

Rencana Pembelajaran Semester (RPS)

UNIVERSITAS PENDIDIKAN GANESHA JURUSAN TEKNIK INFORMATIKA PROGRAM STUDI ILMU KOMPUTER					Kode Dokumen	
RENCANA PEMBELAJARAN SEMESTER						
MATA KULIAH (MK)	KODE	Rumpun MK	BOBOT (sks)		SEMESTER	Tgl Penyusunan
Desain dan Analisis Algoritma	KOMS120403	Mata kuliah inti keilmuan	T=3	P=0	4	04/05/2022
OTORISASI	Pengembang RPS		Koordinator RMK		Ketua PRODI	
	Ni Luh Dewi Sintari, Ph.D.		A.A. Gede Yudhi Paramartha, S.Kom., M.Kom.		A.A. Gede Yudhi Paramartha, S.Kom., M.Kom.	
Capaian Pembelajaran (CP)	CPL-PRODI yang dibebankan pada MK					
	S1	Bertakwa kepada Tuhan Yang Maha Esa dan mampu menunjukkan sikap religius;				
	S2	Menjunjung tinggi nilai kemanusiaan dalam menjalankan tugas berdasarkan agama, moral, dan etika;				
	S8	Menginternalisasi nilai, norma dan etika akademik;				
	S9	Menunjukkan sikap bertanggung jawab atas pekerjaan di bidang keahliannya secara mandiri				
	S10	Menginternalisasi semangat kemandirian, kejuangan, dan kewirausahaan;				
	P1	Mampu memahami dan menguasai konsep dasar ilmu komputer secara umum seperti matematika, algoritma, pemrograman, dan basis data.				
	P2	Mampu memahami dan menguasai konsep pengembangan perangkat lunak, mulai dari analisis kebutuhan, perancangan, pengembangan, dan implementasi perangkat lunak.				
	KU1	Mampu menerapkan pemikiran logis, kritis, sistematis, dan inovatif dalam konteks pengembangan atau implementasi ilmu pengetahuan dan teknologi yang memperhatikan dan menerapkan nilai humaniora yang sesuai dengan bidang ilmu komputer;				
	KU2	Mampu menunjukkan kinerja mandiri, bermutu, dan terukur;				
KK1	Terampil dalam menganalisis kebutuhan, merancang, dan mengimplementasikan rancangan, dan menguji perangkat lunak.					
Capaian Pembelajaran Mata Kuliah (CPMK)						

	CPMK	Mahasiswa mampu merumuskan desain algoritma untuk menyelesaikan masalah dalam Ilmu Komputer, dan memiliki keterampilan untuk mengimplementasikan algoritma tersebut ke dalam bahasa pemrograman, sehingga mampu menjelaskan metode penyelesaian masalah secara sistematis dalam bentuk verbal dan tulisan.
	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	
	Sub-CPMK1	Mahasiswa mampu menjelaskan tahapan desain dan analisis algoritma dengan baik
	Sub-CPMK2	Mahasiswa mampu menghitung kompleksitas waktu algoritma (worst-case, best-case, average-case), menggunakan notasi Big-O, Big-Omega, dan Big-Theta, dan mengklasifikasikan algoritma berdasarkan kompleksitas waktunya dengan benar
	Sub-CPMK3	Mahasiswa mampu menjelaskan tentang konsep strategi brute-force/exhaustive search dan teknik heuristik dengan baik, menganalisis kebenaran dan kompleksitas waktu algoritma brute-force, serta mengaplikasikan strategi tersebut dalam pemecahan masalah dengan baik dan benar
	Sub-CPMK4	Mahasiswa mampu menjelaskan konsep algoritma rekursif, menuliskan pseudocode, menganalisis kebenaran, memformulasikan bentuk rekursif dari fungsi kompleksitas waktunya dan menghitung rumus eksplisit fungsi tersebut, , serta mengaplikasikan metode rekursif dalam pemecahan masalah dan mengimplementasikannya dalam program komputer dengan baik dan benar
	Sub-CPMK5	Mahasiswa mampu menjelaskan strategi Divide-and-Conquer, Decrease-and-Conquer, dan Transform-and-Conquer, menuliskan pseudocode, menganalisis kebenaran dan menghitung fungsi kompleksitas waktu algoritma, serta mengaplikasikan ketiga strategi tersebut dalam pemecahan masalah dengan baik dan benar
	Sub-CPMK6	Mahasiswa mampu menjelaskan konsep algoritma Greedy, membuktikan optimalitas atau menunjukkan ketidak-optimalan algoritma Greedy, mengaplikasikan metode Greedy dalam pemecahan masalah dan mengimplementasikannya dalam program komputer dengan baik dan benar
	Sub-CPMK7	Mahasiswa mampu menjelaskan metode BFS dan DFS dengan baik, menganalisis kompleksitas waktu dan ruang melalui contoh riil, dan mengaplikasikan metode BFS dan DFS dalam pembentukan pohon ruang status pada algoritma graf dinamis dengan baik dan benar
	Sub-CPMK8	Mahasiswa mampu menjelaskan konsep algoritma Backtracking dan Branch-and-Bound, serta mengaplikasikannya dalam pemecahan masalah algoritmik dengan baik dan benar
	Sub-CPMK9	Mahasiswa mampu menjelaskan konsep pemrograman dinamis, melakukan analisis kompleksitas waktu, dan mengaplikasikan pemrograman dinamis dalam pemecahan masalah algoritmik dengan baik dan benar
	Sub-CPMK10	Mahasiswa mampu menjelaskan jenis-jenis permasalahan algoritmik dalam Ilmu Komputer, mengklasifikasikan masalah dalam kelas kompleksitas (P, NP, NP-Complete, dan NP-Hard), serta menentukan strategi algoritma yang tepat dalam pemecahan masalah algoritmik dengan baik dan benar
Deskripsi Singkat MK	Mata kuliah ini mempelajari tentang perancangan dan analisis algoritma, yang mencakup pembahasan mengenai jenis-jenis permasalahan algoritmik pada dunia komputer, analisis efisiensi yaitu kompleksitas waktu dan ruang algoritma, strategi-strategi perancangan algoritma, dan keterbatasan	

	<p>setiap strategi algoritma. Strategi-strategi perancangan algoritma yang dibahas mencakup strategi Brute Force, teknik Rekursif, Divide-and-Conquer, Decrease-and-Conquer, Transform-and-Conquer, Greedy, Backtracking, Branch and Bound, Dynamic Programming, serta kelas kompleksitas algoritma (Teori P, NP, dan NP-Complete). Setelah mengikuti mata kuliah ini, mahasiswa diharapkan memahami berbagai macam strategi perancangan algoritma, serta mampu mengaplikasikan teknik perancangan algoritma untuk menyelesaikan masalah dalam kehidupan nyata.</p>						
Bahan Kajian: Materi Pembelajaran	Bahan Kajian: Pengenalan macam-macam strategi algoritma, metode pembuktian kebenaran algoritma dan analisis kompleksitas waktu algoritma.			Materi Pembelajaran: Modul ajar			
Pustaka	Utama:	Introduction to the Design & Analysis of Algorithms, Anany Levitin, Pearson Education, Inc.					
	Pendukung:	<ul style="list-style-type: none"> - Slide Kuliah Strategi Algoritma, oleh Rinaldi Munir, Institut Teknologi Bandung - Slide Analysis of Algorithms, Robert Sedgewick 					
Dosen Pengampu	Ni Luh Dewi Sintari, Ph.D.						
Matakuliah syarat	Struktur Data dan Algoritma						
Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Penilaian		Bantuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa, [Estimasi Waktu]		Materi Pembelajaran [Pustaka]	Bobot Penilaian (%)
		Indikator	Kriteria & Bentuk	Luring (<i>offline</i>)	Daring (<i>online</i>)		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

1	Mahasiswa mampu menjelaskan tahapan desain dan analisis algoritma dengan baik	<ol style="list-style-type: none"> 1. Ketepatan dalam menuliskan algoritma sederhana dengan benar 2. Ketepatan dalam menjelaskan tahapan proses perancangan algoritma 3. Ketepatan dalam menjelaskan tahapan analisis algoritma 4. Ketepatan dalam menjelaskan permasalahan-permasalahan algoritmik penting dalam Ilmu Komputer 5. Ketepatan dalam menyebutkan macam-macam strategi perancangan algoritma 6. Ketepatan dalam menjelaskan definisi kebenaran algoritma beserta teknik untuk membuktikan kebenaran suatu algoritma 	<p>Bentuk Penilaian:</p> <ul style="list-style-type: none"> • Non-tes, tanya-jawab lisan 	<p><u>Bentuk Pembelajaran:</u> Kegiatan Proses Belajar [3x50']</p> <p><u>Metode Pembelajaran:</u> Diskusi, tanya-jawab, penugasan</p> <p><u>Tugas 1:</u> Penulisan makalah ilmiah aplikasi strategi algoritma (Waktu pengerjaan 12 minggu)</p>	<p><u>Media:</u> elearning.undiksha.ac.id</p>	<ul style="list-style-type: none"> • Kontrak kuliah • Pengantar desain dan analisis algoritma • Jenis-jenis algoritma • Contoh permasalahan dalam Ilmu Komputer 	5%
---	---	--	---	--	---	---	----

2	<p>Mahasiswa mampu menghitung kompleksitas waktu algoritma (worst-case, best-case, average-case), menggunakan notasi Big-O, Big-Omega, dan Big-Theta, dan mengklasifikasikan algoritma berdasarkan kompleksitas waktunya dengan benar</p>	<ol style="list-style-type: none"> 1. Ketepatan dalam menjelaskan konsep dan urgensi dari penghitungan kompleksitas waktu algoritma 2. Ketepatan dalam menghitung fungsi kompleksitas waktu algoritma. 3. Ketepatan dalam menghitung kompleksitas waktu worst-case, best-case, dan average-case suatu algoritma 4. Ketepatan dalam menuliskan kompleksitas waktu dengan notasi asimptotik (Big-O, Big-Sigma, Big-Omega) 5. Ketepatan dalam menghitung operasi aritmetik dengan notasi asimptotis 6. Ketepatan dalam membuktikan secara matematis sifat-sifat sederhana dari operasi aritmetik dengan notasi asimptotis 7. Ketepatan dalam 	<p>Bentuk Penilaian:</p> <ul style="list-style-type: none"> • Tanya-jawab lisan • Quiz • Tugas 	<p><u>Bentuk Pembelajaran:</u> Kegiatan Proses Belajar [3x50']</p> <p><u>Metode Pembelajaran:</u> Diskusi, tanya-jawab, presentasi, penugasan</p> <p><u>Tugas 2:</u> Penghitungan kompleksitas waktu algoritma dan pembuktian sifat-sifat dengan notasi asimptotik</p>	<p><u>Media:</u> elearning.undiksha.ac.id</p>	<ol style="list-style-type: none"> 1. Penghitungan kompleksitas waktu worst-case, best-case, dan average-case algoritma 2. Notasi asimptotik (big-O, big-Theta, dan big-Omega) beserta operasinya 3. Kelas algoritma berdasarkan fungsi kompleksitas waktunya 	7%
---	---	--	---	--	---	--	----

		mengelompokkan algoritma menjadi kelas algoritma berdasarkan kompleksitas waktunya (linier, polinomial, logaritmik, eksponensial, dsb.)					
3	Mahasiswa mampu menjelaskan tentang konsep strategi brute-force/exhaustive search dan teknik heuristik dengan baik, menganalisis kebenaran dan kompleksitas waktu algoritma brute-force, serta mengaplikasikan strategi tersebut dalam pemecahan masalah dengan baik dan benar	<ol style="list-style-type: none"> 1. Ketepatan dalam menjelaskan prinsip dasar dan karakteristik algoritma brute-force 2. Ketepatan dalam merancang algoritma brute-force untuk menyelesaikan permasalahan algoritmik sederhana, seperti: mencari nilai maksimum/minimum pada array, sequential search, menghitung 	<p>Bentuk Penilaian:</p> <ul style="list-style-type: none"> • Tanya-jawab lisan • Presentasi materi • Tugas 	<p><u>Bentuk Pembelajaran:</u> Kegiatan Proses Belajar [3x50']</p> <p><u>Metode Pembelajaran:</u> Diskusi, tanya-jawab, presentasi, penugasan</p> <p><u>Tugas 3:</u> Mendesain algoritma brute-</p>	<p><u>Media:</u> elearning.undiksha.ac.id</p>	<ol style="list-style-type: none"> 1. Pengenalan strategi brute-force 2. Pembuktian kebenaran dan penghitungan kompleksitas waktu algoritma brute-force 3. Strategi exhaustive search untuk permasalahan kombinatorial 	5%

		<p>perpangkatan bilangan, menghitung nilai faktorial, perkalian matriks persegi, pengecekan bilangan prima, interpolasi polinom, mencari pasangan titik terdekat, dan pattern matching, dll.</p> <p>3. Ketepatan dalam menghitung kompleksitas waktu algoritma Brute-force</p> <p>4. Ketepatan dalam memodifikasi algoritma Brute-force untuk meningkatkan efisiensinya</p> <p>5. Ketepatan dalam merancang algoritma exhaustive search untuk menyelesaikan permasalahan kombinatorial, seperti: Traveling Salesman Problem dan 1/0 Knapsack problem</p>		<p>force untuk menyelesaikan permasalahan algoritmik sederhana, dan menganalisis kebenaran dan kompleksitas waktunya</p>			
4		<p>1. Ketepatan dalam menjelaskan implementasi teknik heuristik pada algoritma</p>	<p>Bentuk Penilaian:</p> <ul style="list-style-type: none"> • Tanya-jawab lisan • Presentasi 	<p><u>Bentuk Pembelajaran:</u> Kegiatan Proses Belajar [3x50']</p>	<p><u>Media:</u> elearning.undiksha.ac.id</p>	<p>1. Teknik heuristik untuk peningkatan efisiensi algoritma brute-force</p>	4%

		<p>untuk meningkatkan efisiensi strategi Brute-force, seperti pada permasalahan anagram dan magic square</p> <ol style="list-style-type: none"> 2. Ketepatan dalam menyebutkan kelebihan dan kekurangan dari metode brute force 3. Ketepatan dalam menjelaskan tahapan algoritma Selection sort, Bubble sort, dan Insertion sort 4. Ketepatan dalam membuktikan kebenaran algoritma Selection sort, Bubble sort, dan Insertion sort dengan menggunakan loop-invariant 5. Ketepatan dalam mengaplikasikan algoritma Selection Sort, Bubble Sort, dan Insertion sort dalam pemecahan masalah 6. Ketepatan dalam membuat program implementasi algoritma sorting pada pemecahan masalah 	<p>materi</p> <ul style="list-style-type: none"> • Tugas 	<p><u>Metode Pembelajaran:</u> Diskusi, tanya-jawab, presentasi, penugasan</p> <p><u>Tugas 4:</u> Implementasi algoritma sorting pada Bahasa pemrograman</p>		<ol style="list-style-type: none"> 2. Algoritma sorting berbasis brute-force (Selection Sort, Bubble Sort, dan Insertion Sort) 3. Pembuktian dengan metode loop-invariant 4. Program komputer untuk implementasi algoritma brute-force 	
--	--	--	---	--	--	---	--

5	<p>Mahasiswa mampu menjelaskan konsep algoritma rekursif, menuliskan pseudocode, menganalisis kebenaran, memformulasikan bentuk rekursif dari fungsi kompleksitas waktunya dan menghitung rumus eksplisit fungsi tersebut, serta mengaplikasikan metode rekursif dalam pemecahan masalah dan mengimplementasikannya dalam program komputer dengan baik dan benar</p>	<ol style="list-style-type: none"> 1. Ketepatan dalam menjelaskan prinsip strategi rekursif, karakteristik algoritma rekursif, serta perbedaannya dengan algoritma brute-force 2. Ketepatan dalam merancang algoritma rekursif untuk menyelesaikan permasalahan algoritmik sederhana, seperti menghitung faktorial, mencari nilai maksimum pada array, dan menghitung jumlah elemen pada array 3. Ketepatan dalam menjelaskan tahapan algoritma rekursif untuk menyelesaikan masalah Menara Hanoi 4. Ketepatan dalam menjelaskan tahapan algoritma rekursif pada Binary Search 5. Ketepatan dalam menjelaskan tahapan algoritma rekursif untuk menghitung perpangkatan 6. Ketepatan dalam 	<p>Bentuk Penilaian:</p> <ul style="list-style-type: none"> • Tanya-jawab lisan • Presentasi materi • Tugas 	<p><u>Bentuk Pembelajaran:</u> Kegiatan Proses Belajar [3x50']</p> <p><u>Metode Pembelajaran:</u> Diskusi, tanya-jawab, presentasi, penugasan</p> <p><u>Tugas 5:</u> Penyelesaian masalah algoritmik sederhana dengan metode rekursif, dan implementasinya dalam Bahasa pemrograman</p>	<p><u>Media:</u> elearning.undiksha.ac.id</p>	<ol style="list-style-type: none"> 1. Pengenalan strategi rekursif 2. Penyelesaian masalah Menara Hanoi 3. Binary search 4. Pembuktian kebenaran algoritma rekursif dengan induksi matematika 5. Analisis efisiensi waktu algoritma rekursif 6. Implementasi algoritma rekursif dalam bentuk program komputer 	<p>8%</p>
---	--	---	--	---	---	---	------------------

		<p>membuktikan kebenaran dari algoritma rekursif dengan menggunakan induksi matematika</p> <p>7. Ketepatan dalam menyatakan fungsi kompleksitas waktu dalam formula rekursif dan menghitung bentuk fungsi eksplisitnya Ketepatan dalam menjelaskan redundansi algoritma rekursif pada algoritma konstruksi barisan Fibonacci</p> <p>8. Ketepatan dalam menuliskan pseudocode algoritma rekursif untuk mengatasi tumpang tindih (<i>overlap</i>) komputasi</p> <p>9. Ketepatan dalam membuat program/implementasi algoritma rekursif dan melakukan analisis efisiensi berdasarkan eksperimen</p>					
--	--	---	--	--	--	--	--

6	<p>Mahasiswa mampu menjelaskan strategi Divide-and-Conquer, Decrease-and-Conquer, dan Transform-and-Conquer, menuliskan pseudocode, menganalisis kebenaran dan menghitung fungsi kompleksitas waktu algoritma, serta mengaplikasikan ketiga strategi tersebut dalam pemecahan masalah dengan baik dan benar</p>	<ol style="list-style-type: none"> 1. Ketepatan dalam menjelaskan prinsip algoritma divide-and-conquer, dan mengaplikasikannya untuk menyelesaikan permasalahan algoritmik 2. Ketepatan dalam menghitung kompleksitas waktu algoritma divide-and-conquer 3. Ketepatan dalam menjelaskan algoritma pengurutan data berbasis divide-and-conquer, yaitu: merge sort dan quick sort 4. Ketepatan dalam menggunakan Teorema Master untuk menghitung kompleksitas waktu algoritma divide-and-conquer 5. Ketepatan dalam menjelaskan algoritma Divide-and-Conquer untuk perkalian matriks persegi, serta modifikasi algoritma untuk meningkatkan efisiensi, 	<p>Bentuk Penilaian:</p> <ul style="list-style-type: none"> • Tanya-jawab lisan • Presentasi materi • Tugas teori 	<p><u>Bentuk Pembelajaran:</u></p> <ul style="list-style-type: none"> - Kegiatan Proses Belajar [3x50'] - Tugas mandiri [3x50'] <p><u>Metode Pembelajaran:</u></p> <p>Diskusi, tanya-jawab, presentasi, penugasan</p> <p><u>Tugas 6:</u></p> <p>Implementasi algoritma divide-conquer pada permasalahan perkalian dua polinomial</p>	<p><u>Media:</u></p> <p>elearning.undiksha.ac.id</p>	<ol style="list-style-type: none"> 1. Pengenalan strategi divide-and-conquer 2. Analisis kompleksitas waktu algoritma divide-and-conquer 3. Merge Sort dan Quick Sort 4. Teorema Master 5. Metode Strassen untuk perkalian matriks 6. Metode Karatsuba untuk perkalian bilangan besar 	7%
---	---	---	--	--	--	---	----

		<p>melalui algoritma perkalian matriks Strassen</p> <p>6. Ketepatan dalam menjelaskan algoritma Divide-and-Conquer untuk perkalian bilangan besar, serta modifikasi algoritma untuk meningkatkan efisiensi, melalui perkalian Karatsuba</p>					
7		<p>1. Ketepatan dalam menjelaskan prinsip strategi Decrease-and-Conquer, serta perbedaannya dengan strategi Divide-and-Conquer</p> <p>2. Ketepatan dalam menjelaskan pendekatan 'decrease by a constant', 'decrease by constant factor', dan 'decrease by variable size', serta mengaplikasikannya</p>	<p>Bentuk Penilaian:</p> <ul style="list-style-type: none"> • Tanya-jawab lisan • Presentasi materi • Quiz • Tugas 	<p><u>Bentuk Pembelajaran:</u> Kegiatan Proses Belajar [3x50']</p> <p><u>Metode Pembelajaran:</u> Diskusi, tanya-jawab, presentasi, penugasan</p> <p><u>Tugas 7:</u> Implementasi algoritma decrease-conquer</p>	<p><u>Media:</u> elearning.undiksha.ac.id</p>	<p>1. Pengenalan strategi decrease-and-conquer serta jenis-jenis pendekatannya</p> <p>2. Pengenalan strategi transform-and-conquer serta jenis-jenis pendekatannya</p>	6%

		<p>dalam pemecahan masalah.</p> <p>3. Ketepatan dalam menjelaskan prinsip Transform-and-Conquer dan perbedaannya dengan Divide-and-Conquer dan Decrease-and-Conquer</p> <p>4. Ketepatan dalam menjelaskan pendekatan 'instance representation change', dan 'problem reduction', serta mengaplikasikannya dalam pemecahan masalah</p>		<p>dan transform-conquer pada permasalahan algoritmik</p>			
8	Evaluasi Tengah Semester / Ujian Tengah Semester						10%
9	<p>Mahasiswa mampu menjelaskan konsep algoritma Greedy, membuktikan optimalitas atau menunjukkan ketak-optimalan algoritma Greedy, mengaplikasikan metode Greedy dalam pemecahan masalah dan mengimplementasikannya dalam program komputer dengan baik dan benar</p>	<p>1. Ketepatan dalam menjelaskan prinsip dasar algoritma Greedy</p> <p>2. Ketepatan dalam menjelaskan dan mengidentifikasi komponen algoritma Greedy melalui contoh</p> <p>3. Ketepatan dalam menuliskan pseudocode skema algoritma Greedy</p> <p>4. Ketepatan dalam mengaplikasikan strategi Greedy untuk</p>	<p>Bentuk Penilaian:</p> <ul style="list-style-type: none"> • Tanya-jawab lisan • Presentasi materi • Tugas 	<p><u>Bentuk Pembelajaran:</u> Kegiatan Proses Belajar [3x50']</p> <p><u>Metode Pembelajaran:</u> Diskusi, tanya-jawab, presentasi, penugasan</p> <p><u>Tugas 8:</u> Implementasi algoritma Greedy</p>	<p><u>Media:</u> elearning.undiksha.ac.id</p>	<p>1. Pengenalan strategi Greedy</p> <p>2. Implementasi strategi Greedy dalam pemecahan masalah</p> <p>3. Analisis optimalitas algoritma Greedy</p>	5%

		<p>menyelesaikan permasalahan optimasi sederhana, seperti: <i>Coin exchange problem, Activity selection problem, Time minimization in the system.</i></p> <p>5. Ketepatan dalam membuktikan keoptimalan atau menunjukkan ketak-optimalan algoritma Greedy yang dirancang secara formal</p>		<p>untuk menyelesaikan masalah optimisasi sederhana</p>			
10		<p>1. Ketepatan dalam menjelaskan strategi Greedy untuk penyelesaian masalah 1/0 Knapsack, dengan pendekatan <i>Greedy by profit, Greedy by weight, dan Greedy by density.</i></p> <p>2. Ketepatan dalam menjelaskan strategi Greedy untuk Fractional Knapsack Problem, dengan pendekatan <i>Greedy by profit, Greedy by weight, dan Greedy by density.</i></p> <p>3. Ketepatan dalam</p>	<p>Bentuk Penilaian:</p> <ul style="list-style-type: none"> • Tanya-jawab lisan • Presentasi materi • Tugas tertulis 	<p><u>Bentuk Pembelajaran:</u> Kegiatan Proses Belajar [3x50']</p> <p><u>Metode Pembelajaran:</u> Diskusi, tanya-jawab, presentasi, penugasan</p> <p><u>Tugas 9:</u> Membuktikan keoptimalan/ketak-optimalan algoritma Greedy dalam pemecahan masalah</p>	<p><u>Media:</u> elearning.undiksha.ac.id</p>	<p>1. Penyelesaian masalah 1/0 Knapsack dengan strategi Greedy</p> <p>2. Penyelesaian masalah Fractional Knapsack dengan strategi Greedy</p> <p>3. Konstruksi kode Huffman dengan strategi Greedy</p> <p>4. Analisis optimalitas dari algoritma yang dirancang untuk ketiga permasalahan tersebut</p>	5%

		<p>membuktikan bahwa ketiga pendekatan Greedy pada masalah 1/0 Knapsack tidak selalu memberikan solusi optimal.</p> <p>4. Ketepatan dalam menjelaskan tahapan konstruksi kode Huffman dengan strategi Greedy</p> <p>5. Ketepatan dalam menerapkan algoritma Huffman untuk masalah encoding sederhana</p>					
11		<p>1. Ketepatan dalam menjelaskan tahapan penyelesaian masalah Minimum Spanning Tree dengan algoritma Prim</p> <p>2. Ketepatan dalam menjelaskan tahapan penyelesaian masalah Minimum Spanning Tree dengan algoritma Kruskal</p> <p>3. Ketepatan dalam menjelaskan mengapa algoritma Prim dan Kruskal memberikan hasil optimal untuk masalah Minimum</p>	<p>Bentuk Penilaian:</p> <ul style="list-style-type: none"> • Tanya-jawab lisan • Presentasi materi • Tugas membuat video 	<p><u>Bentuk Pembelajaran:</u> Kegiatan Proses Belajar [3x50']</p> <p><u>Metode Pembelajaran:</u> Diskusi, tanya-jawab, presentasi, penugasan</p> <p><u>Tugas 10:</u> Membuat video penjelasan contoh algoritma untuk MST dan Shortest Path</p>	<p><u>Media:</u> elearning.undiksha.ac.id</p>	<p>1. Algoritma Prim untuk Minimum Spanning Tree</p> <p>2. Algoritma Kruskal untuk Minimum Spanning Tree</p> <p>3. Algoritma Dijkstra untuk Shortest Path</p>	5%

		<p>Spanning Tree</p> <p>4. Ketepatan dalam menjelaskan tahapan algoritma Dijkstra untuk menyelesaikan masalah Shortest Path</p> <p>5. Ketepatan dalam menjelaskan mengapa algoritma Dijkstra memberikan hasil optimal untuk masalah Shortest Path</p> <p>6. Ketepatan dalam mengaplikasikan algoritma Prim, Kruskal, dan Dijkstra untuk menyelesaikan masalah terkait pada graf sederhana</p>					
--	--	---	--	--	--	--	--

12	Mahasiswa mampu menjelaskan metode BFS dan DFS dengan baik, menganalisis kompleksitas waktu dan ruang melalui contoh riil, dan mengaplikasikan metode BFS dan DFS dalam pembentukan pohon ruang status pada algoritma graf dinamis dengan baik dan benar	<ol style="list-style-type: none"> 1. Ketepatan dalam menjelaskan tahapan algoritma BFS pada struktur <i>tree</i> dan struktur graf yang bukan <i>tree</i> 2. Ketepatan dalam menjelaskan tahapan algoritma DFS pada struktur <i>tree</i> dan struktur graf yang bukan <i>tree</i> 3. Ketepatan dalam menjelaskan definisi dan komponen pohon ruang status pada graf dinamis 4. Ketepatan dalam menjelaskan prosedur pembangunan pohon ruang status dengan strategi BFS dan DFS 5. Ketepatan dalam menyelesaikan masalah <i>8-puzzle game</i> melalui pembangunan pohon ruang status. 6. Ketepatan dalam menjelaskan perbandingan efisiensi dari strategi BFS dan DFS 	<p>Bentuk Penilaian:</p> <ul style="list-style-type: none"> • Tanya-jawab lisan • Presentasi materi • Tugas 	<p><u>Bentuk Pembelajaran:</u> Kegiatan Proses Belajar [3x50']</p> <p><u>Metode Pembelajaran:</u> Diskusi, tanya-jawab, presentasi, penugasan</p> <p><u>Tugas 11:</u> Pembangunan pohon ruang status untuk menyelesaikan masalah algoritmik dengan struktur graf dinamis</p>	<p><u>Media:</u> elearning.undiksha.ac.id</p>	<ol style="list-style-type: none"> 1. Strategi BFS dan DFS, serta analisis efisiensinya 2. Pembangunan pohon ruang status pada graf dinamis dengan metode BFS dan DFS 3. Penyelesaian permainan 8-puzzle 	5%
13	Mahasiswa mampu	1. Ketepatan dalam	Bentuk Penilaian:	<u>Bentuk</u>	<u>Media:</u>	1. Pengenalan strategi	6%

	<p>menjelaskan konsep algoritma Backtracking dan Branch-and-Bound, serta mengaplikasikannya dalam pemecahan masalah algoritmik dengan baik dan benar</p>	<p>menjelaskan prinsip dasar strategi backtracking serta keterkaitannya dengan strategi DFS</p> <ol style="list-style-type: none"> 2. Ketepatan dalam menjelaskan langkah-langkah penyelesaian masalah penempatan n-ratu pada papan catur dengan algoritma backtracking 3. Ketepatan dalam menjelaskan Langkah-langkah penyelesaian masalah sirkuit Hamilton dengan algoritma backtracking 4. Ketepatan dalam menjelaskan prinsip dasar strategi branch-and-bound dan perbedaannya dengan strategi backtracking 5. Ketepatan dalam menjelaskan tahapan penyelesaian masalah 1/0 Knapsack dengan algoritma branch-and-bound 	<ul style="list-style-type: none"> • Tanya-jawab lisan • Presentasi materi • Tugas 	<p><u>Pembelajaran:</u> Kegiatan Proses Belajar [3x50']</p> <p><u>Metode</u> <u>Pembelajaran:</u> Diskusi, tanya-jawab, presentasi, penugasan</p> <p><u>Tugas 12:</u> Membuat program computer secara berkelompok dengan materi algoritma backtracking dan branch-and-bound untuk penyelesaian masalah optimisasi</p>	<p>elearning.undiksha.ac.id</p>	<p>backtracking</p> <ol style="list-style-type: none"> 2. Penyelesaian masalah n-ratu dengan strategi backtracking 3. Pencarian sirkuit Hamilton dengan strategi backtracking 4. Pengenalan strategi branch-and-bound 5. Penyelesaian masalah 1/0 Knapsack dengan strategi branch-and-bound 	
--	--	---	---	---	---------------------------------	--	--

14	Mahasiswa mampu menjelaskan konsep pemrograman dinamis, melakukan analisis kompleksitas waktu, dan mengaplikasikan pemrograman dinamis dalam pemecahan masalah algoritmik dengan baik dan benar	<ol style="list-style-type: none"> 1. Ketepatan dalam menjelaskan prinsip dasar strategi pemrograman dinamis (<i>dynamic programming</i>) 2. Ketepatan dalam menjelaskan “Prinsip Optimalitas” pada pemrograman dinamis 3. Ketepatan dalam menjelaskan tahapan pemrograman dinamis untuk menyelesaikan beberapa masalah sederhana, seperti: “Coin-row problem, Change-making problem, dan Coin collecting problem” 4. Ketepatan dalam menjelaskan tahapan pemrograman dinamis untuk menyelesaikan masalah Knapsack 5. Ketepatan dalam menjelaskan konsep “Memory functions” untuk meningkatkan efisiensi pemrograman dinamis pada Knapsack problem, dan masalah algoritmik lainnya 	<p>Bentuk Penilaian:</p> <ul style="list-style-type: none"> • Tanya-jawab lisan • Presentasi materi • Tugas 	<p><u>Bentuk Pembelajaran:</u> Kegiatan Proses Belajar [3x50’]</p> <p><u>Metode Pembelajaran:</u> Diskusi, tanya-jawab, presentasi, penugasan</p> <p><u>Tugas 13:</u> Implementasi pemrograman dinamis pada masalah algoritmik sederhana</p>	<p><u>Media:</u> elearning.undiksha.ac.id</p>	<ol style="list-style-type: none"> 1. Pengenalan konsep dan tahapan pemrograman dinamis 2. Penggunaan program dinamis untuk pemecahan masalah algoritmik sederhana. 3. Penyelesaian 1/0 Knapsack dengan pemrograman dinamis, dan konsep “Memory functions” untuk peningkatan efisiensi 	7%
----	---	---	--	--	---	---	----

		secara umum					
15	Mahasiswa mampu menjelaskan jenis-jenis permasalahan algoritmik dalam Ilmu Komputer, mengklasifikasikan masalah dalam kelas kompleksitas (P, NP, NP-Complete, dan NP-Hard), serta menentukan strategi algoritma yang tepat dalam pemecahan masalah algoritmik dengan baik dan benar	<ol style="list-style-type: none"> 1. Ketepatan dalam menjelaskan perbedaan algoritma deterministik dan non-deterministik 2. Ketepatan dalam menjelaskan perbedaan <i>decision</i> problem dan <i>searching</i> problems, <i>decidable</i> dan <i>undecidable</i> problems, serta <i>tractable</i> dan <i>intractable</i> problems 3. Ketepatan dalam menjelaskan perbedaan kelas P, NP, NP-complete, dan NP-Hard 4. Ketepatan dalam memberikan sebuah contoh permasalahan yang diklasifikasikan sebagai P, NP, NP-complete, atau NP-Hard dan menjelaskan mengapa masalah tersebut diklasifikasikan ke dalam kelas terkait 5. Ketepatan dalam 	<p>Bentuk Penilaian:</p> <ul style="list-style-type: none"> • Tanya-jawab lisan • Presentasi materi 	<p><u>Bentuk Pembelajaran:</u> Kegiatan Proses Belajar [3x50']</p> <p><u>Metode Pembelajaran:</u> Diskusi, tanya-jawab, presentasi, penugasan</p> <p><u>Tugas 14:</u> Presentasi makalah ilmiah penerapan strategi algoritma dalam pemecahan masalah</p>	<p><u>Media:</u> elearning.undiksha.ac.id</p>	<ol style="list-style-type: none"> 1. Jenis-jenis permasalahan algoritmik dalam Ilmu Komputer 2. Kelas P, NP, NP-complete, dan NP-Hard 3. Analisis kebutuhan dan batasan antara kecepatan dan efisiensi pemakaian ruang memori untuk menentukan strategi algoritma 	5%

		menentukan pilihan algoritma untuk penyelesaian masalah berdasarkan urgensi kebutuhan dan batasan, antara kecepatan dan efisiensi pemakaian ruang memori					
16	Evaluasi Akhir Semester / Ujian Akhir Semester						10%

Rujukan:

1. Introduction to The Design & Analysis of Algorithms, Anany Levitin, Pearson Education, Inc.
2. Slide Kuliah Strategi Algoritma, oleh Rinaldi Munir, Institut Teknologi Bandung.
3. Slide Analysis of Algorithms, Robert Sedgewick.
4. Modul Kuliah DAA, Made Windu Antara Kesiman, Universitas Pendidikan Ganesha.