



UNIVERSITAS
ATMA JAYA YOGYAKARTA
serviens in lumine veritatis



Dasar Pemrograman (INFT06204)

Minggu 4

Tipe, Nama dan Nilai

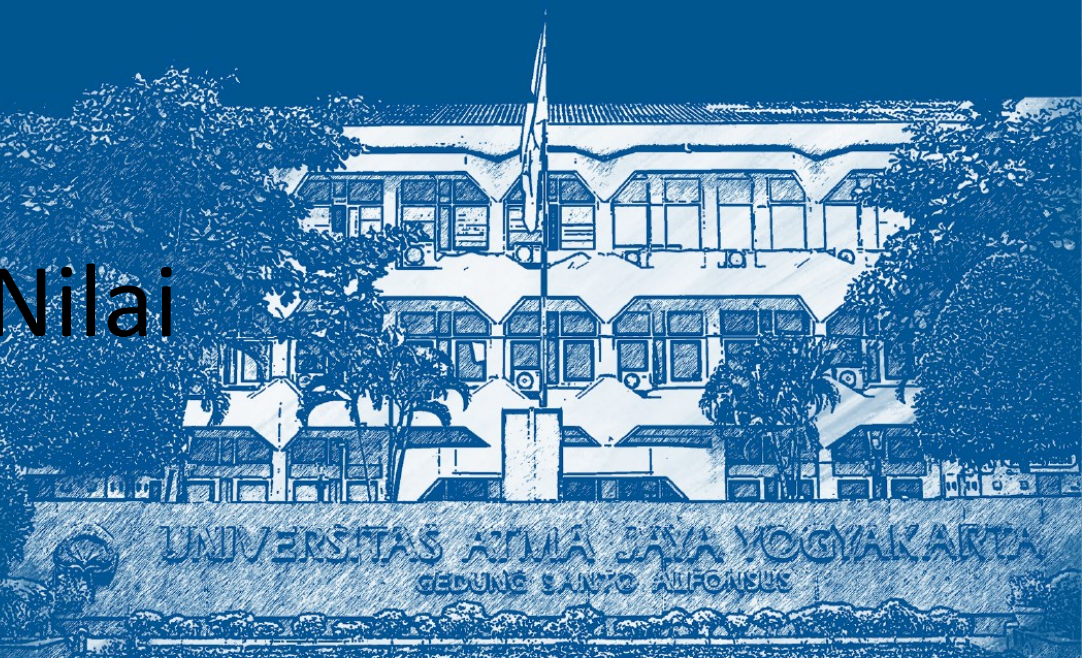




FIGURE 2.1 C Language Elements in Miles-to-Kilometers Conversion Program

```
/*
 * Converts distances from miles to kilometers.
 */

#include <stdio.h>          /* printf, scanf definitions */
#define KMS_PER_MILE 1.609 /* conversion constant */

int main(void)
{
    double miles, /* distance in miles
    kms; /* equivalent distance in kilometers */

    /* Get the distance in miles. */
    printf("Enter the distance in miles> ");
    scanf("%lf", &miles);

    /* Convert the distance to kilometers. */
    kms = KMS_PER_MILE * miles;

    /* Display the distance in kilometers. */
    printf("That equals %f kilometers.\n", kms);

    return (0);
}
```

Diagram labels and arrows:

- preprocessor directive** points to `#include <stdio.h>` and `#define KMS_PER_MILE 1.609`.
- constant** points to `1.609` in the `#define` line.
- reserved word** points to `int` and `main(void)`.
- variable** points to `double miles` and `kms`.
- comment** points to `/* printf, scanf definitions */` and `/* conversion constant */`.
- comment** points to `/* Get the distance in miles. */`.
- standard identifier** points to `printf` and `scanf`.
- special symbol** points to `*` in `KMS_PER_MILE * miles` and `\n` in the final `printf`.
- punctuation** points to `(0)` in `return (0);`.
- reserved word** points to `return`.



Tipe Data

- Program komputer bekerja dengan memanipulasi obyek (data) di dalam memori
- Obyek yang akan diprogram bermacam-macam jenis/tipenya
- Tipe data menyatakan pola penyajian data dalam komputer





Tipe Data

Tipe data terdiri dari :

- Tipe dasar (*built-in*)
 - Tipe yang dapat langsung dipakai (disediakan oleh bahasa pemrograman)
 - Contoh: integer, real, char, **boolean**
- Tipe bentukan (*user-defined datatype*)
 - Tipe yang didefinisikan sendiri oleh pemrogram
 - Tipe yang dibentuk dari tipe dasar atau dari tipe bentukan lain yang sudah didefinisikan
 - Contoh: tipe dasar yang diberi nama tipe baru, record



Tipe Data

Empat hal yang harus diperhatikan dalam pendefinisian tipe:

- Nama
- Domain harga
- Konstanta/Literal
- Operator



Type Data

NAMA	DOMAIN	KONSTANTA	OPERATOR
boolean	true, false	true, false	and, or, xor, not
integer	tak terbatas	ditulis tanpa titik desimal	Aritmatika : +, -, *, div, mod Perbandingan : =, ≠, <, ≤, ≥, >
real	tak terbatas	ditulis dengan titik desimal	Aritmatika : +, -, *, / Perbandingan : =, ≠, <, ≤, ≥, >
char	huruf, angka, tanda baca, operator aritmatika, karakter khusus	harus diapit tanda petik tunggal ('...')	Perbandingan : =, ≠, <, ≤, ≥, >
string	Deretan karakter	harus diapit tanda petik tunggal ('...')	Penyambungan : + Perbandingan : =, ≠



Tipe Data

- X, Y, Z adalah variable bertipe Boolean
- X bernilai true, Y bernilai false, Z bernilai true
- Operasi logika :
 - $(X \text{ and } Y) \text{ or } Z$
 - $X \text{ and } (Y \text{ or } Z)$
 - $\text{Not } (X \text{ and } Z)$
 - $(Y \text{ xor } Z) \text{ and } Y$



Tipe Bentukan (User-Defined)

- Tipe dasar yang diberi nama tipe baru
 - Nama baru untuk tipe dasar menggunakan kata kunci **typedef** (dalam bhs C)
 - Domain nilai, cara menulis konstanta, dan operasi-operasi yang dapat dijalankan pada tipe baru tersebut tidak berubah, sama seperti tipe dasarnya.
 - Contoh: **typedef int BilanganBulat**



Tipe Bentukan (User-Defined)

- Rekaman (record)
 - Rekaman disusun atas satu atau lebih field
 - Tipe field menyimpan data dan tipe dasar tertentu atau dari tipe bentukan lain yang sudah didefinisikan sebelumnya
 - Nama rekaman ditentukan oleh pemrogram
 - Rekaman disebut juga tipe terstruktur



Nama

- Untuk mengidentifikasi dan membedakan objek
- Unik dan tidak boleh sama
- Dalam algoritma nama diberikan pada:
 - **Variabel**
 - Tempat penyimpanan data/informasi di memori yang nilainya dapat diubah selama pelaksanaan program
 - **Konstanta**
 - Tempat penyimpanan di memori yang nilainya tidak dapat diubah selama pelaksanaan program



Nama

- **Tipe bentukan**
 - Tipe data baru yang didefinisikan oleh program dari tipe data yang sudah ada
- **Prosedur**
 - Modul program (sederetan instruksi) yang ditulis terpisah dari badan program utamadan dapat dipanggil berulang dari program utama
- **Fungsi**
 - Prosedur yang mengembalikan suatu nilai dengan tipe data sederhana



Aturan Penulisan Nama

- Harus dimulai dengan huruf alfabet, tidak boleh dimulai dengan angka, spasi, atau karakter khusus lainnya.
- Tidak case sensitif (beda dengan bahasa pemrograman)
- Karakter penyusun nama hanya boleh: huruf alfabet, angka dan “_” (underscore)
- Tidak boleh dipisahkan dengan spasi
- Panjang nama tidak terbatas
- Semua nama yang dipakai harus dideklarasikan dulu pada bagian deklarasi



Contoh Penamaan

- *SALAH*

- 6titik {dimulai dg angka}
- nilai ujian {dipisahkan spasi}
- PT-1 {mengandung operator kurang}
- hari! {mengandung karakter khusus}

- *BENAR*

- titik6 **atau** titik_6
- nilai_ujian **atau** nilaiUjian
- PT_1 **atau** PT1
- hari



Apa kesalahan dari program ini?

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int nilai masukan;
    int nilai$ = 10;

    printf("Masukkan nilai pertama :");
    scanf("%d", &nilai masukan);
    hasil = nilai masukan + nilai$;
    printf("Hasilnya adalah : %d", hasil);
}
```



Nilai (Value)

- Merupakan besaran dari tipe data yang sudah didefinisikan (tipe dasar maupun tipe bentukan)
- Nilai dapat berupa:
 - Isi variabel atau konstanta
 - Nilai dari hasil perhitungan
 - Nilai yang dihasilkan oleh fungsi
- Nilai yang disimpan di variabel dimanipulasi dengan cara:
 - Mengisikan ke variabel lain yang bertipe sama
 - Dipakai untuk perhitungan
 - Dituliskan ke piranti keluaran



Nilai (Value)

Pengisian nilai ke variabel:

- Pengisian nilai secara langsung (assignment)
 - Memasukkan sebuah nilai ke dalam nama variabel langsung di dalam teks algoritma
 - Syaratnya nilai yang diisikan harus bertipe sama dengan tipe peubah
 - Notasi: = (dalam C)
 - Contoh:

variabel = konstanta	NoMhs = 1234
variabel1 = variabel2	Nil_prev = Nil_cur
variabel = ekspresi	$\text{Luas} = 0.5 * p * l$



Nilai (Value)

- Pembacaan nilai dari piranti masukan
 - Nilai untuk nama variabel dapat diisi dari piranti masukan, misalnya dari keyboard.
 - Dinamakan dengan operasi pembacaan data
 - Sintaks dalam bahasa C: `scanf`
 - Contoh:
 - `scanf("%d",&bilangan);`



Ekspresi

- Ekspresi terdiri atas: *operand* dan *operator*
- Operand adalah nilai yang dioperasikan dengan operator tertentu
- Operand dapat berupa konstanta, nama variabel, nama konstanta, atau hasil suatu fungsi
- Hasil evaluasi dari sebuah ekspresi adalah nilai di dalam domain yang sesuai dengan tipe operand yang dipakai, ada tiga macam: ekspresi aritmetik, ekspresi relasional, ekspresi string.



Ekspresi

- Ekspresi Aritmatika

- Ekspresi yang baik operand dan hasilnya berupa numerik
- (ingat: tingkat prioritas operator)
 - i. / , div, mod
 - ii. *
 - iii. +, -

- Ekspresi relasional

- Ekspresi dengan operator <, ≤, >, ≥, =, ≠, not, and, or, dan xor
- Hasil evaluasi adalah nilai bertipe boolean
- Ekspresi string



Ekspresi

- Ekspresi string
- Ekspresi dengan operator penyambungan/concatenation “+”.





UNIVERSITAS
ATMA JAYA YOGYAKARTA
serviens in lumine veritatis



SEKUENS & ASSIGNMENT



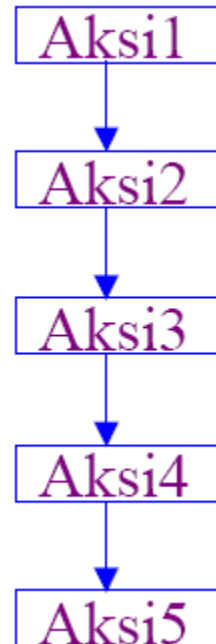


Azas

- Algoritma merupakan runtunan (*sequence*) satu atau lebih instruksi, yang berarti bahwa :
 - Tiap instruksi dikerjakan satu per satu (*dari atas ke bawah dan dari kiri ke kanan*)
 - Tiap instruksi dilaksanakan satu kali
 - Urutan instruksi yang dilaksanakan pemroses sama dengan urutan aksi sebagaimana tertulis di dalam Algoritma
 - Akhir dari instruksi terakhir merupakan akhir algoritma



Contoh 5 sekuens



Runtunan yang terdiri dari 5 buah instruksi.

Tiap instruksi dikerjakan satu per satu secara berurutan, sesuai dengan urutannya di dalam algoritma



Contoh Sekuens

- Contoh :

- $A = 10$
- $B = 20$
- $C = A + B$
- printf("%d",C)

$B = 20$
 $A = 10$
 $C = A + B$
printf("%d",C)



UNIVERSITAS
ATMA JAYA YOGYAKARTA

serviens in lumine veritatis

Urutan Penting

- Urutan di dalam algoritma sangat penting. Urutan instruksi yang berbeda akan menghasilkan hasil yang berbeda pula





Contoh

- Contoh :

A = 10

A = 2 * A

B = A

printf("%d",B)

=20

A = 10

B = A

A = 2 * A

printf("%d",B)

=10



ASSIGNMENT

- Assignment : Penugasan/Pengesetan suatu variable dengan value yang lain.
- Ciri:
 - Value sebelah kiri akan diubah menjadi value sebelah kanan
 - Value sebelah kanan tetap nilainya
- Contoh:
 - `A=10;` {A diset berisi 10. A setelah ini akan berisi 10}
 - `A=B;` {A diset menjadi bernilai sama dengan nilai nya B. B tetap. Lazimnya B sudah berisi sesuatu.}



Tebakan

A=10;

B=20;

A=B;

Berapa isi A dan B?

A=0;

B=A;

B=2;

Berapa isi A dan B

Catt: Bedakan dengan '=='



Tebakan

```
#include <stdio.h>
#include <conio.h>

void main()
{
    int A,B,C;

    A = 10;
    B = 15;
    C = 5;
    printf("Nilai A adalah : %d", A);
    printf("Nilai B adalah : %d", B);
    printf("Nilai C adalah : %d", C);
    B = A;
    C = A + B + C;
    A = C;
    printf("Nilai A adalah : %d", A);
    printf("Nilai B adalah : %d", B);
    printf("Nilai C adalah : %d", C);
}
```



Latihan (1)

- Buatlah program untuk menerima 5 masukan berupa bilangan integer (pakai 5 variabel), dan menampilkan nilai rata-rata dan jumlah total dari seluruh masukan tersebut!





Latihan (2)

- Pada saat kode program di bawah ini dieksekusi, apakah sudah benar? Jika belum, apa penyebabnya? Mengapa?

```
#define phi 3.14

int main()
{
    float r=5, h=9;

    printf("Volume tabung: %.2f\n", V);
    V = phi * r * r * h;

    return 0;
}
```



Latihan (3)

- Pada saat kode program di bawah ini dieksekusi, apakah sudah benar? Jika belum, apa penyebabnya? Mengapa?

```
int main()
{
    int a, t, L;

    printf("Alas    : "); scanf("%d", &a);
    printf("Tinggi : "); scanf("%d", &t);

    L = (a * t)/2;

    printf("Luas segitiga: %d\n", L);

    return 0;
}
```



Latihan (4)

- Program untuk menghitung luas permukaan 3 buah tabung berukuran sama ini apakah sudah benar? Jika belum, apa penyebabnya? Mengapa?

```
#include <math.h>
#define phi 3.14

int main()
{
    float r=5, h=9, L;

    printf("Jari-jari tabung: "); scanf("%f", &r);
    printf("Tinggi tabung   : "); scanf("%f", &h);

    L = 3 * phi * pow(r,2) + 2 * phi * r * h;

    printf("Luas permukaan ketiga tabung: %.2f\n", Luas);

    return 0;
}
```



Question and Answer



Terima kasih



uajy



Universitas Atma Jaya Yogyakarta



www.uajy.ac.id