
> linearku@0.1.0 dev  
> next dev  
  
▲ Next.js 15.0.3  
- Local: http://localhost:3000  
  
✓ Starting...  
✓ Ready in 4.1s
```

Jalankan localhost dengan mengetikan *npm run dev*



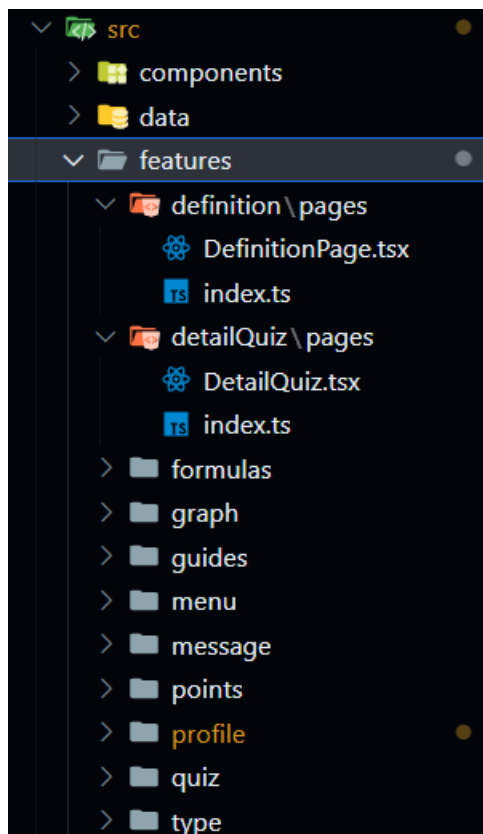
Cek di browser localhost:3000 sesuai port yang diatur (default nya 3000), ini adalah bagian template dari next js.

## B. Buat Landing Page



Kode dari template next js tadi ada didalam file index.tsx ini, kalian bisa mengubahnya disini.





Didalam folder src, buatlah folder features, untuk membuat fitur per section. Didalam folder features buat folder fitur nya (ex:definition). Didalam folder defition buat lagi folder pages. Dan terakhir didalam folder pages buatlah 2 file yang terdiri dari Fitur.tsx dan index.ts

```
index.ts src\features\definition\pages
1 export * from "./DefinitionPage"
```

Didalam file index.ts tuliskan kode seperti ini,namun sesuaikan dengan nama function yang kalian buat di folder tsx kalian contohnya seperti *export const DefinitionPage* yang ada pada gambar dibawah

```

1  export const DefinitionPage = () => {
2    return (
3      <div className="my-20 flex flex-col justify-center px-4 text-center lg:flex-row lg:gap-x-28 lg:space-x-4 lg:px-40 lg:text-center">
4        <TitleSection>Grafik Fungsi Linear</TitleSection>
5        <Image
6          src="/images/matematic.png"
7          alt="definitor.png"
8          width={400}
9          height={400}
10         className="mx-auto"
11       />
12      <div className="description my-auto space-y-2 text-justify">
13        <h1 className="my-5 hidden text-start text-2xl font-bold uppercase text-text dark:text-darkText lg:block">
14          GRAFIK FUNGSI LINEAR
15        </h1>
16        <p>
17          <span className="bg-sky-400 font-bold">Grafik Fungsi Linear </span>
18          Pada dasarnya, definisi dari fungsi linear adalah relasi yang
19          memasangkan setiap anggota yang ada di himpunan A dengan anggota lain
20          di himpunan B. Semua anggota yang ada di himpunan A harus berpasangan
21          dengan anggota yang ada di himpunan B. Fungsi linear ini disebut juga
22          dengan Persamaan Garis Lurus (PGL). Seperti yang telah diungkapkan
23          sebelumnya, fungsi dalam dunia matematika akan berkaitan erat dengan
24          unsur pembentuknya, yakni berupa variabel, koefisien, dan konstanta.
25        </p>
26        <p>
27          Jadi, fungsi linear adalah suatu fungsi yang membentuk grafik secara
28          garis lurus. Fungsi linear ini juga menjadi fungsi yang telah
29          mendapatkan pangkat tertinggi dengan variabelnya sama dengan satu.
30        </p>
31      </div>
32    </div>
33  );
34  };

```

Diatas adalah contoh kode yang dibuat pada file fitur.tsx (ex:DefinitionPage.tsx)

```

1  export default function UserPage() {
2    return (
3      <
4        <Head>
5          <link rel="preconnect" href="https://fonts.googleapis.com" />
6          <link rel="preconnect" href="https://fonts.gstatic.com" />
7          <link
8            href="https://fonts.googleapis.com/css2?family=Public+Sans:ital,wght@0,100..900;1,100..900&display=swap"
9            rel="stylesheet"
10          />
11          <title>LinearKU</title>
12          <meta name="description" content="Generated by create-t3-app" />
13          <link rel="icon" href="/kalkuluslogo.ico" />
14        </Head>
15        <PageContainer>
16          <DefinitionPage />

```

Lalu kita bisa langsung mengimport DefinitionPage tadi dan menaruhnya di folder root index.ts yang ada pada folder pages



### GRAFIK FUNGSI LINEAR

**Grafik Fungsi Linear** Pada dasarnya, definisi dari fungsi linear adalah relasi yang memasangkan setiap anggota yang ada di himpunan A dengan anggota lain di himpunan B. Semua anggota yang ada di himpunan A harus berpasangan dengan anggota yang ada di himpunan B. Fungsi linear ini disebut juga dengan Persamaan Garis Lurus (PGL). Seperti yang telah diungkapkan sebelumnya, fungsi dalam dunia matematika akan berkaitan erat dengan unsur pembentuknya, yakni berupa variabel, koefisien, dan konstanta.

Jadi, fungsi linear adalah suatu fungsi yang membentuk grafik secara garis lurus. Fungsi linear ini juga menjadi fungsi yang telah mendapatkan pangkat tertinggi dengan variabelnya sama dengan satu.

Ini adalah hasil dari pembuatan fitur definition tadi

```

1  export default function UserPage() {
2    return (
3      <
4        <Head>
5          <link rel="preconnect" href="https://fonts.googleapis.com" />
6          <link rel="preconnect" href="https://fonts.gstatic.com" />
7          <link
8            href="https://fonts.googleapis.com/css2?family=Public+Sans:ital,wght@0,100..900;1,100..900&display=swap"
9            rel="stylesheet"
10          />
11          <title>LinearKU</title>
12          <meta name="description" content="Generated by create-t3-app" />
13          <link rel="icon" href="/kalkuluslogo.ico" />
14        </Head>
15        <PageContainer>
16          <DefinitionPage />
17          <FormulaPages />
18          <PointSearch />
19          <GraphPage />
20          <GuidesPage />
21          <ProfilePage />
22          <MessagePage />
23        </PageContainer>
24      )
25    };
26  }

```

Buatlah beberapa fitur lagi sesuai yang dibutuhkan, penulisan kode seperti ini sangat worth it dikarenakan jika ada error pada sebuah fitur, maka kita bisa langsung ke fitur tersebut tidak perlu pusing mencari error nya, beberapa fitur yang sudah dibuat sudah menjadi seperti ini :

### RUMUS FUNGSI LINEAR

Y = MX + C

### RUMUS CARI TITIK POTONG

Bagaimana Cara Mencari Titik Potong Sumbu X ?

Bagaimana Cara Mencari Titik Potong Sumbu Y ?

Section rumus fungsi linear, disini kita membuat sebuah section fungsi linear untuk membuat user memahami bagaimana cara kerja fungsi linear

**BUAT GRAFIK**

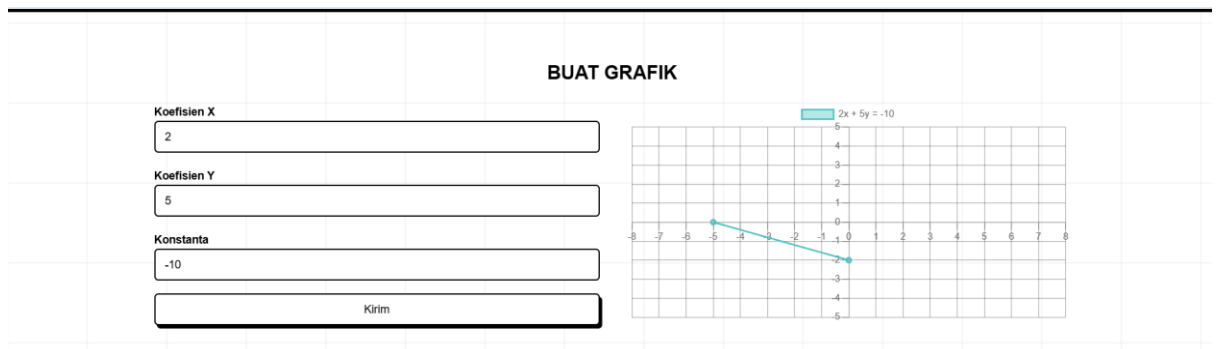
**Koefisien X**

**Koefisien Y**

**Konstanta**

**GRAFIK MU AKAN  
MUNCUL DISINI**

Section buat grafik, yang bertujuan untuk user mengetahui bagaimana grafik yang dihasilkan dari koefisien koefisien yang sudah ditentukan



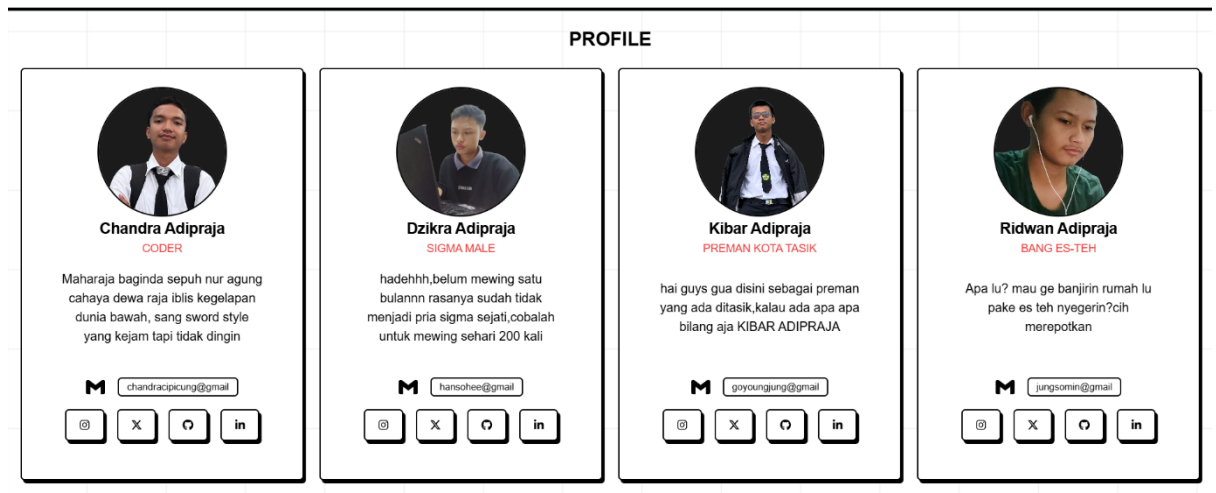
Hasil dari buat grafik

**MANUAL BOOK & CARA MEMBUAT WEBSITE**

**MANUAL BOOK**

**CARA MEMBUAT WEBSITE**

Section guide, section ini bertujuan untuk memberikan guide kepada user bagaimana cara menggunakan website ini, dan juga bagaimana cara pembuatan website tersebut

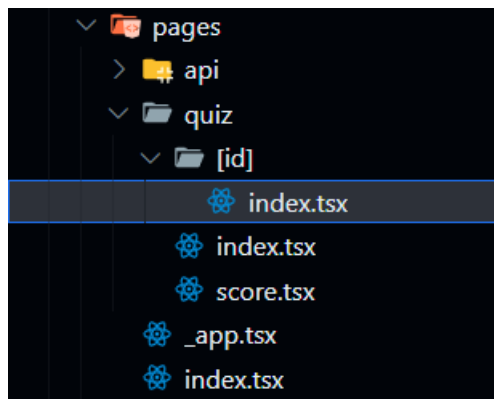


Section ini merupakan section team yang membuat website ini,yang dimulai dari coder, designer, dan lain-lain

The 'PESAN' section contains a form for user feedback. It includes three input fields: 'Nama' (Name) with the value 'Chandra', 'Email' with the value 'chandra@gmail.com', and a larger 'Pesan' (Message) field with the placeholder text 'Kirim Pesan Anda Disini'. A 'Submit' button is located at the bottom of the form.

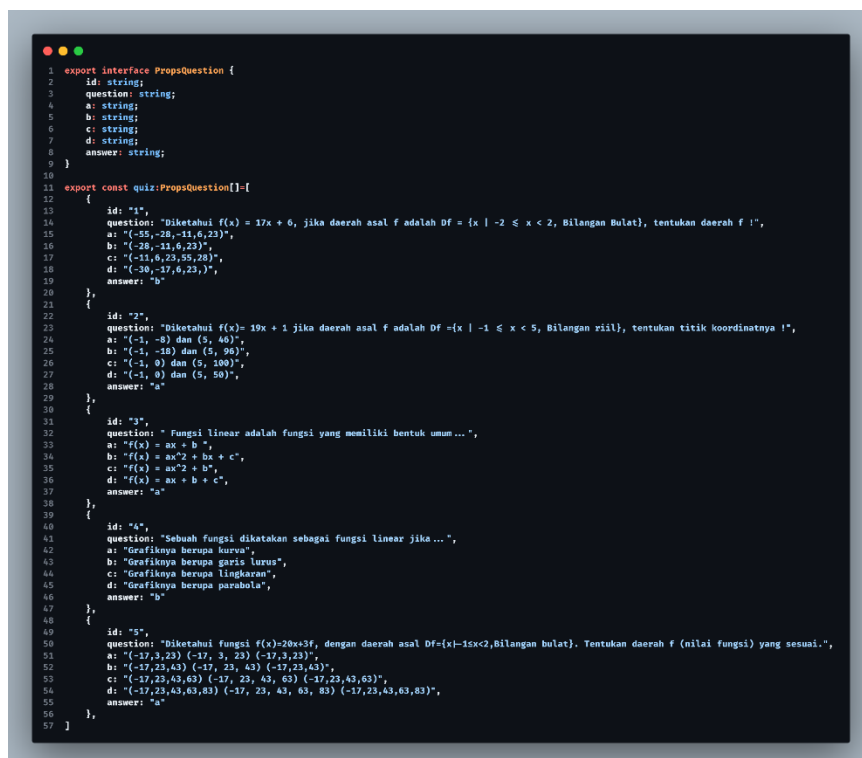
Section pesan ini merupakan sebuah user experience yang dibuat supaya user memberikan saran ataupun kritik dll kepada team dari pembuat website tersebut.

### C. Buat Fitur Website Quiz



Membuat sebuah fitur yang bertujuan supaya user pindah halaman, bisa kita gunakan folder baru di sebuah folder pages (ex: quiz didalam folder pages), dan buat file index.tsx sebagai url base dari url/quiz, juga score.tsx sebagai url Ketika user sudah selesai mengerjakan quiz

Disana kita juga membuat sebuah folder [id] untuk mengambil param, karena kita pasti mempunyai banyak soal maka kita menggunakan id saja, supaya kita tidak perlu membuat banyak file untuk setiap pertanyaan. Contoh dari url nya seperti ini url/quiz/1, url/quiz/2 sampai dengan url/quiz/n.



Buatlah sebuah data statis untuk pertanyaan dan jawaban

```

1 // src/components/QuizQuestion.tsx
2 import React from "react";
3 import { PropsQuestion } from "../index";
4 import { Button } from "~/components/ui/Button";
5 import { useRouter } from "next/router";
6 import { Progress } from "~/components/ui/Progress";
7
8 interface QuizQuestionProps {
9   questionData: PropsQuestion;
10   onNext: () => void;
11   onBack: () => void;
12   selectedOption: string | null;
13   onSelectOption: (option: string) => void;
14   currentStep: number;
15   totalSteps: number;
16 }
17
18 const QuizQuestion: React.FC<QuizQuestionProps> = ({
19   questionData,
20   onNext,
21   onBack,
22   selectedOption,
23   onSelectOption,
24   currentStep,
25   totalSteps,
26 }) => {
27   const router = useRouter();
28   const currentPath = router.asPath;
29   const progress = (currentStep / totalSteps) * 100;
30   return (
31     <div className="flex min-w-[20rem] flex-col items-center justify-center gap-y-8 lg:w-[60rem]">
32       <Progress value={progress} />
33       <div className="flex items-center justify-center">
34         <h2 className="text-center text-xl">{questionData.question}</h2>
35       </div>
36       <div className="flex w-full flex-col items-center justify-center border lg:w-1/2">
37         <ul className="flex w-full flex-col gap-4">
38           {[ "a", "b", "c", "d" ].map((option) => (
39             <Button
40               key={option}
41               variant={"neutral"}
42               className={
43                 selectedOption === option
44                   ? "lg:text-md bg-sky-400 text-xs"
45                   : "lg:text-md text-xs"
46               }
47               onClick={() => onSelectOption(option)}
48             >
49               {questionData[option as keyof PropsQuestion]}
50             </Button>
51           ))}
52         </ul>
53       </div>
54       <div className="flex w-full items-center justify-center gap-x-10">
55         {currentPath === "/quiz/1" & 66 (
56           <Button onClick={onBack} variant={"neutral"} className="w-full">
57             Back
58           </Button>
59         )}
60         <Button onClick={onNext} variant={"neutral"} className="w-full">
61           Next
62         </Button>
63       </div>
64     </div>
65   );
66 };
67
68 export default QuizQuestion;
69

```

Apply data pertanyaan yang sudah kita buat kedalam file.tsx untuk menampilkan pertanyaan di UI.

```

1 import { useRouter } from "next/router";
2 import { quiz } from "~/features/detailQuiz/pages/index";
3 import QuizQuestion from "~/features/detailQuiz/pages/DetailQuiz";
4 import GridPattern from "~/components/ui/grid-pattern";
5 import { useEffect, useState } from "react";
6 import { Progress } from "~/components/ui/Progress";
7 import Head from "next/head";
8 import Swal from "sweetalert2";

```

Selanjutnya kita import beberapa file yang dibutuhkan di file index.tsx yang ada pada folder [id]

```
1  const router = useRouter();
2  const { id } = router.query;
3
4  const [currentStep, setCurrentStep] = useState(1);
5  const totalSteps = 5;
6
7  const [selectedAnswers, setSelectedAnswers] = useState<
8    Record<string, string | null>
9  >({});
10
11 const [timeLeft, setTimeLeft] = useState(300);
```

Tuliskan kode ini didalam export const fitur quiz yang mencakup router, karena kita akan berhubungan dengan halaman, lalu useState untuk menyimpan progress Ketika kita sudah menjawab dihalaman sebelumnya ataupun setelahnya, dan juga setTime sebagai menentukan waktu Ketika quiz (ex:300 detik)

```
1  const calculateScore = () => {
2    return quiz.reduce((score, question) => {
3      const selectedAnswer = selectedAnswers[question.id];
4      return selectedAnswer === question.answer ? score + 1 : score;
5    }, 0);
6  };
7
8  const timeOut = async () => {
9    localStorage.removeItem("quizTimeLeft");
10
11    await Swal.fire({
12      icon: "error",
13      title: "Oops ... ",
14      text: "Waktu Habis!",
15    });
16
17    void router.push({
18      pathname: "/quiz/score",
19      query: { score: calculateScore() },
20    });
21  };
```



Buat ini juga lanjutan dari yang atas sebagai menghitung score dan juga Ketika waktu nya sudah habis maka dipindahkan kehalaman score

```
1 const formatTime = (time: number) => {
2   const minutes = Math.floor(time / 60);
3   const seconds = time % 60;
4   return `${minutes}:${seconds < 10 ? "0" : ""}${seconds}`;
5 }
```

Buat kode ini untuk membuat format waktu yang akan ditampilkan di UI.

```
1 useEffect(() => {
2   const audio = new Audio("/kalkulus-sound.mp3");
3
4   void audio.play();
5   audio.loop = true;
6
7   return () => {
8     audio.pause();
9     audio.currentTime = 0;
10  };
11 }, []);
12
13 useEffect(() => {
14   const storedTime = localStorage.getItem("quizTimeLeft");
15   if (storedTime) {
16     setTimeLeft(parseInt(storedTime, 10));
17   }
18
19   const timer = setInterval(() => {
20     setTimeLeft((prevTime) => {
21       const newTime = prevTime - 1;
22       localStorage.setItem("quizTimeLeft", newTime.toString()); // Simpan waktu sisa di localStorage
23       if (newTime ≤ 0) {
24         clearInterval(timer);
25         void timeOut(); // Timer habis
26       }
27       return newTime;
28     });
29   }, 1000);
30
31   return () => clearInterval(timer);
32 }, []);
33
34 const questionData = quiz.find((question) => question.id === id);
35
36 if (!questionData) {
37   return <div>Question not found!</div>;
38 }
```

Buat useEffect juga sebagai bahan Ketika kita ngerender sebuah website apa yang akana terjadi (ex:munculnya soundtrack)

```

1  const handleNext = async () => {
2    const nextId = Number(id) + 1;
3    if (nextId ≤ quiz.length) {
4      await router.push(`/quiz/${nextId}`);
5    } else {
6      const result = await Swal.fire({
7        title: "Apakah Kamu Yakin Ingin Menyelesaikan Quiz ?",
8        text: "Anda tidak bisa mengubah jawaban setelah menyelesaikan quiz ini !",
9        icon: "question",
10       showCancelButton: true,
11       confirmButtonColor: "#3085d6",
12       cancelButtonColor: "#d333",
13       confirmButtonText: "Ya, Selesaikan Quiz !",
14     });
15     if (result.isConfirmed) {
16       await Swal.fire({
17         title: "Terima Kasih",
18         text: "Anda Telah Menyelesaikan Quiz Ini !",
19         icon: "success",
20       });
21
22       localStorage.removeItem("quizTimeLeft");
23
24       void router.push({
25         pathname: "/quiz/score",
26         query: { score: calculateScore() },
27       });
28     }
29   }
30   if (currentStep < totalSteps) {
31     setCurrentStep(currentStep + 1);
32   }
33 };

```

Tuliskan kode diatas untuk membuat user berpindah halaman ke halaman selanjutnya Ketika mengklik tombol next, dan juga mengarahkannya ke halaman score Ketika pertanyaannya sudah habis

```

1  const handleBack = async () => {
2    const prevId = Number(id) - 1;
3    if (prevId ≥ 1) {
4      await router.push(`/quiz/${prevId}`);
5    } else {
6      // Mungkin tampilkan pesan atau kembalikan ke halaman pertama
7      await router.push(`/quiz/1`);
8    }
9    if (currentStep > 1) {
10     setCurrentStep(currentStep - 1);
11   }
12 };

```

Kode yang hampir sama seperti diatas,namun kode ini dibuat untuk user bila ingin ke halaman sebelumnya

```

1  const handleSelectAnswer = (answer: string) => {
2      setSelectedAnswers((prev) => ({
3          ...prev,
4          [id as string]: answer,
5      }));
6  };
7  if (!id || Array.isArray(id)) {
8      return <div>Loading ... </div>; // Tetap tampilkan hooks sebelum render
9  }

```

Tuliskan kode diatas supaya Ketika user sudah menjawab pertanyaan nomor 1 dan pindah ke halaman selanjutnya, supaya jawaban yang sudah dipilih pada nomor 1 tidak hilang

```

1  return (
2      <
3          <Head>
4              <link rel="preconnect" href="https://fonts.googleapis.com" />
5              <link rel="preconnect" href="https://fonts.gstatic.com" />
6              <link
7                  href="https://fonts.googleapis.com/css2?family=Public+Sans:ital,wght@0,100..900;1,100..900&display=swap"
8                  rel="stylesheet"
9              />
10             <title>Quiz Question</title>
11             <meta name="description" content="Generated by create-t3-app" />
12             <link rel="icon" href="/kalkuluslogo.ico" />
13         </Head>
14         <main className="container mx-auto flex min-h-screen flex-grow p-4 dark:bg-darkBg dark:text-white">
15             <GridPattern className="-z-10" width={100} height={100} opacity={0.6} />
16             <div className="mx-auto flex min-h-screen flex-col items-center justify-center">
17                 <div className="my-5 flex items-center justify-between gap-x-10">
18                     <div>
19                         <h1 className="text-xl tracking-widest lg:text-2xl">
20                             SEMANGAT YA
21                         </h1>
22                     </div>
23                     <div className="rounded-lg border border-black p-2">
24                         <h1 className="text-xl tracking-widest lg:text-2xl">
25                             Waktu : {formatTime(timeLeft)}
26                         </h1>
27                     </div>
28                 </div>
29                 <QuizQuestion
30                     questionData={questionData}
31                     onNext={handleNext}
32                     onBack={handleBack}
33                     selectedOption={selectedAnswers[String(id)] ?? null}
34                     onSelectOption={handleSelectAnswer}
35                     currentStep={currentStep}
36                     totalSteps={totalSteps}
37                 />
38             </div>
39         </main>
40     </
41 );

```

Render beberapa fungsi yang sudah dibuat diatas tadi didalam return jsx nya, disini kita menggunakan fitur QuizQuestion, halaman yang sebelumnya kita buat, dan mengirimkan beberapa fungsi yang sudah dibuat di halaman ini ke halaman QuizQuestion

### III. Deploy

#### A. Setting Github Buat Repository

**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


*Required fields are marked with an asterisk (\*).*

**Repository template**

**No template** ▾


Start your repository with a template repository's contents.


**Owner \*** **Repository name \***

 ChandraAdipraja ▾ /

Great repository names are short and memorable. Need inspiration? How about **super-journey** ?

**Description** (optional)

☒  **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**



☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**

.gitignore template: **None** ▾

Membuat sebuah repository baru dengan nama bebas dan klik create

**Quick setup — if you've done this kind of thing before**

 Set up in Desktop or **HTTPS** **SSH** `https://github.com/ChandraAdipraja/project-dummy.git` 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# project-dummy" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ChandraAdipraja/project-dummy.git
git push -u origin main
```

**...or push an existing repository from the command line**

```
git remote add origin https://github.com/ChandraAdipraja/project-dummy.git
git branch -M main
git push -u origin main
```

Jika sudah seperti ini, kita Kembali ke visual studio code nya dan ketikan beberapa code git pada terminal

```
linearku on ʘ master via ʘ v20.9.0
● > git init
Reinitialized existing Git repository in D:/Coding/linearku/.git/
```

Pertama tama kita ketikan git init, untuk initialisasi awal git repository pada local kita

```
linearku on ʘ master via ʘ v20.9.0
● > git add .
```

Kedua ketikan git add untuk menambahkan pembaruan

```
linearku on ʘ master via ʘ v20.9.0
○ > git commit -m "first commit"
```

Ketiga ketikan git commit -m "your commit" supaya mengetahui waktu kita upload ke github itu apa yang kita ubah

```
linearku on ʘ master via ʘ v20.9.0
○ > git remote add origin https://github.com/ChandraAdipraja/project-dummy.git
```

Keempat kita remote project kita ke github repo yang sudah kita buat, supaya tau tujuan kita push kode ini ke repository mana

```
linearku on ʘ master via ʘ v20.9.0
○ > git push -u origin main
```

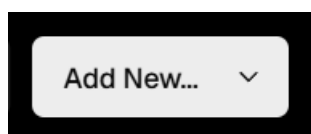
Gunakan push origin main untuk mengupload kode yang sudah kita buat ke repository yang sudah kita remote



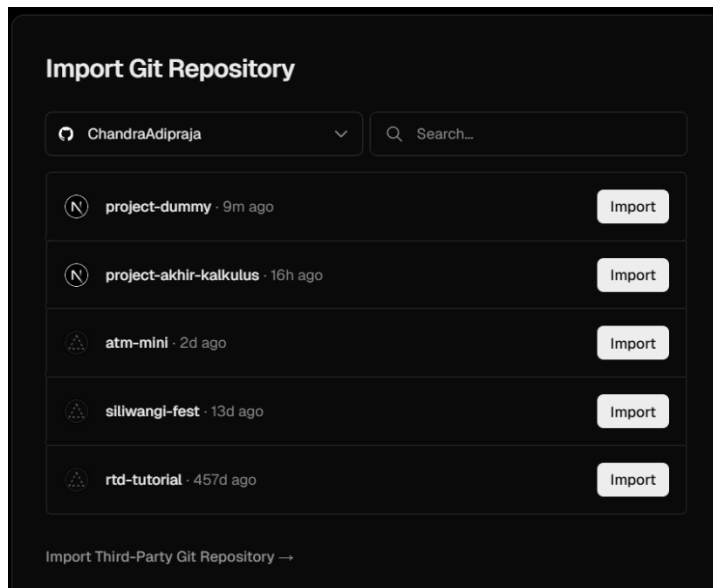
Maka tampilan repository kita akan berubah menjadi seperti ini, Ketika kita sudah berhasil push kodingan kita ke github

## B. Setting Vercel

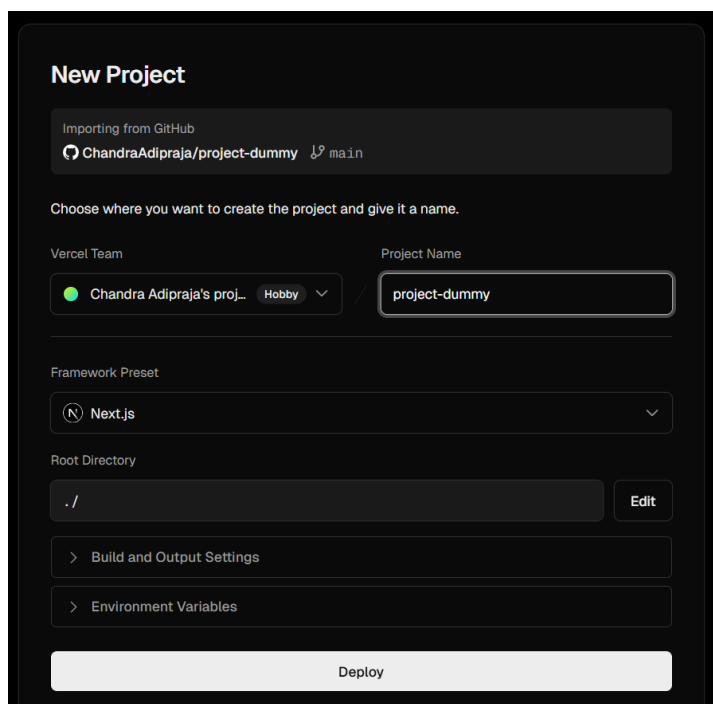
Login menggunakan Akun Github



Klik pada button add new dan pilih opsi project



Pilih repository yang Sudah kita buat tadi untuk di deploy



Selanjutnya klik deploy untuk mendeploy

project-akhir-kalkulus

[Repository](#)
[Usage](#)
[Domains](#)
[Visit](#)

Production Deployment

[Build Logs](#)
[Runtime Logs](#)
[Instant Rollback](#)

The deployment that is available to your visitors.

LinearKU

GRAFIK FUNGSI LINEAR

**Definisi Fungsi Linear**

Fungsi linear adalah suatu fungsi yang grafiknya berupa garis lurus. Fungsi linear memiliki rumus umum  $y = mx + c$ , di mana  $m$  adalah gradien (kemiringan) garis dan  $c$  adalah konstanta. Fungsi linear dapat digambarkan pada bidang kartesius dengan koordinat  $(x, y)$ .

Jika, fungsi linear adalah suatu fungsi yang memberikan grafik secara garis lurus. Fungsi linear ini juga menjadi fungsi yang lebih memudahkan untuk berhitung dengan variabelnya sama dengan satu.

RUMUS FUNGSI LINEAR

$y = mx + c$

RUMUS CARI TITIK POTONG

Bagaimana Cara Mencari Titik Potong Sumbu X ?

Bagaimana Cara Mencari Titik Potong Sumbu Y ?

Deployment

project-akhir-kalkulus-jy16oau1f-chandra-adiprajas-projects.vercel.app

Domains

[linearku.vercel.app](#)
[project-akhir-kalkulus-chandra-adiprajas-projects.vercel.app](#)

+1

Status

Created

Ready

16h ago by ChandraAdipraja

Source

main

c53a274 Add Timer Quiz,Confirm Modal,Backsound

To update your Production Deployment, push to the "main" branch.

Learn More

Jika sudah seperti ini maka project punya kalian sudah berhasil dideploy dan bisa diakses orang lain dengan jaringan yang berbeda beda

Pembuatan Aplikasi LinearKU sudah selesai,dengan ini linearKU sudah bisa diakses oleh orang lain menggunakan url linearku.vercel.app.