



Institut Teknologi
Telkom
Purwokerto

Teknik Informatika - Fakultas Informatika

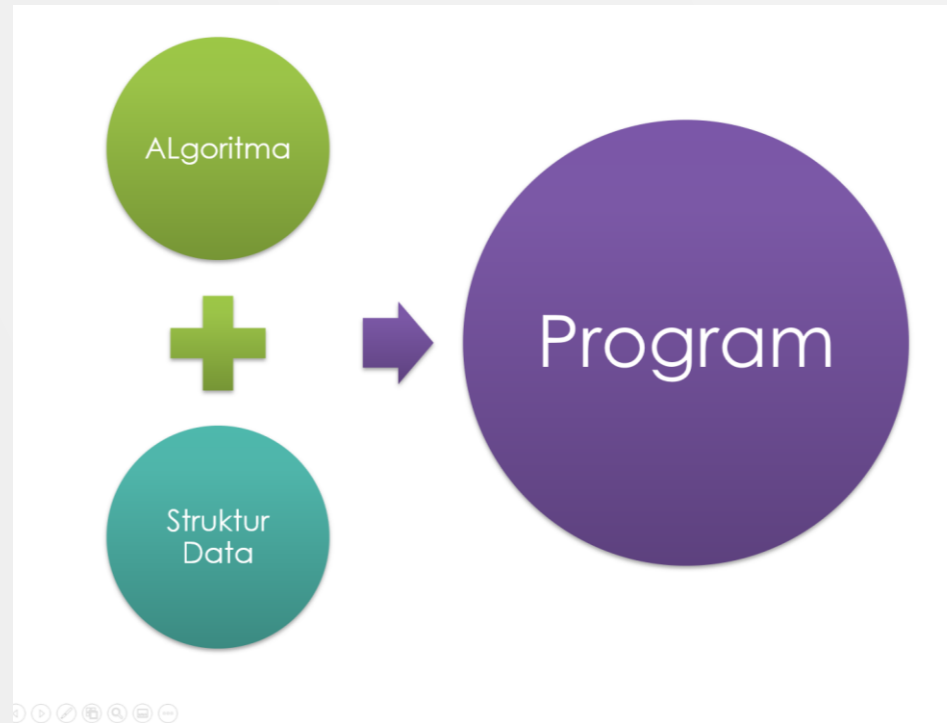
Pertemuan 2 – Algoritma dan Struktur Data

Author: **Condro Kartiko [CKO]**

Co-Author: **Wahyu Andi Saputra [WAA]**

Data dan Struktur data

- Struktur data menjadi dasar dalam langkah awal perancangan program



Data dan Struktur data

Pemakaian **struktur data** yang tepat di dalam proses pemrograman akan menghasilkan :

- Algoritma yang **lebih jelas dan tepat**, sehingga menjadikan program secara keseluruhan **lebih efisien dan sederhana**.

Data dan Struktur data

- Membuat program **lebih ringkas, lebih bersih**, lebih elegan, lebih mudah dan lebih mampu berkinerja tinggi (karena efisien dalam penggunaan memori dan waktu).
- Program berjalan membutuhkan waktu **beberapa detik**, di mana struktur yang lain mungkin akan membutuhkan ribuan detik.

Data

- representasi dari fakta dunia nyata
- fakta atau keterangan tentang kenyataan yang disimpan, **direkam** atau **direpresentasikan** dalam bentuk tulisan, suara, gambar, sinyal atau simbol

Perbedaan Tipe Data, Obyek Data & Struktur Data

- **Tipe Data Standar**
- Tipe data standar merupakan tipe data yang tersedia pada kebanyakan komputer sebagai built-in features, yaitu:
 - Tunggal
 - Integer
 - Real
 - Boolean
 - Char
 - Majemuk
 - String

Perbedaan Tipe Data, Obyek Data & Struktur Data

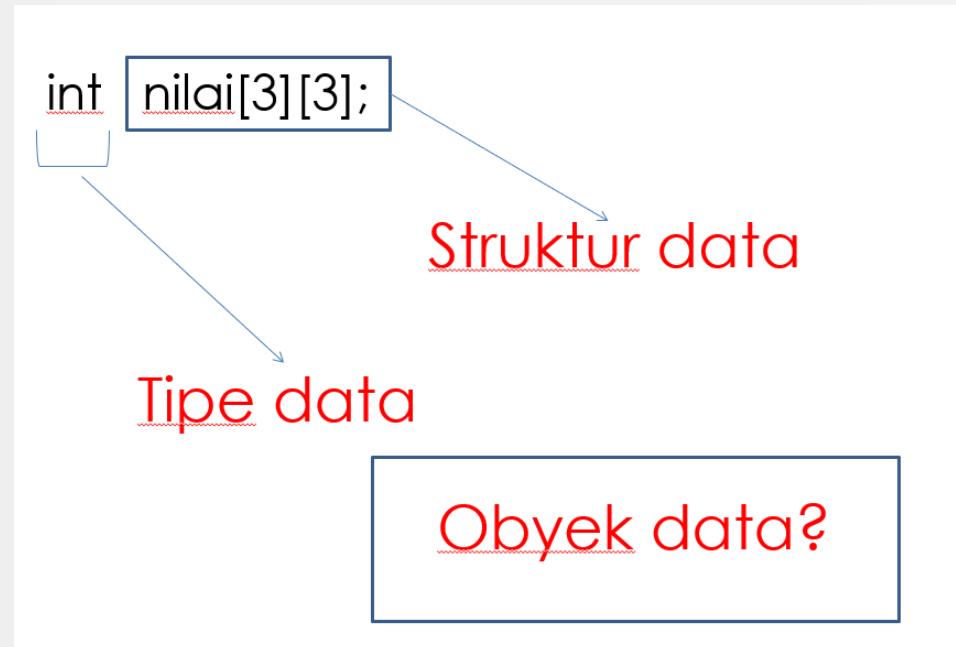
Obyek Data

adalah kumpulan elemen yang mungkin untuk suatu tipe data tertentu.

Mis: integer mengacu pada obyek data -32768 s/d 32767, byte 0 s/d 255, string adalah kumpulan karakter maks 255 huruf

Perbedaan Tipe Data, Obyek Data & Struktur Data

Struktur Data

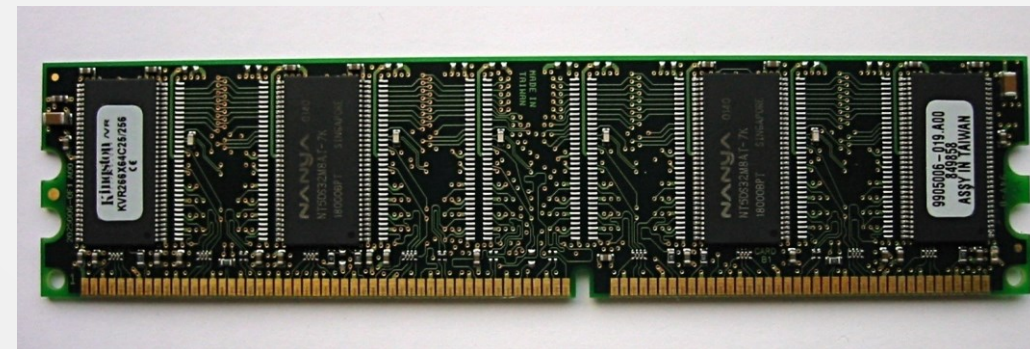


Aktivitas Struktur Data

- Di dalam struktur data kita berhubungan dengan 2 aktivitas:
- Mendeskripsikan kumpulan obyek data yang sesuai dengan tipe data yang ada
- Menunjukkan mekanisme kerja operasi-operasinya
- Contoh: integer (-32768 s/d 32767) dan jenis operasi yang diperbolehkan adalah +, -, *, /, mod, ceil, floor, <, >, != dsb.
- Struktur data = obyek data + [operasi manipulasi data]

Hubungan SD dan memory (RAM)

- Setiap **tipe data dan struktur data** yang dipilih, membutuhkan ruang penyimpanan di memory
- Programmer harus tahu, **seberapa besar alokasi memory** yang digunakan untuk menampung data
- Pemborosan ruang di memory akan menyebabkan **performa program berjalan lambat**



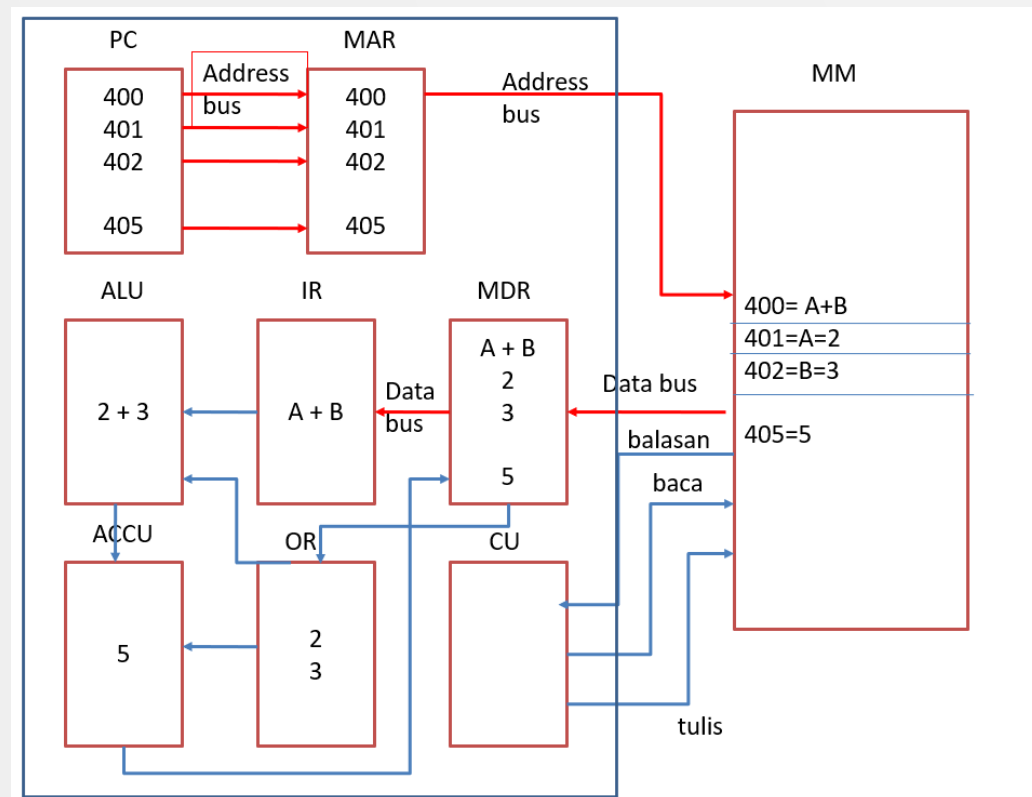
Hubungan SD dan memory (RAM)

RAM

- Bersifat sementara = volatile
- Berisi program dan data yang sedang diproses oleh prosesor
- akses terhadap lokasi-lokasi di dalamnya dapat dilakukan secara acak (random), bukan secara berurutan (sekuensial)
- RAM=main memory

Hubungan SD dan memory (RAM)

Perhatikan ilustrasi tentang pemrosesan instruksi berikut !



Tabel type data dan range

Type	Typical Bit Width	Typical Range
char	1byte	-127 to 127 or 0 to 255
unsigned char	1byte	0 to 255
signed char	1byte	-127 to 127
int	4bytes	-2147483648 to 2147483647
unsigned int	4bytes	0 to 4294967295
signed int	4bytes	-2147483648 to 2147483647
short int	2bytes	-32768 to 32767
unsigned short int	Range	0 to 65,535
signed short int	Range	-32768 to 32767
long int	4bytes	-2,147,483,648 to 2,147,483,647
signed long int	4bytes	same as long int
unsigned long int	4bytes	0 to 4,294,967,295
float	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	8bytes	+/- 1.7e +/- 308 (~15 digits)
long double	8bytes	+/- 1.7e +/- 308 (~15 digits)
wchar_t	2 or 4 bytes	1 wide character

Size of (untuk mengetahui size dari suatu variable)

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << "Size of char : " << sizeof(char) << endl;
7      cout << "Size of int : " << sizeof(int) << endl;
8      cout << "Size of short int : " << sizeof(short int) << endl;
9      cout << "Size of long int : " << sizeof(long int) << endl;
10     cout << "Size of float : " << sizeof(float) << endl;
11     cout << "Size of double : " << sizeof(double) << endl;
12     cout << "Size of wchar_t : " << sizeof(wchar_t) << endl;
13     return 0;
14 }
15
```

Hubungan SD dan Algoritma

- Dengan **pemilihan struktur data** yang baik, maka problem yang kompleks dapat diselesaikan sehingga algoritma dapat digunakan secara **efisien**, operasi-operasi penting dapat dieksekusi dengan sumber daya yang lebih kecil, memori lebih kecil, dan waktu eksekusi yang lebih cepat.

Ciri algoritma yang baik

- Ciri algoritma yang baik menurut **Donald E.Knuth**:
 - Input: ada minimal 0 input atau lebih
 - Output: ada minimal 1 output atau lebih
 - Definite: ada kejelasan apa yang dilakukan
 - Effective: langkah yang dikerjakan harus efektif
 - Terminate: langkah harus dapat berhenti (stop) secara jelas



Institut Teknologi
Telkom
Purwokerto

TERIMA KASIH

Angkat tangan apabila ada pertanyaan