UNIVERSITAS INDONESIA MEMBANGUN

KODE INA047 - PPT - SESI 9
NAMA METODOLOGI PENELITIAN

SYAM GUNAWAN

3. Literature Review

3.1 Literatur Ilmiah

3.2 Teknik Mengelola Paper

3.3 Teknik Mereview Paper

3.1 Literatur Ilmiah

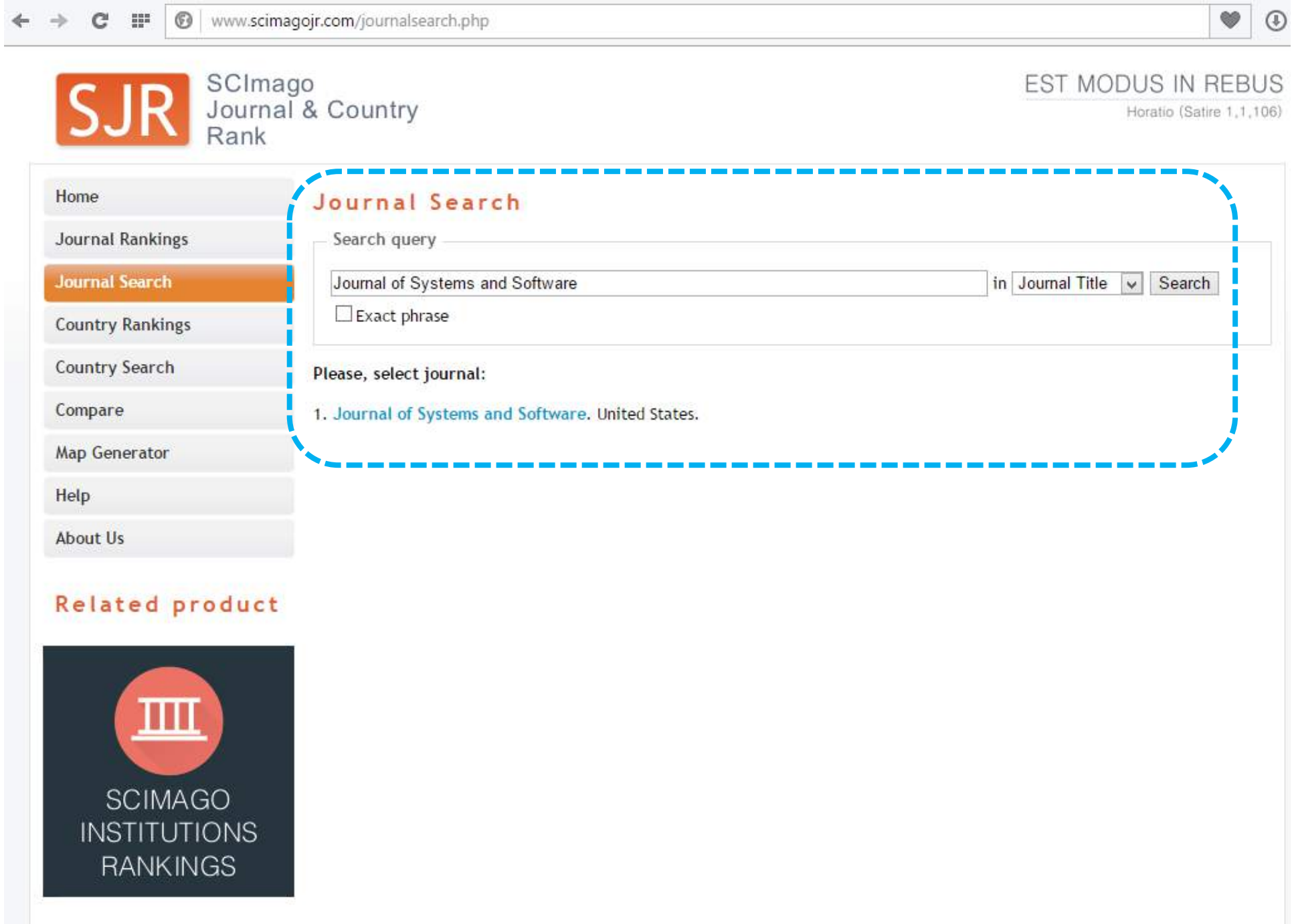
Manfaat Mereview Literatur

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

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*


www.scimagojr.com/journalsearch.php
SJR
SCImago
Journal & Country
Rank
EST MODUS IN REBUS
Horatio (Satire 1,1,106)
Home
Journal Rankings
Journal Search
Country Rankings
Country Search
Compare
Map Generator
Help
About Us
Journal Search
Search query
Journal of Systems and Software
in
Journal Title
Search
Exact phrase
Please, select journal:
1. Journal of Systems and Software. United States.
Related product
SCIMAGO
INSTITUTIONS
RANKINGS

www.scimagojr.com/journalsearch.php

SJR SCImago Journal & Country Rank

EST MODUS IN REBUS
Horatio (Satire 1,1,106)

- Home
- Journal Rankings
- Journal Search
- Country Rankings
- Country Search
- Compare
- Map Generator
- Help
- About Us

Journal Search

Search query

in Journal Title Search

Exact phrase

Journal of Systems and Software

Country: United States

Subject Area: Computer Science

Subject Category:

Category	Quartile (Q1 means highest values and Q4 lowest values)														
	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
Hardware and Architecture	Q2	Q2	Q1	Q2	Q2	Q2	Q1	Q2	Q2	Q2	Q1	Q2	Q1	Q1	Q1
Information Systems	Q3	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q2
Software	Q3	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q2	Q2

Publisher: Elsevier Inc.. **Publication type:** Journals. **ISSN:** 01641212

Coverage: 1979-2014

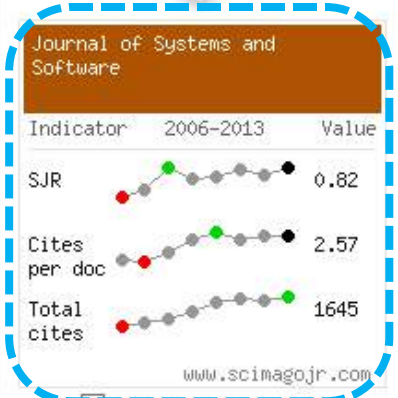
H Index: 60

Scope:

The Journal of Systems and Software publishes papers covering all aspects of programming methodology, software engineering, and related hardware-software-systems issues. [...]

Show full scope

Show this information in your own website



Display Journal title

Journal of Systems and Software

The Journal of Systems and Software publishes papers covering all aspects of programming methodology, software engineering, and related hardware-software-systems issues. Topics of interest include, but...

View full aims and scope

Editor-in-Chief, Software Architecture Papers: H. van Vliet

View full editorial board

Guide for Authors

Submit Your Paper

Track Your Paper

Order Journal

Access Full Text

Impact Factor: 1.277

5-Year Impact Factor: 1.282

Imprint: ELSEVIER

ISSN: 0164-1212

News

Publish multimedia data in your articles

SciVerse App: Expert Search

VIEW ALL

Guidelines

Charting a Course for a Successful Research Career

Ways to Use Journal Articles Published by Elsevier: A Practical Guide

Call for Papers

Special issue on "Trustworthy Software Systems for the Digital Society"

Special Issue on Variability in Software Architecture

VIEW ALL

Most Downloaded Articles

ScienceDirect

Organisasi Yang Mengindeks Journal

1. Thomson Reuters Web of Science
   - Since 1963, formerly produced by ISI, 12032 journals are indexed
   - Pengindeks journal yang memiliki level paling baik
   - *http://wokinfo.com*
2. Scopus
   - Launched by Elsevier in 2004, 20000 journals, conference papers and other are indexed
   - Pengindeks journal level standard, biasa untuk syarat menyelesaikan PhD
   - *http://scopus.com*
3. Google Scholar
   - Launched in 2004, mengindeks semua publikasi ilmiah yang online
   - *http://scholar.google.com*

** Organisasi pengindeks journal selain di atas (EBSCO, DBLP, ProQuest, dsb), boleh dikatakan selevel dengan Google Scholar*

Algoritma Perangkingan Journal

1. Journal Impact Factor (JIF)
   - Data source: ISI Web of Science
2. Eigenfactor Score (ES)
   - Data source: ISI Web of Science
   - *http://www.eigenfactor.org*
3. Scimago Journal Rank (SJR)
   - Data source: Scopus
   - *http://www.scimagojr.com*
4. Source Normalized Impact per Paper (SNIP)
   - Data source: Scopus
   - *http://www.journalindicators.com*
5. h-index
   - Data source: Google Scholar
   - *http://scholar.google.com/intl/en/scholar/metrics.html*

** JIF adalah algoritma yang digunakan oleh ISI, sedangkan SJR adalah algoritma yang digunakan oleh SCOPUS*

Insentif Publikasi Internasional (ITB)

No	Kategori Publikasi	Besar Insentif (*)
1	Jurnal ilmiah internasional terindeks ISI Thomson Reuters dengan IF**):	
	a. IF ≥ 30	Rp. 25.000.000
	b. 20 ≤ IF < 30	Rp. 19.000.000
	c. 10 ≤ IF < 20	Rp. 15.000.000
	d. 5 ≤ IF < 10	Rp. 13.000.000
	e. 1 ≤ IF < 5	Rp. 12.000.000
	f. 0 < IF < 1	Rp. 11.000.000
2	Jurnal ilmiah internasional terindeks Scopus	Rp. 10.000.000
3	Jurnal ilmiah internasional terindeks selain ISI Thomson Reuters dan Scopus	Rp. 5.000.000 s/d Rp. 7.500.000
4	Prosiding Terindeks di CPCI Thomson Reuters atau Scopus	Rp. 3.000.000

Sumber Pencarian Literatur

GRATIS

Journal

- http://sci-hub.io
- http://libgen.org
- http://scholar.google.com
- http://citeseer.ist.psu.edu

Buku

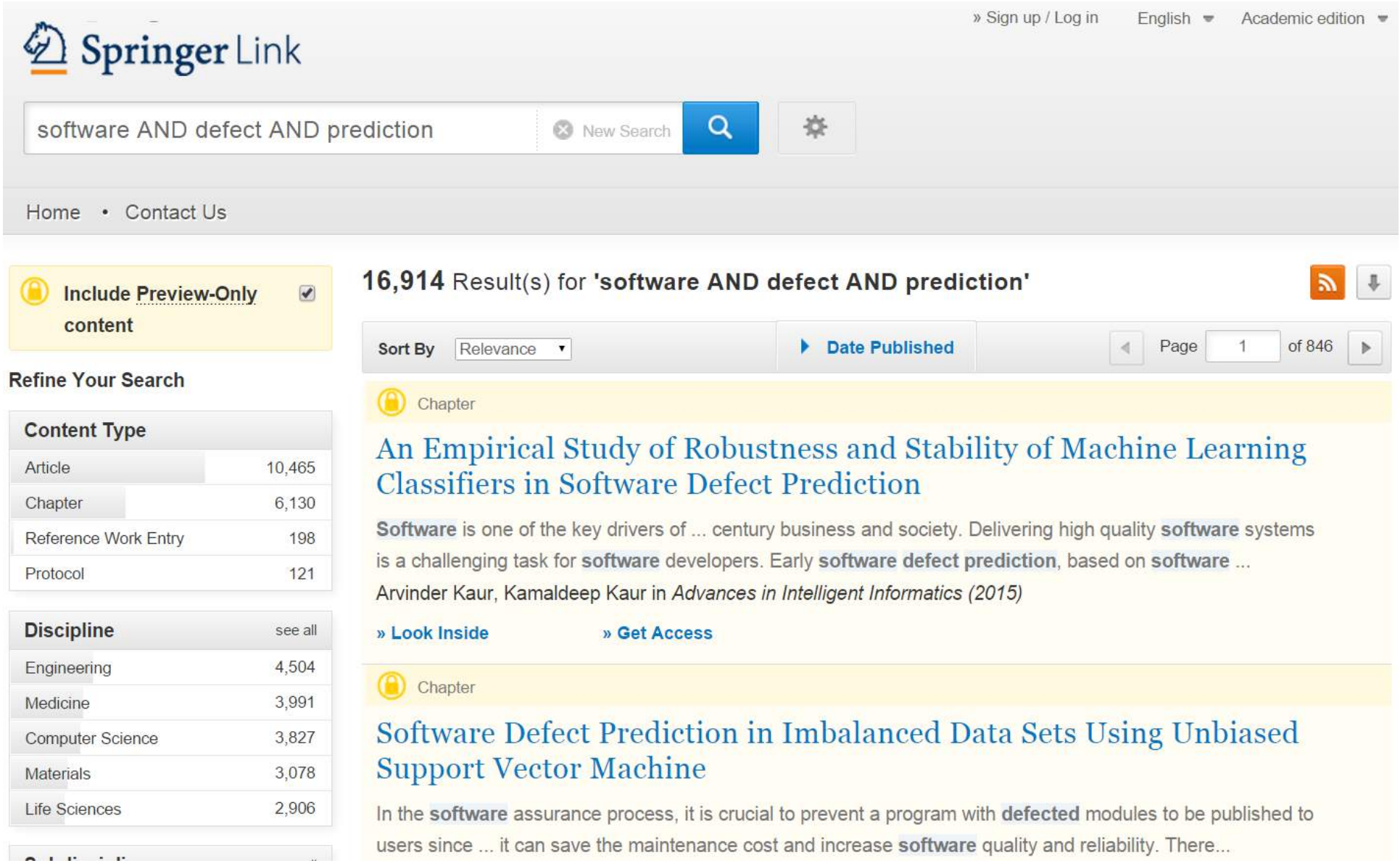
- http://bookzz.org
- http://learnr.pro

BERBAYAR

Journal

- http://sciencedirect.com
- http://www.ebscohost.com
- http://link.springer.com
- http://ieeexplore.ieee.org
- http://dl.acm.org

link.springer.com → sci-hub.io



A dynamic capture–recapture model for software defect prediction

Robert Bucholz · Phillip A. Laplante

Received: 15 May 2009 / Accepted: 13 July 2009 / Published online: 30 July 2009

Abstract Several models have been developed that attempt to predict the total number of defects in a software product. One such approach uses the capture–recapture model, a technique employed by biologists for predicting wildlife populations. In this method once the software is built and defects begin to be identified a prediction can be made for the total number of software defects present. But capture–recapture models rely on expert inspectors and the technique cannot be employed once the software has been released. The work reported here extends the capture–recapture technique to the post-inspection phase and to where inspection data is unavailable, by using user defect reports. The proposed technique does not rely on expert inspectors and is particularly suitable for open source software.

Keywords Software defect prediction · Fault content estimation · Capture–recapture

1 Capture–recapture models

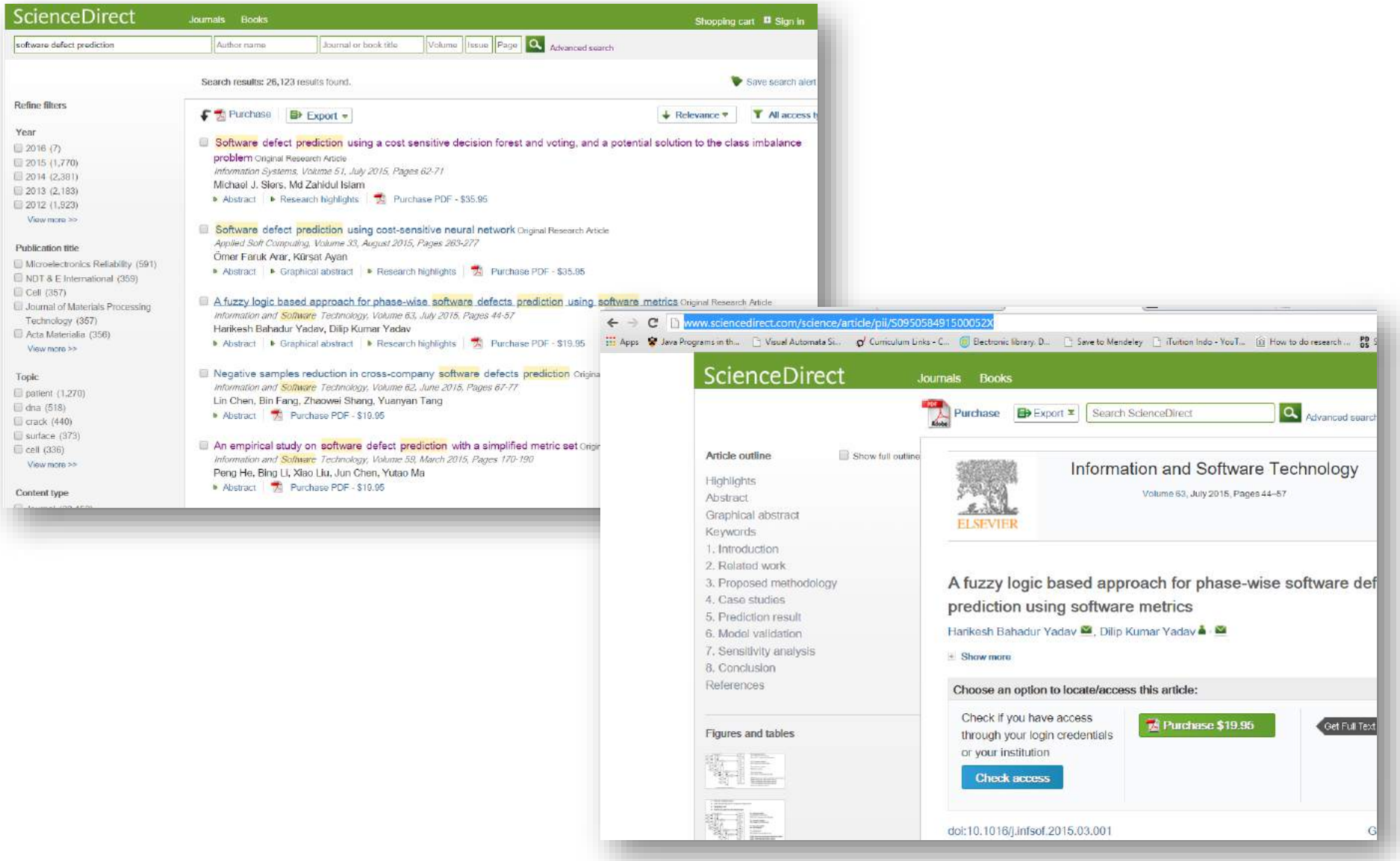
First proposed by French Mathematician Laplace in 1786 [4], the capture–recapture method for population estimates, has been used by biologists for many years and, more recently, by software engineers [3] to estimate the number of software tagged wolves redistribute throughout the park. After this period of time the park manager conducts a second capture event in which n_2 wolves are captured. If the actual number of wolves in the park is close to n_1 then most of the animals captured during the second event will be tagged recaptures. If the actual population of wolves is much greater than n_1, then most of the recaptures will not be tagged animals.

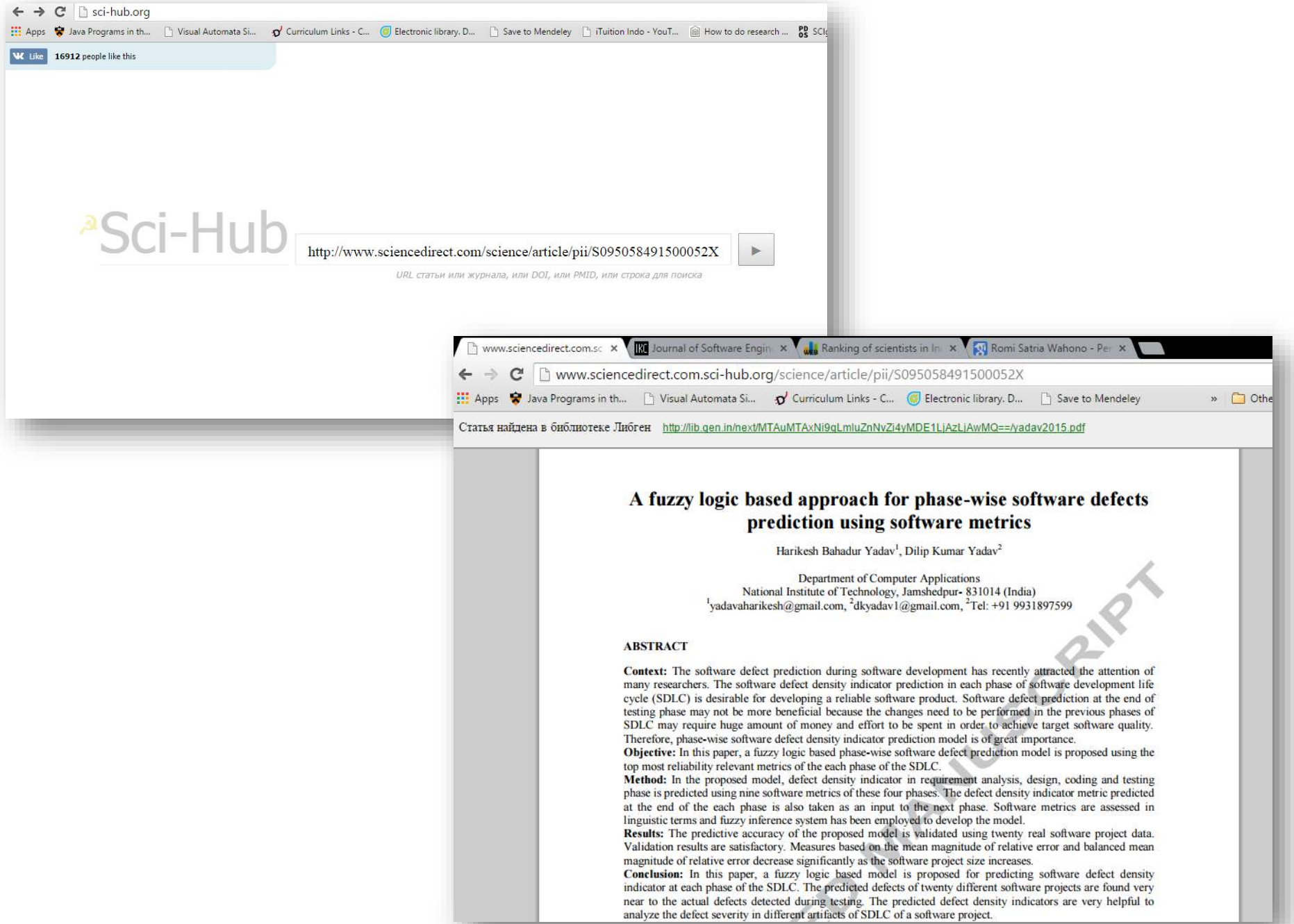
Mills [3] proposed using the capture–recapture method for software defect prediction by seeding the software with a known number of defects and seeing how many are found during code inspection. Eick et al. [2] expanded this method by using two software inspectors to inspect the same code. The number of faults found by each inspector, n_1 and n_2, respectively, along with m_2, the number of defects common to n_1 and n_2, will predict the number of total defects, N, using the Lincoln–Peterson Estimator [6],

$$N = n_1 \cdot n_2 / m_2 \quad (1)$$

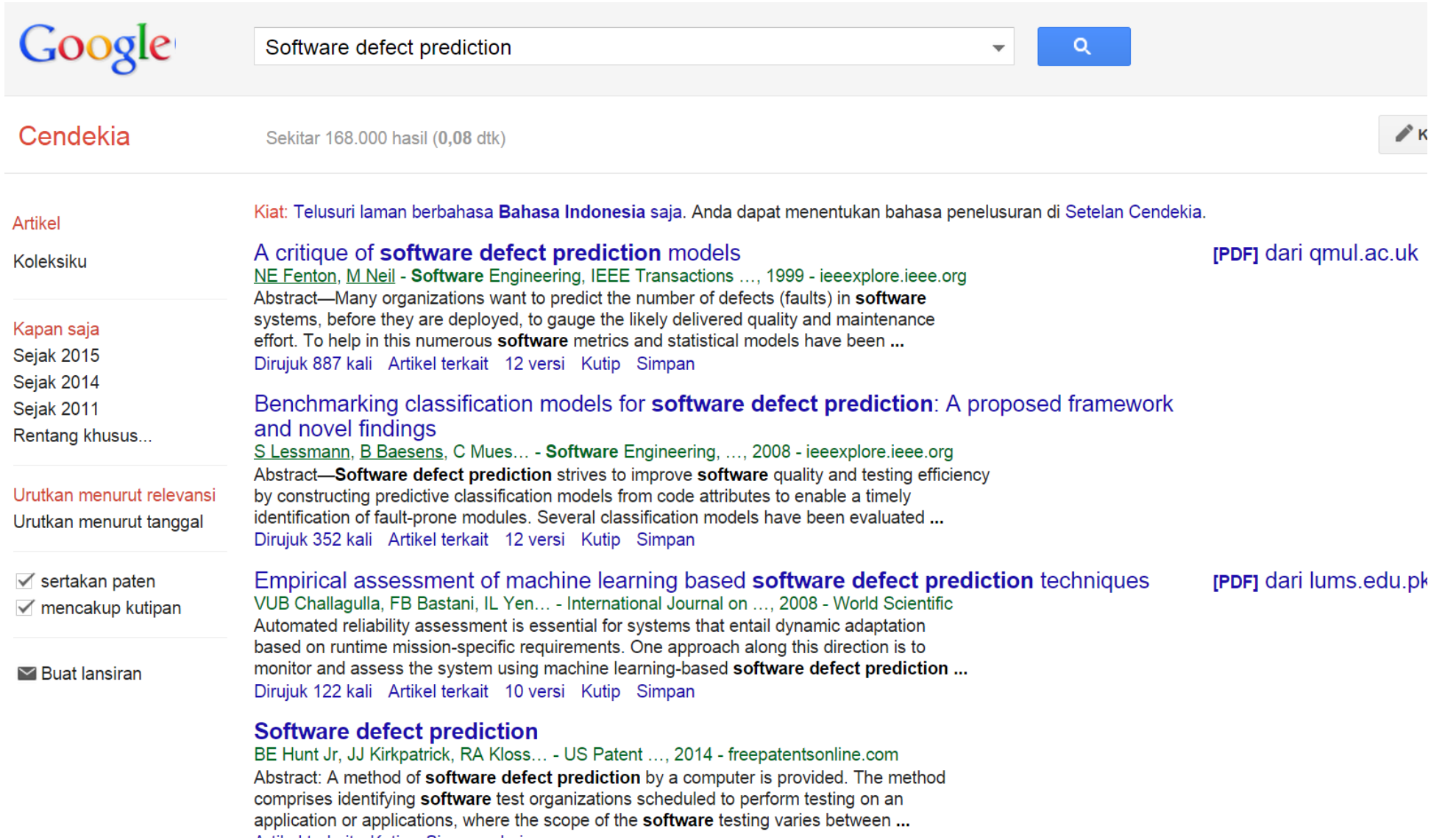
Several variants of the capture–recapture models and their corresponding estimators have been developed differing based on whether the population being modeled is closed (remains the same during the counting period), or open (which allows for births and deaths). The prediction models used in software engineering are closed since, typically, no new defects are introduced or resolved between inspections.

sciencedirect.com → sci-hub.io




sci-hub.org
Apps
Java Programs in th...
Visual Automata Si...
Curriculum Links - C...
Electronic library. D...
Save to Mendeley
iTuition Indo - YouT...
How to do research ...
Like
16912 people like this
Sci-Hub
http://www.sciencedirect.com/science/article/pii/S095058491500052X
URL статьи или журнала, или DOI, или PMID, или строка для поиска
www.sciencedirect.com.sc
Journal of Software Engin
Ranking of scientists in In
Romi Satria Wahono - Per
www.sciencedirect.com.sci-hub.org/science/article/pii/S095058491500052X
Apps
Java Programs in th...
Visual Automata Si...
Curriculum Links - C...
Electronic library. D...
Save to Mendeley
Othe
Статья найдена в библиотеке Либген
http://lib.gen.in/next/MTAuMTAxNi9qLmluZnNvZi4yMDE1LjAzLjAwMQ==/yadav2015.pdf
A fuzzy logic based approach for phase-wise software defects prediction using software metrics
Harikesh Bahadur Yadav[1], Dilip Kumar Yadav[2]
Department of Computer Applications
National Institute of Technology, Jamshedpur- 831014 (India)
[1]yadavaharikesh@gmail.com, [2]dkyadav1@gmail.com, [2]Tel: +91 9931897599
ABSTRACT
Context: The software defect prediction during software development has recently attracted the attention of many researchers. The software defect density indicator prediction in each phase of software development life cycle (SDLC) is desirable for developing a reliable software product. Software defect prediction at the end of testing phase may not be more beneficial because the changes need to be performed in the previous phases of SDLC may require huge amount of money and effort to be spent in order to achieve target software quality. Therefore, phase-wise software defect density indicator prediction model is of great importance.
Objective: In this paper, a fuzzy logic based phase-wise software defect prediction model is proposed using the top most reliability relevant metrics of the each phase of the SDLC.
Method: In the proposed model, defect density indicator in requirement analysis, design, coding and testing phase is predicted using nine software metrics of these four phases. The defect density indicator metric predicted at the end of the each phase is also taken as an input to the next phase. Software metrics are assessed in linguistic terms and fuzzy inference system has been employed to develop the model.
Results: The predictive accuracy of the proposed model is validated using twenty real software project data. Validation results are satisfactory. Measures based on the mean magnitude of relative error and balanced mean magnitude of relative error decrease significantly as the software project size increases.
Conclusion: In this paper, a fuzzy logic based model is proposed for predicting software defect density indicator at each phase of the SDLC. The predicted defects of twenty different software projects are found very near to the actual defects detected during testing. The predicted defect density indicators are very helpful to analyze the defect severity in different artifacts of SDLC of a software project.
MANUSCRIPT

http://scholar.google.com



Tugas

- Jelaskan arti Literatur Review
- Berikan contoh literature paper ilmiah

TERIMA KASIH