

Perancangan Prototipe Website Menggunakan Reactjs Dan Ant Design Ui Library

Yosua Alvin Adi Soetrisno¹, Eko Handoyo², Enda Wista Sinuraya³, Denis⁴, Hadha Afrisal⁵

^{1,2,3,4,5}Teknik Elektro, Fakultas Teknik, Universitas Diponegoro

E-mail: *¹ yosua@live.undip.ac.id

Abstrak

kodingworks merupakan sebuah perusahaan yang menggunakan sistem software as a service (saas), yang artinya perusahaan lebih fokus membuat pelayanan dari aplikasi dan bukan menjual aplikasinya. salah satu fokus kodingworks sebagai penyedia perangkat lunak adalah bagaimana perangkat lunak tersebut dapat berkomunikasi dengan pelanggan melalui antarmuka yang mudah digunakan dan nyaman dilihat, hal ini merupakan proses dari pengembangan antarmuka yang ditanggungjawab oleh seorang atau tim pengembang. salah satu metode untuk membuat antarmuka adalah menggunakan metode membuat prototipe. hasil dari protipe merupakan sebuah aplikasi yang sudah berfungsi dan dapat disebar di website, yang nantinya akan di tes oleh tim pengetes. dalam pembuatannya terdapat alur yang sudah direncanakan oleh perusahaan untuk mencegah terjadinya kesalahan dan memastikan keluaran produk yang dikeluarkan sesuai dengan standar yang berlaku di pasaran.

Kata Kunci— *pelayanan, pengembangan antarmuka, prototipe, website, perangkat lunak.*

1. PENDAHULUAN

Era dimana perkembangan teknologi berkembang sangat pesat, tentunya cara kita berkomunikasi dan menyebarkan informasi mengalami evolusi dari yang awalnya media cetak mendominasi pasar informasi hingga tren yang bergeser ke media virtual seperti media sosial. Tentunya dalam menyampaikan informasi kita tidak hanya dibatasi oleh medium tulisan saja terlebih di abad ini dimana penyampaian secara *visual* menjadi tonggak penting dalam menyampaikan informasi mengingat minat baca anak muda yang semakin menurun. *Tech writer* dan *software tester* adalah contoh dari pekerjaan yang lahir dari perkembangan teknologi. Berfungsi sebagai jembatan antara developer program yang memiliki kemampuan teknis dengan user yang berasal dari berbagai kalangan dan tingkat pendidikan, kedua pekerjaan ini memiliki tugas memastikan program dapat berjalan sesuai kebutuhan, sesuai alur, dan tidak ada kesalahan atau error pada program.

Website merupakan salah satu bentuk produk dalam menyampaikan informasi. Untuk berkomunikasi dengan pengguna tentunya kita butuh merancang website sedemikian rupa agar mudah dimengerti dan digunakan pengguna, maka dari itu inilah peran seorang *frontend developer* dibutuhkan. *Frontend developer* berfungsi sebagai jembatan antara desainer dan *backend developer* untuk memastikan *website* yang dibuat tidak hanya berfungsi tapi juga memiliki estetika di mata pengguna.

Sebelum sebuah website diterbitkan kita perlu melakukan uji coba dulu dengan membuat prototipe. Prototipe sendiri adalah sebuah model awal untuk mencoba apakah produk yang dibuat sesuai dengan konsep atau proses yang sudah dirancang sebelumnya. Ini merupakan proses yang penting mengingat kita hampir tidak mungkin sukses dalam percobaan pertama, setidaknya dengan prototipe kita bisa memberikan bagaimana konsep ini akan berkembang kedepannya..

2. METODE PENELITIAN

Penelitian ini menggunakan vs code (visual studio code) sebagai ide (*integrated development environment*). Ide merupakan perangkat lunak yang digunakan untuk mengedit kode, mengintegrasikan kode, melakukan *autocomplete* dengan berbagai bahasa pemrograman yang disediakan. Ide ini dikembangkan oleh microsoft dengan *electron framework*. Vs code menyediakan fitur seperti penyorotan sintaksis, penyelesaian kode, kutipan kode, merefaktor kode, pengawakutuan, dan fasilitas git yang sudah tertanam.

2.1. Typescript

Typescript adalah bahasa pemrograman yang bersifat gratis dan open-source yang dikembangkan oleh microsoft. *Typescript* sebenarnya merupakan pengembangan dari bahasa pemrograman lainnya yaitu javascript yang berfokus untuk mempermudah pengembang untuk menemukan bug pada kode yang sudah dibuat dengan menambah sintaks pada tipe sistem atau mengatur tipe data pada javascript. Hal ini tentunya dapat mempersingkat waktu dalam membuat kode agar lebih efektif dan mudah dibaca. Penelitian ini menggunakan typescript untuk membantu pembentukan tipe data. Selain itu typescript memberikan inferensi dan anotasi tipe. Dengan terhubung ke language server maka perancangan programnya akan diberi asistensi kode sehingga akan lebih mudah dalam memakai fungsi serta kelas yang tersedia.

2.2. React

React adalah *library* javascript buatan facebook untuk membantu pengembang membangun ui (*user interface*) pada aplikasi mobile dan web. React sendiri berisi kumpulan potongan kode javascript yang biasa disebut sebagai komponen. Komponen merupakan campuran dari html (*hypertext markup language*) dan javascript yang berisi kumpulan logika yang digunakan untuk menampilkan ui dan dapat digunakan berulang-ulang. React pada penelitian ini dipakai backend yang menjembatani interface database dan pengaksesan fungsi untuk *create read update delete (crud)* pada *form* yang akan dirancang pada *ant design*

2.3. Ant Design

Ant design merupakan sebuah react ui library yang berisi komponen yang 'siap pakai' seperti makanan siap saji. Ant design menyediakan berbagai jenis komponen dan elemen yang sering digunakan untuk developer yang tentunya dapat menghemat developer dalam membuat ui tanpa harus sering mengubah styling komponen. Sifatnya yang mudah diintegrasikan dengan sistem dan dokumentasi yang cukup jelas, sangat berguna untuk membuat tampilan website yang lebih estetik dan interaktif. Pada penelitian ini ant design digunakan untuk membuat komponen gui pada react js.

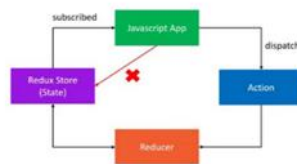
2.4. React Router

React adalah *library* javascript buatan facebook untuk membantu pengembang membangun ui (*user interface*) pada aplikasi mobile dan web. React sendiri berisi kumpulan cuplikan kode javascript yang biasa disebut sebagai komponen. Komponen merupakan campuran dari html (*hypertext markup language*) dan javascript yang berisi kumpulan logika yang digunakan untuk menampilkan ui dan dapat digunakan berulang-ulang. Pada penelitian ini setiap alamat yang diakses pada aplikasi nantinya diarahkan oleh router ke fungsi load tampilan aplikasinya sesuai dengan struktur yang sudah dirancang.

2.5. Manajemen State

Komponen react memiliki objek state bawaan. Objek state (atau state untuk singkatnya) adalah tempat anda menyimpan nilai properti yang dimiliki komponen tersebut. Ketika state berubah, komponen merender ulang. Kesimpulannya state merupakan sebuah objek untuk menyimpan data pada react dan akan di render atau muat ulang ketika data mengalami perubahan. Konsepnya adalah sebuah cara bagi react untuk mengingat nilai-nilai yang disimpan dari sebuah komponen. Ini mirip dengan mendeklarasikan variabel dari dalam komponen, dengan beberapa bonus tambahan.

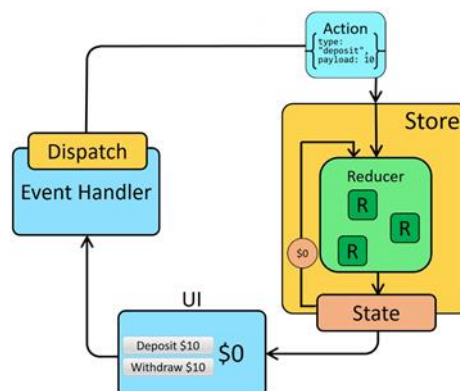
Three Principles Overview



Seiring membuat website tentunya kita membuat banyak state yang berhubungan dengan komponen lainnya, hal ini dapat menyebabkan kode sulit untuk dibaca dan banyak kasus dimana kode yang sama ditulis berulang-ulang. Untuk mencegah hal tersebut maka dibutuhkan sebuah sistem dimana logika dan tampilan dipisahkan untuk mengatur cara kerja state tersebut agar berfungsi menggunakan logika yang dapat digunakan berulang-ulang. Dalam membuat prototipe ini saya menggunakan 2 library untuk manajemen state, yaitu:

2.5.1. React Redux

React redux merupakan gabungan dari react dan redux dengan react seperti yang sudah dijelaskan diatas. Redux sendiri merupakan library yang bisa digunakan secara mandiri di berbagai framework seperti react, vue, angular maupun vanilla javascript. Konsepnya dalam redux terdapat tiga prinsip yang perlu diperhatikan yaitu store, action dan reducer. Dimulai dari action dimana kita mendeskripsikan ‘aksi’ apa yang akan kita lakukan dengan action berisi objek javascript yang sudah dideskripsikan. Selanjutnya ada reducer yang berfungsi untuk mengeksekusi action dan memutuskan bagaimana state diperbarui, reducer berisi fungsi-fungsi yang mengambil state sebelumnya dan action dan mengembalikan nilai state yang baru. Dan terakhir ada store dimana tempat state disimpan, ini memudahkan untuk membuat aplikasi yang universal, karena state dari server dapat diserialisasikan dan dihidrasi ke klien tanpa kode tambahan.



Gambar 1. ilustrasi konsep redux

2.5.2. Redux Persist

Permasalahan dari redux adalah apabila kita mencoba untuk menyegarkan halaman, state akan kembali ke nilai awal. Tentunya hal ini cukup menyulitkan apabila kita ingin state tersebut mempertahankan nilai yang baru. Maka dari itu kita menggunakan redux persist yang digunakan untuk menyimpan nilai state yang baru ke dalam persisted storage seperti local atau session storage. Redux persist juga menyertakan metode yang memungkinkan kita menyesuaikan state yang dipertahankan dan direhidrasi, semuanya dengan api yang mudah diintegrasikan.

2.6. Alur Pengerjaan Tugas di Kodingworks

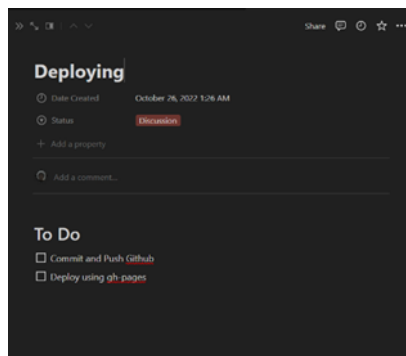
Dalam pelaksanaan magang di kodingworks dalam mengerjakan pekerjaan terdapat alur yang sudah dibuat oleh perusahaan. Hal ini dibuat agar dapat memudahkan melakukan evaluasi dan juga mencegah terjadinya miskomunikasi.

Dikarenakan tidak bisa membagikan semua hal-hal teknis yang saya dapat di kodingworks, disini saya hanya akan membagikan garis besarnya saja. Berikut adalah alurnya :

2.6.1. Mengatur dan Mengorganisir Tugas

Organisir dan mengatur tugas di kodingworks dilakukan menggunakan aplikasi notion. Untuk koordinasinya dibantu oleh aplikasi sosial media telegram untuk berkomunikasi. Berikut adalah langkah-langkahnya :

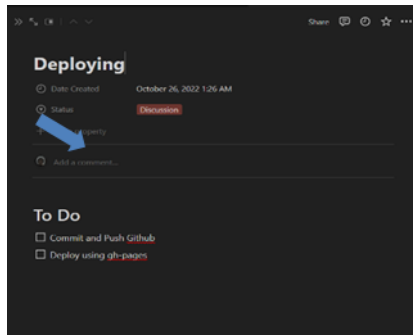
1. Mengambil atau membuat kartu tugas di kolom discussion.



Gambar 2. contoh pembuatan kartu tugas

Langkah awal yang dilakukan adalah mengambil kartu tugas yang sudah disediakan atau apabila memang mau memulai tugas dari awal maka kita diharuskan membuat kartu tugas yang baru dari awal. Kartu tugas berisi deskripsi tugas, jenis tugas dan apa yang akan dilakukan beserta targetnya (seperti target waktu atau target capaian kerja)

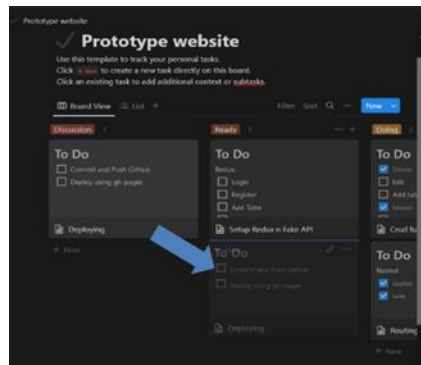
2. Konfirmasi pengambilan tugas di kolom komentar.



Gambar 3. konfirmasi pengambilan tugas di kolom komentar

Apabila langkah pertama sudah dilakukan maka langkah selanjutnya adalah meminta konfirmasi ke pembimbing atau penanggungjawab tugas tersebut. Konfirmasi ditulis di kolom komentar kartu tugas lalu mengabari pembimbing melalui telegram.

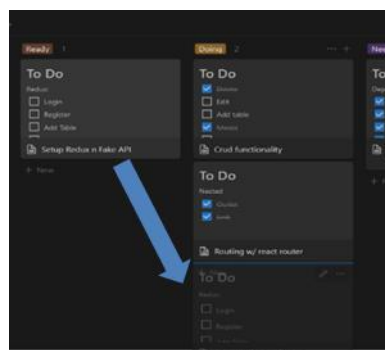
3. Menunggu kartu tugas dipindahkan ke kolom ready oleh pembimbing.



Gambar 4. tugas siap saat kartu dipindah ke kolom ready

Selanjutnya apabila tugas tidak disetujui (biasanya terjadi bila isi kartu tidak lengkap) maka kita akan disuruh untuk merubah atau mengganti kartu tugas tersebut, dan apabila disetujui biasanya pembimbing akan mengabari bahwa kartu tugas sudah dipindah ke kolom ready yang artinya tugas sudah bisa dikerjakan.

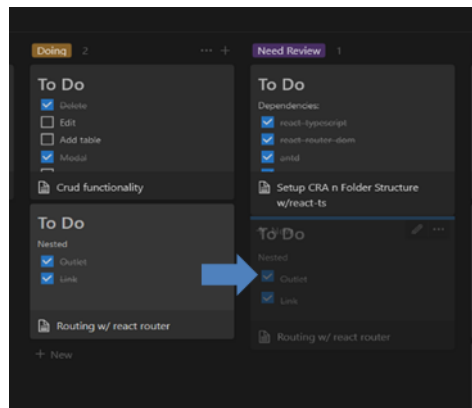
4. Memindahkan kartu tugas ke kolom doing apabila akan dikerjakan



Gambar 5. kartu dipindah ke kolom doing saat mau dikerjakan

Apabila kita sudah mulai mengerjakan tugas maka sebelumnya kartu tugas perlu dipindah ke kolom doing terlebih dahulu.

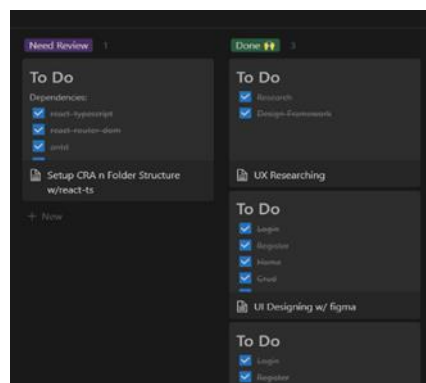
5. Apabila sudah selesai pindahkan kartu tugas ke kolom need review untuk dievaluasi.



Gambar 6. kartu dipindah ke kolom need review saat sudah selesai

Langkah selanjutnya apabila tugas sudah selesai maka kita perlu memindahkan kartu tugas ke kolom need review untuk selanjutnya diproses untuk evaluasi oleh pembimbing. Pastikan seluruh target sudah tercapai dengan menambahkan bukti pengerjaan seperti tangkapan layar atau berbagi link hasil pekerjaan.

6. Konfirmasi lagi apabila tugas sudah yakin selesai.
Jangan lupa untuk konfirmasi lagi seperti di langkah kedua untuk meminta evaluasi dari tugas yang sudah dikerjakan.
7. Menunggu hasil evaluasi.



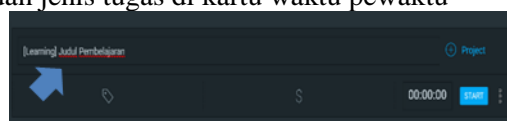
Gambar 7. tunggu evaluasi selesai

Tugas sudah dianggap selesai apabila pembimbing sudah mengabari bahwa tugas selesai dan kartu tugas dipindahkan ke kolom done. Apabila saat proses evaluasi terdapat masalah maka tugas perlu diperbaiki sebelum akhirnya bisa disetujui.

2.6.2. Melacak Waktu Kerja

Aplikasi clockify merupakan aplikasi pelacak waktu kerja yang digunakan di kodingworks. Hal ini digunakan untuk membandingkan target waktu kerja mingguan yang sudah disepakati dengan waktu kerja yang sebenarnya dilakukan. Berikut adalah langkah-langkah cara menggunakannya :

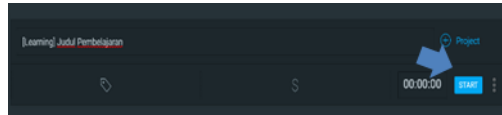
1. Memasukkan nama dan jenis tugas di kartu waktu pewaktu



Gambar 8. membuat kartu pewaktu

Apabila tugas sudah disetujui maka kita bisa mulai mengerjakan tugas, tapi sebelum tugas dikerjakan kita perlu menyiapkan clockify terlebih dahulu. Langkah pertama adalah mengisi nama tugas sesuai format yang sudah diberikan. Jangan lupa untuk menambah jenis tugas di tombol project.

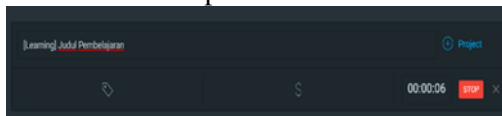
2. Menyalakan stopwatch



Gambar 9. memulai stopwatch

Tekan tombol start untuk memulai stopwatch sebelum mengerjakan tugas.

3. Apabila sudah selesai atau istirahat stopwatch diberhentikan



Gambar 10. memberhentikan stopwatch

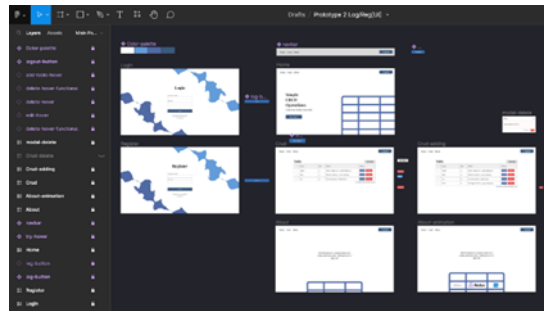
Tekan tombol stop apabila tugas sudah selesai atau butuh melakukan istirahat seperti izin ke belakang atau keperluan diluar kerja. Nantinya di akhir minggu akan ada evaluasi apabila waktu yang dikerjakan kurang dari waktu yang sudah disepakati atau apabila kualitas pekerjaan tidak sesuai dengan waktu kerja.

3. HASIL DAN PEMBAHASAN

Prototipe dibuat untuk membuat konsep yang dibuat menjadi nyata namun berada dalam kondisi dimana produk belum dapat dipasarkan. Dengan membuat prototipe, kita dapat melihat bagian mana yang perlu dirubah atau dibiarkan. Ini juga adalah kesempatan untuk mengumpulkan umpan balik dari tester dan menyempurnakannya. Berikut adalah kurang lebih garis besar yang saya lakukan dalam membuat prototipe website saat kerja praktik.

3.1. Mengecek Desain

Sebagai frontend developer kita ditugaskan untuk mengubah desain menjadi website yang dapat berfungsi. Maka dari itu tentunya dalam proses frontend development cukup wajar apabila dimulai dengan melakukan riset terhadap desain yang akan dibuat. Frontend developer sebenarnya tidak diwajibkan untuk membuat desain sendiri biasanya terdapat tim terpisah yang bertanggung jawab untuk membuat desain, biasanya terdiri dari ui (user interface) dan ux (user experience) designer. Jadi yang dimaksud riset disini adalah bagaimana seorang frontend developer dapat membayangkan proses rencana yang akan dilakukan dari desain yang sudah dilihat seperti langkah-langkah yang perlu diperhatikan dan alat atau hal pendukung yang dibutuhkan dalam membuat website. Dibawah ini adalah contoh dari desain yang saya coba saat magang yang saya buat sendiri.



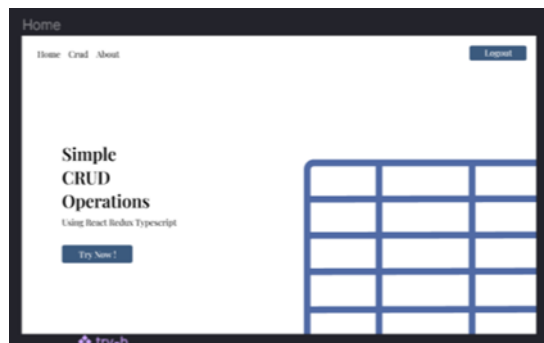
Gambar 11. contoh desain website keseluruhan



Gambar 12. contoh desain komponen navbar



Gambar 13. contoh desain halaman login



Gambar 14. contoh desain halaman home

3.1.1. Setup Dan Manajemen Folder

Untuk menginisiasi membuat aplikasi react masuk ke terminal dan pastikan anda berada di folder yang akan anda gunakan lalu buka terminal. Untuk memulai proyek baru kita butuh menginisiasi dengan mengetikkan `npx create-react-app my-app` di terminal ini berfungsi untuk men-setup aplikasi react. Lalu apabila proses inisiasi sudah selesai mulai aplikasi react dengan mengetikkan `npm start` di terminal.

Apabila tidak ada masalah lanjut dengan menginstall dependencies , dependencies adalah kode pihak ketiga yang kita pakai untuk memudahkan dalam membangun aplikasi react. Biasanya

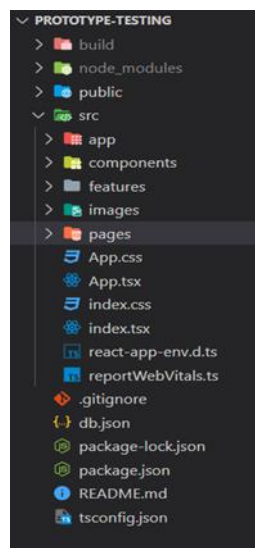
dependencies diinstall menggunakan npm (node package manager). Gambar dibawah adalah beberapa dependencies yang akan dipakai dalam membuat protipe website ini.

```
"dependencies": {
  "@ant-design/icons": "^4.7.0",
  "@reduxjs/toolkit": "^1.8.5",
  "@testing-library/jest-dom": "^5.16.5",
  "@testing-library/react": "^13.4.0",
  "@testing-library/user-event": "^14.4.3",
  "antd": "^4.23.0",
  "axios": "^0.27.2",

  "react": "^18.2.0",
  "react-dom": "^18.2.0",
  "react-redux": "^8.0.2",
  "react-router-dom": "^6.3.0",
  "react-scripts": "5.0.1",

  "redux-persist": "^6.0.0",
  "unique-names-generator": "^4.7.1",
  "web-vitals": "^2.1.4"
},
```

Gambar 15. list dependencies yang dibutuhkan



Gambar 16. manajemen folder

Langkah selanjutnya adalah memajemen folder dengan cara membuat folder yang nantinya akan mengelompokkan file sesuai jenis atau fungsinya. Ada 4 folder yang digunakan untuk membuat prototipe ini yaitu app untuk redux store dan hooks, components untuk komponen yang digunakan berulang, images sesuai namanya untuk gambar dan pages untuk komponen halaman. File diluar folder dianggap sebagai file global. Harap diingat ini hanya dilakukan di folder src diluar itu kita tidak merubah apapun.

3.1.2. Layouting Dan Styling

Langkah selanjutnya adalah melakukan layouting dan styling. Dimana layouting adalah proses untuk membuat kerangka atau desain kasar dari kode yang akan dibuat dan styling adalah proses dimana sebisa mungkin kode yang dibuat mirip dengan desain yang sudah direncanakan

sebelumnya. Dibawah ini adalah beberapa contoh dari proses layouting dengan referensi desain dari gambar 17, seperti berikut :

```
//layout halaman home
<div>
  <div>
    <typography.title>//untuk judul
      judul
    </typography.title>
    <typography.text>
      subjudul
    </typography.text>
    <button />
  </div>
  <img />
</div>
//layout halaman login
<div>
  <div>
    <h1>judul</h1>
    <form>//komponen form berisi 2 input dan 1 button
      <form.item>
        <input />
      </form.item>
      <form.item>
        <input />
      </form.item>
      <form.item>
        <button />
      </form.item>
    </form>
    <div>
      <p>text</p>
      <link />
    </div>
  </div>
</div>
//layout komponen navbar
<div>
  <menu />//menu berisi link navigasi
  <div>
    <button>
      <link />
    </button>
  </div>
</div>
```

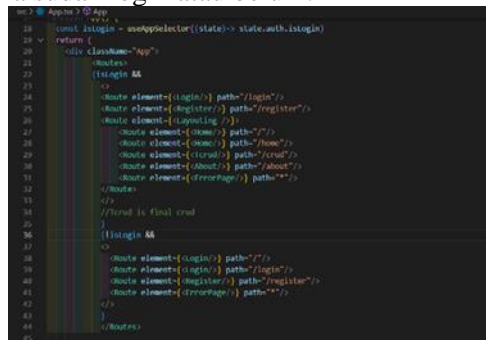
Gambar 17. proses layouting

Dari layout diatas dapat dilihat bahwa ada beberapa html tags yang berbeda, tags yang lain itu merupakan tags komponen yang diambil dari ant design hal ini digunakan untuk mempermudah proses layouting dan styling.

Untuk styling sendiri ada 2 metode yang digunakan saat kerja praktik, yaitu dengan menggunakan global css dan styled component dimana global css adalah menambah styling di file css utama seperti app.css dan styled component dimana kita melakukan styling langsung didalam komponen, hal ini digunakan apabila kita membutuhkan styling yang lebih spesifik yang tidak digunakan di komponen lainnya biasanya dilakukan dengan menambah atribut styles di dalam komponen atau bisa juga dipisah dengan menggunakan function. Styling juga dapat dibantu dengan styling bawaan dari ant design dengan mengimpornya ke global css apabila kita tidak ingin berlama-lama mengutak-atik dengan styling.

3.1.3. Routing

Untuk selanjutnya adalah proses routing yang dibantu menggunakan react router library. Proses routing bertujuan untuk menavigasi halaman react ke halaman lainnya dengan rute yang dapat dikonfigurasi sesuai alur yang akan dibuat. Proses ini juga berfungsi agar url dapat berubah-ubah tapi tetap menjaga agar ui tetap bersinkronasi dengan url. Routing juga dapat digunakan untuk dikustomisasi lebih lanjut seperti menambah kondisi untuk siapa atau apa saja yang dapat mengakses route ini dan apabila tidak akan diarahkan kemana, pada dasarnya routing membuat berbagai skenario agar pengguna diarahkan sesuai dengan akses yang diberikan. Dibawah ini adalah contoh dari routing yang saya buat dengan dua alur menggunakan state islogin untuk memeriksa apakah pengguna sudah login atau belum.



```

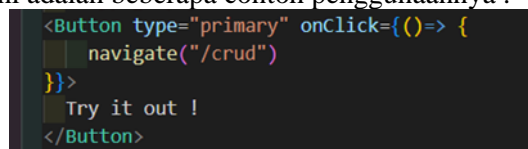
18 const islogin = useSelector((state) => state.auth.islogin)
19
20 return (
21   <div className="App">
22     <Routes>
23       <RouteElement path="/login"/>
24       <RouteElement path="/register"/>
25       <RouteElement path="/home"/>
26       <RouteElement path="/crud"/>
27       <RouteElement path="**"/>
28     </Routes>
29   </div>
30 )
31
32 // Try it out !
33
34 </App>
35
36 </Routes>
37
38 <RouteElement path="/login"/>
39 <RouteElement path="/register"/>
40 <RouteElement path="**"/>
41 </RouteElement>
42 </RouteElement>
43 </RouteElement>
44 </RouteElement>
45 </RouteElement>
46 </RouteElement>
47 </RouteElement>
48 </RouteElement>
49 </RouteElement>
50 </RouteElement>
51 </RouteElement>
52 </RouteElement>
53 </RouteElement>
54 </RouteElement>
55 </RouteElement>
56 </RouteElement>
57 </RouteElement>
58 </RouteElement>
59 </RouteElement>
60 </RouteElement>
61 </RouteElement>
62 </RouteElement>
63 </RouteElement>
64 </RouteElement>
65 </RouteElement>
66 </RouteElement>
67 </RouteElement>
68 </RouteElement>
69 </RouteElement>
70 </RouteElement>
71 </RouteElement>
72 </RouteElement>
73 </RouteElement>
74 </RouteElement>
75 </RouteElement>
76 </RouteElement>
77 </RouteElement>
78 </RouteElement>
79 </RouteElement>
80 </RouteElement>
81 </RouteElement>
82 </RouteElement>
83 </RouteElement>
84 </RouteElement>
85 </RouteElement>
86 </RouteElement>
87 </RouteElement>
88 </RouteElement>
89 </RouteElement>
90 </RouteElement>
91 </RouteElement>
92 </RouteElement>
93 </RouteElement>
94 </RouteElement>
95 </RouteElement>
96 </RouteElement>
97 </RouteElement>
98 </RouteElement>
99 </RouteElement>
100 </RouteElement>

```

Gambar 18. contoh routing yang dibuat

3.1.4. Menambah Fungsionalitas

seperti yang sudah dijelaskan sebelumnya dalam kerja praktik ini kita menggunakan beberapa library untuk menggunakan komponen ui yang siap pakai yang disini menggunakan library ant design. Dengan menggunakan ant design kita tidak perlu membuat komponen dari awal tidak hanya styling-nya saja tetapi secara fungsionalitas juga. Contohnya yang digunakan disini adalah untuk membuat komponen formulir di halaman login, komponen navbar (yang dikustomisasi lagi agar dapat menambahkan komponen lainnya) dan komponen button untuk membuat tombol di setiap halaman yang membutuhkannya. Biasanya untuk komponen button ant design digunakan untuk mengarahkan ke halaman lainnya, untuk itu kita menggunakan komponen dari react router berupa hooks usenavigate (yang dinamakan ulang navigate) yang dipanggil didalam onclick. sementara untuk komponen formulir ant design menyediakan 2 kondisi berupa fungsi untuk meng-handle data yang dimasukkan yaitu onFinish dan onFinishfailed, yang sesuai namanya onFinish untuk apabila data yang sukses dimasukkan dan onFinishfailed apabila data gagal dimasukkan, biasanya untuk fungsi ini diisi dengan fungsi async/await untuk fetching data. Dibawah ini adalah beberapa contoh penggunaannya :



```

<Button type="primary" onClick={() => {
  navigate("/crud")
}}>
  Try it out !
</Button>

```

Gambar 19. contoh komponen button untuk navigasi

```

const onFinish = () => {
  form.validateFields()
    .then((values:any) => {
      dispatch(postLogin(values));
    })
    .catch((error) => error.message);
};
const onFinishFailed = (errorInfo : any) => {
  console.log('Failed:', errorInfo);
  navigate('/login');
};

```

Gambar 20. contoh fungsi di halaman login

3.1.5. Manajemen State Dan Memanggil Api

dalam membuat website tentunya agar dapat berfungsi kita menggunakan data yang dapat berubah-ubah atau dinamik. Kelemahannya adalah apabila kita menggunakan state didalam komponen adalah kode menjadi sulit dibaca apalagi state yang digunakan cukup banyak dan satu lagi kelemahannya adalah apabila kita ingin state tersebut bersinkronasi dengan state di komponen lainnya. Maka dari itu kita perlu manajemen state tersebut untuk mengatasi kekurangan yang sudah disebutkan. Berikut adalah cara yang saya pakai untuk manajemen state saat mengerjakan magang menggunakan react redux dan redux persist, berikut adalah langkah-langkahnya:

1. Membuat slice

seperti yang sudah dijelaskan di dasar teori react redux memiliki 3 konsep yaitu action, reducer dan store, dan slice adalah menggabungkan action dan reducer dalam satu file agar kode lebih mudah dibaca. Seperti yang sudah dijelaskan yang pertama adalah kita membuat action terlebih dahulu, action berisi objek yang berisi informasi, contohnya adalah seperti gambar di bawah ini.

```

type User = {
  id?: number
  email: string
  password: string
}

type InitialState = {
  loading: boolean
  users: User[]
  error: boolean
  isLogin: boolean
  isReg: boolean
}

const initialState: InitialState = {
  loading: false,
  users: [],
  error: false,
  isLogin: false,
  isReg: false
}

```

Gambar 21. membuat redux action

dikarenakan kita menggunakan typescript kita juga perlu mendeskripsikan tipe data dari setiap state. Langkah selanjutnya adalah membuat reducer untuk mengeksekusi apa yang akan kita rubah dari initialstate atau state pertama yang digunakan sebagai perbandingan. Contohnya seperti gambar berikut:

```

const authSlice = createSlice({
  name: 'auth',
  initialState,
  reducers: {
    reset: (state) => initialState
  },
  extraReducers: (builder) => {
    builder.addCase(postLogin.pending, (state) => {
      state.loading = true;
      state.error = false;
    });
    builder.addCase(postLogin.fulfilled, (state, action: PayloadAction<any>) => {
      state.loading = false;
      state.error = false;
      state.isLogin = true;
      state.users = action.payload;
    });
    builder.addCase(postLogin.rejected, (state) => {
      state.loading = false;
      state.error = true;
      state.isLogin = false;
    });
  }
});

```

Gambar 22. membuat redux reducer

dalam react redux terdapat 2 jenis reducer yaitu reducers dan extrareducers. Reducers digunakan untuk mengubah state dari respon action yang sudah dibuat sementara extrareducers sebenarnya merupakan pengembangannya saja, tetapi lebih dirancang untuk menangani lebih banyak opsi, terutama action lain (seperti yang dihasilkan di slice lainnya atau tindakan yang dibuat oleh createAction atau createAsyncThunk). Apabila kita melihat gambar diatas reducers diisi oleh reset yang digunakan untuk mengembalikan state kembali seperti initialState dan extrareducers yang digunakan untuk menangani state dari createAsyncThunk yang akan dijelaskan di langkah selanjutnya.

2. Memanggil api dengan createAsyncThunk

CreateAsyncThunk adalah fungsi yang menerima action redux dan fungsi callback yang nantinya akan mengembalikan promise yang nantinya akan menghasilkan 3 redux action menggunakan createAction yaitu: pending , fulfilled , and rejected. Ini bertujuan untuk menangani state mana saja yang akan dirubah saat data yang diambil dalam kondisi tertunda,berhasil dan ditolak. Berikut adalah contohnya:

```

export const postLogin = createAsyncThunk(
  'auth/postLogin',
  async (users: User) => {
    return axios
      .post('https://reqres.in/api/login', {
        email: users.email,
        password: users.password
      })
      .then((response) => response.data)
      .catch((err) => {
        if (err.response.status === 400) {
          return Promise.reject(err);
        }
      });
  }
);

```

Gambar 23. memanggil api dengan creatasyncthunk

3. Menambahkan store dan hook di folder app

pada gambar 3.6 terdapat folder app, folder ini digunakan untuk men-setup redux store dan hooks karena kita menggunakan typescript. Untuk penggunaannya sama sesuai dengan yang dijelaskan di dokumentasi resmi dari react redux hanya saja dengan merubah reducer sesuai reducer yang kita gunakan.

4. Menambahkan redux persist

untuk mencegah state kembali ke initialState kita membutuhkan bantuan redux persist yang didefinisikan di redux store. Berikut adalah contohnya:

```

const persistConfig = {
  key: 'root',
  storage,
}

const reducers = combineReducers({
  auth: persistReducer<any>(persistConfig, authSlice),
});

export const store = configureStore({
  reducer: reducers,
  middleware: (getDefaultMiddleware) =>
    getDefaultMiddleware({
      serializableCheck: false,
    })
})

export const persistor = persistStore(store)

```

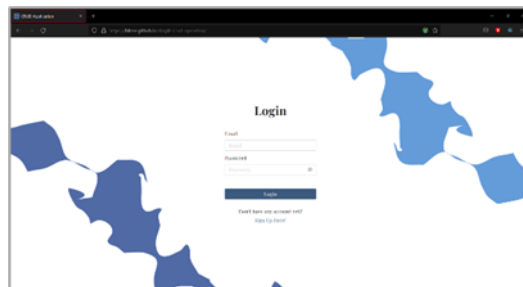
Gambar 24. mengkonfigurasi redux persist

dalam prototipe ini redux digunakan untuk menjaga agar state islogin tetap berada dalam kondisi true saat login dan nantinya disimpan ke localStorage agar halaman masih bisa diakses meskipun halaman dimuat ulang dan baru akan bernilai false apabila pengguna melakukan logout.

3.1.6. Deploying

Apabila semua komponen sudah bisa diasatukan dan berfungsi sebagaimana mestinya tahap terakhir yang dilakukan adalah men-deploy aplikasi react yang sudah kita buat.

Untuk proses ini ada banyak pilihan yang bisa digunakan untuk melakukan deploying, salah satu yang bisa digunakan adalah menggunakan github pages. Github pages memungkinkan kita untuk men-deploy dari kode yang kita unggah di repositori github. Untuk prosesnya seperti biasa dengan menginstall paket github pages menggunakan npm. Lalu lakukan perubahan pada file package.json sesuai instruksi di situs dokumentasi mereka, setelah itu add dan commit kode ke repositori github. Pastikan semua error sudah selesai dibereskan agar proses dapat dilanjutkan. Terakhir adalah men-deploy dengan memasukkan npm run deploy di terminal. Tunggu beberapa menit sampai website muncul, apabila berhasil maka website akan muncul seperti contoh gambar dibawah ini.



Gambar 25. hasil akhir website saat di-deploy

4. KESIMPULAN

Dalam melaksanakan tugas terdapat alur yang sudah ditetapkan dan didukung dengan alat penunjang seperti notion sebagai alat untuk mengorganisir pekerjaan, telegram sebagai alat berkomunikasi dan clockify sebagai alat pelacak waktu kerja. Prototipe website tersusun menggunakan bahasa pemrograman typescript dan bahasa css untuk styling. Ant design library digunakan untuk mempermudah proses styling dengan menggunakan komponen yang siap digunakan dan mudah diintegrasikan dengan sistem. Manajemen state dibuat dengan

menggunakan react redux untuk memudahkan mengorganisir state menggunakan slice yang berisi action dan reducer. Redux persist digunakan untuk mencegah state kembali ke nilai awal dan menyimpannya ke persisted storage. Untuk mengetes keberhasilan desain yang sudah dibuat dapat terkoneksi dengan server desain di tes menggunakan fake api. Hasil akhir dari prototipe di-deploy menggunakan github pages untuk di tes sekali lagi apakah website sudah siap apabila dimasukkan ke dalam real time environment. Dalam proses pembuatan protipe semua langkah dan bug akan dicatat dan nantinya akan didokumentasikan dan diarsip agar dapat mengecek kode yang sudah dibuat dan digunakan sebagai referensi untuk ke depannya. Komunikasi merupakan hal penting yang sangat ditekankan di kodingworks hal ini diterapkan untuk mencegah beredarnya asumsi yang tidak jelas dan mencegah miskomunikasi.

DAFTAR PUSTAKA

- [1] santi, rina candra noor. dan fitriyah, anida. “perancangan interaksi pengguna (user interaction design) menggunakan metode prototyping.”.jurnal teknik informatika 09, no. 2 (2016): 110.
- [2] gandee, todd, dan aquino, chris. front-end web development: the big nerd ranch guide (big nerd ranch guides) 1st edition .indianapolis: pearson education, 2016.
- [3] elverum, christer w. et al. “prototyping in new product development: strategy considerations.”. procedia cirp 50 (2016): 118-119.
- [4] reddy, mukthapuram p., mishra, sujata priyambada. “analysis of component libraries for react js.”. international advanced research journal in science, engineering and technology 08 no. 6 (2021): 43.
- [5] heonsik joo, “a study on understanding of ui and ux, and understanding of design according to user interface change” in.international journal of applied engineering research issn 0973-4562 volume 12, number 20 (2017) : 9931-9935