

# UNIVERSITAS GUNADARMA

FAKULTAS TEKNOLOGI INDUSTRI



## SKRIPSI

APLIKASI LAYANAN ADUAN MASYARAKAT KE PEMERINTAH KOTA  
DEPOK BERBASIS WEB DAN ANDROID

Nama : Agus Rianto  
NPM : 50413378  
Fakultas : Teknologi Industri  
Jurusan : Teknik Informatika  
Pembimbing : Dr. Pipit Dewi Arnesia, SKom., MMSI.

Diajukan Guna Melengkapi Sebagian Syarat  
Dalam Mencapai Gelar Sarjana Strata Satu (S1)

JAKARTA

2017

# Pernyataan Orisinalitas dan Publikasi

Saya yang bertanda tangan di bawah ini:

Nama : Agus Rianto  
NPM : 50413378  
NIRM : NIRM  
Judul Skripsi : APLIKASI LAYANAN ADUAN MASYARAKAT  
KE PEMERINTAH KOTA DEPOK BERBASIS  
WEB DAN ANDROID  
Tanggal Sidang : 13 Juni 2014  
Tanggal Lulus : 13 Juni 2014

Menyatakan bahwa tulisan di atas merupakan hasil karya saya sendiri dan dapat dipublikasikan sepenuhnya oleh Universitas Gunadarma. Segala kutipan dalam bentuk apapun telah mengikuti kaidah dan etika yang berlaku. Semua hak cipta dari logo serta produk yang disebut dalam buku ini adalah milik masing-masing pemegang haknya, kecuali disebutkan lain. Mengenai isi dan tulisan merupakan tanggung jawab Penulis, bukan Universitas Gunadarma.

Demikianlah pernyataan ini dibuat dengan sebenarnya dan dengan penuh kesadaran.

Jakarta, 13 Juni 2017

(Agus Rianto)

# Lembar Pengesahan

## Komisi Pembimbing

Tanggal Sidang: 13 Juni 2014

No.	Nama	Kedudukan
1.	Dr. Pipit Dewi Arnesia, SKom., MMSI.	Ketua
2.	nama pembimbing 1	Anggota
3.	nama pembimbing 2	Anggota

## Panitia Ujian

Tanggal Lulus: 13 Juni 2014

No.	Nama	Kedudukan
1.	Dr. Ravi A. Salim	Ketua
2.	Prof. Dr. Wahyudi Priyono	Sekretaris
3.	nama ketua pembimbing	Anggota
4.	nama pembimbing 1	Anggota
5.	nama pembimbing 2	Anggota

MENGETAHUI

**Pembimbing**

**Bagian Sidang Ujian**

(Dr. Pipit Dewi Arnesia, SKom., MMSI.)

(Dr. Edi Sukirman, MM)

# Abstraksi

Agus Rianto. 50413378

APLIKASI LAYANAN ADUAN MASYARAKAT KE PEMERINTAH KOTA DEPOK  
BERBASIS WEB DAN ANDROID .

Skripsi, Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas  
Gunadarma, 2017.

Kata Kunci : Laravel, Android, Layanan Masyarakat

(10+ 80+ lampiran)

Skripsi ini membahas tentang algoritma paralel pada Graphical Processing Unit (GPU) untuk penyelesaian masalah perhitungan ..... . Penulis mengangkat tema tersebut karena saat ini telah berkembang teknologi pemroses paralel (parallel processor) dalam sebuah kartu grafik (graphics card) dengan teknologi GPU yang sudah dimanfaatkan untuk pengolahan perhitungan grafik komputer serta dimungkinkan pula untuk perhitungan lainnya. Seperti diketahui kompleksitas algoritma perhitungan ..... telah diimplementasikan untuk mensimulasikan .... dengan waktu kompleksitas yang besar. Penelitian ini melakukan transformasi algoritma singular perhitungan ..... kedalam bentuk algoritma paralel yang diimplementasikan dalam sebuah teknologi GPU serta menganalisa tingkat efisiensi berdasarkan perbandingan waktu kompleksitas antara algoritma singular terhadap algoritma paralel, terhadap jumlah data yang digunakan.

Tahapan pengerjaan dalam tulisan ini dimulai dari analisis algoritma singular yang kemudian mencari bagian yang dapat ditransformasi kedalam bentuk pengolahan secara paralel. Implementasi algoritma menggunakan bahasa pemrograman C++ dengan menggunakan CUDA sebagai library untuk pemrograman paralel di perangkat GPU. Hasil percobaan menunjukkan rata-rata speed up sebesar ....., untuk jumlah data yang meningkat. Sehingga dapat disimpulkan bahwa algoritma paralel GPU dapat menurunkan waktu kompleksitas sebesar ..... untuk perhitungan ....

Daftar Pustaka (2000-2013)

# Kata Pengantar

Segala puji dan syukur penulis naikan ke hadirat Allah S.W.T yang Maha Kuasa yang telah memberikan berkat, anugerah dan karunia yang melimpah, sehingga penulis dapat menyelesaikan Tugas Akhir ini pada waktu yang telah ditentukan.

Tugas Akhir ini disusun guna melengkapi sebagian syarat untuk memperoleh gelar Sarjana Teknik Informatika Universitas Gunadarma. Adapun judul Tugas Akhir ini adalah "Implementasi Algoritma Paralel GPU pada perhitungan .....".

Walaupun banyak kesulitan yang penulis harus hadapi ketika menyusun Tugas Akhir ini, namun berkat bantuan dan dorongan dari berbagai pihak, akhirnya Tugas Akhir ini dapat diselesaikan dengan baik. Untuk itu penulis tidak lupa mengucapkan terima kasih kepada:

1. Ibu Prof. Dr. E.S. Margianti, SE, MM, selaku Rektor Universitas Gunadarma.
2. Bapak Prof. Dr. Ir. Bambang Suryawan, selaku Dekan Fakultas Teknologi Industri, Universitas Gunadarma.
3. Bapak Prof. Dr. -Ing. Adang Suhendra, S.Si., S.Kom., MSc., selaku Ketua Jurusan Teknik Informatika, Universitas Gunadarma.
4. Bapak Dr. Edi Sukirman, MM, selaku Bagian Sidang Ujian, Universitas Gunadarma.
5. Bapak Dr. Pipit Dewi Arnesia, SKom., MMSI., sebagai Pembimbing penulis yang ditengah- tengah aktifitas dan kesibukannya telah membimbing penulis dan memberikan dorongan sehingga penulisan ini dapat diselesaikan.
6. Kedua orang tua, kakak dan adik tercinta yang selalu mendukung dan terus memberikan motivasi.

7. Seluruh rekan seperjuangan di Universitas Gunadarma yang telah banyak membantu penulis.
8. Semua pihak yang tidak disebutkan yang telah membantu penyelesaian Tugas Akhir ini, penulis ucapkan juga terima kasih atas segala bantuan dan sarannya.

Sebagai manusia biasa yang tak luput dari kesalahan, maka penulis meminta maaf atas segala kekurangan dan keterbatasan dalam penyusunan Tugas Akhir ini.

Akhir kata, hanya kepada Tuhan jualah segalanya dikembalikan dan penulis sadari bahwa penulisan ini masih jauh dari sempurna, disebabkan karena berbagai keterbatasan yang penulis miliki. Untuk itu penulis mengharapkan kritik dan saran yang bersifat membangun untuk menjadi perbaikan di masa yang akan datang.

Semoga apa yang ada pada penulisan Tugas Akhir ini dapat bermanfaat bagi kita semua. Amin.

Jakarta, Juni 2017

Penulis

# Daftar Isi

<b>Halaman Judul</b>	<b>i</b>
<b>Lembar Orisinalitas</b>	<b>ii</b>
<b>Lembar Pengesahan</b>	<b>iii</b>
<b>Abstraksi</b>	<b>iv</b>
<b>Kata Pengantar</b>	<b>v</b>
<b>Daftar Isi</b>	<b>ix</b>
<b>Daftar Gambar</b>	<b>x</b>
<b>Daftar Tabel</b>	<b>xi</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang Masalah . . . . .	1
1.2 Batasan Masalah . . . . .	2
1.3 Tujuan . . . . .	2
1.4 Metode Penelitian . . . . .	3
1.5 Sistematika Penulisan . . . . .	3
<b>2 TINJAUAN PUSTAKA</b>	<b>4</b>
2.1 Smart City . . . . .	4
2.2 SDLC . . . . .	4
2.3 Android . . . . .	4
2.3.1 Versi Android . . . . .	4
2.3.2 Android Studio . . . . .	6
2.3.3 Android Software Development Kit . . . . .	6
2.3.4 Android Virtual Devices . . . . .	6

2.4	Java . . . . .	7
2.4.1	Java Development Kit . . . . .	7
2.4.2	Java Runtime Environment . . . . .	7
2.4.3	Java Virtual Machine . . . . .	7
2.4.4	Pemrograman Java pada Android . . . . .	8
2.4.5	Object Oriented Programming pada Java . . . . .	8
2.5	Extensible Markup Language . . . . .	9
2.6	Application Programming Interface . . . . .	9
2.7	JavaScript Object Notation . . . . .	10
2.8	PHP . . . . .	11
2.9	CodeIgniter . . . . .	11
2.10	MySQL . . . . .	11
2.11	HTML . . . . .	11
2.12	CSS . . . . .	11
2.13	UML . . . . .	11
<b>3</b>	<b>ANALISIS dan PERANCANGAN</b>	<b>13</b>
3.1	Perencanaan Aplikasi . . . . .	13
3.2	Analisis Kebutuhan . . . . .	13
3.3	Perancangan . . . . .	14
3.3.1	Use Case Diagram . . . . .	14
3.3.2	Activity Diagram . . . . .	15
3.3.3	Database . . . . .	16
3.3.4	Antarmuka . . . . .	16
3.3.4.1	Halaman Login . . . . .	17
3.3.4.2	Halaman Daftar . . . . .	17
3.3.4.3	Halaman Timeline Laporan . . . . .	18
3.3.4.4	Halaman Peta Laporan . . . . .	18
3.3.4.5	Halaman Activity Pengguna . . . . .	19
3.3.4.6	Halaman Profile Pengguna . . . . .	19
<b>4</b>	<b>HASIL DAN PEMBAHASAN</b>	<b>20</b>
4.1	Hasil . . . . .	20
4.2	Pembahasan . . . . .	20
<b>5</b>	<b>PENUTUP</b>	<b>21</b>
5.1	Kesimpulan . . . . .	21
5.2	Saran . . . . .	21



<b>DAFTAR PUSTAKA</b>	<b>22</b>
<b>LAMPIRAN</b>	<b>23</b>

# Daftar Gambar

3.1	Use Case Diagram . . . . .	14
3.2	Activity Diagram . . . . .	15
3.3	Rancangan Database . . . . .	16
3.4	Rancangan Halaman Login . . . . .	17
3.5	Rancangan Halaman Daftar . . . . .	17
3.6	Rancangan Timeline Laporan . . . . .	18
3.7	Rancangan Peta Laporan . . . . .	18
3.8	Rancangan Activity Pengguna . . . . .	19
3.9	Rancangan Halaman Login . . . . .	19

# Daftar Tabel

2.1	Tag XML pada Android . . . . .	9
-----	--------------------------------	---

# Bab 1

## PENDAHULUAN

### Latar Belakang Masalah

*Smart City* adalah konsep kota cerdas yang dirancang guna membantu berbagai hal kegiatan masyarakat, terutama dalam upaya mengelola sumber daya yang ada dengan efisien, serta memberikan kemudahan mengakses informasi kepada masyarakat, hingga untuk mengantisipasi kejadian yang tak terduga sebelumnya. Sebuah kota dikatakan *Smart* apabila kota tersebut benar-benar dapat mengetahui keadaan kota di dalamnya, memahami permasalahan tersebut secara lebih mendalam, hingga mampu melakukan aksi terhadap permasalahan tersebut. Pada saat ini, beberapa kota di Indonesia sudah menerapkan *Smart City* diantaranya Jakarta, Bandung, dan Surabaya.

Kota Depok merupakan salah satu kota di Indonesia yang memiliki visi menjadi kota yang Unggul, Nyaman dan Religius dan misi untuk meningkatkan kualitas pelayanan publik serta pembangunan infrastruktur dan ruang publik yang merata. Untuk dapat mewujudkan pembangunan dan perawatan infrastruktur yang merata maka diperlukan peran serta masyarakat untuk dapat melapor dan memberikan kritik serta saran ke pemerintah setempat. Pada saat ini Kota Depok belum memiliki sarana yang dapat digunakan oleh masyarakat untuk dapat menyampaikan aspirasinya. Sehingga untuk dapat mewujudkan visi dan misi pemerintah kota Depok diperlukan suatu wadah yang dapat digunakan untuk meningkatkan kualitas pelayanannya ke masyarakat Depok.

Aplikasi berbasis web dan android merupakan salah satu penerapan teknologi informasi yang bagus. Aplikasi berbasis web dan android memberikan kemudahan akses dan layanan data yang dapat dijangkau dimana saja dan kapan saja selama memiliki akses internet. CodeIgniter merupakan salah sa-

tu framework PHP yang sangat powerfull dan banyak digunakan saat ini. Dengan menggunakan framework pembuatan aplikasi web dan webservice menjadi lebih mudah dan terstruktur.

Android merupakan salah satu sistem operasi pada ponsel yang paling banyak digunakan saat ini. Ponsel berbasis android memiliki banyak fitur yang dapat dimanfaatkan diantaranya kamera, internet dan GPS. Kamera pada ponsel dapat dimanfaatkan untuk memberikan bukti pendukung, Internet dapat digunakan untuk mengirim informasi dimana saja dan kapan saja, dan GPS dapat digunakan untuk menentukan lokasi penggunanya. Dengan membuat aplikasi berbasis web dan android maka dapat memudahkan penggunanya dalam mengakses informasi secara langsung dari ponselnya.

Berdasarkan hal tersebut, maka penulis membuat penulisan yang berjudul **“Aplikasi Layanan Aduan Masyarakat ke Pemerintah Kota Depok Berbasis Web dan Android ”**

## Batasan Masalah

Berdasarkan uraian pada Latar Belakang Masalah, maka permasalahan yang akan dikaji adalah bagaimana membuat Aplikasi Sahabat Depok pada ponsel dengan sistem operasi Android serta membuat website untuk manajemen-nya. Pembuatan aplikasi android menggunakan *Integrated Development Environment (IDE)* Android Studio 2.3 dengan bahasa pemrograman JAVA dan XML. Sedangkan untuk pembuatan *Application Programming Interface (API)* dan websitenya menggunakan bahasa pemrograman PHP dengan framework CodeIgniter 3 dan database MySQL. Aplikasi ini menampilkan laporan yang dikirim oleh masyarakat dalam bentuk *timeline* dan juga *maps*. Pengguna dapat mengirimkan laporannya dengan mengirimkan deksripsi, gambar, dan lokasinya. Selain itu pengguna juga dapat melaporkan jika ada laporan yang tidak valid dan mengomentari laporan yang telah dikirim.

## Tujuan

Tujuan dari pembuatan aplikasi ini adalah untuk mempermudah masyarakat depok dalam melaporkan masalah ke pemerintah kota depok. Selain itu dengan dibuatnya aplikasi ini semoga pelayanan pemerintah kota depok dapat lebih cepat dalam mengatasi setiap keluhan masyarakat dan mewujudkan Depok *Smart City*.

## Metode Penelitian

Pembuatan aplikasi dilakukan menggunakan tahap-tahap *System Development Life Cycle* (SDLC) model *waterfall* dan terbagi menjadi empat tahap. Tahap pertama adalah analisis kebutuhan, yaitu menguraikan tentang layanan public menggunakan *smartphone*, kebutuhan user dalam menggunakan aplikasi, dan menguraikan perangkat lunak (*software*) maupun perangkat keras (*hardware*) dalam membangun aplikasi. Tahap kedua adalah proses perancangan, dimana mencakup perancangan antarmuka dan alur program menggunakan diagram UML. Tahap ketiga adalah tahap implementasi, pada tahap ini dilakukan pembangunan aplikasi menggunakan *Integrated Development Environment* (IDE) Android Studio dengan bahasa pemrograman JAVA dan XML. Tahap terakhir adalah tahap ujicoba aplikasi, dimana aplikasi akan diujicoba menggunakan *Android Virtual Device* (AVD) dan pengujian berdasarkan kuesioner pengguna.

## Sistematika Penulisan

Penulisan ini terdiri dari lima bagian yaitu pendahuluan, landasan teori, analisis, perancangan dan implementasi, hasil dan pembahasan, serta penutup. Pendahuluan berisi latar belakang pemilihan topik, ruang lingkup, tujuan yang ingin dicapai, metode penelitian dan sistematika penulisan. Landasan teori berisi uraian tentang *Smart City*, SDLC, Sistem Operasi Android, Android Studio, Java, XML, API, PHP, CodeIgniter, MySQL, JSON, dan diagram UML. Analisa, Perancangan dan implementasi berisi tentang perancangan antarmuka, perancangan alur program menggunakan diagram UML, dan pembangunan aplikasi. Hasil dan pembahasan berisi mengenai hasil pembuatan aplikasi. Penutup berisi kesimpulan dan saran.

# Bab 2

## TINJAUAN PUSTAKA

### Smart City

### SDLC

### Android

Android merupakan sistem operasi berbasis Linux yang dirancang untuk perangkat dengan layar sentuh seperti *smartphone* dan komputer tablet. Pada awalnya Android dikembangkan oleh Android, Inc., dan dibeli oleh Google pada tahun 2005. Android bersifat sumber terbuka (open-source), yaitu lisensi perizinan yang memungkinkan perangkat lunak tersebut dimodifikasi secara bebas.

### Versi Android

Versi Android diawali dengan dirilisnya Android beta pada bulan november 2007. Setelah itu pada april 2009 versi Android dikembangkan dengan nama kode yang dinamai berdasarkan makanan pencuci mulut dan panganan manis. Masing-masing versi dirilis sesuai urutan alfabet, yakni Cupcake (1.5), Donut (1.6), Éclair (2.0-2.1), Froyo (2.2-2.2.3), Gingerbread (2.3-2.3.7), Honeycomb (3.0-3.2.6), Ice Cream Sandwich (4.0-4.0.4), Jelly Bean (4.1-4.3), KitKat (4.4+), Lollipop (5.0+), Marshmallow (6.0), dan yang terbaru Nougat (7.0). Berikut ini adalah fitur-fitur pada Android ICS sampai Nougat.

1. Android Ice Cream Sandwich (4.0) Android Ice Cream Sandwich atau biasa disebut ICS dirilis pada bulan Oktober 2011. Android ICS tampil

dengan UI yang sangat modern dan lebih user friendly dibandingkan versi sebelumnya. Tampilan visual dengan nama “holo” ini dibuat lebih rapi, desain minimalis, dan font standar baru dengan nama “Roboto”. Selain itu Android ICS juga memperkenalkan web browser baru, manajemen kontak yang dapat di sinkronkan dengan jejaring sosial, kemampuan untuk mengakses kamera dan pemutar musik pada saat berada di layar yang terkunci (lockscreen), serta fitur face recognition unlocking untuk membuka layar yang terkunci.

2. Android Jelly Bean (4.1) Android Jelly Bean diluncurkan pada saat acara Google I/O Developer yang diselenggarakan pada 27 Juni 2012. Android Jelly Bean berfokus pada perbaikan antarmuka pada perangkat pengguna, dengan penambahan pada fitur Google Search Experience yaitu asisten digital dengan nama Google Now. Selain itu Android Jelly Bean juga mendukung multi user untuk tablet, widget di lockscreen, dan screen saver.
3. Android KitKat (4.4) Android KitKat dirilis secara resmi pada tanggal 3 September 2013 dengan code name “Key Lime Pie”. Fokus utama Android KitKat adalah mengoptimalkan performa sistem operasi pada perangkat kelas menengah dengan sumberdaya yang terbatas. Selain itu Android KitKat juga dirancang untuk menjadi lebih efisien dalam pengelolaan memori, sehingga aplikasi yang berjalan tidak menghabiskan terlalu banyak memori.
4. Android Lollipop (5.0) Android Lollipop dirilis secara resmi pada tanggal 28 Oktober 2014. Salah satu perubahan yang paling menonjol pada versi ini adalah desain tampilan antarmuka pengguna baru dengan konsep material design. Selain itu terdapat peningkatan juga pada notifikasi yang dapat diakses pada saat layar terkunci. Google juga merubah platform internalnya dengan Android Runtime yang secara resmi menggantikan dalvik untuk peningkatan performa aplikasi, dan optimasi dalam penggunaan baterai.
5. Android Marshmallow (6.0) Android Marshmallow diperkenalkan pada saat acara Google I/O yang diselenggarakan pada bulan mei 2015 dengan codename “Android M” dan dirilis secara resmi pada tanggal 5 Oktober 2015. Fokus utama pada Android Marshmallow adalah peningkatan antarmuka dari versi Android Lollipop. Selain itu pada versi



ini juga diperkenalkannya model permissions architecture baru, manajemen sistem pada sumber daya yang dapat mengurangi pemakaian energi pada aplikasi yang berjalan dibalik layar, dan secara alami mendukung fingerprint recognition.

6. Android Nougat (7.0) Android Nougat merupakan versi selanjutnya pada sistem operasi Android dan saat ini baru terdapat versi developer yang dirilis pada tanggal 9 maret 2016. Pada versi ini terdapat split screen multi window mode, yang membuat user dapat membuka beberapa aplikasi dalam satu layar.

## **Android Studio**

Android Studio merupakan *Integrated Development Environment (IDE)* resmi untuk pengembangan aplikasi Android. Android Studio diumumkan pada 16 Mei 2013 pada saat acara Google I/O dan tersedia secara gratis dibawah lisensi Apache. Android Studio menyediakan semua hal yang dibutuhkan untuk membuat aplikasi Android diantaranya, IntelliJ IDE+ Android Studio plugin, Android SDK Tools, Android Platform-tools, dan Android Emulator.

## **Android Software Development Kit**

*Android Software Development Kit (SDK)* adalah tools yang diperlukan untuk mengembangkan aplikasi pada platform Android. Android SDK terdiri dari debugger, libraries, handset emulator, dokumentasi, contoh kode dan tutorial. Android SDK berfungsi sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java.

## **Android Virtual Devices**

*Android Virtual Devices (AVD)* merupakan emulator perangkat Android pada komputer. AVD dapat digunakan untuk menjalankan aplikasi yang sudah dibuat kedalam berbagai jenis perangkat sesuai model yang di pilih. Dengan menggunakan *AVD programmer* dapat mengembangkan dan mencoba aplikasi Android tanpa harus menggunakan perangkat Android sebenarnya.

## Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk ponsel. Java awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems, akan tetapi saat ini merupakan bagian dari Oracle. Bahasa Java banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan runtime atas bawah yang minimal. Aplikasi-aplikasi berbasis Java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai *Java Virtual Machine* (JVM). Java merupakan bahasa pemrograman yang bersifat umum atau non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi Java mampu berjalan di beberapa platform sistem operasi yang berbeda.

### Java Development Kit

Java Development Kit (JDK) merupakan perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode Java ke bytecode yang dapat dimengerti dan dapat dijalankan oleh Java Runtime Environment (JRE). JDK berisi sekumpulan command line tool untuk menciptakan program Java. JDK wajib terinstal pada komputer yang akan melakukan proses pembuatan aplikasi berbasis Java, namun tidak wajib terinstal diperangkat yang akan menjalankan aplikasi yang dibangun dengan Java.

### Java Runtime Environment

Java Development Kit (JDK) merupakan perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode Java ke bytecode yang dapat dimengerti dan dapat dijalankan oleh Java Runtime Environment (JRE). JDK berisi sekumpulan command line tool untuk menciptakan program Java. JDK wajib terinstal pada komputer yang akan melakukan proses pembuatan aplikasi berbasis Java, namun tidak wajib terinstal diperangkat yang akan menjalankan aplikasi yang dibangun dengan Java.

### Java Virtual Machine

Java Runtime Environment (JRE) adalah perangkat lunak yang digunakan untuk menjalankan aplikasi yang dibuat dengan menggunakan bahasa pe-

mrograman Java. Versi JRE harus sama atau lebih tinggi dengan JDK yang digunakan untuk membuat aplikasi Java agar aplikasi tersebut dapat berjalan dengan baik.

## **Pemrograman Java pada Android**

Java Virtual Machine (JVM) adalah inti dari Bahasa Java. JVM berfungsi untuk mengubah bahasa program yang kawan buat menjadi bahasa mesin, Karena bahasa Java adalah bahasa pemrograman tingkat tinggi atau mendekati bahasa manusia maka dibutuhkan pengkonversi ke bahasa mesin. JVM mampu menerjemahkan program yang kawan buat ke hampir semua platform. “write once, run everywhere”.

1. Package, merupakan perintah Java yang digunakan untuk memberitahukan bahwa suatu class adalah anggota dari package, sedangkan nama package dapat berupa susunan direktori tempat dimana file class disimpan atau nama folder. Penulisan nama package dalam Java biasanya diawali dengan huruf kecil. Bentuk umum : Package <namapackage>.<subpackage>.<subsubpackage>;

## **Object Oriented Programming pada Java**

Konsep Object Oriented Programming (OOP) atau pemrograman berbasis objek pada Java terdiri dari 3 yaitu :

1. Inheritance, merupakan pewarisan yang artinya sebuah class dapat mewarisi atribut dan method dari class lain. untuk mendeklarasikan suatu class sebagai subclass dilakukan dengan cara menambahkan kata kunci extends setelah deklarasi nama class, kemudian diikuti dengan nama parent class-nya. Kata kunci extends tersebut memberitahu kompiler Java bahwa kita ingin melakukan perluasan class.
2. Polymorphism, Suatu aksi yang memungkinkan pemrogram menyampaikan pesan tertentu keluar dari hirarki obyeknya, dimana obyek yang berbeda memberikan tanggapan atau respon terhadap pesan yang sama sesuai dengan sifat masing-masing obyek.
3. Encapsulation adalah pembungkus, pembungkus disini dimaksudkan untuk menjaga suatu proses program agar tidak dapat diakses secara

Tabel 2.1: Tag XML pada Android

No.	Bentuk Tag	Fungsi
1.	<TextView>	Membuat teks statis
2.	<EditText>	Membuat kolom input berupa teks
3.	<ImageView>	Menampilkan gambar
4.	<Button>	Membuat tombol
5.	<LinearLayout>	Membuat tata letak agar lurus secara horizontal / vertical
6.	<RelativeLayout>	Membuat tata letak yang dapat diletakan secara bebas
7.	<ScrollView>	Membuat scroll ketika isinya melebihi kapasitas layar

sembarangan atau di intervensi oleh program lain. Dalam OOP Encapsulation di wujudkan dalam bentuk class. Seperti yang telah dijelaskan sebelumnya dalam sebuah class terdapat property dan method yang memiliki hak akses tertentu terhadap environment atau lingkungannya, hak akses ini biasa disebut Access Modifier, yang terdiri dari private, protected, dan public.

## Extensible Markup Language

Extensible Markup Language (XML) adalah suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur. Secara teknis XML didefinisikan sebagai suatu bahasa meta-markup yang menyediakan format tertentu untuk dokumen-dokumen yang mempunyai data terstruktur. XML bukanlah suatu bahasa pemrograman, melainkan sintaks yang digunakan untuk menjelaskan bahasa markup lain sehingga dinamakan meta language.

XML untuk Android pada dasarnya berbeda dengan XML secara umum. Pada XML untuk Android, beberapa nama tag telah didefinisikan oleh Android SDK, sedangkan XML pada umumnya penulisan nama tag dapat dilakukan sesuai keinginan. Berikut ini adalah beberapa tag XML pada Android yang sering digunakan.

## Application Programming Interface

Application Programming Interface (API) adalah sebuah bahasa dan format pesan yang digunakan oleh program aplikasi untuk berkomunikasi dengan system operasi atau program pengendalian lainnya seperti Database Management System (DBMS) atau komunikasi protokol. API diimplementasikan

dengan menulis fungsi panggilan atau sintaks dalam program, yang menyediakan sarana yang diperlukan untuk meminta layanan program. Pada dasarnya, program API mendefinisikan cara yang tepat bagi pengembang untuk meminta layanan dari program itu. Sebuah API dapat digunakan untuk memungkinkan aplikasi lain atau pihak ketiga dan para pengembang untuk memanfaatkan fasilitas-fasilitas pada API tersebut. Beberapa API yang banyak digunakan oleh pengembang diantaranya adalah Google API, Twitter API, dan Facebook API.

## JavaScript Object Notation

JavaScript Object Notation (JSON) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. Walaupun JSON didasarkan pada subset bahasa pemrograman JavaScript, JSON merupakan format teks yang tidak bergantung pada suatu bahasa. Kode untuk pengolahan dan pembuatan data JSON telah tersedia untuk banyak jenis bahasa pemrograman. Sehingga memudahkan para pengembang untuk melakukan pertukaran data pada aplikasinya.

JSON menggunakan bentuk atau struktur sebagai berikut :

1. Object adalah sepasang nama atau nilai yang tidak terurutkan. Object dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama atau nilai dipisahkan oleh , (koma).
2. Array adalah kumpulan nilai yang terurutkan. Array dimulai dengan [ (kurung kotak buka) dan diakhiri dengan ] (kurung kotak tutup), setiap nilai dipisahkan oleh , (koma).
3. Value dapat berupa sebuah string dalam tanda kutip ganda, angka, boolean, null, atau sebuah object maupun array.
4. String adalah kumpulan dari nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan backslash escapes "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string.

5. Angka pada JSON sangat mirip dengan angka di C atau Java, tetapi untuk format oktal dan heksadesimal tidak digunakan.

Contoh data dalam format JSON :

```
{
  "status": true ,
  "message": "Loggin success",
  "data": {
    "id": "3",
    "email": "agus@gmail.com",
    "name": "Agus Rianto",
    "avatar": "avatar_agus.jpg",
    "role": "2",
    "token": "043e5fda03e0a1000f6cb279bf7e034"
  }
}
```

## PHP

## CodeIgniter

## MySQL

## HTML

## CSS

## UML

Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML merupakan sebuah bahasa yang telah menjadi standar dalam industri visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak. Seperti bahasa pada umumnya, UML juga mendefinisikan notasi dan semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggabarkan berbagai diagram, dan memiliki setiap bentuk memiliki makna tertentu.

Sedangkan semantik mendefinisikan bagaimana bentuk tersebut dapat dikombinasikan.

# **Bab 3**

## **ANALISIS dan PERANCANGAN**

### **Perencanaan Aplikasi**

Pada penulisan ini, penulis merencanakan membuat aplikasi pelaporan untuk menyampaikan aspirasi masyarakat kota depok ke pemerintah setempat. Sebelum memulai pembuatan hal pertama yang harus dilakukan adalah menentukan platform dan bahasa pemrograman yang digunakan. Penulis memilih platform Android dengan bahasa pemrograman Java, dan websitenya menggunakan bahasa pemrograman PHP. Penulis memilih Android karena pada saat ini Android merupakan sistem operasi pada ponsel dengan jumlah pengguna terbanyak. Adapun fitur-fitur yang ingin dibuat pada aplikasi Android yaitu pembuatan laporan, fitur komentar pada laporan yang dibuat pengguna, tampilan laporan berdasarkan timeline, tampilan laporan pada peta, dan fitur melaporankan postingan bermasalah.

### **Analisis Kebutuhan**

Dalam pembuatan aplikasi dibutuhkan teks editor Sublime untuk membuat API dan Website dengan menggunakan bahasa pemrograman PHP dan Android Studio sebagai Integrated Development Environment (IDE) untuk membuat aplikasi pada perangkat Android dengan menggunakan bahasa pemrograman JAVA. Perangkat yang dibutuhkan untuk menjalankan Android Studio diantaranya :

- Sistem operasi Windows / Linux / Mac OS X
- Memory RAM 2GB atau 8GB untuk performa maksimal
- Ruang kosong untuk penyimpanan 4GB



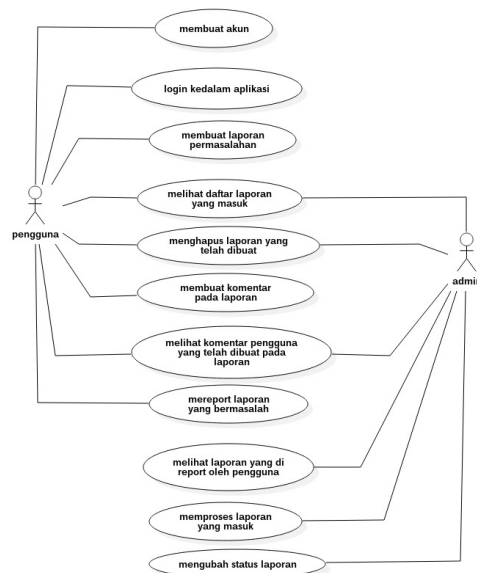
- Minimal resolusi layar 1280 x 800
- Java Development Kit (JDK) 8

## Perancangan

Pada perancangan aplikasi dibuat menggunakan diagram Unified Modelling Language (UML). Jenis diagram yang akan digunakan untuk merancang aplikasi ini yaitu, Use Case Diagram, Activity Diagram, dan Class Diagram. Selain dengan UML juga dilakukan perancangan database dan antarmuka penggunaanya.

### Use Case Diagram

Use Case Diagram pada aplikasi ini menggambarkan interaksi yang dapat dilakukan oleh pengguna dengan aplikasi. Diagram Use Case ini mendefinisikan apa yang dilakukan oleh sistem dan elemennya, bukan bagaimana sistem dan elemennya saling berinteraksi. Rancangan Use Case Diagram dapat dilihat pada gambar 3.1.



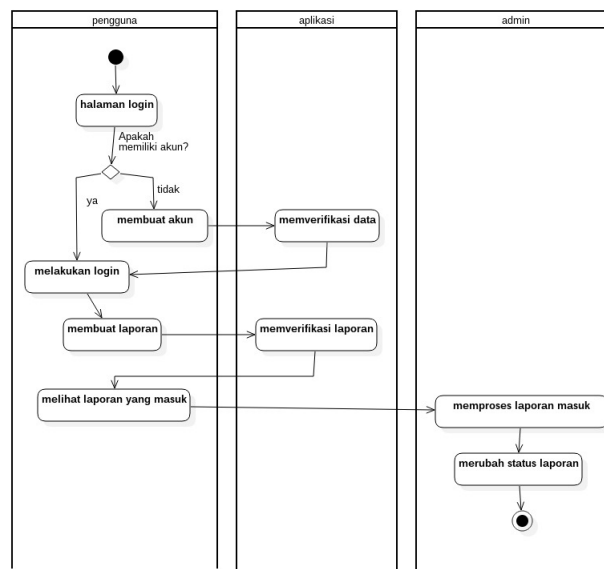
Gambar 3.1: Use Case Diagram

Pada gambar 3.1 menggambarkan interaksi antara pengguna dengan sistem, dimana terdapat 2 aktor yaitu pengguna dan admin. Sebelum menggunakan aplikasi pengguna wajib membuat akun terlebih dahulu kemudian

login dengan menggunakan akun yang telah dibuat. Setelah itu pengguna dapat membuat laporan, melihat laporan yang telah masuk, menghapus laporan yang telah dibuatnya, dan mereport laporan yang bermasalah. Sedangkan admin dapat melihat laporan yang masuk, menghapus laporan yang telah dibuat, melihat komentar pengguna, melihat laporan yang di report oleh pengguna, memproses laporan yang masuk, dan mengubah status laporan.

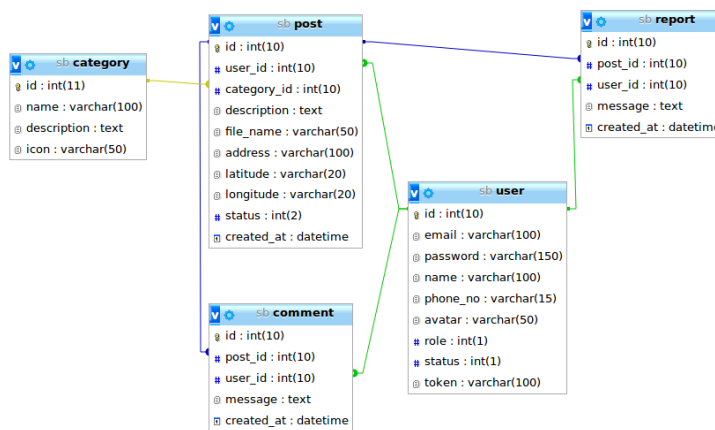
## Activity Diagram

Diagram Activity menggambarkan berbagai aktivitas untuk dapat mengirim laporan sampai laporan di selesaikan. Rancangan Activity Diagram dapat dilihat pada gambar 3.2.



Gambar 3.2: Activity Diagram

Pada gambar 3.2 menggambarkan alur aktivitas pada aplikasi Sahabat Depok. Pada saat pertama kali dibuka atau *initial state* pengguna akan diarahkan ke halaman login. Jika pengguna sudah memiliki akun pengguna langsung dapat melakukan login, jika tidak maka pengguna dapat membuat akun kemudian akan di verifikasi datanya oleh aplikasi. Setelah berhasil login maka pengguna dapat membuat laporan, kemudian laporan akan di verifikasi oleh aplikasi, jika laporan sudah di isi dengan lengkap maka pengguna dapat melihat laporannya di timeline maupun di peta. Setelah itu laporan akan di proses oleh admin jika sudah selesai maka status laporan dapat dirubah kemudian berakhir di *final state*.



Gambar 3.3: Rancangan Database

## Database

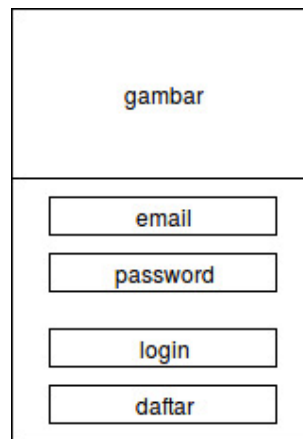
Struktur database menggambarkan kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Pada database menggambarkan relasi-relasi antar table, dimana terdapat 5 table yaitu user, category, post, report, dan comment. Pada table user terdapat data-data mengenai pengguna serta role, status, dan token pengguna. Pada table category berisi mengenai jenis-jenis laporan yang dapat dikirimkan oleh pengguna. Pada table post, berisi mengenai detail laporan pengguna berikut id kategori dan id user yang mengirim laporan. Pada table comment berisi mengenai komentar-komentar yang dikirim oleh pengguna terhadap suatu laporan. Pada table report berisi mengenai id laporan yang direport oleh pengguna dan pesannya. Rancangan database dapat dilihat pada gambar 3.3.

## Antarmuka

Interface (antarmuka) merupakan sarana berinteraksi antara pengguna dengan aplikasi. Antarmuka merupakan komponen yang berhubungan secara langsung dengan pengguna. Pembuatan antarmuka harus dibuat menarik dan memudahkan pengguna dalam berinteraksi dengan aplikasi. Rancangan antarmuka pada aplikasi Sahabat Depok terdiri dari rancangan halaman login.

## Halaman Login

Halaman login merupakan halaman yang digunakan pengguna untuk masuk ke dalam aplikasi dengan akun yang sudah di buatnya. Pada halaman login gambar statis, kolom input email dan password, tombol login, dan tombol daftar. Rancangan tampilan login dapat dilihat pada gambar 3.4.

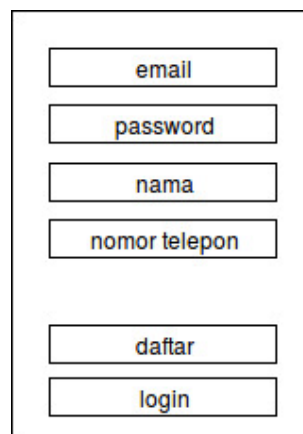


The diagram shows a vertical container for a login page. At the top is a rectangular box labeled 'gambar'. Below this box are four input fields stacked vertically: 'email', 'password', 'login', and 'daftar'. Each input field is represented by a rectangle with its label inside.

Gambar 3.4: Rancangan Halaman Login

## Halaman Daftar

Halaman daftar merupakan halaman yang digunakan pengguna untuk membuat akun. Pada halaman daftar terdapat kolom input email, password, nama, nomer telepon, tombol daftar, dan tombol login. Rancangan tampilan daftar dapat dilihat pada gambar 3.5.



The diagram shows a vertical container for a registration page. It contains six input fields stacked vertically: 'email', 'password', 'nama', 'nomor telepon', 'daftar', and 'login'. Each input field is represented by a rectangle with its label inside.

Gambar 3.5: Rancangan Halaman Daftar

### Halaman Timeline Laporan

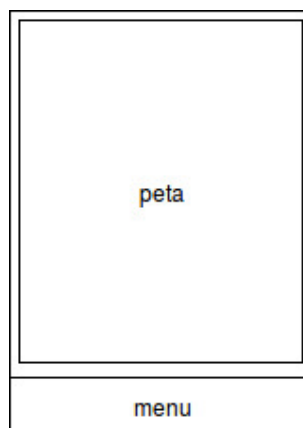
Halaman timeline laporan merupakan halaman yang pertama kali dilihat pengguna ketika sudah berhasil login. Pada tampilan ini, laporan ditampilkan dalam bentuk timeline berdasarkan urutan waktu dari yang terbaru hingga yang terlama. Rancangan tampilan timeline laporan dapat dilihat pada gambar 3.6.



Gambar 3.6: Rancangan Timeline Laporan

### Halaman Peta Laporan

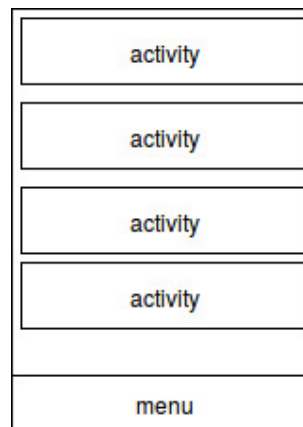
Halaman Peta Laporan merupakan halaman yang digunakan untuk melihat laporan di dalam peta. Rancangan tampilan peta laporan dapat dilihat pada gambar 3.7.



Gambar 3.7: Rancangan Peta Laporan

### Halaman Activity Pengguna

Halaman Activity merupakan halaman yang berisi aktivitas pengguna. Pada halaman ini ditampilkan komentar-komentar yang dikirim oleh pengguna terhadap suatu laporan. Rancangan tampilan activity pengguna dapat dilihat pada gambar 3.8.



Gambar 3.8: Rancangan Activity Pengguna

### Halaman Profile Pengguna

Halaman profile pengguna merupakan halaman yang berisi mengenai informasi pengguna serta laporan yang telah dikirimnya. Rancangan tampilan profile pengguna dapat dilihat pada gambar 3.9.



Gambar 3.9: Rancangan Halaman Login

# **Bab 4**

## **HASIL DAN PEMBAHASAN**

**Hasil**

**Pembahasan**

# Bab 5

## PENUTUP

### Kesimpulan

Berdasarkan hasil penelitian, maka ditunjukkan bahwa algoritma singular untuk perhitungan ..... dapat ditransformasikan kedalam algoritma paralel GPU. Proses transformasi paralel berdasarkan distribusi tahap perhitungan kedalam blok pemrosesan di GPU. Implementasi library CUDA digunakan untuk pemrograman dengan menggunakan bahasa pemrograman C++ . Hasil percobaan menunjukkan bahwa speed-up algoritma paralel terhadap algoritma singular rata-rata

sebesar .... terhadap tingkat pertambahan jumlah data sebesar  $10^n$ . Sehingga dapat disimpulkan bahwa algoritma paralel GPU untuk perhitungan ..... dapat digunakan untuk mempercepat waktu pemrosesan, khususnya untuk data yang berukuran sebesar  $10^x$ .

### Saran

Penelitian ini telah menghasilkan sebuah contoh implementasi algoritma paralel GPU untuk perhitungan ..... Tentu saja algoritma tersebut dapat ditingkatkan tingkat efisiensinya untuk menambah tingkat speed-up algoritma paralel. Diharapkan untuk penelitian selanjutnya dapat meneliti beberapa aspek perhitungan yang dapat diparalelkan kembali, ataupun menggunakan pendekatan algoritma paralel lainnya sehingga dapat meningkatkan pemanfaatan teknologi GPU.



# DAFTAR PUSTAKA

- [1] Yudha Yogasara. *Teknik Hacking untuk Pemula*. Elexmedia Komputindo, 2008.
- [2] Yudha Yogasara. *Google Dorks; Senjata Hacking Dengan Google*. Neome-dia, 2009.

# **LAMPIRAN**