Teknik Kompilasi

Kompilator (Compiler) adalah :
Sebuah program yang membaca suatu program yang ditulis dalam suatu *bahasa sumber* (*source language*) dan menterjemahkannya ke dalam suatu *bahasa sasaran* (*target language*).

Proses kompilasi dapat digambarkan melalui sebuah kotak hitam (black box) berikut :

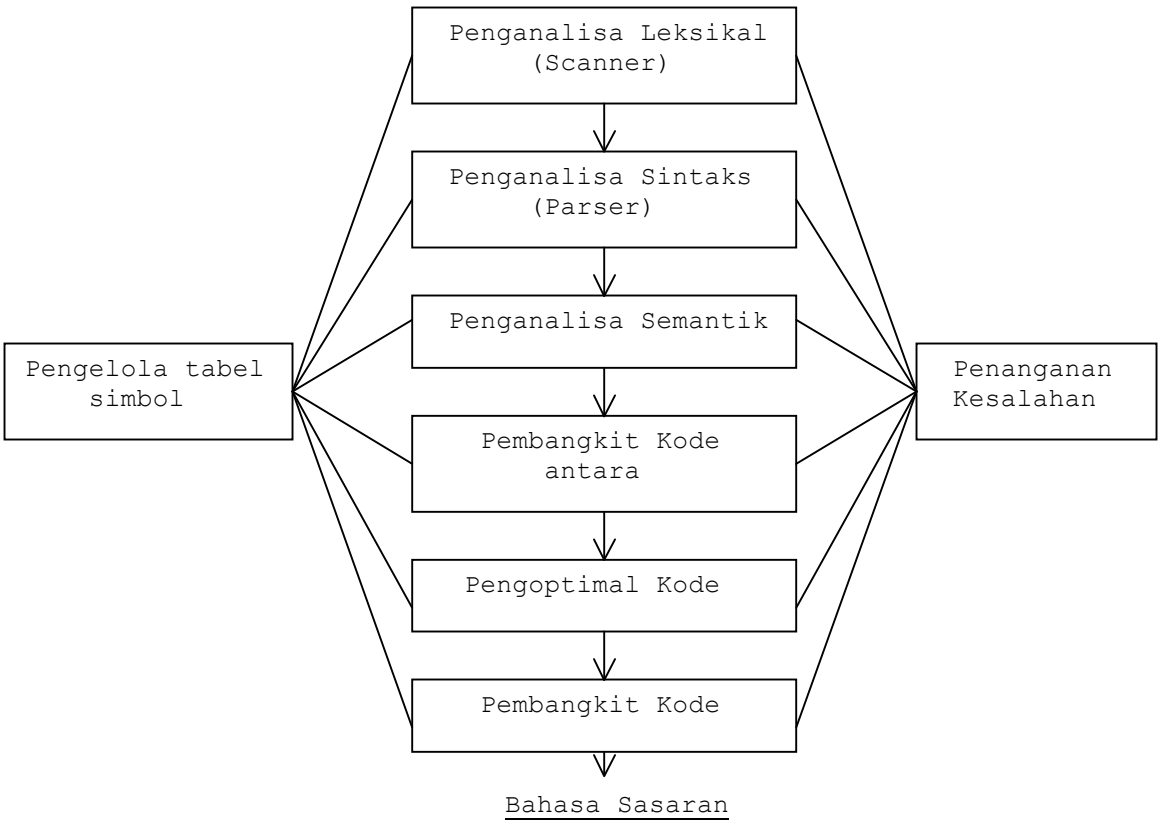
Program Sumber	→	*Kompilator*	→	Bahasa Sasaran
		↓		
Pesan-pesan kesalahan				
		(error messages)		

Proses kompilasi dikelompokkan menjadi :

- Analisa : program sumber dipecah-pecah dan dibentuk menjadi bentuk antara (intermediate presentation)

- Sintesa : membangun program sasaran yang diinginkan dari bentuk antara

Fase-fase proses sebuah kompilasi :



Hal-hal yang dilakukan oleh setiap fase pada proses kompilasi terhadap program sumber :

- ***Penganalisa leksikal*** :

Membaca program sumber, karakter demi karakter

Sederetan karakter dikelompokkan menjadi satu kesatuan, mengacu kepada *pola kesatuan kelompok karakter* (token) yang ditentukan dalam *bahasa sumber*.

Kelompok karakter yang membentuk sebuah token dinamakan *lexeme* untuk token tersebut.

Setiap token yang dihasilkan disimpan di dalam tabel simbol.

Sederetan karakter yang tidak mengikuti pola token akan dilaporkan sebagai *token tak dikenal* (*unidentified token*).

- ***Penganalisa Sintaks*** :

Memeriksa kesesuaian pola deretan token dengan aturan sintaks yang ditentukan dalam bahasa sumber.

Sederetan token yang tidak mengikuti aturan sintaks akan dilaporkan sebagai *kesalahan sintaks* (*syntax error*).

Secara logika deretan token yang bersesuaian dengan sintaks tertentu akan dinyatakan sebagai *pohon parsing* (*parse tree*).

- ***Penganalisa Semantik*** :

Memeriksa token dan ekspresi dari batasan-batasan yang ditetapkan.

Batasan-batasan tersebut misalnya :

a. panjang maksimum token identifier adalah 8 karakter
b. panjang maksimum ekspresi tunggal adalah 80 karakter
c. nilai bilangan bulat adalah -32768 s.d 32767
d. operasi aritmatika harus melibatkan operan-operan yang bertipe sama

- ***Pembangkit Kode Antara*** :

Membangkitkan *kode antara* (*intermediate code*) berdasarkan pohon parsing.

Pohon parse selanjutnya diterjemahkan oleh suatu penterjemah yang dinamakan *penerjemah berdasarkan sintaks* (*sintax directed translator*).

Hasil penerjemahan ini biasanya merupakan *perintah tiga alamat* (*tree-address code*) yang merupakan representasi program untuk suatu mesin abstrak.

Perintah tiga alamat bisa berbentuk :
quadruples (*op, arg1, arg2, result*), *triples* (*op, arg1, arg2*).

Ekspresi dengan satu argumen dinyatakan dengan menetapkan *arg2* dengan - (*strip, dash*).

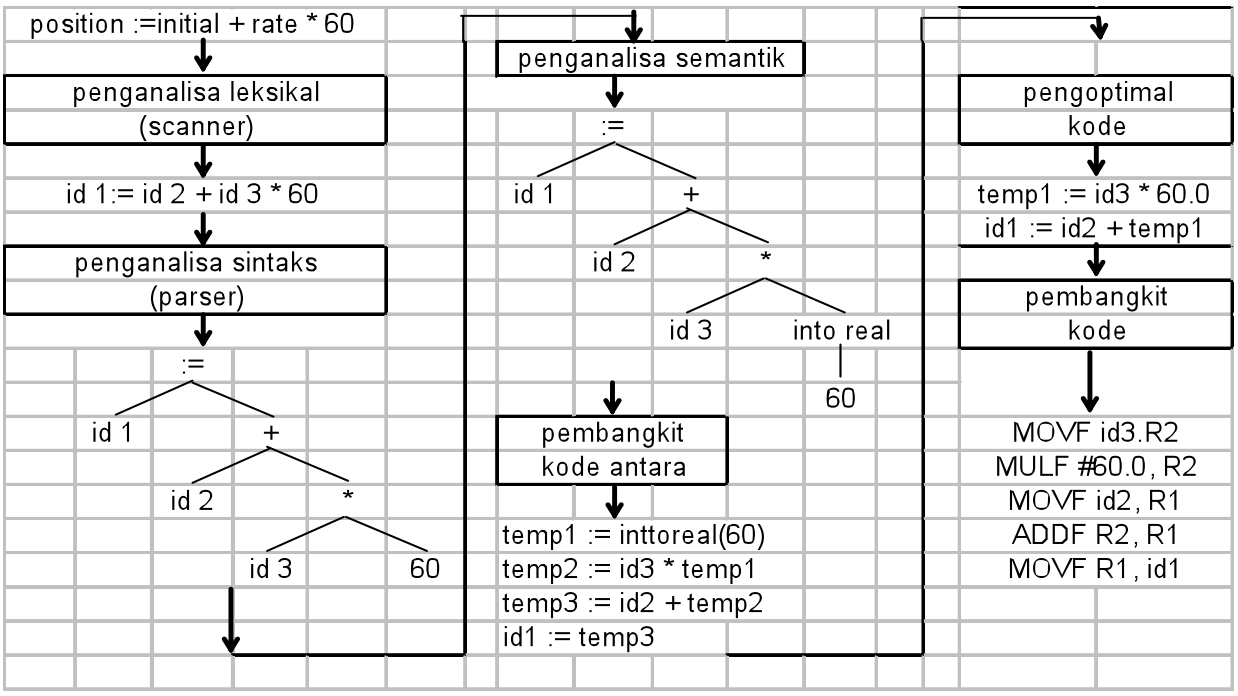
- ***Pengoptimal Kode*** :

Melakukan optimasi (penghematan space dan waktu komputasi), jika mungkin, terhadap kode antara.

- ***Pembangkit Kode*** :

Membangkitkan kode dalam bahasa target tertentu (misalnya bahasa mesin).

Contoh skema penerjemahan suatu ekspresi dalam bahasa sumber, yaitu : position := initial + rate * 60



Keterangan :

- id adalah token untuk *identifier*. Tiga *lexeme* untuk token ini adalah position, initial dan rate.
- Penganalisa semantik secara logika membangkitkan pohon parse.
- Penganalisa semantik mendeteksi *mismatch type*. Perbaikan dilakukan dengan memanggil *procedure* inttoreal yang mengkonversi integer ke real.
- Quadruples dari : *temp2 := id3 * temp1* adalah *(*, id3, temp1, temp2)*, *id := temp3* adalah *(assign, temp3, -, id1)*, *temp1 := inttoreal(60)* adalah *(inttoreal, 60, -, temp1)*.
- Pembangkit kode dalam contoh ini menghasilkan kode bahasa mesin.