

BKPM
(BUKU KERJA PRAKTIK MAHASISWA)



TIF4508
WORKSHOP WEB FRAMEWORK

SEMESTER 4

Penyusun:
Khafidurrohman Agustianto, S.Pd., M.Eng.
NIP. 19911211 201803 1 001
NIDN. 0011129102

D4 TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI JEMBER
2018/2019

**KEMENTRIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI JEMBER**

**LEMBAR PENGESAHAN
WORKSHOP WEB FRAMEWORK**

Penulis,

**Khafidurrohman Agustianto, S.Pd., M.Eng.
NIP. 19911211 201803 1 001**

**Mensetujui,
Ketua Jurusan**

**Mengetahui,
Kepala Program Studi**

**Wahyu Kurnia Dewanto, S.Kom., M.T.
NIP. 19710408 200112 1 003**

**Elly Antika, S.T., M.Kom.
NIP. 19781011 200501 2 002**

DAFTAR ISI

LEMBAR PENGESAHAN	ii
DAFTAR ISI.....	iii
VERSI BAHASA INDONESIA	5
ATURAN PRAKTIK WORKSHOP WEB FRAMEWORK	6
PENILAIAN PRAKTIK WORKSHOP WEB FRAMEWORK.....	7
KELENGKAPAN PRAKTIK WORKSHOP WEB FRAMEWORK.....	8
PRAKTIK 1: GitHub, Rest API, MVC.....	9
1. Kompetensi Dasar	9
2. Dasar Teori	9
3. Kegiatan Praktik	13
4. Refrensi <i>Website</i>	14
PRAKTIK 2: Bootstrap + CI 3	15
1. Kompetensi Dasar	15
2. Dasar Teori	15
3. Kegiatan Praktik	17
4. Refrensi <i>Website</i>	18
PRAKTIK 3: CRUD + CI 3.....	19
1. Kompetensi Dasar	19
2. Dasar Teori	19
3. Kegiatan Praktik	19
4. Refrensi <i>Website</i>	19
PRAKTIK 4: REST API + CI 3.....	21
1. Kompetensi Dasar	21
2. Dasar Teori	21
3. Kegiatan Praktik	23
4. Refrensi <i>Website</i>	23
PRAKTIK 5: Implementasi Project Tahap 1	25
1. Kompetensi Dasar	25
2. Kegiatan Praktik	25
3. Refrensi <i>Website</i>	25
PRAKTIK 6: Implementasi Project Tahap 2	26
1. Kompetensi Dasar	26
2. Kegiatan Praktik	26

3. Refrensi <i>Website</i>	26
PRAKTIK 7: Implementasi Project Tahap 3	27
1. Kompetensi Dasar	27
2. Kegiatan Praktik	27
3. Refrensi <i>Website</i>	27
PRAKTIK 8: Presentasi Project	28
1. Kompetensi Dasar	28
2. Kegiatan Praktik	28
3. Refrensi <i>Website</i>	28
DAFTAR PUSTAKA	29

VERSI BAHASA INDONESIA

ATURAN PRAKTIK WORKSHOP WEB FRAMEWORK

- Masing-masing kelompok membuat Repo. di GitHub untuk Project-nya dan mengundang dosen pengampu dalam Repo.
- Kelas Dilaksanakan 2X/Minggu, dengan ketentuan:
 - Pertemuan pertama digunakan untuk mebahas materi
 - Pertemuan kedua digunakan untuk berkonsultasi dan melaporkan hasil workshop
- Tidak boleh terlambat
 - Keterlambatan dihitung setelah dosen hadir di kelas
- Berpakaian sopan
 - Tidak memakai sandal/Sepatu-Sendal/Sejenis
 - Tidak memakai Baju dan Celana Robek
 - Tidak memakai Kaos/Oblong/T-Shirt
 - Tidak memakai Topi/Penutup Kepala Lainnya
- Dilarang makan tetapi diperbolehkan minum, dengan tetap menjaga jarak aman dengan komputer
- Boleh tidur di kursi jika mengantuk

PENILAIAN PRAKTIK WORKSHOP WEB FRAMEWORK

- Presensi Perkuliahan/Attitude
- Jumlah Commit Masing-Masing Contributor
- Melihat Kontribusi dari Masing-Masing Anggota Kelompok
- Kesesuaian Perencanaan dengan Produk Akhir
 - TIF4503 Kewirausahaan
 - TIF4503 Administrasi Basis Data
 - TIF4504 Perawatan Perangkat Lunak
 - TIF4506 Sistem Informasi Jasa Berbasis Layanan
 - TIF4507 Workshop Aplikasi Mobile
- Penilaian Demo Aplikasi di Akhir Semester
- Penguasaan Masing-Masing Anggota

KELENGKAPAN PRAKTIK WORKSHOP WEB FRAMEWORK

- GitHub Desktop
- Visual Studio Code (Disarankan), atau Bisa Menggunakan Aplikasi Sejenis
- XAMPP
- Composer
- Framework CI 3
- Rest API

PRAKTIK 1: GitHub, Rest API, MVC



Matakuliah : Workshop WEB FrameWork
Minggu Ke : 2 dan 3
Waktu : 2 x 100 menit
Tema : GitHub, Rest API, MVC

1. Kompetensi Dasar

- A. Mahasiswa mampu memahami konsep kolaborasi
- B. Mahasiswa mampu memahami konsep Rest-API
- C. Mahasiswa mampu memahami konsep dasar MVC

2. Dasar Teori

A. GitHub

- <http://rogerdudler.github.io/git-guide/index.id.html>
- <https://github.com/>

GitHub is a web-based hosting service for version control using Git. It is mostly used for computer code. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

GitHub vs Bitbucket

GiHub	Bitbucket
First of all, Github only hosts projects that use the Git VCS. That's it. Nothing else. But Git is far and away the most commonly used VCS, so Github is still the largest code host of them all, with some 13.7 million-plus repositories of code. Git was originally started in 2008, and was written in both Ruby and Erlang. Github is designed to encourage close communication and collaboration within development teams. To this end it includes features like highlighted code comments and collaborative code review.	Right off the bat, Bitbucket's advantage over Github is that it supports the Mercurial VCS in addition to Git. But it also doesn't support SVN, yet. Bitbucket is written in Python and uses the Django web framework. Bitbucket was also launched in 2008 in Australia and was originally an independent startup offering hosting only for Mercurial projects. It was acquired in 2010 by fellow Australian company Atlassian, and about a year later added support for Git repos

<p>Fitur:</p> <ol style="list-style-type: none"> 1. An integrated issue tracker right within your project 2. Milestones and labels within projects 3. Branch comparison views 4. Native applications for Windows and Mac desktops, and also an Android app 5. Support for over 200 programming languages and data formats 6. Github pages, a feature for publishing and hosting within Github 7. Security such as use of SSL, SSH and/or https for all data transmission, and two-factor authentication for login 8. API integration for easy integration of 3rd-party tools, and integration with a large number of other tools and platforms. Some examples are Asana and Zendesk for issue/ bug tracking; CloudBees, Travis and CodeClimate for Continuous Integration (CI); AWS, Windows Azure, Google Cloud, and Heroku cloud hosting. 9. The Github guys also recognize that SVN is also a widely used alternative to Git, so they provide a tool to import SVN repos into Git and host them on Github, although reports are that it's at best a clunky, somewhat awkward solution. And they shrewdly made sure that Github repos are fully accessible on the SVN client. 10. Syntax highlighting. Github users will be used to this as a standard, indispensable feature, but Bitbucket notably continues to lack it. 	<p>Fitur:</p> <ol style="list-style-type: none"> 1. Pull requests and code reviews 2. Unlimited private repos 3. Branch comparison and commit history 4. Bitbucket Mac and Windows client called SourceTree; Android app called BitBeaker 5. Bitbucket for Enterprises, called Stash 6. Integration with tools like Jira, Crucible, Bamboo, Jenkins
--	---

GitHub

Plan Type	Plan	Max. no. of users	Price (US\$ per month)	Private Repositories	Public Repositories
Free	Free	Unlimited	0	0	Unlimited
Individuals	Micro	Unlimited	\$7	5	Unlimited
	Small	Unlimited	\$12	10	Unlimited
	Medium	Unlimited	\$22	20	Unlimited
	Large	Unlimited	\$50	50	Unlimited
Organization	Bronze	Unlimited	\$25	10	Unlimited
	Silver	Unlimited	\$50	20	Unlimited
	Gold	Unlimited	\$100	50	Unlimited
	Platinum	Unlimited	\$200	125	Unlimited

Bitbucket

ax no. of users	Price (US\$ per month)	Private Repositories	Public Repositories
5	Free	Unlimited	Unlimited
10	\$10	Unlimited	Unlimited
25	\$25	Unlimited	Unlimited
50	\$50	Unlimited	Unlimited
100	\$100	Unlimited	Unlimited
Unlimited	\$200	Unlimited	Unlimited

B. RestAPI

- <https://www.codepolitan.com/rest-api-server-sederhana-dengan-codeigniter-58901f324a29f>

REST, singkatan bahasa Inggris dari Representational State Transfer, adalah suatu gaya arsitektur perangkat lunak untuk pendistribusian sistem hipermedia seperti www. Istilah ini diperkenalkan pertama kali pada tahun 2000 pada disertasi doktoral Roy Fielding. Pada arsitektur REST, REST server menyediakan resources (sumber daya/data) dan REST client mengakses dan menampilkan resource tersebut untuk penggunaan selanjutnya.

CodeIgniter merupakan aplikasi sumber terbuka yang berupa framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP. Dalam penerapan REST pada Codeigniter diperlukan beberapa library tambahan yang tidak disediakan secara default pada Codeigniter, salah satu library yang dapat digunakan adalah library dari Chris Kacerguis.

C. MVC

- <https://www.codepolitan.com/tutorial-codeigniter-3-dalam-satu-jam-597820fc3cd23>
- <https://www.petanikode.com/codeigniter-mvc/>

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application.

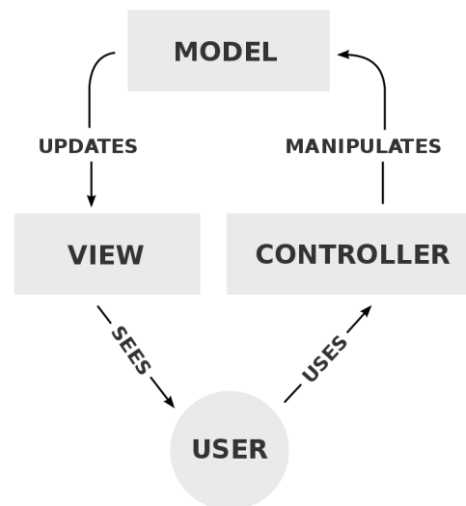


Figure 1. MVC

Model-View-Controller in CodeIgniter

CodeIgniter is based on the Model-View-Controller development pattern. MVC is a software approach that separates application logic from presentation. In practice, it permits your web pages to contain minimal scripting since the presentation is separate from the PHP scripting.

- The Model represents your data structures. Typically your model classes will contain functions that help you retrieve, insert, and update information in your database.
- The View is the information that is being presented to a user. A View will normally be a web page, but in CodeIgniter, a view can also be a page fragment like a header or footer. It can also be an RSS page, or any other type of “page”.
- The Controller serves as an intermediary between the Model, the View, and any other resources needed to process the HTTP request and generate a web page.

CodeIgniter has a fairly loose approach to MVC since Models are not required. If you don’t need the added separation, or find that maintaining models requires more complexity than you want, you can ignore them and build your application minimally using Controllers and Views. CodeIgniter also enables you to

incorporate your own existing scripts, or even develop core libraries for the system, enabling you to work in a way that makes the most sense to you.

Application Flow Chart

The following graphic illustrates how data flows throughout the system:

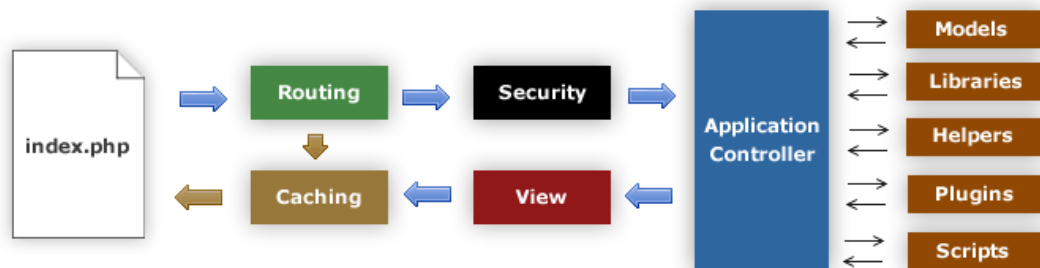


Figure 2. CodeIgniter Application Flow

- The index.php serves as the front controller, initializing the base resources needed to run CodeIgniter.
- The Router examines the HTTP request to determine what should be done with it.
- If a cache file exists, it is sent directly to the browser, bypassing the normal system execution.
- Security. Before the application controller is loaded, the HTTP request and any user submitted data is filtered for security.
- The Controller loads the model, core libraries, helpers, and any other resources needed to process the specific request.
- The finalized View is rendered then sent to the web browser to be seen. If caching is enabled, the view is cached first so that on subsequent requests it can be served.

3. Kegiatan Praktik

- A. Mahasiswa berkumpul dengan anggota kelompoknya
- B. Masing-masing anggota meng-Instal GitHub Desktop
 - Ketua membuat Repo. baru untuk project, disarankan langsung mengambil dari direktori lokal ketua (htdocs).
 - Masing-masing anggota kemudian dimaukan ke dalam Repo
 - Mengundang dosen pengampu dalam Repo
- C. Masing-masing anggota me-refresh GitHub Desktop
- D. Ketua kelompok menginstal CI 3 dengan Rest API pada htdoc-nya
 - Berikut adalah link download:
 - <https://www.codepolitan.com/rest-api-server-sederhana-dengan-codeigniter-58901f324a29f>
- E. Anggota kelompok melakukan Pull terhadap Repo yang telah dibuat ketua

4. Refrensi *Website*

- [1] <https://www.codeigniter.com/>
- [2] <https://codeigniter-id.github.io/user-guide/>
- [3] <https://www.w3schools.com/>
- [4] <https://www.tutorialspoint.com/codeigniter/>
- [5] <https://www.malasngoding.com/category/codeigniter/>
- [6] <https://www.codepolitan.com/>

PRAKTIK 2: Bootstrap + CI 3



Matakuliah : Workshop WEB FrameWork
Minggu Ke : 4 dan 5
Waktu : 2 x 100 menit
Tema : Bootstrap + CI 3

1. Kompetensi Dasar

- A. Mahasiswa mampu memahami konsep kolaborasi
- B. Mahasiswa mampu mengaplikasikan Bootstrap dengan CI 3
- C. Mahasiswa mampu mengkonfigurasi FrameWork yang digunakan sesuai dengan kebutuhan praktik

2. Dasar Teori

A. Bootstrap

- <https://getbootstrap.com/>
- <https://www.malasngoding.com/pengertian-dan-cara-menggunakan-bootstrap/>

Bootstrap adalah sebuah library framework CSS yang di buat khusus untuk bagian pengembangan front-end website. bootstrap merupakan salah satu framework HTML, CSS dan javascript yang paling populer di kalangan web developer. pada saat ini hampir semua web developer telah menggunakan bootstrap untuk membuat tampilan front-end menjadi lebih mudah dan sangat cepat. karena anda hanya perlu menambahkan class-class tertentu untuk misalnya membuat tombol, grid, navigasi dan lainnya.

Bootstrap telah menyediakan kumpulan komponen class interface dasar yang telah di rancang sedemikian rupa untuk menciptakan tampilan yang menarik, bersih dan ringan. selain komponen class interface, bootstrap juga memiliki fitur grid yang berfungsi untuk mengatur layout pada halaman website yang bisa digunakan dengan sangat mudah dan cepat. dengan menggunakan bootstrap kita juga di beri keleluasaan dalam mengembangkan tampilan website yang menggunakan bootstrap yaitu dengan cara mengubah tampilan bootstrap dengan menambahkan class dan CSS sendiri.

Tentu anda bertanya-tanya kenapa sangat banyak yang telah menggunakan bootstrap dalam pengembangan website. berikut ini akan di jelaskan beberapa kegunaan yang telah menjadi kelebihan pada bootstrap. adapun beberapa kelebihan bootstrap adalah sebagai berikut:

- Penggunaan bootstrap sangat menghemat waktu.
- Tampilan bootstrap yang sudah cukup terlihat modern.

- Mobile Friendly yang maksudnya tampilan bootstrap sudah sangat responsive, yaitu tampilan bootstrap sudah mendukung segala jenis resolusi, baik itu pc, laptop, tablet dan smartphone.
- Sangat ringan karena bootstrap di buat dengan sangat terstruktur.
- Dan masih banyak lagi kelebihan dan kegunaan dari bootstrap yang akan anda temukan sendiri setelah mencoba membangun sebuah aplikasi berbasis web dengan menggunakan bantuan framework css bootstrap.

B. CI 3

- <https://www.codeigniter.com/>
- <https://www.malasngoding.com/pengertian-dan-cara-menggunakan-codeigniter/>

Framework adalah kumpulan intruksi-intruksi yang di kumpulkan dalam class dan function-function dengan fungsi masing-masing untuk memudahkan developer dalam pemanggilannya tanpa harus menuliskan syntax program yang sama berulang-ulang. hal ini memiliki kegunaan untuk menghemat waktu dan mencegah penulisan syntax secara berulang-ulang agar tercipta nya source code yang bersih dan terstruktur.

Codeigniter adalah sebuah framework php yang bersifat open source dan menggunakan metode MVC (Model, View, Controller). codeigniter bersifat free alias tidak berbayar jika anda menggunakannya. framework codeigniter di buat dengan tujuan sama seperti framework lainnya yaitu untuk memudahkan developer atau programmer dalam membangun sebuah aplikasi berbasis web tanpa harus membuat nya dari awal.

MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu model, view dan controller.

– Model

Model merupakan bagian penanganan yang berhubungan dengan pengolahan atau manipulasi database. seperti misalnya mengambil data dari database, menginput dan pengolahan database lainnya. semua intruksi yang berhubungan dengan pengolahan database di letakkan di dalam model.

– View

View merupakan bagian yang menangani halaman user interface atau halaman yang muncul pada user. tampilan dari user interface di kumpulkan pada view untuk memisahkannya dengan controller dan model sehingga memudahkan web designer dalam melakukan pengembangan tampilan halaman website.

– Controller

Controller merupakan kumpulan intruksi aksi yang menghubungkan model dan view, jadi user tidak akan berhubungan dengan model secara langsung, intinya dari view kemudian controller yang mengolah intruksi.

Berdasar penjelasan tentang model view dan controller di atas dapat disimpulkan bahwa controller sebagai penghubung view dan model. misalnya pada aplikasi yang menampilkan data dengan menggunakan metode konsep mvc, controller memanggil intruksi pada model yang mengambil data pada database, kemudian controller yang meneruskannya pada view untuk di tampilkan. jadi jelas sudah dan sangat mudah dalam pengembangan aplikasi dengan cara mvc ini karena web designer atau front-end developer tidak perlu lagi berhubungan dengan controller, dia hanya perlu berhubungan dengan view untuk mendesign tampilann aplikasi, karena back-end developer yang menangani bagian controller dan modelnya. jadi pembagian tugas pun menjadi mudah dan pengembangan aplikasi dapat di lakukan dengan cepat.

Kelebihan CodeIgniter

Adapun beberapa kelebihan dari codeigniter adalah:

- Syntax yang terstruktur
- Kemudahan dalam menggunakannya
- Codeigniter menyediakan fasilitas helper dan library yang dapat membantu developer dalam membuat pagination, session, manipulasi url dan lainnya yang akan kita pelajari pada tutorial codeigniter selanjutnya.
- Keamanan yang sudah lumayan karena user atau pengakses aplikasi tidak berhubungan langsung dengan database.

3. Kegiatan Praktik

- A. Mahasiswa berkumpul dengan anggota kelompoknya
- B. Masing-masing anggota me-refresh GitHub Desktop
- C. Mengikuti langkah-langkah pada link di bawah
 - <https://www.malasngoding.com/pengertian-dan-cara-menggunakan-codeigniter/>
 - <https://www.petanikode.com/codeigniter-bootstrap/>
 - Setiap anggota membuat folder baru pada htdocs, dengan nama CI.3.NIM_P2
 - Masukkan bootstrap yang telah dipilih pada folder tersebut, pastikan semua fotur dalam Bottstrap yang dipilih dapat tampil semua
 - Lanjutkan dengan kegiatan berikut:
 1. <https://www.malasngoding.com/cara-membuat-controller-pada-codeigniter/>
 2. <https://www.malasngoding.com/cara-membuat-view-pada-codeigniter/>
 3. <https://www.malasngoding.com/mengenal-helper-html-pada-codeigniter/>
 4. <https://www.malasngoding.com/menghubungkan-codeigniter-dengan-database-mysql/>
 5. <https://www.malasngoding.com/membuat-form-validation-pada-codeigniter/>
 6. <https://www.malasngoding.com/membuat-download-file-dengan-codeigniter/>

7. <https://www.malasngoding.com/membuat-library-sendiri-di-codeigniter/>

8. <https://www.malasngoding.com/pengertian-uri-segment-pada-codeigniter/>

D. Anggota kelompok melakukan Pull terhadap Repo yang telah dibuat ketua

4. Refrensi Website

[1] <https://www.codeigniter.com/>

[2] <https://codeigniter-id.github.io/user-guide/>

[3] <https://www.w3schools.com/>

[4] <https://www.tutorialspoint.com/codeigniter/>

[5] <https://www.malasngoding.com/category/codeigniter/>

[6] <https://www.codepolitan.com/>

PRAKTIK 3: CRUD + CI 3



Matakuliah : Workshop WEB FrameWork
Minggu Ke : 6 dan 7
Waktu : 2 x 100 menit
Tema : CRUD + CI 3

1. Kompetensi Dasar

- A. Mahasiswa mampu memahami pengaplikasian CRUD dengan CI 3
- B. Mahasiswa mampu membuat halaman Web yang mengaplikasikan CRUD
- C. Mahasiswa mampu membuat halaman login sederhana

2. Dasar Teori

A. CRUD

- <https://www.codepolitan.com/belajar-codeigniter-3-crud-1-57a7f9f5bde51-17045>
- <https://www.malasngoding.com/membuat-crud-dengan-codeigniter-input-data-ke-database/>

Fungsi CRUD (Create Read Update dan Delete), CRUD merupakan dasar dari pembuatan suatu Web App yang berhubungan dengan database dan pengolahannya.

3. Kegiatan Praktik

- A. Mahasiswa berkumpul dengan anggota kelompoknya
- B. Masing-masing anggota me-refresh GitHub Desktop
- C. Mengikuti langkah-langkah pada link di bawah
 - 1. Setiap anggota membuat folder baru pada htdocs, dengan nama CI.3.NIM_P3
 - 2. <https://www.codepolitan.com/belajar-codeigniter-3-crud-1-57a7f9f5bde51-17045>
 - 3. <https://www.malasngoding.com/membuat-crud-dengan-codeigniter-menampilkan-data-dari-database/>
 - 4. <https://www.malasngoding.com/membuat-crud-dengan-codeigniter-input-data-ke-database/>
 - 5. <https://www.malasngoding.com/membuat-crud-dengan-codeigniter-hapus-data/>
 - 6. <https://www.malasngoding.com/membuat-crud-dengan-codeigniter-update-data/>
 - 7. <https://www.malasngoding.com/membuat-login-dengan-codeigniter/>
- D. Anggota kelompok melakukan Pull terhadap Repo yang telah dibuat ketua

4. Refrensi Website

- [1] <https://www.codeigniter.com/>
- [2] <https://codeigniter-id.github.io/user-guide/>
- [3] <https://www.w3schools.com/>
- [4] <https://www.tutorialspoint.com/codeigniter/>

[5] <https://www.malasngoding.com/category/codeigniter/>

[6] <https://www.codepolitan.com/>

PRAKTIK 4: REST API + CI 3



Matakuliah : Workshop WEB FrameWork
Minggu Ke : 9 dan 10
Waktu : 2 x 100 menit
Tema : REST API + CI 3

1. Kompetensi Dasar

- A. Mahasiswa mampu memahami konsep RESTFull Web Service
- B. Mahasiswa mampu mengaplikasikan GET, POST, Put dan DELETE
- C. Mahasiswa mampu membuat aplikasi dengan RestAPI sederhana

2. Dasar Teori

A. RESTful Web Services (GET, POST, PUT dan DELETE)

REST (REpresentational State Transfer) merupakan standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web. Umumnya menggunakan HTTP (Hypertext Transfer Protocol) sebagai protocol untuk komunikasi data. REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000.

Pada arsitektur REST, REST server menyediakan resources(sumber daya/data) dan REST client mengakses dan menampilkan resource tersebut untuk penggunaan selanjutnya. Setiap resource diidentifikasi oleh URIs (Universal Resource Identifiers) atau global ID. Resource tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML.

Keuntungan REST:

- bahasa dan platform agnostic
- lebih sederhana/simpel untuk dikembangkan ketimbang SOAP
- mudah dipelajari, tidak bergantung pada tools
- ringkas, tidak membutuhkan layer pertukaran pesan (messaging) tambahan
- secara desain dan filosofi lebih dekat dengan web

Kelemahan REST:

- Mengasumsi model point-to-point komunikasi - tidak dapat digunakan untuk lingkungan komputasi terdistribusi di mana pesan akan melalui satu atau lebih perantara
- Kurangnya dukungan standar untuk keamanan, kebijakan, keandalan pesan, dll, sehingga layanan yang mempunyai persyaratan lebih canggih lebih sulit untuk dikembangkan ("dipecahkan sendiri")
- Berkaitan dengan model transport HTTP

Berikut metode HTTP yang umum digunakan dalam arsitektur berbasis REST.

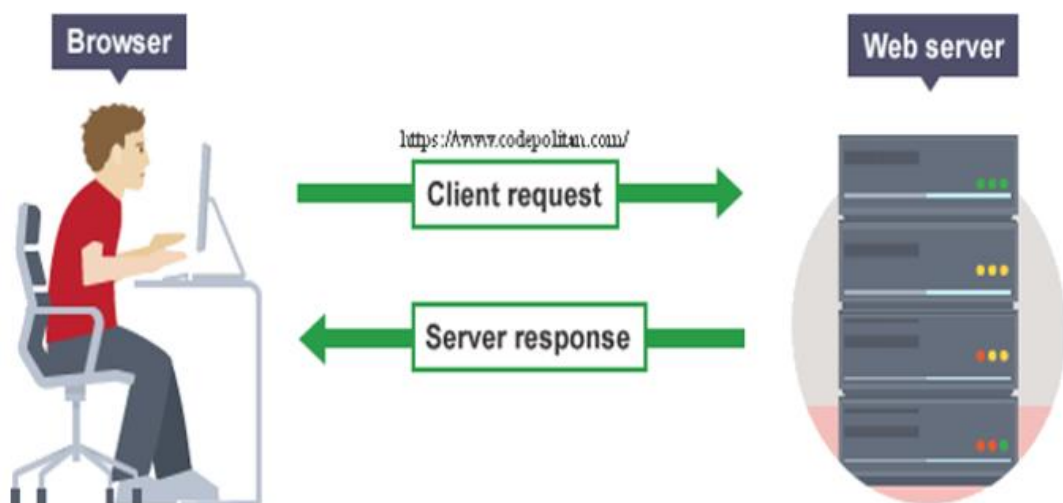
- **GET**, menyediakan hanya akses baca pada resource
- **PUT**, digunakan untuk menciptakan resource baru
- **DELETE**, digunakan untuk menghapus resource
- **POST**, digunakan untuk memperbarui resource yang ada atau membuat resource baru
- **OPTIONS**, digunakan untuk mendapatkan operasi yang disupport pada resource

Web service adalah standar yang digunakan untuk melakukan pertukaran data antar aplikasi atau sistem, karena aplikasi yang melakukan pertukaran data bisa ditulis dengan bahasa pemrograman yang berbeda atau berjalan pada platform yang berbeda. Contoh implementasi dari web service antara lain adalah SOAP dan REST.

Web service yang berbasis arsitektur REST kemudian dikenal sebagai RESTful web services. Layanan web ini menggunakan metode HTTP untuk menerapkan konsep arsitektur REST.

Cara Kerja RESTful Web Services

[caption id="attachment_7934" align="aligncenter" width="700"]



Sebuah client mengirimkan sebuah data atau request melalui **HTTP Request** dan kemudian server merespon melalui **HTTP Response**. Komponen dari **http request**:

- Verb, HTTP method yang digunakan misalnya GET, POST, DELETE, PUT dll.
- Uniform Resource Identifier (URI) untuk mengidentifikasikan lokasi resource pada server.
- HTTP Version, menunjukkan versi dari HTTP yang digunakan, contoh HTTP v1.1.

- Request Header, berisi metadata untuk HTTP Request. Contoh, type client/browser, format yang didukung oleh client, format dari body pesan, seting cache dll.
- Request Body, konten dari data.

Sedangkan komponen dari **http response** :

- Status/Response Code, mengindikasikan status server terhadap resource yang direquest. misal : 404, artinya resource tidak ditemukan dan 200 response OK.
- HTTP Version, menunjukkan versi dari HTTP yang digunakan, contoh HTTP v1.1.
- Response Header, berisi metadata untuk HTTP Response. Contoh, type server, panjang content, tipe content, waktu response, dll
- Response Body, konten dari data yang diberikan.

B. RestAPI

- <https://www.codepolitan.com/rest-api-server-sederhana-dengan-codeigniter-58901f324a29f>

REST, singkatan bahasa Inggris dari Representational State Transfer, adalah suatu gaya arsitektur perangkat lunak untuk untuk pendistribusian sistem hipermedia seperti www. Istilah ini diperkenalkan pertama kali pada tahun 2000 pada disertasi doktoral Roy Fielding. Pada arsitektur REST, REST server menyediakan resources (sumber daya/data) dan REST client mengakses dan menampilkan resource tersebut untuk penggunaan selanjutnya.

CodeIgniter merupakan aplikasi sumber terbuka yang berupa framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP. Dalam penerapan REST pada Codeigniter diperlukan beberapa library tambahan yang tidak disediakan secara default pada Codeigniter, salah satu library yang dapat digunakan adalah library dari Chris Kacerguis.

3. Kegiatan Praktik

- Mahasiswa berkumpul dengan anggota kelompoknya
- Masing-masing anggota me-refresh GitHub Desktop
- Mengikuti langkah-langkah pada link di bawah
 - Setiap anggota membuat folder baru pada htdocs, dengan nama CI.3.NIM_P4
 - Ikuti langkah yang ada pada url berikut:
<https://www.codepolitan.com/rest-api-server-sederhana-dengan-codeigniter-58901f324a29f>
- Anggota kelompok melakukan Pull terhadap Repo yang telah dibuat ketua

4. Refrensi Website

- [1] <https://www.codeigniter.com/>
- [2] <https://codeigniter-id.github.io/user-guide/>

- [3] <https://www.w3schools.com/>
- [4] <https://www.tutorialspoint.com/codeigniter/>
- [5] <https://www.malasngoding.com/category/codeigniter/>
- [6] <https://www.codepolitan.com/>

PRAKTIK 5: Implementasi Project Tahap 1



Matakuliah : Workshop WEB FrameWork
Minggu Ke : 11 dan 12
Waktu : 2 x 100 menit
Tema : Implementasi Project Tahap 1

1. Kompetensi Dasar

- A. Mahasiswa mampu memahami konsep kolaborasi
- B. Mahasiswa mampu membuat proyek yang merupakan pengaplikasian hasil dari perkuliahan:
 - TIF4503 Kewirausahaan
 - TIF4503 Administrasi Basis Data
 - TIF4504 Perawatan Perangkat Lunak
 - TIF4506 Sistem Informasi Jasa Berbasis Layanan
 - TIF4507 Workshop Aplikasi Mobile
- C. Mahasiswa mampu mengevaluasi kesesuaian pekerjaan dengan perencanaan yang telah dibuat

2. Kegiatan Praktik

- A. Mahasiswa berkumpul dengan anggota kelompoknya
- B. Masing-masing anggota me-refresh GitHub Desktop
- C. Anggota kelompok masing-masing mengerjakan bagiannya dalam proyek
- D. Anggota kelompok melakukan Pull terhadap Repo yang telah dibuat ketua

3. Refrensi Website

- [1] <https://www.codeigniter.com/>
- [2] <https://codeigniter-id.github.io/user-guide/>
- [3] <https://www.w3schools.com/>
- [4] <https://www.tutorialspoint.com/codeigniter/>
- [5] <https://www.malasngoding.com/category/codeigniter/>
- [6] <https://www.codepolitan.com/>

PRAKTIK 6: Implementasi Project Tahap 2



Matakuliah : Workshop WEB FrameWork
Minggu Ke : 13 dan 14
Waktu : 2 x 100 menit
Tema : Implementasi Project Tahap 2

1. Kompetensi Dasar

- A. Mahasiswa mampu memahami konsep kolaborasi
- B. Mahasiswa mampu membuat proyek yang merupakan pengaplikasian hasil dari perkuliahan:
 - TIF4503 Kewirausahaan
 - TIF4503 Administrasi Basis Data
 - TIF4504 Perawatan Perangkat Lunak
 - TIF4506 Sistem Informasi Jasa Berbasis Layanan
 - TIF4507 Workshop Aplikasi Mobile
- C. Mahasiswa mampu mengevaluasi kesesuaian pekerjaan dengan perencanaan yang telah dibuat

2. Kegiatan Praktik

- A. Mahasiswa berkumpul dengan anggota kelompoknya
- B. Masing-masing anggota me-refresh GitHub Desktop
- C. Anggota kelompok masing-masing mengerjakan bagiannya dalam proyek
- D. Anggota kelompok melakukan Pull terhadap Repo yang telah dibuat ketua

3. Refrensi Website

- [1] <https://www.codeigniter.com/>
- [2] <https://codeigniter-id.github.io/user-guide/>
- [3] <https://www.w3schools.com/>
- [4] <https://www.tutorialspoint.com/codeigniter/>
- [5] <https://www.malasngoding.com/category/codeigniter/>
- [6] <https://www.codepolitan.com/>

PRAKTIK 7: Implementasi Project Tahap 3



Matakuliah : Workshop WEB FrameWork
Minggu Ke : 15
Waktu : 2 x 100 menit
Tema : Implementasi Project Tahap 2

1. Kompetensi Dasar

- A. Mahasiswa mampu memahami konsep kolaborasi
- B. Mahasiswa mampu membuat proyek yang merupakan pengaplikasian hasil dari perkuliahan:
 - TIF4503 Kewirausahaan
 - TIF4503 Administrasi Basis Data
 - TIF4504 Perawatan Perangkat Lunak
 - TIF4506 Sistem Informasi Jasa Berbasis Layanan
 - TIF4507 Workshop Aplikasi Mobile
- C. Mahasiswa mampu mengevaluasi kesesuaian pekerjaan dengan perencanaan yang telah dibuat

2. Kegiatan Praktik

- A. Mahasiswa berkumpul dengan anggota kelompoknya
- B. Masing-masing anggota me-refresh GitHub Desktop
- C. Anggota kelompok masing-masing mengerjakan bagiannya dalam proyek
- D. Anggota kelompok melakukan Pull terhadap Repo yang telah dibuat ketua

3. Refrensi Website

- [1] <https://www.codeigniter.com/>
- [2] <https://codeigniter-id.github.io/user-guide/>
- [3] <https://www.w3schools.com/>
- [4] <https://www.tutorialspoint.com/codeigniter/>
- [5] <https://www.malasngoding.com/category/codeigniter/>
- [6] <https://www.codepolitan.com/>

PRAKTIK 8: Presentasi Project



Matakuliah : Workshop WEB FrameWork
Minggu Ke : 17
Waktu : 2 x 100 menit
Tema : Presentasi Project

1. Kompetensi Dasar

- A. Mahasiswa mampu memahami konsep kolaborasi
- B. Mahasiswa mampu mengevaluasi kesesuaian pekerjaan dengan perencanaan yang telah dibuat
- C. Mahasiswa mampu menjelaskan kontribusinya pada proyek yang dikerjakan

2. Kegiatan Praktik

- A. Mahasiswa berkumpul dengan anggota kelompoknya
- B. Masing-masing anggota me-refresh GitHub Desktop
- C. Mahasiswa akan dipanggil bisa berkelompok/satu per satu untuk menjelaskan proyek yang telah dikerjakan
- D. Komponen penilaian yang digunakan:
 - a. Presensi Perkuliahan/Attitude
 - b. Jumlah Commit Masing-Masing Contributor
 - c. Melihat Kontribusi dari Masing-Masing Anggota Kelompok
 - d. Kesesuaian Perencanaan dengan Produk Akhir
 - TIF4503 Kewirausahaan
 - TIF4503 Administrasi Basis Data
 - TIF4504 Perawatan Perangkat Lunak
 - TIF4506 Sistem Informasi Jasa Berbasis Layanan
 - TIF4507 Workshop Aplikasi Mobile
 - e. Penilaian Demo Aplikasi di Akhir Semester
 - f. Penguasaan Masing-Masing Anggota
- E. Team penilai dari Workshop WEB FrameWork: Team Dosen Pengampu
- F. Waktu dan Tempat Presentasi/Demo Proyek: Menunggu pengumuman di akhir semester

3. Refrensi Website

- [1] <https://www.codeigniter.com/>
- [2] <https://codeigniter-id.github.io/user-guide/>
- [3] <https://www.w3schools.com/>
- [4] <https://www.tutorialspoint.com/codeigniter/>
- [5] <https://www.malasngoding.com/category/codeigniter/>
- [6] <https://www.codepolitan.com/>

DAFTAR PUSTAKA

- [1] <https://www.codeigniter.com/>
- [2] <https://codeigniter-id.github.io/user-guide/>
- [3] <https://www.w3schools.com/>
- [4] <https://www.tutorialspoint.com/codeigniter/>
- [5] <https://www.malasngoding.com/category/codeigniter/>
- [6] <https://www.codepolitan.com/>