

SKRIPSI



SURYA WONO

NPM: 2011730093

PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2015



**UNDERGRADUATE THESIS**

**ISI SAMA SJUDUL SKRIPSI BAHASA INGGRIS**



**SURYA WONO**

**NPM: 2011730093**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2015**



# DAFTAR ISI

DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	viii
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	1
1.3 Tujuan . . . . .	1
1.4 Batasan Masalah . . . . .	2
1.5 Metodologi Penelitian . . . . .	2
1.6 Sistematika Pembahasan . . . . .	2
<b>2 DASAR TEORI</b>	<b>3</b>
2.1 <i>Mobile Cloud Computing</i> . . . . .	3
2.2 Android . . . . .	3
2.2.1 Pengertian Android . . . . .	3
2.2.2 Arsitektur Android . . . . .	3
2.2.3 <i>Life Cycle</i> . . . . .	3
2.3 Phonegap . . . . .	3
2.3.1 Pengertian Phonegap . . . . .	3
2.3.2 Arsitektur Phonegap . . . . .	5
2.3.3 HTML . . . . .	5
2.3.4 CSS . . . . .	5
2.3.5 Javascript . . . . .	5
2.4 Hadoop and <i>Ecosystem</i> . . . . .	5
2.4.1 Hadoop . . . . .	5
2.4.2 HDFS . . . . .	6
2.4.3 MapReduce . . . . .	6
2.4.4 HBase . . . . .	6
2.4.5 Trafodion . . . . .	6
2.5 Webservice and RESTful . . . . .	6
2.5.1 Webservice . . . . .	6
2.5.2 RESTful . . . . .	7
2.6 Google Open Authentication (OAuth) . . . . .	7
2.6.1 Open Authentication (OAuth) . . . . .	7
2.6.2 Google <i>OAuth</i> . . . . .	7
<b>3 ANALISIS</b>	<b>9</b>
3.1 Deskripsi Masalah . . . . .	9
3.2 <i>Cloud Computing Model</i> untuk kasus pembukuan . . . . .	9
3.3 Analisis Kebutuhan Perangkat Lunak . . . . .	9

3.3.1	Fitur Pada Aplikasi <i>Mobile Device</i> . . . . .	9
3.3.2	Fitur Pada Aplikasi <i>Website</i> . . . . .	9
3.3.3	Fitur Pada <i>Server</i> . . . . .	9
<b>4</b>	<b>INTRODUCTION</b>	<b>11</b>
4.1	Motivation . . . . .	11
	<b>DAFTAR REFERENSI</b>	<b>13</b>

## DAFTAR GAMBAR

2.1	Arsitektur Android . . . . .	4
2.2	Dasar <i>lifecycle</i> Android . . . . .	4
2.3	Arsitektur Phonegap . . . . .	5
2.4	Arsitektur Hadoop . . . . .	6
2.5	Arsitektur HDFS . . . . .	7
2.6	Google OAuth . . . . .	8

## DAFTAR TABEL



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Maraknya penggunaan perangkat *mobile* dan internet memberikan kemudahan dalam memanfaatkan aplikasi kolaborasi dan komunitas. Hal ini lebih dikuatkan lagi dengan adanya aplikasi jejaring sosial yang menyediakan fasilitas kolaborasi berbasis *cloud*. Fasilitas *cloud* merupakan pemanfaatan teknologi komputer dan internet. Dengan berbasis *cloud* pertumbuhan data yang sangat pesat dari berjuta pengguna tidak lagi menjadi masalah dan menimbulkan keawatiran dalam menggunakan perangkat *mobile* yang terbatas dalam kapasitas penyimpanan. Dengan demikian, memberikan ruang pertumbuhan yang tidak terbatas akan penggunaan aplikasi-aplikasi yang memberikan layanan kepada publik.

Pada saat penelitian ini dilakukan kehidupan rumah tangga moderen tidak lepas dari masalah keuangan. Kepada rumah tangga dan ibu rumah tangga tidak mengetahui pentingnya melakukan pengelolaan keuangan. Ketidaktahuan tersebut dikarenakan pengelolaan laporan keuangan yang mengharuskan pencatatan pengeluaran dan pendapatan dari masing-masing anggota keluarga. Pembuatan laporan keuangan juga tidak mudah sebab memakan waktu dan tenaga. Karena hal tersebut baik kepada rumah tangga maupun ibu rumah tangga cenderung tidak mengetahui keseimbangan pendapatan dan pengeluaran mereka.

Untuk meningkatkan pengelolaan keuangan rumah tangga sangat dimungkinkan mengembangkan aplikasi dan layanan pembukuan. Layanan pembukuan ini ditujukan kepada rumah tangga dengan struktur utuh multi keluarga. Sebuah keluarga secara utuh terdiri atas ayah (kepala rumah tangga), ibu (ibu rumah tangga) sebagai pengurus rumah tangga. Juga beranggotakan anak-anak, maupun sanak keluarga seperti: orang tua, saudara, dan lainnya. Masing-masing anggota dapat mencatat pengeluaran dan pendapatan masing-masing serta kepala anggota dapat langsung melihat laporan dari semua anggota keluarganya.

### 1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah:

1. Bagaimana merancang aplikasi pembukuan rumah tangga dengan sistem peran?
2. Bagaimana mengimplementasikan aplikasi pada poin nomor satu dengan *framework* Hadoop sehingga aplikasi dapat menyimpan dan mengolah data yang besar dan secara bersamaan?

### 1.3 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah:

1. Merancang aplikasi pembukuan rumah tangga dengan sistem peran.
2. Mengimplementasikan aplikasi pada poin nomor satu dengan *framework* Hadoop.

## 1.4 Batasan Masalah

## 1.5 Metodologi Penelitian

Skripsi ini bersifat deskriptif, yaitu mendeskripsikan data baik dari literatur dan hasil pengujian.

## 1.6 Sistematika Pembahasan

- BAB 1 Pendahuluan memuat latar belakang, rumusan masalah, tujuan, ruang lingkup kajian, metode dan teknik pengumpulan data, dan sistematika penulisan.
- BAB 2 Dasar Teori memuat teori-teori yang menunjang dalam pembuatan skripsi ini.
- BAB 3 Analisis memuat deskripsi masalah, model *cloud computing*, dan analisis perangkat lunak.
- BAB 4 Disain memuat disain antar muka, disain basis data, disain aplikasi, dan disain Webservice.
- BAB 5 Implementasi dan Pengujian memuat lingkungan implementasi, konfigurasi implementasi, implementasi basis data, implementasi aplikasi, implementasi pengujian fungsional, pengujian eksperimen, dan kesimpulan hasil pengujian.
- BAB 6 Kesimpulan dan Saran memuat kesimpulan dan saran yang berdasarkan hasil analisis implementasi dan pengujian.

## BAB 2

### DASAR TEORI

#### 2.1 *Mobile Cloud Computing*

#### 2.2 Android

Pada sub-bab ini akan dibahas mengenai pengertian dan arsitektur Android.

##### 2.2.1 Pengertian Android

Android merupakan sistem operasi buatan Google untuk *mobile device*. Saat penelitian ini dilakukan Sistem operasi Android menjadi yang paling populer dan sudah digunakan oleh lebih dari 100 juta pengguna di seluruh dunia. Kesuksesan Android tidak lepas dari basis *open-source* yang memungkinkan pembuatan varian kustom dari Android.

##### 2.2.2 Arsitektur Android

Secara umum arsitektur Android dibagi menjadi empat lapisan. Lapisan arsitektur Android dapat dilihat pada Gambar 2.1<sup>1</sup>. Berikut ini adalah penjelasan mengenai empat lapisan pada arsitektur Android.

1. *Applications* merupakan lapisan teratas yang berhubungan dengan pengguna.
2. *Applications Framework* merupakan lapisan yang digunakan oleh para pengembang aplikasi. Pada lapisan ini terdapat *framework* yang dapat digunakan orang para pengembang aplikasi.
3. *Libraries* merupakan kumpulan-kumpulan fungsi yang disediakan oleh Android.
4. *Linux Kernel* merupakan kumpulan-kumpulan fungsi yang berhubungan langsung dengan perangkat keras.

##### 2.2.3 *Life Cycle*

Aplikasi yang berjalan pada Android memiliki *lifecycle* sesuai dengan rancangan sistem operasi Android. *Lifecycle* aplikasi pada Android dapat dilihat pada gambar 2.2.

#### 2.3 Phonegap

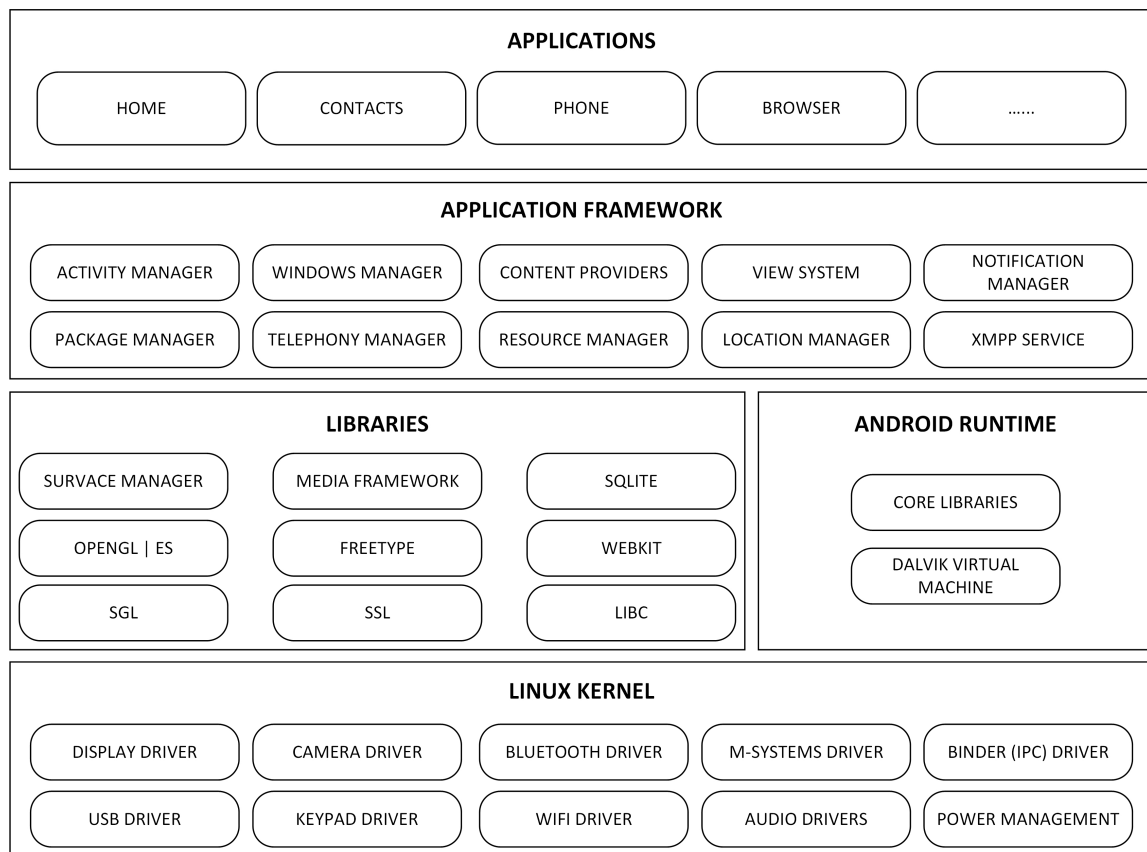
##### 2.3.1 Pengertian Phonegap

Phonegap merupakan suatu *framework* yang digunakan untuk mengembangkan aplikasi pada perangkat mobile. Phonegap memungkinkan aplikasi dibangun di atas Javascript, HTML5, dan CSS3<sup>2</sup>.

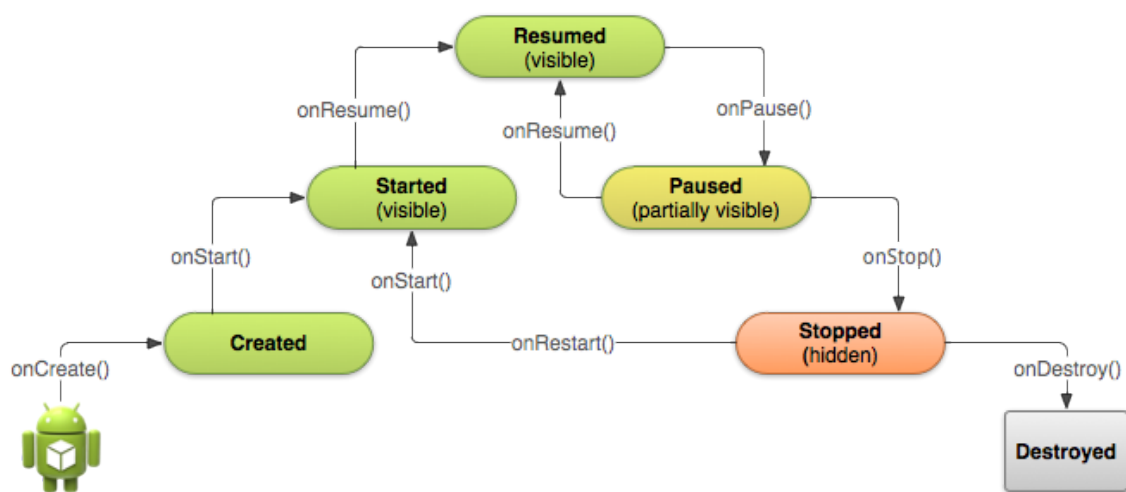
---

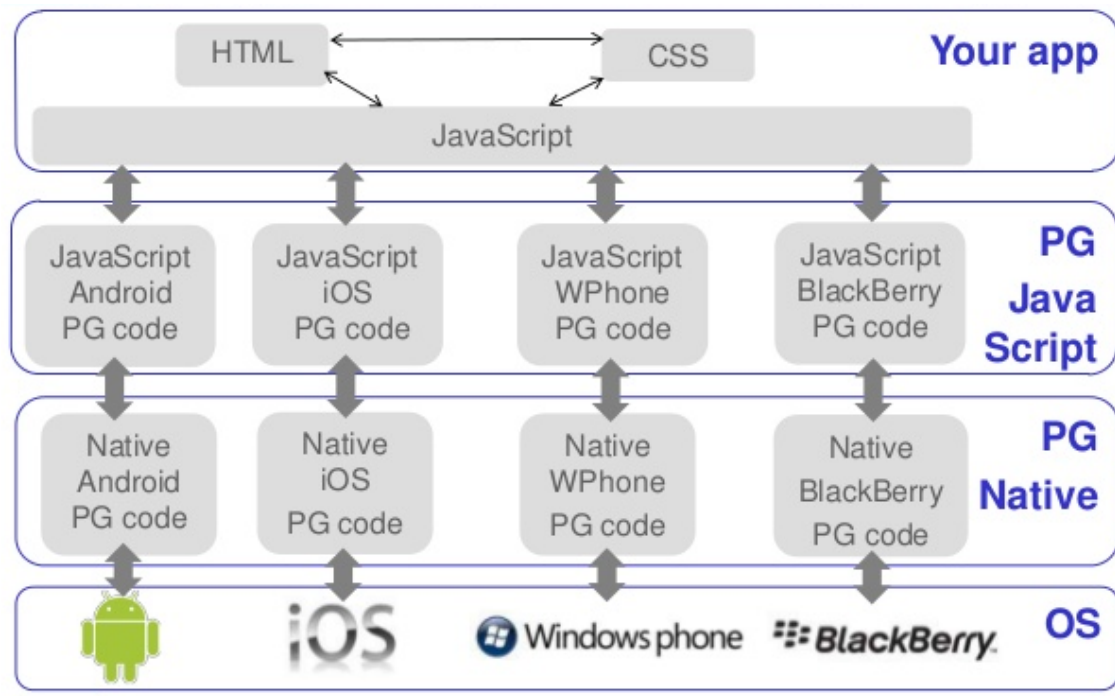
<sup>1</sup>[http://elinux.org/Android\\_Architecture](http://elinux.org/Android_Architecture)

<sup>2</sup>Jose Fermo (April 5, 2009). "PhoneGap Seeks to Bridge the Gap Between Mobile App Platforms"



Gambar 2.1: Arsitektur Android

Gambar 2.2: Dasar *lifecycle* Android



Gambar 2.3: Arsitektur Phonegap

### 2.3.2 Arsitektur Phonegap

Phonegap menggunakan HTML<sup>2.3.3</sup> dan CSS<sup>2.3.4</sup> untuk me-render aplikasi dan JavaScript<sup>2.3.5</sup> digunakan untuk menjalankan logika dari aplikasi yang dibuat. Phonegap membangun API yang dapat digunakan oleh pengembang aplikasi di atas OS *mobile device*. Arsitektur Phonegap dapat dilihat pada Gambar 2.3

### 2.3.3 HTML

HTML merupakan suatu bahasa standar yang digunakan untuk membuat halaman situs<sup>3</sup>.

### 2.3.4 CSS

CSS merupakan suatu bahasa yang digunakan untuk menformat tampilan suatu dokumen.

### 2.3.5 Javascript

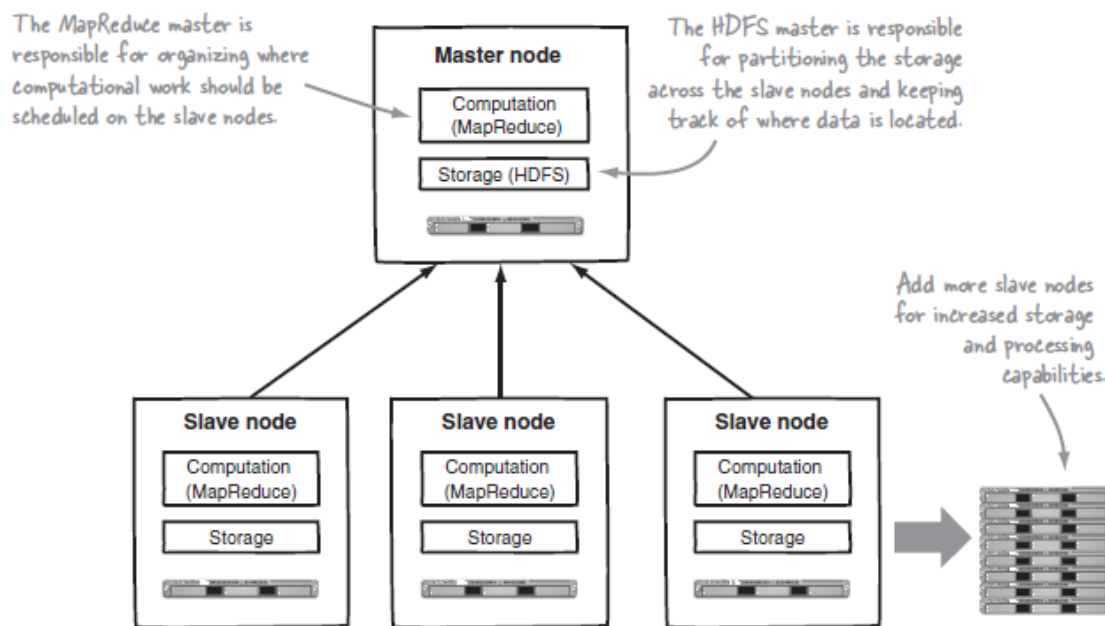
Javascript merupakan bahasa pemrograman yang pada umumnya digunakan pada *web browser*.

## 2.4 Hadoop and *Ecosystem*

### 2.4.1 Hadoop

Hadoop merupakan sebuah *platform* yang menyediakan penyimpanan data terdistribusi dan kemampuan komputasi. Kemampuan komputasi pada Hadoop merupakan *distributed master-slave architecture* yang terdiri dari Hadoop Distributed File System (HDFS)<sup>2.4.2</sup> untuk penyimpanan data dan MapReduce <sup>2.4.3</sup>. Arsitektur Hadoop dapat dilihat pada Gambar 2.4<sup>[1]</sup>.

<sup>3</sup>[http://www.merriam-webster.com/dictionary/hypertext markup language](http://www.merriam-webster.com/dictionary/hypertext%20markup%20language)



Gambar 2.4: Arsitektur Hadoop

#### 2.4.2 HDFS

HDFS adalah komponen penyimpanan data dari Hadoop yang merupakan sistem penyimpanan data terdistribusi. Arsitektur HDFS dapat dilihat pada Gambar 2.5

#### 2.4.3 MapReduce

MapReduce merupakan *batch-based*, komputasi terdistribusi *framework* yang memungkinkan komputasi paralel terhadap data yang cukup besar. MapReduce menyederhanakan pemrosesan paralel oleh abstraksi kerja yang kompleks. Dengan abstraksi ini, MapReduce memungkinkan *programmer* untuk berfokus pada kebutuhan bisnis dibandingkan memikirkan sistem distribusinya.

#### 2.4.4 HBase

HBase merupakan *real-time, column-oriented* basis data yang dapat diintegrasikan ke dalam HDFS melalui MapReduce.

#### 2.4.5 Trafodion

Trafodion merupakan *open source project* yang disponsori oleh HP. Trafodion juga diinkubasi di HP Labs dan HP-IT yang digunakan untuk mengembangkan SQL-on-Hadoop berskala *enterprise* terhadap data yang besar<sup>4</sup>.

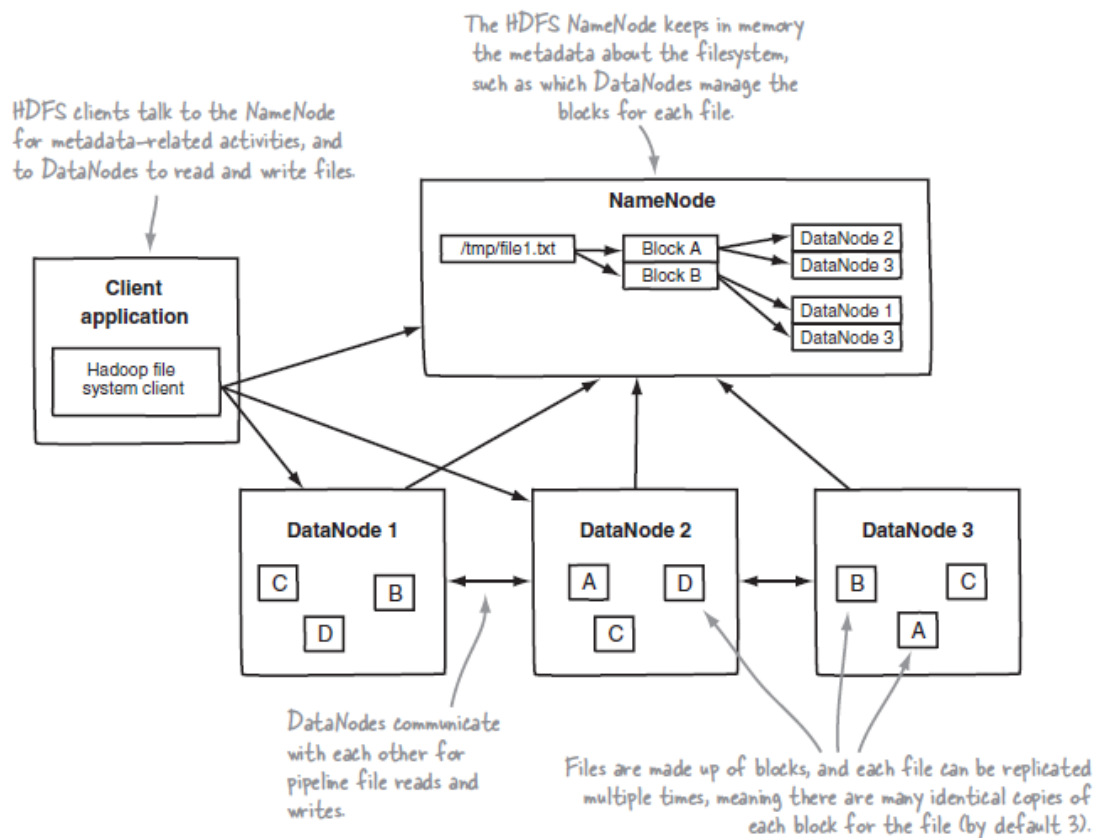
## 2.5 Webservice and RESTful

Pada sub-bab ini akan dibahas mengenai Webservice dan RESTful.

### 2.5.1 Webservice

Webservice merupakan suatu sistem yang menyediakan fungsi-fungsi dari suatu perangkat lunak di atas internet melalui *web*.

<sup>4</sup>[https://wiki.trafodion.org/wiki/index.php/Main\\_Page](https://wiki.trafodion.org/wiki/index.php/Main_Page)



Gambar 2.5: Arsitektur HDFS

## 2.5.2 RESTful

*Representational State Transfer* (REST) merupakan gaya arsitektur suatu perangkat lunak yang terdiri dari pedoman dan praktek terbaik untuk membuat suatu *webservice* yang *scalable*.

## 2.6 Google Open Authentication (OAuth)

Pada sub-bab ini akan dibahas mengenai OAuth dan Google Oauth.

### 2.6.1 Open Authentication (OAuth)

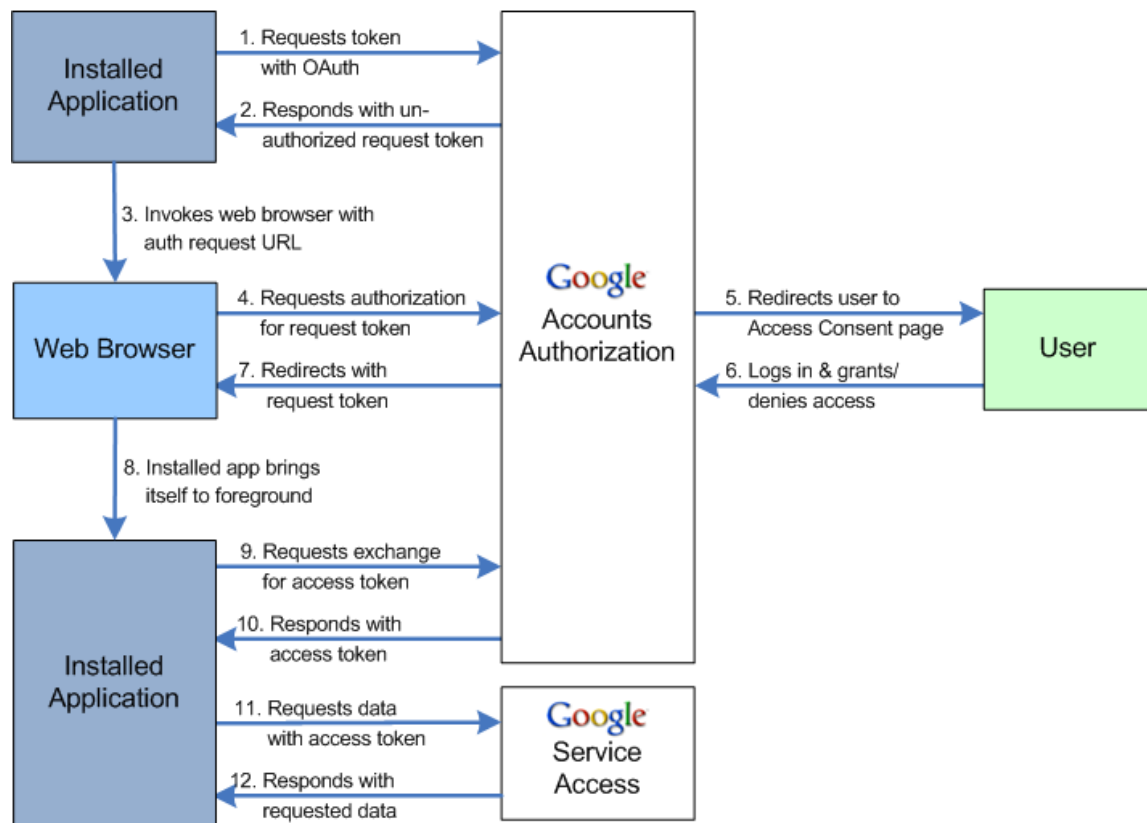
OAuth merupakan standar terbuka untuk autentikasi. OAuth menyediakan akses yang aman kepada klien untuk mengakses *server*. Hal ini menjadikan *server* dapat diakses oleh *third-party*. Desain OAuth diatas HTTP. Prinsip OAuth pada dasarnya menyediakan akses token kepada klien/pengguna akhir sehingga dapat digunakan untuk bertransaksi dengan *server*.

### 2.6.2 Google OAuth

Google OAuth merupakan protokol OAuth yang digunakan oleh google untuk memberikan akses kepada *third-party* untuk mengakses API mereka. Skema untuk mengakses Google OAuth dapat dilihat pada Gambar 2.6.

<sup>5</sup>Fielding, R. T.; Taylor, R. N. (2000). "Principled design of the modern Web architecture". pp. 407416. doi:10.1145/337180.337228

<sup>6</sup><http://tools.ietf.org/html/rfc6749>



Gambar 2.6: Google OAuth



## BAB 3

## ANALISIS

### 3.1 Deskripsi Masalah

Pada penelitian ini, akan dibuat suatu aplikasi yang dapat membantu suatu rumah tangga dalam pengelolaan keuangan mereka. Aplikasi ini dapat digunakan oleh setiap anggota rumah tangga untuk mencatat semua transaksi yang mereka lakukan baik pengeluaran maupun pendapatan. Aplikasi ini juga dapat menampilkan laporan sesuai dengan transaksi yang telah tercatat.

Aplikasi ini sendiri terbagi menjadi dua bagian yaitu aplikasi *end-user* yang digunakan langsung oleh para anggota rumah tangga dan aplikasi yang digunakan oleh admin untuk mengelolah data-data aplikasi.

Data-data yang tercatat tentunya akan disimpan kedalam sebuah basis data sehingga aplikasi ini sendiri akan berkomunikasi dengan *server* yang berfungsi sebagai penyimpanan dan pengolahan data yang dibangun diatas *framework* Hadoop. Untuk komunikasi aplikasi dan *server* akan menggunakan HTTP dimana aplikasi akan mengakses *webservice* yang telah disediakan oleh *server*.

### 3.2 *Cloud Computing Model* untuk kasus pembukuan

### 3.3 Analisis Kebutuhan Perangkat Lunak

Pada sub-bab ini akan dibahas fitur-fitur yang disediakan aplikasi dan *server*.

#### 3.3.1 Fitur Pada Aplikasi *Mobile Device*

#### 3.3.2 Fitur Pada Aplikasi *Website*

#### 3.3.3 Fitur Pada *Server*



1

## BAB 4

2

## INTRODUCTION

### 163 4.1 Motivation



## DAFTAR REFERENSI

- 165 [1] A. Holmes, *Hadoop in practice*. Manning Publications Co., 2012.