

Penerapan Fungsi Hash pada URL Shortening Service

Biyani Satyanegara / 13508057
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
if18057@students.if.itb.ac.id

Abstract— Makalah ini berisi tentang penerapan fungsi hash pada URL Shortening service. Latar belakang pembuatan makalah ini adalah banyaknya penggunaan URL pada internet sebagai sumber dan penyebaran informasi. Namun alamat web dinilai terlalu panjang sehingga saat ini pada internet banyak bermunculan URL Shortening service. URL shortening service adalah aplikasi untuk meringkas URL yang saat ini sering digunakan untuk mempermudah penyebaran informasi dalam dunia internet. Fungsi hash merupakan algoritma untuk membentuk string unik yang menerima masukan berupa string. Hasil keluaran dari fungsi hash ini adalah sebuah string heksadesimal yang unik. Selain itu URL shortening service juga menggunakan tabel hash yang digunakan untuk membantu pencarian dari kunci hash yang telah dihasilkan. Makalah ini bertujuan untuk mengetahui penggunaan fungsi hash pada aplikasi URL shortening dan untuk mengetahui cara kerja dari aplikasi URL shortening. Selain itu makalah ini juga akan membahas efektifitas penggunaan fungsi hash sebagai salah satu metode dalam URL shortening. Makalah ini hanya akan membahas metode URL shortening yang menggunakan fungsi hash. Terdapat berbagai macam metode dalam URL shortening service yang memanfaatkan fungsi hash. Namun dari beberapa metode tersebut metode yang paling cocok untuk menggunakan fungsi hash adalah metode yang menggunakan tabel hash. Kesimpulan yang didapatkan dari hasil analisis adalah fungsi hash merupakan metode yang penting dalam URL shortening. Fungsi hash berguna untuk membentuk kunci yang unik pada setiap halaman atau alamat yang dituju.

Kata kunci — hash, kunci, string, tabel hash dan URL.

I. PENDAHULUAN

Saat ini internet menjadi salah satu kebutuhan primer manusia. Internet digunakan untuk sebagai sumber referensi manusia terutama dalam mencari informasi. Tidak hanya menjadi sumber informasi, internet kini menjadi media yang efektif untuk penyebaran dan publikasi informasi. Proses pencarian dan penyebaran informasi ini terkait erat dengan penggunaan URL pada internet. URL (Uniform Resource Locator) merupakan serangkaian karakter menurut suatu format standar tertentu yang digunakan untuk menunjukkan alamat suatu sumber dokumen dan gambar di internet. Pengguna harus mengetahui URL yang benar agar dapat menuju alamat

dari informasi yang dituju. Namun terkadang URL yang digunakan terlalu panjang dan rumit sehingga terkadang menyulitkan pengguna untuk berbagi informasi. Oleh karena itu, saat ini terdapat aplikasi baru yang beredar di internet yaitu URL shortening service seperti : bit.ly, tinyurl dll. Service ini berguna untuk meringkas URL menjadi karakter yang lebih ringkas dibandingkan URL yang asli. Aplikasi ini memudahkan dan membantu pengguna untuk berbagi informasi pada internet.

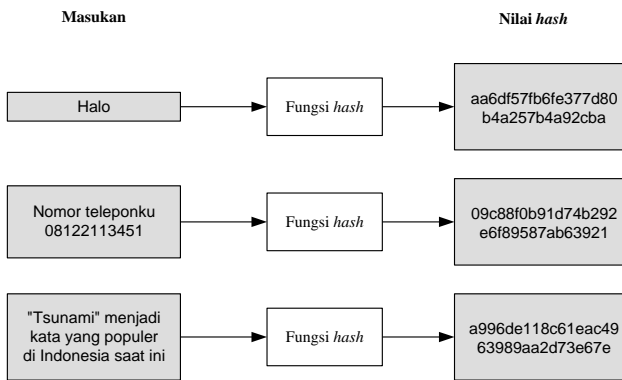
Fungsi hash merupakan sebuah algoritma yang digunakan untuk mengubah suatu string menjadi bentuk unik yang memiliki ukuran tetap. Hal ini dimanfaatkan untuk menjadikan keunikan dari setiap URL baru yang dihasilkan.

URL shortening service dalam penerapannya menggunakan fungsi hash sebagai salah satu langkah untuk menyederhanakan URL tersebut. Fungsi utama hash dalam URL shortening service adalah untuk menghasilkan kunci yang unik dari tiap alamat yang dituju. Makalah ini akan membahas penggunaan fungsi hash pada sebuah aplikasi URL shortening service. Batasan masalah yang akan dikaji adalah terbatas pada penerapan fungsi hash yang digunakan saat ini pada URL shortening device dan analisis efektifitas penggunaan jenis-jenis fungsi hash yang telah ada pada aplikasi tersebut.

II. LANDASAN TEORI

A. Fungsi Hash

Fungsi hash adalah fungsi yang menerima masukan string yang panjangnya sembarang lalu mentransformasikannya menjadi string keluaran yang panjangnya tetap (fixed). Umumnya hasil dari fungsi hash berukuran jauh lebih kecil daripada ukuran string semula.



Gambar 1 Contoh fungsi hash pada string

Persamaan fungsi hash :

$$h = H(M)$$

Keterangan :

- M = pesan ukuran sembarang
- h = nilai hash atau pesan ringkas (message-digest)
- H = fungsi hash

Contoh : size (M) = 1 MB → size(h) = 128 bit

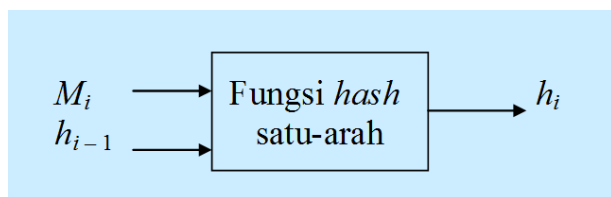
Fungsi hash adalah fungsi satu arah yaitu pesan yang sudah diubah menjadi message digest tidak dapat dikembalikan lagi menjadi pesan semula (irreversible). Sifat-sifat fungsi hash satu-arah adalah sebagai berikut:

1. Fungsi H dapat diterapkan pada blok data berukuran berapa saja.
2. H menghasilkan nilai (h) dengan panjang tetap (fixed-length output).
3. H(x) mudah dihitung untuk setiap nilai x yang diberikan.
4. Untuk setiap h yang dihasilkan, tidak mungkin dikembalikan nilai x sedemikian sehingga H(x) = h. Itulah sebabnya fungsi H dikatakan fungsi hash satu-arah (one-way hash function).
5. Untuk setiap x yang diberikan, tidak mungkin mencari y ≠ x sedemikian sehingga H(y) = H(x).
6. Tidak mungkin mencari pasangan x dan y sedemikian sehingga H(x) = H(y).

Masukan fungsi hash adalah blok pesan (M) dan keluaran dari hashing blok pesan sebelumnya,

$$H_i = H(M_i, h_{i-1})$$

Skema fungsi hash ditunjukkan pada gambar berikut :



Gambar 2 Skema fungsi hash satu arah

Ada beberapa fungsi hash satu-arah yang sudah dibuat orang, antara lain:

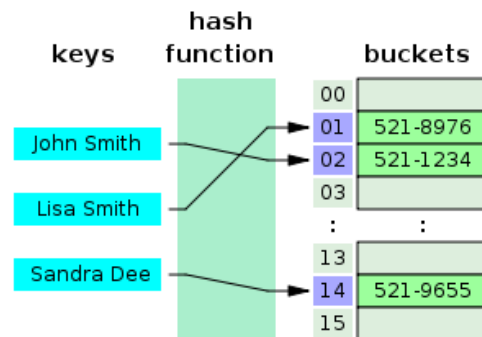
- MD2, MD4, MD5,
- Secure Hash Function (SHA),
- Snefru,
- N-hash,
- RIPE-MD, dan lain-lain

Fungsi hash satu arah memiliki beberapa aplikasi dalam penggunaannya seperti berikut :

1. Menjaga integritas data
Fungsi hash sangat peka terhadap perubahan 1 bit pada pesan. Pesan berubah 1 bit, nilai hash berubah sangat signifikan.
2. Menghemat waktu pengiriman
Mengirimkan pesan hanya dengan menggunakan message digestnya. Jika message digest salinan arsip sama dengan message digest arsip asli berarti salinan arsip tersebut sama dengan arsip master.
3. Menormalkan panjang data yang beraneka ragam
Panjang nilai hash tetap sehingga panjang tiap string yang dihasilkan selalu sama berapa panjang masukan yang diberikan.

B. Tabel Hash

Tabel hash adalah struktur data yang menggunakan fungsi hash untuk memetakan kunci dengan nilai-nilai yang berhubungan dengan kunci tersebut. Tabel hash mengimplementasikan associative array. Fungsi hash digunakan untuk mengubah kunci menjadi indeks dari elemen array yang berhubungan langsung dengan nilai yang ada.



Gambar 3 Skema tabel hash

Fungsi hash seharusnya dapat memetakan setiap kemungkinan kunci ke suatu indeks tertentu, tetapi keadaan ideal ini jarang terjadi kecuali kunci hash memiliki jumlah yang pasti. Contohnya : data baru tidak pernah ditambahkan setelah tabel dibuat. Kebanyakan desain tabel hash mengasumsikan hash collisions (kunci yang berbeda mengacu pada nilai hash yang sama) pasti terjadi dan harus ditangani dengan cara tertentu.

Algoritma inti dari tabel hash adalah menggunakan array sederhana yang terdiri dari beberapa item. Algoritma tabel hash menghitung indeks dari kunci yang dimiliki oleh suatu benda dan menggunakan indeks tersebut untuk menempatkan benda tersebut ke dalam array. Berikut ini adalah implementasi dari fungsi hash :

$$Index = f(key, arrayLength)$$

Fungsi hash menghitung indeks berdasarkan kunci data dan ukuran array.

Hash collision tidak dapat dihindari ketika melakukan hash secara acak pada kunci yang banyak. Contohnya jika hash dilakukan pada 2500 kunci, walaupun dilakukan dengan uniform random distribution masih terdapat kemungkinan dua kunci mengacu pada tempat yang sama. Oleh karena itu, sebagian besar implementasi tabel hash memiliki strategi untuk penanganan hash collision. Semua metode ini membutuhkan kunci yang disimpan di dalam tabel bersama-sama dengan nilai yang berhubungan.

Performansi dari sebagian besar metode collision resolution tidak hanya bergantung dengan jumlah n item yang disimpan, tetapi juga bergantung pada tabel load factor. Rasio n/s antara n dan ukuran s dari array buckets. Terkadang hal ini dianggap sebagai fill factor, yang dapat merepresentasikan bagian dari s buckets di dalam struktur yang diisi dengan n data. Dengan menggunakan fungsi hash yang baik, rata-rata nilai lookup akan menjadi konstan seperti load factor yang bernilai dari 0 sampai 0.7. Jika lebih dari nilai itu maka kemungkinan untuk terjadi collision akan meningkat. Namun jika load factor mendekati nilai 0, ukuran dari tabel hash meningkat dengan sedikit perbaikan pada search cost dan menyianiyakan memori. Terdapat dua macam metode untuk melakukan collision resolution yaitu separate chaining dan open addressing.

Untuk menjaga agar load factor dibawah batas tertentu, banyak implementasi tabel memperluas tabel ketika data baru masuk. Dikarenakan bucket biasanya diimplementasikan pada awal dari dynamic array dan semua constant proportion untuk pengukuran ulang lebih besar daripada 1 akan menjaga load factor tetap dibawah batas yang diinginkan. Resizing akan diikuti oleh tabel incremental baru yang diikuti dengan pemetaan benda ke lokasi bucket yang baru. Untuk membatasi agar memori tidak digunakan sia-sia oleh bucket yang kosong, beberapa implementasi menyusutkan ukuran dari tabel yang dibuat ketika benda tertentu dihapus. Operasi ini mirip dengan dealokasi pada dynamic array.

III. PENGGUNAAN FUNGSI HASH PADA URL SHORTENING SERVICE

URL Shortening service adalah proses yang menterjemahkan alamat website yang panjang menjadi nama URL yang singkat dan mudah untuk digunakan kembali. Alamat URL yang baru ini disediakan untuk menghubungkan pengguna dengan alamat website yang asli. Aplikasi URL shortening semakin banyak di internet seperti : is.gd, tinyurl.com, bilt.ly, snipurl.com dll. Berikut ini adalah contoh penggunaan aplikasi URL shortening service :

URL yang ingin diubah :

<http://www.howtogeek.com/howto/windows-vista/what-is-cfmonexe-and-why-is-it-running/>

Hasil penggunaan URL Shortening service :

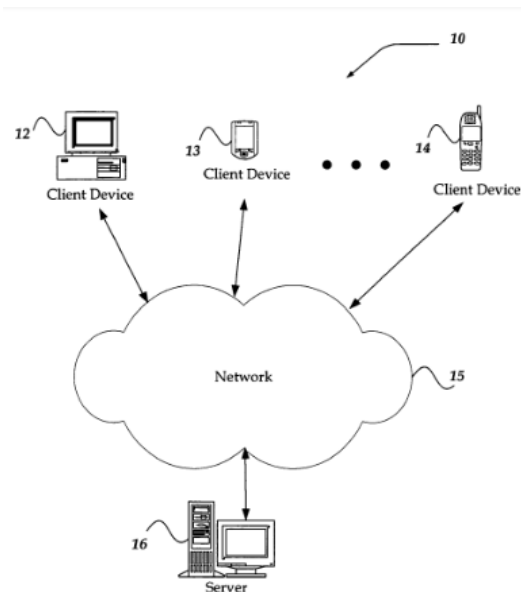
<http://is.gd/4E3EN>
<http://tinyurl.com/2th782>
<http://bit.ly/ApWlc>
<http://snipurl.com/sth8b>

URL shortening service akan mengubah URL yang panjang menjadi URL yang lebih singkat dan sederhana. Alamat URL yang singkat ini ketika digunakan akan berubah menjadi bentuk yang asli sebelum disingkat dan dapat berfungsi untuk menghubungkan alamat website yang dituju.

Terdapat beberapa cara penerapan URL shortening service. Cara penerapan yang paling sederhana adalah dengan menggunakan fungsi hash. Cara ini hanya mengubah alamat URL yang asli menjadi bentuk heksadesimal. URL yang dihasilkan bergantung pada fungsi hash yang digunakan. Berikut ini adalah contoh penggunaan fungsi hash dengan menggunakan masukan salah satu URL tertentu :

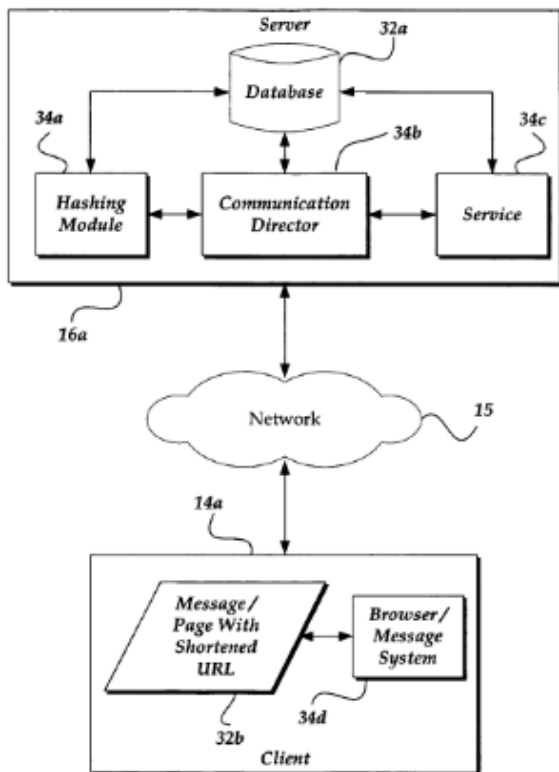
Fungsi Hash	Message digest
Adler32	399014e3
CRC32	78aa9d1a
MD2	286c50c2db4fcad77adb4edeb3a937b2
MD4	387ac3f6aae7956c4fab176271bb4518
MD5	f061a171dfc30635462850684f98b886
SHA-1	3c93b6d332091b2970fb660d644d0ba3d756e322

Selain dengan menggunakan fungsi hash, URL shortening service menggunakan metode lain dalam penerapannya. Salah satunya adalah dengan menggabungkan antara fungsi hash dengan tabel hash. Cara kerja URL Shortening service dengan menggunakan tabel hash untuk mengakses link dari sebuah shortened URL digambarkan sebagai berikut :



Gambar 4 Ilustrasi Hubungan device dengan aplikasi URL shortening

Gambar 4 menggambarkan lingkungan aplikasi URL Shortening dapat berjalan. Pengguna dapat menggunakan aplikasi URL shortening dengan menggunakan alat-alat tertentu seperti komputer, handphone atau alat-alat lain yang terhubung dengan suatu jaringan tertentu. Jaringan tersebut juga harus terhubung dengan suatu server tempat dimana aplikasi URL shortening diaplikasikan. Umumnya jaringan yang digunakan adalah internet sehingga seluruh peralatan yang terhubung dengan internet dapat menggunakan aplikasi URL Shortening tersebut. Selain itu jaringan tersebut juga dapat menggunakan wireless, LAN, USB port dll. Berikut ini adalah gambar detail arsitektur URL shortening device :



Gambar 5 Arsitektural pembentuk aplikasi URL Shortening

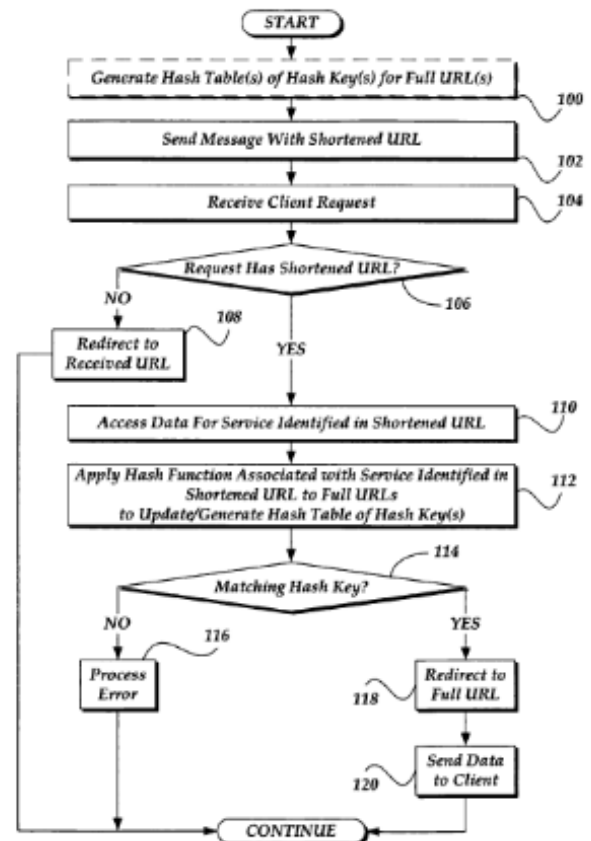
Langkah pertama yang dilakukan adalah modul hash memeriksa URL asli yang berhubungan dengan alamat web tertentu ataupun suatu layanan tertentu. Modul hash menggunakan fungsi hash untuk tiap URL asli ataupun potongan dari URL asli untuk menghasilkan kunci hash. Sebuah URL asli dapat terdiri dari jalur yang kompleks, string query atau elemen lain. Modul hash mengaplikasikan fungsi hash pada bagian tertentu dari URL string untuk membentuk kunci hash. Modul hash dapat menggunakan fungsi hash yang berbeda pada setiap layanan dan karakteristik tertentu. Suatu karakteristik dapat dijadikan parameter untuk fungsi hash. URL asli dan kunci hash dapat dipetakan menjadi satu atau beberapa tabel hash sebagai referensi. Identifikasi servis, indentifikasi user, status sign-in dan data lain dapat digunakan pada tabel hash. Penambahan, pengubahan dan perubahan pada URL dapat ditambahkan pada tabel hash saat perubahan terjadi ataupun ditambahkan setelah

permintaan URL baru. Tabel hash dapat disimpan pada periode tertentu ataupun disimpan permanen. Beberapa kunci hash dapat dihasilkan tanpa disimpan.

URL baru dibuat dan dikirimkan pada halaman web, sms ataupun berbagai bentuk pesan yang diberikan kepada pengguna. Pada umumnya URL baru menyimpan kunci hash dari URL yang lama. URL baru juga terdiri dari kode yang merepresentasikan alamat ataupun layanan yang diberikan pada URL asli. Nama domain dari URL baru mengacu pada alamat jaringan dari URL asli. Contoh struktur dari URL baru sebagai berikut :

http://domain_name/property_code/hash_key

URL baru pada umumnya memiliki ukuran bytes yang lebih kecil jika dibandingkan dengan URL lama. URL baru akan memiliki ukuran yang tetap, walaupun terkadang URL yang dihasilkan dapat lebih singkat ataupun lebih panjang dari sebelumnya. URL baru digunakan pada device pengguna dan biasanya berupa hyperlink yang langsung terhubung dengan alamat web yang dituju.



Gambar 6 Langkah URL shortening

Pada saat permintaan URL dari pengguna masuk maka URL tersebut akan diperiksa apakah mengandung format URL tertentu. Jika URL yang diberikan tidak mengandung format URL yang baru maka URL tersebut akan dialihkan pada proses tertentu. Jika URL yang dimasukkan mengandung format URL baru tertentu maka URL yang dimasukkan akan dipecah berdasarkan format tertentu. Informasi yang didapatkan dari URL yang dimasukkan akan dicocokkan dengan dengan tabel hash

yang telah dibuat pada awal proses. Jika tidak ada tabel hash yang cocok atau tabel hash perlu diupdate maka modul hash akan mengakses URL asli yang terhubung dengan alamat website tertentu. Identifikasi user, status sign-in dan karakteristik lain dapat digunakan sebagai akses untuk menggunakan tabel hash, URL asli dan alamat/servis web tertentu.

Langkah selanjutnya adalah mengevaluasi kunci hash pada tabel hash. Modul hash mengaplikasikan fungsi hash ke URL lama secara real time untuk memperbaharui tabel hash yang berhubungan ataupun membuat tabel hash berdasarkan kunci hash yang dihasilkan. Kemudian kunci hash dari URL yang masuk akan dibandingkan dengan kunci hash dari tabel hash. Pada proses ini juga terdapat validasi berdasarkan identifikasi user, sign-in status dan karakteristik lain yang dapat menjadi alat pembandingan antara tabel hash dengan permintaan URL yang dimasukkan oleh pengguna. Jika tidak ditemukan URL yang cocok pada tabel hash ataupun validasi tidak terpenuhi maka akan menunjukkan perintah error pada operasi tersebut. Jika URL yang dimasukkan sesuai maka pengguna akan dihubungkan dengan operasi yang ada pada URL semula. Servis yang terhubung dengan URL lama akan dijalankan dan ditampilkan kepada pengguna.

IV. ANALISIS PENGGUNAAN FUNGSI HASH

Aplikasi URL shortening service masih menggunakan fungsi hash dalam salah satu metodenya. Pada umumnya hal ini dilakukan untuk mendapatkan kunci hash sehingga hasil yang didapatkan unik pada setiap alamat yang ingin diringkas. Fungsi hash sangat efektif dalam menghasilkan string yang unik. Hal ini dibuktikan apabila pada URL masukan pengguna diubah satu string saja maka akan mengubah seluruh heksadesimal yang dihasilkan. Namun fungsi hash ini juga memiliki beberapa kelemahan dalam penerapannya. Kelemahan fungsi hash antara lain :

1. Hasil memiliki ukuran yang sama
URL yang dimasukkan akan menghasilkan besar dan panjang string yang sama. Hal ini berguna jika URL yang dimasukkan memiliki string yang cukup panjang. Namun jika string masukkan URL lebih singkat dibandingkan dengan string hasil fungsi hash maka alamat URL akan diubah menjadi lebih panjang dari semula. Hal ini tentu saja tidak sesuai dengan tujuan dari URL shortening service yang berfungsi untuk meringkas alamat URL yang dimasukkan.
2. Fungsi Hash bersifat satu arah
Hasil dari fungsi hash merupakan string heksadesimal yang tidak dapat dikonversi lagi menjadi bentuk sebelumnya. Hal ini tentu saja tidak sesuai dengan prinsip dari URL Shortening Service yang bertujuan untuk meringkas alamat web dari sebuah URL tanpa menghilangkan fungsi URL tersebut untuk menghubungkan dengan suatu alamat tertentu.

Oleh karena itu, penggunaan fungsi hash dalam URL shortening service biasanya digabungkan dengan metode lain. Fungsi hash tidak dapat berdiri sendiri untuk aplikasi URL shortening service. Pada aplikasi yang ada saat ini penggunaan fungsi hash digabungkan metode lain seperti tabel hash ataupun menggunakan indexing pada database. Penggunaan database ini membutuhkan memori yang besar dan waktu yang lama untuk pencarian. Namun metode yang cocok untuk diaplikasikan dengan fungsi hash adalah dengan menggunakan tabel hash. Metode ini dapat menjaga keunikan dari suatu URL tertentu dengan kunci hash yang telah dihasilkan oleh fungsi hash. Pencarian juga dimudahkan dengan menggunakan indeks dari kunci yang dihasilkan sebelumnya. Walaupun terdapat kesulitan yaitu pada saat ada penambahan atau perubahan tabel hash baru dikarenakan ada URL yang belum terdaftar. Metode tabel hash juga memiliki kekurangan yaitu membutuhkan memori yang besar karena semua URL harus disimpan sebagai referensi untuk pencocokkan antara URL baru dan alamat web yang dituju.

V. KESIMPULAN

Berdasarkan hasil analisis dapat diambil beberapa kesimpulan :

1. Fungsi hash digunakan dalam URL Shortening service untuk pembentukan kunci yang unik
2. Metode URL yang menggunakan fungsi hash ada tiga yaitu : fungsi hash sederhana, penggunaan database dan tabel hash
3. Metode yang paling baik digunakan adalah metode tabel hash karena dapat memaksimalkan penggunaan fungsi hash dengan pencarian konten yang sesuai

REFERENCES

- Munur, Rinaldi. 2005, "Diktat Kuliah IF2251 Strategi Algoritmik". Bandung : Institut Teknologi Bandung
<http://www.codinghorror.com/blog/2007/08/url-shortening-hashes-in-practice.html>
Tanggal Akses : 6 Mei 2011
- <http://www.docstoc.com/docs/78275739/URL-Shortening-And-Authentication-With-Reverse-Hash-Lookup---Patent-7698269>
Tanggal Akses : 5 Mei 2011
- <http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/hashTable.htm>
Tanggal Akses : 4 Mei 2011
- <http://www.codeproject.com/KB/webservices/url-shortening-svcs-bitly.aspx>
Tanggal Akses : 4 Mei 2011
- <http://stackoverflow.com/questions/2253693/shortening-rehashing-uuids>
Tanggal Akses : 4 Mei 2011
- <http://www.articledashboard.com/Article/How-Does-URL-Shortening-Work-/1294571>
Tanggal Akses : 4 Mei 2011

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Mei 2011

A handwritten signature in black ink, appearing to read 'Biyan', with a large, sweeping flourish underneath.

Biyan Satyanegara / 13508057