

Rencana Pembelajaran Semester (RPS)

		Sekolah Tinggi Manajemen Informatika dan Komputer Jakarta STI&K MANAJEMEN INFORMATIKA				Kode Dokumen
RENCANA PEMBELAJARAN SEMESTER						
MATA KULIAH (MK)		KODE	Rumpun MK		BOBOT (sks)	
Algoritma dan Pemrograman B		DK-21402			T = 2	P = 0
OTORISASI		Pengembang RPS		Koordinator RMK		Ketua PRODI
		Yudi Irawan Chandra, SKom., MMSI				Dr. Hariyanto
Capaian Pembelajaran (CP)	CPL-PRODI yang dibebankan pada MK					
	1. Sikap					
	CPL1.A	1. Beriman dan bertaqwa kepada Tuhan Yang Maha Esa.				
	2. Keterampilan Umum					
	CPL2.A	1. Mampu menunjukkan kinerja mandiri, bermutu dan terukur				
	CPL2.B	2. Mampu mengkaji penerapan ilmu pengetahuan dan teknologi dalam bidang teknologi informasi untuk menghasilkan prototype, prosedur baku dan atau perancangan serta menyusun hasil kajiannya dalam bentuk kertas kerja, makalah, poster dan lain sebagainya.				
	CPL2.C	3. Mampu mengelola pembelajaran secara mandiri 4.				
	3. Keterampilan Khusus					
	CPL3.A	1. Mampu menerapkan matematika, dan prinsip rekayasa ke dalam prosedur, proses, sistem, atau metodologi terapan bidang rekayasa perangkat lunak				
	CPL3.B	2. Mampu mengidentifikasi, memformulasikan, melakukan penelusuran referensi atau standar, menganalisis dan menyelesaikan masalah rekayasa perangkat lunak menggunakan perangkat analisa pada bidang teknologi informasi.				
	CPL3.C	3. Mampu menggunakan teknologi modern dalam melaksanakan pekerjaan di bidang teknologi rekayasa perangkat lunak				
	4. Pengetahuan					
	CPL4.A	1. Menguasai konsep teoritis sains alam, aplikasi matematika teknik; prinsip-prinsip rekayasa (engineering principles), sains rekayasa dan perancangan rekayasa yang diperlukan untuk analisis dan perancangan sistem, proses, produk atau komponen pada rekayasa perangkat lunak				

CPL4.B	2. Menguasai konsep prinsip-prinsip rekayasa (engineering principles), sains rekayasa dan perancangan rekayasa yang diperlukan untuk analisis dan perancangan perangkat lunak
CPL4.C	3. Menguasai prinsip dan teknik perancangan perangkat lunak menggunakan teknologi pada tataran praktikal (practical area)
CPL4.D	4. Menguasai konsep teoritis teknologi rekayasa yang diperlukan pada rekayasa perangkat lunak
CPL4.E	5. Menguasai pengetahuan tentang codes dan standard yang berlaku untuk penyelesaian masalah rekayasa perangkat lunak
Capaian Pembelajaran Mata Kuliah (CPMK)	
CPMK1	Mahasiswa mampu memahami jenis-jenis bahasa pemrograman dan siklus pengembangan program komputer.
CPMK2	Mahasiswa mampu memahami algoritma dan pemodelan matematik
CPMK3	Mahasiswa mampu melakukan pengujian program dan penelusuran error
CPMK 4	Mahasiswa mampu membuat program komputer sederhana
Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	
Sub-CPMK1	Mahasiswa mampu menerapkan algoritma dan flowchart dalam pemecahan masalah
Sub-CPMK2	Mahasiswa mampu menjelaskan struktur penulisan bahasa pemrograman C
Sub-CPMK3	Mahasiswa mampu menerapkan fungsi input dan output dalam pembuatan program
Sub-CPMK4	Mahasiswa mampu membuat program dengan konsep percabangan
Sub-CPMK5	Mahasiswa mampu membuat program dengan konsep percabangan lanjut
Sub-CPMK6	Mahasiswa mampu membuat program dengan konsep perulangan
Sub-CPMK7	Mahasiswa mampu membuat program dengan konsep perulangan lanjut
Sub-CPMK8	Mahasiswa mampu membuat program dengan konsep array
Sub-CPMK9	Mahasiswa mampu membuat program dengan konsep string
Sub-CPMK10	Mahasiswa mampu menggunakan fungsi dalam pembuatan program

		4 Liberty, J., Rao, S., Jones, B. 2008.Sams teach yourself C++ in one hour a day. Sams.					
Dosen Pengampu		Yudi Irawan Chandra, SKom., MMSI Saepul Lukman, SKom., MMSI.					
Matakuliah syarat		-					
Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
1	Mahasiswa mampu menerapkan algoritma dan flowchart dalam pemecahan masalah	1 Menjelaskan konsep dasar algoritma 2 Mengidentifikasi notasi-notasi flowchart 3 Menerapkan algoritma dan flowchart ke dalam suatu penyelesaian masalah	Pendekatan: Saintifik Model: Pembelajaran berbasis masalah Metode: Diskusi, Presentasi, Praktikum 4 X 50	1 Mahasiswa dapat mengidentifikasi masalah yang diberikan dan merumuskan algoritma yang tepat untuk menyelesaikannya. 2 Mahasiswa mampu menggambarkan algoritma dalam bentuk flowchart yang jelas dan mudah dipahami. 3 Mahasiswa dapat menerjemahkan algoritma yang telah dirancang ke dalam kode pemrograman menggunakan bahasa C++. 4 Mahasiswa mampu menggunakan flowchart sebagai alat bantu visual dalam mengidentifikasi kesalahan atau debug program.	Penugasan, Kuis dan Laporan Hasil Praktik	5%	1, 2, 3

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
2	Mahasiswa mampu menjelaskan struktur penulisan bahasa pemrograman C++	1 Mengidentifikasi jenis-jenis tipe data 2 Menjelaskan aturan pendefinisian identifier 3 Mengidentifikasi perbedaan variabel dan konstanta 4 Mengidentifikasi jenis-jenis operator 5 Menjelaskan prioritas operator aritmatika	Pendekatan: Sainifik Model: Pembelajaran berbasis masalah Metode: Diskusi, Presentasi, Praktikum 4 X 50	1.Memahami sintaks dasar bahasa C++ termasuk deklarasi variabel, tipe data, dan operator. 2.Mampu menerapkan struktur penulisan yang baik dan benar dalam bahasa C++ untuk meningkatkan keterbacaan dan pemeliharaan kode program. 3.Mahasiswa mampu mengimplementasikan konsep-konsep tersebut dalam program sederhana dan dapat menjelaskan bagaimana struktur penulisan dan tipe data yang digunakan dalam program tersebut	Penugasan, Kuis dan Laporan Hasil Praktik	5%	1, 2, 3
3	Mahasiswa mampu menerapkan fungsi input dan output dalam pembuatan program	1 Mengidentifikasi jenis-jenis fungsi input dan output 2 Menerapkan fungsi input dan output dalam program	Pendekatan: Sainifik Model: Pembelajaran berbasis masalah Metode: Diskusi, Presentasi, Praktikum 4 X 50	1 Kemampuan menggunakan fungsi input standar, seperti cin, untuk mengambil input dari pengguna. 2 Kemampuan menggunakan fungsi output standar, seperti cout, untuk menampilkan output kepada pengguna.	Penugasan, Kuis dan Laporan Hasil Praktik	5%	1, 2, 3

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
		1		3 Mahasiswa dapat mendemonstrasikan penggunaan operator aliran (<< dan >>) untuk melakukan input dan output pada tipe data dasar seperti integer, float, double, char, dan string. 4 Mahasiswa dapat menggunakan fungsi-fungsi bawaan dari pustaka C++ (misalnya getline() untuk membaca baris teks) untuk mengelola input dan output yang lebih kompleks.			
4	Mahasiswa mampu membuat program dengan konsep percabangan	2 Mengidentifikasi perbedaan kondisi dan aksi 3 Menjelaskan percabangan tunggal, majemuk dan bertingkat 4 Menjelaskan percabangan menggunakan selection case 5 Menerapkan konsep percabangan ke dalam program	Pendekatan: Saintifik Model: Pembelajaran berbasis masalah Metode: Diskusi, Presentasi, Praktikum 4 X 50	1 Mahasiswa dapat membuat program dengan menggunakan pernyataan kondisional seperti if, else if, dan else untuk mengatur alur program berdasarkan kondisi-kondisi tertentu. 2 Mahasiswa dapat memahami dan mengimplementasikan struktur percabangan	Penugasan, Kuis dan Laporan Hasil Praktik	10%	1, 2, 3

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
		6		bersarang, di mana pernyataan kondisional terdapat di dalam pernyataan kondisional lainnya, untuk menangani skenario yang lebih kompleks. 3 Mahasiswa dapat menggunakan operator logika (&&,) dan operator perbandingan (==, !=, <, >, <=, >=) secara tepat dalam kondisi-kondisi percabangan untuk menghasilkan hasil yang diharapkan.			
5	Mahasiswa mampu membuat program dengan konsep percabangan lanjut	1 Mengidentifikasi perbedaan kondisi dan aksi 2 Menjelaskan percabangan tunggal, majemuk dan bertingkat 3 Menjelaskan percabangan menggunakan selection case 4 Menerapkan konsep percabangan ke dalam program	Pendekatan: Saintifik Model: Pembelajaran berbasis masalah Metode: Diskusi, Presentasi, Praktikum 4 X 50	1 Mahasiswa dapat membuat program dengan percabangan yang melibatkan lebih dari dua kondisi, seperti menggunakan pernyataan switch-case untuk memilih tindakan berdasarkan berbagai nilai variabel. 2 Mahasiswa dapat melakukan pengujian terhadap program yang mereka buat dengan	Penugasan, Kuis dan Laporan Hasil Praktik	5%	1, 2, 3

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
		5		3 percabangan, termasuk mengidentifikasi dan memperbaiki kesalahan (debugging) yang mungkin muncul dalam implementasi percabangan tersebut. 4 Kemampuan dalam menangani percabangan dengan jumlah kondisi yang kompleks, serta merancang solusi yang mudah dimengerti dan dikelola. Mampu menggunakan percabangan untuk mengontrol aliran program secara efektif, memastikan program dapat menangani berbagai skenario dengan benar.			
6	Mahasiswa mampu membuat program dengan konsep perulangan	1 Mengidentifikasi jenis-jenis perulangan 2 Menjelaskan struktur perulangan 3 Menerapkan konsep perulangan ke dalam program	Pendekatan: Saintifik Model: Pembelajaran berbasis masalah Metode: Diskusi, Presentasi, Praktikum 4 X 50	1 Mampu menggunakan struktur perulangan seperti for, while, dan do-while dengan benar dan efisien dalam pembuatan program. 2 Memahami konsep increment dan decrement untuk	Penugasan, Kuis dan Laporan Hasil Praktik	10%	1, 2, 3

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
		4		mengendalikan iterasi perulangan. 3 Mampu menerapkan perulangan bersarang (nested loops) untuk menangani kasus yang kompleks. 4 Kemampuan dalam memahami dan mengimplementasikan algoritma yang memerlukan penggunaan perulangan, seperti pencarian, pengurutan, atau pemrosesan data dalam jumlah besar. Mampu menghindari perulangan tak terbatas (infinite loops) dan menangani situasi di mana perlu dilakukan keluar dari perulangan.			
7	Mahasiswa mampu membuat program dengan konsep perulangan lanjut	1 Mengidentifikasi jenis-jenis perulangan 2 Menjelaskan struktur Perulangan . 3 Menerapkan konsep perulangan ke dalam program	Pendekatan: Saintifik Model: Pembelajaran berbasis masalah Metode: Diskusi, Presentasi, Praktikum 4 X 50	1 Kemampuan untuk merancang program dengan menggunakan teknik perulangan yang efisien, seperti penggunaan perulangan berdasarkan kondisi (conditional loops).	Penugasan, Kuis dan Laporan Hasil Praktik	5%	1, 2, 3

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
		4		2 Mampu menerapkan kontrol aliran di dalam perulangan, seperti break dan continue, untuk mengontrol aliran eksekusi program. 3 Memahami konsep penggunaan indeks dalam perulangan untuk mengakses elemen-elemen dalam array atau struktur data. 4 Kemampuan dalam memahami dan mengimplementasikan perulangan rekursif untuk menyelesaikan masalah yang memerlukan pendekatan rekursif. Mampu menganalisis kompleksitas waktu (time complexity) dari program yang menggunakan perulangan untuk mengevaluasi efisiensi kinerja program			

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
8	Evaluasi Tengah Semester / Ujian Tengah Semester						
9	Mahasiswa mampu membuat program dengan konsep array	1 Menjelaskan definisi array 2 Mengidentifikasi jenis-jenis array 3 Menjelaskan cara mendeklarasikan masing-masing array 4 Menerapkan array dalam program	Pendekatan: Saintifik Model: Pembelajaran berbasis masalah Metode: Diskusi, Presentasi, Praktikum 4 X 50	1. Kemampuan untuk mendeklarasikan, menginisialisasi, dan mengakses elemen-elemen array dengan benar sesuai dengan aturan sintaksis bahasa pemrograman C++. 2. Keterampilan dalam memanipulasi data dalam array, seperti pencarian, pengurutan, atau operasi lainnya. 3. Kemampuan untuk menggunakan konsep indeks array dengan tepat untuk mengakses dan memanipulasi nilai-nilai array dalam berbagai konteks pemrograman. 4. Pemahaman tentang penggunaan array dalam konteks aplikasi nyata, seperti pengolahan data, pemodelan masalah, atau implementasi struktur data yang lebih kompleks.	Penugasan, Kuis dan Laporan Hasil Praktik	5%	1, 2, 3

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
		5		5. Kemampuan untuk menerapkan teknik-teknik pemrograman defensif untuk menghindari kesalahan terkait memori, seperti buffer overflow atau segfault, yang sering terjadi dalam manipulasi array.			
10	Mahasiswa mampu membuat program dengan konsep string	1 Menjelaskan definisi string 2 Menjelaskan cara mendeklarasikan variabel string 3 Menjelaskan cara memasukkan dan menampilkan isi dari variabel string 4 Menjelaskan cara mengakses elemen string 5 Menerapkan string dalam program	Pendekatan: Saintifik Model: Pembelajaran berbasis masalah Metode: Diskusi, Presentasi, Praktikum 4 X 50	1. Kemampuan dalam mengimplementasikan operasi dasar pada string, seperti penggabungan (concatenation), pencarian karakter atau substring, dan manipulasi string lainnya menggunakan fungsi-fungsi bawaan dari bahasa C++. 2. Pemahaman yang baik tentang konsep array karakter (char array) dan penggunaan pointer dalam manipulasi string, termasuk penggunaan pointer untuk	Penugasan, Kuis dan Laporan Hasil Praktik	10%	1, 2, 3

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
		1		mengakses karakter-karakter individu dalam string. 3. Keahlian dalam menggunakan library string standar (STL) dari C++ untuk memanipulasi string, seperti penggunaan kelas string dan fungsi-fungsi yang tersedia dalam header <string>. 4. Kemampuan untuk merancang dan mengimplementasikan fungsi-fungsi kustom untuk operasi khusus pada string, seperti fungsi untuk mengonversi string menjadi huruf besar atau kecil, membalikkan string, atau memeriksa apakah sebuah string merupakan palindrom. 5. Kapasitas untuk menulis program-program yang efisien dan efektif dalam memanipulasi string, dengan mempertimbangkan			

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
		2		kinerja dan penggunaan memori, serta menerapkan praktik-praktik terbaik dalam pemrograman C++ untuk mengoptimalkan kualitas dan kinerja kode.			
11	Mahasiswa mampu menggunakan fungsi dalam pembuatan program	3 Menjelaskan konsep dasar fungsi 4 Menjelaskan cara mendeklarasikan fungsi 5 Menjelaskan cara memanggil fungsi 6 Menerapkan fungsi dalam program	Pendekatan: Sainifik Model: Pembelajaran berbasis masalah Metode: Diskusi, Presentasi, Praktikum 4 X 50	1. Mahasiswa mampu membuat fungsi dengan benar, termasuk menetapkan tipe data kembalian dan parameter yang sesuai. 2. Mahasiswa dapat menggunakan fungsi yang telah didefinisikan dalam program utama atau dalam fungsi-fungsi lainnya dengan memahami konsep pemanggilan dan transfer parameter. 3. Mahasiswa dapat membuat fungsi untuk melakukan berbagai tugas, mulai dari perhitungan matematika sederhana hingga pemrosesan data yang kompleks.	Penugasan, Kuis dan Laporan Hasil Praktik	10%	1, 2, 3

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
		7		4. Mahasiswa mampu menggunakan rekursi untuk menyelesaikan permasalahan yang dapat dipecahkan secara iteratif, serta memahami prinsip-prinsip dasar rekursi. 5. Mahasiswa dapat mengorganisir kode program ke dalam fungsi-fungsi yang berbeda untuk meningkatkan keterbacaan, kemudahan pemeliharaan, dan modularitas program.			
12	Mahasiswa mampu menggunakan fungsi rekursif dalam pembuatan program	1. Menjelaskan definisi fungsi rekursif 2. Menjelaskan cara mendeklarasikan fungsi rekursif 3. Menjelaskan cara memanggil fungsi rekursif 4. Mengidentifikasi persamaan dan perbedaan fungsi iteratif dan rekursif 6. Mengidentifikasi kelebihan dan kekurangan fungsi rekursif 7. Menerapkan fungsi rekursif dalam program	Pendekatan: Saintifik Model: Pembelajaran berbasis masalah Metode: Diskusi, Presentasi, Praktikum 4 X 50	1. Mahasiswa diharapkan mampu menunjukkan pemahaman yang mendalam tentang konsep rekursi dan dapat mengaplikasikannya secara efektif dalam pemecahan masalah pemrograman. 2. Mahasiswa harus mampu mengidentifikasi kasus dasar (base case) dan langkah rekursif dalam	Penugasan, Kuis dan Laporan Hasil Praktik	5%	1, 2, 3

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
				suatu masalah, serta dapat mengevaluasi dan mengoptimalkan kinerja program rekursif. 3. Mahasiswa diharapkan mampu mengaplikasikan rekursi dalam penyelesaian masalah seperti pencarian dan pengurutan data, pembangunan struktur data seperti pohon dan graf, serta dalam pemodelan dan analisis algoritma.			
13	Mahasiswa mampu menggunakan pointer dalam pembuatan program	1 Menjelaskan konsep dasar pointer 2 Menjelaskan cara menggunakan pointer 3 Menerapkan pointer dalam program	Pendekatan: Saintifik Model: Pembelajaran berbasis masalah Metode: Diskusi, Presentasi, Praktikum 4 X 50	1. Mahasiswa dapat menjelaskan secara tepat tentang konsep dasar pointer, termasuk pengertian pointer, cara mendeklarasikan, menginisialisasi, mengakses nilai yang disimpan di alamat yang ditunjuk, dan menggambarkan penggunaan pointer dalam pemrograman C++.	Penugasan, Kuis dan Laporan Hasil Praktik	5%	1, 2, 3

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
				2. Mahasiswa dapat menunjukkan kemampuan dalam menggunakan pointer dalam penulisan kode program C++, termasuk penggunaan pointer untuk manipulasi alamat memori, akses ke variabel, dan penggunaan pointer untuk membuat struktur data yang kompleks seperti linked lists, trees, atau dynamic memory allocation. 3. Mahasiswa mampu menerapkan konsep pointer dalam menyelesaikan masalah pemrograman yang lebih kompleks, termasuk penggunaan pointer untuk transfer data antar fungsi, dynamic memory management untuk menghindari memory leaks, atau peng-optimalan performa program dengan menggunakan			

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
				pointer. Mahasiswa juga dapat menunjukkan pemahaman tentang keamanan penggunaan pointer, seperti menghindari null pointer dereferencing atau dangling pointers.			
14	Mahasiswa mampu membuat program dengan konsep struktur	1 Menjelaskan konsep dasar struktur 2 Menjelaskan cara mendeklarasikan variabel struktur 3 Menerapkan variabel struktur dalam program	Pendekatan: Saintifik Model: Pembelajaran berbasis masalah Metode: Diskusi, Presentasi, Praktikum 4 X 50	1. Mahasiswa dapat menjelaskan konsep dasar dari struktur (struct) dalam bahasa pemrograman C++, termasuk bagaimana mendefinisikan struktur, mendeklarasikan variabel struktur, dan mengakses anggota struktur dengan menggunakan operator titik. 2. Mahasiswa dapat merancang dan mengimplementasikan program-program dalam bahasa C++ yang menggunakan struktur untuk merepresentasikan entitas atau objek dalam suatu sistem. Ini mencakup	Penugasan, Kuis dan Laporan Hasil Praktik	10%	1, 2, 3

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
				kemam-puan dalam men-desain struktur data yang tepat untuk menyimpan informasi yang diperlukan dan mengakses informasi tersebut dengan benar dalam program. 3. Mahasiswa mampu mengidentifikasi dan merancang solusi menggunakan struktur dalam bahasa C++ untuk menyelesaikan berbagai macam masalah pemrograman, seperti pemrosesan data, pengelolaan informasi, atau representasi objek dalam program. Mereka juga dapat mengintegrasikan struktur dengan konsep pemrograman lainnya, seperti penggunaan fungsi atau algoritma, untuk mencapai solusi yang efisien dan efektif.			

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
15	Mahasiswa mampu membuat program untuk operasi file	1 Mengidentifikasi perbedaan file teks dan file biner 2 Mengidentifikasi jenis-jenis operasi file dalam file teks dan file biner 3 Menerapkan operasi file dalam pembuatan program	Pendekatan: Saintifik Model: Pembelajaran berbasis masalah Metode: Diskusi, Presentasi, Praktikum 4 X 50	1. Kemampuan Mahasiswa dalam Membuat Program dengan Fungsi Bawaan (Built-in Functions) C++: Indikator ini mencakup pemahaman mahasiswa tentang fungsi-fungsi bawaan dalam bahasa C++ yang digunakan untuk operasi file, seperti fopen(), fclose(), fread(), fwrite(), fseek(), dan lainnya. Mahasiswa diharapkan mampu menggunakan fungsi-fungsi ini secara efektif dalam membuat program untuk membuka, menuliskan, membaca, dan menutup file. 2. Pemahaman Mahasiswa tentang Konsep Input dan Output File: Indikator ini mencakup kemampuan mahasiswa dalam memahami konsep input dan output file dalam bahasa C++, termasuk	Penugasan, Kuis dan Laporan Hasil Praktik	10%	1, 2, 3

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
		4		cara membaca dan menulis data ke dalam file teks dan biner. Mahasiswa diharapkan mampu mengimplementasikan proses input-output file secara benar dan efisien dalam program yang mereka buat. 3. Keterampilan Mahasiswa dalam Mengelola Error dan Eksepsi: Indikator ini mencakup kemampuan mahasiswa dalam mengelola error dan eksepsi yang terjadi saat melakukan operasi file. Mahasiswa diharapkan mampu menggunakan mekanisme exception handling atau pendekatan lainnya untuk menangani situasi-situasi yang mungkin menyebabkan kesalahan dalam operasi file, seperti file tidak ditemukan, gagal membuka file, atau.			

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Materi Pembelajaran	Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa	Penilaian		Bobot Penilaian (%)	Referensi Materi
				Indikator	Kriteria & Teknik		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
		5		kegagalan saat menulis ke file. Mahasiswa juga diharapkan mampu memberikan pesan error yang jelas dan informatif kepada pengguna dalam menangani situasi-situasi tersebut			
16	Evaluasi Akhir Semester / Ujian Akhir Semester						

Catatan :

- 1 **Capaian Pembelajaran Lulusan PRODI (CPL-PRODI)** adalah kemampuan yang dimiliki oleh setiap lulusan PRODI yang merupakan internalisasi dari sikap, penguasaan pengetahuan dan ketrampilan sesuai dengan jenjang prodinya yang diperoleh melalui proses pembelajaran.
- 2 **CPL yang dibebankan pada mata kuliah** adalah beberapa capaian pembelajaran lulusan program studi (CPL-PRODI) yang digunakan untuk pembentukan/pengembangan sebuah mata kuliah yang terdiri dari aspek sikap, ketrampilan umum, ketrampilan khusus dan pengetahuan.
- 3 **CP Mata kuliah (CPMK)** adalah kemampuan yang dijabarkan secara spesifik dari CPL yang dibebankan pada mata kuliah, dan bersifat spesifik terhadap bahan kajian atau materi pembelajaran mata kuliah tersebut.
- 4 **Sub-CP Mata kuliah (Sub-CPMK)** adalah kemampuan yang dijabarkan secara spesifik dari CPMK yang dapat diukur atau diamati dan merupakan kemampuan akhir yang direncanakan pada tiap tahap pembelajaran, dan bersifat spesifik terhadap materi pembelajaran mata kuliah tersebut.
- 5 **Indikator penilaian** kemampuan dalam proses maupun hasil belajar mahasiswa adalah pernyataan spesifik dan terukur yang mengidentifikasi kemampuan atau kinerja hasil belajar mahasiswa yang disertai bukti-bukti.
- 6 **Kriteria Penilaian** adalah patokan yang digunakan sebagai ukuran atau tolok ukur ketercapaian pembelajaran dalam penilaian berdasarkan indikator-indikator yang telah ditetapkan. Kriteria penilaian merupakan pedoman bagi penilai agar penilaian konsisten dan tidak bias. Kriteria dapat berupa kuantitatif ataupun kualitatif.

- 7 **Bentuk penilaian:** tes dan non-tes.
- 8 **Bentuk pembelajaran:** Kuliah, Responsi, Tutorial, Seminar atau yang setara, Praktikum, Praktik Studio, Praktik Bengkel, Praktik Lapangan, Penelitian, Pengabdian Kepada Masyarakat dan/atau bentuk pembelajaran lain yang setara.
- 9 **Metode Pembelajaran:** Small Group Discussion, Role-Play & Simulation, Discovery Learning, Self-Directed Learning, Cooperative Learning, Collaborative Learning, Contextual Learning, Project Based Learning, dan metode lainnya yg setara.
- 10 **Materi Pembelajaran** adalah rincian atau uraian dari bahan kajian yg dapat disajikan dalam bentuk beberapa pokok dan sub-pokok bahasan.
- 11 **Bobot penilaian** adalah prosentasi penilaian terhadap setiap pencapaian sub-CPMK yang besarnya proposional dengan tingkat kesulitan pencapaian sub-CPMK tsb., dan totalnya 100%.
- 12 TM=Tatap Muka, PT=Penugasan terstruktur, BM=Belajar mandiri.

Kode Dokumen

MATA KULIAH (MK)

Rumpun MK

SEMESTER

Tgl Penyusunan

DK-21402

P = 0

1

24 Desember 2021

2

Tugas ke: 1

1. Mengidentifikasi jenis-jenis tipe data
2. Menjelaskan aturan pendefinisian identifier
3. Mengidentifikasi perbedaan variabel dan konstanta
4. Mengidentifikasi jenis-jenis operator
5. Menjelaskan prioritas operator aritmatika

1. Obyek

Membuat Program dengan berbagai jenis tipe data, operator dan konstanta dalam Bahasa C++

2. Yang dilakukan

1. Membuat program dengan sintaks dasar bahasa C++ termasuk deklarasi variabel, tipe data, dan operator.
2. Membuat program dengan struktur penulisan yang baik dan benar dalam bahasa C++ untuk meningkatkan keterbacaan dan pemeliharaan kode program.
3. Membuat program sederhana dengan struktur penulisan dan tipe data

		4. Metode/Cara pengerjaan	[1] Dapat membuat program dengan diskusi kelompok / personal [2] Program dibuat menggunakan PC atau Laptop
		5. Deskripsi luaran tugas	Program dengan menggunakan variabel, tipe data, operator dan konstanta
	Kriteria Penilaian:		
		a. Kelengkapan	25 %
		b. Kebenaran	50 %
		a. Ketelitian	25 %
Minggu ke:	4	Tugas ke:	2
	Tujuan tugas:		
		1. Mengidentifikasi jenis-jenis fungsi input dan output 2. Menerapkan fungsi input dan output dalam program 3. Mengidentifikasi perbedaan kondisi dan aksi 4. Menjelaskan percabangan tunggal, majemuk dan bertingkat 5. Menjelaskan percabangan menggunakan selection case	
	Uraian tugas:		
		a. Obyek	Membuat Program Input dan Kondisi (percabangan)
		b. Yang dilakukan	1. Dapat menggunakan fungsi input standar, seperti cin, untuk mengambil input dari pengguna. 2. Dapat menggunakan fungsi output standar, seperti cout, untuk menampilkan output kepada pengguna.

			3. Dapat mendemonstrasikan penggunaan operator aliran (<< dan >>) untuk melakukan input dan output pada tipe data dasar seperti integer, float, double, char, dan string. 4. Dapat menggunakan fungsi-fungsi bawaan dari pustaka C++ (misalnya getline()) untuk membaca baris teks) untuk mengelola input dan output yang lebih kompleks. 5. Dapat membuat program dengan menggunakan pernyataan kondisional seperti if, else if, dan else untuk mengatur alur program berdasarkan kondisi-kondisi tertentu. 6. Dapat mengimplementasikan struktur percabangan bersarang, di mana pernyataan kondisional terdapat di dalam pernyataan kondisional lainnya, untuk menangani skenario yang lebih kompleks.
		c. Metode/Cara pengerjaan	1. Dapat membuat program dengan diskusi kelompok / personal 2. Program dibuat menggunakan PC atau Laptop
		d. Deskripsi luaran tugas	Program dengan menggunakan input, tampilan terformat dengan cout dan program dengan menggunakan kondisi
	Kriteria Penilaian:		
		a. Kelengkapan	25 %
		b. Kebenaran	50 %
		c. Ketelitian	25 %
Minggu ke:	6	Tugas ke:	3
	Tujuan tugas:		
		1. Mengidentifikasi jenis-jenis perulangan 2. Menjelaskan struktur perulangan 3. Menerapkan konsep perulangan ke dalam program	

	Uraian tugas:		
		a. Obyek	Membuat program dengan teknik perulangan
		b. Yang dilakukan	1. Dapat membuat program dengan struktur perulangan seperti for, while, dan do-while dengan benar dan efisien 2. Dapat membuat program dengan konsep increment dan decrement untuk mengendalikan iterasi perulangan. 3. Dapat membuat proogram perulangan bersarang (nested loops) untuk menangani kasus yang kompleks. 4. Dapat menghindari perulangan tak terbatas (infinite loop)
		c. Metode/Cara pengerjaan	1. Dapat membuat program dengan diskusi kelompok / personal 2. Program dibuat menggunakan PC atau Laptop
		d. Deskripsi luaran tugas	Program dengan menggunakan teknik perulangan
	Kriteria Penilaian:		
		a. Kelengkapan	25 %
		b. Kebenaran	50 %
		c. Ketelitian	25 %
Minggu ke:	9	Tugas ke:	4
	Tujuan tugas:		

		1. Menjelaskan definisi array 2. Mengidentifikasi jenis-jenis array 3. Menjelaskan cara mendeklarasikan masing-masing array 4. Menerapkan array dalam program
	Uraian tugas:	
	Obyek	Membuat program dengan Array 1 dimensi dan multidimensi
	Yang dilakukan	1. Dapat mendeklarasikan, menginisialisasi, dan mengakses elemen-elemen array dengan benar sesuai dengan aturan sintaksis bahasa pemrograman C++. 2. Dapat memanipulasi data dalam array, seperti pencarian, pengurutan, atau operasi lainnya. 3. Dapat menggunakan konsep indeks array dengan tepat untuk mengakses dan memanipulasi nilai-nilai array dalam berbagai konteks pemrograman. 4. Dapat menggunakan array dalam konteks aplikasi nyata, seperti pengolahan data, pemodelan masalah, atau implementasi struktur data yang lebih kompleks. 5. Dapat membuat program dengan teknik-teknik pemrograman defensif untuk menghindari kesalahan terkait memori, seperti buffer overflow atau segfault, yang sering terjadi dalam manipulasi array.
	Metode/Cara pengerjaan	1. Dapat membuat program dengan diskusi kelompok / personal 2. Program dibuat menggunakan PC atau Laptop
	Deskripsi luaran tugas	Program dengan Array 1 dimensi dan multidimensi
	Kriteria Penilaian:	
	a. Kelengkapan	25 %

		b. Kebenaran	50 %
		c. Ketelitian	25 %
Minggu ke:	11	Tugas ke:	5
	Tujuan tugas:		
		1. Menjelaskan cara mendeklarasikan variabel string 2. Menjelaskan cara memasukkan dan menampilkan isi dari variabel string 3. Menjelaskan cara mengakses elemen string dan menerapkan string dalam program 4. Menjelaskan konsep dasar fungsi dan cara mendeklarasikan fungsi 5. Menjelaskan cara memanggil fungsi dan menerapkan fungsi dalam program	
	Uraian tugas:		
		Obyek	Membuat program string dan fungsi
		Yang dilakukan	1. Dapat mengimplementasikan operasi dasar pada string, seperti penggabungan (concatenation), pencarian karakter atau substring, dan manipulasi string lainnya menggunakan fungsi-fungsi bawaan dari bahasa C++. 2. Dapat menggunakan library string standar (STL) dari C++ untuk memanipulasi string, seperti penggunaan kelas string dan fungsi-fungsi yang tersedia dalam header <string>. 3. Dapat merancang dan mengimplementasikan fungsi-fungsi kustom untuk operasi khusus pada string, seperti fungsi untuk mengonversi string menjadi huruf besar atau kecil, membalikkan string, atau memeriksa apakah sebuah string merupakan palindrom. 4. Dapat membuat fungsi dengan benar, termasuk menetapkan tipe data kembalian dan parameter yang sesuai. 5. Dapat menggunakan fungsi yang telah didefinisikan dalam program utama atau dalam fungsi-fungsi lainnya dengan memahami konsep pemanggilan dan transfer parameter.

			6. Dapat membuat fungsi untuk melakukan berbagai tugas, mulai dari perhitungan matematika sederhana hingga pemrosesan data yang kompleks. 7. Dapat menggunakan rekursi untuk menyelesaikan permasalahan yang dapat dipecahkan secara iteratif, serta memahami prinsip-prinsip dasar rekursi. 8. Dapat mengorganisir kode program ke dalam fungsi-fungsi yang berbeda untuk meningkatkan keterbacaan, kemudahan pemeliharaan, dan modularitas program.
		Metode/Cara pengerjaan	1. Dapat membuat program dengan diskusi kelompok / personal 2. Program dibuat menggunakan PC atau Laptop
		Deskripsi luaran tugas	Program manipulasi String dan Fungsi
	Kriteria Penilaian:		
		d. Kelengkapan	25 %
		e. Kebenaran	50 %
		f. Ketelitian	25 %
Minggu ke:	13	Tugas ke: 6	
	Tujuan tugas:		
		1. Menjelaskan konsep dasar pointer dan cara menggunakan pointer 2. Menerapkan pointer dalam program 3. Menjelaskan konsep dasar struktur 4. Menjelaskan cara mendeklarasikan variabel struktur dan menerapkan variabel struktur dalam program	
	Uraian tugas:		

		Obyek	Membuat program dengan pointer dan struktur
		Yang dilakukan	1. Dapat menjelaskan secara tepat tentang konsep dasar pointer, termasuk pengertian pointer, cara mendeklarasikan, menginisialisasi, mengakses nilai yang disimpan di alamat yang ditunjuk, dan menggambarkan penggunaan pointer dalam pemrograman C++. 2. Dapat menunjukkan kemampuan dalam menggunakan pointer dalam penulisan kode program C++, termasuk penggunaan pointer untuk memanipulasi alamat memori, akses ke variabel, dan penggunaan pointer untuk membuat struktur data yang kompleks seperti linked lists, trees, atau dynamic memory allocation. 3. Dapat menerapkan konsep pointer dalam menyelesaikan masalah pemrograman yang lebih kompleks, termasuk penggunaan pointer untuk transfer data antar fungsi, dynamic memory management untuk menghindari memory leaks, atau pengoptimalan performa program dengan menggunakan pointer. Mahasiswa juga dapat menunjukkan pemahaman tentang keamanan penggunaan pointer, seperti menghindari null pointer dereferencing atau dangling pointers. 4. Dapat menjelaskan konsep dasar dari struktur (struct) dalam bahasa pemrograman C++, termasuk bagaimana mendefinisikan struktur, mendeklarasikan variabel struktur, dan mengakses anggota struktur dengan menggunakan operator titik. 5. Dapat merancang dan mengimplementasikan program-program dalam bahasa C++ yang menggunakan struktur untuk merepresentasikan entitas atau objek dalam suatu sistem. Ini mencakup kemampuan dalam mendesain struktur data yang tepat untuk menyimpan informasi yang diperlukan dan mengakses informasi tersebut dengan benar dalam program. 6. Dapat mengidentifikasi dan merancang solusi menggunakan struktur dalam bahasa C++ untuk menyelesaikan berbagai macam masalah pemrograman, seperti pemrosesan data, pengelolaan informasi, atau representasi objek dalam program. 7. Dapat mengintegrasikan struktur dengan konsep pemrograman lainnya, seperti penggunaan fungsi atau algoritma, untuk mencapai solusi yang efisien dan efektif
		Metode/Cara pengerjaan	1. Dapat membuat program dengan diskusi kelompok / personal 2. Program dibuat menggunakan PC atau Laptop
		Deskripsi luaran tugas	Program menggunakan teknik Pointer dan Struktur

	Kriteria Penilaian:	
	g. Kelengkapan	25 %
	h. Kebenaran	50 %
	i. Ketelitian	25 %

KETERANGAN

1. TUJUAN TUGAS

adalah rumusan kemampuan yang diharapkan dapat dicapai oleh mahasiswa bila berhasil mengerjakan tugas ini (hardskill dan softskill).

2. URAIAN TUGAS

- a. Obyek : berisi deskripsi obyek material yang akan di pelajari dalam tugas ini
- b. Yang dilakukan : uraian besaran, Tingkat kerumitan dan keluasan masalah dari obyek material yang harus di pelajari, Tingkat ketajaman dan kedalaman studi yang distandarkan. Bisa juga ditetapkan hasil yang harus dipresentasikan di forum diskusi.
- c. Metode / cara pengerjaan: berupa petunjuk tentang teori/Teknik/alat yang sebaikna dihunakan, alternatif Langkah-langkah yang bisa ditempuh, data dan buku acuan yang wajib dan yang disarankan untuk digunakan, ketentuan dikerjakan secara kelompok/individual.

d. Deskripsi luaran tugas : adalah uraian tentang bentuk hasil studi/kinerja yang harus ditunjukkan/disajikan(missal hasil studi tersaji dalam paper minimum 20 halaman termasuk skema, tabel dan gambar dengan ukuran kertas kuarto, diketik dengan tipe dan besaran huruf tertentu dan mungkin dilengkapi sajian dalam bentuk CD dengan format power point).

3. KRITERIA PENILAIAN

Berisi butir-butir indikator yang dapat menunjukkan Tingkat keberhasilan mahasiswa dalam usaha mencapai kompetensi yang telah dirumuskan.