

# **Tugas**

## **II3140 - Pengembangan Aplikasi Web dan Mobile**

Mata Kuliah: Pengembangan Aplikasi Web dan Mobile

Dosen:

Daniel Wiyogo Dwiputro, S.T., M.T.

Atina Putri, S.Kom., M.T.

Dr. Fadhil Hidayat, S.Kom., M.T.



Nama: Kenzie Raffa Ardhana

NIM: 18223127

**PROGRAM STUDI SISTEM DAN TEKNOLOGI INFORMASI**  
**FAKULTAS SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**2025**

## Daftar Isi

<b>BAB I.....</b>	<b>4</b>
<b>PENDAHULUAN.....</b>	<b>4</b>
1.1 Latar Belakang.....	4
1.2 Rumusan Masalah.....	4
1.3 Tujuan.....	5
1.4 Manfaat.....	5
<b>BAB II.....</b>	<b>5</b>
<b>LANDASAN TEORI.....</b>	<b>5</b>
<b>BAB III.....</b>	<b>6</b>
<b>PERANCANGAN SISTEM.....</b>	<b>6</b>
3.1 Arsitektur Sistem.....	6
3.2 Fitur Utama.....	6
3.3 Kurikulum.....	13
<b>BAB IV.....</b>	<b>13</b>
<b>IMPLEMENTASI.....</b>	<b>13</b>
4.1 Teknologi yang Digunakan.....	13
4.2 Desain Antarmuka.....	13
4.3 Simulasi Eksekusi Kode.....	14
4.4 Mekanisme Skoring.....	14
4.5 Contoh Soal.....	14
<b>BAB V.....</b>	<b>15</b>
<b>HASIL DAN PEMBAHASAN.....</b>	<b>15</b>
5.1 Keunggulan.....	15
5.2 Keterbatasan.....	15
5.3 Rencana Pengembangan.....	15
<b>BAB VI.....</b>	<b>15</b>
<b>RENCANA PENGEMBANGAN.....</b>	<b>15</b>
6.1 Integrasi Interpreter Python Asli.....	16
6.2 Sistem Akun & Penyimpanan Progres.....	16
6.3 Bank Soal Adaptif.....	16
6.4 Gamifikasi & Leaderboard.....	16
6.5 Kolaborasi & Diskusi.....	16
6.6 Dukungan Materi Lanjutan.....	17
6.7 Deploy Skala Lebih Besar.....	17
<b>BAB VII.....</b>	<b>17</b>
<b>KESIMPULAN.....</b>	<b>17</b>

## Daftar Gambar

Gambar 3.2.1 Navigasi Dinamis.....	6
Gambar 3.2.2.2 Beranda.....	7
Gambar 3.2.2.3 Beranda.....	7
Gambar 3.2.2.4 Beranda.....	8
Gambar 3.2.2.5 Beranda.....	8
Gambar 3.2.3.1 Bab 1.....	9
Gambar 3.2.3.2 Bab 2.....	9
Gambar 3.2.3.3 Bab 3.....	9
Gambar 3.2.3.4 Bab 4.....	10
Gambar 3.2.3.5 Bab 5.....	10
Gambar 3.2.4 Editor Kode Simulasi.....	11
Gambar 3.2.5.1 Latihan Soal Terintegrasi.....	12
Gambar 3.2.5.2 Latihan Soal Terintegrasi.....	12
Gambar 3.2.6.1 Drag&Drop.....	13
Gambar 3.2.6.2.2 Visualisasi Canvas.....	14
Gambar 3.2.6.2.2 Visualisasi Canvas.....	14

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Mata kuliah *Berpikir Komputasional* di TPB ITB merupakan fondasi bagi mahasiswa untuk memahami logika pemrograman dan penyelesaian masalah dengan komputer. Namun, pembelajaran pemrograman sering kali dianggap sulit karena:

- Konsep yang abstrak dan teoritis.
- Perlu instalasi software tambahan (Python, IDE, dll).
- Kurangnya interaktivitas dalam pembelajaran.

Untuk menjawab tantangan tersebut, dikembangkan Virtual Lab – sebuah platform pembelajaran berbasis web yang memungkinkan mahasiswa belajar Python langsung di browser tanpa instalasi tambahan. Virtual Lab menggabungkan materi, editor kode interaktif, latihan soal dengan penilaian otomatis, video pembelajaran, dan aktivitas visualisasi dalam satu platform.

### 1.2 Rumusan Masalah

1. Bagaimana menyediakan media pembelajaran pemrograman Python yang mudah diakses tanpa instalasi?
2. Bagaimana membuat pembelajaran lebih interaktif dengan simulasi kode, latihan soal, dan visualisasi?
3. Bagaimana menilai pemahaman mahasiswa secara otomatis melalui latihan soal berbasis kasus nyata?

### 1.3 Tujuan

- Mengembangkan platform web interaktif untuk belajar pemrograman Python.
- Menyediakan kurikulum modular dari dasar hingga konsep fungsi.
- Memberikan pengalaman *learning by doing* melalui editor kode, drag-and-drop, dan visualisasi.
- Menyediakan sistem latihan dengan umpan balik instan dan penilaian otomatis.

## **1.4 Manfaat**

- Mahasiswa: belajar lebih menyenangkan, praktis, dan bisa langsung praktik.
  - Dosen: media tambahan untuk memperkuat materi kuliah.
  - Institusi: inovasi pembelajaran digital sesuai era modern.
- 

## **BAB II**

### **LANDASAN TEORI**

1. Berpikir Komputasional  
Merupakan pendekatan penyelesaian masalah dengan membagi persoalan menjadi komponen lebih kecil (dekomposisi), mengenali pola, membuat abstraksi, dan merancang algoritma.
  2. Pemrograman Python  
Bahasa pemrograman populer untuk pemula karena sintaksnya sederhana, mendukung paradigma multiprogramming, dan banyak digunakan di industri.
  3. E-Learning Interaktif  
Konsep pembelajaran yang memanfaatkan teknologi interaktif seperti simulasi, latihan soal otomatis, dan visualisasi untuk meningkatkan keterlibatan mahasiswa.
- 

## **BAB III**

### **PERANCANGAN SISTEM**

#### **3.1 Arsitektur Sistem**

Virtual Lab dibangun berbasis client-side tanpa backend. Struktur file utama:

- index.html → struktur halaman.
- styles.css → tampilan antarmuka (UI/UX).
- script.js → logika interaktif, editor simulasi, soal & skor.

## 3.2 Fitur Utama

### 1. Navigasi Dinamis

- Dibangun dengan JavaScript (showPage()) untuk berpindah antar halaman tanpa reload.
- Menu: Beranda, Materi, Latihan, Drag&Drop, Visualisasi.



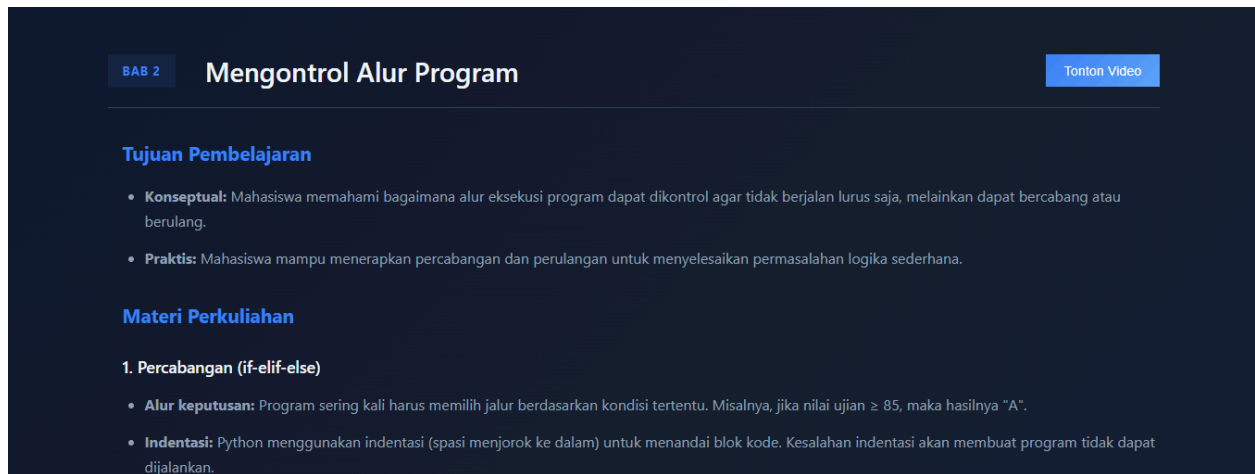
Gambar 3.2.1 Navigasi Dinamis

### 2. Beranda

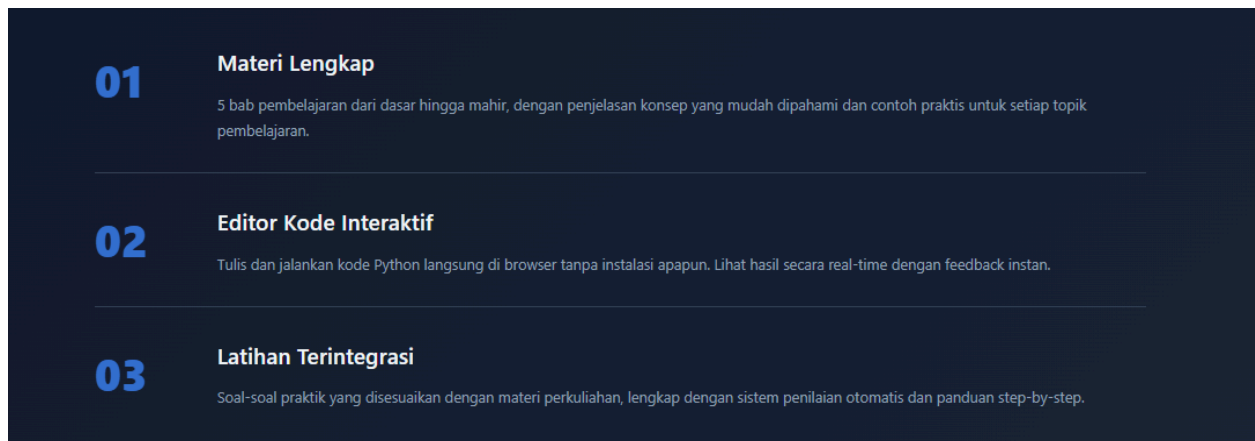
- Menyambut pengguna dengan **hero section** berisi judul “*Selamat Datang di Virtual Lab Berpikir Komputasional*”.
- Memberikan deskripsi singkat tujuan platform: media interaktif untuk belajar pemrograman Python.
- Menyediakan tombol “**Mulai Belajar Sekarang**” yang langsung mengarahkan pengguna ke halaman materi.
- Menampilkan **highlight fitur platform** dalam bentuk 3 poin utama:
  - Materi Lengkap (5 bab dengan penjelasan teori dan contoh).
  - Editor Kode Interaktif (menulis & menjalankan kode langsung di browser).
  - Latihan Terintegrasi (soal dengan penilaian otomatis).
- Terdapat bagian **Kurikulum Timeline** yang menampilkan daftar 5 bab pembelajaran secara visual, memudahkan mahasiswa memahami alur progres dari dasar ke lanjutan.



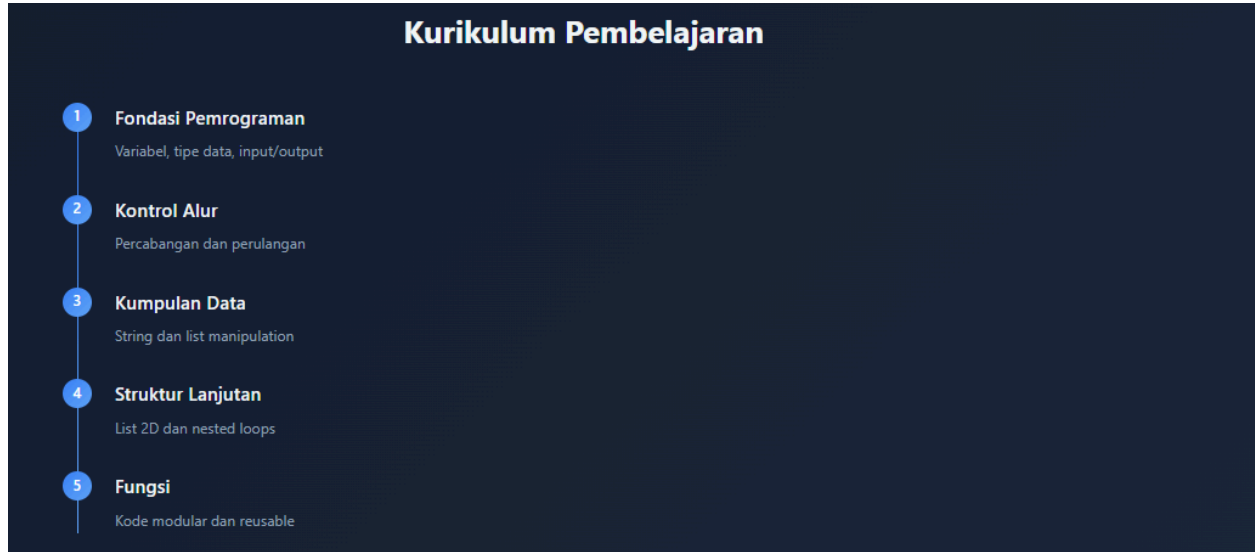
**Gambar 3.2.2.2 Beranda**



**Gambar 3.2.2.3 Beranda**



**Gambar 3.2.2.4 Beranda**



**Gambar 3.2.2.5 Beranda**

### 3. Materi Pembelajaran

- Disusun menjadi 5 bab, dari dasar hingga fungsi.
- Tiap bab dilengkapi tujuan pembelajaran, materi, dan video pendukung (YouTube API).





**Gambar 3.2.3.1 Bab 1**

BAB 2

Mengontrol Alur Program

Tonton Video

**Tujuan Pembelajaran**

- **Konseptual:** Mahasiswa memahami bagaimana alur eksekusi program dapat dikontrol agar tidak berjalan lurus saja, melainkan dapat bercabang atau berulang.
- **Praktis:** Mahasiswa mampu menerapkan percabangan dan perulangan untuk menyelesaikan permasalahan logika sederhana.

**Materi Perkuliahan**

1. Percabangan (if-elif-else)
  - **Alur keputusan:** Program sering kali harus memilih jalur berdasarkan kondisi tertentu. Misalnya, jika nilai ujian  $\geq 85$ , maka hasilnya "A".
  - **Indentasi:** Python menggunakan indentasi (spasi menjorok ke dalam) untuk menandai blok kode. Kesalahan indentasi akan membuat program tidak dapat dijalankan.

**Gambar 3.2.3.2 Bab 2**

BAB 3

Bekerja dengan Kumpulan Data

Tonton Video

**Tujuan Pembelajaran**

- **Konseptual:** Mahasiswa memahami perbedaan antara tipe data yang bisa diubah (mutable) dan yang tidak bisa diubah (immutable).
- **Praktis:** Mahasiswa mampu menyimpan, mengakses, dan memanipulasi data dalam jumlah banyak menggunakan string dan list.

**Materi Perkuliahan**

1. String sebagai Sekuens Data
  - String adalah kumpulan karakter yang bersifat immutable. Artinya, setelah string dibuat, isinya tidak dapat diubah.
  - Operasi seperti mengubah huruf menjadi kapital atau menghapus spasi hanya menghasilkan string baru, bukan memodifikasi string asli.
  - String memiliki banyak metode bawaan, seperti mengubah ke huruf besar, menghapus spasi di awal/akhir, mengganti kata tertentu, atau memisahkan string menjadi daftar kata.

**Gambar 3.2.3.3 Bab 3**

BAB 4

Struktur Data Lanjutan dan Pola Perulangan

Tonton Video

**Tujuan Pembelajaran**

- **Konseptual:** Mahasiswa memahami bagaimana data dua dimensi dapat direpresentasikan dalam bentuk baris dan kolom.
- **Praktis:** Mahasiswa mampu membuat dan mengolah list 2D dengan menggunakan perulangan bersarang.

**Materi Perkuliahan**

1. Perulangan Bersarang
  - Perulangan bersarang adalah perulangan di dalam perulangan.
  - Analogi seperti jam, di mana jarum panjang (menit) bergerak penuh setiap kali jarum pendek (jam) bergeser satu langkah. Dengan kata lain, loop dalam harus menyelesaikan seluruh iterasinya untuk setiap satu iterasi loop luar.
  - Nested loop umum digunakan untuk memproses data berbentuk tabel atau grid.

**Gambar 3.2.3.4 Bab 4**

BAB 5

Membuat Kode Modular dengan Fungsi

Tonton Video

### Tujuan Pembelajaran

- **Konseptual:** Mahasiswa memahami konsep abstraksi dan dekomposisi masalah dengan membagi program besar menjadi bagian-bagian kecil.
- **Praktis:** Mahasiswa mampu membuat fungsi yang rapi, efisien, serta dapat digunakan kembali.

### Materi Perkuliahan

#### 1. Abstraksi dan Dekomposisi

- Dekomposisi: Memecah masalah besar menjadi sub-masalah kecil yang lebih mudah dikelola.
- Abstraksi: Menyembunyikan detail implementasi dan hanya menampilkan cara penggunaan. Misalnya, saat menggunakan fungsi bawaan seperti print(), kita tidak perlu tahu cara kerjanya, cukup tahu hasil yang diberikan.
- Fungsi dapat diberi dokumentasi (docstring) untuk menjelaskan tujuan, parameter, dan hasil keluaran.

**Gambar 3.2.3.5 Bab 5**

#### 4. Editor Kode Simulasi

- Area *textarea* untuk menulis kode Python.
- Sistem simulasi (simulatePythonExecution) memberikan hasil sesuai pola kode.
- Ada fitur Jalankan Kode, Reset, dan skor otomatis.

### Soal 1.1: Pendapatan Penjual Sayur

Seorang pedagang sayur menjual 3 jenis sayur: kangkung, wortel, dan kol. Masukkan jumlah keranjang untuk tiap jenis dan harga per keranjang. Hitung total pendapatan kotor, potong biaya transportasi tetap, dan tampilkan pendapatan bersih.

Python Editor	Bab 1	Output Console	Hasil Eksekusi
<pre># Soal 1.1 - Pendapatan Penjual Sayur kangkung = int(input('Keranjang kangkung: ')) wortel = int(input('Keranjang wortel: ')) kol = int(input('Keranjang kol: ')) harga = float(input('Harga per keranjang (Rp): ')) # Lengkapi program...</pre>		<p>Tekan tombol "Jalankan Kode" untuk melihat hasil...</p>	

Jalankan Kode

Reset

Skor: 0/100

### **Gambar 3.2.4 Editor Kode Simulasi**

#### **5. Latihan Soal Terintegrasi**

- Soal berbasis kasus nyata (belanja, kalkulator, matriks, dll).
- Soal dibagi per bab.
- Skor otomatis: hijau ( $\geq 70$ ), kuning (40–69), merah ( $< 40$ ).

**Soal 5.1: Fungsi Hitung Total Belanja**

Buat fungsi `hitung_total(subtotal)` yang menambahkan PPN 11% dan diskon 5% bila subtotal  $\geq 500000$ . Program utama meminta subtotal, memanggil fungsi, dan menampilkan total.

Python Editor	Bab 5	Output Console	Hasil Eksekusi
<pre>def hitung_total(subtotal):     """Hitung total dengan PPN 11% dan diskon 5% jika subtotal &gt;= 500000."""     ppn = 0.11 * subtotal     total = subtotal + ppn     if subtotal &gt;= 500000:         total *= 0.95     return total subtotal = float(input("Subtotal: Rp ")) print(f"Total bayar: Rp {hitung_total(subtotal):,.2f}")</pre>		<p>Simulasi Soal 5.1:</p> <p>Subtotal: Rp 600000              Total bayar: Rp 632,700.00              (Dengan diskon 5% karena <math>\geq 500000</math>)</p> <p>Simulasi tanpa diskon:</p> <p>Subtotal: Rp 200000              Total bayar: Rp 222,000.00</p>	

Jalankan Kode
Reset
Skor: 100/100

**Gambar 3.2.5.1 Latihan Soal Terintegrasi**

**Pilih Bab Untuk Berlatih**

Setiap bab memiliki 5 soal latihan yang dapat Anda kerjakan dengan editor kode interaktif

- 01 Fondasi Pemrograman**  
 Input, tipe data, operasi dasar  
5 SOAL TERSEDIA
- 02 Kontrol Alur Program**  
 Percabangan dan perulangan  
5 SOAL TERSEDIA
- 03 Kumpulan Data**  
 String dan list manipulation  
5 SOAL TERSEDIA
- 04 Struktur Data Lanjutan**  
 List 2D dan nested loops  
5 SOAL TERSEDIA
- 05 Fungsi**  
 Kode modular dan reusable

**Gambar 3.2.5.2 Latihan Soal Terintegrasi**

## 6. Aktivitas Tambahan

- Drag&Drop: untuk memahami urutan logika.

## Latihan Drag & Drop - Menyusun Fungsi Hitung Rata-rata

Susun blok kode berikut menjadi program yang benar dengan cara drag & drop!

### Blok Kode Tersedia

```
print(f'Rata-rata: {hitung_rata(nilai_siswa)}')
```

```
return total / len(nilai)
```

```
total = sum(nilai)
```

```
def hitung_rata(nilai):
```

```
    nilai_siswa = [80, 90, 75, 85]
```

### Area Solusi (Drag di sini)

```
print(f'Rata-rata: {hitung_rata(nilai_siswa)}')
```

```
return total / len(nilai)
```

```
total = sum(nilai)
```

Periksa Jawaban
Reset

**Gambar 3.2.6.1 Drag&Drop**

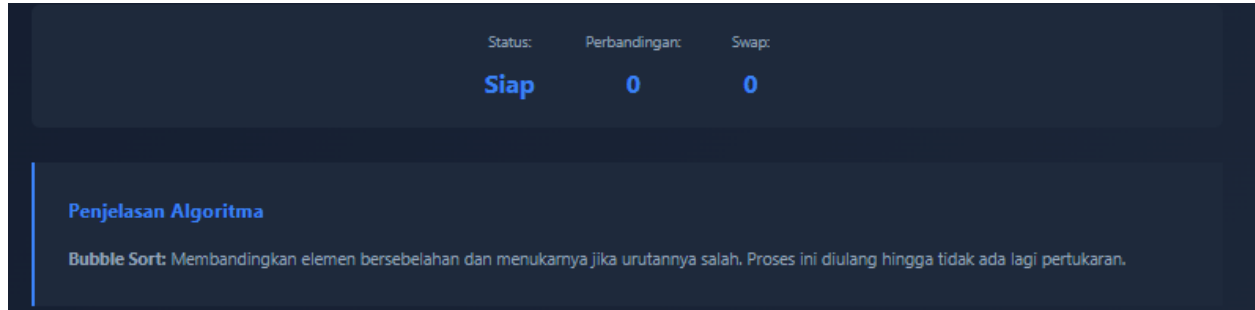
- Visualisasi Canvas: untuk menampilkan data atau algoritma secara grafis.

## Visualisasi Algoritma Sorting

Lihat bagaimana algoritma sorting bekerja secara visual! Pilih algoritma dan tekan "Mulai Sorting" untuk melihat animasinya.

Generate Array Baru
Bubble Sort ▼
Mulai Sorting
Reset

### Gambar 3.2.6.2.2 Visualisasi Canvas



### Gambar 3.2.6.2.2 Visualisasi Canvas

## 3.3 Kurikulum

- Bab 1: Fondasi Pemrograman (variabel, tipe data, I/O).
- Bab 2: Kontrol Alur (percabangan, perulangan).
- Bab 3: Kumpulan Data (string & list).
- Bab 4: Struktur Lanjutan (list 2D, nested loop).
- Bab 5: Fungsi (modularisasi kode).

---

## BAB IV

### IMPLEMENTASI

#### 4.1 Teknologi yang Digunakan

- HTML5 → struktur halaman.
- CSS3 (modern, tema gelap, responsif).
- JavaScript Vanilla → logika interaktif (tanpa framework).

#### 4.2 Desain Antarmuka

- Tema gelap dengan aksen biru (variabel CSS).
- Responsif untuk desktop dan mobile.
- Komponen utama: navbar, hero section, daftar bab, editor kode, console output.

### 4.3 Simulasi Eksekusi Kode

Karena tidak ada interpreter Python asli di browser, digunakan metode:

- Sistem mencocokkan kata kunci dalam kode (misal input, print, if, for).
- Jika pola sesuai, sistem menampilkan output simulasi predefined.
- Contoh: Soal 1.2 (konversi suhu) → jika kode mengandung 9/5 dan 273.15, maka output simulasi ditampilkan.

### 4.4 Mekanisme Skoring

- Fungsi calculateScore() (di script.js) memberi skor 0–100.
- Indikator skor ditampilkan dengan warna:
  - Hijau = Excellent ( $\geq 70$ ).
  - Abu-abu = Perlu diperbaiki (40–69).
  - Merah = Salah ( $< 40$ ).

### 4.5 Contoh Soal

- Soal 1.5 (Rata-rata Nilai) → input 3 angka, hitung rata-rata, tampilkan Lulus/Tidak.
  - Soal 2.3 (Game Tebak Angka) → random number, 7 kesempatan tebak.
  - Soal 3.2 (Palindrom) → cek string apakah palindrom.
  - Soal 4.2 (Tic-Tac-Toe) → representasi papan 3x3 dengan nested list.
  - Soal 5.1 (Fungsi Total Belanja) → fungsi dengan diskon & PPN.
-

## **BAB V**

### **HASIL DAN PEMBAHASAN**

#### **5.1 Keunggulan**

- Interaktif: langsung mencoba kode.
- Akses mudah: hanya butuh browser.
- Feedback instan: simulasi output & skor.
- Materi lengkap: 5 bab sesuai kurikulum.
- Desain modern: UI nyaman & responsif.

#### **5.2 Keterbatasan**

- Simulasi Python masih sederhana (berbasis pola, bukan interpreter asli).
- Latihan soal terbatas pada contoh predefined.
- Belum ada backend untuk menyimpan progres mahasiswa.

#### **5.3 Rencana Pengembangan**

- Integrasi interpreter Python asli (mis. *Skulpt* atau *Pyodide*).
  - Sistem login agar progres mahasiswa tersimpan.
  - Menambah soal otomatis adaptif.
  - Menambahkan fitur leaderboard / gamifikasi.
- 

## **BAB VI**

### **RENCANA PENGEMBANGAN**



Agar Virtual Lab semakin bermanfaat dan dapat digunakan dalam jangka panjang, berikut rencana pengembangan ke depan:

### **6.1 Integrasi Interpreter Python Asli**

- Menggunakan library berbasis browser seperti Skulpt atau Pyodide agar kode Python benar-benar dapat dieksekusi langsung, bukan hanya simulasi.
- Hal ini akan meningkatkan realisme latihan dan mengurangi keterbatasan validasi berbasis pola.

### **6.2 Sistem Akun & Penyimpanan Progres**

- Menambahkan fitur login mahasiswa/dosen agar progres belajar (materi selesai, skor latihan, dll.) dapat tersimpan di database.
- Dosen bisa memonitor perkembangan mahasiswa.
- Integrasi dengan SIAK/learning management system (LMS) ITB di masa depan.

### **6.3 Bank Soal Adaptif**

- Soal latihan akan diperluas dengan bank soal yang dapat ditambah secara dinamis.
- Sistem adaptif: kesulitan soal akan menyesuaikan performa mahasiswa.
- Mendukung berbagai tipe soal: coding, pilihan ganda, drag-and-drop logika, debugging.

### **6.4 Gamifikasi & Leaderboard**

- Menambahkan fitur poin, badge, dan leaderboard untuk memotivasi mahasiswa.
- Sistem reward seperti “Completed Chapter”, “100% Score”, dll.
- Gamifikasi akan meningkatkan engagement.

### **6.5 Kolaborasi & Diskusi**

- Fitur ruang diskusi dalam platform agar mahasiswa bisa saling bertanya.
- Integrasi dengan forum atau chatbot AI sebagai asisten belajar.

## 6.6 Dukungan Materi Lanjutan

- Selain dasar Python, materi dapat diperluas ke:
  - Struktur Data & Algoritma
  - Data Science dasar (NumPy, Pandas, Matplotlib)
  - Pemrograman Berorientasi Objek (OOP)
  - Proyek mini berbasis real-life problem

## 6.7 Deploy Skala Lebih Besar

- Optimalisasi hosting (Vercel/Netlify).
  - Jika digunakan banyak pengguna, bisa dikembangkan backend berbasis Node.js + database (MongoDB/PostgreSQL).
  - Dukungan offline (Progressive Web App / PWA).
- 

# BAB VII

## KESIMPULAN

Virtual Lab saat ini sudah berfungsi sebagai platform pembelajaran interaktif dengan materi terstruktur, editor kode simulasi, latihan soal, dan aktivitas tambahan. Namun, masih ada ruang pengembangan besar agar platform ini bisa menjadi sistem pembelajaran komprehensif, real-time, dan adaptif.

Dengan roadmap pengembangan (integrasi interpreter asli, sistem akun, bank soal adaptif, gamifikasi, hingga dukungan materi lanjutan), Virtual Lab berpotensi menjadi ekosistem pembelajaran digital yang modern dan skalabel, bukan hanya untuk TPB ITB, tetapi juga bisa diperluas ke universitas lain maupun pembelajar umum.