



Universitas Negeri Surabaya
Fakultas Teknik
Program Studi S1 Teknik Informatika

Kode Dokumen

RENCANA PEMBELAJARAN SEMESTER

		Minggu Ke																
		CPMK	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CPMK-1																		
CPMK-2																		
CPMK-3																		
CPMK-4																		
CPMK-5																		
CPMK-6																		
CPMK-7																		
CPMK-8																		
CPMK-9																		
CPMK-10																		
CPMK-11																		
CPMK-12																		
Deskripsi Singkat MK		Mata kuliah Analisis dan Desain Perangkat Lunak membahas konsep, prinsip, metode, dan teknik untuk menganalisis kebutuhan serta merancang perangkat lunak sesuai dengan kebutuhan pengguna. Mahasiswa akan mempelajari analisis kebutuhan fungsional dan non-fungsional, pemodelan sistem dengan UML, perancangan arsitektur, basis data, antarmuka pengguna, serta penerapan pola desain. Pada akhir perkuliahan, mahasiswa mampu menghasilkan dokumen analisis dan desain serta mengembangkan sistem sebagai bukti kesesuaian antara rancangan dengan perangkat lunak yang dibangun.																
Pustaka		Utama :																
		1. Pressman, Roger S., Sofware Engineering: A Practitioner's Approach, 5th edition, McGraw-Hill Internasional, 2001 2. Sommerville, Ian, software engineering, 7th Addison Wesley Publishing Company, 2003 3. Langer, A. M. (2016). Analysis and Design of Information Systems (3rd ed.). Springer. 4. Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2012). Systems Analysis and Design in a Changing World (6th ed.). Cengage Learning.																
		Pendukung :																
		1. Pressman, Roger S., Sofware Engineering: A Practitioner's Approach, 5th edition, McGraw-Hill Internasional, 2001 2. Sommerville, Ian, software engineering, 7th Addison Wesley Publishing Company, 2003																
Dosen Pengampu																		
Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Penilaian				Bantuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa, [Estimasi Waktu]				Materi Pembelajaran [Pustaka]				Bobot Penilaian (%)				
		Indikator	Kriteria & Bentuk	Luring (offline)	Daring (online)													
(1)	(2)	(3)	(4)	(5)	(6)									(7)	(8)			
1	Mahasiswa mampu memahami dan menjelaskan ruang lingkup, konsep dasar, serta prosedur penyusunan kontrak kuliah dan pengantar analisis serta desain perangkat lunak.	1.Mahasiswa mampu menjelaskan kembali isi dan tujuan kontrak kuliah (meliputi aturan, penilaian, dan tanggung jawab mahasiswa serta dosen). 2.Mahasiswa mampu menguraikan konsep dasar analisis dan desain perangkat lunak, termasuk pengertian, tujuan, dan perannya dalam rekayasa perangkat lunak. Mahasiswa mampu mengidentifikasi ruang lingkup kegiatan analisis dan desain pada fase pengembangan perangkat lunak.	Kriteria: 1.Keaktifan 2.Tugas Bentuk Penilaian : Aktifitas Partisipatif	Ceramah 3x50	3x60					Materi: Pengantar Analisis dan Desain Perangkat Lunak Pustaka: Pressman, Roger S., Sofware Engineering: A Practitioner's Approach, 5th edition, McGraw-Hill Internasional, 2001				2%				

2	<p>1.Mahasiswa mampu mengidentifikasi jenis kebutuhan perangkat lunak, meliputi kebutuhan fungsional dan non-fungsional.</p> <p>2.Mahasiswa mampu mengklasifikasikan level kebutuhan perangkat lunak (tingkat pengguna, sistem, dan perangkat lunak)</p> <p>3.Mahasiswa mampu menyusun rancangan awal dokumen Software Requirement Specification (SRS) sesuai standar rekayasa perangkat lunak.</p>	<p>1.Mahasiswa mampu menjelaskan perbedaan antara kebutuhan fungsional dan non-fungsional secara tepat</p> <p>2.Mahasiswa mampu menentukan level kebutuhan perangkat lunak (user, system, software) dalam studi kasus yang diberikan</p> <p>3.Mahasiswa mampu merancang dan menyusun dokumen SRS yang mencakup kebutuhan fungsional utama sistem</p>	<p>Kriteria: Penilaian tugas</p> <p>Bentuk Penilaian : Aktifitas Partisipatif, Penilaian Hasil Project / Penilaian Produk</p>	Ceramah dan Diskusi 3x50	3x60	<p>Materi: Jenis dan level kebutuhan fungsional dan merencanakan software requirement system</p> <p>Pustaka: Langer, A. M. (2016). <i>Analysis and Design of Information Systems</i> (3rd ed.). Springer.</p>	5%
3	<p>1.Mahasiswa mampu menjelaskan konsep dasar pemodelan perangkat lunak dan manfaatnya dalam proses analisis sistem.</p> <p>2.Mahasiswa mampu mengidentifikasi aktor dan use case berdasarkan kebutuhan pengguna.</p> <p>3.Mahasiswa mampu menyusun use case scenario yang menggambarkan interaksi antara aktor dan sistem.</p> <p>4.Mahasiswa mampu membuat use case diagram sesuai standar UML untuk merepresentasikan kebutuhan fungsional sistem.</p>	<p>1.Mahasiswa mampu menjelaskan konsep dan tujuan pemodelan perangkat lunak dengan benar</p> <p>2.Mahasiswa mampu mengidentifikasi aktor dan use case dari studi kasus yang diberikan</p> <p>3.Mahasiswa mampu menyusun skenario use case yang logis dan lengkap</p> <p>4.Mahasiswa mampu menggambar use case diagram sesuai aturan UML</p> <p>5.Mahasiswa aktif berdiskusi dan memberikan analisis terhadap model yang dibuat</p>	<p>Kriteria: Penilaian tugas</p> <p>Bentuk Penilaian : Aktifitas Partisipatif, Penilaian Portofolio, Praktik / Unjuk Kerja</p>	Ceramah dan Diskusi 3x50		<p>Materi: Pemodelan perangkat lunak serta konsep use case scenario dan use case diagram</p> <p>Pustaka: Langer, A. M. (2016). <i>Analysis and Design of Information Systems</i> (3rd ed.). Springer.</p>	5%
4	<p>1.Mahasiswa mampu menginterpretasikan kebutuhan pengguna ke dalam use case scenario yang terstruktur dan konsisten.</p> <p>2.Mahasiswa mampu mengimplementasikan hasil analisis ke dalam use case diagram menggunakan perangkat bantu pemodelan (misalnya: StarUML, Visual Paradigm, atau draw.io)</p> <p>3.Mahasiswa mampu mengevaluasi kesesuaian antara use case scenario dan use case diagram untuk memastikan kelengkapan dan keterpaduan model.</p>	<p>1.Mahasiswa mampu menerjemahkan kebutuhan pengguna menjadi skenario use case yang runtut dan jelas</p> <p>2.Mahasiswa mampu membuat use case diagram dengan perangkat bantu sesuai notasi UML</p> <p>3.Mahasiswa mampu menghubungkan skenario dengan elemen diagram secara konsisten</p> <p>4.Mahasiswa mampu mempresentasikan hasil implementasi model dengan argumentasi logis</p> <p>5.Mahasiswa menunjukkan kemampuan teknis dalam penggunaan tools pemodelan</p>	<p>Kriteria: Penilaian tugas</p> <p>Bentuk Penilaian : Aktifitas Partisipatif</p>	Presentasi 3x50		<p>Materi: pemodelan perangkat lunak menggunakan use case scenario dan use case diagram</p> <p>Pustaka: Langer, A. M. (2016). <i>Analysis and Design of Information Systems</i> (3rd ed.). Springer.</p>	5%

5	<p>1.Mahasiswa mampu menjelaskan konsep dasar dan tujuan pemodelan proses menggunakan Activity Diagram dan DFD.</p> <p>2.Mahasiswa mampu mengidentifikasi aktivitas, aliran data, entitas, dan proses utama dalam sistem.</p> <p>3.Mahasiswa mampu menganalisis keterkaitan antara Activity Diagram dan DFD dalam menggambarkan alur logika dan data sistem.</p> <p>4.Mahasiswa mampu membuat model awal Activity Diagram dan DFD sederhana berdasarkan studi kasus.</p>	<p>1.Mahasiswa mampu menjelaskan konsep dan fungsi Activity Diagram serta DFD dengan benar</p> <p>2.Mahasiswa mampu mengidentifikasi proses, data store, entitas, dan aliran data dari studi kasus</p> <p>3.Mahasiswa mampu menggambarkan hubungan antara aktivitas sistem dan aliran data secara logis</p> <p>4.Mahasiswa mampu membuat Activity Diagram sesuai notasi UML dan DFD sesuai aturan Gane-Sarson atau Yourdon</p> <p>5.Mahasiswa aktif berdiskusi dalam menganalisis model proses sistem</p>	<p>Kriteria: Penilaian tugas</p> <p>Bentuk Penilaian : Aktifitas Partisipatif, Praktik / Unjuk Kerja</p>	Presentasi 3x50		<p>Materi: pemodelan perangkat lunak activity and data flow diagram</p> <p>Pustaka: Langer, A. M. (2016). <i>Analysis and Design of Information Systems</i> (3rd ed.). Springer.</p>	5%
6	<p>1.Mahasiswa mampu menjelaskan perbedaan dan keterkaitan antara conceptual, logical, dan physical data model.</p> <p>2.Mahasiswa mampu mengidentifikasi entitas, atribut, dan relasi utama dalam sistem berdasarkan kebutuhan data.</p> <p>3.Mahasiswa mampu membuat Entity Relationship Diagram (ERD) sebagai dasar conceptual data model.</p> <p>4.Mahasiswa mampu menurunkan conceptual model menjadi logical dan physical model dengan memperhatikan integritas data dan normalisasi.</p> <p>5.Mahasiswa mampu menyusun Data Dictionary yang menjelaskan struktur data secara rinci untuk setiap entitas dan atribut.</p>	<p>1.Mahasiswa mampu menjelaskan konsep dan perbedaan antara conceptual, logical, dan physical data model</p> <p>2.Mahasiswa mampu mengidentifikasi entitas, atribut, dan relasi dengan benar dari studi kasus</p> <p>3.Mahasiswa mampu menggambar ERD sesuai notasi standar (Chen, Crow's Foot, atau UML Class Diagram)</p> <p>4.Mahasiswa mampu menurunkan ERD menjadi logical dan physical data model yang konsisten dan terstruktur</p> <p>5.Mahasiswa mampu menyusun Data Dictionary yang lengkap dan konsisten dengan model data</p> <p>6.Mahasiswa berpartisipasi aktif dalam diskusi dan review model data</p>	<p>Kriteria: Penilaian tugas</p> <p>Bentuk Penilaian : Aktifitas Partisipatif, Penilaian Hasil Project / Penilaian Produk, Penilaian Portofolio, Praktik / Unjuk Kerja</p>	Presentasi 3x50		<p>Materi: pemodelan data menggunakan conceptual, logical, dan physical data modeling berdasarkan entity relationship model dan data dictionary</p> <p>Pustaka: Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2012). <i>Systems Analysis and Design in a Changing World</i> (6th ed.). Cengage Learning.</p>	5%
7	<p>1.Mahasiswa mampu menjelaskan struktur, tujuan, dan komponen utama dokumen SRS dan SDD sesuai standar IEEE.</p> <p>2.Mahasiswa mampu mengorganisasikan hasil analisis kebutuhan ke dalam dokumen SRS yang sistematis dan terukur.</p> <p>3.Mahasiswa mampu merancang dan mendokumentasikan desain perangkat lunak dalam format SDD, mencakup arsitektur, modul, antarmuka, serta diagram pendukung.</p> <p>4.Mahasiswa mampu memastikan konsistensi antara dokumen SRS dan SDD sebagai acuan pengembangan perangkat lunak.</p>	<p>1.Mahasiswa mampu menjelaskan isi dan fungsi setiap bagian dalam dokumen SRS dan SDD</p> <p>2.Mahasiswa mampu menyusun dokumen SRS berdasarkan hasil analisis kebutuhan fungsional dan nonfungsional</p> <p>3.Mahasiswa mampu membuat dokumen SDD berdasarkan hasil perancangan sistem</p> <p>4.Mahasiswa mampu menjaga keterkaitan antara SRS dan SDD (traceability)</p> <p>5.Mahasiswa menunjukkan kemampuan dokumentasi teknis dengan format dan tata bahasa profesional</p>	<p>Kriteria: Penilaian tugas</p> <p>Bentuk Penilaian : Aktifitas Partisipatif, Penilaian Hasil Project / Penilaian Produk</p>	Presentasi 3x50		<p>Materi: dokumentasi perangkat lunak berupa Software Requirement System dan Software Design Description</p> <p>Pustaka: Pressman, Roger S., <i>Software Engineering: A Practitioner's Approach</i>, 5th edition, McGraw-Hill Internasional, 2001</p>	5%

8	UTS / Ujian Tengah Semester: Evaluasi formatif yang dimaksudkan untuk melakukan improvement proses pembelajaran berdasarkan assessment yang telah dilakukan	Mampu menjawab dengan benar evaluasi yang diberikan	Kriteria: Ujian Tulis Bentuk Penilaian : Aktifitas Partisipatif, Tes	Ujian Tulis		Materi: UTS Pustaka: Pressman, Roger S., Software Engineering: A Practitioner's Approach, 5th edition, McGraw-Hill Internasional, 2001 Materi: Evaluasi formatif yang dimaksudkan untuk melakukan improvement proses pembelajaran Pustaka: Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2012). Systems Analysis and Design in a Changing World (6th ed.). Cengage Learning.	15%
9	1.Mahasiswa mampu menjelaskan konsep, tujuan, dan pola umum arsitektur perangkat lunak (seperti layered, MVC, client-server, dan microservices). 2.Mahasiswa mampu menganalisis kebutuhan sistem untuk menentukan gaya arsitektur yang sesuai. 3.Mahasiswa mampu merancang diagram arsitektur perangkat lunak yang menggambarkan komponen, relasi, dan aliran data antar modul. 4.Mahasiswa mampu merancang antarmuka pengguna (user interface) berdasarkan prinsip usability dan konsistensi desain. 5.Mahasiswa mampu mengintegrasikan desain arsitektur dan antarmuka dalam dokumen Software Design Description (SDD).	1.Mahasiswa mampu menjelaskan konsep dan pola arsitektur perangkat lunak dengan benar 2.Mahasiswa mampu menentukan gaya arsitektur yang sesuai dengan karakteristik sistem 3.Mahasiswa mampu membuat diagram arsitektur yang lengkap dan konsisten 4.Mahasiswa mampu mendesain antarmuka pengguna yang fungsional dan sesuai prinsip usability 5.Mahasiswa mampu mendokumentasikan hasil desain arsitektur dan antarmuka ke dalam SDD	Kriteria: Penilaian tugas Bentuk Penilaian : Aktifitas Partisipatif, Penilaian Hasil Project / Penilaian Produk, Penilaian Portofolio	Ceramah dan Presentasi		Materi: arsitektur perangkat lunak dan antar muka perangkat lunak Pustaka: Langer, A. M. (2016). Analysis and Design of Information Systems (3rd ed.). Springer.	5%
10	1.Mahasiswa mampu menjelaskan isi, tujuan, dan poin-poin utama dalam dokumen Software Design Description (SDD) kepada pihak non-teknis (customer). 2.Mahasiswa mampu menyajikan hasil desain perangkat lunak secara jelas dan terstruktur dalam bentuk presentasi atau demo desain. 3.Mahasiswa mampu menanggapi umpan balik dan pertanyaan dari customer secara profesional dan argumentatif. 4.Mahasiswa mampu menyusun berita acara atau form persetujuan desain (design approval form) berdasarkan hasil komunikasi dan revisi desain.	1.Mahasiswa mampu menjelaskan isi dan struktur dokumen SDD dengan bahasa yang mudah dipahami oleh customer 2.Mahasiswa mampu mempresentasikan hasil desain menggunakan media visual (diagram, mockup, arsitektur) secara profesional 3.Mahasiswa mampu menerima dan menanggapi masukan dari customer secara terbuka dan konstruktif 4.Mahasiswa mampu memperbarui desain sesuai hasil kesepakatan dengan customer 5.Mahasiswa mampu menyusun form persetujuan desain atau berita acara komunikasi desain	Kriteria: Penilaian tugas Bentuk Penilaian : Penilaian Hasil Project / Penilaian Produk, Penilaian Portofolio, Praktik / Unjuk Kerja	Studi Kasus		Materi: komunikasi terkait kesepakatan desain yang sudah dirancang dalam software design description kepada customer Pustaka: Pressman, Roger S., Software Engineering: A Practitioner's Approach, 5th edition, McGraw-Hill Internasional, 2001	15%

11	<p>1.Mampu menyiapkan lingkungan pengembangan (development environment) sesuai kebutuhan proyek.</p> <p>2.Mampu menerjemahkan desain arsitektur dan antarmuka ke dalam kode program.</p> <p>3.Mampu menerapkan prinsip clean code dan software engineering best practices.</p> <p>4.Mampu melakukan unit testing dan integration testing terhadap modul perangkat lunak.</p> <p>5.Mampu mendemonstrasikan perangkat lunak yang berfungsi sesuai kebutuhan pengguna.</p>	<p>1.Instalasi dan konfigurasi tools berjalan dengan benar</p> <p>2.Struktur kode sesuai rancangan arsitektur (layer, modul, atau komponen)</p> <p>3.Kode mudah dibaca, terdokumentasi, dan mengikuti konvensi penamaan.</p> <p>4.Hasil pengujian menunjukkan kesesuaian dengan spesifikasi.</p> <p>5.Fungsi utama aplikasi berjalan sesuai SRS.</p>	<p>Kriteria: Penilaian tugas</p> <p>Bentuk Penilaian : Aktifitas Partisipatif, Penilaian Hasil Project / Penilaian Produk</p>	Ceramah dan Penugasan		<p>Materi: mengimplementasikan perangkat lunak sesuai dengan perancangan perangkat lunak yang sudah dibuat</p> <p>Pustaka: Pressman, Roger S., Software Engineering: A Practitioner's Approach, 5th edition, McGraw-Hill Internasional, 2001</p>	6%
12	<p>1.Mampu menyiapkan lingkungan pengembangan (development environment) sesuai kebutuhan proyek.</p> <p>2.Mampu menerjemahkan desain arsitektur dan antarmuka ke dalam kode program.</p> <p>3.Mampu menerapkan prinsip clean code dan software engineering best practices.</p> <p>4.Mampu melakukan unit testing dan integration testing terhadap modul perangkat lunak.</p> <p>5.Mampu mendemonstrasikan perangkat lunak yang berfungsi sesuai kebutuhan pengguna.</p>	<p>1.Instalasi dan konfigurasi tools berjalan dengan benar</p> <p>2.Struktur kode sesuai rancangan arsitektur (layer, modul, atau komponen)</p> <p>3.Kode mudah dibaca, terdokumentasi, dan mengikuti konvensi penamaan.</p> <p>4.Hasil pengujian menunjukkan kesesuaian dengan spesifikasi.</p> <p>5.Fungsi utama aplikasi berjalan sesuai SRS.</p>	<p>Kriteria: Penilaian tugas</p> <p>Bentuk Penilaian : Aktifitas Partisipatif, Penilaian Hasil Project / Penilaian Produk</p>	Ceramah dan Penugasan		<p>Materi: mengimplementasikan perangkat lunak sesuai dengan perancangan perangkat lunak yang sudah dibuat</p> <p>Pustaka: Pressman, Roger S., Software Engineering: A Practitioner's Approach, 5th edition, McGraw-Hill Internasional, 2001</p>	6%
13	<p>1.Mampu menyiapkan lingkungan pengembangan (development environment) sesuai kebutuhan proyek.</p> <p>2.Mampu menerjemahkan desain arsitektur dan antarmuka ke dalam kode program.</p> <p>3.Mampu menerapkan prinsip clean code dan software engineering best practices.</p> <p>4.Mampu melakukan unit testing dan integration testing terhadap modul perangkat lunak.</p> <p>5.Mampu mendemonstrasikan perangkat lunak yang berfungsi sesuai kebutuhan pengguna.</p>	<p>1.Instalasi dan konfigurasi tools berjalan dengan benar</p> <p>2.Struktur kode sesuai rancangan arsitektur (layer, modul, atau komponen)</p> <p>3.Kode mudah dibaca, terdokumentasi, dan mengikuti konvensi penamaan.</p> <p>4.Hasil pengujian menunjukkan kesesuaian dengan spesifikasi.</p> <p>5.Fungsi utama aplikasi berjalan sesuai SRS.</p>	<p>Kriteria: Penilaian tugas</p> <p>Bentuk Penilaian : Aktifitas Partisipatif, Penilaian Hasil Project / Penilaian Produk</p>	Ceramah dan Penugasan		<p>Materi: mengimplementasikan perangkat lunak sesuai dengan perancangan perangkat lunak yang sudah dibuat</p> <p>Pustaka: Pressman, Roger S., Software Engineering: A Practitioner's Approach, 5th edition, McGraw-Hill Internasional, 2001</p>	6%
14	<p>1.Mampu menyusun materi presentasi proyek perangkat lunak secara sistematis dan informatif.</p> <p>2.Mampu menyampaikan presentasi dengan komunikasi yang baik dan profesional.</p> <p>3.Mampu mendemonstrasikan perangkat lunak hasil proyek dengan efektif.</p> <p>4.Mampu bekerja sama dan berkontribusi aktif dalam tim selama proses presentasi.</p>	<p>1.Struktur presentasi mencakup latar belakang, tujuan, metodologi, desain, implementasi, dan hasil.</p> <p>2.Gaya bicara jelas, percaya diri, dan mudah dipahami.</p> <p>3.Aplikasi dapat dijalankan dengan baik saat demo.</p> <p>4.Pembagian peran antar anggota tim proporsional dan jelas.</p>	<p>Kriteria: 5</p> <p>Bentuk Penilaian : Aktifitas Partisipatif, Penilaian Hasil Project / Penilaian Produk</p>	Presentasi		<p>Materi: mempresentasikan hasil proyek perangkat lunak</p> <p>Pustaka: Pressman, Roger S., Software Engineering: A Practitioner's Approach, 5th edition, McGraw-Hill Internasional, 2001</p>	5%

15	<p>1.Mampu menyusun materi presentasi proyek perangkat lunak secara sistematis dan informatif.</p> <p>2.Mampu menyampaikan presentasi dengan komunikasi yang baik dan profesional.</p> <p>3.Mampu mendemonstrasikan perangkat lunak hasil proyek dengan efektif.</p> <p>4.Mampu bekerja sama dan berkontribusi aktif dalam tim selama proses presentasi.</p> <p>5.Mampu melakukan pengujian perangkat lunak dengan beragam metode pengujian</p>	<p>1.Struktur presentasi mencakup latar belakang, tujuan, metodologi, desain, implementasi, dan hasil.</p> <p>2.Gaya bicara jelas, percaya diri, dan mudah dipahami.</p> <p>3.Aplikasi dapat dijalankan dengan baik saat demo.</p> <p>4.Pembagian peran antar anggota tim proporsional dan jelas.</p>	<p>Kriteria: 5</p> <p>Bentuk Penilaian : Aktifitas Partisipatif, Penilaian Hasil Project / Penilaian Produk</p>	Presentasi		<p>Materi: mempresentasikan hasil proyek perangkat lunak</p> <p>Pustaka: Pressman, Roger S., Software Engineering: A Practitioner's Approach, 5th edition, McGraw-Hill Internasional, 2001</p>	5%
16	<p>1.Mampu menyusun materi presentasi proyek perangkat lunak secara sistematis dan informatif.</p> <p>2.Mampu menyampaikan presentasi dengan komunikasi yang baik dan profesional.</p> <p>3.Mampu mendemonstrasikan perangkat lunak hasil proyek dengan efektif.</p> <p>4.Mampu bekerja sama dan berkontribusi aktif dalam tim selama proses presentasi.</p>	<p>1.Struktur presentasi mencakup latar belakang, tujuan, metodologi, desain, implementasi, dan hasil.</p> <p>2.Gaya bicara jelas, percaya diri, dan mudah dipahami.</p> <p>3.Aplikasi dapat dijalankan dengan baik saat demo.</p> <p>4.Pembagian peran antar anggota tim proporsional dan jelas.</p>	<p>Kriteria: 5</p> <p>Bentuk Penilaian : Aktifitas Partisipatif, Penilaian Hasil Project / Penilaian Produk</p>	Presentasi		<p>Materi: mempresentasikan hasil proyek perangkat lunak</p> <p>Pustaka: Pressman, Roger S., Software Engineering: A Practitioner's Approach, 5th edition, McGraw-Hill Internasional, 2001</p>	5%

Rekap Persentase Evaluasi : Project Based Learning

No	Evaluasi	Percentase
1.	Aktifitas Partisipatif	43.09%
2.	Penilaian Hasil Project / Penilaian Produk	29.42%
3.	Penilaian Portofolio	9.59%
4.	Praktik / Unjuk Kerja	10.42%
5.	Tes	7.5%
		100%

Catatan

- Capaian Pembelajaran Lulusan Prodi (CPL - Prodi)** adalah kemampuan yang dimiliki oleh setiap lulusan prodi yang merupakan internalisasi dari sikap, penguasaan pengetahuan dan ketrampilan sesuai dengan jenjang prodinya yang diperoleh melalui proses pembelajaran.
- CPL yang dibebankan pada mata kuliah** adalah beberapa capaian pembelajaran lulusan program studi (CPL-Prodi) yang digunakan untuk pembentukan/pengembangan sebuah mata kuliah yang terdiri dari aspek sikap, ketrampilan umum, ketrampilan khusus dan pengetahuan.
- CP Mata Kuliah (CPMK)** adalah kemampuan yang dijabarkan secara spesifik dari CPL yang dibebankan pada mata kuliah, dan bersifat spesifik terhadap bahan kajian atau materi pembelajaran mata kuliah tersebut.
- Sub-CPMK Mata Kuliah (Sub-CPMK)** adalah kemampuan yang dijabarkan secara spesifik dari CPMK yang dapat diukur atau diamati dan merupakan kemampuan akhir yang direncanakan pada tiap tahap pembelajaran, dan bersifat spesifik terhadap materi pembelajaran mata kuliah tersebut.
- Indikator penilaian** kemampuan dalam proses maupun hasil belajar mahasiswa adalah pernyataan spesifik dan terukur yang mengidentifikasi kemampuan atau kinerja hasil belajar mahasiswa yang disertai bukti-bukti.
- Kreteria Penilaian** adalah patokan yang digunakan sebagai ukuran atau tolok ukur ketercapaian pembelajaran dalam penilaian berdasarkan indikator-indikator yang telah ditetapkan. Kreteria penilaian merupakan pedoman bagi penilai agar penilaian konsisten dan tidak bias. Kreteria dapat berupa kuantitatif ataupun kualitatif.
- Bentuk penilaian:** tes dan non-tes.
- Bentuk pembelajaran:** Kuliah, Responsi, Tutorial, Seminar atau yang setara, Praktikum, Praktik Studio, Praktik Bengkel, Praktik Lapangan, Penelitian, Pengabdian Kepada Masyarakat dan/atau bentuk pembelajaran lain yang setara.
- Metode Pembelajaran:** Small Group Discussion, Role-Play & Simulation, Discovery Learning, Self-Directed Learning, Cooperative Learning, Collaborative Learning, Contextual Learning, Project Based Learning, dan metode lainnya yg setara.
- Materi Pembelajaran** adalah rincian atau uraian dari bahan kajian yg dapat disajikan dalam bentuk beberapa pokok dan sub-pokok bahasan.
- Bobot penilaian** adalah prosentasi penilaian terhadap setiap pencapaian sub-CPMK yang besarnya proposisional dengan tingkat kesulitan pencapaian sub-CPMK tsb., dan totalnya 100%.
- TM=Tatap Muka, PT=Penugasan terstruktur, BM=Belajar mandiri.

