

EXERCISE 1.13

Tugas Mata Kuliah SK5003 Pemrograman dalam Sains
Chapter I

Mohammad Rizka Fadhli (Ikang)
20921004@mahasiswa.itb.ac.id

03 September 2021

Contents

SOAL 1.1	4
Jawab	4
SOAL 1.2	4
Jawab	4
SOAL 1.3	5
Jawab	5
SOAL 1.4	6
Jawab	6
SOAL 1.5	6
Jawab	6
SOAL 1.6	7
Jawab	7
SOAL 1.7	7
Jawab	7
SOAL 1.8	8
Jawab	8
SOAL 1.9	9
Jawab	9
SOAL 1.10	10
Jawab	10

List of Figures

1	Flow Melakukan Research	4
2	Macam-macam Compiler	8

SOAL 1.1

Explain the differences between a computational model and a mathematical model.

Jawab

Model adalah representasi dari sistem atau masalah. Bisa jadi model hanya berisi bagian tertentu saja dari sistem atau masalah. Suatu model bisa juga lebih sederhana dari masalah sebenarnya asalkan masih relevan dalam beberapa aspek.

Mathematical model adalah formulasi matematis dari suatu permasalahan *real* sedangkan *computational model* adalah implementasi model matematika yang bertujuan untuk menyelesaikan permasalahan secara komputasi di komputer.

Mathematical model dituliskan dalam ekspresi matematik sedangkan *computational model* dituliskan dalam bentuk algoritma.

SOAL 1.2

Explain the reason why the concept of abstraction is important in developing computational models.

Jawab

Abstraction adalah salah satu pilar utama dalam *computational thinking*. Sebelum kita memecahkan suatu masalah secara komputasi, kita perlu mengenal terlebih dahulu permasalahan yang kita temui. Sama halnya dengan melakukan suatu penelitian.

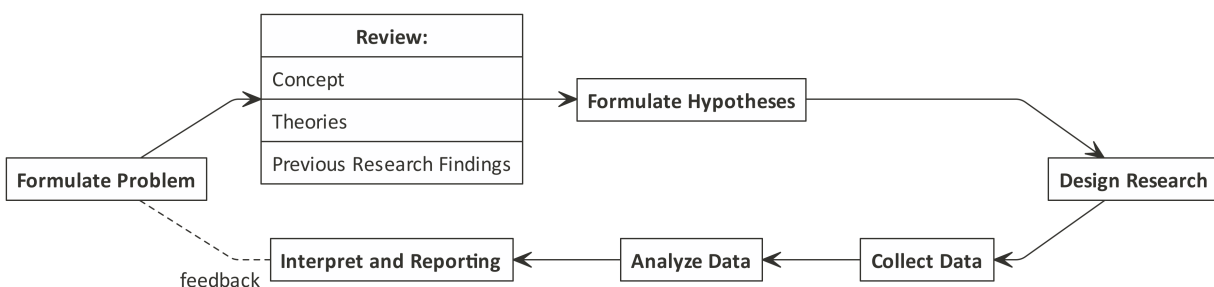


Figure 1: Flow Melakukan Research

Salah satu tahap kritis yang paling penting adalah **formulasi masalah**. *Abstraction* yang baik akan membantu kita memilih elemen esensial dari permasalahan dan memilah mana yang *less importance*.

SOAL 1.3

Investigate and write a short report on the programming languages used to implement computational models.

Jawab

Saya akan membahas secara singkat dua *programming languages* yang familiar bagi saya untuk menjalankan *computational models*, yakni Python dan **R**. Keduanya merupakan program *open source* yang bisa diunduh dan dipakai oleh masyarakat luas. Keduanya juga merupakan *high level programming languages*.

Pada dasarnya Python dan **R** merupakan program *command line interface (CLI)* yang bisa dijalankan di *multiple operating system* termasuk di *cloud* seperti Google Colab¹. Namun ada banyak sekali *integrated development environment (IDE)* yang bisa digunakan.

Baik Python dan **R** memiliki *packages* atau *libraries* yang dikembangkan oleh banyak pihak. *Packages* tersebut berisi himpunan *function* dan perintah yang memudahkan komputasi.

Lantas apa saja perbedaannya?

R is a statistical tool used by academics, engineers and scientists without any programming skills. Python is a production-ready language used in a wide range of industry, research and engineering workflows.²

R applications are ideal for visualizing your data in beautiful graphics. In contrast, Python applications are easier to integrate in an engineering environment.

R programming is better suited for statistical learning, with unmatched libraries for data exploration and experimentation. Python is a better choice for machine learning and large-scale applications, especially for data analysis within web applications.

Python³

Python disebut-sebut sebagai *programming language* terpopuler dunia saat ini. Python diciptakan oleh Guido van Rossum⁴, seorang *programmer* asal Belanda. Bahasa Python dinilai memiliki kesamaan dengan *Perl*, **C**, dan *Java*⁵.

Salah satu ciri khusus pada *script* Python adalah penggunaan *indentation*. Python tidak mengenal penggunaan *bracket { }* sehingga diganti dengan *indentation*.

¹Menggunakan **R** di Google Colab. <https://ikanx101.com/blog/google-colab/>

²<https://www.ibm.com/cloud/blog/python-vs-r>

³<https://www.python.org/>

⁴https://en.wikipedia.org/wiki/Guido_van_Rossum

⁵https://en.wikipedia.org/wiki/Python_syntax_and_semantics

R⁶

R dibuat oleh dua orang *statistician* Ross Ihaka⁷ and Robert Gentleman⁸. Oleh karena itu program ini dinamakan **R**, yakni inisial dari nama penemunya.

Salah satu keunggulan **R** dibandingkan program lainnya adalah keberadaan *portal* **The R Journal**⁹, yakni sebuah portal jurnal *open access* dari berbagai macam *projects* terkait *statistical computing*.

SOAL 1.4

What is a programming language? Why are they needed?

Jawab

Otak manusia memiliki keterbatasan dalam hal melakukan komputasi yang kompleks dan repetitif. Oleh karena itu mereka menciptakan alat bantu sehingga mereka bisa menyelesaikan masalah-masalah yang mereka hadapi. Salah satu alat bantu tersebut adalah komputer. Agar komputer bisa mengerjakan apa yang manusia perintahkan, manusia perlu membuat *programming language* agar manusia dan mesin bisa berinteraksi.

Manusia memberikan kumpulan perintah untuk komputer lakukan komputasi.

SOAL 1.5

Explain why there are many programming languages.

Jawab

Masing-masing *programming language* memiliki kelebihan dan kelemahan masing-masing. Hampir tidak ada satu *programming language* yang **paling cocok** dengan semua permasalahan komputasi dan tujuan penggunaannya. Oleh karena itu ada banyak sekali *programming language* sehingga pengguna bisa memilih mana yang paling sesuai dengan permasalahan, tujuan, dan preferensinya masing-masing.

⁶<https://cran.r-project.org/>

⁷https://en.wikipedia.org/wiki/Ross_Ihaka

⁸[https://en.wikipedia.org/wiki/Robert_Gentleman_\(statistician\)](https://en.wikipedia.org/wiki/Robert_Gentleman_(statistician))

⁹<https://journal.r-project.org/>

SOAL 1.6

What are the differences between compilation and interpretation in high-level programming languages?

Jawab

Compilation berarti proses mengubah bahasa pemrograman menjadi bahasa mesin sehingga bisa dijalankan oleh komputer.

Mengubah *high-level language* ke dalam *low-level language*.¹⁰

Salah satu contohnya adalah bahasa pemrograman Pascal yang memungkinkan kita meng-*compile* algoritma menjadi *executable file* (.exe).

Sedangkan *interpretation* dalam suatu *programming language* hanya berfungsi untuk mengecek perintah dan melaksanakan perintah yang dimasukkan saat itu saja (saat *software*-nya dipakai).

SOAL 1.7

Explain the purpose of compilation. How many compilers are necessary for a given application? What is the difference between program compilation and program execution? Explain.

Jawab

Tujuan utama dari *compilation* adalah agar algoritma yang ada sudah dijadikan bahasa mesin dan dalam *executable files*. Sehingga kita tidak lagi memerlukan (meng-*install*) bahasa pemrograman tersebut ke dalam komputer atau mesin.

User bisa menjalankan langsung algoritma yang ada tanpa harus membuka *interpreter*.

Satu *programming language* bisa memiliki beberapa *compilers* tergantung kebutuhan.

¹⁰<https://codeforwin.org/2017/05/compiler-and-its-need.html>

Some popular compilers

Language	Compilers	Developed by
BASIC	FreeBASIC	FreeBASIC development team
	QuickBASIC	Microsoft
	Visual Basic	Microsoft
C	GCC C	GNU Project
	Borland Turbo C	Embarcadero
	Quick C	Microsoft
C++	GCC	GNU Project
	Borland C++	Borland
	Visual C++	Microsoft
C#	Visual C#	Microsoft
	Mono	Xamarin
Java	javac	Sun Microsystems (Owned by Oracle)
	gcj	GNU Project

Figure 2: Macam-macam Compiler

Sumber gambar: <https://codeforwin.org/2017/05/compiler-and-its-need.html>

Program execution dilakukan setelah algoritma dalam suatu *programming language* ter-*compile* dan menjadi suatu program. *Program execution* sudah termasuk ke dalam proses interpretasi tanpa membutuhkan program *compiler*.

SOAL 1.8

What is the real purpose of developing a program? Can we just use a spreadsheet program to solve numerical problems? Explain.

Jawab

Tujuan pemrograman adalah membuat sekumpulan perintah di komputer sehingga kita bisa menyelesaikan masalah secara otomatis. Pada kasus *numerical problems*, tentu kita bisa menggunakan *spreadsheets* untuk menyelesaikan permasalahan yang bersifat sederhana. Namun saat masalah yang dihadapi kompleks, tentunya *spreadsheets* belum cukup untuk bisa dipakai.

Selain itu, *spreadsheets* tidak bisa bekerja iteratif secara otomatis. Sehingga masih diperlukan tenaga manusia untuk melakukan komputasi.

Sedangkan dengan membuat algoritma atau *computer codes*, kita bisa menyuruh komputer melakukan itersi secara otomatis. Selain itu ada keuntungan lainnya jika kita melakukan *coding*, yakni *customize-able* dan *reproducible*.

SOAL 1.9

Explain the differences between data definitions and instructions in a program written in a high-level programming language.

Jawab

High-level programming language memungkinkan kita memakai bahasa yang lebih mudah untuk berinteraksi dengan komputer. Demikian pula halnya saat kita melakukan:

1. Pendefinisian sesuatu data atau *object*.
 - Pendefinisian biasa dilakukan dengan notasi `=`.
2. Memberikan instruksi atau perintah.
 - Instruksi biasanya menggunakan perintah dalam bahasa Inggris, seperti `for`, `while`, `print`, atau dengan notasi aritmatika seperti `+` `-` `*` `/`.

SOAL 1.10

For developing small programs, is it still necessary to use a software development process? Explain. What are the main advantages in using a process for program development? What are the disadvantages?

Jawab

Walaupun program yang sedang dibuat merupakan *small programs*, tidak ada salahnya tetap menggunakan *software development process* yang terdokumentasi dengan baik. Karena dengan adanya dokumentasi yang baik, akan memudahkan kita saat:

1. Melakukan *fine tuning* atau perbaikan program ke depannya.
2. Melakukan *maintenance*. Terutama saat ada pergantian *developer* yang terlibat.
3. Menambahkan modul lain ATAU membuat program lain yang memerlukan program kita saat ini sebagai salah satu modul pelengkap.

== END ==