

# APLIKASI SISTEM PENAGIHAN MENGGUNAKAN JALUR TERPENDEK PADA PERUSAHAAN *LEASING* PT. ILUFA

Tri Pujadi, Mariono Fabianus

Jurusan Sistem Informasi Fasilkom Universitas Bina Nusantara  
Jl. KH Syahdan No. 9 Palmerah Jakarta Barat

Email : [tripujadi@gmail.com](mailto:tripujadi@gmail.com), [mariono\\_fabianus@yahoo.com](mailto:mariono_fabianus@yahoo.com)

## Abstract

Billing on PT. ILUFA a leasing company use coupon system as proof of daily installment. The problem began when the number of unpaid coupon increased so the finance found difficulties in calculating coupons nominal, while the administration was waiting the finance finished their job to note down billing coupons. With the increasing number of customers, difficulties to determine the shortest billing path arose. Methodology which is used to determine the billing shortest path algorithm is algorithm analysis, while the methodology for analysis and design of information system is object oriented, including create a new work flow for the administration and the finance with activity diagram. The result of algorithm analysis concludes that Dijkstra algorithm is more appropriate to determine shortest billing path. New work flow supports more effective and efficient business processes. Integrated information system is also facilitating coupon note down and coupon nominal calculation. After the billing information system implemented, the billing performance increase.

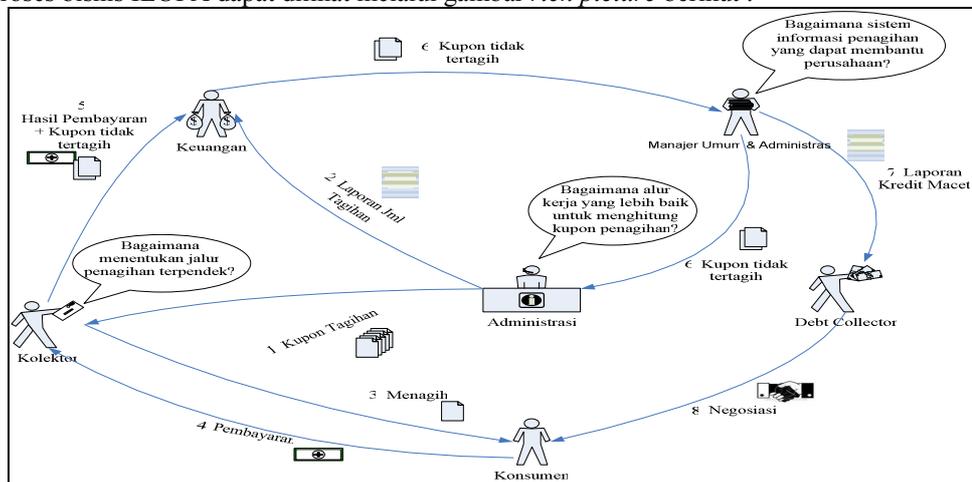
**Keywords:** shortest path, work flow, algorithm analysis, coupon, billing, leasing

## 1. PENDAHULUAN

Perusahaan *leasing* ILUFA memiliki komitmen untuk membantu konsumen tingkat menengah ke bawah agar bisa mendapatkan barang elektronik ataupun peralatan rumah tangga dengan cicilan yang sangat terjangkau, sehingga tidak hanya orang yang mampu saja yang bisa menikmati peralatan-peralatan tersebut. Komitmen ini diwujudkan bidang usaha *leasing*, yaitu usaha sewa guna dimana konsumen dapat memakai barang yang mereka inginkan dengan angsuran dengan nominal yang sangat terjangkau untuk periode waktu tertentu yang dibayar secara harian, dan bila angsuran telah lunas, maka terjadi transfer hak kepemilikan barang kepada konsumen

Perusahaan ini belum memiliki sistem yang terintegrasi untuk menunjang kegiatannya. Kegiatan operasional dalam perusahaan *leasing* ILUFA dinyatakan dalam alur kerja yang mencakup pembuatan perjanjian *lease* sampai dengan penagihan kepada konsumen yang dilakukan setiap hari oleh kolektor perusahaan ILUFA. Penagihan menggunakan sistem kupon. Seiring dengan tumbuhnya perusahaan, maka konsumen dari perusahaan ILUFA bertambah banyak. Oleh karena itu, maka muncul permasalahan pada kinerja perusahaan dalam hal penagihan. Selain itu, karena jumlah kolektor terbatas, sementara konsumen yang harus ditagih terus bertambah maka kemungkinan terjadi konsumen yang tidak tertagih. Masalah lain, untuk melakukan penghitungan tagihan kupon di hari berikutnya juga cukup memakan waktu, karena penghitungan tagihan baru dapat dilakukan setelah kolektor mengembalikan kupon tak tertagih pada hari itu. Untuk mendukung kegiatan penagihan perusahaan *leasing* ILUFA, maka dibutuhkan jalur penagihan terpendek sehingga kegiatan penagihan dapat dilakukan secara lebih efektif dan suatu sistem informasi penagihan untuk mengelola proses penagihan sekaligus mengolah data menjadi informasi yang diperlukan

Proses bisnis ILUFA dapat dilihat melalui gambar *rich picture* berikut :



Gambar 1. Rich Picture Sistem Penagihan di Perusahaan *Leasing* ILUFA

Bagian administrasi mengecek ketersediaan kupon yang harus ditagihkan ke konsumen. Jika ada, maka kupon yang akan ditagihkan dicatat dan diserahkan kepada kolektor. Bagian administrasi membuat laporan jumlah tagihan diserahkan kepada bagian keuangan. Laporan ini digunakan oleh bagian keuangan untuk *cross check* jumlah tagihan oleh kolektor. Semua konsumen yang harus dikunjungi untuk ditagih tertera pada lembar kupon tagihan. Konsumen melakukan pembayaran kepada kolektor sesuai dengan nominal yang tertera pada kupon tagihan. Setelah selesai menagih, kolektor menyetorkan uang tertagih dan kupon yang tidak tertagih (kupon balik) ke bagian keuangan. Bagian keuangan kemudian melakukan *cross check* nominal yang seharusnya tertagih di hari itu dibanding dengan nominal uang yang diserahkan oleh kolektor ditambah nominal semua kupon yang tidak tertagih. Bagian keuangan mengembalikan kupon yang tidak tertagih kepada manajer umum dan administrasi. Manajer akan mencatat kupon balik yang ada, kemudian menyerahkannya ke bagian administrasi. Lalu proses bisnis akan berulang ke langkah pertama lagi setiap harinya. Manajer akan mengeluarkan laporan kredit macet secara periodik kepada *debt collector*. Bagian *debt collector* melakukan negosiasi dengan konsumen dengan harapan masalah pembayaran yang macet dapat diselesaikan.

## 2. TINJAUAN PUSTAKA

*Leasing* adalah suatu kontrak antara *lessor* dengan *lessee* untuk penyewaan suatu jenis barang atau aset tertentu secara langsung, dari pabrik atau agen penjual oleh *lessee*. Hak kepemilikan barang tersebut tetap berada pada *lessor*. *Lessee* memiliki hak pakai atas barang tersebut dengan membayar sewa dengan jumlah dan jangka waktu yang telah ditetapkan. (Equipment *Leasing* Association - ELA-UK)

*Leasing* adalah kegiatan pembiayaan dalam bentuk penyediaan barang modal baik secara *leasing* dengan hak opsi (*finance lease*) maupun *leasing* tanpa hak opsi atau sewa guna usaha biasa (*operating lease*) untuk digunakan oleh *lessee* selama jangka waktu tertentu berdasarkan pembayaran secara berkala. Yang dimaksud dengan *finance lease* adalah kegiatan *leasing* dimana *lessee* pada akhir kontrak mempunyai hak opsi untuk membeli objek *leasing* berdasarkan nilai sisa yang disepakati. Sedangkan yang dimaksud dengan *operating lease* adalah kegiatan *leasing* dimana *lessee* pada akhir kontrak tidak mempunyai hak opsi untuk membeli objek *leasing* KepMen Keuangan No. 1169/KMK.01/1991 Tanggal 21 November 1991

### 2.1 Analisis Algoritma *Shortest-Path*

Analisis algoritma dilakukan untuk mengetahui efektivitas dari dua algoritma *single source shortest-path*, yaitu antara algoritma Bellman-Ford dan algoritma Dijkstra. Algoritma yang lebih efektif kemudian akan dipilih menjadi algoritma yang akan dipakai dalam menentukan jalur penagihan terpendek. Cara analisis algoritma diterapkan dalam 2 langkah, yaitu sebagai berikut :

#### 1. Analisis Perkiraan Priori (*priori estimate*)

Analisis ini dilakukan sebelum algoritma tersebut diproses dengan suatu komputer, fungsi waktu ini disimbolkan dengan notasi O (notasi *big oh*).

##### a. Time Complexity Algoritma Bellman Ford

Perulangan yang terjadi pada algoritma Bellman Ford dapat dilihat pada potongan *pseudocode* berikut :

```
8   for (int k=2; k<=n-1; k++)
9   for (each u such that u!=v and u has
10  at least one incoming edge)
11  for (each <i,u> in the graph)
12  if (dist[u] > dist[i]+cost[i][u])
13  dist[u] = dist[i]+cost[i][u];
```

Dalam analisis *time complexity* untuk Bellman Ford, penjelasan untuk baris 8 – 13 sebagai berikut :

- o Baris ke-8 menyatakan iterasi akan dilakukan dari k=2 sampai k<=n-1
- o Baris ke-9 menyatakan perulangan untuk setiap verteks, dimana verteks dimana verteks tersebut memiliki 1 edge dan verteks tersebut bukan merupakan verteks tujuan.
- o Baris ke-11 menyatakan perulangan untuk setiap edge yang ditemukan dalam graph, dilakukan perbandingan jarak terpendek.

Dari nilai fungsi n di atas, terlihat ada 3 perulangan yang bersarang (*nested loop*), sehingga nilai seluruh *time complexity* untuk n *node* dari Bellman Ford  $O(n^3)$ .

##### b. Time Complexity Algoritma Dijkstra

Perulangan yang terjadi pada algoritma Dijkstra dapat dilihat pada potongan *pseudocode* berikut :

```
14  for (int num = 2; num < n; num++) {
15  // Determine n-1 paths from v.
16  choose u from among those vertices not
17  in S such that dist[u] is minimum;
18  S[u] = true; // Put v in S.
19  for (int w=1; w<=n; w++) // Update distances.
20  if ((S[q]==false&&(dist[w]>dist[u]+cost[u][w]))
```

```

21  dist[w] = dist[u] + cost[u][w];
22  }
    
```

Dalam analisis *time complexity* untuk algoritma Dijkstra, penjelasan *pseudocode* di atas adalah sebagai berikut :

- o Baris ke-14 menyatakan iterasi akan dilakukan dari sebanyak  $n - 2$  kali.
- o Tiap eksekusi perulangan yang dilakukan, akan membutuhkan waktu sebesar  $O(n)$  pada baris ke-16 dan ke-17 untuk memilih verteks berikutnya dan juga untuk perulangan pada baris ke-19, dimana perulangan tersebut digunakan untuk meng-*update* array *dist*.

Dari nilai fungsi  $n$  di atas, maka nilai seluruh *time complexity* untuk  $n$  *node* dari algoritma Dijkstra adalah  $O(n^2)$ .

## 2.2 Sistem Informasi Geografi

Menurut Turban (2009, p277), sistem informasi geografi adalah teknologi visualisasi data yang menangkap, menyimpan, memeriksa, menggabungkan, memanipulasi, dan menampilkan data melalui peta yang digital. Secara garis besar dapat dikatakan bahwa sistem informasi geografi dapat merepresentasikan gambaran dunia nyata (peta) di atas monitor komputer dan ditambah dengan kemampuan analisisnya. Menurut Prahasta (2002, p109), model data peta adalah notasi untuk mendeskripsikan dan mengorganisasikan data dari sistem yang sedang diamati. Model data peta yang digunakan dapat dibagi menjadi :

1. Titik (*Node*), adalah representasi grafis yang paling sederhana untuk suatu objek. Representasi ini tidak memiliki dimensi tetapi dapat diidentifikasi di peta dan ditampilkan di layar dengan menggunakan simbol-simbol (Prahasta, 2002, p111). Jadi model data peta dapat dikatakan *node* pada *graph*.
2. Garis, bentuk linier yang menghubungkan titik-titik untuk merepresentasikan objek 1 dimensi (Prahasta, 2002, p112). Jadi garis merupakan bentuk perwujudan dari *edge* pada suatu *graph*.

## METODE PENELITIAN

Penelitian ini dilakukan pada unit-unit kerja di lingkungan ILUFA yaitu pada unit yang berkaitan dengan administrasi, penagihan, keuangan, pada bulan Juli 2008 sampai Oktober 2008. Data yang dihimpun dalam survai, digunakan untuk melakukan analisis dan merancang aplikasi sistem e-procurement.

Adapun metode analisis dan perancangan menggunakan metode berbasis objek, sebagai berikut :

- 1) Metode analisis : Dalam metode analisis dilakukan dengan dua tahapan yaitu Menganalisis sistem yang sedang berjalan dan Identifikasi kebutuhan informasi. Hasil analisis proses bisnis digambarkan pada rich-picture dimana pihak-pihak yang berkaitan dengan penagihan digambarkan melalui keterkaitan dengan proses dan berbagai dokumen yang digunakan untuk pelaksanaan kerjanya. Dari rich picture tersebut akan dapat diidentifikasi objek-objek yang terkait dengannya. Objek tersebut akan dipergunakan untuk menentukan Actor dan class-data yang ada di dalam sistem.
- 2) Metode perancangan sistem  
 Metode perancangannya adalah *Object Oriented Analysis Design* (OOAD). Adapun tahapan terdiri dari :
  - a. *Problem domain analysis*, dengan menggambarkan: *Structur dan Class diagram serta Behaviour*
  - b. *Application domain analysis*, dengan menggambarkan: *Usage, Function dan User interface*.
  - c. *Architecture design, Component design*, menggambarkan: *Component dan process architecture*.

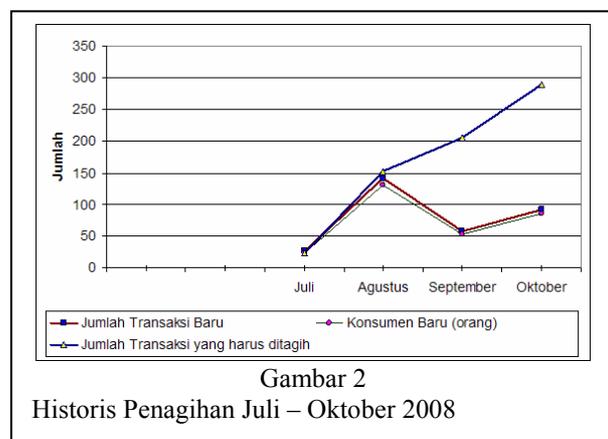
## 3. HASIL DAN PEMBAHASAN

### 3.1 Hasil Pengumpulan Data

Dari hasil pengumpulan data penagihan selama bulan Juli sampai dengan Oktober 2008, hasilnya adalah sbb:

Bulan	Jumlah Transaksi Baru	Konsumen Baru	Jumlah Transaksi yang harus ditagih	Jumlah Kolektor	Rasio Tagihan dengan Kolektor	Jumlah Sales
		(orang)	(orang)	(orang)		(orang)
Juli	27	23	23	1	23	2
Agustus	141	130	153	2	77	5
September	58	53	205	2	103	5
Oktober	91	85	290	3	97	6
Rata-rata	79	73			75	

Tabel 1  
 Historis Penagihan Juli – Oktober 2008



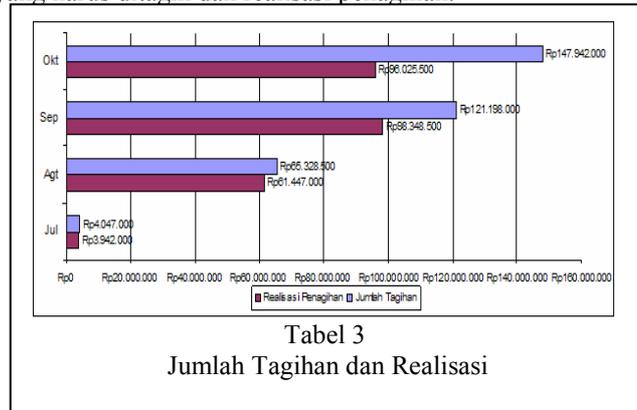
Dari tabel di atas, dapat dilihat kalau rata-rata transaksi baru per bulannya adalah sekitar 79 transaksi, dan rata-rata konsumen baru per bulannya adalah sekitar 73 orang. Jika dirasionalkan, maka 1 kolektor rata-rata akan menangani 75 transaksi. Dari 75 konsumen yang ditangani, semuanya harus dikunjungi agar konsumen dapat melakukan pembayaran. Grafik yang menggambarkan historis penagihan dari bulan Juli 2008 sampai

Oktober 2008 lebih memeperjelas; disini terlihat peningkatan jumlah transaksi, oleh karenanya, diperlukan optimasi dalam menentukan jalur penagihan terpendek, sehingga proses penagihan dapat lebih baik.

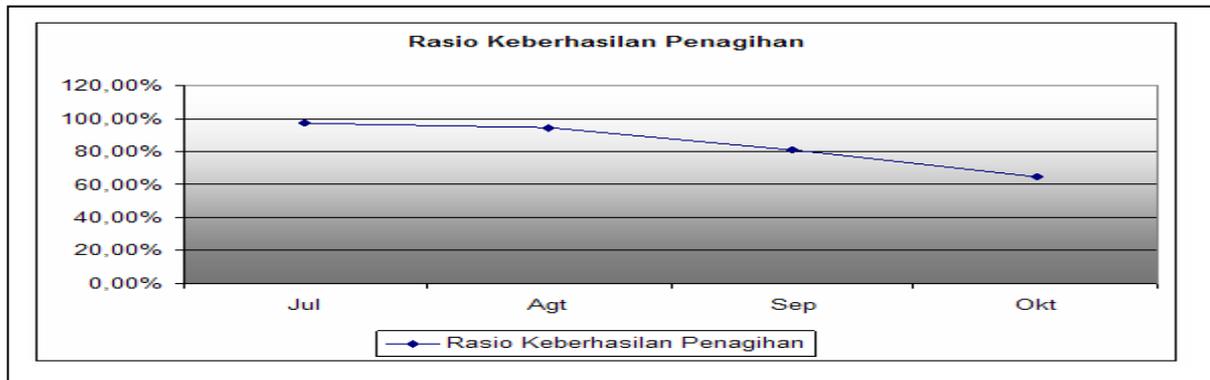
Berikut merupakan data tentang rasio jumlah yang harus ditagih dan realisasi penagihan.

Bulan	Jumlah Tagihan	Realisasi Penagihan	Rasio Keberhasilan Penagihan
Juli	Rp 4.047.000	Rp 3.942.000	97,41%
Agustus	Rp 65.328.500	Rp 61.447.000	94,06%
September	Rp 121.198.000	Rp 98.348.500	81,15%
Oktober	Rp 147.942.000	Rp 96.025.500	64,91%

Tabel 2  
Rasio Keberhasilan Penagihan



Dari tabel dan grafik ini terlihat jumlah tagihan semakin menurun. Hal ini karena makin banyak konsumen yang tidak membayar tepat waktu, juga konsumen yang tidak tertagih oleh kolektor. Sehingga diperlukan suatu rute penagihan terpendek, agar kolektor dapat mengunjungi konsumen secara lebih efisien.



### 3.2 Uji dengan Analisis Algoritma Shortest-Path

Analisis algoritma dilakukan untuk mengetahui efektivitas dari dua algoritma *single source shortest-path*, yaitu antara algoritma Bellman-Ford dan algoritma Dijkstra. Algoritma yang lebih efektif kemudian akan dipilih menjadi algoritma yang akan dipakai dalam menentukan jalur penagihan terpendek. Cara analisis algoritma diterapkan dalam 2 langkah, yaitu sebagai berikut :

Dari hasil uji priori di atas, maka didapat algoritma Bellman Ford memiliki kompleksitas  $O(n^3)$  dan algoritma Dijkstra memiliki kompleksitas  $O(n^2)$ . Maka baik dalam jumlah instruksi maupun jumlah waktu yang dibutuhkan, algoritma Dijkstra selalu lebih efisien dibanding dengan algoritma Bellman Ford dalam mencari rute terpendek. Hal ini dapat dilihat dalam tabel pertumbuhan n berikut :

n	n <sup>2</sup>	n <sup>3</sup>
10	100	1000
20	400	8000
30	900	27000
40	1600	64000
50	2500	125000
100	10000	1000000

n	n <sup>2</sup>	n <sup>3</sup>
10	0,1 μs	1 μs
20	0,4 μs	8 μs
30	0,9 μs	27 μs
40	1,6 μs	64 μs
50	2,5 μs	125 μs
100	10 μs	1000 μs

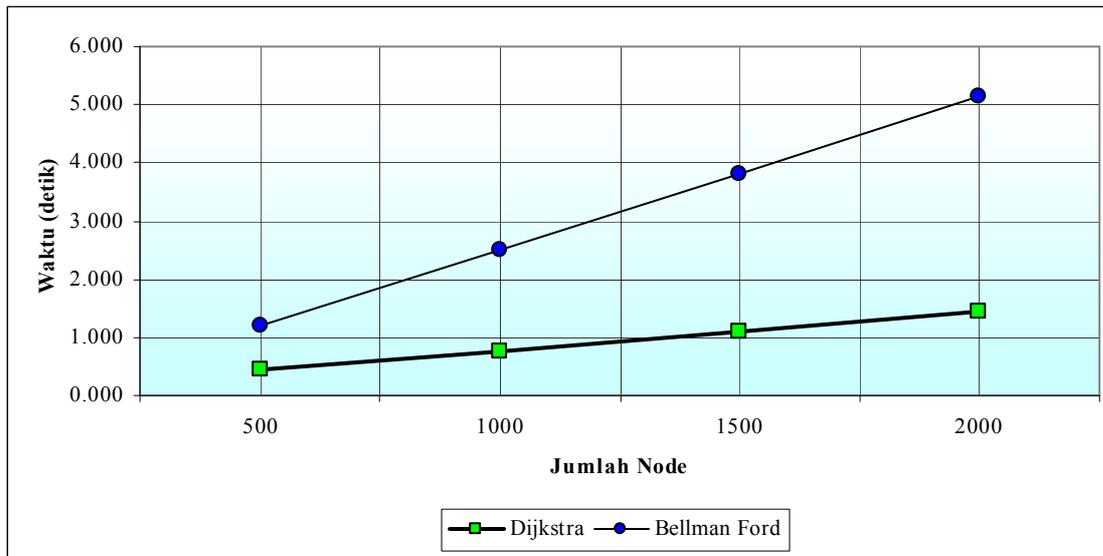
Berdasarkan tabel Fungsi Pertumbuhan n, terlihat adanya perbedaan jumlah langkah yang ditempuh untuk menghasilkan output yang sama, dari sumber permasalahan yang sama antara algoritma Bellman Ford dengan algoritma Dijkstra.

Jika dibanding dengan algoritma Dijkstra, terlihat kalau algoritma Bellman Ford membutuhkan waktu yang lebih besar (untuk *output* yang sama) dalam menghitung jumlah langkah, seiring dengan pertumbuhan jumlah  $n$ .

#### Analisis Pengujian Posteriori (*posteriori testing*)

Pengujian dilakukan dengan cara sebagai berikut :

- 1 Pengujian menggunakan suatu jaringan yang terdiri dari 2000 *node*.
- 2 Pengujian dilakukan dengan mencari jarak terpendek antara *node* awal dengan *node* tujuan, yaitu :
  - *Node* ke-1 dengan *node* ke-2000
  - *Node* ke-1 dengan *node* ke-1500
  - *Node* ke-1 dengan *node* ke-1000
  - *Node* ke-1 dengan *node* ke-500
- 3 Masing-masing pengujian akan dilakukan sebanyak 30 kali, dan dicatat waktunya. Kemudian waktu yang didapat dirata-rata untuk dijadikan waktu acuan.



Gambar 4 Grafik Perbandingan Waktu Algoritma Dijkstra – Bellman Ford

Dari grafik, dapat dilihat kalau algoritma Dijkstra ternyata membutuhkan waktu yang lebih singkat dalam menentukan rute terpendek jika dibandingkan dengan algoritma Bellman Ford, walau *output* yang dihasilkan oleh kedua algoritma adalah sama.

Dari hasil analisis perkiraan priori dan analisis pengujian posteriori memberikan kesimpulan yang sama, yaitu dibandingkan dengan Bellman Ford, algoritma Dijkstra lebih baik dalam mencari jalur terpendek. Jadi algoritma yang digunakan dalam menentukan jalur penagihan terpendek adalah algoritma Dijkstra.

### 3.3 Identifikasi Masalah

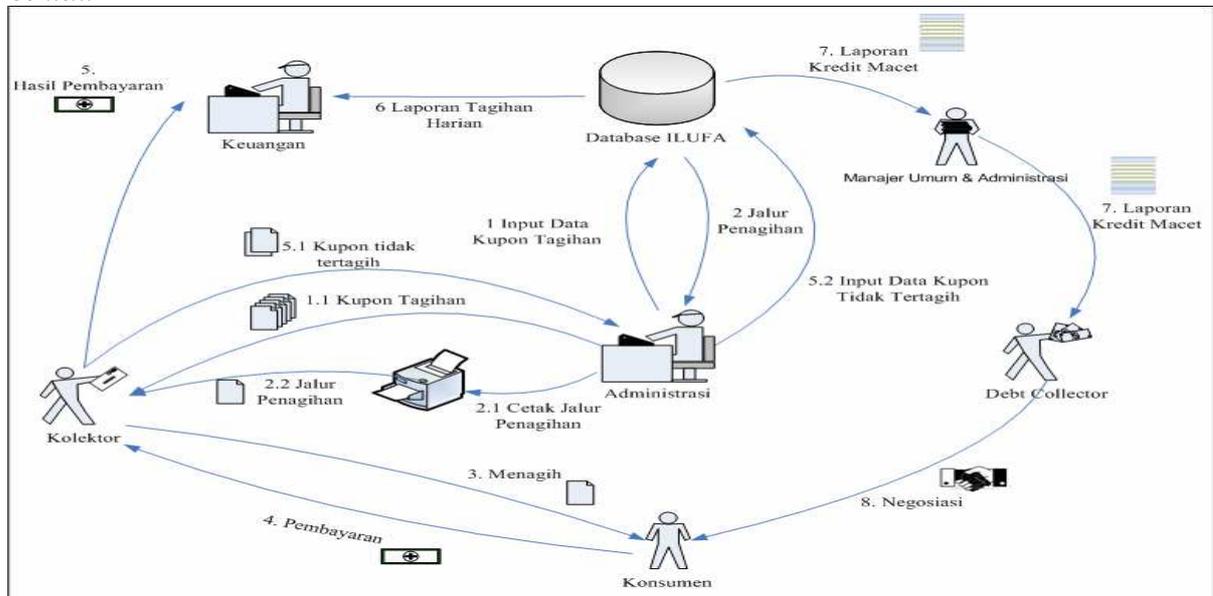
- Untuk mengeluarkan dan melakukan pencatatan atas kupon yang akan ditagihkan di hari esoknya, bagian administrasi harus menunggu sampai bagian keuangan mengembalikan kupon yang tidak tertagih di hari ini, sehingga kupon yang tidak tertagih tersebut dapat kemudian digabungkan dengan kupon yang akan ditagihkan hari esoknya. Masalah timbul ketika jumlah kupon yang tidak tertagih dari hari ke hari makin banyak, yang disebabkan karena adanya konsumen yang menunda-nunda pembayaran, kredit macet, sampai pada alasan lainnya seperti terdapat konsumen yang tidak terkunjungi oleh kolektor. Akibatnya bagian administrasi menjadi makin sering bekerja di luar jam kerja untuk menyelesaikan pekerjaan sehari-harinya.
- Hasil pencatatan terhadap kupon yang keluar harus diserahkan kepada bagian keuangan, supaya mereka bisa melakukan *cross check* pada hasil penagihan kolektor. Karena belum ada sistem yang terintegrasi, maka hasil pencatatan harus selalu diserahkan secara *hardcopy* kepada bagian keuangan. Hal ini tentu akan tidak efisien dalam penggunaan sumber daya seperti kertas, tinta, dan sebagainya.
- Dalam melakukan penghitungan dan *cross check* hasil penagihan kolektor, maka bagian keuangan harus menunggu bagian administrasi menyerahkan laporan berupa *hardcopy*. Hal ini juga disebabkan karena tidak adanya sistem yang terintegrasi.
- Bagian keuangan harus melakukan penghitungan terhadap kupon yang tidak tertagih. Kolektor dan administrasi juga harus menunggu proses penghitungan sampai selesai. Ketika kupon yang tidak tertagih makin banyak, maka semakin lama proses dan semakin besar kemungkinan kesalahan penghitungan kupon yang tidak tertagih.
- Keuangan harus membuat laporan kupon yang tidak tertagih dan yang tertagih kepada bagian administrasi, sehingga kinerja penagihan dapat dipantau dari hari ke hari.

## PERANCANGAN SISTEM INFORMASI

### 4.1 Analisis Pemecahan Masalah

Masalah – masalah yang dialami oleh bagian administrasi dan bagian keuangan yang telah disebutkan di atas, dapat dipecahkan dengan menggunakan suatu sistem yang mengintegrasikan bagian administrasi dengan bagian keuangan, dan suatu prosedur kerja yang disebut sistem transfer kupon. Sistem transfer kupon yang diusulkan adalah kupon yang tidak tertagih pada hari ini akan ditagihkan (ditransferkan) ke hari kerja berikutnya, dan pencatatan tanggal keluar kupon ke dalam sistem akan dilakukan secara otomatis.

#### Context

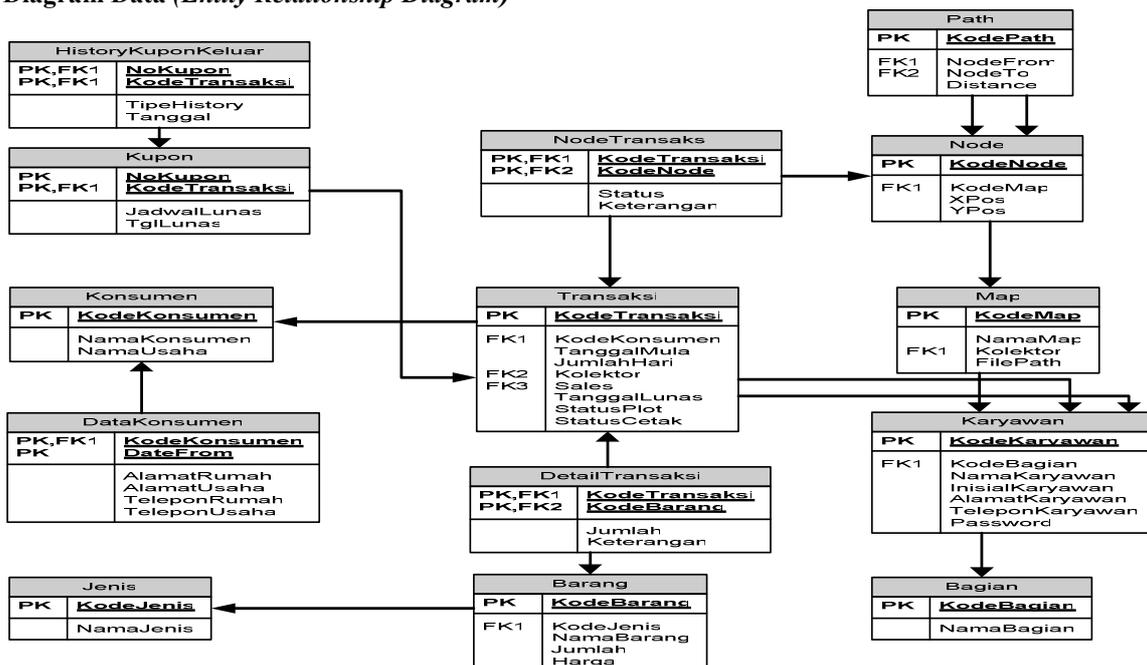


Gambar 5 Rich Picture Sistem yang Akan Dikembangkan

### 4.2 Application Domain

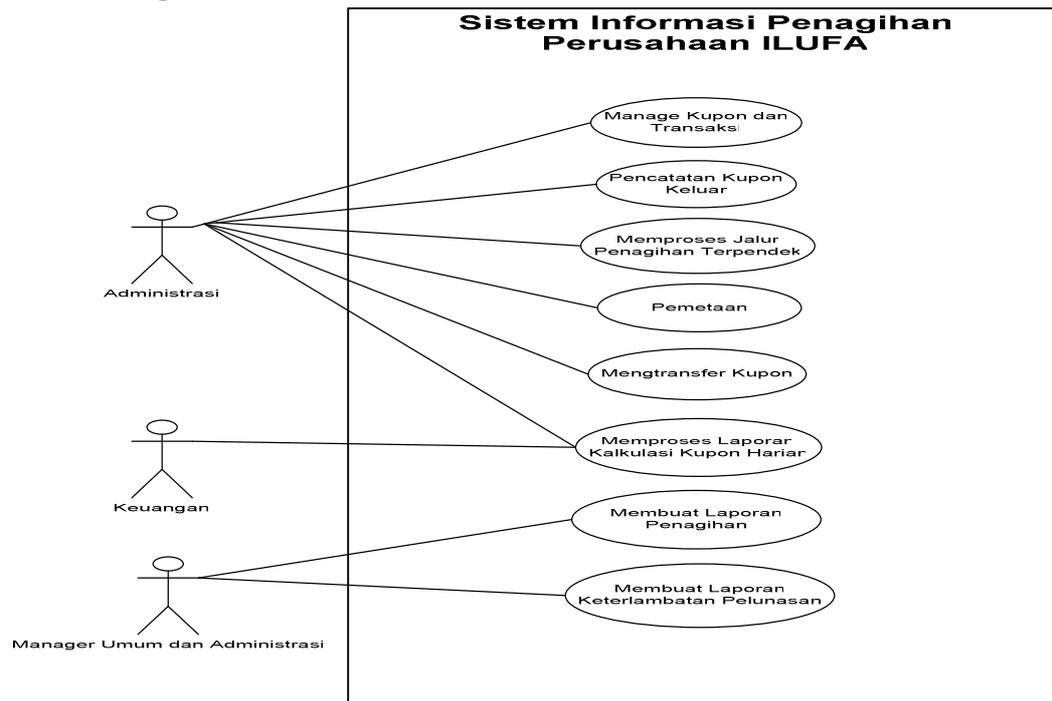
Sistem informasi penagihan akan digunakan oleh bagian administrasi, keuangan, kolektor dan manajer divisi umum dan administrasi. Sedangkan manajer juga dapat menggunakan sistem untuk kegiatan *controlling*. Bagian administrasi akan menggunakan sistem untuk melakukan pencatatan terhadap kupon tagihan yang dikeluarkan sehari – hari. Sistem informasi penagihan juga harus membantu kolektor dalam mendapatkan jalur penagihan terpendek, dan hal tersebut dimungkinkan dengan penggunaan algoritma jalur terpendek Dijkstra.. Adanya kupon yang tidak tertagih yang diinput oleh bagian administrasi, juga harus dapat dilihat oleh bagian keuangan, sehingga mereka mengatur pengeluaran kupon untuk hari berikutnya.

#### Diagram Data (Entity Relationship Diagram)



Gambar 6 Entity Relationship Diagram

## Use Case Diagram



Gambar 7. Use Case Diagram

**Actor** Administrasi adalah orang yang memiliki tanggung jawab dalam *manage* kupon dan transaksi yang sedang berjalan. Semua kupon yang keluar untuk ditagihkan kepada konsumen akan dicatat oleh administrasi. Bagian ini membuat laporan kalkulasi kupon berdasarkan kupon yang keluar tiap harinya. Administrasi memiliki tanggung jawab dalam memetakan jalur penagihan, dan memproses jalur penagihan terpendek, untuk diberikan kepada kolektor. Selain itu bertanggung jawab *manage* kupon yang tidak tertagih.

**Actor** Keuangan dalam proses penagihan memiliki tanggung jawab dalam mengelola uang hasil penagihan. Bagian ini akan mendapatkan laporan kalkulasi kupon harian dari sistem, dimana laporan tersebut memiliki ringkasan total nominal kupon yang keluar untuk ditagih dan total nominal kupon tidak tertagih. Selanjutnya bagian keuangan melakukan *cross check* hasil penagihan yang seharusnya diserahkan oleh kolektor.

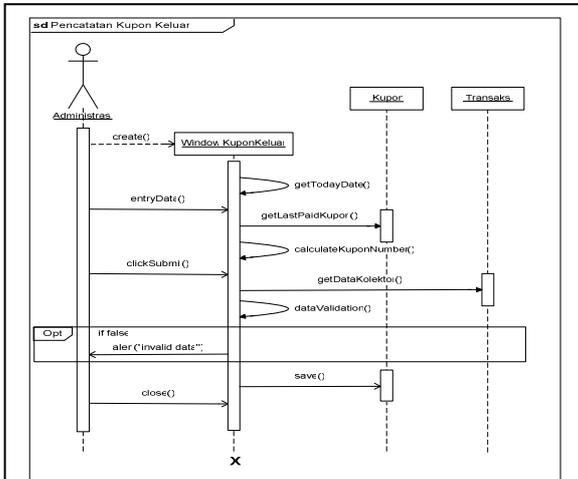
**Actor** Manajer umum dan administrasi bertanggung jawab sebagai pimpinan dan membuat laporan penagihan secara periodik. Manajer secara periodik akan mengeluarkan laporan tentang keterlambatan pembayaran, supaya dapat di-*follow up* oleh bagian lain. Secara periodik pula, manajer umum dan administrasi akan membuat laporan kinerja penagihan selama 1 periode tertentu, sebagai laporan pertanggungjawaban.

**Use Case** *Manage* kupon dan transaksi dimulai oleh administrasi. Bila terdapat perubahan pada transaksi ketika penagihan sedang berjalan, contohnya terdapat perubahan alamat penagihan konsumen, atau terdapat peralihan tanggung jawab kolektor yang menagih atau sales yang *handle* konsumen, maka dapat dilakukan *update* data. Dalam fungsinya untuk manajemen kupon, *use case* ini menerangkan status suatu kupon telah tercetak atau belum. Jika kupon belum tercetak, maka ada fasilitas untuk mencetak kupon penagihan. Selain itu, lewat *use case* ini pula, administrasi dapat melakukan *track record* atas kupon keluar dan transfer kupon serta memperbaiki kesalahan dalam pencatatan kupon keluar ataupun pencatatan transfer kupon.

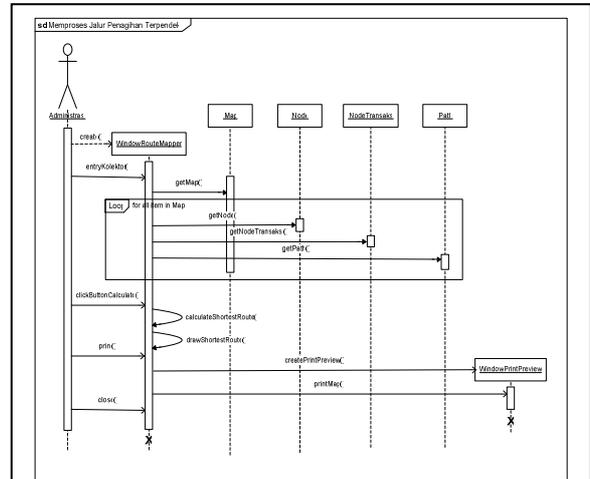
**Use Case** Memproses jalur penagihan terpendek digunakan oleh administrasi. Tujuan dari memproses jalur penagihan terpendek adalah supaya kolektor mengetahui jalur penagihan terpendek yang harus dilalui. Administrasi akan memasukan kolektor, dan sistem akan menampilkan kawasan penagihan kolektor yang bersangkutan. Kemudian sistem juga akan *load* semua data seperti *node*, *path* dan *node* transaksi yang masih aktif yang berada pada kawasan bersangkutan. Ketika administrasi menginstruksikan sistem untuk memproses jalur penagihan terpendek, maka sistem akan menghitung jalur penagihan terpendek dengan menggunakan algoritma Dijkstra. Jalur penagihan terpendek akan ditampilkan pada tampilan peta. **Use Case** Pemetaan akan digunakan oleh administrasi untuk memetakan jalur penagihan. Administrasi dapat memasukan peta kawasan penagihan baru untuk kolektor tertentu. Kemudian, administrasi dapat memasukan *node - node* baru. Administrasi juga dapat memasukan *path* baru yang menghubungkan *node - node* pada peta dan akan disimpan ke dalam sistem. *Node* transaksi akan merepresentasikan suatu transaksi, sehingga *node* transaksi akan disimpan layaknya *node* biasa, ditambah atribut *node* transaksi itu sendiri, seperti yang telah digambarkan dalam *class diagram*. *Node* transaksi baru akan diberi *state active* pada sistem. Apabila diperlukan, administrasi dapat membaca data peta yang telah ada dalam sistem, dan sistem menampilkan jalur dan *node* di peta.

**Sequence Diagram**

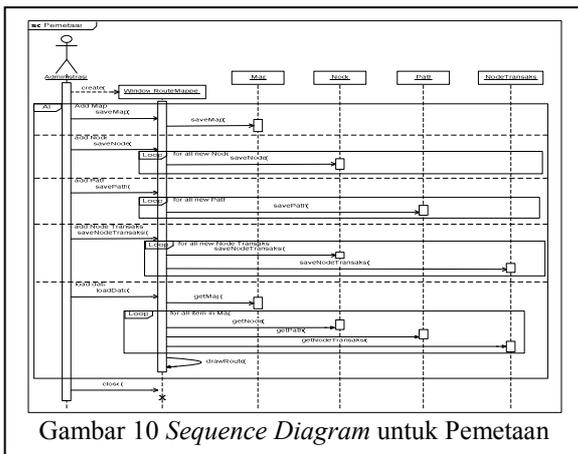
Sequence Diagram merupakan suatu diagram yang menggambarkan interaksi antara actor, objek, serta user interface yang ada dalam sistem informasi. Sequence Diagram juga menggambarkan urutan pemanggilan prosedur dan event message yang dikirimkan antara entitas satu dengan lainnya dalam suatu urutan waktu.



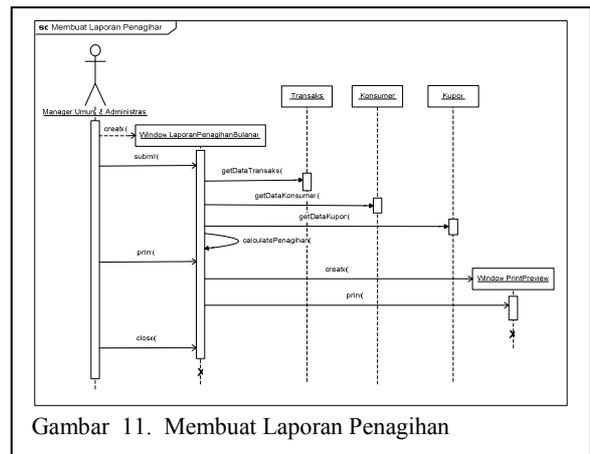
**Gambar 8**  
 Sequence Diagram Pencatatan Kupon Keluar



**Gambar 9** Sequence Diagram untuk Memproses Jalur Penagihan Terpendek



**Gambar 10** Sequence Diagram untuk Pemetaan



**Gambar 11.** Membuat Laporan Penagihan



- Horowitz, E., S. Sahni dan S. Rajasekaran. (1998). *Computer Algorithms/C++*. Computer Science Press, United States of America.
- Johnsonbaugh, R. (2009). *Discrete Mathematics*. Seventh Edition. Pearson Prentice Hall, United States of America.
- Lau, Yun-Tung. (2001). *The Art of Objects : Object Oriented Design and Architecture*. Addison Wesley, United States of America.
- Mathiassen, L., Munk-Madsen, A., Nielsen, P. A., Stage, J. (2000). *Object-Oriented Analysis and Design*. Marko, Aalborg.
- Prahasta, Eddy. (2002). *SIG : Konsep-konsep dasar edisi 2*. CV Informatika, Bandung.
- Sigit Triandaru dan Totok Budisantoso. (2006). *Bank dan Lembaga Keuangan Lain (Edisi 2)*. Salemba Empat, Jakarta.
- Taha, Hamdy A. (2003). *Operation Research Seventh Edition*. Prentice Hall, United States of America.
- Turban, Efraim dan Rainer, R.Kelly. (2009). *Introduction to Information Systems, Enabling and Transforming Business*. John Wiley and Sons, Inc., Asia.