

Algoritme dan Pemrograman

Kuliah #1

- Kontrak Perkuliahan
 - Pendahuluan
- Pembuatan algoritme

Kenapa belajar algoritme dan pemrograman?

- Profesi di banyak bidang (tidak hanya bidang komputer) melibatkan:
 - Pengolahan data yang banyak
 - Penyajian data dan informasi
 - Pengambilan kesimpulan berdasarkan informasi
- Komputer memungkinkan:
 - Pengerjaan teknik lama dengan lebih efisien (pencarian, pengurutan, dll)
 - Pengembangan teknik-teknik baru (kecerdasan buatan, dll)
- Banyak teknik komputasi yang terinspirasi dari bidang lain:
 - Jaringan syaraf tiruan
 - *Simulated annealing* (dari pembuatan baja)
 - Algoritme genetika
 - Algoritme koloni semut (dari perilaku semua dalam mencari makanan)

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Contoh penerapan

- Statistika: quick count, *computational statistics* (banyak dipakai di kecerdasan buatan)
- Klimatologi: simulasi iklim, prediksi musim, pengolahan dan penyajian data stasiun cuaca (cari: NOAA dan IRI/LDEO Climate Data Library)
- Matematika: *computational mathematics* (pembuktian dengan komputer, metode numerik)
- Fisika: simulasi partikel, pengolahan data fisika elementer, *computational physics* (banyak metode numerik)
- Biologi: bioinformatika
- Teknik sipil: studi struktur, simulasi (banyak metode numerik)

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Kiat Sukses

- Sadari perbedaan mata kuliah ini dengan yang lainnya
 - Membuat solusi sendiri, bukan semata mengikuti langkah-langkah yang telah ditetapkan
- Perbanyak latihan membuat algoritme
- Perbanyak latihan membuat program
- Buatlah soal-soal baru (mis. dengan memvariasikan soal yang ada) untuk latihan
- Aktif bertanya baik di kelas, lab, maupun di milis
- Kecurangan dalam pengumpulan tugas tidak akan membantu di ujian

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Program Komputer

- Komputer hanyalah sebuah **mesin**.
- Komputer bekerja sesuai **perintah** atau **instruksi** yang diberikan.
- Perintah atau instruksi yang diberikan kepada komputer untuk memecahkan suatu masalah disebut **program komputer**.
- Program komputer ini memandu kerja dari komputer yang secara berstruktur disusun oleh manusia.
- Bahasa yang digunakan untuk menyusun program komputer disebut **bahasa pemrograman komputer**.

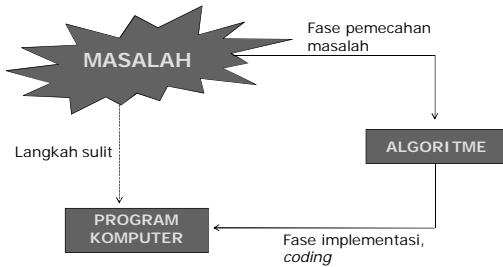
DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Membuat Program Komputer

- Pemrograman komputer harus dilakukan secara **sistematis**, dimulai dengan merancang atau mendesain langkah-langkah yang tepat yang menjelaskan jawaban dari suatu masalah.
- Langkah-langkah yang disusun secara **berstruktur** dan **terurut** untuk menjawab suatu persoalan dengan menggunakan bahasa manusia inilah yang sering disebut dengan **algoritme**.

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Langkah Membuat Program Komputer



DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Penulisan algoritme vs program

- Sebenarnya, cara menuliskan algoritme adalah bebas, sedangkan program dituliskan sesuai bahasa pemrograman tertentu (dalam MK ini: bahasa pemrograman C)
- Namun, agar dapat dipahami oleh orang lain, maka penulisan algoritme cenderung baku, sering disebut sebagai **pseudocode**. Salah satu contoh:

```

procedure jumlah {
  read a, b
  c ← a+b
  print c
}
  
```

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Contoh Algoritme #1

Menjumlahkan dua bilangan bulat

ALGORITME 1a. MENJUMLAHKAN DUA BILANGAN

1. baca dua buah bilangan bulat (misalnya a dan b).
2. jumlahkan dua bilangan tersebut dan simpan hasilnya ke peubah c ($c \leftarrow a + b$).
3. cetak nilai c.

ALGORITME 1b. MENJUMLAHKAN DUA BILANGAN

1. baca a dan b.
2. $c \leftarrow a + b$.
3. cetak nilai c.

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Algoritme vs Program (Contoh #1)

Algoritme

1. baca a dan b.
2. $c \leftarrow a + b$.
3. cetak nilai c.

Program

```

#include <stdio.h>

int main()
{
  short int a, b, c;
  scanf("%hd %hd", &a, &b);
  c = a + b;
  printf("%hd\n", c);
}
  
```

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Contoh Algoritme #2

Menentukan kelulusan dari tiga nilai

- Perumusan masalah:
Diberikan tiga buah nilai ujian. Jika rata-rata dari tiga nilai kurang dari 50, maka statusnya adalah TIDAK LULUS, sedangkan lainnya (lebih besar atau sama dengan 50) dinyatakan LULUS.

```

procedure kelulusan {
  read nil1, nil2, nil3
  r ← (nil1+nil2+nil3)/3
  if (r < 50) then print "TIDAK LULUS"
  else print "LULUS"
}
  
```

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Program C Contoh #2

```

#include <stdio.h>

int main()
{
  short int nil1, nil2, nil3, r;
  scanf("%hd %hd %hd", &nil1, &nil2, &nil3);
  r = (nil1 + nil2 + nil3)/3;
  if (r < 50) printf("TIDAK LULUS\n");
  else printf("LULUS\n");
}
  
```

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Latihan Kelas

Buat langkah-langkah untuk melakukan hal-hal berikut:

1. Menjumlahkan 4 bilangan bulat.
2. Menentukan bilangan terbesar dan terkecil dari tiga bilangan bulat.
3. Memeriksa apakah suatu bilangan bulat adalah positif, negatif, atau 0. Jika positif, cetak kata 'POSITIF'; jika negatif, cetak 'NEGATIF'.
4. Menentukan apakah sebuah bilangan bulat habis dibagi a ataukah tidak ($a > 0$).
5. Menentukan apakah sebuah bilangan bulat adalah bilangan genap atau ganjil.

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Algoritme apakah ini?

1. Baca (a, b, c)
2. Jika $a \geq b$, maka:
 besar \leftarrow a, kecil \leftarrow b,
 selainnya:
 besar \leftarrow b, kecil \leftarrow a
3. Jika $c \geq$ besar, maka besar \leftarrow c
4. Jika $c \leq$ kecil, maka kecil \leftarrow c
5. Cetak (besar, kecil)

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Elemen Pseudocode

- Variabel
- Perintah atau instruksi
- Assignment atau penugasan dan ekspresi
- Struktur kendali aliran
- Komentar
- Catatan: pseudocode yang kita gunakan mengadopsi beberapa ketentuan bahasa pemrograman C agar mudah diubah ke bahasa pemrograman C

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Elemen Pseudocode: VARIABEL

- Variabel adalah elemen dari algoritme untuk menyimpan nilai tertentu pada suatu saat, dan pada saat yang lain nilai ini dapat diubah menjadi nilai lainnya sesuai kebutuhan.
- Beberapa catatan agar mudah dipindah ke program C
 - Dituliskan dengan suatu nama yang unik, terdiri dari alphanumeric, tetapi harus dimulai dengan huruf atau *underscore*.
 - *Case-sensitive* (variabel X berbeda dengan variabel x)
- Contoh: a, b, c, n, sum,
- Pada contoh #1, mana saja yang termasuk variabel?

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Elemen Pseudocode: INSTRUKSI/PERINTAH

- Perintah/instruksi adalah satuan operasional dari suatu algoritme.
- Dinyatakan dalam kalimat sehari-hari yang mudah dipahami dan konsisten.
- Diakhiri tanda titik koma (;)
- Contoh:
 - read(a,b,c);
 - print(sum);

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Elemen Pseudocode: EKSPRESI dan ASSIGNMENT

- Ekspresi adalah operasi yang akan dievaluasi (diproses) untuk menghasilkan suatu nilai. Ekspresi dapat bersifat aritmetika (menghasilkan angka) maupun logika (menghasilkan kondisi *TRUE* atau *FALSE*).
- Assignment atau penugasan adalah pemberian suatu nilai di sisi kanan ke suatu penampung (mis. variabel) di sisi kiri.
 - Nilai di sisi kanan dapat berupa nilai literal, variabel lain, atau suatu ekspresi.
- Tanda "=" menggantikan tanda " \leftarrow "
- **Bukan** bermakna kesamaan
- Contoh:
 - sum = 0; a = -7; max = 9.513;
 - x = n; b = a + 4; c = 4 / x;
 - c = a + b; sum = sum + x;

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Elemen Pseudocode: STRUKTUR KENDALI ALIRAN

- Struktur kendali aliran adalah suatu bentuk atau struktur yang memiliki peranan khusus untuk mengatur atau mengendalikan urutan pengerjaan satu atau lebih operasi atau instruksi.
- Struktur kondisi (*conditional*): if, if – else, switch
 - Pilihan berdasarkan syarat/kondisi tertentu
- Struktur pengulangan (looping): for, while
 - Perulangan berdasarkan syarat/kondisi tertentu
- Struktur lainnya: continue, exit, return

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Elemen Pseudocode: KOMENTAR

- Komentar adalah suatu catatan tambahan yang dituliskan ke dalam algoritme dengan tujuan untuk memperjelas, tetapi tidak mempengaruhi operasi apa pun didalam algoritme.
- Diberi notasi awal `/**`
- Dapat dituliskan sebagai baris terpisah atau setelah suatu pernyataan.
- Mulai tanda `/**` hingga akhir baris tidak diproses (tidak berpengaruh pada perilaku dan hasil program)
- Contoh:

```
// Algoritme menjumlahkan dua bilangan bulat
// Dibuat pada 2 September 2012
```

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Contoh #3 Masalah dan Perumusannya

- Masalah:
Buat algoritme menentukan apakah suatu bilangan bulat habis dibagi oleh a ($a > 0$)
- Pemecahan masalah:
Untuk menentukan apakah satu bilangan (misal b) habis dibagi oleh a , dapat dilakukan dengan menentukan sisa pembagian, yaitu menggunakan operasi modulo (mod). Jika bilangan $b \text{ mod } a = 0$ maka bilangan b tersebut berarti habis dibagi oleh a , dan algoritme akan mencetak teks YA. Selainnya, akan mencetak teks TIDAK.

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Contoh #3 Algoritme

```
// Memeriksa apakah b habis dibagi oleh a

procedure habisDibagi {
  read(b,a);
  t = b mod a;
  // Perhatikan tanda "==" untuk tes kesamaan
  if (t == 0) // Periksa nilai t
    print(" YA");
  else
    print(" TIDAK");
}
```

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Contoh #4 Masalah dan Perumusannya

- Masalah:
Buat algoritme mencetak bilangan 1, 2, 3, sampai dengan n ($n > 0$)
- Perumusan masalah:
Masalah ini mirip dengan mesin penghitung (*counter*) yang biasa digunakan pramugari untuk menghitung banyak penumpang pesawat yang masuk sampai jumlah penumpang yang seharusnya tercapai. Di awal, *counter* diset ke 0 (disebut inisialisasi, misalnya $c=0$). Setelah itu, tombol ditekan sehingga nilai sebelumnya bertambah 1 ($c=c+1$). Hal ini dilakukan seterusnya selama (while) c kurang dari n sambil mencetak nilai c .

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Contoh #4 Algoritme

```
// Mencetak bilangan 1, 2, sampai dengan n

procedure printN {
  read(n); // sampai bilangan berapa?
  c = 0; // inisialisasi
  while (c < n) { // selama c kurang dari n
    c=c+1; // nilai c sebelumnya ditambah 1
    print(c);
  }
}
```

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Men-trace algoritme

- Men-trace algoritme adalah suatu proses untuk menelusuri langkah demi langkah dari suatu algoritme dan menentukan output akhirnya.
- Dalam pemrograman, proses ini disebut juga sebagai proses *debug*.
- Tips:
 - Ikuti langkah demi langkah
 - Catat setiap perubahan nilai dari suatu variabel
 - Jika menemukan suatu pola tertentu, *output* akhir dapat langsung ditentukan tanpa mengikuti algoritme sampai selesai (hanya dilakukan oleh orang yang sudah sangat memahami algoritme)

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

Contoh men-trace algoritme

// Misal input n=4

```
procedure printN {
  read(n);
  c = 0;
  while (c<n) {
    c=c+1;
    print(c);
  }
}
```

n

c

output

1

2

3

4

STOP

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

LATIHAN #1

- Masalah:
Buat algoritme mencetak hasil perhitungan $b^2 - 4ac$.
- Contoh input (nilai a b c):
2 3 4
- Contoh output:
-23

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

LATIHAN #2

- Masalah:
Buat algoritme mencetak bilangan bulat ganjil (gasal) dari 1 sampai dengan n .
- Contoh input:
8
- Contoh output:
1
3
5
7

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

LATIHAN #3

- Masalah:
Buat algoritme mencetak bilangan bulat dimulai dari n sampai dengan 1.
- Contoh input:
5
- Contoh output:
5
4
3
2
1

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR

LATIHAN #4

- Masalah:
Buat algoritme menghitung jumlah dari beberapa bilangan bulat.
- Input:
Beberapa bilangan bulat dan diakhiri dengan nilai minus 99 atau -99 (bilangan terakhir ini tidak dijumlahkan)
- Output:
Jumlah dari bilangan masukan selain bilangan terakhir (minus 99)
- Contoh input:
20 -8 0 26 37 14 -99
- Contoh output:
89

DEPARTEMEN ILMU KOMPUTER INSTITUT
PERTANIAN BOGOR