

BAB II

LANDASAN TEORI

2.1 Alarm

Alarm secara umum dapat didefinisikan bunyi peringatan atau pemberitahuan. *Alarm* telah dipatenkan pada tahun 1853 oleh *Pendeta Russell Paus Augustus (1819-1858)* dari *Somerville* (Karen, 1992).

2.1.1 Perkembangan Generasi Alarm

A. Generasi Awal

Di jaman dahulu, banyak orang yang mempunyai sistem alarm local yang simpel untuk mengontrol *bell* dan *sirene* (Petruzzellis, 1994). Peningkatan teknologi alarm anti pencuri dan alarm anti kebakaran mulai terjadi sejak awal 1880-an saat Chauncey McCulloch dari Baltimore mendirikan sebuah sistem pembagian jaringan sirkuit tunggal menjadi beberapa bagian yang dihubungkan ke stasiun pusat untuk menghemat biaya penghubungan jaringan.

Selanjutnya, alat deteksi bahaya terus berkembang. Di awal 1880-an, seorang insinyur dan perancang *lokomotif* bernama Frederick Grinnel secara radikal membuat sistem keamanan kebakaran dengan mematenkan teknologi percikan (api) yang dapat terbuka ketika di sekelilingnya terjadi suhu panas yang ekstrem untuk menghindari terjadinya kebakaran besar. Kemudian, muncul teknologi *ionisasi* di mana teknologi yang digunakan adalah mendeteksi asap dan *sensor* gerakan yang dapat mendeteksi gerakan mencurigakan untuk mengaktifkan alarm keselamatan, meningkatkan *detektor* panas, dan mengaktifkan sensor

magnetik pada pintu dan jendela rumah ataupun gedung perkantoran untuk menghindari bahaya pencurian.

B. Generasi Internet

Teknologi internet merevolusi sistem keamanan komunikasi dan kemampuan pengawasan jarak jauh. Sejak era komunikasi digital pada tahun 1970-an, pemberian sinyal melalui stasiun pusat telah memberikan kontribusi terhadap teknologi komunikasi. Teknologi semakin berkembang dan memunculkan teknologi nirkabel yang terhubung melalui koneksi IP di lebih dari 97 negara di dunia. *Konvergensi* komunikasi *internet* dan *digital* membuat pengawasan video pada stasiun pusat menjadi lebih baik dan rinci kepada setiap pengguna alarm. Peralatan deteksi berkembang semakin baik dengan *sensor* yang dapat diandalkan dan mampu memberikan sinyal keamanan dengan lebih sensitif.

Impian yang direalisasikan oleh para pelopor seperti Gamewell dan Holmes lebih dari 150 tahun yang lalu telah membawa perkembangan sistem alarm modern menuju level yang lebih tinggi terhadap keamanan dan keselamatan publik yang tidak pernah ada dalam industri sebelumnya. Generasi masa kini dapat menikmati teknologi keselamatan yang dapat melindungi kehidupan mereka melalui alarm keselamatan.

Alarm mempunyai beberapa jenis, diantaranya : *Alarm* rumah, *Alarm* mobil, *Alarm* kebakaran, *Alarm* banjir, *Alarm* gempa, *Alarm* bayi, *Alarm* komputer, *Alarm online*, *Alarm* telepon genggam, *Alarm* jam, *Sirine*, *Klakson*.

2.2 *Global Positioning System (GPS)*

GPS adalah satu-satunya sistem satelit navigasi global untuk penentuan lokasi, kecepatan, arah, dan waktu yang telah beroperasi secara penuh di dunia saat ini (*Wireless Glossary of Terms*, 2012). *GPS* menggunakan konstelasi 27 buah satelit yang mengorbit bumi, dimana sebuah *GPS receiver* menerima informasi dari tiga atau lebih satelit tersebut untuk menentukan posisi. *GPS receiver* harus berada dalam *line-of-sight (LoS)* terhadap keempat satelit tersebut untuk menentukan posisi, sehingga *GPS* hanya ideal untuk digunakan dalam *outdoor positioning*.

GPS menyediakan posisi dengan ketepatan akurasi hingga 15 meter, yang berarti jika *GPS receiver* memberikan koordinat terhadap suatu lokasi tertentu, maka boleh diharapkan lokasi sebenarnya berada dalam radius 15 meter dari koordinat tertentu (El-Rabbany, 2002). Ketepatan *GPS* bergantung daripada lokasi *GPS receiver*-nya dan halangan terhadap sinyal satelit *GPS*. Meski secara umum, *GPS* menawarkan tingkat ketelitian 15 meter, namun akurasi ini dapat ditingkatkan dengan berbagai teknik, seperti *Assisted GPS (A-GPS)*, *Differential GPS (D-GPS)*, atau *Wide Area Augmentation System (WAAS)*. Pada kenyataan di lapangan, akurasi dapat bervariasi sesuai keadaan.

2.3 *Location Based Service (LBS)*

Merupakan suatu layanan yang bereaksi aktif terhadap perubahan entitas posisi sehingga mampu mendeteksi letak objek dan memberikan informasi layanan sesuai dengan letak objek yang telah diketahui tersebut. Informasi ini dapat diakses menggunakan piranti *mobile* melalui jaringan Internet dan seluler

serta memanfaatkan kemampuan penunjuk lokasi pada piranti *mobile* (Riyanto, 2009).

2.4 *Android*

Android adalah sistem operasi untuk selular atau *tablet pc* yang berbasis linux. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam-macam piranti bergerak (*mobile*) (Yopa, 2011).

Awalnya, *Google Inc.* membeli *android Inc.* pendatang baru yang membuat peranti lunak untuk perangkat ponsel. Kemudian untuk mengembangkan *android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, piranti lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia* (Yopa, 2011).

2.4.1 Jenis-Jenis dan varian *Android*:

- a) *Android* versi 1.1 pada 9 maret 2009, *Google* Merilis *android* versi 1.1. *Android* versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam, *Voice Search*(pencarian suara), pengiriman pesan dengan *Gmail*, dan pemberitahuan email.
- b) *Android* versi 1.5 (*cupcake*) pada pertengahan maret 2009. Terdapat beberapa pemberharuan termasuk juga penambahan beberapa fitur dalam ponsel versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke youtube dan gambar picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung

secara otomatis ke *headset Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

- c) *Android* versi 1.6 (*Donut*) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol *applet VPN*. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, *camcorder* dan galeri yang diintegrasikan; *CDMA/EVDO*, *802.1x*, *VPN*, *Gestures* dan *text to speech engine*; kemampuan dial kontak; teknologi pengadaan resolusi *VWGA*; *teknologi text to change speech*.
- d) *Android* versi 2.0/2.1 (*Eclair*) dirilis pada 3 desember 2009. Perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan *google maps 3.1.2*, perubahan UI dengan browser barudan dukungan *HTML5*. Daftar kontak yang baru, dukungan flash untuk kamera 3.2 MP, digital zoom, dan mengadakan kompetisi aplikasi terbaik(*killer apps*).
- e) *Android* versi 2.2 (*Froyo: Frozen yoghurt*) dirilis pada mei 2010. Perubahan yang dilakukan berupa penambahan dukungan terhadap *adobe flash 10.1*, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi *V8 engine script* yang dipakai *google chrome* yang digunakan untuk mempercepat kemampuan rendering pada *browser*, pemasangan aplikasi dalam *SDCard*, kemampuan *wifihostspot portable*, dan kemampuan *auto update* pada aplikasi *android market*.
- f) *Android* versi 2.3 (*Gingerbread*) dirilis pada 6 desember 2010. Perubahan umum yang dilakukan adalah meningkatkan kemampuan permainan(*gaming*). Peningkatan fungsi *copy paste*, layar antar muka di

desain ulang, dukungan format video VP8 dan WebM, efek audio baru(*reverb, equalization, bass boost*), dukungan kemampuan *Near Field Communication* (NFC), dan dukungan jumlah kamera yang lebih dari satu.

2.5 GPS (*Global Positioning System*) dengan *Android Location-Based API*

Android LBS dapat membaca lokasi *smartphone* dari GPS dan lokasi dapat kita lihat sebagai *longitude, latitude* dan waktu pembacaan lokasi. Untuk bisa membuat aplikasi berbasis lokasi maka sangatlah dibutuhkan pengetahuan tentang bagaimana Android LBS dapat membaca GPS dan apa saja fungsi dari LBS (Dimarzio, 2008: 212).

2.5.1 Elemen-elemen Penting dari LBS

LBS memiliki beberapa element penting, namun ada dua element penting yang perlu dipelajari (Meier, 2010:246). Dua element penting dan fungsinya adalah:

- 1.) Location Manager, berfungsi sebagai penghubung ke servis LBS. Dan dengan terhubung ke service LBS, maka aplikasi dapat mengambil lokasi pada waktu itu juga, aplikasi juga dapat melakukan *tracking*, menyetting pemberitahuan apabila masuk ke suatu area yang sudah ditentukan, dan dapat mengetahui *location providers* yang sedang aktif.
- 2.) Location Providers, berfungsi sebagai representasi salah satu dari berbagai provider yang dipakai untuk pengambilan lokasi.

2.6 GPS Tidak Bisa Aktif Langsung dari Program

Banyak dari forum mengatakan bahwa GPS tidak bisa diaktifkan dari program, GPS hanya bisa diaktifkan dari *Setting Android* sendiri. Salah satu pembahasan tentang cara mengaktifkan GPS dari program yang diakses dari <http://stackoverflow.com/questions/4721449/enable-gps-programmatically-like-tasker> pada tanggal 27 Februari 2013 pukul 23:00 dapat dilihat pada gambar berikut :

```

try {
    pacInfo = pacman.getPackageInfo("com.android.settings", PackageManager.GET
} catch (NameNotFoundException e) {
    return false; //package not found
}

if(pacInfo != null){
    for(ActivityInfo actInfo : pacInfo.receivers){
        //test if receiver is exported. if so, we can toggle GPS.
        if(actInfo.name.equals("com.android.settings.widget.SettingsAppWidgetP
            return true;
        }
    }
}

return false; //default
}

```

share | improve this answer edited Aug 4 '11 at 19:56 answered Mar 14 '11 at 23:26
Ben H
904 4 11

hmmm interesting... I was sure it would not be a very "legal" way to do this... thanks! – maid450 Mar 15 '11 at 10:06

1 At the time of this (my) comment, the links in this answer seem to indicate that the bug this exploits has recently been fixed. I just wanted to point out that the exploit still seems to work just fine in my own test environment, so you shouldn't give up on trying this... just be sure that your code will handle any errors if it doesn't work! – SillithCrowe Jun 22 '11 at 16:47

1 As of this comment's writing, this exploit **still works** on a 2.2.1 Android phone. Nice find, Ben H. – Qix Aug 17 '11 at 15:33

yes, the bug was supposedly fixed in android 2.3. however, some custom 2.3+ ROMs include the unfixed version of the power control widget, so it works on them. – Ben H Aug 22 '11 at 21:53

Hello , here com.android.settings is app Package name or what ? – girishce26 Nov 14 '11 at 11:19

Gambar 2.1 Pembahasan Cara Mengaktifkan GPS

Pada gambar 2.1 terlihat bahwa beberapa komentar mengatakan bahwa tidak ada *code* yang bersifat *legal* dari *google* pada android untuk mengaktifkan GPS langsung.

2.7 Menggunakan layanan SMS pada Aplikasi Android

Android menyediakan fungsi sms yang bisa dipakai aplikasi melalui *SMS Manager*. Dengan menggunakan *SMS Manager* kita bisa menggantikan aplikasi sms default android untuk mengirim pesan sms, menerima pesan sms yang masuk, atau digunakan sebagai media transfer data (Meier, 2010:399).

2.7.1 Mengirim SMS dari Application Menggunakan *Intents* and the *Native Client*

Banyak keadaan yang memungkinkan aplikasi android mengirim sms dengan cara termudah. Cara termudah untuk melakukan pengiriman SMS adalah dengan menjalankan activity dengan menggunakan sebuah *intent* dengan *action* "*Intent.ACTION_SENDTO*". Tambahkan nomor target sms yang dituju sesuai skema *intent data*. Tambahkan juga pesan yang mau dikirim dengan memasukkannya ke dalam *intent* menggunakan *extra sms_body*. Berikut adalah source code mengirim sms menggunakan activity :

```
Intent smsIntent = new Intent(Intent.ACTION_SENDTO,
Uri.parse("sms:55512345"));
smsIntent.putExtra("sms_body", "Press send to send me");
startActivity(smsIntent);
```

2.7.2 Mengirim SMS Secara Manual

Pengiriman SMS pada *android* menggunakan *SmsManager*. Kita bisa mendapatkan referensi dari *SMS Manager* menggunakan *method SmsManager.getDefault*. Seperti yang ditunjukkan pada code di bawah ini :

```
SmsManager smsManager = SmsManager.getDefault();
```


Untuk bisa mengirim sms secara manual, dalam manifest harus di spesikasikan *uses-permission*. Seperti dibawah ini :

```
<uses-permission android:name="android.permission.SEND_SMS"/>
```

Sedangkan contoh code pengiriman sms secara manual adalah sebagai berikut : *smsManager.sendMessage(noHp, null, pesan, null, null);*

2.8 UML (Unified Modelling Language)

UML adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi obyek.

UML merupakan standar yang relatif terbuka yang dikontrol oleh OMG (*Object Management Group*), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar-standar yang mendukung interoperabilitas, khususnya interoperabilitas sistem berorientasi obyek. OMG mungkin lebih dikenal dengan standar-standar CORBA (*Common Object Request Broker Architecture*).

UML lahir dari penggabungan banyak bahasa pemodelan grafis berorientasi obyek yang berkembang pesat pada akhir 1980-an dan awal 1990. Sejak kehadirannya pada 1997, UML menghancurkan menara Babel tersebut dan menjadi sejarah (Fowler, 2004: 1-2).

Tujuan UML diantaranya adalah :

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.

2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasapemrograman dan proses rekayasa.
3. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.

Untuk membuat suatu model, UML memiliki diagram grafis sebagai berikut:

- a. *Business Use Case Model*.
- b. *Activity Diagram*.
- c. *Use Case Model*.
- d. *Behavior Diagram* antara lain *Sequence Diagram*.
- e. *Implementation Diagram*, meliputi *Component Diagram* dan *Deployment Diagram*.
- f. *Generate Code*.

Diagram-diagram tersebut diberi nama berdasarkan sudut pandang yang berbeda-beda terhadap sistem dalam proses analis atau rekayasa.

Dibuatnya berbagai jenis diagram diatas karena:

- a. Setiap sistem yang kompleks lebih baik jika dilakukan pendekatan melalui himpunan berbagai sudut pandang yang kecil yang satu sama lain hampir saling bebas (*independent*). Sudut pandang tunggal senantiasa tidak mencukupi untuk melihat isi sistem yang lebih besar dan kompleks.
- b. Diagram yang berbeda-beda tersebut dapat menyatakan tingkatan yang berbeda-beda dalam proses rekayasa.
- c. Tujuan adanya diagram-diagram tersebut adalah agar model yang dibuat semakin mendekati realitas.

Diagram-diagram ini ditambah dengan kemampuan dokumentasi sebagai *artifacts* utama UML. *Data-flow diagram* dan tipe diagram lain yang tidak terdapat dalam UML tidak termasuk dalam paradigma *object-oriented*. *Activity diagram* dan *collaboration diagram* yang terdapat dalam UML menggantikan *data-flow diagram*. *Activity diagram* juga sangat bermanfaat untuk membuat *workflow*.

STIKOM SURABAYA