Programming Best Practices

Pelatnas I TOKI di ITB

Oleh : Inggriani Liem

Oktober2014

Tujuan

- Dalam melakukan pemecahan masalah, koding dan debugging, siswa melakukan dengan waktu yang singkat untuk hasil yang efektif.
- Membentuk “kebiasaan” koding yang cepat tanpa salah.

Proses Pemecahan persoalan

- Memahami persoalan, dari cerita menarik makna deskripsi ringkas persoalan yang harus diprogram
- Menganalisis, membuat spesifikasi (I-P-O)
- Merancang sketsa solusi, “Design” algoritma
- Koding dalam salah satu bahasa pemrograman
- Debugging dan testing [dengan test case], test case sesuai spesifikasi

Tips

- Memikirkan dengan baik solusinya sebelum koding, akan menghindari koding yang berkali-kali. Analogi : Pikirkan “target”, buat strategi, baru “menembak”
- Ukur kompilasi : dua kali kompilasi harus benar

Pengetikan program

- Perhatikan indentasi
- Ketik dengan cermat (prinsip Yin-Yang)
- Incremental coding
- Satuan ukuran teks
- Komentar
- Kebiasaan meninggalkan pekerjaan dan kompilasi, selalu menyimpan program yang benar
- Hasilkan “clean Code”

Program Yang baik

- Benar
  - sesuai dengan yang diharapkan untuk pemecahan masalah
- Readable
  - Jika bermasalah dapat dibaca dengan mudah, mempermudah diagnosa/debugging
  - Kriteria : indentation, variable name, .. Coding rules
- Robust
  - Tahan banting, tidak gagal untuk data set apapun yang sesuai spesifikasi
- Untuk Lomba : sesuai dengan batasan time & memry limit

Program Bermasalah

- Duplicate code (harus dibuat prosedur)
- Literal bertebaran, kalau harus diubah banyak usaha dan resiko kelupaan
- IF – kondisional :
  - kasus “bocor”, tidak tertangani
  - Terlalu “nested” – dapat disederhanakan dengan membuat predikat
- Loop :
  - Tidak efisien
  - Tidak jelas traversal atau search
  - Peran variabel pengontrol/proses tidak jelas
  - Proses akumulasi, counting tidak jelas

Variabel

- Nama variabel sesuai “kebiasaan” sehingga memberikan konotasi tunggal
- Variabel global vs lokal
- Inisialisasi variabel yang perlu, dengan inisialisasi (bukan instruksi)

Prosedur - Fungsi

- Prosedur atau sekumpulan teks yang sebetulnya lebih tepat disebut prosedur namun krn alasan efisiensi:harus jelas namanya dan apa yang dikerjakan, tidak tumpang tindih
- Fungsi : sebaiknya satu “return” di akhir; jika terpaksa lebih dari satu lakukan analisis kasus dengan baik
- Tentukan apakah prosedur atau fungsi sesuai dengan ketentuan

Predikat

- Predikat adalah fungsi dengan kembalian bertype boolean (hasilnya true atau false)
- Berguna untuk mengisolasi kode yang potensi berubah aturan. Misalnya buat predikat IsSeparator (CC) kalau karakter yang dianggap sebagai separator bisa berubah
- Boolean : tentukan nilai FALSE adalah 0 dan TRUE adalah satu

Keterlaluan jika masih…..

- Salah sintaks karena teks tidak “balanced” (tidak yin-yang) semacam masih ada kurung tidak berpasangan
- Ada variable atau nama tidak dideklarasi shg mengakibatkan salah sintaks
- salah kalkulasi akibat operator presedence tidak seperti yang “kupikirkan”
- Salah kalkulasi/meleset karena ekspresi yg tidak ketat type
- Pembulatan: saya tidak menentukan sendiri “floor”, ceiling atau pembulatan

Latihan

- Terjemahkan dengan cepat, beberapa program bahasa pascal yang pernah anda buat, dengan mengacu ke padanan yang diberikan dalam slides ini dan spreadsheet padanan pascal bahasa C
- Terjemahkan beberapa instruksi dalam bahasa Algoritmik yang diberikan