

MEMBANGUN MODUL PROGRAM YANG EFEKTIF

Oleh : Rahmat Robi Waliyansyah, M.Kom.

PENDAHULUAN

- Manusia bila menghadapi permasalahan yang (cukup) besar langkah awal yang dilakukan adalah **membagi/memecah permasalahan tersebut menjadi beberapa bagian yang lebih kecil.**
- Selanjutnya **menyelesaikan secara bertahap bagian demi bagian**, baik secara sendiri maupun berkelompok, sehingga diperoleh solusi dari permasalahan yang besar tersebut.
- Sebagai gambaran, kita diminta untuk “menyelesaikan” (baca: memakan) buah semangka sampai habis.
- Langkah awal yang dilakukan adalah “**mempartisi**” (memotong) buah semangka menjadi beberapa bagian dan memakannya satu persatu, sendiri atau bersama-sama.

PENDAHULUAN

- Dalam lingkup pengembangan perangkat lunak (program) analogi di atas harus dipakai, apabila akan menghasilkan perangkat lunak yang baik.
- Program yang besar harus **dipartisi menjadi beberapa modul** yang mudah diselesaikan.
- Modul program dapat berupa **prosedur** dan/atau **fungsi**.

PENDAHULUAN

- Untuk mempartisi program menjadi beberapa bagian terdapat dua paradigma pokok, yaitu **Structured Programming** (SP) dan **Object Oriented Programming** (OOP).
- SP mempartisi program berdasarkan **kata kerja** (fungsi sistem) atau **behavior** sedangkan OOP mempartisi program berdasarkan **kata benda** (objek diskrit).
- Dalam SP, struktur data dan behavior **terpisah**, di sisi lain OOP **mengkapsulasi** struktur data dan behavior **dalam satu objek**.

MENGUKUR MODUL PROGRAM YANG BAIK

- Reusable adalah **kunci pokok dalam pengembangan perangkat lunak**, tema inilah yang mengilhami perancangan modul program dan perkembangan paradigma pengembangan perangkat lunak secara umum.
- Reusable bisa diperoleh bila menerapkan “**information hiding**”.

MENGUKUR MODUL PROGRAM YANG BAIK

Yang dimaksud Information hiding adalah: [Presman 2000] :

- Setiap modul **tersembunyi** dengan yang lain.
- Modul harus dirancang agar informasi (prosedur dan data) yang berada di dalam modul **tidak dapat diakses oleh oleh modul lain yang tidak memerlukan informasi tersebut.**

MENGUKUR MODUL PROGRAM YANG BAIK

Keuntungan modul yang efektif :

- Mengurangi kompleksitas
- Mempermudah perubahan
- Lebih mudah diimplementasikan dan dapat dikerjakan secara paralel (Tim)

MENGUKUR MODUL PROGRAM YANG BAIK

Dua Mekanisme pengaktifan modul:

- Diinvokasi by refference (**statemen call**)
- Diinvokasi oleh interupsi (**sistem real-time**)

MENGUKUR MODUL PROGRAM YANG BAIK

- Functional Independence merupakan kunci perancangan yang baik dan kunci kualitas program.
- Keuntungan Modul yang independen:
 1. Mudah membagi dalam tim
 2. Mudah diubah
 3. Perambatan kesalahan berkurang
 4. Reusable bertambah

MENGUKUR MODUL PROGRAM YANG BAIK

Independensi diukur dengan dengan dua kriteria kualitatif, yaitu **Cohesion** dan **Coupling**.

- ▶ **COHESION**: mengukur **kekuatan fungsional** relatif suatu modul.
- ▶ **COUPLING**: mengukur **interdependensi** relatif antar modul.

MENGUKUR MODUL PROGRAM YANG BAIK

Ciri Modul yang baik:

- **High cohesion (functional cohesion):** modul hanya melakukan satu tugas dan memerlukan sedikit interaksi dengan modul lain dalam satu program.
- **Low coupling:** modul memiliki kopling antar modul yang lemah atau sebebas mungkin dengan modul yang lain (independen). Kopling tergantung pada kompleksitas antarmuka modul.

STUDI KASUS

- ▶ Pada halaman berikut diberikan contoh kasus penyelesaian program secara modular. Pada kolom sebelah kiri, program sudah terdiri dari beberapa modul tetapi masih memiliki kopling tinggi. Kolom sebelah kanan merupakan hasil perbaikan yang menunjukkan kopling rendah. Kohesi tidak begitu tampak dalam kasus ini.

STUDI KASUS

BROWSE

KESIMPULAN

- Dalam menyusun program tidak sekedar memecah program menjadi beberapa modul, tetapi harus memperhatikan bagaimana kualitas modul tersebut.
- Modul yang baik adalah reusable.
- Reusable dalam program yang sama maupun reusable untuk program yang lain.