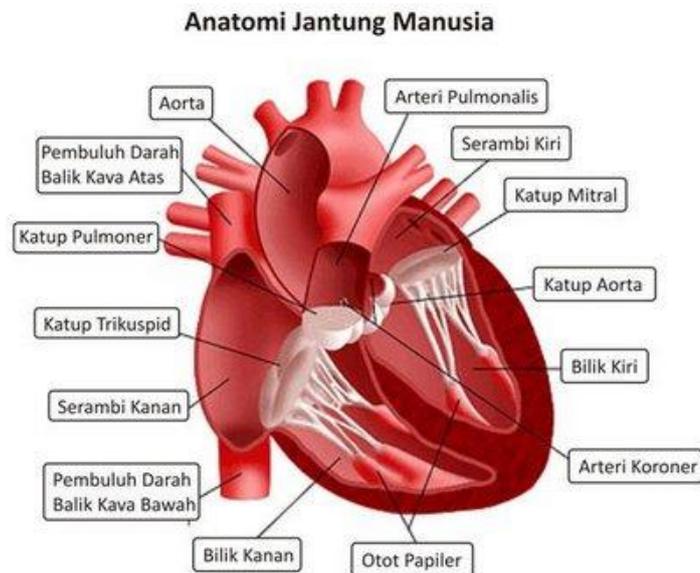


BAB II TINJAUAN PUSTAKA

2.1 Jantung

Jantung merupakan organ yang paling penting dalam tubuh manusia. Tanpa henti memompa oksigen dan nutrisi melalui darah ke seluruh tubuh. Jantung kita berdetak 100ribu kali per hari atau memompa sekitar 2000 galon per hari. Ketika berdetak, jantung memompa darah melalui pembuluh – pembuluh darah ke seluruh tubuh. pembuluh darah ini sangat elastic dan bisa membawa darah ke setiap ujung organ tubuh kita. Darah sangat penting karena berfungsi untuk mengangkat oksigen dari paru-paru dan nutrisi ke setiap jaringan tubuh, juga membawa sisa-sisa seperti karbondioksida keluar dari jaringan-jaringan tubuh. (sumber: <http://id.wikipedia.org/wiki/Jantung>)



Gambar 2.1 Anatomi Jantung Manusia

(sumber : <https://marsenorhudy.files.wordpress.com/2010/10/anatomi-jantung.jpg>)

Ada tiga tipe pembuluh darah :

1. Pembuluh darah arteri : berfungsi mengangkat oksigen melalui darah dari jantung ke seluruh jaringan tubuh, akan semakin mengecil ketika darah melewati pembuluh menuju organ lainnya.

2. Pembuluh kapiler : bentuknya kecil dan tipis, menghubungkan pembuluh arteri dan pembuluh vena. Lapisan dindingnya yang tipis memudahkan untuk dilewati oleh oksigen, nutrisi, karbondioksida serta bahan sisa lainnya dan ke organ sel lainnya.
3. Pembuluh vena : fungsinya menyalurkan aliran darah yang berisi bahan sisa kembali ke jantung untuk dipecahkan dan dikeluarkan dari tubuh. pembuluh vena semakin membesar ketika mendekati jantung. Bagian atas vena (superior) membawa darah dari tangan dan kepala menuju jantung. Sedangkan bagian bawah (inferior) membawa darah dari bagian perut dan kaki menuju jantung. (sumber: <http://www.sridianti.com/fungsi-pembuluh-darah.html>)

2.2 Sejarah Operating Sistem *Android*

Android Merupakan sebuah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak.

2.2.1 *Android* Versi Awal (2007 – 2008)

Pada September 2007 Google mengajukan hak paten aplikasi telepon seluler. Google mengenalkan Nexus One, salah satu jenis telepon pintar GSM yang menggunakan *Android* pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja *Android* ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, *Android*, perangkat bergerak (mobile) yang merupakan modifikasi kernel Linux 2.6. Sejak *Android* dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru. Smartphone yang memakai sistem operasi *Android* adalah HTC Dream, yang

dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan *Android*.

2.2.2 *Android* versi 1.5 (Cupcake)



Gambar 2.2. Android Versi 1.5

(sumber: <http://www.swalt.info/os/Android/83-sistem-operasi-Android.html>
diakses pada 17 April 2015)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan *Android* dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

2.2.3 *Android* versi 1.6 (Donut)



Gambar 2.3 Android Versi 1.6

(sumber: <http://www.swalt.info/os/Android/83-sistem-operasi-Android.html>,
diakses pada 17 April 2015)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi text to change speech (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

2.2.4 *Android* versi 2.0 / 2.1 (Éclair)



Gambar 2.4. *Android* Versi 2.0/2.1

(sumber: <http://www.swalt.info/os/Android/83-sistem-operasi-Android.html>, diakses pada 17 April 2015)

Pada 3 Desember 2009 kembali diluncurkan ponsel *Android* dengan versi 2.0/2.1 (Éclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1. Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (killer apps - aplikasi unggulan). Kompetisi ini berhadiah \$25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik. Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset *Android*, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi *Android*. Aplikasi terkenal yang diubah ke dalam

sistem operasi *Android* adalah *Shazam*, *Backgrounds*, dan *WeatherBug*. Sistem operasi *Android* dalam situs Internet juga dianggap penting untuk menciptakan aplikasi *Android* asli, contohnya oleh *MySpace* dan *Facebook*.

2.2.5 *Android* versi 2.2 (*Froyo* : *Frozen Yoghurt*)



Gambar 2.5. *Android* Versi 2.2

(sumber: <http://www.swalt.info/os/Android/83-sistem-operasi-Android.html>, diakses pada 17 April 2015)

Pada 20 Mei 2010, *Android* versi 2.2 (*Froyo*) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 *JavaScript engine* yang dipakai *Google Chrome* yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi *Android Market*.

2.2.6 *Android* versi 2.3 (*Gingerbread*)



Gambar 2.6. *Android* Versi 2.3

(sumber: <http://www.swalt.info/os/Android/83-sistem-operasi-Android.html>, diakses pada 17 April 2015)

Pada 6 Desember 2010, *Android* versi 2.3 (*Gingerbread*) diluncurkan. Perubahan-perubahan umum yang didapat dari *Android* versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

2.2.7 *Android* versi 3.0 / 3.1 (*Honeycomb*)



Gambar 2.7. *Android* Versi 3.0/3.1

(sumber: <http://www.swalt.info/os/Android/83-sistem-operasi-Android.html>, diakses pada 17 April 2015)

Android Honeycomb dirancang khusus untuk tablet. *Android* versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom. Perangkat tablet dengan platform *Android* 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.

2.2.8 *Android versi 4.0 (Ice Cream)*



Gambar 2.8. Android Versi 4.0

(sumber: <http://www.swalt.info/os/Android/83-sistem-operasi-Android.html>, diakses pada 17 April 2015)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC. Ponsel pertama yang menggunakan sistem operasi ini adalah Samsung Galaxy Nexus.

2.2.9 *Android versi 4.1 (Jelly Bean)*



Gambar 2.9. Android Versi 4.1

(sumber: <http://www.swalt.info/os/Android/83-sistem-operasi-Android.html>, diakses pada 17 April 2015)

Android Jelly Bean yang diluncurkan pada acara Google I/O lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui Voice Search yang lebih cepat. Tak ketinggalan Google Now

juga menjadi bagian yang diperbarui. Google Now memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi *Android Jelly Bean 4.1* muncul pertama kali dalam produk tablet Asus, yakni Google Nexus 7. Fitur photo sphere untuk panorama, daydream sebagai screensaver, power control, lock screen widget, menjalankan banyak user (dalam tablet saja), widget terbaru. *Android 4.2* Pertama kali dikenalkan melalui LG Google Nexus 4

2.2.10 *Android* Versi 4.4 (*KitKat*)



Gambar 2.10. *Android* Versi 4.4 (*KitKat*)

(sumber: <http://dito.blog.uns.ac.id/2014/08/31/macam-Android-dari-dulu-hingga-sekarang/> diakses pada 17 April 2015)

Kehadiran *Android* kitkat merupakan peluncuran produk OS anyar yang diluncurkan pada 4 september 2013, sebelumnya banyak kabar beredar jikalau *Android* akan meluncurkan OS baru yang bernama *Android Key Lime Pie* namun setelah di analalisa tidak sesuai dengan ejaan orang umum, sehingga namanya diganti dengan OS *Android KitKat* yang sebagian besar orang sudah familiar dengan itu seperti yang dilangsing BBC dalam wawancaranya dengan John Lagerling selaku perwakilan dari google.

2.2.11 *Android Versi Lollipop*



Gambar 2.11. *Android Versi Lollipop*

(sumber: <http://dito.blog.uns.ac.id/2014/08/31/macam-Android-dari-dulu-hingga-sekarang/> diakses pada 17 April 2015)

Di *Android* versi L yang katanya merupakan ‘Lollipop’, Google lebih memfokuskan pada sisi desain dan performa. Dari segi performa, *Android* L sudah mensupport prosesor 64 bit dan juga *Android* L memiliki runtime, grafis dan daya tahan baterai yang lebih baik dibanding *Android* versi sebelumnya.

Sebagai catatan bahwa dalam jangka waktu 3 tahun ini Google hanya menerbitkan versi *Android* baru dengan perubahan kecil saja, yaitu pada *Android* Ice Cream Sandwich hingga *Android* Kitkat. Sedangkan pada *Android* L ini Google akan melakukan perubahan besar yang di dalamnya terdapat berbagai macam fitur-fitur baru. Kita nantikan saja kemunculan *Android* L ini, yang katanya akan dilaunching pada akhir 2014 ini.

2.3 **Fitur yang tersedia di *Android***

1. **Kerangka aplikasi:** itu memungkinkan penggunaan dan penghapusan komponen yang tersedia.
2. **Dalvik mesin virtual:** mesin virtual dioptimalkan untuk perangkat telepon seluler.
3. **Grafik:** grafik di 2D dan grafis 3D berdasarkan pustaka OpenGL.
4. **SQLite:** untuk penyimpanan data.
5. **Mendukung media:** audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)

6. **GSM, Bluetooth, EDGE, 3G, 4G dan WiFi** (tergantung piranti keras)
7. **Kamera, Global Positioning System (GPS), kompas, NFC dan accelerometer** (tergantung piranti keras)

2.4 LCD (*Liquid Crystal Display*)

Display elektronik adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD (*Liquid Cristal Display*) adalah salah satu jenis display elektronik yang dibuat dengan teknologi CMOS logic yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap front-lit atau mentransmisikan cahaya dari back-lit. LCD (*Liquid Cristal Display*) berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik.



Gambar 2.12. LCD

(sumber: <http://elektronika-dasar.web.id/teori-elektronika/lcd-liquid-cristal-display/> diakses pada 17 April 2015)

Microntroller pada suatu LCD (*Liquid Cristal Display*) dilengkapi dengan memori dan register. Memori yang digunakan microcontroler internal LCD adalah DDRAM (*Display Data Random Access Memory*) merupakan memori tempat karakter yang akan ditampilkan berada. CGRAM (*Character Generator Random Access Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat diubah-ubah sesuai dengan keinginan. CGROM (*Character Generator Read Only Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut merupakan karakter dasar yang sudah ditentukan secara permanen oleh pabrikan pembuat LCD (*Liquid Cristal Display*) tersebut sehingga pengguna tinggal mangambilnya sesuai

alamat memorinya dan tidak dapat merubah karakter dasar yang ada dalam CGROM.

2.5 Mikrokontroller

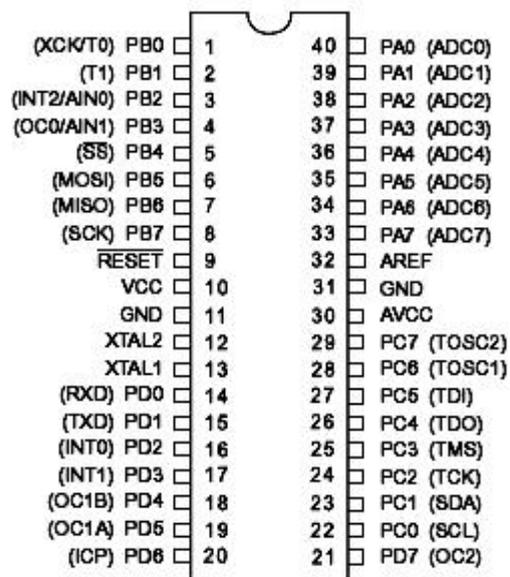
Mikrokontroller adalah sebuah sistem komputer lengkap dalam satu serpih (*chip*). Mikrokontroller lebih dari sekedar sebuah mikroprosesor karena sudah terdapat atau berisikan ROM (*Read-Only Memory*), RAM (*Read-Write Memory*), beberapa Port masukan maupun keluaran, dan beberapa peripheral seperti pencacah/pewaktu, ADC (*Analog to Digital converter*), DAC (*Digital to Analog converter*) dan serial komunikasi. Salah satu mikrokontroller yang banyak digunakan saat ini yaitu mikrokontroller AVR. AVR adalah mikrokontroller RISC (*Reduce Instruction Set Compute*) 8 bit berdasarkan arsitektur *Harvard*. Secara umum mikrokontroller AVR dapat dikelompokkan menjadi 3 kelompok, yaitu keluarga AT90Sxx, ATmega dan ATtiny. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fiturnya. Seperti mikroprosesor pada umumnya, secara internal mikrokontroller ATmega16 terdiri atas unit-unit fungsionalnya Arithmetic and Logical Unit (ALU), himpunan register kerja, register dan dekoder instruksi, dan pewaktu beserta komponen kendali lainnya. Berbeda dengan mikroprosesor, mikrokontroller menyediakan memori dalam serpih yang sama dengan prosesornya (*in chip*). (Radianujinugraha,2008:1)

2.5.1 Mikrokontroller ATmega 16

Pin-pin pada ATmega16 dengan kemasan 40-pin DIP (*dual in-line package*) ditunjukkan oleh gambar 2.13 guna memaksimalkan performa, AVR menggunakan arsitektur *Harvard* (dengan memori dan bus terpisah untuk program dan data). Berikut beberapa keistimewaan ATmega 16 yaitu :

1. Saluran *Input/Output* (I/O) ada 32 buah, yaitu PORTA, PORTB, PORTC, PORTD
2. ADC / *Analog to Digital Converter* 10 bit sebanyak 8 channel pada PORTA

3. 2 buah *timer/counter* 8-bit dan 1 buah *timer/counter* 16-bit dengan *prescalers* dan kemampuan pembandingan
4. *Watchdog timer* dengan osilator internal
5. Tegangan operasi 2,75 - 5,5 V pada ATMega16L dan 4,5 - 5,5 V pada ATMega16
6. EEPROM sebesar 512 byte yang dapat diprogram saat operasi
7. Antarmuka komparator analog
8. 4 *channel* PWM
9. kecepatan nilai (*speed grades*) 0 - 8 MHz untuk ATMega16L dan 0 - 16 MHz untuk ATMega16



Gambar 2.13. ATMega 16

(sumber: <http://ilmukomputer.org/wp-content/uploads/2008/08/sholihul-atmega16.pdf> diakses pada 20 April 2015)

2.6 Basic Compiler

Compiler adalah suatu program yang menerjemahkan bahasa program (source code) kedalam bahasa objek (*obyek code*). *Compiler* menggabungkan keseluruhan bahasa program, mengumpulkannya dan kemudian menyusunnya kembali. *Kompiler* memerlukan waktu untuk membuat suatu program dapat di eksekusi oleh computer, program yang dieksekusi oleh *compiler* adalah dapat

berjalan lebih cepat disbanding program yang diproduksi oleh *interpreter*, disamping itu juga bersifat independen. Contoh program yang menggunakan compiler adalah *Visual Basic, Visual Delvi, dan Pascal*. Tahap Visualisasi yaitu :

1. Pertama *source code* (program yang ditulis) dibaca memori computer).
2. *Source code* tersebut diubah menjadi objek code (bahasa *Assembly*).
3. *Objek code* di hubungkan dengan library yang dibutuhkan untuk membentuk file yang bisa dieksekusi.

(sumber:<http://adiferiismail.blogspot.com/2015/03/pengertian-compiler-interpreter.html> diakses pada 20 April 2015)

2.6.1 Tipe Data

Tipe data merupakan bagian program yang paling penting karena sangat berpengaruh setiap instruksi yang akan dilaksanakan oleh komputer pada program. Pemilihan tipe data yang tepat maka operasi data menjadi lebih efeasian dan efektif. (Iswanto, 2009:31)

Tabel 2.1. Tipe data pada Bascom AVR

(Sumber : Iswanto,2009:31)

No	Tipe	Jangkauan
1	<i>Bit</i>	0 atau 1
2	<i>Byte</i>	0 – 255
3	<i>Integer</i>	-32,768 –32,767
4	<i>Word</i>	0 – 65535
5	<i>Long</i>	-2147483648 – 2147483647
6	<i>Single</i>	$1.5 \times 10^{-45} - 3.4 \times 10^{38}$
7	<i>Double</i>	$5.0 \times 10^{-324} \text{ to } 1.7 \times 10^{308}$
8	<i>String</i>	>254 byte

2.6.2 Konstanta

Konstanta merupakan suatu nilai yang tidak dapat diubah selama proses program berlangsung. Nilai konstanta selalu tetap. Konstanta harus didefinisikan terlebih dahulu di awal program. Konstanta dapat bernilai

integer, pecahan, karakter ataupun string. Contoh konstanta : $A=5$ $c=1.1$.

(Iswanto, 2009:31)

2.6.3 Variabel

Variabel adalah suatu pengenal (*identifier*) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program yang dapat diubah-ubah sesuai dengan kebutuhan. Nama dari *variable* terserah sesuai dengan yang diinginkan namun hal yang terpenting adalah setiap variabel diharuskan :

1. Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf, max 32 karakter.
2. Tidak boleh mengandung spasi atau symbol-simbol khusus seperti : \$, ?, %, #, !, &, *, (,), -, +, = dan lain sebagainya kecuali *underscore*.
3. Deklarasi, sangat diperlukan bila akan menggunakan pengenal (*identifier*) dalam suatu program. (Fahmiza, 2010:1)

2.6.4 Operator

Terdapat beberapa macam jenis operator dalam *software Basic Compiler* yaitu :

1. Operator Penugasan

Operator Penugasan (*Assignment operator*) dalam Bahasa Basic berupa “=”.

2. Operator Aritmatika

* : untuk perkalian
 / : untuk pembagian
 + : untuk penambahan
 - : untuk pengurangan
 % : untuk sisa pembagian (modulus)

3. Operator Hubungan (Perbandingan)

Operator hubungan digunakan untuk membandingkan hubungan dua buah operand atau sebuah nilai / variable, misalnya :

= 'Equality $X = Y$

< '*Less than X < Y*
 > '*Greater than X > Y*
 <= '*Less than or equal to X <= Y*
 >= '*Greater than or equal to X >= Y*

4. Operator Logika

Operator logika digunakan untuk membandingkan logika hasil dari operator-operator hubungan. Operator logika ada empat macam, yaitu :

NOT '*Logical complement*
 AND '*Conjunction*
 OR '*Disjunction*
 XOR '*Exclusive or*

5. Operator Bitwise

Operator *bitwise* digunakan untuk memanipulasi bit dari data yang ada di memori. Operator bitwise dalam Bahasa Basic :

Shift A, Left, 2 : Pergeseran bit ke kiri
 Shift A, Right, 2 : Pergeseran bit ke kanan
 Rotate A, Left, 2 : Putar bit ke kiri
 Rotate A, right, 2 : Putar bit ke kanan

2.6.5 Pernyataan Kondisional

Pernyataan ini digunakan untuk melakukan pengambilan keputusan terhadap dua buah bahkan lebih kemungkinan untuk melakukan suatu blok pernyataan atau tidak. Konstruksi penulisan pernyataan

IF-THEN-ELSE-END IF pada bahasa BASIC sebagai berikut:

IF pernyataan kondisi 1 THEN

'blok pernyataan 1 yang dikerjakan bila kondisi 1 terpenuhi

IF pernyataan kondisi 2 THEN

'blok pernyataan 2 yang dikerjakan bila kondisi 2 terpenuhi

IF pernyataan kondisi 3 THEN

'blok pernyataan 3 yang dikerjakan bila kondisi 3 terpenuhi

Setiap penggunaan pernyataan *IF-THEN* harus diakhiri dengan perintah *END IF* sebagai akhir dari pernyataan kondisional

2.7 *Java*

Java adalah salah satu bahasa pemrograman berorientasi objek (OOP-*Object Oriented Programming*). Paradigma OOP menyelesaikan masalah dengan merepresentasikan masalah ke model objek. (Wahanakomputer, 2003:3)

Proyek *Java* dimulai pada tahun 1991, ketika sejumlah insinyur perusahaan Sun yang dimotori oleh James Gosling mempunyai keinginan untuk mendesain sebuah bahasa komputer kecil yang dapat dipergunakan untuk peralatan konsumen seperti kotak tombol saluran TV. Proyek ini kemudian diberi nama sandi *Green*.

Keharusan untuk membuat bahasa yang kecil, dan kode yang ketat mendorong mereka untuk menghidupkan kembali model yang pernah dicoba oleh bahasa UCSD Pascal, yaitu mendesain sebuah bahasa yang portable yang menghasilkan kode intermediate. Kode intermediate ini kemudian dapat digunakan pada banyak komputer yang interpretnya telah disesuaikan.

Karena orang-orang Sun memiliki latar belakang sebagai pemakai *unix* sehingga mereka lebih menggunakan C++ sebagai basis bahasa pemrograman mereka, maka mereka secara khusus mengembangkan bahasa yang berorientasi objek bukan berorientasi prosedur. Seperti yang dikatakan Gosling ”*Secara keseluruhan, bahasa hanyalah sarana, bukan merupakan tujuan akhir*”. Dan Gosling memutuskan menyebut bahasanya dengan nama “Oak” (diambil dari nama pohon yang tumbuh tepat diluar jendela kantornya di Sun), tetapi kemudian nama Oak diubah menjadi *Java*, karena nama Oak merupakan nama bahasa komputer yang sudah ada sebelumnya.

Pada tahun 1994 sebagian besar orang menggunakan mosaic, browser web yang tidak diperdagangkan yang berasal dari pusat *Supercomputing* Universitas Illinois pada tahun 1993. (Mosaic sebagian ditulis oleh Marc Andreessen dengan bayaran \$6.85 per jam, sebagai mahasiswa yang melakukan studi praktek. Di kemudian hari ia meraih ketenaran sebagai salah seorang pendiri dan pemimpin teknologi di *netscape*)

Browser yang sesungguhnya dibangun oleh Patrick Naughton dan Jonathan Payne dan berkembang ke dalam browser *HotJava* yang kita miliki saat

ini. Browser *HotJava* ditulis dalam *Java* untuk menunjukkan kemampuan *Java*. Tetapi para pembuat juga memiliki ide tentang suatu kekuatan yang saat ini disebut dengan applet, sehingga mereka membuat browser yang mampu menerjemahkan kode byte tingkat menengah. “Teknologi yang Terbukti” ini diperlihatkan pada *SunWorld '95* pada tanggal 23 Mei 1995, yang mengilhami keranjingan terhadap *Java* terus berlanjut. (Risawati pdf, 2008:1)

2.7.1 Tata Bahasa *Java*

Program *Java* adalah kumpulan spasi, komentar, kata kunci, identifier, literal, operator, dan pemisah. Spasi *Java* adalah bahasa bebas bentuk. Asalkan ada sekurang-kurangnya satu spasi, tab, atau baris baru diantara setiap token sebelum disisipi operator atau pemisah lain. Komentar ada beberapa bentuk :

1. Komentar baris tunggal

Diawali dengan tanda // dan diletakkan diakhir baris yang diberi komentar.

2. Komentar baris banyak

Diawali dengan tanda /* dan ditutup dengan */ semua diantara kedua tanda tersebut dianggap komentar dan akan diabaikan oleh compiler. Contoh penulisan :

```
/*
 *   komentar.....
 *   komentar.....
 */
```

3. Komentar terdokumentasi

Menggunakan piranti *Javadoc*, yang menggunakan komponen compiler *Java* untuk secara otomatis menghasilkan dokumentasi antarmuka public suatu class. Aturan pembuatan komentar yang dapat diolah oleh *Javadoc* adalah : sebelum deklarasi class, method, dan variabel public harus digunakan komentar bertanda /** untuk menyatakan komentar dokumentasi, diakhiri dengan tanda */. *Javadoc* akan mengenali sejumlah variabel khusus yang didahului dengan tanda @ didalam bagian komentar. Contoh penulisan:

```
/**
 *   komentar....
 *   komentar.....
 */
```

2.7.2 Variabel

variabel adalah satuan dasar penyimpanan dalam program *Java*. Suatu variabel didefinisikan dengan kombinasi identifier, tipe, dan cakupan. Bergantung pada tempat kita mendeklarasikannya, variabel dapat bersifat lokal atau sementara, misalnya didalam perulangan *for*, atau dapat juga berupa variabel instans yang dapat diakses oleh semua method dalam class. Cakupan lokal dinyatakan dalam kurung kurawal.

Table 2.2 Tabel Jenis – jenis Variabel

(Risdawati, 2008: 3)

Simbol	Nama	Fungsi
()	Kurung	Digunakan untuk menghimpun parameter dalam definisi dan pemanggilan method, juga digunakan untuk menyatakan tingkatan pernyataan, menghimpun pernyataan untuk pengaturan alur program dan menyatakan tipe cast.
{ }	kurung kurawal	Digunakan untuk menghimpun nilai yang otomatis dimasukkan kedalam array, juga digunakan untuk mendefinisikan blok program, untuk cakupan class, method, dan lokal.
[]	kurung siku	Digunakan untuk menyatakan tipe array, juga digunakan untuk membedakan nilai array.
;	titik-koma	pemisah pernyataan.
,	Koma	Pemisah urutan identifier dalam deklarasi variabel, juga digunakan untuk mengaitkan pernyataan didalam pernyataan <i>for</i> .
.	Titik	Dugunakan untuk memisahkan nama paket dari sub-paket dan class, juga digunakan untuk memisahkan variabel atau method dari variabel referensi.

2.7.3 Tipe Data

Java merupakan contoh bahasa yang *strongly typed language*. Hal ini berarti bahwa setiap variabel harus memiliki tipe yang sudah dideklarasikan. Terdapat 8 tipe primitif, 6 diantaranya adalah tipe bilangan (4 tipe *integer*, 2 tipe *floating point*), 1 tipe karakter char, digunakan mengawa-sandi (*encode*) *Unicode*, dan 1 tipe boolean.

a. Integer

Tabel 2.3 Jenis Tipe Integer
(Risdawati, 2008: 6)

Tipe	Tempat yang Diperlukan	Jangkauan (<i>inclusive</i>)
<i>Int</i>	4 byte	- 2.147.483.648 sampai 2.147.483.647 (hanya lebih dari 2 miliar)
<i>Short</i>	2 byte	- 32.768 sampai 32.767
<i>Long</i>	8 byte	-9.223.372.036.854.775.808L sampai 9.223.372.036.854.775.807L
<i>Byte</i>	1 byte	- 128 sampai 127

b. Floating Point

Tabel 2.4 Jenis Floating Point
(Risdawati, 2008: 6)

Tipe	Tempat Yang Dibutuhkan	Jangkauan
<i>float</i>	4 byte	secara kasar 3,40282347E+38F (7 digit desimal signifikan)
<i>double</i>	8 byte	secara kasar $\pm 1,79769313486231570E+308$ (15 digit desimal signifikan)

c. Char

Tipe *char* menggunakan tanda kutip tunggal untuk menyatakan suatu *char*. Tipe *char* juga menyatakan karakter dalam upaya mengawakan-unicode, yang merupakan kode 2-byte. Karakter unicode paling sering dinyatakan dalam istilah skema pengkodean hexadesimal yang dimulai dari `\u0000` sampai `\uFFFF`. Selain karakter bebas (escape ‘`\u`’ yang menyatakan karakter unicode di *Java* terdapat juga.

Tabel 2.5 Tabel Jenis Char

(Risdawati, 2008: 6)

<code>\b</code>	backspace	<code>\u0008</code>
<code>\t</code>	tab	<code>\u0009</code>
<code>\n</code>	linefeed	<code>\u000a</code>

\r	carriage return	\u000d
\"	double quote	\u0022
\'	single quote	\u0027
\\	a backslash	\u005c

d. Boolean

Tipe *boolean* memiliki nilai *true* dan *false*. Tipe ini digunakan untuk *logical testing* dengan menggunakan operator relasional.

Konversi antar Nilai Numerik

Operasi biner apapun pada variabel numerik dengan tipe yang berbeda dapat diterima dan diperlakukan dengan cara seperti dibawah ini :

1. Jika tipe operand adalah *double*, maka yang lain juga akan diperlakukan sebagai *double* pada lingkup operasi tersebut.
2. Jika operand adalah *float*, maka yang lain juga akan diperlakukan sebagai *float*.
3. Jika operand adalah *long*, maka yang lain juga akan diperlakukan sebagai *long*.

konversi yang diijinkan adalah sebagai berikut :

byte → *short* → *int* → *long* → *float* → *double*

Dimana kita dapat memberikan nilai variabel suatu tipe disebelah kiri ke tipe disebelah kanannya. (Risawati, 2008: 6)

2.7.4 Assignment Operator (=)

a. Shorthand assignment operator

Tabel 2.6 Tabel Jenis Assignment Operator

(Risawati, 2008: 7)

Operator	Usage	Meaning
+=	X += Y	X = X + Y
-=	X -= Y	X = X - Y
*=	X *= Y	X = X * Y
/=	X /= Y	X = X / Y
%=	X %= Y	X = X % Y

b. Arithmetic Operator

Tabel 2.7 Tabel Jenis Arithmetic Operator

(Risawati, 2008: 7)

Operator	Operation
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo

c. Bitwise Operator

Tabel 2.8 Tabel Bitwise Operator

(Risawati, 2008: 7)

Operator	Operation
&	AND
	OR
^	XOR
>>	Shift Kanan
<<	Shift Kiri
>>>	Shift Kanan isi dengan nol

d. Unary Operator

Tabel 2.9. Tabel Unary Operator

(Risawati, 2008: 7)

Operator	Operation
~	Unary NOT
-	Minus
++	Increment
--	Decrement

e. Relational Operator

Tabel 2.10. Tabel Relational Operator

(Risdawati, 2008: 8)

Operator	Operation
==	<i>Equal To</i>
!=	<i>Not Equal To</i>
>	<i>Greater Than</i>
<	<i>Less Than</i>
>=	<i>Greater or Equal To</i>
<=	<i>Less or Equal To</i>

f. Logical Operator

Tabel 2.11. Tabel Logical Operator

(Risdawati, 2008: 8)

Operator	Operation
!	<i>Short-circuit NOT</i>
&&	<i>Short-circuit AND</i>
	<i>Short-circuit OR</i>
?:	<i>Operator ternary if-then-else</i>

g. Preseden Operator

Tabel 2.12. Tabel Preseden Operator

(Risdawati, 2008: 8)

Tertinggi			
()	[]	.	
++	--	~	!
*	/	%	
+	-		

>>	>>>	<<	
>	>=	<	<=
==	!=		
&			
^			
&&			
?:			
=	op=		
Terendah			

2.7.5 CONTROL FLOW

2.7.5.1 IF – ELSE

Bentuk *IF-ELSE* menyebabkan eksekusi dijalankan melalui sekumpulan keadaan boolean sehingga hanya bagian tertentu program yang dijalankan. Bentuk umum pernyataan *IF-ELSE* :

if (boolean expression) statement 1; [else statement 2;]

Klausula *else* bersifat *optional*, setiap statement dapat berupa satu statement tunggal atau dapat berupa satu blok statement yang ditandai dengan tanda { } (kurung kurawal). *Boolean expression* dapat berupa sembarang pernyataan boolean yang menghasilkan besaran boolean.

2.7.5.2 Break

Java tidak memiliki pernyataan goto. Penggunaan goto adalah untuk membuat percabangan secara sembarang yang membuat program sulit dimengerti dan mengurangi optimasi compiler tertentu. Pernyataan break pada *Java* dirancang untuk mengatasi semua kasus tersebut. Istilah break mengacu kepada proses memecahkan blok program. Proses tersebut memerintahkan runtime untuk

menjalankan program dibelakang blok tertentu. Untuk dapat ditunjuk blok diberi nama/label. Break juga dapat digunakan tanpa label untuk keluar dari suatu *loop* dan pernyataan *switch*. Penggunaan break menunjukkan bahwa kita akan keluar dari satu blok program.

2.7.5.3 Switch

Pernyataan *switch* memberiiikan suatu cara ubtuk mengirimkan bagian program berdasarkan nilai suatu variabel atau pernyataan tunggal. Bentuk umum pernyataan *switch* :

```
switch (expression)
{
    case value1 :
        Statement;
        break;
    case value2 :
        Statement;
        break;
    case valueN :
        Statement;
        break;
    default;
}
```

Expression dapat menghasilkan suatu tipe sederhana, dan setiap *value* yang disebutkan pada pernyataan case harus berupa tipe yang cocok. Pernyataan switch bekerja dengan cara membandingkan nilai expression dengan setiap nilai pada pernyataan *case*. Jika ada yang cocok maka urutan program yang ada di pernyataan case tersebut akan dijalankan, jika tidak ada yang cocok, program akan menjalankan *default*

2.7.5.4 Return

Java menggunakan bentuk *sub-routine* yang disebut method untuk mengimplementasikan antarmuka prosedural ke *class* objek. Setiap saat dalam method dapat digunakan pernyataan return yang menyebabkan eksekusi mencabang kembali ke pemanggil method.

2.7.5.5 Perulangan (Looping)

Looping artinya mengulangi eksekusi blok program tertentu sampai tercapai kondisi untuk menghentikannya. Setiap perulangan memiliki 4 bagian:

- a. Inisialisasi, adalah program yang menyiapkan bagian awal perulangan
- b. Badan program, adalah pernyataan yang ingin kita ulangi.
- c. Iterasi, adalah program yang sering digunakan untuk penambahan atau pengurangan pencacah dan index.
- d. Terminasi, adalah pernyataan boolean yang diperiksa setiap kali selama perulangan dilaksanakan untuk melihat apakah perulangan sudah saatnya dihentikan.

2.7.5.6 While

While adalah pernyataan perulangan yang paling mendasar pada *Java*. Penggunaan pernyataan *while* akan menyebabkan rekursi pernyataan secara terus menerus selama pernyataan booleannya bernilai *true*. Bentuk umum dari pernyataan *while*:

```
[initialization;]
while (termination)
{
    body program ;
    [iteration;]
}
```

2.7.5.7 Do-while

Penggunaan pernyataan *do-while* menyebabkan body program akan dieksekusi sekurang-kurangnya 1 kali walaupun pernyataan booleannya menghasilkan nilai *false*. Pemeriksaan terminasi dilaksanakan pada akhir program. Bentuk umum pernyataan *do – while* :

```
[initialization;]
do
{
    body program;
    [iteration;]
}
while [termination];
```

2.7.5.8 For

Pernyataan *for* adalah cara praktis untuk menyatakan suatu perulangan. Bentuk umum pernyataan *for* :

```
for {initialization; termination; iteration } body program;
```

Jika keadaan awal tidak menyebabkan termination bernilai true maka pernyataan *body program* dan *iteration* tidak akan dijalankan.

2.7.5.9 Pernyataan koma

Kadang-kadang ada keadaan dimana kita ingin memasukkan lebih dari satu pernyataan inisialisasi atau terminasi, *Java* menyediakan cara lain untuk menyatakan beberapa pernyataan sekaligus. Penggunaan koma untuk memisahkan pernyataan di batasi hanya untuk digunakan didalam tanda kurung untuk pernyataan *for*.

2.7.5.10 Continue

Pernyataan *continue* akan menghentikan iterasi yang bersangkutan tetapi program akan menjalankan sisa perulangan sampai dengan selesai.

2.8 *Eclipse*

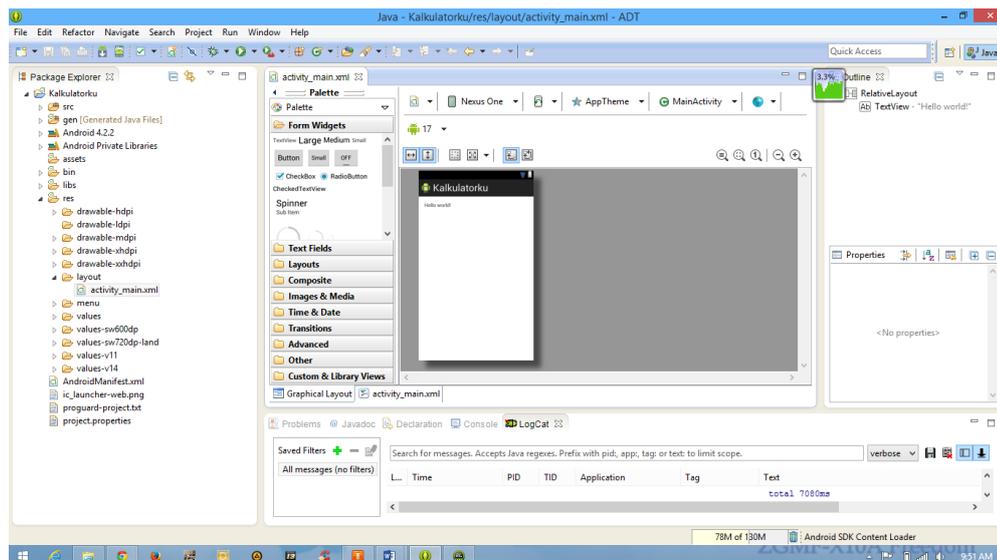
Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat dari *Eclipse*:

1. **Multi-platform:** Target sistem operasi *Eclipse* adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. **Mult-language:** *Eclipse* dikembangkan dengan bahasa pemrograman *Java*, akan tetapi *Eclipse* mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. **Multi-role:** Selain sebagai IDE untuk pengembangan aplikasi, *Eclipse* pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman

perangkat lunak ini. Selain itu, kelebihan dari *Eclipse* yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

(sumber:http://elib.unikom.ac.id/files/disk1/582/jbptunikompp-gdl-arandhitap-29086-9-unikom_a-i.pdf diakses pada 20 April 2015)



Gambar 2.14. Tampilan *Eclipse*

(sumber: Screenshot Software *Eclipse* di laptop penulis)

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM Visual Age for *Java* 4.0. Produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US\$ 40 juta. untuk pengembangannya. Semenjak itu konsorsium *Eclipse* Foundation mengambil alih untuk pengembangan *Eclipse* lebih lanjut dan pengaturan organisasinya.

Sejak versi 3.0, *Eclipse* pada dasarnya merupakan sebuah *kernel*, yang mengangkat *plug-in*. Apa yang dapat digunakan di dalam *Eclipse* sebenarnya adalah fungsi dari *plug-in* yang sudah diinstal. Ini merupakan basis dari *Eclipse* yang dinamakan *Rich Client Platform* (RCP). Berikut ini adalah komponen yang membentuk RCP:

1. *Core platform*
2. OSGi

3. SWT (*Standard Widget Toolkit*)
4. JFace
5. *Eclipse Workbench*

Secara standar *Eclipse* selalu dilengkapi dengan JDT (*Java Development Tools*), *plug-in* yang membuat *Eclipse* kompatibel untuk mengembangkan program *Java*, dan PDE (*Plug-in Development Environment*) untuk mengembangkan *plug-in* baru. *Eclipse* beserta *plug-in*-nya diimplementasikan dalam bahasa pemrograman *Java*.

Konsep *Eclipse* adalah IDE yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik^[2]. Jadi, *Eclipse* tidak saja untuk mengembangkan program *Java*, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal *plug-in* yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat *plug-in* CDT (*C/C++ Development Tools*). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh *Eclipse*, *plug-in* UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat *plug-in* sesuai dengan keinginannya. Salah satu situs yang menawarkan *plug-in* secara gratis seperti *Eclipse downloads by project*. (Hernita, 2013: 2)

2.9 ADT (*Android Developer Tools*)

Android Development Tools (ADT) adalah sebuah *plugin* yang digunakan pada *Eclipse* IDE. *Plugin* ini dirancang untuk memberikan keleluasaan dan kemudahan dalam membangun sebuah aplikasi berbasis OS *Android*.

ADT memperluas kemampuan *Eclipse* sehingga kita dapat dengan mudah membuat sebuah projek *Android*, membuat UI nya, menambahkan komponen berdasarkan *Android Framework API* (*Application Programming Interface*), men-debug aplikasi dengan menggunakan *Android SDK* dan bahkan mengeksport file APK untuk mendistribusikan aplikasi yang kita buat. ((Hernita, 2013: 126)