



MODUL

PRAKTIKUM PBO

14 Pertemuan

Dosen Pengampu : Freddy Wicaksono

- Disusun oleh
- RizkyPratama 220511030
- Ibnu Risfai 220511002

I. Identitas Mata Pelajaran:

- **Mata Pelajaran:** Pemrograman Berorientasi Objek (PBO)
- **Kelas/Semester:** 3/ TEKNIK INFORMATIKA
- **Lama Pembelajaran:** 1 SEMESTER
- **Jumlah SKS:** [3 SKS]

II. Standar Kompetensi: Peserta didik dapat mengimplementasikan konsep dan prinsip Pemrograman Berorientasi Objek (PBO) dalam bahasa pemrograman Python.

III. Kompetensi Dasar:

1. Peserta didik mampu menjelaskan konsep dasar Pemrograman Berorientasi Objek.
2. Peserta didik dapat membuat dan menggunakan class dan objek.
3. Peserta didik dapat mengimplementasikan inheritance dan polymorphism.
4. Peserta didik mampu menggunakan Exception Handling dalam pemrograman berorientasi objek.
5. Peserta didik dapat menerapkan PBO dalam pembuatan aplikasi sederhana.

IV. Indikator Pencapaian Kompetensi:

1. Peserta didik dapat menjelaskan konsep dasar PBO melalui lisan atau tulisan.
2. Peserta didik dapat membuat class dan objek dalam konteks pemrograman Python.
3. Peserta didik mampu mengidentifikasi dan menerapkan inheritance dan polymorphism.
4. Peserta didik dapat menggunakan Exception Handling untuk menangani kesalahan dalam program.
5. Peserta didik dapat membuat aplikasi sederhana menggunakan Pemrograman Berorientasi Objek.

V. Materi Pembelajaran:

1. Pengenalan Pemrograman Berorientasi Objek

- Konsep dasar PBO.
- Perbedaan antara PBO dan paradigma pemrograman lainnya.

2. Dasar-dasar Python dan Struktur Program

- Variabel, tipe data, dan struktur kontrol.
- Fungsi dan prosedur.

3. Konsep Dasar Pemrograman Berorientasi Objek

- Class dan objek.
- Encapsulation, Inheritance, dan Polymorphism.
- Abstraksi dan enkapsulasi.

4. Membuat Class dan Objek di Python

- Pendefinisian class dan objek.
- Atribut, metode, konstruktor, dan destruktur.

5. Inheritance dan Polymorphism

- Implementasi inheritance.
- Polymorphism dan overriding.
- Abstract classes dan interfaces.

6. Exception Handling

- Konsep dasar exception handling.
- Penggunaan try, except, finally.
- Pembuatan custom exceptions.

7. File Handling dalam Konteks PBO

- Membaca dan menulis file menggunakan objek.
- Serialisasi dan deserialisasi objek.

8. Penggunaan Modul dan Package

- Pembuatan dan penggunaan modul.
- Membuat dan mengelola package.

9. GUI Programming dengan Tkinter

- Pengenalan GUI programming.
- Membuat aplikasi sederhana menggunakan Tkinter.

10. Testing dan Debugging - Konsep dasar testing. - Menerapkan unit testing dengan unittest. - Strategi debugging dalam PBO.

11. Database dan Pemrograman Berorientasi Objek - Penggunaan database dalam PBO. - Menerapkan CRUD operations menggunakan objek. - Penggunaan ORM (Object Relational Mapping).

12. Project PBO Sederhana - Diskusi dan perencanaan proyek. - Implementasi proyek menggunakan konsep PBO. - Presentasi dan evaluasi proyek.

13. Review dan Konsolidasi - Mereview materi-materi sebelumnya. - Diskusi kasus-kasus nyata dan aplikasi PBO. - Evaluasi pemahaman dan penyelesaian tugas terakhir.

14. Ujian Akhir - Ujian tertulis. - Evaluasi proyek dan presentasi.

VI. Metode Pembelajaran:

- Ceramah

- Diskusi Kelompok
- Demonstrasi
- Praktek Langsung
- Tugas Individu dan Kelompok
- Proyek Akhir

VII. Penilaian:

- Ujian Tengah Semester
- Tugas Individu dan Kelompok
- Ujian Akhir Semester
- Penilaian Proyek

Kompetensi Dasar	Indikator Pencapaian Kompetensi	Materi Pokok	Alokasi Waktu (JP)	Kegiatan Pembelajaran
1	2	3	4	5
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.	<ul style="list-style-type: none"> • Konsep pemrograman berorientasi obyek. • Prosedur pemrograman berorientasi obyek. • Keunggulan pemrograman berorientasi obyek terhadap pemrograman struktural. 	8	<ul style="list-style-type: none"> • 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 4.2 Membuat kode program dasar	3.2.1 Menjelaskan berbagai tools pengembang program berorientasi obyek. 3.2.2 Menjelaskan konfigurasi tools pengembang	<ul style="list-style-type: none"> • Macam-macam perangkat lunak pengembang program berorientasi obyek. • Prosedur 	24	<ul style="list-style-type: none"> • Mengamati untuk mengidentifikasi dan merumuskan masalah tentang dasar dan aturan pemrograman berorientasi objek. • Mengumpulkan

<p>dan aturan pemrograman berorientasi objek</p>	<p>perangkat lunak berorientasi obyek.</p> <p>3.2.3 Menjelaskan kebutuhan tools editor program berorientasi obyek.</p> <p>3.2.4 Menjelaskan struktur dasar dalam pemrograman berorientasi obyek.</p> <p>3.2.5 Menjelaskan prosedur kompilasi program.</p> <p>3.2.6 Menjelaskan tata cara menjalankan program hasil kompilasi.</p> <p>3.2.7 Menerapkan prosedur aturan pemrograman berorientasi obyek.</p> <p>4.2.1 Menginstalasi tools pengembang program aplikasi berorientasi obyek kedalam komputer.</p> <p>4.2.2 Mengkonfigurasi tools pengembang program aplikasi berorientasi obyek.</p> <p>4.2.3 Merencanakan program aplikasi sederhana berorientasi obyek.</p> <p>4.2.4 Membuat kode program aplikasi sederhana berorientasi obyek.</p> <p>4.2.5 Menjalankan program aplikasi sederhana berorientasi obyek.</p>	<p>instalasi dan konfigurasi tools pengembang aplikasi program berorientasi obyek.</p> <ul style="list-style-type: none"> • Prosedur Kompilasi • Menjalankan program hasil kompilasi. • Prosedur patching komputer. • Aplikasi editor • Virtual machine 		<p>data tentang data dan aturan pemrograman berorientasi obyek</p> <ul style="list-style-type: none"> • Mengolah data tentang dasar d aturan pemrograman berorientasi obyek • Mengomunikasikan tentang dasar d aturan pemrograman berorientasi obyek
<p>3.3 Menerapkan konsep <i>class</i>, <i>method</i> dan <i>object</i></p> <p>4.3 Membuat <i>class</i>, <i>method</i> dan <i>object</i></p>	<p>3.3.1 Menjelaskan konsep class dalam pemrograman berorientasi obyek.</p> <p>3.3.2 Menerapkan class dalam pemrograman berorientasi obyek.</p> <p>3.3.3 Menjelaskan konsep method dalam pemrograman berorientasi obyek.</p>	<ul style="list-style-type: none"> • Konsep Class, object, method dalam oop. • Penamaan file dalam oop • Penerapan class dan method dalam program aplikasi oop. 	32	<ul style="list-style-type: none"> • Mengamati untuk mengidentifikasi dan merumuskan masalah tentang konsep <i>class</i>, <i>method</i> dan <i>object</i> • Mengumpulkan data tentang konsep <i>class</i>, <i>method</i> dan <i>object</i> • Mengolah data tentang konsep

	<p>3.3.4 Menerapkan method dalam pemrograman berorientasi obyek.</p> <p>3.3.5 Menjelaskan konsep obyek dalam pemrograman berorientasi obyek.</p> <p>3.3.6 Menerapkan obyek dalam pemrograman berorientasi obyek.</p> <p>4.3.1 Merancang program aplikasi berorientasi obyek dengan penerapan class, method dan obyek.</p> <p>4.3.2 Membuat kode program aplikasi berorientasi obyek dengan penerapan class, method dan obyek.</p> <p>4.3.3 Menguji program aplikasi berorientasi obyek yang menerapkan class, method dan obyek.</p>			<p><i>class, method dan object</i></p> <ul style="list-style-type: none"> • Mengomunikasikan tentang konsep <i>class, method dan object</i>
<p>3.4 Menerapkan konsep enkapsulasi dalam melindungi data dan informasi</p> <p>4.4 Membuat kode program enkapsulasi dalam melindungi data</p>	<p>3.4.1 Menjelaskan konsep dan tujuan proses enkapsulasi dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.4.2 Menjelaskan prosedur enkapsulasi dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.4.3 Menjelaskan persyaratan proses enkapsulasi dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.4.4 Menerapkan prosedur enkapsulasi dalam pemrograman aplikasi berorientasi obyek.</p> <p>4.4.1 Merancang program aplikasi berorientasi obyek dengan penerapan enkapsulasi dalam</p>	<ul style="list-style-type: none"> • Konsep enkapsulasi dalam program aplikasi berbasis oop. • Prosedur enkapsulasi dalam program aplikasi berbasis oop. • Prosedur rancangan program aplikasi berbasis oop dengan penerapan enkapsulasi. • Keunggulan enkapsulasi dalam program aplikasi berbasis oop 	32	<ul style="list-style-type: none"> • Mengamati untuk mengidentifikasi dan merumuskan masalah tentang konsep enkapsulasi dalam melindungi data dan informasi • Mengumpulkan data tentang konsep enkapsulasi dalam melindungi data dan informasi • Mengolah data tentang konsep enkapsulasi dalam melindungi data dan informasi • Mengomunikasikan tentang konsep enkapsulasi dalam melindungi data dan informasi

	<p>melindungi data dan informasi.</p> <p>4.4.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan enkapsulasi.</p> <p>4.4.3 Menguji program aplikasi berorientasi obyek yang menerapkan enkapsulasi.</p>			
<p>3.5 Menerapkan konsep pewarisan</p> <p>4.5 Membuat hubungan antara <i>class</i> dengan pola pewarisan</p>	<p>3.5.1 Menjelaskan konsep dan tujuan proses pewarisan dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.5.2 Menjelaskan prosedur pewarisan dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.5.3 Menjelaskan keterkaitan antara class dan pewarisan dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.5.4 Menerapkan prosedur pewarisan dalam pemrograman aplikasi berorientasi obyek</p> <p>4.5.1 Merancang program aplikasi berorientasi obyek dengan penerapan pewarisan dalam melindungi data dan informasi.</p> <p>4.5.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan pewarisan.</p> <p>4.5.3 Menguji program aplikasi berorientasi obyek yang menerapkan pewarisan.</p>	<ul style="list-style-type: none"> • Konsep enheritance (pewarisan) dalam program aplikasi berbasis oop. • 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 	32	<ul style="list-style-type: none"> • Mengamati untuk mengidentifikasi dan merumuskan masalah tentang konsep pewarisan • 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>	3.6.1 Menjelaskan konsep <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.	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang konsep <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i>
4.6 Membuat kode program sesuai konsep <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i>	3.6.2 Menjelaskan prosedur <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i> dalam pemrograman aplikasi berorientasi obyek.	• Prosedur <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i> dalam program aplikasi berbasis oop.		• 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	• Prosedur rancangan program aplikasi berbasis oop dengan penerapan <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i> .		• Mengolah data tentang konsep <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i>
	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.	• Keunggulan <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i> dalam program aplikasi berbasis oop		• Mengomunikasikan tentang konsep <i>polymorphism</i> dengan <i>overloading</i> dan <i>overriding</i>
	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> .			
3.7 Menerapkan penggunaan <i>package</i> dalam aplikasi	3.7.1 Menjelaskan konsep <i>package</i> dalam pemrograman aplikasi berorientasi obyek.	• Konsep <i>package</i> dalam program aplikasi berbasis oop.	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang penggunaan <i>package</i> dalam aplikasi
4.7 Membuat aplikasi melalui pengelompokan	3.7.2 Menjelaskan proses pengelompokan class dalam pembuatan <i>package</i> .	• Prosedur pembuatan <i>package</i> dalam program aplikasi		• Mengumpulkan data tentang

<p><i>class</i> dalam <i>package</i></p>	<p>3.7.3 Menjelaskan prosedur pembuatan <i>package</i> dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.7.4 Menjelaskan proses kompilasi <i>package</i>.</p> <p>3.7.5 Menerapkan <i>package</i> dalam pemrograman aplikasi berorientasi obyek</p> <p>4.7.1 Merancang program aplikasi berorientasi obyek dengan penerapan <i>package</i></p> <p>4.7.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan <i>package</i>.</p> <p>4.7.3 Menguji program aplikasi berorientasi obyek yang menerapkan <i>package</i>.</p>	<p>berbasis oop.</p> <ul style="list-style-type: none"> • Prosedur rancangan program aplikasi 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 		<p>penggunaan <i>package</i> dalam aplikasi</p> <ul style="list-style-type: none"> • Mengolah data tentang penggunaan <i>package</i> dalam aplikasi • Mengomunikasikan tentang penggunaan <i>package</i> dalam aplikasi
<p>3.8 Merancang pembuatan <i>interface</i></p> <p>4.8 Membuat rancangan aplikasi dengan <i>interface</i></p>	<p>3.8.1 Menjelaskan konsep <i>interface</i> dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.8.2 Menjelaskan prosedur pembuatan <i>interface</i> dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.8.3 Menerapkan <i>interface</i> dalam pemrograman aplikasi berorientasi obyek</p> <p>4.8.1 Merancang program aplikasi berorientasi obyek dengan penerapan <i>interface</i></p> <p>4.8.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan <i>interface</i>.</p> <p>4.8.3 Menguji program aplikasi berorientasi obyek yang</p>	<ul style="list-style-type: none"> • 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. • Keunggulan <i>interface</i> dalam program aplikasi berbasis oop 	<p>32</p>	<ul style="list-style-type: none"> • 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>

	menerapkan <i>interface</i> .			
3.9 Menganalisis pemanfaatan <i>class library</i>	3.9.1 Menjelaskan konsep <i>class library</i> dalam pemrograman aplikasi berorientasi obyek.	• Konsep <i>class library</i> dalam program aplikasi berbasis oop.	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang pemanfaatan <i>class library</i>
4.9 Menggunakan <i>class library</i> dalam memecahkan masalah	3.9.2 Menjelaskan prosedur pemanfaatan <i>class library</i> dalam pemrograman aplikasi berorientasi obyek.	• Prosedur penerapan <i>class library</i> dalam program aplikasi berbasis oop.		• Mengumpulkan data tentang pemanfaatan <i>class library</i>
	3.9.3 Menganalisis pemanfaatan <i>class library</i> dalam pemrograman aplikasi berorientasi obyek.	• Prosedur rancangan program aplikasi berbasis oop dengan penerapan <i>class library</i> .		• Mengolah data tentang pemanfaatan <i>class library</i>
	3.9.4 Menentukan penggunaan <i>class library</i> dalam pemrograman aplikasi berorientasi obyek.	• Prosedur kompilasi program aplikasi dengan penerapan <i>class library</i>		• Mengomunikasikan tentang pemanfaatan <i>class library</i>
	3.9.5 Menerapkan <i>class library</i> dalam pemrograman aplikasi berorientasi obyek.	• Keunggulan program dengan penerapan <i>class library</i> dalam program aplikasi berbasis oop		
	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> .			
3.10 Mengevaluasi penanganan kesalahan dalam program	3.10.1 Menjelaskan konsep <i>error handling</i> (penanganan kesalahan) dalam pemrograman aplikasi berorientasi obyek.	• Konsep penanganan kesalahan dalam program aplikasi berbasis oop.	32	• Mengamati untuk mengidentifikasi dan merumuskan masalah tentang penanganan kesalahan dalam program
4.10 Memodifikasi kode		• Prosedur		

<p>program untuk penanganan kesalahan dalam program</p>	<p>3.10.2 Menjelaskan prosedur <i>error handling</i> (penanganan kesalahan) dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.10.3 Menganalisis penanganan kesalahan (<i>error handling</i>) dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.10.4 Menentukan penggunaan <i>error handling</i> (penanganan kesalahan) dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.10.5 Menerapkan <i>error handling</i> (penanganan kesalahan) dalam pemrograman aplikasi berorientasi obyek.</p> <p>4.10.1 Merancang program aplikasi berorientasi obyek dengan penerapan <i>error handling</i> (penanganan kesalahan).</p> <p>4.10.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan <i>error handling</i> (penanganan kesalahan).</p> <p>4.10.3 Menguji program aplikasi berorientasi obyek yang menerapkan <i>error handling</i> (penanganan kesalahan).</p> <p>4.10.4 Memperbaiki penanganan kesalahan (<i>error handling</i>) dalam program aplikasi berorientasi obyek.</p> <p>4.10.5 Memodifikasi</p>	<p>penangan kesalahan (<i>error handling</i>) dalam pemrograman berbasis oop.</p> <ul style="list-style-type: none"> • Prosedur pembuatan program aplikasi berbasis oop dengan penanganan kesalahan 	<ul style="list-style-type: none"> • Mengumpulkan data tentang penanganan kesalahan dalam program • Mengolah data tentang penanganan kesalahan dalam program • Mengomunikasikan tentang penanganan kesalahan dalam program
---	---	--	---

	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. 4.11.2 Membuat kode program aplikasi berorientasi obyek	<ul style="list-style-type: none"> • Konsep penggunaan data tipe string dalam pemrograman berbasis oop. • Prosedur pembuatan program berbasis oop dengan menerapkan data tipe string. 	8	<ul style="list-style-type: none"> • 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

		<p>yang menerapkan data bentuk string dan pelbagai propertinya.</p> <p>4.11.3 Menguji program aplikasi berorientasi obyek yang menerapkan data bentuk string dan pelbagai propertinya.</p>		
<p>3.12 Menerapkan <i>collection</i> sebagai media penyimpanan data</p> <p>4.12 Membuat kode program <i>collection</i> sebagai penyimpan data</p>	<p>3.12.1 Menjelaskan konsep <i>collection</i> sebagai media penyimpanan data dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.12.2 Menjelaskan prosedur <i>collection</i> sebagai media penyimpanan data dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.12.3 Menentukan penggunaan <i>collection</i> sebagai media penyimpanan data dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.12.4 Menerapkan <i>collection</i> sebagai media penyimpanan data dalam pemrograman aplikasi berorientasi obyek.</p> <p>4.12.1 Merancang program aplikasi berorientasi obyek dengan</p>	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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.

	<p>penerapan <i>collection</i> sebagai media penyimpanan data.</p> <p>4.12.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan <i>collection</i> sebagai media penyimpanan data.</p> <p>4.12.3 Menguji program aplikasi berorientasi obyek yang menerapkan <i>collection</i> sebagai media penyimpanan data.</p>			
<p>3.13 Menerapkan operasi <i>file</i> dan <i>Input Output</i>(IO)</p> <p>4.13 Membuat kode program untuk operasi <i>file</i> dan operasi <i>Input Output</i></p>	<p>3.13.1 Menjelaskan konsep operasi <i>file</i> dan <i>Input Output</i>(IO) dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.13.2 Menjelaskan prosedur operasi <i>file</i> dan <i>Input Output</i>(IO) dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.13.3 Menentukan penggunaan operasi <i>file</i> dan <i>Input Output</i>(IO) dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.13.4 Menerapkan operasi <i>file</i> dan <i>Input Output</i>(IO) pemrograman aplikasi berorientasi obyek.</p> <p>4.13.1 Merancang program aplikasi berorientasi obyek dengan penerapan operasi <i>file</i> dan <i>Input Output</i>(IO).</p> <p>4.13.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan operasi <i>file</i> dan <i>Input Output</i>(IO).</p> <p>4.13.3 Menguji program</p>	<ul style="list-style-type: none"> • Konsep operasi <i>file</i> dalam pemrograman berbasis oop. • Konsep input dan output dalam pemrograman aplikasi berbasis oop. • Prosedur pembuatan program aplikasi berbasis oop untuk operasi <i>file</i>. • Prosedur pembuatan program aplikasi berbasis oop untuk input dan output. • Membuat program aplikasi berbasis oop untuk operasi <i>file</i> yang dapat mengedit, menambah, menghapus dan membaca isi <i>file</i>. 	24	<ul style="list-style-type: none"> • 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 dan <i>Input Output</i>(IO) • Mengomunikasikan tentang operasi dan <i>Input Output</i>(IO)

		aplikasi berorientasi obyek yang menerapkan operasi <i>file</i> dan <i>Input Output</i> (IO).		
3.14 Menerapkan dasar dasar <i>thread</i> 4.14 <i>Membuat</i> 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> .	<ul style="list-style-type: none"> • Konsep dasar thread dalam oop. • Prosedur penggunaan thread dalam oop • Membuat program oop menerapkan thread. 	32	<ul style="list-style-type: none"> • Mengamati untuk mengidentifikasi dan merumuskan masalah tentang dasar dasar <i>thread</i> • Mengumpulkan data tentang da dasar <i>thread</i> • Mengolah data tentang dasar dasar <i>thread</i> • Mengomunikas tentang dasar dasar <i>thread</i>
3.15 Merancang antar muka atau <i>User Interface</i> (UI) 4.15 Membuat kode program untuk antar muka atau	3.15.1 Menjelaskan konsep antar muka atau <i>User Interface</i> (UI) dalam pemrograman aplikasi berorientasi obyek. 3.15.2 Menjelaskan prosedur antar muka atau <i>User</i>	<ul style="list-style-type: none"> • Konsep antar muka (<i>user interface</i>) dalam pemrograman aplikasi berbasis oop • Prosedur 	32	<ul style="list-style-type: none"> • Mengamati untuk mengidentifikasi dan merumuskan masalah tentang antar muka atau <i>User Interface</i> (UI) • Mengumpulkan

<p><i>User Interface (UI)</i></p>	<p><i>Interface (UI)</i> dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.15.3 Menentukan penggunaan antar muka atau <i>User Interface (UI)</i> dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.15.4 Menerapkan antar muka atau <i>User Interface (UI)</i> pemrograman aplikasi berorientasi obyek.</p> <p>4.15.1 Merancang program aplikasi berorientasi obyek dengan penerapan antar muka atau <i>User Interface (UI)</i>.</p> <p>4.15.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan antar muka atau <i>User Interface (UI)</i>.</p> <p>4.15.3 Menguji program aplikasi berorientasi obyek yang menerapkan antar muka atau <i>User Interface (UI)</i>.</p>	<p>pembuatan <i>user interface</i> dalam pemrograman aplikasi berbasis oop.</p> <ul style="list-style-type: none"> • Membuat contoh program aplikasi yang menerapkan <i>user interface</i>. 		<p>data tentang antar muka atau <i>User Interface (UI)</i></p> <ul style="list-style-type: none"> • Mengolah data tentang antar muka atau <i>User Interface (UI)</i> • Mengomunikasikan tentang antar muka atau <i>User Interface (UI)</i>
<p>3.16 Menerapkan konektifitas dalam akses basis data</p> <p>4.16 Membuat konektifitas basis data</p>	<p>3.16.1 Menjelaskan konsep konektifitas akses basis data dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.16.2 Menjelaskan prosedur akses basis data menggunakan <i>library</i> konektor dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.16.3 Menentukan penggunaan konektor basis data dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.16.4 Menerapkan library</p>	<ul style="list-style-type: none"> • Konsep konektifitas untuk akses basis data dalam program aplikasi berbasis oop. • Prosedur koneksi basis data dengan program aplikasi berbasis oop. • Penggunaan konektor basis data dalam program aplikasi. • Membuat contoh 	<p>32</p>	<ul style="list-style-type: none"> • Mengamati untuk mengidentifikasi dan merumuskan masalah tentang konektifitas dalam akses basis data. • Mengumpulkan data tentang konektifitas dalam akses basis data. • Mengolah data tentang konektifitas dalam akses basis data. • Mengomunikasikan tentang konektifitas dalam akses basis data.

	<p>konektor akses basis data dalam pemrograman aplikasi berorientasi obyek.</p> <p>4.16.1 Merancang program aplikasi berorientasi obyek akses basis data menggunakan <i>library konektor</i>.</p> <p>4.16.2 Membuat kode program aplikasi berorientasi obyek akses basis data menggunakan <i>library konektor</i>.</p> <p>4.16.3 Menguji program aplikasi berorientasi obyek akses basis data yang menggunakan <i>library konektor</i>.</p>	program aplikasi berbasis oop untuk koneksi basis data.		
<p>3.17 Merancang <i>User Interface</i> menggunakan <i>library</i></p> <p>4.17 Mendesain <i>User Interface</i> menggunakan <i>library</i></p>	<p>3.17.1 Menjelaskan konsep antar muka atau <i>User Interface</i> menggunakan <i>library</i> dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.17.2 Menjelaskan prosedur antar muka atau <i>User Interface</i> menggunakan <i>library</i> dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.17.3 Menentukan penggunaan antar muka atau <i>User Interface</i> menggunakan <i>library</i> dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.17.4 Menerapkan antar muka atau <i>User Interface</i> menggunakan <i>library</i> dalam pemrograman aplikasi berorientasi obyek.</p>	<ul style="list-style-type: none"> • Konsep pembuatan <i>user interface</i> menggunakan <i>library</i> bawaan dalam pemrograman aplikasi berbasis oop • Prosedur pembuatan <i>user interface</i> dengan menggunakan <i>library</i> dalam pemrograman aplikasi berbasis oop 	32	<ul style="list-style-type: none"> • Mengamati untuk mengidentifikasi dan merumuskan masalah tentang <i>User Interface</i> menggunakan <i>library</i> • Mengumpulkan data tentang <i>User Interface</i> 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>

	<p>4.17.1 Merancang program aplikasi berorientasi obyek dengan penerapan antar muka atau <i>User Interface</i> menggunakan <i>library</i>.</p> <p>4.17.2 Membuat kode program aplikasi berorientasi obyek yang menerapkan antar muka atau <i>User Interface</i> menggunakan <i>library</i>.</p> <p>4.17.3 Menguji program aplikasi berorientasi obyek yang menerapkan antar muka atau <i>User Interface</i> menggunakan <i>library</i>.</p>			
<p>3.18 Menerapkan objek multimedia dalam aplikasi</p> <p>4.18 Membuat aplikasi yang dapat menampilkan gambar, audio dan video</p>	<p>3.18.1 Menjelaskan konsep objek multimedia dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.18.2 Menjelaskan prosedur objek multimedia dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.18.3 Menentukan penggunaan objek multimedia dalam pemrograman aplikasi berorientasi obyek.</p> <p>3.18.4 Menerapkan objek multimedia dalam pemrograman aplikasi berorientasi obyek.</p> <p>4.18.1 Merancang program aplikasi berorientasi obyek dengan penerapan obyek multimedia.</p> <p>4.18.2 Membuat kode program aplikasi berorientasi obyek dengan penerapan</p>	<ul style="list-style-type: none"> • Konsep perenapan multi media dalam pemrograman aplikasi berbasis oop • Prosedur penerapan akses multimedia dalam pemrograman aplikasi berbasis oop. • Membuat contoh program aplikasi berbasis oop dengan menerapkan multi media. 	32	<ul style="list-style-type: none"> • Mengamati untuk mengidentifikasi dan merumuskan masalah tentang objek multimedia dalam aplikasi • Mengumpulkan data tentang ob multimedia dal aplikasi • Mengolah data tentang objek multimedia dal aplikasi • Mengomunikas tentang objek multimedia dal aplikasi

	obyek multimedia. 4.18.3 Menguji program aplikasi berorientasi obyek yang menerapkan obyek multimedia.			
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 View Controller</i> (MVC) untuk pemrograman aplikasi berorientasi obyek. 4.19.1 Mempresentasikan rancangan program aplikasi berorientasi obyek berbasis MVC.	<ul style="list-style-type: none"> Konsep penggunaan aplikasi <i>model view controller</i> (MVC) dalam pemrograman aplikasi berbasis oop. Prosedur penggunaan MVC dalam pemrograman aplikasi berbasis oop Macam-macam aplikasi mvc untuk oop. Membuat program aplikasi sederhana berbasis oop menggunakan MVC 	32	<ul style="list-style-type: none"> 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> (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).	<ul style="list-style-type: none"> Konsep pengembangan program aplikasi berbasis oop menggunakan MVC. Mengembangkan program aplikasi oop dengan menggunakan MVC 	32	<ul style="list-style-type: none"> 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 dalam aplikasi dengan pelbagai hasil

	<p>4.20.2 Membuat kode program aplikasi berorientasi obyek berbasis <i>Model View Controller</i> (MVC).</p> <p>4.20.3 Menguji program aplikasi berorientasi obyek berbasis <i>Model View Controller</i> (MVC).</p>			<ul style="list-style-type: none"> • Mengomunikas tentang aplikas berbasis MVC dalam aplikasi dengan pelbaga hasil
--	--	--	--	---

Pertemuan 1: Pengenalan Pemrograman Berorientasi Objek dan Python

- Konsep dasar Pemrograman Berorientasi Objek (PBO)
- Perkenalan bahasa pemrograman Python
- Instalasi dan pengaturan lingkungan pengembangan Python

Pertemuan 2: Dasar-dasar Python dan Struktur Program Dan Tugas

- Variabel dan tipe data dalam Python
- Struktur kontrol (if, else, dan elif)
- Perulangan (for, while)
- Fungsi dan prosedur dalam Python

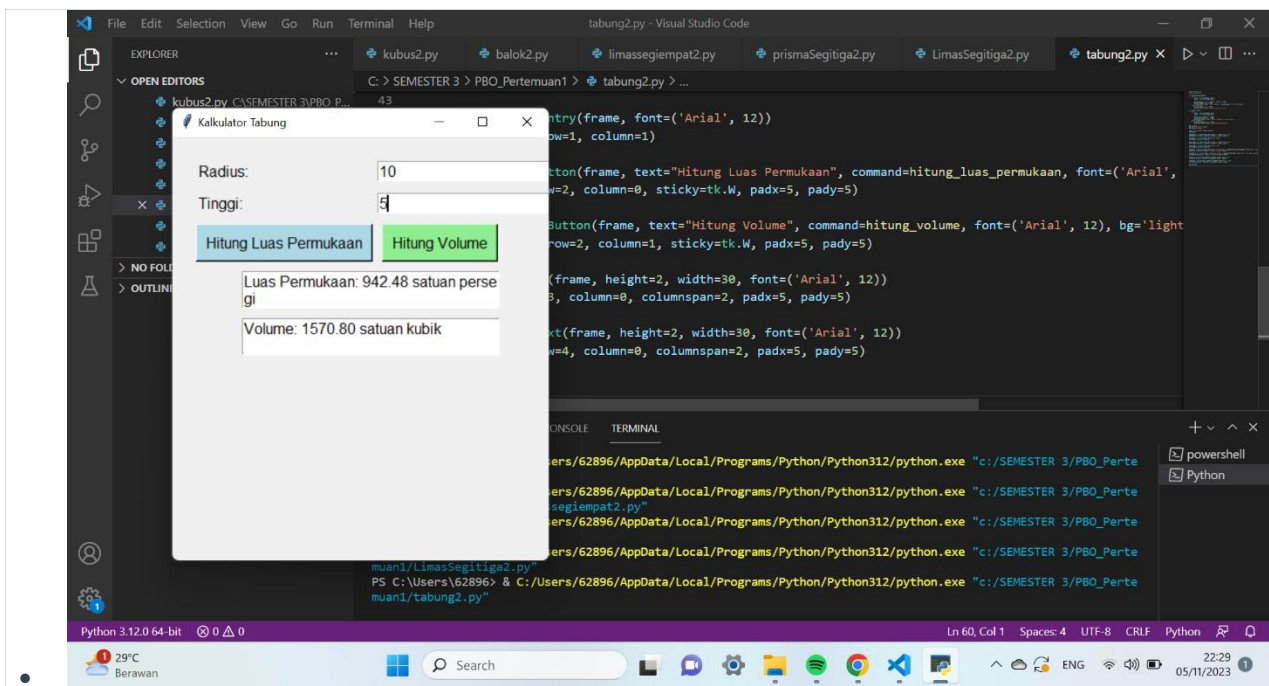


If some text appears garbled, you can change th...



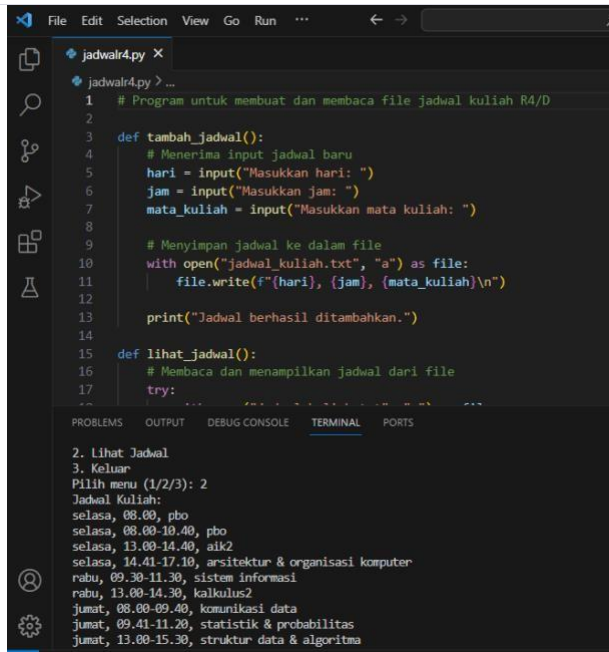
1. Jelaskan apa itu pemrograman python
 2. Jelaskan sejarah singkat python
 3. Bagaimana proses instalasi python
 4. Bagaimana cara memastikan keberhasilan instalasi python
 5. Jelaskan Tipe Data yang ada di Python
 6. Jelaskan tentang variabel di python dan berikan contohnya
 7. Jelaskan Operator yang ada di python dan berikan contohnya
 8. Jelaskan tentang IF (kondisi) dalam pemrograman python serta berikan contohnya
 9. Jelaskan tentang Pengulangan (Loop) serta berikan contohnya
 10. Jelaskan tentang List, Tuple, Set, dan Dictionary serta berikan contohnya
- Jawaban ditulis tangan dengan jelas (dapat dibaca orang lain) di atas kertas folio bergaris, dikumpulkan paling lambat hari sabtu tanggal 28 oktober 2023 jam 12.00 siang.

Pertemuan 3: Program Kalkulator Bangun Ruang



Pertemuan 4: Membuat Class dan Objek di Python

- Membuat jadwal kuliah sederhana



```
File Edit Selection View Go Run ...
jadwal4.py x
jadwal4.py > ...
1 # Program untuk membuat dan membaca file jadwal kuliah R4/D
2
3 def tambah_jadwal():
4     # Menerima input jadwal baru
5     hari = input("Masukkan hari: ")
6     jam = input("Masukkan jam: ")
7     mata_kuliah = input("Masukkan mata kuliah: ")
8
9     # Menyimpan jadwal ke dalam file
10    with open("jadwal_kuliah.txt", "a") as file:
11        file.write(f"{hari}, {jam}, {mata_kuliah}\n")
12
13    print("Jadwal berhasil ditambahkan.")
14
15 def lihat_jadwal():
16     # Membaca dan menampilkan jadwal dari file
17     try:
18         with open("jadwal_kuliah.txt", "r") as file:
19             for line in file:
20                 print(line)
21     except FileNotFoundError:
22         print("Jadwal belum ditambahkan.")
23
24 if __name__ == "__main__":
25     pilih_menu = input("Pilih menu (1/2/3): ")
26     if pilih_menu == "1":
27         tambah_jadwal()
28     elif pilih_menu == "2":
29         lihat_jadwal()
30     elif pilih_menu == "3":
31         exit()
32     else:
33         print("Pilih menu yang valid.")
34
35 2. Lihat Jadwal
36 3. Keluar
37 Pilih menu (1/2/3): 2
38 Jadwal Kuliah:
39 Selasa, 08.00, pbo
40 Selasa, 08.00-10.40, pbo
41 Selasa, 13.00-14.40, aik2
42 Selasa, 14.41-17.10, arsitektur & organisasi komputer
43 Rabu, 09.30-11.30, sistem informasi
44 Rabu, 13.00-14.30, kalkulus2
45 Jumat, 08.00-09.40, komunikasi data
46 Jumat, 09.41-11.20, statistik & probabilitas
47 Jumat, 13.00-15.30, struktur data & algoritma
```

Pertemuan 6: Translator

A. Translator

Untuk membuat translator sederhana pada Python, kita dapat menggunakan pustaka seperti `googletrans` yang memanfaatkan layanan terjemahan Google Translate. Kode bahasa di setiap negara berbeda, tuliskan kode bahasa yang diperlukan, misalnya 'id' untuk Indonesia, 'en' untuk Inggris, 'ru' untuk Russian dan sebagainya. Pastikan untuk menginstal pustaka tersebut terlebih dahulu dengan perintah: ((`pip install googletrans==4.0.0-rc1`))

Berikut contoh pengaplikasian translator

```

65         # menampilkan hasil terjemahan
66         self.txtHasil1.insert(END,hasil1.text)
67         self.txtHasil2.insert(END,hasil2.text)
68         self.txtHasil3.insert(END,hasil3.text)
69         self.txtHasil4.insert(END,hasil4.text)
70
71     def onKeluar(self, event=None):
72         # memberikan perintah menutup aplikasi
73         self.parent.destroy()
74
75     if __name__ == '__main__':
76         root = Tk()
77         aplikasi = FrmTranslator(root, "Program Translator")
78         root.mainloop()

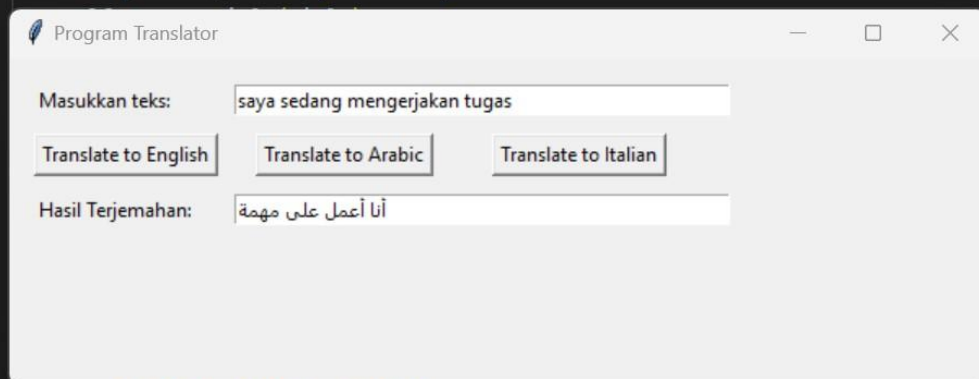
```

Translator > FrmTranslator.py > ...

```

1  from tkinter import Frame, Label, Entry, Button, YES, BOTH, END, Tk, W
2  from googletrans import Translator
3
4  class Translate:
5      def __init__(self, parent, title):
6          self.parent = parent
7          self.parent.geometry("600x200")
8
9
10
11
12
13
14
15
16
17
18
19
20
21

```



Pertemuan 7:

A. Playsound

Pustaka playsound dapat digunakan untuk memutar file suara (MP3) di Python. Yang perlu disiapkan terlebih dahulu adalah dengan menginstall ((`pip install playsound`))


```

1  from playsound import playsound
2
3  playsound("memories.mp3", True)

```

B. Play MP4

Pustaka opencv-python dapat digunakan untuk memutar video di Python.

Yang perlu disiapkan terlebih dahulu adalah dengan menginstall (([pip install opencv-python](#)))

```

1  # importing libraries
2  import cv2
3  import numpy as np
4
5  # Create a VideoCapture object and read from input file
6  cap = cv2.VideoCapture('media/video1.mp4')
7
8  # Check if camera opened successfully
9  if (cap.isOpened() == False):
10     print("Error opening video file")
11
12  # Read until video is completed
13  while(cap.isOpened()):
14
15     # Capture frame-by-frame
16     ret, frame = cap.read()
17     if ret == True:
18         # Display the resulting frame
19         cv2.imshow('Frame', frame)
20
21         # Press Q on keyboard to exit
22         if cv2.waitKey(25) & 0xFF == ord('q'):
23             break
24
25     # Break the loop
26     else:
27         break
28
29  # When everything done, release
30  # the video capture object
31  cap.release()
32
33  # Closes all the frames
34  cv2.destroyAllWindows()

```

C. Text to Sound

Pustaka gTTS memungkinkan konversi teks ke suara menggunakan layanan Text-to-Speech dari Google. Yang perlu disiapkan terlebih dahulu adalah dengan menginstall ((`pip install gtts`))

```
1 from gtts import gTTS
2 from playsound import playsound
3
4 mytext = 'Selamat Datang di Fakultas Teknik! Namaku Nadzwa Nurul Hikmah dari Teknik Informatika'
5 language = 'id'
6 myobj = gTTS(text=mytext, lang=language, slow=False)
7
8 myobj.save("selamat_datang.mp3")
9 playsound("selamat_datang.mp3", True)
```

D. Text from Image

Pustaka pytesseract dapat digunakan untuk mengekstraksi teks dari gambar menggunakan Tesseract OCR. Yang perlu disiapkan terlebih dahulu adalah dengan menginstall ((`pip install pytesseract`)). Juga perlu menginstal Tesseract OCR pada system. Unduh dari dan atur pathnya. ((<https://github.com/tesseract-ocr/tesseract>))

Pertemuan 8: File Handling dalam Konteks PBO

- Membaca dan menulis file menggunakan objek
- Serialisasi dan deserialisasi objek
- Penggunaan file sebagai objek

Pertemuan 9: Penggunaan Modul dan Package

- Membuat dan mengimpor modul
- Membuat dan mengelola package
- Menggunakan modul dan package dalam proyek PBO

Pertemuan 10: Instal XAMPP

A. Penginstallan XAMPP

Instalasi XAMPP Control Panel melibatkan beberapa langkah sederhana. Berikut adalah panduan langkah demi langkah untuk menginstal XAMPP Control Panel:

1) Unduh XAMPP

- Kunjungi situs resmi XAMPP di ((
<https://www.apachefriends.org/>))
- Pilih versi XAMPP yang sesuai dengan sistem operasi Anda (Windows, Linux, atau macOS).

2) Jalankan Instalator

- Setelah pengunduhan selesai, buka file instalator yang diunduh.
- Ikuti petunjuk pada layar untuk menginstal XAMPP pada sistem Anda.

3) Pilih Komponen

- Pada saat instalasi, Anda akan diminta untuk memilih komponen yang ingin Anda instal. Pastikan Anda memilih "Apache" dan "MySQL", karena XAMPP Control Panel mengelola layanan ini.

4) Pilih Lokasi Instalasi

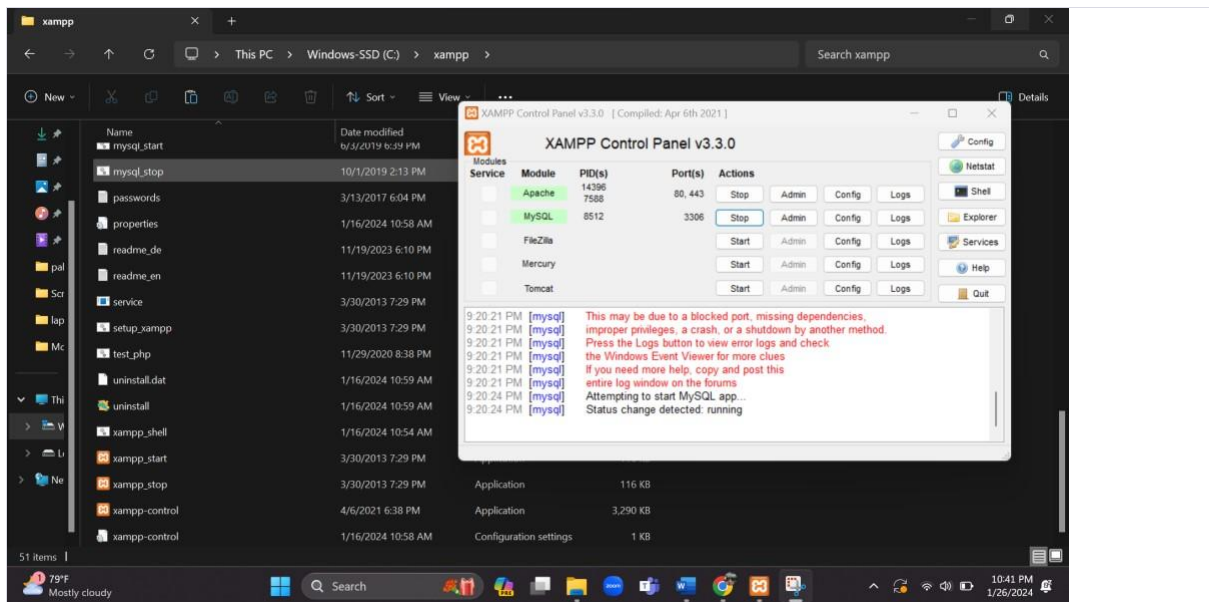
- Tentukan lokasi di mana Anda ingin menginstal XAMPP. Secara default, XAMPP akan diinstal di direktori `C:\xampp` pada sistem Windows.

5) Selesaikan Instalasi

- Lanjutkan dengan mengklik tombol "Next" dan ikuti langkahlangkah instalasi hingga selesai. 6) Mulai XAMPP Control Panel
- Setelah instalasi selesai, cari folder instalasi XAMPP dan buka folder tersebut. Di dalamnya, Anda akan menemukan file `xamppcontrol.exe`. Jalankan file ini untuk membuka XAMPP Control Panel.

7) Mulai Layanan Apache dan MySQL

- Pada XAMPP Control Panel, Anda akan melihat daftar layanan seperti Apache, MySQL, FileZilla, dll. Pastikan untuk memulai layanan Apache dan MySQL dengan mengklik tombol "Start" di sebelah keduanya. 8) Verifikasi Instalasi
- Buka browser web dan kunjungi `http://localhost` atau `http://127.0.0.1`. Jika instalasi berhasil, Anda akan melihat halaman selamat datang XAMPP.



Contoh hasil dari XAMPP :

```
Administrator: Command Prompt - mysql -u root -p
MariaDB [campus]> insert into matakuliah(kodemk,namamk,sks)values('1002','Bhs. Inggris',2);
Query OK, 1 row affected (0.126 sec)

MariaDB [campus]> insert into matakuliah(kodemk,namamk,sks)values('1003','Bhs. Indonesia',2);
Query OK, 1 row affected (0.066 sec)

MariaDB [campus]> insert into matakuliah(kodemk,namamk,sks)values('1004','Agama Islam',2);
Query OK, 1 row affected (0.078 sec)

MariaDB [campus]> select * from matakuliah;
+----+-----+-----+----+
| id | kodemk | namamk | sks |
+----+-----+-----+----+
| 1 | 1001 | PBO | 3 |
| 2 | 1002 | Bhs. Inggris | 2 |
| 3 | 1003 | Bhs. Indonesia | 2 |
| 4 | 1004 | Agama Islam | 2 |
+----+-----+-----+----+
4 rows in set (0.111 sec)

MariaDB [campus]> insert into matakuliah(kodemk,namamk,sks)values('1005','Algoritma',2);
Query OK, 1 row affected (0.068 sec)

MariaDB [campus]> insert into matakuliah(kodemk,namamk,sks)values('1006','Kewirausahaan',2);
Query OK, 1 row affected (0.041 sec)

MariaDB [campus]> insert into matakuliah(kodemk,namamk,sks)values('1007','Basis Data',3);
Query OK, 1 row affected (0.042 sec)

MariaDB [campus]> select * from matakuliah;
+----+-----+-----+----+
| id | kodemk | namamk | sks |
+----+-----+-----+----+
| 1 | 1001 | PBO | 3 |
| 2 | 1002 | Bhs. Inggris | 2 |
| 3 | 1003 | Bhs. Indonesia | 2 |
| 4 | 1004 | Agama Islam | 2 |
| 5 | 1005 | Algoritma | 2 |
| 6 | 1006 | Kewirausahaan | 2 |
| 7 | 1007 | Basis Data | 3 |
+----+-----+-----+----+
7 rows in set (0.001 sec)

MariaDB [campus]>
```

Pertemuan 11: MY SQL

MySQL adalah sebuah sistem manajemen basis data relasional (RDBMS) yang bersifat open-source. RDBMS adalah jenis basis data yang menyimpan data dalam bentuk tabel terstruktur yang terkait satu sama lain melalui kunci (key) atau hubungan (relationship). MySQL

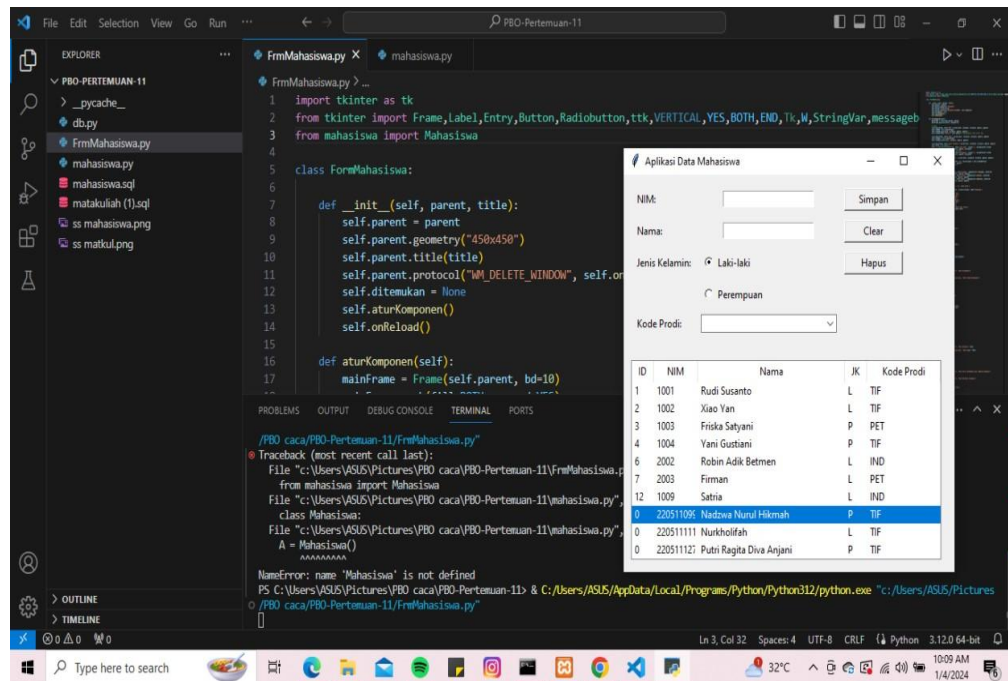
dikembangkan oleh perusahaan Swedia, MySQL AB, dan kemudian diakuisisi oleh Oracle Corporation.

Berikut beberapa konsep dasar MySQL:

- 1)** Database: MySQL menyimpan data dalam basis data. Basis data adalah kumpulan tabel yang saling terkait. Misalnya, sebuah toko online mungkin memiliki basis data yang berisi tabel untuk produk, pelanggan, dan pesanan.
- 2)** Tabel: Setiap basis data terdiri dari satu atau lebih tabel. Tabel adalah struktur utama yang menyimpan data. Setiap tabel terdiri dari kolom dan baris. Kolom menyimpan tipe data tertentu, seperti angka, teks, atau tanggal.
- 3)** Baris (Record): Baris dalam tabel disebut juga sebagai record. Ini merepresentasikan satu set data yang lengkap untuk suatu entitas. Sebagai contoh, jika Anda memiliki tabel pelanggan, setiap baris mungkin mewakili satu pelanggan dengan semua informasi terkait.
- 4)** Kunci (Key): Kunci adalah suatu cara untuk mengidentifikasi secara unik setiap record dalam tabel. Kunci utama (primary key) adalah kunci yang digunakan untuk mengidentifikasi record secara unik. Selain itu, ada juga kunci asing (foreign key) yang menghubungkan satu tabel dengan tabel lainnya.
- 5)** Query: Dalam MySQL, Anda menggunakan SQL (Structured Query Language) untuk berinteraksi dengan basis data. SQL adalah bahasa yang digunakan untuk mengambil, memasukkan, mengupdate, dan menghapus data dari tabel.
- 6)** Relasi: MySQL mendukung relasi antar tabel. Misalnya, Anda bisa memiliki tabel produk dan tabel pesanan yang terkait melalui kunci asing. Ini memungkinkan Anda untuk mengaitkan pesanan dengan produk yang dibeli.
- 7)** Open Source: MySQL bersifat open source, artinya kode sumbernya dapat diakses dan dimodifikasi oleh siapa pun. Ini membuat MySQL populer di kalangan pengembang perangkat lunak.
- 8)** Manajemen Pengguna dan Akses: MySQL memiliki sistem manajemen pengguna yang memungkinkan administrator untuk mengontrol akses ke basis data. Ini melibatkan pembuatan pengguna, memberikan hak akses, dan mengelola keamanan.

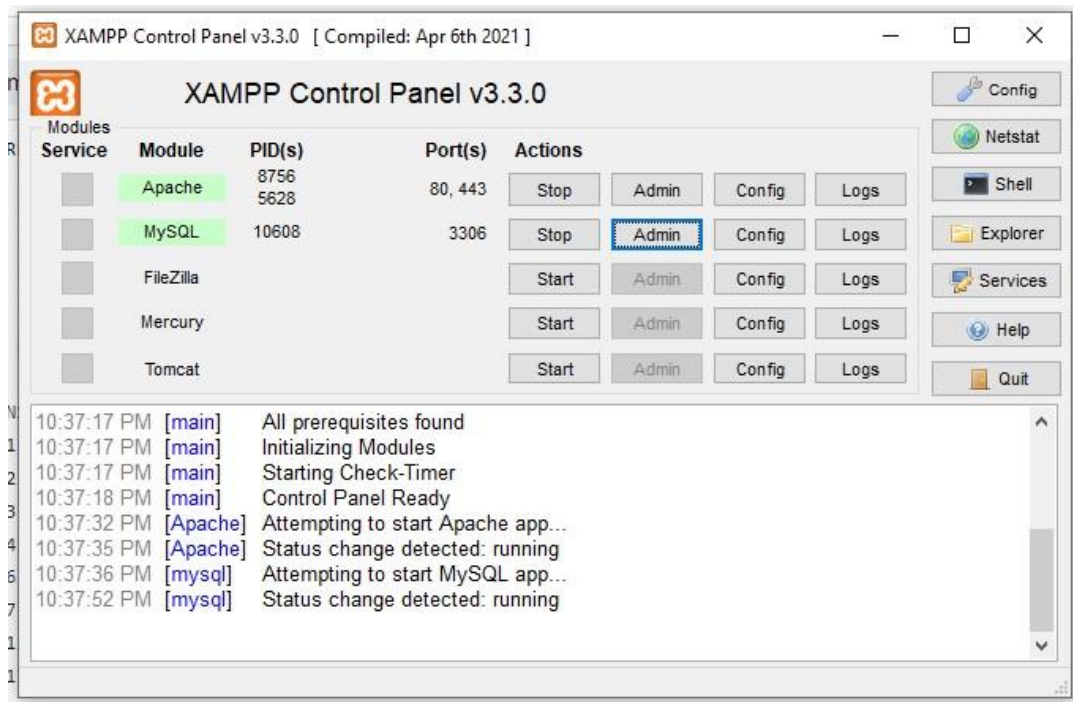
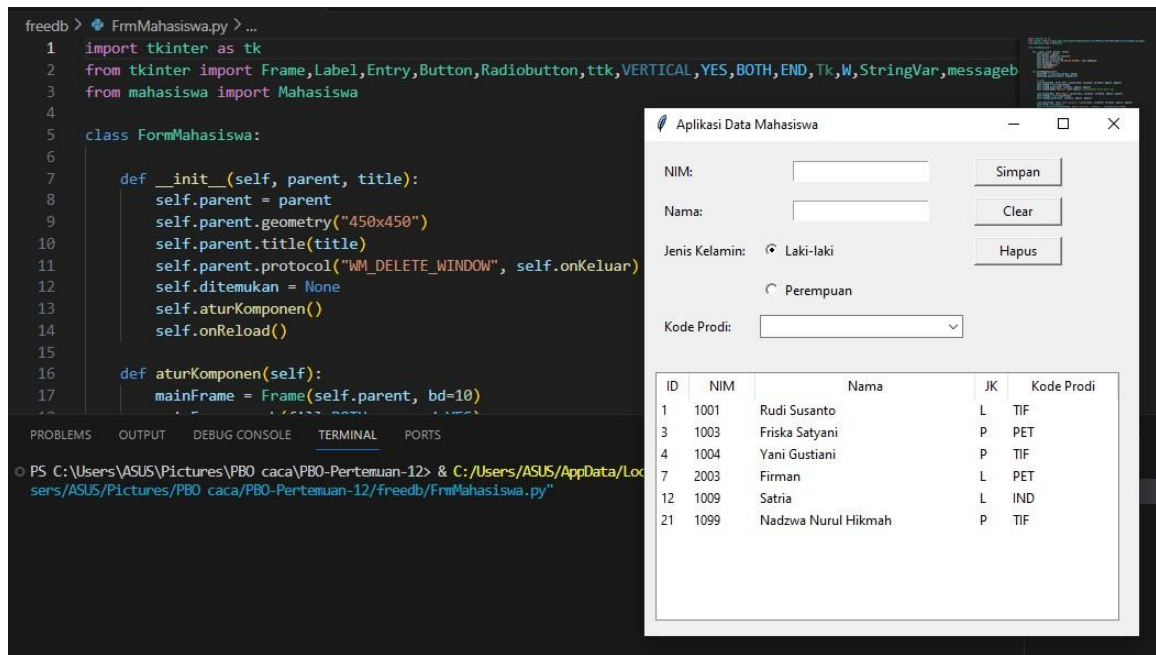
Penting mengetahui bahwa MySQL adalah bagian dari ekosistem basis data yang lebih besar, dan banyak aplikasi dan situs web menggunakan MySQL sebagai sistem manajemen basis data mereka. MySQL biasanya digunakan dengan bahasa pemrograman seperti PHP, Python, atau Java untuk mengembangkan aplikasi berbasis web. A.

Aplikasi Data Mahasiswa



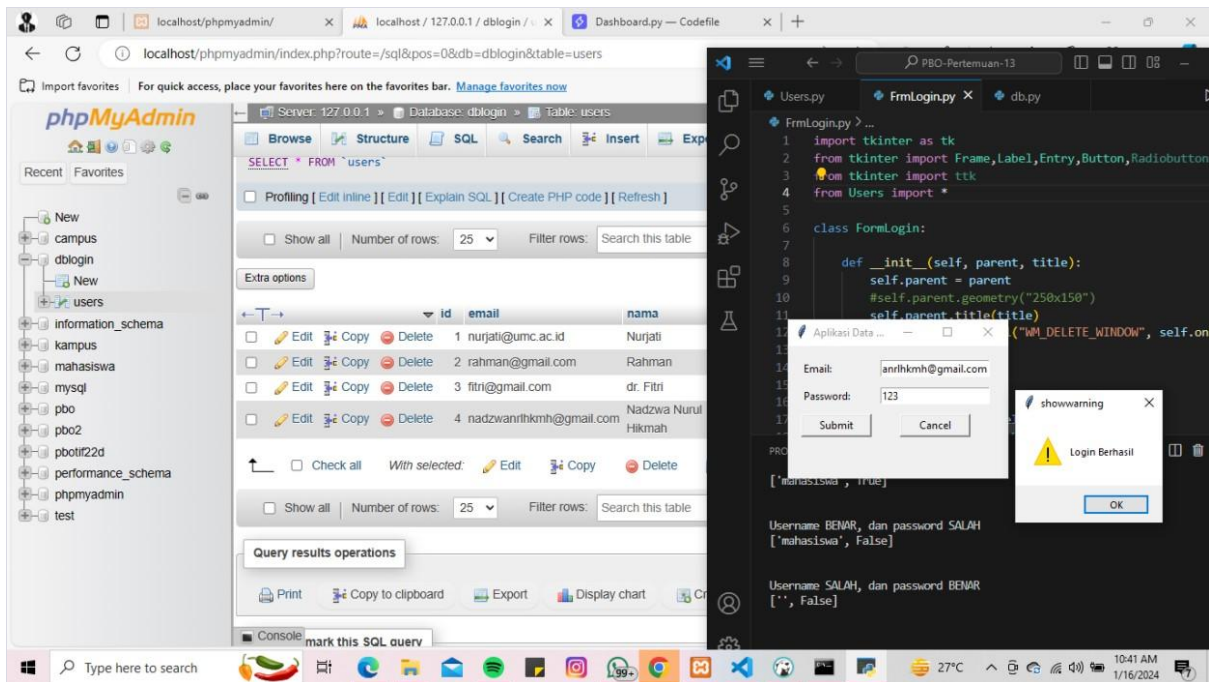
Pertemuan 12: phpMyAdmin

Pertemuan kali ini melanjutkan pembahasan pertemuan sebelumnya.



Pertemuan 13: Aplikasi Python Login Sederhana

membuat Aplikasi Python Login Multi User Berbasis Tkinter MySQL adalah dengan menginstall ((`pip install bcrypt`)), selanjutnya adalah dengan membuat dashboard menggunakan freedb/xampp, sesuaikan host, port, nama database, user, dan password dengan yang ada pada freedb/xampp



****Rencana Pelaksanaan Pembelajaran (RPP)****

****Mata Kuliah : ** Pemrograman Berorientasi Objek**

****Pertemuan Ke- : ** 4**

****Durasi Waktu : ** 2 jam**

****Alokasi Waktu : ** 14.00 - 16.00**

****Pertemuan Tanggal: ** [Tanggal Pertemuan]**

****A. Standar Kompetensi: ****

Peserta didik mampu mengimplementasikan konsep dasar Pemrograman Berorientasi Objek (PBO) dalam bahasa pemrograman Python.

****B. Kompetensi Dasar:****

1. Peserta didik dapat mendefinisikan class dan objek.
2. Peserta didik dapat membuat atribut dan metode pada class.
3. Peserta didik dapat mengimplementasikan konstruktor dan destruktur.

****C. Indikator Pencapaian Kompetensi:****

1. Peserta didik mampu menjelaskan konsep class dan objek.
2. Peserta didik mampu membuat class dengan atribut dan metode.
3. Peserta didik mampu mengimplementasikan konstruktor dan destruktur pada class.

****D. Tujuan Pembelajaran:****

1. Peserta didik memahami konsep dasar PBO.
2. Peserta didik dapat membuat class dan objek dalam bahasa Python.
3. Peserta didik mampu mengimplementasikan konstruktor dan destruktur.

****E. Materi Pembelajaran:****

- Pengenalan konsep dasar PBO.
- Mendefinisikan class dan objek.
- Membuat atribut dan metode pada class.
- Implementasi konstruktor dan destruktur.

****F. Metode Pembelajaran:****

- Ceramah singkat.
- Diskusi kelompok kecil.
- Demonstrasi dan praktek langsung.

****G. Media Pembelajaran:****

- Proyektor dan slide presentasi.
- Papan tulis dan spidol.
- Laptop dan lingkungan pengembangan Python.

****H. Langkah-Langkah Pembelajaran:****

1. Pembukaan (10 menit)

- Sambutan dan pengantar materi.
- Pengenalan konsep dasar PBO.

2. Penjelasan Materi (30 menit)

- Definisi class dan objek.
- Atribut dan metode pada class.

3. Diskusi Kelompok (20 menit)

- Diskusi kelompok kecil tentang contoh penggunaan class dan objek.

4. Demonstrasi dan Praktek (40 menit)

- Demonstrasi pembuatan class dan objek.

- Praktek langsung peserta didik membuat class dan objek.

5. Evaluasi dan Umpan Balik (15 menit)

- Penilaian hasil praktek.
- Pemberian umpan balik.

6. Penutup (5 menit)

- Kesimpulan materi.
- Tugas untuk pertemuan berikutnya.

****I. Penilaian:****

- Partisipasi aktif peserta didik.
- Hasil praktek individu.
- Tugas individu.

****J. Tugas:****

- Membuat program sederhana menggunakan class dan objek.

--