

PENGGUNAAN KEMBALI (REUSE) PERANGKAT LUNAK

I.MASALAH-MASALAH MANAJEMEN

A. KENDALA UNTUK REUSE

1. Beberapa perusahaan dan organisasi memiliki sesuatu yang sangat mirip dengan rencana reusabilitas per.lunak
2. Meski semakin banyak vendor perangkat lunak yg menjual peranti atau memberi bantuan terhadap reuse, pengembang tidak menggunakannya
3. Tersedia latihan yg sedikit untuk membantu perekayasa dan manajer memahami dan mengaplikasikan reuse
4. Banyak praktisi percaya bahwa reuse lebih merupakan “masalah”

5. Banyak perusahaan menganjurkan metodologi pengembangan perangkat lunak tidak memfasilitasi reuse
6. Beberapa perusahaan memberi insentif untuk menghasilkan reuse. Kenyataannya ada disinsentif

B. ANALOGI PERANGKAT KERAS

Agar benar-benar reusable, komponen perangkat lunak harus dapat dipakai di beberapa tempat lain. Maka perancang hrs membuat komponen lebih umum. Peningkatan generalitas dapat menurunkan efisiensi, atau bahkan meningkatkan kompleksitas.

C. SARAN-SARAN UNTUK MEMBANGUN PENDEKATAN REUSE

Langkah-langkah berikut akan membantu:

1. Membangun rencana reuse internal

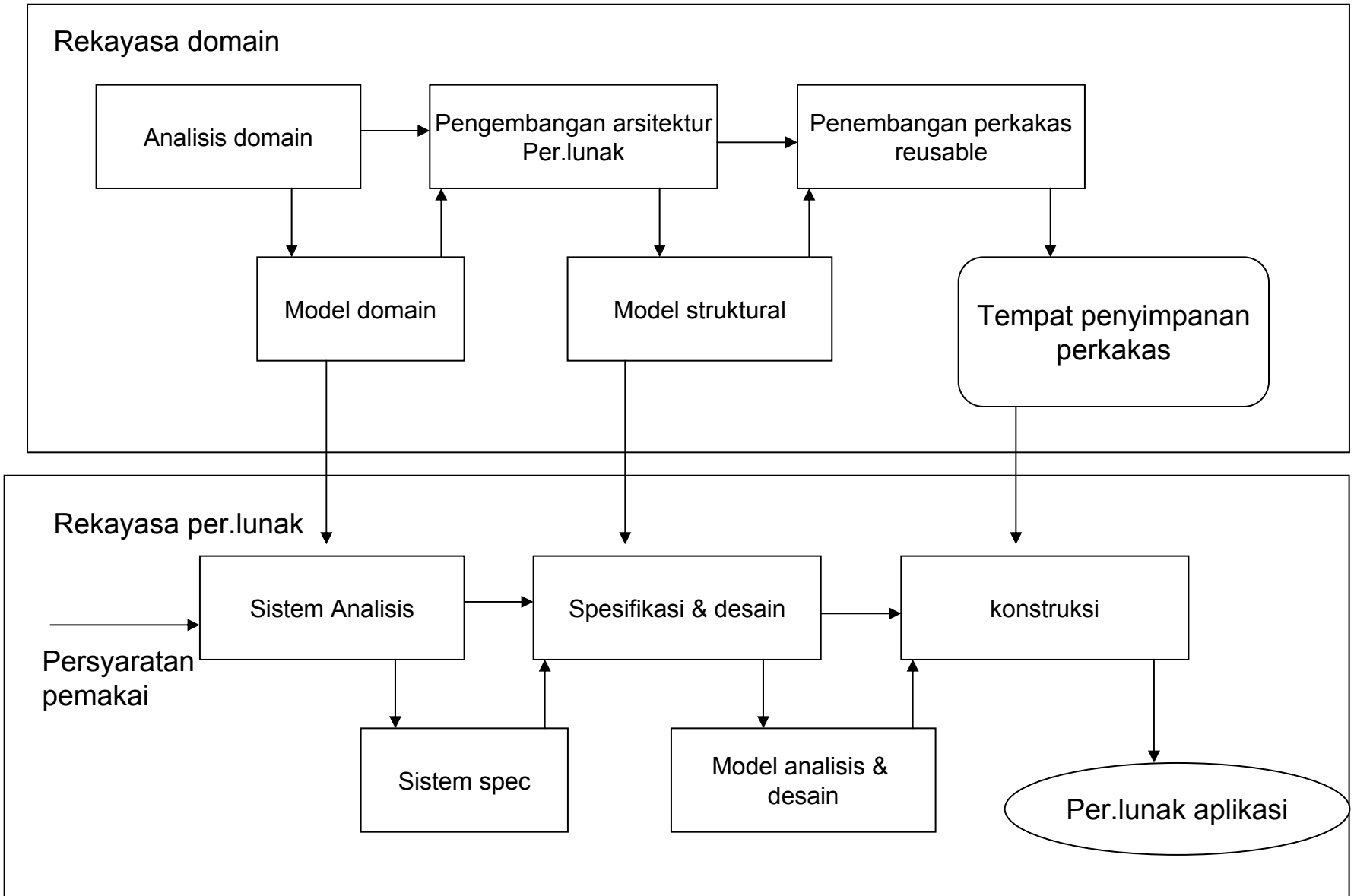
2. Mengharuskan reusabilitas menjadi bagian internal dari pelatihan manajerial dan teknik.
3. Dapatkan peranti dan perpustakaan yang dapat memberi kontribusi positif bagi reuse
4. Dukung metode dan piranti yang telah didemonstrasikan
5. Telusuri dan ukur reuse serta pengaruhnya
6. Manajemen harus mendukung dan aktif
7. Kenali peranti, data test, desain, rencana, lingkungan dan jenis per.lunak yang dapat di reuse
8. Reuse bukan merupakan “ bisnis biasa”

II. PROSES REUSE

A. PERKAKAS REUSABLE

- a. Rencana proyek
- b. Perkiraan biaya
- c. Arsitektur
- d. Model dan spesifikasi persyaratan
- e. Desain
- f. Kode sumber
- g. Dokumentasi teknis dan pemakai
- h. Interface manusia
- i. Data
- j. Test case

B. Model proses



III. Rekayasa Domain

Tujuan : mengidentifikasi, mengkonstruksi, meng-katalog dan menyebarkan serangkaian perkakas perangkat lunak yang memiliki aplikabilitas dan perangkat lunak masa depan didalam suatu domain aplikasi tertentu

Rekayasa domain meliputi 3 aktifitas,yaitu analisis, konstruksi dan penyebaran.

A. Proses Analisis Domain

Langkah-langkah proses itu didefinisikan sebagai:

1. Tentukan domain yang akan diselidiki.
2. Kategorikan item yang akan di-ekstrak dari domain.

3. Kumpulkan sampel yang presentatif dari aplikasi didalam domain.
4. Analisis masing-masing aplikasi didalam sampel.
5. Kembangkan sebuah analisis bagi objek.

B. Fungsi-Fungsi Karakterisasi

serangkaian karakteristik domain untuk suatu perkakas reusable dapat direpresentasikan sebagai $\{D_p\}$ dimana

D_{pi} , merepresentasikan karakteristik domain spesifik ;

nilai yang diberikan ke D_{pi} merepresentasikan skala ordinal yang merupakan indikasi mengenai relevansi dari karakteristik untuk perkakas p.

Tipe skala:

1. tidak relevan
2. relevan dalam situasi yang tidak biasa

3. relevan, dapat dimodifikasi meskipun berbeda
4. sangat relevan bila perangkat lunak tidak memiliki karakteristik ini reuse tidak efisien
5. sangat relevan bila perangkat lunak tidak memiliki karakteristik ini reuse tidak efektif, reuse tidak disetujui

Bila perangkat lunak baru, w , akan dibangun didalam domain aplikasi, serangkaian karakteristik domain ditarik untuknya. D_{p_i} dan D_{w_i} kemudian dibandingkan untuk menentukan apakah perkakas yang ada (p) dapat secara efektif digunakan lagi didalam aplikasi w .

Tabel Karakteristik Domain

Produk	Proses	Personil
Stabilitas persyaratan	Model proses	Motivasi
Per.lunak saat ini	Konformansi proses	Pendidikan
Batasan memori	Lingkungan proyek	Pengalaman - domain aplikasi - proses - platform - bahasa
Ukuran aplikasi	Batasan jadwal	
Kompleksitas interface	Batas anggaran	
Bhs. Pemrograman	produktivitas	
Keamanan/reliabilitas		
Lifetime requirement		Produktivitas tim
Kualitas produk		
Reliabilitas produk		

C. Pemodelan Struktural dan Point Struktur

Pemodelan struktural adalah suatu pola yang berdasarkan pendekatan rekayasa domain aplikasi yang bekerja dengan asumsi bahwa setiap domain aplikasi memiliki pola pengulangan (fungsi, data, tingkah laku) yang memiliki potensi reuse.

Point struktur memiliki 3 karakteristik:

1. Point struktur adl abstraksi yg harus memiliki sejumlah contoh terbatas.
2. Interface ke point struktur harus sederhana.
3. Point struktur hrs mengimplementasi penyembunyian informasi yg diisikan di dlm point struktur itu sendiri.

IV. Membangun komponen reusable

A. Analisis dan desain untuk reuse

Idealnya, model analisis di analisis untuk menentukan elemen-elemen model yang menunjuk ke perkakas reusable yang ada.

Kunci untuk desain reuse: (Design For Reuse/DFR)

- Data standar. Domain aplikasi, dan struktur data global standar (struktur file atau database) lengkap harus diidentifikasi.
- Protokol interface standar. Tiga tingkat protokol harus dibangun; sifat interface intramodular, desain interface teknis eksternal dan interface manusia mesin.
- Template program. Model struktur dapat berfungsi sebagai template untuk desain arsitektur dari program baru.

B. Metode Konstruksi

Konstruksi dapat dilakukan dengan bahasa pemrograman generasi ketiga yang konvensional, bahasa generasi keempat dan generator kode, teknik pemrograman visual dan peranti yang lebih canggih. Serangkaian komponen generik dan dapat diadaptasikan yang disebut kerangka (frame) atau teknologi kerangka.

C. Pengembangan Berbasis Komponen

Pendekatan konstruksi kadang diacukan ke pengembangan berbasis komponen atau perangkat lunak komponen.

Implementasi pengembangan berbasis komponen:

- Model pertukaran data. Mekanisme yang memungkinkan pemakai dan aplikasi berinteraksi dan mentransfer data untuk semua komponen reusable.

- Otomasi. Berbagai peranti macro dan script harus diimplementasi untuk memfasilitasi interaksi antar komponen reusable.
- Penyimpanan terstruktur. Data heterogen (grafis, suara, teks dan numeris) dikumpulkan dan diakses sebagai struktur data tunggal.
- Model obyek yang mendasari. Interface Definition Language (IDL)
- OpenDoc. Standar dokumen gabungan dan perangkat lunak komponen.
- OMG/CORBA. Object Request Broker/ORB memberi pelayanan komponen reusable berkomunikasi dengan komponen lain tanpa melihat lokasi didalam sistem.
- OLE 2.0 (Object Linking and Embedding). Merupakan bagian dari Component Object Model (COM).

V. Pengklasifikasian dan Pengambilan kembali Komponen

A. Penggambaran Komponen Reuse.

Konsep harus menggambarkan isi komponen, yaitu: isi dan konteks.

Skema klasifikasi komponen perangkat lunak dibagi menjadi tiga kategori:

- Enumerated Classification. Komponen digambarkan dengan menentukan struktur hirarki dimana kelas dan sub kelas yang bervariasi dari komponen diterapkan.
- Faceted Classification. Sebuah area domain dianalisis dan serangkaian ciri deskriptif dasar diidentifikasi. Ciri-ciri tersebut disebut faceted.
- Klasifikasi atribut-nilai. Sama dengan klasifikasi faced.

B. Lingkungan Reuse.

- Database komponen
- Sistem manajemen pustaka
- Sistem pemanggilan kembali perangkat lunak (OBR)
- Peranti CASE

VI. Ekonomi Penggunaan Kembali Perangkat Lunak

A. Pengaruh Terhadap Kualitas Produktivitas dan Biaya

- Kualitas. Dengan reuse cacat ditemukan dan dieliminasi sebagai hasilnya kualitas komponen meningkat
- Produktivitas. Lebih sedikit waktu yang digunakan untuk membuat rencana, model, dokumen, kode dan data yang diperlukan untuk membuat sistem.
- Biaya. Penghematan waktu neto untuk reuse diperkirakan dengan memproyeksikan bila proyek dikembangkan sejak awal.

B. Analisis Biaya Dengan Menggunakan Poin Struktur

Idealnya adaptasi, integrasi dan biaya pemeliharaan yang berhubungan pada masing-masing komponen pada suatu pustaka reuse disimpan dari masing-masing contoh penggunaan, lalu dianalisa untuk mengembangkan proyeksi biaya bagi contoh reuse selanjutnya.

$$\text{Usaha keseluruhan} = E_{\text{new}} + E_{\text{adapt}} + E_{\text{int}}$$

Dimana E_{new} adalah usaha untuk komponen perangkat lunak baru.

E_{adapt} adalah usaha untuk menyelesaikan reuse dari tiga point struktur yaitu SP_1 , SP_2 , SP_3

E_{int} adalah usaha untuk mengintegrasikan SP_1 , SP_2 , SP_3

SP_1 , SP_2 dan SP_3 ditentukan dengan mengambil rata-rata data historis.

C. Metrik Reuse

Untuk mengukur manfaat reuse pada sistem berbasis komputer. Reuse pada sistem S diekspresikan sebagai:

$$R_b(S) = [C_{\text{noreuse}} - C_{\text{reuse}}] / C_{\text{noreuse}}$$

Dimana C_{noreuse} adalah biaya pengembangan tanpa reuse

C_{reuse} adalah biaya pengembangan dengan reuse

$R_b(S)$ diekspresikan sebagai nilai nondimensional dalam range:

$$0 \leq R_b(S) \leq 1$$

Semakin tinggi nilai $R_b(S)$, reuse akan semakin menarik.

Pengukuran reuse dalam sistem OO disebut *reuse leverage* ditentukan sebagai:

$$R_{\text{lev}} = \text{OBJ}_{\text{reused}} / \text{OBJ}_{\text{built}}$$

dimana $\text{OBJ}_{\text{reused}}$ adalah jumlah objek yang digunakan kembali

$\text{OBJ}_{\text{built}}$ adalah jumlah objek yang dibangun

Bila R_{lev} bertambah R_b juga bertambah