

BAB II

LANDASAN TEORI

2.1 Sejarah Java

Java bermula dari proyek penelitian perusahaan *Sun Microsystems* dengan nama sandi *Green* pada tahun 1991. Terdapat prediksi bahwa mikroprosesor akan digunakan luas pada peralatan-peralatan elektronik. Karena adanya bermacam tipe mikroprosesor, maka dibutuhkan sebuah bahasa pemrograman yang dapat berjalan di semua mikroprosesor.

Terciptalah sebuah bahasa pemrograman baru. Oleh James Gosling, yaitu salah satu orang yang berperan besar dalam proyek tersebut, program ini diberi nama Oak. Sesuai dengan pohon Oak yang tumbuh dan bisa dilihat melalui jendela kerjanya di *Sun Microsystems*.

Selang beberapa waktu kemudian, ditemukan bahwa sudah ada bahasa pemrograman dengan nama Oak. Akhirnya setelah beberapa pegawai *Sun* mengunjungi sebuah kedai kopi nama bahasa pemrograman ini diganti dengan Java. Java merupakan salah satu jenis biji kopi yang ada di kedai tersebut, yaitu biji kopi Jawa.

Sun Microsystems mengumumkan kehadiran bahasa Java secara formal di tahun 1995. Bahasa ini mulai disambut hangat masyarakat luas seiring dengan meledaknya era internet. (Rachmad Hakim S. & Sutarto, 2009, p.1).

2.2 Keunggulan Java

Rachmad Hakim S., dan Sutarto (2009) menyatakan bahwa, dibanding bahasa pemrograman lain, Java mempunyai beberapa kelebihan yang membuatnya dipakai luas terutama untuk aplikasi enterprise. Java bahkan diprediksi menjadi bahasa masa depan. Java telah banyak diajarkan di lingkungan pendidikan dan dipakai secara luas oleh perusahaan dan lembaga berskala internasional, seperti IBM dan NASA.

Beberapa keunggulan Java, antara lain :

2.2.1 Relatif Mudah Dipelajari

Menurut Rachmad Hakim S., dan Sutarto (2009) Java merupakan bahasa pemrograman bahasa pemrograman berorientasi objek (*Object Oriented Programming*) yang mudah dipelajari, terutama bagi yang sudah familiar dengan bahasa C, C++, atau C#.

Java meningkatkan pemrograman objek pada C++ dengan menghilangkan pointer pada tipe data biasa dan *multiple inheritance*. Java juga menggunakan pengalokasian memori secara otomatis termasuk menerapkan garbage collection.

2.2.2 Mudah Dikembangkan

Menurut Rachmad Hakim S., dan Sutarto (2009) dengan pemrograman berorientasi objek, komponen program Java terpecah menjadi satu atau beberapa *class*. Pengembangan bahasa Java sangat mudah karena tinggal mengubah atau menambah *class* yang pernah dibuat. Selain itu, kita juga dapat menggunakan *class-class* yang sudah ada dalam Java API ataupun dibuat oleh pihak ketiga.

2.2.3 Platform Independent

Menurut Rachmad Hakim S., dan Sutarto (2009) dengan semboyan *write one, run everywhere*, Anda dapat menggunakan program Java pada beberapa mesin yang berbeda. Mulai dari PDA, handphone, komputer,

hingga server. Dengan mesin hardware yang berbeda dan sistem operasi berbeda, Java tetap dapat digunakan. Asalkan sudah mendukung java platform. Misalnya, program Java dapat dijalankan pada PC Windows, PC Linux, dan Apple Mac OS (p.6).

2.3 Komponen Java

Nofriadi (2015) mengatakan, Java memiliki dua komponen, yaitu:

2.3.1 JVM (*Java Virtual Machine*)

Supaya bahasa pemrograman Java dapat berjalan di sebuah sistem operasi, maka suatu komponen yang nama *Java Virtual Machine* (JVM). JVM juga mempunyai dua buah pembagian, yaitu *Java Runtime Environment* (JRE) dan *Java Development Kit* (JDK).

2.3.1.1 JRE (*Java Runtime Environment*)

Java Runtime Environment atau kita sering menyebutnya Java merupakan yang digunakan untuk menjalankan aplikasi Windows yang dibuat dengan platform Java sangat berbeda dengan sebuah software yang dibuat dengan bahasa lain, seperti *Visual Basic*. Pasti masih banyak yang belum tahu fungsi dari JRE ini. Fungsi dari JRE yaitu untuk menjalankan aplikasi yang biasa disebut dengan Applets yang dibuat dengan menggunakan bahasa pemrograman Java. Intinya, *software* yang dibuat dengan bahasa pemrograman Java, wajib memasang JRE. Jika tidak, secara mutlak aplikasi tersebut tidak bisa berjalan.

2.3.1.2 JDK (*Java Development Kit*)

Java Development Kit berisi sekumpulan *tools* baris perintah (*coman line tool*) untuk menciptakan program Java. Di dalam JDK terdapat beberapa komponen utama, antara lain:

1. Compiler

Kompilasi Java yang menghasilkan sebuah fail dengan extensi.class.

2. Interpreter

Berfungsi untuk menjalankan program bersifat *bytecode*. Interpreter ini bertindak sebagai pembantu baris perintah untuk menjalankan program Java non-grafis.

3. Applet

Berfungsi untuk menguji Java Applet secara minimal.

4. Debugger

Berfungsi untuk mengetahui debug program di saat di-*running*.

5. Class Fail Disassembler

Berfungsi untuk melakukan penguasaan terhadap *class*.

6. Header and Subfile Generator

Untuk menghasilkan header dan fail kode sumber untuk implementasi *method*.

7. Java Documentation Generating

Merupakan alat bantu untuk menghasilkan dokumentasi API secara langsung dari kode sumber Java.

2.3.2 IDE (*Integrated Development Environment*)

IDE merupakan sebuah teks editor untuk penulisan script bahasa pemrograman Java. Ada beberapa teks editor yang bisa digunakan, antara lain: Notepad, Jcreator, Netbeans dan Eclips (p.1-3).

2.4 Flowchart

Menurut Prof. Dr. Nizwardi Jalinus, M. Ed., & Dr. Ambiyar, M. Pd. (2016) dia mengatakan, *Flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma.

Ada dua macam *flowchart* yang menggambarkan proses dengan komputer, yaitu:

1. *System Flowchart*

Menggambarkan suatu system peralatan computer yang digunakan dalam proses pengolahan data serta hubungan antar peralatan tersebut.

2. *Program Flowchart*

Menggambarkan urutan logika dari suatu prosedur pemecahan masalah. Ada dua jenis program chart, yaitu: (1) *conceptual flowchart*, dan (2) *detail flowchart*.

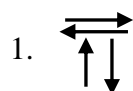
2.4.1 Simbol - Simbol Flowchart

Flowchart disusun dengan simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Simbol-simbol yang digunakan yaitu :


1. *Flow Direction Symbols* (Simbol penghubung/alur)
2. *Processing Symbols* (Simbol proses)
3. *Input-output Symbols* (Simbol input-output).


2.4.1.1 Flow Direction Symbols


Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga *connecting line*. Simbol-simbol tersebut adalah :



Simbol arus / *flow*
Untuk menyatakan jalannya arus suatu proses.

2. 


Simbol *Communication link*
 Untuk menyatakan bahwa adanya transisi suatu data/ informasi dari satu lokasi ke lokasi lainnya.
3. 


Simbol *Connector*
 Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang sama.
4. 

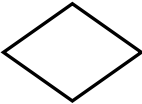
Simbol *Offline Connector*
 Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.


2.4.1.2 Processing Symbols


Simbol yang menunjukan jenis operasi pengolahan dalam suatu proses / prosedur. Simbol-simbol tersebut adalah :

1. 


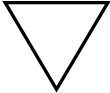

Simbol *Process*
 Untuk menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
2. 

Simbol *Manual*
 Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer (manual).
3. 

Simbol *Decision / logika*
 Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer (manual).
4. 

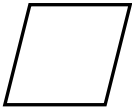




Simbol *Predefined Proses*
 Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
5. 

Simbol *Terminal*
 Untuk menyatakan permulaan atau akhir satu program.

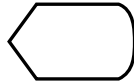
- | | | |
|----|---|--|
| 6. |  | <p>Simbol <i>Keying Operation</i></p> <p>Untuk menyatakan segala jenis proses yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard</p> |
| 7. |  | <p>Simbol <i>Offline Storage</i></p> <p>Untuk menunjukan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.</p> |
| 8. |  | <p>Simbol <i>Manual Input</i></p> <p>Untuk memasukan data secara manual dengan menggunakan <i>online</i> keyboard.</p> |

2.4.1.3 Input-Output Symbols

Simbol yang menunjukan jenis peralatan yang digunakan sebagai media input atau output. Simbol-simbol tersebut adalah :

- | | | |
|----|---|--|
| 1. |  | <p>Simbol <i>Input-Output</i></p> <p>Untuk menyatakan proses input dan output tanpa tergantung dengan jenis peralatanya.</p> |
| 2. |  | <p>Simbol <i>Puched Card</i></p> <p>Untuk menyatakan input berasal dari suatu kartu atau output ditulis ke kartu.</p> |
| 3. |  | <p>Simbol <i>Magnetic-tape unit</i></p> <p>Untuk menyatakan input berasal dari pita magnetic atau output disimpan ke pita <i>magnetic</i>.</p> |
| 4. |  | <p>Simbol <i>Disk storage</i></p> <p>Untuk menyatakan input berasal dari disk atau output disimpan ke disk.</p> |
| 5. |  | <p>Simbol <i>Document</i></p> <p>Untuk mencetak laporan ke printer.</p> |

6.

Simbol *Display*

Untuk menyatakan peralatan input yang digunakan berupa layar (video, komputer).

Dari uraian diatas terlihat bahwa simbol *flowchart* dibagi kedalam beberapa kelompok, masing-masing kelompok mempunyai fungsi tersendiri dan penggunaanya sesuai kebutuhan.

2.5 NetBeans

NetBeans merupakan sebuah aplikasi *Integrated Development Environment* (IDE) yang berbasiskan Java dari Sun Microsystems yang berjalan di atas *swing* dan banyak digunakan sekarang sebagai editor untuk berbagai bahasa pemrograman. Pada Netbeans, kita bisa membuat bahasa pemrograman Java, JavaScript, PHP, Python, Groovy, C, C++, Scala, Clojure. *Swing* merupakan teknologi Java untuk pengembangan aplikasi desktop yang bisa dijalankan di berbagai sistem operasi, seperti Windows, Linux, Mac OS X dan Solaris (Nofriadi, 2015, p.4).

2.6 Bangun Datar

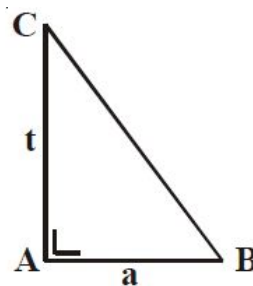
1. Pengertian Bangun Datar

Bangun datar dapat didefinisikan sebagai bangun yang rata yang mempunyai dua dimensi yaitu panjang dan lebar tetapi tidak mempunyai tinggi dan tebal. Dalam kehidupan sehari-hari mengambil contoh bangun datar tidaklah mudah. Misalkan saja kita ambil selembar kertas Houtvrij Schrijfpapier (HVS) atau kertas Koran sebagai bangun datar. Kalau benar-benar diperiksa, kertas itu selain mempunyai panjang dan lebar juga kertas itu mempunyai tebal ataupun tinggi. Dengan alat ukur yang mempunyai ketelitian yang tinggi tebal kertas dapat diukur. Benda-benda dilihat dengan mata telanjang terlihat rata atau datar belum tentu memenuhi syarat untuk digolongkan sebagai bangun datar. Dengan demikian pengertian bangun datar adalah abstrak.

2. Jenis-Jenis Bangun Datar

Pelajaran matematika materi sifat-sifat bangun datar kelas 5 semester 2 tercantum dalam Kurikulum Tingkat Satuan Pendidikan (KTSP). Adapun dalam penelitian ini, peneliti memilih materi tentang sifat-sifat bangun datar. Menurut Soenarjo (2008: 226) sifat-sifat bangun datar dipaparkan sebagai berikut:

1. Segitiga



Gambar 1. Segitiga

Segitiga adalah bangun datar yang memiliki tiga sisi dan tiga titik sudut.

Keterangan dari gambar tersebut adalah:

- 1) a adalah Alas.
- 2) t adalah Tinggi.
- 3) AB,BC,CA adalah sisi segitiga.

Sifat-sifat umum segitiga, sebagai berikut:

- 1) Mempunyai 3 sisi.
- 2) Mempunyai 3 sudut.
- 3) Jumlah sudut segitiga 180^0 .

Rumus mencari segitiga, sebagai berikut:

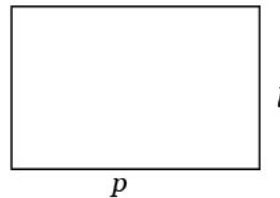
- 1) Menghitung Luas Segitiga

$$L = \frac{1}{2} \times a \times t$$

2) Menghitung Keliling Segitiga

$$K = AB + BC + CA$$

2. Persegi Panjang



Gambar 2. Persegi Panjang

Keterangan dari gambar tersebut adalah:

- 1) p adalah Panjang.
- 2) l adalah Lebar.

Sifat-sifat persegi panjang sebagai berikut:

- 1) Semua sifat umum segiempat.
- 2) Mempunyai 2 pasang sisi berhadapan sejajar dan sama panjang.
- 3) Ke empat sudutnya berbentuk sudut siku-siku.

Rumus mencari persegi panjang, sebagai berikut:

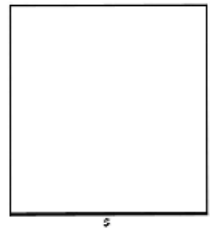
1) Menghitung Luas Persegi Panjang

$$L = p \times l$$

2) Menghitung Keliling Persegi Panjang

$$K = 2 \times (p + l)$$

3. Persegi



Gambar 3. Persegi

Keterangan dari gambar tersebut adalah:

- 1) s adalah Sisi.

Sifat-sifat persegi sebagai berikut:

- 1) Semua sifat umum segiempat.
- 2) Keempat sisinya sama panjang.
- 3) Keempat sudutnya berbentuk sudut siku-siku.

Rumus mencari segitiga, sebagai berikut:

- 1) Menghitung Luas Persegi

$$L = s \times s$$

- 2) Menghitung Keliling Persegi

$$K = 4 \times s$$

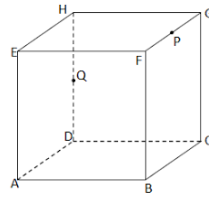
2.7 Bangun Ruang

1. Pengertian Bangun Ruang

Bangun Ruang adalah suatu bangun tiga dimensi yang memiliki volume atau isi.

2. Jenis-Jenis Bangun Ruang

1. Kubus



Gambar 1. Kubus

Kubus adalah bangun ruang yang semua sisinya berbentuk persegi dan memiliki rusuk-rusuk yang sama Panjang.

Keterangan dari gambar tersebut adalah:

- 1) EABF adalah Permukaan sisi kubus bagian depan.
- 2) HDCG adalah Permukaan sisi kubus bagian belakang.
- 3) EFGH adalah Permukaan sisi kubus bagian atas.
- 4) ABCD adalah Permukaan sisi kubus bagian bawah.
- 5) EHDA adalah Permukaan sisi kubus bagian kiri.
- 6) FGBC adalah Permukaan sisi balok bagian kanan.

Sifat-sifat umum Kubus, sebagai berikut:

- 1) Mempunyai 8 titik sudut.
- 2) Mempunyai 6 sisi.
- 3) Mempunyai 12 rusuk.

Rumus mencari kubus, sebagai berikut:

- 1) Menghitung Luas Kubus

$$L = 6 \times s^2$$

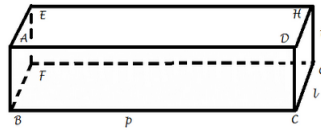
- 2) Menghitung Keliling Kubus

$$K = 12 \times s$$

- 3) Menghitung Volume Kubus

$$V = s \times s \times s$$

2. Balok



Gambar 2. Balok

Balok adalah bangun ruang yang semua sisinya berbentuk persegi Panjang.

Keterangan dari gambar tersebut adalah:

- 1) p adalah Panjang.
- 2) l adalah Lebar.
- 3) t adalah Tinggi.
- 4) ABCD adalah Permukaan sisi balok bagian depan.
- 5) EFGH adalah Permukaan sisi balok bagian belakang.
- 6) AEHD adalah Permukaan sisi balok bagian atas.
- 7) BFGC adalah Permukaan sisi balok bagian bawah.
- 8) ABFE adalah Permukaan sisi balok ujung kiri.
- 9) DCGH adalah Permukaan sisi balok ujung kanan.

Sifat-sifat umum balok, sebagai berikut:

- 1) Mempunyai 8 titik sudut.
- 2) Mempunyai 6 sisi.
- 3) Mempunyai 12 rusuk.

Rumus mencari balok, sebagai berikut:

- 1) Menghitung Luas Balok

$$L = 2 (pl + pt + lt)$$

- 2) Menghitung Keliling Balok

$$K = 4 (p + l + t)$$

3) Menghitung Volume Balok

$$V = p \times l \times t$$