

BAB III

METODOLOGI PENELITIAN

Dalam penelitian ini penulis menggunakan metode eksplorasi dan studi literatur yaitu mengkaji teori serta konsep yang berkaitan dengan permasalahan. Selain itu penulis juga menggunakan metode observasi dengan melakukan pengamatan terhadap implementasi *Geochat* menggunakan teknologi *Google App Engine* dengan *Python Programming Language*. Metode penelitian ini meliputi alat dan bahan penelitian, metode penelitian dan juga desain penelitian.

2.1 Alat dan Bahan Penelitian

2.2.1 Alat Penelitian

Alat penelitian yang digunakan dalam skripsi ini adalah sebuah komputer yang memiliki spesifikasi cukup untuk melakukan pemrograman sebagai implementasi dari pembangunan aplikasi *Geochat* menggunakan teknologi *Google App Engine* dengan *Python Programming Language* sebagai media *share* penyebaran informasi lalu lintas. Alat penelitian yang digunakan yaitu :

1. Sistem komputer dengan spesifikasi :
 - a. Processor Intel(R) Pentium 4 CPU 1.80 GHz
 - b. Memory 768 MB RAM
 - c. Harddisk Quantum Fireball 40 GB
 - d. Monitor dengan resolusi 1024x768 pixel
 - e. Mouse dan keyboard

2. Sistem operasi Microsoft Windows XP Professional SP2
3. Software sebagai *support* program
 - a. *Python 2.6.4*
 - b. *IDLE Python*
 - c. *Google App Engine Launcher*
 - d. *App Engine SDK Python*
 - e. *Mozilla Firefox 3.6.3*

2.2.2 Bahan Penelitian

Bahan penelitian dalam skripsi ini adalah materi dan juga konsep yang menunjang pembangunan aplikasi. Bahan penelitian yang digunakan yaitu :

1. *Geochat*
2. *Google App Engine*
3. *Python Programming Language*

2.2 Metode Penelitian

Adapun metode dan langkah-langkah penelitian yang digunakan adalah sebagai berikut :

1. *Requirement*

Pada tahap *requirement*, penulis mengumpulkan informasi-informasi serta permasalahan yang terkait dengan penelitian ini kemudian merumuskannya menjadi suatu batasan permasalahan yang menjadi topik penelitian ini.

2. *Analisis*

Pada tahap analisis, penulis melakukan peninjauan ulang secara lebih mendalam terhadap objek yang menjadi penelitian. Sehingga diharapkan proses ini dapat menjadi acuan terhadap proses-proses selanjutnya.

3. *Design*

Pada tahap *design*, penulis merancang perangkat lunak berupa aplikasi yang menjadi hasil penelitian ini. Diharapkan dengan adanya tahapan *design* ini, permasalahan akan menjadi lebih jelas untuk diselesaikan. Tahapan *design* ini mengikuti prosedur dari proses analisis.

4. *Implementation*

Pada tahap implementasi, penulis menerapkan penelitian ini pada sebuah aplikasi yang dirancang untuk memecahkan permasalahan. Diharapkan aplikasi yang dihasilkan nantinya dapat menjadi media *share* penyebaran informasi lalu lintas yang bermanfaat bagi para pengemudi.

5. *Testing*

Pada tahap *testing*, hasil penelitian ini akan diujicobakan sehingga hasilnya akan terlihat apakah aplikasi ini dapat berjalan dengan baik atau tidak. Tahapan *testing* ini dilakukan berulang-ulang untuk mengetahui apakah aplikasi ini memiliki tingkat kesalahan kecil atau besar.

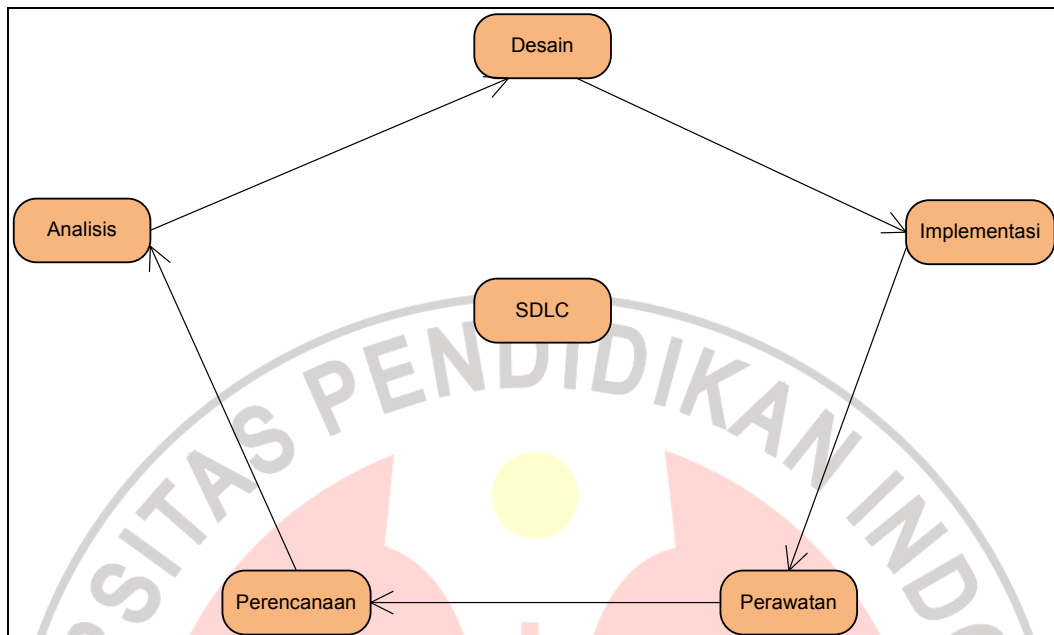
6. *Documentation*

Pada tahap dokumentasi, hasil penelitian akan dilaporkan sehingga dapat bermanfaat bagi penelitian lain yang berhubungan dengan penelitian yang dilakukan penulis.

2.3 Desain Penelitian

Penelitian dimulai dari pengumpulan *history* informasi lalu lintas di Jakarta. *History* informasi lalu lintas yang dimaksud adalah informasi ruas jalan yang menjadi titik kemacetan. Selain itu, koordinat letak kemacetan menjadi perhatian dalam penelitian ini. Hal ini dibutuhkan untuk menampilkan peta sesuai lokasi titik kemacetan yang terjadi, sehingga diharapkan letaknya pun tepat saat ditampilkan di *maps*. Untuk pengembangan selanjutnya, cakupan aplikasi dapat berkembang menjadi lebih luas dari segi lokasi kemacetan yang dapat diterapkan di kota lain.

SDLC (SOFTWARE DEVELOPMENT LIFECYCLE)



Gambar 2.1 Siklus Pengembangan Software

Pengembangan perangkat lunak dapat dianggap sebagai lingkaran pemecahan masalah. Untuk menyelesaikan masalah besar, dipecah menjadi kecil terus-menerus sampai ketinggian yang paling kecil, kemudian diselesaikan. Daur hidup perangkat lunak adalah model proses untuk rekayasa perangkat lunak yang dipilih berdasarkan sifat aplikasi dan proyeknya, metode dan *tools* yang digunakan, serta kontrol dan *deliverable* yang diinginkan.

Dalam perancangan perangkat lunak ini penulis menggunakan model *Rapid Application Development (RAD)*. Model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek. Menekankan perkembangan komponen program yang bisa dipakai ulang (*reusability*) sehingga mendasari konsep *Object-Oriented*.

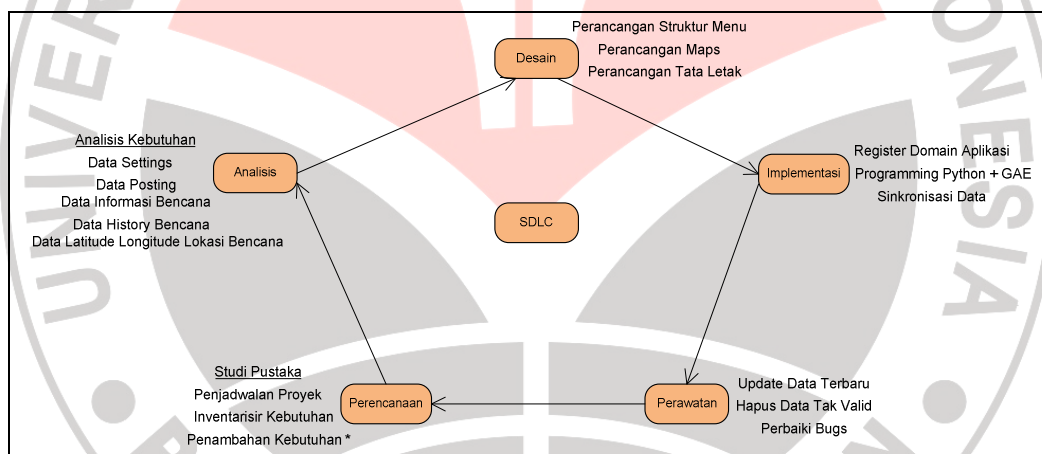
Fase pendekatan *Rapid Application Development* :

1. *Bussines Modeling*
2. *Data Modeling*
3. *Proses Modeling*
4. *Application Generation* : RAD lebih banyak memproses kerja untuk memakai lagi komponen program atau menciptakan komponen yang bisa dipakai lagi.
5. *Testing and Turn Over* : Banyak komponen program yang telah diuji sebelumnya sehingga mengurangi keseluruhan waktu pengujian. Tapi komponen baru harus diuji dan semua *interface* harus dilatih secara penuh.

Model RAD adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik *incremental* (bertingkat). RAD menekankan pada siklus pembangunan pendek/singkat/cepat. Waktu yang singkat adalah batasan yang penting untuk model ini. Model RAD mengadopsi model *waterfall* dan pembangunan dalam waktu singkat yang dicapai dengan menerapkan :

1. *Component Based Construction* (pemrograman berbasis komponen).
2. Penekanan pada penggunaan ulang (*reuse*) komponen perangkat lunak yang telah ada.
3. Pembangkitan kode program otomatis/semi otomatis.
4. *Multiple team* (banyak tim), tiap tim menyelesaikan satu tugas yang selevel tapi tidak sama. Banyaknya tim tergantung dari area dan kompleksitasnya sistem yang dibangun.

Jika kebutuhan yang diinginkan pada tahap analisa telah lengkap dan jelas, maka waktu yang dibutuhkan untuk menyelesaikan secara lengkap perangkat lunak yang dibuat adalah berkisar 60 sampai 90 hari. Model RAD ini sebenarnya hampir sama dengan model *waterfall*, bedanya siklus pengembangan yang ditempuh model ini sangat pendek dengan penerapan teknik yang cepat. Sistem dibagi-bagi menjadi beberapa modul dan dikerjakan beberapa tim dalam waktu yang hampir bersamaan dalam waktu yang sudah ditentukan. Model ini melibatkan banyak tim, dan setiap tim mengerjakan tugas yang selevel, namun berbeda. Sesuai dengan pembagian modul sistem.

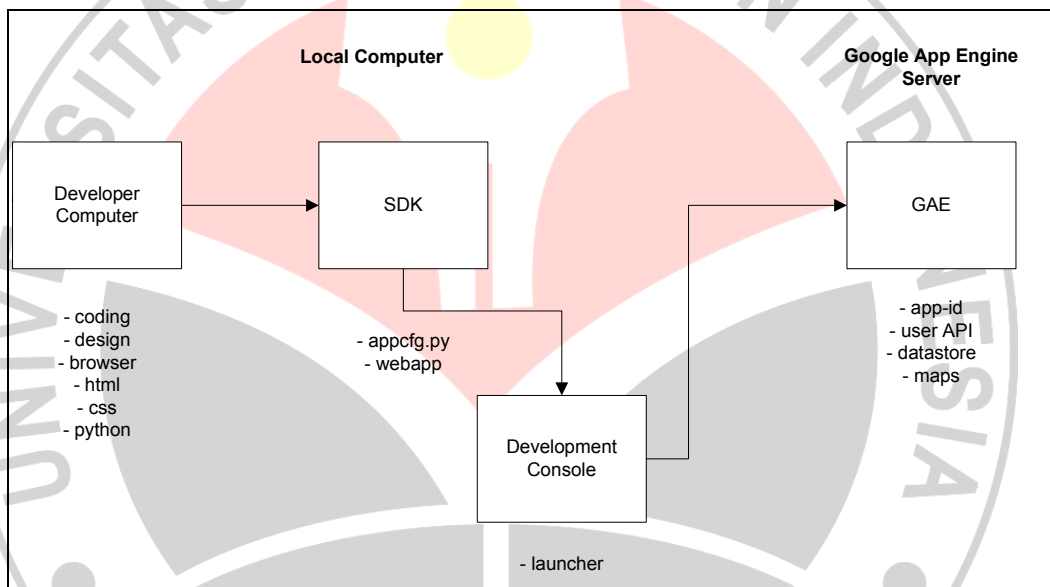


Gambar 2.2 RAD Detail Model Proses

3.4 Implementasi

3.4.1 Prosedur Pengerjaan Penelitian

Penelitian ini membutuhkan beberapa tahap pengerjaan mulai dari pengumpulan data informasi lalu lintas di Jakarta, *eksplorasi* konsep *Geochat*, penggunaan teknologi App Engine, pemahaman bahasa pemrograman *Python* sampai merancang menjadi sebuah aplikasi yang terintegrasi.



Gambar 3.3 Model Rancangan Program Keseluruhan

Adapun urutan pengerjaan yang harus dilalui adalah sebagai berikut :

1. Pengumpulan data yang berhubungan dengan data informasi lalu lintas dan *history* kemacetan atau kecelakaan yang terjadi di Jakarta.

2. Riset data *latitude* dan *longitude* suatu lokasi. Hal ini diperlukan untuk ketepatan lokasi pada informasi yang dimaksud dan juga *settings default* lokasi *user*.
3. Membuat rancangan dengan *SDK Python* yang telah dikuasai. Dalam langkah ini mungkin terlibat beberapa *tool* baik *software standalone* maupun *library* yang bertujuan agar tingkat efektifitas perancangan menjadi lebih baik.
4. Pembangunan *interface* aplikasi di komputer lokal yang disesuaikan dengan fitur-fitur yang akan ada dalam aplikasi ini yaitu :
 - a. Informasi Lalu Lintas
 - b. *Posting* Informasi Lalu Lintas
 - c. Cari Lokasi
 - d. *Settings Default* Lokasi
 - e. *History* Informasi Lalu Lintas
 - f. Pilih *History* Lalu Lintas
 - g. *View* Tabel *History* Lalu Lintas
 - h. Rekap Informasi Lalu Lintas Jakarta
 - i. Lihat Peta
 - j. *Help*
5. Perancangan dan pembangunan *datastore* yang disesuaikan dengan kebutuhan fitur. Penulis membuat 2 kelas *datastore* yang dibutuhkan yaitu *class posting* untuk menyimpan hasil *posting-an*

user dan *class settings* untuk menyimpan lokasi *settings default user*.

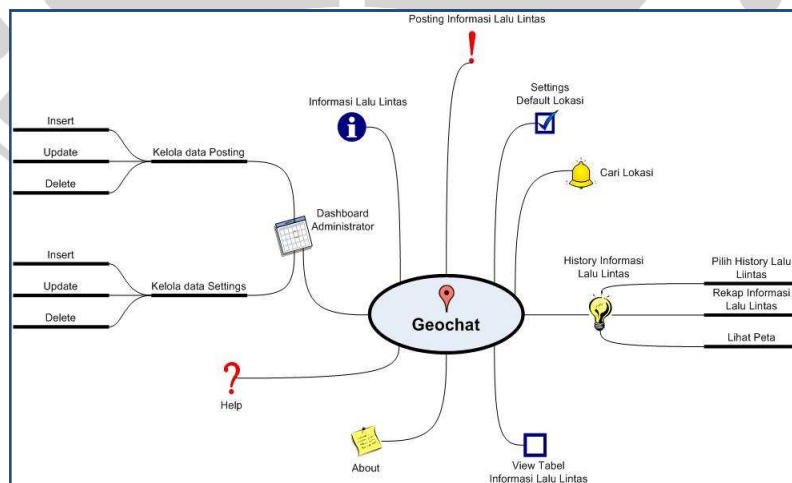
6. Proses *deploy* aplikasi ke *server* Google App Engine agar dapat diakses secara *online* oleh *user*.

3. 4.2 Pengumpulan Data

Pengumpulan data yang dimaksud meliputi data informasi mengenai informasi lalu lintas berupa informasi kemacetan, lokasi kemacetan, data koordinat *latitude* dan *longitude*, waktu kejadian dan informasi terkini dari informasi lalu lintas tersebut. Selain itu, penulis juga mencari artikel-artikel yang berkaitan dengan informasi lalu lintas tersebut kemudian menghubungkannya dengan aplikasi.

2.4.3 Struktur Program

Berikut ini penulis sertakan bagan struktur keseluruhan dari aplikasi *Geochat* ini :



Gambar 2.4 Bagan Struktur Program