Experimentation and Implementation

Module name	Experimentation and Implementation	
Module level	Undergraduate	
Code	IF221207	
Courses (if applicable)	Experimentation and Implementation	
Semester	5/6	
Lecturer	Yisti Vita Via, S.ST, M.Kom (PIC) Sugiarto, S.Kom., M.Kom.	
Language	Bahasa Indonesia and English	
Relation to curriculum	Elective; 5th or 6th semester	
Type of teaching, contact hours	Lectures, < 20 students	
Teaching Methods	Discussion, case-study, simulation, project-based learning, problem-based learning.	
Workload	1. Lectures: 3 sks x 50 = 150 minutes (2 hours 30 minutes) per week. 2. Exercises and Assignments: 3 x 60 = 180 minutes (3 hours) per week. 3. Private study: 3 x 60 = 180 minutes (3 hours) per week	
Credit points	3 credit points (sks)	
Requirements according to the examination regulations	A student must have attended at least 80% of the lectures to sit in the exams.	
Mandatory prerequisites	Software Engineering	
Courses description	Experimentation and implementation is a course that studies the steps needed to or prepared to conduct trials of software/computer applications, methods that can be applied in the implementation of these trials to software/computer applications, creating a report from the testing model whose scenarios have been created in the process of testing software/computer applications, and creating and implementing a report in the IEEE format for each selected case study.	
Learning outcomes and their corresponding PLOs	After completing this module, a student is expected to:	
	CO1 Students are able to understand concepts of analytical, numerical, linear, and nonlinear methods..	PLO9,PLO10
	CO2 Students are able to understand and apply solutions for linear and nonlinear equations using a programming language.	PLO9,PLO10
	CO3 Students are able to understand and implement differentiation and integration concepts using a programming language.	PLO9,PLO10
Content	The material studied by students in this course includes: software/computer application testing, software/application development, testing using white-box testing methods, testing using black-box testing methods, usability testing methods, object-oriented testing models (OOA/OOD), support tools for testing, presentation of the progress of trial and implementation projects from each team/group according to the case study, testing planning and	

	completion estimation, testing process control, implementation of testing templates used in the field, and the creation of planning and execution reports for each project group.
Media employed	LCD, whiteboard, websites, books (as references), online meeting, etc.
Assessments and Evaluation	One written Midterm assessment (60 minutes) and one final oral exam (30 minutes), two short computer-based quizzes, takehome written assignments
Study and examination requirements and forms of examination	The final grade in the module is composed of: • Two short computer-based quizzes: 15% x 2 = 30% • Take-home written assignments : 15% • Written Midterm assessment: 25% • Final oral exam: 30% Students must have a final grade of 55.6% or higher to pass.
Reading List	● C. Kaner, J. Bach, and B. Pettichord, Lessons learned in software testing: a context-driven approach. Wiley, 2020. ● B. Pettichord and B. Marick, Software testing: a comprehensive approach. John Wiley & Sons, 2020. ● L. Hohmann, Beyond the project: the implementor's guide to software adoption. Addison-Wesley Professional, 2020. ● P. Ammann and J. Offutt, Introduction to software testing, 3rd ed. Cambridge University Press, 2023. ● R. S. Pressman and B. Maxim, Software engineering: a practitioner's approach, 9th ed. McGraw-Hill Education, 2020. ● P. Leloudas, *Software testing strategies*. BPB Publications, 2024, 369 pp. [Online]. Available: https://portal.igpublish.com/iglibrary/search/BPB0000742.html ● N. Kaul, *Implementing automated software testing*. Arcler Press, 2023, 280 pp. [Online]. Available: https://portal.igpublish.com/iglibrary/search/ARCLER0001250.html