Tutorial Pembuatan Educandy

saya akan menjelaskan cara membuat aplikasi Anda. Berikut adalah langkah-langkah umum untuk membuat aplikasi React dengan Vite, Tailwind CSS, dan DaisyUI:

Untuk memulai pembuatan aplikasi, kita harus paham tentang unsur utama pembangun website, HTML, CSS, Javascript

Sebelum melakukan langkah-langkah, kita harus menginstal beberapa software seperti

1. NodeJS, sebagai lingkungan untuk menjalankan Javascript diluar peramban web, [Node.js — Run JavaScript Everywhere (nodejs.org)](https://nodejs.org/en), npm dan npx tersedia bersama paket ini saat anda menginstall
2. Text Editor, sebagai penyunting kode-sumber buatan Microsoft, [Download Visual Studio Code - Mac, Linux, Windows](https://code.visualstudio.com/Download)
3. Microsoft Edge, sebagai peramban web (Edge biasanya sudah terinstal di device anda, anda bisa menggunakan chrome atau peramban lainna, opsional)

Siapkan bahan bahan   
1. suara hijaiyahnya dengan text-to-speech Arabic ([Text To Speech Arabic (narakeet.com)](https://www.narakeet.com/languages/text-to-speech-arabic/))

2. buat file json untuk menyimpan data yang akan kita gunakan untuk data hijaiyah nanti

[

    {

*"arabic"*: "ا",

*"latin"*: "alif",

*"braille"*: [1,0,0,0,0,0]

    },

    {

*"arabic"*: "ب",

*"latin"*: "ba",

*"braille"*: [1,0,1,0,0,0]

    },

    {

*"arabic"*: "ت",

*"latin"*: "ta",

*"braille"*: [0,1,1,1,1,0]

    },

...

]

3. dapatkan gambar untuk icon aplikasi, lalu pergi ke Favicon.io ([Favicon Generator - Image to Favicon - favicon.io](https://favicon.io/favicon-converter/)) untuk menyiapkan paket icon yang digunakan sebagai icon aplikasi

4. dapatkan font-family untuk mengubah font awal menjadi lebih kekinian, anda bisa mendapatkan poppins di google font ([Poppins - Google Fonts](https://fonts.google.com/specimen/Poppins))

Mari kita amasuk ke acara

1. Inisialisasi Proyek

* Buat Proyek

npm create vite@latest

* Masuk ke direktori proyekmelalui proses penginstalan pilih React sebagai framework dan pilih Javascript sebagai variant

1. Instal Dependensi

Buka Text Editor anda lalu masuk ke folder

* Install dependency yang diperlukan

npm install react-router-dom react-icons react-qr-code react-qrcode-scanner use-sound react-error-boundary

1. react-router-dom: Library ini digunakan untuk menangani routing dalam aplikasi React.
2. react-icons: Sebuah library yang menyediakan koleksi besar icon populer sebagai komponen React.
3. react-qr-code: Library ini digunakan untuk menghasilkan QR code dalam aplikasi React
4. react-qrcode-scanner: Komponen React untuk memindai QR code menggunakan kamera perangkat
5. use-sound: Sebuah React hook untuk memutar suara dengan mudah dalam aplikasi
6. react-error-boundary: Komponen untuk menangani error dalam aplikasi React secara graceful. (opsional)

* Install Tailwind

untuk menginstall tailwind dengan vite, silahkan kunjungi laman berikut untuk detailnya [Install Tailwind CSS with Vite - Tailwind CSS](https://tailwindcss.com/docs/guides/vite)

* Install Daisyui

Untuk menginstall daisyui, silahkan kunjungi laman berikut untuk detailnya [Install daisyUI as a Tailwind CSS plugin — Tailwind CSS Components ( version 4 update is here )](https://daisyui.com/docs/install/)

1. Implementasi komponen

Buat komponen dengan jsx lalu design dengan tailwindcss, beberapa komponen juga sudah tersedia di daisyui ([Components — Tailwind CSS Components ( version 4 update is here ) (daisyui.com)](https://daisyui.com/components/))

Saya menggunakan beberapa komponen yang diambil dari daisyui seperti:

1. Navbar [Tailwind Navbar Component — Tailwind CSS Components ( version 4 update is here ) (daisyui.com)](https://daisyui.com/components/navbar/)
2. Hero [Tailwind Hero Component — Tailwind CSS Components ( version 4 update is here ) (daisyui.com)](https://daisyui.com/components/hero/)
3. Footer [Tailwind Footer Component — Tailwind CSS Components ( version 4 update is here ) (daisyui.com)](https://daisyui.com/components/footer/)
4. Card [Tailwind Card Component — Tailwind CSS Components ( version 4 update is here ) (daisyui.com)](https://daisyui.com/components/card/)

untuk kustomisasi lebih seperti layout dan perubahan bentuk komponen pada setiap dimensi device dan lain lain, kita harus melakukannya manual dengan CSS, saatnya kita gunakan framework tailwindcss ([Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.](https://tailwindcss.com/)) untuk mempersingkat waktu pembuatan design komponen dari nol

1. Menampilkan Hijaiyah
2. Import

import hurufHijayah from '../utils/hijaiyah.json'

File JSON ini berisi array objek, di mana setiap objek mewakili satu huruf Hijaiyah

1. Struktur komponen Hijaiyah

export default function Hijaiyah() {

*// ... kode lainnya ...*

return (

<div className="flex flex-col gap-2 h-full overflow-auto p-4">

{*/\* ... judul dan deskripsi ... \*/*}

<div className='flex flex-wrap gap-2 items-center justify-center'>

{hurufHijayah.map(x => <Card item={x} key={x.latin}/>)}

</div>

</div>

)

}

1. Penggunaan map untuk menampilkan Card

{hurufHijayah.map(x => <Card item={x} key={x.latin}/>)}

* Ini mengiterasi setiap objek dalam array hurufHijayah.
* Untuk setiap objek, komponen Card dibuat dengan prop item yang berisi objek huruf tersebut.
* key diset menggunakan x.latin untuk optimisasi rendering React.

1. Penggunaan Objek hijaiyah dalam komponen Card

export default function Card({item}) {

const [play] = useSound(`/hijaiyah/${item.latin}.mp3`)

return (

<div className="card bg-base-100 shadow-xl w-full max-w-sm text-neutral-900">

<div className="px-10 pt-10">

<button onClick={play}>{item.arabic}</button>

</div>

<div className="card-body flex flex-row justify-between">

<Braille cell={item.braille}/>

<h2 className="card-title">{item.latin}</h2>

{*/\* ... \*/*}

</div>

{*/\* ... modal content ... \*/*}

</div>

)

}

* item.arabic digunakan untuk menampilkan karakter Arab.
* item.latin digunakan untuk nama file audio, judul, dan nilai QR code.
* item.braille digunakan untuk menampilkan representasi Braille.

1. Komponen Braille

function Braille({cell}) {

return <div className="grid grid-cols-2">

{cell.map((x, i) => x ? <FaDotCircle key={i}/> : <FaRegDotCircle key={i}/>)}

</div>

}

Menggunakan array cell (yang adalah item.braille) untuk menampilkan titik-titik Braille.

1. Penerapan dependency
2. react-router-dom
3. Setup di main.jsx:

* <BrowserRouter> digunakan untuk membungkus seluruh aplikasi. Ini memungkinkan routing berbasis browser di aplikasi Anda.
* Semua komponen di dalam <BrowserRouter> bisa menggunakan fitur-fitur routing.

1. Penggunaan di App.jsx:

* useNavigate(): Ini adalah hook yang memberikan fungsi untuk berpindah halaman secara programatis. Anda menggunakannya untuk navigasi saat mengklik menu.
* useLocation(): Hook ini memberikan informasi tentang URL saat ini. Anda menggunakannya untuk menyorot menu yang aktif.

1. Mendefinisikan rute:

* <Routes> digunakan untuk mengelompokkan semua rute dalam aplikasi.
* Setiap <Route> mendefinisikan satu rute:
  + path="/" adalah homepage
  + path="/hijaiyah" mengarah ke komponen Hijaiyah
  + path="/pindai" mengarah ke komponen Scan
  + path="/qr" mengarah ke komponen Qrcode

1. Navigasi:

* Anda menggunakan navigate('/') untuk berpindah ke homepage saat logo diklik.
* Untuk menu lain, Anda menggunakan navigate('/hijaiyah'), navigate('/pindai'), dan navigate('/qr') saat item menu diklik.

1. react-router-dom

1) Impor QRCode

import QRCode from "react-qr-code"

Ini mengimpor komponen QRCode dari library react-qr-code

2) Penggunaan QRCode

<QRCode value={item.latin} level="L"/>

Komponen QRCode digunakan file Card.jsx dan Qrcode.jsx. Berikut penjelasannya:

* value={item.latin}: Ini adalah data yang akan dikodekan menjadi QR code. Dalam kasus ini, nilai Latin dari karakter Hijaiyah.

1. react-qrcode-scanner

1) Komponen Qrcode

<QrScanner

onError={handleError}

onScan={handleScan}

style={{height: 'full', width: '100%'}}

flipHorizontally={mirror}

facingMode={facingMode ? 'environment' : 'user'}

/>

* onError: Menangani error yang mungkin terjadi saat pemindaian.
* onScan: Callback yang dipanggil ketika QR code berhasil dipindai.
* flipHorizontally: Membalik tampilan kamera secara horizontal.
* facingMode: Mengatur kamera mana yang digunakan (depan atau belakang).

2) Penanganan hasil scan

const handleScan = (data) => {

console.log('scanned', data);

setScanResult(data)

}

Fungsi ini dipanggil ketika QR code berhasil dipindai, menyimpan hasilnya ke dalam state scanResult.

3) Pemutaran suara

 useEffect(() => {

if (!scanResult) return

const audio = new Audio(`/hijaiyah/${scanResult}.mp3`);

audio.play();

setTimeout(() => {

setScanResult('')

}, 1800);

}, [scanResult])

* Effect ini berjalan setiap kali scanResult berubah.
* Jika ada hasil scan, audio yang sesuai akan diputar.
* Nama file audio diambil dari hasil scan (misalnya, jika hasil scan adalah "alif", akan memutar "/hijaiyah/alif.mp3").
* Setelah 1.8 detik, scanResult direset.

4) control kamera

* setShowCam: Mengontrol visibilitas kamera.
* setMirror: Mengontrol apakah tampilan kamera dibalik secara horizontal.
* setFacingMode: Mengontrol penggunaan kamera depan atau belakang.

1. use-sound

1) Import useSound

import useSound from "use-sound";

Ini mengimpor hook useSound dari library use-sound.

const [play] = useSound(`/hijaiyah/${item.latin}.mp3`)

* useSound digunakan untuk membuat fungsi yang akan memutar file audio.
* Path audio dibuat secara dinamis berdasarkan item.latin.
* Fungsi play yang dihasilkan dapat dipanggil untuk memutar audio.

2) Penggunaan useSound

const [play] = useSound(`/hijaiyah/${item.latin}.mp3`)

* useSound digunakan untuk membuat fungsi yang akan memutar file audio.
* Path audio dibuat secara dinamis berdasarkan item.latin.
* Fungsi play yang dihasilkan dapat dipanggil untuk memutar audio.

3) Penerapan pada tombol

<button className="btn btn-secondary rounded-xl p-4 place-items-center text-center text-5xl w-full h-full" onClick={play}>{item.arabic}</button>

* Tombol ini menampilkan karakter Arab (item.arabic).
* Ketika diklik, fungsi play akan dipanggil, memutar audio yang sesuai.

1. Menampilkan logo unwahas selama dua detik di halaman home

* Tambah referensi dan fungsi untuk melakukan klik pada elemen

*const* toggleSplashRef = useRef(null)

*function* toggleSplash() {

    toggleSplashRef.current.click()

  }

* Gunakan kode berikut untuk menampilkan logo unwahas

<label htmlFor="splash" className="btn" ref={toggleSplashRef}/>

      <input type="checkbox" id="splash" className="modal-toggle" />

      <div className="modal text-neutral-700" role="dialog">

        <div className="modal-box">

          <img src="/unwahas-512x512.png" alt="unwahas" />

        </div>

      </div>

Elemen modal akan ditampilkan ketika input di klik, label juga bisa melakukannya, kita sudah meyimpan referensi table dan akan menggunakannya nanti

* Tambahkan efek ketika komponen dirender

useEffect(() *=>* {

*const* splashShown = sessionStorage.getItem('splashShown');

    if (!splashShown) {

      toggleSplash()

      sessionStorage.setItem('splashShown', 'true');

      setTimeout(() *=>* {

        toggleSplash()

      }, 2000);

    }

  }, []);

Kode diatas untuk mendapatkan data splashShown di penyimpanan sesi kali ini berlangsung, lalu melakukan kondisi ketika splashShown tidak ditemukan maka tampilkan logo dengan toggleSplash() lalu tambah data di penyimpanan sesi untuk menandai sesi kali ini sudah dilakukan, setelah dua detik berlalu splash akan hilang oleh toggleSplash() yang dijalankan berikutnya

1. Pemasangan icon

* Tambahkan manifest.json pada root directory lalu isi seperti berikut

{

*"name"*: "Educandy",

*"icons"*: [

        {

*"src"*: "/android-chrome-192x192.png",

*"sizes"*: "192x192",

*"type"*: "image/png",

*"purpose"*: "any"

        },

        {

*"src"*: "/android-chrome-512x512.png",

*"sizes"*: "512x512",

*"type"*: "image/png"

        },

        {

*"src"*: "/favicon-16x16.png",

*"sizes"*: "16x16",

*"type"*: "image/png"

        },

        {

*"src"*: "/favicon-32x32.png",

*"sizes"*: "32x32",

*"type"*: "image/png"

        }

    ],

*"start\_url"*: "/",

*"display"*: "standalone"

}

Kemudian hubungkan manifest.json ke website kita dengan menambahkan baris berikut di tag head HTML

<link rel="manifest" href="/manifest.json">

Mari kita host ke layanan hosting

* + 1. Buka [GitHub](https://github.com/), daftar atau masuk ke akun anda lalu buat repository baru, beri nama educandy lalu masukkan kode command line berikut di terminal folder directory untuk mengunggah kesana

echo "# educandy" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M main

git remote add origin https://github.com/Forisma/educandy.git

git push -u origin main

* + 1. Buka Netlify ([Scale & Ship Faster with a Composable Web Architecture | Netlify](https://www.netlify.com/)) untuk menjalankan aplikasi kita, daftar atau masuk ke akun anda lalu tambahkan situs baru, impor project yang sudah ada, pilih deploy dengan Github, pilih educandy sebagai repository, terakhir konfigurasi situs seperti memberi nama domain lalu deploy, tunggu saat dan website educandy sudah aktif