

PANDUAN



RCODE-X

APLIKASI REMOTE SCRIPT INTERPRETER BERBASIS WEB API

ABSTRACT

Buku ini berisi deskripsi tentang RCode-X, sebuah aplikasi/modul yang bekerja menggunakan teknologi web API, yang berfungsi untuk menjalankan kode program/script berbasis interpreter secara remote. Dalam buku panduan ini juga akan dibahas mekanisme implementasinya, yang meliputi cara instalasi dan penggunaannya.

Rosihan Ari Yuana

Email: rosihanari@staff.uns.ac.id

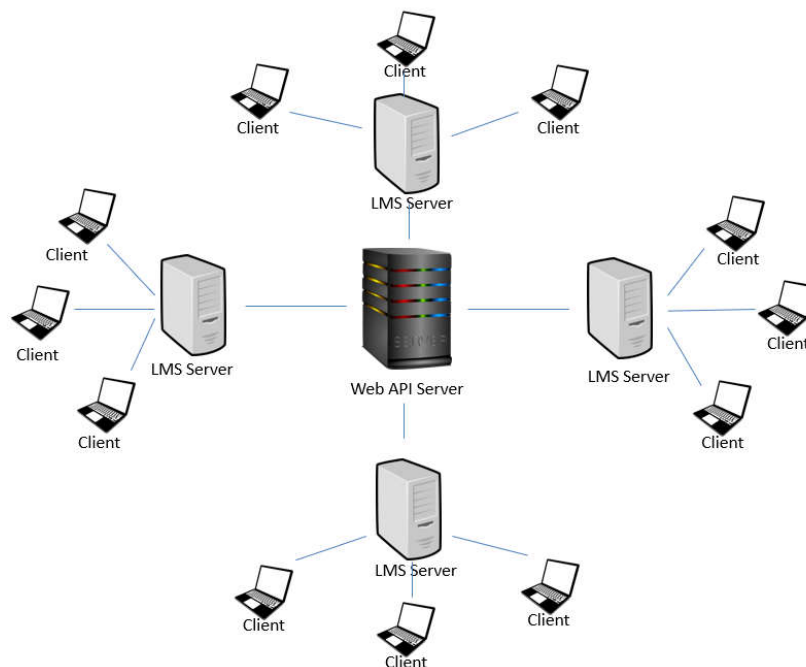
DAFTAR ISI

Deskripsi Modul	2
Spesifikasi Kebutuhan RCode-X	3
Spesifikasi Kebutuhan Server	3
Spesifikasi Kebutuhan LMS.....	3
Cara Instalasi RCode-X	4
Instalasi XAMPP/Apache	4
Instalasi Python	6
Instalasi dalam LMS	6
Cara Penggunaan RCode-X	7
Contoh Implementasi	8

RCode-X adalah modul yang digunakan untuk menjalankan script bahasa pemrograman yang berbasis interpreter secara *remote*. Kegunaan dari RCode-X adalah dapat diintegrasikan dengan Learning Management System (LMS), seperti Moodle, Atutor, Eliademy, Dokeos, dll. Bahkan selain itu RCode-X dapat diintegrasikan dengan LMS yang merupakan hasil *custom*. Setiap LMS yang terintegrasi dengan RCode-X akan dapat bermanfaat bagi pengguna LMS (khususnya siswa) dalam pembelajaran pemrograman komputer. Dengan adanya RCode-X ini dapat mendukung pembelajaran pemrograman komputer secara adaptif.

Modul RCode-X dikembangkan menggunakan bahasa PHP, dan saat ini baru mendukung untuk menjalankan script Python dan PHP secara *remote*.

Secara topologi, implementasi dari modul RCode-X digambarkan pada gambar berikut ini:



Pada gambar di atas, RCode-X diinstall di dalam sebuah Web API Server. Selanjutnya RCode-X ini dapat diintegrasikan dengan beberapa jenis LMS yang diinstal di dalam LMS Server. Dengan adanya integrasi ini, memungkinkan setiap pengguna dari LMS (*client*) dapat menjalankan kode program yang dibuatnya dalam proses pembelajaran pemrograman komputer. Kode program yang dikirim oleh client melalui LMS akan dijalankan oleh modul RCode-X. Kemudian, hasil atau output dari proses ini dikembalikan kepada client. Dengan demikian, client dapat melihat keakuratan hasil kode program yang dibuatnya.

Manfaat dari sistem di atas adalah efisiensi terhadap resource. Dalam hal ini, di setiap LMS server tidak perlu tersedia modul script interpreter sendiri-sendiri sehingga membutuhkan resource yang besar dan spesifikasi server yang tinggi.

SPESIFIKASI KEBUTUHAN RCODE-X

Pada bagian ini akan dijelaskan spesifikasi kebutuhan bagi server maupun LMS supaya RCode-X dapat diimplementasikan.

SPESIFIKASI KEBUTUHAN SERVER

Secara umum RCode-X tidak membutuhkan server berspesifikasi besar. Berdasarkan penelitian yang dilakukan oleh pengembang modul, resource yang dibutuhkan oleh RCode-X baik memory, CPU processing time, dan response timenya tidak begitu besar.

Berikut ini adalah spesifikasi kebutuhan minimum untuk server yang akan diinstal RCode-X:

- Sistem operasi: Linux, Windows
- RAM: min 2 GB
- Processor: dual core
- Disk space: 10 GB
- Software pendukung:
 - XAMPP atau Apache
 - Python 3.X
 - PHP 5.X

SPESIFIKASI KEBUTUHAN LMS

LMS yang dapat diintegrasikan dengan RCode-X tidak ada batasan tertentu, namun sangat disarankan LMS yang dikembangkan dengan menggunakan bahasa PHP.

CARA INSTALASI RCODE-X

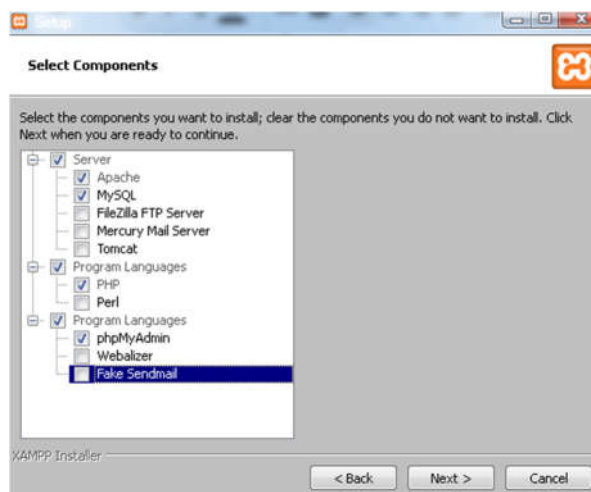
Pada bagian ini akan dijelaskan cara instalasi RCode-X pada server yang menggunakan sistem operasi Windows.

INSTALASI XAMPP/APACHE

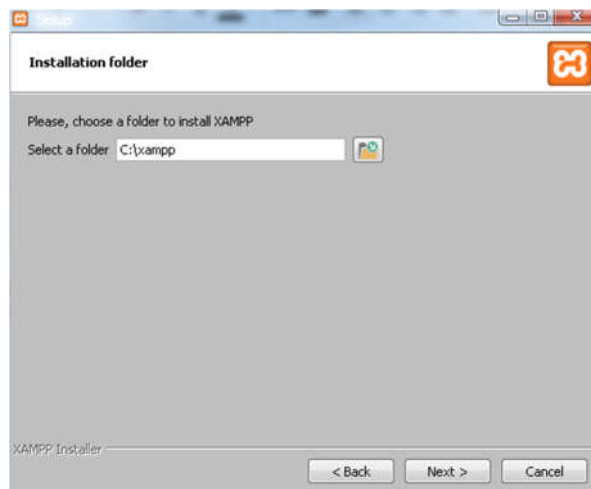
1. Download XAMPP melalui website Apache Friends (<https://www.apachefriends.org/index.html>)
2. Lakukan instalasi setelah Anda selesai mengunduh. Selama proses instalasi mungkin Anda akan melihat pesan yang menanyakan apakah Anda yakin akan menginstalnya. Silakan tekan Yes untuk melanjutkan instalasi.
3. Klik tombol Next.



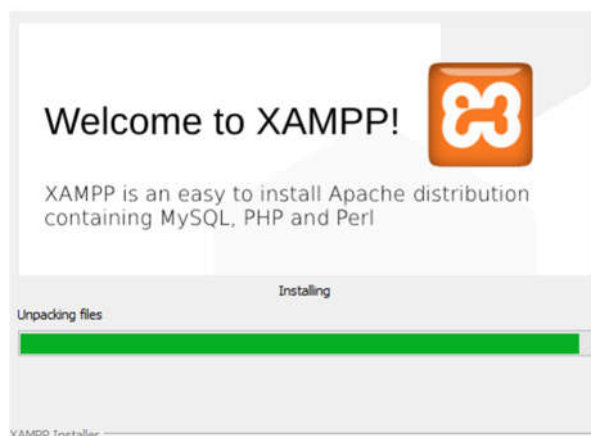
4. Pada tampilan selanjutnya akan muncul pilihan mengenai komponen mana dari XAMPP yang ingin dan tidak ingin Anda instal. Beberapa pilihan seperti Apache dan PHP adalah bagian penting untuk menjalankan website dan akan otomatis diinstal. Silakan centang MySQL dan phpMyAdmin, untuk pilihan lainnya biarkan saja.



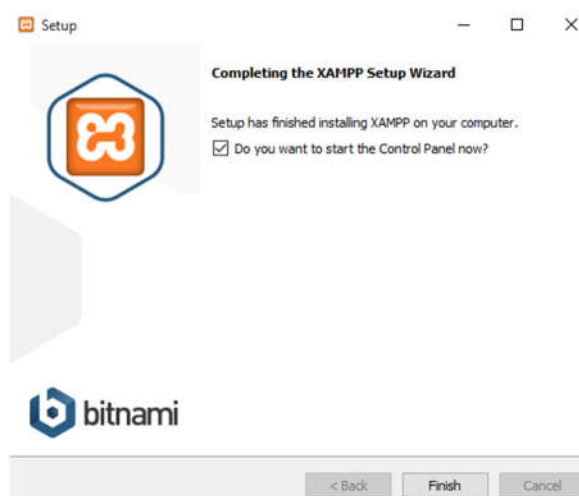
5. Berikutnya silakan pilih folder tujuan dimana XAMPP ingin Anda instal, pada tutorial ini pada direktori C:\xampp.



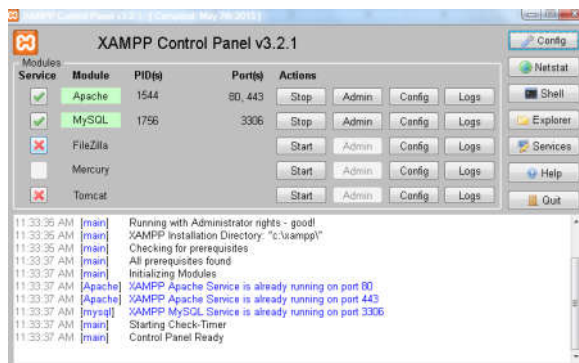
6. Pada langkah ini proses instalasi XAMPP akan dimulai. Silakan klik tombol Next.



7. Setelah berhasil diinstal, akan muncul notifikasi untuk langsung menjalankan control panel. Silakan klik Finish.



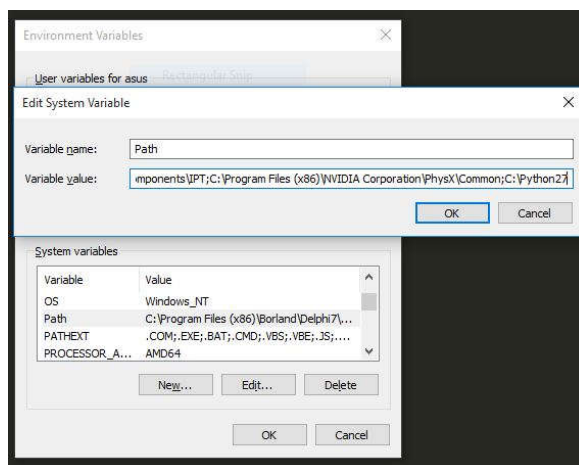
8. Silakan buka aplikasi XAMPP kemudian klik tombol Start pada Apache dan MySQL. Jika berhasil dijalankan, Apache dan MySQL akan berwarna hijau seperti gambar di bawah ini.



9. Selanjutnya semua script modul RCode-X yang ada di dalam folder **'server-api-modules'** dicopykan ke direktori C:\xampp\htdocs
10. Setelah XAMPP ini selesai diinstall, maka secara otomatis di dalam server juga telah terinstall PHP.

INSTALASI PYTHON

1. Unduh Python 3.X di <https://python.org>
2. Jalankan file instalasi Python yang telah diunduh. Secara default Python akan terinstall di folder C:\Python.
3. Selanjutnya mengatur variable environment, buka Control Panel->System and Security->System.
4. Klik Advanced system settings, Environment Variables.
5. Pada System variables, pilih Path, klik Edit.
6. Pada Variable value, tambahkan ;C:\Python.



INSTALASI DALAM LMS

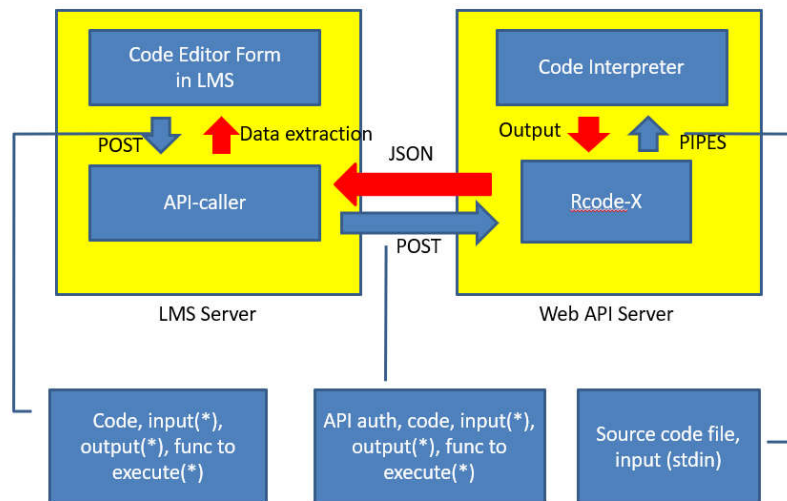
Setiap LMS yang akan diintegrasikan dengan RCode-X, di dalamnya harus sudah memiliki kedua file berikut ini:

- php-api-call.php
- python-api-call.php

Kedua file tersebut disarankan diletakkan di dalam direktori di mana aplikasi LMS tersebut terinstall. Kedua file tersebut digunakan untuk mengirim kode program dari LMS ke server RCode-X melalui API call.

CARA PENGGUNAAN RCODE-X

Pada bagian ini akan dijelaskan cara menggunakan RCode-X. Cara kerja dari RCode-X digambarkan melalui gambar berikut ini.



Berdasarkan gambar di atas, cara kerja RCode-X dimulai dari LMS server. Kode program yang dibuat oleh user/client melalui code editor akan disubmit ke modul API-Caller melalui mekanisme POST method. Parameter-parameter data yang disubmit ini terdiri dari kode program itu sendiri, input, output yang diharapkan, dan juga nama function yang akan dijalankan. Oleh API-Caller, selanjutnya data ini diteruskan kepada RCode-X untuk diproses. Melalui pipes, kode program yang diterima oleh RCode-X akan dijalankan (*running program*) oleh interpreter dalam hal ini PHP dan Python. Selanjutnya hasil running program yang berupa output ini akan dikirimkan kembali ke API-Caller yang ada di LMS server dalam bentuk JSON sebagai responnya. JSON ini nantinya akan diekstraksi oleh LMS untuk disajikan kepada user/client.

Adapun struktur JSON sebagai respon dari RCode-X adalah sebagai berikut:

```
{
  "output"      : ...,
  "errorStatus" : ...,
  ["errorType"  : ...],
  ["errorDetail": ...],
  ["outputStatus": ...]
}
```

Penjelasan field dari JSON:

- **output** : akan menampilkan output dari hasil eksekusi kode program
- **errorStatus** : bernilai 0 jika kode program tidak ada syntax error
 : bernilai 1 jika kode program ada syntax error
- **errorType** : akan menampilkan tipe syntax error (hasil ini akan muncul jika errorStatus = 1)
- **errorDetail** : akan menampilkan pesan syntax error secara penuh
 : (hasil ini akan muncul jika errorStatus = 1)
- **outputStatus** : bernilai 0 jika output running program tidak sesuai dengan yang diharapkan,
 : bernilai 1 jika output running program sesuai yang diharapkan

CONTOH IMPLEMENTASI

Pada bagian ini akan disajikan contoh implementasi penggunaan RCode-X di dalam LMS server. Untuk keperluan penulisan kode program, di LMS harus disediakan form, minimal dengan bentuk form HTML seperti berikut ini:

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>
    <h1>RCode-X Interpreter Sample</h1>

    <form method="post" action="python-api-call.php">
        <!-- form untuk menuliskan kode program -->
        <textarea name="code"></textarea>
        <!-- form untuk memberikan input -->
        <textarea name="input"></textarea>
        <!-- form untuk memberikan output yg diharapkan -->
        <textarea name="output"></textarea>
        <!-- form untuk menentukan nama function yg akan dijalankan -->
        <textarea name="callFunction"></textarea>

        <!-- tombol submit program -->
        <input type="submit" name="submit" value="Run Code">
    </form>

    <!-- bagian untuk menampilkan respons dari RCode-X -->
    <div id="output"></div>

    </form>
</body>
</html>
```

Berdasarkan contoh di atas, komponen yang harus ada di dalam form adalah sebagai berikut:

- Komponen dengan nama 'code'. Komponen ini digunakan untuk menuliskan kode program yang akan dijalankan
- Komponen dengan nama 'input'. Komponen ini digunakan untuk menuliskan input kode program ketika dijalankan. Input bersifat optional (boleh diisi atau tidak)
- Komponen dengan nama 'output'. Komponen ini digunakan untuk menuliskan output kode program ketika dijalankan. Output bersifat optional (boleh diisi atau tidak). Nilai output ini nanti digunakan untuk

menjustifikasi kesesuaian antara output program ketika dijalankan dengan output yang diharapkan. Hal ini ditujukan untuk mengecek benar tidaknya program yang dibuat.

- Komponen dengan nama 'callFunction'. Komponen ini nantinya digunakan untuk menuliskan nama function yang akan diuji ketika running program. Sifat dari callFunction ini optional. Fitur ini nanti hanya bisa digunakan untuk kode program Python.

Keterangan:

Kode form HTML di atas digunakan untuk menjalankan kode program Python. Sedangkan untuk menjalankan kode program PHP, pada bagian action diganti dengan '**php-api-call.php**'.

Berikut ini adalah tampilan contoh kode program Python yang ditulis melalui form HTML melalui LMS.

```
1 def myFunction():
2     print('Hello World', end='')
3
4 myFunction()
```

Run Code

Apabila diklik tombol Run Code, maka selanjutnya akan muncul response dalam bentuk JSON yang menunjukkan output program setelah dijalankan (gambar di bawah ini).

```
1 def myFunction():
2     print('Hello World', end='')
3
4 myFunction()
```

Run Code

```
stdClass Object
(
    [output] => Hello World
    [errorStatus] => 0
)
```

Berdasarkan respon yang muncul dari gambar di atas, tampak bahwa outputnya adalah 'Hello World', dengan erroStatus bernilai 0. Maksudnya adalah bahwa tidak ada error ketika program dijalankan.

Sedangkan pada contoh berikut ini, akan ditunjukkan bentuk respon dari RCode-X jika ada kesalahan sintaks dalam penulisan kode program.

```

1 def myFunction():
2     print('Hello World, end='')
3
4 myFunction()

```

Run Code

Bentuk respon RCode-X dari hasil running program kode di atas adalah sebagai berikut:

```

1 def myFunction():
2     print('Hello World, end='')
3
4 myFunction()

```

Run Code

```

stdClass Object
(
  [output] => File "temp_files/python-files/6fidjcrglj74qjbd1m8qec2sv4_temp.py", line 2
    print('Hello World, end='')
              ^
SyntaxError: EOL while scanning string literal

  [errorStatus] => 1
  [errorType] => SyntaxError
  [errorDetail] => SyntaxError: EOL while scanning string literal
)

```

Berdasarkan respon JSON yang diterima oleh LMS dari RCode-X tersebut, akan ditunjukkan pada baris mana dari kode program yang salah. Selain itu keterangan errornya, yang meliputi jenis error dan penjelasan detil errornya, juga ditunjukkan.

Selanjutnya akan ditunjukkan contoh implementasinya untuk mengecek keakuratan dari output program. Pada contoh berikut ini, pengguna LMS diminta membuat program untuk menjumlahkan dua bilangan yaitu bil1 = 10 dan bil2 = 30. Oleh karena itu untuk menguji benar tidaknya kode program yang dibuat perlu menambahkan output pada form HTML nya. Dalam hal ini diset output = 40.

```

1 # program menjumlahkan 2 bilangan
2
3 bil1 = 10
4 bil2 = 30
5
6 hasil = bil1 + bil2
7
8 print(hasil)

```

Run Code

Setelah kode program dijalankan oleh RCode-X akan muncul respon seperti di bawah ini:

```
1 # program menjumlahkan 2 bilangan
2
3 bil1 = 10
4 bil2 = 30
5
6 hasil = bil1 + bil2
7
8 print(hasil)
```

Run Code

```
stdClass Object
(
  [output] => 40
  [errorStatus] => 0
  [outputStatus] => 1
)
```

Berdasarkan respon di atas, tampak bahwa nilai outputStatus = 1 yang artinya adalah program benar yaitu menghasilkan output seperti yang diharapkan.

Selanjutnya adalah contoh implementasi kode program yang di dalamnya ada input.

```
1 # program menjumlahkan 2 bilangan
2
3 bil1 = int(input('Masukkan bil pertama: '))
4 bil2 = int(input('Masukkan bil kedua: '))
5
6 hasil = bil1 + bil2
7
8 print('Hasilnya: ', hasil)
```

Run Code

Tampilan respon JSON yang diterima LMS apabila diberikan input 40 dan 20 adalah seperti pada gambar di bawah ini.

```
1 # program menjumlahkan 2 bilangan
2
3 bil1 = int(input('Masukkan bil pertama: '))
4 bil2 = int(input('Masukkan bil kedua: '))
5
6 hasil = bil1 + bil2
7
8 print('Hasilnya: ', hasil)
```

Run Code

```
stdClass Object
(
  [output] => Masukkan bil pertama: 40
  Masukkan bil kedua: 20
  Hasilnya: 60
  [errorStatus] => 0
  [outputStatus] => 0
)
```