10 MITOS

Software Engineering

Romi Satria Wahono
romi@romisatriawahono.net
http://romisatriawahono.net
08118228331

Romi Satria Wahono

- SMA Taruna Nusantara Magelang (1993)
- B.Eng, M.Eng and Ph.D in Software Engineering
  Saitama University Japan (1994-2004)
  Universiti Teknikal Malaysia Melaka (2014)
- Core Competency in Enterprise Architecture, Software Engineering and Machine Learning
- LIPI Researcher (2004-2007)
- Founder and CEO:
  - PT Brainmatics Cipta Informatika (2005)
  - PT IlmuKomputerCom Braindevs Sistema (2014)
- Professional Member of IEEE, ACM and PMI
- IT and Research Award Winners from WSIS (United Nations), Kemdikbud, Ristekdikti, LIPI, etc
- SCOPUS/ISI Indexed Q1 Journal Reviewer: Information and Software Technology, Journal of Systems and Software, Software: Practice and Experience, Empirical Software Engineering, etc
- Industrial IT Certifications: TOGAF, ITIL, CCAI, CCNA, etc
- Enterprise Architecture Consultant: KPK, RistekDikti, INSW, BPPT, Kemsos Kemenkeu (Itjend, DJBC, DJPK), Telkom, FIF, PLN, PJB, Pertamina EP, etc

RomiSatriaWahono.Net

Lecture Notes in Software Engineering, Computing Research and Technopreneurship

HOME ABOUT ME IN THE NEWS LECTURES

RomiSatriaWahono.Net

YouTube ID Search

Romi Satria Wahono

3.55K subscribers

HOME VIDEOS PLAYLISTS COMMUNITY CHANNELS ABOUT

Uploads PLAY ALL

Data Mining untuk Mahasiswa Galau

268 views • 9 hours ago

Menjadi Programmer Technopreneur

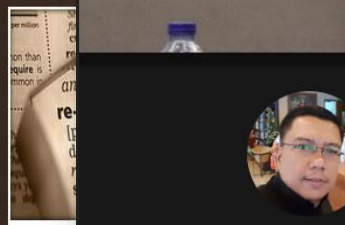
4.3K views • 5 years ago

Kuliah 10 Menit tentang Enterprise Architecture

10K views • 5 years ago

Kuliah 20 Menit tentang Metodologi Penelitian

136K views • 5 years ago



eluruh materi kuliah bisa diunduh dan
rse description, standard competency,

ed January 2015)
2015)
ed March 2015)

October 2013)

updated January 2015)
otation (updated January 2015)

BRAINDEVS

1 Hour Online Training

Software Engineering Research Trends

Romi Satria Wahono, Ph.D
Researcher & Technopreneur. Founder & CEO PT Braindevs & PT Brainmatics.
TOGAF & ITIL Certified. Enterprise Architect di KPK, Pertamina EP, BPPT, PLN, PJB, FIF, INSW, RistekDikti, Kemenkeu. SCOPUS Q1 Journal Reviewer:
Information & Software Technology, Journal of System & Software, etc.

GRATIS

Jum'at, 8 Mei 2020
09.00 - 10.00 WIB
Terbatas 500 Peserta, Registrasi Sebelum
Kamis, 7 Mei 2020, Pukul 17:00 Melalui
sert1hour.brainmatics.com

#TrainingDirumahAja

Contact Us:

Irma	08118228881
Annisa	08118228882
Vina	08118228883
Lina	08118228884
Rachma	08118228885

Menara Bidakara 1 Suite 0205
Jl. Gatot Subroto Kav 71-73 Jakarta 12870
info@brainmatics.com http://brainmatics.com
BrainmaticsID

BRAINMATICS

Software Myths

Mitos (Indonesia)

Myths (Inggris)

Mythos (Yunani)

Mythe (Belanda)

Cerita turun temurun sejak masa lampau, yang mengandung penafsiran tentang alam semesta, dan dianggap benar-benar terjadi oleh para pengikut dan penganutnya

10 MITOS SOFTWARE ENGINEERING

Mitos 1: Cara Sekarang Masih Manual, Karena Itu Butuh Software

Mitos 2: Kemampuan Terpenting Bagi Pengembang adalah Coding

Mitos 3: Kualitas Software Dinilai dari Teknologi yang Digunakan

Mitos 4: SDLC itu Waterfall dan Itu Buruk dan Sudah Kuno

Mitos 5: Project Molor? Tambah Jumlah Pengembang Supaya Bisa Cepat Selesai

Mitos 6: Penghitungan Cost Pengembangan Software itu Pakai Insting dan Pengalaman

Mitos 7: Software Engineering Itu Langsung Praktek Saja, Jangan Kebanyakan Teori

Mitos 8: Penelitian Software Engineering itu Hasilnya adalah Produk Software

Mitos 9: Ingin Membuat Software Seperti Yang Ada Sekarang

Mitos 10: Tidak Suka Software, Tapi Yakin Sukses Bisnis Software

MITOS 1

Cara Sekarang Masih Manual, Karena Itu Saya Akan Kembangkan Software

Kapan Project Software Diinisiasi?

- Ketika ada seseorang yang melihat peluang menciptakan business value dengan menggunakan software dan teknologi informasi
  - Seseorang itu kemudian membuat System Request

- System Request kemudian dianalisis kelayakannya (*Feasibility Analysis*) untuk menentukan apakah akan diteruskan projectnya atau tidak
  - Di dalam analisis kelayakan, dilakukan juga penghitungan usaha pengembangan software (butuh berapa bulan dan berapa orang)
    - sehingga analisis kelayakan ekonomi bisa dibangun dengan akurat

Siklus Pengembangan Software:

Alur, Peran, dan Tahapan (*Deliverable*) *(Tilley, 2012) (Dennis, 2016) (Valacich, 2017)*

1. **User/Product Owner** membawa permintaan kebutuhan (perubahan) software (System Request) ke System Analyst
2. **System Analyst** membuat analisis kelayakan (Feasibility Analysis) dari System Request tersebut

Planning
(System Proposal)

3. Setelah dinyatakan layak, System Analyst melakukan analysis dan design, menghasilkan System Specification (SRS, BRD, FSD, etc)
   - **Business Analyst** membantu System Analyst memahami proses bisnis dari software yang akan dibangun

Analysis and Design
(System Specification)

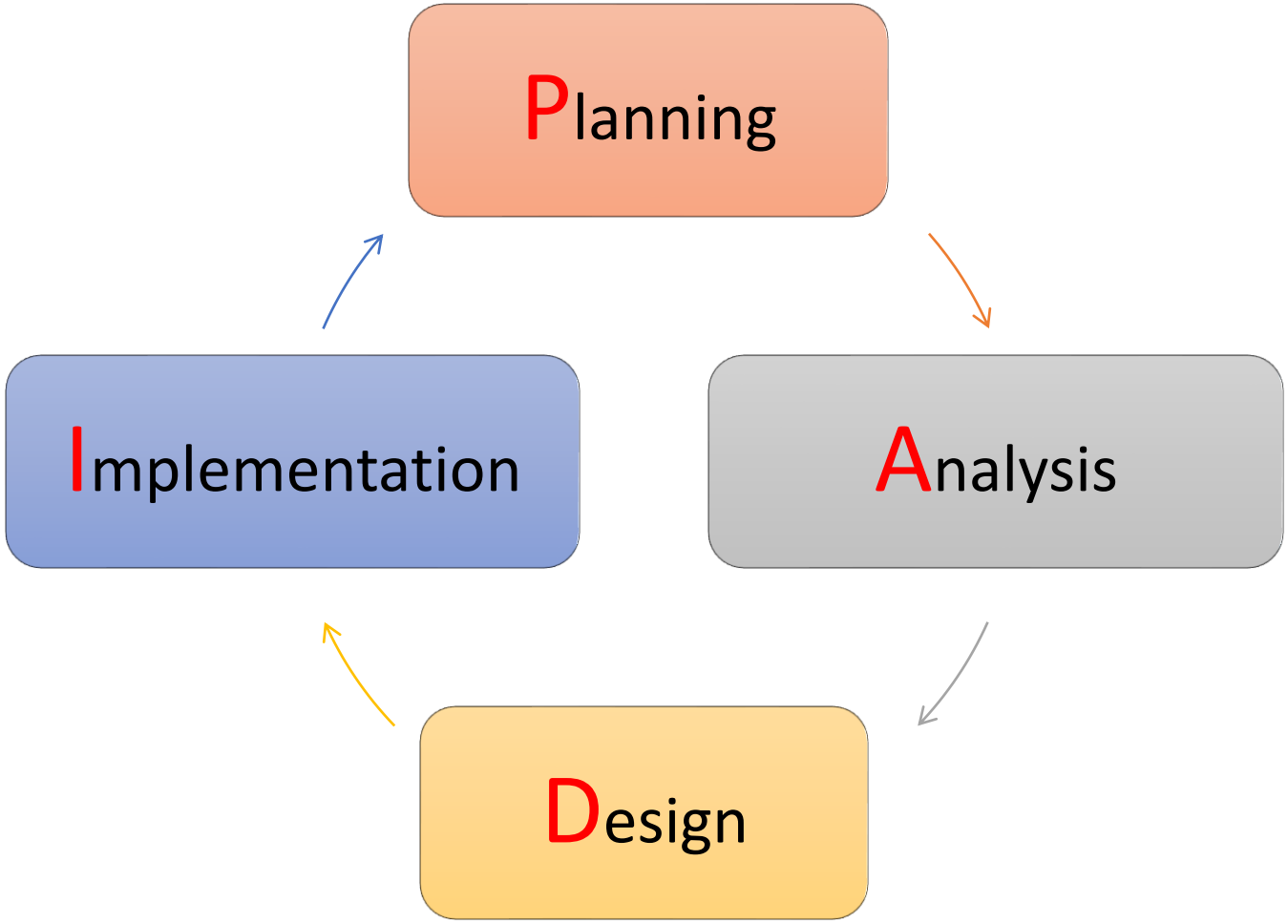
4. System Specification diserahkan oleh System Anayst ke **Programmer** untuk dilakukan Konstruksi (*Coding*)
5. Hasil Konstruksi berupa Kode Program diserahkan ke **Software Tester** untuk dilakukan Pengujian (*Unit, Integration, System, User Acceptance Testing*)
6. Instalasi (*delivery*) software dan manajemen perubahan
   - Software = Kode Program + Dokumentasi (Pengembangan dan Penggunaan)

Implementation
(Software)

7. Siklus kembali ke 1 apabila ada permintaan perubahan (Permintaan Perubahan Software)

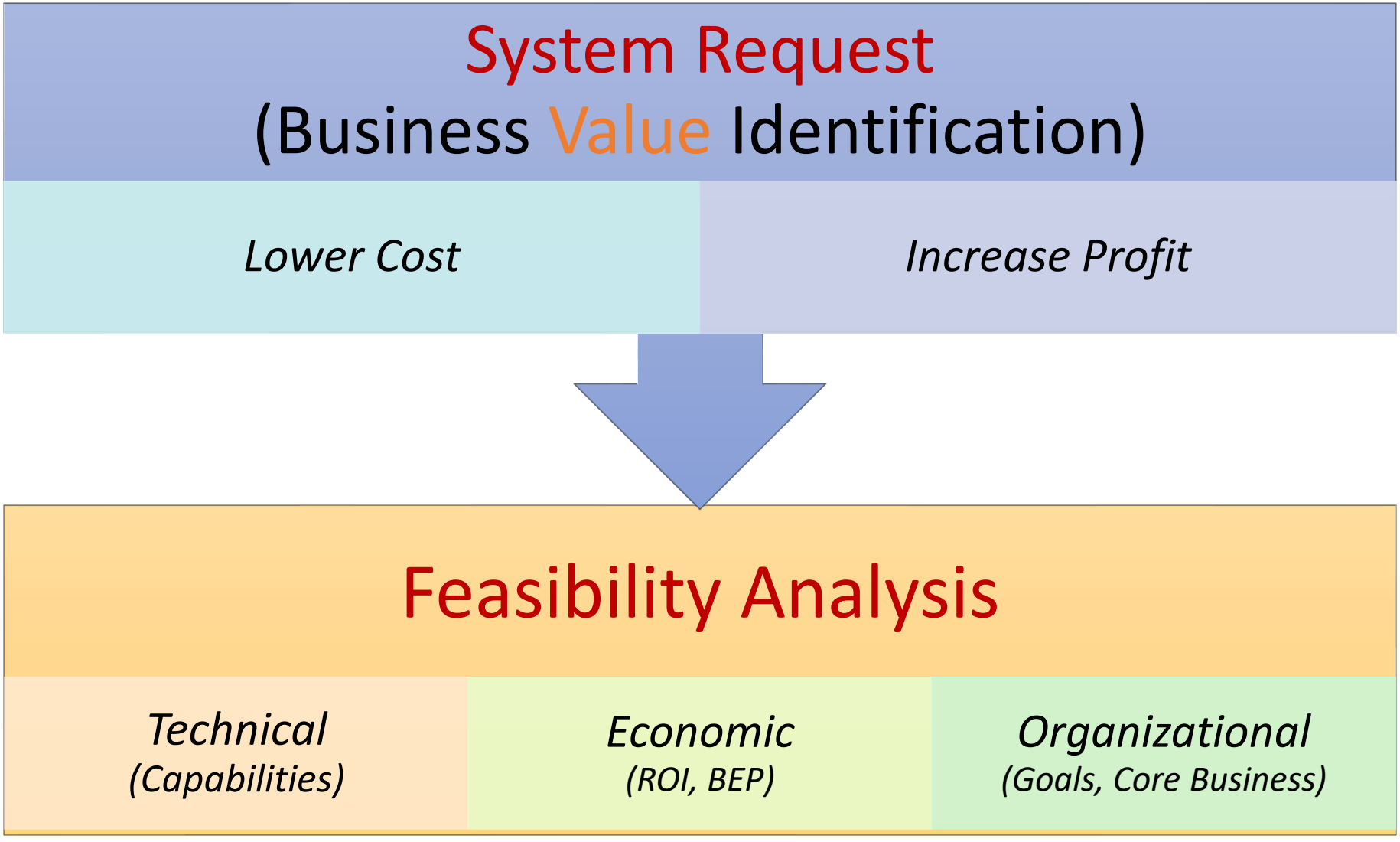
Maintenance
(Updated Software)

Siklus Pengembangan Software



(Shelly, 2012) *(Dennis, 2016)* *(Valacich, 2017)*

Planning


System Request
(Business Value Identification)
Lower Cost
Increase Profit
Feasibility Analysis
Technical
(Capabilities)
Economic
(ROI, BEP)
Organizational
(Goals, Core Business)

System Request: Sistem Penjualan Musik Online

Project Sponsor: Margaret Mooney, Vice President of Marketing

Business Needs: Project ini dibangun untuk:

1. Mendapatkan pelanggan baru lewat Internet
2. Meningkatkan efisiensi penanganan masalah pelanggan melalui internet

Business Requirements:

Sistem yang mendukung penjualan musik secara online. Fitur-fitur yang harus ada:

1. Fitur Pencarian Produk
2. Fitur Pencarian Toko yang Menyediakan Stok Produk
3. Fitur Pemesanan Produk Melalui Toko yang Menyediakan
4. Fitur Pembayaran dengan Berbagai Pilihan Pembayaran

Business Value:

Intangible Value:

- Meningkatkan kenyamanan dan kepuasan pelanggan
- Meningkatkan brand recognition tentang perusahaan di dunia Internet

Tangible Value:

1. Meningkatkan penjualan dari pelanggan baru lewat Internet:
   - Rp 400 juta peningkatan penjualan dari pelanggan baru dan Rp 600 juta dari pelanggan lama
2. Mengurangi biaya operasional untuk menangani komplain dari pelanggan
   - Rp 100 juta pengurangan tahunan biaya telepon untuk menangani pelanggan

Studi Kelayakan Sistem Penjualan Musik Online

Margaret Mooney dan Alec Adams membuat studi kelayakan untuk pengembangan Sistem Penjualan Musik Online

Kelayakan Teknis

Sistem penjualan musik online layak secara teknis, meskipun memiliki beberapa risiko.

Risiko Berhubungan dengan **Kefamilieran dengan Aplikasi**: Resiko Tinggi

- Divisi Marketing tidak memiliki pengalaman menggunakan sistem penjualan online
- Divisi IT memiliki pemahaman yang baik tentang sistem penjualan offline, akan tetapi tidak berpengalaman mengembangkan sistem penjualan musik online

Risiko Berhubungan dengan **Kefamilieran dengan Teknologi**: Resiko Sedang

- Divisi IT tidak menguasai masalah infrastruktur dan ISP, tetapi akan menyewa konsultan
- Divisi IT cukup familier dengan framework dan IDE yang akan digunakan
- Divisi Marketing tidak memiliki pengalaman menggunakan teknologi Web

Risiko berhubungan dengan **Ukuran Project**: Risiko Rendah

- Perusahaan memiliki total 30 orang pengembang
- Project dikerjakan oleh 5 orang pengembang dengan estimasi waktu 6 bulan

Kompatibilitas dengan sistem dan infrastruktur yang ada: Risiko Rendah

- Sistem pemesanan yang ada sekarang menggunakan *open standard*, jadi sangat kompatibel dengan sistem penjualan berbasis web yang akan dibangun

Kelayakan Ekonomi

Cost benefit analysis telah dilakukan. Sistem Penjualan musik online memiliki peluang yang baik untuk bisa meningkatkan pendapatan perusahaan.

- Return on Investment (ROI) setelah 3 tahun: **31%**
- Break-even point (BEP): **2.25 tahun**
- Total keuntungan setelah 3 tahun: **Rp. 503.559.986,-**

Kelayakan Organisasi

- Secara organisasi, resikonya rendah. Tujuan dari pengembangan sistem penjualan musik online adalah meningkatkan penjualan perusahaan. Dan ini selaras dengan KPI marketing yang ke arah peningkatan kuantitas penjualan
- Project champion dari pengembangan sistem penjualan musik online ini adalah Margaret Mooney, Vice President of Marketing

	2019	2020	2021
Peningkatan penjualan dari pelanggan baru	0	400,000,000	500,000,000
Peningkatan penjualan dari pelanggan lama	0	600,000,000	700,000,000
Pengurangan biaya operasional dan telepon	0	100,000,000	100,000,000
Total Benefits:	**0**	**1,100,000,000**	**1,300,000,000**
PV of Benefits:	**0**	**978,996,084**	**1,091,505,068**
PV of All Benefits:	**0**	**978,996,084**	**2,070,501,152**
Honor Tim (Planning, Analysis, Design and Implementation)	360,000,000	0	0
Honor Konsultan Infrastruktur Internet	90,000,000	0	0
Total Development Costs:	**450,000,000**	**0**	**0**
Honor Pengelola Web	60,000,000	70,000,000	80,000,000
Biaya Lisensi Software	50,000,000	60,000,000	70,000,000
Hardware upgrades	100,000,000	100,000,000	100,000,000
Biaya Komunikasi	20,000,000	30,000,000	40,000,000
Biaya Marketing	100,000,000	200,000,000	300,000,000
Total Operational Costs:	**330,000,000**	**460,000,000**	**590,000,000**
Total Costs:	**780,000,000**	**460,000,000**	**590,000,000**
PV of Costs:	**735,849,057**	**409,398,362**	**495,375,377**
PV of all Costs:	**735,849,057**	**1,145,247,419**	**1,640,622,796**
Total Project Costs Less Benefits:	**-780,000,000**	**640,000,000**	**710,000,000**
Yearly NPV:	**-735,849,057**	**569,597,722**	**596,129,691**
Cumulative NPV:	**-735,849,057**	**-166,251,335**	**429,878,356**
Return on Investment (ROI) di Tahun 3: 26.2%	**429,878,356/1,640,622,796**		**0.262021445**
Break-even Point (BEP): 2.28 tahun	**2 + (596,129,691-429,878,356)/ 596,129,691**		**2.278884507**

System Request

(Business Value)

ROI dan BEP

Return on Investment (ROI)

The amount of revenue or cost savings results from a given investment

$$PV = \frac{\text{Total Benefits - Total Costs}}{\text{Total Costs}}$$

$$\frac{3{,}204{,}752 - 2{,}575{,}331}{2{,}575{,}331} = 0.2444$$

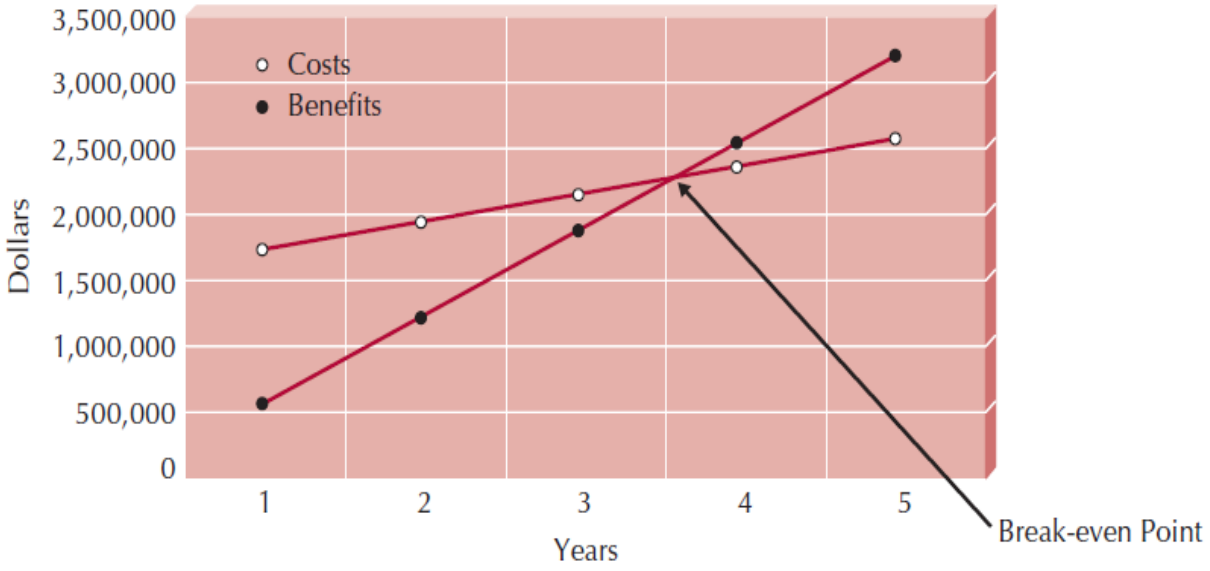
	2003	2004	2005	2006	2007	Total
Increased sales	500,000	530,000	561,800	595,508	631,238	
Reduction in customer complaint calls	70,000	70,000	70,000	70,000	70,000	
Reduced inventory costs	68,000	68,000	68,000	68,000	68,000	
Total Benefits:	638,000	668,000	699,800	733,508	769,238	
PV of Benefits:	619,417	629,654	640,416	651,712	663,552	3,204,752
PV of All Benefits:	619,417	1,249,072	1,889,488	2,541,200	3,204,752	
2 Servers @ $125,000	250,000	0	0	0	0	
Printer	100,000	0	0	0	0	
Software licenses	34,825	0	0	0	0	
Server software	10,945	0	0	0	0	
Development labor	1,236,525	0	0	0	0	
Total Development Costs:	1,632,295	0	0	0	0	
Hardware	54,000	81,261	81,261	81,261	81,261	
Software	20,000	20,000	20,000	20,000	20,000	
Operational labor	111,788	116,260	120,910	125,746	130,776	
Total Operational Costs:	185,788	217,521	222,171	227,007	232,037	
Total Costs:	1,818,083	217,521	222,171	227,007	232,037	
PV of Costs:	1,765,129	205,034	203,318	201,693	200,157	2,575,331
PV of All Costs:	1,765,129	1,970,163	2,173,481	2,375,174	2,575,331	
Total Project Benefits—Costs:	(1,180,083)	450,479	477,629	506,501	537,201	
Yearly NPV:	(1,145,712)	424,620	437,098	450,019	463,395	629,421
Cumulative NPV:	(1,145,712)	(721,091)	(283,993)	166,026	629,421	
Return on Investment:	24.44%	(629,421/2,575,331)				
Break-even Point:	3.63 years	(break-even occurs in year 4; [450,019 – 166,026] / 450,019 = 0.63)				
Intangible Benefits:	This service is currently provided by competitors Improved customer satisfaction					

Break Even Point (BEP)

- The point in time when the costs of the project equal the value it has delivered

$$BEP = \frac{\text{Yearly NPV}^* - \text{Cumulative NPV}}{\text{Yearly}^* \text{ NPV}}$$

- Use the yearly NPV amount from the first year in which project has positive cash flow



Alasan Utama Software Dikembangkan

- Software dikembangkan karena ada kebutuhan dan benefit:
  - Mengurangi biaya operasional
  - Meningkatkan performa kerja
  - Meningkatkan efisiensi
  - Meningkatkan pendapatan perusahaan

- Dibuktikan dengan fase planning yang akan menghitung apakah sebuah software layak dikembangkan (feasibility analysis):
  - Kelayakan Teknis (mampu buat dan pakai)
  - Kelayakan Ekonomi (untung dan cepat balik modal)
  - Kelayakan Organisasi (sesuai tugas dan fungsi)

MITOS 2

Kemampuan Terpenting Bagi Pengembang adalah Kemampuan Coding

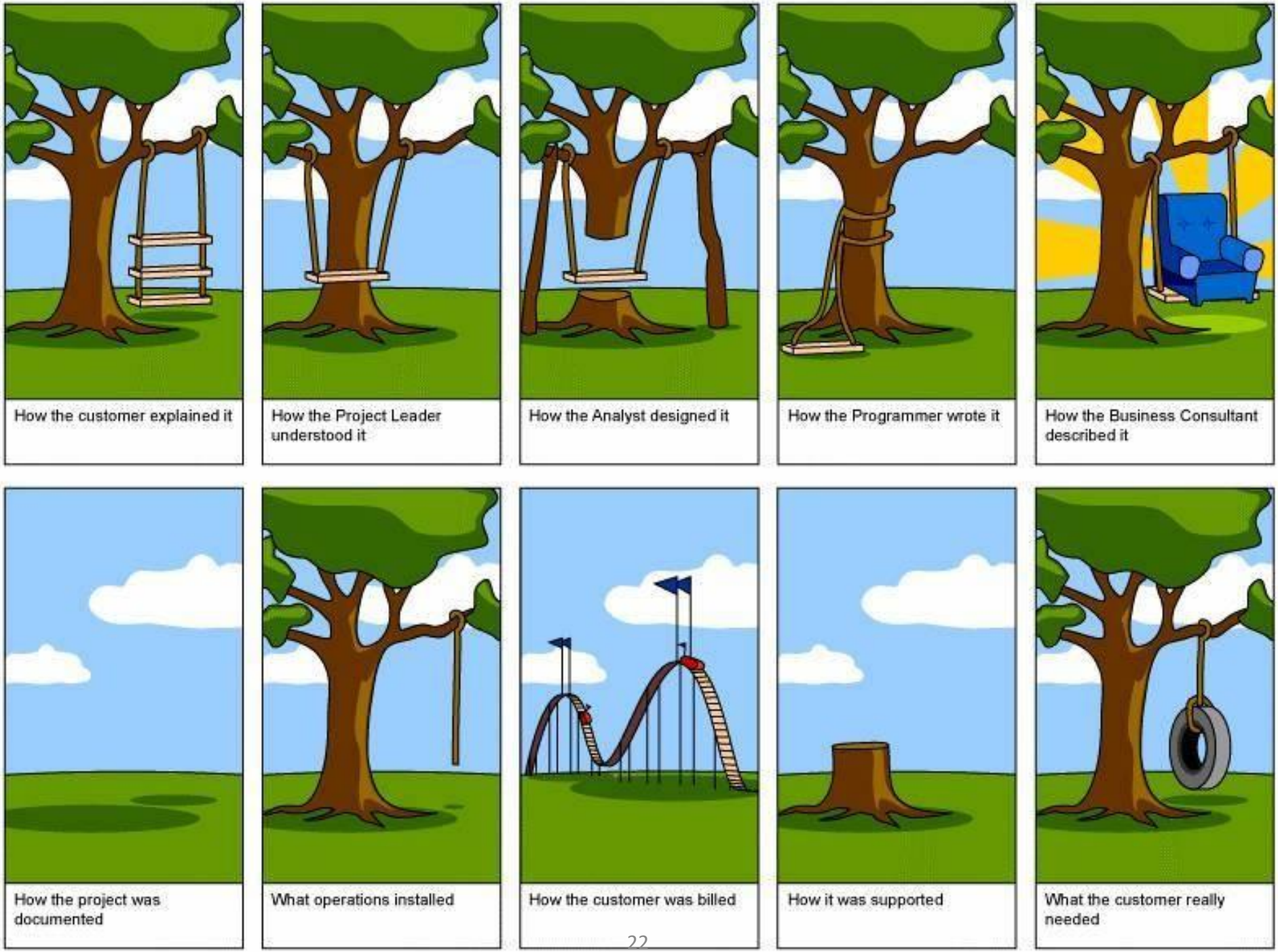
Kegagalan Project Software

50% lebih project teknologi informasi gagal
(42% - Standish Group, 53% - General Accounting Office)

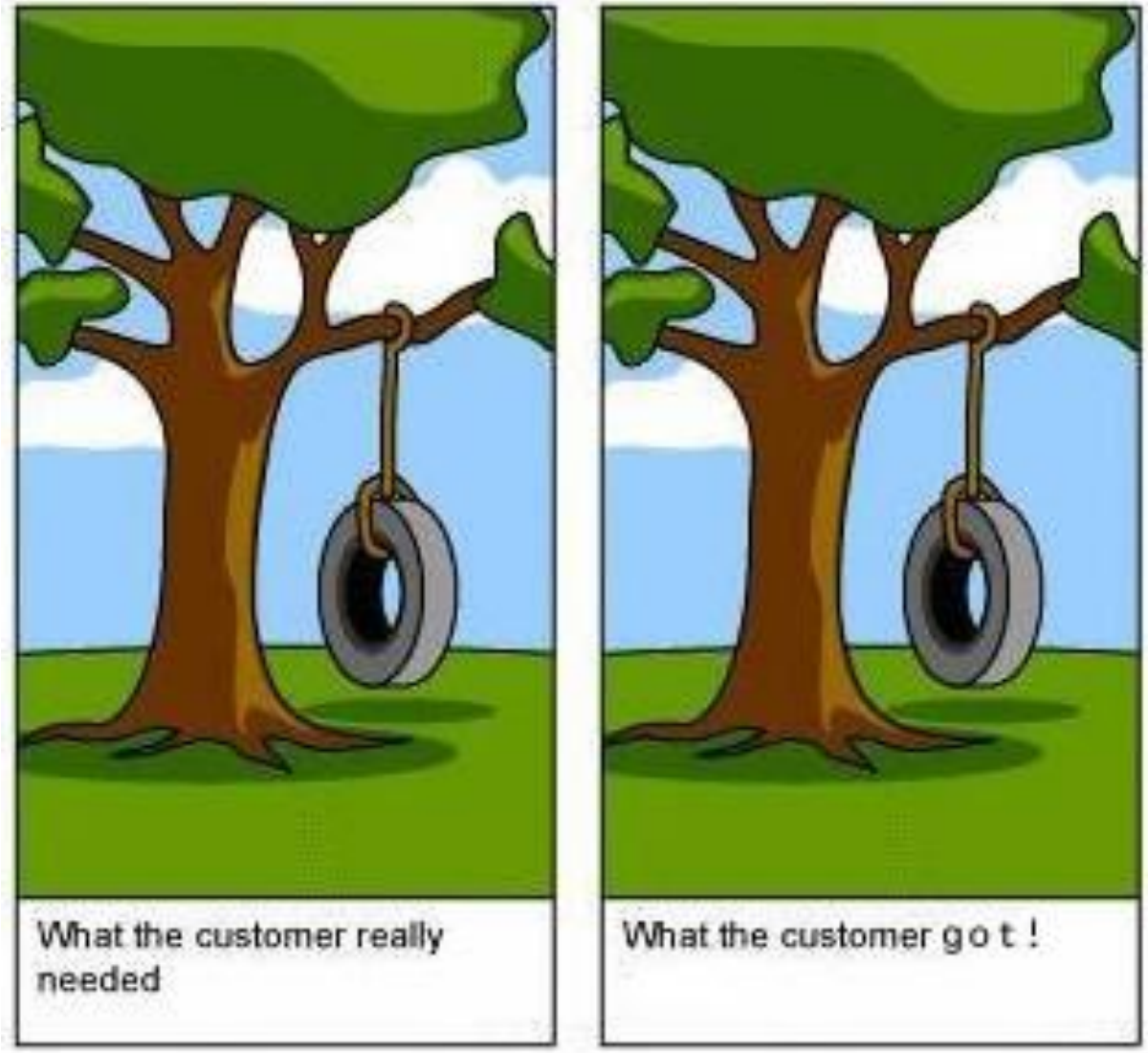
- Dibatalkan sebelum selesai
- Selesai tapi tidak pernah dipakai
- Tidak bermanfaat bagi pengguna
- Tidak sesuai dengan keinginan pengguna

Size Berbanding Lurus dengan Kegagalan

Organization	Year	Problems and Damage Cost
Canada Central Government	2014	Government website portal, projects with ongoing problems, $37.4 million loss
US State government	2012	Healthcare exchange website, cancelled, client and supplier both sued each other $200 million loss
Hudson Bay (Canada)	2005	Inventory system problems lead to $33.3 million loss
UK Inland Revenue	2005	$3.45 billion tax-credit overpayment caused by software errors
Avis Europe PLC (UK)	2004	Enterprise resource planning (ERP) system cancelled after $54.5 million spent
Ford Motor Co.	2004	Purchasing system abandoned after deployment costing approximately $400 million
Hewlett-Packard Co.	2004	ERP system problems contribute to $160 million loss
AT&T Wireless	2004	Customer relations management (CRM) system upgrade problems lead to $100 million loss


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Tantangan Pengembangan Software

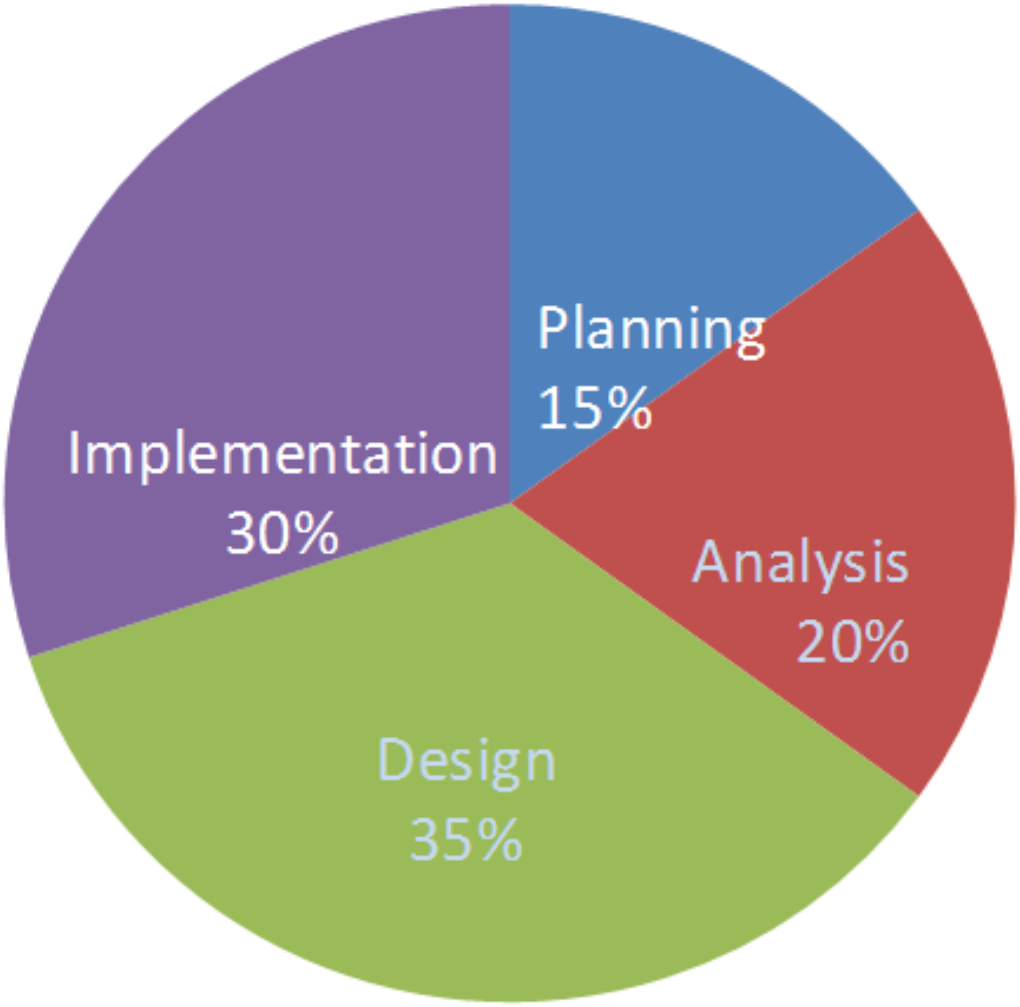


Glass' Law (1998)

Requirement deficiencies are the prime source of project failures

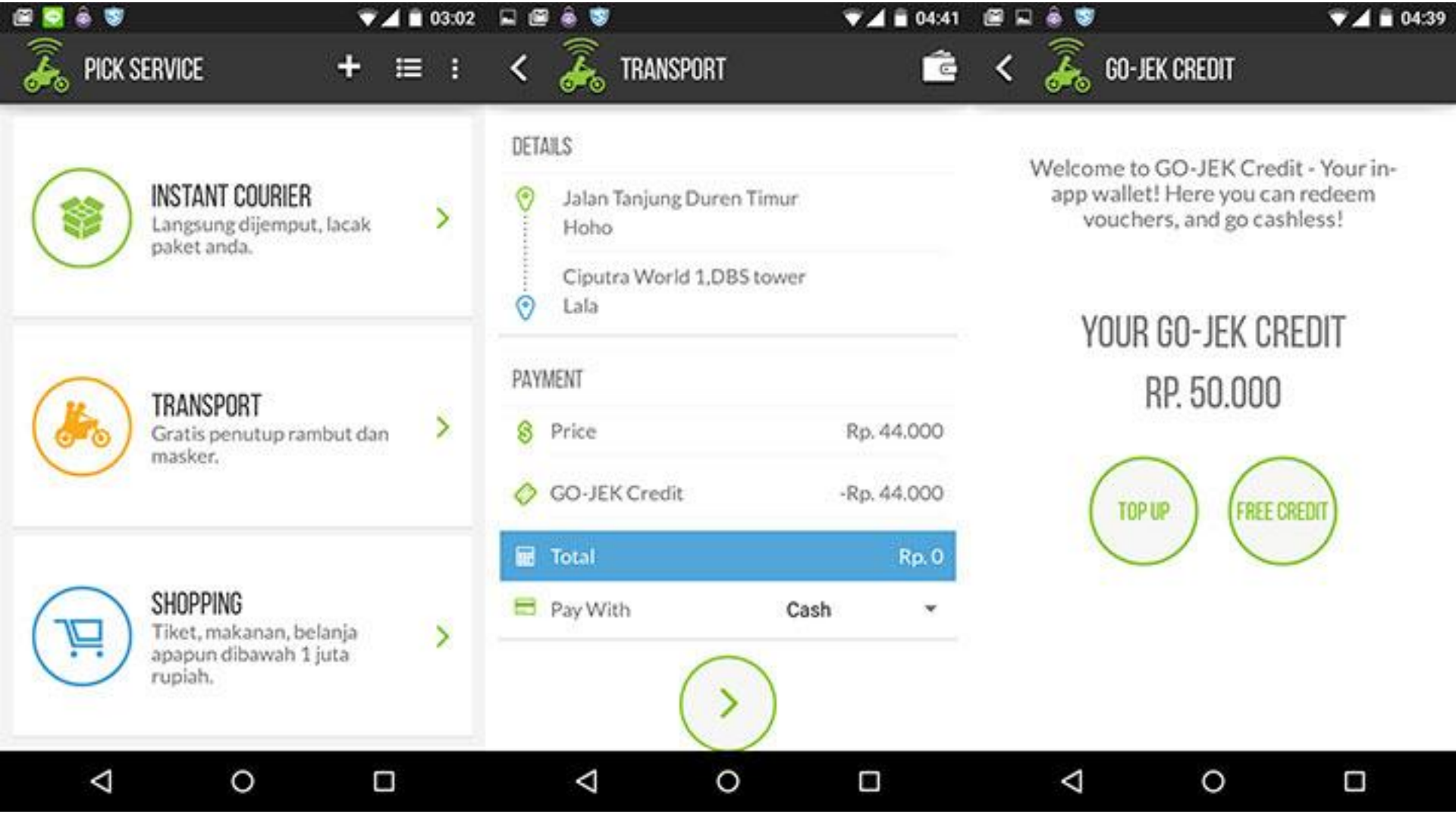
[L1]

Distribusi Effort Pengembangan Software



(Dennis, 2016) *(Jones, 2012)*

Berapa Harga Kode Gojek?



Berapa Harga (Ide) Software Gojek?

Bos Gojek dan Bukalapak Masuk Daftar Orang Terkaya di Indonesia

Reporter: Imam Hamdi
Editor: Kodrat Setiawan

Kamis, 26 Juli 2018 06:21 WIB

0 KOMENTAR | 71 | 0 | 72

Pendiri dan CEO Go-Jek (Gojek) Indonesia, Nadiem Makarim. TEMPO/Ratih Purnama

TEMPO.CO, Jakarta - Empat pendiri start up atau perusahaan rintisan masu
terkaya di Indonesia versi Majalah Globe Asia yang dirilis pada Juni 2018. M
Unardy, 30 tahun, pendiri Traveloka; Wiliam Tanu Jaya, 36 tahun, (Tokopedi
tahun, (Bukalapak) dan Nadiem Makarim, 33 tahun, (Gojek).

Baca juga: Bos Djarum Masih Orang Terkaya di Indonesia

Keempat nama pendiri start up tersebut merupakan wajah baru yang masu
negara ini. Globe Asia mencatat kekayaan Ferry yang menduduki urutan 14
juta. Wiliam berada di urutan 148 (US$ 130 juta), Achmad di urutan 149 (US
di urutan 150 (US$ 100 juta).

Google Tanam Modal, Valuasi Go-Jek Capai Rp 53 Triliun?

FATIMAH KARTINI BOHANG
Kompas.com - 18/01/2018, 16:37 WIB

Ojek Online yang manggkal di bawah kolong flyover dekat Stasiun Tebet (Stanly)

KOMPAS.com — Raksasa internet Google dikabarkan menggelontorkan dana segar ke layanan *ride-sharing* Go-Jek pada awal 2018 ini. Kesepakatan keduanya diteken paling cepat pekan depan.

Tak diumbar jumlah pastinya, tetapi investasi itu menaikkan valuasi Go-Jek menjadi 4 miliar dollar AS atau setara Rp 53,3 triliun. Setidaknya begitu menurut sumber dalam yang familiar dengan penanaman modal ini.

Jika hal itu benar, Go-Jek yang hanya fokus di Indonesia semakin dekat mengejar Grab yang bisnisnya lebih luas seantero Asia Tenggara. Valuasi terakhir Grab disebutkan berada di angka 6 miliar dollar AS atau senilai Rp 80

4 HappyFresh

HappyFresh
TOTAL PENDANAAN
US$12 juta

TAHAP PENDANAAN: SERI-A
KATEGORI: E-COMMERCE

...sa pesan kebutuhan ...harian dari HappyFresh, ...an diantarkan oleh jasa ...gsung ke depan pintu ...amu.

5 Qraved

Qraved
TOTAL PENDANAAN
US$8 juta

TAHAP PENDANAAN: SERI-B
KATEGORI : FOOD TECH

1 tokopedia

Tokopedia
TOTAL PENDANAAN
US$100 juta

TAHAP PENDANAAN : SERI-E
KATEGORI : E-COMMERCE

Marketplace e-commerce ini adalah *startup* kawakan di Indonesia yang kini hampir berusia 7 tahun. Investasi US$100 juta yang mereka raih dari Sequoia dan Softbank pada 2014 berhasil menorehkan sejarah dalam peta *startup* di Indonesia.

...RYBENKA.com ...IS JUST A CLICK AWAY

...rybenka
...L PENDANAAN
...$5 juta

...ENDANAAN : SERI-B
...RI : E-COMMERCE

...commerce khusus ...ini didanai oleh Gree ...s.

8 SCOOP

Scoop
TOTAL PENDANAAN
US$3.21 juta

TAHAP PENDANAAN : SERI-B
KATEGORI : MEDIA

Aplikasi pembaca e-book dan majalah ini didirikan oleh Willson Cuaca yang juga menjabat sebagai *managing partner* di East Ventures.

9 ralali.com Industrial Online Store

Ralali
TOTAL PENDANAAN
US$2.5 juta

TAHAP PENDANAAN: SERI-A
KATEGORI : MARKETPLACE & PLATFORM

Ralali merupakan toko *online* yang menyediakan kebutuhan dan perlengkapan industri.

2 BHINNEKA.COM INDONESIA'S NO.1 ONLINE STORE

Bhinneka
TOTAL PENDANAAN
US$22 juta

TAHAP PENDANAAN : TAHAP AKHIR
KATEGORI : E-COMMERCE

3 Orami

Orami
TOTAL PENDANAAN
US$15 juta

TAHAP PENDANAAN: TAK DIPUBLIKASI
KATEGORI : E-COMMERCE

10 fabelio

Fabelio
TOTAL PENDANAAN
US$2.5 juta

TAHAP PENDANAAN : SERI-A
KATEGORI: E-COMMERCE

Jika kamu sedang mencari meja

11 Bizzy EVERYTHING FOR BUSINESS

Bizzy
TOTAL PENDANAAN
US$2.5 juta

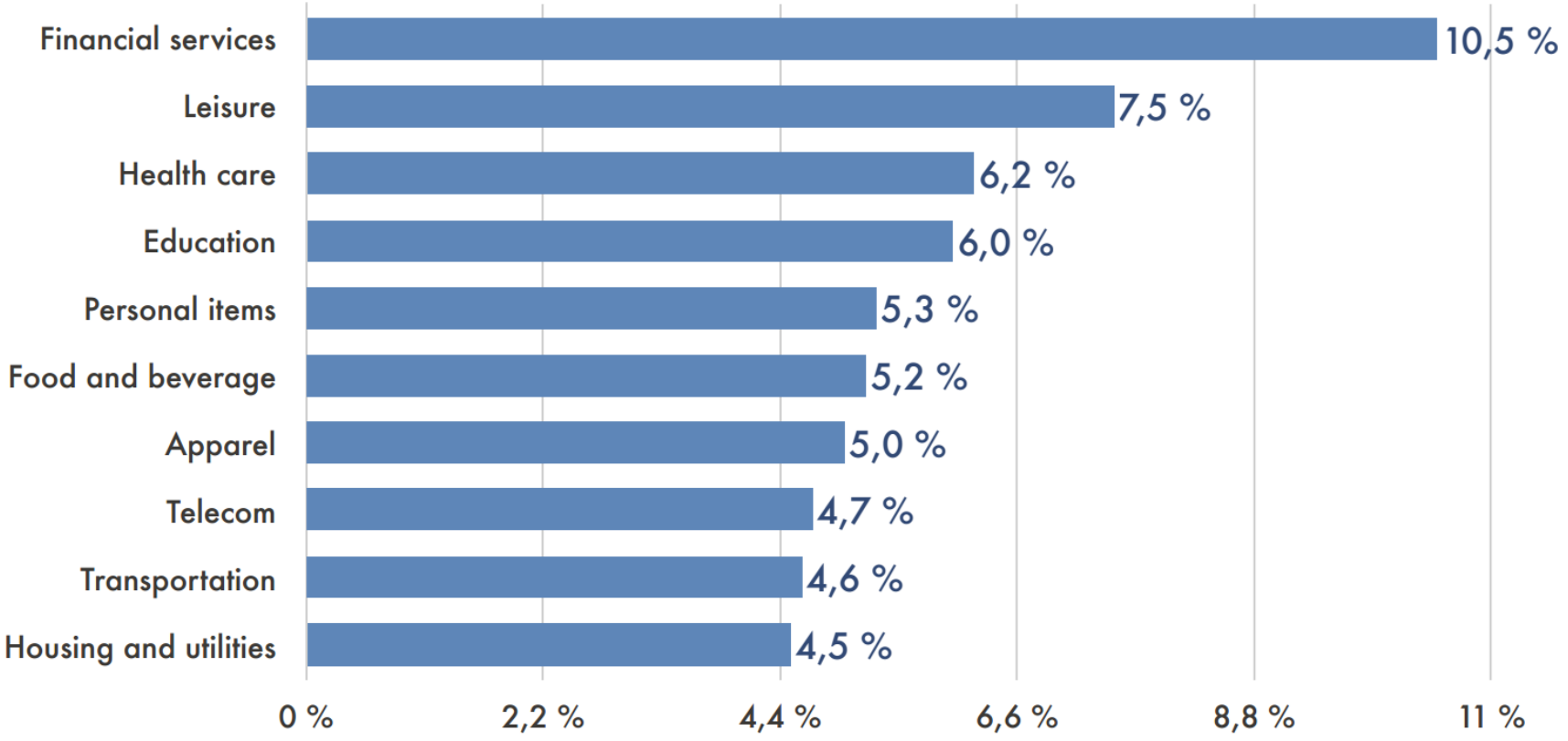
TAHAP PENDANAAN : TAHAP AWAL
KATEGORI : MARKETPLACE & PLATFORM

Kemampuan Coding?

- Penting!

- Tapi lebih penting lagi kemampuan membaca kebutuhan riil masyarakat dan kemampuan analisis kelayakan dari software yang kita kembangkan

Statistik Penggunaan Uang Orang Indonesia



RAPID URBANIZATION, RISING INCOME LEVELS, FAVORABLE DEMOGRAPHIC PATTERNS AND CHANGING LIFESTYLE TRENDS ARE JUST SOME OF THE FACTORS THAT ARE BOOSTING CONSUMER SPENDING IN INDONESIA

Source: McKinsey 2013

Programmer vs System Analyst

Mendidik programmer lebih mudah daripada system analyst, software engineer atau enterprise architect

Norman's Law (1993)

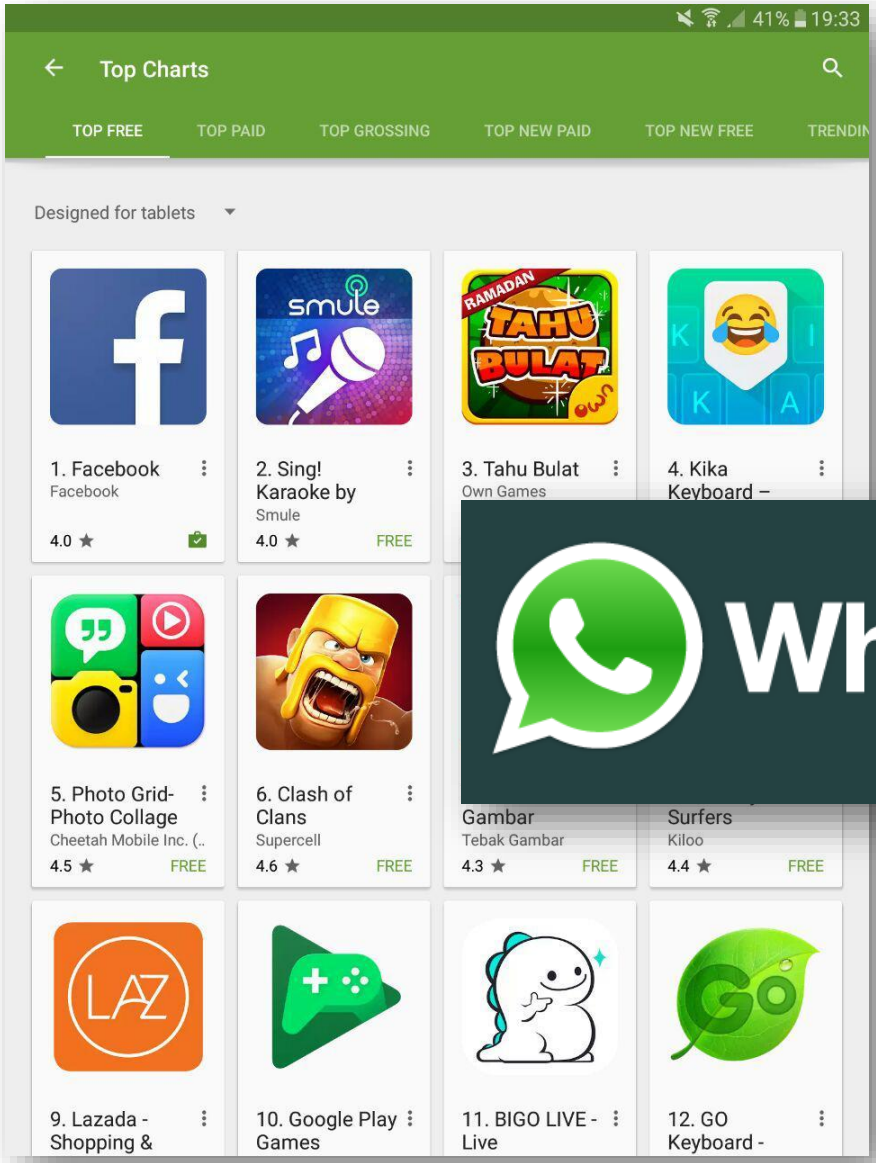
It takes 5000 hours to turn a novice into an expert

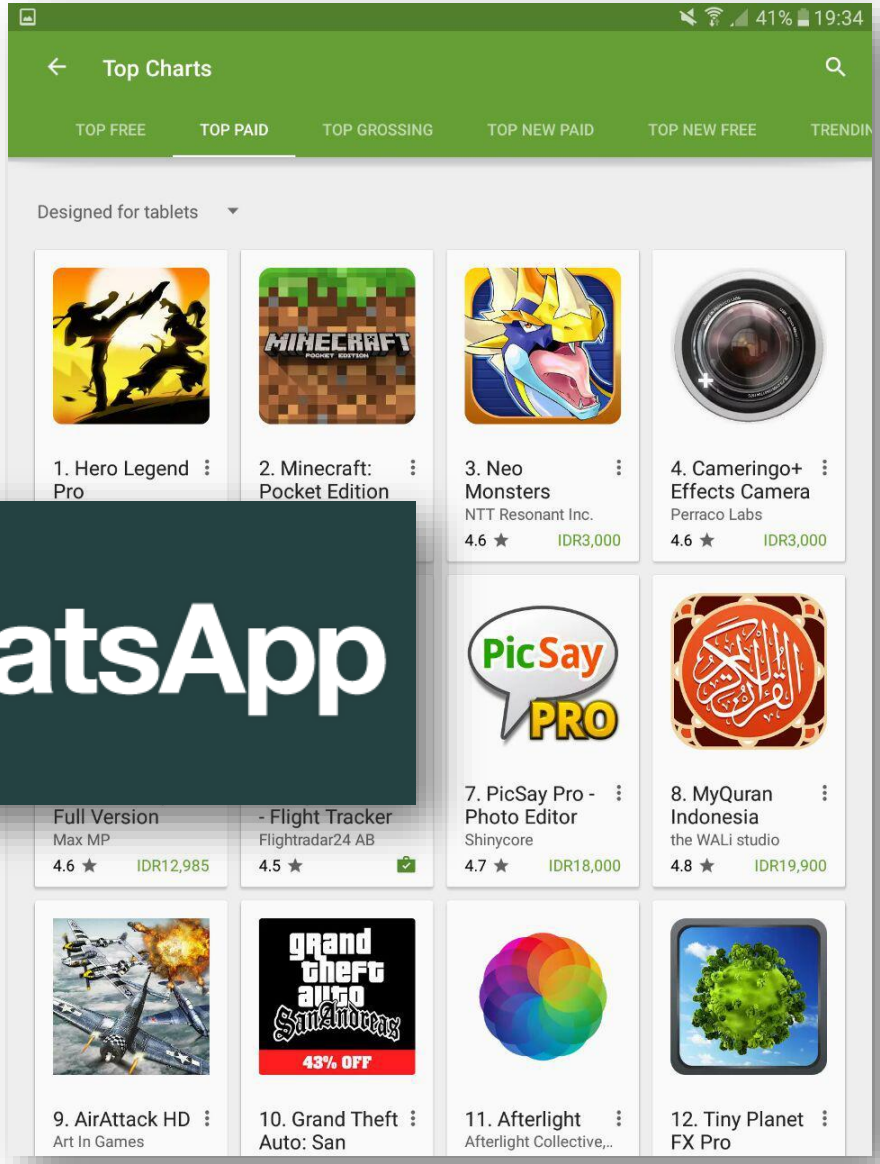
[L43]

MITOS 3

Kualitas Software Dinilai Dari Teknologi Yang Digunakan

Teknologi Dibalik Software yang Sukses?





Software Berkualitas?

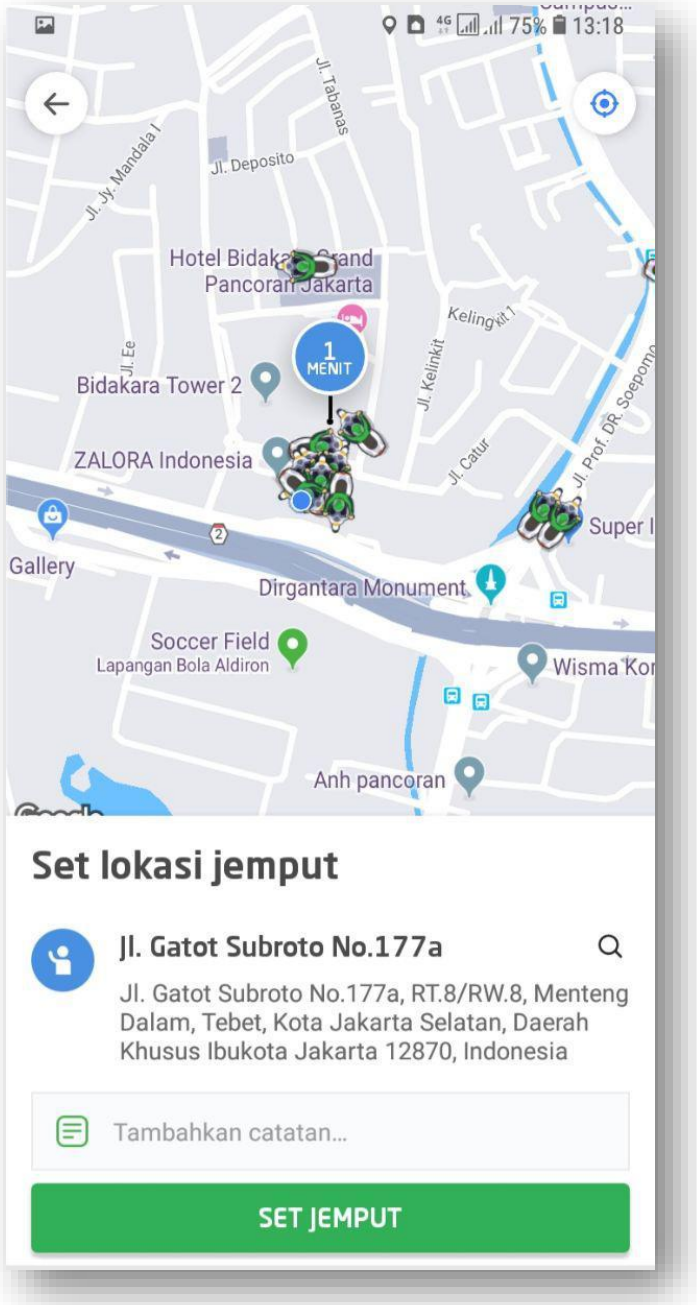
Software quality is *(IEEE, 1991)*:

1. The degree to which a system, component, or

Sesuai Kebutuhan

2. The degree to which a system, component, or process meets customer

Ada Keuntungan



Komputer Datang untuk Efisiensi!

- Alan Turing dengan komputer yang dibuat berhasil mendekripsi pesan enigma dengan cepat
  - Dilakukan manusia puluhan atau ratusan tahun, dengan komputer hanya 3-4 jam
  - Menyelesaikan perang yang harusnya bisa terjadi puluhan tahun, menjadi hanya 3 tahun, dan menghemat miliaran USD

- Komputer adalah alat untuk efisiensi, untuk mengurangi *cost*, mempercepat waktu, meningkatkan *income*
  - Investasi pengadaan komputer dan software harus bisa meningkatkan Return on Investment (ROI), mempercepat Break Event Point (BEP) pada organisasi

Software Sistem Kita Berkualitas?

1. KTP Manual

3 Hari

2. 

Komputer

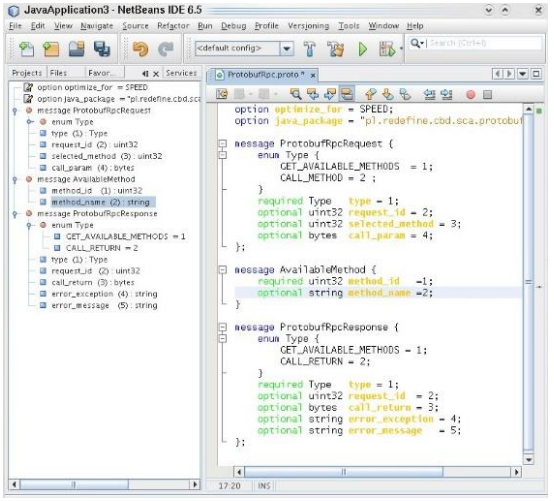
3 Minggu!

3. KTP-EL

1 Tahun!

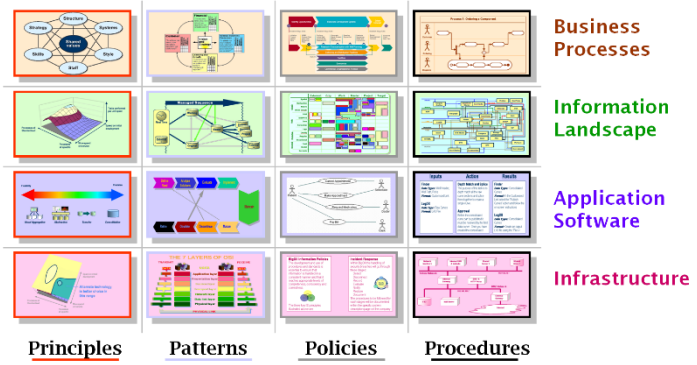
Software Development Evolution

1. How to Write a Code (Coder or Programmer)

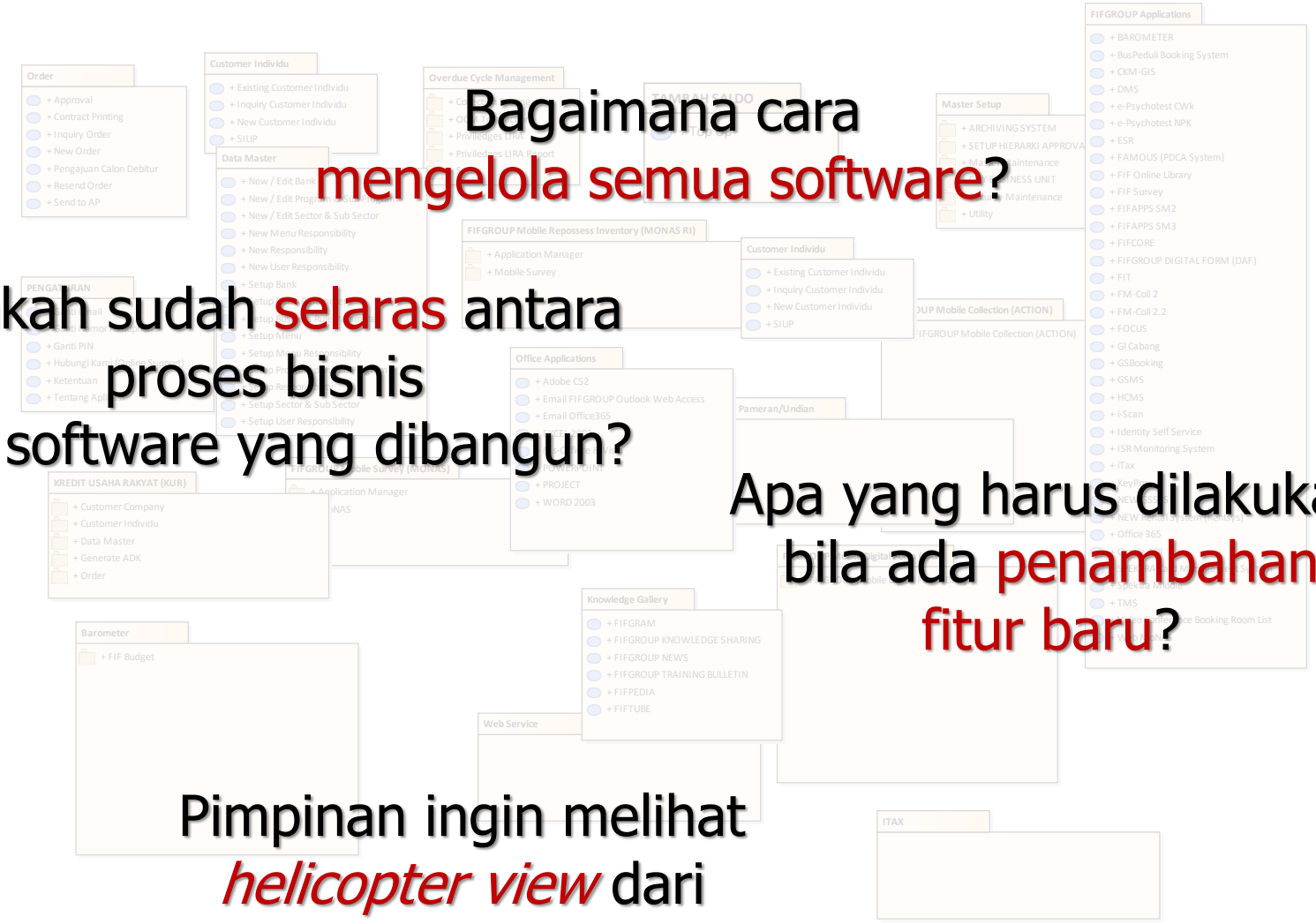


2. How to Develop a Software (Software Engineer)

3. How to Manage Software (Enterprise Architect)



Software Berkembang, Semakin Banyak dan Tumbuh Besar!

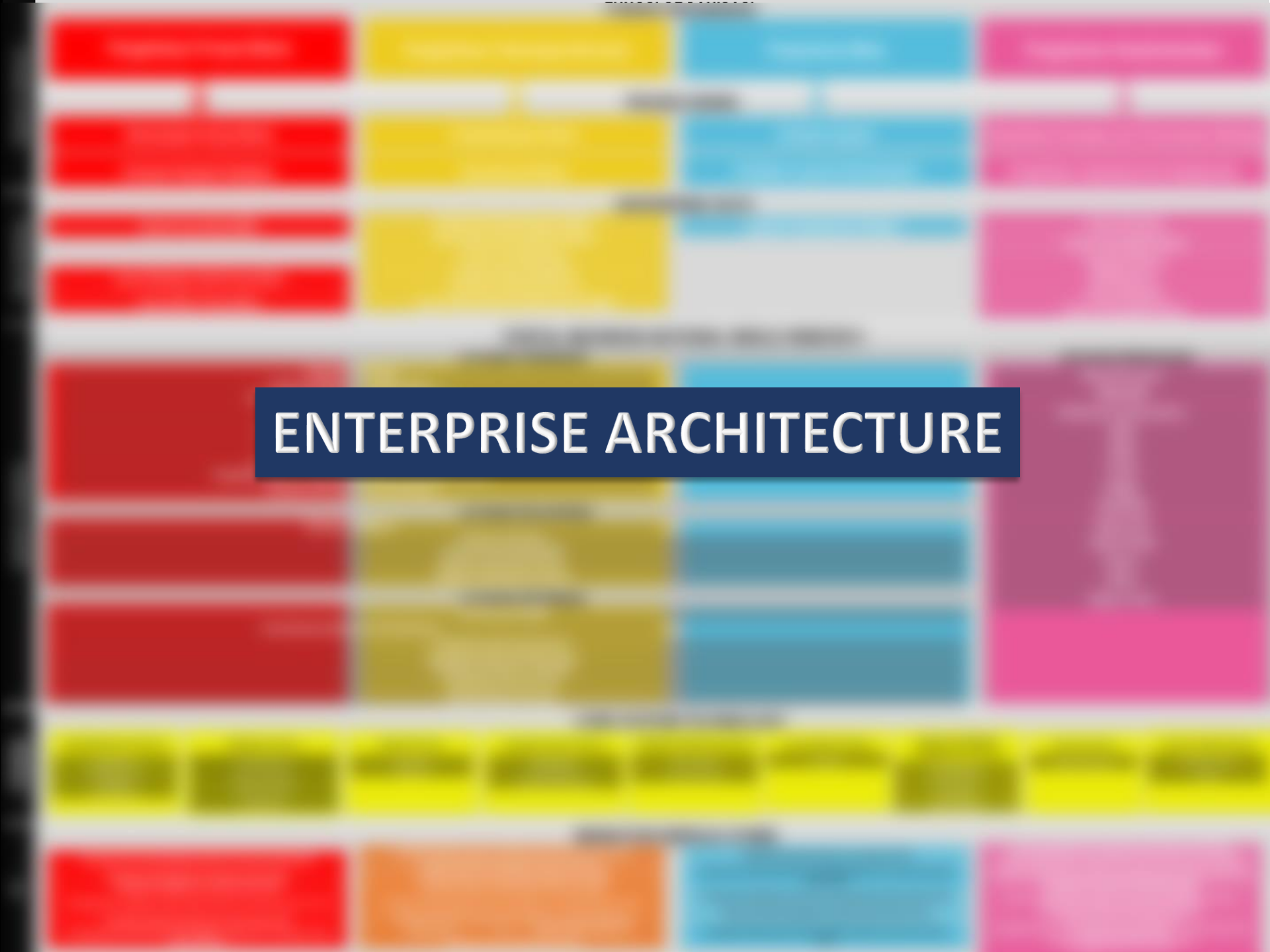


Bagaimana cara
mengelola semua software?

Apakah sudah selaras antara
proses bisnis
dan software yang dibangun?

Apa yang harus dilakukan
bila ada penambahan
fitur baru?

Pimpinan ingin melihat
helicopter view dari
semua software yang ada?


ENTERPRISE ARCHITECTURE

Morris–Ferguson Law (1993)

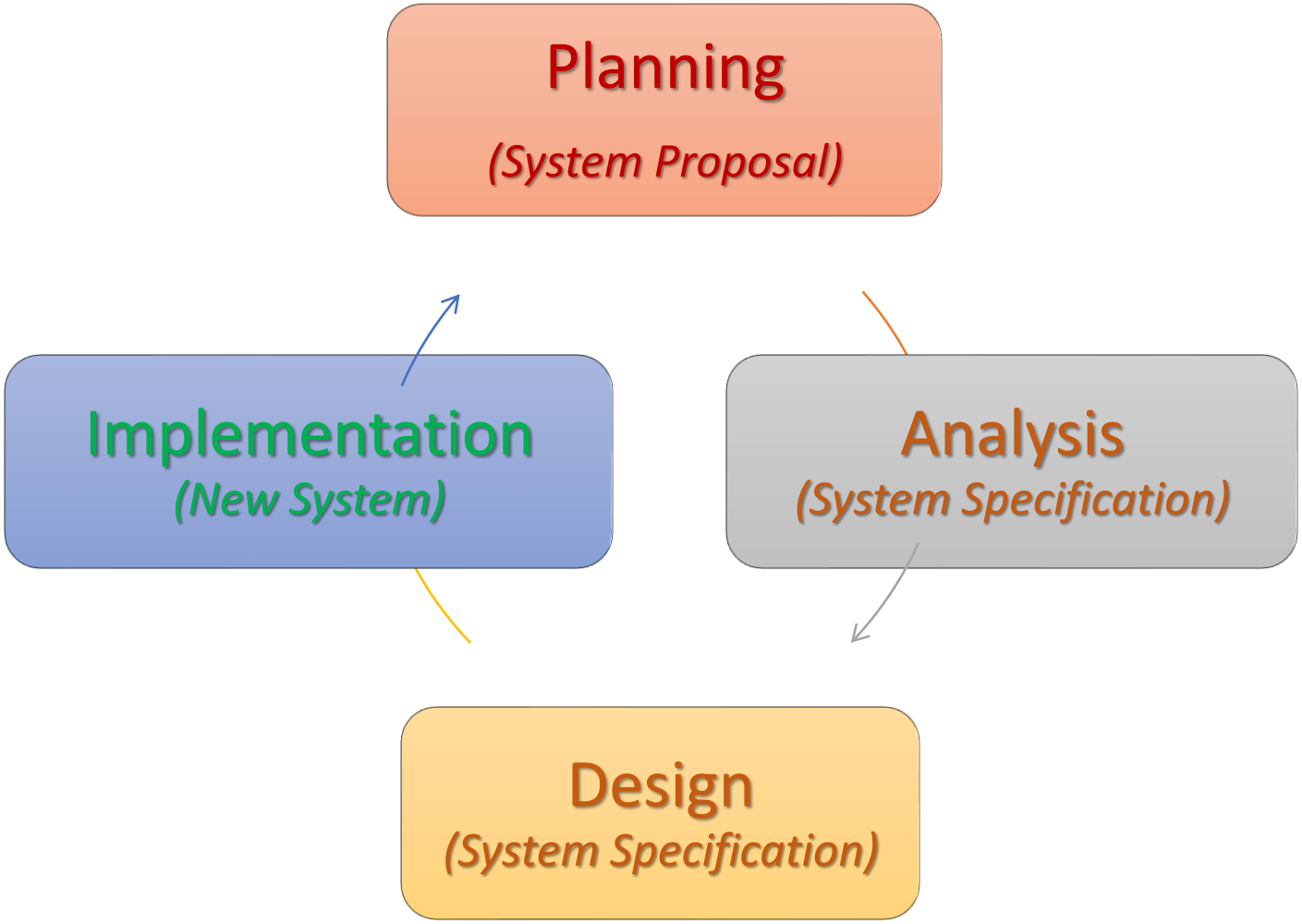
Architecture wins over technology

[L48]

MITOS 4

SDLC itu Waterfall dan Itu Buruk dan Sudah Kuno

Siklus Pengembangan Software



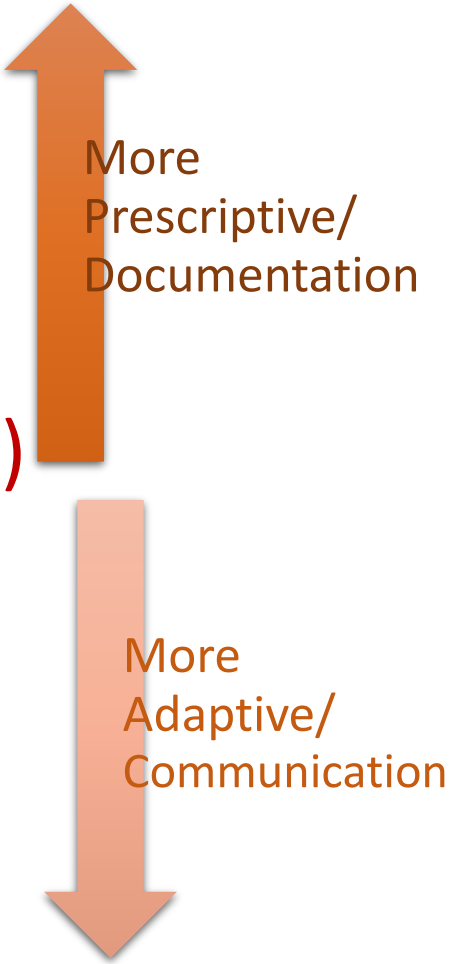
(Tilley, 2012) *(Dennis, 2016)* *(Valacich, 2017)*

Metodologi Pengembangan Software (Model Process)

- Pendekatan yang digunakan untuk menerapkan siklus pengembangan software *(Dennis, 2012)*
- A simplified representation of a software process *(Sommerville, 2015)*
- A distinct set of activities, actions, tasks, milestones, and work products required to engineer high quality software *(Pressman, 2015)*

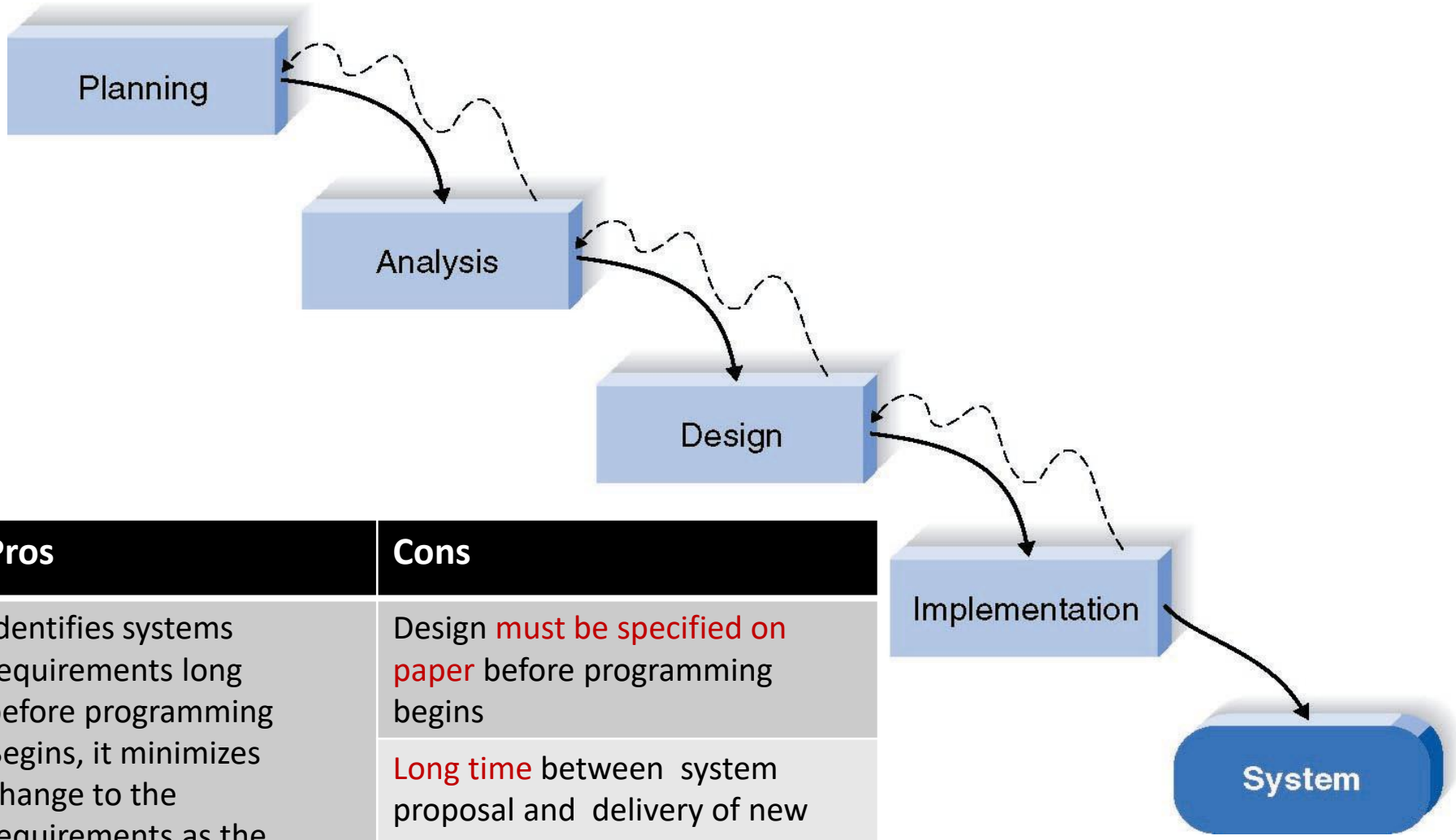
Metodologi Pengembangan Software

1. Structured Design (SD)
   - Waterfall method
   - Parallel development
2. Rapid Application Development (RAD)
   - Phased Development
   - Prototyping
3. Agile Development
   - Extreme Programming (XP)
   - Scrum



(Dennis, 2012)

1. SD: Waterfall Method



Pros	Cons
Identifies systems requirements long before programming Begins, it minimizes change to the requirements as the project proceed (mature)	Design must be specified on paper before programming begins
	Long time between system proposal and delivery of new system
	Rework is very hard

Planning

System Request: Sistem Penjualan Musik Online

Project Sponsor:	Margaret Mooney, Vice President of Marketing
Business Needs:	Project ini dibangun untuk:
	1. Mendapatkan pelanggan baru lewat Internet
	2. M

Business Requirement
Sistem yang menduku
1. Fitur Pencarian Pro
2. Fitur Pencarian Tok
3. Fitur Pemesanan Pr
4. Fi

Busi
Intar
Tang

Studi Kelayakan Sistem Penjualan Musik Online

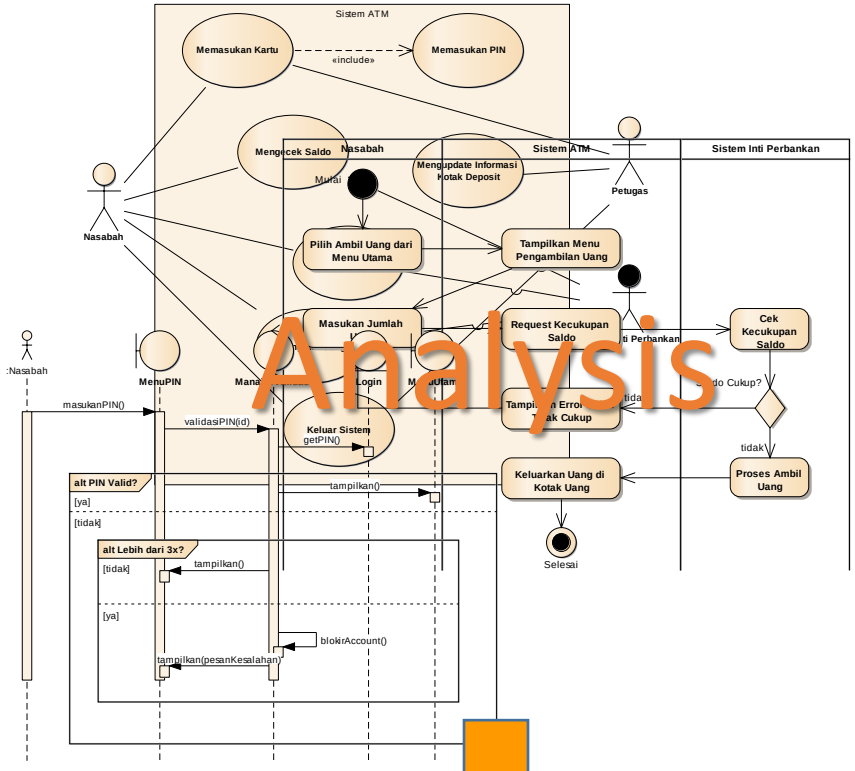
Margaret Mooney dan Alec Adams membuat studi kelayakan untuk pengembangan Sistem Penjualan Musik Online

Kelayakan Teknis

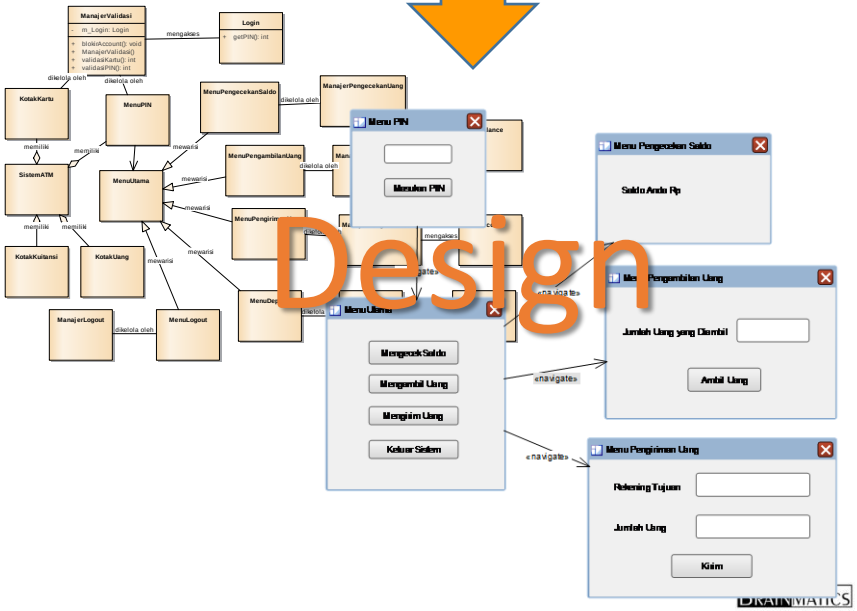
Sistem penjualan musik online layak secara teknis, meskipun memiliki beberapa risiko.

	2016	2017	2018
Peningkatan penjualan dari pelanggan baru	0	400,000,000	500,000,000
Peningkatan penjualan dari pelanggan lama	0	600,000,000	700,000,000
Pengurangan biaya operasional dan telepon	0	100,000,000	100,000,000
Total Benefits:	**0**	**1,100,000,000**	**1,300,000,000**
PV of Benefits:	System R	8,99 084	091
PV of All Benefits:	ness Value)	8,99 084	070,601,1 2
Honor Tim (Planning, A ysis, Design Implementation)	60,00 ,000	0	
Honor Konsultan Infra truktur Internet	00,000,000	0	
Total Development Costs:	**450,000,000**	**0**	**0**
Honor Pengelola Web	60,000,000	70,000,000	80,000,000
Biaya Lisensi Software	50,000,000	60,000,000	70,000,000
Hardware upgrades	100,000,000	100,000,000	100,000,000
Biaya Komunikasi	20,000,000	30,000,000	40,000,000
Biaya Marketing	100,000,000	200,000,000	300,000,000
Total Operational Costs:	**330,000,000**	**460,000,000**	**590,000,000**
Total Costs:	**780,000,000**	**460,000,000**	**590,000,000**
PV of Costs:	**735,849,057**	**409,398,362**	**495,375,377**
PV of all Costs:	**735,849,057**	**1,145,247,419**	**1,640,622,796**
Total Project Costs Less Benefits:	**-780,000,000**	**640,000,000**	**710,000,000**
Yearly NPV:	**-735,849,057**	**569,507,722**	**596,129,691**
Cumulative NPV:	**-735,849,057**	**-166,251,335**	**429,878,356**
Return on Investment (ROI) di Tahun 3: 26.2%	**429,878,356/1,640,622,796**		**0.262021445**
Break-even Point (BEP): 2.28 tahun	**2 + (596,129,691-429,878,356)/ 596,129,691**		**2.278884507**

Analysis

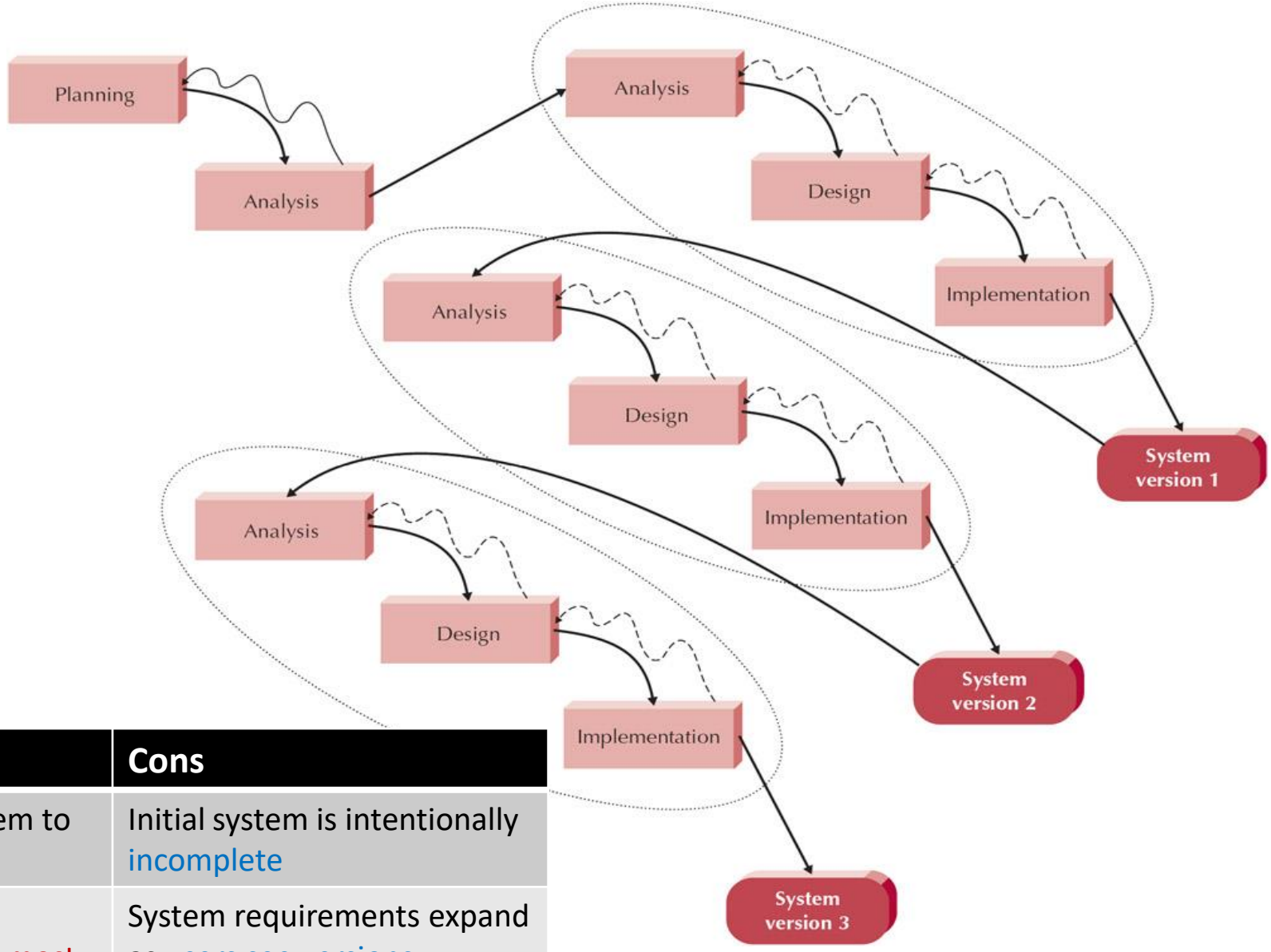


Design



Implementation

2. RAD: Phased Development



Pros	Cons
Gets useful system to users quickly	Initial system is intentionally incomplete
Most important functions tested most	System requirements expand as users see versions

System Request: Sistem Penjualan Musik Online

Project Sponsor: Margaret Mooney, Vice President of Marketing

Business Needs: Project ini dibangun untuk:

1. Mendapatkan pelanggan baru lewat Internet

Studi Kelayakan Sistem Penjualan Musik Online

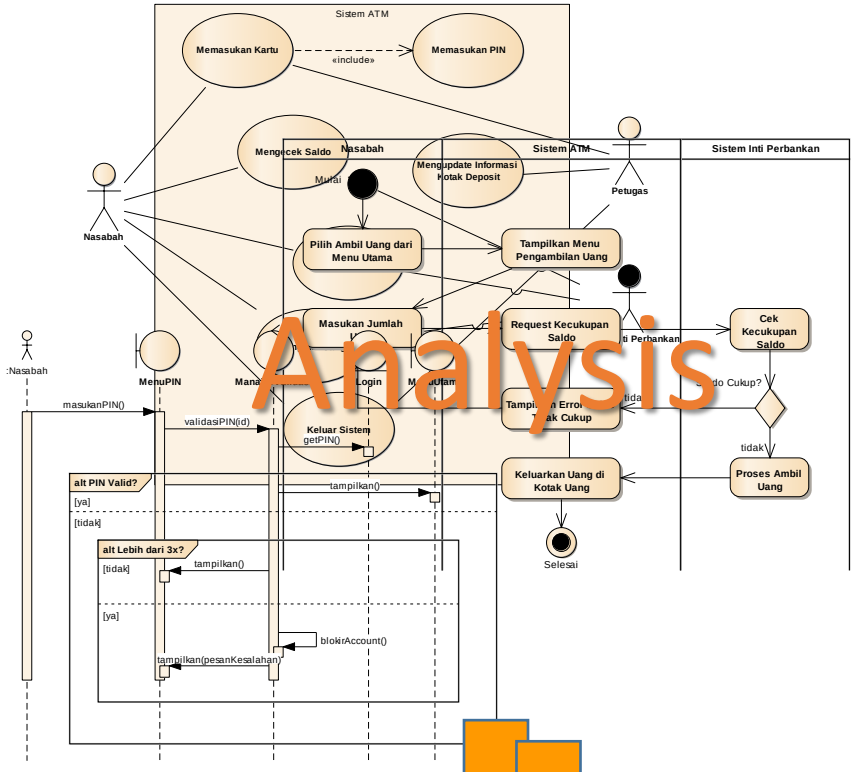
Margaret Mooney dan Alec Adams membuat studi kelayakan untuk pengembangan Sistem Penjualan Musik Online

Kelayakan Teknis

Sistem penjualan musik online layak secara teknis, meskipun memiliki beberapa risiko.

	2016	2017	2018
Peningkatan penjualan dari pelanggan baru	0	400,000,000	500,000,000
Peningkatan penjualan dari pelanggan lama	0	600,000,000	700,000,000
Pengurangan biaya operasional dan telepon	0	100,000,000	100,000,000
Total Benefits:	**0**	**1,100,000,000**	**1,300,000,000**
PV of Benefits:	System Value	8,99 084	091 ,35 ,3
PV of All Benefits:	ness Value)	9 8,99 084	070,601,1 2
Honor Tim (Planning, Analysis, Design and Implementation)	60,00 ,000	0	
Honor Konsultan Infrastruktur Internet	00,000,000	0	
Total Development Costs:	**450,000,000**	**0**	**0**
Honor Pengelola Web	60,000,000	70,000,000	80,000,000
Biaya Lisensi Software	50,000,000	60,000,000	70,000,000
Hardware upgrades	100,000,000	100,000,000	100,000,000
Biaya Komunikasi	20,000,000	30,000,000	40,000,000
Biaya Marketing	100,000,000	200,000,000	300,000,000
Total Operational Costs:	**330,000,000**	**460,000,000**	**590,000,000**
Total Costs:	**780,000,000**	**460,000,000**	**590,000,000**
PV of Costs:	**735,849,057**	**409,398,362**	**495,375,377**
PV of all Costs:	**735,849,057**	**1,145,247,419**	**1,640,622,796**
Total Project Costs Less Benefits:	**-780,000,000**	**640,000,000**	**710,000,000**
Yearly NPV:	**-735,849,057**	**569,507,722**	**596,129,691**
Cumulative NPV:	**-735,849,057**	**-166,251,435**	**429,878,356**
Return on Investment (ROI) di Tahun 3: 26.2%	**429,878,356/1,640,622,796**		**0.262021445**
Break-even Point (BEP): 2.28 tahun	**2 + (596,129,691-429,878,356)/ 596,129,691**		**2.278884507**

Planning

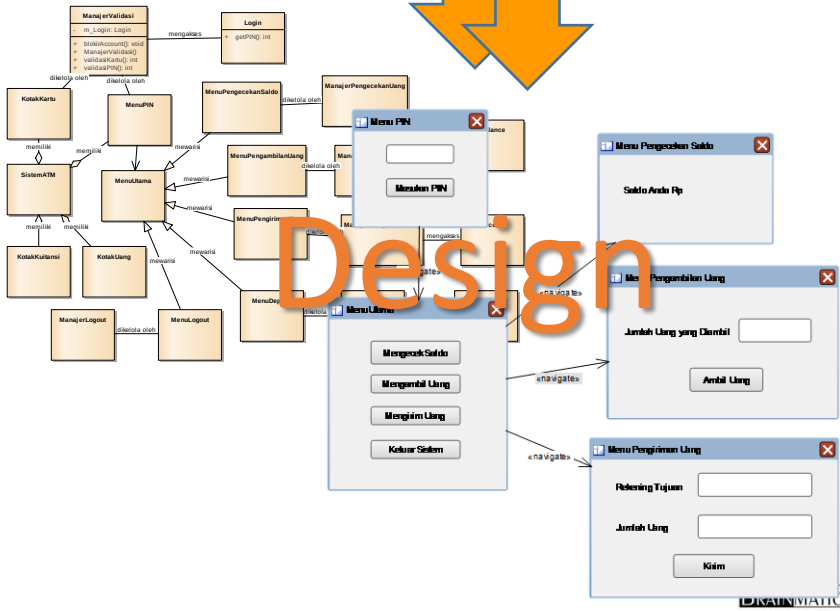


Analysis

Versi 1

Versi 2

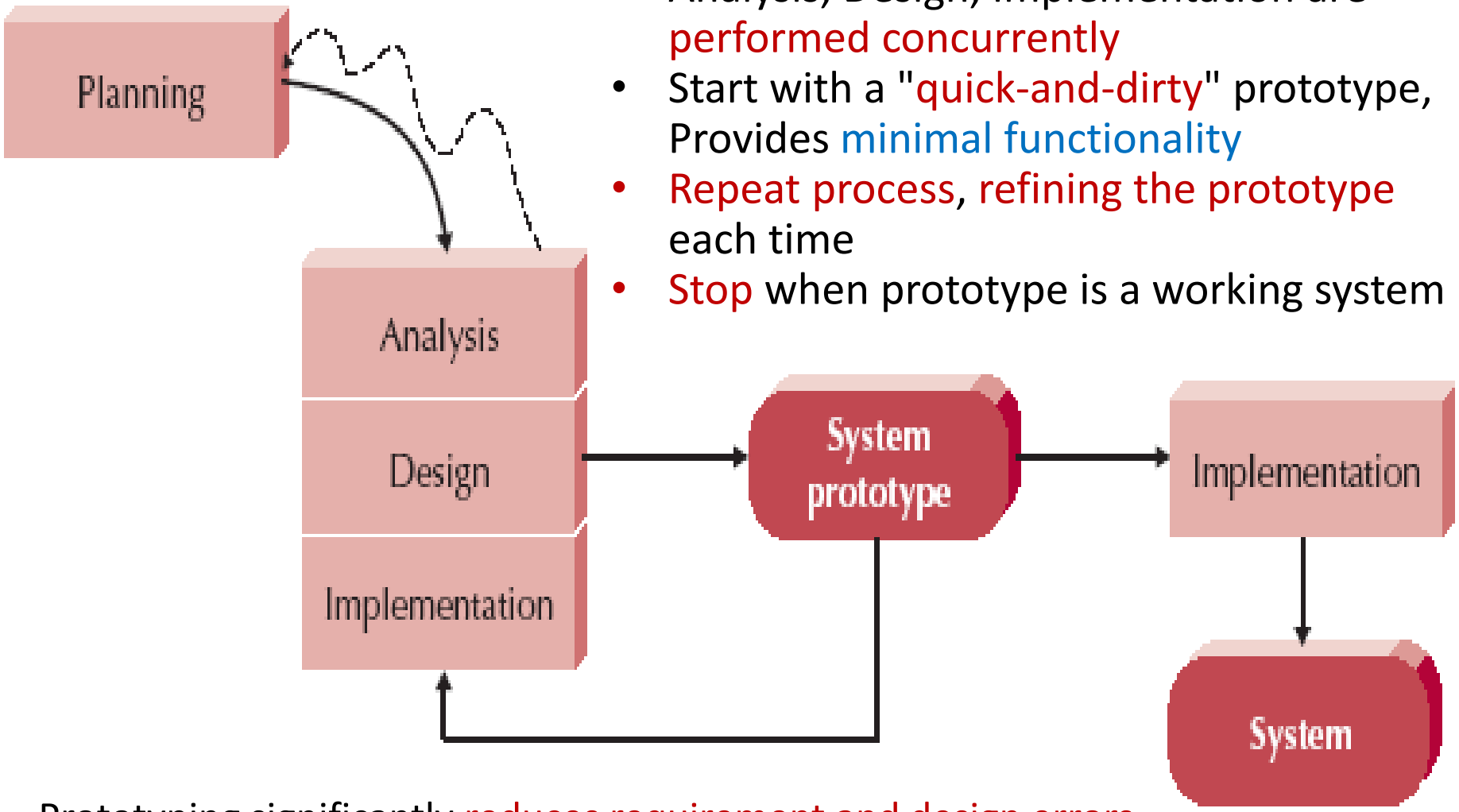
Implementation



Design

2. RAD: Prototyping

- Analysis, Design, Implementation are performed concurrently
- Start with a "quick-and-dirty" prototype, Provides minimal functionality
- Repeat process, refining the prototype each time
- Stop when prototype is a working system



Prototyping significantly reduces requirement and design errors, especially for user interfaces *(Boehm's First Law, Endres, 2013)* [***L3***]

System Request: Sistem Penjualan Musik Online

Project Sponsor:	Margaret Mooney, Vice President of Marketing
Business Needs:	Project ini dibangun untuk:
	1. Mendapatkan pelanggan baru lewat Internet
	2. M

Business Requirement

Sistem yang menduku

1. Fitur Pencarian Pro
2. Fitur Pencarian Tok
3. Fitur Pemesanan Pr
4. Fi

Busi

Intar

Tang

Studi Kelayakan Sistem Penjualan Musik Online

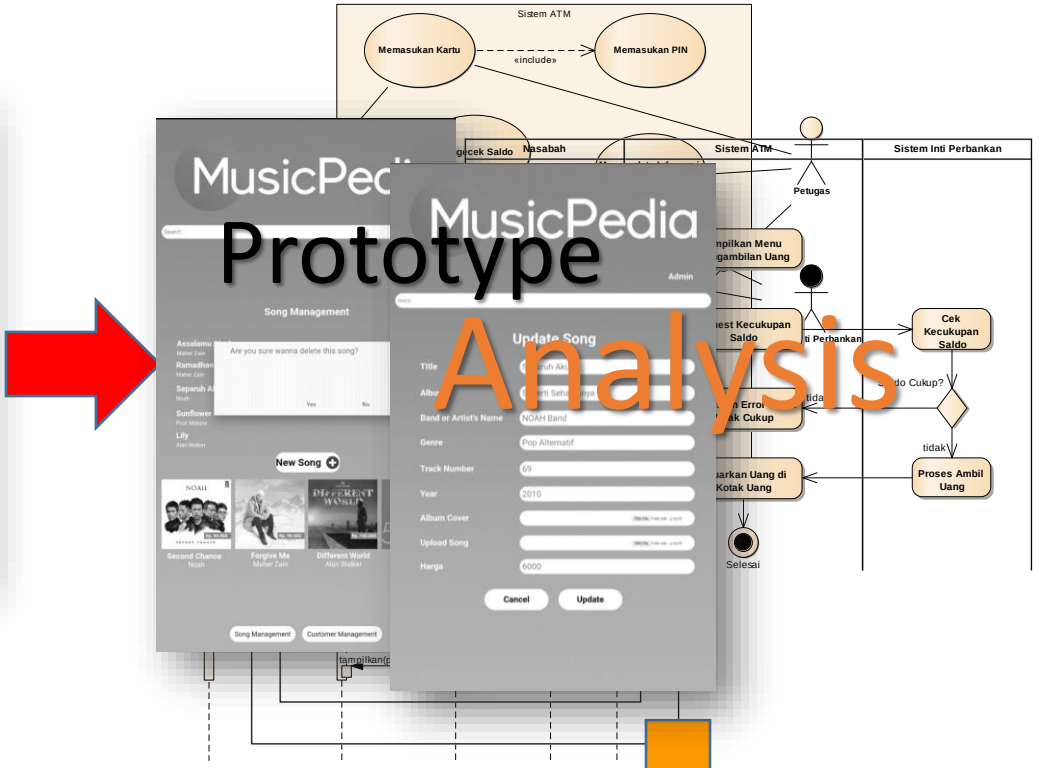
Margaret Mooney dan Alec Adams membuat studi kelayakan untuk pengembangan Sistem Penjualan Musik Online

Kelayakan Teknis

Sistem penjualan musik online layak secara teknis, meskipun memiliki beberapa risiko.

	2016	2017	2018
Peningkatan penjualan dari pelanggan baru	0	400,000,000	500,000,000
Peningkatan penjualan dari pelanggan lama	0	600,000,000	700,000,000
Pengurangan biaya operasional dan telepon	0	100,000,000	100,000,000
Total Benefits:	**0**	**1,100,000,000**	**1,300,000,000**
PV of Benefits:	System Re	8,99 084	091
PV of All Benefits:	ness alue)	8,99 084	070,601,1 2
Honor Tim (Planning, A ysis, Design d Implementation)	60,00 ,000	0	0
Honor Konsultan Infra truktur Internet	0,000,000	0	0
Total Development Costs:	**450,000,000**	**0**	**0**
Honor Pengelola Web	60,000,000	70,000,000	80,000,000
Biaya Lisensi Software	50,000,000	60,000,000	70,000,000
Hardware upgrades	100,000,000	100,000,000	100,000,000
Biaya Komunikasi	20,000,000	30,000,000	40,000,000
Biaya Marketing	100,000,000	200,000,000	300,000,000
Total Operational Costs:	**330,000,000**	**460,000,000**	**590,000,000**
Total Costs:	**780,000,000**	**460,000,000**	**590,000,000**
PV of Costs:	**735,849,057**	**409,398,362**	**495,375,377**
PV of all Costs:	**735,849,057**	**1,145,247,419**	**1,640,622,796**
Total Project Costs Less Benefits:	**-780,000,000**	**640,000,000**	**710,000,000**
Yearly NPV:	**-735,849,057**	**569,507,722**	**596,129,691**
Cumulative NPV:	**-735,849,057**	**-166,251,335**	**429,878,356**
Return on Investment (ROI) di Tahun 3: 26.2%	**429,878,356/1,640,622,796**		**0.262021445**
Break-even Point (BEP): 2.28 tahun	**2 + (596,129,691-429,878,356)/ 596,129,691**		**2.278884507**

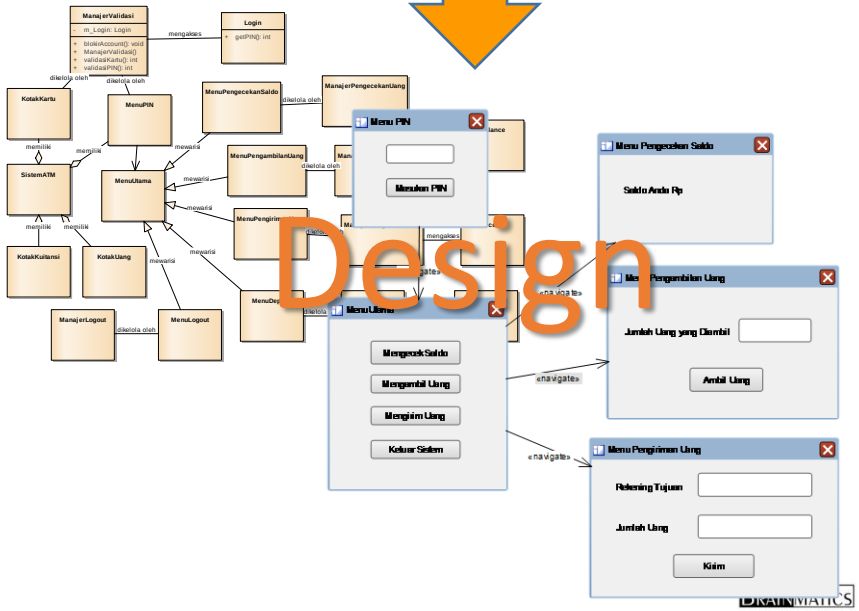
Planning



Prototype

Analysis

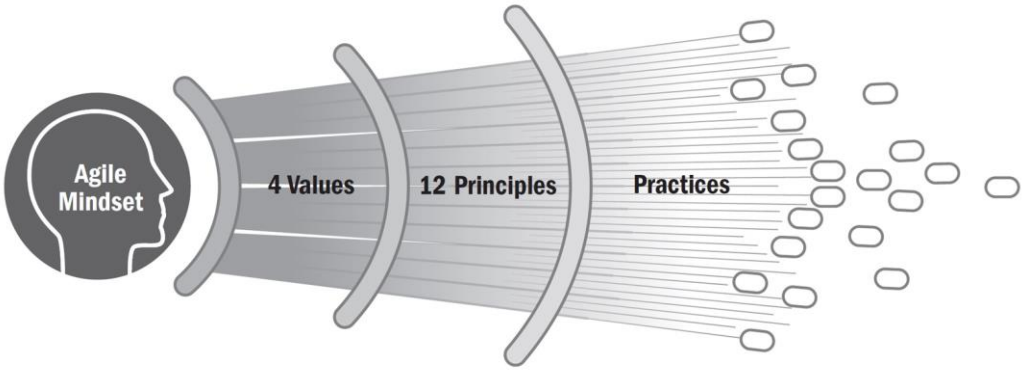
Implementation



Design

3. Agile Development

- Agile Approach:
  1. Agile Values
  2. Agile Principles
  3. Agile Practices



- Mempercepat proses SDLC dengan konsep (Values)
  - Mengurangi pemodelan dan dokumentasi
  - Meningkatkan interaksi antar pengembang dan kolaborasi dengan product owner
  - Mengembangkan software dengan simple, iterative dan reponsif terhadap perubahan dari product owner

- Menggunakan beberapa aturan yang mudah dipahami dan diikuti (Agile Practices)

4 Agile Values

Individuals and interactions over

processes and tools

Working software over

comprehensive documentation

Customer collaboration over

contract negotiation

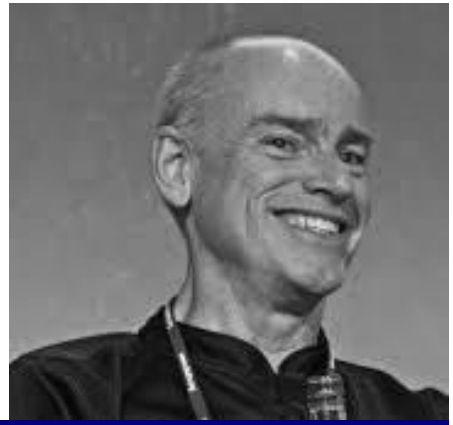
Responding to change over

following a plan

Agile Principles

February 2001, Jeff Sutherland and Ken Schwaber were amongst the 17 software development leaders creating the Manifesto for Agile Software Development

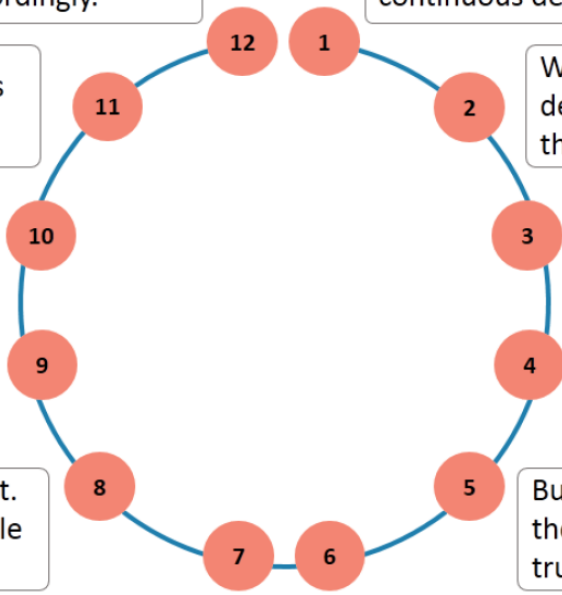
Jeff Sutherland



Ken Schwaber

12 Agile Principles (Manifesto)

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from every couple of weeks to couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity, the art of maximizing the amount of work not done, is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.



Agile Practices

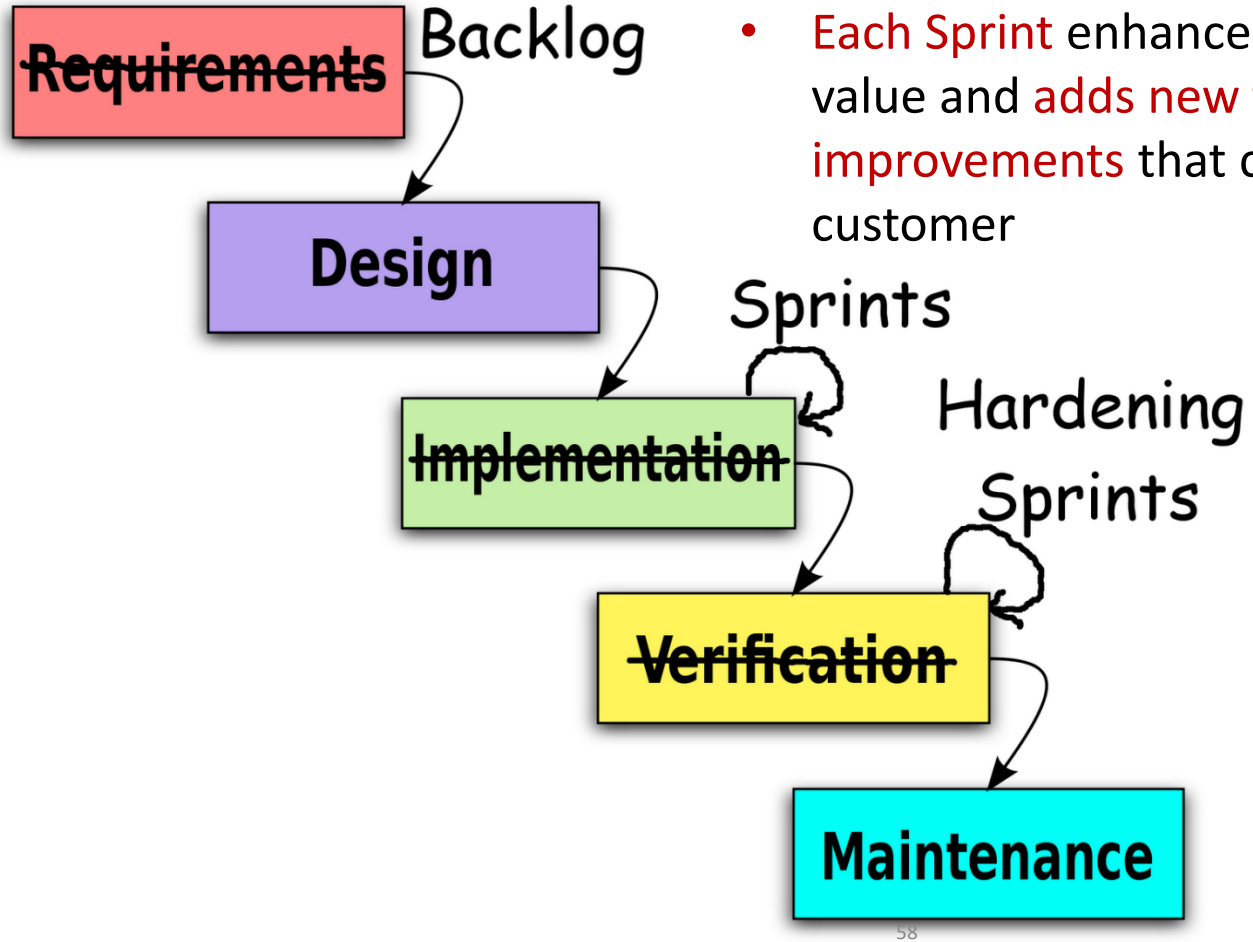
- Extreme Programming (XP) (Kent Beck)
- Scrum (Ken Schwaber and Jeff Sutherland)
- Lean Development (Mary Poppendieck and Tom Poppendieck)
- Dynamic Systems Development Model (DSDM) (Dane Faulkner)

Extreme Programming

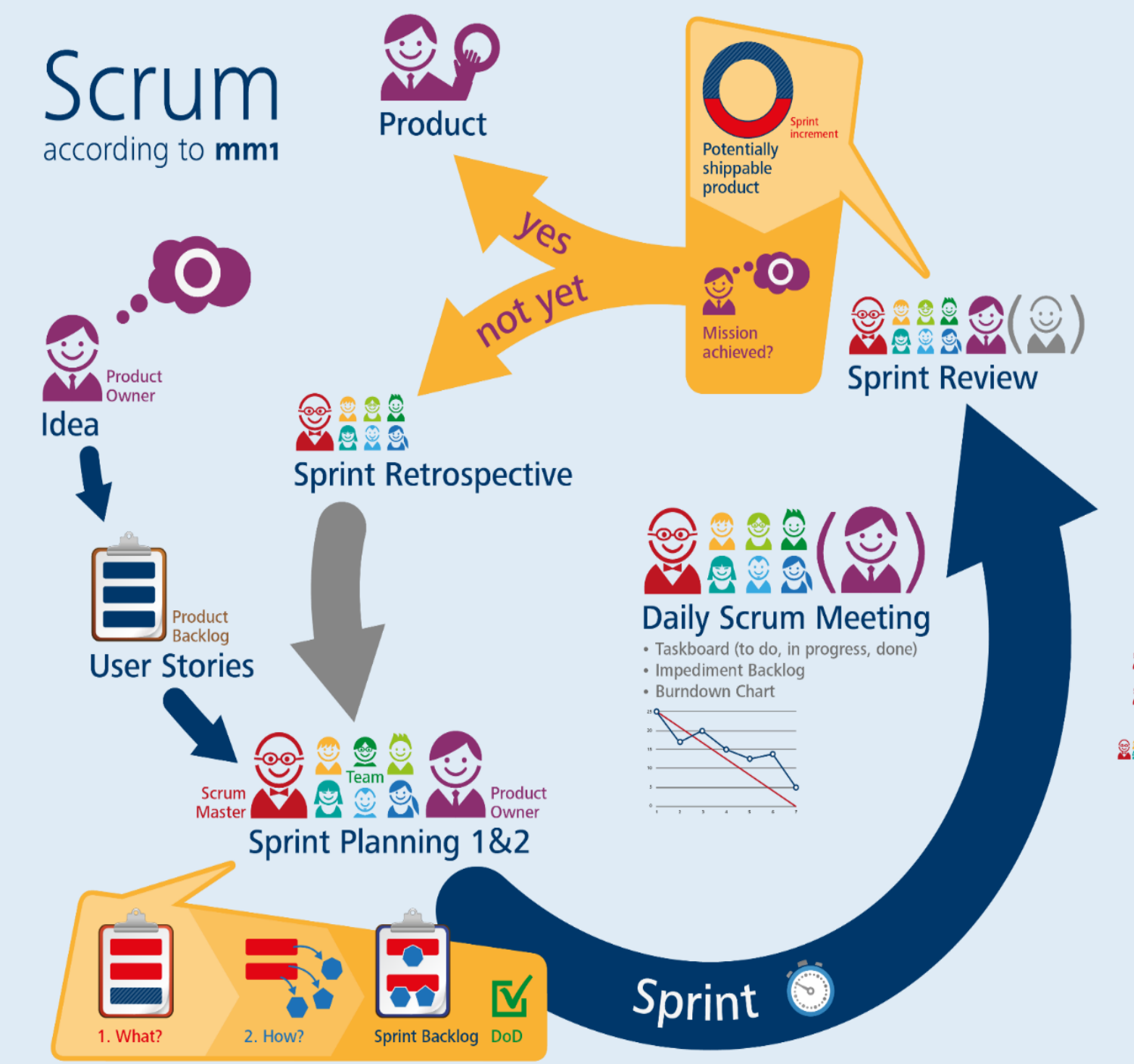
1. Communication: Building software requires communicating requirements to the developers
   1. Pair Programming
   2. Communication replace documentation
2. Simplicity: Encourages starting with the simplest solution, extra functionality can then be added later
3. Feedback:
   1. Feedback from the system: by writing unit tests, or running periodic integration tests, the programmers have direct feedback from the state of the system after implementing changes
   2. Feedback from the customer: The acceptance tests are planned once in every two or three weeks so the customer can easily steer the development
   3. Feedback from the team: When customers come up with new requirements in the planning game the team directly gives an estimation of the time that it will take to implement
4. Courage: Several practices embody courage. One is the commandment to always design and code for today and not for tomorrow

Scrum

- Project members form a Scrum Team consisting of 3-9 people
- The goal of the Sprint is determined and the prioritized functionality is broken down into detailed tasks
- The team is self-organized and the members have a joint responsibility for the results
- Each Sprint enhances the product's market value and adds new functions and improvements that can be delivered to the customer



Scrum
according to **mm1**



Roles

Product Owner: the person responsible for maintaining the product backlog by representing the interests of the stakeholders, ensuring the value of the work the development team does.

Scrum Master: the person responsible for the scrum process, making sure it is used correctly and maximizing its benefits. Although the designation of a scrum master and its presence in scrum meetings is generally advisable, teams with a lot of scrum experience may also work without this role.

Development Team: a cross-functional group of people responsible for delivering potentially shippable increments of the product at the end of every sprint.

Stakeholders: are the people who enable the project and for whom the project produces the agreed upon benefit. They are only directly involved in the process during the sprint reviews. The main stakeholders are managers, customer and user.

Artifacts

Product Backlog: an ordered list of "requirements" that is maintained for a product. The features are commonly written in user story format. It is open and editable by anyone, but the product owner is ultimately responsible for ordering the stories. The product backlog contains rough estimates of both business value and development effort.

Sprint Backlog: a list of work the development team must address during the next sprint. The list is created by selecting stories/features from the top of the product backlog until the development team feels it has enough work to fill the sprint, keeping in mind the velocity of its previous sprints. The stories/features are broken down into tasks by the development team. Often an accompanying task board is used to see and change the state of the tasks of the current sprint, like "to do", "in progress" and "done".

Story/feature: a description of a certain product feature or behavior. Ideally, it is formulated strictly from the user's point of view ("user story").

Task: a unit of work which should be feasible within 12 hours or less, and which must be accomplished in order to implement a story/feature.

Burn Down Charts: are publicly displayed charts showing invested and remaining work. They are often used to visualize the sprint progress as sprint burn down chart. Other types comprise the release burn down chart that shows the amount of work left to complete the target commitment for a Product Release.

Impediment Backlog: list of current impediments maintained by the scrum master.

Definition of Done: a checklist of activities required to declare the implementation of a story to be completed. The definition is determined at the beginning of but may be changed in the course of the project.

Meetings

Sprint Planning 1: (60 min per sprint week) it held to select the work to be done for the next sprint (the "what"). the product owner explains the stories of the product backlog to the team and answers their question. After this analysis phase the team should have understood the requirements and its comes in the scope for the sprint.

Sprint Planning 2: (60 min per sprint week) the designing phase for the selected backlog (the "how"). The team discusses a solution for the selected stories and creates according task for each story.

Daily Scrum: (ca. 15 min) short, time boxed meeting, every day at the same time. Every team member answers three questions:
1) What have I done since yesterday?
2) What am I planning to do today?
3) What are my impediments?

Sprint Review: (ca. 60 min per sprint week) used to present and review the work that was completed and not completed during a sprint. It should include a demonstration of the realized product increment.

Sprint Retrospective: (ca. 45 min per sprint week) a reflection on the past sprint used to make continuous process improvements. Two main questions are asked in the sprint retrospective:
1) What went well during the sprint?
2) What could be improved in the next sprint?

Estimation Meeting: (max. 60 min) used to introduce and estimate new backlog items and to refine existing estimations as well as acceptance criteria. It is also used to break large stories into smaller ones.

mm1
Consulting & Management

System Request: Sistem Penjualan Musik Online

Project Sponsor: Margaret Mooney, Vice President of Marketing

Business Needs: Project ini dibangun untuk:

1. Mendapatkan pelanggan baru lewat Internet

Studi Kelayakan Sistem Penjualan Musik Online

Margaret Mooney dan Alec Adams membuat studi kelayakan untuk pengembangan Sistem Penjualan Musik Online

Kelayakan Teknis

Sistem penjualan musik online layak secara teknis, meskipun memiliki beberapa risiko.

	2016	2017	2018
Peningkatan penjualan dari pelanggan baru	0	400,000,000	500,000,000
Peningkatan penjualan dari pelanggan lama	0	600,000,000	700,000,000
Pengurangan biaya operasional dan telepon	0	100,000,000	100,000,000
Total Benefits:	**0**	**1,100,000,000**	**1,300,000,000**
PV of Benefits:			
PV of All Benefits:			
Honor Tim (Planning, Analysis, Design and Implementation)			
Honor Konsultan Infrastruktur Internet			
Total Development Costs:	**450,000,000**	**0**	**0**
Honor Pengelola Web	60,000,000	70,000,000	80,000,000
Biaya Lisensi Software	50,000,000	60,000,000	70,000,000
Hardware upgrades	100,000,000	100,000,000	100,000,000
Biaya Komunikasi	20,000,000	30,000,000	40,000,000
Biaya Marketing	100,000,000	200,000,000	300,000,000
Total Operational Costs:	**330,000,000**	**460,000,000**	**590,000,000**
Total Costs:	**780,000,000**	**460,000,000**	**590,000,000**
PV of Costs:	**735,849,057**	**409,398,362**	**495,375,377**
PV of all Costs:	**735,849,057**	**1,145,247,419**	**1,640,622,796**
Total Project Costs Less Benefits:	**-780,000,000**	**640,000,000**	**710,000,000**
Yearly NPV:	**-735,849,057**	**569,507,722**	**596,129,691**
Cumulative NPV:	**-735,849,057**	**-166,251,435**	**429,878,356**
Return on Investment (ROI) di Tahun 3: 26.2%	**429,878,356/1,640,622,796**		**0.262021445**
Break-even Point (BEP): 2.28 tahun	**2 + (596,129,691-429,878,356)/ 596,129,691**		**2.278884507**

Planning

Sprint 1

Analysis

Sprint 2

Hasil Sprint 1

Hasil Sprint 2

Implementation

Design

Faktor Penentu Pemilihan Metodologi

1. Clarity of User Requirements
2. Familiarity with Technology
3. System Complexity
4. System Reliability
5. Short Time Schedules
6. Schedule Visibility

Ability to Develop Systems	Structured Methodologies		RAD Methodologies			Agile Methodologies	
	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	XP	SCRUM
With Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent	Excellent
With Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Good	Good
That Are Complex	Good	Good	Good	Poor	Poor	Good	Good
That Are Reliable	Good	Good	Good	Poor	Poor	Excellent	Excellent
With a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent	Excellent
With Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Excellent	Excellent

Pemilihan Metodologi: Sistem SDM

- Seandainya, anda adalah seorang software engineer di perusahaan PT BlackSoft, sebuah perusahaan IT yang memiliki kantor cabang di berbagai tempat di dunia
- Divisi Sumber Daya Manusia pada PT BlackSoft ingin membangun sebuah sistem yang bisa merekam, mengubah, menghapus dan menampilkan data pegawai yang dimiliki, baik itu lokasi saat ini, latar belakang pendidikan, jadwal pekerjaan dan pengalaman kerja yang dimiliki
- PT BlackSoft memiliki jaringan internasional dimana kantor cabang di berbagai negara menggunakan hardware dan software yang berbeda
- Divisi Sumber Daya Manusia berkantor di Jakarta, sedangkan Divisi Teknologi Informasi yang mendapatkan tugas mengembangkan software berkantor di Denpasar, pertemuan untuk requirement gathering harus dischedulekan sejak lama
- Manajemen ingin agar sistem dapat selesai dikerjakan dan mulai bisa berjalan dalam satu tahun

Pemilihan Metodologi: Sistem DSS

- Seandainya, anda adalah seorang software engineer di perusahaan PT BlackSoft, sebuah perusahaan IT yang memiliki kantor cabang di berbagai tempat di dunia
- Divisi Business Development pada PT BlackSoft ingin membangun sebuah sistem Decision Support System (DSS). DSS akan mengolah berbagai data perusahaan, kemudian mengubahnya menjadi pola-pola pengetahuan, yang akhirnya bisa digunakan untuk pengambilan keputusan manajemen
- Divisi Business Development paham betapa pentingnya sistem ini, karena itu akan all out dan bersedia masuk ke tim pengembangan
- Divisi Business Development belum bisa menjelaskan secara pasti software yang dibangun ini akan seperti apa. Oleh karena itu dibutuhkan diskusi intensif dan mendalam, serta bila perlu ada pertemuan harian, sehingga software bisa dikembangakn sesuai kebutuhan
- Divisi Business Development berharap bahwa sistem dapat selesai dikerjakan dan mulai bisa berjalan dalam enam bulan. Release pertama harus dalam 1 bulan. Untuk itu monitoring progress akan dilakukan secara harian

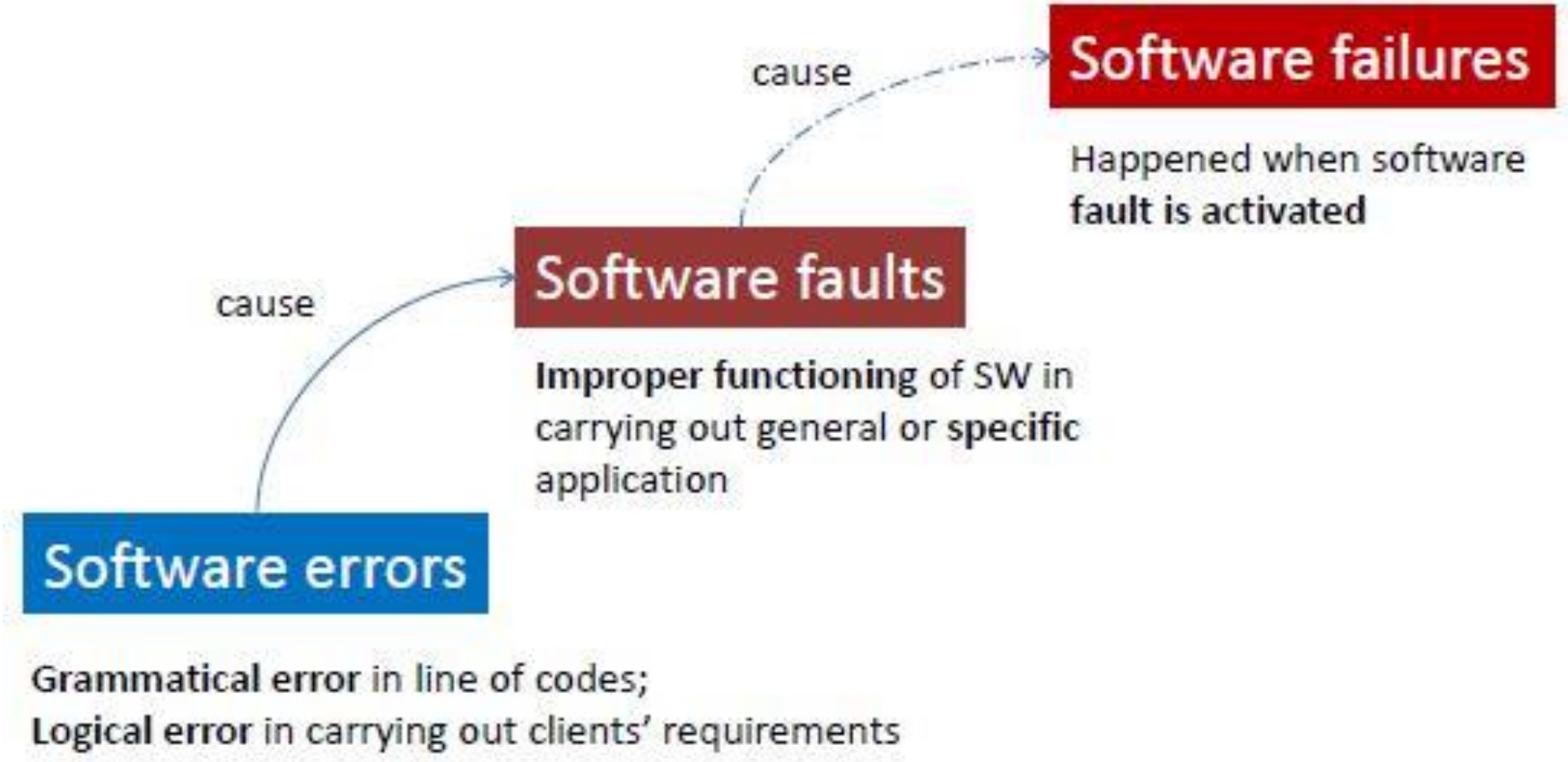
MITOS 5

Project Molor? Tambah Jumlah Pengembang Supaya Bisa Cepat Selesai

Keunikan dari Software

Karakteristik	Software	Hardware
Kompleksitas	Tingkat kompleksitas dari produk software tinggi, dengan kemungkinan perubahan parameter dan fungsi yang sangat beragam	Tingkat kompleksitas produk lain rendah, dengan kemungkinan perubahan parameter dan fungsi tidak beragam
Visibilitas Produk	Produk tidak terlihat dengan kasat mata, termasuk bila ada cacat (*defect*) dari produk	Produk terlihat dengan kasat mata, termasuk bila ada cacat (*defect*) dari produk
Sumber Daya Manusia	Kuantitas SDM Tidak berhubungan dengan Kualitas dan Kecepatan Kerja	Kuantitas SDM berhubungan dengan Kualitas dan Kecepatan Kerja

Software Errors, Faults, Failures



Software Errors != Failures

- Suatu perusahaan PT ABC memproduksi software yang akan ditanam ke dalam suatu device
- Salah satu fungsi yang terdapat pada software adalah akan mematikan device secara otomatis apabila suhu ruangan lebih besar daripada 30^{o} celcius
- Programmer salah menuliskan logika menjadi:

```
...
if (suhu > 3) shutdownDevice();
...
```

- Error ini tidak pernah menyebabkan failure pada software, dan perusahaan PT ABC sampai saat ini terkenal sebagai perusahaan yang memproduksi software tanpa bug
- Jelaskan mengapa bisa terjadi demikian!

Warranty Lawsuits

- Mortenson vs. Timeberline Software (TS) (≈1993)
  - Mortenson menggunakan software yang diproduksi TS untuk membuka tender pembangunan rumah sakit
  - Software memiliki bug sehingga memenangkan perusahaan yang mengajukan proposal paling mahal (kerugian 2 miliar USD)
  - TS tahu tentang bug itu, tapi tidak mengirimkan update ke Mortenson
  - Pengadilan di Amerika Serikat memenangkan perusahaan TS

- Uniform Computer Information Transaction Act (UCITA) allows software manufacturers to:
  - disclaim all liability for defects
  - prevent the transfer of software from person to person

Disclaimer of Warranties

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, MICROSOFT AND ITS SUPPLIERS PROVIDE TO YOU THE SOFTWARE COMPONENT, AND ANY (IF ANY) SUPPORT SERVICES RELATED TO THE SOFTWARE COMPONENT ("SUPPORT SERVICES") **AS IS AND WITH ALL FAULTS**; AND MICROSOFT AND ITS SUPPLIERS HEREBY DISCLAIM WITH RESPECT TO THE SOFTWARE COMPONENT AND SUPPORT SERVICES ALL WARRANTIES AND CONDITIONS, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY (IF ANY) WARRANTIES OR CONDITIONS OF OR RELATED TO: TITLE, NON-INFRINGEMENT, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, LACK OF VIRUSES, ACCURACY OR COMPLETENESS OF RESPONSES, RESULTS, LACK OF NEGLIGENCE OR LACK OF WORKMANLIKE EFFORT, QUIET ENJOYMENT, QUIET POSSESSION, AND CORRESPONDENCE TO DESCRIPTION. **THE ENTIRE RISK ARISING OUT OF USE OR PERFORMANCE OF THE SOFTWARE COMPONENT AND ANY SUPPORT SERVICES REMAINS WITH YOU**.

Brooks' Law (1975)

Adding manpower to a late project makes it later

(Endres, 2003) *[L36]*

Brooks' Law (1975)

Just because a woman can make a baby in nine months, it does not follow that nine women can make a baby in one month

Adagium di Pengembangan Software

- Project Manager is a person who thinks nine women can deliver a baby in one month
- Marketing Manager is a person who thinks he can deliver a baby even if no man and woman are available
- Human Resource is a person who thinks that a donkey can deliver a human baby if given 9 months
- Client is the one who doesn't know why he wants a baby
- Developer is a person who thinks it will take 18 months to deliver a baby
- Tester is a person who always tells his wife that this is not the right baby

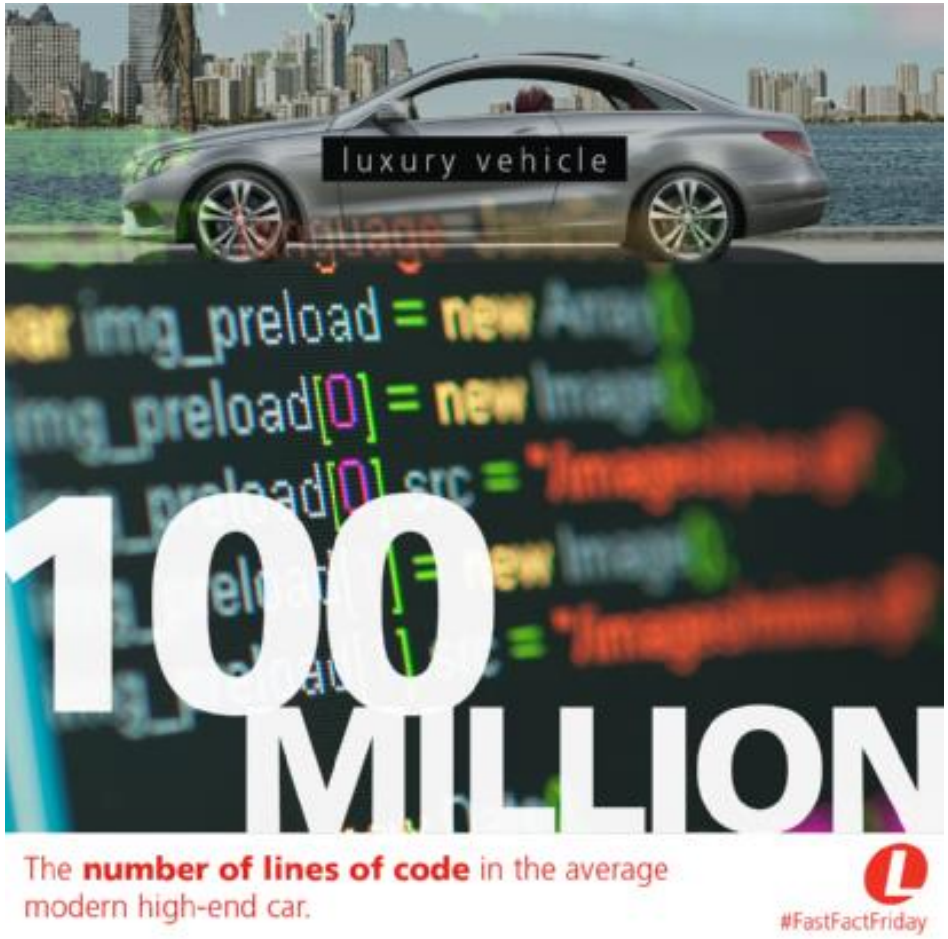
MITOS 6

Penghitungan Cost Pengembangan Software itu Pakai Insting dan Tidak Ada Cara Pasti atau Teorinya

“Size” of Software Systems

Year	Operating System	SLOC (Million)
1993	Windows NT 3.1	4-5[1]
1994	Windows NT 3.5	7–8[1]
1996	Windows NT 4.0	11–12[1]
2000	Windows 2000	more than 29[1]
2001	Windows XP	45[2][3]
2003	Windows Server 2003	50[1]

Source: Wikipedia



“Size” of Software Systems

Organizations	Function Points	Lines of Code
Internal Revenue Service	150,000	7,500,000
Banks	125,000	6,250,000
Insurance companies	125,000	6,250,000
Credit card companies	125,000	6,250,000
Credit bureaus	120,000	6,000,000
Census Bureau	100,000	5,000,000
State tax boards	90,000	4,500,000
Airlines	75,000	3,750,000
Police organizations	75,000	3,750,000
Hospitals	75,000	3,750,000
Web-based stores	75,000	3,750,000
Municipal tax boards	50,000	2,500,000
Motor vehicle department	50,000	2,500,000
Physicians offices	30,000	1,500,000
Dental offices	30,000	1,500,000
Schools/universities	25,000	1,250,000
Clubs and associations	20,000	1,000,000
Retail stores	20,000	1,000,000

Caper Jones, The Economic of Software Quality (2012)

Software Size (Effort) Estimation Methods

1. Simply Method

(Industry Std Percentages)

1. Use the time spent for planning
2. Along with industry standard percentages
3. Estimate the overall time for the project

2. Function Points

(Allen Albrecht, 1979)

1. Estimate System Size (Function Points)
2. Estimate Effort Required (Person-Month)
3. Estimate Time Required (Month)

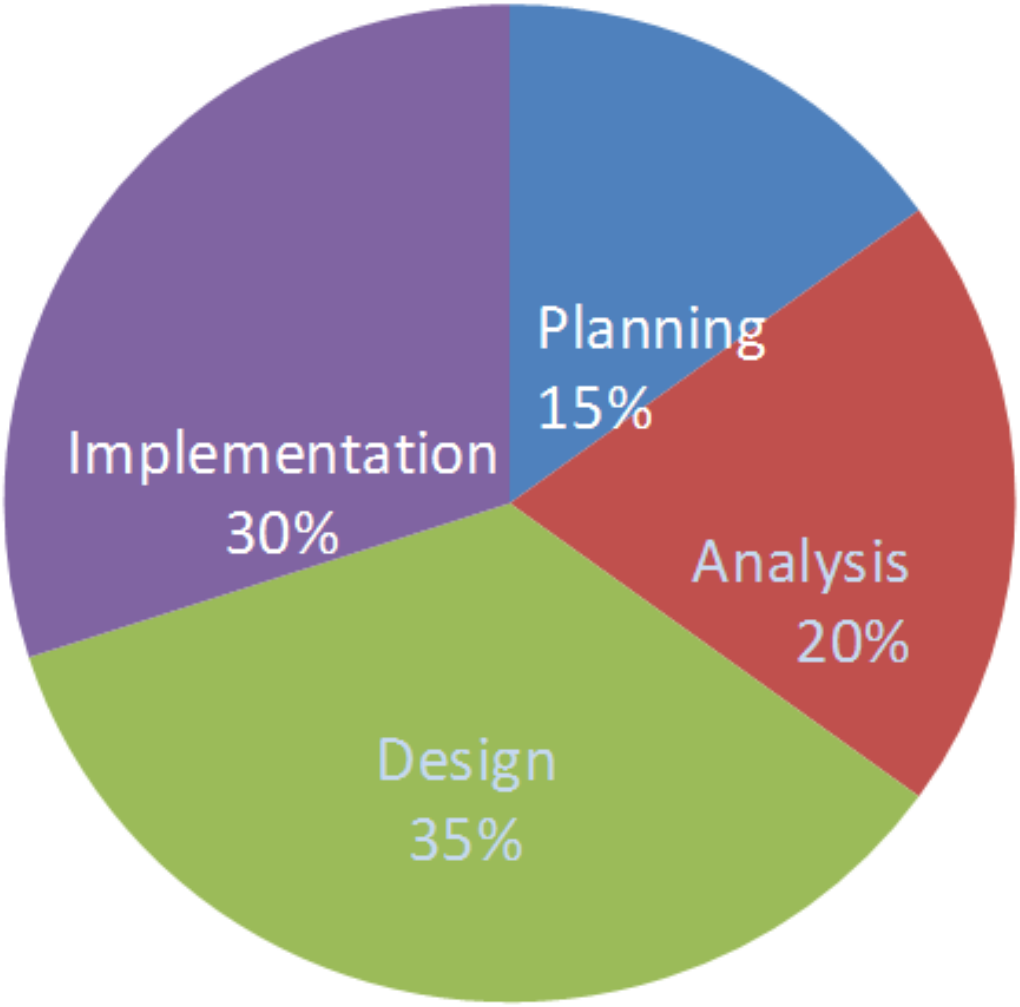
3. Use Case Points

(Gustav Karner, 1993)

1. Estimate System Size (Use Case Points)
2. Estimate Effort Required (Person-Month)
3. Estimate Time Required (Month)

1. Simply Method

Distribusi Effort Pengembangan Software



(Dennis, 2016) *(Jones, 2012)*

Estimate the Overall Time

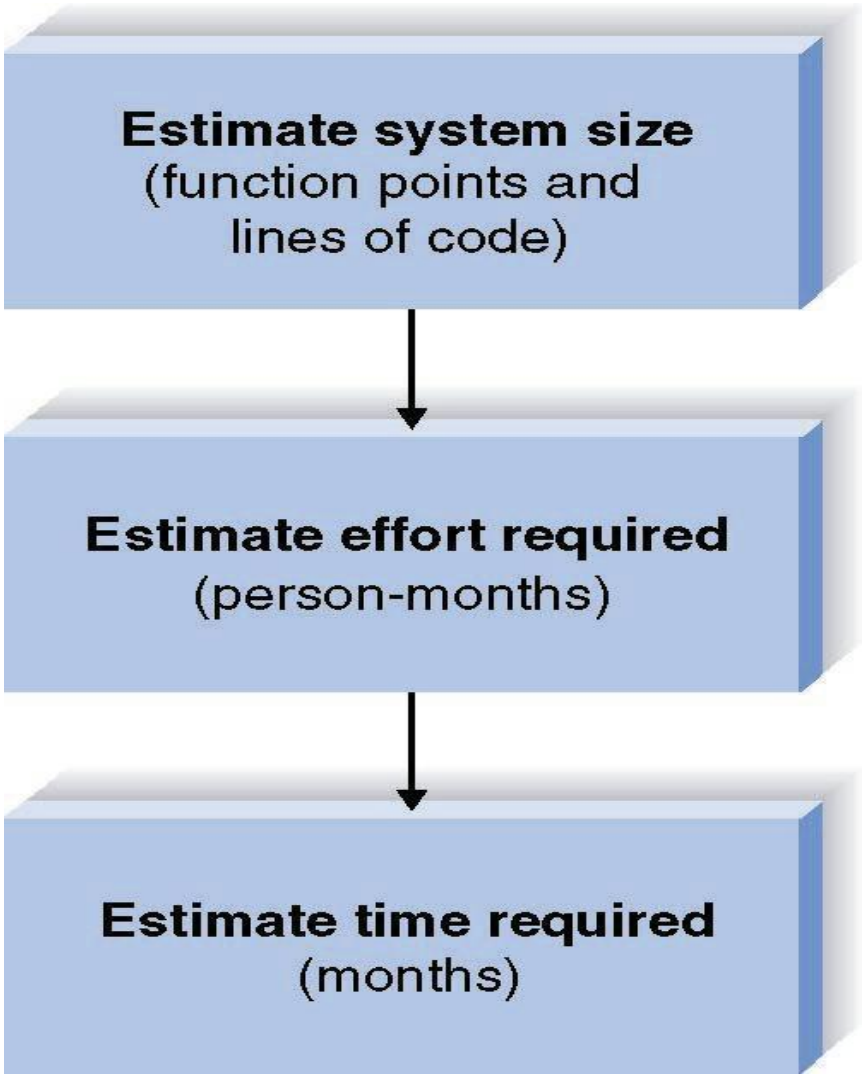
	Planning (Actual)	Analysis	Design	Implementation
Effort Distribution	15%	20%	35%	30%
Effort in Time (Month)	4 month	5.33 month	9.33 month	8 month
Effort in Person (Man)	2 person	2.6 person	4.6 person	4 person
Formula per Phase	Actual Time and Person	0.2 * (Planning/0.15)	0.35 * (Planning/0.15)	0.3 * (Planning/0.15)
Overall Time and Person	Planning/0.15			

$$\text{Planning time} = 0.15 \times \text{Overall time}$$

$$\text{Overall time} = \frac{\text{Planning time}}{0.15} \qquad \text{Analysis time} = 0.2 \times \frac{\text{Planning time}}{0.15}$$

2. Function Points

Function Point



(Allen Albrecht, 1979)

A. Function Points Estimation:
Step One (TUFP)

Complexity				
Description	Low	Medium	High	Total
Inputs	__x 3	__x 4	__x 6	____
Outputs	__x 4	__x 5	__x 7	____
Queries	__x 3	__x 4	__x 6	____
Files	__x 7	__x 10	__x 15	____
Program Interfaces	__x 5	__x 7	__x 10	____
TOTAL UNADJUSTED FUNCTION POINTS				____

Example: CD Selection System

System Components:

Description		Complexity			
	Total Number	Low	Medium	High	Total
Inputs	6	3 × 3	2 × 4	1 × 6	23
Outputs	19	4 × 4	10 × 5	5 × 7	101
Queries	10	7 × 3	0 × 4	3 × 6	39
Files	15	0 × 7	15 × 10	0 × 15	150
Program Interfaces	3	1 × 5	0 × 7	2 × 10	25
Total Unadjusted Function Points (TUFP):					338

Function Points Estimation:
Step Two (Processing Complexity)

	Scale of 0 to 3
Data Communications	_____
Heavy Use Configuration	_____
Transaction Rate	_____
End-User efficiency	_____
Complex Processing	_____
Installation Ease	_____
Multiple sites	_____
Performance	_____
Distributed functions	_____
On-line data entry	_____
On-line update	_____
Reusability	_____
Operational Ease	_____
Extensibility	_____
Processing Complexity (PC)	_____

Example: Sistem Penjualan Musik Online

Data communications	3
Heavy use configuration	0
Transaction rate	0
End-user efficiency	0
Complex processing	0
Installation ease	0
Multiple sites	0
Performance	0
Distributed functions	2
Online data entry	2
Online update	0
Reusability	0
Operational ease	0
Extensibility	0
Total Processing Complexity (PC):	7

Function Point Estimation:
Step Three (TAFP)

Processing Complexity (PC) = 7
(From Step Two)

Adjusted Processing
Complexity (PCA) = 0.65 + (0.01 * 7) = 0.72

Total Adjusted
Function Points (TAFP): 338 * 0.72 = 243
(From Step One)

Adjusted Processing Complexity

Choose standard Adjusted Project Complexity (PCA) from the range:

1. 0.65	Simple systems
2. 1.0	"Normal" systems
3. 1.35	Complex systems

Konversi Function Points ke Lines of Code

Language	LOC/TAFP
C	130
COBOL	110
JAVA	55
C++	50
Turbo Pascal	50
Visual Basic	30
PowerBuilder	15
HTML	15
Packages (e.g., Access, Excel)	10-40

Source: Capers Jones, Software Productivity Research

Language	QSM SLOC/FP Data			
	Avg	Median	Low	High
ABAP (SAP) *	28	18	16	60
ASP*	51	54	15	69
Assembler *	119	98	25	320
Brio +	14	14	13	16
C *	97	99	39	333
C++ *	50	53	25	80
C# *	54	59	29	70
COBOL *	61	55	23	297
Cognos Impromptu Scripts +	47	42	30	100
Cross System Products (CSP) +	20	18	10	38
Cool:Gen/IEF *	32	24	10	82
Datastage	71	65	31	157
Excel *	209	191	131	315
Focus *	43	45	45	45
FoxPro	36	35	34	38
HTML *	34	40	14	48
J2EE *	46	49	15	67
Java *	53	53	14	134
JavaScript *	47	53	31	63
JCL *	62	48	25	221
LINC II	29	30	22	38
Lotus Notes *	23	21	19	40
Natural *	40	34	34	53
.NET *	57	60	53	60

Lines of Codes (LOC)

Line of Codes (LOC) = TAFP * LOC/TAFP

Example:

If TAFP = 243 Then we build the software using Java

*LOC = 243 * 55 = 13365 line of codes*

Contoh Jenis Aplikasi dan FP

Organizations	Function Points	Lines of Code
Internal Revenue Service	150,000	7,500,000
Banks	125,000	6,250,000
Insurance companies	125,000	6,250,000
Credit card companies	125,000	6,250,000
Credit bureaus	120,000	6,000,000
Census Bureau	100,000	5,000,000
State tax boards	90,000	4,500,000
Airlines	75,000	3,750,000
Police organizations	75,000	3,750,000
Hospitals	75,000	3,750,000
Web-based stores	75,000	3,750,000
Municipal tax boards	50,000	2,500,000
Motor vehicle department	50,000	2,500,000
Physicians offices	30,000	1,500,000
Dental offices	30,000	1,500,000
Schools/universities	25,000	1,250,000
Clubs and associations	20,000	1,000,000
Retail stores	20,000	1,000,000

Caper Jones, The Economic of Software Quality (2012)

B. Estimating Effort

Effort = 1.4 * thousands-of- lines-of-code
(in Person- Months)

Example:

If LOC = 13365 Then...
*Effort = 1.4 * 13365/1000 = 18.711 Person Months*

C. Estimating Time

Time (in Months) = $3.0 * \text{person-months}^{1/3}$

Example:

If LOC = 13365 Then...

*Effort = 1.4 * 13.365 = 18.711 person-months*

Time = $3.0 * 18.711^{1/3}$ = 7.9 month

Boehm's Third Law (1981):

Development effort is a non-linear function of product size

Hitung Size dari Sistem dengan Function Point (TUFP – TAFP – LOC – PM – M)

- Sebuah perusahaan membutuhkan sistem *job seeker* untuk pencari kerja dan perusahaan pembuka lowongan pekerjaan
- Sistem memungkinkan pencari kerja untuk **menginput** data curriculum vitae. Di sisi lain, perusahan pembuka lowongan kerja bisa menginput data perusahaan dan lowongan pekerjaan yang disediakan
- Pencari kerja dapat melakukan **pencarian (query)** tentang lowongan pekerjaan apa saja yang tersedia, pembuka lowongan kerja dapat mencari siapa saja yang sudah mendaftar di suatu lowongan pekerjaan
- Sistem mampu **memproduksi laporan** statistik lengkap tentang pencari kerja, perusahaan, jenis lowongan pekerjaan dan tren lowongan kerja yang sedang populer
- Laporan statistik disajikan dalam dua format file, bentuk infografik (image), juga tersedia dalam bentuk file pdf yang bisa didownload (file)
- Sistem akan dikembangkan dengan menggunakan bahasa Java

TUFP

Complexity				
Description	Low	Medium	High	Total
Inputs	__x 3	__x 4	__x 6	____
Outputs	__x 4	__x 5	__x 7	____
Queries	__x 3	__x 4	__x 6	____
Files	__x 7	__x 10	__x 15	____
Program Interfaces	__x 5	__x 7	__x 10	____
		TOTAL UNADJUSTED FUNCTION POINTS		____

TUFP

	Fungsi	Bobot (Low)	Total
Input	3	3	9
Output	4	4	16
Queries	2	3	6
File	2	7	14
Program Interface	5	5	25
TUFP			**70**

Processing Complexity

Data Communications	1
Heavy Use Configuration	0
Transaction Rate	0
End-User efficiency	0
Complex Processing	0
Installation Ease	0
Multiple sites	0
Performance	0
Distributed functions	0
On-line data entry	2
On-line update	2
Reusability	0
Operational Ease	1
Extensibility	0

TOTAL: 6

TAFP

1. Processing Complexity (PC) = 6

2. Adjusted Processing Complexity (PCA)

   0.65 + (0.01 * 6) = 0.71

3. Total Adjusted Function Points (**TAFP**)

   70 * 0.71 = **49.7**

LOC → Effort (ManMonth)→ Time (Month)

1. LOC = 55*49.7 = 2733.5

2. Effort = 1.4*2.7335 = 3.83 MM

3. Time = 3.0 * 3.83 $^{(1/3)}$ = 4.6 M

Example: Schedule

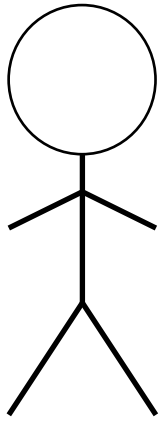
	Fase	M	PM	Cost
1	Planning	1	3	15
2	Analysis	2	3	30
3	Design	2	4	50
4	Implementation	4	5	80
				175jt

3. Use Case Points

Use Case Points

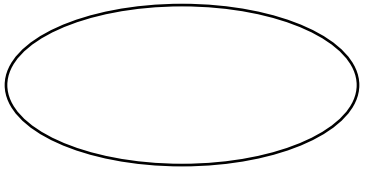
Unadjusted Actor Weighting (UAW)

Actor Type	Description	Weighting Factor
Simple (Easy)	External System with well-defined API	1
Average (Medium)	External System using a protocol-based interface, e.g., HTTP, TCT/IP, SQL	2
Complex (Difficult)	Human	3



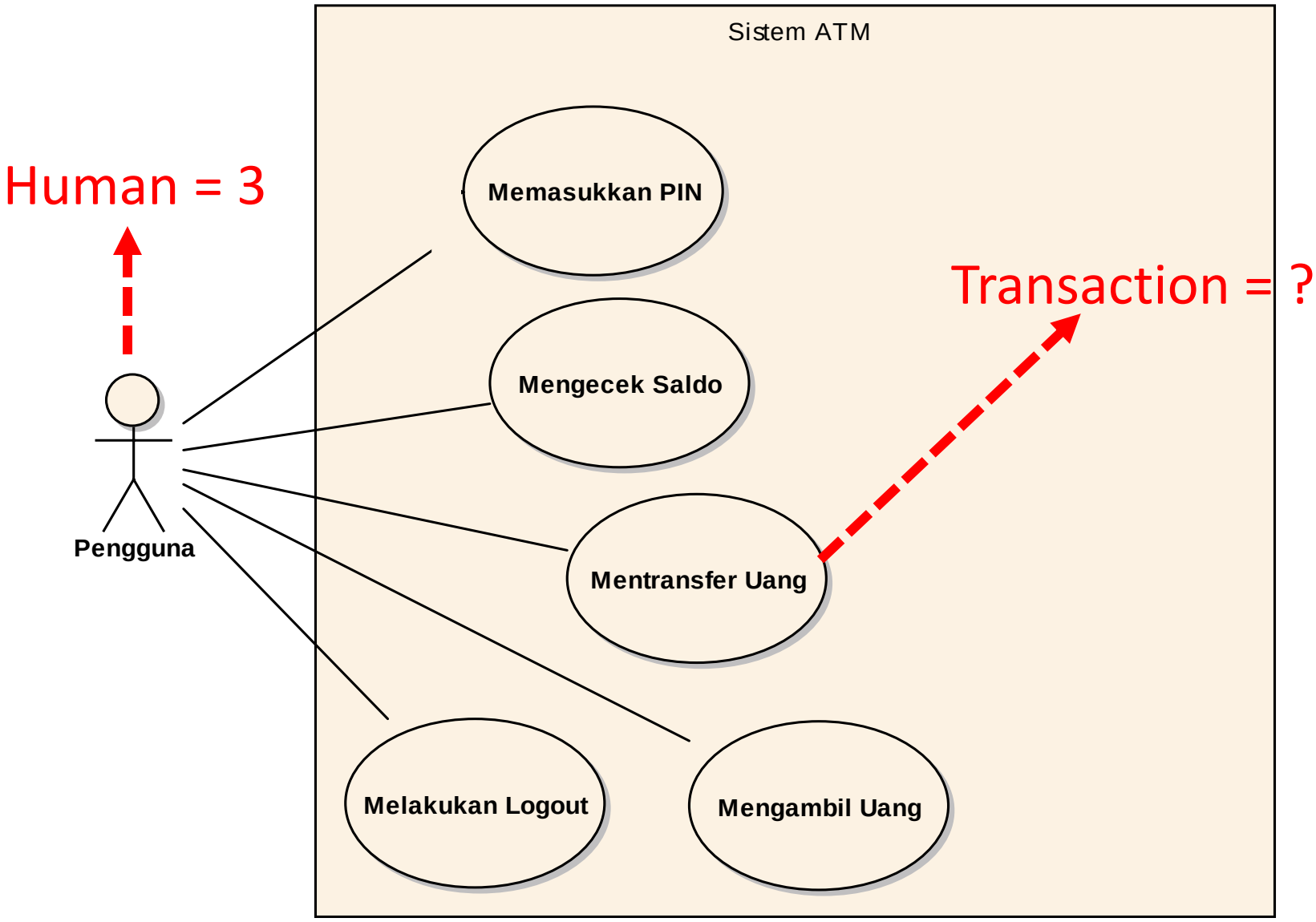
Unadjusted Use Case Weighting (UUCW)

Use-Case Type	Description	Weighting Factor
Simple (Easy)	1-3 transactions	5
Average (Medium)	4-7 transactions	10
Complex (Difficult)	More than 7 transactions	15

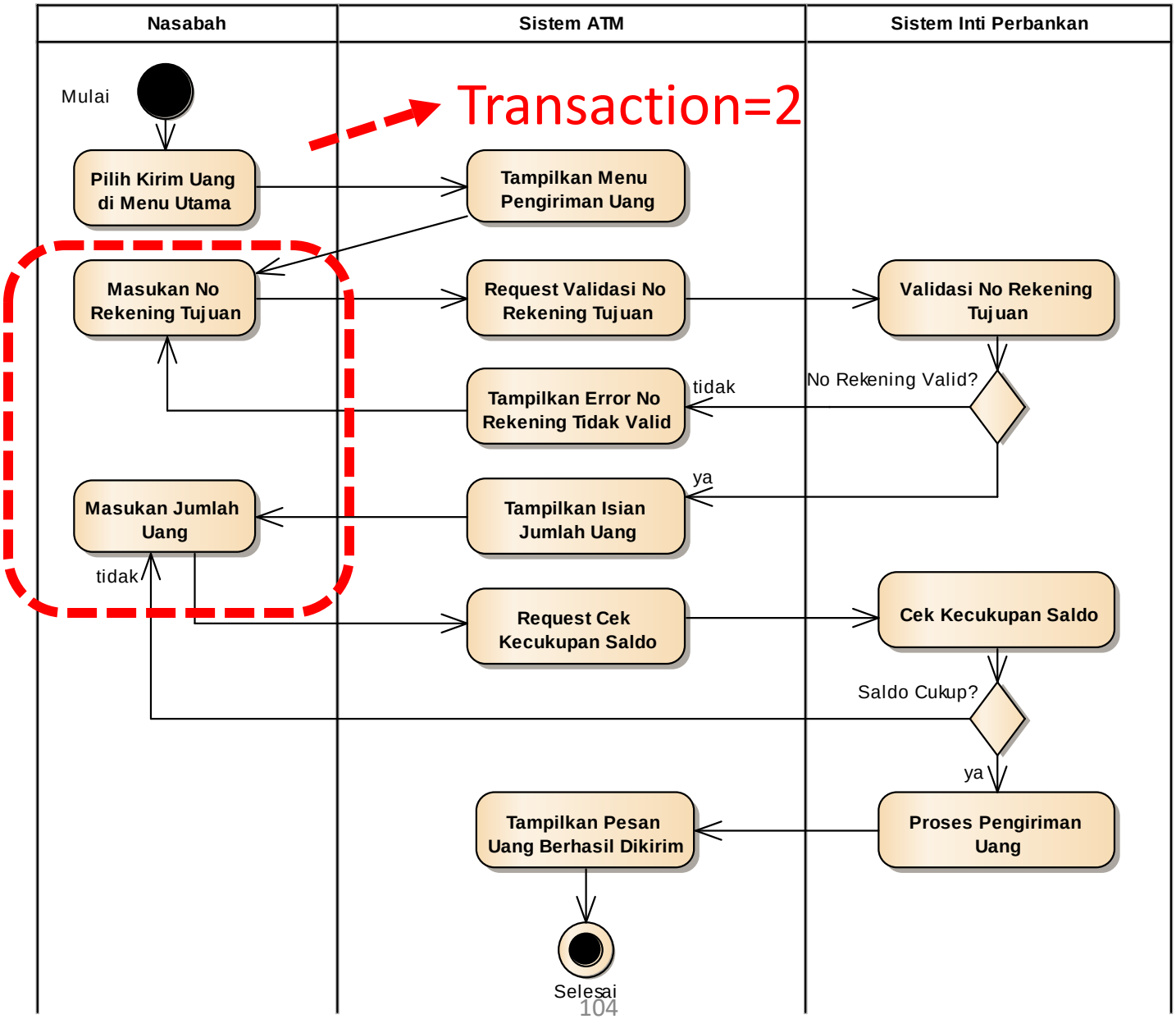


Unadjusted Use Case Points (UUCP) = UAW + UUCW

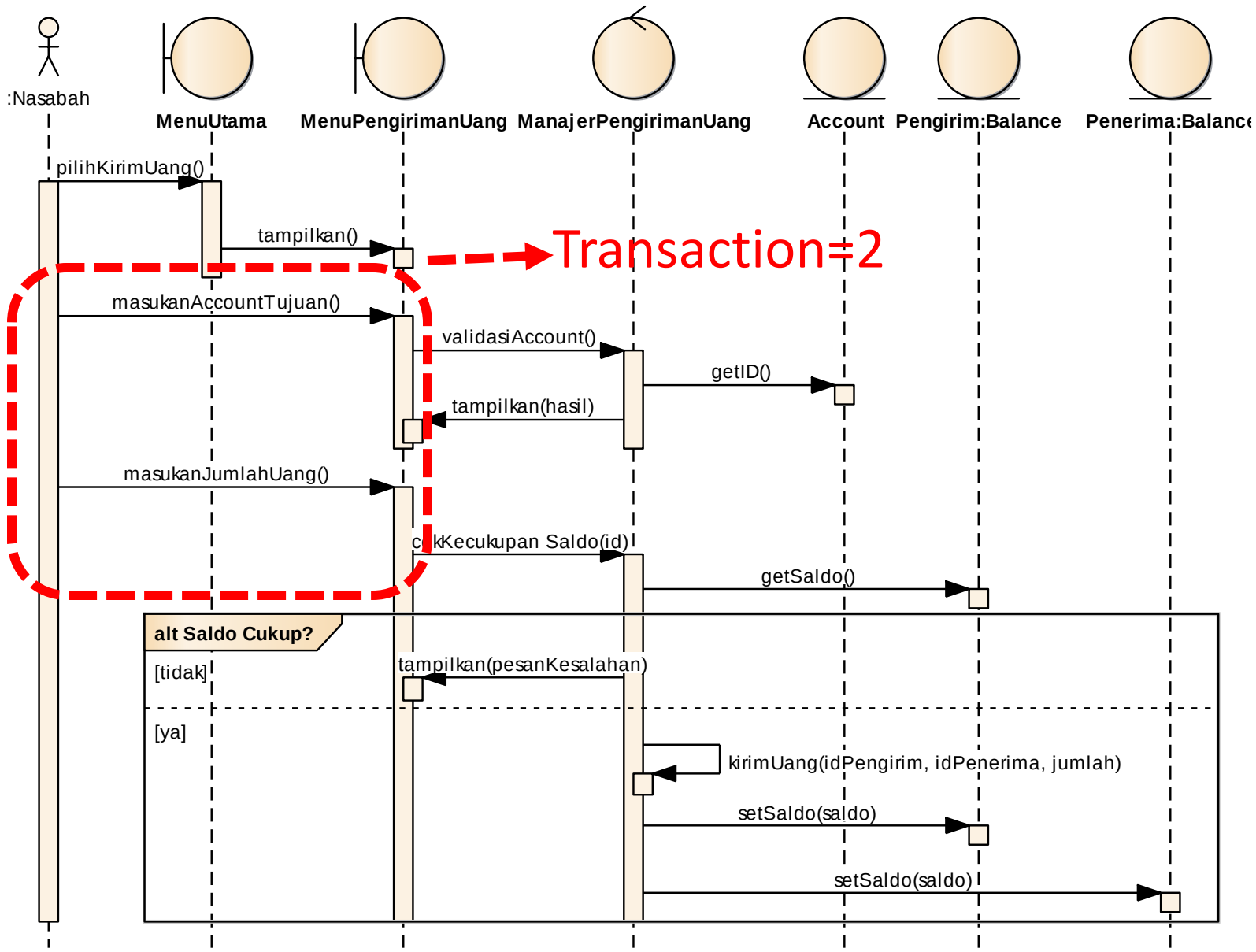
Sistem ATM – Use Case Diagram



Activity Diagram: Mengirim Uang



Sequence Diagram: Mengirim Uang



Technical Complexity Factors (TCF)

Factor Number	Description	Weight
T1	Distributed system	2.0
T2	Response time or throughput performance objectives	1.0
T3	End-user online efficiency	1.0
T4	Complex internal processing	1.0
T5	Reusability of code	1.0
T6	Easy to install	0.5
T7	Ease of use	0.5
T8	Portability	2.0
T9	Ease of change	1.0

$$TCF = 0.6 + (0.01 * TFactor)$$

Environmental Complexity Factors (ECF)

Factor Number	Description	Weight
E1	Familiarity with system development process in use	1.5
E2	Application experience	0.5
E3	Object-oriented experience	1.0
E4	Lead analyst capability	0.5
E5	Motivation	1.0
E6	Requirements stability	2.0
E7	Part time staff	-1.0
E8	Difficulty of programming language	-1.0

$$ECF = 1.4 + (-0.03 * EFactor)$$

Computing Use Case Points

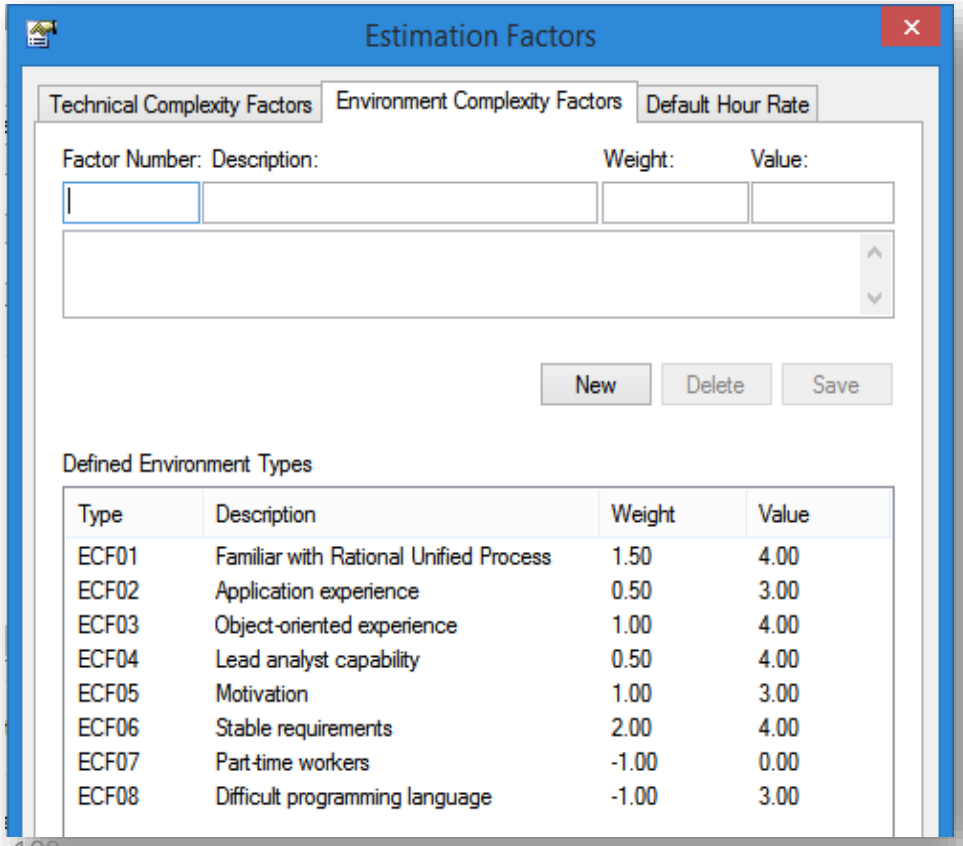
- Adjusted Use Case Points (UCP) = UUCP * TCF * ECF
- Effort in Person Hours = UCP * PHM

Person Hour Multiplier (PHM)

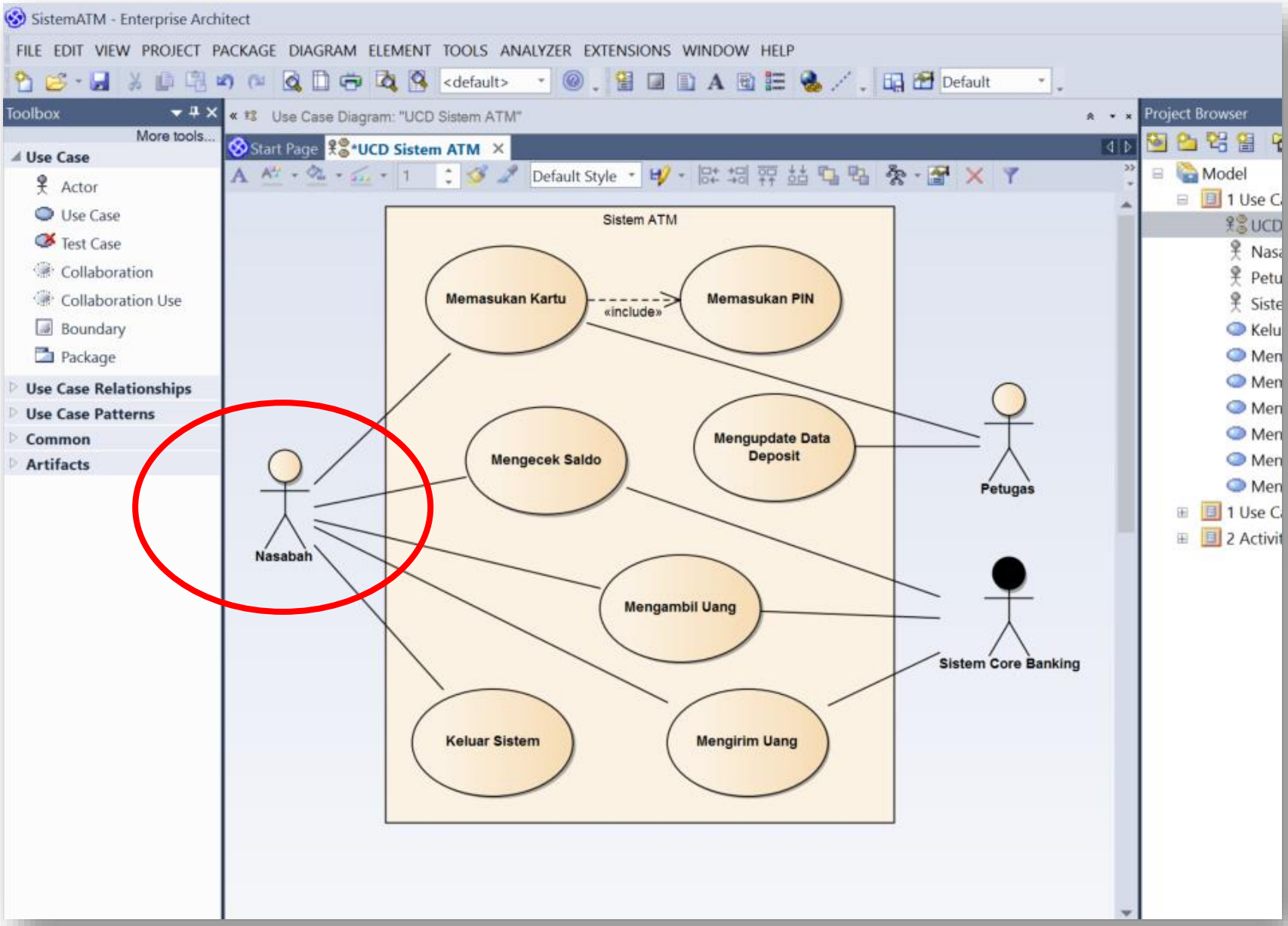
Let F1 = Number of ECF1 to ECF6 that are < 3

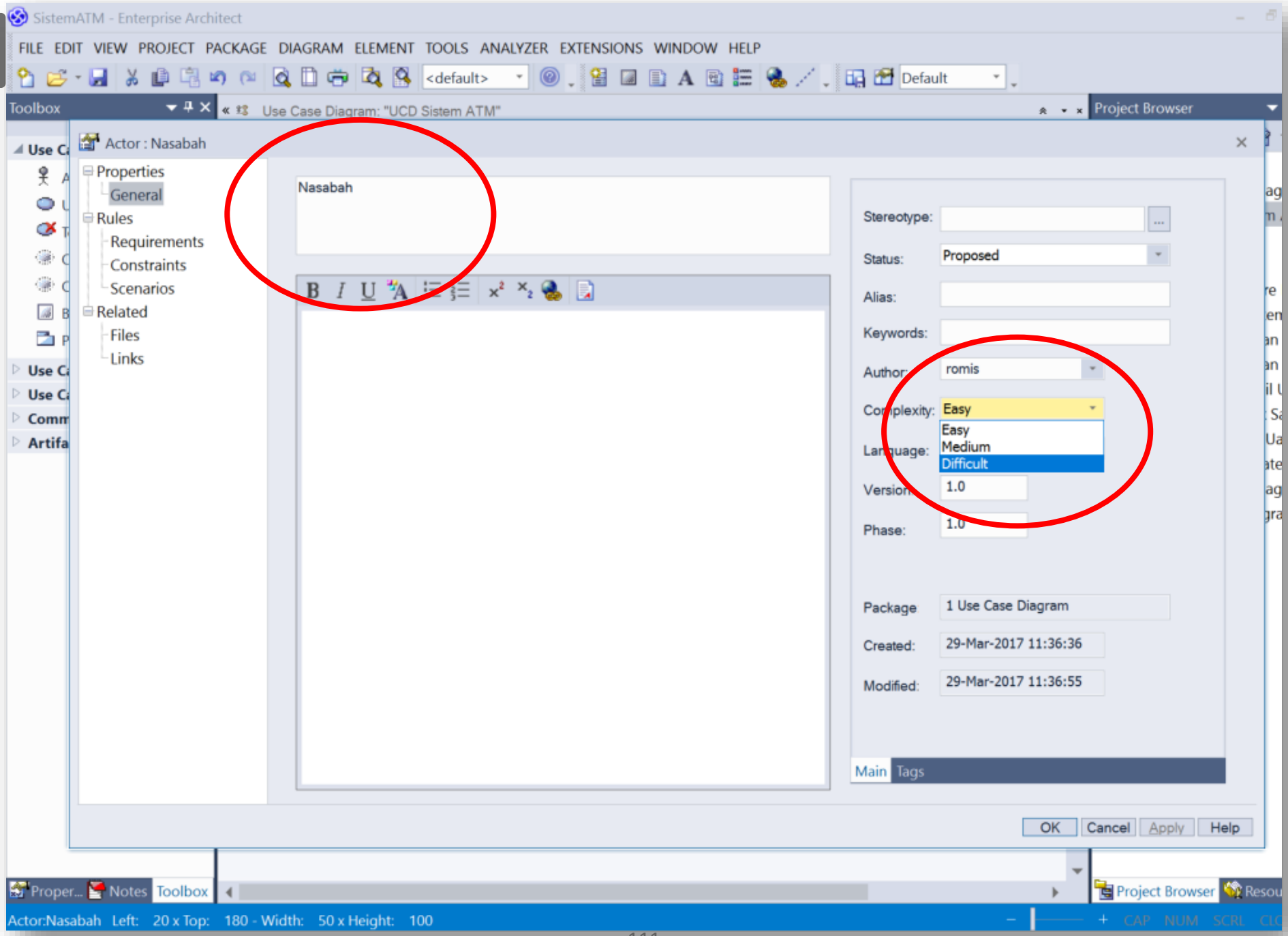
Let F2 = Number of ECF7 and ECF8 that are > 3

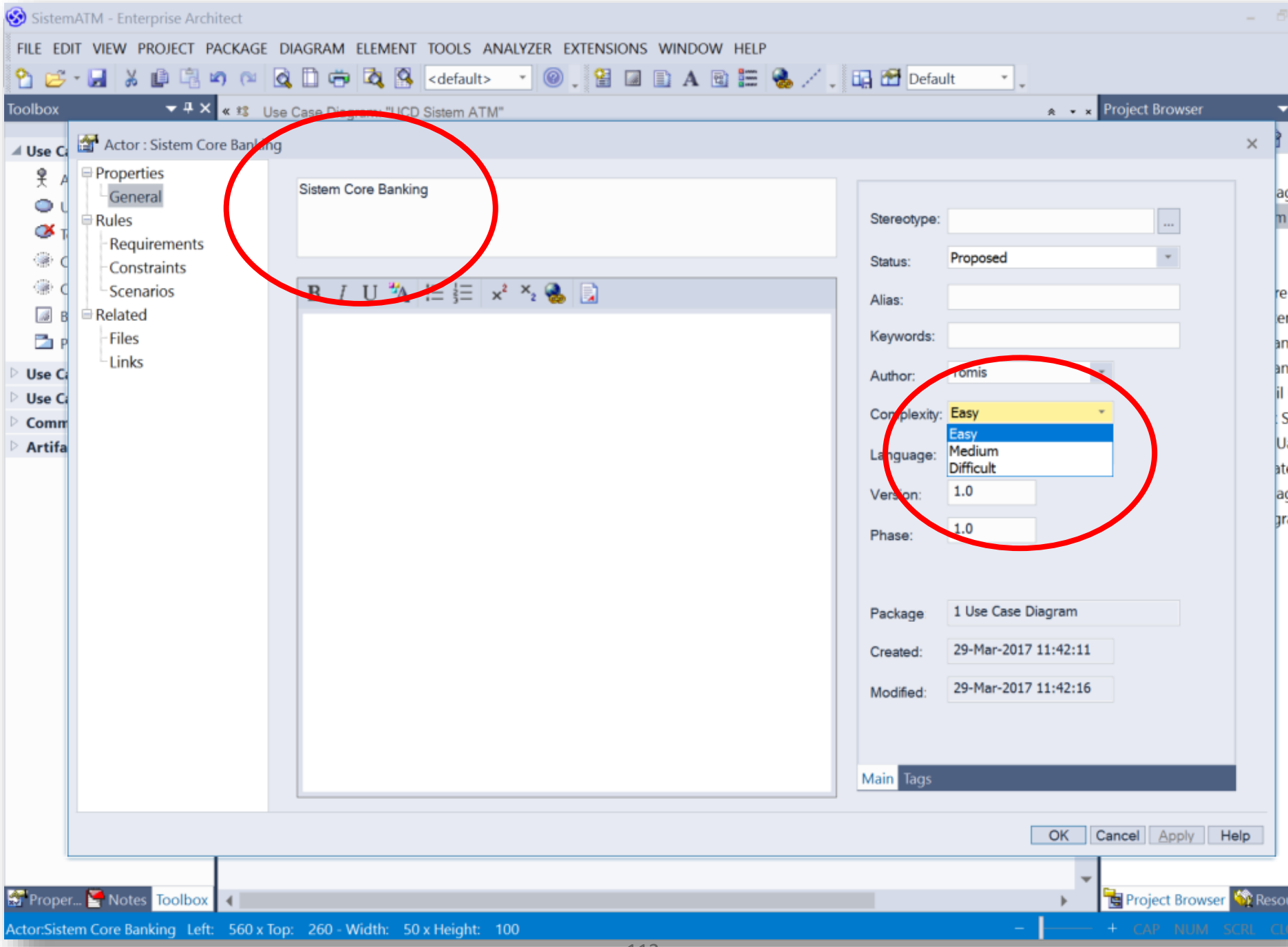
If F1 + F2 <= 2
    PHM = 20
Else if F1 + F2 = 3 or 4
    PHM = 28
Else
    Scrap the project



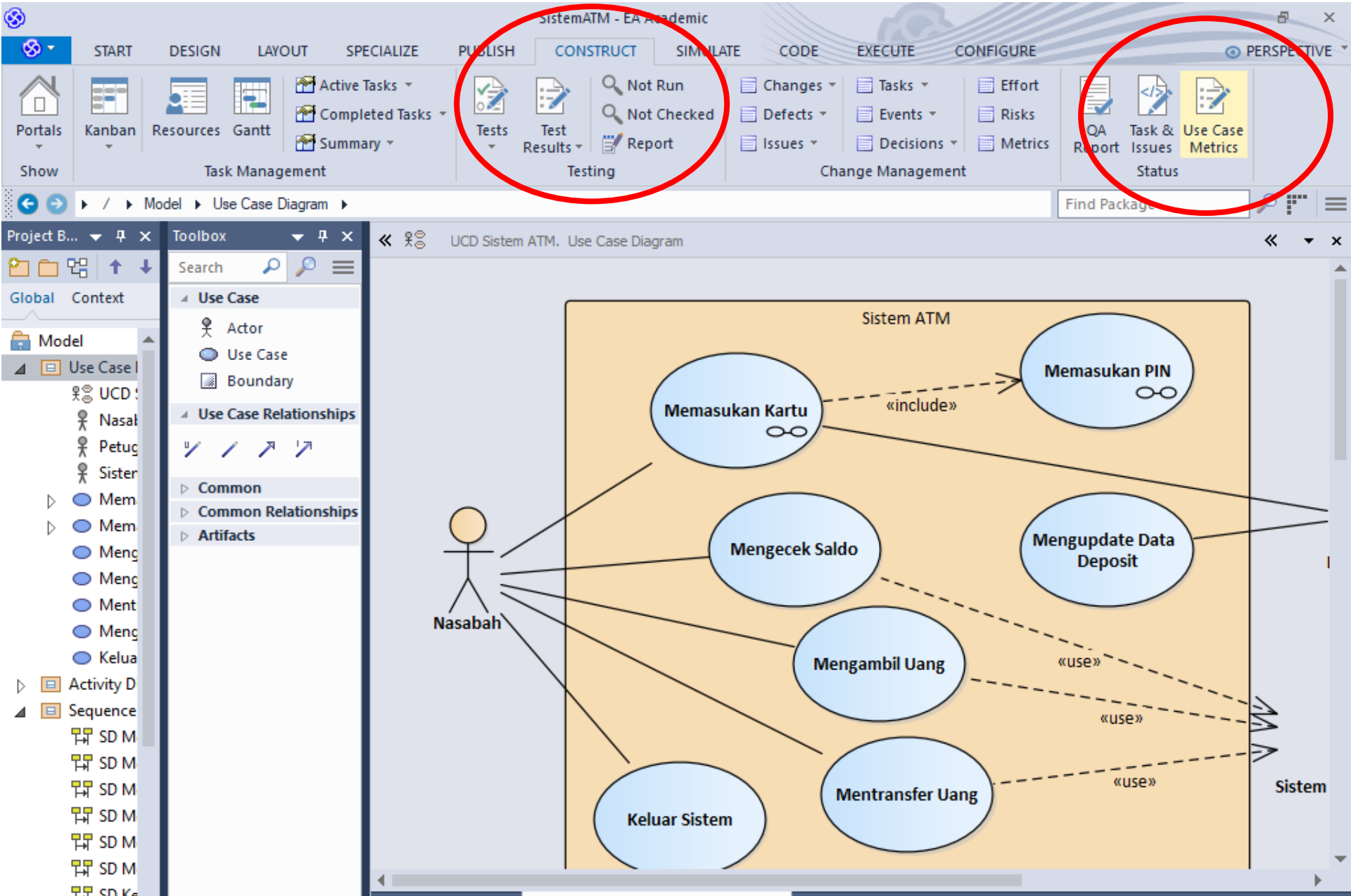
Computing Use Case Points with Sparx EA

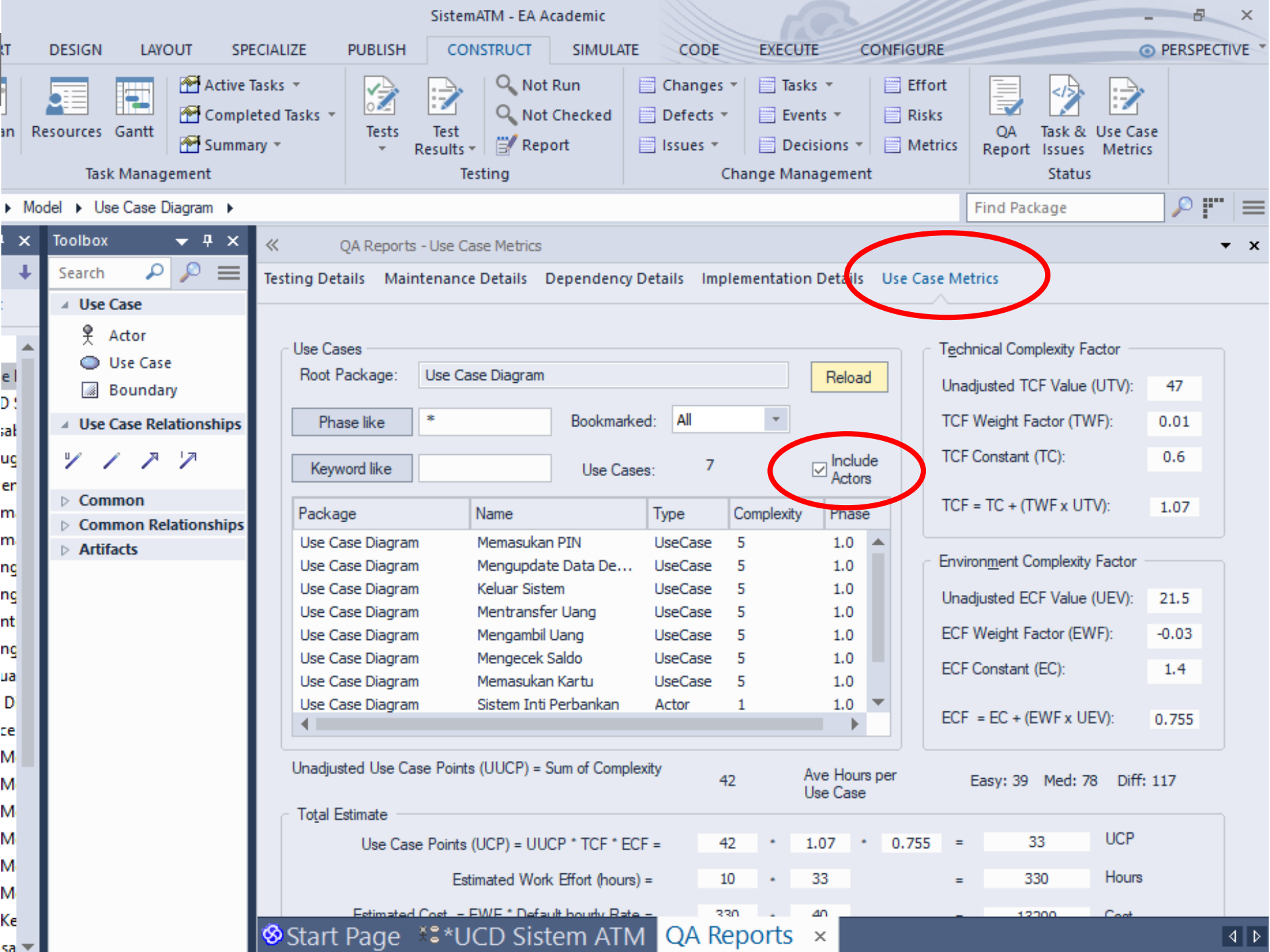


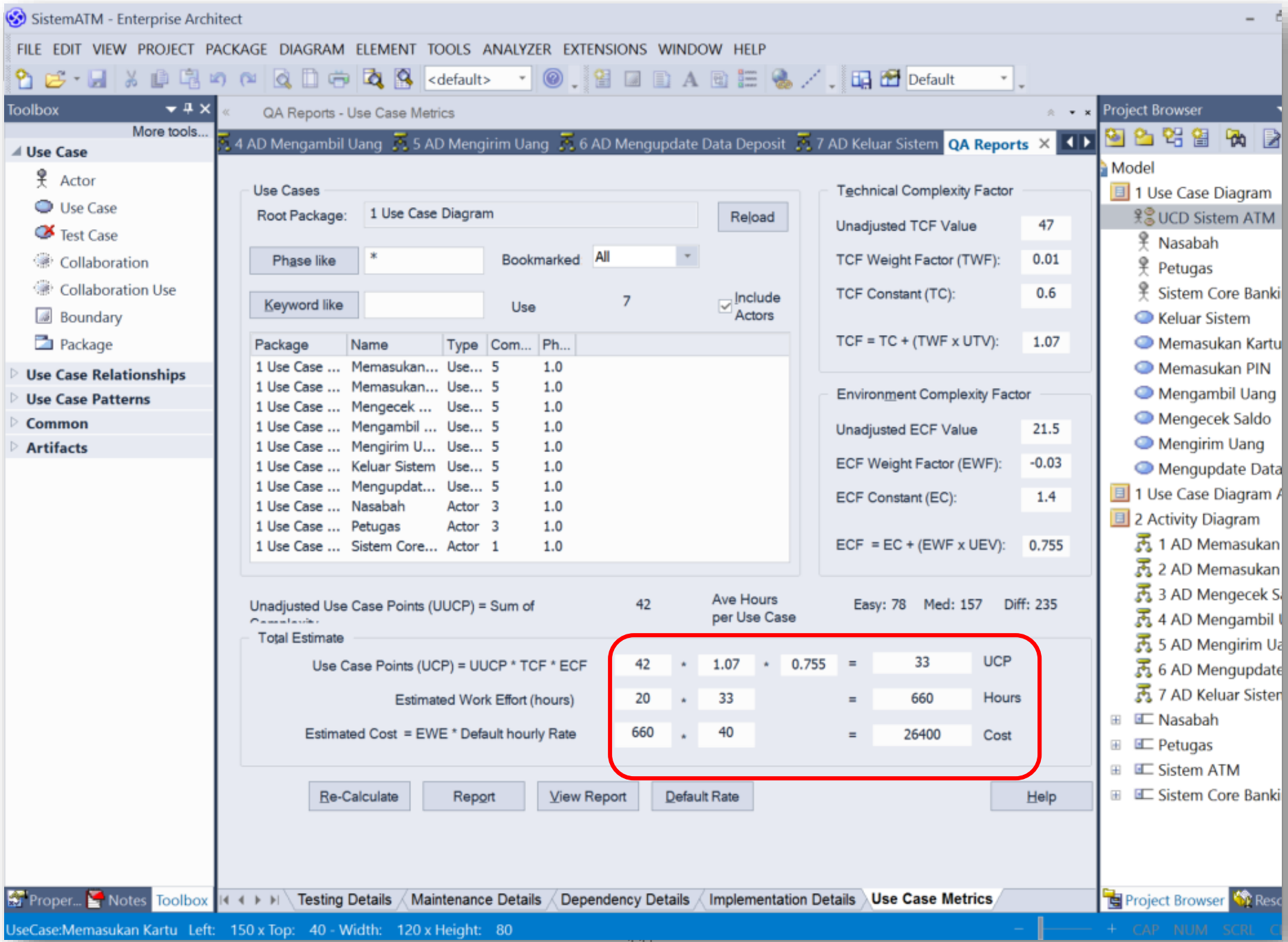




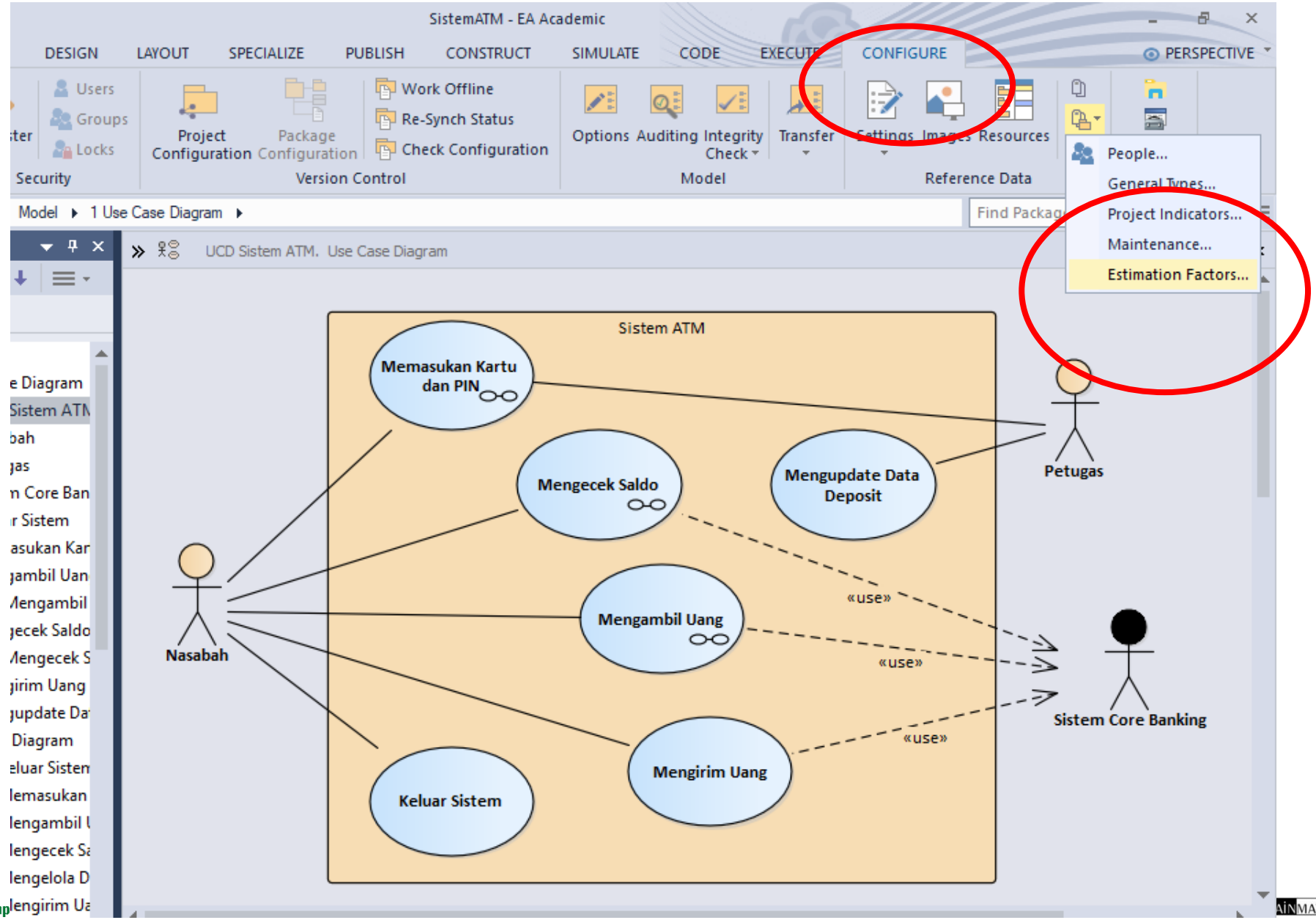
Use Case Points in Sparx EA 14



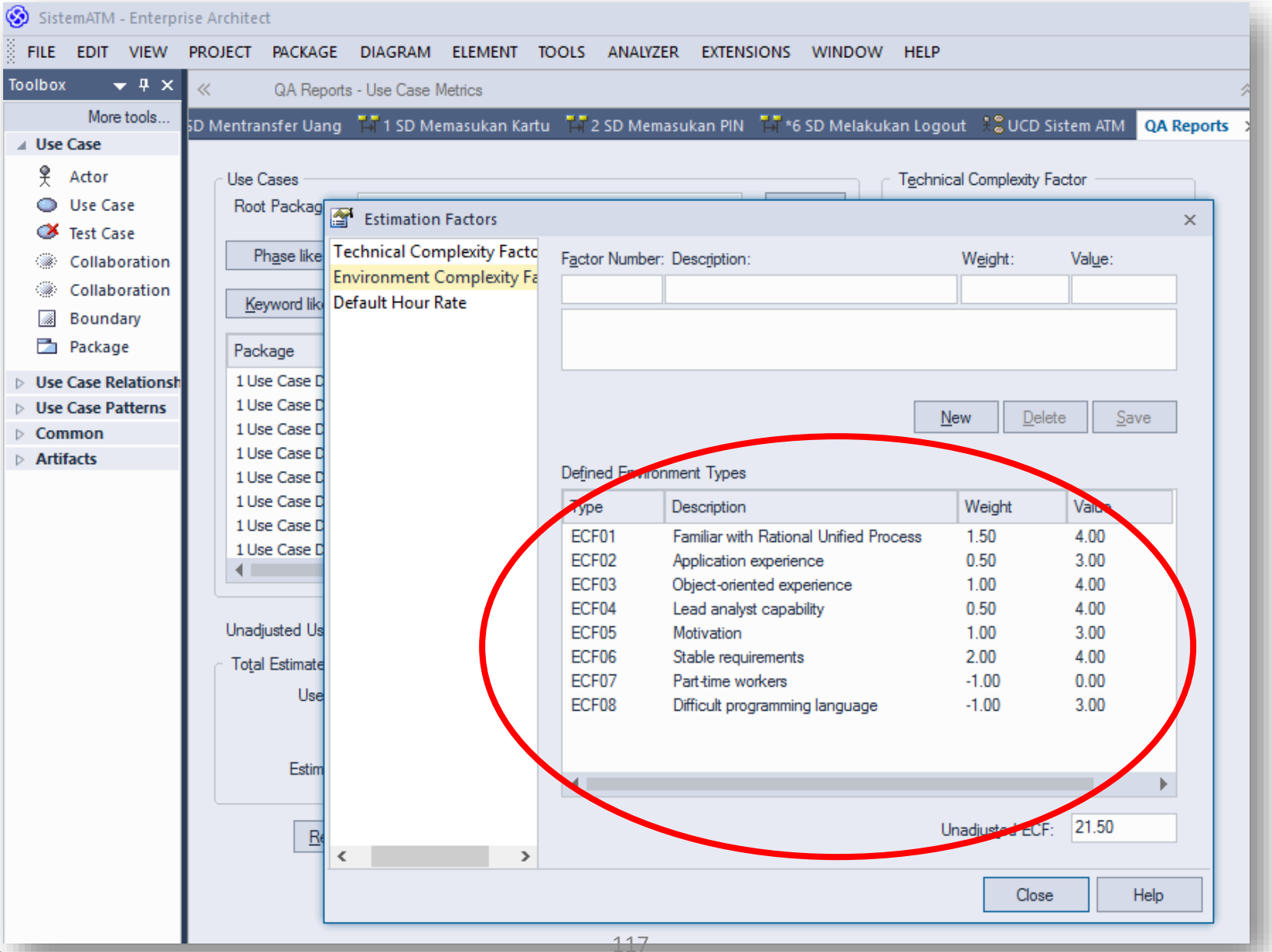

SistemATM - EA Academic
DESIGN
LAYOUT
SPECIALIZE
PUBLISH
CONSTRUCT
SIMULATE
CODE
EXECUTE
CONFIGURE
PERSPECTIVE
Resources
Gantt
Active Tasks
Completed Tasks
Summary
Task Management
Tests
Test Results
Not Run
Not Checked
Report
Testing
Changes
Defects
Issues
Tasks
Events
Decisions
Effort
Risks
Metrics
Change Management
QA Report
Task & Issues
Use Case Metrics
Status
Model
Use Case Diagram
Find Package
Toolbox
Search
Use Case
Actor
Use Case
Boundary
Use Case Relationships
Common
Common Relationships
Artifacts
QA Reports - Use Case Metrics
Testing Details
Maintenance Details
Dependency Details
Implementation Details
Use Case Metrics
Use Cases
Root Package:
Use Case Diagram
Reload
Phase like
*
Bookmarked:
All
Keyword like
Use Cases:
7
Include Actors
Package
Name
Type
Complexity
Phase
Use Case Diagram
Memasukan PIN
UseCase
5
1.0
Use Case Diagram
Mengupdate Data De...
UseCase
5
1.0
Use Case Diagram
Keluar Sistem
UseCase
5
1.0
Use Case Diagram
Mentransfer Uang
UseCase
5
1.0
Use Case Diagram
Mengambil Uang
UseCase
5
1.0
Use Case Diagram
Mengecek Saldo
UseCase
5
1.0
Use Case Diagram
Memasukan Kartu
UseCase
5
1.0
Use Case Diagram
Sistem Inti Perbankan
Actor
1
1.0
Technical Complexity Factor
Unadjusted TCF Value (UTV):
47
TCF Weight Factor (TWF):
0.01
TCF Constant (TC):
0.6
TCF = TC + (TWF x UTV):
1.07
Environment Complexity Factor
Unadjusted ECF Value (UEV):
21.5
ECF Weight Factor (EWF):
-0.03
ECF Constant (EC):
1.4
ECF = EC + (EWF x UEV):
0.755
Unadjusted Use Case Points (UUCP) = Sum of Complexity
42
Ave Hours per Use Case
Easy: 39
Med: 78
Diff: 117
Total Estimate
Use Case Points (UCP) = UUCP * TCF * ECF =
42
1.07
0.755
33
UCP
Estimated Work Effort (hours) =
10
33
330
Hours
Start Page
*UCD Sistem ATM
QA Reports



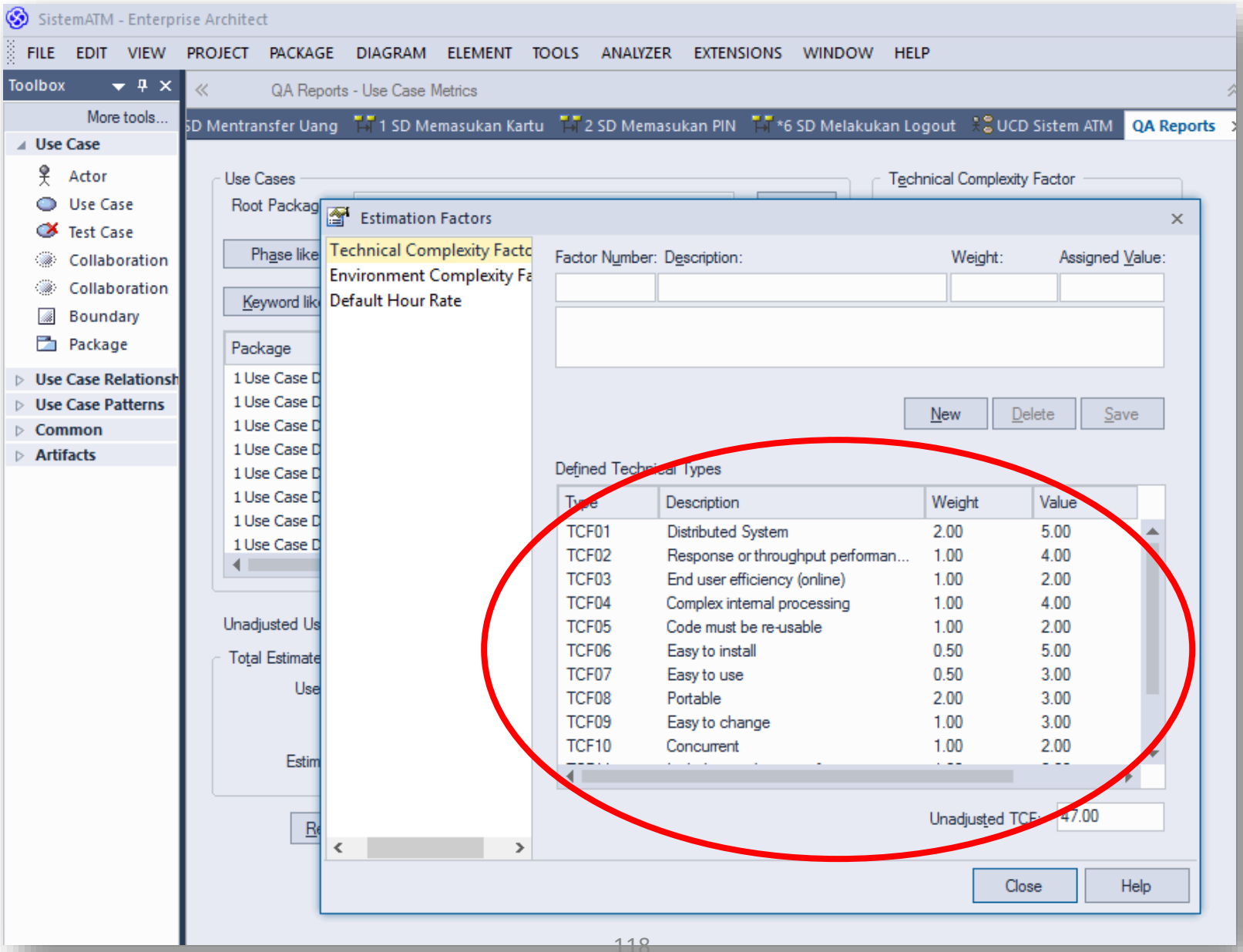
Estimation Factors in Sparx EA 14



Environment Complexity Factors (ECF)



Technical Complexity Factors (TCF)



Example: Sistem ATM

UCP	= 33	
PHM	= 20	
PH	= 20*33	= 660 PH
	28*33	= 924
PM	= 660/8/22	= 3.75 PM
	= 924/8/22	= 5.25 PM
TIME (M)	= 3.0 * PM $^{1/3}$	
TIME (M)	= 3.0 * 3.75 $^{1/3}$	= 4.6 M
	= 3.0 * 5.25 $^{1/3}$	= 5.21 M

Example: Sistem ERP

UCP = 1517

PHM = 20

PH = 20*1517 = 30340 PH

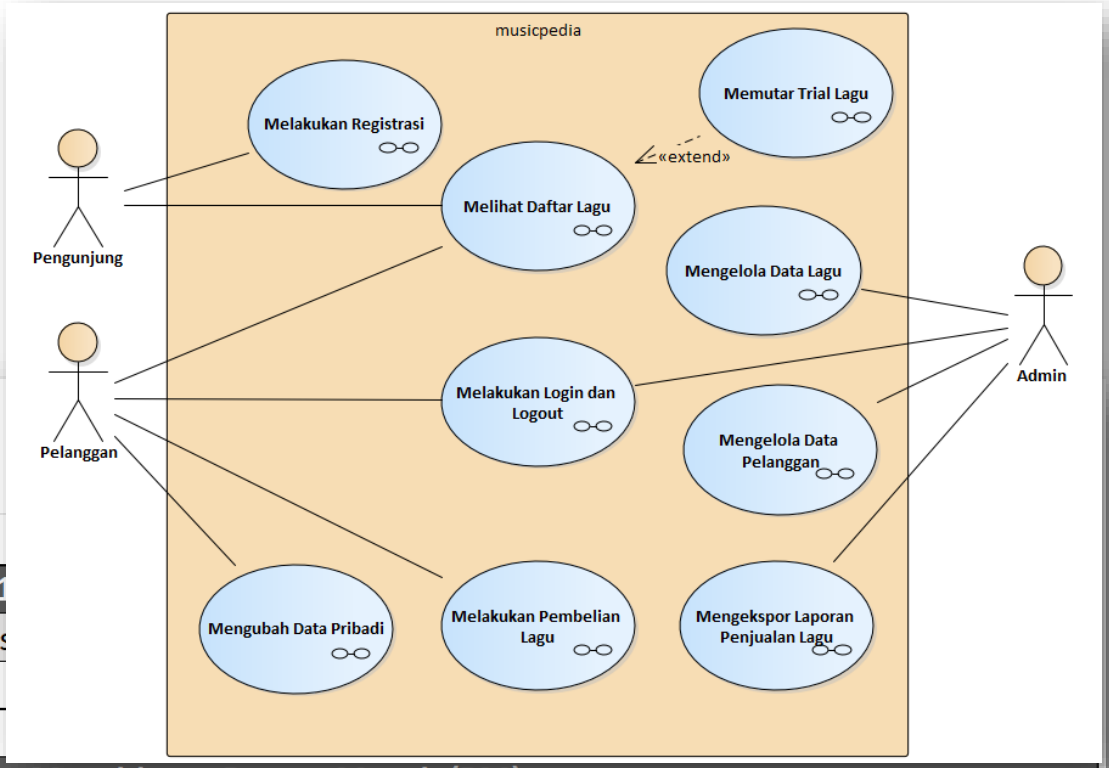
28*1517

PM = 30340/8/22 = 172 PM

TIME (M) = $3.0 * PM^{1/3}$

TIME (M) = $3.0 * 172^{1/3}$ = 14 M

Example: MusicPedia



Technical Feasibility

Use Case Points

Tahap 1	
Use Case Points (UCP)	Person Hours
51	
51	

Tahap 2 - Menghitung Person Month (PM)

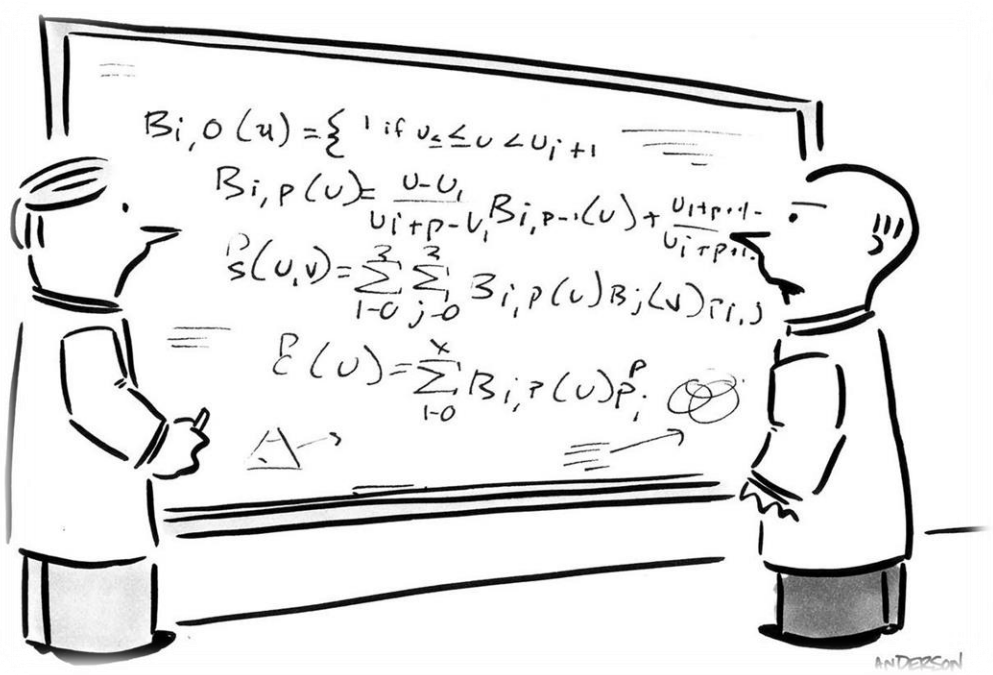
PHM	Person Hours (PH)	Lama Bekerja Perhari	Jumlah Bekerja Sebulan	Person Months (PM)
20	1020	8	22	5.80
	1020	10	26	3.92
28	1428	8	22	8.11
	1428	10	26	5.49

Tahap 3 - Menghitung Time (Month)

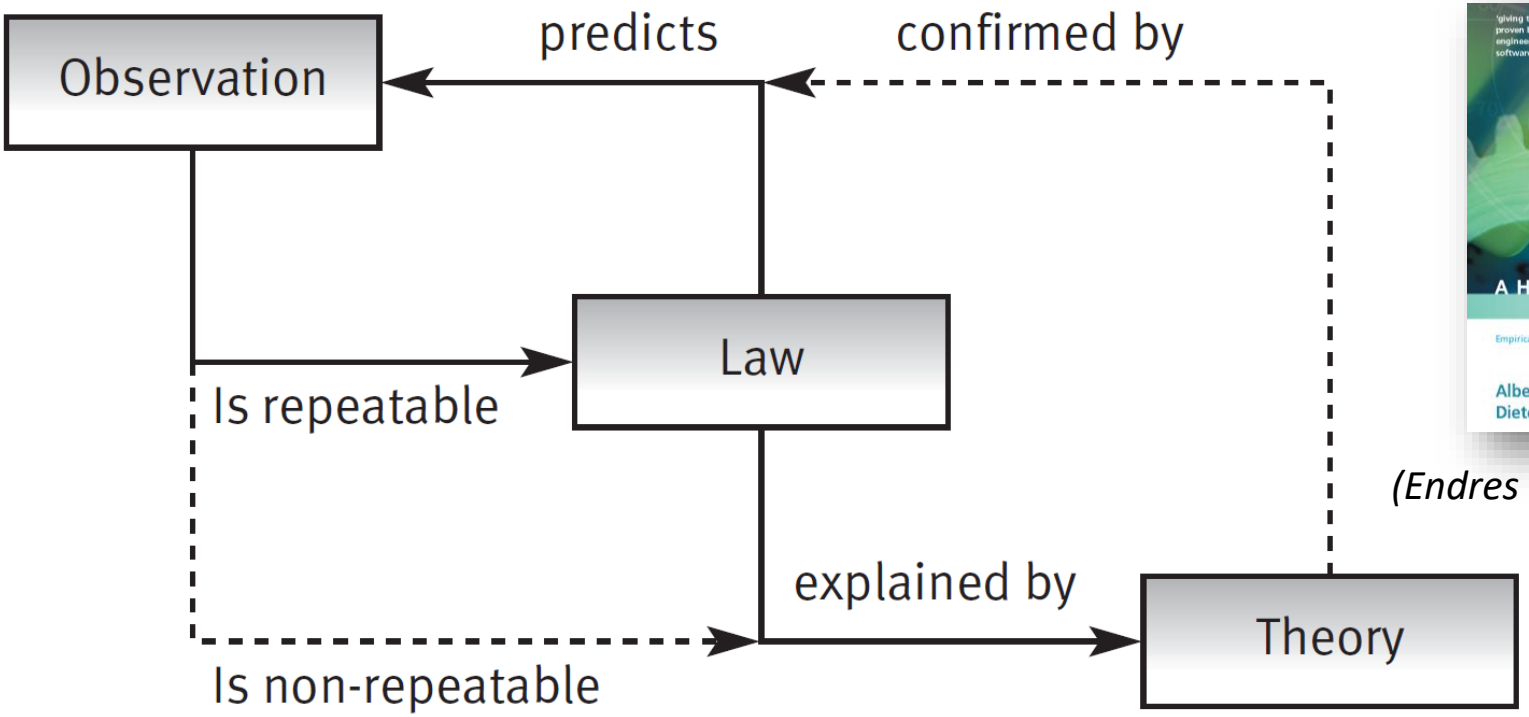
PHM	Formula Penghitung Waktu	Jumlah Bekerja Sebulan	Waktu dalam Bulan (M)
20	3 * PM^(1/3)	22	5.39
		26	4.73
28		22	6.03
		26	5.29

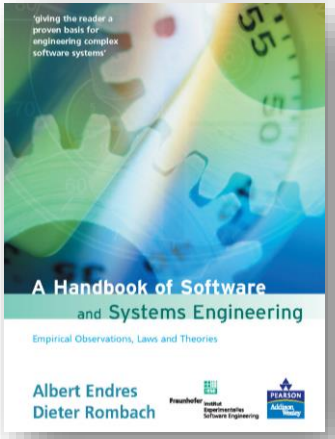
MITOS 7

Software Engineering Itu Langsung Praktek Saja, Jangan Kebanyakan Teori



Observations, Laws and Theories





(Endres & Rombach, 2003)

- Repeatable observations can often be stated as a *law*
- Using this law, we can then make predictions about further observations
- A law tells us how things occur, but not why. *Theories* explain why it is true and order our observations

Software Engineering Laws

ANALYSIS AND DESIGN

1. Requirement deficiencies are the prime source of project failures (Glass)
2. Errors are most frequent during the requirements and design activities and are the more expensive the later they are removed (Boehm 1)
3. Prototyping significantly reduces requirement and design errors, especially for user interfaces (Boehm 2)
4. The value of models depends on the view taken, but none is best for all purposes (Davis)
5. Good designs require deep application domain knowledge (Curtis)
6. Hierarchical structures reduce complexity (Simon)
7. A structure is stable if cohesion is strong and coupling low (Constantine)
8. Only what is hidden can be changed without risk (Parnas)
9. Separation of concerns leads to standard architectures (Denert)
10. Screen pointing-time is a function of distance and width (Fitts-Shneiderman)

Software Engineering Laws

CONSTRUCTION

11. What applies to small systems does not apply to large ones (DeRemer)
12. Productivity and reliability depend on the length of a program's text, independent of language level used (Corbató)
13. Well-structured programs have fewer errors and are easier to maintain (Dijkstra-Mills-Wirth)
14. The larger and more decentralized an organization, the more likely it is that it has reuse potential (Lanergan)
15. Software reuse reduces cycle time and increases productivity and quality (McIlroy)
16. A system reflects the organizational structure that built (Conway)

Software Engineering Laws

TESTING

17. Inspections significantly increase productivity, quality, and project stability (Fagan)
18. Effectiveness of inspections is fairly independent of its organizational form (Porter–Votta)
19. Perspective-based inspections are (highly) effective and efficient (Basili)
20. A combination of different V&V methods outperforms any single method alone (Hetzel–Myers)
21. Online debugging is more efficient than offline debugging (Sackman 1)
22. Testing can show the presence but not the absence of errors (Dijkstra)
23. A developer is unsuited to test his or her code (Weinberg)
24. Approximately 80 percent of defects come from 20 percent of modules (Pareto–Zipf)
25. Performance testing benefits from system-level benchmarks (Gray–Serlin)
26. Usability is quantifiable (Nielsen-Norman)

Software Engineering Laws

EVOLUTION

27. A system that is used will be changed (Lehman 1)
28. An evolving system increases its complexity, unless work is done to reduce it (Lehman 2)
29. System evolution is determined by a feedback process (Lehman 3)
30. Smaller changes have a higher error density than large ones (Basili–Möller)

Software Engineering Laws

PLANNING

31. Individual developer performance varies considerably (Sackman 2)
32. A multitude of factors influence developer productivity (Nelson–Jones)
33. Development effort is a (non-linear) function of product size (Boehm 3)
34. Most cost estimates tend to be too low (DeMarco–Glass)
35. Mature processes and personal discipline enhance planning, increase productivity, and reduce errors (Humphrey)
36. Adding manpower to a late project makes it later (Brooks)
37. Products replace services through productivity gains (Baumol)

Software Engineering Laws

HUMAN RESOURCE

38. Humans receive most information through the visual system and store it in a spatially organized memory (Kupfmüller)
39. Humans tend to structure what they see to form cohesive patterns (Gestalt)
40. Short-term memory is limited to 7 ± 2 chunks of information (Miller)
41. Multimodal information is easier to remember than single mode (Krause)
42. The more knowledge that is available, the more effort has to be spent on the processes to use it (Librarian)
43. It takes 5000 hours to turn a novice into an expert (Apprentice)
44. Human needs and desires are strictly prioritized (Maslow-Herzberg)

Software Engineering Laws

TECHNOLOGY

45. The price and performance of processors is halved every 18 months (Moore)
46. The capacity of magnetic devices increases by a factor of ten every decade (Hoagland)
47. Wireless bandwidth doubles every 2.5 years (Cooper)
48. Architecture wins over technology (Morris-Ferguson)
49. The value of a network increases with the square of its users (Metcalfe)
50. The probability that a hypothesis is true increases the more unlikely the new event is that confirms this hypothesis (Bayes)

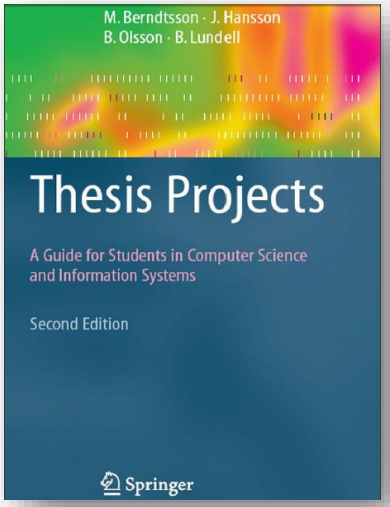
MITOS 8

Penelitian Software Engineering itu Hasilnya adalah Produk Software

Mengapa Melakukan Penelitian?

- Berangkat dari adanya masalah penelitian
  - yang mungkin sudah diketahui metode pemecahannya
  - tapi belum diketahui metode pemecahan yang lebih baik

- Research (Inggris) dan recherche (Prancis)
  - re (kembali)
  - to search (mencari)



- The process of exploring the unknown, studying and learning new things, building new knowledge about things that no one has understood before
  (Berndtsson et al., 2008)

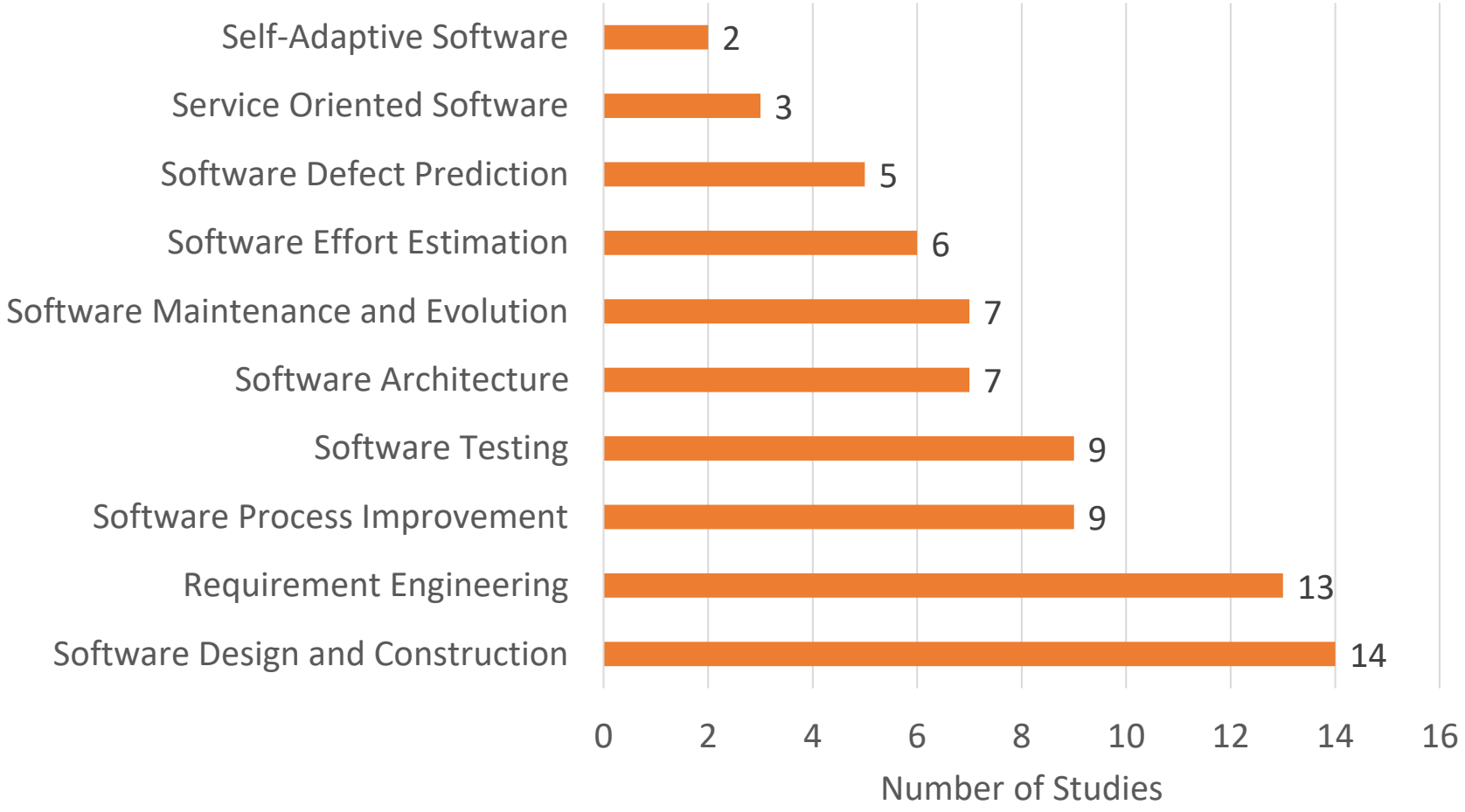
Pengembangan Software vs Penelitian

- Membangun software bukanlah tujuan utama penelitian, hanya *testbed* untuk mempermudah kita dalam mengukur hasil penelitian
  - Tidak ada listing code, UML atau screenshot software di paper-paper journal (SCOPUS/WoS), kecuali penelitian tentang perbaikan paradigma pemrograman, analisis design, dsb

- Ketika pada penelitian kita mengusulkan perbaikan suatu algoritma (*proposed method*)
  - Bidang image processing, topik penelitian face recognition, memikirkan perbaikan metode/algoritma untuk pengenalan wajah dengan akurat/efisien
  - Bidang data mining, topik decision tree, memikirkan perbaikan algoritma decision tree sehingga bisa memprediksi (klasifikasi) dengan lebih akurat
  - Untuk mempermudah eksperimen dan evaluasi, kita menulis kode program (software) untuk menguji dan mengevaluasi performance dari algoritma yang kita usulkan

Penelitian Bidang Software Engineering?

Penelitian bidang software engineering bukan penelitian tentang pengembangan software yang hasil akhirnya produk software, tapi penelitian untuk perbaikan metodologi pengembangan software



Resources: Survey Papers from ScienceDirect, SpringerLink, and IEEE Explore (2011-2014)

MITOS 9

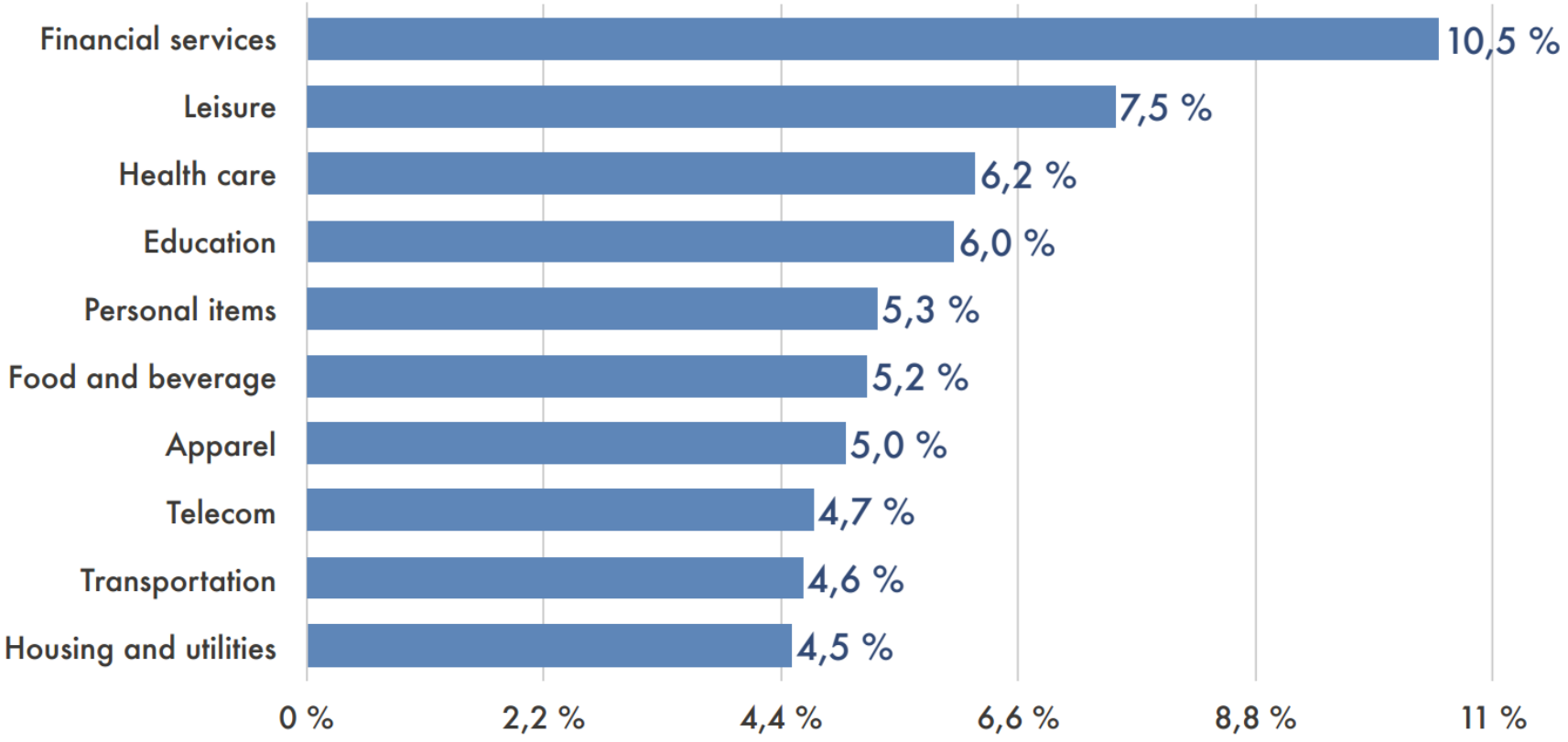
Saya Akan Membuat Aplikasi Seperti Yang Ada Sekarang

Kejar Ceruk Pasar Baru

- Don't Reinvent The Wheel!
- Jangan pernah membuat aplikasi yang sama saja dengan yang ada saat ini

- Lakukan komparasi terhadap aplikasi sejenis, lihat dimana ada gap dan ceruk pasar yang belum tergarap
- Buat aplikasi untuk segmen pasar baru dan diprediksi akan tumbuh besar

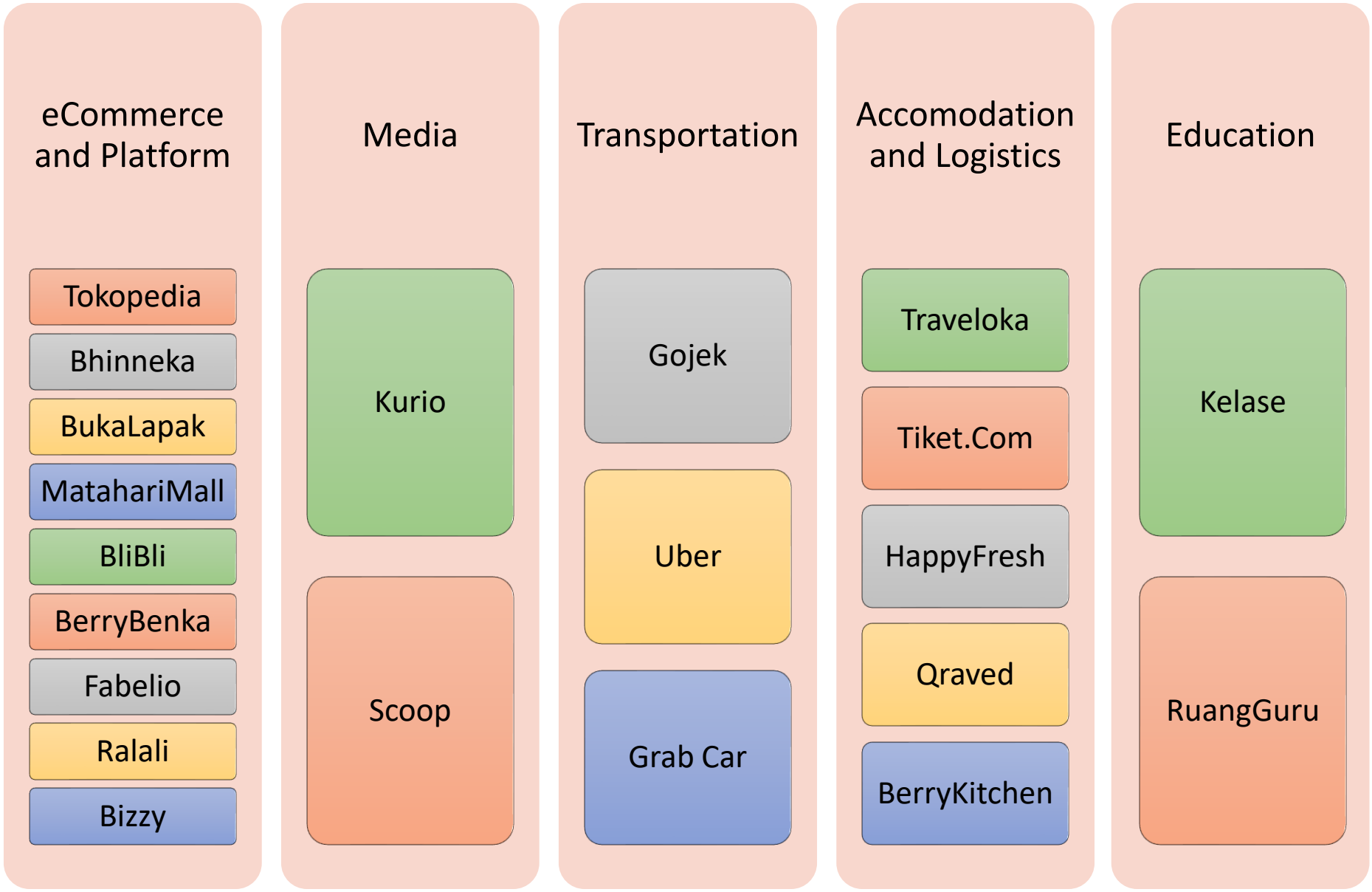
Consumer Spending



RAPID URBANIZATION, RISING INCOME LEVELS, FAVORABLE DEMOGRAPHIC PATTERNS AND CHANGING LIFESTYLE TRENDS ARE JUST SOME OF THE FACTORS THAT ARE BOOSTING CONSUMER SPENDING IN INDONESIA

Source: McKinsey 2013

Indonesia Software Innovation Maps



Evolution of Sciences

- Sebelum 1600: Empirical science
  - Disebut sains kalau bentuknya kasat mata
- 1600-1950: Theoretical science
  - Disebut sains kalau bisa dibuktikan secara matematis atau eksperimen
- 1950s-1990: Computational science
  - Seluruh disiplin ilmu bergerak ke komputasi
  - Lahirnya banyak model komputasi
- 1990-sekarang: Data science
  - Kultur manusia menghasilkan data besar
  - Kemampuan komputer untuk mengolah data besar
  - Datangnya data mining sebagai arus utama sains

Jim Gray and Alex Szalay, The World Wide Telescope: An Archetype for Online Science, Comm. ACM, 45(11): 50-54, Nov. 2002

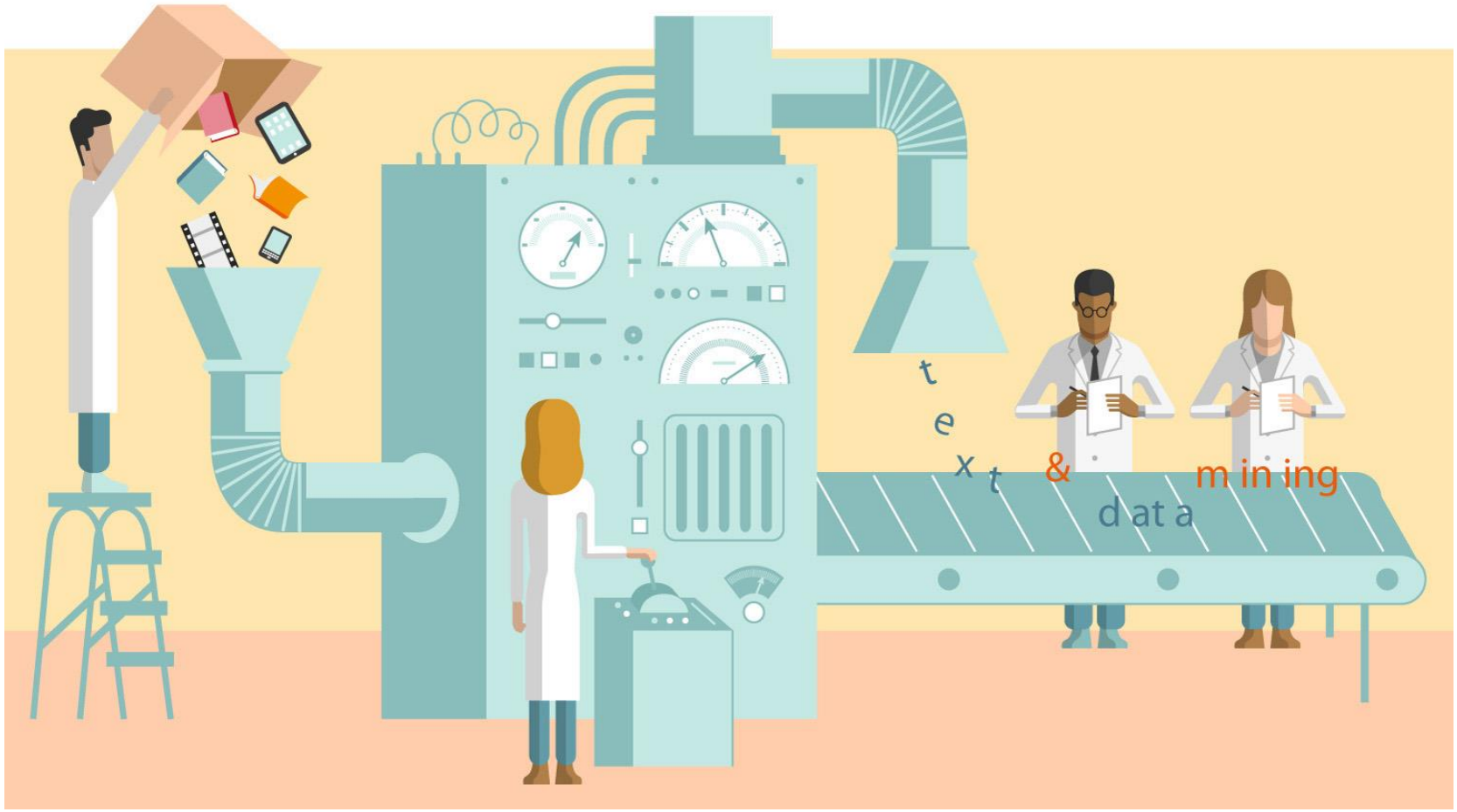
Datangnya Tsunami Data

kilobyte (**kB**)	10^3
megabyte (**MB**)	10^6
gigabyte (**GB**)	10^9
terabyte (**TB**)	10^{12}
petabyte (**PB**)	10^{15}
exabyte (**EB**)	10^{18}
zettabyte (**ZB**)	10^{21}
yottabyte (**YB**)	10^{24}

- Mobile Electronics market
  - 7B smartphone subscriptions in 2015

- Web & Social Networks generates amount of data
  - Google processes 100 PB per day, 3 million servers
  - Facebook has 300 PB of user data per day
  - Youtube has 1000PB video storage

Mining dari Data ke Pengetahuan

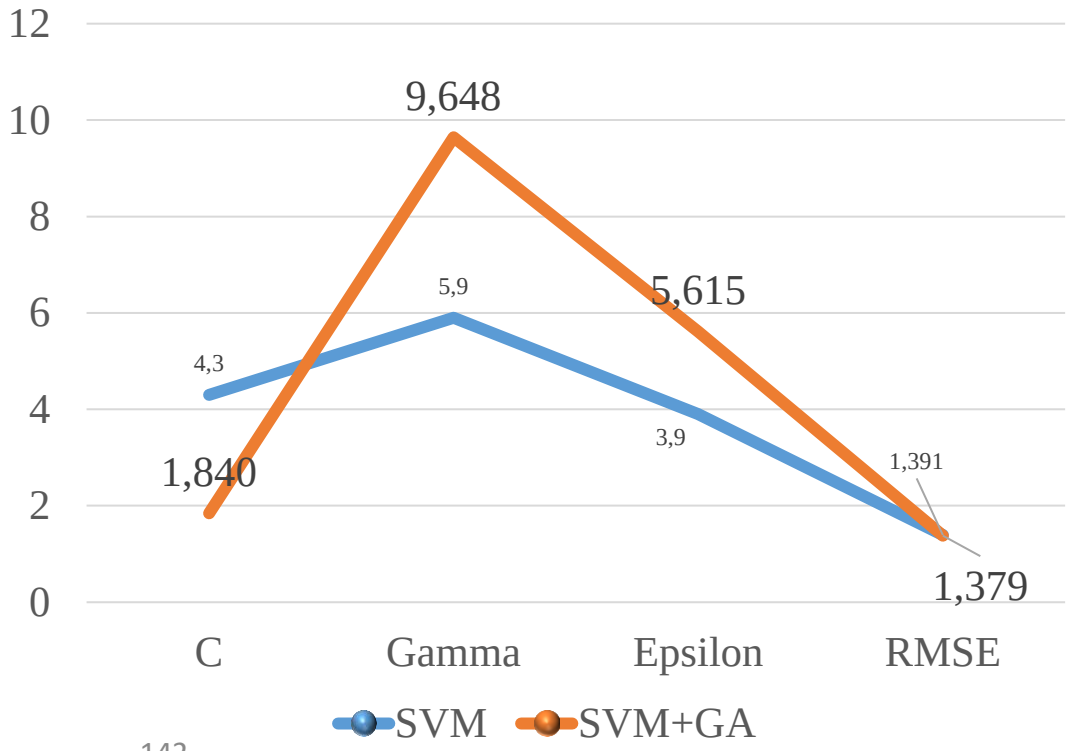


Disiplin ilmu yang mempelajari metode untuk mengekstrak pengetahuan atau menemukan pola dari suatu data yang besar

Sistem Prediksi Kebakaran Hutan

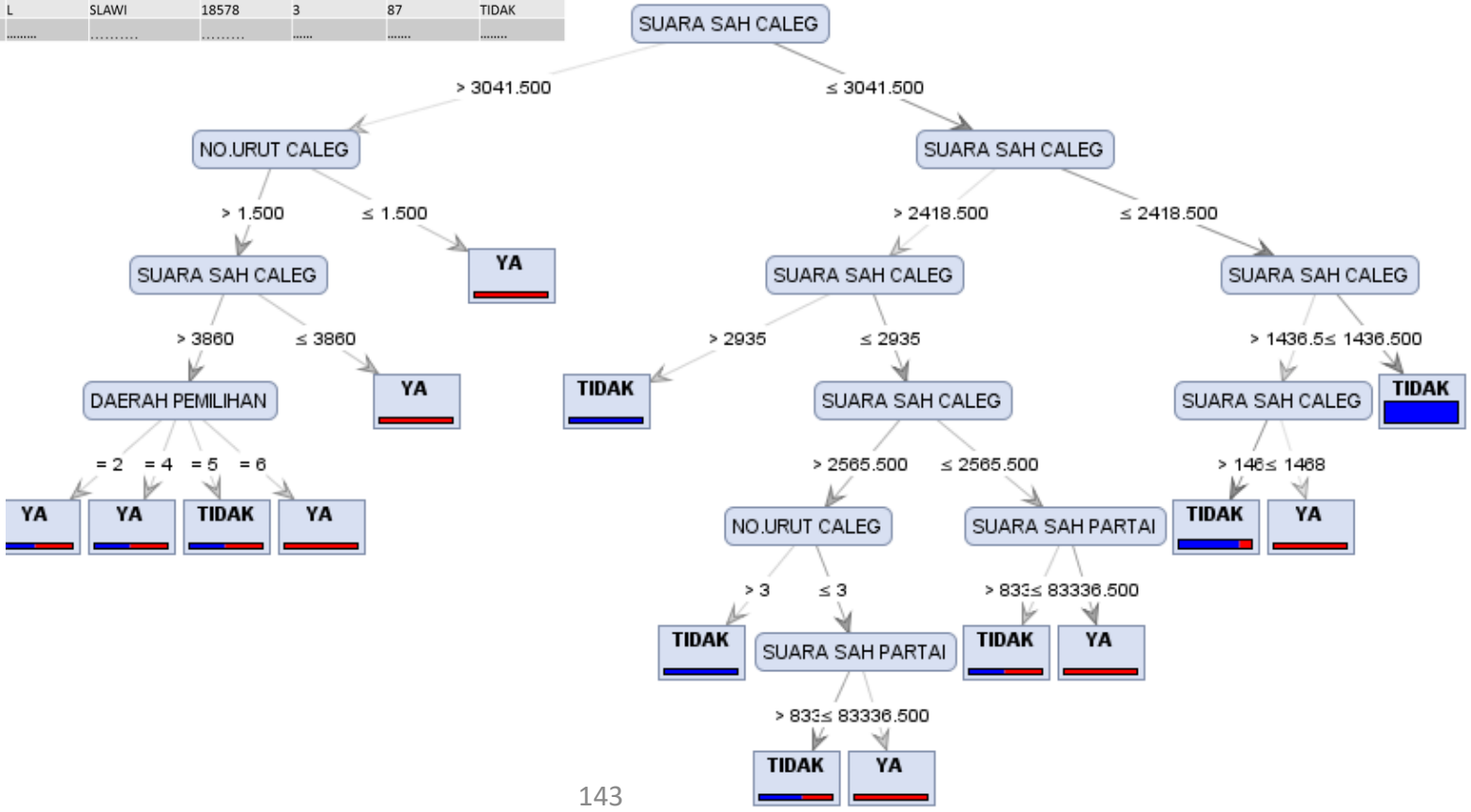
FFMC	DMC	DC	ISI	temp	RH	wind	rain	ln(area+1)
93.5	139.4	594.2	20.3	17.6	52	5.8	0	0
92.4	124.1	680.7	8.5	17.2	58	1.3	0	0
90.9	126.5	686.5	7	15.6	66	3.1	0	0
85.8	48.3	313.4	3.9	18	42	2.7	0	0.307485
91	129.5	692.6	7	21.7	38	2.2	0	0.357674
90.9	126.5	686.5	7	21.9	39	1.8	0	0.385262
95.5	99.9	513.3	13.2	23.3	31	4.5	0	0.438255

	SVM	SVM+GA
C	4.3	1,840
Gamma (γ)	5.9	9,648
Epsilon (ε)	3.9	5,615
RMSE	1.391	1.379



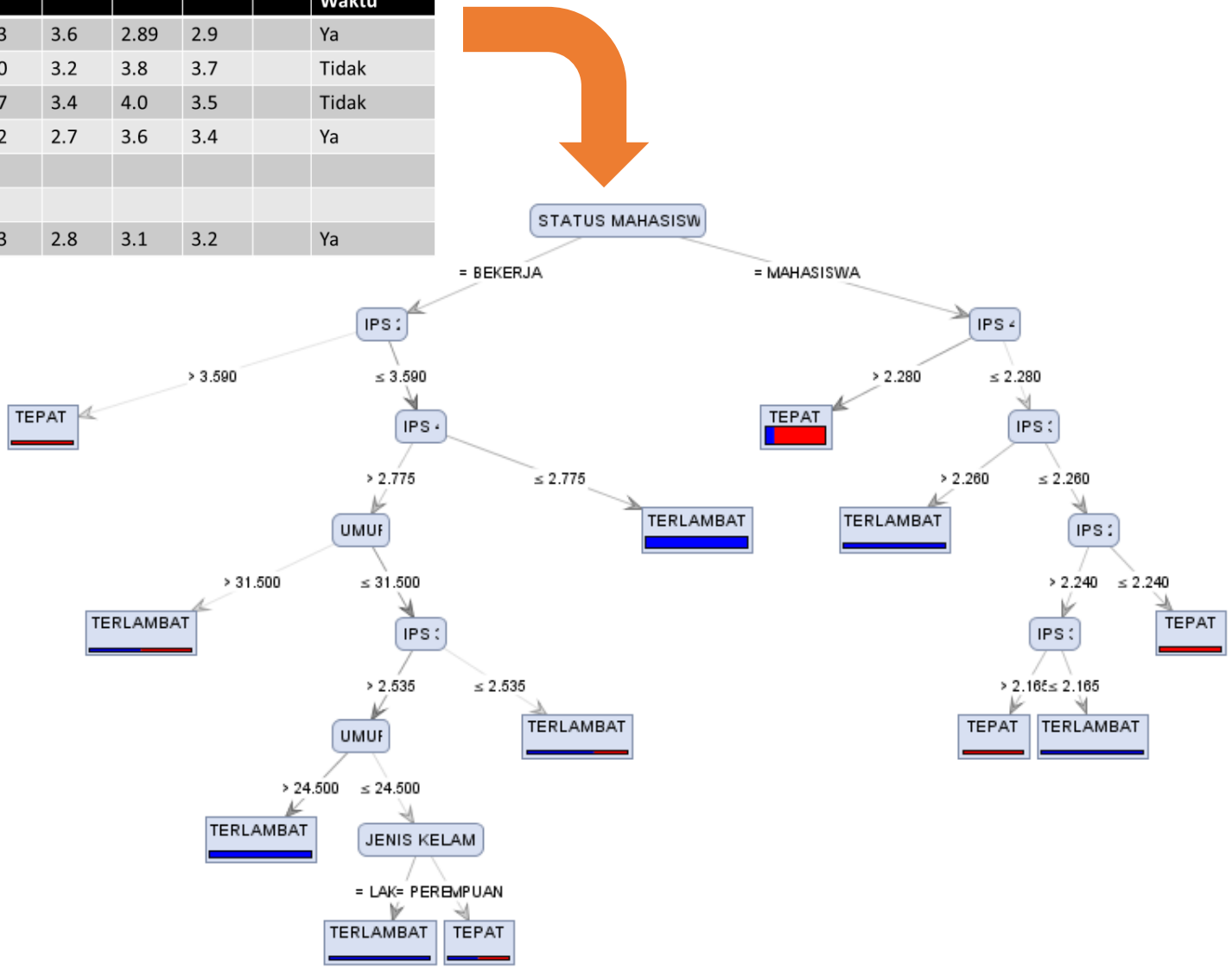
Prediksi Calon Legislatif DKI Jakarta

NAMA PARTAI POLITIK	NAMA CALON LEGESLATIF	JENIS KELAMIN	KECAMATAN	SUARA SAH PARTAI	DAERAH PEMILIHAN	SUARA SAH CALEG	TERPILIH ATAU TIDAK
HANURA	TOTO SUKISNO,BSc	L	LEBAKSIU	18578	1	594	TIDAK
HANURA	EDI PURYANTO,SH	L	SLAWI	18578	1	943	TIDAK
PKB	ELI RETNOWATI,SH	P	SLAWI	18578	1	1730	TIDAK
PKB	SAHYUDIN	L	DUKUHWARU	18578	1	2508	YA
GOLKAR	H.FAJAR SIGIT KUSUMAJAYA,SH	L	SLAWI	18578	2	923	TIDAK
GOLKAR	SUMIRAH	P	TARUB	18578	2	308	TIDAK
GOLKAR	DARYOTO	L	TARUB	18578	2	54	TIDAK
PKS	KHAPIP APRONI,S.Pdi	L	BOJONG	18578	3	1682	TIDAK
PKS	ENDANG SUCI RAHAYU	P	JATINEGARA	18578	3	918	TIDAK
PDI-P	KH.CHAFIDZ ISA MUFTI ,LC	L	SLAWI	18578	3	87	TIDAK
.....	.....	.....	.....	.....	.....	.....	.....

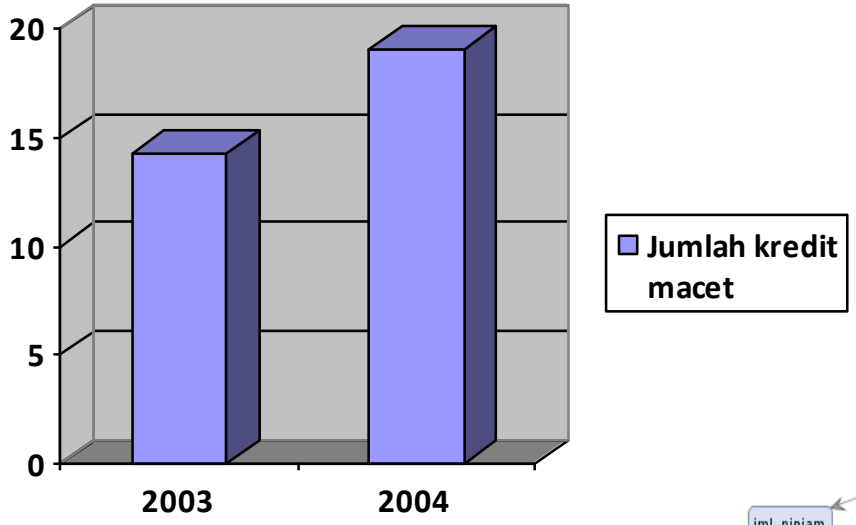


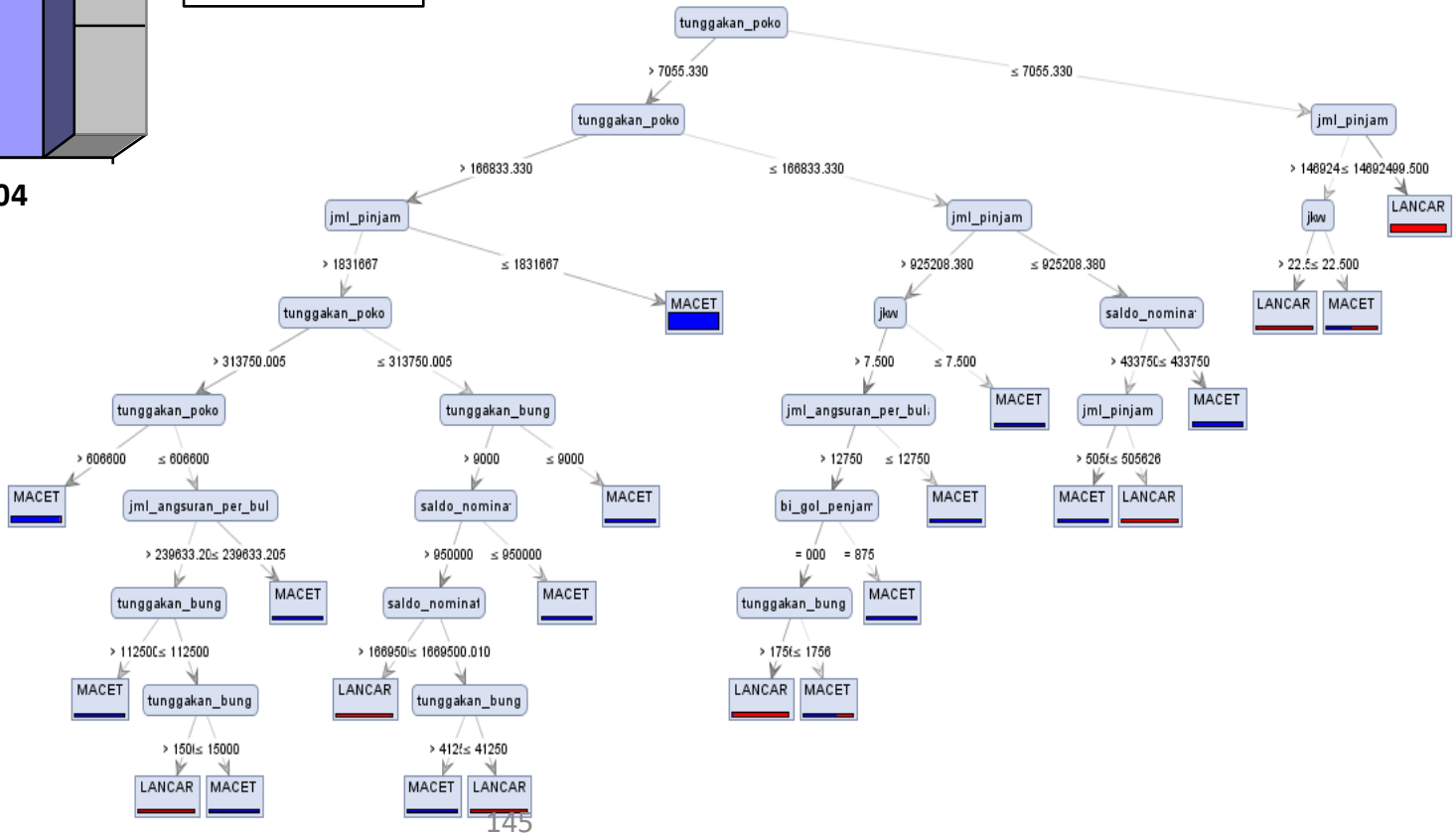
Prediksi Kelulusan Mahasiswa

NIM	Gender	Nilai UN	Asal Sekolah	IPS1	IPS2	IPS3	IPS 4	...	Lulus Tepat Waktu
10001	L	28	SMAN 2	3.3	3.6	2.89	2.9		Ya
10002	P	27	SMA DK	4.0	3.2	3.8	3.7		Tidak
10003	P	24	SMAN 1	2.7	3.4	4.0	3.5		Tidak
10004	L	26.4	SMAN 3	3.2	2.7	3.6	3.4		Ya
...									
...									
11000	L	23.4	SMAN 5	3.3	2.8	3.1	3.2		Ya

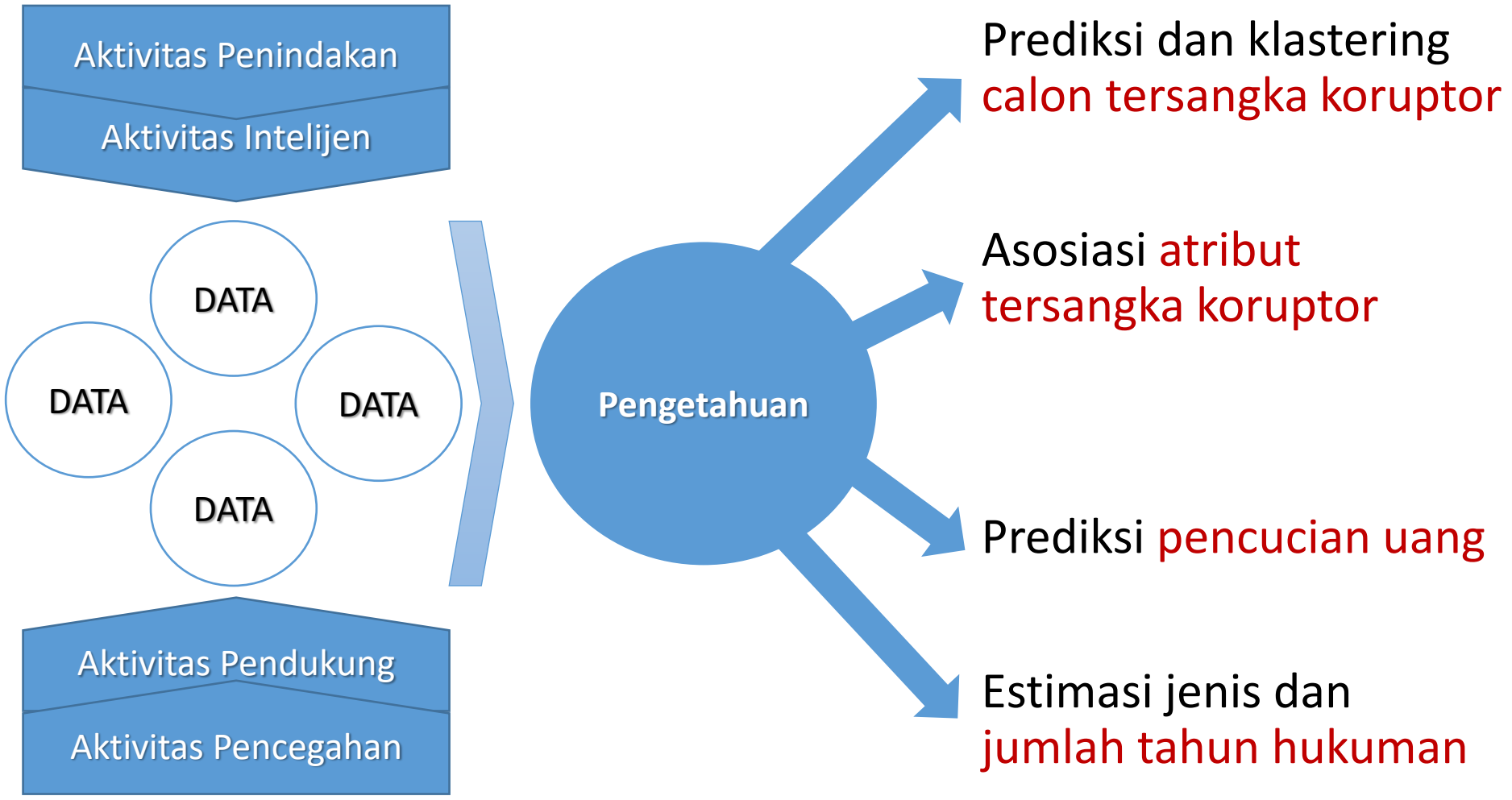


Penentuan Kelayakan Kredit



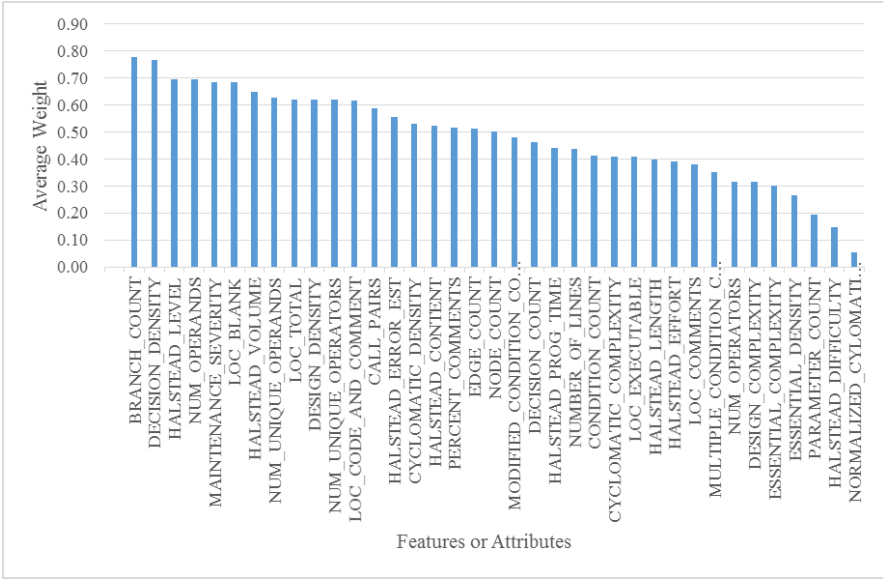


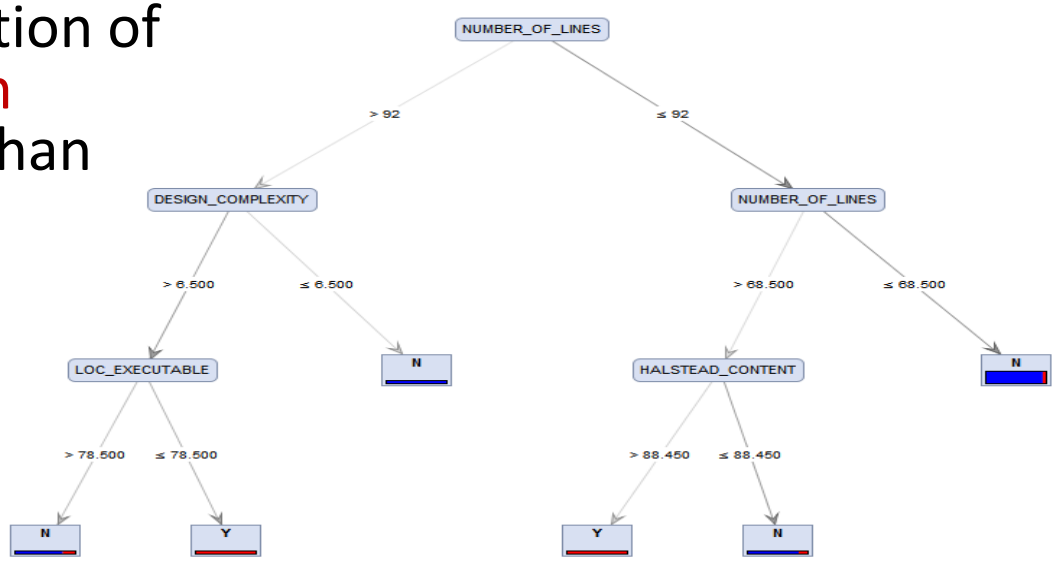
Sistem Prediksi Koruptor



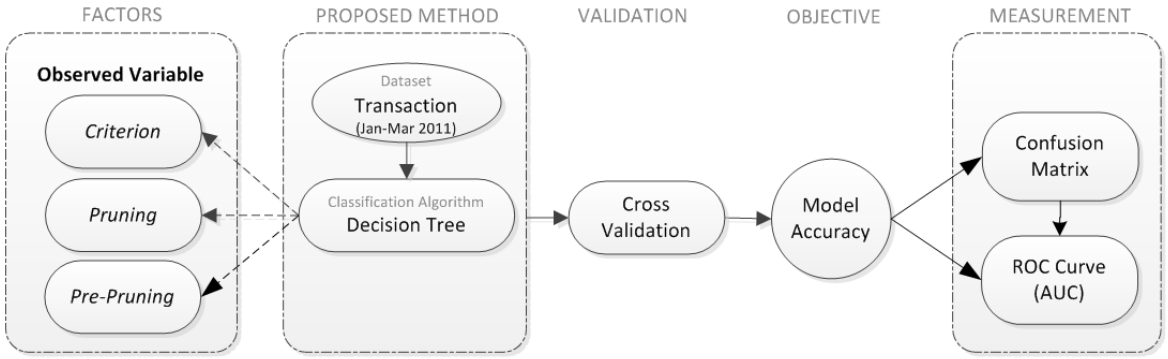
Software Fault Prediction

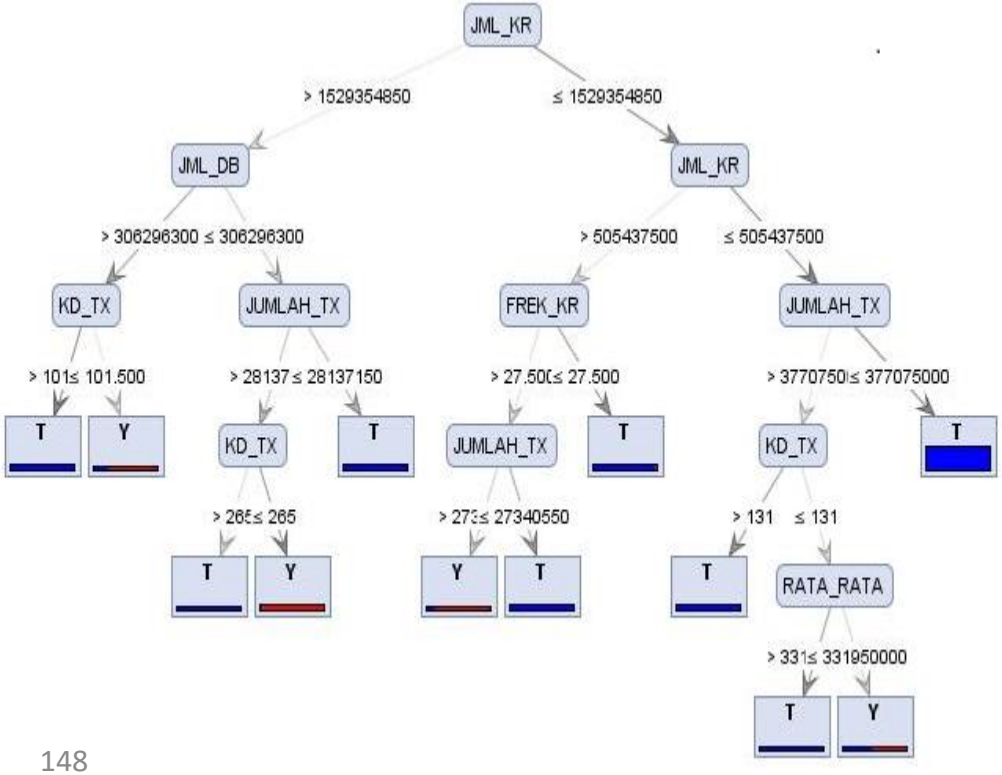
- The cost of capturing and correcting defects is expensive
  - $14,102 per defect in post-release phase *(Boehm & Basili 2008)*
  - $60 billion per year (NIST 2002)
- Industrial methods of manual software reviews activities can find only 60% of defects (Shull et al. 2002)
- The probability of detection of software fault prediction models is higher (71%) than software reviews (60%)





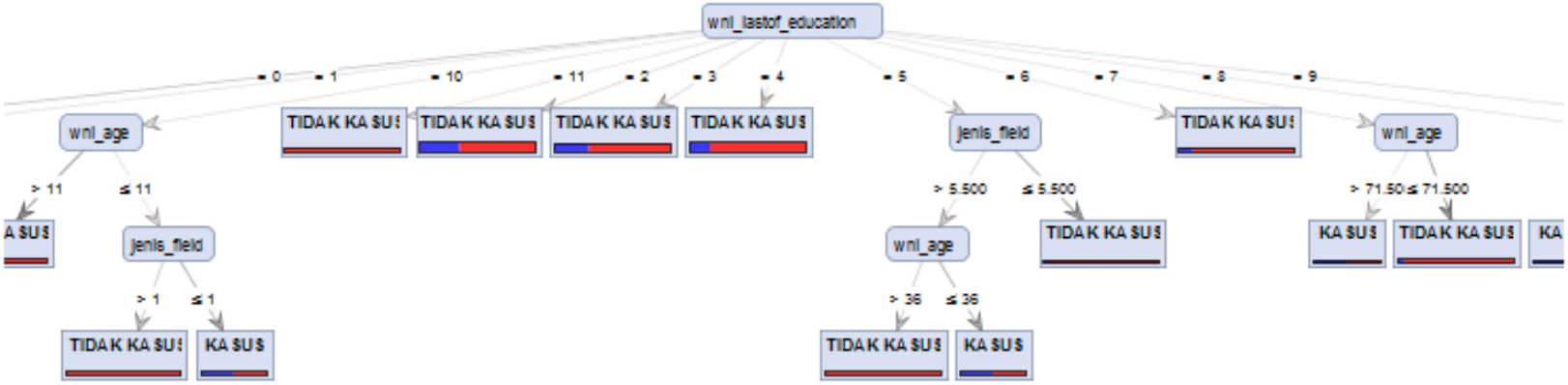
Deteksi Pencucian Uang



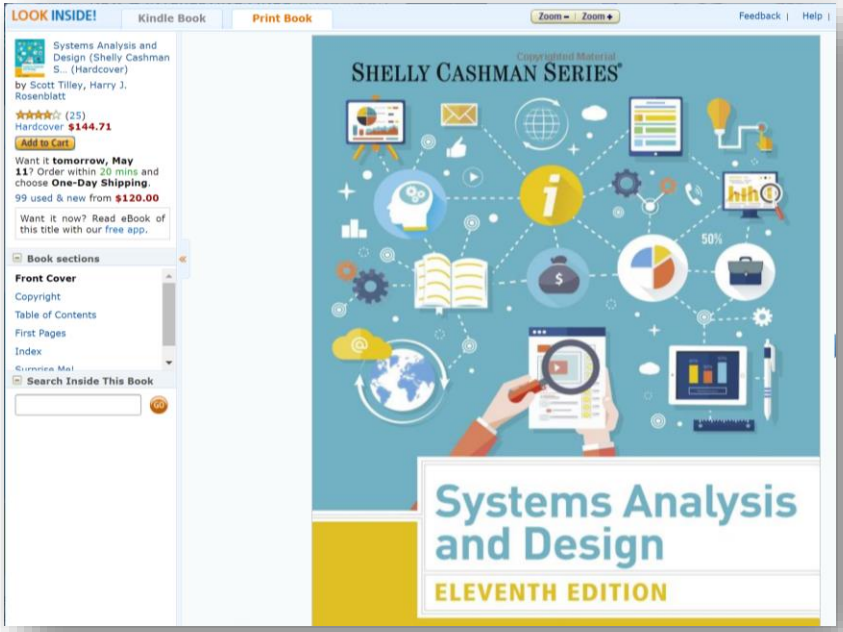


Profiling dan Deteksi Kasus TKI

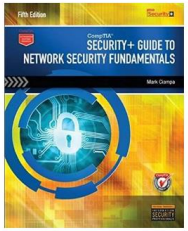
Row No.	status_kasus	wni_age	wni_lastof_...	gender_name	wni_marital...	wni_local_p...	self_report_...	jenis_field
1	KASUS	-183	3	Perempuan	5	32	PEA	3
2	KASUS	-181	0	Perempuan	5	32	Yordania	6
3	KASUS	-4	0	Perempuan	0	36	RRC	6
4	KASUS	-3	0	Perempuan	0	33	Suriah	6
5	KASUS	-1	0	Laki-laki	0	12	Libya	2
6	KASUS	-1	0	Perempuan	0	32	Libanon	6
7	KASUS	0	0	Laki-laki	0	11	Jepang	2
8	KASUS	0	0	Laki-laki	0	11	Jepang	5
9	KASUS	0	0	Laki-laki	0	11	Libya	2
10	KASUS	0	0	Laki-laki	0	11	Malaysia	3
11	KASUS	0	0	Laki-laki	0	11	Malaysia	6
12	KASUS	0	0	Laki-laki	0	11	Yaman	2
13	KASUS	0	0	Laki-laki	0	12	Amerika Seri...	3



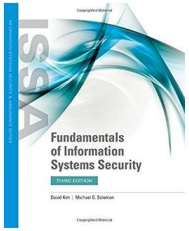
Aturan Asosiasi di Amazon.com



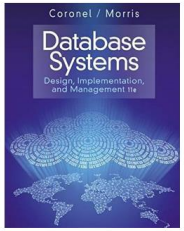
Customers who bought this item also bought



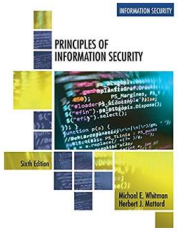
CompTIA Security+ Guide to Network Security Fundamentals (with...
Mark Ciampa
90
Paperback
$134.89 prime



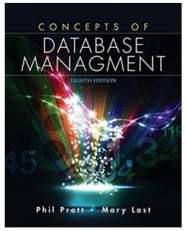
Fundamentals of Information Systems Security
David Kim
14
Paperback
$65.14 prime



Database Systems: Design, Implementation, Management
› Carlos Coronel
100
Hardcover
$90.00 prime



Principles of Information Security (MindTap Course List)
Michael E. Whitman
130
Paperback
$107.20 prime



Concepts of Database Management
› Philip J. Pratt
78
Paperback
$84.09 prime

Information Technology Project Management
Kathy Schwalbe
33
#1 Best Seller in Microso
Project Guides
Paperback
$72.19 prime

From Stupid Apps to Smart Apps

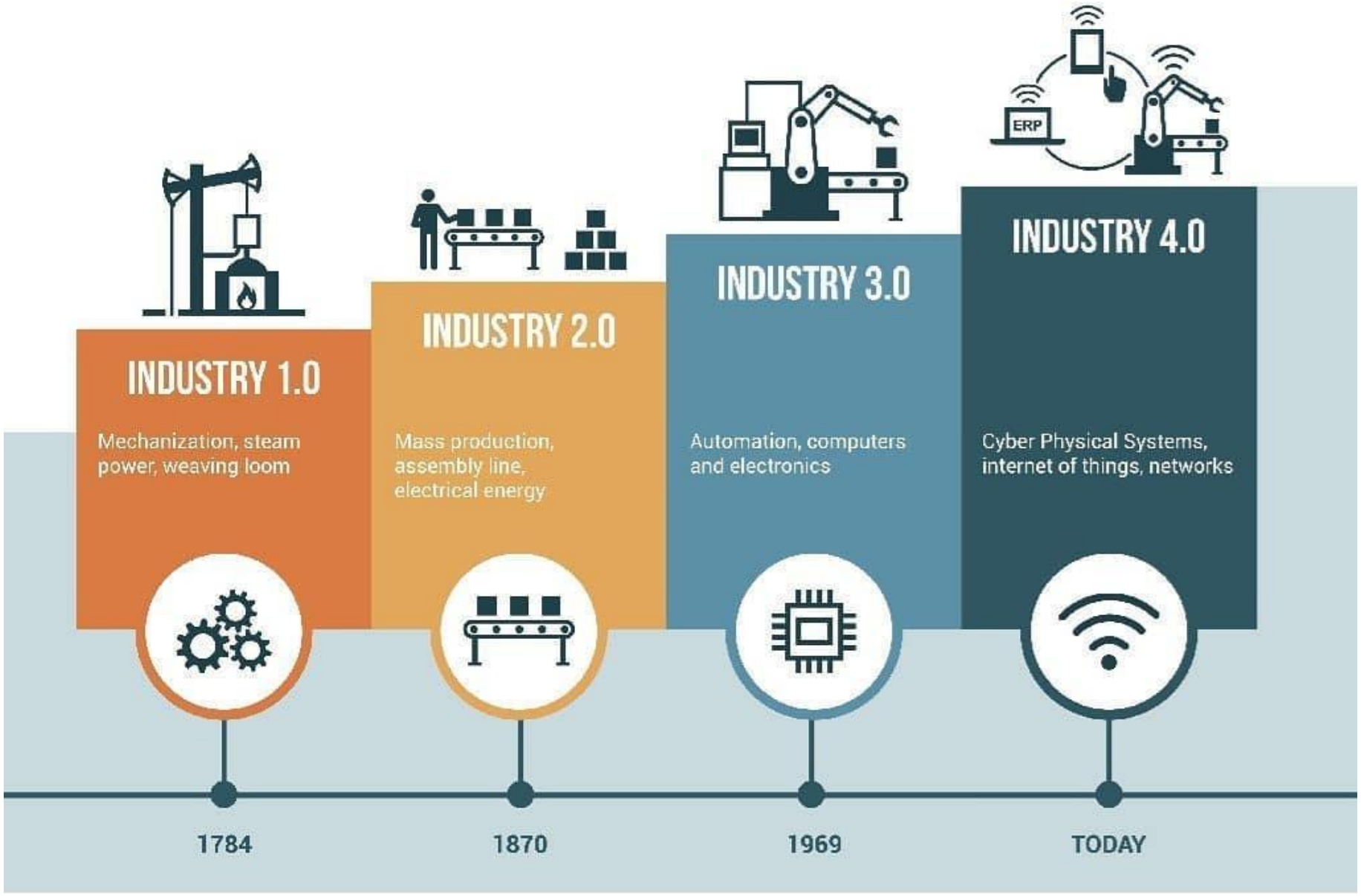
Stupid (CRUD) Applications

- Sistem Informasi Akademik
- Sistem Pencatatan Pemilu
- Sistem Laporan Kekayaan Pejabat
- Sistem Pencatatan Kredit

Smart Applications

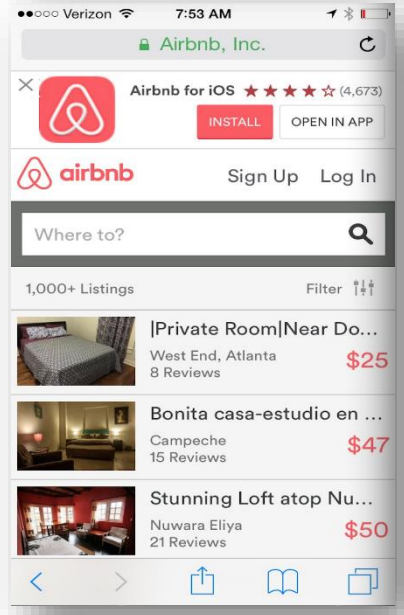
- Sistem Prediksi Kelulusan Mahasiswa
- Sistem Prediksi Hasil Pemilu
- Sistem Prediksi Profile Koruptor
- Sistem Penentu Kelayakan Kredit

Revolusi Industri 4.0



Aplikasi Berbasis Pengetahuan

- Uber - the world's largest taxi company, owns no vehicles
- Google - world's largest media/advertising company, creates no content
- Alibaba - the most valuable retailer, has no inventory
- Airbnb - the world's largest accommodation provider, owns no real estate
- Gojek - perusahaan angkutan umum terbesar di Indonesia, tanpa memiliki kendaraan



Forbes / Tech

JAN 20, 2016 @ 02:39 PM 15,446 VIEWS The Little Black

Report: Why "Data Scientist" Is The Best Job To Pu 2016

Gregory Ferenstein, CONTRIBUTOR

FULL BIO

Opinions expressed

(Ferenstein Wire)
jobs in America, a
company review s
voluntary reviews
company's massiv
a composite score
openings, and car

According to the r
Scientist is an imp

CNBC

JOBS

ECONOMY | WORLD ECONOMY | US ECONOMY | THE FED | CENTRAL BANKS | JOBS |

Data science jobs top Glassdoor survey for best work-life balance

Uptin Saiidi | @uptin
Tuesday, 4 Oct 2016 | 3:40 AM ET
CNBC

glassdoor Jobs Companies Salaries Interviews Sign In
Search Jobs or Companies... Employers | Try Free Job Postings

- Employees' Choice Awards
- Other Lists
  - Oddball Interview Questions
  - Best Jobs
  - Best Cities for Jobs
- Trends

Additional Resources
Award FAQ
Trends FAQ
Free Employer Account
Press Center

25 Best Jobs in America

2.5k Shares

Want a new job? Glassdoor is here to help, identifying the 25 Best Jobs in America for 2016. The jobs that make this list have the highest overall Glassdoor Job Score, determined by combining three key factors – number of job openings, salary and career opportunities rating. These jobs stand out across all three categories.

United States 2016

1

Data Scientist

Job Openings	1,736
Median Base Salary	$116,840
Career Opportunity	4.1
Job Score	4.7

2

Tax Manager

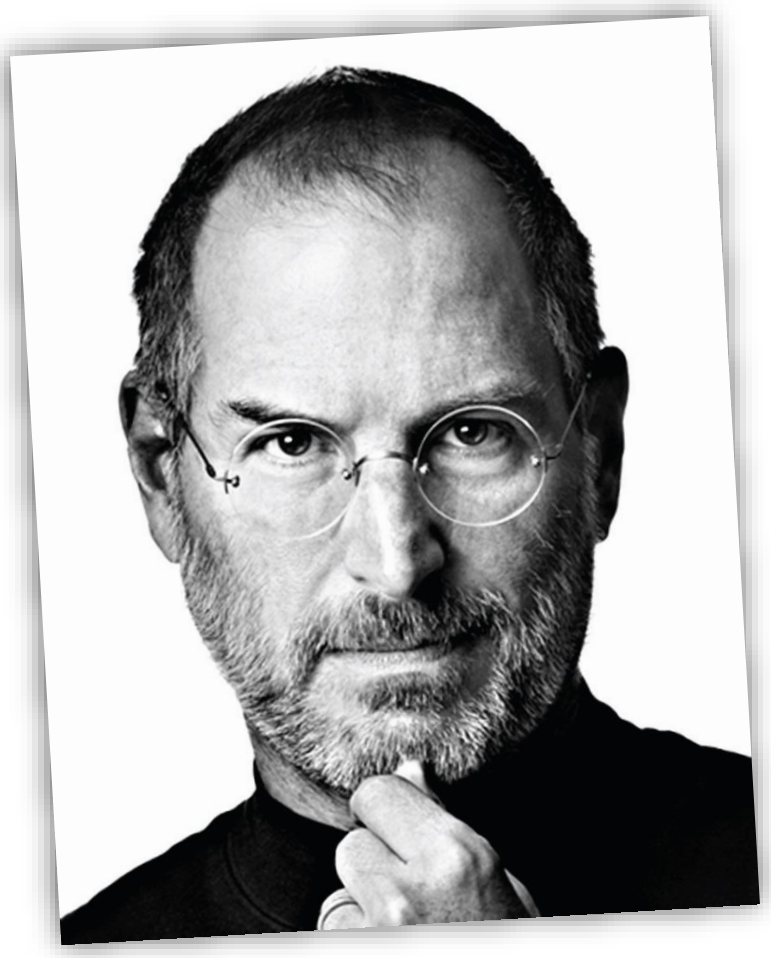
Job Openings	1,574
Median Base Salary	$108,000
Career Opportunity	3.9
Job Score	4.7

MITOS 10

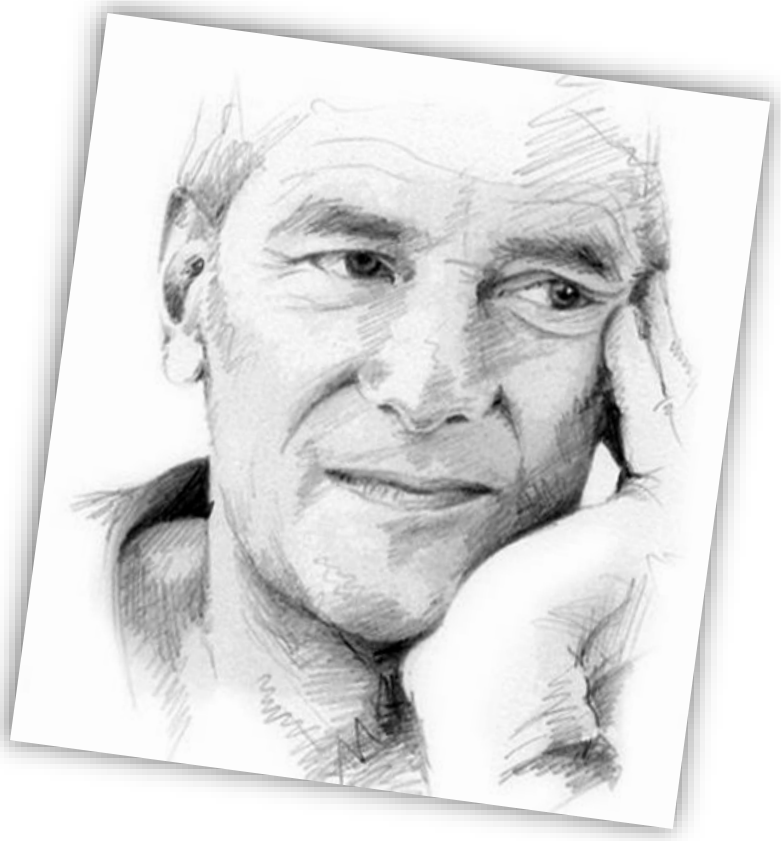
Nggak Suka Software, Tapi Saya Yakin Software Saya Akan Sukses

Satu hal yang membuat saya tetap bertahan adalah bahwa saya **mencintai apa yang saya lakukan**...

(Steve Jobs)




iMac
27-inch LED 16:9 widescreen computer
cBook Air
Extreme



Saya tidak keberatan dengan **5 tahun dan 5126 kegagalan** saya dalam membuat penyedot debu *dual cyclone* tanpa kantung...

(James Dyson)

Kesalahan terbesar saya adalah mencoba membuat alat pancing, padahal **saya tidak suka memancing dan tidak pernah pergi memancing**...

(Eli Harari)

SanDisk

Memory Cards

USB Flash Drives

Saya **suka pesawat**, karena itu saya masuk jurusan konstruksi pesawat, dan akhirnya membangun **industri pesawat terbang**...

(BJ Habibie)

NC212-200

C212-400

CN235-220 MPA

BELL 412 EP

CN235-220M

NAS332 SUPER PUMA

10 MITOS SOFTWARE ENGINEERING

Mitos 1: Cara Sekarang Masih Manual, Karena Itu Butuh Software

Mitos 2: Kemampuan Terpenting Bagi Pengembang adalah Coding

Mitos 3: Kualitas Software Dinilai dari Teknologi yang Digunakan

Mitos 4: SDLC itu Waterfall dan Itu Buruk dan Sudah Kuno

Mitos 5: Project Molor? Tambah Jumlah Pengembang Supaya Bisa Cepat Selesai

Mitos 6: Penghitungan Cost Pengembangan Software itu Pakai Insting dan Pengalaman

Mitos 7: Software Engineering Itu Langsung Praktek Saja, Jangan Kebanyakan Teori

Mitos 8: Penelitian Software Engineering itu Hasilnya adalah Produk Software

Mitos 9: Ingin Membuat Software Seperti Yang Ada Sekarang

Mitos 10: Tidak Suka Software, Tapi Yakin Sukses Bisnis Software

Terima Kasih

Romi Satria Wahono
romi@romisatriawahono.net
http://romisatriawahono.net
08118228331