

## **BAB 13**

### **ESTIMASI (PENDUGAAN / PERKIRAAN)**

#### **KELEMAHAN LINK DALAM JARINGAN**

##### **13.1 Pendahuluan**

Estimasi merupakan sebuah proses pengulangan. Pemanggilan ulang estimasi yang pertama dilakukan selama fase definisi, yaitu ketika Anda menulis rencana proyek awal. Hal ini perlu dilakukan, karena Anda membutuhkan sebuah estimasi untuk proposal. Tetapi berdasarkan statistik dari DEC, NASA, TRW, dan lembaga yang lainnya, menunjukkan bahwa keakuratan estimasi yang dilakukan memiliki point sebesar 50%-100%. Setelah fase analisis direncanakan ulang. Anda harus memeriksa estimasi dan merubah dari rencana proyek awal menjadi rencana proyek akhir. Pada tahap ini keakuratan estimasi Anda menjadi berkurang, yaitu sekitar 25%-50%. Setelah dikerjakan sampai pada tingkat menengah, Anda periksa kembali estimasi tersebut. Dengan menggunakan pengetahuan yang didapat seiring dengan waktu, keakuratan estimasi tersebut hanya mencapai 10%. Meskipun tidak bisa dikatakan sebagai aktivitas eksplisit dalam fase yang lain, rencana untuk memperbaiki estimasi setiap waktu, memerlukan pengetahuan yang baru.

Semua teknik yang akan Kita diskusikan adalah tergantung pada kesulitan, seperti membawa sesuatu ke dalam keadaan yang lebih kecil. Hal ini penting untuk mendapatkan hasil kerja yang baik sesuai strukturnya (didiskusikan pada Bab 3), sebelum beberapa estimasi dicoba.

##### **13.2 Teknik-teknik Estimasi**

Ada 3 (tiga) teknik yang digunakan untuk melakukan estimasi, yaitu : keputusan profesional, sejarah dan rumus-rumus.

###### **➤ Keputusan Profesional**

Katakanlah, bahwa Anda merupakan orang yang memiliki pengalaman yang luas dalam memprogram “report generation modules”. Anda melakukannya dengan pendekatan design report tersebut dan memperkirakan berapa lama waktu yang dibutuhkan untuk membuat program tersebut. Setelah mempelajari desain program selama 5 menit, programer lalu menutup matanya selama 5 menit (Dia tidak tidur, tetapi berhitung), dan kemudian mengatakan “15 hari”. Inilah yang disebut dengan Keputusan Profesional murni.

Keuntungan dari teknik ini adalah cepat, dan jika seseorang sudah ahli dalam teknik ini, maka estimasinya pasti akan lebih akurat. Sedangkan kerugian dari teknik ini yaitu bahwa Anda membutuhkan seorang ahli yang berpengalaman dalam bidang ini dan biasanya beberapa ahli tersebut akan bekerja keras untuk mendapatkan estimasi yang tepat. Lagi pula, estimasi yang didapat adalah untuk jangka waktu yang lama dan ini harus dikerjakan oleh seorang ahli. Hal ini tidak dapat tergantung hanya pada ahli tersebut, jika yang lain ingin mengerjakannya tidak menjadi masalah. Tapi, hal ini sangat berbahaya jika mempercayakannya kepada seseorang yang memiliki sedikit pengetahuan dalam bidang ini.

###### **➤ Sejarah**

Jalan keluar dari ketergantungan pada orang dan untuk membuat estimasi lebih khusus yaitu Anda harus mengerti tentang sejarahnya. Tulislah berapa lama masing-masing tugas dapat diselesaikan dan siapa yang bertanggung jawab atas tugas tersebut. Anda dapat membandingkan tugas yang akan di estimasi dengan tugas yang sama yang dikerjakan lebih awal, setelah itu mulailah dengan melakukan estimasi. Hal ini dimaksudkan agar Anda menjabarkan suatu proyek ke dalam beberapa tugas yang biasanya diulang dan mudah untuk dibandingkan. Untuk memprogramnya, mungkin generasi dalam bentuk formulir input, sebuah laporan, perhitungan rumus-rumus, dan lain-lain. Perusahaan atau departemen akan membuat tipe proyek yang sama. Carilah dasar untuk membangun “blocks” dan dokumen yang dibutuhkannya. Jika Anda sangat intelligent, lakukan langkah pertama dan bangunlah blocks tersebut dalam bentuk yang siap

pakai. Anda dapat menduga bahwa sebuah penggunaan ulang ternyata lebih akurat jika dibandingkan dengan penulisan ulang.

Dalam membandingkan satu apel dengan apel yang lainnya, Anda harus menuliskan perbedaannya. Catatan statistik dari IBM dan DEC memperlihatkan bahwa kemungkinan perbandingan antara produksi komputer yang baik dengan yang buruk adalah sebesar 8 : 1.

➤ **Rumus-rumus**

Ada beberapa rumus yang digunakan dalam software estimasi. Software yang baik untuk diketahui yaitu *COCOMO* (referensi 15). *COCOMO* dapat digunakan untuk memperkirakan biaya proyek, usaha (person months), jadwal (months) dan staf (number of staff) untuk masing-masing fase berikut ini :

Preliminary Design	-	our Analysis Phase
Detailed Design (DD)	-	our Design Phase
Code and Unit test (CUT)	-	same as ours
System Test	-	our System Test and Acceptance Phase

Ada 3 tipe penginputan dengan *COCOMO* : pertama, pemasukan biaya bulanan dari staf. Baik staf yang berkedudukan sebagai programmer, analis, designer, test staff, administrasi dan technical writer. Gambar 13.1 menunjukkan sebuah layar penginputan yang digunakan untuk tipe ke dua dari penginputan. Faktor-faktor ini mencirikan level keseluruhan dari kelengkapan software yang ada, ukuran dan kemampuan dari komputer yang digunakan untuk pengembangan, kemampuan menampung dan pengalaman staf, dan juga pemrograman praktis serta alat-alat yang digunakan.

Estimation Mode Form

Name : Test

Mode : Simple (Intermediate Complex)

Outputs :

PDCOST : 5500 (Prel. Des. = Analysis Phase)

DDCOST : 5500 (Detail Des. = Design Phase)

CUTOST : 5500 (Ccode&Unit Test)

ITCOST : 4800 (Int. & Test)

Inputs :

Line of Source Code : 10000

Factors (1 – low through to 5 – extra high)

Relability :	3	Exec time const:	1	Analyst cap.	1
Data base size :	2	RAM cconstrained :	3	Applicat'n exp	3
SW complexity :	3	VM volatility	1	Progrm'r cap	2
		Turnaround	2	VM experience	3
				Lang exp	4
Modern programming practices :				3	
Software tools :				4	
Schedule constrained :				3	

The factors are :

1 - Very Low   2 - Low   3 - Nominal   4 - High   5 - Very High

**Gambar 13.1** Tampilan dengan menggunakan software *COCOMO*

Pada hal ini, Anda mungkin akan merasa bahwa *COCOMO* akan melakukan pendugaan yang baik, sejak software ini selalu tepat menentukan proyek yang lama. Tetapi, kesulitannya yaitu setiap akhir dari penggunaan software ini *COCOMO* selalu menanyakan nomor garis yang

terdapat pada kode sumber (LOSC). Pada saat itu, Anda telah memiliki pengetahuan yang cukup mengenai sistem untuk memperkirakan LOSC dengan teliti, Anda tidak memerlukan beberapa rumus. Namun, Anda hanya memperkirakan keseluruhan proyek dengan teliti.

Titik fungsi rumus-rumus. Pendekatan COCOMO dapat diperbaiki oleh produk-produk yang menghitung LOSC, yang diberikan fungsi-fungsi dari sebuah produk tersebut dan hasilnya dimasukkan ke dalam rumus-rumus COCOMO. Salah satu produknya adalah Before You Leap (BYL) oleh Gordon Group. Gambar 13.2 adalah tampilan dari BYL yang digunakan cepat untuk titik fungsi seperti bahasa yang digunakan. Hasil-hasil yang diberikan oleh BYL seperti keseluruhan dari COCOMO, kecuali hasil akhirnya akan ditampilkan dalam grafik-grafik, seperti pie chart atau diagram batang.

FUNCTION POINT ANALYSIS				Model 1			
Function Count and Complexity							
External Input/Inquiry	25 Simple	10 Avarage	4 Complex	Compiler :	[ ]		
External Output	10 Simple	5 Avarage	3 Complex	Ada			
Logical Internal File	1 Simple	1 Avarage	0 Complex	Coefficient :			
External Interface File	5 Simple	2 Avarage	0 Complex	71			
Processing Complexity – Degree of Influence							
Data Communications	None	Insignif	<u>Mod</u>	Avg	Signif	Strong	Function Pt. Analysis [ ]
Distributed Functions	None	Insignif	Mod	<u>Avg</u>	Signif	Strong	
Performance	None	Insignif	Mod	Avg	<u>Signif</u>	Strong	
F1 = HELP							
<default set of data>		Calib : dft	CstDrv:dft	28MM 9Months			

Gambar 13.2 BYL Function Point Analysis Screen

Harga produk yang lain tergantung pada perkiraan-perkiraan dari asosiasi komputer. CA – Estimasi memperbolehkan Anda untuk memberikan biaya, upaya, jadwal dan susunan staf di dalam COCOMO, tetapi beberapa penambahan disarankan atas permintaan hardware-nya (oriented IBM), rata-rata analisa keuangan, analisa resiko dan biaya pemeliharaan untuk single maupun keseluruhan proyek lingkungannya. CA- Estimasi dapat dimasukkan ke sistem pengembangan peralatan dan bentuk aslinya. Itu dapat diperkirakan untuk menarik pembelian dalam bentuk eceran atau bungkusan. Bentuk-bentuk dari faktor pemasukan ke dalam CA – Estimasi adalah terdaftar dalam gambar 13.3. Catatan dari ini lebih lengkap dan lebih berpengalaman dari pada COCOMO.

INPUT FACTORS	AFFECTED ESTIMATE
Business costs	Payback
Tool costs	
Hardware costs	
Customer complexity	Effort, Function point,
Customer geography	Maintanance
Developer's familiarity	
Target system complexity	
Development strategy	Staff, Cost
Skills Deployment	

System Type	Hardware required
Application category	
System size	Risk analysis
Project organization	
Deadlines	Multi project aspects
Other project	
Background Workloads	

**Gambar 13.3** Tabel dari input/output CA-Estimasi

### Program Estimasi

Pendekatan satu rumus yang telah berhasil untuk fase estimasi perhitungan program adalah kemiripan sebuah pendekatan fungsi point. Dalam hal ini akan dijelaskan mengenai pemahaman bagaimana semua rumus itu bekerja. Jika Anda melakukan latihan untuk tingkat pemrograman, Anda akan lebih mengerti dari tingkat-tingkat yang lain.

Pada dasarnya ada 2 faktor yang mempengaruhi lamanya waktu dari sebuah tugas , kerumitan dari tugas (C) dan produktivitas dari seseorang yang memperagakannya. Produktivitas dari seorang tergantung pada lamanya pengalaman seseorang (G) dalam bidang tersebut dan pengetahuan dari pekerjaan yang khusus(J). Rumus ini dapat digambarkan sebagai berikut :

$$D = C \times (G \times J) \quad \text{(Rumus 1)}$$

Dimana :

- D adalah lamanya waktu
- C adalah faktor kompleksitas kesulitan)
- G adalah faktor pengalaman
- J adalah faktor pengetahuan pekerjaan

### Complexity (Kerumitan)

Untuk memecahkan faktor yang rumit dari sebuah tugas, Anda harus memecahkannya menjadi bagian yang paling kecil, yang dapat diulang fungsi-fungsinya diantara tugas-tugas dan tambahkan kompleksitas dari masing-masing fungsi. Untuk sebuah tugas program, hal ini disebut fungsi point. Fungsi point dapat berupa input pemakai, user display, peripheral I/O, penyusunan data, pemeriksaan kondisi, perhitungan, percabangan dan pemanggilan, dan lain-lain (kadang kala, instruksi dari bahasa pemrograman seperti SEQUENCE, IF WHILE, UNTIL, FOR, CASE dan ASSIGMENT sangat diperhitungkan penggunaannya). Kerumitan suatu program tentu saja tergantung pada yang digunakan dan kerumitan masing-masing fungsi pointnya. Semua ini terdapat dalam sebuah tabel seperti pada gambar 13.4.

PROGRAMMING ESTIMATE COMPLEXITY FACTORS ( C )

LANGUAGE	FUNCTION POINT	SIMPLE	CMPLX	VERY CMLPX
Interp't	User Input	1	3	4
	User Display	1	3	4
	Peripheral Input	3	6	8
High-Level	User Input	2	4	5
	User Display	2	4	5
	Peripheral Input	4	7	9

Assembly	User Input	4	5	8
	User Display	4	5	8
	Peripheral Input	6	8	10
Changing an	User Input	1		
Existing Pgm.	User Display	1		
	Peripheral Input	1	2	

**Gambar 13.4** Faktor-faktor complexity program

Faktor-faktor tersebut ditentukan dengan ukuran yang nyata/benar kemudian diatur sedemikian rupa, sehingga rumus  $D = C \cdot x (G + J)$  dapat digunakan dalam kehidupan sehari-hari. Faktor-faktor dalam gambar 13.4 sampai gambar 13.6 itu berdasarkan atas keterangan dari IBM dan hanya merupakan sebuah metode ilustrasi, sehingga Anda harus menemukan sendiri faktor-faktor yang lainnya. Total complexity ( C ) untuk sebuah program akan menjumlah semua faktor-faktor untuk fungsi point.

### Produktivitas

Anda membutuhkan beberapa faktor untuk membangun produktivitas pegawai. Hal ini lebih sulit untuk dilakukan jika dibandingkan dengan faktor-faktor complexity, sejak keberhasilan masyarakat dapat mengubah ketergantungan pada daya ketertarikan mereka, perasaan dan sebagainya. Keberhasilan itu dipengaruhi oleh pengalaman dan pengetahuan di bidang pekerjaan. Berikut ini adalah daftar faktor-faktor berdasarkan pengalaman seseorang secara umum.

PRODUCTIVITY FACTORS		
BASED ON YEARS OF GENERAL EXPERIENCE (G)		
Prg. Type	Years of Experience	Factor Range
Senior	5 +	0.5 - 0.75
Avarage	1.5 - 5	1.0 - 1.5
Junior	0.5 - 1.5	2.0 - 3.0
Trainer	0.0 - 0.5	3.5 - 4.0

**Gambar 13.5** Faktor-faktor pengalaman secara umum (G)

Catatan mengenai faktor penyebab adalah menunjukkan rata-rata perhitungan untuk berbagai macam masyarakat. Angka-angka ini juga berdasarkan atas keterangan dari IBM. Kembangkanlah faktor-faktor yang Anda punya berdasarkan ernyataan '1' di tengah-tengah masyarakat dan isilah data untuk masyarakat lain berdasarkan atas pengalaman mereka.

Kecepatan dimana seorang profesional akan menghasilkan sebuah produk tidak hanya bergantung pada pengalaman umum (G), tetapi juga berdasarkan atas pengalaman masyarakat yang memiliki keistimewaan dalam hal kerajinan dan hubungan pekerjaan. Dalam hal ini, banyaknya pengetahuan dituntut untuk dijadikan sebagai salah satu faktor di dalamnya. Tabel di bawah ini bisa digunakan untuk menilai kuantitas dari pengetahuan (J).

PRODUKTIVITY FACTORS  
BASED ON KNOWLEDGE OF THE PARTICULAR JOB (J)

JOB KNOWLEDGE	KNOWLEDGE REQUIRED		
	Much	Some	None
Detailed knowledge of this job and detailed knowledge of related jobs	0.75	0.25	0.00
Good knowledge of this job and fair knowledge of related jobs	1.25	0.50	0.00
Fair knowledge of this job and no knowledge of related jobs	1.50	0.75	0.00
No knowledge of this job and detailed knowledge of related jobs	1.75	1.00	0.25
No knowledge of this job and no knowledge of related jobs	2.00	1.25	0.25

**Gambar 13.6** Faktor-faktor job knowledge (J)

**Contoh Penggunaan Rumus 1,  $D = C \times (G + J)$**

Mari kita perkirakan berapa lama waktu untuk menuliskan sebuah program PASCAL. Angka-angka dalam tanda kurung menunjukkan baris-baris dalam perhitungan di bawah ini. (1) Awal program yang digunakan user untuk sesuatu, (2) membaca jawaban user, (3) disahkan, (4) membaca record dari disk, (5) menghitung sebuah angka, (6) menulis kembali record ke disk, (7) tampilkan hasil dari user, (8) Panggil kembali modul yang lain, (9) programmer yang memiliki pengalaman selama 2 tahun adalah sangat baik, dan rata-rata semua programmer memilikinya, (10) Pengetahuan merupakan bagian dari aplikasi, namun tidak ada pengetahuan yang berhubungan dengan aplikasi, (11) Beberapa pengetahuan dalam hal pekerjaan sangat dibutuhkan untuk melakukan pekerjaan ini. Perhitungan complexity ( C ). Faktor-faktor dalam gambar 13.4 adalah untuk high level language.

Function	Factor
(1) USER DISPLAY (SIMPLE)	2
(2) USER INPUT (SIMPLE)	2
(3) CONDITION CHECKING (CMLPX)	4
(4) PERIPHERAL INPUT (SIMPLE)	4
(5) CALCULATION (SIMPLE)	2
(6) PERIPHERAL OUTPUT (SIMPLE)	4
(7) USER DISPLAY (SIMPLE)	2
(8) CALLING (SIMPLE)	3
TOTAL COMPLEXITY	$C = 23$

**Perhitungan Produktivitas**

Percobaan G secara umum (faktor pada gambar 13.5)

(9) Rata-rata programmer (contoh: 2 tahun)  $G = 100$

Pengalaman kerja J (faktor pada gambar 13.6)

(10) Pengetahuan tentang aplikasi, tidak berhubungan dengan pengetahuan melainkan

(11) Diinginkan  $J = 0.75$

Keseluruhan dari rumus 1 ini :

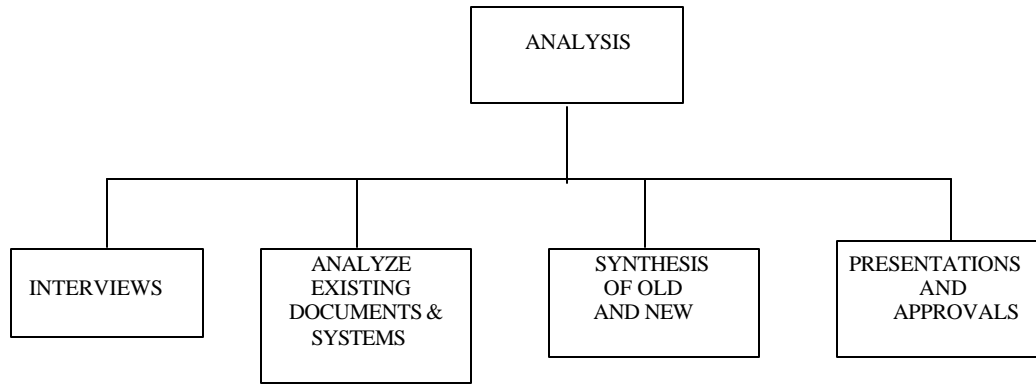
$$\text{Jangka waktunya} = 23 \times (1.00 + 0.75) = 40.25$$

Sehingga, beberapa orang akan menggunakan waktu selama 40 hari untuk perencanaan, dokumen, kode dan tes program.

Kesimpulan dari metode rumus ini. Metode ini akan bekerja jika Anda mengembangkan faktor-faktor dengan tepat. Keindahan dari pendekatan ini dapat digunakan untuk berbagai macam tugas, baik untuk menghasilkan program atau untuk membangun rumah. Catatan metode ini seperti catatan dari beberapa penafsiran metode yang lain, tergantung bagaimana Anda mengelompokkannya.

### PENAFSIRAN PADA FASE ANALISIS

Penafsiran pada fase analisis sangat sulit karena menunjukkan arah aktivitas manusia. Di sini ada satu metode. Awalnya, hal itu tergantung akan kerusakan sebuah tulisan dalam beberapa komponen. Berikut gambar 13.7 adalah level 2 dari WBS untuk analisis.



**Gambar 13.7** Analysis major components

Lembar kerja pada gambar 13.8 adalah berasal dari level 3 dalam WBS : masing-masing level 2 tidak berhubungan dengan penyediaan sub komponen dan kegiatan. Faktor multiplikasi menunjukkan masing-masing aktivitas masyarakat dalam kehidupan sehari-hari.

#### ANALYSIS ESTIMATING WORKSHEET

INTERVIEWS							
TYPE	NO.	FACTORS	DAYS	REF	SUB	IPTS	TOT
Management		.5		1.2			
Supervisor		1.0		1.2			
TOT INTER.						1.25	

ANALYZE EXISTING DOC.'S, SYSTEMS			
TYPE	NO.	FACTOR	TOT
Input forms		.5	
Output Forms		.5	
Systems		4	

SYNTHESIS			
TYPE	NO.	FACTOR	TOT
Interviews		2.5	
Documents		0.5	
Old system functions kept		1.0	

PREPARE FUNC. SPEC			
TYPE	NO.	FACTOR	TOT
Documents and systems		.5	
Synthesis		.5	
PRESENTATION (INC. PREP)			
TYPE	NO.	FACTOR	TOT
Attendees		.25	
Locations		.5	
Approvals		1.0	

**Gambar 13.8** Analysis estimating worksheet

Unsur-unsur pada contoh 13.8 yang berasal dari data statistik DEC dan dapat digunakan untuk perkalian unsur untuk mengkalkulasi bagaimana seharusnya penempatan aktifitas lainnya.

### Contoh Analisis Perkiraan

Contoh 13.9 dibawah adalah sebuah pengisian dalam memperkirakan analisa. Nomor baris dalam tanda kurung yang terlampir untuk membantu dalam penjelasan. Item/bagian-bagian yang diberi bercetak tebal telah diisi oleh estimator.

Analysis Estimate For ABC System							
Interview							
Type	No.	Factor	Days	Ref.	Sub	IPTS	TOT
(1) Management	2	.5	1.0	1.2	1.2		
(2) Supervisor	2	1.0	2.0	1.2	2.4		
(3) Technical	1	1.5	1.5		1.5		
(4) Clerical		.5					
(5) TOT INTER.	5				5.1	1.25	6.4
Analyze Existing DOC.'S, Systems							
Type	No.	Factor					TOT
(6) Input forms	10	.5					5
(7) Output forms	20	.5					10
(8) Manual Files	2	2.0					4
(9) Automated Files		2.0					
(10) Tables	2	2.0					4
(11) Systems	1	4.0					4
(12) TOTAL EXISTING SYSTEMS	35	2					5
Synthesis							
Type	No.	Factor					TOT
(13) Interviews	5	.5					12.5
(14) Documents	35	.5					17
(15) Old System Function Kept	10	1.0					10
(16) Old System Function Changed	5	1.0					5
(17) New Functions	10	1.5					15
(18) Alternatives Presented		2.0					
(19) TOTAL SYNTHESIS	64						59.5

Prepare Func. Spec.			
Type	No.	Factor	TOT
(20) Interviews	5	.25	1
(21) Documents and System	35	.25	9
(22) Synthesis	64	.25	16
(23) Other			
(24) TOTAL PREPARE FUNC. SPEC.	104		26
Presentation (INC)			
Type	No.	Factor	TOT
(25) Attendees	10	.25	2.5
(26) Locations	2	.5	1
(27) Approvals	1	1.0	1
(28) Charts/Slidos	1	.5	2.5
(29) TOTAL PRESENTATION			7
(30) GRAND TOTAL ANALYSIS (PD)			125.9

Figure 13.9 Analysis estimate example

### Penjelasan

*Wawancara.* (1) 2 manajer perlu untuk diwawancarai, faktor waktu 5 pada 1 (dalam hari). REF yang ada diatas untuk menunjukkan : kamu perlu untuk menelponnya, buat perjanjian, pemberian lamaran , dan sebagainya. Mengalikan dan meletakkan jumlah dibawah subtotal. Mirip dengan suntotal untuk wawancara 2 supervisor dan 1 teknisi dihitung. Total dari sub dijumlahkan dan dikalikan dengan 1,25 untuk menghambat selama wawancara untuk mendapatkan total wawancara dalam 6,4 hari.

*Menganalisa dokumen dan sistem.* (6) ada 10 bentuk input manual, (7) 20 laporan, (8) 2 file dalam pengisian kabinet (siswa dan lembaga), (10) 2 tabel (bahan x course, harga x course), (11) dan 1 sistem. Hasil dari baris ini, ketika dikalikan oleh faktor pendukung memberikan total keberadaan sistem selama 27 hari.

*Synthesis.* Adalah berfikir, berdiskusi, dan bagian penganalisaan. (13) untuk masing-masing dari 5 (No. dari (5)) wawancara disini adalah sebuah faktor sintesis dari .5. (14) Kemiripan untuk 35 (No. dari (12)) dokumen. Sekarang kembali pada bagian sulit. Jika ada sebuah sistem pengadaan manual atau otomatis, jumlahnya mendekati berapa banyak yang ada dalam sistem yang lama yang akan dihasilakan dalam sistem yang baru dan meletakkannya dibawah No. pada (15). Jumlah nomor dalam fungsi sistem lama yang diganti untuk sistem baru dan tempat pada (16). No. (17) adalah nomor dalam seluruh fungsi baru untuk dikerjakan dalam sistem yang baru. Jika semua alternatif mendekati untuk dianalisa, meletakkannya pada (18). Mengalikan dalam faktor dan total didalam sintesis total, yaitu pada contoh 59.5.

*Mempersiapkan fungsi khusus.* Perhitungan bagian ini menunjukkan waktu yang akan dihabiskan dalam penulisan terkini, pengolah kata, proofreading (percobaan membaca) dan memeriksa FS. Pada (20) kami mengatur untuk persiapan sebagai hasil dari 5 (dari baris (5)) wawancara; pada (21) untuk 35 (dari (12)) pengadaan dokumen dan sistem, pada (22) total sintesis (dari (19)). Letakkan apapun yang dapat kamu pikirkan dalam (23), Kalikan hal itu semua dan tambahkan, yang mana akan kembali pada 26 dalam baris (24).

*Presentasi.* Memperhatikan setiap faktor dalam (25) karena mereka menanyakan beberapa pertanyaan, setiap bagian lokasi (26), mengabulkan permintaan (27) (Saya akan menambahkan yang satu ini !) dan faktor dalam slide atau chart yang telah dibuat untuk dipresentasikan. Perkalian dan penjumlahan ada pada baris (30). Total analisis keseluruhan adalah jumlah dari subtotal pada baris (5),(12),(19),(24), dan (29).

**Kesimpulan dari penggunaan analisa estimating chart.** Dengan jelas chart ini tidak akan memberi anda estimasi yang akurat seperti bidangnya. Bagaimanapun juga, ia tetap berguna, jika anda membuat faktor yang bekerja dalam aplikasi anda. Jika tak ada, ia menyajikan sebuah pilihan atau WBS untuk semua sktifitas yang terlibat dalam estimasi.

Ini sangat penting untuk dapat menganalisa dengan baik. Jika anda menggunakan dua fase proses proposal (lihat sesi 4.2) tahapan pertama menganalisa sendiri. Setiap DEC, dengan semua keahliannya dalam mengestimasi, lebih mengerjakan fase analisa dari sebuah proyek pada satu waktu dan basis materi (nilai plus).

## ATIOS

Jika anda cenderung mengerjakan tipe yang serupa dari proyek, anda akan menemukan bahwa proyek anda akan terdiri dari aktifitas utama yang serupa, dan tiap aktifitas akan utama menempatkan posisi yang sama untuk waktu dari proyek ke proyek. Oleh karena itu anda akan mengkalkulasi proporsi itu atau rasio untuk proyek yang lampau. Ini benar kecuali jika anda menemukan level tertinggi dari WBS untuk proyek yang cenderung sama. Misal, jika level 1 pada WBS dari proyek anda selalu terdiri dari 7 fase (yang seharusnya !) anda harus mempunyai ide yang bagus dari presentasi total usaha fase lain yang akan dibutuhkan. Contoh 13.10 adalah tabel dari industri rata-rata dari bagian kecil untuk ukuran tengah proyek komersial :

Phase	60% of Projects	90% of Projects
Definition	10%	40%
Analysis	20%	40%
Design	10%	40%
Programming	20%	20%
System Test	17%	40%
Acceptance	7%	40%
Operation	16%	40%

Figure 13.10 Ratio of effort in the 7 phases

Contoh 13.10, memperlihatkan bahwa hanya 60% dari proyek terpasang pada persentasi yang baik. Persentasi yang tinggi terdapat pada 40% PLAN, 20% BUID, 40% TEST ratio.

Pembolehan (Resferensi 3) terdapat untuk O/S 360 (proyek besar, sistem operasi pertama di dunia, penulisan di Assembler) hal ini terdapat 33% untuk PLAN tersebut, 17% untuk CODE tersebut, 25% untuk mengerjakan MODULE TEST AND LOW LEVEL INTEGRATION, dan 25 % untuk mengerjakan SYSTEM TEST. Membuat keadaan rasio dengan level lebih rendah dari WBS merupakan hal yang baik. Semisalnya, Gildersleeve (referensi 2 ) terdapat :

**WAKTU UNTUK DESIGN, DOKUMEN, & KODE MODEL SAMA DENGAN WAKTU DEBUGNYA**

### Bagaimana cara menggunakan rasio ?

Anda mungkin untuk sementara akan mengerjakan sebuah estimasi yang akurat untuk satu fase, dan memperhitungkan estimasi untuk fase sisa dengan menggunakan rasio lampau. Hal ini sangat tidak dipercaya, sejak kebanyakan proyek selalu berbeda. Bahkan, fase lain bisa lebih baik dari yang anda dapatkan, kemudian bandingkan rasio baru dengan lampau untuk melihat jika tidak terdapat item diluar aturan. Perbaiki estimasi untuk item tersebut, memberikan hal lebih baik untuk resiko dan kemungkinan yang lain.

## 13.5 DEC ( AND OTHER LARGE CORPORATIONS)

### ESTIMATING RULE OF THUMB

Apakah perusahaan besar seperti DEC menggunakan pendekatan tersebut? Ya, mereka menggunakan formulanya, tetapi menja ga untuk mengikuti kewajiban –kewajiban :

1. *Tak pernah bertanya tentang pengalaman seseorang untuk estimasi.* Penilaian merupakan permasalahan yang krusial pada pengalaman . Disini adalah bagaimana anda mengajar seorang yang baru untuk estimasi : Kemampuan kerja dengan pengalamannya merupakan estimasinya, atau pengalaman yang dimiliki dalam semua pekerjaan lain

merupakan formula detail dan memberikannya untuk seseorang yang baru sebagai “pemeriksaan”.

2. *Estimate dalam kelompok jika anda dapat mendorong kekuasaan.* Pertemuan dari beberapa pemikiran mengikuti kewajiban : dua pikiran adalah empat waktu yang efektif seperti yang pertama; tiga pikiran sepuluh waktu dan begitu seterusnya. Ketika saya mengajarkan proyek software manajemen untuk kelas, saya selalu masuk dalam bagian studi untuk penilaian dalam kelompok kecil. Saya menemukan sebuah proyek yang akan menempatkan satu orang disana selama empat hari untuk estimate sendiri sehingga dapat diestimasi oleh sebuah kelompok dari tiga atau empat orang dalam satu atau dua jam ! Hal yang menarik adalah kelompok yang dinamik. Anggota termotivasi menjadi produktif \_ tiap penekanan selain penurunan. Kelompok selalu mendapatkan ide yang besar, mengidentifikasi segala resiko dan setiap orang dalam kelompok akan bertahan dalam estimasi.
3. *Tak ada kekuatan estimasi pada profesional seperti seorang programmer.* Seorang supervisor seharusnya yang pertama bertanya pada programmer berapa lama contoh yang akan dapat ia ambil. Jika supervisor tidak setuju maka ia harus bernegosiasi.
4. *Tidak meletakkan rata-rata pada estimasi yang berbeda.* Untuk produk yang utama, mempunyai dua atau lebih pemisahan kelompok atau estimasi individu. Jika hal itu sangat berbeda antara estimasi, dapatkan seseorang bersama dan cari sebuah persetujuan. Hanya alasan dua orang dengan pengalaman yang serupa akan mengestimasi hal yang sama perbedaannya jika terdapat satu dari sesuatu yang terlupa.
5. *Terjadi penurunan dalam satu minggu atau lebih.* Sebuah software mempunyai tugas kira-kira satu minggu terlihat menjadi lebih pada seseorang dapat mengkonsep dan estimasi yang baik.
6. *Selalu Menambah(Perkalian?) untuk kesatuan.* Lihat sesi 2.4 tentang resiko manajemen.
  - Selalu menghasilkan suatu batas ketika memberikan suatu estimasi kepada manager atau client. Ada suatu cara psikologi yang menarik untuk mengutip sebuah estimasi. Jika Anda mengatakan pada manager bahwa hal ini akan memakan waktu selama 12 bulan, lalu manager akan berpikir bahwa Anda memiliki sebuah angka yang tepat; namun jika Anda mengatakan pada manager bahwa hal ini akan berlangsung selama 10 – 14 bulan, maka manager akan mengetahui bahwa itulah sebuah estimasi.
  - Gunakanlah perasaan Anda. Setelah Anda mengembangkan suatu estimasi, pikirkanlah dan tanyakanlah pada diri Anda sendiri, “Apakah estimasi ini benar ?” Hal ini sangat mengagumkan, dimana perasaan yang kuat dapat menjadikan segala sesuatu sebagai pengalaman yang berharga. Paling tidak, hal ini akan menjadi bahan pemikiran Anda kembali jika estimasi tersebut kurang tepat.

## PROSES ESTIMASI

Jika Anda mempunyai proyek manajemen software yang telah tersedia, Anda dapat mengembangkan WBS Anda pada software tersebut dan memasukkan beberapa estimasi untuk tugas-tugas yang ada di dalamnya. Susunan yang baik akan menambah berbagai macam estimasi dan kualitasnya, dan bahkan akan membawa tugas tersebut ke dalam suatu level di WBS. Jika Anda tidak memiliki susuna yang seperti itu, kembangkan dan gunakanlah formulir seperti di bawah ini :

PROJECT ESTIMATE SUMMARY	
Project _____	Date _____
Client _____	Author _____
Units _____	
Definition _____	
Analysis _____	
System Design _____	
Programming (DES, Doc, Code, Test) _____	
Acceptance _____	
Client Training _____	
Project Meetings _____	
Other ( _____ ) _____	
Total _____	

**Gambar 13.11** Project estimate summary

Pada gambar 13.11 terdapat suatu perkiraan untuk 7 (tujuh) fase dari suatu proyek, perkiraan ini sangat penting dan juga memiliki aktivitas, seperti persiapan ATP, manual user, client training, manajemen proyek, dan meeting proyek. Summary form tersebut akan menjadi sempurna jika sudah Anda lengkapi sesuai dengan keperluannya. Anda dapat memulainya dengan menggunakan formulir, yang sudah mengelompokkan item-item yang diperlukan dalam suatu proyek, seperti pada gambar 13.12.

Task Estimate Detail For Project				
Item _____				Author _____
Date _____				WBS Reference _____
TASK	DESCRIPTION LX	CMP DYS	EST	COMMENTS
TOTALS				

**Gambar 13.12** Task estimate detailed

Bagian dari CMPLX merupakan suatu bagian tugas yang lengkap : mulai dari tingkat rendah, menengah dan tinggi. Untuk menganalisa perkiran yang ada, gunakan bagan dalam gambar 13.9. Untuk tugas pemrograman, master program tersebut harus selalu di design, didokumenkan, di beri code dan di test. Sehingga formulir seperti pada gambar 13.13 dapat digunakan.

Programming Estimate Detail for Project					
Item _____				Author _____	
Date _____				WBS Reference _____	
TASK	DESCRIPTION	CMP LX	ESTIMATES	COMMENTS	
			DES DOC CODE TEST		
TOTALS					

**Gambar 13.13** Programming task estimate detail

Formulir yang detail mungkin akan disempurnakan oleh formulir yang lebih detail yang lainnya, dan hal ini tidak memiliki batas tertentu.

## **KESIMPULAN**

Estimasi, yang merupakan judul dari bab ini, adalah gambaran dari kelemahan Kita. Baik kelemahan dalam perencanaan maupun pengontrolan sesuatu, dimana kedua kelemahan tersebut sangat bergantung pada kemampuan estimasi Kita.

Estimasi bersifat iterative atau pengulangan, dan suatu estimasi sangat diharapkan untuk mengalami perubahan. Itulah sebabnya, mengapa Kita harus meletakkan suatu tanda dalam pengembangan sebuah produk. Tanda-tanda tersebut akan membatasi Kita, dalam memperkirakan berapa lama waktu untuk mengembangkan sebuah produk tersebut.

Estimasi merupakan sebuah seni. Tidak ada software yang tepat atau peralatan manual yang dapat membantu Kita pada saat sedang melakukan suatu perkiraan/estimasi. Kuncinya adalah dengan mengelompokkan sesuatu ke dalam bagian yang lebih kecil. Perlu diketahui juga bahwa data statistik sangat membantu Kita dalam hal ini. Jika suatu tugas berada dalam susunan yang baik dan jika Anda mencari sebuah perkiraan yang baik untuk tugas tersebut, maka beberapa di antara Kalian akan menemukan suatu keberhasilan dalam tugas tersebut dan sebagian lagi akan menemukan kegagalan. Pada akhirnya semua tugas tersebut akan menghasilkan sesuatu yang sama rata dan hal ini hanyalah akhir dari suatu permasalahan.

**TUGAS**  
**PENGELOLAAN PROYEK SISTEM INFORMASI**

**BAB 13**  
**ESTIMASI**

**DI SUSUN OLEH :**

**KELOMPOK VIII**

**AMALIA NUGRAHATI (10197252)**  
**ARIE BRAMANTYA (10197425)**  
**MERRY NANCYLIA PERMANA SARI (12197928)**  
**M. RIDWAN EFFENDI (13197414)**  
**NIA KURNIATI (12197206)**  
**PERDIANA (12197371)**  
**PRIYO BUDIONO (12197406)**  
**RATMA APRIANTO (12197512)**

**KELAS: 3KA-03**

**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS GUNADARMA**