


**RPS (RENCANA PEMBELAJARAN SEMESTER)
RENCANA EVALUASI PROSES BELAJAR MENGAJAR**

**MATA KULIAH
PEMROGRAMAN BERORIENTASI OBJEK (JAVA)**



NAMA DOSEN :
PRODI TEKNIK INFORMATIKA

STMIK HANDAYANI MAKASSAR
TAHUN 2017

	FORMULIR	No. Dokumen	
		No. Revisi	
	SILABUS	Tanggal Berlaku	Maret 2017

SILABUS PEMBELAJARAN

Fakultas / Program Studi : Teknik Informatika

Mata Kuliah : PEMROGRAMAN BERORIENTASI OBJEK (JAVA)

Kode Mata Kuliah : KK021636

Semester : VI

SKS : 3

Capaian Pemb. Mt.Kuliah :

1. Mahasiswa mengetahui sejarah OOP.
2. Mahasiswa memahami perlunya OOP untuk menyelesaikan permasalahan, yang kurang tepat jika procedural oriented diterapkan.
3. Mahasiswa mengetahui bahasa pemrograman Java yang akan digunakan untuk mempelajari OOP Mahasiswa memahami perbedaan antara kelas dan object.
4. Mahasiswa mampu membuat kelas (sederhana) dan object dalam bahasa Java.
5. Mahasiswa memahami penggunaan serta dapat membuat program sederhana menggunakan tipe primitif, array, keyword, reference variable, deklarasi dan assignment objek,
6. Mahasiswa memahami dan mengimplementasikan dalam prograam tentang siklus objek (penggunaan constructor dan garbage collector) Mahasiswa memahami makna dan perlunya enkapsulasi.
7. Mahasiswa memahami dan dapat mengimplementasikan enkapsulasi dalam suatu program sederhana. Mahasiswa memahami diagram kelas beserta makna simbol relasi yang terdapat pada diagram kelas.
8. Mahasiswa memahami makna dan perlunya inheritance dan Polymorphism.
9. Mahasiswa memahami dan dapat mengimplementasikan inheritance Dan olymorphism dalam suatu Program sederhana.
10. Mahasiswa memahami perlunya dan mengimplementasikan interface dan abstract classess dalam suatu program sederhana.
11. Mahasiswa memahami perlunya dan mengimplementasikan static & final variabel dan method dalam suatu program sederhana.
12. Mahasiswa mampu memahami dan menggunakan Exception Handling, Object Persistence dan Multithreading.

13. Mahasiswa mampu menggunakan Java API untuk membantu membuat aplikasi.
14. Mahasiswa mampu menggunakan struktur data yang telah terdefinisi dalam Collections untuk membuat aplikasi.
15. Mahasiswa mampu membuat koneksi dari aplikasi berbasis java ke DBMS Mahasiswa mampu membuat aplikasi berbasis GUI/Swing.
16. Mahasiswa mampu mengorganisasikan dan men-deploy aplikasi.

Deskripsi 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.

Bahan Kajian :

1. Course Introduction
2. Classes and object
3. Encapsulation
4. Class relationship (class diagram)
5. Inheritance & polymorphism
6. Interface & abstract classes
7. Static & final method & variables
8. Exception handling
9. Object Persistence
10. Multithreading
11. Using java library (java API)
12. Collections
13. Making connection with database
14. GUI & SWING
15. Deployment

Referensi :

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
4. Pollice, David West, O'Reilly Media

Mengetahui,

Ketua Prodi Teknik Informatika

Billy Eden William Asrul, S.Kom.,MT



RENCANA PEMBELAJARAN SEMESTER

TEKNIK INFORMATIKA

STMIK HANDAYANI

MATA KULIAH		KODE	RUMPUN MK	BOBOT (sks)		SEMESTER
PEMROGRAMAN BERORIENTASI OBJEK (JAVA)		KK021636	TEKNIK INFORMATIKA	T=3	P=0	VI
CAPAIAN PEMBELAJARAN (CP)	Program Studi	1. Mampu mengidentifikasi, memformulasikan dan memecahkan permasalahan kebutuhan informasi dari suatu organisasi. 2. Mampu mengintegrasikan solusi berbasis teknologi informasi secara efektif pada suatu organisasi. 3. Mampu menerapkan konsep-konsep dasar komputer yang dibutuhkan dalam merancang dan mengimplementasikan solusi teknologi informasi. 4. Dapat berkarya dengan perilaku etika sesuai bidang keprofesian teknologi informasi.				
	Mata Kuliah	1. Mahasiswa mampu menerapkan konsep pemrograman berorientasi objek dan mengaplikasikannya dengan Java				
DISKRIPSI SINGKAT MK		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.				
PUSTAKA	Utama :	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 4. Pollice, David West, O'Reilly Media				
MEDIA PEMBELAJARAN	Software		Hardware			

				laptop, dan LCD projector.		
TEAM TEACHING						
MATAKULIAH SYARAT						
MG Ke-	CP-MK (SESUAI TAHAPAN BELAJAR)	MATERI PEMBELAJARAN (PUSTAKA)	METODE STRATEGI PEMBELAJARAN (ESTIMASI WAKTU)	ASSESSMENT		
				INDIKATOR	BENTUK	BOBOT
1	<ul style="list-style-type: none">• Mahasiswa mengetahui sejarah OOP.• Mahasiswa memahami perlunya OOP untuk menyelesaikan permasalahan, yang kurang tepat jika procedural oriented diterapkan.• Mahasiswa mengetahui bahasa pemrograman Java yang akan digunakan untuk mempelajari OOP	Course Introduction <ul style="list-style-type: none">• OOP history• compare procedural and oop by giving a simple case study• Introduction to java	Ceramah	<ul style="list-style-type: none">• Mampu menjelaskan Sejarah OOP dan memahami perlunya OOP untuk menyelesaikan permasalahan• Mampu menjelaskan bahasa pemrograman Java	Pemberian tugas terkait materi perkuliahan yang telah diberikan	
2, 3	<ul style="list-style-type: none">• Mahasiswa memahami perbedaan antara kelas dan object.	Classes and object <ul style="list-style-type: none">• The difference between class and object, by giving some examples	Ceramah Demo	<ul style="list-style-type: none">• Dapat menjelaskan perbedaan Kelas dan Object• Dapat membuat bahasa pemrograman Java yang sederhana	Melalui Tanya jawab dan diskusi mengenai materi kuliah dan pemberian tugas sesuai dengan materi kuliah yang telah diberikan	10%

	<ul style="list-style-type: none"> • Mahasiswa mampu membuat kelas (sederhana) dan object dalam bahasa Java. • Mahasiswa memahami penggunaan serta dapat membuat program sederhana menggunakan tipe primitif, array, keyword, reference variable, deklarasi dan assignment objek, • Mahasiswa memahami dan mengimplementasikan dalam program tentang siklus objek (penggunaan constructor dan garbage collector) 	<ul style="list-style-type: none"> • making a simple first object (Fokus: first programming with java) • Primitive and reference: Type, keywords, reference variable, object declaration & assignment, objects in garbage collector, arrays • Object cycle: life and death • Constructor & Garbage collector 		<ul style="list-style-type: none"> • Dapat memahami dan mengimplementasikan program tentang siklus objek 		
--	---	--	--	---	--	--

4	<ul style="list-style-type: none"> • Mahasiswa memahami makna dan perlunya enkapsulasi. • Mahasiswa memahami dan dapat mengimplementasikan enkapsulasi dalam suatu program sederhana. 	Encapsulation <ul style="list-style-type: none"> • Why do we need encapsulation? • Object state • Instance variable vs local variables • Methods argument & return type • Pass by value • Getters and setters • Sample class diagram representing encapsulation 	Ceramah Demo Practice	<ul style="list-style-type: none"> • Mampu menjelaskan enkapsulasi serta implementasinya 	Melalui Tanya jawab dan diskusi mengenai materi kuliah dan pemberian tugas sesuai dengan materi kuliah yang telah diberikan	10%
5, 6	Mahasiswa memahami diagram kelas beserta makna simbol relasi yang terdapat pada diagram kelas	Class relationship (class diagram) <ul style="list-style-type: none"> • Association, aggregation, composition • Example: simple class diagram & code in java related <p>*) Hanya mengenalkan saja, karena nantinya satu persatu relasi yg ada pada diagram kelas akan dibutuhkan saat menjelaskan materi</p>	Ceramah Demo Practice	Ketepatan dalam mengimplementasikan class diagram ke dalam kode program	Melalui Tanya jawab dan diskusi mengenai materi kuliah dan pemberian tugas sesuai dengan materi kuliah yang telah diberikan	15%

		inheritance, polimorfisme, kelas abstrak, interface				
7,8	<ul style="list-style-type: none"> • Mahasiswa memahami makna dan perlunya inheritance dan Polymorphism. • Mahasiswa memahami dan dapat mengimplementasikan inheritance Dan Polymorphism dalam suatu Program sederhana 	Inheritance & polymorphism <ul style="list-style-type: none"> • Why do we need inheritance & polymorphism? • How inheritance promotes software reusability • Designing for inheritance • Superclass and subclass relationship & access modifier • Method Overriding • Polymorphism: in object declaration & assignment, in method argument & return type • Method overloading • Constructor overloading (revisited) & superclass constructor 	Ceramah Demo Practice	Mampu menjelaskan dan mengimplementasikan Inheritance dan Polymorphism dalam suatu program sederhana	Melalui Tanya jawab dan diskusi mengenai materi kuliah dan pemberian tugas sesuai dengan materi kuliah yang telah diberikan	10%

		<ul style="list-style-type: none"> • Calls to super(), this () • Sample class diagram representing inheritance & polymorphism 				
9,10	Mahasiswa memahami perlunya dan mengimplementasikan interface dan abstract classess dalam suatu program sederhana.	Interface & abstract classes <ul style="list-style-type: none"> • Why do we need abstract classes? Abstract vs concrete class • Abstract classess and methods • Overloading and overriding (revisited) • Casting • Why do we need interface? • Interface: making & using • Sample class diagram representing interface & abstract class 	Demo Practice	Mampu menjelaskan dan mengimplementasikan Interface & abstract classes dalam suatu program sederhana	Melalui Tanya jawab dan diskusi mengenai materi kuliah dan pemberian tugas sesuai dengan materi kuliah yang telah diberikan	
11	Mahasiswa memahami perlunya dan mengimplementasikan static & final	Static & final method & variables <ul style="list-style-type: none"> • Static method • Static variable 	Ceramah Demo	Mampu menjelaskan dan mengimplementasikan Static & final method & variables dalam suatu program sederhana	Melalui Tanya jawab dan diskusi mengenai materi kuliah dan pemberian tugas sesuai dengan materi kuliah yang telah diberikan	

	variabel dan method dalam suatu program sederhana	<ul style="list-style-type: none"> • Constants (static final variable) • Final method & class • Wrapper classes & autoboxing • Static imports • Example: Math class • Instance variable vs static variable 				
12	EVALUASI TENGAH SEMESTER (EVALUASI FORMATIF-EVALUASI YG DIMAKSUDKAN UNTUK MELAKUKAN IMPROVEMENT PROSES PEMBELAJARAN BEDASARKAN ASSESSMENT YANG TELAH DILAKUKAN)					
13	<ul style="list-style-type: none"> • Mahasiswa mampu memahami dan menggunakan Exception Handling, , Object Persistence dan Multithreading 	<ul style="list-style-type: none"> • Exception handling • Object Persistence • Multithreading 	Ceramah Demo	Ketepatan dalam mengimplementasikan Exception handling, Object Persistence dan Multithreading	Melalui Tanya jawab dan diskusi mengenai materi kuliah dan pemberian tugas sesuai dengan materi kuliah yang telah diberikan	10%
14	<ul style="list-style-type: none"> • Mahasiswa mampu menggunakan Java API untuk membantu membuat aplikasi • Mahasiswa mampu menggunakan 	<ul style="list-style-type: none"> • Using java library (java API) • Collections • Making connection with Database 	Ceramah Demo	Ketepatan dalam mengimplementasikan Using java library (java API), Collections, Making connection with Database	Melalui Tanya jawab dan diskusi mengenai materi kuliah dan pemberian tugas sesuai dengan materi kuliah yang telah diberikan	10%

	struktur data yang telah terdefinisi dalam Collections untuk membuat aplikasi								
	<ul style="list-style-type: none"> Mahasiswa mampu membuat koneksi dari aplikasi berbasis java ke DBMS 								
15	<ul style="list-style-type: none"> Mahasiswa mampu membuat aplikasi berbasis GUI/Swing. Mahasiswa mampu mengorganisasikan dan men-deploy aplikasi 	<ul style="list-style-type: none"> GUI & SWING Deployment 	Ceramah Demo	Ketepatan dalam mengimplementasikan GUI & SWING dan Deployment	Melalui Tanya jawab dan diskusi mengenai materi kuliah dan pemberian tugas sesuai dengan materi kuliah yang telah diberikan	10%			
16	EVALUASI AKHIR SEMESTER (EVALUASI YANG DIMAKSUDKAN UNTUK MENGETAHUI CAPAIAN AKHIR HASIL BELAJAR MAHASISWA)								
Catatan : 1 sks = (50' TM + 50' PT + 60' BM)/ Minggu			BM = Belajar Mandiri				T = Teori (aspek ilmu pengetahuan)		
	TM = Tatap Muka (Kuliah)			PS = Praktikum Simulasi (160 menit/minggu)			P = Praktek (aspek keterampilan kerja)		
	PT = Penugasan Terstruktur.			PL = Praktikum laboratorium (160 menit/minggu)					

FORMAT RANCANGAN TUGAS QUIZ dan PR

Nama Mata Kuliah	: Pemrograman Berorientasi Objek (Java)	SKS	: 3 sks
Program Studi	: Teknik Informatika	Pertemuan ke	: 2,3,5,6,9,10,13,14,15

A. TUJUAN TUGAS:

Mahasiswa mampu menerapkan konsep pemrograman berorientasi objek dan mengaplikasikannya dengan Java, sesuai materi yang diberikan

B. URAIAN TUGAS:

a. Obyek Garapan :

16. Course Introduction
17. Classes and object
18. Encapsulation
19. Class relationship (class diagram)
20. Inheritance & polymorphism
21. Interface & abstract classes
22. Static & final method & variables
23. Exception handling
24. Object Persistence
25. Multithreading
26. Using java library (java API)
27. Collections
28. Making connection with database
29. GUI & SWING
30. Deployment

b. Yang harus dikerjakan dan batasan-batasan

1. Discovery Teoritis (Quis)

	Spesifikasi Tugas	Keterkaitan Tugas
1	Membaca literatur materi kuliah yang akan diberikan	Kesiapan Mhs Untuk meneirna materi baru
2	Mengingat materi kuliah minggu lalu	Pemahaman mahasiswa dari materi minggu lalu

2. Discovery Analisis (Mengerjakan soal pemahaman) PR

	Spesifikasi Tugas	Keterkaitan Tugas
1	Membaca literatur materi kuliah yang Sudah diberikan	Melengkapi materi yang belum diberikan di kelas
2	Memahami materi kuliah minggu lalu	Mengerjakan Soal pemahaman yang diberikan

c. Metode/Cara Pengerjaan (acuan cara/langkah pengerjaan)

1. Discovery Teoritis (Quis)

	Spesifikasi Tugas	Batasan Tugas
1	Dosen Menyampaikan Pertanyaan Quiz di awal pertemuan	Dapat Diselesaikan 20 menit
2	Mahasiswa Menjawab Pertanyaan	Dapat menjawab 80 %
3	Menerima hasil koreksi dari dosen	Disimpan oleh Mahasiswa

2. Discovery Analisis (Mengerjakan soal pemahaman - PR)

	Spesifikasi Tugas	Batasan Tugas
1	Diberikan pada setiap pokok bahasan	Maksimal 30 soal
2	Cara Pengerjaan	Mandiri
3	Cara pelaporan hasil Tugas	Mandiri, ditulis di buku kertas folio

d. Deskripsi Luaran tugas yang dihasilkan:

Pemahaman mahasiswa dari konsep teori dan analisis minimal 80 %

C.KRITERIA PENILAIAN

POLA PENILAIAN KOMPETENSI : Quis dan PR Soal

KRITERIA : Pemahaman Teori dan Analisis

DIMENSI	Sangat Memuaskan	Memuaskan	Batas	Kurang Memuaskan	Di bawah standard	SKOR
Skor	(Skor 100)	(Skor 80)	(Skor 60)	(Skor 40)	(Skor20)	
Quis	Mengerjakan seluruh soal dengan benar	Mengerjakan 80% soal dengan benar	Mengerjakan 60% soal dengan benar	Mengerjakan 40% soal dengan benar	Mengerjakan 20% soal dengan benar	30%
Tugas PR	Mengerjakan seluruh soal dengan benar	Mengerjakan 80% soal dengan benar	Mengerjakan 60% soal dengan benar	Mengerjakan 40% soal dengan benar	Mengerjakan 20% soal dengan benar	70%

D. Rubrik Penilaian

Jenjang/Grade	Angka/Skor	Deskripsi/Indikator Kerja
E	<=45	Merupakan perolehan mahasiswa yang tidak melaksanakan tugas dan sama sekali tidak memahami materi.
D	45-51	Merupakan perolehan mahasiswa yang mengikuti perkuliahan dan mengerjakan tugas seadanya, tidak memiliki kemauan dan tanggung jawab untuk memahami materi.
C	51.01-61	Merupakan perolehan mahasiswa yang mengikuti perkuliahan dengan cukup baik, berusaha memahami materi namun kurang persisten sehingga baru mampu menyelesaikan sebagian dari masalah / tugas dengan akurasi yang kurang.
C+	61.01-66	Merupakan perolehan mahasiswa yang mengikuti perkuliahan dengan baik, berusaha memahami materi namun baru mampu menyelesaikan sebagian masalah / tugas dengan akurasi cukup.
B-	66.01-71	Merupakan perolehan mahasiswa yang mengikuti perkuliahan dengan baik, mampu memahami materi dan mampu menyelesaikan masalah / tugas dengan akurasi cukup.
B	71.01-76	Merupakan perolehan mahasiswa yang mengikuti perkuliahan dengan baik, mampu memahami materi dan mampu menyelesaikan masalah / tugas dengan akurasi bagus.
B+	76.01-81	Merupakan perolehan mahasiswa yang mengikuti perkuliahan dengan baik, mampu memahami materi dan mampu menyelesaikan masalah / tugas dengan akurasi bagus.

- | | | |
|----|-----------|--|
| A- | 81.01-85 | Merupakan perolehan mahasiswa yang mengikuti perkuliahan dengan sangat baik, memahami materi dengan sangat baik, memiliki tingkat proaktif dan kreatifitas tinggi dalam mencari informasi terkait materi, mampu menyelesaikan masalah / tugas dengan akurasi sangat baik. |
| A | 85.01-100 | Merupakan perolehan mahasiswa superior, yaitu mereka yang mengikuti perkuliahan dengan sangat baik, memahami materi dengan sangat baik bahkan tertantang untuk memahami lebih jauh, memiliki tingkat proaktif dan kreatifitas tinggi dalam mencari informasi terkait materi, mampu menyelesaikan masalah dengan akurasi sempurna bahkan mampu mengenali masalah nyata pada masyarakat / industri dan mampu mengusulkan konsep solusinya. |

Mengetahui

Kaprodi Teknik Informatika

Billy Eden William Asrul,S.Kom,MT