**Lembar Kerja Penilaian dan Diskusi**

**(LKPD)**

**Mata Pelajaran : Layanan Komputasi Awan XII**

**Topik : Object-Oriented Programming Python**

|  |  |
| --- | --- |
| Nama Siswa | Budi Setiawan |
| No. Presensi | 12 |
| Kelas | XII SIJA 1 |

* **Instruksi Tugas :** Proyek Pemrograman Berorientasi Objek (OOP) dengan Python.
* **Tujuan Pembelajaran :**
* Memodifikasi pemrograman Python untuk mengembangkan program AI sederhana di kehidupan sehari-hari.
* Menganalisis contoh-contoh program AI sederhana yang ada di kehidupan sehari-hari.
* Menyimpulkan fungsi dan penerapan pemrograman AI di kehidupan sehari-hari.
* **Tujuan Spesifik :**

Memungkinkan siswa untuk mendapatkan pemahaman yang mendalam tentang konsep OOP dan bagaimana mengaplikasikannya dalam pemecahan masalah nyata melalui pembuatan proyek Python.

* Deskripsi Tugas:

1. Pengenalan Konsep OOP:

Kita akan memulai proyek yang akan memungkinkan siswa untuk menjalankan konsep OOP dalam bahasa pemrograman Python. Pemahaman dasar tentang OOP adalah kunci untuk berhasil dalam tugas ini.

1. Pengenalan Objek-Oriented Programming
2. Latihan Soal (file dapat diunduh melalui E-learning SMK N 2 Yogyakarta)
3. Berdasarkan latihan soal yang telah diberikan, jawablah pertanyaan berikut dengan tepat.
4. Bagaimana cara mendefinisikan sebuah Class dan objek dalam sebuah permasalahan?
5. Tentukan nama Class dan atribut-atribut yang akan dimiliki oleh Class tersebut.
6. Tentukan metode-metode yang akan dimiliki oleh Class tersebut.
7. Definisikan Class tersebut menggunakan kata kunci class.
8. Buat objek dari Class tersebut menggunakan operator ().
9. Bagaimana cara membuat subclass (subclass) dan superclass (parent class) dalam Python?

Untuk membuat subclass, kita dapat menggunakan kata kunci class. Kemudian, kita dapat menggunakan kata kunci super() untuk mengakses atribut dan metode dari superclass.

Misalnya, kita ingin membuat subclass Mahasiswa dari Class Student. Berikut adalah cara membuat subclass Mahasiswa:

class Student:

def \_\_init\_\_(self, id, name, age, address):

self.id = id

self.name = name

self.age = age

self.address = address

class Mahasiswa(Student):

def \_\_init\_\_(self, id, name, age, address, nim, jurusan):

super().\_\_init\_\_(id, name, age, address)

self.nim = nim

self.jurusan = jurusan

mahasiswa = Mahasiswa(id="123456789", name="John Doe", age=20, address="Jl. Merdeka No. 1", nim="1234567890", jurusan="Teknik Informatika")

mahasiswa.lihat\_data\_siswa()

Pada kode di atas, Student berperang sebagai superclass sedangkan Mahasiswa merupakan subclass dikarenakan class Mahasiswa mewarisi atribut dan metode dari kelas Student.

1. Bagaimana cara mengakses atribut dalam pemrograman OOP Python?

Untuk mengakses atribut dalam pemrograman OOP Python, kita dapat menggunakan operator titik ‘.’. Misalnya, untuk mengakses atribut name dari objek student yang dibuat dari Class Student, kita dapat mengetikkan “student.name” di baris kode python kita.

1. Bagaimana cara memanggil metode dan menampilkan output dari program yang telah kalian buat?

Untuk memanggil metode dalam pemrograman OOP Python, kita dapat menggunakan operator (). Misalnya, untuk memanggil metode lihat\_data\_siswa dari objek student yang dibuat dari Class Student kita dapat menulis “student.lihat\_data\_siswa()”.

1. Pemilihan Topik:

Siswa diminta untuk memilih satu topik atau masalah yang ingin siswa pecahkan melalui proyek ini. Topik yang dipilih harus relevan dengan kehidupan sehari-hari atau minat pribadi siswa.

1. Definisikan Objek dan Class:

Identifikasi objek-objek utama yang akan terlibat dalam pemecahan masalah. Objek-objek ini kemudian akan direpresentasikan sebagai class-class dalam program Python.

1. Definisikan Atribut dan Metode:

Untuk setiap class yang kalian identifikasi, definisikan atribut-atribut yang sesuai dan metode-metode yang diperlukan. Ini akan menjadi "template" untuk objek-objek kalian.

1. Implementasi Class:

Selanjutnya, implementasikan class-class yang sudah kalian definisikan dalam bahasa pemrograman Python. Pastikan atribut dan metode dalam class berfungsi dengan baik.

1. Menggunakan Objek:

Gunakan objek-objek yang sudah kalian buat untuk memecahkan masalah yang kalian pilih. Jelaskan bagaimana OOP membantu dalam memodelkan dan menyelesaikan masalah tersebut.

1. Dokumentasi Kode:

Penting untuk mendokumentasikan kode kalian dengan baik. Ini mencakup penjelasan tentang class, atribut, metode, serta cara penggunaan program.



1. Pengujian dan Debugging:

Lakukan pengujian terhadap program kalian. Pastikan semuanya berjalan dengan baik. Jika ada kesalahan (bugs), perbaiki mereka.

1. Analisis:

Di akhir fase pembelajaran, siswa akan diminta untuk membuat hasil analisis tentang proyek yang telah dibuat. Siswa akan menjelaskan topik, masalah yang dipecahkan, konsep OOP yang digunakan, hasil output/ demonstrasi penggunaan program, dan pelajaran yang diperoleh selama proyek.

Topik : membuat class “StudentList” yang berfungsi untuk menambahkan, menghapus, mengubah, dan menampilkan data siswa.

Konsep OOP: membuat kelas daftar siswa. Beberapa konsep OOP yang digunakan dalam proyek ini meliputi:

* + - 1. Kelas: Dibuat kelas StudentList sebagai kerangka untuk mengelola data siswa yang terdaftar.
      2. Atribut: Atribut students digunakan untuk menyimpan daftar siswa.
      3. Metode: Metode-metode seperti lihat\_data\_siswa, simpan\_data\_siswa, ubah\_data\_siswa, dan hapus\_data\_siswa digunakan untuk mengelola data siswa.
      4. Inisialisasi: Metode \_\_init\_\_ digunakan untuk menginisialisasi objek StudentList.

Hasil output :

1. Program ini memungkinkan pengguna untuk memilih opsi dari menu yang ditampilkan.

2. Pengguna dapat menambahkan data siswa baru ke dalam program, menghapus data siswa, mengubah data yang sudah ada, dan menampilkan seluruh daftar siswa yang tersimpan.

3. Setiap aksi yang diambil oleh pengguna akan memberikan feedback, seperti pesan "Data siswa telah diubah!" atau "Tidak ada data siswa yang ditemukan!"

Pelajaran yang didapat:

1. Pembuatan kelas dan objek

2. Penggunaan atribut dan metode dalam kelas

1. Evaluasi:

Siswa akan dinilai berdasarkan:

* Pemahaman tentang konsep OOP.
* Kualitas implementasi proyek, termasuk class, atribut, dan metode.
* Kemampuan memecahkan masalah melalui proyek.
* Kualitas dokumentasi kode.
* Keterampilan pengujian dan debugging.
* Kemampuan pemahaman dan menganalisis proyek yang telah dibuat.

NB.

* Pastikan untuk mengisi kode di dalam fungsi-fungsi yang diberikan.
* Siswa diperkenankan mengimprovisasi kode, mengganti nama variabel, fungsi, maupun Class, tetapi tidak diperkenankan mengurangi struktur kode.
* Program yang kalian buat tidak boleh sama/identik dengan teman ataupun sumber internet lainnya.
* Kesamaan program yang identik dianggap plagiarisme.
* Kerjakan sesuai instruksi yang ada pada komentar.
* Pastikan kode berjalan tanpa error dan hasilnya sesuai dengan yang diharapkan.
* Buatlah laporan Jobsheet 8 yang menjawab pertanyaan pada poin **1.c** dan **9**.
* Laporan dikumpulkan dalam format .docx atau .pdf.
* Pengumpulan tugas berupa file laporan dan file python .py, kemudian di konversi dalam bentuk archive (.rar/.zip) dengan nama (**Jobsheet8\_No\_Nama\_Kelas.rar/zip**)

Ini adalah kesempatan untuk mendalami pemahaman kalian tentang OOP dan bagaimana mengaplikasikannya dalam pemecahan masalah dunia nyata. Anda akan memiliki waktu yang cukup untuk mengerjakan proyek ini, jadi manfaatkan waktu sebaik mungkin. Jika memerlukan bimbingan tambahan, jangan ragu untuk meminta bantuan dari guru atau instruktur. Selamat mengerjakan proyek!