

PERANCANGAN SISTEM INFORMASI GEOGRAFIS PERPAJAKAN PADA PERANGKAT BERGERAK MENGGUNAKAN SISTEM OPERASI ANDROID

Mario Christy Sinuraya^{*)}, Kodrat Iman Satoto, and R. Rizal Isnanto

Jurusan Teknik Elektro, Universitas Diponegoro Semarang
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)}E-mail: *mario.sinuraya@gmail.com*

Abstrak

Pajak merupakan sumber pendapatan negara yang terbesar. Namun kesadaran warga Negara Indonesia untuk membayar pajak masih sangat rendah. Pajak Bumi dan Bangunan (PBB) adalah salah satu pajak yang sering kali diacuhkan oleh para wajib pajaknya. Salah satu solusi untuk mengatasi masalah tersebut adalah dengan penarikan pajak secara langsung oleh petugas kepada wajib pajak. Untuk mempermudah proses penarikan pajak ini maka perlu dibangun sebuah Sistem Informasi Geografis (SIG) yang bisa diakses melalui perangkat bergerak yang mampu menyediakan lokasi wajib pajak yang telah jatuh tempo dan besar tagihan yang dikenakan padanya. Penelitian ini menghasilkan sebuah sistem informasi geografis yang mampu membantu menyajikan data-data yang dibutuhkan oleh petugas penagih pajak dalam melaksanakan tugasnya.

Kata Kunci : Sistem Informasi Geografis, Pajak Bumi dan Bangunan, Android, PHP, CodeIgniter

Abstract

Taxes are the largest source of state revenue. However, awareness of Indonesian citizens to pay taxes is still very low. Land and Building Tax is one of the taxes that is often ignored by the tax payers. One solution to overcome this problem is by direct taxation by the officers to the taxpayers. To simplify the process of tax collection, it is necessary to build a Geographic Information System (GIS) which can be accessed through mobile devices which are able to provide the location and the number of tax bill from taxpayers who has passed the payment date. In this final project, a Geographic Information System had been developed. The system can present the data which is required by the tax collector officers in carrying out their duties.

Keywords : Geographic Information System, Land and Building Tax, Android, PHP, CodeIgniter

1. Pendahuluan

Pajak merupakan sumber pendapatan negara yang terbesar. Namun kesadaran warga Negara Indonesia untuk membayar pajak masih sangat rendah. Pajak Bumi dan Bangunan (PBB) adalah salah satu pajak yang sering kali diacuhkan oleh para wajib pajaknya.

Salah satu solusi untuk mengatasi masalah tersebut adalah dengan penarikan pajak secara langsung oleh petugas kepada wajib pajak. Untuk mempermudah proses penarikan pajak ini maka perlu dibangun sebuah Sistem Informasi Geografis (SIG) yang bisa diakses melalui perangkat bergerak yang mampu menyediakan lokasi wajib pajak yang telah jatuh tempo dan besar tagihan yang dikenakan padanya.

Dalam Penelitian ini dirancang sebuah sistem informasi geografis pada perangkat bergerak yang sistem informasi geografis yang mampu membantu menyajikan data-data yang dibutuhkan oleh petugas penagih pajak dalam melaksanakan tugasnya sehingga dapat mempermudah pekerjaan petugas.

2 Metode

2.1. Sistem Informasi

Sistem informasi adalah segala sesuatu yang bekerja bersama-sama yang dibuat oleh manusia untuk mencapai suatu tujuan yaitu menyajikan informasi. Komponen sistem informasi adalah perangkat lunak, perangkat keras, manusia, data dan prosedur. Sebuah sistem informasi merupakan gabungan antara perangkat keras dan

perangkat lunak yang mengolah data dan menyajikan informasi yang dibutuhkan.

2.2. Sistem Informasi Geografis

SIG (Sistem Informasi Geografis) merupakan teknologi untuk mengelola, menganalisa dan menyebarkan informasi geografis. SIG adalah sistem yang berbasis komputer yang digunakan untuk menyimpan data dan manipulasi informasi geografis. SIG atau GIS (*Geographic Information System*) suatu bentuk sistem informasi yang menyajikan informasi dalam bentuk grafis dengan menggunakan peta sebagai antar muka.

2.2.1 Komponen-komponen dalam SIG

Aplikasi SIG dapat berjalan apabila dilengkapi dengan komponen-komponen yang terdiri dari:

1. perangkat keras : alat masukan (input), alat pemrosesan, alat keluaran (output)
2. perangkat lunak
3. manusia sebagai pengguna.

2.2.2 Data Spasial dalam SIG

Data spasial adalah data yang bereferensi geografis atas representasi objek di bumi. Data spasial pada umumnya berdasarkan peta yang berisikan interpretasi dan proyeksi seluruh fenomena yang berada di bumi. Fenomena tersebut berupa fenomena alamiah dan buatan manusia.

Data spasial memiliki dua jenis tipe yaitu vektor dan raster. Model data vektor menampilkan, menempatkan, dan menyimpan data spasial dengan menggunakan titik-titik, garis-garis atau kurva, atau poligon beserta atribut-atributnya. Model data Raster menampilkan, dan menyimpan data spasial dengan menggunakan struktur matriks atau piksel-piksel yang membentuk grid. Pemanfaatan kedua model data spasial ini menyesuaikan dengan peruntukan dan kebutuhannya.

2.2.3 Sumber Data Spasial

Data Spasial yang digunakan dalam SIG berasal dari :

1. Peta analog
2. Data dari sistem penginderaan jauh
3. Data hasil pengukuran lapangan
4. Data GPS

2.3 Sistem Operasi Android

Android adalah sebuah sistem operasi untuk perangkat bergerak berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru dalam pengembangan piranti lunak untuk ponsel atau

smartphone. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance* yang merupakan konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, T-Mobile dan Nvidia.

2.3.1 Android Software Development Kit

Android SDK adalah *tools API (Application Program Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemrograman Java. Android merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang dirilis oleh Google.

2.3.2. The Dalvik Virtual Machine (DVM)

Semua perangkat keras yang berbasis Android dijalankan dengan menggunakan virtual machine untuk eksekusi aplikasi sehingga pengembang tidak perlu khawatir tentang implementasi perangkat keras tertentu. Dalvik Virtual Machine mengeksekusi *executable file*, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *Executable file* tersebut diciptakan dengan mengubah kelas bahasa java dan dikompilasi menggunakan tools yang disediakan dalam SDK Android.

2.3.3. Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

2.3.4. Google Maps API

Google Maps adalah layanan gratis Google yang cukup populer. pengembang dapat menambahkan fitur Google Maps dalam web yang dikembangkannya dengan Google Maps API.

2.4 PHP

PHP merupakan singkatan dari *PHP Hypertext Preprocessor* yang merupakan bahasa berbentuk skrip yang ditempatkan pada *server* dan diproses di *server*. Hasilnya kemudian dikirimkan ke *browser* klien.

2.5 MySQL

MySQL merupakan basis data yang menggunakan konsep RDBMS, sebuah tabel merupakan struktur penyimpanan dasar. Satu tabel atau lebih membentuk sebuah basis data relasional. MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (General Public Licence). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk komersial.

2.6 CodeIgniter

CodeIgniter adalah sebuah *framework* PHP yang dapat membantu mempercepat developer dalam pengembangan aplikasi web berbasis PHP dibanding jika menulis semua kode program dari awal.

CodeIgniter adalah *framework* PHP yang dibuat berdasarkan kaidah *Model-View-Controller*. Dengan MVC, maka memungkinkan pemisahan antara *layer application-logic* dan *presentation*. Sehingga, dalam sebuah pengembangan web, seorang *programmer* bisa berkonsentrasi pada *core-system*, sedangkan web *designer* bisa berkonsentrasi pada tampilan web. Menariknya, skrip PHP, *query* MySQL, Javascript dan CSS bisa saling terpisah, tidak dibuat dalam skrip berukuran besar yang membutuhkan *resource* besar pula untuk mengesekusinya.

2.7. Spesifikasi Sistem

Penelitian ini terbagi menjadi 2 bagian, yaitu aplikasi pada ponsel Android (selanjutnya disebut ponsel) dan *website* yang berfungsi sebagai penyedia data koordinat lokasi dan data para wajib pajak. Pemilihan *website* sebagai penyedia data bertujuan agar meringankan kerja aplikasi, sehingga kinerja aplikasi akan lebih efisien.

Aplikasi yang dibuat pada Penelitian ini berfungsi untuk menampilkan data posisi dan data pribadi wajib pajak. Untuk proses penambahan dan pengolahan data akan ditangani oleh *website*. Pembagian tugas tersebut bertujuan untuk mempermudah proses penambahan dan perubahan data. Karena data yang dibutuhkan berasal dari *website* maka ponsel yang digunakan harus terhubung dengan Internet.

Website dirancang mempunyai 2 buah level pengguna, yaitu admin dan operator. Admin mempunyai akses penuh terhadap penambahan dan penyuntingan data, sedangkan operator hanya dapat mengakses data tanpa mampu melakukan proses penambahan dan penyuntingan. Pembatasan aksesibilitas dari operator bertujuan untuk menjamin keamanan data wajib pajak. Sedangkan untuk membatasi akses ponsel terhadap *website*, maka digunakan pembatasan menggunakan kode MEID dari ponsel. Jika kode MEID tidak terdaftar pada *website*,

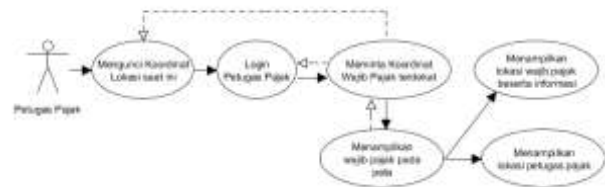
maka ponsel tidak mengakses data dari *website* meski telah terinstalasi aplikasi tersebut

2.8. Diagram Perancangan Aplikasi Android

2.8.1. Diagram Fungsional

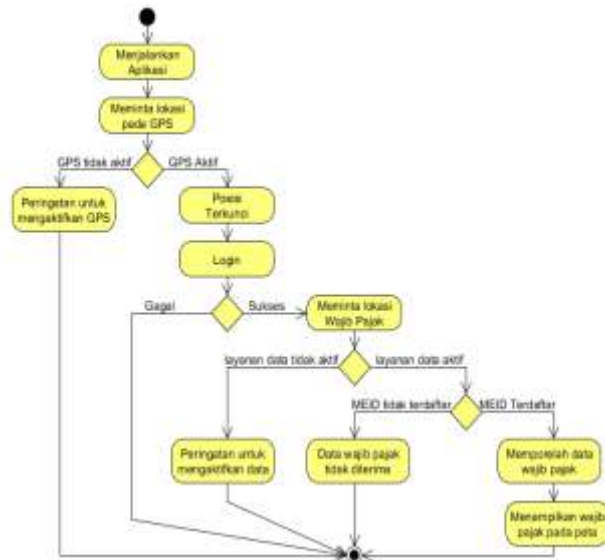
1. Diagram Use-case

Diagram *use case* adalah sebuah diagram yang digunakan untuk menunjukkan tampilan grafis dari fungsionalitas yang diberikan oleh sistem dilihat dari sisi aktor, tujuan aktor, dan hal yang berkaitan dengan *use case* yang ada. Berikut ini adalah diagram *uce-case* dari aplikasi Android.



Gambar 1. Diagram *use-case* aplikasi Android

2. Diagram Aktivitas

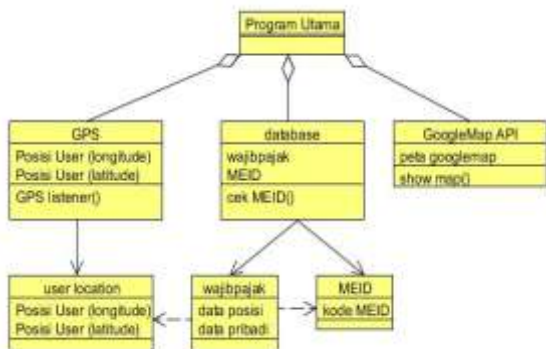


Gambar 2. Diagram aktivitas aplikasi Android

Diagram aktivitas memodelkan tingkah laku dalam sebuah proses. Diagram ini juga merupakan salah satu bagian dari diagram fungsional, dan juga seperti halnya dengan diagram *use-case*, diagram aktivitas juga digunakan pada fase analisis. Berikut ini adalah diagram aktivitas dari aplikasi Android

2.8.2 Diagram Struktural

Diagram ini diwakilkan oleh diagram kelas yang merupakan sebuah tampilan statis struktur program. Diagram kelas menggambarkan struktur dengan memperlihatkan kelas dari sistem, baik atribut maupun relasi dengan kelas lain. Gambar 3 menunjukkan diagram kelas dari aplikasi Android.



Gambar 3. Diagram kelas dari aplikasi Android

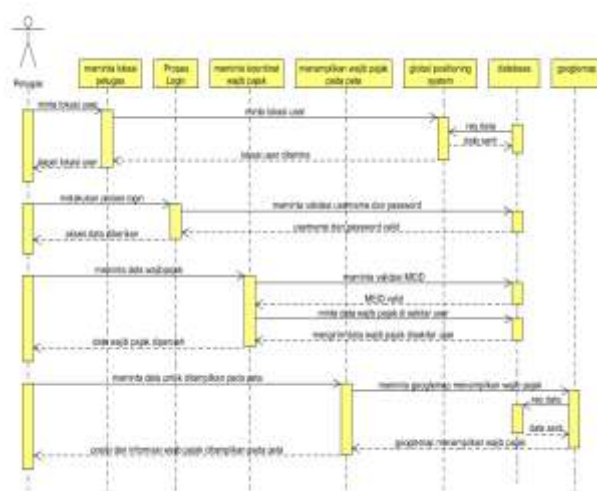
Diagram di atas menunjukkan kelas-kelas yang ada pada aplikasi Android. Kelas GPS mempunyai turunan kelas user location yang menentukan akses program utama terhadap kelas wajib pajak. Jika kelas user location tidak diakses maka wajib pajak juga tidak dapat diakses. Selain oleh kelas user location, akses pada kelas wajib pajak juga dibatasi oleh kelas MEID. Kelas GoogleMap API berfungsi untuk memanggil peta GoogleMap yang nantinya akan dipanggil oleh program, utama setelah kelas GPS dan basis diakses.

2.8.3. Diagram Tingkah Laku

Diagram urutan adalah sebuah diagram interaksi di dalam UML yang menunjukkan proses suatu operasi dengan lainnya dengan fokus ke arah urutan dari proses mana yang mendahului maupun yang tertinggal.

Diagram urutan digunakan untuk menunjukkan interaksi antar pengguna dengan objek-objek lain yang terkait. Diagram ini lebih menitikberatkan pada urutan pesan yang dikirimkan antar pengguna dan obyek berdasarkan waktu. Biasanya diagram ini digunakan untuk menggambarkan skenario dalam suatu use-case yang ingin dijelaskan secara detail atau juga dapat digunakan untuk menggambarkan sebuah proses dalam use-case.

Diagram urutan menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, tampilan, dan sebagainya) berupa pesan yang digambarkan terhadap waktu. Diagram urutan terdiri antar dimensi vertikal (waktu) dan dimensi horisontal (objek-objek yang terkait).



Gambar 4. Diagram urutan dari aplikasi Android

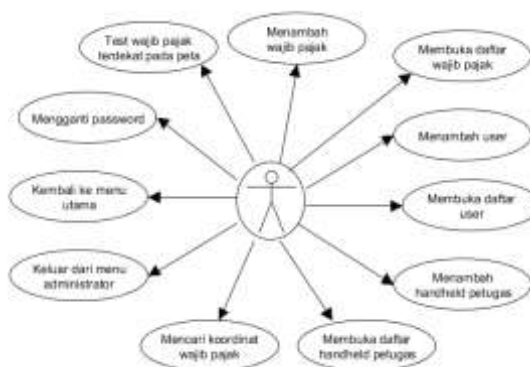
Dari diagram di atas dapat dilihat bahwa ada 4 aktivitas utama yang dilakukan oleh pengguna, yaitu meminta lokasi petugas, melakukan login, meminta koordinat wajib pajak dan menampilkan wajib pajak peta. Aktivitas meminta login, melakukan validasi *username* dan *password* kepada basis data dan apabila dinyatakan valid maka akses pada data akan diberikan

2.9. Diagram Perancangan Website

2.9.1. Diagram Fungsional

1. Diagram Use-case

Diagram *use-case website* ditunjukkan oleh Gambar 5.

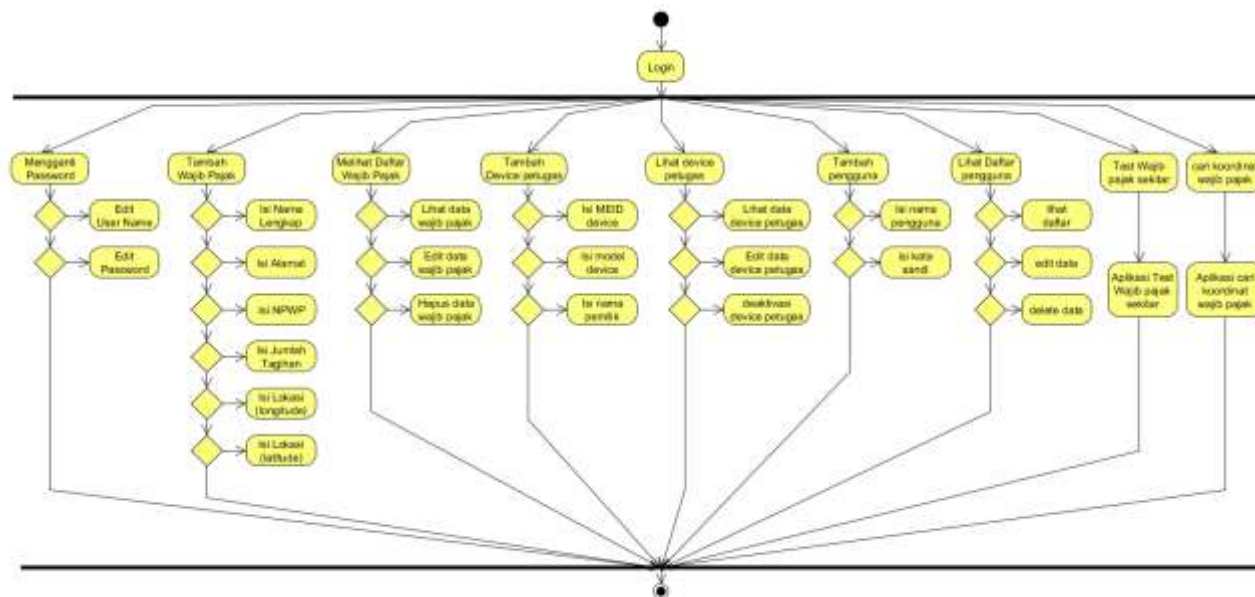


Gambar 5. Diagram use-case Admin pada website

Gambar di atas adalah diagram *use-case* dari administrator. Administrator mempunyai 11 *use-case* yang berguna untuk menyunting tabel yang ada pada basis data. *Use-case test wajib pajak terdekat pada peta* dan *mencari koordinat wajib pajak*, digunakan admin untuk memperoleh koordinat dan data wajib pajak dan mensimulasikannya pada peta GoogleMaps.

2. Diagram Aktivitas

Diagram aktivitas *website* ditunjukkan oleh Gambar 6.



Gambar 6. Diagram aktivitas *website*

Setelah berhasil melakukan *login*, admin dapat melakukan 9 buah aktivitas. Aktivitas tersebut berfungsi untuk menambahkan atau menyunting data yang ada pada basis data. Untuk proses penambahan data dilakukan pada aktivitas tambah wajib pajak, tambah device petugas dan tambah pengguna. Untuk proses penyuntingan data ditangani oleh aktivitas melihat daftar wajib pajak, lihat device petugas dan lihat daftar pengguna. Untuk aktivitas melihat daftar wajib pajak dan lihat daftar pengguna, selain dapat melakukan penyuntingan, kedua aktivitas ini dapat melakukan penghapusan data. Sedangkan untuk aktivitas lihat device petugas aktivitas penghapusan diganti dengan aktivitas aktivasi/deaktivasi ponsel Android (*device*).

Untuk pencarian data koordinat yang dibutuhkan pada aktivitas tambah wajib pajak, maka dibuat aktivitas cari koordinat wajib pajak yang berinteraksi dengan GoogleMaps.

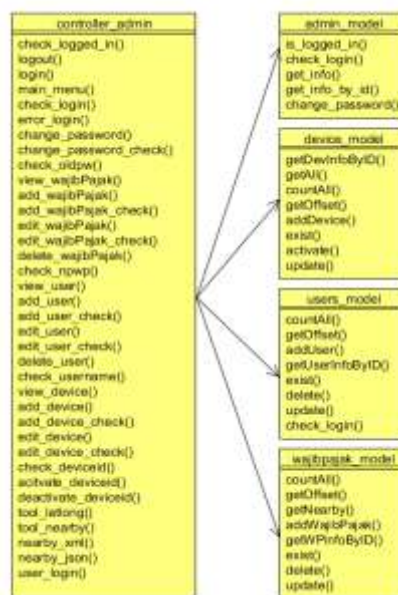
2.9.2 Diagram Struktural

Diagram ini diwakilkan oleh diagram kelas yang merupakan sebuah tampilan statis struktur program. *Website* ini mempunyai 5 kelas, 1 kelas controller dan 4 buah kelas model. Kelas controller (*controller_admin*) mempunyai banyak *method*. Beberapa *method* yang dimiliki kelas **controller_admin**, membutuhkan akses terhadap basis data. Untuk mengatur akses terhadap basis data maka diperlukan kelas model.

Pada diagram kelas di atas terdapat 4 buah kelas model, yaitu **admin_model**, **device_model**, **wajibpajak_model** dan **users_model**. Tiap-tiap model mengatur akses ke

tabel yang berbeda pada basis data. Model **admin_model** mengatur akses ke tabel **admin**. Model **device_model**

mengatur akses ke tabel **wajibpajak**. Model **users_model** mengatur akses ke tabel **users**. Untuk lebih jelasnya dapat dilihat diagram kelas *website* yang ditunjukkan oleh Gambar 7.

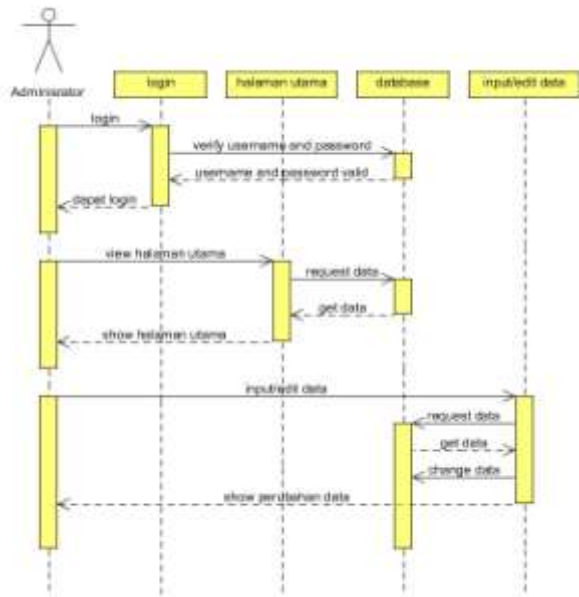


Gambar 7. Diagram kelas *website*

2.9.3. Diagram Tingkah Laku

Diagram urutan digunakan untuk menunjukkan interaksi antar pengguna dengan objek-objek lain yang terkait.

Diagram urutan dari *website* pada level pengguna administrator dapat dilihat pada Gambar 8.



Gambar 8. Diagram urutan *website*

Admin harus melakukan login terlebih dahulu. Bila *username* dan *password* sesuai dengan basis data maka akses terhadap *website* diberikan pada admin. Selanjutnya admin masuk ke dalam halaman utama dan dapat memulai tugasnya untuk menambah dan menyunting data

3. Hasil dan Analisis

3.1. Pengujian Website

Website mempunyai banyak menu yang berguna untuk menambah dan menyunting data pada basis data. Selain itu website juga mempunyai menu untuk melakukan simulasi pemanggilan peta yang disertai dengan penampilan data wajib pajak di dalamnya.

3.1.1. Pengujian Menu Login

Menu login digunakan untuk membatasi akses terhadap website, sehingga hanya pengguna yang mempunyai *username* dan *password* saja yang bisa mengaksesnya.



Gambar 9. Halaman Login Website

Pemanggilan menu login dilakukan dengan senarai yang dapat dijabarkan sebagai berikut.

```

function check_login($username, $password) {
    $this->db->where('username', $username);
    $this->db->where('password',
    md5($password));
    $query=$this->db->get('admin');
    if ($query->num_rows()>0) {
        return TRUE;
    } else {
        return FALSE;}}
    
```

3.1.2. Pengujian Menu Tambah Wajib Pajak

Menu **Tambah Wajib Pajak** berguna untuk menambah data ke tabel **wajibpajak** pada basis data. Berikut tampilan menu Tambah Wajib Pajak yang berhasil dipanggil



Gambar 10. Menu Tambah Wajib Pajak

Setelah semua data diisi dan tombol tambahkan ditekan maka terjadi penambahan pada tabel **wajibpajak** pada basis data. Pemanggilan menu tambah wajib pajak dilakukan dengan senarai yang dapat dijabarkan sebagai berikut.

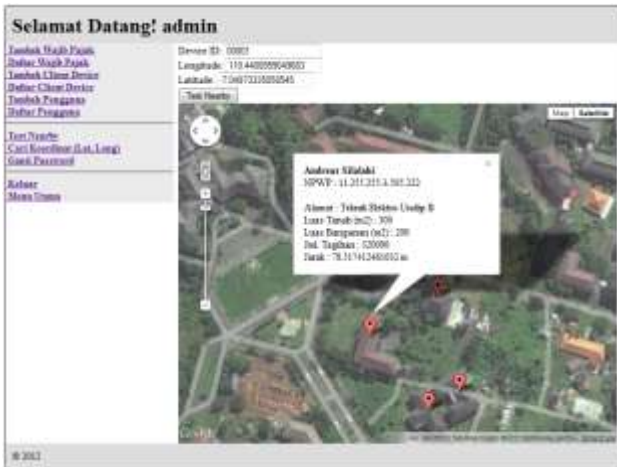
```

function add_wajibPajak() {
    $this->load->model('admin_model');
    if (!$this->admin_model->is_logged_in())
    {
        redirect('admin/login');
        return;
    }
    $data['title']='Tambah Baru - Wajib Pajak';
    $data['sidebar']=$this->load-
    >view('admin/sidebar','', true);
    $this->load->view('admin/add_wajibpajak',
    $data);}
    
```

3.1.2. Pengujian Menu Test Nearby

Menu Test Nearby berfungsi untuk memberkan simulasi data wajib pajak kepada pengguna. Dalam simulasi data-data tabel wajibpajak yang berada di sekitar koordinat uji coba akan ditampilkan pada peta GoogleMaps. Dengan

demikian data yang nantinya dikirim ke aplikasi Android semakin teruji. Berikut tampilan dari menu Test Nearby.



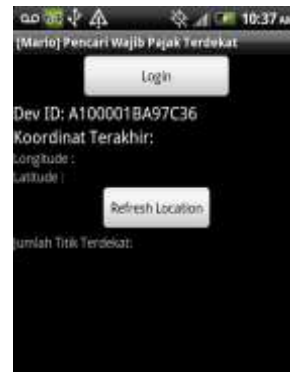
Gambar 11. Menu Test Nearby

Pemanggilan menu Get Nearby dilakukan dengan senarai yang dapat dijabarkan sebagai berikut.

```
function tool_nearby() {
//prototype untuk test nearby
    $this->load->model('admin_model');
    if
(!$this->admin_model->is_logged_in()) {
        redirect('admin/login');
        return;
    }
    $data['extra_js']=$this->load-
>view('admin/gmaps_nearby_js','',TRUE);
    $data['title']='Test nearby';
    $data['sidebar']=$this->load-
>view('admin/sidebar','', true);
    $data['admin_name']=$this->session-
>userdata('username');
    $this->load-
>view('admin/gmaps_nearby_test', $data);
}
```

3.2. Pengujian Aplikasi Android

Pengujian ini dilakukan menggunakan ponsel Android dengan merk HTC tipe EVO 4G dengan sistem operasi Android versi 2.3.3 GingerBread. Ponsel ini mempunyai kode MEID A10001BA97C36 yang telah terdaftar pada tabel **device** pada basis data. Berikut ini adalah tampilan halaman utama dari aplikasi Android



Gambar 12. Halaman Utama Aplikasi Android

3.2.1 Pengujian Login

Tombol **login** ini berfungsi untuk membatasi akses terhadap aplikasi sehingga hanya yang mempunyai *username* dan *password* saja yang dapat menjalankan aplikasi ini dengan sempurna. Ketika tombol **login** disentuh maka ia akan memanggil form login. Berikut tampilan dari form login.

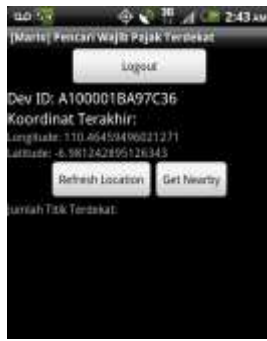


Gambar 13. Halaman Form login

Setelah login berhasil dilakukan maka tombol login akan berubah nama menjadi log out. Pada halaman utama juga terjadi penambahan tombol baru yaitu tombol Get Nearby.

3.2.2 Pengujian Tombol Refresh Location

Tombol **refresh location** berfungsi untuk mengunci lokasi pengguna kepada GPS. Namun sebelum menekan tombol tersebut, fasilitas GPS pada ponsel harus dalam keadaan aktif. Jika GPS dalam kondisi tidak aktif maka akan muncul pesan untuk menyalakan GPS. Jika GPS telah dinyalakan dan posisi telah dikunci oleh GPS maka pada layar akan muncul pesan " Done update location request". Setelah itu koordinat dari pengguna ditampilkan pada kolom longitude dan latitude. Berikut tampilan halaman utama setelah tombol **refresh location** disentuh.



Gambar 14. Tampilan layar setelah posisi terkunci

3.2.3. Pengujian Tombol Tampilkan Peta

Tombol **tampilkan peta** awalnya disembunyikan. Tombol ini baru akan muncul setelah tombol **Get Nearby** disentuh. Activity yang digunakan pada tombol ini tidak hanya **MainActivity** saja, melainkan ada 1 activity lagi yang berperan yaitu **MapsActivity**. Penggunaan **MapsActivity** pada tombol ini dikarenakan fungsi ini adalah untuk memanggil peta. Peta yang digunakan adalah peta GoogleMaps.

MapsActivity mempunyai beberapa tugas, yaitu memanggil peta dari GoogleMaps, menampilkan titik-titik wajib pajak pada peta, memunculkan informasi dari titik pengguna (operator) dan menampilkan informasi dari titik-titik wajib pajak.

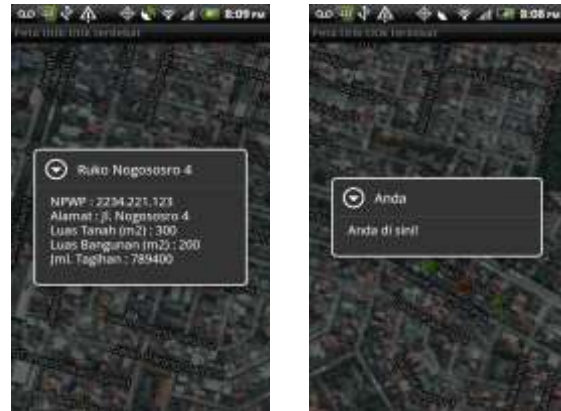
Ketika tombol **tampilkan peta** disentuh maka peta tidak muncul di halaman utama, melainkan akan memanggil halaman baru yang hanya berisikan peta saja. Jika pengguna (operator) berpindah lokasi maka ia harus keluar dari halaman peta dan langsung masuk ke halaman peta untuk mengunci lokasi yang baru dan meminta data kembali. Berikut ini tampilan halaman peta yang berhasil dipanggil oleh tombol **tampilkan peta**.



Gambar 15. Tampilan halaman peta

Ketika salah satu titik wajib pajak (berwarna hijau) maka akan muncul informasi pribadi dari wajib pajak tersebut. Sedangkan bila titik merah yang ditekan maka akan

muncul informasi yang menunjukkan bahwa itu posisi operator. Untuk lebih jelasnya dapat dilihat pada gambar berikut.



Gambar 16. Tampilan pesan titik hijau (kiri) dan merah (kanan)

4. Kesimpulan

Dari proses pengujian dapat disimpulkan bahwa sistem informasi geografis perpajakan pada perangkat bergerak menggunakan sistem operasi Android berjalan dengan baik. Data dari basis data dapat diterima oleh aplikasi dan ditampilkan pada peta dengan lancar. Ada waktu tunda yang berbeda-beda ketika melakukan penguncian lokasi kepada GPS dan ketika memanggil peta GoogleMaps. Hal tersebut terjadi bukan dikarenakan kesalahan sistem melainkan adanya faktor dari luar seperti lokasi pengguna dan layanan data yang digunakan ponsel. Aplikasi Android masih dapat dikembangkan dengan penambahan fitur-fitur terbaru seperti teknologi face detection yang dimiliki oleh Android 4.0 Ice Cream Sandwich yang diaplikasikan kepada menu login. Selain itu aplikasi juga dapat dikembangkan dengan menambah fitur kunci posisi otomatis dan meminta (*request*) data secara otomatis.

Referensi

- [1]. Basuki, Awan Pribadi. *Membangun Web Berbasis PHP dengan Framework CodeIgniter*. Lokomedia, 2010.
- [2]. Kadir, A. *Penuntun Praktis Belajar SQL*. Penerbit Andi, 2002
- [3]. Mulyadi. *Membuat Aplikasi Untuk Android*. Multimedia Centre Publishing, 2010.
- [4]. Munawar. *Pemodelan Visual dengan UML*. Graha Ilmu, 2005.
- [5]. Riyanto. *Membuat Sendiri Aplikasi Mobile GIS platform Java ME, BlackBerry, & Android*. Andi Offset, 2010.
- [6]. Safaat, Nazruddin. *Android : Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Informatika, 2011.