10 MITOS

Kesalahan Penelitian Computing

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

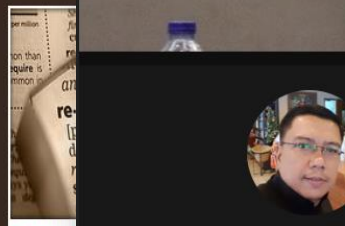
RomiSatriaWahono.Net

Lecture Notes in Software Engineering, Computing Research and Technopreneurship

HOME ABOUT ME IN THE NEWS LECTURES

BAGAI

Pada ar
penelitia
to do Gc
artikel y



TAHAP
PENELI
MAHAS

Jujur, se
kecewa
mahasis
masuk k

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)
lotation (updated January 2015)

Research 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

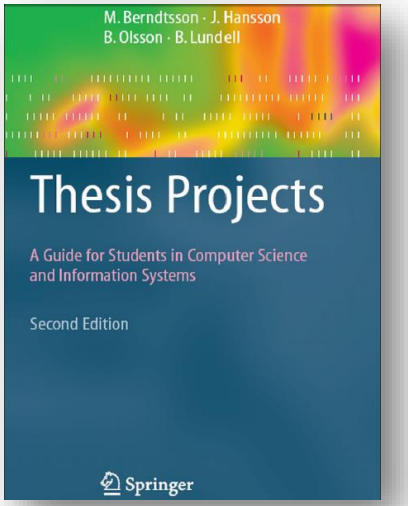
MITOS 1

Penelitian Computing Harus Ada Pengembangan 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)

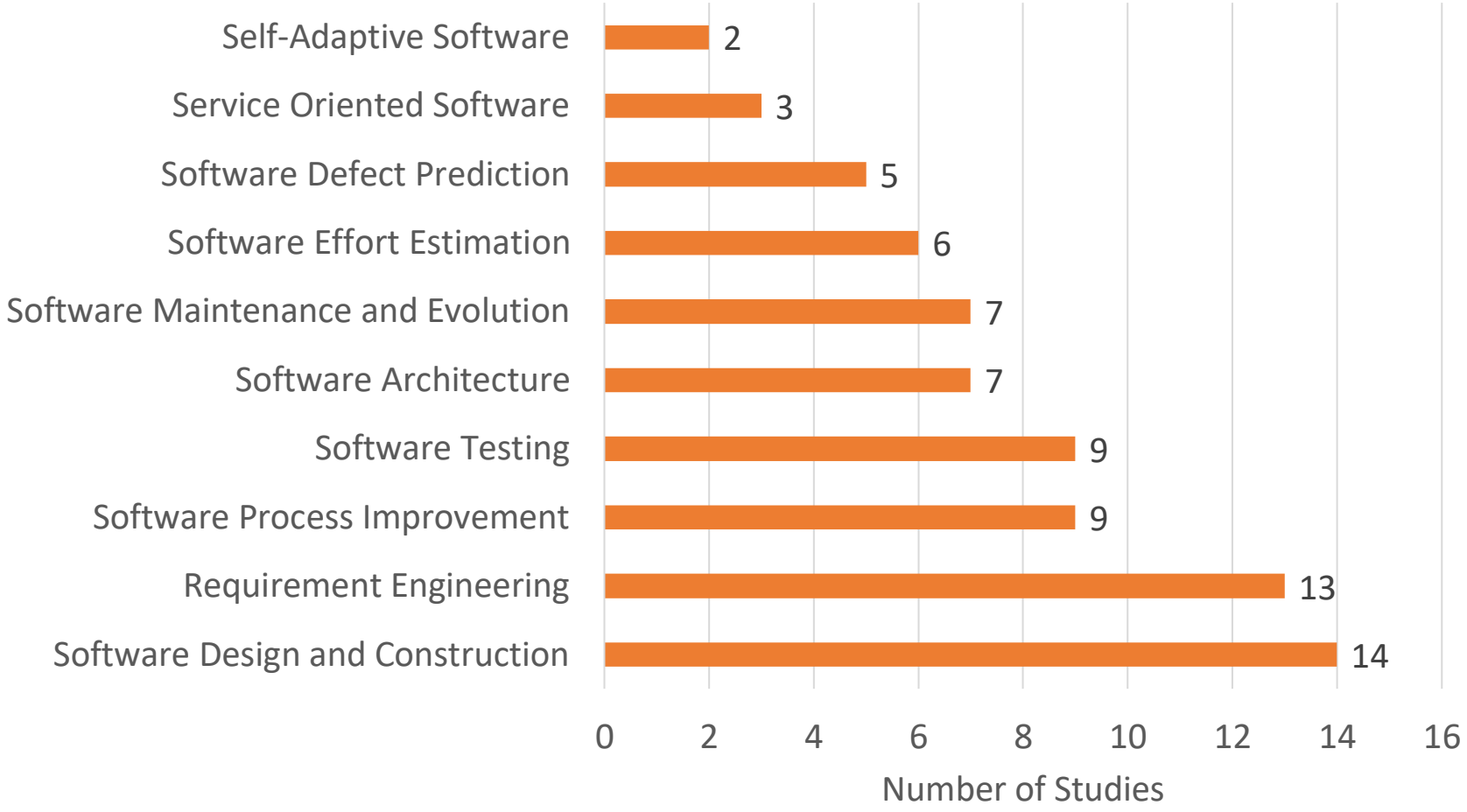
Pengembangan Software vs Penelitian

- 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

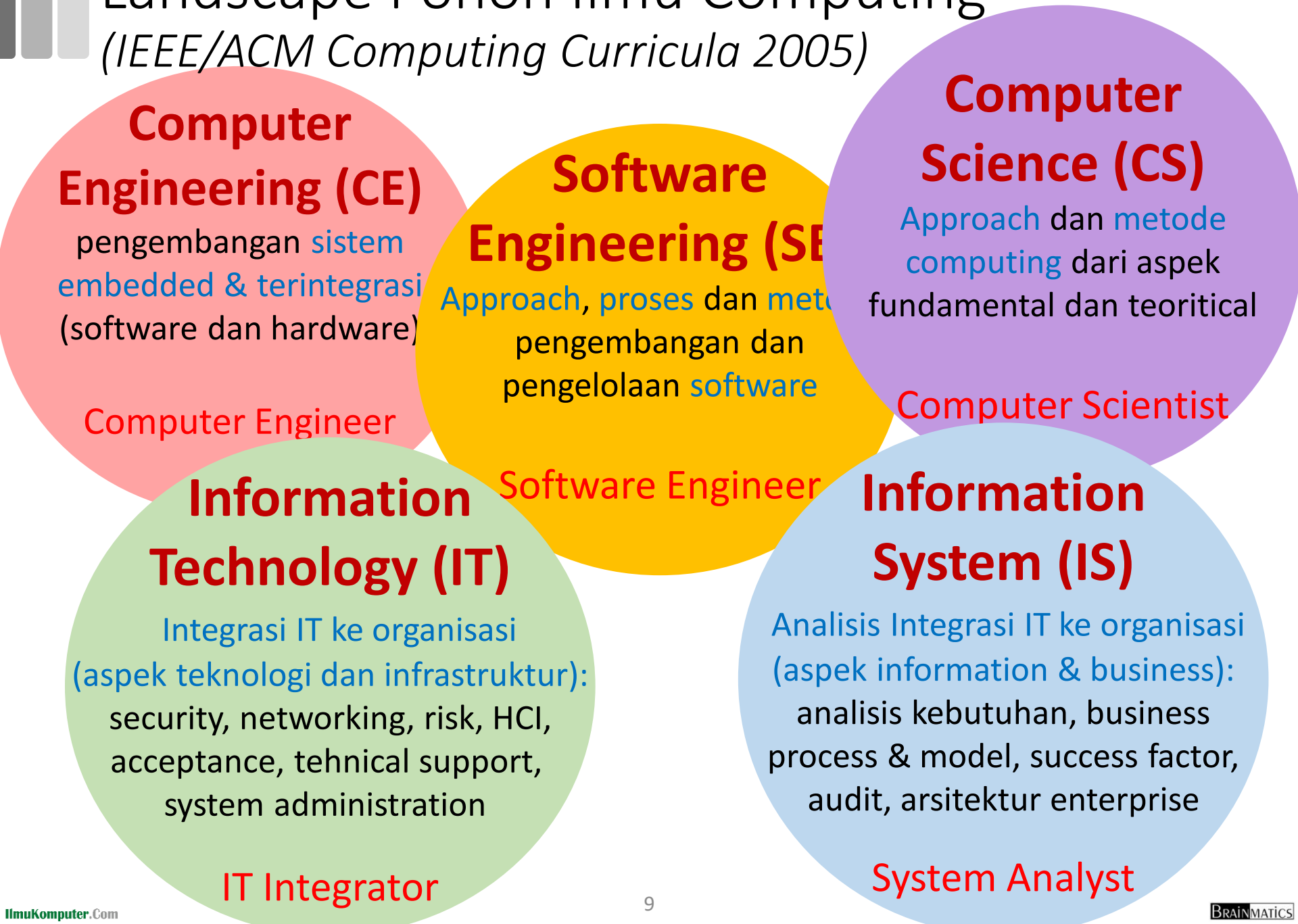
Penelitian Bidang Software Engineering?

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)

Landscape Pohon Ilmu Computing
(IEEE/ACM Computing Curricula 2005)



Computing Program Portfolio

1. **Computer Science**: to serve those students who wish to proceed as generalists in computing or who aspire to graduate study, research positions, or cross-disciplinary innovation
2. **Software Engineering**: to serve students who have the intellectual and technical aptitude to excel as software developers and who want to become expert at developing large scale software, working in teams and producing robust products that meet customer needs
3. **Information Technology**: to serve students who want a computing career that features a mix of technical and people issues rather than a unilateral focus on technology and who are attracted to the widespread need for IT professionals in a variety of organizations and settings
4. **Information Systems**: to serve students who want a career that focuses on the information needs of organizations and who are interested in technology primarily as a vehicle to meet such needs
5. **Computer Engineering**: to serve students who want a career that is focused on developing computer-based devices (embedded systems)

(IEEE/ACM Computing Curricula 2005)

Computer Science Profession

1. They design and implement software. Computer scientists take on challenging programming jobs. They supervise other programmers, keeping them aware of new approaches **(Software Development)**
2. They devise new ways to use computers. Progress in the CS areas of networking, database, and human-computer-interface enabled the development of the World Wide Web. Now CS researchers are working with scientists from other fields to make robots become practical and intelligent aides, to use databases to create new knowledge, and to use computers to help decipher the secrets of our DNA **(Computing Algorithm Applications)**
3. They develop effective ways to solve computing problems. Computer scientists develop the best possible ways to store information in databases, send data over networks, and display complex images. Their theoretical background allows them to determine the best performance possible, and their study of algorithms helps them to develop new approaches that provide better performance **(Computing Algorithm Inventions)**

(Computing Curricula 2005: The Overview Report, ACM and IEEE CS, 2006)

Information Systems Profession

- IS specialists focus on integrating information technology solutions and business processes to meet the information needs of businesses and other enterprises, enabling them to achieve their objectives in an effective, efficient way
- IS specialists concerned with the information that computer systems can provide to aid an enterprise in defining and achieving its goals, and the processes that an enterprise can implement or improve using information technology
- They must understand both technical and organizational factors, and they must be able to help an organization determine how information and technology-enabled business processes can provide a competitive advantage

(Computing Curricula 2005: The Overview Report, ACM and IEEE CS, 2006)

Springer Link

» Sign up / Log in English

Search

Home • Books

ELSEVIER

SEARCH MENU

Home › Journals › Information Systems › Most Downloaded Articles

Submit Your Paper

Supports Open

View Articles

Guide for Autho

Abstracting/ Ind

Track Your Pape

Order Journal

Journal Metrics

CiteScore: 4.2

Impact Factor: 2.0

5-Year Impact Fact

Source Normalize (SNIP): 2.444

SCImago Journal

Most Downloaded Information Systems Articles

Downloaded *Cited*

1 January 2004
Evaluating collaborative filtering recommender...
Jonathan L. Herlocker, Joseph A. Konstan, + 2
Total Downloads 30,505

2 April 1994
Technological frames: making sense of information technolog...
Wanda J. Orlikowski, Debra C. Gash
Total Downloads 15,429

3 April 1994
Finding a happy medium: explaining the negative effects ...
M. L. Markus
Total Downloads 12,879

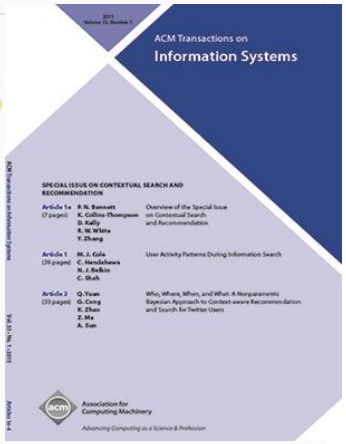
4 October 2005
Evolution of web site design patterns
Melody Y. Ivory, Rodrick Megraw
Total Downloads 12,707

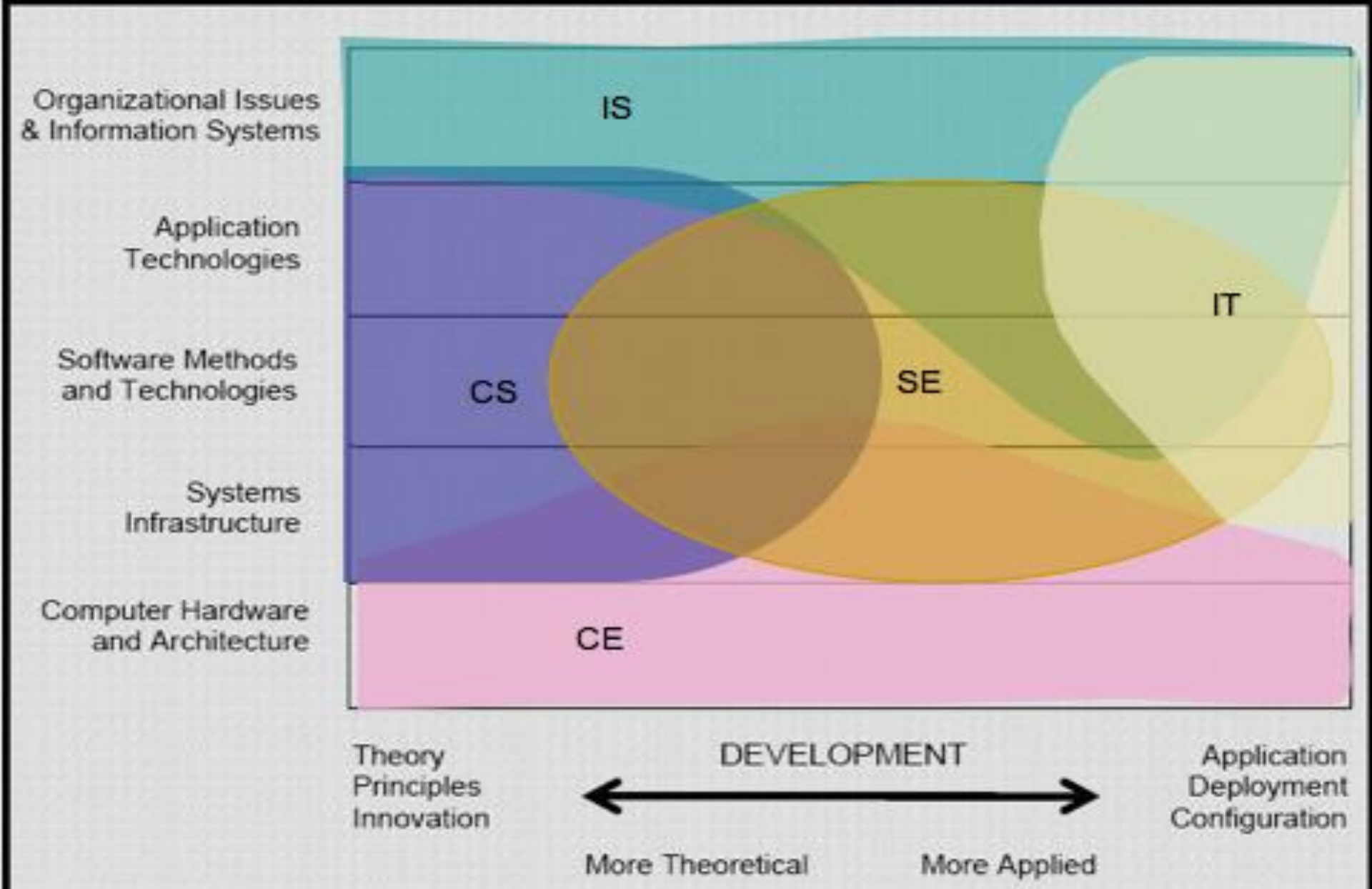
5 January 2004
Item-based top-*N* recommendation algorithms
Mukund Deshpande, George Karypis
Total Downloads 9,546

6 April 1994
Information systems strategy and implementation: a cas...
G. Walsham, T. Waema
Total Downloads 9,146

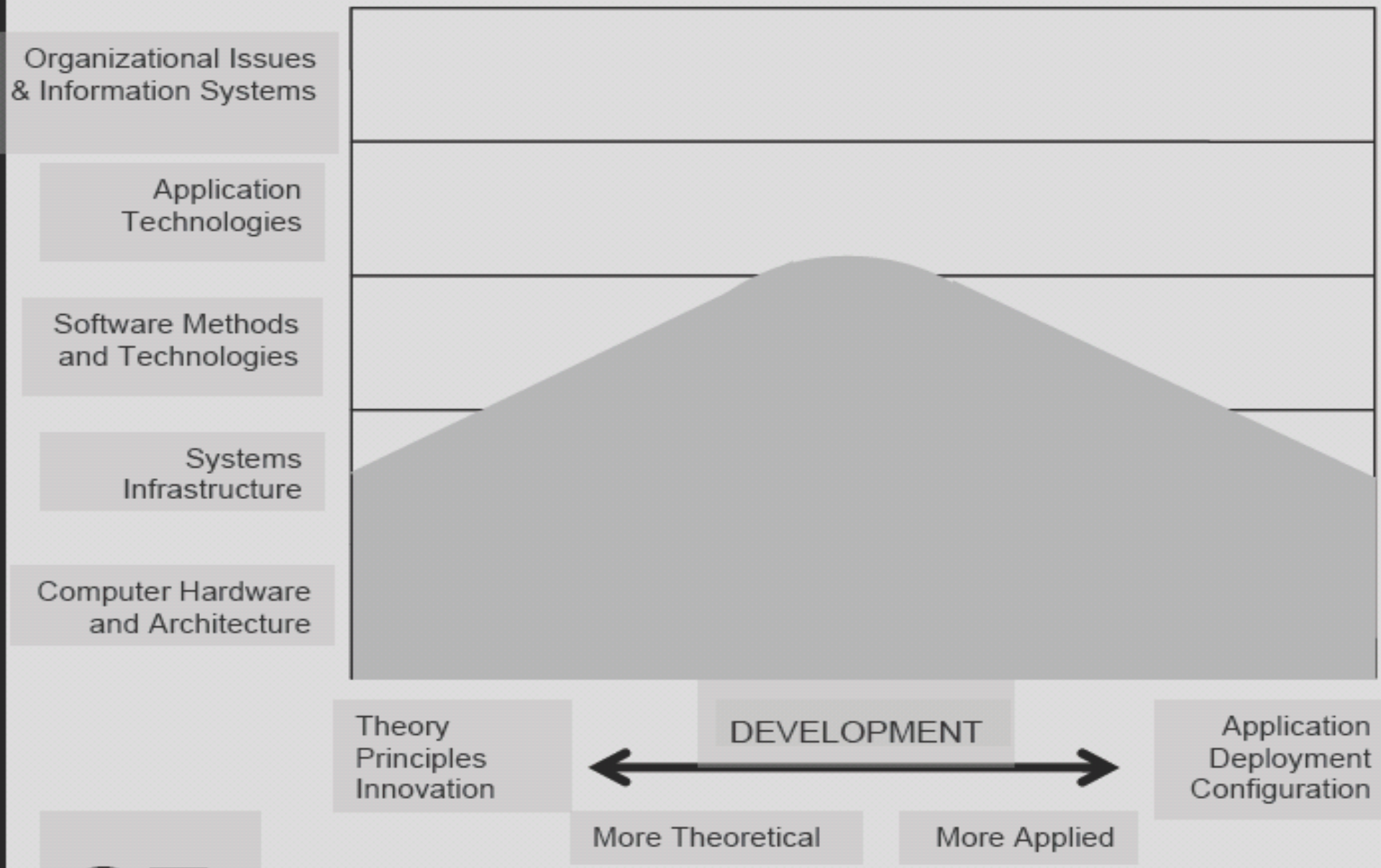
7 January 2004
Ontological user profiling in recommender...
Stuart E. Middleton, Nigel R. Shadbolt, David C. De Roure
Total Downloads 8,632

8

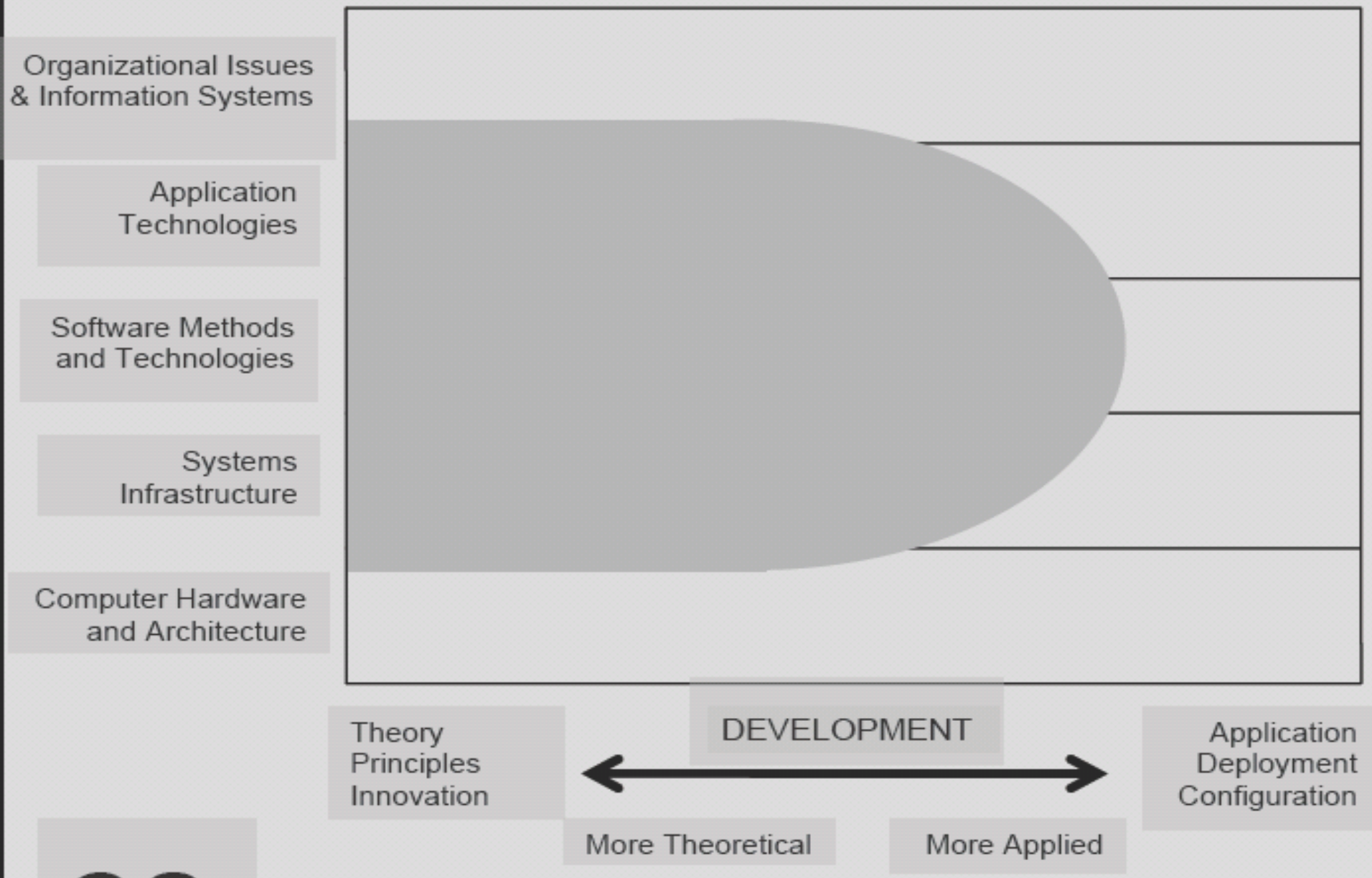


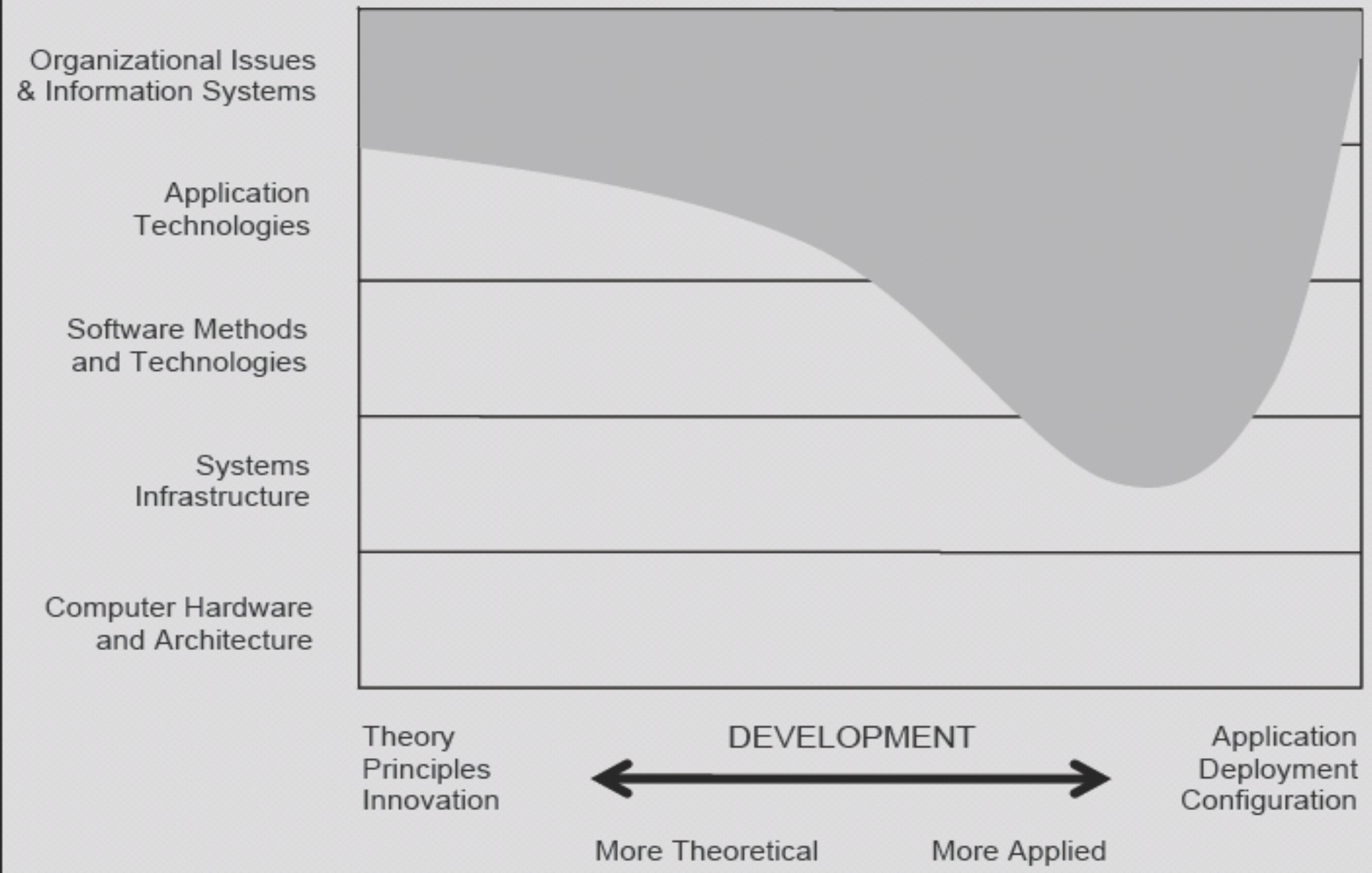


Computing Curricula 2005

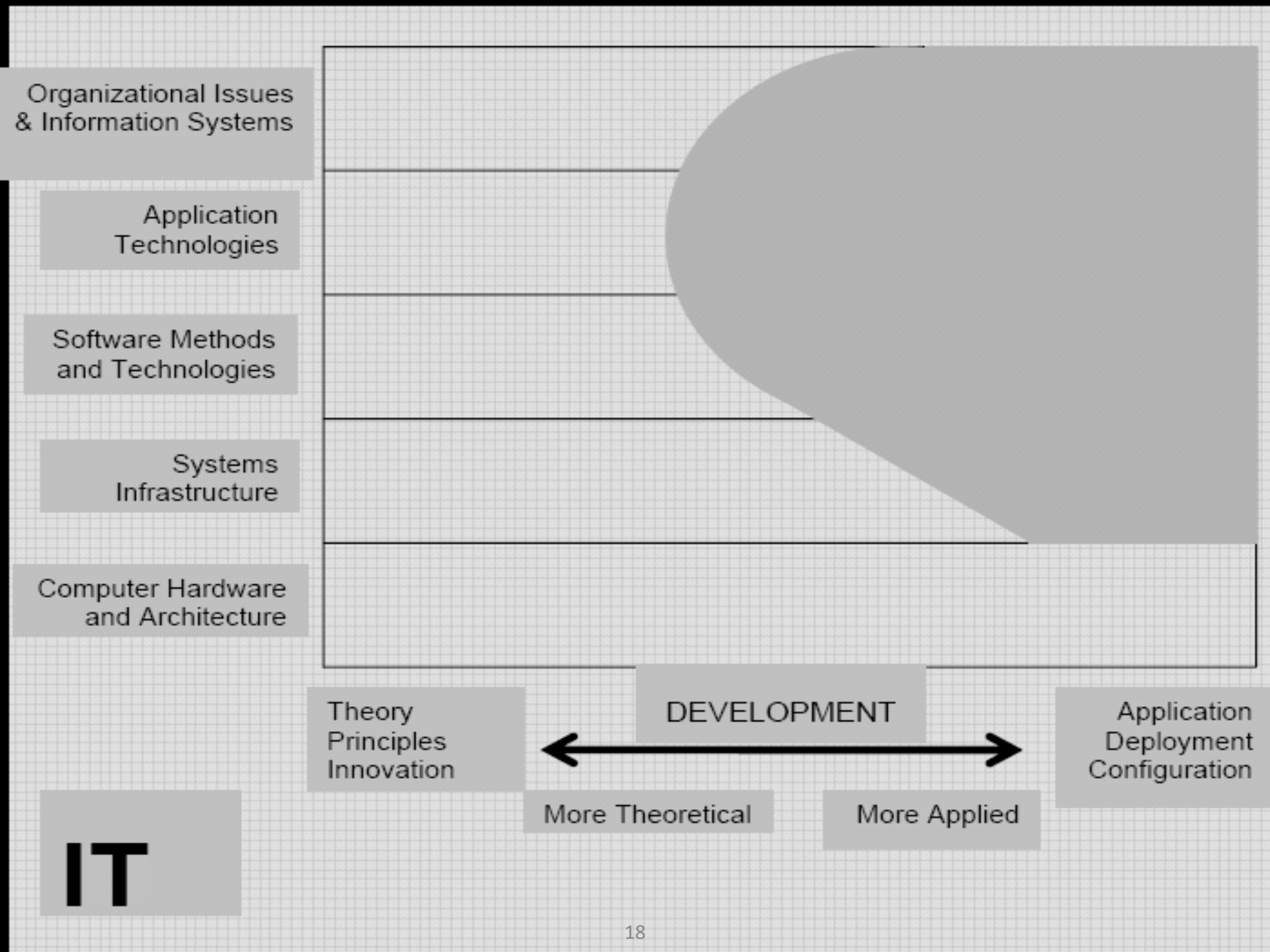

Organizational Issues & Information Systems
Application Technologies
Software Methods and Technologies
Systems Infrastructure
Computer Hardware and Architecture
Theory
Principles
Innovation
DEVELOPMENT
Application
Deployment
Configuration
More Theoretical
More Applied

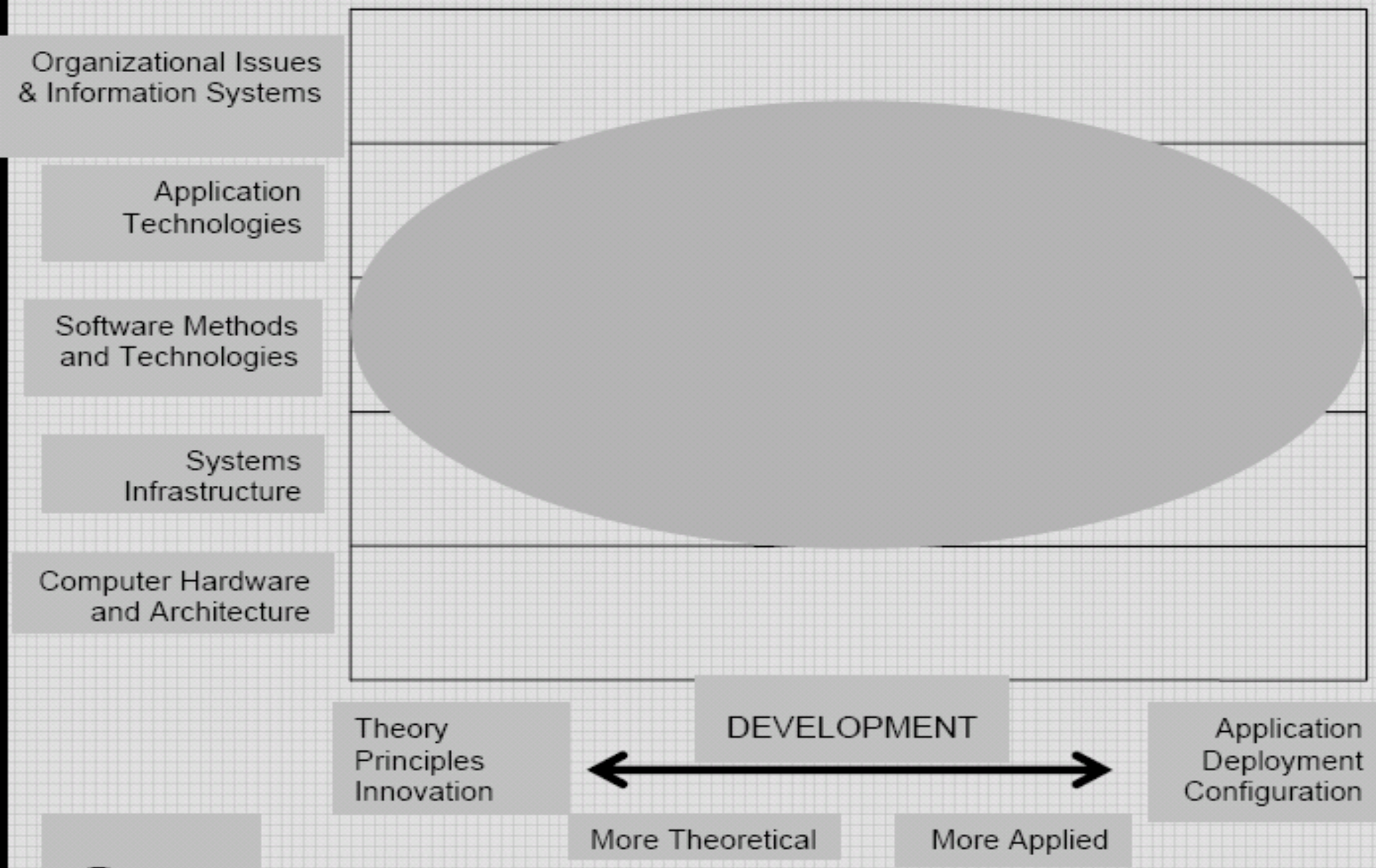
CE


Organizational Issues & Information Systems
Application Technologies
Software Methods and Technologies
Systems Infrastructure
Computer Hardware and Architecture
Theory Principles Innovation
DEVELOPMENT
Application Deployment Configuration
More Theoretical
More Applied
CS


Organizational Issues & Information Systems
Application Technologies
Software Methods and Technologies
Systems Infrastructure
Computer Hardware and Architecture
Theory
Principles
Innovation
DEVELOPMENT
Application
Deployment
Configuration
More Theoretical
More Applied

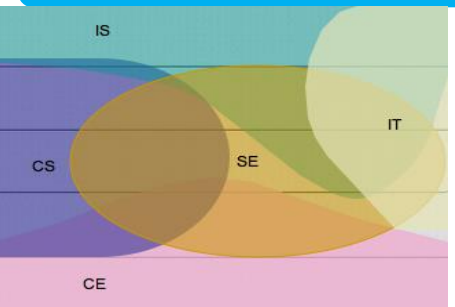
IS


Organizational Issues & Information Systems
Application Technologies
Software Methods and Technologies
Systems Infrastructure
Computer Hardware and Architecture
Theory
Principles
Innovation
DEVELOPMENT
Application
Deployment
Configuration
More Theoretical
More Applied
IT


Organizational Issues & Information Systems
Application Technologies
Software Methods and Technologies
Systems Infrastructure
Computer Hardware and Architecture
Theory Principles Innovation
DEVELOPMENT
Application Deployment Configuration
More Theoretical
More Applied
SE

General Computing Courses -1-

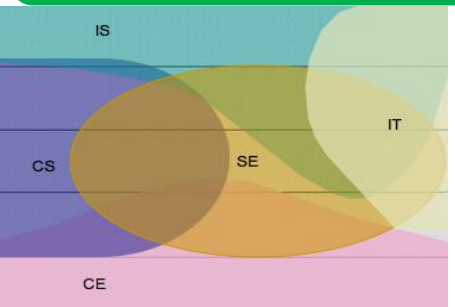
Knowledge Area	CE		CS		IS		IT		SE	
	min	max	min	max	min	max	min	max	min	max
Programming Fundamentals	4	4	4	5	2	4	2	4	5	5
Integrative Programming	0	2	1	3	2	4	3	5	1	3
Algorithms and Complexity	2	4	4	5	1	2	1	2	3	4
Computer Architecture and Organization	5	5	2	4	1	2	1	2	2	4
Operating Systems Principles & Design	2	5	3	5	1	1	1	2	3	4
Operating Systems Configuration & Use	2	3	2	4	2	3	3	5	2	4
Net Centric Principles and Design	1	3	2	4	1	3	3	4	2	4
Net Centric Use and configuration	1	2	2	3	2	4	4	5	2	3
Platform technologies	0	1	0	2	1	3	2	4	0	3
Theory of Programming Languages	1	2	3	5	0	1	0	1	2	4
Human-Computer Interaction	2	5	2	4	2	5	4	5	3	5
Graphics and Visualization	1	3	1	5	1	1	0	1	1	3
Intelligent Systems (AI)	1	3	2	5	1	1	0	0	0	0
Information Management (DB) Theory	1	3	2	5	1	3	1	1	2	5
Information Management (DB) Practice	1	2	1	4	4	5	3	4	1	4
Scientific computing (Numerical mthds)	0	2	0	5	0	0	0	0	0	0
Legal / Professional / Ethics / Society	2	5	2	4	2	5	2	4	2	5
Information Systems Development	0	2	0	2	5	5	1	3	2	4
Analysis of Business Requirements	0	1	0	1	5	5	1	2	1	3
E-business	0	0	0	0	4	5	1	2	0	3



(Computing Curricula 2005: The Overview Report, ACM and IEEE CS, 2006)

General Computing Courses -2-

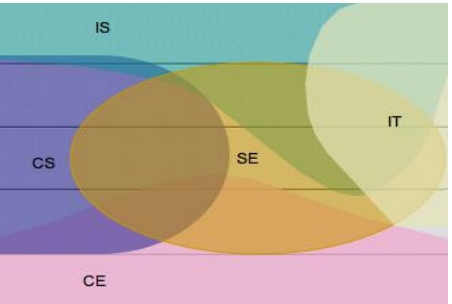
Knowledge Area	CE		CS		IS		IT		SE	
	min	max	min	max	min	max	min	max	min	max
Analysis of Technical Requirements	2	5	2	4	2	4	3	5	3	5
Engineering Foundations for SW	1	2	1	2	1	1	0	0	2	5
Engineering Economics for SW	1	3	0	1	1	2	0	1	2	3
Software Modeling and Analysis	1	3	2	3	3	3	1	3	4	5
Software Design	2	4	3	5	1	3	1	2	5	5
Software Verification and Validation	1	3	1	2	1	2	1	2	4	5
Software Evolution (maintenance)	1	3	1	1	1	2	1	2	2	4
Software Process	1	1	1	2	1	2	1	1	2	5
Software Quality	1	2	1	2	1	2	1	2	2	4
Comp Systems Engineering	5	5	1	2	0	0	0	0	2	3
Digital logic	5	5	2	3	1	1	1	1	0	3
Embedded Systems	2	5	0	3	0	0	0	1	0	4
Distributed Systems	3	5	1	3	2	4	1	3	2	4
Security: issues and principles	2	3	1	4	2	3	1	3	1	3
Security: implementation and mgt	1	2	1	3	1	3	3	5	1	3
Systems administration	1	2	1	1	1	3	3	5	1	2
Management of Info Systems Org.	0	0	0	0	3	5	0	0	0	0
Systems integration	1	4	1	2	1	4	4	5	1	4
Digital media development	0	2	0	1	1	2	3	5	0	1
Technical support	0	1	0	1	1	3	5	5	0	1



(Computing Curricula 2005: The Overview Report, ACM and IEEE CS, 2006)

General Non Computing Courses

Knowledge Area	CE		CS		IS		IT		SE	
	min	max	min	max	min	max	min	max	min	max
Organizational Theory	0	0	0	0	1	4	1	2	0	0
Decision Theory	0	0	0	0	3	3	0	1	0	0
Organizational Behavior	0	0	0	0	3	5	1	2	0	0
Organizational Change Management	0	0	0	0	2	2	1	2	0	0
General Systems Theory	0	0	0	0	2	2	1	2	0	0
Risk Management (Project, safety risk)	2	4	1	1	2	3	1	4	2	4
Project Management	2	4	1	2	3	5	2	3	4	5
Business Models	0	0	0	0	4	5	0	0	0	0
Functional Business Areas	0	0	0	0	4	5	0	0	0	0
Evaluation of Business Performance	0	0	0	0	4	5	0	0	0	0
Circuits and Systems	5	5	0	2	0	0	0	1	0	0
Electronics	5	5	0	0	0	0	0	1	0	0
Digital Signal Processing	3	5	0	2	0	0	0	0	0	2
VLSI design	2	5	0	1	0	0	0	0	0	1
HW testing and fault tolerance	3	5	0	0	0	0	0	2	0	0
Mathematical foundations	4	5	4	5	2	4	2	4	3	5
Interpersonal communication	3	4	1	4	3	5	3	4	3	4



(Computing Curricula 2005: The Overview Report, ACM and IEEE CS, 2006)

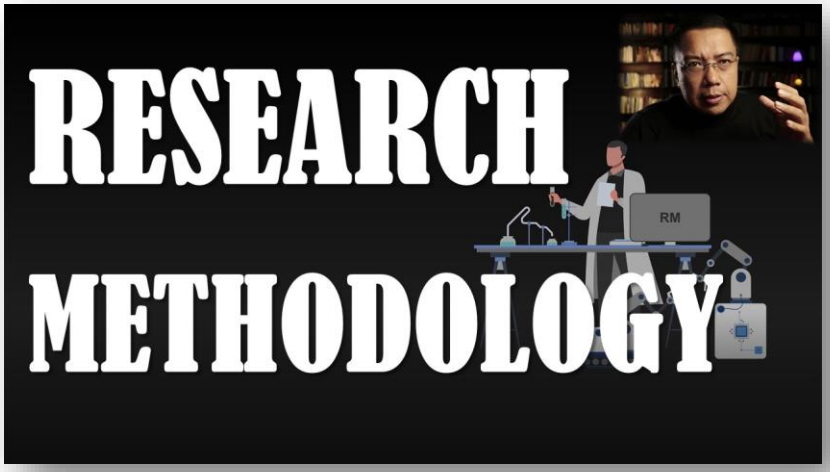
MITOS 2

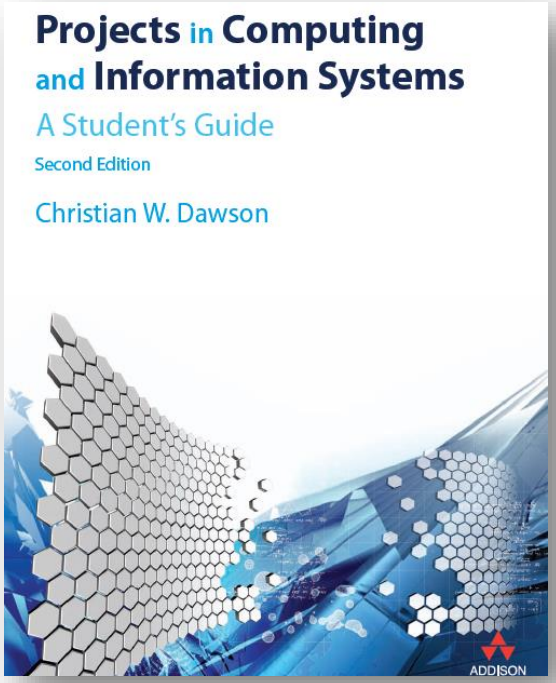
Tujuan Utama Penelitian adalah Adanya Kontribusi ke Masyarakat

Apa Yang Dikejar di Penelitian?

Research is a considered activity, which aims to make an original contribution to knowledge

(Dawson, 2009)



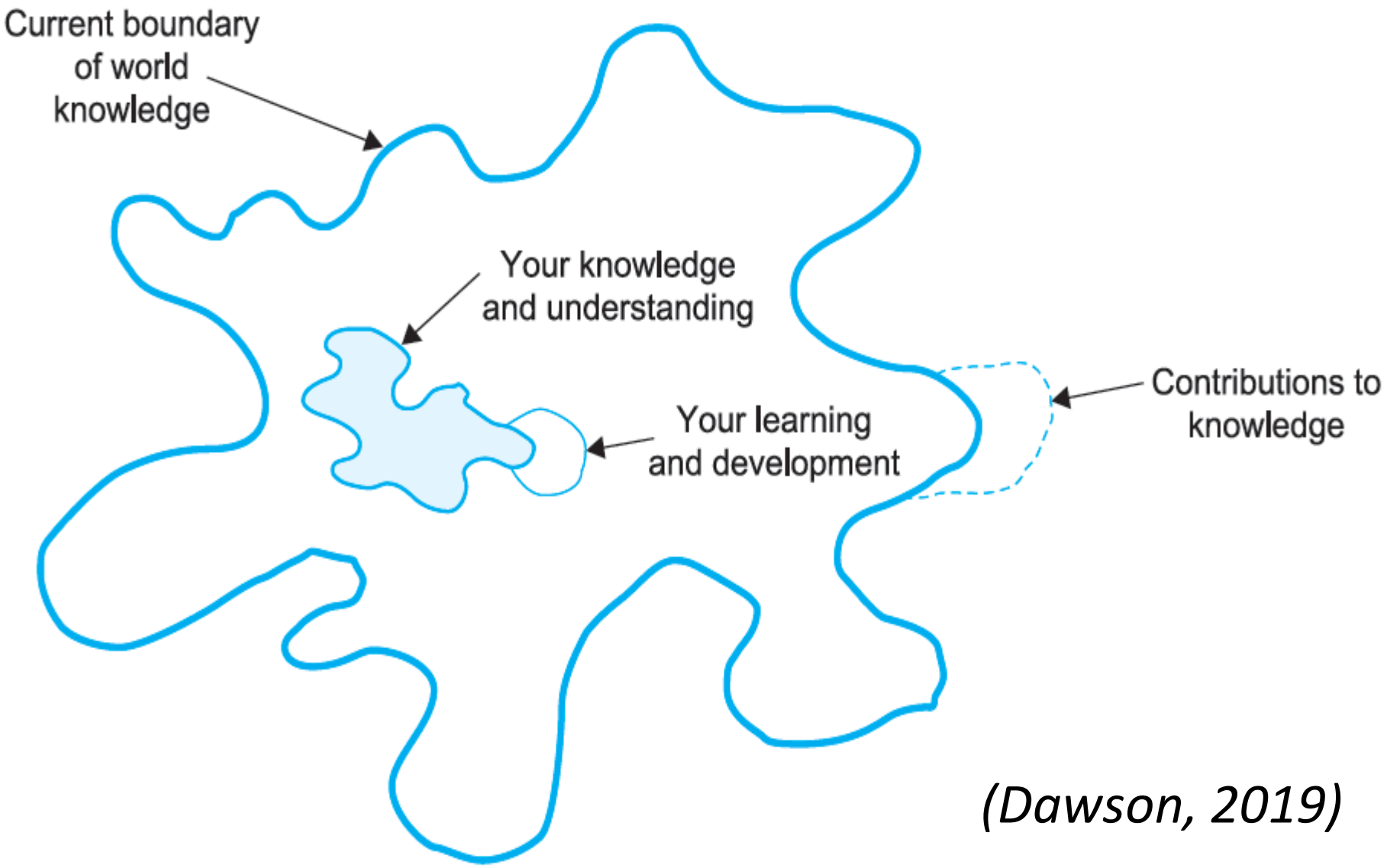


Bentuk Kontribusi ke Pengetahuan

Kegiatan penyelidikan dan investigasi terhadap suatu masalah yang dilakukan secara berulang-ulang dan sistematis, dengan tujuan untuk menemukan atau merevisi teori, metode, fakta, dan aplikasi

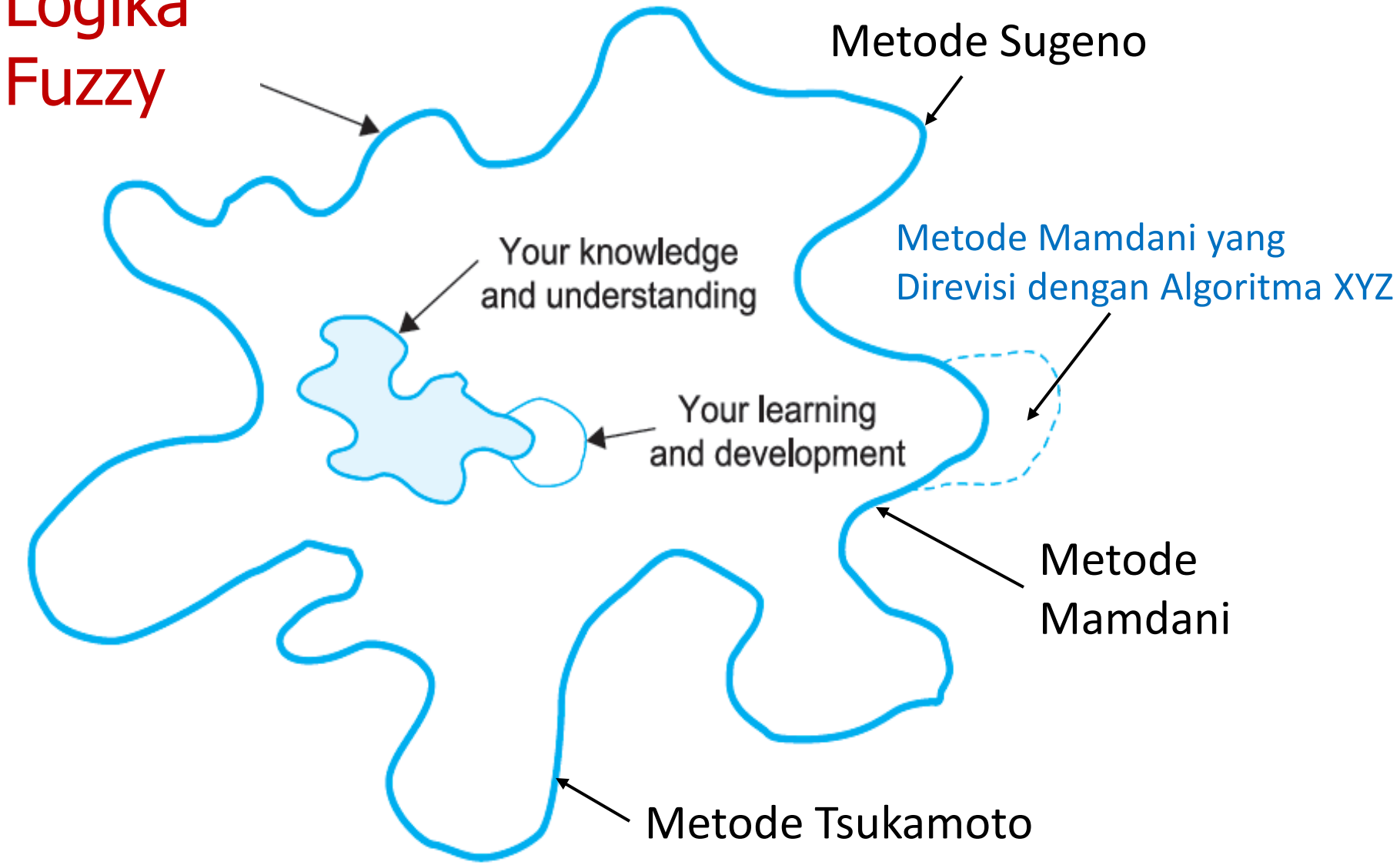
(Berndtsson et al., 2008)

Bentuk Kontribusi ke Pengetahuan

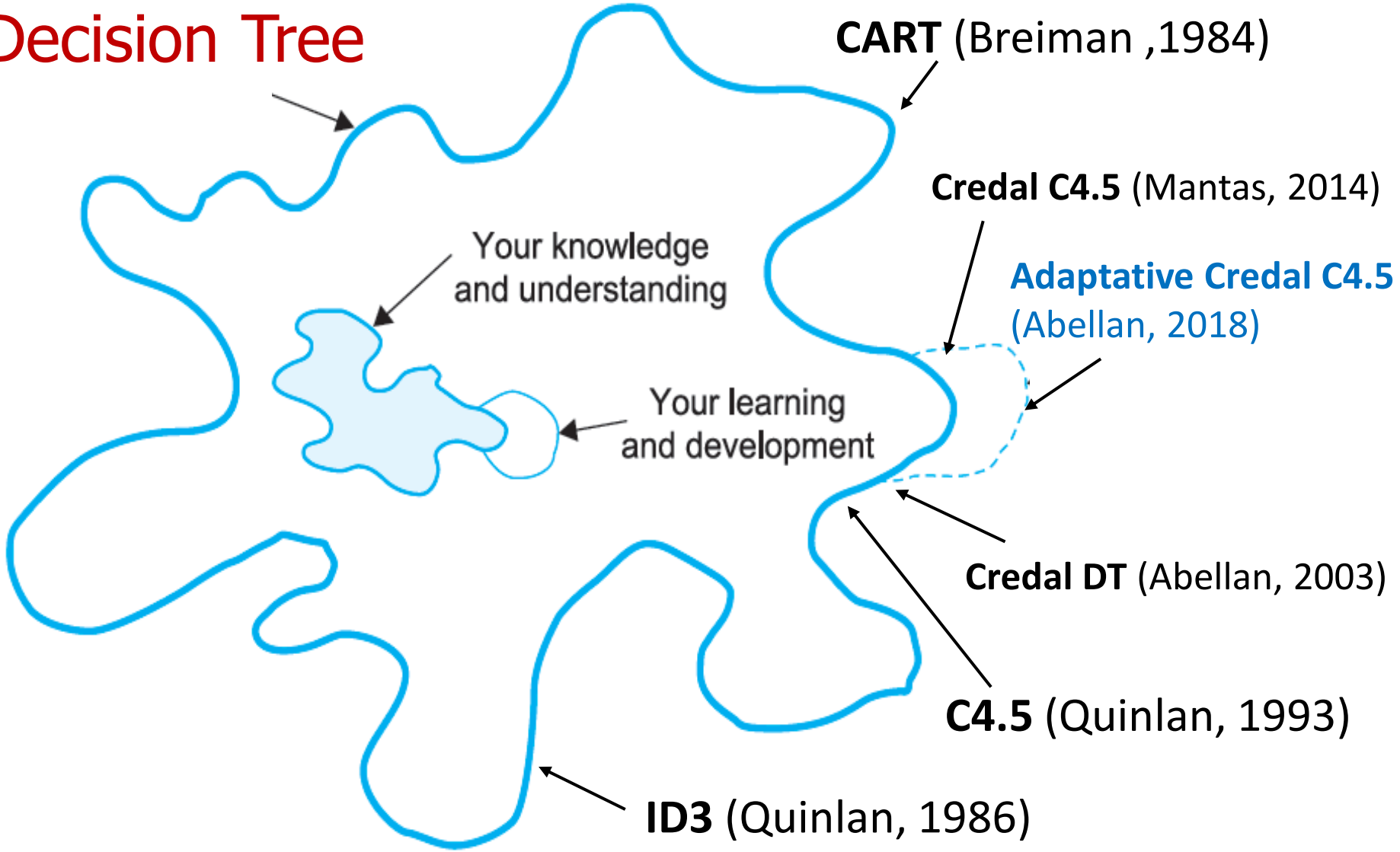


(Dawson, 2019)

Bentuk Kontribusi ke Pengetahuan



Bentuk Kontribusi ke Pengetahuan



Contoh Ide Dasar Penelitian

- Ingin menerapkan Algoritma Genetika
- Untuk penentuan nilai parameter (panjang, tinggi, lebar, dalam, bentuk, jenis material, etc) yang paling optimal pada desain bendungan
- Sehingga bisa menampung kubik air terbesar

Contoh Penelitian Dengan Kontribusi

- Judul:

  Penerapan Metode XYZ untuk Pemecahan Masalah Konvergensi Prematur pada Algoritma Genetika untuk Penentuan Desain Bendungan

- Kontribusi:

  Menerapkan Metode XYZ yang sebelumnya tidak pernah digunakan orang untuk memecahkan masalah konvergensi premature pada Algoritma Genetika

Contoh Penelitian Tanpa Kontribusi

- Penerapan Algoritma Genetika untuk Penentuan Desain Bendungan di Bendungan Jatiluhur
- Penerapan Algoritma Genetika untuk Penentuan Desain Bendungan di Bendungan Gajah Mungkur
- Penerapan Algoritma Genetika untuk Penentuan Desain Bendungan di Bendungan Karang Kates

* banyak peneliti computing di Indonesia yang terjebak dengan penelitian tanpa kontribusi dan hanya mengganti obyek tempat, akhirnya ditolak ketika publikasi ke journal internasional terindeks

Penelitian Yang Memiliki Kontribusi?

- Penerapan algoritma genetika untuk penjadwalan mata kuliah 
  - Penerapan algoritma genetika berbasis *guided local search strategies* untuk penjadwalan mata kuliah (Yang, 2011) 

- Penerapan algoritma C4.5 untuk penentuan kelulusan mahasiswa tepat waktu: Studi Kasus STMIK XYZ 
  - Penerapan algoritma C4.5 dengan penghitungan entropi berbasis metode ABC untuk penentuan kelulusan mahasiswa tepat waktu 

Hanya penelitian dengan kontribusi ke pengetahuan yang bisa menembus jurnal-jurnal internasional terindeks

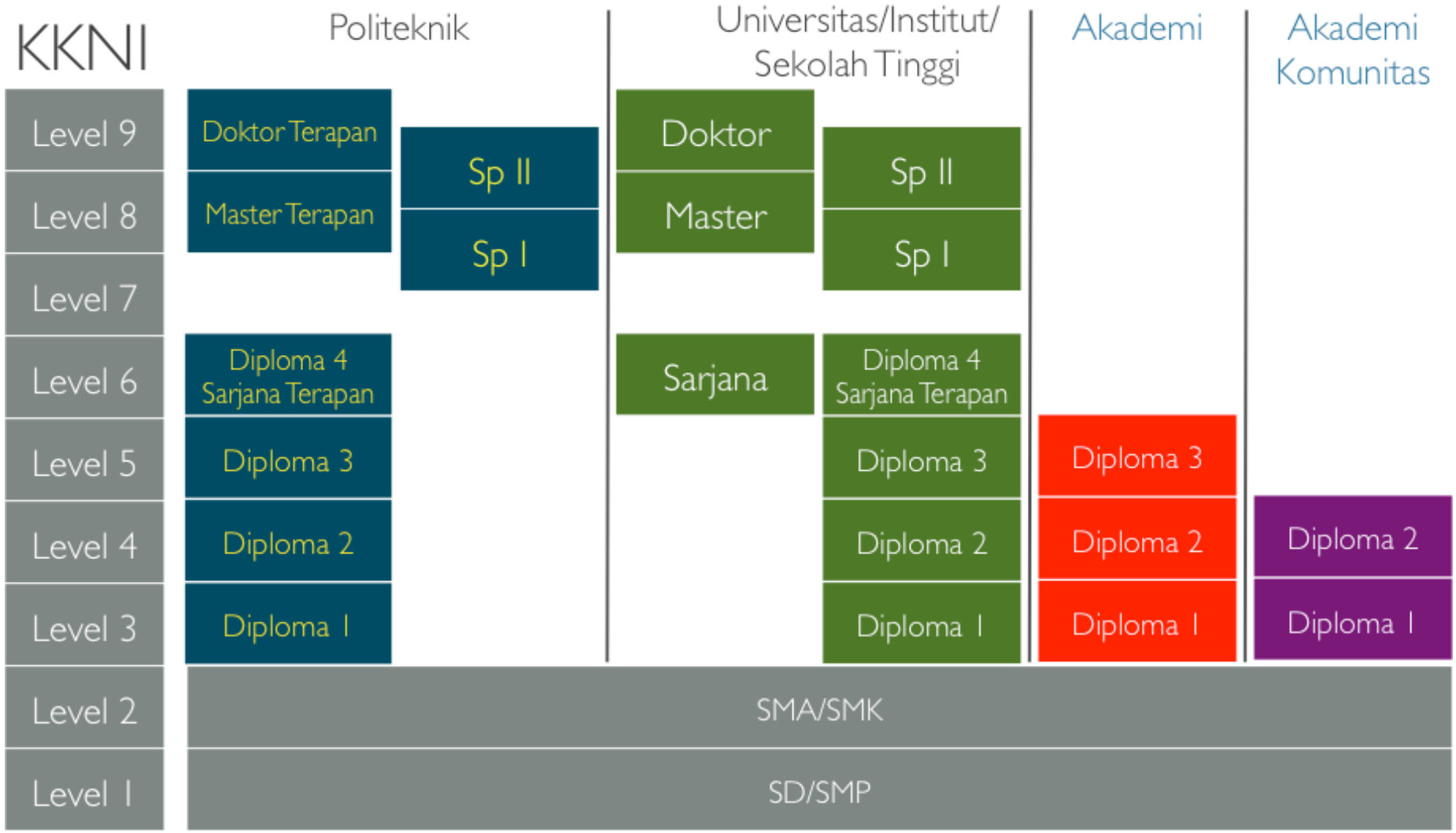
Penelitian Yang Memiliki Kontribusi?

No	Judul	
1	Penerapan Neural Network untuk Prediksi Harga Saham pada Perusahaan ABC	
2	Pemilihan Arsitektur Jaringan pada Neural Network Secara Otomatis dengan Menggunakan Algoritma Semut	
3	Modifikasi Penghitungan Gain dan Entropi untuk Peningkatan Akurasi pada Algoritma C4.5	
4	Penerapan Framework TOGAF untuk Pengembangan Enterprise Architecture pada Organisasi ABC	
5	Penerapan Framework TOGAF yang Dimodifikasi untuk Pengembangan Enterprise Architecture pada Perusahaan Skala Kecil dan Menengah	
6	Penerapan COBIT untuk Tata Kelola Organisasi ABC	
7	Integrasi COBIT dan TOGAF untuk Tata Kelola Organisasi ABC yang Lebih Komprehensif	
8	Penerapan algoritma genetika untuk penjadwalan mata kuliah: Studi Kasus STMIK ABC	

Komparasi Level Penelitian D3/D4 vs S1 vs S2 vs S3

Aspek	Tugas Akhir (D3/D4)	Skripsi (S1)	Tesis (S2)	Disertasi (S3)
Level Kontribusi	Penguasaan Kemampuan Teknis	Pengujian Teori	Pengembangan Teori	Penemuan Teori Baru
Bentuk Kontribusi	Implementasi dan pengembangan	Implementasi dan pengembangan	Perbaikan Secara Inkremental dan Terus Menerus	Substansial dan Invention
Target Publikasi	-	Domestic Conference	International Conference	International Journal

Jenjang Kualifikasi Kelulusan



(Perpres no 8 tahun 2012 tentang Kerangka Kualifikasi Nasional Indonesia (KKNI))

Level 6 (Sarjana)

- Menguasai konsep teoritis bidang pengetahuan tertentu secara umum dan konsep teoritis bagian khusus dalam bidang pengetahuan tersebut secara mendalam, serta mampu memformulasikan penyelesaian masalah prosedural
- Mampu mengaplikasikan bidang keahliannya dan memanfaatkan IPTEKS pada bidangnya dalam penyelesaian masalah serta mampu beradaptasi terhadap situasi yang dihadapi
- Mampu mengambil keputusan yang tepat berdasarkan analisis informasi dan data, dan mampu memberikan petunjuk dalam memilih berbagai alternatif solusi secara mandiri dan kelompok.
- Bertanggung jawab pada pekerjaan sendiri dan dapat diberi tanggung jawab atas pencapaian hasil kerja organisasi

Level 8 (Magister)

- Mampu mengembangkan pengetahuan, teknologi, dan atau seni di dalam bidang keilmuannya atau praktek profesionalnya melalui riset, hingga menghasilkan karya inovatif dan teruji
- Mampu memecahkan permasalahan sains, teknologi, dan atau seni di dalam bidang keilmuannya melalui pendekatan inter atau multidisipliner
- Mampu mengelola riset dan pengembangan yang bermanfaat bagi masyarakat dan keilmuan, serta mampu mendapat pengakuan nasional maupun internasional

Level 9 (Doktor)

- Mampu mengembangkan pengetahuan, teknologi, dan atau seni baru di dalam bidang keilmuannya atau praktek profesionalnya melalui riset, hingga menghasilkan karya kreatif, original, dan teruji
- Mampu memecahkan permasalahan sains, teknologi, dan atau seni di dalam bidang keilmuannya melalui pendekatan inter, multi atau transdisipliner
- Mampu mengelola, memimpin, dan mengembangkan riset dan pengembangan yang bermanfaat bagi ilmu pengetahuan dan kemaslahatan umat manusia, serta mampu mendapat pengakuan nasional maupun internasional

Komparasi Penelitian D3/D4 vs S1 vs S2 vs S3

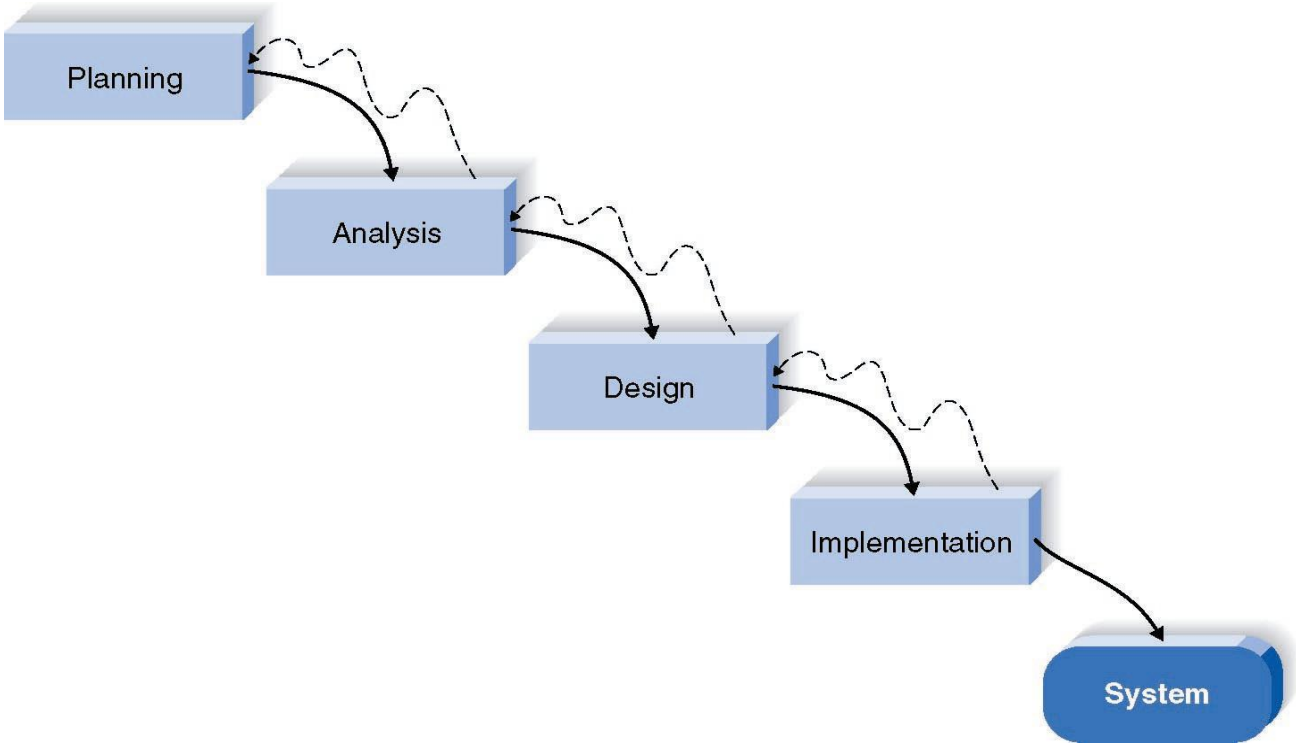
- D3/D4:
  - Pengembangan Sistem Informasi Rumah Sakit untuk Rumah Sakit "Suka Sembuh"
  - *Karakter: menguasai skill teknis (programming, networking, dsb)*
- S1:
  - Sistem Cerdas Berbasis Neural Network untuk Prediksi Harga Saham
  - *Karakter: menguji teori, dan terapkan dalam code (software)*
- S2/S3:
  - Penerapan Algoritma Genetika untuk Pemilihan Arsitektur Jaringan Secara Otomatis pada Neural Network untuk Prediksi Harga Saham
  - *Karakter: mengembangkan teori (perbaikan metode), ada kontribusi ke pengetahuan*

Kontribusi ke Pengetahuan vs Kontribusi ke Masyarakat

Kontribusi ke masyarakat tidak secara langsung bisa diukur, karena itu tidak dimasukkan ke tujuan penelitian, tapi ke manfaat penelitian

MITOS 3

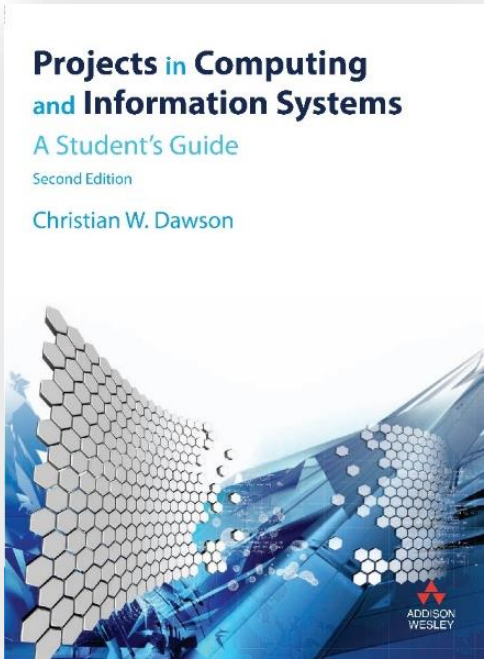
Waterfall adalah Metode Penelitian yang Saya Gunakan



Metode Penelitian

1. Penelitian Tindakan
   - Studi berupa monitoring dan pencatatan penerapan sesuatu oleh peneliti secara hati-hati, yang tujuannya untuk memecahkan masalah dan mengubah situasi *(Herbert, 1990)*
   - Penelitian Tindakan Kelas (PTK) di bidang Pendidikan
2. Eksperimen
   - Investigasi hubungan sebab akibat dengan menggunakan ujicoba yang dikontrol oleh peneliti
   - Melibatkan pengembangan dan evaluasi
   - Penelitian bidang Science dan Teknik
3. Studi Kasus
   - Eksplorasi satu situasi secara mendalam dan hati hati *(Cornford and Smithson, 2006)*
   - Penelitian bidang Sosial, Ekonomi, Politik
4. Survei
   - Pengumpulan data dari populasi yang bisa diukur, dengan cara yang ekonomis *(Saunders et al., 2007)*
   - Melibatkan penggunaan kuesioner dan interview

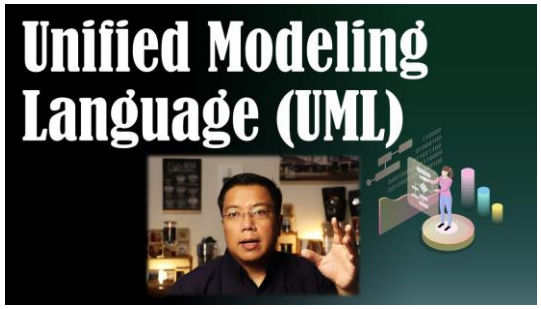
(Dawson, 2009)



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

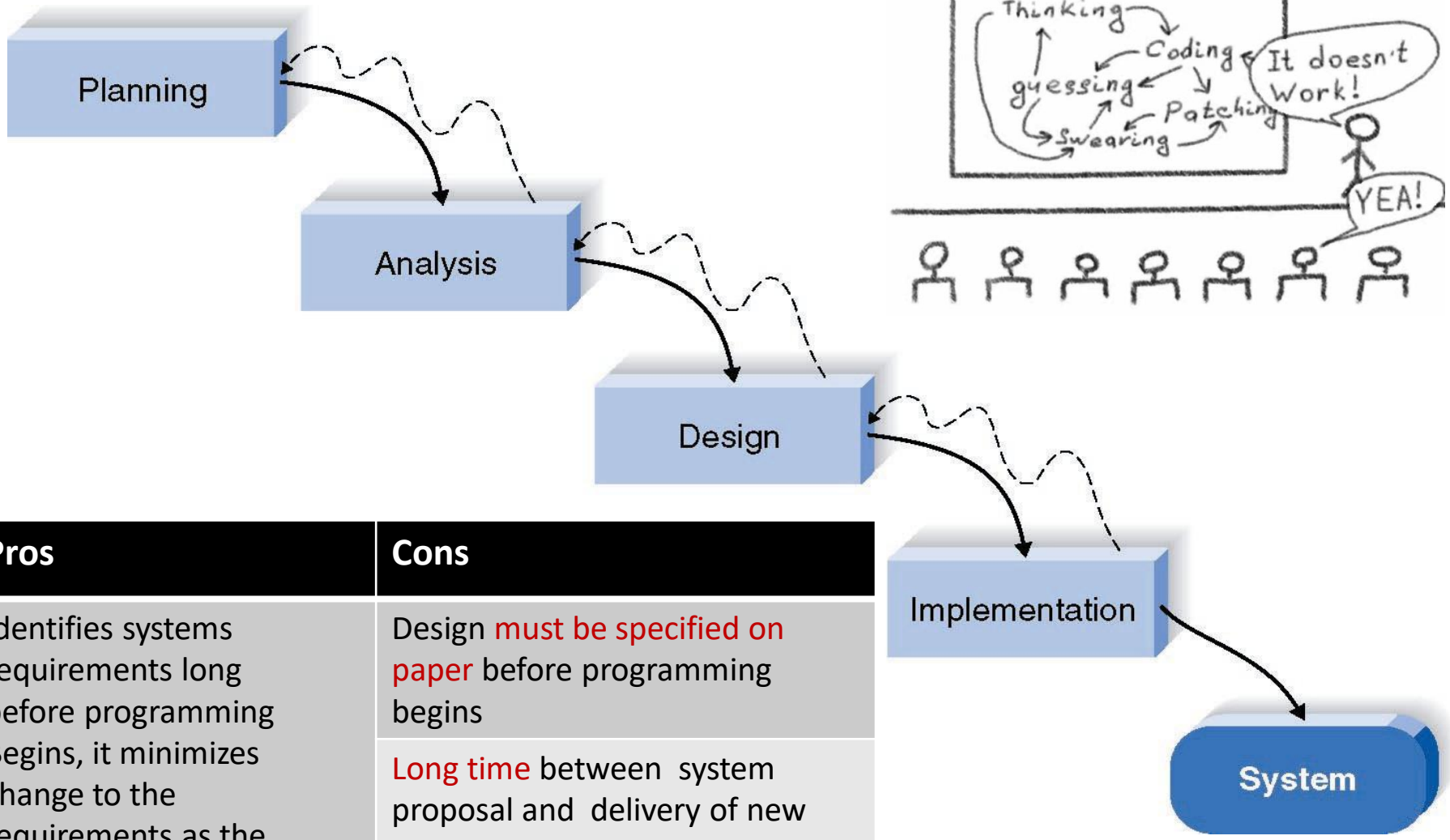
(Dennis, 2012)



More Prescriptive/ Documentation

More Adaptive/ Communication

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 mendukung
1. Fitur Pencarian Prod
2. Fitur Pencarian Toko
3. Fitur Pemesanan Pr
4. Fit

Busi
Intan
Tang

Studi Kelayakan Sistem Penjualan Musik Online

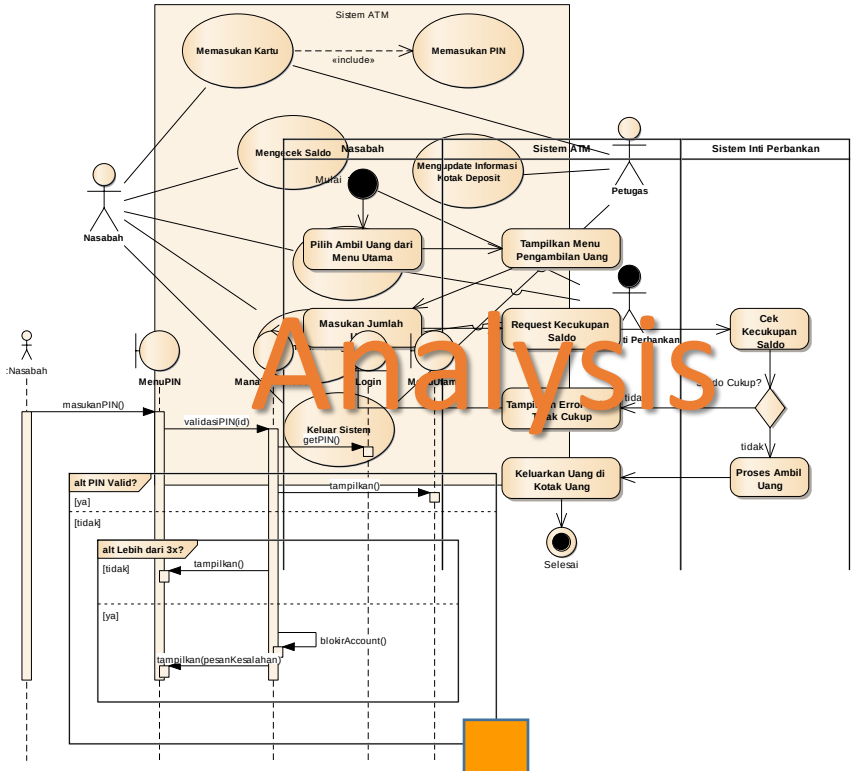
Margaret Mooney dan Alec Adams membuat studi kelayakan untuk pengembangan Sistem Penjualan Musik Online

Kelayakan Teknis

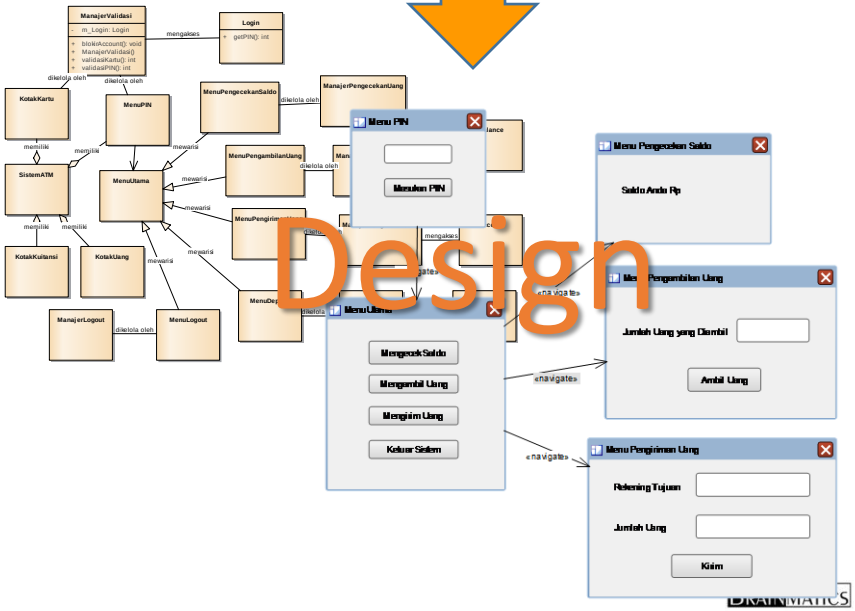
Sistem penjualan musik online layak secara teknis, meskipun memiliki beberapa risiko.

	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 Requ	8,95 084	091
PV of All Benefits:	(Present Value)	9 8,95 084	070,501,1 2
Honor Tim (Planning, Analysis, Design dan Implementation)	360,0 ,000	0	
Honor Konsultan Infrastruktur Internet	90,00 ,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**	**569,597,722**	**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**

Analysis

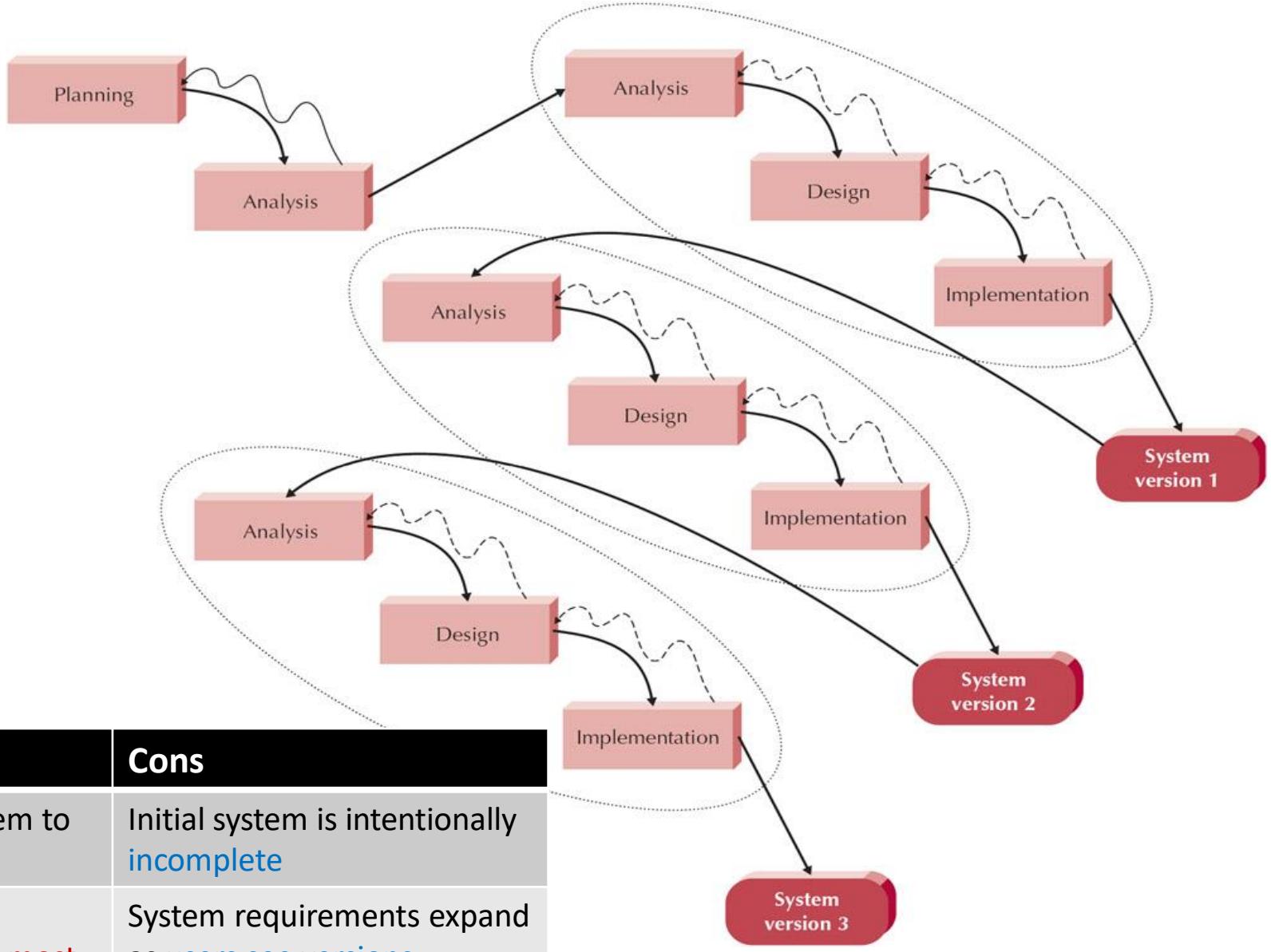


Design



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

Business Requirement

Sistem yang mendukung

1. Fitur Pencarian Produk
2. Fitur Pencarian Toko
3. Fitur Pemesanan Produk
4. Fit

Busi

Intan

Tang

Studi Kelayakan Sistem Penjualan Musik Online

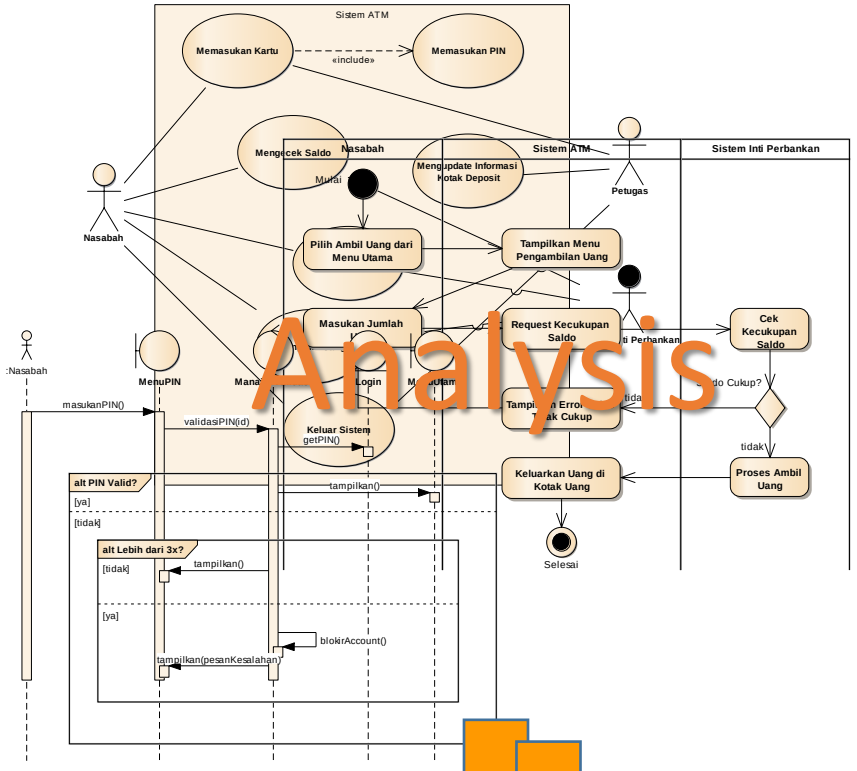
Margaret Mooney dan Alec Adams membuat studi kelayakan untuk pengembangan Sistem Penjualan Musik Online

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 Reque	8,95 084	091 ,3
PV of All Benefits:	(mess alue)	9 8,95 084	070,601,1 2
Honor Tim (Planning, Analysis, Design and Implementation)	360,0 ,000	0	
Honor Konsultan Infrastruktur Internet	90,00 ,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**	**569,597,722**	**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**

Planning

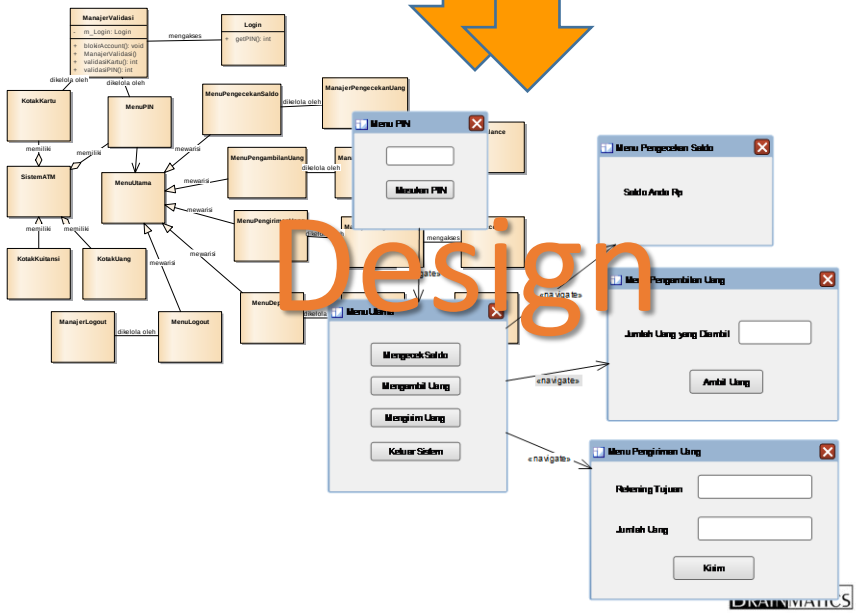


Analysis

Versi 1

Versi 2

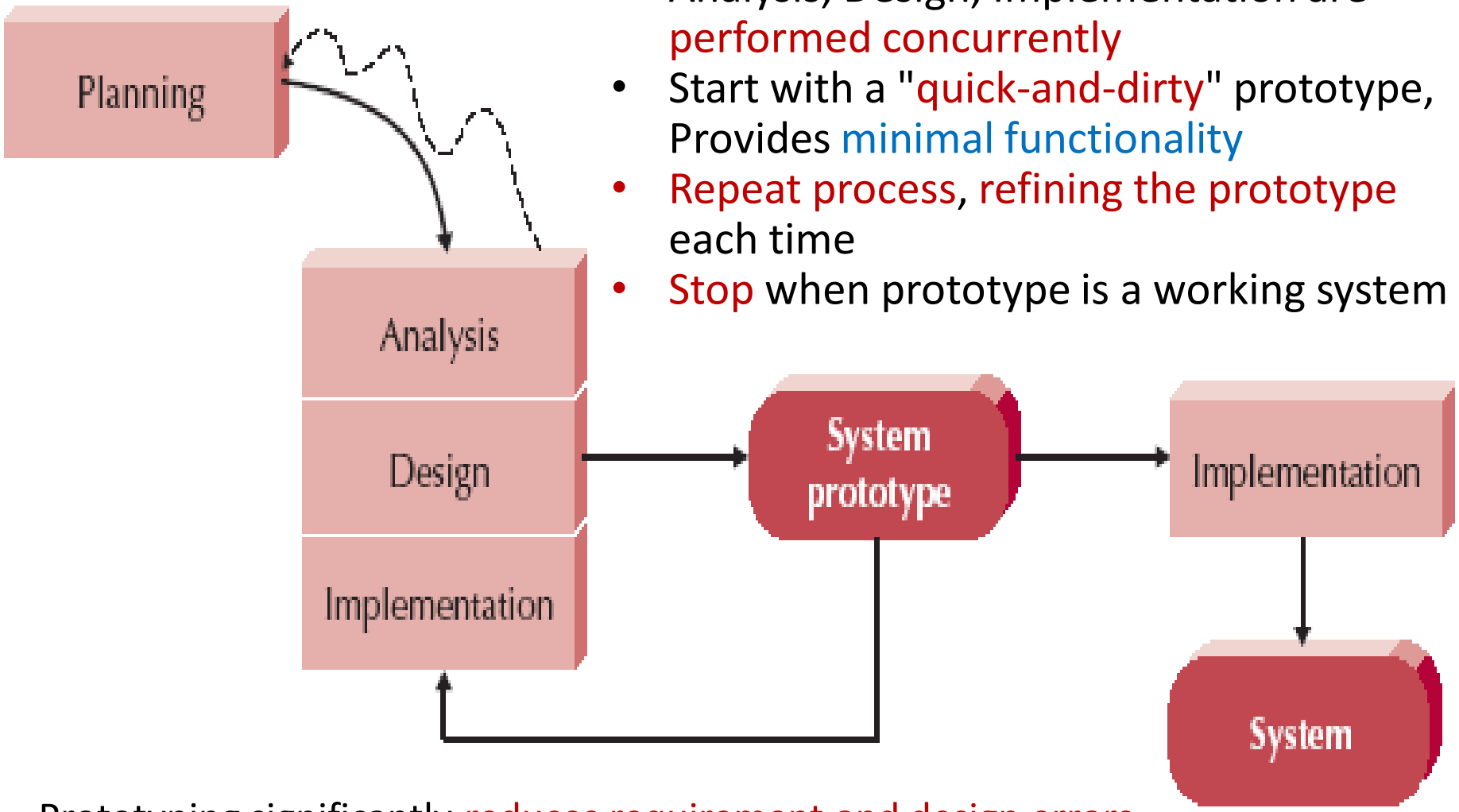
Implementation



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

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 dan 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**	**569,597,722**	**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**

Planning

Analysis

Design

Implementation

3. Agile Development

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

- Mempercepat proses SDLC
  - Mengurangi pemodelan dan dokumentasi
  - Mengembangakan software dengan simple dan iteratif

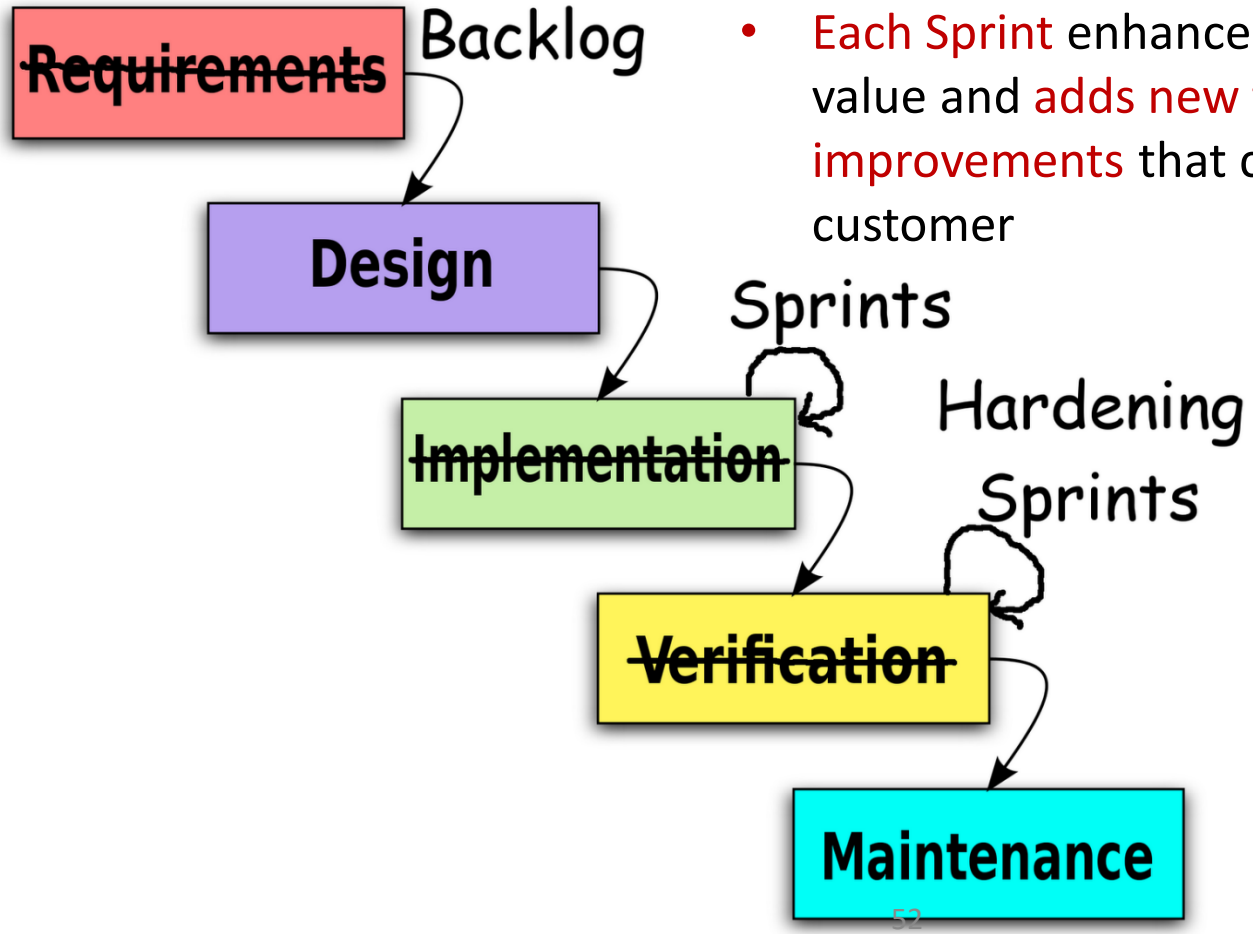
- Agile Approach:
  1. Agile Values
  2. Agile Principles
  3. Agile Practices

3. Agile: Extreme Programming

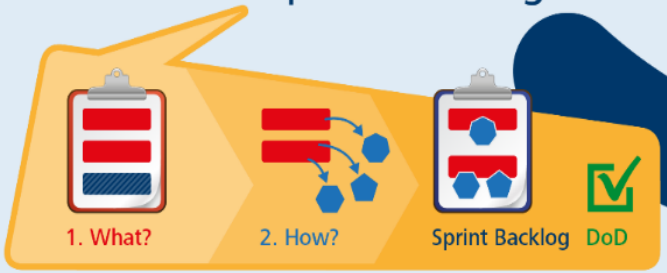
1. Communication: Building software requires communicating requirements to the developers
   1. Pair Programming
   2. Communication replace documentation
2. Simplicity: Encourages starting with the simplest solution, extra functionality can then be added later
3. Feedback:
   1. Feedback from the system: 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

3. Agile: 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:	System Reque	8,95,084	1,091,[illegible]
PV of All Benefits:	[illegible]	9[illegible]8,95,084	1,070,601,1[illegible]2
Honor Tim (Planning, Analysis, Design dan Implementation)	360,0[illegible],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**

Planning

Sprint 1

Analysis

Sprint 2

Hasil Sprint 1

Hasil Sprint 2

Implementation

Design

Metodologi Penelitian vs Metodologi Pengembangan Software

- Metodologi pengembangan software bukan metodologi penelitian!

- Jangan pernah masukan waterfall/RAD/Agile di Bab 3 skripsi/tesis/disertasi sebagai metode penelitian!
  - Bab 3 seharusnya membahas tentang bagaimana metode, teknik, cara, tahapan dan evaluasi pada penelitian kita (metodologi penelitian)

MITOS 4

Masalah Penelitian itu adalah Masalah Yang Muncul di Masyarakat

Konsepsi Masalah Penelitian

- Penelitian dilakukan karena ada masalah penelitian

- Dimana masalah penelitian sendiri muncul karena ada latar belakang masalah penelitian

- Latar belakang masalah penelitian itu berangkatnya bisa dari masalah kehidupan (obyek penelitian)

Contoh Alur Latar Belakang Masalah Penelitian:

Penerapan XYZ untuk Masalah E pada SVM untuk Prediksi Nilai Tukar Uang

- Nilai tukar uang adalah faktor penting pada perekonomian suatu negara. Nilai tukar uang perlu diprediksi supaya kebijakan perekonomian bisa diambil dengan lebih akurat dan efisien…
- Metode untuk prediksi nilai tukar yang saat ini digunakan adalah regresi linier, neural network dan support vector machine…
- Regresi linier memiliki kelebihan A dan kelemahan B…
- Neural network memiliki kelebihan C dan kelemahan D…
- Support vector machine memiliki kelebihan bisa mengatasi masalah B (pada regresi linier) dan D (pada neural network)… tapi memiliki kelemahan E

- Masalah penelitian pada penelitian di atas?
  - Kebijakan perekonomian negara?
  - Prediksi nilai tukar yang?
  - Metode apa yang seharusnya dipakai untuk prediksi nilai tukar?

- Masalah: Support vector machine memiliki kelebihan memecahkan masalah B dan D (argumentasi dipilih), tapi memiliki kelemahan E
- Tujuan: Menerapkan metode XYZ untuk memecahkan masalah E pada support vector machine

Contoh Alur Latar Belakang Masalah Penelitian:

Penerapan XYZ untuk E pada Fuzzy Logic untuk Pengaturan Lampu Lalu Lintas Dinamis

- Kemacetan lalu lintas di kota besar semakin meningkat
- Penyebab kemacetan adalah traffic light persimpangan jalan
- Traffic light yang ada adalah statis (tetap waktunya) sehingga tidak dapat menyelesaikan kondisi kepadatan kendaraan yang di berbagai waktu
- Traffic light harus didesain dinamis sesuai perubahan berbagai parameter
- Metode untuk menentukan waktu yang tepat secara dinamis dapat menggunakan AHP, ANP, Fuzzy Logic,
- AHP memiliki kelebihan A dan kelemahan B…
- ANP memiliki kelebihan C dan kelemahan D…
- Fuzzy logic memiliki kelebihan bisa mengatasi masalah B (pada AHP) dan D (pada ANP)… tapi memiliki kelemahan E

- Masalah penelitian pada penelitian di atas?
  - Bagaimana mengata[X]emacetan lalu lintas?
  - Bagaimana mendes[X]raffic light?
  - Metode apa yang sebaiknya dipakai untuk penentuan traffic light secara dinamis?
- Masalah: Fuzzy logic memiliki kelebihan memecahkan masalah B dan D (argumentasi dipilih), tapi memiliki kelemahan E
- Tujuan: Menerapkan metode XYZ untuk memecahkan masalah E pada fuzzy logic

Contoh Masalah Penelitian

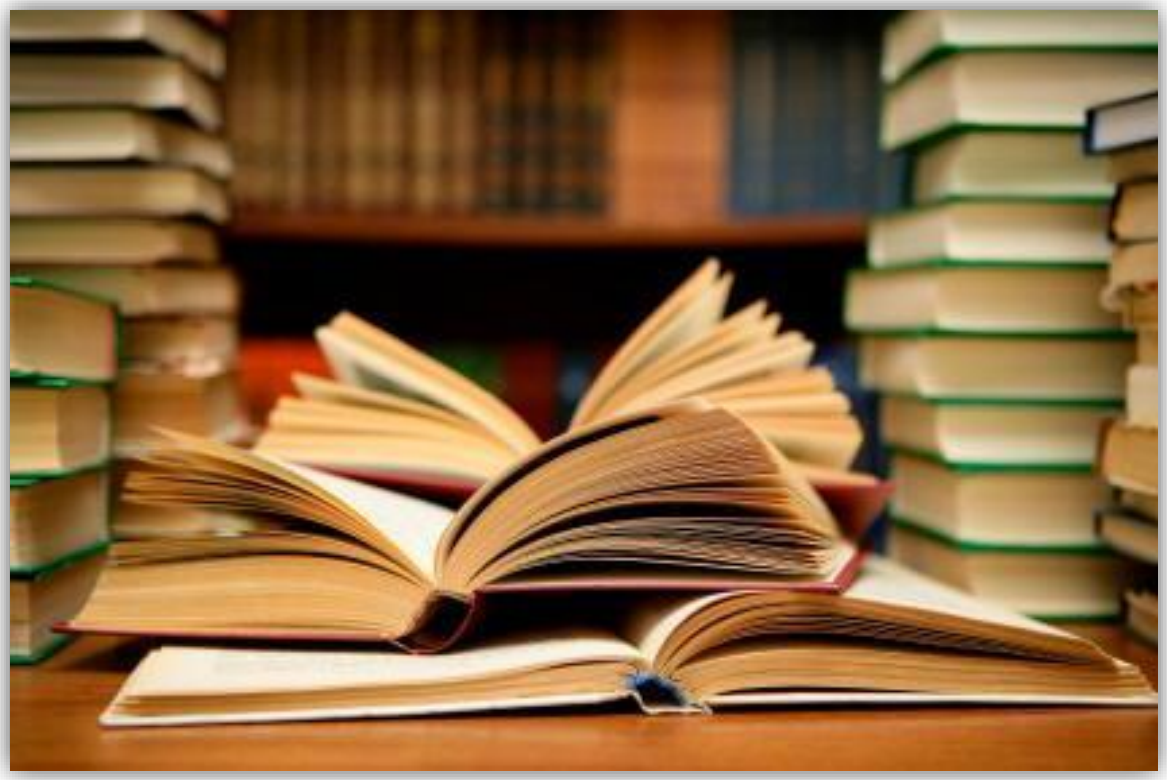
- ***Ungu**: Obyek Data (Opsional, Bisa Data Publik)*
- ***Oranye**: Topik (Obyek Metode yang Diperbaiki)*
- ***Merah**: Masalah Penelitian*
- ***Hijau**: Metode Perbaikan yang Diusulkan*
- ***Biru**: Pengukuran Penelitian (Tidak Harus Masuk Judul)*

Penerapan Particle Swarm Optimization untuk Pemilihan Parameter Secara Otomatis pada Support Vector Machine untuk Prediksi Produksi Padi

Research Problem (RP)	Research Question (RQ)	Research Objective (RO)
SVM dapat memecahkan masalah 'over-fitting', lambatnya konvergensi, dan sedikitnya data training, akan tetapi memiliki kelemahan pada sulitnya pemilihan parameter SVM yang sesuai yang mengakibatkan akurasi tidak stabil	Seberapa meningkat akurasi metode SVM apabila PSO diterapkan pada proses pemilihan parameter?	Menerapkan PSO untuk pemilihan parameter yang sesuai pada SVM (C, lambda dan epsilon) , sehingga hasil prediksinya lebih akurat

MITOS 5

Studi Literatur Berisi Berbagai Teori Dasar dan Definisi yang Ada di Buku



Manfaat Studi Literatur

- Memperdalam pengetahuan tentang bidang yang diteliti (*Textbooks*)
- Mengetahui hasil penelitian yang berhubungan dan yang sudah pernah dilaksanakan (Related Research) (*Paper*)
- Mengetahui perkembangan ilmu pada bidang yang kita pilih (state-of-the-art) (*Paper*)
- Mencari dan memperjelas masalah penelitian (*Paper*)

Studi Literatur adalah bingkai dari siklus penelitian (kita lakukan pada setiap tahapan penelitian)

Tahapan Penelitian Computing

Literature Review

1. Penentuan Bidang Penelitian (***Research Field***)
2. Penentuan Topik Penelitian (***Research Topic***)
3. Penentuan Masalah Penelitian (***Research Problem***)
4. Perangkuman Metode-Metode Yang Ada (***State-of-the-Art Methods***)
5. Penentuan Metode Yang Diusulkan (***Proposed Method***)
6. Evaluasi Metode Yang Diusulkan (***Evaluation***)
7. Penulisan Ilmiah dan Publikasi Hasil Penelitian (***Publications***)

**https://www.site.uottawa.ca/~bochmann/dsrg/how-to-do-good-research/*
**http://romisatriawahono.net/2013/01/23/tahapan-memulai-penelitian-untuk-mahasiswa-galau/*

Literature Review

- Literature Review is a critical and in-depth evaluation of previous research (Shuttleworth, 2009) (*https://explorable.com/what-is-a-literature-review*)
- A summary and synopsis of a particular area of research, allowing anybody reading the paper to establish the reasons for pursuing a particular research
- A good Literature Review evaluates quality and findings of previous research (State-of-the-Art Methods)

Jenis Literatur Ilmiah

1. **Paper dari Journal**
2. Paper dari Book Chapter
3. Paper dari Conference (Proceedings)
4. Thesis dan Disertasi
5. Report (Laporan) dari Organisasi yang Terpercaya
6. Buku Textbook

** Prioritaskan mengambil paper journal yang terindeks oleh ISI dan SCOPUS, cek dengan http://scimagojr.com*

Literature Review Methods

- Types and Methods of Literature Review:
  1. Traditional Review
  2. Systematic Literature Review or Systematic Review
  3. Systematic Mapping Study (Scoping Study)
  4. Tertiary Study

- SLR is now well established review method in the field of software engineering

(Kitchenham & Charters, Guidelines in performing Systematic Literature Reviews in Software Engineering, EBSE Technical Report version 2.3, 2007)

Contoh dan Studi Kasus SLR

Romi Satria Wahono, A Systematic Literature Review of Software Defect Prediction: Research Trends, Datasets, Methods and Frameworks, Journal of Software Engineering, Vol. 1, No. 1, pp. 1-16, April 2015

https://romisatriawahono.net/2016/05/15/systematic-literature-review-pengantar-tahapan-dan-studi-kasus/

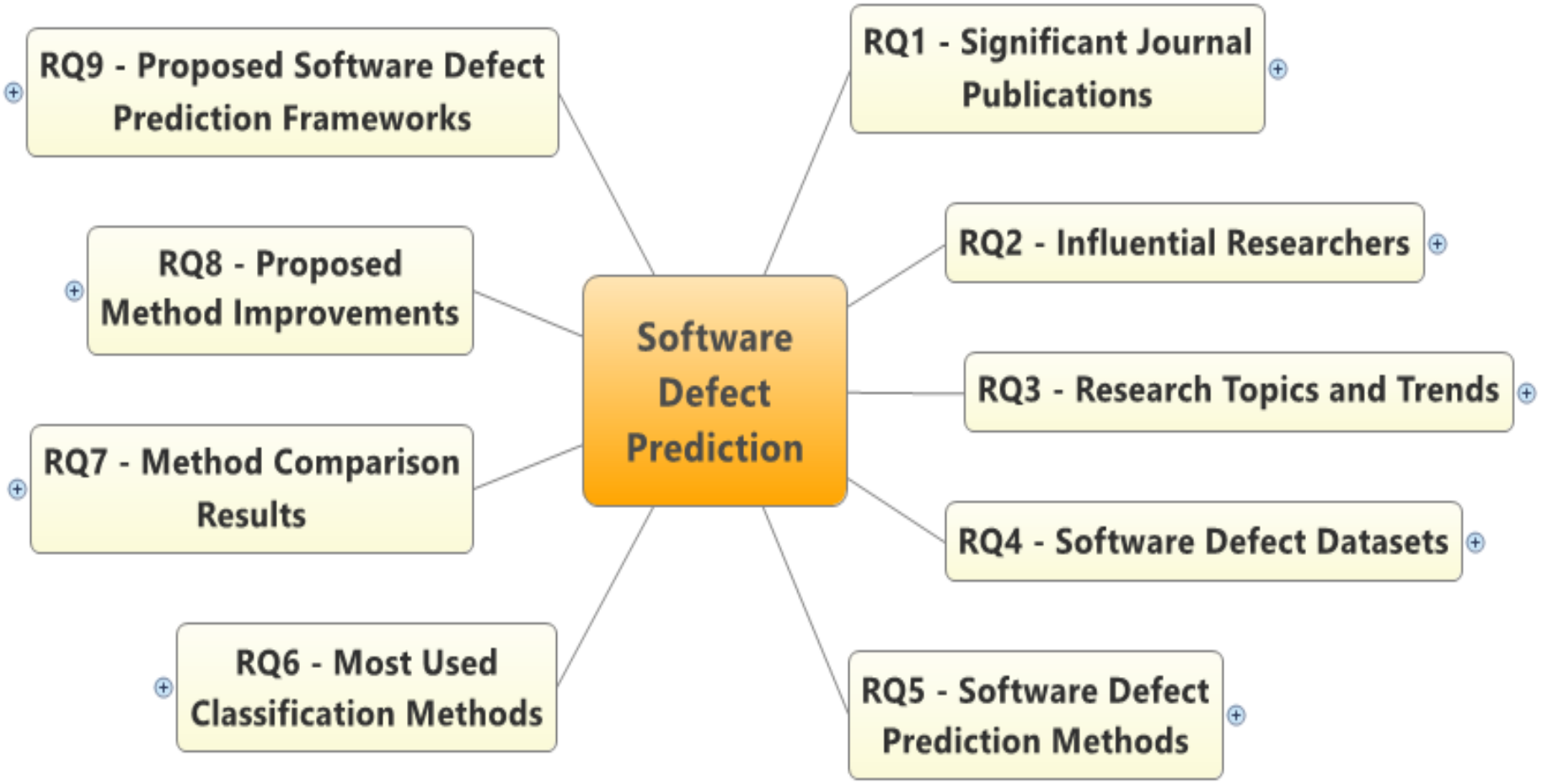
PICOC

Population	Software, software application, software system, information system
Intervention	Software defect prediction, fault prediction, error-prone, detection, classification, estimation, models, methods, techniques, datasets
Comparison	n/a
Outcomes	Prediction accuracy of software defect, successful defect prediction methods
Context	Studies in industry and academia, small and large data sets

Research Question (RQ)

ID	Research Question
RQ1	Which journal is the most significant software defect prediction journal?
RQ2	Who are the most active and influential researchers in the software defect prediction field?
RQ3	What kind of research topics are selected by researchers in the software defect prediction field?
RQ4	What kind of datasets are the most used for software defect prediction?
RQ5	What kind of methods are used for software defect prediction?
RQ6	What kind of methods are used most often for software defect prediction?
RQ7	Which method performs best when used for software defect prediction?
RQ8	What kind of method improvements are proposed for software defect prediction?
RQ9	What kind of frameworks are proposed for software defect prediction?

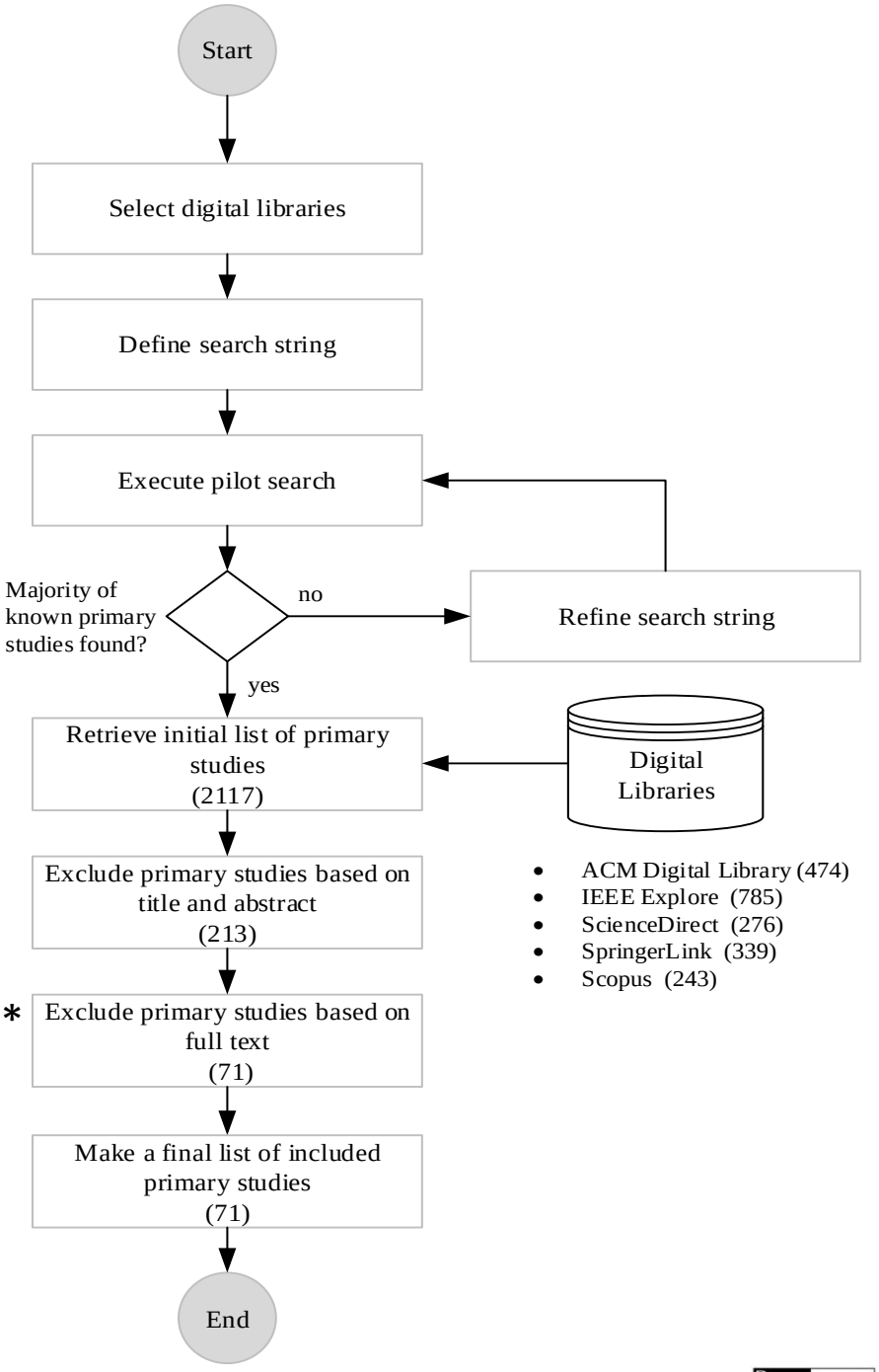
Research Question (RQ)



Studies Selection Strategy

- Publication Year:
  - ✓2000-2013
- Publication Type:
  - ✓Journal
  - ✓Conference Proceedings
- Search String:

  software
  AND
  (fault* OR defect* OR quality OR error-prone)
  AND
  (predict* OR prone* OR probability OR assess* OR detect* OR estimat* OR classificat*)
- Selected Studies:
  - ✓71



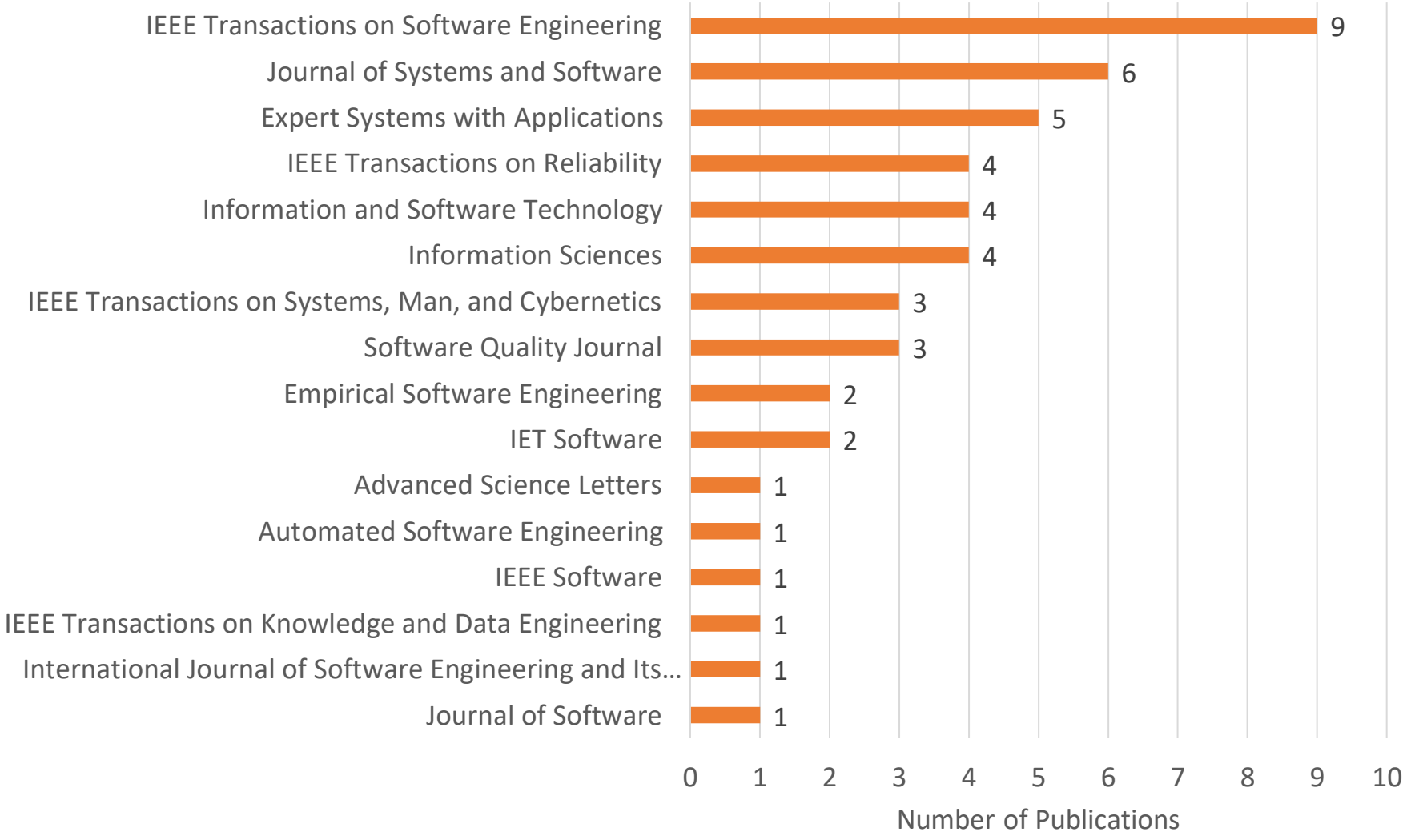
Inclusion and Exclusion Criteria

Inclusion Criteria	Studies in academic and industry using large and small scale data sets
	Studies discussing and comparing modeling performance in the area of software defect prediction
	For studies that have both the conference and journal versions, only the journal version will be included
	For duplicate publications of the same study, only the most complete and newest one will be included
Exclusion Criteria	Studies without a strong validation or including experimental results of software defect prediction
	Studies discussing defect prediction datasets, methods, frameworks in a context other than software defect prediction
	Studies not written in English

Result

Romi Satria Wahono, A Systematic Literature Review of Software Defect Prediction: Research Trends, Datasets, Methods and Frameworks, Journal of Software Engineering, Vol. 1, No. 1, pp. 1-16, April 2015

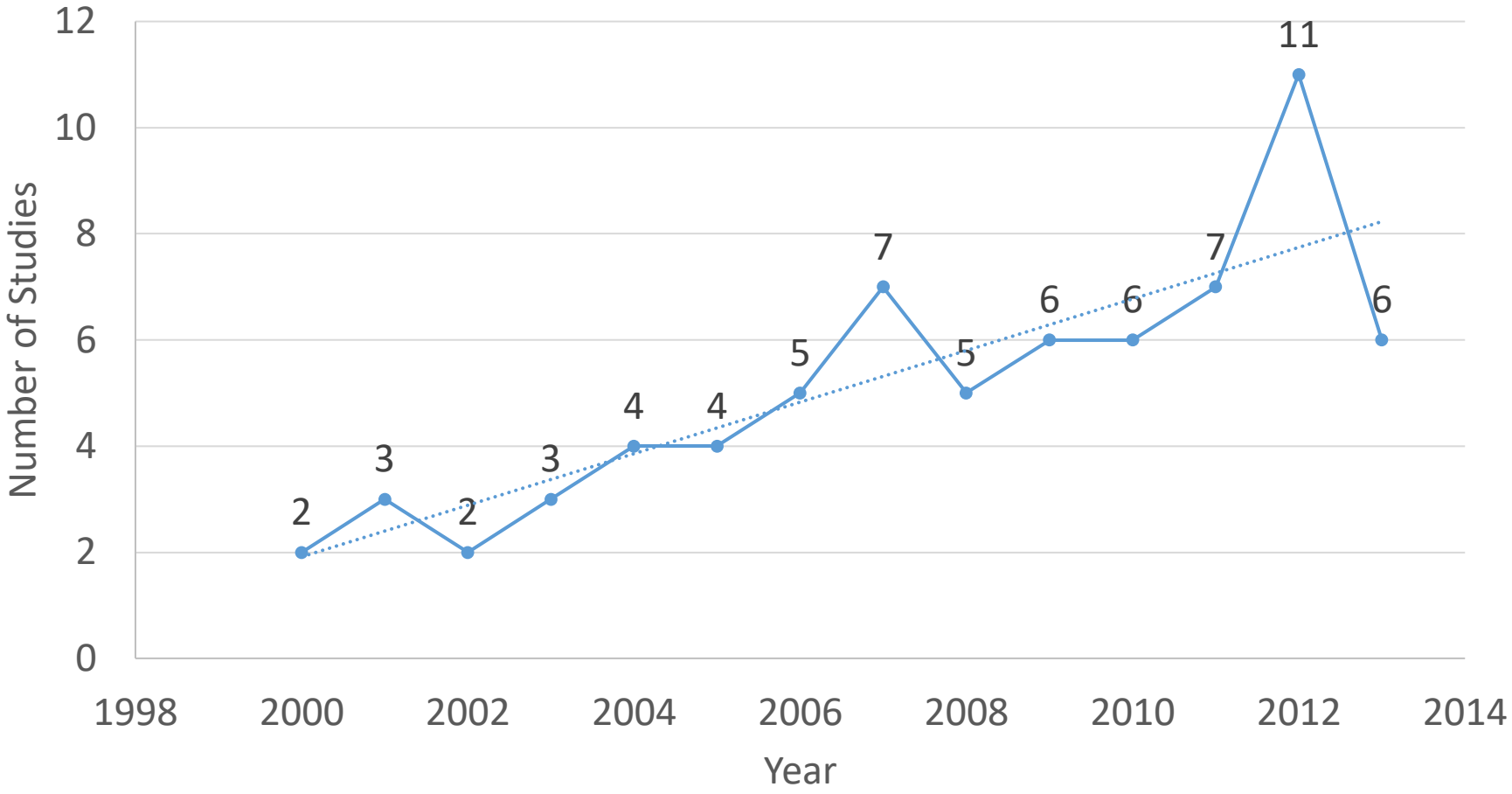
RQ1: Significant Journal Publications



Journal Quality Level of Selected Studies

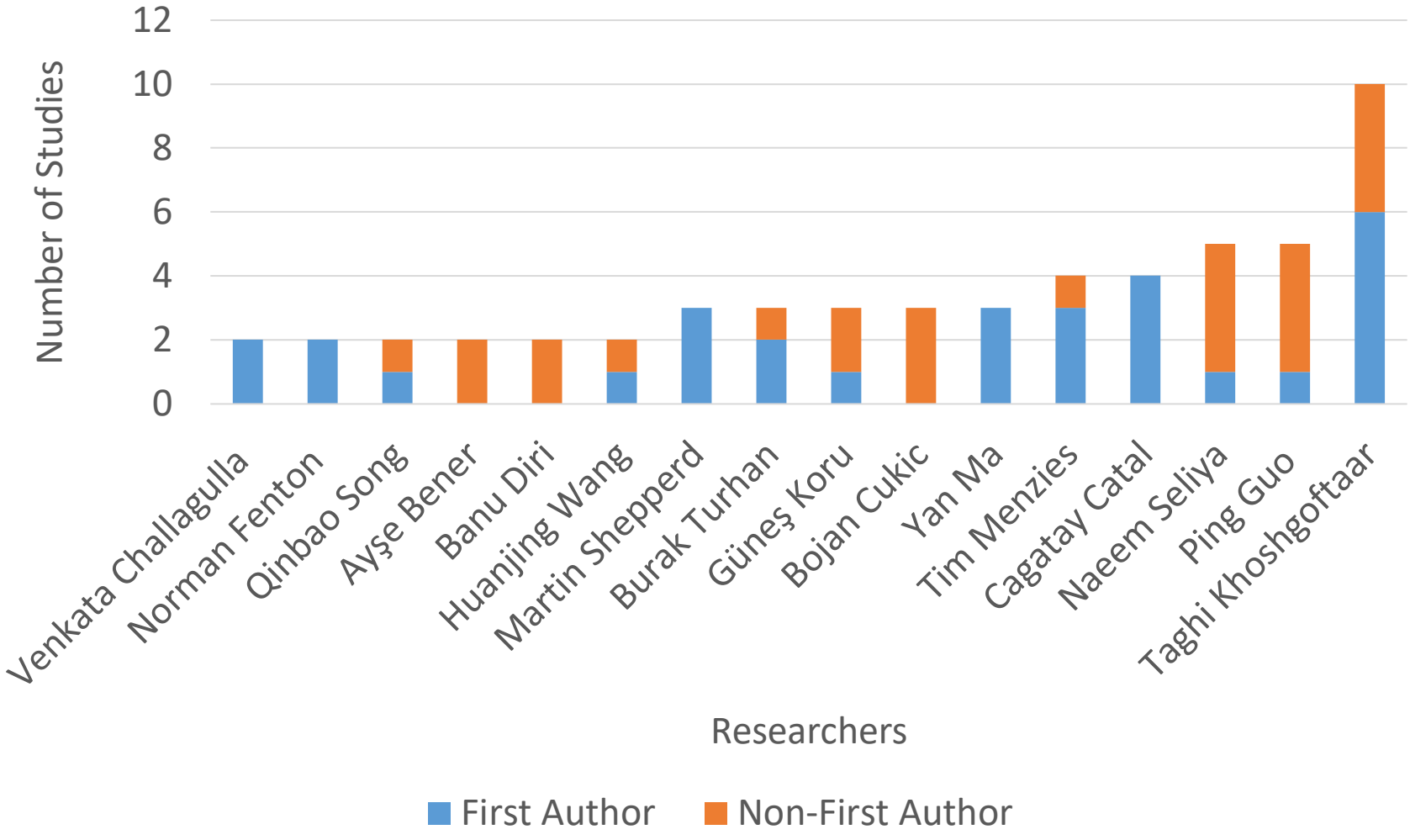
No	Journal Publications	SJR	Q Category
1	IEEE Transactions on Software Engineering	3.39	Q1 in Software
2	Information Sciences	2.96	Q1 in Information Systems
3	IEEE Transactions on Systems, Man, and Cybernetics	2.76	Q1 in Artificial Intelligence
4	IEEE Transactions on Knowledge and Data Engineering	2.68	Q1 in Information Systems
5	Empirical Software Engineering	2.32	Q1 in Software
6	Information and Software Technology	1.95	Q1 in Information Systems
7	Automated Software Engineering	1.78	Q1 in Software
8	IEEE Transactions on Reliability	1.43	Q1 in Software
9	Expert Systems with Applications	1.36	Q2 in Computer Science
10	Journal of Systems and Software	1.09	Q2 in Software
11	Software Quality Journal	0.83	Q2 in Software
12	IET Software	0.55	Q2 in Software
13	Advanced Science Letters	0.24	Q3 in Computer Science
14	Journal of Software	0.23	Q3 in Software
15	International Journal of Software Engineering and Its Application	0.14	Q4 in Software

Distribution of Selected Studies by Year



- The interest in software defect prediction has changed over time
- Software defect prediction research is still very much relevant to this day

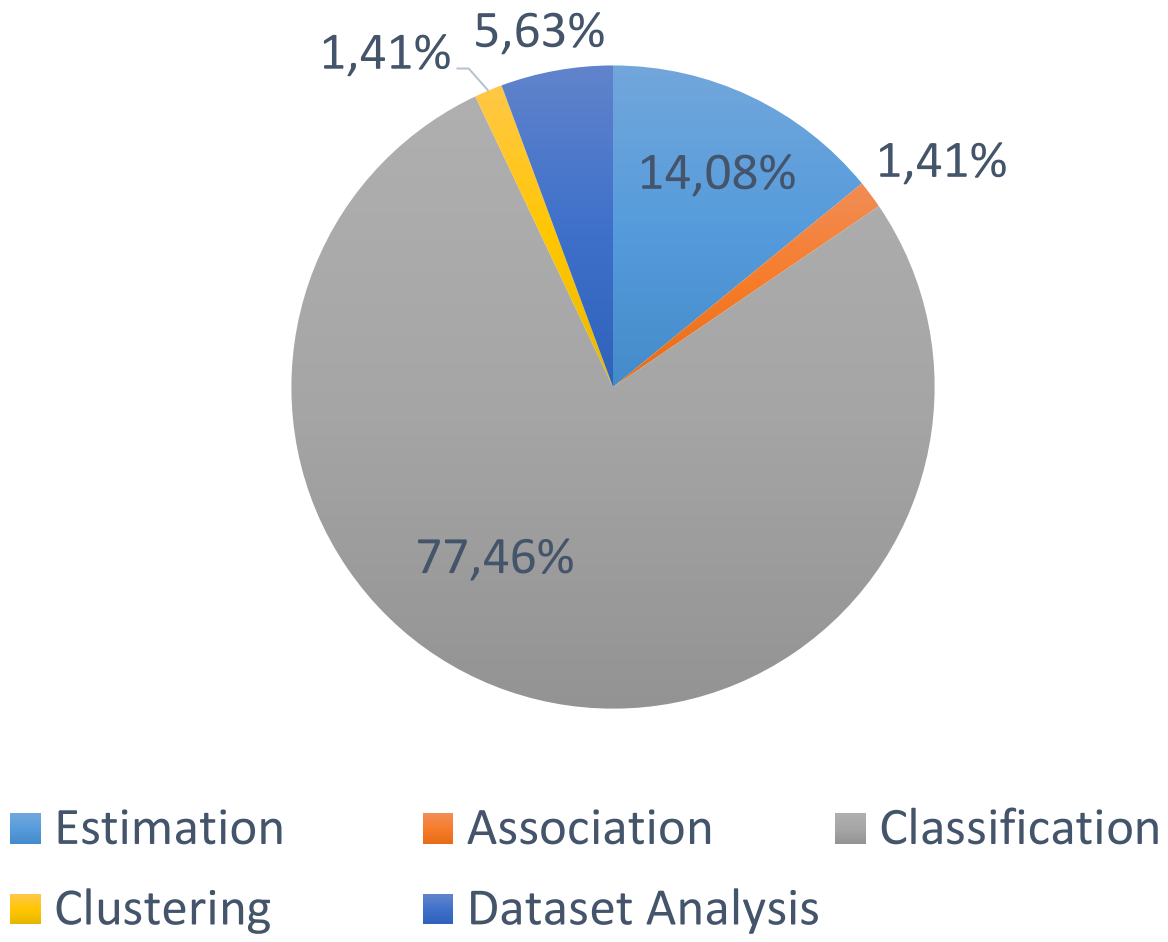
RQ2: Influential Researchers



RQ3: Research Topics and Trends

1. Estimating the number of defects remaining in software systems using estimation algorithm (Estimation)
2. Discovering defect associations using association rule algorithm (Association)
3. Classifying the defect-proneness of software modules, typically into two classes, defect-prone and not defect-prone, using classification algorithm (Classification)
4. Clustering the software defect based on object using clustering algorithm (Clustering)
5. Analyzing and pre-processing the software defect datasets (Dataset Analysis)

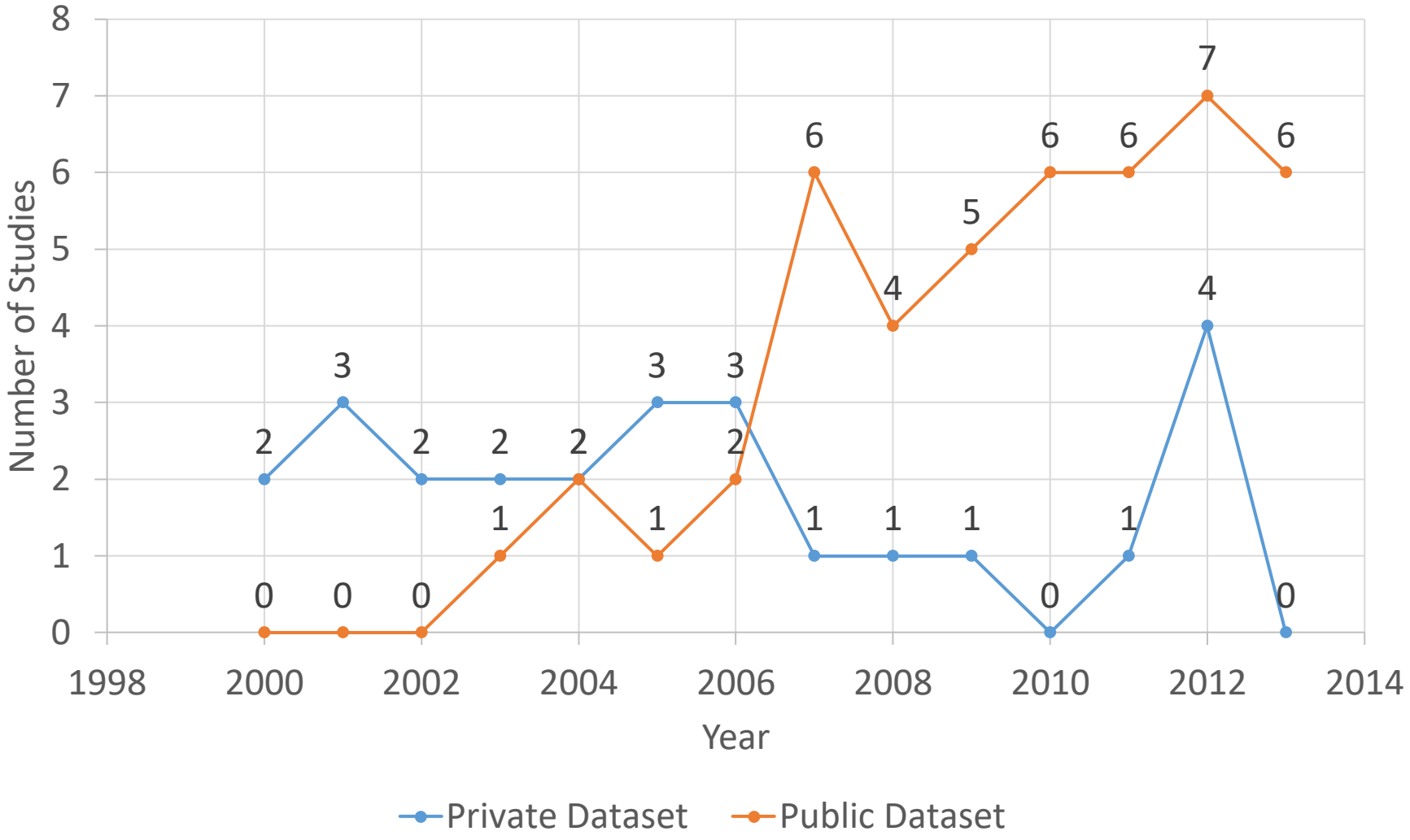
Distribution of Research Topics and Trends



Example Distribution of Research Topics and Trends

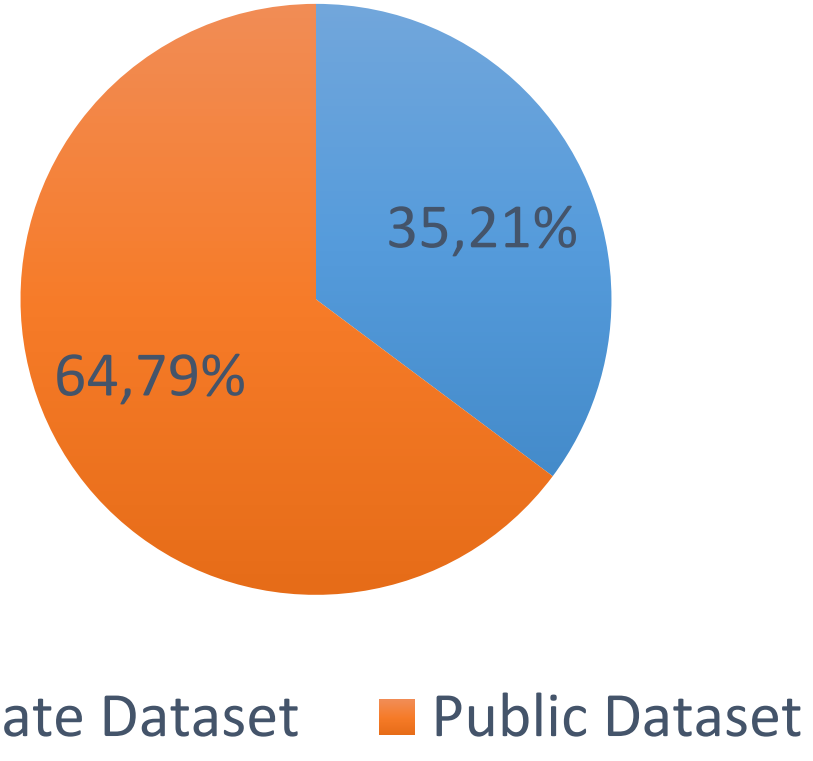
Year	Primary Studies	Publications	Datasets	Topics
2008	(Lessmann et al., 2008)	IEEE Transactions on Software Engineering	Public	Classification
	(Bibi et al., 2008)	Expert Systems with Applications	Private	Estimation
	(Gondra, 2008)	Journal of Systems and Software	Public	Classification
	(Vandecruys et al., 2008)	Journal of Systems and Software	Public	Classification
	(Elish and Elish 2008)	Journal of Systems and Software	Public	Classification
2012	(Gray et al., 2012)	IET Software	Public	Dataset Analysis
	(Ying Ma, Luo, Zeng, & Chen, 2012)	Information and Software Technology	Public	Classification
	(Benaddy and Wakrim 2012)	International Journal of Software Engineering	Private	Estimation
	(Y. Peng, Wang, & Wang, 2012)	Information Sciences	Public	Classification
	(Zhang and Chang 2012)	International Conference on Natural Computation	Private	Estimation
	(Bishnu and Bhattacherjee 2012)	IEEE Transactions on Knowledge and Data Engineering	Private	Clustering
	(Sun, Song, & Zhu, 2012)	IEEE Transactions on Systems, Man, and Cybernetics	Public	Classification
	(Pelayo and Dick 2012)	IEEE Transactions on Reliability	Public	Classification
	(Jin, Jin, & Ye, 2012)	IET Software	Public	Classification
	(Cao, Qin, & Feng, 2012)	Advanced Science Letters	Public	Classification
2013	(Park et al., 2013)	Information Sciences	Public	Classification
	(Dejaeger, Verbraken, & Baesens, 2013)	IEEE Transactions on Software Engineering	Public	Classification
	(Shepperd, Song, Sun, & Mair, 2013)	IEEE Transactions on Software Engineering	Public	Dataset Analysis
	(Wang and Yao 2013)	IEEE Transactions on Reliability	Public	Classification
	(Peters, Menzies, Gong, & Zhang, 2013)	IEEE Transactions on Software Engineering	Public	Dataset Analysis
	(Radjenović et al., 2013)	Information and Software Technology	Public	Dataset Analysis

RQ4: Software Defect Datasets



Distribution of Software Defect Datasets

- The use of public data sets makes the research repeatable, refutable, and verifiable (Catal & Diri 2009a)
- Since 2005 more public datasets were used
- NASA MDP repository have been developed in 2005 and researchers started to be aware regarding the use of public datasets



NASA MDP Dataset

Dataset	Project Description	Language	Number of Modules	Number of *fp* Modules	Faulty Percentage
CM1	Spacecraft instrument	C	505	48	12.21%
KC1	Storage management for ground data	C++	1571	319	15.51%
KC3	Storage management for ground data	Java	458	42	18%
MC2	Video guidance system	C	127	44	34.65%
MW1	Zero gravity experiment related to combustion	C	403	31	10.23%
PC1	Flight software from an earth orbiting satellite	C	1059	76	8.04%
PC2	Dynamic simulator for attitude control systems	C	4505	23	1.01%
PC3	Flight software for earth orbiting satellite	C	1511	160	12.44%
PC4	Flight software for earth orbiting satellite	C	1347	178	12.72%

Code Attributes		Symbols	Description
LOC counts	LOC_total		The total number of lines for a given module
	LOC_blank		The number of blank lines in a module
	LOC_code_and_comment	NCSLOC	The number of lines which contain both code and comment in a module
	LOC_comments		The number of lines of comments in a module
	LOC_executable		The number of lines of executable code for a module
	number_of_lines		Number of lines in a module
Halstead	content	μ	The halstead length content of a module $\mu = \mu_1 + \mu_2$
	difficulty	D	The halstead difficulty metric of a module D = 1/L
	effort	E	The halstead effort metric of a module E = V/L
	error_est	B	The halstead error estimate metric of a module $B = E^{2/3}/1000$
	length	N	The halstead length metric of a module $N = N_1+N_2$
	level	L	The halstead level metric of a module $L = (2^* \mu_2)/ \mu_1{}^*N_2$
	prog_time	T	The halstead programming time metric of a module T = E/18
	volume	V	The halstead volume metric of a module $V = N^*\log_2(\mu_1+ \mu_2)$
	num_operands	N_1	The number of operands contained in a module
	num_operators	N_2	The number of operators contained in a module
	num_unique_operands	μ_1	The number of unique operands contained in a module
	num_unique_operators	μ_2	The number of unique operators contained in a module
McCabe	cyclomatic_complexity	v(G)	The cyclomatic complexity of a module v(G) = e – n +2
	cyclomatic_density		v(G) / NCSLOC
	design_complexity	iv(G)	The design complexity of a module
	essential_complexity	ev(G)	The essential complexity of a module
Misc.	branch_count		Branch count metrics
	call_pairs		Number of calls to functions in a module
	condition_count		Number of conditionals in a given module
	decision_count		Number of decision points in a module
	decision_density		condition_count / decision_count
	edge_count		Number of edges found in a given module from one module to another
	essential_density		Essential density is calculated as: (ev(G)-1)/(v(G)-1)
	parameter_count		Number of parameters to a given module
	maintenance_severity		Maintenance Severity is calculated as: ev(G)/v(G)
	modified_condition_count		The effect of a condition affect a decision outcome by varying that condition only
	multiple_condition_count		Number of multiple conditions within a module
	global_data_complexity	gdv(G)	the ratio of cyclomatic complexity of a module's structure to its parameter count
	global_data_density		Global Data density is calculated as: gdv(G)/v(G)
	normalized_cyclo_cmplx		v(G) / numbe_of_lines
	percent_comments		Percentage of the code that is comments
	node_count		Number of nodes found in a given module

Code Attributes		NASA MDP Dataset								
		CM1	KC1	KC3	MC2	MW1	PC1	PC2	PC3	PC4
LOC counts	LOC_total	√	√	√	√	√	√	√	√	√
	LOC_blank	√	√	√	√	√	√		√	√
	LOC_code_and_comment	√	√	√	√	√	√	√	√	√
	LOC_comments	√	√	√	√	√	√	√	√	√
	LOC_executable	√	√	√	√	√	√	√	√	√
	number_of_lines	√		√	√	√	√	√	√	√
Halstead	content	√	√	√	√	√	√	√	√	√
	difficulty	√	√	√	√	√	√	√	√	√
	effort	√	√	√	√	√	√	√	√	√
	error_est	√	√	√	√	√	√	√	√	√
	length	√	√	√	√	√	√	√	√	√
	level	√	√	√	√	√	√	√	√	√
	prog_time	√	√	√	√	√	√	√	√	√
	volume	√	√	√	√	√	√	√	√	√
	num_operands	√	√	√	√	√	√	√	√	√
	num_operators	√	√	√	√	√	√	√	√	√
	num_unique_operands	√	√	√	√	√	√	√	√	√
	num_unique_operators	√	√	√	√	√	√	√	√	√
McCabe	cyclomatic_complexity	√	√	√	√	√	√	√	√	√
	cyclomatic_density	√		√	√	√	√	√	√	√
	design_complexity	√	√	√	√	√	√	√	√	√
	essential_complexity	√	√	√	√	√	√	√	√	√
Misc.	branch_count	√	√	√	√	√	√	√	√	√
	call_pairs	√		√	√	√	√	√	√	√
	condition_count	√		√	√	√	√	√	√	√
	decision_count	√		√	√	√	√	√	√	√
	decision_density	√		√	√	√	√	√	√	√
	edge_count	√		√	√	√	√	√	√	√
	essential_density	√		√	√	√	√	√	√	√
	parameter_count	√		√	√	√	√	√	√	√
	maintenance_severity	√		√	√	√	√	√	√	√
	modified_condition_count	√		√	√	√	√	√	√	√
	multiple_condition_count	√		√	√	√	√	√	√	√
	global_data_complexity			√	√					
	global_data_density			√	√					
	normalized_cyclo_complx	√		√	√	√	√	√	√	√
	percent_comments	√		√	√	√	√	√	√	√
	node_count	√		√	√	√	√	√	√	√
Programming Language		C	C++	Java	C	C	C	C	C	C
Number of Code Attributes		37	21	39	39	37	37	36	37	37
Number of Modules		344	2096	200	127	264	759	1585	1125	1399
Number of fp Modules		42	325	36	44	27	61	16	140	178

Code Attribute

```
1.  void main()
2.  {
3.          //This is a sample code

4.          //Declare variables
5.          int a, b, c;

6.          // Initialize variables
7.          a=2;
8.          b=5;

9.          //Find the sum and display c if greater
    than zero
10.         c=sum(a,b);
11.         if c < 0
12.           printf("%d\n", a);
13.         return;
14. }


15. int sum(int a, int b)
16. {
17.         // Returns the sum of two numbers
18.         return a+b;
19. }
```

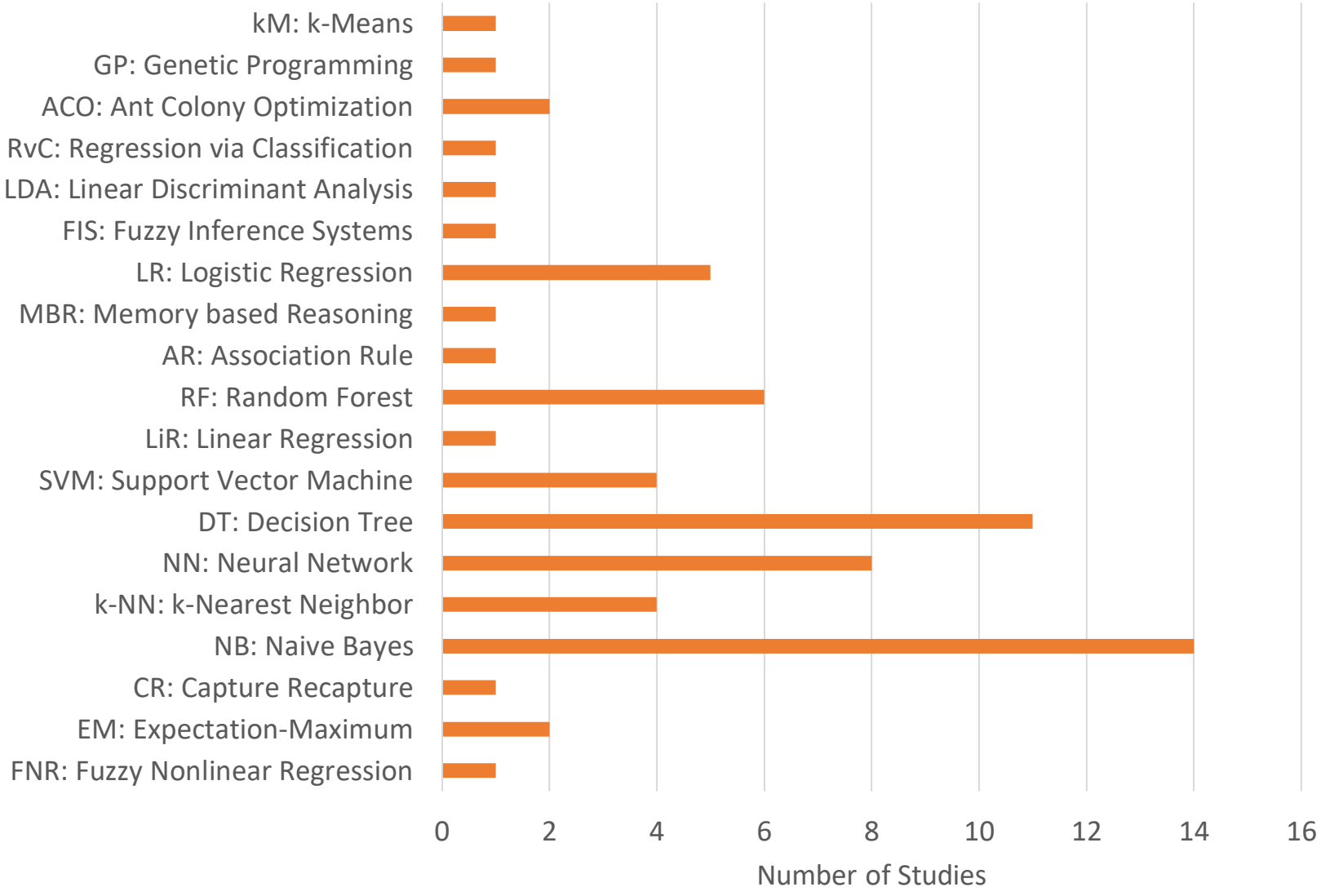
c > 0

c

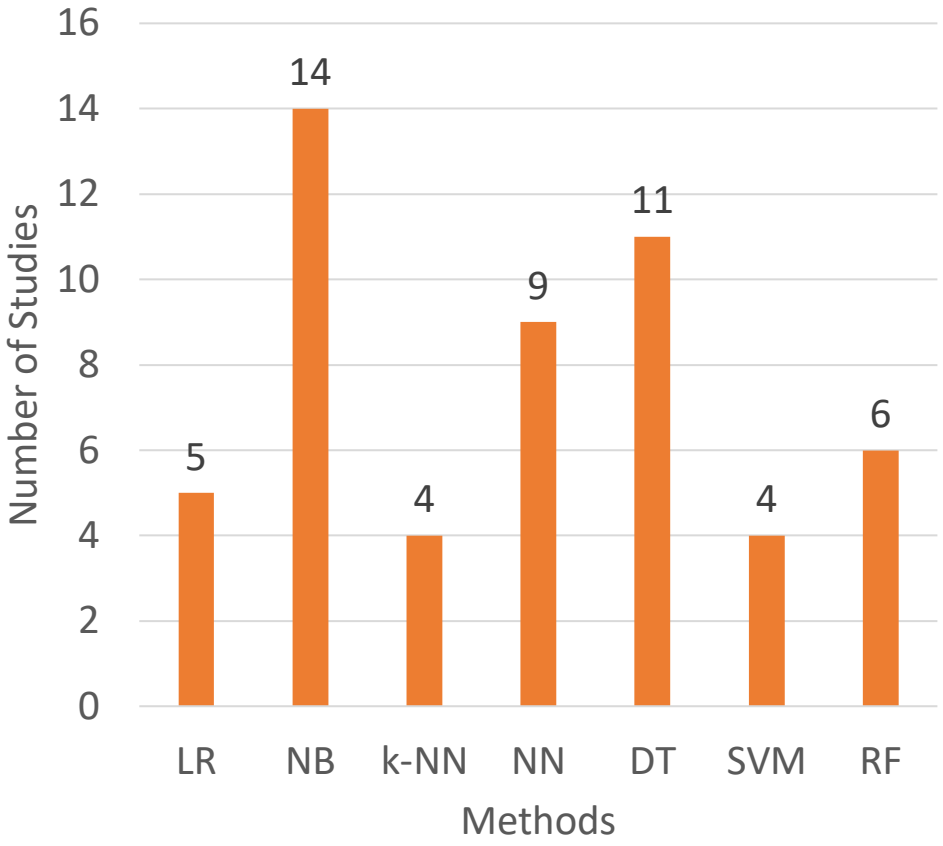
Module	LOC	LOCC	V	CC	Error
main()	16	4	5	2	**2**
sum()	5	1	3	1	**0**

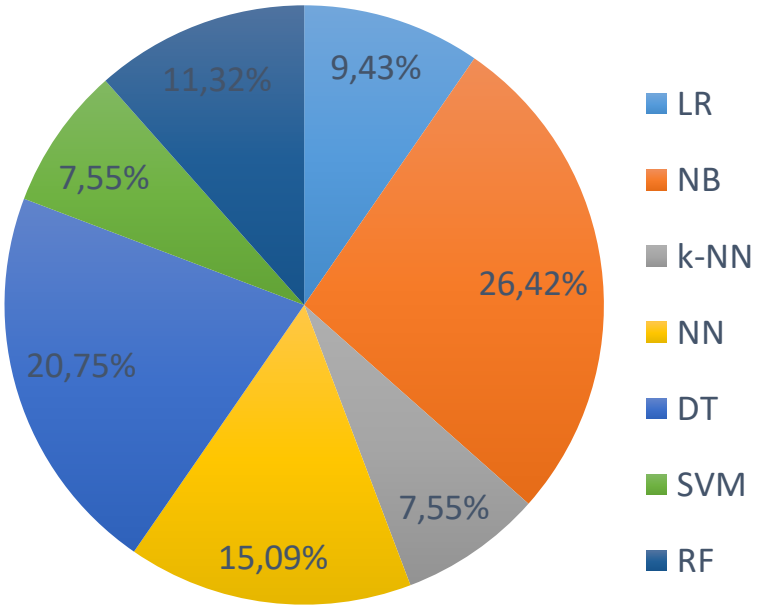
LOC: Line of Code
LOCC: Line of commented Code
V: Number of unique operands&operators
CC: Cyclometric Complexity

RQ5: Software Defect Prediction Methods



RQ6: Most Used Software Defect Prediction Methods





RQ7: Method Comparison Results

- The comparisons and benchmarking result of the defect prediction using machine learning classifiers indicate that:
  - ✓Poor accuracy level is dominant (Lessmann et al. 2008)
  - ✓No significant performance differences could be detected (Lessmann et al. 2008)
  - ✓No particular classifiers that performs the best for all the data sets (Song et al. 2011) (Hall et al. 2012)

- The accurate and reliable classification algorithms to build a better prediction model is an open issue in software defect prediction

RQ8: Method Improvement Efforts

- Researchers proposed some techniques for improving the accuracy of classifiers for software defect prediction
- Recent proposed techniques try to increase the prediction accuracy of a generated model:
  - ✓ By modifying and ensembling some machine learning methods (Mısırlı et al. 2011) (Tosun et al. 2008)
  - ✓ By using boosting algorithm (Zheng 2010) (Jiang et al. 2011)
  - ✓ by adding feature selection (Gayatri et al. 2010) (Khoshgoftaar & Gao, 2009) (Song et al. 2011)
  - ✓ By using parameter selection for some classifiers (Peng & Wang 2010) (Lin et al. 2008) (Guo et al. 2008)
- While considerable works have been done separately, limited research can be found on investigating them all together

RQ9: Existing Frameworks

Three frameworks have been highly cited and influential in software defect prediction field

Menzies Framework

(Menzies et al. 2007)

Lessmann Framework

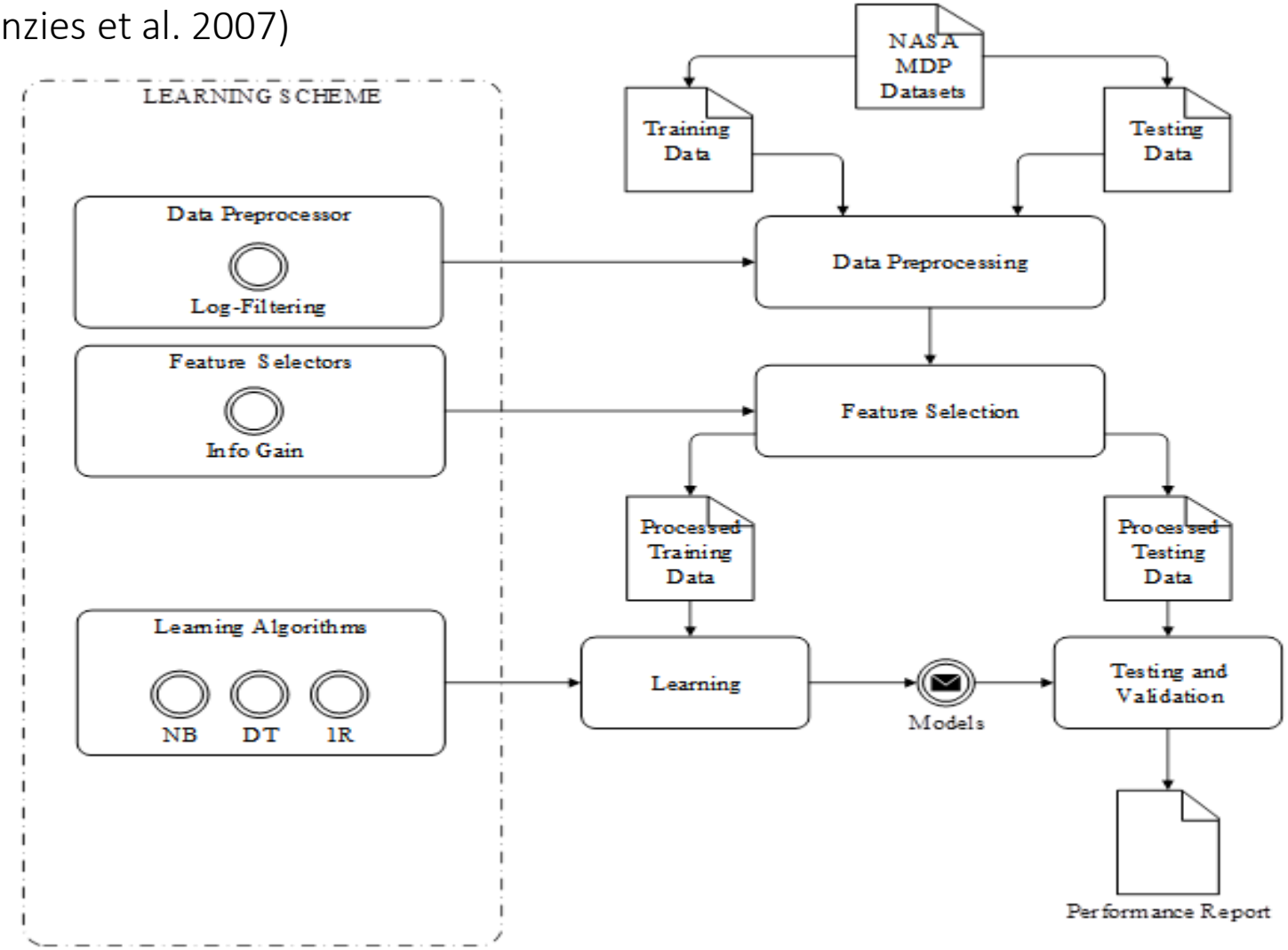
(Lessmann et al. 2008)

Song Framework

(Song et al. 2011)

Menzies Framework

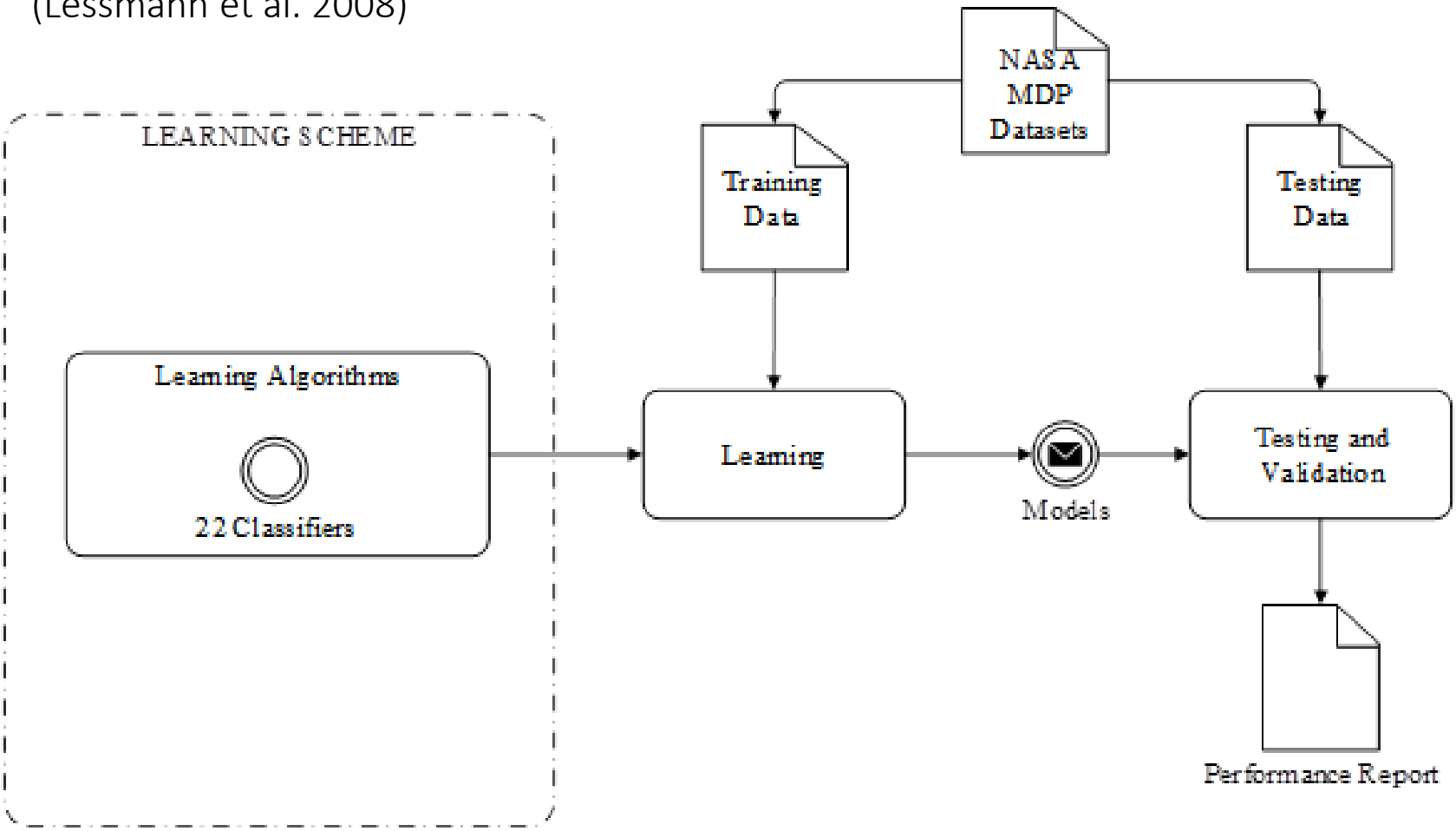
(Menzies et al. 2007)



Framework	Dataset	Data Preprocessor	Feature Selectors	Meta-learning	Classifiers	Parameter Selectors	Validation Methods	Evaluation Methods
(Menzies et al. 2007)	NASA MDP	**Log Filtering**	**Info Gain**	-	**3 algorithms (DT, 1R, NB)**	-	10-Fold X Validation	ROC Curve (AUC)

Lessmann Framework

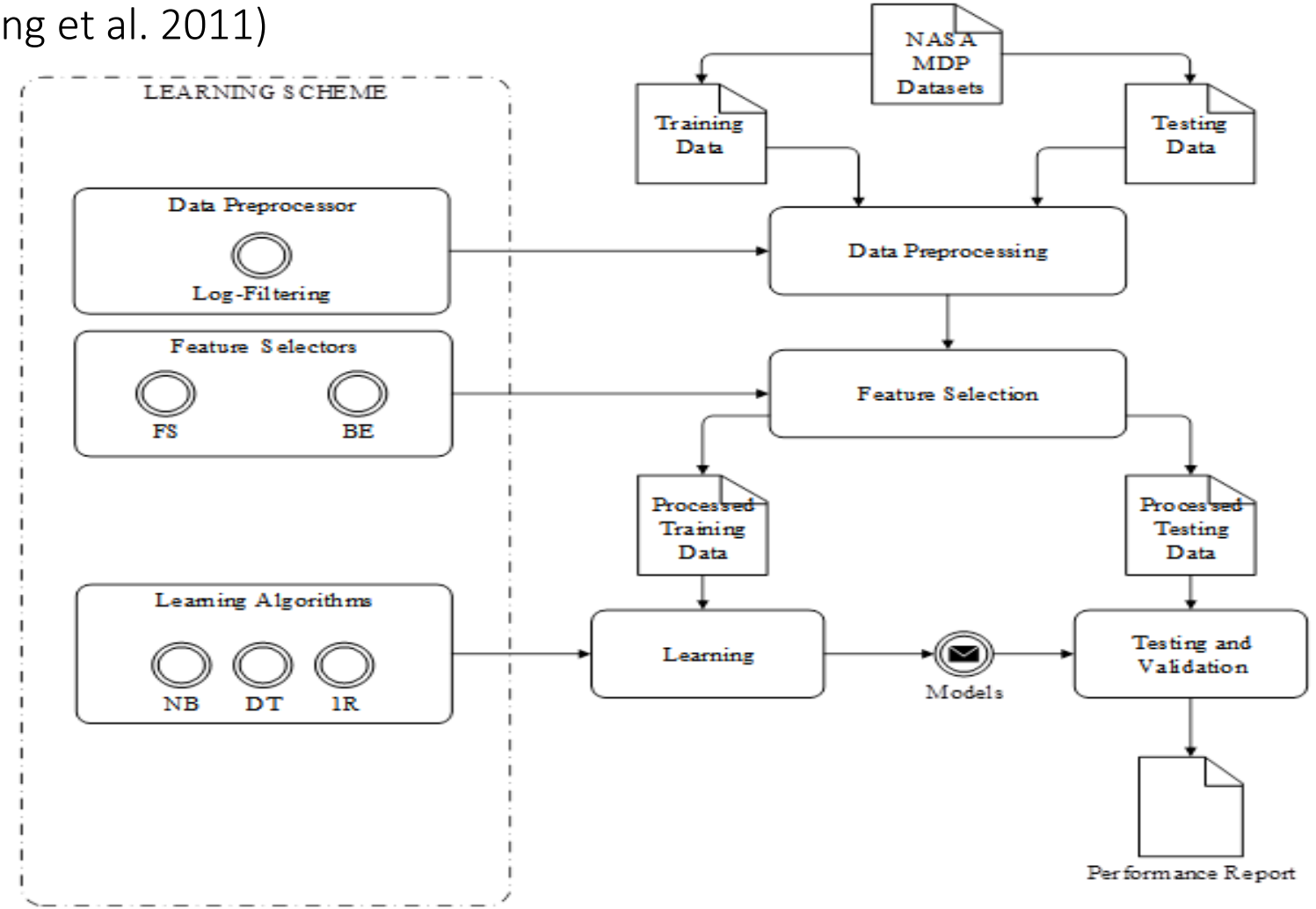
(Lessmann et al. 2008)



Framework	Dataset	Data Preprocessor	Feature Selectors	Meta-learning	Classifiers	Parameter Selectors	Validation Methods	Evaluation Methods
(Lessman et al. 2008)	NASA MDP	-	-	-	**22 algorithms**	-	10-Fold X Validation	ROC Curve (AUC)

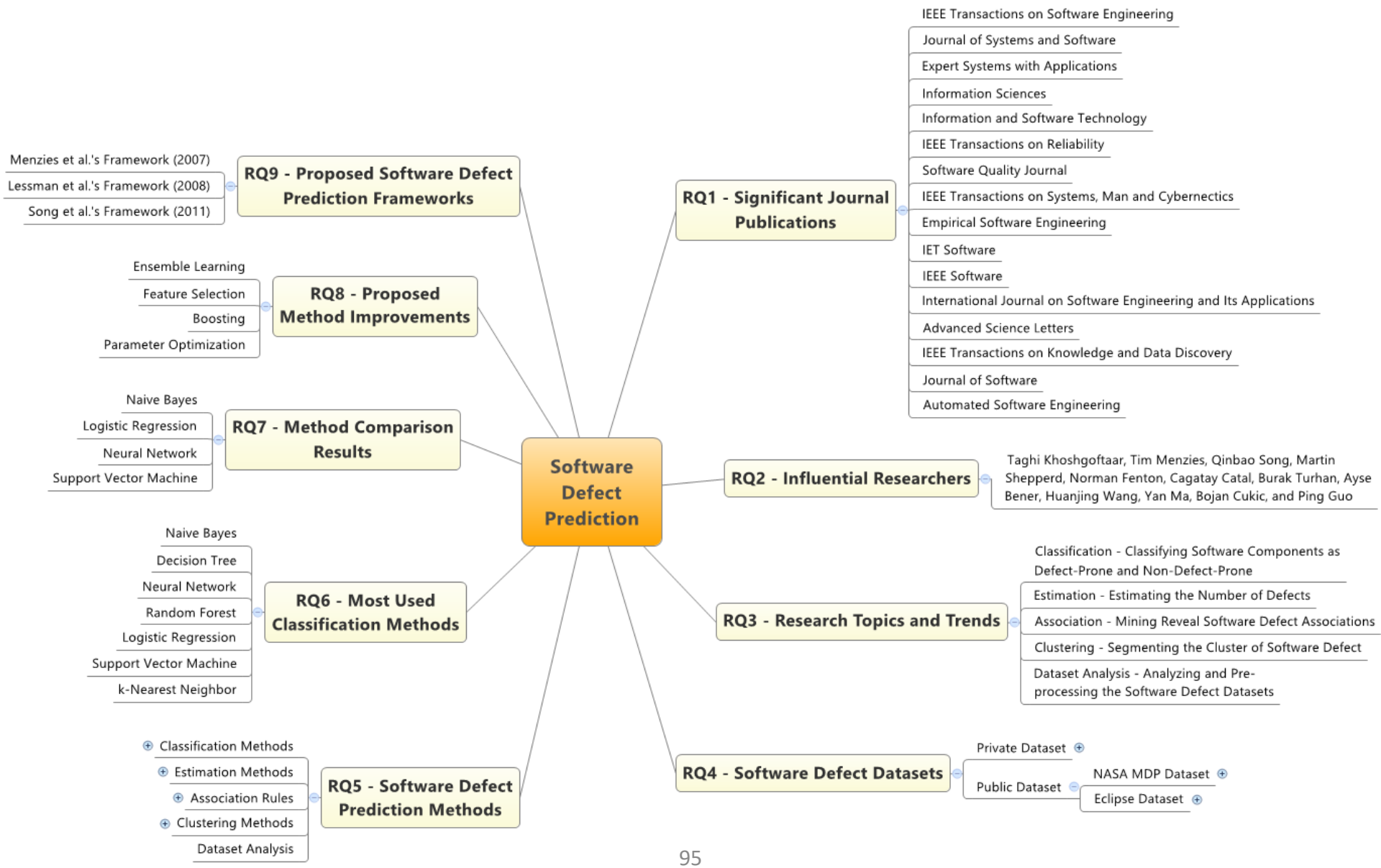
Song Framework

(Song et al. 2011)



Framework	Dataset	Data Preprocessor	Feature Selectors	Meta-learning	Classifiers	Parameter Selectors	Validation Methods	Evaluation Methods
(Song et al. 2011)	NASA MDP	**Log Filtering**	**FS, BE**	-	**3 algorithms (DT, 1R, NB)**	-	10-Fold X Validation	ROC Curve (AUC)

Mind Map of the SLR Results



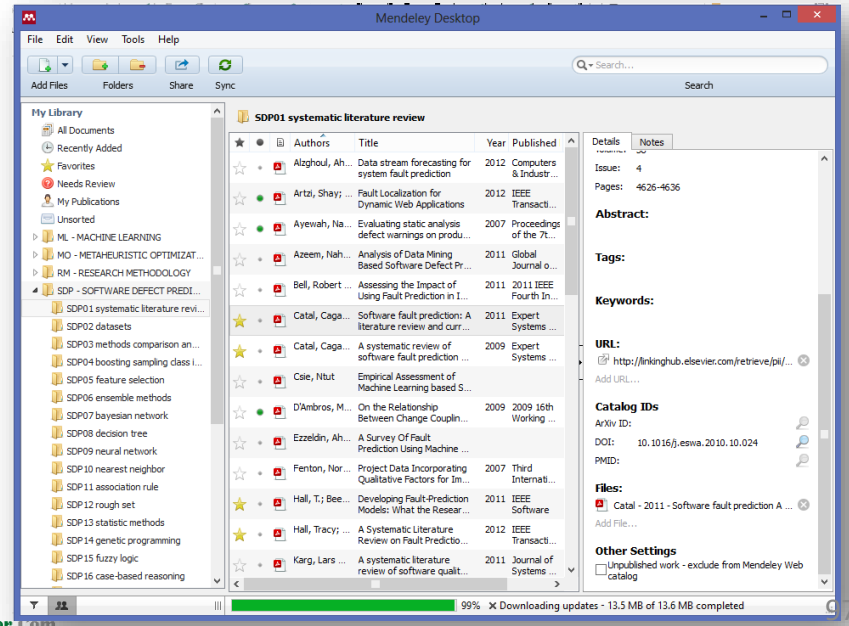
MITOS 6

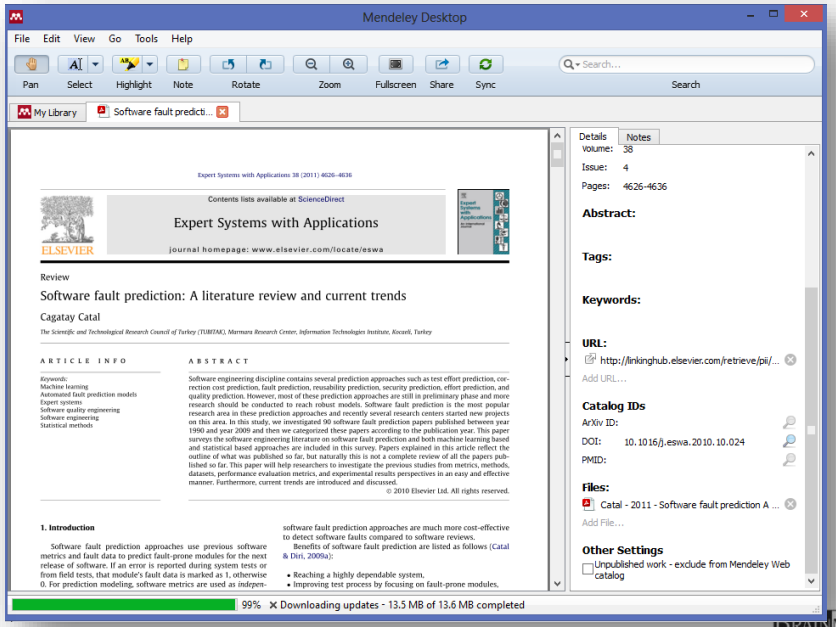
Semakin Banyak Literatur yang Saya Baca, Saya Semakin Pusing

Jumlah Literatur yang Harus Dibaca

- Adagium level pendidikan dan jumlah literatur yang harus dibaca untuk penyelesaian penelitian
  - **S1**: 20-70 paper
  - **S2**: 70-200 paper
  - **S3**: 200-700 paper

- Kepala jadi pusing, bukan karena kita banyak membaca, tapi karena yang kita baca memang "belum banyak"





Tahapan Penelitian Computing

Literature Review

1. Penentuan Bidang Penelitian (***Research Field***)
2. Penentuan Topik Penelitian (***Research Topic***)
3. Penentuan Masalah Penelitian (***Research Problem***)
4. Perangkuman Metode-Metode Yang Ada (***State-of-the-Art Methods***)
5. Penentuan Metode Yang Diusulkan (***Proposed Method***)
6. Evaluasi Metode Yang Diusulkan (***Evaluation***)
7. Penulisan Ilmiah dan Publikasi Hasil Penelitian (***Publications***)

**https://www.site.uottawa.ca/~bochmann/dsrg/how-to-do-good-research/*
**http://romisatriawahono.net/2013/01/23/tahapan-memulai-penelitian-untuk-mahasiswa-galau/*

SLR Melahirkan Research Gaps

Dari Hasil SLR, Kita Menemukan Research Gaps yang Menjadi Kandidat Masalah Penelitian yang Kita Angkat pada Penelitian Kita

Gap Analysis of Framework

1. The comparisons and benchmarking result of the defect prediction using machine learning classifiers indicate that:
   - Poor accuracy level is dominant (Lessmann et al. 2008)
   - No significant performance differences could be detected (Lessmann et al. 2008)
   - No particular classifiers that performs the best for all the data sets (Song et al. 2011) (Hall et al. 2012)
2. Noisy attribute predictors and imbalanced class distribution of software defect datasets result in inaccuracy of classification models
3. Neural network and support vector machine have strong fault tolerance and strong ability of nonlinear dynamic processing of software fault data, but practicability of neural network and support vector machine are limited due to difficulty of selecting appropriate parameters

Masalah Penelitian dan Landasannya

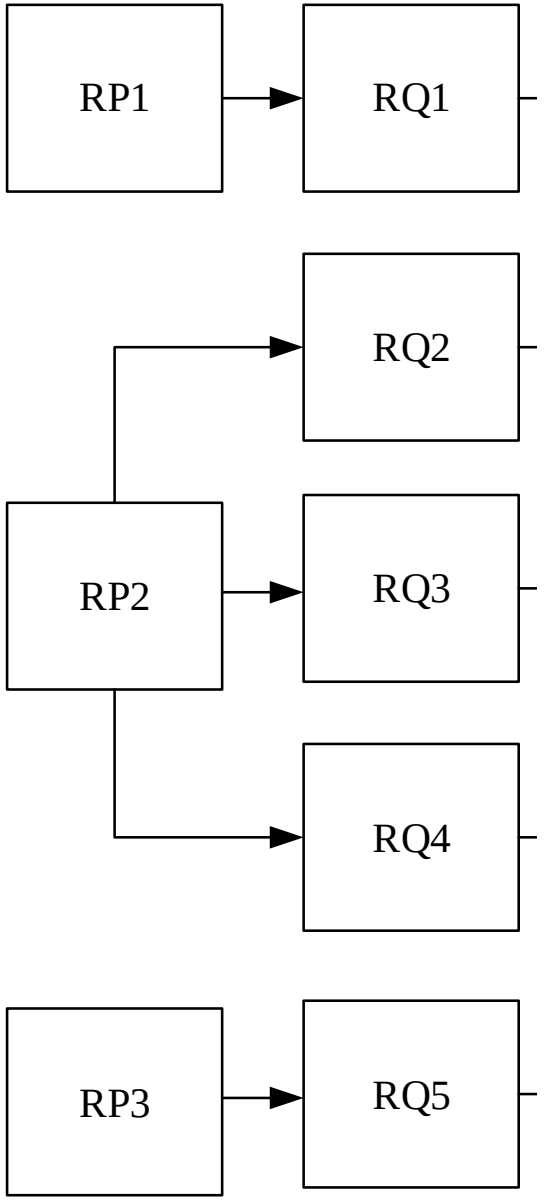
Masalah Penelitian	Landasan Literatur
Data set pada prediksi cacat software berdimensi tinggi, memiliki atribut yang bersifat noisy, dan classnya bersifat tidak seimbang, menyebabkan penurunan akurasi pada prediksi cacat software	There are noisy data points in the software defect data sets that can not be confidently assumed to be erroneous using such simple method *(Gray, Bowes, Davey, & Christianson, 2011)*
	The performances of software defect prediction improved when irrelevant and redundant attributes are removed *(Wang, Khoshgoftaar, & Napolitano, 2010)*
	The software defect prediction performance decreases significantly because the dataset contains noisy attributes *(Kim, Zhang, Wu, & Gong, 2011)*
	Software defect datasets have an imbalanced nature with very few defective modules compared to defect-free ones *(Tosun, Bener, Turhan, & Menzies, 2010)*
	Imbalance can lead to a model that is not practical in software defect prediction, because most instances will be predicted as non-defect prone *(Khoshgoftaar, Van Hulse, & Napolitano, 2011)*
	Software fault prediction data sets are often highly imbalanced *(Zhang & Zhang, 2007)*

Research Problems (RP)

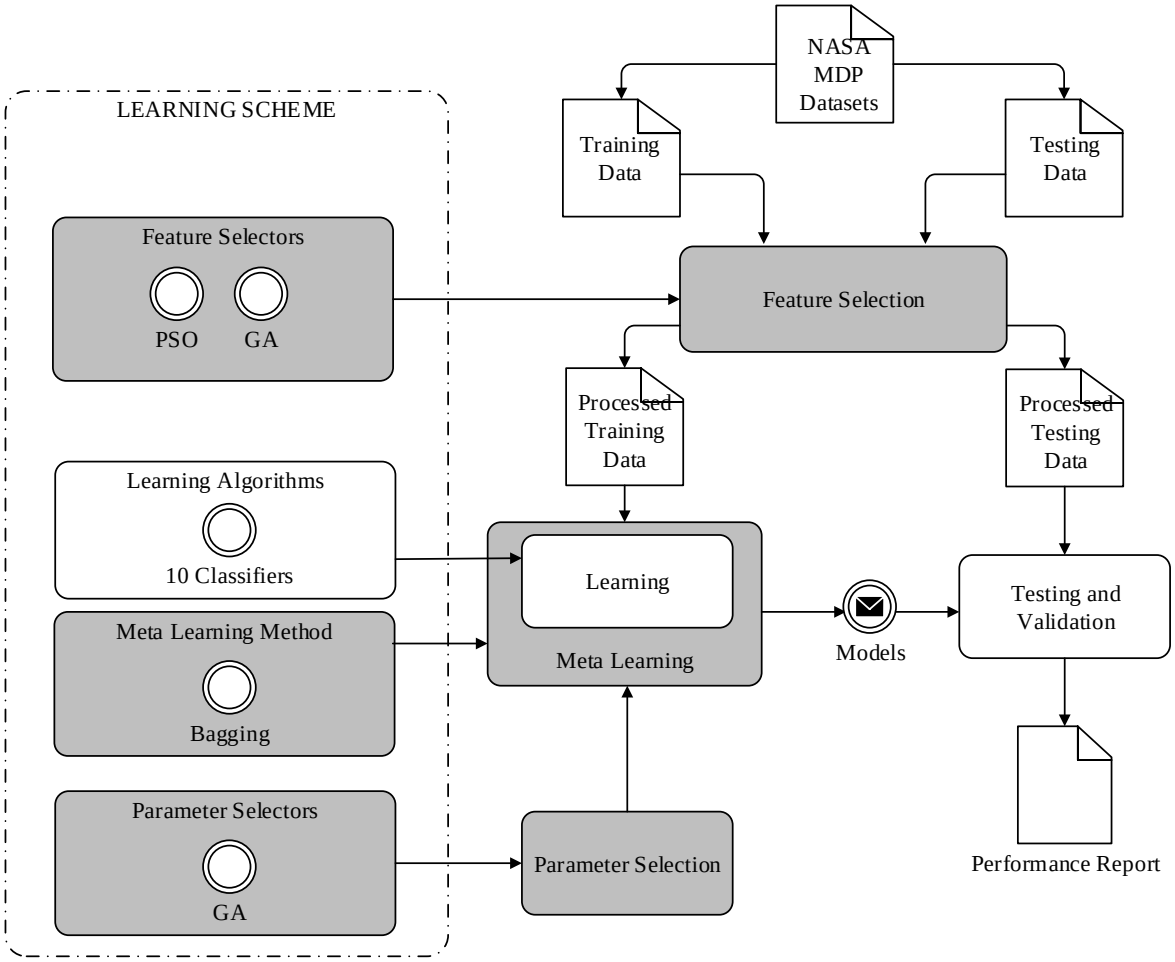
RP1 While many studies on software defect prediction report the comparative performance of the classification algorithms used, but there is no strong consensus on which classifiers perform best when individual studies are looked separately

RP2 Noisy attribute predictors and imbalanced class distribution of software defect datasets result in inaccuracy of classification models

RP3 Neural network has strong fault tolerance and strong ability of nonlinear dynamic processing of software fault data, but practicability of neural network is limited due to difficulty of selecting appropriate parameters



Proposed Framework



Framework	Dataset	Data Preprocessor	Feature Selectors	Meta-Learning	Classifiers	Parameter Selectors	Validation Methods	Evaluation Methods
(Menzies et al. 2007)	NASA MDP	Log Filtering	Info Gain		3 algorithm (DT, 1R, NB)	-	10-Fold X Validation	ROC Curve (AUC)
(Lessman et al. 2008)	NASA MDP	-	-		22 algorithm	-	10-Fold X Validation	ROC Curve (AUC)
(Song et al. 2011)	NASA MDP	Log Filtering	FS, BE		3 algorithm (DT, 1R, NB)	-	10-Fold X Validation	ROC Curve (AUC)
Proposed Framework	NASA MDP	-	**PSO, GA**	**Bagging**	**10 algorithms**	**GA**	10-Fold X Validation	ROC Curve (AUC)

Akademisi vs Technopreneur

Meja Indah

Meja Kuat

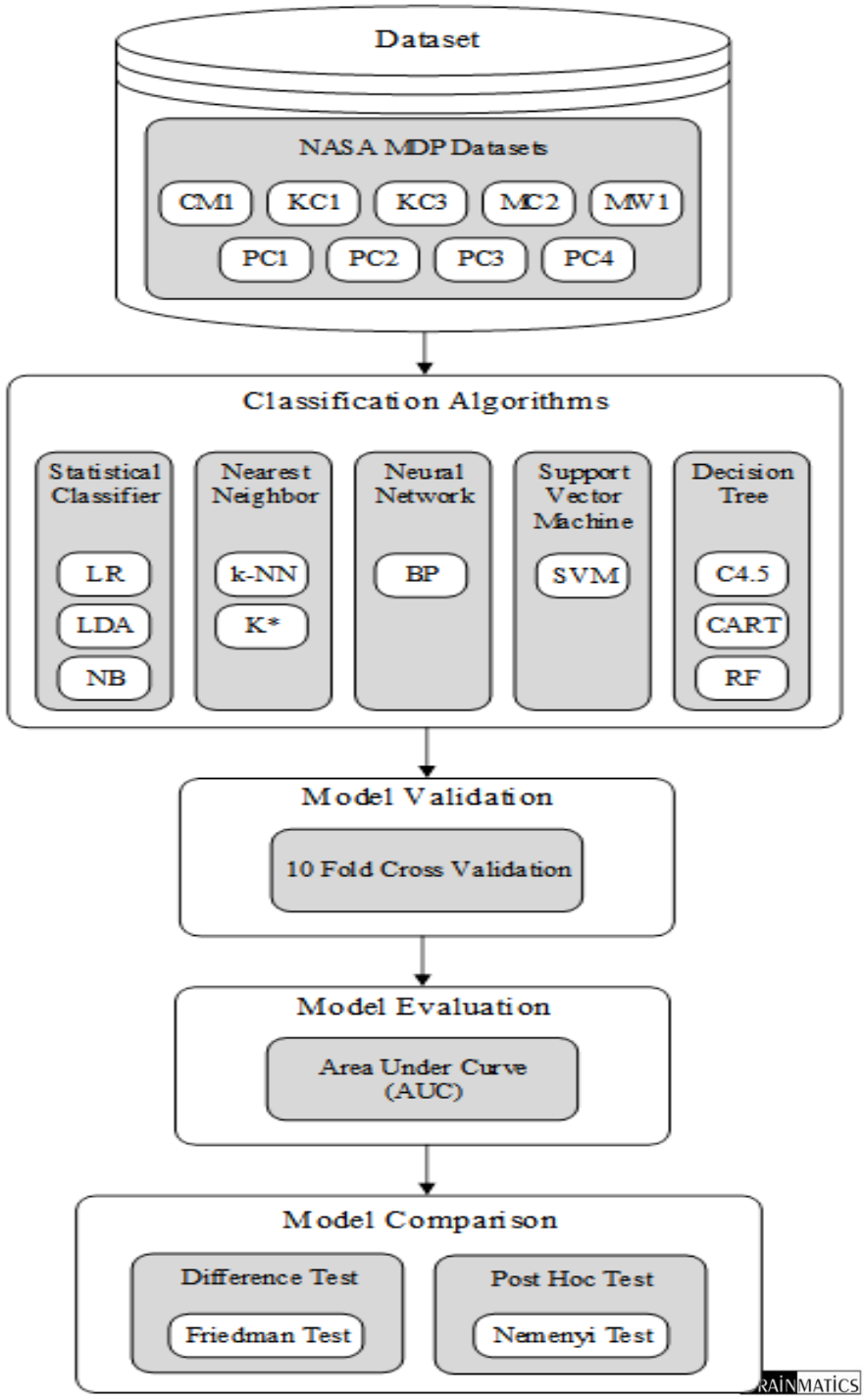
Meja Luas

- Technopreneur?
  1. Jual Produk
  2. Beri Nilai Tambah Produk
  3. Jadikan Aset, Jual Layanan
- Akademisi?
  - Pelajari, Preteli Komponen
  - Ciptakan Meja Baru yang Berbeda dengan 3 Meja Itu

Research Result on RQ1

Research Problems (RP)		Research Questions (RQ)		Research Objectives (RO)	
RP1	While many studies on software defect prediction report the comparative performance of the modelling techniques they have used, no clear consensus on which classifier perform best emerges when individual studies are looked at separately	RQ1	Which machine learning classification algorithms perform best when used in software defect prediction?	RO1	To identify and determine the best machine learning classification algorithms when used in software defect prediction

A Comparison Framework of Classification Models for Software Defect Prediction (CF SDP)



AUC and Friedman Test Results

	CM1	KC1	KC3	MC2	MW1	PC1	PC2	PC3	PC4	*M*	*R*
LR	**0.763**	**0.801**	**0.713**	**0.766**	0.726	**0.852**	0.849	**0.81**	0.894	**0.797**	**1.44**
LDA	0.471	0.536	0.447	0.503	0.58	0.454	0.577	0.524	0.61	0.522	8.33
NB	0.734	0.786	0.67	0.739	0.732	0.781	0.811	0.756	0.838	0.761	3
k-NN	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	8.778
K*	0.6	0.678	0.562	0.585	0.63	0.652	0.754	0.697	0.76	0.658	5.33
BP	0.713	0.791	0.647	0.71	0.625	0.784	0.918	0.79	0.883	0.762	3.22
SVM	0.753	0.752	0.642	0.761	0.714	0.79	0.534	0.75	**0.899**	0.733	3.33
C4.5	0.565	0.515	0.497	0.455	0.543	0.601	0.493	0.715	0.723	0.567	7.78
CART	0.604	0.648	0.637	0.482	0.656	0.574	0.491	0.68	0.623	0.599	6.89
RF	0.573	0.485	0.477	0.525	**0.74**	0.618	0.649	0.678	0.2	0.549	6.89

- LR is dominant in most datasets
- *R* rank: LR has the highest rank, followed by NB, BP, and SVM
- *M* results: no excellent or good models, and a few fair models

AUC	Meaning	Symbol
0.90 - 1.00	excellent classification	
0.80 - 0.90	good classification	
0.70 - 0.80	fair classification	
0.60 - 0.70	poor classification	
< 0.60	failure	

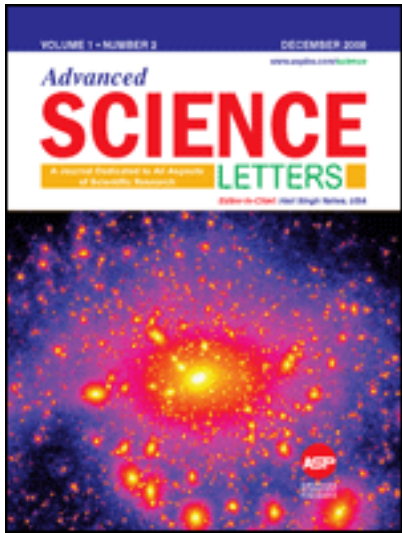
P-value of Nemenyi Post Hoc Test

	LR	LDA	NB	k-NN	K*	BP	SVM	C4.5	CART	RF
LR	1	**0.0001**	0.986	**0.0001**	0.164	0.965	0.949	**0.000**	**0.005**	**0.005**
LDA	**0.0001**	1	**0.007**	1.000	0.526	**0.013**	**0.017**	1.000	0.992	0.992
NB	0.986	**0.007**	1	**0.002**	0.831	1.000	1.000	**0.028**	0.164	0.164
k-NN	**0.0001**	1.000	**0.002**	1	0.318	**0.004**	**0.005**	1.000	0.949	0.949
K*	0.164	0.526	0.831	0.318	1	0.901	0.927	0.789	0.986	0.986
BP	0.965	**0.013**	1.000	**0.004**	0.901	1	1.000	**0.046**	0.232	0.232
SVM	0.949	**0.017**	1.000	**0.005**	0.927	1.000	1	0.058	0.273	0.273
C4.5	**0.000**	1.000	**0.028**	1.000	0.789	**0.046**	0.058	1	1.000	1.000
CART	**0.005**	0.992	0.164	0.949	0.986	0.232	0.273	1.000	1	1.000
RF	**0.005**	0.992	0.164	0.949	0.986	0.232	0.273	1.000	1.000	1

- If *P value < 0.05* (boldfaced print), it indicate that there is significant different between two classifiers
- Based on significant difference results, there is no significant difference between LR, NB, BP, and SVM models

Research Publication on RQ1

1. Romi Satria Wahono, Nanna Suryana Herman and Sabrina Ahmad, A Comparison Framework of Classification Models for Software Defect Prediction, Proceedings of the 2014 International Conference on Internet Services Technology and Information Engineering (ISTIE 2014), Bali, Indonesia, May 2014
2. Romi Satria Wahono, Nanna Suryana Herman and Sabrina Ahmad, A Comparison Framework of Classification Models for Software Defect Prediction, Advanced Science Letters, Vol. 20, No. 8, August 2014 (SCOPUS SJR: 0.240)

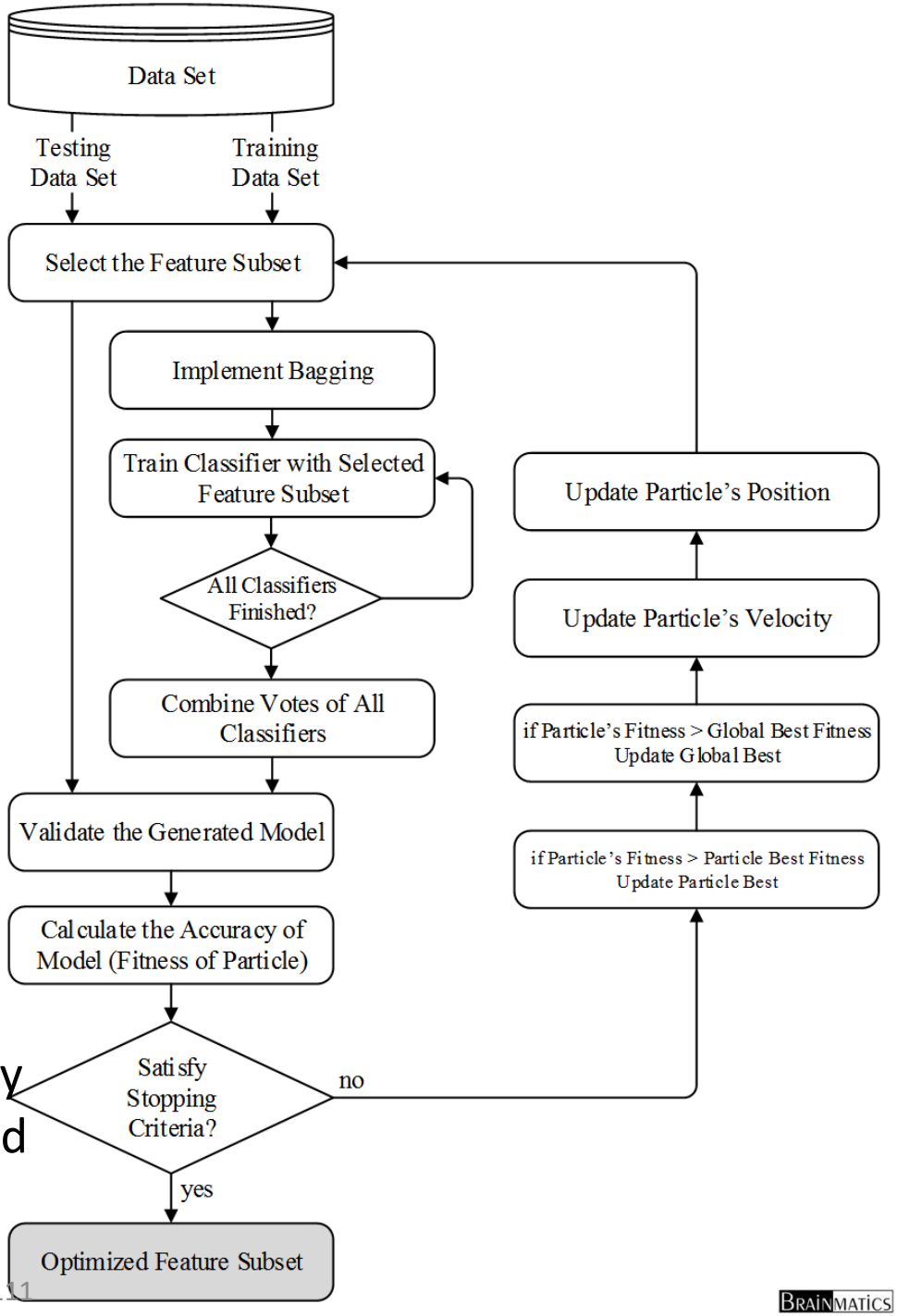


Research Result on RQ3

Research Problems (RP)		Research Questions (RQ)		Research Objectives (RO)	
RP2	**Noisy attribute predictors and imbalanced class distribution of software defect datasets result in inaccuracy of classification models**	RQ2	How does the integration between genetic algorithm based feature selection and bagging technique affect the accuracy of software defect prediction?	RO2	To develop a hybrid genetic algorithm based feature selection and bagging technique for improving the accuracy of software defect prediction
		RQ3	**How does the integration between particle swarm optimization based feature selection and bagging technique affect the accuracy of software defect prediction?**	**RO3**	**To develop a hybrid particle swarm optimization based feature selection and bagging technique for improving the accuracy of software defect prediction**
		RQ4	Which metaheuristic optimization techniques perform best when used in feature selection of software defect prediction?	RO4	To identify the best metaheuristic optimization techniques when used in feature selection of software defect prediction

A Hybrid Particle Swarm Optimization based Feature Selection and Bagging Technique for Software Defect Prediction (PSOFS+B)

- Each particle represents a feature subset, which is a candidate solution
- Implement bagging technique and train the classifier on the larger training set based on the selected feature subset and the type of kernel
- If all classifiers are finished, combine votes of all classifiers
- Finally, measure validation accuracy on testing dataset via the generated model



Results: With PSOFS+B

Classifiers		CM1	KC1	KC3	MC2	MW1	PC1	PC2	PC3	PC4
Statistical Classifier	LR	0.738	0.798	0.695	**0.78**	**0.751**	0.848	0.827	**0.816**	**0.897**
	LDA	0.469	**0.627**	**0.653**	**0.686**	**0.632**	**0.665**	0.571	**0.604**	**0.715**
	NB	**0.756**	**0.847**	**0.71**	0.732	**0.748**	**0.79**	**0.818**	**0.78**	**0.85**
Nearest Neighbor	k-NN	**0.632**	**0.675**	**0.578**	**0.606**	**0.648**	**0.547**	**0.594**	**0.679**	**0.738**
	K*	**0.681**	**0.792**	**0.66**	**0.725**	0.572	**0.822**	0.814	**0.809**	**0.878**
Neural Network	BP	0.7	**0.799**	**0.726**	**0.734**	**0.722**	**0.809**	0.89	**0.823**	**0.915**
Support Vector Machine	SVM	0.721	0.723	**0.67**	0.756	0.667	**0.792**	0.294	0.735	**0.903**
Decision Tree	C4.5	**0.682**	**0.606**	**0.592**	**0.648**	**0.615**	**0.732**	**0.732**	**0.78**	**0.769**
	CART	**0.611**	**0.679**	**0.787**	**0.679**	**0.682**	**0.831**	**0.794**	**0.845**	**0.912**
	RF	**0.62**	**0.604**	**0.557**	**0.533**	0.714	**0.686**	**0.899**	**0.759**	**0.558**

- Almost all classifiers that implemented PSOFS+B outperform the original method
- Proposed PSOFS+B method affected significantly on the performance of the class imbalance suffered classifiers

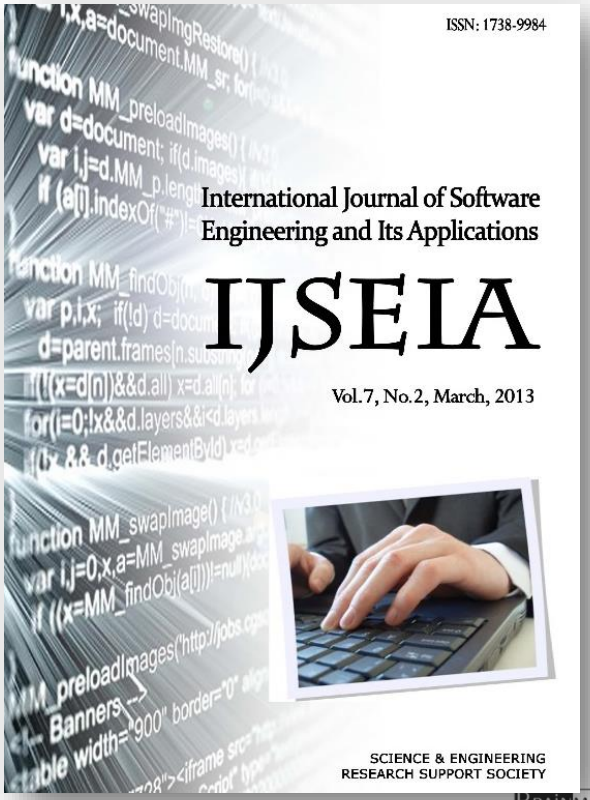
Without PSOFS+B vs With PSOFS+B

Classifiers		*P* value of t-Test	Result
Statistical Classifier	LR	0.323	Not Sig. ($P > 0.05$)
	LDA	**0.003**	**Sig. ($P < 0.05$)**
	NB	**0.007**	**Sig. ($P < 0.05$)**
Nearest Neighbor	k-NN	**0.00007**	**Sig. ($P < 0.05$)**
	K*	**0.001**	**Sig. ($P < 0.05$)**
Neural Network	BP	**0.03**	**Sig. ($P < 0.05$)**
Support Vector Machine	SVM	0.09	Not Sig. ($P > 0.05$)
Decision Tree	C4.5	**0.0002**	**Sig. ($P < 0.05$)**
	CART	**0.002**	**Sig. ($P < 0.05$)**
	RF	**0.01**	**Sig. ($P < 0.05$)**

- Although there are two classifiers that have no significant difference ($P > 0.05$), the results have indicated that those of remaining eight classifiers have significant difference ($P < 0.05$)
- The proposed PSOFS+B method makes an improvement in prediction performance for most classifiers

Research Publication on RQ3

Romi Satria Wahono and Nanna Suryana, *Combining Particle Swarm Optimization based Feature Selection and Bagging Technique for Software Defect Prediction*, International Journal of Software Engineering and Its Applications, Vol 7, No 5, September 2013



Research Result on RQ4

Research Problems (RP)		Research Questions (RQ)		Research Objectives (RO)	
RP2	**Noisy attribute predictors and imbalanced class distribution of software defect datasets result in inaccuracy of classification models**	RQ2	How does the integration between genetic algorithm based feature selection and bagging technique affect the accuracy of software defect prediction?	RO2	To develop a hybrid genetic algorithm based feature selection and bagging technique for improving the accuracy of software defect prediction
		RQ3	How does the integration between particle swarm optimization based feature selection and bagging technique affect the accuracy of software defect prediction?	RO3	To develop a hybrid particle swarm optimization based feature selection and bagging technique for improving the accuracy of software defect prediction
		RQ4	**Which metaheuristic optimization techniques perform best when used in feature selection of software defect prediction?**	**RO4**	**To identify the best metaheuristic optimization techniques when used in feature selection of software defect prediction**

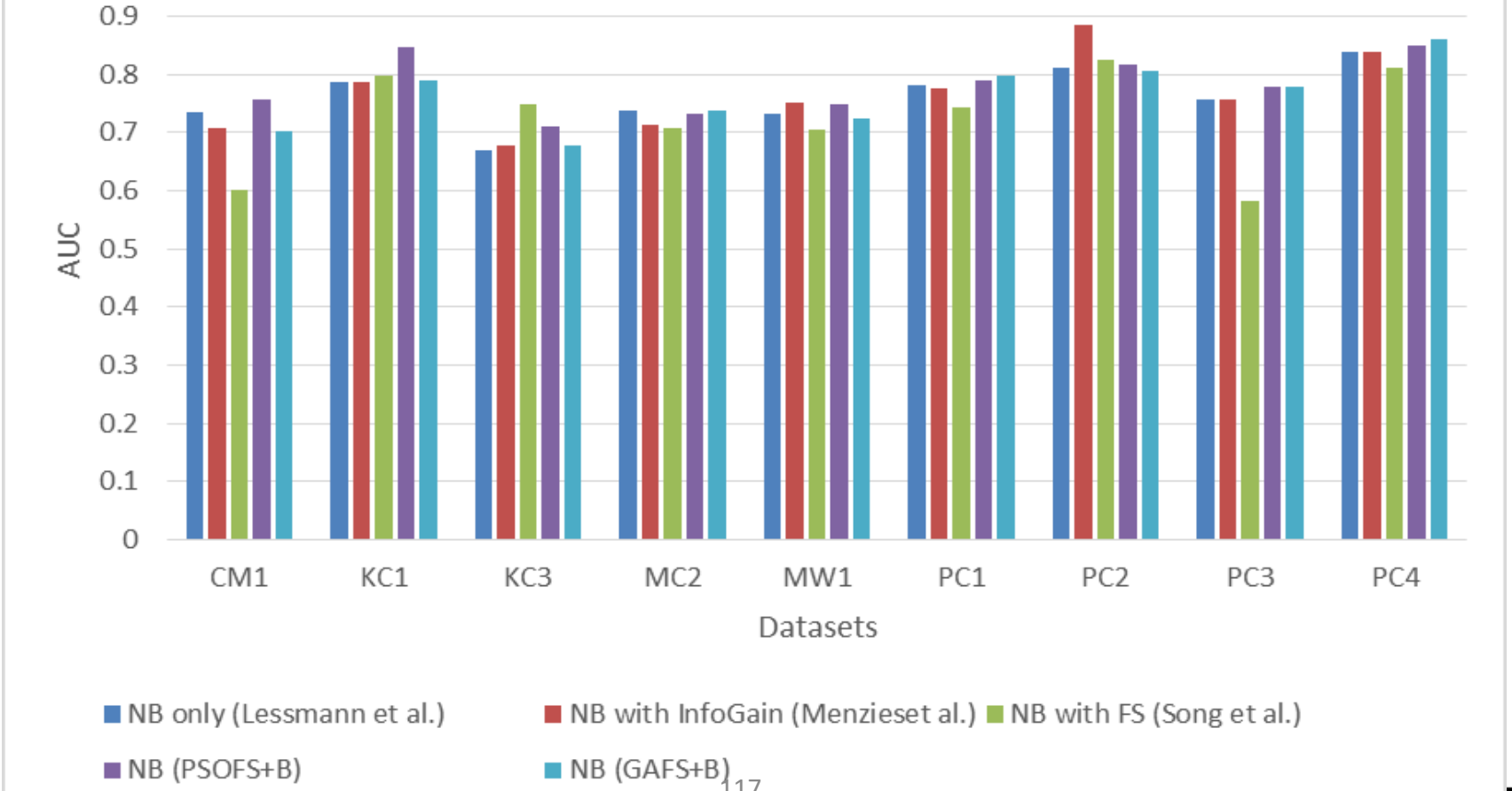
GAFS+B vs PSOFS+B

Classifiers		*P* value of t-Test	Result
Statistical Classifier	**LR**	0.25	Not Sig. ($\alpha > 0.05$)
	LDA	0.19	Not Sig. ($\alpha > 0.05$)
	NB	0.044	**Sig. ($\alpha < 0.05$)**
Nearest Neighbor	**k-NN**	0.063	Not Sig. ($\alpha > 0.05$)
	K*	0.268	Not Sig. ($\alpha > 0.05$)
Neural Network	**BP**	0.203	Not Sig. ($\alpha > 0.05$)
Support Vector Machine	**SVM**	0.003	**Sig. ($\alpha < 0.05$)**
Decision Tree	**C4.5**	0.3	Not Sig. ($\alpha > 0.05$)
	CART	0.216	Not Sig. ($\alpha > 0.05$)
	RF	0.088	Not Sig. ($\alpha > 0.05$)

- Although there are two classifier that have significant difference ($P < 0.05$) (NB and SVM), the results have indicated that those of remaining eight classifiers have no significant difference ($P > 0.05$)
- There is no significant difference between PSO and GA when used as feature selection for most classifiers

Proposed Methods vs Other Methods

	CM1	KC1	KC3	MC2	MW1	PC1	PC2	PC3	PC4
NB only (Lessmann et al.)	0.734	0.786	0.67	**0.739**	0.732	0.781	0.811	0.756	0.838
NB with InfoGain (Menzies et al.)	0.708	0.786	0.677	0.712	**0.752**	0.775	**0.885**	0.756	0.84
NB with FS (Song et al.)	0.601	0.799	**0.749**	0.707	0.704	0.742	0.824	0.583	0.812
NB (PSOFS+B)	**0.756**	**0.847**	0.71	0.732	0.748	0.79	0.818	**0.78**	0.85
NB (GAFS+B)	0.702	0.79	0.677	**0.739**	0.724	**0.799**	0.805	**0.78**	**0.861**

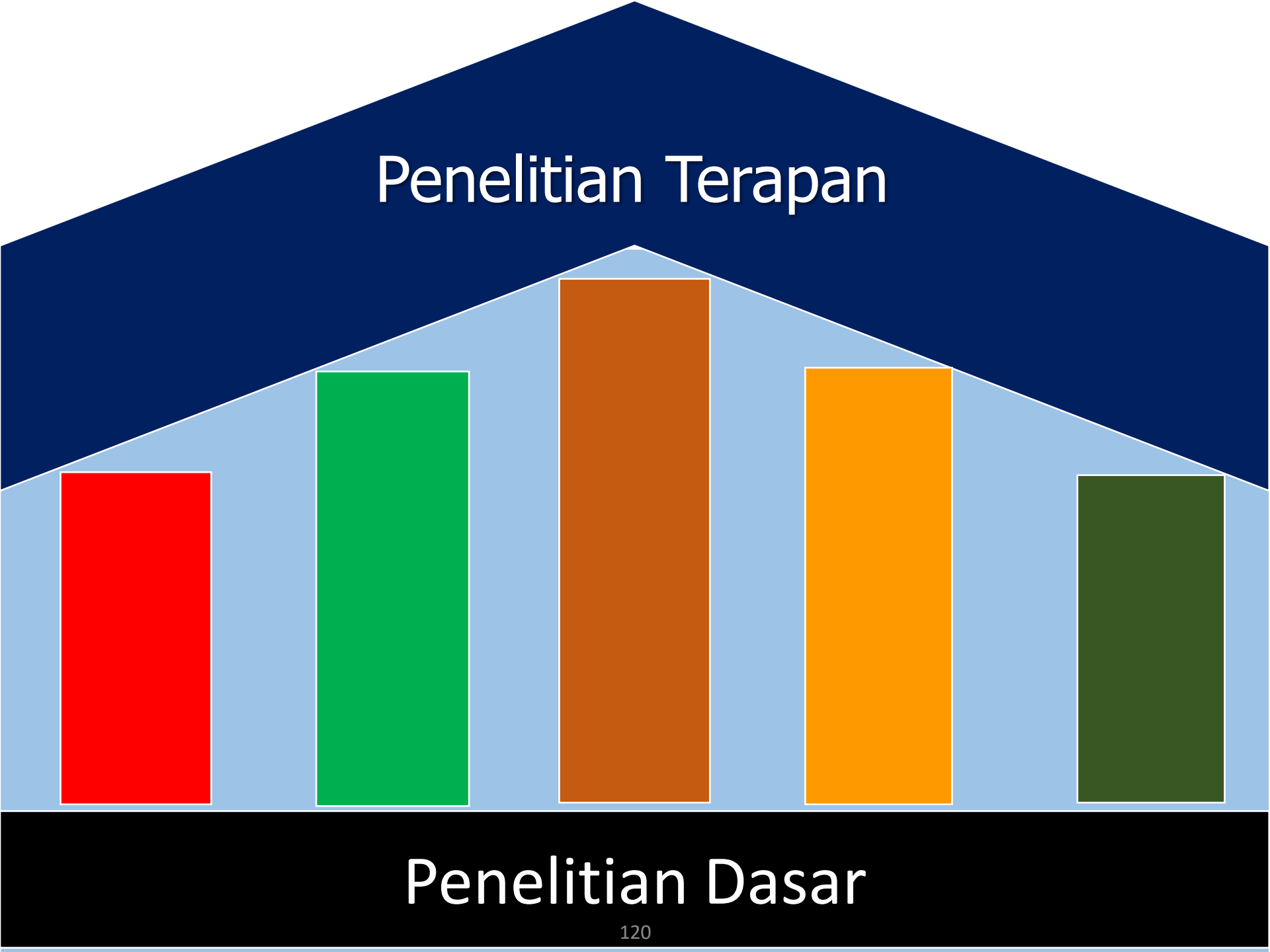


Research Publication on RQ4

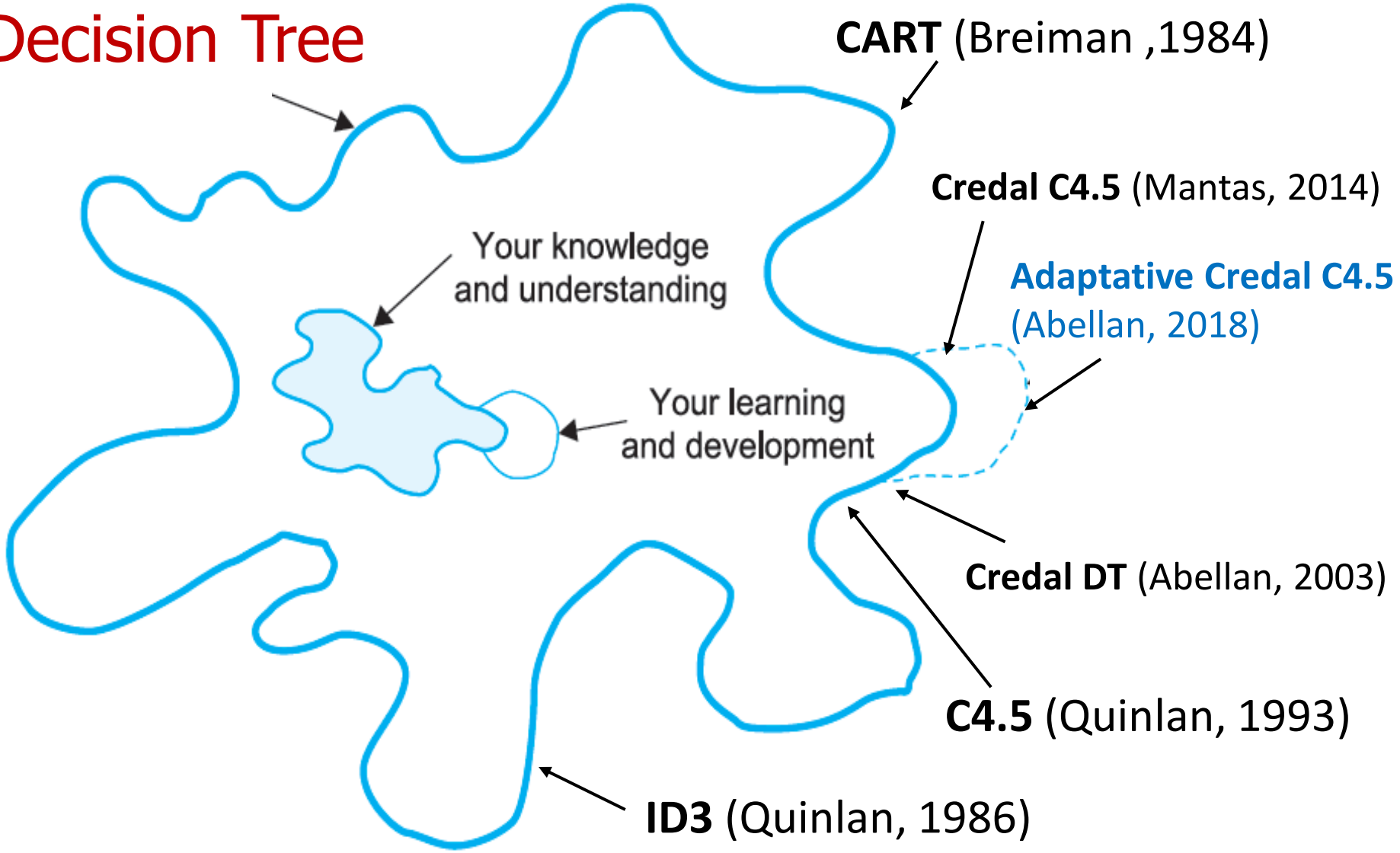
Romi Satria Wahono, Nanna Suryana and Sabrina Ahmad, *Metaheuristic Optimization based Feature Selection for Software Defect Prediction*, Journal of Software, Vol. 9, No. 5, May 2014
(SCOPUS SJR: 0.260)

MITOS 7

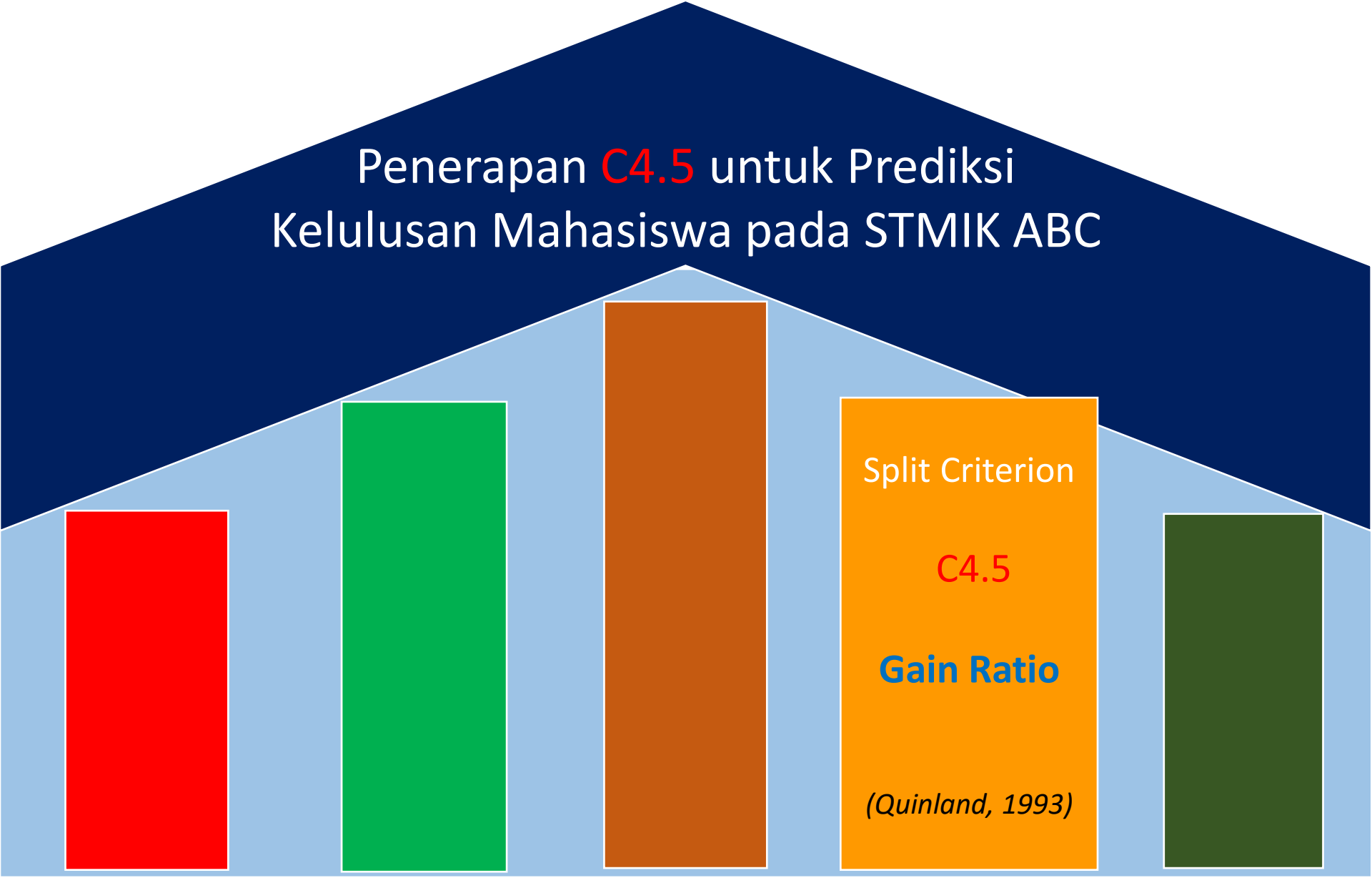
Penelitian Itu Semakin Aplikatif dan Terapan Semakin Mudah Masuk Jurnal Terindeks


Penelitian Terapan
Penelitian Dasar

Bentuk Kontribusi ke Pengetahuan

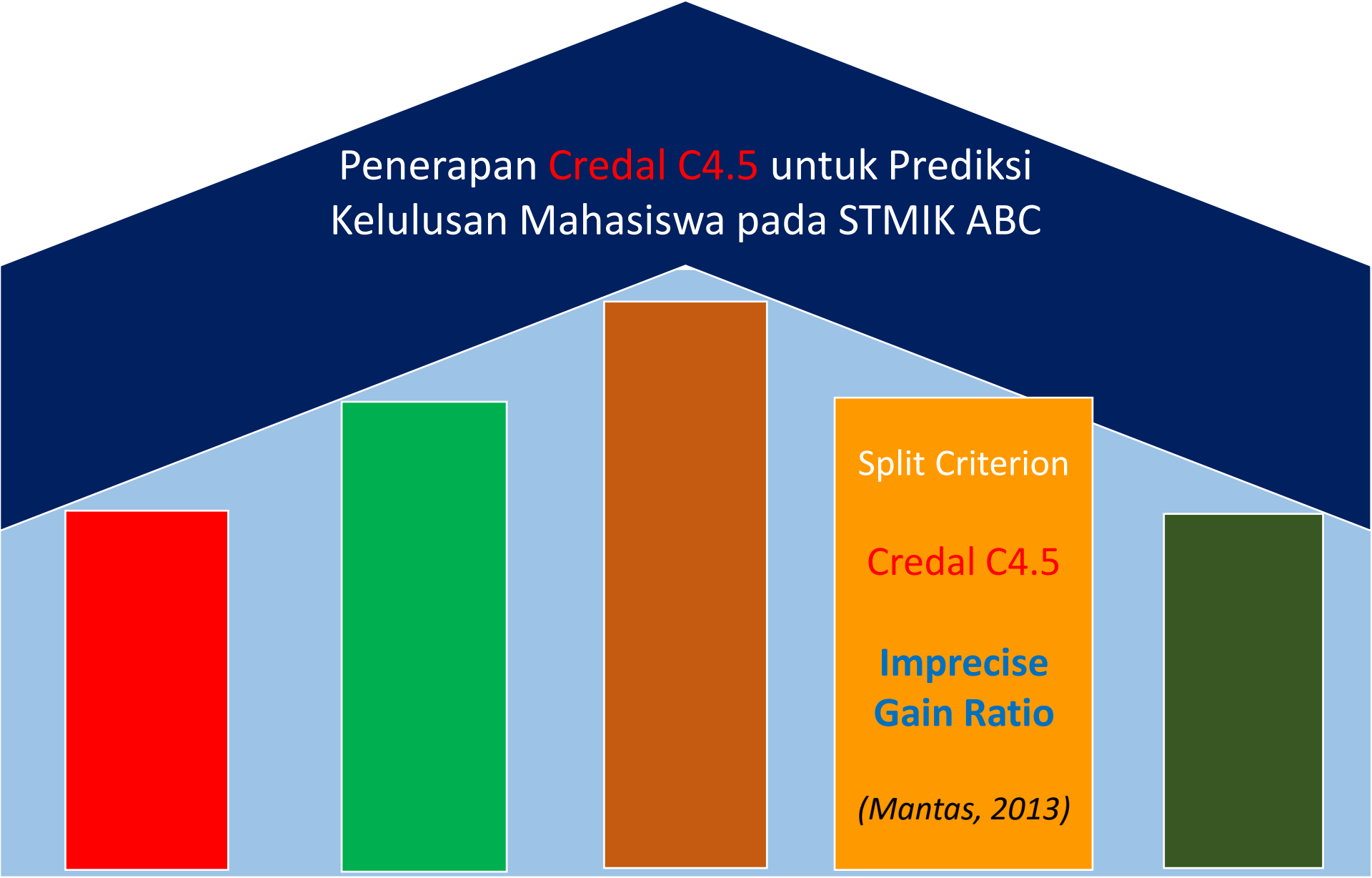


Penerapan C4.5 untuk Prediksi Kelulusan Mahasiswa pada STMIK ABC



Teori Gain *(Kullback & Leibler, 1951)*

Penerapan Credal C4.5 untuk Prediksi Kelulusan Mahasiswa pada STMIK ABC



Imprecise Probability Theory *(Walley, 1996)*

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Memperbaiki C4.5

Credal-C4.5: Decision tree based on imprecise probabilities to classify noisy data

Carlos J. Mantas, Joaquín Abellán *

Department of Computer Science & Artificial Intelligence, University of Granada, ETSI Informática, c/Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Memperbaiki Use Case Points

Simplifying effort estimation based on Use Case Points ☆

M. Ochodek *, J. Nawrocki, K. Kwarciak

Poznan University of Technology, Institute of Computing Science, ul. Piotrowo 2, 60-965 Poznań, Poland

Memperbaiki Genetic Algorithms

Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling

Shengxiang Yang, *Member, IEEE*, and Sadaf Naseem Jat

***Abstract*—The university course timetabling problem (UCTP) is a combinatorial optimization problem, in which a set of events has to be scheduled into time slots and located into suitable rooms. The design of course timetables for academic institutions is a very difficult task because it is an NP-hard problem. This paper investigates genetic algorithms (GAs) with a guided search strategy and local search (LS) techniques for the UCTP. The guided search strategy is used to create offspring into the population based on a data structure that stores information extracted from good individu-**

The research on timetabling problems has a long history of more than 40 years, starting with Gotlieb in 1962 [22]. Researchers have proposed various timetabling approaches by using graph coloring methods, constraint-based methods, population-based approaches (e.g., genetic algorithms (GAs), ant-colony optimization, and memetic algorithms), metaheuristic methods (e.g., tabu search (TS), simulated annealing (SA),

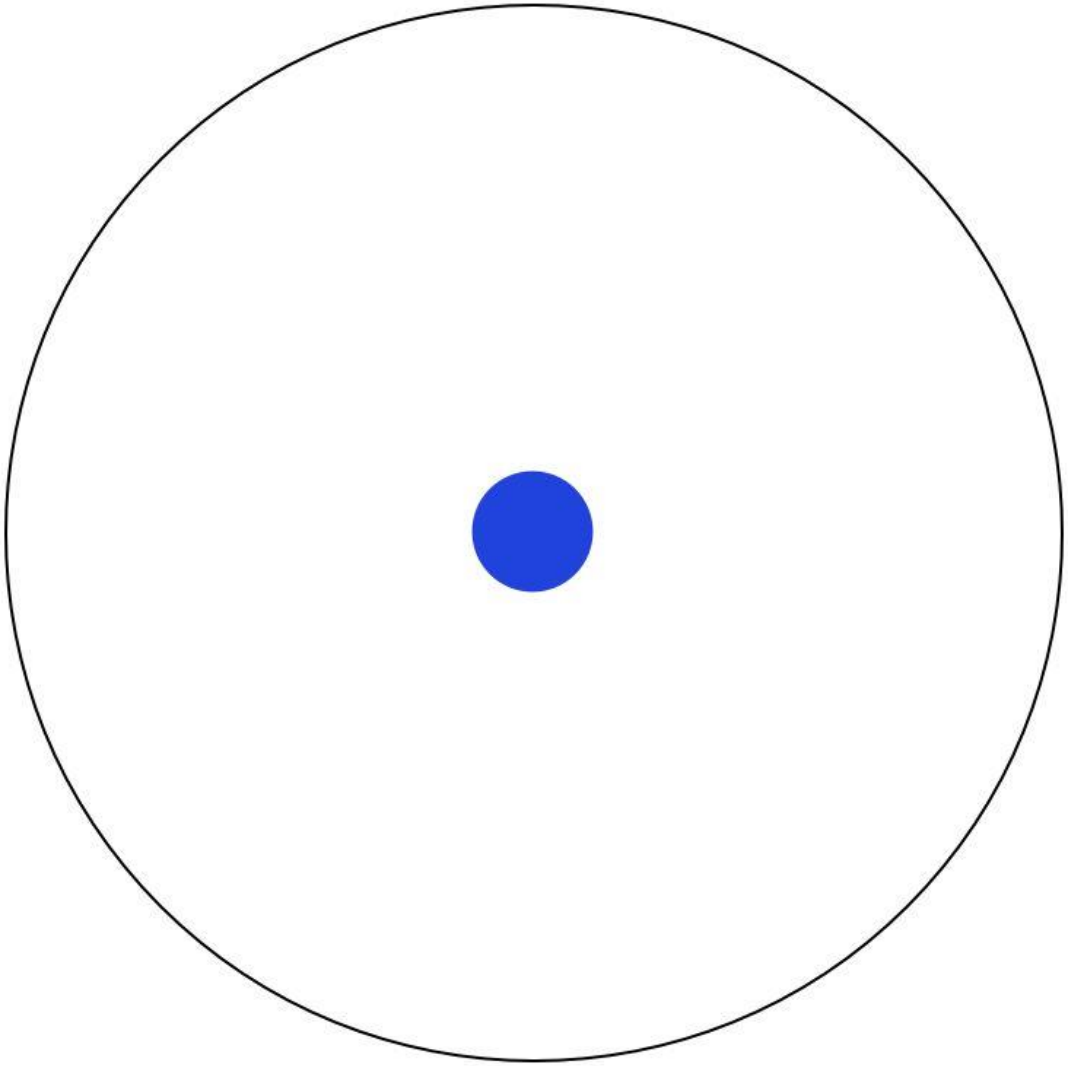
MITOS 8

Penelitian yang Baik itu Topik dan Skalanya Besar, serta Berhubungan dengan Banyak Bidang

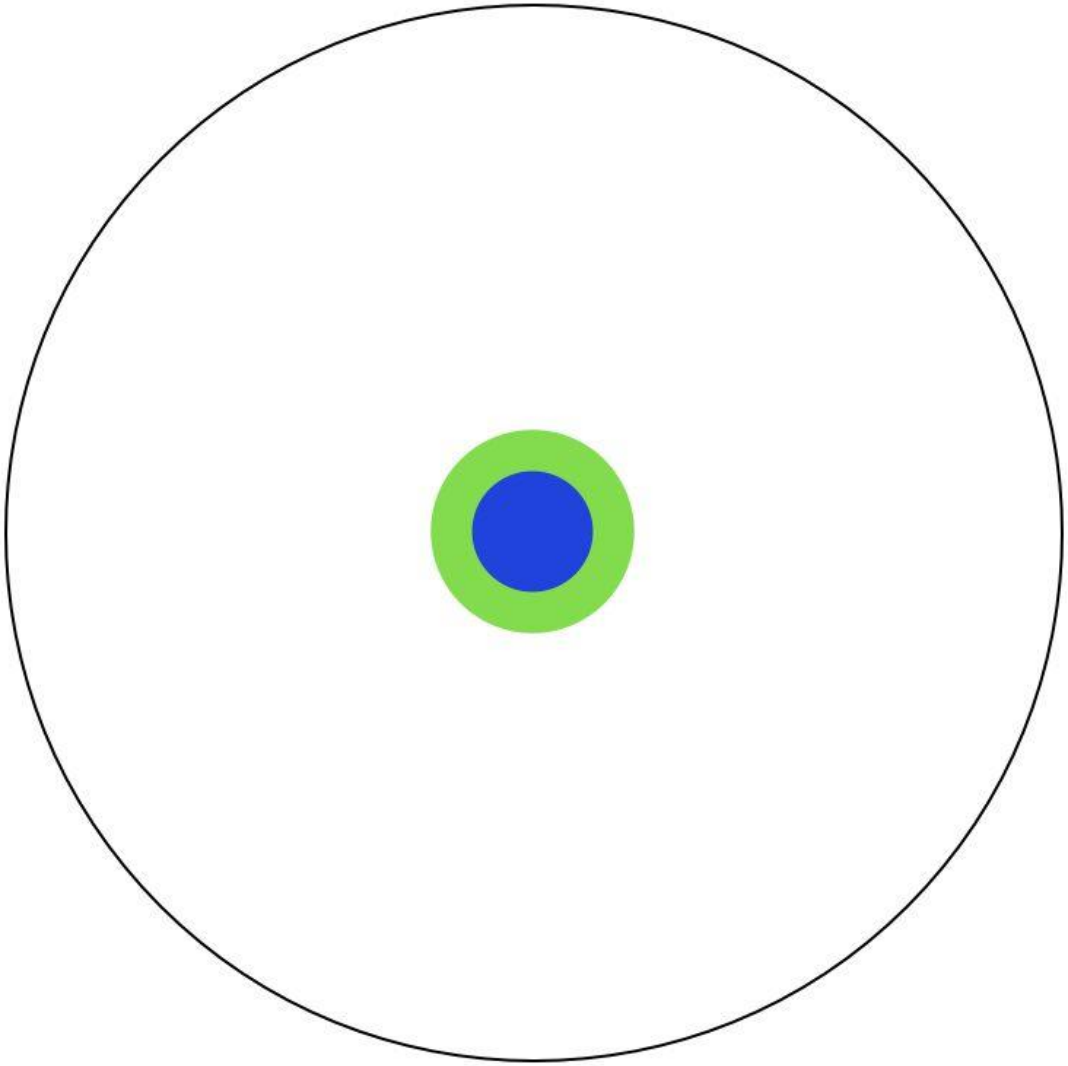
Penelitian yang Berkualitas Tinggi

Topik dan skalanya kecil, fokus, dalam, dan membawa pengaruh yang besar ke bidang penelitian kita

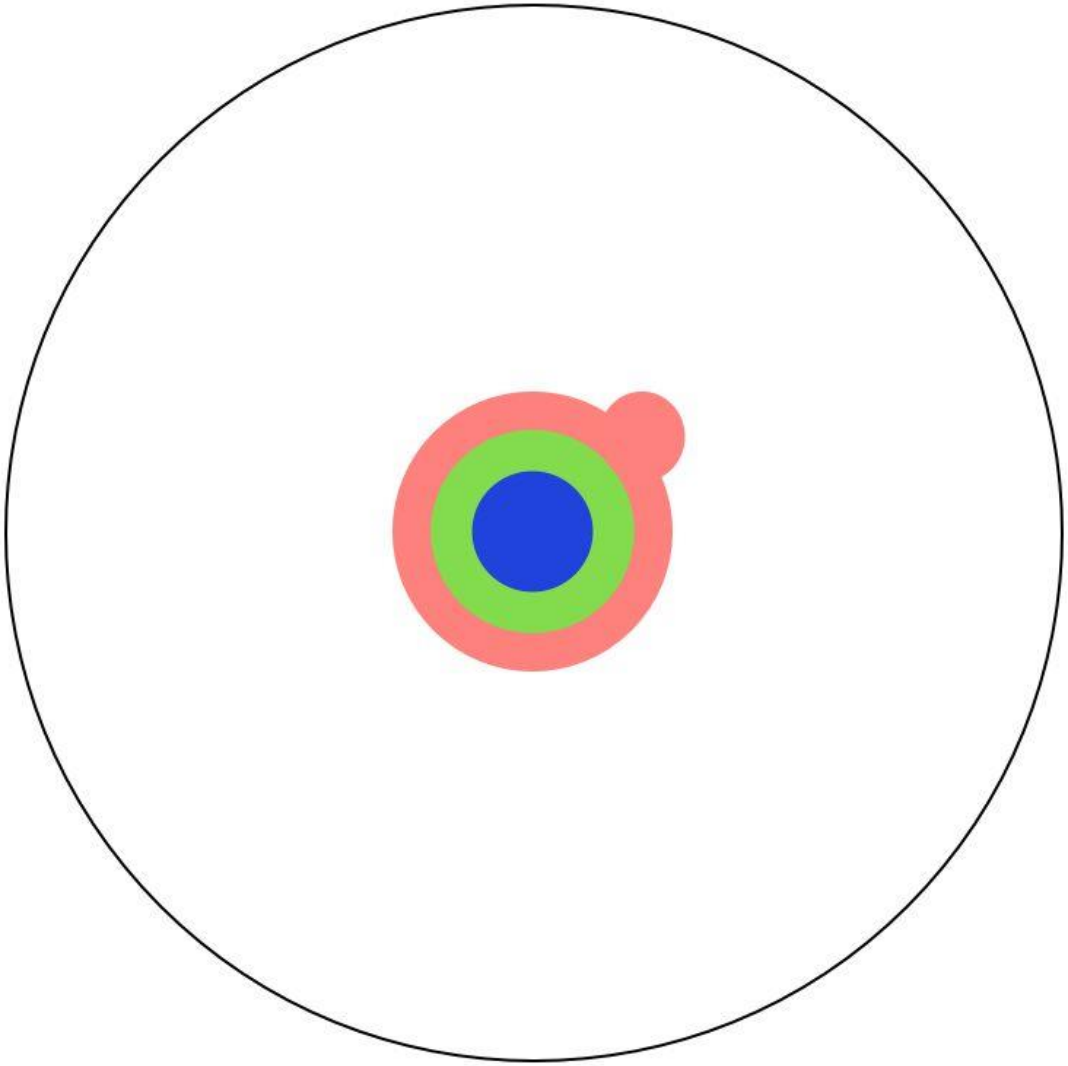
The Illustrated Guide to a Ph.D (Might, 2010)



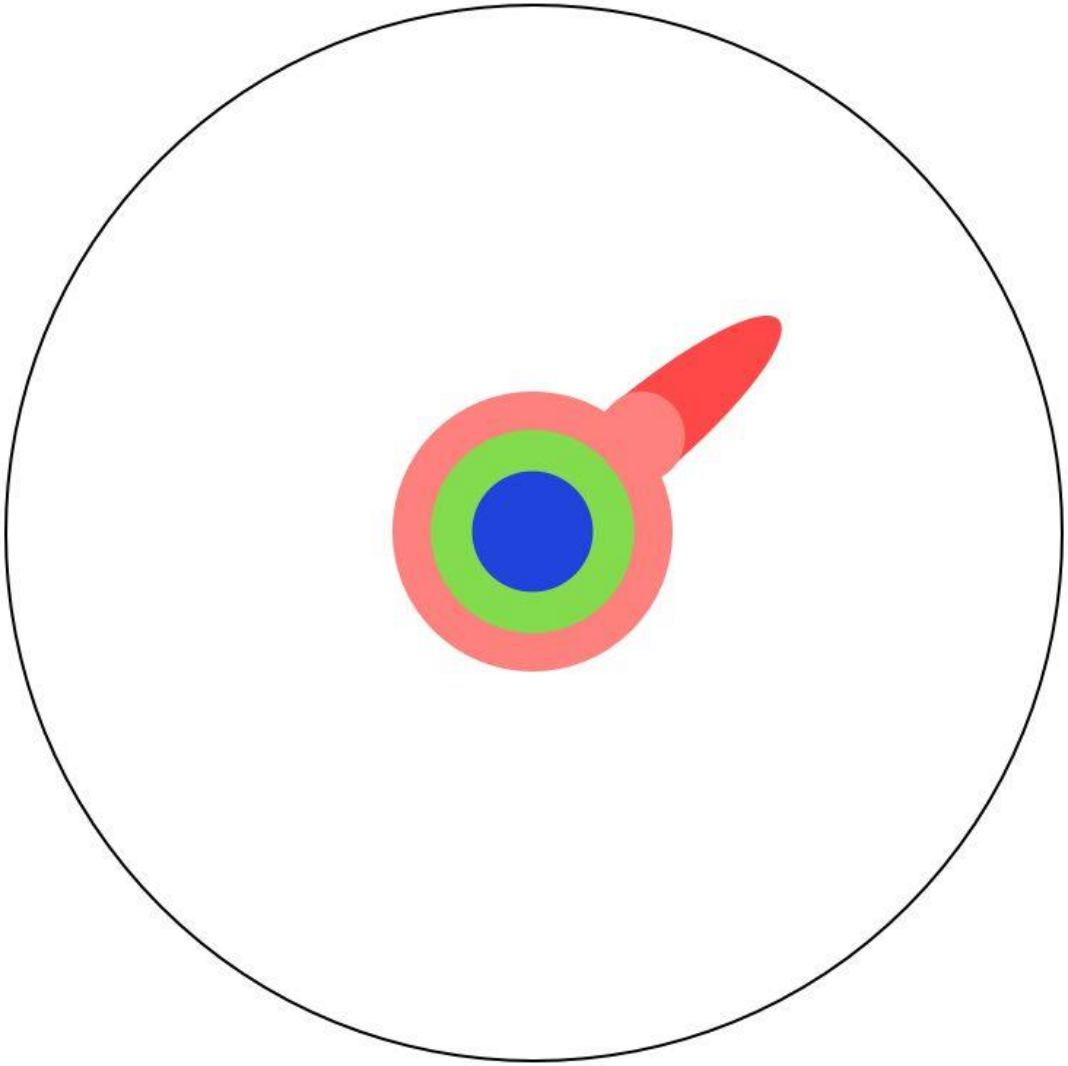
The Illustrated Guide to a Ph.D (Might, 2010)



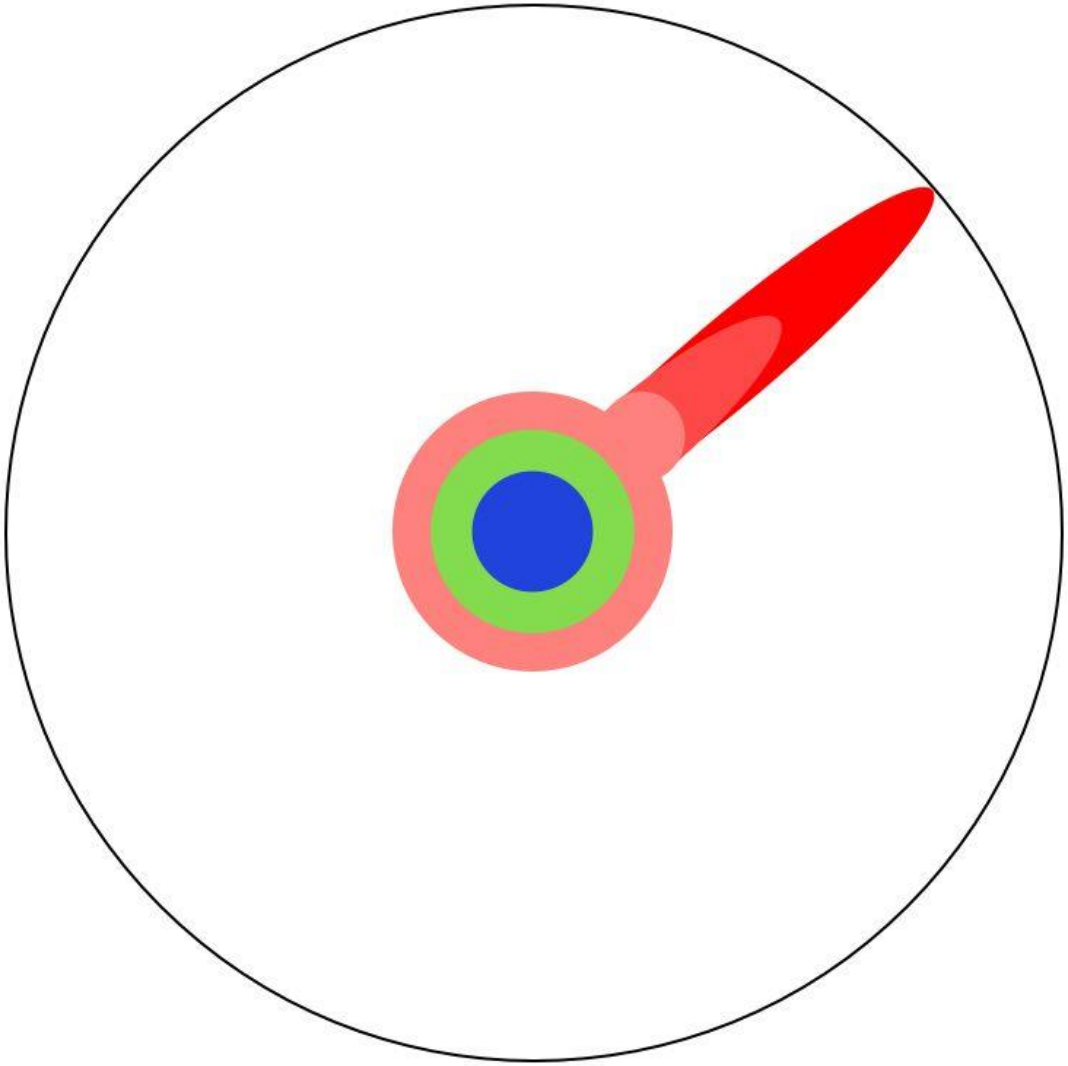
The Illustrated Guide to a Ph.D (Might, 2010)



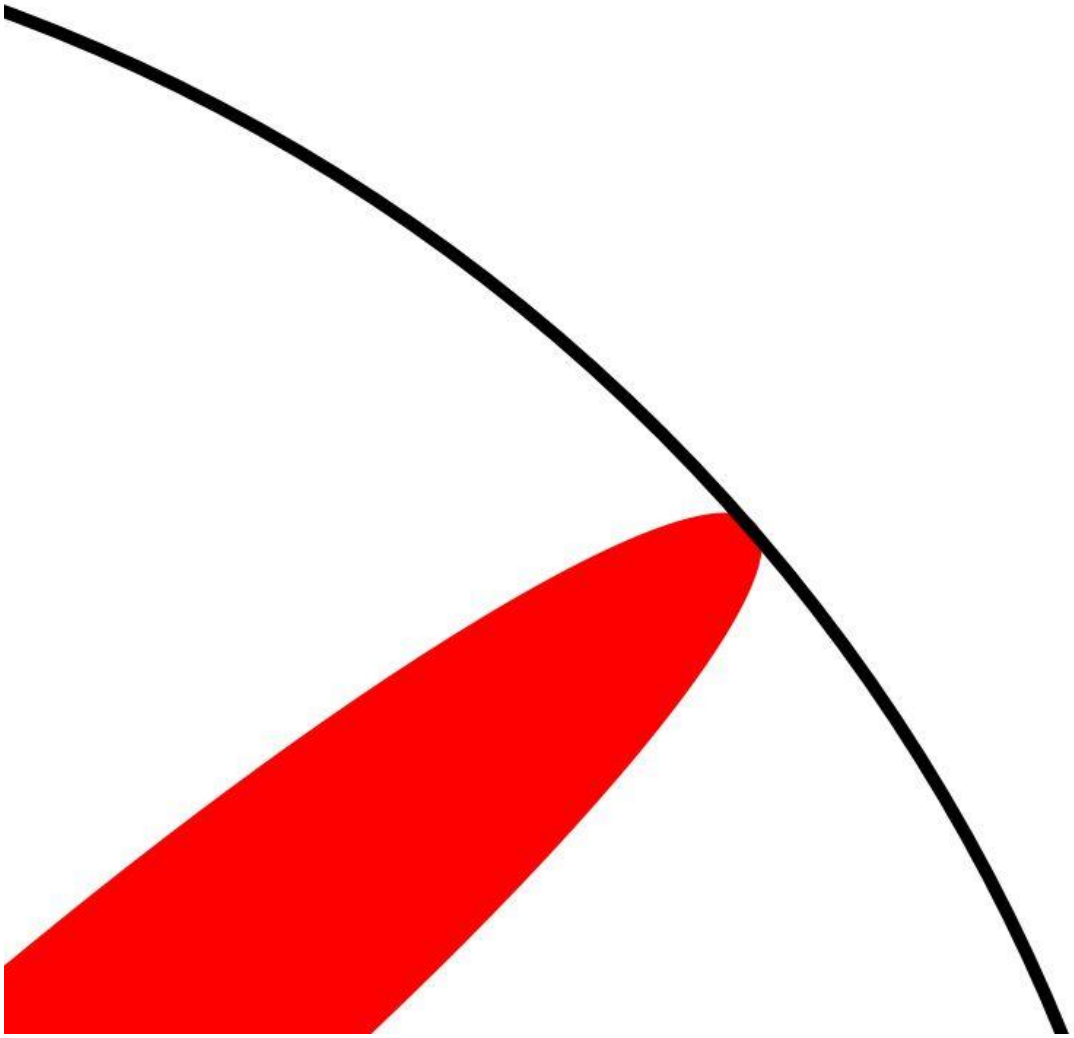
The Illustrated Guide to a Ph.D (Might, 2010)



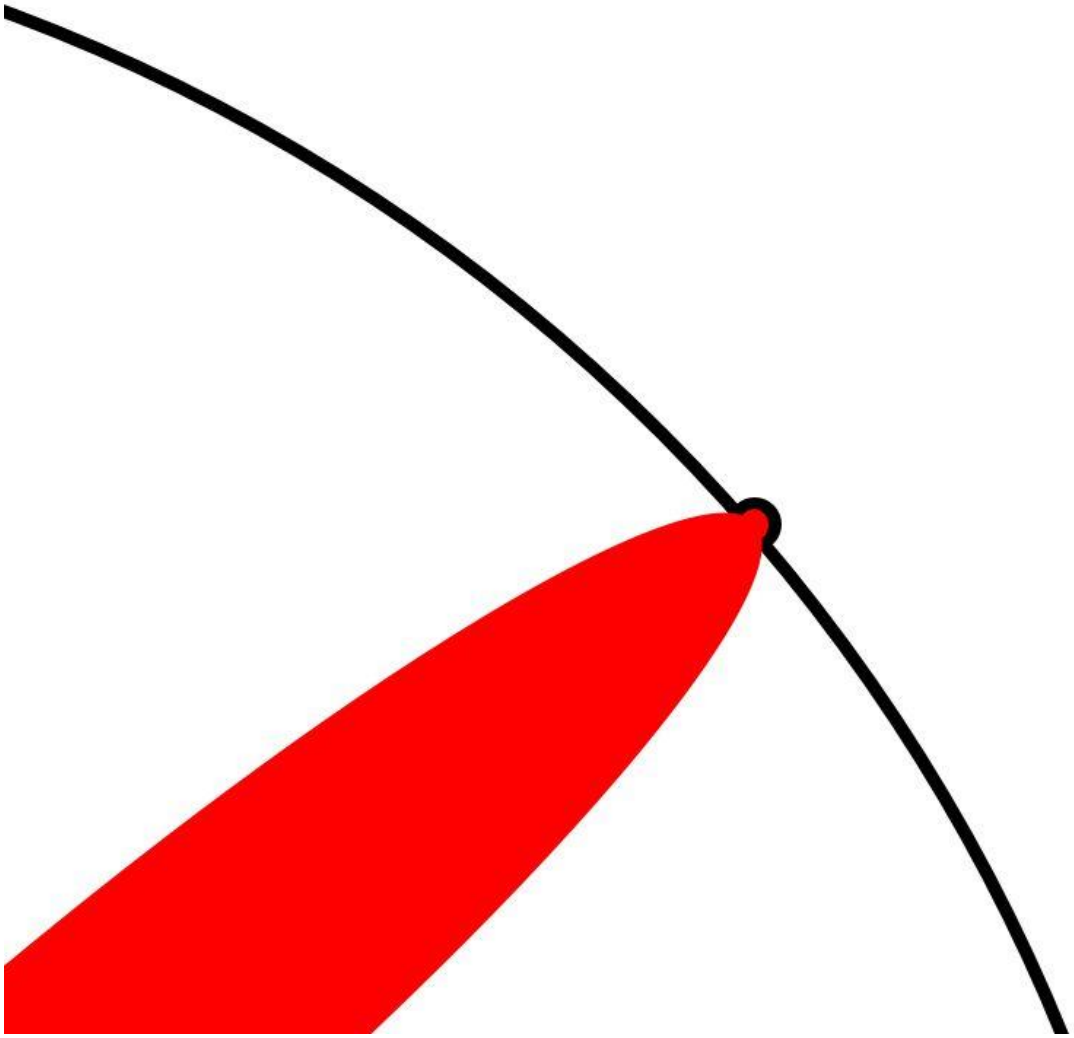
The Illustrated Guide to a Ph.D (Might, 2010)



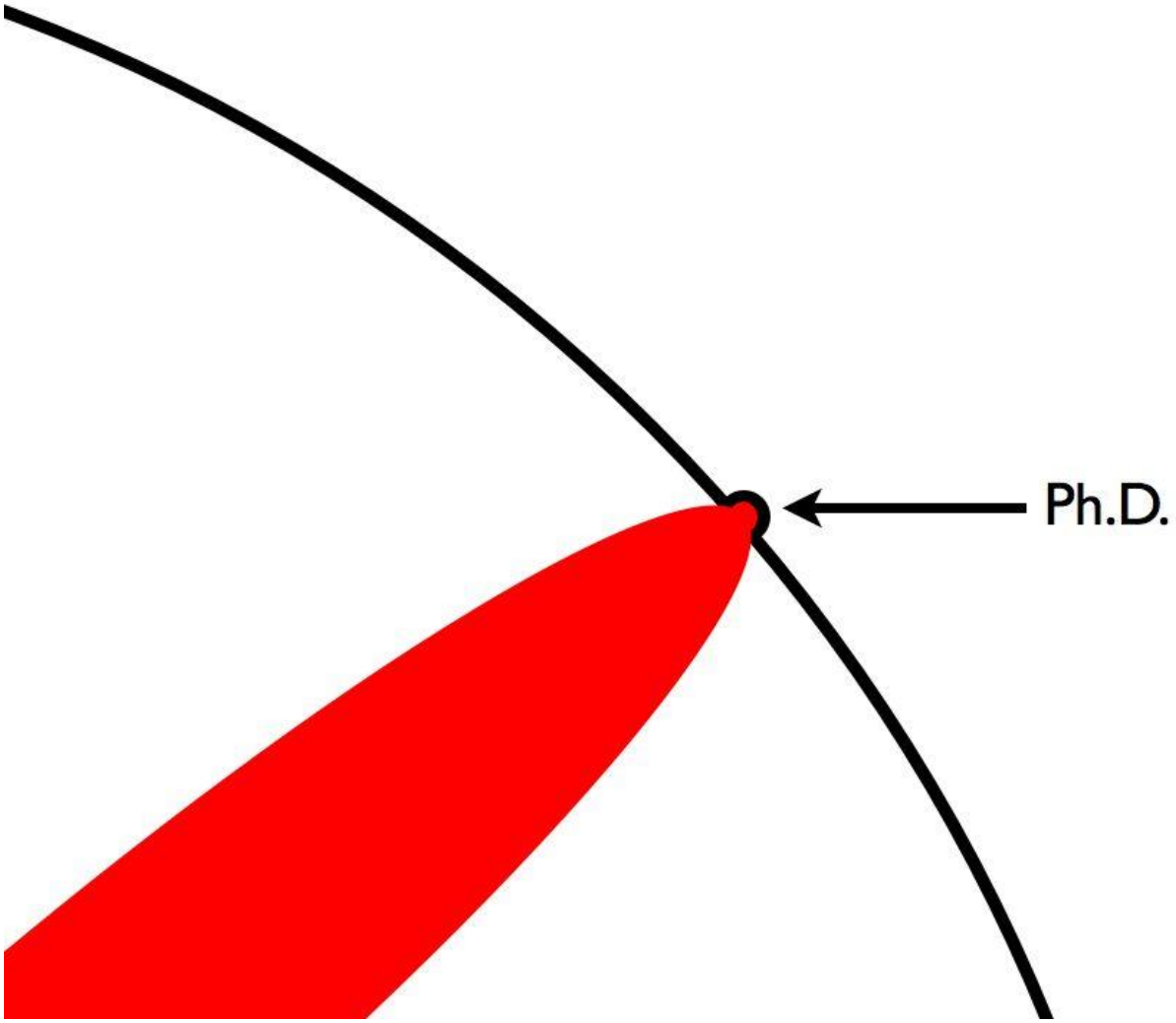
The Illustrated Guide to a Ph.D (Might, 2010)



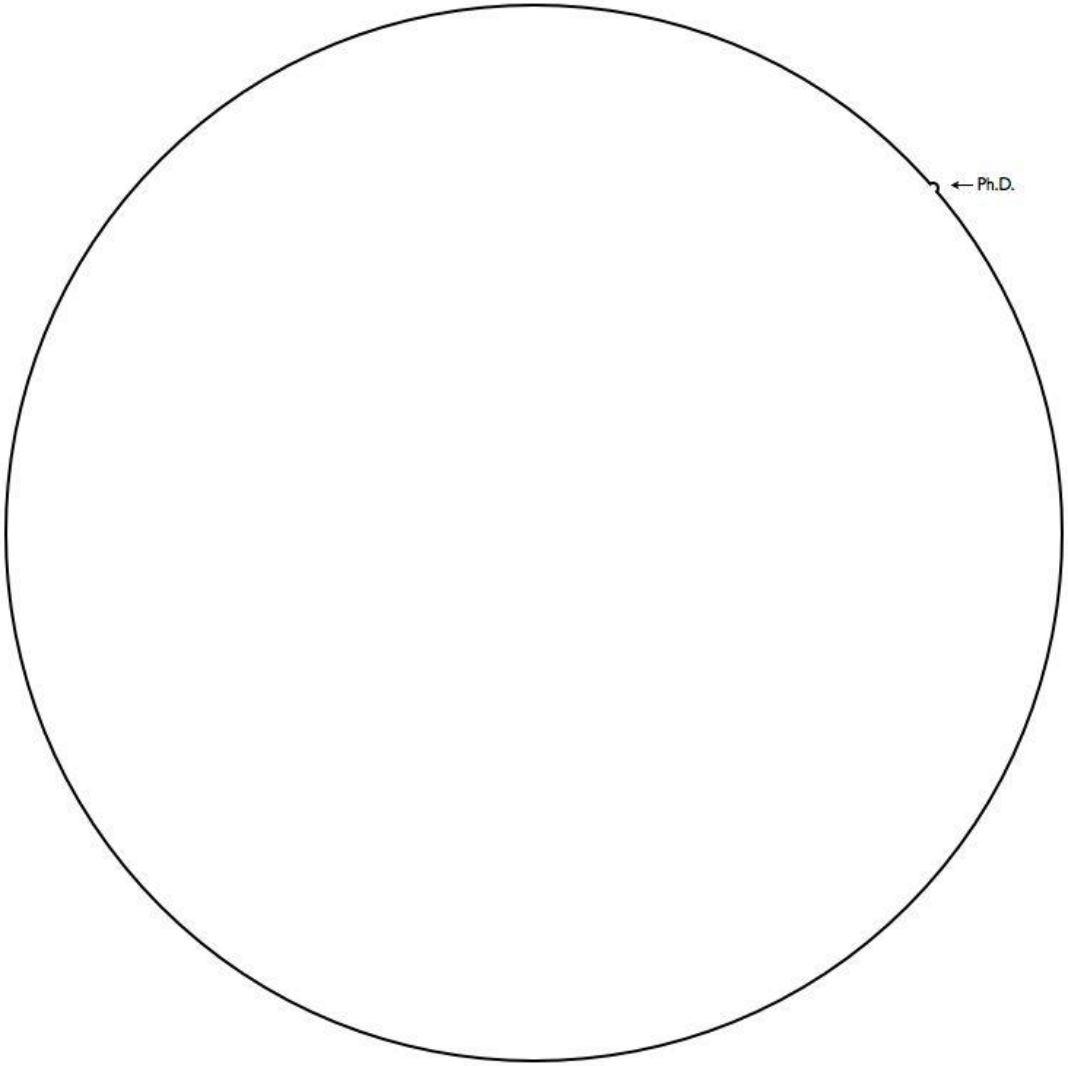
The Illustrated Guide to a Ph.D (Might, 2010)



The Illustrated Guide to a Ph.D (Might, 2010)



The Illustrated Guide to a Ph.D (Might, 2010)



MITOS 9

Saya Melakukan Citation dengan Meng-*Copy Paste* Kalimat dan Paragraf dari Paper Lain

Jenis *Citation*

1. Kutipan (Quotation): Kata-kata yang diambil persis sama dengan apa yang dituliskan (tanpa perubahan). Ditulis dalam tanda kutip
2. Paraphrase: Menyusun kembali pemikiran penulis dan mengungkapkannya dengan kata-kata sendiri
3. Ringkasan: Sari dari suatu tulisan
4. Evaluasi: Interpretasi dalam bentuk komentar, baik setuju atau tidak dengan menyebutkan alasannya

(Beast & Kohn, 1998)

Konsep Dasar Penulisan

- Kutipan itu tidak berarti bahwa satu paragraf kita copy-paste. Praktek seperti ini tetap disebut plagiarism meskipun referensi disebutkan
- Kutipan hanya untuk hal penting (hasil penelitian, teori, data, model, definisi) dalam paper
- Segala kalimat yang tidak merujuk atau menunjuk ke kutipan, berarti adalah tulisan karya sendiri
- Daftar referensi bukan daftar bacaan, tapi daftar rujukan atau kutipan (dibaca langsung, bukan dari penulis ketiga)

Mensitasi Sitasi Orang Lain

- Mensitasi (mengutip) hasil rangkuman dan kutipan yang dilakukan orang lain di buku atau papernya
  - Definisi logika fuzzy menurut Lotfie Zadeh dalam Suyanto (Suyanto, 2009) adalah: blablabla

- Jangan terlalu banyak dilakukan kecuali dalam keadaan:
  - Kita tidak bisa mengakses publikasi asli
  - Bahasa asli publikasi bukan bahasa inggris (sulit dipahami)

MITOS 10

Satu Hasil Eksperimen Penelitian Bisa Jadi Banyak Paper dan Dipublikasikan di Banyak Jurnal

Gap Analysis of Framework

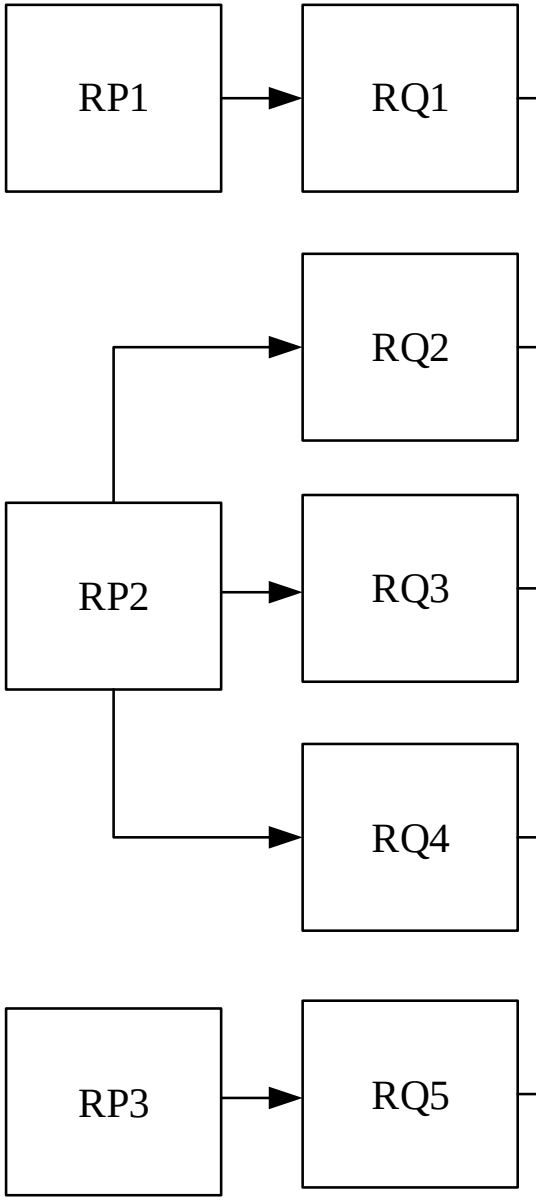
1. The comparisons and benchmarking result of the defect prediction using machine learning classifiers indicate that:
   - Poor accuracy level is dominant (Lessmann et al. 2008)
   - No significant performance differences could be detected (Lessmann et al. 2008)
   - No particular classifiers that performs the best for all the data sets (Song et al. 2011) (Hall et al. 2012)
2. Noisy attribute predictors and imbalanced class distribution of software defect datasets result in inaccuracy of classification models
3. Neural network and support vector machine have strong fault tolerance and strong ability of nonlinear dynamic processing of software fault data, but practicability of neural network and support vector machine are limited due to difficulty of selecting appropriate parameters

Research Problems (RP)

RP1 While many studies on software defect prediction report the comparative performance of the classification algorithms used, but there is no strong consensus on which classifiers perform best when individual studies are looked separately

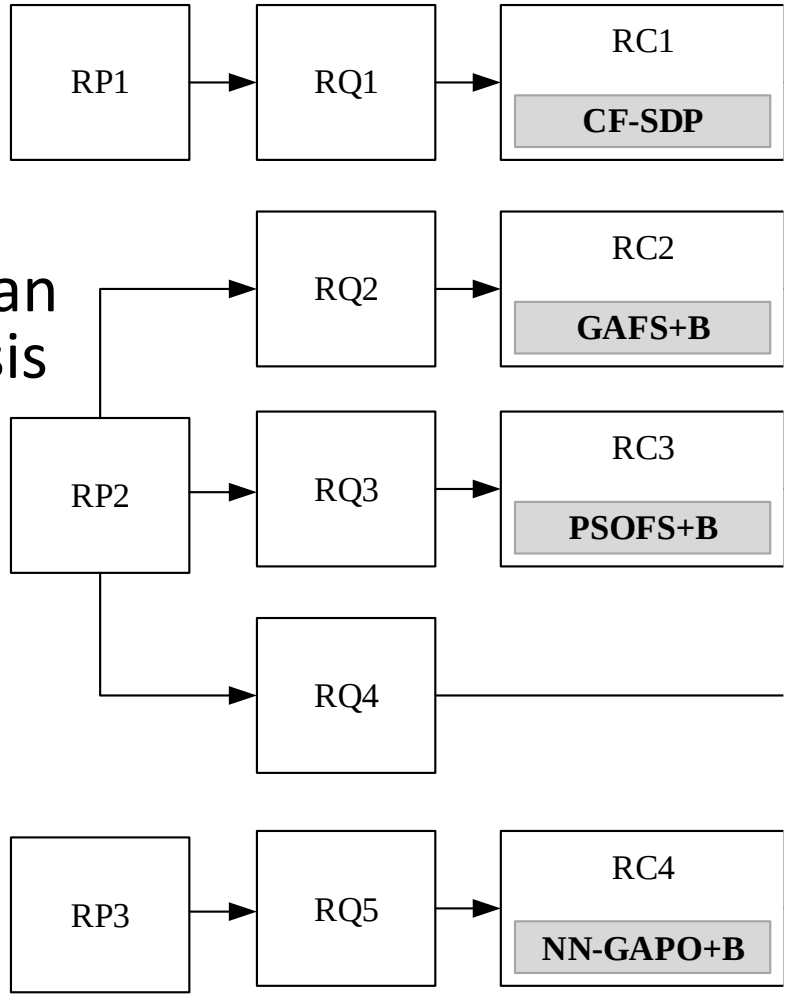
RP2 Noisy attribute predictors and imbalanced class distribution of software defect datasets result in inaccuracy of classification models

RP3 Neural network has strong fault tolerance and strong ability of nonlinear dynamic processing of software fault data, but practicability of neural network is limited due to difficulty of selecting appropriate parameters

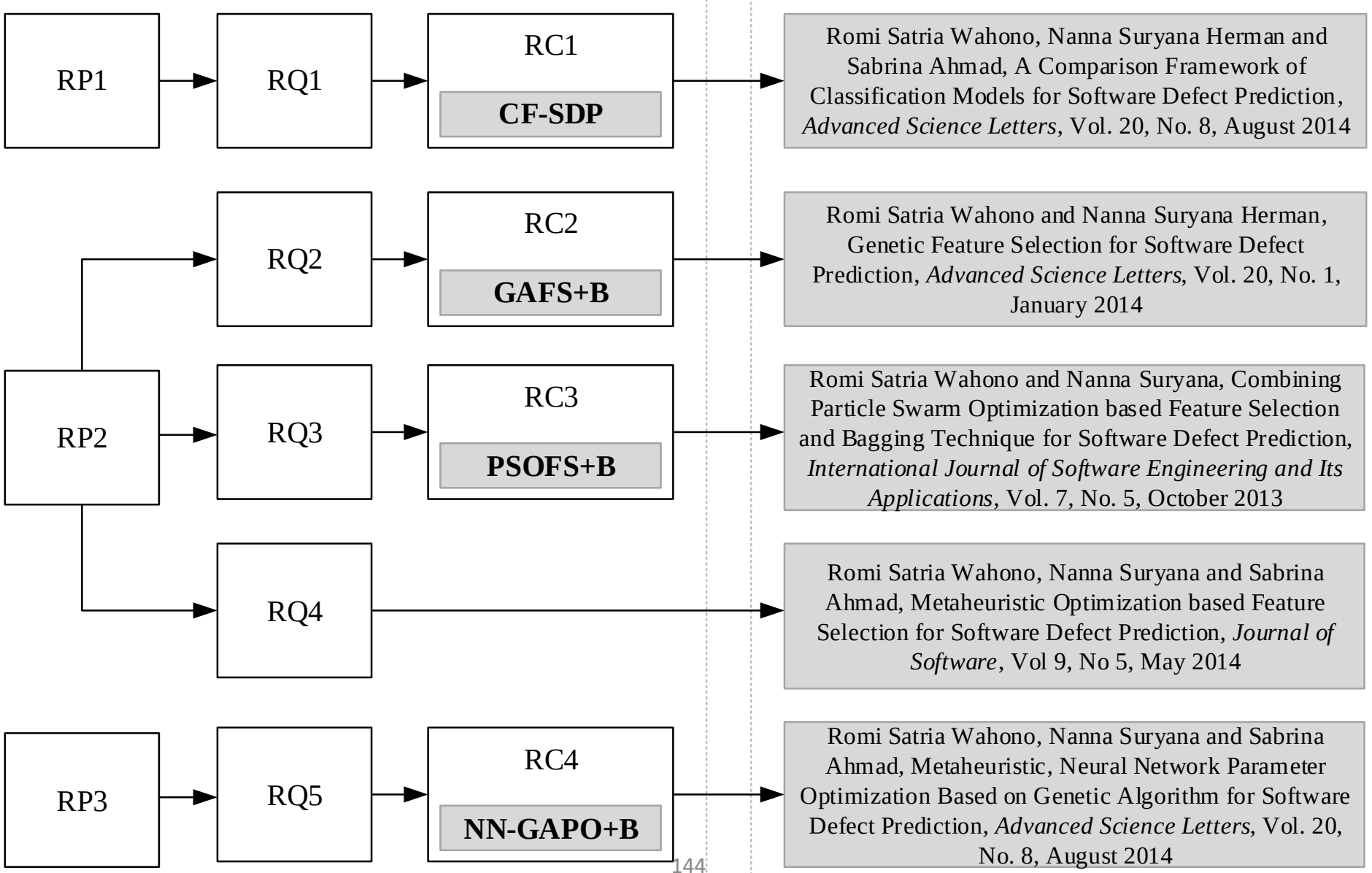


Pola RP – RQ – RC pada Penelitian

- Research Problem (RP) atau masalah penelitian adalah alasan kita melakukan penelitian
- Satu RP bisa coba dipecahkan dengan banyak cara/metode/solusi/hipotesis (Research Question (RQ))
- Satu RQ akan membentuk satu kontribusi ke pengetahuan (Research Contribution (RC))
- Satu RC akan menjadi satu paper publikasi



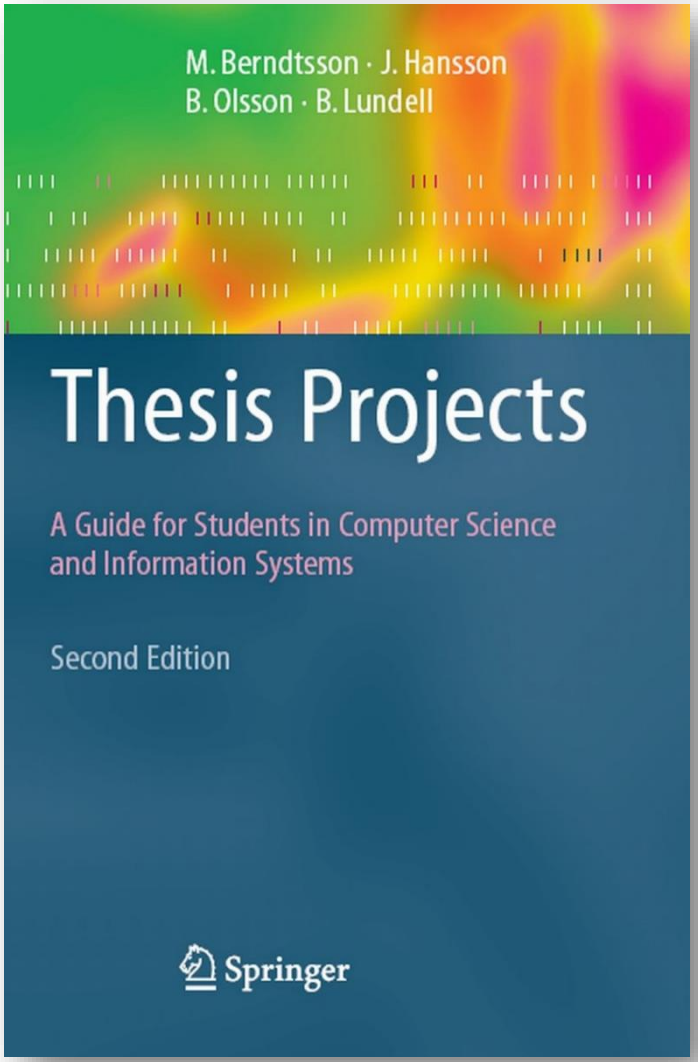
Software Defect Prediction Framework based on Hybrid Metaheuristic Optimization Methods



Terima Kasih

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

Reference



Reference

- Abbott, M., & McKinney, J. (2013). Understanding and Applying Research Design. John Wiley & Sons, Inc.
- Berndtsson, M., Hansson, J., & Olsson, B. (2008). Thesis Projects: a Guide for Students in Computer Science and Information Systems (2nd ed.). London: Springer-Verlag
- Blaxter, L., Hughes, C., & Tight, M. (2006). How to Research (3rd ed.). Open University Press
- Blessing, L. T. M., & Chakrabarti, A. (2009). DRM, a Design Research Methodology. Springer-Verlag London
- Cohen, L., Manion, L., & Morrison, K. (2005). Research Methods in Education (5th ed.). Taylor & Francis Group.
- Dawson, C. W. (2009). Projects in Computing and Information Systems A Student's Guide (2nd ed.). Pearson Education Limited
- Jonker, J., & Pennink, B. (2010). The Essence of Research Methodology. Springer-Verlag Berlin Heidelberg
- Lichtfouse, E. (2013). Scientific Writing for Impact Factor Journals. Nova Science Publishers, Inc.

Reference

- Kothari, C. (2004). Research Methodology: Methods and Techniques. New Age International
- Might, M. (2010). The Illustrated Guide to a Ph.D. Matt.might.net. Retrieved from *http://matt.might.net/articles/phd-school-in-pictures/*
- Marczyk, G., DeMatteo, D., & Fertinger, D. (2005). Essentials of Research Design and Methodology. John Wiley & Sons, Inc.
- Rea, L. M., & Parker, R. A. (2014). Designing and Conducting Survey Research: A Comprehensive Guide (4th ed.). John Wiley & Sons, Inc.
- Runeson, P., Host, M., Rainer, A., & Regnell, B. (2012). Case Study Research in Software Engineering: Guidelines and Examples. John Wiley & Sons, Inc.
- Sahu, P. K. (2013). Research Methodology: A Guide for Researchers In Agricultural Science, Social Science and Other Related Fields. Springer.
- Veit, R., Gould, C., & Gould, K. (2013). Writing, Reading, and Research (9th ed.). Cengage Learning.