

ALGORITMA DAN PEMROGRAMAN

[Komang Aryasa | [Pertemuan 04]

Outline



- Tipe data dasar
- Tipe data bentukan
-
-
-



Elemen Data

- ***Tipe data (data type)*** : setiap data memiliki tipe data, apakah data itu merupakan angka bulat (integer), angka biasa (real), atau berupa karakter (char), dsb.
- ***Variabel (variable)*** : setiap data diwakili oleh suatu variable, data memerlukan wadah yaitu variabel, dan variable perlu identifikasi, sehingga perlu diberi nama agar bisa dibedakan terhadap variable lainnya.
- ***Nilai (data value)*** : setiap data memiliki harga atau nilai, misalnya umur seseorang diwakili oleh variabel UMUR yang bertipe bilangan bulat, dan memiliki nilai 20 tahun. Perlu diketahui bahwa dalam representasi nilai data dalam komputer setiap tipe data memiliki batasan nilai masing-masing.



Tipe Data

- Ada dua kategori utama, yaitu:
 - Tipe data dasar
 - Tipe data bentukan



Tipe Data Dasar

- Tipe data yang secara otomatis disediakan oleh setiap bahasa pemrograman, antara lain:
 - Bilangan bulat (integer)
 - Bilangan tidak bulat / biasa
 - Bilangan tetap (konstan)
 - Karakter
 - Logik



Bilangan bulat

- Bilangan atau angka yang tidak memiliki titik desimal atau tidak ada pecahan, seperti: 10, +255, -1024, +32767.
- Tipe dituliskan sebagai : **integer** atau **int**
- Jangkauan nilai : bergantung pada implementasi perangkat keras komputer, misalnya dari -32768 s/d +32767, untuk algoritma tidak kita batasi.
- Operasi aritmetik : tambah + , kurang - , kali * , bagi / , sisa hasil bagi %
- Operasi perbandingan : lebih kecil < , lebih kecil atau sama < = , lebih besar > , lebih besar atau sama > = , sama = atau == , tidak sama >< , atau < > atau !=



Bilangan tidak bulat / biasa

- Bilangan atau angka yang bisa memiliki titik desimal atau pecahan, dan ditulis sebagai: 235.45, +1023.55, -987.3456 atau dalam notasi ilmiah seperti: $1.245E+03$, $7.45E-02$, $+2.34E-04$, $-5.43E+04$, dsb.
- Tipe dituliskan sebagai : **real** atau **float**
- Jangkauan nilai : bergantung pada implementasi perangkat keras komputer, misalnya dari $-2.9E-39$ s/d $+1.7E+38$, untuk algoritma tidak kita batasi.
- Operasi aritmatik dan perbandingan juga berlaku bagi bilangan biasa.



Bilangan tetap / konstan

- Bilangan tetap (**const**) adalah tipe bilangan baik bernilai bulat maupun tidak yang nilainya tidak berubah selama algoritma dilaksanakan.
- Tipe dituliskan sebagai : **const**
- Jangkauan nilai meliputi semua bilangan yang mungkin
- Contoh: $\pi = 3.14152$



Karakter (Character)

- Karakter adalah data tunggal yang mewakili semua huruf, simbol baca, dan juga simbol angka yang tidak dapat dioperasikan secara matematis, misalnya: 'A', 'B', ... , 'Z', 'a', 'b', ..., 'z', '?', '!', ':', ';' dst.
- Tipe dituliskan sebagai : **char**
- Jangkauan nilai meliputi semua karakter dalam kode ASCII, atau yang tertera pada setiap tombol keyboard.
- Operasi perbandingan dapat dilakukan dan dievaluasi menurut urutan kode ASCII, sehingga huruf 'A' (Hex 41) sebenarnya lebih kecil dari huruf 'a' (Hex 61).



Logik (Logic)

- Tipe data logik adalah tipe data yang digunakan untuk memberi nilai pada hasil perbandingan, atau kombinasi perbandingan.
- Tipe dituliskan sebagai : **boolean**
- Jangkauan nilai ada dua: **true (T)** dan **false (F)**
- Contoh: $45 > 56$ hasilnya **false**, $\text{Amir} < \text{Husni}$ hasilnya **true**
- Ada beberapa operasi untuk data jenis logik, antara lain: **and**, **or**, dan **not**



A	B	A and B	A or B	Not A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False



Tipe Bentuk

- Tipe data yang harus didefinisikan oleh pengguna-nya, dan dibentuk berdasarkan tipe data dasar, antara lain:
 - Larik
 - Record
 - String

Larik (array)

- Suatu larik dibentuk untuk menampung beberapa atau N buah data sejenis, $N > 1$.
- Kumpulan bilangan bulat adalah "integer array", kumpulan bilangan bulat disebut "real number array".
- Cara mendefinisikan suatu larik bilangan bulat yang terdiri atas 10 bilangan adalah:
 - Integer larik_bilbulat[10];
- Berikut ini adalah larik dari 7 nilai ujian:
 - Float Nilai_ujian[7];



String

- String adalah tipe data bentukan yang merupakan deretan/larik karakter yang membentuk satu kata atau satu kalimat, yang biasanya diapit oleh dua tanda kutip.
- Sebagai contoh : nama, alamat, dan judul adalah tipe string.
- Cara mendefinisikannya adalah:
 - ***String Nama, Alamat;*** atau
 - ***Char Nama[20], Alamat[25];***

Record

Record adalah tipe data bentukan yang merupakan wadah untuk menampung elemen data yang tipe-nya tidak perlu sama dengan tujuan untuk mewakili satu jenis objek.

- Sebagai contoh, mahasiswa sebagai satu jenis objek memiliki beberapa elemen data seperti: `nomer_stb`, `nama`, `umur`, `t4lahir`, `jenkel`.
- Cara mendefinisikan record mahasiswa tersebut adalah sbb:
- ***Type DataMhs : record***
 - < `nomer_stb : integer,`
 - `nama_mhs : string,`
 - `umur : integer,`
 - `t4lahir : string,`
 - `jenkel : char;`
 - >

Peubah (Variable)

- Variabel adalah suatu nama yang diberikan kepada suatu data, sehingga bisa mewakili data, seperti: nama, alamat, umur, nilai_ujian, no_mhs, dsb.
- Pada dasarnya setiap orang bebas memilih nama untuk variabel yang akan digunakan dalam algoritma/program-nya.
- Namun harus mengikuti suatu aturan antara lain sebagai berikut:



Aturan Nama Peubah

- Harus dimulai dengan abjad, tidak boleh dengan angka atau simbol
- Tidak boleh ada spasi diantaranya
- Jangan menggunakan simbol-simbol yang bisa membingungkan seperti titik dua, titik koma, koma, tanda tambah, dsb.
- Sebaiknya memiliki arti yang sesuai dengan elemen data
- Sebaiknya tidak terlalu panjang



Contoh nama variabel

- Contoh **nama variable yang benar** : Nama, Alamat, Nilai_Ujian, no_mhswa, nilai_x1
- Contoh **nama variable yang salah** : 4XYZ, x 1, IP rata, Var:+xy,458;

Assignment (pemberian nilai)

- Setiap variabel harus memiliki nilai agar dapat diproses, cara memberikan nilai disebut “assignment” adalah sebagai berikut:
 1. Beberapa buku algoritma menggunakan tanda panah ke-kiri, misalnya:
 - Nama \leftarrow “Ali bin AbuThalib”;
 - Jarak \leftarrow 100.56;
 - X \leftarrow Jarak;
 - Rentang \leftarrow X + 50 – 3*Y;
 2. Sering pula dijumpai tanda panah ke-kiri diganti tanda sama-dengan, misalnya: umur = 20, dsb



Operator

- Operator digunakan untuk melakukan operasi terhadap variabel atau nilai, antara lain:
- Operator aritmetik: + - * /
- Operator relasional: > >= == <= < !=
- Operator logik : and (&&), or (||), not (!)



Operator aritmetika

- Memiliki presedence (urutan pelaksanaan):
 - **urutan pertama** : ekspresi dalam tanda kurung
 - **urutan kedua** : perkalian atau pembagian
 - **urutan ketiga** : penjumlahan atau pengurangan
- Sebagai contoh:
 - $3 + 5 * 2$ hasilnya 13
 - $6 / 3 * 2 + 4$ hasilnya 8
 - $(3 + 5) * 2$ hasilnya 16
 - $6 / (3 * 2) + 4$ hasilnya 5



Ekspressi (expression)

- Ekspressi adalah transformasi data dan peubah/variabel dalam bentuk persamaan yang direlasikan oleh *operator* dan *operand*.
 - Operand** adalah data, tetapan, peubah/variabel, atau hasil dari suatu fungsi.
 - Operator** adalah simbol-simbol yang memiliki fungsi untuk menghubungkan operand sehingga terjadi transformasi



Ekspressi Aritmetika

- **Ekspressi Aritmetika** : ekspresi yang memuat variabel dan operator aritmetika, contoh:
 - $T \leftarrow 5 * (C + 32) / 9;$
 - $Y \leftarrow 5 * ((a + b) / (c + d) + m / (e * f));$
 - $Gaji \leftarrow GaPok * (1 + JumNak * 0.05 + Lembur * 1.25);$



Ekspressi Relasional

- **Ekspressi Relasional** : ekspresi yang memuat variable dan operator relational, contoh:
 - $\text{Nilai_A} > \text{Nilai_B}$
 - $(A + B) < (C + D)$
 - $(x + 57) \neq (y + 34)$



Ekspressi Logik

- **Ekspressi Logik** : ekspresi yang memuat variable dan operator logik, contoh:
 - $m \leftarrow (x > y) \ \&\& \ (5 + z)$
 - $n \leftarrow (!A \ || \ !(B \ \&\& \ C))$



Contoh Algoritma ekspresi

SOAL-1: Susun algoritma yang menghitung pajak pertambahan nilai (ppn) 12.50% dengan meminta harga barang yang dibeli dari pengguna program.

Langkah utama dalam masalah ini adalah: menetapkan harga barang, hitung pajaknya, hitung total yang harus dibayar, kemudian menampilkan hasilnya.



Algoritma PPN

{ menghitung pajak pertambahan nilai 12.50% dari harga barang }

Defenisi Variabel

real harga, pajak, total;

Rincian Langkah

{ Menetapkan harga barang }

Harga \leftarrow 15250;

{ Menghitung pajak }

pajak \leftarrow $0.125 * \text{harga}$;

{ Menghitung total pembayaran }

total = harga + pajak;

{ Tampilkan hasil }

Tampilkan("Harga = ", harga, " pajaknya = ", pajak);

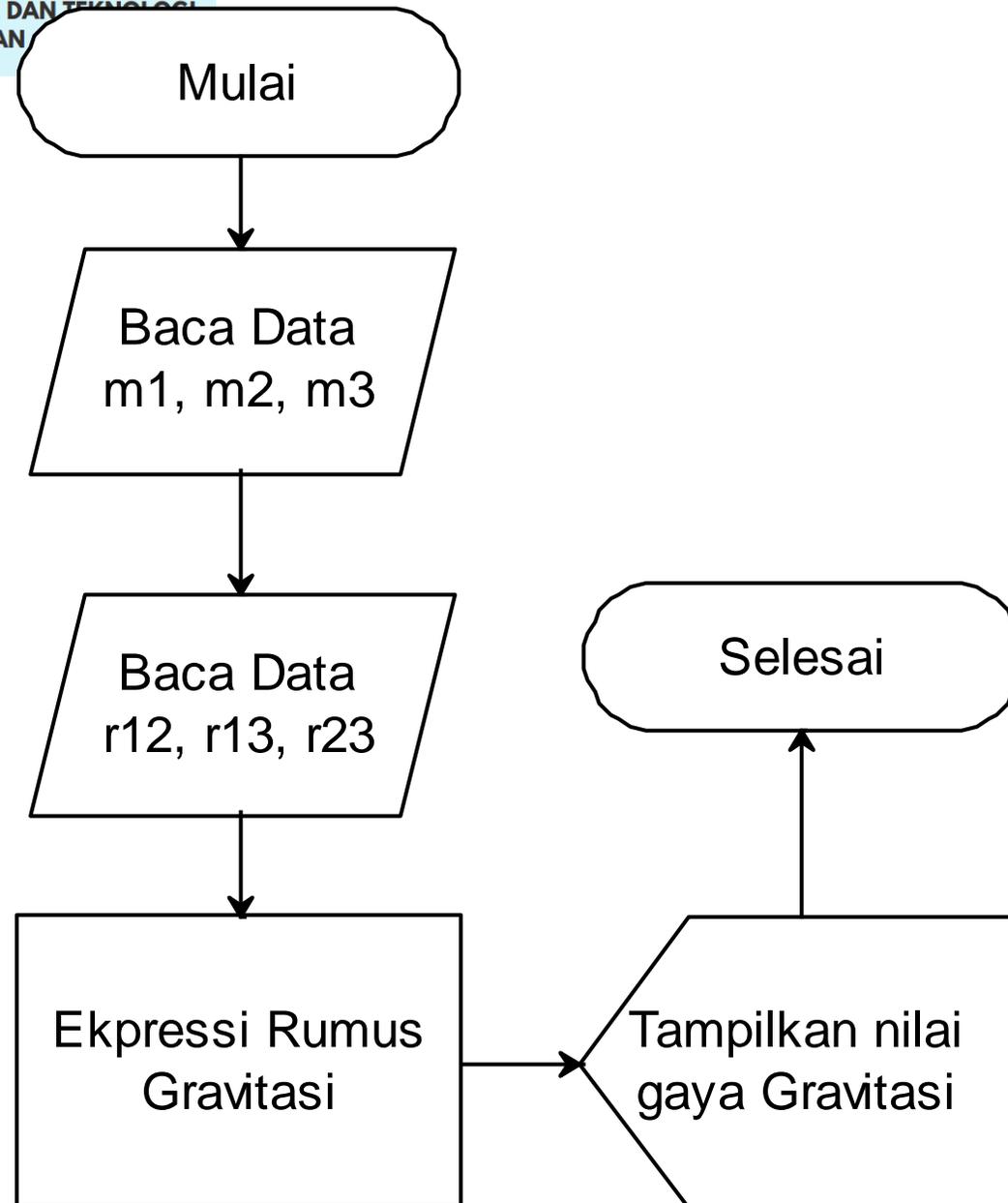
Tampilkan("Total = ", total);



- **SOAL 2** : MenuruthukumGravitasi Newton, massaakanmemilikigayagravitasi diantaramereka. Susunalgoritma yang membacamassatigabenda m_1 , m_2 , dan m_3 , kemudianjarakdiataramereka r_{12} , r_{13} , dan r_{23} . Tampilkanbesarnyagravitasi diantaraketigabendaini yang mengikutirumusgravitasisebagaiberikut:

- $$F = G \left(\frac{m_1 * m_2}{r_{12}} + \frac{m_1 * m_3}{r_{13}} + \frac{m_2 * m_3}{r_{23}} \right)$$

- dimana $G = 6.67 \times 10^{-11}$ newton- m^2/kg^2 , tetapangravitasi universal.





- { menghitung gaya gravitasi diantara massa }

-

- **Definisi Variabel**

- **const** G = 6.67E-11;

- **real** m1, m2, m3;

- **real** r12, r13, r23;

- **real** gaya;

-

- **Rincian Langkah**

- { membaca data massa dan jarak }

- Tampilkan ("Masukkan massa 1 : ");

- Baca(m1);

- Tampilkan("Masukkan massa 2 : ");

- Baca(m2);

- Tampilkan("Masukkan massa 3 : ");

- Baca(m3);

- Tampilkan ("Masukkan jarak antara massa-1 dan massa-2 : ");

- Baca(r12);

- Tampilkan("Masukkan jarak antara massa-1 dan massa-3 : ");

- Baca(r13);

- Tampilkan ("Masukkan jarak antara massa-2 dan massa-3 : ");

- Baca(r23);

- { menghitung gaya gravitasi }

- gaya $\leftarrow G * (m1*m2/r12 + m1*m3/r13 + m2*m3/r23)$;

- { menampilkan hasil }

- Tampilkan ("Besarnya gaya diantara ketiga massa ini = ", gaya);