



PENGEMBANGAN DAN PENYELENGGARAAN
PEMBELAJARAN DIGITAL (P3D)



Modul Pembelajaran **SISTEM INFORMASI**

Pemrograman Web Framework



Disusun oleh : Wicaksono Yuli Sulistyio



Modul Ajar: Membuat Sistem Login Lengkap Menggunakan CodeIgniter 3

1. Pendahuluan

1.1. Tujuan Pembelajaran

Setelah menyelesaikan modul ini, peserta didik diharapkan mampu:

- Memahami konsep dasar sistem login dan registrasi dalam pengembangan web.
- Menguasai dasar-dasar penggunaan framework CodeIgniter 3 untuk membangun aplikasi web.
- Membangun sistem manajemen pengguna dengan berbagai level akses, seperti admin, user biasa, dan lainnya.
- Mengintegrasikan template Bootstrap untuk membuat antarmuka pengguna yang modern dan responsif.
- Mengelola pengembangan proyek dari awal hingga implementasi dengan struktur yang terorganisir.

Dalam modul ini, peserta didik akan diajarkan tidak hanya cara membuat sistem login tetapi juga bagaimana merancang struktur aplikasi yang baik, mengelola data pengguna, dan membuat aplikasi yang aman serta mudah digunakan.

1.2. Sasaran Pembelajaran

Modul ini dirancang untuk peserta didik yang memiliki:

- **Pemahaman Dasar tentang PHP:** Siswa harus memahami sintaks dasar PHP, termasuk cara mendeklarasikan variabel, membuat fungsi, dan bekerja dengan loop serta kondisi.
- **Pengetahuan tentang OOP (Object-Oriented Programming):** Karena CodeIgniter menggunakan prinsip OOP, penting bagi peserta didik untuk memahami konsep dasar seperti class, object, inheritance, dan encapsulation.
- **Pemahaman Konsep MVC (Model-View-Controller):** MVC adalah pola arsitektur yang digunakan CodeIgniter. Memahami cara kerja MVC akan sangat membantu dalam memahami alur kerja framework ini.
- **Dasar-Dasar HTML, CSS, dan Bootstrap:** Peserta didik harus memiliki pemahaman tentang HTML dan CSS untuk membangun tampilan web. Pengetahuan tentang Bootstrap akan membantu dalam membuat antarmuka yang responsif.
- **Pengetahuan Dasar tentang Database:** Peserta didik harus memahami cara kerja database, termasuk konsep relasional, cara membuat tabel, dan melakukan query dasar seperti SELECT, INSERT, UPDATE, dan DELETE.



1.3. Prasyarat

Sebelum memulai modul ini, peserta didik perlu menyiapkan beberapa alat dan lingkungan pengembangan:

- **Code Editor:** Seperti Visual Studio Code atau Sublime Text, yang akan digunakan untuk menulis dan mengedit kode.
- **Web Server Lokal:** Seperti XAMPP atau WAMP, yang memungkinkan pengujian aplikasi web di lingkungan lokal sebelum diunggah ke server online.
- **Web Browser:** Google Chrome atau Mozilla Firefox untuk menguji dan melihat hasil pengembangan aplikasi.
- **Framework CodeIgniter 3:** Pastikan untuk mengunduh versi terbaru dari CodeIgniter 3 dari situs resminya.
- **Bootstrap:** Mengunduh Bootstrap dari situs resminya atau menggunakan CDN untuk integrasi cepat ke dalam proyek.
- **Template Admin (Opsional):** Untuk manajemen pengguna, menggunakan template admin seperti Nice Admin yang bisa diunduh dari bootstrapmade.com.

2. Materi Pembelajaran

2.1. Pengenalan Framework CodeIgniter 3

2.1.1. Apa itu CodeIgniter?

CodeIgniter adalah framework PHP yang ringan dan cepat yang memudahkan pengembangan aplikasi web dengan menggunakan pola arsitektur Model-View-Controller (MVC). Dengan CodeIgniter, pengembang dapat membuat aplikasi yang terstruktur dengan baik dan mudah dipelihara.

Keunggulan CodeIgniter:

1. **Ukuran Kecil dan Cepat:** CodeIgniter dirancang untuk menjadi framework yang ringan, sehingga dapat memuat dengan cepat dan tidak membebani server.
2. **Dokumentasi Lengkap:** CodeIgniter didukung oleh dokumentasi yang sangat lengkap, yang memudahkan pengembang pemula maupun berpengalaman untuk mempelajari dan mengimplementasikan berbagai fitur yang tersedia.
3. **Kompatibilitas:** CodeIgniter kompatibel dengan berbagai versi PHP dan dapat diintegrasikan dengan berbagai jenis database seperti MySQL, PostgreSQL, dan SQLite.
4. **Keamanan:** CodeIgniter menyediakan fitur bawaan untuk mencegah serangan umum seperti SQL Injection dan Cross-Site Scripting (XSS).

2.1.2. Pola Arsitektur MVC (Model-View-Controller)

MVC adalah pola desain arsitektur yang memisahkan aplikasi menjadi tiga komponen utama:



- **Model:** Bertanggung jawab untuk menangani data aplikasi. Model berinteraksi langsung dengan database dan berisi logika bisnis yang diperlukan untuk mengelola data.
- **View:** Bertanggung jawab untuk menampilkan data kepada pengguna. View adalah representasi antarmuka pengguna dari data yang dikelola oleh model.
- **Controller:** Menghubungkan model dan view, menangani permintaan pengguna, memproses input, dan menentukan output yang akan ditampilkan melalui view.

Pemisahan ini memungkinkan pengembangan aplikasi yang lebih terstruktur, dengan kode yang lebih mudah dipelihara dan diperbarui.

2.1.3. Persiapan Lingkungan Pengembangan

Sebelum mulai bekerja dengan CodeIgniter, penting untuk mempersiapkan lingkungan pengembangan:

- **Instalasi XAMPP/WAMP:** Instal server lokal yang mencakup Apache, MySQL, dan PHP. Ini akan memungkinkan Anda untuk menjalankan aplikasi CodeIgniter di komputer Anda.
- **Konfigurasi CodeIgniter:** Setelah mengunduh CodeIgniter, ekstrak file zip ke folder htdocs di XAMPP atau www di WAMP. Buka file config.php di folder application/config untuk mengatur base URL sesuai dengan lokasi proyek Anda.
- **Menyiapkan Database:** Buat database baru di phpMyAdmin untuk digunakan dengan aplikasi CodeIgniter. Ini adalah langkah penting sebelum Anda mulai membuat model untuk berinteraksi dengan database.

2.2. Membuat Sistem Login dan Registrasi

2.2.1. Desain Database untuk Sistem Login

Sistem login memerlukan struktur database yang efisien untuk mengelola data pengguna. Struktur dasar tabel users mungkin termasuk kolom berikut:

- id: Integer, Primary Key, Auto Increment – Sebagai penanda unik untuk setiap pengguna.
- username: Varchar(50), Unique – Nama pengguna yang unik.
- password: Varchar(255) – Kata sandi yang dienkripsi (gunakan hashing untuk keamanan).
- email: Varchar(100) – Email pengguna, juga bisa digunakan sebagai pengenal unik.
- user_role Integer – Menentukan level akses pengguna (misalnya, 1 untuk admin, 2 untuk pengguna biasa).

Model dalam CodeIgniter adalah komponen yang berinteraksi dengan database. Model `User_model` akan berisi fungsi untuk melakukan operasi CRUD (Create, Read, Update, Delete) pada data pengguna. Model ini akan mengelola data pengguna, termasuk proses validasi dan autentikasi.

Contoh Metode dalam `User_model`:



- **get_user(\$username):** Fungsi ini mengambil data pengguna berdasarkan username. Data ini kemudian digunakan untuk memverifikasi login.
- **create_user(\$data):** Fungsi ini memasukkan data pengguna baru ke dalam database setelah pendaftaran.

2.2.3. Membuat Controller untuk Autentikasi

Controller adalah bagian yang menangani logika untuk proses login dan registrasi. Anda akan membuat controller Auth yang mengatur bagaimana pengguna mendaftar, login, dan logout.

Contoh Metode dalam Auth Controller:

- **login():** Fungsi ini menangani input login dari pengguna, memvalidasi kredensial mereka, dan jika valid, menyimpan data pengguna dalam session.
- **register():** Fungsi ini menangani pendaftaran pengguna baru, termasuk memvalidasi input, hashing password, dan menyimpan data pengguna ke dalam database.
- **logout():** Fungsi ini menghapus session pengguna, mengakhiri sesi mereka, dan mengarahkan mereka kembali ke halaman login.

2.2.4. Membuat View untuk Login dan Registrasi

View dalam CodeIgniter adalah komponen yang menangani tampilan antarmuka pengguna. Anda akan membuat view login.php dan register.php yang berisi formulir untuk login dan registrasi.

Komponen yang Perlu Diperhatikan:

- **Form Validation:** Pastikan setiap input dalam form divalidasi sebelum data dikirimkan. Misalnya, username harus unik dan password harus memiliki panjang minimal.
- **User Feedback:** Tampilkan pesan sukses atau error berdasarkan hasil dari proses login atau registrasi.

2.3. Manajemen Pengguna dan Level Akses

2.3.1. Mengelola Akses Pengguna

Sistem login tidak hanya memverifikasi kredensial pengguna tetapi juga harus mengelola akses ke berbagai bagian dari aplikasi berdasarkan level pengguna. Misalnya, hanya pengguna dengan level admin yang dapat mengakses halaman manajemen pengguna.

Pengelolaan Akses:

- **Redirecting:** Pengguna yang mencoba mengakses halaman yang tidak sesuai dengan hak akses mereka harus dialihkan ke halaman yang tepat, seperti halaman error atau dashboard utama mereka.



2.3.2. Membuat Fitur Manajemen Pengguna

Fitur manajemen pengguna memungkinkan admin untuk melihat, menambahkan, mengedit, dan menghapus pengguna lain. Ini bisa dilakukan dengan membuat controller Admin yang hanya bisa diakses oleh pengguna dengan level admin.

Komponen Utama Fitur Manajemen:

- **Tabel Daftar Pengguna:** Menampilkan semua pengguna dalam bentuk tabel dengan opsi untuk mengedit atau menghapus.
- **Form Tambah/Edit Pengguna:** Formulir yang digunakan untuk menambahkan pengguna baru atau mengedit data pengguna yang ada.
- **Pemberian dan Penghapusan Hak Akses:** Admin dapat mengubah level akses pengguna atau menghapus akses sepenuhnya jika diperlukan.

2.4. Integrasi dengan Bootstrap untuk Antarmuka Pengguna

2.4.1. Mengapa Menggunakan Bootstrap?

Bootstrap adalah framework front-end yang memungkinkan pengembang untuk membuat situs web yang responsif dan modern dengan cepat. Bootstrap menyediakan berbagai komponen UI seperti tombol, formulir, navigasi, dan lainnya yang dapat diintegrasikan ke dalam aplikasi CodeIgniter dengan mudah.

2.4.2. Menggunakan Template Bootstrap

Untuk membuat tampilan yang profesional, Anda dapat menggunakan template Bootstrap seperti Nice Admin. Integrasikan template ini ke dalam aplikasi CodeIgniter dengan menempatkan file CSS dan JavaScript di dalam folder assets.

Langkah-Langkah Integrasi:

1. **Mengunduh Template:** Unduh template yang diinginkan dari situs penyedia seperti [BootstrapMade](#).
2. **Menempatkan File:** Tempatkan file CSS, JS, dan font dari template ke dalam folder assets di proyek CodeIgniter Anda.
3. **Menghubungkan View dengan Template:** Edit view Anda untuk menggunakan template ini dengan menghubungkan file CSS dan JS di dalam view.

2.4.3. Menyesuaikan Tampilan Login dan Dashboard

Setelah template terintegrasi, sesuaikan tampilan halaman login, registrasi, dan dashboard agar sesuai dengan gaya visual dari template. Anda juga dapat menambahkan komponen UI seperti modal, alert, dan lain-lain untuk meningkatkan interaksi pengguna.



2.5. Studi Kasus: Membangun Sistem Login Lengkap

Sebagai penerapan dari semua konsep yang telah dipelajari, peserta didik akan diminta untuk membangun sistem login yang lengkap dengan fitur-fitur berikut:

- **Registrasi Pengguna Baru:** Formulir pendaftaran yang menyimpan data pengguna ke dalam database dengan hashing password.
- **Login dan Logout:** Sistem autentikasi yang memungkinkan pengguna untuk login, mengakses dashboard, dan logout.
- **Manajemen Pengguna:** Dashboard admin yang memungkinkan pengelolaan pengguna dan level akses mereka.
- **Tampilan Responsif:** Menggunakan Bootstrap untuk memastikan antarmuka yang responsif dan menarik.

Peserta didik akan mengembangkan aplikasi ini secara bertahap, dimulai dari pengaturan lingkungan hingga implementasi fitur-fitur lanjutan seperti manajemen pengguna dan pengelolaan akses. Studi kasus ini akan memberikan pengalaman langsung dalam mengembangkan aplikasi web yang lengkap menggunakan CodeIgniter dan Bootstrap.

3. Kesimpulan

3.1. Ringkasan

Dalam modul ini, peserta didik telah mempelajari langkah demi langkah cara membangun sistem login lengkap menggunakan CodeIgniter 3. Modul ini tidak hanya membahas aspek teknis dari pembuatan sistem login tetapi juga menekankan pentingnya desain yang responsif, keamanan aplikasi, dan pengelolaan level akses pengguna. Dengan pengetahuan ini, peserta didik diharapkan dapat mengembangkan aplikasi web yang aman, efisien, dan mudah digunakan.