

## VI033301 Pemrograman Berorientasi Obyek

<b>Jam/Minggu</b> 3 Jam	<b>Semester :</b> 3 (tiga)									
<b>Kode Mata Kuliah</b>	VI033301									
<b>Nama Matakuliah</b>	Pemrograman Berorientasi obyek									
<b>Silabus ringkas</b>	Kuliah ini mengajarkan konsep pemrograman berorientasi obyek disertai contoh penerapan konsep pemrograman berorientasi obyek pada bahasa Java. Pada kuliah ini mahasiswa diharapkan dapat merancang dan mengimplementasikan permasalahan ke dalam konsep pemrograman berorientasi obyek dengan diberikan tugas-tugas pemrograman dan tugas proyek.									
<b>Tujuan Instruksional Umum (TIU)</b>	<p>Mahasiswa diharapkan dapat:</p> <ul style="list-style-type: none"> <li>• Memahami konsep pemrograman berorientasi obyek yang meliputi abstraksi data, enkapsulasi, penurunan dan polymorphism.</li> <li>• Mengimplementasikan konsep pemrograman berorientasi obyek ke dalam bahasa pemrograman Java.</li> <li>• Memecahkan masalah menjadi sebuah desain konsep pemrograman berorientasi obyek dan kemudian mengimplementasikannya menjadi sebuah program komputer.</li> </ul>									
<b>Mata Kuliah Penunjang</b>	1. Keterampilan Komputer 2. Konsep Pemrograman 3. Struktur Data									
<b>Penilaian</b>	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;"></td> <td style="text-align: right;">UTS =</td> <td style="text-align: right;">35%</td> </tr> <tr> <td></td> <td style="text-align: right;">UAS =</td> <td style="text-align: right;">40 %</td> </tr> <tr> <td></td> <td style="text-align: right;">Tugas =</td> <td style="text-align: right;">25 %</td> </tr> </table>		UTS =	35%		UAS =	40 %		Tugas =	25 %
	UTS =	35%								
	UAS =	40 %								
	Tugas =	25 %								
<b>Daftar Pustaka</b>	1. Sun Java Programming, <i>Sun Educational Services</i> , Student Guide, Sun Microsystems 2001 2. John R. Hubbard, <i>Programming With Java</i> , McGraw-Hill 3. Patrick Niemeyer, Jonathan Knudsen, <i>Learning Java</i> , O'reilly, CA, 2000 4. Philip Heller, Simon Roberts, <i>Complete Java 2 Certification Study Guide</i> , Third Edition, Sybex, 2002, San Francisco, London									

### Uraian Rinci Materi Kuliah

Mg#	Tujuan Instruksional Khusus (TIK)	Topik	Sub Topik	Referensi	Media PT/OHP/LCD/PC
1.	<ul style="list-style-type: none"> <li>• Memberikan latar belakang serta pemahaman tentang konsep pemrograman berorientasi obyek dan perbandingannya dengan pemrograman procedural</li> <li>• Memahami konsep dasar pemrograman berorientasi obyek</li> <li>• Memperkenalkan dasar-dasar dan aturan pemrograman di Java sebagai basis pemrograman OOP</li> </ul>	Pengenalan dan Dasar pemrograman Java	<ul style="list-style-type: none"> <li>• Sejarah Java</li> <li>• Teknologi Java(J2SE, J2ME, J2EE, Java Card)</li> <li>• Perbandingan pemrograman prosedural dan pemrograman berorientasi obyek</li> <li>• Kelebihan Java</li> <li>• Komentar</li> <li>• Struktur Java (package, import, main method)</li> <li>• identifiers</li> <li>• Kata-kata kunci</li> <li>• Tipe data primitif</li> <li>• Inisialisasi &amp; nilai default</li> <li>• Casting &amp; promotion</li> </ul>	1, 2, 3, 4	LCD/PC/PT
2	<ul style="list-style-type: none"> <li>• Memperkenalkan operator-operator dasar yang ada di Java</li> <li>• Mengetahui berbagai macam flow control dan perulangan yang sesuai untuk menyelesaikan berbagai kasus.</li> </ul>	Operator, percabangan dan perulangan	<ul style="list-style-type: none"> <li>• Bitwise</li> <li>• AND (&amp;), OR ( ), XOR (^), NOT (!)</li> <li>• Short circuit AND (&amp;&amp;), OR (  )</li> <li>• Shift</li> <li>• Left shift (&lt;&lt;), right shift (&gt;&gt;), unsigned right shift (&gt;&gt;&gt;)</li> <li>• increment (++)</li> <li>• decrement (--)</li> <li>• Operator precedence</li> <li>• if, if-else, else-if, switch, for, while, do-while, break &amp; continue</li> </ul>	1, 2, 3, 4	LCD/PC/PT
3	<ul style="list-style-type: none"> <li>• Memahami bagaimana cara membuat class dan cara pengaksesan anggota class</li> <li>• Mengetahui cara membuat konstruktor</li> <li>• Memahami teknologi garbage collection.</li> <li>• Pengenalan UML</li> </ul>	Pembuatan class	<ul style="list-style-type: none"> <li>- Mendeklarasikan class, attribute dan method</li> <li>- Mengakses anggota object</li> <li>- Mengenal instanceof</li> <li>- Membuat konstruktor</li> <li>- Garbage Collection</li> <li>- Mengenal notasi UML (Class Diagram)</li> </ul>	1, 2, 3, 4	
4	Memahami konsep String dan array dengan berbagai dimensi dan dapat menerapkannya dalam kasus yang sesuai	Array	<ul style="list-style-type: none"> <li>- Declaring arrays</li> <li>- Creating arrays</li> <li>- Initializing arrays</li> <li>- Declaring &amp; creating arrays</li> <li>- Declaring, creating &amp; initializing arrays</li> <li>- Multidimensional arrays</li> <li>- Array bounds</li> <li>- Array resizing</li> <li>- Copying arrays</li> <li>- Membuat dan bekerja dengan String</li> </ul>	1, 2, 3, 4	LCD/PC/PT

Mg#	Tujuan Instruksional Khusus (TIK)	Topik	Sub Topik	Referensi	Media PT/OHP/LCD/PC
5	Memahami konsep String dan array dengan berbagai dimensi dan dapat menerapkannya dalam kasus yang sesuai	String	Penyimpanan string - String Immutability - Konstruktor dan method Class String - StringBuffer class - Konstruktor dan method Class StringBuffer		
6	Memahami dasar-dasar pemrograman OOP	Dasar-dasar pemrograman OOP	- Encapsulation - Information hiding - Declaring constructor - Overloading constructor - Importing classes - this keyword	1, 2, 3, 4	
7	Memahami konsep inheritance	Inheritance	- Basic Concept of Inheritance - Declaring inheritance in Java - Simple program - Accessing parent's member - Accesibility criteria - super keyword - Constructor does not been inherited - Invoking parent's constructor	1, 2, 3, 4	
8	Memahami konsep polimorfisme	Polimorfisme	- Basic concept of polymorphism - Overloading methods - Overriding methods - Virtual method Invocation - Rules about overridden methods - Polymorphic arguments - instanceof statements - Casting objects	1, 2, 3, 4	
UTS					
9	- Memahami bagaimana cara membandingkan obyek kelas, dan memakai wrapper class - Memahami pemakaian static dan final - Memahami abstract, interface, nested dan inner class	Kelas lanjut	- Comparing objects - Wrapper class - static & final keyword Abstract class - Intefaces - Nested dan inner class	1, 2, 3, 4	
10	Mengetahui berbagai jenis error dan sifatnya sehingga mampu melakukan proses penanganan error	Exception handling	<ul style="list-style-type: none"> <li>• Exception categories</li> <li>• Common exception</li> <li>• Exeption</li> <li>• Exception handling: try – catch - finally</li> <li>• The handle or declare rule</li> <li>• Method overriding and exception</li> <li>• Creating your own exception</li> </ul>	1, 2, 3, 4	LCD/PC/PT

Mg#	Tujuan Instruksional Khusus (TIK)	Topik	Sub Topik	Referensi	Media PT/OHP/LCD/PC
11	Memahami penggunaan io stream untuk penanganan input dan output.	Input dan Output	<ul style="list-style-type: none"> <li>o Navigating a file system: Directory separator, method of the file Class</li> <li>o Java io hierarchy</li> <li>o Utilizing FileInputStream, FileOutputStream, RandomAccessFile</li> <li>o Translating between Unicode Text and other Character Encodings</li> <li>o Stream</li> <li>o Readers</li> <li>o Writers</li> </ul>		
12	Memahami penggunaan java.util package untuk menyelesaikan suatu kasus penyimpanan koleksi	Java.util package : Collection API	<ul style="list-style-type: none"> <li>• Collection framework</li> <li>• Collection interface</li> <li>• Set</li> <li>• Sorted Set</li> <li>• List</li> <li>• Map</li> <li>• Sorted Map</li> <li>• Pengurutan data menggunakan Comparable dan Comparator</li> </ul>	1, 2, 3, 4	LCD/PC/PT
13	Memahami konsep dan penggunaan Generic	Generic	<ul style="list-style-type: none"> <li>• Introduction</li> <li>• Defining Simple Generics</li> <li>• Generics and Subtyping</li> <li>• Generics Methods</li> <li>• Interoperating with Legacy Code (Using Legacy Code in Generic Code, Erasure and Translation, Using Generic Code in Legacy Code)</li> <li>• The Fine Print (a Generic Class is Shared by all its Invocations, Casts and InstanceOf, Arrays)</li> </ul>		
14	Aplikasi OOP untuk penanganan GUI	GUI	<ul style="list-style-type: none"> <li>- Overview of AWT</li> <li>- Package of java.awt</li> <li>- Container</li> <li>- Frame</li> <li>- Panel</li> <li>- Layout managers (Flow layout, Border layout, Grid layout + mixed layout)</li> </ul>	1	
15	Penanganan event handling	Event handling	<ul style="list-style-type: none"> <li>- Event definition</li> <li>- Delegation model</li> <li>- Simple program</li> <li>- Event categories</li> <li>- Complex example</li> </ul>	1	

<b>Mg#</b>	<b>Tujuan Instruksional Khusus (TIK)</b>	<b>Topik</b>	<b>Sub Topik</b>	<b>Referensi</b>	<b>Media PT/OHP/LCD/PC</b>
			<ul style="list-style-type: none"> <li>- Multiple listeners</li> <li>- Event adapters</li> <li>- Event handling using -</li> <li>- Inner classes</li> <li>- Event handling using Anonymous classes</li> </ul>		
16	Memahami bagaimana cara membangun aplikasi berbasis GUI	Membangun aplikasi berbasis GUI	<ul style="list-style-type: none"> <li>- AWT components</li> <li>- Component events</li> <li>- Pembuatan menu, menubar, menuitem</li> <li>- Pembuatan aplikasi berbasis GUI</li> </ul>	1	
UAS					