ADT Map/ Associative Array

IF1210 – Algoritma dan Pemrograman 1
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

Map – Definisi

Map adalah kumpulan pasangan/*binding* ⟨*key*, *value*⟩, dengan nilai *key* bersifat unik di dalam kumpulan tsb.

Dikenal juga sebagai *associative array*, *symbol table*, atau *dictionary*.

Operasi terhadap Map:

- Penambahan sebuah pasangan baru
- Penghapusan suatu pasangan
- Modifikasi nilai *value* dari suatu pasangan
- Pencarian nilai *value* dari suatu *key* tertentu.

Map – Definisi Fungsional

Jika diberikan *M* adalah **Map** dengan elemen pasangan ⟨*K*,*V*⟩

```
CreateMap: → M                { Membuat sebuah Map kosong }

  isEmpty: M → boolean        { Tes terhadap M: true jika M kosong,
                                false jika M tidak kosong }

      set: K × V × M → M      { Menambahkankan pasangan (K,V) ke M jika
                                belum ada elemen dengan nilai key=K pada M,
                                atau mengubah nilai value dari pasangan
                                dengan nilai key=K menjadi V }

    unset: K × M → M          { Menghapus pasangan dengan nilai key=K
                                dari M }

     find: K × M → V          { Mengembalikan value dari pasangan dengan
                                nilai key=K }
```

Axiomatic Semantics (fungsional)

1) new() returns a map
2) find(k, set(k, v, M)) = v
3) find(k, set(j, v, M)) = find(k, M) if k ≠ j
4) unset(k, new()) = new()
5) unset(k, set(k, v, M)) = unset(k, M)
6) unset(k, set(j, v, M)) = set(j, v, unset(k, M)) if k ≠ j

dengan k dan j adalah key, v adalah value, dan M adalah map.

Map – Implementasi

Jika jumlah elemen sedikit, dapat digunakan *association list* dalam bentuk array dengan elemen pasangan ⟨*key, value*⟩.

Jika nilai *key* terbatas pada selang nilai integer tertentu, dapat digunakan *direct addressing* terhadap tabel: pasangan ⟨*key, value*⟩ disimpan dengan memberi nilai *value* pada indeks ke-*key*.

Jika nilai *key* memiliki rentang yang besar namun data hanya sedikit, *direct addressing* menjadi boros memori.

- Perlu dilakukan pemetaan *key* ke *address*/indeks.
- Implementasi yang paling umum digunakan adalah menggunakan *hash table*.

ADT Map (bandingkan dengan Set!)

```
KAMUS UMUM
constant  CAPACITY: integer = ... { Banyaknya elemen maksimum }
constant VAL_UNDEF: ElType = ...

type KeyType: ...
type ElType: ...
type MapEntry: < key: KeyType, value: ElType > { elemen map }

type Map:
    < buffer: array [0..CAPACITY-1] of MapEntry, { array penyimpan elemen map }
      length: integer >                          { jumlah elemen map }
```