

QUIZ 1

PRINSIP BAHASA PEMROGRAMAN

1. Jelaskan apa yang dimaksud dengan paradigma pemrograman ? mengapa bahasa pemrograman sebaiknya dapat men-support multi paradigm ?
2. Jelaskan apa yang dimaksud dengan “Paradigm’s Kernel Language” ?
3. Jelaskan apa yang dimaksud dengan *creative extension principle* ?
4. Elemen apa saja yang menjadi bagian penting pada *Programming Concept* ?
5. Apa yang dimaksud deterministik dan non deterministik pada *concurrent programming*, berikan contohnya?
6. Dalam taksonomi paradigma pemrograman apakah yang membedakan antara *First-order functional programming* dengan *functional programming*?
7. Dalam taksonomi, paradigma di bedakan kedalam *observable nondeterminism* dan *non observable nondeterminism*. *Observable nondeterminism* yang dimaksud dalam taksonomi ini adalah apa?
8. Jelaskan yang dimaksud dengan *Data Abstraction* ?
9. Sebutkan apa saja keunggulan dalam menerapkan *Data Abstraction* !
10. Konsep *abstraction* pada paradigma berorientasi *object* diimplementasikan dalam ?

Jawab

1. Paradigma pemrograman adalah cara untuk mengklasifikasikan bahasa pemrograman berdasarkan fitur. Bahasa pemrograman sebaiknya support multi paradigm agar bisa menyediakan kerangka kerja dimana para programmer dapat bekerja dalam berbagai gaya, secara bebas mencampurkan konstruksi dari paradigma yang berbeda, dan sebagai solusi bahwa pada beberapa tahap satu paradigma tidak mampu menyelesaikan semua masalah dengan cara termudah atau paling efisien. Paradigma program berorientasi objek dapat digunakan untuk masalah dengan jumlah besar (java,c++), paradigma fungsional diasumsikan bahwa ada fungsi fungsi dasar yang dapat digunakan sehingga penyelesaian masalah berdasarkan fungsi tersebut(Logo,APL), paradigma multi programming mendukung lebih dari satu paradigma pemrograman (Wolfram).
2. Setiap paradigma di definisikan dari beberapa konsep pemrograman, yang diorganisasi menjadi Bahasa yang simple, banyak sekali Bahasa pemrograman tetapi hanya sedikit paradigma.
3. Sebuah konsep tidak digabungkan secara sembarangan untuk membentuk suatu paradigma, yang mana diatur oleh *creative extension principle*. Prinsip ini didefinisikan oleh Fellesien. Yang mana memberi petunjuk bahwa program menjadi rumit karena alasan teknis yang tidak memiliki hubungan langsung dengan masalah yang harus di pecahkan.
4. Programming Concept terdiri dari 4 yaitu :
 - i. Record : record adalah struktur data yang merepresentasikan suatu data dengan indeks akses ke setiap item
 - ii. Lexically scoped closure : konsep yang kuat yang berada di pusat pemrograman. Pemrograman fungsional, yaitu pemrograman dengan closure, merupakan paradigma sentral, dari implementasi closure menggabungkan prosedur dengan referensi eksternal, dari programmer

closure adalah sebuah program yang dapat mengubah instruksi apapun menjadi closure di satu titik pemrograman, dan meneruskan ke titik yang mengeksekusinya. Hasilnya sama seperti instruksi dijalankan pada saat penutupan di buat.

- a) Instan, generik, biasanya pemrograman berorientasi objek
 - b) Pemisahan masalah, biasanya pemrograman berorientasi aspek
 - c) Pemrograman berbasis komponen adalah gaya pemrograman dimana program diatur sebagai komponen
- iii. Independence (concurrency) : membangun program sebagai bagian independent, tetapi tidak sesederhana kelihatannya, untuk mengimplementasikan independent kita membutuhkan konsep pemrograman baru yang disebut concurrency. Ketika dua bagian tidak berinteraksi sama sekali, dapat dikatakan keduanya bersamaan. Terdapat 3 level concurrency :
- a) Sistem terdistribusi : sekumpulan komputer yang terhubung melalui jaringan.
 - b) Operating system : software yang manage komputer.
 - c) Aktivitas di dalam suatu proses, aktifitas serentak disebut thread, thread dijalankan independent tetapi berbagi ruangan memori yang sama.
- iv. Named State : State memperkenalkan gagasan abstrak tentang waktu di dalam program, di dalam program fungsional, tidak ada pengertian tentang waktu, Ketika fungsi panggil dengan argument yang sama maka hasilnya akan selalu sama, rule yang baik adalah tidak pernah memperlihatkan named state.
5. Salah satu masalah dari pemrograman konkruen adalah nondeterministik, nondeterministik terjadi jika pada satu titik tertentu selama eksekusi ada pilihan apa yang harus dilakukan selanjutnya, dan nondeterministik terjadi secara alami Ketika ada konkurensi, contoh nondeterministik terjadi jika :
- $C = \{\text{newCell } 0\}$
thread c=1
thread c=2
maka dari itu hindari nondeterministik dalam bahasa yang berbarengan

Deterministik : semua terjadi secara ketat, tidak ada inputan lebih lanjut selama runtime, tidak ada pilihan selama eksekusi.

- 6. Perbedaan antara first order functional programming dengan imperative programming adalah Teknik mengimplementasikan lexically scoped name serta rekaman yang menyimpan fungsi yang bersamaan dengan environment.
- 7. *Observable nondeterminism* dalam taksonomi adalah tanda bahwa paradigma tersebut adalah nondeterminism yang dapat diamati ataukah tidak, dengan demikian *Observable nondeterminism* di dukung hanya jika diperlukan.
- 8. Data abstraction adalah cara untuk mengatur penggunaan struktur data sesuai dengan aturan yang tepat yang menjamin bahwa struktur data digunakan dengan benar.
- 9. Keuntungan data abstraksi ada 3 yaitu :
 - i. Ada jaminan bahwa abstraksi data akan selalu berfungsi dengan baik

Nama : Muhammad Hargi Muttaqin

Kelas : 2A-D4

NIM : 191524027

- ii. Program lebih mudah dipahami karena tidak perlu memahami bagaimana abstraksi diimplementasikan
 - iii. Memungkinkan membangun program yang sangat besar.
10. Implementasikan pada OOP dimana paradigma pemrograman berdasarkan konsep objek yang berisi data dalam bentuk field, serta code dalam bentuk prosedur atau method. Pada konsep abstraction ini yang berorientasi objek adalah polymorphism dan inheritance.