10 MITOS

Penelitian Computing

Romi Satria Wahono
romi@romisatriawahono.net
http://romisatriawahono.net
081586220090

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)
- Research Interests in Software Engineering and Machine Learning
- LIPI Researcher (2004-2007)
- Founder and CEO:
  - PT IlmuKomputerCom Braindevs Sistema
  - PT Brainmatics Cipta Informatika
- Professional Member of IEEE, ACM and PMI
- IT and Research Award Winners from WSIS (United Nations), Kemdikbud, LIPI, etc
- SCOPUS/ISI Indexed Journal Reviewer: Information and Software Technology, Journal of Systems and Software, Software: Practice and Experience, etc
- Industrial IT Certifications: TOGAF, ITIL, CCNA, etc
- Enterprise Architecture Consultant: KPK, Ristek Dikti, LIPI, DJPK Kemenkeu, Kemsos, INSW, UNSRI, etc

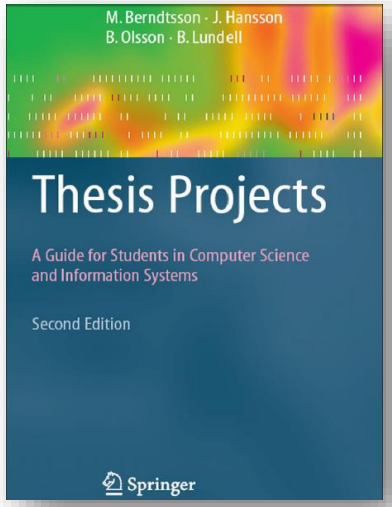
MITOS 1

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)

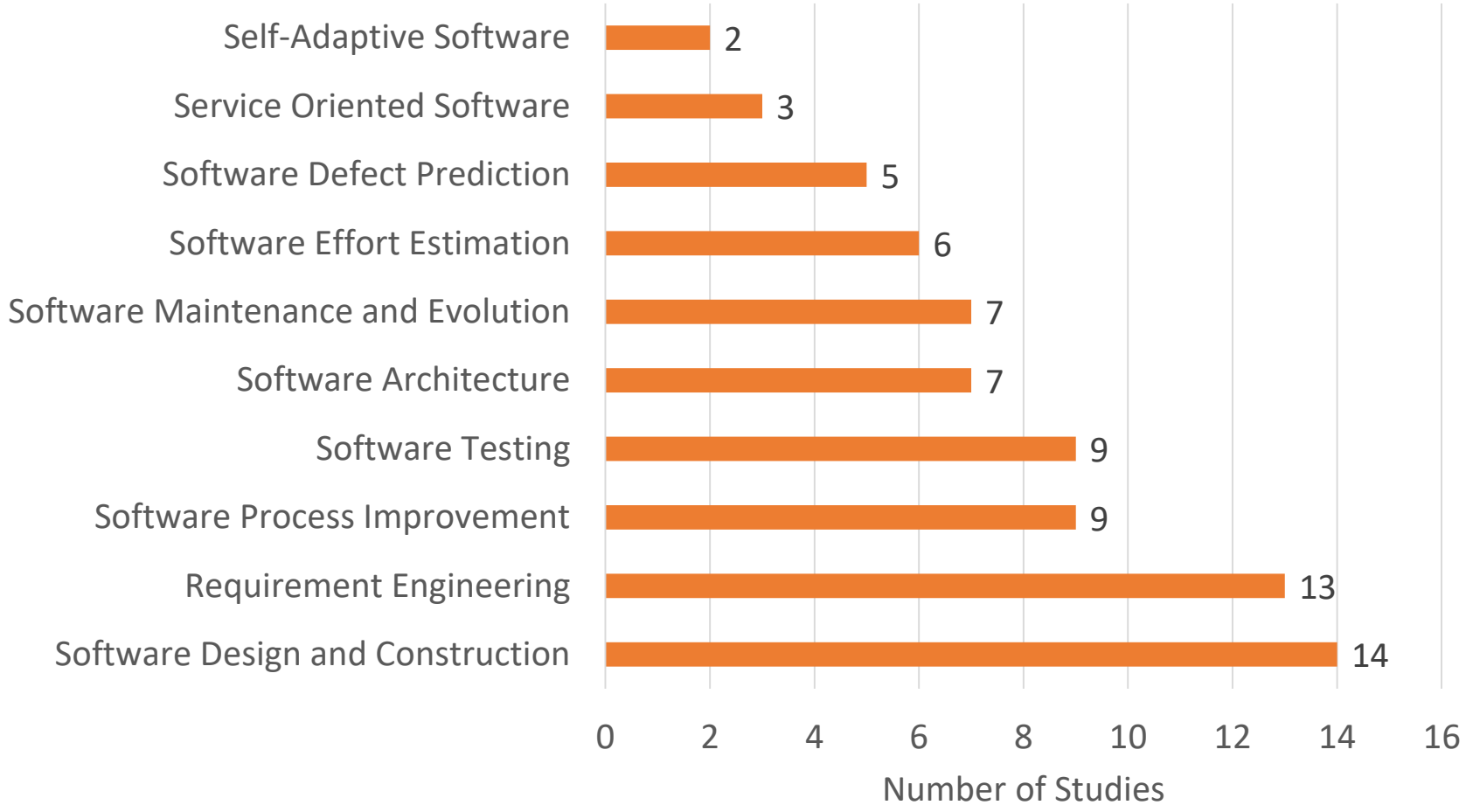
Pemahaman Penelitian yang Benar

- Membangun software bukanlah tujuan utama penelitian, hanya *testbed* untuk mempermudah kita dalam mengukur hasil penelitian

- Ketika kita mengusulkan perbaikan suatu algoritma (*proposed method*)
  - Untuk mempermudah eksperimen dan evaluasi, kita menulis kode program untuk menguji efisiensi algoritma yang kita usulkan

Penelitian Software Engineering?

Penelitian software engineering bukan ke arah pengembangan software, tapi ke metodologi pengembangan software



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

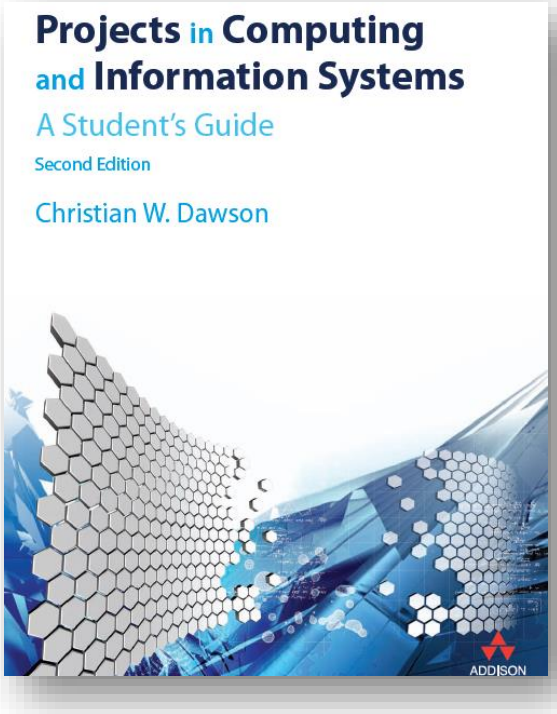
MITOS 2

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)

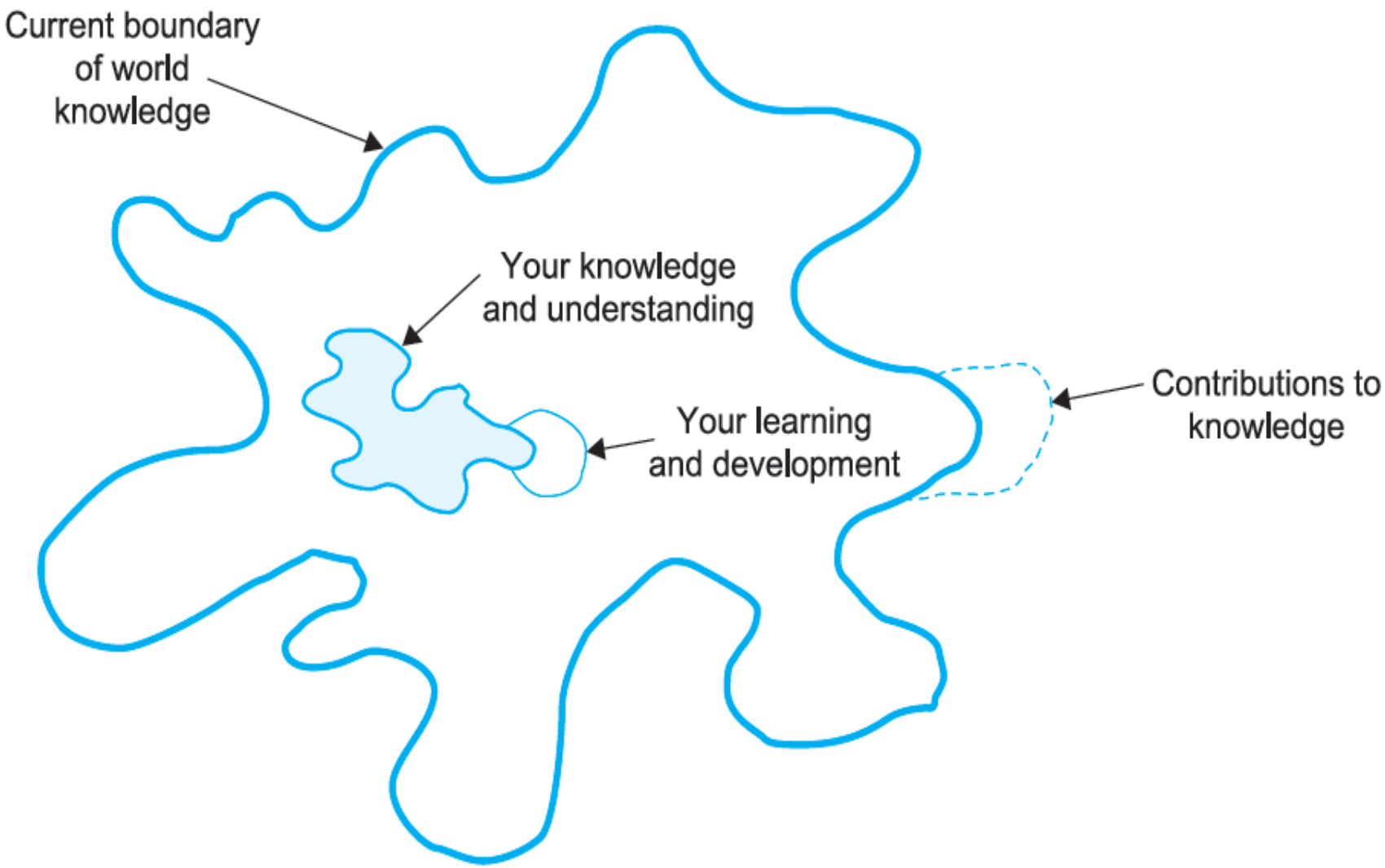


Bentuk Kontribusi ke Pengetahuan

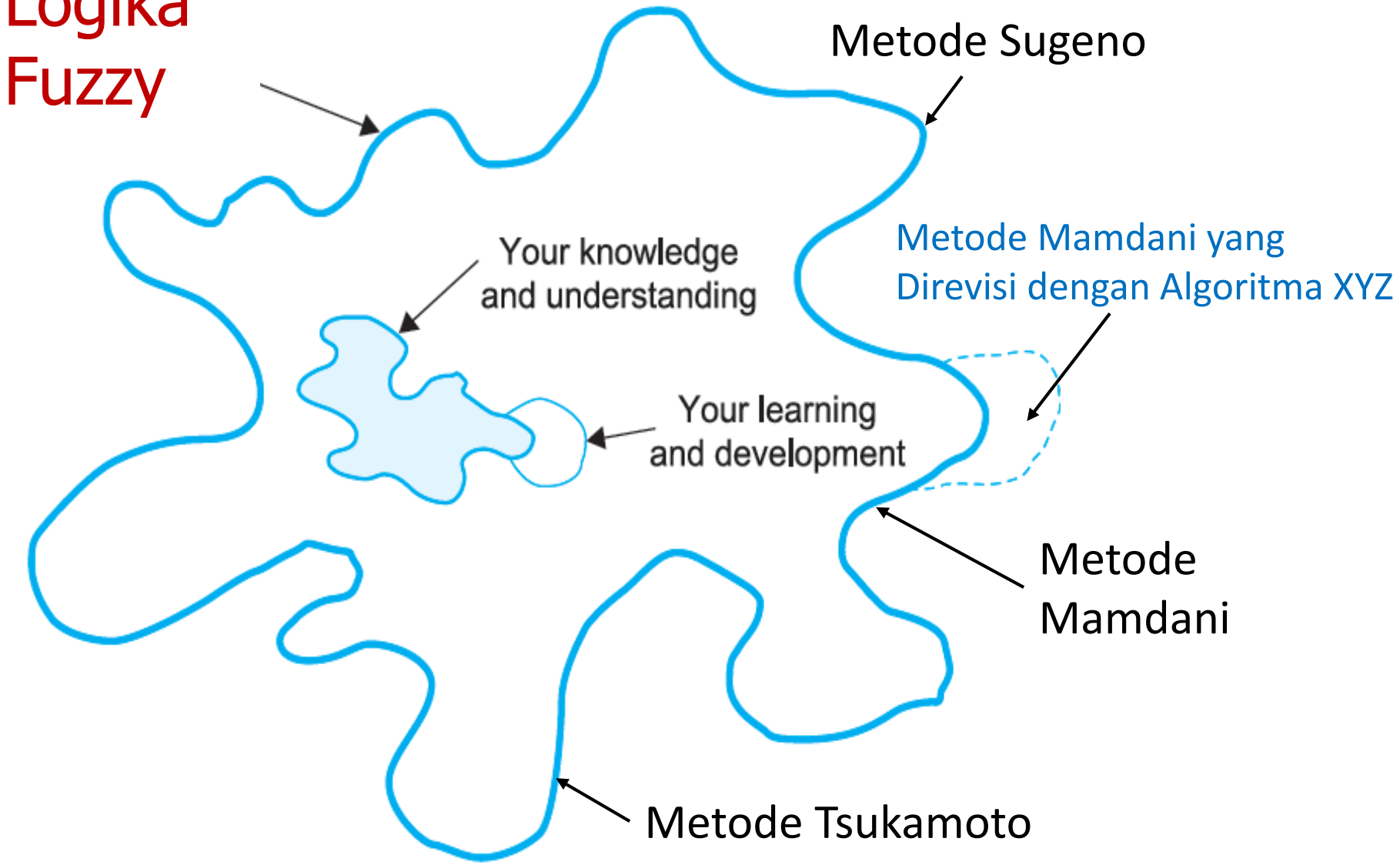
Kegiatan penyelidikan dan investigasi terhadap suatu masalah yang dilakukan secara berulang-ulang dan sistematis, dengan tujuan untuk menemukan atau merevisi teori, metode, fakta, dan aplikasi

(Berndtsson et al., 2008)

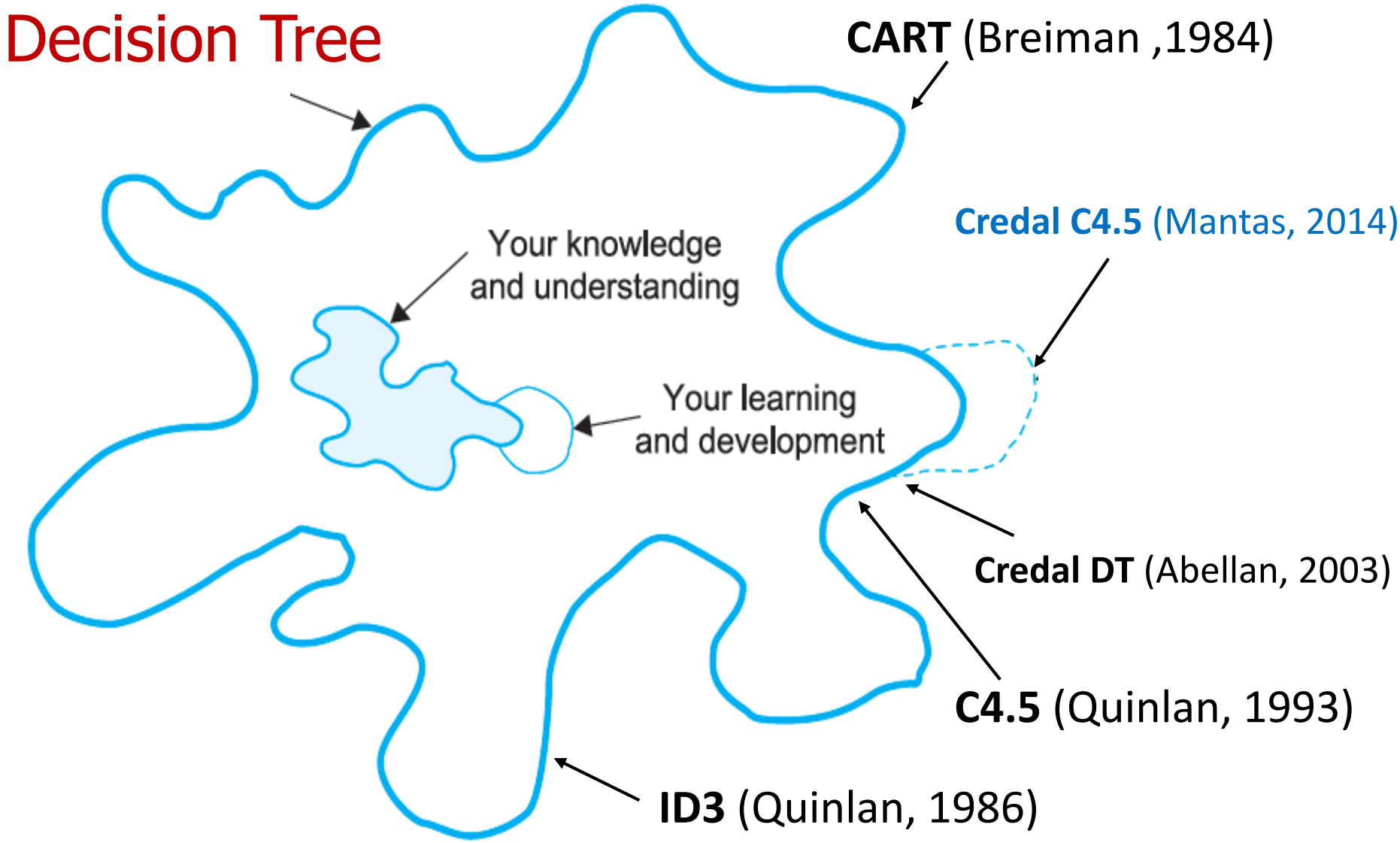
Bentuk Kontribusi ke Pengetahuan



Bentuk Kontribusi ke Pengetahuan



Bentuk Kontribusi ke Pengetahuan



Orisinalitas Penelitian

1. Orisinalitas pada Metode:
   - Memecahkan masalah yang orang lain sudah pernah mengerjakan sebelumnya, tapi dengan metode yang berbeda
   - Model penelitian yang kontribusi ada pada method improvement

2. Orisinalitas pada Masalah:
   - Memecahkan suatu masalah yang orang lain belum pernah mengerjakan sebelumnya
   - Model penelitian yang kontribusi ada pada penemuan masalah baru sebagai obyek penerapan metode

(Dawson, 2009)

Contoh Tema Penelitian

Algoritma Genetika untuk
Penentuan Desain Bendungan
yang Paling Optimal

Contoh Kontribusi pada Metode

- 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 Kontribusi pada Masalah

- Judul:

  Penerapan Algoritma Genetika untuk Penentuan Desain Bendungan dengan Tujuh Parameter

- Kontribusi: Penentuan Desain Bendungan dengan Tujuh Parameter (kebanyakan peneliti menggunakan tiga parameter)

Contoh Kontribusi pada Masalah dan Metode

- Judul:

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

- Kontribusi:

  1. Penerapan metode XYZ untuk memecahkan masalah konvergensi premature pada algoritma genetika
  2. Penentuan Desain Bendungan dengan Tujuh Parameter

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 di Indonesia yang terjebak dengan penelitian tanpa kontribusi dan hanya mengganti obyek tempat, akhirnya kesulitan ketika harus publikasi ke journal 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

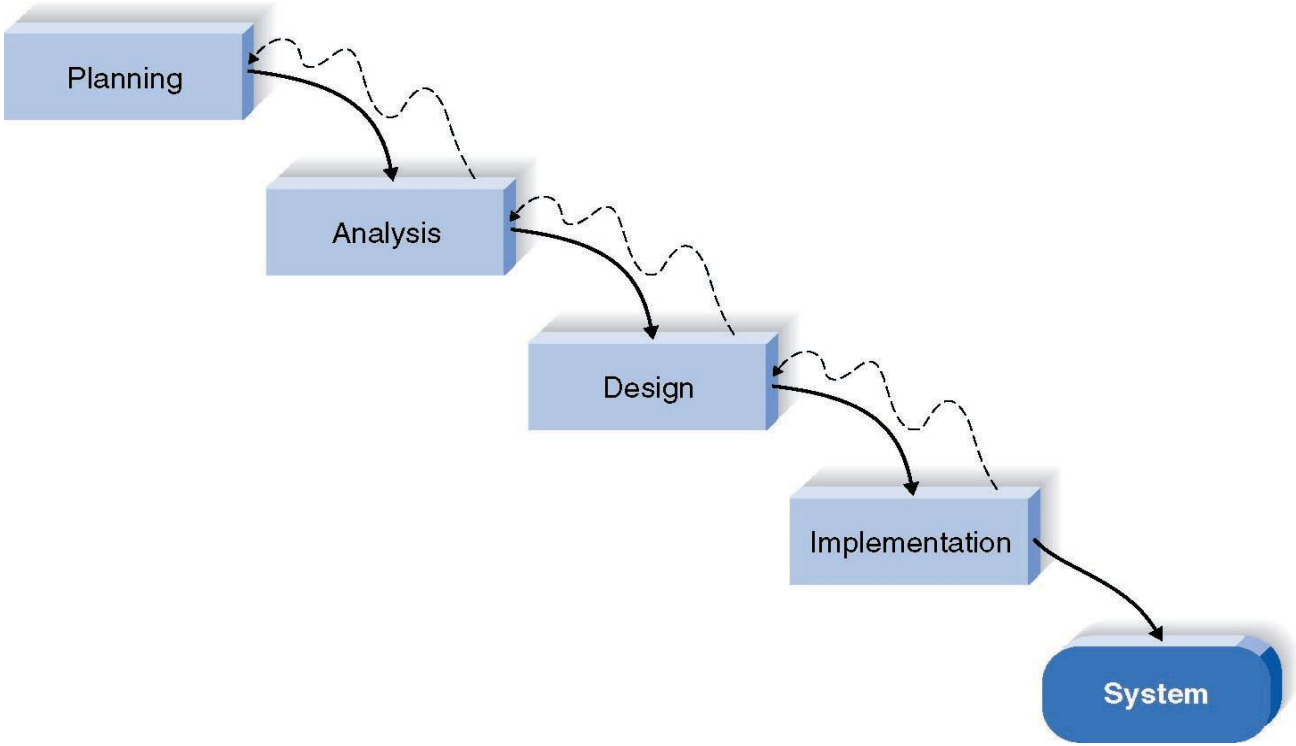
Aspek	Tugas Akhir (D3/D4)	Skripsi (S1)	Tesis (S2)	Disertasi (S3)
Level Kontribusi	Penguasaan Kemampuan Teknis	Pengujian Teori	Pengembangan Teori	Penemuan Teori Baru
Bentuk Kontribusi	Implementasi dan pengembangan	Implementasi dan pengembangan	Perbaikan Secara Inkremental dan Terus Menerus	Substansial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

Pemahaman Penelitian yang Benar

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

MITOS 3

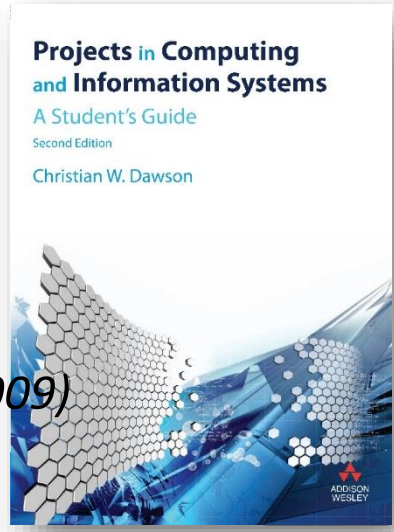
Waterfall adalah Metode Penelitian yang Saya Gunakan



Metode Penelitian

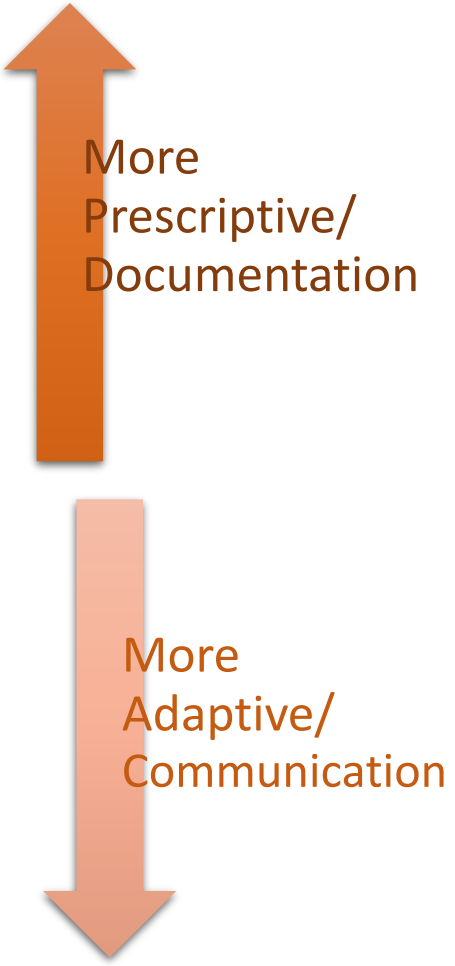
1. Penelitian Tindakan
   - Studi berupa monitoring dan pencatatan penerapan sesuatu oleh peneliti secara hati-hati, yang tujuannya untuk memecahkan masalah dan mengubah situasi *(Herbert, 1990)*
2. Eksperimen
   - Investigasi hubungan sebab akibat dengan menggunakan ujicoba yang dikontrol oleh peneliti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

(Dawson, 2009)



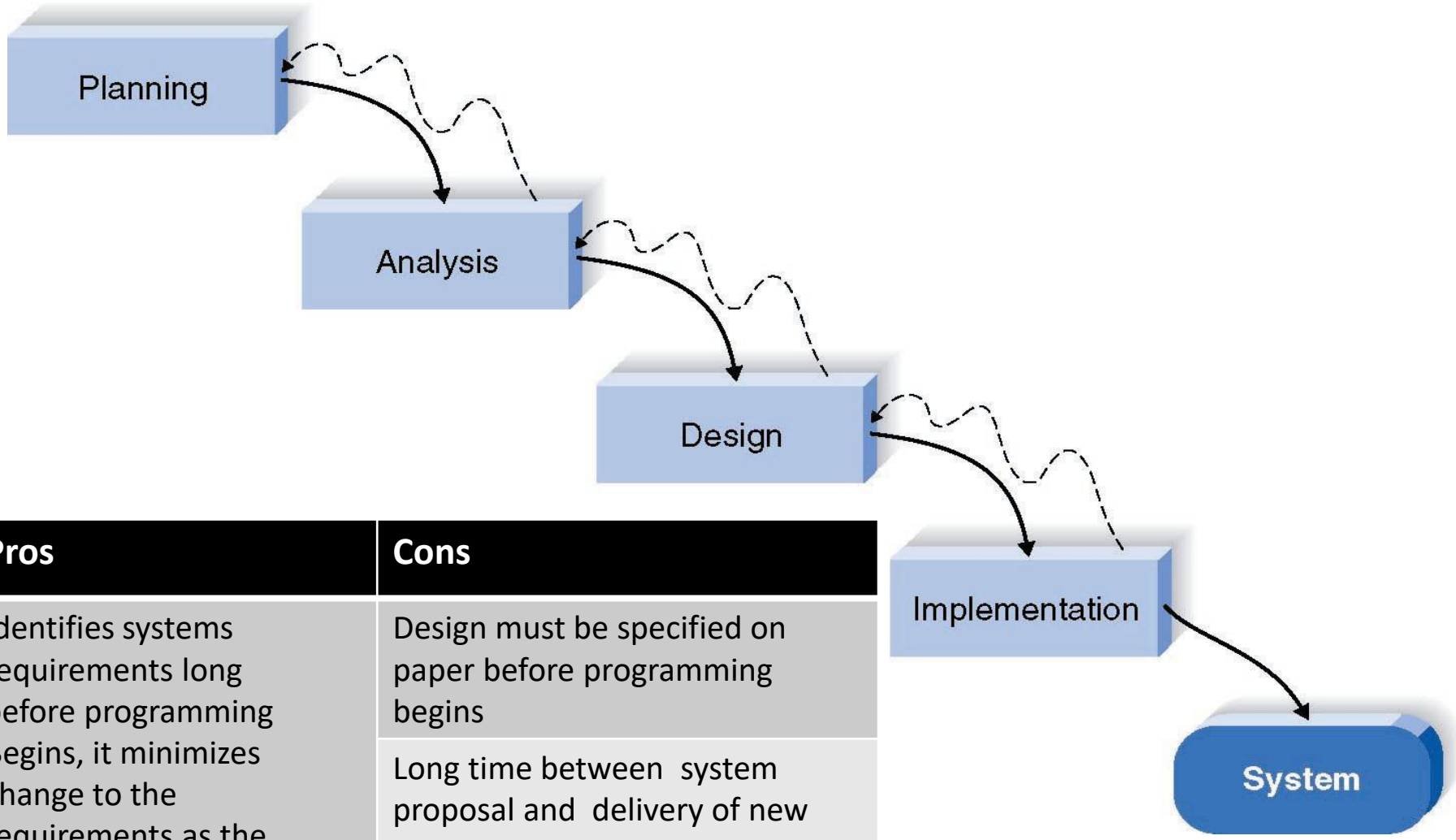
Metodologi Pengembangan Software

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



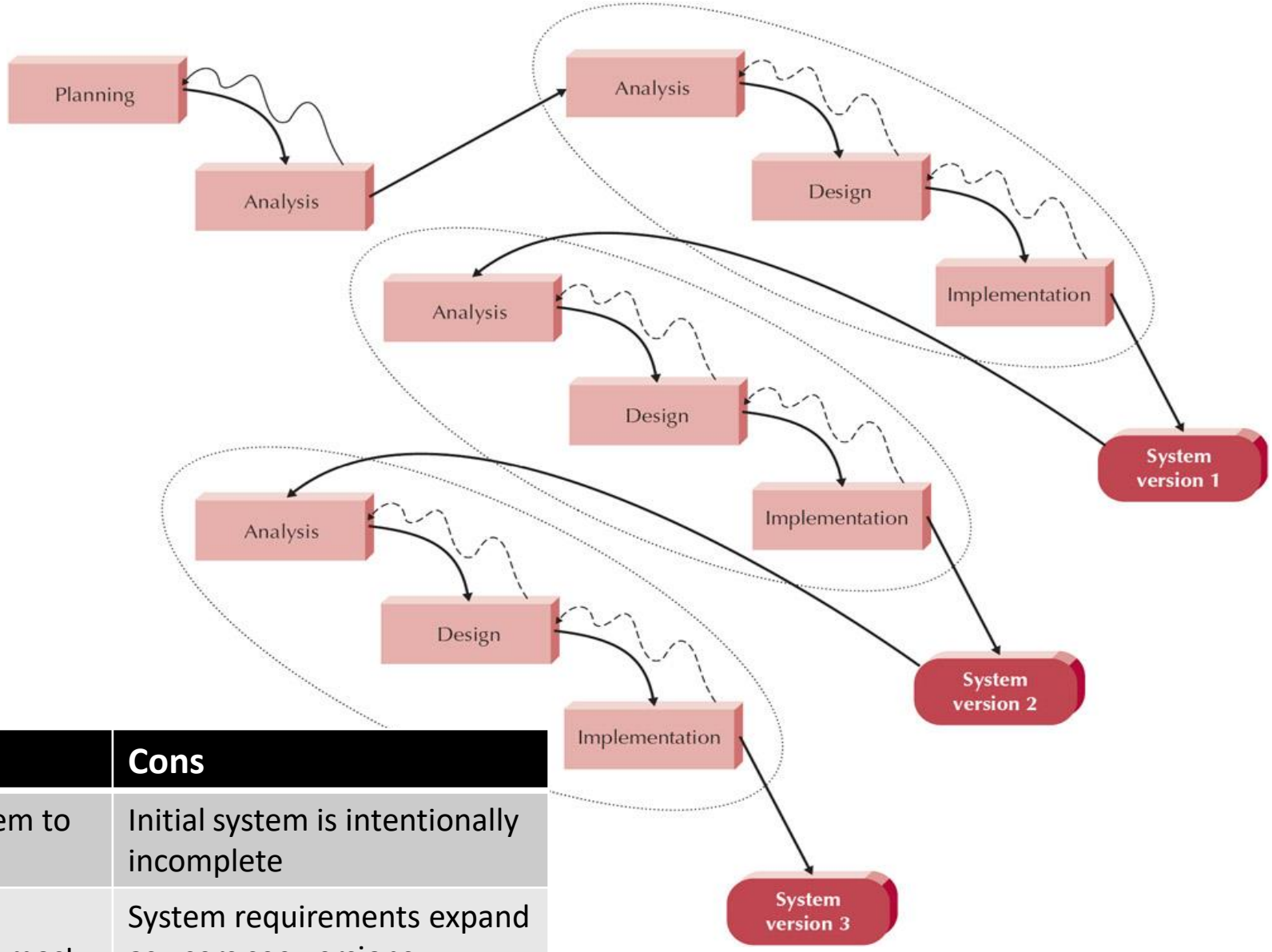
(Dennis, 2012)

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

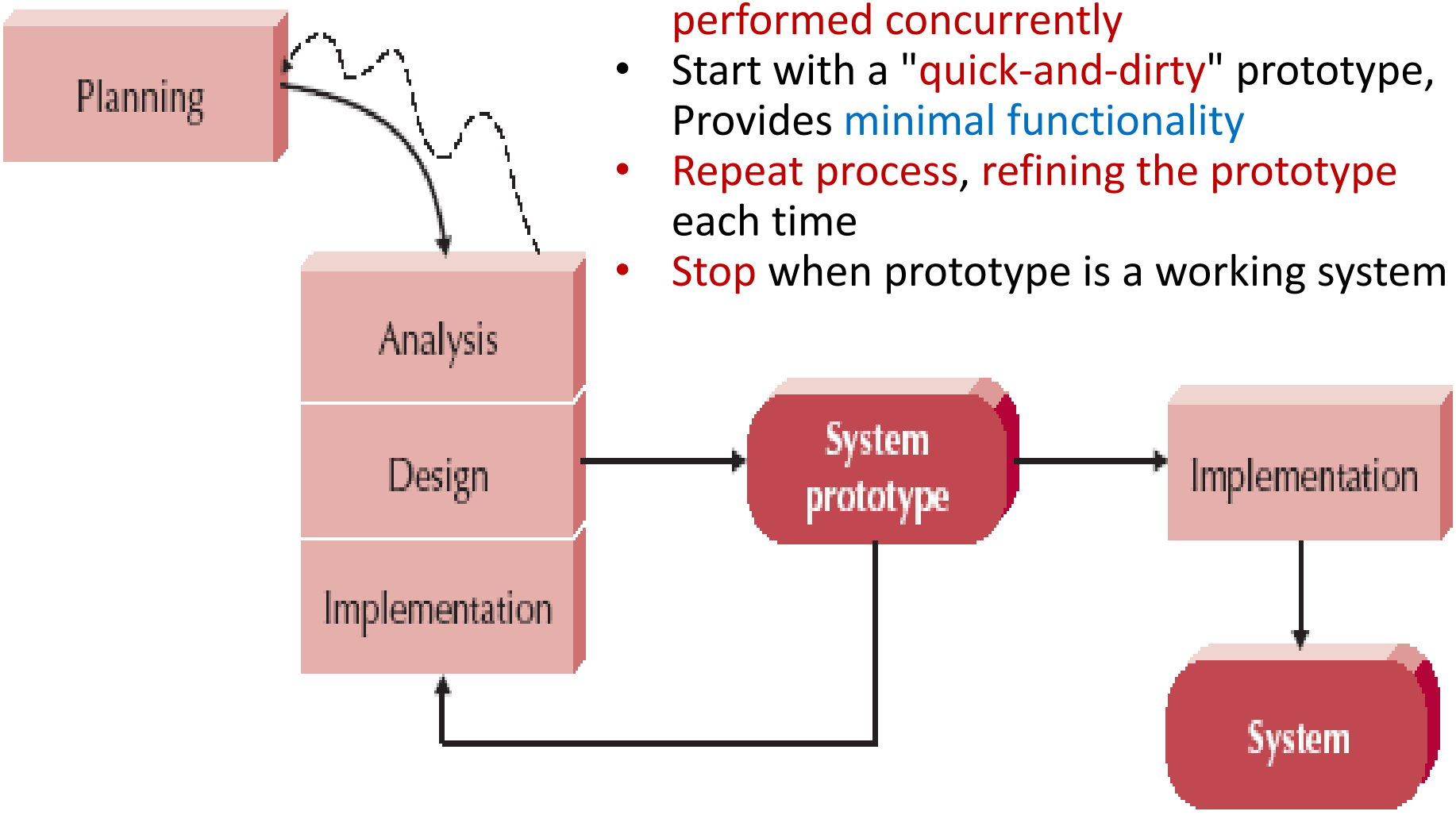
Phased Development



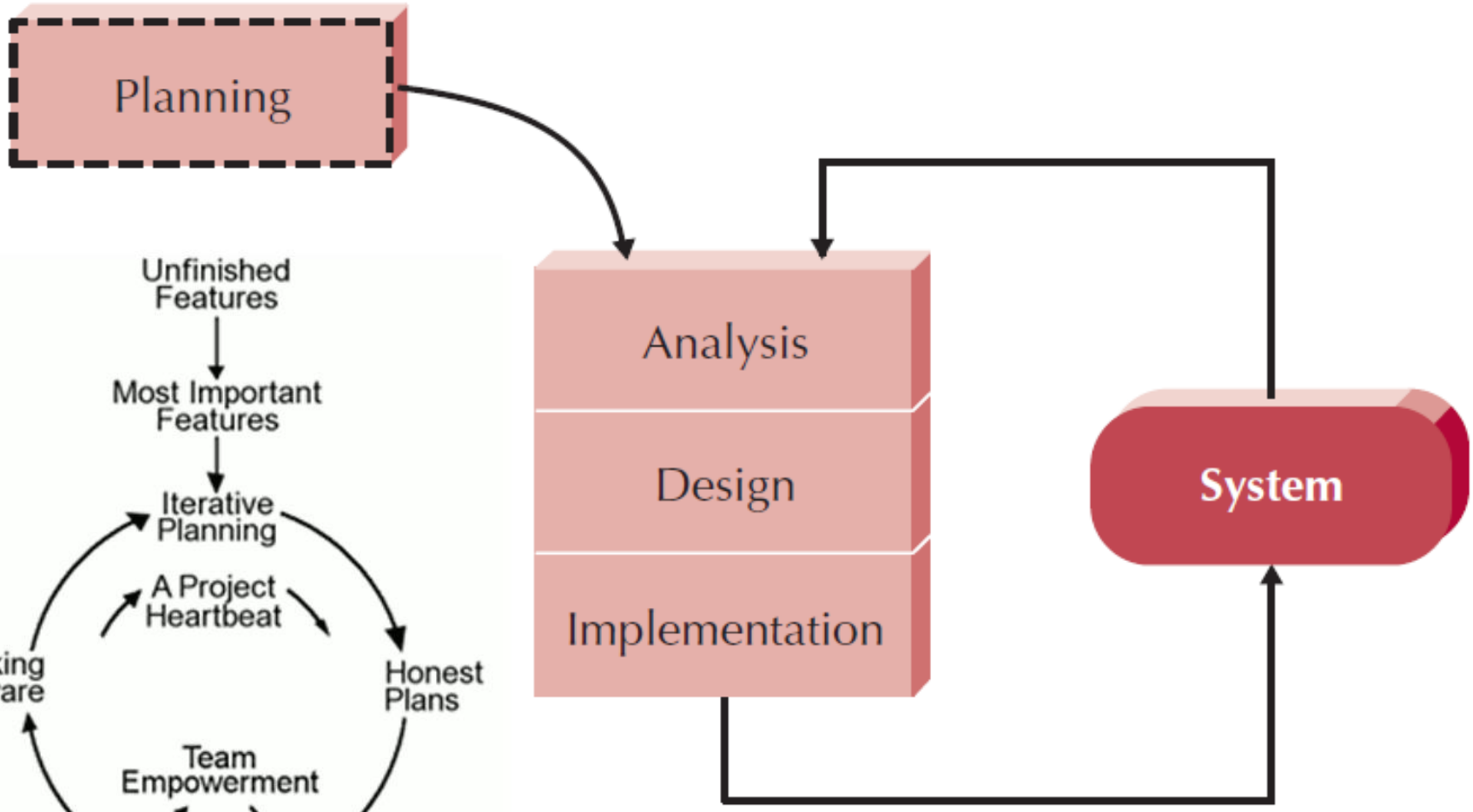
Pros	Cons
Gets useful system to users quickly	Initial system is intentionally incomplete
Most important functions tested most	System requirements expand as users see versions

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

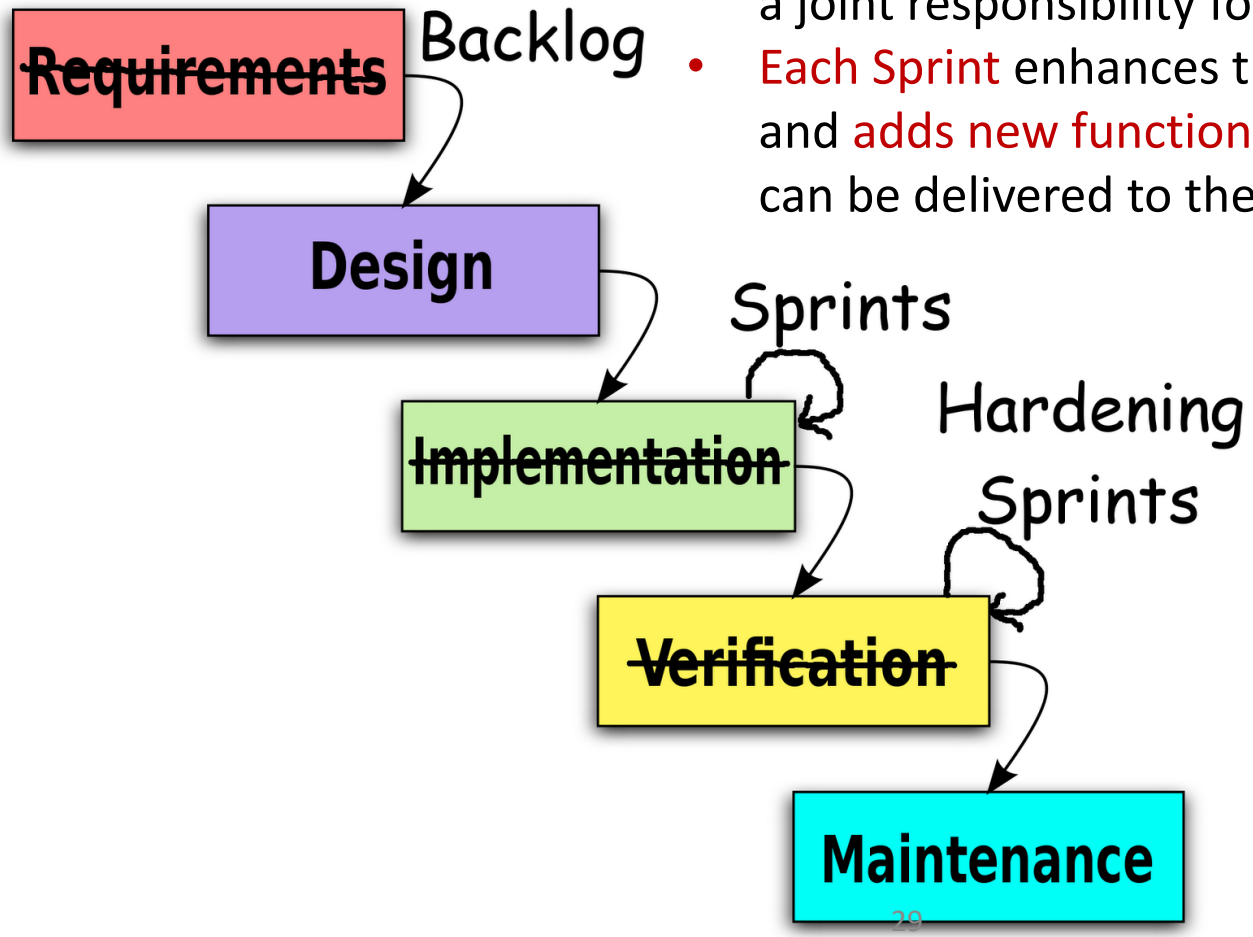


Extreme Programming



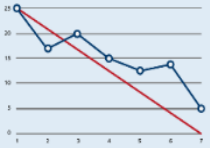
Scrum

- Project members form a Scrum Team consisting of 5–9 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**

- Taskboard (to do, in progress, done)
- Impediment Backlog
- Burndown Chart



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 Teams: 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

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)

Studi Literatur dan Masalah Penelitian

- 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*)

Contoh Masalah Penelitian

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

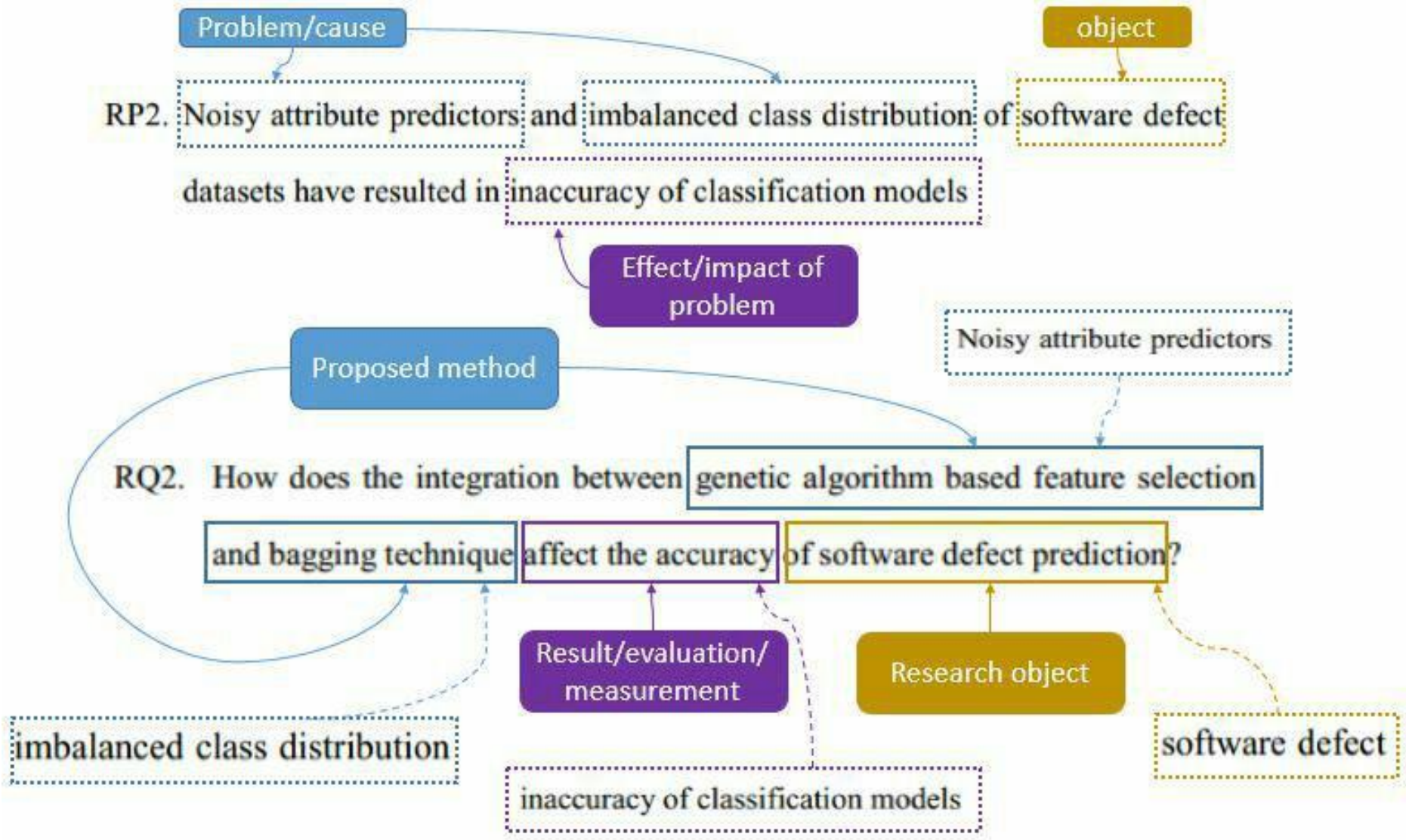
Contoh Masalah Penelitian

- Masalah Penelitian (*Research Problem*):
  - Neural network terbukti memiliki performa bagus untuk menangani data besar seperti pada data prediksi harga saham, akan tetapi memiliki kelemahan pada pemilihan arsitektur jaringannya yang harus dilakukan secara trial error, sehingga tidak efisien dan mengakibatkan hasil prediksi kurang akurat
- Rumusan Masalah (*Research Question*):
  - Bagaimana peningkatan akurasi dan efisiensi neural network apabila pada pemilihan arsitektur jaringan diotomatisasi menggunakan algoritma genetika?
- Tujuan Penelitian (*Research Objective*):
  - Menerapkan algoritma genetika untuk mengotomatisasi pemilihan arsitektur jaringan pada neural nework sehingga lebih efisien dan hasil prediksi lebih akurat

Contoh Masalah Penelitian

- Research Problem (RP):
  - Algoritma K-Means merupakan algoritma clustering yang populer karena efisien dalam komputasi, akan tetapi memiliki kelemahan pada sulitnya penentuan K yang optimal dan komputasi yang tidak efisien bila menangani data besar (Zhao, 2010)
- Research Question (RQ):
  - Seberapa efisien algoritma Bee Colony bila digunakan untuk menentukan nilai K yang optimal pada K-Means?
- Research Objective (RO):
  - Menerapkan algoritma bee colony untuk menentukan nikai K yang optimal pada K-Means sehingga komputasi lebih efisien

Contoh Masalah Penelitian

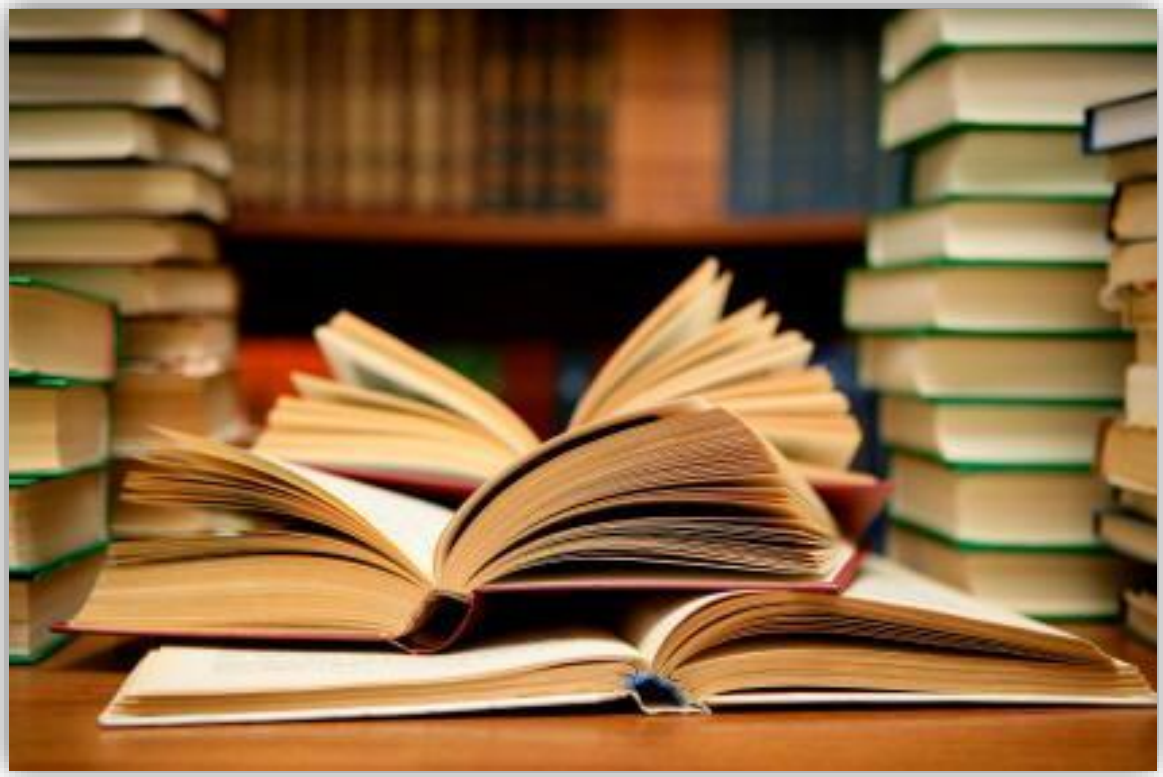


Masalah Penelitian dan Landasannya

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)*

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*)

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)

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)

Example of SLR: PICOC

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

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

Example of SLR: Research Question (RQ)

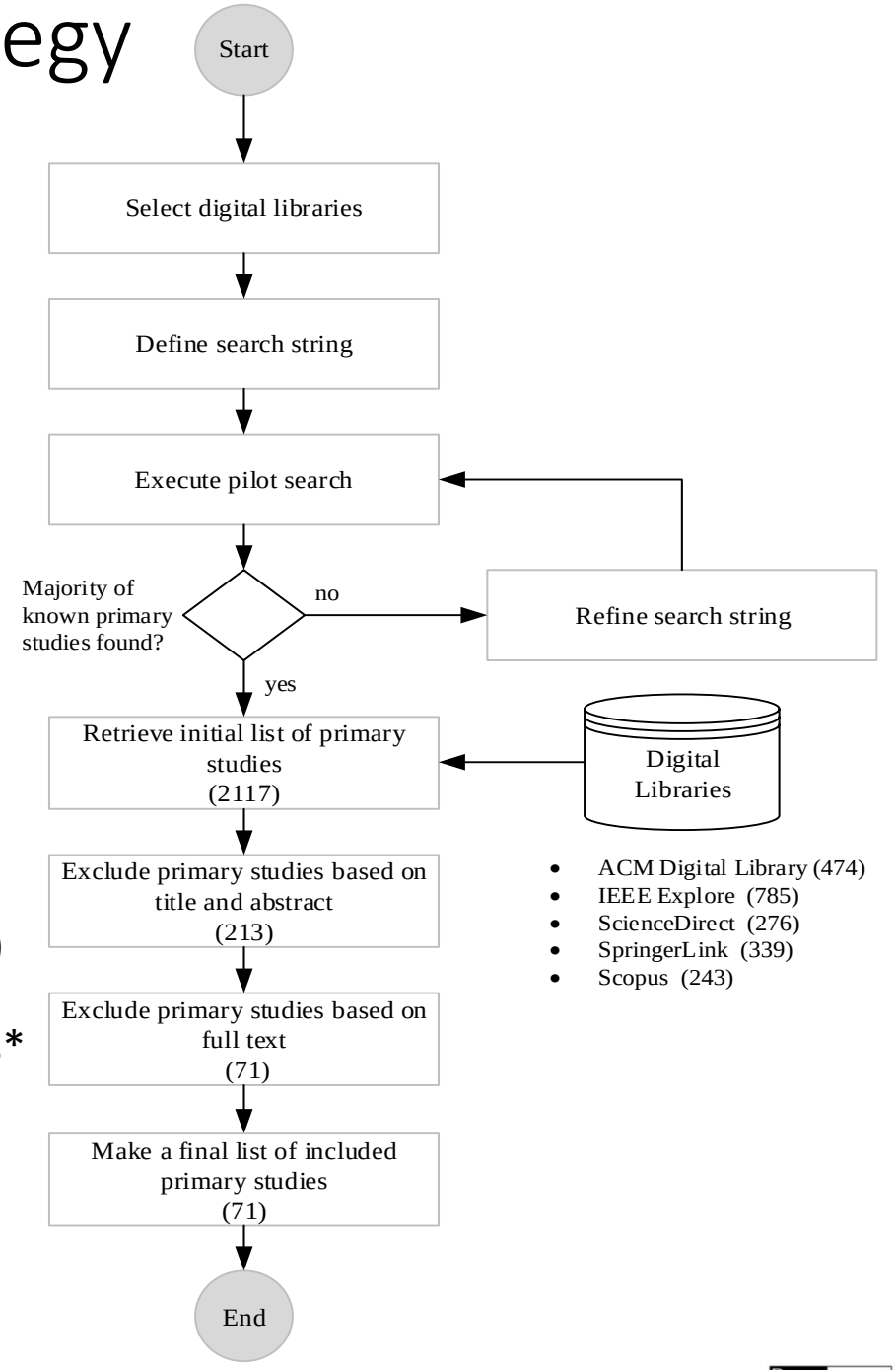
Romi Satria Wahono, A Systematic Literature Review of Software Defect Prediction: Research Trends, Datasets, Methods and Frameworks, *Journal of Software Engineering, Vol. 1, No. 1*, pp. 1-16, April 2015

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?

Studies Selection Strategy (Wahono, 2015)

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

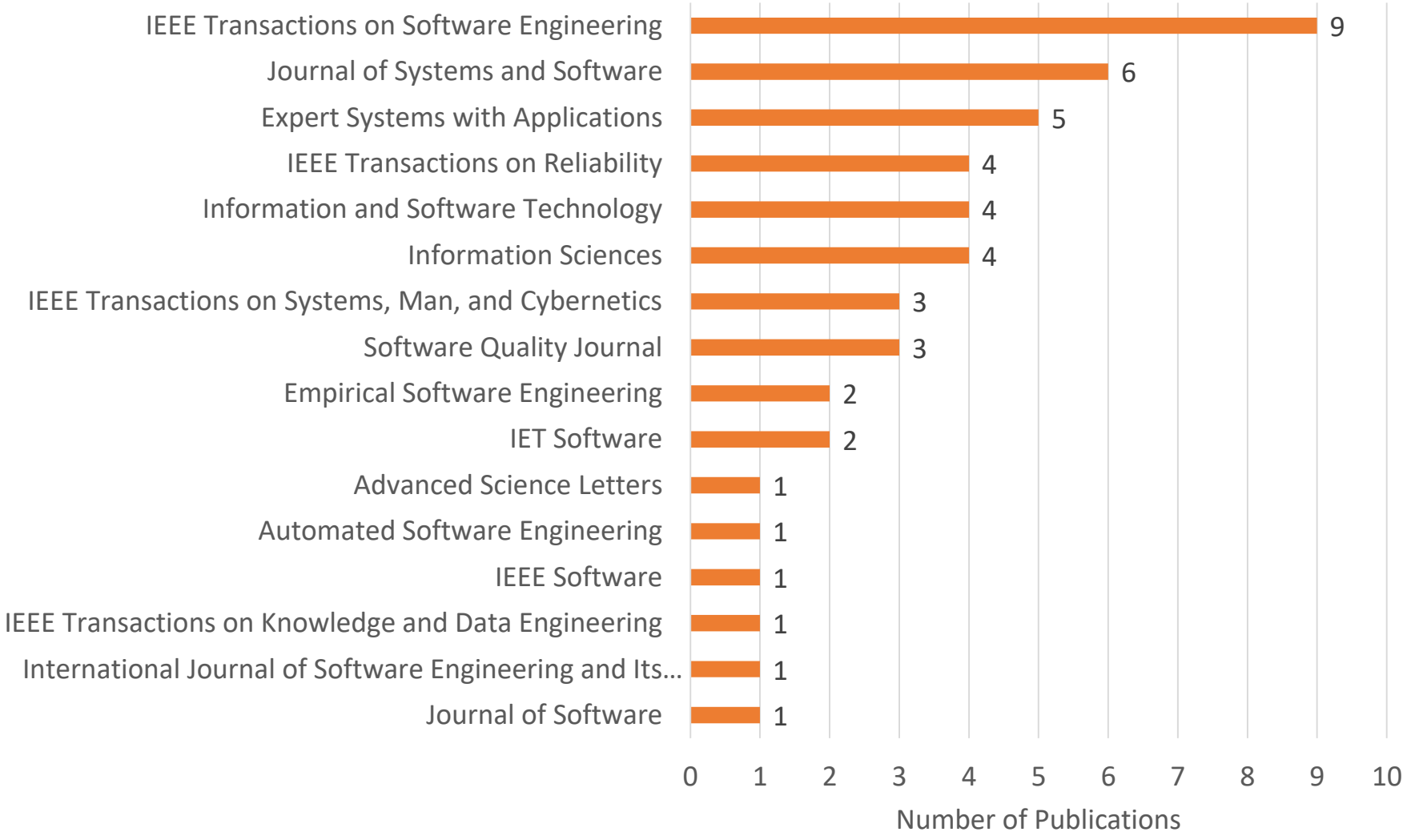
  software
  AND
  (fault* OR defect* OR quality OR error-prone)
  AND
  (predict* OR prone* OR probability OR assess* OR detect* OR estimat* OR classificat*)
- Selected Studies:
  - ✓71



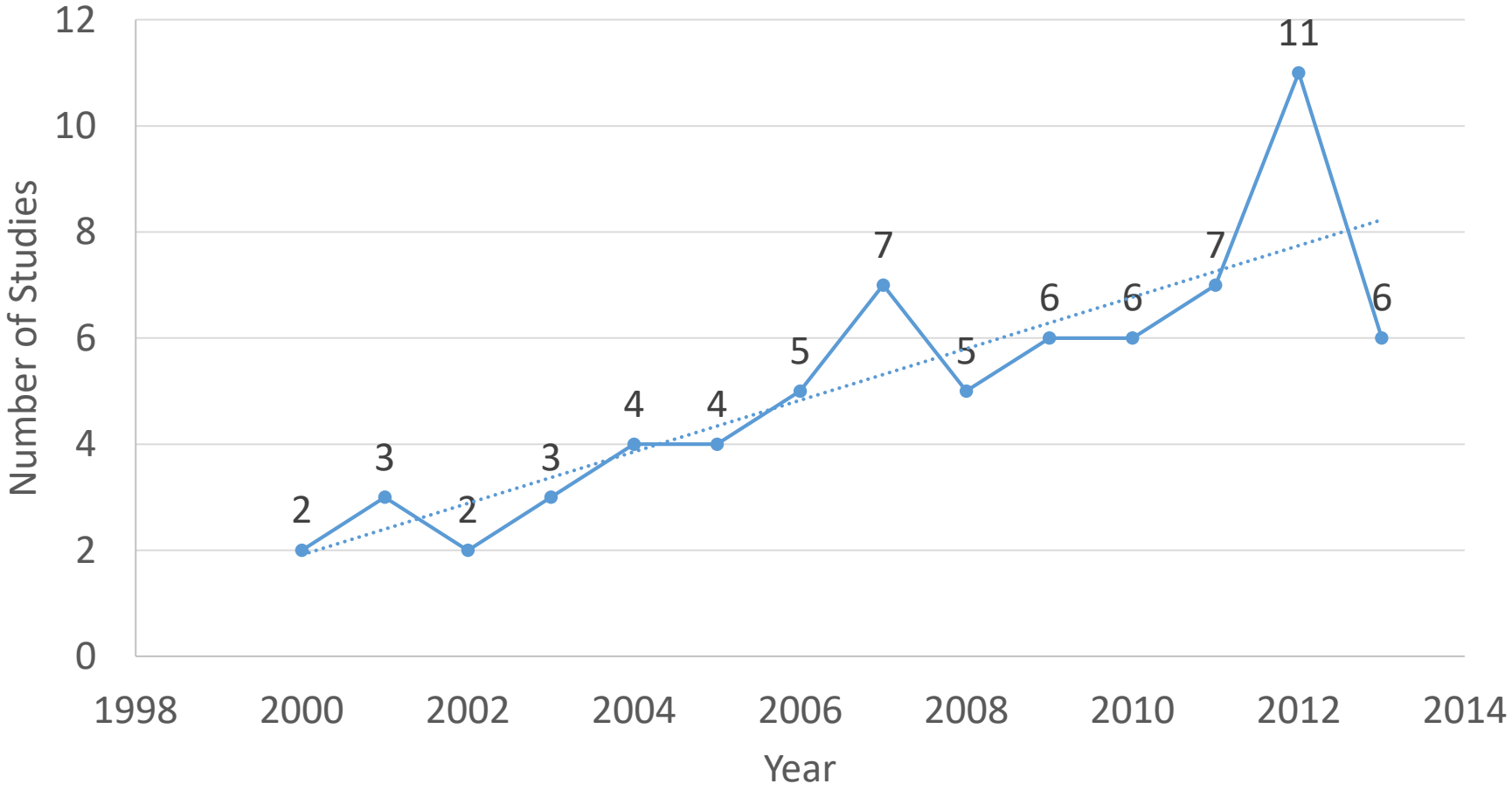
Selection of Studies *(Wahono, 2015)*

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

RQ1: Significant Journal Publications *(Wahono, 2015)*



Distribution of Selected Studies by Year
(Wahono, 2015)

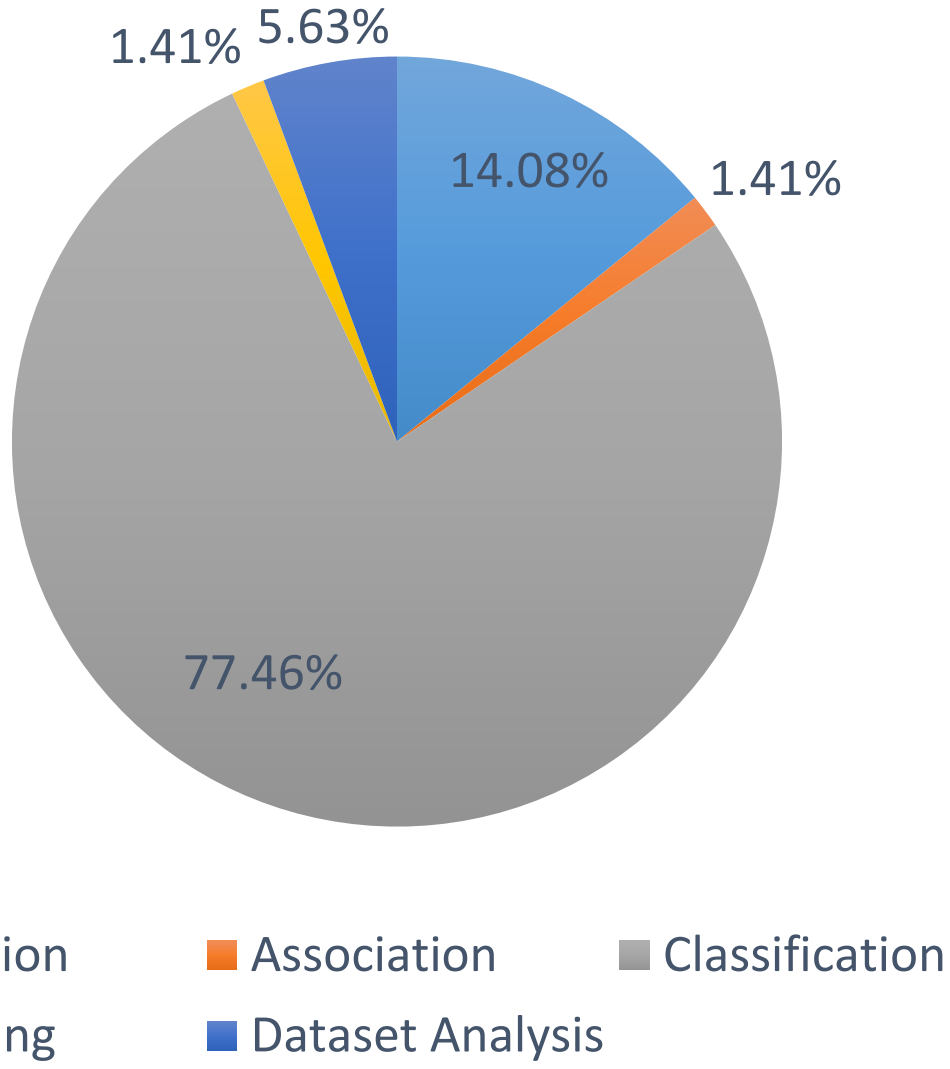


- The interest in software defect prediction has changed over time
- Software defect prediction research is still very much relevant to this day

RQ3: Research Topics and Trends
(Wahono, 2015)

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 *(Wahono, 2015)*



RQ9: Existing Frameworks

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

Menzies Framework

(Menzies et al. 2007)

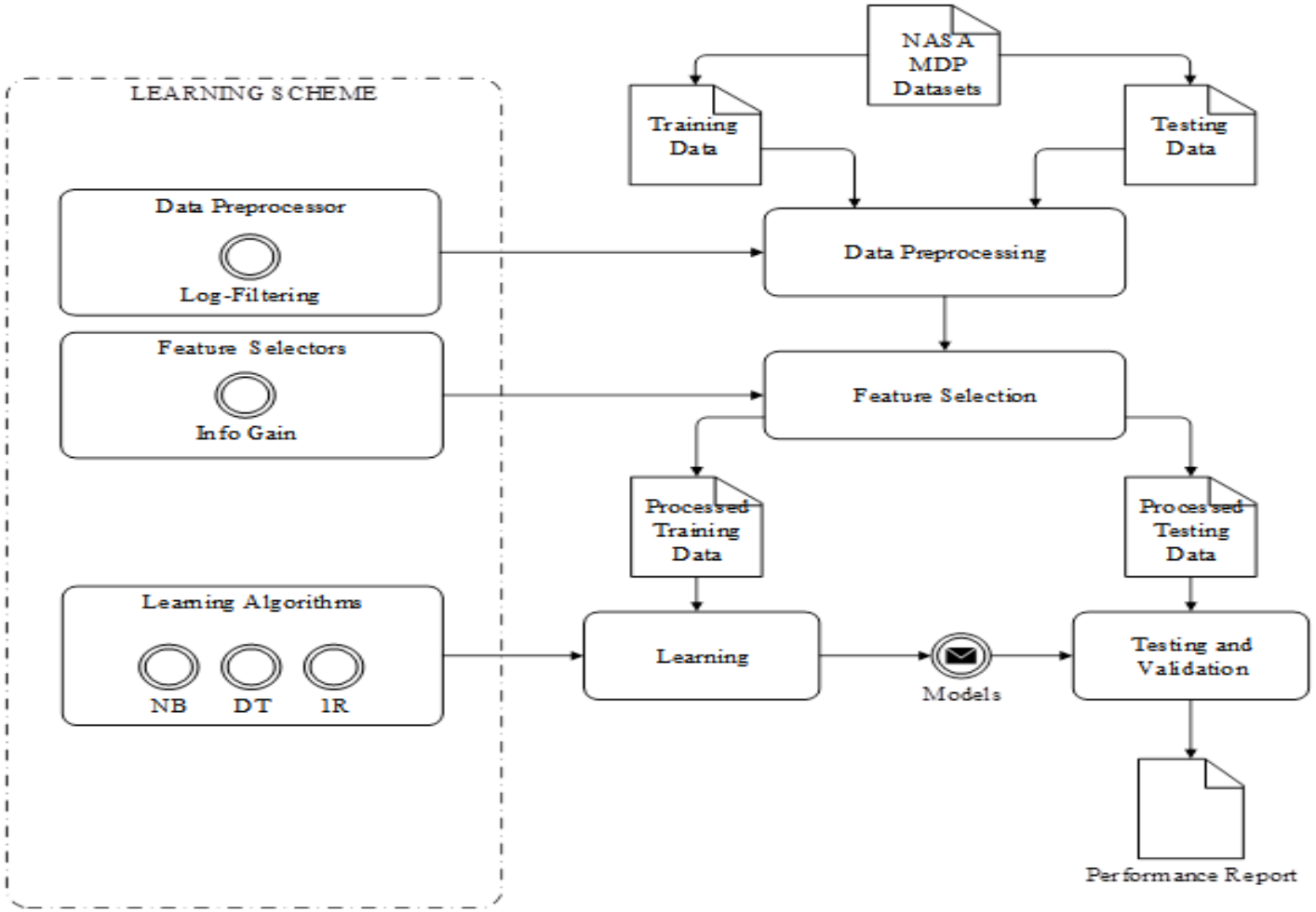
Lessmann Framework

(Lessmann et al. 2008)

Song Framework

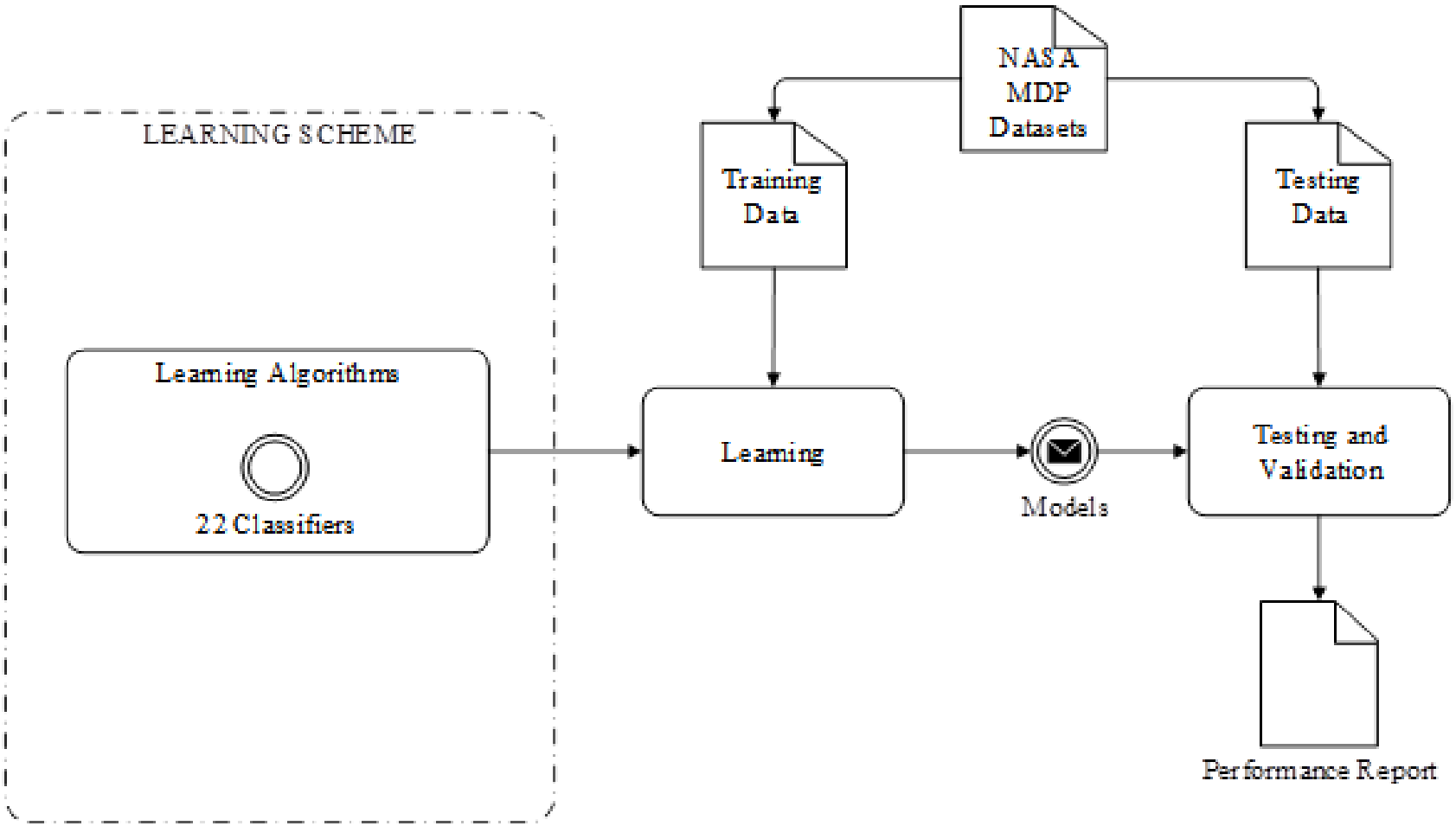
(Song et al. 2011)

Menzies Framework (Menzies et al. 2007)



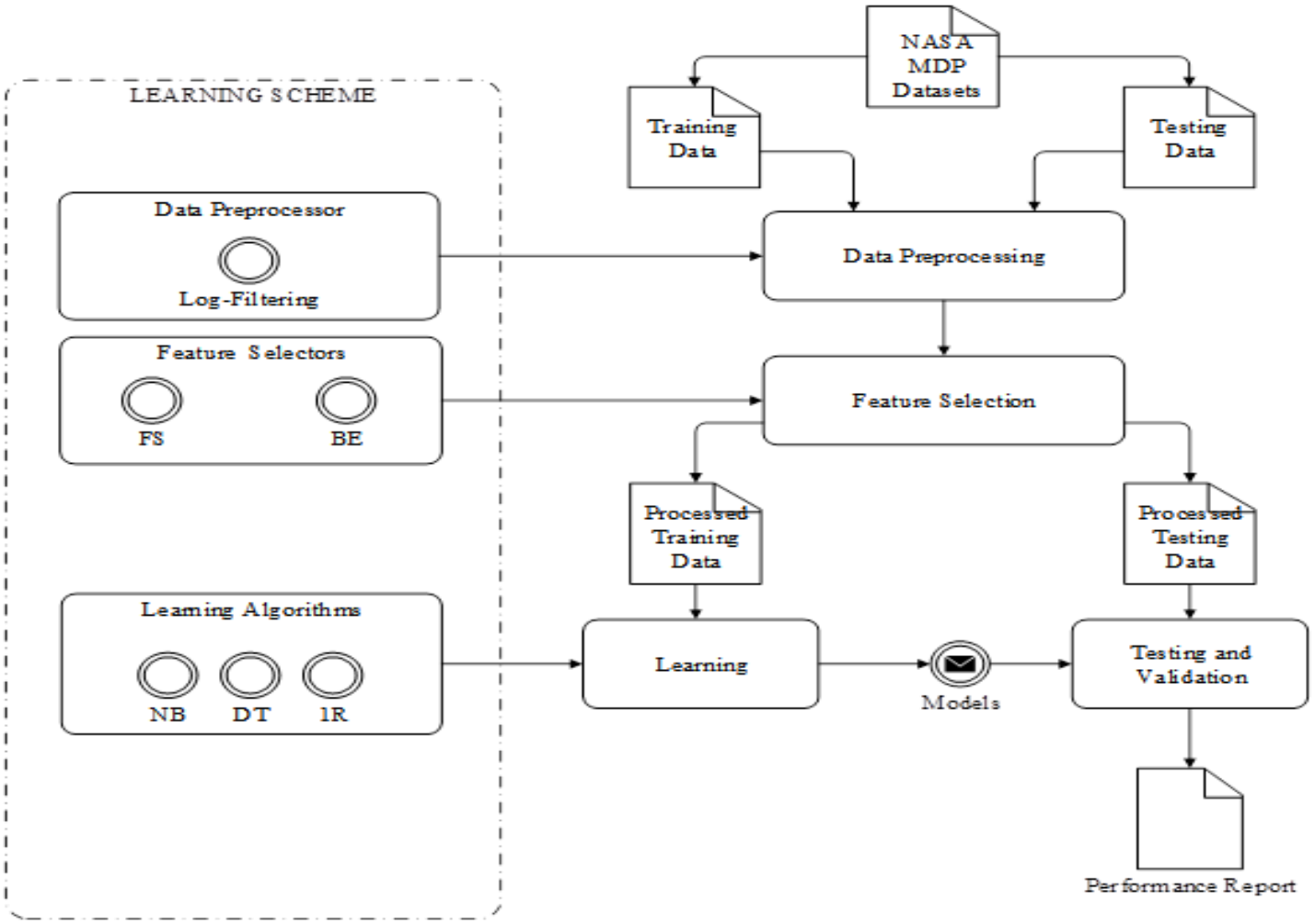
Framework	Dataset	Data Preprocessor	Feature Selectors	Meta-learning	Classifiers	Parameter Selectors	Validation Methods	Evaluation Methods
(Menzies et al. 2007)	NASA MDP	**Log Filtering**	**Info Gain**	-	**3 algorithms (DT, 1R, NB)**	-	10-Fold X Validation	ROC Curve (AUC)

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)



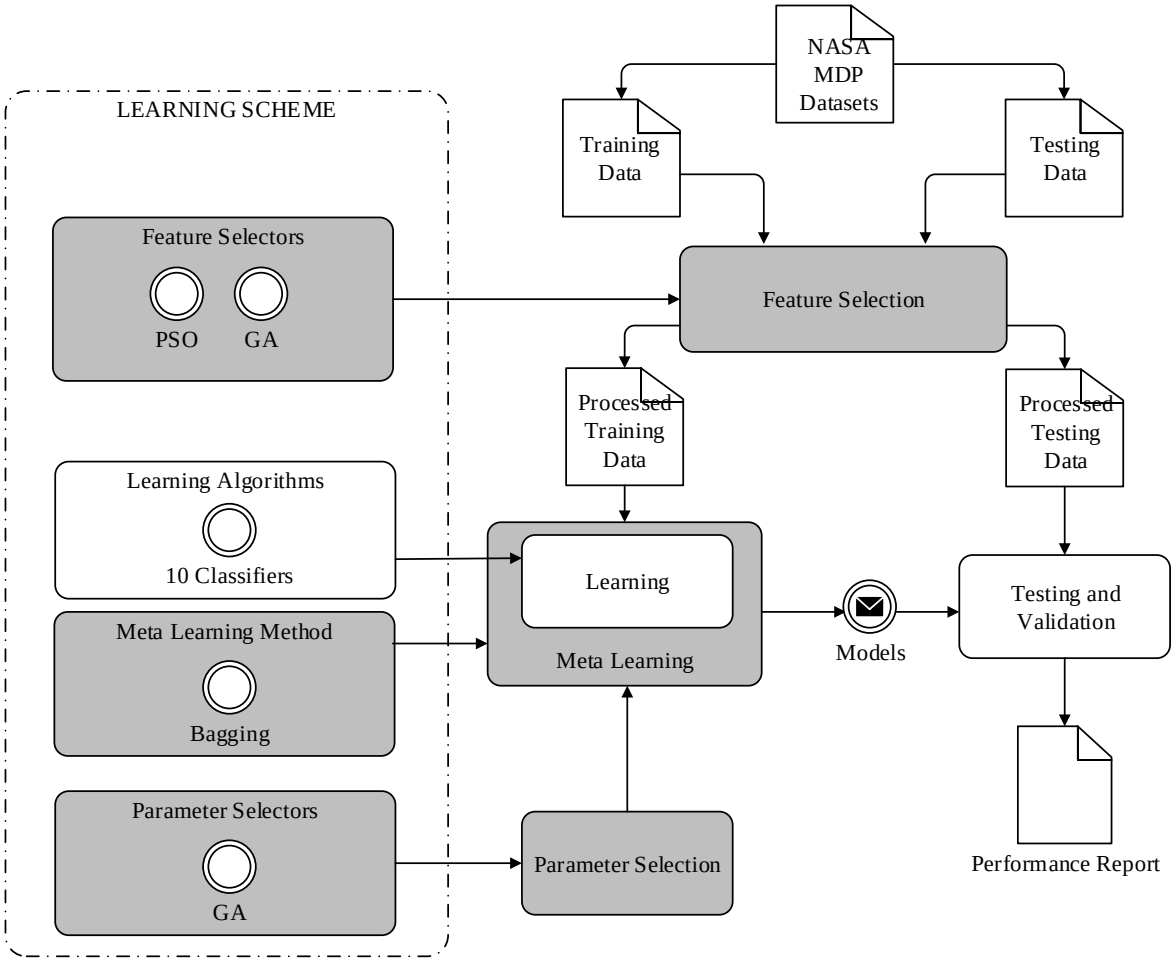
Framework	Dataset	Data Preprocessor	Feature Selectors	Meta-learning	Classifiers	Parameter Selectors	Validation Methods	Evaluation Methods
(Song et al. 2011)	NASA MDP	**Log Filtering**	**FS, BE**	-	**3 algorithms (DT, 1R, NB)**	-	10-Fold X Validation	ROC Curve (AUC)

Gap Analysis of Framework

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

- 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

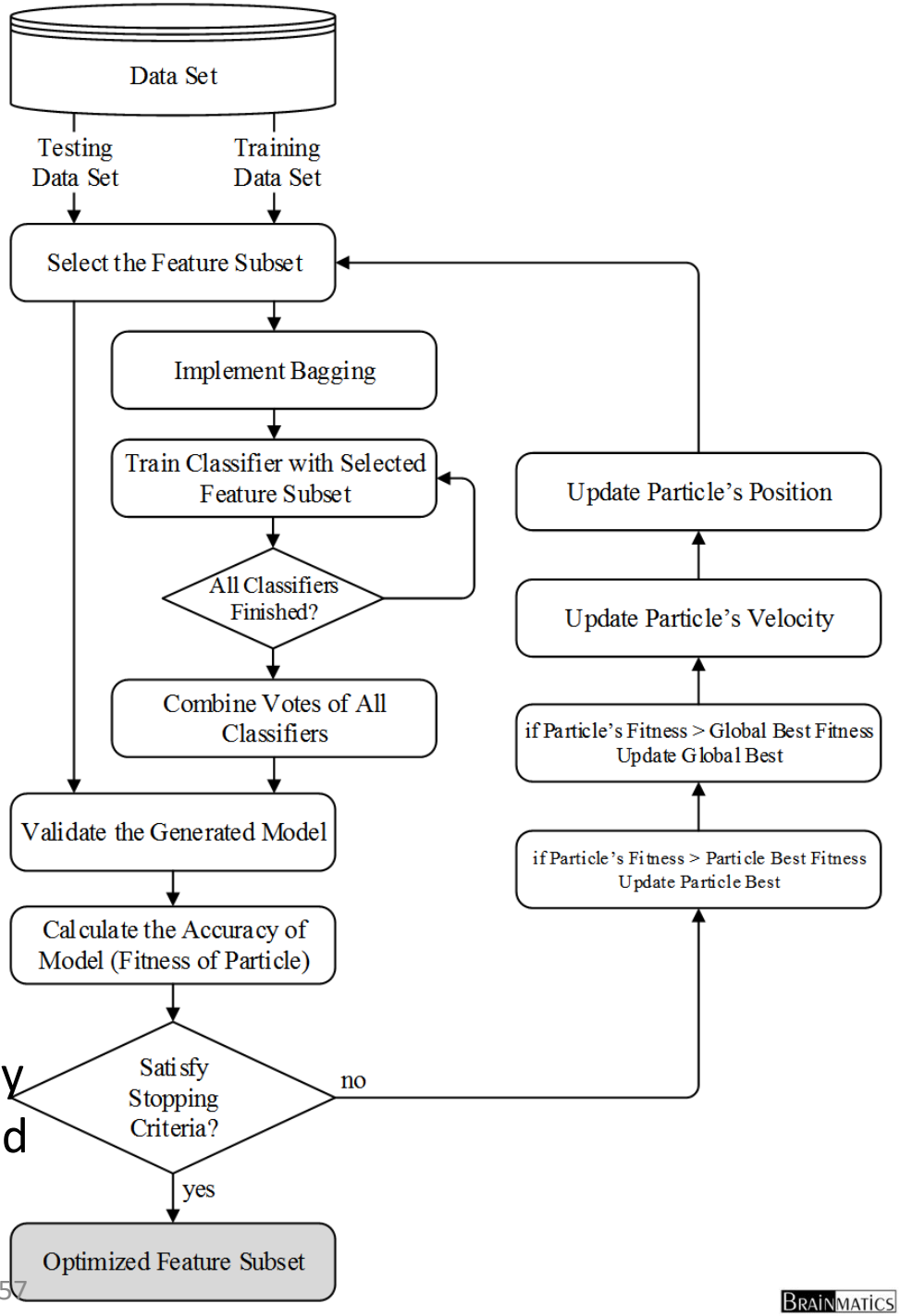
Proposed Framework



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)

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

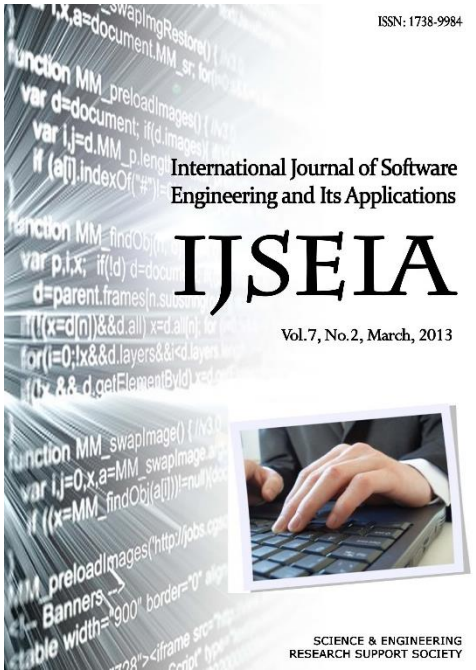
Without PSOFS+B vs With PSOFS+B

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 (SCOPUS SJR: 0.390)

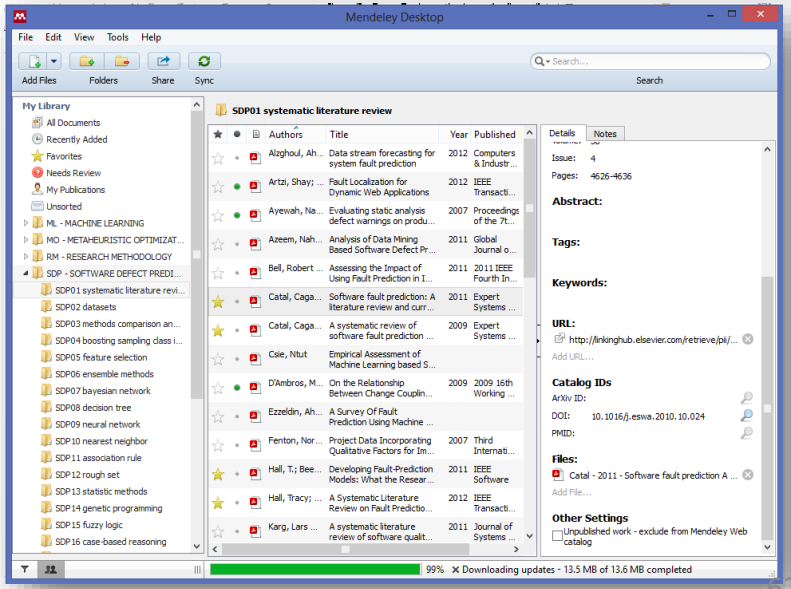


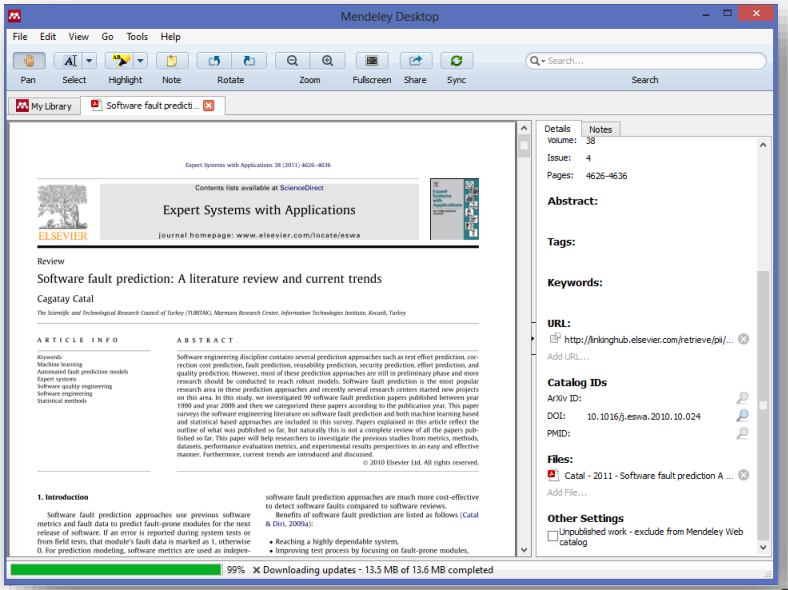
MITOS 6

Semakin Banyak Literatur yang Saya Baca, Saya Semakin Pusing

Jumlah Literatur yang Dibaca

- Level pendidikan dan jumlah literatur
  - 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

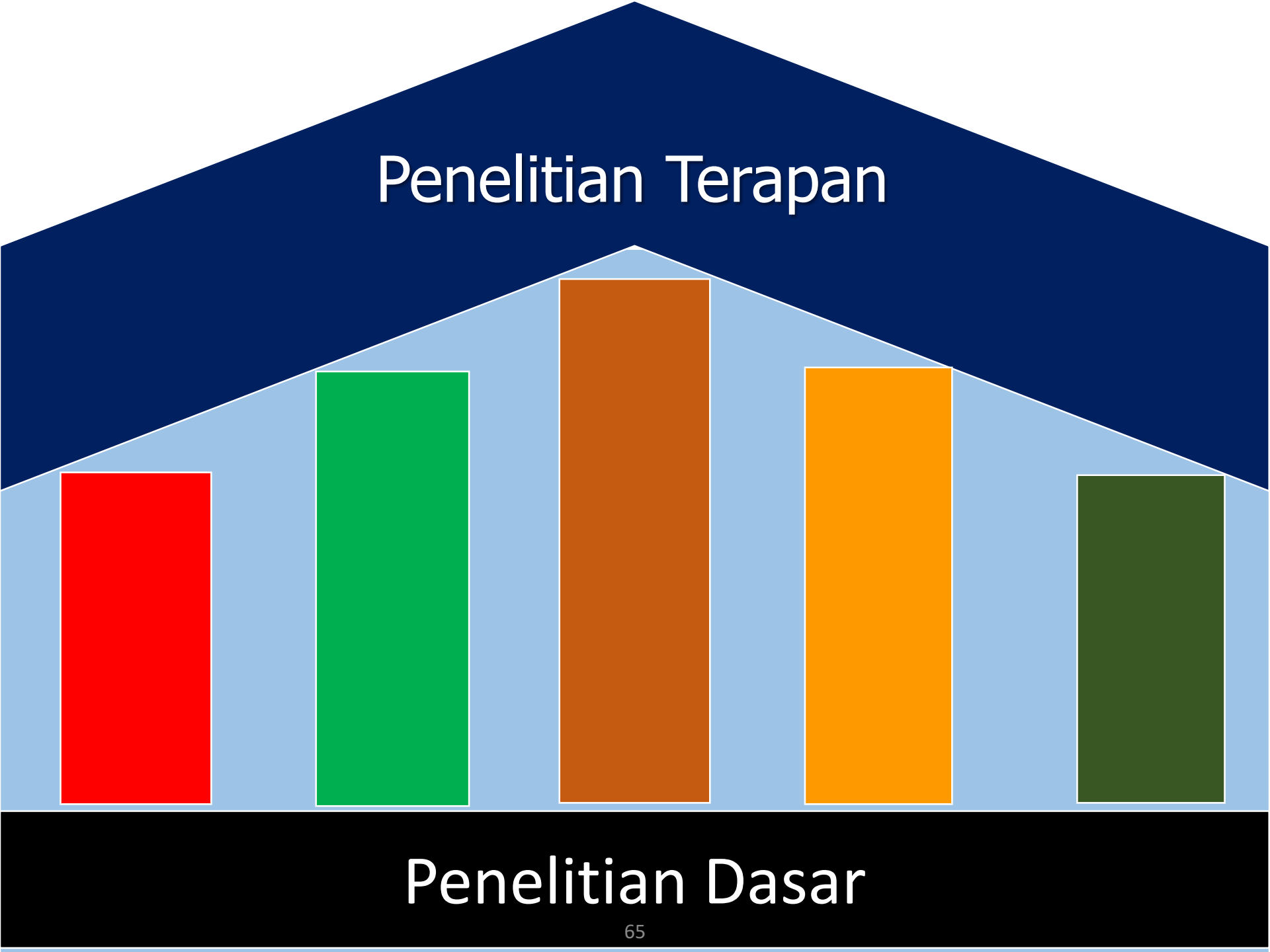
Literature Review

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/*

MITOS 7

Penelitian Itu Semakin Aplikatif dan Terapan Semakin Mudah Masuk Jurnal Terindeks


Penelitian Terapan
Penelitian Dasar

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

IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 41, NO. 1, JANUARY 2011 93

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),

Memperbaiki Genetic Algorithms

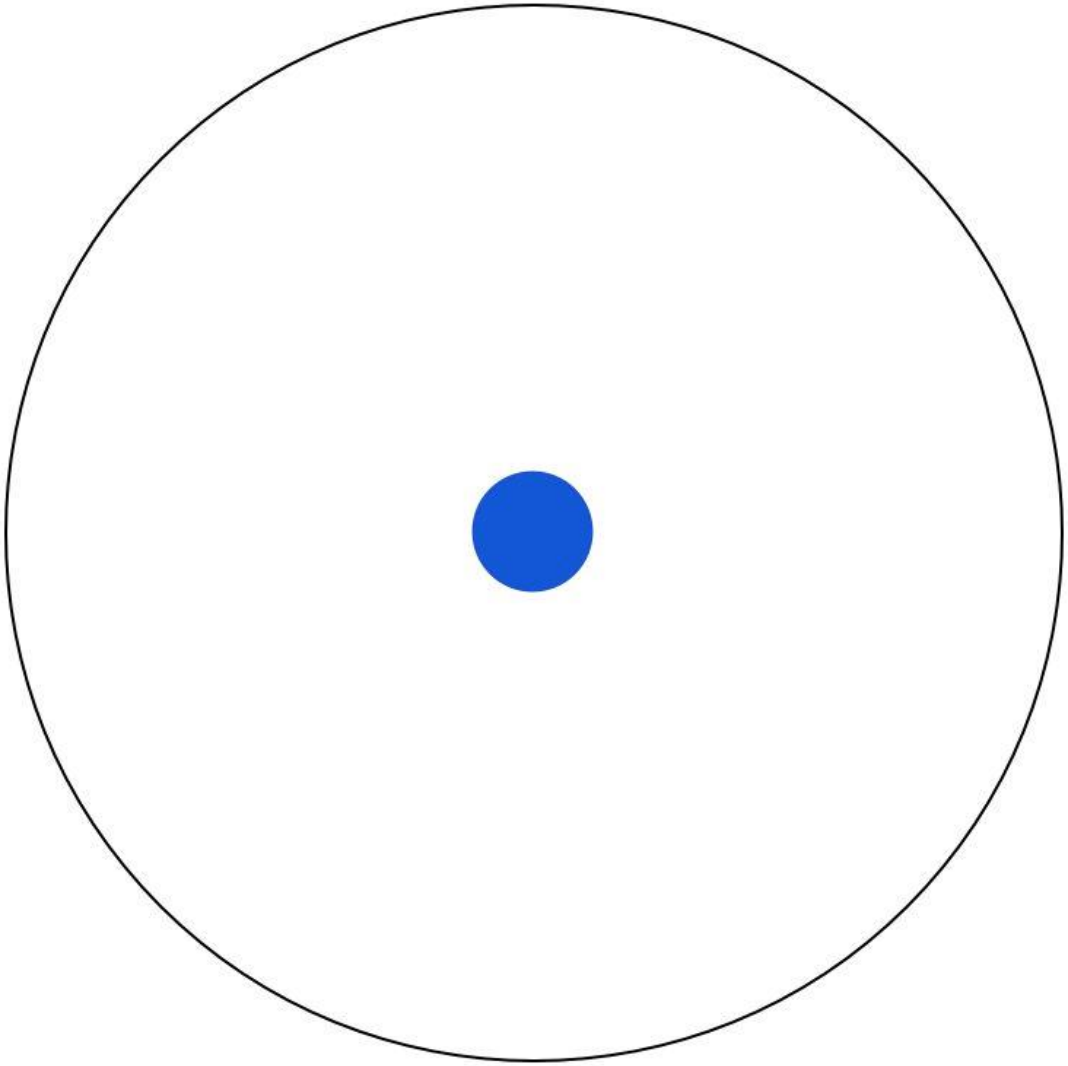
MITOS 8

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

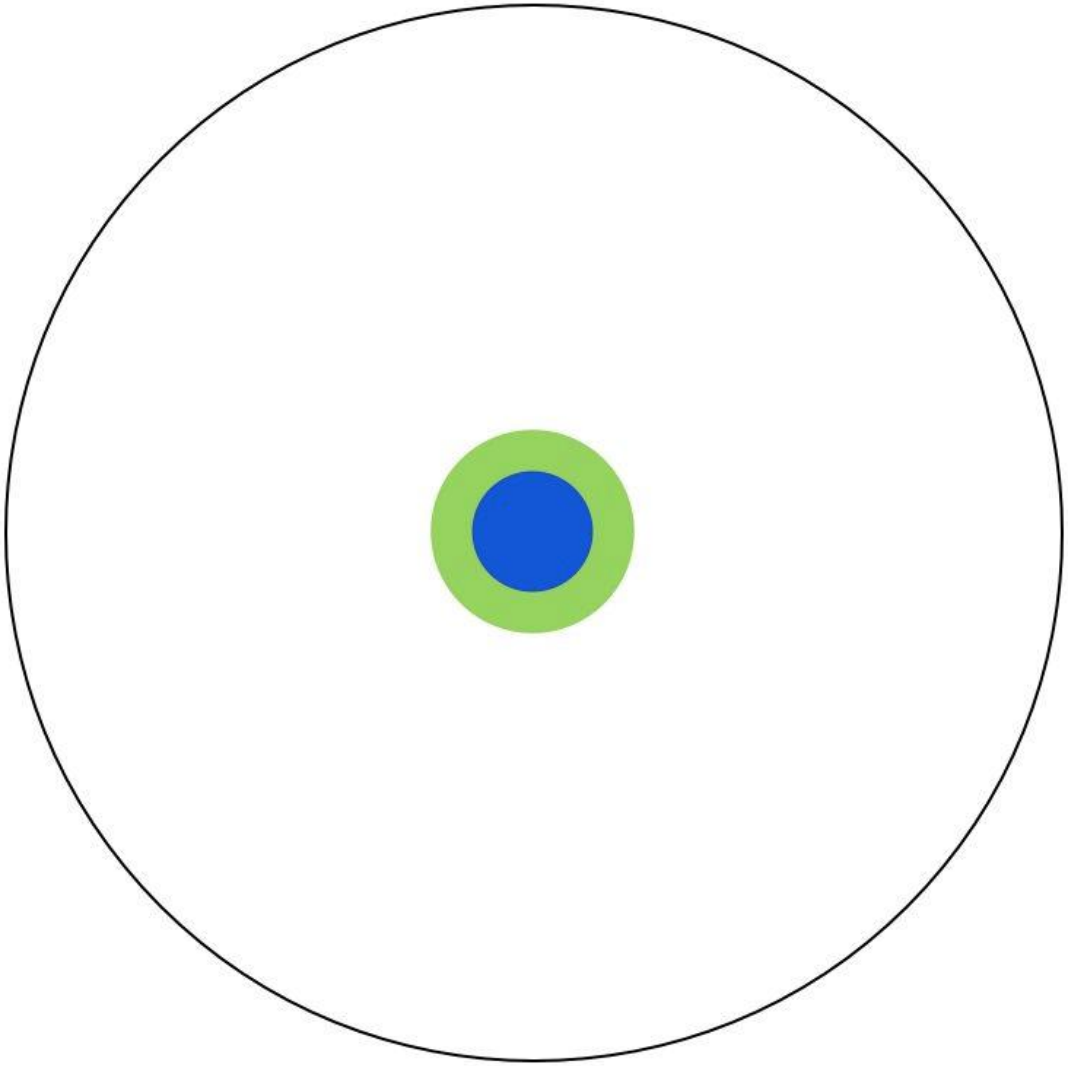
Penelitian yang Berkualitas Tinggi

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

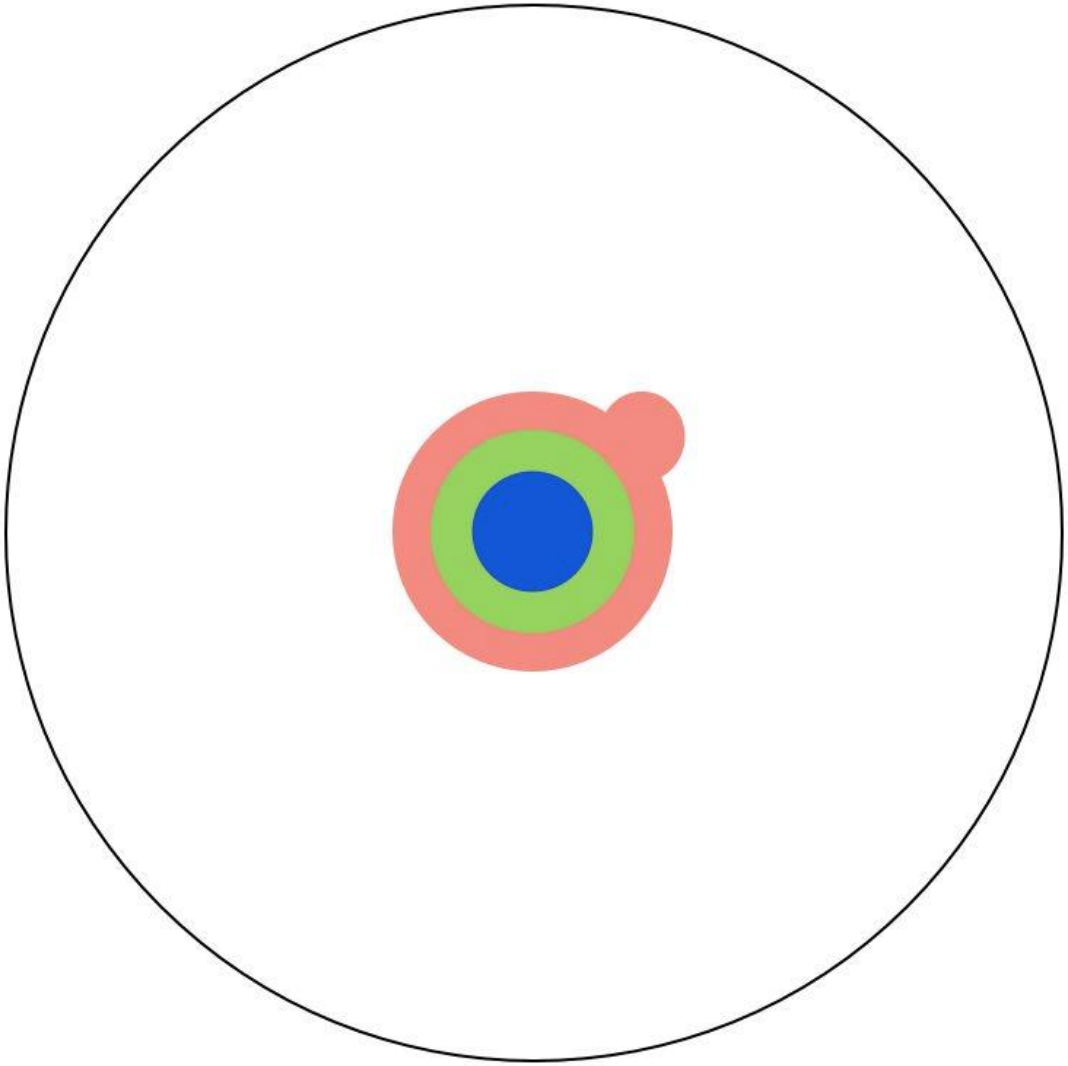
The Illustrated Guide to a Ph.D (Might, 2010)



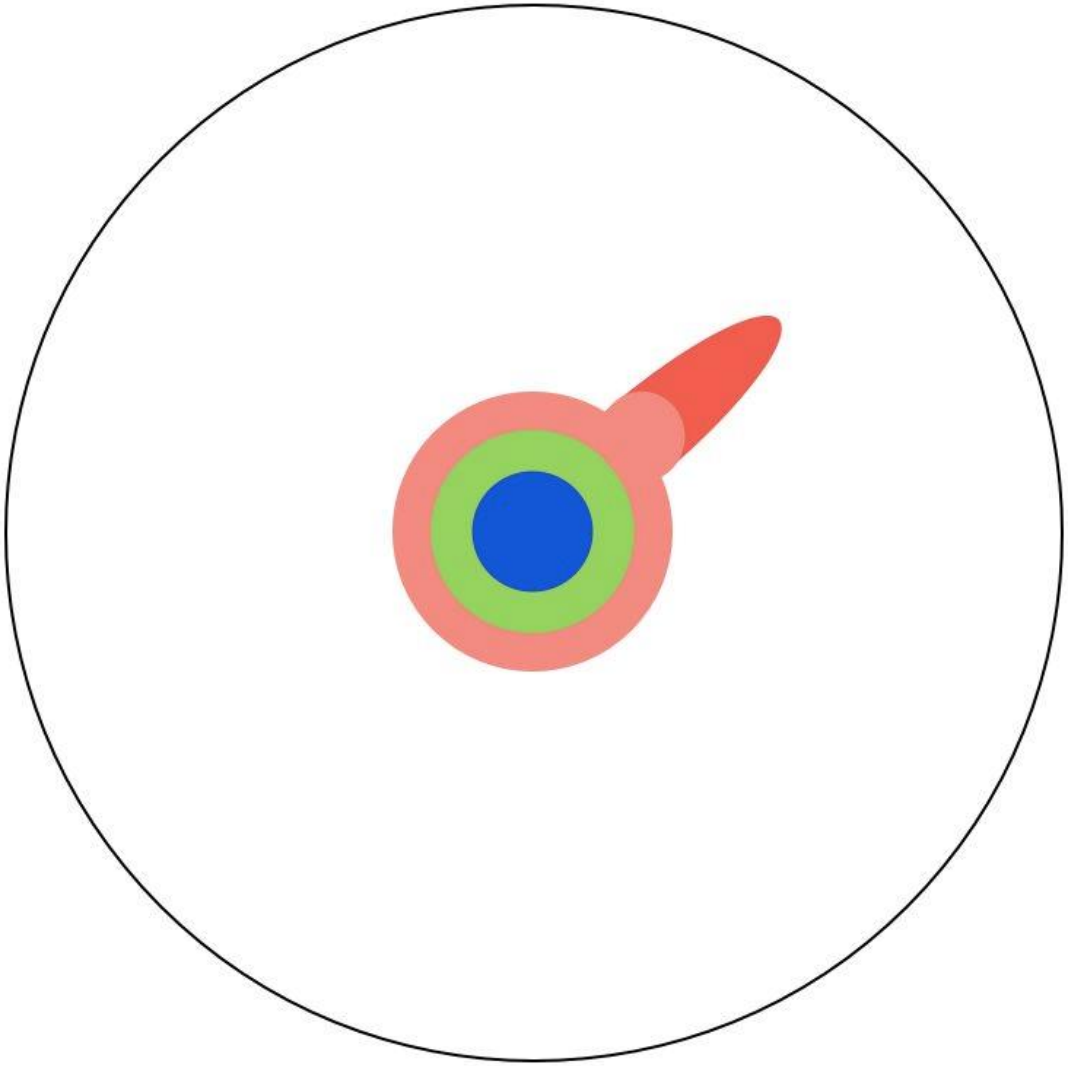
The Illustrated Guide to a Ph.D (Might, 2010)



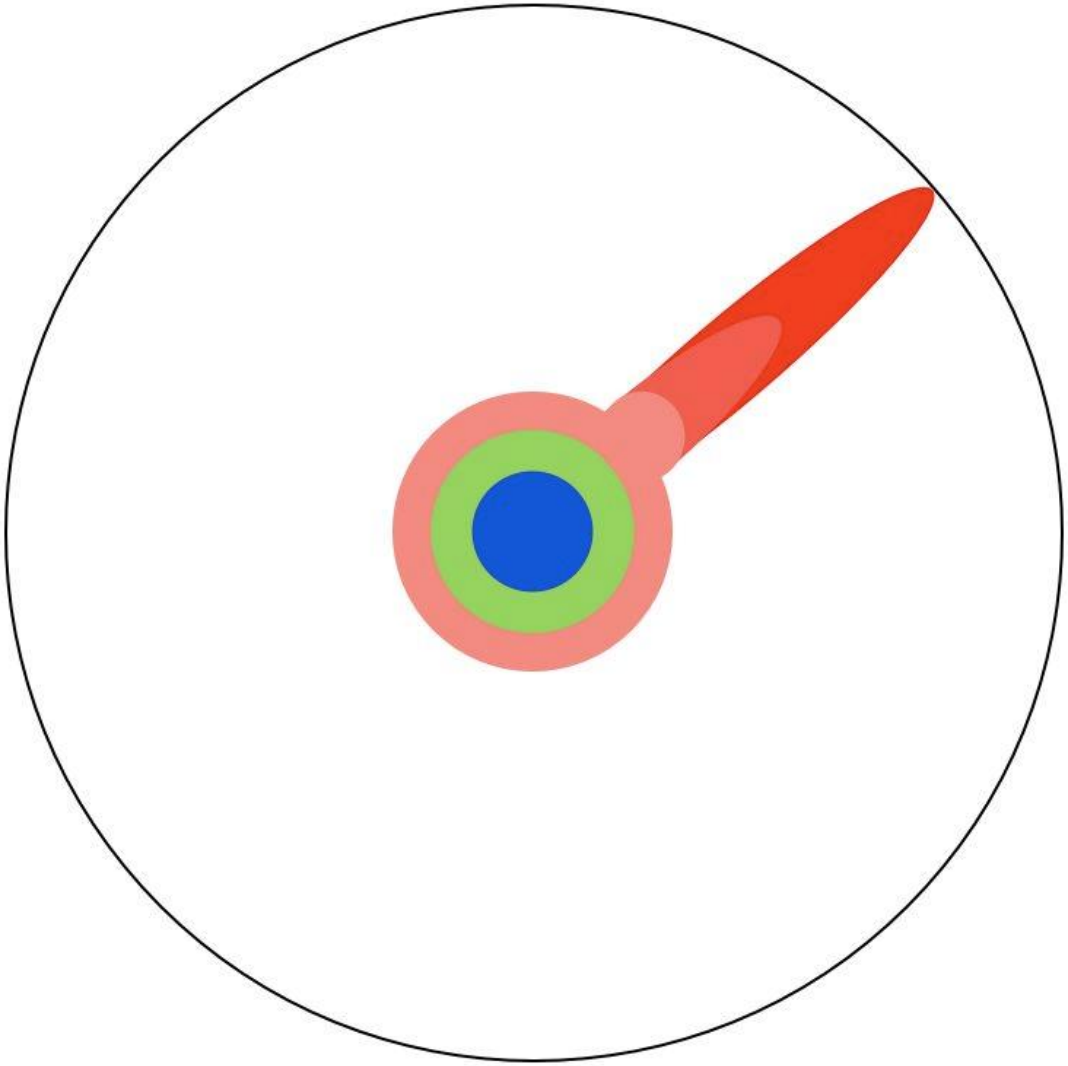
The Illustrated Guide to a Ph.D (Might, 2010)



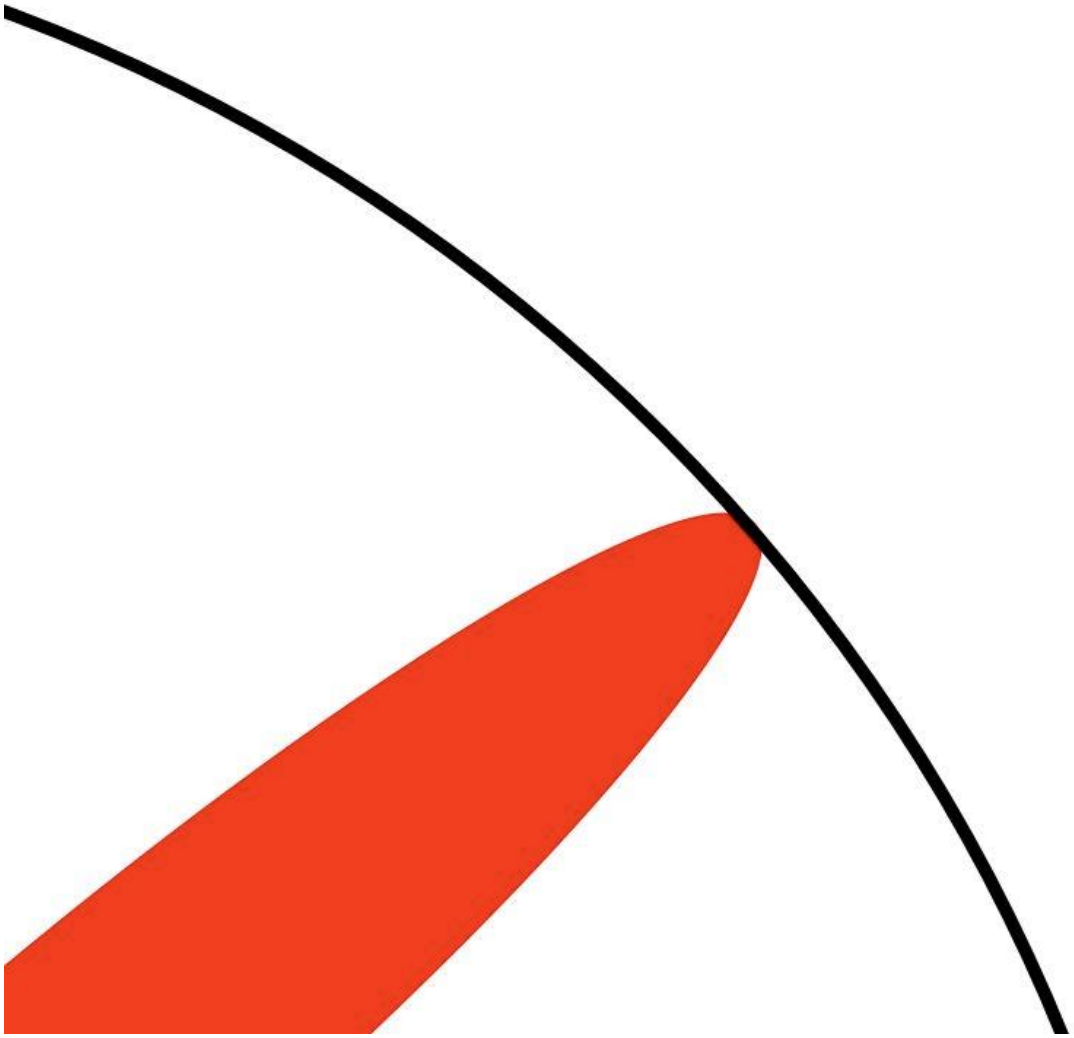
The Illustrated Guide to a Ph.D (Might, 2010)



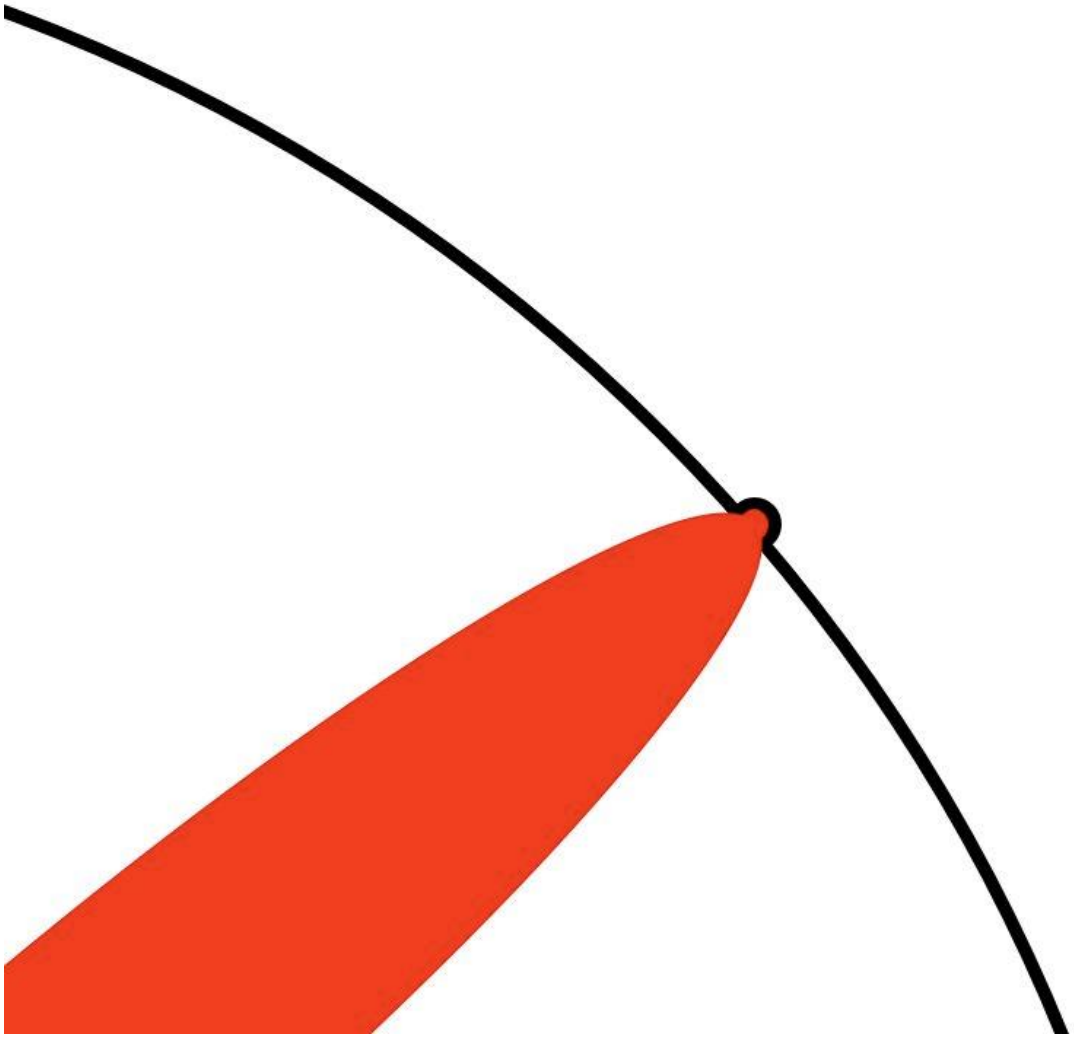
The Illustrated Guide to a Ph.D (Might, 2010)



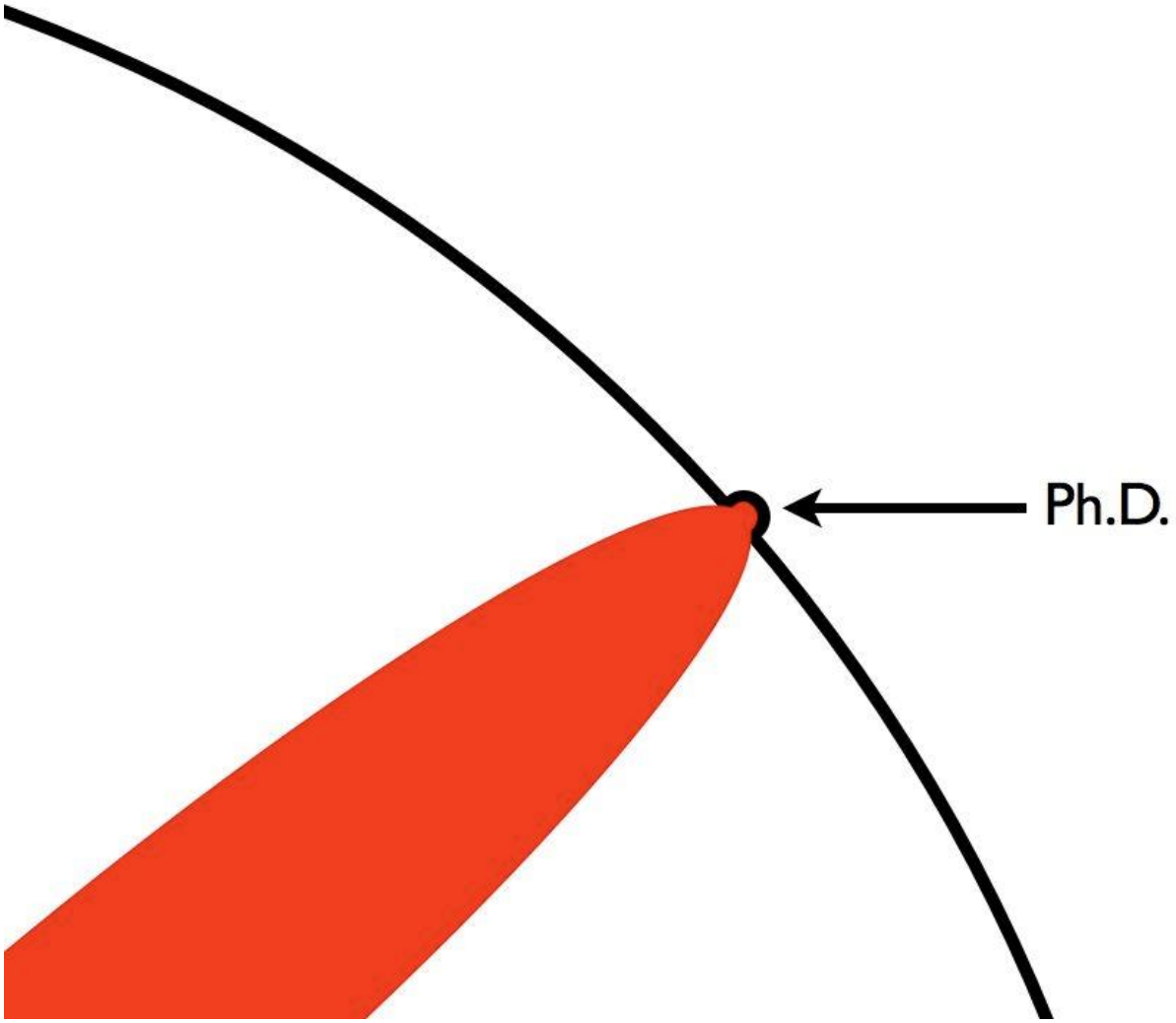
The Illustrated Guide to a Ph.D (Might, 2010)



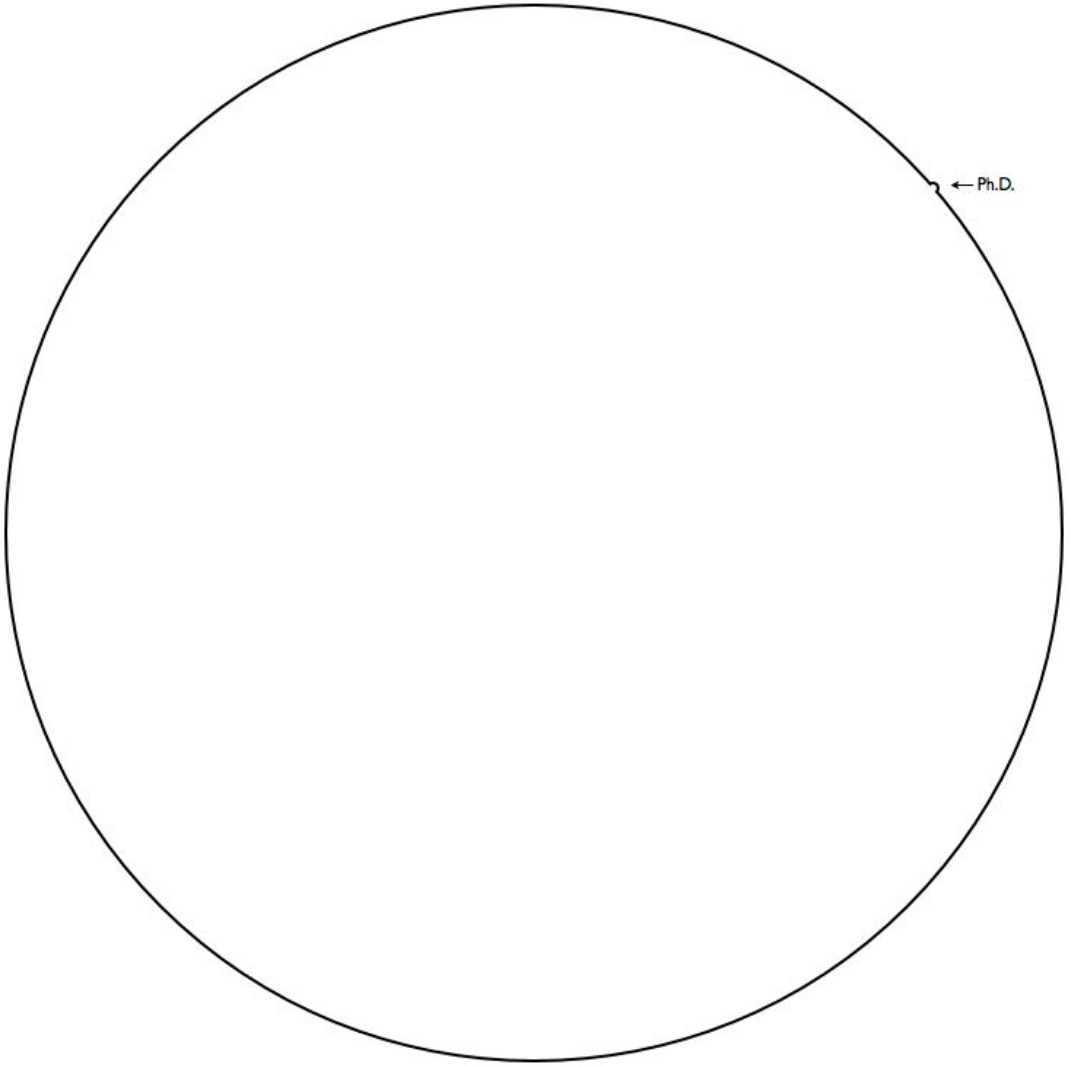
The Illustrated Guide to a Ph.D (Might, 2010)



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

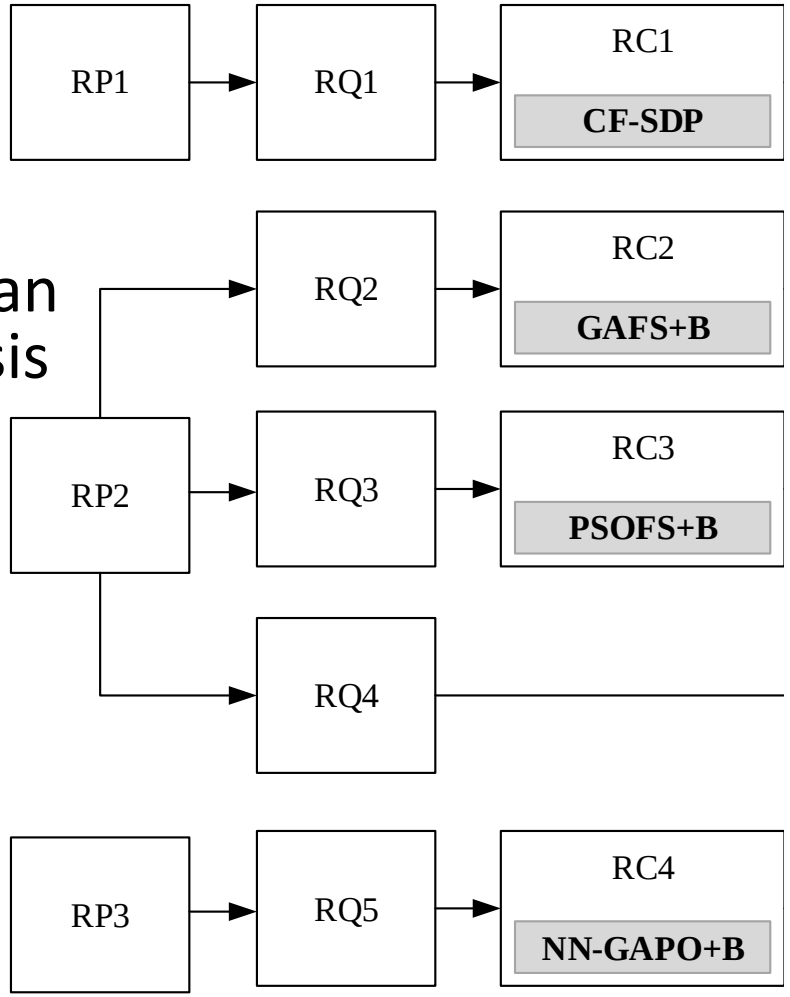
- Jangan terlalu banyak dilakukan kecuali dalam keadaan:
  - Kita tidak bisa mengakses publikasi asli
  - Bahasa asli publikasi bukan bahasa inggris (sulit dipahami)

MITOS 10

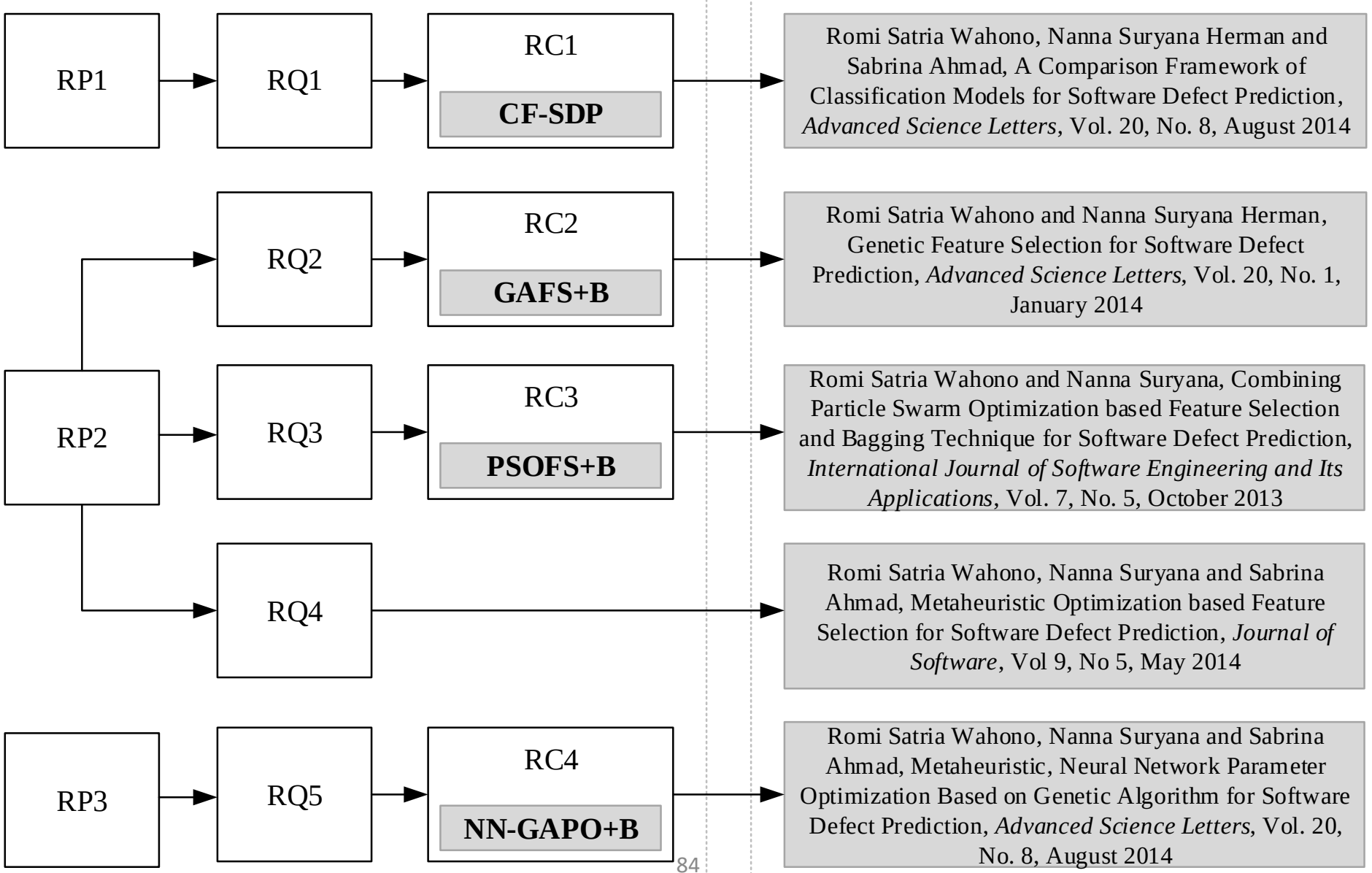
Satu Hasil Eksperimen Penelitian Bisa Jadi Banyak Paper dan Dipublikasikan di Banyak Jurnal

Pola RP – RQ – RC pada Penelitian

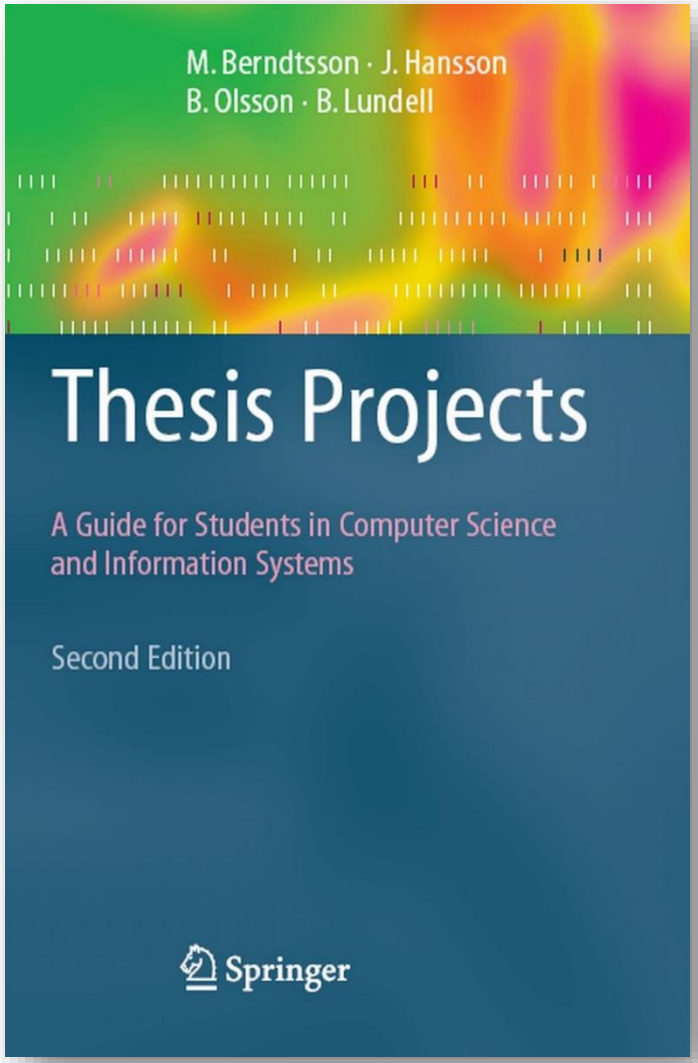
- Research Problem (RP) atau masalah penelitian adalah alasan kita melakukan penelitian
- Satu RP bisa coba dipecahkan dengan banyak cara/metode/solusi/hipotesis (Research Question (RQ))
- Satu RQ akan membentuk satu kontribusi ke pengetahuan (Research Contribution (RC))
- Satu RC akan menjadi satu paper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



Software Defect Prediction Framework based on Hybrid Metaheuristic Optimization Methods



Reference



Romi Satria Wahono
http://romisatriawahono.net
romi@romisatriawahono.net
081586220090