

**RENCANA PEMBELAJARAN SEMESTER**

Mata Kuliah	Kode MK	Rumpun MK	Semester	Bobot (sks)	Tgl Penyusunan
Pemrograman Berbasis Kerangka Kerja	IFC31F3	Pemrograman Terapan	P5	T=3 P=	1 Desember 2022
<b>OTORISASI</b>	<b>Pengembang RPS</b>		<b>Koordinator RMK</b>		<b>Ketua Prodi</b>
					
	Daud Muhajir, S.Kom., M.Kom.		Ahmad Wali Satria Bahari Johan, S.S.T., M.Kom.		Muhammad Dzulfikar Fauzi, S.Kom., M.Cs.
<b>Capaian Pembelajaran (CP)</b>	<b>CPL Prodi (Kode S, P, KU, KK)</b>				
	P.4	Menguasai konsep dan prinsip-prinsip algoritma dan pemrograman			
	KU.2	Mampu bekerja secara mandiri dan bekerjasama dalam tim yang interdisiplin dan multidisiplin			
	KU.3	Mampu berkomunikasi secara efektif, kritis, dan dapat dipertanggungjawabkan			
	KK.4	Mampu merancang dan menganalisa algoritma untuk menyelesaikan permasalahan secara efektif dan efisien menggunakan kaidah-kaidah pemrograman dan bahasa pemrograman yang sesuai			
	<b>CPMK (Kode M)</b>				
	M1	Mahasiswa mampu menerapkan konsep frontend-backend dalam membangun aplikasi (P.4)			
	M2	Mahasiswa mampu membangun aplikasi menggunakan kerangka kerja modern (P.4)			
	M3	Mahasiswa mampu mengimplementasikan konsep frontend-backend dalam membangun aplikasi menggunakan kerangka kerja untuk menyelesaikan suatu permasalahan baik secara individual maupun kerjasama tim (KU.2, KU.3, KK.4)			
	<b>SUB-CPMK (Kode L)</b>				
	L1	Mahasiswa mampu menjelaskan mengenai pengetahuan dan konsep kerangka kerja dalam membuat aplikasi. (M1)			
	L2	Mahasiswa mampu menerapkan React sebagai dasar pemrograman frontend. (M1)			
	L3	Mahasiswa mampu membuat aplikasi frontend menggunakan React (M2)			
	L4	Mahasiswa mampu membuat aplikasi frontend menggunakan NextJS (M2)			
	L5	Mahasiswa mampu membuat aplikasi backend menggunakan ExpressJS dan memanfaatkannya pada frontend (M1)			
	L6	Mahasiswa mampu menerapkan NestJS sebagai dasar pemrograman backend.			
	L7	Mahasiswa mampu membuat project NestJS sebagai backend (M2)			
	L8	Mahasiswa mampu membangun aplikasi menggunakan kerangka kerja untuk menyelesaikan suatu permasalahan. (M3)			
<b>Deskripsi Singkat Mata Kuliah</b>	<b>Deskripsi</b>				
	Mata kuliah ini mempelajari tentang bagaimana membuat aplikasi web menggunakan framework dengan konsep frontend-backend. Framework ReactJS/NextJS digunakan untuk frontend, dan framework expressJS/NestJS digunakan untuk backend. Nantinya mahasiswa akan dapat membuat aplikasi web untuk menyelesaikan permasalahan baik secara individu maupun kelompok.				
<b>Materi Pembelajaran/Pokok Bahasan</b>	<b>Bahan Kajian</b>				
	Pemrograman Berbasis Kerangka Kerja				
	<b>Topik Bahasan</b>				
	1. Pengenalan Kerangka Kerja (framework)				
	1.1. Pengenalan framework				
	1.2. Mengapa menggunakan framework				
	1.3. Framework Frontend dan Backend				
	1.4. Perbedaan framework dan library?				
	1.5. Konsep frontend-backend				
	2. Pengenalan React Tingkat Dasar				
	2.1. Pengertian React				
	2.2. Sejarah dan Perkembangan React				
	2.3. EcmaScript untuk React				
	2.4. JSX (Javascript XML)				
	2.5. React Component				
	2.6. React Props, Event dan State				
	3. Pengenalan React Tingkat Lanjut				
	3.1. React Hooks				
	3.2. React Form Processing				
	3.3. React DOM Manipulation				
	3.4. Siklus Hidup Komponen React				
	4. Studi Kasus : Create React App CRUD React				
	4.1. Create React App				
	4.2. Create pada State React				
	4.3. Read pada State React				
	4.4. Update pada State React				
	4.5. Delete pada State React				
	4.6. Build React App				
	5. NextJS Tingkat Dasar				
	5.1. Pengenalan NextJS				
	5.2. Buat Project dan Instalasi				
	5.3. Server Component				
	5.4. Struktur Folder NextJS				
	5.5. Routing, Layout & Page				
	5.6. Layouting				
	5.7. Link & Navigation				
	5.8. Route Group				
	5.9. Dynamic Route				
	6. NextJS Tingkat Lanjut				
	6.1. Loading UI				
	6.2. Page Not Found				
	6.3. Middleware				
	6.4. Simple Auth with Middleware				
	6.5. Data Fetching & konfigurasi API				
	6.6. Lazy Loading				
	7. Implementasi ExpressJS dan Networking				
	7.1. Buat Project dan Instalasi				
	7.2. Koneksi dengan database Mysql				
	7.3. Buat CRUD API				
	7.4. Testing API				
	7.5. Networking dengan Frontend				
	8. Fundamental NestJS				
	8.1. Buat Project dan instalasi				
	8.2. Struktur Folder NestJS				
	8.3. Konsep Modular & Service Pattern				

	8.4. Basic Routing 8.5. Module 8.6. Controller 8.7. Service 8.8. Service Provider & Dependency Injection 9. NestJS Rest API 9.1. Membuat Module, Controller, dan Service Task 9.2. Inject Service ke dalam Controller 9.3. Membuat API 9.4. Pengenalan DTO (Data Transfer Object) 9.5. Database (Prisma ORM) 9.6. Validation 10. Studi Kasus : Authentication, Relational Database, CRUD Generator 10.1. Authentication Menggunakan JWT 10.2. Relation Database 10.3. CRUD Generator 11. Final Project 11.1. Penentuan latar belakang masalah 11.2. Penerapan dan pengembangan aplikasi berbasis kerangka kerja 11.3. Penyusunan laporan akhir 11.4. Demo akhir aplikasi berbasis kerangka kerja
<b>Pustaka</b>	<b>Utama</b> 1. Pratama, Andre. 2022. Javascript Uncover. Duniaikom. <b>Pendukung</b> 2. Pratama, Andre. 2022. React Uncover. Duniaikom 3. React Official : <a href="https://react.dev/">https://react.dev/</a> 4. NextJS Official : <a href="https://nextjs.org/">https://nextjs.org/</a> 5. ExpressJS Official : <a href="https://expressjs.com/">https://expressjs.com/</a> 6. NestJS Official : <a href="https://nestjs.com/">https://nestjs.com/</a> 7. Pendukung : <a href="https://www.mistercoding.com/">https://www.mistercoding.com/</a>
<b>Media Pembelajaran</b>	<b>Software</b> Power Point, Visual Studio Code, Browser <b>Hardware</b> PC & LCD Projector
<b>Teacher/Team Teaching/Tim LS</b>	-
<b>Assessment</b>	Pengetahuan: Tes tulis (UTS, UAS), Psikomotorik: Kinerja (Tugas). Sikap: Observasi harian
<b>Mata Kuliah Syarat</b>	Pemrograman Berorientasi Objek

Pertemuan Ke	Kemampuan Akhir yang direncanakan	Indikator Pencapaian Kompetensi	Materi Pokok	Bentuk dan Metode Pembelajaran	Pengalaman Belajar Mahasiswa	Estimasi Waktu	Penilaian			Referensi
							Bentuk & Kriteria	Indikator Penilaian	Bobot (%)	
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
1	Mahasiswa mampu menjelaskan mengenai pengetahuan dan konsep kerangka kerja dalam membuat aplikasi.	1.1. Mahasiswa mampu menjelaskan mengenai pengetahuan dan konsep dasar pembuatan aplikasi menggunakan framework 1.2. Mahasiswa mampu menjelaskan alasan mengapa menggunakan framework 1.3. Mahasiswa mampu menyebutkan dan menjelaskan framework untuk frontend dan backend 1.4. Mahasiswa mampu menjelaskan perbedaan framework dan library. 1.5. Mahasiswa mampu menjelaskan mengenai pengetahuan dan konsep dasar pembuatan aplikasi menggunakan konsep frontend-backend	1. Pengenalan Kerangka Kerja (framework) 1.1. Pengenalan framework 1.2. Mengapa menggunakan framework 1.3. Framework Frontend dan Backend 1.4. Perbedaan framework dan library? 1.5. Konsep frontend-backend	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, tanya jawab, presentasi.	Mendiskusikan pengetahuan dan konsep dasar kerangka kerja dalam membuat aplikasi.	TM: 1x(3x50") BT: 1x(3x60") BM: 1x(3x60")	Bentuk nontest: Meringkas materi kuliah Kriteria: Rubrik Holistik	1. Ketepatan dalam menjelaskan materi. 2. Kerapian penyajian laporan. 3. Tidak mengandung plagiasi.	5	2,3,4,5,6
2	Mahasiswa mampu menerapkan React sebagai dasar pemrograman frontend.	2.1. Mahasiswa mampu menjelaskan tentang pengertian React 2.2. Mahasiswa mampu menjelaskan sejarah dan perkembangan React 2.3. Mahasiswa mampu menerapkan EcmaScript untuk React 2.4. Mahasiswa mampu menerapkan JSX pada React 2.5. Mahasiswa mampu menerapkan penggunaan React Component 2.6. Mahasiswa mampu menerapkan Props, Event, State pada React	2. Pengenalan React Tingkat Dasar 2.1. Pengertian React 2.2. Sejarah dan Perkembangan React 2.3. EcmaScript untuk React 2.4. JSX (Javascript XML) 2.5. React Component 2.6. React Props, Event dan State	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, tanya jawab, presentasi dan tutorial	Mendiskusikan dan menerapkan konsep React sebagai dasar pemrograman frontend.	TM: 1x(3x50") BT: 1x(3x60") BM: 1x(3x60")	Bentuk nontest: penugasan praktik Kriteria: Rubrik Holistik	1. Ketepatan dalam menerapkan konsep React sebagai pemrograman frontend. 2. Kerapian penyajian laporan. 3. Tidak mengandung plagiasi.	2,5	1,2,3
3	Mahasiswa mampu menerapkan React sebagai dasar pemrograman frontend.	3.1. Mahasiswa mampu menerapkan React Hooks 3.2. Mahasiswa mampu menerapkan React Form Processing 3.3. Mahasiswa mampu menerapkan React DOM Manipulation 3.4. Mahasiswa mampu menjelaskan dan menerapkan Siklus Hidup Komponen React	3. Pengenalan React Tingkat Lanjut 3.1. React Hooks 3.2. React Form Processing 3.3. React DOM Manipulation 3.4. Siklus Hidup Komponen React	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, tanya jawab, presentasi dan tutorial	Mendiskusikan dan menerapkan konsep React sebagai dasar pemrograman frontend.	TM: 1x(3x50") BT: 1x(3x60") BM: 1x(3x60")	Bentuk nontest: penugasan praktik Kriteria: Rubrik Holistik	1. Ketepatan dalam menerapkan konsep React sebagai pemrograman frontend. 2. Kerapian penyajian laporan. 3. Tidak mengandung plagiasi.	2,5	1,2,3
4	Mahasiswa mampu membuat aplikasi frontend menggunakan React	4.1. mahasiswa mampu menerapkan create react app dalam membuat project 4.2. Mahasiswa mampu menerapkan Create pada state React 4.3. Mahasiswa mampu menerapkan Read pada state React 4.4. Mahasiswa mampu menerapkan Update pada state React 4.5. Mahasiswa mampu menerapkan Delete pada state React 4.6. Mahasiswa mampu menerapkan build aplikasi React	4. Studi Kasus : Create React App CRUD React 4.1. Create React App 4.2. Create pada State React 4.3. Read pada State React 4.4. Update pada State React 4.5. Delete pada State React 4.6. Build React App	Bentuk: Kuliah Metode: Case Based Learning	Mendiskusikan dan membuat CRUD aplikasi React.	TM: 1x(3x50") BT: 1x(3x60") BM: 1x(3x60")	Bentuk nontest: Meringkas materi kuliah Kriteria: Rubrik Holistik	1. Ketepatan dalam menerapkan studi kasus yang diberikan. 2. Kerapian penyajian laporan. 3. Tidak mengandung plagiasi.	2,5	1,2,3
5	Mahasiswa mampu membuat aplikasi frontend	5.1. Mahasiswa mampu menjelaskan framework NextJS	5. NextJS Tingkat Dasar	Bentuk: Kuliah	Mendiskusikan dan menerapkan pembuatan frontend menggunakan	TM: 1x(3x50") BT: 1x(3x60")	Bentuk nontest: Meringkas	1. Ketepatan dalam	2,5	1,2,3,4

	menggunakan NextJS	5.2. Mahasiswa mampu menerapkan instalasi dan buat project NextJS 5.3. Mahasiswa mampu menjelaskan server component pada NextJS 5.4. Mahasiswa mampu menjelaskan struktur folder NextJS 5.5. Mahasiswa mampu menerapkan Routing, Layout & Page pada NextJS 5.6. Mahasiswa mampu menerapkan Layouting pada NextJS 5.7. Mahasiswa mampu menerapkan Link & Navigation pada NextJS 5.8. Mahasiswa mampu menerapkan Route Group pada NextJS 5.9. Mahasiswa mampu menerapkan Dynamic Route pada NextJS	5.1. Pengenalan NextJS 5.2. Buat Project dan Instalasi 5.3. Server Component 5.4. Struktur Folder NextJS 5.5. Routing, Layout & Page 5.6. Layouting 5.7. Link & Navigation 5.8. Route Group 5.9. Dynamic Route	Metode: Ceramah, diskusi kelompok/kelas, tanya jawab, presentasi dan tutorial	NextJS.	BM: 1x(3x60')	materi kuliah Kriteria: Rubrik Holistik	menerapkan konsep NextJS sebagai pemrograman frontend. 2. Kerapian penyajian laporan. 3. Tidak mengandung plagiasi.		
6	Mahasiswa mampu membuat aplikasi backend menggunakan NextJS	6.1. Mahasiswa mampu menerapkan Loading UI pada NextJS 6.2. Mahasiswa mampu menerapkan Page Not Found pada NextJS 6.3. Mahasiswa mampu menerapkan Middleware pada NextJS 6.4. Mahasiswa mampu menerapkan Authentikasi sederhana menggunakan Middleware pada NextJS 6.5. Mahasiswa mampu menerapkan Data Fetching dan konfigurasi API pada NextJS 6.6. Mahasiswa mampu menerapkan Lazy Loading pada NextJS	6. NextJS Tingkat Lanjut 6.1. Loading UI 6.2. Page Not Found 6.3. Middleware 6.4. Simple Auth with Middleware 6.5. Data Fetching & konfigurasi API 6.6. Lazy Loading	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, tanya jawab, presentasi dan tutorial	Mendiskusikan dan menerapkan pembuatan frontend menggunakan NextJS.	TM: 1x(3x50') BT: 1x(3x60') BM: 1x(3x60')	Bentuk nontest: Meringkas materi kuliah Kriteria: Rubrik Holistik	1. Ketepatan dalam menerapkan konsep NextJS sebagai pemrograman frontend. 2. Kerapian penyajian laporan. 3. Tidak mengandung plagiasi.	2,5	1,2,3,4
7	Mahasiswa mampu membuat aplikasi backend menggunakan ExpressJS dan memanfaatkannya pada frontend	7.1. Mahasiswa mampu menerapkan instalasi dan pembuatan project ExpressJS 7.2. Mahasiswa mampu menerapkan koneksi ke database mysql pada ExpressJS 7.3. Mahasiswa mampu membuat CRUD API pada ExpressJS 7.4. Mahasiswa mampu melakukan testing pada API yang telah dibuat 7.5. Mahasiswa mampu menghubungkan aplikasi frontend dengan backend	7. Implementasi ExpressJS dan Networking 7.1. Buat Project dan Instalasi 7.2. Koneksi dengan database Mysql 7.3. Buat CRUD API 7.4. Testing API 7.5. Networking dengan Frontend	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, tanya jawab, presentasi dan tutorial	Mendiskusikan dan menerapkan pembuatan backend menggunakan ExpressJS dan komunikasinya dengan frontend.	TM: 1x(3x50') BT: 1x(3x60') BM: 1x(3x60')	Bentuk nontest: Meringkas materi kuliah Kriteria: Rubrik Holistik	1. Ketepatan dalam menerapkan konsep ExpressJS sebagai pemrograman backend dan pemanfaatannya. 2. Kerapian penyajian laporan. 3. Tidak mengandung plagiasi.	2,5	1,2,3,4,5
8	UTS									
9	Mahasiswa mampu menerapkan NestJS sebagai dasar pemrograman backend.	8.1. Mahasiswa mampu menerapkan instalasi dan pembuatan project NestJS 8.2. Mahasiswa mampu menjelaskan struktur folder pada NestJS 8.3. Mahasiswa mampu menjelaskan konsep modular dan service pattern pada NestJS 8.4. Mahasiswa mampu menerapkan basic routing pada NestJS 8.5. Mahasiswa mampu menerapkan konsep module pada NestJS 8.6. Mahasiswa mampu menerapkan konsep controller pada NestJS 8.7. Mahasiswa mampu menerapkan service pada NestJS 8.8. Mahasiswa mampu menerapkan service provider dan dependency injection pada NestJS	8. Fundamental NestJS 8.1. Buat Project dan instalasi 8.2. Struktur Folder NestJS 8.3. Konsep Modular & Service Pattern 8.4. Basic Routing 8.5. Module 8.6. Controller 8.7. Service 8.8. Service Provider & Dependency Injection	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, tanya jawab, presentasi dan tutorial	Mendiskusikan dan menerapkan pembuatan backend menggunakan NestJS.	TM: 1x(3x50') BT: 1x(3x60') BM: 1x(3x60')	Bentuk nontest: Meringkas materi kuliah Kriteria: Rubrik Holistik	1. Ketepatan dalam menerapkan konsep NestJS sebagai pemrograman backend. 2. Kerapian penyajian laporan. 3. Tidak mengandung plagiasi.	2,5	1,2,3,4,6,7
10	Mahasiswa mampu menerapkan NestJS sebagai dasar pemrograman backend.	9.1. Mahasiswa mampu membuat module, controller, dan service task pada NestJS 9.2. Mahasiswa mampu menerapkan inject service kedalam controller pada NestJS 9.3. Mahasiswa mampu membuat API menggunakan NestJS 9.4. Mahasiswa mampu menjelaskan DTO (data transfer object) pada NestJS 9.5. Mahasiswa mampu menerapkan Prisma ORM pada NestJS 9.6. Mahasiswa mampu menerapkan validation pada NestJS	9. NestJS Rest API 9.1. Membuat Module, Controller, dan Service Task 9.2. Inject Service ke dalam Controller 9.3. Membuat API 9.4. Pengenalan DTO (Data Transfer Object) 9.5. Database (Prisma ORM) 9.6. Validation	Bentuk: Kuliah Metode: Ceramah, diskusi kelompok/kelas, tanya jawab, presentasi dan tutorial	Mendiskusikan dan menerapkan pembuatan backend menggunakan NestJS.	TM: 1x(3x50') BT: 1x(3x60') BM: 1x(3x60')	Bentuk nontest: Meringkas materi kuliah Kriteria: Rubrik Holistik	1. Ketepatan dalam menerapkan konsep NestJS sebagai pemrograman backend. 2. Kerapian penyajian laporan. 3. Tidak mengandung plagiasi.	2,5	1,2,3,4,6,7
11-12	Mahasiswa mampu membuat project NestJS sebagai backend	10.1. Mahasiswa mampu menerapkan authentication menggunakan JWT (json web token) pada NestJS 10.2. Mahasiswa mampu menerapkan relation database pada NestJS 10.3. Mahasiswa mampu membuat CRUD generator pada NestJS	10. Studi Kasus : Authentication, Relational Database, CRUD Generator 10.1. Authentication Menggunakan JWT 10.2. Relation Database 10.3. CRUD Generator	Bentuk: Kuliah Metode: Case Based Learning	Mendiskusikan dan menerapkan pembuatan aplikasi dengan konsep frontend dan backend.	TM: 2x(3x50') BT: 2x(3x60') BM: 2x(3x60')	Bentuk nontest: Meringkas materi kuliah Kriteria: Rubrik Holistik	1. Ketepatan dalam menerapkan studi kasus yang diberikan. 2. Kerapian penyajian laporan. 3. Tidak mengandung plagiasi.	5	1,2,3,4,6,7
13-15	Mahasiswa mampu membangun aplikasi menggunakan kerangka kerja untuk menyelesaikan suatu permasalahan.	11.1. Mahasiswa mampu merumuskan latar belakang permasalahan 11.2. Mahasiswa mampu menerapkan dan mengembangkan aplikasi berbasis kerangka kerja sebagai solusi permasalahan	11. Final Project 11.1. Penentuan latar belakang masalah	Bentuk: Kuliah & Responsi Metode: Project Based Learning	Merumuskan permasalahan kemudian mengembangkan aplikasi berbasis kerangka kerja sebagai solusi.	TM: 3x(3x50') BT: 3x(3x60') BM: 3x(3x60')	Bentuk nontest: Proposal, Laporan Kemajuan dan Laporan Akhir Final Project Kriteria: Rubrik Holistik	1. Ketepatan dalam menentukan latar belakang permasalahan, menerapkan aplikasi berbasis framework	20	1,2,3,4,6,7

		11.3. Mahasiswa mampu menyusun laporan akhir	11.2. Penerapan dan pengembangan aplikasi berbasis kerangka kerja			PLANSUR	11.1. Menentukan sebagai solusi permasalahan.		
		11.4. Mahasiswa mampu mendemonstrasikan aplikasi berbasis kerangka kerja yang telah dikembangkan secara bertanggung jawab	11.3. Penyusunan laporan akhir				2. Kerapian penyajian laporan.		
			11.4. Demo akhir aplikasi berbasis kerangka kerja				3. Tidak mengandung plagiasi.		
16	UAS								
Catatan :									
1. 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 studinya yang diperoleh melalui proses pembelajaran.									
2. 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.									
3. 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.									
4. Sub-CP 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.									
5. 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.									
6. 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.									
Catatan tambahan:									
(1). Bobot SKS (P = Praktek; T= Teori).									
(2). TM: Tatap Muka; BT: Beban Tugas; BM: Belajar Mandiri.									
(3). 1 sks = (50' TM + 50' PT + 60' BM)/Minggu									
(4). Simbol-simbol elemen KKNi pada CPL-Prod: S = Sikap; KU = Ketrampilan Umum; KK = Ketrampilan Khusus; P = Pengetahuan									