

RENCANA PEMBELAJARAN SEMESTER

Mata Kuliah : **Pemrograman Berorientasi Objek**  
Semester : **2**  
Kode : **TIF1202**  
SKS : **3**  
Program Studi : **Teknologi Informasi**  
Dosen : **1. Opim Salim Sitompul**

**Capaian Pembelajaran:**  
Setelah mengikuti Mata Kuliah Pemrograman Berorientasi Objek, mahasiswa semester 2 Program Studi S1 Teknologi Informasi akan dapat:  
a) menerapkan teknik pemrograman berorientasi objek untuk membangun program komputer  
b) menghasilkan program komputer yang terstruktur, *readable*, dan mudah di-*maintain*  
c) mengambil keputusan secara tepat dalam konteks penyelesaian masalah pada pengembangan program menggunakan teknik pemrograman berorientasi objek

MINGGU KE	KEMAMPUAN AKHIR YANG DIHARAPKAN	BAHAN KAJIAN ( MATERI AJAR)	BENTUK PEMBELAJARAN	KRITERIAN PENILAIAN ( INDIKATOR)	BOBOT NILAI
1	Mampu membuat kelas dan objek	1. Pengertian kelas dan objek 2. Sasaran pemrograman berorientasi objek 3. Memahami kelas dan objek 4. Mengetahui kelas <i>I/O Stream</i> 5. Mendefinisikan kelas 6. Menurunkan objek 7. Namespace 8. Mengakses anggota kelas 9. Penentu akses 10. Diagram kelas berdasarkan UML	1. Sinkronus	1. Quiz 2. Tugas Pemrograman 3. UTS 4. UAS	10% 30% 25% 35%
2	Mampu menggunakan fungsi input dan output	1. Anggota fungsi input dan output 2. Membaca satu baris dari keyboard (stdin) 3. Menghitung karakter dalam satu baris input 4. Membaca input karakter per karakter 5. Mengetahui tanda akhir string	1. Sinkronus 2. Project-Based Learning		
3	Mampu membuat constructor dan destructor pada kelas	1. Mengetahui constructor 2. Mengetahui destructor 3. Mendefinisikan constructor dan destructor 4. Mendefinisikan constructor dan destructor 5. Default constructor 6. Multiple constructor	1. Sinkronus		
4	Mampu menggunakan file I/O untuk membaca/menyimpan data dari/ke file	1. Constructor dan manipulator 2. Array of classes 3. Membuka file untuk operasi input 4. Menguji keberhasilan file I/O 5. Operasi file biner	1. Sinkronus 2. Project-Based Learning		

5	Mampu membuat kelas yang menerapkan konsep inheritance	<ol style="list-style-type: none"> <li>1. Pengertian inheritance</li> <li>2. Menggunakan fungsi constructor pada inheritance</li> <li>3. Multiple inheritance</li> <li>4. Method overriding</li> </ol>	1. Sinkronus		
6	Mampu menerapkan penyembunyian data pada program	<ol style="list-style-type: none"> <li>1. Konsep dasar abstraksi</li> <li>2. Konsep dasar enkapsulasi</li> <li>3. Access specifiers</li> <li>4. Mutator (fungsi setter dan getter)</li> </ol>	1. Sinkronus		
7	Mampu membuat fungsi berlebihan beban	<ol style="list-style-type: none"> <li>1. Pengantar fungsi berlebihan beban</li> <li>2. Fungsi berlebihan beban pada array berjenis <i>int</i> dan array berjenis <i>float</i></li> <li>3. Operator berlebihan beban pada <i>constructor class</i></li> <li>4. Fungsi dapat memiliki 2 atau 3 buah nilai parameter</li> <li>5. Default parameter</li> </ol>	1. Sinkronus		
8	Mampu membuat operator berlebihan beban	<ol style="list-style-type: none"> <li>1. Overloading Unary Operator <ul style="list-style-type: none"> <li>- Tanpa argumen</li> <li>- Dengan argumen</li> <li>- Objek sementara tanpa nama</li> <li>- Postfix dan prefix</li> </ul> </li> <li>2. Overloading binary operator <ul style="list-style-type: none"> <li>- Aritmatika</li> <li>- Polar</li> <li>- String</li> </ul> </li> </ol>	<ol style="list-style-type: none"> <li>1. Sinkronus</li> <li>2. Project-Based Learning</li> </ol>		
9	Mampu mengidentifikasi lingkup atribut	<ol style="list-style-type: none"> <li>1. Deklarasi vs definisi</li> <li>2. Dasar-dasar ruang lingkup</li> <li>3. Ruang lingkup fungsi</li> <li>4. Ruang lingkup file</li> <li>5. Ruang lingkup kelas</li> <li>1. Global resolution operator</li> </ol>	1. Sinkronus		
10	Mampu membuat template fungsi	<ol style="list-style-type: none"> <li>2. Konsep template</li> <li>3. Template fungsi</li> <li>4. Mendefinisikan template</li> <li>5. Menggunakan template</li> <li>6. Argumen deduction</li> <li>7. Parameter template</li> </ol>	<ol style="list-style-type: none"> <li>1. Sinkronus</li> <li>2. Project-Based Learning</li> </ol>		
11	Mampu menerapkan konsep fungsi virtual dan polimorfisme pada program	<ol style="list-style-type: none"> <li>1. Konsep dasar</li> <li>2. Deklarasi fungsi virtual</li> <li>3. Polimorfisme</li> <li>4. Non-polimorfisme</li> </ol>	1. Sinkronus		

		5. Polimorfisme vs overloading 6. Menggunakan anggota fungsi kelas dasar			
12	Mampu menerapkan exception handling untuk menangani kesalahan pemrograman	1. Memahami exception 2. Bagaimana menangani exception 3. Mengembalikan lebih banyak informasi dengan exception 4. Multiple exception 5. Exception interface 6. Nested exception	1. Sinkronus		
13	Mampu menerapkan kelas kontainer (container class) dan map	1. Pengertian container class 2. Vektor 3. Fungsi pada vektor 4. Map 5. Multimap	1. Sinkronus		
14	Mampu menerapkan alokasi memori dinamik pada program	1. Memahami konsep dasar alokasi memori dinamik 2. Pengalokasian memori statik 3. Pengalokasian memori dinamik 4. Linked-list 5. Doubly linked-list	1. Sinkronus 2. Project-Based Learning		
15	Mampu mengimplementasikan queue menggunakan linked-list	1. Mendeklarasikan jenis data menggunakan typedef 2. Mendefinisikan node 3. Mendefinisikan class queue 4. Fungsi is_empty() dan is_full() 5. Menambah antrian 6. Keluar dari antrian 7. Antrian berprioritas	1. Sinkronus 2. Project-Based Learning		
16	Mampu mengimplementasikan stack menggunakan linked-list	1. Mendeklarasikan node 2. Mendefinisikan class stack 3. Fungsi is_empty() dan is_full() 4. Operasi push() 5. Operasi pop()	1. Sinkronus 2. Project-Based Learning		

Referensi:

1. Jamsa, Kris. Success with C++, Jamsa Pr, 1993
2. Hortsman, Cay. C++ for Everyone, Wiley, 2010
3. Sitompul, Opim S. Modul Pemrograman Berorientasi-Objek, 2009

Keterangan:

\* Metode pembelajaran yang akan digunakan sesuai dengan nomor pertemuan

\*\* Bobot nilai (%) untuk satu semester