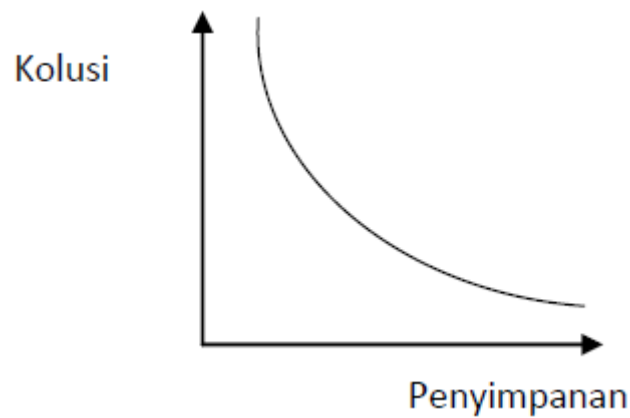


MANAJEMEN KOLISI

Salah satu fungsi hash adalah akan mendistribusikan data secara merata ke dalam berkas. Jika tujuan tersebut tidak tercapai, salah satu strategi yang bisa diambil adalah mengkombinasikan beberapa fungsi sederhana dalam satu aplikasi. Fungsi hash menghasilkan banyak kolisi atau sinonim dikatakan memiliki kluster primer. Makin sedikit jumlah kolisi, makin baik fungsi hashing tersebut karena makin sedikit waktu yang diperlukan rekaman yang diinginkan, dan juga akan mempertahankan probe atau akses terhadap penyimpanan agar mendekati satu. Meminimalkan jumlah probe pembacaan rekaman merupakan isu yang sangat penting mengingat berbagai aplikasi komersial pada umumnya memiliki volume data yang sangat besar sehingga data tersebut harus disimpan di dalam penyimpanan luar (auxiliary). Beberapa cara yang dapat ditempuh untuk mereduksi kolisi adalah mengganti fungsi hashing, atau dengan mereduksi factor-packing. Factor-packing suatu berkas adalah perbandingan (atau rasio) antara jumlah rekaman yang disimpan dalam berkas dengan kapasitas berkas, atau dapat dinyatakan sebagai berikut:

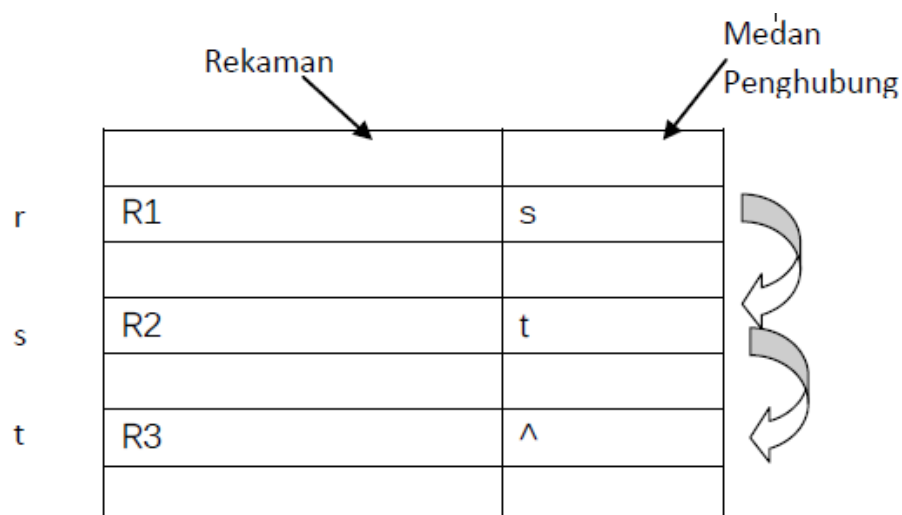
$$\text{Factor-packing (FP)} = \frac{\text{Jumlah rekaman yang disimpan}}{\text{Jumlah total lokasi penyimpanan}}$$

Factor-packing (sinonim dengan densitas-packing atau factor-loading) mengukur utilitasi penyimpanan. Factor-packing berbanding lurus dengan jumlah kolisi. Jika nilai packing-factor meningkat, maka kolisi akan lebih sering terjadi. Mengurangi nilai factor-packing dengan tujuan untuk mengurangi jumlah kolisi, membawa pada konsekuensi diperlukannya ruang yang lebih luas untuk menyimpan jumlah rekaman yang sama. Fenomena pemilihan factor-packing adalah contoh dari suatu trade-off waktu dan ruang. Jika factor-packing diperkecil, maka diperlukan ruang yang lebih luas (dan ini mahal), tetapi dengan demikian kemungkinan untuk kolisi menjadi berkurang atau dengan kata lain meningkatkan kinerja. Kolisi pada umumnya akan naik secara cepat bila factor-packing melebihi 90%. Hubungan antara jumlah kolisi dengan tempat penyimpanan dapat diilustrasikan dengan grafik berikut:



Mengubah fungsi hashing atau mengubah factor-packing akan dapat mengurangi jumlah kolusi. Akan tetapi, pada umumnya tidak akan mengeliminasi kolusi tersebut. Oleh karena itu, diperlukan suatu prosedur untuk menempatkan sinonimnya ke posisi lain bila kolusi memang benar terjadi. Yang menjadi tujuan utama metoda resolusi kolusi adalah menempatkan rekaman sinonim pada suatu lokasi yang membutuhkan probe tambahan yang minimum dari home-address rekaman tersebut. Perlu diingat bahwa probe adalah akses ke lokasi yang berbeda.

Salah satu penyelesaian yang dapat dilakukan adalah memberikan penunjuk pada lokasi rekaman sinonim. Bila terjadi sinonim jamak pada satu home-address tertentu, akan dibentuk rantai rekaman sinonim. Sebagai contoh adalah R1, R2, dan R3 sinonim dengan home-address r maka rantai sinonim akan berbentuk seperti dilihatkan pada gambar 9.1. Medan penghubung lokasi r menunjuk pada lokasi s di mana tersimpan sinonim pertama, yaitu R2. Simbol \wedge atau penunjuk nol pada medan penghubung t menunjukkan akhir dari rantai sinonim.



Gambar 9.1 Rantai sinonim

A. Coalesed-Hashing

Coalesed-Hashing adalah metode resolusi yang menggunakan penunjuk untuk menghubungkan elemen-elemen dari sebuah rantai sinonim. Sebagai ilustrasi, perhatikan gambar 9.1 yang memperlihatkan rekamaa-rekaman R1, R2, dan R3 yang semuanya memiliki home-address r. R1 disisipkan pertama dan memperoleh lokasi sebagai home-addressnya yaitu r. Pada saat menyisipkan R2, terjadilah kolusi karena hasil fungsi dari R2 juga r. Hal tersebut mengisyaratkan agar R2 diletakkan pada lokasi lain karena r sudah ditempati R1. Pada saat tersebut tersedia ruang di s, maka R2 akan disisipkan pada lokasi tersebut dan penunjuk pada medan penghubung r diatur agar menunjuk ke s. Kolisi terjadi kembali pada saat R3 akan disisipkan. Rekaman R3 kemudian disisipkan pada ruang yang masih tersedia, yaitu t, dan penunjuk di medan penghubung s diarahkan ke t, sementara medan penghubung t memiliki symbol ^ yang berarti akhir dari rantai sinonim akan dibaca sampai rekaman ditemukan atau sampai terbaca akhir mata rantai sinonim.

Coalesced-hashing (kolisi) terjadi bila terdapat usaha untuk menyisipkan sebuah rekaman dengan home-address yang sudah di okupasi oleh rekaman dari rantai yang memiliki home-address yang berbeda. Misal terdapat rekaman R4 yang hasil hashnya adalah s. Kolisi akan terjadi pada saat diinginkan untuk menyisipkan R4 pada alamat s ke dalam tabel seperti gambar 9.1, yang terjadi adalah dua rantai sinonim dengan rekaman-rekaman yang memiliki home-address berbeda mengalami kolisi. Algoritma untuk coalesced-hashing adalah sebagai berikut:

- I. Lakukan hashing pada semua kunci rekaman yang akan disisipkan untuk mendapatkan home-address atau calon-address yang mungkin akan ditempati oleh rekaman-rekaman tersebut.
- II. Jika home-address kosong, sisipkan rekaman pada lokasi tersebut, jika rekaman ternyata kembar, akhiri program dengan pesan ,Rekaman Kembar`, jika tidak:
 - a) Cari lokasi terakhir rantai-sinonim dengan mengikuti penunjuk pada medan-penghubung sampai menemukan symbol ^ yang menandakan akhir dari rantai.
 - b) Cari lokasi paling bawah dalam berkas (atau memiliki alamat paling besar). Jika tidak ditemukan, akhiri program dengan pesan ,Berkas Penuh`.

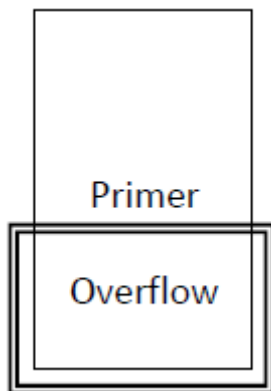
- c) Sisipkan rekaman ke dalam lokasi yang kosong sudah teridentifikasi dan atur medan-penghubung rekaman terakhir dalam rantai-sinonim agar menunjuk ke lokasi rekaman yang baru saja disisipkan.

B. LICH dan EISCH

Berbagai usaha telah dilakukan untuk mengurangi kolisi pada rantai sinonim sehingga mengurangi probe pembacaan dan akhirnya meningkatkan kinerja. Beberapa varian dari coalesced-hashing dapat diklasifikasikan ke dalam tiga cara:

- ❑ Mengorganisasi berkas (dengan atau tanpa overflow)
- ❑ Menghubungkan item yang terkolisi kedalam rantai
- ❑ Memilih lokasi yang belum ada penghuninya

Kolisi mungkin dapat direduksi dengan memodifikasi organisasi berkas. Cara pertama adalah dengan memisahkan antara area untuk data prima dengan area untuk data overflow, sehingga memiliki bentuk sebagai berikut:



Area primer adalah ruang alamat yang cocok dengan fungsi hash. Overflow (atau Cellar) adalah area yang hanya berisi rekaman-rekaman yang sinonim. Faktor alamat adalah perbandingan antara area primer dengan ukuran total berkas.

$$\text{Faktor alamat} = \frac{\text{Area Primer}}{\text{Ukuran total berkas}}$$

Untuk ukuran berkas yang tetap, jika faktor alamat mengecil maka ukuran cellar akan meningkat, yang berarti mereduksi kolisi. Akan tetapi, karena area primer menjadi mengecil maka jumlah kolisi akan membesar. Kolisi yang lebih banyak membawa konsekuensi pada diperlukannya probe pembacaan yang lebih besar. Algoritma yang diimplementasikan pada coalesced-hashing sebagaimana dilakukan pada contoh terdahulu disebut 'Late Insertion Standard Coalesced Hashing' atau LISCH, karena rekaman yang baru disisipkan pada akhir rantai sinonim. Kata 'standard' memiliki arti 'tidak adanya cellar'. Varian dari algoritma tersebut adalah LICH atau 'Late Insertion Standard Coalesced Hashing'.

C. Progressive Overflow

Kerugian utama penggunaan coalesced-hashing adalah diperlukannya penyimpanan tambahan untuk medan penghubung. Bila penyimpanan tambahan tersebut tidak tersedia, maka penghubung yang sifatnya fisik tidak tersedia, sehingga perlu dipertimbangkan teknik resolusi kolisi yang menggunakan konversi untuk menentukan kemana selanjutnya rekaman harus dicari. Salah satu konversi yang sederhana adalah penggunaan overflow yang progresif atau probing secara linier. Sesuai namanya bila lokasi yang akan ditempati telah terisi, maka lokasi selanjutnya dilihat apakah masih belum terisi. Secara progresif lokasi selanjutnya di overflow. Pertanyaannya adalah, apa yang akan terjadi bila overflow dilakukan sampai pada alamat tertinggi, atau akhir berkas? Pertimbangan berkas memiliki struktur yang melingkar, dengan lokasi pertama berada tepat sesudah lokasi terakhir. Pencarian dilanjutkan sampai ditemukan slot yang kosong atau sampai ditemukan home-address rekaman untuk kedua kalinya. Menandakan bahwa berkas telah penuh. Untuk pembacaan kembali, dilakukan proses yang sama. Dan bagaimana dengan pencarian yang tidak berhasil? Pencarian harus terus dilakukan sampai dijumpai slot kosong atau alamat pencarian yang pertama. Jelas bahwa algoritma untuk pencarian yang tidak berhasil memiliki kinerja yang tidak bagus, tetapi algoritma tersebut sangat sederhana.

D. Penggunaan Buckets

Dalam diskusi mengenai prosedur resolusi kolisi, diasumsikan bahwa hanya sebuah rekaman saja yang dapat disimpan pada sebuah alamat penyimpanan. Jumlah pengaksesan dapat direduksi dengan meletakkan lebih dari satu rekaman pada satu alamat penyimpanan.

Kemungkinan tersebut dapat direalisasikan bila digunakan sistem 'buckets' (disebut juga 'blok' atau 'halaman'). Jadi, bucket dapat didefinisikan sebagai unit penyimpanan yang berada di antara rekaman dengan berkas, juga sebuah unit dengan informasi yang dapat diakses dan dipindahkan antar peralatan penyimpanan. Jumlah rekaman yang dapat diletakkan pada satu bucket disebut factor-blocking. Jika factor-blocking meningkat, jumlah akses terhadap penyimpanan akan mengecil karena beberapa rekaman yang berkolisi dapat disimpan dalam satu alamat yang sama.

E. Pembagian Linier

Resolusi kolisi dengan teknik pembagian-linier merupakan varian dari progressive-overflow. Kalau ada progressive-overflow inkremen untuk menuju ke lokasi berikutnya adalah konstan (1=satu), maka pada pembagian-linier digunakan inkremen yang bersifat variable. Tujuan inkremen yang variable adalah mereduksi pengklusteran sekunder yang terjadi pada progressive-overflow sehingga jumlah probe untuk pembacaan kembali juga berkurang. Pengklusteran sekunder akan terjadi pada skema hashing dengan fungsi inkremen yang berupa konstanta atau yang hanya bergantung pada home-address rekaman. Dari semua metoda hashing yang telah dibahas, hanya progressive-overflow yang memiliki kemungkinan terjadinya pengklusteran-skunder. Pada pembagian-linier, inkremen merupakan fungsi dari kunci yang akan disisipkan. Fungsi tersebut dapat juga disebut fungsi hashing lain karena fungsi tersebut digunakan untuk mengkonversi kunci menjadi sebuah inkremen. Untuk alasan tersebut, maka pembagian-linier dapat dikelompokkan sebagai metoda resolusi kolisi dengan hashing-ganda, mengingat pertama dilakukan hash untuk menentukan home-address (menggunakan fungsi H1), dan kedua untuk mendapatkan inkremen (menggunakan fungsi H2). Terdapat dua alternatif bentuk H2, yaitu:

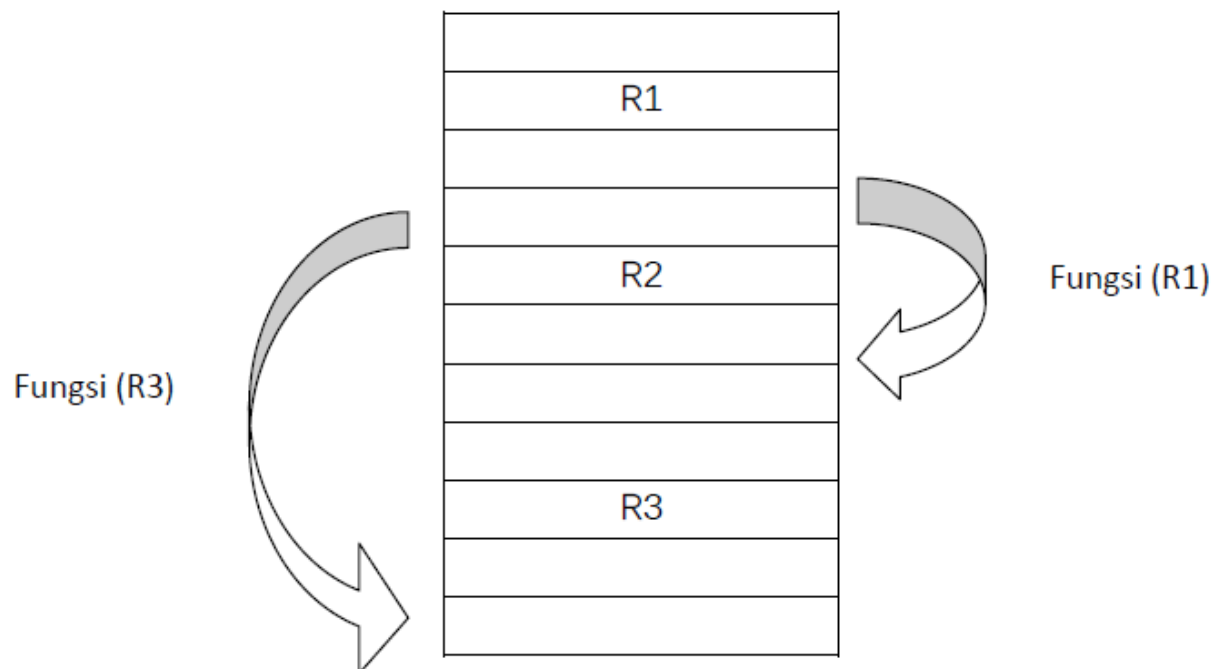
$H2 = \text{Pembagian (kunci/P) modulus P atau}$

$H2' = (\text{kunci modulus (P-2)} + 1$

dengan P merupakan bilangan primer dari ukuran berkas. Kedua fungsi hashing akan menghasilkan variable inkremen dalam cakupan ruang alamat. Pertanyaan yang muncul adalah fungsi hashing yang mana yang sebaiknya digunakan? Misalnya fungsi hashing akan dieksekusi dengan menggunakan mesin, maka H2 memerlukan dua buah operasi pembagian: pertama dalam pembagian, dan kedua dalam mencari modulusnya. Seperti

diketahui, operasi pembagian merupakan operasi yang relatif lambat. Sementara itu $H2'$ hanya memerlukan satu operasi pembagian, yaitu dalam mencari modulusnya. Jadi, untuk praktisnya $H2'$ -lah yang lebih baik untuk digunakan. Akan tetapi, perlu diingat bahwa persamaan seperti $H2'$ sulit diselesaikan oleh manusia. Dua hal yang saling bertentangan tersebut merupakan gambaran mengenai perbedaan kemampuan yang dimiliki baik oleh manusia maupun komputer.

Untuk selanjutnya, dalam contoh-contoh akan digunakan fungsi hash $H2$. Home-address untuk rekaman akan dihitung berdasar sisa hasil pembagian antara kunci dengan ukuran berkas, sedangkan inkremen akan dihitung berdasar hasil pembagian dari operasi yang sama. Tidak seperti rantai coalesced, sinonim pada pembagian-linier umumnya tidak berada pada rantai probe yang sama. Jika $R1$, $R2$, dan $R3$ adalah sinonim pada home-address h seperti pada gambar di bawah ini, $R2$ dan $R3$ biasanya memiliki inkremen yang berbeda mengingat inkremen merupakan fungsi kunci yang akan disisipkan. Karena inkremen yang berbeda, maka pengklusteran sekunder dapat direduksi.



Algoritma untuk metoda resolusi kolisi dengan pembagian-linier adalah sebagai berikut:

- I. Hash kunci yang akan disisipkan ke dalam berkas untuk memperoleh home-address untuk menyimpan rekaman.
- II. Jika home-address kosong, sisipkan rekaman ke dalam lokasi tersebut, jika tidak maka:

- a. Tentukan inkremen dengan menghitung hasil bagi kunci dengan ukuran berkas. Jika hasilnya nol, maka inkremennya = 1 (satu).
- b. Beri harga awal pencacah untuk perhitungan lokasi yang akan dicari dengan 1 (satu).
- c. Selama jumlah lokasi yang dicari lebih kecil dari ukuran berkas,

maka:

- a. Hitung alamat yang akan dicari berikutnya dengan menambahkan inkremen terhadap alamat terakhir dan kemudian cari modulusnya terhadap ukuran berkas.
- b. Jika alamat tersebut tidak ada yang menempati, maka sisipkan rekaman serta akhiri penyisipan dengan sukses.
- c. Jika rekaman yang menempati memiliki kunci yang sama dengan yang akan disisipkan, akhiri proses dengan pesan ,rekaman dobel`.
- d. Tambahkan 1 pada pencacah pencarian lokasi.
- e. Akhiri proses dengan pesan ,Berkas penuh`

Algoritma tersebut mensyaratkan ukuran berkas berupa bilangan prima untuk menghindari terjadinya siklus pencarian terhadap rekaman berulang-ulang dan kemudian memberikan indikasi bahwa rekaman sudah penuh padahal masih terdapat ruang yang belum ditempati.