



Pertemuan #13

Desain Program



Pengertian

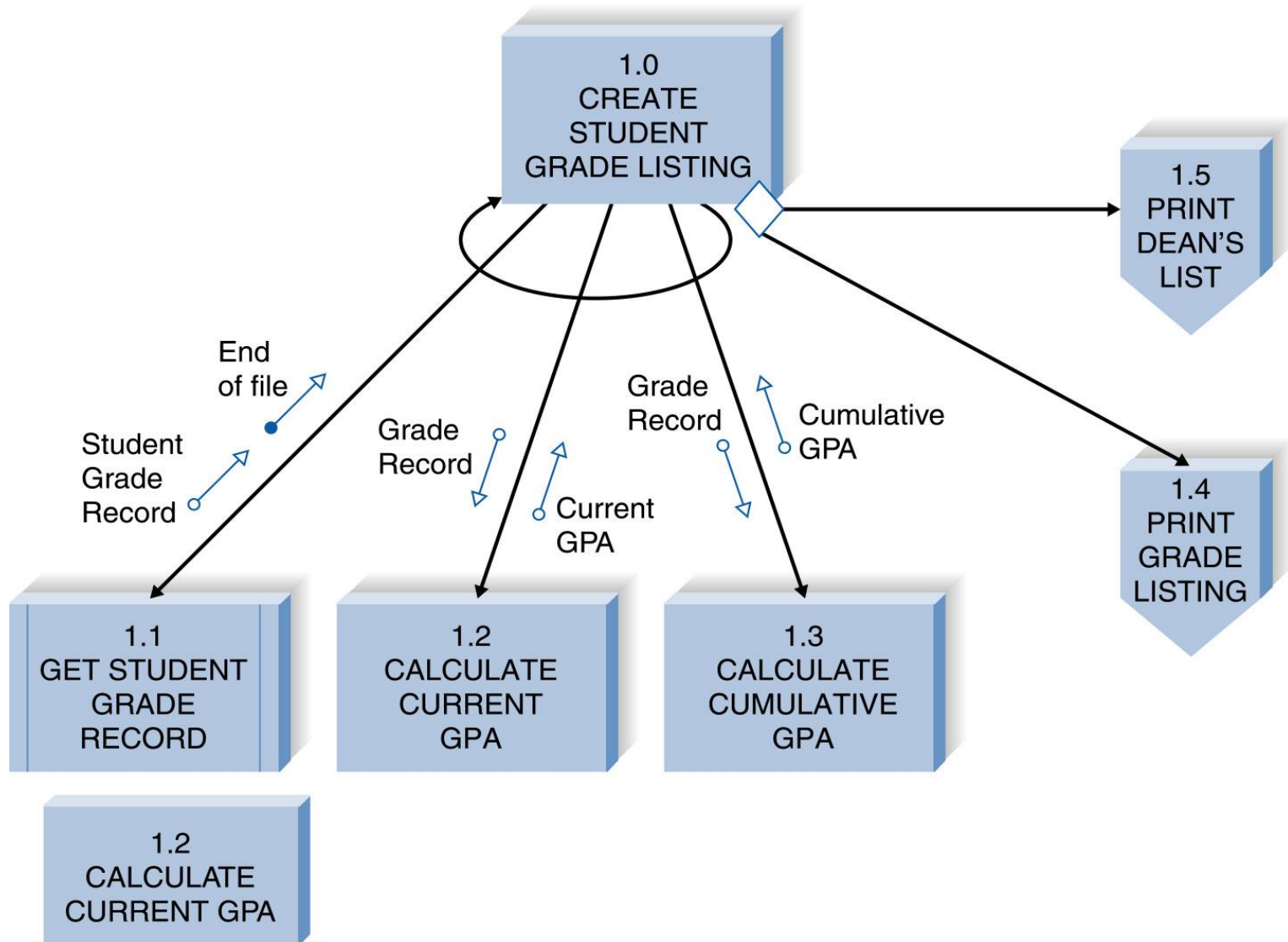
- **Desain program** - membuat instruksi untuk programmer
- **Pendekatan modular dari atas ke bawah** - dimulai dengan "gambaran besar" dan secara bertahap menambahkan detail
- **Dokumen desain program** - semua bagan struktur dan spesifikasi yang dibutuhkan oleh programmer untuk mengimplementasikan sistem



Structure Chart

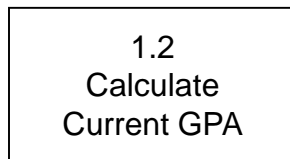
- **Structure Chart** merupakan teknik desain program yang sering digunakan
- Menampilkan semua komponen kode dalam format hierarkis
 - Urutan (sequence)
 - Pilihan (branching)
 - Pengulangan (looping)

Contoh Structure Chart

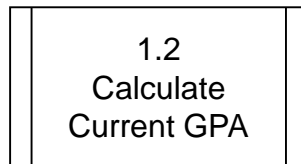


Element Structure Chart

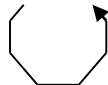
Module



Library Module



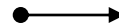
Loop



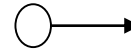
Conditional Line



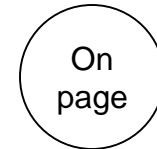
Control Couple



Data Couple



Off
Page



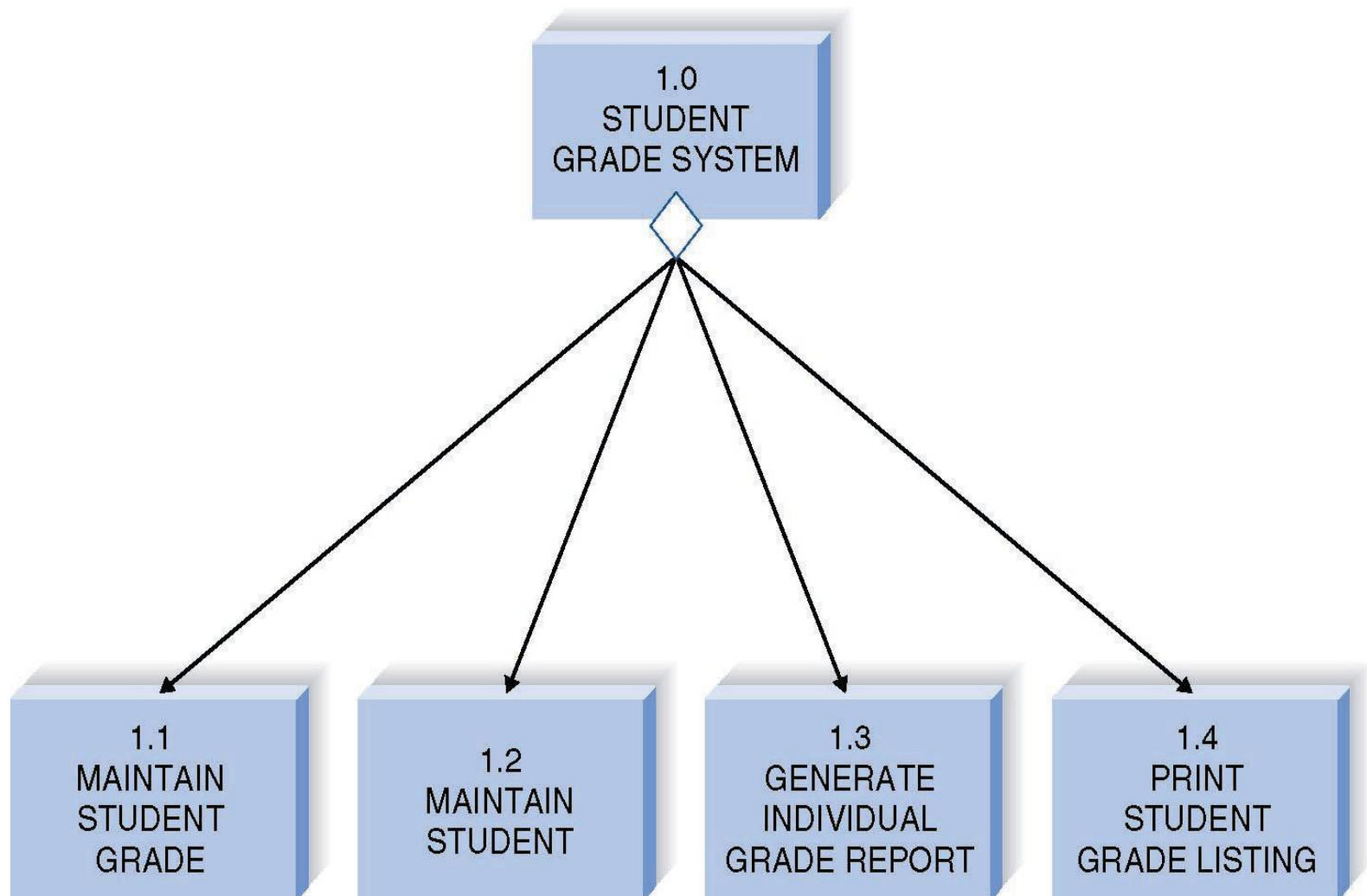
Pembuatan Structure Chart

- **Satu proses dalam DFD cenderung mewakili satu modul pada structure chart**
 - Proses afferent - memberikan input ke sistem
 - Proses central - melakukan operasi sistem
 - Proses efferent - menangani keluaran sistem
- **Leveling pada DFD dapat berkorespondensi dengan hierarki pada structure chart**

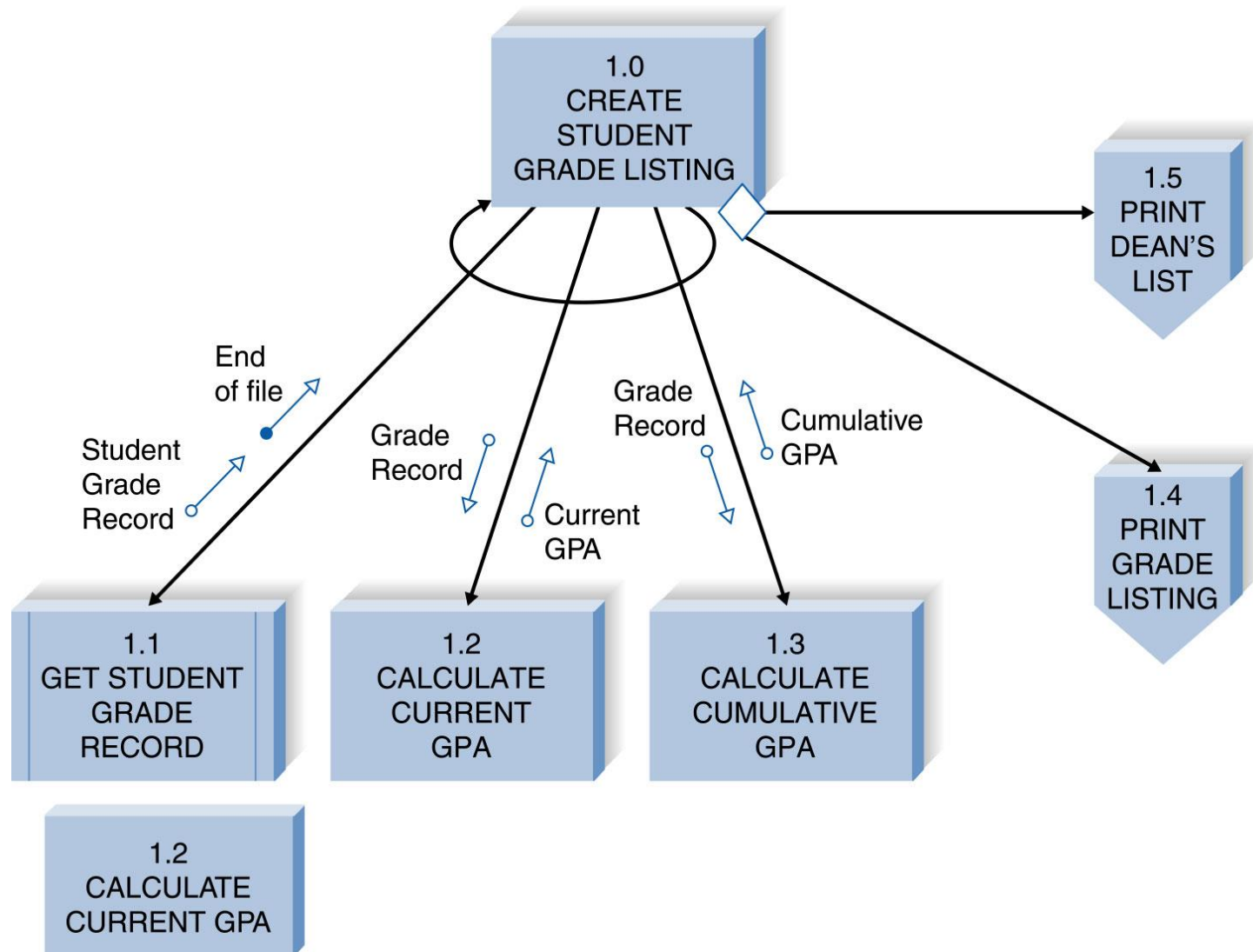
Tipe Structure Chart

- **Transaction structure** - modul kontrol memanggil modul bawahan, yang masing-masing menangani transaksi tertentu
 - Banyak proses afferent
 - Sedikit proses efferent
 - Berada pada tingkat atas pada structure chart
 - Menggunakan input untuk membuat output baru
- **Transform structure** - modul kontrol memanggil beberapa modul bawahan secara berurutan
 - Setiap bawahan melakukan langkah dalam proses yang mengubah input menjadi output
 - Sedikit proses afferent
 - Banyak proses efferent
 - Berada pada tingkat yang lebih rendah pada structure chart

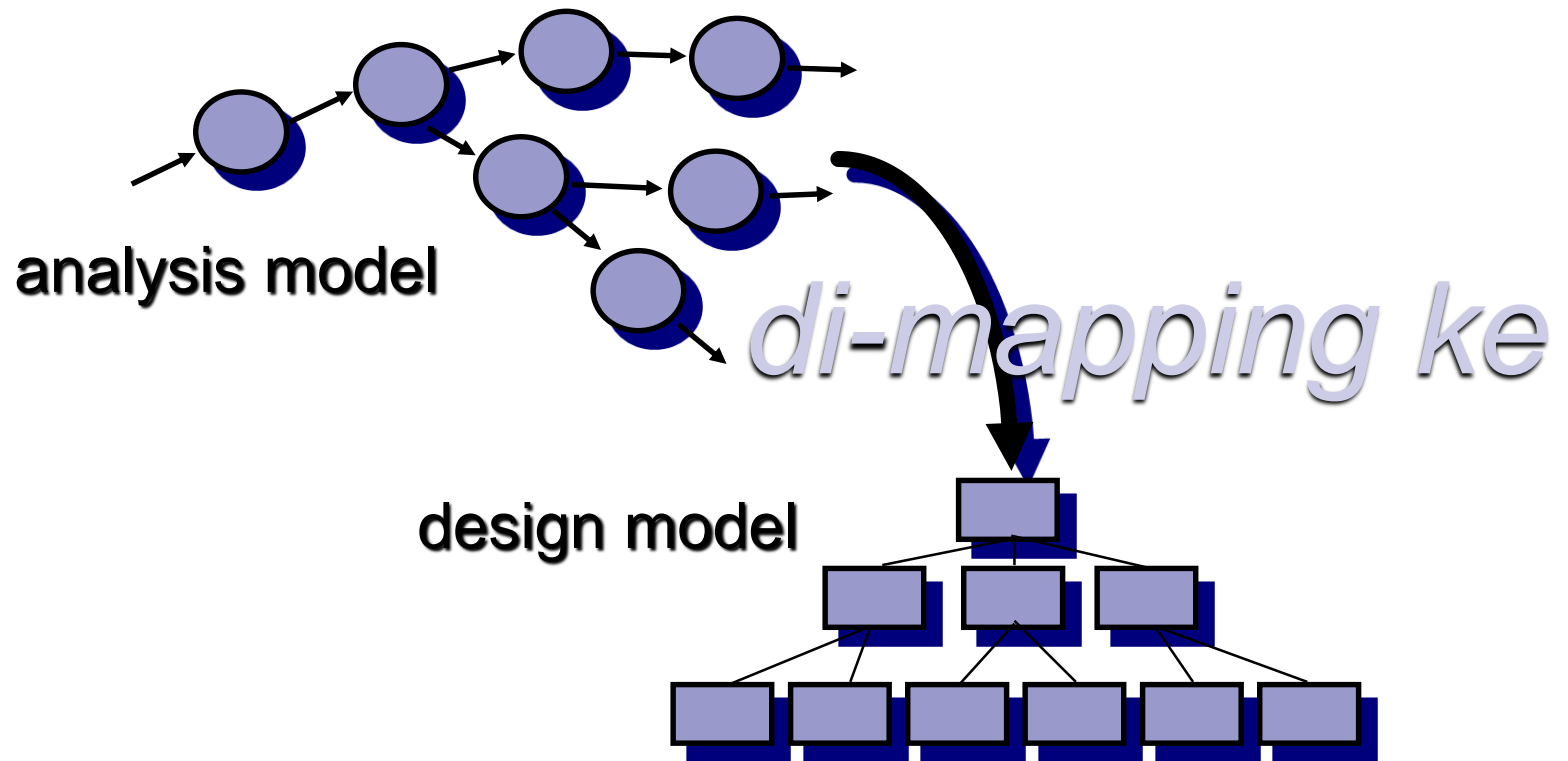
Transaction Structure Chart



Transform Structure Chart



Mapping DFD ke Structure Chart



Tahapan Mapping DFD ke SC

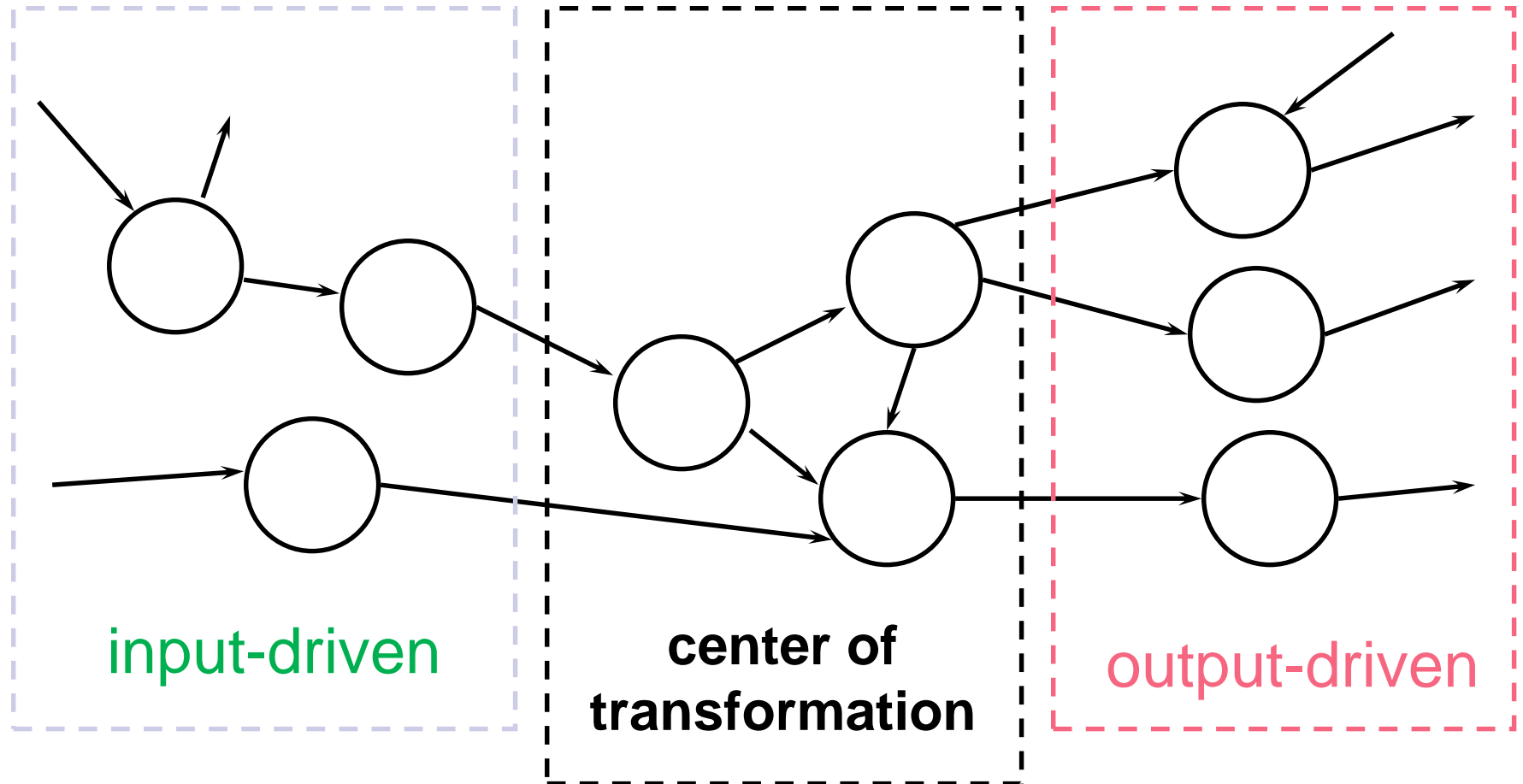
- Tentukan aliran informasi pada DFD di sistem
- Identifikasi/Analisis pola struktural yang khas di DFD
- Gunakan metode heuristik untuk memetakan DFD ke dalam Structure Chart → Firstcut Structure Chart
- Periksa kembali dan perbaiki Structure Chart untuk mengoptimalkan hasil Structure Chart (dilakukan secara terus menerus) → Final Structure Chart



Dua Pola Mapping

- Transaction Flow Analysis
- Transform Flow Analysis

Transform Flow

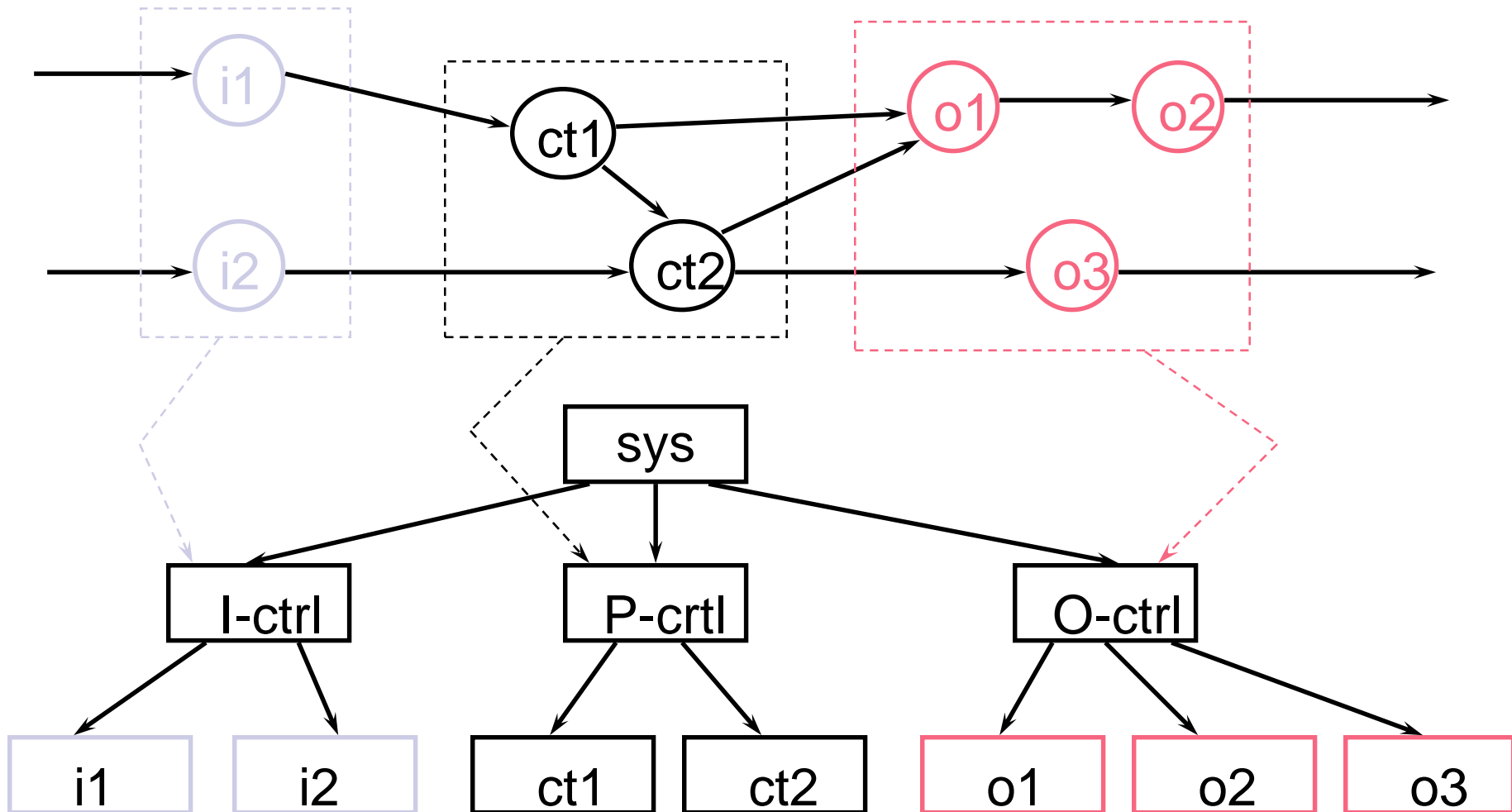




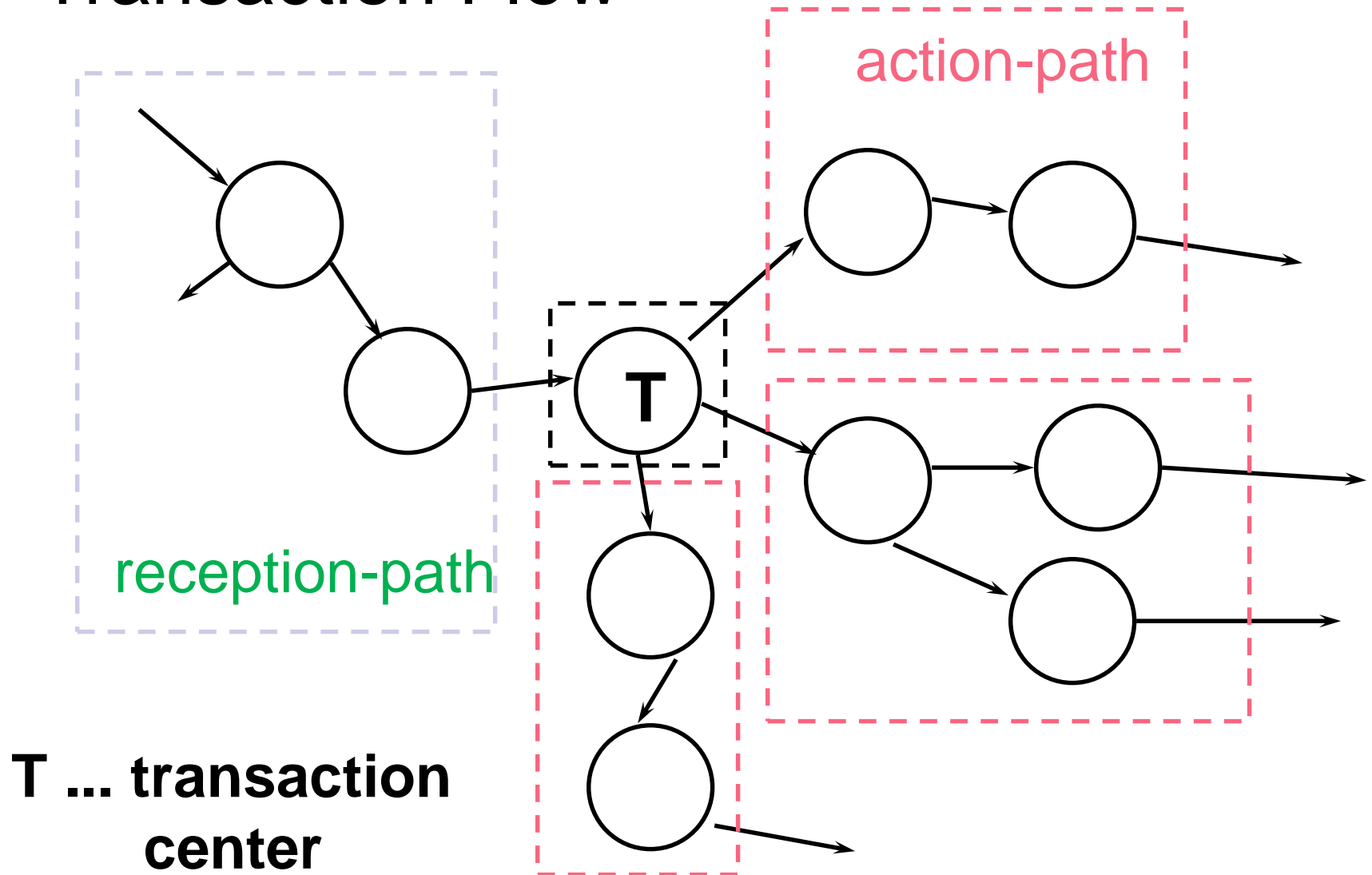
Karakteristik Transform Flow

- sistem memiliki tujuan tunggal yang koheren
- pusat transformasi (center of transformation) mengeksekusi algoritma, transformasi data, manipulasi database, ...
- proses input-driven menyaring, memeriksa dan menerjemahkan aliran data eksternal
- proses output-driven memformat hasil untuk presentasi ke lingkungan (pengguna)
- beberapa jalur untuk mendapatkan input

Mapping Transform secara Heuristic



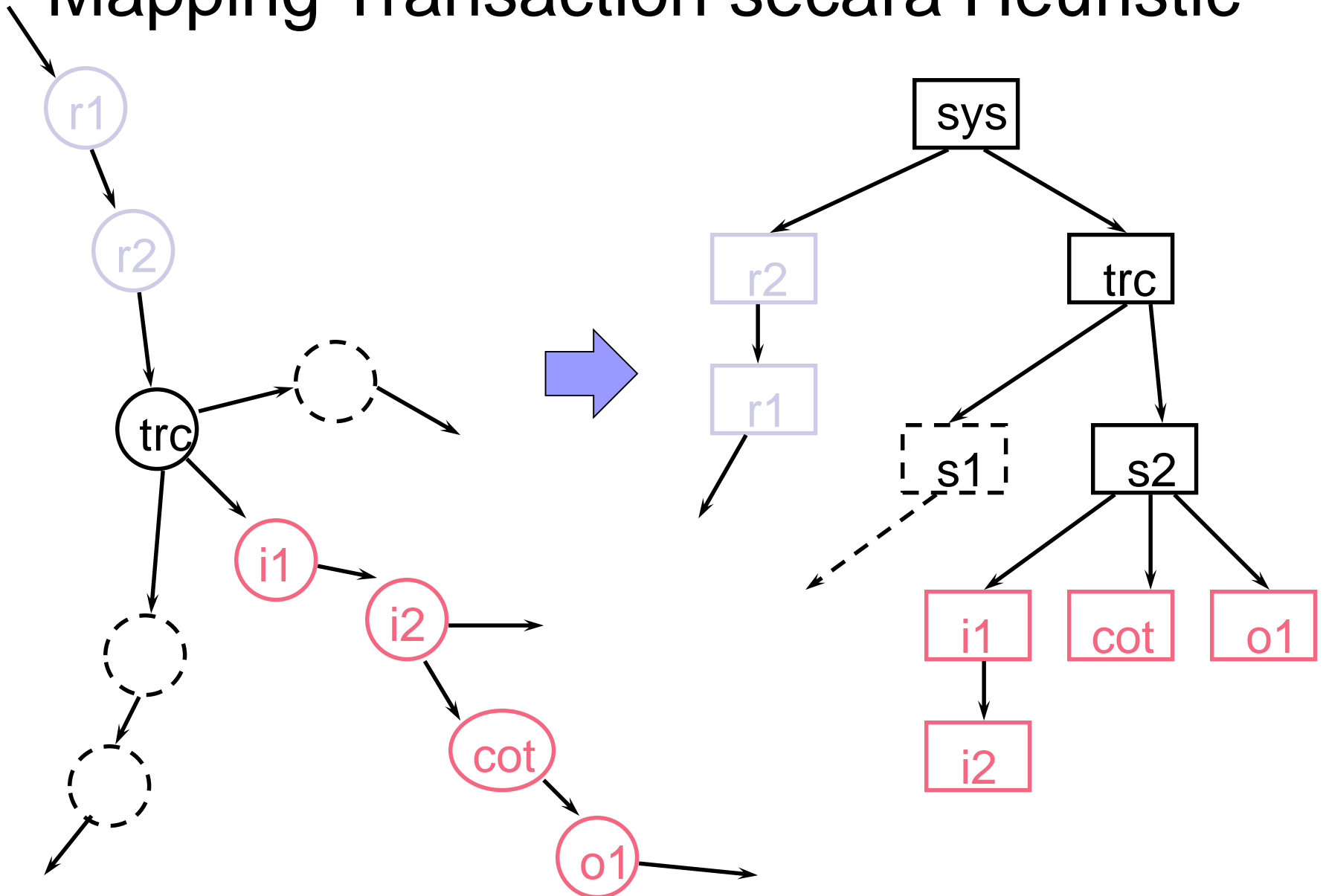
Transaction Flow



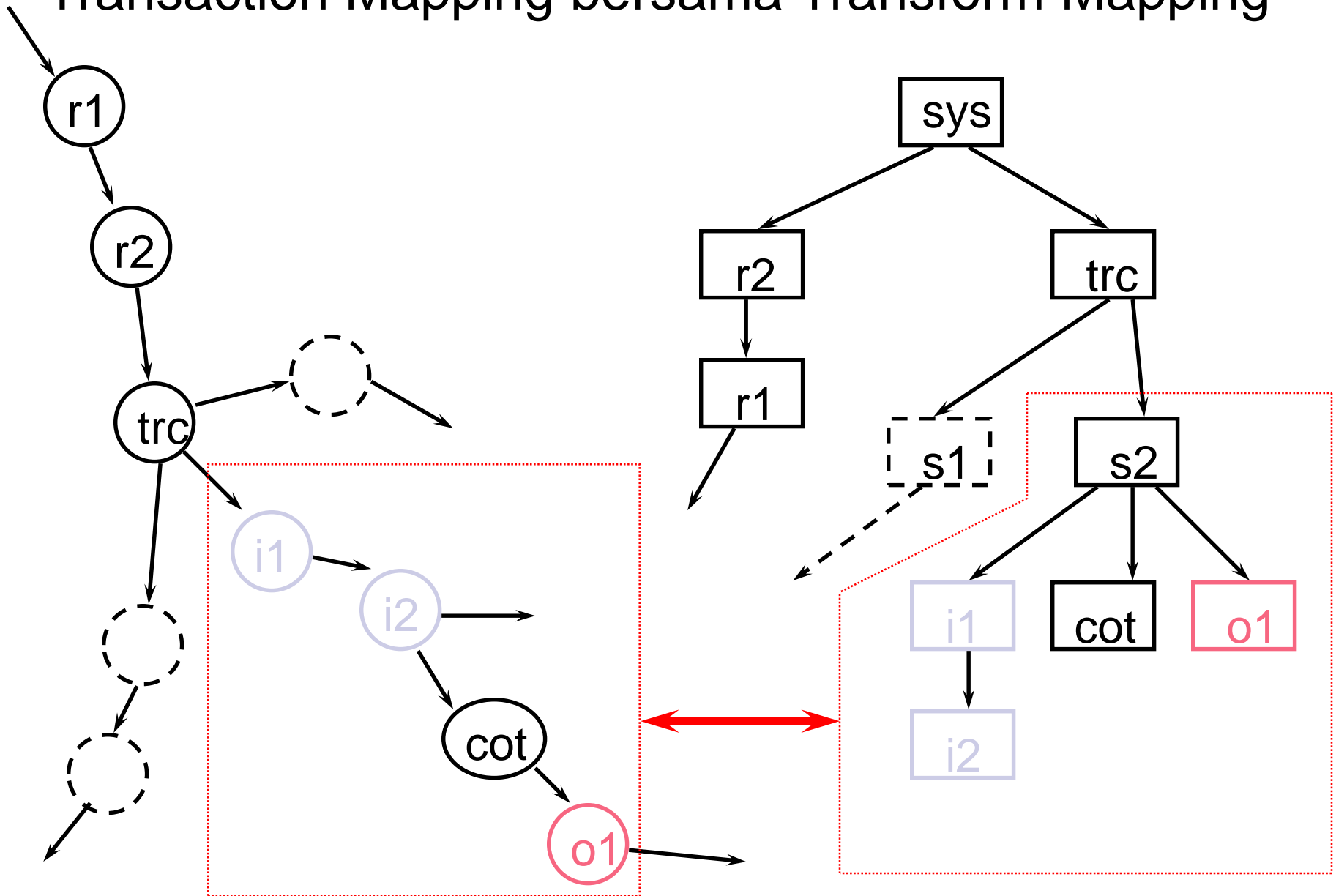
Karakteristik Transaction Flow

- satu jalur proses penerimaan (*reception path*)
- transaksi: item data tunggal yang mencakup semua informasi yang diperlukan untuk eksekusi
- pusat transaksi (*transaction center*) mengevaluasi transaksi & menginisialisasi jalur-aksi yang benar
- action-paths mengimplementasi berbagai jenis fungsionalitas sistem
- sebuah action-paths bisa merupakan sub-sistem lengkap dengan karakteristik transform flow

Mapping Transaction secara Heuristic



Transaction Mapping bersama Transform Mapping



Optimalisasi Structure Chart

- Pembuatan structure chart dilakukan secara heuristic (perbaikan secara berkelanjutan) sampai didapatkan structure chart yang optimal
- Ciri structure chart berkualitas tinggi:
 - ☐ menghasilkan program yang modular
 - ☐ dapat digunakan kembali
 - ☐ mudah diimplementasikan
- Tindakan yang bisa dilakukan:
 - ☐ Meningkatkan kohesi dalam modul
 - ☐ Menurunkan kopling antar modul
 - ☐ Level fan-in dan fan-out yang sesuai

Tipe Kohesi

- **Functional**: dikelompokkan karena menjalankan satu fungsi
- **Sequential**: dikelompokkan karena dijalankan secara berurutan
- **Communicational**: dikelompokkan karena mengelola data yang sama
- **Procedural**: output dari satu bagian menjadi input di bagian lainnya
- **Temporal**: karena kesamaan waktu proses
- **Logical**: dikelompokkan karena secara logika melakukan hal yang sama
- **Coincidental**: setiap bagian tidak mempunyai hubungan yang penting

Baik

Buruk

Tipe Kopling

- **Data:** antar modul berbagi data, misalnya melalui parameter
- **Stamp:** modul berbagi struktur data komposit dan hanya menggunakan bagian-bagiannya
- **Control:** satu modul mengendalikan aliran modul lainnya
- **Common:** beberapa modul memiliki akses ke data global (common) yang sama
- **Content:** satu modul menggunakan kode modul lain

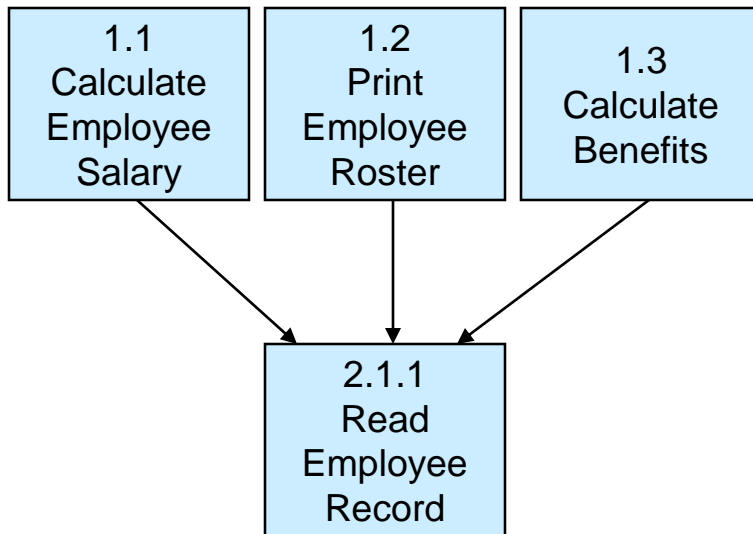
Baik



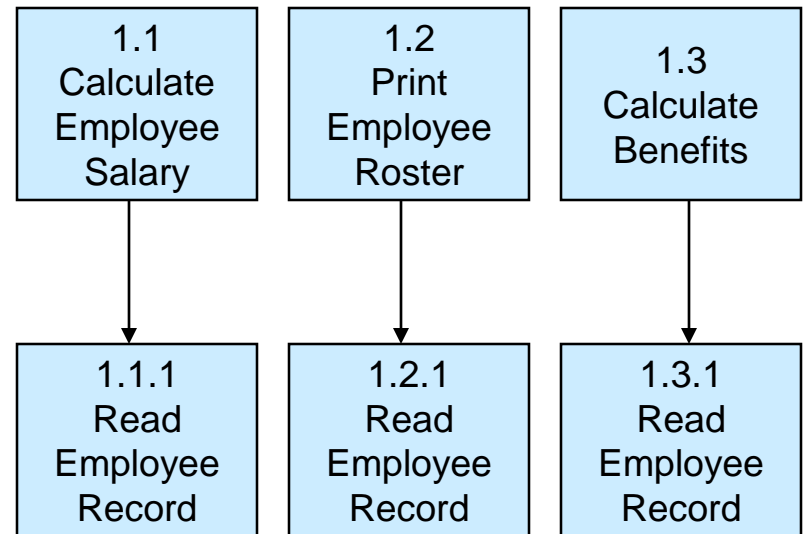
Buruk

Fan-in dan Fan-out

- **Fan-in:** jumlah modul yang masuk ke suatu modul
- **Fan-out:** jumlah modul yang keluar dari suatu modul
- Fan-in yang tinggi lebih baik dibanding Fan-in yang rendah



Fan-in tinggi,
penggunaan kembali modul
sebelumnya tinggi



Fan-in rendah,
kurang baik