



**INSTITUT TEKNOLOGI TELKOM SURABAYA**  
**FAKULTAS TEKNOLOGI INFORMASI DAN INDUSTRI**  
**PRODI STUDI TEKNOLOGI INFORMASI**

**Kode Dokumen**

**RENCANA PEMBELAJARAN SEMESTER**

Mata Kuliah (MK)	Kode MK	Rumpun MK/Kelompok Keahlian (KK)	Bobot (SKS)	Semester	Tanggal Penyusunan
Pemecahan Masalah Berbasis Obyek	ITA2223	Algoritma & Pemrograman / <i>Software Development</i>	3	3	26 Maret 2018
<b>OTORISASI</b>	<b>Pengembang RPS</b>		<b>Koordinator RMK</b>		<b>Ketua Program Studi</b>
	Farah Zakiyah Rahmanti, M.T.		-		Farah Zakiyah Rahmanti, M.T.
<b>Capaian Pembelajaran (CP)</b>	<b>CPL-PRODI</b>				
	[S-3]	Berkontribusi dalam peningkatan mutu kehidupan bermasyarakat, berbangsa, bernegara, dan kemajuan peradaban berdasarkan Pancasila.			
	[KU-1]	Mampu menerapkan pemikiran logis, kritis, sistematis dan inovatif dalam konteks pengembangan atau implementasi ilmu pengetahuan dan teknologi yang memperhatikan dan menerapkan nilai humaniora yang sesuai dengan bidang keahliannya.			
	[KK-3]	Menerapkan konsep dan teori pemrograman untuk membangun dan mengembangkan sistem aplikasi TIK.			
	[P-01]	Menjelaskan konsep dan teori dasar logika dan struktur diskrit untuk mendukung pemodelan dan analisa masalah.			
<b>CP-MK</b>	<b>CP-MK</b>				
	[ C - 2 ]	Mahasiswa mampu menjelaskan konsep-konsep dasar OOP dalam menyelesaikan permasalahan.			
	[ P - 4 ]	Mahasiswa mampu merancang penyelesaian permasalahan pada OOP berupa kelas diagram.			
	[ A - 4 ]	Mahasiswa mampu menerapkan konsep OOP pada aplikasi java berbasis GUI/Swing yang terkoneksi dengan DBMS.			
<b>Deskripsi Singkat MK</b>	Mata kuliah ini mempelajari konsep-konsep dasar pada object oriented programming (OOP), seperti class, object, encapsulation, inheritance, polymorphism, interface, abstract class. Secara spesifik konsep-konsep tersebut akan dipelajari dengan bantuan bahasa pemrograman Java. Selain konsep-konsep dasar OOP, pada mata kuliah ini juga akan dipelajari konsep exception handling, multithreading, object persistence, berbagai Collection, Java API, database connection, GUI (Graphical User Interface), serta deployment.				
<b>Materi Pembelajaran/ Pokok Bahasan</b>	<ol style="list-style-type: none"> <li>1. Pengenalan Mata Kuliah Pemecahan Masalah Berbasis Obyek</li> <li>2. Class &amp; Object</li> <li>3. Encapsulation</li> <li>4. Class Relationship ( Class Diagram)</li> <li>5. Inheritance &amp; Polymorphism</li> <li>6. Interface &amp; Abtract Classes</li> </ol>				

	7. Static & Final Method & Variables 8. Exception Handling 9. Object Persistence 10. Multi Threading 11. Using Java Library 12. Collections 13. Making Connection with Database 14. GUI & Swing 15. Deployment	
Pustaka	<b>Utama</b>	[1] Head First Java, 2nd edition, 2008, Bert Bates and Kathy Sierra, O'Reilly [2] Java™ How to Program, 9th, 2012, Prentice Hall [3] Head First Object Oriented Design and Analysis, 1st edition, 2006, Brett D. McLaughlin, Gary Pollice, David West, O'Reilly Media
	<b>Pendukung</b>	
Media Pembelajaran	<b>Perangkat Keras</b>	<b>Perangkat Lunak</b>
	Komputer, LCD, Projector	Netbeans / Eclipse, Power Point, PDF, Web Browser
Team Teaching	[1] Farah Zakiyah Rahmanti, M.T.	
Matakuliah Prasyarat	Algoritma dan Struktur Data	

Minggu ke-	Sub-CPMK (Kemampuan Akhir yang diharapkan)	Bahan Kajian [Pustaka/Materi Ajar]	Metode Pembelajaran [Waktu]	Indikator	Kriteria dan Bentuk Penilaian	Bobot Nilai (%)
(1)	(2)	(3)	(4)	(5)	(6)	(7)
1	Mahasiswa mampu menjelaskan tentang sejarah OOP, perlunya OOP untuk menyelesaikan permasalahan, dan perbedaan konsep OOP dengan prosedural.  Mahasiswa mampu menjelaskan tentang konsep kelas dan obyek, tipe data primitive dan referensi, object	<b>Pengenalan Mata Kuliah PMBO</b> - Sejarah OOP. - Perlunya OOP untuk menyelesaikan permasalahan - Perbedaan Konsep OOP dengan Konsep Prosedural - Bahasa pemrograman Java yang akan digunakan untuk mempelajari OOP.	<b>Kuliah</b>  <b>Diskusi</b> [TM : 1 x (3 x 50'')]  <b>Tugas 1 :</b> Merancang kelas dan obyek sesuai studi kasus yang diberikan ke dalam	<b>Pencapaian kemampuan yang direncanakan :</b> -Ketepatan menjelaskan sejarah OOP. -Ketepatan menjelaskan perlunya OOP untuk menyelesaikan permasalahan. -Ketepatan menjelaskan	<b>Kriteria :</b> Rubrik Holistic  <b>Bentuk Non-Test :</b> -Membuat kelas dan obyek ke dalam Bahasa Java. -Presentasi	

	<p>cycle, constructor &amp; garbage.</p> <p>[C2, A3]</p>	<p><b>Class &amp; Object</b></p> <ul style="list-style-type: none"> <li>- Perbedaan antara kelas dan object.</li> <li>- Pembuatan kelas (sederhana) dan object dalam bahasa Java.</li> <li>-Primitif dan Referensi : Tipe data, kata kunci, reference variable, object declaration &amp; assignment, objects in garbage collector, arrays</li> <li>- Object cycle : life and death</li> <li>- Constructor &amp; Garbage</li> </ul>	<p>pemrograman Java.</p> <p><b>[BT + BM : (1+1)x(3x60'')]</b></p>	<p>perbedaan konsep OOP dengan prosedural.</p> <ul style="list-style-type: none"> <li>-Ketepatan implementasi konsep OOP dalam Bahasa Java.</li> <li>-Ketepatan menjelaskan konsep kelas dan obyek.</li> <li>-Ketepatan memahami penggunaan serta dapat membuat program sederhana menggunakan tipe data, kata kunci, reference variable, object declaration &amp; assignment, arrays.</li> <li>-Ketepatan menjelaskan konsep life and death dari Object Cycle</li> <li>-Ketepatan menjelaskan penggunaan constructor &amp; garbage.</li> </ul>		
2	<p>Mahasiswa mampu menjelaskan konsep enkapsulasi dalam suatu program sederhana.</p>	<p><b>Encapsulation</b></p> <ul style="list-style-type: none"> <li>- Perlunya enkapsulasi</li> <li>- Object state</li> <li>- Instance variable &amp; local variables</li> <li>- Methods argument &amp; return type</li> <li>- Pass by value</li> <li>- Getters and setters</li> <li>- Contoh class diagram yang merepresentasikan enkapsulasi</li> </ul>	<p><b>Kuliah</b></p> <p><b>Diskusi</b></p> <p><b>[TM : 1 x (3 x 50'')]</b></p> <p><b>Tugas 2 :</b> Mengimplementasikan konsep enkapsulasi pada studi kasus yang diberikan.</p> <p><b>[BT + BM : (1+1)x(3x60'')]</b></p>	<p><b>Pencapaian kemampuan yang direncanakan :</b></p> <ul style="list-style-type: none"> <li>-Ketepatan menjelaskan perlunya enkapsulasi.</li> <li>-Ketepatan memahami dan mengimplementasikan enkapsulasi ke dalam bentuk program sederhana.</li> </ul>	<p><b>Kriteria :</b> Rubrik Holistic</p> <p><b>Bentuk Non-Test :</b></p> <ul style="list-style-type: none"> <li>-Membuat beberapa kelas dan obyek sehingga dapat mengimplementasikan konsep enkapsulasi.</li> <li>-Presentasi</li> </ul>	

3	Mahasiswa mampu memahami diagram kelas beserta makna simbol relasi yang terdapat pada diagram kelas.	<b>Class Relationship ( Class Diagram)</b> - association, aggregation, composition	<b>Kuliah</b>  <b>Diskusi</b> [TM : 1 x (3 x 50'')]  <b>Tugas 3 :</b> Menyusun kelas diagram dengan contoh kasus tertentu.  <b>[BT + BM :</b> <b>(1+1)x(3x60'')]</b>	<b>Pencapaian kemampuan yang direncanakan :</b> -Ketepatan menjelaskan diagram kelas beserta makna simbol relasi yang terdapat pada diagram kelas.	<b>Kriteria :</b> Rubrik Holistic  <b>Bentuk Non-Test :</b> -Menyusun kelas diagram dari suatu kasus yang diberikan. -Presentasi	
4	Mahasiswa mampu menjelaskan konsep inheritance & polymorphism.	<b>Inheritance &amp; Polymorphism</b> - Perlunya inheritance & polymorphism - Bagaimana inheritance mengembangkan software reusability - Merancang inheritance - Superclass dan hubungan subclass & akses modifier - Method Overriding - Polymorphism : Deklarasi obyek & assignment, Method argument & return type - Method overloading - Constructor overloading (revisited) & superclass constructor - PAnggilan ke super(), this () - Contoh kelas diagram yang merepresentasikan inheritance & polymorphism	<b>Kuliah</b>  <b>Diskusi</b> [TM : 1 x (3 x 50'')]  <b>Tugas 4 :</b> Mengimplementasikan konsep inheritance & polymorphism dari kelas diagram yang diberikan ke dalam Bahasa Java.  <b>[BT + BM :</b> <b>(1+1)x(3x60'')]</b>	<b>Pencapaian kemampuan yang direncanakan :</b> -Ketepatan menjelaskan perlunya inheritance & polymorphism -Ketepatan memahami dan mengimplementasikan inheritance & polymorphism dalam suatu program sederhana.	<b>Kriteria :</b> Rubrik Holistic  <b>Bentuk Non-Test :</b> -Mengimplementasikan inheritance & polymorphism dari studi kasus yang diberikan ke dalam bentuk program Java sederhana. -Presentasi	
5	Mahasiswa mampu memahami perlunya dan mengimplementasikan	<b>Interface &amp; Abstract Classes</b> -Perlunya abstract classes.	<b>Kuliah</b>	<b>Pencapaian kemampuan yang direncanakan :</b>	<b>Kriteria :</b> Rubrik Holistic	

	interface dan abstract classes dalam suatu program sederhana.	<p>Perbedaan antara abstract &amp; concrete class</p> <ul style="list-style-type: none"> <li>- Abstract classess &amp; methods</li> <li>- Overloading &amp; overriding (revisited)</li> <li>- Casting</li> <li>- Perlunya interface</li> <li>- Pembuatan &amp; penggunaan interface</li> <li>- Contoh kelas diagram yang merepresentasikan interface &amp; abstract class</li> </ul>	<p><b>Diskusi</b> [TM : 1 x (3 x 50'')]</p> <p><b>Tugas 5 :</b> Mengimplementasikan konsep interface &amp; abstract classes ke dalam program sederhana.</p> <p>[BT + BM : (1+1)x(3x60'')]</p>	<ul style="list-style-type: none"> <li>-Ketepatan menjelaskan perlunya interface &amp; abstract classes</li> <li>-Ketepatan memahami dan mengimplementasikan interface &amp; abstract classes dalam suatu program sederhana.</li> </ul>	<p><b>Bentuk Non-Test :</b></p> <ul style="list-style-type: none"> <li>-Mengimplementasikan interface &amp; abstract classes dari studi kasus yang diberikan ke dalam bentuk program Java sederhana.</li> <li>-Presentasi</li> </ul>	
6	Mahasiswa mampu memahami perlunya dan mengimplementasikan static & final variabel dan method dalam suatu program sederhana.	<p><b>Static &amp; Final Method &amp; Variables</b></p> <ul style="list-style-type: none"> <li>- Static method</li> <li>- Static variable</li> <li>- Constants (static final variable)</li> <li>- Final method &amp; class</li> <li>- Wrapper classes &amp; autoboxing</li> <li>- Static imports</li> <li>- Contoh : Math class</li> <li>- Perbedaan Instance variable &amp; static variable</li> </ul>	<p><b>Kuliah</b></p> <p><b>Diskusi</b> [TM : 1 x (3 x 50'')]</p> <p><b>Tugas 6 :</b> Mengimplementasikan konsep static &amp; final method &amp; variable ke dalam program sederhana.</p>	<p><b>Pencapaian kemampuan yang direncanakan :</b></p> <ul style="list-style-type: none"> <li>-Ketepatan memahami konsep static &amp; final method &amp; variable.</li> <li>-Ketepatan memahami perbedaan antara instance variable &amp; static variable.</li> </ul>	<p><b>Kriteria :</b> Rubrik Deskriptif</p> <p><b>Bentuk Non-Test :</b> -Penilaian program sederhana dari penggunaan static &amp; final method &amp; variable.</p>	
7	Mahasiswa mampu memahami dan menggunakan Exception Handling, membuat kelas Exception sendiri.	<p><b>Exception Handling</b></p> <ul style="list-style-type: none"> <li>- Exception sebagai obyek</li> <li>- try-catch</li> <li>- throws</li> <li>- Try-catch- finally</li> <li>- Multiple exception</li> <li>- Polymorphism di dalam exception</li> <li>- Pembuatan exception sendiri</li> </ul>	<p><b>Kuliah</b></p> <p><b>Diskusi</b> [TM : 1 x (3 x 50'')]</p> <p><b>Tugas 7 :</b> Menyusun kelas Exception sendiri dan mengimplementasikan ke dalam program sederhana.</p>	<p><b>Pencapaian kemampuan yang direncanakan :</b></p> <ul style="list-style-type: none"> <li>-Ketepatan memahami konsep exception handling.</li> <li>-Ketepatan menggunakan exception handling.</li> <li>-Ketepatan membuat kelas exception sendiri.</li> </ul>	<p><b>Kriteria :</b> Rubrik Deskriptif</p> <p><b>Bentuk Test :</b> Soal tes menyusun kelas exception sendiri dengan contoh kasus yang diberikan.</p>	

8 Evaluasi Tengah Semester: Melakukan validasi hasil penilaian, evaluasi dan perbaikan proses pembelajaran berikutnya						
9	Mahasiswa mampu memahami dan mengimplementasikan Persistensi Objek dalam file.	<b>Object Persistence</b> <ul style="list-style-type: none"> <li>- Saving object state</li> <li>- Writing a serialized object to a file</li> <li>- Object serialization (serializable interface)</li> <li>- Deserializing an object</li> </ul>	<b>Kuliah</b>  <b>Diskusi</b> <b>[TM : 1 x (3 x 50'')]</b>  <b>Tugas 9 :</b> Mengimplementasikan persistensi & obyek ke dalam file.	<b>Pencapaian kemampuan yang direncanakan :</b> <ul style="list-style-type: none"> <li>-Ketepatan memahami persistensi dan obyek dalam file.</li> <li>-Ketepatan mengimplementasikan persistensi dan obyek dalam file.</li> </ul>	<b>Kriteria :</b> Rubrik Holistic  <b>Bentuk Non-Test :</b> <ul style="list-style-type: none"> <li>-Mengimplementasikan persistensi &amp; obyek ke dalam file pada studi kasus yang diberikan ke dalam bentuk program Java sederhana.</li> <li>-Presentasi</li> </ul>	
10	Mahasiswa mampu memahami dan mengimplementasikan konsep sinkronisasi menggunakan thread.	<b>Multi Threading</b> <ul style="list-style-type: none"> <li>- Launching new thread</li> <li>- Runnable interface</li> <li>- Thread' state</li> <li>- Runnable - running loop</li> <li>- Thread scheduler</li> <li>- Making and starting two thread</li> <li>- Locking object</li> <li>- Synchronized method using a lock</li> <li>- deadlock</li> </ul>	<b>Kuliah</b>  <b>Diskusi</b> <b>[TM : 1 x (3 x 50'')]</b>  <b>Tugas 10 :</b> Mengimplementasikan sinkronisasi menggunakan thread ke dalam bentuk program sederhana.	<b>Pencapaian kemampuan yang direncanakan :</b> <ul style="list-style-type: none"> <li>-Ketepatan memahami konsep sinkronisasi menggunakan thread</li> <li>-Ketepatan mengimplementasikan sinkronisasi dengan thread</li> </ul>	<b>Kriteria :</b> Rubrik Deskriptif  <b>Bentuk Non-Test :</b> <ul style="list-style-type: none"> <li>-Penilaian program sederhana dari sinkronisasi dengan thread.</li> </ul>	
11	Mahasiswa mampu menggunakan Java API untuk membantu membuat aplikasi.	<b>Using Java Library</b> <ul style="list-style-type: none"> <li>- Bagaimana cara menggunakan Java Library</li> <li>- Contoh sederhana</li> </ul>	<b>Kuliah</b>  <b>Diskusi</b> <b>[TM : 1 x (3 x 50'')]</b>  <b>Tugas 11 :</b> Mengimplementasikan program sederhana menggunakan Java API dalam membuat suatu	<b>Pencapaian kemampuan yang direncanakan :</b> <ul style="list-style-type: none"> <li>-Ketepatan memahami penggunaan Java API dalam membuat aplikasi.</li> </ul>	<b>Kriteria :</b> Rubrik Deskriptif  <b>Bentuk Non-Test :</b> <ul style="list-style-type: none"> <li>-Penilaian program sederhana menggunakan Java API dalam membuat aplikasi sederhana yang telah ditentukan.</li> </ul>	

			aplikasi sederhana yang telah ditentukan.			
12	Mahasiswa mampu menggunakan struktur data yang telah terdefinisi dalam Collections untuk membuat aplikasi.	<b>Collections</b> - Interface Collection and class Collections - Type wrapper classes for primitive types - Autoboxing and auto-unboxing - List: ArrayList - Iterator, LinkedList - Stack - Sets - Maps	<b>Kuliah</b>  <b>Diskusi</b> [TM : 1 x (3 x 50'')]  <b>Tugas 12 :</b> Mengimplementasikan struktur data yang telah terdefinisi dalam Collections untuk membuat aplikasi.	<b>Pencapaian kemampuan yang direncanakan :</b> -Ketepatan memahami penggunaan struktur data yang telah terdefinisi dalam Collections untuk membuat aplikasi.	<b>Kriteria :</b> Rubrik Holistic  <b>Bentuk Non-Test :</b> -Mengimplementasikan struktur data yang telah terdefinisi pada Collections untuk membuat aplikasi. -Presentasi	
13	Mahasiswa mampu membuat koneksi dari aplikasi berbasis java ke DBMS.	<b>Making Connection with Database</b> - JDBC - Oracle dan tau mysql	<b>Kuliah</b>  <b>Diskusi</b> [TM : 1 x (3 x 50'')]  <b>Tugas 13 :</b> Mengimplementasikan koneksi dari aplikasi Java ke DBMS.	<b>Pencapaian kemampuan yang direncanakan :</b> -Ketepatan membuat koneksi dari masing-masing aplikasi berbasis java yang telah dibuat ke DBMS.	<b>Kriteria :</b> Rubrik Deskriptif  <b>Bentuk Non-Test :</b> -Penilaian aplikasi berbasis java yang telah dibuat ke DBMS.	
14	Mahasiswa mampu membuat aplikasi berbasis GUI/Swing.  Mahasiswa mampu mengorganisasikan dan melakukan deploy pada aplikasi.	<b>GUI &amp; Swing</b> - User event - Listener interface  <b>Deployment</b> - Separate source code & class files - Membuat executable JAR - Menjalankan executable JAR - Package	<b>Kuliah</b>  <b>Diskusi</b> [TM : 1 x (3 x 50'')]  <b>Tugas 14 :</b> Mengimplementasikan aplikasi berbasis GUI/Swing.	<b>Pencapaian kemampuan yang direncanakan :</b> -Ketepatan membuat aplikasi berbasis GUI/Swing. -Ketepatan dalam mengorganisasikan dan deploy pada aplikasi.	<b>Kriteria :</b> Rubrik Deskriptif  <b>Bentuk Non-Test :</b> -Penilaian aplikasi berbasis GUI/Swing.	

15	Mahasiswa mampu membuat aplikasi java dengan mengimplementasikan konsep OOP berbasis GUI/Swing yang didalamnya terkoneksi pada DBMS.	<b>Presentasi Tugas Besar</b> - Rancangan kelas diagram pada aplikasi yang dibuat pada tugas besar. - Implementasi konsep OOP pada aplikasi. - Implementasi GUI/Swing pada aplikasi. - Implementasi koneksi aplikasi pada DBMS.	[TM : 1 x (3 x 50'')]	<b>Pencapaian kemampuan yang direncanakan :</b> -Ketepatan mengimplementasikan konsep OOP ke dalam aplikasi java. -Ketepatan melakukan koneksi aplikasi dengan DBMS. -Ketepatan mengimplementasikan program menggunakan GUI/Swing. -Ketepatan dalam melakukan deploy aplikasi.	<b>Kriteria :</b> Rubrik Holistic  <b>Bentuk Non-Test :</b> -presentasi & demo aplikasi tugas besar	
----	--	---	-----------------------	--	---	--

**16** **Evaluasi Akhir Semester: Melakukan validasi penilaian akhir dan menentukan kelulusan mahasiswa**

**Catatan:**  
(1). TM: Tatap Muka; TS: Penugasan Terstruktur; BM: Belajar Mandiri.  
(2). 1 sks = (50' TM + 50' PT + 60' BM)/Minggu  
(3). CPL-Prodi: Capaian Pembelajaran Lulusan Program Studi; CP-MK: Capaian Pembelajaran Mata-Kuliah  
(4). Simbol-simbol elemen KKNI pada CPL-Prodi: S = Sikap; KU = Ketrampilan Umum; KK = Ketrampilan Khusus; P = Pengetahuan

Rubrik Holistic pada Tugas Besar pada pertemuan ke-15 : Membuat aplikasi berbasis java menggunakan GUI/Swing yang menerapkan konsep OOP didalamnya dan aplikasi terkoneksi dengan DBMS.

<b>Grade</b>	<b>Skor</b>	<b>Indikator Kerja</b>
Kurang sekali	0	Tidak hadir. Tidak presentasi.
Kurang	1	Program tidak terselesaikan. Penyusunan rancangan kelas diagram tidak terselesaikan. Tidak komunikatif dan tidak menguasai materi dalam penyampaian presentasi. Tidak mampu menjawab pertanyaan.
Cukup	2	Program tidak berjalan dengan baik. Penyusunan rancangan kelas diagram tidak sesuai dengan konsep OOP. Kurang komunikatif dan kurang menguasai materi dalam penyampaian presentasi. Tidak mampu menjawab pertanyaan.
Baik	3	Fitur program ada yang tidak berjalan dengan baik. Penyusunan rancangan kelas diagram sesuai dengan konsep OOP. Komunikatif dan menguasai materi dalam penyampaian presentasi. Tidak mampu menjawab pertanyaan.
Sangat Baik	4	Program berjalan dengan baik (tanpa error). Penyusunan rancangan kelas diagram sesuai dengan konsep OOP. Komunikatif dan menguasai materi dalam penyampaian presentasi. Mampu menjawab pertanyaan.



**INSTITUT TEKNOLOGI TELKOM SURABAYA**  
**FAKULTAS TEKNOLOGI INFORMASI DAN INDUSTRI**  
**PRODI STUDI TEKNOLOGI INFORMASI**

**RENCANA TUGAS MAHASISWA**

Mata Kuliah (MK)	Kode MK	Rumpun MK/Kelompok Keahlian (KK)	Bobot (SKS)	Semester	Tahun Akademik
Pemecahan Masalah Berbasis Obyek	IT21T01	Algoritma dan Pemrograman / <i>Software Development</i>	3	3	Ganjil 2019/2020

**Dosen Pengampu**

Farah Zakiyah Rahmanti, M.T.

TUGAS KE-	JUDUL TUGAS
15	Membuat aplikasi berbasis java menggunakan GUI/Swing yang menerapkan konsep OOP didalamnya dan aplikasi terkoneksi dengan DBMS.

**SUB-CAPAIAN PEMBELAJARAN MATA-KULIAH**

Mahasiswa mampu membuat aplikasi java dengan mengimplementasikan konsep OOP berbasis GUI/Swing yang didalamnya terkoneksi pada DBMS.

**TUJUAN PENUGASAN**

Mahasiswa memahami konsep OOP, GUI/Swing, dan koneksi pada DBMS.

DESKRIPSI TUGAS	METODE Pengerjaan Tugas
<p>Objek Garapan:            Pembuatan aplikasi berbasis java menggunakan GUI/Swing yang menerapkan konsep OOP didalamnya dan aplikasi terkoneksi dengan DBMS.            (Studi kasus setiap mahasiswa berbeda)</p> <p>Batasan:            Bahasa pemrograman yang digunakan adalah Bahasa Java.            Mengimplementasikan aplikasi dengan GUI/Swing.            Melakukan koneksi dengan DBMS menggunakan oracle atau mysql.</p>	<ol style="list-style-type: none"> <li>1. Membuat aplikasi java dengan mengimplementasikan konsep OOP berbasis GUI/Swing yang didalamnya sudah terhubung dengan DBMS.</li> <li>2. Menyusun resume mengenai perancangan kelas diagram pada studi kasus yang telah diambil.</li> <li>3. Menyusun slide presentasi.</li> <li>4. Presentasi aplikasi java di kelas.</li> </ol>

<p>Format penyusunan rancangan kelas diagram bebas sesuai dengan format yang telah diunggah pada google drive.</p> <p>Relevansi: Dasar-dasar Pemrograman</p> <p>Manfaat:  Mahasiswa dapat menganalisa permasalahan dari studi kasus yang ada.  Mahasiswa dapat menyelesaikan permasalahan menggunakan konsep OOP.  Mahasiswa dapat mengimplementasikan studi kasus ke dalam aplikasi berbasis Java menggunakan GUI/Swing yang sudah terhubung dengan DBMS.  Mahasiswa dapat melakukan deploy aplikasi yang telah dibuat.</p>	
<b>BENTUK DAN FORMAT LUARAN TUGAS</b>	<b>INDIKATOR, KRITERIA DAN BOBOT PENILAIAN</b>
<p>Bentuk Luaran :</p> <p>Demo program komputer dilakukan dikelas, program harus sudah bisa dijalankan tanpa eror.</p> <p>Perancangan kelas diagram ditulis dengan MS Word sesuai dengan konsep OOP yang telah diunggah pada google drive.  Dikumpulkan dalam bentuk hardcopy.</p> <p>Slide Presentasi PowerPoint, terdiri dari : Text, grafik, tabel, gambar, animasi ataupun video clips, minimum 15 slide.  Dikumpulkan dalam bentuk softcopy format ekstensi (*.ppt, *.pptx), dengan sistematika nama file : nim1_namamhs.ppt</p>	<p>Program aplikasi Java (bobot 40%)  Program dapat berjalan dengan baik.  Semua fitur nya dapat berjalan dengan baik.  Program dapat menyelesaikan permasalahan dengan konsep OOP berbasis GUI/Swing yang terkoneksi dengan DBMS.  Program sudah dilakukan deploy.</p> <p>Resume rancangan kelas diagram (20%)  Ketepatan merancang ke dalam kelas diagram dalam menyelesaikan permasalahan dengan konsep OOP.</p> <p>Penyusunan Slide Presentasi (20%)  Jelas dan konsisten, sederhana dan inovatif, didukung dengan gambar atau video yang relevan.</p> <p>Presentasi (20%)  Bahasa komunikatif, penguasaan materi, penguasaan audiensi, pengendalian waktu, kejelasan dan ketajaman paparan, penguasaan media presentasi.</p>
<b>JADWAL PELAKSANAAN TUGAS</b>	<b>CATATAN /LAIN-LAIN</b>

Dilakukan pada minggu terakhir sebelum UAS (pertemuan ke-15) di kelas. Pengumuman hasil penilaian pada Minggu ke – 16.	Bobot penilaian tugas besar adalah 30% dari 100% penilaian mata kuliah ini. Tugas dikerjakan dan dipresentasikan secara individu.
<b>DAFTAR RUJUKAN</b>	
1. Head First Java, 2nd edition, 2008, Bert Bates and Kathy Sierra, O'Reilly 2. Java™ How to Program, 9th, 2012, Prentice Hall 3. Head First Object Oriented Design and Analysis, 1st edition, 2006, Brett D. McLaughlin, Gary Pollice, David West, O'Reilly Media	