

Perancangan Ulang Sistem Informasi Pos dengan Integrasi SMS Gateway di PT. Pos Indonesia (Persero)

Roni Zakaria*, Herindra Veriawan, dan Yuniaristanto

Laboratorium Sistem Logistik dan Bisnis, Jurusan Teknik Industri, Universitas Sebelas Maret
Jl. Ir. Sutami 36A, Surakarta 57126, Telp/Fax. (0271) 632110

Abstrak

Perancangan ulang sistem informasi pos dilakukan dengan mengintegrasikan semua fase layanan dengan teknologi SMS Gateway dalam satu sistem aplikasi. Sistem mengirim status kiriman mulai dari pencatatan kiriman (fase *collecting*) hingga pengantaran kiriman (fase *delivery*). Setiap pergerakan kiriman termonitor oleh sistem dan pelanggan pos mendapat pembaharuan status kiriman via SMS. Perancangan dimulai dari perancangan aliran data dengan melibatkan semua entitas dalam sistem. Berdasarkan diagram aliran data dapat diidentifikasi kamus data sebagai dasar perancangan basis data dengan model ER (*Entity Relationship*). Selanjutnya, dilakukan perancangan antar-muka aplikasi dengan menggunakan prinsip komposisi kesederhanaan layout, posisi kontrol, konsistensi dan kontras warna. Uji coba aplikasi menggunakan metode *black-box testing* dengan penentuan kriteria berdasarkan hasil identifikasi sistem awal.

Kata kunci: IPOS, SMS Gateway, *black-box testing*.

1. Pendahuluan

Suatu sistem menjadi handal dan mudah dikontrol ketika terdapat kesatuan dalam sistem ini. PT. Pos Indonesia (Persero) memiliki empat fase layanan yang siap melayani pengiriman kiriman dengan cepat, tepat dan akurat. Fase layanan ini terdiri dari fase *collecting*, *processing*, *transporting* dan *delivery*. Aktivitas fase *collecting* terjadi di loket penerimaan, fase *processing* di bagian puri kirim, fase *transporting* di bagian distribusi dan fase *delivery* ditangani oleh bagian antaran pos. Setiap fase atau bagian jasa layanan pos memiliki peran penting dan tugas masing-masing dalam pengiriman pos. Saat ini terdapat aplikasi *Integrated Postal Operations System* (IPOS) yang berfungsi menangani semua aktivitas di tiap fase layanan pos. Aplikasi ini dapat meningkatkan kinerja layanan pos kepada kosumen. Berdasarkan hasil observasi awal, aplikasi IPOS belum bekerja secara efektif untuk mendukung aktivitas dari fase layanan pos. Hampir semua bagian dari kesatuan sistem aplikasi memiliki kelemahan vital. Kelemahan pada aplikasi IPOS dapat menurunkan kinerja pos dalam melayani permintaan pelanggan sehingga berdampak pada penurunan pendapatan kantor pos.

Fase *collecting*, melayani pelanggan dalam hal penerimaan kiriman. Petugas memasukkan data kiriman ke aplikasi untuk mencetak bukti penerimaan kiriman. Aktivitas memasukkan data kiriman ke aplikasi, belum didukung dengan kemampuan aplikasi dalam menampilkan harga untuk semua jenis pengiriman. Hal ini membuat petugas harus mengulangi proses memasukkan data kiriman ketika pelanggan ingin mengetahui harga untuk jenis pengiriman lain. Penelitian ini untuk mengakomodir pembuatan menu tambahan yang dapat menampilkan harga untuk semua jenis pengiriman.

Fase *processing*, bertugas memilah kiriman berdasarkan kode pos tujuan kirim. Petugas memasukkan data kantong kiriman ke aplikasi untuk mencetak surat kirim kantong. Aktivitas memasukkan data *barcode* kiriman belum didukung dengan kemampuan aplikasi dalam melakukan pemanggilan data kiriman yang sudah dimasukkan oleh petugas pada fase *collecting*.

* *Correspondance* : ronny01@runbox.com

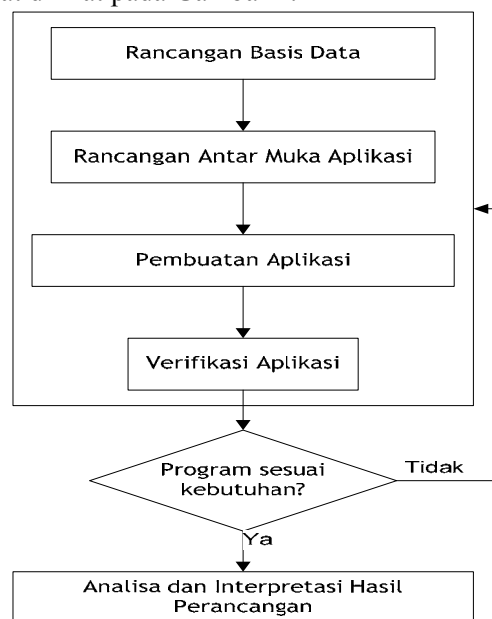
Hal ini membuat petugas pada fase *processing* mengulangi proses pemasukkan data ke aplikasi. Penelitian ini juga untuk mengakomodir pembuatan menu tambahan yang dapat melakukan pemanggilan data kiriman berdasarkan *barcode* kiriman.

Fase *transporting*, bertugas mengirimkan kiriman ke kantor pos tujuan kirim berdasarkan surat perintah jalan. Petugas memasukan *barcode* kantong kiriman dalam aplikasi untuk mencetak surat perintah jalan. Surat perintah jalan belum menggunakan *barcode*. Hal ini memperlama waktu proses pemasukkan data *barcode* kantong kiriman. Penelitian ini untuk mengakomodir pembuatan *barcode* surat perintah jalan yang dicetak di dalam surat perintah tersebut. Aplikasi fase *delivery* dibuat berdasarkan deskripsi tugas petugas antaran kiriman, yaitu mencatat status kiriman serta mengirim status kiriman kepada pengirim dan penerima kiriman.

Layanan *track and trace* kiriman pada sistem saat ini, menuntut pelanggan untuk aktif melakukan pelacakan. Berdasarkan observasi awal, terdapat 68,42% pelanggan yang mengetahui sistem ini dan hanya 36,84% pelanggan yang pernah menggunakan sistem ini. Hal ini membuktikan kalau sistem saat ini kurang diminati pelanggan. Penelitian ini untuk mengefektifkan layanan *Track and Trace* dengan pengiriman notifikasi status kiriman secara otomatis ke nomor *handphone* pengirim dan penerima kiriman. Pengiriman notifikasi otomatis menggunakan teknologi SMS (*Short Message Service*) Gateway.

2. Metode Penelitian

Metode penelitian dapat dilihat pada Gambar 1.



Gambar 1. Metodologi Penelitian

IPOS dirancang dengan menggunakan metode *prototyping*. Metode ini dilakukan dengan mengerjakan rancangan sesuai dengan keperluan sistem, menguji *prototype* sistem yang dirancang, menyampaikan hasil kepada pengguna sistem, lalu dilakukan rekayasa sistem apabila sistem dianggap belum memenuhi kebutuhan. Tahapan perancangan sistem, yaitu:

1. Rancangan basis data.

Perancangan basis data menggunakan dua teknologi yaitu teknologi ADO.DB dan MySQL *Server*. Basis data kode pos menggunakan ADO.DB sedangkan basis data aplikasi menggunakan MySQL *Server*.

2. Rancangan antarmuka.

Rancangan antarmuka (*user-interface*) merupakan perancangan tampilan aplikasi yang mudah untuk dipahami oleh *end-user*. Tujuan dari antarmuka pengguna adalah untuk memungkinkan pengguna menjalankan setiap tugas dalam kebutuhan pengguna. Perancangan antarmuka menggunakan aplikasi *Microsoft Visual Basic 6.0*.

3. Pembuatan aplikasi *Integrated* POS.

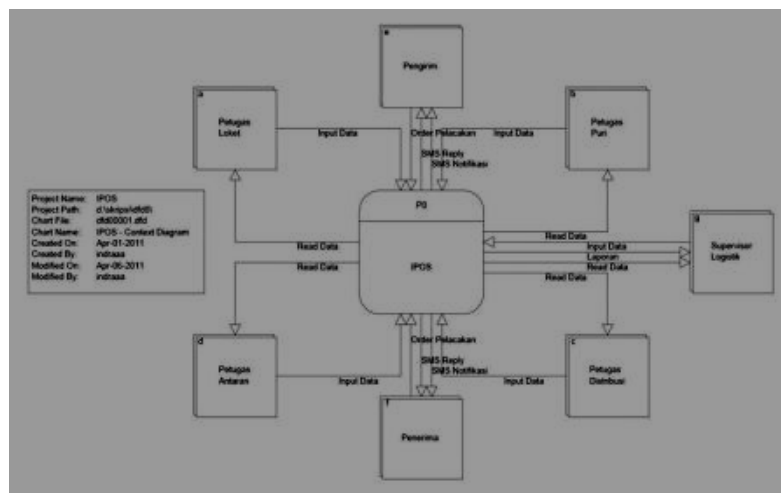
Aplikasi *Integrated* POS yang mengintegrasikan loket pengiriman paket, puri pengiriman, pencarian kode pos serta fitur *track and trace* menggunakan media penyimpanan basis data *offline* dan *online*. Simulasi *web-server* menggunakan *localhost*.

4. Verifikasi aplikasi *Integrated* POS.

Pada tahap verifikasi dilakukan pengujian terhadap fungsi utama aplikasi IPOS. Apabila dalam pengujian ditemukan kelemahan dalam sistem yang tidak sesuai dengan tujuan maka dilakukan proses perbaikan.

3. Hasil dan Pembahasan

Perancangan aplikasi dimulai dengan membuat desain sistem usulan. Desain ini yang menjadi dasar dalam perancangan basis data dan perancangan *graphical user interface* (GUI) aplikasi. Langkah awal dalam perancangan basis data dimulai dengan merancang aliran data atau disebut *Data Flow Diagram* (DFD). Perancangan DFD menggunakan alat bantu aplikasi EasyCase. Aplikasi EasyCase merupakan aplikasi yang digunakan dalam perancangan sistem terstruktur. Aplikasi ini mampu melakukan pengetesan jalur atau arah *rule* antar data atau proses sehingga dipastikan *level balance* dan *rule check* dari suatu sistem menjadi benar. EasyCase mampu mengkompilasi apakah suatu DFD sistem sudah benar atau masih ada kesalahan (Setyadi, 2008).



Gambar 2. Diagram Konteks Sistem

3.1 Rancangan basis data

Perancangan basis data menggunakan dua teknologi yaitu teknologi ADO.DB dan MySQL Server. Basis data kode pos menggunakan ADO.DB sedangkan basis data aplikasi menggunakan MySQL. Perancangan basis data dengan model Entity Relationship terdapat tiga tahapan terdiri dari tahap analisis kebutuhan basis data (kamus data), tahap perancangan relasi antar tabel dan tahap perancangan fisik basis data. Pada tahap analisis kebutuhan basis data dibuat suatu kamus data yang berisi entitas serta atributnya. Entitas ini merupakan *data store* dari DFD yang dirancang sebelumnya. Berikut merupakan beberapa contoh kamus data yang dibuat, yaitu:

- a. Data Bea = {NoResi, Bea Kirim, Diskon, PPN, HTNB, BeaSMS, Total, Kurs}

- b. Data Distribusi = {NoSuratJalan, Konf, BarcodeKantong, IsiKantong, Sopir1, Sopir2, KodeKendaraan}
- c. Data Outbox SMS = {ID, SendingDateTime, SMSCNumber, DestinationNumber, TextDecoded}
- d. Data Inbox SMS = {UpdatedInDB, ReceivingDateTime, SenderNumber, SMSCNumber, TextDecoded}
- e. Data Karyawan = {NIPPOS, NamaKaryawan, Jabatan, JenisKelamin, Alamat, Telepon, HP, Email}

Pada tahap perancangan relasi antar tabel, hubungan antar tabel dapat dilihat dengan memperhatikan *key* yang ada. Penggambaran skema relasi dalam *entity relationship diagram* membantu dalam pembuatan basis data. Pada tahap perancangan fisik basis data, tabel yang terdapat pada ERD diwujudkan dalam bentuk fisik. Bentuk fisik meliputi bentuk basis data MySQL dan basis data ini diletakkan pada basis data *server*.

3.2 Rancangan antar-muka aplikasi

Rancangan antar muka aplikasi ditujukan untuk memudahkan pengguna dalam menggunakan aplikasi (*user friendly*), produktif dan mudah dikelola (Kusumo, 2005). Prinsip dalam merancang *user interface* meliputi komposisi kesederhanaan *layout*, posisi kontrol, konsistensi dan kontras warna. Pada aplikasi IPOS terdapat lima bagian utama aplikasi terdiri dari aplikasi loket penerimaan, aplikasi puri pengiriman, aplikasi distribusi, aplikasi *tracking* dan aplikasi kode pos. Setiap aplikasi memiliki jumlah dan desain *form* yang berbeda, disesuaikan dengan keperluan pengguna. Pada aplikasi IPOS terdapat hak akses penggunaan aplikasi (*privileges*) disesuaikan dengan keperluan karyawan.

3.3 Pembuatan aplikasi

Aplikasi dikembangkan dengan modifikasi arsitektur *fat-client model* dengan menempatkan program aplikasi pada *client* sedangkan *server* mengurus manajemen data. Basis data Microsoft Access (.mdb) digunakan menyimpan data kode pos, harga tiap jenis pengiriman, konfigurasi kantor pos dan konfigurasi perangkat keras. Basis data ini ditanamkan di komputer *client* dan tergabung dengan instalasi aplikasi. Basis data MySQL (.sql) digunakan untuk menyimpan data paket, kantong kiriman, distribusi, historis, karyawan, penerima, pengirim, puri, resi, dan transaksi. Penanaman basis data MySQL dilakukan di dua *server* berfungsi sebagai *server* utama dan satunya berfungsi sebagai *server* cadangan. *Server* cadangan digunakan ketika koneksi ke *server* utama mengalami gangguan. Gangguan dapat berupa gangguan internet, *server-down* maupun *server-maintenance*. Basis data utama ditanamkan di *server* milik jurusan Teknik Industri UNS yang sudah ada dan ditempatkan di Unit Pelayanan Teknis Pusat Komunikasi (PUSKOM) UNS. *Server* cadangan ditanamkan pada komputer lokal.

3.4 Verifikasi aplikasi

Verifikasi aplikasi ditujukan untuk mengetahui apakah aplikasi dapat berjalan sesuai dengan kriteria awal perancangan serta dapat memenuhi semua keperluan sistem. Verifikasi aplikasi dilakukan dengan pengujian atau menjalankan aplikasi pada lingkungan yang mirip dengan lingkungan sebenarnya. Uji coba ini dilakukan untuk menguji setiap fungsi aplikasi apakah sudah sesuai dengan kebutuhan sistem. Performansi aplikasi diketahui ketika aplikasi sudah dapat menjalankan semua fungsi yang diharapkan. Menerapkan uji coba implementasi sistem aplikasi berdasarkan kriteria validasi aplikasi merupakan investasi berharga dalam menjamin kualitas aplikasi dan mencegah terjadinya kelemahan sistem (Sangwan et al., 2006).

Pada uji coba implementasi sistem, proses *application-testing* menggunakan metode *black-box testing*. Metode ini menguji data masukkan pada aplikasi dan melihat informasi keluaran yang dihasilkan oleh aplikasi, apakah sudah sesuai dengan harapan atau tidak. Metode ini tidak terfokus pada proses pengolahan data dalam aplikasi (Test Plant, Ltd, 2011). Berikut merupakan hasil verifikasi aplikasi, sebagai berikut:

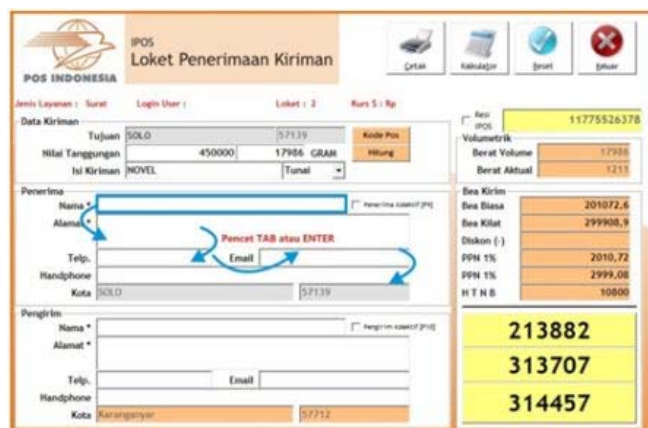
- a. Keberhasilan *switch* basis data ke *server* lokal ketika koneksi internet mengalami *down* atau *maintenance*. Aplikasi sudah dapat melakukan *switch* basis data secara otomatis serta

memberitahukan kepada pengguna jika basis data tidak terkoneksi dengan basis data *server* pusat. Status memuat status jumlah *inbox* SMS, koneksi basis data dan NIPPOS pengguna aplikasi.



Gambar 3. Informasi status koneksi basis data

- b. Kemudahan penggunaan aplikasi dengan minimasi pergerakan *mouse* atau minimasi pengguna dalam menekan *mouse* pada *form*. Perpindahan antar *textbox* isian dapat dilakukan dengan menekan tombol TAB atau ENTER pada *keyboard*. Hal ini mempermudah pengguna dalam mengoperasikan aplikasi.



Gambar 4. Perpindahan antar *textbox* dengan menekan tombol TAB atau ENTER

- c. Keberhasilan rancangan basis data dalam menyimpan data yang sudah dimasukkan melalui aplikasi. Indikator keberhasilan pada saat aplikasi dapat melakukan proses tambah, *update* dan hapus pada basis data yang digunakan. Proses tambah data atau simpan data sudah dapat berfungsi dengan baik. Pada Gambar 5 menunjukkan aplikasi dapat menambah rekaman data pada tabel status *delivery* lewat aplikasi *delivery*.



Gambar 5. Aplikasi dapat melakukan proses simpan data

- d. Keberhasilan aplikasi dalam bertukar data antar tiap fase layanan pos. Aplikasi yang dirancang pada penelitian ini merupakan aplikasi *integrated* layanan pos. Jadi kemampuan aplikasi dalam bertukar data antar bagian layanan pos diperlukan untuk kelanjutan proses aplikasi.



Gambar 6. Aplikasi puri mengambil data dari bagian loket pos (*collecting*)

Pada Gambar 6 menunjukkan aplikasi puri pengiriman (*processing*) dapat memanggil data dari bagian loket pos (*collecting*). Aplikasi dapat memanggil data sesuai dengan yang diperlukan oleh aplikasi. Ketika pengguna memasukkan *barcode* kantong kirim maka secara otomatis aplikasi mencari *barcode* kantong dalam basis data. *Barcode* kiriman dalam kantong dimuat dalam daftar *barcode* kiriman yang telah tersedia.

- e. Keberhasilan aplikasi dalam mencetak *barcode* pada laporan serta membaca *barcode* tersebut dengan *barcode reader*. Aplikasi pada penelitian ini menggunakan *font barcode* ID-Automation Code-39 dengan lisensi GNU-GPL. *Font* ini digunakan secara bebas untuk tujuan sosial maupun komersial.



Gambar 7. *Barcode* hasil cetakan aplikasi telah dapat terbaca dengan baik

- f. Keberhasilan aplikasi dalam mengirim SMS notifikasi kepada pengirim dan penerima paket mengenai status pengiriman. SMS notifikasi status kiriman dikirimkan ke nomor *handphone* pelanggan layanan kilat +. Notifikasi dikirimkan ke nomor pengirim dan penerima kiriman. Setiap perubahan status kiriman diberitahukan kepada pelanggan.



Gambar 8. Layanan notifikasi SMS untuk jenis kiriman Kilat + telah berfungsi dengan baik

Setiap perubahan status kiriman diberitahukan kepada pelanggan. Perubahan status meliputi penerimaan kiriman pada loket pos, pemberangkatan kiriman ke kota tujuan kirim (bagian distribusi kota asal) dan ketika kiriman tiba di kota tujuan (bagian distribusi kota tujuan).

- g. Keberhasilan aplikasi dalam merespon SMS secara otomatis dalam hal pencarian status kiriman maupun pencarian kode pos. Pencarian kiriman berdasarkan nomor *barcode* kiriman dibalas oleh sistem dengan informasi status terakhir kiriman. Status terakhir merupakan status terbaru pada kiriman yang dicari. Pesan balasan dari sistem juga menyampaikan tempat kantor pos dimana kiriman tersebut dipindai. Pesan teks balasan dari sistem dapat dilihat pada Gambar 9.



Gambar 9. Balasan sistem pencarian kiriman berdasarkan nomor *barcode* kiriman

Pencarian kode pos dengan kata kunci nama kecamatan, kelurahan atau kota akan dibalas oleh sistem dengan memberikan nomor kode pos yang paling relevan. Jika, didapatkan nama yang sama maka sistem memberikan dua pilihan kepada pengguna sistem dan jika hanya satu nama maka sistem memberikan satu kode pos yang sesuai. Pesan teks balasan dari sistem dengan pencarian menggunakan kata kunci nama kecamatan dapat dilihat pada Gambar 10.



Gambar 10. Balasan sistem pencarian kode pos berdasarkan kata kunci kecamatan, kelurahan atau kota

4. Kesimpulan dan Saran

Perancangan ulang sistem informasi pos (IPOS) dapat mengintegrasikan semua fase jasa layanan pos mulai dari fase *collecting* (loket penerimaan), fase *processing* (bagian puri), fase *transporting* (bagian distribusi) dan fase *delivery* (bagian antaran). IPOS menggunakan dua basis data yaitu basis data lokal dan basis data utama sehingga dapat digunakan dengan atau tanpa koneksi internet. Integrasi SMS Gateway menggunakan *engine* Gammu versi 1.2.98.

Daftar Pustaka

- Black-box vs White-box Testing: Choosing the Right Approach to Deliver Quality Applications.* (2011). Diakses 30 Juni 2011, dari Test Plant, Ltd: http://www.testplant.com/wp-content/uploads/downloads/2011/06/BB_vs_WB_Testing-1.pdf
- Čihař, M. (2011). *[GW]ammu*. Diakses 21 Juli 2011, dari <http://wammu.eu/gammu/>
- Kusumo, A. (2005). *Buku Latihan Pemrograman Visual Basic 2005*. Jakarta: PT. Elex Media Komputindo.
- Sangwan, R., Bass, M., Mullick, N., Paulish, D. J., & Kazmeier, J. (2006). Critical Success Factors for Global Software Development. Dalam *Global Software Development Handbook* (hal. 9-20). New York: CRC Press: Taylor & Francis Group.
- Setyadi, A. (2008). *Debuging Program dengan Easy Case*. Diakses 3 Januari 2011, dari <http://oke.or.id>
- Williams, L. (2006). Diakses 30 Juli 2011, dari <http://agile.csc.ncsu.edu/SEMaterials/BlackBox.pdf>