

Nama : Anita Rahma Pramoda Cahyani

NIM : 121450154

Mata Kuliah : Pemrograman Berbasis Fungsi

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

Kelebihan	Kekurangan
Paradigma <i>procedural</i> memiliki suatu kemampuan yang dapat menyelesaikan suatu masalah dengan lebih cepat dan efisien dibandingkan dengan <i>object oriented</i> karena tidak memerlukan penyimpanan data dan method yang lebih kompleks.	Paradigma <i>procedural</i> mempunyai pendekatan yang linear dan terpisah dari konsep OOP yang mengakibatkan sulit untuk mengatasi permasalahan yang kompleks dan menjaga konsistensi data.
Paradigma <i>procedural</i> memiliki kemampuan yang lebih baik didalam hal portabilitas dibandingkan dengan <i>object oriented</i> , mengapa demikian karena lebih menekankan pada proses dan logika. Dengan demikian lebih mudah dipindahkan ke dalam sistem yang berbeda.	Paradigma <i>procedural</i> memiliki suatu keterbatasan di dalam membangun abstraksi yang sesuai dengan model sistem yang diterapkan, dengan demikian sulit untuk mengelompokkan data serta perilaku yang berhubungan dan memanipulasinya secara efisien.
Di dalam Paradigma <i>procedural</i> dokumentasi program lebih mudah dilakukan karena lebih menekankan pada alur serta proses dengan demikian memudahkan dalam pemahaman dan pemeliharaan program.	Pemeliharaan dan perbaikan dari kode dalam paradigma <i>procedural</i> lebih sulit serta tidak efisien karena sulit memahami alur dari program dan bagaimana caranya data diproses.

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

Kelebihan	Kekurangan
<i>Object oriented</i> memungkinkan para pengguna untuk membuat abstraksi dari realitas dunia nyata yang mana membuatnya lebih mudah dipahami dan diterapkan kepada para pengguna.	Paradigma <i>object oriented</i> pembuatan unit test lebih sulit karena membutuhkan interaksi antar <i>class</i> dan <i>object</i> .
<i>Object oriented</i> menyediakan mekanisme untuk menyembunyikan implementasi dalam <i>object</i> dengan demikian dapat menunjukkan interface kepada pengguna.	Pemrograman <i>object oriented</i> lebih kompleks dibandingkan dengan <i>functional programming</i> karena menggunakan <i>class</i> , <i>inheritance</i> , dan <i>abstraction</i> .
Kode dapat dibuat sebagai <i>object</i> dapat digunakan kembali pada aplikasi lain dengan sedikit atau tanpa modifikasi.	Karena penggunaan <i>class</i> dan <i>inheritance</i> , pemeliharaan serta pembaruan kode pada paradigma <i>object oriented</i> lebih sulit jika dibandingkan dengan <i>functional programming</i> .

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

Kelebihan	Kekurangan
Paradigma <i>procedural</i> memecahkan suatu permasalahan dengan cara yang lebih terstruktur dan logis. Dan membuat proses pengembangan lebih mudah.	Di dalam paradigma <i>procedural</i> sulit membuat program yang skalabel dikarenakan tidak ada cara membuat kode yang terbagi menjadi bagian-bagian yang kecil dan mudah dikelola.
Paradigma <i>procedural</i> memiliki sintaks yang lebih sederhana serta mudah dipahami oleh pemogram pemula.	Paradigma <i>procedural</i> lebih berfokus pada tahapan yang dilakukan pada data daripada data itu sendiri.
Paradigma <i>procedural</i> memiliki kecepatan eksekusi yang lebih tinggi dibandingkan dengan paradigkma yang lain. Karena lebih fokus pada prosedur dan tugas spesifik.	Paradigma <i>procedural</i> sulit digunakan untuk membuat program yang fleksibel serta dapat diubah-ubah. Karena tidak ada cara untuk membuat kode yang dapat beradaptasi dengan perubahan data serta input yang berbeda.

4. Apa yang dimaksud dengan *Pure Funtional Programming Language*

Pure Funtional Programming Language ialah gaya membangun struktur serta elemen program komputer yang memerlukan semua perhitungan sebagai evaluasi matematika. *Pure Funtional Programming Language* juga memastikan bahwa fungsi, didalam paradigma fungsional apakah hanya bergantung pada argumen mereka, terlepas dari keadaan global atau lokal. Di dalam paradigma ini tidak ada perubahan data yang dilakukan secara permanen maka dari itu setiap operasi yang dilakukan hanya akan menghasilkan output baru tanpa dimodifikasi input.

5. Dari bahasa pemrograman FP di slide sebelumnya (slide 14), manakah yang termasuk *pure functional programming language*

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

6. Program untuk nomor 6 dapat dilihat pada github.