



**PERGURUAN TINGGI** : UNIVERSITAS DIAN NUSWANTORO  
**FAKULTAS** : ILMU KOMPUTER  
**PROGRAM STUDI** : TEKNIK INFORMATIKA – S1

### RENCANA PEMBELAJARAN SEMESTER (RPS)

Mata Kuliah	Kode	Rumpun Mata Kuliah	SKS	Semester	Tanggal Penyusunan
Algoritma Pemrograman	A11.54206	Wajib Program Studi	2 Teori, 2 Praktik	2	25 Februari 2022
Otorisasi	Dosen Pengembang RPS		Koordinator RMK		Ketua Program Studi
	Abas Setiawan, S.Kom, M.Cs		Hanny Haryanto, S.Kom, M.T		Dr. Muljono S.Si, M.Kom
Capaian Pembelajaran (CP)	Capaian Pembelajaran Program Studi				
	S8	Menunjukkan sikap bertanggungjawab atas pekerjaan di bidang keahliannya secara mandiri			
	P1	Menguasai konsep teoritis bidang pengetahuan Ilmu Komputer /Informatika secara umum dan konsep teoritis bagian khusus dalam bidang pengetahuan tersebut secara mendalam, serta mampu memformulasikan penyelesaian masalah prosedural.			
	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 keahliannya.			
	KU2	Mampu menunjukkan kinerja mandiri, bermutu, dan terukur.			
	KU10	Memiliki kemampuan menggunakan teknologi informasi dan komunikasi dalam pengembangan keilmuan dan implementasi bidang keahlian.			
	KK3	Memahami konsep-konsep algoritma dan kompleksitas, meliputi konsep-konsep sentral dan kecakapan yang dibutuhkan untuk merancang, menerapkan dan menganalisis algoritma untuk menyelesaikan masalah.			
	KK4	Menguasai konsep dan prinsip algoritma serta teori ilmu komputer yang dapat digunakan dalam pemodelan dan desain sistem berbasis komputer.			
	KK5	Menguasai konsep-konsep bahasa pemrograman, serta mampu membandingkan berbagai solusi serta berbagai model bahasa pemrograman.			
	Capaian Pembelajaran Mata Kuliah				
	M1	Mengidentifikasi pentingnya algoritma dalam proses penyelesaian masalah dan bagaimana masalah dapat diselesaikan dengan beberapa algoritma yang memiliki sifat yang berbeda-beda.			
	M2	Menerapkan pemrograman dasar yang meliputi tipe primitif, variabel, assignment, input/output, percabangan, dan pengulangan.			

	M3	Menjelaskan, membuat dan menerapkan konsep dekomposisi dan abstraksi dalam pemrograman dengan tidak terbatas oleh bahasa pemrograman tertentu.
	M4	Menjelaskan dan membuat algoritma untuk penyelesaian masalah pencarian implisit ataupun eksplisit dengan tidak terbatas oleh bahasa pemrograman tertentu.
	M5	Menjelaskan, membuat, dan memilih algoritma untuk penyelesaian masalah pengurutan dengan tidak terbatas oleh bahasa pemrograman tertentu.
	M6	Menjelaskan, membuat, menerapkan, dan menganalisis fungsi rekursif didalam suatu algoritma pemrograman dengan tidak terbatas oleh bahasa pemrograman tertentu.
	M7	Membandingkan solusi iteratif dengan solusi rekursif pada proses penyelesaian masalah pada beberapa algoritma.
	M8	Mengidentifikasi, membuat, dan menggunakan struktur data <i>built-in</i> pada bahasa pemrograman tertentu
	M9	Menjelaskan, membuat, dan menggunakan <i>Abstract Data Type</i> pada pemrograman.
	M10	Mengidentifikasi dan menghitung kompleksitas algoritma sebagai pendahuluan untuk mengukur efisiensi algoritma.
	<b>Deskripsi Singkat Mata Kuliah</b> Algoritma pemrograman merupakan matakuliah dasar fundamental bagi mahasiswa Teknik Informatika S1 dalam rangka untuk memberikan pengetahuan tentang pemrograman. Matakuliah ini memerlukan prasyarat yaitu sudah menyelesaikan mata kuliah Dasar Pemrograman. Secara umum mata kuliah algoritma pemrograman memberikan pengetahuan tentang konsep-konsep dasar ilmu komputer (secara teori dan praktek) seperti dekomposisi dan abstraksi, algoritma pengurutan, analisis rekuren, algoritma pencarian, alokasi dan manajemen memory dengan pointer, dan <i>Abstract Data Type</i> (termasuk built-in ADT). Bahasa pemrograman yang mendukung mata kuliah ini adalah C/C++.	
<b>Materi Pembelajaran/ Pokok Bahasan</b>		<ol style="list-style-type: none"> <li>1. <b>Algoritma</b> untuk problem-solving sederhana.</li> <li>2. <b>Dekomposisi dan Abstraksi</b>; Notasi fungsi dan fungsi sebagai prosedur.</li> <li>3. <b>Algoritma Pengurutan</b>; <i>Monkey/bogo/stupid sort, bubble sort, selection sort, dan merge sort.</i></li> <li>4. <b>Algoritma Pencarian</b>; meliputi <i>linear/sequential search</i>, linier search dengan berbagai variasi dan <i>binary search</i></li> <li>5. <b>Analisis Rekuren</b>; Konsep dasar rekursif, penerapan rekursif pada beberapa algoritma.</li> <li>6. <b>Alokasi Memory dan Pointer</b>; pointer, array dan pointer, dan string.</li> <li>7. <b>Abstract Data Type</b>; Pengenalan struct, manipulasi struct dan typedef sebagai tipe data</li> </ol>
<b>Pustaka</b>	<b>Utama :</b>	
	<ol style="list-style-type: none"> <li>1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, Introduction to Algorithm Third Edition, MIT Press (2009)</li> <li>2. Bjarne Stroustrup, 2014, Programming: Principles and Practice Using C++ (Second Edition), Addison-Wesley Professional</li> <li>3. Ingriani Liem, Diktat Pemrograman Fungsional, Teknik Informatika, Institut Teknologi Bandung.</li> </ol>	
	<b>Pendukung :</b>	
	<ol style="list-style-type: none"> <li>1. Liem, Ingriani. Diktat Pemrograman Prosedural Informatika ITB. IF-ITB. 2007</li> <li>2. Introduction to Computer Science and Programming in Python, MIT  <a href="https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016">https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016</a> </li> </ol>	

	3. Introduction to Computer Science and Programming, MIT <a href="https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00sc-introduction-to-computer-science-and-programming-spring-2011/index.htm">https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00sc-introduction-to-computer-science-and-programming-spring-2011/index.htm</a>	
Media Pembelajaran	<b>Perangkat Lunak :</b>	<b>Perangkat Keras :</b>
	C/C++ Codeblock Command Prompt/Notepad/Notepad++/Visual Studio Code	Komputer
Tim Teaching	Abas Setiawan, S.Kom, M.Cs Danang Wahyu Utomo, M.Kom De Rosal Ign Moses, M.Kom Eko Hari Rachmawanto, M.Kom Fahri Firdausillah, S.Kom, M.CS Feri Agustina, M.Kom Ifan Rizqa, M.Kom Muhammad Saifur Rahman, S.Kom, M.CS Nisa'ul Hafidhoh, M.T	
Mata Kuliah Syarat	Dasar Pemrograman	

Mgg ke	Sub CP MK (sebagai kemampuan akhir yang diharapkan)	Indikator	Kriteria & Bentuk Penilaian	Metode Pembelajaran [Estimasi Waktu] Pengalaman Belajar	Materi pembelajaran	Bobot Penilaian
(1)	(2)	(3)	(4)	(5)	(6)	(7)
1	Setelah mengikuti matakuliah ini mahasiswa dapat menjelaskan, berdiskusi, dan menerapkan konsep algoritma secara umum dengan diikuti oleh teknik pemrograman dasar yang dibutuhkan untuk pemecahan masalah algoritmik sederhana [SDF/Algorithm and Design LO: 1,2,3,4] [SDF/Fundamental Programming Concept LO: 1 s/d 7 untuk review makul Dasar Pemrograman].	<b>TEORI</b> <ol style="list-style-type: none"> <li>1. Perkenalan dan Kontrak Kuliah.</li> <li>2. Menjelaskan algortima dan permasalahan algoritmik.</li> <li>3. Menjelaskan notasi algortima dengan paradigma fungsional.</li> <li>4. Menjelaskan konsep dasar memprogram seperti variabel, assignment, ekspresi kondisional atau percabangan, pengulangan, dan array sebagai koleksi tipe data homogen.</li> </ol> <b>Praktek</b> <ol style="list-style-type: none"> <li>1. Membuat kode program dengan menerapkan konsep dasar memprogram sekaligus sebagai <i>refresh</i> mata kuliah dan pengingat bagi mahasiswa.</li> <li>2. Kemampuan mahasiswa untuk mencoba memecahkan permasalahan komputasional sederhana.</li> </ol>	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk non test:</b> -	<b>Teori</b> <ul style="list-style-type: none"> <li>• Ceramah &amp; Diskusi [TM:2x50']</li> </ul> <b>Praktek</b> <ul style="list-style-type: none"> <li>• Tutorial singkat dan pengenalan laboratorium dan <i>tools</i>.</li> <li>• Praktikum [TM:2x50']</li> <li>• <b>Tugas 1:</b> Mahasiswa berlatih membuat program sederhana. [BT+BM: 2x(1+1)x(4x50')]</li> </ul>	<b>Teori</b> <ol style="list-style-type: none"> <li>a. Kontrak kuliah</li> <li>b. Konsep dasar algoritma dan perannya dalam pemecahan masalah.</li> <li>c. Pemecahan masalah algoritmik sederhana.</li> <li>d. Menulis Notasi algoritma.</li> </ol> <b>Praktek</b> <ol style="list-style-type: none"> <li>a. Pembuatan kode program sederhana dengan memanfaatkan konsep dasar memprogram.</li> <li>b. Menumbuhkan semangat dan minat memprogram sebagai dasar dari berpikir komputasional.</li> </ol>	
2-3	Setelah mengikuti matakuliah ini mahasiswa dapat menjelaskan konsep dekomposisi dan abstraksi serta menerapkannya pada pemrograman dengan membuat suatu fungsi atau prosedur [SDF/Algorithm and Design LO: 8][review	<b>TEORI</b> <ol style="list-style-type: none"> <li>1. Menjelaskan konsep dekomposisi yaitu memecah permasalahan algoritmik menjadi sub-sub permasalahan.</li> <li>2. Menjelaskan konsep abstraksi sebagai kotak hitam dalam melihat suatu sub permasalahan serta pentingnya dokumentasi.</li> </ol>	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk non test:</b> -	<b>Teori</b> <ul style="list-style-type: none"> <li>• Ceramah &amp; Diskusi [TM:4x50']</li> </ul> <b>Praktek</b> <ul style="list-style-type: none"> <li>• Tutorial singkat dan pengenalan laboratorium dan <i>tools</i>.</li> <li>• Praktikum [TM:4x50']</li> </ul>	<b>Teori</b> <ol style="list-style-type: none"> <li>a. Teori dekomposisi dan abstraksi.</li> <li>b. Realisasi dekomposisi dan abstraksi pada notasi algoritma sederhana</li> <li>c. Realisasi dekomposisi dan abstraksi pada notasi fungsi tanpa pengembalian</li> </ol>	

Mgg ke	Sub CP MK (sebagai kemampuan akhir yang diharapkan)	Indikator	Kriteria & Bentuk Penilaian	Metode Pembelajaran [Estimasi Waktu] Pengalaman Belajar	Materi pembelajaran	Bobot Penilaian
(1)	(2)	(3)	(4)	(5)	(6)	(7)
	<b>makul Dasar Pemrograman].</b>	3. Menjelaskan realisasi dekomposisi dengan suatu fungsi dan realisasi abstraksi didalam komentar suatu fungsi dan prosedur. 4. Menjelaskan Parameter formal dan aktual dalam fungsi  <b>Praktek</b> 3. Membuat kode program dengan fungsi dengan pengembalian untuk perhitungan aritmatika, bangun datar/ruang, dan sub-permasalahan komputasional lain. 4. Kemampuan mahasiswa untuk memanggil fungsi-fungsi yang telah dibuat menjadi satu kesatuan program untuk memecahkan permasalahan komputasional.		<ul style="list-style-type: none"> <li><b>Tugas 1:</b> Mahasiswa berlatih membuat beberapa fungsi atau prosedur <b>[BT+BM: 4x(1+1)x(8x50')]</b></li> </ul>	d. Parameter formal dan aktual  <b>Praktek</b> a. Pembuatan kode program dengan bentuk fungsi dengan pengembalian. b. Pemanggilan fungsi kedalam main program. c. Fungsi-fungsi yang dibuat masih dalam satu file dengan main program. d. Pembuatan kode program dengan bentuk fungsi tanpa pengembalian atau fungsi sebagai prosedur. e. Pemanggilan prosedur kedalam main program. f. Fungsi-fungsi (termasuk prosedur) yang dibuat sudah dalam file yang berbeda dengan main program. g. Parameter fungsi h. Variabel global vs lokal	
4	Setelah mengikuti matakuliah ini mahasiswa mampu mengenal permasalahan komputasional sederhana dengan algoritma sederhana yang umum digunakan <b>[SDF/Algorithm and Design LO: 1,2,3,4].</b>	<b>TEORI</b> 1. Menjelaskan tentang deret angka yang berbentuk list dan pengolahannya seperti penggunaan fungsi penjumlahan dan beberapa pengolahan statistik dasar. 2. Menjelaskan tentang algoritma pertukaran sederhana.	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk non test:</b> -	<b>Teori</b> <ul style="list-style-type: none"> <li>Ceramah &amp; Diskusi <b>[TM:2x50']</b></li> </ul> <b>Praktek</b> <ul style="list-style-type: none"> <li>Tutorial singkat.</li> <li>Praktikum <b>[TM:2x50']</b></li> <li><b>Tugas 2:</b> Mahasiswa berlatih menerapkan</li> </ul>	<b>Teori</b> a. Penjumlahan deret angka dan menghitung rata-rata. b. Penjumlahan elemen-elemen nested list (seakan matriks). c. Metode pertukaran variabel. d. Konsep Algoritma euclidean.	

Mgg ke	Sub CP MK (sebagai kemampuan akhir yang diharapkan)	Indikator	Kriteria & Bentuk Penilaian	Metode Pembelajaran [Estimasi Waktu] Pengalaman Belajar	Materi pembelajaran	Bobot Penilaian
(1)	(2)	(3)	(4)	(5)	(6)	(7)
		3. Menjelaskan tentang algoritma euclidean.  <b>Praktek</b> 1. Membuat kode program untuk pengolahan statistik sederhana pada data deret angka yang berbentuk list.  2. Membuat kode program untuk menukar nilai baik itu numerik maupun string.  3. Membuat kode program faktor persekutuan terbesar dengan menerapkan algoritma euclidean.		beberapa algoritma sederhana. <b>[BT+BM: 2x(1+1)x(4x50')]</b>	<b>Praktek</b> a. Pembuatan kode program dengan menerapkan fungsi penjumlahan dan pengolahan statistik sederhana. b. Pembuatan fungsi sederhana untuk menjumlahkan nested list (matriks) c. Pembuatan beberapa fungsi atau prosedur pertukaran. d. Pembuatan kode untuk menyelesaikan permasalahan FPB dengan algoritma euclidean.	
5	Setelah mengikuti matakuliah ini mahasiswa mampu menjelaskan dan menerapkan algoritma pengurutan ( <i>sorting</i> ) <i>bogo sort</i> , <i>bubble sort</i> , dan <i>selection sort</i> . <b>[SDF/Algorithm and Design LO: 1,2,3,4]</b>	<b>TEORI</b> 1. Menjelaskan konsep dasar algoritma pengurutan. 2. Penyelesaian kasus pengurutan dengan bogo sort. 3. Penyelesaian kasus pengurutan dengan bubble sort. 4. Penyelesaian kasus pengurutan dengan selection sort.  <b>Praktek</b> 1. Membuat kode program untuk kasus pengurutan dengan <i>bogo sort</i> , <i>bubble sort</i> , dan <i>selection sort</i> yang diimplementasikan pada suatu fungsi/prosedur.	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk non test:</b> -	<b>Teori</b> • Ceramah & Diskusi <b>[TM:2x50']</b>  <b>Praktek</b> • Tutorial singkat. • Praktikum <b>[TM:2x50']</b> • <b>Tugas 5:</b> Mahasiswa berlatih menerapkan penyelesaian pengurutan pada suatu koleksi array/list dengan algoritma <i>bogo sort</i> , <i>bubble sort</i> , dan <i>selection sort</i> didalam	<b>Teori</b> a. Konsep dasar algoritma pengurutan. b. Algoritma pengurutan <i>bogo sort</i> , <i>bubble sort</i> , dan <i>selection sort</i> secara iteratif. c. Analisis kelebihan dan kekurangan algoritma <i>bogo sort</i> , <i>bubble sort</i> , dan <i>selection sort</i> .  <b>Praktek</b> a. Pembuatan kode program untuk penyelesaian pengurutan dengan <i>bogo sort</i> .	

Mg ke	Sub CP MK (sebagai kemampuan akhir yang diharapkan)	Indikator	Kriteria & Bentuk Penilaian	Metode Pembelajaran [Estimasi Waktu] Pengalaman Belajar	Materi pembelajaran	Bobot Penilaian
(1)	(2)	(3)	(4)	(5)	(6)	(7)
		2. Analisis dan membandingkan ketiga algoritma yang telah diimplementasikan sebelumnya dengan melihat kelebihan dan kekurangan masing-masing.		suatu fungsi atau prosedur. <b>[BT+BM: 2x(1+1)x(4x50')]</b>	b. Pembuatan kode program untuk penyelesaian pengurutan dengan <i>bubble sort</i> . c. Pembuatan kode program untuk penyelesaian pengurutan dengan <i>selection sort</i> .	
6	Setelah mengikuti matakuliah ini mahasiswa mampu menjelaskan dan menerapkan algoritma pengurutan ( <i>sorting</i> ) dengan algoritma <i>insertion sort</i> <b>[SDF/Algorithm and Design LO: 1,2,3,4,7]</b>	<b>TEORI</b> 1. Penyelesaian kasus pengurutan dengan <i>insertion sort</i> . 2. Penyelesaian kasus pengurutan dengan <i>merge sort</i> .  <b>Praktek</b> 1. Membuat kode program untuk kasus pengurutan dengan <i>insertion sort</i> yang diimplementasikan pada suatu fungsi/prosedur. 2. Analisis dan membandingkan kedua algoritma yang baru dipelajari dengan ketiga algoritma yang telah diimplementasikan pada pertemuan sebelumnya dengan melihat kelebihan dan kekurangan masing-masing.	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk non test:</b> -	<b>Teori</b> • Ceramah & Diskusi <b>[TM:2x50']</b>  <b>Praktek</b> • Tutorial singkat. • Praktikum <b>[TM:2x50']</b> • <b>Tugas 6:</b> Mahasiswa berlatih menerapkan penyelesaian pengurutan pada suatu koleksi array/list dengan algoritma <i>insertion sort</i> , <i>merge sort</i> , dan boleh algoritma <i>sorting</i> yang lain seperti <i>quick sort</i> atau <i>radix sort</i> didalam suatu fungsi atau prosedur. <b>[BT+BM: 2x(1+1)x(4x50')]</b>	<b>Teori</b> a. Algoritma pengurutan <i>insertion sort</i> secara iteratif. b. Analisis kelebihan dan kekurangan algoritma <i>insertion sort</i> .  <b>Praktek</b> a. Pembuatan kode program untuk penyelesaian pengurutan dengan <i>insertion sort</i> . b.	
7	Setelah mengikuti matakuliah ini mahasiswa mampu menjelaskan dan	<b>TEORI</b>	<b>Kriteria :</b> Ketepatan dan penguasaan	<b>Teori</b> • Ceramah & Diskusi	<b>Teori</b>	

Mgg ke	Sub CP MK (sebagai kemampuan akhir yang diharapkan)	Indikator	Kriteria & Bentuk Penilaian	Metode Pembelajaran [Estimasi Waktu] Pengalaman Belajar	Materi pembelajaran	Bobot Penilaian
(1)	(2)	(3)	(4)	(5)	(6)	(7)
	menerapkan algoritma pencarian linear dan pencarian biner [SDF/Algorithm and Design LO: 1,2,3,4,7]	<ol style="list-style-type: none"> <li>Menjelaskan konsep dasar algoritma pencarian.</li> <li>Algoritma pencarian linier</li> <li>Algoritma pencarian linier dengan sentinel</li> <li>Algoritma pencarian linier dengan list terurut</li> <li>Penyelesaian pencarian eksplisit dengan algoritma biner.</li> </ol> <p><b>Praktek</b></p> <ol style="list-style-type: none"> <li>Membuat kode program untuk kasus pencarian pada list dengan algoritma pencarian linear.</li> <li>Membuat kode program untuk kasus pencarian pada list dengan algoritma pencarian linear dengan sentinel.</li> <li>Membuat kode program untuk kasus pencarian pada list terurut dengan algoritma pencarian linear.</li> <li>Membuat kode program untuk kasus pencarian eksplisit pada suatu koleksi list dengan algoritma biner.</li> </ol>	<p><b>Bentuk non test:</b></p> <p>-</p>	<p>[TM:2x50']</p> <p><b>Praktek</b></p> <ul style="list-style-type: none"> <li>Tutorial singkat.</li> <li>Praktikum</li> </ul> <p>[TM:2x50']</p> <p><b>Tugas 3:</b> Mahasiswa berlatih menerapkan penyelesaian pencarian linier didalam suatu fungsi atau prosedur.</p> <p>[BT+BM: 2x(1+1)x(4x50')]</p>	<ol style="list-style-type: none"> <li>Konsep dasar algoritma pencarian.</li> <li>Konsep Pencarian linier</li> <li>Konsep pencarian linier dengan sentinel</li> <li>Konsep pencarian linier pada list terurut.</li> <li>Konsep algoritma pencarian biner.</li> </ol> <p><b>Praktek</b></p> <ol style="list-style-type: none"> <li>Pembuatan kode program untuk algoritma pencarian linier.</li> <li>Pembuatan kode program untuk algoritma pencarian linier dengan sentinel.</li> <li>Pembuatan kode program untuk algoritma pencarian linier pada list terurut.</li> <li>Pembuatan kode program untuk penyelesaian pencarian dengan algoritma pencarian biner.</li> </ol>	
8	<b>UJIAN TENGAH SEMESTER</b>					
9	Setelah mengikuti matakuliah ini mahasiswa mampu menjelaskan konsep analisis rekuren, membuat fungsi rekursif, dan menganalisis ekspresi rekursif. [SDF/Algorithm and Design LO: 5]	<p><b>TEORI</b></p> <ol style="list-style-type: none"> <li>Penjelasan tentang konsep analisis rekuren.</li> <li>Membuat fungsi rekursif pada kasus permasalahan komputasional sederhana.</li> <li>Menganalisis ekspresi rekursif untuk mengetahui alur kerja fungsi</li> </ol>	<p><b>Kriteria :</b></p> <p>Ketepatan dan penguasaan</p> <p><b>Bentuk non test:</b></p> <p>-</p>	<p><b>Teori</b></p> <ul style="list-style-type: none"> <li>Ceramah &amp; Diskusi</li> </ul> <p>[TM:2x50']</p> <p><b>Praktek</b></p> <ul style="list-style-type: none"> <li>Tutorial singkat.</li> <li>Praktikum</li> </ul> <p>[TM:2x50']</p>	<p><b>Teori</b></p> <ol style="list-style-type: none"> <li>Konsep analisis rekuren</li> <li>Fungsi rekursif didalam algoritma pemrograman</li> <li>Analisis ekspresi rekursif</li> </ol> <p><b>Praktek</b></p>	



Mg ke	Sub CP MK (sebagai kemampuan akhir yang diharapkan)	Indikator	Kriteria & Bentuk Penilaian	Metode Pembelajaran [Estimasi Waktu] Pengalaman Belajar	Materi pembelajaran	Bobot Penilaian
(1)	(2)	(3)	(4)	(5)	(6)	(7)
	[SDF/Fundamental Programming Concept LO: 8,9].	rekursif didalam penyelesaian permasalahan komputasional.  <b>Praktek</b> 1. Membuat kode program untuk kasus komputasional sederhana dengan fungsi rekursif (misalnya: faktorial atau fibonnaci atau perkalian. 2. Melakukan analisis ekspresi rekursif pada program sederhana yang dibuat sebelumnya.		<ul style="list-style-type: none"> <li><b>Tugas 7:</b> Mahasiswa berlatih menerapkan analisis rekuren pada kasus yang belum dipelajari dikelas. <b>[BT+BM: 2x(1+1)x(4x50')]</b></li> </ul>	a. Pembuatan kode program untuk penyelesaian kasus sederhana dengan fungsi rekursif. b. Analisis dari fungsi yang telah dibuat sebelumnya.	
10	Setelah mengikuti matakuliah ini mahasiswa mampu menerapkan analisis rekuren pada algoritma pencarian dan pengurutan serta mampu membandingkan solusi iteratif dengan solusi rekursif. <b>[SDF/Algorithm and Design LO: 5,6]</b>	<b>TEORI</b> 1. Lanjutan dari pertemuan sebelumnya tentang pendalaman materi analisis rekuren. 2. Konsep divide and conquer 3. Membuat fungsi rekursif pada algoritma pencarian dan pengurutan yang sebelumnya sudah dilakukan dengan metode iteratif. 4. Membandingkan antara solusi iteratif dengan rekursif pada algoritma sederhana. 5. Memperkirakan kapan menggunakan solusi rekursif atau iteratif.  <b>Praktek</b> 1. Membuat kode program fungsi rekursif pada algoritma pencarian dan pengurutan.	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk non test:</b> -	<b>Teori</b> <ul style="list-style-type: none"> <li>Ceramah &amp; Diskusi <b>[TM:2x50']</b></li> </ul> <b>Praktek</b> <ul style="list-style-type: none"> <li>Tutorial singkat.</li> <li>Praktikum <b>[TM:2x50']</b></li> <li><b>Tugas 7:</b> Mahasiswa berlatih menerapkan analisis rekuren pada algoritma yang sudah dikenal secara iteratif. <b>[BT+BM: 2x(1+1)x(4x50')]</b></li> </ul>	<b>Teori</b> a. Pendalaman konsep analisis rekuren b. Konsep divide and conquer c. Fungsi rekursif pada <i>linear search</i> d. Fungsi rekursif pada <i>merge sort</i> e. Perbandingan solusi iteratif dengan rekursif.  <b>Praktek</b> a. Pembuatan kode program untuk penyelesaian kasus pencarian dan pengurutan dengan fungsi rekursif.	

Mgg ke	Sub CP MK (sebagai kemampuan akhir yang diharapkan)	Indikator	Kriteria & Bentuk Penilaian	Metode Pembelajaran [Estimasi Waktu] Pengalaman Belajar	Materi pembelajaran	Bobot Penilaian
(1)	(2)	(3)	(4)	(5)	(6)	(7)
11-12	Setelah mengikuti matakuliah ini mahasiswa mampu menjelaskan dan menerapkan pointer untuk alokasi dinamis, manajemen memory pada program, keterkaitan array dengan pointer, dan string. <b>[SDF/Fundamental Data Structure LO: 1]</b>	<b>TEORI</b> <ol style="list-style-type: none"> <li>Menjelaskan dan menerapkan pointer untuk pengalamatan memory.</li> <li>Menjelaskan dan menerapkan alokasi dinamis.</li> <li>Membandingkan pointer dengan array.</li> <li>Menjelaskan dan menerapkan String.</li> </ol> <b>Praktek</b> <ol style="list-style-type: none"> <li>Membuat kode program fungsi dan/atau prosedur dengan memanfaatkan pointer untuk alokasi dinamis, array dari pointer, dan pengelolaan tipe string.</li> </ol>	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk non test:</b> -	<b>Teori</b> <ul style="list-style-type: none"> <li>Ceramah &amp; Diskusi <b>[TM:4x50']</b></li> </ul> <b>Praktek</b> <ul style="list-style-type: none"> <li>Tutorial singkat.</li> <li>Praktikum <b>[TM:4x50']</b></li> <li><b>Tugas 8:</b> Mahasiswa berlatih menerapkan pointer pada penyelesaian masalah sederhana. <b>[BT+BM: 4x(1+1)x(8x50')]</b></li> </ul>	<b>Teori</b> <ol style="list-style-type: none"> <li>Pointer dan Alokasi Memory</li> <li>Array vs Pointer</li> <li>Manipulasi String</li> </ol> <b>Praktek</b> <ol style="list-style-type: none"> <li>Pembuatan kode program dengan memanfaatkan pointer.</li> <li>Pembuatan kode program dengan array dan pointer sebagai array.</li> <li>Pembuatan kode program yang memanfaatkan string.</li> </ol>	
13-14	Setelah mengikuti matakuliah ini mahasiswa mampu mengidentifikasi dan menerapkan komponen data dan perilaku dari satu Abstract data Type (ADT). <b>[SDF/Algorithm and Design LO: 9]</b>	<b>TEORI</b> <ol style="list-style-type: none"> <li>Menjelaskan tentang paradigma pemrograman berorientasi objek.</li> <li>Menerapkan struct sebagai Abstract Data Type pada bahasa pemrograman C/C++.</li> <li>Mengidentifikasi dan menggunakan variable/property dan method</li> </ol> <b>Praktek</b> <ol style="list-style-type: none"> <li>Membuat ADT sederhana.</li> <li>Menerapkan ADT dengan meng-<i>instance</i> dan mengakses variabel serta fungsi yang ada.</li> </ol>	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk non test:</b> -	<b>Teori</b> <ul style="list-style-type: none"> <li>Ceramah &amp; Diskusi <b>[TM:4x50']</b></li> </ul> <b>Praktek</b> <ul style="list-style-type: none"> <li>Tutorial singkat.</li> <li>Praktikum <b>[TM:4x50']</b></li> <li><b>Tugas 9:</b> Mahasiswa berlatih ADT sederhana pada penyelesaian masalah sederhana. <b>[BT+BM: 4x(1+1)x(8x50')]</b></li> </ul>	<b>Teori</b> <ol style="list-style-type: none"> <li>Pengertian ADT</li> <li>Akses memory ADT</li> <li>Pembentukan dan penggunaan ADT</li> <li>Variabel dan fungsi dalam ADT</li> </ol> <b>Praktek</b> <ol style="list-style-type: none"> <li>Pembuatan kode program dengan ADT sederhana pada suatu struct disertai dengan variabel dan fungsinya.</li> <li>Penerapan ADT yang sudah dibuat sebelumnya pada</li> </ol>	

Mg ke	Sub CP MK (sebagai kemampuan akhir yang diharapkan)	Indikator	Kriteria & Bentuk Penilaian	Metode Pembelajaran [Estimasi Waktu] Pengalaman Belajar	Materi pembelajaran	Bobot Penilaian
(1)	(2)	(3)	(4)	(5)	(6)	(7)
					suatu penyelesaian masalah algoritmik.	
15	Setelah mengikuti matakuliah ini mahasiswa mampu mengidentifikasi, menganalisis dan membandingkan beberapa algoritma dalam hal efisiensi. <b>[SDF/Algorithm and Design LO: 1,2,3, 11]</b>	<b>TEORI</b> <ol style="list-style-type: none"> <li>Menjelaskan konsep dan properti suatu algoritma.</li> <li>Mengenal efisiensi algoritma dengan notasi Big-O.</li> <li>Membandingkan secara informal tentang efisiensi beberapa algoritma (sorting dan searching)</li> </ol> <b>Praktek</b> <ol style="list-style-type: none"> <li>Membuat dan menerapkan beberapa konsep pada pertemuan-pertemuan sebelumnya pada suatu permasalahan tingkat menengah/kompleks.</li> <li>Melakukan review terhadap praktikum-praktikum sebelumnya dan mengidentifikasi performa dalam hal kompleksitas algoritmanya.</li> </ol>	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk non test:</b> -	<b>Teori</b> <ul style="list-style-type: none"> <li>Ceramah &amp; Diskusi <b>[TM:2x50']</b></li> </ul> <b>Praktek</b> <ul style="list-style-type: none"> <li>Tutorial singkat.</li> <li>Praktikum <b>[TM:2x50']</b></li> <li><b>Tugas 10:</b> Mahasiswa berlatih menyelesaikan masalah algoritmik tingkat menengah dengan menerapkan konsep-konsep pada pertemuan-pertemuan sebelumnya. <b>[BT+BM: 2x(1+1)x(4x50')]</b></li> </ul>	<b>Teori</b> <ol style="list-style-type: none"> <li>Konsep dan properti Algoritma</li> <li>Analisis efektifitas algoritma kasus terbaik, rata-rata, dan terburuk.</li> <li>Notasi Big-O dan tingkatan kompleksitas algoritma.</li> <li>Membandingkan efisiensi dari segi kompleksitas dengan notasi Big-O pada algoritma-algoritma pengurutan dan pencarian.</li> </ol> <b>Praktek</b> <ol style="list-style-type: none"> <li>Pembuatan kode program dengan menerapkan konsep-konsep pada pertemuan-pertemuan sebelumnya yang digunakan sebagai review perkuliahan untuk menangani kasus penyelesaian masalah tingkat menengah.</li> </ol>	
16	UJIAN AKHIR SEMESTER					

**Catatan :**

[1]. TM : tatap Muka

[2]. **[TM:3x50']** : Kuliah tatap muka 1 kali (minggu) x 3 sks x 50 menit=150 menit

[3]. **[BT+BM:(1+1)x(2x50')]** : Belajar terstruktur 1 kali (minggu) dan belajar mandiri 1 kali (minggu) x 3 sks x 50 menit = 300 menit ( 5 jam)

- [4]. RPS : Rencana Pembelajaran Semester, RMK : Rumpun Mata Kuliah, Prodi : Program Studi
- [5] Ujian Tengah Semester dapat dilakukan dengan tes pemrograman individu
- [6] Ujian Akhir Semester merupakan presentasi proyek akhir mahasiswa
- [7] Penugasan didalam rancangan RPS ini merupakan penugasan utama, dosen dapat merubah atau menambah sesuai dengan kebutuhan
- [8] **SDF merupakan singkatan dari Software Development Fundamental dan LO adalah Learning Outcome dari kurikulum ACM/IEEE 2013 (Computer Science Curricula 2013, ACM – IEEE) terdapat 3 topik/bisa jadi matakuliah yang terkait yaitu SDF/Fundamental Programming Concept, SDF/Algorithm and Design, dan SDF/Fundamental Data Structure.**

