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**
	ALGORITMA PEMROGRAMAN	4TLMKK218	Teknik	2	SKS	4	01/08/2025
Otoritas	**Pengembang RPS**			**Ketua Kelompok Keahlian**		**Ka PRODI**	
	TIM Microteaching			TIM Microteaching		(M. ABYAN)	
Deskripsi Mata Kuliah	Mata kuliah Algoritma Pemrograman membahas konsep dasar pemecahan masalah secara sistematis melalui perancangan algoritma dan penerapannya dalam bentuk program komputer. Mahasiswa akan mempelajari logika pemrograman, struktur data sederhana, alur kontrol, fungsi, serta teknik pemrograman terstruktur untuk menyelesaikan permasalahan di bidang teknik dan logistik. Setelah mengikuti mata kuliah ini, mahasiswa diharapkan mampu memahami konsep algoritma, menyusun solusi dalam bentuk pseudocode atau flowchart, serta mengimplementasikannya menggunakan bahasa pemrograman dasar. Mata kuliah ini juga menekankan pada keterampilan analitis, ketelitian, serta kemampuan berpikir logis yang menjadi fondasi bagi mata kuliah pemrograman lanjutan.						
Capaian Pembelajaran Lulusan & Capaian Pembelajaran Mata Kuliah	**Capaian Pembelajaran Lulusan (CPL) PRODI**						
	CPL03	Mampu menerapkan pengetahuan matematika, sains, dan rekayasa dalam bidang teknik logistik.					
	CPL04	Mampu mengidentifikasi, memformulasikan, dan menyelesaikan masalah rekayasa logistik dengan pendekatan algoritmik dan teknologi informasi.					
	Kode CPMK	**Capaian Pembelajaran Mata Kuliah**			**CPL yang Didukung**		
	CPMK114	Menjelaskan konsep dasar algoritma & pemrograman.			CPL03		
	CPMK115	Menyusun algoritma dalam bentuk pseudocode & flowchart.			CPL03		
	CPMK116	Mengimplementasikan percabangan & perulangan.			CPL03		

	CPMK117	Mengelola data menggunakan array & string.	CPL03, CPL04
	CPMK118	Merancang solusi terintegrasi berbasis studi kasus logistik.	CPL04

Penilaian

Id CPMK	Bobot per Bentuk Penilaian					TOTAL BOBOT PER CPMK (%)
	Tugas 1	Tugas 2	Tugas 3	Proyek 1	Proyek 2	
CPMK114	5	0	0	0	0	5
CPMK115	5	0	0	0	0	5
CPMK116	5	10	0	0	0	15
CPMK117	5	5	15	0	0	25
CPMK118	0	5	0	0	0	5
UTS	0	0	0	20	0	20
UAS	0	0	0	0	25	25
Total per penilaian	**20**	**20**	**15**	**20**	**25**	**100**

Pustaka

Utama:

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4th ed.). The MIT Press.
2. Malik, D. S. (2018). Data Structures and Algorithms in C++ (4th ed.). Cengage Learning.
3. Wirth, N. (2013). Algorithms and Data Structures. Springer.

Pustaka Pendukung:

1. Downey, A. (2021). Think Python: How to Think Like a Computer Scientist (2nd ed.). Green Tea Press.
2. Budd, T. (2020). An Introduction to Object-Oriented Programming (3rd ed.). Addison-Wesley.
3. Sebesta, R. W. (2019). Concepts of Programming Languages (12th ed.). Pearson.
4. Sedgewick, R., & Wayne, K. (2011). Algorithms (4th ed.). Addison-Wesley.

Media Pembelajaran	**Software:**	Hardware :
		Komputer/Laptop; Projector
Team Teaching	TIM Microteaching	
Matakuliah Syarat		
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 dan pemrograman, menjelaskan peran algoritma dalam pemecahan masalah, serta mengidentifikasi komponen utama dalam penyusunan algoritma.	a. Mahasiswa dapat menjelaskan definisi algoritma dan pemrograman. b. Mahasiswa mampu memberikan contoh sederhana permasalahan yang dapat diselesaikan dengan algoritma. c. Mahasiswa menunjukkan pemahaman awal melalui diskusi dan kuis singkat.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Kriteria: Tingkat ketepatan jawaban, keaktifan dalam diskusi, dan kemampuan mengaitkan konsep dengan contoh praktis. Teknik: Observasi partisipasi kelas, tanya jawab, dan evaluasi kuis singkat.	Kuliah tatap muka, diskusi kelas, dan latihan soal sederhana (2 × 50 menit) Membuat ringkasan tentang konsep algoritma dan memberikan contoh sederhana kasus logistik yang dapat diselesaikan dengan algoritma.	Penjelasan materi melalui Learning Management System (LMS), forum diskusi online, serta kuis formatif berbasis e-learning (1 × 50 menit) Membuat ringkasan tentang konsep algoritma dan memberikan contoh sederhana kasus logistik yang dapat diselesaikan dengan algoritma.	Materi Pembelajaran [Pustaka] : Konsep dasar algoritma dan pemrograman. Peran algoritma dalam pemecahan masalah. Struktur dasar algoritma: input, proses, output. Referensi: Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4th ed.). MIT Press. Downey, A. (2021). Think Python: How to Think Like a Computer Scientist (2nd ed.). Green Tea Press.	2%
2	Mahasiswa mampu menjelaskan notasi algoritma, menyusun algoritma sederhana dalam bentuk pseudocode dan flowchart, serta memahami perbedaan keduanya dalam merepresentasikan solusi masalah.	a. Mahasiswa dapat menuliskan pseudocode untuk permasalahan sederhana. b. Mahasiswa dapat menggambarkan flowchart sesuai dengan algoritma yang dibuat. c. Mahasiswa menunjukkan ketelitian dalam menyusun langkah logis.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Kriteria: Kesesuaian antara pseudocode dan flowchart, ketepatan alur logika, serta kerapihan penyajian. Teknik: Penilaian tugas individu, observasi dalam latihan kelas, dan evaluasi hasil pekerjaan tertulis.	Ceramah interaktif, latihan membuat pseudocode dan flowchart di kelas (2 × 50 menit). Membuat pseudocode dan flowchart untuk kasus sederhana (contoh: menghitung rata-rata nilai, menentukan bilangan ganjil/genap).	Latihan mandiri menggunakan software flowchart online (misalnya Lucidchart, Draw.io) dan diskusi via LMS (1 × 50 menit). Membuat pseudocode dan flowchart untuk kasus sederhana (contoh: menghitung rata-rata nilai, menentukan bilangan ganjil/genap).	Materi Pembelajaran [Pustaka] Pengenalan pseudocode. Pengenalan flowchart. Hubungan antara pseudocode dan flowchart dalam pemecahan masalah. Referensi: Malik, D. S. (2018). Data Structures and Algorithms in C++ (4th ed.). Cengage Learning. Downey, A. (2021). Think Python: How to Think Like a Computer Scientist (2nd ed.). Green Tea Press.	2%
3	Mahasiswa mampu memahami dan mengimplementasikan struktur kontrol dasar dalam pemrograman, khususnya percabangan (selection) menggunakan	a. Mahasiswa dapat menuliskan algoritma dengan percabangan sederhana. b. Mahasiswa mampu menggambarkan flowchart percabangan dengan benar.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Kriteria: Kebenaran logika percabangan, kesesuaian algoritma dengan flowchart, serta keberhasilan	Kuliah tatap muka, latihan soal percabangan, serta praktik pemrograman di laboratorium (2 × 50 menit). Membuat program sederhana dengan percabangan, misalnya menentukan nilai kelulusan, klasifikasi	Tutorial online melalui LMS dan latihan soal berbasis komputer (1 × 50 menit). Membuat program sederhana dengan percabangan, misalnya menentukan nilai kelulusan, klasifikasi angka ganjil/genap, atau	Materi Pembelajaran [Pustaka] Struktur kontrol percabangan (selection). Penggunaan if, if–else, dan nested if. Penerapan percabangan dalam kasus sederhana. Referensi: Malik, D. S. (2018). Data	2%

	perintah if, if–else, dan nested if dalam bentuk pseudocode, flowchart, maupun program sederhana.	c. Mahasiswa dapat membuat program sederhana menggunakan percabangan.	implementasi dalam program. Teknik: Penilaian tugas individu, observasi praktik di kelas/laboratorium, serta kuis singkat.	angka ganjil/genap, atau menentukan diskon pembelian.	menentukan diskon pembelian.	Structures and Algorithms in C++ (4th ed.). Cengage Learning. Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language (2nd ed.). Prentice Hall.	
4	Mahasiswa mampu memahami konsep perulangan (looping), menjelaskan perbedaan berbagai jenis perulangan (for, while, dan do-while), serta mengimplementasikannya dalam menyelesaikan masalah pemrograman sederhana.	a. Mahasiswa dapat menuliskan algoritma dengan perulangan menggunakan pseudocode. b. Mahasiswa mampu membuat flowchart perulangan dengan benar. c. Mahasiswa dapat membuat program sederhana menggunakan for, while, dan do-while.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Kriteria: Kebenaran alur logika perulangan, efisiensi algoritma, serta keberhasilan program dijalankan sesuai tujuan. Teknik: Penilaian melalui tugas pemrograman, latihan di kelas/laboratorium, dan kuis singkat.	Kuliah tatap muka, diskusi kelas, serta praktik langsung membuat program perulangan di laboratorium (2 × 50 menit). Membuat program sederhana dengan perulangan, misalnya menghitung jumlah bilangan, menampilkan deret angka, atau membuat tabel perkalian.	Penjelasan tambahan melalui LMS, forum diskusi online, serta latihan mandiri berbasis komputer (1 × 50 menit). Membuat program sederhana dengan perulangan, misalnya menghitung jumlah bilangan, menampilkan deret angka, atau membuat tabel perkalian.	Materi Pembelajaran [Pustaka] Konsep perulangan (looping). Struktur for, while, dan do-while. Studi kasus pemrograman dengan perulangan. Referensi: Downey, A. (2021). Think Python: How to Think Like a Computer Scientist (2nd ed.). Green Tea Press. Malik, D. S. (2018). Data Structures and Algorithms in C++ (4th ed.). Cengage Learning.	2%
5	Mahasiswa mampu memahami konsep array (larik), menjelaskan perbedaan antara array satu dimensi dan multidimensi, serta mengimplementasikan array dalam program sederhana untuk menyimpan dan mengolah data.	a. Mahasiswa dapat menuliskan algoritma yang menggunakan array. b. Mahasiswa mampu mendeklarasikan dan mengakses elemen array dengan benar. c. Mahasiswa dapat membuat program sederhana dengan array satu dimensi maupun dua dimensi.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Kriteria: Kebenaran deklarasi array, ketepatan logika dalam pengolahan data, serta hasil program sesuai dengan permasalahan yang diberikan. Teknik: Penilaian tugas individu, observasi praktik di laboratorium, dan evaluasi hasil program.	Kuliah tatap muka mengenai konsep array, diskusi kelas, serta praktik membuat program array di laboratorium (2 × 50 menit). Membuat program untuk menginput dan menampilkan data array (misalnya daftar nilai mahasiswa) serta program sederhana menggunakan array dua dimensi	Modul pembelajaran online melalui LMS, latihan soal mandiri, dan forum diskusi (1 × 50 menit). Membuat program untuk menginput dan menampilkan data array (misalnya daftar nilai mahasiswa) serta program sederhana menggunakan array dua dimensi.	Materi Pembelajaran [Pustaka] Konsep array satu dimensi dan dua dimensi. Deklarasi, inisialisasi, dan penggunaan array. Studi kasus pengolahan data menggunakan array. Referensi: Malik, D. S. (2018). Data Structures and Algorithms in C++ (4th ed.). Cengage Learning. Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language (2nd ed.). Prentice Hall.	2%
6	Mahasiswa mampu memahami konsep fungsi (function), menjelaskan	a. Mahasiswa dapat mendefinisikan fungsi dengan benar. b. Mahasiswa mampu	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75	Kuliah tatap muka mengenai teori fungsi, diskusi contoh program, dan praktik membuat	Materi tambahan melalui LMS, tutorial pemrograman berbasis video, serta latihan	Materi Pembelajaran [Pustaka] Konsep fungsi (function) dalam pemrograman.	2%

	manfaat modularisasi program, serta mengimplementasikan fungsi dalam program sederhana, termasuk penggunaan parameter dan nilai balik (return value).	menggunakan parameter dan nilai balik dalam fungsi. c. Mahasiswa dapat membuat program sederhana dengan membagi logika menjadi beberapa fungsi.	Kurang: 0–55 Kriteria: Kebenaran definisi fungsi, ketepatan alur logika program, serta keterbacaan kode. Teknik: Penilaian melalui tugas pemrograman, latihan di laboratorium, dan kuis singkat.	fungsi di laboratorium (2 × 50 menit). Membuat program sederhana dengan minimal dua fungsi, misalnya program menghitung luas bangun datar atau program konversi suhu.	soal mandiri (1 × 50 menit). Membuat program sederhana dengan minimal dua fungsi, misalnya program menghitung luas bangun datar atau program konversi suhu.	Parameter dan nilai balik (return value). Modularisasi program. Referensi: Malik, D. S. (2018). Data Structures and Algorithms in C++ (4th ed.). Cengage Learning. Downey, A. (2021). Think Python: How to Think Like a Computer Scientist (2nd ed.). Green Tea Press.	
7	Mahasiswa mampu memahami konsep rekursi, menjelaskan perbedaan antara rekursi dan iterasi, serta mengimplementasikan rekursi dalam program sederhana untuk menyelesaikan permasalahan logika maupun matematis.	a. Mahasiswa dapat menjelaskan prinsip kerja rekursi. b. Mahasiswa mampu menuliskan algoritma rekursif untuk kasus sederhana. c. Mahasiswa dapat membuat program rekursif, misalnya perhitungan faktorial atau deret Fibonacci.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Kriteria: Kebenaran konsep rekursi, ketepatan logika pemanggilan fungsi, serta hasil program sesuai dengan algoritma. Teknik: Penilaian melalui tugas pemrograman, praktik di laboratorium, dan kuis singkat.	Kuliah tatap muka mengenai teori rekursi, diskusi contoh kasus, serta praktik pemrograman rekursif di laboratorium (2 × 50 menit). Membuat program rekursif sederhana, misalnya menghitung faktorial, deret Fibonacci, atau mencari nilai maksimum dalam array.	Penjelasan tambahan melalui LMS, forum diskusi online, serta latihan mandiri dengan soal rekursi (1 × 50 menit). Membuat program rekursif sederhana, misalnya menghitung faktorial, deret Fibonacci, atau mencari nilai maksimum dalam array.	Materi Pembelajaran [Pustaka] Konsep rekursi. Perbedaan rekursi dan iterasi. Studi kasus pemrograman dengan rekursi. Referensi: Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4th ed.). MIT Press. Downey, A. (2021). Think Python: How to Think Like a Computer Scientist (2nd ed.). Green Tea Press.	2%
8	Ujian tengah semester (UTS) Mahasiswa mampu mengintegrasikan seluruh materi pertemuan minggu 1–7 (algoritma dasar, pseudocode, flowchart, percabangan, perulangan, array, fungsi, dan rekursi) serta menerapkannya dalam bentuk penyelesaian	a. Mahasiswa dapat menyelesaikan soal ujian yang mencakup teori dasar algoritma dan penerapan pemrograman. b. Mahasiswa mampu membuat algoritma dan program sesuai dengan permasalahan yang diberikan. c. Mahasiswa dapat menunjukkan keterampilan	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Kriteria: Tingkat pemahaman konsep, ketepatan algoritma, kebenaran logika program, dan hasil keluaran yang sesuai. Teknik: Ujian tertulis dan praktik pemrograman berbasis komputer.	Pelaksanaan ujian tengah semester secara tatap muka di kelas/laboratorium (durasi: 2 × 50 menit). Menyelesaikan soal ujian teori dan praktik pemrograman dalam waktu yang ditentukan	Ujian berbasis LMS atau aplikasi ujian online dengan pengawasan. Menyelesaikan soal ujian teori dan praktik pemrograman dalam waktu yang ditentukan	Materi Pembelajaran [Pustaka] Seluruh materi pertemuan minggu 1–7: Konsep dasar algoritma & pemrograman. Pseudocode dan flowchart. Struktur kontrol percabangan (if, if–else, nested if). Struktur perulangan (for, while, do-while). Array satu dimensi & dua dimensi. Fungsi dan modularisasi program. Rekursi.	20%

	masalah melalui program komputer sederhana.	analitis, logis, dan sistematis dalam penyelesaian soal.				Referensi: Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4th ed.). MIT Press. Malik, D. S. (2018). Data Structures and Algorithms in C++ (4th ed.). Cengage Learning.	
9	Mahasiswa mampu memahami konsep string, menjelaskan cara deklarasi dan manipulasi string, serta mengimplementasikan operasi dasar string dalam program sederhana..	a. Mahasiswa dapat mendeklarasikan dan menginisialisasi string dengan benar. b. Mahasiswa mampu menggunakan fungsi-fungsi dasar string (misalnya menghitung panjang, menggabungkan, atau membandingkan string). c. Mahasiswa dapat membuat program sederhana yang mengolah data berbasis string.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Kriteria: Kebenaran logika penggunaan string, ketepatan hasil program, dan kerapihan penulisan kode. Teknik: Penilaian melalui latihan praktikum, tugas pemrograman individu, dan kuis singkat.	Kuliah tatap muka, diskusi contoh kasus pengolahan string, dan praktik membuat program string di laboratorium (2 × 50 menit). Membuat program sederhana, misalnya menghitung jumlah huruf vokal dalam sebuah kata, menggabungkan dua string, atau membandingkan dua kata.	Tutorial string melalui LMS, forum diskusi online, dan latihan mandiri dengan soal berbasis string (1 × 50 menit). Membuat program sederhana, misalnya menghitung jumlah huruf vokal dalam sebuah kata, menggabungkan dua string, atau membandingkan dua kata.	Materi Pembelajaran [Pustaka] Konsep string dalam pemrograman. Operasi dasar string (deklarasi, panjang string, penggabungan, perbandingan). Studi kasus pemrograman menggunakan string. Referensi: Downey, A. (2021). Think Python: How to Think Like a Computer Scientist (2nd ed.). Green Tea Press. Malik, D. S. (2018). Data Structures and Algorithms in C++ (4th ed.). Cengage Learning.	2%
10	Mahasiswa mampu memahami konsep pengolahan file dalam pemrograman, menjelaskan perbedaan file teks dan file biner, serta mengimplementasikan operasi dasar file (membaca, menulis, dan menutup file) dalam program sederhana.	a. Mahasiswa dapat membuka, membaca, dan menulis file dengan benar. b. Mahasiswa mampu mengimplementasikan program sederhana yang menyimpan dan menampilkan data dari file. c. Mahasiswa dapat mengidentifikasi kesalahan umum dalam pengolahan file dan memperbaikinya.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Kriteria: Kebenaran sintaks penggunaan file, ketepatan logika alur program, serta hasil keluaran sesuai dengan data pada file. Teknik: Penilaian tugas pemrograman individu, praktik di laboratorium, dan kuis singkat.	Kuliah tatap muka mengenai teori file, diskusi contoh kasus, serta praktik membuat program pengolahan file di laboratorium (2 × 50 menit). Membuat program sederhana, misalnya mencatat data mahasiswa ke dalam file teks dan menampilkannya kembali di layar.	Modul pembelajaran online di LMS, latihan soal mandiri, dan forum diskusi (1 × 50 menit). Membuat program sederhana, misalnya mencatat data mahasiswa ke dalam file teks dan menampilkannya kembali di layar.	Materi Pembelajaran [Pustaka] Konsep file dalam pemrograman (file teks vs file biner). Operasi dasar file: open, read, write, close. Studi kasus pengolahan file sederhana. Referensi: Malik, D. S. (2018). Data Structures and Algorithms in C++ (4th ed.). Cengage Learning. Downey, A. (2021). Think Python: How to Think Like	2%

						a Computer Scientist (2nd ed.). Green Tea Press.	
11	Mahasiswa mampu memahami konsep struktur data record/struct, menjelaskan perbedaan antara array dan struct, serta mengimplementasikan struct dalam program sederhana untuk menyimpan dan mengolah data yang lebih kompleks.	a. Mahasiswa dapat mendefinisikan struct dengan benar. b. Mahasiswa mampu mengakses dan memodifikasi elemen struct. c. Mahasiswa dapat membuat program sederhana menggunakan struct untuk menyimpan data majemuk (misalnya data mahasiswa, barang, atau produk logistik).	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Kriteria: Kebenaran deklarasi struct, ketepatan logika dalam pengolahan data, dan keberhasilan program dijalankan sesuai tujuan. Teknik: Penilaian tugas pemrograman individu, observasi praktik di laboratorium, dan kuis singkat.	Kuliah tatap muka mengenai teori struct, diskusi contoh kasus, serta praktik membuat program dengan struct di laboratorium (2 × 50 menit). Membuat program dengan struct untuk menyimpan data barang logistik (kode barang, nama, jumlah, harga) dan menampilkannya dalam bentuk tabel sederhana.	Modul pembelajaran online di LMS, latihan soal mandiri, dan forum diskusi (1 × 50 menit). Membuat program dengan struct untuk menyimpan data barang logistik (kode barang, nama, jumlah, harga) dan menampilkannya dalam bentuk tabel sederhana.	Materi Pembelajaran [Pustaka] Konsep struct dalam pemrograman. Perbedaan array dengan struct. Studi kasus struct dalam pengolahan data kompleks. Referensi: Malik, D. S. (2018). Data Structures and Algorithms in C++ (4th ed.). Cengage Learning. Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language (2nd ed.). Prentice Hall.	2%
12	Mahasiswa mampu memahami konsep pointer dalam pemrograman, menjelaskan hubungan pointer dengan memori, serta mengimplementasikan pointer dalam program sederhana, termasuk penggunaannya dengan array dan fungsi.	a. Mahasiswa dapat mendeklarasikan dan menggunakan pointer dengan benar. b. Mahasiswa mampu menjelaskan cara kerja alamat memori dan dereferensi pointer. c. Mahasiswa dapat membuat program sederhana menggunakan pointer, misalnya mengakses array melalui pointer atau mengirim parameter ke fungsi menggunakan pointer.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Kriteria: Kebenaran sintaks pointer, pemahaman konsep alamat memori, serta ketepatan hasil program. Teknik: Penilaian tugas pemrograman, latihan di laboratorium, serta kuis singkat berbasis praktik.	Kuliah tatap muka mengenai teori pointer, demonstrasi contoh program, serta praktik penggunaan pointer di laboratorium (2 × 50 menit). Membuat program sederhana untuk mengakses array dengan pointer dan membuat fungsi pertukaran nilai (swap) menggunakan pointer.	Modul pembelajaran pointer di LMS, forum diskusi, dan latihan soal mandiri (1 × 50 menit). Membuat program sederhana untuk mengakses array dengan pointer dan membuat fungsi pertukaran nilai (swap) menggunakan pointer.	Materi Pembelajaran [Pustaka] Konsep pointer dan alamat memori. Pointer dan array. Pointer sebagai parameter fungsi. Referensi: Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language (2nd ed.). Prentice Hall. Malik, D. S. (2018). Data Structures and Algorithms in C++ (4th ed.). Cengage Learning.	2%
13	Mahasiswa mampu memahami konsep dasar algoritma pengurutan (sorting) dan	a. Mahasiswa dapat menjelaskan perbedaan dan kegunaan algoritma sorting dan	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55	Kuliah tatap muka mengenai teori sorting dan searching, diskusi algoritma, serta praktik pemrograman di	Materi tambahan melalui LMS, latihan mandiri berupa implementasi algoritma sorting/searching, dan	Materi Pembelajaran [Pustaka] Konsep algoritma sorting dan searching. Implementasi Bubble Sort,	6%

	pencarian (searching), menjelaskan prinsip kerja beberapa metode sederhana, serta mengimplementasikannya dalam bentuk program.	searching. b. Mahasiswa mampu menuliskan algoritma Bubble Sort, Selection Sort, dan Linear Search. c. Mahasiswa dapat membuat program sederhana untuk mengurutkan data dan melakukan pencarian data.	Kriteria: Kebenaran logika algoritma, efisiensi proses, serta hasil keluaran program sesuai dengan data yang diolah. Teknik: Penilaian tugas pemrograman, praktik laboratorium, dan kuis singkat.	laboratorium (2 × 50 menit). a. Membuat program sederhana untuk mengurutkan data mahasiswa berdasarkan nilai (Bubble Sort atau Selection Sort). b. Membuat program pencarian data menggunakan metode Linear Search.	forum diskusi (1 × 50 menit). a. Membuat program sederhana untuk mengurutkan data mahasiswa berdasarkan nilai (Bubble Sort atau Selection Sort). b. Membuat program pencarian data menggunakan metode Linear Search.	Selection Sort, dan Linear Search. Studi kasus pengurutan dan pencarian data. Referensi: Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4th ed.). MIT Press. Malik, D. S. (2018). Data Structures and Algorithms in C++ (4th ed.). Cengage Learning..	
14	Mahasiswa mampu memahami dan mengimplementasikan algoritma pencarian lanjutan (Binary Search), menjelaskan perbedaan dengan Linear Search, serta mengenal konsep dasar analisis kompleksitas algoritma (time complexity dan space complexity) pada kasus sederhana.	a. Mahasiswa dapat menjelaskan prinsip kerja Binary Search. b. Mahasiswa mampu menuliskan algoritma Binary Search dengan benar. c. Mahasiswa dapat membuat program sederhana menggunakan Binary Search. d. Mahasiswa menunjukkan pemahaman awal tentang analisis kompleksitas algoritma.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Kriteria: Kebenaran algoritma Binary Search, ketepatan hasil pencarian, serta kemampuan menjelaskan kompleksitas waktu (time complexity) secara sederhana. Teknik: Penilaian tugas pemrograman, praktik laboratorium, dan tanya jawab/diskusi kelas.	Kuliah tatap muka mengenai Binary Search dan analisis kompleksitas, diskusi kasus, serta praktik pemrograman di laboratorium (2 × 50 menit). a. Membuat program pencarian data dengan Binary Search. b. Membandingkan hasil dan waktu eksekusi antara Linear Search dan Binary Search untuk dataset kecil.	Modul pembelajaran melalui LMS, forum diskusi, serta latihan mandiri implementasi Binary Search (1 × 50 menit). a. Membuat program pencarian data dengan Binary Search. b. Membandingkan hasil dan waktu eksekusi antara Linear Search dan Binary Search untuk dataset kecil.	Materi Pembelajaran [Pustaka] Algoritma Binary Search. Perbedaan Linear Search dan Binary Search. Pengenalan analisis kompleksitas algoritma (Big-O Notation). Referensi: Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4th ed.). MIT Press. Sedgewick, R., & Wayne, K. (2011). Algorithms (4th ed.). Addison-Wesley.	5%
15	Mahasiswa mampu mengintegrasikan seluruh konsep algoritma dan pemrograman yang telah dipelajari (pseudocode, flowchart, percabangan, perulangan, array, fungsi, pointer, rekursi, sorting,	a. Mahasiswa dapat merancang algoritma penyelesaian masalah kompleks. b. Mahasiswa mampu mengimplementasikan solusi dalam bentuk program terstruktur. c. Mahasiswa dapat mendokumentasikan program dengan	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Kriteria: Kesesuaian solusi dengan permasalahan, ketepatan logika algoritma, keberhasilan program dijalankan, serta kualitas	Bimbingan proyek, presentasi kelompok, dan diskusi hasil implementasi di kelas/laboratorium (2 × 50 menit). Mengembangkan program aplikasi sederhana berbasis studi kasus: a. Sistem pencatatan barang masuk/keluar gudang. b. Aplikasi sederhana	Upload program dan laporan proyek melalui LMS, forum diskusi antar kelompok (1 × 50 menit). Mengembangkan program aplikasi sederhana berbasis studi kasus: a. Sistem pencatatan barang masuk/keluar gudang. b. Aplikasi sederhana perhitungan rute	Materi Pembelajaran [Pustaka] Integrasi seluruh materi algoritma dan pemrograman (minggu 1–14). Studi kasus implementasi algoritma dalam dunia nyata/logistik. Referensi: Malik, D. S. (2018). Data Structures and Algorithms	5%

	searching) dalam bentuk studi kasus atau proyek akhir yang relevan dengan bidang teknik logistik maupun permasalahan sehari-hari.	baik (komentar, struktur kode, dan laporan singkat). d. Mahasiswa menunjukkan kemampuan bekerja mandiri maupun berkelompok dalam menyelesaikan proyek kecil.	dokumentasi. Teknik: Penilaian tugas proyek (program dan laporan), presentasi singkat hasil proyek, serta diskusi kelas.	perhitungan rute distribusi. c. Program pengolahan nilai mahasiswa dengan fitur sorting dan searching.	distribusi. c. Program pengolahan nilai mahasiswa dengan fitur sorting dan searching.	in C++ (4th ed.). Cengage Learning. Downey, A. (2021). Think Python: How to Think Like a Computer Scientist (2nd ed.). Green Tea Press.	
16	UAS (Ujian Akhir Semester) Mahasiswa mampu mengevaluasi dan mengintegrasikan seluruh materi pembelajaran (minggu 1–15) untuk menyelesaikan permasalahan kompleks dalam pemrograman, baik secara teoritis maupun praktis, serta menunjukkan keterampilan berpikir logis, analitis, dan sistematis.	a. Mahasiswa dapat menyelesaikan soal teori mengenai konsep algoritma dan struktur data dasar. b. Mahasiswa mampu merancang algoritma dengan pseudocode atau flowchart untuk kasus tertentu. c. Mahasiswa dapat mengimplementasikan solusi dalam bentuk program yang berjalan dengan benar. d. Mahasiswa mampu menganalisis efisiensi algoritma sederhana.	Kriteria Penilaian: Istimewa: 90–100 Sangat Baik: 76–89 Cukup: 56–75 Kurang: 0–55 Kriteria: Tingkat pemahaman konsep, kebenaran logika algoritma, hasil program sesuai kebutuhan, serta kualitas analisis. Teknik: Ujian tertulis (teori dan analisis algoritma) serta ujian praktik pemrograman di laboratorium.	Ujian Akhir Semester secara tatap muka di kelas/laboratorium dengan durasi 2 × 50 menit. Menyelesaikan soal teori dan praktik pemrograman sesuai instruksi soal UAS.	Ujian berbasis LMS atau aplikasi ujian online dengan pengawasan. Menyelesaikan soal teori dan praktik pemrograman sesuai instruksi soal UAS.	Materi Pembelajaran [Pustaka] Seluruh materi pertemuan minggu 1–15: Algoritma dasar, pseudocode, flowchart. Percabangan, perulangan, array, string, struct, pointer. Fungsi dan rekursi. Sorting dan searching. Studi kasus integratif. Referensi: Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4th ed.). MIT Press. Malik, D. S. (2018). Data Structures and Algorithms in C++ (4th ed.). Cengage Learning.	30%