

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Dalam kegiatan magang ini, penulis berkesempatan untuk bergabung ke dalam tim *Fullstack Developer* dibawah naungan divisi *Technology Engineering* sebagai *Front-end Engineer*. Tugas dari tim *Fullstack Developer* itu sendiri adalah untuk mengelola *website*, mengembangkan fitur pada *website* sesuai dengan permintaan pengguna, dan memperbaiki *bug* dan memelihara *website*.

Sebelum memulai magang, para pemegang akan dibagikan menjadi beberapa tim sesuai dengan kemampuan yang dimiliki dan diberikan tugas untuk diselesaikan pada saat magang berlangsung, penulis bergabung ke dalam tim 4 bersama dua *Front-end Engineer* lainnya, yaitu Faisal Abu Bakar Riza dan Yulius Setyawan. Pada saat magang berlangsung, akan dipantau dan dievaluasi oleh Samuel Pangeran Aletheia selaku CTO dari Torche Education melalui aplikasi Discord.

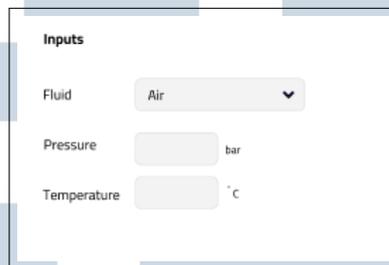
Dalam pengerjaan tugas pada kegiatan magang, tim akan berkomunikasi dengan menggunakan Discord dan rapat dua minggu (*Biweekly Meeting*) bersama dengan tim lainnya menggunakan Zoom. Pengerjaan tugas dilakukan dengan rapat awal untuk menentukan apa yang akan dikerjakan dalam kurun waktu 2 minggu, pada hari ke-8 akan dilakukan rapat tim untuk memberikan progres kepada rekan dan CTO jika ada hal yang bersangkutan. Dan pada minggu ke dua, akan dilakukan rapat 2 minggu atau *Biweekly Meeting* dengan tim lainnya untuk membagikan hal apa saja yang sudah dikerjakan, hal apa yang sedang dikerjakan, dan hal apa saja yang nantinya akan dikerjakan. Pengerjaan tugas yang dilakukan oleh tim 4 akan menggunakan *Browser-based IDE* bernama Replit dengan tujuan untuk pengembangan fitur secara kolaboratif.

3.2 Tugas yang Dilakukan

Tugas yang diterima oleh penulis saat pelaksanaan kerja magang adalah membuat ulang fitur kalkulator berbasis *Website* berjenis kalkulator ilmiah atau saintifik. Kalkulator ilmiah atau saintifik itu sendiri dirancang untuk menyelesaikan permasalahan yang membutuhkan perhitungan yang spesifik yang bisa dipakai di berbagai bidang ilmiah seperti matematika, fisika, teknik, dan lain - lain.

[9] Kalkulator ilmiah atau saintifik sendiri dibuat oleh Torche Education yang dimana mayoritas anggota dari bimbingan belajar tersebut berasal dari jurusan teknik kimia dan bioproses. Oleh sebab itu, Torche Education menyediakan fitur tersebut dengan tujuan memudahkan anggota untuk menyelesaikan permasalahan yang membutuhkan perhitungan tertentu.

Cara kerja dari kalkulator ilmiah atau saintifik yaitu pengguna akan memilih jenis kalkulator ilmiah atau saintifik mana yang akan digunakan dengan keperluan perhitungan tertentu, lalu pengguna akan memasukkan angka-angka ke dalam tabel input seperti contoh gambar berikut:



Inputs	
Fluid	Air ▼
Pressure	<input type="text"/> bar
Temperature	<input type="text"/> °C

Gambar 3.1. Contoh Input pada Kalkulator Ilmiah atau Saintifik

Lalu setelah memasukkan angka - angka pada tabel input, pengguna akan mengetahui hasil dari perhitungan tersebut pada tabel output seperti pada contoh gambar berikut :



Results	
Molecular Mass	12 kg/mol
Critical Pressure	bar
Molecular Mass	°C
Critical Temperature	bar
Triple Point Pressure	°C
Triple Point Temperature	
Accentric Factor	
Phase	
Compressibility Factor	
Density	kg/m ³
Specific Volume	°C
Specific Enthalphy	m ³ /kg
Specific Internal Energy	kJ/kg
Specific Entropy	kJ/kg
Specific Entropy	
Specific Entropy	

Gambar 3.2. Contoh Output pada Kalkulator Ilmiah dan Saintifik

Selain membuat kalkulator ilmiah atau saintifik berbasis *Website*, penulis dan tim juga mengerjakan desain ulang *Landing Page* dan halaman masuk kalkulator yang sudah dirancang oleh tim UI/UX.

3.3 Uraian Pelaksanaan Magang

Seperti yang sudah dijelaskan sebelumnya, pelaksanaan kerja magang dilakukan dengan diberikan tugas dan lama waktu pengerjaan adalah dua minggu, dan setelah minggu pertama, akan dilakukan evaluasi dan melanjutkan tugas yang sudah diberikan. Uraian pelaksanaan magang bisa dilihat pada Tabel 3.1.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Handover Meeting & Kickoff Meeting
2	Riset & Pencarian Referensi
3	<i>Scraping</i> Referensi dan <i>DIY Design</i>
4	<i>Biweekly Meeting</i> Pertama
5	Revisi <i>DIY Design</i> dan Konsultasi dengan Mentor
6	Menentukan desain yang sesuai dan menunggu tim UI/UX
7	<i>Biweekly Meeting</i> Kedua
8	Pengerjaan Desain sesuai Referensi
9	Pelanjutan Pengerjaan sesuai referensi
10	<i>Biweekly Meeting</i> Ketiga
11	Pengerjaan Landing Page
12	Pengerjaan Halaman Pilihan Calculator
14	<i>Biweekly Meeting</i> Keempat
15	Dokumentasi
16	Persiapan untuk <i>Handover Meeting</i>
17	<i>Handover Meeting</i>
20	Rapat Akhir dan Penutupan

3.3.1 Proses Pelaksanaan

Dalam proses pelaksanaan kerja magang, digunakan perangkat keras dan perangkat lunak untuk menunjang kegiatan kerja magang. Berikut adalah perangkat lunak yang digunakan dalam pengerjaan tugas magang :

- JavaScript (Bahasa Pemrograman)
- React (*JavaScript Framework*)
- TailwindCSS (*CSS Framework*)
- Replit (*Web - Based IDE*)
- Windows 11 Home (64 Bit) (Sistem Operasi)
- Google Chrome (*Browser*)

Adapun perangkat keras yang digunakan dalam kegiatan magang ini yaitu dengan menggunakan laptop Acer Swift 3 SF314-41 dengan spesifikasi berikut :

- Prosesor: AMD Ryzen 5 3500U
- Kartu Grafis: Radeon Vega Mobile Gfx
- Kartu Memori: 12 *Giga Bytes*
- Penyimpanan: 500 *Giga Bytes*

3.3.2 Referensi dan Pengembangan Desain

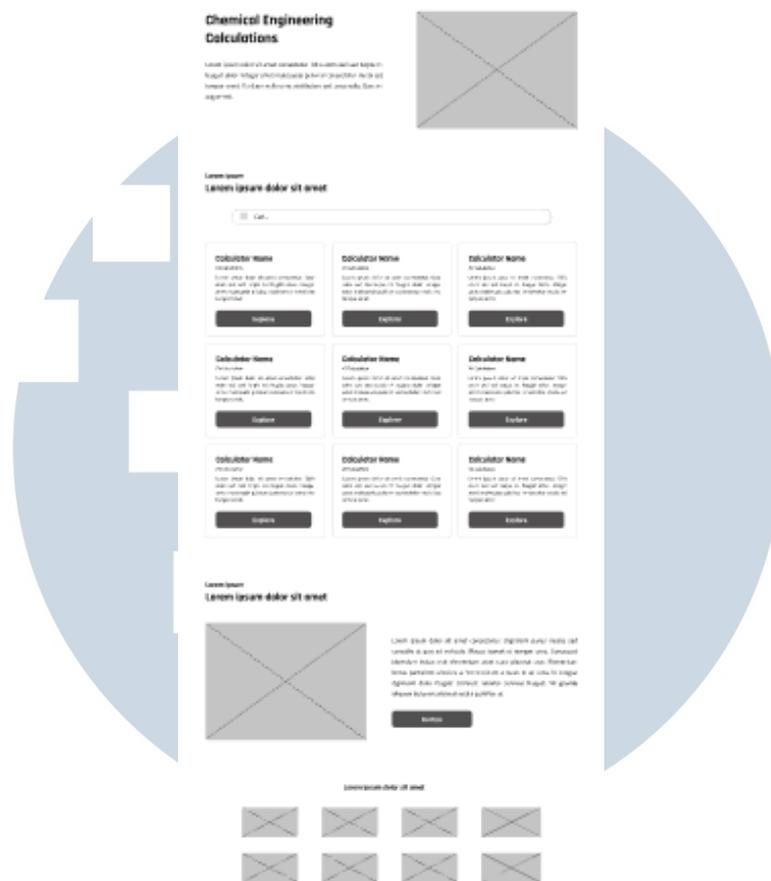
Pada pengembangan aplikasi kalkulator ilmiah atau saintifik berbasis *website* untuk *website* Torche Education, penulis dan tim diberikan referensi oleh tim UI/UX yang oleh berikut :

- omnicalculator.com
- codecalculation.com
- checalc.com

Pada *website* omnicalculator.com, terdapat 3 halaman yang akan dijadikan referensi oleh tim UI/UX dalam pengembangan kalkulator ilmiah atau saintifik, yaitu *Landing Page*, Pilihan Kalkulator, dan kalkulator. Dari referensi tersebut, tim UI/UX melakukan pembuatan *Low-Fidelity Prototype* dan *High-Fidelity Prototype*. Hasil *Low-Fidelity Prototype* dan *High-Fidelity Prototype* yang dilakukan oleh tim UI/UX adalah sebagai berikut :

A. *Landing Page*

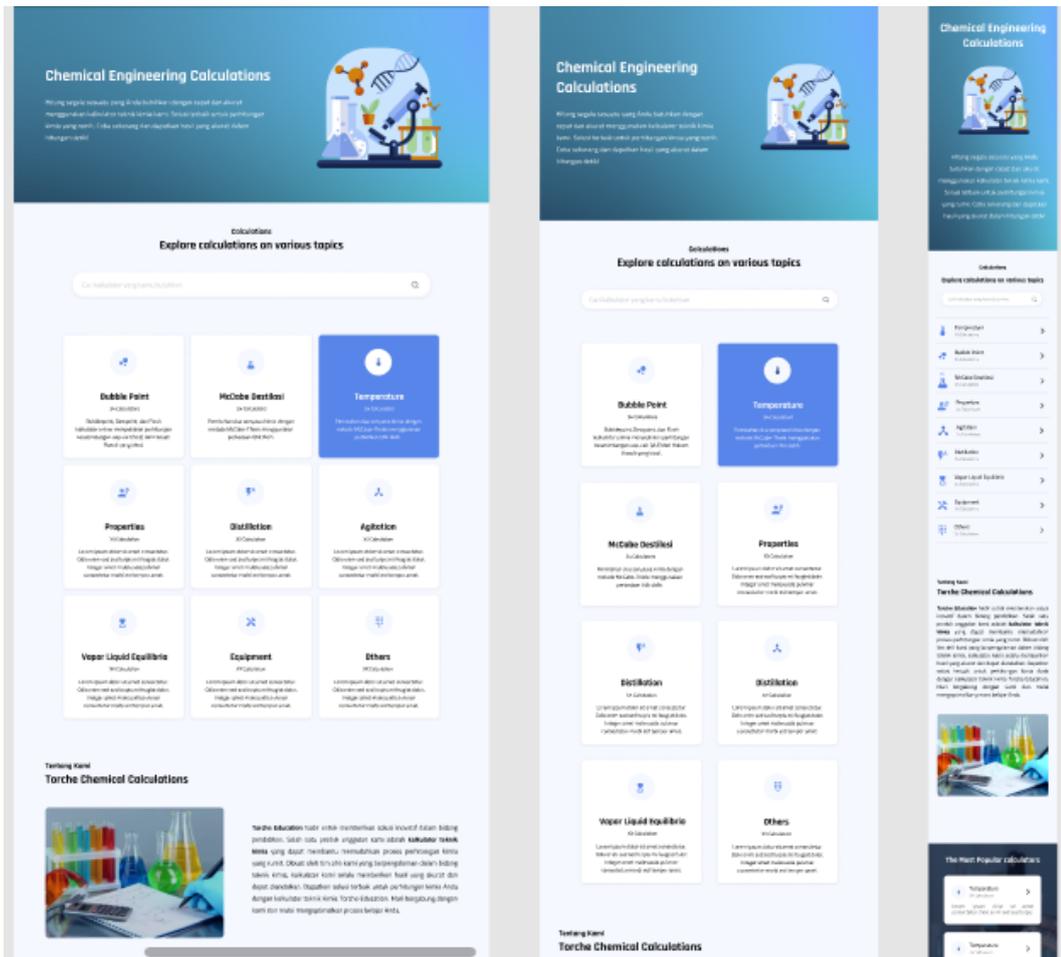
Pada umumnya, *Landing Page* adalah halaman utama di dalam *website* yang didesain khusus untuk tujuan tertentu. Dalam konteks proyek yang dikerjakan oleh penulis dan tim, *Landing Page* dibuat dengan tujuan untuk pengguna dalam memilih satu atau lebih kalkulator ilmiah atau saintifik yang akan digunakan dalam tujuan spesifik. Tim UI/UX telah mendesain *Landing Page* dalam bentuk *Low-Fidelity* adalah sebagai berikut:



Gambar 3.3. *Low-Fidelity Landing Page*

Berdasarkan hasil dari pengerjaan *Low-Fidelity Landing Page* yang dilakukan oleh tim UI/UX, terdapat 2 bagian penting dalam *Landing Page* tersebut, yaitu bagian *hero* dan bagian daftar kalkulator. Pada bagian *hero*, terdapat judul, deskripsi, dan grafis yang biasanya berformat PNG atau SVG. Pada bagian daftar kalkulator, daftar kalkulator yang akan tersedia dibentuk dalam bentuk kartu (*Card*) yang berisi judul, jumlah kalkulator, dan deskripsi dengan teks yang rata kiri. Kartu (*Card*) tersebut disusun dengan tampilan yang berbentuk *Grid*.

Setelah pengerjaan *Landing Page* dalam bentuk *Low-Fidelity*, selanjutnya akan dilakukan pengerjaan dalam bentuk *High-Fidelity*, disini akan berfokus kepada responsivitas yang ditujukan untuk memberikan kenyamanan pada pengguna untuk mengakses fitur tersebut di berbagai perangkat. Hasil dari pengerjaan tersebut adalah sebagai berikut :



Gambar 3.4. *High-Fidelity Landing Page* di Setiap Perangkat

Berdasarkan hasil pengerjaan tersebut, terdapat 3 jenis desain yang dibedakan berdasarkan jenis perangkat, pada desain di perangkat *desktop* di sebelah kiri, pada bagian *hero* masih sama persis dengan desain *Low-Fidelity* dengan memberikan grafis gambar yang berbentuk PNG atau SVG. pada bagian daftar kalkulator, *card* jenis kalkulator mengalami perubahan dengan memberikan sebuah grafis kecil diatas judul dan teks dari judul, jumlah kalkulator dan deskripsi diubah menjadi rata tengah.

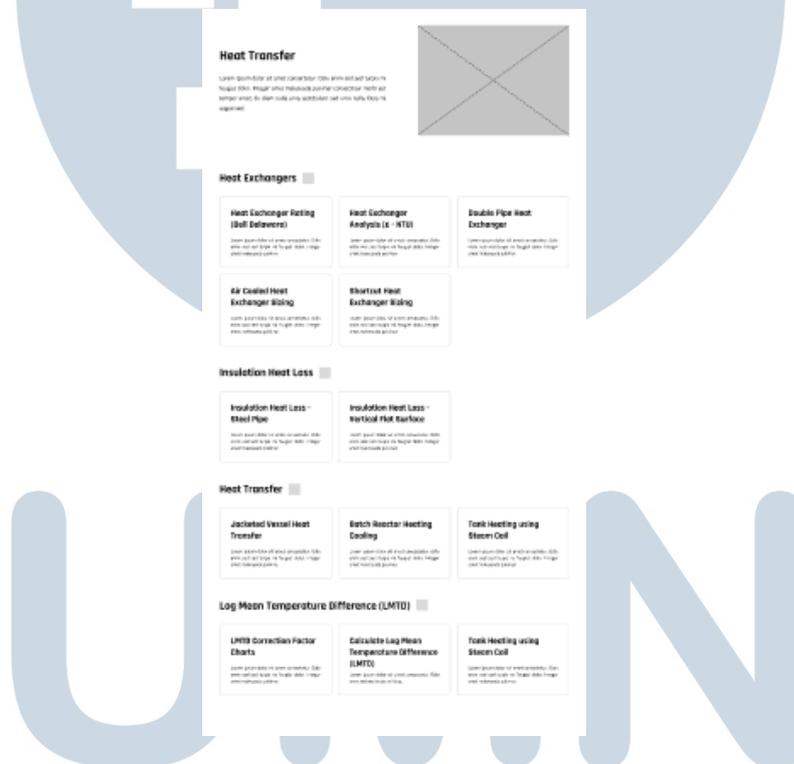
Pada desain di perangkat *tablet* sama seperti desain pada perangkat *desktop*, perbedaannya terdapat pada jumlah kolom dari *grid* dari sekumpulan *card* kalkulator dari 3 kolom menjadi 2 kolom.

Terakhir, pada desain di perangkat *smartphone*, terdapat perubahan yang cukup signifikan, dari bagian *hero* yang dimana grafis tersebut berada di tengah antara judul dan deskripsi, dan bagian daftar kalkulator tersebut yang awalnya berbentuk kartu (*card*) yang disusun dengan tampilan *grid* berubah menjadi bentuk

list dengan mempertahankan elemen dasarnya seperti judul, jumlah kalkulator, deskripsi, dan grafis.

B. Halaman Pilihan Kalkulator

Halaman pilihan kalkulator adalah lanjutan dari *Landing Page*. Setelah pengguna memilih salah satu jenis kalkulator yang telah dipilih pada halaman tersebut, akan menunjukkan beberapa sub jenis dari jenis kalkulator yang dipilih. Tim UI/UX telah mendesain halaman pilihan kalkulator dalam bentuk *Low-Fidelity* adalah sebagai berikut:

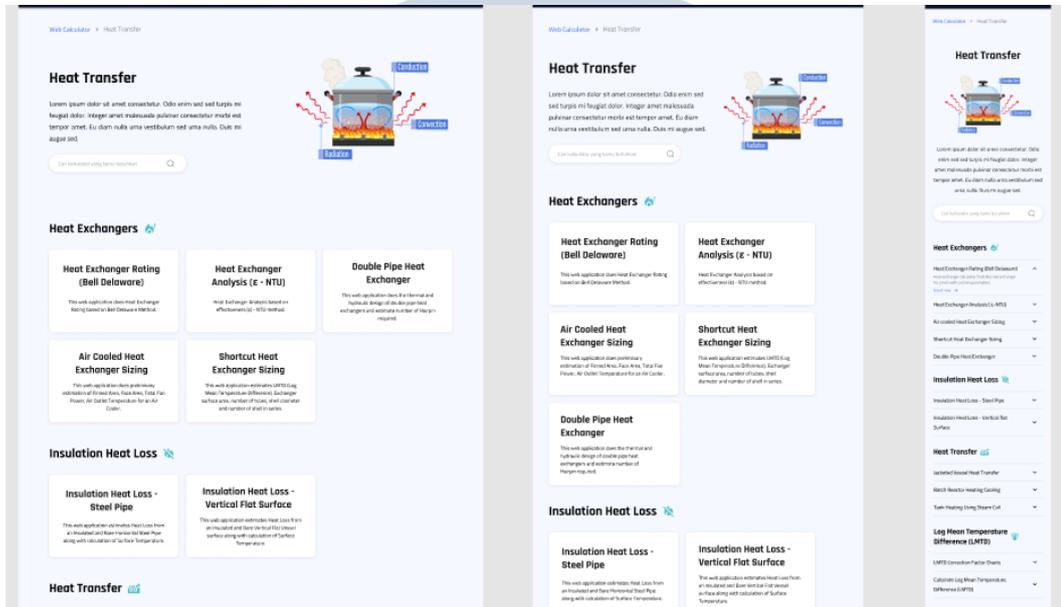


Gambar 3.5. *Low-Fidelity* Halaman Pilihan Kalkulator

Dari hasil pengerjaan tersebut, desain *Low-Fidelity* halaman pilihan kalkulator sama seperti dengan desain untuk *Landing Page*, hanya berbeda pada untuk desain pada daftar kalkulator yang dibedakan berdasarkan sub jenis dari kalkulator tersebut. Contoh seperti gambar 3.5 yang mengambil salah satu jenis kalkulator yang dipilih oleh pengguna yaitu *Heat Transfer* atau perpindahan panas yang dibagi menjadi 4 sub jenis.

Setelah itu, tim UI/UX mendesain halaman pilihan kalkulator dengan bentuk *High-Fidelity* dengan cara yang sama seperti sebelumnya, yaitu berfokus kepada

responsivitas berdasarkan perangkat, hasil dari pengerjaan tersebut adalah sebagai berikut :



Gambar 3.6. *High-Fidelity* Halaman Pilihan Kalkulator Setiap Perangkat

Dari hasil desain tersebut umumnya sama seperti pada desain pada *Landing Page* seperti *Card* yang disusun dalam tampilan *Grid* pada *desktop* dan *tablet* dan bentuk *List* pada *smartphone*, namun *Card* pada kalkulator tersebut dibuat berdasarkan sub jenis yang ada.

C. Halaman Kalkulator

Halaman kalkulator adalah halaman inti dari fitur yang dikembangkan pada proyek tersebut, halaman ini akan digunakan oleh pengguna untuk melakukan perhitungan yang dibutuhkan oleh pengguna. Berdasarkan salah satu referensi dari pengerjaan tersebut yaitu *omnicalculator.com*, ada 2 bagian penting dari halaman tersebut, yaitu kalkulator yang dibentuk dengan posisi *Sticky* yaitu salah satu elemen posisi di *CSS* yang memadukan antara nilai posisi *relative* dan *fixed*, yang bergantung kepada *scroll position*. [10] Elemen *sticky* akan bernilai *relative* hingga melewati tampilan halaman *browser* akan menempel. dan deskripsi yang berisi dari materi yang sesuai dengan kalkulator yang digunakan.

Dari referensi tersebut, tim UI/UX mendesain halaman kalkulator dalam bentuk *Low-Fidelity* adalah sebagai berikut :



Penjelasan Materi

Dasar-dasar Distilasi Biner

Distilasi adalah metode pemisahan dua komponen kimia dengan memanfaatkan perbedaan titik didih. Tujuan dari perhitungan ini adalah untuk mendapatkan tahapan teoritis dengan menentukan fraksi mol produk atas dan produk bawah. Asumsi yang dimodelkan dalam metode ini adalah sebagai berikut:

- Molar penguapan komponen umpan adalah sama, yang berarti tidak ada reaksi yang terjadi,
- Kesetimbangan uap-cair molar, yang berarti bahwa untuk setiap mol cairan yang diuapkan, satu mol uap terkondensasi, dan
- Hukum gas ideal diterapkan, yang berarti bahwa efek panas seperti panas larutan dapat diabaikan.

Distilasi biner menggunakan konsep Hukum Raoult yang dapat dipelajari lebih lanjut [di sini](#).

Diagram McCabe-Thiele

Ada beberapa komponen dalam Diagram McCabe-Thiele:

Diagram x-y

Diagram x-y menggambarkan data kesetimbangan uap-cair dari komponen-komponen biner, dimana setiap titik pada grafik menunjukkan jumlah cairan yang seimbang dengan uap pada suhu tertentu. Diagram akan membantu menentukan volatilitas relatif (α) untuk memprediksi jumlah pemisahan yang mungkin. Saat proses pemisahan akan membutuhkan lebih banyak bakli karena komponen sulit dipisahkan menggunakan metode ini. Volatilitas relatif adalah sebagai berikut:

$$\alpha = \frac{y_1/x_1}{y_2/x_2}$$

Dimana adalah fraksi uap, adalah fraksi cair, dan adalah komponen kimianya. Dalam sistem biner, persamaan di atas dapat dinyatakan sebagai:

$$\alpha = \frac{y_1/x_1}{(1-y_1)/(1-x_1)}$$

References

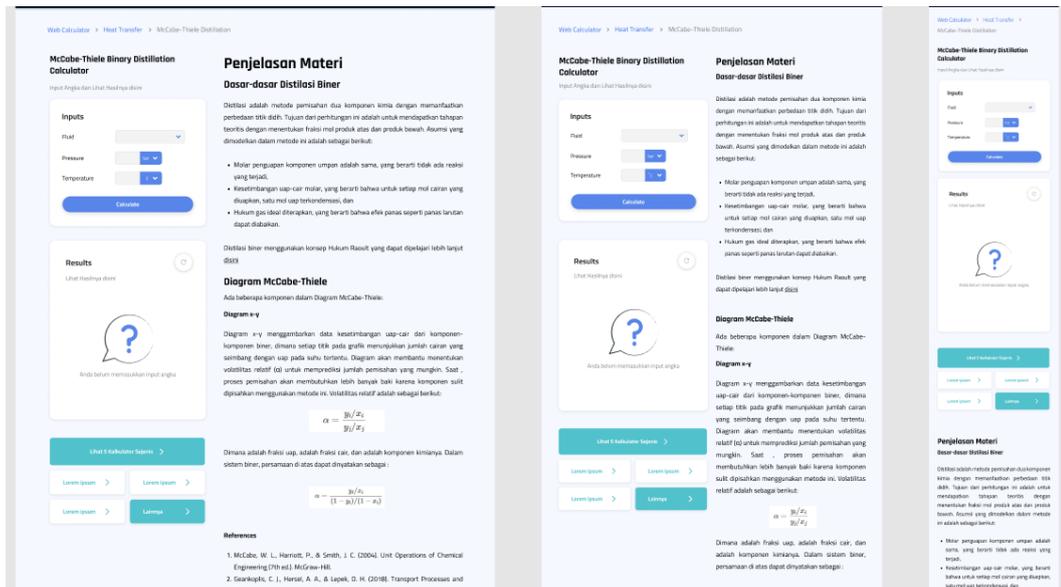
1. McCabe, W. L., Harriott, P., & Smith, J. C. (2004). Unit Operations of Chemical Engineering (7th ed.). McGraw-Hill.
2. Geankoplis, C. J., Hersel, A. A., & Lepek, D. H. (2018). Transport Processes and Separation Process Principles (5th ed.). Prentice Hall.

Gambar 3.7. *Low-Fidelity* Halaman Kalkulator

Dari desain tersebut, pada bagian kalkulator terdapat 2 bagian, yaitu bagian *input* yang berguna untuk memasukkan angka atau kategori yang pengguna inginkan dan bagian *output* yang berguna untuk melihat hasil dari perhitungan yang sebelumnya sudah diberikan angka masukan oleh pengguna.

Selain itu, pada bagian *input* pengguna bisa mengubah satuan dengan menggunakan tombol *Units*, setelah menekan tombol tersebut akan memunculkan *Modal* yang akan muncul di tengah dari halaman kalkulator yang berisikan satuan-satuan dari input, dan juga pengguna bisa membagikan hasil perhitungan dengan menekan tombol *Share* dan sama seperti saat menekan tombol *Units*, akan memunculkan *Modal* untuk memberikan opsi cara untuk membagikan hasil dari perhitungan.

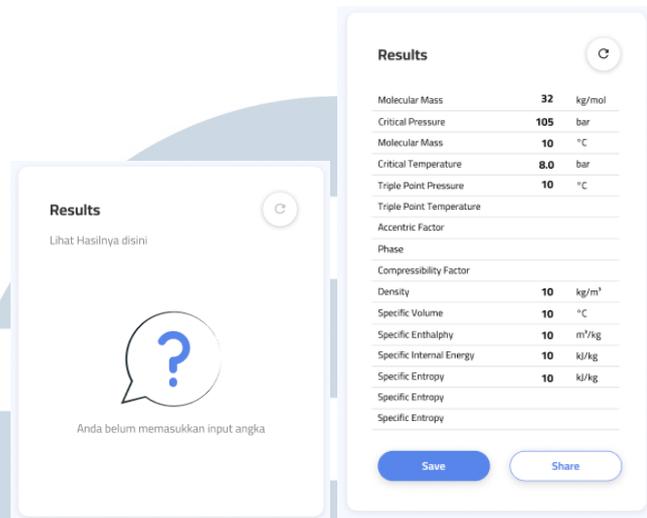
Dari pengerjaan halaman kalkulator, selanjutnya tim UI/UX mendesain halaman kalkulator dalam bentuk *High-Fidelity* adalah sebagai berikut :



Gambar 3.8. *High-Fidelity* Halaman Kalkulator Setiap Perangkat

Pada desain halaman kalkulator dalam bentuk *High-Fidelity*, hal yang bisa dibedakan terdapat kepada perubahan untuk mengubah satuan yang awalnya akan menggunakan *Modal* diubah dengan menggunakan *select option* yang diletakkan di sebelah di setiap *form input*, dan tampilan untuk *smartphone* yang dimana tampilan untuk dua elemen penting berubah dari kiri kanan menjadi atas bawah. Dan pada bagian output terdapat grafis yang menandakan bahwa belum ada nya interaksi pada tombol *Calculate*, jika ditekan, akan berubah seperti pada desain *Low-Fidelity*. Contohnya adalah sebagai berikut :





(a) Sebelum

(b) Sesudah

Gambar 3.9. Perubahan Pada Bagian Output

D. Warna dan *Font*

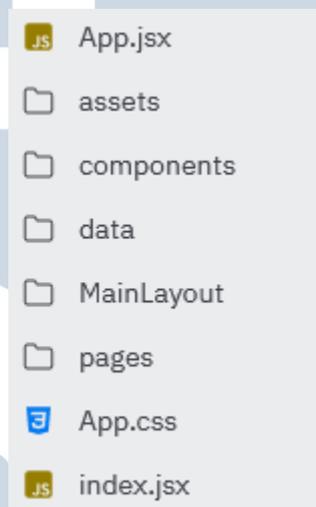
Warna dan *font* adalah salah satu elemen penting dalam pengembangan desain tersebut, warna memberikan kesan estetik pada desain *web* yang sedang dikembangkan dan *font* memberikan karakter pada *website* itu sendiri. Torche Education sendiri memiliki panduan mengenai warna dan *font* yang digunakan dalam pengembangan *website*. Warna yang digunakan adalah sebagai berikut :

- Natural Navy (#0B122A)
- Absorbing Blue (#5885E9)
- Molecular White (#D8D8D8)
- Molecular Purple (#9284F1)
- Molecular Orange (#FF6442)
- Molecular Tosca (#51C1CB)
- Molecular Yellow (#FFAE5F)

Dan *font* yang digunakan berdasarkan panduan dari Torche Education adalah Rajdhani dengan ketebalan 700 untuk judul dan 500 untuk sub judul, dan Cairo dengan ketebalan 400 untuk paragraf.

3.3.3 Struktur *File* Proyek

Setelah tim UI/UX selesai mendesain halaman fitur tersebut, tim penulis bersiap untuk mengkonfigurasi struktur *file*, sebelumnya penulis melakukan instalasi untuk mengunduh file - file React Js dengan bantuan alat *Frontend* yang banyak digunakan yaitu Node.js dan juga NPM (*Node Package Manager*) dengan menjalankan *command* `npx create-react-app project_name`. Setelah itu, terdapat dua *folder* utama dalam kerangka kerja React Js, yaitu *public* dan *src*. Yang difokuskan terdapat pada *folder src* yang dimana penulis dan tim akan melakukan pengerjaan untuk fitur tersebut. Isi dari *folder* tersebut adalah sebagai berikut:



Gambar 3.10. Struktur *Folder* Pada Proyek

Berdasarkan gambar 3.10, terdapat 5 *folder*, yaitu *assets*, *components*, *data*, *MainLayout*, dan *pages*. *Folder assets* berguna untuk menyimpan beberapa aset, contohnya menyimpan gambar dan beberapa ikon yang berguna untuk sebagai simbol, biasanya gambar dan ikon berformat PNG atau SVG. Pada *folder components* berisi beberapa komponen-komponen yang dibuat sesuai dengan desain yang dibuat oleh tim UI/UX. Pada *folder data*, berisi data-data dalam bentuk *array*. Pada *folder MainLayout*, berisi file dalam bentuk JSX yang dimana pada *file* tersebut sudah dikonfigurasi terdapat *header* dan *footer* yang akan digunakan pada setiap halaman yang akan dikerjakan, potongan kode untuk *file* tersebut terdapat pada gambar 3.11.

A screenshot of a code editor window with a dark background and a white text area. The code is written in a light-colored font. It shows the following code:

```
import React from 'react'

import Header from '../components/Header'
import Footer from '../components/Footer'

const MainLayout = (props) => {
  return (
    <Header />
    {props.children}
    <Footer />
  )
}
```

Gambar 3.11. Potongan Kode *MainLayout*

Pada potongan kode diatas, terdapat dua komponen yang telah disebut sebelumnya, yaitu *header* dan *footer* yang sudah diimpor ke dalam *file MainLayout*, setelahnya, diberikan *const function* yang diberikan parameter *props* yang nantinya parameter *props* itu diberikan variabel *children* yang akan berisi halaman yang sedang dikerjakan. Terakhir pada *folder pages* berisi beberapa *file* halaman - halaman yang akan nanti dikerjakan.

3.3.4 Implementasi Desain

Setelah tim UI/UX selesai melakukan desain baik dalam bentuk *Low-Fidelity* dan *High-Fidelity*, selanjutnya tim penulis mulai mengerjakan pengembangan *website* untuk fitur kalkulator ilmiah berbasis *website*. Cara pengerjaan fitur tersebut diawali dengan teknik *slicing*.

Pada umumnya, *slicing* adalah sebuah teknik pemotongan gambar/desain pada *website*, memecahnya menjadi beberapa bagian untuk kemudian disatukan kembali menggunakan kode CSS dan HTML. Hal ini bertujuan memudahkan pengerjaan halaman *website* sesuai dengan desain yang dirancang. Selain itu, *slicing* juga berguna jika ingin mengunduh halaman melalui *browser* yang dipakai, akan memunculkan *folder* yang berisikan potongan beberapa bagian dari halaman tersebut.

Berdasarkan penjelasan tersebut, tugas penulis dan tim adalah melakukan *slicing* terhadap desain yang telah dirancang oleh tim UI/UX. Pengerjaan *slicing* tersebut akan menggunakan salah satu kerangka kerja JavaScript yang

sudah direncanakan sebelumnya, yaitu React dengan kerangka kerja CSS yaitu TailwindCSS. Proses *slicing* akan menggunakan IDE berbasis *website* bernama Replit. Mulai pengerjaannya dilakukan pada *Landing Page* terlebih dahulu dan diakhiri di halaman kalkulator itu sendiri. Pengerjaan tersebut akan dilakukan dengan pengerjaan untuk desain pada perangkat *smartphone* terlebih dahulu.

Pada *Landing Page*, terdapat 4 bagian yang akan dikerjakan, yaitu *hero*, daftar kalkulator, kalkulator populer, dan tutorial. Pada laporan ini akan berfokus kepada bagian *hero* dan daftar kalkulator. Sesuai dengan desain yang diberikan oleh tim UI/UX, pada bagian *hero* terdapat judul, deskripsi, dan grafis. Potongan kode pada bagian hero adalah sebagai berikut:



```
<section className="bg-cover bg-gradient w-full p-12 h-fit flex flex-col items-center justify-center md:flex-row">
  <div className="w-3/4 h-fit mt-16 mb-8 lg:mx-8">
    <h1 className="font-headline font-bold text-4xl text-[#ffffff] text-center md:text-left mb-8">
      Title
    </h1>
    <p className="h-fit leading-9 font-bodycopy font-normal text-xl text-[#ffffff] hidden md:inline-block">
      Description
    </p>
  </div>

  <div className="w-64 aspect-square md:w-80 md:py-24 lg:w-[560px] lg:py-18">
    <img src={Hero} className="" />
  </div>

  <p className="w-[320px] px-4 mt-2 mb-16 font-bodycopy font-normal text-lg text-[#ffffff] text-center md:hidden">
    Alt Desc
  </p>
</section>
```

Gambar 3.12. Potongan Kode Bagian *Hero*

Berdasarkan gambar potongan kode diatas, penulis dan tim menggunakan tag HTML *section* yang merepresentasikan sebuah bagian dokumen atau aplikasi dengan tujuan untuk mengelompokkan konten/dokumen menjadi beberapa bagian berdasarkan konteks yang sedang dikerjakan. Selanjutnya, pada tag *section* diberikan *utility class* CSS yang disediakan oleh TailwindCSS yang dengan tipe tampilan *flex* dengan posisi vertikal. posisi *flex* akan berubah ke posisi horizontal jika pengguna menggunakan perangkat *tablet* dan *desktop*. Selanjutnya, untuk bagian judul dan deskripsi, akan dibungkus oleh tag *div* yang lebarnya tiga per empat dari luas tag *section* dan di dalamnya terdapat tag *h1* yaitu header tipe satu dan tag *p* untuk deskripsi. Untuk deskripsi yang di dalam tag *div*, tag tersebut akan bersifat *hidden* atau hilang di perangkat *smartphone*. Sebagai gantinya untuk

deskripsi di *smartphone*, penulis menambahkan deskripsi alternatif di bawah tag `div`. Pada untuk grafis, rasio yang dipakai adalah 1:1 yang ditandai dengan *utility class* yaitu `aspect-square` dengan ukuran untuk *smartphone* yaitu 16 rem atau 256 piksel dan 20 rem atau 320 piksel untuk *tablet* dan *desktop*.

Bagian berikut yang akan dibahas adalah bagian daftar kalkulator, di bagian disini terdapat beberapa kalkulator berdasarkan jenis kebutuhannya. Potongan kode dari bagian daftar kalkulator adalah sebagai berikut:

A screenshot of a code editor window with a dark background and a white text area. The code is written in a light-colored font and shows the structure of a calculator list. It includes a section with a pink background, a heading 'Calculations', a sub-heading 'Explore calculations on various topics', a search input field, and a grid of calculator cards. The code uses Tailwind CSS classes for styling and layout.

```
<section className="bg-[#F5F8FF] w-full px-[26px] md:px-[120px] lg:px-[130px] py-20">
  <h1 className="text-center text-base font-bold text-black mb-5 md:text-2xl lg:text-2xl">
    Calculations
  </h1>
  <h1 className="text-center font-bold text-black text-xl md:text-4xl lg:text-4xl">
    Explore calculations on various topics
  </h1>
  <SearchInput />
  <div className="w-full mt-[40px] mb-[70px] lg:grid lg:grid-cols-3 lg:gap-5 md:grid-cols-2 md:gap-5 md:grid">
    <hr className="border-[1px] border-solid border-[#D6D6D6] lg:hidden md:hidden" />
    <CalculatorCard />
  </div>
</section>
```

Gambar 3.13. Potongan Kode Daftar Kalkulator di *Landing Page*

Berdasarkan potongan kode di atas, terdapat komponen yang telah dibuat sebelumnya yang sudah di impor, salah satunya adalah komponen kartu atau *card* yang bernama `CalculatorCard`. komponen `CalculatorCard` dibuat sendiri dengan file yang terpisah, Pengerjaan komponen `CalculatorCard` tersebut akan mengerjakan 2 model daftar kalkulator yang sesuai dengan desain sebelumnya, potongan kode dari dua model komponen tersebut adalah sebagai berikut:

```

{ /* This is for mobile */ }
<div className="w-full flex flex-col items-left my-3 lg:hidden md:hidden">
  <div className=" flex flex-row justify-between items-center">
    <div className="w-5/12 flex flex-row">
      <img src={Icon} alt="icon" className="object-contain" />
      <div className="flex flex-col px-3">
        <h1 className="font-medium text-base">Temprature</h1>
        <p className="text-xs text-[#818181]">24 Calculators</p>
      </div>
    </div>
    <img
      src={Arrow}
      alt="Arrow"
      className={`transition-all ${show} && 'rotate-90'`}
      onClick={() => setShow(!show)}
    />
  </div>
  <div className={`pl-12 pt-2 ${show ? '' : 'hidden'}`>
    <p className="text-xs font-medium text-[#818181]">
      Pemisahan dua senyawa kimia dengan metode McCabe-Thiele menggunakan
      perbedaan titik didih.
    </p>
    <span className="text-sm font-semibold text-[#5885E9]">Explore Now</span>
  </div>
</div>

```

(a) Komponen CalculatorCard Untuk Mobile

```

{ /*This is for desktop and tablet */ }
<div className="group h-[375px] bg-white rounded-[10px] px-[34px] hidden md:flex lg:flex flex-col
items-center justify-center transition-all hover:bg-[#5885E9] hover:text-white md:shadow-md lg:shadow-
md">
  <div className="w-[82px] h-[82px] bg-[#F5F8FF] rounded-full flex justify-center items-center group-
hover:bg-white">
    <img src={Icon} alt="icon" className="object-contain" />
  </div>
  <div className="text-center ">
    <h1 className="font-bold text-3xl mt-[48px] mb-[8px]">Temprature</h1>
    <p className="font-medium text-base mb-[14px]">24 Calculators</p>
    <p className="text-base">
      Pemisahan dua senyawa kimia dengan metode McCabe-Thiele menggunakan
      perbedaan titik didih.
    </p>
  </div>
</div>

```

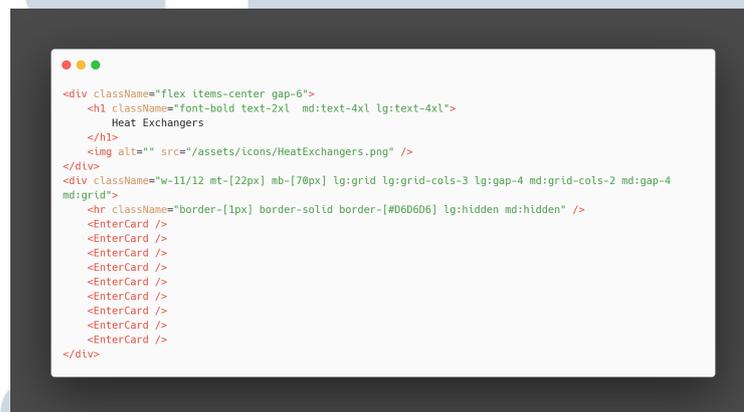
(b) Komponen CalculatorCard untuk Tablet dan Dekstop

Gambar 3.14. Potongan Kode Komponen CalculatorCard setiap Perangkat

Pada potongan kode diatas, pada gambar 3.14b terdapat *utility class* hidden yang digunakan untuk menyembunyikan konten tersebut, karena untuk potongan kode 3.14b akan muncul untuk perangkat *tablet* dan *desktop*. Begitupun sebaliknya,

pada potongan kode 3.14a, pada breakpoint md dan lg yaitu untuk ukuran *tablet* dan *desktop* diberikan *utility class* hidden juga. Hal ini dilakukan dengan tujuan jika terjadi perbedaan kelebaran pada perangkat yang digunakan, maka komponen CalculatorCard akan berubah bentuk.

Setelah untuk *Landing Page*, selanjutnya pengerjaan untuk halaman pilihan kalkulator. Pada halaman pilihan kalkulator kurang lebih sama seperti pengembangan *Landing Page*, yaitu berfokus kepada dua bagian penting, yaitu bagian *hero* dan daftar kalkulator, namun perbedaannya adalah di dalam halaman pilihan kalkulator terdapat beberapa kalkulator yang dibedakan berdasarkan sub jenis dari jenis yang dipilih oleh pengguna dan desain yang dipakai sesuai dengan desain yang sebelumnya telah dirancang oleh tim UI/UX. Potongan kode untuk bagian daftar kalkulator adalah sebagai berikut:



```
<div className="flex items-center gap-6">
  <h1 className="font-bold text-2xl md:text-4xl lg:text-4xl">
    Heat Exchangers
  </h1>
  
</div>
<div className="w-11/12 mt-[22px] mb-[70px] lg:grid lg:grid-cols-3 lg:gap-4 md:grid-cols-2 md:gap-4 md:grid">
  <hr className="border-[1px] border-solid border-[#606060] lg:hidden md:hidden" />
  <EnterCard />
</div>
```

Gambar 3.15. Potongan Kode Daftar Kalkulator Pada Halaman Pilihan Kalkulator

Potongan kode pada gambar 3.15 bisa diaplikasikan beberapa kali sesuai dengan kebutuhan sub jenis yang ada berdasarkan jenis kalkulator yang dipilih oleh pengguna. Komponen EnterCard hampir sama dengan komponen CalculatorCard yang dipakai di *Landing Page*, hanya membedakan pada tidak memakai grafis di atas judul kalkulator.

Terakhir, untuk halaman kalkulator, terdapat dua bagian penting, yaitu bagian input dan output, dan deskripsi yang berisikan materi dari kalkulator yang digunakan. Bagian yang akan difokuskan adalah pada bagian input dan output. Pada bagian input dan output akan dibungkus oleh tag div yang diberikan *utility class* sticky untuk perangkat *tablet* dan *desktop*. Potongan kode untuk implementasi sticky adalah sebagai berikut:

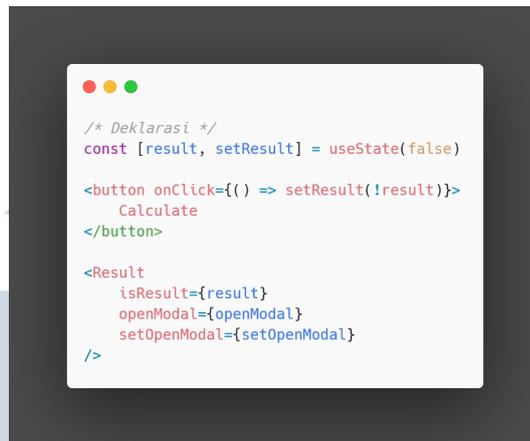
```
<div className="sticky top-20 z-40">
  <h1 className="font-bold text-xl md:text-2lg lg:text-3xl">Title</h1>
  <p className="font-normal text-xs text-[#668383] my-2 lg:my-3 md:text-lg lg:text-lg">
    Input Form
  </p>
  <div className="w-full bg-white rounded-2xl shadow-md px-[35px] py-9 mb-10">
    <h1 className="font-bold md:text-2xl lg:text-2xl">Inputs</h1>
    <form>FORM FIELD</form>
    <button
      className="bg-[#5885E9] w-full px-2.5 py-2 rounded-3xl text-white transition-color duration-500
      transform hover:bg-[#0B122A] hover:border-[#080808] text-xs md:text-base lg:text-base font-bold
      shadow-md"
      onClick={() => setResult(!result)}
    >
      Calculate
    </button>
    <Result
      isResult={result}
      openModal={openModal}
      setOpenModal={setOpenModal}
    />
  </div>
</div>
```

Gambar 3.16. Potongan Kode Implementasi Sticky

Pada potongan kode di gambar 3.16, setelah menggunakan *sticky*, untuk tag *div* yang diisi oleh *form* input dan output, akan diberikan *utility class* *top* untuk memberikan posisi absolut dan *z-index* untuk mengontrol urutan *stack* di dalam *div*. Di potongan kode yang sama, terdapat tombol *Calculate* yang dimana ada *event* *onClick* yang dimana jika pengguna menekan tombol tersebut akan menjalankan suatu fungsionalitas yang sudah dirancang sebelumnya, di dalam konteks dari pengembangan ini, tombol tersebut akan menjalankan fungsi untuk menampilkan komponen *Result*.

Di kerangka kerja JavaScript React, terdapat fitur yang bernama *React Hooks* yang digunakan untuk melakukan *state management* dan *side effects* di dalam proyek yang sedang dikembangkan. Salah satu *Hooks* yang digunakan pada pengembangan fitur ini adalah *useState*. *useState* digunakan untuk membuat *state* di dalam *function*. Contoh dari implementasi *useState* pada pengembangan fitur ini adalah sebagai berikut:

UNIVERSITAS
MULTIMEDIA
NUSANTARA

A screenshot of a code editor window with a dark background and a white text area. The code is written in a light blue font. It shows a JavaScript/React code snippet for implementing the useState hook. The code includes a comment, a const declaration for state and setter, a button click handler, and a Result component with props.

```
/* Deklarasi */
const [result, setResult] = useState(false)

<button onClick={() => setResult(!result)}>
  Calculate
</button>

<Result
  isResult={result}
  openModal={openModal}
  setOpenModal={setOpenModal}
/>
```

Gambar 3.17. Implementasi useState Untuk Komponen Result

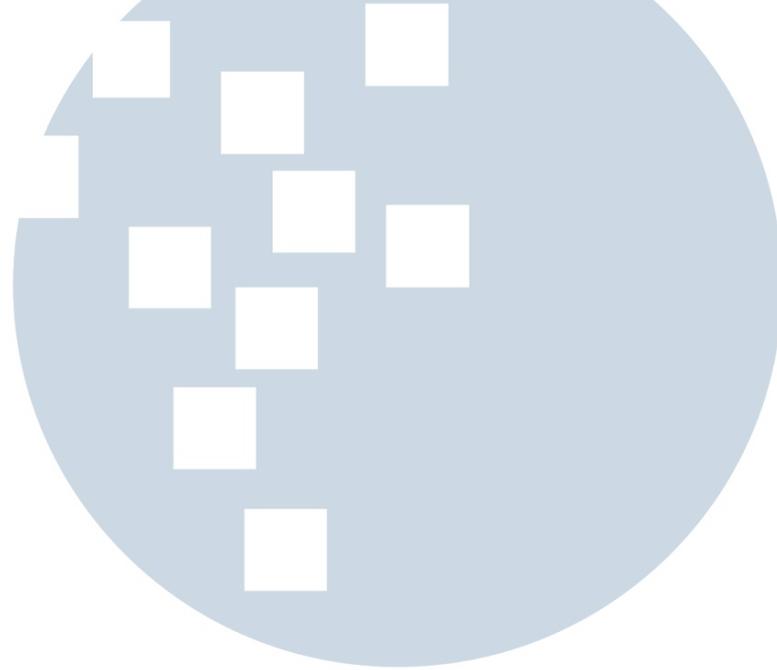
Berdasarkan potongan kode diatas, fungsi dari useState tersebut untuk memunculkan komponen Result. Di potongan kode diatas, penulis mendeklarasikan *const function*, dalam *const function* tersebut akan diberikan dua variabel didalamnya, variabel yang pertama adalah *state* yang berguna untuk memberikan nilai awal dari *const function*, nama *state* yang diberikan di kedua *const function* tersebut adalah *result* dengan nilai awal adalah boolean *false*. Variabel kedua adalah *setState* yang digunakan untuk mengubah nilai dari *state*. Pemberian namanya sesuai dengan nama dari variabel pertama ditambah dengan kata *set*.

Untuk implementasinya, di dalam tag *button*, pada *event* *onClick* akan diberikan nilai baru dengan bantuan *setState* *setResult* yang menghasilkan nilai baru untuk *state* *result* yaitu *!result* yang berarti nilai tersebut adalah boolean *true*. Dalam komponen *Result* juga diberikan *event* buatan, yaitu *isResult* yang diberikan nilai pada *state* *result*.

3.4 Kendala dan Solusi yang Ditemukan

Selama pengerjaan fitur kalkulator ilmiah atau saintifik berbasis *Website*, kendala yang ditemukan pada pengerjaan fitur tersebut adalah pada pengembangan desain oleh tim UI/UX yang sangat lama, kendala dari tim UI/UX tersebut adalah kurangnya waktu pengerjaan yang disebabkan oleh kendala diluar kegiatan magang dan akibatnya, untuk tim penulis mengalami kelambatan untuk mengerjakan implementasi fitur tersebut. Solusi dari kendala tersebut adalah dari tim UI/UX diberikan akses ke dalam Figma untuk dikerjakan langsung selama tim UI/UX merancang desain sampai tahap final. Selanjutnya, pada rencana sebelumnya, tim penulis juga diberikan tugas untuk membuat logika matematika untuk kalkulator,

namun pada salah satu referensi yang dijadikan acuan untuk pengembangan fitur tersebut, logika matematika yang merupakan bagian krusial dari fitur tersebut tidak disimpan di dalam *source*. Keputusan dari mentor adalah penulis dan tim cukup berfokus kepada pengerjaan *slicing* saja.



UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA