

3) Syntax Analyzer / parser

- \* Lebih kompleks dari scanner
- \* Berupa pohon syntax
- \* Grammar digunakan oleh syntax analyzer untuk menentukan struktur program sumber.

4) Semantic Analyzer

- \* Fungsinya untuk menentukan "makna" dari suatu program sumber.

5) Intermediate form

- \* Dihasilkan dari analisis semantik. CO... / ekspresi
- $(A+B) * (C+D)$

6) Code Generator

- \* Menerima keluaran dari semantic analyzer
- \* Bentuk program sumber ditranslasikan ke bahasa assembly atau mesin.

7) Code Optimizer

- \* Tujuan prosesnya untuk menghasilkan program objek yang lebih efisien
- \* Berkaitan dengan alokasi register secara optimal
- \* Memperkecil pengulangan.

## Mutu Compiler

- 1) Kecepatan dan waktu proses kompilasi
- 2) Mutu Program objek
- 3) Integrated environment.

## Pembuatan Compiler

Penibahan dengan menggunakan

### 1. Bahasa Mesin

Tingkat kesulitan yang sangat tinggi, ketergantungnya terhadap mesin target. Pembuatan bahasa mesin dalam pembuatan komputer



## 2). Bahasa Assembly

Keuntungan penggunaan :

Program hasil, memiliki ukuran yang lebih kecil

Kekurangan penggunaan :

Memerlukan usaha yang lebih besar, karena instruksi assembly pendek-pendek dan memiliki fasilitas yg terbatas.

## 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.

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

- \* Keuntungan dan kerugiannya sama dengan no. 3.

- \* Memerlukan banyak tahapan proses kompilasi sebelum dapat kompilator yg diinginkan.

## 5). Bootstrap

- \* Gagasan dr bootstrap adalah kita bisa membangun suatu yg besar dg lebih dulu membuat intinya.

- \* Diperkenalkan oleh Niklaus Wirth pelopor membuat kompilator untuk bahasa pascal.