

E-STATEMENT PADA SMARTPHONE BLACKBERRY

Samuel Mahatma Putra, Hartini Deliana, Rosmiyani, Fernando Halim

ABSTRAKS

Tujuan dari penelitian ini ialah menganalisa dan membangun aplikasi pada smartphone BlackBerry untuk menangani penerimaan tagihan dalam bentuk *e-Statement*, membangun aplikasi pada pihak server perusahaan untuk mengirimkan informasi tagihan, membangun aplikasi pada pihak server *third party* sebagai perantara antara server perusahaan dengan smartphone BlackBerry dalam pengiriman tagihan, mengurangi waktu pemrosesan dan pengiriman tagihan serta mengurangi pemakaian kertas. Metode penelitian yang digunakan ialah metode analisis dan metode perancangan. Metode analisis meliputi survei atas sistem yang telah ada, melakukan wawancara dan menyebarkan kuesioner, mempelajari dokumen perusahaan dan sistem yang telah ada, dan analisis terhadap hasil wawancara dan dokumen perusahaan. Metode perancangan yang digunakan adalah *Object Oriented Analysis Design (OOAD)* melalui pendekatan proses rekayasa perangkat lunak model *Waterfall* dan menggunakan *Unified Modeliing Language (UML)*, yaitu *Use Case Diagram*, *Class Diagram*, *Sequence Diagram*, dan *State Diagram*, serta perancangan *ERD*. Hasil yang dicapai ialah sistem pengiriman tagihan yang menggunakan aplikasi mobile *e-Statement* pada smartphone BlackBerry. Simpulan yang dicapai ialah penerima tagihan dapat melihat seluruh tagihannya pada satu aplikasi, waktu pengiriman tagihan yang lebih cepat, menyediakan layanan pengiriman tagihan yang lebih cepat, mengurangi kelalaian pembayaran tagihan, memberikan kemudahan untuk melakukan *complaint* atas ketidaksesuaian informasi tagihan.

Kata Kunci : pengiriman, tagihan, *e-Statement*, BlackBerry, *statement*

I. PENDAHULUAN

Pada masa dewasa ini, pengiriman tagihan seperti tagihan kartu kredit, tagihan listrik (PLN), tagihan air (PAM), dan tagihan lainnya masih menggunakan jasa kurir (via pos) atau dapat disebut sebagai sistem manual karena tagihan masih menggunakan kertas tagihan yang dihasilkan oleh sistem dan diberikan kepada kurir untuk diantarkan ke alamat tagihan. Lamanya pengiriman pun ada yang dalam beberapa hari saja sejak akhir bulan pemakaian dan bahkan ada tagihan yang tidak sampai ke pelanggan. Akibatnya pelanggan yang tidak mendapat tagihan pun lupa membayar karena tidak mendapat kiriman tagihan dan berdampak pelanggan dikenakan sanksi berupa denda ataupun sanksi lainnya. Pengiriman tagihan ini menggunakan sangat banyak kertas dengan banyaknya pelanggan dan banyaknya tagihan yang diterima setiap pelanggan untuk setiap bulannya. Cara pengiriman tagihan lainnya adalah pengiriman tagihan dalam bentuk *email*. Tetapi masih sedikit perusahaan yang menggunakan *email* untuk mengirimkan tagihan kepada pelanggan dikarenakan alasan keamanan ataupun masalah dimana pelanggan melakukan *complaint* bahwa pelanggan tidak menerima tagihan berupa *email* padahal perusahaan sudah melakukan pengiriman.

Seiring dengan berkembangnya teknologi komunikasi, salah satu teknologi yang berkembang dengan pesat saat ini adalah teknologi *Push-API* pada smartphone BlackBerry, maka akan lebih efisien jika tagihan dikirim menggunakan *e-Statement* yang dibangun pada smartphone BlackBerry, dimana *e-Statement* adalah layanan *online* bagi pelanggan sehingga setiap pelanggan dapat menerima tagihan tanpa menggunakan kertas

dan proses pengiriman yang lebih cepat dibandingkan sistem yang manual.

II. LANDASAN TEORI

A. Pengenalan smartphone BlackBerry

BlackBerry adalah perangkat genggam nirkabel yang memiliki kemampuan layanan *push e-mail*, telepon selular, sms, faksimili *internet*, menjelajah *internet*, dan berbagai kemampuan nirkabel lainnya[1]. Gambar 2.1 adalah salah satu model BlackBerry, yaitu BlackBerry 8900 atau Javelin.



Gambar 2.1 Device BlackBerry (8900)

BlackBerry menyediakan berbagai perangkat lunak yang dapat disesuaikan dengan kebutuhan operasi, yaitu :

1. BlackBerry Enterprise Server (BES)

Perangkat genggam BlackBerry terintegrasi pada sistem *email* yang terorganisasi melalui paket perangkat lunak yang disebut BlackBerry Enterprise System (BES). BES dapat digunakan oleh jaringan *email* yang berbasis *Microsoft Exchange*, *Lotus Domino*, dan *Novell Group Wise*. Khusus pada

pengguna individu, mereka dapat menggunakan layanan *email* nirkabel yang disediakan oleh *provider* tanpa harus menginstalasi BES. Para pengguna individu dapat menggunakan BlackBerry *Internet Solution* tanpa harus menginstalasi BES di *smartphone* mereka. BES memang ditujukan bagi pelanggan korporasi dengan cakupan usaha yang besar. Perangkat lunak ini mengintegrasikan seluruh *smartphone* BlackBerry pada suatu organisasi dengan sistem perusahaan yang telah ada. Pada C. setiap *smartphone* BlackBerry mempunyai ID yang dinamakan PIN, yang digunakan sebagai pengenalan BlackBerry pada BES. Keuntungan yang diperoleh adalah memperluas komunikasi nirkabel dan data perusahaan kepada pengguna aktif dengan cara yang aman.

2. BlackBerry Professional Software (BPS)

BPS merupakan komunikasi nirkabel dan kolaborasi solusi bagi usaha kecil dan menengah. Ia menghadirkan berbagai fitur yang dibutuhkan karyawan, dalam sebuah paket yang mudah dipasang dan harga yang lebih murah.

3. BlackBerry Internet Service (BIS)

Perangkat lunak yang diperuntukkan bagi pengguna pribadi ini memungkinkan Anda untuk mengintegrasikan *smartphone* dengan 10 akun *e-mail* yang berbasis *Post Office Protocol* (POP3) dan *Internet Message Access Protocol* (IMAP), menerima dan mengirim pesan instan, serta berselancar di *internet*. Dengan BIS, kita juga dapat membuka tambahan data (*attachment*) dalam bentuk excel, word, powerpoint, pdf, zip, jpg, gif dengan tingkat kompresi data yang tinggi.

4. BlackBerry Mobile Data System (BlackBerry MDS)

Sebuah aplikasi optimisasi pengembangan kerangka kerja untuk BlackBerry *Enterprise Solution*, menyediakan Anda sebuah alat pengembangan untuk membangun, menyebarluaskan, serta mengatur interaksi antara *smartphones* BlackBerry dan aplikasi perusahaan.

B. Java pada *smartphone* BlackBerry

Smartphone BlackBerry dibangun sebagai perangkat berbasis Java. Semua aplikasi pada *smartphone* ini dibangun dari Java® ME. Semua *smartphone* BlackBerry mendukung setidaknya MIDP 1.0 dan CLDC 1.0, dan *smartphone* yang menggunakan BlackBerry *Device Software* v4.0 atau lebih tinggi MIDP 2.0/CLDC1.1.

Java API pada *smartphone* BlackBerry memungkinkan untuk mengembangkan aplikasi-aplikasi *client* yang menyediakan:

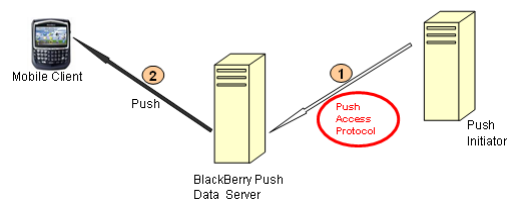
1. Fitur *customize user interface*
 2. Penyimpanan data lokal pada perangkat
 3. *Event listening* dan *system interfaces*
 4. Transportasi nirkabel aman melalui HTTP dan TCP
 5. Jangkauan jaringan dan dukungan *seamless roaming*
- Research In Motion (RIM) telah mengembangkan *world-class development tools* untuk membantu

pengembangan aplikasi BlackBerry dengan mudah. *Tools* ini untuk para pengembang tanpa dikenakan biaya dan terus-menerus diperbarui untuk menyertakan fitur terbaru. Salah satu *tools* yang disediakan oleh RIM adalah BlackBerry JDE Plugin untuk Eclipse yang memungkinkan para pengembang untuk membangun aplikasi Java pada BlackBerry dengan menggunakan Eclipse.

C. Layanan *Push* dengan BlackBerry *Push Infrastructure*

Push memungkinkan pengiriman konten (data) ke BlackBerry tanpa BlackBerry harus memintanya. Gambar 2.2 menunjukkan infrastruktur dari *push*. Data dikirim ke sebuah *port* pada BlackBerry di mana aplikasi selalu mendengarkan (*listening*) datangnya data (*event*) [2].

Push Initiator mengajukan permintaan kepada *Hosted Data Push Service* (*BlackBerry Push Data Server*). Layanan *push* berisi instruksi pengiriman (*delivery instruction*) dan *payload*. Instruksi pengiriman berisi kemana dan bagaimana data dikirim. *Payload* dapat berisi data apa saja khususnya data yang dikirim.



Gambar 2.2 *Push Infrastructure*

Menurut RIM, keuntungan dari pemakaian BlackBerry *Push-API* adalah sebagai berikut :

1. Informasi yang cepat
Push data ke *smartphone* BlackBerry merupakan cara yang paling efisien untuk mendapatkan informasi untuk pengguna *smartphone* BlackBerry. Dengan BlackBerry *Push Application Programming Interface* (API), data dikirimkan ke aplikasi pada *smartphone* BlackBerry

2. Penghematan biaya

Aplikasi yang dikembangkan dengan menggunakan BlackBerry *Push-API* tidak perlu berulang kali melakukan *poll server* untuk data baru. Meskipun secara individual *polling request* dan *response* adalah kecil, namun biaya dapat bertambah dengan cepat seiring dengan adanya *polling* beberapa aplikasi yang digunakan.

BlackBerry *Push-API* ini dirancang untuk mem-*push* sejumlah data. Penyedia konten dapat mendorong hingga 8 KB data untuk *smartphone* BlackBerry dengan mem-*push* data dari setiap permintaan. Jika banyak data yang tersedia, pengguna dapat memutuskan apakah akan mengambil atau tidak, dengan adanya ukuran kontrol atas bagaimana data dikirim dan dibayar.

3. Penurunan *Network Latency*

Aplikasi yang menggunakan BlackBerry *Push-API* juga dapat mengurangi dampak dari *network latency* (*delay* yang dihasilkan ketika *me-request* / mengirim paket). BlackBerry *Push-API* memberikan data ke *smartphone* BlackBerry dari *background* tanpa keterlibatan pengguna. Data terbaru akan disinkronisasikan dan tersedia pada *smartphone* BlackBerry mereka saat mereka membuka aplikasi.

4. Pemakaian baterai yang lebih tahan lama

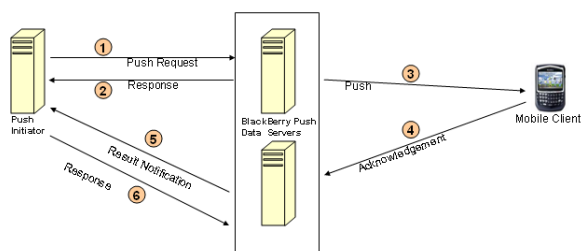
BlackBerry *Push-API* meminimalkan dampak pada pemakaian baterai. Dibanding terus-menerus memeriksa adanya data baru, aplikasi hanya perlu berada pada *background* untuk mendengarkan server untuk mem-*push data*. Setelah data sampai, aplikasi dapat memproses data sebagaimana diperlukan.

5. Sedikit atau tidak adanya *Learning Curve*

Pengembang yang sudah akrab dengan standar *Password Authentication Protocol* (PAP) sudah tahu semua yang diperlukan untuk mengakses BlackBerry *Push Technology*. Banyak kompleksitas dari *pushed-based data delivery service* yang tersembunyi karena BlackBerry *Infrastructure* yang melakukan proses pengelolaan dan proses pengiriman data.

Cara kerja BlackBerry Push-API untuk Java Application seperti pada Gambar 2.3:

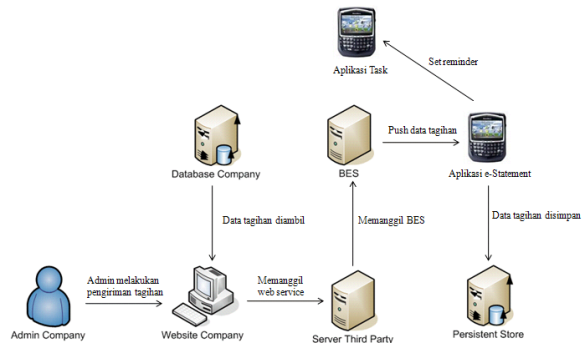
1. Penyedia konten mengirimkan permintaan *push*
2. BlackBerry *Infrastructure* mengembalikan tanggapan (*response*)
3. BlackBerry *Infrastructure* melakukan *push data* ke *smartphone* BlackBerry
4. *Smartphone* BlackBerry mengembalikan tanggapan (*response*) terhadap BlackBerry *Infrastructure*
5. BlackBerry *Infrastructure* mengirimkan pemberitahuan ke penyedia konten
6. Penerimaan pemberitahuan dikembalikan ke BlackBerry *Infrastructure*
- 7.



Gambar 2.3 Cara kerja *push*

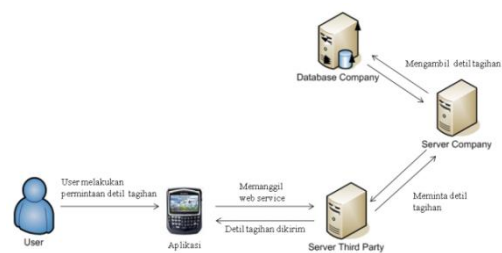
III. SOFTWARE DEVELOPMENT

A. Model Konseptual



Gambar 3.1 Proses pengiriman tagihan

Gambar 3.1 merupakan proses pengiriman tagihan. Setiap data tagihan akan disimpan pada *database* perusahaan yang akan dikirim per bulan sesuai dengan periode tagihan tersebut. Data tagihan akan dikirimkan berdasarkan PIN (*Personal Identification Number*) BlackBerry yang unik untuk setiap perangkatnya. Pihak admin perusahaan mengirimkan data tagihan, tetapi tidak langsung ke perangkat BlackBerry, melainkan melalui BES (*BlackBerry Enterprise Server*). BES yang berupa BlackBerry *Push Data Server* disediakan oleh pihak *server third party* yang memberikan akses *web service* kepada pihak admin perusahaan untuk melakukan *push*. BES akan mengirimkan data tagihan tersebut menggunakan BlackBerry *Push-API* sehingga aplikasi pada perangkat *smartphone* BlackBerry tidak perlu melakukan *poll server* untuk mendapatkan data terbaru. Teknologi BlackBerry *Push-API* ini akan memberikan data tanpa keterlibatan pengguna, sehingga ketika data baru diterima oleh perangkat, maka data tersebut akan segera disinkronisasikan dan tersedia pada *smartphone* BlackBerry pengguna saat mereka membuka aplikasi. Penyimpanan data tagihan pada perangkat *smartphone* BlackBerry menggunakan *Persistent Store*. *Persistent Store* pada *smartphone* BlackBerry dirancang untuk menyediakan tempat penyimpanan yang fleksibel. Dengan menggunakan *Persistent Store*, BlackBerry dapat menyimpan objek Java ke *memory* tanpa harus *serialize* data tersebut, dan begitu aplikasi dibuka, data-data yang disimpan pada *Persistent Store* tersebut dapat diambil dan ditampilkan kembali. Aplikasi juga memanggil aplikasi *task* untuk membuat *reminder* pembayaran tagihan.

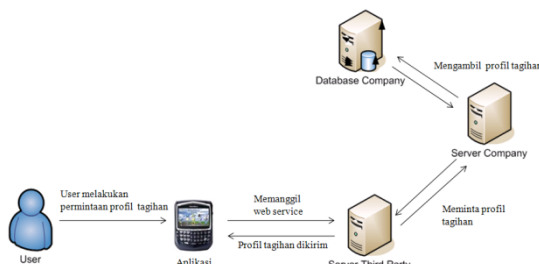


Gambar 3.2 Proses permintaan detail tagihan



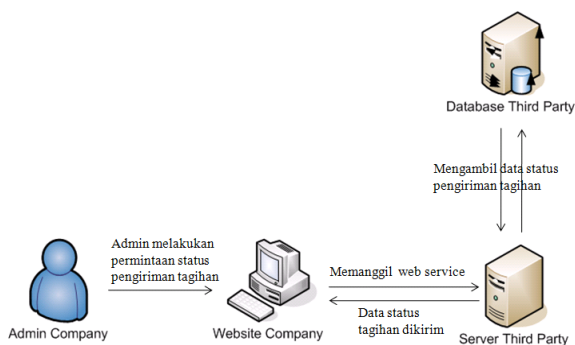
Gambar 3.3 Proses *complaint* tagihan

User dapat melihat detail tagihan yang telah dikirim ke BlackBerry dengan melakukan permintaan detail tagihan, seperti pada Gambar 3.2. Setelah user melakukan permintaan detail tagihan dengan memilih detail tagihan yang ingin ditampilkan pada aplikasi dan melakukan login, aplikasi akan melakukan pull data dari server third party. Server third party akan meminta data dari server company. Server company akan mengambil data detail tagihan dari database. Setelah permintaan data dari server company ke server third party berhasil, maka data dikirim ke BlackBerry user. Jika tagihan tidak sesuai user dapat mengirim email complaint dengan memilih menu Send Email pada aplikasi, seperti pada Gambar 3.3.



Gambar 3.4 Proses permintaan profil tagihan

User dapat melihat profil tagihan dengan melakukan permintaan profil tagihan seperti pada Gambar 3.4. Setelah user melakukan permintaan profil tagihan dengan aplikasi aplikasi akan melakukan pull data dari server third party. Server third party akan meminta data dari server company. Server company akan mengambil data detail tagihan dari database. Setelah permintaan data dari server company ke server third party berhasil, maka data dikirim ke BlackBerry user.

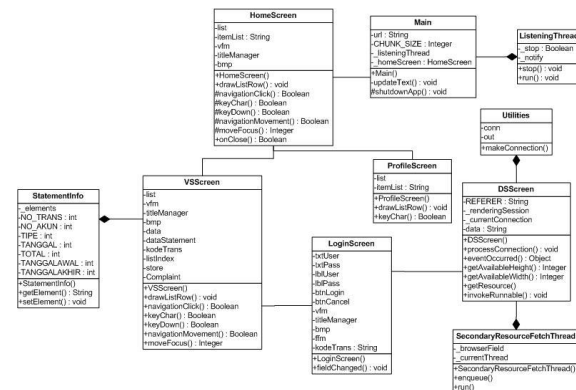


Gambar 3.5 Proses permintaan status pengiriman tagihan

Admin perusahaan dapat melihat status pengiriman tagihan dengan melakukan permintaan status tagihan kepada server third party, seperti pada Gambar 3.5. Setelah admin perusahaan melakukan permintaan status pengiriman tagihan dengan memanggil web service third party, server third party akan mengambil data dari database third party dan dikirim ke admin perusahaan.

Design Model

Aplikasi yang dibangun digambarkan dengan menggunakan Class Diagram, seperti pada Gambar 3.5.



Gambar 3.5 Class Diagram

C. Kebutuhan Perangkat Lunak

Untuk mengimplementasikan rancangan aplikasi dan basis data yang telah dibuat, diperlukan perangkat lunak dengan spesifikasi tertentu.

1. Sisi Client

Sistem operasi BlackBerry (BlackBerry OS)

2. Sisi Server

Adapun spesifikasi perangkat lunak minimum yang dibutuhkan admin perusahaan untuk menjalankan aplikasi adalah sebagai berikut:

- Windows Server 2003 atau Windows XP Professional x64 Editon.
- IIS 6.0.
- SQL Server 2005.
- Web Browser, Internet Explorer 6.0 atau Mozilla Firefox 2.0.

Adapun spesifikasi perangkat lunak minimum yang dibutuhkan admin third party untuk menjalankan aplikasi adalah sebagai berikut:

- Windows Server 2003 SP 1.
- Microsoft Exchange 2003 System Manager SP 1.
- BlackBerry Enterprise Server Version 4.1 untuk Microsoft Exchange.
- Microsoft XML Version 4.0 SP 2 (MSXML).
- Java™ Runtime Environment, SE 1.4.2_03.
- IIS 6.0.
- SQL Server 2005.

D. Kebutuhan Perangkat Keras

1. Sisi Client

Adapun spesifikasi *hardware* minimum yang dibutuhkan *client* untuk menjalankan aplikasi adalah sebagai berikut:

- *Smartphone* BlackBerry, terutama BlackBerry seri 8900 atau biasa disebut BlackBerry Javeline.
- Layanan BIS (*BlackBerry Internet Service*) sebagai koneksi *internet*.

2. Sisi *Server*

Adapun spesifikasi *hardware* minimum yang dibutuhkan *admin* perusahaan untuk menjalankan aplikasi adalah sebagai berikut:

- CPU 1.6 GHz, jika pada Windows Vista: CPU 2.4 GHz.
- RAM 384 MB, jika pada Windows Vista: RAM 768 MB.
- Monitor dengan *resolusi display* 1024 x 768.
- Hard Disk 5400 RPM.

Adapun spesifikasi *hardware* minimum yang dibutuhkan *admin third party* untuk menjalankan aplikasi adalah sebagai berikut:

- *Dual PIV* – 2.8 GHz *processor* (*Hyper Threading* dimatikan).
- Memori 3920 MB.
- HP NC7781 Gigabit Server Adapter.
- 236 – GB (15.000 RPM) drives pada RAID 0 Array dengan Smart Array 5i Controller.

IV. EVALUASI

1. Uji coba waktu pengiriman tagihan

Percobaan dilakukan dengan *admin company* mengirim 5 tagihan ke user, seperti pada Tabel 4.1.

Tabel 4. 1 Hasil Uji coba waktu pengiriman tagihan

Waktu Tagihan Dikirim oleh Admin	Waktu Tagihan Diterima Aplikasi e-Statement	Lama Proses
7 Januari 2010 pk 16:22:10	7 Januari 2010 pk 16:22:13.13	3.13 detik
7 Januari 2010 pk 16:25:21	7 Januari 2010 pk 16:25:24.23	3.23 detik
7 Januari 2010 pk 16:27:24	7 Januari 2010 pk 16:27:27.23	3.23 detik
7 Januari 2010 pk 16:29:04	7 Januari 2010 pk 16:29:07.25	3.25 detik
7 Januari 2010 pk 16:31:10	7 Januari 2010 pk 16:31:13.95	3.95 detik
Rata-rata waktu pemrosesan		3.358 detik

2. Uji coba tingkat keberhasilan pengiriman tagihan

Percobaan dilakukan dengan *admin company* mengirim tagihan dalam jumlah tertentu ke *user*, seperti pada Tabel 4.2.

Tabel 4. 2 Hasil Uji coba tingkat keberhasilan pengiriman tagihan

Jumlah Tagihan yang Dikirim ke User	Jumlah Tagihan yang Dikirim ke User	Tingkat Keberhasilan
1	1	100%
5	5	100%
10	10	100%

3. Uji coba waktu *request* detail tagihan

Percobaan dilakukan dengan *user* melakukan 5 kali *request* detail tagihan (detail tagihan yang berbeda), seperti pada Tabel 4.3.

Tabel 4. 3 Hasil Uji coba waktu request detail tagihan

Waktu Detail Tagihan di-Request oleh User	Waktu Detail Tagihan Diterima oleh User	Lama Proses
7 Januari 2010 pk 16:48:10	7 Januari 2010 pk 16:48:14.17	4.17 detik
7 Januari 2010 pk 16:49:19	7 Januari 2010 pk 16:49:22.91	3.91 detik
7 Januari 2010 pk 16:50:24	7 Januari 2010 pk 16:50:28.71	4.71 detik
7 Januari 2010 pk 16:51:31	7 Januari 2010 pk 16:51:34.82	3.82 detik
7 Januari 2010 pk 16:51:50	7 Januari 2010 pk 16:51:53.33	3.33 detik
Rata-rata waktu pemrosesan		3.988 detik

4. Uji coba waktu *request* profil tagihan

Percobaan dilakukan dengan *user* melakukan 4 kali *request* profil tagihan, seperti pada Tabel 4.4.

Tabel 4. 4 Hasil Uji coba waktu request profil tagihan

Waktu Profil Tagihan di-Request oleh User	Waktu Profil Tagihan Diterima oleh User	Lama Proses
7 Januari 2010 pk 16:52:05	7 Januari 2010 pk 16:52:08.23	3.23 detik
7 Januari 2010 pk 16:53:03	7 Januari 2010 pk 16:53:06.30	3.30 detik
7 Januari 2010 pk 16:53:25	7 Januari 2010 pk 16:53:28.43	3.43 detik
7 Januari 2010 pk 16:54:10	7 Januari 2010 pk 16:54:13.20	3.20 detik
Rata-rata waktu pemrosesan		3.29 detik

Perbandingan waktu antara sistem baru dengan sistem yang berjalan, seperti pada Tabel 4.5:

Tabel 4. 5 Perbandingan waktu

No.	Hal yang	Waktu proses
-----	----------	--------------

	dibandingkan	Sistem berjalan	Sistem baru
1	Pencetakan tagihan	± 1 Hari	~
2	Pengiriman tagihan	± 7 Hari	± 3.358 Detik

PUSTAKA

- [1]. Research In Motion (2008). *Fundamentals Guide - BlackBerry Java Development Environment Version 4.7.0*. <http://www.blackberry.com>
- [2]. Kline, Rob.(2008). *Sneak Peak: Hosted Data Push Services for Consumer Applications*. <http://www.blackberry.com>