

	RENCANA PEMBELAJARAN SEMESTER PROGRAM STUDI S1 TEKNIK LOGISTIK FAKULTAS TEKNIK UNIVERSITAS IBNU SINA						
Identitas Mata Kuliah	NAMA MK	KODE MK	RUMPUN MATA KULIAH	BOBOT(SKS)		SEMESTER	Direvisi
	PRAKTIKUM ALGORITMA PEMROGRAMAN	4TLMKK122	Teknik	1	SKS	4	01/08/2025
Otoritas	Pengembang RPS			Ketua Kelompok Keahlian		Ka PRODI	
	TIM Microteaching			TIM Microteaching		Dr. A. L. Setyabudhi, S.T., M.M T., IPM	
Deskripsi Mata Kuliah	Mata kuliah Praktikum Algoritma Pemrograman memberikan pengalaman langsung kepada mahasiswa dalam merancang, menuliskan, dan mengimplementasikan algoritma menggunakan bahasa pemrograman. Mahasiswa dilatih untuk memahami logika pemrograman dasar, struktur data sederhana, kontrol alur program, serta pemecahan masalah melalui pendekatan algoritmik. Melalui praktikum ini, mahasiswa diharapkan mampu mengembangkan keterampilan komputasional dan mengaplikasikan algoritma untuk menyelesaikan berbagai permasalahan di bidang teknik logistik maupun disiplin ilmu terkait.						
Capaian Pembelajaran Lulusan & Capaian Pembelajaran Mata Kuliah	Capaian Pembelajaran Lulusan (CPL) PRODI						
	CPL02	Mampu mengumpulkan, menganalisis, dan menerapkan data untuk mendukung pengambilan keputusan dalam bidang logistik					
Pembelajaran Mata Kuliah	CPL10	Mampu mengembangkan solusi berbasis teknologi informasi untuk mendukung efisiensi sistem logistik.					
	Kode CPMK	Capaian Pembelajaran Mata Kuliah				CPL yang Didukung	
	CPMK109	Memahami konsep dasar algoritma & logika pemrograman				CPL02	
	CPMK110	Mengimplementasikan struktur kontrol (percabangan & perulangan)				CPL02	
	CPMK111	Menggunakan array (1D & 2D) untuk pemrosesan data				CPL02, CPL10	

Ambang Batas Kelulusan Mahasiswa	50.01	
Ambang Batas Kelulusan MK	85.00%	

Minggu Ke-	Kemampuan akhir tiap tahapan belajar (Sub- CPMK)	Penilaian		Bentuk Pembelajaran; Metode Pembelajaran; Penugasan Mahasiswa; [Estimasi Waktu]		Materi Pembelajaran [Pustaka]	Bobot Penilaian (%)
		Indikator	Kriteria & Teknik				
(1)	(2)	(3)	(4)	Luring(5)	Daring(6)	(7)	(8)

1	Mahasiswa mampu memahami konsep dasar algoritma, logika pemrograman, serta mengenali lingkungan pemrograman yang akan digunakan pada praktikum.	a. Mahasiswa dapat menjelaskan pengertian algoritma dan pemrograman. b. Mahasiswa mampu mengidentifikasi langkah-langkah dasar dalam penulisan algoritma sederhana. c. Mahasiswa mampu menggunakan IDE atau compiler untuk menulis kode sederhana.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Observasi praktikum, tanya jawab, dan penilaian hasil latihan.	Pengenalan mata kuliah & aturan praktikum. Demonstrasi penggunaan perangkat lunak pemrograman (IDE/compiler). Diskusi kelompok kecil untuk merumuskan contoh algoritma sederhana. Estimasi Waktu: 2 × 50 menit.	Materi pengantar algoritma dan pemrograman diunggah dalam bentuk slides dan video. Forum diskusi mengenai contoh algoritma sehari-hari. Estimasi Waktu: 60 menit.	Materi Pembelajaran [Pustaka] Konsep algoritma & pemrograman dasar. Pengenalan lingkungan pemrograman (compiler/IDE). Contoh algoritma sederhana dalam kehidupan sehari-hari. Referensi: Deitel, P. & Deitel, H. (2017). C: How to Program (8th Edition). Pearson. Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language (2nd Edition). Prentice Hall.	2%
2	Mahasiswa mampu memahami struktur dasar program, menuliskan sintaks sederhana, serta menggunakan perintah input-output untuk membuat program sederhana.	a. Mahasiswa dapat menjelaskan struktur dasar program (deklarasi, main, perintah). b. Mahasiswa mampu menuliskan kode sederhana dengan input-output. c. Mahasiswa dapat menjalankan program tanpa error sederhana.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Penilaian latihan praktikum, observasi, dan review kode program.	Penjelasan struktur program dasar. Praktik menulis program sederhana dengan perintah input-output. Latihan membuat program membaca dan menampilkan data mahasiswa. Estimasi Waktu: 2 × 50 menit.	Materi tambahan berupa contoh program sederhana. Diskusi online mengenai kesalahan umum saat menulis program. Tugas individu: membuat program input-output dengan minimal 2 variabel. Estimasi Waktu: 60 menit.	Materi Pembelajaran [Pustaka] Struktur dasar program. Perintah input-output (scanf/printf atau cin/cout tergantung bahasa). Contoh program sederhana. Referensi: Deitel, P. & Deitel, H. (2017). C: How to Program (8th Edition). Pearson. Malik, D. S. (2018). Data Structures Using C++ (2nd Edition). Cengage Learning.	2%
3	Mahasiswa mampu memahami dan mengimplementasikan struktur kontrol percabangan (if, if-else, nested-if) dalam program sederhana untuk menyelesaikan masalah logika.	a. Mahasiswa dapat menjelaskan konsep percabangan dalam algoritma. b. Mahasiswa mampu menuliskan kode dengan if/else sesuai kasus. c. Mahasiswa dapat menguji program percabangan dengan	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Penilaian hasil latihan praktikum, uji coba program, dan review kode.	Penjelasan teori singkat percabangan (if, if-else, nested-if). Demonstrasi program percabangan sederhana. Praktik: membuat program menentukan nilai lulus/tidak lulus berdasarkan skor. Estimasi Waktu: 2 × 50 menit.	Modul singkat dan contoh kode percabangan. Forum diskusi: membahas contoh kasus percabangan dalam kehidupan sehari-hari (misalnya: cek diskon harga, penentuan kategori nilai). Tugas individu: membuat program percabangan	Materi Pembelajaran [Pustaka] Konsep percabangan (if, if-else, nested-if). Contoh kasus penerapan percabangan. Latihan program sederhana dengan berbagai kondisi. Referensi: Deitel, P. & Deitel, H. (2017). C: How to Program	2%

		berbagai input.			dengan minimal 3 kondisi berbeda. Estimasi Waktu: 60 menit.	(8th Edition). Pearson. Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language (2nd Edition). Prentice Hall.	
4	Mahasiswa mampu memahami dan mengimplementasikan struktur kontrol switch-case serta penggunaan ekspresi logika (boolean) dalam menyelesaikan permasalahan pemrograman sederhana.	a. Mahasiswa dapat menjelaskan konsep switch-case dan logika boolean. b. Mahasiswa mampu menulis kode program dengan switch-case untuk memilih berbagai kondisi. c. Mahasiswa dapat memanfaatkan operator logika (AND, OR, NOT) dalam percabangan.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Penilaian hasil latihan praktikum, observasi pelaksanaan program, dan review kode.	Penjelasan perbedaan if-else dan switch-case. Demonstrasi program sederhana menggunakan switch-case. Praktikum: membuat program menu sederhana (contoh: kalkulator sederhana dengan 4 operasi aritmatika). Estimasi Waktu: 2 × 50 menit.	Materi tambahan berupa slides dan contoh kode. Diskusi: membandingkan efisiensi if-else dan switch-case dalam kasus nyata. Tugas individu: membuat program dengan switch-case yang melibatkan minimal 4 pilihan menu. Estimasi Waktu: 60 menit.	Materi Pembelajaran [Pustaka] Struktur kontrol switch-case. Operator logika (&&, , !). Studi kasus pemilihan menu program sederhana. Referensi: Deitel, P. & Deitel, H. (2017). C: How to Program (8th Edition). Pearson. Malik, D. S. (2018). Data Structures Using C++ (2nd Edition). Cengage Learning.	2%
5	Mahasiswa mampu memahami konsep perulangan (looping) dan mengimplementasikan struktur for, while, dan do-while dalam menyelesaikan masalah pemrograman yang membutuhkan eksekusi berulang.	a. Mahasiswa dapat menjelaskan konsep perulangan dan perbedaannya dengan percabangan. b. Mahasiswa mampu menuliskan program dengan for, while, dan do-while. c. Mahasiswa dapat memilih jenis perulangan yang sesuai dengan kasus.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Penilaian hasil latihan praktikum, pengujian program, dan review kode.	Penjelasan teori singkat perulangan for, while, dan do-while. Demonstrasi program sederhana dengan perulangan. Praktikum: membuat program menghitung jumlah bilangan, menampilkan tabel perkalian, atau deret Fibonacci sederhana. Estimasi Waktu: 2 × 50 menit.	Materi tambahan berupa contoh kasus perulangan. Diskusi online: kapan menggunakan for, while, atau do-while. Tugas individu: membuat program perulangan dengan minimal 2 jenis loop yang berbeda. Estimasi Waktu: 60 menit.	Materi Pembelajaran [Pustaka] Konsep perulangan (looping). Struktur for, while, dan do-while. Studi kasus perulangan sederhana (deret, tabel, perhitungan). Referensi: Deitel, P. & Deitel, H. (2017). C: How to Program (8th Edition). Pearson. Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language (2nd Edition). Prentice Hall.	2%
6	Mahasiswa mampu memahami dan mengimplementasikan nested loop (perulangan bersarang) serta menggunakannya	a. Mahasiswa dapat menjelaskan konsep perulangan bersarang. b. Mahasiswa mampu menuliskan program dengan nested loop.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Penilaian hasil	Penjelasan teori singkat tentang nested loop. Demonstrasi program pola sederhana menggunakan perulangan bersarang. Praktikum: membuat	Materi tambahan berupa contoh kasus nested loop. Diskusi online: tantangan logika dalam nested loop (misalnya pola	Materi Pembelajaran [Pustaka] Konsep nested loop (for dalam for, while dalam while). Studi kasus: pola bintang, tabel perkalian, matriks	2%

	untuk menyelesaikan masalah pemrograman yang melibatkan pola, tabel, atau data berbentuk matriks sederhana.	c. Mahasiswa dapat menghasilkan output pola atau tabel sesuai instruksi.	praktikum, uji coba program, dan review kode.	program pola bintang segitiga, tabel perkalian, dan menampilkan isi matriks. Estimasi Waktu: 2 × 50 menit.	piramida, angka berurutan). Tugas individu: membuat program menampilkan pola tertentu dengan nested loop. Estimasi Waktu: 60 menit.	sederhana. Referensi: Deitel, P. & Deitel, H. (2017). C: How to Program (8th Edition). Pearson. Malik, D. S. (2018). Data Structures Using C++ (2nd Edition). Cengage Learning.	
7	Mahasiswa mampu memahami konsep array satu dimensi, mendeklarasikan, mengisi, dan mengakses elemen array untuk menyelesaikan permasalahan penyimpanan data dalam jumlah banyak.	a. Mahasiswa dapat menjelaskan konsep array satu dimensi. b. Mahasiswa mampu menuliskan program untuk menyimpan dan menampilkan data menggunakan array. c. Mahasiswa dapat mengolah data array dengan perulangan (contoh: menghitung jumlah, rata-rata, atau mencari nilai maksimum/minimum).	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Penilaian latihan praktikum, observasi, dan review kode.	Penjelasan konsep dasar array satu dimensi. Demonstrasi program array untuk menyimpan data numerik. Praktikum: membuat program menghitung rata-rata nilai mahasiswa menggunakan array. Estimasi Waktu: 2 × 50 menit.	Materi tambahan berupa contoh program array. Forum diskusi: kapan sebaiknya data disimpan menggunakan array. Tugas individu: membuat program array yang dapat mencari nilai maksimum dan minimum dari sekumpulan data. Estimasi Waktu: 60 menit.	Materi Pembelajaran [Pustaka] Konsep array satu dimensi. Deklarasi, inisialisasi, dan akses elemen array. Studi kasus sederhana: pengolahan data nilai mahasiswa. Referensi: Deitel, P. & Deitel, H. (2017). C: How to Program (8th Edition). Pearson. Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language (2nd Edition). Prentice Hall.	2%
8	ujian tengah semester (UTS) Mahasiswa mampu mengintegrasikan pemahaman tentang struktur dasar program, percabangan, perulangan, nested loop, dan array satu dimensi untuk menyelesaikan permasalahan pemrograman sederhana secara mandiri.	a. Mahasiswa dapat menuliskan program sesuai soal ujian tanpa kesalahan sintaks. b. Mahasiswa mampu menerapkan logika percabangan dan perulangan dalam satu program. c. Mahasiswa dapat memanfaatkan array untuk mengolah data sesuai kebutuhan soal.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Ujian praktik di laboratorium, penilaian hasil kode program, serta uji coba langsung oleh dosen pengampu.	Pelaksanaan ujian praktik pemrograman individu. Mahasiswa mengerjakan soal berbentuk studi kasus sederhana dengan waktu terbatas. Estimasi Waktu: 2 × 50 menit.	Ujian dapat dilakukan dengan sistem take-home project berbatas waktu, diserahkan melalui LMS. Estimasi Waktu: sesuai ketentuan dosen.	Materi Pembelajaran [Pustaka] Seluruh materi pertemuan Minggu ke-1 s.d. Minggu ke-7. Referensi: Deitel, P. & Deitel, H. (2017). C: How to Program (8th Edition). Pearson. Malik, D. S. (2018). Data Structures Using C++ (2nd Edition). Cengage Learning.	20%
9	Mahasiswa mampu memahami konsep array multidimensi,	a. Mahasiswa dapat menjelaskan perbedaan array	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89	Penjelasan konsep array multidimensi (khususnya array 2D).	Materi tambahan berupa contoh program array 2D. Forum diskusi: penerapan	Materi Pembelajaran [Pustaka] Konsep array multidimensi.	2%

	mendeklarasikan, menginisialisasi, serta mengakses elemen-elemen array dua dimensi untuk menyelesaikan permasalahan pemrograman sederhana.	satu dimensi dan multidimensi. b. Mahasiswa mampu menuliskan program sederhana menggunakan array dua dimensi. c. Mahasiswa dapat menyelesaikan kasus pengolahan data berbentuk tabel/matriks dengan array multidimensi.	Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Penilaian hasil praktikum, observasi pelaksanaan program, dan review kode.	Demonstrasi program tabel menggunakan array dua dimensi. Praktikum: membuat program pengolahan nilai mahasiswa berbentuk tabel (misalnya nilai beberapa mahasiswa pada beberapa mata kuliah). Estimasi Waktu: 2 × 50 menit.	array multidimensi dalam kasus nyata (misalnya matriks, jadwal, data logistik). Tugas individu: membuat program array 2D untuk menampilkan matriks dan menghitung jumlah setiap baris/kolom. Estimasi Waktu: 60 menit.	Deklarasi, inisialisasi, dan akses elemen array 2D. Studi kasus: tabel nilai, matriks sederhana. Referensi: Deitel, P. & Deitel, H. (2017). C: How to Program (8th Edition). Pearson. Malik, D. S. (2018). Data Structures Using C++ (2nd Edition). Cengage Learning.	
10	Mahasiswa mampu memahami konsep fungsi dan prosedur, mendeklarasikan, memanggil, serta mengimplementasikan fungsi sederhana untuk menyusun program yang lebih terstruktur dan modular.	a. Mahasiswa dapat menjelaskan perbedaan fungsi dan prosedur. b. Mahasiswa mampu menuliskan program dengan fungsi sederhana (tanpa parameter maupun dengan parameter). c. Mahasiswa dapat mengelompokkan bagian program ke dalam fungsi agar lebih terstruktur.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Penilaian hasil praktikum, uji coba program, dan review kode.	Penjelasan konsep dasar fungsi dan prosedur. Demonstrasi program dengan fungsi sederhana (contoh: fungsi penjumlahan). Praktikum: membuat program kalkulator sederhana dengan beberapa fungsi (penjumlahan, pengurangan, perkalian, pembagian). Estimasi Waktu: 2 × 50 menit.	Materi tambahan berupa slides dan contoh kode fungsi. Diskusi online: manfaat penggunaan fungsi dalam program besar. Tugas individu: membuat program yang menggunakan minimal 3 fungsi berbeda. Estimasi Waktu: 60 menit.	Materi Pembelajaran [Pustaka] Konsep fungsi dan prosedur. Deklarasi, pemanggilan, dan parameter fungsi. Studi kasus: program kalkulator sederhana. Referensi: Deitel, P. & Deitel, H. (2017). C: How to Program (8th Edition). Pearson. Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language (2nd Edition). Prentice Hall.	2%
11	Mahasiswa mampu mengimplementasikan fungsi dengan nilai balik (return value) serta memahami penggunaan parameter by value dan by reference dalam pemrograman untuk menyelesaikan masalah yang lebih kompleks.	a. Mahasiswa dapat menjelaskan konsep fungsi dengan nilai balik. b. Mahasiswa mampu menulis program dengan fungsi yang mengembalikan nilai. c. Mahasiswa dapat menggunakan parameter by value dan by reference sesuai kebutuhan.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Penilaian hasil latihan praktikum, pengujian kode, dan review program.	Penjelasan konsep fungsi dengan nilai balik dan parameter. Demonstrasi program menggunakan fungsi untuk menghitung nilai faktorial atau bilangan prima. Praktikum: membuat program yang menggunakan fungsi dengan parameter untuk melakukan pengolahan data (contoh: mencari nilai maksimum/minimum	Materi tambahan berupa contoh kode fungsi dengan return value. Diskusi: perbedaan dan penggunaan parameter by value vs by reference. Tugas individu: membuat program dengan minimal 2 fungsi bernilai balik dan 1 fungsi dengan parameter by reference. Estimasi Waktu: 60 menit.	Materi Pembelajaran [Pustaka] Fungsi dengan nilai balik (return value). Parameter by value dan by reference. Studi kasus: fungsi pencarian nilai maksimum/minimum, faktorial, atau bilangan prima. Referensi: Deitel, P. & Deitel, H. (2017). C: How to Program (8th Edition). Pearson.	2%

				dari array). Estimasi Waktu: 2 × 50 menit.		Malik, D. S. (2018). Data Structures Using C++ (2nd Edition). Cengage Learning.	
12	Mahasiswa mampu memahami konsep rekursi dan mengimplementasikan annya dalam menyelesaikan permasalahan pemrograman, serta mampu membandingkan efisiensi antara solusi rekursif dan iteratif.	a. Mahasiswa dapat menjelaskan konsep dasar rekursi. b. Mahasiswa mampu menulis program rekursif sederhana. c. Mahasiswa dapat membandingkan solusi rekursif dan iteratif untuk kasus yang sama.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Penilaian hasil praktikum, uji coba program, dan review kode.	Penjelasan konsep rekursi dan cara kerjanya. Demonstrasi program rekursif sederhana (contoh: faktorial, deret Fibonacci). Praktikum: membuat program rekursif untuk menyelesaikan kasus tertentu (contoh: pencarian bilangan prima, perhitungan pangkat). Estimasi Waktu: 2 × 50 menit.	Materi tambahan berupa contoh kasus rekursi. Diskusi online: kelebihan dan kekurangan rekursi dibanding iterasi. Tugas individu: membuat program rekursif minimal dengan 2 kasus berbeda. Estimasi Waktu: 60 menit.	Materi Pembelajaran [Pustaka] Konsep rekursi. Studi kasus rekursi: faktorial, Fibonacci, pangkat bilangan. Perbandingan rekursi dan iterasi. Referensi: Deitel, P. & Deitel, H. (2017). C: How to Program (8th Edition). Pearson. Wirth, N. (2013). Algorithms and Data Structures. Springer.	2%
13	Mahasiswa mampu memahami konsep string (teks), melakukan deklarasi dan manipulasi string, serta mengimplementasikan operasi dasar string dalam program sederhana.	a. Mahasiswa dapat menjelaskan konsep string dalam pemrograman. b. Mahasiswa mampu menuliskan program untuk membaca, menampilkan, dan memproses string. c. Mahasiswa dapat mengimplementasikan operasi string sederhana (panjang string, penggabungan, pencarian kata).	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Penilaian hasil praktikum, observasi, serta review kode program.	Penjelasan konsep string dan perbedaan dengan array karakter. Demonstrasi program manipulasi string sederhana. Praktikum: membuat program menghitung panjang string, menggabungkan dua string, serta mencari substring. Estimasi Waktu: 2 × 50 menit.	Materi tambahan berupa contoh program manipulasi string. Diskusi online: contoh penerapan string dalam aplikasi nyata (contoh: login, validasi kata sandi, pencarian kata). Tugas individu: membuat program validasi sederhana, misalnya cek apakah kata yang diinput adalah palindrom. Estimasi Waktu: 60 menit.	Materi Pembelajaran [Pustaka] Konsep string dalam pemrograman. Operasi dasar string: panjang, gabung, pencarian. Studi kasus: validasi teks, pencarian kata, palindrom. Referensi: Deitel, P. & Deitel, H. (2017). C: How to Program (8th Edition). Pearson. Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language (2nd Edition). Prentice Hall.	6%
14	Mahasiswa mampu memahami konsep struktur data sederhana (struct), mendeklarasikan, menginisialisasi, dan mengakses data dalam struct untuk menyelesaikan	a. Mahasiswa dapat menjelaskan perbedaan antara array dan struct. b. Mahasiswa mampu menuliskan program dengan deklarasi dan penggunaan struct.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Penilaian hasil praktikum, uji coba program, dan review	Penjelasan konsep struct dan perbedaannya dengan array. Demonstrasi program dengan struct sederhana (contoh: menyimpan data mahasiswa berisi nama, NIM, dan nilai). Praktikum: membuat	Materi tambahan berupa contoh kode struct. Diskusi online: penerapan struct dalam kasus nyata (contoh: data logistik, produk, pelanggan). Tugas individu: membuat program dengan struct yang memuat minimal 3	Konsep struct dalam pemrograman. Deklarasi, inisialisasi, dan akses data struct. Studi kasus: pengelolaan data mahasiswa, produk, atau barang logistik. Referensi: Deitel, P. & Deitel, H.	5%

	kasus pemrograman yang melibatkan kumpulan data dengan berbagai tipe.	c. Mahasiswa dapat mengimplementasikan struct untuk menyimpan dan menampilkan data kompleks (misalnya data mahasiswa).	kode.	program untuk input-output data mahasiswa menggunakan struct. Estimasi Waktu: 2 × 50 menit.	field berbeda dan menampilkan hasilnya. Estimasi Waktu: 60 menit.	(2017). C: How to Program (8th Edition). Pearson. Malik, D. S. (2018). Data Structures Using C++ (2nd Edition). Cengage Learning.	
15	Mahasiswa mampu memahami konsep manajemen file (file handling), menulis program untuk membaca dan menyimpan data ke dalam file teks, serta mengolah data dari file sebagai bagian dari pemrosesan informasi.	a. Mahasiswa dapat menjelaskan konsep dasar input-output file. b. Mahasiswa mampu menuliskan program untuk membuka, membaca, dan menulis file. c. Mahasiswa dapat mengimplementasikan file handling untuk menyimpan dan menampilkan data sesuai kebutuhan.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Penilaian hasil praktikum, uji coba kode, dan review program.	Penjelasan konsep dasar file handling (buka file, baca, tulis, tutup). Demonstrasi program membaca dan menulis file sederhana. Praktikum: membuat program input data mahasiswa, menyimpan ke file, lalu membaca kembali isi file. Estimasi Waktu: 2 × 50 menit.	Materi tambahan berupa contoh kode file handling. Diskusi online: manfaat penggunaan file dalam aplikasi nyata (contoh: penyimpanan data logistik, data transaksi). Tugas individu: membuat program sederhana yang menyimpan minimal 5 data ke file lalu menampilkannya kembali. Estimasi Waktu: 60 menit.	Materi Pembelajaran [Pustaka] Konsep input-output file. Operasi dasar file: baca, tulis, simpan, tutup. Studi kasus: penyimpanan dan pembacaan data mahasiswa/produk. Referensi: Deitel, P. & Deitel, H. (2017). C: How to Program (8th Edition). Pearson. Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language (2nd Edition). Prentice Hall.	5%
16	UAS (Ujian Akhir Semester) Mahasiswa mampu mengintegrasikan seluruh materi yang telah dipelajari (struktur dasar program, percabangan, perulangan, array, fungsi, rekursi, string, struct, dan file handling) ke dalam sebuah program komprehensif yang dapat menyelesaikan permasalahan nyata secara mandiri.	a. Mahasiswa dapat merancang algoritma untuk menyelesaikan permasalahan yang diberikan. b. Mahasiswa mampu menuliskan program lengkap menggunakan berbagai konsep pemrograman. c. Mahasiswa dapat menguji dan memastikan program berjalan dengan output sesuai kebutuhan studi kasus.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Teknik Penilaian: Ujian praktik di laboratorium berbasis studi kasus, penilaian hasil kode program, dan uji coba langsung.	Pelaksanaan UAS berupa ujian praktik individu. Mahasiswa menyelesaikan studi kasus pemrograman dengan memanfaatkan konsep yang telah dipelajari selama semester. Estimasi Waktu: 2 × 50 menit.	UAS dapat dilaksanakan dengan bentuk take-home project terbatas waktu melalui LMS. Estimasi Waktu: sesuai ketentuan dosen pengampu.	Materi Pembelajaran [Pustaka] Seluruh materi pertemuan Minggu ke-1 s.d. Minggu ke-15. Referensi: Deitel, P. & Deitel, H. (2017). C: How to Program (8th Edition). Pearson. Malik, D. S. (2018). Data Structures Using C++ (2nd Edition). Cengage Learning. Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language (2nd Edition). Prentice Hall.	30%