LAPORAN PRAKTIK
PEMROGRAMAN BERORENTASI OBJEK

UJIAN AKHIR SEMESTER

Disusun oleh :

Rifqy Rivaldi

V3922040

Dosen :

Akhmad Syarif S.Kom M.Kom

PS D-III TEKNIK INFORMATIKA

SEKOLAH VOKASI

UNIVERSITAS SEBELAS MARET

2023

SOAL

1. **kapan memanfaatkan metode pemrograman berorientasi object?**
   2. apa manfaat dari penggunaan metode pemrograman berorientasi object?
   3. buat 1 project/program (bebas tema program) dengan menerapkan metode pemrograman berorientasi object java, dan berikan keterangan/penjelasan disetiap coding
   keterangan : semakin banyak metode object orientasi di implementasikan akan semakin tinggi nilai yg diberikan.

JAWABAN

1) beberapa situasi di mana memanfaatkan OOP dapat bermanfaat:

1. Mengelola kompleksitas: OOP memungkinkan Anda membagi program besar menjadi bagian-bagian kecil yang lebih mudah diatur. Anda dapat membuat objek yang mewakili entitas dunia nyata atau konsep-konsep tertentu, dan setiap objek tersebut memiliki tugas dan tanggung jawabnya sendiri. Ini membantu mengelola kompleksitas dan membuat kode lebih mudah dipahami.

2. Reusabilitas kode: Dengan menggunakan OOP, Anda dapat membuat kelas-kelas yang dapat digunakan kembali. Anda dapat membuat kelas umum yang memiliki atribut dan metode yang relevan untuk banyak situasi, dan kemudian menggunakan kelas tersebut di berbagai bagian program Anda. Dengan demikian, Anda dapat menghindari penulisan kode yang berulang-ulang dan mempercepat pengembangan aplikasi.

3. Pemeliharaan dan perbaikan: OOP memisahkan data dan perilaku ke dalam objek. Jika ada masalah dalam program Anda, Anda dapat fokus pada objek atau kelas tertentu tanpa harus memahami seluruh program. Ini mempermudah pemeliharaan dan perbaikan kode.

4. Kolaborasi tim: OOP memfasilitasi kolaborasi dalam tim pengembangan perangkat lunak. Tim dapat bekerja pada kelas-kelas yang berbeda secara mandiri tanpa mengganggu bagian-bagian kode yang lain. Hal ini juga memungkinkan untuk membagi pekerjaan antara anggota tim berdasarkan spesialisasi dan kemampuan mereka.

5. Polimorfisme dan warisan: Konsep seperti polimorfisme dan warisan dalam OOP memungkinkan Anda untuk menulis kode yang lebih fleksibel dan mudah diubah. Dengan polimorfisme, Anda dapat menggunakan objek dengan tipe yang berbeda secara bersamaan, sementara dengan warisan, Anda dapat mewarisi sifat dan perilaku dari kelas yang ada dan memperluasnya.

6. Pengembangan skala besar: OOP adalah pendekatan yang baik untuk pengembangan aplikasi yang berskala besar. Dengan membagi program menjadi objek-objek yang independen, tim pengembang dapat bekerja secara paralel pada bagian-bagian yang berbeda dari program tersebut. Ini memfasilitasi pengembangan skala besar yang efisien.

Penting untuk dicatat bahwa OOP bukanlah satu-satunya metode pemrograman yang ada, dan pilihan penggunaannya tergantung pada konteks dan kebutuhan proyek Anda

2) Pemrograman berorientasi objek (OOP) memiliki beberapa manfaat, di antaranya:

1. Modularitas: OOP memungkinkan programmer untuk membagi program menjadi bagian-bagian yang lebih kecil dan terorganisir dengan baik, yang disebut objek. Hal ini memudahkan dalam pengembangan dan pemeliharaan program.
2. Peningkatan efisiensi: OOP memungkinkan programmer untuk menggunakan kembali kode yang sudah ada, sehingga menghemat waktu dan sumber daya.
3. Peningkatan keamanan: OOP memungkinkan programmer untuk menyembunyikan detail implementasi dari objek, sehingga mencegah akses yang tidak sah ke data dan fungsi.
4. Peningkatan fleksibilitas: OOP memungkinkan programmer untuk dengan mudah menambahkan atau mengubah fitur program, karena objek dapat dengan mudah dimodifikasi.
5. Peningkatan pemeliharaan: Kode berorientasi objek lebih mudah dipelihara karena mengikuti konvensi pengkodean yang agak ketat dan ditulis dalam format penjelasan sendiri. Sebagai contoh, ketika seorang developer memperluasnya, merefaktorinya, atau mendebugnya, mereka dapat dengan mudah mengetahui struktur pengkodean bagian dalam dan mempertahankan kode dari waktu ke waktu.
6. Abstraksi dan kompleksitas tersembunyi: OOP memungkinkan penggunaan konsep abstraksi untuk menyembunyikan kompleksitas yang tidak relevan. Anda dapat membuat objek-objek dengan antarmuka yang sederhana dan menyembunyikan detail implementasinya. Ini memudahkan penggunaan dan pemahaman objek tersebut, serta mengurangi kompleksitas bagi pengguna lain.