PROGRAM STUDI

S1 SISTEM KOMPUTER

UNIVERSITAS DIPONEGORO

Algoritma dan Pemrograman

Oky Dwi Nurhayati, ST, MT
email: okydn@undip.ac.id

Buku Pegangan

- Anany Levitin, Introduction to the Design & Analysis of Algorithms, Addison-Wesley, 2003.
- Enem, S. Graph Algorithms, Computer Science Press, Inc, 1999.
- Kruth, D.E. : Fundamental Algorithms, Addison-Esley, 1975.
- La Budda, K : Structure Programming Concepts. Mc.Graw – Hill.
- Pemrograman Borland Delphi 6, Antony Pranata, Andi offset Yogyakarta

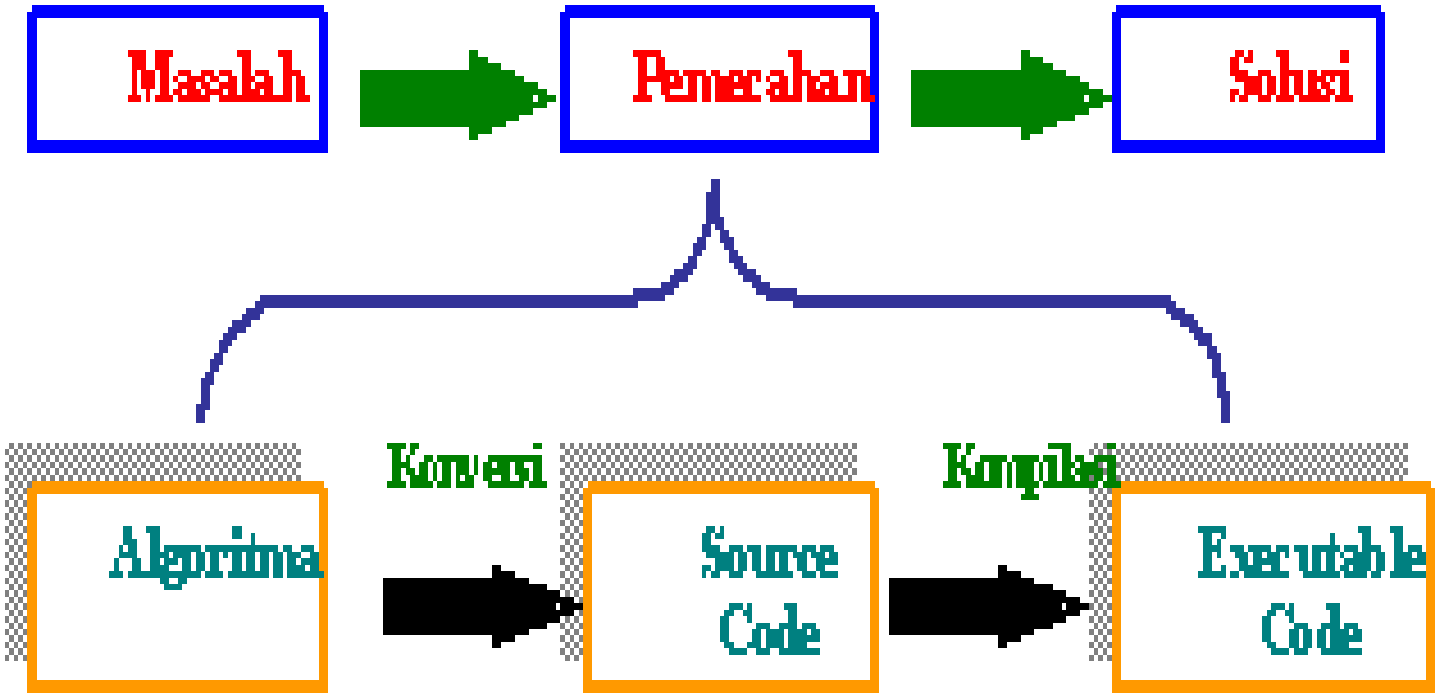
Minggu ke	Topik	Materi
1	Pendahuluan	- Definisi algoritma - Notasi matematis - Tahapan algoritma (Proses Pemrograman)
2	Dasar algoritma	- Penulisan algoritma dengan pseudocode - Masalah analisis algoritma - Masalah komputasi
3,4	Pendekatan Pemrograman Modular	- Model Top Down - Algoritma Prim - Algoritma Bouruvka

5	Sorting dan Searching	- Bubble sort - Selection sort - Insertion sort - Shell sort - Merge sort - Quick sort
6,7	Algoritma Greedy	- Definisi - Skema umum algoritma greedy - Minimisasi Waktu di dalam Sistem (Penjadwalan) - Pemecahan masalah dengan algoritma Greedy - Pohon merentang minimum - Kompleksitas algoritma: $O(n^2)$

8,9	Algoritma Divide and Conquer	- Definisi - Skema umum algoritma divide and conquer - Penyelesaian masalah dengan algoritma divide and conquer - Kompleksitas waktu algoritma - Algoritma pengurutan dengan divide and conquer
10,11	Pemrograman Dinamis	- Prinsip optimalitas - Karakteristik persoalan program dinamis - Contoh persoalan program dinamis

12	Algoritma paralel	- Model komputasi parallel - Teknik dasar - Evaluasi paralel - Parallel sorting
13,14	Kompleksitas Algoritma	- Pendahuluan - Reduksi linear - Kompleksitas beberapa algoritma

Pendahuluan



Gambar 1. Algoritma dalam kerangka Pemecahan masalah

Pendahuluan

Asal Usul Kata

Kata algoritma dari nama Abu Ja'fat Mohammed Ibn Musa al-Khowarizmi, seorang ilmuan Persia yang menulis buku berjudul Kitab al jabr w'al-muqabala (*rules of restoration and reduction*) sekitar tahun 825

Pendahuluan

Asal Usul Kata

pada tahun 1950 istilah *algorithm* selalu diasosiasikan dengan *Euclid's algorithm*, yaitu suatu proses yang menjelaskan cara mencari bilangan pembagi terbesar untuk dua buah bilangan.

Pendahuluan

Asal Usul Kata

Merriam-Webster's Collegiet Dictionary istilah *algorithm* diartikan sebagai prosedur langkah demi langkah untuk memecahkan masalah atau Penyelesaian suatu tugas khususnya dengan menggunakan bantuan komputer

Pendahuluan

Syarat Algotitma

Menurut Donald E Knuth algoritma harus memenuhi persyaratan ;

Finiteness
Definiteness
Input
Output
Effectiveness

Pendahuluan

Sebagai basis pemerograman komputer, algoritma mendeskripsikan kan urutan langkah-langkah yang diperlukan untuk pemecahan masalah (penyelesaian persoalan), yang memiliki ciri-ciri sebagai berikut;

- selalu memiliki terminasi/langkah akhir
- setiap langkah dinyatakan secara jelas dan tegas
- setiap langkah sederhana, sehingga kinerjanya sehubungan dengan waktu yang effisien/bisa diterima akal
- memberikan hasil (*output*), mungkin dengan satu atau tanpa input.

Pendahuluan

Tahapan Algoritma (Proses Pemrograman)

Proses pemecahan masalah dengan algoritma tertentu hingga menjadi program dapat dibagi dalam sembilan tahap;

- Mendefinisikan masalah

  Masalah yang ingin dipecahkan harus jelas lingkupnya.

- Membuat model

  Yang dimaksud model ini adalah model (bentuk) matematis yang dapat digunakan untuk memecahkan masalah, misalnya apakah harus dilakukan pengurutan terhadap data, apakah menggunakan perhitungan kombinatorik dan sebagainya.

Pendahuluan

Tahapan Algoritma (Proses Pemrograman)

- Merancang algoritma (*flowchart/pseudocode*)
  Apa maksudnya, bagaimana rincian prosesnya, apa keluarannya.
- Menulis program
  Ubah algoritma menjadi program (*source code*) dalam bahasa pemrograman tertentu.
- Mengubah *source code* menjadi *executable code* melalui proses *compiling.*
- Memeriksa hasil *compiling*, jika salah maka kembali ke tahap empat.

Pendahuluan

Tahapan Algoritma (Proses Pemrograman)

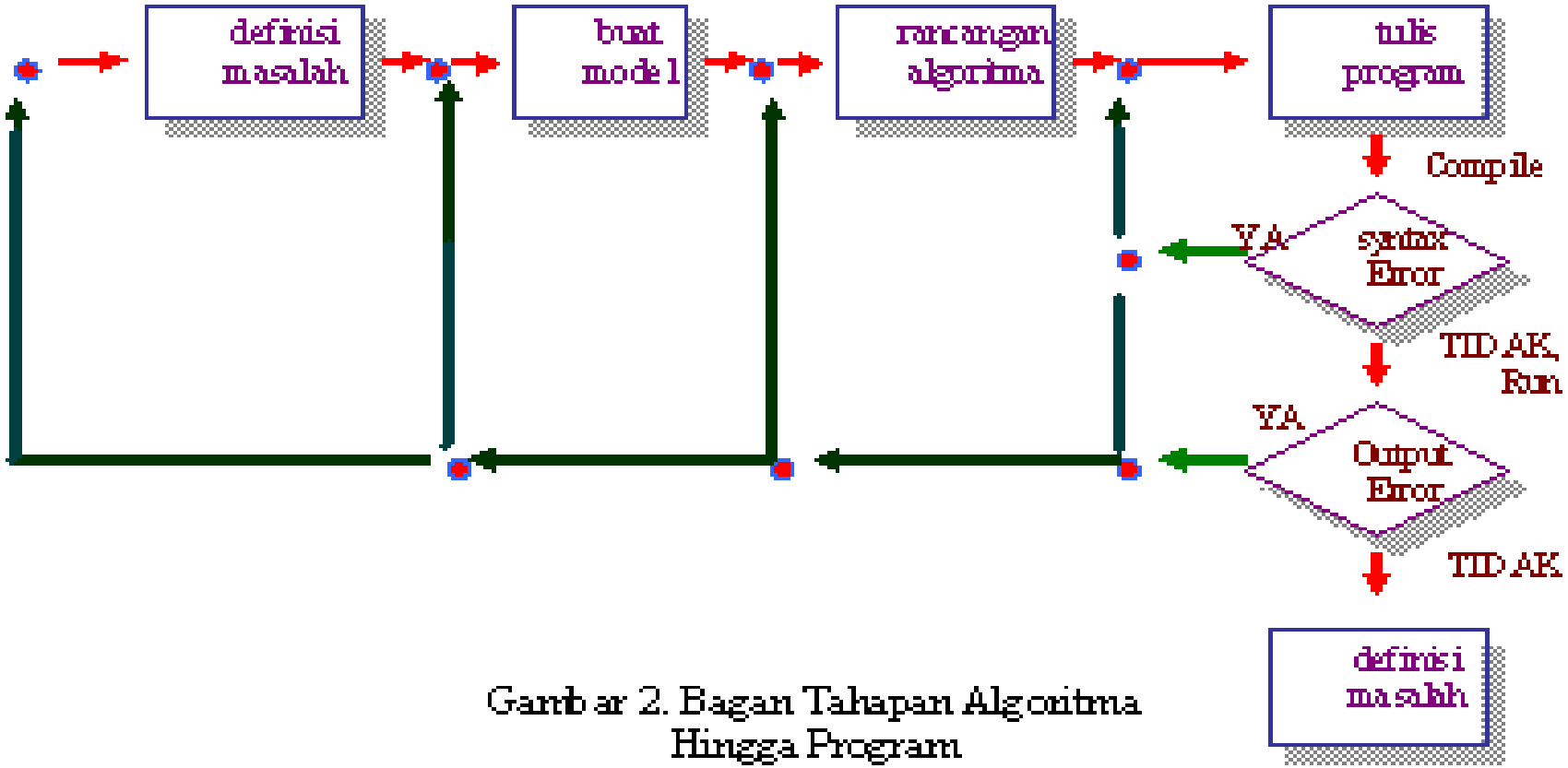
- Menjalankan program (*run*) untuk diuji kebenarannya dengan menggunakan berbagai data
- Memperbaiki kesalahan (*debugging dan testing*)

  Apabila hasilnya salah, kesalahan mungkin terjadi saat konversi rancangan algoritma manjadi program, atau salah rancang algoritma, atau salah menentukan model, atau salah mendefinisikan masalah. Ulangi langkah yang sesuai.

Pendahuluan

Tahapan Algoritma (Proses Pemrograman)

- Mendokumentasi program bila sudah benar.



Gambar 2. Bagan Tahapan Algoritma Hingga Program

Pendahuluan

Penulisan Algoritma dengan Pseudocode

Dengan menggunakan pseudocode pemecahan masalah di atas dapat ditulis sebagai berikut;

Contoh;

```
baca jumlah data
tulis jumlah data
while          data belum habis
      hitung   data yang dibaca
      baca              data no_mhs, nama, nilai
      tulis             no_mhs dan nama
               if       nilai > 6,0
               else     then
                        tulis     “LULUS”
               else
                        tulis     “TIDAK LULUS”
               else if
when
tulis garis penutup tabal
selesai
```

Implementasi program dalam bahasa BASIC;

baca jumlah data

Pendahuluan

Jenis Proses Algoritma

Langkah yang membentuk suatu algoritma dapat dibagi manjadi 3 kelompok proses;

1). *Sequence proces*

2). Selection proces

3). Iteration proces

Pendahuluan

Macam Algoritma

- Metode Seleksi
- Metode Sisipan
- Metode *Shell*
- *Metode Bubble*
- *Metode Cepat*
- *Metode Radix*
- *Metode Merge*
- *Metode Pohon Biner*
- *Metode Tournament*
- Metode *Heap*

Pendahuluan

Masalah Analisis Algoritma

Kinerja yang perlu ditelaah pada algoritma;

- beban komputasi
- efisiensi penggunaan memory

Pendahuluan

Masalah Analisis Algoritma

Tantangan yang dihadapi dalam membandingkan kinerja berbagai algoritma sangat berguna, yang perlu diperhatikan ;

- Kasus rata-rata; running time untuk tipikal data tertentu.
- Kasus terjelek; running time yang mungkin paling jelek pada konfigurasi masukan data tertentu
- Program → bahasa yang dipakai
- Program sensitif terhadap input
- Program sulit dimengerti, dan secara matematis hasil tah tersedia/ diketahui
- Sering kali program tidak bisa membandingkan, misal untuk data tertentu sangat efisien, tetapi yang lain pada kondisi yang sangat berbeda.

Pendahuluan

Masalah Komputasi

Menurut kesepakatan para ilmuwan komputer, algoritma berdayaguna untuk komputasi bila kompleksitas waktu O berkembang secara polinomial dalam respect terhadap ukuran input : *n* . Secara lebih positif, wajah polinomial suatu algoritma seperti:

O (*n*) , O (*n log n*),
O (*n*3), O (106 *n*8),
O (2*n*) , O (*n log n*),
O (*n* !) .

Pengelompokan Algoritma Berdasarkan Notasi *O*-Besar

Kelompok Algoritma	Nama
$O(1)$	konstan
$O(\log n)$	logaritmik
$O(n)$	lanjar
$O(n \log n)$	$n \log n$
$O(n^2)$	kuadratik
$O(n^3)$	kubik
$O(2^n)$	eksponensial
$O(n!)$	faktorial

Urutan spektrum kompleksitas waktu algoritma adalah :

$$\underbrace{O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^3) < \ldots <}_{\text{algoritma polinomial}} \underbrace{O(2^n) < O(n!)}_{\text{algoritma eksponensial}}$$

Pendahuluan

Masalah Komputasi

Dalam perkembangan yang lebih menguntungkan kompleksitas waktu dapat seperti:

$O(n)$, $O(n \log n)$, $O(n^2)$

$O(n^3)$, $O(108\, n^4)$, $O(2n)$

$O(10n)$, $O(n \log n)$, $O(n!)$

Pendahuluan

Menghadapi Banyaknya paket dalam TI dewasa ini

- Kita dituntut untuk dapat melakukan pengamatan tentang kinerja paket-paket yang bersangkutan dan menyajikan informasi sehubungan dengan efektivitas dan efisiensi masing-masing paket bagi persoalan yang akan dihadapi.
- Perlu memperhatikan masalah kompatabilitas sistem.
- Perlu memperhatikan tingkat kemudahan oprasional dan antarmuka.
- Perlu memperhatikan tingkat kemudahan perolehan paket dan populasi pamakaiannya.
- Perhatikan masalah pemeliharaan di masa mendatang.

Pendahuluan

Teknik Pemrograman

Penekanan pada pemrograman tersetruktur

- Struktur dasar
  Menggunakan *flow chart* dan *pseudocode*

Menggunakan sistem modular.

- Program dibuat dalam bentuk modul-modul untuk fungsi tertentu maupun subroutine tertentu.
- Modul-modul dikendalikan oleh modul utama (program utama)

Pendahuluan

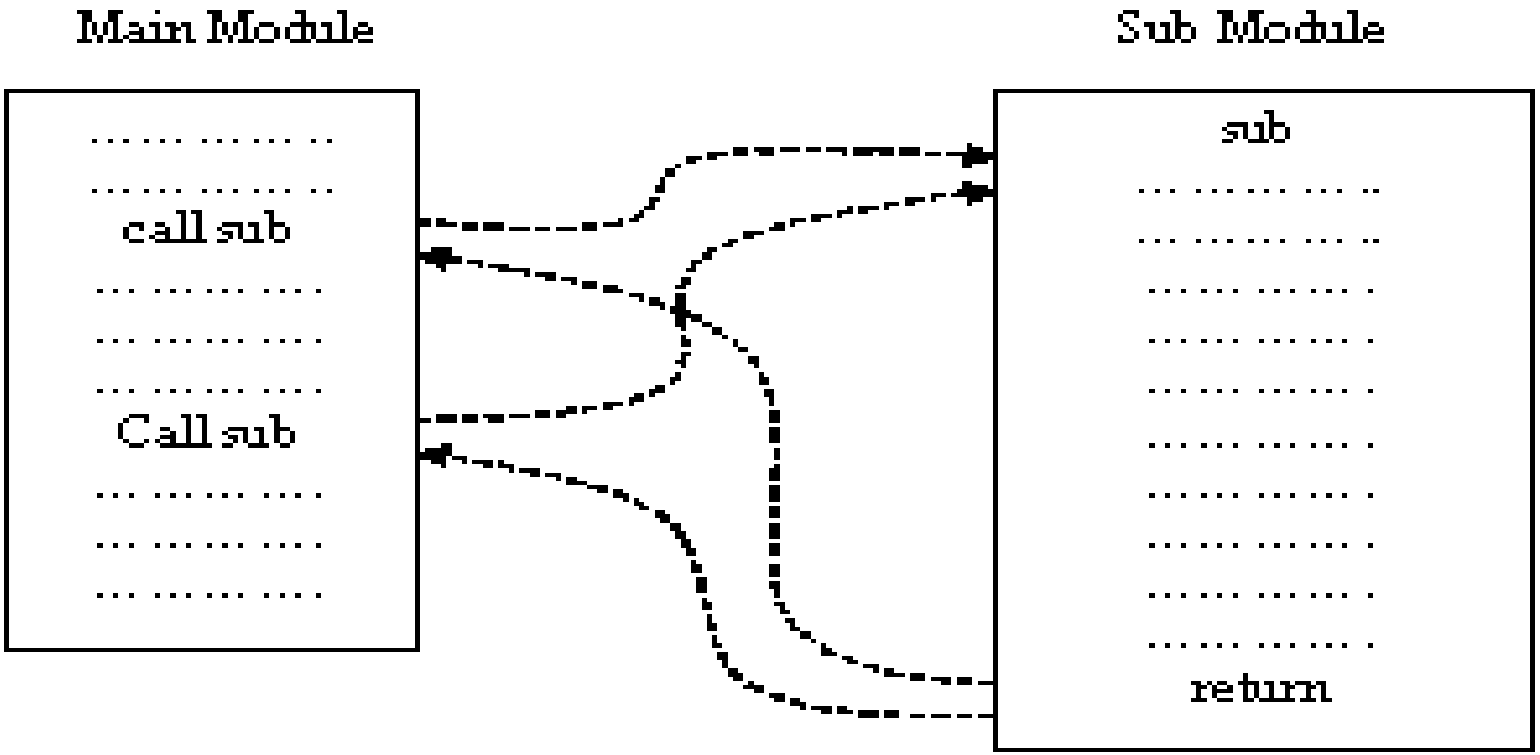
Teknik Pemrograman

- Modul bersifat independen (tidak ada modul yang dapat akses langsung ke modul lain, kecuali modul pemanggil dan submodulnya sendiri)
- Modul dapat diubah secara radikal tanpa mempengaruhi modul lain sejauh fungsi modul tidak berubah..

Pendahuluan

Teknik Pemrograman

Implementasi pemrograman modul menggunakan subroutine–subroutine digambarkan sebagai berikut;



Gambar 1. Main module & Sub Module

Pendahuluan

Bentuk sub module

- Internal subroutine; berupa bagian dari program yang menggunakannya
- External subroutine;
  - bukan bagian dari program yang menggunakan modul itu.
  - Modul tersimpan dalam *library* dalam bentuk "*object module*" yang siap digunakan oleh modul-modul yang akan menggunakan.
  - Modul dapat digunakan untuk tugas lebih dari satu program
- Dalam menggunakan, pemrograman perlu mengetahui di mana diperoleh, namanya, bagaimana mengirim data padanya, dan jabawan baliknya.