KISI-KISI ASESMEN SUMATIF

Satuan Pendidikan : SMK Budi Luhur
Nama Guru Mapel : Cindy Putri Az Zahra, S.Kom
Mata Pelajaran : Pemrograman Berorientasi Objek
Kelas : XII RPL

No	Tujuan Pembelajaran	Materi	Indikator Soal	Bentuk Soal
1	Peserta didik dapat menjelaskan konsep dasar OOP (Object, Class, Instance) dan mengimplementasikan fungsi dan objek literal untuk merepresentasikan data.	Konsep Dasar OOP: Object, Class, Instance, Atribut, Behavior. Dasar JavaScript: Fungsi, Objek Literal.	● Menjelaskan definisi dan hubungan antara Object, Class, dan Instance. ● Mengidentifikasi Atribut dan Behavior dari sebuah objek data. ● Mengimplementasikan objek data menggunakan sintaks objek literal atau fungsi di JavaScript.	PG, PG Kompleks, Benar Salah
2	Peserta didik dapat menggunakan constructor function dan sintaks Class (ES6) untuk membuat blueprint objek yang dapat digunakan kembali, serta membuat objek (instance) dari Class yang telah dirancang.	Constructor Function (pre-ES6). Sintaks Class (ES6) - class, constructor. Membuat Instance (new).	● Mengkonversi desain objek literal menjadi Constructor Function atau sintaks Class (ES6). ● Menganalisis fungsi dari kata kunci constructor dalam Class. ● Membuat instance objek baru dengan benar menggunakan kata kunci new.	PG, PG Kompleks, Benar Salah
3	Peserta didik dapat merancang	Inheritance (extends,	● Mengimplementasikan Inheritance menggunakan kata	PG, PG

	hirarki class yang logis dan mengimplementasikan Inheritance (extends, super()). menerapkan Encapsulation melalui properti private (#) dan getter/setter untuk kontrol data.	super()). Encapsulation (properti private \#, Getter dan Setter). Latihan Soal Gabungan.	kunci extends dan super() pada Class turunan. ● Menganalisis kebutuhan penggunaan super() dalam constructor subclass. ● Menerapkan Encapsulation dengan menggunakan properti private (#) dan mendefinisikan Getter/Setter untuk mengontrol akses data.	Kompleks, Benar Salah
4	Peserta didik dapat menerapkan Polymorphism melalui Method Overriding untuk memberikan perlakuan berbeda pada class turunan, dan menggunakan static method untuk fungsionalitas level Class.	Polymorphism (Method Overriding). Static Method dan Static Property.	● Mengimplementasikan Method Overriding untuk memodifikasi perilaku metode di subclass. ● Mengidentifikasi situasi yang tepat untuk menerapkan Method Overriding dalam hirarki Class. ● Menggunakan static method atau static property untuk fungsionalitas yang tidak memerlukan instance objek.	PG, PG Kompleks, Benar Salah
5	Peserta didik dapat memahami penggunaan abstract class, serta perbedaan dan penggunaan composition atau inheritance.	Abstraction (Konsep Abstract Class dan Interface - diimplementasikan secara konseptual di JS). Composition vs Inheritance ("has-a" vs "is-a").	● Membandingkan dan menjelaskan perbedaan konseptual antara Composition ("has-a") dan Inheritance ("is-a"). ● Menganalisis skenario desain Class dan menentukan apakah lebih tepat menggunakan Composition atau Inheritance. ● Menjelaskan peran konseptual dari abstract class dalam desain OOP.	PG, PG Kompleks, Benar Salah
6	Peserta didik mampu menjelaskan logika dan struktur desain OOP yang dibuat.	Review menyeluruh (Object, Class, 4 Pilar OOP). Presentasi Konsep Desain OOP.	● Merangkum dan menghubungkan keempat pilar OOP (Encapsulation, Inheritance, Polymorphism, Abstraction) dalam sebuah studi kasus. ● Menyajikan desain Class dan menjelaskan pemilihan struktur OOP (misalnya, mengapa memilih Inheritance daripada Composition).	PG, PG Kompleks, Benar Salah

7	Peserta didik mampu merancang, mengimplementasikan, dan menguji solusi OOP untuk memodelkan sistem nyata secara mandiri.	Proyek Akhir: Penerapan OOP pada Web/Android/Studi Kasus Nyata.	● Merancang sistem Class yang logis untuk memodelkan sistem nyata. ● Mengimplementasikan minimal tiga dari empat pilar OOP (Encapsulation, Inheritance, Polymorphism) dalam kode proyek. ● Menguji fungsionalitas sistem yang dibangun untuk memastikan instance objek bekerja sesuai dengan desain Class.	PG, PG Kompleks, Benar Salah