



BAHAN AJAR PERKULIAHAN

Algoritma

Pemrograman

Departemen	Program Studi	Kode MK	Disusun Oleh	SKS	SEMESTER
Teknik Elektronika	Teknik Informatika	INF1.62.1007	Drs. Denny Kurniadi, M.Kom	3	Ganjil

Kegiatan Belajar 1

Sistem Komputer, Algoritma dan Pemrograman

A. Pendahuluan

1. Deskripsi Singkat

Secara umum, tujuan kegiatan belajar 1 ini untuk membekali mahasiswa dengan pengetahuan dan keterampilan dasar dalam sistem komputer, algoritma, dan pemrograman, sehingga mereka dapat memahami konsep-konsep yang mendasari pengembangan perangkat lunak dan mampu memecahkan masalah dengan menggunakan algoritma dan bahasa pemrograman yang relevan.

Secara khusus, tujuan kegiatan belajar 1 ini adalah agar peserta mampu:

1. Menjelaskan dasar sistem komputer, program dan sistem operasi
 - Memahami komponen-komponen dasar sistem komputer, seperti CPU, memori, input/output, dan perangkat penyimpanan.
 - Mengetahui bagaimana komponen-komponen tersebut bekerja bersama untuk menjalankan program komputer.
 - Memahami arsitektur komputer dan struktur organisasi yang digunakan dalam pemrosesan data.
2. Menjelaskan konsep algoritma program
 - Memahami konsep dasar algoritma dan kegunaannya dalam memecahkan masalah.
 - Mampu merancang algoritma yang efisien dan efektif untuk berbagai masalah pemrograman.
 - Mampu menganalisis dan mengevaluasi algoritma dari segi kompleksitas dan kecepatan eksekusi.
3. Menjelaskan pemrograman
 - Menguasai bahasa pemrograman C.
 - Memahami sintaks dan struktur dasar dalam pemrograman.
 - Mampu mengimplementasikan algoritma ke dalam bahasa pemrograman yang dipilih.
4. Penyelesaian Masalah:
 - Mampu menerapkan konsep algoritma dan pemrograman untuk memecahkan masalah nyata.
 - Mengembangkan keterampilan analitis dalam memecahkan masalah secara sistematis.
 - Mampu mengidentifikasi masalah, merancang algoritma yang tepat, dan

mengimplementasikannya dalam kode program.

2. Panduan Belajar

Proses pembelajaran untuk materi modul 1 kegiatan belajar 1 dapat berjalan dengan lancar apabila Anda mengikuti langkah-langkah belajar sebagai berikut:

1. Pahami dulu kegiatan penting dalam program pelatihan ini dengan memperhatikan isi capaian pembelajaran setiap kegiatan belajar.
2. Lakukan kajian terhadap setiap materi dalam kegiatan belajar, agar memudahkan proses pembelajaran.
3. Pelajari dahulu kegiatan belajar 1 yang setiap akhir kegiatan belajar menyelesaikan tugas yang harus dikerjakan secara langsung.
4. Keberhasilan program pembelajaran ini tergantung dengan kesungguhan Anda dalam mengerjakan setiap tugas dalam kegiatan belajar.
5. Bila Anda menemukan kesulitan, silahkan hubungi instruktur pembimbing atau fasilitator yang mengajar modul ini.

B. Inti

1. Capaian Pembelajaran Matakuliah

Setelah mengikuti seluruh tahapan pada kegiatan belajar, peserta mampu memahami prinsip dasar dan lingkungan pemrograman C serta cara menulis Algoritma.

2. Pokok Materi

- a. Sistem komputer
- b. Konsep Algoritma
- c. Pemograman Dasar

3. Uraian Materi

A. Sistem Komputer

Komputer memerlukan sebuah sistem yang terdiri dari elemen-elemen yang saling terhubung untuk mengolah data dan menghasilkan informasi. Komponen-komponen utama dalam sistem komputer meliputi:

- a. Hardware, merupakan sumber daya utama untuk proses komputasi. Hardware atau perangkat keras komputer terdiri dari CPU (Central Processing Unit), unit memory dan perangkat unit input dan unit output. Bagian unit CPU ini dapat di rinci terdiri dari ALU (Arithmetic Logic Unit) dan CU (Control Unit);

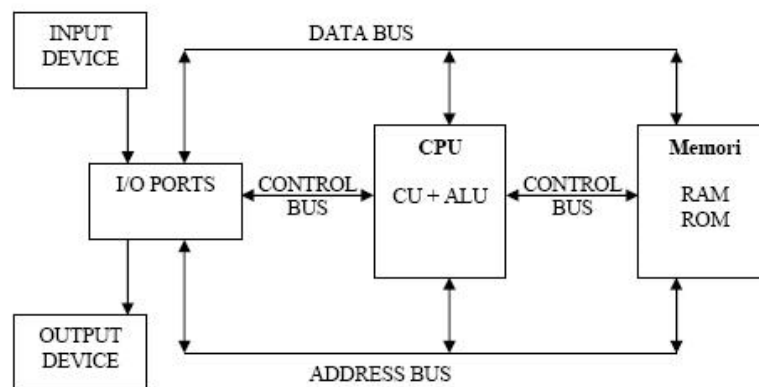
- b. Software merupakan program-program komputer yang berguna untuk menjalankan suatu pekerjaan sesuai dengan yang dikehendaki. Program tersebut ditulis dengan bahasa khusus yang dimengerti oleh komputer;
- c. Brainware ; pengguna sistem, terdiri dari orang, mesin atau komputer lain.



Gambar 1. Hardware, Software dan Brainware

Dari Gambar 1, dapat diamati bahwa perangkat keras hanya memiliki fungsi jika didukung oleh perangkat lunak. Demikian pula, baik perangkat keras maupun perangkat lunak memerlukan interaksi manusia agar dapat beroperasi.

Dalam konteks sistem komputer, penting untuk memahami struktur komputer yang melibatkan hubungan dan organisasi komponen-komponen tersebut. Struktur komputer adalah hubungan antara setiap komponen yang saling terkait. Secara sederhana, struktur komputer dapat diilustrasikan dalam bentuk diagram blok berikut ini:



Gambar 2. Diagram Blok Komputer

1. Input Device merupakan perangkat keras komputer yang berfungsi sebagai alat untuk memasukan data atau perintah ke dalam komputer.
2. Output Device merupakan perangkat keras komputer yang berfungsi untuk menampilkan keluaran sebagai hasil pengolahan data. Keluaran dapat berupa hard-copy (ke kertas), soft-copy (ke monitor), ataupun berupa suara.
3. I/O Ports merupakan bagian ini digunakan untuk menerima ataupun mengirim data ke luar sistem. Peralatan input dan output di atas terhubung melalui port ini.
4. CPU (Central Processing Unit) merupakan otak sistem komputer, dan memiliki dua bagian fungsi operasional, yaitu: ALU (Arithmetical Logical Unit) sebagai pusat

pengolah data, dan CU (Control Unit) sebagai pengontrol kerja komputer. CPU terdiri dari dua bagian utama yaitu unit kendali (control unit) dan unit aritmatika dan logika (ALU). Disamping itu, CPU mempunyai beberapa alat penyimpanan yang berukuran kecil yang disebut dengan register.

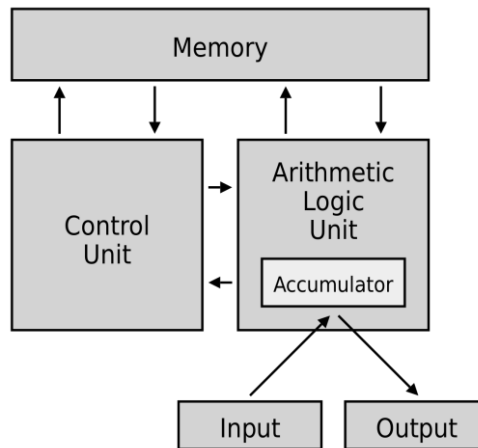
CU (Control Unit) / Unit Kendali, tugas dari unit kendali ini adalah:

- ✓ Mengatur dan mengendalikan alat-alat input dan output.
 - ✓ Mengambil instruksi-instruksi dari memori utama.
 - ✓ Mengambil data dari memori utama (jika diperlukan) untuk diproses.
 - ✓ Mengirim instruksi ke ALU bila ada perhitungan aritmatika atau perbandingan logika serta mengawasi kerja dari ALU.
 - ✓ Menyimpan hasil proses ke memori utama.
5. ALU (Arithmetic and Logic Unit). Tugas utama dari ALU adalah melakukan semua perhitungan aritmatika (matematika) yang terjadi sesuai dengan instruksi program. Melakukan keputusan dari suatu operasi logika sesuai instruksi program.
 6. Register merupakan alat penyimpanan kecil yang mempunyai kecepatan akses tinggi, yang digunakan untuk menyimpan data dan instruksi yang sedang diproses sementara data dan instruksi lainnya yang menunggu giliran untuk diproses masih disimpan di dalam memori utama.
 7. Memori terbagi menjadi dua bagian yaitu memori internal dan memori eksternal. Memori internal berupa RAM (Random Access Memory) yang berfungsi untuk menyimpan program yang di olah untuk sementara waktu, dan ROM (Read Only Memory) yaitu memori yang hanya bisa dibaca dan berguna sebagai penyedia informasi pada saat komputer pertama kali dinyalakan.
 8. Ukuran memori ditunjukkan oleh satuan byte, misalnya 512 Mega Byte, 1 Giga Byte (1000 MB), 2 GB, 4 GB, atau bahkan ada yang sampai puluhan GB. Pada umumnya 1 byte memori terdiri dari 8 – 32 bit (binary digit), yaitu banyaknya digit biner (0 atau 1) yang mampu disimpan dalam satu chip memori.
 9. Data Bus adalah jalur-jalur perpindahan data antar modul dalam sistem komputer. Karena pada suatu saat tertentu masing-masing saluran hanya dapat membawa 1 bit data, maka jumlah saluran menentukan jumlah bit yang dapat ditransfer pada suatu saat. Lebar data bus ini menentukan kinerja sistem secara keseluruhan. Sifatnya bidirectional, artinya CPU dapat membaca dan menirma data melalui data bus ini. Data bus biasanya terdiri atas 8, 16, 32, atau 64 jalur paralel.
 10. Address Bus digunakan untuk menandakan lokasi sumber ataupun tujuan pada proses transfer data. Pada jalur ini, CPU akan mengirimkan alamat memori yang

akan ditulis atau dibaca. Address bus biasanya terdiri atas 8, 16, 32 atau 64 jalur paralel.

11. Control Bus digunakan untuk mengontrol penggunaan serta akses ke Data Bus dan Address Bus. Terdiri atas 4 samapai 10 jalur paralel.

Dengan memahami struktur komputer, kita dapat melihat bagaimana arsitektur komputer membentuk dasar operasional dan desain sistem komputer secara keseluruhan. Arsitektur Komputer merupakan desain atau struktur internal komputer, seperti arsitektur von Neumann yang menggambarkan aliran data dan instruksi dalam komputer.



Gambar 3. Arsitektur Komputer

Pada Gambar 4 arsitektur komputer terdiri dari empat komponen utama yaitu: (1) Unit Pemrosesan Pusat (Central Processing Unit/CPU) bertanggung jawab atas eksekusi instruksi dan pemrosesan data; (2) Memori sebagai tempat penyimpanan instruksi dan data yang digunakan oleh CPU; (3) Input/Output (I/O) Devices adalah perangkat-perangkat yang digunakan untuk menerima input dan menghasilkan output dari sistem komputer; (4) Bus merupakan jalan komunikasi yang menghubungkan CPU, memori, dan perangkat I/O untuk mentransfer data dan instruksi.

Penting untuk diingat bahwa arsitektur komputer terus berkembang seiring dengan kemajuan teknologi, dan ada berbagai variasi dan pendekatan yang digunakan dalam implementasi arsitektur komputer yang berbeda.

Kemudian didalam komputer terdapat sistem bilangan yang merupakan notasi yang digunakan untuk merepresentasikan angka dalam komputer, seperti sistem bilangan biner (basis 2), desimal (basis 10), atau heksadesimal (basis 16). Sistem bilangan menjadi dasar dalam representasi, pemrosesan, dan komunikasi data di dalam komputer. Dengan menggunakan sistem bilangan biner, komputer dapat memanipulasi data dan melakukan operasi matematika dengan efisien, memungkinkan berbagai fungsi dan aplikasi yang ada pada komputer saat ini.

B. Konsep Algoritma

Awalnya, algoritma digunakan dalam matematika, namun sekarang lebih sering digunakan dalam pemrograman komputer. Algoritma menjadi konsep dasar yang penting dalam menciptakan program komputer yang efisien, efektif, dan terstruktur. Algoritma adalah serangkaian aturan atau prosedur logis yang digunakan untuk memecahkan masalah dengan efisiensi. Istilah "algoritma" berasal dari nama seorang penulis buku Arab terkenal, Al-Khuwarizmi, yang juga memberi kontribusi dalam bidang matematika. Istilah "algorism" kemudian berkembang menjadi "algorithm" karena adanya kebingungan dengan istilah "arithmetic". Algoritma adalah urutan langkah-langkah logis yang disusun secara sistematis dan logis untuk menyelesaikan masalah. Saat ini, algoritma dikenal sebagai jantung dari ilmu komputer atau informatika. Beberapa karakteristik penting dari algoritma adalah:

- a. Terstruktur: Algoritma harus memiliki struktur yang terorganisir dan jelas. Langkah-langkahnya harus diatur secara logis sehingga mudah diikuti dan dipahami.
- b. Terbatas: Algoritma harus memiliki batasan yang jelas dan dapat diselesaikan dalam waktu yang terbatas. Setiap langkah dalam algoritma harus dapat diselesaikan dengan jumlah langkah yang terbatas.
- c. Deterministik: Algoritma harus memiliki hasil yang pasti untuk setiap input tertentu. Artinya, jika diberikan input yang sama, algoritma akan menghasilkan output yang sama setiap kali dijalankan.
- d. Efisien: Algoritma yang baik dirancang untuk mencapai tujuan dengan cara yang efisien. Ini berarti menggunakan jumlah sumber daya (seperti waktu dan memori) yang minimum untuk menyelesaikan tugas yang diberikan.
- e. Dapat dimengerti oleh manusia: Algoritma harus ditulis dengan jelas dan dapat dimengerti oleh manusia. Ini memungkinkan orang lain untuk membaca, memahami, dan mengimplementasikan algoritma tersebut.

Algoritma memainkan peran yang sangat penting dalam pemrograman dan ilmu komputer karena mereka membantu merancang solusi yang efisien untuk berbagai masalah. Dengan menggunakan algoritma yang baik, kita dapat merancang program-program yang menyelesaikan tugas dengan cara yang terstruktur dan efisien. Beberapa cabang ilmu komputer yang menggunakan algoritma sebagai dasar mereka meliputi:

- Algoritma perutean (routing) pesan dalam jaringan komputer.
- Algoritma Berensenhham untuk menggambar garis lurus pada bidang grafik komputer.
- Algoritma Knuth-Morris-Pratt untuk mencari pola dalam teks pada bidang information retrieval.

Contoh-contoh tersebut menunjukkan bagaimana algoritma digunakan di berbagai bidang ilmu komputer untuk memecahkan masalah spesifik dan mencapai tujuan tertentu. Implementasi algoritma sangat umum digunakan oleh praktisi, akademisi, dan bahkan orang awam, meskipun seringkali orang awam tidak menyadarinya. Contohnya, dalam penelitian ilmu kedokteran, praktisi melakukan implementasi algoritma ketika meneliti proses penerimaan asupan makanan oleh tubuh manusia, mulai dari pencernaan hingga penghasilan energi dalam sel darah otak.

Struktur algoritma terdiri dari beberapa bagian. Bagian kepala (header) berisi keterangan permasalahan yang akan diimplementasikan dalam algoritma. Pendeklarasian dilakukan untuk menentukan variabel atau objek yang terkait dengan tipe data dalam bagian kerangka kepala. Tahap terakhir adalah deskripsi rinci langkah-langkah sistematis dan logis untuk menyelesaikan masalah dan menghasilkan solusi atau output.

Sebagai contoh sederhana, dalam kehidupan sehari-hari, Dina, seorang mahasiswa tahun pertama di UNP, setiap hari pergi ke kampus setelah sarapan. Suatu hari, ketika Dina ingin sarapan, hanya ada nasi tanpa sayur dan lauk. Dina kemudian memutuskan untuk membuat tumis taugé tempe dengan langkah-langkah berikut:

1. Siapkan bahan-bahan seperti taugé, tempe, cabai, bawang merah, bawang putih, lengkuas, dan bahan lainnya.
2. Iris bawang merah, bawang putih, cabai, dan lengkuas.
3. Panaskan minyak dan masukkan semua bahan yang telah diiris.
4. Goreng tempe sebentar.
5. Tambahkan taugé, kecap manis, garam, dan sedikit air.
6. Aduk hingga semua bumbu meresap.
7. Cicipi rasanya. Jika ada yang kurang, tambahkan bumbu lain secukupnya.
8. Taugé tumis siap dihidangkan.

Contoh di atas merupakan penyelesaian masalah dengan menggunakan langkah-langkah yang dinamakan algoritma. Setiap hari, ketika seseorang menjalankan aktivitas, mereka akan memilih langkah mana yang akan dilakukan terlebih dahulu, seperti saat bangun tidur, sarapan, atau bahkan saat memakai pakaian di pagi hari. Algoritma yang baik adalah tindakan yang benar dan masuk akal. Terdapat berbagai cara untuk mengkomunikasikan algoritma, termasuk menggunakan bagan alir, pseudocode, atau bahasa pemrograman. Bagan alir adalah bentuk algoritma yang mudah dibaca dan dipahami.

Pada saat menyusun dan menulis algoritma program, terdapat beberapa tahapan yang perlu diikuti secara sistematis. Tahapan ini membantu kita dalam merancang

langkah-langkah logis yang efisien dan terstruktur untuk menyelesaikan masalah secara programatik. Berikut ini adalah tahapan-tahapan yang umumnya dilakukan dalam penyusunan dan penulisan algoritma program.

1. Bagaimana merencanakan suatu algoritma

Dalam mencari solusi untuk suatu masalah, penting untuk menentukan beberapa model atau desain sebagai opsi yang mungkin. Dengan demikian, akan ada banyak variasi desain atau model yang dapat dipertimbangkan untuk memilih yang terbaik. Berikut ini beberapa contoh model yang dapat dipertimbangkan saat mencari solusi untuk suatu masalah:

- ✓ Model Jaringan: Digunakan untuk memodelkan interaksi antara entitas atau elemen dalam suatu sistem, seperti model jaringan komputer atau model jaringan transportasi.
- ✓ Model Simulasi: Menggunakan representasi matematika atau komputer untuk mensimulasikan sistem nyata atau proses, seperti model simulasi cuaca atau model simulasi lalu lintas jalan raya.
- ✓ Model Proses Bisnis: Menjelaskan aliran kerja dan proses dalam suatu organisasi atau bisnis, seperti model proses bisnis untuk meningkatkan efisiensi atau memperbaiki operasional.
- ✓ Model Prediksi: Menggunakan data historis dan metode statistik untuk membuat prediksi tentang hasil masa depan, seperti model prediksi penjualan atau model prediksi cuaca.
- ✓ Model Optimasi: Mencari solusi optimal dengan mempertimbangkan berbagai batasan dan tujuan tertentu, seperti model optimasi jadwal penugasan atau model optimasi pemotongan bahan.
- ✓ Model Heuristik: Menggunakan aturan praktis atau metode pendekatan untuk mencari solusi yang baik, meskipun tidak secara eksplisit optimal, seperti model heuristik dalam permasalahan rute terpendek atau penjadwalan tugas.
- ✓ Model Keputusan: Memberikan kerangka kerja untuk pengambilan keputusan dengan mempertimbangkan berbagai faktor dan konsekuensi, seperti model keputusan dalam analisis risiko atau penentuan harga.

Catatan bahwa contoh-contoh ini hanya beberapa di antara banyak jenis model yang dapat digunakan dalam mencari solusi untuk berbagai masalah. Pemilihan model tergantung pada sifat masalah yang dihadapi dan tujuan yang ingin dicapai.

2. Bagaimana menyatakan suatu algoritma

Bagaimana menyatakan suatu algoritma merupakan cara untuk

mengungkapkan langkah-langkah penyelesaian masalah secara terstruktur dan terurut dalam bentuk yang dapat dipahami oleh manusia dan dapat diimplementasikan oleh komputer. Terdapat beberapa cara untuk menyatakan suatu algoritma, di antaranya:

- a) Dengan Bahasa semu (Pseudocode): yaitu dengan menggunakan bahasa sehari-hari, tetapi harus jelas dan terstruktur.

Contoh:

Untuk menghitung Luas Segitiga:

1. Masukkan Nilai Alas
2. Masukkan Nilai Tinggi
3. Hitung Luas = $(\text{Alas} * \text{Tinggi}) / 2$
4. Cetak Luas

- b) Dengan diagram alur atau flowchart: yaitu dengan membuat suatu penulisan atau penyajian algoritma berupa diagram yang menggambarkan susunan alur logika dari suatu permasalahan.

- c) Dengan Statement Program/Penggalan Program

Contoh:

1. Read Alas
2. Read Tinggi
3. Luas = $(\text{Alas} * \text{Tinggi}) / 2$
4. Write(luas)

3. Bagaimana validitas suatu algoritma

Penting untuk menyatakan suatu algoritma dengan memastikan bahwa solusi yang dihasilkan merupakan penyelesaian sebenarnya dari masalah yang ada, bukan menciptakan masalah baru.

4. Bagaimana menganalisa suatu algoritma

Salah satu cara untuk melihat running time atau waktu yang diperlukan dalam menyelesaikan suatu masalah, serta jumlah memori yang digunakan selama proses penyelesaian masalah tersebut, adalah dengan melakukan pengamatan terhadap kedua faktor tersebut.

5. Bagaimana menguji program dari suatu algoritma

Salah satu cara untuk menyajikan algoritma adalah dengan menggunakan salah satu bahasa pemrograman seperti BASIC, PASCAL, FORTRAN, JAVA, C++, atau bahasa pemrograman lainnya. Selama proses ini, program akan mengalami beberapa tahap pengujian oleh komputer, yang meliputi:

- ✓ Tahap Debugging: Tahap ini terjadi selama eksekusi program dan bertujuan untuk

memperbaiki kesalahan yang ada dalam program. Kesalahan tersebut dapat berupa kesalahan logika atau kesalahan sintaksis dalam penulisan program.

- ✓ Tahap Profiling: Setelah program diperbaiki dan berhasil melewati tahap debugging, tahap ini bekerja untuk mengamati dan mengukur waktu eksekusi atau running time yang diperlukan serta jumlah memori atau storage yang digunakan dalam menyelesaikan suatu algoritma. Dalam tahap profiling, informasi mengenai waktu eksekusi dan penggunaan memori akan direkam dan dianalisis untuk memperoleh pemahaman yang lebih baik tentang performa algoritma yang telah diimplementasikan.

Untuk menganalisis faktor efisiensi dan efektivitas suatu algoritma, kita dapat mempertimbangkan beberapa hal sebagai berikut:

Waktu tempuh (Running Time) algoritma: Ini adalah waktu yang diperlukan oleh algoritma untuk menyelesaikan suatu masalah. Faktor-faktor yang mempengaruhi waktu tempuh meliputi:

- Jumlah langkah atau instruksi: Semakin banyak langkah yang diperlukan untuk menyelesaikan masalah, semakin lama waktu tempuh yang dibutuhkan.
- Besar dan jenis input data: Ukuran dan jenis data yang digunakan dalam algoritma akan mempengaruhi proses perhitungan. Misalnya, penggunaan data dengan tingkat ketelitian tunggal (Single precision) akan lebih cepat daripada tingkat ketelitian ganda (double precision).
- Jenis operasi: Waktu tempuh juga dipengaruhi oleh jenis operasi yang dilakukan, seperti operasi matematika, logika, atau operasi lainnya. Misalnya, operasi perkalian atau pembagian biasanya memakan waktu lebih lama daripada operasi penjumlahan atau pengurangan.
- Komputer dan kompilator: Faktor terakhir yang mempengaruhi waktu tempuh adalah kemampuan komputer dan kompilator. Meskipun ini berada di luar tahap perancangan algoritma yang efisien, performa algoritma akan bergantung pada kemampuan komputer dan kompilator yang digunakan.
- Penggunaan Memori: Jumlah langkah dan jenis variabel data yang digunakan dalam algoritma akan mempengaruhi penggunaan memori. Dalam hal ini, penting untuk memperkirakan kebutuhan memori yang diperlukan selama proses berlangsung agar dapat menyediakan penyimpanan yang memadai tanpa hambatan atau kekurangan memori.

Dengan mempertimbangkan faktor-faktor tersebut, kita dapat mengevaluasi efisiensi dan efektivitas suatu algoritma dalam menyelesaikan masalah yang diberikan.

Beberapa hal yang perlu diperhatikan dalam menyusun suatu algoritma adalah:

1. Banyaknya langkah instruksi harus berhingga: Algoritma harus memiliki jumlah langkah operasional yang terbatas agar dapat diakhiri atau diselesaikan. Dengan demikian, diharapkan algoritma dapat menghasilkan solusi yang baik.
2. Langkah atau instruksi harus jelas: Setiap langkah dalam algoritma harus memiliki arti yang spesifik dan dapat dibedakan antara penulisan langkah untuk manusia (pseudocode) dan penulisan langkah untuk komputer (bahasa pemrograman). Simbol-simbol yang mudah dipahami manusia, seperti diagram alur (flowchart), dapat membantu pemahaman algoritma.
3. Proses harus jelas dan mempunyai batasan: Rangkaian langkah instruksi dalam algoritma harus ditetapkan dengan jelas, baik, dan pasti. Setiap instruksi harus memiliki pelaksana yang berfungsi sebagai pemroses data. Algoritma juga harus memiliki instruksi dasar yang membatasi proses yang akan dilakukan oleh komputer.
4. Input dan Output harus mempunyai batasan: Input adalah data yang dimasukkan ke dalam algoritma untuk diproses oleh komputer, sedangkan output adalah hasil dari proses yang dilakukan oleh komputer. Input dan output harus sesuai dengan jenis bahasa pemrograman yang digunakan.
5. Efektivitas: Instruksi yang diberikan pada komputer harus hanya menjalankan proses yang mampu dilaksanakan oleh komputer. Algoritma harus ditulis dengan instruksi yang lengkap, benar, dan jelas.
6. Batasan ruang lingkup: Suatu algoritma yang baik hanya ditujukan untuk memecahkan masalah tertentu. Susunan input harus ditentukan lebih dulu untuk menentukan sifat umum dari algoritma yang bersangkutan.

Dengan memperhatikan hal-hal di atas, dapat disusun algoritma yang efektif dan sesuai dengan kebutuhan dalam menyelesaikan suatu masalah.

1) Cara Penulisan Algoritma

Algoritma adalah suatu cara atau langkah-langkah dalam menyelesaikan suatu masalah. Cara penyajian atau penulisan Algoritma ada 3 cara yaitu :

1. Menggunakan **Pseudocode**

Pseudocode adalah deskripsi dari algoritma pemrograman komputer yang menggunakan konvensi struktural dari bahasa pemrograman tertentu. Tujuannya adalah memudahkan pemahaman manusia terhadap algoritma dengan menggunakan format yang lebih ringkas dan tidak terikat pada sistem tertentu. Pseudocode biasanya tidak mencakup detail-detail seperti deklarasi variabel dan kode yang hanya diperlukan oleh mesin. Terdapat tiga struktur utama dalam

pseudocode, yaitu judul, deskripsi (isi), dan implementasi yang merupakan inti dari algoritma.

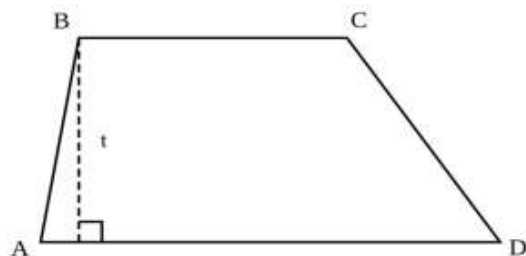
```
Judul
{Berisi Judul Algoritma}
Deskripsi
{Berisi Deklarasi Variabel atau Konstantan}
Implementasi
{Berisi Inti Algoritma}
```

Contoh Penulisan : Menghitung Penjumlahan

```
Mulai
Input a, b
c = a + b
Print "c"
Selesai
```

Pseudocode menggunakan banyak sekali perintah-perintah, yang banyak kita pelajari, misalnya saja, IF, ELSE, THEN, serta menggunakan kode atau tanda- tanda tertentu, seperti tanda underscore (_), tanda koma, titik, tanda kutip, tanda panah kiri dan kanan, serta tanda lainnya.

1) Contoh pseudocode menghitung luas Trapesium



- Diketahui 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:
 $AB = 10$
 $BC = 6$
 $CD = AD = 8$
 $t = 7$

Maka dalam penulisan Pseudocode sebagai berikut:

```

program hitung_luas_trapesium

deklarasi
var luas, ab, cd, t:integer;

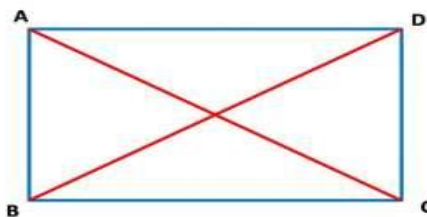
algoritma:
ab <-- 10;
cd <-- 8;
t <-- 7;

luas <-- 1/2 * ab * cd * t;

write <-- luas ;

```

2) Contoh pseudocode menghitung Luas 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);

```

2. Menggunakan **Flowchart**

Flowchart adalah representasi visual dari urutan proses dalam sebuah program, menggunakan simbol-simbol khusus untuk menggambarkan hubungan antara instruksi-instruksi dan proses-proses dalam detail. Dalam perancangan flowchart, tidak ada rumus atau patokan yang mutlak karena flowchart merupakan hasil pemikiran dalam menganalisis suatu masalah komputer. Setiap analisis dapat menghasilkan variasi yang berbeda. Namun, secara umum, setiap

perancangan flowchart terdiri dari tiga bagian utama, yaitu input, proses, dan output.



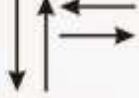








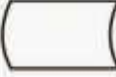






Untuk pengolahan data dengan komputer, dapat dirangkum urutan dasar untuk pemecahan suatu masalah, yaitu :

- **START** : Berisi intruksi untuk persiapan peralatan yang di perlukan sebelum menangani pemecahan masalah
- **READ** : Berisi instruksi untuk membaca data dari suatu peralatan input.
- **PROCESS** : Berisi kegiatan yang berkaitan dengan pemecahan persoalan sesuai dengan data yang dibaca.
- **WRITE** : Berisi instruksi untuk merekam hasil kegiatan ke perlatan output.
- **END** : Mengakhiri kegiatan pengolahan

Bila seorang analis dan programmer akan membuat flowchart, ada beberapa petunjuk yang harus diperhatikan, seperti:

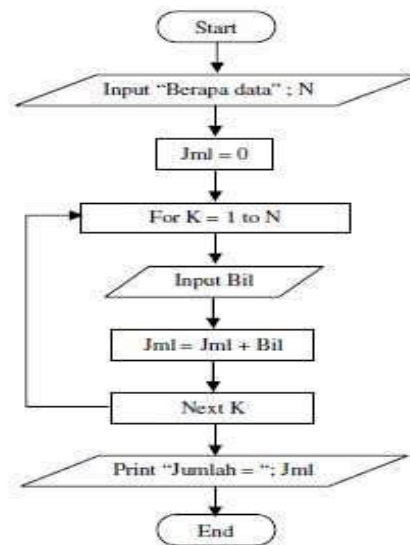
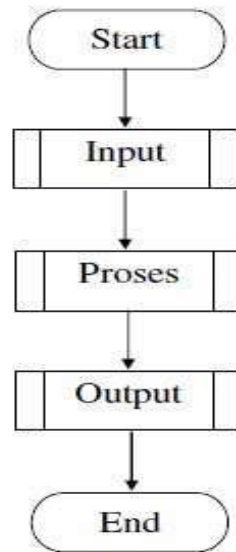
- Flowchart digambarkan dari halaman atas ke bawah dan dari kiri kekanan.
- Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat di mengerti oleh pembacanya
- Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas.
- Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja
- Setiap langkah dari aktivitas harus berada pada urutan yang benar.
- Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.

Berikut ini adalah beberapa simbol standar yang digunakan dalam menggambar suatu flowchart :

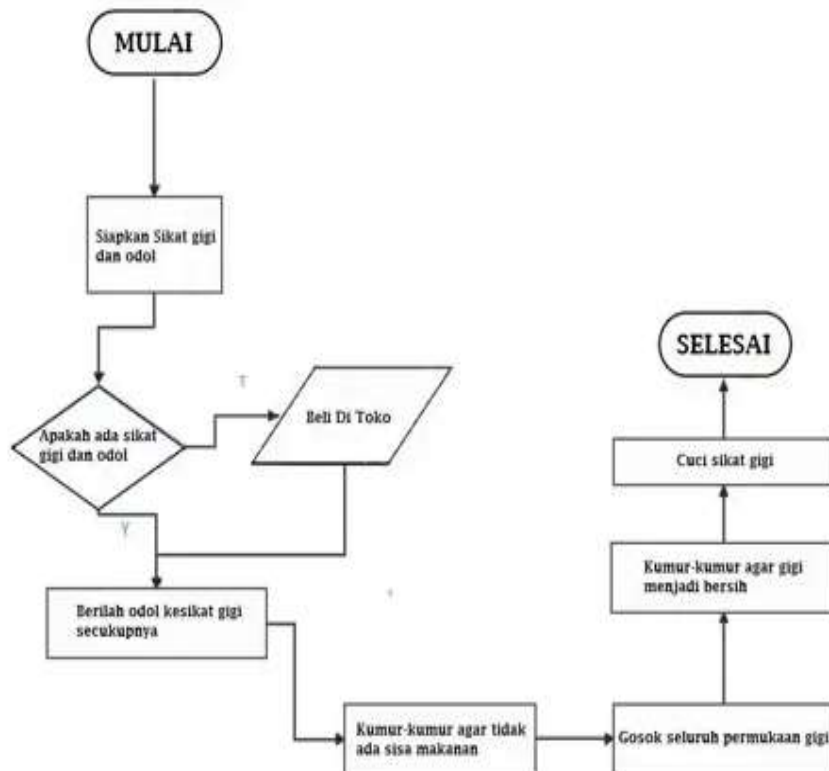
	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau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

Gambar 4. Simbol-simbol flowchart

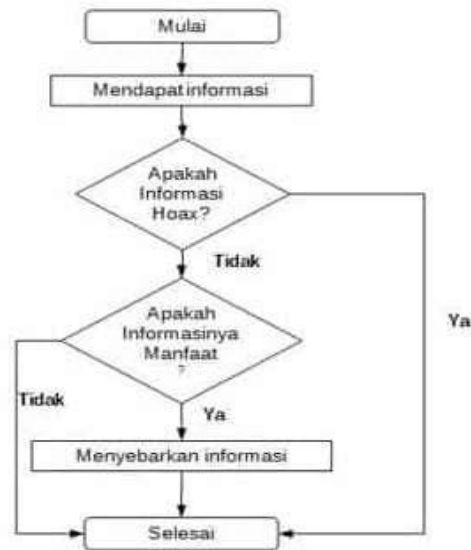
- Contoh Diagram Flowchart



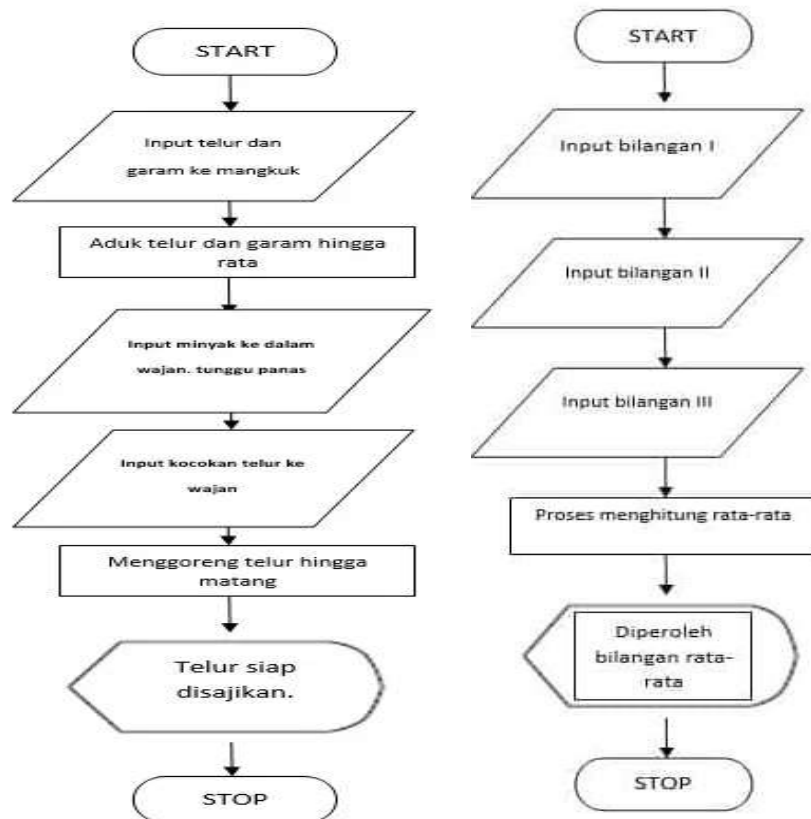
- Contoh Flowchart Menyikat Gigi



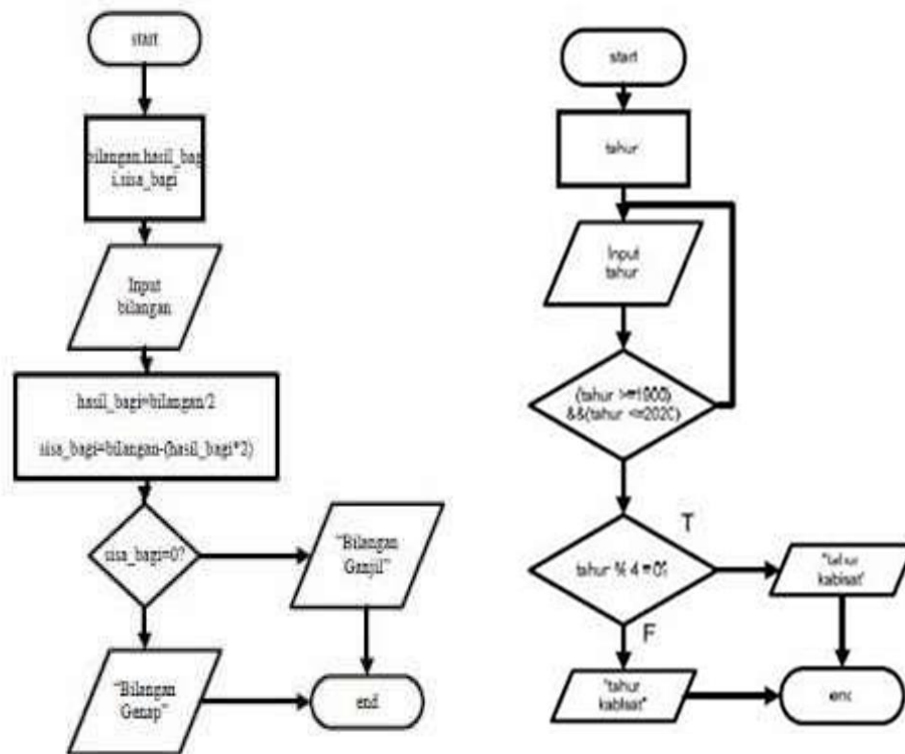
- Contoh Flowchart Menyebarakan Informasi



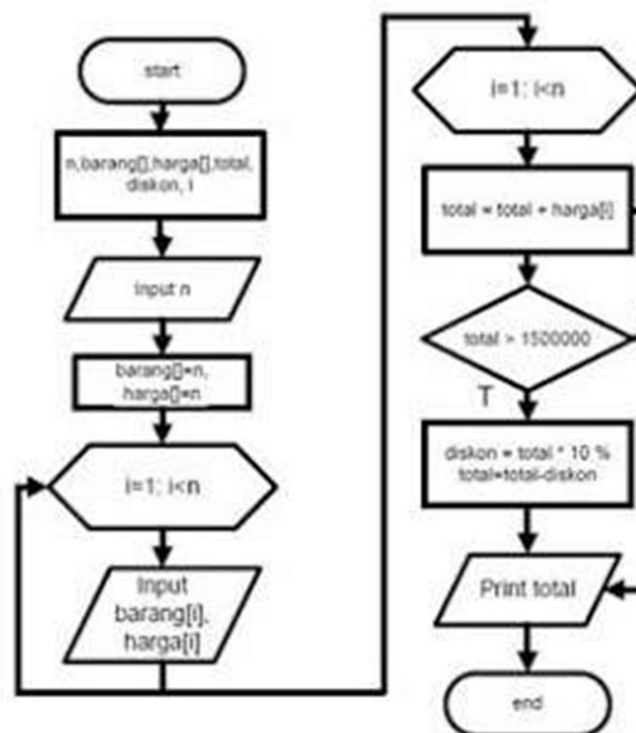
- Contoh Flowchart Menggoreng telur dadar dan menghitung rerata 3 bilangan



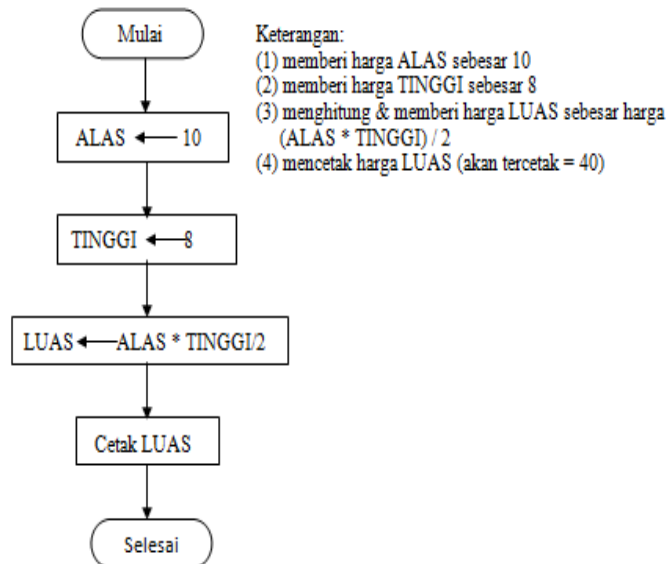
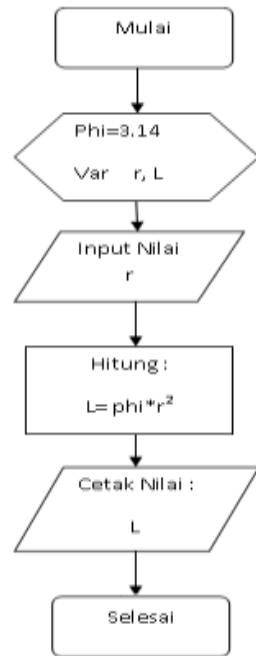
- Contoh Flowchart Menentukan bilangan ganjil atau genap dan menentukan tahun kabisat



- Contoh menghitung harga yang harus dibayar setelah mendapatkan diskon



- Contoh menghitung luas lingkaran dan menghitung luas segitiga



3. Menggunakan Kalimat Deskriptif

Notasi alami atau notasi bahasa deskriptif adalah cara penulisan algoritma dengan kalimat-kalimat deskriptif yang jelas. Tidak ada aturan baku dalam notasi ini, sehingga setiap orang dapat membuat aturan penulisan sendiri. Notasi ini menggunakan kata kerja dan terdiri dari tiga bagian utama.

a. Judul Algoritma (Header)

Bagian ini terdiri dari nama algoritma dan penjelasan tentang

algoritma tersebut. Nama algoritma sebaiknya singkat tetapi menggambarkan tugas yang dilakukan. Di bawah nama algoritma, terdapat penjelasan singkat atau intisari yang disebut spesifikasi algoritma yang dituliskan dalam kurung kurawal ({}). Algoritma harus sesuai dengan spesifikasi yang didefinisikan.

<u>Algoritma Penjumlahan</u>	← <u>Judul Algoritma</u>
{ <u>Menjumlahakan 2 bilangan. Algoritma menerima masukan dua bilangan, kemudian menjumlahkannya dan mencetak hasilnya</u> }	
	← <u>Spesifikasi</u>

b. Bagian Deklarasi

Bagian deklarasi atau kamus dalam algoritma digunakan untuk mendefinisikan semua nama yang digunakan di dalam algoritma, seperti variabel, konstanta, tipe data, prosedur, dan fungsi. Namun, semua nama tersebut harus didefinisikan terlebih dahulu di dalam bagian deklarasi sebelum digunakan dalam algoritma. Penulisan nama-nama dalam deklarasi sebaiknya dikelompokkan berdasarkan jenisnya. Selain itu, pendefinisian nama konstanta juga melibatkan pemberian nilai konstanta, sedangkan pendefinisian nama fungsi atau prosedur juga mencakup spesifikasi dan parameter.

<u>Deklarasi:</u>
<u>bil1, bil2, hasil = int</u>

c. Bagian Deskripsi

Bagian deskripsi merupakan inti dari struktur algoritma. Bagian ini berisi uraian langkah-langkah yang digunakan untuk menyelesaikan masalah. Langkah-langkah tersebut ditulis menggunakan notasi yang umum digunakan dalam penulisan algoritma. Setiap langkah algoritma dibaca dari atas ke bawah, dan urutan penulisan menentukan urutan pelaksanaan perintah. Algoritma dapat terdiri dari tiga struktur dasar, yaitu runtunan, pemilihan, dan pengulangan. Ketiga jenis langkah ini membentuk konstruksi algoritma. Bagian deskripsi adalah tempat di mana tiga struktur algoritma tersebut digunakan.

Deskripsi:

1. Input bil1, bil2
2. Hitung hasil = bil1 + bil2
3. Tampilkan hasil
4. selesai

Algortima Penjumlahan ← Judul Algoritma

{Menjumlahakan 2 bilangan. Algoritma menerima masukan dua bilangan, kemudian menjumlahkannya dan mencetak hasilnya} ← Spesifikasi

Deklarasi:

bil1, bil2, hasil = int

Deskripsi:

1. Input bil1, bil2
2. Hitung hasil = bil1 + bil2
3. Tampilkan hasil
- selesai

C. Program

1) Logika Manusia dan Program

Manusia memiliki kelebihan unik dibandingkan makhluk lain, seperti akal pikiran untuk memilih yang benar dan bermanfaat serta meninggalkan yang tidak bermanfaat. Kemampuan berpikir dan bernalar dengan rasionalitas dan moralitas menjadi ciri khas manusia. Meskipun demikian, emosi dan faktor lain kadang-kadang mempengaruhi kejernihan dan logika berpikir, menyebabkan kesalahan dan kesimpulan yang salah. Oleh karena itu, penting bagi manusia untuk menggunakan pola pikir logis dalam setiap situasi, sesuai dengan hukum logika.



Gambar 5. Logika Manusia

Logika adalah cabang ilmu filsafat yang berfokus pada nalar yang diutarakan dalam kata-kata dan bahasa. Bersama dengan epistemologi, etika, dan estetika, logika menjadi bagian penting dari filsafat. Dalam pembahasan ini,

logika digunakan untuk mempelajari cara berpikir yang lurus, tepat, runtut, dan teratur dalam kehidupan sehari-hari. Meskipun kecerdasan buatan berkembang, logika tetap relevan dan terus dipelajari oleh manusia.

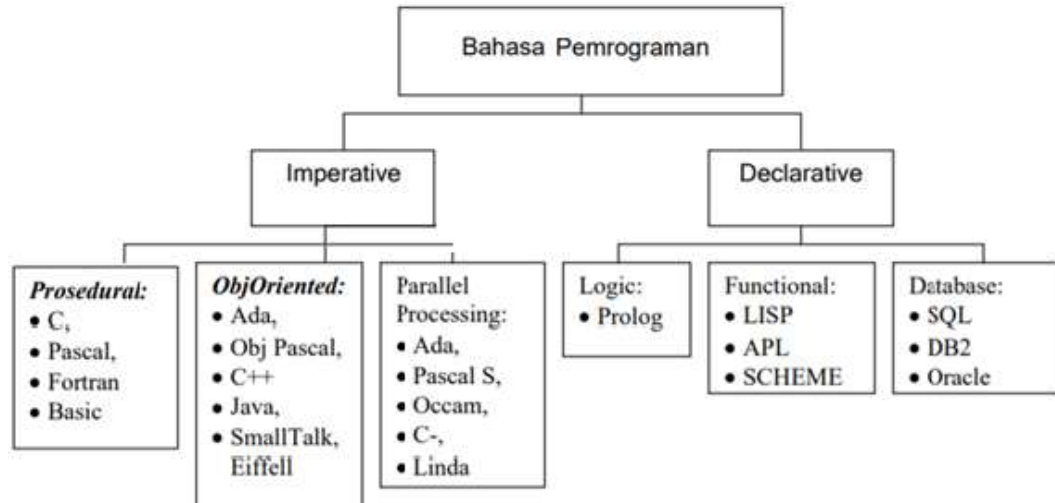
Logika memungkinkan ungkapan kebenaran melalui penalaran logis dengan bukti-bukti. Fungsinya mencegah kesesatan berpikir. Pemahaman logika bermanfaat dalam menjaga keberpikiran sistematis, meningkatkan daya pikir, dan kemampuan berkomunikasi yang cermat, objektif, dan efektif. Hal ini juga menghindari kesalahan berpikir dan menumbuhkan cinta pada kebenaran. Penting untuk mengakui kelebihan otak kanan yang kreatif dan manusiawi serta memperhatikan pendekatan pembelajaran logika yang moderat dan visualisasi yang baik. Logika berperan dalam menarik kesimpulan berdasarkan penalaran deduktif dan induktif.

Penalaran induktif	Penalaran deduktif
Penarikan kesimpulan yang bergerak dari pernyataan benar yang umum ke khusus.	Penarikan kesimpulan yang bergerak dari pernyataan benar yang khusus ke umum.
Pernyataan Premis 1: Semua manusia adalah makhluk mortal. Pernyataan Premis 2: Saya adalah manusia. Kesimpulan: Oleh karena itu, saya adalah makhluk mortal.	Pernyataan Premis 1: Setiap kali saya melihat anjing, mereka memiliki bulu. Pernyataan Premis 2: Setiap kali saya melihat kucing, mereka juga memiliki bulu. Kesimpulan: Jadi, semua hewan mamalia memiliki bulu.

2) Pemrograman dan Bahasa Pemrograman

Pemrograman adalah proses menciptakan program komputer dengan bahasa pemrograman dan algoritma. Ini penting dalam ilmu komputer dan teknologi informasi untuk mengembangkan perangkat lunak, aplikasi, dan sistem. Bahasa pemrograman seperti Python, Java, C++ digunakan sebagai alat komunikasi dengan komputer, dengan sintaks dan aturan tertentu. Pemrograman melibatkan penggunaan algoritma untuk memecahkan masalah dan mencapai tujuan. Struktur

program yang baik dan pengujian program penting dalam pemrograman. Pemrograman digunakan untuk berbagai tujuan seperti aplikasi desktop, web, perangkat lunak bisnis, hingga teknologi canggih seperti AI, IoT, dan komputasi cloud. Ini adalah keterampilan penting yang melibatkan pemecahan masalah, logika, dan kreativitas untuk mengembangkan solusi inovatif dan bermanfaat.



Gambar 6. Bahasa Pemrograman

Terdapat 5 bahasa pemrograman populer yang digunakan oleh developer di seluruh duniaseperti: Java, PHP, Python, C, dan C++.

1. Bahasa Pemrograman Java

Java, bahasa pemrograman populer, dikembangkan pada tahun 1991 dalam proyek Green oleh Patrick Naughton dan James Gosling di Sun Microsystems.



Gambar 7. Logo Java

Kini bahasa pemrograman Java sudah resmi dimiliki oleh perusahaan Oracle setelah Sun Microsystems melepasnya. Java ini merupakan salah satu bahasa pemrograman yang bertenaga untuk mengembangkan aplikasi desktop, web, bahkan mobile.

Sejarah singkat bahasa pemrograman C

Bahasa pemrograman C dikembangkan oleh Dennis Ritchie pada tahun 1972 di Bell Laboratories. Awalnya, C dirancang sebagai bahasa pemrograman sistem yang efisien dan portabel untuk mengembangkan sistem operasi UNIX. Bahasa C didasarkan pada bahasa pemrograman sebelumnya yang disebut B, yang juga dikembangkan oleh Bell Laboratories.

Dalam perkembangannya, C menjadi populer dan banyak digunakan dalam pengembangan perangkat lunak, terutama karena keunggulannya dalam kinerja, fleksibilitas, dan portabilitas. Kelebihan bahasa C memungkinkan programmer untuk mengakses dan mengendalikan perangkat keras secara langsung melalui pemrograman rendah tingkat.

Pada tahun 1989, American National Standards Institute (ANSI) mengeluarkan standar resmi untuk bahasa pemrograman C yang dikenal sebagai ANSI C. Standar ini menetapkan sintaks dan aturan yang harus diikuti oleh implementasi C yang sesuai dengan standar.

C menjadi dasar bagi banyak bahasa pemrograman modern seperti C++, C#, dan Objective-C. Keberhasilan C sebagai bahasa pemrograman sistem telah menjadikannya sebagai salah satu bahasa pemrograman yang paling populer dan diakui di dunia teknologi.

Dengan sejarah dan kelebihanannya, bahasa pemrograman C terus digunakan dalam berbagai aplikasi, termasuk pengembangan perangkat lunak, sistem operasi, mikrokontroler, perangkat embedded, dan banyak lagi.

Bahasa pemrograman Java ini populer di kalangan developer Android karena rata-rata untuk pengembangan aplikasi di platform Android ini menggunakan bahasa pemrograman Java.

2. Bahasa Pemrograman PHP

Awalnya PHP merupakan singkatan dari Personal Home Page dan

seiring berjalannya waktu singkatan PHP menjadi PHP : Hypertext Propocesor. Bahasa pemrograman ini sendiri dikembangkan oleh Rasmus Lerdorf pada tahun 1994.



Gambar 8. Logo PHP

PHP ini merupakan bahasa pemrograman yang populer digunakan untuk membangun website dinamis karena teknologinya yang server side scripting. Dengan teknologi pemrograman web ini membuat skript dikompilasi atau diterjemahkan di server sehingga memungkinkan dalam menghasilkan halaman web yang dinamis.

Bahasa pemrograman ini juga dapat digunakan untuk membuat CMS. Beberapa CMS yang terkenal menggunakan PHP adalah WordPress dan Joomla.

3. Bahasa Pemograman Python

Bahasa pemrograman yang satu ini menjadi populer karena mudah dipelajari dan bertenaga. Python sendiri awalnya dirancang oleh Guido Van Rossum pada tahun 1991. Nama Python sendiri berawal dari acara komedi Monty Python yang hit di tahun 70-an.



Gambar 9. Logo Phyton

Python merupakan bahasa pemrograman interpretatif yang memiliki kapabilitas, kemampuan dengan sintaksis kode yang jelas dan fungsionalitas pustaka standar yang besar serta komprehensif,. Bisa dibilang sintaks-sintaks yang ditawarkan oleh Python ini dapat dengan mudah dipahami dan dipelajari. Salah satu perusahaan besar yang menggunakan Python ini adalah Google.

4. Bahasa Pemrograman C++

C++ merupakan bahasa pemrograman yang dikembangkan oleh Bjarne Stroust pada tahun 1980-an yang merupakan pengembangan dari bahasa C. Salah satu hal paling mendasar yang membedakan C++ dengan bahasa C adalah dukungan terhadap konsep pemrograman berorientasi objek atau Object Oriented Programming.



Gambar 10. Logo C++

Dengan menggunakan C++ ini developer dapat mengembangkan perangkat lunak yang bertenaga seperti game dan aplikasi desktop pada komputer. Selain itu, C++ ini banyak digunakan sebagai dasar dari algoritma dan pemrograman pada beberapa universitas di Indonesia.

5. Bahasa Pemrograman C

Bahasa pemrograman C ini biasa diajarkan pada tingkat awal pada jurusan TI perguruan tinggi di Indonesia sebagai penunjang mata kuliah algoritma dan dasar pemrograman. Bisa dibilang C merupakan bahasa pemrograman yang sudah tua dikembangkan pada tahun 1970-an oleh Dennis Ritchie.



Gambar 11. Logo C

Bahasa C ini banyak digunakan untuk memprogram sistem dan jaringan komputer. Namun developer juga banyak yang mengembangkan perangkat lunak dengan menggunakan C ini. Telah banyak bahasa pemrograman yang merupakan turunan dari C ini seperti C++, C#, dan masih banyak lagi.

3) Hubungan Logika Manusia dan Program

Logika manusia penting dalam merancang dan mengembangkan program komputer yang efisien. Program komputer mengimplementasikan logika melalui instruksi dalam bahasa pemrograman. Pemahaman logika manusia membantu merancang algoritma yang tepat dan memastikan program berjalan dengan benar. Pemahaman program komputer membantu memperkuat kemampuan logika manusia dan meningkatkan efektivitas dalam memecahkan masalah dengan menggunakan komputasi. Gabungan logika manusia dan program komputer memungkinkan pembuatan program yang sesuai dengan tujuan dan kebutuhan..

4) Proses Pembuatan Program

Proses pembuatan program yaitu user/programmer menulis kode sumber pada teks editor misalnya notepad kemudian mengubahnya menjadi bahasa mesin yang bisa dieksekusi oleh CPU. Proses pengubahan kode sumber (source code) menjadi bahasa mesin (machine language) ini terdiri dari dua macam yaitu kompilasi dan interpretasi.

1. Kompilasi

Dalam proses kompilasi semua kode sumber dibaca terlebih dahulu dan jika tidak ada kesalahan dalam menulis program maka akan dibentuk kode mesinnya sehingga program bisa dijalankan. Program yang melakukan tugas

ini disebut Compiler. Program hasil kompilasi akan berbentuk executable. Program bisa langsung dijalankan tanpa harus memiliki Compiler di komputer yang menjalankan program tersebut. Bahasa yang menggunakan teknik kompilasi misalnya bahasa C, C++, Pascal, Assembly dan masih banyak lagi.

2. Interpretasi (Interpretation)

Bahasa yang menggunakan teknik interpretasi akan membaca kode sumber perbaris dan dieksekusi perbaris. Jika ditemukan kesalahan dalam penulisan program maka di baris kesalahan itulah program akan dihentikan. Program yang melakukan tugas ini disebut Interpreter. Pada teknik interpretasi tidak ada akan dihasilkan program standalone, artinya untuk menjalankan program kita harus mempunyai kode sumbernya sekaligus interpreter program tersebut. Bahasa yang menggunakan teknik interpretasi misalnya bahasa Perl, Python, Ruby dan masih banyak lagi.

3. Kompilasi Sekaligus Interpretasi

Ada juga bahasa pemrograman yang menghasilkan programnya dengan teknik kompilasi sekaligus interpretasi. Misalnya bahasa java. Dalam pembuatan program java kode sumber diubah menjadi bytecode. Meskipun tampak seperti bahasa mesin namun ini bukanlah bahasa mesin dan tidak executable. Untuk menjalankan bytecode tersebut kita membutuhkan Java Runtime Environment (JRE) yang bertugas sebagai interpreter sehingga menghasilkan program dari bytecode tersebut.

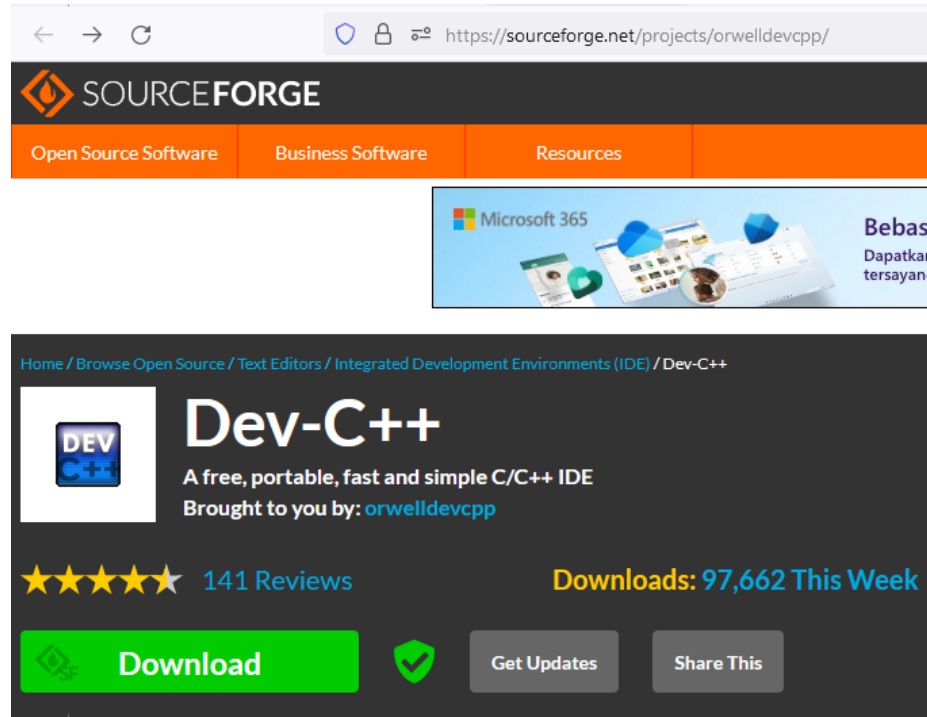
Meskipun setiap bahasa pemrograman dibuat untuk membuat program namun setiap bahasa dibuat dengan tujuan dan fungsi yang berbeda-beda. Misalnya untuk membuat driver hardware kita tidak bisa menggunakan bahasa Visual Basic. Untuk membuat program berbasis sistem seperti driver kita bisa gunakan bahasa C atau Assembly. Contohnya sistem operasi linux yang open source. Jika anda melihat kode sumbernya anda akan menemukan bahwa linux dibuat menggunakan bahasa C. Sedangkan untuk pemrograman desktop kita bisa menggunakan Visual Basic. Bahasa tersebut dirancang oleh Microsoft untuk pemrograman desktop dengan tampilan GUI.

5) Langkah-langkah Instalasi IDE DevC++

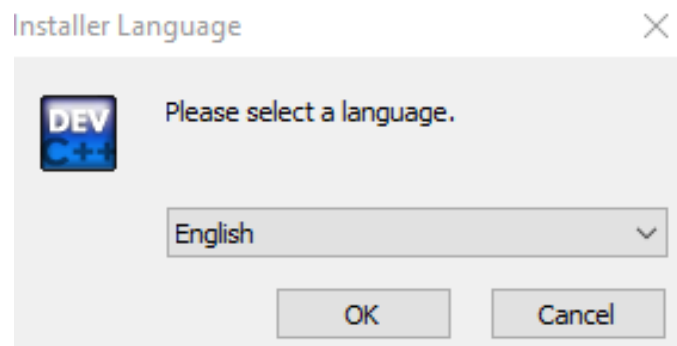
1. **Unduh Installer Dev-C++:** Kunjungi situs resmi Dev-C++,

(<https://sourceforge.net/projects/orwelldvcpp/>),(<https://www.bloodshed.net/>)

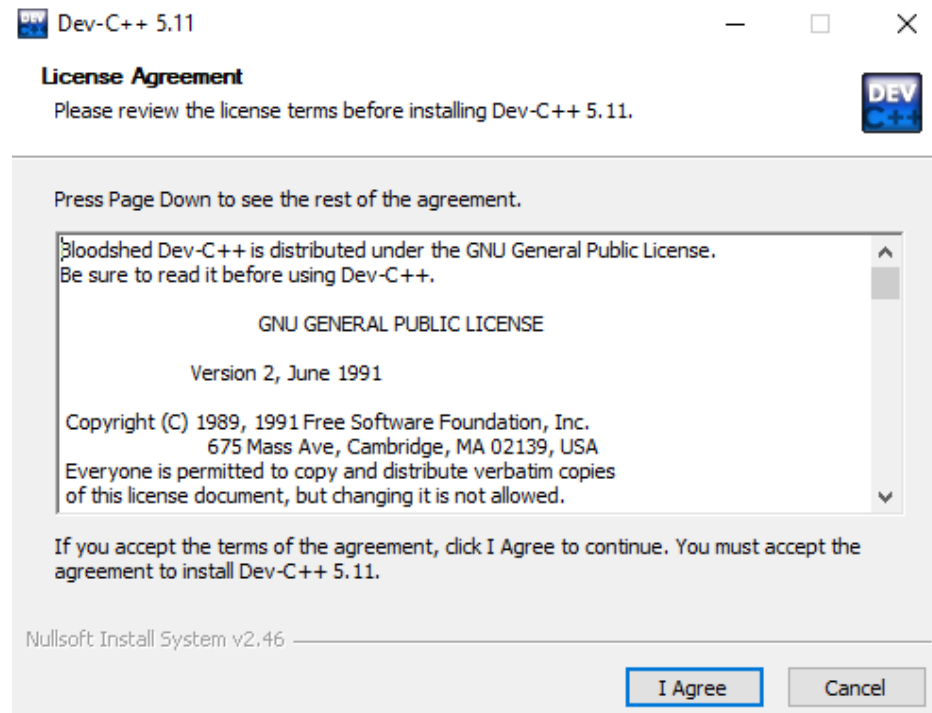
*sumber terpercaya lainnya yang menyediakan installer Dev-C++. Pastikan Anda mengunduh versi terbaru dan aman.



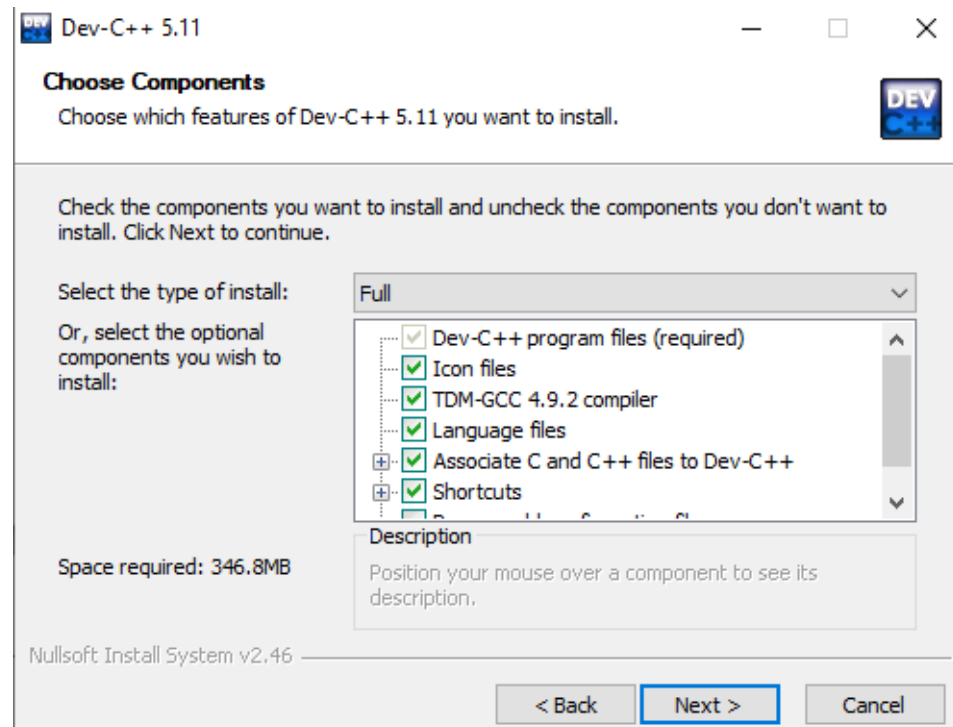
2. **Jalankan Installer:** Setelah mengunduh installer, jalankan file installer tersebut. Anda mungkin perlu memberikan izin administratif tergantung pada sistem operasi yang Anda gunakan.
3. **Pilih Bahasa:** Installer biasanya akan memulai dengan menanyakan bahasa yang ingin Anda gunakan selama proses instalasi. Pilih bahasa yang diinginkan dan lanjutkan.



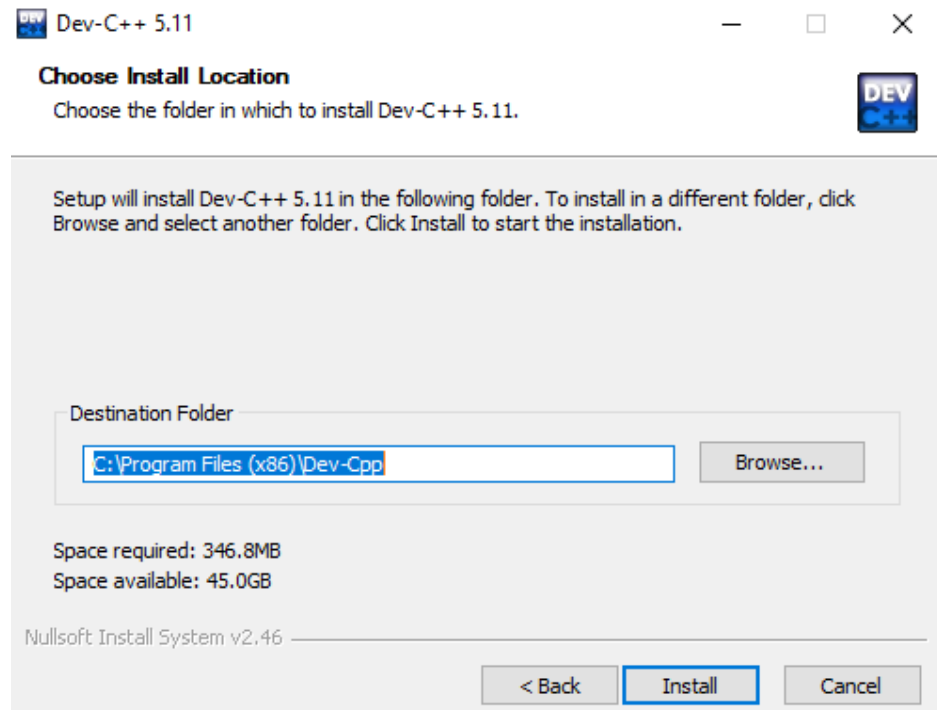
4. **Persetujuan Lisensi:** Bacalah dan terima persetujuan lisensi pengguna akhir (EULA) jika Anda setuju dengan ketentuan yang diberikan.



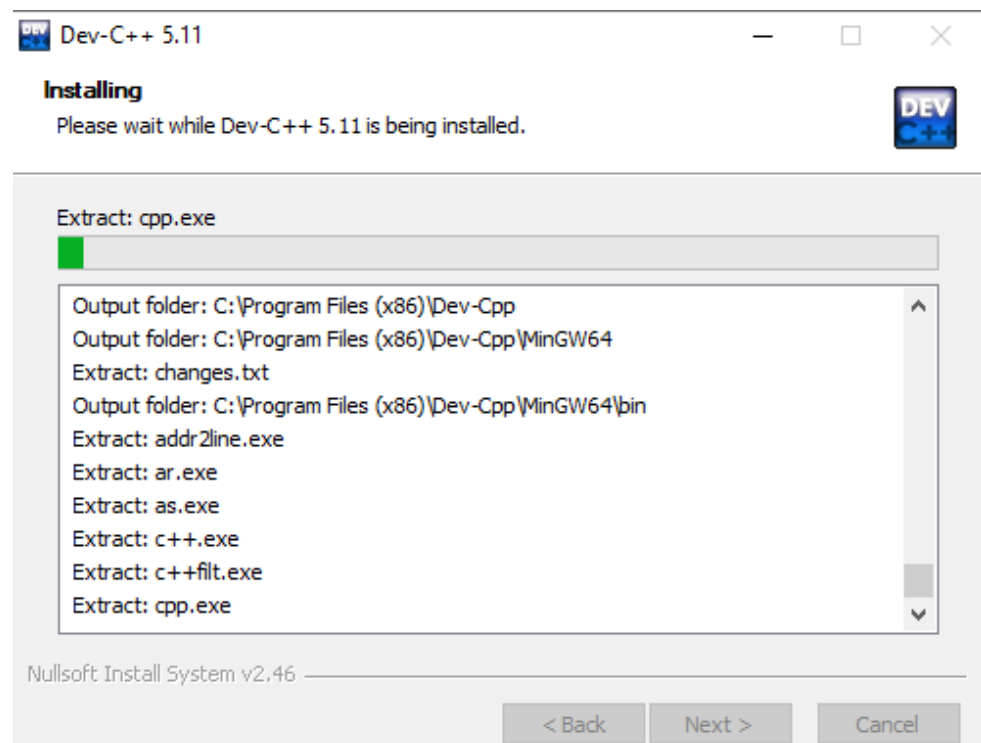
5. **Pilih Komponen:** Di beberapa installer, Anda mungkin memiliki opsi untuk memilih komponen tambahan yang ingin diinstal. Pilih sesuai kebutuhan.



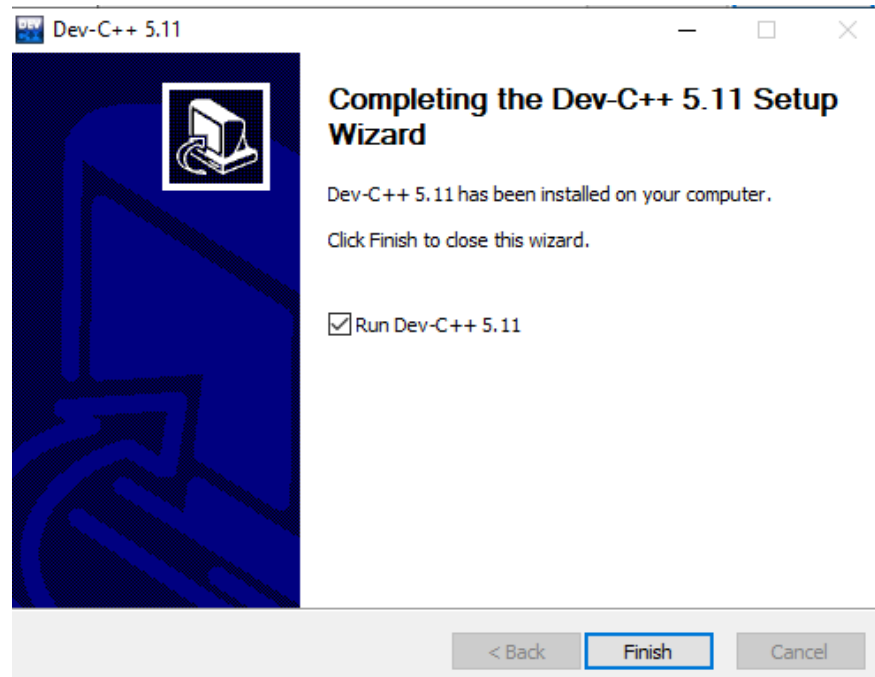
6. **Lokasi Instalasi:** Pilih lokasi di mana Anda ingin menginstal Dev-C++. Anda dapat menggunakan lokasi default atau memilih lokasi khusus.



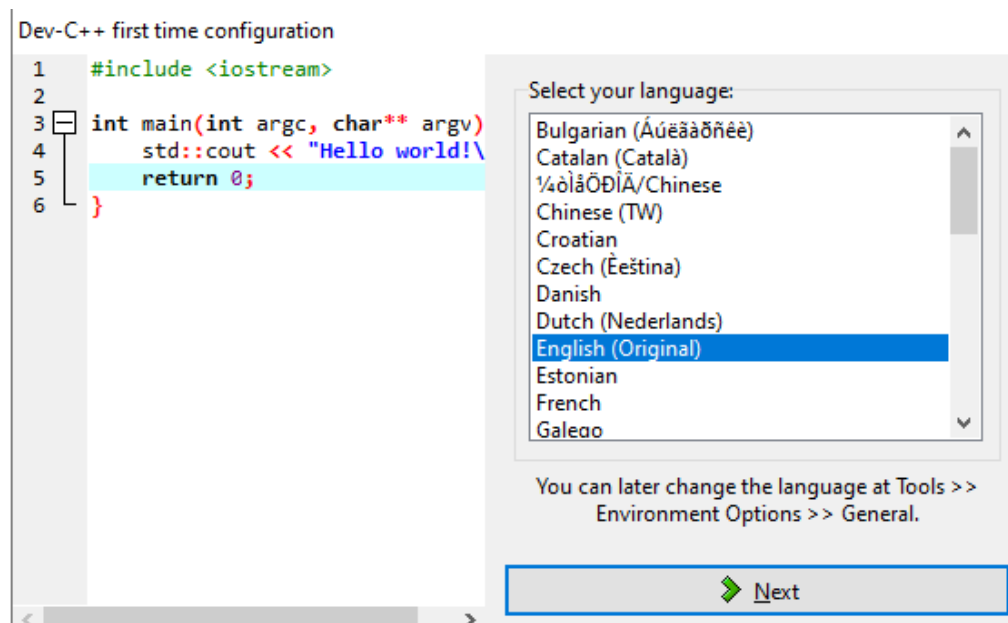
7. **Mulai Instalasi:** Setelah Anda memilih lokasi dan komponen, klik tombol untuk memulai instalasi.



8. **Selesaikan Instalasi:** Tunggu sampai proses instalasi selesai. Setelah selesai, Anda akan menerima pemberitahuan bahwa instalasi telah berhasil.



9. **Buka Dev-C++:** Cari pintasan Dev-C++ di menu Start atau di desktop dan jalankan aplikasinya.



Dev-C++ first time configuration

```
1  #include <iostream>
2
3  int main(int argc, char** argv)
4  {
5      std::cout << "Hello world!\n";
6      return 0;
7  }
```

Select your theme:


Font:

Color:

Icons:



You can later change themes at Tools >> Editor Options >> Fonts/Colors.

 Next

Dev-C++ first time configuration

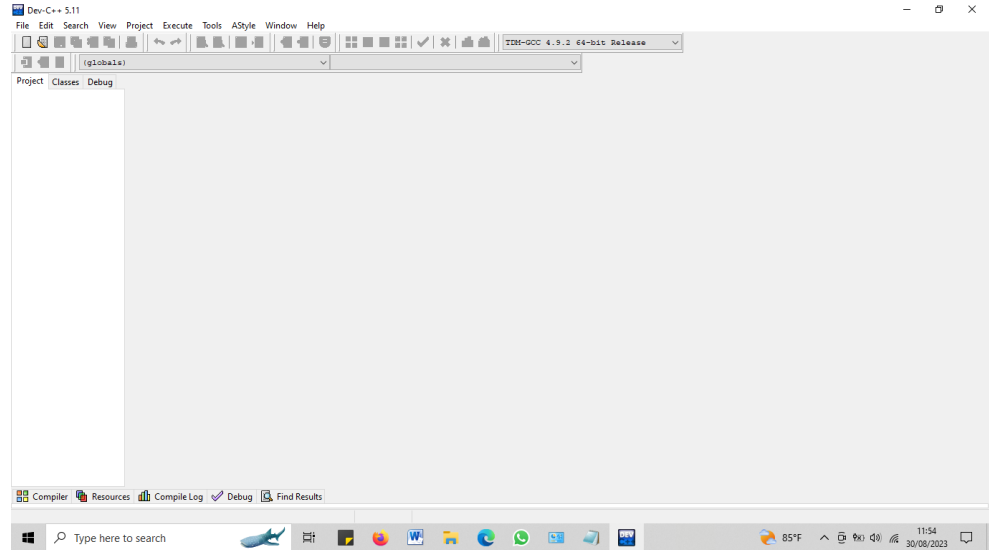
```
1  #include <iostream>
2
3  int main(int argc, char** argv)
4  {
5      std::cout << "Hello world!\n";
6      return 0;
7  }
```

Dev-C++ has been configured successfully.

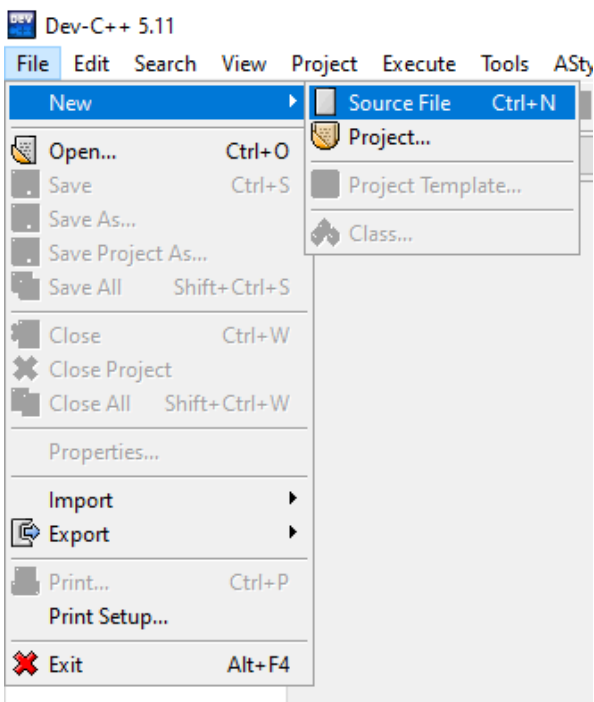
If you need help using Dev-C++, please refer to the Dev-C++ help file in the Help menu or send the developer a message (he doesn't mind!).

You can also download packages (like libraries or tools) to use with Dev-C++ using WebUpdate, which you will find in Tools menu >> Check for Packages.

 OK



10. **Mulai Menggunakan Dev-C++:** Setelah Dev-C++ terbuka, Anda dapat mulai membuat proyek baru, menulis kode C++, mengeditnya, dan mengompilasi serta menjalankan program.



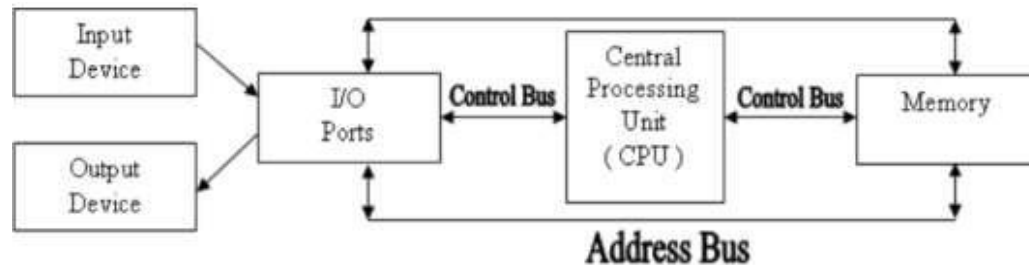
4. Forum Diskusi

Diskusikanlah pentingnya algoritma dalam pemrograman!

C. Penutup

1. Rangkuman

- ✓ Tujuan pokok dari sistem komputer adalah mengolah data untuk menghasilkan informasi sehingga perlu didukung oleh elemen-elemennya. Berikut diagram blok komputer :



- ✓ Program adalah kata, ekspresi, pernyataan atau kombinasi yang disusun dan dirangkai menjadi satu kesatuan prosedur yang menjadi urutan langkah untuk menyesuaikan masalah yang diimplementasikan dengan bahasa pemrograman.
- ✓ Bahasa pemrograman komputer adalah instruksi standar untuk memerintah komputer.
- ✓ Proses pengubahan kode sumber (source code) menjadi bahasa mesin (machine language) ini terdiri dari dua macam yaitu kompilasi dan interpretasi.
- ✓ Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis. Kata logis merupakan kata kunci dalam algoritma. Langkah-langkah dalam algoritma harus logis dan harus dapat ditentukan bernilai salah atau benar.
- ✓ Cara penulisan algoritma :
 - Menggunakan **pseudocode**
 - Menggunakan **flowchart**

2. Tugas

Buatlah algoritma untuk beberapa persoalan di bawah ini dan implementasikan dalam Bahasa C.

1. Menjumlahkan dua buah bilangan yang diinputkan user, lalu menampilkan hasilnya di layar.
2. Menjumlahkan N buah bilangan yang diinputkan user, menggunakan struktur perulangan, lalu menampilkan hasilnya di layar.
3. Menghitung nilai rata-rata dari sejumlah bilangan yang diinputkan user, lalu menampilkan hasilnya di layar.
4. Mencari nilai akar pangkat dua dari sebuah bilangan dan menampilkannya di layar
5. Mencari nilai akar pangkat x dari sebuah bilangan dan menampilkannya ke layar

6. Mempertukarkan isi dua buah variabel menggunakan variabel perantara dan menampilkan ke layar saat sebelum bertukar dan setelah bertukar isi
7. Sama seperti soal nomor 6, tetapi tanpa menggunakan variabel perantara.
8. Mencari nilai akar-akar dari suatu persamaan kuadrat dan menampilkan hasilnya ke layar.

Daftar Pustaka

Utama
✓ Rinaldi Munir. 2016. <i>Algoritma dan Pemrograman</i> . Bandung. Informatika ITB
✓ Noel Kalicharan. 2015. <i>Learn to Program with C</i> . New York, Springer-Science
✓ Harry H. Chaudhary. 2014. <i>C Programming Step by Step</i> . LLC USA. Amazon Inc.
Pendukung
✓ Mike McGrath. 2015. <i>Coding for Beginners</i> . Leamington Spa. Easy Step Limited.
✓ Dan Gookin. 2014. <i>Beginning Programming with C for Dummies</i> . New Jersey. John Wiley & Sons.
✓ www.tutorialspoint.com
✓ www.javatpoint.com
✓ www.programiz.com