

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Pengertian Komputer**

Istilah komputer mempunyai arti luas, kata komputer berasal dari bahasa Latin yaitu *computare* yang artinya menghitung, dalam bahasa Inggris disebut *to compute*. Secara definisi komputer diterjemahkan sebagai sekumpulan alat elektronik yang saling bekerja sama, dapat menerima data (*input*), mengolah data (proses) dan memberikan informasi (*output*) yang terorganisasi dibawah kontrol program yang tersimpan dalam *hardisk* dan atau media penyimpanan (Fatoni, 2011).

Komputer adalah sistem elektronik yang terdiri dari perangkat lunak (*software*) dan perangkat keras (*hardware*) yang di dalamnya terdapat ribuan bahkan jutaan komponen yang dapat saling bekerja sama, serta membentuk sebuah sistem kerja yang rapi dan teliti. Sistem ini kemudian dapat digunakan untuk melaksanakan serangkaian pekerjaan secara otomatis, berdasar urutan instruksi ataupun program yang diberikan kepadanya. Definisi yang ada memberi makna bahwa komputer memiliki lebih dari satu bagian yang saling bekerja sama, dan bagian-bagian itu baru bisa bekerja jika ada aliran listrik

yang mengalir didalamnya. Istilah mengenai sekelompok mesin, ataupun istilah mengenai jutaan komponen kemudian dikenal sebagai *hardware* komputer atau perangkat keras komputer. Sedangkan yang disebut program adalah kumpulan dari beberapa intruksi atau perintah terperinci yang sudah disiapkan agar komputer dapat melakukan fungsinya dengan cara yang sudah ditentukan (Fatoni, 2011).

## **2.2 Software Aplikasi yang digunakan**

### **2.2.1 Lazarus**

Lazarus adalah suatu lingkungan pengembangan terintegrasi (IDE) untuk objek bahasa Pascal dan dapat digunakan untuk membuat semua jenis aplikasi tidak hanya aplikasi GUI tetapi juga aplikasi yang dapat dijalankan di *command line* (Anonim, 2012a).

Lazarus merupakan sistem pengembangan sumber terbuka yang dibangun berdasarkan pada *compiler Free Pascal* dengan menambahkan lingkungan pengembangan terintegrasi (IDE) yang menyertakan editor penerangan-sintaks kode dan desainer formulir visual, juga librari komponen yang sangat kompatibel dengan *Visual Component Library* (VCL) pada Delphi. *Lazarus Component Library* (LCL) menyertakan persamaan untuk banyak kontrol VCL familier seperti form, tombol, kotak teks yang dipakai untuk membuat aplikasi yang memiliki antarmuka pengguna grafis (GUI).

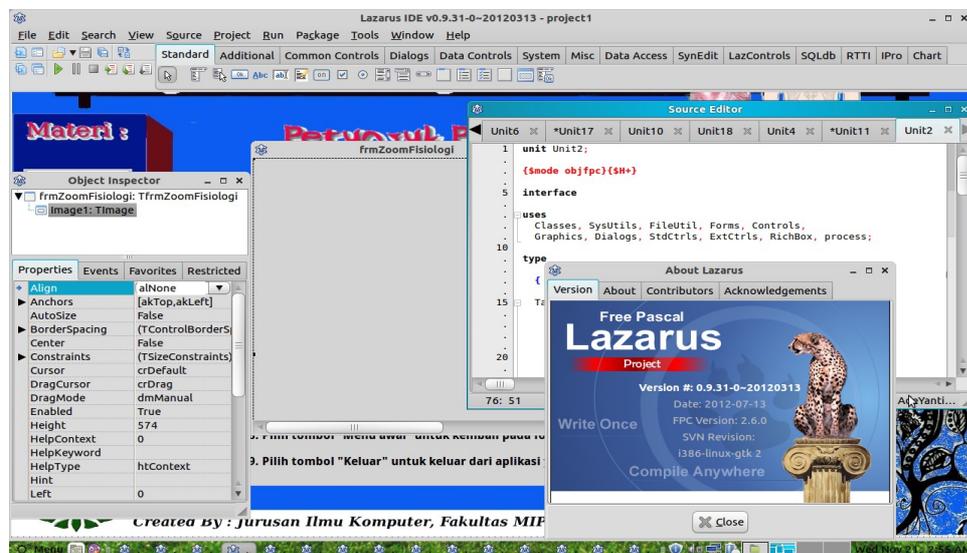
Lazarus ditulis dalam Pascal. Kode sumber lengkap tersedia tidak hanya untuk kompilator Lazarus IDE, tapi juga untuk semua unit yang membuat Lazarus LCL (Anonim, 2012b).

IDE (*Integrated Development Environment*) yang juga disebut sebagai *Integrated Design/Debugging Environment*, adalah perangkat lunak komputer yang berfungsi untuk membantu pemrogram dalam mengembangkan perangkat lunak. Singkatnya, IDE merupakan suatu lingkungan pengembangan aplikasi yang terintegrasi yaitu lengkap dengan beragam *tools* atau utilitas pendukung (Anonim, 2012b).

Lazarus bersifat *open source*, tersedia untuk banyak platform terutama Linux, Windows dan Macintosh. Pengembangan aplikasi yang disediakan oleh Lazarus terasa sebagaimana yang terdapat di Delphi, IDE Pascal Visual yang terkenal di lingkungan Windows (Anonim, 2012b).

Seperti Delphi, Lazarus adalah piranti pemrograman keperluan umum. Lazarus dan Delphi keduanya adalah contoh bagus dari aplikasi GUI dengan fitur lengkap. Ketika mengembangkan aplikasi GUI dengan Lazarus, *user* tidak hanya mengembangkan unit-unit dari kode Pascal, *user* juga mendesain formulir yang berisi kontrol visual seperti tombol dan kotak daftar. Seperti Delphi, desain form dalam Lazarus dikerjakan secara visual. Properti kontrol dapat disetel dalam IDE atau dalam kode. (Anonim, 2012b)

Aplikasi yang terintegrasi memberikan banyak sekali keuntungan. Adapun keuntungan aplikasi yang terintegrasi diantaranya yaitu memudahkan masing-masing bagian dalam sistem untuk saling berkomunikasi. Dengan demikian kerja menjadi semakin efektif dan efisien, baik dalam hal waktu ataupun tenaga (Anonim, 2012b).



Gambar 2.1. Tampilan Lazarus

## 2.2.2 Free Pascal

*Free Pascal* (FPC) adalah kompilator Pascal sumber terbuka dengan dua fitur utama: kompatibilitas Delphi tingkat tinggi dan ketersediaan berbagai platform, termasuk Windows, Mac OS X, dan Linux. Kompatibilitas *Free Pascal* dengan Delphi termasuk tidak hanya dukungan yang sama untuk bahasa pemrograman *Object Pascal* yang digunakan Delphi, tapi juga untuk banyak pustaka. Ini termasuk unit familier seperti *System*, *SysUtils*, *StrUtils*, *DateUtils*, *Classes*, *Variants*, *Math*, *IniFiles* dan *Registry*, yang

disertakan dengan *Free Pascal* pada semua *platform* yang didukung. *Free Pascal* juga menyertakan unit seperti Windows, ShellAPI, BaseUnix, Unix dan DynLibs untuk mengakses spesifik fitur ke sistem operasi. Ada lusinan unit yang menjadi inti dari apa yang biasanya dirujuk sebagai *Free Pascal run-time library* (RTL) (Anonim, 2012c).

### 2.3 MySQL

MySQL merupakan *database* yang paling banyak dipakai di kalangan *programmer web*, dengan alasan bahwa program ini merupakan *database* yang sangat kuat dan cukup stabil untuk digunakan sebagai media penyimpanan data. Kemampuan lain yang dimiliki MySQL adalah mampu mendukung *Relasional Database Manajemen System* (RDBMS), sehingga dengan kemampuan ini MySQL akan mampu menangani data-data perusahaan yang berukuran sangat besar hingga berukuran Giga Byte. (Nugroho, 2004).

Fungsi MySQL dapat dikatakan sebagai *interpreter query*, karena setiap menggunakan *query* SQL (perintah SQL) harus meletakkannya di dalam fungsi ini. Dengan kata lain, SQL tidak dapat dijadikan tanpa adanya fungsi MySQL. MySQL termasuk jenis *Relational Database Management System* (RDBMS). Sehingga istilah seperti tabel, baris dan kolom tetap digunakan dalam MySQL. Pada MySQL, sebuah *database* mengandung beberapa tabel,

tabel terdiri dari sejumlah baris dan kolom.

SQL merupakan kependekan *Structured Query Language*. SQL digunakan untuk berkomunikasi dengan sebuah *database*. SQL adalah bahasa yang meliputi perintah-perintah untuk menyimpan, menerima, memelihara, dan mengatur akses-akses ke basis data serta digunakan untuk memanipulasi dan menampilkan data dari *database* (Rosari, 2008).

### 2.3.1 Database dan Tabel dalam MySQL

Ada dua perintah yang perlu diketahui untuk mendapatkan informasi yang berkenaan dengan *database* dan tabel. Informasi pertama adalah mengetahui daftar *database*, dan yang kedua adalah informasi mengenai keberadaan tabel.

Berikut adalah perintah yang digunakan untuk membuat dan mengaktifkan *database*.

```
mysql> CREATE DATABASE data_sekolahdb;  
Query OK, 1 row affected (0,05 sec)
```

Perintah di atas adalah perintah untuk membuat *database* baru dengan nama *data\_sekolahdb*. Pernyataan "Query OK, 1 row affected (0,05 sec)" menandakan perintah yang telah ditulis sudah berjalan dengan baik, sehingga *database* baru telah berhasil dibuat. Untuk bekerja pada *database data\_sekolahdb* perlu diaktifkan *database* yang bersangkutan terlebih dahulu dengan menggunakan perintah berikut.

```
Mysql> USE data_sekolahdb;
Database changed
```

Sedangkan apabila hendak melihat informasi mengenai keberadaan tabel pada suatu *database*, menggunakan perintah sebagai berikut.

```
Mysql> SHOW TABLES FROM mysql;
```

(Nugroho, 2009)

### 2.3.2 Perintah DDL (*Data Definition Language*)

DDL (*Data Definition Language*) adalah sub bahasa yang dimiliki SQL untuk melakukan pendefinisian data. Kemampuan yang dapat dilakukannya adalah membuat tabel, merubah struktur maupun menghapus struktur tabel dan menghapus tabel serta *database*.

#### 1. Menggunakan perintah *CREATE*

Perintah ini merupakan bagian dari perintah DDL yang digunakan untuk membangun atau mendesain struktur tabel baru. Untuk membuat tabel, sintaks yang digunakan adalah:

```
CREATE TABLE nama_tabel
(Kolom_a TYPE (lebar),
Kolom_b TYPE (lebar),
.....',
Kolom_n TYPE (lebar)
);
```

#### 2. Menggunakan perintah *ALTER*

*ALTER* merupakan perintah DDL yang digunakan untuk melakukan perubahan desain tabel menjadi tabel dengan struktur baru.

```
Mysql> ALTER TABLE (nama_tabel)
- > CHANGE (nama field yang akan diubah) (nama field yang
diinginkan) TYPE;
```

### 3. Menggunakan perintah *DROP*

Perintah *DROP* digunakan untuk menghapus kolom, maka perintah yang dapat digunakan adalah:

```
mysql> ALTER TABLE (nama_tabel)
      - > DROP (nama_kolom);
```

(Nugroho, 2009)

### 2.3.3 *Data Manipulation Language (DML)*

Perintah DML merupakan sub bahasan SQL yang digunakan untuk memanipulasi data atau manajemen data yang ada di dalam *database* MySQL.

#### 1. Menggunakan perintah *INSERT*

Perintah *INSERT* digunakan untuk memasukkan data ke dalam tabel tertentu. Dengan menggunakan perintah ini dapat dilakukan penambahan data baru ke dalam tabel yang masih kosong ataupun tabel yang telah berisi.

Berikut adalah sintaks dasar untuk memasukkan data:

```
mysql> INSERT INTO namatabel
      (koloma, kolomb, ..... , kolomn)
      VALUES('isi_koloma', 'isi_kolomb', '.....',
      'isi_kolomn');
```

#### 2. Menggunakan perintah *SELECT*

Perintah *SELECT* digunakan untuk menyeleksi atau memilih atau menampilkan sebagian semua data yang ada di dalam sebuah kolom. Perintah *SELECT* dapat digunakan untuk menampilkan semua data (kolom dan baris) dari sebuah tabel. Parameter yang digunakan untuk menunjuk

pada semua kolom adalah tanda bintang (\*). Berikut adalah sintaksnya:

```
mysql> SELECT * FROM namatabel;
```

### 3. Menggunakan perintah *UPDATE*

Perintah *UPDATE* digunakan untuk meng-update atau memperbaharui data lama menjadi data terkini (terbaru). Berikut adalah sintaks yang digunakan:

```
mysql> UPDATE namatabell SET koloma='databarur',
kolomb='databarur' ,....., kolomn='databarukun'
WHERE kolomkunci='kodebaris';
```

### 4. Menggunakan perintah *DELETE*

Perintah *DELETE* digunakan untuk menghilangkan atau menghapus sebagian atau semua data yang ada di dalam tabel. Seperti halnya dengan perintah *UPDATE*, untuk menggunakan perintah *DELETE* juga diperlukan kondisi dengan klausa *WHERE* untuk menunjuk baris mana yang hendak dihilangkan. Sintaks dasar perintah *DELETE* adalah sebagai berikut:

```
mysql> DELETE FROM namatabel WHERE kolomkunci='kunci';
```

(Nugroho, 2009)

## 2.4 Model Pengembangan Perangkat Lunak

Model atau metode pengembangan yang digunakan di dalam pembuatan perangkat lunak ini adalah metode *waterfall*. Metode ini merupakan metode yang bersifat sistematis dan berurutan. Adapun tahapan dari metode ini adalah sebagai berikut:

1. Analisis Kebutuhan

Tahapan ini merupakan tahapan pertama pada metode *waterfall* dan merupakan tahapan proses untuk mendefinisikan semua kebutuhan pembuatan *software*.

2. Perancangan sistem perangkat lunak

Tahap ini merupakan tahapan perancangan sistem yang meliputi desain *input* dan desain *output*, *use case* sistem dan lain-lain. Kegiatan ini menentukan arsitektur sistem secara keseluruhan.

3. Implementasi dan pengujian unit

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian unit melibatkan verifikasi bahwa setiap unit telah memenuhi spesifikasinya.

4. Integrasi dan pengujian sistem

Unit program atau program individual diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa persyaratan sistem telah dipenuhi.

5. Operasi dan pemeliharaan

Sistem yang sudah dibuat diinstal dan dipakai. Pemeliharaan mencakup koreksi dari berbagai *error* yang tidak ditemukan di tahap-tahap terdahulu.

(Sommerville, 2003)

## 2.5 Use Case Diagram

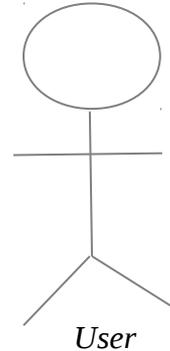
*Use Case Diagram* adalah diagram yang menunjukkan gambaran fungsionalitas yang diharapkan dari sistem, lingkungannya, dan relasi antara sistem dengan lingkungannya atau bisa disebut juga sebagai deskripsi suatu sistem dari sudut pandang *user*. Secara umum, *use case* secara sederhana merupakan narasi tertulis yang menggambarkan peran aktor pada saat berinteraksi dengan sistem (Pressman, 2009).

Berikut ini adalah simbol yang digunakan pada *use-case* diagram:

1. *Actor* : menggambarkan siapa yang menggunakan sistem.
2. *Use-case* : menggambarkan apa yang dilakukan oleh pengguna sistem.



Simbol *use-case*



*User*

Gambar 2.2 Simbol *Use Case*

(Nugroho, 2012)

## 2.6 Data Flow Diagram (DFD)

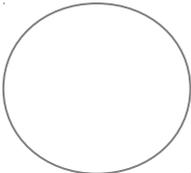
*Data Flow Diagram* (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama *Bubble chart*, *Bubble diagram*, model proses, diagram alur kerja, atau model fungsi (Parno, 2012).

DFD merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisis maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program (Parno, 2012).

Komponen *Data Flow Diagram*:

1.  = *Terminator*

*Terminator* mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan. Biasanya *terminator* dikenal dengan nama entitas luar (*external entity*).

2.  = Proses

Komponen proses menggambarkan bagian mentransformasikan *input* menjadi *output*. Proses diberi nama untuk menjelaskan proses/kegiatan apa yang sedang atau akan dilaksanakan.

3.  = *Data Store*

*Data store* ini biasanya berkaitan dengan penyimpanan-penyimpanan, seperti *file* atau *database* yang berkaitan dengan penyimpanan secara komputerisasi (berhubungan dengan media penyimpanan).

4.  = Alur data

Suatu data *flow* atau alur data digambarkan dengan anak panah, yang menunjukkan arah menuju ke dan keluar dari suatu proses. Alur data ini digunakan untuk menerangkan perpindahan data atau paket data/informasi dari satu bagian sistem ke bagian lainnya (Parno, 2012).

## 2.7 Algoritma

Algoritma adalah jantung ilmu komputer atau informatika. Banyak cabang dari ilmu komputer yang diacu dalam terminologi algoritma, misalnya algoritma *Brensenham* untuk menggambar garis lurus (bidang grafika komputer), algoritma *searching* untuk mencari suatu kata tertentu di dalam suatu teks dan sebagainya (Munir, 2011).

Algoritma merupakan urutan langkah-langkah yang disusun secara sistematis untuk menyelesaikan suatu masalah. Dalam matematika dan komputasi, algoritma merupakan kumpulan perintah untuk menyelesaikan suatu masalah. Perintah-perintah ini dapat diterjemahkan secara bertahap dari awal hingga akhir (Munir, 2011).

Algoritma biasanya disajikan dalam dua bentuk, yaitu :

1) Menggunakan *Flow Chart* (diagram alir)

*Flow Chart* menggambarkan secara keseluruhan urutan proses atau logika, dimana persoalan yang ada akan diselesaikan, dan menunjukkan tingkat detail penyelesaian persoalan (Munir, 2011).

2) Menggunakan *Pseudo-Code*

Para ilmuwan komputer lebih menyukai menuliskan algoritma dalam notasi yang lebih praktis, yaitu *pseudo-code*. *Pseudo-code* (*pseudo* artinya semu atau tidak sebenarnya) adalah notasi yang mirip dengan notasi bahasa pemrograman tingkat tinggi. Lebih tepatnya *pseudo-code* adalah campuran antara bahasa alami dengan bahasa pemrograman. Namun, tidak seperti bahasa pemrograman yang direpotkan dengan tanda titik koma (*semicolon*), indeks, format keluaran, dan sebagainya. Keuntungan menggunakan notasi *pseudo-code* adalah kemudahan mengonversinya atau mentranslasi ke notasi bahasa pemrograman. Karena terdapat korespondensi antara setiap *pseudo-code* dengan notasi bahasa pemrograman. Korespondensi ini dapat dilakukan

dengan menggunakan tabel translasi dari notasi algoritmik ke notasi bahasa pemrograman apapun. (Munir, 2011).

### 2.7.1 Algoritma Pencarian

Pencarian (*searching*) merupakan suatu proses yang fundamental dalam bidang komputasi atau pengolahan data. Pencarian digunakan pada setiap aplikasi dimana diperlukan untuk mengetahui apakah suatu elemen terdapat di dalam daftar yang ada atau menemukan data tertentu di dalam sekumpulan data yang bertipe sama (baik bertipe dasar atau bertipe bentukan) (Munir, 2011).

Algoritma pencarian (*searching algorithm*) adalah algoritma yang menerima sebuah argumen kunci dengan langkah-langkah tertentu akan mencari rekaman dengan kunci tersebut. Setelah proses pencarian dilaksanakan, akan diperoleh salah satu dari dua kemungkinan, yaitu data yang dicari ditemukan (*successfull*) atau tidak ditemukan (*unsuccessfull*).

Ada dua macam teknik pencarian yaitu pencarian sekuensial dan pencarian biner. Perbedaan dari dua teknik ini terletak pada keadaan data. Pencarian sekuensial digunakan apabila data dalam keadaan acak atau tidak terurut. Sebaliknya, pencarian biner digunakan pada data yang sudah dalam keadaan terurut (Munir, 2011).

### 2.7.1.1 Algoritma Pencarian Beruntun (*sequential searching*)

Pencarian beruntun atau pencarian sequensial merupakan metode pencarian yang sederhana. Pencarian berurutan bekerja dengan cara data yang ada dibandingkan satu per satu secara berurutan dengan yang dicari sampai data tersebut ditemukan atau data sudah diperiksa seluruhnya.

Algoritma pencarian beruntun ini pada dasarnya melakukan pengulangan dari 1 sampai dengan jumlah data yang ada sehingga waktu pencarian sebanding dengan jumlah elemen larik. Pada setiap pengulangan, dibandingkan data ke-i dengan yang dicari. Apabila sama, berarti data telah ditemukan. Sebaliknya, apabila sampai akhir pengulangan tidak ada data yang sama, berarti data tidak ada. Pada kasus yang paling buruk, untuk N elemen data harus dilakukan pencarian sebanyak N kali pula. Berikut ini merupakan algoritma pencarian linear atau *sequential search*: (Isnani, 2010).

```
procedure SeqSearch ( input L : LarikInt, input n : integer,
```

```
input x : integer, output ketemu : boolean)
```

```
{ Mencari keberadaan nilai x di dalam larik L[1..n]. }
```

```
{ K.Awal : x dan larik L[1..n] sudah terdefinisi nilainya. }
```

```
{ K.Akhir : ketemu bernilai true jika x ditemukan. Jika x tidak ditemukan, ketemu bernilai false. }
```

```
DEKLARASI
```

```
    i : integer          { pencatat indeks larik }
```

ALGORITMA:

```

i ← 1
while (i < n) and ( L[i] ≠ x) do
    i ← i+1
endwhile
{ i = n or L[i] = x}
if L[i] = x then      { x ditemukan }
    ketemu ← true
else
    ketemu ← false { x tidak ada di dalam
                    larik L }
endif

```

Setiap elemen larik L dibandingkan dengan x dimulai dari elemen pertama, L[1]. Aksi perbandingan dilakukan selama indeks larik i belum melebihi n dan L[i] tidak sama dengan x. Aksi perbandingan dihentikan bila L[i] = x atau i = n. Elemen terakhir, L[n], diperiksa secara khusus. Keluaran yang dihasilkan oleh prosedur pencarian adalah suatu peubah boolean (misal nama peubahnya *ketemu*) yang bernilai *true* jika x ditemukan, atau bernilai *false* jika x tidak ditemukan. (Munir, 2011)

Berikut ini adalah contoh dari penerapan algoritma *sequential search*:

**Contoh.** Di bawah ini adalah larik L dengan 6 elemen, n = 6 elemen:

13	16	14	21	76	15
----	----	----	----	----	----

Misalkan nilai yang dicari adalah:  $x = 21$

Elemen yang dibandingkan (berturut-turut): 13, 16, 14, 21 (ditemukan!)

Indeks larik yang dikembalikan:  $idx = 4$

Misalkan nilai yang dicari adalah:  $x = 13$

Elemen yang dibandingkan (berturut-turut): 13 (ditemukan!)

Indeks larik yang dikembalikan:  $idx = 1$

Misalkan nilai yang dicari adalah:  $x = 15$

Elemen yang dibandingkan (berturut-turut): 13, 16, 14, 21, 76, 21 (tidak ditemukan!)

Indeks larik yang dikembalikan:  $idx = -1$

(Munir, 2011)

## **2.8 Black Box Testing**

Pengujian *Black Box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian ini memungkinkan perekrutan perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *Black Box* bukan merupakan alternatif dari teknik *White Box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode *White Box* (Al Bahra, 2006).

Pengujian *Black Box* didasarkan pada spesifikasi persyaratan dan tidak ada kebutuhan untuk memeriksa kode. Pengujian ini dilakukan pada produk yang telah selesai dikembangkan (Puspaningrum, 2012).

Pengujian *Black Box* berusaha menemukan kesalahan dalam katagori sebagai berikut :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan antarmuka
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

(Al Bahra, 2006).

### **2.8.1 Teknik-teknik *Black Box Testing***

Terdapat beberapa teknik yang termasuk dalam *Black Box Testing*, yaitu :

#### *1. Equivalent Class Partitioning*

*Equivalent Class Partitioning* adalah metode *Black Box Testing* yang membagi domain *input* dari suatu program ke dalam kelas data dari mana *test case* dapat dilakukan. Desain *test case* pada pengujian ini didasarkan pada evaluasi terhadap kelas ekivalensi untuk suatu kondisi *input*. Kelas ekivalen merepresentasikan serangkaian keadaan valid atau *invalid* untuk kondisi *input*. Suatu kondisi *input* dapat berupa harga numeris, rentang harga, serangkaian harga terkait, atau kondisi *Boolean* (Al Bahra, 2006).

## 2. *Boundary Value Analysis*

Pengujian dengan teknik *Boundary Value Analysis* dilakukan dengan menciptakan tes yang melaksanakan tepi kelas *input* dan *output* yang diidentifikasi dalam spesifikasi. Uji kasus dapat diturunkan dari 'batas' dari kelas kesetaraan. Biasanya kesalahan terjadi pada pemrograman batas kelas kesetaraan yang dikenal sebagai "*Boundary Value Analysis*". Terkadang programmer gagal untuk memeriksa proses khusus yang diperlukan terutama pada batas-batas kelas kesetaraan. Misalnya, programmer dapat menggunakan "kurang dari" bukan "kurang dari sama dengan". Pilihan nilai batas termasuk di atas, di bawah dan di antara batas kelas (Puspaningrum, 2012).

## 3. *Decision Tables*

*Decision Tables* adalah aturan yang dapat dibaca manusia yang digunakan untuk mengekspresikan pengetahuan ahli tes atau desainer dalam bentuk yang kompak. *Decision Tables* dapat digunakan ketika hasil atau logika yang terlibat dalam program didasarkan pada seperangkat keputusan dan aturan yang harus diikuti. Tabel keputusan utamanya terdiri dari empat bidang yang disebut rintisan kondisi, kondisi masukkan, rintisan action dan tindakan akhir masukkan (Puspaningrum, 2012).

#### 4. *State Transition Diagrams*

*State Transition Diagrams* adalah alat yang sangat baik untuk menangkap beberapa jenis persyaratan sistem dan dokumen desain sistem internal. Ketika sistem harus mengingat apa yang terjadi sebelum atau ketika perintah valid dan tidak valid dari operasi yang ada, dan kemudian pengujian *State Transition* dapat digunakan. Grafik *state* ini digunakan ketika sistem bergerak dari satu *state* ke *state* lain. Grafik *state* diwakili dengan simbol, lingkaran digunakan untuk menggambarkan negara, panah digunakan untuk menggambarkan transisi, dan *event* digambarkan dengan label pada transisi. Dengan demikian dari keadaan mulai sampai keadaan akhir, berbagai transisi dan rute direpresentasikan dalam bentuk diagram transisi seperti yang disebutkan (Puspaningrum, 2012).

#### 5. *Comparison*

Pengujian ini dilakukan dengan membandingkan beberapa perangkat lunak sejenis untuk menemukan kesalahan *output* dari perangkat lunak tersebut. Pengujian *Comparison* dilakukan pada saat perangkat lunak redundan dikembangkan yaitu dengan mengembangkan versi independen dari perangkat lunak tersebut dengan spesifikasi yang sama (Al Bahra, 2006).

#### 6. *Graph Based*

Pengujian ini dilakukan dengan membuat grafik dari objek-objek yang penting dan hubungan objek-objek sehingga ditemukan kesalahan dari objek

dan hubungan yang ada. Node digunakan untuk merepresentasikan objek, link digunakan untuk merepresentasikan hubungan antar objek, *node weight* menggambarkan properti dari suatu simpul, dan *links weight* digunakan untuk menggambarkan beberapa karakteristik suatu relasi atau hubungan (Al Bahra, 2006).

## **2.9 Pengembangan Sistem Pembelajaran Berbasis Komputer**

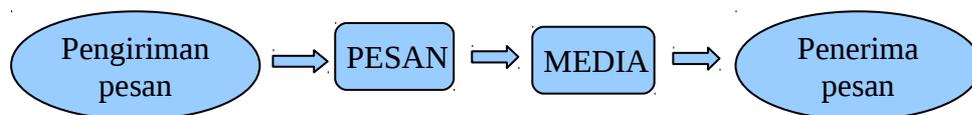
Pembelajaran merupakan proses interaksi antara pengajar atau pendidik dengan peserta didik yang tujuannya untuk membantu peserta didik agar dapat belajar dengan baik. Sedangkan sistem pembelajaran adalah keseluruhan komponen pembelajaran yang saling berhubungan secara terpadu untuk mencapai tujuan pembelajaran yang diinginkan. Adapun komponen-komponen dalam sistem pembelajaran adalah pendidik, peserta didik, materi, strategi dan tujuan pembelajaran, media pembelajaran, evaluasi pembelajaran, dan lain-lain.

Media pembelajaran merupakan pengembangan sistem pembelajaran yang menggunakan komputer sebagai alat bantu pelaksanaannya. Perangkat lunak (*software*) media pembelajaran merupakan hasil dari pengembangan sistem pembelajaran berbasis komputer.

### 2.9.1 Media Pembelajaran Berbasis Komputer

Media adalah suatu alat yang berfungsi untuk menyampaikan pesan. Media merupakan bentuk jamak dari kata “medium” yang berasal dari bahasa latin yang berarti “antara”. Istilah media dapat kita artikan sebagai segala sesuatu yang menjadi perantara atau penyampai informasi dari pengirim pesan (*transmitter*) kepada penerima pesan (*receiver*). (Supriatna, 2009)

Dengan berkembangnya teknologi pada pertengahan abad ke-20 para pengajar juga menggunakan alat bantu audio visual dalam proses pembelajarannya. Hal ini dilakukan untuk menghindari verbalisme yang mungkin terjadi jika hanya menggunakan alat bantu visual saja.



Gambar 2.3 Proses Komunikasi dengan Media

Dalam konteks komunikasi seperti di atas, fungsi media adalah sebagai alat bantu untuk pendidik dalam mengomunikasikan pesan, agar proses komunikasi berjalan dengan sebaik-baiknya (Sanjaya, 2010).

Penggunaan media dalam pembelajaran dapat membantu peserta didik dalam memberikan pengalaman yang bermakna. Penggunaan media dalam pembelajaran dapat mempermudah dalam memahami sesuatu yang abstrak menjadi lebih konkrit.

Secara umum media pembelajaran mempunyai manfaat sebagai berikut:

1. Memperjelas penyajian suatu pesan agar tidak terlalu bersifat verbal.
2. Mengatasi keterbatasan ruang, waktu dan daya indera, misalnya dengan diletakkannya *software* yang sudah dibuat diletakkan di sebuah *server* tertentu sehingga bisa diakses kapan saja, dimana saja dan oleh siapapun.
3. Dengan menggunakan media pembelajaran secara tepat dan bervariasi dapat mengatasi sikap pasif peserta didik. Dalam hal ini media pembelajaran berguna untuk:
  - a) Menimbulkan gairah belajar
  - b) Memungkinkan interaksi langsung antara peserta didik dengan lingkungan dan kenyataan.
4. Dengan sifat yang unik pada peserta didik juga dengan lingkungan dan pengalaman yang berbeda-beda, sedangkan kurikulum dan materi pembelajaran yang sama untuk setiap peserta didik, masalah ini dapat diatasi dengan media pembelajaran dalam kemampuannya:
  - a) Memberikan perangsang yang sama
  - b) Menimbulkan persepsi yang sama tentang materi yang sama.

(Supriatna, 2009)

Adapun keuntungan pembelajaran berbantuan komputer antara lain yaitu:

1. Pembelajaran berbantuan komputer bila dirancang dengan baik, merupakan media pembelajaran yang sangat efektif, dapat dimudahkan dan meningkatkan kualitas pembelajaran.

2. Meningkatkan motivasi belajar
3. Mendukung pembelajaran individual sesuai kemampuan.
4. Melatih untuk terampil memilih bagian-bagian isi pembelajaran yang dikehendaki.
5. Dapat digunakan sebagai penyampai balikan langsung.
6. Materi dapat diulang-ulang sesuai keperluan pengajar atau narasumbernya.

(Haryanto, 2008)

Akan tetapi metode pembelajaran berbantuan komputer juga memiliki keterbatasan:

1. Keterbatasan bentuk dialog/komunikasi karena informasi hanya berjalan satu arah dari *software* ke *user* atau sebaliknya.
2. Terlalu sering menggunakan komputer dapat menyebabkan ketergantungan yang berakibat kurang baik misalnya, mengurangi sikap interaksi sosial yang seharusnya merupakan bagian dalam pendidikan.

(Haryanto, 2008)

## **2.10 Jantung Manusia**

Pada sistem pembelajaran yang dibuat ini, pembahasan meliputi tiga materi utama yaitu anatomi jantung manusia, fisiologi jantung manusia, dan patologi jantung manusia. Anatomi adalah ilmu yang melukiskan letak dan hubungan bagian-bagian tubuh manusia, binatang, atau tumbuh-tumbuhan. Fisiologi

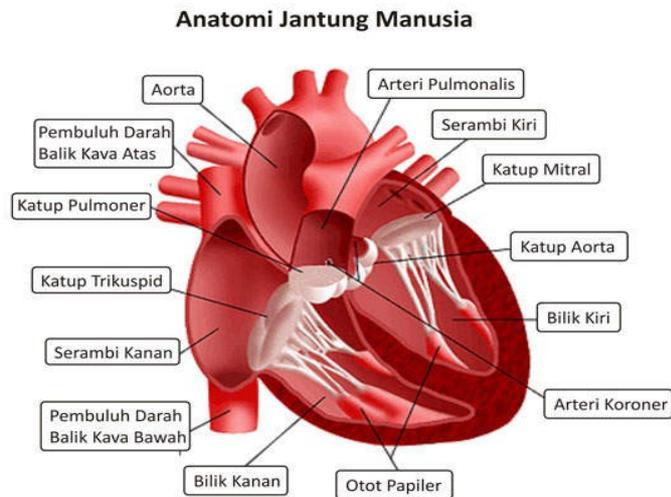
adalah cabang biologi yang berkaitan dengan fungsi dan kegiatan kehidupan atau zat hidup (organ, jaringan, atau sel). Patologi adalah ilmu tentang penyakit.

### 2.10.1 Anatomi Jantung

Jantung manusia merupakan jantung berongga yang memiliki 2 *atrium* dan 2 *ventrikel*. Jantung merupakan organ berotot yang mampu mendorong darah ke berbagai bagian tubuh. Jantung manusia berbentuk seperti kerucut dan berukuran sebesar kepalan tangan, terletak di rongga dada sebelah kiri. Jantung dibungkus oleh suatu selaput yang disebut *perikardium*. Jantung bertanggung jawab untuk mempertahankan aliran darah dengan bantuan sejumlah klep yang melengkapinya. Untuk menjamin kelangsungan sirkulasi, jantung berkontraksi secara periodik. Terdapat beberapa bagian jantung (secara anatomis) diantaranya yaitu :

#### a. Bentuk Serta Ukuran Jantung.

Jantung merupakan organ utama dalam sistem kardiovaskuler. Jantung dibentuk oleh organ-organ *muscular*, *apex* dan *basis cordis*, *atrium* kanan dan kiri serta *ventrikel* kanan dan kiri. Ukuran jantung panjangnya kira-kira 12 cm, lebar 8-9 cm serta tebal kira-kira 6 cm. Berat jantung sekitar 7-15 ons atau 200 sampai 425 gram dan sedikit lebih besar dari kepalan tangan. Setiap harinya jantung berdetak 100.000 kali dan dalam masa periode itu jantung memompa 2000 galon darah atau setara dengan 7.571 liter darah (Anonim, 2012d).



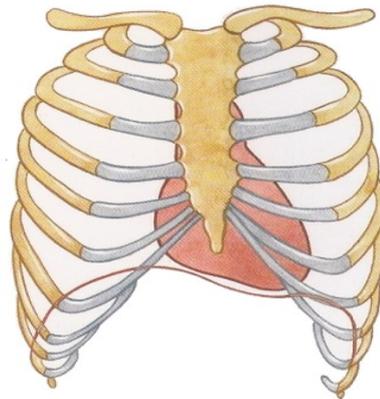
Gambar 2.4 Anatomi jantung manusia

(Sumber gambar: Anonim, 2012h)

Jantung terletak di dalam rongga *mediastum* dari rongga dada (*toraks*), di atas paru-paru. Jantung terdiri dari dua kelompok otot yang secara fungsional dipisahkan oleh jaringan ikat yang disebut *annulus fibrosus*. Kedua kelompok itu adalah:

1. *Superficialis bulbospiral bundle*
2. *Deep bulbospiral bundle*
3. *Superficial sinospiral bundle*
4. *Deep sinospiral bundle*

Arah serat-serat otot *bulbospiral* dan *sinospiral bundle* membentuk sudut tertentu sehingga memperkuat dinding *ventrikel*. (Utari,2007).



**Posisi jantung; inspirasi**

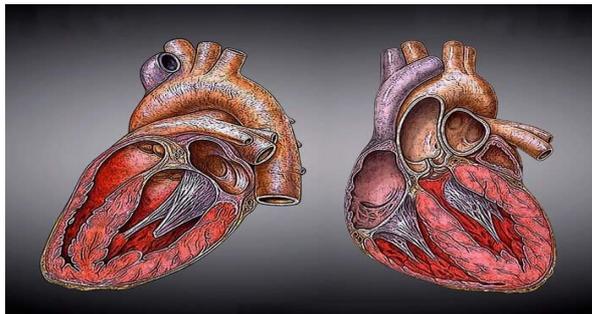
Gambar 2.5 Posisi jantung pada saat inspirasi

(Sumber gambar: Pabst dan Putz, 2007)

Penentuan besarnya jantung sangat penting, baik pada pemeriksaan klinik maupun patologik. Pada wanita berat normal jantung 250-300 gram dan pada pria 300-350 gram. Harus diperhitungkan pula tinggi dan bentuk kerangka (skelet). Oleh karena jantung dapat berhenti dalam keadaan *sistole* atau *diastole* atau dapat mengalami dilatasi sebelum meninggal, ukuran dan lain-lain juga perlu diketahui. Tebal dinding *ventrikel* kanan 3 sampai 5 mm dan *ventrikel* kiri 1,3 – 1,5 cm. (Himawan, 1990).

Jantung terdiri dari empat rongga, serambi kanan dan kiri, bilik kanan dan kiri. Dinding serambi jauh lebih tipis dibandingkan dinding bilik karena bilik harus melawan gaya gravitasi bumi untuk memompa dari bawah ke atas dan memerlukan gaya yang lebih besar untuk mensuplai peredaran darah besar, khususnya pembuluh *aorta*, untuk memompa ke seluruh bagian tubuh yang

memiliki pembuluh darah. Tiap serambi dan bilik pada masing-masing belahan jantung disambungkan oleh sebuah katup. Katup di antara serambi kanan dan bilik kanan disebut katup *trikuspidalis* atau katup berdaun tiga. Sedangkan katup yang ada di antara serambi kiri dan bilik kiri disebut katup mitralis atau katup *bikuspidalis* (katup berdaun dua). (Anonim, 2012d).



Gambar 2.6 Penampang Melintang Jantung Manusia

(Sumber gambar: Heikenwaelder, 2006)

### 2.10.2 Fisiologi Jantung

Tugas jantung sebagai pompa darah dengan dua sistem sirkulasi yang terpisah. Sistem sirkulasi yang lebih besar, meliputi seluruh jaringan tubuh, sehingga untuk itu jantung memompakan darah ke pembuluh nadi lewat *aorta*. Sedang sistem sirkulasi yang lebih kecil, meliputi sirkulasi darah ke paru-paru (*pulmonum*), tempat dimana terjadi pertukaran udara (*oksigenasi*). Setelah kembali dari paru-paru darah yang kembali ke jantung itu merupakan darah bersih yang kaya akan zat asam (*oksigen*). Untuk kemudian dipompakan oleh jantung ke seluruh tubuh. Setelah darah memberi makan seluruh jaringan tubuh, maka darah kembali ke jantung

lewat pembuluh balik (*vena*), darah ini miskin akan zat asam. Darah ini kemudian dipompakan ke paru-paru kembali untuk diperbaharui (*dioksigenasi*).

Arah aliran darah jantung:

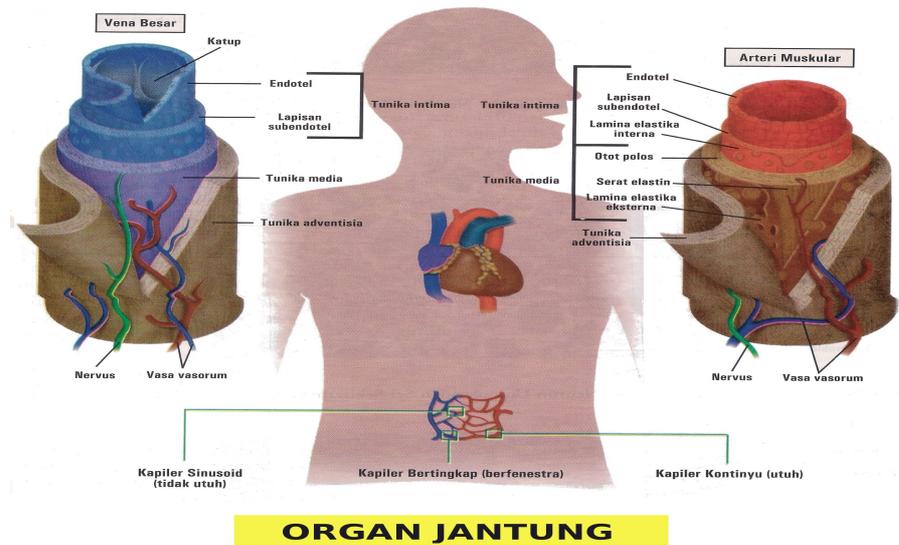
- a. *Sentripetal*, yaitu aliran darah yang menuju ke jantung melalui saluran pembuluh balik (*vena*).
- b. *Sentrifugal*, yaitu aliran darah yang keluar atau meninggalkan jantung melalui saluran pembuluh nadi (*arteri*).

(Irianto, 2007)

Saluran-saluran yang dilalui oleh peredaran darah:

- a. *Vena kava*, yaitu saluran yang dilalui darah dari tubuh ke serambi kiri (*atrium dekstra*) menuju ke paru-paru.
- b. *Arteri pulmonalis*, yaitu saluran yang dilalui darah dari bilik kiri (*ventrikel dekstra*) menuju ke paru-paru.
- c. *Vena pulmonalis*, yaitu saluran yang dilalui oleh darah dari paru-paru menuju ke serambi kanan (*atrium sinistra*) – bilik kanan (*ventrikel sinistra*).
5. *Aorta*, yaitu saluran yang dilalui darah dari bilik kanan menuju tubuh.

(Irianto, 2007)



Gambar 2.7 Organ Jantung

(Chenko, 2003)

Sifat-sifat jantung :

- Automasi, artinya jantung itu masih dapat melakukan fungsinya tanpa dipengaruhi saraf dan dibuktikan dengan cara merusak otak atau sumsum punggung dimana jantung tetap normal melakukan fungsinya untuk beberapa saat.
- Termolabil, jantung dapat berubah denyutnya karena pengaruh suhu lingkungan.
- Sinsitium, organ berupa serabut yang bekerja sebagai satu unit.

(Irianto, 2007)

### 2.10.3 Patologi Jantung

Penyakit kardiovaskular merupakan penyebab utama kematian dan kecacatan di seluruh dunia. Fakta dari WHO (*World Health Organization*) terjadi satu kematian akibat penyakit kardiovaskular setiap dua detik, serangan jantung setiap lima detik dan akibat stroke setiap enam detik dan setiap tahunnya diperkirakan 17 juta orang meninggal akibat penyakit kardiovaskular. Penyakit jantung dewasa ini merupakan penyebab paling utama keadaan sakit dan kematian bangsa berindustri maju. Kira-kira 88% disebabkan karena penyakit jantung iskemik (*Ichemic Heart Disease*) yang juga merupakan penyakit jantung koroner (*Coronary Heart Disease*) (Robbins, 1995).

Kematian sebagai akibat penyakit jantung biasanya disebabkan karena gangguan irama jantung atau kelemahan pemompaan progresif. Gangguan irama jantung terjadi bila jalur konduksi normal dihambat oleh *nekrosis*, radang dan *fibrosis*, maupun bila kesalahan metabolisme lokal menimbulkan fokus iritasi listrik. Selain itu semua penyakit jantung utama, bila dalam keadaan parah, dapat berpengaruh pada kapasitas fungsi pemompaan. Penyakit jantung dikelompokkan dalam beberapa macam yaitu:

#### 1. Gagal Jantung Kongestif (*Congestif Heart Failure*)

Gagal jantung kongestif dimaksud sebagai sindrom klinik yang disebabkan oleh berkurangnya volume pemompaan jantung untuk

keperluan relatif tubuh, disertai hilangnya curah jantung dalam mempertahankan aliran balik *vena*. Jantung melebar melampaui batas ketegangan kontraksi yang disajikan secara wajar. Penambahan volume darah berakibat bendungan nyata, yang selanjutnya menekan jantung. Akhirnya curah jantung menjadi merosot (Robbins, 1995).

## 2. Penyakit Jantung Iskemik

Penyakit jantung iskemik adalah sekelompok sindrom yang berkaitan erat yang disebabkan oleh ketidakseimbangan antara kebutuhan oksigen *miokardium* dan aliran darah. Penyebab tersering penyakit jantung iskemik adalah menyempitnya *lumen arteria koronaria* oleh *aterosklerosis*, sehingga penyakit jantung iskemik sering disebut penyakit jantung koroner atau penyakit *arteria koronaria*. Penyakit jantung iskemik dapat timbul dari satu dari empat sindrom, bergantung pada kecepatan dan keparahan penyempitan *arteria koronaria* dan respons *miokardium*:

- (1) Berbagai bentuk *angina pectoris* (nyeri dada)
  - (2) *Infark miokardium* akut (MI)
  - (3) Kematian jantung mendadak
  - (4) *Penyakit jantung iskemik kronis* disertai *gagal jantung kongestif*
- (Kumar, 2007).

### 3. Penyakit Jantung Hipertensif

Tekanan darah tinggi atau hipertensi berarti tekanan (tensi) tinggi di dalam *arteri*. Hipertensi tidak berarti kelebihan tekanan emosional, walaupun tekanan emosional dan stres dapat meningkatkan tekanan darah untuk jangka waktu pendek. Tekanan darah tinggi biasanya didefinisikan sebagai tingkat yang lebih dari 140/90 mmHg. Tekanan darah sistolik, yang merupakan angka di atas, menunjukkan tekanan dalam *arteri* sewaktu jantung berkontraksi dan memompakan darah ke dalam peredaran darah. Tekanan diastolik, yang merupakan angka di bawah, menunjukkan tekanan dalam *arteri* sewaktu jantung rileks setelah kontraksi. Oleh karenanya, tekanan darah diastolik menunjukkan tekanan minimum tereksposnya *arteri* (Anonim, 2012e).

### 4. *Kor Pulmonale*

Istilah *kor pulmonale*, atau *penyakit jantung pulmonalis*, digunakan untuk menjelaskan penyakit rongga jantung kanan akibat hipertensi *pulmonal* yang disebabkan oleh penyakit pembuluh darah paru atau *parenkim* paru (Kumar, 2007).

### 5. Penyakit Katup Jantung

Katup Jantung mengatur aliran darah melalui empat bilik jantung. Dua bilik atas yang kecil dan bulat merupakan serambi jantung, dan dua bilik bawah yang lebih besar dan berbentuk seperti kerucut merupakan

*ventrikel*. Katup jantung bisa gagal-fungsi karena bocor yang mengakibatkan regurgitasi (aliran balik). Katup jantung mungkin juga tidak membuka secara memadai, sehingga menyumbat aliran jantung, yang mengakibatkan stenosis (Anonim, 2012e).

#### 6. Penyakit *Miokardium* Primer

Penyakit *miokardium* primer ini adalah sekelompok penyakit yang beragam yang mencakup gangguan peradangan (*miokarditis*), penyakit imunologik (misal, demam reumatik), gangguan metabolik sistemik (misal, *hemokromatosis*), distrofi otot (misal, distrofi otot Duchene), dan kategori tambahan penyakit idiopatik yang disebut *kardiomiopati* (Kumar, 2007).

#### 7. Penyakit Jantung Konginetal

Penyakit jantung bawaan (PJB) atau *congenital heart disease* adalah suatu kelainan formasi dari jantung atau pembuluh besar dekat jantung. (Anonim, 2012f).

#### 8. Penyakit *perikardium*

*Perikardium* (selaput jantung) berfungsi sebagai benteng pertahanan dari jantung. *Perikardium* disebut juga sebagai kantung jantung, disamping melakukan pekerjaan dari jantung (fungsi jantung diwakilinya). Jika terdapat serangan dari luar yang menimbulkan sakit pada jantung, maka pericardium ini bertindak mewakilinya menderita.

## 9. Tumor jantung

Tumor adalah setiap jenis pertumbuhan yang tidak normal, baik bersifat kanker (*malignant*) atau bukan kanker (*benign*). Tumor yang dimulai di jantung disebut tumor primer yang bisa terbentuk di setiap jaringan jantung dan bisa bersifat kanker atau bukan kanker. Tumor jantung primer langka, terjadi kurang dari 1 berbanding 2000 orang (Anonim, 2012g).