

Rancang Bangun Aplikasi Pengintai Aktifitas Komputer Menggunakan Layanan Cloud To Device Messaging (C2DM) Pada Smartphone Android

Azwar Anas¹⁾, Anjik Sukmaaji, S.Kom.,M.Eng²⁾, Teguh Sutanto, M.Kom.,MCP³⁾

¹⁾ Mahasiswa S1 Sistem Informasi STIKOM Surabaya, ²⁾ Dosen Jurusan Sistem Informasi STIKOM Surabaya, ³⁾ Dosen Jurusan Sistem Informasi STIKOM Surabaya
e-mail: ¹⁾ azwar.nazz@gmail.com, ²⁾ anjik@stikom.edu, ³⁾ teguh@stikom.edu

Abstract

Computer ownership is increasing makes users more or less will affect the number of users who interact with it. Based on the results of research conducted National Center Education Statistics (NCES, 2003). Segmentation of Internet use in adolescents in this age range as much as 42 use it to complete school assignments, while 38 percent use email or chat facilities which includes playing games online, and 22 percent subsequently use the internet for browsing and searching for information. But not all the content containing educational information, and also a lot of negative information that is not compliant accessible to children. To solve this problem it created a system that can monitor the activities of users such as children when accessing the computer in realtime using the Cloud To Device Messaging Service (C2DM) on Android smartphones by parents. Given the android smartphone is also a handset that is able to always connect to the Internet for realtime data synchronization is different from ordinary mobile phones to be done manually. Given this system, the role of parents to monitor their children when accessing the computer is not overlooked.

Keyword: Android, C2DM, Monitors Computer Activity, Keylogger

Pendahuluan

The National Center for Education Statistic (NCES) adalah entitas federal utama untuk mengumpulkan, menganalisa dan melaporkan data yang terkait dengan pendidikan di Amerika dan negar-negara lainnya. Untuk memenuhi mandate kongres dalam mengumpulkan, menyusun, menganalisa dan melaporkan penuh dan lengkap pada kondisi di Amerika. Melakukan dan mempublikasikan analisis laporan khusus serta membantu negara dan lembaga setempat dalam meningkatkan sistem statistik serta ulasan dan melaporkan kegiatan pendidikan di luar negeri.

Menurut hasil survei dari *National Center Education Statistic* (NCES, 2003), rumah dan sekolah adalah tempat utama anak-anak dan remaja

menggunakan komputer, 81 persen dari anak usia (5-17) tahun menggunakan komputer disekolah dan 65 persen menggunakan komputer dirumah secara umum penggunaan komputer disekolah lebih banyak daripada dirumah. Segmentasi penggunaan internet pada remaja di rentang usia ini sebanyak 42 menggunakannya untuk menyelesaikan tugas sekolah, sementara 38 persen menggunakan sarana email atau *chatting* termasuk didalamnya adalah bermain *game online*, dan 22 persen selanjutnya menggunakan internet untuk *browsing* dan mencari informasi.

Mengingat dari waktu ke waktu, kepemilikan komputer semakin meningkat tiap tahun, sedikit banyak akan mempengaruhi jumlah pengguna yang berinteraksi dengannya. Berbagai-bagai konten di internet misalnya, tidak semua konten berisi hal-

hal berbau pendidikan maupun pengetahuan yang bisa digunakan sebagai wawasan belajar, namun banyak juga konten - konten yang berisi informasi negatif yang tidak selayaknya diakses oleh seorang anak dan remaja. Oleh karena itu, akibat dari meningkatnya peran komputer rumah dan informasi yang beragam dalam kehidupan anak-anak maka diperlukan sebuah perhatian khusus yang tidak boleh lepas dari orang tua.

Dari permasalahan diatas maka diperlukan suatu sistem yang mampu memantau aktifitas pengguna misalnya anak pada saat mengakses komputer rumah atau dimanapun berada secara *realtime* menggunakan perangkat *mobile smartphone android* bagi orang tua. Dengan adanya sistem ini, maka peran orang tua untuk memantau aktifitas anaknya dalam penggunaan komputer dan akses internet tidak akan terlewatkan.

Keylogger

(Kurniawati, 2010) *keylogger* adalah program untuk memonitor (memantau) segala aktifitas yang dilakukan oleh pengguna komputer, jenis *keylogger* yang digunakan pada sistem ini adalah *keylogger software* berupa aplikasi yang nantinya akan diinstal pada komputer yang mampu merekam segala inputan dari *keyboard*.

Untuk dapat merekam segala aktifitas *keyboard*, *keylogger* harus dijalankan setelah sistem operasi berjalan dan masuk pada tampilan desktop. Karena *keylogger* merupakan perangkat lunak yang bersifat memantau maka *keylogger* akan dijalankan secara *background process* dan tidak diketahui oleh pengguna yang menggunakan komputer tersebut.

XML Web Service

(Ramadha, 2010) *Web service* merupakan perangkat lunak yang mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web service* ini akan memfasilitasi suatu sistem yang membutuhkan layanan untuk berkomunikasi dengan sistem yang lain.

Karena sistem yang akan dibangun bersifat *client server* maka dibutuhkan sebuah *web service* untuk dapat berkomunikasi dengan sistem yang lain. Informasi yang dikirim dari *client* ke *server* atau dari *server* ke *client* di format dalam bentuk XML sehingga data ini dapat diakses oleh sistem walaupun berbeda *platform*.

Sedangkan untuk proses pengiriman atau penerimaan data binary image dari *client* ke *server* atau dari *server* ke *client* tetap menggunakan format XML *web service* sebagai medianya, namun dibutuhkan suatu pengkodean yang mampu mengubah format *binary image* menjadi *ASCII string* yaitu menggunakan format *Base64 encode* dan *Base64 Decode*, metode ini mampu mngubah data yang semula berupa *binary image* menjadi *ASCII string*.

Android C2DM

Semenjak *android* versi 2.2 (*Froyo*) diluncurkan, *Google* telah menambahkan layanan baru untuk membantu pengembang untuk dapat mengirim data dari *server* ke aplikasi mereka pada ponsel *android*. *Android Cloud To Device Messaging (C2DM)* membuat lebih mudah aplikasi *mobile* untuk melakukan sinkronisasi data dengan *server* (Huang, 2010).

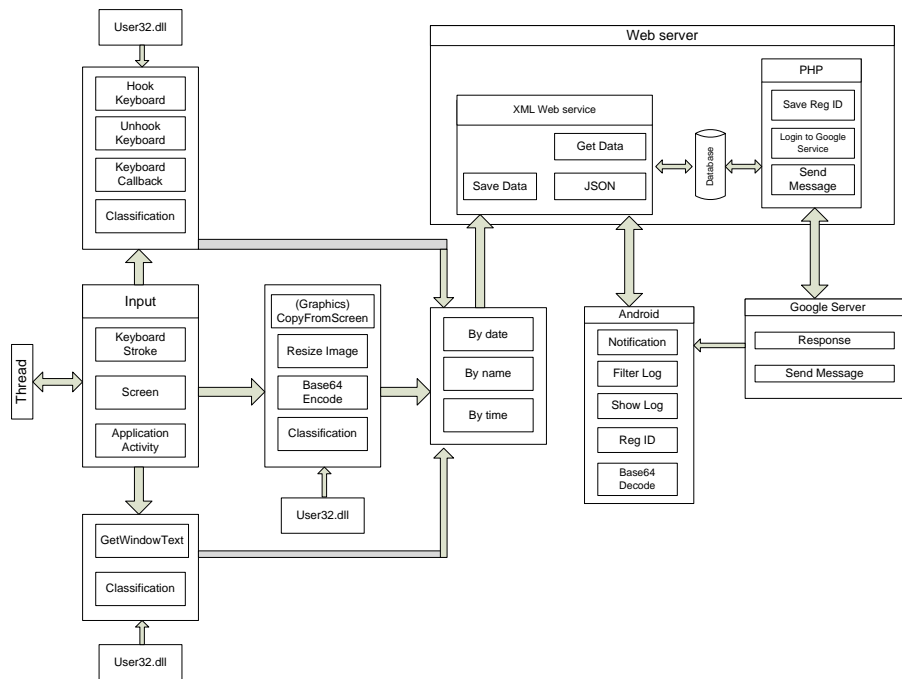
Sebelum dapat menggunakan layanan *C2DM* aplikasi harus

didaftarkan terlebih dahulu pada *server Google* menggunakan nama paket aplikasi misalnya *com.project.ispy* dan menyertakan email *Google* yang masih aktif untuk diverifikasi oleh pihak *Google*.

Sedangkan mekanisme pengiriman pesan melalui *C2DM*, pertama-tama aplikasi *android* melakukan koneksi ke *C2DM Frontend* menggunakan email yang sudah diverifikasi untuk mendapatkan registrasi id, setiap perangkat akan mendapatkan registrasi id yang berbeda-beda, namun registrasi id tersebut bersifat tidak permanen oleh karena itu sistem akan memperbaharui registrasi id secara berkala. Selanjutnya aplikasi

android harus mengirimkan registrasi id tersebut pada *web server* untuk disimpan tujuannya adalah sebagai asumsi saat pengiriman data agar tidak terjadi kesalahan pada penerima. Pengiriman pesan dari *web server* ke *server C2DM* dimulai dari *web server* melakukan login pada *server Google* melalui *Client Login Service* dan jika berhasil maka akan mendapatkan *Auth token* sementara yang unik, *Auth Token* dan registrasi id yang disimpan tersebut disertakan pada saat mengirim pesan dari *web server* ke *C2DM Frontend*. Permintaan yang sudah di autentikasi selanjutnya akan dimasukkan kedalam *C2DM Message Queue* yang nantinya diteruskan ke aplikasi *mobile* menggunakan koneksi *TCP/IP*.

Gambaran Umum Sistem



Gambar 1 Gambaran Umum Sistem

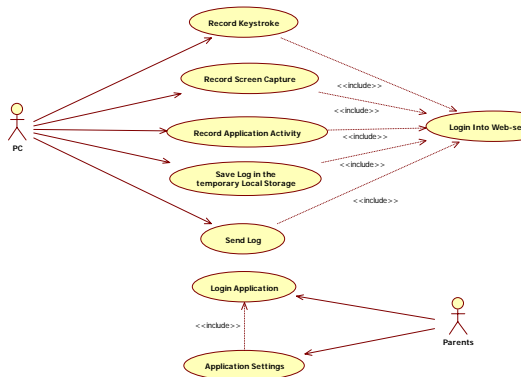
Konsepnya yaitu membangun tiga buah aplikasi yang mendukung *server side* dan *client side*. Aplikasi yang berjalan di sisi *server side* berupa *web*

server dan *web service* yang berfungsi menyimpan data *log* atau *history* dan penghubung antara aplikasi *client side*. Sedangkan aplikasi *client side* terdiri

dari aplikasi *keylogger* yang diinstall pada komputer yang akan diintai aktifitasnya dan aplikasi *mobile* yang digunakan untuk *smartphone android* sebagai media penerima informasi *log*. Agar dapat melakukan fitur dari sistem ini semua perangkat harus terhubung dengan internet terutama aplikasi *keylogger*, karena berdasarkan interval yang sudah ditentukan *keylogger* akan secara otomatis mengirimkan *log* atau *history* ke *web server* untuk disimpan.

Pembahasan

Use Case Diagram

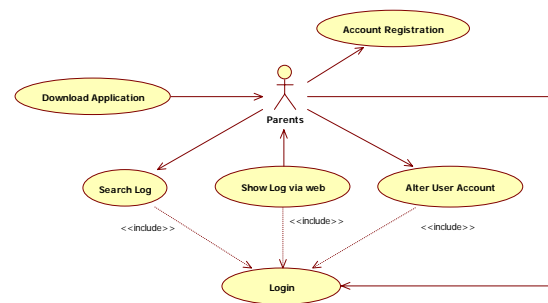


Gambar 2 Use Case Diagram Aplikasi Keylogger

Tabel 1 Penjelasan singkat *Use case diagram* aplikasi desktop *keylogger*

Nama Use Case	Deskripsi
<i>Login Application</i>	Proses yang menangani login pada aplikasi <i>keylogger</i> sebelum dapat melakukan pengaturan aplikasi <i>keylogger</i> .
<i>Application Setting</i>	Proses yang menangani pengaturan aplikasi <i>keylogger</i> seperti interval <i>capture</i> dan interval <i>log</i> .
<i>Login into Web-server</i>	Proses yang menangani login ke server sebelum dapat melakukan pengiriman <i>log</i> ke server.
<i>Record</i>	Proses yang merekam segala

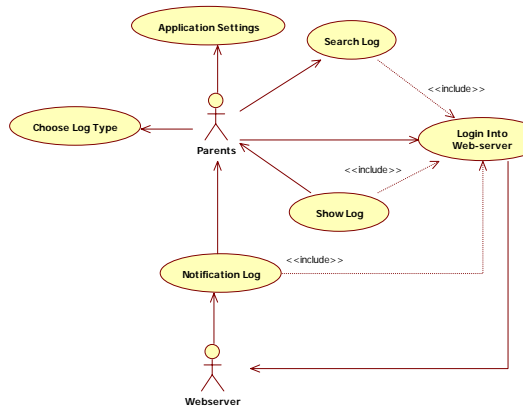
<i>Keystroke</i>	aktivitas keyboard
<i>Record Screen Capture</i>	Proses yang <i>capture</i> aplikasi yang berjalan
<i>Record Application Activity</i>	Proses yang merekam segala aktivitas aplikasi
<i>Save log in the Temporary Local Storage</i>	Proses yang digunakan untuk menyimpan sementara <i>log</i> pada penyimpanan lokal sebelum dikirim ke server
<i>Send Log</i>	Proses yang menangani pengiriman <i>log</i> ke server via internet.



Gambar 3 Use Case Diagram Aplikasi Website

Tabel 2 Penjelasan singkat *Use Case Diagram* aplikasi web site untuk *User Member*

Nama Use Case	Deskripsi
<i>Account Registration</i>	Proses yang menangani pendaftaran pengguna baru yang akan menggunakan sistem ini
<i>Search Log</i>	Proses yang menangani Pencarian <i>Log</i>
<i>Download Application</i>	Proses yang digunakan untuk melakukan unduh aplikasi dari web site
<i>Login</i>	Proses yang menangani login pengguna sebelum masuk pada halaman <i>member area</i> .
<i>Alter User Account</i>	Proses yang menangani pengaturan pengguna akun.
<i>Show Log Via Web</i>	Proses yang menangani penampilan <i>log</i> via web site.



Gambar 4 Use Case Diagram Aplikasi Android

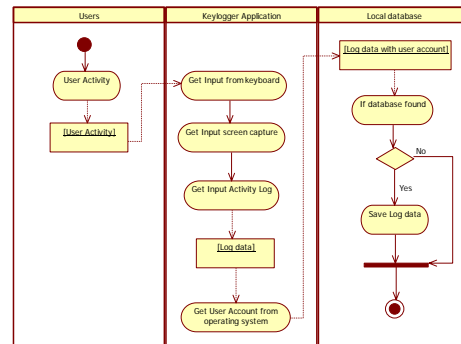
Tabel 3 Penjelasan Singkat Use Case

Diagram Aplikasi Mobile

Nama Use Case	Deskripsi
<i>Login Into Web server</i>	Proses yang menangani login ke <i>Web server</i> sebelum dapat menerima log
<i>Application Settings</i>	Proses yang menangani pengaturan aplikasi
<i>Show Log</i>	Proses yang digunakan untuk menampilkan log
<i>Search Log</i>	Proses yang digunakan untuk mencari log
<i>Choose Log</i>	Proses yang digunakan untuk memilih tipe log yang digunakan
<i>Notification Log</i>	Proses yang dijalankan ketika ada pemberitahuan log baru

Activity Diagram

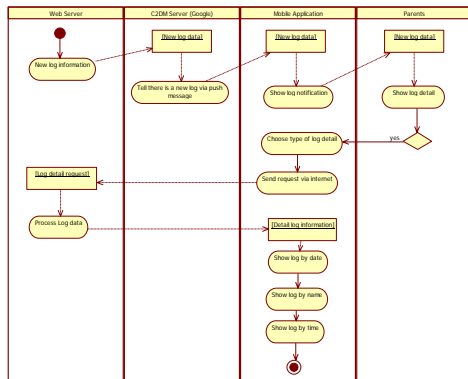
Proses merekam aktifitas *keyboard* dimulai saat ada interaksi dari pengguna yang memakai komputer atau disebut dengan *user activity*, *user activity* bisa berupa aktivitas yang terjadi saat *keyboard* komputer ditekan dan berupa *screen capture* dari layar aplikasi yang sedang aktif hasilnya berupa data *log*, proses ini dilakukan secara otomatis oleh aplikasi. Selanjutnya aplikasi akan menambahkan informasi pengguna yang menggunakan komputer tersebut, data ini diambil dari *user account* masing-masing komputer. Selanjutnya data data log akan disimpan sementara pada lokal database sebelum dikirim ke *web server*. Proses berikutnya jika *database* lokal ditemukan maka data *log* akan disimpan sementara pada tabel *keystroke*, namun jika *database* lokal tidak ditemukan maka data *log* tidak akan disimpan. Untuk lebih jelasnya dapat dilihat pada gambar 5.



Gambar 5 Activity Diagram Merekam Aktifitas Keyboard

Proses menampilkan *log* dimulai dari dari *web server* yang telah menerima *log* baru dari aplikasi *keylogger* akan memberitahukan pada *server C2DM* bahwa ada *log* baru yang harus diberitahukan pada ponsel *android*, ponsel *android* yang menerima

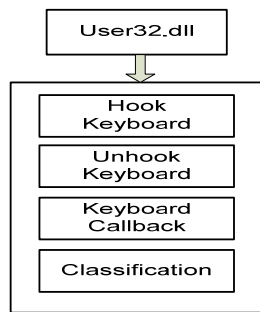
pesan tersebut akan menampilkan notifikasi *log* baru pada *parents*, informasi *log* baru tersebut bukan keseluruhan *log* melainkan hanya *header log* berupa tanggal dan pengguna komputer. *Parents* yang ingin melihat lebih detail lagi informasi *log* dapat memilih menu *preview log* pada aplikasi *mobile*, selanjutnya permintaan akan dikirim ke *web server* menggunakan media internet, *web server* akan segera memproses permintaan dan menghasilkan informasi detail *log* dan mengirimkannya kembali pada ponsel. Untuk lebih jelasnya dapat dilihat pada gambar 6.



Gambar 6 Activity Diagram

Hasil Pembahasan

Implementasi Keylogger



Gambar 7 Diagram Keylogger

Proses ini terjadi dilakukan ketika pengguna menekan setiap tombol di *keyboard*, kejadian ini ditangkap menggunakan fungsi *HookKeyboard()* yang merupakan *Windows API* dari sistem operasi *windows*, oleh karena itu proses ini memerlukan *library user32.dll* yang sudah disediakan *windows*.

```

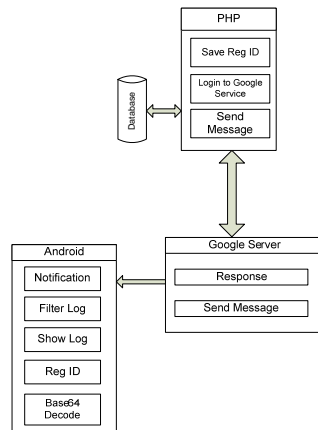
'create the hook
Public Sub HookKeyboard()
    callback = New
    KeyboardHookDelegate(AddressOf
    KeyboardCallback)
    KeyboardHandle = SetWindowsHookEx(13,
    callback,
    Process.GetCurrentProcess.MainModule.BaseAddress
    s, 0)
End Sub
  
```

Selanjutnya fungsi *HookKeyboard()* akan memanggil setiap alamat pada tombol *keyboard* yang sebelumnya berbentuk karakter ASCII Hex untuk diubah menjadi ASCII karakter, karena ASCII Hex berupa angka dan sulit untuk dibaca. Variabel pengganti karakter ASCII Hex ke bentuk ASCII karakter lebih lengkapnya dapat dilihat pada lampiran. Selanjutnya hasil dari fungsi *HookKeyboard()* akan diklasifikasikan berdasarkan nama *user account*, tanggal, waktu, judul aplikasi yaitu judul yang terdapat pada setiap aplikasi yang berjalan dan *log* dari inputan *keyboard* yang sudah berbentuk karakter string. Proses merekam aktifitas *keyboard* dapat dihentikan dengan cara memanggil fungsi *Unhookkeyboard()*.

```

Public Sub UnhookKeyboard()
    If (Hooked()) Then
        If
        UnhookWindowsHookEx(KeyboardHandle)
        <> 0 Then
            KeyboardHandle = 0
        End If
    End If
End Sub
  
```

Implementasi C2DM



Gambar 8 Diagram keylogger

Proses pengiriman pesan ke *android* menggunakan *push message* harus lewat *Google C2DM Server* terlebih dahulu. Oleh karena itu diperlukan sebuah *server* khusus yang menangani proses pengiriman pesan, *web server* harus mampu melakukan login ke *server C2DM* untuk mendapatkan token autentikasi. Alamat login ke *server google* sudah disediakan seperti berikut

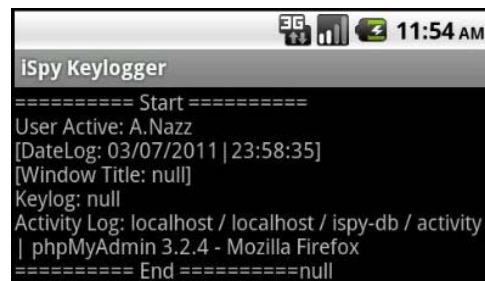
```
curl
https://www.google.com/accounts
/ClientLogin -d
Email=theEmailYouWhitelisted -d
Passwd=pass***word -d
accountType=HOSTED_OR_GOOGLE -d
source=Google-cURL-Example -d
service=ac2dm
```

Setelah mendapatkan token autentikasi maka *web server* mendapatkan ijin untuk mengirim pesan melalui *push message*. Namun pesan yang dikirim menggunakan *push message* hanya terbatas pada 1024 karakter.

```
url --header "Authorization:
GoogleLogin
auth=**authFromRegistrationAbov
e**"
"https://android.apis.google.co
m/c2dm/send" -d
registration_id=**phoneRegistra
tionId(riciever)** -d
"data.message=StringToPass" -d
collapse_key=something -k
```

Selain informasi berupa notifikasi, sistem juga mempunyai fungsi pencarian *log* agar pengguna juga dapat mencari *log* berdasarkan tanggal, nama pengguna dan waktu. Namun proses pencarian *log* ini sudah tidak lagi menggunakan C2DM sebagai pengirim pesan, sistem menggunakan protokol *http://* melalui internet untuk *me-request* dan menerima data dari *web server* dalam format XML dan JSON.

Berikut adalah hasil log yang muncul pada ponsel.



Gambar 10 Hasil Log pada ponsel

Kesimpulan

Berdasarkan implementasi dan evaluasi yang telah dilakukan, dapat disimpulkan bahwa :

- Dengan layanan *push message Android C2DM* proses pengawasan aktifitas penggunaan komputer dapat dilakukan secara *realtime*. Dengan adanya sistem ini, peran orang tua untuk memantau aktifitas penggunaan komputer oleh anaknya tidak sampai terlewatkan.
- Proses integrasi aplikasi *keylogger*, *web server* dan *mobile android* berhasil dilakukan dengan melalui jaringan komunikasi internet.

Saran

Adapun saran yang dapat diberikan kepada peneliti berikutnya apabila ingin mengembangkan sistem yang telah

dibuat ini agar menjadi lebih baik adalah :

- a. Menambahkan fitur yang mampu membaca history *web site* pada browser di aplikasi *keylogger*.
- b. Menambahkan fitur *paging* pada aplikasi *mobile android* agar dapat menghemat *resource* saat menampilkan *log*.

Daftar Pustaka

[1] Fowler, Martin, 2004, *UML Distilled Edisi 3 Panduan Singkat Bahasa Pemodelan Objek Standar*, ANDI, Yogyakarta.

[2] Huang, Wei, 2010, *Android Cloud To Device Messaging*, <http://android-developers.blogspot.com/2010/05/android-cloud-to-device-messaging.html> diakses 04-05-2011 pukul 11.00

[3] Kurniawati, Dewi, 2010, *Kegunaan Keylogger*, Jurnal Teknologi Informasi.

[4] Ramadha, Kartika, *Pengenalan Web servers*, Universitas Gunadarma .

[5] Wibowo, Arip, 2011, *Web Server*, <http://unyildadakan.com/internet/web-server/> diakses tanggal 06-05-2011 pukul 19:00

[6] Ramadah, Kartika, 2010, *Web Service*, <http://kartikanurramadha.info/a/web-service.pdf> diakses tanggal 19-07-2011 pukul 22.00