	Universitas Negeri Surabaya Faculty of Mathematics and Natural Sciences Undergraduate Mathematics Study Program	Document Code

SEMESTER LEARNING PLAN

Courses	CODE	Course Family	Credit Weight			SEMESTER	Compilation Date
Data Structures and Algorithms	4420103132	Compulsory Study Program Subjects	T=3	P=0	ECTS=4.77	4	January 22, 2024

AUTHORIZATION	SP Developer	Course Cluster Coordinator	Study Program Coordinator
	Riska Wahyu Romadhonia, M.Sc.	.....	Prof. Dr. Raden Sulaiman, M.Si.

Learning model	**Project Based Learning**

Program Learning Outcomes (PLO)

PLO study program that is charged to the course

Program Objectives (PO)

PO - 1	Able to complete tasks in groups with full empathy as fellow citizens and religious communities by utilizing techno-ecopreneur.
PO - 2	Able to analyze and solve mathematical problems based on understanding data structures and algorithms
PO - 3	Able to implement and simulate mathematical problems related to data structures and algorithms into computer programs
PO - 4	Able to answer problems given, prepare answers/reports on problems given in writing and/or communicate them orally
PO - 5	Able to solve given problems by utilizing computer programs
PO - 6	Able to demonstrate knowledge of the concepts of abstract data structures, linked, stack, queue, tree, sorting algorithms, and searching algorithms and their applications

PLO-PO Matrix

P.O
PO-1
PO-2
PO-3
PO-4
PO-5
PO-6

PO Matrix at the end of each learning stage (Sub-PO)

P.O	Week															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
PO-1																
PO-2																
PO-3																
PO-4																
PO-5																
PO-6																

Short Course Description	Study the concepts of data structures and algorithms that can be applied to computer programs. The discussion begins with basic data structures which include linked-list, stack, queue, and tree. Then we discuss simple algorithms that use these data structures, such as searching and sorting. Next we discuss algorithms and data structures that are suitable for solving problems in everyday life through individual and group task-based learning, presented in theory and practice and demonstrating the results in computer programs.
References	**Main :**

	1. Goodrich, M.T., Tamassia R., Goldwasser M.H. 2013. Data Structures and Algorithms in Phyton. USA: John Wiley&Sons
	Supporters:
	1. Baka, Benjamin. 2017. Python Data Structures and Algorithms. Birmingham: Packt Publishing Ltd 2. Bullinaria, J. 2019. Lecture Notes for Data Structures and Algorithms. University of Birmingham, UK 3. Lambert, Kenneth A. 2019. Fundamental of Python: Data Structure, 2nd Ed. Boston: Cengange Learning Inc. 4. Padmaja, B.2017. Lecture Notes On Data Structure. Institute of Aeronautical Engineering
Supporting lecturer	Dr. Atik Wintarti, M.Kom. Dr. Dian Savitri, S.Si., M.Si. Dr. Elly Matul Imah, M.Kom. Riska Wahyu Romadhonia, S.Si., M.Sc.

Week-	**Final abilities of each learning stage (Sub-PO)**	**Evaluation**		**Help Learning, Learning methods, Student Assignments, [Estimated time]**		**Learning materials [References]**	**Assessment Weight (%)**
		Indicator	**Criteria & Form**	**Offline (*offline*)**	**Online (*online*)**		
(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**	**(8)**
1	Able to differentiate data types and abstract data structures	1. Explain data types and abstract data structures (arrays, structures, and classes) 2. Using abstract data structures in computer programs	**Criteria:** Observation of student activities in class (Non-Test) **Form of Assessment** : Participatory Activities	• Scientific approach: observing, asking, exploring • Method: lecture, discussion, question and answer, giving assignments • Learning strategy: accentuation of information processing (cognitive) 3 X 50		**Material:** Struct **Reference:** *Lambert, Kenneth A. 2019. Fundamentals of Python: Data Structure, 2nd Ed. Boston: Cengange Learning Inc.*	2%
2	Able to integrate linked-list data structures and their forms	1. Explain the linked-list data structure, along with its forms (double linked-list, circular linked-list) 2. Using the linked-list data structure and its operations in computer programs	**Criteria:** Observation of student activities in class (Non-Test) and Practicum **Form of Assessment** : Participatory Activities, Practical Assessment	• Scientific approach: observing, asking, exploring • Method: lecture, discussion, question and answer, giving assignments • Learning strategy: accentuation of information processing (cognitive) 3 X 50		**Material:** Linked-List **Bibliography:** *Padmaja, B. 2017. Lecture Notes On Data Structure. Institute of Aeronautical Engineering*	3%
3	Able to integrate Stack data structures and their implementation	1. Mention the characteristics of the stack data structure 2. Implementing push operations on the stack in computer programs 3. Implementing the pop operation on the stack and how to declare it in the program	**Criteria:** Observation of student activities in class (Non-Test) **Form of Assessment** : Participatory Activities	• Scientific approach: observing, asking, exploring • Method: lecture, discussion, question and answer, giving assignments • Learning strategy: accentuation of information processing (cognitive) 3 X 50		**Material:** Stack **Library:** *Bullinaria, J. 2019. Lecture Notes for Data Structures and Algorithms. University of Birmingham, UK*	2%

4	Able to integrate queue data structures and their implementation	1. Mention the characteristics of the queue data structure 2. Implementing the enqueue operation on queues in computer programs 3. Implementing the dequeue operation on queues in computer programs	**Criteria:** Observation of student activities in class (Non-Test) and Practicum **Form of Assessment** : Practical Assessment	• Scientific approach: observing, asking, exploring • Method: lecture, discussion, question and answer, giving assignments • Learning strategy: accentuation of information processing (cognitive) 3 X 50		**Material:** Queue **Reference:** *Bullinaria, J. 2019. Lecture Notes for Data Structures and Algorithms. University of Birmingham, UK*	3%
5	Able to integrate graph data structures and their application	1. Mention the characteristics of graph data structures 2. Implementing matrix adjance operations on graphs in computer programs 3. Implementing adjency list operations on graphs in computer programs	**Criteria:** Observation of student activities in class (Non-Test) and Practicum **Form of Assessment** : Practical Assessment	• Scientific approach: observing, asking, exploring • Method: lecture, discussion, question and answer, giving assignments • Learning strategy: accentuation of information processing (cognitive) 3 X 50		**Material:** Graph **References:** *Goodrich, MT, Tamassia R., Goldwasser MH 2013. Data Structures and Algorithms in Python. USA: John Wiley&Sons*	3%
6	Able to integrate tree data structures and their application	1. Mention the characteristics of tree data structures 2. State the meaning of tree composition (root, leaf, children, ancestors, descendants, interior nodes) 3. Implement basic tree operations in Python programs	**Criteria:** Observation of student activities in class (Non-Test) and Practicum **Form of Assessment** : Participatory Activities, Practical Assessment	• Scientific approach: observing, asking, exploring • Method: lecture, discussion, question and answer, giving assignments • Learning strategy: accentuation of information processing (cognitive) 3 X 50		**Material:** Tree **References:** *Goodrich, MT, Tamassia R., Goldwasser MH 2013. Data Structures and Algorithms in Python. USA: John Wiley&Sons*	2%
7	Able to integrate binary-tree data structures and traversal in binary-tree	1. Explain the meaning of a binary tree, and its differences from a general tree 2. Explain the differences between preorder traversal, inorder traversal, and postorder traversal 3. Implementing binary tree traversal operations in Python programs	**Criteria:** Observation of student activities in class (Non-Test) and Practicum **Form of Assessment** : Practical Assessment	• Scientific approach: observing, asking, exploring • Method: lecture, discussion, question and answer, giving assignments • Learning strategy: accentuation of information processing (cognitive) 3 X 50		**Material:** Binary Tree **References:** *Goodrich, MT, Tamassia R., Goldwasser MH 2013. Data Structures and Algorithms in Python. USA: John Wiley&Sons*	3%

8	Midterm exam	able to complete UTS honestly, correctly and on time	**Criteria:** UTS test **Form of Assessment** : Test	Written Exam 3 X 50		**Material:** Week 1 to Week 7 **Material References:** *Goodrich, MT, Tamassia R., Goldwasser MH 2013. Data Structures and Algorithms in Python. USA: John Wiley&Sons*	20%
9	Able to integrate algorithms and their complexity	1. Mention the meaning of algorithms 2. Explaining an algorithm about a mathematical topic 3. Analyzing the running time of an algorithm with certain complexity (N2, Nlog N, N)	**Criteria:** Observation of student activities in class (Non-Test) **Form of Assessment** : Participatory Activities	• Scientific approach: observing, asking, exploring • Method: lecture, discussion, question and answer, giving assignments • Learning strategy: accentuation of information processing (cognitive) 3 X 50		**Material:** Algorithmic Complexity **References:** *Lambert, Kenneth A. 2019. Fundamentals of Python: Data Structure, 2nd Ed. Boston: Cengange Learning Inc.*	2%
10	Able to integrate sorting algorithms and their implementation	1. Explains sorting algorithms, including Linear-sorting algorithms, Selection Sort, Insertion Sort, Bubble Sort, Merge Sort, Quick Sort, and Bucket Sort. 2. Analyze and implement sorting algorithms in programs	**Criteria:** Observation of student activities in class (Non-Test) and practicum **Form of Assessment** : Practical Assessment	• Scientific approach: observing, asking, exploring • Method: lecture, discussion, question and answer, giving assignments • Learning strategy: accentuation of information processing (cognitive) 3 X 50		**Material:** Sorting Algorithm **References:** *Goodrich, MT, Tamassia R., Goldwasser MH 2013. Data Structures and Algorithms in Python. USA: John Wiley&Sons*	3%
11	Able to integrate searching algorithms and their application	1. Explains searching algorithms, including Linear search algorithms and Binary search trees. 2. Analyze and implement searching algorithms in programs	**Criteria:** Observation of student activities in class (Non-Test) **Form of Assessment** : Participatory Activities	• Scientific approach: observing, asking, exploring • Method: lecture, discussion, question and answer, giving assignments • Learning strategy: accentuation of information processing (cognitive) 3 X 50		**Material:** Searching Algorithm **References:** *Goodrich, MT, Tamassia R., Goldwasser MH 2013. Data Structures and Algorithms in Python. USA: John Wiley&Sons*	2%

12	Able to integrate searching algorithms: Depth First Search and Best First Search	1. Explains searching algorithms, including the Depth First Search (DFS) and Best First Search (BFS) algorithms. 2. Analyze and implement DFS and BFS algorithms in programs	**Criteria:** Observation of student activities in class (Non-Test) and practicum **Form of Assessment** : Participatory Activities, Practical Assessment	• Scientific approach: observing, asking, exploring • Method: lecture, discussion, question and answer, giving assignments • Learning strategy: accentuation of information processing (cognitive) 3 X 50		**Material:** Searching Algorithm **References:** *Goodrich, MT, Tamassia R., Goldwasser MH 2013. Data Structures and Algorithms in Python. USA: John Wiley&Sons* **Material:** DFS and BFS Algorithm **References:** *Goodrich, MT, Tamassia R., Goldwasser MH 2013. Data Structures and Algorithms in Python. USA: John Wiley&Sons*	3%
13	Able to design computer programs to solve problems related to mathematics	1. Students are able to apply the concept of dynamic data structures (linked-list / stack / queue / graph / tree) to the problems raised 2. Students are able to apply the concept of sorting and searching algorithms to the problems raised	**Criteria:** Observation of student activities in class (Non-Test) **Form of Assessment** : Project Results Assessment / Product Assessment	Learning is carried out offline with the following PJBL stages: • Providing basic questions regarding the urgency of the problems raised • Determining the boundaries of the problem in creating the application • Types of abstraction of programming concepts (data structures and algorithms) that will be used • Agreeing on the implementation schedule in working on the 3 X 50 project		**Material:** Applied to everyday problems **References:** *Goodrich, MT, Tamassia R., Goldwasser MH 2013. Data Structures and Algorithms in Python. USA: John Wiley&Sons*	7%
14	Able to design computer programs to solve problems related to mathematics	1. Students are able to apply the concept of dynamic data structures (linked-list / stack / queue / graph / tree) to the problems raised 2. Students are able to apply the concept of sorting and searching algorithms to the problems raised	**Criteria:** Independent Group Work **Form of Assessment** : Project Results Assessment / Product Assessment		Learning is carried out online with the PJBL stages as follows: • Students and their groups independently work on a project according to an approved topic • Students receive guidance from the lecturer if they encounter difficulties in working on their project 3 X 50	**Material:** Applied to everyday problems **References:** *Goodrich, MT, Tamassia R., Goldwasser MH 2013. Data Structures and Algorithms in Python. USA: John Wiley&Sons*	5%

15	Able to design computer programs to solve problems related to mathematics	1. Students are able to apply the concept of dynamic data structures (linked-list / stack / queue / graph / tree) to the problems raised 2. Students are able to apply the concept of sorting and searching algorithms to the problems raised	**Criteria:** Group Progress Presentation **Form of Assessment** : Project Results Assessment / Product Assessment	Learning is carried out offline with the following PJBL stages: • Monitoring the student process in implementing and realizing the project through presentations of the progress of each group and facilitating students in discussions and questions and answers about the project carried out 3 X 50		**Material:** Applied to everyday problems **References:** *Goodrich, MT, Tamassia R., Goldwasser MH 2013. Data Structures and Algorithms in Python. USA: John Wiley&Sons*	10%
16	Final exams	1. Students are able to apply the concept of dynamic data structures (linked-list / stack / queue / graph / tree) to the problems raised 2. Students are able to apply the concept of sorting and searching algorithms to the problems raised	**Criteria:** Final Project Presentation **Form of Assessment** : Project Results Assessment / Product Assessment	Learning is carried out offline with the PJBL stages as follows: • Students present the final results of the project carried out, as well as carry out a demo of the program created with report and presentation provisions according to the template provided. 3 X 50		**Material:** Applied to everyday problems **References:** *Goodrich, MT, Tamassia R., Goldwasser MH 2013. Data Structures and Algorithms in Python. USA: John Wiley&Sons*	30%

Evaluation Percentage Recap: Project Based Learning

No	Evaluation	Percentage
1.	Participatory Activities	12%
2.	Project Results Assessment / Product Assessment	52%
3.	Practical Assessment	16%
4.	Test	20%
		100%

Notes

1. **Learning Outcomes of Study Program Graduates (PLO - Study Program)** are the abilities possessed by each Study Program graduate which are the internalization of attitudes, mastery of knowledge and skills according to the level of their study program obtained through the learning process.
2. **The PLO imposed on courses** are several learning outcomes of study program graduates (CPL-Study Program) which are used for the formation/development of a course consisting of aspects of attitude, general skills, special skills and knowledge.
3. **Program Objectives (PO)** are abilities that are specifically described from the PLO assigned to a course, and are specific to the study material or learning materials for that course.
4. **Subject Sub-PO (Sub-PO)** is a capability that is specifically described from the PO that can be measured or observed and is the final ability that is planned at each learning stage, and is specific to the learning material of the course.
5. **Indicators for assessing** abilities in the process and student learning outcomes are specific and measurable statements that identify the abilities or performance of student learning outcomes accompanied by evidence.
6. **Assessment Criteria** are benchmarks used as a measure or measure of learning achievement in assessments based on predetermined indicators. Assessment criteria are guidelines for assessors so that assessments are consistent and unbiased. Criteria can be quantitative or qualitative.
7. **Forms of assessment:** test and non-test.
8. **Forms of learning:** Lecture, Response, Tutorial, Seminar or equivalent, Practicum, Studio Practice, Workshop Practice, Field Practice, Research, Community Service and/or other equivalent forms of learning.
9. **Learning Methods:** Small Group Discussion, Role-Play & Simulation, Discovery Learning, Self-Directed Learning, Cooperative Learning, Collaborative Learning, Contextual Learning, Project Based Learning, and other equivalent methods.
10. **Learning materials** are details or descriptions of study materials which can be presented in the form of several main points and sub-topics.
11. **The assessment weight** is the percentage of assessment of each sub-PO achievement whose size is proportional to the level of difficulty of achieving that sub-PO, and the total is 100%.
12. TM=Face to face, PT=Structured assignments, BM=Independent study.