

PENGEMBANGAN APLIKASI HISTORY GPS TRACKER BERBASIS WEB PADA HANDPHONE

Mustar Aman, M.Kom
mustar_ip@yahoo.com

STMIK Insan Pembangunan
Jl. Raya Serang Km.10 Bitung-Tangerng

ABSTRAK

Global Positioning System (GPS) merupakan sistem satelit navigasi yang sedang populer dan banyak diaplikasikan di dunia, baik di darat, di laut, di udara, maupun di angkasa. Saat ini, beberapa *handphone* tertentu sudah di sertai fasilitas *Gps*. Dengan adanya *Gps* pada *handphone* banyak sekali manfaat yang kita dapatkan, salah satunya adalah mengetahui posisi kita saat ini. Pekerjaan yang kita miliki sering kali menyita waktu kita sehingga kita tidak dapat mengawasi keluarga dan anak-anak kita. Diharapkan dengan adanya aplikasi ini kita dapat mengetahui posisi terakhir seseorang jika terjadi penculikan atau hal-hal yang kita tidak inginkan pada keluarga kita. Dengan adanya aplikasi ini kita dapat dengan mudah melacak posisi terakhir seseorang. Penelitian ini bertujuan membuat aplikasi yang mampu melacak keberadaan seseorang dengan memanfaatkan fasilitas *Gps* yang terdapat pada *handphone*. Metode yang di gunakan pada penelitian ini adalah model proses extreme programming karena aplikasi yang akan dibuat tidak membutuhkan waktu yang lama. Bahasa pemrogramannya menggunakan *J2ME* pada *handphone* dan *ASP.NET/C#* pada webnya dengan *SQLServer* sebagai databasenya. Inti dari penelitian ini yaitu terletak pada data-data *latitude* dan *longitude* yang dikirimkan oleh *handphone* ke server. Data dan informasi yang ditampilkan pada aplikasi ini diharapkan dapat menjawab masalah bagi orang tua untuk mengetahui posisi anak atau keluarganya berada.

Kata Kunci: *Gps, Handphone, extreme programming, J2ME, ASP.NET, C#, SQLServer*

PENDAHULUAN

Global Positioning System (GPS) merupakan sistem satelit navigasi yang banyak diaplikasikan dimana saja, salah satunya di andpone. Dengan adanya *GPS* pada *handphone* banyak sekali manfaat yang kita dapatkan. Salah satunya mengetahui lokasi kita berada saat ini. Aplikasi *GPS* bawaan yang terdapat pada *handphone* cenderung hanya memberikan informasi posisi pemilik dan belum dapat memberikan informasi posisi orang lain yang kita inginkan, sebut saja keluarga kita. Pekerjaan

yang kita miliki sering kali menyita waktu kita sehingga kita tidak dapat mengawasi keluarga dan anak-anak kita.

Diharapkan dengan adanya aplikasi ini kita dapat mengetahui posisi terakhir seseorang jika terjadi penculikan dan sebagainya. Dengan demikian kita dapat dengan mudah melacak posisi orang tersebut.

Pada penelitian ini peneliti membuat aplikasi yang dapat melacak posisi orang lain misalnya keluarga kita yaitu dengan cara melacak posisi *handphone* yang digunakan. Aplikasi ini lebih murah karena memanfaatkan

handphone GPS yang dapat digunakan sebagai alat komunikasi dibandingkan dengan *GPS Personal* yang banyak di tawarkan di internet. Selain itu, aplikasi ini dapat di kembangkan sesuai dengan kebutuhan.

METODOLOGI

Penelitian ini dilakukan pada tahun 2010 dengan 2 (dua) metodologi. Metodologi yang pertama adalah metodologi pengumpulan data, adapun metode yang peneliti gunakan untuk mendapatkan data-data tersebut dengan menggunakan studi pustaka dan literatur.

Pada pengumpulan data menggunakan studi pustaka, peneliti mengumpulkan teori-teori yang berhubungan dengan penelitian ini dari berbagai sumber diantaranya berasal dari jurnal, karya ilmiah, buku-buku ilmu komputer dan buku elektronik yang berstandar. Sedangkan pada pengumpulan data dengan studi literature penelitian ini di ambil dari proyek akhir yang dibuat oleh Tri Krida Hartanto Putro yang berjudul "*Sistem Online Untuk Pelacakan Paket Menggunakan GPS*". Proyek akhir tersebut memanfaatkan *SMS* sebagai pengiriman datanya, dimana *SMS* tersebut akan diterima oleh *GSM Modem* pada alat *GPS* kemudian *Mikrocontroller* membaca instuksi pada *SMS* dan mengeksekusinya. Jika instuksinya adalah *request* posisi terakhir, maka mikro *controller* akan mengambil data koordinat dari *GPS* kemudian mengirimkan data itu melalui *SMS* ke *GSM Modem Server*. Data koordinat itu kemudian ditampilkan pada peta digital di *browser*.

Metodologi yang kedua adalah metodologi pengembangan system, Metode pengembangan sistem yang akan digunakan yaitu metode yang mengikuti kaidah berbasis objek (*object oriented/OO*). Pada kasus ini kami menggunakan *Agile Software Development* dan proses model menggunakan *Extreme Programming* (yang selanjutnya akan disingkat

XP). Pemilihan metode ini dikarenakan aplikasi yang akan dibuat tidak terlalu kompleks dan tergolong perangkat lunak kecil dan juga hanya membutuhkan waktu pengerjaan yang singkat. *XP* mencakup beberapa aturan dan praktek yang terdiri atas *planning, design, coding* dan *testing*.

Pada tahap *planning* perencanaan terhadap *software* yang diinginkan mengacu pada *user stories*. *User stories* menggambarkan fitur dan fungsi yang dibutuhkan terhadap *software* tersebut. Ketika semua *user stories* telah ditentukan, *developer* akan menentukan lama pengerjaan untuk tiap-tiap *user stories*.

Pada tahap *design*, proses desain pada *XP* mengikuti prinsip *KIS (Keep It Simple)*. Desain akan berisikan semua implementasi dari *stories* tanpa ada pengurangan maupun penambahan. Pada tahap ini, perancangan aplikasi akan terdiri dari beberapa bagian diantaranya perancangan *class* yang dibutuhkan dengan *CRC*, perancangan *layout design*, dan perancangan desain *GUI*.

Pada tahap proses, proses pengembangan tidak langsung melakukan implementasi terhadap desain yang telah dibuat. Pembuatan *unit test* untuk tiap-tiap *stories* yang nantinya akan diimplementasikan. Saat *unit test* selesai dibuat, pengembang lebih baik fokus terhadap apa yang akan diimplementasikan untuk melewati *unit test*. Tahap ini akan mengacu pada desain sebelumnya. Karena pembuatan *unit test* dilakukan terlebih dahulu. Maka implementasi desain sebaiknya dibuat untuk melewati *unit test* yang dibuat. Aplikasi akan dibuat menggunakan *NetBeans IDE 6.8* dan *Visual Studio 2008* sebagai *GUI*.

Pada tahap *testing*, akan menggunakan *unit test* yang sebelumnya telah dibuat. Karena pembuatan dari *unit test* adalah pendekatan utama dari *XP*. *Unit test* yang dibuat diimplementasikan dengan penggunaan simulator untuk otomatisasi. *Testing* akan dilakukan dengan *unit test* yang sebelumnya

telah dibuat. *Wireless Toolkit* akan digunakan sebagai simulator untuk melakukan *testing* terhadap aplikasi.

KEBUTUHAN SISTEM

Gambaran Umum Sistem

Sistem yang akan di buat berbasis web dan mobile, dimana data akan dikirimkan dari aplikasi mobile ke server yang sudah di tanamkan aplikasi web, data yang di kirimkan adalah koordinat longitude dan latitude dari gps pada handphone.

Harapan

Harapanya sistem ini dapat bermanfaat untuk masyarakat, karna dengan system ini kita dapat mengetahui posisi sanak saudara kita.

Kebutuhan Fungsional

- Mengirim data koordinat ke server
- Menampilkan data secara real time
- Melihat history data sebelumnya

Kebutuhan Non Fungsional

- Ketersedian akses pada server
- Kecepatan aplikasi
- Keamanan aplikasi

Kebutuhan Performance

Untuk menunjang performance, Server harus memiliki spek yang memadai, sinyal GPS dan koneksi internet pada handphone harus stabil agar tidak terjadi kehilangan kontak antara handphone dengan server.

ANALISIS DAN DESIGN

Planning

Tahapan *planning* pada *XP* merupakan tahapan awal dari semua proses pengembangan sistem. Tahapan ini merupakan gabungan dari tahap analisis dan *planning* pada metode pengembangan klasik. Tahap *planning* pada

agile dibuat sesederhana mungkin agar tidak banyak dokumentasi namun *programmer* dapat mengerti apa yang harus dikodekan. Pada tahapan *planning* dapat diketahui gambaran besar dari aplikasi yang hendak dikembangkan dan teknologi-teknologi pendukung yang akan digunakan.

GPS Tracker adalah sebuah system *GPS* yang akan memberikan *Location Based Services (LBS)*. Antar muka dari *system* ini memiliki dua pendekatan teknologi yaitu teknologi berbasis *mobile* dan teknologi berbasis *web* yang terintegrasi. Antar muka yang berbasis *mobile* dapat memberikan informasi *latitude* dan *longitude* terkini. Sedangkan antar muka yang berbasis *web* akan memberikan *reporting* yang lebih kaya informasi meliputi lokasi terkini, *realtime latitude/longitude* dan *historical tracking* dalam rentang waktu tertentu yang mana rentang waktunya dapat ditentukan secara dinamis. Tampilan *latitude* dan *longitude* menggunakan aplikasi peta dari *google* yaitu *google maps*. Teknologi *google* dipilih dikarenakan *superioritas google* dalam teknologi ini dan juga kemudahan penggunaan *API* yang disediakan oleh *google*. Komunitas pengguna teknologi *Microsoft* di *codeplex* telah membuat *wrapper* pada *API google maps* sehingga *API* dapat lebih mudah lagi digunakan terutama oleh aplikasi yang hendak dikembangkan diatas teknologi *Microsoft*. Kemudahan yang didapat diantaranya adalah merubah paradigma *client-side* dari *google API* menjadi paradigma *server-side* yang akan memberikan kemudahan dalam pengembangan terutama pada saat *de-bugging* dan koneksi data ke *database back-end*. Dengan teknologi *update panel* dari *AJAX* maka paradigma *server-side* bisa terlihat seolah-olah seperti *client-side*.

Fitur kunci dari *system GPS Tracker* yang akan dirancang ini adalah sistem yang mudah untuk diinstal dan dioperasikan, *multiplatform* dengan penggunaan dua buah antarmuka yaitu *mobile based* dan *web based*, *low budget* karena menggunakan teknologi *GPRS* untuk komunikasi data *GPS*, *realtime information*

karena data *latitude* dan *longitude* langsung dikirimkan ke *database* yang mana dari *database* bisa langsung ditampilkan dan yang terakhir interaktif karena data *latitude* dan *longitude* bisa langsung ditampilkan dan disinkronkan dengan *google map*.

Tujuan dibuatnya system ini diantaranya adalah memberikan data *realtime* dari *latitude* dan *longitude*, menampilkan *history* dari *latitude* dan *longitude*, menampilkan manajemen *user* pengguna, menyediakan *framework* untuk pengembangan aplikasi *Location Based Services (LBS)* lebih lanjut.

Pihak yang akan menggunakan system ini adalah administrator dan *user*. Administrator adalah pihak pengelola aplikasi *web based* dan *mobile based*. Fungsi dari administrator untuk melakukan manajemen *user* dan manajemen *GPS*.

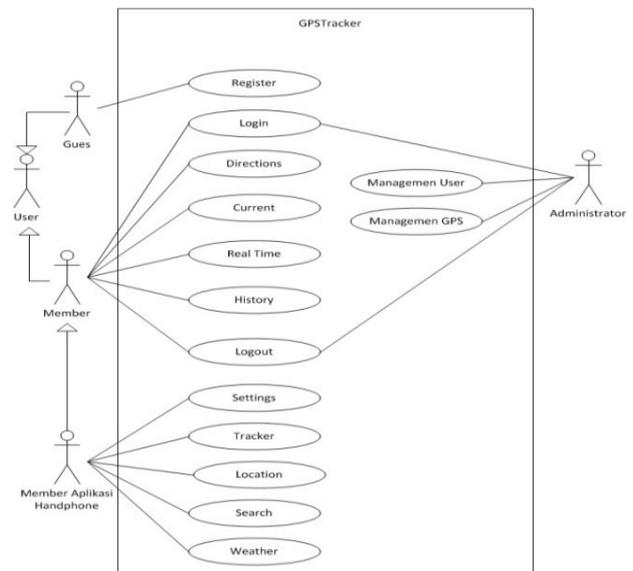
Informasi yang akan ditampilkan adalah informasi *latitude* dan *longitude*. Informasi lokasi tersebut akan ditampilkan dan disinkronkan dengan *google map* sebagai *map viewernya*. Informasi lokasi dapat berupa *current location*, *historical location* dan *realtime location*.

Sistem *GPS Tracker* membutuhkan beberapa teknologi pendukung baik dari sisi perangkat keras maupun dari sisi perangkat lunak. Perangkat keras yang digunakan yaitu *PC Hardware* dengan minimum 512 RAM dengan *Space* minimum 10 GB dan *Handphone* dengan *chip GPS*. Sedangkan perangkat lunak yang digunakan yaitu *Microsoft Windows Server 2003/Windows Server 2008*, *Microsoft .NET Framework 2.0 / 3.5 / 4.0*, *Microsoft IIS 6 / IIS 7*, *Microsoft SQL Server 2005/2008*, *DotNetNuke Web Application Framework*, dan *google map API/ subgurim framework*.

Perangkat keras dan perangkat lunak yang digunakan oleh peneliti dalam mengembangkan aplikasi *Rikarotrack* ini diperoleh melalui *erudeye shared hosting*. *Erudeye* adalah sebuah perusahaan *hosting* dan *cloud computing* yang memberikan layanan

virtualisasi *Microsoft server environment* dengan harga terjangkau.

Dari analisis kebutuhan di atas dapat kita buat *use cases*-nya sebagai berikut:



Dari analisis di atas, peneliti dapat menyimpulkan melalui *user stories* sebagai berikut:

1. *User Administrator* dan *User Klien* dapat *login* dan *logout*.
2. *User Administrator* mempunyai *username* dan *password*.
3. *User Administrator* dapat melakukan manajemen *user*.
4. *User Administrator* dapat melakukan manajemen *GPS*.
5. *User member* mempunyai *username*, *password* untuk melakukan *login*.
6. *User member* dapat melihat *current latitude* dan *longitude*.
7. *User member* dapat melihat *historical latitude* dan *longitude*.
8. *User member* dapat melihat *realtime latitude* dan *longitude*.
9. *User member* dapat melihat *directions latitude* dan *longitude*.
10. *User member aplikasi handphone* dapat menyetting *gps id* pada *handphone*.

11. *User member aplikasi handphone* dapat melakukan tracker pada *handphone*.
12. *User member aplikasi handphone* dapat melihat posisi saat ini pada *handphone*.
13. *User member aplikasi handphone* dapat mencari posisi orang lain pada *handphone*.
14. *User member aplikasi handphone* dapat melihat perkiraan cuaca sesuai posisi pada *handphone*.

Design

Dalam mendesain aplikasi ini peneliti menggunakan pendekatan yaitu membuat, *design CRD Card*, dan *design GUI*. Proses desain pada tahap *design CRD Card* menggunakan *Class Responsibilities, and Collaboration (CRC) card*. Penggunaan *CRC card* hanya dipakai jika pengembangan *software* berbasis objek. Setiap *CRC card* akan menjelaskan tiap objek yang dibutuhkan.

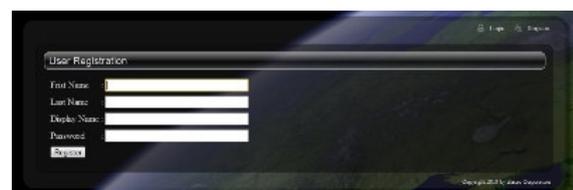
Pada tahap *design CRD Card* dibuat 5 (lima) kelas yaitu kelas *GetData*, kelas *Maps*, kelas *Register*, kelas *GPS*, dan kelas *Manager*. (1) **Kelas *GetData***, pengguna aplikasi dapat berinteraksi dengan *server*. Di mana paket data yang di kirimkan melalui *handphone* akan di olah di dalam kelas ini dan akan di simpan kedalam *database*. Fungsi kelas ini diantaranya menyimpan paket data yang di kirimkan oleh *user* dan menampilkan peta lokasi berupa *images* pada *handphone user*. (2) **Kelas *Maps***, *user* dapat melihat data dan informasi yang pernah *user* kirimkan sebelumnya. Fungsi dari kelas ini diantaranya melihat posisi terakhir *user* berada, melihat posisi *user* secara *real time*, melihat *history* perjalanan yang telah di lakukan *user* sebelumnya. (3) **Kelas *Register***, disinilah *user* pertama kali mendaftarkan dirinya. Fungsi dari kelas ini diantaranya mengecek aktifnya *email* yg di kirimkan oleh *user*, di mana untuk melakukan *registrasi user* harus memasukan alamat *email* yang aktif, setelah itu *user* akan

mendapatkan *email* konfirmasi berupa link untuk melanjutkan *registrasi*. Merandom *GPSID*, pada tahap ke dua *user* diwajibkan mengisi data-data yg di perlukan seperti nama dan *password*, setelah itu *user* akan mendapatkan *email* konfirmasi ke dua, di mana di dalam *email* tersebut terdapat informasi yang *user* butuhkan seperti *username*, *password*, dan *GPSID*. (4) **Kelas *GPS***, disinilah yang paling penting, karena kelas ini berfungsi untuk koneksi ke dalam *database*, setiap kelas yang membutuhkan koneksi ke *database* akan melalui kelas ini. (5) **Kelas *Manager***, kelas ini berfungsi untuk memanager *user* yang telah terdaftar. Fungsi-fungsi di dalam kelas ini diantaranya adalah *Create*, *Read*, *Update*, dan *Delet (CRUD)*.

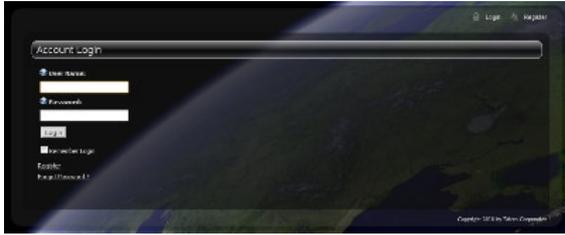
Pada tahap *GUI*, perancangan secara grafis keseluruhan aplikasi dilakukan. Dalam merancangnya digunakan prinsip-prinsip dari bidang ilmu Interaksi Manusia dan Komputer. Contohnya dalam hal pemilihan warna, besar huruf, kebiasaan pengguna dll. Tahap dimaksudkan agar waktu pengerjaan lebih singkat dan tidak ada dokumentasi yang sia-sia.

Berikut tampilan hasil rancangan:

1. Desain halaman *registrasi*



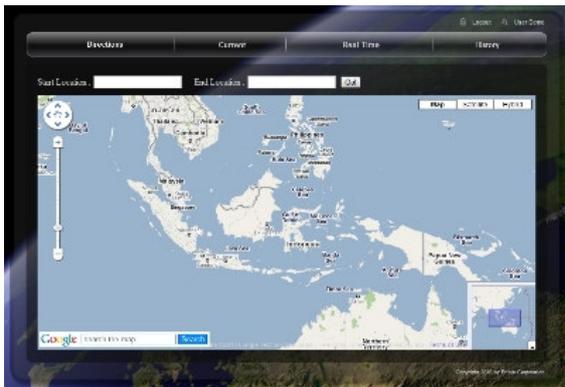
2. Desain halaman *login*



3. Desain halaman *directions*



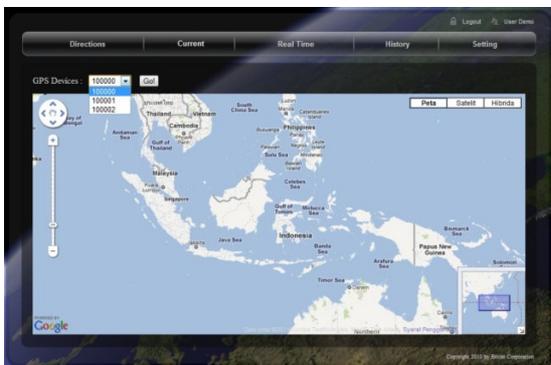
6. Desain halaman *history*



4. Desain halaman *current*



7. Desain halaman *setting*



5. Desain halaman *real time*



8. Desain halaman *tracker*



9. Desain halaman *location*



10. Desain halaman *search*



11. Desain halaman *weather*



12. Desain halaman *setting*



Coding

Penelitian kode menggunakan pendekatan teknologi *MSF (Microsoft Solution Framework)*. Jadi bisa dikatakan bahwa *MSF* akan mengisi siklus *XP (Extreme Programming)*. Pendekatan *MSF* meliputi penggunaan *.NET Framework* dan penerapan *Application Layering* yang umum dilakukan. *OOP (Object Oriented Programming)* adalah teknik *programming* yang digunakan, karena semua teknologi *.NET Framework* dikembangkan dengan menggunakan *OOP*.

Beberapa fitur *OOP* yang digunakan adalah konsep *class inheritance* dan *properties*. Dalam pengembangan aplikasi ini peneliti membuat sebuah *base class* yang akan berbagi *properties* yang umum digunakan seperti *properties ConnectionString* yang akan berbagi akses ke *ConnectionString*. *ConnectionString* adalah sebuah informasi kredensial *MSSQL server* sehingga *source code* dapat membuka koneksi ke *database* dan kemudian melakukan operasi *CRUD (Create, Retrieve, Update, Delete)*.

Selain menggunakan *.NET Framework* sebagai teknologi dasar dari aplikasi yang peneliti kembangkan, peneliti juga menggunakan *DotNetNuke* yang merupakan sebuah *WAF (Web Application Framework)*. *DotNetNuke* merupakan sebuah *framework* aplikasi yang sangat populer di kalangan pengguna teknologi *Microsoft*. *DotNetNuke* menggunakan *ASP.NET Framework* sebagai basis teknologinya. Model pengembangan aplikasi menggunakan *DotNetNuke web*

application framework adalah menggunakan konsep modular yaitu aplikasi akan dikategorikan berdasarkan modul-modul dengan fungsionalitas tertentu. Secara *programming*, *module* berisi kumpulan *web user control (.ascx)* yang akan *diload* secara dinamis. Kode program secara keseluruhan dapat dilihat pada lampiran B.

Testing

Pengetesan ini dilakukan dengan menggunakan metode *black box testing* yaitu pengetesan menggunakan metoda pengujian logika program yaitu contoh per kasus atau masalah yang diajukan. Pada tahap akhir akan diujikan pada kondisi sebenarnya yaitu dengan *implementation testing*.

Tabel Hasil Pengujian Dengan Unit Testing

No.	Modul	Prasyarat	Hasil yang diharapkan	Hasil Pengujian
1.	Registration	Akun <i>email</i>	Link konfirmasi keabsahan <i>email</i> .	OK
2.	GPS ID generation	Link konfirmasi	<i>Email</i> konfirmasi yang berisi <i>GPS ID</i> , <i>username</i> , <i>password</i> .	OK
3.	Login	<i>Username</i> dan <i>Password</i>	Masuk ke aplikasi	OK
4.	Direc	Login.	Menampil	OK

	tions	<i>Start Location</i> dan <i>End Location</i>	kan rute perjalanan dari titik A ke titik B di atas peta	
5.	Current	Login.	Google maps menampilkan informasi lokasi terkini.	OK
6.	Realtime	Login.	Google maps menampilkan pergerakan lokasi secara real time.	OK
7.	History	Login. Data <i>latitude</i> dan <i>longitude</i> untuk interval waktu tertentu	Muncul path di google map untuk data interval waktu tertentu.	OK
8.	Setting	Login	Memanager GPS ID	OK
9.	Manager	Login	Memanager user	OK
10.	Trac	GPS ID. Data	Mengirim paket data lokasi dari	OK

		<i>latitude</i> dan <i>longitude</i> untuk interval waktu tertentu	handphone ke server	
11.	Locat	GPS ID. Data <i>latitude</i> dan <i>longitude</i>	Menampilkan informasi lokasi saat ini	OK
12.	Searc	GPS ID	Menampilkan informasi lokasi posisi orang yang kita inginkan	OK
13.	Weat	GPS ID	Menampilkan Informasi cuaca berdasarkan lokasi	OK
14.	Setti	GPS ID	Menyimpan informasi GPS ID dan interval	OK

Berikut bukti dari pengujian aplikasi:

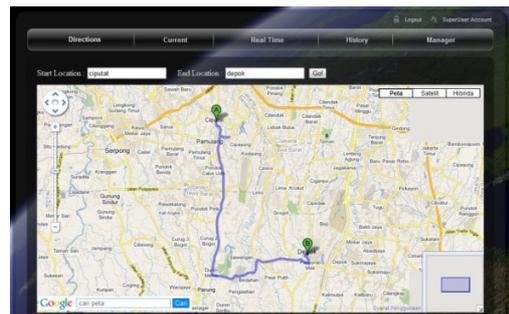
1. Registration



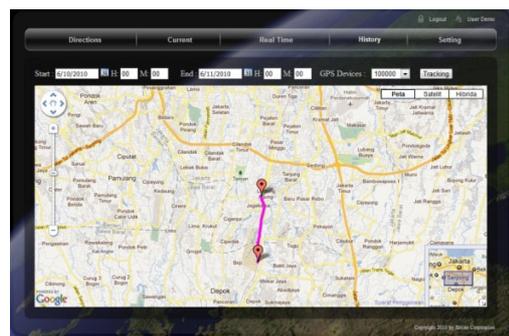
2. GPS ID Generation



3. Direction

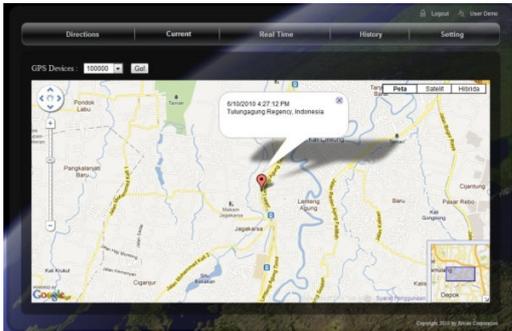


4. History Location

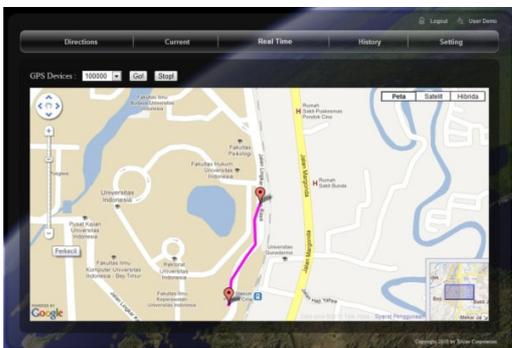


5. Current Location

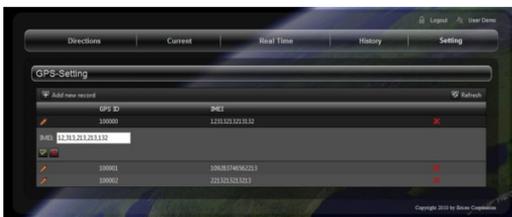
Dari hasil pengujian di atas hampir semua modul dapat berjalan sesuai apa yang diharapkan.



6. Realtime



7. Setting



8. Tracker



9. Location



10. Search



11. Weather



12. Setting



Selanjutnya pada tahap *Implementation Testing*, aplikasi akan diuji pada keadaan sebenarnya yaitu kondisi pada saat sistem sudah berjalan. *Server* yang digunakan adalah *server* dari perusahaan *shared hosting erudeye*.

Alat-alat yang digunakan pada pengujian ini diantaranya adalah 1 (satu) buah *netbook* dan *Modem 3G/HSDP* dengan Indosat sebagai *providernya*. sebuah *handphone* Nokia E71, *database* dan *Server Receiver* (*memori 4 GB*, system operasi Windows *Server 2008 R2*, *Database Server MSSQL 2008*, *.NET Framework: 2.0 / 3.5 / 4.0*), dan *Laptop merk Byon* dengan Windows 7.

System akan berjalan sempurna dan mulus jika sinyal *GPRS* tidak terblok sehingga *feed* informasi *latitude* dan *longitude* akan lancar dan untuk melihat pergerakan lokasi *realtime* secara halus maka diperlukan koneksi *internet* yang cukup yaitu minimal 123 Kbps.

Tabel Hasil Pengujian Kualitas

No.	Modul	Prasyarat	Hasil yang diharapkan	Hasil Pengujian
1.	Registration	Akun <i>email</i>	Link konfirmasi keabsahan <i>email</i> .	OK
2.	<i>GPS ID</i> generation	Link konfirmasi	<i>Email</i> konfirmasi yang berisi <i>GPS ID, username, password</i> .	OK
3.	Login	<i>Username</i> dan <i>Password</i>	Masuk ke aplikasi	OK
4.	Directions	Login. <i>Start Location</i> dan <i>End Location</i>	Menampilkan rute perjalanan dari titik A ke titik B di atas peta	OK
5.	Current	Login.	Google maps menampilkan informasi lokasi terkini.	OK
6.	Realtime	Login.	Google maps menampilkan pergerakan lokasi secara real time.	OK
7.	History	Login. Data <i>latitude</i> dan	Muncul path di google map untuk data interval waktu	OK

		<i>longitude</i> untuk interval waktu tertentu	tertentu.	
8.	Setting	Login	Memanager GPS ID	OK
9.	Manager	Login	Memanager user	OK
10.	Tracker	GPS ID. Data <i>latitude</i> dan <i>longitude</i> untuk interval waktu tertentu	Mengirim paket data lokasi dari handphone ke server	OK
11.	Location	GPS ID. Data <i>latitude</i> dan <i>longitude</i>	Menampilkan informasi lokasi saat ini	OK
12.	Search	GPS ID	Menampilkan informasi lokasi posisi orang yang kita inginkan	OK
13.	Weather	GPS ID	Menampilkan Informasi cuaca berdasarkan lokasi	OK
14.	Setting	GPS ID	Menyimpan informasi GPS ID dan interval	OK

KESIMPULAN

Berdasarkan analisis dan perancangan aplikasi *Gps Tracker*, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Aplikasi *Gps Tracker* ini dapat membantu *user* untuk mengetahui *history* perjalanan *user* dan dapat di lihat saat di butuhkan.
2. Aplikasi *Gps Tracker* ini dapat membantu *user* untuk mengetahui posisi dimana dia berada melalui satelit *GPS* dan dapat membantu *user* untuk mengetahui posisi orang lain.

DAFTAR PUSTAKA

1. Hariyanto, Bambang. 2007. *Esensi-esensi Bahasa Pemrograman JAVA*. Bandung: Informatika Bandung
2. Jubilee Enterprise. 2010. *Google Untuk Guru*. : Elex Media Komputindo.
3. Kamus Besar Bahasa Indonesia
4. Nugroho, Bunafit. 2004. *Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL*. Yogyakarta: Gava Media
5. Parkinson, B.W. 1996. *Global Positioning System: Theory and Applications, chap. 1: Introduction and Heritage of NAVSTAR, the Global Positioning System*. Washington,

D.C: American Institute of Aeronautics and Astronautics.

6. Pressman, Roger S. 2005. *Software Engineering- ixth Edition*. New York: McGraw Hill.
7. Shalahuddin Rossa A.S. Muhammad. 2008. *Pemrograman J2ME Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile*. Bandung: Informatika
8. Tanoe, Andre. 2009. *GPS Bagi pemula, dasar-dasar pemakaian sehari hari*. Jakarta
9. Wahana Komputer 2008. *Belajar Pemrograman C#*. Yogyakarta: Penerbit Andi.
10. Wahidin. 2004. *ASP untuk Orang Awam*. Palembang: Penerbit Maxikom.
11. http://id.wikipedia.org/wiki/Aplikasi_web di akses tahun 2010
12. <http://id.wikipedia.org/wiki/Handphone> di akses tahun 2010
13. http://id.wikipedia.org/wiki/Sql_server di akses tahun 2010
14. <http://id.wikipedia.org/wiki/Web> di akses tahun 2010
15. <http://kiosbukugema.wordpress.com/page/91/> di akses tahun 2010
16. <http://www.clearleadinc.com/> di akses tahun 2010
17. <http://www.websmithing.com> di akses tahun 2010