

## DAFTAR PUSTAKA

- A. Yudi Permana, & Puji Romadlon. (2019). *PERANCANGAN SISTEM INFORMASI PENJUALAN PERUMAHAN MENGGUNAKAN METODE SDLC PADA PT. MANDIRI LAND PROSPEROUS BERBASIS MOBILE.*
- Bentley, Lonnie D, Whitten, & Jeffrey L. (2007). *Systems analysis and design for the global enterprise.*
- Brin, S. L. (1998). *Computer Networks and ISDN Systems* (Vol. 30).
- Clay B, & Esparza, S. (2012). *SEARCH ENGINE OPTIMIZATION ALL-IN-ONE FOR DUMMIES JOHN WILEY & SONS.*
- Faqih, A. H., Laksana, T. G., Febriati, A., Teknologi, I., Purwokerto, T., Panjaitan, J. D. I., 128, N., Kidul, P., Sel, P., & Tengah, J. (2018). *SISTEM INFORMASI REPORTING CURICULUM VITAE KARYAWAN MENGGUNAKAN METODE RAPID APPLICATION DEVELOPMENT BERBASIS WEBSITE DI PT. PINS INDONESIA.*
- Ge, D., & Ding, Z. (2016). *Robots Exclusion and Guidance Protocol* (Vol. 21, Issue 6).
- Hidayat, A., Yani, A., Studi Sistem Informasi, P., & Mahakarya, S. (2019). *MEMBANGUN WEBSITE SMA PGRI GUNUNG RAYA RANAU MENGGUNAKAN PHP DAN MYSQL* (Vol. 2, Issue 2).
- Kendall, Kenneth E, & E, J. (2003). *System analysis and design fifth edition.*
- Kusumaningati S. W. (2014). *PENGEMBANGAN SISTEM INFORMASI KARTU MENUJU SEHAT SEBAGAI ALTERNATIF PENGELOLAAN POSYANDU SECARA DIGITAL.*
- M. Jabir Al Haiyan. (2017). *PERANCANGAN DAN PEMBUATAN CRAWLER UNTUK PENGUMPULAN BANK SOAL MATEMATIKA SEKOLAH DASAR DENGAN FRAMEWORK CODEIGNITER, LIBRARY PHPCRAWL, DAN SIMPLE HTML DOM.*
- Mitchell, R. (Ryan E. ). (2018). *Web scraping with Python : collecting more data from the modern web.*
- Mochamad Fiqri. (2020). *PEMBANGUNAN APLIKASI MANAJEMEN EVENT (STUDI KASUS WAH WAH EVENT).*
- Nafi'iyah, N., & Sulistiono, E. (2016). *PEMANFAATAN ROBOT CRAWLER PADA PEMBUATAN TOKO BUKU ONLINE.*
- Purnomo, A. (2022). *Impementasi Web Scraping Pada OJS Dengan Metode CSS Selector. Media Online).*

- Rehni Jayana Purba. (2021). *SISTEM INFORMASI BERBASIS WEB PADA TOKO BUKU DI KOTA BATAM*.
- Riziq sirfatullah Alfarizi, M., Zidan Al-farish, M., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). PENGGUNAAN PYTHON SEBAGAI BAHASA PEMROGRAMAN UNTUK MACHINE LEARNING DAN DEEP LEARNING. In *Karimah Tauhid* (Vol. 2, Issue 1).
- Siti Humayroh. (2022). *Maraknya Penjualan buku Bajakan di E-Commerce*.  
<https://kumparan.com/siti-humayroh-1670211402564137241/maraknya-penjualan-buku-bajakan-di-e-commerce-1zOxx9LBqD9>
- Suharyanto, E. (2022). PERANCANGAN APLIKASI PENGENALAN BUDAYA NUSANTARA BERBASIS ANDROID DENGAN METODE RAD. *Jurnal Ilmu Komputer JIK*, 2022.
- Tuti Susilawati, Fanny Yuliansyah, Muhammad Romzi, & Rinta Aryani. (2020). *MEMBANGUN WEBSITE TOKO ONLINE PEMPEK NTHREE MENGGUNAKAN PHP DAN MYSQL*.

# LAMPIRAN

## 1. *Crawling Deepublishstore (deep.py)*

```
deep.py > ...
1 import requests
2 from bs4 import BeautifulSoup
3 import csv
4 import pandas as pd
5 import time
6 import mysql.connector
7
8 base_url = 'https://deepublishstore.com/shop/'
9 page_number = 1
10
11 crawl = []
12
13 while True:
14     url = f'{base_url}page/{page_number}'
15
16     response = requests.get(url)
17
18     if response.status_code == 200:
19         # melakukan ekstraksi data yang Anda butuhkan dari halaman ini.
20         soup = BeautifulSoup(response.text, 'html.parser')
21
22         for item in soup.findAll("li"):
23             # nama
24             nama_books = item.find(
25                 "h2", class_="woocommerce-loop-product__title")
26
27             # harga
28             prices = item.find("span", "woocommerce-Price-amount amount")
29
30             # link
31             links = item.find(
32                 "a", "woocommerce-LoopProduct-link woocommerce-loop-product__link")
33
34             # gambar
35             gambar = item.find(
36                 "img", "attachment-woocommerce_thumbnail size-woocommerce_thumbnail")
37
38             # pemisahan dari tag html
39             nama_buku = nama_books.text.strip() if nama_books else None
40             price = prices.text.strip() if prices else None
41             link_buku = links['href'] if links and 'href' in links.attrs else None
42             gambar_buku = gambar['src'] if gambar and 'src' in gambar.attrs else None
43
44             # kondisi dimana ketika data yang diambil menghasilkan nilai none maka tidak akan dimasukan kedalam list
45             if None not in [nama_buku, price, link_buku, gambar_buku]:
46                 crawl.append([nama_buku, price, link_buku, gambar_buku])
```

Lampiran 1. *Crawling Deepublishstore*

```

44     # kondisi dimana ketika data yang diambil menghasilkan nilai none maka tidak akan dimasukan kedalam list
45     if None not in [nama_buku, price, link_buku, gambar_buku]:
46         crawl.append([nama_buku, price, link_buku, gambar_buku])
47
48     # sistem akan mengecek apakah ada page selanjutnya agar bisa mengambil informasi dari page berikutnya
49     next_button = soup.find('a', class_='next page-numbers')
50
51     if next_button:
52         page_number += 1
53     else:
54         print("halaman habis")
55         break # Keluar dari perulangan jika tidak ada halaman berikutnya
56
57     else:
58         # jika gagal dan response status di atas 200an
59         print(f'Gagal mengunduh halaman web {url}', "URL tidak ditemukan")
60         break
61
62
63     print(response)
64
65     # Membuat koneksi ke database
66     db_connection = mysql.connector.connect(
67         host="localhost",
68         user="root",
69         password=""
70     )
71     cursor = db_connection.cursor()
72
73     # Mengecek apakah database multibookstore sudah ada
74     cursor.execute("SHOW DATABASES LIKE 'multibookstore'")
75     database_exists = cursor.fetchone()
76
77     if not database_exists:
78         # Jika database belum ada, buat database multibookstore
79         cursor.execute("CREATE DATABASE multibookstore")
80         print("Database multibookstore berhasil dibuat.")
81     else:
82         print("Database multibookstore sudah tersedia.")
83
84     # Menggunakan database multibookstore
85     cursor.execute("USE multibookstore")
86
87     # Mengecek apakah tabel deepublishstore12 sudah ada
88     cursor.execute("SHOW TABLES LIKE 'deepublishstore12'")
89     table_exists = cursor.fetchone()

```

```

86
87     # Mengecek apakah tabel deepublishstore12 sudah ada
88     cursor.execute("SHOW TABLES LIKE 'deepublishstore12'")
89     table_exists = cursor.fetchone()
90
91     if not table_exists:
92         # Jika tabel belum ada, buat tabel deepublishstore12
93         create_table_query = """
94         CREATE TABLE deepublishstore12 (
95             id INT AUTO_INCREMENT PRIMARY KEY,
96             namabuku VARCHAR(255),
97             harga VARCHAR(50),
98             linkbuku VARCHAR(255),
99             gambarbuku VARCHAR(255)
100        )
101        """
102        cursor.execute(create_table_query)
103        print("Tabel deepublishstore12 berhasil dibuat.")
104    else:
105        print("Tabel deepublishstore12 sudah tersedia.")
106
107    # Menutup kursor dan koneksi
108    cursor.close()
109    db_connection.close()
110
111
112    db_connection = mysql.connector.connect(
113        host="localhost",
114        user="root",
115        password="",
116        database="multibookstore"
117    )
118
119    if db_connection.is_connected():
120        print("koneksi DB berhasil")
121
122    # Membuat kursor
123    cursor = db_connection.cursor()
124
125    # Memasukkan data ke dalam tabel
126    for data in crawl:
127        data[1] = data[1].replace("Rp", "").strip()
128        sql_query = "INSERT INTO deepublishstore12 (id, namabuku, harga, linkbuku, gambarbuku) VALUES (CONCAT('dp', LAST_INSERT_ID()+1),%s, %s, %s, %s)"
129        cursor.execute(sql_query, data)
130
131    # Commit perubahan ke database

```

```
deep.py > ...
111
112 db_connection = mysql.connector.connect(
113     host="localhost",
114     user="root",
115     password="",
116     database="multibookstore"
117 )
118
119 if db_connection.is_connected():
120     print("koneksi DB berhasil")
121
122 # Membuat kursor
123 cursor = db_connection.cursor()
124
125 # Memasukkan data ke dalam tabel
126 for data in crawl:
127     data[1] = data[1].replace("Rp", "").strip()
128     sql_query = "INSERT INTO deepublishstore12 (id, namabuku, harga, linkbuku, gambarbuku) VALUES (CONCAT('dp', LAST_INSERT_ID()+1),%s, %s, %s, %s)"
129     cursor.execute(sql_query, data)
130
131 # Commit perubahan ke database
132 db_connection.commit()
133
134 # Menutup kursor dan koneksi
135 cursor.close()
136 db_connection.close()
137 print("berhasil memasukan data kedalam database")
138
139 else:
140     print("ada yang eror db tidak terhubung!!!")
141
142
143 df = pd.DataFrame(
144     crawl, columns=["nama_buku", "harga", "link buku", "gambar buku"])
145
146 time.sleep(5) # program akan berjalan selama 30 detik
147
148 # print(df)
149
150 # df.to_csv("ayenbuku2.csv", index=False)
151
```

Ln 99, Col 32 Spaces: 4 UTF-8 CRLF MagicPython 3.8.1 64-bit Go Live

## 2. Crawling Grobmart (grob.py)

```
1 import requests
2 from bs4 import BeautifulSoup
3 import csv
4 import pandas as pd
5 import mysql.connector
6 import time
7
8
9 base_url = 'https://www.grobmart.com/Buku'
10 page_number = 1
11
12 crawl = []
13 print("Memulai crawling...")
14 while True:
15     url = f'{base_url}&page={page_number}'
16
17     response = requests.get(url)
18
19     if response.status_code == 200:
20         # melakukan ekstraksi data yang dibutuhkan dari halaman.
21         soup = BeautifulSoup(response.text, 'html.parser')
22
23         # tag yang akan dicrawl
24         for item in soup.findAll("div", "product-thumb,"):
25             # judul
26             nama_books = item.find("div", "name")
27             nama_buku = nama_books.text.strip() if nama_books else None
28
29             # harga
30             prices = item.find("span", "price-new")
31             price = prices.text.strip() if prices else None
32
33             # link
34             links = item.find('div', class_='name')
35             link_buku = None
36             if links:
37                 a_tag = links.find('a')
38                 if a_tag and 'href' in a_tag.attrs:
39                     link_buku = a_tag['href']
40
41             # image
42             img = item.find('a', class_='product-img')
43             gambar_buku = None
44
45             if img:
46                 b_tag = img.find('img')
```

Lampiran 2. *Crawling Grobmart*

```

46         b_tag = img.find('img')
47         if b_tag and 'data-src' in b_tag.attrs:
48             gambar_buku = b_tag['data-src']
49
50         # menambahkan kedalam list
51         if None not in [nama_buku, price, link_buku, gambar_buku]:
52             crawl.append([nama_buku, price, link_buku, gambar_buku])
53
54         # pengecekan button untuk next page
55         next_button = soup.find('a', class_='next')
56
57         if next_button:
58             page_number += 1
59
60         else:
61             print(response)
62             print("halaman habis")
63             break # Keluar dari perulangan jika tidak ada halaman berikutnya
64
65     else:
66         # jika gagal dan response status di atas 200an
67         print(f'Gagal mengunduh halaman web {url}',
68             "URL tersebut tidak ditemukan")
69
70
71 # Membuat koneksi ke database
72 db_connection = mysql.connector.connect(
73     host="localhost",
74     user="root",
75     password=""
76 )
77 cursor = db_connection.cursor()
78
79 # Mengecek apakah database multibookstore sudah ada
80 cursor.execute("SHOW DATABASES LIKE 'multibookstore'")
81 database_exists = cursor.fetchone()
82
83 if not database_exists:
84     # Jika database belum ada, buat database multibookstore
85     cursor.execute("CREATE DATABASE multibookstore")
86     print("Database multibookstore berhasil dibuat.")
87 else:
88     print("Database multibookstore sudah tersedia.")
89
90 # Menggunakan database multibookstore
91 cursor.execute("USE multibookstore")

```

```

Y.PY > ...
91 cursor.execute("USE multibookstore")
92
93 # Mengecek apakah tabel grobmart sudah ada
94 cursor.execute("SHOW TABLES LIKE 'grobmart'")
95 table_exists = cursor.fetchone()
96
97 if not table_exists:
98     # Jika tabel belum ada, buat tabel grobmart
99     create_table_query = """
100     CREATE TABLE grobmart (
101         id INT AUTO_INCREMENT PRIMARY KEY,
102         namabuku VARCHAR(255),
103         harga VARCHAR(50),
104         linkbuku VARCHAR(255),
105         gambarbuku VARCHAR(255)
106     )
107     """
108     cursor.execute(create_table_query)
109     print("Tabel grobmart berhasil dibuat.")
110 else:
111     print("Tabel grobmart sudah tersedia.")
112
113 # Menutup kursor dan koneksi
114 cursor.close()
115 db_connection.close()
116
117 # Membuat koneksi ke database
118 db_connection = mysql.connector.connect(
119     host="localhost",
120     user="root",
121     password="",
122     database="multibookstore"
123 )
124 cursor = db_connection.cursor()
125
126 if db_connection.is_connected():
127     print("koneksi DB berhasil")
128
129 # Membuat kursor
130 cursor = db_connection.cursor()
131 # Memasukkan data ke dalam tabel
132 for data in crawl:
133     data[1] = data[1].replace("Rp", "").strip()
134     insert_query = "INSERT INTO grobmart (namabuku, harga, linkbuku, gambarbuku) VALUES (%s, %s, %s, %s)"
135     cursor.execute(insert_query, data)
136

```

```

Y.PY > ...
131 # Memasukkan data ke dalam tabel
132 for data in crawl:
133     data[1] = data[1].replace("Rp", "").strip()
134     insert_query = "INSERT INTO grobmart (namabuku, harga, linkbuku, gambarbuku) VALUES (%s, %s, %s, %s)"
135     cursor.execute(insert_query, data)
136
137 # Commit perubahan ke database
138 db_connection.commit()
139
140 # Menutup kursor dan koneksi
141 cursor.close()
142 db_connection.close()
143 print("berhasil memasukan data kedalam database")
144
145 else:
146     print("ada yang eror db tidak terhubung!!!")
147

```

```

46         b_tag = img.find('img')
47         if b_tag and 'data-src' in b_tag.attrs:
48             gambar_buku = b_tag['data-src']
49
50         # menambahkan kedalam list
51         if None not in [nama_buku, price, link_buku, gambar_buku]:
52             crawl.append([nama_buku, price, link_buku, gambar_buku])
53
54         # pengecekan button untuk next page
55         next_button = soup.find('a', class_='next')
56
57         if next_button:
58             page_number += 1
59
60         else:
61             print(response)
62             print("halaman habis")
63             break # Keluar dari perulangan jika tidak ada halaman berikutnya
64
65     else:
66         # jika gagal dan response status di atas 200an
67         print(f'Gagal mengunduh halaman web {url}',
68             "URL tersebut tidak ditemukan")
69
70
71 # Membuat koneksi ke database
72 db_connection = mysql.connector.connect(
73     host="localhost",
74     user="root",
75     password=""
76 )
77 cursor = db_connection.cursor()
78
79 # Mengecek apakah database multibookstore sudah ada
80 cursor.execute("SHOW DATABASES LIKE 'multibookstore'")
81 database_exists = cursor.fetchone()
82
83 if not database_exists:
84     # Jika database belum ada, buat database multibookstore
85     cursor.execute("CREATE DATABASE multibookstore")
86     print("Database multibookstore berhasil dibuat.")
87 else:
88     print("Database multibookstore sudah tersedia.")
89
90 # Menggunakan database multibookstore
91 cursor.execute("USE multibookstore")

```

```

Y.PY > ...
91 cursor.execute("USE multibookstore")
92
93 # Mengecek apakah tabel grobmart sudah ada
94 cursor.execute("SHOW TABLES LIKE 'grobmart'")
95 table_exists = cursor.fetchone()
96
97 if not table_exists:
98     # Jika tabel belum ada, buat tabel grobmart
99     create_table_query = """
100     CREATE TABLE grobmart (
101         id INT AUTO_INCREMENT PRIMARY KEY,
102         namabuku VARCHAR(255),
103         harga VARCHAR(50),
104         linkbuku VARCHAR(255),
105         gambarbuku VARCHAR(255)
106     )
107     """
108     cursor.execute(create_table_query)
109     print("Tabel grobmart berhasil dibuat.")
110 else:
111     print("Tabel grobmart sudah tersedia.")
112
113 # Menutup kursor dan koneksi
114 cursor.close()
115 db_connection.close()
116
117 # Membuat koneksi ke database
118 db_connection = mysql.connector.connect(
119     host="localhost",
120     user="root",
121     password="",
122     database="multibookstore"
123 )
124 cursor = db_connection.cursor()
125
126 if db_connection.is_connected():
127     print("koneksi DB berhasil")
128
129 # Membuat kursor
130 cursor = db_connection.cursor()
131 # Memasukkan data ke dalam tabel
132 for data in crawl:
133     data[1] = data[1].replace("Rp", "").strip()
134     insert_query = "INSERT INTO grobmart (namabuku, harga, linkbuku, gambarbuku) VALUES (%s, %s, %s, %s)"
135     cursor.execute(insert_query, data)
136

```

```

Y.PY > ...
131 # Memasukkan data ke dalam tabel
132 for data in crawl:
133     data[1] = data[1].replace("Rp", "").strip()
134     insert_query = "INSERT INTO grobmart (namabuku, harga, linkbuku, gambarbuku) VALUES (%s, %s, %s, %s)"
135     cursor.execute(insert_query, data)
136
137 # Commit perubahan ke database
138 db_connection.commit()
139
140 # Menutup kursor dan koneksi
141 cursor.close()
142 db_connection.close()
143 print("berhasil memasukan data kedalam database")
144
145 else:
146     print("ada yang eror db tidak terhubung!!!")
147

```

### 3. *Index.php*

```
index.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Book Search</title>
8   <link rel="stylesheet" href="tes.css">
9 </head>
10
11 <body>
12   <section>
13     <div class="main">
14       <?php include 'search_form.php'; ?>
15     </div>
16   </section>
17 </body>
18
19 </html>
```

Lampiran 3. *Index.php*

### 4. *Search\_form.php*

```
search_form.php > html > body > div.container > form
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Book Search</title>
8   <link rel="stylesheet" href="style.css">
9 </head>
10
11 <body style="background-color: whitesmoke;">
12   <div class="container"
13     style="display: flex; flex-direction: column; justify-content: center; align-items: center; height: 100vh;">
14     
15     <form action="search_result.php" method="GET"
16       style="width: 500px; text-align: center; border-radius: 30px; padding: 10px;">
17       <input class="form-control me-4 rounded-pill" name="q" value="" type="search"
18         placeholder="Search your book" aria-label="Search"
19         style="width: 500px; border-radius: 40px; padding: 15px;">
20     </form>
21   </div>
22 </body>
```

Lampiran 4. *Search\_form.php*

## 5. Search\_result.php

```
search_result.php > html > body > div.container.fixed-top > nav.bg-body-tertiary > h4
1 <?php
2 // Koneksi ke database
3 $db_host = 'localhost';
4 $db_user = 'root';
5 $db_pass = '';
6 $db_name = 'multibookstore';
7
8 $conn = new mysqli($db_host, $db_user, $db_pass, $db_name);
9
10 if ($conn->connect_error) {
11     die("Koneksi gagal: " . $conn->connect_error);
12 }
13
14 // Ambil query pencarian dari formulir
15 $query = isset($_GET['q']) ? $_GET['q'] : '';
16
17 // Lakukan pencarian di tabel 'deepublishstore' dan 'grobmart' dan urutkan hasil dari harga terendah
18 $sql = "SELECT namabuku, harga, linkbuku, gambarbuku, logo FROM (
19     SELECT namabuku, harga, linkbuku, gambarbuku, logo FROM deepublishstore WHERE namabuku LIKE '%$query%'
20     UNION
21     SELECT namabuku, harga, linkbuku, gambarbuku, logo FROM grobmart WHERE namabuku LIKE '%$query%'
22 ) AS combined_results
23 ORDER BY CAST(REPLACE(harga, 'rb', '')) AS DECIMAL(10,2)) ASC;";
24
25 $result = $conn->query($sql);
26
27 // Hitung jumlah hasil pencarian
28 $total_results = $result->num_rows;
29
30 // Tutup koneksi
31 $conn->close();
32 ?>
33
34 <!doctype html>
35 <html lang="en">
36
37 <head>
38     <meta charset="utf-8">
39     <meta name="viewport" content="width=device-width, initial-scale=1">
40     <title><?php echo $query?></title>
41     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
42         integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhJy6hW+ALEwIH" crossorigin="anonymous">
43 </head>
```

Lampiran 5. Search\_result.php

```

45 <body>
46 <div class="container fixed-top">
47 <nav class="bg-body-tertiary">
48 <form class="d-flex mx-auto" role="search" action="search_result.php" method="GET">
49 <input class="form-control me-4" name="q" value="<?php echo $query?>" type="search"
50 placeholder="Search" aria-label="Search" style="width: 400px;">
51 <button class="btn btn-outline-success" type="submit">Search</button>
52 </form>
53 <h4>
54 Hasil Pencarian untuk: <?php echo htmlspecialchars($query);?>
55 </h4>
56 <p>
57 Ditemukan <?php echo $total_results ?> hasil pencarian
58 <strong><?php echo htmlspecialchars($query); ?></strong>"
59 </p>
60 </nav>
61 </div>
62 <div class="container mt-5">
63 <?php
64 if ($total_results > 0) {
65 // Tampilkan hasil pencarian sebagai card
66 echo "<div class='row'>";
67 while ($data = mysqli_fetch_array($result)) {
68 echo "<div class='col-md-3 mt-3'>";
69 echo "<a href='\" . htmlspecialchars($data['linkbuku']) . \"' class='card-link' style='text-decoration: none;'>";
70 echo "<div class='card' style='width: 18rem;'>";
71 echo "<img src='\" . htmlspecialchars($data['gambarbuku']) . \"' class='card-img-top' alt='cover book' width='250' height='250'>";
72 echo "<div class='card-body'>";
73 echo "<h5 class='card-title'>\" . htmlspecialchars(substr($data['namabuku'], 0, 35)) . \"</h5>";
74 echo "<p class='card-text'> Rp.\" . htmlspecialchars($data['harga']) . \"</p>";
75 echo "<img src='\" . htmlspecialchars($data['logo']) . \"' alt='logo' width='80' height='30'>";
76 echo "</div>";
77 echo "</div>";
78 echo "</a>";
79 echo "</div>";
80 }
81 echo "</div>"; // Tutup row
82 } else {
83 echo "<p>Tidak ada hasil ditemukan untuk pencarian: \" . htmlspecialchars($query) . \"</p>";
84 }
85 ?>
86 </div>
87 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
88 integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdS1K1eN7N6jIeHz" crossorigin="anonymous">
89 </script>

```