

# MODEL KOMPILATOR

## FUNGSI UMUM BAGIAN KOMPILATOR

Dosen Pengampu:  
Utami Dewi Widianti,S.Kom

# (1) Program Sumber

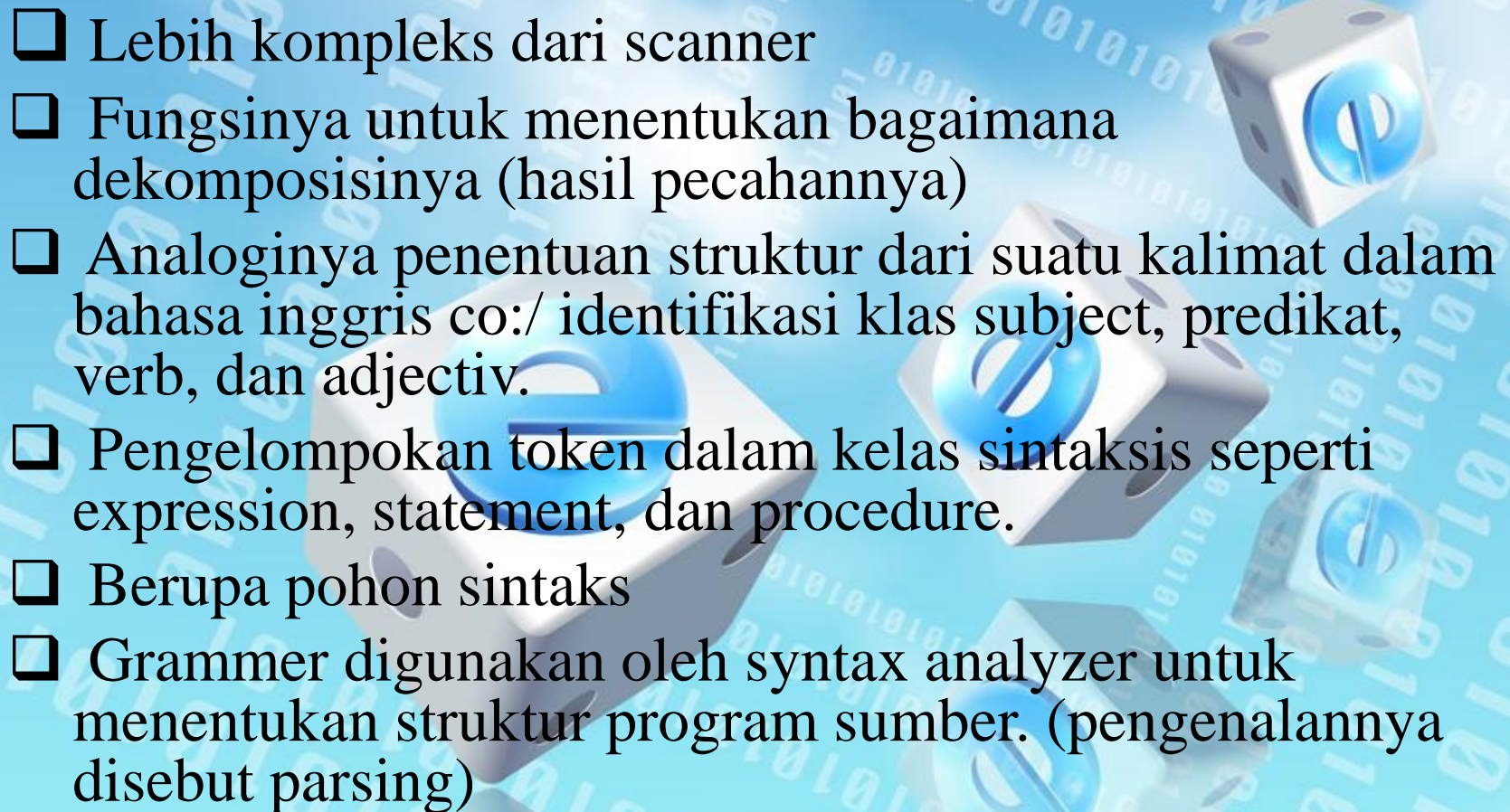
- ❑ Deretan simbol berupa huruf, digit, atau simbol tertentu seperti +, -, dan ,
- ❑ Program sumber memuat konstruksi bahasa yang mendasar seperti nama variable, label, konstanta, keyword, dan operator.
- ❑ Konstruksi bahasa merupakan definisi dari bahasa itu sendiri.



## (2) Lexical Analyzer / Scanner

- ❑ Tujuannya memisahkan text yang masuk ke dalam token, co:/ konstanta, nama variabel, keyword (DO,IF,THEN) dan operator.
- ❑ Melakukan analisis sintaksis level rendah.
- ❑ Klas dari token diberikan suatu angka representasi internal yang unik (alasan efisiensi), co:/ PERIKSA :  
IF A>B THEN X=Y

### (3) Syntax Analyzer / Parser

- ☐ Lebih kompleks dari scanner
  - ☐ Fungsinya untuk menentukan bagaimana dekomposisinya (hasil pecahannya)
  - ☐ Analoginya penentuan struktur dari suatu kalimat dalam bahasa inggris co:/ identifikasi klas subject, predikat, verb, dan adjectiv.
  - ☐ Pengelompokan token dalam kelas sintaksis seperti expression, statement, dan procedure.
  - ☐ Berupa pohon sintaks
  - ☐ Grammer digunakan oleh syntax analyzer untuk menentukan struktur program sumber. (pengenalannya disebut parsing)
- 
- The background of the slide features a light blue gradient with a pattern of binary code (0s and 1s) in a slightly darker shade. Several 3D cubes are floating in the space, each with a blue circular logo on one of its faces. The cubes are positioned at various angles and depths, creating a sense of three-dimensional movement.



## (4) Semantic Analyzer

- ❑ Fungsinya untuk menentukan “maksud” dari suatu program sumber.
- ❑ Secara konseptual sintaks program sumber dipisahkan dari semantiknya tetapi bekerja sama secara dekat.
- ❑ Co:/ untuk ekspresi  $(A+B)*(C+D)$ , semantik analyzer harus menentukan aksi apa yg ditentukan oleh operator aritmetika penjumlahan atau pengurangan. Saat parser mengenali simbol + atau -, akan memanggil rutin semantik yang menspesifikasikan aksi yang akan dilakukan.

## (5) Intermediate Form

□ Dihasilkan dari analisis semantik,co:/ ekspresi  $(A+B)*(C+D)$  kode antaranya bisa berupa kuadrupel berikut:

$(+, A, B, T1)$

$(+, C, D, T2)$

$(*, T1, T2, T3)$

Ekspresi infix bisa dirubah ke notasi polish- sufi menjadi  $AB+CD+*$



## (6) Code Generator

- ❑ Menerima keluaran dari semantic analyzer.
- ❑ Bentuk program sumber ditranslasikan ke bahasa assembly atau mesin.
- ❑ Co:/ translasi kuadrupel dengan format 1 alamat menjadi:

LDA A

ADD B

STO T1

LDA C

ADD D

STO T2

LDA T1

MUL T2

STO T3

# (7) Code Optimizer

- ❑ Tujuan prosesnya untuk menghasilkan program objek yang lebih efisien.
- ❑ Optimasi level lokal dari mulai evaluasi ekspresi konstan, pemakaian properti operator tertentu. Co:/ karena operator perkalian bersifat kumulatif, kode assembler sesuai contoh sebelumnya bisa dikurangi menjadi:

LDA A

ADD B

STO T1

LDA C

ADD D

MUL T1

STO T2

- ❑ Berkaitan dengan alokasi register secara optimal
- ❑ Memperkecil Pengulangan



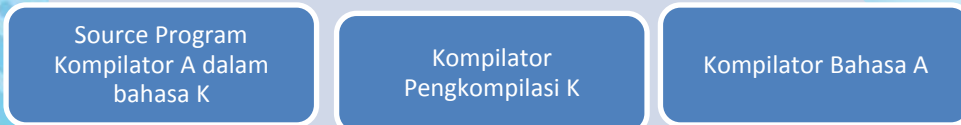
# MUTU KOMPILATOR

Pertimbangan dalam Pemilihan  
Kompilator



# Mutu Kompilator:

1. Kecepatan dan waktu proses kompilasi  
Penulisan algoritma kompilator dan kompilator pengkompilasi (mutu bahasa yg membangun kompilator pengkompilasi)



2. Mutu program objek  
Ditentukan oleh ukuran dan kecepatan eksekusi dari program objek.Co;/ Turbo Pascal 6 lebih baik dari Turbo Pascal 5.
3. Integrated Environment  
Fasilitas terintegrasi yg dimiliki oleh kompilator tersebut.Co;/ editing,debugging dan testing  
Turbo pascal = penyuntingan,kompilasi dan debug  
Clipper = penyuntingan dilakukan oleh prog.editor yang terpisah dari kompilatornya.



# PEMBUATAN KOMPILATOR

Pembuatan Kompilator dapat  
dilakukan dengan:



# 1. Bahasa Mesin

Tingkat kesulitan yang sangat tinggi (sangat tidak manusiawi, ketergantungan terhadap mesin tinggi). penggunaan bahasa mesin dalam pembuatan kompilator hampir mustahil dilakukan. Bahasa mesin mungkin digunakan saat membuat assembler.



## 2. Bahasa Assembly

Bahasa assembly bisa dan biasa digunakan.

Keuntungan penggunaan:

Program hasil (object code) memiliki ukuran yang lebih kecil

Kerugian penggunaan:

Memerlukan usaha yang lebih besar, karena instruksi assembly pendek-pendek (sulit dimengerti) dan memiliki fasilitas yang terbatas. Sulit untuk dikembangkan.

### 3. Bahasa Tingkat Tinggi

- ☐ Proses pembuatan akan lebih mudah, karena menggunakan bahasa tingkat tinggi yang lebih mudah dimengerti.
- ☐ Fasilitasnya lebih baik dan banyak dibandingkan assembly.
- ☐ Kerugiannya, kompilator yang dihasilkan akan memiliki ukuran yang besar.
- ☐ Co:/ pengembangan kompilator untuk bahasa BASIC dengan menggunakan Turbo Pascal.

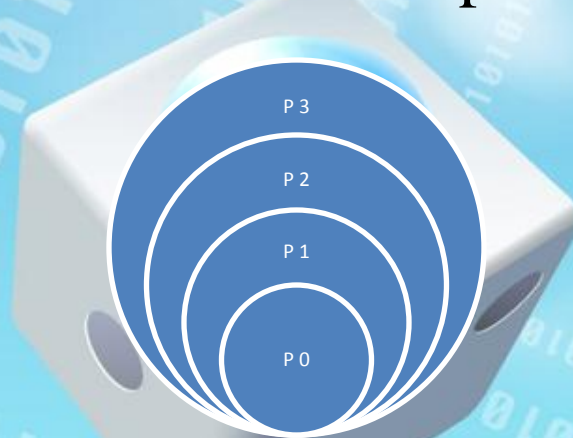


## 4. Bahasa Tingkat Tinggi yang sama pada mesin yang berbeda

- ☐ Keuntungan dan kerugiannya sama dengan no 3.
- ☐ Memerlukan beberapa tahapan proses kompilasi sebelum didapat kompilator yang diinginkan pada mesin tujuan.
- ☐ Co:/ pembuatan kompilator C untuk DOS, berdasarkan kompilator C yang ada di UNIX.

# 5. Bootstrap

- ❑ Gagasan dari Bootstrap adalah kita bisa membangun suatu yang besar dengan lebih dulu membuat intinya.
- ❑ Diperkenalkan oleh Niklaus Wirth pada saat membuat kompilator untuk bahasa pascal.



P 0 dibangun dengan assembly, P1 dibangun oleh P0, P2 dibangun oleh P1 dan P3 dibangun oleh P2. Jadi kompilator bahasa P dibuat tanpa harus secara keseluruhan menggunakan assembly.



# Tugas 1 ( Individu )

1. Mengapa kita memerlukan bahasa pemrograman?
2. Berikan penjelasan dari istilah-istilah berikut:  
Kompilator, translator, interpreter, assembler, emulator.
3. Berikan beberapa contoh produk yang ada di pasaran untuk setiap istilah pada no 2!
4. Lakukan perbandingan komponen mutu dari beberapa kompilator yaitu Turbo pascal, Quick basic, Turbo basic, Turbo C, Microsoft C!
5. Apa keuntungan dari penggunaan metode Bootstrap?

# Ketentuan Tugas 1:

- ☐ Dikumpulkan pada pertemuan berikutnya Selasa 26 Maret 2013 sesuai jadwal mata
- ☐ Format pengerjaan ditulis tangan dengan mencantumkan Nama Mhs dan NIM
- ☐ Pengerjaan Tugas disertai dengan Daftar Pustaka

Selamat Mengerjakan