**Pengenalan I : Algoritma dan Peran Pemrograman dalam Algoritma**

****

**Gambar 1**

Seiring perkembangan teknologi, muncul juga berbagai perangkat-perangkat yang menemani kehidupan sehari-hari seperti Smartphone, PC, Laptop dan lainnya. Perangkat-perangkat ini tidak akan dirasa berguna tanpa adanya berbagai program. Program-program yang terdapat di dalamnya diantaranya sistem operasi sebagai basis dari berjalannya program-program lain, perangkat lunak spesifik / produktifitas (seperti text processor atau spreadsheet, kalender, reminder), perangkat lunak utilitas (seperti penghemat baterai, penghapus berkas tidak berguna dll), dan bahkan game juga merupakan program untuk menemani di waktu luang atau sekedar menghabiskan waktu.

Sebagai pengguna, kita hanya tinggal menikmati keseluruhan dari apa yang ada di dalam perangkat-perangkat tersebut. Pada smartphone di zaman sekarang, kita hanya tinggal menuju store untuk mencari apa perangkat yang diinginkan kemudian melakukan instalasi. Tetapi pernahkah pengguna berpikir, bagaimana sebenarnya program-program itu dibuat ? Sebagai mahasiswa informatika, tentunya kita harus tahu lebih dalam dari pengguna awam.

Jika dilihat lebih dalam, sebuah ***program*** pada dasarnya dibuat untuk menyelesaikan ***masalah***. Kita ambil sebuah contoh misalnya sebuah program spreadsheet yang sudah sering kita gunakan yaitu Microsoft Excel. Program ini dapat menyelesaikan banyak masalah mulai dari masalah perhitungan sederhana, logika, statistika, matematik sampai pada pembuatan laporan.

Kita ambil lagi sebuah contoh program game Poker, apakah ada masalah yang diselesaikan ? Walaupun Poker cuma merupakan program Game, tetapi dibelakangnya banyak masalah-masalah yang diselesaikan agar dapat menjadi sebuah permainan yang menyenangkan untuk dimainkan. Berbagai masalah-masalah yang diselesaikan oleh poker misalnya adalah : pengacakan kartu, pembagian kartu, pengecekan kemenangan kartu, bagaimana agar AI selalu menang. Poker tentunya tidak akan menyenangkan jika ternyata pada pembagian kartu, terdapat kartu yang berulang, proses pembagian kartu yang tidak acak / berulang-ulang, dan pembayaran hadiah kemenangan yang salah). Sampai disini, kita dapat mengatakan bahwa sebuah program umumnya menyelesaikan banyak masalah sekaligus.

Pertanyaan selanjutnya adalah bagaimana program dapat menyelesaikan masalah ? Apakah ada sebuah entitas cerdas di dalamnya yang menyelesaikan masalah dan beradaptasi pada setiap permintaan-permintaan pengguna ?

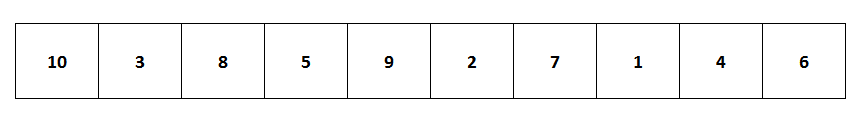
Tidak, masing-masing masalah yang diselesaikan oleh sebuah program diselesaikan oleh apa yang disebut dengan algoritma. Misal pada program Microsoft Excel, proses penggunaan rumus-rumus memiliki algoritmanya sendiri, proses membuatan laporan memiliki algoritmanya sendiri dan seterusnya. pada game Poker, proses pengacakan memiliki algoritmanya sendiri, proses pengecekan kemenangan memiliki algoritmanya sendiri dan seterusnya. Sampai disini kita tahu bahwa, sebuah program selain menyelesaikan banyak masalah juga terdiri dari banyak algoritma(jika masalah yang diselesaikan semakin banyak) dan algoritma adalah sesuatu yang digunakan untuk menyelesaikan masalah.

Untuk mengetahui lebih dalam mengenai algoritma akan dijelaskan lebih lanjut pada bab berikut Dari point ini sampai selanjutnya akan dijelaskan mengenai Masalah & Algoritma

**MASALAH**

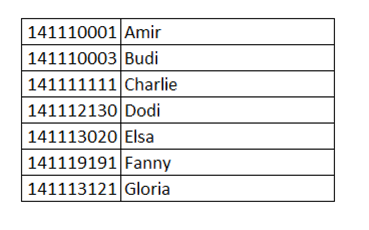
Jika sebelumnya diceritakan mengenai masalah dari sisi program, pada bab ini kita akan melihat masalah pada kehidupan sehari -hari. Coba perhatikan berbagai masalah kehidupan sehari-hari berikut ini :

a. Anda diminta untuk menyusun voucher-voucher berikut ini menjadi terurut, sehingga pihak panitia akan mudah melakukan tracking terhadap voucher yang sudah masuk.



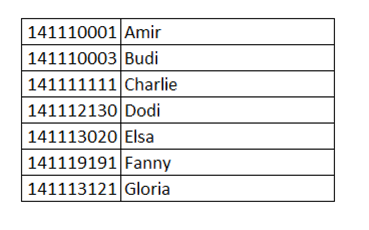
**Gambar 2**

b. Anda diminta untuk mencari apakah NIM 141112020 merupakan peserta ujian ?



**Gambar 3**

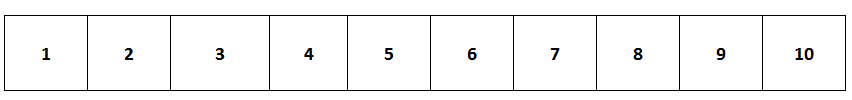
c. Anda diminta melakukan output pejabat dengan kekayaan tertinggi !



**Gambar 4**

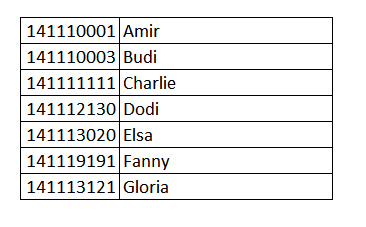
Berikut adalah solusi-solusi masalah-masalah diatas :

a.



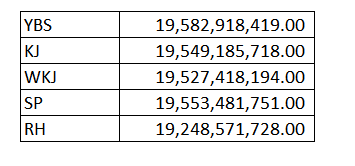
**Gambar 5**

b. 141112020 bukan peserta ujian



**Gambar 6**

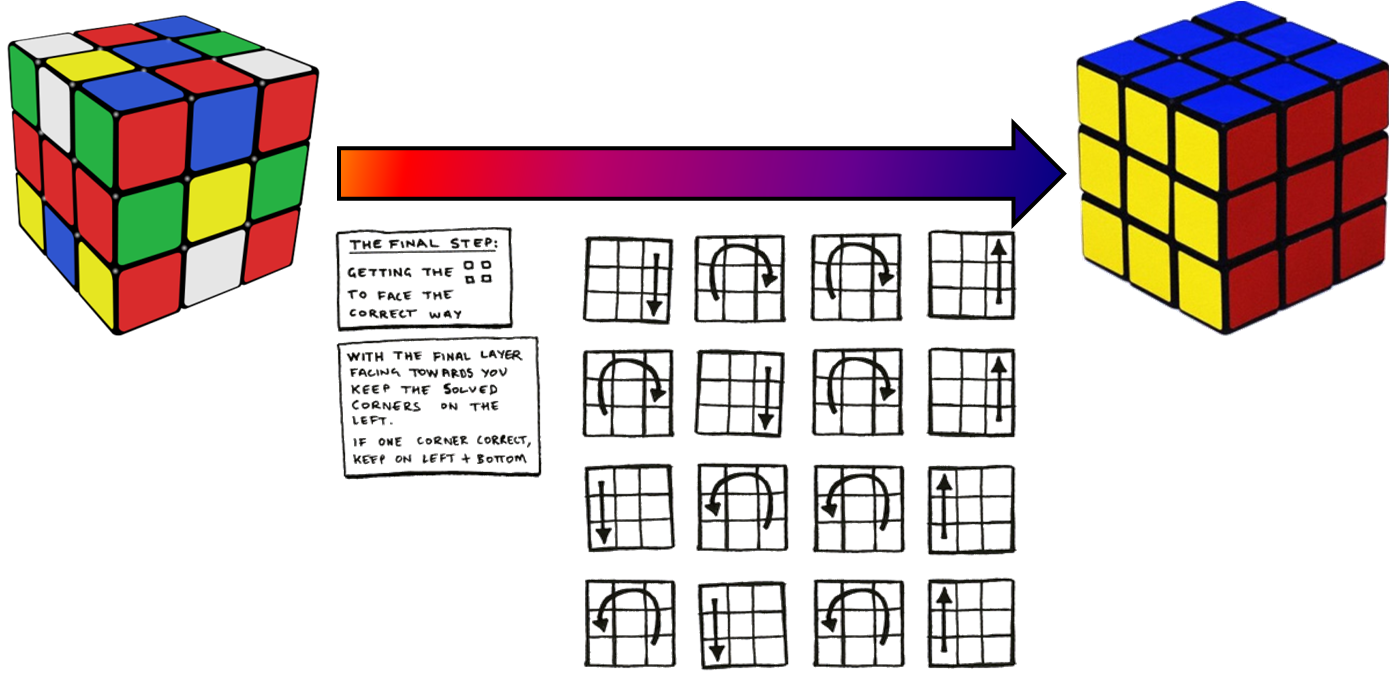
c. YBS adalah pejabat terkaya



**Gambar 7**

**ALGORITMA**

Sepintas melihat pada masalah-masalah diatas, secara manusiawi kita dapat dengan mudah atau bahkan luar kepala menyelesaikan masalah-masalah diatas. Tetapi jika jumlah masalah lebih besar atau rumit, kita akan merasakan bahwa sebenarnya kita berpikir keras, kita melakukan langkah-langkah dalam menyelesaikan masalah tersebut disadari atau tidak. Pada akhirnya, kita akan menyelesaikan proses berpikir atau langkah-langkah kita kita ketika kita sudah menemukan solusi. Proses berpikir atau langkah-langkah kita dalam menyelesaikan masalah inilah yang disebut dengan ALGORITMA.



**Gambar 8**

Coba bayangkan kembali, jika sebuah Rubik Cube yang belum terselesaikan adalah sebuah masalah dan Rubik Cube yang sudah selesai adalah solusinya. Maka langkah-langkah penyelesaian yang dilakukan dari masalah menjadi solusi adalah algoritmanya. Perlu diingat, algoritma yang baik adalah algoritma yang terurut dan efektif. Perhatikan kasus berikut dimana kita akan memindahkan isi dari gelas A ke gelas B dengan bantuan gelas kosong C, dapatkan Anda menentukan masalah awal, solusi, dan algoritmanya yang efektif ?

**Gambar 9**

Algoritma yang baik juga harus berhenti ketika sudah menemukan solusi. Sama seperti menyelesaikan pengurutan voucher atau menyelesaikan rubik cube, kita harus berhenti jika sudah menyelesaikan masalahnya.

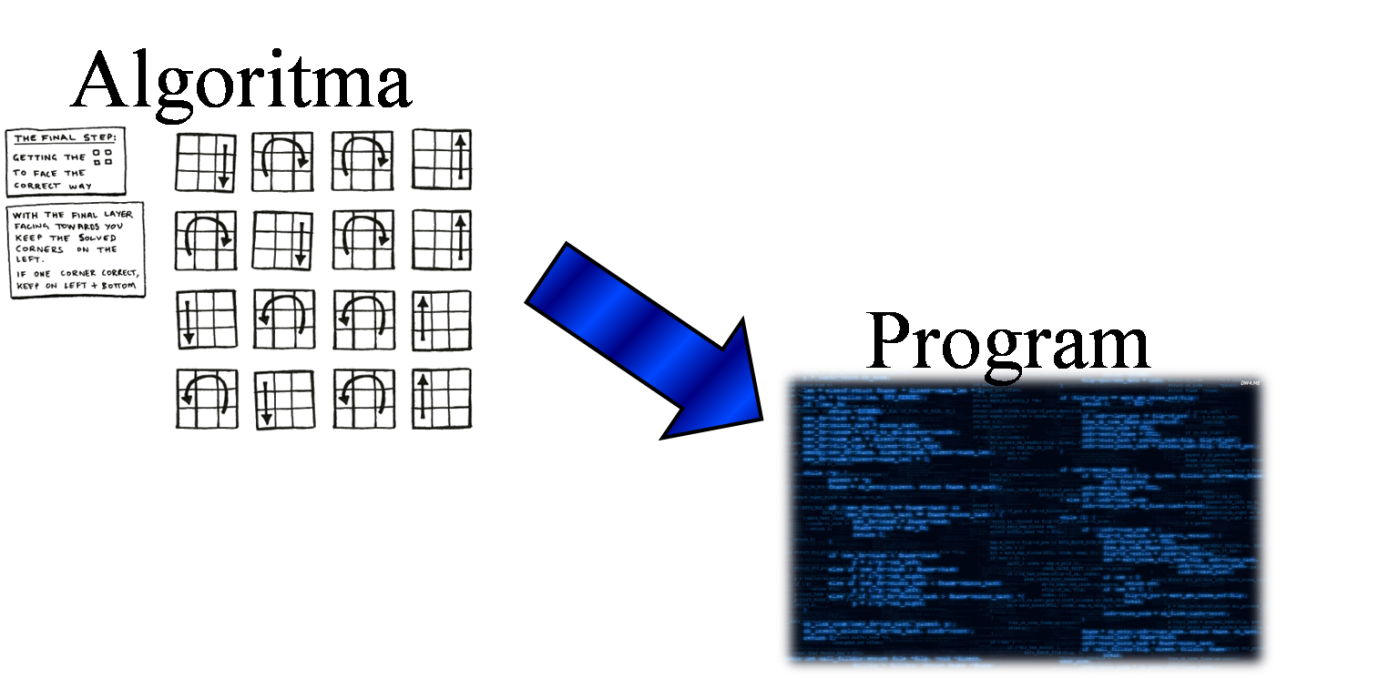
**PERAN PROGRAMMING PADA ALGORITMA**

**Gambar 10**

Perhatikan salah satu masalah diatas yang sudah kita selesaikan sebelumnya. Anggaplah *S* merupakan himpunan masalah yang akan kita selesaikan dan *n* merupakan besar / ruang masalah yang akan diselesaikan. Dengan *S* dan *n*, masalah akan cukup mudah diselesaikan oleh manusia dan dengan cepat.

**Gambar 11**

Tetapi bagaimana dengan S yang merupakan himpunan angka acak dari 1 sampai 10.000, tentunya dengan jumlah n sebesar 10.000 angka. Masalah ini juga bisa diselesaikan oleh manusia, tetapi akan membutuhkan waktu. Dengan perkembangan teknologi saat ini, masalah dengan ruang yang besar dapat diselesaikan oleh kekuatan komputasional (Komputer). Algoritma dapat diterjemahkan menjadi bahasa komputer untuk membantu menyelesaikan masalah. Algoritma dapat diterjemahkan menjadi bahasa komputer melalui pemrograman oleh seorang programmer. Disinilah peran pemrograman dan hubungannya dengan algoritma.



**Gambar 12**

Terdapat banyak bahasa pemrograman dalam dunia Computer Science, kita dapat memilih salah satu yang paling kita sukai dan sesuai dengan tujuan untuk mengakomodasi kita dalam menyelesaikan masalah. Bedasarkan “kemanusiawian”, bahasa pemrograman dikategorikan menjadi dua jenis :

* **Low-Level Programming Language,** merupakan bahasa pemrograman yang lebih dekat dengan bahasa mesin. Bahasa pemrograman ini masih berupa instruksi-instruksi yang terkadang tidak memiliki arti dengan bahasa manusia normalnya.
  + Assembly, OPCODE
* **High-Level Programming Language,** merupakan bahasa-bahasa pemrograman yang dibangun diatas Low-Level Programming Language, namun memiliki bahasa yang lebih masuk akal dan lebih bisa dimengerti karena mengikuti kaidah-kaidah bahasa yang ada.
  + Pascal, Ada, Cobol, Basic, Fortran, C, C++, **Python**

Pada mata kuliah ini, kita akan menggunakan bahasa pemrogrman Python. Perlu diingat bahwa, kita bukan mempelajari bahasa pemrograman Python, namun menggunakan Python sebagai alat untuk membantu kita mempelajari penyelesaian masalah melalui pembuatan algoritma dalam bahasa pemrograman.

Tentunya sebagai bahasa pemrograman pertama yang dipelajari, kita perlu tahu bagaimana menginstalasi Python, menggunakan Python dan menjalankan algoritma pertama kita dengan menggunakan Python. Latihan pada bab ini dapat dikerjakan untuk membantu pemahaman penggunaan Python sebelum masuk ke pemahaman algoritma lebih lanjut.

**Latihan**

1. Pada latihan ini, Dosen akan mendemokan instalasi Python versi 3.3. Silahkan melakukan pencatatan dan mengajukan pertanyaan untuk mendapatkan informasi lebih lanjut mengenai instalasi Python.

2. Pada latihan ini, Dosen akan menjelaskan dan mendemokan konsep IDE pada pemrograman. Silahkan melakukan pencatatan dan mengajukan pertanyaan untuk mendapatkan informasi lebih lanjut mengenai penggunaan IDE Python.

3. Pada latihan ini, Dosen akan mendemokan beberapa algoritma sederhana yang ditulis dalam bahasa Python.

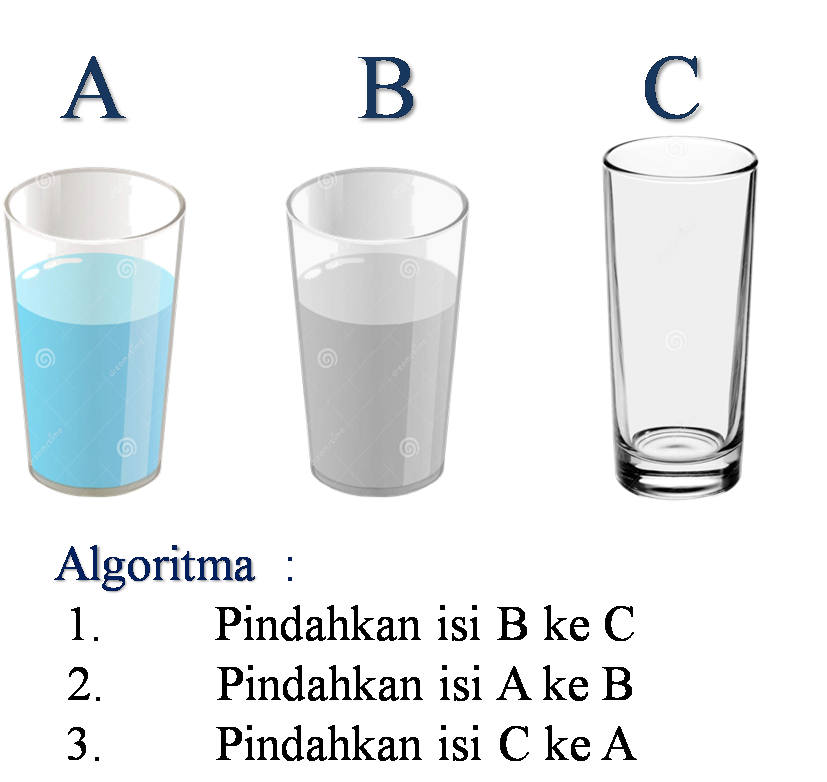
4. Pada latihan ini, Dosen akan menjelaskan dan mendemokan konsep Compiling dan Running bahasa pemrograman Python. Silahkan melakukan pencatatan dan mengajukan pertanyaan untuk mendapatkan informasi lebih lanjut mengenai penggunaan IDE Python.

**Pengenalan IIa : Struktur Dasar Algoritma**

Pada bab sebelumnya, kita mengetahui bahwa Program merupakan kumpulan dari algoritma-algoritma. Algoritma sendiri di dalamnya juga merupakan kumpulan dari struktur-struktur dasar pembangun algoritma. Terdapat tiga jenis utama struktur algoritma yaitu :

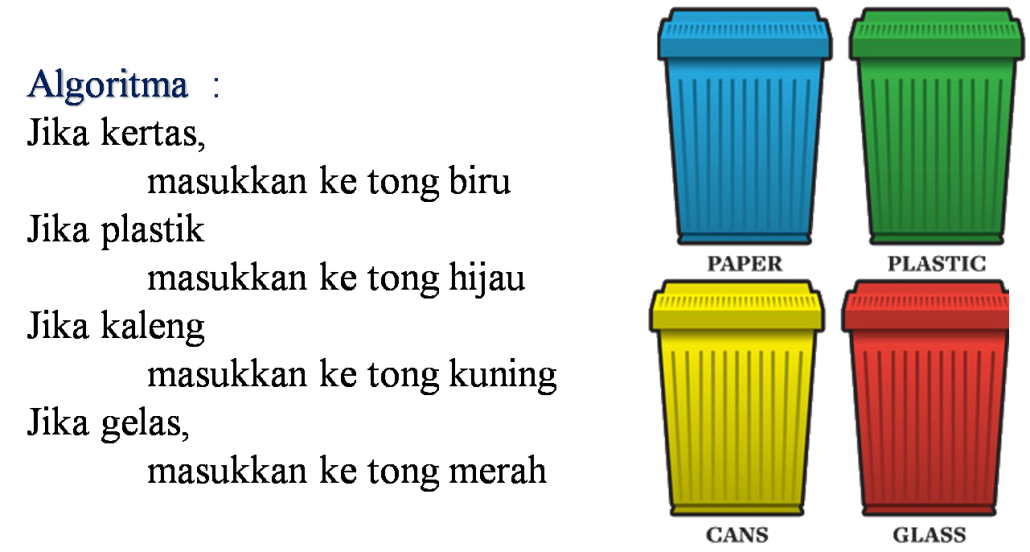
* + Sequence (Runtunan)
  + Selection / Branching (Pemilihan / Percabangan)
  + Repetition / Looping (Pengulangan)

**Sequence (Runtunan)** merupakan struktur algoritma paling dasar yang berisi rangkaian instruksi yang diproses secara sekuensial, satu per satu, mulai dari instruksi pertama sampai instruksi terakhir. Salah satu contoh Sequence(runtunan) ada pada contoh bab sebelumnya yaitu algoritma memindahkan isi dari kedua gelas.

****

**Gambar 13**

**Selection / Branching (Pemilihan / Percabangan)** merupakan struktur algoritma yang akan digunakan dimana jika terdapat alternatif/pilihan beberapa sequence(runtunan) yang akan dijalankan jika memenuhi syarat tertentu. Percabangan dapat diibaratkan sebagai persimpangan jalan yang harus dipilih, jika kita memilih sebuah jalan dari persimpangan tersebut sudah pasti kita tidak akan menjalani yang lainnya. Salah satu contoh percabangan dapat dilihat pada contoh “Algoritma Memilih Tempat Pembuangan” berikut ini :



**Gambar 14**

**Repetition / Looping (Pengulangan)** merupakan struktur algoritma yang akan digunakan dimana jika terdapat pengulangan terhadap satu pernyataan atau sequence (runtunan) tertentu. Beberapa contoh nyata penggunaan pengulangan misalnya pada saat harus melakukan penelusuran angka-angka voucher dari awal sampai akhir. Contoh lain yang lebih dekat dengan dunia nyata misalnya adalah algoritma “pembagian permen” berikut ini :

****

**Gambar 15**

Percabangan dan Perulangan akan dibahas lebih lanjut pada pertemuan 3 ke atas. Perlu diperhatikan bahwa masing-masing dari struktur dasar ini tidak berdiri sendiri dan terpisah. Pada kasus pembuatan algoritma sebenarnya, struktur dasar ini akan saling dipasangkan satu sama lain. Beberapa kasus dari penggabungan struktur dasar ini adalah sebagai berikut :

* + Perulangan bersarang
  + Percabangan bersarang
  + Percabangan dalam Percabangan
  + Repetition dalam Percabangan
  + Percabangan dalam Perulangan
  + Perulangan dalam Selection
  + Perulangan dalam Perulangan dengan Selection
  + Dan kombinasi lainnya

**Pengenalan IIb: Variabel, Tipe Data & Operator**

**TIPE DATA**

Dalam mengubah masalah sehari-hari menjadi algoritma, kemudian dari algoritma menjadi bahasa program, kita harus melakukan proses penyederhanaan informasi-informasi dari masalah tersebut. Misalnya, ada masalah menukar isi dalam gelas, kita memerlukan sesuatu yang dapat merepresentasikan / mewakili informasi isi dari gelas seperti sirup atau air sebenarnya ke dalam algoritma atau bahasa program. Dari sini kita memiliki beberapa alternatif, apakah kita akan mewakilinya dengan kata atau dengan nilai. Dari sinilah muncul konsep tipe-data dalam algoritma. Pada pengenalan akan dikenalkan dahulu beberapa tipe data dasar, kemudian seiring dengan perkembangan mata kuliah, tipe data baru dan yang lebih muktahir akan diperkenalkan.

Beberapa tipe data dasar yang umum pada algoritma adalah sebagai berikut :

* + Bilangan Bulat: Integer, Long
  + Bilangan Riil: Float
  + Logika : Boolean
  + Karakter : Char
  + Kumpulan Karakter : String

Secara umum, tipe data-tipe data inilah yang digunakan untuk mewakiliki kasus-kasus dunia nyata yang akan diterjemahkan menjadi algoritma kemudian menjadi bahasa program.

**VARIABEL**

Sebagai wadah untuk merepresentasikan dunia nyata pada algoritma, tipe data sendiri tidak cukup. Dengan adanya tipe data saja, maka dalam merepresentasikan masalah kita hanya terbatas pada pembentukan KONSTANTA saja. Misal : “Sirup”, “86,6”, “151”. Padahal pada kasusnya, informasi-informasi pada masalah sehari-sehari memiliki nilai yang mungkin berubah-ubah. algoritma juga harus menyediakan wadah untuk menangani hal ini. Disinilah peran VARIABEL dibutuhkan.

VARIABEL pada algoritma digunakan untuk menampung nilai dengan tipe data tertentu. Variabel dimaksudkan untuk menyimpan informasi berupa nilai yang dapat secara dinamis berubah, diberi nilainya ataupun dibaca kembali pada saat dibutuhkan. Jika “Sirup” merupakan sebuah nilai dengan tipe data String, dan kita dapat menampungnya dalam sebuah variabel yang bernama GelasA. Yang kita tahu pada kasusnya nanti GelasA akan berubah nilainya ketika ditukar dengan isi dari GelasB.

Beberapa peraturan dalam penggunaan variabel adalah sebagai berikut :

* Penamaan variabel sebaiknya eksplisit sesuai dengan tujuan dari pembuatan variabel tersebut.
* Penamaan variabel tidak boleh melibatkan spasi !

Misalnya : GelasA daripada A, GelasB daripada B

* Penamaan variabel sebaiknya dimulai dengan huruf
* Pada bahasa pemrograman tertentu, penamaan variabel biasanya case-sensitive.

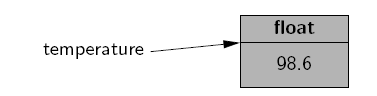
Contoh : nama berbeda dengan Nama berbeda dengan NAMA berbeda dengan namA

* Penamaan variabel jangan bertabrakan dengan reserved-word pada bahasa pemrograman tertentu !

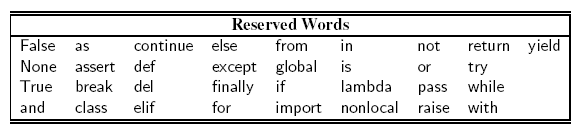
Beberapa hal yang akan sering kita lakukan pada pembuatan variabel di algoritma adalah deklarasi dan assignment. Proses inisialisasi awal variabel yang melibatkan pemberian tipe data disebut deklarasi. Proses pemberian nilai pada variabel disebut assigment

**VARIABEL & TIPE DATA PADA PYTHON**

Khusus pada bahasa pemrograman Python, tidak ada deklarasi variabel yang dilibatkan. Pembuatan variabel pada python pada umumnya langsung dibuat melalui proses assignment. Setelah proses assignment, Python akan langsung menyesuaikan tipe data pada variabel tersebut. Hal ini disebabkan karena Python adalah dynamically-typed language. (Optional : Perlu diingat kita bukan mempelajari Python, tetapi Pengantar Algoritma !)

****

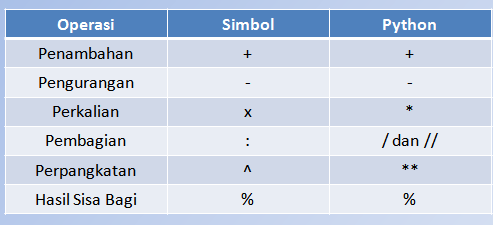
Perlu diingat juga, peraturan penggunaan variabel sama seperti sub-bab sebelumnya. Dengan tambahan daftar reserved-words adalah sebagai berikut :

****

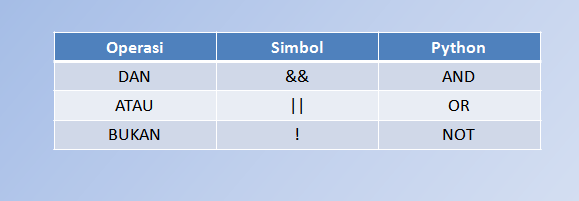
**OPERATOR**

Pada algoritma, terdapat konsep yang dinamakan operator. Operator berfungsi untuk menghubungkan satu atau lebih variabel sehingga menghasilkan nilai baru. Variabel-variabel yang turut serta dalam sebuah operator disebut operand. Beberapa jenis operator dasar pada pemrograman adalah sebagai berikut :

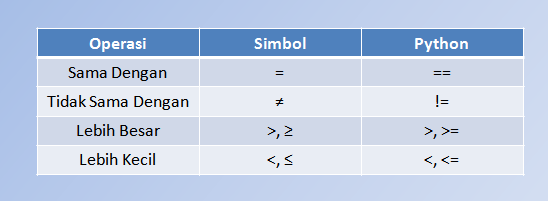
* Operator Aritmatika

****

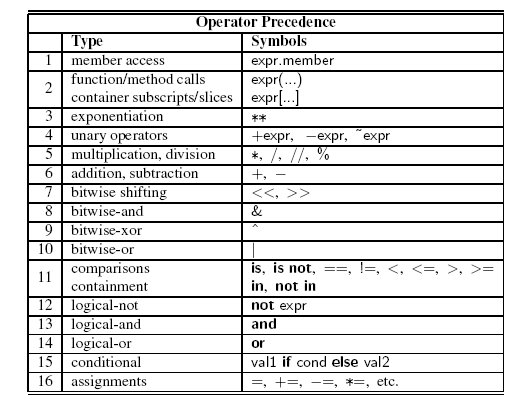
* Operator Logika

****

* Operator Perbandingan

****

Operator memiliki prioritas eksekusi. Sama seperti pada matematika dimana pada sebuah perhitungan, operator kali akan dijalankan terlebih dahulu dibanding penambahan dan pengurangan karena memiliki kekuatan operator yang lebih tinggi. Berikut ini merupakan kekuatan operator pada bahasa pemrograman Python : (Tidak masalah jika terdapat operator-operator yang belum diketahui, Anda cukup mengetahui urutan prioritas operator-operator dasar sebelumnya dari tabel dibawah)

****

**Pengenalan IIc : Input Proses Output**

Sampai bab ini, kita sudah memiliki pengetahuan yang cukup untuk membuat salah struktur algoritma dasar yaitu sequence(runtunan). Kembali lagi perlu diingatkan bahwa algoritma dibuat untuk menyelesaikan masalah. Ketika bertemu masalah, agar cara berpikir kita rapi dan efektif, maka alangkah baiknya mengikuti “rule of thumb” berikut ini :

* Apa **masukkan** dari masalah ini ? Ini berhubungan dengan bentuk awal dari masalah. Pertanyaan-pertanyaan ini selanjutnya boleh diperluas dengan memikirkan, berapa variabel yang dibutuhkan di awal? tipe variabel apa saja yang dibutuhkan di awal ?
* Apa **keluaran** dari masalah ini ? Kapan algoritma akan berhenti ? Apakah algoritma cukup berhenti jika semua proses sudah dijalankan ? Jika sudah berhenti variabel apa yang akan saya tampilkan kepada pengguna ? Atau bentuk output apa yang akan saya tampilkan kepada pengguna ? Apakah
* Terakhir, apa **proses(algoritma inti) dari** masalah ini ? Apakah saya cukup menggunakan runtunan ? Apakah perlu struktur dasar lainnya (Pertemuan berikut !) ? Operator-operator apa saja yang perlu saya gunakan. Perlukah ada variabel-variabel pembantu lainnya ?

Mari kita gunakan rule of thumb ini kembali pada masalah yang sudah sering kita gunakan yaitu masalah menukarkan isi dari gelas.

Apa **input** dari masalah ini ? Dari gambar dibawah, kita tahu bahwa keadaan awal dari masalah ini adalah terdapat dua gelas(Gelas A dan Gelas B) dengan isi yang berbeda. Kita dapat menyimpulkan bahwa kita memerlukan dua variabel berbeda dengan isi yang berbeda.

GelasA = “Kopi”

GelasB = “Teh”

Apa **output** dari masalah ini ? Output dari masalah ini adalah kita cukup mencetak isi dari masing-masing gelas tersebut. Secara intuitif kita akan tahu bahwa algoritma sudah benar ketika isi dari GelasA dan GelasB sudah tertukar. Program juga akan berhenti ketika seluruh proses sudah dijalankan, bisa jadi program berhenti dalam keadaan GelasA dan GelasB tidak tertukar. Hal ini dikarenakan terdapat kesalahan pada proses. Tetapi mari kita simpulkan bahwa output yang diharapkan adalah :

GelasA berisi Teh

GelasB berisi Kopi

Bagian terpenting adalah apa proses yang baik untuk masalah ini. Apakah diperlukan variabel tambahan ? Mari kita melakukan analisa dengan melihat dari sisi dunia nyata. Perhatikan jika kita langsung menuangkan A langsung ke B. Maka yang akan terjadi adalah B akan menjadi Teh-Kopi, begitu juga sebaliknya. Dari sini kita memutuskan boleh menggunakan Gelas tambahan yaitu gelas C. Misalnya kita memilih isi gelas A untuk dipindahkan ke gelas C, maka gelas A akan kosong. Setelah itu kita dapat memindahkan isi gelas B ke gelas A, sehingga gelas A berisi teh, kemudian gelas C kita pindahkan ke gelas A sehingga gelas A sudah berisi kopi. Dengan begini kita akan mendapatkan hasil akhir sesuai output yang diinginkan.

Secara algoritma kita dapat menuliskannya sebagai berikut :

GelasC = GelasA

GelasA = GelasB

GelasB = GelasC

Secara keseluruhan adalah sebagai berikut :

**TukarIsiGelas**(GelasA:String, GelasB:String)

Deklarasi : GelasC:String

GelasC = GelasA

GelasA = GelasB

GelasB = GelasC

Output(GelasA, GelasB)

End

Latihan

1. Sesuai petunjuk Dosen, catatlah petunjuk bagaimana melakukan konversi algoritma sebelumnya menjadi bahasa pemrograman !

2. Sesuai petunjuk Dosen, catatlah petunjuk bagaimana input-output berbeda sesuai pada kontes pemrograman !

3. Konversi Masalah berikut menjadi algoritma, kemudian konversi lagi menjadi bahasa pemrograman Python :

* Menghitung Luas Persegi Panjang
* Menghitung Luas Lingkaran
* Konversi Suhu
* Mencari titik tengah dari 2 buah titik
* Detik ke Menit

Dosen akan menunjukkan dua input yang berbeda (Pertama, input dengan gaya normal dan Kedua, input gaya kontes pemrograman)