



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian Komputer

Ramadhani (2015:5), “Komputer merupakan perangkat yang mampu melakukan komputasi dan membuat keputusan logis.”

Sobri (2012:17), “Komputer adalah serangkaian ataupun sekelompok mesin elektronik yang terdiri dari ribuan bahkan jutaan komponen yang dapat saling bekerja sama serta membentuk sebuah sistem kerja yang rapi dan teliti”.

Kesimpulannya, Komputer adalah perangkat yang saling bekerja sama guna mencapai satu tujuan.

##### 2.1.2 Pengertian Webserver

Sadeli (2013:2), “ Web Server adalah serangkaian ataupun sekelompok mesin elektronik yang terdiri dari ribuan bahkan jutaan komponen yang dapat saling bekerja sama serta membentuk sebuah sistem kerja yang rapi dan teliti.”

Marisa (2017:24), “ Web Server merupakan software dalam server yang berfungsi untuk menerima permintaan request berupa halaman web melalui HTTP atau HTTPS dari klien yang dikenal dengan web browser dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk HTML.”

Kesimpulannya, Web server adalah software dalam server sebagai tempat request dan response melalui HTTP.

##### 2.1.3 Pengertian Android

Kadir, Abdul (2013:2), “Android adalah sistem operasi yang bersifat open source untuk digunakan pada ponsel, netbook, e-reader, dan tablet”.

Nazaruddin (2015:1), “Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi”.

Kesimpulannya, Aplikasi adalah sistem operasi open source untuk perangkat mobile berbasis linux.



---

#### 2.1.4 Pengertian Website

Irwansyah dan Moniaga (2014:34) menjelaskan, *website* adalah kumpulan halaman yang menyediakan informasi. Sedangkan Hidayat (2010:2) menjelaskan, *website* adalah kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.”

Berdasarkan pendapat diatas, dapat disimpulkan bahwa *website* adalah kumpulan halaman yang digunakan untuk menyediakan informasi yang membentuk satu rangkaian bangunan yang saling terkait.

### 2.2 Teori Judul

#### 2.2.1 Pengertian Aplikasi

Menurut Permana, dalam jurnal Rancang Bangun Sistem Informasi Klinik Praktik Dokter Berbasis Web (2015:129), Aplikasi adalah penggunaan dalam suatu komputer, instruksi atau pernyataan yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output.

Menurut Wikipedia, Aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna.

Kesimpulannya, Aplikasi adalah instruksi atau penggunaan dalam suatu komputer untuk melakukan suatu tugas yang diinginkan pengguna sehingga komputer dapat memproses input menjadi output.

#### 2.2.2 Pengertian Pemeriksaan

Ridwan (2012), “Pemeriksaan adalah serangkaian kegiatan menghimpun dan mengolah data, keterangan, dan/atau bukti yang dilaksanakan secara objektif dan profesional berdasarkan suatu standar pemeriksaan untuk menguji kepatuhan pemenuhan kewajiban perpajakan dan/atau untuk tujuan lain dalam rangka melaksanakan ketentuan peraturan perundang-undangan dan/atau perpajakan”. (<http://walangkopo99.blogspot.com/2012/06/pengertian-pemeriksaan.html>)



---

### 2.2.3 Pengertian Sasaran Operasi

Satu Energi (2016), “Sasaran operasi adalah pelanggan yang terindikasi melakukan pelanggaran berdasarkan peraturan direksi PLN No. 088 Z.P/DIR/2016 tentang penertiban pemakaian tenaga listrik”. (<http://www.satuenergi.com/2016/11/bagaimana-cara-penentuan-target-operasi.html>)

### 2.2.4 Pengertian P2TL (Penertiban Pemakaian Tenaga Listrik)

Dody (2014), “P2TL adalah rangkaian kegiatan meliputi perencanaan, pemeriksaan, tindakan dan penyelesaian yang dilakukan oleh PLN terhadap aset jaringan dan proteksi milik PLN terkait adanya pemakaian tenaga listrik yang tidak tertib.” (<https://dodybastian.wordpress.com/2014/06/27/p2tl/>)

### 2.2.5 Pengertian Metode *Rapid Application Development*

Rosa dan Shalahuddin (2013:156), “*RAD* adalah model proses pengembangan perangkat lunak yang bersifat incremental terutama untuk waktu pengerjaan yang pendek. Model *RAD* adalah adaptasi dari model air terjun untuk pengembangan setiap komponen perangkat lunak.”

Berdasarkan definisi diatas dapat disimpulkan bahwa *RAD* merupakan suatu metode pengembangan sistem yang bertujuan untuk mempersingkat waktu pengerjaan suatu perangkat lunak atau aplikasi.

### 2.2.6 Pengertian *Web Service*

Web Services Architecture Working Group (2004), “*Web service* adalah sebuah system software yang di desain untuk mendukung interoperabilitas interaksi mesin ke mesin melalui sebuah jaringan. Interface web service dideskripsikan dengan menggunakan format yang mampu diproses oleh mesin (khususnya WSDL). Sistem lain yang akan berinteraksi dengan webservice hanya memerlukan SOAP, yang biasanya disampaikan dengan HTTP dan XML sehingga mempunyai korelasi dengan standar Web.” (<https://wildananurul.wordpress.com/2016/09/22/apa-itu-web-service/>)



---

### 2.2.7 Pengertian Arsitektur Komunikasi

Tantri (2016) “Arsitektur Komunikasi adalah urutan proses kerja pada sebuah Host untuk proses pengiriman atau penerimaan.” (<http://blog.unnes.ac.id/ayukwitantri/2016/02/25/pengertian-arsitektur-dan-komunikasi-data-komputer/>)

### 2.2.8 Pengertian *Restful API*

*Restful API* merupakan implementasi dari *API (Application Programming Interface)*. *REST (Representational State Transfer)* adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan aplikasi.

Dari pengertian diatas dapat ditarik kesimpulan yaitu, *Restful API* merupakan suatu arsitektur komunikasi yang berfungsi sebagai jembatan penghubung untuk pertukaran data atau informasi. (<https://kudo.co.id/engineering/2016/09/15/mengenal-restful-api>) diakses pada 14 Maret 2018.

### 2.2.9 Pengertian Aplikasi Pemeriksaan Sasaran Operasi P2TL dengan Metode Rapid Application Development berbasis Android Webservice dengan Arsitektur Komunikasi RestfulAPI pada PT PLN WS2JB

Aplikasi Pemeriksaan Sasaran Operasi P2TL dengan Metode Rapid Application Development berbasis Android *Webservice* dengan Arsitektur Komunikasi RestfulAPI pada PT PLN WS2JB adalah suatu aplikasi yang dapat melakukan pembagian dan pelaporan hasil pemeriksaan sasaran operasi P2TL.

## 2.3 Teori Khusus

### 2.3.1 Pengertian *UML (Unified Modeling Language)*

Rosa A.S dan M. Shalauddin (2015:133), “*UML (Unified Modeling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika

berkembangnya bahasa pemrograman *Smalltalk* pada awal 1980-an. Pada 1996, *Object Management Group* (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 *Unified Modeling Language* (UML) diakomodasi oleh *Object Management Group* (OMG) sehingga sampai saat ini *Unified Modeling Language* (UML) telah memberikan kontribusinya yang cukup besar dalam metodologi berorientasi objek.

### 2.3.2 Usecase Diagram

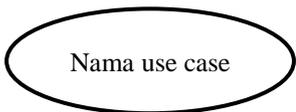
Rosa A.S dan M. Shalahuddin (2015:155), “*Usecase* atau diagram *usecase* merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Usecase* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”.

Syarat penamaan pada *usecase* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *usecase* yaitu pendefinisian apa yang disebut aktor dan *usecase*.

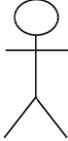
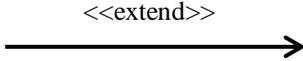
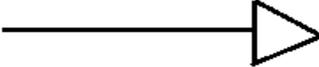
1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Usecase* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *usecase* menurut Rosa dan Shalahuddin (2013:156):

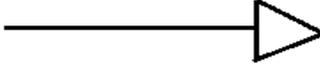
**Tabel 2.1** Simbol-simbol diagram *use case*

No.	Simbol	Keterangan
1.		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>

Lanjutan Tabel 2.1 Simbol-simbol *use case* diagram

No	Simbol	Keterangan
2.	<p>Aktor / <i>actor</i></p>  <p>nama aktor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor</p>
3.	<p>Asosiasi / <i>association</i></p> 	<p>Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor</p>
4.	<p>Ekstensi / <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; ditambahkan, misal arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
5.	<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya</p>

Lanjutan Tabel 2.1 Simbol-simbol *use case diagram*

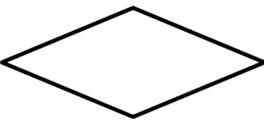
No	Simbol	Keterangan
6.	<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p>&lt;&lt;include&gt;&gt;</p>  <p>&lt;&lt;uses&gt;&gt;</p> 	<p>Relasi use case tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i>.</p>

Sumber : Rosa dan Shalahuddin (2013:156)

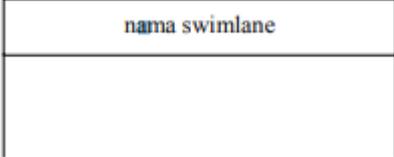
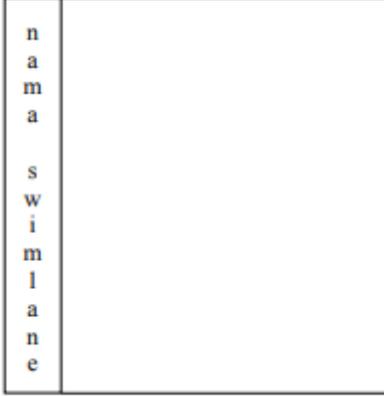
### 2.3.3 Activity Diagram

Sukanto dan Shalahuddin (2013:161), “Diagram aktivitas atau activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. Berikut adalah simbol-simbol yang ada pada diagram aktivitas menurut Rosa dan Shalahuddin (2013:162):

Tabel 2.2 Simbol-simbol *activity diagram*

No.	Simbol	Keterangan
1.	<p>Status Awal</p> 	<p>Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal</p>
2.	<p>Aktivitas</p> 	<p>Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja</p>
3.	<p>Percabangan / <i>decision</i></p> 	<p>Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu</p>
4.	<p>Penggabungan / <i>join</i></p> 	<p>Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu</p>

Lanjutan Tabel 2.2 Simbol-simbol *activity diagram*

No.	Simbol	Keterangan
5.	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6.	Swimlane  atau 	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber : Rosa A.S dan M. Shalahudin (2014:162)

### 2.3.4 Class Diagram

Sukamto dan Shalahuddin (2013:141), “Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian use case (*controller*)

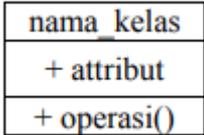
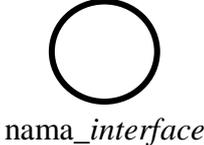
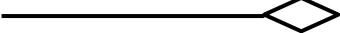
Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian use case, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

4. Kelas yang diambil dari pendefinisian data (*model*)

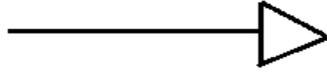
Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Berikut adalah simbol-simbol yang ada pada diagram kelas menurut Rosa dan Shalahuddin (2013:146):

**Tabel 2.3** Simbol-simbol *class diagram*

No.	Simbol	Keterangan
1.	Kelas 	Kelas pada struktur sistem
2.	Antarmuka / <i>interface</i> 	Sama dengan konsep interface dalam pemrograman berorientasi objek
3.	Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
4.	Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
5.	Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua- bagian ( whole-part)

Lanjutan Tabel 2.3 Simbol-simbol *class diagram*

No.	Simbol	Keterangan
6.		Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
7.		Relasi antarkelas dengan makna kebergantungan antarkelas

Sumber : Rosa A.S dan M. Shalahudin (2014:146)

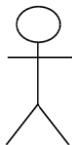
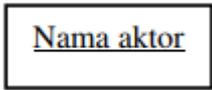
### 2.3.5 Sequence Diagram

Rosa A.S dan M. Shalahuddin (2015:165), “*Sequence diagram* atau diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek”.

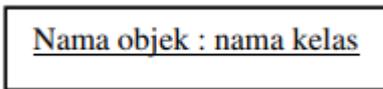
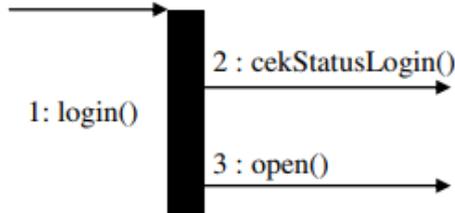
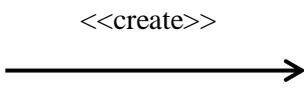
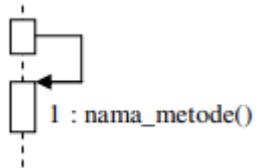
Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada diagram sekuen menurut Rosa dan Shalahuddin (2013:146):

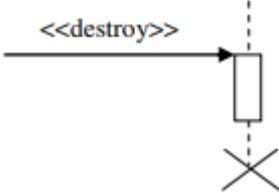
Tabel 2.4 Simbol-simbol *sequence diagram*

No.	Simbol	Keterangan
1.	<p>Aktor</p>  <p>nama aktor</p> <p>atau</p>  <p>tanpa waktu aktif</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.

Lanjutan Tabel 2.4 Simbol-simbol *sequence diagram*

No.	Simbol	Keterangan
2.	Garis Hidup / lifeline 	Menyatakan kehidupan suatu objek
3.	Objek 	Menyatakan objek yang berinteraksi pesan
4.	Waktu Aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya 
5.	Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.	Pesan tipe <i>call</i> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, 

Lanjutan Tabel 2.4 Simbol-simbol *sequence diagram*

No.	Simbol	Keterangan
		Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
7.	Pesan tipe <i>send</i> 1 : masukkan 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8.	Pesan tipe <i>return</i> 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9.	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy

Sumber : Rosa A.S dan M. Shalahudin (2014:146)

### 2.3.6 Pengertian Kamus Data (Data Dictionary)

Rosa A.S dan M. Shalahuddin (2015:73) Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan).

Rosa A.S dan M. Shalahuddin (2015:73), menjelaskan bahwa kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan, yaitu sebagai berikut:

**Tabel 2.5** Simbol-simbol dalam Kamus Data

No	Notasi	Arti
1.	=	disusun atau terdiri dari
2.	+	<i>Dan</i>
3.	[   ]	baik... atau...
4.	{ } <sup>n</sup>	n kali diulang/bernilai banyak
5.	( )	data operasional
6.	*...*	Batas komentar

(Sumber: Sukanto dan Shalahuddin, 2014:74)

## 2.4 Teori Program

### 2.4.1 Pengertian PHP (*Hypertext Preprocessor*)

Hidayatullah dan Kawistara. (2015:231) menjelaskan, PHP atau *Hypertext Preprocessor*, adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server*.

Sedangkan Winarno dan Zaki (2014:1) berpendapat, PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman untuk web yang menganut client server.

### 2.4.2 Pengertian *Restful API*

*Restful API* merupakan implementasi dari *API (Application Programming Interface)*. *REST (Representational State Transfer)* adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan aplikasi. Dimana tujuannya adalah untuk menjadikan sistem yang memiliki performa yang baik, cepat dan mudah untuk di kembangkan terutama dalam pertukaran dan komunikasi data. Konsep *REST* pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000. Cara kerjanya, *REST server* menyediakan jalur untuk akses *resource* atau data, sedangkan *REST client* melakukan akses *resource* dan kemudian menampilkan atau menggunakannya. *Resource* yang dihasilkan sebenarnya berupa teks, namun



formatnya bisa bermacam-macam tergantung keinginan developer, umumnya adalah JSON dan XML.

*Restful API* memiliki 4 komponen penting di dalamnya diantaranya adalah:

1. URL Design
2. HTTP Verbs
3. HTTP Response Code
4. Format Response

Beberapa keuntungan dari *Restful API* yaitu:

- a. Lebih sederhana/simpel untuk dikembangkan ketimbang SOAP.
- b. Mudah dipelajari, tidak bergantung pada tools.
- c. Ringkas, tidak membutuhkan layer pertukaran pesan tambahan.
- d. Secara desain dan filosofi lebih dekat dengan web.

(<https://kudo.co.id/engineering/2016/09/15/mengenal-restful-api/>)

### 2.4.3 Pengertian *Java*

Rosa A.S dan M. Shalahuddin (2015:103) menjelaskan, *Java* adalah bahasa pemrograman objek murni karena semua kode programnya dibungkus dalam kelas.

Menurut Ramadhani (2015:5), *Java* adalah bahasa pemrograman yang dapat dijalankan diberbagai computer termasuk ponsel. *Java* merupakan bahasa pemrograman bersifat umum/nonspesifik (*general purpose*), dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin.

### 2.4.4 Pengertian *Object Oriented Programming (OOP)*

Hidayatullah dan Kawistara. (2015:306), *Object Oriented Programming* atau yang disingkat dengan OOP ialah paradigma atau teknik pemrograman yang berorientasikan kepada objek. Semua data dan fungsi didalam OOP dibagi dalam kelas-kelas atau objek-objek.

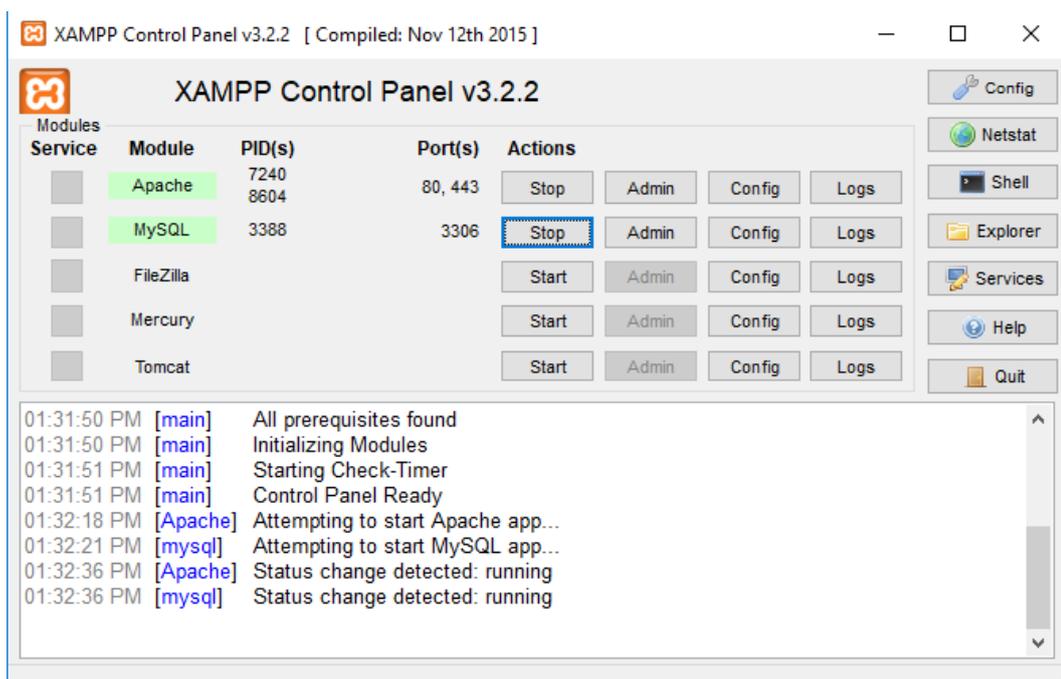
Dari definisi diatas dapat disimpulkan bahwa pemrograman berorientasi objek adalah program komputer dari berbagai objek yang melakukan suatu tindakan terhadap masing-masing objek

### 2.4.5 Pengertian MySQL

Hidayatullah dan Kawistara. (2015:180) menjelaskan, MySQL merupakan aplikasi DBMS (atau *server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user* (*multi-user*), dan dapat melakukan suatu proses secara sinkron atau bersamaan (*multi-threaded*).

### 2.4.6 Pengertian XAMPP

Hidayatullah dan Kawistara. (2015:125), XAMPP adalah paket program web lengkap yang dapat anda pakai untuk belajar pemrograman web, yang dapat dijalankan di windows, linux, mac dsb dimana *apache* sebagai *web server*-nya, MySQL sebagai *Database Management System* (DBMS)-nya dan PHP dan Perl sebagai bahasa yang didukungnya.

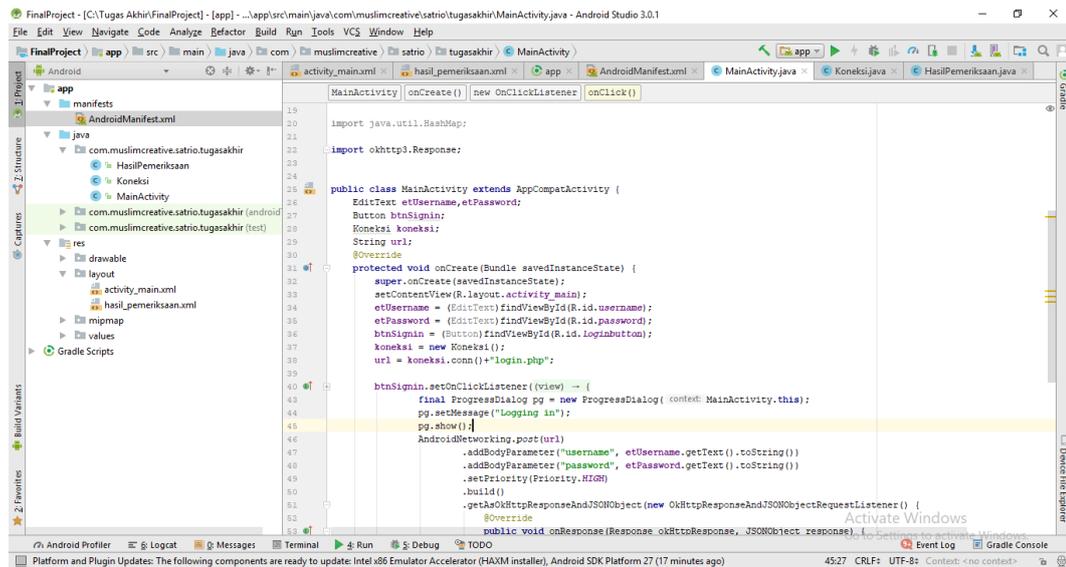


Gambar 2.1 Tampilan XAMPP

### 2.4.7 Pengertian Android Studio

*Android Studio* adalah sebuah *Integrated Development Environment* (IDE) untuk platform Android. *Android Studio* tersedia bagi pengembang untuk mencoba secara gratis. *Android Studio* berada di awal tahap *preview access* mulai

dari versi 0.1 pada Mei 2013, kemudian memasuki tahap beta mulai dari versi 0.8 yang dirilis pada bulan Juni 2014. Berdasarkan software JetBrains IntelliJ IDEA, *Android Studio* dirancang khusus untuk pengembangan Android. ([http://en.wikipedia.org/wiki/Android\\_Studio](http://en.wikipedia.org/wiki/Android_Studio))



Gambar 2.2 Tampilan Android Studio