INSTITUT TEKNOLOGI TELKOM SURABAYA
FAKULTAS TEKNOLOGI INFORMASI DAN BISNIS
PROGRAM STUDI INFORMATIKA

RENCANA PEMBELAJARAN SEMESTER

Mata Kuliah	Kode MK	Rumpun MK	Semester	Bobot (sks)		Tgl Penyusunan
Perancangan dan Analisis Algoritma	IFC22G4	Pemrograman Terapan	4	T=3	P=1	1 Desember 2022

OTORISASI	Pengembang RPS	Koordinator RMK	Ketua Prodi
	 Vessa Rizky Oktavia, S.Kom., M.Kom.	 Ahmad Wali Satria Bahari Johan, S.S.T., M. Kom.	 Muhammad Dzulfikar Fauzi, S.Kom., M.Cs.

Capaian Pembelajaran (CP)		
	CPL Prodi (Kode S, P, KU, KK)	
	P.4	Menguasai konsep dan prinsip-prinsip algoritma dan pemrograman
	KU.2	Mampu bekerja secara mandiri dan bekerjasama dalam tim yang interdisiplin dan multidisiplin
	KK.4	Mampu merancang dan menganalisa algoritma untuk menyelesaikan permasalahan secara efektif dan efisien menggunakan kaidah-kaidah pemrograman dan bahasa pemrograman yang sesuai
	CPMK (Kode M)	
	M1	Mahasiswa mampu menganalisis problem yang membutuhkan solusi pemecahan algoritma (P.4, KU.2)
	M2	Mahasiswa mampu merancang algoritma yang sesuai untuk memecahkan problem yang telah dianalisis (KU.2, KK.4)
	M3	Mahasiswa mampu menghitung kompleksitas dari algoritma yang diaplikasikan (KU.2, KK.4)
	SUB-CPMK (Kode L)	
	L1	Menjelaskan konsep analisis dan perancangan algoritma (M1)
	L2	Menganalisis problem yang disajikan (M1)
	L3	Menentukan solusi untuk menyelesaikan problem atau persoalan yang diberikan (M1)
	L4	Merancang algoritma untuk menyelesaikan problem (M2)
	L5	Mengklasifikasikan jenis-jenis masalah algoritma (M1, M2)
	L6	Mengaplikasikan algoritma ke dalam program komputer untuk menyelesaikan masalah (M2)
	L7	Menghitung kompleksitas dari sebuah algoritma (M3)
	L8	Membandingkan kompleksitas antar algoritma yang berbeda untuk menyelesaikan sebuah permasalahan (M2, M3)
	L9	Membuat program untuk menyelesaikan sebuah masalah dengan mempertimbangkan efisiensinya (M2, M3)
Deskripsi Singkat Mata	**Deskripsi**	

Kuliah	Mata kuliah Perancangan dan Analisis Algoritma membantu mahasiswa untuk menjelaskan konsep analisis dan perancangan algoritma. Kemudian, mahasiswa juga akan belajar untuk menentukan kompleksitas sebuah algoritma menggunakan notasi asimtotik. Inti dari mata kuliah ini adalah tentang menganalisis dan merancang program menggunakan fungsi rekursif, algoritma Brute Force, Greedy Algorithm, searching dan sorting, BFS dan DFS, dan teori algoritma Divide and Conquer. Selain itu, mahasiswa akan belajar menganalisis dan merancang algoritma yang digunakan untuk menyelesaikan Shortest Path Problem dan menghitung Minimum Spanning Tree. Perkuliahan akan dilakukan di kelas dengan bentuk ceramah, diskusi kelompok/kelas, tanya jawab, presentasi dan di laboratorium dengan bentuk praktikum.
Materi Pembelajaran/Pokok Bahasan	**Bahan Kajian** Perancangan dan Analisis Algoritma **Topik Bahasan** 1. Konsep Analisis dan Perancangan Algoritma 1.1. Ruang lingkup kajian desain dan perancangan algoritma 1.2. Pengenalan Algoritma 1.3. Analisis Algoritma 1.4. Perancangan Algoritma 1.5. Tantangan dalam Penerapan Algoritma 1.6. Efisiensi Algoritma 1.7. Kriteria Algoritma yang Baik 2. Kompleksitas Algoritma 2.1. Kompleksitas Algoritma dari segi waktu dan memori 2.2. Pengukuran Run Time 2.3. Estimasi Run Time 2.4. Growth Rate 2.5. Constant Factor 2.6. Notasi Asimtotik 3. Fungsi Rekursif 3.1. Definisi Fungsi Rekursif 3.2. Sifat Fungsi Rekursif 3.3. Format Fungsi rekursif 3.4. Analisis Efisiensi Algoritma Rekursif 3.5. Worst Case, Average Case, dan Best Case 3.6. Persoalan Rekursif 4. Brute Force 4.1. Definisi Brute Force 4.2. Karakteristik Brute Force 4.3. Studi kasus: Operasi aritmatika 4.4. Studi kasus: Sequential search 4.5. Studi kasus: Bubble sort 5. Greedy Algorithm 5.1. Definisi Greedy Algorithm 5.2. Karakteristik GA 5.3. Elemen Identifikasi GA

	5.4. Skema Umum GA 5.5. Penukaran Uang 5.6. Knapsack Problem 5.7. Antrean 6. Divide and Conquer 6.1. Definisi Divide and Conquer 6.2. Master Theorem 6.3. Max and Min 6.4. Closest Pair 7. Searching and Sorting 7.1. Search Algorithm 7.2. Linear Search Algorithm 7.3. Binary Search Algorithm 7.4. Sort Algorithm 7.5. Merge Sort 7.6. Selection Sort 7.7. Quick Sort 7.8. Heap Sort 8. BFS dan DFS 8.1. Breadth First Search Algorithm 8.2. Pseudo-code BFS 8.3. Depth First Search Algorithm 8.4. Pseudo-code DFS 8.5. Penerapan BFS-DFS 9. Macam-Macam Algoritma 9.1. Shortest Path Problem 9.2. Djikstra Algorithm 9.3. Floyd-Warshall Algorithm 9.4. Minimum Spanning Tree 9.5. Prim's Algorithm 9.6. Kruskal's Algorithm
Pustaka	**Utama** 1. Anany Levitin. 2012. Introduction to The Design & Analysis of Algorithm 3rd Edition. Pearson Education. **Pendukung** 2. Jon Kleinberg, Eva Tardos. 2006. Algorithm Design. Pearson Education. 3. Sandeep Sen, Amit Kumar. 2019. Design and Analysis of Algorithms: A Contemporary Perspective. Cambridge University Press.
Media Pembelajaran	**Software**: PowerPoint, DevC, VSCode, Codeblock **Hardware**: PC & LCD Projector

Teacher/Team Teaching/Tim LS	-
Assessment	Pengetahuan: Tes tulis (UTS, UAS), Psikomotorik: Kinerja (Tugas). Sikap: Observasi harian
Mata Kuliah Syarat	Aljabar Linear

Pertemuan Ke	Kemampuan Akhir yang direncanakan	Indikator Pencapaian Kompetensi	Materi Pokok	Bentuk dan Metode Pembelajaran	Pengalaman Belajar Mahasiswa	Estimasi Waktu	Penilaian			Referensi
							Bentuk & Kriteria	Indikator Penilaian	Bobot (%)	
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
1	Menjelaskan konsep analisis dan perancangan algoritma	1.1. Menjelaskan inti kajian dari mata kuliah Perancangan dan Analisis Algoritma 1.2. Menjelaskan definisi algoritma 1.3. Menjelaskan tentang analisis algoritma 1.4. Menjelaskan tentang perancangan algoritma 1.5. Menjabarkan tantangan dalam penerapan algoritma 1.6. Menjelaskan tentang efisiensi algoritma 1.7. Menyebutkan kriteria algoritma yang baik	1. Konsep Analisis dan Perancangan Algoritma 1.1. Ruang lingkup kajian desain dan perancangan algoritma 1.2. Pengenalan Algoritma 1.3. Analisis Algoritma 1.4. Perancangan Algoritma 1.5. Tantangan dalam Penerapan Algoritma 1.6. Efisiensi Algoritma 1.7. Kriteria Algoritma yang Baik	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, praktikum, tanya jawab, presentasi	Mendiskusikan konsep analisis dan perancangan algoritma	TM: 1x(4x50") BT: 1x(4x60") BM: 1x(4x60")	Tes: Tulis Pedoman Penskoran	Ketepatan menjelaskan konsep analisis dan perancangan algoritma	10	1
2, 3	Menentukan kompleksitas sebuah algoritma menggunakan notasi asimtotik	2.1. Menguraikan kompleksitas algoritma dari segi waktu dan memori 2.2. Menjelaskan cara mengukur running time sebuah program 2.3. Menjelaskan cara menghitung estimasi running time sebuah program 2.4. Menghitung growth rate sebuah algoritma 2.5. Menjelaskan tentang constant factor sebuah algoritma 2.6. Membedakan ketiga notasi asimtotik yang digunakan untuk menyatakan kompleksitas sebuah algoritma	2. Kompleksitas Algoritma 2.1. Kompleksitas Algoritma dari segi waktu dan memori 2.2. Pengukuran Run Time 2.3. Estimasi Run Time 2.4. Growth Rate 2.5. Constant Factor 2.6. Notasi Asimtotik	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, praktikum, tanya jawab, presentasi	Mendiskusikan cara menentukan kompleksitas sebuah algoritma menggunakan notasi asimtotik	TM: 2x(4x50") BT: 2x(4x60") BM: 2x(4x60")	Tes: Tulis Pedoman Penskoran	Ketepatan menentukan kompleksitas sebuah algoritma menggunakan notasi asimtotik	15	1,2
4, 5	Merancang dan menganalisis program yang menggunakan fungsi rekursif	3.1. Menjelaskan definisi fungsi rekursif 3.2. Menguraikan sifat fungsi rekursif 3.3. Menguraikan format fungsi rekursif 3.4. Menganalisis efisiensi algoritma rekursif	3. Fungsi Rekursif 3.1. Definisi Fungsi Rekursif 3.2. Sifat Fungsi Rekursif 3.3. Format Fungsi rekursif	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, praktikum, tanya jawab, presentasi	Mendiskusikan cara menganalisis dan merancang program yang menggunakan fungsi rekursif	TM: 2x(4x50") BT: 2x(4x60") BM: 2x(4x60")	Tes: Tulis Pedoman Penskoran	Ketepatan merancang dan menganalisis program yang menggunakan fungsi rekursif	10	1,2

		3.5. Menguraikan worst case, average case, dan best case dari sebuah algoritma 3.6. Menyebutkan persoalan yang membutuhkan solusi rekursif	3.4. Analisis Efisiensi Algoritma Rekursif 3.5. Worst Case, Average Case, dan Best Case 3.6. Persoalan Rekursif							
6	Menganalisis dan merancang program yang menggunakan algoritma Brute Force	4.1. Menjelaskan definisi algoritma Brute Force 4.2. Menyebutkan karakteristik algoritma Brute Force 4.3. Menganalisis dan merancang program yang menyelesaikan problem operasi aritmatika menggunakan metode Brute Force 4.4. Menganalisis dan merancang program yang menggunakan metode sequential search 4.5. Menganalisis dan merancang program yang menggunakan metode bubble sort	4. Brute Force 4.1. Definisi Brute Force 4.2. Karakteristik Brute Force 4.3. Studi kasus: Operasi aritmatika 4.4. Studi kasus: Sequential search 4.5. Studi kasus: Bubble sort	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, praktikum, tanya jawab, presentasi	Mendiskusikan cara menganalisis dan merancang program yang menggunakan algoritma Brute Force	TM: 1x(4x50") BT: 1x(4x60") BM: x(4x60")	Tes: Tulis Pedoman Penskoran	Ketepatan menganalisis dan merancang program yang menggunakan algoritma Brute Force	10	1,2,3
7	Menganalisis dan merancang program yang menggunakan Greedy Algorithm	5.1. Menjelaskan definisi Greedy Algorithm 5.2. Menguraikan karakteristik Greedy Algorithm 5.3. Menyebutkan elemen identifikasi Greedy Algorithm 5.4. Menjabarkan skema umum Greedy Algorithm 5.5. Menganalisis dan merancang program yang menyelesaikan permasalahan penukaran uang menggunakan GA 5.6. Menganalisis dan merancang program yang menyelesaikan Knapsack Problem menggunakan GA 5.7. Menganalisis dan merancang program yang menyelesaikan permasalahan antrean menggunakan GA	5. Greedy Algorithm 5.1. Definisi Greedy Algorithm 5.2. Karakteristik GA 5.3. Elemen Identifikasi GA 5.4. Skema Umum GA 5.5. Penukaran Uang 5.6. Knapsack Problem 5.7. Antrean	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, praktikum, tanya jawab, presentasi	Mendiskusikan cara menganalisis dan merancang program yang menggunakan Greedy Algorithm	TM: 1x(4x50") BT: 1x(4x60") BM: x(4x60")	Tes: Tulis Pedoman Penskoran	Ketepatan menganalisis dan merancang program yang menggunakan Greedy Algorithm	10	1,2,3
8	UTS									
9	Menganalisis dan merancang program yang menggunakan teori algoritma Divide and Conquer	6.1. Menjelaskan definisi Divide and Conquer 6.2. Menjabarkan tentang Master Theorem	6. Divide and Conquer 6.1. Definisi Divide and Conquer	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, praktikum, tanya jawab, presentasi	Mendiskusikan cara menganalisis dan merancang program yang menggunakan teori algoritma Divide and	TM: 1x(4x50") BT: 1x(4x60") BM: x(4x60")	Tes: Tulis Pedoman Penskoran	Ketepatan menganalisis dan merancang program yang menggunakan	15	1,2,3

	Conquer	6.3. Menganalisis dan merancang program yang menyelesaikan permasalahan Max and Min 6.4. Menganalisis dan merancang program yang menyelesaikan permasalahan Closest Pair	6.2. Master Theorem 6.3. Max and Min 6.4. Closest Pair	tanya jawab, presentasi	algoritma Divide and Conquer			menggunakan teori algoritma Divide and Conquer		
10, 11	Menganalisis dan merancang program yang menggunakan metode searching dan sorting	7.1. Menjelaskan tentang search algorithm 7.2. Menganalisis dan merancang program menggunakan Linear Search Algorithm 7.3. Menganalisis dan merancang program menggunakan Binary Search Algorithm 7.4. Menjelaskan tentang sort algorithm 7.5. Menganalisis dan merancang program menggunakan Merge Sort 7.6. Menganalisis dan merancang program menggunakan Selection Sort 7.7. Menganalisis dan merancang program menggunakan Quick Sort 7.8. Menganalisis dan merancang program menggunakan Heap Sort	7. Searching and Sorting 7.1. Search Algorithm 7.2. Linear Search Algorithm 7.3. Binary Search Algorithm 7.4. Sort Algorithm 7.5. Merge Sort 7.6. Selection Sort 7.7. Quick Sort 7.8. Heap Sort	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, praktikum, tanya jawab, presentasi	Mendiskusikan cara menganalisis dan merancang program yang menggunakan metode searching dan sorting	TM: 2x(4x50") BT: 2x(4x60") BM: 2x(4x60")	Tes: Tulis Pedoman Penskoran	Ketepatan menganalisis dan merancang program yang menggunakan metode searching dan sorting	10	1,2,3
12, 13	Menganalisis dan merancang program yang menggunakan metode BFS dan DFS	8.1. Menjelaskan tentang BFS 8.2. Menganalisis Pseudo-Code dari metode BFS 8.3. Menjelaskan tentang DFS 8.4. Menganalisis Pseudo-Code dari metode DFS 8.5. Merancang program yang menggunakan metode BFS dan DFS	8. BFS dan DFS 8.1. Breadth First Search Algorithm 8.2. Pseudo-code BFS 8.3. Depth First Search Algorithm 8.4. Pseudo-code DFS 8.5. Penerapan BFS-DFS	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, praktikum, tanya jawab, presentasi	Mendiskusikan cara menganalisis dan merancang program yang menggunakan metode BFS dan DFS	TM: 2x(4x50") BT: 2x(4x60") BM: 2x(4x60")	Tes: Tulis Pedoman Penskoran	Ketepatan menganalisis dan merancang program yang menggunakan metode BFS dan DFS	10	1,2
14, 15	Menganalisis dan merancang algoritma yang digunakan untuk menyelesaikan Shortest Path Problem dan menghitung Minimum Spanning Tree	9.1. Menganalisis dan menjelaskan Shortest Path Problem 9.2. Menganalisis dan merancang program yang menggunakan Djikstra Algorithm	9. Macam-Macam Algoritma 9.1. Shortest Path Problem	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, praktikum, tanya jawab, presentasi	Mendiskusikan cara menganalisis dan merancang algoritma yang digunakan untuk menyelesaikan Shortest Path Problem dan menghitung Minimum	TM: 2x(4x50") BT: 2x(4x60") BM: 2x(4x60")	Tes: Tulis Pedoman Penskoran	Ketepatan menganalisis dan merancang algoritma yang digunakan untuk menyelesaikan	10	1,2,3

	Spanning Tree	9.3. Menganalisis dan merancang program yang menggunakan Floyd-Warshall Algorithm 9.4. Menganalisis dan menjelaskan Minimum Spanning Tree 9.5. Menganalisis dan merancang program yang menggunakan Prim's Algorithm 9.6. Menganalisis dan merancang program yang menggunakan Kruskal's Algorithm	9.2. Djikstra Algorithm 9.3. Floyd-Warshall Algorithm 9.4. Minimum Spanning Tree 9.5. Prim's Algorithm 9.6. Kruskal's Algorithm		menghitung Minimum Spanning Tree			menyelesaikan Shortest Path Problem dan menghitung Minimum Spanning Tree		
16	UAS									

Catatan :

1. Capaian Pembelajaran Lulusan PRODI (CPL-PRODI) adalah kemampuan yang dimiliki oleh setiap lulusan PRODI yang merupakan internalisasi dari sikap, penguasaan pengetahuan dan ketrampilan sesuai dengan jenjang prodinya yang diperoleh melalui proses pembelajaran.

2. CPL yang dibebankan pada mata kuliah adalah beberapa capaian pembelajaran lulusan program studi (CPL-PRODI) yang digunakan untuk pembentukan/pengembangan sebuah mata kuliah yang terdiri dari aspek sikap, ketrampulan umum, ketrampilan khusus dan pengetahuan.

3. CP Mata kuliah (CPMK) adalah kemampuan yang dijabarkan secara spesifik dari CPL yang dibebankan pada mata kuliah, dan bersifat spesifik terhadap bahan kajian atau materi pembelajaran mata kuliah tersebut.

4. Sub-CP Mata kuliah (Sub-CPMK) adalah kemampuan yang dijabarkan secara spesifik dari CPMK yang dapat diukur atau diamati dan merupakan kemampuan akhir yang direncanakan pada tiap tahap pembelajaran, dan bersifat spesifik terhadap materi pembelajaran mata kuliah tersebut.

5. Kreteria Penilaian adalah patokan yang digunakan sebagai ukuran atau tolok ukur ketercapaian pembelajaran dalam penilaian berdasarkan indikator-indikator yang telah ditetapkan. Kreteria penilaian merupakan pedoman bagi penilai agar penilaian konsisten dan tidak bias. Kreteria dapat berupa kuantitatif ataupun kualitatif.

6. Indikator penilaian kemampuan dalam proses maupun hasil belajar mahasiswa adalah pernyataan spesifik dan terukur yang mengidentifikasi kemampuan atau kinerja hasil belajar mahasiswa yang disertai bukti-bukti.

Catatan tambahan:

(1). Bobot SKS (P = Praktek; T= Teori).

(2). TM: Tatap Muka; BT: Beban Tugas; BM: Belajar Mandiri.

(3). 1 sks = (50’ TM + 50’ PT + 60’ BM)/Minggu

(4). Simbol-simbol elemen KKNI pada CPL-Prodi: S = Sikap; KU = Ketrampilan Umum; KK = Ketrampilan Khusus; P = Pengetahuan