

	RENCANA PEMBELAJARAN SEMESTER PROGRAM STUDI S1 SISTEM INFORMASI FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS IBNU SINA						
Identitas Mata Kuliah	NAMA MK	KODE MK	RUMPUN MATA KULIAH	BOBOT(SKS)		SEMESTER	Direvisi
	Rekayasa Perangkat Lunak	4SIMKK334	Teknik	3	SKS	4	23/08/2025
Otoritas	Pengembang RPS			Ketua Kelompok Keahlian		Ka PRODI	
	TIM Microteaching			TIM Microteaching		Dr. Army Trilidia D, S.Kom,M.Pd.T	
Deskripsi Mata Kuliah	Mata kuliah Rekayasa Perangkat Lunak memberikan pemahaman komprehensif tentang prinsip, metodologi, dan praktik terbaik dalam pengembangan perangkat lunak yang sistematis, terstandarisasi, dan terukur. Mata kuliah ini mencakup konsep dasar RPL dan krisis perangkat lunak, model proses perangkat lunak (Waterfall, Agile, Scrum, Kanban, XP), rekayasa kebutuhan (requirements engineering), pemodelan perangkat lunak dengan UML, arsitektur perangkat lunak (layered, microservices, MVC), desain perangkat lunak (design principles, pattern), implementasi dan quality assurance, pengujian perangkat lunak (unit, integration, system, acceptance testing), pemeliharaan, manajemen konfigurasi, manajemen proyek perangkat lunak, serta tren RPL modern (DevOps, CI/CD). Mahasiswa diharapkan mampu menerapkan praktik RPL profesional dalam pengembangan sistem informasi. The Software Engineering course provides a comprehensive understanding of the principles, methodologies, and best practices in systematic, standardized, and scalable software development. This course covers the basic concepts of RPL and software crisis, software process models (Waterfall, Agile, Scrum, Kanban, XP), requirements engineering, software modeling with UML, software architecture (layered, microservices, MVC), software design (design principles, patterns), implementation and quality assurance, software testing (unit, integration, system, acceptance testing), maintenance, configuration management, software project management, and modern RPL trends (DevOps, CI/CD). Students are expected to be able to apply professional RPL practices in information systems development.						
Capaian Pembelajaran Lulusan & Capaian Pembelajaran Mata Kuliah	Capaian Pembelajaran Lulusan (CPL) PRODI						
	CPL04	Memiliki kompetensi untuk menganalisis persoalan computing yang kompleks untuk mengidentifikasi solusi pengelolaan proyek teknologi bidang informatika/ilmu komputer dengan mempertimbangkan wawasan perkembangan ilmu transdisiplin.					
	CPL05	Menguasai konsep teoritis bidang pengetahuan Ilmu Komputer/Informatika dalam mendesain dan mensimulasikan aplikasi teknologi multi-platform yang relevan dengan kebutuhan industri dan masyarakat.					

	CPL08	Kemampuan mengimplementasi kebutuhan computing dengan mempertimbangkan berbagai metode/algorithm yang sesuai.					
	CPL10	Kemampuan mendesain, mengimplementasi dan mengevaluasi solusi berbasis computing multi-platform yang memenuhi kebutuhan computing pada sebuah organisasi.					
	Capaian Pembelajaran Mata Kuliah (CPMK)					CPL yang di dukung	
	CPMK 133	Mahasiswa mampu memahami konsep dasar RPL, model proses perangkat lunak, dan melakukan rekayasa kebutuhan secara sistematis.				CPL04,CPL05	
	CPMK 134	Mahasiswa mampu melakukan pemodelan dan perancangan arsitektur perangkat lunak menggunakan UML dan design principles.				CPL04,CPL05,CPL10	
	CPMK 135	Mahasiswa mampu merancang dan melaksanakan strategi pengujian perangkat lunak yang komprehensif untuk menjamin kualitas produk SI.				CPL05,CPL08,CPL10	
	CPMK 136	Mahasiswa mampu menerapkan manajemen konfigurasi, DevOps, CI/CD, dan praktik RPL modern dalam pengembangan sistem informasi profesional.				CPL04,CPL05,CPL08,CPL10	
Penilaian	Id CPMK	Bobot per Bentuk Penilaian					TOTAL BOBOT PER CPMK
		Tugas 1	Tugas 2	Tugas 3	Proyek 1	Proyek 2	
	CPMK 133	20%	10%	-	-	-	30%
	CPMK 134	-	10%	10%	5%	-	25%
	CPMK 135	-	-	10%	5%	-	15%
	CPMK 136	-	-	-	-	15%	15%
	-	-	-	-	5%	10%	15%
	Total per penilaian	20%	20%	20%	15%	25%	100%
Pustaka	Utama:						
	1. Pressman, R. S., & Maxim, B. R. (2021). Software Engineering: A Practitioner's Approach (9th ed.). New York: McGraw-Hill. 2. Sommerville, I. (2022). Software Engineering (10th ed.). Hoboken: Pearson Education. 3. Beck, K. (2020). Extreme Programming Explained: Embrace Change (2nd ed.). Boston: Addison-Wesley. 4. Schwaber, K., & Sutherland, J. (2020). The Scrum Guide. Scrum.org.						
	Pustaka Pendukung:						
	1. Martin, R. C. (2020). Clean Code. Boston: Prentice Hall. 2. Fowler, M. (2021). Refactoring: Improving the Design of Existing Code (2nd ed.). Boston: Addison-Wesley. 3. Kim, G., Humble, J., Debois, P., & Willis, J. (2021). The DevOps Handbook (2nd ed.). Portland: IT Revolution Press. 4. Bass, L., Clements, P., & Kazman, R. (2022). Software Architecture in Practice (4th ed.). Boston: Addison-Wesley.						
Media Pembelajaran	Software:					Hardware :	

	StarUML / draw.io; Git & GitHub / GitLab; Jenkins / GitHub Actions; JIRA / Trello; VS Code; JUnit / pytest; SonarQube; Platform E-Learning (LMS); Zoom/Google Meet		Komputer/Laptop (min. RAM 8GB); LCD Projector; Whiteboard; Akses Internet
Team Teaching	TIM Microteaching		
Matakuliah Syarat			
Ambang Batas Kelulusan Mahasiswa	50.01		
Ambang Batas Kelulusan MK	85.00%		

Minggu Ke-	Kemampuan akhir tiap tahapan belajar (Sub- CPMK)	Penilaian		Bentuk Pembelajaran; Metode Pembelajaran; Penugasan Mahasiswa; [Estimasi Waktu]		Materi Pembelajaran [Pustaka]	Bobot Penilaian (%)
		Indikator	Kriteria & Teknik				
(1)	(2)	(3)	(4)	Luring(5)	Daring(6)	(7)	(8)
1	Mahasiswa mampu memahami konsep dasar RPL, krisis perangkat lunak, dan karakteristik software yang baik (CPMK 133)	1.1 Mendefinisikan rekayasa perangkat lunak dan pentingnya 1.2 Menjelaskan krisis perangkat lunak dan penyebabnya 1.3 Mengidentifikasi karakteristik perangkat lunak berkualitas	Keaktifan diskusi Ketepatan pemahaman konsep dasar RPL	Ceramah; Diskusi kasus kegagalan software; Demo JIRA [TM: 3x50']	Video pengantar RPL; Kuis konsep dasar [BM: 3x60']	Definisi RPL; Krisis Perangkat Lunak; Karakteristik Software Berkualitas; Lapisan RPL: Proses, Metode, Tools [Pustaka Utama 1, 2]	-

		1.4 Memahami lapisan RPL: proses, metode, tools, quality focus					
2	Mahasiswa mampu memahami dan membandingkan model proses perangkat lunak (CPMK 133)	2.1 Menjelaskan model Waterfall dan kelemahan-kelebihannya 2.2 Menjelaskan model Iteratif, Spiral, dan Prototyping 2.3 Menjelaskan prinsip-prinsip Agile Manifesto 2.4 Membandingkan pendekatan plan-driven dan agile	Kualitas perbandingan model proses Kemampuan memilih model yang tepat untuk kasus	Ceramah; Diskusi perbandingan model; Studi kasus [TM: 3x50']	Video model proses; Kuis Agile Manifesto [BM: 3x60']	Model Waterfall, Iteratif, Spiral, Prototyping; Agile Manifesto; Plan-Driven vs Agile; Pemilihan Model [Pustaka Utama 1, 2]	-
3	Mahasiswa mampu menerapkan Scrum sebagai framework Agile dalam pengembangan SI (CPMK 133)	3.1 Menjelaskan peran Scrum: Product Owner, Scrum Master, Dev Team 3.2 Memahami artefak Scrum: Product Backlog, Sprint Backlog, Increment 3.3 Menjelaskan event Scrum: Sprint, Daily Standup, Review, Retrospective 3.4 mempraktikkan Sprint Planning dan estimasi dengan Story Points	Tugas 1: Simulasi Sprint Planning dengan Scrum Rubrik: product backlog, story points, sprint planning, laporan	Ceramah; Workshop Scrum; Simulasi Sprint Planning JIRA [TM: 3x50']	Tugas 1: Simulasi Scrum Sprint Planning [BM: 3x60']	Scrum Framework; Peran Scrum; Artefak: Backlog & Increment; Event Scrum; Sprint Planning & Story Points [Pustaka Utama 4, Pendukung 2]	Tugas 1 (20%)
4	Mahasiswa mampu melakukan rekayasa kebutuhan yang sistematis (CPMK 133, CPMK 134)	4.1 Melakukan elisitasi kebutuhan dengan berbagai teknik 4.2 Menganalisis dan memodelkan kebutuhan dengan use case 4.3 Mengelola kebutuhan: traceability matrix 4.4 Menyusun Software Requirements Specification (SRS)	Kualitas dokumen kebutuhan dan requirements traceability Ketepatan use case dan SRS	Ceramah; Workshop elisitasi; Demo traceability [TM: 3x50']	Video requirements engineering; Latihan traceability [BM: 3x60']	Elisitasi Kebutuhan; Use Case Modeling; Requirements Traceability Matrix; Software Requirements Specification [Pustaka Utama 1, 2]	-

5	Mahasiswa mampu merancang arsitektur perangkat lunak (CPMK 134)	5.1 Menjelaskan peran arsitektur dalam RPL 5.2 Menganalisis gaya arsitektur: layered, microservices, event-driven 5.3 Merancang arsitektur MVC dan Clean Architecture 5.4 Mendokumentasikan arsitektur menggunakan C4 Model	Kualitas desain arsitektur perangkat lunak Kemampuan mendokumentasikan dengan C4 Model	Ceramah; Workshop arsitektur; Demo C4 Model [TM: 3x50']	Video arsitektur PL; Latihan C4 Model [BM: 3x60']	Gaya Arsitektur: Layered, Microservices, Event-Driven; MVC & Clean Architecture; C4 Model [Pustaka Utama 2, Pendukung 4]	-
6	Mahasiswa mampu menerapkan prinsip desain perangkat lunak (CPMK 134)	6.1 Menerapkan prinsip SOLID dalam desain komponen 6.2 Memahami prinsip DRY, KISS, dan YAGNI 6.3 Menggunakan design pattern: Creational, Structural, Behavioral 6.4 Melakukan refactoring untuk meningkatkan kualitas kode	Tugas 2: Desain perangkat lunak dengan prinsip SOLID dan design pattern Rubrik: penerapan SOLID, pattern, refactoring, dokumentasi	Ceramah; Workshop SOLID & pattern; Demo refactoring [TM: 3x50']	Tugas 2: Desain dengan SOLID & Design Pattern [BM: 3x60']	SOLID Principles; DRY, KISS, YAGNI; Design Pattern dalam RPL; Refactoring Techniques [Pustaka Utama 1, Pendukung 1, 2]	Tugas 2 (10%)
7	Mahasiswa mampu merancang strategi pengujian perangkat lunak (CPMK 135)	7.1 Menjelaskan tingkatan pengujian: unit, integration, system, acceptance 7.2 Membuat test plan dan test case 7.3 Memahami teknik black-box testing: equivalence partitioning, boundary value 7.4 Memahami teknik white-box testing: statement dan branch coverage	Kualitas test plan dan test case Kemampuan membuat test case black-box dan white-box	Ceramah; Workshop test plan; Latihan test case [TM: 3x50']	Video strategi testing; Persiapan UTS [BM: 3x60']	Tingkatan Pengujian; Test Plan & Test Case; Black-Box: Equivalence, Boundary Value; White-Box: Coverage [Pustaka Utama 1, 2]	Tugas 3 (10%)
8	UJIAN TENGAH SEMESTER (UTS) Evaluasi CPMK 133, CPMK 134, dan	Ketepatan jawaban soal UTS mencakup materi Pertemuan 1-7	Ujian tulis terjadwal Rubrik: model proses, Scrum, kebutuhan, arsitektur, desain,	Ujian Tulis [TM: 3x50']	-	Review komprehensif materi Pertemuan 1 s.d. 7	Proyek 1 (5%)

	CPMK 135		strategi testing				
9	Mahasiswa mampu melaksanakan pengujian unit dan integrasi menggunakan framework (CPMK 135)	9.1 Menulis unit test dengan JUnit 5 / pytest 9.2 Mengimplementasikan Test-Driven Development (TDD) 9.3 Melakukan integration testing pada komponen SI 9.4 Mengukur code coverage dan kualitas test suite	Kualitas unit test dan code coverage Kemampuan menerapkan TDD secara disiplin	Ceramah; Live coding TDD; Workshop integration testing [TM: 3x50']	Tutorial JUnit/pytest; Latihan TDD [BM: 3x60']	Unit Testing dengan JUnit 5/pytest; TDD: Red-Green-Refactor; Integration Testing; Code Coverage; Test Quality [Pustaka Utama 1, Pendukung 1]	-
10	Mahasiswa mampu melakukan system testing, performance testing, dan UAT (CPMK 135)	10.1 Melakukan functional system testing 10.2 Melakukan performance testing dasar dengan JMeter 10.3 Memahami User Acceptance Testing (UAT) 10.4 Menggunakan Selenium untuk automated UI testing dasar	Tugas 3 (lanjutan): System testing dan UAT planning Rubrik: test case system testing, UAT plan, laporan testing	Ceramah; Workshop JMeter & UAT; Demo Selenium [TM: 3x50']	Tugas 3: System testing & UAT [BM: 3x60']	Functional System Testing; Performance Testing dengan JMeter; UAT Planning; Selenium UI Testing Dasar [Pustaka Utama 1, 2]	Tugas 3 (10%)
11	Mahasiswa mampu menerapkan manajemen konfigurasi dengan Git (CPMK 136)	11.1 Memahami konsep manajemen konfigurasi perangkat lunak 11.2 Menggunakan Git untuk version control dan branching strategy 11.3 Menerapkan GitFlow atau GitHub Flow dalam tim 11.4 Menggunakan pull request dan code review workflow	Kualitas penerapan Git workflow dalam tim Kemampuan melakukan code review yang efektif	Ceramah; Workshop GitFlow; Demo pull request & review [TM: 3x50']	Tutorial GitFlow; Latihan code review [BM: 3x60']	Manajemen Konfigurasi PL; Git Branching Strategy; GitFlow & GitHub Flow; Pull Request & Code Review [Pustaka Utama 2, Pendukung 3]	-
12	Mahasiswa mampu memahami dan menerapkan Continuous Integration dan	12.1 Menjelaskan konsep DevOps dan CI/CD pipeline 12.2 Mengkonfigurasi CI pipeline dengan GitHub	Kualitas konfigurasi CI/CD pipeline Kemampuan mengotomasi build dan testing	Ceramah; Workshop GitHub Actions; Demo CI/CD pipeline [TM: 3x50']	Tutorial GitHub Actions; Latihan CI pipeline [BM: 3x60']	Konsep DevOps; CI/CD Pipeline; GitHub Actions; Build & Test Otomatis; Deployment; IaC Dasar [Pustaka Utama 2,	-

	Continuous Deployment (CPMK 136)	Actions 12.3 Mengotomasi build, test, dan deployment 12.4 Memahami konsep Infrastructure as Code (IaC)				Pendukung 3]	
13	Mahasiswa mampu melakukan code quality analysis dan software metrics (CPMK 136)	13.1 Memahami metrik perangkat lunak: LOC, cyclomatic complexity, coupling 13.2 Menggunakan SonarQube untuk analisis kualitas kode 13.3 Memahami technical debt dan cara mengelolanya 13.4 Menerapkan code review checklist yang efektif	Kualitas analisis kode dengan SonarQube Kemampuan mengidentifikasi dan mengelola technical debt	Ceramah; Demo SonarQube; Workshop code metrics [TM: 3x50']	Tutorial SonarQube; Latihan analisis kualitas kode [BM: 3x60']	Metrik PL: LOC, Cyclomatic Complexity; SonarQube Analysis; Technical Debt; Code Review Checklist [Pustaka Utama 1, Pendukung 1]	-
14	Mahasiswa mampu memahami pemeliharaan dan evolusi perangkat lunak (CPMK 136)	14.1 Mengklasifikasikan jenis pemeliharaan: corrective, adaptive, perfective, preventive 14.2 Memahami legacy system dan strategi modernisasi 14.3 Menerapkan teknik refactoring untuk pemeliharaan 14.4 Memahami open source software dan lisensi PL	Kualitas analisis pemeliharaan dan strategi modernisasi Kemampuan menerapkan refactoring	Ceramah; Workshop refactoring; Studi kasus legacy system [TM: 3x50']	Video pemeliharaan PL; Latihan refactoring [BM: 3x60']	Jenis Pemeliharaan PL; Legacy System & Modernisasi; Refactoring untuk Pemeliharaan; Open Source & Lisensi [Pustaka Utama 2, Pendukung 2]	-
15	Mahasiswa mampu menerapkan seluruh praktik RPL dalam proyek pengembangan SI (CPMK 133-4)	15.1 Menerapkan Scrum untuk mengelola proyek SI tim 15.2 Mengimplementasikan CI/CD pipeline untuk proyek 15.3 Memastikan kualitas melalui testing dan SonarQube 15.4 Mempresentasikan	Proyek 2: Proyek RPL SI tim (Scrum + CI/CD + Testing + Kualitas) Rubrik: proses Scrum, CI/CD, test coverage, kualitas kode, presentasi	Demo & presentasi proyek RPL; Code review final [TM: 3x50']	Upload repository & laporan; Peer assessment [BM: 3x60']	Integrasi: Scrum, Arsitektur, SOLID, Design Pattern, Testing, Git, CI/CD, SonarQube dalam Proyek SI Tim [Pustaka Utama 1, 2, 4]	Proyek 2 (25%)

		proyek akhir RPL secara profesional					
16	UJIAN AKHIR SEMESTER (UAS) Evaluasi CPMK 133, CPMK 134, CPMK 135, dan CPMK 136	Ketepatan jawaban UAS mencakup materi Pertemuan 9-15	Ujian tulis terjadwal Rubrik: TDD, CI/CD, kualitas kode, pemeliharaan, DevOps	Ujian Tulis [TM: 3x50']	-	Review komprehensif materi Pertemuan 9 s.d. 15	Proyek 2 (sudah dihitung)