Ir. Djoko Luknanto, M.Sc., Ph.D.
mailto:luk@tsipil.ugm.ac.id

Pemrograman Komputer

2. Pengembangan Program

(luk@tsipil.ugm.ac.id)

2. Pengembangan Program

1. Analisis masalah dan spesifikasi
2. Design
3. Koding program
4. Eksekusi dan Testing
5. Rekayasa perangkat lunak

(luk@tsipil.ugm.ac.id)

2.1 Analisis Masalah & Spesifikasi

- Pada awal suatu proyek biasanya permasalahan yang dihadapi belum jelas sekali.
- Diperlukan suatu analisis masalah yang akan ditangani.
- Perlu diformulasikan dengan jelas "spesifikasi" dari masalah terkait:
  - "input," informasi yang harus diberikan sebagai data masukan dari masalah terkait,
  - "output," informasi yang harus dihasilkan dari masalah terkait.

(luk@tsipil.ugm.ac.id)

Kasus 1: Peluruhan radioaktif

- Seorang ahli nuklir dari Jurusan T. Nuklir UGM melakukan riset terhadap elemen radioaktif Polonium.
  - waktu paruh Polonium 140 hari yaitu waktu yang dibutuhkan Polonium sehingga massanya tinggal separuh dari massa awal.
  - ingin diketahui berapa Polonium yang tersisa setelah 180 hari, jika pada awal percobaan terdapat 10 mg.

(luk@tsipil.ugm.ac.id)

Kasus 1: input & output

- Input:
  - massa awal: 10 mg
  - waktu paruh: 140 hari
  - waktu percobaan: 180 hari
- Output:
  - massa tersisa
- Catatan: nama periset, nama universitas tidak penting dalam konteks ini.

(luk@tsipil.ugm.ac.id)

Kasus 1: Generalisasi

- Membuat program hanya untuk menyelesaikan kasus khusus kurang banyak gunanya.
- Oleh karena itu dibutuhkan generalisasi terhadap kasus khusus di atas.
- Generalisasinya menjadi:
  - Input: massa awal, waktu paruh, waktu percobaan
  - Output: massa tersisa

(luk@tsipil.ugm.ac.id)

Kasus 2: Index Polusi

- Polusi udara di Yogya akan diukur dengan index polusi.
- Pengukuran polusi dilakukan di 3 lokasi: di Malioboro, di Madukismo, dan di daerah pemukiman yang dipilih secara random.
- Rerata pengukuran di tiga lokasi ini didefinisikan sebagai index polusi.

(luk@tsipil.ugm.ac.id)

Kasus 2: ... lanjutan

- Nilai index polusi diatas 50 menunjukkan tingkat polusi yang membahayakan, sedangkan dibawah nilai tersebut dianggap aman.
- Karena index polusi harus dihitung harian, maka Pemda menginginkan sebuah perangkat lunak yang menentukan apakah polusi membahayakan atau tidak.

(luk@tsipil.ugm.ac.id)

Kasus 2: Generalisasi

- Nilai ambang polusi tidak hanya 50, namun digeneralisasi, sehingga analisis masalah menjadi:
  - input:
    - tiga data pengukuran
    - nilai ambang batas
  - output:
    - index polusi = rerata 3 pengukuran
    - kondisi polusi: bahaya atau tidak.

(luk@tsipil.ugm.ac.id)

Kasus 3: Waktu rerata sebelum gagal

- Salah satu faktor penting dalam mengukur reliabilitas suatu barang adalah "waktu rerata sebelum gagal," yang dapat digunakan untuk mengukur waktuguna suatu barang.
- Sebuah pabrik elektronik ingin mengetahui waktuguna sebuah komponen televisi.

(luk@tsipil.ugm.ac.id)

Kasus 3: input & output

- Input:
  - beberapa koleksi data pengukuran (yang jumlahnya tidak diketahui)
- Output:
  - jumlah pengukuran
  - nilai rerata pengukuran

(luk@tsipil.ugm.ac.id)

2.2 Design

- Setelah spesifikasi permasalahan selesai, kemudian disusun sebuah rancangan untuk membuat perangkat lunak yang sesuai dengan spesifikasi tersebut.
- Dua aspek penting dalam perancangan ini adalah:
  - memilih struktur yang tepat untuk mengatur dan menyimpan data, dan
  - merancang prosedur untuk memproses data.

(luk@tsipil.ugm.ac.id)

2.2 ... algoritma ...

- Pada dasarnya komputer tidak mempunyai kemampuan untuk menyelesaikan masalah.
- Oleh karena itu, prosedur yang dirancang harus rinci dan sederhana.
- Prosedur semacam ini biasa disebut algoritma.

26 Feb 2001 Pemrograman Komputer 13

(luk@tsipil.ugm.ac.id)

2.2 ... algoritma dan program terstruktur ...

- Algoritma atau program terstruktur dirancang dengan menggunakan tiga komponen:
  1. berurutan: tiap langkah harus dikerjakan secara urut satu per satu,
  2. pemilihan: satu dari beberapa alternatif dipilih dan dieksekusi,
  3. pengulangan: satu atau beberapa langkah diulang-ulang.

26 Feb 2001 Pemrograman Komputer 14

(luk@tsipil.ugm.ac.id)

2.2 ... penulisan algoritma ...

- program untuk mengimplementasikan algoritma harus ditulis dalam bahasa yang dimengerti komputer.
- oleh karena itu diusahakan dalam penulisan algoritma digunakan bahasa yang mirip dengan bahasa yang akan digunakan menulis program.

26 Feb 2001 Pemrograman Komputer 15

(luk@tsipil.ugm.ac.id)

2.2 ... beberapa cara penulisan ...

- operasi aritmetika:
  penambahan: +, pengurangan: -,
  perkalian: *, pembagian: /, pangkat: **
- data yang akan diproses diberi nama/simbol,
- komentar dan keterangan sebaiknya ditambahkan dalam algoritma,
- beberapa kata kunci dari bahasa pemrograman sebaiknya digunakan, misalkan read, write, display, print, dlsb.
- indentasi sebaiknya digunakan untuk menandai sebuah blok.

26 Feb 2001 Pemrograman Komputer 16

(luk@tsipil.ugm.ac.id)

Kasus 1: Peluruhan radioaktif

- Input yang dibutuhkan adalah: massa, waktu paruh, waktu selang.
- Outputnya adalah massa di akhir percobaan.
- Variabel yang akan digunakan kita rancang sbb:
  - MassaAwal, WaktuParuh, WaktuSelang, MassaAkhir

26 Feb 2001 Pemrograman Komputer 17

(luk@tsipil.ugm.ac.id)

Kasus 1: ... analisis hitungan ...

- Massa awal Polonium = 10 mg, setelah waktu paruh 140 hari, maka massanya tinggal separuhnya = 10*0.5 mg yang tersisa.
- Pada akhir 280 hari, massa yang tersisa (10*0.5)*0.5 atau $10*0.5^2$
- Pada akhir 420 hari, massa yang tersisa $10*0.5^3$
- Rumus umum menjadi:

$$Massa_{sisa} = Massa_{awal} * (0.5)^{\text{waktu selang/waktu paruh}}$$

26 Feb 2001 Pemrograman Komputer 18

(luk@tsipil.ugm.ac.id)

... algoritmanya menjadi ...

1. Masukkan nilai MassaAwal, WaktuParuh, WaktuSelang
2. Hitung:
   MassaAkhir = MassaAwal*0.5**(WaktuSelang /WaktuParuh)
3. Display hasil: MassaAkhir

   Catatan: tiap langkah di atas dieksekusi sekali dan berurutan dari atas ke bawah.

(luk@tsipil.ugm.ac.id)

Kasus 2: Index Polusi - Pilihan

- Input yang dibutuhkan adalah
  - kadar polusi di 3 lokasi,
  - ambang batas polusi.
- Outputnya adalah:
  - index polusi yaitu rerata nilai polusi di 3 lokasi,
  - tingkat polusi.
- Variabel yang akan digunakan kita rancang sbb:
  - Polusi1, Polusi2, Polusi3, AmbangBatas, Index

(luk@tsipil.ugm.ac.id)

Kasus 2: algoritma

1. Masukkan:
   Polusi1, Polusi2, Polusi3, AmbangBatas
2. Hitung:
   Index = (Polusi1 + Polusi2 + Polusi3)/3
3. Jika (Index < AmbangBatas) maka
   a. tulis "Kondisi aman"
   kalau tidak
   b. tulis "Kondisi bahaya"

(luk@tsipil.ugm.ac.id)

Kasus 2: ... algoritma

- Setiap langkah dalam algoritma dieksekusi oleh komputer dari atas ke bawah satu per satu.
- Kecuali pada langkah nomor 3, terdapat pilihan, dimana terdapat 2 pilihan yang mungkin diambil.
- Salah satu langkah tersebut adalah 3.a atau 3.b, hanya salah satu yang akan dieksekusi oleh komputer.

Ir. Djoko Luknanto, M.Sc., Ph.D.
mailto:luk@tsipil.ugm.ac.id

Sampai di sini

(luk@tsipil.ugm.ac.id)

Kasus 3: Waktu gagal

- Input: satu set data yang menunjukkan waktu gagal komponen elektronik.
- Output: jumlah data dan rerata waktu gagal.
- Untuk menyusun algoritma rinci, kita kerjakan dulu dengan kalkulator, agar diperoleh pengertian yang mendalam langkah yang harus diambil.

(luk@tsipil.ugm.ac.id)

Kasus 3: analisis

Data	Kounter	Jumlah
	0	0.0
3.4	1	3.4
4.2	2	7.6
6.0	3	13.6
5.5	4	19.1

- Kounter yang harus selalu ditambah 1 setiap data diproses
- Jumlah merupakan jumlah kumulatif dari setiap data yang diproses

(luk@tsipil.ugm.ac.id)

Kasus 3: ... analisis ...

- Pada saat dikerjakan manual, kita tahu data terakhir yang tersedia, namun jika digunakan komputer harus ada metoda untuk memberitahukan bahwa data telah habis.
- Cara yang biasa digunakan adalah dengan memasukkan "nilai buatan" yang biasa disebut "data flag" atau "data sentinel." Data ini tidak akan diproses hanya untuk menghentikan proses.
- Dirancang variabel yang digunakan:
  - Data, Kounter, Jumlah, Rerata

(luk@tsipil.ugm.ac.id)

Kasus 3: algoritma

1. Set Kounter = 0
2. Set Jumlah = 0.0
3. Masukkan Data pertama
4. While (Data <> Data flag) kerjakan
   a. Tambah Kounter dengan 1
   b. Tambahkan Data kedalam Jumlah
   c. Masukkan Data berikutnya
5. Hitung Rerata = Jumlah/Kounter
6. Display hasil: Rerata dan Kounter

(luk@tsipil.ugm.ac.id)

2.3 Koding Program

- Dua langkah pertama dari pengembangan program sangat penting.
- Jika kedua langkah pertama telah dijalankan dengan cermat, maka langkah ketiga Koding Program biasanya mudah dilaksanakan.
- Koding Program adalah menulis algoritma dengan menggunakan bahasa pemrograman tertentu, misalkan FORTRAN.
- Akan dijelaskan secara singkat bagian-bagian penting dari bahasa FORTRAN.

(luk@tsipil.ugm.ac.id)

Variabel

- Variabel digunakan untuk memberi nama "kuantitas" yang akan diproses. Contoh: MassaAwal, WaktuParuh, WaktuSelang, MassaAkhir.
- Variabel harus diawali dengan huruf dan diikuti dengan huruf atau angka.
- Jumlah huruf dan angka yang terdapat dalam nama variabel biasanya 6 buah, namun compiler FORTRAN saat ini mengijinkan penggunaan lebih dari 6 buah huruf atau angka dalam nama variabel.

(luk@tsipil.ugm.ac.id)

Jenis/tipe variabel

- Setiap variabel sebaiknya didefinisikan terlebih dahulu sebelum digunakan.
- Dua jenis bilangan dikenal dalam FORTRAN yaitu REAL dan INTEGER.
- Pada awal program definisi tersebut ditulis.
- Contoh:

```
REAL MassaAwal, WaktuParuh, WaktuSelang
INTEGER Kounter
```

(luk@tsipil.ugm.ac.id)

Operasi dan Assignment

- Operasi dalam FORTRAN:
  - penambahan +
  - pengurangan –
  - perkalian *
  - pembagian /
  - pangkat **
- Assignment dalam FORTRAN ditandai dengan =
- contoh:
  Rerata = Jumlah/Kounter
- artinya hasil dari Jumlah/Kounter dimasukkan kedalam Rerata.
- atau lebih tepat ditulis sebagai Rerata ← Jumlah/Kounter

(luk@tsipil.ugm.ac.id)

Input/Output

- Dalam algoritma: masukan adalah operasi input, sedangkan display adalah operasi output.
- Untuk input, FORTRAN menggunakan:
  READ *, WaktuParuh, WaktuSelang
- Untuk output, FORTRAN menggunakan:
  PRINT *, 'Index polusi = ', IndexPolusi
  atau
  WRITE (*,*) 'Index polusi = ', IndexPolusi

(luk@tsipil.ugm.ac.id)

Komentar dan Heading

- Dalam FORTRAN, komentar dimulai dengan huruf C atau * pada kolom pertama dari suatu baris. Contoh:
  C Ini adalah komentar
- Heading digunakan untuk memberi nama program. Contoh:
  PROGRAM HitunganPolusi

(luk@tsipil.ugm.ac.id)

Contoh koding Kasus 1

```
C--------------------------------------------
      PROGRAM Peluruhan
C--------------------------------------------
C Program untuk menghitung massa zat radioaktif
C
C Input : MassaAwal, WaktuParuh, WaktuSelang
C Output: MassaAkhir
C--------------------------------------------
      REAL MassaAwal, WaktuParuh, WaktuSelang, MassaAkhir

      PRINT *, 'Masukkan massa, waktu paruh, dt = '
      READ *, MassaAwal, WaktuParuh, WaktuSelang
      MassaAkhir = MassaAwal*0.5**(WaktuSelang/WaktuParuh)
      PRINT*, 'Massa akhir = ', MassaAkhir

      STOP 'Selesai bozz'
      END
```


(luk@tsipil.ugm.ac.id)

2.4 Eksekusi dan Testing

- Kesalahan mungkin terjadi pada saat proses pembuatan perangkat lunak.
- Deteksi error dan pembetulan merupakan bagian dari proses yang biasa disebut validasi dan verifikasi.
- Validasi: meneliti apakah algoritma sesuai dengan spesifikasi yang dikehendaki.
- Verifikasi: meneliti apakah algoritma benar dan lengkap.

(luk@tsipil.ugm.ac.id)

Validasi dan verifikasi

- Validasi menjawab pertanyaan:
  Apakah kita menjawab permasalahan yang benar?
- Verifikasi menjawab pertanyaan:
  Apakah kita menjawab permasalahan dengan benar?
- Langkah penting untuk validasi dan verifikasi adalah eksekusi program dengan menggunakan beberapa set data untuk tes ini.

(luk@tsipil.ugm.ac.id)

2.5 Rekayasa perangkat lunak

- Siklus pembuatan suatu perangkat lunak melalui 5 tahap:
  1. Analisis masalah dan spesifikasi
  2. Perancangan
  3. Koding
  4. Eksekusi dan testing
  5. Pemeliharaan.

26 Feb 2001 Pemrograman Komputer 37

(luk@tsipil.ugm.ac.id)

2.5.1 Analisis masalah dan spesifikasi

- Sebagai contoh digunakan "proyek pembangunan bangunan sipil."
- Agar langkah ini berhasil, beberapa hal umum yang harus dijawab:
  - informasi apa yang tersedia,
  - bagaimana data tersebut diakses oleh program,
  - apakah data sudah divalidasi, atau harus dicek langsung oleh program,
  - dalam format yang bagaimana hasil harus disajikan,
  - apakah laporan harus dibuat untuk eksekutif perusahaan, dewan lingkungan, pemda, dlsb.

26 Feb 2001 Pemrograman Komputer 38

(luk@tsipil.ugm.ac.id)

2.5.1 Analisis masalah ...

- Beberapa hal (yang memerlukan proses rinci) yang harus dijawab:
  - bahan-bahan apa yang dibutuhkan,
  - peralatan apa yang akan digunakan,
  - apakah pegawai digaji jam-jaman atau bulanan atau kombinasi,
  - bagaimana gaji untuk lembur,
  - bagaimana dengan pembayaran pajak dlsb.

26 Feb 2001 Pemrograman Komputer 39

(luk@tsipil.ugm.ac.id)

2.5.1 Analisis masalah ...

- Beberapa hal (yang berkenaan dengan pemrograman) yang harus dijawab:
  - apakah program digunakan oleh pengguna yang sudah canggih,
  - ataukah program harus user-friendly agar penguna awam tidak mengalami kesulitan,
  - seberapa sering program digunakan,
  - berapa "response time" yang diharapkan,
  - berapa umur program,
  - perangkat lunak dan keras yang telah tersedia.
- Kadang diperlukan seorang "analis sistem" untuk melakukan langkah ini.

26 Feb 2001 Pemrograman Komputer 40

(luk@tsipil.ugm.ac.id)

2.5.2 Perancangan

- Gunakan cara "divide and conquer" yaitu memecah masalah menjadi:
  - bagian-bagian kecil,
  - yang lebih sederhana,
  - dan independen,
  - kemudian selesaikan masing-masing bagian.
- Dimulai dengan mengidentifikasi tugas-tugas utama.

26 Feb 2001 Pemrograman Komputer 41

Ir. Djoko Luknanto, M.Sc., Ph.D.
mailto:luk@tsipil.ugm.ac.id

Sampai di sini

26 Feb 2001 Pemrograman Komputer 42

(luk@tsipil.ugm.ac.id)

2.5.2 ... diagram permasalahan

- Diagram struktur
  - Masalah utama

(luk@tsipil.ugm.ac.id)

2.5.2 ... pembagian lebih rinci ...

- Biasanya satu atau lebih dari tugas-tugas di depan masih harus dinyatakan secara lebih rinci.
- Misalkan spesifikasi proyek dapat dirinci menjadi:
  - dapatkan kebutuhan SDM,
  - dapatkan kebutuhan peralatan,
  - dapatkan kebutuhan material.
- Demikian pula dengan hitungan; dapat dirinci menjadi:
  - hitung biaya SDM,
  - hitung biaya peralatan,
  - hitung biaya material.

(luk@tsipil.ugm.ac.id)

2.5.2 ...rincian ...

- Demikian masing-masing tugas dapat dirinci menjadi modul-modul kecil yang seolah-olah bebas satu sama lain.
- Modul inilah yang akan dikoding oleh pemrogram yang berbeda.
- Kemudian modul-modul bebas ini diintegrasikan menjadi satu perangkat lunak.

(luk@tsipil.ugm.ac.id)

2.5.3 Koding ...

- Koding adalah implementasi algoritma modul kedalam salah satu bahasa pemrograman.
- Koding akan menjadi bagus jika:
  - benar, mudah dibaca, mudah dimengerti,
  - struktur program jelas.

(luk@tsipil.ugm.ac.id)

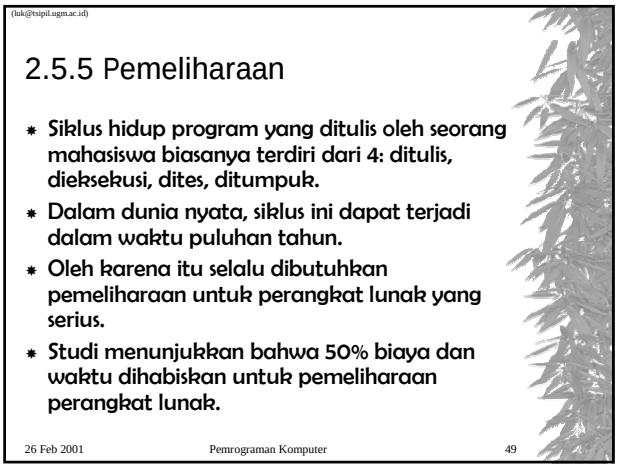
2.5.3 ... koding

- Setiap koding harus didokumentasi:
  - setiap unit harus ada penjelasan pendahuluan,
  - komentar harus diberikan pada blok-blok penting,
  - nama variabel dipilih yang mempunyai arti
- Tingkat terbaca harus tinggi:
  - gunakan spasi dimana perlu,
  - tambahkan baris kosong jika dibutuhkan,
  - gunakan indentasi pada blok-blok yang membutuhkan.

(luk@tsipil.ugm.ac.id)

2.5.4 Eksekusi dan testing

- Jelas bahwa suatu perangkat lunak harus benar.
- Bagaimanapun bagus organisasinya dan terdokumentasikannya suatu program, tanpa menyajikan hasil yang benar, maka tidak ada gunanya.
- Program yang berjalan tanpa “error” belum tentu benar.
- Diperlukan eksekusi dan testing.


(luk@tsipil.ugm.ac.id)
2.5.5 Pemeliharaan
Siklus hidup program yang ditulis oleh seorang mahasiswa biasanya terdiri dari 4: ditulis, dieksekusi, dites, ditumpuk.
Dalam dunia nyata, siklus ini dapat terjadi dalam waktu puluhan tahun.
Oleh karena itu selalu dibutuhkan pemeliharaan untuk perangkat lunak yang serius.
Studi menunjukkan bahwa 50% biaya dan waktu dihabiskan untuk pemeliharaan perangkat lunak.
26 Feb 2001
Pemrograman Komputer
49