

# **PENGANTAR CODING BERBASIS C/C++**

**ASRUL HUDA**

**NOPER ARDI**

**AKRIMULLAH MUBAI**

UNDANG-UNDANG REPUBLIK INDONESIA  
NO 19 TAHUN 2002  
TENTANG HAK CIPTA  
PASAL 72  
KETENTUAN PIDANA SANGSI PELANGGARAN

1. Barang siapa dengan sengaja dan tanpa hak mengumumkan atau memperbanyak suatu Ciptaan atau memberi izin untuk itu, dipidana dengan pidana penjara paling singkat 1 (satu) bulan dan denda paling sedikit Rp 1.000.000, 00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan denda paling banyak Rp 5.000.000.000, 00 (lima milyar rupiah)
2. Barang siapa dengan sengaja menyerahkan, menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu Ciptaan atau barang hasil pelanggaran Hak Cipta atau Hak Terkait sebagaimana dimaksud dalam ayat (1), dipidana dengan pidana penjara paling lama 5 (lima) tahun dan denda paling banyak Rp 500.000.000, 00 (lima ratus juta rupiah).

# **PENGANTAR CODING BERBASIS C/C++**

**ASRUL HUDA  
NOPER ARDI  
AKRIMULLAH MUBAI**



**2021**

# **PENGANTAR CODING BERBASIS C/C++**

editor, Tim editor UNP Press  
Penerbit UNP Press, Padang, 2021  
1 (satu) jilid; 17.6 x 25 cm (B5)  
Jumlah halaman ix + 193

ISBN : 978-602-1178-87-4

---

Hak Cipta dilindungi oleh undang-undang pada penulis  
Hak penerbitan pada UNP Press

---

Penyusun: Dr. Asrul Huda, S.Kom., M.Kom, Noper Ardi S.Pd., M.Eng,  
Akrimullah Mubai, S.Pd., M.Pd.T  
Editor Substansi: Tim UNP Press  
Editor Bahasa: Prof. Dr. Harris Effendi Thahar, M.Pd  
Desain Sampul & Layout: Tim UNP Press

## KATA PENGANTAR

Puji syukur selalu kita panjatkan kehadiran Allah Tuhan Yang Maha Esa, karena atas rahmat-Nya penulis dapat menyelesaikan buku berjudul “Pengantar Coding Berbasis C/C++”. Sholawat serta salam tak lupa pula dikirimkan pada junjungan Nabi Muhammad ﷺ yang telah membawa kita dari zaman Jahiliyah yang penuh kegelapan ke zaman Islamiyah yang terang benderang serta dalam keadaan berilmu pengetahuan.

Buku ini disusun untuk menambah pemahaman dalam pembelajaran Pengantar Coding. Topik-topik yang dikemukakan berguna dalam pembelajaran teori Pengantar Coding Berbasis C/C++. Buku Pengantar Coding Berbasis C/C++ ini terdiri dari 11 Bab meliputi: Dasar-dasar Algoritma, Tipe data, identifier dan operator, Input dan Output, Struktur Kontrol - Selection, Struktur Kontrol - Looping, Fungsi bawaan untuk pengolahan String, Prosedur, Fungsi, Array, Konsep Searching dan Konsep Sorting.

Dengan adanya buku ini diharapkan dapat membantu dan memberikan manfaat dalam memahami Pengantar Coding Berbasis C/C++. Buku ini masih jauh dari sempurna, untuk itu kritik dan saran dari pembaca akan sangat membantu untuk meningkatkan kualitas buku Pengantar Coding Berbasis C/C++ ini. Terima Kasih.

Padang, 2 Desember 2021

Penulis

## DAFTAR ISI

<b>KATA PENGANTAR .....</b>	<b>V</b>
<b>DAFTAR ISI .....</b>	<b>VI</b>
<b>DAFTAR GAMBAR .....</b>	<b>VIII</b>
<b>DAFTAR TABEL .....</b>	<b>IX</b>
<b>BAB 1 DASAR-DASAR ALGORITMA .....</b>	<b>1</b>
KEGIATAN BELAJAR 1 : DASAR-DASAR ALGORITMA .....	1
1.    DEFINISI ALGORITMA .....	1
2.    ATURAN PENULISAN ALGORITMA .....	4
<b>BAB 2 TIPE DATA, IDENTIFIER, DAN OPERATOR .....</b>	<b>20</b>
KEGIATAN BELAJAR 2 & 3 : TIPE DATA, IDENTIFIER, DAN OPERATOR	20
1.    SEJARAH BAHASA C/C++ .....	20
2.    VERSI BAHASA C .....	25
3.    STRUKTUR BAHASA C++ .....	26
<b>BAB 3 INPUT DAN OUTPUT .....</b>	<b>45</b>
KEGIATAN BELAJAR 4 : PENANGANAN INPUT DAN OUTPUT PADA PROGRAM .....	45
1.    STREAM .....	45
2.    CONSOLE .....	47
3.    OUTPUT .....	48
4.    INPUT .....	49
<b>BAB 4 STRUKTUR KONTROL (KONSEP PEMILIHAN).....</b>	<b>53</b>
KEGIATAN BELAJAR 5 : STRUKTUR KONTROL (PEMILIHAN IF, SWITCH) .....	53
1.    STATEMENT .....	54
2.    MACAM-MACAM FLOW CONTROL .....	55
<b>BAB 5 STRUKTUR KONTROL (KONSEP PERULANGAN) .....</b>	<b>70</b>
KEGIATAN BELAJAR 6 & 7 : STRUKTUR KONTROL (KONSEP PERULANGAN) .....	70
1.    PERNYATAAN PENGULANGAN ( <i>LOOP</i> ) .....	70
2.    PERNYATAAN LOMPATAN ( <i>JUMP</i> ) .....	76
3.    PENGECUALIAN ( <i>EXCEPTIONS</i> ) .....	85
<b>BAB 6 FUNGSI BAWAAN UNTUK MENGOLAH STRING.....</b>	<b>88</b>
KEGIATAN BELAJAR 9 : FUNGSI BAWAAN UNTUK MENGOLAH STRING .....	88
1.    KONSTANTA DAN VARIABEL TEKS/STRING .....	88
2.    OPERASI INPUT OUTPUT STRING .....	90
3.    MENGAKSES ELEMEN STRING .....	93

4. FUNGSI-FUNGSI PENGOLAHAN TEKS/STRING .....	95
<b>BAB 7 PROSEDUR.....</b>	<b>102</b>
KEGIATAN BELAJAR 10 : PROSEDUR.....	102
1. DEFINISI PROSEDUR.....	102
2. KONSEP DIVIDE AND COHESIVE .....	104
3. FUNGSI .....	113
<b>BAB 8 FUNGSI.....</b>	<b>115</b>
KEGIATAN BELAJAR 11 : FUNGSI DALAM C++.....	115
1. DEFINISI FUNGSI.....	115
2. DEKLARASI FUNGSI.....	117
3. PARAMETER.....	117
<b>BAB 9 ARRAY .....</b>	<b>134</b>
KEGIATAN BELAJAR 12 & 13 : KONSEP ARRAY DALAM C++ .....	134
1. MACAM-MACAM ARRAY.....	135
2. DEKLARASI ARRAY .....	137
3. INISIALISASI NILAI ARRAY .....	137
4. MENGAKSES NILAI ARRAY DENGAN PERULANGAN.....	139
<b>BAB 10 SEARCHING.....</b>	<b>145</b>
KEGIATAN BELAJAR 14 : KONSEP SEARCHING DALAM C/C++ .....	145
1. SEQUENTIAL SEARCH .....	146
2. BINARY SEARCH.....	154
3. INTERPOLATION SEARCH .....	157
<b>BAB 11 SORTING .....</b>	<b>161</b>
KEGIATAN BELAJAR 15 : KONSEP PENGURUTAN (SORTING).....	161
1. BUBBLESORT.....	163
2. QUICK SORT .....	167
3. MAXIMUM DAN MINIMUM SORT .....	169
4. METODE SHELL SORT .....	181
5. METODE MERGE SORT .....	184
<b>DAFTAR PUSTAKA .....</b>	<b>195</b>
<b>GLOSARIUM.....</b>	<b>198</b>
<b>INDEKS.....</b>	<b>191</b>
<b>INTISARI RINGKASAN BUKU.....</b>	<b>193</b>

## DAFTAR GAMBAR

Gambar 1. 1 Proses Penyelesaian masalah dalam algoritma.....	3
Gambar 1. 2 Algoritma kalimat deskriptif .....	8
Gambar 1. 3 Simbol-simbol flowchart.....	10
Gambar 1. 4 Contoh flowchart operasi bilangan.....	11
Gambar 1. 4 Contoh flowchart Memasak Nasi Goreng .....	12
Gambar 1. 6 Trapesium.....	15
Gambar 1. 7 Persegi Panjang .....	15
Gambar 1. 8 Segitiga.....	16
Gambar 1. 9 Lingkaran.....	17
Gambar 2. 1 Dennis Ritchie dan Ken Thompson.....	21
Gambar 2. 2 Bahasa PHP dikembangkan dari bahasa C dan C++ .....	24
Gambar 2. 3 Lingkup variabel.....	34
Gambar 3. 1 Proses input output dalam bahasa C.....	47
Gambar 4. 1 Flowchart Selection.....	57
Gambar 5. 1 Flowchart Penrulangan (Looping).....	71
Gambar 5. 2 Flowchart do-while.....	74
Gambar 6. 1 Contoh String .....	89
Gambar 6. 2 Variabel string dan data string.....	91
Gambar 7. 2 struktur prosedur sederhana.....	102
Gambar 7. 3 Skema algoritma divide & conquer .....	106
Gambar 7. 4 cara mendeklarasikan Fungsi dalam C++.....	110
Gambar 8. 1 Konsep pemrograman Modular .....	115
Gambar 9. 1 array.....	135
Gambar 9. 2 Inisialisasi array LARIK .....	138
Gambar 10. 1 MergeSort.....	184



## DAFTAR TABEL

Tabel 2. 1 key word c++ .....	31
Tabel 2. 2 Tipe data.....	32
Tabel 2. 3 karakter khusus dalam C .....	36
Tabel 2. 3 Operator Relasional.....	38
Tabel 2. 4 Operator Logika .....	39
Tabel 2. 5 Operator Bitwise .....	40
Tabel 2. 6 Prioritas Operator .....	41
Tabel 9. 1 Contoh Sequential Search .....	147
Tabel 10. 1 Contoh BubbleSort.....	164

# **BAB 1**

## **Dasar-Dasar Algoritma**

### **Kegiatan Belajar 1 : Dasar-dasar Algoritma**

#### **A. Tujuan**

Setelah mempelajari proses perancangan Desain Grafis, diharapkan mahasiswa dapat:

1. Menjelaskan Algoritma
2. Menjelaskan Struktur Algoritma
3. Menjelaskan jenis-jenis penulisan algoritma
4. Membuat Algoritma sederhana untuk menyelesaikan kasus-kasus sederhana

#### **B. Uraian Materi**

Komputer pada dasarnya adalah mesin yang tidak bisa apa-apa. Kita harus memberikan perintah untuk dapat berbicara (berkomunikasi) dengan komputer, dengan cara memberikan serangkaian instruksi kepada komputer agar komputer dapat memecahkan masalah. Langkah-langkah yang kita lakukan dalam memberikan instruksi untuk memecahkan masalah kita kita namakan pemrograman komputer. Untuk menyusun sebuah program yang besar dan kompleks, pemrograman membutuhkan tahapan penyusunan yang sistematis dan terpadu, yaitu:

1. Definisi Masalah
2. Analisis Kebutuhan
3. Penyusunan Algoritma
4. Pengkodean/Pemrograman
5. Testing dan Debugging
6. Pemeliharaan
7. Dokumentasi

#### **1. Definisi Algoritma**

Algoritma adalah kunci dari bidang ilmu komputer, dan pada dasarnya setiap hari kita melakukan aktivitas algoritma. Kata algoritma berasal dari kata algorism dan ritmis (Abu Abdullah Muhammad Ibn Musa Al Khwarizmi (825M) dalam

buku Al-jabr wa-al muqabla, ahli matematika Uzbekistan). Algoritma adalah urutan langkah-langkah berhingga untuk memecahkan masalah logika atau matematika. Algoritma adalah logika, metode dan tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan. Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis. Algoritma adalah urutan logis pengambilan keputusan untuk pemecahan masalah. Berdasarkan beberapa definisi tersebut, dapat kita simpulkan algoritma sebagai:

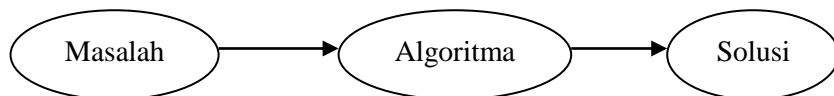
“Suatu algoritma harus menghasilkan output yang tepat guna (efektif) dalam waktu yang relatif singkat dan penggunaan memori yang relatif sedikit (efisien) dengan langkah yang berhingga dan prosesnya berakhir baik dalam keadaan diperoleh suatu solusi ataupun tidak adanya solusi.”

Adapun kriteria algoritma yang baik adalah sebagai berikut:

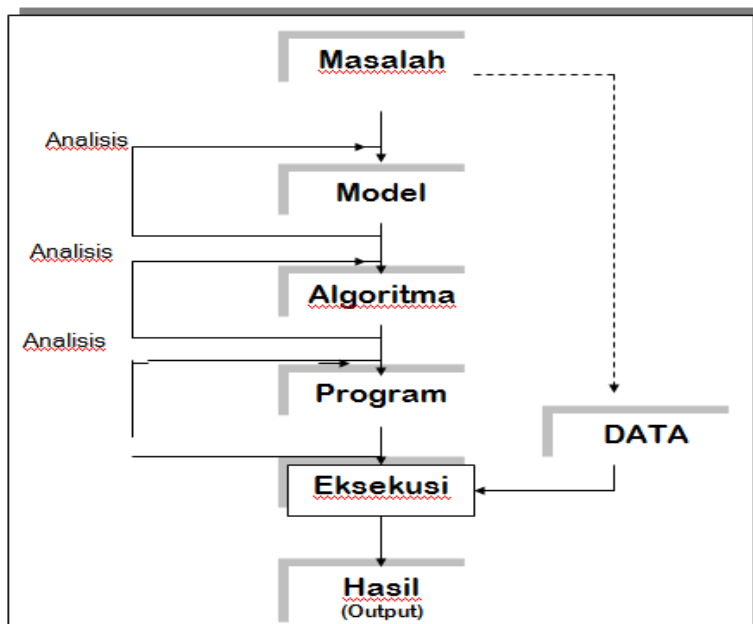
1. Tepat, benar, sederhana, standar dan efektif
2. Logis, terstruktur dan sistematis
3. Semua operasi terdefinisi
4. Semua proses harus berakhir setelah sejumlah langkah dilakukan
5. Ditulis dengan bahasa yang standar dengan format pemrograman agar mudah untuk diimplementasikan dan tidak menimbulkan arti ganda.

Dalam urutannya dalam penyelesaian masalah, urutan algoritma dapat kita buat sebagai berikut:

Proses Penyelesaian Masalah



Secara lebih detail urutannya dapat dilihat pada gambar berikut:



Gambar 1. 1 Proses Penyelesaian masalah dalam algoritma

Algoritma penting untuk dipelajari sebagai dasar dari pemrograman karena algoritma adalah jantung ilmu komputer atau informatika. Banyak cabang ilmu komputer yang diacu dalam terminologi algoritma tetapi algoritma tidak selalu identik dengan ilmu komputer namun kehidupan sehari-hari terdapat proses yang dinyatakan dalam suatu Algoritma. Komputer hanyalah salah satu pemroses. Agar dapat dilaksanakan oleh komputer algoritma harus ditulis dalam notasi bahasa pemrograman sehingga dinamakan program. Belajar memprogram tidak sama dengan belajar bahasa pemrograman. Belajar memprogram adalah belajar tentang metodologi pemecahan masalah. Belajar bahasa pemrograman adalah belajar memakai bahasa aturan-aturan tata bahasanya, instruksi-instruksinya, tata cara pengoperasian compilernya, dan manfaat instruksi-instruksi tersebut untuk membuat program yang ditulis dalam bahasa itu saja.

Penanda-penanda tersebut adalah:

- 1) Awal/akhir program (terminal)

- 2) Suatu proses dalam program (process)
- 3) Input/output
- 4) Proses yang telah terdefinisi (Predefined Process)
- 5) Pengambilan keputusan (decision)

## **2. Aturan penulisan Algoritma**

### **a. Kalimat Deskriptif**

Notasi penulisan algoritma menggunakan bahasa deskriptif biasa juga disebut dengan notasi alami. Penulisan Algoritma dengan cara ini dilakukan dengan menuliskan instruksi-instruksi yang harus dilaksanakan secara berurutan dalam bentuk uraian kalimat deskriptif dengan menggunakan bahasa yang jelas.

Dasar dari notasi bahasa deskriptif adalah Bahasa Inggris, namun dapat dimodifikasi dengan bahasa sehari-hari termasuk Bahasa Indonesia. Karena tidak ada aturan baku dalam menuliskan algoritma dengan notasi deskriptif maka tiap orang dapat membuat aturan penulisan dan notasi algoritma sendiri. Hal ini dapat dimengerti karena teks algoritma tidak sama dengan teks program. Program adalah implementasi algoritma dalam notasi bahasa pemrograman tertentu. Namun, agar notasi algoritma mudah ditranslasi ke dalam notasi bahasa pemrograman, maka sebaiknya notasi algoritma tersebut berkoresponden dengan notasi bahasa pemrograman pada umumnya. Kata kerja adalah jenis kata yang biasa digunakan dalam penulisan bahasa deskriptif, contohnya tulis, baca, hitung, tampilkan, ulangi, bandingkan, dan lain-lain.

Notasi jenis ini cocok untuk algoritma yang pendek. Tapi untuk masalah algoritma yang panjang, notasi ini kurang efektif. Cara penulisan algoritma dengan notasi bahasa deskriptif paling mudah dibuat, namun demikian cara ini paling sulit untuk diterjemahkan ke dalam bahasa pemrograman. Pada dasarnya teks algoritma dengan bahasa deskriptif disusun oleh tiga bagian utama yaitu:

- 1) Bagian judul (header)
- 2) Bagian deklarasi (kamus)
- 3) Bagian deskripsi

Setiap bagian disertai dengan komentar untuk memperjelas maksud teks yang dituliskan. Komentar adalah kalimat yang diapit oleh pasangan tanda kurung kurawal ('{' dan '}').

### **Judul Algoritma**

Merupakan bagian yang terdiri atas nama algoritma dan penjelasan (spesifikasi) tentang algoritma tersebut. Dibagian ini juga digunakan untuk menentukan apakah teks algoritma yang dibuat tersebut adalah program, prosedur, atau fungsi. Nama algoritma sebaiknya singkat namun cukup menggambarkan apa yang dilakukan oleh algoritma tersebut.

Di bawah nama algoritma disertai dengan penjelasan singkat (intisari) tentang apa yang dilakukan oleh algoritma. Penjelasan dibawah nama algoritma sering dinamakan juga spesifikasi algoritma yang dituliskan dalam kurung kurawal ({}). Algoritma harus ditulis sesuai dengan spesifikasi yang didefinisikan.

Contoh menghitung luas lingkaran yang disertai dengan penjelasan singkat.

```
Algoritma Luas_Lingkaran <- Judul algoritma
{Menghitung luas lingkaran untuk ukuran jari-
jari tertentu. Algoritma menerima masukan jari-
jari lingkaran, menghitung luasnya, lalu cetak
luasnya ke piranti keluaran} <- Spesifikasi
```

### **Bagian Deklarasi**

Di dalam algoritma, deklarasi atau kamus adalah bagian untuk mendefinisikan semua nama yang dipakai di dalam algoritma. Nama tersebut dapat berupa nama variabel, nama konstanta, nama tipe, nama prosedur atau nama fungsi. Semua nama tersebut baru dapat digunakan di dalam algoritma jika telah didefinisikan terlebih dahulu didalam bagian deklarasi. Penulisan sekumpulan nama dalam bagian deklarasi sebaiknya dikelompokkan menurut jenisnya. Pendefinisian nama konstanta sekaligus memberikan nilai konstanta. Pendefinisian nama

fungsi atau prosedur sekaligus dengan pendefinisian spesifikasi dan parameternya. berikut adalah bentuk umum bagian deklarasi.

**Deklarasi :**

```
{Nama Type variabel yang bukan tipe data dasar}
type waktu:<hh:mm:ss: integer> {Type waktu
terdiri dari 3 data masukan yaitu "hh" sebagai
jam, "mm" sebagai menit, dan "ss" sebagai detik}

{Nama konstanta, harus menyebutkan tipe data dan
nilai}
constant PHI: real = 3.141592653589793
constant E: real = 2.718281828459045

{Nama variabel yang menggunakan tipe data dasar}
nama : String {suatu nilai yang merupakan
kumpulan karakter}
ketemu : boolean {suatu nilai logika (true
atau false)}
beratBadan : real {suatu nilai bilangan
pecahan}
jumlahAnak : integer {suatu nilai bilangan
bulat}

{Nama fungsi, mendefinisikan domain, nama,
jumlah, tipe dan jumlah parameter, serta tipe
data keluaran}
function tambah(x:int, y:int): int {proses:
menambahkan dua nilai data dan hasil penambahan
sebagai nilai keluaran fungsi}
```

Berikut ini adalah contoh bagian deklarasi algoritma menghitung luas lingkaran:

**Deklarasi :**

```
jari_jari = real { tipe data bilangan  
pecahan }  
luas = real { tipe data bilangan pecahan }  
PHI = 3.14
```

**Bagian deskripsi.**

Deskripsi adalah bagian inti dari struktur algoritma. Bagian ini berisi uraian langkah-langkah penyelesaian masalah. Langkah-langkah ini dituliskan dengan notasi yang lazim dalam penulisan algoritma. Setiap langkah algoritma dibaca dari langkah paling atas hingga langkah paling bawah. Urutan penulisan menentukan urutan pelaksanaan perintah. Seperti telah dijelaskan di bab satu bahwa penyusun atau struktur dasar algoritma adalah langkahlangkah. Suatu Algoritma dapat terdiri dari tiga struktur dasar, yaitu runtunan, pemilihan dan pengulangan. Ketiga jenis langkah tersebut membentuk konstruksi suatu algoritma. Pada bagian deskripsi inilah letak tiga struktur algoritma tersebut.

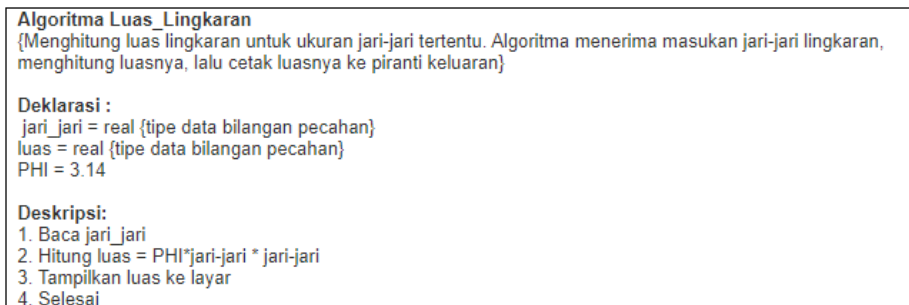
berikut adalah adalah contoh bagian deskripsi algoritma menghitung luas lingkaran.

**Deskripsi:**

1. Baca jari\_jari
2. Hitung luas =  $PHI * jari-jari * jari-jari$
3. Tampilkan luas ke layar
4. Selesai

Berikut adalah contoh algoritma menghitung luas lingkaran yang dituliskan menggunakan kalimat deskriptif secara lengkap.





Gambar 1. 2 Algoritma kalimat deskriptif

## b. Flowchart

Flowchart adalah adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program. Flowchart merupakan penjabaran urutan proses yang akan dikerjakan oleh program yang akan kita rancang dalam bentuk diagram (grafis). Tiap-tiap penanda dalam program mempunyai simbol dalam flowchart. Flowchart atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah.

Flowchart berperan penting dalam memutuskan sebuah langkah atau fungsionalitas dari sebuah proyek pembuatan program yang melibatkan banyak orang sekaligus. Selain itu dengan menggunakan bagan alur proses dari sebuah program akan lebih jelas, ringkas, dan mengurangi kemungkinan untuk salah penafsiran. Penggunaan flowchart dalam dunia pemrograman juga merupakan cara yang bagus untuk menghubungkan antara kebutuhan teknis dan non-teknis.

Fungsi utama dari flowchart adalah memberi gambaran jalannya sebuah program dari satu proses ke proses lainnya. Sehingga, alur program menjadi mudah dipahami oleh semua orang. Selain itu, fungsi lain dari flowchart adalah untuk menyederhanakan rangkaian prosedur agar memudahkan pemahaman terhadap informasi tersebut.

Flowchart sendiri terdiri dari lima jenis, masing-masing jenis memiliki karakteristik dalam penggunaannya. Berikut adalah jenis-jenisnya:

1) Flowchart dokumen

Pertama ada flowchart dokumen (document flowchart) atau bisa juga disebut dengan paperwork flowchart. Flowchart dokumen berfungsi untuk menelusuri alur form dari satu bagian ke bagian yang lain, termasuk bagaimana laporan diproses, dicatat, dan disimpan.

2) Flowchart program

Selanjutnya kita akan membahas flowchart program. Flowchart ini menggambarkan secara rinci prosedur dari proses program. Flowchart program terdiri dari dua macam, antara lain: flowchart logika program (program logic flowchart) dan flowchart program komputer terinci (detailed computer program flowchart).

3) Flowchart proses

Flowchart proses adalah cara penggambaran rekayasa industrial dengan cara merinci dan menganalisis langkah-langkah selanjutnya dalam suatu prosedur atau sistem.

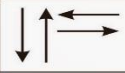




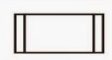










4) Flowchart sistem

Yang keempat ada flowchart sistem. Flowchart sistem adalah flowchart yang menampilkan tahapan atau proses kerja yang sedang berlangsung di dalam sistem secara menyeluruh. Selain itu flowchart sistem juga menguraikan urutan dari setiap prosedur yang ada di dalam sistem.

5) Flowchart skematik

Terakhir ada flowchart skematik. Flowchart ini menampilkan alur prosedur suatu sistem, hampir sama dengan flowchart sistem. Namun, ada perbedaan dalam penggunaan simbol-simbol dalam menggambarkan alur. Selain simbol-simbol, flowchart skematik juga menggunakan gambar-gambar komputer serta peralatan lainnya untuk mempermudah dalam pembacaan flowchart untuk orang awam.

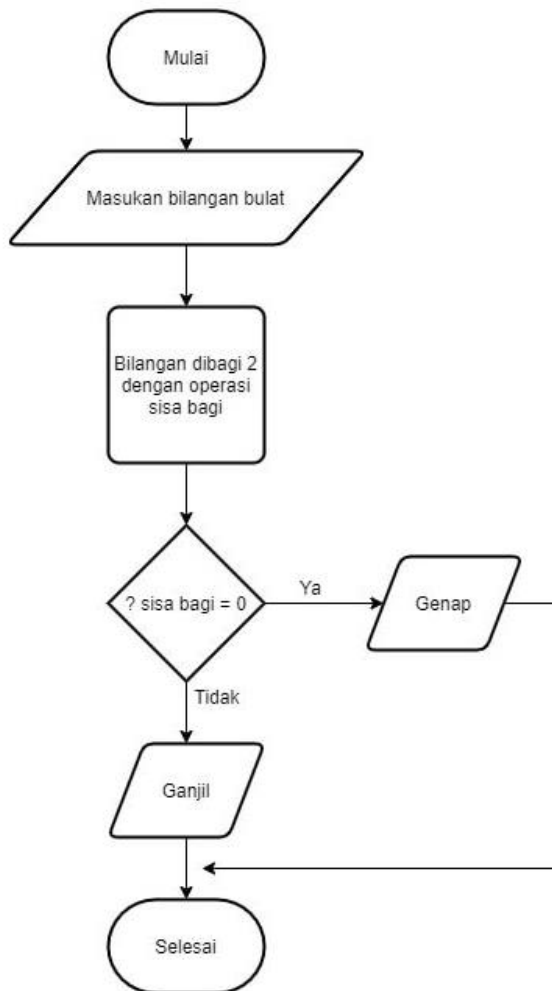
Simbol-simbol flowchart dapat dilihat pada gambar berikut:

	<b>Flow Direction symbol</b> Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		<b>Simbol Manual Input</b> Simbol untuk pemasukan data secara manual on-line keyboard
	<b>Terminator Symbol</b> Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		<b>Simbol Preparation</b> Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	<b>Connector Symbol</b> Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		<b>Simbol Predefine Proses</b> Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	<b>Connector Symbol</b> Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		<b>Simbol Display</b> Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	<b>Processing Symbol</b> Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		<b>Simbol disk and On-line Storage</b> Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	<b>Simbol Manual Operation</b> Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer		<b>Simbol magnetik tape Unit</b> Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	<b>Simbol Decision</b> Simbol pemilihan proses berdasarkan kondisi yang ada.		<b>Simbol Punch Card</b> Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	<b>Simbol Input-Output</b> Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		<b>Simbol Dokumen</b> Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

Gambar 1. 3 Simbol-simbol flowchart

Simbol-simbol pada gambar di atas memiliki jenis dan fungsi yang berbeda-beda. Ada yang berfungsi untuk menghubungkan satu simbol dengan simbol lainnya seperti simbol flow, on-page dan off-page reference. Selain itu ada juga simbol yang berfungsi untuk menunjukkan suatu proses yang sedang berjalan, dan yang terakhir terdapat simbol yang berfungsi untuk memasukan input dan menampilkan output.

Berikut adalah contoh pembuatan flowchart:



Gambar 1. 4 Contoh flowchart operasi bilangan

Pembahasan:

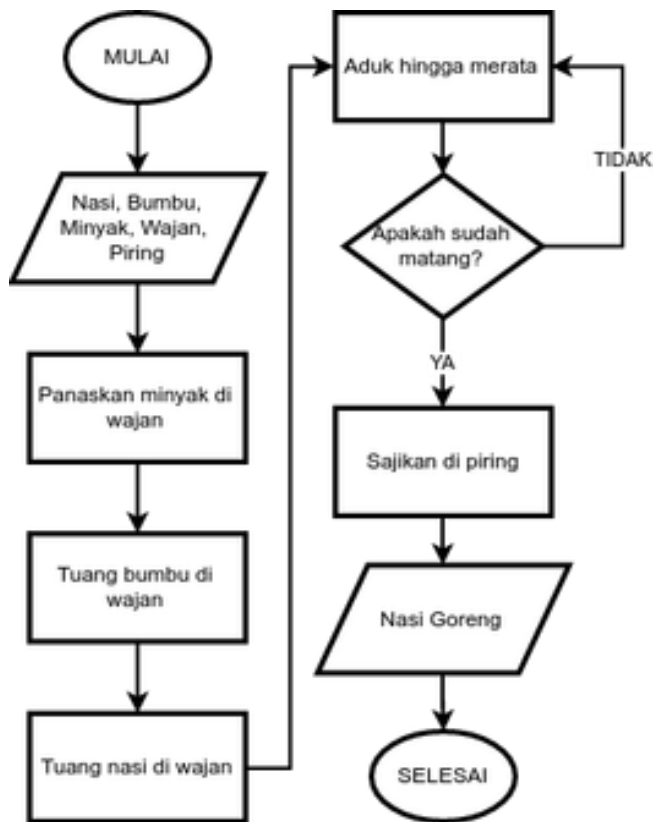
Pertama pengguna menginput data yang berupa nilai dari bilangan bulat.

Kemudian nilai yang dimasukan diproses dengan cara dibagi dengan angka 2.

Jika sisa bagi sama dengan 0 berarti bilangan yang dimasukan adalah bilangan genap.

Jika sisa bagi tidak sama dengan 0 berarti bilangan yang dimasukan adalah bilangan ganjil.

Selesai.



Gambar 1. 5 Contoh flowchart Memasak Nasi Goreng

Flowchart diatas merupakan flowchart dari algoritma membuat nasi goreng. Setiap urutan proses pembuatan nasi goreng dibuat secara runtut dari awal proses sampai akhir proses.

### c. Pseudocode

Menurut bahasa Pseudocode merupakan gabungan kata dari kata Pseudo yang artinya palsu, imitasi, menyerupai, atau mirip, dan kata code yang berarti sebuah kode pemrograman. Dalam bahasa pemrograman hal yang harus ada adalah sebuah atauran penulisan kode atau yang biasa kita sebut dengan syntax. Karena dalam pseudocode tidak memiliki aturan yang

spesifik/baku dalam penulisan kode, jadi pseudocode tidak termasuk ke dalam bahasa pemrograman.

Adapun pengertian dari pseudocode adalah sebuah deskripsi atau ringkasan tingkat tinggi dari algoritma pemrograman komputer dengan konvensi struktural. Dengan menuliskan kode semu ini kita bisa menuliskan algoritma agar bisa dibaca oleh manusia, dan kode ini tidak ditunjukkan untuk robot/mesin. Pseudocode adalah cara penulisan algoritma yang menyerupai bahasa pemrograman tingkat tinggi. Pseudocode menggunakan bahasa yang hampir menyerupai bahasa pemrograman. Biasanya pseudocode menggunakan bahasa yang mudah dipahami secara universal dan juga lebih ringkas dari pada cara penulisan algoritma sebelumnya. Pseudocode bisa membantu kita dalam proses coding yang berbasis pada bahasa pemrograman seperti Pascal, C++, Basic, dan lainnya.

Pseudocode adalah deskripsi dari algoritma pemrograman komputer yang menggunakan konvensi struktural dari suatu bahasa pemrograman, dan ditujukan agar dapat dibaca oleh manusia dan bukan oleh mesin.

Tujuan pseudocode adalah agar manusia dapat dengan mudah dalam pemahaman dibandingkan dengan menggunakan bahasa pemrograman yang umumnya digunakan.

Pseudocode lebih mudah untuk dipahami jika dibandingkan dengan bahasa pemrograman yang ada. Berikut fungsi dan juga tujuan lain penulisan menggunakan pseudocode:

- 1) Mempermudah penggunaannya dalam memahami dan memperjelas cara menyelesaikan suatu masalah.
- 2) sebagai alat dokumentasi.
- 3) Membantu penggunaannya dalam menuliskan suatu algoritma yang dibuatnya.
- 4) Mudah karena tidak bergantung pada suatu sistem tertentu dan juga sapek dari Pseudocode termasuk ringkas, dan itulah yang merupakan perinsip utama dari algoritma

Ciri-ciri Pseudocode adalah sebagai berikut:

- 1) Pseudocode merupakan sebuah tanda/notasi/kode yang mirip, dan juga merupakan penjelasan untuk cara penyelesaian suatu masalah.
- 2) Pseudocode juga sering digunakan para penggunanya untuk menuliskan suatu algoritma dari suatu permasalahan.
- 3) Pseudocode berisi step-step (langkah-langkah) untuk menyelesaikan suatu masalah (seperti halnya algoritma), akan tetapi bentuk dari masalahnya itu sedikit berbeda dari algoritma.
- 4) Pseudocode tidak memiliki aturan baku yang mengikat penggunanya.
- 5) Pseudocode merupakan bahasa yang mirip dengan bahasa pemrograman, akan tetapi Pseudocode menggunakan bahasa yang mudah dipahami secara universal, dan juga lebih ringkas dari pada algoritma.

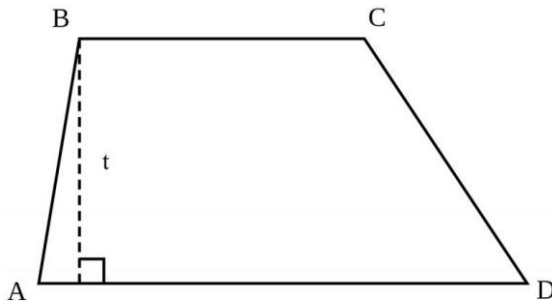
Ada 3 struktur yang membentuk sebuah Pseudocode yaitu:

- 1) Judul (berisi judul algoritma)
- 2) Deskripsi (berisi deklarasi variabel dan konstanta)
- 3) Implementasi (berisi inti algoritma)

ke tiga struktur ini harus berurut mulai dari judul kemudian deskripsi yaitu isinya, dan implementasi yang didalamnya merupakan bagian inti dari algoritma itu.

Berikut adalah contoh penulisan pseudocode pemrogramana:

- 1) Pseudocode Menghitung Luas Trapesium



Gambar 1. 6 Trapesium

kita tahu bahwasanya rumus keliling dari Trapesium adalah  $K = AB + BC + CD + AD$  sedangkan rumus luasnya adalah  $L = (AB + CD) \times t / 2$ . Misalkan pada gambar diatas diketahui:

$$\begin{aligned} AB &= 10 \\ BC &= 6 \\ CD &= AD = 8 \\ t &= 7 \end{aligned}$$

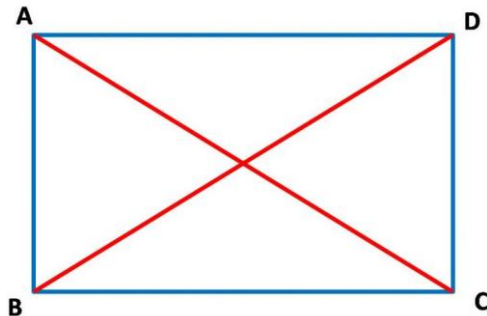
Maka dalam penulisan Pseudocode sebagai berikut:

```

program hitung_luas_trapesium deklarasi
var luas, ab, cd, t:integer; algoritma:
ab <-- 10; cd <-- 8; t <-- 7; luas <--
1/2 * ab * cd * t; write <-- luas ;

```

## 2) Pseudocode Menghitung Luas Persegi Panjang



Gambar 1. 7 Persegi Panjang



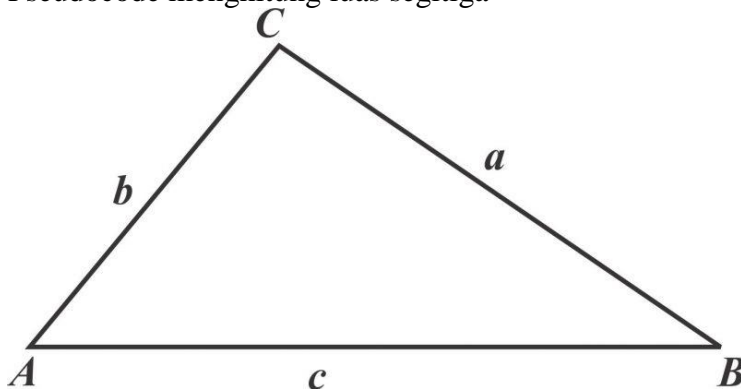
Dalam hal ini kita akan menghitung pseudocode dengan rumus persegi panjang , rumus luas persegi panjang adalah  $L = p \times l$  sedangkan rumus kelilingnya adalah  $K = 2 \times (p + l)$ . Dalam Algoritma Pseudocode menghitung luas persegi panjang, maka panjang dan lebarnya diinput dan luasnya di tampilkan.

```

program hitung_luas_segi_panjang
deklarasi var panjang, lebar, luas:integer;
algoritma: read(panjang); read(lebar);
luas <-- panjang * lebar; write(luas);

```

### 3) Pseudocode menghitung luas segitiga



Gambar 1. 8 Segitiga

Bangun datar segitiga memiliki rumus luas yaitu  $L_{\Delta ABC} = \frac{1}{2} \cdot \text{alas} \cdot \text{tinggi}$  , drngan luas ini kita bisa membuat algoritma pseudocode pada luas sigitiga, dan kita harus mengetahui tinggi dan alas segitiga tersebut.

sebagai contoh perhitungan luas segitiga yang akan kita buat menjadi algoritma pseudocode, diketahui suatu bangun datar segitiga memiliki alas = 10 dan tinggi 15, berapa luas segitiga tersebut?

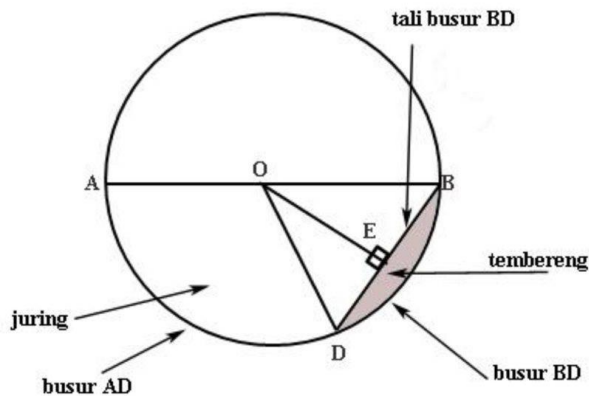
maka dalam bahasa algoritma pseudocode adalah sebagai berikut:

```

program hitung_luas_segitiga deklarasi var
luas,alas,tinggi:integer; algoritma: alas
<-- 10; tinggi <-- 15; luas <-- 1/2 * alas
* tinggi write(luas)

```

#### 4) Pseudocode Menghitung luas lingkaran



Gambar 1. 9 Lingkaran

Rumus keliling dan luas lingkaran adalah  $L = \pi \times d^2/4 = \pi \times r^2$ , sedang kan untuk kelilingnya adalah  $K = \pi \times d = 2 \times \pi \times r$ , untuk menghitung diameter lingkaran kita bisa menghitungnya dengan rumus  $D = 2 \times r$ .

Untuk membuat Algoritma Pseudocode dalam menghitung luas lingkaran kita bisa menggunakan pendekatan bahasa pemrograman pascal, yaitu jari-jari lingkaran dimasukkan oleh penggunaanya dan hasilnya dicetak. sebagai berikut:

```

program hitung_luas_lingkaran deklarasi
var phi : float; var r,luas:integer;
algoritma: phi <-- 3.14; read(r); {diinput
user} luas <-- phi * r *r; write(luas);

```

#### 5) Pseudocode Menghitung

Berikut akan dibuat pseudocode dengan menginput 3 bilangan dan menentukan bilangan terbesar, terkecil, dan rata-ratanya. berikut bentuk algoritma dalam Pseudocode:

```
If (A>B ; A>C) then Print "A paling
besar" If (B>A ; B>C) then Print "B
paling besar" Else Print "C terkecil"
End if
```

#### 6) Pseudocode Mengubah satuan waktu

Berikut akan dibuat waktu (detik, menit, dan jam) dimana dari detik ke stuan jam, dan ke satuan menit. untuk membuat Algoritma Pseudocode kita bisa memasukkan detik, menit = 60 detik, jam = 3600 detik, tampilkan menit, dan tampilkan jam.

```
Read detik M = 60 s H = 3600 s Output
M Output H
```

#### 7) Pseudocode Menentukan Bilangan Ganjil dan Genap

Sebelum kita membuat tulisan pseudocode, hal yang harus dilakukan adalah menginput sebuah bilangan, bagi bilangan yang tadi dimasukkan dengan angka dua, jika bilangannya menghasilkan sisa pembagian nol maka itu adalah genap, sedangkan jika bilangan tidak menghasilkan sisa pembagian 0 maka itu adalah bilangan ganjil.

```
Read Bilangan If bil mod 2 = 0 then,
"Output Genap" Else "Output Ganjil" End if
```

### C. Rangkuman

Ada beberapa cara dalam penulisan atau notasi sebuah algoritma

1. Notasi algoritma deskriptif dilakukan dengan cara menuliskan intruksi-intruksi yang musti dilaksanakan dalam bentuk untaian kalimat deskriptif dengan menggunakan bahasa yang jelas. Notasi deskriptif ini disarankan untuk algoritma yang pendek karena apabila untuk algoritma yang panjang notasi deskriptif kurang efektif.
2. Flowchart adalah analisis langkah langkah dari suatu proses berupa , sedangkan diagram alur menjelaskan langkah-langkah suatu program dengan cara grafis. Algoritme dan diagram alur

membantu memperjelas semua langkah untuk memecahkan masalah.

3. Pseudocode adalah istilah yang sering digunakan dalam pemrograman dan bidang berbasis algoritma. Ini adalah metodologi yang memungkinkan programmer untuk mewakili implementasi dari suatu algoritma. Sederhananya, kita dapat mengatakan bahwa itu adalah representasi algoritme yang sudah matang.

#### **D. Tes essay**

1. Jelaskan apa yang dimaksud dengan algoritma!
2. Jelaskan kenapa algoritma dibutuhkan dalam pembuatan program!
3. Jelaskan urutan penyelesaian masalah dengan menggunakan notasi algoritma!
4. Jelaskan cara-cara penulisan algoritma!
5. Jelaskan kelebihan dan kekurangan masing-masing notasi algoritma!

#### **E. Lembar kerja praktek**

1. Buatlah algoritma membuat mie rebus menggunakan notasi penulisan algoritma.
  - a. kalimat deskriptif,
  - b. Flowchart dan
  - c. Pseudocode

## DAFTAR PUSTAKA

- Abdul, Kadir., 1995. Menggunakan TurboC++ dan Borland C++. Yogyakarta: Andi Offset.
- ., 2010, Mudah Mempelajari Database Access. Yogyakarta : Andi.
- ., 2014, Pengenalan Sistem Informasi Edisi Revisi. Yogyakarta : Andi.
- ., 2012. Algoritma dan Pemrograman Menggunakan Java. Yogyakarta: Andi.
- Ali, Ridho Barakbah., 2013, Logika dan Algoritma. Surabaya : Politeknik Elektronika Negeri Surabaya.
- Alatas, Husein. 2013. Responsive Web Design Dengan PHP & Bootstrap. Yogyakarta: Lokomedia.
- Al Fatta Hanif, 2007. “Analisis dan Perancangan Sistem Informasi Untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern”. Andi Offset ; Yogyakarta.
- Arief, Rudyanto. 2011. Pemrograman Web Dinamis Menggunakan PHP Dan MySQL. Yogyakarta: Andi.
- Astrom, K.J. & Wittenmark, B. 1997. Computer Controlled system theory and design. New Jersey: Prentice Hall International Editions. ATIKA
- Bejo.2008. C dan AVR rahasia kemudahan bahasa C dalam mikrokontroller ATmega8535, Yogyakarta: Graha Ilmu
- Binanto, iwan. 2009. Lebih Lanjut dengan pemrograman c++ di linux. Yogyakarta : Andi Publisher.
- Darmawan, ERico, dan Risal. 2014. Pemrograman Berorientasi Objek C# yang susah jadi mudah. Bandung: Informatika Bandung.
- Enterprise, Jubile. 2014. Visual c# untuk Pemula. Yogyakarta. PT. Elex Media Komputindo
- Handoyo, Erico Darmawan & Laurentius Risal.2011. Pemrograman Berorientasi Objek C#. Bandung : Informatika.
- Hariato, Bambang. 2004. Rekayasa Sistem Berorientasi Objek. Bandung: Informatika.
- Huda, Asrul & Ardi, Noper. 2020. Dasar-dasar pemrograman Berbasis Python. Padang: UNPPRESS

- Indra, Yatini B., 2010, Flowchart, Algoritma dan Pemrograman Menggunakan Bahasa C++ Builder. Yogyakarta : Penerbit Graha Ilmu.
- Jamsa, K. 1996. 1001 Tip C/C++ Yogyakarta: Andi Offset
- Janner, Simarmata., 2010, Rekayasa Web. Yogyakarta : CV. Andi Offset.
- Jeperson, Hutahaeen., 2014, Konsep Sistem Informasi. Yogyakarta : Deepublish.
- Jogiyanto, 1993. Konsep Dasar Pemrograman Bahasa C. Yogyakarta : Andi Offset.
- Kernighan, Brian W. & Ritchie, Dennis M.1988. The Ansi C Programming Language, Prentice Hall
- Knuth. 1998. The Art of Computer Programming vol. 2 (3rd ed.). Boston: Addison–Wesley
- Malvino, A.P. 1994. Elektronika Komputer Digital. Jakarta: Penerbit Erlangga.
- Munir, Rinaldi. 2007. Algoritma dan Pemrograman dalam Bahasa Pascal dan C. Bandung: Informatika.
- . 2011. Algoritma dan Pemrograman dalam bahasa Pascal dan C. Bandung : Informatika Bandung
- Nugroho, Adi. 2005. Rational Rose untuk Pemodelan Berorientasi Obyek. Bandung: Informatika
- Rosen, Kenneth H.1999. Discrete Mathematics and Its Application, Edisi Keempat, McGraw-Hill.
- Sholih., dan Imam Robandi. , 2010, Analisis dan Perancangan Berorientasi Objek. Bandung : CV. Muara Indah.
- Sitorus, lamhot. 2015. ALGORITMA DAN PEMROGRAMAN. Yogyakarta : CV. ANDI OFFSET
- Smith, N.E. 1994. Borland C++ Panduan Praktis Berilustrasi. Jakarta: Media Komputindo.

- Soetam, Rizky., 2011, Konsep Dasar Rekayasa Perangkat Lunak. Jakarta : Prestasi Pustaka.
- Suarga. 2012. Algoritma dan Pemrograman. Edisi Kedua. Yogyakarta: Andi
- Tirtamiharja, S. 1996. Elektronika digital. Yogyakarta: Andi Offset
- Utami, Ema, Sukrisno (2005). 10 Langkah Belajar Menggunakan Logika dan Algoritma, Menggunakan Bahasa C dan C++ di GNU/LINUX. Yogyakarta: Andi
- Yulikuspartono, 2009. Pengantar Logaritma dan Algoritma. Yogyakarta : Andi.

## GLOSARIUM

Algoritma	: Urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis
Argumen	: Parameter yang menyertai pemanggilan function
Array	: Sebagai kumpulan dari nilai-nilai data bertipe sama dalam urutan tertentu yang menggunakan sebuah nama yang sama
Break	: Pernyataan lompatan yang dapat mengendalikan aliran pengekseskuan CPU
<i>Cohesive</i>	: Mengukur seberapa kuat masing-masing fungsi terkait dalam suatu modul
Combine	: Menggabungkan solusi masing-masing upa-masalah sehingga membentuk solusi masalah semula.
Conquer	: Memecahkan (menyelesaikan) masing-masing masalah (secara rekursif)
Console	: Interface dasar pada computers, biasanya berupa keyboard dan monitor
Coupling	: Mengukur seberapa besar masing-masing modul program bergantung pada modul program lainnya
<i>Decrement</i>	: Digunakan untuk mengurangi variabel sebanyak 1 angka
Deklarasi	: Proses pendefinisian suatu variabel
Divide	: Membagi masalah menjadi beberapa upa-masalah yang memiliki kemiripan dengan masalah semula namun berukuran lebih kecil
Elemen	: Bagian dari program
<i>Exception</i>	: Pernyataan yang memungkinkan kita menangani kesalahan saat program berjalan
<i>Flowchart</i>	: Bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program



<i>Flow Control</i>	: Suatu teknik untuk menjamin bahwa entitas pengirim tidak akan membanjiri data kepada entitas penerima
Fungsi ( <i>Function</i> )	: Sekumpulan kode yang ditaruh dalam sebuah blok dan dibuat untuk menjalankan tugas khusus
Identifier	: Untaian satu atau lebih huruf, angka, atau garis bawah
<i>Increment</i>	: Digunakan untuk menambah variabel sebanyak 1 angka
Instansiasi	: Pembuatan instance/objek dari suatu kelas
<i>Jump</i>	: Salah satu cara kita untuk mengendalikan aliran eksekusi CPU, yang memungkinkan kita untuk membuat CPU melompat beberapa pernyataan dan menuju ke suatu baris tertentu.
Konstanta	: Sebuah tempat atau container dari suatu nilai. Sesuai dengan namanya, nilai dari konstanta bersifat tetap (konstan) dan tidak bisa diubah sepanjang program.
<i>Library</i>	: Koleksi dari rutin-rutin program yang digunakan untuk membangun dan mengembangkan perangkat lunak
Logis logika	: proses berpikir yang rasional atau dengan
Looping	: Cara yang memungkinkan kita untuk mengendalikan proses pengekseskuan program, membuat aliran memproses pernyataan yang sama secara berulang-ulang hingga kondisi yang diinginkan terpenuhi
Modularisasi	: Suatu program yang besar dan kompleks dapat dibagi ke dalam beberapa prosedur sehingga setiap prosedur merupakan bagian yang mudah dikerjakan
Notasi	: Lambang-lambang yang digunakan dalam algoritma maupun pemrograman
Parameter	: Tempat penyimpanan (variabel) di dalam function, yang digunakan untuk melakukan pemberian data dari pemanggil ke dalam function

Prosedur	: Bagian dari suatu program yang disusun secara terpisah untuk melakukan suatu tugas khusus atau fungsi tertentu
Pseudocode	: Sebuah deskripsi atau ringkasan tingkat tinggi dari algoritma pemrograman komputer dengan konvensi struktura
Rekursif	: Proses dimana fungsi memanggil dirinya sendiri
<i>Searching</i>	: Tindakan untuk mendapatkan suatu data dalam kumpulan data berdasarkan satu kunci (key) atau acuan data
<i>Selection</i>	: Mengambil suatu keputusan berdasarkan kondisi tertentu yang sedang dialami oleh program
<i>Stream</i>	: Nama umum yang diberikan untuk menyebutkan suatu proses aliran data
String	: Bentuk data yang biasa dipakai dalam bahasa pemrograman untuk keperluan menampung dan memanipulasi data teks, misalnya untuk menampung (menyimpan) suatu kalimat
<i>Sorting</i>	: Suatu proses untuk menyusun kembali himpunan obyek menggunakan aturan tertentu
Variabel	: Lokasi penyimpanan dan terkait nama simbolis yang berisi beberapa kuantitas yang diketahui atau tidak diketahui atau informasi, nilai

## INDEKS

### A

*Algoritma*, v, vii, 1, 2, 3, 4, 7, 15, 82, 99, 101, 102, 141, 147  
*Argumen*, 26, 89, 93, 112, 114, 115, 119, 120, 121, 122, 123, 124, 128  
*Array*, v, vii, viii, 25, 27, 85, 86, 88, 90, 92, 102, 129, 139, 142, 144, 145, 147, 148, 150, 151, 152, 153

### B

*Break*, 24, 25, 26, 52, 53, 74, 75, 76, 83, 136

### C

*Cohesive*, 101, 102, 103  
*Combine*, 101, 105  
*Conquer*, 101, 102, 105, 147, 148  
*Console*, vi, 44, 47, 48, 49  
*Coupling*, 101, 102, 103

### D

*Decrement*, 24, 25, 39  
*Deklarasi*, vii, viii, 4, 5, 6, 13, 14, 15, 16, 47, 85, 86, 87, 88, 89, 106, 107, 108, 114, 116, 117, 119  
*Divide*, vii, viii, 101, 102, 147

### E

*Elemen*, 86, 87, 90, 91, 101, 102, 129, 130, 131, 135, 136, 138, 141, 142, 143, 145, 154, 155, 156, 157, 160, 162, 164, 165, 173, 178, 180, 183  
*Exception*, vi, 82

### F

*Flowchart*, viii, 7, 8, 9, 10, 11, 54, 57, 68, 71  
*Flow control*, vi, 51, 52, 53, 82  
*Fungsi*, v, vi, vii, viii, 5, 8, 9, 10, 12, 24, 25, 26, 27, 43, 44, 45, 46, 49, 55, 68, 70, 74, 76, 78, 80, 85, 87, 88, 89, 92, 93, 94, 95, 96, 97, 99, 100, 109, 111, 116, 117

### I

*Increment*, 39, 134,  
*Identifler*, v, vi, 19, 28, 29, 30, 31, 92, 112  
*Instansiasi*, 73, 86, 87, 90, 91, 132, 153

### J

*Jump*, 74, 77, 81

### K

*Konstanta*, 5, 6, 13, 28, 46, 55, 85, 86, 115

### L

*Library*, 25, 44, 45, 49, 106  
*Logis*, 2, 3  
*Looping*, v, viii, 30, 67, 68, 83

### M

*Modularisasi*, 100, 102

### N

*Notasi*, 3, 4, 7, 13, 18

## **P**

**Parameter**, vii, 6, 107, 109, 111, 112, 113, 114, 115, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 130, 131

**Prosedur**, v, vi, viii, 5, 6, 8, 9, 51, 52, 99, 100, 101, 107, 109, 110

**Pseudocode**, 12, 13, 14, 15, 16, 17, 18

## **S**

**Searching**, v, vii, 139, 140, 141, 147, 152

**Selection**, v, viii, 53, 54, 102, 156, 162, 176

**Stream**, vi, 26, 43, 44, 45, 46, 47, 48, 49

**String**, v, 6, 33, 34, 40, 46, 49, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97

**Sorting**, v, vii, 150, 154, 155, 157, 158, 163, 183

## **V**

**Variabel**, viii, 5, 6, 13, 28, 29, 30, 31, 32, 35, 38, 40, 41, 46, 47, 48, 62, 78, 79, 85, 86, 87, 88, 89, 90, 91, 93, 97, 100, 101, 108, 109, 113, 114, 115, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 131, 132, 135, 138, 142, 145

## **INTISARI RINGKASAN BUKU**

Buku ini memberikan panduan yang mudah bagi pembaca dalam memahami Pengantar Coding berbasis C/C++. Pada bagian awal, pembaca dikenalkan dengan dasar-dasar algoritma dan sejarah pemrograman C/C++. Kemudian dilanjutkan dengan pengenalan tata cara menulis dan menggunakan algoritma dalam menangani masalah sederhana. Setelah faham mengenai konsep dasar algoritma, pembaca akan dikenalkan secara langsung dengan Bahasa C/C++ serta akan dibimbing untuk membuat program sederhana. Selanjutnya, secara lebih komprehensif dibahas mengenai konsep pemrograman di C/C++ yang meliputi Tipe Data, Identifier dan Operator sebagai pondasi awal dalam memahami konsep pemrograman C/C++. Selanjutnya Pembaca akan disuguhkan materi mengenai konsep *Input & Output* beserta Struktur kontrol untuk memperkaya pemahamannya. Lalu dilanjutkan dengan pembahasan mengenai kelas-kelas khusus dalam pengolahan String. Selanjutnya dibahas tentang Prosedur dan fungsi untuk mengenalkan konsep pemrograman Modular ke pembaca. Dalam buku ini juga dilengkapi dengan pembahasan tentang Array dan kelas-kelas khusus dalam string yang dibahas secara interaktif dan menarik dengan beberapa contoh penerapan sederhana yang mudah dicerna pemula. Kemudian Pada bagian penutup dipaparkan topik mengenai Konsep *Searching* dan *Sorting*.