



Sesi 9 - Lecture

Kebijakan Investasi SI/TI

Cakupan bahasan

- Pembuatan prioritas pengembangan aplikasi SI
- Metode keselarasan dengan CSF
- Metode Parker
- Metode Use Case Points

Kebijakan Investasi dan Penentuan Prioritas

- Meskipun kadang-kadang investasi IT dihitung sebagai “biaya” pada tahun ybs, sebenarnya manfaat IT tidak hanya pada tahun itu saja, namun terakumulasi sampai tahun-tahun berikutnya. Jadi lebih mirip sebagai “investasi”.
- Hardware acapkali dihitung sebagai aset, sedangkan development aplikasi software dihitung sebagai biaya. Padahal sebenarnya aplikasi software itulah yang bernilai sebagai “aset”.
- Problem lain dari investasi IT adalah masalah menentukan prioritas. Jangan sampai aplikasi yang tidak penting dibuat terlebih dahulu sehingga menghasilkan investasi balik yang lebih kecil.
- Yang menjadi penentu setelah kita membuat prioritas adalah ketersediaan sumber daya, dan acap kali adalah masalah ketersediaan sumber daya manusia baik dari kuantitas maupun kualitas

Generic IT Benefits (Remenyi, 1985)

- Pembagian berdasarkan pengaruh:
 - **1. *Tangible benefit:***
keuntungan yang secara langsung mempengaruhi profit perusahaan.
 - **2. *Intangible benefit:***
keuntungan yang memiliki efek positif pada perusahaan, tetapi tidak secara langsung mempengaruhi profit perusahaan.
- Pembagian berdasarkan keterukuran:
 - Quantifiable: *bisa terukur secara objektif*
 - Unquantifiable: *sulit terukur secara objektif*

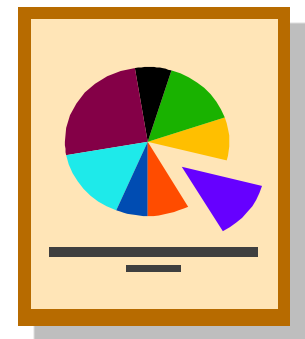
Remenyi (1985)

- Karena itulah, pembagian dapat dijadikan:
 - Quantifiable tangible
 - langsung mempengaruhi profit perusahaan dan efeknya bisa diukur secara objektif
 - Unquantifiable tangible
 - langsung mempengaruhi profit perusahaan tapi sulit diukur langsung
 - Quantifiable intangible
 - bisa diukur scr objektif tapi tidak mempengaruhi langsung profit perusahaan
 - Unquantifiable intangible

		<i>Objectively Measureable / Quantifiable</i>	
		High	Low
Tangible	High	Quantifiable-Tangible: - Pengurangan jumlah karyawan ■ Pengurang kebutuhan working capital ■ Peningkatan volume penjualan ■ Pengurangan kebutuhan biaya operasional Teknik: Cost benefit analysis	Unquantifiable-Tangible: - Informasi yang lebih baik Teknik: Management scoring/ranking
	Low	Quantifiable-Intangible: - kecepatan mendapatkan informasi - meningkatkan kepuasan pelanggan - meningkatkan kinerja karyawan Teknik: Opinion surveys	Unquantifiable-Intangible: - persepsi pelanggan - reaksi pasar Teknik: Market surveys

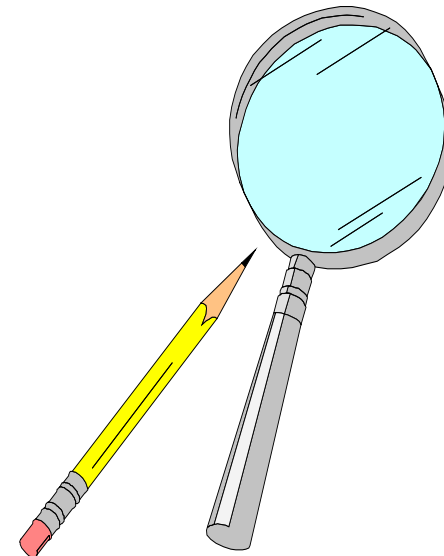
Survey tentang Investasi TI

- Cooke and Parish (1992): 70% perusahaan tidak ada justifikasi formal dan evaluasi investasi TI.
- Farbey, Land dan Targett (1992): hanya 50% dari proyek dilakukan penilaian pra-investasi.
Kurang dari 25% mempergunakan metode yang formal yang dikenali.
Hanya 30% yang melakukan penilaian hasil proyek.
- Ballantine, Galliers dan Stray (1994) serta Willcocks and Lester (1994): teknik analisis finansial tradisional masih sering dipergunakan, tetapi semakin sulit dipergunakan untuk melakukan kuantifikasi.
- Hochstrasser (1990), Peters (1990) dan Symons (1994): teknik berbeda untuk proyek yang berbeda.
- Lincoln & Shorrock (1990): “strategic” IS/IT investment sering *tidak* menggunakan justifikasi investasi formal.
- Grindley (1991):
 - 83% manager TI mengakui bahwa bahwa analisis atas keuntungan proyek TI adalah fiktif
 - Kata seorang CEO, “Justifikasi investasi IT ibarat konspirasi besar-besaran untuk membesar-besarkan keuntungan”



Beberapa isu-isu

- Perlu disadari bahwa ada cost untuk investasi teknologi infrastruktur. Jadi kalau mau membeli jaringan atau komputer server baru, harus bisa dilihat kapasitas yang akan disediakan dan keuntungannya secara makro.
- Juga ada problem dimana investasi IT dihitung nilainya terdepresiasi habis-habisan, tetapi sebenarnya masih bernilai.
- Kadang kala juga, tidak seluruh biaya dimasukkan ke dalam akunting, misalnya biaya spesifikasi sistem dan uji coba.



Information Economics

- Parker, Benson dan Trainor (1988) membagi aplikasi berdasarkan keuntungannya:
 - Subtitutif: mesin menggantikan manusia, fokus pada efisiensi. Lebih cepat.
 - Misalnya: penghitungan nilai mahasiswa menggunakan komputer akan jauh lebih cepat ketimbang perhitungan manual.
 - Komplementer: meningkatkan produktifitas dengan memungkinkan bekerja dengan cara yang baru sehingga lebih efektif.
 - Misalnya: pengisian formulir menggunakan kertas, kini digantikan oleh pengisian formulir melalui web-based application. Prosesnya bisa berubah, mungkin sebagian entry checking dilakukan oleh komputer.
 - Inovatif: agar memiliki keunggulan kompetitif dengan cara mengubah business model perusahaan, dsb.
 - Misalnya: melakukan distance learning berbasis TI, melakukan integrasi vertikal karena memanfaatkan SCM, dsb.

Cara Mengevaluasi Keuntungan

■ Cost benefit analysis: efisiensi karena adanya otomatisasi.

- Misalnya: menghilangkan biaya data entry. Arahnya ke cost saving.

■ Value linking: menaksir peningkatan performa bisnis karena membbaiknya hubungan antar proses/aktifitas.

- Contoh dari business performance adalah:
 - berkurangnya tagihan macet
 - meningkatnya kepuasan pelanggan

■ Value accelleration: benefit karena lebih cepat

■ Value restructuring: produktifitas akibat perubahan organisasi dan restrukturisasi. Jadi orang bekerja lebih sesuai fungsinya.

- Misalnya seorang manager dengan dibekali aplikasi BSC, akan bekerja lebih baik sebagai seorang manager. Dia tidak dibebani pekerjaan mencari data-data untuk memonitor kinerjanya, karena sudah tersedia pada aplikasi BSC.

■ Innovation revolution: berusaha melakukan evaluasi berdasarkan bisnis baru yang dikembangkan dengan bantuan IS/IT.



Information Economics

	Substitutive (Efficiency)	Complementary (Effectiveness)	Innovative (Competitive)
1. Cost / Benefit	✓	✓	✓
2. Value Linking	✓	✓	✓
3. Value Acceleration	✓	✓	✓
4. Value Restructuring		✓	✓
5. Innovation Evaluation			✓

Support (under Substitutive)
 High Potential (under Innovative)
 Key Operational (under Substitutive and Complementary)
 Strategic (under Complementary and Innovative)

Figure 10.1 Relationship between benefits and application types

Keterukuran investasi TI

- Meskipun penting untuk melakukan kuantifikasi secara finansial, perlu kita sadari bahwa tidak mungkin membuat seluruh faktor terkalkulasi secara finansial.
- Nanti effortnya berlebihan untuk membuat kalkulasi finansial. Padahal ada beberapa pendekatan lain.
- Sebagai contoh untuk peningkatan “moral” pegawai, ini bisa dilakukan dengan survey.
- Yang penting adalah: investasi dan keuntungannya dapat terukur.

Pendekatan Portofolio Ward

- Justifikasi kuantitas lebih mudah pada key operational dan support
- Satu macam teknik justifikasi hanya akan menghasilkan satu macam aplikasi saja.
- Masalah manajemen TI juga akan mempengaruhi: misalnya aplikasi yang terintegrasi pada lapisan corporate, atau aplikasi yang hanya memenuhi kebutuhan lokal dari tiap divisi.

Portofolio Justifikasi Investasi TI

STRATEGIC	HIGH POTENTIAL
<p>Enable the achievement of <i>business objectives</i> via explicit</p> <p>£ Critical Success Factors.</p>	<p>R + D project to explore potential value and cost - fund from R & D budget.</p> <p>£ <i>Risk money.</i></p>
<p>Disadvantage/Risk if it is not done (<i>Critical Failure Factors</i>) +/or</p> <p>£ quantified performance improvement.</p>	<p>Net cost reduction through <i>quantified savings.</i></p> <p>£</p>
KEY OPERATIONAL	SUPPORT

£ extent to which benefits can be justified financially

Support

- masalah efisiensi
- potensi keuntungan dapat teridentifikasi sebelum investasi dengan mudah
- aplikasi harus bisa menunjukkan keuntungan yang nyata
- asumsi: scarce resource strategy

Key Operational

- yang paling “ekonomis” belum tentu yang paling efektif. Bisa jadi “menyewa” lebih murah dalam waktu pendek. Tapi kalau penting sekali bagi core business?
- Harus ada feasibility study yang mendalam dari segi biaya, keuntungan dan resiko.
- bisnis bisa rugi kalau tidak ada key operational application: critical failure factor
- mungkin lebih baik kalau dibuat integrated application, sehingga bisa menjadi basis untuk aplikasi strategis
- monopoly: central control

Strategic

- alasan untuk dibuat banyak yang “intangible”, tetapi dapat dilihat dari CSF
- central planning: strategi perusahaan dibahas bersamaan dengan strategi IT-nya.
- oke kalau ternyata aplikasi bisa mendukung tujuan dan strategi perusahaan.
- penentunya adalah steering group/management
- jadi kesuksesannya adalah bagaimana mengalokasikan sumber daya dengan efisien untuk membuat aplikasi strategis dalam waktu yang optimal.

High potential

- keuntungannya tidak jelas.
- perlu evaluasi yang jelas dalam sebuah TOR proyek 'research'.

Faktor penentu prioritas

- apa keuntungannya
- sumber daya yang tersedia
- resiko kegagalan pengembangan yang ada

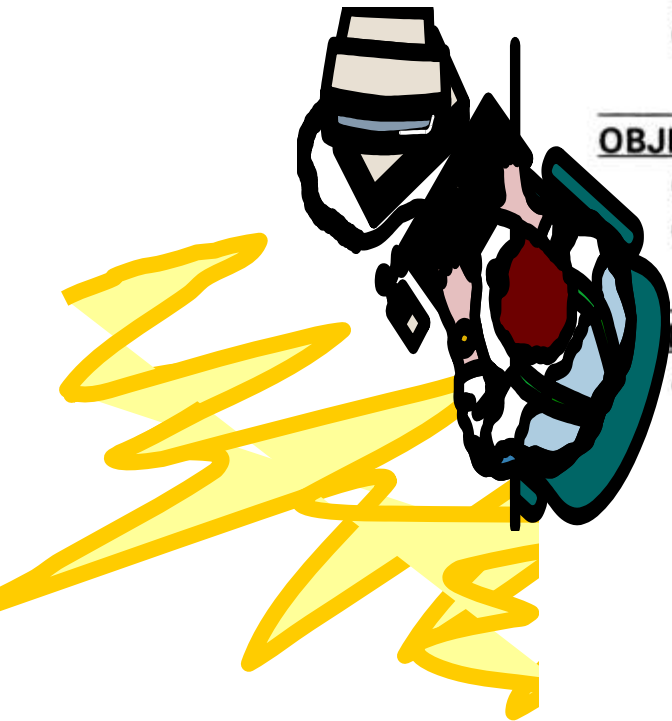
Penentua Prioritas Pada Portofolio Aplikasi Ward

- Support
 - masalah cost menjadi penentu prioritas
- Strategic
 - Aplikasi yang memberikan kontribusi bisnis paling besar dan menggunakan resource paling sedikit haruslah yang didahulukan.
 - Ini bisa dilakukan dengan membuat matrix keputusan per aplikasi “strategic”.
 - Dan hasilnya jangan ditafsirkan leterlek: skor 25 dan skor 24 bukan berarti yang 24 lebih jelek dari yang 25, tetapi kurang lebih memiliki signifikansi yang sama.
 - Kalau kita membaginya dengan *jumlah resource yang dibutuhkan*, maka akan didapatkan prioritasnya dengan lebih baik...



- **Key Operational**, argumentasi keuntungannya:
 - keekonomisan
 - CSF
 - resiko terhadap bisnis saat ini
 - perbaikan infrastruktur: standar yang sama, peningkatan ketrampilan SDM, peningkatan fleksibilitas infrastruktur, dsb.
- **High Potential**
 - Kalau ada ide yang “kena” ke CSF, harus dibawa keluar dari R&D dan dicoba apakah bisa jadi strategic application.

Metode CSF



OBJECTIVE A:
CSF 1
CSF 2
CSF 3 etc.

OBJECTIVE B:
CSF 1
CSF 2 etc.

OBJECTIVE C:
CSF 1
CSF 2 etc.

OBJECTIVE D:
etc.

TOTALS

**OVERALL
TOTAL**

APPLICATION CONTRIBUTION		
HIGH (3)	MEDIUM (2)	LOW (1)

Figure 10.3 Strategic weighting via critical success factors

Penentuan Prioritas Pengembangan Aplikasi dengan Metoda Kesesuaian CSF

- Bagaimana memprioritaskan aplikasi yang hendak dibangun?

LEGENDS

Application Legends	
1	Balance Scorecard
2	Document Management
3	Corporate Portal
4	Knowledge Management
5	Quality Control Module
6	Corporate Websites
7	Database Outlet
8	Special Interest Group Websites
9	Supply Chain Management
10	Supplier Rating
11	Product & Formula Development Application
12	Customer Relationship Management
13	Customer Response Center

Application Roles	
0	None
1	Low
2	Medium
3	High

Bobot yang akan digunakan setiap aplikasi pd setiap CSF

Pembobotan Aplikasi pada CSF

Objectives & CSF	Weight	Application Roles												
		1	2	3	4	5	6	7	8	9	10	11	12	13
Peningkatan PV SBU	10	3	1	1	1	2	2	2	2	2	2	2	2	2
Sales ekspor mencapai 30 % dari total sales	9	1	2	0.5	0	0.5	0.5	1.5	0.5	1.5	1	1	1	1
<i>Merrespons kebutuhan target pasar dengan cepat</i>		1	1	1	0	1	1	3	1	3	2	2	2	2
<i>Menyempumakan system prosedur yang berkaitan dengan International Market</i>		1	3	0	0	0	0	0	0	0	0	0	0	0
Pertumbuhan sales sebesar 60% setahun	9	2	0.67	0	0.67	1	0	0.33	0	0	0	1	0	0
<i>Peninjauan BAM</i>		2	1	0	0	0	0	1	0	0	0	0	0	0
<i>Mengoptimalkan dan menyesuaikan kapasitas</i>		2	0	0	0	1	0	0	0	0	0	0	0	0
<i>Fokus kepada produk potensial</i>		2	1	0	2	2	0	0	0	0	0	3	0	0

⋮

⋮

Terciptanya organisasi pembelajar yang inovatif	7	1	0	2	0	0	0	0	0	0	0	0	0	0
<i>Sosialisasi budaya cocotefasera</i>		0	0	3	0	0	0	0	0	0	0	0	0	0
<i>Meningkatkan kecepatan, fleksibilitas dan komitmen</i>		3	0	2	0	0	0	0	0	0	0	0	0	0
<i>Mendorong kelompok-kelompok belajar yang mandiri</i>		0	0	1	0	0	0	0	0	0	0	0	0	0
TOTAL WEIGHTED SCORE		192	143	76.2	122	180	162	133	155	99.5	47	171	189	171

Prioritas berdasarkan value

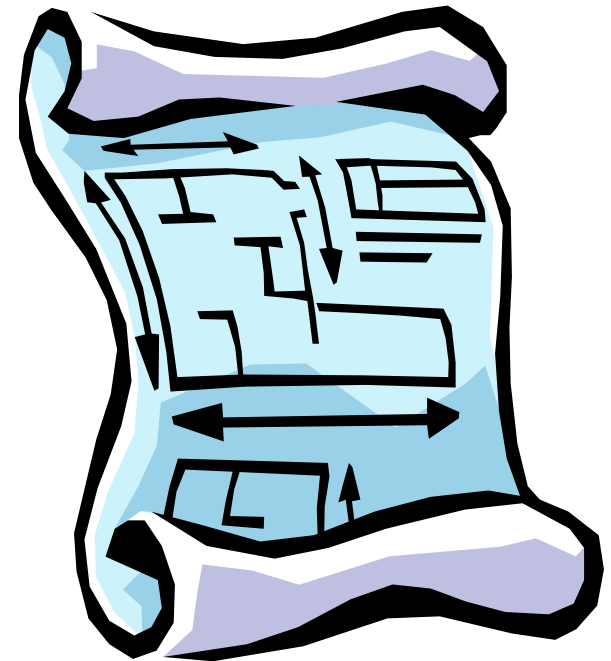
- Jika dirasakan perlu, dan jika sudah didapatkan biaya (estimte) pengembangan aplikasi, maka kita dapat mencari *value* dari aplikasi
- Value = Skor bobot berdasarkan CSF / biaya
- Lalu kita dapat melakukan ranking berdasarkan 'Value' tersebut

Pengaruh konteks bisnis pada prioritas investasi aplikasi

Faktor	Penekanan bobot			
	CSF	Resiko bisnis	Infrastruktur	Keekonomisan
Semua investasi harus menghitung ROI	↓	↓	↓	↑
Bisnis lagi agak menurun, sedang butuh short term profit	↓	↔	↓	↑
Bisnis sedang tumbuh dengan cepat	↑	↑	↔	↓
Pasar sangat kompetitif	↑	↔	↓	↑
Peremajaan teknologi	↔	↔	↑	↓
Butuh sistem baru untuk mendukung perubahan usaha/organisasi	↔	↑	↔	↓

Pentingnya Prioritas Dibuat Formal

- realokasi sumber daya secara beralasan (logis)
- perencanaan sumber daya yang lebih tepat untuk masa depan



Faktor waktu, kualitas & biaya

STRATEGIC	HIGH POTENTIAL
TIME Quality Cost	(R + D PROJECTS)
Time QUALITY Cost	Time Quality COST
KEY OPERATIONAL	SUPPORT

Therefore risk factors must be considered in the context of how they affect

	STRATEGIC	KEY OP	SUPPORT
TIME	***	**	*
QUALITY	**	***	**
COST	*	**	***

Jenis, faktor & potensi implikasi resiko

Jenis	Faktor	Strategic	Key-Op	Support
Manusia	Keikutsertaan manager senior	↑	↔	↓
	Pengetahuan anggota tim proyek ttg bisnis	↑	↔	↓
	Kemampuan teknis dan pengalaman tim	↔	↑	↔
	Koordinasi antara staf bisnis dan teknis	↑	↔	↔
Ukuran	Jumlah SDM yang diperlukan	↑	↑	↑
	Lamanya proyek	↑	↔	↑
Kontrol proyek	Penggunaan methodology pengembangan yang standar dan formal	↔	↑	↔
	Kontrol terhadap pengujian dan revisi	↔	↑	↔
	Kontrol anggaran	↓	↔	↑
Novelty	Perubahan organisasi	↔	↔	↑
Stability	Kejelasan ruang lingkup	↔	↑	↑

- High risk (↑↑): proyek pasti tidak berhasil kalau tidak ada usaha untuk menangani resiko sebelum proyek berjalan.
- Medium risk (↔): perlu ada contingency plan kalau resiko muncul
- Low risk (↓↓): tidak ada resiko dalam kondisi normal

Implementasi Manajemen Resiko

No.	Description of Threat & Vulnerability	Probability of Threat & Vulnerability (angka 1 s/d 10)	Impact to Project (angka 1 s/d 10 & deskripsi)	P x I	Counter-measures or Control Procedures
		5 Hampir pasti 4 Mungkin 3 Netral 2 Kurang mungkin 1 Tidak mungkin	3 Berat 2 Sedang 1 Ringan		
1.	Teknologi hanya dikuasai oleh vendor	Mungkin (4)	Sedang (2), karena ada teknologi substitusi dari vendor lain	8	Buat rencana alternatif menggunakan berbagai teknologi
2.	Aturan pemerintah yang berubah-ubah mempengaruhi proses bisnis	Mungkin (4)	Berat (3)	12	Menggunakan workflow engine yang mudah diubah



Menaksir Biaya Pengembangan Aplikasi

Teknik Use Case Point

Steps for UCP estimation

1. Determine the UAW (Unadjusted Actor weight)
2. Determine number of UUCW (Unadjusted Use case Weight)
3. Determine Total UUCP (Unadjusted Use Case Point)
4. Computing technical and environmental factor
5. Calculating the Adjusted Use Case Points
6. Mengkalkulas man-hours dan biaya total

Determine Unadjusted Actor Weight

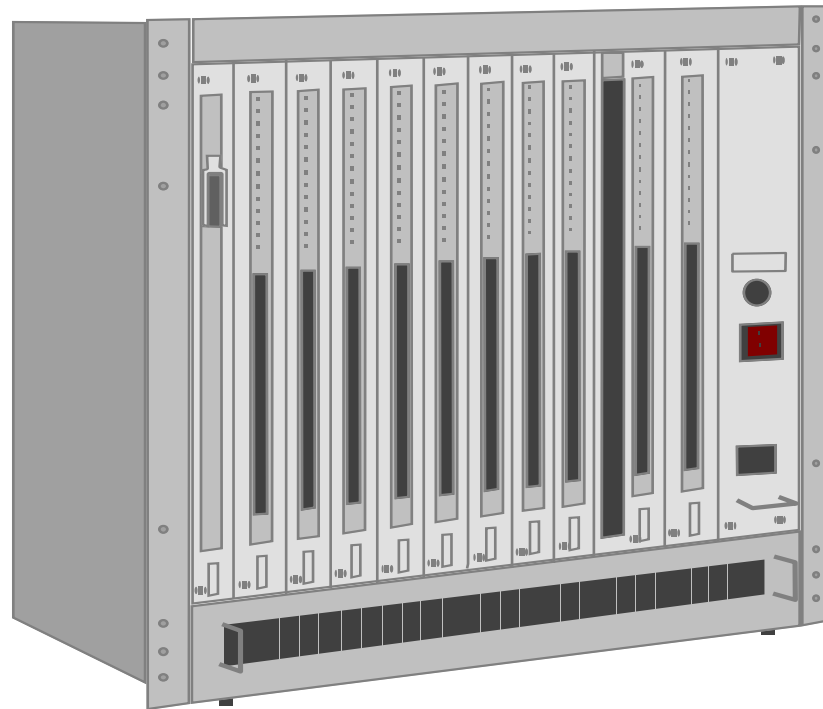
Classification	Litmus test to recognize classifications	Factor
Simple actors	Simple actors are those which communicate to System through API.	1
Average actors	Average actors are recognized if they following properties <ul style="list-style-type: none">○ Actors who are interacting the system through some protocol(HTTP,FTP, or probably some user defined protocol)○ Actor which are data store(Files, RDBMS)	2
Complex	Complex actor is interacting normally through GUI.	3

Determine number of Unadjusted Use case Weight

Use case Type	Litmus test to recognize classifications	Factor
Simple	Greater than or equal to 3 transactions	5
Average	Between 4 to 7 transactions	10
Complex	Greater than 7 transactions	15

Determine Total Unadjusted Use Case Point

- Total UUCP = Total UAW + Total UUCW



Computing Technical Factors

Code	Technical factor	Weight	Description
t1	Distributed System	2	Is the system having distributed architecture or centralized architecture?
t2	Response time	1	Does the client need the system to fast? Is time response one of the important criteria?
t3	End user efficiency	1	How's the ends users efficiency?
t4	Complex Internal Processing	1	Is the Business process very complex ?. Like complicated accounts closing,Inventory tracking,heavy tax calculation etc
t5	Reusable Code	1	Do we intend to keep the reusability high. So will increase the design complexity.
t6	Installation Ease	0.5	Is client looking for installation ease?.By default we get many installers which create package. But if the client is looking for some custom installation probably depending on module wise .One of our client has requirement that when the client wants to install he can choose which modules he can install. If the requirement is such that when there is a new version there should be auto installation. These factors will count when assigning value to this factor.
t7	Easy use	0.5	Is user friendly at the top priority?
t8	Portable	2	Is the customer looking for also cross platform implementation?
t9	Easy to change	1	Is the customer looking for high customization in the future? So that also increases the Architecture design complexity and hence this factor.
t10	Concurrent	1	Is the customer looking at large numbers of users working with locking support. This will increase the architecture complexity and hence this value.
t11	Security objectives	1	Is the Customer looking at having heavy security like SSL or have to write custom code logic for encryption.
t12	Direct access to third parties	1	Does the project depend in using third party controls. So for understanding the third-party controls and studying its pros and cons considerable effort will be required. So this factor should be rated accordingly.
t13	User training facilities	1	Will the software from user perspective be so complex that separate training has to be provided. So this factor will vary accordingly.

- All technical factor will be assigned a value from 0 to 5 depending on complexity.
- Equation for Tfactor = $\text{sum}(T1....T13)$
- TCF(Technical Complexity Factor) : $\text{TCF} = 0.6 + (0.01 * \text{Tfactor})$

Computing Environmental Factors

Code	Environmental Factor	Weight	Description
e1	Familiarity with project	1.5	Are all the people working in the project familiar with domain and technical details of the project?. So probably you will spend your most time in explaining them all know-how's.
e2	Application experience	0.5	How much is the application experience?
e3	Objects-oriented Experience	1	As use-case documents are inputs to Object oriented design. Its important that people on the project should have basic knowledge of OOP's concept.
e4	Lead analyst capability	0.5	How the analyst who is leading the project?. Does he have enough knowledge of the domain?
e5	Motivation	1	Are the programmers motivated for working on the project. As instability in project will always lead to people leaving half way there source code. And the hand over becomes really tough. This Factor you can put according to how software industry is going on? Example if the software market is very good put this at maximum value. As good the market more the jobs and more the programmers will jump.
e6	Stable requirements	2	Is the client clear of what he wants?. I have seen clients expectations are the most important factor in stability of requirements. If the client is of highly changing nature put this value to maximum.
e7	Part-Time Staff	-1	Are there part-time staffs in project like consultants etc?
e8	Difficult programming language	-1	How the language complexity Assembly, Vb6, c++, c etc

- All environmental factor will be assigned a value from 0 to 5 depending on complexity.
- Efactor = SUM(e1...e8).
- Calculating Environmental Factor = $EF = 1.4 + (-0.03 * Efactor)$

Adjusted Use Case Points & Biaya Akhir

- AUCP (Adjusted Use Case Points) =
$$\text{UUCP} \times \text{TCF} \times \text{EF}$$
- Karner proposed a factor of 20 staff hours per use case point for a project estimate.
- Schneider and winters mengatakan bahwa bisa bervariasi antara 20 s/d 36 tergantung pada EF (tidak dibahas di sini)

Contoh Kasus: TNC Customer's Credit Card Data Entry

- TNC company till now was using manual way of maintaining its customer database and there credit card information.
- Data entry operator manually validates credit card information from external payment gateway.
They maintain Customer Code, Customer Name, Customer Address, Customer phone and validated Customer Credit card information in Customer registry.
- Customer Code is unique for a customer, so TNC manually check for the validations and enters in the customer registry.
- TNC wants the data entry project to be automated.

Spesifikasi

- Customer Code assigned should be checked for uniqueness automatically.
- Customer Code should not exceed 8 length.
- Credit card validation should be automatic for the current System. TNC has already given the API documentation of how to interact with the third party payment system.
- Credit card length should not exceed more than 10 length.
- Data entry operator should be able to add/update/Delete customer information.
- The database will be in the TNC head office and only data entry operators will be allowed to use the Data entry Software.
- Software should work on Windows platform. At this moment TNC has Windows 2000 client installed in all computers.

Use Case Document

Use Case #	DATAENTRYPROJECTCUST-1009
Use Case Name	Maintain Customer
Description	This use case depicts full maintenance of customer from project "Data Entry".
Scope and Level	<ul style="list-style-type: none"> • Data Entry System (Internal) • Credit Card System (External)
Level	User Goal Level (If this property is not understood look at the reference for the book Writing Effective Use Cases (**PRE-PUB. DRAFT#3**) :Alistair Cockburn Humans and technology)
Primary and secondary actors	Data Entry operator.
Stakeholders and Interests	Trigger Data entry operator clicks on Menu "Add New Customer"
Preconditions	<ul style="list-style-type: none"> • Data entry operator should be logged in. • Data entry operator should have access to internet.
Assumptions	Customer information received is entered manually. No Automated Import routine is in the Scope
Failed End Condition	<ul style="list-style-type: none"> • Customer is not added to Database and appropriate error message is displayed. • Customer Code already existing in the Customer Database. • Customer Code length limit is exceeded. • Customer Credit Card limit is exceeded. • Customer Credit Card validation failed with the payment gateway.

	<ul style="list-style-type: none"> • Customer Credit Card limit is exceeded. • Customer Credit Card validation failed with the payment gateway.
Action	Add New Customer
Main success scenario (or basic Flow):	<ol style="list-style-type: none"> 1. Data entry operator receives customer information. 2. Data entry operator enters following information <ul style="list-style-type: none"> ○ Customer Code ○ Customer Name ○ Customer Address ○ Customer Phone 3. Customer Code is checked if it exists in Customer Table. <ul style="list-style-type: none"> ○ If the Customer Code is existing then "Duplicate Customer Code" error is raised. ○ If the Customer Code is more than 8 length then "Customer code length limit crossed" error is raised. 4. After step 3 is passed ok. Data entry operator enters Credit Card information. 5. If the credit card length is more than 10 length then "Credit card length limit crossed" error is raised. 6. Credit card information is send to the external payment gateway. Appropriate APIs of the external payment gateway will be used for validity. 7. External Payment Gateway returns "OK" if credit card is validated or else will return "NOT VALID" flag. 8. Data entry operator then Adds the customer in database.
Alternate scenario	Update Existing Customer

Alternate scenario (Extensions):	Update Existing Customer
	<ol style="list-style-type: none"> 1. Data Entry operator enter Customer Code to retrieve the customer which has to be updated. 2. Data Entry operator make appropriate changes to the customer information. All steps and business validation from to 6 of Add new Customer is repeated. 3. Data Entry operator update the customer information.
Alternate scenario (Extensions):	Delete Existing Customer
	<ol style="list-style-type: none"> 1. Data Entry Operator enters Customer Code to retrieve the Customer which has to be Deleted. 2. Data Entry Operator Deletes the Customer. Data Entry Operator is alerted "Are you sure you want to delete the Customer?" <ul style="list-style-type: none"> o If the Data entry operator clicks "Yes". Then the customer is deleted from the database. o If the Data entry operator click "NO" no action is taken.
Success Guarantee (Post conditions)	<ul style="list-style-type: none"> • Customer is added to Customer Database. • Customer is updated to Customer Database. • Customer is deleted from Customer Database.
Special Requirements (including Business rules):	
Technology and Data Variations List:	If Credit Card Payment Gateway API changes the interaction of the data entry customer module will have to changed accordingly.
Frequency of occurrence:	
Notes and Open Issues:	

Penerapan UCP

- Determining Unadjusted Use Actor Weights (UAW): In this project we have identified only one actor “Data Entry Operator”. The upper Actor (Data entry operator) is complex as data entry operator will be interacting through GUI. So $UAW=3$
- Determine number of UUCW (Unadjusted Use Case Weight): There are 12 transactions, (adding also the alternative flows). $UUCW=15$
- Total UUCP = $15 + 3 = 18$



Technical Factors

Technical factor	Weight	Value	Weighted Value	Explanation
t1 Distributed System	2	1	2	Simple two tier architecture is decided.
t2 Response time	1	4	4	Speed is of importance as the data entry operator have to enter data quiet fast.
t3 End user efficiency	1	3	3	Data entry operator has High user efficiency.
t4 Complex Internal Processing	1	2	2	Its simple entry screen and no business process has been scoped by the client. Only credit card check and duplicate customer code is the business check.
t5 Reusable Code	1	1	1	No reusability as project is small and customer is no looking for any further changes for at least two years.
t6 Installation Ease	0.5	0	0	TNC has good in house development team and installation problems will be handled by them. Technology thought is c# and .NET setup wizard will be enough to make the installation process easy
t7 Easy use	0.5	4	2	Yes data entry operator for fast entry of data has to have user friendly menus and shortcut keys.
t8 Portable	2	1	2	TNC has windows 2000 client as specified in the scope document.
t9 Easy to change	1	0	0	None specified by client
t10 Concurrent	1	0	0	Client has not clarified about this issue as such in the scope document. So assumed least concurrent.
t11 Security objectives	1	0	0	None specified by client. Even credit card information will be passed with out encryption.
t12 Direct access to third parties	1	3	3	Using the credit card check API
t13 User training facilities	1	0	0	The screen is simple and data entry operator can operate with out any training
Total			19	

Calculating the Technical Factor (TCF)

$$= 0.6 + (0.01 * Tfactor)$$

$$= 0.6 + (0.01 * 19)$$

$$= 0.79$$

Environmental Factor

	Environmental Factor	Value	Weight	Weight Column	Description
e1	Familiarity with project	5	1.5	7.5	It's a simple project so familiarity with project is not so much needed.
e2	Application experience	5	0.5	2.5	Its simple application.
e3	Objects-oriented Experience	5	1	5	Every one has well oops knowledge.
e4	Lead analyst capability	5	0.5	2.5	Its simple project no lead analyst needed til now.
e5	Motivation	1	1	1	Motivation is little down as programmers are reluctant to work on the project because of its simplicity.
e6	Stable requirements	4	2	8	Client is very clear with what he wants?
e7	Part-Time Staff	0	-1	0	No part time staffs
e8	Difficult programming language	3	-1	-3	C# will be used. And most of programming guys are new the C# technology.

Calculating EF
 $= 1.4 + (-0.03 * \text{Efactor})$
 $= 1.4 + (-0.03 * 23.5)$
 $= 0.695$

Penyelesaian Akhir

- Calculating AUCP = UUCP * TCF * EF = $18 \times 0.79 \times 0.695 = 9.88$ approx = 10 Use Case Points.
- According to Karner i.e 20 staff hours per use case points = $10 \times 20 = 200$ hours for the total project.
- If programmer works for 8 hours for a day then $340/8 = 25$ days.
- Lalu dikalikan dengan rate per man hours.



Costing Teknologi Informasi



Pengalokasian Biaya

Komponen Bisnis	Keuntungan	Kerugian
Service centre: dimana user tidak di-charge sedikitpun untuk penggunaan TI	<ul style="list-style-type: none"> - Menstimulasi eksperimen - Menghindari konflik organisasi - Mempromosikan penggunaan service TI 	<ul style="list-style-type: none"> - memungkinkan penggunaan yang tidak ekonomis - tidak accountable - Menghasilkan demand yang berlebihan dan tidak ada prioritas
Cost Centre: user dicharge agar biaya yang dikeluarkan untuk TI setidaknya tertutup.	<ul style="list-style-type: none"> - membuat user melakukan justifikasi investasi - bagian IT dapat terkontrol - membuat user sadar akan biaya - memungkinkan dilakukan prioritas 	<ul style="list-style-type: none"> - bisa membuat orang tidak pakai IT - fokus pada biaya, bukan pada keuntungan - sulit untuk menentukan sistem charging yang tepat
Profit centre: user di-charge dengan biaya recover plus "laba"	<ul style="list-style-type: none"> - IS/IT dapat mengontrol biayanya sendiri - IS/IT menjad proaktif - Mendorong pembuatan keputusan oleh user 	<ul style="list-style-type: none"> - user dapat mencari support IT dari luar/ external - resource bisa tidak optimal penggunaannya - IS/IT akan fokus pada pekerjaan yang 'menguntungkan'
Hybrid centre	<ul style="list-style-type: none"> - Memungkinkan beberapa jenis pengembangan TI - bisa mengakomodir teknologi baru - "pricing" bisa dijadikan sarana untuk menghasilkan aplikasi jenis tertentu 	<ul style="list-style-type: none"> - bisa membingungkan user - akuntingnya kompleks - kontrol IS/IT lebih lemah - Perlu monitor terus menerus - Bisa menyebabkan konflik.

Implikasi dari cara pembayaran

Cara charging	Tipe Manajemen	Pola IT	Aplikasi yang dihasilkan
Tidak ada charge	Leading edge	Service centre	High Potential
Average cost	Scarce resource	Cost centre	Support
Standard cost	Monopoly	Cost centre	Key Operational
Market price	Free market	Profit centre	Support, High potential
Flexible	Centrally planned	Hybrid centre	Strategic dan high potential

- Average cost: total cost dibagi pemakaian per user
- Standard cost: charging setiap kali pemakaian