TUGAS PERTEMUAN 1

Nama : Dara Cantika Dewi

NIM : 121450127

Mata Kuliah : Pemrograman Berbasis Fungsi RB

1. **Kelebihan dan kekurangan Paradigma *Procedural* dibandingkan dengan *Object Oriented***

Kelebihan	Kekurangan
Paradigma procedural lebih mudah dipahami dan diterapkan dibandingkan dengan object oriented, karena lebih menekankan pada proses dan alur dalam menyelesaikan suatu masalah.	Paradigma procedural memiliki keterbatasan dalam membangun abstraksi yang sesuai dengan model sistem yang diterapkan, sehingga sulit untuk mengelompokkan data dan perilaku yang berhubungan dan memanipulasinya secara efisien.
Paradigma procedural memiliki kemampuan menyelesaikan suatu masalah dengan lebih cepat dan efisien dibandingkan dengan object oriented, karena tidak membutuhkan penyimpanan data dan method yang kompleks.	Paradigma procedural memiliki pendekatan yang lebih linear dan terpisah dari konsep OOP sehingga sulit untuk mengatasi permasalahan yang kompleks dan menjaga konsistensi data.
Dalam paradigma procedural, proses debugging lebih mudah dilakukan dibandingkan dengan object oriented, karena lebih memfokuskan pada alur dan proses program.	Dalam paradigma procedural, modul kode menjadi lebih rumit dan tidak terstruktur karena harus memperhatikan banyak hal seperti variabel global, passing argument, dan sebagainya.
Paradigma procedural memiliki kemampuan yang lebih baik dalam hal portabilitas dibandingkan dengan object oriented, karena lebih menekankan pada logika dan proses, sehingga lebih mudah dipindahkan ke sistem yang berbeda.	Pemeliharaan dan perbaikan kode dalam paradigma procedural lebih sulit dan tidak efisien karena sulit untuk memahami alur dari program dan bagaimana data diproses.
Dalam paradigma procedural, dokumentasi program lebih mudah dilakukan dibandingkan dengan object oriented, karena lebih menekankan pada proses dan alur, sehingga memudahkan dalam pemahaman dan pemeliharaan program.	Konflik nama dapat terjadi dalam paradigma procedural karena tidak ada konsep pembatasan hak akses seperti yang ada dalam OOP. Variabel global dapat dikonflikkan dengan variabel lokal dan sulit untuk memahami konteks dari variabel tersebut.

2. **Kelebihan dan kekurangan Paradigma *Object Oriented* dibandingkan dengan *Functional***

Kelebihan	Kekurangan
Object Oriented menyediakan mekanisme untuk menyembunyikan implementasi dalam object, sehingga hanya menunjukkan interface kepada user.	Pemrograman Object Oriented biasanya lebih kompleks dibandingkan dengan functional programming, karena memerlukan penggunaan class, inheritance, dan abstraction.
Kode dapat dibuat sebagai object dan dapat digunakan kembali pada aplikasi lain dengan sedikit atau tanpa modifikasi.	Karena adanya penggunaan class dan inheritance, pemeliharaan dan pembaruan kode pada paradigma Object Oriented lebih sulit dibandingkan dengan functional programming.
Object Oriented memungkinkan pengguna untuk membuat abstraksi dari realitas dunia nyata yang membuatnya lebih mudah untuk dipahami dan diterapkan.	Debugging pada paradigma Object Oriented lebih sulit dan memerlukan waktu yang lebih lama dibandingkan dengan debugging pada functional programming.
Kelebihan dari inheritance adalah memungkinkan untuk memodifikasi dan mengextend class tanpa memodifikasi kode sumber yang sudah ada.	Pada paradigma Object Oriented, pembuatan unit test lebih sulit karena memerlukan interaksi antar class dan objek.
Object Oriented memungkinkan sebuah object untuk memiliki beberapa bentuk, sehingga lebih mudah untuk melakukan generalisasi dan spesialisasi.	Meskipun Object Oriented merupakan paradigma yang umum digunakan, kurva belajar untuk memahami dan mempraktikkan Object Oriented lebih tinggi dibandingkan dengan functional programming.

3. **Kelebihan dan kekurangan Paradigma *Procedural* dibandingkan dengan *Functional***

Kelebihan	Kekurangan
Paradigma procedural memiliki sintaks yang lebih sederhana dan mudah dipahami oleh pemrogram pemula.	Dalam paradigma procedural, kode ditulis dalam bentuk step-by-step, sehingga sulit untuk memisahkan bagian-bagian yang berbeda dan menggunakannya lagi pada bagian lain dalam program.
Paradigma procedural memungkinkan pemecahan masalah dengan cara yang lebih terstruktur dan logis, membuat proses pengembangan lebih mudah.	Dalam paradigma procedural, sulit untuk menggunakan ulang kode yang sudah dibuat sebelumnya, karena tidak ada mekanisme untuk mengubah data atau memberikan input yang berbeda pada kode tersebut.
Paradigma procedural memiliki kemampuan untuk memecahkan berbagai jenis masalah dan dapat diterapkan pada berbagai lingkungan dan platform.	Dalam paradigma procedural, sulit untuk membuat program yang skalabel karena tidak ada cara untuk membuat kode yang terbagi menjadi bagian-bagian yang lebih kecil dan mudah dikelola.
Dokumentasi kode dalam paradigma procedural seringkali lebih baik dan lebih jelas dibandingkan dengan paradigma lain, membuat proses pemeliharaan dan perbaikan kode lebih mudah.	Dalam paradigma procedural, sulit untuk membuat program yang fleksibel dan dapat diubah-ubah, karena tidak ada cara untuk membuat kode yang dapat beradaptasi dengan perubahan data dan input yang berbeda.
Paradigma procedural memiliki kecepatan eksekusi yang lebih tinggi dibandingkan dengan paradigma lain, karena fokus pada prosedur dan tugas spesifik.	Dalam paradigma procedural, fokus lebih pada tahapan-tahapan yang dilakukan pada data daripada pada data itu sendiri.

4. **Pure Functional Programming Language**
   Pure functional programming language adalah suatu paradigma pemrograman yang hanya menggunakan fungsi-fungsi matematis sebagai dasar untuk memecahkan masalah dan membuat program. Dalam paradigma ini, tidak ada perubahan data yang dilakukan secara permanen, sehingga setiap operasi yang dilakukan hanya menghasilkan output baru tanpa memodifikasi input. Ini membuat program lebih mudah diprediksi dan dites, serta mengurangi potensi bug dan masalah lainnya. Contohnya adalah Haskell, Lisp, dan Scheme.

5. Lisp, Haskell, dan Standard ML termasuk dalam bahasa pemrograman pure functional programming.

6. Program untuk nomor 6 dapat dilihat pada github.