

KONSEP MEMBANGUN APLIKASI *FRONT END* BERBASIS *WEB APPML* (*APPLICATION MODELING LANGUAGE*)

Joko Triyono¹⁾, Prita Haryani²⁾

^{1,2)}Jurusan Teknik Informatika, Fakultas Teknologi Industri, IST AKPRIND Yogyakarta
Email: ¹⁾jack@akprind.ac.id, ²⁾printaharyani@akprind.ac.id

Abstrak

Teknologi web akhir-akhir ini berkembang sangat pesat, setelah HTML 5 di luncurkan, maka terjadi banyak perubahan dalam teknik pemrograman berbasis web, APPML (Application Modeling Language) merupakan salah satu teknik pemrograman yang berbasis pada HTML5. Semakin banyak web di bangun maka akan semakin banyak pula data beredar di dunia maya. Dan mau tidak mau, sebuah informasi akan menjadi referensi bagi informasi yang lain dan begitu seterusnya sehingga informasi akan menjadi sebuah mata rantai yang saling kait mengkait satu sama lain. Walaupun sistem informasi telah online selama 24 jam dan bisa diakses dari mana saja selama ada koneksi internet tetapi akan menjadi tidak efektif dan efisien jika harus mengunjungi masing-masing sistem informasi yang ada untuk mendapatkan informasi yang dibutuhkan. Maka diperlukan sebuah metode atau teknik dalam membangun aplikasi front end dengan melakukan koleksi data dari berbagai sumber sistem informasi yang relevan, dengan menanamkan service server dan service client sebagai penghubung antar sistem dengan dukungan JSON, XML. Dengan teknik APPML (Application Modeling Language) yang diterapkan pada sebuah aplikasi front end berbasis html 5 tanpa melakukan koneksi database secara langsung, tetapi cukup memanggil service berbasis JSON maka akan di peroleh data atau informasi yang dibutuhkan tanpa harus mengunjungi sistem informasi yang ada secara langsung.

Keywords: appml, json, service, xml

1. PENDAHULUAN

Teknologi web akhir-akhir ini berkembang sangat pesat, setelah HTML 5 di luncurkan dengan dikeluarkannya rekomendasi dari W3C pada tahun 28 Oktober 2014, maka terjadi banyak sekali perubahan dalam teknik pemrograman berbasis web, APPML (*Application Modeling Language*) merupakan salah satu teknik pemrograman yang berbasis pada HTML5. Semakin banyak web di bangun maka akan semakin banyak pula data beredar di dunia maya, seiring dengan meningkatnya kebutuhan masyarakat akan informasi, baik itu hanya sebagai berita maupun informasi sebagai kebutuhan pokok dalam menjalankan kegiatan harian. Dan mau tidak mau, sebuah informasi akan menjadi referensi bagi informasi yang lain dan begitu seterusnya sehingga informasi akan menjadi sebuah mata rantai yang saling kait mengkait satu sama lain. Walaupun sistem informasi telah *online* selama 24jam dan bisa diakses dari mana saja selama ada koneksi internet tetapi akan menjadi tidak efektif dan efisien jika harus mengunjungi masing-masing sistem informasi yang ada untuk mendapatkan informasi yang dibutuhkan.

Maka diperlukan sebuah metode atau teknik dalam membangun aplikasi *front end*

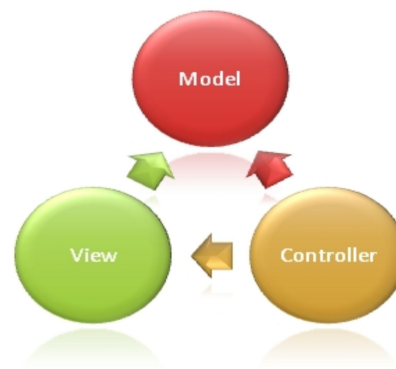
dengan melakukan koleksi data dari berbagai sumber sistem informasi yang relevan sehingga sebuah aplikasi *front end* akan membantu *end user* untuk melihat informasi secara cepat dan *live*.

Dalam sebuah penelitian tentang pengembangan sistem disimpulkan bahwa sistem yang difokuskan dalam mencari kriteria yang sesuai untuk pemesanan boarding untuk kamar asrama bisa dilakukan dengan sistem aplikasi [5]. Keberhasilan dalam mengimplemen-tasikan sistem ini adalah dapat bekerja dengan baik untuk membantu siswa dalam mencari lokasi tempat tinggal yang tepat. Dari pencarian, arahan, hingga pemesanan secara online dan respon fasilitas sistem dengan menggunakan SMS (*Short Message Service*) kepada pemilik kos yang memudahkan proses pemesanan. Dalam penelitian yang berbeda di peroleh hasil tentang pengembangan sebuah *prototype* sistem informasi yang di kombinasikan dengan jejaring sosial *twitter*, dimana jejaring *twitter* digunakan petani untuk melaporkan semua kegiatannya ke sistem informasi, dengan menggunakan fasilitas APIs (*Application Programming Language*) maka informasi yang masuk akan di kirimkan ke sistem informasi dengan menggunakan *account* dari *twitter* pengirim. Dengan metode ini secara

teknologi dan biaya petani tidak mengalami kesulitan dalam melaporkan kegiatannya, sedangkan dari sisi investor akan bisa melihat perkembangan investasinya [7]. Juga pada penelitian lain tentang komunikasi data dihasilkan bahwa penerapan sistem pengendali dan pemantau kendaraan jarak jauh pada penelitian ini diimplementasikan dengan membuat *prototype* alat *IoT (Internet of Things)* menggunakan *Raspberry Pi 3 Model B*, modem GSM, GPS USB VK-172 Glonass, dan relay DC [6]. Perangkat *IoT* akan dilengkapi aplikasi yang akan dikoneksikan dengan *server Firebase* dan selanjutnya dipasang pada kendaraan. Jika kendaraan dihidupkan maka alat *IoT* akan mengirimkan data posisi kendaraan dari satelit GPS ke server *Firebase*. Penggunaan *server Firebase* dipilih karena mendukung pemrograman *socket*, memiliki pustaka yang lengkap untuk berbagai *platform web* dan piranti bergerak, dapat digabungkan dengan berbagai *framework*, dan menyediakan layanan *DbaaS (Database as a Service)*. Hasilnya, data posisi kendaraan yang diterima server, secara langsung (*realtime*) akan ditampilkan melalui aplikasi web sistem pengendali dan pemantau jarak jauh dengan tampilan visualisasi *Google Maps* dan tombol *lock/unlock* untuk menghidupkan dan mematikan/mengunci kendaraan. Tentang kepuasan pengguna *website* disimpulkan bahwa dari hasil uji hipotesis menunjukkan bahwa dimensi *citizen support, content and appearance of information, reability, functionality of the interaction environment, trust, dan dimensi ease of use* berpengaruh terhadap kualitas layanan *e-government* [3]. Kualitas layanan *e-government* berpengaruh terhadap kepuasan pengguna *website*, kualitas layanan *e-government* berpengaruh terhadap intensitas pengguna *website* dan kepuasan pengguna *website* berpengaruh terhadap intensitas penggunaan *website* Pemerintah Kota Yogyakarta. Dari beberapa penelitian yang telah ada di dapat disimpulkan bahwa metode penampilan maupun pengiriman informasi masih terkait dengan sebuah aplikasi tertentu, atau tersaji pada sebuah aplikasi. Kelengkapan informasi di *website* akan meningkatkan tingkat kepuasan pengguna, sehingga di harapkan tingkat kunjungan *website* akan meningkat.

Dalam penelitian lain tentang *AppML* menyimpulkan bahwa *APPML* adalah sebuah teknik yang patut di perhitungkan untuk memangkas waktu pengembangan aplikasi web, yang dengan mudah bisa dikombinasikan dengan

arsitektur aplikasi web serta memperluas kemampuan yang ada pada *HTML* [4]. *AppML* adalah singkatan dari *Application Modeling Language*, berjalan di halaman *HTML* manapun serta tidak perlu instalasi [2]. *AppML* adalah alat untuk membawa data ke aplikasi *HTML* baik itu *object*, file maupun database. *AppML Architecture* dalam dokumen resmi pada *w3schools* dijelaskan bahwa sebagai sebuah Arsitektur Web Modern dengan menggabungkan teknik dan gagasan terkini tentang pengembangan *web modern* [1]. *AppML* fokus pada kecepatan, kesederhanaan, dan biaya rendah meliputi : Arsitektur *MVC* Berorientasi Layanan, Konsumsi *Bandwidth* Sangat Rendah, Optimalisasi untuk *Cloud Computing*, Pemisahan Penuh Konten dari Presentasi, Pemrograman Deklaratif Cerdas, Perkembangan *Web Development* yang cepat, Sangat *Scalable* dan *Testable*, Mudah Konfigurasi dan Rekonfigurasi, Dukungan Pintar untuk Akun dan Peran Pengguna. *AppML* menggunakan arsitektur *MVC* seperti pada Gambar 1.



Gambar 1. Arsitektur MVC

Model mendeskripsikan aplikasi, Model tidak memperhatikan presentasi atau *user interface (UI)*, model diimplementasikan dengan *JSON*.

```

{"rowsperpage" : 10,
 "database" : {
   "connection" : "localmysql",
   "sql" : "SELECT CustomerName, City, Country
FROM Customers",
   "orderby" : "CustomerName"
 },
 "filteritems" : [
  {"item" : "CustomerName", "label" : "Customer"},
  {"item" : "City"},
  {"item" : "Country"}
 ],
 "sortitems" : [

```

```
{"item" : "CustomerName", "label" : "Customer"},  
{ "item" : "City"},  
{ "item" : "Country"}}
```

View hanya sebuah *script HTML*, sebagai *userinterface* untuk menginputkan maupun menampilkan data.

```
<!DOCTYPE html>  
<html lang="en-US">  
<title>Customers</title>  
<link rel="stylesheet" href="/w3css/4/w3.css">  
<script src="/appml/2.0.3/appml.js"></script>  
<body>  
<div class="w3-container" appml-  
data="local?model=model_customers">  
<h1>Customers</h1>  
<div appml-include-  
html="inc_listcommands.htm"></div>  
<div appml-include-html="inc_filter.htm"></div>  
<table class="w3-table-all">  
<tr>  
<th>Customer</th>  
<th>City</th>  
<th>Country</th>  
</tr>  
<tr appml-repeat="records">  
<td>{{CustomerName}}</td>  
<td>{{City}}</td>  
<td>{{Country}}</td>  
</tr>  
</table>  
</div>  
</body>  
</html>
```

Sedangkan *Controller* sebagai *script* dari *client* dan atau *server*, pada sisi *client script web* ini akan mengontrol aplikasi untuk menampilkan data sebagaimana didefinisikan dalam model dan atribut *HTML*, dapat berjalan independen dari *HTML* (tersem-bunyi) dan bisa meminta data model dari server web (server *SQL*), Edit data maupun mengirimkan data ke *server web*. Sedangkan Skrip *server* dapat mengontrol aplikasi dengan menerima permintaan dari klien (*browser*), mengembalikan, menerima data dari klien dan memperbarui data di *server* serta menjaga autentikasi dan keamanan.

2. METODE PENELITIAN

Metode penelitian dilakukan dalam skala laboratorium di Laboratorium Jaringan dan Internet Institut Sains dan Teknologi AKPRIND Yogyakarta, data atau model yang digunakan dalam penelitian ini adalah Aplikasi Sistem Informasi, *Service Server*, *Service Client*, *APPML* yang dari beberapa sumber tersebut di olah sedemikian rupa sehingga bisa menjelaskan penelitian yang dimaksud.

Pengembangan aplikasi menggunakan metode pengembangan *waterfall*. Metode *waterfall* ini dibangun oleh *Winston W. Royce* pada tahun 1970 untuk menggambarkan praktik produk perangkat lunak. Model *waterfall* terdiri dari lima fase yaitu *analysis*, *design*, *implementation*, *testing* dan *maintenance*. Setiap fase pada model *waterfall* dapat dilihat pada Gambar 2.

a. Analysis

Pada tahap analisis dilakukan proses pengumpulan data untuk mendapatkan informasi mengenai perangkat lunak yang akan dikembangkan sesuai dengan kebutuhan pengguna. Teknik pengumpulan data untuk tahap analisis ini bisa dengan observasi, wawancara maupun kuesioner.

b. Design

Tahap *design* digunakan untuk menerjemahkan kebutuhan perangkat lunak dari tahap analisis ke desain. Desain yang dibuat berupa sketsa desain aplikasi dengan menggunakan *UML*.

c. Implementation

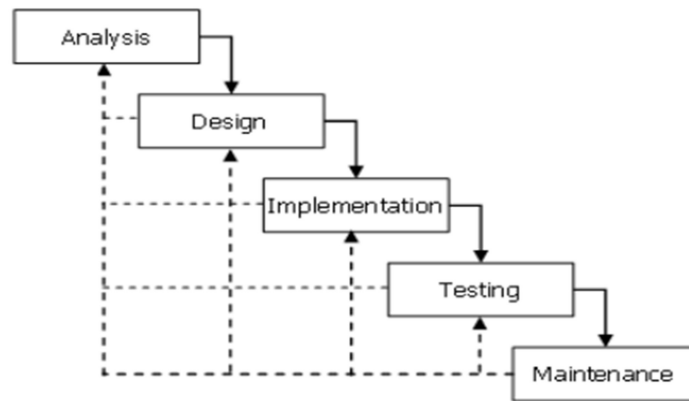
Setelah desain dibuat, tahap selanjutnya yaitu membuat atau mengembangkan aplikasi perangkat lunak.

d. Testing

Pada tahap *testing*, verifikasi dan validasi aplikasi perangkat lunak dilakukan. Tahap *testing* digunakan untuk menemukan kesalahan atau *debug* pada saat pembuatan aplikasi. Tahapan pengujian secara keseluruhan mencakup pengujian unit, pengujian integrasi, pengujian sistem, dan pengujian penerimaan.

e. Maintenance

Tahap terakhir dari model *waterfall* ini adalah tahapan *maintenance*. Proses instalasi produk dan pemeliharaan dilakukan dalam tahap *maintenance*.



Gambar 2. Fase penelitian pada model *waterfall*

Kebutuhan Sistem

Bahan dan alat yang dibutuhkan untuk penelitian ini meliputi hardware dan software, diantaranya yaitu:

1. Hardware, laptop Lenovo dengan spesifikasi Intel® Core™ i5-5200U CPU @ 2.20GHz × 4 Ram 7,7 GiB 64bit Hardisk 500GB.
2. Software Sistem operasi Ubuntu 16.04 LTS.
3. Software Web Server, PHP
4. Database MySQL
5. Perangkat jaringan dan server *public* di Laboratorium Komputer Jaringan dan Internet.

Metode Pengumpulan Data

Metode yang digunakan dalam pengumpulan data pada penelitian ini terdiri dari beberapa metode, yaitu:

Metode Observasi, Metode observasi ini digunakan untuk pengumpulan data dengan pengamatan secara langsung maupun tidak langsung terhadap obyek yang diteliti.

Metode Studi Kepustakaan, Metode studi kepustakaan merupakan sebuah cara dalam pengumpulan data dengan mempelajari bahan pustaka baik berupa dokumen tertulis ataupun berupa gambar dengan membandingkan beberapa referensi.

Metode Eksperimen, Metode ini digunakan dengan mengadakan uji coba dan simulasi yang telah dibuat menggunakan dua

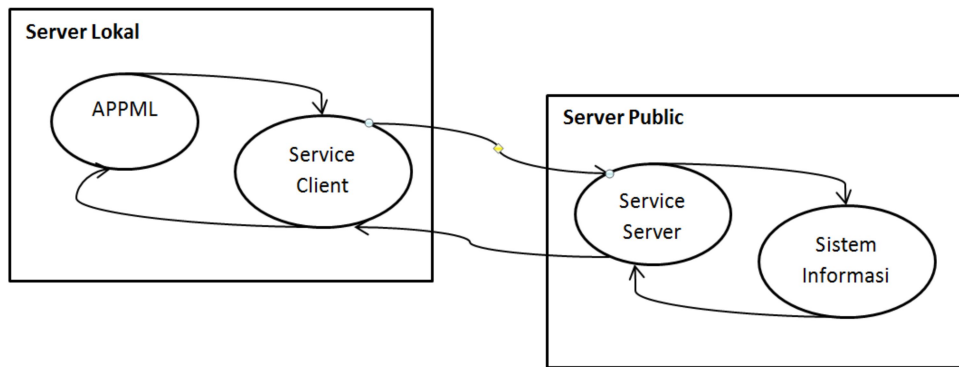
buah Web Server, satu web server di server Public milik Laboratorium Jaringan dan Internet (untuk selanjutnya di sebut Server Public) yang di isi sebuah sistem informasi web online berbasis PHP dan MySQL, sedangkan web server kedua di letakkan di *localhost* /Laptop (yang untuk selanjutnya di sebut server lokal) yang di isi sebuah aplikasi berbasis web HTML, PHP dan MySQL.

Peraturan Bisnis

Peraturan bisnis pada obyek penelitian adalah bahwa server public adalah server sistem aplikasi yang telah berjalan, sedangkan server lokal adalah sebuah sistem informasi yang akan mengambil data dari server public melalui service yang disediakan di server public yang kemudian di tampilkan di server lokal menggunakan metode APPML. Service Client pada server lokal akan memanggil service server di server public menggunakan JSON, sedangkan aplikasi APPML di server lokal akan memanggil service client dan akan menerima data berbentuk XML.

Desain Sistem

Berdasarkan peraturan bisnis yang telah dijelaskan, maka dalam tulisan ini di implementasikan dalam sebuah gambar design seperti yang di tunjukkan pada Gambar 3.



Gambar 3. Desain Sistem

Dalam penelitian ini akan di bangun:

1. Sebuah sistem informasi di *server* lokal
2. *Service server* di *server public*
3. *Service client* di server lokal

Dari gambar 3 bisa dijelaskan sebagai berikut:

1. Sistem aplikasi pada server lokal akan menampilkan data berbasis HTML yang akan mengambil data melalui APPML pada server lokal.
2. APPML sendiri akan mendapatkan data dari sebuah service client yang ada di server lokal berbasis Server Side dalam hal ini PHP yang berbentuk array assosiativ dan diubah jadi XML.
3. Service Client berbasis server side php akan memanggil sebuah service server di server public melalui JSON dan akan mendapatkan data berbentuk JSON dan diubah jadi array assosiativ.
4. Service Server berbasis server side php akan membaca data pada sistem informasi, baik itu fungsi atau koneksi database yang telah di sediakan dan akan mendapatkan data berbentuk array assosiativ dan di ubah jadi JSON untuk dikirimkan ke Service Client.

Perancangan Sistem

1. Sistem Informasi Aplikasi Lokal

Akan menampilkan data mahasiswa dengan format tabel Nomor Urut, Nomor Mahasiswa, Nama Mahasiswa dan Jurusan. Berikut ini adalah listing aplikasi di sisi APPML mhs1.html.

```

1. <head>
2. <script src="appml.js"></script>
3. <title>DATA MAHASISWA VIA APPML</title>
4. </head>
5. <body>
6. <div align=center><h1>DATA MAHASISWA</h1>
7. <div appml-
  data="http://localhost/lh/service/service.php">

```

```

8. <table border=1 cellpadding="0"
  cellspacing="0" width="100%">
9. <tr>
10. <td width="5%">Nu</td><td
  width="10%">NIM</td><td
  width="75%">NAMA MAHASISWA</td><td
  width="10%">Kode Jur</td>
11. </tr>
12. <tr appml-repeat="records">
13. <td
  align=center>{{nu}}</td><td>{{nim}}</td><td>{{
  nama}}</td><td>{{kodejur}}</td>
14. </tr>
15. </table>
16. </div>
17. </div>
18. </body>

```

Pada listing *mhs1.html* bisa dijelaskan sebagai berikut:

- Pada baris 2. *appml.js*, adalah fungsi utama APPML yang digunakan sebagai bagian utama dari sistem berbasis APPML.
- Pada baris 7. Adalah menyatakan n atau mengarahkan bahwa data APPML di ambil dari sumber di alamat tersebut, bisa localhost maupun alamat publik.
- Pada baris 12. *appml-repeat="records"* sebagai loop data dari *records* hasil dari pemanggilan baris 7 dengan format JSON.
- Pada baris 13. Adalah proses menampilkan elemen data dari JSON menggunakan *{{nim}}* dan seterusnya sesuai dengan format data dari JSON.

2. Sistem Service Client

Sistem *Service Client* adalah aplikasi *server side* berbentuk PHP yang dipanggil oleh aplikasi lokal dan akan mengirimkan data ke *service server*. Berikut ini adalah listing aplikasi di sisi *service client service.php*

```

1. <?php
2. $url="http://labora.lab.akprind.ac.id/service
   /service.php";
3. $jsonfile = file_get_contents($url);
4. echo "{ \"records\" : \".$jsonfile.\"}";
5. ?>

```

Pada listing `service.php` bisa dijelaskan sebagai berikut:

Pada baris 2, `$url` adalah alamat dari *service server* yang akan diambil datanya.

Pada baris 2. `file_get_content` adalah proses untuk mengambil data pada *service server*.

Pada baris 3. Data hasil `file_get_content` berbentuk JSON di tambahkan "`records :`" agar format datanya standart dengan aplikasi APPML.

3. Service Server

Service Server adalah aplikasi di ranah publik berbasis *server side* yang akan berhubungan langsung dengan sistem informasi maupun database, dan bertugas untuk memberikan informasi kepada *service client* dalam format JSON. Berikut ini adalah listing aplikasi *service server service.php*.

```

1. <?php
2. include_once "buka.php";
3. $sql="select nim,nama,kodejur from alum
   limit 10";
4. $query = $db->prepare($sql);
5. $uu=$query->execute();
6. $i=0;
7. $arr = array();
8. $data = $query->fetchAll();
9. foreach ($data as $r) {

```

Web <http://labora.lab.akprind.ac.id> adalah *web public* milik laboratorium komputer yang menampung dan mengelola data pendaftaran praktikum, di dalam database sistem ini terdapat tabel *mahasisw* yang menampung data

```

10. $i++;
11. $tmp=array("nu" => $i,"nim" =>
    $r['nim'], "nama" => $r['nama'], "kodejur" =>
    $r['kodejur'] );
12. array_push($arr, $tmp);
13. }
14. $jsonfile = json_encode($arr);
15. echo $jsonfile;
16. ?>

```

Pada listing `service.php` bisa dijelaskan sebagai berikut:

- Pada baris 1 sd 9, adalah proses koneksi database dan pembacaan data.
- Pada baris 11, adalah pemasukan data rekord kedalam sebuah `$tmp` dengan format *array assosiative*.
- Pada baris 12, adalah pemasukan data `$tmp` kedalam array `$arr`.
- Pada baris 14, adalah proses pengubahan data dari *array assosiative* ke dalam format JSON.
- Pada baris 15, adalah meng-outputkan data JSON ke peminta informasi.

3. HASIL DAN PEMBAHASAN

Aplikasi yang menggunakan APPML ini perlu di ujicoba untuk mengetahui sejauh mana informasi ini bisa di berikan kepada *end user* sesuai dengan yang di rencanakan atau diinginkan.

Dari hasil pengujian terhadap web praktikum di laboratorium komputer di dapatkan hasil yang memuaskan, di peroleh bahwa output aplikasi berbasis APPML ini seperti pada Gambar 4 tentang hasil aplikasi APPML.

mahasiswa yang mengikuti praktikum. Gambar 5 menunjukkan salah satu tampilan web `labora.lab.akprind.ac.id`, sedangkan pada Gambar 6 menunjukkan tabel *mahasisw* di lihat dari *tools phpmyadmin*.

| Nu | NIM | NAMA MAHASISWA | Kode Jur |
|----|-----------|------------------------------|----------|
| 1 | 111051005 | SAMUEL KAREL LABOK | 05 |
| 2 | 111051011 | FARIZ SATYA WIRANTO | 05 |
| 3 | 111051023 | Yusuf Febriyanto | 05 |
| 4 | 111051028 | Vera Laurensiana Kocu | 05 |
| 5 | 111051063 | Syifa Nurul Huda | 05 |
| 6 | 111051074 | FUAD AZIZ | 05 |
| 7 | 111051075 | Agostinho Fernandes Da Silva | 05 |
| 8 | 111051092 | Agradira Dwi Wahyuda | 05 |
| 9 | 112053006 | Permaedi Yulianto | 05 |
| 10 | 121051001 | Frederico F.e. Da Cunha | 05 |

Gambar 4. Hasil Aplikasi APPML

SIM-LABORATORY IST-AKPRIND
Sistem Informasi dan Manajemen Laboratorium

Tanggal: Tuesday, 26 - Sep - 2017 Masa Registrasi : 17.1 Semester GANJIL Tahun Akademik 2017/2018

Home
Login
Login for Staff
Menu Utama :
Laboratorium
Mata Praktikum
Asisten
Jadwal Praktikum
Laporan Praktikum
Pedoman Laporan

FOLLOWER IS DEATH -- LAB BASE LABORATORY

LAPORAN AKHIR PRAKTIKUM (KEHADIRAN DAN NILAI AKHIR) [CETAK (nomhs)] [CETAK (nama)]

Mata Praktikum : (MIP 2402)Jaringan Komputer Jurusan 35
Di Ruang Komputer Jaringan , Laboratorium Komputer Jaringan

| Nu | Nomhs | Nama Mahasiswa | Kehadiran | Nilai Akhir | Shift |
|----|-----------|----------------------------|------------|-------------|-------|
| 1 | 151351001 | Fathur Rahman Afrudin | 0 (0,00 %) | 0 | KM-2 |
| 2 | 151351006 | Wahidiansyah Gusdi Handika | 0 (0,00 %) | 0 | KM-2 |
| 3 | 161351001 | Andreas Syah Akbar Hidayat | 0 (0,00 %) | 0 | KM-2 |
| 4 | 161351002 | Novia Pelita Harani | 0 (0,00 %) | 0 | KM-2 |
| 5 | 161351005 | Dyan Lutfi | 0 (0,00 %) | 0 | KM-2 |
| 6 | 161351007 | Adhitya Putra Utama | 0 (0,00 %) | 0 | KM-2 |
| 7 | 161357003 | Putri Nida Ulfa | 0 (0,00 %) | 0 | KM-2 |

Gambar 5. Tampilan web <http://labora.lab.akprind.ac.id>

Server: localhost Database: labora Table: mahasiswa

| nomhs | nama | jalur | jur | ang |
|-----------|------------------------------|-------|-----|------|
| 111051005 | SAMUEL KAREL LABOK | 4 | 05 | 2011 |
| 111051011 | FARIZ SATYA WIRANTO | 5 | 05 | 2011 |
| 111051023 | Yusuf Febriyanto | 5 | 05 | 2011 |
| 111051028 | Vera Laurensiana Kocu | 4 | 05 | 2011 |
| 111051063 | Syifa Nurul Huda | 4 | 05 | 2011 |
| 111051074 | FUAD AZIZ | 4 | 05 | 2011 |
| 111051075 | Agostinho Fernandes Da Silva | 4 | 05 | 2011 |
| 111051092 | Agradira Dwi Wahyuda | 5 | 05 | 2011 |
| 112053006 | Permadi Yulianto | 3 | 05 | 2011 |
| 121051001 | Frederico F.e. Da Cunha | 4 | 05 | 2012 |
| 121051005 | Yosi Iman Setiadi | 4 | 05 | 2012 |

Gambar 6. Tampilan tabel mahasiswa melalui *phpmyadmin*

Dari pengujian di sisi *service server* <http://labora.lab.akprind.ac.id/service/service.php> didapatkan hasil sebagai berikut:

```
{{"nu":1,"nim":"111051005","nama":"SAMUEL KAREL LABOK","kodejur":"05"}, {"nu":2,"nim":"111051011","nama":"FARIZ SATYA WIRANTO","kodejur":"05"}, {"nu":3,"nim":"111051023","nama":"Yusuf Febriyanto","kodejur":"05"}, {"nu":4,"nim":"111051028","nama":"Vera Laurensiana Kocu","kodejur":"05"}, {"nu":5,"nim":"111051063","nama":"Syifa Nurul Huda","kodejur":"05"}, {"nu":6,"nim":"111051074","nama":"FUAD AZIZ","kodejur":"05"}, {"nu":7,"nim":"111051075","nama":"Agostinho Fernandes Da Silva","kodejur":"05"}, {"nu":8,"nim":"111051092","nama":"Agradira Dwi Wahyuda","kodejur":"05"}, {"nu":9,"nim":"112053006","nama":"Permadi Yulianto","kodejur":"05"}, {"nu":10,"nim":"121051001","nama":"Frederico F.e. Da Cunha","kodejur":"05"}}
```

Data dengan format JSON yang berisi data mahasiswa, adapun format JSON bisa dijelaskan sebagai berikut:

```
[ {"nu":1, → elemen nomor urut "nim":"111051005", → elemen NIM "nama":"SAMUEL KAREL LABOK", → elemen Nama Mahasiswa "kodejur":"05" → elemen Kode Jurusan }, ... ]
```

Tiap record data berisi nu, nim, nama dan kodejur. Sedangkan pengujian di sisi *service client* pada <http://localhost/lh/service/service.php> diperoleh data sebagai berikut:

```
{ "records" : [{"nu":1,"nim":"111051005","nama":"SAMUEL KAREL LABOK","kodejur":"05"},
```

```
{
  "nu":2,"nim":"111051011","nama":"FARIZ SATYA WIRANTO", "kodejur":"05"},
  {"nu":3,"nim":"111051023", "nama":"Yusuf Febriyanto", "kodejur":"05"},
  {"nu":4,"nim":"111051028", "nama":"Vera Laurensiana Kocu", "kodejur":"05"},
  {"nu":5,"nim":"111051063", "nama":"Syifa Nurul Huda", "kodejur":"05"},
  {"nu":6,"nim":"111051074","nama":"FUAD AZIZ", "kodejur":"05"},
  {"nu":7,"nim":"111051075","nama":"Agostinho Fernandes Da Silva", "kodejur":"05"},
  {"nu":8,"nim":"111051092","nama":"Agradira Dwi Wahyuda", "kodejur":"05"},
  {"nu":9,"nim":"112053006","nama":"Permadi Yulianto", "kodejur":"05"},
  {"nu":10,"nim":"121051001","nama":"Frederico F.e. Da Cunha", "kodejur":"05"}
}
```

Pada pengujian ini juga diperoleh dalam format JSON, dengan perincian sebagai berikut:

```
{
  "records" : [
    {
      "nu":1,
      "nim":"111051005",
      "nama":"SAMUEL KAREL LABOK",
      "kodejur":"05"
    },
    ....
  ]
}
```

Penambahan elemen “records” pada bagian ini, yang akan digunakan oleh APPML untuk melakukan *looping data*.

4. KESIMPULAN

Dari paparan hasil penelitian diatas, maka bisa dibuktikan bahwa menggunakan teknik APPML dengan mengkombinasikan dengan JSON akan bisa mendukung komunikasi antar sistem informasi, sehingga penyebaran dan atau penyajian informasi tidak hanya dilakukan oleh sebuah sistem informasi saja, melainkan bisa dilakukan oleh sistem-sistem informasi yang lain. Dari sisi keamanan, juga bisa di tingkatkan, karena sistem informasi *service client* tidak perlu mengetahui user maupun password untuk akses database, bahkan tidak perlu tahu juga nama tabel, relasi dan komponen-komponen lain yang ada pada sistem informasi dan *service server*. *Service client* hanya akan mengetahui hasil data berupa format JSON yang telah disediakan dan tinggal mengolah menjadi output pada tampilan web APPML. Dari sisi networking, maka pada

saat membuat aplikasi *end user*, maka bisa dilakukan secara *localhost* tanpa harus melakukan duplikasi database maupun sistem informasi, karena data bisa langsung di peroleh secara langsung antara *service client* dan *service server* secara realtime selama ada koneksi internet.

5. REFERENSI

- [1] dari w3schools.com:
https://www.w3schools.com/appml/appml_architecture.asp , diakses 05 April 2017.
- [2] AppML Tutorial. 2017. w3schools.com:
<https://www.w3schools.com/appml/default.asp>, diakses 05 April 2017.
- [3] Haryani, P. 2016. Evaluasi Kualitas Layanan E-Government Pemerintah Kota Yogyakarta Dengan Metode E-GovQual Modifikasi. Simposium Nasional Ke-15 RAPI 2016. Hal. 379-386. Surakarta: Fakultas Teknik Universitas Muhammadiyah Surakarta.
- [4] Mangal, M. & Dubey, M. 2015. AppML : An Idea of Mordern Web Development. International Journal Of Engineering & Computer Science (IJECS). P. 11761-11764.
- [5] Saputro, G. A. Hamzah, A. & Triyono, J. 2013. Sistem Informasi Pengolahan Data Rumah Kos Menggunakan Framework Codeigniter Di Kelurahan Demangan Kecamatan Gondokusuman Yogyakarta. *Jurnal Script*.
- [6] Susanti, E. & Triyono, J. 2016. Prototype Alat IoT (Internet Of Things) untuk Pengendali dan Pemantau Kendaraan Secara Realtime. Simposium Nasional RAPI XV 2016.
- [7] Triyono, J. 2015. Sistem Informasi Agroteknologi Berbasis Web Dan Jejaring Sosial Twitter. *Seminar Nasional IENACO*. P. 205-212.