



MODUL I Struktur Data

Judul	DASAR DASAR STRUKTUR DATA	
Penyusun	Distribusi	Perkuliahan
Nixon Erzed	Teknik Informatika Universitas Esa Unggul	Pertemuan – I online

Tujuan :

1. Mahasiswa memahami pengertian dan posisi struktur data dalam pemrograman
2. Mahasiswa mengenal dan memahami tipe data dan urgensi representasi data dalam pemrograman

Materi:

1. Pemrograman Komputer
2. Input output
3. Logika proses dan algoritma
4. Data dan deklarasi struktur data
5. Penyelesaian masalah dengan computer
6. Konstanta dan variabel

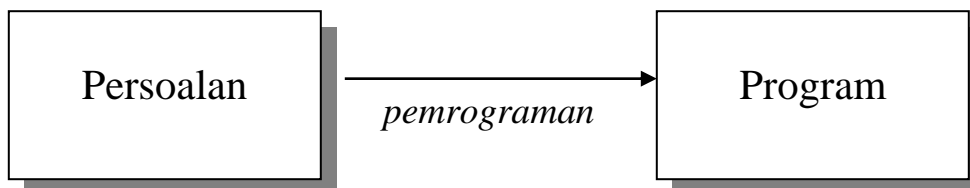
Referensi :

1. Algoritma dan Pemrograman dalam Bahasa Pascal, C, dan C++, Rinaldi Munir 2016
2. Struktur Data, Bambang Hariyanto, Infomatika, Bandung, 2003
3. Algoritma dan Struktur Data 1 & 2 , Moh. Syukani, Mitra Wacana Media, 2012
4. Fundamental of Data Structure, Ellis Horowitz, Pitman International Text, 1978

DASAR-DASAR STRUKTUR DATA

I. PEMROGRAMAN KOMPUTER

Komputer pada dasarnya adalah mesin yang tidak bisa apa-apa. Harus diberikan serangkaian instruksi kepada komputer agar dapat memecahkan suatu masalah. Langkah-langkah yang dilakukan untuk memberikan instruksi kepada komputer untuk memecahkan masalah dinamakan *pemrograman komputer*.



Hal-hal yang berhubungan dengan pemrograman komputer :

1. Bahasa pemrograman
2. Source program (program sumber)
3. Interpreter atau compiler
4. File program (execute file)

Bahasa Pemrograman

Aturan penulisan instruksi komputer dalam bahasa yang dimengerti oleh manusia.

Kelompok bahasa pemrograman : Bahasa Tingkat Rendah, Bahasa Generasi III, dan Bahasa Generasi IV

Umumnya bahasa pemrograman menggunakan kata atau singkatan kata dalam bahasa Inggris.

Bahasa Tingkat Rendah atau assembly adalah spesifik mikroprosesor, yang menggunakan singkatan-singkatan dalam bahasa Inggris, tapi dengan sintaks yang sangat kaku karena merupakan pemetaan satu-satu dari instruksi bahasa mesin.

Bahasa Generasi III (3GL / bahasa tingkat tinggi) menggunakan sintaks yang lebih dekat dengan bahasa manusia, sehingga lebih mudah digunakan.

Termasuk 3GL : Basic, Pascal, C, C++, Cobol, dan sebagainya.

Bahasa Generasi IV (4GL) adalah bahasa yang memiliki kemudahan untuk mengembangkan aplikasi basis data. Salah satu contohnya adalah SQL (Structured Query Language) yang banyak diadopsi oleh Sistem Manajemen Basis Data dan *aplikasi buildemya*.

Program Sumber (*Source Program*)

Program sumber adalah kumpulan instruksi yang ditulis menggunakan dan mengikuti aturan bahasa pemrograman untuk menyelesaikan suatu persoalan tertentu.

Assembler, Interpreter dan Compiler

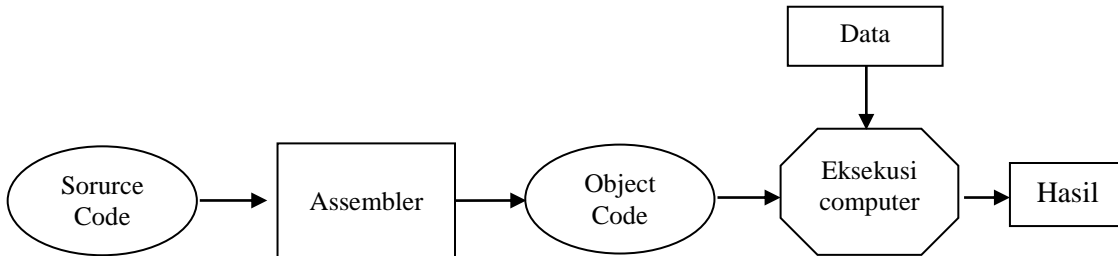
Assembler, interpreter dan compiler adalah perangkat lunak untuk menterjemahkan program sumber ke bahasa mesin.

Assembler adalah perangkat lunak yang mengkonversikan perintah-perintah assembly ke bahasa mesin.

Interpreter menterjemahkan program baris per baris ke bahasa mesin sebelum dieksekusi, artinya interpretasi dilakukan ketika program akan dieksekusi dan hasil interpretasi tidak disimpan secara permanen dalam memory sekunder.

Compiler menterjemahkan semua perintah program sumber kedalam bahasa mesin, hasil kompilasi disimpan secara permanen dalam memory sekunder.

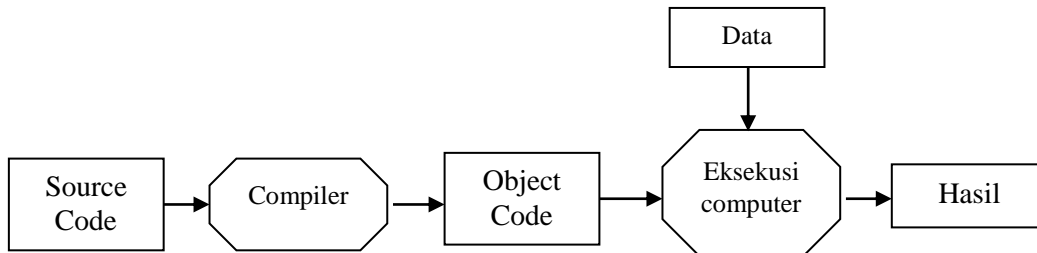
a. Struktur Sistem Assembler :



Source code adalah bahasa assembly, object code adalah bahasa mesin.

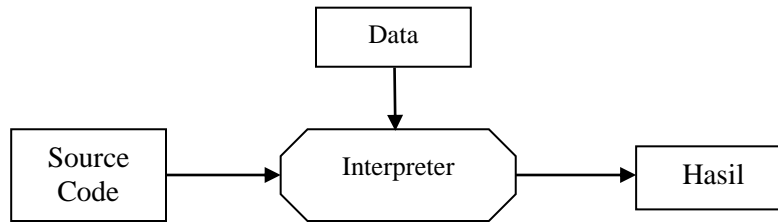
Contohnya : Turbo Assembler dan Macro Assembler

b. Struktur Sistem Kompilator :



Source code adalah bahasa tingkat tinggi, object kode adalah bahasa mesin atau bahasa assembler. Source code dan data diproses pada saat yang berbeda

c. Struktur Sistem Interpreter :



Interpreter tidak membangkitkan object code, hasil translasi hanya dalam bentuk internal.

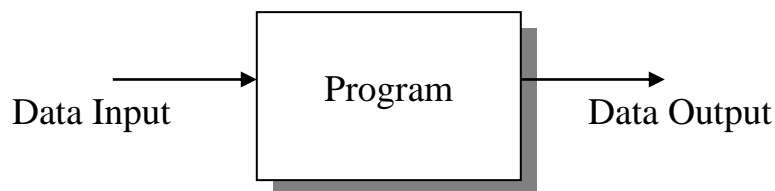
Execute File/Program

Execute file atau *execute program* adalah file hasil kompilasi yang siap untuk dijalankan oleh mesin (komputer).

II. INPUT OUTPUT

Program komputer adalah sebuah sistem. Sesuai dengan konsep sistem program akan bekerja jika mendapat masukan tertentu. Hasil kerjanya diidentifikasi berdasarkan output yang dihasilkannya.

Input dan output dalam program komputer berbentuk data.



Dalam sistem komputer, data input akan disimpan dalam suatu variabel yang merepresentasikan data tersebut dalam proses-proses yang dilakukan program.

Data input dapat diterima melalui masukan langsung dengan keyboard atau perangkat masukan lainnya, atau masukan yang berasal dari file data yang tersimpan dalam memory sekunder.

Hasil pengolahan oleh program dapat dikeluarkan langsung ke perangkat keluaran atau disimpan/ditulis ke file data untuk disimpan dalam memory sekunder.

Pengorganisasian data yang merepresentasikan persoalan dalam program dikenal sebagai struktur data.

III. LOGIKA PROSES DAN ALGORITMA

Hasil analisis terhadap persoalan akan memberikan **metoda-tepat** bagaimana masalah akan diselesaikan dengan sistem komputer, yang merupakan logika proses/penyelesaian masalah.

Logika proses tersebut akan dituangkan dalam sekumpulan langkah-langkah terbatas. Setiap langkah mungkin memerlukan satu operasi atau lebih.

Ciri-ciri algoritma :

1. Input
2. Output
3. Definite
4. Effective
5. Terminate

Input : Terdapat nol atau lebih masukan yang diberikan secara eksternal

Output : Sedikitnya terdapat satu keluaran yang harus dihasilkan

Definite : Harus secara sempurna menyatakan apa yang akan dikerjakan

Contoh :

- a. Hitung $5/0$
- b. Tambahkan 6 atau 7 ke x

Contoh instruksi tersebut tidak diijinkan karena :

Kasus a : tidak jelas hasil operasinya.

Kasus b : tidak jelas yang harus dilakukan.

Effective : Setiap instruksi harus dapat dilakukan secara manual menggunakan pensil dan kertas dalam jumlah waktu yang berhingga.

Terminate : Harus berhenti setelah sejumlah terbatas operasi.

IV. DATA DAN DEKLARASI STRUKTUR DATA

Penganalisan persoalan untuk dapat diselesaikan dengan komputer diwujudkan melalui penelitian terhadap data-data yang mewakili persoalan tersebut. Pemrosesan oleh komputer hanya dapat dilakukan terhadap data kuantitatif. Jika dalam analisis persoalan terdapat data kualitatif, maka data tersebut mesti dikuantifisir, artinya harus didapatkan padanan angka eksak dari setiap nilai kualitatif.

Contoh :

Persoalan posisi untung rugi warung makan. Analisis terhadap persoalan diketahui bahwa untung rugi usaha dapat dihitung berdasarkan data nilai penjualan, biaya bahan baku dan biaya operasional.

Setiap data yang akan diolah oleh program, memerlukan variabel sebagai wadah baik sebagai penampung data awal maupun untuk menyimpan data hasil proses. Untuk memberikan identitas dan batasan dalam algoritma/program, setiap variabel yang terlibat dalam proses harus didefinisikan/dideklarasikan diawal algoritma atau program.

Deklarasi data meliputi :

- penamaan variabel
- penetapan type variabel
- panjang/ukuran data

Untuk contoh diatas misalnya dideklarasikan sebagai berikut :

Deklarasi

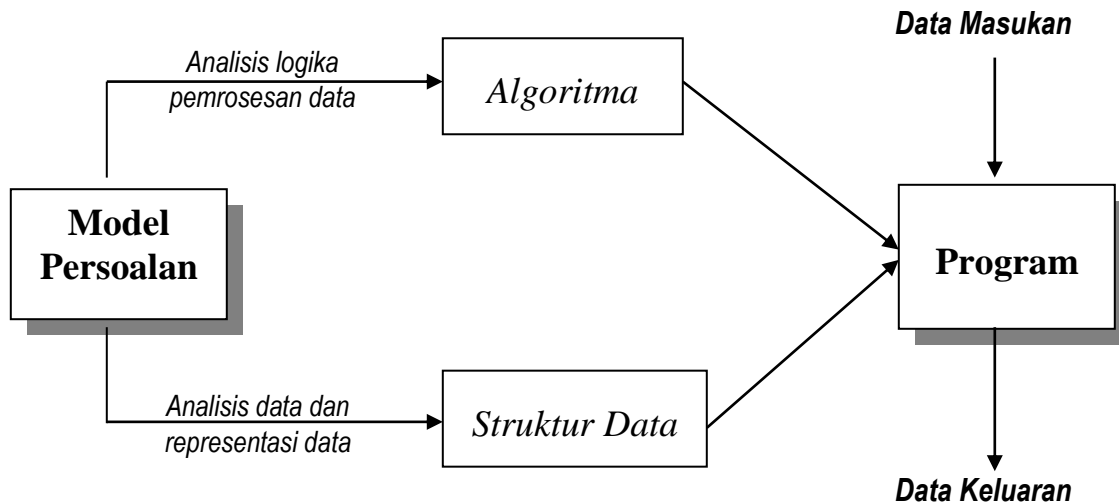
<i>N_jual</i>	<i>variabel larik bertipe real dengan panjang 100 data</i>
<i>B_bahan</i>	<i>variabel larik bertipe real dengan panjang 100 data</i>
<i>B_Opers</i>	<i>variabel larik bertipe real dengan panjang 100 data</i>

Aturan pendeklarasian struktur data tergantung pada aturan tata bahasa pemrograman.

Contoh diatas adalah model untuk bahasa algoritma.

V. PENYELESAIAN MASALAH DENGAN KOMPUTER

Model penyelesaian persoalan dengan komputer :



Pada penyelesaian persoalan yang besar dan kompleks, akan sangat membantu jika persoalan dipecah menjadi sub-persoalan (*clustering dan decoupling*). Dalam melakukan analisis logika pemrosesan data, clustering/decoupling akan menghasilkan modulisasi program, sedangkan pada analisis data dan representasi data akan menghasilkan sub-sub struktur data. Untuk penyelesaian persoalan komputasi bisnis, sub-sub struktur data tersebut tercermin dalam rancangan basis data dan sistem file.

Beberapa hal yang harus diperhatikan dalam penyelesaian persoalan dengan komputer:

1. Pemahaman secara menyeluruh keterhubungan elemen-elemen data yang relevan terhadap solusi persoalan
2. Operasi-operasi yang akan dilakukan terhadap elemen data
3. Perancangan metoda representasi elemen-elemen data di memori sehingga memenuhi kriteria berikut :
 - a. Memenuhi keterhubungan logika antara elemen-elemen data
 - b. Operasi-operasi terhadap elemen data dapat dilakukan dengan mudah dan efisien
4. Memilih bahasa pemrograman yang paling sesuai untuk menterjemahkan solusi persoalan menjadi program

TYPE DATA & REPRESENTASI DATA

I. KONSTANTA DAN VARIABEL

Konstanta :

Konstanta (literal) adalah suatu nilai yang tetap berada di dalam program. Jika dalam membuat suatu program sering menggunakan bilangan numerik atau suatu kalimat string sama berkali-kali, akan lebih baik jika bilangan atau string tersebut dideklarasikan sebagai *konstanta*.

Manfaatnya : memudahkan dalam pemahaman dan perbaikan program. Misalnya untuk sebuah bilangan yang kompleks, akan lebih baik jika diberi sebuah nama yang mudah. Jika ingin dilakukan perubahan terhadap konstanta, maka hanya diperlukan penggantian satu kali saja.

Contoh (dalam sintaks Pascal) :

```
Const Pi = 3.1415926536
```

Didalam blok program jika akan digunakan, cukup ditulis *Pi*.
Jika ingin diganti nilai *Pi* menjadi 3.14 maka cukup diganti pada deklarasi konstanta.

Deklarasi Konstanta :

Sesuai aturan penulisan bahasa pemrograman. Pada Pascal, dideklarasikan pada awal program sebelum blok (*begin-end*)

Jenis jenis konstanta :

- a. Konstanta biasa :
mendeklarasikan suatu nilai numerik, character, atau string
- b. Konstanta bertipe :
selain nilai, pada konstanta juga dapat didefinisikan type tertentu

Contoh (dalam sintaks Pascal) :

<i>Pi = 3.14;</i>	(konstanta biasa)
<i>Indeks = 'A';</i>	(konstanta biasa)
<i>Nama = 'Esa Unggul';</i>	(konstanta biasa)
<i>JumlahMahasiswa : integer = 15000;</i>	(konstanta bertipe)
<i>CekNilai : boolean=true;</i>	(konstanta bertipe)
<i>JdlHal : string[25]='Pendapatan Tahunan';</i>	(konstanta bertipe)

VARIABEL

Variabel adalah suatu lokasi dimemori yang disiapkan oleh programmer dan diberi nama khas untuk menampung suatu nilai dan atau mengambil kembali nilai tersebut.

Pendeklarasian variabel merupakan bagian langkah pembangunan struktur data. Penamaan variabel yang baik merepresentasikan data yang diwadahnya. Setiap bahasa pemrograman memiliki aturan penamaan.

Contoh :

JmlMahasiswa : untuk merepresentasikan data jumlah mahasiswa
Luas : untuk merepresentasikan data luas
NilaiUTS [i] : untuk merepresentasikan data nilai mahasiswa

Deklarasi variabel mencakup :
pemberian nama identifier
pedefinisian type
pendefinisian jangkauan (khususnya untuk variabel larik)

Contoh :

Pascal

JmlPesertaMK : integer;
NilaiUTS : array [1.. 100] of real;

Dimana :

JmlPesertaMK dan *NilaiUTS* adalah nama variabel
Integer dan *array..of real* adalah type variabel
[1 .. 100] adalah jangkauan

C

int jmlpeserta
float nilaiUTS [20]

Dimana :

jmlPeserta dan *nilaiUTS* adalah nama variabel
Int dan *float* adalah type variabel
[20] adalah jangkauan

II. STRUKTUR DATA

Type Data

Type data adalah jenis data yang ditangani oleh bahasa pemrograman. Tiap bahasa pemrograman memiliki kumpulan type data *built-in*, sehingga :

- Memungkinkan deklarasi variabel bertipe tersebut.
- Menyediakan kumpulan operasi untuk manipulasi variabel bertipe tersebut.

Beberapa type data dasar sudah disediakan oleh pemroses, sehingga secara mudah bahasa pemrograman mengadopsinya, yaitu : *boolean, integer dan real*. Sedangkan type lain merupakan type bentukan yang memerlukan rutin khusus untuk mengimplementasikannya dalam bahasa pemrograman, misalnya : *string, character dan lain-lain*.

Contoh

Bahasa Pascal : *boolean, char, integer, real, string*

Bahasa C : *int, short int, long int, float, double, character, string*

Bahasa Fortran : *character, integer, real, logical, double precision*

Bahasa Snobol : *string, character*

Bahasa Lisp : *list (ekspresi S)*

Ukuran dan jangkauan data dari suatu tipe data tergantung dari [kompilator](#) dan [komputer](#) yang digunakan. Berikut adalah jangkauan dan ukuran tipe integer pada beberapa bahasa pemrograman.

Bahasa Pemrograman C

Tipe integer standar yang digunakan dalam bahasa C adalah tipe *int*. Ukuran dan jangkauan data dari tipe integer adalah sebagai berikut :

- **char** : tipe data ini digunakan untuk menyimpan karakter dalam kode [ASCII](#), tapi dapat juga digunakan untuk menyimpan integer dari 0 sampai 255
- **short int** : ukuran 2 byte, jangkauan -32,768 sampai 32,767
- **int** : ukuran 4 byte, jangkauan -2,147,483,648 hingga 2,147,483,647

Tipe-tipe data di atas dapat menyimpan integer negatif dan positif.

Untuk menyimpan bilangan positif dan nol saja, dapat digunakan kata kunci *unsigned* sebelum tipe data. Sebagai contoh:

- **unsigned short int**, ukuran 2 byte, jangkauan 0 sampai 65,535
- **unsigned int**, ukuran 4 byte, jangkauan 0 sampai 4,294,967,295

Bahasa Pemrograman Pascal

Dalam bahasa Pascal, integer mampu menampung 16-bit. Walaupun memiliki ukuran 2 [byte](#) (16 [bit](#)) tetapi karena integer adalah type data signed maka hanya mampu di-assign nilai antara -215 hingga 215-1 yaitu -32768 sampai 32767. Ini disebabkan karena 1 bit digunakan sebagai penanda positif/negatif.

Meskipun memiliki istilah yang sama, tetapi tipe data integer pada bahasa pemrograman [Visual Basic.NET](#) dan [Borland Delphi](#) memiliki ukuran 4 [byte](#) atau 32 [bit](#) signed sehingga dapat di-assign nilai antara -2,147,483,648 hingga 2,147,483,647.

Selain tipe integer, bahasa Pascal juga memiliki beberapa tipe lain:

- [byte](#), ukuran 1 byte, jangkauan dari 0 sampai 255
- [smallint](#), ukuran 1 byte, jangkauan dari -128 sampai 127
- [word](#), ukuran 2 byte, jangkauan dari 0 sampai 65,535

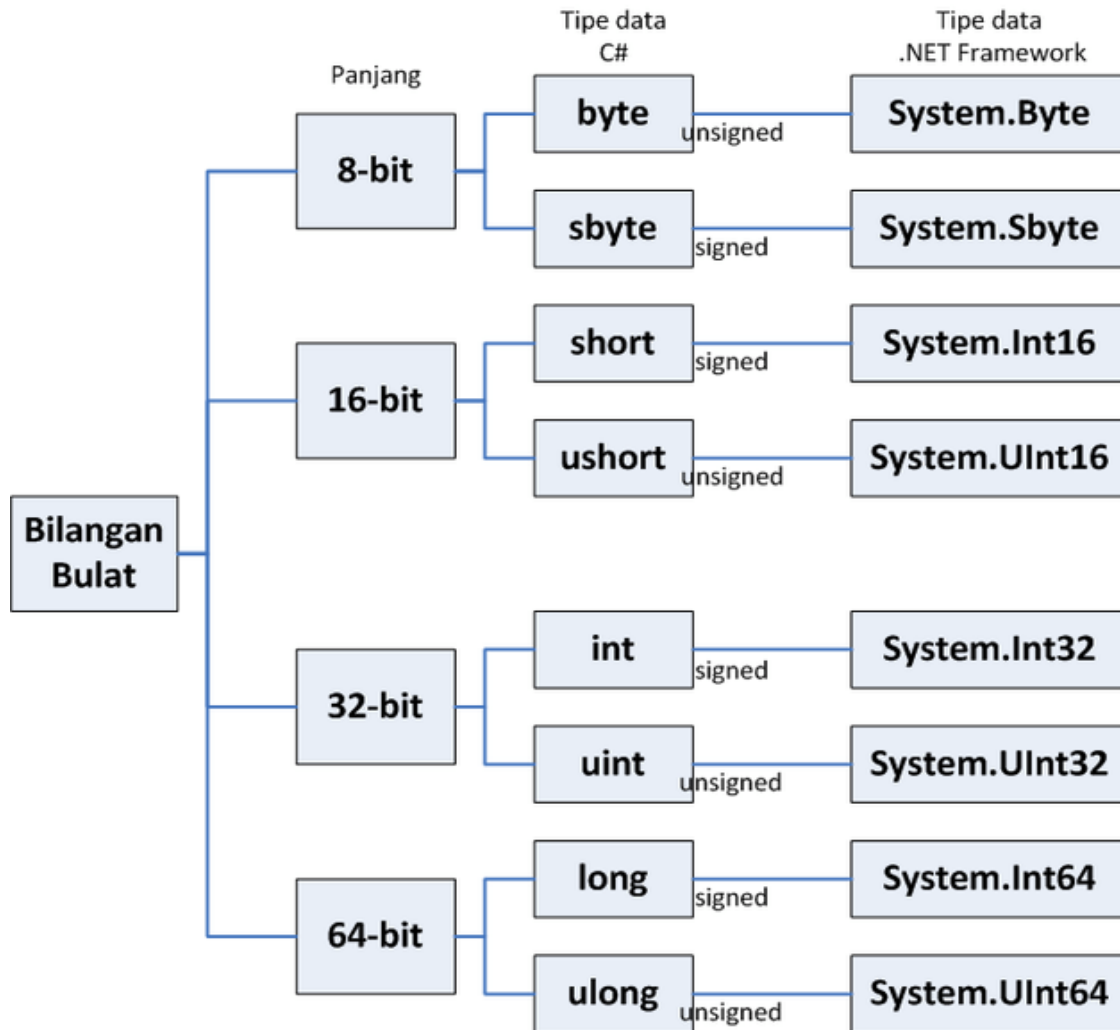
Pada [kompilator](#) Pascal yang lebih baru, juga dikenal tipe-tipe data yang lebih besar seperti:

- [longint](#), ukuran 4 byte, jangkauan dari -2,147,483,648 sampai 2,147,483,647
- [cardinal](#), ukuran 4 byte, jangkauan dari 0 sampai 4,294,967,295

Perbandingan nama tipe data bilangan bulat dalam bahasa pemrograman C# dan Microsoft .NET Framework, sebagai berikut :

- [byte](#): bilangan bulat tak bertanda (unsigned integer) 8-bit. Ekuivalen dengan tipe data [System.Byte](#) dalam [Microsoft .NET Framework](#).
- [sbyte](#): bilangan bulat bertanda (signed integer) [8-bit](#). Ekuivalen dengan tipe data System.Sbyte dalam Microsoft .NET Framework.
- [short](#): bilangan bulat bertanda 16-bit. Ekuivalen dengan tipe data System.Int16 dalam Microsoft .NET Framework.
- [ushort](#): bilangan bulat tak bertanda (unsigned integer) [16-bit](#). Ekuivalen dengan tipe data System.UInt16 dalam Microsoft .NET Framework.
- [int](#): bilangan bulat bertanda (signed integer) [32-bit](#). Ekuivalen dengan tipe data System.Int32 dalam Microsoft .NET Framework.
- [uint](#): bilangan bulat tak bertanda (unsigned integer) 32-bit. Ekuivalen dengan tipe data System.UInt32 dalam Microsoft .NET Framework.

- **long**: bilangan bulat bertanda (signed integer) 64-bit. Ekuivalen dengan tipe data System.Int64 dalam Microsoft .NET Framework.
- **ulong**: bilangan bulat tak bertanda (unsigned integer) 64-bit. Ekuivalen dengan tipe data System.UInt64 dalam Microsoft .NET Framework.



Objek Data

Objek data adalah mengacu kumpulan elemen D (Domain)

Misalnya :

Objek data *integer* mengacu domain berikut :

$$D = \{ 0, \pm 1, \pm 2, \pm 3, \dots \}$$

Objek data string karakter dengan panjang kurang dari 3 mengacu kepada domain berikut:

$$D = \{ ' ', 'A', 'B', \dots, 'Z', 'AA', \dots, 'ZZ', 'AAA', 'BBB', \dots \}$$

D dapat berjumlah berhingga atau tak berhingga. Jika D sangat besar maka diperlukan cara khusus untuk merepresentasikan elemen-elemen Domain D di komputer.

Struktur Data

Untuk membuat menjadi struktur data, kita harus melakukan aktifitas terhadap objek data, yaitu :

- Mendeskripsikan kumpulan operasi sah yang diterapkan terhadap elemen-elemen objek data
- Menunjukkan mekanisme kerja operasi-operasi

Secara terstruktur : struktur data adalah objek data ditambah himpunan operasi untuk memanipulasi objek.

$$\text{Struktur data} = \text{objek data} + \{\text{operasi manipulasi}\}$$

Operasi manipulasi harus dapat menjamin terpenuhinya spesifikasi relasi antar elemen-elemen data atau menjamin invarian.

Contoh :

1. Struktur data integer:

$$\begin{aligned} \text{Integer} : D & \{ 0, \pm 1, \pm 2, \pm 3, \dots \} \\ \text{Operasi} & = \{ +, -, *, \text{mod}, \text{div} \} \end{aligned}$$

Operasi manipulasi +, -, * menjamin terpenuhinya spesifikasi relasi antar elemen-elemen data integer. Sementara itu operasi bagi (/) tidak dapat menjamin spesifikasi relasi dan invarian, karena hasil operasi bagi (/) antara 2 integer tidak lagi menghasilkan type real, bukan integer. Operasi yang memenuhi syarat adalah DIV mengambil nilai bulat pembagian dan MOD mengambil sisa pembagian.

2. Struktur Data Waktu : $t = (j : m : d)$

$$\begin{aligned} \text{Waktu} : \text{Domain } D & \{ j, m, d \mid 0 \leq j \leq 23, 0 \leq m \leq 59, 0 \leq d \leq 59, j, m, d \in \mathbb{Z} \} \\ \text{Operasi} & \{ \text{totaldetik} = f(t), \text{lama} = f(t_1, t_2) \} \end{aligned}$$