

# Mengenal Compiler

# Bahasa pemrograman

- Merupakan notasi untuk mendeskripsikan komputasi kepada orang dan mesin

Compiler Principles, Techniques, & Tools (2<sup>nd</sup> edition), hal. 1

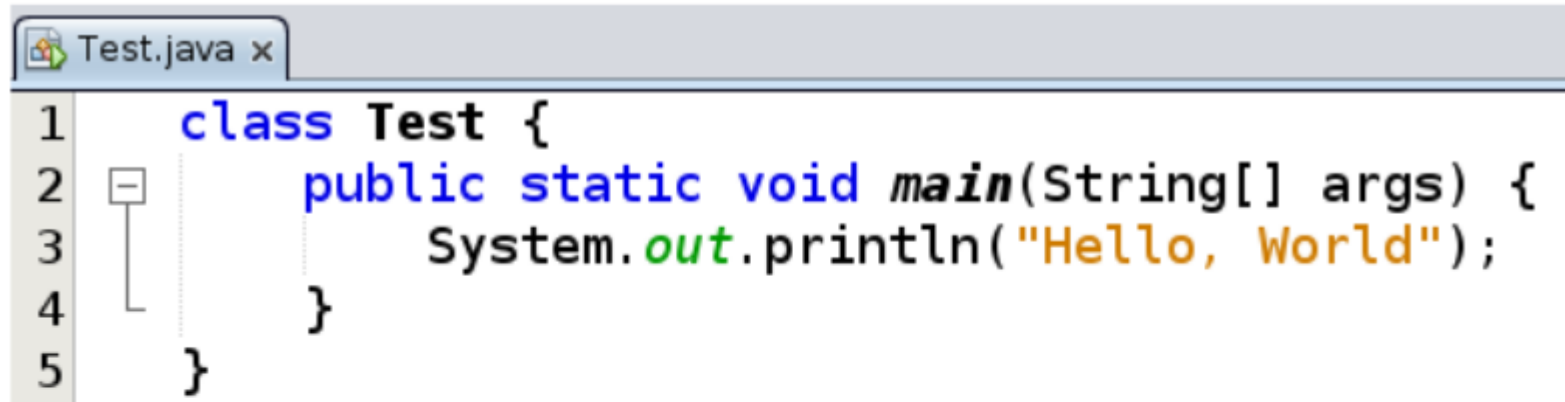
# Bahasa pemrograman

- Saat ini, ada ribuan bahasa pemrograman \*
  - Puluhan diantaranya umum digunakan \*\*

\* Compiler Principles, Techniques, & Tools (2<sup>nd</sup> edition), hal. 13

\*\* Merujuk pada indeks TIOBE

# Contoh program



```
Test.java x
1  class Test {
2      public static void main(String[] args) {
3          System.out.println("Hello, World");
4      }
5  }
```



```
javac Test.java
```



```
java Test
Hello, World
```

Mengenal dan Menggunakan Bahasa Pemrograman Java®, hal. 23

# Contoh program

## Test.class

```
00000000 feca beba 0000 3400 1d00 000a 0006 090f file Test.class
00000010 1000 1100 0008 0a12 1300 1400 0007 0715 Test.class: compiled Java class data, version 52.0 (Java 1.8)
00000020 1600 0001 3c06 6e69 7469 013e 0300 2928
00000030 0156 0400 6f43 6564 0001 4c0f 6e69 4e65
00000040 6d75 6562 5472 6261 656c 0001 6d04 6961
00000050 016e 1600 5b28 6a4c 7661 2f61 616c 676e
00000060 532f 7274 6e69 3b67 5629 0001 530a 756f
00000070 6372 4665 6c69 0165 0900 6554 7473 6a2e
00000080 7661 0c61 0700 0800 0007 0c17 1800 1900
00000090 0001 480c 6c65 6f6c 202c 6f57 6c72 0764
000000a0 1a00 000c 001b 011c 0400 6554 7473 0001
000000b0 6a10 7661 2f61 616c 676e 4f2f 6a62 6365
000000c0 0174 1000 616a 6176 6c2f 6e61 2f67 7953
000000d0 7473 6d65 0001 6f03 7475 0001 4c15 616a
000000e0 6176 692f 2f6f 7250 6e69 5374 7274 6165
000000f0 3b6d 0001 6a13 7661 2f61 6f69 502f 6972
00001000 746e 7453 6572 6d61 0001 7007 6972 746e
00001100 6e6c 0001 2815 6a4c 7661 2f61 616c 676e
00001200 532f 7274 6e69 3b67 5629 2000 0500 0600
00001300 0000 0000 0200 0000 0700 0800 0100 0900
00001400 0000 1d00 0100 0100 0000 0500 b72a 0100
00001500 00b1 0000 0001 000a 0000 0006 0001 0000
00001600 0001 0009 000b 000c 0001 0009 0000 0025
00001700 0002 0001 0000 b209 0200 0312 00b6 b104
00001800 0000 0100 0a00 0000 0a00 0200 0000 0300
00001900 0800 0400 0100 0d00 0000 0200 0e00
000019e
```

Mengenal dan Menggunakan Bahasa Pemrograman Java®, hal. 24

# Kompilasi: sisi programmer

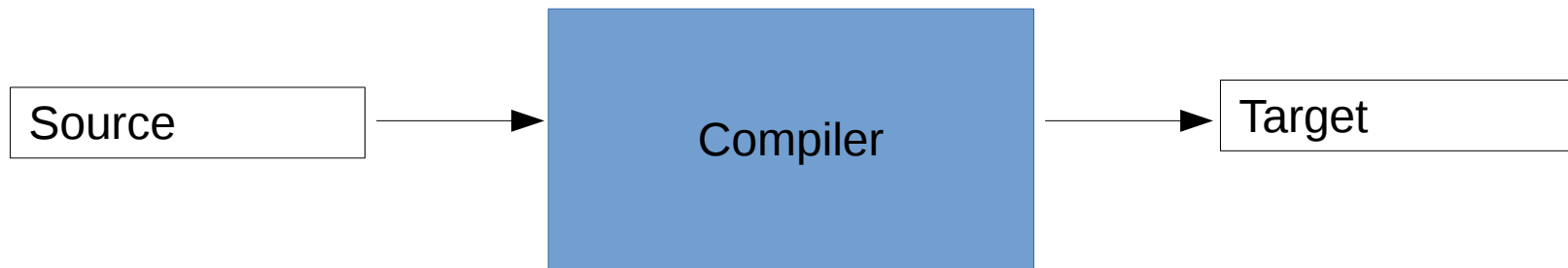
- Kita berikan sebuah source code
- Apabila tidak terdapat kesalahan:
  - Sesuatu akan dihasilkan (target)
- Apabila terdapat kesalahan:
  - Pesan kesalahan akan ditampilkan
  - Umumnya dilengkapi detail kesalahan, dan di mana kesalahan tersebut ditemukan

# Kompilasi: dari sisi compiler

- Compiler: baca dari satu bahasa (source), terjemahkan ke bahasa lain (target)
- Apapun kesalahan yang terdeteksi perlu dilaporkan

Compiler Principles, Techniques, & Tools (2<sup>nd</sup> edition), hal. 1

# Kompilasi: dari sisi compiler





# Target

- Native code
- Byte code (Virtual machine)
- Bahasa pemrograman lain
- Dan lainnya

# Compiler: pekerja keras

- Sejumlah tahapan perlu dilalui
- Sebagai bonus, compiler yang hebat mungkin akan:
  - Melaporkan potensi permasalahan dari source yang diberikan
  - Melakukan optimisasi (terlepas dari bagaimana programmer menuliskan source code nya)

# Tahapan kompilasi (1)

- Lexical analysis / scanning
  - Baca karakter-karakter dalam source code
  - Kelompokkan dalam lexeme (deretan karakter yang memiliki arti)
  - Menghasilkan token

Compiler Principles, Techniques, & Tools (2<sup>nd</sup> edition), hal. 5

# Tahapan kompilasi (2)

- Syntax analysis (parsing)
  - Token yang dihasilkan dari lexical analyzer akan digunakan
  - Sebuah representasi (tree) akan dibuat, sesuai aturan bahasa
  - Terdapat berbagai teknik parsing

Compiler Principles, Techniques, & Tools (2<sup>nd</sup> edition), hal. 8

# Tahapan kompilasi (3)

- Semantic analysis
  - Tree yang dihasilkan dari syntax analyzer akan digunakan
  - Apakah konsisten secara semantik?
  - Barangkali pemeriksaan tipe – sampai level tertentu – akan dilakukan

Compiler Principles, Techniques, & Tools (2<sup>nd</sup> edition), hal. 8

# Tahapan kompilasi (4)

- Intermediate code generation
  - Intermediate representation: bisa dalam berbagai format
  - Alangkah baiknya:
    - Mudah dihasilkan
    - Mudah diterjemahkan ke target
      - Terutama apabila compiler dapat menghasilkan beberapa jenis target

Compiler Principles, Techniques, & Tools (2<sup>nd</sup> edition), hal. 9

# Tahapan kompilasi (5)

- Code optimization
  - Dari intermediate representation, compiler berusaha agar dapat menghasilkan kode yang lebih optimum
  - Terdapat sejumlah teknik
  - Optimizing compiler → keren

Compiler Principles, Techniques, & Tools (2<sup>nd</sup> edition), hal. 10

# Tahapan kompilasi (6)

- Code generation
  - Intermediate representation akan digunakan dalam menghasilkan kode target
    - Native code
    - Byte code (virtual machine)
    - Bahasa pemrograman lain
    - ...
  - Optimisasi bisa dilakukan lagi apabila dimungkinkan

Compiler Principles, Techniques, & Tools (2<sup>nd</sup> edition), hal. 10



# Compiler yang Anda gunakan

- Apa bahasa pemrograman yang Anda gunakan saat ini?
- Sangat umum terdapat beberapa (bahkan banyak) compiler untuk sebuah bahasa pemrograman
- Mari kita contohkan beberapa compiler untuk sebuah bahasa pemrograman

# Terima kasih

- Diskusi