

**PENERAPAN METODE SINKRONISASI DATABASE  
BERBASIS REST WEB SERVICE PADA KAMUS  
MOBILE JAWA-INDONESIA**

**SKRIPSI**

oleh:

**MUHAMMAD HUBAIB FADLAL**

**NIM. 07650019**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2014**

**PENERAPAN METODE SINKRONISASI DATABASE  
BERBASIS REST WEB SERVICE PADA KAMUS  
MOBILE JAWA-INDONESIA**

**SKRIPSI**

**Diajukan Kepada:  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk Memenuhi Salah Satu Persyaratan Dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:**

**MUHAMMAD HUBAIB FADLAL  
NIM. 07650019**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG**

**2014**

**PENERAPAN METODE SINKRONISASI DATABASE  
BERBASIS REST WEB SERVICE PADA KAMUS  
MOBILE JAWA-INDONESIA**

**SKRIPSI**

Oleh:

**MUHAMMAD HUBAIB FADLAL  
NIM. 07650019**

Telah Disetujui  
Malang, 27 Juni 2014

**Pembimbing I**

**Pembimbing II**

**A'LA SYAUQI, M.Kom**  
NIP. 197712012008011007

**Dr. M. AMIN HARIYADI, MT**  
NIP. 196701182005011001

Mengetahui,  
Ketua Jurusan Teknik Informatika

**Dr. Cahyo Crysdiان, M.CS**  
NIP. 197404242009011008

**PENERAPAN METODE SINKRONISASI DATABASE  
BERBASIS REST WEB SERVICE PADA KAMUS  
MOBILE JAWA-INDONESIA**

**SKRIPSI**

Oleh:

**MUHAMMAD HUBAIB FADLAL  
NIM. 07650019**

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan  
Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)

Tanggal, 15 Juli 2014

Susunan Dewan Penguji:	Tanda Tangan
1. Penguji Utama : <u>Irwan Budi Santoso, M.Kom</u> NIP. 197701032011011004	( )
2. Ketua Penguji : <u>Fatchurrochman, M.Kom</u> NIP. 197007312005011002	( )
3. Sekretaris Penguji: <u>A'la Syauqi, M.Kom</u> NIP. 197712012008011007	( )
4. Anggota Penguji : <u>Dr. M. Amin Hariyadi, MT</u> NIP. 196701182005011001	( )

Mengetahui,  
Ketua Jurusan Teknik Informatika

**Dr. Cahyo Crysdiyan, M.CS**  
NIP. 197404242009011008

## SURAT PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama : Muhammad Hubaib Fadlal  
NIM : 07650019  
Fakultas / Jurusan : Sains Dan Teknologi / Teknik Informatika  
Judul Penelitian : Penerapan Metode Sinkronisasi Database Berbasis REST  
Web Service Pada Kamus Mobile Jawa-Indonesia

Menyatakan dengan sebenar-benarnya bahwa hasil penelitian saya ini tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur jiplakan, maka saya bersedia untuk mempertanggung jawabkan, serta diproses sesuai peraturan yang berlaku.

Malang, 23 Juni 2014

Yang Menyatakan,

Muhammad Hubaib Fadlal  
NIM. 07650019

## MOTTO

“Sesungguhnya sesudah kesulitan itu ada kemudahan” (QS Al Insyiroh : 6)

- ❖ Ketika Allah mengujimu dengan satu masalah, sesungguhnya Allah juga telah menyediakan berjuta-juta jalan untuk mengatasinya.
- ❖ Allah tidak akan menguji hamba melebihi batas kemampuannya, oleh karena itu berusaha semampu kita dan hindarkan diri dari sifat putus asa.
- ❖ Sebaik – baik manusia adalah orang yang paling banyak manfaatnya bagi masyarakat sekitarnya (Al Hadits).
- ❖ Salah satu alasan utama mengapa seseorang tidak sukses dalam hidupnya karena ketakutannya terhadap apa yang akan terjadi (Robert T. Kiyosaki)

## LEMBAR PERSEMBAHAN

### **Yang Utama Dari Segalanya...**

Sembah sujud serta syukur kepada Allah SWT. Taburan cinta dan kasih sayang-Mu telah memberikanku kekuatan. Atas karunia serta kemudahan yang Engkau berikan akhirnya skripsi yang sederhana ini dapat terselesaikan. Sholawat dan salam selalu terlimpahkan keharibaan Rasulullah Muhammad SAW.

Dengan kerendahan hati kupersembahkan karya sederhana ini kepada orang yang selalu memberikan motivasi, kasih sayang dan do'anya yang begitu tulus kepadaku:

### **Ibunda Khusnul Khotimah dan Ayahanda Sholihan Tercinta**

Sebagai tanda bakti, hormat, dan rasa terima kasih yang tiada terhingga kupersembahkan karya sederhana ini kepada Ibu dan Ayah yang telah memberikan kasih sayang, segala dukungan, dan cinta kasih yang tiada terhingga yang tiada mungkin dapat kubalas hanya dengan selembar kertas yang bertuliskan kata cinta dan persembahan. Semoga ini menjadi langkah awal untuk membuat Ibu dan Ayah bahagia karna kusadar, selama ini belum bisa berbuat yang lebih. Untuk Ibu dan Ayah yang selalu membuatku termotivasi dan selalu menyirami kasih sayang, selalu mendo'akanku, selalu menasehatiku menjadi lebih baik, Terima Kasih Ibu.... Terima Kasih Ayah...

### **Kakak Perempuan dan Adik-adikku..**

#### **Mbak Elly Fauziyah, Miladul Ainiyah dan Ahmad Atha'ul Karim**

Untuk mbak Elly yang selalu memberikan motivasi untuk segera menyelesaikan skripsi ini dan untuk adik-adiku yang telah bersama-sama berjuang untuk bisa menyelesaikan skripsi, semoga kita semua menjadi anak yang sholeh/hah yang selalu berbakti kepada orang tua.

### **Kepada Guru-guruku mulai dari TK Hidayatus Sibyan, SDI Riyadlul Muhtadi'in Kedok, MTs Negeri Turen, MAN 3 Malang, beserta Dosen-dosen UIN Maliki Malang**

Terimakasih untuk motivasi dan do'anya juga untuk semua ilmu yang telah diberikan, semoga menjadi ilmu yang bermanfaat untuk kehidupan dunia dan akhirat.

### **Murid-muridku di MTsN Turen**

Kebersamaan yang tidak lama tapi penuh makna, terimakasih untuk keceriaan yang kalian bagikan sehingga saya bisa ikut merasakannya. Terimakasih untuk murid-muridku yang telah mendoakan saya dan memotivasi saya untuk menyelesaikan skripsi ini. Mohon maaf bila saya tidak bisa menemani kalian untuk belajar bersama sampai akhir.

### **Para Sahabat-sahabatku dan teman-teman seperjuangan (TI UIN 2007)**

Seperti pada lirik lagu "persahabatan bagai kepompong, mengubah ulat menjadi kupu-kupu" semoga persahabatan kita bisa membawa kita pada kebaikan dan manfaat. Untuk teman-teman seperjuangan yang berjuang bersama sampai akhir walaupun jalan kita panjang untuk meraih ini tetap semangat untuk segera meraih sukses yang tertunda.

### **Dan semua pihak yang belum bisa ku sebutkan satu persatu,**

Hanya terimakasih yang bisa kusampaikan untuk semuanya.

## KATA PENGANTAR

*Alhamdulillah rabbil 'alamin*, puji syukur kehadirat Allah SWT, yang melimpahkan segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penerapan Metode Sinkronisasi Database Berbasis REST Web Service Pada Kamus Mobile Jawa-Indonesia” yang menjadi salah satu syarat untuk menyelesaikan program studi Teknik Informatika jenjang Strata-1 Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang. Dan semoga Allah melimpahkan rahmat atas Nabi Muhammad SAW yang senantiasa memberikan cahaya petunjuk kepada kita.

Dengan segala kerendahan hati, penulis menyadari bahwa dalam menyelesaikan skripsi ini tidak lepas dari peran berbagai pihak yang telah banyak memberikan bantuan, bimbingan dan dorongan. Dalam kesempatan ini penulis ingin mengucapkan terima kasih yang tak terhingga khususnya kepada:

1. Prof. Dr. Mudjia Raharjo, M,Si selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang. yang telah banyak memberikan pengetahuan dan pengalaman yang berharga.
2. Dr. drh. Hj. Bayyinatul Muchtaromah, M.Si. selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Chrysdian, M.CSselaku ketua jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.
4. A’la Syauqi, M.Komdan Dr. M. Amin Hariyadi, MTselaku dosen pembimbing skripsi, yang telah banyak memberikan bimbingan serta motifasi kepada penulis dalam menyelesaikan skripsi ini.
5. Seluruh Dosen Teknik Informatika UIN Maulana Malik Ibrahim Malang yang telah mengajar penulis selama empat tahun lamanya, dan memberikan dukungan untuk menyelesaikan penulisan skripsi ini.
6. Seluruh keluarga yang selalu mendoakan, memberikan motivasi dan dorongan dalam penyelesaian skripsi ini.
7. Teman-teman seperjuangan Teknik Informatika angkatan 2007, terima kasih atas segala bantuan, dukungan, motivasi, dan kebersamaannya selama ini.

Semoga Allah SWT memberikan balasan yang setimpal atas jasa dan bantuan yang telah diberikan.

8. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah banyak membantu dalam penyelesaian skripsi ini.

Penulis menyadari sepenuhnya bahwa sebagai manusia biasa tentunya tidak akan luput dari kekurangan dan keterbatasan. Maka dengan segenap kerendahan hati, penulis mengharapkan saran dan kritik yang dapat menyempurnakan penulisan ini sehingga dapat bermanfaat dan berguna untuk pengembangan ilmu pengetahuan.

Malang, 23 Juni 2014

Penulis

## DAFTAR ISI

Halaman Pengajuan.....	i
Halaman Persetujuan.....	ii
Halaman Pengesahan .....	iii
Halaman Pernyataan.....	iv
Motto.....	v
Lembar Persembahan .....	vi
Kata Pengantar .....	vii
Daftar Isi.....	ix
Daftar Tabel .....	xiii
Daftar Gambar.....	xiv
Abstrak.....	xvi
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian .....	4
1.5 Sistematika Penulisan Laporan .....	5
<b>BAB II LANDASAN TEORI .....</b>	<b>6</b>
2.1 Penelitian Terkait .....	6

2.2 Web Service .....	8
3.2.1 SOAP.....	10
3.2.2 REST .....	11
2.3 JSON .....	11
2.4 JSON VS XML .....	15
2.5 Android.....	19
2.5.1 Arsitekture Android.....	20
2.5.1.1 Linux Kernel .....	21
2.5.1.2 Android Runtime.....	21
2.5.1.3 Libraries .....	22
2.5.1.4 Aplication Framework .....	23
2.5.1.5 Aplication.....	23
2.6 Bahasa Jawa .....	23
<b>BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM.....</b>	<b>26</b>
3.1 Analisis Permasalahan Sistem.....	26
3.2 Analisa Kebutuhan .....	27
3.2.1 Software .....	28
3.2.2 Hardware .....	29
3.3 Tahapan Pembuatan .....	30
3.4 Spesifikasi Aplikasi.....	31
3.5 Spesifikasi Pengguna.....	32
3.6 Deskripsi Sistem.....	32
3.7 Desain Sistem.....	33

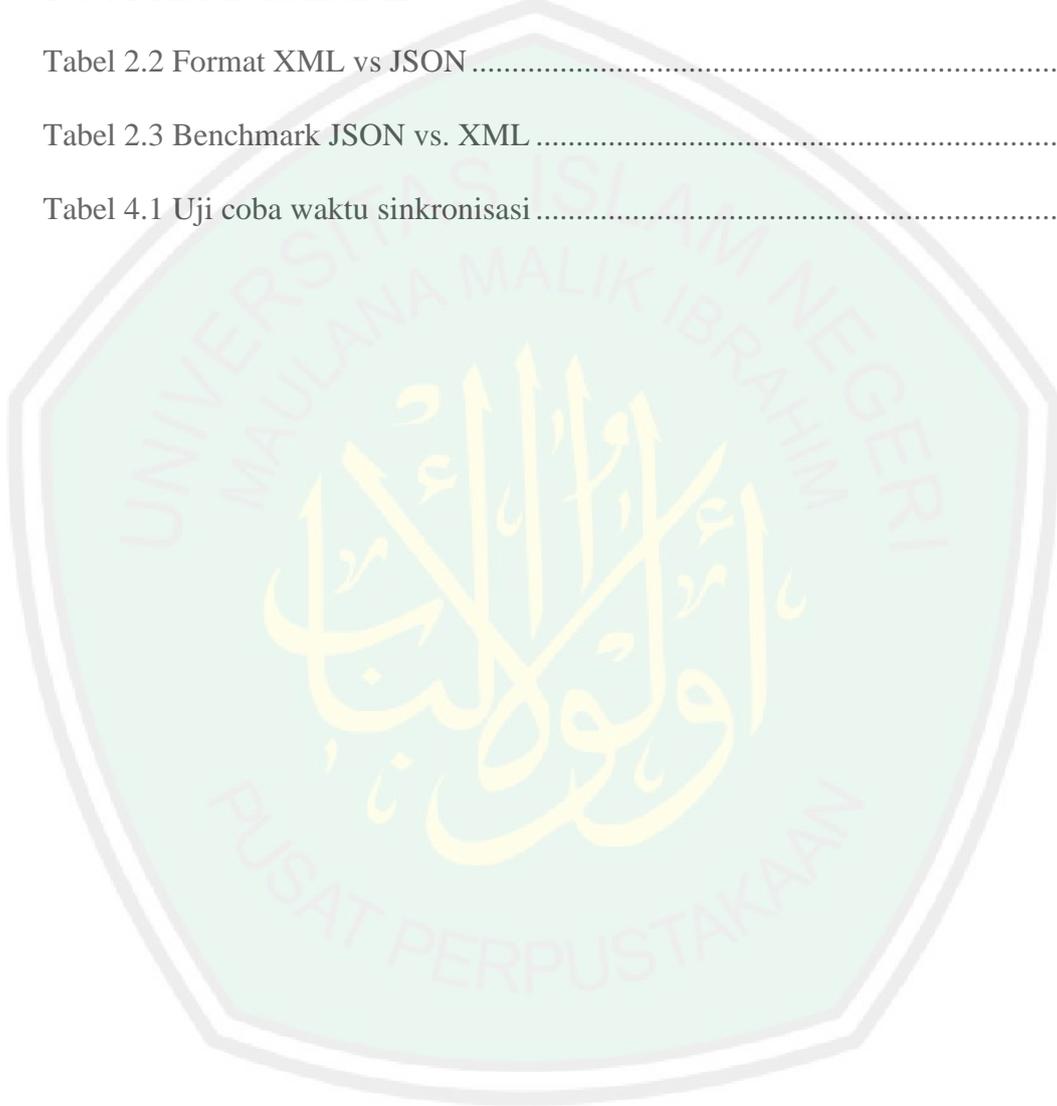
3.7.1 Server .....	34
3.7.2 Client .....	35
3.8 Desain Interface.....	37
3.8.1 Menu Kamus .....	38
3.8.2 Menu Update .....	39
3.8.2 Menu Bantuan dan Tentang .....	40
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>42</b>
4.1 Implementasi Sistem .....	42
4.1.1 Implementasi Aplikasi Server REST Web Service.....	42
4.1.2 Implementasi Aplikasi Client Kamus Mobile.....	45
4.2 Implementasi Interface .....	45
4.2.1 Menu Utama .....	46
4.2.2 Menu Kamus .....	47
4.2.3 Menu Update .....	49
4.2.3 Menu Bantuan .....	52
4.2.3 Menu Tentang .....	53
4.3 Uji Coba Sistem .....	53
4.3.1 Uji Coba Aplikasi Kamus .....	54
4.3.2 Uji Coba Sinkronisasi Database .....	55
4.3.2.1 Uji Coba Kesamaan Database.....	55
4.3.2.2 Uji Coba Waktu Sinkronisasi.....	58
4.4 Kajian Bahasa Jawa dari sisi Islam .....	62

<b>BAB V PENUTUP</b> .....	64
5.1 Kesimpulan.....	64
5.2 Saran.....	65
<b>DAFTAR PUSTAKA</b> .....	66
<b>LAMPIRAN</b> .....	68



## DAFTAR TABEL

Tabel 2.1 Karakteristik XML vs. JSON.....	15
Tabel 2.2 Format XML vs JSON .....	18
Tabel 2.3 Benchmark JSON vs. XML .....	19
Tabel 4.1 Uji coba waktu sinkronisasi .....	61



## DAFTAR GAMBAR

Gambar 2.1 Object dalam JSON .....	12
Gambar 2.2 Array dalam JSON .....	13
Gambar 2.3 Value dalam JSON .....	13
Gambar 2.4 String dalam JSON .....	14
Gambar 2.5 Number dalam JSON .....	14
Gambar 2.6 Arsitektur Android .....	20
Gambar 3.1 Fase-fase metode Waterfall .....	30
Gambar 3.2 Deskripsi sitem secara umum .....	33
Gambar 3.3 Skema proses komunikasi dan pertukaran data .....	33
Gambar 3.4 Flowchart alur proses pada Server .....	34
Gambar 3.5 Flowchart proses translasi pada Client .....	35
Gambar 3.6 Flowchat proses sinkronisasi database .....	36
Gambar 3.7 Menu Utama kamus mobile .....	37
Gambar 3.8 Desain interface aplikasi Kamus .....	38
Gambar 3.9 Desain interface aplikasi Update .....	39
Gambar 3.10 Desain interface menu Bantuan .....	40
Gambar 3.11 Desain interface aplikasi Tentang .....	41
Gambar 4.1 Kode program utama REST Web Service .....	44
Gambar 4.2 Struktur table kamus “javaind” pada server .....	45
Gambar 4.3 Interface Menu Utama .....	46
Gambar 4.4 Interface menu Kamus .....	47

Gambar 4.5 Kode program untuk fungsi penerjemahan .....	48
Gambar 4.6 Interface menu Update .....	49
Gambar 4.7 Kode program aksi update data/ sinkronisasi.....	50
Gambar 4.8 Respon data JSON dari Server .....	51
Gambar 4.9 Interface menu Bantuan .....	52
Gambar 4.10 Interface menu Tentang.....	53
Gambar 4.11 Hasil terjemah kamus .....	54
Gambar 4.12 Isi table awal pada database local SQLite .....	56
Gambar 4.13 Isi table awal pada database server MySQL .....	57
Gambar 4.14 Isi table pada SQLite setelah update/ sinkronisasi data .....	58
Gambar 4.15 Grafik uji coba waktu sinkronisasi.....	60

## ABSTRAK

Hubaib Fadlal, Muhammad. 2014. **Penerapan Metode Sinkronisasi Database Berbasis REST Web Service Pada Kamus Mobile Jawa-Indonesia.**  
Pembimbing: (1) A'la Syauqi, M.Kom (2) Dr. M. Amin Hariyadi, MT

---

Kata Kunci: REST, Web Service, JSON, Database, Kamus

Saat ini penggunaan *smartphone* semakin meluas di masyarakat, terutama *smartphone* berbasis android. Hal ini dikarenakan harganya yang terjangkau oleh seluruh kalangan masyarakat. Hal ini bisa dimanfaatkan untuk pengembangan aplikasi dengan tujuan basis pengguna yang besar. Salah satu pemanfaatannya dalam bidang budaya dan pendidikan yaitu pengembangan aplikasi kamus mobile Jawa-Indonesia. Aplikasi kamus *mobile* haruslah bersifat fleksibel dalam artian bisa digunakan kapan dan dimana saja. Untuk itu kamus *mobile* haruslah memiliki database sendiri agar bisa diakses tanpa harus ada koneksi internet. Pada android database lokal yang tersedia dan bisa dimanfaatkan yaitu SQLite. Selain fleksibel kamus *mobile* juga harus efisien, pada database SQLite apabila ada *update* atau tambahan kosakata seharusnya tidak perlu dilakukan update aplikasi secara keseluruhan karena update terjadi hanya pada database dan itu termasuk tidak efisien. Untuk itulah dibangun sebuah REST *Web Service* sebagai layanan update database yang berfungsi untuk mensinkronkan database lokal SQLite dengan database *server* MySQL agar *update* dapat dilakukan pada sisi database saja.

## ABSTRACT

Hubaib Fadlal, Muhammad. 2014. **The Application of Database Synchronization Method Based REST Web Service In Mobile Dictionary Java-Indonesia.**

Supervisor: (1) A'la Syauqi, M.Kom (2) Dr. M. Amin Hariyadi, MT

---

Keywords: REST, Web Service, JSON, Database, Dictionary

Nowadays, the use of smartphones in society spreads widely, especially Android-based smart phones. This is because the price is affordable by the entire community. It can be used for application development with the aim of a large user base. One of the benefits in the field of culture and education is a mobile Java-Indonesian dictionary. Mobile dictionary application must be flexible in the sense that can be used anytime and anywhere. Therefore, Mobile dictionary must have its own database that can be accessed without any internet connection. At the local database android available and able to be utilized is called SQLite. Besides, mobile dictionary should be not only flexible, but also it must be efficient. If any updates or additional vocabulary, the SQLite database should not be necessary to update the application as a whole because the update occurs only on the database and it is not efficient. Consequently, it is built a REST web service as a service to update the database that serves to synchronize the local database SQLite to MySQL database server so that updates can be performed on the database side only.

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Bahasa adalah sebuah alat komunikasi, di Indonesia kita memiliki bahasa persatuan yaitu bahasa Indonesia. Selain memiliki bahasa persatuan, Indonesia juga memiliki banyak bahasa daerah. Banyaknya bahasa daerah di Indonesia ini tidak lepas dari banyaknya suku di Indonesia dan hampir setiap suku memiliki bahasa daerah masing-masing. Tentang dijadikannya manusia bersuku-suku ini sesuai adanya dengan firman Allah dalam surat Al Hujuraat ayat 13.

يَتَأْتِيهَا النَّاسُ إِنَّا خَلَقْنَاهُمْ مِنْ ذَكَرٍ وَأُنْثَىٰ وَجَعَلْنَاهُمْ شُعُوبًا وَقَبَائِلَ لِتَعَارَفُوا ۗ إِنَّ

أَكْرَمَكُمْ عِنْدَ اللَّهِ أَتَقَنُّكُمْ ۗ إِنَّ اللَّهَ عَلِيمٌ خَبِيرٌ ﴿١٣﴾

*“Hai manusia, sesungguhnya Kami menciptakan kamu dari seorang laki-laki dan seorang perempuan dan menjadikan kamu berbangsa - bangsa dan bersuku-suku supaya kamu saling kenal-mengenal. Sesungguhnya orang yang paling mulia diantara kamu disisi Allah ialah orang yang paling taqwa diantara kamu. Sesungguhnya Allah Maha Mengetahui lagi Maha Mengenal”. (QS. Al-Hujuraat:13)*

Sesuai dengan paparan ayat 13 surat Al Hujarat, tujuan dijadikan bersuku-suku adalah agar saling mengenal. Untuk bisa saling mengenal bisa dilakukan melalui komunikasi dan untuk berkomunikasi diperlukan bahasa. Alat yang paling umum digunakan untuk belajar bahasa tertentu adalah dengan menggunakan kamus, akan tetapi penggunaan kamus yang berbentuk buku dirasa kurang efisien saat dihadapkan dengan pencarian kosakata, yang membutuhkan ketelitian untuk menelusuri kata demi kata. Selain itu untuk pengguna dengan mobilitas yang tinggi kamus berbentuk buku ini tidak praktis untuk dibawa bepergian.

Hal inilah yang melatarbelakangi penulis untuk mengembangkan aplikasi kamus *mobile* Jawa-Indonesia. Pemilihan bahasa Jawa sendiri dikarenakan bahasa Jawa adalah bahasa yang dikuasai penulis. Aplikasi kamus yang banyak beredar saat ini apabila ada penambahan kata maka akan dirilis *update* versi baru dalam bentuk aplikasi satu utuh padahal *update* yang terjadi hanya pada database. Untuk permasalahan ini peneliti mencoba memanfaatkan REST *Web Service* untuk memberikan layanan sinkronisasi database *server* (MySQL) dengan database lokal (SQLite) yang ada pada android. Untuk format pertukaran data penulis memilih menggunakan JSON dibanding XML karena dengan konten yang sama JSON bisa menghasilkan ukuran file yang jauh lebih kecil.

Dengan adanya aplikasi kamus Jawa-Indonesia ini tentu akan lebih efisien (menghemat waktu) dibanding menggunakan kamus buku dalam hal pencarian kata. Dengan model sinkronisasi database akan lebih menghemat *bandwidth* dibanding harus mengunduh ulang seluruh aplikasi update secara keseluruhan.

Selain itu pemilihan format data JSON ini juga akan menghemat bandwidth untuk transfer data. Hal ini merupakan langkah atau wujud untuk melaksanakan perintah Allah untuk berhemat (larangan berlaku boros) yang Allah firmankan dalam surat Al-Isra ayat 26-27.

وَأَاتِ ذَا الْقُرْبَىٰ حَقَّهُ وَالْمِسْكِينَ وَابْنَ السَّبِيلِ وَلَا تُبَذِّرْ تَبْذِيرًا ﴿٢٦﴾

إِنَّ الْمُبَذِّرِينَ كَانُوا إِخْوَانَ الشَّيْطَانِ ۖ وَكَانَ الشَّيْطَانُ لِرَبِّهِ كَفُورًا ﴿٢٧﴾

*“Dan berikanlah kepada keluarga-keluarga dekat akan haknya, kepada orang-orang miskin dan yang dalam perjalanan, dan janganlah kamu menghambur-hamburkan hartamu dengan boros. Sesungguhnya pemboros-pemboros itu adalah saudara-saudara syaitan, dan syaitan itu adalah sangat ingkar kepada TuhanNya” (QS. Al-Isra: 26-27)*

## 1.2 Rumusan Masalah

1. Bagaimana membangun *server* database kamus Jawa-Indonesia beserta aplikasi REST *Web Service*.
2. Apakah aplikasi *client* kamus Jawa-Indonesia pada android bisa berkomunikasi dengan menggunakan REST *Web Service*.

3. Bagaimana sinkronisasi database lokal kamus Jawa-Indonesia pada *device* Android dengan database *server* dengan memanfaatkan REST *Web Service*.

### 1.3 Batasan Masalah

Penerjemahan Jawa-Indonesia pada kamus *mobile* ini berupa penerjemahan kata bukan kalimat.

### 1.4 Tujuan Penelitian

Tujuan penelitian ini adalah membangun sistem *client-server* yaitu aplikasi *client* berupa kamus *mobile* Jawa-Indonesia berbasis yang dapat digunakan untuk mempermudah mencari kosa-kata hasil terjemahan Jawa-Indonesia dan aplikasi *server* berupa *web service* berbasis REST *Web Service* yang dapat memberikan layanan sinkronisasi update database pada aplikasi *client*.

### 1.5 Manfaat Penelitian

Sebagai salah satu pemanfaatan teknologi dalam melestarikan salah satu budaya daerah kita, dalam hal ini yaitu bahasa daerah khususnya bahasa Jawa. Selain itu aplikasi ini juga bermanfaat sebagai media pembelajaran di lingkungan pendidikan.

## **1.6 Sistematika Penulisan Laporan**

Untuk mempermudah mendapatkan gambaran mengenai isi dari pembahasan Skripsi ini, maka penulis menyusun secara sistematis dalam beberapa bab sebagai berikut:

### **BAB I PENDAHULUAN**

Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

### **BAB II LANDASAN TEORI**

Pada bab ini membahas tentang teori-teori yang menjadi acuan dalam pembuatan analisa dan pemecahan dari permasalahan yang dibahas, sehingga memudahkan penulis dalam menyelesaikan masalah.

### **BAB III DESAIN DAN PERANCANGAN SISTEM**

Pada bab ini menjelaskan mengenai tahapan-tahapan yang dilalui dalam penyelesaian tugas akhir ini, mulai dari tahapan perancangan blok diagram sistem (desain umum sistem), perancangan *flowchart* diagram, dan perancangan desain *interface*.

### **BAB IV HASIL DAN PEMBAHASAN**

Hasil dan pembahasan berisi kebutuhan peralatan, cara instalasi program, cara pemakaian program yang dijelaskan dalam gambar dan penjelasan proses aplikasi yang terjadi pada sistem.

### **BAB V PENUTUP**

Berisi tentang kesimpulan yang diambil dari pembahasan program aplikasi sistem informasi ini dan saran untuk pengembangan program aplikasi ini.

## BAB II

### LANDASAN TEORI

#### 2.1 Peneletian Terkait

Telah banyak penelitian yang berhubungan dengan *web service* dan aplikasi kamus, diantaranya adalah pemanfaatan *web service* pada kamus online bahasa Indonesia (Fauzan Natsir, 2013), aplikasi kamus *mobile* bahasa sunda berbasis *Symbian OS* (Iriani Leginingsih, 2006), aplikasi kamus online Indonesia-Karo berbasis web (Jhoni Pranata Sembiring, 2013), aplikasi kamus Jerman-Indonesia berbasis Android (Muhammad Fadlullah, 2012), dan aplikasi kamus *mobile* Aceh-Indonesia berbasis Java (Viska Mutiawani, 2011).

Fauzan Natsir membuat kamus online bahasa Indonesia dengan memanfaatkan teknologi *web service*. Aplikasi yang dihasilkan dapat menyediakan layanan *web service* dalam format XML yang memberikan informasi kata pada kamus bahasa Indonesia lengkap dengan definisi kata, kelas kata, sinonim, antonim, kata majemuk, kata jadian, hiponim dan meronim. Kelebihan kamus ini adalah informasi yang ditampilkan lengkap bukan hanya sekedar definisi atau arti kata. Sedangkan kekurangan aplikasi ini adalah hanya bisa digunakan jika ada koneksi internet.

Pembangunan Aplikasi Kamus Bahasa Sunda pada *Smartphone* Berbasis Sistem Operasi *Symbian 7.0* oleh Iriani Leginingsi diujikan pada *smartphone* Sony Ericsson P800. Aplikasi kamus yang dikembangkan ini memiliki fungsi standar kamus yaitu untuk menterjemahkan kata dari bahasa Sunda ke dalam

bahasa Indonesia. Pada kamus *mobile* ini hanya digunakan metode query database untuk proses pencarian terjemahan kata.

Jhoni Pranata Sembiring membangun sebuah kamus online bahasa Indonesia-Karo (salah satu bahasa daerah di Indonesia). Kelebihan aplikasi kamus berbasis web yang dikembangkan ini adalah adanya metode pencarian *Sequential Search* sehingga pencarian dapat dilakukan dengan cepat. Kelemahan aplikasi kamus berbasis web ini adalah harus adanya koneksi internet untuk dapat menggunakannya.

Muhammad Fadlullah mengembangkan sebuah aplikasi kamus *mobile* berbasis Android untuk menerjemahkan kata bahasa Jerman-Indonesia dan juga Indonesia-Jerman. Aplikasi kamus *mobile* ini menyimpan database kata pada SQLite sehingga bisa digunakan tanpa harus terkoneksi internet seperti halnya pada kamus *online*. Pada kamus berbasis Android ini tidak digunakan metode tertentu untuk proses pencarian terjemahan kata.

Viska Mutiawani mengembangkan kamus *mobile* berbasis Java Micro MIDP yang berfungsi untuk menerjemahkan kata bahasa Aceh ke dalam bahasa Indonesia. Kelebihan aplikasi kamus ini adalah pada proses pencarian kata digunakan 2 (dua) metode pencarian, yang pertama metode pencarian biner untuk pencarian kata terjemahan dan yang kedua metode pencarian *approximate string matching* yang digunakan untuk mencari padanan kata apabila tidak ditemukan dengan metode pencarian yang pertama.

## 2.2 Web Service

*Web service* adalah teknologi yang mengubah kemampuan *internet* dengan menambahkan kemampuan *transactionalweb*, yaitu kemampuan *web* untuk saling berkomunikasi dengan pola *program-to-program* (P2P).

*Web Service* sebenarnya adalah kumpulan dari fungsi dan *method* yang terdapat pada sebuah *server* yang dapat dipanggil oleh *client* dari jarak jauh, kemudian untuk memanggil *method-method* tersebut kita bebas menggunakan aplikasi yang akan dibuat dengan bahasa pemrograman apa saja yang dijalankan pada *platform* apa saja.

*Web Service* diperlukan karena pada masa sekarang ini perangkat keras, sistem operasi, aplikasi hingga bahasa pemrograman semakin beraneka ragam jenisnya. Keadaan tersebut dapat menimbulkan masalah dalam proses pertukaran data antar perangkat yang menggunakan aplikasi dan *platform* yang berbeda.

Beberapa alasan mengapa digunakannya *web service* adalah sebagai berikut :

1. *Web service* dapat digunakan untuk mentransformasikan satu atau beberapa *business logic* atau *class* dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangan dengan baik.
2. *Web service* memiliki kemudahan dalam proses *deployment*-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. *Web service* cukup di-*upload* ke *web server* dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.

3. *Web service* berjalan di port 80 yang merupakan protokol standard HTTP, dengan demikian *web service* tidak memerlukan konfigurasi khusus pada *firewall*.

Adapun fitur-fitur kunci yang ditawarkan oleh *web service* adalah sebagai berikut:

1. *Web service* adalah teknologi yang *loosely-coupled* dengan *client*. Jika sebuah *client* yang melakukan *request* ke sebuah *Web Service*, maka *web service* akan mengembalikan hasil dan kemudian koneksi akan ditutup jika tidak terdapat koneksi yang permanen dan kompleksitas-kompleksitas lainnya yang disebutkan di atas. *Interface* dari *Web Service* dapat terus dikembangkan, kita dapat menambahkan *method-method* baru dan parameter-parameter tanpa mempengaruhi *client-client*, selama *method-method*, dan parameter-parameter yang lama masih tetap tersedia.
2. *Web Service* bersifat *stateless*, *Web Service* tidak menyimpan koneksi ke *client*. Sehingga memudahkan untuk *scaling out* dan *up*, dan mengutamakan *resource* yang ada untuk melayani *Web Service*. Protokol HTTP yang digunakan oleh *Web Service* juga bersifat *stateless*. Akan tetapi tetap memungkinkan untuk melakukan penanganan *state* pada *Web Service* seperti bagaimana yang dilakukan pada *web* sekarang dengan menggunakan *cookies*.
3. *Web service* menggunakan teknologi-teknologi standard seperti SOAP, REST, XML, HTTP sehingga memungkinkan untuk diterapkan pada

berbagai *platform* dan *vendor* yang berbeda-beda. *Web service* diakses melalui aplikasi-aplikasi yang dapat berupa sebuah aplikasi *web*, aplikasi *windows*, dan tipe-tipe aplikasi lainnya. Sebuah *Web Service* dapat juga digunakan oleh *Web Service* lainnya. Ada metode yang berbeda untuk menyediakan *Web Service* tetapi yang paling umum adalah SOAP dan REST .

### 2.2.1 SOAP

SOAP (*Simple Object Access Protocol*) didesain untuk menjadi protokol baru pada *environment* yang desentralisasi dan terdistribusikan yang memanfaatkan kekuatan *internet* dan XML untuk menyampaikan pesan yang diketik antara *node*.

SOAP pada dasarnya adalah sebuah paradigme pertukaran pesan satu arah antara *node*, dari pengirim SOAP menuju penerima SOAP. Dengan menggabungkan pertukaran satu arah dengan fitur yang disediakan oleh protokol *transport* yang mendasari atau aplikasi yang spesifik, SOAP dapat digunakan untuk membuat interaksi yang lebih kompleks seperti *request-response*, *request-multiresponses*, dan lain-lain.

Ada dua jenis SOAP *request*. Yang pertama adalah *Remote Procedure Call* (RPC) yaitu *client* mengirim pesan dan menunggu untuk mendapatkan tanggapan atau pesan *error* dari *server*. Tipe kedua adalah *Document Request* yaitu sebuah dokumen XML dilewatkan ke atau dari *client* dan *server*, di dalam SOAP *message*.

### 2.2.2 REST

REST (*Representational State Transfer*) adalah salah satu jenis *web service* yang menerapkan konsep perpindahan antara *state*. *State* di sini dapat digambarkan seperti jika *browser* meminta suatu halaman *web*, maka *server* akan mengirimkan *state* halaman *web* yang sekarang ke *browser*. Bernavigasi melalui *link-link* yang disediakan sama halnya dengan mengganti *state* dari halaman *web*. Begitu pula REST bekerja, dengan bernavigasi melalui *link-link* HTTP untuk melakukan aktivitas tertentu, seakan-akan terjadi perpindahan *state* satu sama lain. Perintah HTTP yang bisa digunakan adalah fungsi GET, POST, PUT, dan DELETE. Proses pengiriman dan penerimaan data dalam bentuk XML atau JSON tanpa ada protokol pemaketan data, sehingga informasi yang diterima lebih mudah dibaca dan di-*parsing* di sisi *client*.

Dalam pengaplikasiannya, REST lebih banyak digunakan untuk *web service* yang berorientasi pada *resource*. Maksudnya orientasi pada *resource* adalah orientasi yang menyediakan *resource-resource* sebagai layanannya dan bukan kumpulan dari aktifitas yang mengolah *resource* itu.

### 2.3 JSON

JSON (JavaScript Object Notation) adalah sebuah format *data-interchange* yang ringan. JSON merupakan bagian dari bahasa pemrograman JavaScript (Standard ECMA-262 3rd Edition – December 1999). JSON merupakan format teks yang sepenuhnya independen tetapi menggunakan konvensi yang familiar dengan bahasa pemrograman dari keluarga-C, termasuk C,

C++, C#, Java, JavaScript, Perl, Python, dan sebagainya. Kelebihan inilah yang membuat JSON menjadi sebuah bahasa data-interchange yang ideal. Format JSON dispesifikasikan di RFC 4627 oleh Douglas Crockford. Tipe media Internet resmi JSON adalah `application/json` sedangkan ekstensi berkasnya adalah `.json`.

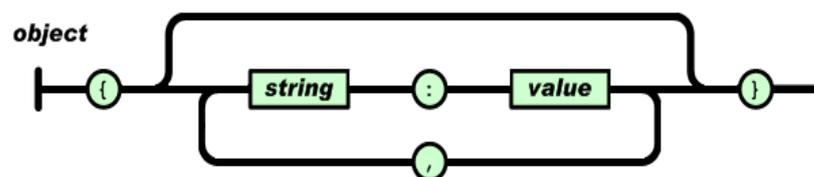
JSON dibangun dalam dua struktur:

1. Beberapa pasangan dari nama/ nilai. Dalam beberapa bahasa perograman biasa disebut dengan istilah *object*, *record*, *struct*, *table hash*, *key list* atau *associative array*.
2. Nilai-nilai yang terusun secara *ordered list*. Biasa disebut dengan *array*, *vector*, *list* atau daftar dalam bahasa pemrograman.

JSON adalah struktur data yang universal, dalam artian bisa digunakan dalam berbagai bahasa pemrograman. Hampir semua bahasa pemrograman mendukung penuh JSON dalam berbagai format. Hal ini memungkinkan format data yang dapat dipertukarkan menggunakan bahasa pemrograman juga menggunakan dasar dari struktur JSON.

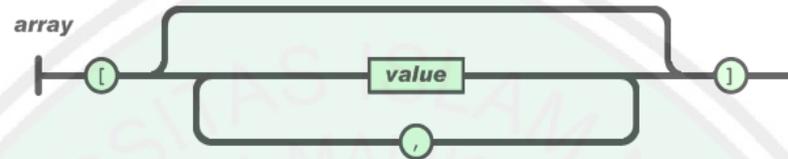
Format data JSON mempunyai aturan sebagai berikut:

*Object* adalah satu set nama/nilai yang tidak terurut (An object is an unordered set of name/value pairs). Penulisan object dimulai dengan tanda { (left brace) dan diakhiri dengan tanda } (right brace). Setiap nama diikuti oleh tanda : (colon) dan pasangan nama/nilai dipisahkan dengan tanda , (comma).



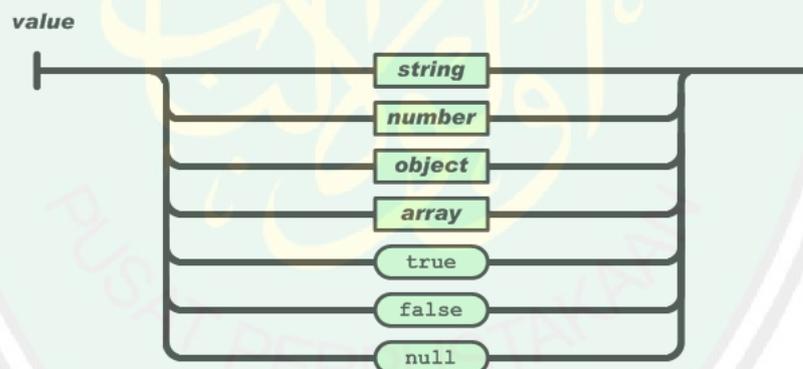
Gambar 2.1 Object dalam JSON (json.org)

1. *Array* adalah sekumpulan nilai yang teratur (*An array is an ordered collection of values*). Penulisan sebuah array dimulai dengan tanda [ (*left bracket*) dan diakhiri dengan tanda ] (*right bracket*). Nilai dipisahkan menggunakan tanda , (*comma*).



Gambar 2.2 Array dalam JSON (json.org)

2. *Value* atau nilai bisa berupa string dalam tanda kutip, atau number (angka), TRUE atau FALSE atau NULL, sebuah object atau sebuah array. Struktur ini dapat ditulis menggunakan metode bersarang.



Gambar 2.3 Value dalam JSON (json.org)

3. *String* adalah rangkaian atau urutan karakter unicode yang berada dalam tanda kutip, bisa juga hanya berisi karakter kosong, menggunakan tanda \ (*backslash*) untuk escape. Karakter di representasikan sebagai string tunggal. Tipe data string pada JSON sangat mirip dengan definisi pada bahasa C atau Java.



## 2.4 JSON VS XML

Mengingat kedua format tersebut biasa digunakan dalam pertukaran data, berikut perbandingan kedua format tersebut dilihat dari beberapa aspek.

Tabel 2.1 Karakteristik XML vs. JSON (navioo.com)

Karakteristik	XML	JSON
Tipe Data	Tidak memberikan pengertian dari tipe data. Salah satu harus bergantung pada Skema XML untuk menambahkan jenis informasi.	Menyediakan tipe data skalar dan kemampuan untuk mengekspresikan data terstruktur melalui array dan objek.
Dukungan Array	Array harus diekspresikan melalui konvensi, misalnya melalui penggunaan elemen luar pengganti yang memodelkan isi array sebagai elemen inti. Biasanya, elemen luar menggunakan bentuk jamak dari nama yang digunakan untuk elemen inti.	Mendukung Array secara langsung ( <i>native support</i> ).
Dukungan Objek	Obyek harus diekspresikan melalui konvensi, seringkali melalui penggunaan campuran	Mendukung Objek secara langsung ( <i>native support</i> ).

	atribut dan elemen.	
Dukungan Null	Membutuhkan penggunaan <b>XSI:nil</b> pada unsur-unsur dalam sebuah dokumen contoh XML ditambah impor dari namespace yang sesuai.	Mengenali null secara langsung ( <i>native support</i> ).
Komentar	Dukungan secara langsung ( <i>native support</i> ) dan biasanya tersedia melalui API.	Tidak mendukung komentar.
Namespaces	Mendukung namespaces, untuk menghilangkan resiko tabrakan nama ketika menggabungkan dokumen. Namespaces juga memungkinkan adanya perluasan standar berbasis XML dengan aman.	Tidak ada konsep namespaces, tabrakan penamaan biasanya dihindari dengan objek bersarang atau menggunakan awalan nama anggota objek.
Format keputusan	Kompleks. Membutuhkan upaya yang lebih besar untuk memutuskan bagaimana memetakan jenis aplikasi untuk elemen XML dan	Sederhana. Menyediakan pemetaan secara langsung untuk data aplikasi. Satu-satunya pengecualian tidak adanya tanggal/ waktu literal.

	atribut. Dapat menciptakan perdebatan apakah pendekatan elemen-sentris atau atribut-sentris yang lebih baik.	
Ukuran	Dokumen cenderung besar dalam ukuran, terutama ketika pendekatan elemen-sentris untuk format yang digunakan.	Sintaks yang sangat singkat dan hasil format teks yang sebagian besar ruang digunakan untuk mempresentasikan data.
Parsing dalam Javascript	Membutuhkan implementasi XML DOM dan kode aplikasi tambahan untuk memetakan teks kembali ke objek JavaScript.	Tidak ada kode aplikasi tambahan yang dibutuhkan untuk mengurai teks, dapat menggunakan JavaScript <b>eval</b> fungsi.
Kurva Pembelajaran	Umumnya cenderung memerlukan penggunaan beberapa teknologi yang diizinkan: XPath, XML Schema, XSLT, Namespaces XML, yang DOM, dan sebagainya.	Sangat sederhana stack teknologi yang sudah akrab bagi pengembang dengan latar belakang JavaScript atau bahasa pemrograman dinamis.

Pada tabel 2.1 dijelaskan perbedaan karakteristik JSON dengan XML. Sedangkan untuk format penulisan JSON dengan XML bisa dilihat pada table berikut.

Tabel 2.2 Format XML vs JSON (navioo.com)

Versi XML	Versi JSON
<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;root&gt;   &lt;data&gt;     &lt;id&gt;1&lt;/id&gt;     &lt;name&gt;PHP &lt;/name&gt;   &lt;/data&gt;   &lt;data&gt;     &lt;id&gt;2&lt;/id&gt;     &lt;name&gt;Table of Contents&lt;/name&gt;   &lt;/data&gt;   &lt;data&gt;     &lt;id&gt;7&lt;/id&gt;     &lt;name&gt;Getting Started&lt;/name&gt;   &lt;/data&gt;   &lt;data&gt;     &lt;id&gt;8&lt;/id&gt;     &lt;name&gt;Introduction&lt;/name&gt;   &lt;/data&gt;   &lt;data&gt;     &lt;id&gt;9&lt;/id&gt;     &lt;name&gt;What is PHP?&lt;/name&gt;   &lt;/data&gt;   &lt;data&gt;     &lt;id&gt;10&lt;/id&gt;     &lt;name&gt;What can PHP do?&lt;/name&gt;   &lt;/data&gt;   &lt;data&gt;     &lt;id&gt;11&lt;/id&gt;     &lt;name&gt;A simple tutorial&lt;/name&gt;   &lt;/data&gt;   &lt;data&gt;     &lt;id&gt;12&lt;/id&gt;     &lt;name&gt;What do I need?&lt;/name&gt;   &lt;/data&gt; &lt;/root&gt; </pre>	<pre> {   "data": [     {       "id" : "1",       "name" : "PHP"     },     {       "id" : "2",       "name" : "Table of Contents"     },     {       "id" : "7",       "name" : "Getting Started"     },     {       "id" : "8",       "name" : "Introduction"     },     {       "id" : "9",       "name" : "What is PHP?"     },     {       "id" : "10",       "name" : "What can PHP do?"     },     {       "id" : "11",       "name" : "A simple tutorial"     },     {       "id" : "12",       "name" : "What do I need?"     }   ] } </pre>

Berikut perbandingan hasil *benchmark* JSON vs. XML yang bersumber dari navioo.com. Benchmark dilakukan dengan dua skenario pada beberapa browser.

1. Skenario pertama (Menerima 1000 baris dan 2 kolom dari tabel database)  
test.js => 55.35 Kb vs. test.xml => 127.39 KB
2. Skenario kedua (Menerima 1000 baris dan 4 kolom dari table database)  
test4.js => 124.21 KB vs. test4.xml ->249.27 KB

Untuk hasil *benchmark* dapat diketahui bahwa parsing JSON lebih cepat 2-10 kali dibanding XML. Untuk lebih detailnya dapat dilihat pada table berikut.

Tabel 2.3 *Benchmark* JSON vs. XML (navioo.com)

Browser	Skenario	AJAX JSON	AJAX XML
FIREFOX 3	1	jsonT.data.length= 1000 jsonT.data[999].name= RESPONSE FROM SERVER Total time JSON in = 47 ms	XML.length= 1000 XML.data[999].name= RESPONSE FROM SERVER Total time XML in = 198 ms
	2	json.data.length= 1000 json.data[900].name= RESPONSE FROM SERVER() Total time JSON in = 68 ms	XML.length= 1000 XML.data[900].name=RESPONSE FROM SERVER() Total time XML in =768 ms
INTERNET EXPLORER	1	jsonT.data.length= 1000 jsonT.data[900].name= RESPONSE FROM SERVER Total time JSON in = 43 ms	XML.length= 1000 XML.data[900].name= RESPONSE FROM SERVER Total time XML in = 196 ms
	2	json.data.length= 1000 json.data[900].name= RESPONSE FROM SERVER() Total time JSON in = 77 ms	XML.length= 1000 XML.data[900].name=RESPONSE FROM SERVER() Total time XML in =770 ms
SAFARI	1	jsonT.data.length= 1000 jsonT.data[900].name= RESPONSE FROM SERVER Total time JSON in = 48 ms	XML.length= 1000 XML.data[900].name= RESPONSE FROM SERVER Total time XML in = 207 ms
	2	json.data.length= 1000 json.data[900].name= RESPONSE FROM SERVER() Total time JSON in = 69 ms	XML.length= 1000 XML.data[900].name=RESPONSE FROM SERVER() Total time XML in =827 ms
CHROME	1	jsonT.data.length= 1000 jsonT.data[900].name= RESPONSE FROM SERVER Total time JSON in = 49 ms	XML.length= 1000 XML.data[900].name= RESPONSE FROM SERVER Total time XML in = 211 ms
	2	json.data.length= 1000 json.data[900].name= RESPONSE FROM SERVER() Total time JSON in = 72 ms	XML.length= 1000 XML.data[900].name=RESPONSE FROM SERVER() Total time XML in =810 ms

## 2.5 Android

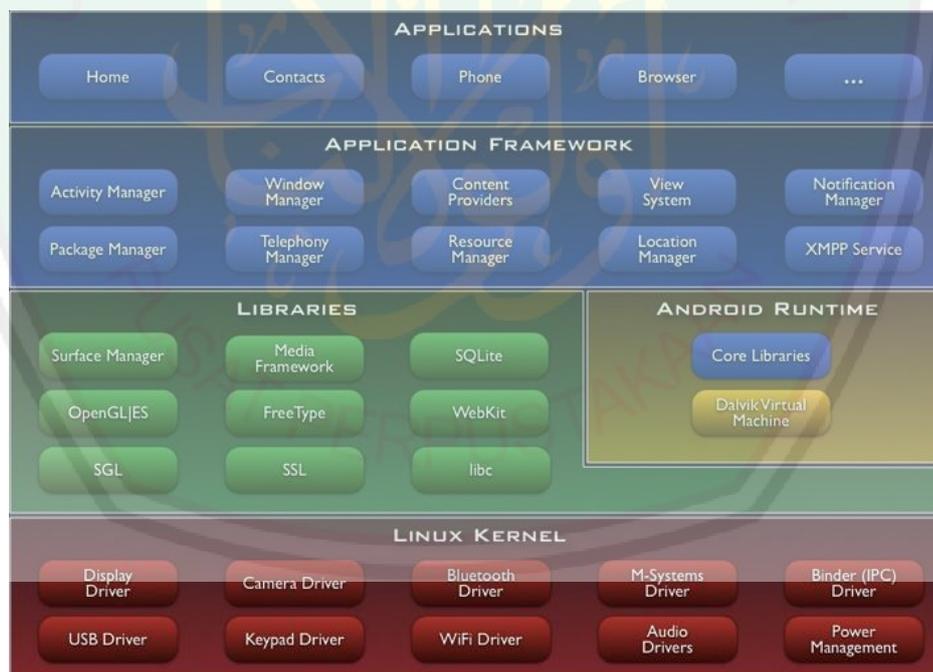
Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Google merilis kode-kode Android di bawah lisensi Apache, sebuah

lisensi perangkat lunak dan standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

### 2.5.1 Arsitektur Android

Sistem Operasi Android memiliki 4 lapisan (layer) yang merupakan komponen sistem Android. Gambar berikut merupakan lapisan arsitektur sistem operasi Android:



Gambar 2.6 Arsitektur Android (Smieh, 2012)

Google mengibaratkan Android sebagai sebuah tumpukan software. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi.

### 2.5.1.1 Linux Kernel

Tumpukan paling bawah pada arsitektur Android ini adalah kernel. Google menggunakan kernel Linux untuk membangun sistem Android, yang mencakup *memory management*, *security setting*, *power management*, dan beberapa *driver hardware*. Kernel berperan sebagai *abstraction layer* antara *hardware* dan keseluruhan *software*.

Secara keseluruhan android bukanlah linux, karena dalam android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada android hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan *driver*. Kernel linux menyediakan *driver* layar, kamera, *keypad*, WiFi, *Flash Memory*, audio, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan.

### 2.5.1.2 Android Runtime

Lapisan setelah Kernel Linux adalah Android *Runtime*. Android *Runtime* ini berisi *Core Libraries* dan *Dalvik Virtual Machine*.

1. *Core Libraries*, mencakup serangkaian inti library Java, artinya Android menyertakan satu set *library-library* dasar yang menyediakan sebagian besar fungsi-fungsi yang ada pada *library-library* dasar bahasa pemrograman Java.

2. *Dalvik Virtual Machine*, Dalvik adalah *Java Virtual Machine* yang memberi kekuatan pada sistem Android. Dalvik VM ini di optimalkan untuk telepon seluler. Setiap aplikasi yang berjalan pada Android berjalan pada processnya sendiri, dengan instance dari *Dalvik Virtual Machine*. Dalvik telah dibuat sehingga sebuah piranti yang memakainya dapat menjalankan multi *Virtual Machine* dengan efisien. Dalvik VM dapat mengeksekusi file dengan format *Dalvik Executable* (.dex) yang telah dioptimasi untuk menggunakan minimal *memory footprint*. *Virtual Machine* ini *register-based*, dan menjalankan *class-class* yang dicompile menggunakan *compiler* Java yang kemudian ditransformasi menjadi format .dex menggunakan “dx” tool yang telah disertakan. *Dalvik Virtual Machine* (VM) menggunakan kernel Linux untuk menjalankan fungsi-fungsi seperti *threading* dan *low-level memory management*.

### 2.5.1.3 Libraries

Bertempat di level yang sama dengan Android *Runtime* adalah *Libraries*. Android menyertakan satu set library-library dalam bahasa C/C++ yang digunakan oleh berbagai komponen yang ada pada sistem Android. Kemampuan ini dapat diakses oleh programmer melewati Android *application framework*.

*Library-library* tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi Android 1.5, pengembang dapat membuat dan menggunakan pustaka sendiri menggunakan *Native Development Toolkit* (NDK).

#### 2.5.1.4 Application Framework

Lapisan selanjutnya adalah application framework, yang mencakup program untuk mengatur fungsi-fungsi dasar *smartphone*. Application Framework merupakan serangkaian tool dasar seperti alokasi *resource smartphone*, aplikasi telepon, pergantian antar – proses atau program, dan pelacakan lokasi fisik telepon. Para pengembang aplikasi memiliki aplikasi penuh kepada tool-tool dasar tersebut, dan memanfaatkannya untuk menciptakan aplikasi yang lebih kompleks.

#### 2.5.1.5 Application

Di lapisan teratas bercokol aplikasi itu sendiri. Di lapisan inilah anda menemukan fungsi-fungsi dasar *smartphone* seperti menelepon dan mengirim pesan singkat, menjalankan web browser, mengakses daftar kontak, dan lain-lain. Bagi rata-rata pengguna, lapisan inilah yang paling sering mereka akses. Mereka mengakses fungsi-fungsi dasar tersebut melalui *user interface*.

### 2.6 Bahasa Jawa

Bahasa Jawa adalah bahasa yang digunakan penduduk suku bangsa Jawa terutama di beberapa bagian Banten terutama di kabupaten Serang dan Tangerang, Jawa Barat khususnya kawasan Pantai utara terbentang dari pesisir utara Karawang, Subang, Indramayu dan Cirebon, Jawa Tengah & Jawa Timur di Indonesia.

Terdapat tiga bentuk utama variasi didalam bahasa, yaitu ngoko (kasar), madya (biasa), dan *krama* (halus). Pembagian semacam ini muncul pada masa

awal Kerajaan Mataram pimpinan Sultan Agung, padahal sebelumnya bahasa Jawa tidak mengenal susunan seperti tersebut.

Ngoko terbagi lagi menjadi 2 macam yaitu:

1. Ngoko lugu
2. Ngoko andhap atau ngoko alus

Madya terbagi lagi menjadi 3 macam yaitu:

1. Madya ngoko
2. Madyantara
3. Madya karma

Sedangkan karma terbagi lagi menjadi 6 macam yaitu:

1. Krama Lugu atau disebut “krama” saja
2. Mudha krama
3. Wredha krama
4. Krama inggil atau krama alus
5. Krama desa
6. Krama kedhaton

Namun secara garis besar (menurut kebanyakan orang Jawa) bahasa Jawa terbagi menjadi 3 macam yang umum digunakan, yaitu:

Krama lugu atau disebut “krama” saja, penggunaan Krama Lugu, umum diucapkan seorang anak kepada orang tua, murid kepada guru, bawahan kepada atasan, atau seseorang yang baru pertama kali bertemu. Bahasa ini lebih mudah dan masih sering dijumpai dalam percakapan dibandingkan dengan Krama Inggil.

Krama inggil, bahasa ini jarang sekali digunakan karena memang mempunyai aturan dan rumus tertentu sedangkan masyarakat Jawa modern sekarang ini, bisa dikatakan banyak yang tidak bisa daripada yang bisa. Hal ini bukan semata-mata tidak menghargai bahasa daerah, melainkan, jika bahasa yang biasanya diucapkan dalam acara resmi seperti pernikahan dengan adat Jawa murni ini dipakai dalam dialog sehari-hari, justru akan terasa kaku dan tak umum.

Ngoko, bahasa inilah yang banyak digunakan dalam dialog sehari-hari masyarakat Jawa. Bahasa Ngoko (Jawa kasar) umum dipakai dalam percakapan sesama teman, orang seusia, atau orang yang sudah akrab satu sama lain. Meskipun disebut Jawa kasar, bahasa ini bukan serta merta didefinisikan sebagai bahasa yang tak sopan, melainkan jika dibandingkan Krama Inggil dan Krama Lugu, bahasa ini memang yang paling mudah.

Menurut Poedjosoedarsono dkk (1979), bahasa ngoko sebenarnya adalah dasar dari semua kosakata bahasa Jawa. Karena itulah, maka bahasa krama tidak dapat disamakan atau disetarakan dengan ragam ngoko, karena terlalu banyaknya kata ngoko yang tidak memiliki padanan kromonya. Jumlah kosakata ngoko mencapai ratusan ribu, sedangkan krama hanya berjumlah 850 kata, bahkan untuk krama Inggil hanya berjumlah 250 kata.

## BAB III

### METODE PENELITIAN DAN PERANCANGAN SISTEM

#### 3.1 Analisa Permasalahan Sistem

Ada banyak model atau bentuk aplikasi kamus yang beredar saat ini, yang pertama dan paling sering kita jumpai adalah aplikasi kamus untuk desktop komputer atau laptop (*native OS* seperti Windows, MacOS, Linux, dll), yang kedua adalah kamus online yang bisa kita akses melalui web browser dan harus ada koneksi internet, yang ketiga adalah aplikasi kamus *mobile* untuk perangkat bergerak seperti *handphone* atau *smartphone*.

Aplikasi kamus model pertama untuk native OS harus di *install* pada komputer atau laptop sesuai dengan OS-nya, kelemahan model aplikasi kamus untuk desktop komputer atau laptop ini adalah tidak praktis saat mau dibawa bepergian terutama yang diinstall pada komputer ini mustahil untuk dibawa bepergian semisal berwisata, yang menginstall aplikasi tersebut pada laptop masih dirasa tidak praktis jika untuk dibawa bepergian. Selain tidak praktis untuk dibawa bepergian untuk pengoperasiannya juga butuh waktu lebih mulai dari menunggu *booting* OS sampai tampil desktop dan baru memulai aplikasi.

Aplikasi model kedua ini bisa diakses melalui *web browser* baik browser komputer atau browser *handphone* atau *smartphone*. Untuk aplikasi model kedua ini bisa menjadi lebih praktis untuk dibawa bepergian karena bisa diakses melalui *web browser handphone* atau *smartphone*. Hanya satu kelemahan model aplikasi kamus *online* ini harus selalu terhubung ke internet dan jika berada pada tempat

yang tidak ada sinyal operator (tidak ada koneksi data internet) maka aplikasi ini tidak bisa digunakan.

Aplikasi model ketiga aplikasi kamus *mobile*, ini adalah yang paling unggul dimana memiliki kepraktisan selayaknya kamus asli yang mudah dibawa bepergian. Ada banyak *platform mobile* aplikasi (*Smartphone OS*), oleh karena itu perlu dipertimbangkan akan dibangun untuk platform apakah kamus *mobile* tersebut. Dan pilihan penulis jatuh pada platform Android dengan pertimbangan Android saat ini adalah *platform* yang paling banyak digunakan.

Aplikasi kamus *mobile* yang dibuat ini harus memiliki database tersendiri agar proses penerjemahan bisa dilakukan ketika tidak ada koneksi internet (tidak harus *online*). Kebanyakan update aplikasi kamus terjadi jika ada penambahan kata pada database dan ini seharusnya tidak perlu melakukan *update* dengan *download* ulang aplikasi secara keseluruhan. Untuk permasalahan ini maka diperlukan kamus *mobile* dengan kemampuan sinkronisasi data dengan *server* (database) agar supaya *update* data bisa dilakukan disisi database saja.

### 3.2 Analisa Kebutuhan

Pada pembahasan berikut ini akan dianalisa apa saja kebutuhan untuk mengembangkan atau membangun aplikasi kamus *mobile* tersebut, baik dari perangkat lunak (*software*) dan juga perangkat keras (*hardware*).

### 3.2.1 Software

*Software* adalah program atau aplikasi komputer yang dibutuhkan untuk membangun sebuah sistem. *Software* yang dibutuhkan untuk proses pembuatan aplikasi *mobile* learning antara lain sebagai berikut :

- a. Sistem Operasi, merupakan sistem yang menjembatani antara komputer dan user. Ada beberapa pilihan sistem operasi yang bisa digunakan untuk mengembangkan aplikasi kamus *mobile* pada *platform* android, yaitu Windows, MacOS, dan Linux. Penulis pada pembuatan atau pengembangan aplikasi kamus *mobile* ini menggunakan sistem operasi Linux distro Elementary OS Luna.
- b. *Java Development Kit* (JDK), merupakan paket *platform* java yang terdiri dari berbagai macam *library*, JVM, *compiler*, dan debugger. JDK ini wajib diinstal karena merupakan *dependency* untuk bisa menjalankan Android SDK.
- c. Android SDK, merupakan *tools* yang digunakan untuk mengembangkan aplikasi berbasis android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode, dan *tutorial*.
- d. Eclipse ADT (*Android Development Tools*), Eclipse adalah sebuah IDE (*Integrated Development Environment*) yang bisa digunakan untuk mengembangkan untuk mengembangkan aplikasi android dengan tambahan ADT plugin sehingga bisa terintegrasi dengan Android SDK.
- e. *Android Virtual Device*, merupakan sebuah mesin *virtual* yang mengemulasikan *device* android berfungsi untuk menguji aplikasi android

yang sedang dikembangkan pada komputer tanpa harus menggunakan *device* (*smartphone*) yang sebenarnya. Pada android SDK sendiri sudah disertakan *Virtual Device* bawaan akan tetapi penulis disini memilih menggunakan Genymotion dengan pertimbangan jauh lebih ringan untuk dijalankan.

- f. Apache *Web Server*, digunakan sebagai *server* aplikasi *web service* yang dibangun menggunakan bahasa pemrograman PHP dan digunakan dalam pengembangan *web service* tersebut
- g. MySQL database, digunakan untuk menyimpan database kamus yang berfungsi sebagai database *server* untuk proses sinkronisasi.
- h. *Web Browser*, digunakan untuk administrasi database MySQL dengan menggunakan phpMyAdmin pada *server* Apache.

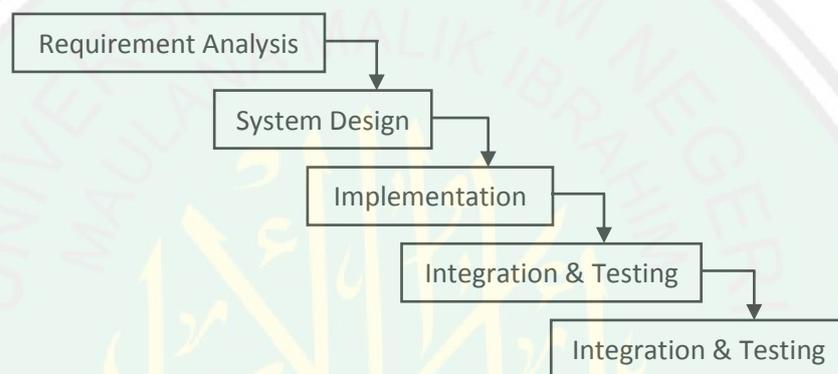
### 3.2.2 Hardware

Hardware adalah perangkat keras atau *device* yang digunakan untuk menunjang dalam pembuatan sebuah sistem. Dalam pembuatan aplikasi kamus *mobile*, hardware yang dibutuhkan antara lain :

- a. Komputer / Laptop, digunakan untuk membangun sistem aplikasi kamus *mobile* dan *web service*.
- b. *Smartphone* Android, digunakan untuk mengaplikasikan program yang telah dibuat untuk mengetahui apakah program tersebut telah memenuhi spesifikasi yang diinginkan. Program yang berjalan di *Virtual Device*, belum tentu dapat berjalan dengan baik di *smartphone* yang sesungguhnya.

### 3.3 Tahapan Pembuatan

Dalam pengembangan aplikasi kamus *mobile* Jawa-Indonesia dengan sinkronisasi database menggunakan REST *Web Service* ini penulis menggunakan metode pengembangan perangkat lunak *Waterfall*. Adapun fase-fase pengembangan perangkat lunak dengan metode *Waterfall* dapat dilihat pada gambar 3.1 berikut.



Gambar 3.1 Fase-fase metode *Waterfall*

Adapun penjelasan setiap fase-fase pada metode *Waterfall* sehubungan dengan yang penulis lakukan dalam pengembangan aplikasi kamus *mobile* ini adalah sebagai berikut:

1. *Requirement Analysis*, pada tahap ini penulis menganalisis seluruh kebutuhan, termasuk didalamnya kegunaan *software* yang diharapkan pengguna dan batasan *software*.
2. *System Design*, pada tahap ini penulis lakukan sebelum melakukan coding. Tahap ini bertujuan untuk memberikan gambaran apa yang seharusnya dikerjakan dan bagaimana tampilannya.
3. *Implementation*, dalam tahap ini penulis mulai melakukan pemrograman (*coding*). Pengembangan *software* dipecah menjadi

modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya.

4. *Integration & Testing*, di tahap ini penulis melakukan penggabungan modul-modul yang sudah dibuat dan melakukan pengujian untuk mengetahui apakah *software* yang dibuat sudah sesuai dengan desainnya dan masih terdapat kesalahan atau tidak.
5. *Operation & Maintenance*, pada tahap terakhir dalam model *Waterfall* ini *software* yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. kebutuhan baru.

### 3.4 Spesifikasi Aplikasi

Aplikasi kamus *mobile* Jawa Indonesia ini dirancang atau dikembangkan dengan memiliki kemampuan sebagai berikut :

1. Aplikasi kamus *mobile* memiliki fungsi dasar yaitu untuk menerjemahkan kata dari bahasa Jawa ke bahasa Indonesia.
2. Aplikasi kamus *mobile* sebagai *client* memiliki fungsi atau kemampuan untuk mensinkronkan database lokal (SQLite) dengan database yang ada di *server* (MySQL).
3. Aplikasi pada *server* dengan teknologi REST *Web Service* memiliki fungsi untuk menjawab *request* dari aplikasi *client* (kamus *mobile*) dalam proses sinkronisasi.

### 3.5 Spesisikasi Pengguna

Aplikasi kamus *mobile* ini dapat digunakan oleh semua orang yang memiliki *smartphone* android dengan android versi minimal 2.3 Gingerbread dan diatasnya.

### 3.6 Deskripsi Sistem

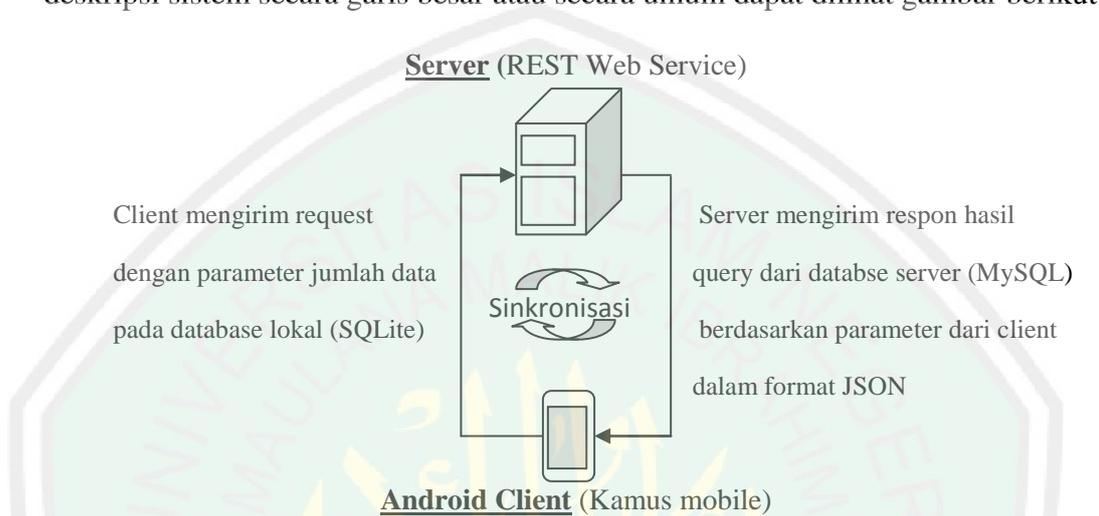
Sistem yang dibuat terbagi menjadi dua buah sub-sistem, yaitu *client-server*. Dalam hal ini *server* berupa web database yang berfungsi sebagai penyedia layanan (*web service*) dengan teknologi REST *Web Service*. Aplikasi *client* adalah kamus translasi Jawa-Indonesia.

Sesuai dengan penjelasan sebelumnya, sistem ini dikembangkan dalam dua bagian *client* dan *server*. *Client* dalam hal ini adalah aplikasi *mobile* berbasis Android yang memiliki fitur secara umum.

1. Mampu melakukan translasi secara *offline* dengan menggunakan database lokal (SQLite).
2. Mampu melakukan sinkronisasi database lokal (SQLite) dengan database *server* (MySQL) dengan mengambil data terbaru di *server* dan belum ada pada database lokal.

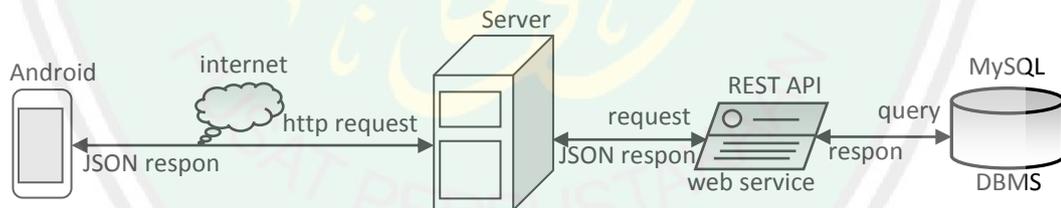
Sedangkan untuk aplikasi *server* dibangun menggunakan teknologi PHP-MySQL dengan memiliki fitur utama yaitu sebagai *web service* yang menyediakan layanan sumber update database untuk proses sinkronisasi dengan aplikasi *client*.

Aplikasi *client-server* memerlukan koneksi internet agar aplikasi *client* bisa melakukan *request* ke *server* dan menerima respon dari *server*. Adapun deskripsi sistem secara garis besar atau secara umum dapat dilihat gambar berikut.



Gambar 3.2 Deskripsi sistem secara umum

Untuk gambaran lebih jelas tentang proses komunikasi dan pertukaran data antara *client-server* dapat dilihat pada gambar 3.3 berikut:



Gambar 3.3 Skema proses komunikasi dan pertukaran data

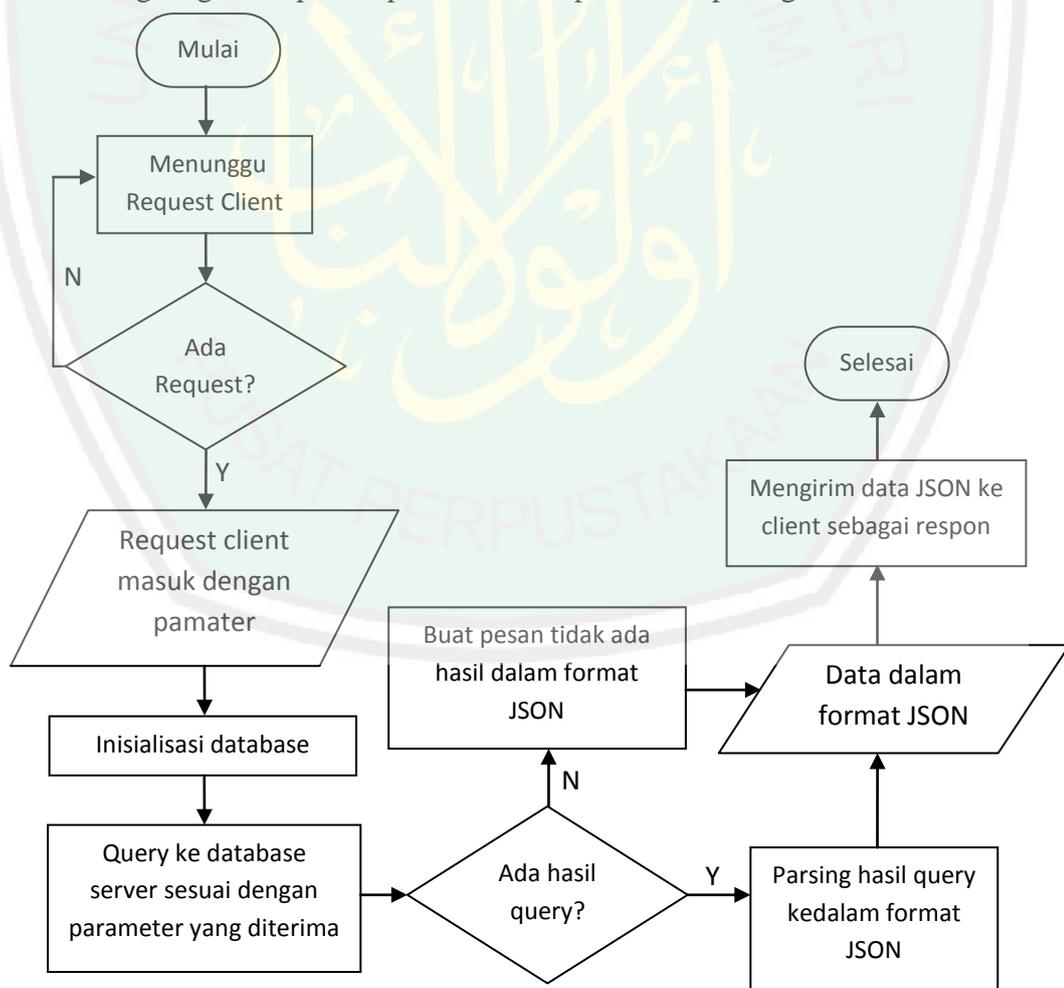
### 3.7 Desain Sistem

Untuk membangun system ini dibutuhkan dua perangkat lunak yang menjadi pendukung kelengkapan dari sistem ini. Aplikasi tersebut diwujudkan dalam bentuk aplikasi *client-server*.

### 3.7.1 Server

Perangkat lunak ini berbentuk web database berbasis PHP-MySQL dan berfungsi sebagai sumber database dari kamus Jawa-Indonesia dalam menyediakan layanan untuk *client* melalui *web service*-nya.

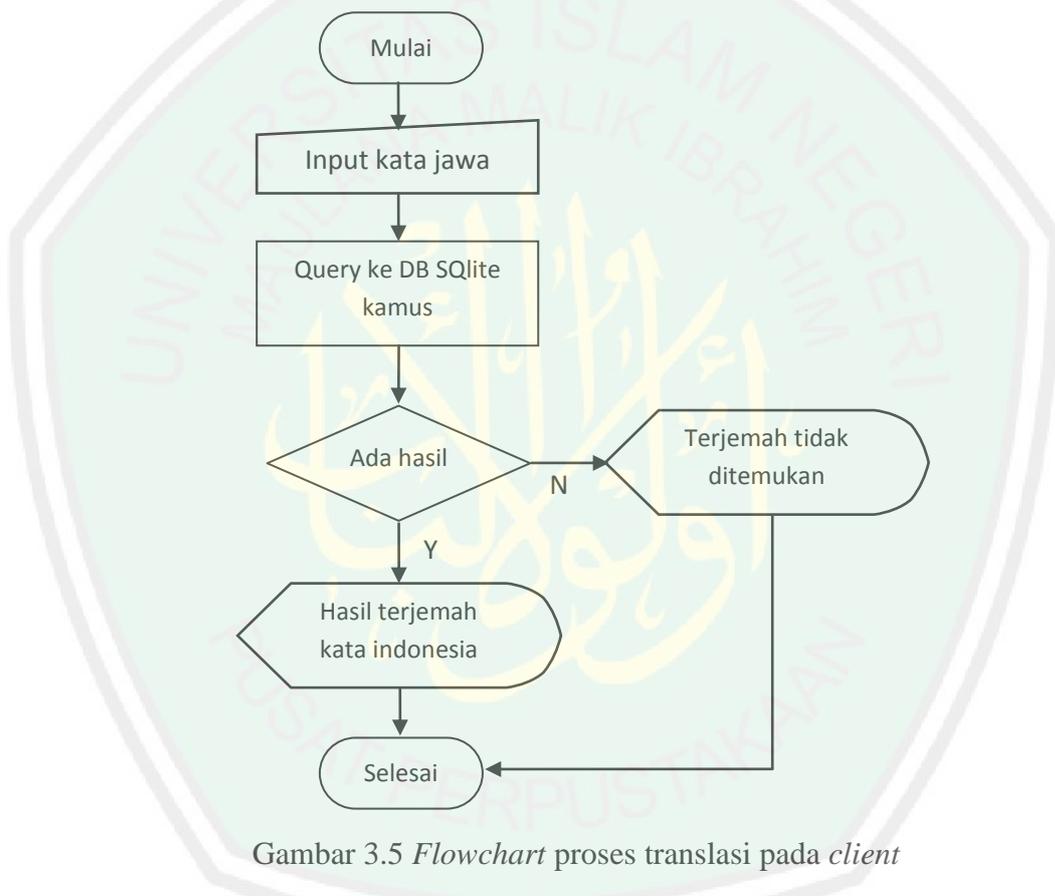
*Server* ini sendiri tidak dibuatkan *interface* khusus untuk memmanage database di *server*. Fungsi utama *server* ini adalah sebagai *web service* dan tugas utamanya hanya memberikan layanan saat *client* melakukan request. Jadi untuk memmanage database di *server* dilakukan melalui interface web phpMyAdmin secara langsung. Alur proses pada *server* dapat dilihat pada gambar 3.4 berikut.



Gambar 3.4 Flowchart alur proses pada *server*

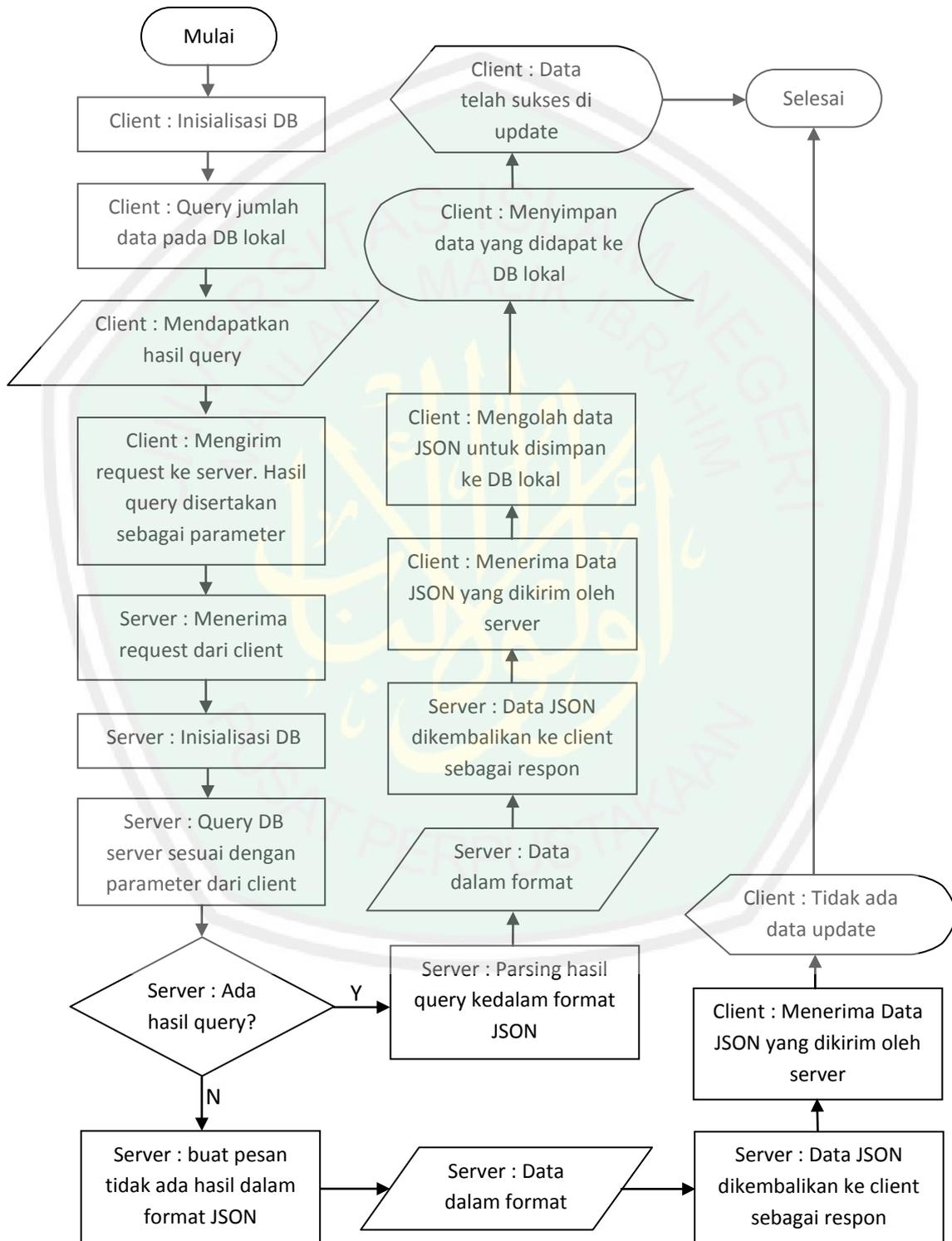
### 3.7.2 Client

Perangkat lunak ini berbentuk aplikasi *mobile* berbasis android yang memiliki database lokal kamus Jawa-Indonesia berbasis SQLite dengan kemampuan sinkronisasi database dengan *server* melalui layanan *web service*. Adapun proses translasi kata Jawa-Indonesia dapat dilihat pada gambar berikut.



Kemampuan proses sinkronisasi database lokal (SQLite) pada kamus *mobile* dengan database *server* (MySQL) dilakukan dengan cara *client* mengirim *request* ke *server* dengan parameter jumlah data yang ada pada SQLite untuk selanjutnya dijadikan *request* ke *server* sehingga *server* bisa melakukan *query* database sesuai dengan parameter yang diterima dan mendapatkan hasil *query*

data yang akan dikembalikan ke *client* dalam format JSON. Untuk lebih jelasnya akan dijelaskan dalam gambar alur proses berikut.



Gambar 3.6 Flowchart proses sinkronisasi database

### 3.8 Desain Interface

Dalam mendesain sebuah sistem, salah satu hal yang perlu diperhatikan adalah rancangan tersebut harus dapat memudahkan user dalam menggunakan sistem aplikasi yang dibuat. Sehingga perlu diperhatikan dalam mengatur letak *button*, *textfield*, menu ataupun komponen visual yang lain sehingga tidak membingungkan *user* dalam pemakaian. Berikut adalah perancangan menu utama sistem aplikasi kamus *mobile*:



Gambar 3.7 Desain *interface* menu utama kamus *mobile*

Gambar 3.7 adalah desain menu utama aplikasi kamus Jawa-Indonesia yang mencakup empat sub menu, dan sub menu yang berperan penting atau inti aplikasi adalah sub menu Kamus dan sub menu *Update* sedangkan untuk sub menu Bantuan dan Tentang hanya sebagai pelengkap.

### 3.8.1 Menu Kamus

Pada desain menu utama sebelumnya apabila dipilih sub menu Kamus maka akan ditampilkan user *interface* Kamus yang berfungsi untuk menterjemahkan kata dari bahasa Jawa ke dalam bahasa Indonesia. Adapun desain *interface* aplikasi kamus adalah sebagai berikut:

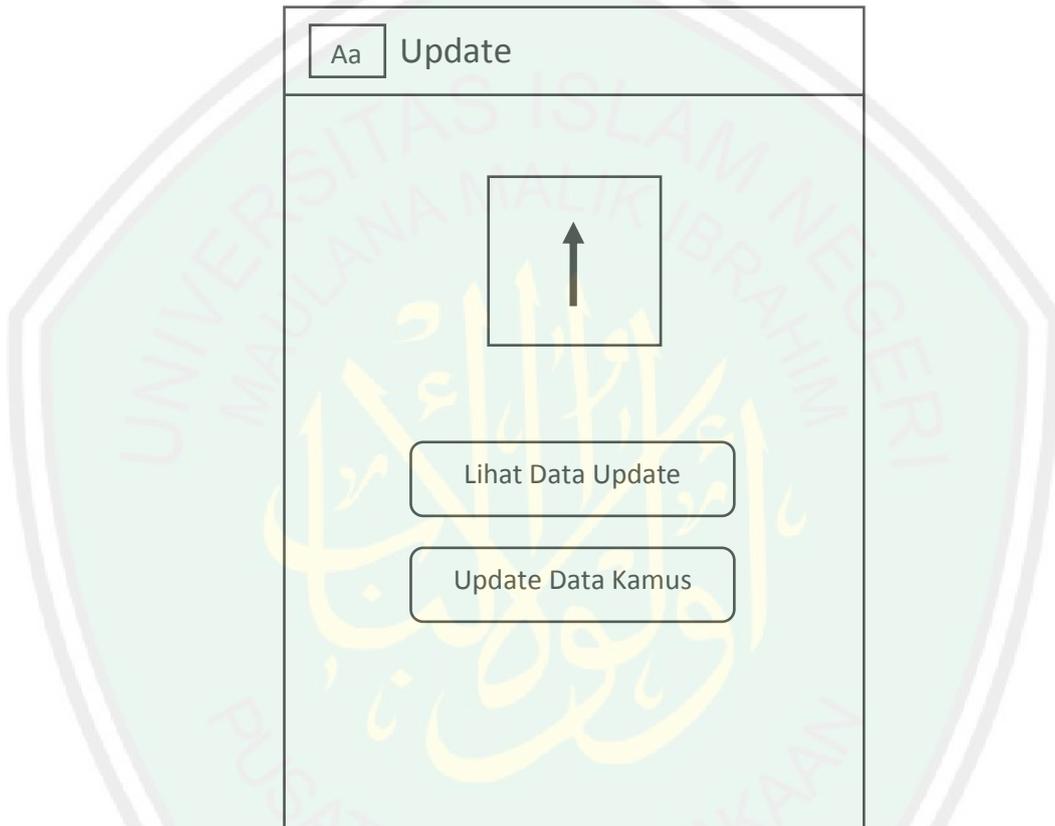


Gambar 3.8 Desain *interface* aplikasi Kamus

Pada gambar 3.8 desain *interface* aplikasi kamus terdapat satu *textfield* sebagai input “Kata Bahasa Jawa” yang ingin diterjemahkan dan juga terdapat satu tombol “Terjemahkan” yang berfungsi untuk memproses terjemahan kata Jawa-Indonesia dan hasilnya akan ditampilkan pada *textfield* “Hasil Terjemahan”.

### 3.8.2 Menu Update

Menu *Update* ini didesain untuk menangani proses sinkronisasi database kamus *mobile* dengan database *server*. Adapaun desain *interface* untuk aplikasi *Update* adalah sebagai berikut:

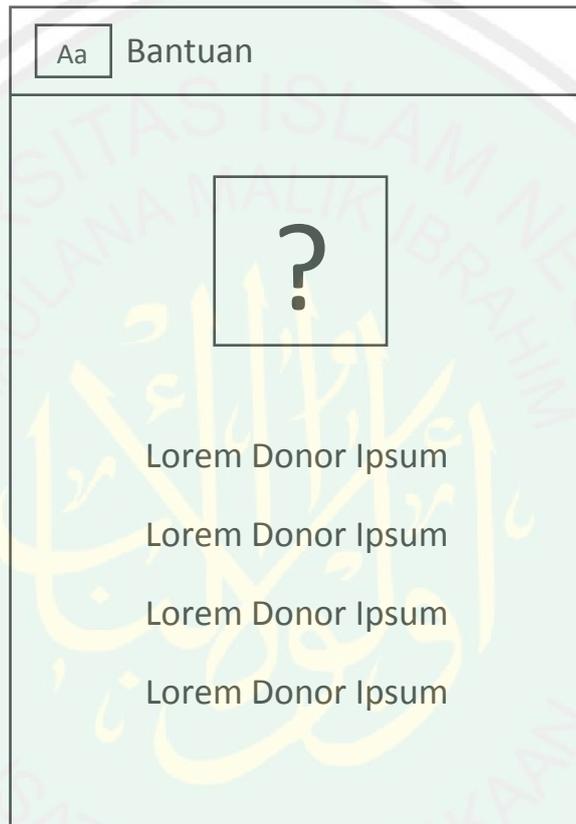


Gambar 3.9 Desain *interface* aplikasi *Update*

Pada desain *interface* aplikasi kamus ini terdapat ikon dibagian atas dan dua tombol dibawahnya, yaitu tombol “Lihat Data Update” dan tombol “Update Data Kamus”.

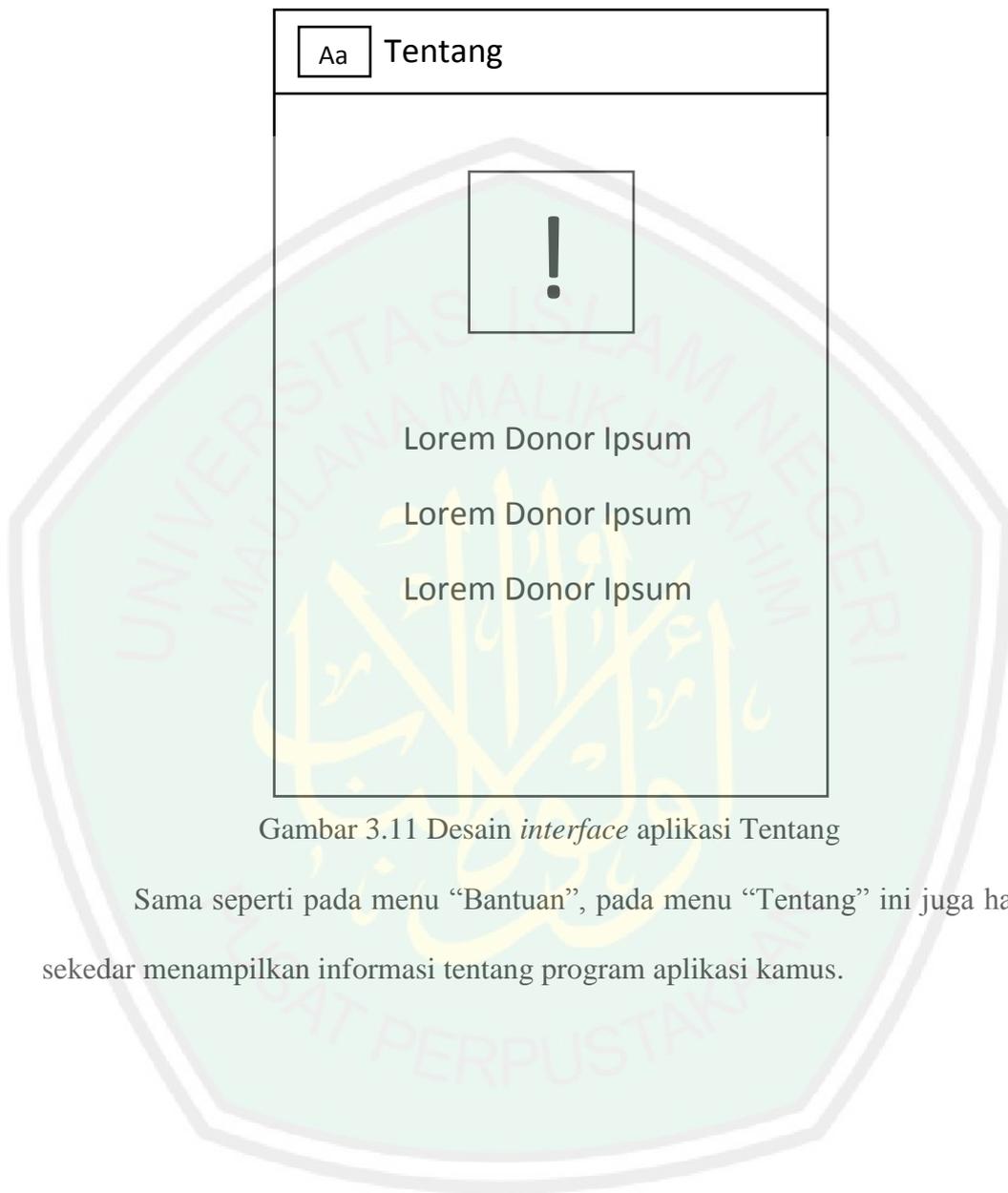
### 3.8.3 Menu Bantuan dan Tentang

Dua menu ini merupakan menu pelengkap saja, tidak ada *input*, *output* atau proses. Pada kedua menu tersebut hanya menampilkan informasi saja. Adapun desain menu Bantuan dan Tentang dapat dilihat pada gambar berikut:



Gambar 3.10 Desain *interface* menu Bantuan

Pada gambar 3.10 desain menu Bantuan hanya berisi informasi yang bisa membantu untuk pengguna aplikasi Kamus. Sedangkan untuk desain *interface* menu Tentang dapat dilihat pada gambar 3.10 berikut.



Gambar 3.11 Desain *interface* aplikasi Tentang

Sama seperti pada menu “Bantuan”, pada menu “Tentang” ini juga hanya sekedar menampilkan informasi tentang program aplikasi kamus.

## BAB IV

### HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai hasil uji coba terhadap sistem aplikasi kamus *mobile* yang telah dibuat. Uji coba ini bertujuan untuk mengetahui apakah sistem aplikasi yang dibuat telah dapat berjalan sebagaimana mestinya sesuai dengan skenario yang telah dijelaskan pada bab 3. Pada bab ini juga akan dibahas mengenai fitur dan tampilan yang terdapat pada aplikasi kamus *mobile* ini.

#### 4.1 Implementasi Sistem

Implementasi sistem adalah tahap untuk mewujudkan aplikasi melalui aktifitas pemrograman. Dalam tahap implementasi ini, sistem yang telah didesain mulai diterapkan dengan membangun komponen-komponen yang telah direncanakan.

##### 4.1.1 Implementasi Aplikasi Server REST Web Service

Aplikasi *server* database yang menyediakan layanan *update* database atau sinkronisasi diimplementasikan menggunakan bahasa pemrograman PHP dengan Apache *Web Server* serta database MySQL. Untuk protokol komunikasinya sendiri antara *client* dan *server* menggunakan protokol http, protokol yang umum digunakan pada internet. Teknologi *web service* yang digunakan pada aplikasi *server* adalah REST *web service* dengan JSON sebagai representasi data. Aplikasi *server* ini tidak dibuatkan *interface* untuk menangani database karena aplikasi

*server* ini dimaksudkan sebagai fungsi utamanya yaitu menyediakan layanan atau sebagai *service*, untuk penanganan database sendiri menggunakan phpMyAdmin secara langsung.

Untuk spesifikasi komputer atau laptop yang digunakan sebagai *server* adalah Acer 4750 dengan spesifikasi utama *processor* Intel Core i3-2310M dengan RAM 4GB dengan Sistem Operasi Linux Elementary OS Luna. Untuk *web server*-nya sendiri menggunakan Apache versi 2.4.7 dengan PHP versi 5.5.9 dan MySQL versi 5.0.11.

Berikut adalah kode program utama REST *Web Service* pada *server* yang berfungsi menangani *request* dari *client* dan memarsing data pada database MySQL ke dalam format JSON yang akan dikembalikan ke *client* sebagai respon. Kode program tersebut ditulis menggunakan bahasa pemrograman PHP dan disimpan di *web server* dengan nama “get\_all\_updates.php” yang nantinya diakses melalui URL oleh aplikasi *client* kamus *mobile*. Adapun kode program utama REST *Web Service* dapat dilihat pada gambar 4.1 berikut.

```

<?php

// array JSON response
$response = array();

// include db connect class
require_once __DIR__ . '/db_connect.php';

// menghubungkan database
$db = new DB_CONNECT();

// memeriksa post data
if (isset($_GET["uid"])) {
    $uid = $_GET['uid'];

    // mendapatkan data dari tabel kamus
    $result = mysql_query("SELECT * FROM javaind WHERE uid > $uid");

    if (!empty($result)) {
        // memeriksa jika tidak ada hasil
        if (mysql_num_rows($result) > 0) {
            // loop semua hasil
            $response["javaind"] = array();

            while ($row = mysql_fetch_array($result)) {
                // temp array
                $supkamus = array();
                $supkamus["uid"] = $row["uid"];
                $supkamus["katajawa"] = $row["katajawa"];
                $supkamus["kataindonesia"] = $row["kataindonesia"];

                array_push($response["javaind"], $supkamus);
            }
            // success
            $response["success"] = 1;

            // menampilkan JSON response
            echo json_encode($response);
        } else {
            // data tidak ditemukan
            $response["success"] = 0;
            $response["message"] = "Tidak ditemukan Update";

            echo json_encode($response);
        }
    } else {
        $response["success"] = 0;
        $response["message"] = "Tidak ditemukan Update";

        echo json_encode($response);
    }
} else {
    $response["success"] = 0;
    $response["message"] = "Isian yang dibutuhkan tidak ada";

    echo json_encode($response);
}
?>

```

Gambar 4.1 Kode program utama REST *Web Service*

Untuk struktur tabel kamus “javaind” yang diakses oleh REST *Web Service* sebagai database *server* dapat dilihat pada gambar 4.2 berikut.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	<u>uid</u>	int(7)			No	None	AUTO_INCREMENT
2	katajawa	varchar(25)	latin1_swedish_ci		No	None	
3	kataindonesia	varchar(250)	latin1_swedish_ci		No	None	

Gambar 4.2 Struktur Tabel kamus “javaind” pada *server*

#### 4.1.2 Implementasi Aplikasi Client Kamus Mobile

Untuk aplikasi kamus *mobile* Jawa-Indonesia ini implementasi sistem menggunakan bahasa *native* android yaitu bahasa pemrograman Java dengan database SQLite. Pada aplikasi kamus *mobile* ini diimplementasikan dua fungsi utama yaitu untuk menerjemahkan kata Jawa-Indonesia berdasarkan dari database lokal android SQLite dan fungsi yang kedua adalah kemampuan aplikasi kamus *mobile* untuk berkomunikasi dengan *server* yang menggunakan teknologi REST *Web Service* dengan database MySQL untuk pengambilan update database terbaru untuk disinkronisasikan dengan database SQLite.

#### 4.2 Implementasi Interface

Pada implementasi *interface* hanya diimplemenetasikan pada aplikasi *client* kamus *mobile*. Seperti yang dijelaskan sebelumnya aplikasi *server* dibuat hanya sebatas *service* saja. Adapun desain *interface* aplikasi kamus *mobile* terdiri dari lima bagian yaitu menu utama, menu kamus, menu *update*, menu bantuan,

dam menu tentang. Masing-masing menu akan dibahas lebih detail pada bahasan berikutnya.

#### 4.2.1 Menu Utama

Tampilan *interface* menu utama dapat dilihat dari *screenshot* atau tangkapan layar pada gambar 4.3 berikut.



Gambar 4.3 *Interface* menu utama

Pada menu ini terdapat empat pilihan sub menu yang memiliki fungsi masing-masing sesuai dengan menu yang dipilih.

#### 4.2.2 Menu Kamus

Tampilan interface menu “Kamus” dapat dilihat dari *screenshot* atau tangkapan layar pada gambar 4.4 berikut.



Gambar 4.4 *Interface* menu Kamus

*Interface* kamus ini dibuat untuk menangani proses utama aplikasi kamus *mobile* yaitu untuk menerjemahkan kata Jawa-Indonesia. Terdapat dua *textfield* yang atas sebagai input kata Jawa yang akan diterjemahkan dan yang bawah untuk menampilkan kata bahasa Indonesia hasil terjemahan. Diantara dua *textfield* terdapat satu tombol yang berfungsi untuk melakukan proses penerjemahan.

Kode program dari aplikasi kamus *mobile* yang berfungsi untuk menterjemahkan kata dari bahasa Jawa-Indonesia dapat dilihat pada gambar 4.5 berikut.

```

package com.kamusjawind;

import android.app.Activity;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;

public class KamusActivity extends Activity {

    private SQLiteDatabase db = null;
    private Cursor kamusCursor = null;
    private EditText txtJawa;
    private EditText txtIndo;
    private DataKamus datakamus = null;
    public static final String JAWA = "jawa";
    public static final String INDONESIA = "indonesia";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        datakamus = new DataKamus(this);
        db = datakamus.getWritableDatabase();

        setContentView(R.layout.kamus);
        txtJawa = (EditText) findViewById(R.id.txtJawa);
        txtIndo = (EditText) findViewById(R.id.txtIndo);
    }

    public void getTerjemahan(View view) {
        String result = "";
        String katajawa = txtJawa.getText().toString();
        kamusCursor = db.rawQuery(
            "SELECT ID, JAWA, INDONESIA " + "FROM kamusjwind where
JAWA='"
                + katajawa + "' ORDER BY JAWA", null);
        Log.d("Kamus activity ", katajawa);
        if (kamusCursor.moveToFirst()) {
            result = kamusCursor.getString(2);
            for (; !kamusCursor.isAfterLast(); kamusCursor.moveToNext()) {
                result = kamusCursor.getString(2);
            }
        }
        if (result.equals("")) {
            result = "Terjemahan tidak ditemukan";
        }
        txtIndo.setText(result);
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        db.close();
    }
}

```

Gambar 4.5 Kode program untuk fungsi penerjemahan

### 4.3.3 Menu Update

Tampilan interface menu “Update” dapat dilihat dari *screenshot* atau tangkapan layar pada gambar 4.6 berikut.



Gambar 4.6 *Interface* menu *Update*

Pada *interface* menu update ini terdapat dua tombol yang pertama berfungsi hanya untuk melihat data update atau terbaru yang ada di *server*, tombol yang kedua berfungsi untuk melakukan sinkronisasi database lokal SQLite dengan database *server* MySQL. Pengguna bisa langsung melakukan update tanpa harus melihat data *update* yang ada di *Server*.

Kode program pada kamus *mobile* yang menangani aksi ketika tombol “Update Data Kamus” ditekan dapat dilihat pada gambar 4.7 berikut.

```

public class AllUpdateActivity extends Activity {
    private ProgressDialog pDialog;
    // JSON Parser object
    JSONParser jParser = new JSONParser();
    private static String url_all_update = "http://192.168.150.1/kamusjwind/get_all_updates.php";
    private DataKamus datakamus = null;
    private AlertDialog.Builder builder;
    int success;
    String message;

    // JSON Node
    private static final String TAG_SUCCESS = "success";
    private static final String TAG_JAVAIND = "javaind";
    private static final String TAG_KATAJAWA = "katajawa";
    private static final String TAG_KATAINDONESIA = "kataindonesia";

    // JSONArray
    JSONArray javaind = null;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.update);
        datakamus = new DataKamus(this);
        builder = new AlertDialog.Builder(this);
        builder.setMessage("Selesai, data telah diupdate.")
            .setCancelable(false)
            .setPositiveButton("OK", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    finish();
                }
            });
        new AllUpdate().execute();
    }

    class AllUpdate extends AsyncTask<String, String, String> {
        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            pDialog = new ProgressDialog(AllUpdateActivity.this);
            pDialog.setMessage("Mohon tunggu...");
            pDialog.setIndeterminate(false);
            pDialog.setCancelable(false);
            pDialog.show();
        }

        protected String doInBackground(String... args) {
            int countRow = datakamus.getDataCount();
            Log.d("COUNT", String.valueOf(countRow));
            List<NameValuePair> params = new ArrayList<NameValuePair>();
            params.add(new BasicNameValuePair("uid", String.valueOf(countRow)));
            JSONObject json = jParser.makeHttpRequest(url_all_update, "GET",
                params);

            try {
                success = json.getInt(TAG_SUCCESS);
                Log.d("All Data Update: ", json.toString());
                if (success == 1) {
                    javaind = json.getJSONArray(TAG_JAVAIND);
                    if (javaind.length() != 0) {
                        for (int i = 0; i < javaind.length(); i++) {
                            JSONObject c = javaind.getJSONObject(i);
                            String katajawa = c.getString(TAG_KATAJAWA);
                            String kataindonesia = c.getString(TAG_KATAINDONESIA);
                            datakamus.addData(katajawa, kataindonesia);
                        }
                    }
                } else {
                    message="Tidak ada data update";
                }
            } else {
                message="Tidak ada data update";
            }
        } catch (JSONException e) {
            message="Server error";
        }
        return null;
    }

    protected void onPostExecute(String file_url) {
        pDialog.dismiss();
        runOnUiThread(new Runnable() {

            public void run() {
                if (success == 1) {
                    AlertDialog alert = builder.create();
                    alert.show();
                } else {
                    builder.setMessage(message);
                    AlertDialog alert = builder.create();
                    alert.show();
                }
            }
        });
    }
}
}

```

Gambar 4.7 Kode program aksi update data/ sinkronisasi

Pada kode program tersebut pengaksesan REST *Web Service* oleh *client* digunakan *method* GET untuk menyertakan parameter yang disertakan melalui URL [http://192.168.150.1/kamusjwind/get\\_all\\_updates.php](http://192.168.150.1/kamusjwind/get_all_updates.php). Parameter sendiri didapatkan dari hasil hitung/ *count* jumlah data/ kosakata yang ada pada database lokal SQLite, sebagai contoh apabila didapatkan dari hasil hitung/ *count* adalah 13 maka listing kode program diatas akan menghasilkan [http://192.168.150.1/kamusjwind/get\\_all\\_updates.php?uid=13](http://192.168.150.1/kamusjwind/get_all_updates.php?uid=13) sebagai *request* ke *server* dan *server* akan merespon dengan melakukan *query* ke database berdasarkan parameter “uid” yang dikirim *client* dan jika ada hasilnya akan dikembalikan dalam format JSON. Berikut ini pada gambar 4.8 adalah contoh respon atau kembalian dari *server* kepada *client* yang berupa JSON data.

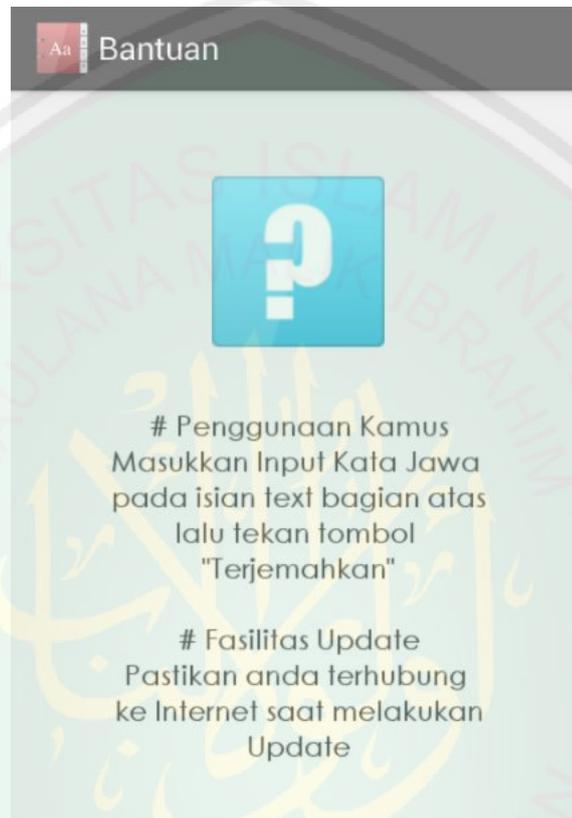
```
{
  "javaind": [
    { "uid": "14", "katajawa": "adhang", "kataindonesia": "menunggu di tempat yang akan dilewati" },
    { "uid": "15", "katajawa": "adhem", "kataindonesia": "dingin" },
    { "uid": "16", "katajawa": "adhep", "kataindonesia": "hadap" },
    { "uid": "17", "katajawa": "adhi", "kataindonesia": "adik" },
    { "uid": "18", "katajawa": "adoh", "kataindonesia": "jauh" },
    { "uid": "19", "katajawa": "adol", "kataindonesia": "menjual" },
    { "uid": "20", "katajawa": "adu", "kataindonesia": "adu" },
    { "uid": "21", "katajawa": "adus", "kataindonesia": "mandi" },
    { "uid": "22", "katajawa": "agama", "kataindonesia": "besar; agung" },
    { "uid": "23", "katajawa": "agul-agul", "kataindonesia": "andalan; jagoan" },
    { "uid": "24", "katajawa": "agung", "kataindonesia": "api" },
    { "uid": "25", "katajawa": "agem", "kataindonesia": "pakai" },
    { "uid": "26", "katajawa": "ageman", "kataindonesia": "pakaian" },
    { "uid": "27", "katajawa": "ageng", "kataindonesia": "besar" }
  ],
  "success": 1
}
```

Gambar 4.8 Respon data JSON dari *server*

Format data JSON tersebut itulah yang diterima *client* sebagai respon dari *server* yang selanjutnya diproses untuk sinkronisasi/ update database lokal SQLite.

#### 4.3.4 Menu Bantuan

Tampilan interface “Bantuan” update dapat dilihat dari *screenshot* atau tangkapan layar pada gambar 4.9 berikut.



Gambar 4.9 *Interface* menu Bantuan

Pada menu ini berisi tentang informasi tambahan berupa kontak penulis sebagai pengembang aplikasi yang bisa dihubungi oleh pengguna yang mengalami kesulitan atau ada yang perlu ditanyakan.

#### 4.3.5 Menu Tentang

Tampilan interface menu “Tentang” dapat dilihat dari *screenshot* atau tangkapan layar pada gambar 4.10 berikut.



Gambar 4.10 *Interface* menu Tentang

Interface atau tampilan menu “Tentang” ini hanya berisi tentang informasi tambahan dan pelengkap saja.

#### 4.3 Uji Coba Sistem

Tahap uji coba sistem ini terdiri dari dua bagian utama yaitu fungsi utama aplikasi yaitu kamus Jawa-Indonesia yaitu untuk menterjemahkan kata. Untuk fungsi utama yang kedua yaitu pengujian sistem update yaitu proses sinkronisasi database lokal SQLite yang ada pada *client* android dengan database yang ada

pada *server* yaitu MySQL dengan menggunakan teknologi http REST *Web Service*. Uji sinkronisasi database ini terdiri dari dua pengujian yaitu dalam hal kesamaan data dan juga waktu sinkronisasi.

#### 4.3.1 Uji Coba Aplikasi Kamus

Pada tahap uji coba aplikasi kamus ini dilakukan pengujian fungsi kamus yaitu untuk menerjemahkan kata Jawa-Indonesia. Berikut hasil uji coba yang didapatkan.



Gambar 4.11 Hasil terjemah kamus

Pada gambar 4.11 uji coba terjemah, pada *textfield* atas adalah input dari pengguna dan akan menampilkan hasil terjemah pada *textfiled* bawah ketika user menekan tombol “Terjemahkan”. Akan tetapi jika tidak ditemukan kata yang

dicari pada database SQLite maka akan pada *textfield* bawah akan ditampilkan “Terjemahan tidak ditemukan”.

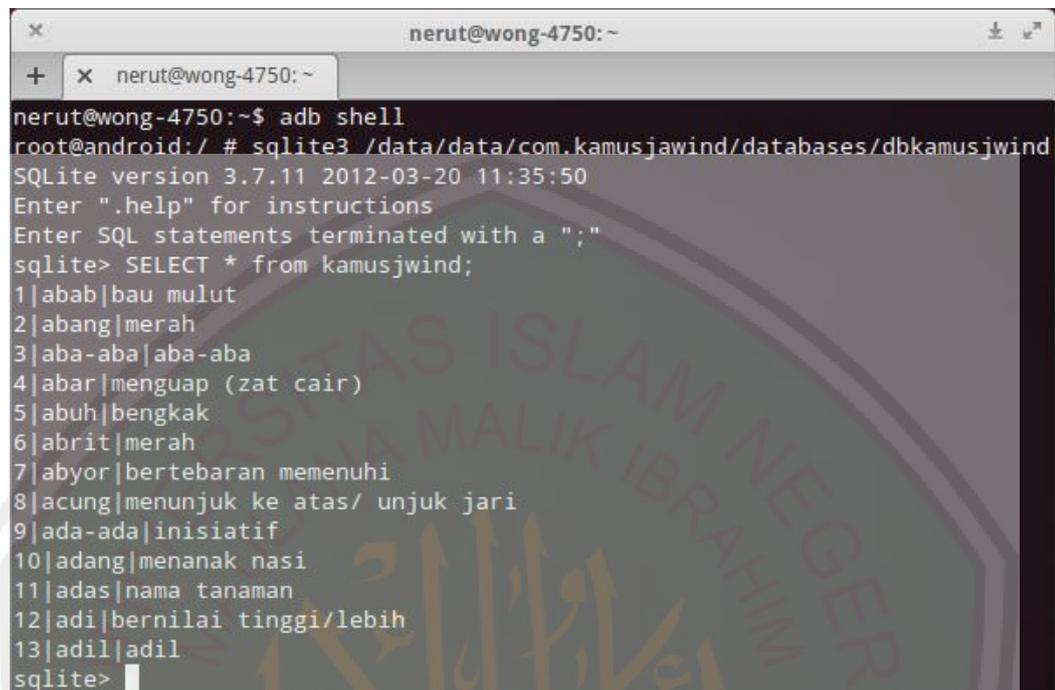
#### 4.3.2 Uji Coba Sinkronisasi Database

Seperti yang telah dijelaskan sebelumnya untuk uji coba sinkronisasi data ini terdiri dari dua pengujian, yang pertama pengujian kesamaan hasil sinkronisasi yang telah dilakukan dan untuk yang kedua pengujian dalam hal waktu yang dibutuhkan dalam proses sinkronisasi.

##### 4.3.2.1 Uji Coba Kesamaan Database

Pada uji coba kesamaan database ini menggunakan android virtual device (Genymotion) yang ada pada satu komputer dengan *server*. Alasan penggunaan *virtual device* adalah agar mudah dalam melihat struktur atau isi table yang ada pada database SQLite.

Pengujian dilakukan dengan cara, pertama penulis akan tampilkan isi tabel kamus pada database lokal SQLite sebelum di sinkronisikan dengan database *server*. Data awal pada database lokal SQLite berjumlah 13 kata Jawa beserta terjemahnya dalam bahasa Indonesia. Berikut isi dari tabel kamus Jawa-Indonesia pada database lokal SQLite.



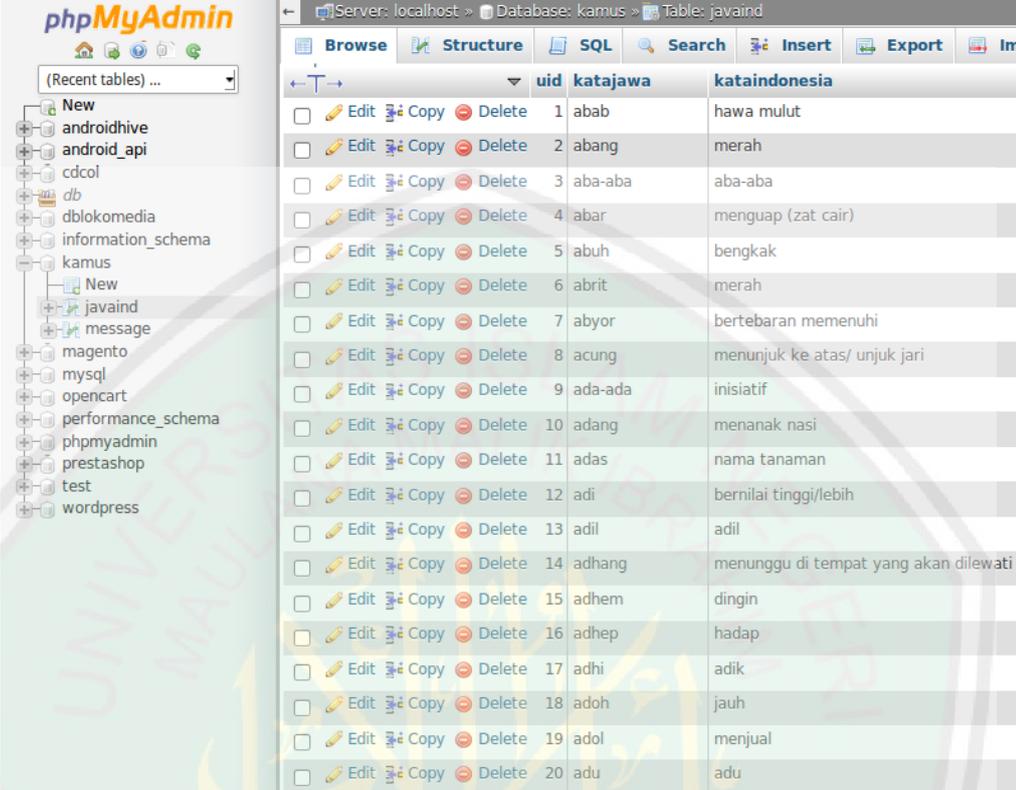
```

nerut@wong-4750: ~
+ x nerut@wong-4750: ~
nerut@wong-4750:~$ adb shell
root@android:/ # sqlite3 /data/data/com.kamusjawind/databases/dbkamusjwind
SQLite version 3.7.11 2012-03-20 11:35:50
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> SELECT * from kamusjwind;
1|abab|bau mulut
2|abang|merah
3|aba-aba|aba-aba
4|abar|menguap (zat cair)
5|abuh|bengkak
6|abrit|merah
7|abyor|bertebaran memenuhi
8|acung|menunjuk ke atas/ unjuk jari
9|ada-ada|inisiatif
10|adang|menanak nasi
11|adas|nama tanaman
12|adi|bernilai tinggi/lebih
13|adil|adil
sqlite>

```

Gambar 4.12 Isi tabel awal pada database lokal SQLite

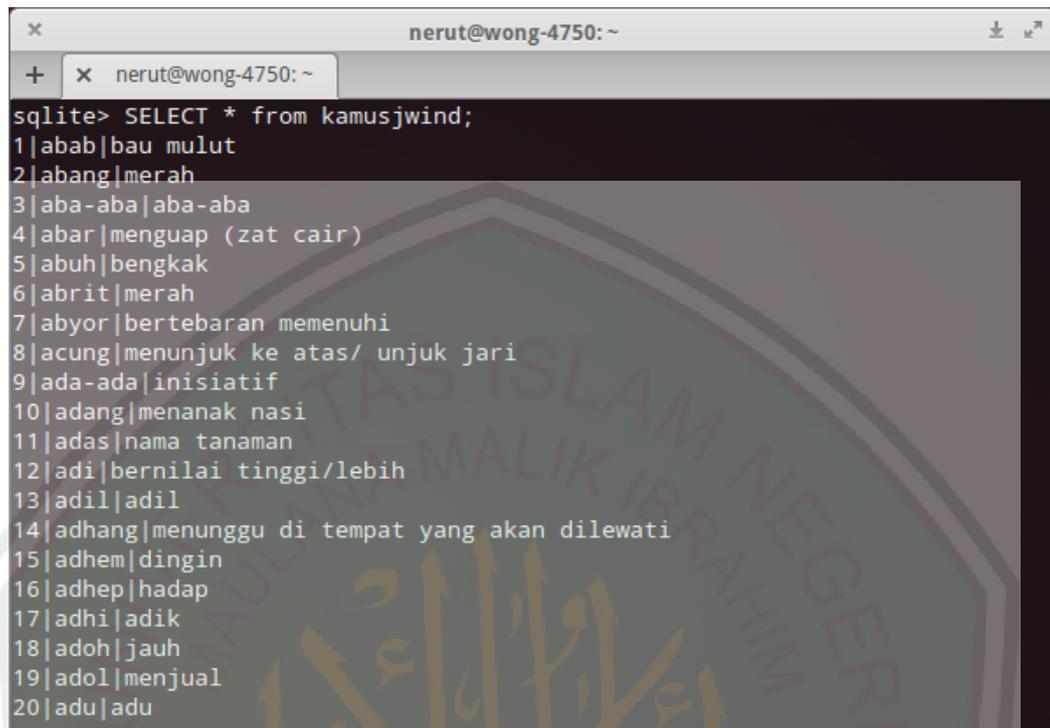
Langkah kedua penulis akan menampilkan isi tabel kamus pada database *server* MySQL yang akan disinkronisasikan dengan database lokal SQLite. Isi tabel kamus pada database *server* MySQL berjumlah 20 kata Jawa beserta terjemahannya (penambahan 7 kata yang tidak terdapat pada database lokal SQLite). Berikut isi tabel kamus pada database *server* MySQL.



	uid	katajawa	kataindonesia
<input type="checkbox"/>	1	abab	hawa mulut
<input type="checkbox"/>	2	abang	merah
<input type="checkbox"/>	3	aba-aba	aba-aba
<input type="checkbox"/>	4	abar	menguap (zat cair)
<input type="checkbox"/>	5	abuh	bengkak
<input type="checkbox"/>	6	abrit	merah
<input type="checkbox"/>	7	abyor	bertebaran memenuhi
<input type="checkbox"/>	8	acung	menunjuk ke atas/ unjuk jari
<input type="checkbox"/>	9	ada-ada	inisiatif
<input type="checkbox"/>	10	adang	menanak nasi
<input type="checkbox"/>	11	adas	nama tanaman
<input type="checkbox"/>	12	adi	bernilai tinggi/lebih
<input type="checkbox"/>	13	adil	adil
<input type="checkbox"/>	14	adhang	menunggu di tempat yang akan dilewati
<input type="checkbox"/>	15	adhem	dingin
<input type="checkbox"/>	16	adhep	hadap
<input type="checkbox"/>	17	adhi	adik
<input type="checkbox"/>	18	adoh	jauh
<input type="checkbox"/>	19	adol	menjual
<input type="checkbox"/>	20	adu	adu

Gambar 4.13 Isi tabel awal pada database *server* MySQL

Langkah yang terakhir atau yang ketiga adalah melakukan *update* data pada kamus agar database bisa tersinkronisasi. Setelah proses sinkronisasi akan dilihat lagi isi dari tabel kamus pada database lokal SQLite apakah isinya sama dengan isi tabel pada database *server* MySQL. Berikut ini adalah isi tabel setelah proses *update* database pada kamus *mobile*.



```

x nerut@wong-4750: ~
+ x nerut@wong-4750: ~
sqlite> SELECT * from kamusjwind;
1|abab|bau mulut
2|abang|merah
3|aba-aba|aba-aba
4|abar|menguap (zat cair)
5|abuh|bengkak
6|abrit|merah
7|abyor|bertebaran memenuhi
8|acung|menunjuk ke atas/ unjuk jari
9|ada-ada|inisiatif
10|adang|menanak nasi
11|adas|nama tanaman
12|adi|bernilai tinggi/lebih
13|adil|adil
14|adhang|menunggu di tempat yang akan dilewati
15|adhem|dingin
16|adhep|hadap
17|adhi|adik
18|adoh|jauh
19|adol|menjual
20|adu|adu

```

Gambar 4.14 Isi tabel pada SQLite setelah *update* / sinkronisasi data

Dari gambar 4.14 dapat kita lihat bahwa isi tabel pada database lokal SQLite setelah di *update* hasilnya sama persis dengan isi tabel pada database *server* MySQL. Dari sini dapat kita simpulkan bahwa *update* database pada kamus *mobile* untuk mensinkronkan dengan database *server* MySQL hasilnya sama.

#### 4.3.2.2 Uji Coba Waktu Sinkronisasi

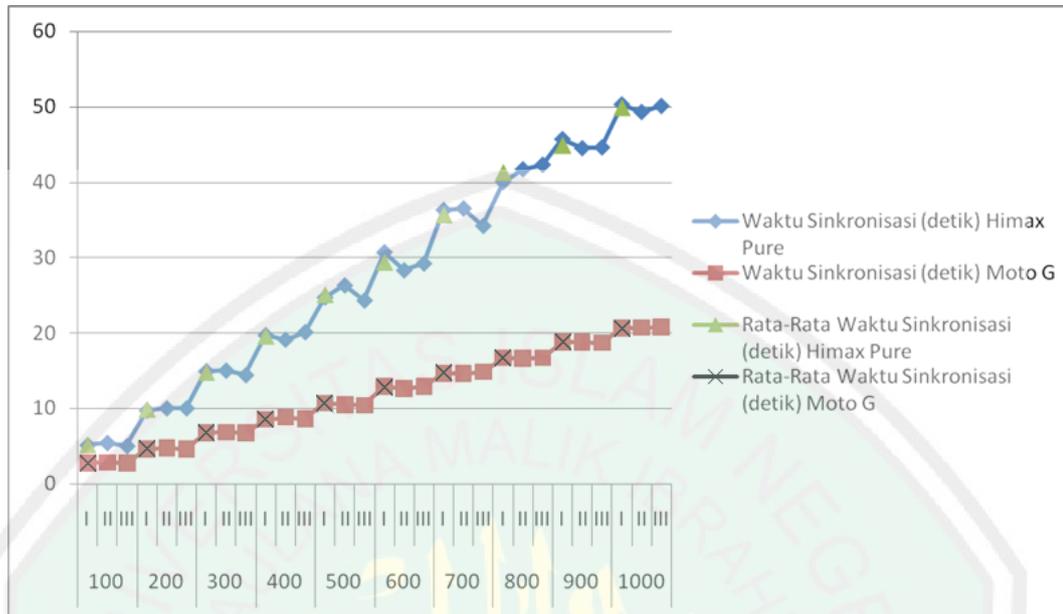
Pada uji coba waktu sinkronisasi ini langsung menggunakan *device* android yang sebenarnya, tidak lagi menggunakan android *virtual device*. Pada tahap uji coba ini untuk koneksi antara *server* dan *client* dilakukan simulasi melalui koneksi wifi antara komputer atau laptop yang bertindak sebagai *server* dengan device android sebagai *client*.

Langkah pengujian dilakukan dengan cara menghitung waktu selama proses *update* berlangsung. Pengujian dilakukan beberapa kali dengan dengan interval jumlah data yang berbeda pada setiap pengujian. Pengujian dilakukan menggunakan 2 *device* yang berbeda spesifikasi.

Adapun spesifikasi *device* android yang digunakan dalam uji coba kecepatan sinkronisasi update database adalah sebagai berikut:

- a. Himax Pure dengan spesifikasi utama *processor* Mediatek MTK6577 *dual core* 1.0 GHz dan RAM 512MB dengan sistem operasi Android 4.1.2 Jelly Bean
- b. Motorola Moto G dengan spesifikasi utama *processor* Qualcomm MSM8226 Snapdragon 400 *quad core* 1.2 GHz dan RAM 1GB dengan sistem operasi Android 4.4.2 Kitkat

Untuk data yang diujikan adalah jumlah kosakata beserta terjemahannya sebanyak 100 kosakata untuk pengujian pertama dan selanjutnya kelipatan 100 sampai 10 kali pengujian atau sampai 1000 kosakata. Dan setiap ujicoba 100 kosakata atau kelipatannya diulangi masing-masing sampai 3 kali uji coba untuk didapatnya rata-rata waktu sinkronisasinya. Adapun hasil uji coba dapat dilihat pada grafik yang terdapat pada gambar 4.14 berikut.



Gambar 4.15 Grafik Uji Coba Waktu Sinkronisasi

Untuk lebih jelasnya lagi hasil uji coba waktu sinkronisasi dapat dilihat pada tabel 4.1 berikut ini.

Tabel 4.1 Uji Coba Waktu Sinkronisasi

Jumlah Kosakata	Ukuran File JSON	Waktu Sinkronisasi (detik)			Rata-Rata Waktu Sinkronisasi (detik)	
		Uji Coba	Himax Pure	Moto G	Himax Pure	Moto G
100	6.2 KB	I	5.1	2.7	5.2	2.7
		II	5.4	2.8		
		III	5	2.7		
200	12.6 KB	I	9.7	4.6	9.9	4.6
		II	10	4.7		
		III	10	4.6		
300	19.1 KB	I	14.9	6.7	14.8	6.7
		II	15	6.8		
		III	14.4	6.7		
400	26.1 KB	I	19.7	8.5	19.6	8.6
		II	19.1	8.8		
		III	20.1	8.6		
500	33.1 KB	I	24.7	10.7	25.1	10.5
		II	26.3	10.5		
		III	24.3	10.4		
600	40.1 KB	I	30.7	12.9	29.4	12.8
		II	28.3	12.6		
		III	29.2	12.9		
700	46.7 KB	I	36.3	14.6	35.7	14.7
		II	36.5	14.6		
		III	34.2	14.8		
800	53.4 KB	I	40	16.7	41.3	16.7
		II	41.7	16.6		
		III	42.3	16.7		
900	60.2 KB	I	45.7	18.8	44.9	18.8
		II	44.5	18.8		
		III	44.6	18.7		
1000	66.5 KB	I	50.3	20.6	49.9	20.7
		II	49.3	20.7		
		III	50.1	20.8		

Dari hasil uji coba yang ditampilkan grafik hasil uji coba pada gambar 4.15 dan pada tabel 4.1 dapat dilihat bahwa dari hasil uji coba menunjukkan pada Himax Pure waktu yang lebih lama dibandingkan pada Motorola Moto G dan

pada Himax Pure pada 3 kali pengujian pada setiap data kelipan 100 sering terjadi selisih waktu yang cukup signifikan 1-2 detik berbeda dengan Moto G yang hampir konstan perbedaan pada 3 kali pengujian setiap data kelipatan 100 hanya selisih waktu dalam milidetik.

#### 4.4 Kajian Bahasa Jawa Dari Sisi Islam

Secara garis besar dalam Bahasa Jawa terdapat beberapa tingkatan mulai dari Ngoko yang digunakan untuk berbicara dengan orang yang lebih muda, Krama Madya yang digunakan untuk berbicara dengan teman sebaya dan Krama Inggil yang digunakan untuk berbicara kepada yang dihormati atau orang yang lebih tua atau orang yang dituakan termasuk di situ orang tua. Krama Inggil juga digunakan untuk berbicara kepada Tuhan dalam artian berdo'a.

Adanya tingkatan-tingkatan itu, menunjukkan karakter masyarakat Jawa yang berpegang teguh pada norma-norma kesopanan. Seorang pemuda akan dicap tidak mempunyai sopan santun apabila menggunakan Bahasa Ngoko saat berbicara kepada orang tuanya karena di rasa kasar dan tidak sesuai pada norma-norma yang berlaku dimasyarakat Jawa. Hal itu sangat sesuai ajaran agama Islam yang memerintah untuk bersikap saling menghormati dan berkata-kata yang baik khususnya pada orang tua. Jangankan membentak atau berkata-kata kasar kepada orang tua sekedar berkata "Ah" saja sudah dilarang. Seperti firman Allah dalam Alqur'an surat Al-Israa' ayat 23-24.

وَقَضَىٰ رَبُّكَ أَلَّا تَعْبُدُوا إِلَّا إِيَّاهُ وَبِالْوَالِدَيْنِ إِحْسَانًا ۗ إِمَّا يَبُلُغَنَّ عِنْدَكَ الْكِبَرَ

أَحَدُهُمَا أَوْ كِلَاهُمَا فَلَا تَقُلْ لَهُمَا أُفٍّ وَلَا تَنْهَرَهُمَا وَقُلْ لَهُمَا قَوْلًا كَرِيمًا ﴿٢٣﴾

وَأَخْفِضْ لَهُمَا جَنَاحَ الذُّلِّ مِنَ الرَّحْمَةِ وَقُلْ رَبِّ ارْحَمْهُمَا كَمَا رَبَّيَانِي صَغِيرًا ﴿٢٤﴾

*“Dan Tuhanmu telah memerintahkan supaya kamu jangan menyembah selain Dia dan hendaklah kamu berbuat baik pada ibu bapakmu dengan sebaik-baiknya. jika salah seorang di antara keduanya atau Kedua-duanya sampai berumur lanjut dalam pemeliharaanmu, Maka sekali-kali janganlah kamu mengatakan kepada keduanya Perkataan "ah" dan janganlah kamu membentak mereka dan ucapkanlah kepada mereka Perkataan yang mulia. Dan rendahkanlah dirimu terhadap mereka berdua dengan penuh kesayangan dan ucapkanlah: "Wahai Tuhanku, kasihilah mereka keduanya, sebagaimana mereka berdua telah mendidik aku waktu kecil".” (QS Al-Israa’ 23-24)*

Mengingat begitu banyaknya kosa kata dalam bahasa Jawa, juga cara penggunaannya yang beragam. Penulis mencoba menyederhanakannya dengan membuat sebuah Aplikasi kamus *mobile* Jawa-Indonesia. Yang bertujuan untuk memudahkan masyarakat untuk lebih memahami tentang bahasa Jawa agar tidak lupa akan Bahasa Jawa yang sangat kaya dengan nilai-nilai kesopanan yang selaras dengan ajaran agama Islam.

## BAB V

### PENUTUP

Pada bab ini dibahas mengenai kesimpulan yang dapat diambil dari tujuan pembuatan perangkat lunak, serta hasil uji coba yang telah dilakukan. Selain itu, terdapat beberapa saran untuk pengembangan lebih lanjut.

#### 5.1 KESIMPULAN

Dari hasil pengamatan selama perancangan, implementasi, dan uji coba perangkat lunak, dapat diambil kesimpulan sebagai berikut :

1. Aplikasi *server* database dapat dibangun menggunakan teknologi REST *web service*.
2. Aplikasi *client* kamus *mobile* dapat berkomunikasi dengan *server* database menggunakan REST *Web Service*.
3. Sinkronisasi dapat dilakukan dengan cara membandingkan jumlah data yang ada pada database *client* kamus *mobile* dengan database *server*, yaitu dengan cara menghitung jumlah data yang ada pada *client* untuk dijadikan parameter request data ke *server* dan jika data di *server* lebih banyak (ada penambahan) maka akan dilakukan proses sinkronisasi.

## 5.2 SARAN

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang, berdasarkan pada hasil perancangan, implementasi, dan uji coba yang telah dilakukan.

1. *Algoritma* translasi dapat di *improve* dengan algoritma lain yang lebih efisien dalam pencarian data agar program dapat berjalan lebih cepat.
2. Dapat diimplementasikan untuk bahasa daerah lain dan untuk *device* lain yang sekarang juga banyak dipakai seperti iPhone, Blackberry, dan Windows Phone.
3. *Algoritma* sinkronisasi dapat di *improve* dengan algoritma yang lain agar lebih baik dan lebih sempurna.
4. Pertakaran data dalam bentuk JSON dapat di *improve* lagi dengan menambahkan kompresi data agar lebih hemat *bandwidth* dan juga bias ditambahkan enkripsi untuk keamanan pertukaran data.

**DAFTAR PUSTAKA**

- Ariona, Rian. 2012. *Apa itu JSON?*, <http://www.ariona.net/apa-itu-json/> diakses 18 September 2012
- Fadlullah, Muhammad. 2012. *Rancang Bangun Aplikasi Kamus Jerman-Indonesia Berbasis Android*. Skripsi, Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Universitas STIKUBANK Semarang.
- Janah, Latif Nur. 2012. *Bahasa Indonesia Melumpuhkan Bahasa Daerah*, <http://bahasa.kompasiana.com/2012/09/07/bahasa-indonesia-melumpuhkan-bahasa-daerah/> diakses 20 September 2012
- Kahfi, Seto El. 2012. *Berkenalan Dengan JSON (JavaScript Object Notation) – Pengertian JSON*, <http://setoelkahfi.web.id/berkenalan-dengan-json-javascript-object-notation-pengertian-json/> diakses 18 September 2012
- Leginingsih, Iriani. 2006. *Pengembangan Aplikasi Kamus Bahasa Sunda pada Smartphone berbasis Sistem Operasi Symbian v.7.0*. Paper, Jurusan Teknik Informasi, Fakultas Teknik, Univeristas Komputer Indonesia.
- Liciu, Cristian. *Benchmarking AJAX (JSON vs. XML)*. [http://www.navioo.com/ajax/ajax\\_json\\_xml\\_Benchmarking.php](http://www.navioo.com/ajax/ajax_json_xml_Benchmarking.php) diakses 1 Oktober 2012
- Natsir, Fauzan. 2013. *Implementasi Web Service pada Aplikasi Kamus Bahasa Indonesia*. Paper, iKomuniTi, Vol.1 No.1 April 2013
- Noviyanto. 2009. *Sistem Terdistribusi – BAB 2 Komnikasi*. [http://viyan.staff.gunadarma.ac.id/Downloads/files/11848/2\\_Komunikasi.pdf](http://viyan.staff.gunadarma.ac.id/Downloads/files/11848/2_Komunikasi.pdf) diakses 17 September 2012

Putra, Candra Adi. 2011. *Arsitektur Android OS*.

<http://www.candra.web.id/2011/01/30/arsitektur-android-os/> diakses  
21 September 2012

Safaat H, Nazruddin. 2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Penerbit Informatika.

Sariono, Agus. 2002. *Bahasa dan Sastra Using: Ragam dan Alternatif Kajian*. Jakarta: Penerbit Tapal Kuda bekerja sama dengan Jurusan Sastra Indonesia, Fakultas Sastra UNEJ, Puslit Budaya Jawa dan Madura, Lemlit UNEJ.

Sembiring, Jhoni Pranata. 2013. *Perancangan Aplikasi Kamus Bahasa Indonesia-Karo Online berbasis Web dengan Metode Sequential Search*. Paper, Pelita Informatika Budi Dharma Volume IV, Nomor.2 Agustus 2013.

V. Mutiawani, I. Zamanhuri, Juwita. 2011. *Aplikasi Kamus Dwi Bahasa Aceh-Indonesia Berbasis Java Untuk Telepon Genggam*. Paper, Seminar Nasional Informatika 2011 (SNIF) Medan.

Wicaksono, Rizky Septiandy. 2011. *Sinkronisasi berbasis XML-RPC pada Studi Kasus Kamus Jawa-Indonesia*. Skripsi, Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh November Surabaya.

Zulaeha, Ella. 2011. *Uniknya Bahasa Jawa*,

<http://bahasa.kompasiana.com/2011/05/16/uniknya-bahasa-jawa/>  
diakses 19 September 2012

## LAMPIRAN

### Respon data JSON dari Server 100 kosakata

```
{
  "javaind": [
    { "uid": "14", "katajawa": "adhang", "kataindonesia": "menunggu di tempat yang akan dilewati" },
    { "uid": "15", "katajawa": "adhem", "kataindonesia": "dingin" },
    { "uid": "16", "katajawa": "adhep", "kataindonesia": "hadap" },
    { "uid": "17", "katajawa": "adhi", "kataindonesia": "adik" },
    { "uid": "18", "katajawa": "adoh", "kataindonesia": "jauh" },
    { "uid": "19", "katajawa": "adol", "kataindonesia": "menjual" },
    { "uid": "20", "katajawa": "adu", "kataindonesia": "adu" },
    { "uid": "21", "katajawa": "adus", "kataindonesia": "mandi" },
    { "uid": "22", "katajawa": "agama", "kataindonesia": "besar; agung" },
    { "uid": "23", "katajawa": "agul-agul", "kataindonesia": "andalan; jagoan" },
    { "uid": "24", "katajawa": "agung", "kataindonesia": "api" },
    { "uid": "25", "katajawa": "agem", "kataindonesia": "pakai" },
    { "uid": "26", "katajawa": "ageman", "kataindonesia": "pakaian" },
    { "uid": "27", "katajawa": "ageng", "kataindonesia": "besar" },
    { "uid": "28", "katajawa": "agni", "kataindonesia": "api" },
    { "uid": "29", "katajawa": "aja", "kataindonesia": "jangan" },
    { "uid": "30", "katajawa": "ajag", "kataindonesia": "anjing hutan" },
    { "uid": "31", "katajawa": "ajak", "kataindonesia": "ajak" },
    { "uid": "32", "katajawa": "ajang", "kataindonesia": "wadah" },
    { "uid": "33", "katajawa": "ajar", "kataindonesia": "ajar, belajar" },
    { "uid": "34", "katajawa": "ajeg", "kataindonesia": "tetap" },
    { "uid": "35", "katajawa": "ajeng", "kataindonesia": "akan" },
    { "uid": "36", "katajawa": "aji", "kataindonesia": "nilai; harga" },
    { "uid": "37", "katajawa": "ajur", "kataindonesia": "hancur" },
    { "uid": "38", "katajawa": "akas", "kataindonesia": "alon; perai\ / keras (untuk nasi)" },
    { "uid": "39", "katajawa": "ala", "kataindonesia": "buruk" },
    { "uid": "40", "katajawa": "alangan", "kataindonesia": "halangan" },
    { "uid": "41", "katajawa": "alas", "kataindonesia": "hutan" },
    { "uid": "42", "katajawa": "alem", "kataindonesia": "puji" },
    { "uid": "43", "katajawa": "aleman", "kataindonesia": "manja" },
    { "uid": "44", "katajawa": "alesan", "kataindonesia": "alasan" },
    { "uid": "45", "katajawa": "aling-aling", "kataindonesia": "bersembunyi di balik" },
    { "uid": "46", "katajawa": "alis", "kataindonesia": "alis" },
    { "uid": "47", "katajawa": "alok", "kataindonesia": "berkata" },
    { "uid": "48", "katajawa": "alu", "kataindonesia": "antan" },
    { "uid": "49", "katajawa": "alum", "kataindonesia": "layu" },
    { "uid": "50", "katajawa": "alun-alun", "kataindonesia": "lapangan di tengah kota" },
    { "uid": "51", "katajawa": "alus", "kataindonesia": "halus" },
    { "uid": "52", "katajawa": "aluwung", "kataindonesia": "lebih baik" },
    { "uid": "53", "katajawa": "ama", "kataindonesia": "hama" },
    { "uid": "54", "katajawa": "aman", "kataindonesia": "aman" },
    { "uid": "55", "katajawa": "amarga", "kataindonesia": "karena" },
    { "uid": "56", "katajawa": "amargi", "kataindonesia": "karena" },
    { "uid": "57", "katajawa": "amba", "kataindonesia": "lebar\ / luas" },
    { "uid": "58", "katajawa": "ambah", "kataindonesia": "jejak\ / jelajah\ / datang" },
    { "uid": "59", "katajawa": "ambal", "kataindonesia": "ulang" },
    { "uid": "60", "katajawa": "ambar", "kataindonesia": "tersebar (untuk bau" }
  ]
}
```

harum"}}, {"uid": "61", "katajawa": "ambeg", "kataindonesia": "berwatak"}, {"uid": "62", "katajawa": "amben", "kataindonesia": "balai-balai"}, {"uid": "63", "katajawa": "amber", "kataindonesia": "meluap (air)"}, {"uid": "64", "katajawa": "ambet", "kataindonesia": "bau"}, {"uid": "65", "katajawa": "ambblas", "kataindonesia": "lenyap seketika"}, {"uid": "66", "katajawa": "ables", "kataindonesia": "melesak"}, {"uid": "67", "katajawa": "ambrol", "kataindonesia": "runtuh"}, {"uid": "68", "katajawa": "ambruk", "kataindonesia": "tumbang\roboh"}, {"uid": "69", "katajawa": "ambu", "kataindonesia": "bau"}, {"uid": "70", "katajawa": "ambung", "kataindonesia": "cium"}, {"uid": "71", "katajawa": "ambus", "kataindonesia": "endus"}, {"uid": "72", "katajawa": "mbyar", "kataindonesia": "berserakan"}, {"uid": "73", "katajawa": "ambyuk", "kataindonesia": "menjatuhkan diri"}, {"uid": "74", "katajawa": "ambyur", "kataindonesia": "mencemplungkan diri ke dalam air"}, {"uid": "75", "katajawa": "amek", "kataindonesia": "mencari"}, {"uid": "76", "katajawa": "amem", "kataindonesia": "melempem; sunyi"}, {"uid": "77", "katajawa": "amis", "kataindonesia": "anyir"}, {"uid": "78", "katajawa": "ampas", "kataindonesia": "ampas"}, {"uid": "79", "katajawa": "ampe", "kataindonesia": "sulit bernafas; sesak (untuk dada)"}, {"uid": "80", "katajawa": "ampil", "kataindonesia": "pinjam"}, {"uid": "81", "katajawa": "ampak-ampak", "kataindonesia": "kepuluan debu"}, {"uid": "82", "katajawa": "amping-amping", "kataindonesia": "berlindung di balik sesuatu"}, {"uid": "83", "katajawa": "ampo", "kataindonesia": "nama jajanan terbuat dari tanah jenis tertentu"}, {"uid": "84", "katajawa": "amrih", "kataindonesia": "agar, supaya"}, {"uid": "85", "katajawa": "anak", "kataindonesia": "anak"}, {"uid": "86", "katajawa": "anda", "kataindonesia": "tangga"}, {"uid": "87", "katajawa": "andaka", "kataindonesia": "banteng"}, {"uid": "88", "katajawa": "andika", "kataindonesia": "anda"}, {"uid": "89", "katajawa": "ancang-ancang", "kataindonesia": "persiapan, mengambil kuda-kuda"}, {"uid": "90", "katajawa": "ancas", "kataindonesia": "tujuan"}, {"uid": "91", "katajawa": "ancur", "kataindonesia": "air raksa"}, {"uid": "92", "katajawa": "ancik", "kataindonesia": "menginjak"}, {"uid": "93", "katajawa": "ancer-ancer", "kataindonesia": "prakiraan; ancar-ancar"}, {"uid": "94", "katajawa": "andhap", "kataindonesia": "bagian bawah\rendah"}, {"uid": "95", "katajawa": "andhang", "kataindonesia": "tangga kayu berkaki empat"}, {"uid": "96", "katajawa": "andheng-andheng", "kataindonesia": "tahi lalat"}, {"uid": "97", "katajawa": "andhong", "kataindonesia": "sejenis kereta kuda"}, {"uid": "98", "katajawa": "andum", "kataindonesia": "berbagi"}, {"uid": "99", "katajawa": "angen-angen", "kataindonesia": "pemikiran\ingatan"}, {"uid": "100", "katajawa": "anget", "kataindonesia": "hangat"}, {"uid": "101", "katajawa": "angga", "kataindonesia": "tubuh"}, {"uid": "102", "katajawa": "anggak", "kataindonesia": "sombong"}, {"uid": "103", "katajawa": "anggara", "kataindonesia": "selasa"}, {"uid": "104", "katajawa": "anggarbini", "kataindonesia": "hamil"}, {"uid": "105", "katajawa": "anggep", "kataindonesia": "anggap"}, {"uid": "106", "katajawa": "angger", "kataindonesia": "asalkan"}, {"uid": "107", "katajawa": "angger-angger", "kataindonesia": "peraturan"}, {"uid": "108", "katajawa": "anggur", "kataindonesia": "anggur"}, {"uid": "109", "katajawa": "anggon", "kataindonesia": "tempat"}, {"uid": "110", "katajawa": "angin", "kataindonesia": "angin"}

```
sia": "angin"}, {"uid": "111", "katajawa": "angin-  
angin", "kataindonesia": "mencari  
udara"}, {"uid": "112", "katajawa": "angkah", "kataindonesia": "maksud"},  
{"uid": "113", "katajawa": "angkara", "kataindonesia": "angkara"}], "suc  
cess": 1}
```

