

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Sistem operasi untuk aplikasi bergerak yang mengalami perkembangan yang cukup pesat yaitu Android. Android adalah sistem operasi berbasis Linux dan bersifat *open source*. Perubahan beralihnya keseharian pengguna yang sebelumnya hanya sebatas telepon, pesan singkat, pesan elektronik, *browsing* menjadi pengguna *device* untuk menunjang kegiatan sehari-hari misalnya gaya hidup, hiburan, dan kegiatan lainnya termasuk dalam hal berbagi informasi melalui media sosial seperti Facebook, Twitter, Line, Whatsapp dan yang lainnya merupakan contoh *website* dan aplikasi sosial terbesar saat ini.

Banyak informasi-informasi yang bisa didapat dari status orang lain, grup atau halaman yang sengaja dibuat untuk berbagi informasi. Bahkan banyak informasi yang mencari tentang kehilangan barang atau mungkin menemukan barang yang bukan miliknya. Dari surat-surat berharga, barang elektronik maupun dompet. Terbatasnya teman serta tidak terfokusnya pencarian tentang berita kehilangan maupun berita penemuan barang sehingga pencarian barang tersebut kurang maksimal. Sebagai contoh ketika seseorang menemukan dompet berisi surat-surat berharga atau uang maka mereka membuat status tentang penemuan tersebut, namun karena terbatasnya teman sehingga orang lain yang bukan merupakan temannya tidak melihat informasi tersebut. Atau ketika seseorang kehilangan selain melapor ke pihak yang berwajib tak jarang menulis status tentang kehilangan berharap ada orang lain yang menemukan atau melihatnya. Hal seperti itu bisa jadi barang yang hilang bisa ditemukan tetapi sangat sulit bahkan membutuhkan waktu yang lama.

Oleh karena itu, dibutuhkan sebuah media khusus untuk informasi kehilangan atau penemuan barang yang bertujuan untuk membantu mencari dan memudahkan menemukan kembali barang yang hilang atau mengembalikan barang yang ditemukan.

1.2 Rumusan Masalah

Rumusan masalah proyek akhir ini adalah:

1. Bagaimana anggota dapat melihat *list thread* kehilangan dan *list thread* penemuan?
2. Bagaimana anggota dapat mempublikasikan barang kehilangan?
3. Bagaimana sesama anggota dapat memberikan informasi jika menemukan barang yang hilang?

1.3 Tujuan

Tujuan proyek akhir ini adalah:

1. Membangun aplikasi *mobile* berbasis Android yang dapat memperlihatkan *list thread* kehilangan dan *list thread* penemuan ke seluruh anggota.
2. Membangun aplikasi *mobile* berbasis Android yang dapat menjadi sarana untuk mengumumkan kehilangan dan penemuan barang bagi anggota yang sedang kehilangan barang.

1.4 Batasan Masalah

Batasan masalah dari proyek akhir ini adalah:

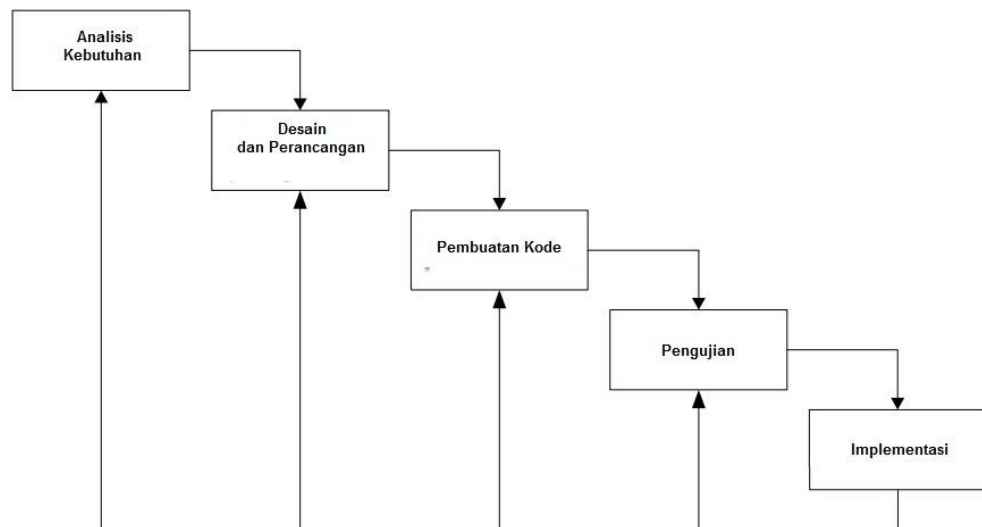
1. Aplikasi ini hanya dapat dijalankan pada perangkat bergerak (*mobile*) yang menggunakan sistem operasi Android dengan minimal versi 2.2 (Froyo).
2. Tidak membahas peta barang hilang.
3. Dalam aplikasi ini, terdapat mekanisme yang dapat digunakan untuk mengubah status barang yang telah kembali ke pemiliknya. Perubahan status info barang dilakukan oleh pengguna.

1.5 Definisi Operasional

Aplikasi *tell-me* aplikasi pencatatan kehilangan dan penemuan barang berbasis Android ini merupakan aplikasi yang dirancang untuk membantu proses pencarian barang dan penemuan barang dalam lingkup kampus. Aplikasi ini dibangun dengan menggunakan bahasa pemrograman Java dan diimplementasikan pada *smartphone* berbasis Android.

1.6 Metode Pengerjaan

Metode yang digunakan dalam penyelesaian masalah dalam penulisan proyek akhir ini adalah: metode *Waterfall* pada tahapan *Software Development Life Cycle* (SDLC) meliputi: analisis, perancangan, pembuatan kode, implementasi dan pengujian. *Waterfall* adalah model pengembangan klasik yang sistematis dimana setiap hasil dari suatu tahap akan menjadi masukan awal untuk tahap selanjutnya[1].



Gambar 1- 1 Model *Waterfall* SDLC

A. Analisis Kebutuhan

Dalam tahapan ini, dilakukan pengumpulan data kebutuhan secara lengkap untuk dianalisis dan didefinisikan kebutuhan untuk membangun aplikasi pencatat kehilangan dan penemuan barang berbasis Android. Pengumpulan data dilakukan dengan pengamatan langsung. Tahap ini harus dikerjakan secara lengkap agar mendapatkan informasi pengguna yang kemudian akan digunakan untuk tahap selanjutnya.

B. Desain dan Perancangan Sistem

Setelah mendapatkan informasi dan data-data yang dibutuhkan tahap selanjutnya adalah melakukan perancangan sistem yang meliputi desain antar muka pengguna.

C. Pembuatan Aplikasi/*Coding*

Pada tahap ini merupakan tahap pembuatan kode aplikasi. Pembuatan kode menggunakan *tools Injelij* dan penulisan program ini di-*convert* dalam bentuk APK dan di jalankan pada *divice* berbasis Android.

D. Pengujian

Setelah program selesai baru kita melakukan pengujian program yang mana pada tahap ini menggunakan metode *Black Box Testing*, kita memastikan tidak ditemukannya lagi *error* atau kesalahan pada saat aplikasi dijalankan dan fungsionalitas berjalan dengan baik dan benar.

E. Implementasi

Apabila semua tahap telah dilaksanakan dan aplikasi berjalan tanpa ada *error* maka aplikasi tersebut dapat digunakan oleh pengguna Android.

BAB 2

TINJAUAN PUSTAKA

2.1 Barang

Barang adalah benda-benda yang berwujud, yang digunakan masyarakat untuk memenuhi kebutuhannya atau untuk menghasilkan benda lain yang akan memenuhi kebutuhan masyarakat[2]. Sehingga barang begitu berguna pada pemilik agar dapat melakukan aktivitasnya.

2.2 Aplikasi

Aplikasi adalah komponen yang berguna melakukan pengolahan data maupun kegiatan-kegiatan seperti dokumen atau pengolahan data yang dapat berinteraksi langsung dengan *user*. Aplikasi berjalan diatas sistem operasi, sehingga agar bisa diaktifkan, kita perlu melakukan instalasi sistem operasi terlebih dahulu[3].

2.3 Website

Website adalah salah satu sumber data internet yang berkembang pesat yang didistribusikan melalui pendekatan *hyperlink*, yang memungkinkan teks, gambar ataupun obyek lain menjadi acuan untuk membuka halaman-halaman *web* yang lain [4].

2.4 Android

Android adalah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Saat ini sudah banyak *vendor-vendor smartphone* yang memproduksi *smartphone* berbasis *android* seperti Samsung, Sony dan LG. Tidak hanya menjadi sistem operasi di *smartphone*, saat ini Android menjadi pesaing utama dari Apple pada sistem operasi *Tablet PC*.

Android adalah sistem operasi berbasis Linux yang dirancang terutama untuk perangkat *touchscreen mobile* seperti *smartphone* dan komputer *tablet*. Awalnya

dikembangkan oleh Android Inc. Kemudian diakuisisi oleh Google pada pertengahan tahun 2005. Antarmuka pengguna Android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata, seperti menggesek, mengetuk, mencubit dan membalikkan cubitan untuk manipulasi obyek di layar. Android adalah sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah *Lisensi Apache* [5].

2.5 *Unified Modelling Language (UML)*

Unified Modelling Language adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi obyek. Pemodelan (*modelling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga mudah dipelajari atau dipahami. Adapun tujuan pemodelan yaitu sebagai sarana analisis, pemahaman, visualisasi dan komunikasi antar anggota tim pengembang serta sebagai saran dokumentasi. UML memiliki banyak jenis diagram. UML juga disebut sebagai bahasa standar untuk penulisan *blueprint software* [6].

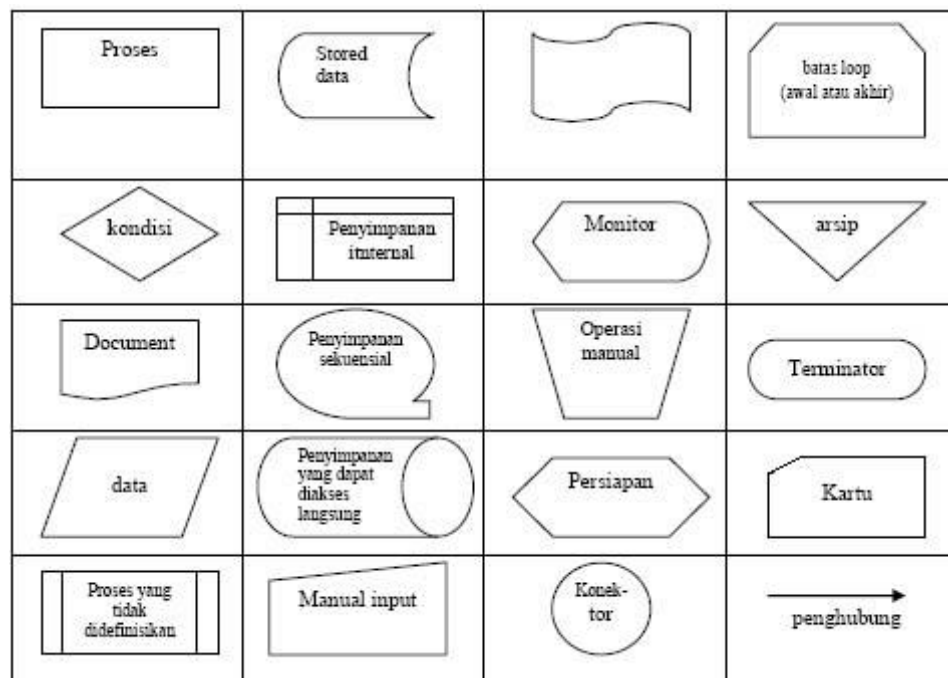
2.6 *My Structure Query Language (MySQL)*

MySQL merupakan perangkat lunak yang juga bersifat *open source*. Sesuai namanya, bahasa standar yang digunakan adalah SQL (*Structure Query Language*) dan merupakan bahasa standar untuk pengolahan *database*. Pada umumnya, perintah yang paling sering digunakan dalam MySQL adalah perintah SELECT yang berfungsi untuk menampilkan data dari *database*. Selain itu, SQL juga menyediakan perintah untuk membuat *database*, *field*, ataupun *index* untuk menambah dan menghapus data [7].

2.7 *Flowmap*

Flowmap merupakan diagram yang menggambarkan aliran dokumen pada suatu prosedur kerja di organisasi. Penggambaran biasanya diawali dengan mengamati dokumen apa yang menjadi media atau informasi. Selanjutnya ditelusuri bagaimana dokumen tersebut terbentuk, ke bagian atau entitas mana dokumen tersebut mengalir, perubahan apa yang terjadi pada dokumen tersebut, proses apa yang terjadi terhadap dokumen tersebut, dan seterusnya. Fungsi *flowmap* adalah untuk mendefinisikan hubungan antara bagian (pelaku proses), proses (manual/berbasis

komputer) dan aliran data (dalam bentuk dokumen keluaran dan masukan). Pada gambar 2 berikut terlampir beberapa simbol *flowmap* yang sering digunakan [8].

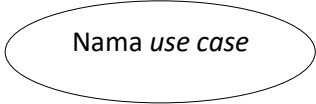


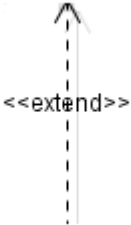



Gambar 2 - 1 Gambar Simbol *Flowmap*

2.8 Use Case Diagram

Use Case Diagram digunakan untuk memodelkan fungsionalitas-fungsionalitas sistem atau perangkat lunak dilihat dari pengguna yang ada di luar sistem (yang sering disebut aktor). *Use case* pada dasarnya unit fungsionalitas koheren yang diekspresikan sebagai transaksi-transaksi yang terjadi antara actor dan *system*. Pada tabel berikut terlampir beberapa simbol yang digunakan pada *Use Case* [9].

Tabel 2- 1 Tabel Use Case

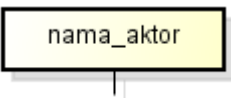

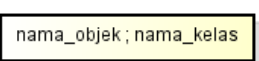
Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i>. Biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p>
<p>Aktor/ <i>Actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang dibuat di luar sistem informasi yang akan dibuat itu sendiri. Jadi, walaupun simbol dari aktor adalah gambar orang, tapi <i>actor</i> belum tentu orang. Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Asosiasi</p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p>Ekstensi/ <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yaitu <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu. Mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi obyek. Biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>
<p>Generalisasi/ <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>


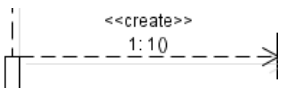
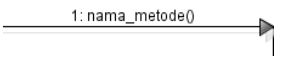
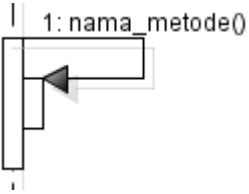
2.9 Sequence Diagram

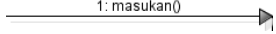
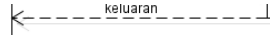
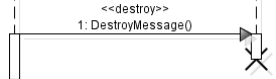
Sequence Diagram menggambarkan kelakuan obyek pada *Use Case* dengan mendeskripsikan waktu obyek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui obyek-obyek yang terlibat dalam sebuah *Use Case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi obyek itu.

Banyaknya *sequence* diagram yang harus digambarkan adalah sebanyak *Use Case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada *Sequence* Diagram sehingga semakin banyak *Use Case* yang didefinisikan maka *Sequence* Diagram yang dibuat juga semakin banyak [10].

Tabel 2- 2
Simbol *Sequence* Diagram

Simbol	Nama	Deskripsi
	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat sendiri, jadi walaupun simbol dari aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal <i>frase</i> nama aktor.
	Garis hidup / <i>lifeline</i>	Menyatakan kehidupan suatu obyek.
	Obyek	Menyatakan obyek yang berinteraksi pesan.

Simbol	Nama	Deskripsi
	Waktu aktif	Menyatakan obyek dalam keadaan aktif dan berinteraksi pesan.
	Pesan tipe <i>create</i>	Menyatakan suatu obyek membuat obyek yang lain, arah panah mengarah pada objek yang dibuat.
	Pesan tipe <i>call</i>	<p>Menyatakan suatu obyek memanggil operasi/ metode yang ada pada obyek lain atau dirinya sendiri.</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/ metode, karena ini memanggil operasi/ metode maka operasi/ metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas obyek yang berinteraksi.</p>

Simbol	Nama	Deskripsi
	Pesan tipe <i>send</i>	Menyatakan bahwa suatu obyek mengirimkan data/ masukan/ informasi ke obyek lainnya, arah panah mengarah pada obyek yang dikirim.
	Pesan tipe <i>return</i>	Menyatakan bahwa suatu obyek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke obyek tertentu, arah panah mengarah pada obyek yang menerima kembalian.
	Pesan tipe <i>destroy</i>	Menyatakan suatu obyek mengakhiri hidup obyek yang lain, arah panah mengarah pada obyek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .