



Pendahuluan

Kuliah online : Tekkom [2013/VI]



Sekilas tentang bahasa

- Dalam dunia sehari-hari, Bahasa digunakan sebagai alat untuk berkomunikasi, contohnya bahasa Indonesia
- Sedangkan untuk berkomunikasi dengan komputer, Segala instruksi harus dimasukan ke dalam komputer dengan bahasa yang dimengerti oleh komputer yang disebut **BAHASA PEMROGRAMAN**



Bahasa Pemrograman

- Digunakan sebagai sarana komunikasi antara manusia dan permasalahan – permasalahannya dengan komputer yang dipakai untuk membantu memperoleh pemecahan.
- Untuk menjembatani antara pemikiran manusia yang tidak terstruktur dengan kepastian yang diperlukan oleh komputer untuk dieksekusi.
- Suatu solusi untuk suatu masalah akan menjadi lebih mudah bila bahasa pemrograman yang dipergunakan lebih dekat dengan permasalahan tersebut



Tingkatan bahasa pemrograman

- Bahasa Mesin (Machine Level Language)
- Bahasa Rakitan (Assembly Language)
- Bahasa tingkat tinggi (User Oriented Language)
- Bahasa Berorientasi Masalah (Problem Oriented Language)



Bahasa Mesin

- Merupakan bentuk terendah dari bahasa Komputer.
- Setiap instruksi dalam program direpresentasi dengan kode numeric, yang secara fisik berupa deretan angka 0&1.
- Sekumpulan instruksi dari bahasa mesin bisa dibentuk menjadi *microcode* yaitu semacam prosedur dalam bahasa mesin



Bahasa Assembly

- Merupakan bentuk simbolik dari bahasa mesin.
- Setiap kode operasi memiliki kode simbolik misalnya: ADD (penjumlahan), MUL (Perkalian).
- Sekumpulan instruksi dalam bahasa assembly bisa dibentuk menjadi makroinstruksi



Bahasa Tingkat Tinggi (User Oriented)

- Disebut tingkat tinggi karena lebih dekat dengan manusia
- Memberikan fasilitas yang lebih banyak, kontrol program yang terstruktur, bercabang (nested), block dan prosedur
- Contoh : Pascal dan BASIC



Bahasa yang Problem Oriented

- Memungkinkan penyelesaian untuk suatu masalah atau aplikasi yang spesifik.
- Contoh : SQL untuk aplikasi database



Keuntungan Bahasa tingkat tinggi dibandingkan tingkat rendah

- Kemudahan untuk dipelajari, tidak membutuhkan latar belakang pengetahuan perangkat keras.
- Lebih mendekati permasalahan yang akan diselesaikan.
- Pemrogram tidak perlu mengetahui bagaimana representasi data kedalam bentuk internal memory.
- Memberikan banyak pilihan struktur kontrol seperti : kondisional, looping, struktur blok dan nested statement.



Keuntungan Bahasa tingkat tinggi dibandingkan tingkat rendah

- Program lebih mudah di-debug
- Kemampuan struktur data yang lebih, sehingga memfasilitasi pengekspresian suatu solusi dari masalah tertentu.
- karena ketersediaan *features* seperti prosedur, bahasa tingkat tinggi memungkinkan suatu deskripsi modular dan hirarkis dalam pemrograman.
- kompatibilitas dan dokumentasi yang lebih baik dalam pengembangan program.
- tidak bergantung pada mesin sehingga memiliki portabilitas tinggi



Translator

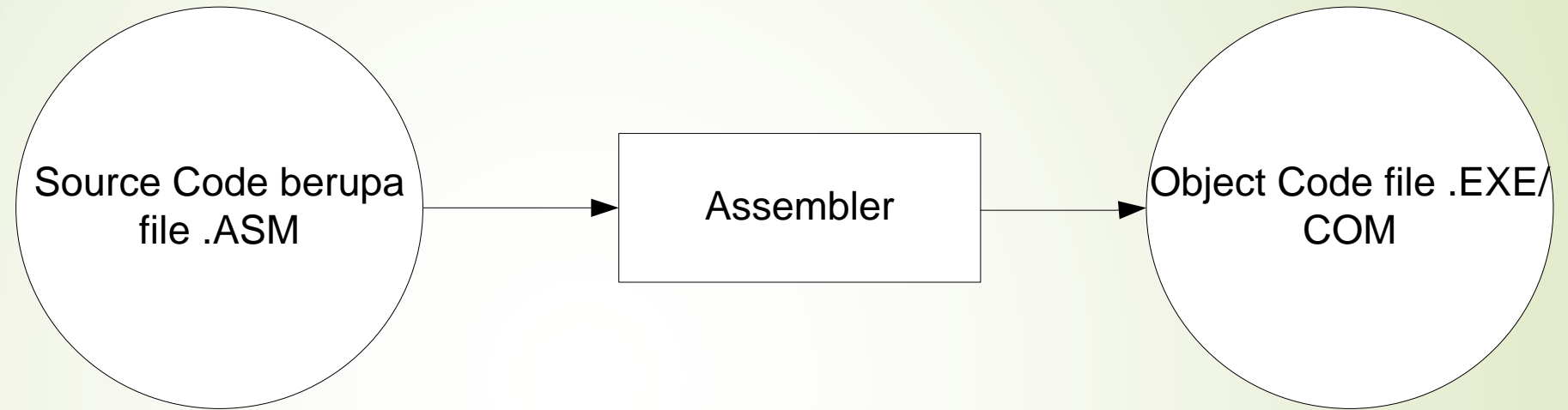
- Sebuah translator melakukan pengubahan source code/source program (program sumber) ke dalam target code/object code/object program(program object).
- Source code ditulis dalam bahasa sumber sedang object code bias berupa suatu bahasa pemrograman lain atau bahasa mesin pada suatu komputer
- Ada beberapa macam Translator : Assembler, Kompilator dan Interpreter



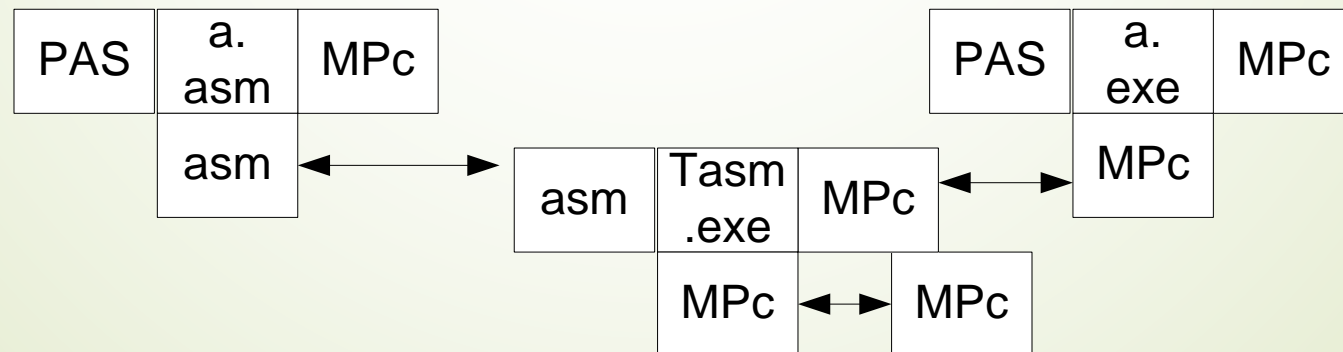
Assembler

- Source Code adalah bahasa assembler, object code adalah bahasa mesin
- Contohnya : Turbo Assembler dan macro assembler

Assembler



ATAU

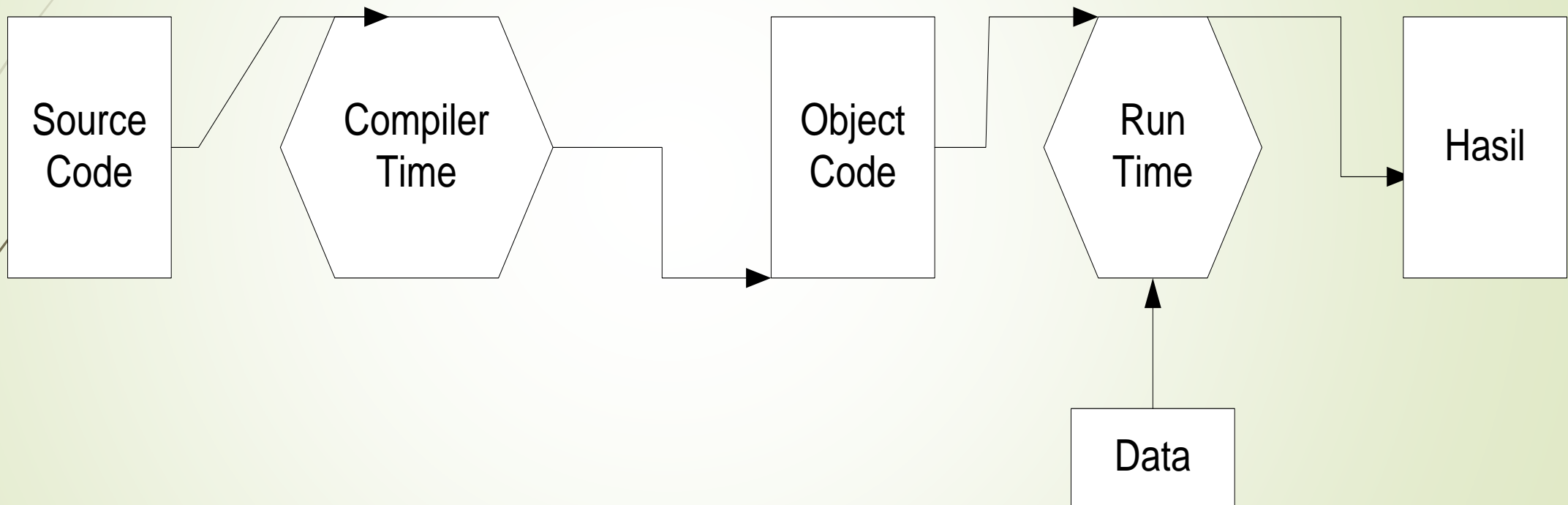




Kompilator (Kompile)

- Source code adalah bahasa tingkat tinggi (misal bahasa Pascal)
- Object Code adalah bahasa mesin atau bahasa assembly.
- Source code dan data diproses pada saat yang berbeda.
- Contohnya : Turbo Pascal,

Kompilator (Compiler)

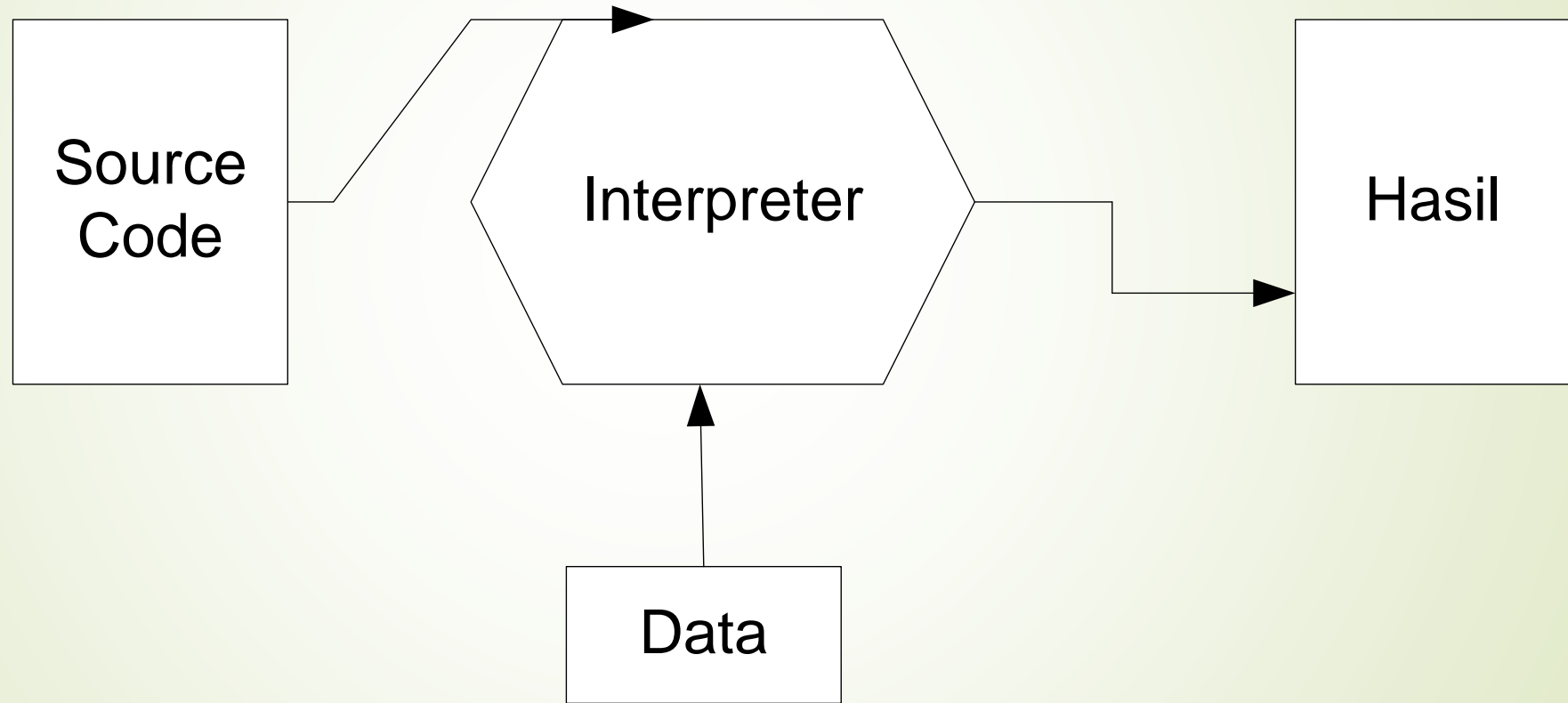




Interpreter

- Interpreter tidak membangkitkan object code, hasil translasi hanya dalam bentuk internal.
- Contoh interpreter : BASIC, LISP, SMALL TALK, MATLAB.
- Source Code data diproses dalam waktu yang sama.
- Proses interpretasi dapat dilihat pada gambar :

Interpreter



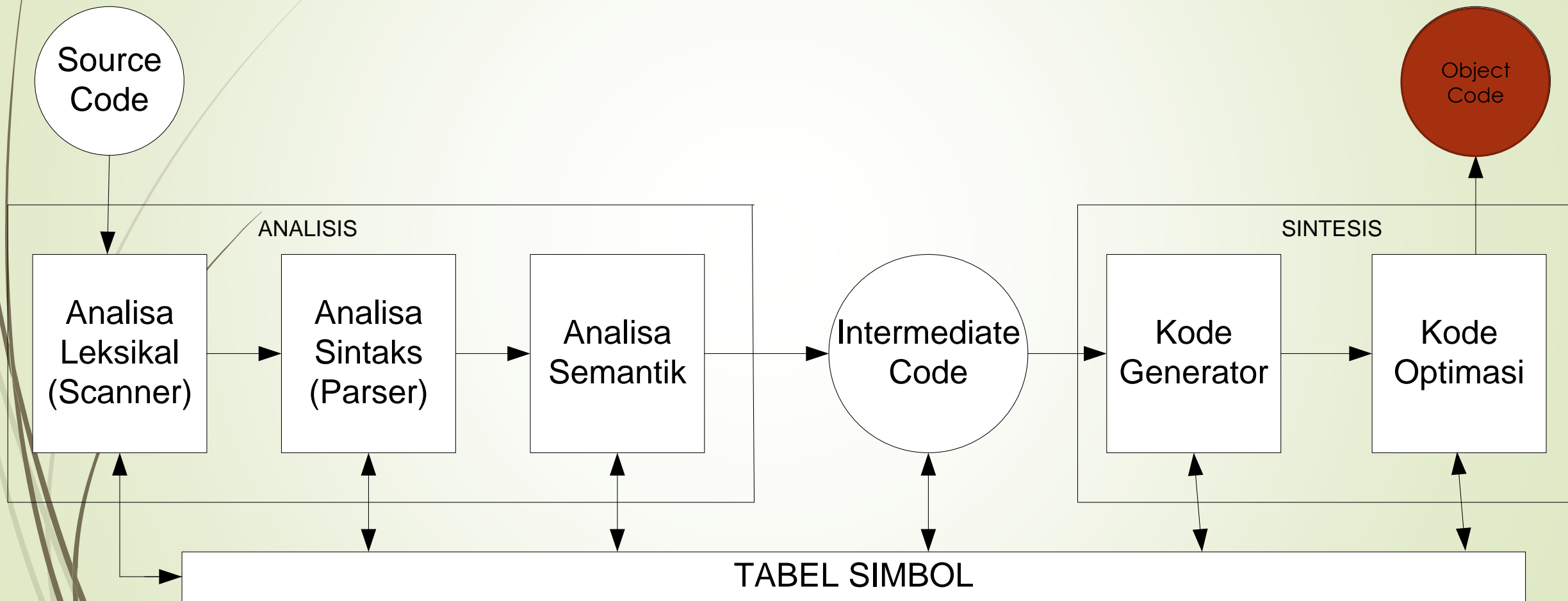


Model Kompilator



- Pengembangan kompilator untuk sebuah bahasa merupakan pekerjaan yang kompleks.
- Sebuah kompilator pada umumnya memiliki 2 tugas pokok :
 - Fungsi Analisis (Front End) -> Tugasnya melakukan dekomposisi program sumber menjadi bagian-bagian dasarnya.
 - Fungsi Sintesis (Back End) -> Tugasnya melakukan pembangkitan dan optimasi program objek.

Model Kompilator



$position := initial + rate * 60$

penganalisa leksikal
(*scanner*)

$id1 := id2 + id3 * 60$

penganalisa sintaks
(*parser*)

$id1 := id2 + id3 * 60$

penganalisa semantik

$id1 := id2 + id3 * \text{inttoreal}(60)$

pembangkit
kode antara

$temp1 := \text{inttoreal}(60)$
 $temp2 := id3 * temp1$
 $temp3 := id2 + temp2$
 $id1 := temp3$

pengoptimal kode

$temp1 := id3 * 60.0$
 $id1 := id2 + temp1$

pembangkit kode

$MOV\ id3,\ R2$
 $MUL\ \#60.0,\ R2$
 $MOV\ id2,\ R1$
 $ADD\ R2,\ R1$
 $MOV\ R1,\ id1$



Mutu Kompilator

- Mutu sebuah kompilator tergantung dari beberapa factor, yaitu
- 1. Kecepatan dan waktu proses kompilasi -> tergantung pada penulisan algoritma kompilator dan kompilator pengkompilasi
- 2. Mutu Program Objek -> Ukurannya lebih kecil dan lebih cepat dieksekusi
- 3. Integrated Environment -> fasilitas-fasilitas terintegrasi yang dimiliki oleh kompilator, spt *editing*, *debugging* dan *testing*



Pembuatan Kompilator

- Bahasa Mesin
 - Bahasa Assembly
 - Bahasa tingkat tinggi
 - Bootstrap
- 



Bahasa Mesin

- Sangat sukar dan sangat sedikit kemungkinannya untuk membuat *compiler* dengan bahasa ini, karena manusia susah mempelajari bahasa mesin
- Sangat tergantung pada mesin
- Bahasa mesin kemungkinan digunakan pada saat pembuatan *Assembler*



Bahasa Assembler

- Hasil dari program mempunyai ukuran yang relative kecil
- Sulit dimengerti karena statement/perintahnya singkat-singkat
- Fasilitas yang dimiliki masih terbatas

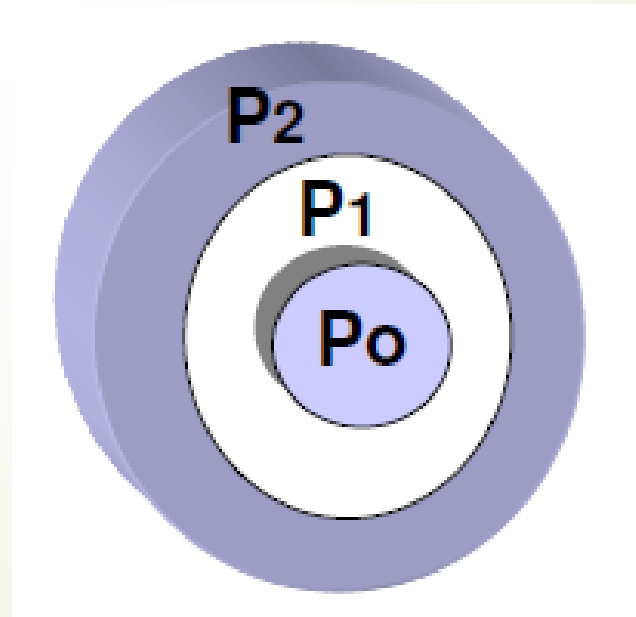


Bahasa Tingkat tinggi

- Lebih mudah dipelajari
- Fasilitas yang dimiliki lebih banyak
- Memiliki ukuran yang relatif besar, misal membuat compiler pascal dengan menggunakan bahasa C
- Untuk mesin yang berbeda, perlu dikembangkan tahapan-tahapan tambahan, misal membuat compiler C pada Dos berdasarkan compiler C pada Unix

BootStrap

- Untuk membangun sesuatu yang besar, dibangun/dibuat dulu bagian intinya.
- Cara ini diperkenalkan oleh Niklaus Wirth saat membuat kompilator untuk bahasa pascal





Tugas 1



1. Jelaskan, mengapa kita memerlukan bahasa pemrograman?
2. Berikan penjelasan, mengenai istilah2 berikut : kompilator, translator, interpreter, assembler & emulator
3. Berikan beberapa contoh produk yang ada di pasaran untuk setiap istilah pada no 2.
4. Model Kompilator dapat disebut sebagai one-pass compiler, jelaskan alasannya
5. Apa keuntungan dari penggunaan metode BootStrap



감사합니다 [Terima kasih 😊]