

Dokumen Pengembangan Web Service

Disusun oleh:

Muhammad Faisal Aziz / 18215044



**Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung
Jalan Ganeca No. 10 Bandung 40131**

Deskripsi

Webservice yang dibuat adalah web service untuk mengetahui course-course yang tersedia secara real-time. Dari sistem ini dapat dibuat pendefinisian course baru, atau penghapusan course yang tidak digunakan. Pengembangan lebih lanjut dari webservice ini dapat dilakukan pada platform e-learning.

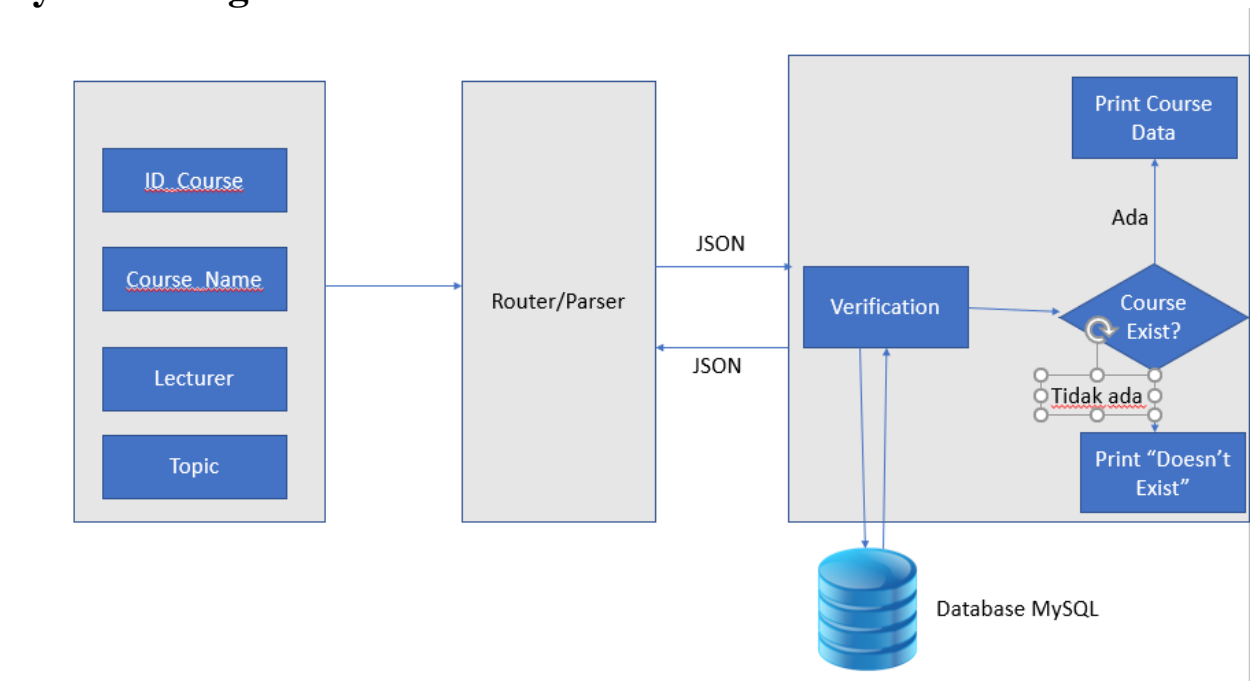
Kebutuhan Fungsional

1. Sistem dapat memberikan list course yang tersedia
2. Sistem dapat mencari sebuah course berdasarkan ID dari course tersebut
3. Sistem dapat membuat sebuah course baru
4. Sistem dapat melakukan perubahan terhadap course yang telah tersedia
5. Sistem dapat menghapus suatu course

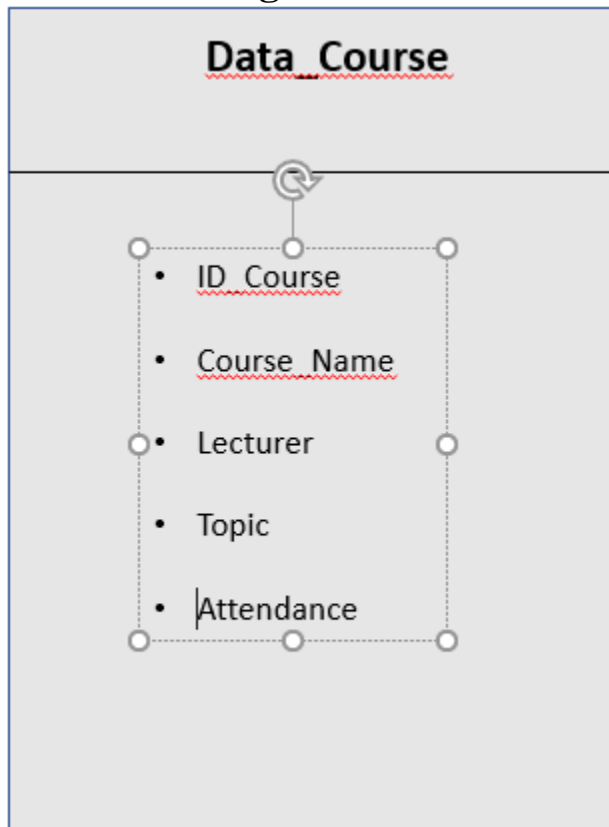
System Requirements

1. Golang 1.9
2. MongoDB
3. OS Ubuntu 16.04
4. Storage 10 GB
5. Penggunaan framework Gorilla/mux
6. Menggunakan encoding binary-json (bson)

System Design



Database Design



Instalasi

- Instalasi Golang 1.9
- Instalasi Gorilla/mux, mongodb package, dan toml (library untuk melakukan routing dan parsing)

```
$go get github.com/gorilla/mux
```

```
$go get gopkg.in/mgo.v2
```

```
$go get github.com/BurntSushi/toml
```

- Clone project ke dalam \$GOPATH

```
$ cd $GOPATH/src
```

```
$ git clone https://github.com/muhfai/courses.git
```

Melakukan Run Project

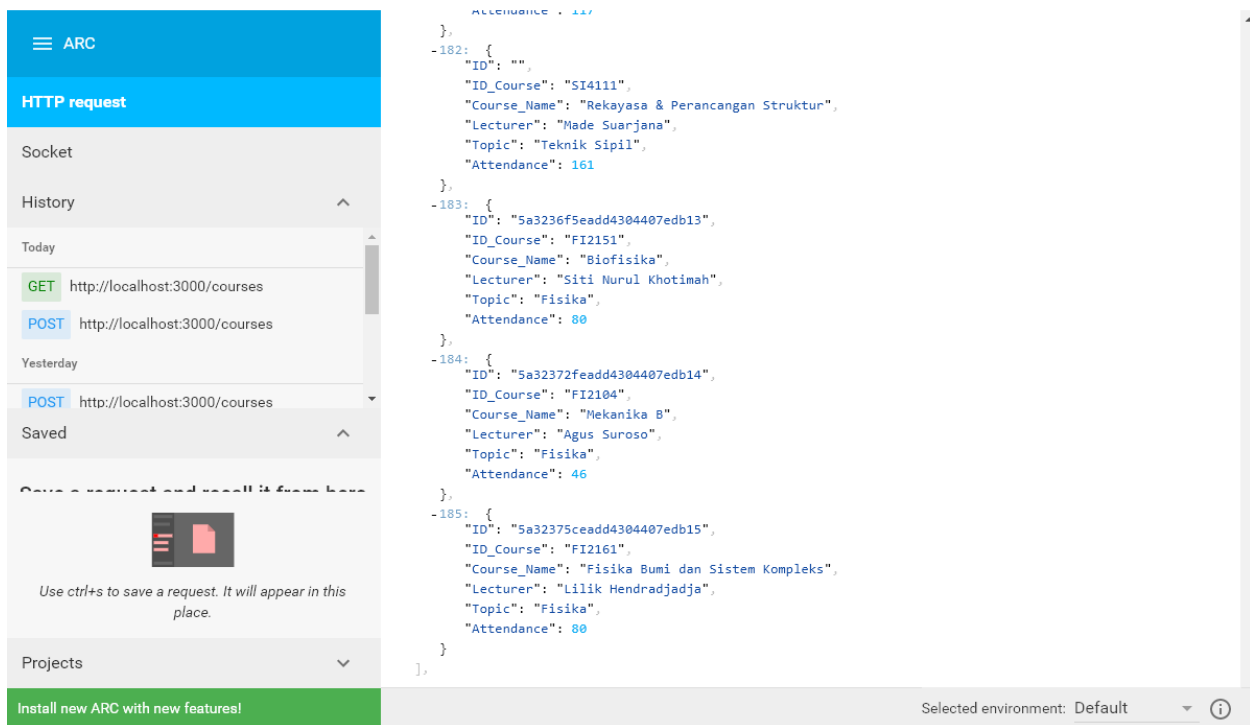
```
$ go build
```

```
$ ./courses
```

Akses dan Testing

Sistem hanya dapat diakses melalui jaringan internet ITB karena menggunakan server pada ITB.

1. Untuk mengakses sistem, gunakan 167.205.67.234:8080/courses
2. Pengetesan Get All (menggunakan Advance REST Client)



3. Pengetesan Get by ID

The screenshot shows the ARC REST client interface. On the left sidebar, the 'HTTP request' tab is selected, and the 'History' list shows a GET request to `http://localhost:3000/courses/II3160`. The main panel displays the details of this request. The 'Method' is 'GET' and the 'Request URL' is `http://localhost:3000/courses/II3160`. The 'Parameters' section is empty. The 'Headers' section shows a single header: 'Content-Type' with the value 'application/json'. Below the headers, a green status bar indicates '200 OK' with a response time of '13.00 ms'. The response body is a JSON object:

```
{
  "ID": "",
  "ID_Course": "II3160",
  "Course_Name": "Pemrograman Integratif",
  "Lecturer": "I Gusti Bagus Baskara Nugraha",
  "Topic": "Sistem dan Teknologi Informasi",
  "Attendance": 44
}
```

4. Pengetesan create course (POST)

The screenshot shows the ARC REST client interface. On the left sidebar, the 'HTTP request' tab is selected, and the 'History' list shows a POST request to `http://localhost:3000/courses`. The main panel displays the details of this request. The 'Method' is 'POST' and the 'Request URL' is `http://localhost:3000/courses`. The 'Parameters' section is empty. The 'Headers' section shows a single header: 'Content-Type' with the value 'application/json'. Below the headers, a green status bar indicates '201 Created' with a response time of '15.00 ms'. The response body is a JSON object:

```
{
  "ID": "5a3236f5eadd4304407edb13",
  "ID_Course": "FI2151",
  "Course_Name": "Biofisika",
  "Lecturer": "Siti Nurul Khotimah",
  "Topic": "Fisika",
  "Attendance": 80
}
```

5. Pengetesan update course (PUT)

Method

Request URL

PUT

http://localhost:3000/courses

SEND

Parameters ^

Headers

Body

Variables

Body content type

Editor view

application/json

Raw input

FORMAT JSON MINIFY JSON

```
2151","Course_Name":"Biofisika","Lecturer":"Siti Nurul Khotimah","Topic":"Fisika","Attendance":99}
```

FORMAT JSON MINIFY JSON

```
2151","Course_Name":"Biofisika","Lecturer":"Siti Nurul Khotimah","Topic":"Fisika","Attendance":99}
```

200 OK 215.00 ms

DETAILS v



```
{  
  "result": "success"  
}
```

6. Pengetesan Delete

The screenshot shows the ARC REST client interface. On the left, there's a sidebar with 'HTTP request' selected, a 'History' list containing several DELETE and GET requests, and a 'Saved' section. The main area is titled 'Request' and shows a DELETE method with the URL 'http://localhost:3000/courses'. The 'Body content type' is set to 'application/json' and the 'Editor view' is 'Raw input'. The request body contains a JSON object: `{304407edb14,"ID_Course":"FI2104","Course_Name":"Mekanika B","Lecturer":"Agus Suroso","Topic":"Fisika","Attendance":46}`. The interface also has tabs for 'Headers', 'Body', and 'Variables', and buttons for 'FORMAT JSON' and 'MINIFY JSON'.

FORMAT JSON MINIFY JSON

```
{304407edb14,"ID_Course":"FI2104","Course_Name":"Mekanika B","Lecturer":"Agus Suroso","Topic":"Fisika","Attendance":46}
```

200 OK 18.00 ms

DETAILS ▼

```
{
  "result": "success"
}
```


Link Penting

<http://167.205.67.234:8080/courses> (to check all)

http://167.205.67.234:8080/courses/{ID_Courses} (to check a course based on its ID)

Pengetesan menggunakan Advance REST Client dan Postman App