

# **LAPORAN KELOMPOK**

## **PROYEK PEMROGRAMAN BERORIENTASI OBJEK**



### **Disusun Oleh :**

Nama Kelompok

1. Bayu Pamungkas (G1F022029)
2. Cindy Andira (G1F022059)
3. Lola Citra Tama (G1F022061)

### **Asisten Dosen :**

Randi Julian Saputra

### **Dosen Pengampu :**

1. Ferzha Putra Utama, S.T.,M.Eng
2. Arie Vatesia, S.T.,M.TI,Ph.D.

**PROGRAM STUDI SISTEM INFORMASI**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS BENGKULU**  
**T.A 2023/2024**

# **BAB I**

## **LANDASAN TEORI**

### **1.1 Pemrograman Berorientasi Objek**

OOP (Object Oriented Programming) merupakan teknik membuat suatu program berdasarkan objek dan apa yang bisa dilakukan objek tersebut. OOP terdiri dari objek-objek yang berinteraksi satu sama lain untuk menyelesaikan sebuah tugas. Kode-kode dibreakdown agar lebih mudah di-manage. Breakdown berdasarkan objek-objek yang ada pada program tersebut. Dianjurkan diimplementasikan untuk program dengan berbagai ukuran karena lebih mudah untuk men-debug. Pemrograman Berorientasi Objek (PBO) atau Object Oriented Programming (OOP) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari PBO/OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, sedangkan objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Misalkan diambil contoh Pesawat, sebagai sebuah objek. Pesawat itu sendiri terbentuk dari beberapa objek yang lebih kecil lagi seperti mesin, roda, baling-baling, kursi, dan lain-lain. Pesawat sebagai objek yang terbentuk dari objek-objek yang lebih kecil saling berhubungan, berinteraksi, berkomunikasi dan saling mengirim pesan kepada objek-objek yang lainnya. Begitu juga dengan program, sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, dimana objek-objek tersebut saling berkomunikasi, dan saling berkiripesan kepada objek yang lain. Sejak diperkenalkannya PBO, sejumlah besar bahasa pemrograman modern sekarang menggunakan konsep ini. Beberapa di antaranya adalah Visual basic, Java, C, dan C++. Ada beberapa masalah kompatibilitas, karena banyak bahasa pemrograman yang tidak dirancang dengan pendekatan PBO. Pemrograman bahasa berorientasi objek yang murni tidak memiliki banyak method yang diperlukan oleh programmer. Pemrograman Terstruktur adalah suatu proses untuk mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dalam bentuk program. Pemrograman Prosedural pada dasarnya adalah metode pemrograman yang berupa baris-baris perintah yang dieksekusi secara urut mulai dari baris atas hingga bawah.

#### **Berikut pembahasan konsep OOP:**

##### **1. Encapsulation**

- Anggap saja kita sedang membangun program dengan beberapa objek berbeda yang saling berkomunikasi satu dengan yang lainnya.

- Enkapsulasi terjadi ketika objek menjaga statusnya agar tetap pribadi dalam kelas.
- Mudahnya objek ini disatukan dan dibuat “kapsul” menjadi satu data.
- Objek bisa berdiri sendiri dalam class dan hanya bisa berkomunikasi dengan objek lain dengan public functions.

## 2. Abstraction

- Abstraction atau abstraksi bisa disebut perpanjangan dari enkapsulasi.
- Dalam proses pembuatan sistem berorientasi objek, seringkali sistem yang dibuat sangat besar.
- Objek saling terpisah dan berkomunikasi dengan menggunakan fungsi, jika mempertahankan seperti ini dalam jangka waktu lama akan sulit dilakukan.
- Untuk itu abstraksi adalah konsep yang dibuat untuk meringankan masalah ini.
- Jadi seorang developer bisa melakukan satu fungsi tanpa harus mengetahui bagaimana fungsi tersebut berjalan.

## 3. Inheritance

- Dalam membangun sistem dengan OOP tentunya memerlukan banyak class sehingga satu dengan lainnya memiliki kemiripan.
- Nah solusinya bisa menggunakan Inheritance bisa disebut warisan.
- Jadi kita membuat class baru (anak) dari class induk yang memiliki fungsi mirip dengan class sebelumnya.
- Inheritance menggunakan sistem hirarki bertingkat, jadi semakin banyak tingkatannya maka classnya memiliki sedikit kemiripan.

## 4. Polymorphism

- Konsep OOP yang keempat adalah polymorphism yang artinya memiliki banyak bentuk.
- Jadi Polymorphism adalah kemampuan dari satu pesan atau data untuk bisa diproses menjadi lebih dari satu bentuk.

### 1.2 Bahasa Pemrograman Python

Python adalah bahasa pemrograman yang banyak digunakan dalam aplikasi web, pengembangan perangkat lunak, ilmu data, dan machine learning (ML). Developer menggunakan Python karena efisien dan mudah dipelajari serta dapat dijalankan di berbagai platform. Perangkat lunak Python dapat diunduh secara gratis, terintegrasi baik dengan semua tipe sistem, dan meningkatkan kecepatan pengembangan. Python adalah salah satu bahasa pemrograman yang dapat melakukan eksekusi sejumlah instruksi multi guna secara langsung

(interpretatif) dengan metode orientasi objek (Object Oriented Programming) serta menggunakan semantik dinamis untuk memberikan tingkat keterbacaan syntax. Sebagian lain mengartikan Python sebagai bahasa yang kemampuan, menggabungkan kapabilitas, dan sintaksis kode yang sangat jelas, dan juga dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Walaupun Python tergolong bahasa pemrograman dengan level tinggi, nyatanya Python dirancang sedemikian rupa agar mudah dipelajari dan dipahami. Python adalah bahasa pemrograman multi-paradigma. Pemrograman berorientasi objek dan pemrograman terstruktur juga didukung penuh, dan banyak fiturnya mendukung pemrograman fungsional dan pemrograman berorientasi aspek (termasuk dengan metaprogramming dan metaobjects (metode ajaib)). Banyak paradigma lain yang didukung melalui ekstensi, termasuk desain berdasarkan kontrak dan pemrograman logika.

### **Sejarah Bahasa Python**

#### **1980-an: Konsep Awal dan Pengembangan**

- Bahasa pemrograman Python pertama kali dikonsep oleh Guido van Rossum pada akhir tahun 1980-an saat dia bekerja di Institut Penelitian Matematika dan Ilmu Komputer (CWI) di Belanda.
- Guido ingin menciptakan bahasa pemrograman yang lebih sederhana, mudah dibaca, dan intuitif bagi para pengembang. Dia terinspirasi oleh bahasa ABC yang dikembangkan di CWI. Sehingga bahasa Python tak jarang disebut sebagai penerus ABC.
- Pada Februari 1991, versi pertama Python, Python 0.9.0, dirilis dengan fitur-fitur awal.

#### **1990-an: Perkembangan Awal dan Versi Pertama**

- Pada tahun 1991, Guido merilis versi pertama Python (Python 0.9.0) dengan fitur seperti blok kode terindentasi, pengelolaan eksepsi, dan beberapa tipe data dasar.
- Python 1.0 dirilis pada tahun 1994 dengan penambahan fitur-fitur seperti dukungan penuh untuk pemrograman berorientasi objek (OOP).

#### **2000-an: Python 2, Peningkatan dan Ekspansi**

- Python 2.0 dirilis pada tahun 2000 dengan penambahan fitur seperti *list comprehensions*, *garbage collection*, dan dukungan *Unicode*.
- Versi Python 2.7, dirilis pada tahun 2010, menjadi versi terakhir dari cabang Python 2. Python 2 tetap berlanjut dalam penggunaan sebagian besar hingga akhir 2019, meskipun pengembangan resmi telah berfokus pada Python 3.

### 2008: Pengenalan Python 3

- Python 3.0 dirilis pada tahun 2008 dengan perubahan signifikan dalam sintaksis dan fitur. Python 3.0 bertujuan untuk membersihkan bahasa dan mengatasi masalah desain dari Python 2.
- Pemisahan antara string dan byte menjadi lebih jelas di Python 3, yang menghasilkan lebih sedikit kebingungan tentang pengelolaan karakter.

### 2010-an: Pertumbuhan Pesat dan Popularitas

- Python semakin populer dalam pengembangan web, analisis data, kecerdasan buatan, dan banyak bidang lainnya.
- Dukungan kuat dari komunitas pengembang dan ekosistem library yang berkembang membuat Python semakin diminati.

### 2020-an: Masa Kehadiran yang Kuat

- Python terus menjadi salah satu bahasa pemrograman paling populer dan serbaguna di dunia.
- Python banyak digunakan dalam berbagai industri, termasuk teknologi, ilmu data, kecerdasan buatan, pengembangan web, dan lainnya.

### Manfaat Mempelajari Python

#### 1. Peluang Mendapatkan Pekerjaan dengan Gaji Besar

Menurut situs Indeed, Python adalah bahasa pemrograman dengan bayaran tertinggi kedua. Dengan memiliki keahlian ini, kamu bisa melamar pekerjaan di perusahaan besar dengan gaji yang menggiurkan.

#### 2. Mudah Dipahami

Salah satu manfaat Python adalah kemudahannya. Tidak hanya mudah dipahami, ia juga mudah digunakan. Sintaksnya menggunakan bahasa Inggris sehingga mudah dipelajari. Selain itu, ia juga bisa menangani kompleksitas dengan baik. Dan yang terpenting, sifatnya itu gratis dan *open-source*.

#### 3. Fleksibel

Python tidak hanya mudah dipelajari tapi juga fleksibel. Ada lebih dari 125.000 library dari pihak ketiga yang memungkinkan kamu menggunakannya untuk machine learning, pemrosesan website bahkan biologi. Selain itu, *library* yang berfokus pada data seperti pandas, NumPy dan matplotlib membuatnya sangat mudah memproses, memanipulasi serta memvisualisasikan data.

## BAB II

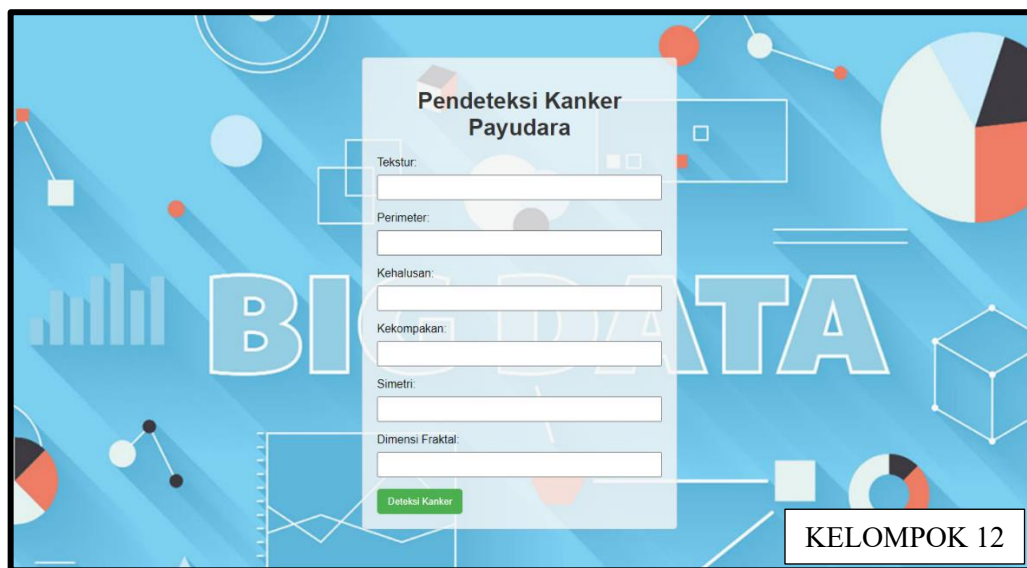
### SOAL DAN PEMBAHASAN

#### 2.1 Soal

Prediksi kanker payudara menggunakan Python: Proyek ilmu data ini membantu mendeteksi Kanker Payudara menggunakan kumpulan data.

#### 2.2 Pembahasan

Pembahasan Tampilan:

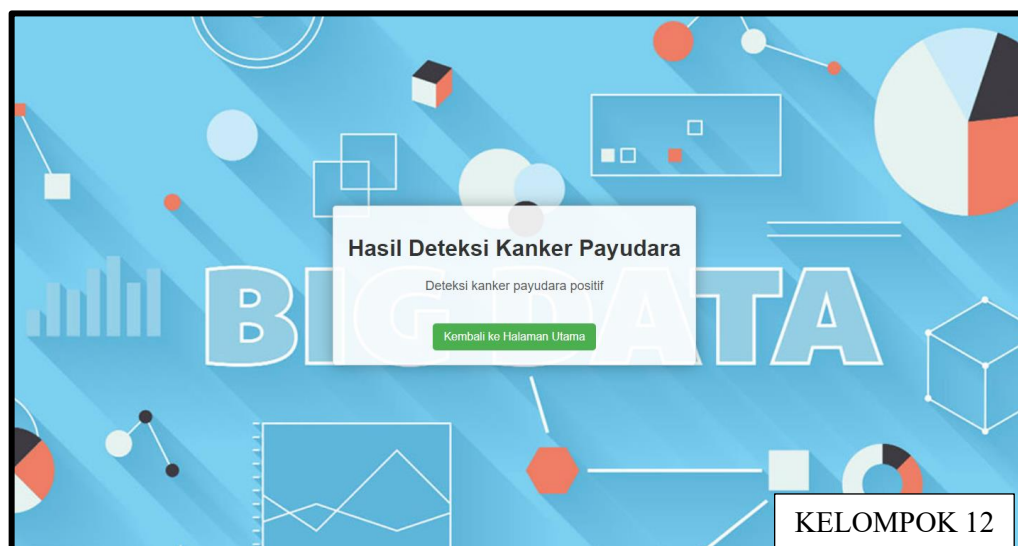


The screenshot shows a web application interface with a blue background featuring various data visualization icons like bar charts, pie charts, and line graphs. The word 'BIG DATA' is prominently displayed in the center. A white form titled 'Pendeteksi Kanker Payudara' is centered on the screen. The form contains six input fields labeled 'Tekstur:', 'Perimeter:', 'Kehalusan:', 'Kekompakan:', 'Simetri:', and 'Dimensi Fraktal:'. Below these fields is a green button labeled 'Deteksi Kanker'. In the bottom right corner, there is a white box with the text 'KELOMPOK 12'.

Gambar 2.1 Tampilan awal Pendeteksi kanker Payudara

Pembahasan:

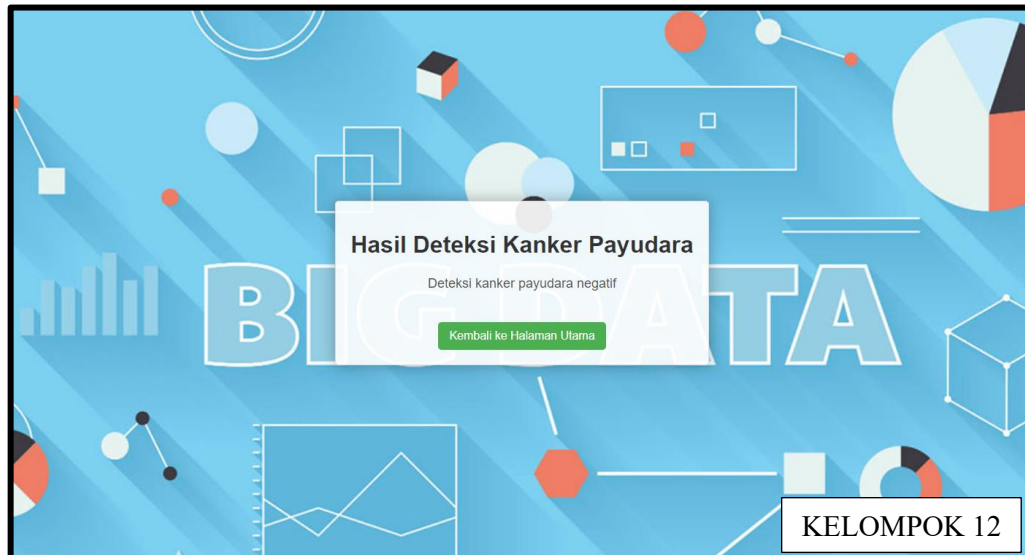
Pada gambar 2.1 tersebut merupakan gambar tampilan awal untuk mendeteksi kanker payudara. Pengguna bisa memasukan data-data yang telah disediakan seperti tekstur, perimeter, kehalusan, kekompakan, simetri, dan juga dimensi faktal agar bisa dideteksi kankernya tersebut



Gambar 2.2 Tampilan dari hasil pendeteksi kanker payudara yang positif

Pembahasan:

Pada gambar 2.2 tersebut merupakan gambar tampilan dari hasil pendeteksi kanker payudara yang bersifat positif. Pengguna bisa melihat hasil dari data-data yang telah mereka masukan sebelumnya yang bersifat positif bahwasanya pengguna diaknosa memiliki penyakit kanker payudara.



Gambar 2.3 Tampilan dari hasil pendeteksi kanker payudara yang negatif

Pembahasan:

Pada gambar 2.3 tersebut merupakan gambar tampilan dari hasil pendeteksi kanker payudara yang bersifat negatif. Pengguna bisa melihat hasil dari data-data yang telah mereka masukan sebelumnya yang bersifat positif bahwasanya pengguna diaknosa tidak memiliki penyakit kanker payudara.

### Pembahasan Code:

File (deteksi\_kanker.py)

```
from flask import Flask, render_template, request

app = Flask(__name__)

class PendeteksiKankerPayudara:
    def __init__(self, tekstur, perimeter, kehalusan, kekompakan, simetri, dimensi_fraktal):
        self.tekstur = tekstur
        self.perimeter = perimeter
        self.kehalusan = kehalusan
        self.kekompakan = kekompakan
        self.simetri = simetri
        self.dimensi_fraktal = dimensi_fraktal
```

```

def analisis_fitur(self):
    hasil = self.tekstur * 0.5 + self.perimeter * 0.3 + self.kehalusan *
0.2
    return hasil

def deteksi_kanker(self):
    hasil_analisis = self.analisis_fitur()

    if hasil_analisis > 0.7:
        return "Deteksi kanker payudara positif"
    else:
        return "Deteksi kanker payudara negatif"

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        tekstur = float(request.form['tekstur'])
        perimeter = float(request.form['perimeter'])
        kehalusan = float(request.form['kehalusan'])
        kekompakan = float(request.form['kekompakan'])
        simetri = float(request.form['simetri'])
        dimensi_fraktal = float(request.form['dimensi_fraktal'])

        data_pasien = PendeteksiKankerPayudara(tekstur, perimeter, kehalusan,
kekompakan, simetri, dimensi_fraktal)

        hasil = data_pasien.deteksi_kanker()

        return render_template('hasil.html', hasil=hasil)

    return render_template('index.html')

if __name__ == '__main__':
    app.run(debug=True)

```

Berikut adalah penjelasan singkat dari setiap bagian kode:

1. Import Modul Flask dan Dependencies:

```
from flask import Flask, render_template, request
```

Mengimport kelas Flask, render\_template, dan request dari modul Flask.

2. Inisialisasi Aplikasi Flask:

```
app = Flask(__name__)
```

Membuat instance Flask dengan nama aplikasi `app`.

3. Kelas PendeteksiKankerPayudara:

```
class PendeteksiKankerPayudara:
```

```
# ... (konstruktor dan metode lainnya)
```



Membuat kelas untuk objek PendeteksiKankerPayudara dengan atribut seperti tekstur, perimeter, kehalusan, dll., dan metode untuk analisis fitur dan deteksi kanker.

4. Route Utama ('/') dengan Metode GET dan POST:

```
@app.route('/', methods=['GET', 'POST'])
def index():
    # ... (logika untuk menangani permintaan GET dan POST)
```

Mendefinisikan route utama yang merespons permintaan HTTP GET dan POST. Jika permintaan adalah POST, data formulir diproses untuk membuat objek PendeteksiKankerPayudara dan hasilnya dirender pada template 'hasil.html'. Jika permintaan adalah GET, tampilkan template 'index.html'.

5. Logika Pengolahan Data POST:

```
if request.method == 'POST':
    # ... (mengambil data formulir, membuat objek PendeteksiKankerPayudara, dan mendeteksi kanker)
```

Jika permintaan adalah POST, ambil data formulir yang dikirim, konversi ke tipe data numerik, buat objek PendeteksiKankerPayudara, dan tentukan hasil deteksi kanker.

6. Menjalankan Aplikasi Flask:

```
if __name__ == '__main__':
    app.run(debug=True)
```

Pada program di atas, konsep OOP diterapkan dengan cara berikut:

- Kelas PendeteksiKankerPayudara didefinisikan sebagai kelas yang mewakili objek pendeteksi kanker payudara. Kelas ini memiliki data tekstur, perimeter, kehalusan, kekompakan, simetri, dan dimensi\_fraktal. Kelas ini juga memiliki perilaku analisis\_fitur() dan deteksi\_kanker().
- Objek data\_pasien dari kelas PendeteksiKankerPayudara dibuat dengan memasukkan data dari formulir.
- Metode deteksi\_kanker()`` dari objekdata\_pasien` digunakan untuk melakukan deteksi kanker payudara.

Berikut adalah penjelasan lebih rinci dari masing-masing konsep OOP yang diterapkan pada program di atas:

- Pemrograman berorientasi objek (OOP)

Pemrograman berorientasi objek diterapkan pada program di atas dengan cara membagi program menjadi kumpulan objek yang saling berinteraksi. Objek-objek tersebut mewakili konsep-konsep yang ada di dunia nyata, seperti pendeteksi kanker payudara.

- Objek

Objek PendeteksiKankerPayudara mewakili objek pendeteksi kanker payudara di dunia nyata. Objek ini memiliki data tekstur, perimeter, kehalusan, kekompakan, simetri, dan dimensi\_fraktal. Data-data tersebut mewakili fitur-fitur yang digunakan untuk mendeteksi kanker payudara.

- Kelas

Kelas PendeteksiKankerPayudara adalah cetak biru untuk membuat objek PendeteksiKankerPayudara. Kelas ini menentukan data dan perilaku yang akan dimiliki oleh objek PendeteksiKankerPayudara.

- Instansi

Objek data\_pasien adalah instansi dari kelas PendeteksiKankerPayudara. Objek ini dibuat dengan memasukkan data dari formulir.

- Metode `analisis\_fitur()``

Metode analisis\_fitur()`` digunakan untuk menghitung skor dari fitur-fitur yang diberikan. Skor ini kemudian digunakan oleh metodedeteksi\_kanker()`` untuk menentukan apakah kanker payudara positif atau negatif.

- Metode `deteksi\_kanker()``

Metode deteksi\_kanker()`` digunakan untuk mendeteksi kanker payudara berdasarkan skor yang dihasilkan oleh metodeanalisis\_fitur()``.

index.html

```
<!DOCTYPE html>

<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Pendeteksi Kanker Payudara</title>
  <style>
    body {
      font-family: 'Arial', sans-serif;
      margin: 0;
      padding: 0;
      background-image: url('static/kanker.jpg');
      background-size: cover;
```

```

background-position: center;
background-repeat: no-repeat;
display: flex;
align-items: center;
justify-content: center;
height: 100vh;
margin: 0;
}

form {
    max-width: 400px;
    margin: 20px auto;
    padding: 20px;
    border: 1px solid #ddd;
    border-radius: 5px;
    background-color: rgba(248, 248, 248, 0.8); /* Ubah nilai alpha
untuk transparansi */
}

h1 {
    color: #333;
    text-align: center; /* Menengahkan teks "Pendeteksi Kanker
Payudara" */
}

label {
    display: block;
    margin-bottom: 8px;
}

input {
    width: 100%;
    padding: 8px;
    margin-bottom: 16px;
    box-sizing: border-box;
}

button {
    background-color: #4caf50;
    color: white;
    padding: 10px 15px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
}
</style>

```

```

</head>
<body>

  <form method="post" action="/">
    <h1>Pendeteksi Kanker Payudara</h1>
    <label for="tekstur">Tekstur:</label>
    <input type="text" name="tekstur" required><br>

    <label for="perimeter">Perimeter:</label>
    <input type="text" name="perimeter" required><br>

    <label for="kehalusan">Kehalusan:</label>
    <input type="text" name="kehalusan" required><br>

    <label for="kekompakan">Kekompakan:</label>
    <input type="text" name="kekompakan" required><br>

    <label for="simetri">Simetri:</label>
    <input type="text" name="simetri" required><br>

    <label for="dimensi_fraktal">Dimensi Fraktal:</label>
    <input type="text" name="dimensi_fraktal" required><br>

    <button type="submit">Deteksi Kanker</button>
  </form>
</body>
</html>

```

Berikut adalah penjelasan singkat dari setiap bagian kode:

Deklarasi dokumen dan pengaturan awal:

- `<!DOCTYPE html>`: Menyatakan dokumen ini sebagai sebuah file HTML.
- `<html lang="en">`: Menandakan bahwa bahasa utama dokumen ini adalah bahasa Inggris.

Pengaturan kepala dokumen:

- `<head>`: Bagian ini berisi informasi meta tentang dokumen yang tidak ditampilkan langsung di halaman.
- `<meta charset="UTF-8">`: Menentukan karakter encoding yang digunakan untuk menampilkan karakter dengan benar.
- `<meta http-equiv="X-UA-Compatible" content="IE=edge">`: Memastikan kompatibilitas tampilan halaman dengan Internet Explorer.
- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Menyesuaikan tampilan halaman untuk berbagai ukuran layar perangkat.
- `<title>Pendeteksi Kanker Payudara</title>`: Menentukan judul halaman web yang akan ditampilkan pada tab browser.

Penataan gaya:

- `<style>`: Bagian ini berisi pengaturan gaya visual halaman web.
- Aturan-aturan CSS di dalam `<style>` mengatur berbagai aspek tampilan elemen HTML, seperti font, warna, background image, layout, dan sebagainya.

Penataan isi halaman:

- `<body>`: Bagian ini berisi elemen-elemen yang akan ditampilkan langsung pada halaman web.
- `<form method="post" action="/">`: Memulai formulir pengisian data yang akan dikirimkan dengan metode POST ke halaman yang sama ("/").
- `<h1>Pendeteksi Kanker Payudara</h1>`: Menambahkan judul "Pendeteksi Kanker Payudara" di tengah halaman.
- Label dan input untuk 6 fitur pemeriksaan payudara: Tekstur, Perimeter, Kehalusan, Kekompakan, Simetri, dan Dimensi Fraktal.
- Tombol "Deteksi Kanker" untuk mengirimkan data formulir.

Penutup deklarasi:

- `</body>`: Menutup tag body yang menandakan akhir isi halaman.
- `</html>`: Menutup tag html yang menandakan akhir dokumen.

Meskipun kode HTML yang diberikan sendiri tidak secara langsung menunjukkan implementasi OOP, kita dapat menganalisis potensinya dan memberikan penjelasan konseptual:

Potensi Penggunaan OOP:

1. Kelas Data Pasien: Formulir pengisian data dapat mewakili representasi objek Data Pasien. Kelas ini menyimpan informasi seperti tekstur, perimeter, kehalusan, kekompakan, simetri, dan dimensi fraktal sebagai atributnya.
2. Kelas Pendeteksi Kanker: Logika pemrosesan data dan deteksi kanker bisa diimplementasikan dalam kelas Pendeteksi Kanker. Kelas ini memiliki metode untuk menerima data pasien, melakukan analisis fitur berdasarkan model tertentu, dan memberikan hasil deteksi (positif, negatif, atau risiko tertentu).
3. Enkapsulasi dan Abstraksi: Kelas Pendeteksi Kanker dapat mengenkapsulasi logika pemrosesan dan model deteksi, menyembunyikan detail implementasi dari pengguna (abstraksi).

4. Polimorfisme (opsional): Kelas Pendeteksi Kanker bisa diadaptasi dengan berbagai algoritma deteksi kanker (misalnya, Naive Bayes, Support Vector Machine), memungkinkan fleksibilitas dan peningkatan akurasi deteksi.

Hasil.html

```
<!DOCTYPE html>

<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Hasil Deteksi Kanker Payudara</title>
  <style>
    body {
      font-family: 'Arial', sans-serif;
      margin: 0;
      padding: 0;
      background-image: url('static/kanker.jpg');
      background-size: cover;
      background-position: center;
      background-repeat: no-repeat;
      display: flex;
      align-items: center;
      justify-content: center;
      height: 100vh;
      margin: 0;
    }

    .result-container {
      max-width: 600px;
      padding: 20px;
      border: 1px solid #ddd;
      border-radius: 5px;
      box-shadow: 0 0 20px rgba(0, 0, 0, 0.3);
      background-color: rgba(255, 255, 255, 0.9);
      text-align: center;
    }

    h1 {
      color: #333;
      margin-bottom: 20px;
    }

    p {
      font-size: 18px;
```

```

    color: #555;
    margin-top: 20px;
    line-height: 1.6; /* Menambahkan jarak antar baris */
}

button {
    background-color: #4caf50;
    color: white;
    padding: 10px 15px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    margin-top: 20px;
    font-size: 16px;
    transition: background-color 0.3s ease; /* Efek transisi warna latar
belakang */
}

button:hover {
    background-color: #45a049;
}

</style>
</head>
<body>

    <div class="overlay">
        <div class="result-container">
            <h1>Hasil Deteksi Kanker Payudara</h1>
            <p>{{ hasil }}</p>
            <button onclick="redirectToIndex()">Kembali ke Halaman
Utama</button>
        </div>
    </div>

    <script>
        function redirectToIndex() {
            window.location.href = "/";
        }
    </script>

</body>
</html>

```

Berikut adalah penjelasan singkat dari setiap bagian kode:

Deklarasi dokumen dan pengaturan awal:

- Sama seperti index.html, bagian awal mendeklarasikan tipe dokumen, bahasa dan mengatur tampilan dasar halaman.

Penataan gaya:

- Gaya dasar body sama seperti index.html dengan background image.
- `.result-container` mengatur tampilan box berisi hasil deteksi, dengan box-shadow dan background transparan.
- Gaya elemen lain seperti h1, p dan button juga diatur untuk tampilan yang lebih informatif.

Penataan isi halaman:

- `<div class="overlay">`: Membungkus seluruh konten dengan kelas "overlay" untuk efek overlay pada halaman sebelumnya.
- `.result-container`: Menampung elemen-elemen hasil deteksi.
- `<h1>Hasil Deteksi Kanker Payudara</h1>`: Judul halaman.
- `<p>{{ hasil }}</p>`: Paragraf untuk menampilkan hasil deteksi yang diambil dari variabel hasil.
- `<button onclick="redirectToIndex()">Kembali ke Halaman Utama</button>`: Tombol untuk kembali ke halaman index.html dengan fungsi javascript redirectToIndex().

Skrip:

- `function redirectToIndex()`: Fungsi javascript untuk redirect ke halaman index.html saat tombol ditekan.

Meskipun kode HTML ini tidak secara langsung menunjukkan implementasi OOP, kita dapat menganalisis potensi penggunaannya berdasarkan struktur halaman:

Potensi Penggunaan OOP:

1. Objek Hasil Deteksi: Mungkin ada Objek Hasil Deteksi yang menyimpan informasi seperti status deteksi (positif, negatif, atau risiko tertentu) dan pesan penjelasan lainnya. Halaman ini kemudian mengambil dan menampilkan data dari objek tersebut.
2. Kelas Tampilan Hasil: Kelas ini bisa bertanggung jawab untuk mengatur format dan gaya tampilan halaman hasil deteksi, termasuk pesan dan tombol kembali.
3. Enkapsulasi dan Abstraksi: Kelas Tampilan Hasil dapat mengenkapsulasi logika pengaturan elemen HTML dan gaya, menyembunyikan detail implementasi dari pengguna (abstraksi).
4. Polimorfisme (opsional): Berdasarkan jenis hasil deteksi (positif, negatif, risiko), kelas Tampilan Hasil bisa menyesuaikan konten dan gaya pesan yang ditampilkan (misalnya, warna latar belakang berbeda untuk hasil positif dan negatif).



## **BAB III**

### **KESIMPULAN DAN SARAN**

#### **3.1 Kesimpulan**

OOP (Object Oriented Programming) merupakan teknik membuat suatu program berdasarkan objek dan apa yang bisa dilakukan objek tersebut. Pemrograman Berorientasi Objek (PBO) atau Object Oriented Programming (OOP) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari PBO/OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari.

Python adalah bahasa pemrograman yang banyak digunakan dalam aplikasi web, pengembangan perangkat lunak, ilmu data, dan machine learning (ML). Developer menggunakan Python karena efisien dan mudah dipelajari serta dapat dijalankan di berbagai platform. Python adalah salah satu bahasa pemrograman yang dapat melakukan eksekusi sejumlah instruksi multi guna secara langsung (interpretatif) dengan metode orientasi objek (Object Oriented Programming) serta menggunakan semantik dinamis untuk memberikan tingkat keterbacaan syntax.

#### **3.2 Saran**

Dalam pengimplementasian pemrograman berorientasi objek dengan menggunakan Bahasa Pemrograman Python ini sebenarnya cukup mudah, kita perlu mempelajari dasar-dasar dari Bahasa pemrograman Python itu sendiri. terlebih Pada pembuatan tugas ini kami menyarankan sebelum menggunakan Bahasa pemrograman Python dalam OOP sangat harus memahaminya terlebih dahulu karena jika tidak paham apa itu Python dalam OOP maka pada saat proses pembuatan akan terjadi error atau tidak bisa di jalankan.

## DAFTAR PUSTAKA

- Eril Obeit Choiri, (Maret 2022). "*OOP: Pengertian, Konsep, Kelebihan dan Kekurangannya*"  
<https://qwords.com/blog/pengertian-oop/> Diakses pada tanggal 10 Desember 2023.
- Heni Jusuf (2020). "*Konsep Pemrograman Berorientasi Objek*"  
<https://pustaka.ut.ac.id/lib/wp-content/uploads/pdfmk/MSIM4301-M1A.pdf>  
Diakses pada tanggal 11 Desember 2023.



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
UNIVERSITAS BENGKULU  
**FAKULTAS TEKNIK**  
**PROGRAM STUDI SISTEM INFORMASI**

Jalan W.R. Supratman Kandang Limun Bengkulu  
Bengkulu 38371 A Telepon : (0736) 344087, 22105 – 227

**LEMBAR ACC PROYEK PEMEROGRAMAN BERORIENTASI OBJEK**

Nama Kelompok :

1. Bayu Pamungkas (G1F022029)
2. Cindy Andira (G1F022059)
3. Lola Citra Tama (G1F022061)

Dosen :

1. Ferzha Putra Utama, S.T.,M.Eng
2. Arie Vatesia, S.T.,M.TI,Ph.D.

Asisten : Randi Julian Saputra

Laporan Praktikum  
Hari/Tanggal Laporan diberikan

Tanda ACC  
Hari/Tanggal Laporan di ACC

Laporan Tugas Akhir