

BAB II

LANDASAN TEORI

2.1 Aplikasi

Aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan. Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi software yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

1. Aplikasi software spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

2.2 *Global Positioning System (GPS)*

GPS adalah kependekan dari Global Positioning System, merupakan sistem navigasi berbasis satelit yang dikembangkan oleh Departemen Pertahanan Amerika yang didukung oleh 27 jaringan satelit. GPS terdiri dari 3 segmen: Segmen angkasa, Segmen kontrol atau pengendali dan Segmen pengguna. Dimana segmen angkasa terdiri dari 24 satelit yang beroperasi dalam 6 orbit pada ketinggian 20.200 km dan inklinasi 55 derajat dengan periode 12 jam (satelit akan kembali ke titik yang sama dalam 12 jam).

Satelit tersebut memutari orbitnya sehingga minimal ada 6 satelit yang dapat dipantau pada titik manapun di bumi ini. Satelit tersebut mengirimkan posisi dan waktu kepada pengguna seluruh dunia. Setiap satelit mentransmisikan dua sinyal

yaitu L1 (1575.42 MHz) dan L2 (1227.60 MHz). Sinyal L1 dimodulasikan dengan dua sinyal pseudo-random yaitu kode P (Protected) dan kode C/A (coarse/aquisition). Sinyal L2 hanya membawa kode P. Setiap satelit mentransmisikan kode yang unik sehingga penerima (perangkat GPS) dapat mengidentifikasi sinyal dari setiap satelit.

Perangkat GPS menerima sinyal yang ditransmisikan oleh satelit GPS. Dalam menentukan posisi, kita membutuhkan paling sedikit 3 satelit untuk penentuan posisi 2 dimensi (lintang dan bujur) dan 4 satelit untuk penentuan posisi 3 dimensi (lintang, bujur, dan ketinggian).

2.3 Location Based System (LBS)

Location based services adalah layanan berbasis lokasi atau istilah umum yang sering digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang pengguna gunakan. Layanan ini menggunakan teknologi *global positioning service* (GPS) dan *cell-based location* dari Google.

Selain itu, LBS tersebut terdiri dari beberapa komponen di antaranya *mobile devices*, *communication network*, *position component*, dan *service and content provider*. *Mobile devices* merupakan komponen yang sangat penting. Piranti *mobile* tersebut diantaranya adalah *smartphone*, PDA, dan lainnya yang dapat berfungsi sebagai alat navigasi atau seperti halnya alat navigasi berbasis GPS.

Komponen *communication network* ini berupa jaringan telekomunikasi bergerak yang memindahkan data pengguna dari perangkat ke penyedia layanan. *Position component* yang dimaksud adalah posisi pengguna harus

ditentukan. Posisi ini dapat didapatkan dengan jaringan telekomunikasi atau dengan GPS. Sedangkan *service and content provider* adalah penyedia layanan yang menyediakan layanan berbeda ke pengguna seperti pencarian rute, kalkulasi posisi, dan lainnya.

2.4 Google Maps API

Google maps API adalah fungsi fungsi pemrograman yang disediakan oleh *Google maps* agar *Google maps* bisa di integrasikan kedalam *Web* atau aplikasi yang sedang buat. Contoh sederhanya misalkan anda ingin membuat Sistem Informasi Geografis, dengan memanfaatkan *Google Maps* API bisa membuat GIS tanpa perlu memikirkan peta, hanya dengan menggunakan *Google maps* dan memanggil fungsi-fungsi yang dibutuhkan seperti menampilkan peta, menempatkan marker dan sebagainya.

2.5 Perangkat Lunak Pendukung

2.5.1 C++

Bahasa Pemrograman C++ adalah bahasa Pemrograman Komputer Tingkat Tinggi (*High Level Language*), tapi C++ juga dimungkinkan untuk menulis Bahasa Pemrograman Tingkat Rendah (*Low Level Language*) di dalam pengkodean.

C++ adalah peluasan dan penyempurnaan dari bahasa pemrograman sebelumnya yaitu bahasa C, oleh Bjarne Stroustrup pada tahun 1980. Awal C++ mempunyai nama yaitu “*C with Classes*” dan berganti nama menjadi C++ pada tahun 1983. Bjarne Stroustrup membuat bahasa pemrograman C++ dengan

tambahan fasilitas, yang sangat berguna pada tahun itu sampai sekarang, yaitu bahasa pemrograman yang mendukung OOP (*Object Oriented Programming*).

2.5.2 JavaScript

JavaScript merupakan modifikasi dari bahasa *c++* dengan pola penulisan yang lebih sederhana. Interpreter bahasa ini sudah disediakan ASP ataupun internet *explorer*. Kelebihan *JavaScript* adalah berinteraksi dengan HTML, ini membolehkan pembuat web untuk memasukkan web mereka dengan kandungan-kandungan yang dinamik, menukar warna *background*, menukar banner, efek mouse, menu interaktif dan sebagainya.

2.5.3 Kotlin

Kotlin adalah sebuah bahasa pemrograman dengan pengetikan statis yang berjalan pada Mesin Virtual Java ataupun menggunakan kompiler LLVM yang dapat pula dikompilasikan kedalam bentuk kode sumber JavaScript. Pengembang utamanya berasal dari tim programer dari JetBrains yang bermarkas di Rusia. Meskipun sintaksisnya tidak kompatibel dengan bahasa Java, Kotlin didesain untuk dapat bekerja sama dengan kode bahasa Java dan bergantung kepada kode bahasa Java dari Kelas Pustaka Java yang ada, seperti berbagai framework Java yang ada.

Tim Pengembang memutuskan menamakannya Kotlin dengan mengambil nama dari sebuah pulau di Rusia, sebagaimana Java yang mengambil nama dari pulau Jawa di Indonesia. Setelah Google mengumumkan bahwa Kotlin menjadi

bahasa kelas satu bagi Android, maka bersama Java dan C++, Kotlin menjadi bahasa resmi untuk pengembangan aplikasi-aplikasi Android.

2.5.4 Android Studio

Android Studio merupakan Integrated Development Environment (IDE) resmi untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Android berubah menjadi platform yang begitu cepat dalam melakukan inovasi. Hal ini tidak lepas dari pengembangan utama dibelakangnya, yaitu *Google*. *Google* yang mengakuisisi *Android* dan kemudian membuatkan sebuah *platform*.

Platform android terdiri dari Sistem Operasi berbasis *Linux*, sebuah GUI (*Graphic User Interface*), sebuah *web browser* dan Aplikasi Studio *End-User* yang dapat di *download* dan juga para pengembang bisa dengan leluasa berkarya serta menciptakan aplikasi yang terbaik dan terbuka untuk digunakan oleh berbagai macam perangkat.

2.6 Metode Pengembangan Sistem

Metode adalah suatu cara atau teknik yang sistematis untuk mengerjakan sesuatu. Metode penelitian adalah prosedur-prosedur, konsep-konsep pekerjaan, aturan-aturan yang akan digunakan oleh suatu ilmu pengetahuan.

2.6.1 *eXtreme Programming*

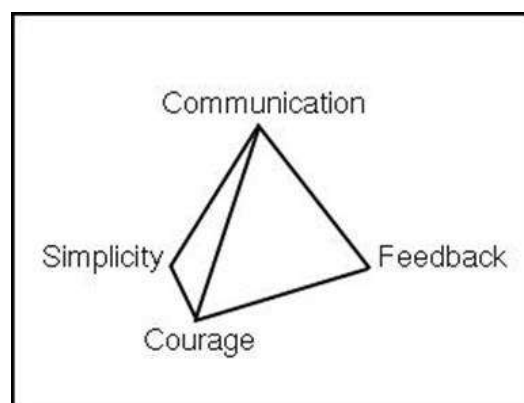
Extreme Programming merupakan suatu model yang tergolong dalam pendekatan *agile* yang diusulkan oleh *Kent Back*. Menurut penjelasannya, definisikan *Extreme Programming* adalah “*Extreme Programming (XP) is a lightweight, efficient, low risk, flexible, predictable, scientific, and fun way to*

develop software”. Model ini cenderung menggunakan pendekatan *object oriented*.

Secara umum *Extreme Programming* (XP) dapat dijabarkan sebagai sebuah pendekatan pengembangan perangkat lunak yang mencoba meningkatkan efisiensi dan fleksibilitas dari sebuah proyek pengembangan perangkat lunak dengan mengkombinasikan berbagai ide simpel atau sederhana tanpa mengurangi kualitas *software* yang akan dibangun.

Pembangunan sistem pada metode *Extreme Programming* (XP) dilakukan secara cepat. *Extreme Programming* (XP) tepat digunakan saat kondisi keperluan berubah dengan cepat, resiko tinggi dan ada proyek dengan tantangan yang baru, mampu mengotomatiskan tes, dan ada peran serta pelanggan atau *user* secara langsung.

Extreme Programming (XP) dikembangkan oleh *Beck, Cunningham*, dan *Jeffries*. Ini merupakan *lightweight* dan disiplin pengembangan perangkat lunak berdasarkan empat *core value*.



Gambar 2.1 *Core Value Extreme Programming* (XP) (Santoso, Rais. 2015)

Extreme Programming (XP) dikembangkan oleh *Beck, Cunningham*, dan *Jeffries* model dibagi dalam 4 *core value* sebagai berikut:

1. *Communication*

Extreme Programming (XP) mengfokuskan pada hubungan komunikasi yang baik antara tim klien, anggota tim, dan manajer proyek komunikasi dalam *Extreme Programming* dibangun dengan melakukan pemrograman berpasang. Klien harus dilibatkan dalam proses pengembangan perangkat lunaknya dengan tujuannya untuk memberikan pandangan pengembang sesuai dengan pandangan pengguna sistem yang dibangun.

2. *Simplicity*

Extreme Programming (XP) melakukan semua dengan sederhana dan praktis tanpa mengurangi fungsi utamanya.

3. *Feedback*

Selalu mengevaluasi terhadap perangkat lunak yang sedang dikerjakan, segala informasi harus dikumpulkan setiap *interval* waktu yang konsisten dan diskusikan kesalahan-kesalahan yang muncul selama proses pengembangan. Umpan balik tersebut berfungsi sebagai indikator kemajuan proyek dan menginformasikan pemimpin proyek apabila perlu dibuat.

4. *Courage*

Programmer XP didorong untuk berani beresperimentasi dan menulis ulang kode jika tidak puas dengan kode yang sudah ada atau desain. Hal ini membantu mempertahankan moral serta integritas para pengembang proyek dan dapat mendukung lebih lanjut komunikasi dengan anggota proyek lainnya.

2.7 Metode Pengujian Black Box

Black-box testing adalah metode pengujian perangkat lunak yang melakukan pengetesan fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja (pengujian *white-box*). Pengetahuan khusus dari kode aplikasi atau struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, apa yang seharusnya aplikasi lakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih input yang valid dan tidak valid dan menentukan *output* yang benar.

Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan masukan. Ini biasanya terdiri dari kebanyakan jika tidak semua pengujian pada tingkat yang lebih tinggi, tetapi juga bisa mendominasi unit *testing* juga. Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari *software*. Karena itu ujicoba *blackbox* memungkinkan pengembang *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *blackbox* bukan merupakan alternatif dari ujicoba *whitebox*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *whitebox*.

Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan performa
5. Kesalahan inisialisasi dan terminasi

2.8 State of The Art

Penelitian terkait dari Rena Ariyanti, dkk yang berjudul “Pemanfaatan Google Maps Api Pada Sistem Informasi Geografis Direktori Perguruan Tinggi Di Kota Bengkulu” dengan permasalahan tidak semua orang awam tahu tentang perguruan tinggi di Kota Bengkulu terutama letak atau lokasi perguruan tinggi. Penelitian dikaji tentang merancang sebuah sistem informasi geografis berbasis *web* yang berfungsi untuk memberikan kemudahan dalam mencari informasi letak perguruan tinggi di Kota Bengkulu (Ariyanti, Khairil, dkk, 2015).

Penelitian terkait dari Faya Mahdia dan Fiftin Noviyanto yang berjudul “Pemanfaatan *Google Maps API* Untuk Pembangunan Sistem Informasi Manajemen Bantuan Logistik Pasca Bencana Alam Berbasis *Mobile Web* (Studi Kasus: Badan Penanggulangan Bencana Daerah Kota Yogyakarta)” dengan permasalahan proses pendistribusian bantuan ke posko-posko bencana alam dari pemerintah, instansi dan masyarakat sekitar seringkali kurang merata, sedangkan masih banyak korban lain yang belum mendapatkan bantuan. Penelitian dikaji tentang merancang sebuah sistem berbasis *web mobile* agar sistem dapat diakses melalui perangkat *mobile smartphone* yang telah dilengkapi dengan GPS (*Global*

Positioning System) sehingga dapat mengetahui koordinat letak posko, yaitu berupa data *latitude* dan *longitude* (Mahdia, Noviyanto. 2013).

Penelitian terkait dari Danang Adi Sumarto dan Tedi Setiadi yang berjudul “Sistem Informasi Geografis Monitoring KKN Posdaya Universitas Ahmad Dahlan Berbasis *Google Maps API*” dengan permasalahan sulitnya dalam melakukan monitoring secara berkelanjutan dalam kegiatan posdaya yang ada. Sehingga banyak posdaya tidak terpantau yang menyebabkan banyak posdaya tidak berjalan. Penelitian dikaji tentang merancang sebuah Sistem Informasi Geografis berbasis *web* dengan *API Google Maps* yang dapat menampilkan pemetaan Posdaya berbasis *web* di setiap desa atau kelurahan dimana diselenggarakan posdaya (Sumarto, Setiadi. 2014).

Penelitian yang akan dibuat dengan judul “Pemanfaatan *Google Maps API* untuk Pemetaan Lokasi Wisata di Kabupaten Garut Berbasis Android” akan ditambahkan hal baru sebagai penyempurna dari penelitian-penelitian yang telah dibuat sebelumnya, yaitu penambahan dalam segi informasi lokasi tempat wisata yang akan ditampilkan lebih detail dengan menambah pengetahuan sejarah dan latar belakang dari lokasi wisata serta dalam *marker* akan diberi perbedaan untuk tempat wisata favorit untuk dikunjungi.