**LAPORAN**

**PEMROGRAMAN WEB PRAKTIK**

**Dosen Pengampu:**

**Aditya Dimas Dewanto, S.T., M.T.**

****

**Dibuat oleh:**

|  |  |
| --- | --- |
| **Nama** | **: Muhammad Zaki As Shidiqi** |
| **NPM** | **: 5240411230** |

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**YOGYAKARTA**

**2025**

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc211805952)

[BAB I MENJALANKAN MYSQL DAN INTEGRASI DENGAN FLASK 3](#_Toc211805953)

[1.1 MySQL 3](#_Toc211805955)

[1.2 Integrasi MySQL dengan Flask 5](#_Toc211805956)

[BAB II PENERAPAN SESSION PADA FLASK 9](#_Toc211805957)

[2.1 Sistem Login pada Flask 9](#_Toc211805959)

[2.2 Tugas 13](#_Toc211805960)

[KESIMPULAN 20](#_Toc211805961)

# 

# BAB I

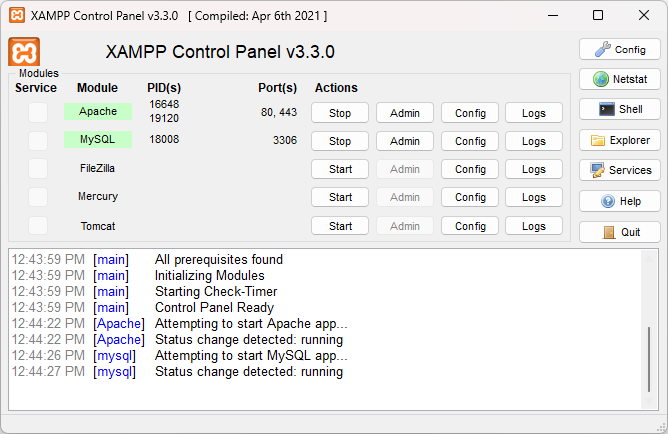
# MENJALANKAN MYSQL DAN INTEGRASI DENGAN FLASK

## MySQL

MySQL adalah sebuah sistem manajemen *database* relasional (RDBMS) *opensource* yang berfungsi untuk menyimpan, mengelola, dan memproses data. Untuk menjalankan MySQL pada komputer, kita membutuhkan satu tools yaitu XAMPP. XAMPP adalah sebua software yang mengintegrasikan server web Apache dan MySQL. Fungsinya untuk menjalankan MySQL di server lokal pada komputer.

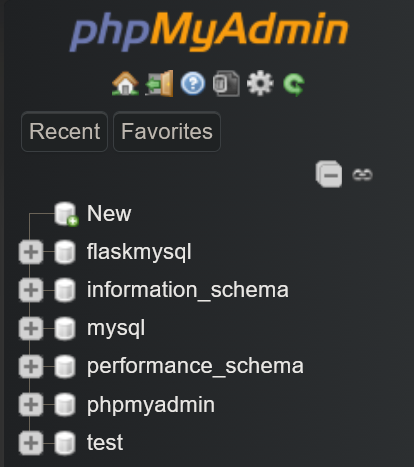
1. Menjalankan MySQL.

Jalankan *software* XAMPP pada komputer. Setelah terbuka, klik tombol start pada modul Apache. Setelah Apache sudah berjalan, kemudian klik tombol start pada modul MySQL. Kemudian klik tombol admin pada modul MySQL untuk membuka panel phpMyAdmin.



1. Membuat database

Setelah phpMyAdmin terbuka, kemudia klik tombol *New* untuk membuat *database* baru. Setelah di klik pada bagian kanan akan muncul menu untuk menambahkan *database*. Karena pada materi kali ini tentang integrasi Flask dengan MySQL nama untuk *database*-nya “flaskmysql”. Setelah memberikan nama klik tombol *create* untuk membuat *database*-nya.

A screenshot of a computer

AI-generated content may be incorrect.

1. Membuat tabel users.

Setelah *database* berhasil dibuat, akan muncul menu untuk menambahkan tabel pada *database*. Pada kolom *Table name* masukan “users” karena kita akan membuat tabel yang bernama “users” yang akan berisi banyak data pengguna nantinya. Pada kolom *Number of cloumns* masukan 4 karena tabel ini akan memiliki 4 kolom. Kemudian klik tombol *create* untuk membuat tabel tersebut.

A screenshot of a computer

AI-generated content may be incorrect.

1. Membuat kolom.

A screenshot of a computer

AI-generated content may be incorrect.

1. Mengisi tabel.

A screenshot of a computer

AI-generated content may be incorrect.

## Integrasi MySQL dengan Flask

Flask adalah sebuah micro framework yang memungkinkan kita memungkinkan kita membuat webserver menggunakan python. Prinsip kerjanya dengan me-*return* html pada setiap function yang kita berikan route diatasnya.

1. Perintah pada CLI.

Sebelum menginstal *flask-mysqldb* pastikan sudah membuat environment menggunakan *virtualenv* dan sudah mengaktifkannya. Selanjutnya untuk menginstall *flask-mysqldb* kita dapat mengetikan command berikut pada terminal dan tunggu proses instalasi sampai selesai.

A black background with blue letters

AI-generated content may be incorrect.

1. Membuat sturktur folder.

Jika proses penginstalan sudah selesai, maka kita bisa memulai proses setup proyek kita. Kita bisa memulai dengan membuat struktur folder terlebih dahulu. Berikut adalah struktur folder pada proyek flask ini.

A black screen with white text

AI-generated content may be incorrect.

1. Menghubungkan Flask dengan MySQL

Selanjutnya kita mulai menuliskan kode pada file app.py. Pada awal baris kita harus meng-*import* beberapa *library* yaitu Flask, render\_template, dan MySQL. Untuk membuat aplikasi web menggunakan Flask kita perlu menuliskan sintaks “app = Flask(\_\_name\_\_)”. Kemudian menuliskan sintaks untuk menghubungkan MySQL dengan aplikasi Flask. Untuk mendapatkan data *users*, kita lakukan query ke database. Selain itu, pada *return* dari *route* *home* selain “index.html” ada *users*  yang diambil dari hasil *query* ke database.



1. Membuat layout untuk template atau acuan halaman lainnya.

Setelah selesai menuliskan kode pada file app.py kita lanjutkan menuliskan kode html pada file layout.html yang berada di folder templates. File ini akan menjadi layout utama untuk halaman lainnya. Kode file ini berupa tag paling luar dari html yaitu <html>, <head>, dan <body>. Selain ada ketiga tak tersebut, terdapat 2 block yaitu head dan body, kedua block ini yang nantinya adalah bagian isi dari website kita.



1. Membuat isi untuk halaman yang menampilkan tabel *users*.

Pada halaman ini kita menggunakan tag <table> untuk membuat sebuah tabel, <thead> untuk membuat header dari tabel, <tbody> untuk membuat isi dari tabel, <tr> untuk membuat baris. Untuk menampilkan data dari tabel *users* kita menggunakan *for loop* supaya setiap data *users* akan membuat baris baru pada tabel.

A screen shot of a computer program

AI-generated content may be incorrect.

1. Hasil

A screenshot of a computer

AI-generated content may be incorrect.

# BAB II

# PENERAPAN SESSION PADA FLASK

## Sistem Login pada Flask

Sistem login adalah sebuah sistem yang mengharuskan *client* untuk melakukan validasi data sebelum mengakses halaman atau fitur lain pada sebuah aplikasi. Dengan sistem ini akan membuat aplikasi yang kita buat lebih aman dari *spam request* yang biasanya dilakukan oleh bot.

1. Struktur folder

A screenshot of a computer

AI-generated content may be incorrect.

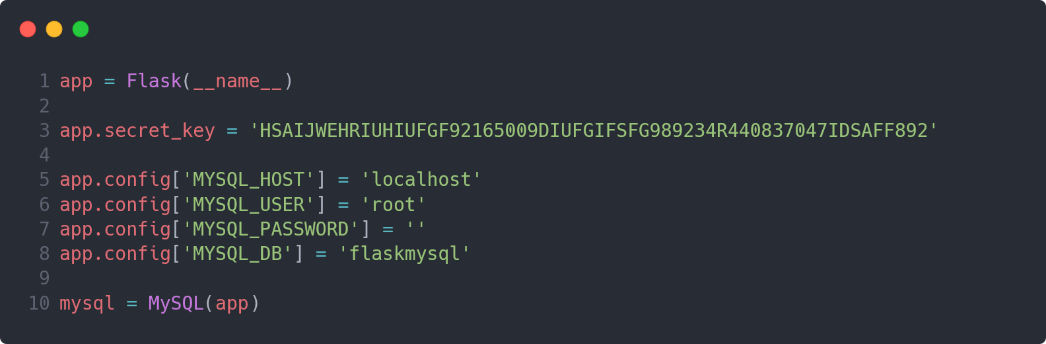
1. Import library yang dibutuhkan

Pada awal baris di file app.py *import* *library* yang dibutuhkan, berbeda dengan sebelumnya untuk membuat sistem login pada aplikasi Flask membutuhkan beberapa tambahan library seperti session, request, redirect, dan url\_for.



1. Membuat *secret key* dan menghubungkan Flask dengan MySQL

Untuk menerapkan *session* pada aplikasi Flask, kita perlu membuat *secret key*. Ini nantinya akan tersimpan pada browser pengguna dan akan hilang dalam rentang tertentu.



1. Route login

Login *route* ini akan terletak pada *path* root, supaya dapat menangkap isi form yang dikirimkan perlu memberikan izin penggunaan *method* “POST”. Untuk menangkap apa yang dikirmkan dari form kita menggunakan *library request* yang sudah kita *import* sebelumnya. Sebelum melakukan query ke *database* kita lakukan validasi apakah form kosong atau tidak. Jika tidak selanjutnya akan menampung isi dari filed email ke dalam variabel email dan *password* ke dalam variabel passwd. Kemudian dilakukan query ke *database* untuk mengambil data pengguna dengan email dan password yang cocok, jika ada maka akan diberikan session dan diarahkan ke halaman home. tetapi jika tidak ada akan di kembalikan lagi ke halaman login.



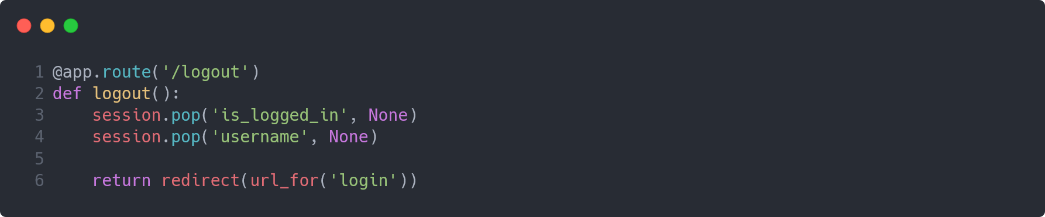
1. Route home

*Home route* ini akan terletak pada *path* “/home”. Ketika pengguna mengakses halaman home, akan dilakukan validasi apakah memiliki session atau tidak. Jika *client* memiliki session maka akan dilakukan query untuk mengambil semua data *user* ke database lalu menampung hasilnya ke variabel data, baru kemudian me-*return* home.html dan variabel users yang mengambil dari data. Jika *client* tidak memiliki *session* maka akan diarahkan ke halaman login.



1. Route logout

*Logout route* ini akan terletak pada *path* “/logout”. Pada *route* ini akan dilakukan penghapusan *session* dan akan mengarahkan *client* ke halaman login.



1. Halaman login

Ini adalah isi dari login.html. Pada halaman ini terdapat form login dengan method “POST” yang memiliki dua field input yaitu email dan password. Selain itu, ada juga tombol untuk submit form.



1. Halaman home

Pada halaman ini kita menggunakan tag <table> untuk membuat sebuah tabel, <thead> untuk membuat header dari tabel, <tbody> untuk membuat isi dari tabel, <tr> untuk membuat baris. Untuk menampilkan data dari tabel *users* kita menggunakan *for loop* supaya setiap data *users* akan membuat baris baru pada tabel.

A screen shot of a computer program

AI-generated content may be incorrect.

1. Hasil

A screenshot of a login box

AI-generated content may be incorrect.

A screenshot of a computer

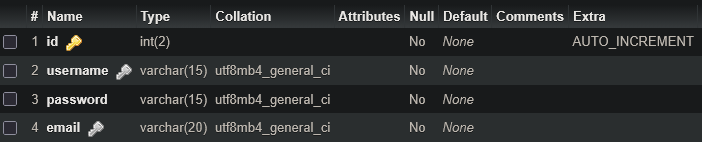
AI-generated content may be incorrect.

## Tugas

Saya tidak lagi menggunakan Bootstrap untuk mengerjakan tugas ini, saya menggunakan TailwindCSS karena lebih *customizable*.

1. Menambahkan rule *auto increment* pada id

Sebelum menambahkan fitur *register* perlu untuk mengubah kolom id menjadi *auto increment* supaya id ter-*generate* otomatis saat ada data baru yang masuk. Selain itu, untuk kolom username dan email diatur *unique* supaya tidak terdapat duplikasi data.



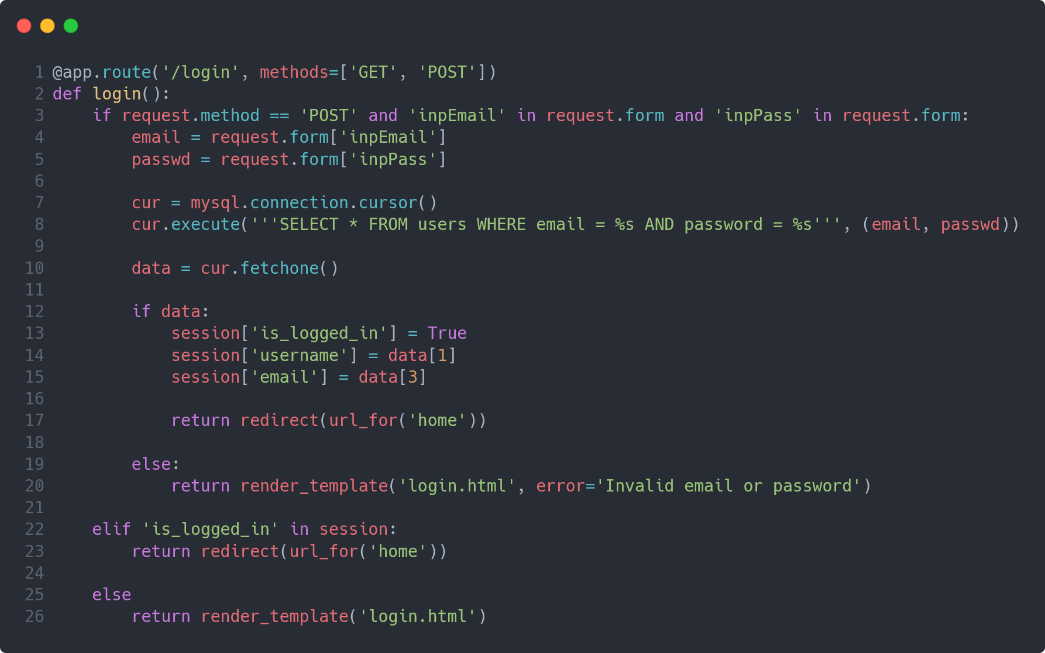
1. Struktur folder

**A screenshot of a computer

AI-generated content may be incorrect.**

1. Route login

Saya mengubah path dari awalnya “/” menjadi “/login” supaya path lebih sesuai dengan isi halaman. Saya juga menambahkan validasi yang berfungsi untuk mengarahkan *client* yang sudah memiliki session, ketika mengakses path /login maka akan di arahkan ke path “/” atau halaman home. Saya juga menambahkan informasi email dan password tidak valid ke *client* jika email dan password yang diinputkan tidak ada pada database. Saya juga menambahkan username dan email ke dalam session supaya bisa menampilkan data *user* yang sedang digunakan.



1. Route root

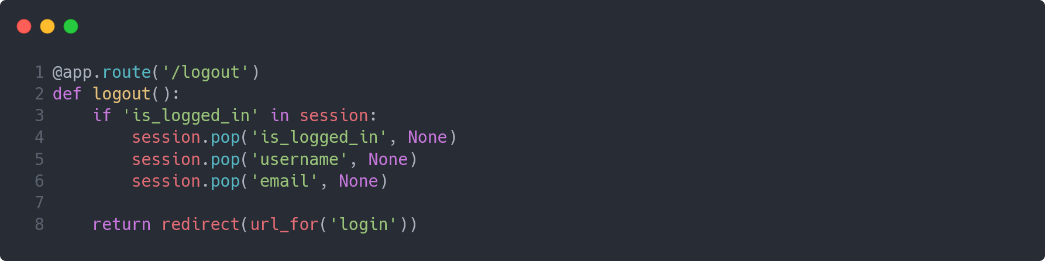
Saya juga mengubah path yang awalnya “/home” menjadi “/”. Saya juga membuat variabel user yang akan memuat username dan email yang didapatkan dari session ke dalam sebuah list.

A screen shot of a computer code

AI-generated content may be incorrect.

1. Route logout

Saya menambahkan validasi supaya dilakukan penghapusan session jika *client* sudah login saja.



1. Route Register

Route ini berfungsi untuk meng-*handle* request client untuk melakukan registrasi pada *path* “/register”. Karena pada route ini mengambil data dari form, maka perlu untuk mengizinkan *method* “POST”. Pada route ini juga terdapat validasi untuk memastikan semua kolom input tidak kosong. Jika semua kolom input sudah di validasi maka semua isi dari input *client* akan ditampung ke masing-masing variabel yaitu username, email, passwd. Kemudian akan dilakukan *query* ke *database* untuk memasukan data ke *database*, jika berhasil maka *client* akan diarahkan ke halaman login. Tidak lupa, di sini juga terdapat validasi supaya mengarahkan client yang sudah login ke path “/” jika mengakses *path* ini.

A screen shot of a computer program

AI-generated content may be incorrect.

1. Halaman login

Saya menabahkan informasi jika email dan password tidak valid, ini akan muncul setelah *client* melakukan submit.

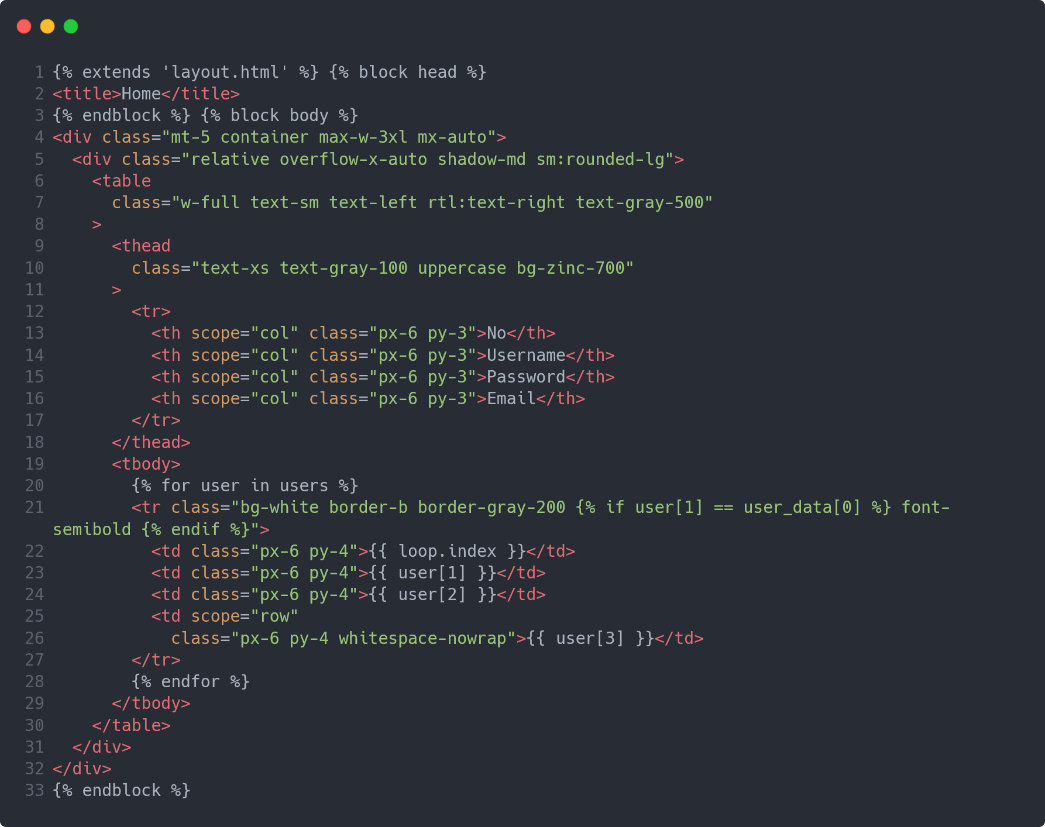


1. Halaman register



1. Halaman home

Saya menambahkan sedikit detail pada halaman home yaitu menampilkan *username* dan email *user* yang sedang login, dan memberikan text tebal pada tabel untuk *user* yang sedang login.



1. Hasil

A screenshot of a login screen

AI-generated content may be incorrect.

A screenshot of a login screen

AI-generated content may be incorrect.

A screen shot of a login form

AI-generated content may be incorrect.

**A screenshot of a computer

AI-generated content may be incorrect.**

# KESIMPULAN

Dengan demikian dapat disiumpulkan bahwa sistem berhasil mengimplementasikan fitur login, register, dan logout menggunakan Flask, MySQL, dan session. Proses autentikasi pengguna dilakukan dengan memanfaatkan mekanisme session untuk menyimpan data pengguna yang sedang aktif. Saat pengguna melakukan login, sistem akan memvalidasi data yang dikirim melalui form menggunakan method “POST”, kemudian memeriksa kecocokan antara email dan password di database. Jika data sesuai, maka session akan dibuat dan pengguna diarahkan ke halaman home, sedangkan jika tidak sesuai, pengguna akan dikembalikan ke halaman login.

Dengan adanya session ini, aplikasi dapat mengenali status pengguna tanpa harus melakukan login berulang kali. Selain itu, penggunaan Flask memudahkan dalam pengelolaan rute, koneksi ke database MySQL, serta pengaturan alur autentikasi. Secara keseluruhan, program ini telah berhasil mewujudkan sistem login yang aman, terstruktur, dan mudah dikembangkan untuk kebutuhan web berbasis Python.

**Repository Gihub:** <https://github.com/muzaaqi/pemrograman-web-praktik/tree/main/meet-4>