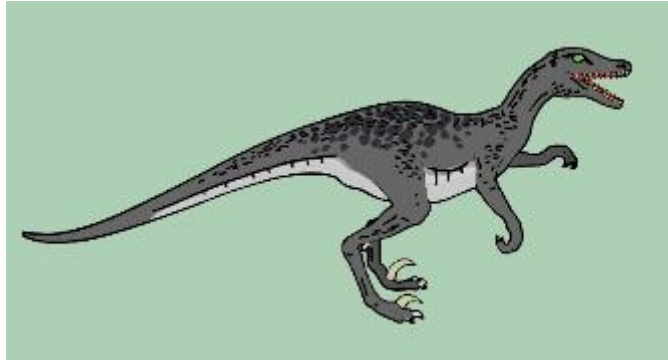


# SIMULASI ALGORITMA DAN FLOWCHART VIA RAPTOR

[*RAPTOR* merupakan pemrograman yang berbasis *flowchart*, *RAPTOR* dirancang khusus untuk membantu memvisualisasikan algoritma yang telah kita buat. Program *RAPTOR* diciptakan secara visual dan dieksekusi secara visual dengan menelusuri eksekusi melalui *flowchart*. Biasanya kita lebih suka menggunakan *flowchart* untuk mengekspresikan algoritma, dan lebih berhasil menciptakan algoritma menggunakan *RAPTOR* daripada menggunakan bahasa tradisional atau menulis *flowchart* tanpa *RAPTOR*.]

# RAPTOR – FLOWCHART INTERPRETER



RAPTOR merupakan pemrograman yang berbasis *flowchart*, RAPTOR dirancang khusus untuk membantu memvisualisasikan algoritma yang telah kita buat. Program RAPTOR diciptakan secara visual dan dieksekusi secara visual dengan menelusuri eksekusi melalui *flowchart*. Biasanya kita lebih suka menggunakan *flowchart* untuk mengekspresikan algoritma, dan lebih berhasil menciptakan algoritma menggunakan RAPTOR daripada menggunakan bahasa tradisional atau menulis *flowchart* tanpa RAPTOR.

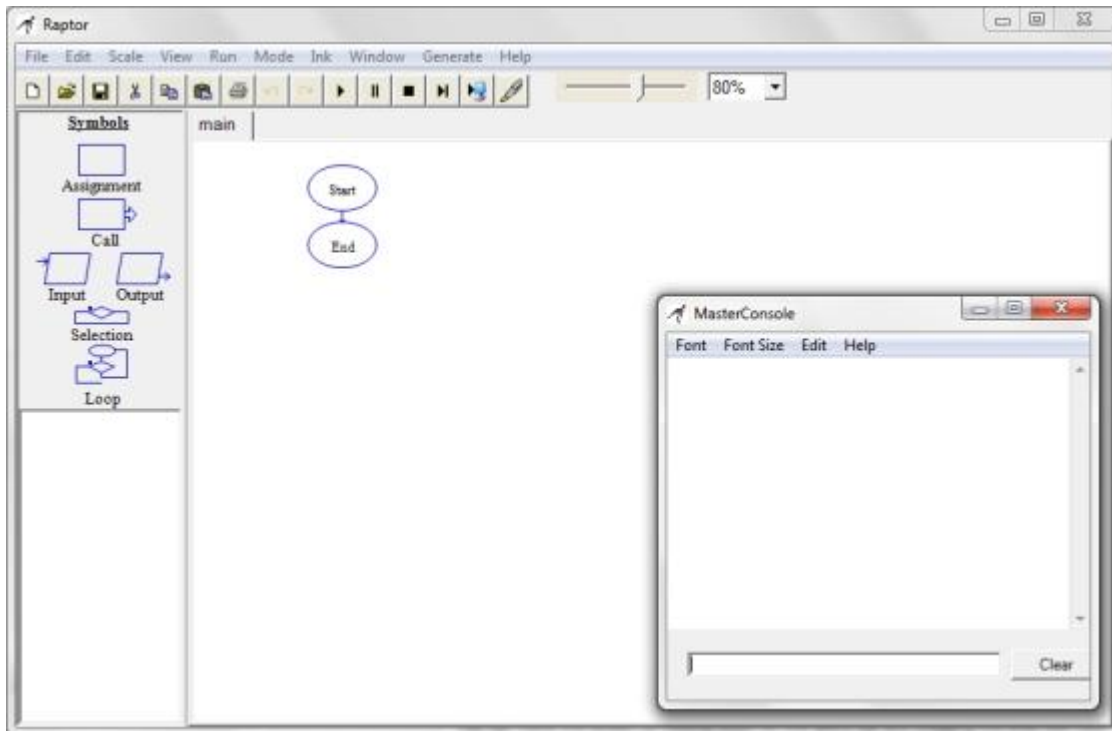
RAPTOR ditulis dalam kombinasi dari bahasa C # dan A # (*port* dari Ada untuk .NET Framework.) dan hanya didukung pada Windows. RAPTOR telah bereksperimen dengan Mono di Mac OS X dan Ubuntu. Versi Mac tidak berjalan sama sekali, tetapi RAPTOR dapat berjalan pada Ubuntu dengan beberapa fitur yang harus dihilangkan.

RAPTOR memiliki beberapa mode, secara *default* kita memakai mode *Novice*. Mode *Novice* memiliki global namespace tunggal untuk setiap variabel. Mode *Intermediate* digunakan untuk membuat prosedur yang memiliki ruang lingkup mereka sendiri (memperkenalkan gagasan lewat parameter dan mendukung rekursi). Mode baru RAPTOR adalah mode yang berorientasi object, yaitu versi 2009.

RAPTOR bebas untuk didistribusikan sebagai layanan kepada masyarakat. RAPTOR pada awalnya dikembangkan oleh dan untuk US Air Force Academy, Departemen Ilmu Komputer, namun penggunaannya telah menyebar dan RAPTOR sekarang digunakan untuk pendidikan di lebih 17 negara pada setidaknya 4 benua.

RAPTOR juga dilengkapi dengan proses generate *flowchart* ke beberapa source code yang sudah banyak di kenal seperti C++, Java, C# dan lain-lain. Sehingga pengguna tidak perlu lagi membangun dari awal sebuah source




code, karena dari *flowchart* yang telah di buat langsung di terjemahkan ke *Source Code* oleh *RAPTOR*.



Keunggulan dari software *RAPTOR Interpreter Flowchart* adalah dapat mengeksekusi *flowchart* yang telah di bangun menjadi sebuah visualisasi yang nyata, sehingga pengguna dapat mengetahui *step by step flowchart* yang mereka buat melalui eksekusi secara visual dalam tiap langkahnya. Selain itu kelebihan dari *Raptor Interpreter Flowchart* ini ada pada saat kita membuat *flowchart* dengan penulisan variabel, *RAPTOR* di dukung dengan adanya fitur *Auto Complete* seperti layaknya pada *Pemrograman Visual Basic*.

Gambar diatas merupakan *interface* dari *RAPTOR*. Disebelah kiri merupakan simbol-simbol yang dapat kita gunakan untuk membuat sebuah *flowchart*. Berikut adalah fungsi dari simbol-simbol diatas :

Purpose	Symbol	Name	Description
PROCESSING		Assignment	Digunakan untuk mengubah nilai variabel menggunakan beberapa jenis perhitungan matematis.
INPUT		input statement	Digunakan untuk untuk memasukkan data. Setiap nilai data disimpan dalam variabel.
PROCESSING		procedure call	Digunakan untuk menjalankan sekelompok instruksi yang didefinisikan dalam prosedur bernama. Dalam beberapa kasus beberapa argumen

			prosedur (yaitu, variabel) akan diubah oleh instruksi prosedur itu
OUTPUT		output statement	Digunakan untuk tampilan (menyimpan ke file) nilai variabel
Selection		Condition	Digunakan untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya / tidak
Loop		Looping	Digunakan untuk menunjukkan suatu perulangan

## Operator dan Fungsi

Operator atau fungsi memerintahkan komputer untuk melakukan beberapa perhitungan pada data. Operator ditempatkan antara data yang dioperasikan (yaitu  $X / 3$ ,  $Y + 7$ , dll) sedangkan fungsi menggunakan tanda kurung untuk menunjukkan data tersebut beroperasi pada (yaitu  $\text{sqrt}(4.7)$ ,  $\text{sin}(2,9)$ ). Ketika dieksekusi, operator dan fungsi melakukan perhitungan dan mengembalikan hasil. RAPTOR memiliki operator dan fungsi sebagai berikut :

- *basic math* : +, -, \*, /, ^, \*\*, rem, mod, sqrt, log, abs, ceiling, floor.
- *trigonometry* : sin, cos, tan, cot, arcsin, arcos, arctan, arccot.
- *relational* : =, !=, /=, <, >, >=, <=.
- *logical* : and, or, not.
- *miscellaneous* : random, Length\_of.

Operator matematika dasar dan fungsi yang termasuk biasa (+, -, \*, /) serta beberapa yang tidak biasa.

- “\*\*” dan “^” adalah *exponentiation*, seperti  $2 ** 4$  adalah 16,  $3 ^ 2$  adalah 9.
- *rem (remainder)* dan *mod (modulus)* mengembalikan sisa (apa yang tersisa) ketika operan kanan membagi operan kiri, contoh :  $10 \text{ rem } 3$  adalah 1,  $10 \text{ mod } 3$  adalah 1 juga.
- *sqrt* mengembalikan akar kuadrat, contoh :  $\text{sqrt}(4)$  adalah 2.
- *log* mengembalikan logaritma natural, contoh :  $\log(e)$  adalah 1.
- *abs* mengembalikan nilai absolut, contoh :  $\text{abs}(-9)$  adalah 9.
- *ceiling* pada seluruh nomor, contoh :  $\text{ceiling}(3,14159)$  adalah 4.
- *floor* pada seluruh nomor, contoh :  $\text{floor}(10/3)$  adalah 3.
- “+” juga bekerja sebagai operasi *concatenation* untuk menggabungkan dua string atau string dan angka, contoh : “rata-rata adalah” + (Jumlah / Angka).

- *length\_of* mengembalikan jumlah karakter dalam sebuah variabel string (juga jumlah elemen dengan sebuah array), contoh : Nama ← "Stuff" diikuti dengan *Length\_Of* (Nama) adalah 5.

Kita terbiasa dengan fungsi trigonometri (sin, cos, tan, cot, arcsin, arcos, arctan, arccot). Mereka bekerja pada berbagai unit yang bernilai radian. (kita harus mengkonversi dari derajat ke radian sebelum menggunakan fungsi tersebut.). arctan dan arccot adalah versi kedua parameter fungsi ini. (yaitu arctan (X / Y) ditulis dalam RAPTOR sebagai arctan (X, Y)).

Dalam RAPTOR, operator relasional dan operator logika hanya dapat digunakan dalam pengambilan keputusan sebagai bagian dari statement *Selection* dan *Loop*. Operator relasional adalah != = (tidak sama dengan), / = (tidak sama dengan), <, >, > = dan <=. Operator relasional mengembalikan nilai "Boolean" dalam "True" atau "False" (ya atau tidak). Sebagai contoh, operasi X < Y akan mengembalikan TRUE jika nilai yang tersimpan dalam variabel X kurang dari nilai yang disimpan dalam variabel Y. Jika tidak nilai FALSE dikembalikan. Hasil dari operasi relasional dapat digunakan oleh operator logika.

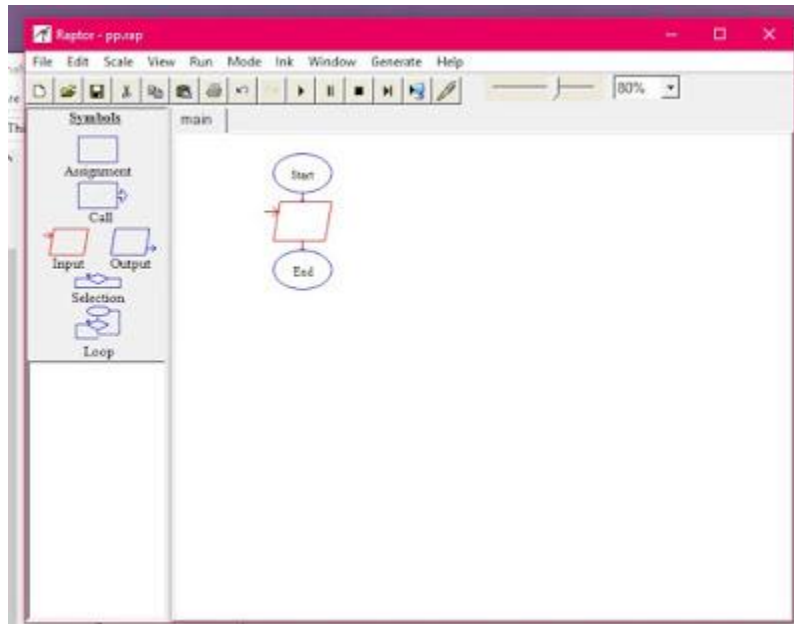
Operator logika didefinisikan oleh tabel berikut. Operan yang digunakan oleh operator logika harus bernilai "Boolean" (artinya nilai-nilai yang dikembalikan oleh operator relasional atau operator logis).

Fungsi secara acak mengembalikan angka antara 0 dan 1, contoh : X ← secara acak bisa menjadi 0, 0,23, 0,46578, dll. Jika kita memerlukan nomor acak dalam kisaran yang berbeda maka kita bisa menggabungkan fungsi acak dengan operasi lain. Misalnya, random \* 100 akan mengevaluasi ke angka antara 0 dan 100. ceiling (random \* 100) akan mengevaluasi ke seluruh nomor antara 1 dan 100.

Adapun langkah untuk membuat flowchart menggunakan aplikasi RAPTOR dapat diikuti dengan kasus untuk menghitung luas persegi panjang sebagai berikut :

### Langkah 1

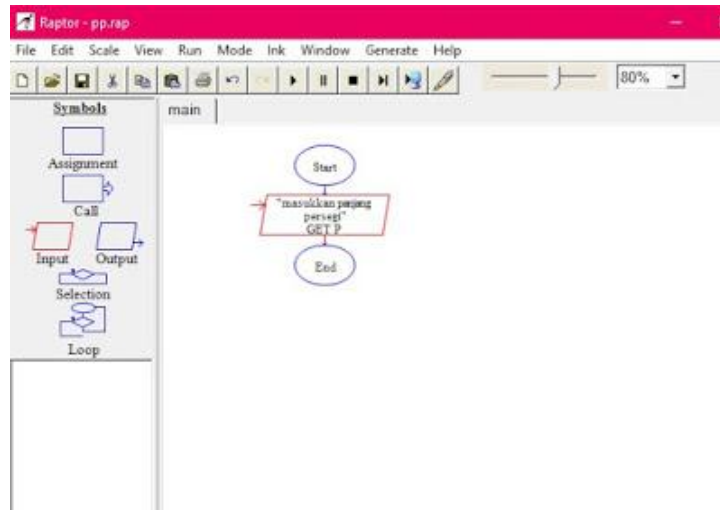
Pada awal akan muncul simbol start untuk memulai dan end untuk mengakhiri, kemudian masukkan simbol input untuk menginputkan program.



Klik ganda pada simbol input maka akan muncul kotak dialog

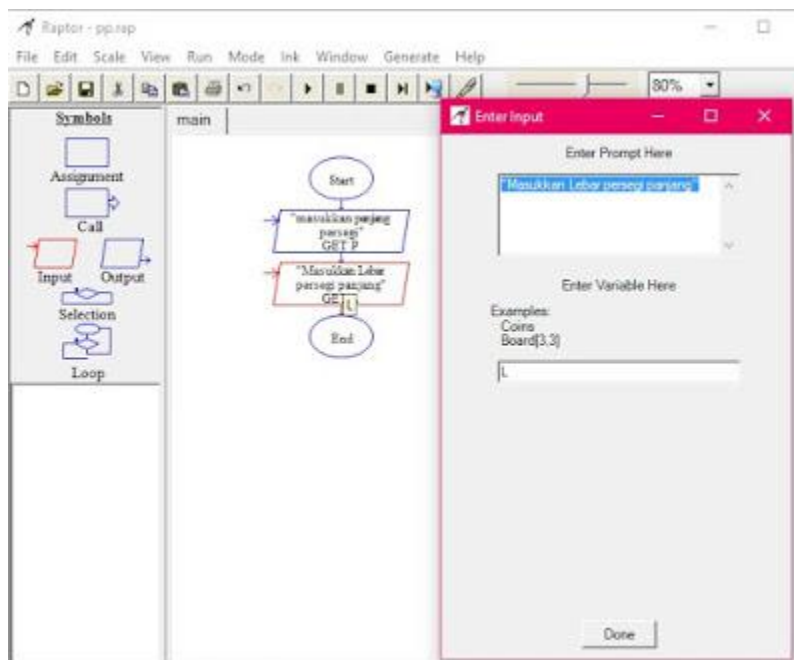
The "Enter Input" dialog box is shown with a title bar containing the text "Enter Input". It has a standard Windows window control bar with minimize, maximize, and close buttons. The dialog contains two main sections. The first section is labeled "Enter Prompt Here" and features a text input field with the text "masukkan panjang persegi" inside. The second section is labeled "Enter Variable Here" and includes a list of examples: "Coins", "Board[3,3]", and a text input field with the character "P" inside. At the bottom of the dialog is a "Done" button.

Enter Prompt Here digunakan untuk label keterangan contohnya “Masukkan Nilai Panjang” sedangkan Enter Variabel Here digunakan untuk Memasukkan nama variabelnya, contohnya panjang. Setelah itu klik done maka akan tampil seperti dibawah ini :



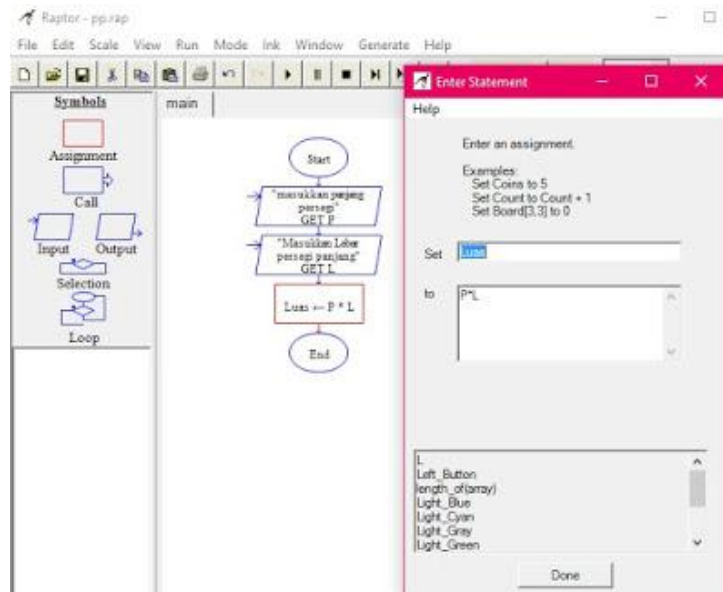
## Langkah 2

Klik kembali simbol input dengan cara yang sama seperti langkah 1



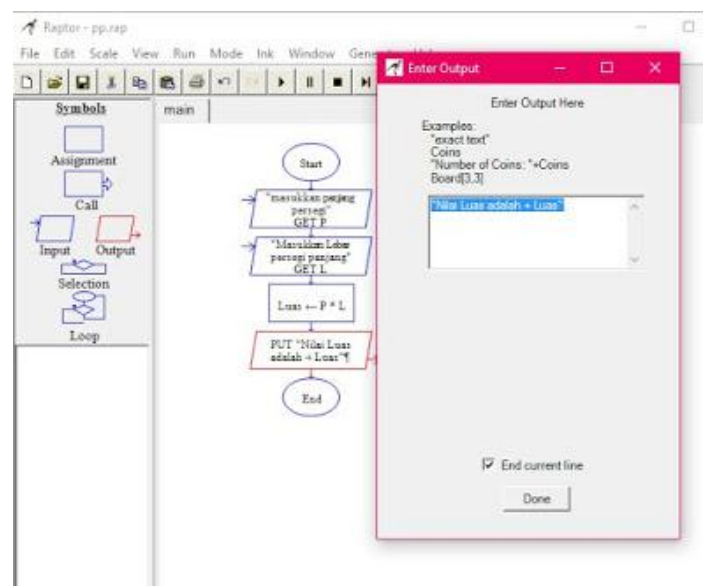
### Langkah 3

Untuk menghitung luas maka gunakan simbol assignment letakkan diatas simbol end



### Langkah 4

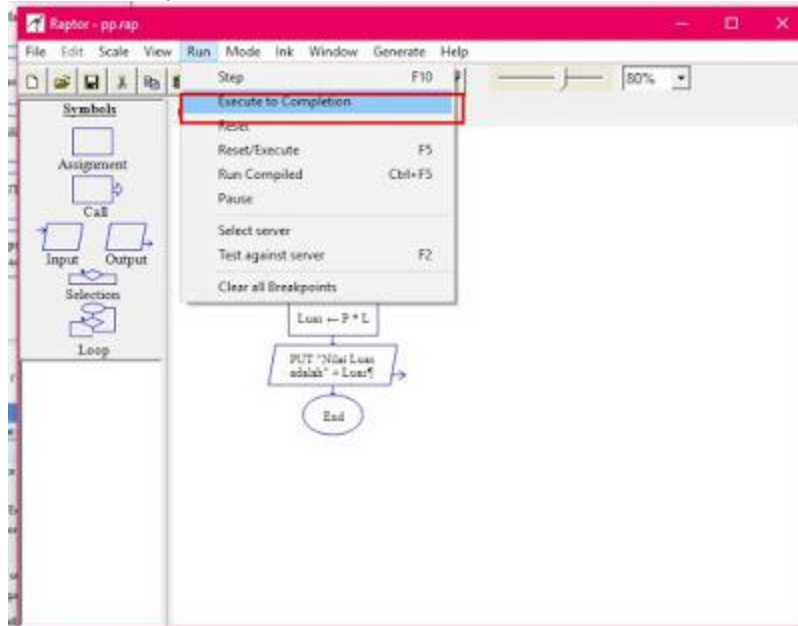
Mencetak nilai luas dengan cara klik simbol output, kemudian letakkan di atas end. Klik 2 kali pada simbol output kemudian muncul kotak dialog Enter output, pada Enter Output Here ketikkan "Nilai Luas adalah " + luas kemudian done.





## Langkah 5

Maka flowchart telah selesai, selanjutnya adalah menjalankan programnya. Dengan cara klik menu Run pilih Execute to Completion maka program akan berjalan.



Tugas 1.

Buatlah algoritma dan flowchart menggunakan Raptor untuk menghitung luas dan keliling dari persegi panjang

Tugas 2.

Buatlah algoritma dan flowchart menggunakan Raptor untuk menghitung luas dan keliling lingkaran

Tugas 3.

Buatlah algoritma dan flowchart menggunakan Raptor untuk menghitung volume balok

Tugas 4.

Buatlah algoritma dan flowchart menggunakan Raptor untuk menghitung volume tabung

Tugas 5.

Buatlah algoritma dan flowchart menggunakan Raptor untuk menghitung konversi suhu dari  $C^{\circ}$  ke  $F^{\circ}$ ;  $R^{\circ}$ ;  $K^{\circ}$