






## RENCANA PEMBELAJARAN SEMESTER (SEMESTER LESSON PLAN)

Nomor Dok	FRM/KUL/01/02
Nomor Revisi	02
Tgl. Berlaku	1 Januari 2018
Standar SPMI	3.3.2

Disusun oleh ( <i>Prepared by</i> )	Diperiksa oleh ( <i>Checked by</i> )	Disetujui oleh ( <i>Approved by</i> )	Tanggal Validasi ( <i>Valid date</i> )
			
<b>Alek Wijaya, S.Kom., MIT</b>	<b>Alek Wijaya, S.Kom., MIT</b>	<b>Dedi Syamsuar, M.I.T., Ph.D.</b>	

- |  |   |  |  |
|--|---|--|--|
| 1. Fakultas ( <i>Faculty</i> )                   | : Ilmu Komputer   | Jenjang ( <i>Grade</i> )                     | : Strata 1 (S1)  |
| 2. Program Studi ( <i>Study Program</i> )        | : Teknik Informatika  | SKS ( <i>Credit</i> )                        | : 4 Semester ( <i>Semester</i> ) : 4   |
| 3. Mata Kuliah ( <i>Course</i> )                 | : Pemrograman Berorientasi Objek  | Sertifikasi ( <i>Certification</i> )         | : <input type="checkbox"/> Ya ( <i>Yes</i> ) <input checked="" type="checkbox"/> Tidak ( <i>No</i> ) |
| 4. Kode Mata Kuliah ( <i>Code</i> )              | : 2214224014  |  |  |
| 5. Mata Kuliah Prasyarat ( <i>Prerequisite</i> ) | : -   |  |  |
| 6. Dosen Koordinator ( <i>Coordinator</i> )      | : Ahmad Syazili , M.Kom.  | <input type="checkbox"/> Tim ( <i>Team</i> ) | <input checked="" type="checkbox"/> Mandiri ( <i>Personal</i> )                                      |
| 7. Dosen Pengampuh ( <i>Lecturer</i> )           | : Ahmad Syazili , M.Kom.<br>Hutrianto, M.M., M.Kom.<br>Firamon Syakti, M.M., M.Kom.<br>Ahmad Mutatkin Bakti, M.M., M.Kom. |  |  |

8. Capaian Pembelajaran Mata Kuliah (*Course Learning Outcomes*) :

Capaian Pembelajaran Lulusan (CPL) ( <i>Programme Learning Outcomes</i> )	CPL01	Memiliki kemampuan untuk mengumpulkan data dan menganalisis data tersebut untuk kebutuhan sebuah sistem
	CPL02	Memiliki kemampuan untuk merancang dan membangun sistem sesuai kebutuhan organisasi
Capaian Pembelajaran	CPMK011	Mampu mengumpulkan data kebutuhan sistem

Mata Kuliah (CPMK) ( <i>Course Learning Outcomes</i> )	CPMK012 CPMK021 CPMK022	Mampu menganalisa data tersebut untuk kebutuhan sistem Mampu merancang sistem dengan pendekatan yang sesuai Mampu membangun sistem sesuai dengan hasil perancangan				
SUB-CPMK0111-033	Mampu memahami perlunya OOP untuk menyelesaikan permasalahan					
SUB-CPMK0112-033	Mampu memahami perbedaan antara kelas dan object dalam bahasa Java					
SUB-CPMK0113-033	Mampu memahami makna dan perlunya enkapsulasi					
SUB-CPMK0121-033	Mampu memahami diagram kelas beserta makna simbol relasi yang terdapat pada diagram kelas					
SUB-CPMK0122-033	Mampu memahami makna dan perlunya inheritance dan polymorphism					
SUB-CPMK0211-033	Mampu mengimplementasikan interface dan abstract classes dalam suatu program sederhana.					
SUB-CPMK0212-033	Mampu memahami perlunya dan mengimplementasikan static & final variabel dan method dalam suatu program sederhana					
SUB-CPMK0213-033	Mampu memahami dan menggunakan Exception Handling					
SUB-CPMK0214-033	Mampu memahami dan mengimplementasikan Persistensi Objek dalam file.					
SUB-CPMK0215-033	Mampu memahami dan mengimplementasikan konsep sinkronisasi menggunakan thread					
SUB-CPMK0221-033	Mampu menggunakan Java API untuk membantu membuat aplikasi					
SUB-CPMK0222-033	Mampu menggunakan struktur data yang telah terdefinisi dalam Collections untuk membuat aplikasi					
SUB-CPMK0223-033	Mampu membuat koneksi dari aplikasi berbasis java ke DBMS					
SUB-CPMK0224-033	Mampu membuat aplikasi berbasis GUI/Swing					
SUB-CPMK0225-033	Mampu mengorganisasikan dan men-deploy aplikasi					
Matriks Sub-CPMK terhadap CPL dan CPMK						
			CPL 01		CPL02	
		SUB-CPMK	CPMK011	CPMK012	CPMK021	CPMK022
		SUB-CPMK0111-033	√			
		SUB-CPMK0112-033	√			
		SUB-CPMK0113-033	√			
		SUB-CPMK0121-033		√		
		SUB-CPMK0122-033		√		
		SUB-CPMK0211-033			√	
	SUB-CPMK0212-033			√		
	SUB-CPMK0213-033			√		

	SUB-CPMK0214-033			√	
	SUB-CPMK0215-033			√	
	SUB-CPMK0221-033				√
	SUB-CPMK0222-033				√
	SUB-CPMK0223-033				√
	SUB-CPMK0224-033				√
	SUB-CPMK0225-033				√

#### 9. Deskripsi Mata Kuliah (*Course Description*)

Deskripsi Singkat Mata Kuliah	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.
-------------------------------	--

#### 10. Bahan Kajian (*Main Study Material*)

Bahan Kajian (Materi Pembelajaran)	<ol style="list-style-type: none"> <li>1. Course Introduction</li> <li>2. Classes and object</li> <li>3. Encapsulation</li> <li>4. Class relationship (class diagram)</li> <li>5. Inheritance &amp; polymorphism</li> <li>6. Interface &amp; abstract classes</li> <li>7. Static &amp; final method &amp; variables</li> <li>8. Exception handling</li> <li>9. Object Persistence</li> <li>10. Multithreading</li> <li>11. Using java library (java API)</li> <li>12. Collections</li> <li>13. Making connection with database</li> <li>14. GUI &amp; SWING</li> <li>15. Deployment</li> </ol>
------------------------------------	--

11. Implementasi Pembelajaran Mingguan (*Implementation Process of weekly learning time*)

Minggu (Week)	Sub CPMK (Kemampuan akhir yang direncanakan) (Lesson Learning Outcomes)	Bahan Kajian/Materi Pembelajaran (Study Material)	Bentuk dan Metode Pembelajaran [Estimasi Waktu] (Learning Method)	Sumber Belajar (Learning Resource)	Penilaian		
					Indikator (Indicator)	Kriteria & bentuk (Criteria)	Bobot (%)
1	<ul style="list-style-type: none"> <li>Mahasiswa mengetahui sejarah OOP.</li> <li>Mahasiswa memahami perlunya OOP untuk menyelesaikan permasalahan, yang kurang tepat jika procedural oriented diterapkan.</li> <li>Mahasiswa mengetahui bahasa pemrograman Java yang akan digunakan untuk mempelajari OOP</li> </ul>	Course Introduction <ul style="list-style-type: none"> <li>OOP history</li> <li>compare procedural and oop by giving a simple case study</li> <li>Introduction to java</li> </ul>	Kuliah dan Diskusi (Luring) Tatap Muka [TM:2x(2x50’)] Tugas : Menjelaskan ringkasan mengenai OOP	Head First Java, 2nd edition, 2008, Bert Bates and Kathy Sierra, O'Reilly.	Mahasiswa dapat menjelaskan, sejarah OOP, menyelesaikan permasalahan dan dapat mempelajari Pemrograman Java	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk :</b> Diskusi Tanya jawab	5%
2-3	<ul style="list-style-type: none"> <li>Mahasiswa memahami perbedaan antara kelas dan object.</li> <li>Mahasiswa mampu membuat kelas (sederhana) dan object dalam bahasa Java.</li> <li>Mahasiswa memahami penggunaan serta dapat membuat program sederhana menggunakan tipe primitif, array, keyword, reference variable, deklarasi dan assignment objek,</li> <li>Mahasiswa memahami dan mengimplementasikan dalam program tentang siklus objek (penggunaan</li> </ul>	Classes and object <ul style="list-style-type: none"> <li>The difference between class and object, by giving some examples</li> <li>making a simple first object (Fokus: first programming with java)</li> <li>Primitive and reference: Type, keywords, reference variable, object declaration &amp; assignment, objects in garbage collector, arrays</li> <li>Object cycle: life and death</li> <li>Constructor &amp; Garbage collector</li> </ul>	Kuliah dan Diskusi (Luring) [TM:2x(2x50’)] Tugas : Membuat program sederhana dan mengimplementasikannya ke dalam kode program	Head First Java, 2nd edition, 2008, Bert Bates and Kathy Sierra, O'Reilly.	Mahasiswa dapat membuat program sederhana dan mengimplement asikannya ke dalam kode program	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk :</b> Diskusi, tanya jawab	5%

	constructor dan garbage collector)						
4	<ul style="list-style-type: none"> <li>• Mampu memahami makna dan perlunya enkapsulasi.</li> <li>• Mampu memahami dan dapat mengimplementasikan enkapsulasi dalam suatu program sederhana.</li> </ul>	<p>Encapsulation</p> <ul style="list-style-type: none"> <li>• Why do we need encapsulation?</li> <li>• Object state</li> <li>• Instance variable vs local variables</li> <li>• Methods argument &amp; return type</li> <li>• Pass by value</li> <li>• Getters and setters</li> <li>• • Sample class diagram representing encapsulation</li> </ul>	<p>Kuliah dan Diskusi (Daring) [TM:2x(2x50”)] Tugas : Menjelaskan perlunya enkapsulasi dan mengimplementasikannya dalam program sederhana</p>	<p>Head First Java, 2nd edition, 2008, Bert Bates and Kathy Sierra, O'Reilly.</p>	<p>Mahasiswa dapat menjelaskan tentang makna enkapsulasi dan dapat mengimplementasikan enkapsulasi dalam suatu program sederhana</p>	<p><b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk :</b> Diskusi, tugas</p>	5%
5-6	<p>Mahasiswa memahami diagram kelas beserta makna simbol relasi yang terdapat pada diagram kelas.</p>	<p>Class relationship (class diagram)</p> <ul style="list-style-type: none"> <li>• association, aggregation, composition</li> <li>• Example: simple class diagram &amp; code in java related</li> </ul> <p>*) Hanya mengenalkan saja, karena nantinya satu persatu relasi yg ada pada diagram kelas akan dibutuhkan saat menjelaskan materi inheritance, polimorfisme, kelas abstrak, interface</p>	<p>Kuliah dan Diskusi (Luring) [TM:2x(2x50”)] Tugas : Merancang diagram kelas</p>	<p>Head First Java, 2nd edition, 2008, Bert Bates and Kathy Sierra, O'Reilly.</p>	<p>Mahasiswa Dapat menjelaskan diagram kelas beserta makna simbol relasi yang terdapat pada diagram kelas</p>	<p><b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk :</b> Diskusi, tugas</p>	5%
7-8	<ul style="list-style-type: none"> <li>• Mahasiswa memahami makna dan perlunya inheritance dan</li> </ul>	<p>Inheritance &amp; polymorphism</p> <ul style="list-style-type: none"> <li>• Why do we need inheritance &amp;</li> </ul>	<p>Kuliah dan Diskusi (Daring) Tatap Muka</p>	<p>Head First Java, 2nd edition, 2008, Bert Bates</p>	<p>Mahasiswa Dapat menjelaskan</p>	<p><b>Kriteria :</b> Ketepatan dan penguasaan</p>	5%

	<p>Polymorphism</p> <ul style="list-style-type: none"> <li>• Mahasiswa memahami dan dapat mengimplementasikan inheritance dan Polymorphism dalam suatu program sederhana.</li> </ul>	<p>Polymorphism?</p> <ul style="list-style-type: none"> <li>• How inheritance promotes software reusability</li> <li>• Designing for inheritance</li> <li>• Superclass and subclass relationship &amp; access modifier</li> <li>• Method Overriding</li> <li>• Polymorphism: in object declaration &amp; assignment, in method argument &amp; return type</li> <li>• Method overloading</li> <li>• Constructor overloading (revisited) &amp; superclass constructor</li> <li>• Calls to super(), this ()</li> <li>• Sample class diagram representing inheritance &amp; polymorphism</li> </ul>	<p>[TM:2x(2x50")]</p> <p>Tugas :</p> <p>Merancang class diagram, flowchart aplikasi dan ER diagram (basis data) dan Mengimplementasikan class diagram ke dalam kode program</p>	<p>and Kathy Sierra, O'Reilly.</p>	<p>diagram kelas beserta makna simbol relasi yang terdapat pada diagram kelas</p>	<p><b>Bentuk :</b> Diskusi, Tanya jawab UTS</p>	
9-10	<p>Mahasiswa memahami perlunya dan mengimplementasikan interface dan abstract class dalam suatu program sederhana.</p>	<p>Interface &amp; abstract classes</p> <ul style="list-style-type: none"> <li>• Why do we need abstract classes? Abstract vs concrete class</li> <li>• Abstract classes and methods</li> <li>• Overloading and overriding (revisited)</li> <li>• Casting</li> <li>• Why do we need interface?</li> <li>• Interface: making &amp; using</li> <li>• Sample class diagram representing interface &amp; abstract class</li> </ul>	<p>Kuliah dan Diskusi (Luring)</p> <p>[TM:1x(2x50")]</p> <p>Tugas :</p> <p>Merancang aplikasi dan membangun kode program aplikasi berdasarkan class diagram yang dirancang</p>	<p>Head First Java, 2nd edition, 2008, Bert Bates and Kathy Sierra, O'Reilly</p>	<p>Mahasiswa Dapat menjelaskan mengenai interface dan abstract classes dalam suatu program sederhana</p>	<p><b>Kriteria :</b> Kesesuaian, kualitas presentasi dan sistematika <b>Bentuk :</b> Diskusi, Tanya jawab UTS</p>	5%

11-12	Mahasiswa memahami perlunya dan mengimplementasikan static & final variabel dan method dalam suatu program sederhana.	Static & final method & variables <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>• Example: Math class</li> <li>• Instance variable vs static variable</li> </ul>	Kuliah dan Diskusi (Daring) [TM:1x(2x50'')] Tugas : mengimplementasikan static & final variabel dan method dalam suatu program sederhana	Head First Java, 2nd edition, 2008, Bert Bates and Kathy Sierra, O'Reilly	Mahasiswa Dapat menjelaskan mengenai static & final variabel	<b>Kriteria :</b> Kesesuaian, kualitas presentasi dan sistematika <b>Bentuk :</b> Diskusi, Tanya jawab UTS	5%
13	UTS						
14	<ul style="list-style-type: none"> <li>• Mahasiswa mampu memahami dan menggunakan Exception Handling.</li> <li>• Mahasiswa dapat membuat kelas Exception sendiri.</li> </ul>	Exception handling <ul style="list-style-type: none"> <li>• Exception as an object</li> <li>• try-catch</li> <li>• throws</li> <li>• Try-catch- finally</li> <li>• Multiple exception</li> <li>• Polymorphism in exception</li> <li>• Making your own exception</li> </ul>	Kuliah dan Diskusi (Luring) [TM:1x(2x50'')] Tugas : membuat kelas Exception	Head First Java, 2nd edition, 2008, Bert Bates and Kathy Sierra, O'Reilly	• Mahasiswa Memahami Exception Handling.	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk :</b> Tugas UAS	5%
15	Mahasiswa mampu memahami dan mengimplementasikan Persistensi Objek dalam file.	Object Persistence <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>	Kuliah dan Diskusi (Daring) [TM:1x(2x50'')] Tugas : Menjelaskan Persistensi Objek dalam file	Head First Java, 2nd edition, 2008, Bert Bates and Kathy Sierra, O'Reilly	Mahasiswa memahami Persistensi Objek dalam file	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk :</b> Tugas UAS	5%
16	Mahasiswa mampu memahami dan	Multithreading <ul style="list-style-type: none"> <li>• Launching new thread</li> </ul>	Kuliah dan Diskusi (Luring)	Head First Java, 2nd edition,	Memahami dan mengimplement	<b>Kriteria :</b> Ketepatan dan	5%

	mengimplementasikan konsep sinkronisasi menggunakan thread	<ul style="list-style-type: none"> <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>	[TM:1x(2x50")] Tugas : Menjelaskan konsep sinkronisasi menggunakan thread	2008, Bert Bates and Kathy Sierra, O'Reilly	asikan konsep sinkronisasi menggunakan thread	penguasaan <b>Bentuk :</b> Tugas UAS	
17	Mahasiswa mampu menggunakan Java API untuk membantu membuat aplikasi.	Using java library (java API) <ul style="list-style-type: none"> <li>• How to use it</li> <li>• Simple example</li> </ul>	Kuliah dan Diskusi (Luring) [TM:1x(2x50")] Tugas : Membuat Aplikasi menggunakan Java API	Head First Java, 2nd edition, 2008, Bert Bates and Kathy Sierra, O'Reilly	Mampu Membuat Aplikasi menggunakan Java API	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk :</b> Tugas UAS	10%
18	Mahasiswa mampu menggunakan struktur data yang telah terdefinisi dalam Collections untuk membuat aplikasi	Collections <ul style="list-style-type: none"> <li>• Interface Collection and class Collections</li> <li>• Type wrapper classess for primitive types</li> <li>• Autoboxing and autounboxing</li> <li>• List: ArrayList - Iterator, LinkedList</li> <li>• Stack</li> <li>• Sets</li> <li>• Maps</li> </ul>	Kuliah dan Diskusi (Daring) [TM:1x(2x50")] Tugas : Membuat struktur data dalam Collections	Head First Java, 2nd edition, 2008, Bert Bates and Kathy Sierra, O'Reilly	mampu membuat struktur data yang telah terdefinisi dalam Collections untuk membuat aplikasi	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk :</b> Tugas UAS	10%
19	Mahasiswa mampu membuat koneksi dari aplikasi berbasis java ke DBMS	Making connection with database <ul style="list-style-type: none"> <li>• JDBC</li> <li>• Oracle/mysql (?)</li> </ul>	Kuliah dan Diskusi (Luring) [TM:1x(2x50")] Tugas : Mengkoneksikan Java ke DBMS	Head First Java, 2nd edition, 2008, Bert Bates and Kathy Sierra, O'Reilly	Mampu membuat koneksi dari aplikasi berbasis java ke DBMS	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk :</b> Tugas UAS	10%
20-22	Mahasiswa mampu membuat aplikasi berbasis GUI/Swing.	GUI & SWING <ul style="list-style-type: none"> <li>• User event</li> </ul>	Kuliah dan Diskusi (Daring) [TM:1x(2x50")]	Head First Java, 2nd edition, 2008, Bert Bates	Mampu membuat aplikasi berbasis	<b>Kriteria :</b> Ketepatan dan penguasaan	10%



		<ul style="list-style-type: none"> <li>• Listener interface</li> <li>• Action event</li> <li>• Inner class</li> <li>• Swing component</li> </ul>	Tugas : membuat aplikasi berbasis GUI/Swing	and Kathy Sierra, O'Reilly	GUI/Swing.	<b>Bentuk :</b> Tugas UAS	
23	Mahasiswa mampu mengorganisasikan dan men-deploy aplikasi	Deployment <ul style="list-style-type: none"> <li>• Separate source code and class files</li> <li>• Making an executable JAR</li> <li>• Running an executable JAR</li> <li>• Package</li> </ul>	Kuliah dan Diskusi (Luring) [TM: 1x(2x50")] Tugas : membuat aplikasi berbasis GUI/Swing	Head First Java, 2nd edition, 2008, Bert Bates and Kathy Sierra, O'Reilly	mampu mengorganisasi kan dan men-deploy aplikasi	<b>Kriteria :</b> Ketepatan dan penguasaan <b>Bentuk :</b> Tugas UAS	10%
24	<b>UAS</b>						

12. Pengalaman Belajar Mahasiswa (*Student Learning Experiences*)  
 Telah mempelajari pengetahuan dasar terkait Perograman Java

13. Kriteria dan Bobot Penilaian (*Criteria and Evaluation*)

a. Kriteria Penilaian:

CPL	CPMK	MBKM	Observasi (Praktek)	Unjuk Kerja (Presentasi)	Tugas	Tes Tertulis			Tes Lisan (Tgs Kel)
						Kuis	UTS	UAS	
CPL 01	CPMK011					√			
CPL 01	CPMK012					√			
CPL 02	CPMK021						√		
CPL 02	CPMK022		√					√	

CPL	CPMK	Tahap Penilaian	Teknik Penilaian	Instrumen	Kriteria	Bobot
CPL 01	CPMK011	Sebelum UTS	Tes Tertulis (Kuis)	Rubrik	Kelengkapan Jawaban	5%
CPL 01	CPMK012	Sebelum UTS	Tes Tertulis (Kuis)	Rubrik	Kelengkapan Jawaban	5%
CPL 02	CPMK021	Ujian Tengah Semester	Tes Tertulis (UTS)	Rubrik	Kelengkapan Jawaban	30%
CPL 02	CPMK022	Setelah UTS, Akhir Semester	Observasi (Praktek), Tes Tertulis (UAS)	Rubrik, Rubrik	Hasil Praktek, Kelengkapan Jawaban	15% 45%

CPL	CPMK	MBKM	Observasi (Praktek)	Unjuk Kerja (Presentasi)	Tugas	Tes Tertulis			Tes Lisan (Tgs Kel)	Total
						Kuis	UTS	UAS		
CPL 01	CPMK011					5				5
CPL 01	CPMK012					5				5
CPL 02	CPMK021						30			30
CPL 02	CPMK022		15					45		60
Jumlah Total MK										100

b. Rubrik Penilaian:

Kategori	CPMK	Model Soal	Indikator Penilaian				
			Sangat Kurang 0 - < 55	Kurang 55 - < 65	Cukup 65 - < 75	Baik 75 - < 85	Sangat Baik 85 - 100
UTS	CPMK011	<ul style="list-style-type: none"> <li>Menjelaskan perlunya OOP untuk menyelesaikan permasalahan</li> </ul>	<ul style="list-style-type: none"> <li>Mahasiswa tidak dapat Menjelaskan perlunya OOP untuk menyelesaikan permasalahan</li> </ul>	<ul style="list-style-type: none"> <li>Mahasiswa dapat Menjelaskan perlunya OOP untuk menyelesaikan permasalahan</li> </ul>	<ul style="list-style-type: none"> <li>Mahasiswa dapat menjelaskan secara singkat dan dapat memberikan contoh tentang perlunya OOP untuk menyelesaikan permasalahan</li> </ul>	<ul style="list-style-type: none"> <li>Mahasiswa dapat menjelaskan dengan baik dan dapat memberikan contoh yang lebih detail dan komprehensif tentang perlunya OOP untuk menyelesaikan permasalahan</li> </ul>	<ul style="list-style-type: none"> <li>Mahasiswa dapat menjelaskan dengan sangat baik dan dapat memberikan contoh yang lebih detail dan komprehensif tentang perlunya OOP untuk menyelesaikan permasalahan</li> </ul>

Kategori	CPMK	Model Soal	Indikator Penilaian				
			Sangat Kurang 0 - < 55	Kurang 55 - < 65	Cukup 65 - < 75	Baik 75 - < 85	Sangat Baik 85 - 100
	CPMK012	<ul style="list-style-type: none"> <li>Membuat class diagram dalam bahasa Java, memahami dan mengimplementasikan enkapsulasi, memahami diagram kelas beserta makna simbol relasi, mengimplementasikan interface dan abstract classess dalam suatu program sederhana, mengimplementasikan static &amp; final variabel dan method dalam suatu program sederhana</li> </ul>	<ul style="list-style-type: none"> <li>Mahasiswa tidak dapat membuat class diagram dalam bahasa Java, memahami dan mengimplementasikan enkapsulasi, memahami diagram kelas beserta makna simbol relasi, mengimplementasikan interface dan abstract classess dalam suatu program sederhana, mengimplementasikan static &amp; final variabel dan method dalam suatu program sederhana</li> </ul>	<ul style="list-style-type: none"> <li>Mahasiswa cukup mampu membuat class diagram dalam bahasa Java, memahami dan mengimplementasikan enkapsulasi, memahami diagram kelas beserta makna simbol relasi, mengimplementasikan interface dan abstract classess dalam suatu program sederhana, mengimplementasikan static &amp; final variabel dan method dalam suatu program sederhana</li> </ul>	<ul style="list-style-type: none"> <li>Mahasiswa dapat membuat class diagram dalam bahasa Java, memahami dan mengimplementasikan enkapsulasi, memahami diagram kelas beserta makna simbol relasi, mengimplementasikan interface dan abstract classess dalam suatu program sederhana, mengimplementasikan static &amp; final variabel dan method dalam suatu program sederhana</li> </ul>	<ul style="list-style-type: none"> <li>Mahasiswa dapat membuat class diagram dalam bahasa Java, memahami dan mengimplementasikan enkapsulasi, memahami diagram kelas beserta makna simbol relasi, mengimplementasikan interface dan abstract classess dalam suatu program sederhana, mengimplementasikan static &amp; final variabel dan method dalam suatu program sederhana dengan baik</li> </ul>	<ul style="list-style-type: none"> <li>Membuat class diagram dalam bahasa Java, memahami dan mengimplementasikan enkapsulasi, memahami diagram kelas beserta makna simbol relasi, mengimplementasikan interface dan abstract classess dalam suatu program sederhana, mengimplementasikan static &amp; final variabel dan method dalam suatu program sederhana dengan sangat baik</li> </ul>
UAS	CPMK021	Menjelaskan Exception Handling, Persistensi	Mahasiswa tidak dapat	Mahasiswa dapat	Mahasiswa dapat menjelaskan	Mahasiswa dapat menjelaskan	Mahasiswa dapat

Kategori	CPMK	Model Soal	Indikator Penilaian				
			Sangat Kurang 0 - < 55	Kurang 55 - < 65	Cukup 65 - < 75	Baik 75 - < 85	Sangat Baik 85 - 100
		Objek dalam file, konsep sinkronisasi menggunakan thread	Menjelaskan Exception Handling, Persistensi Objek dalam file, konsep sinkronisasi menggunakan thread	menjelaskan secara singkat dan tidak mampu memberikan contoh yang komprehensif tentang Exception Handling, Persistensi Objek dalam file, konsep sinkronisasi menggunakan thread	secara singkat dan cukup mampu memberikan contoh yang komprehensif tentang Exception Handling, Persistensi Objek dalam file, konsep sinkronisasi menggunakan thread	secara singkat dan mampu memberikan contoh yang komprehensif tentang Exception Handling, Persistensi Objek dalam file, konsep sinkronisasi menggunakan thread	menjelaskan secara detail dan mampu memberikan contoh yang komprehensif tentang contoh yang komprehensif tentang Exception Handling, Persistensi Objek dalam file, konsep sinkronisasi menggunakan thread
	CPMK022	Membuat aplikasi menggunakan Java API, struktur data yang telah terdefinisi dalam Collections, koneksi dari aplikasi berbasis java ke DBMS, aplikasi berbasis GUI/Swing, mengorganisasikan dan men-deploy aplikasi	Mahasiswa tidak dapat Membuat aplikasi menggunakan Java API, struktur data yang telah terdefinisi dalam Collections, koneksi dari	Mahasiswa dapat Membuat aplikasi menggunakan Java API, struktur data yang telah terdefinisi dalam Collections, koneksi dari	Mahasiswa cukup mampu Membuat aplikasi menggunakan Java API, struktur data yang telah terdefinisi dalam Collections, koneksi dari aplikasi berbasis java ke DBMS,	Mahasiswa dapat Membuat aplikasi menggunakan Java API, struktur data yang telah terdefinisi dalam Collections, koneksi dari aplikasi berbasis java ke DBMS,	Mahasiswa dapat Membuat aplikasi menggunakan Java API, struktur data yang telah terdefinisi dalam Collections, koneksi dari

Kategori	CPMK	Model Soal	Indikator Penilaian				
			Sangat Kurang 0 - < 55	Kurang 55 - < 65	Cukup 65 - < 75	Baik 75 - < 85	Sangat Baik 85 - 100
			aplikasi berbasis java ke DBMS, aplikasi berbasis GUI/Swing, mengorganisasi kan dan men-deploy aplikasi	aplikasi berbasis java ke DBMS, aplikasi berbasis GUI/Swing, mengorganisasi kan dan men-deploy aplikasi	aplikasi berbasis GUI/Swing, mengorganisasika n dan men-deploy aplikasi	aplikasi berbasis GUI/Swing, mengorganisasika n dan men-deploy aplikasi dengan baik	aplikasi berbasis java ke DBMS, aplikasi berbasis GUI/Swing, mengorganisasi kan dan men-deploy aplikasi dengan sangat baik

c. Bobot penilaian:

- $\geq 85$  = A
- $\geq 70$  s.d  $< 85$  = B
- $\geq 60$  s.d  $< 70$  = C
- $\geq 50$  s.d  $< 60$  = D
- $< 50$  = E

## 15. Buku Sumber (*References*)

[1] Head First Java, 2nd edition, 2008, Bert Bates and Kathy Sierra, O'Reilly