

SILABUS MATAKULIAH

Revisi : 2
 Tanggal Berlaku : Maret 2014

A. Identitas

1. Nama Matakuliah : A11. 54405/ Rekayasa Perangkat Lunak
2. Program Studi : Teknik Informatika-S1
3. Fakultas : Ilmu Komputer
4. Bobot sks : 3 SKS
5. Elemen Kompetensi : MKK
6. Jenis Kompetensi : Kompetensi Dasar
7. Alokasi waktu total : 14 X 150 Menit

B. Unsur-unsur Silabus

Kompetensi Dasar	Indikator	Pokok Bahasan/Materi	Aktifitas Pembelajaran	Alokasi Waktu	Rujukan	Evaluasi
Mahasiswa mampu menjelaskan permasalahan umum perangkat lunak, dan mahasiswa mampu menjelaskan perbedaan antara software dan system engineering	Setelah mengikuti kuliah ini, mahasiswa akan dapat: <ol style="list-style-type: none"> 1. Mengetahui kewajiban dan hak berkaitan dengan proses perkuliahan 2. mengetahui tujuan perkuliahan dan referensi yang digunakan 3. menguraikan perkembangan ilmu rekayasa perangkat lunak 	Kontrak Belajar Deskripsi matakuliah, silabus, referensi Introduction to Software Engineering <ol style="list-style-type: none"> a. Perkembangan ilmu rekayasa perangkat lunak, b. Perkembangan perangkat lunak c. Permasalahan perangkat lunak d. Jenis Aplikasi Perangkat Lunak e. Mitos Perangkat Lunak f. Perbedaan Software, 	<ol style="list-style-type: none"> 1. Menjelaskan sejarah kemunculan rekayasa perangkat lunak 2. Menjelaskan perkembangan ilmu rekayasa perangkat lunak. 3. Menjelaskan isu dan mitos rekayasa perangkat lunak. 4. Menjelaskan perbedaan Software, Software Engineering, System Engineering, System Komputer 5. Melakukan tanya jawab pemahaman mahasiswa tentang cakupan kajian 	150 menit	1, 2	<ol style="list-style-type: none"> a. Melakukan tanya jawab pemahaman mahasiswa b. Memberi latihan di kelas. c. Memberi tugas kepada mahasiswa untuk dikerjakan di rumah.

Kompetensi Dasar	Indikator	Pokok Bahasan/Materi	Aktifitas Pembelajaran	Alokasi Waktu	Rujukan	Evaluasi
	4. memahami isu dan mitos rekayasa perangkat lunak	Software Engineering, System Engineering, System Komputer g. Isu dari Tanggungjawab Profesional	rekayasa perangkat lunak. 6. Memberi latihan di kelas.			
Mahasiswa mampu menjelaskan roadmap untuk membangun produk software berkualitas tinggi	Setelah mengikuti kuliah ini, mahasiswa akan dapat: 1. Memahami dan menjelaskan framework proses 2. mengidentifikasi komponen membangun rekayasa perangkat lunak 3. memahami langkah membangun software	Software Process a. Framework Proses b. CMMI (Capability Maturity Model Integration) c. Process Assessment d. Proses Model (Waterfall Model, Increment Model, RAD Model), Evolutionary Process Model (Prototyping Model, Spiral Mode) e. An Agile View of Processl	1. Menjelaskan Framework Proses 2. Mendefinisikan konsep CMMI (Capability Maturity Model Integration) 3. Memberikan contoh Process Assessment 4. Menjelaskan Proses Model (Waterfall Model, Increment Model, RAD Model), Evolutionary Process Model (Prototyping Model, Spiral Mode) 5. An Agile View of Process 6. Melakukan tanya jawab pemahaman mahasiswa 7. Memberi latihan di kelas.	2 x 150 menit	1	a. Melakukan tanya jawab pemahaman mahasiswa b. Memberi latihan di kelas. c. Memberi tugas kepada mahasiswa untuk dikerjakan di rumah.
Mahasiswa mampu menjelaskan prinsip, konsep, dan method dalam software Engineering.	Setelah mengikuti kuliah ini, mahasiswa akan dapat: 1. Mengidentifikasi langkah-langkah software practice 2. Memahami dan menjelaskan	Software Engineering Practice a. Core and the essence of practice Software Engineering b. Planning Practices c. Modeling Practices d. Construction Practices	1. Menjelaskan pentingnya practice Software Engineering 2. Menyebutkan langkah-langkah Planning Practices 3. Menjelaskan jenis Modeling Practices	150 menit	1, 2	a. Melakukan tanya jawab pemahaman mahasiswa b. Memberi latihan di kelas. c. Memberi tugas kepada mahasiswa untuk dikerjakan di rumah.

Kompetensi Dasar	Indikator	Pokok Bahasan/Materi	Aktifitas Pembelajaran	Alokasi Waktu	Rujukan	Evaluasi
	langkah membangun sistem model proses.	e. Computer-Based System Element f. System Engineering Hierarchy g. System Model Process	4. Construction Practices 5. Computer-Based System Element 6. System Engineering Hierarchy 7. Bagaimana membangun System Model Process 8. Melakukan tanya jawab pemahaman mahasiswa			
Mahasiswa dapat menjelaskan dan memahami keinginan customer, dapat menganalisis kebutuhan dari sebuah studi kasus sederhana.	Setelah mengikuti kuliah ini, mahasiswa akan dapat: 1. Memahami langkah menganalisis kebutuhan customer 2. Menggambarkan desain sistem dengan Use Case	Requirement Engineering a. Requirement Engineering Task (inception, elicitation, elaboration, negotiation, specification, validation, requirement management) b. Initiating The Requirements Engineering Process. c. Collaborative Requirements Gathering d. Developing Use Case	1. Menjelaskan konsep Requirement Engineering Task (inception, elicitation, elaboration, negotiation, specification, validation, requirement management) 2. Menjelaskan langkah Initiating The Requirements Engineering Process. 3. Collaborative Requirements Gathering 4. Memberikan contoh Developing Use Case 5. Melakukan tanya jawab pemahaman mahasiswa 6. Memberi latihan di kelas.	150 menit	1, 2	a. Melakukan tanya jawab pemahaman mahasiswa b. Memberi latihan di kelas. c. Memberi tugas kepada mahasiswa untuk dikerjakan di rumah.
Mahasiswa dapat membedakan antara pemodelan analisis, data, dan analisis objek	Setelah mengikuti kuliah ini, mahasiswa akan dapat: 1. Menganalisa kebutuhan	Analysis Modeling a. Requirement Analysis b. Analysis Modeling Approaches c. Data Modeling	1. Menjelaskan langkah-langkah Requirement Analysis 2. Menjelaskan Analysis Modeling Approaches	150 menit	1, 2	a. Melakukan tanya jawab pemahaman mahasiswa b. Memberi latihan di kelas. c. Memberi tugas kepada mahasiswa untuk

Kompetensi Dasar	Indikator	Pokok Bahasan/Materi	Aktifitas Pembelajaran	Alokasi Waktu	Rujukan	Evaluasi
oriented	customer 2. Melakukan pemodelan data 3. Melakukan analisa Object Oriented	Concepts d. Object Oriented Analysis	3. Menjelaskan Data Modeling Concepts 4. Memberikan contoh Object Oriented Analysis 5. Memberi latihan di kelas.			
Mahasiswa dapat memilih dan membuat model analisis yang efektif dan tepat	Setelah mengikuti kuliah ini, mahasiswa akan dapat: 1. menjelaskan dan menerapkan prosedur perancangan software 2. menjelaskan dan menerapkan prosedur pengembangan sistem	Desain Rekayasa perangkat lunak Analysis Modeling a. Scenario-Based Modeling b. Flow Oriented Modeling c. Class-Based Modeling d. Class-Responsibility (CRC) Modeling	1. Menjelaskan proses perancangan rekayasa perangkat lunak 2. Menjelaskan langkah membangun Scenario-Based Modeling 3. Menjelaskan Flow Oriented Modeling 4. Menjelaskan Class-Based Modeling 5. Class-Responsibility (CRC) Modeling 6. Memberi latihan di kelas.	150 menit	1, 2	a. Melakukan tanya jawab pemahaman mahasiswa b. Memberi latihan di kelas. c. Memberi tugas kepada mahasiswa untuk dikerjakan di rumah.
Ujian Tengah Semester						
Mahasiswa dapat merancang sebuah model atau representasi yang nyaman dan jelas.	Setelah mengikuti kuliah ini, mahasiswa akan dapat: 1. Menjelaskan konsep desain sistem 2. Menjelaskan langkah-langkah membangun model perancangan	Design Engineering a. Transformasi dari Model Analisis ke Model Perancangan b. Design Process and Design Quality c. Design Concepts (Abstract, Architecture, Modularity, Information Hiding, Functional Independence, Refinement)	1. Menjelaskan langkah Transformasi dari Model Analisis ke Model Perancangan 2. Menjelaskan konsep Design Process and Design Quality 3. Menjelaskan Design Concepts (Abstract, Architecture, Modularity, Information Hiding, Functional	150 menit	1, 2	a. Melakukan tanya jawab pemahaman mahasiswa b. Memberi latihan di kelas. c. Memberi tugas kepada mahasiswa untuk dikerjakan di rumah.

Kompetensi Dasar	Indikator	Pokok Bahasan/Materi	Aktifitas Pembelajaran	Alokasi Waktu	Rujukan	Evaluasi
		d. Design Model	Independence, Refinement) 4. Design Model 5. Melakukan tanya jawab pemahaman mahasiswa 6. Memberi latihan di kelas.			
Mahasiswa dapat memetakan requirement ke dalam perancangan arsitektur perangkat lunak	Setelah mengikuti kuliah ini, mahasiswa akan dapat: 1. menjelaskan konsep, method dalam desain arsitektur 2. mengidentifikasi kebutuhan untuk membangun arsitektur sistem 3. Merancang arsitektur sistem	Creating An Architecture Design a. Software Architecture b. Data design c. Architecture Styles and Pattern d. Architecture Design e. Mapping Data Flow into a Software Architecture	1. Menjelaskan Software Architecture 2. Mendefinisikan Data design 3. Menjelaskan Architecture Styles and Pattern 4. Konsep Architecture Design 5. Mapping Data Flow into a Software Architecture 6. Memberi latihan di kelas.	150 menit	1, 2	a. Melakukan tanya jawab pemahaman mahasiswa b. Memberi latihan di kelas. c. Memberi tugas kepada mahasiswa untuk dikerjakan di rumah.
Mahasiswa dapat merancang interface perangkat lunak level menengah	Setelah mengikuti kuliah ini, mahasiswa akan dapat: 1. Memahami dan menjelaskan konsep Golden Rules 2. Merancang interface perangkat lunak	Interface Design a. The Golden Rules b. User Interface Analysis and Design c. Interface Analysis d. Interface Design Steps	1. Menjelaskan konsep The Golden Rules 2. Mendefinisikan User Interface Analysis and Design 3. Melakukan langkah Interface Analysis 4. Memberi contoh Interface Design Steps 5. Melakukan tanya jawab pemahaman mahasiswa 6. Memberi latihan di kelas.	150 menit	1, 2	a. Melakukan tanya jawab pemahaman mahasiswa b. Memberi latihan di kelas. c. Memberi tugas kepada mahasiswa untuk dikerjakan di rumah.

Kompetensi Dasar	Indikator	Pokok Bahasan/Materi	Aktifitas Pembelajaran	Alokasi Waktu	Rujukan	Evaluasi
Mahasiswa dapat merencanakan dengan baik tahapan dari pembuatan perangkat lunak, yaitu dengan mampu membuat road map pembuatan s/w (bagian dari testing)	Setelah mengikuti kuliah ini, mahasiswa akan dapat: <ol style="list-style-type: none"> menerapkan strategi untuk software testing melakukan validasi terhadap sistem 	Testing Strategi <ol style="list-style-type: none"> Strategy Approach to Software Testing Test Strategy for Conventional Software Test Strategy for Object-Oriented Software Validation Testing System Testing 	<ol style="list-style-type: none"> Pendekatan untuk Software Testing Menjelaskan jenis-jenis strategi untuk testing software Menjelaskan langkah Validation Testing Menjelaskan langkah System Testing Memberi contoh studi kasus Melakukan tanya jawab pemahaman mahasiswa Memberi latihan di kelas. 	2 x 150 menit	1, 2	<ol style="list-style-type: none"> Melakukan tanya jawab pemahaman mahasiswa Memberi latihan di kelas. Memberi tugas kepada mahasiswa untuk dikerjakan di rumah.
Mahasiswa dapat membangun sebuah realisasi Use case yang memodelkan kolaborasi antara instans kelas yang telah diidentifikasi	Setelah mengikuti kuliah ini, mahasiswa akan dapat: <ol style="list-style-type: none"> Menyebutkan dan menjelaskan langkah-langkah testing software menerapkan software testing dalam contoh-contoh kasus 	Testing Tactic <ol style="list-style-type: none"> Software Testing Fundamentals Black-Box and White-Box Object-Oriented Testing Methods 	<ol style="list-style-type: none"> Menjelaskan Software Testing Fundamentals Menjelaskan langkah Black-Box and White-Box testing Menjelaskan Object-Oriented Testing Methods Melakukan tanya jawab pemahaman mahasiswa Memberi latihan di kelas. 	2 x 150 menit	1, 2	<ol style="list-style-type: none"> Melakukan tanya jawab pemahaman mahasiswa Memberi latihan di kelas. Memberi tugas kepada mahasiswa untuk dikerjakan di rumah.
Ujian Akhir Semester						

Daftar Referensi

Wajib :

1. Roger S. Pressman, "Software Engineering A Practitioner's Approach", 7th edition, McGraw-Hill International Edition, tahun 2010.
2. Ian Sommerville, "Software Engineering", 8th edition, China Machine Press. 2006

Disusun oleh :	Diperiksa oleh :		Disahkan oleh :
Dosen Pengampu	Penanggungjawab Keilmuan	Program Studi	Dekan
Ayu Pertiwi, S.Kom, MT	Ayu Pertiwi, S.Kom, MT	Heru Agus Santoso , Ph. D	DR. Drs. Abdul Syukur, MM