

BAB 5

PERENCANAAN PROYEK PERANGKAT LUNAK

Proses manajemen proyek perangkat lunak dimulai dengan kegiatan project planning (perencanaan proyek). Yang pertama dari aktifitas ini adalah estimation (perkiraan). Estimasi membawa resiko yang inheren (dari diri sendiri) dan resiko inilah yang membawa ketidakpastian. Yang mempengaruhi estimasi :

- Project complexity (kompleksitas proyek)
- Project size (ukuran proyek)
- Struktural uncertainty (ketidakpastian struktural)

Tujuan Perencanaan Proyek Perangkat Lunak :

menyediakan sebuah kerangka kerja yang memungkinkan manajer membuat estimasi yang dapat dipertanggungjawabkan terhadap sumber daya, biaya dan jadwal pada awal proyek yang dibatasi oleh waktu.

Aktifitas Perencanaan Proyek PL

1. Menentukan ruang lingkup PL
2. Mengestimasi sumber daya yang dibutuhkan

RUANG LINGKUP PL

Ruang lingkup PL menggambarkan : fungsi, kinerja, batasan, interface dan reliabilitas.

Fungsi yang digambarkan dlm statemen ruang lingkup dievaluasi untuk memberikan awalan yang lebih detail pada saat dimulai estimasi. Kinerja melingkupi pemrosesan dan kebutuhan waktu respon. Batasan mengidentifikasi batas yang ditempatkan pada PL oleh perangkat keras eksternal, memori atau sistem lain.

Informasi yang dibutuhkan (awal pertemuan antara pelanggan dan pengembang)

- * Pertanyaan berfokus pada pelanggan, tujuan keseluruhan serta keuntungan.
 - Siapa di belakang permintaan kerja ini?
 - Siapa yang akan memakai solusi ini?
 - Apakah keuntungan ekonomi dari solusi yang sukses?
 - Adakah sumber daya lain bagi solusi ini?

- * Pertanyaan yang memungkinkan analis memahami masalah lebih baik dan pelanggan menyuarakan persepsi tentang sebuah solusi.

- Bagaimana Anda (pelanggan) menandai output yg baik yg akan dihasilkan oleh sebuah solusi yg baik?
- Masalah apa yang dituju solusi ini?
- Dapatkah anda menggambarkan lingkungan dimana solusi akan dipakai?
- Adakah batasan atau isu kinerja khusus yg akan mempengaruhi

PL berinteraksi dengan elemen sistem berbasis komputer. Konsep sebuah interface diinterpretasi untuk menentukan:

1. Hardware yg mengeksekusi PL dan device yg dikontrol secara tidak langsung oleh PL
2. Software yg sudah ada dan harus dihubungkan dengan PL yg baru
3. Manusia yg menggunakan PL melalui keyboard atau perangkat I/O lain
4. Prosedur

SUMBER DAYA

1. Manusia
2. Perangkat Lunak
 - Kategori yg diusulkan BEUNATAN
 - Komponen Off-the-self
 - Komponen Full-Experience
 - Komponen Partial-Experience
 - Komponen Baru
3. Lingkungan (Software Engineering Environment - SEE), menggabungkan PL dan Perangkat Keras.

Estimasi biaya dan usaha dapat dilakukan dengan cara :

1. Menunda estimasi sampai akhir proyek.
2. Berdasarkan estimasi pada proyek yg mirip sebelumnya.
3. Menggunakan 'teknik dekomposisi' yg relatif sederhana u/ estimasi biaya dan usaha proyek.
4. Menggunakan satu atau lebih model empiris bagi estimasi usaha dan biaya PL.

Akurasi estimasi proyek PL didasarkan pada :

1. Tingkat dimana perencana telah dengan tepat mengestimasi ukuran produk yg akan dibuat.
2. Kemampuan mengestimasi ukuran ke dalam kerja manusia, waktu kalender, dan dolar.
3. Tingkat dimana rencana proyek mencerminkan kemampuan tim PL.
4. Stabilitas syarat produk serta lingkungan yg mendukung usaha pengembangan PL.

Putnam dan Myers mengusulkan 4 masalah penentuan ukuran :

- **Fuzzy-logic sizing (logika kabur)**
Perencana harus mengidentifikasi tipe aplikasi, membuat besarannya dalam skala kuantitatif kemudian dibandingkan dengan rentang orisinal.
- **Function point sizing**
Perencana mengembangkan estimasi berdasarkan karakteristik domain informasi.
- **Standard component sizing**
PL dibangun dari sejumlah 'komponen standar' yg umum (subsistem, modul, laporan, program interaktif).
- **Change sizing**
Digunakan jika PL yang ada harus dimodifikasi dengan banyak cara sebagai bagian dari proyek.

Data baris kode (LOC) dan titik fungsi (FP) pada estimasi proyek digunakan sbg :

1. variabel estimasi yg dipakai untuk mengukur masing-masing elemen PL.
2. metrik baseline yg dikumpulkan dari proyek yg lalu dan dipakai dengan variabel estimasi untuk mengembangkan proyeksi kerja dan biaya.

Expected Value untuk variabel estimasi :

$$EV = (S_{opt} + 4S_m + S_{pess}) / 6$$

EV = Expected value

S_{opt} = Estimasi optimistik

S_m = Estimasi paling sering

S_{pess} = Estimasi pesimistik

Apakah estimasi ini benar ? ' Kita tidak yakin!'

Bagaimanapun canggih teknik estimasi harus di-cross-check dengan pendekatan lain.

Contoh estimasi berbasis LOC

PL CAD akan menerima data geometri dua dan tiga dimensi dari seorang perancang yang akan berinteraksi dan mengontrol sistem CAD melalui suatu interface pemakai. Kajian spesifikasi sistem menunjukkan bahwa PL akan mengeksekusi Workstation dan harus berinteraksi dengan berbagai peripheral grafis komputer spt mouse, digitizer dan printer laser.

Diketahui :

Perhitungan LOC untuk fungsi analisis geometri 3D (3DGA) :

optimis : 4600
 most likely : 6900
 pesimistik : 8600

$$EV = (4600 + 4 \cdot 6900 + 8600) / 6$$

$$= 6800 \text{ LOC}$$

Jumlah tersebut dimasukkan ke dalam tabel, begitu juga untuk perhitungan yang lain. Sehingga diperoleh :

Tabel perkiraan (estimasi) untuk metode LOC

Fungsi	LOC terestimasi
interface pemakai & fasilitas kontrol (UICF)	2.300
analisis geometrik dua dimensi (2DGA)	5.300
analisis geometrik tiga dimensi (3DGA)	6.800
manajemen databse (DBM)	3.350
fasilitas display grafis komputer (CGDF)	4.950
kontrol periperal (PC)	2.100
modul analisis desain (DAM)	8.400
baris kode terestimasi	33.200

Jika :

Produktifitas rata-rata organisasional = 620 LOC/person-month

Upah karyawan = \$8.000 per bulan

Biaya per baris kode = \$13

Maka : Tingkat produktifitas = $\frac{\text{jumlah titik fungsi}}{\text{jumlah orang-bulan}}$

$$\text{Jumlah karyawan} = \frac{33200 \text{ LOC}}{620 \text{ LOC/bln}} = 53,5 \approx 54 \text{ orang}$$

Estimasi biaya proyek berdasar LOC

$$= 33.200 \text{ LOC} * \$ 13$$

$$= \$ 431.600$$

Estimasi biaya proyek berdasar upah

$$= 54 \text{ orang} * \$8.000$$

$$= \$432.000$$

Estimasi berbasis FP (Function Point)

Dekomposisi untuk perhitungan berbasis FP berfokus pada harga domain info daripada fungsi PL. Perencana proyek memperkirakan input, output, inquiry, file dan interface eksternal. Untuk tujuan perkiraan tersebut faktor pembobotan kompleksitas diasumsikan menjadi rata-rata.

Setiap faktor pembobotan kompleksitas diestimasi dan faktor penyesuaian kompleksitas dihitung seperti dibawah ini :

Faktor	Harga
Backup and recovery	4
Komunikasi data	2
Pemrosesan terdistribusi	0
Kinerja kritis	4
Lingkungan operasi yang ada	3
Entri data on-line	4
Transaksi input pada layar ganda	5
File master yang diperbarui on-line secara on-line	3
Nilai kompleks domain informasi	5
Pemrosesan internal yang kompleks	5
Kode yg didesain untuk dapat dipakai lagi	4
Konversi/instalasi dalam disain	3
Instalasi ganda	5
Aplikasi yg didesain bagi perubahan	5
Faktor penyesuaian kompleksitas	1.17
TOTAL	53.17

Perkiraan harga domain informasi :

Tabel perkiraan harga domain informasi

nilai domain informasi	opt	likely	pess	jumlah estimasi	bobot	jumlah FP
jumlah input	20	24	30	24	4	96
jumlah output	12	15	22	16	5	80
jumlah inquiry	16	22	28	22	4	88
jumlah file	4	4	5	4	10	40

jumlah interface eksternal	2	2	3	2	7	14
jumlah total						318

jumlah estimasi (lihat rumus EV)

bobot (lihat kembali bab 4)

jumlah FP = jumlah estimasi * bobot

Total faktor pembobotan = $\sum F_i = 53.17$

Total FP = 318

FP terestimasi = jumlah total * (0.65 + 0.01 * $\sum F_i$)
 = 318 * (0.65 + 0.01 * 53.17)
 = 375

Diketahui :

Produktifitas = 6.5 LOC/pm (dari historis)

Upah = \$ 8.000/m

Biaya FP = $\frac{\$ 8.000}{65 \text{ LOC}}$ = \$ 1.230

Estimasi biaya proyek

= Biaya FP * FP terestimasi

= \$ 1.230 * 375

= \$ 461.250

Usaha terestimasi

= $\frac{\text{Total biaya}}{\text{upah/p}}$ = $\frac{\$ 461.250}{\$ 8.000}$ = 58 p/m

MODEL COCOMO

Barry Boehm memperkenalkan hirarki model estimasi PL dengan nama **COCOMO**

(**CO**nstructive **CO**st **MO**del = Model Biaya Konstruktif) yang berbentuk sbb :

1. Model COCOMO Dasar

Menghitung usaha pengembangan PL (dan biaya) sbg fungsi dari ukuran program yg diekspresikan dalam baris kode yg diestimasi (LOC).

2. Model COCOMO Intermediate

Menghitung usaha pengembangan PL sbg fungsi ukuran program dan serangkaian 'pengendali biaya' yg menyangkut penilaian yg subyektif thd produk, perangkat keras, personil dan atribut proyek.

3. Model COCOMO Advance

Menghubungkan semua karakteristik versi intermediate dg penilaian thd pengaruh pengendali biaya pd setiap langkah (analisis, perancangan, dll) dari proses rekayasa PL.

Model COCOMO mendefinisikan 3 kelas proyek PL yi :

1. Model Organik

Ukuran proyek relatif kecil, PL yang dibuat atau dikembangkan lebih simpel dengan aplikasi kerja yg baik. Misal program analisis termal yang dikembangkan untuk kelompok transfer panas.

2. Model Semi Detached

Ukuran proyek dan kekompleksan perangkat cukup besar dengan pengalaman kerja campuran (ada yg telah berpengalaman dan ada yg belum berpengalaman). Misal sistem pemrosesan transaksi dengan syarat tertentu untuk perangkat keras terminal dan perangkat lunak database.

3. Model Embedded

Ukuran proyek dan kekompleksan PL yg dikembangkan atau dikerjakan besar. Misal perangkat lunak kontrol penerbangan untuk pesawat udara.

Pesamaan COCOMO Dasar

$$E = a_b (KLOC)^{b_b}$$

$$D = c_b E^{d_b}$$

Dimana :

E = Effort (usaha yang diaplikasikan - pm)

D = waktu pengembangan (m)

KLOC = jumlah perkiraan baris kode (dalam ribuan)

a_b, b_b, c_b, d_b = koefisien (lihat tabel)

Tabel Model COCOMO Dasar

Proyek PL	a_b	b_b	c_b	d_b
organik	2.4	1.05	2.5	0.38
semi-detached	3.0	1.12	2.5	0.35
embedded	3.6	1.20	2.5	0.32

Model dasar ini dapat diperluas dengan mempertimbangkan kumpulan 'atribut pengendali biaya' yg dikelompokkan dalam 4 kategori utama :

1. Atribut produk
 - ukuran keandalan proyek
 - ukuran dari aplikasi database
 - kekompleksan produk
2. Atribut perangkat keras
 - kendala performansi run-time
 - kendala memori
 - lingkungan dari violability dari virtual memori
 - waktu perputaran yg diperlukan
3. Atribut personil
 - kemampuan sistem analis
 - kemampuan software engineering
 - pengalaman aplikasi
 - pengalaman virtual mesin
 - pengalaman bahasa pemrograman
4. Atribut proyek
 - pemakaian alat bantu PL
 - metode aplikasi software engineering
 - jadwal pengembangan

Masing-masing dari 15 atribut di atas dirata-rata dlm sebuah skala 6 titik dg rentang dari 'sangat rendah' ke 'sangat tinggi' (dlm kepentingan atau harga).

Persamaan COCOMO Intermediate

$$E = a_i (\text{KLOC})^{b_i} * \text{EAF}$$

dimana :

EAF = Effort Adjustment Factor (faktor penyesuaian usaha)
yg mempunyai range antara 0.9 sampai 1.4

a_i, b_i = koefisien (lihat tabel)

Tabel Model COCOMO Intermediate

Proyek PL	a_i	b_i
organik	3.2	1.05

semi-detached	3.0	1.12
embedded	2.8	1.20

Contoh estimasi model COCOMO

Kita aplikasikan model dasar pada contoh PL CAD sebelumnya dengan koefisien seperti pada tabel

$$E = a_b (KLOC)^{b_b}$$

$$\begin{aligned} E &= 2.4 (KLOC)^{1.05} \\ &= 2.4 (33.2)^{1.05} \\ &= 95 \text{ pm} \end{aligned}$$

Harga ini jauh lebih tinggi dibanding estimasi sebelumnya karena model COCOMO mengasumsikan tingkat LOC/pm yang jauh lebih rendah.

Untuk menghitung durasi proyek :

$$D = c_b E^{d_b}$$

$$\begin{aligned} D &= 2.5 (E)^{0.38} \\ &= 2.5 (95)^{0.38} \\ &= 12.3 \text{ bulan} \end{aligned}$$

Harga durasi proyek memungkinkan perencana untuk menentukan jumlah orang yang disetujui (N)

$$\begin{aligned} N &= E/D \\ &= 95/12.3 \\ &= 7,7 \approx 8 \text{ orang} \end{aligned}$$

Kenyataannya, perencana dapat memutuskan hanya menggunakan 4 orang saja dan memperpanjang durasi proyek.

Catatan : Hubungan antara usaha dan waktu tidak linier.

KEPUTUSAN MAKE-BUY

Pada aplikasi PL, dari segi biaya sering lebih efektif membeli dari pada mengembangkan sendiri. Manajer RPL dihadapkan pada keputusan make-buy dengan pilihan :

1. PL dapat dibeli (atau lisensi) off-the-self.
2. Komponen PL full-experience dan partial-experience, dapat diperoleh dan kemudian dimodifikasi dan integrasi untuk memenuhi kebutuhan sendiri.
3. PL dapat dibuat custom-built oleh kontraktor luar untuk memenuhi spesifikasi pembeli.

Untuk produk PL yang mahal, langkah-langkah di bawah ini dapat dipertimbangkan:

1. Kembangkan spesifikasi untuk fungsi dan kinerja PL yg diperlukan.
2. Perkirakan biaya internal untuk pengembangan dan tanggal penyampaian.
- 3a. Pilih tiga atau empat calon aplikasi yang paling cocok dengan aplikasi anda.
- 3b. Pilih komponen yang reusable yg dapat membantu konstruksi aplikasi yg diperlukan.
4. Kembangkan sebuah matriks perbandingan untuk membandingkan calon PL.
5. Evaluasi masing-masing paket PL berdasarkan kualitas produk sebelumnya, dukungan penjual, arah proyek, reputasi dsb.
6. Hubungi pemakai PL lain dan mintalah pendapat mereka.

Pada analisis akhir, keputusan make-buy berdasarkan kondisi sbb:

1. Tanggal penyampaian
2. Biaya yang diperlukan
3. Dukungan

MEMBUAT POHON KEPUTUSAN

Rekayasa atau organisasi PL dapat menggunakan teknik statistik analisis pohon keputusan dengan pilihan :

1. membangun sistem X dari permulaan
2. menggunakan lagi komponen partial experience yang ada untuk membangun sistem
3. membeli sebuah produk perangkat lunak yang dapat diperoleh dan dimodifikasi untuk memenuhi kebutuhan lokal
4. mengkontrakkan pengembangan PL ke vendor luar

Bila sistem dibangun dari permulaan, hanya 70% probabilitasnya sehingga pekerjaan menjadi sulit. Perencana proyek dapat memproyeksikan usaha pengembangan yang sulit berbiaya \$450.000, usaha yang sederhana diperkirakan berbiaya \$380.000.

Expected value untuk biaya dihitung sepanjang cabang pohon keputusan, adalah :

$$\text{Expected Cost} = \sum (\text{jalur probabilitas})_i * (\text{biaya jalur terestimasi})_i$$

dimana i adalah garis edar pohon keputusan.

Contoh :

$$\text{expected cost}_{\text{build}} = 0.30 (\$380 \text{ K}) + 0.70 (\$450 \text{ K}) = \$ 429 \text{ K}$$

$$\begin{aligned} \text{expected cost}_{\text{reuse}} &= 0.40 (\$275 \text{ K}) + 0.60 (0.20 (\$310 \text{ K}) + 0.80 (\$490 \text{ K})) \\ &= \$ 382 \text{ K} \end{aligned}$$

$$\text{expected cost}_{\text{buy}} = 0.70 (\$210 \text{ K}) + 0.30 (\$400 \text{ K}) = \$ 267 \text{ K}$$

$$\text{expected cost}_{\text{contract}} = 0.60 (\$350 \text{ K}) + 0.40 (\$500 \text{ K}) = \$ 410 \text{ K}$$

Berdasar biaya probabilitas dan proyeksi, expected cost yang paling rendah adalah pilihan buy

Catatan : Banyak kriteria yang harus dipertimbangkan, bukan hanya biaya, seperti pengalaman pengembang/ vendor/ kontraktor, penyesuaian kebutuhan, kecenderungan perubahan dapat mempengaruhi keputusan akhir!