UNESA	**Universitas Negeri Surabaya** **D4 Manajemen Informatika**	**Kode Dokumen**

RENCANA PEMBELAJARAN SEMESTER

MATA KULIAH (MK)	KODE	Rumpun MK	BOBOT (sks)	SEMESTER	Tgl Penyusunan
Prak. Pemrograman Berorientasi Objek	5730102211		**2**		25 Agustus 2026

OTORISASI	Pengembang RPS	Koordinator RMK	Koordinator Program Studi
	Rosita, S.Pd., M.Pd.	I Gde Agung Sri Sidhimantra S.Kom, M.Kom	DODIK ARWIN DERMAWAN

Deskripsi Singkat MK	Mata kuliah ini mengajarkan konsep dan teknik pemrograman berorientasi objek. Dalam mengajarkan konsep PBO tersebut, bahasa pemrograman Python akan digunakan karena Python menggunakan konsep objek dan class dalam pembuatan program. Materi ajar meliputi pengantar konsep OOP, Python Visual Studio Code, objek, class, method, constructor, I/O, pewarisan sifat (inheritance), enkapsulasi (encapsulation), polymorphism (polymorphism), overloading, UML, file, Swing, applet dan exception.

Capaian Pembelajaran (CP)

CPL-PRODI yang dibebankan pada MK

CPL-7	Dapat mengkaji dan memanfaatkan ilmu pengetahuan dan teknologi dalam memberikan solusi serta bertanggung jawab terhadap hasil kerja kelompok maupun mandiri yang bermutu dan terukur dengan sifat dan konteks yang sesuai dengan bidang keahlian terapannya.
CPL-9	Mampu memilih sumberdaya dan memanfaatkan perangkat atau teknologi modern untuk merancang dan mewujudkan bidang rekayasa perangkat lunak sesuai dengan kebutuhan fungsional dan non fungsional secara spesifik dengan pertimbangan yang tepat terhadap masalah kultural, sosial, dan lingkungan dengan mengacu kepada metode dan standar industri.
CPL-11	Mampu meningkatkan kinerja atau mutu suatu proses dalam perangkat lunak melalui pengujian, pengukuran obyek kerja, analisis dan interpretasi data sesuai prosedur dan standar.

Capaian Pembelajaran Mata Kuliah (CPMK)

CPMK-1	Mahasiswa dapat memahami konsep dasar pemrograman berorientasi objek seperti class, objects, abstraction, encapsulation, inheritance dan polymorphism dalam pembuatan program
CPMK-2	Mahasiswa dapat merancang sebuah aplikasi dan melakukan debugging pengujian program dengan menggunakan pemrograman berorientasi objek
CPMK-3	Mahasiswa dapat merancang dan menangani user interface dalam sebuah aplikasi

Matrik CPL - CPMK

CPMK	CPL-7	CPL-9	CPL-11
CPMK-1	✔		
CPMK-2		✔	
CPMK-3			✔

Matrik CPMK pada Kemampuan akhir tiap tahapan belajar (Sub-CPMK)

CPMK	Minggu Ke															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CPMK-1	✔	✔	✔	✔	✔	✔										
CPMK-2							✔	✔				✔	✔	✔	✔	✔
CPMK-3									✔	✔	✔					

Pustaka

Utama :

1. Harold, E.R. 2014. Python Network Programming, 4th edition. O'Reilly
2. "Think Python: How to Think Like a Computer Scientist" Allen B. Downey, 2nd Edition (2015)

Pendukung :

1. Jaworski, J. 1998. Python 2 Unleashed. Sams Publishing
2. Modul Teori dan Student Activity
3. Holmes, B.J., Joice D.T. 2001. Object-Oriented Programming With Python, second edition.
4. Bakker, J. 2005. Beginning Python Objects From Concepts to Code, second edition, Apress.
5. Classes: Dokumentasi resmi mengenai class, object, dan abstraction.
6. Inheritance: Dokumentasi lengkap tentang inheritance, polymorphism, dan konsep terkait.
7. Errors and Exceptions: Dokumentasi resmi untuk debugging dan exception handling.

Mg Ke-	Kemampuan akhir tiap tahapan belajar (Sub-CPMK)	Penilaian		Bentuk Pembelajaran, Metode Pembelajaran, Penugasan Mahasiswa, [Estimasi Waktu]		Materi Pembelajaran	Bobot Penilaian (%)
		Indikator	Kriteria & Bentuk	Luring (*offline*)	*Daring (online)*		
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
1	Pengenalan OOP dan penerapan konsep class dan objek dalam pemrograman python	1. Memahami konsep dasar Pemrograman Berorientasi Objek (OOP) 2. 1.2. Mengenal sintaks dasar Python dalam konteks OOP 3. Mampu membuat kelas dan objek sederhana dalam Python.	Keaktifan mahasiswa dalam bertanya	Kuliah, Tugas Praktikum : Membuat kelas dan objek Laptop, membuat pola dasar dengan turtle menggunakan konsep class dan objek			5%
2	Menerapkan konsep konstruktur dan enkapsulasi dalam pemrograman python	1. Memahami konsep konstruktor dalam OOP 2. Memahami dan menerapkan prinsip enkapsulasi dalam Python 3. Mampu mengimplementasikan akses modifier (public, private) untuk mengatur aksesibilitas atribut dan metode 4. Mahasiswa dapat membuat objek bergerak sederhana di Pygame dengan prinsip OOP	Keaktifan mahasiswa dalam bertanya	Kuliah, Tugas Praktikum: Membuat program Smartphone menggunakan konsep konstruktur dan enkapsulasi			5%
3	Menerapkan konsep metode dan atribut kelas dalam pemrograman python	1. Memahami konsep metode dan atribut kelas (class methods dan class attributes) 2. Membedakan antara metode/atribut instance dan metode/atribut kelas 3. Mengimplementasikan metode dan atribut kelas dalam program Python	Keaktifan mahasiswa dalam bertanya	Kuliah, Tugas Praktikum: Membuat program Laptop menggunakan konsep metode dan atribut			5%

4	Menerapkan konsep Inheritance dalam pemrograman python	1. Memahami konsep pewarisan (inheritance) dalam Pemrograman Berorientasi Objek (OOP) 2. Menerapkan pewarisan dalam pembuatan kelas di Python 3. Memahami peran superclass dan subclass serta cara mengakses anggota dari superclass	Keaktifan mahasiswa dalam bertanya	Kuliah, Tugas Praktikum: Membuat program perangkat elektronik menggunakan konsep inheritance			5%
5	Menerapkan konsep Inheritance dalam pemrograman python	1. Memahami konsep dasar polimorfisme dalam Pemrograman Berorientasi Objek (OOP) 2. Menerapkan polimorfisme dalam metode dan kelas yang berbeda 3. Mengimplementasikan polimorfisme dengan teknik overriding dan penggunaan metode umum pada kelas turunan	Keaktifan mahasiswa dalam bertanya	Kuliah, Tugas Praktikum: Membuat program hewan dengan konsep polimorfisme			5%
6	Menerapkan konsep abstraksi dalam pemrograman python	1. Memahami konsep dasar abstraksi dalam Pemrograman Berorientasi Objek (OOP) 2. Menerapkan abstraksi dalam pembuatan kelas abstrak menggunakan modul abc pada Python 3. Menerapkan metode abstrak dalam kelas turunan	Keaktifan mahasiswa dalam bertanya	Kuliah, Tugas Praktikum: Membuat program kendaraan menggunakan konsep abstraksi			5%
7	Menerapkan exception handling dalam pemrograman python	1. Memahami konsep error dan exception dalam Python 2. Menggunakan blok try-except untuk menangani exception 3. Menangani berbagai jenis exception dan menggunakan blok finally untuk membersihkan sumber daya 4. Membuat exception sendiri (custom exception) dalam Python	Keaktifan mahasiswa dalam bertanya	Kuliah, Tugas Praktikum: Membuat program kalkulator dengan menerapkan exception handling			5%
8	Evaluasi Tengah Semester / Ujian Tengah Semester	Membuat exception sendiri (custom exception) dalam Python	Kemampuan menyelesaikan tugas	Evaluasi Tengah Semester / Ujian Tengah Semester : Quiz			15%
9	Menerapkan konsep OOP dalam pembuatan game sederhana python	1. Memahami konsep dasar OOP (class, object, inheritance, encapsulation) dalam pengembangan game dengan PyGame 2. Mengaplikasikan konsep OOP dalam pembuatan game sederhana menggunakan PyGame 3. Membuat program game berbasis PyGame dengan menggunakan class, objek, dan prinsip-prinsip OOP	Keaktifan mahasiswa dalam bertanya	Kuiah, Tugas Praktikum : Membuat game pygame sederhana dengan menerapkan PBO			5%

10	Menerapkan konsep OOP dalam pembuatan game sederhana python	1. Memahami konsep dasar OOP (class, object, inheritance, encapsulation) dalam pengembangan game dengan PyGame 2. Mengaplikasikan konsep OOP dalam pembuatan game sederhana menggunakan PyGame 3. Membuat program game berbasis PyGame dengan menggunakan class, objek, dan prinsip-prinsip OOP	Keaktifan mahasiswa dalam bertanya	Kuiah, Tugas Praktikum : Membuat game pygame sederhana dengan menerapkan PBO			5%
11	Menerapkan konsep OOP dalam pembuatan game sederhana python	1. Memahami konsep dasar OOP (class, object, inheritance, encapsulation) dalam pengembangan game dengan PyGame 2. Mengaplikasikan konsep OOP dalam pembuatan game sederhana menggunakan PyGame 3. Membuat program game berbasis PyGame dengan menggunakan class, objek, dan prinsip-prinsip OOP	Keaktifan mahasiswa dalam bertanya	Kuiah, Tugas Praktikum : Membuat game pygame sederhana dengan menerapkan PBO			5%
12	Menerapkan konsep OOP dalam pembuatan proyek akhir	1. Mahasiswa mampu menentukan ide proyek berbasis PBO 2. Mahasiswa mampu membuat perancangan awal (blueprint) dari sistem menggunakan UML 3. Mahasiswa mampu mengimplementasikan class dan objek berdasarkan perancangan yang sudah dibuat 4. Mahasiswa mampu menerapkan prinsip OOP dalam kode mereka 5. Mahasiswa mampu menerapkan inheritance dan polimorfisme dalam proyek 6. Mahasiswa mampu menerapkan inheritance dan polimorfisme dalam proyek 7. Mahasiswa mampu menyelesaikan proyek berbasis PBO 8. Mahasiswa mampu mendemonstrasikan hasil proyek kepada kelas	Keaktifan mahasiswa diskusi kelompok	Kuiah, Tugas Praktikum : Membuat projek kelompok dengan menerapkan PBO			5%

13	Menerapkan konsep OOP dalam pembuatan proyek akhir	1. Mahasiswa mampu menentukan ide proyek berbasis PBO 2. Mahasiswa mampu membuat perancangan awal (blueprint) dari sistem menggunakan UML 3. Mahasiswa mampu mengimplementasikan class dan objek berdasarkan perancangan yang sudah dibuat 4. Mahasiswa mampu menerapkan prinsip OOP dalam kode mereka 5. Mahasiswa mampu menerapkan inheritance dan polimorfisme dalam proyek 6. Mahasiswa mampu menerapkan inheritance dan polimorfisme dalam proyek 7. Mahasiswa mampu menyelesaikan proyek berbasis PBO 8. Mahasiswa mampu mendemonstrasikan hasil proyek kepada kelas	Keaktifan mahasiswa diskusi kelompok	Kuiah, Tugas Praktikum : Membuat projek kelompok dengan menerapkan PBO			5%
14	Menerapkan konsep OOP dalam pembuatan proyek akhir	1. Mahasiswa mampu menentukan ide proyek berbasis PBO 2. Mahasiswa mampu membuat perancangan awal (blueprint) dari sistem menggunakan UML 3. Mahasiswa mampu mengimplementasikan class dan objek berdasarkan perancangan yang sudah dibuat 4. Mahasiswa mampu menerapkan prinsip OOP dalam kode mereka 5. Mahasiswa mampu menerapkan inheritance dan polimorfisme dalam proyek 6. Mahasiswa mampu menerapkan inheritance dan polimorfisme dalam proyek 7. Mahasiswa mampu menyelesaikan proyek berbasis PBO 8. Mahasiswa mampu mendemonstrasikan hasil proyek kepada kelas	Keaktifan mahasiswa diskusi kelompok	Kuiah, Tugas Praktikum : Membuat projek kelompok dengan menerapkan PBO			5%

15	Menerapkan konsep OOP dalam pembuatan proyek akhir	1. Mahasiswa mampu menentukan ide proyek berbasis PBO 2. Mahasiswa mampu membuat perancangan awal (blueprint) dari sistem menggunakan UML 3. Mahasiswa mampu mengimplementasikan class dan objek berdasarkan perancangan yang sudah dibuat 4. Mahasiswa mampu menerapkan prinsip OOP dalam kode mereka 5. Mahasiswa mampu menerapkan inheritance dan polimorfisme dalam proyek 6. Mahasiswa mampu menerapkan inheritance dan polimorfisme dalam proyek 7. Mahasiswa mampu menyelesaikan proyek berbasis PBO 8. Mahasiswa mampu mendemonstrasikan hasil proyek kepada kelas	Keaktifan mahasiswa diskusi kelompok	Kuiah, Tugas Praktikum : Membuat projek kelompok dengan menerapkan PBO			5%
16	Menerapkan konsep OOP dalam pembuatan proyek akhir	Mahasiswa mampu mendemonstrasikan hasil proyek kepada kelas	Keaktifan mahasiswa diskusi kelompok	Evaluasi Akhir Semester / Ujian Akhir Semester : Demo Proyek			15%

Catatan

1. **Capaian Pembelajaran Lulusan Program Studi (CPL-Prodi)** adalah rumusan kemampuan yang harus dimiliki oleh lulusan program studi yang mencakup sikap, penguasaan pengetahuan, keterampilan umum, dan keterampilan khusus sesuai dengan jenjang kualifikasi, yang diperoleh melalui proses pembelajaran.
2. **CPL-Prodi yang dibebankan pada Mata Kuliah** adalah bagian dari CPL-Prodi yang dipilih dan relevan untuk dikembangkan melalui suatu mata kuliah, sebagai dasar perumusan capaian pembelajaran mata kuliah.
3. **Capaian Pembelajaran Mata Kuliah (CPMK)** adalah kemampuan yang dirumuskan secara spesifik dan diturunkan dari CPL-Prodi yang dibebankan pada mata kuliah, serta menggambarkan kompetensi yang harus dicapai mahasiswa setelah menyelesaikan mata kuliah.
4. **Sub-CPMK** adalah kemampuan akhir yang direncanakan pada setiap tahap pembelajaran, sebagai penjabaran lebih rinci dari CPMK, bersifat terukur dan dapat diamati sesuai dengan materi pembelajaran.
5. **Indikator Penilaian** adalah pernyataan spesifik dan terukur yang digunakan untuk menilai ketercapaian Sub-CPMK, baik dari proses maupun hasil belajar mahasiswa, yang didukung dengan bukti penilaian.
6. **Kriteria Penilaian** adalah tolok ukur yang digunakan untuk menentukan tingkat ketercapaian hasil belajar mahasiswa berdasarkan indikator penilaian, guna menjamin penilaian yang objektif, konsisten, dan adil. Kriteria penilaian dapat bersifat kuantitatif dan/atau kualitatif.
7. **Bentuk Penilaian** meliputi penilaian tes dan non-tes sesuai dengan karakteristik capaian pembelajaran.
8. **Bentuk Pembelajaran** meliputi kuliah, responsi, tutorial, seminar atau bentuk lain yang setara, praktikum, praktik studio, praktik bengkel, praktik lapangan, penelitian, dan pengabdian kepada masyarakat.
9. **Metode Pembelajaran** merupakan cara penyampaian pembelajaran yang digunakan untuk mencapai capaian pembelajaran, antara lain Small Group Discussion, Project Based Learning, Case Based Learning, Collaborative Learning, Discovery Learning, Self-Directed Learning, serta metode lain yang relevan.
10. **Materi Pembelajaran** adalah bahan kajian yang disusun secara sistematis dalam bentuk pokok dan sub-pokok bahasan untuk mendukung pencapaian Sub-CPMK.
11. **Bobot Penilaian** adalah persentase kontribusi setiap komponen penilaian terhadap nilai akhir mata kuliah, yang ditetapkan secara proporsional sesuai tingkat kompleksitas dan ketercapaian Sub-CPMK, dengan total keseluruhan sebesar 100%.
12. TM = Tatap Muka, PT = Penugasan Terstruktur, BM = Belajar Mandiri.