

# RANCANG BANGUN MANAGEMENT APLIKASI DAN CDR (CALL DATA RECORD) PADA SISTEM INTERNET TELEPHONY BERBASIS EMBEDDED EBOX-4300

Ahmad Khadafi Sanu<sup>1</sup>, Achmad Subhan Khalilullah<sup>2</sup>

Jurusan Teknik Telekomunikasi – Politeknik Elektronika Negeri Surabaya  
Institut Teknologi Sepuluh Nopember (ITS) Surabaya  
Kampus PENS-ITS, Sukolilo, Surabaya  
Email: [khadafi\\_sanu@yahoo.com](mailto:khadafi_sanu@yahoo.com); [subhankh@lecturer.eepis-its.edu](mailto:subhankh@lecturer.eepis-its.edu)

## ABSTRAK

*Kebutuhan akan suatu informasi merupakan hal yang tidak bisa dihilangkan. Permintaan yang sangat tinggi inilah yang membuat Teknologi Telekomunikasi berkembang sangat cepat untuk membuat informasi tersebut dapat saling dikoneksikan. Dalam Proyek Akhir ini dibuat sebuah aplikasi pengaturan penggunaan aplikasi internet telephony. Sistem ini memungkinkan aplikasi telepon dan internet berjalan bersama-sama dengan ketentuan dari data billing yang telah tervalidasi sebelumnya. Aplikasi ini dijalankan pada sebuah perangkat embedded eBox-4300 dengan Sistem Operasi yang telah dimodifikasi khusus. Sistem ini juga mengatur sinkronisasi data billing dengan dua aplikasi lainnya yaitu telepon dan internet dengan komunikasi inter-proses antar aplikasi. Hasil dari Proyek Akhir ini dapat diterapkan sebagai aplikasi manajemen internet telephony yang dipakai di tempat umum.*

**Kata Kunci** : Windows CE, eBox-4300, Platform Builder, Microsoft Visual C#, Manajemen Aplikasi, CDR.

## 1. PENDAHULUAN

Di Indonesia sekarang ini perkembangan teknologi informasi semakin berkembang. Dengan adanya teknologi tersebut maka dapat dengan mudah untuk mendapatkan informasi yang dibutuhkan dari mana saja dan kapan saja. Namun ternyata akses semua informasi melalui internet hanya bisa dinikmati oleh orang kaya saja seperti banyak kasus yang beredar saat sekarang ini, adanya kesenjangan antara orang kaya dengan orang miskin karena hanya orang-orang kaya yang mampu membeli ponsel yang mahal dan memiliki fitur-fitur yang lengkap.

Oleh karenanya munculah wartel, warnet dan telepon umum. Wartel merupakan fasilitas untuk melakukan koneksi telepon. Sedangkan warnet merupakan fasilitas untuk melakukan koneksi internet. Kedua sistem yang dilengkapi billing membuat dengan mudah untuk mengolah dan mengatur dari koneksi tersebut karena pada billing terdapat suatu database yang merekam semua kegiatan aplikasi user. Pemodelan yang lain yaitu telepon umum yang media pembayarannya menggunakan uang koin atau kartu untuk melakukan panggilan. Tetapi telepon umum telah bergeser prioritasnya karena banyaknya kelemahan dari sistem ini.

Namun ternyata dari ketiga sistem diatas masih berdiri sendiri tanpa bisa berjalan bersama-sama, melakukan koneksi internet dengan telepon dalam satu sistem. Untuk mewujudkan suatu sistem komunikasi informasi tersebut, maka perlu adanya sebuah sistem yang mampu menjalankan kedua sistem secara bersama-sama dengan maksud untuk mempermudah akses fasilitas koneksi internet telephony terutama di tempat umum yang dapat dengan praktis, mudah dan murah dapat dinikmati semua lapisan masyarakat. Sistem ini tentu saja dengan dilengkapi elemen database yang selalu merekam aplikasi user saat menjalankan kedua sistem tersebut.

## 2. LANDASAN TEORI

### Image dan Kernel Sistem Operasi

*Image* dalam sistem komputer bukanlah diartikan sebagai foto atau gambaran, melainkan merupakan sebuah perwakilan dari program komputer dan data yang berhubungan, seperti kernel, sistem file, library, dan program dari komputer di suatu titik waktu.

Sedangkan dalam sistem komputasi, kernel merupakan komponen pusat dari sebagian besar sistem operasi komputer, tanggung jawabnya termasuk sistem pengolahan *resource* (komunikasi antara komponen hardware dan software). Biasanya sebagai komponen dasar dari sistem operasi, sebuah kernel dapat menyediakan layer terendah seperti *resource* (khususnya memori, processor dan Input / Output dari perangkat) untuk menjalankan sebuah software aplikasi.

### Windows Embedded CE 6.0 R2

Windows Embedded CE adalah 32-bit sistem operasi yang dirancang untuk memenuhi kebutuhan berbagai perangkat cerdas. Perangkat ini mulai dari alat-alat perusahaan, seperti industri pengendali, komunikasi, dan untuk penjualan produk-produk konsumen seperti kamera, telepon, dan perangkat hiburan rumah.

CE 6.0 menyediakan ratusan fitur OS, sumber kode akses, dan kompatibilitas dengan berbagai arsitektur prosesor. Kernel untuk CE 6.0 dapat menjalankan sebanyak 32.000 proses serentak, dengan 2GB memori virtual.

### Board Support Package (BSP)

Board Support Package (BSP) merupakan komponen penting yang dibutuhkan device hardware

handheld untuk mengembangkan OS Design dan untuk men-generate *image* CE. Karena BSP berisi konfigurasi dasar dari hardware beserta driver perangkat yang terinstall di dalamnya.

ICOP\_eBox4300\_60CS adalah BSP yang dibutuhkan untuk membuat OS Desain dan membangun *image* untuk eBox-4300. Instalasi ICOP\_eBox4300\_60CS BSP digunakan setelah instalasi Visual Studio 2005, CE 6.0 SP1 dan CE 6.0 R2 dengan bantuan x86 CPU. x86 CPU merupakan tool yang mendukung Platform Builder CE 6.0 dibutuhkan untuk menggunakan ICOP\_eBox4300\_60CS BSP untuk membuat OS desain dan membangun *image* CE 6.0 untuk eBox-4300.

### Software Development Kit (SDK)

Software Development Kits (SDK) adalah perangkat lunak yang dibutuhkan untuk membuat/mengembangkan aplikasi yang akan ditanamkan pada target device. Penggunaan dan instalasi eBox4300\_WINCE600\_SDK ini dilakukan setelah Visual Studio 2005, CE 6.0, CE 6.0 SP1 dan CE 6.0 R2 diinstall dengan dukungan x86 CPU.

### CoreCon

CoreCon adalah komponen yang dibutuhkan untuk menyambungkan koneksi antara device CE 6.0 dengan Visual Studio 2005 dengan tujuan untuk mengunduh aplikasi Visual Studio 2005 ke dalam device CE 6.0 untuk testing dan debug. Komponen file corecon untuk device x86 terdapat pada instalasi VS2005 dan terletak pada folder instalasi, yaitu

```
..\ProgramFiles\CommonFiles\MicrosoftShared\CoreCon\1.0\Target\WCE400\x86\
```

### eBox-4300

Mini PC eBox-4300 adalah sebuah komputer mini yang sangat compact dan cocok untuk aplikasi baik indoor maupun outdoor. Komputer mini ini dapat dipasang (mounted) di belakang LCD monitor, sehingga tidak memakan tempat.

Perangkat eBox-4300 merupakan sebuah embedded CPU dengan *image* sistem operasi Window embedded CE 6.0R2 yang dikonfigurasi untuk dapat melakukan boot pada DOS dan melakukan fitur-fitur yang disediakan untuk dijalankan pada *image* OS CE 6.0.

### Billing System

Billing merupakan suatu software aplikasi yang membantu untuk memonitor dan menganalisa penggunaan komunikasi secara terperinci. Dalam sistem telepon informasi yang dicatat meliputi: tanggal, jam bicara, nomor telepon, kota dituju, lama bicara, biaya, extension, departement serta personnel dari setiap penggunaan telepon. Billing biasanya terdapat pada suatu PABX untuk merecord semua pemakaian telepon secara terperinci oleh extensionnya, atau pada warnet dengan merecord kegiatan semua ID user. Pada warnet billing digunakan hampir sama seperti pada PABX, dengan merecord semua detail panggilan masing-masing KBU warnet. Dengan billing ini akan membantu mengontrol pemakaian telepon.

### Warnet (Warung Telekomunikasi)

Warnet atau warung telekomunikasi adalah tempat yang disediakan untuk pelayanan jasa telekomunikasi untuk umum yang ditunggu baik bersifat sementara maupun tetap dan merupakan bagian dari telepon umum. Di dalam warnet terdapat Kamar Bicara Umum (KBU) berisi pesawat telepon untuk digunakan pemakai jasa dan juga dilengkapi display berupa tampilan durasi konektivitas serta biaya yang dibebankan. Telepon di dalam kamar umum bisa digunakan untuk pembicaraan telepon lokal, interlokal (SLJJ), maupun sambungan langsung internasional (SLI). Biaya pemakaian jasa telekomunikasi dibayar langsung di tempat oleh konsumen sesuai tarif yang berlaku ditambah tarif pelayanan.

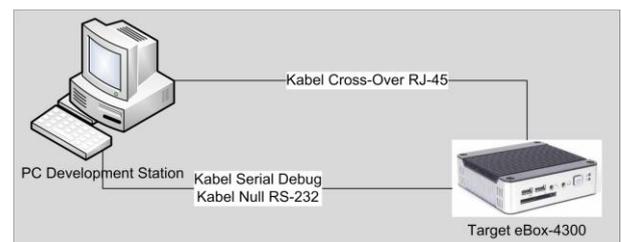
### Warnet (Warung Internet)

Warnet internet adalah salah satu jenis usaha yang menyewakan jasa internet kepada khalayak umum. Warnet dilengkapi dengan aplikasi billing untuk mencatat siapa yang masuk dan berapa lama pengguna memakai komputer sehingga dapat dihitung biaya yang harus dibayarkan. Namun disamping kelebihanannya tersebut, ternyata warnet tidak dapat menawarkan secara langsung untuk dapat berkomunikasi langsung atau bercakap-cakap dengan orang lain layaknya telepon maupun mengirim sms.

## 3. METODOLOGI

### Pembuatan *Image* Windows Embedded CE 6.0 R2

Untuk membuat *image* OS Windows Embedded CE 6.0 R2<sup>1</sup>, harus dibentuk terlebih dahulu lingkungan setup yang dibutuhkan. Dalam pembuatan dan pengembangan aplikasi yang akan dioperasikan oleh eBox-4300, dapat dikerjakan dengan tanpa layanan DHCP service seperti ditunjukkan pada gambar 3.1

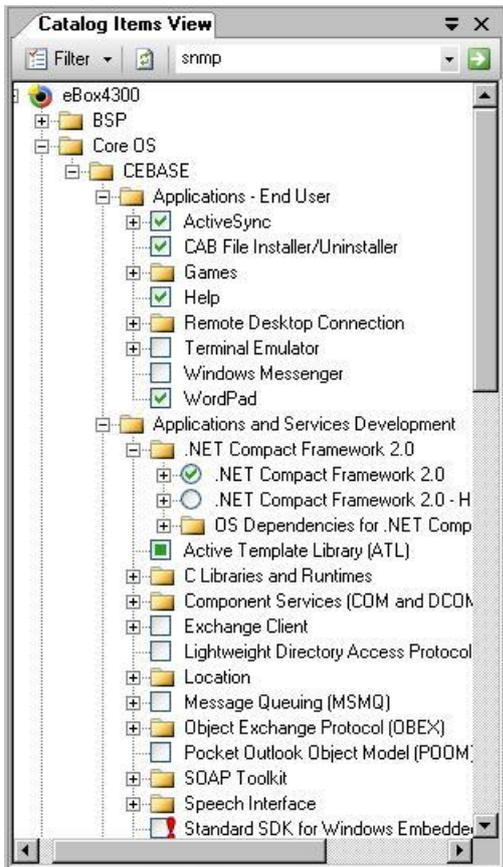


Gb 3.1. Development Environment Setup dengan layanan DHCP

Setelah lingkungan setup dibuat, dengan menggunakan project OS desain pada Visual Studio 2005 dapat dibuat konfigurasi *image* OS dengan kustomisasi atau penambahan dari fitur sistem yang dilakukan dengan menambahkan komponen katalog dari Platform Builder yaitu pada saat membangun *image*, penambahan library aplikasi dan menambahkan konfigurasi dari registri dari *image* yang telah dibentuk.

Konfigurasi penambahan aplikasi standart lainnya adalah USB Host Support/USB Storage Class Driver. Kemudian komponen .NET Compact Framework 2.0 juga dibutuhkan untuk mendukung pengaturan kode aplikasi Visual Studio 2005. Pada Core OS ini juga

diikuti item katalog File System and Data Storage yang terdiri dari RAM and ROM File System, untuk mendukung pembacaan data dari ROM dan RAM file system, juga Hive-based Registry, mendukung fungsi sistem untuk pengaturan registri saat power off.



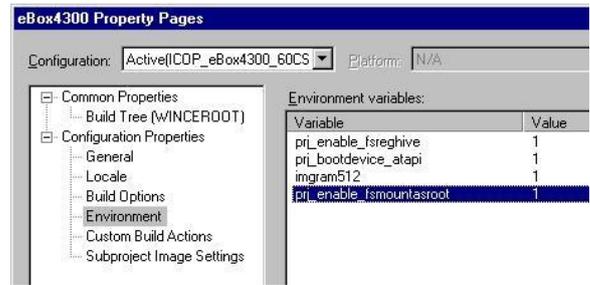
Gb 3.2. Detail Item Katalog pada Kustomisasi Fitur Image

Pemilihan debug yang digunakan adalah image versi release karena image akan menggunakan konfigurasi registri yang ada pada file image NK.BIN sebagai hasil compile yang di dalamnya juga terdapat registri untuk image, yaitu registri yang telah disetting sebelumnya ketika image dibangun.

Pengaturan berikutnya adalah desain properti OS yang dibutuhkan untuk mendukung Ethernet debug yang dijalankan pada saat booting oleh image OS. Juga untuk menggunakan Remote Tools, yaitu sebuah fitur tambahan untuk meremote hasil OS Desain yang telah dijalankan pada target eBox-4300. Maka pada konfigurasi desain OS, akan diaktifkan pilihan :

- Enable eboot space in memory (IMGEBOOT=1)
- Enable kernel debugger (no IMGDEBUGGER=1)
- Enable KITL (no IMGNOKITL=1)

Kustomisasi berikutnya adalah Environment Variables dari desain OS. Dengan membuat nama variabel baru untuk mendukung perangkat yang mempunyai memori RAM sebesar 512MB seperti eBox-4300, untuk menetapkan registri saat perangkat dimatikan, untuk mendesain perangkat dapat booting melalui penyimpanan IDE, juga untuk memungkinkan file sistem dapat me-mount sebagai root file sistem.



Gb 3.3. Halaman Properti Image pada Halaman Lingkungan Image

Tahap proses pembuatan Image OS CE 6.0 dapat memakan waktu 20 hingga 30 menit, tergantung dari kecepatan PC yang mendesain. Hasil image versi release OS dapat ditemukan pada direktori berikut:

C:\WINCE600\OSDesigns\eBox4300\Box4300\RelDir\ICOP\_eBox4300\_60CS\_x86\_Release

Image yang telah jadi kemudian akan diunduh ke dalam target eBox-4300 dengan mengatur menu Connectivity Options pada Visual Studio 2005 dan memilih Ethernet untuk pilihan Transport Download. Setelah eBox-4300 selesai mengunduh image OS nk.bin, eBox-4300 akan melakukan booting hingga selesai dan muncul layar pada desktop layar eBox-4300.



Gb 3.4. Layar pada eBox-4300 Ketika Image Telah Dijalankan

### Remote Tools

Untuk fitur penguncian dan keamanan proses aplikasi, maka dilakukan pengaturan pada registri agar toolbar akan autohide dan ketika image dijalankan maka secara otomatis aplikasi akan dijalankan<sup>2</sup>.

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Shell\AutoHide]
@=dword:1
```

```
[HKEY_LOCAL_MACHINE\init]
"Launch90"="\Startup\Billing.exe"
"Depend90"=hex:0A,00,14,00,1E,00
```

### Billing System dan CDR

Aplikasi billing sistem ini dibutuhkan untuk menghitung atau mengkonversi dari hasil aplikasi yang lain menjadi dalam bentuk nominal rupiah. Sebagai

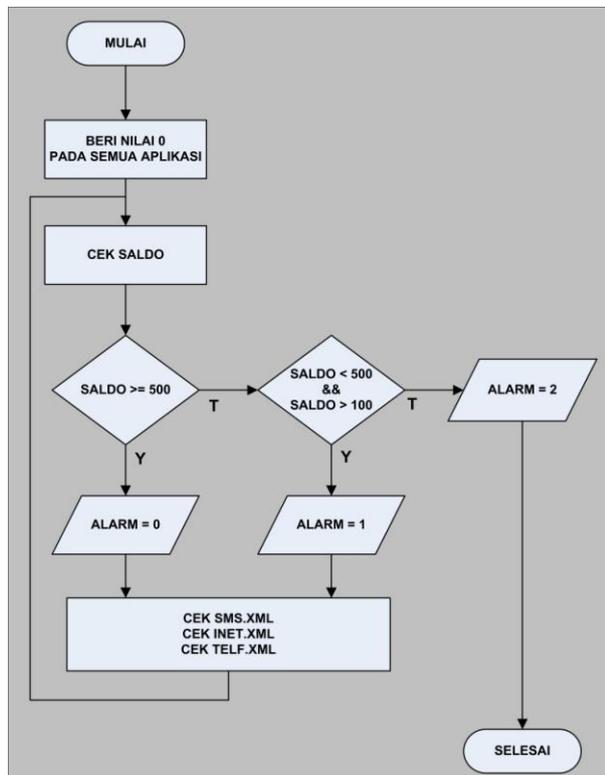
contoh pada aplikasi telepon, billing sistem akan mengecek file yang berisi durasi percakapan dan mengkonversinya kedalam bentuk nominal rupiah sehingga perhitungan saldo didapatkan.

Pada contoh yang lain yaitu pada aplikasi internet, aplikasi billing sistem mengkonversi jumlah byte paket *upstream* dan *downstream* sehingga mengetahui biaya yang akan dikalkulasikan untuk didapatkan sisa jumlah saldo. Begitu juga pada aplikasi SMS yaitu mendapatkan biaya SMS tersebut berdasar jumlah pulsa paket yang dikirimkan sehingga diketahui biaya SMS tersebut. Semua hasil aplikasi tersebut akan selalu dicatat pada CDR sebagai log history.

Aplikasi ini didesain dengan keakuratan yang stabil karena sistem harus bisa mengantisipasi kegiatan pengguna saat menjalankan aktivitas telepon, berinternet, maupun kirim sms. Aplikasi billing harus dapat menangani interproses ketiganya dalam Sistem Operasi Windows Embedded CE 6.0 R2. Bahasa pemrograman yang dibuat pada aplikasi serial ini adalah menggunakan pemrograman Visual C#.

Dalam mengantisipasi jumlah nominal saldo habis atau kegagalan sistem, digunakan suatu alarm. Alarm akan bernilai nol ketika jumlah nominal adalah aman yaitu lebih besar dari Rp.500,- sehingga semua aplikasi dapat dijalankan. Kemudian alarm akan bernilai satu ketika nominal saldo bernilai antara Rp.500,- sampai Rp.100,- maka aplikasi yang lain diharapkan mengirimkan warning pada pengguna.

Berikutnya alarm bernilai dua adalah ketika jumlah nominal saldo kurang dari jumlah minimum yaitu Rp.300,- sehingga semua aplikasi akan dihentikan. Selanjutnya jika sistem mengalami kegagalan maka alarm akan bernilai tiga dan semua aplikasi akan stop.



Gb 3.5. Flowchart Utama Aplikasi Billing

Sedangkan untuk mengatur harga dari pentarifan, aplikasi ini juga menyediakan suatu form untuk mengganti pentarifan yang ada. Form ini hanya bisa diakses dengan user admin. Pentarifan akan disesuaikan dengan level pengguna yang sedang menggunakan aplikasi ini. Terdapat tiga level user yaitu *free\_user*, *premium* dan *admin*. Gambar 3.6 merupakan form pentarifan oleh admin.

Gb 3.6. Form Pentarifan

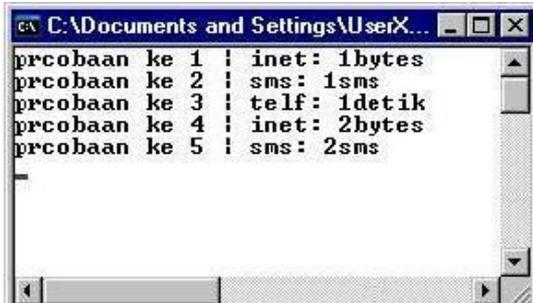
Dari form di atas, dapat dilihat suatu pentarifan untuk level pengguna *free\_user* maka pada form ini aplikasi telepon dilabelkan “Call Sesama Free” untuk panggilan sesama operator, dan “Call Bada Free” untuk pengguna *free\_user* dengan panggilan sesama operator. Sedangkan pada pengguna level premium akan dilabelkan “Call Sesama Premium” untuk panggilan sesama operator dan “Call Bada Premium” untuk panggilan beda operator. Sedangkan untuk aplikasi internet hanya disediakan satu tarif dengan label “Internet”. Begitu juga untuk tarif sms hanya satu tarif dengan label “SMS”.

### Pengujian dan Analisis

Setelah semua sistem dibuat, pengujian dilakukan berdasarkan beberapa parameter, diantaranya adalah kemampuan sistem dalam performa, ketangguhan sistem dalam menangani keragaman kemungkinan pada file yang dihasilkan dari aplikasi yang lain. Parameter yang lain adalah ketepatan waktu pada aplikasi billing untuk memproses paket-paket hasil aplikasi menjadi nilai nominal dan waktu yang dibutuhkan untuk memperbarui nilai saldo yang ada. Berikutnya adalah parameter ketepatan perhitungan atau ketelitian hasil perhitungan oleh aplikasi billing sistem. Juga *resource* yang dibutuhkan oleh *Image* ini untuk menjalankan seluruh aplikasi.

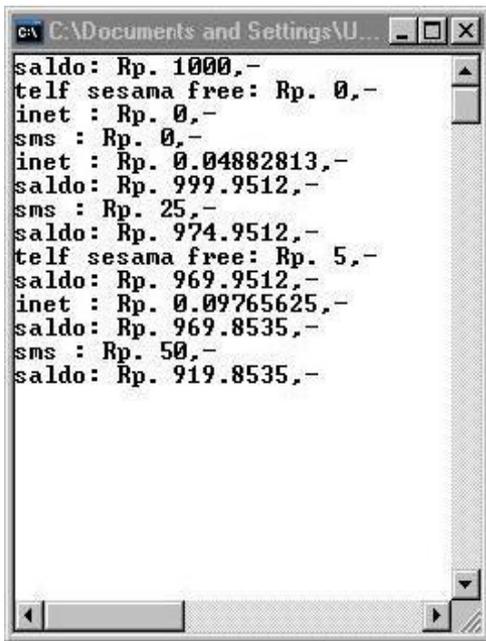
Untuk menguji aplikasi billing dan CDR, maka dibuatlah sebuah aplikasi yang secara berulang-ulang memproses dan memperbarui nilai yang ada pada file-file hasil aplikasi telepon, sms dan internet secara berkala. Pada aplikasi ini nilai sms yang ada akan diulang untuk diperbarui setiap sepuluh detik sekali dengan perulangan antara satu sms sampai dengan tiga sms. Selanjutnya pada aplikasi ini, nilai pada file hasil aplikasi telepon akan diulang antara nilai satu sampai sepuluh yang mempunyai arti file tersebut memiliki

nilai flat durasi antara satu detik sampai sepuluh detik. File ini akan selalu diulang untuk diperbarui setiap tujuh detik sekali. Sedangkan untuk file hasil aplikasi internet akan selalu diulang pada nilai satu byte hingga seratus byte dengan perulangan setiap lima detik sekali. Gambar 3.7 merupakan hasil dari aplikasi pengujian.



Gb 3.7. Aplikasi Pengujian

Ketika terjadi perubahan pada file-file tersebut, maka secara otomatis billing akan merespon dengan memproses perhitungannya. Gambar 3.8 merupakan respon dari aplikasi billing.



Gb 3.8. Respon Aplikasi Billing

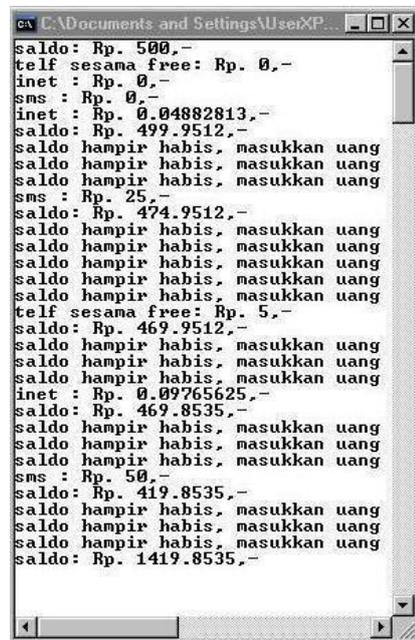
Terlihat pada setiap perhitungan billing akan menampilkan saldo terakhir setelah mengurangi dengan biaya dari hasil perhitungan billing. Jika dilihat lebih lanjut aplikasi billing ini telah mampu melakukan perhitungan dengan ketelitian hingga delapan digit di belakang koma. Namun untuk perhitungan pada saldo dan penulisan pada CDR, aplikasi ini mampu menuliskan empat digit di belakang koma. Gambar 3.9 merupakan hasil pencatatan pada CDR.



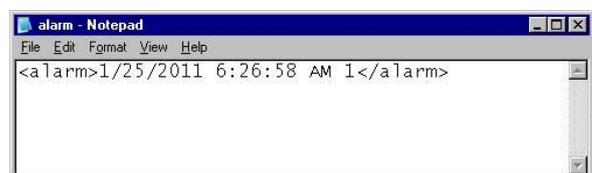
Gb 3.9. Pencatatan pada CDR Log History

Pada hasil pengujian didapatkan bahwa sistem ini telah dapat menangani keamanan dan kegagalan sistem seperti hilangnya salah satu file ataupun pengeditan data saldo. Semua proses aplikasi yang dijalankan juga mampu dicatat dalam CDR sehingga terdapat suatu history log.

Pengujian juga dilakukan pada saat saldo akan habis yaitu ketika alarm mempunyai nilai satu pada saat saldo diantara Rp. 500,- hingga Rp. 100,-. Pada gambar 3.10 berikut adalah salah satu contoh ketika billing mendeteksi bahwa saldo telah mencapai range tersebut kemudian pengguna memasukkan uangnya kembali sehingga nilai nominal saldo menjadi diperbarui kembali.

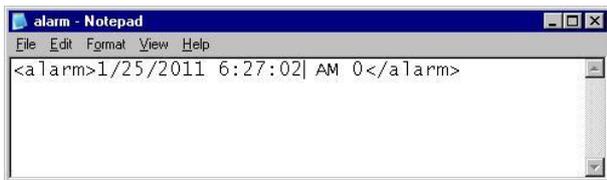


Gambar 3.10. Alarm Bernilai 1 pada Billing



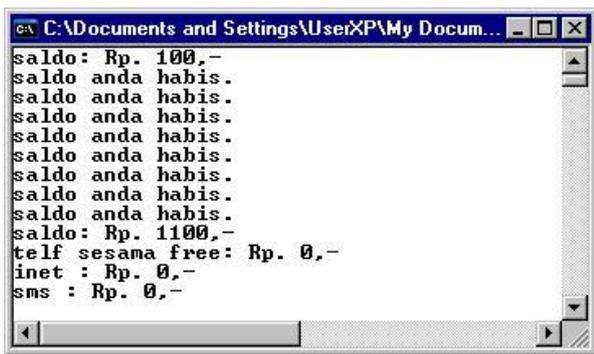
Gambar 3.11. Alarm Bernilai 1 pada File alarm.xml

Dari gambar 3.10 di atas dapat dilihat bahwa ketika nilai saldo kurang dari Rp. 500,- maka akan diberlakukan alarm bernilai satu, ditunjukkan pada gambar 3.11 sedangkan pada aplikasi billing dituliskan pesan “saldo hampr habis, masukkan uang”. Berikutnya setelah uang kembali dimasukkan maka saldo akan diperbarui yang pada hal ini pengguna memasukkan uang sebesar Rp. 1000,- sehingga saldo akan diperbarui menjadi Rp. 1419.8535,- dan alarm akan kembali menjadi bernilai nol yaitu ditunjukkan pada gambar 3.10 dan gambar 3.12.

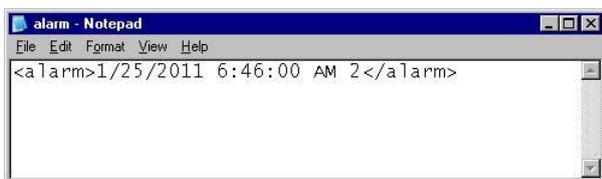


**Gambar 3.12.** Alarm Kembali Bernilai 0 Setelah Bernilai 1

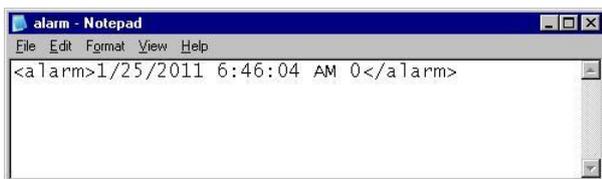
Berikutnya adalah pesan yang ditunjukkan oleh billing ketika saldo telah kurang dari saldo minimal yaitu kurang atau sama dengan Rp. 100,-. Gambar 3.13 dan gambar 3.14 adalah penunjukan ketika alarm bernilai dua muncul dan pada billiing terdapat pesan “saldo anda habis.”.



**Gambar 3.13.** Saldo Kurang dari Saldo Minimal dan Saldo Updated



**Gambar 3.14.** Alarm Bernilai 2

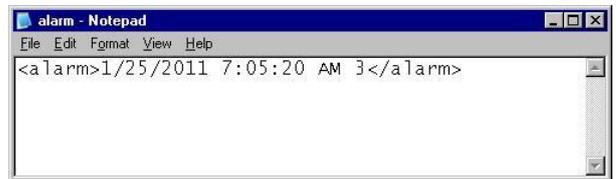


**Gambar 3.15.** Alarm Kembali Bernilai 0 Setelah Bernilai 2

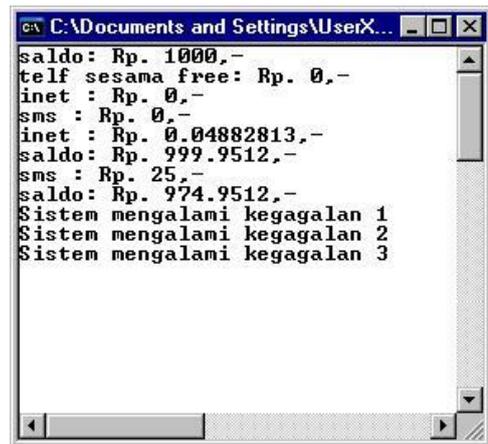
Sedangkan pada gambar 3.15 merupakan penunjukan setelah pengguna memasukkan uang kembali sehingga nilai saldo telah mencukupi dari saldo minimal sehingga nilai pada alarm akan kembali bernilai nol.

Berikutnya adalah aksi dari aplikasi billing ketika terjadi kegagalan sistem yang pada hal ini dimisalkan

terdapat file yang hilang. Ketika kondisi ini muncul maka nilai alarm akan bernilai tiga yang ditunjukkan pada gambar 3.16. dan pada gambar 3.17 merupakan gambar billing ketika terjadi kegagalan yang menunjukkan pesan “sistem mengalami kegagalan”.



**Gambar 3.16.** Alarm Bernilai 3



**Gambar 3.17.** Sistem Gagal

Performa dari pengujian ini ditentukan pada waktu delay yang dibutuhkan sistem untuk memproses semua kegiatan billing. Berikut adalah tabel 3.1 sebagai penunjukan analisa performa ketika pengguna memasukkan uang meskipun pada saat itu alarm bernilai nol.

**Tabel 3.1.** Update Nilai Saldo

	percobaan ke	update saldo(detik)
waktu (detik)	1	0,56
	2	0,6
	3	0,65
	4	0,62
	5	0,58
	6	0,51
	7	0,58
	8	0,68
	9	0,59
	10	0,61
rata-rata (detik)		0,598

Dari tabel 3.1 di atas dapat dilihat bahwa waktu rata-rata delay yang dibutuhkan sistem untuk memperbarui nilai saldo meskipun pada saat itu alarm bernilai nol adalah 0,598 detik.

Berikutnya adalah tabel 3.2 yang merupakan hasil pengujian waktu yang dibutuhkan sistem ketika alarm dari bernilai nol menjadi bernilai satu karena saldo telah mencapai range tidak aman dan ketika pengguna

memasukkan uangnya kembali sehingga alarm kembali bernilai nol.

**Tabel 3.2.** Update Nilai Saldo dari Alarm 0 ke 1 dan Kembali ke 0

	percobaan ke	alarm 1 (detik)	alarm kembali 0 (detik)
waktu (detik)	1	0,46	4,35
	2	0,59	4,8
	3	0,5	4,32
	4	0,54	4,65
	5	0,49	4,56
	6	0,5	5,12
	7	0,53	4,68
	8	0,52	4,97
	9	0,58	4,87
	10	0,48	5,16
rata-rata (detik)		0,519	4,748

Dari tabel 3.2 di atas dapat dilihat bahwa waktu rata-rata yang dibutuhkan sistem untuk memperbarui nilai saldo ketika alarm berubah dari nilai nol menjadi nilai satu adalah 0,519 detik. Sedangkan pada saat pengguna memasukkan uangnya kembali maka rata-rata waktu yang dibutuhkan sistem untuk memperbaruinya adalah 4,748 detik.

Selanjutnya pada tabel 3.3 di bawah menunjukkan waktu yang dibutuhkan sistem ketika alarm berubah nilainya dari bernilai satu yaitu ketika pengguna mempunyai saldo pada range tidak aman menjadi nilai dua karena saldo telah di bawah saldo minimal dan menjadi nilai nol yaitu ketika pengguna memasukkan uangnya kembali.

**Tabel 3.3.** Update Nilai Saldo dari Alarm 1 ke 2 dan Kembali ke 0

	percobaan ke	alarm 2 (detik)	alarm kembali 0 (detik)
waktu (detik)	1	0,57	4,23
	2	0,55	4,84
	3	0,46	4,65
	4	0,43	4,72
	5	0,48	4,64
	6	0,59	4,97
	7	0,45	4,59
	8	0,42	5,01
	9	0,68	4,56
	10	0,59	5,16
rata-rata (detik)		0,522	4,737

Dari tabel 3.3 di atas dapat dilihat bahwa waktu rata-rata yang dibutuhkan sistem untuk memperbarui nilai alarm dari bernilai satu ke nilai dua adalah 0,522 detik. Sedangkan ketika pengguna memasukkan uangnya kembali maka waktu rata-rata yang dibutuhkan sistem untuk memperbarui kembali ke nilai nol adalah 4,737 detik.

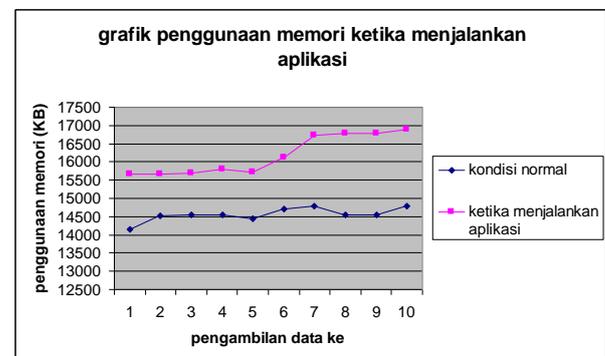
**Tabel 3.4.** Update Nilai Saldo dari Alarm 1 ke 2 dan Kembali ke 0

	komputasi (detik)		dari masukan (detik)	
waktu (detik)	alarm dari 0 ke 1	0,519	alarm dari 1 ke 0	4,748
	alarm dari 1 ke 2	0,522	alarm dari 2 ke 0	4,737
	rata-rata (detik)	0,5205	rata-rata (detik)	4,7425

Berdasarkan serangkaian proses pengujian yang telah dilakukan maka dapat ditabelkan perbandingan waktu rata-rata yang dibutuhkan sistem yaitu pada tabel 3.4 ketika billing memperbarui nilai saldo dari perhitungan dirinya sendiri rata-rata waktu yang dibutuhkan adalah 0,5205 detik sedangkan waktu rata-rata yang dibutuhkan untuk memperbarui nilai saldo berdasarkan pembacaan file hasil input saldo dari pengguna adalah 4,7425 detik.

Sedangkan analisis untuk *Image* yang digunakan dapat menggunakan Remote Tools System Information dapat dilihat secara umum sistem informasi dari perangkat target eBox-4300, termasuk nama sistem operasi, nama dan model CPU juga menampilkan informasi tentang resource atau sistem memori yang digunakan.

Sehingga dapat dianalisis memori yang digunakan ketika aplikasi berjalan. Gambar grafik 3.18 menunjukkan penggunaan memori ketika aplikasi serial berjalan dan juga saat memproses aplikasi billing dan CDR. Yang kemudian penggunaan dari memori tersebut akan dibandingkan dengan memori pada saat kondisi normal.



**Gambar 3.18.** Penggunaan Memory pada *Image* Windows Embedded CE

Pada percobaan yang dilakukan di atas, pengambilan data kesatu sampai dengan kelima adalah ketika *Image* hanya memproses aplikasi komunikasi serial dan billing utama tanpa memproses aplikasi telepon, internet dan sms. Dan ketika pengambilan data keenam sampai kesepuluh, *Image* dicoba untuk menjalankan seluruh aplikasi yang ada beserta fitur-fiturnya. Yaitu menerima komunikasi serial, memproses / menghitung data, memproses billing, melakukan management aplikasi dan membuat log file CDR. Sehingga dapat dilihat bahwa pada pengambilan data keenam sampai kesepuluh, memori yang digunakan *Image* akan meningkat.

#### 4. KESIMPULAN

##### Kesimpulan

Setelah melakukan serangkaian proses pengujian terhadap system yang telah dibuat, maka dapat diambil kesimpulan bahwa:

1. File nk.bin hasil kompilasi *Image* yang telah dilakukan hanya sebesar 19.1 MB. *Image* ini mampu menangani komunikasi serial, *User Interface Metrics* dengan resolusi layar 800 x 600, mampu menangani I/O 10/100 Mbps LAN, mengenali CF Card sebagai media penyimpanan *hard Disk*, dan mampu mengoperasikan aplikasi berbasis *Audio Interface*.
2. Kustomisasi image OS yang dibuat dengan Platform Builder yang ada pada Visual Studio 2005 hanya memakan memori yang kecil untuk dijalankan. Yaitu kondisi normal hanya membutuhkan sekitar 15MB dan ketika mejalankan proses aplikasi hanya membutuhkan memori sekitar 16.5MB.
3. Aplikasi billing dan management sistem telah mampu menangani keragaman pengguna dalam menjalankan aplikasi sistem dan kegagalan sistem. Waktu rata-rata yang dibutuhkan sistem untuk memproses perhitungan dan memperbarui datanya berdasarkan perhitungan aplikasi adalah 0,5205 detik sedangkan waktu rata-rata yang dibutuhkan untuk memperbarui nilai saldo berdasarkan masukan dari pengguna adalah 4,7425 detik.
4. Aplikasi billing yang dijalankan telah mampu untuk menangani inter proses antar ketiga aplikasi yang lain yaitu aplikasi telepon, aplikasi internet dan aplikasi sms dengan record dalam sebuah CDR dengan ketelitian perhitungan delapan digit di belakang koma untuk billing dan empat digit di belakang koma untuk pencatatan dalam CDR.

##### Saran

Dalam pembuatan *Image* ini masih jauh dari kesempurnaan, ada beberapa hal yang harus diperbaiki untuk mencapai kesempurnaan sebuah Sistem Operasi.

1. Menambahkan fitur SNMP pada *Image* untuk menjalankan aplikasi internet sehingga lebih mudah untuk mengambil jumlah besar data paket yang lewat.
2. Menambahkan interface pada CDR dalam bentuk grafik sehingga lebih mudah untuk dianalisis.

Algoritma yang lebih mudah atau ringkas pada perhitungan billing sehingga didapatkan waktu yang relatif lebih singkat untuk membaca masukan dari pengguna.

#### 5. DAFTAR PUSTAKA

1. Phung Samuel, "eBox-4300 Windows Embedded CE 6.0 Jump Start Kit", ICOP Technology Inc, 18 Pebruari 2008.
2. Tim Fakultas Teknik Universitas Negeri Yogyakarta, "Teknik Operasional Perangkat Wartel", Direktorat Jenderal Pendidikan Dasar dan Menengah Departemen Pendidikan Nasional, Desember 2003.

3. Wandani. SH Sulistiyani, "Perjanjian Penyelenggaraan Warung Telekomunikasi (Wartel) antara PT.Telkom dengan Pihak Pengelola Wartel di Kab. Kendal", Program Pasca Sarjana Universitas Diponegoro, Semarang 2003.
4. Phung Samuel, "Windows CE - Launch Application at Startup", [eMVP], Desember 2006.
5. Phung Samuel, "Windows CE 6.0 – Hive-Based Registry", [eMVP], 4 Juli 2009.
6. Ahmad Khadafi Sanu, "Perancangan dan Pembuatan Aplikasi Data Akusisi Informasi Kondisi Hutan dengan Sistem Komunikasi 802.15.4 pada Perangkat Embedded eBox-4300", Politeknik Elektronika Negeri Surabaya PENS-ITS, Surabaya 2009,
7. Achmad Subhan KH, Fajar Baskoro, Gilang Kharisma, Ahmad Khadafi, "Sistem Pemantau Kebakaran Hutan dan Pembalakan Liar Hutan Menggunakan Perangkat Embedded Server eBox-4300 dan Jaringan Sensor Nirkabel 802.15.4", Prosiding IES<sup>10th</sup> PENS-ITS, Surabaya 2009.