

Soal Logika

1. Buat fungsi dengan menampilkan bilangan cacah kelipatan 3 atau 7 sebanyak N, serta menampilkan huruf Z saat bilangan tersebut kelipatan 3 dan 7.

Contoh :

N = 13

Output : 3, 6, 7, 9, 12, 14, 15, 18,

Z

, 24, 27, 28, 30

Jawaban

```
//function bilangan cacah
const n = ()=>{
  for (let i = 3; i <= 30; i++) {
    if (i % 3 == 0 && i % 7 == 0) {
      console.info('Z');
    } else if (i % 3 == 0 && i % 7 != 0) {
      console.info(i);
    } else if (i % 3 != 0 && i % 7 == 0) {
      console.info(i);
    }
  }
}
n();

//output
3
6
7
9
12
14
15
18
Z
24
27
28
30

let n = 4
while(n < 10){
  n* = n/2
  return
}

a.64.0
b.32.0
c.12.0
d.none above
```

Referensi : keyword FizzBuzz

2. Buat fungsi pencarian 'sang gajah', 'serigala', 'harimau'.

Dengan contoh masukan dan keluaran sebagai berikut :

Input : Berikut adalah kisah sang gajah. Sang gajah memiliki teman serigala bernama DoeSang. Gajah sering dibela oleh serigala ketika harimau mendekati gajah.

Output : sang gajah - sang gajah - serigala - serigala - harimau

Jawaban

```
const sangGajah = () => {
  const str =
    'Berikut adalah kisah sang gajah. Sang gajah memiliki teman serigala bernama DoeSang. Gajah sering dibela oleh serigala ketika hari

  console.info(
    str
      .match(/sang gajah|serigala|harimau/gi)
      .join(' - ')
      .toLowerCase()
    );
};

sangGajah();

//Output:
// sang gajah - sang gajah - serigala - serigala - harimau
```

Referensi : <https://www.facebook.com/codingtute/posts/javascript-string-methods-cheat-sheet-save-the-postdownload-the-pdf-file-from-th/208448061196891/>

3. Buatlah fungsi pengecekan kata sandi, dengan ketentuan sebagai

- Kata sandi minimal 8 karakter
- Kata sandi maksimal 32 karakter
- Karakter awal tidak boleh angka
- Harus memiliki angka
- Harus memiliki huruf kapital dan huruf kecil

Contoh

Input : 5andiwar4

Output : Karakter awal tidak boleh angka

Input : sandiwar4

Output : Harus memiliki huruf kapital dan huruf kecil

Input : Sandiwar4

Output : Kata sandi valid

Jawaban

```
import React, { useState } from 'react';
import { useForm } from 'react-hook-form'; //saya menggunakan react hook form untuk validasi login
import { Helmet } from 'react-helmet'; //react helmet berfungsi untuk merubah title di website kita

export default function Register() {
  //state
  const [showPassword, setShowPassword] = useState(false);

  //declare form
  const {
    register,
    watch,
    handleSubmit,
    formState: { errors },
  }
```

```

} = useForm();

//handle form
const handleRegister = (data) => {
  localStorage.setItem(data, JSON.stringify("user"))
  .then(res => console.info(res))
  .catch(err => console.error(err))
};

//toggle password
const togglePassword = () => {
  setShowPassword(!showPassword); //disini saya menggunakan conditional rendering untuk menampilkan/menghide password
};

return (
  <>
    //react helmet yang saya sebut diatas tadi, dan berikut pengaplikasiannya
    <Helmet>
      <meta charSet="utf-8" />
      <title>Register</title>
      <meta name="judul" content="register" />
      <meta name="viewport" content="width=device-width, initial-scale=1.0"></meta>
    </Helmet>
    <div className="w-screen min-h-screen flex justify-center items-center">
      <form className="w-[500px] flex flex-col gap-4 p-4 bg-slate-200 shadow-lg rounded-md" onSubmit={handleSubmit(handleRegister)} autoC
      <div className="w-full flex flex-col gap-2">
        <label htmlFor="text">Email</label>
        <input
          type="text"
          id="email"
          {...register('email', {
            required: {
              value: true,
              message: 'email harus diisi',
            },
            pattern: {
              value: /^[a-zA-Z0-9.!#$%&'*/+=?^`{|}~]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/ ,
              message: 'email tidak valid!',
            },
            //pattern dari react hook form untuk memvalidasi email, agar user harus memasukan email yang valid
            //disini saya menggunakan regex untuk memvalidasinya
            //hal yang sama saya gunakan di password
          })},
          className={`h-10 border-none outline-none px-2 ${errors.email && 'ring-2 ring-red-300'}`}`)
        </div>

        {errors?.email && <small className="text-red-500">{errors?.email.message}</small>}

      <div className="w-full flex flex-col gap-2">
        <label htmlFor="password">Password</label>
        <input
          type={showPassword ? 'text' : 'password'}
          id="password"
          {...register('password', {
            required: {
              value: true,
              message: 'password harus diisi',
            },
            pattern: {
              value: /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,32}$/ ,
              message: 'Minimal 8 karakter, maksimal 32 karakter, setidaknya satu huruf besar, satu huruf kecil, satu angka dan satu kar
            },
          })},
          className="h-10 border-none outline-none px-2"
        </div>

        {errors?.password && <small>{errors?.password.message}</small>}

      <div className="w-full flex flex-col gap-2">
        <label htmlFor="password">Ulangi Password</label>
        <input
          type={showPassword ? 'text' : 'password'}
          id="password2"
          {...register('password2', {
            required: {
              validate: (value) => {
                if (watch('password') !== value) {
                  return (errors.password2 = 'password harus sama');
                }
                //validai untuk mengecek password
              }
            }
          })}
        </div>

```

```

    }
  },
  {}
  className={`h-10 border-none outline-none px-2 ${errors.password2 && 'ring-2 ring-red-300'}`}
  />

  {errors?.password2 && <small>{errors?.password2.message}</small>}

</div>
<button type="button" onClick={togglePassword}>
//berikut implementasi untuk button show / hide password
  Tampilkan password
</button>
<button className="h-10 bg-blue-500 text-white px-5 rounded-md">Register</button>
</form>
</div>
</>
);
}

```

Hasilnya :

Haris Munandar

Bilangan Cacah Validasi Password Slicing

Email

fsfdsgdsg

email tidak valid!

Password

.....

Minimal 8 karakter, maksimal 32 karakter, setidaknya satu huruf besar, satu huruf kecil, satu angka dan satu karakter khusus

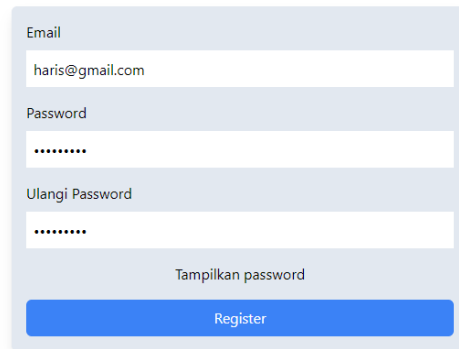
Ulangi Password

.....

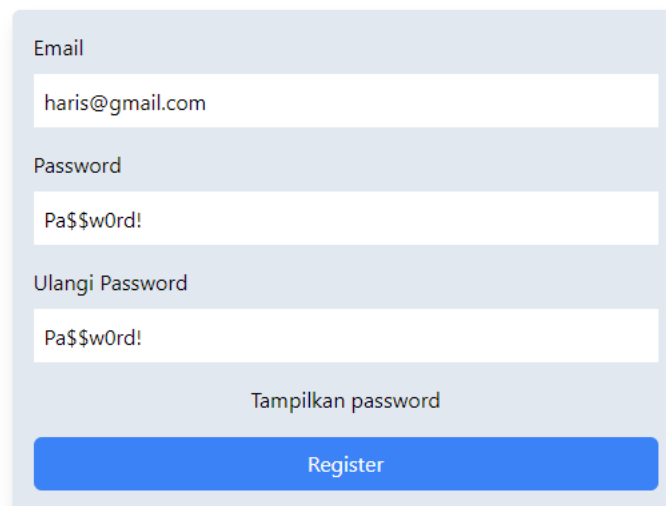
Tampilkan password

Register

Ketika user memasukkan email yang tidak valid, maka akan muncul sebuah warning



A registration form with a light blue background and rounded corners. It contains three input fields: 'Email' with the value 'haris@gmail.com', 'Password' with masked characters '.....', and 'Ulangi Password' (Repeat Password) also with masked characters '.....'. Below the inputs is a link 'Tampilkan password' (Show password) and a blue 'Register' button.



The same registration form as above, but the password fields now show the text 'Pa\$\$w0rd!' instead of masked characters. The 'Tampilkan password' link is still present, and the blue 'Register' button remains at the bottom.

Referensi : react-hook form

Dan berikut adalah ketika user menginputkan data yang benar, maka validasinya tidak muncul, saat user meng-klik tombol tampilkan password maka user dapat melihat passwordnya.

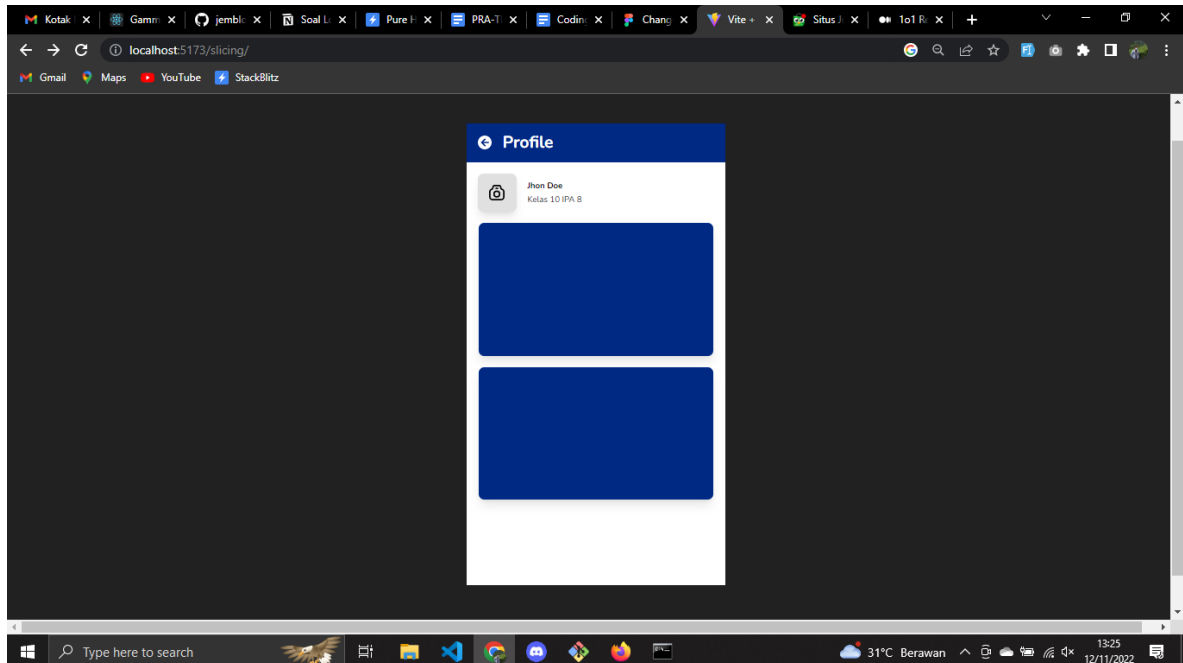
TES Front End

Buat halaman untuk mengubah foto profil dengan ketentuan sebagai berikut

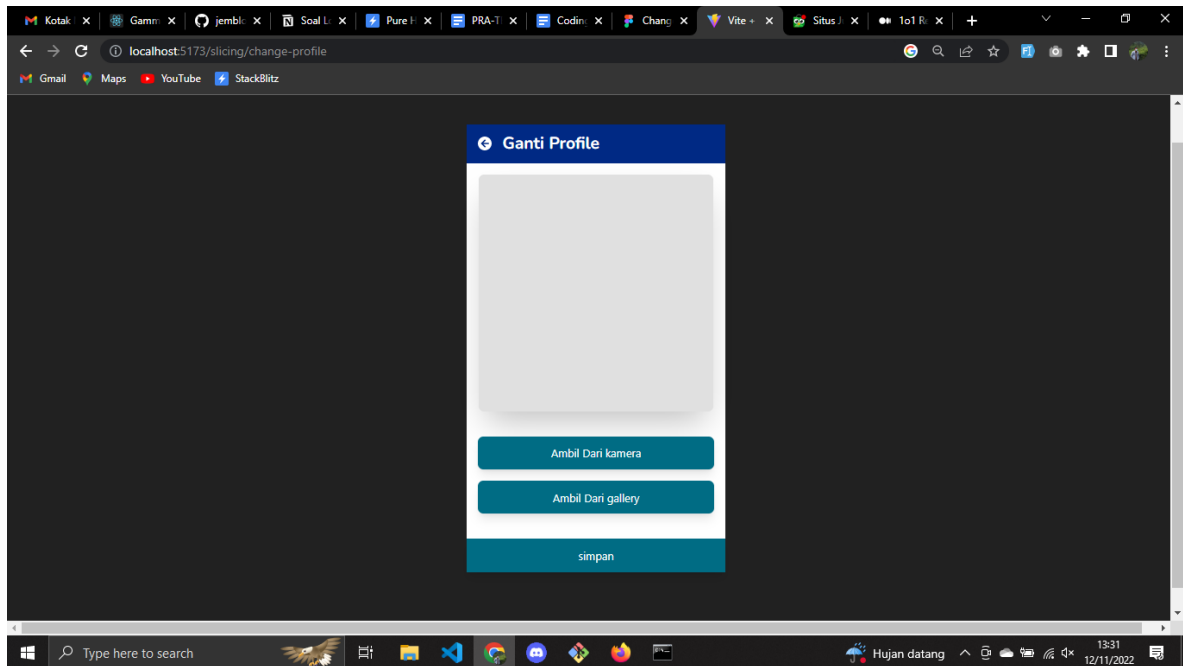
1. Buka figma berikut <https://www.figma.com/file/hAxLcsE9GYU2S0o6TQaL7Z/Change-Profile-Test?node-id=0%3A1>
2. Implementasikan figma di atas **semirip mungkin**
3. Ketika pilih ambil dari kamera, halaman akan berpindah ke halaman ambil foto (sesuai figma). Hasil tangkapan kamera akan dipotong (crop) dan dikompres secara otomatis sesuai bingkai.
4. Ketika ambil dari galeri, pengguna akan memilih 1 gambar dari galeri kemudian kembali ke aplikasi agar gambar dapat dipotong (crop) dan dikompres.
5. Untuk tampilan web desktop, bagian kanan dan kiri diberi warna #212121 , jadi tampilan seperti mobile apps

1. install pnpm create vite

- buat folder
- pnpm i react-router-dom untuk menginisialisasi react router dom karena react adalah SPA (single page application)
- Lalu saya membuat tampilan awal
- Ketika logo camera di click maka user akan dialihkan ke page “ganti prfile”
- Disini saya menggunakan useNavigate yang disediakan oleh react-router-dom package yang sudah saya install sebelumnya



Berikut setelah user meng-click icon camera maka akan menjadi tampilan seperti gambar dibawah:

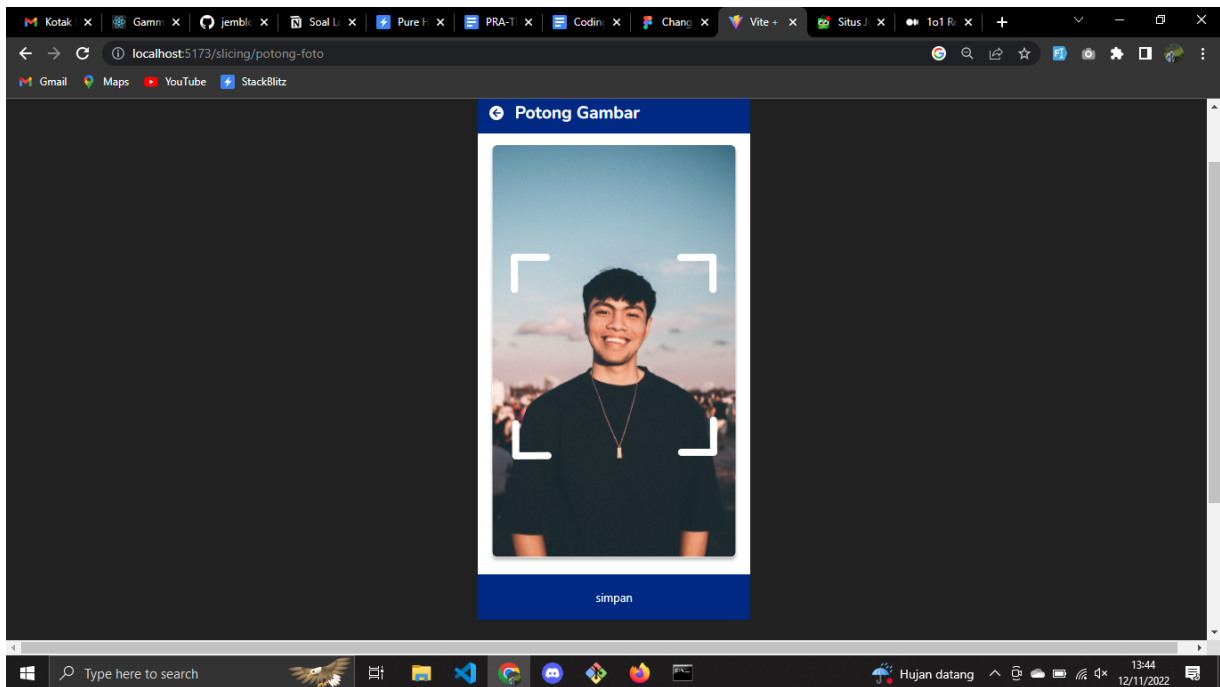


Saya membuat component navbar dengan nesting component menggunakan props, supaya navbar disetiap page bisa dimanis.

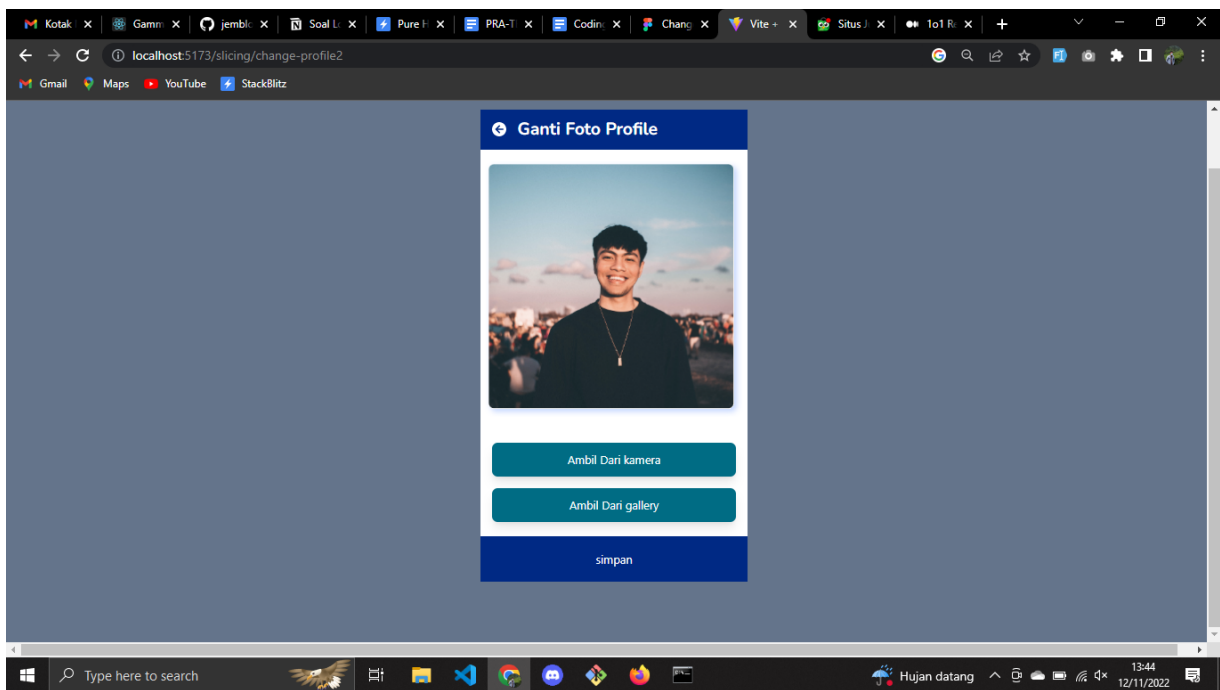
2. Setelah user meng-click "Ambil Dari Camera" maka user akan dialihkan lagi ke page "Ambil Foto"



Lalu ketika user mengambil foto, maka muncul halaman selanjutnya



Ketika simpan akan dialihkan lagi untuk user bisa mengatur ukuran foto



Maka foto user sudah di upload

