Penerapan Teknik Kriptografi dalam Skema *Micropayment*

Mia Nur Indah[1], Martharany R[2], dan Tessa Rahma Dewi[3]

Departemen Teknik Informatika
Institut Teknologi Bandung
Jalan Ganesha 10 Bandung 40132

E-mail : if12006@students.if.itb.ac.id[1], if12062@students.if.itb.ac.id[2], if12073@students.if.itb.ac.id[3]

Abstract

Micropayment merupakan salah satu alternatif pembayaran elektronik berbasis internet. Prosedur pembayaran ini melibatkan tiga pihak, yaitu *users, vendors*, dan *brokers*. Pembayaran yang dilakukan pada mekanisme ini dilakukan untuk jumlah uang yang relatif kecil dan intensitas transaksi yang tinggi.

Aspek keamanan dalam sistem pembayaran ini dilakukan dengan cara meminimalisasi penggunaan algoritma kriptografi kunci publik dan mengoptimalkan pemanfaatan fungsi *hash*. Hal ini disebabkan oleh tingginya biaya yang diperlukan untuk mengimplementasikan algoritma kunci publik ini. Melalui penggunaan fungsi *hash* maka komputasi proses pembayaran dapat dilakukan dengan lebih cepat. Fungsi *hash* 10.000 kali lebih cepat dibanding kunci publik.

Terdapat dua skema dasar terkait dengan rancangan *micropayment* ini, yaitu *payword* dan *MicroMint*. Kedua skema ini diajukan oleh Ronald Rivest dan Adi Shamir. Tujuan utama dari skema ini adalah efisiensi komputasi yang dilakukan dalam pemrosesan pembayaran. Dengan demikian pembayaran elektronik ini membutuhkan waktu yang relatif singkat.

***Key words**: micropayment, payword, MicroMint, fungsi hash, kunci publik*

1. Pendahuluan

Pengunaan internet dalam dekade terakhir menunjukkan angka yang signifikan. Data terakhir terkait dengan penggunaan internet (2005) di seluruh dunia mencapai angka 6.091.715.663[1].

Tingginya populasi penggunaan internet ini mengimplikasikan perubahan kultur sosial dalam kehidupan bermasyarakat. Dewasa ini seluruh aspek kehidupan masyarakat diwarnai dengan penggunaan teknologi informasi.

Hal ini menyebabkan terbukanya peluang dalam mengembangkan proses bisnis berbasis internet. Transaksi keuangan melalui media internet merupakan alternatif baru bagi masyarakat dalam menjalankan aktivitas perekonomian.

Dengan adanya alternatif ini masyarakat luas akan dimudahkan dalam bertransaksi tanpa harus mengeluarkan usaha ekstra melalui cara-cara konvensional. Dalam konteks ini cara konvensional mengacu pada *electronic checks*, *credit card*, dan *debit card*.

Micropayment merupakan salah satu instansiasi aplikasi perekonomian berbasis

internet. *Micropayment* merupakan salah satu bentuk pembayaran yang dilakukan secara elektronik[2].

Penggunaan *micropayment* ini dikhususkan pada pembayaran dengan nominal relatif kecil dan frekuensi transaksi dengan intensitas yang tinggi.

Salah satu contoh penggunaannya adalah pembayaran mp3 yang di-*download* dari mp3server. Biasanya intensitas transaksi mp3 ini dilakukan berulang kali, sehingga pembayaran untuk setiap kali download dengan menggunakan kartu kredit atau kartu debit menjadi tidak efisien dan membutuhkan biaya yang tinggi. Oleh karena itu pembayaran untuk transaksi sejenis memerlukan prosedur lain, diantaranya dengan menggunakan *micropayment*.

Kriptografi merupakan faktor kunci dalam memasyarakatkan penggunaan aplikasi *micropayment* ini karena pembayaran elektronik yang tidak ditangani secara hati-hati akan menimbulkan peluang baru dalam kriminalitas dunia virtual.

2. Skema *micropayment*

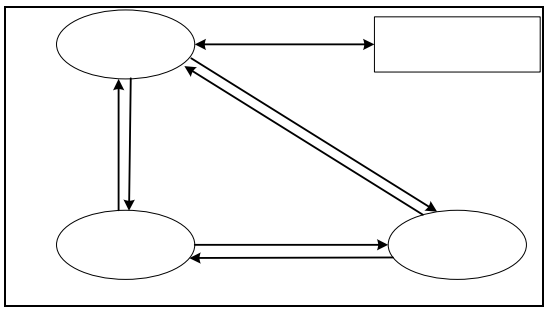
Skema dasar yang mewakili pembangunan aplikasi *micropayment* diperkenalkan oleh Ronald L. Rivest dari *MIT for laboratory for computer science* dan Adi Shamir dari *Weizmann Insitute of Science* pada tahun 1996[3].

Skema dasar yang diajukan adalah *payword* dan *MicroMint*. Tujuan utama dari kedua skema ini adalah meminimalisasi jumlah dari operasi kunci publik yang dibutuhkan dalam pembayaran. Hal ini disebabkan oleh tingginya biaya yang diperlukan dalam mengimplementasikan algoritma kriptografi kunci publik.

Alternatif lain yang digunakan oleh Rivest dan Shamir dalam manajemen keamanan aplikasi adalah dengan mengoptimalkan penggunaan operasi *hash*. Berikut adalah gambaran kasar perbandingan operasi *hash* dan kriptografi kunci publik[3].

RSA *sign* (*private key*)	2/sec
RSA *verify* (*public key*)	200/sec
Hash function	20000/sec

Entitas yang terkait dalam skema di atas terdiri dari tiga elemen, yaitu *brokers*, *users*, dan *vendors*. Gambar 1 merupakan representasi hubungan yang terjadi antar entitas tersebut.



Gambar 1 *Micropayment Entities* [3]

Dari gambar 1 dapat dilihat bahwa *brokers* mengotorisasi *users* agar dapat melakukan *micropayment* kepada *vendors*. Dengan demikian pembayaran yang dilakukan kepada *vendors* dapat dilakukan hanya jika operasi otorisasi berhasil dilakukan. Selanjutnya pembayaran yang telah terkumpul di pihak *vendors* akan ditukarkan kepada *brokers*.

Hubungan yang terjadi antara *user* dengan *vendor* merupakan relasi yang sementara waktu. Hal ini disebabkan karena interaksi *user* dan *vendor* dilakukan hanya jika diperlukan dalam rangka pembayaran terhadap barang yang sudah diperjualbelikan.

Di sisi lain hubungan antara *broker-user* dan *broker-vendor* merupakan relasi jangka panjang. Hal ini sejalan dengan alur yang ditunjukkan oleh gambar 1.

Dalam skema ini, digunakan kriptografi kunci publik (misalnya RSA dengan eksponen publik pendek). Kunci publik dari *broker* B, *user* U dan *vendor* V dinotasikan dengan $PK_{B,}$ $PK_{U,}$ dan PK_V dan kunci privat dinotasikan dengan $SK_{B,}$ $SK_{U,}$ dan $SK_{V.}$ Pesan M dengan tanda tangan digital yang dihasilkan dari kunci privat SK dinotasikan dalam $\{M\}_{SK.}$ *Signature* ini dapat diverifikasi dengan kunci publik PK yang berkorespondensi. *h* menotasikan fungsi *hash*, seperti MD5 atau SHA.

Properti penting dari fungsi *hash* adalah sifat satu arah dan ketahanan akan adanya *collision*. Dibutuhkan penelitian/percobaan yang besar untuk mendapatkan satu masukan yang menghasilkan output yang diberikan, atau menemukan dua masukan yang menghasilkan output yang sama. Panjang input, dalam beberapa kasus, dapat/mungkin sama dengan panjang output.

2.1 *Payword*

Payword merupakan skema berbasis kredit. *user* membentuk/membuat *account* dengan *broker*, yang memberikan sertifikat *Payword* yang telah ditandatangani secara digital dan mengandung nama penjual, nama pengguna serta alamat IP, kunci publik pengguna, tanggal tenggat waktu (*expiration date*) dan informasi lainnya. Sertifikat harus diperbaharui oleh *broker* untuk setiap periode waktu tertentu, misalnya sekali dalam sebulan. Sertifikat ini memberikan otorisasi kepada *user* untuk membuat rantai *Payword* dan memberikan jaminan kepada *vendor* bahwa *payword user* dapat ditebus oleh *broker*.

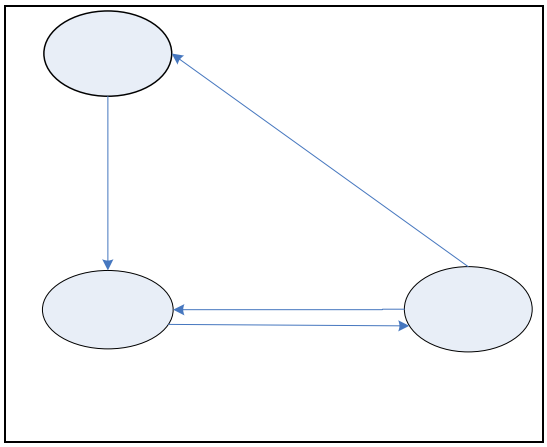
Skema *payword* cocok digunakan untuk *micropayment* karena skema ini menggunakan rantai *hash* ketimbang digital *signature* dalam prosedur pembayarannya. Oleh karena itu, skema ini menawarkan biaya komputasi yang minimal. Selain itu, skema ini juga dapat mengurangi jumlah dari komunikasi *online* antara bank dan *vendor* karena *vendor* tidak harus terus menerus menetap untuk setiap transaksi pembelian, *vendor* hanya perlu meng-update nilai *hash* pada database dan menampilkan proses pembersihan dengan menunjukkan nilai *hash* yang baru dari bloknya.

Protokol skema *Payword* adalah sebagai berikut :

1. *User* meminta *broker* untuk memberikan sertifikat kunci publik PK_u dan kredit.
2. *Broker* memberikan *user* sebuah sertifikat yang telah ditandatangani, $C_{U,}$ yang dapat digunakan untuk membuat rantai *payword* selama satu bulan .
3. Sertifikat berisi C_U = {*broker*, *user*, alamat IP *user*, $PK_{U,}$ tanggal tenggat, batas kredit, dll$\}_{SKB}$ yang menjamin validitas dari kunci publik U dan kreditnya. PK_U adalah kunci publik RSA milik *user*.

4. Sertifikat ini mengotorisasi pengiriman barang hanya kepada alamat internet tertentu.
5. *User* memverifikasi C_U, dan mengkonfirmasi apakah *broker* B menjamin validitas dari PK_C. Selanjutnya, *user* mengkonfirm nilai dari kreditnya.
6. *User* menghasilkan nilai w_n secara random dan menghitung rantai *hash* $w_n \rightarrow ... \rightarrow w_0$. *User* dapat memilih panjang n dari rantai *hash*.
7. *User* memberikan nilai *root* w_0 kepada *vendor* dengan menandatangani pesan komitmen M_{UV}.
8. *User* menghitung komitmen M dari akarnya w_0 : M = {*user*, *vendor*, w_0, C_U, tanggal tenggat}$_{SKU}$. Komitmen mengandung sertifikat *user* C_U.
9. Dengan memverifikasi M dengan kunci publik PK_C , *vendor* memeriksa apakah *user* adalah pelanggan yang valid. Selanjutnya, dengan memverifikasi C_U dengan kunci publik PK_B, *Vendor* mengkonfirmasi bahwa *user* telah terotentikasi dan kredit *user* telah dijamin oleh B.
10. *User* mengirimkan pesanan (yang akan dibeli) dan nilai *hash* (w_i,*i*) sebagai pembayaran ke *vendor*. i merupakan indeks dari nilai *hash*.
11. *Vendor* memverifikasi $w_{i-1} = h(w_i)$ dan mengkonfirmasi validitas dari pembayaran.
12. Jika *vendor* menemukan semua tes verifikasi telah terlewati, *vendor* akan mengirimkan barangnya atau menyediakan layanannya kepada *user*.
13. *Vendor* meyakinkan *broker* terhadap validitas PK_U dengan memeriksa C_U dan validitas transaksi antara *user* dan *vendor* dengan mengecek M menggunakan PK_U. *Vendor* mengirimkan nilai *hash* (w_i,*i*) untuk pembayaran kepada *broker*. Dalam hal ini, *vendor* dapat pula hanya mengirim koin terakhir (w_k, *k*) yang diterima dari *user* untuk pembayaran.
14. *Broker* memverifikasi M dan (w_k,k) yang diterima dari S dengan informasi yang ada pada basis data *broker*. Jika valid, *broker* mengirimkan pembayaran dari rekening *user* ke rekening *broker*.
15. *Broker* meng-update dan menyimpan hanya koin terakhir yang diterima dalam basis data B.
16. *Broker* tidak terlibat secara langsung dalam transaksi pembayaran.

Gambar 2 berikut adalah gambaran umum mengenai skema *Payword*.



Gambar 2 Skema *Payword*

Payword Cost

Cost yang terdapat dalam skema ini diantaranya *cost* untuk :

1. Satu tanda tangan oleh *broker* untuk setiap *user* pada setiap bulan (*CU*)
2. Satu tanda tangan oleh *user* untuk setiap *vendor* perharinya (*MUV*)
3. Dua proses verifikasi oleh *vendor* terhadap setiap *user* setiap harinya (*CU* dan *MUV*)

4. Sebuah verifikasi oleh *broker* untuk setiap *vendor* dari setiap *user* perharinya (untuk *MUV*)
5. Sebuah perhitungan fungsi *hash* oleh masing-masing *user*, *vendor* dan *broker* untuk setiap satu sen pembayaran.

2.2 *MicroMint*

MicroMint didesain untuk menyediakan keamanan dengan ongkos yang rendah dan memiliki optimasi untuk pembayaran dengan nilai yang rendah. *MicroMint* tidak menggunakan operasi kunci publik sama sekali.

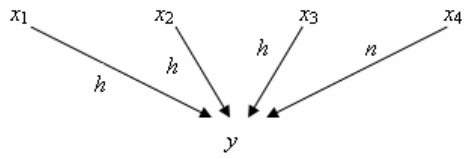
Pada skema ini, *broker* akan membuat alat pembayaran berupa koin yang kemudian akan dibeli oleh *user*. *User* akan memberikan koin tersebut kepada *vendor* sebagai alat pembayaran dan *vendor* akan mengembalikan lagi koin tersebut kepada *broker* untuk ditukar dengan nilai fungsi lain (uang, dll).

Dalam hal ini, koin adalah string bit yang validitasnya dapat dengan mudah di cek oleh siapapun namun sulit untuk dibangkitkan. Setiap koin direpresentasikan dengan nilai *integer* k (digunakan *integer* 32 bit) sedemikian hingga nilai *hash*nya (misalnya dengan MD5) memiliki hasil urutan n bits yang sama. Hal ini dapat mengatasi kekurangan dari kunci publik yang kompleksitasnya cukup tinggi jika hanya digunakan untuk transaksi bernilai kecil, misalnya satu sen.

Broker akan memberikan koin-koin baru pada setiap awal bulan dan validitas koin-koin tersebut akan berakhir pada akhir bulan. Koin yang sudah tidak terpakai (oleh *user*) akan dikembalikan kepada *broker* pada setiap akhir bulan dan pada awal bulannya koin baru dapat kembali di beli. *Vendor* dapat mengembalikan koin yang telah dikumpulkan kepada *broker* sesuai keinginan, misalnya dalam jangka waktu perhari.

K-way collision

Andaikan fungsi *hash h* memetakan *m* (48 bit string) menjadi n (36 bit string). *k-way collision* adalah kumpulan dari sejumlah *k* nilai *x* ($x_1, x_2, \ldots, x_k$) yang memiliki nilai fungsi *hash y* yang sama sehingga $h(x_1) = h(x_2) = \ldots = h(x_k)$



Gambar 3 K-way collision

k-way collision ($x_1, x_2, \ldots, x_k$) merepresentasikan sebuah koin dan validitas dari koin ini mudah diverifikasi oleh siapapun dengan mengecek nilai setiap *x* yang berbeda dan nilai $h(x_1) = h(x_2) = \ldots = h(x_k) = y$.

Protokol skema *MicroMint* adalah sebagai berikut:

1. *broker* dapat memiliki/menetapkan profit atau keuntungan dalam setiap koin yang akan dijualnya kepada *user*
2. *broker* memilih nilai *k*, untuk *k-way collision*
3. Untuk membuat 2^u*-1* koin, *broker* harus memilih *u* sehingga *broker* membuat

array untuk 2^u tempat penyimpanan (bins), yang setiap elemennya dapat menyimpan hingga 4 nilai *x* yang akan dikomputasi dengan fungsi *hash* menjadi nilai *n* bit yang merupakan hasil penggabungan dari sejumlah *t* bit pola *z* yang sudah tetap dan sejumlah *u*-bit index dari tempat penyimpanan tsb.

4. *Broker* akan melemparkan 4 buah nilai pada tiap-tiap elemen array yang dengan demikian, *broker* telah membangkitkan sejumlah $4*(2^u) = 2^{u+2}$ nilai *x* yang akan menghasilkan nilai *y*. Karena setiap 2^u bins menghasilkan koin dengan probabilitas 0.5, maka jumlah koin yang dihasilkan pun akan berkisar 2^u. Pembuatan koin dilakukan dengan mekanisme *k-way collision.*

5. Setelah mencetak koinnya, *broker* menjual koin tersebut kepada *user*

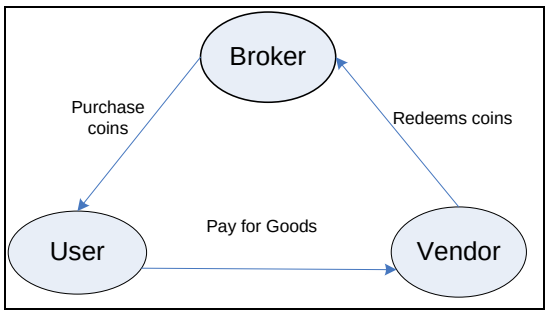
6. *User* akan membelanjakan koinnya kepada *vendor***.**

7. Setiap kali *user* membeli sebuah halaman web (atau content dari web), *user* akan memberi *vendor* koin yang belum dibelanjakan ($x_1, x_2, ..., x_k$). *Vendor* akan memverifikasi bahwa yang dibayarkan adlaah *k-way* collision dengan kondisi yang masih baik dengan menghitung nilai *hash* $h(x_i)$ untuk $1 \leq i \leq k$, dan mengecek kesamaan nilainya. Dalam hal ini, proses verifikasi oleh *vendor* dinilai efisien karena hanya membutuhkan *k* kali perhitungan nilai *hash* dan perbandingan.

8. *Vendor* akan menyimpan/menukarkan koin tersebut kepada *broker*.

9. *Vendor* akan mengembalikan koin yang telah dikumpulkannya dalam satu hari kepada *broker*. *Broker* mengecek setiap koin untuk melihat apakah sebelumnya pernah dikembalikan, dan jika tidak tagihan dari *vendor* akan dibayarkan (1 cent untuk setiap koin).

Gambar 4 berikut adalah gambaran umum mengenai skema *MicroMint*.



Gambar 4 Skema *MicroMint*

2.3 Perbandingan *payword* dan *MicroMint*

Berikut adalah perbandingan skema *payword* dan *MicroMint* sesuai dengan paparan tentang kedua skema ini:

1. *Payword* menggunakan kriptografi kunci publik dalam proporsi yang relatif kecil. *Vendor* membutuhkan verifikasi tanda tangan sekali sehari untuk setiap *user* yang melakukan pembelian.
2. Masing-masing *user* membangkitkan koin. Pada skema *MicroMint brokers* mengeluarkan nilai inisial *hash* yang akan digunakan pada koin, sebaliknya pada skema *payword brokers* hanya membuat *user certificates*.
3. *Payword* merupakan aplikasi yang berbasis kredit. Transaksi debit terhadap *user's account* dilakukan setelah *brokers* mengembalikan uang kepada *vendor* yang bersangkutan. Sebaliknya untuk *MicroMint*, *brokers* mengirimkan sejumlah koin kepada *user* per bulan.
4. Terkait dengan kebutuhan akan media penyimpanan, pada skema *payword broker* hanya membutuhkan penyimpanan akan sertifikasi *user*. Pada skema *MicroMint broker* membutuhkan media penyimpanan yang lebih banyak

untuk setiap nilai *hash* yang diproduksi setiap bulan.

5. Pada skema *payword*, *vendors* hanya butuh penyimpanan terhadap koin terakhir yang diterima *user*. Sebaliknya pada skema *MicroMint*, *vendors* harus menyimpan untuk setiap koin yang dibelanjakan.

3. *Security Concerns*

Pada poin ini isu keamanan dibagi menjadi dua bagian, yaitu serangan dengan skala kecil dan serangan dengan skala besar. Sebuah serangan dikatakan serangan dengan skala kecil jika angka kerugian yang disebabkan serangan tersebut kecil. Sebaliknya serangan besar jika angka kerugian relatif besar dibanding serangan skala kecil.

3.1 Pemalsuan

Pemalsuan dengan skala kecil memerlukan biaya yang sangat tinggi untuk dilakukan oleh pihak ketiga (lawan). Untuk menghasilkan sebuah koin palsu, alat pembayaran pada *micropayment*, pihak lawan membutuhkan 2^{45} operasi *hash*. Sementara itu komputasi yang biasa dilakukan oleh standar PC terbatas pada 2^{14} operasi *hash* per detik. Dengan demikian untuk memproduksi sebuah koin palsu, dibutuhkan 2^{31} detik $\supseteq$ 80 tahun.

Oleh karena itu pemalsuan yang dilakukan pihak lawan biasanya merupakan pemalsuan dengan skala besar. Untuk pemalsuan skala kecil biasanya dimungkinkan dilakukan oleh aksi curang *user*s dan atau *vendors*.

Pemalsuan skala besar dapat dideteksi dan dihitung sebagai berikut:

1. Semua koin palsu secara otomatis menjadi tidak valid pada akhir bulan
2. Koin palsu tidak dapat dibangkitkan hingga *broker* mengeluarkan kriteria validitas di awal bulan
3. *Broker* dapat mendeteksi adanya pemalsuan pada saat *broker* menerima koin yang tidak pernah dikeluarkan sebelumnya.
4. *Broker* dapat mendeklarasikan habisnya masa berlaku periode terkini sewaktu-waktu. Dengan demikian koin yang tersebar dapat ditarik kembali untuk kemudian diganti dengan koin yang menggunakan validasi yang baru.

3.2 Pencurian

Kemungkinan terjadinya pencurian koin adalah pada saat distribusi awal koin kepada *user* atau selama pengembalian uang dari *broker* ke *vendor*.

Untuk mencegah terjadinya pencurian ini, maka transmisi koin sedianya dilakukan dalam format yang telah dienkripsi. Mekanisme enkripsi yang dilakukan pada transmisi ini tidak menggunakan algoritma kriptografi kunci publik. Karena dari awal motivasi adanya *micropayment* dilatarbelakangi untuk menimilasisasi penggunaan kunci publik.

Salah satu alternatif yang digunakan sebagai pengganti kunci publik ini adalah dengan menggunakan mekanisme coins *user* spesifik. Detail mekanisme sesuai dengan

penjabaran yang ada pada sub poin *MicroMint* sebelumnya.

3.3 *Double spending*

Penggunaan koin yang sama dalam waktu yang berbeda dapat diidentifikasi *vendor* ketika menerima koin yang sama. *Vendor* dapat menentukan *user* mana yang berhak atas koin tersebut, karena *MicroMint* merupakan skema yang *non anonymous*.

Large scale cheater (*user* atau *vendor*) dapat diidentifikasi melalui pembelian dan penukaran koin (*redemption*). Dalam hal ini, maka *broker* dapat mengeliminasi penipu yang bersangkutan dari sistem. Sementara untuk *small scale cheater* susah untuk diidentifikasi.

4. Kelemahan skema *payword* dan *MicroMint*

Secara umum kelemahan dalam skema *payword* dan *MicroMint* terletak pada aspek keamanannya. Pada skema ini keamanan relatif kurang *robust* jika dibandingkan dengan algoritma kriptografi kunci publik.

Hal ini merupakan konsekuensi logis dari motivasi utama yang melatarbelakangi skema ini. Sebagaimana yang telah dipaparkan di awal bahwa *micropayment* disusun untuk menyederhanakan komputasi yang dilakukan dalam manajemen keamanan dengan cara meminimalkan penggunaan kriptografi kunci publik ini.

6. *Open issues*

Aplikasi *micropayment* merupakan aplikasi yang prospektif untuk jangka panjang. Pengembangan lebih lanjut kerangka desain *micropayment* sedianya harus memenuhi kriteria berikut [5]:

Kriteria teknis

Kriteria teknis terkait dengan struktur internal dan fungsionalitas dari sistem *micropayment*.

1. *Token-based* atau *account-based* menentukan media transmisi (pertukaran) data. Pada *token-based* menggunakan koin elektronik, sebaliknya *account-based* baik *user* maupun *vendor* harus memiliki *account* di bank.
2. Kemudahan penggunaan aplikasi, hal ini berhubungan dengan antarmuka aplikasi dan infrastuktur yang dibutuhkan.
3. *Anonymity* terhadap informasi *user* dan non-anonymous terhadap *vendor*.
4. *Scalability* yang berhubungan dengan kemampuan sistem *micropayment* dalam menambah jumlah *user* tanpa mengurangi performansi sistem.
5. *Validation,* terkait dengan kemampuan sistem untuk memproses pembayaran dengan atau tanpa *online* dengan pihak ketiga (*broker*). Validasi *online* berarti semua bagian ikut terlibat dalam setiap pembayaran. Validasi semi-*online* berarti setiap bagian ikut terlibat, tetapi tidak untuk setiap pembayaran. Validasi offline berarti pembayaran dapat dilakukan tanpa pihak ketiga (pembayaran tunai).
6. *Security,* mencegah dan mendeteksi serangan pada sistem pembayaran dan kasus penipuan, serta melindungi informasi mengenai pembayaran yang dilakukan. *Security* dibutuhkan karena kasus penipuan dan serangan terhadap

sistem pembayaran melalui internet sering ditemukan.

7. *Interoperability,* memungkinkan *user* yang berasal dari satu sistem pembayaran tertentu dapat membayari atau dibayari oleh *user* dari sistem pembayaran yang berbeda.

Kriteria Non Teknis

Kriteria ini berkaitan dengan aspek ekonomis dan kegunaan dari sistem *micropayment*.

1. *Trust,* memberi jaminan kepercayaan kepada *user* terhadap sistem *micropayment* dan operatornya. Hal ini dapat dilakukan dengan meningkatkan teknik-teknik keamanan yang terkait dengan sistem.
2. *Coverage,* menampilkan persentase dari *vendor* dan *user* yang menggunakan sistem pembayaran ini.
3. *Privacy,* berhubungan dengan perlindungan informasi personal dan pembayarannya.
4. *Pre-paid or post-paid,* menentukan bagaimana *user* menggunakan sistem pembayaran. Dalam sistem *pre-paid*, customer harus mengirimkan uang ke sistem terlebih dahulu sebelum melakukan inisiasi *micropayment*.
5. *Range of payments* dan *multicurrency support,* menspesifikasikan nilai maksimum dan minimum dari pembayaran yang didukung oleh sistem, serta menentukan apakah sistem mendukung *multiple currency*

5. Kesimpulan

Micropayment merupakan salah satu instansiasi aplikasi perekonomian berbasis internet. *Micropayment* merupakan salah satu bentuk pembayaran yang dilakukan secara elektronik[2].

Penggunaan *micropayment* ini dikhususkan pada pembayaran dengan nominal relatif kecil dan frekuensi transaksi dengan intensitas yang tinggi.

Terdapat dua buah skema *micropayment* yang diajukan oleh Rivest dan Shamir, yaitu skema *Payword* dan *MicroMint* dengan kelebihan dan kekurangannya masing-masing. Kedua skema ini melibatkan tiga buah entitas yaitu *user*, *vendor*, dan *broker* sebagai pihak ketiga. Dalam skema ini aspek keamanan tidak terlalu ditekankan karena lebih memperhatikan aspek efisiensi pembayaran skala kecil. Skema yang diajukan tersebut memiliki implementasi yang berbeda-beda untuk setiap aplikasi.

[1] *Internet Usage Statistics*, http://www.InternetWorldStats.com, diakses tanggal 4 Januari 2006.
[2] *Micro payment definition*, http://research.compaq.com/src/millicent/html/glossary.html, diakses tanggal 5 Januari 2006.
[3] R. Rivest and A. Shamir, *Payword and MicroMint: Two simple Micropayment Schemes,* http://theory.lcs.mit.edu/~cis/pubs/rivest/RivestShamir-mpay.ppt, diakses tanggal 2 Januari 2006.
[4] Third International Conference on Financial Cryptography (FC '99) February 22--25, 1999 Anguilla, British West Indies, http://www.ieee-security.org/Cipher/ConfReports/1999/CR1999-FC99.html, diakses tanggal 5 January 2006
[5] R. Parhonyi, *Second Generation Micropayment System*, University of Twente, Faculty of Engineering, Mathematics and Computer Science, Netherland.