

SILABUS MATA PELAJARAN

- Nama Sekolah : SMK WIRASWASTA CIMAHI
- Bidang Keahlian : Teknologi Informasi dan Komunikasi
- Kompetensi Keahlian : Rekayasa Perangkat Lunak
- Mata Pelajaran : Pemrograman Berorientasi Obyek
- Durasi (Waktu) : 560 jam
- Kelas/Semester : XI,XII
- KI-3 (Pengetahuan) : Memahami, menerapkan, menganalisis, dan mengevaluasi tentang pengetahuan faktual, konseptual, operasional dasar, dan metakognitif sesuai dengan bidang dan lingkup kerja Teknik Komputer dan Informatika pada tingkat teknis, spesifik, detil, dan kompleks, berkenaan dengan ilmu pengetahuan, teknologi, seni, budaya, dan humaniora dalam konteks pengembangan potensi diri sebagai bagian dari keluarga, sekolah, dunia kerja, warga masyarakat nasional, regional, dan internasional.
- KI-4 (Keterampilan) : Melaksanakan tugas spesifik, dengan menggunakan alat, informasi, dan prosedur kerja yang lazim dilakukan serta menyelesaikan masalah sederhana sesuai dengan bidang dan lingkup kerja Teknik Komputer dan Informatika. Menampilkan kinerja mandiri dengan mutu dan kuantitas yang terukur sesuai dengan standar kompetensi kerja. Menunjukkan keterampilan menalar, mengolah, dan menyaji secara efektif, kreatif, produktif, kritis, mandiri, kolaboratif, komunikatif, dan solutif dalam ranah abstrak terkait dengan pengembangan dari yang dipelajarinya di sekolah, serta mampu melaksanakan tugas spesifik dibawah pengawasan langsung. Menunjukkan keterampilan mempersepsi, kesiapan, meniru, membiasakan gerak mahir, menjadikan gerak alami, sampai dengan tindakan orisinal dalam ranah konkret terkait dengan pengembangan dari yang dipelajarinya di sekolah, serta mampu melaksanakan tugas spesifik dibawah pengawasan langsung.

Kompetensi Dasar	Indikator Pencapaian Kompetensi	Materi Pokok	Alokasi Waktu (JP)	Kegiatan Pembelajaran	Penilaian
1	2	3	4	5	6
3.1 Memahami konsep pemrograman berorientasi objek 4.1 Mempresentasikan konsep pemrograman berorientasi objek	3.1.1 Menjelaskan konsep dasar pemrograman berorientasi obyek. 3.1.2 Menjelaskan prosedur pemrograman berorientasi obyek. 3.1.3 Menjelaskan perbandingan antara pemrograman berorientasi obyek dengan pemrograman struktural. 4.1.1 Mempresentasikan konsep pemrograman berorientasi obyek.	• Konsep pemrograman berorientasi obyek. • Prosedur pemrograman berorientasi obyek. • Keunggulan pemrograman berorientasi obyek terhadap pemrograman struktural.	8	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang konsep pemrograman berorientasi objek • Mengumpulkan data tentang konsep pemrograman berorientasi objek • Mengolah data tentang konsep pemrograman berorientasi objek • Mengomunikasikan tentang konsep pemrograman berorientasi objek	

3.2 Menerapkan dasar dan aturan pemrograman berorientasi objek	3.2.1 Menjelaskan berbagai tools pengembang program berorientasi obyek. 3.2.2 Menjelaskan konfigurasi tools pengembang perangkat lunak berorientasi obyek. 3.2.3 Menjelaskan kebutuhan tools editor program berorientasi obyek. 3.2.4 Menjelaskan struktur dasar dalam pemrograman berorientasi obyek. 3.2.5 Menjelaskan prosedur kompilasi program. 3.2.6 Menjelaskan tata cara menjalankan program hasil kompilasi. 3.2.7 Menerapkan prosedur aturan pemrograman berorientasi obyek.	•Macam-macam perangkat lunak pengembang program berorientasi obyek. •Prosedur instalasi dan konfigurasi tools pengembang aplikasi program berorientasi obyek. •Prosedur Kompilasi •Menjalankan program hasil kompilasi. •Prosedur patching komputer. •Aplikasi editor •Virtual machine	24	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang dasar dan aturan pemrograman berorientasi objek • Mengumpulkan data tentang dasar dan aturan pemrograman berorientasi objek • Mengolah data tentang dasar dan aturan pemrograman berorientasi objek • Mengomunikasikan tentang dasar dan aturan pemrograman berorientasi objek	
4.2 Membuat kode program dasar dan aturan pemrograman berorientasi objek	4.2.1 Menginstalasi tools pengembang program aplikasi berorientasi obyek kedalam komputer. 4.2.2 Mengkonfigurasi tools pengembang program aplikasi berorientasi obyek. 4.2.3 Merencanakan program aplikasi sederhana berorientasi obyek. 4.2.4 Membuat kode program aplikasi sederhana berorientasi obyek. 4.2.5 Menjalankan program aplikasi sederhana berorientasi obyek.				

3.3 Menerapkan konsep <i>class</i> , <i>method</i> dan <i>object</i>	3.3.1 Menjelaskan konsep class dalam pemrograman berorientasi obyek.	• Konsep Class, object, method dalam oop.	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang konsep <i>class</i> , <i>method</i> dan <i>object</i>	
4.3 Membuat <i>class</i> , <i>method</i> dan <i>object</i>	3.3.2 Menerapkan class dalam pemrograman berorientasi obyek.	• Penamaan file dalam oop		• Mengumpulkan data tentang konsep <i>class</i> , <i>method</i> dan <i>object</i>	
	3.3.3 Menjelaskan konsep method dalam pemrograman berorientasi obyek.	• Penerapan class dan method dalam program aplikasi oop.		• Mengolah data tentang konsep <i>class</i> , <i>method</i> dan <i>object</i>	
	3.3.4 Menerapkan method dalam pemrograman berorientasi obyek.			• Mengomunikasikan tentang konsep <i>class</i> , <i>method</i> dan <i>object</i>	
	3.3.5 Menjelaskan konsep obyek dalam pemrograman berorientasi obyek.				
	3.3.6 Menerapkan obyek dalam pemrograman berorientasi obyek.				
	4.3.1 Merancang program aplikasi berorientasi obyek dengan penerapan class, method dan obyek.				
	4.3.2 Membuat kode program aplikasi berorientasi obyek dengan penerapan class, method dan obyek.				
	4.3.3 Menguji program aplikasi berorientasi obyek yang menerapkan class, method dan obyek.				

3.4 Menerapkan konsep enkapsulasi dalam melindungi data dan informasi	3.4.1 Menjelaskan konsep dan tujuan proses enkapsulasi dalam pemrograman aplikasi berorientasi obyek.	• Konsep enkapsulasi dalam program aplikasi berbasis oop.	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang konsep enkapsulasi dalam melindungi data dan informasi	
4.4 Membuat kode program enkapsulasi dalam melindungi data	3.4.2 Menjelaskan prosedur enkapsulasi dalam pemrograman aplikasi berorientasi obyek.	• Prosedur enkapsulasi dalam program aplikasi berbasis oop.		• Mengumpulkan data tentang konsep enkapsulasi dalam melindungi data dan informasi	
	3.4.3 Menjelaskan persyaratan proses enkapsulasi dalam pemrograman aplikasi berorientasi obyek.	• Prosedur rancangan program aplikasi berbasis oop dengan penerapan enkapsulasi.		• Mengolah data tentang konsep enkapsulasi dalam melindungi data dan informasi	
	3.4.4 Menerapkan prosedur enkapsulasi dalam pemrograman aplikasi berorientasi obyek.	• Keunggulan enkapsulasi dalam program aplikasi berbasis oop		• Mengomunikasikan tentang konsep enkapsulasi dalam melindungi data dan informasi	
	4.4.1 Merancang program aplikasi berorientasi obyek dengan penerapan enkapsulasi dalam melindungi data dan informasi.				
	4.4.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan enkapsulasi.				
	4.4.3 Menguji program aplikasi berorientasi obyek yang menerapkan enkapsulasi.				
3.5 Menerapkan konsep pewarisan	3.5.1 Menjelaskan konsep dan tujuan proses pewarisan dalam pemrograman aplikasi berorientasi obyek.	• Konsep enheritance (pewarisan) dalam program aplikasi berbasis oop.	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang konsep pewarisan	
4.5 Membuat hubungan antara					

<i>class</i> dengan pola pewarisan	3.5.2 Menjelaskan prosedur pewarisan dalam pemrograman aplikasi berorientasi obyek. 3.5.3 Menjelaskan keterkaitan antara class dan pewarisan dalam pemrograman aplikasi berorientasi obyek. 3.5.4 Menerapkan prosedur pewarisan dalam pemrograman aplikasi berorientasi obyek 4.5.1 Merancang program aplikasi berorientasi obyek dengan penerapan pewarisan dalam melindungi data dan informasi. 4.5.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan pewarisan. 4.5.3 Menguji program aplikasi berorientasi obyek yang menerapkan pewarisan.	• Prosedur enheritance (pewarisan) dalam program aplikasi berbasis oop. • Prosedur rancangan program aplikasi berbasis oop dengan penerapan enheritance (pewarisan). • Keunggulan enheritance (pewarisan) dalam program aplikasi berbasis oop		• Mengumpulkan data tentang konsep pewarisan • Mengolah data tentang konsep pewarisan • Mengomunikasikan tentang konsep pewarisan	
3.6 Menerapkan konsep <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i> 4.6 Membuat kode program sesuai konsep <i>polymorphism</i>	3.6.1 Menjelaskan konsep <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i> dalam pemrograman aplikasi berorientasi obyek. 3.6.2 Menjelaskan prosedur <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i> dalam pemrograman aplikasi berorientasi obyek.	• Konsep <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i> dalam program aplikasi berbasis oop. • Prosedur <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i>	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang konsep <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i> • Mengumpulkan data tentang konsep <i>polymorphism</i>	

dengan <i>overloading</i> dan <i>overriding</i>	3.6.3 Menerapkan prosedur <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i> dalam pemrograman aplikasi berorientasi obyek 4.6.1 Merancang program aplikasi berorientasi obyek dengan penerapan <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i> dalam melindungi data dan informasi. 4.6.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i>. 4.6.3 Menguji program aplikasi berorientasi obyek yang menerapkan <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i>.	dalam program aplikasi berbasis oop. • Prosedur rancangan program aplikasi berbasis oop dengan penerapan <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i>. • Keunggulan <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i> dalam program aplikasi berbasis oop		dengan <i>overloading</i> dan <i>overriding</i> • Mengolah data tentang konsep <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i> • Mengomunikasikan tentang konsep <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i>	
3.7 Menerapkan penggunaan <i>package</i> dalam aplikasi 4.7 Membuat aplikasi melalui pengelompokan <i>class</i> dalam <i>package</i>	3.7.1 Menjelaskan konsep <i>package</i> dalam pemrograman aplikasi berorientasi obyek. 3.7.2 Menjelaskan proses pengelompokan <i>class</i> dalam pembuatan <i>package</i>. 3.7.3 Menjelaskan prosedur pembuatan <i>package</i> dalam pemrograman aplikasi berorientasi obyek.	• Konsep <i>package</i> dalam program aplikasi berbasis oop. • Prosedur pembuatan <i>package</i> dalam program aplikasi berbasis oop. • Prosedur rancangan program aplikasi	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang penggunaan <i>package</i> dalam aplikasi • Mengumpulkan data tentang penggunaan <i>package</i> dalam aplikasi	

	3.7.4 Menjelaskan proses kompilasi <i>package</i>. 3.7.5 Menerapkan <i>package</i> dalam pemrograman aplikasi berorientasi obyek 4.7.1 Merancang program aplikasi berorientasi obyek dengan penerapan <i>package</i> 4.7.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan <i>package</i>. 4.7.3 Menguji program aplikasi berorientasi obyek yang menerapkan <i>package</i>.	berbasis oop dengan penerapan <i>package</i>. • Prosedur kompilasi <i>package</i> dalam pemrograman berbasis oop. • Keunggulan <i>package</i> dalam program aplikasi berbasis oop		• Mengolah data tentang penggunaan <i>package</i> dalam aplikasi • Mengomunikasikan tentang penggunaan <i>package</i> dalam aplikasi	
3.8 Merancang pembuatan <i>interface</i> 4.8 Membuat rancangan aplikasi dengan <i>interface</i>	3.8.1 Menjelaskan konsep <i>interface</i> dalam pemrograman aplikasi berorientasi obyek. 3.8.2 Menjelaskan prosedur pembuatan <i>interface</i> dalam pemrograman aplikasi berorientasi obyek. 3.8.3 Menerapkan <i>interface</i> dalam pemrograman aplikasi berorientasi obyek 4.8.1 Merancang program aplikasi berorientasi obyek dengan penerapan <i>interface</i> 4.8.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan <i>interface</i>.	• Konsep <i>interface</i> dalam program aplikasi berbasis oop. • Prosedur pembuatan <i>interface</i> dalam program aplikasi berbasis oop. • Prosedur rancangan program aplikasi berbasis oop dengan penerapan <i>interface</i>. • Prosedur kompilasi <i>package</i> dalam pemrograman berbasis oop.	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang pembuatan <i>interface</i> • Mengumpulkan data tentang pembuatan <i>interface</i> • Mengolah data tentang pembuatan <i>interface</i> • Mengomunikasikan tentang pembuatan <i>interface</i>	

	4.8.3 Menguji program aplikasi berorientasi obyek yang menerapkan <i>interface</i> .	•Keunggulan <i>interface</i> dalam program aplikasi berbasis oop			
3.9 Menganalisis pemanfaatan <i>class library</i> 4.9 Menggunakan <i>class library</i> dalam memecahkan masalah	3.9.1 Menjelaskan konsep <i>class library</i> dalam pemrograman aplikasi berorientasi obyek. 3.9.2 Menjelaskan prosedur pemanfaatan <i>class library</i> dalam pemrograman aplikasi berorientasi obyek. 3.9.3 Menganalisis pemanfaatan <i>class library</i> dalam pemrograman aplikasi berorientasi obyek. 3.9.4 Menentukan penggunaan <i>class library</i> dalam pemrograman aplikasi berorientasi obyek. 3.9.5 Menerapkan <i>class library</i> dalam pemrograman aplikasi berorientasi obyek. 4.9.1 Merancang program aplikasi berorientasi obyek dengan penerapan <i>class library</i> . 4.9.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan <i>class library</i> . 4.9.3 Menguji program aplikasi berorientasi obyek yang menerapkan <i>class library</i> .	•Konsep <i>class library</i> dalam program aplikasi berbasis oop. •Prosedur penerapan <i>class library</i> dalam program aplikasi berbasis oop. •Prosedur rancangan program aplikasi berbasis oop dengan penerapan <i>class library</i>. •Prosedur kompilasi <i>program</i> aplikasi dengan penerapan <i>class library</i> pemrograman berbasis oop. •Keunggulan program dengan penerapan <i>class library</i> dalam program aplikasi berbasis oop	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang pemanfaatan <i>class library</i> • Mengumpulkan data tentang pemanfaatan <i>class library</i> • Mengolah data tentang pemanfaatan <i>class library</i> • Mengomunikasikan tentang pemanfaatan <i>class library</i>	

3.10 Mengevaluasi penanganan kesalahan dalam program 4.10 Memodifikasi kode program untuk penanganan kesalahan dalam program	3.10.1 Menjelaskan konsep <i>error handling</i> (penanganan kesalahan) dalam pemrograman aplikasi berorientasi obyek. 3.10.2 Menjelaskan prosedur <i>error handling</i> (penanganan kesalahan) dalam pemrograman aplikasi berorientasi obyek. 3.10.3 Menganalisis penanganan kesalahan (<i>error handling</i>) dalam pemrograman aplikasi berorientasi obyek. 3.10.4 Menentukan penggunaan <i>error handling</i> (penanganan kesalahan) dalam pemrograman aplikasi berorientasi obyek. 3.10.5 Menerapkan <i>error handling</i> (penanganan kesalahan) dalam pemrograman aplikasi berorientasi obyek. 4.10.1 Merancang program aplikasi berorientasi obyek dengan penerapan <i>error handling</i> (penanganan kesalahan). 4.10.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan <i>error handling</i> (penanganan kesalahan). 4.10.3 Menguji program aplikasi berorientasi obyek yang menerapkan <i>error handling</i> (penanganan kesalahan).	• Konsep penanganan kesalahan dalam program aplikasi berbasis oop. • Prosedur penanganan kesalahan (<i>error handling</i>) dalam pemrograman berbasis oop. • Prosedur pembuatan program aplikasi berbasis oop dengan penanganan kesalahan	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang penanganan kesalahan dalam program • Mengumpulkan data tentang penanganan kesalahan dalam program • Mengolah data tentang penanganan kesalahan dalam program • Mengomunikasikan tentang penanganan kesalahan dalam program	
--	---	---	----	---	--

	4.10.4 Memperbaiki penanganan kesalahan (<i>error handling</i>) dalam program aplikasi berorientasi obyek. 4.10.5 Memodifikasi penanganan kesalahan (<i>error handling</i>) dalam program aplikasi berorientasi obyek.				
3.11 Menerapkan data bentuk string dan pelbagai propertinya 4.11 Membuat kode program untuk mengolah data bentuk string	3.11.1 Menjelaskan konsep data bentuk string dan pelbagai propertinya dalam pemrograman aplikasi berorientasi obyek. 3.11.2 Menjelaskan prosedur data bentuk string dan pelbagai propertinya dalam pemrograman aplikasi berorientasi obyek. 3.11.3 Menentukan penggunaan data bentuk string dan pelbagai propertinya dalam pemrograman aplikasi berorientasi obyek. 3.11.4 Menerapkan data bentuk string dan pelbagai propertinya dalam pemrograman aplikasi berorientasi obyek. 4.11.1 Merancang program aplikasi berorientasi obyek dengan penerapan data bentuk string dan pelbagai propertinya.	• Konsep penggunaan data tipe string dalam pemrograman berbasis oop. • Prosedur pembuatan program berbasis oop dengan menerapkan data tipe string.	8	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang data bentuk string dan pelbagai propertinya • Mengumpulkan data tentang data bentuk string dan pelbagai propertinya • Mengolah data data bentuk string dan pelbagai propertinya • Mengomunikasikan tentang data bentuk string dan pelbagai propertinya	

	4.11.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan data bentuk string dan pelbagai propertinya. 4.11.3 Menguji program aplikasi berorientasi obyek yang menerapkan data bentuk string dan pelbagai propertinya.				
3.12 Menerapkan <i>collection</i> sebagai media penyimpanan data 4.12 Membuat kode program <i>collection</i> sebagai penyimpan data	3.12.1 Menjelaskan konsep <i>collection</i> sebagai media penyimpanan data dalam pemrograman aplikasi berorientasi obyek. 3.12.2 Menjelaskan prosedur <i>collection</i> sebagai media penyimpanan data dalam pemrograman aplikasi berorientasi obyek. 3.12.3 Menentukan penggunaan <i>collection</i> sebagai media penyimpanan data dalam pemrograman aplikasi berorientasi obyek. 3.12.4 Menerapkan <i>collection</i> sebagai media penyimpanan data dalam pemrograman aplikasi berorientasi obyek. 4.12.1 Merancang program aplikasi berorientasi obyek dengan penerapan <i>collection</i> sebagai media penyimpanan data. 4.12.2 Membuat kode program aplikasi berorientasi obyek yang	• Konsep penerapan <i>collection</i> sebagai penyimpan data dalam pemrograman berbasis oop. • Prosedur pembuatan program aplikasi berbasis oop dengan menerapkan <i>collection</i> sebagai media penyimpan data.	16	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang <i>collection</i> sebagai media penyimpanan data • Mengumpulkan data tentang <i>collection</i> sebagai media penyimpanan data • Mengolah data tentang <i>collection</i> sebagai media penyimpanan data • Mengomunikasikan tentang <i>collection</i> sebagai media penyimpanan data	

	menerapkan <i>collection</i> sebagai media penyimpanan data. 4.12.3 Menguji program aplikasi berorientasi obyek yang menerapkan <i>collection</i> sebagai media penyimpanan data.				
3.13 Menerapkan operasi <i>file</i> dan <i>Input Output</i> (IO) 4.13 Membuat kode program untuk operasi <i>file</i> dan operasi <i>Input Output</i>	3.13.1 Menjelaskan konsep operasi <i>file</i> dan <i>Input Output</i> (IO) dalam pemrograman aplikasi berorientasi obyek. 3.13.2 Menjelaskan prosedur operasi <i>file</i> dan <i>Input Output</i> (IO) dalam pemrograman aplikasi berorientasi obyek. 3.13.3 Menentukan penggunaan operasi <i>file</i> dan <i>Input Output</i> (IO) dalam pemrograman aplikasi berorientasi obyek. 3.13.4 Menerapkan operasi <i>file</i> dan <i>Input Output</i> (IO) pemrograman aplikasi berorientasi obyek. 4.13.1 Merancang program aplikasi berorientasi obyek dengan penerapan operasi <i>file</i> dan <i>Input Output</i> (IO). 4.13.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan operasi <i>file</i> dan <i>Input Output</i> (IO). 4.13.3 Menguji program aplikasi berorientasi obyek yang	• Konsep operasi file dalam pemrograman berbasis oop. • Konsep input dan output dalam pemrograman aplikasi berbasis oop. • Prosedur pembuatan program aplikasi berbasis oop untuk operasi file. • Prosedur pembuatan program aplikasi berbasis oop untuk input dan output. • Membuat program aplikasi berbasis oop untuk operasi file yang dapat mengedit, menambah,	24	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang operasi <i>file</i> dan <i>Input Output</i>(IO) • Mengumpulkan data tentang operasi <i>file</i> dan <i>Input Output</i>(IO) • Mengolah data tentang operasi <i>file</i> dan <i>Input Output</i>(IO) • Mengomunikasikan tentang operasi <i>file</i> dan <i>Input Output</i>(IO)	

	menerapkan operasi <i>file</i> dan <i>Input Output</i> (IO).	menghapus dan membaca isi file.			
3.14 Menerapkan dasar dasar <i>thread</i> 4.14 Membuat instruksi-instruksi <i>thread</i>	3.14.1 Menjelaskan konsep dasar instruksi <i>thread</i> dalam pemrograman aplikasi berorientasi obyek. 3.14.2 Menjelaskan prosedur instruksi <i>thread</i> dalam pemrograman aplikasi berorientasi obyek. 3.14.3 Menentukan penggunaan instruksi <i>thread</i> dalam pemrograman aplikasi berorientasi obyek. 4.14.1 Menerapkan instruksi <i>thread</i> dalam pemrograman aplikasi berorientasi obyek. 4.14.2 Merancang program aplikasi berorientasi obyek dengan penerapan instruksi <i>thread</i> . 4.14.3 Membuat kode program aplikasi berorientasi obyek yang menerapkan instruksi <i>thread</i> . 4.14.4 Menguji program aplikasi berorientasi obyek yang menerapkan instruksi <i>thread</i> .	• Konsep dasar thread dalam oop. • Prosedur penggunaan thread dalam oop • Membuat program oop menerapkan thread.	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang dasar dasar <i>thread</i> • Mengumpulkan data tentang dasar dasar <i>thread</i> • Mengolah data tentang dasar dasar <i>thread</i> • Mengomunikasikan tentang dasar dasar <i>thread</i>	
3.15 Merancang antar muka atau <i>User Interface</i> (UI)	3.15.1 Menjelaskan konsep antar muka atau <i>User Interface</i> (UI) dalam pemrograman aplikasi berorientasi obyek.	• Konsep antar muka (<i>user interface</i>) dalam pemrograman aplikasi berbasis oop	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang antar muka atau <i>User Interface</i> (UI)	

4.15 Membuat kode program untuk antar muka atau <i>User Interface (UI)</i>	3.15.2 Menjelaskan prosedur antar muka atau <i>User Interface (UI)</i> dalam pemrograman aplikasi berorientasi obyek. 3.15.3 Menentukan penggunaan antar muka atau <i>User Interface (UI)</i> dalam pemrograman aplikasi berorientasi obyek. 3.15.4 Menerapkan antar muka atau <i>User Interface (UI)</i> pemrograman aplikasi berorientasi obyek. 4.15.1 Merancang program aplikasi berorientasi obyek dengan penerapan antar muka atau <i>User Interface (UI)</i> . 4.15.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan antar muka atau <i>User Interface (UI)</i> . 4.15.3 Menguji program aplikasi berorientasi obyek yang menerapkan antar muka atau <i>User Interface (UI)</i> .	• Prosedur pembuatan <i>user interface</i> dalam pemrograman aplikasi berbasis oop. • Membuat contoh program aplikasi yang menerapkan <i>user interface</i>.		• Mengumpulkan data tentang antar muka atau <i>User Interface (UI)</i> • Mengolah data tentang antar muka atau <i>User Interface (UI)</i> • Mengomunikasikan tentang antar muka atau <i>User Interface (UI)</i>	
3.16 Menerapkan konektifitas dalam akses basis data 4.16 Membuat konektifitas basis data	3.16.1 Menjelaskan konsep konektifitas akses basis data dalam pemrograman aplikasi berorientasi obyek. 3.16.2 Menjelaskan prosedur akses basis data menggunakan <i>library</i> konektor dalam	• Konsep konektifitas untuk akses basis data dalam program aplikasi berbasis oop. • Prosedur koneksi basis data dengan	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang konektifitas dalam akses basis data • Mengumpulkan data tentang	

	pemrograman aplikasi berorientasi obyek. 3.16.3 Menentukan penggunaan konektor basis data dalam pemrograman aplikasi berorientasi obyek. 3.16.4 Menerapkan library konektor akses basis data dalam pemrograman aplikasi berorientasi obyek. 4.16.1 Merancang program aplikasi berorientasi obyek akses basis data menggunakan <i>library konektor</i>. 4.16.2 Membuat kode program aplikasi berorientasi obyek akses basis data menggunakan <i>library konektor</i>. 4.16.3 Menguji program aplikasi berorientasi obyek akses basis data yang menggunakan <i>library konektor</i>.	program aplikasi berbasis oop. ● Penggunaan konektor basis data dalam program aplikasi. ● Membuat contoh program aplikasi berbasis oop untuk koneksi basis data.		konektifitas dalam akses basis data ● Mengolah data tentang konektifitas dalam akses basis data ● Mengomunikasikan tentang konektifitas dalam akses basis data	
3.17 Merancang <i>User Interface</i> menggunakan <i>library</i> 4.17 Mendesain <i>User Interface</i> menggunakan <i>library</i>	3.17.1 Menjelaskan konsep antar muka atau <i>User Interface</i> menggunakan <i>library</i> dalam pemrograman aplikasi berorientasi obyek. 3.17.2 Menjelaskan prosedur antar muka atau <i>User Interface</i> menggunakan <i>library</i> dalam	● Konsep pembuatan <i>user interface</i> menggunakan <i>library</i> bawaan dalam pemrograman aplikasi berbasis oop ● Prosedur pembuatan <i>user</i>	32	● Mengamati untuk mengidentifikasi dan merumuskan masalah tentang <i>User Interface</i> menggunakan <i>library</i> ● Mengumpulkan data tentang <i>User Interface</i>	

	pemrograman aplikasi berorientasi obyek. 3.17.3 Menentukan penggunaan antar muka atau <i>User Interface</i> menggunakan <i>library</i> dalam pemrograman aplikasi berorientasi obyek. 3.17.4 <i>Menerapkan</i> antar muka atau <i>User Interface</i> menggunakan <i>library</i> pemrograman aplikasi berorientasi obyek. 4.17.1 Merancang program aplikasi berorientasi obyek dengan penerapan antar muka atau <i>User Interface</i> menggunakan <i>library</i>. 4.17.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan antar muka atau <i>User Interface</i> menggunakan <i>library</i>. 4.17.3 Menguji program aplikasi berorientasi obyek yang menerapkan antar muka atau <i>User Interface</i> menggunakan <i>library</i>.	<i>interface</i> dengan menggunakan <i>library</i> dalam pemrograman aplikasi berbasis oop		menggunakan <i>library</i> ● Mengolah data tentang <i>User Interface</i> menggunakan <i>library</i> ● Mengomunikasikan tentang <i>User Interface</i> menggunakan <i>library</i>	
3.18 Menerapkan objek multimedia dalam aplikasi	3.18.1 Menjelaskan konsep objek multimedia dalam pemrograman aplikasi berorientasi obyek.	● Konsep penerapan multi media dalam pemrograman aplikasi berbasis oop	32	● Mengamati untuk mengidentifikasi dan merumuskan masalah tentang objek multimedia dalam aplikasi	

4.18 Membuat aplikasi yang dapat menampilkan gambar, audio dan video	3.18.2 Menjelaskan prosedur objek multimedia dalam pemrograman aplikasi berorientasi obyek. 3.18.3 Menentukan penggunaan objek multimedia dalam pemrograman aplikasi berorientasi obyek. 3.18.4 Menerapkan objek multimedia dalam pemrograman aplikasi berorientasi obyek. 4.18.1 Merancang program aplikasi berorientasi obyek dengan penerapan obyek multimedia. 4.18.2 Membuat kode program aplikasi berorientasi obyek dengan penerapan obyek multimedia. 4.18.3 Menguji program aplikasi berorientasi obyek yang menerapkan obyek multimedia.	• Prosedur penerapan akses multimedia dalam pemrograman aplikasi berbasis oop. • Membuat contoh program aplikasi berbasis oop dengan menerapkan multi media.		• Mengumpulkan data tentang objek multimedia dalam aplikasi • Mengolah data tentang objek multimedia dalam aplikasi • Mengomunikasikan tentang objek multimedia dalam aplikasi	
3.19 Memahami <i>Model View Controller</i> (MVC) dalam pemrograman berorientasi objek 4.19 Mempresentasikan <i>Model View Controller</i> (MVC) dalam pemrograman berorientasi objek	3.19.1 Menjelaskan konsep <i>Model View Controller</i> (MVC) dalam pemrograman aplikasi berorientasi obyek. 3.19.2 Menjelaskan prosedur pemanfaatan <i>Model View Controller</i> (MVC) dalam pemrograman aplikasi berorientasi obyek. 3.19.3 Menjelaskan berbagai perangkat lunak aplikasi <i>Model</i>	• Konsep penggunaan aplikasi <i>model view controller</i> (MVC) dalam pemrograman aplikasi berbasis oop. • Prosedur penggunaan MVC dalam pemrograman aplikasi berbasis oop	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang <i>Model View Controller</i> (MVC) dalam pemrograman berorientasi objek • Mengumpulkan data tentang <i>Model View Controller</i>	

	<i>View Controller</i> (MVC) untuk pemrograman aplikasi berorientasi obyek. 4.19.1 Mempresentasikan rancangan program aplikasi berorientasi obyek berbasis MVC.	• Macam-macam aplikasi mvc untuk oop. • Membuat program aplikasi sederhana berbasis oop menggunakan MVC		(MVC) dalam pemrograman berorientasi objek • Mengolah data tentang <i>Model View Controller</i> (MVC) dalam pemrograman berorientasi objek • Mengomunikasikan tentang <i>Model View Controller</i> (MVC) dalam pemrograman berorientasi objek	
3.20 Menerapkan aplikasi berbasis MVC dalam aplikasi dengan pelbagai hasil 4.20 Membuat aplikasi dengan <i>Model View Controller</i> (MVC) dengan hasil final (<i>meta-data</i>)	3.20.1 Menentukan penggunaan perangkat lunak aplikasi <i>Model View Controller</i> (MVC) dalam pemrograman aplikasi berorientasi obyek. 3.20.2 Menerapkan perangkat lunak aplikasi <i>Model View Controller</i> (MVC) dalam pemrograman aplikasi berorientasi obyek. 4.20.1 Merancang program aplikasi berorientasi obyek berbasis <i>Model View Controller</i> (MVC). 4.20.2 Membuat kode program aplikasi berorientasi obyek berbasis <i>Model View Controller</i> (MVC).	• Konsep pengembangan program aplikasi berbasis oop menggunakan MVC. • Mengembangkan program aplikasi oop dengan menggunakan MVC	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang aplikasi berbasis MVC dalam aplikasi dengan pelbagai hasil • Mengumpulkan data tentang aplikasi berbasis MVC dalam aplikasi dengan pelbagai hasil • Mengolah data tentang aplikasi berbasis MVC	

	4.20.3 Menguji program aplikasi berorientasi obyek berbasis <i>Model View Controller</i> (MVC).			dalam aplikasi dengan pelbagai hasil ● Mengomunikasikan tentang aplikasi berbasis MVC dalam aplikasi dengan pelbagai hasil	
--	---	--	--	---	--