



FEBRUARI 2022

MODUL PRAKTIKUM ALGORITMA

DEV C++

SISTEM INFORMASI

Ir. Muhammad Amrin Lubis, M.Sc
Suardinata, M.Kom

DAFTAR ISI

Daftar Gambar	6
MODUL 1 PENGANTAR ALGORITMA	8
1. Tujuan	8
2. Dasar Teori	8
IDE Dev C++	8
Teks Editor	9
CTRL +	10
Compiler C++	10
Bagaimana Compiler C++ Bekerja	10
Struktur Bahasa Program C++	11
Keyword C++	12
Compiler	13
Algoritma	13
Ciri-ciri Algoritma	14
Program dan Bahasa Pemrograman	14
Langkah-langkah Membangun Program	15
Defenisi Masalah	15
Penyusunan Algoritma	16
Dokumentasi	16
Penyajian Algoritma	17
Contoh soal algoritma	17
3. Tugas Pendahuluan	18
4. Percobaan	19
5. Laporan Praktikum	20
MODUL 2 OPERATOR, TYPE DATA DAN VARIABLE	21
1. Tujuan	21
2. Dasar Teori	21
Operator	21
Operator Aritmatika	21
Operator Relational	22
Operator Logika	23
Contoh-1.	23

Type data.....	24
Variabel.....	24
Contoh penulisan variabel.....	25
Contoh Program C++ Type Data	25
Output	26
3. Tugas Pendahuluan	26
Contoh logika AND.....	26
Program Flowchart AND	27
Algoritma AND	27
Program C++.....	28
Output	28
Contoh logika OR	29
Program Flowchart OR.....	29
Algoritma	30
Program C++.....	30
Output	31
4. Percobaan	31
5. Laporan Praktikum	32
MODUL 3 PROGRAM FLOWCHART.....	33
1. Tujuan.....	33
2. Dasar Teori.....	33
Simbol Dasar Program Flowchart.....	34
Simbol Lanjutan Program Flowchart.....	37
Flowchart IF	37
Flowchart IF..ELSE.....	39
Contoh Soal.....	39
3. Tugas Pendahuluan	40
Contoh-1 IF.....	40
Program Flowchart IF- Ujian Algoritma	41
Algoritma	41
Program C++ Ujian Algoritma.....	42
Output Program Ujian Algoritma.....	42
Contoh-2 IF..ELSE	43

Program Flowchart IF..ELSE - Ujian Algoritma	43
Algoritma	44
Program C++ IF..ELSE.....	45
Output Program	46
4. Percobaan	46
5. Laporan Praktikum	47
MODUL 4 FUNCTION DAN PROCEDURE BAGIAN-1.....	48
1. Tujuan	48
2. Dasar Teori.....	48
Konsep Kerja Function	48
3. Tugas Pendahuluan	49
Contoh.....	49
Program Flowchart.....	49
Algoritma	49
Program C++ HITUNGUSIA	51
Output	52
4. Percobaan	52
5. Laporan Praktikum	53
MODUL 5 FUNCTION DAN PROCEDURE BAGIAN-2.....	54
1. Tujuan	54
2. Dasar Teori.....	54
Procedure.....	54
3. Tugas Pendahuluan	55
Contoh.....	55
Program Flowchart.....	55
Algoritma	55
Program C++ Keliling Segi Tiga	56
Output	56
4. Percobaan	57
5. Laporan Praktikum	58
MODUL 6 STUKTUR KONTROL IF DAN IF ELSE.....	59
1. Tujuan	59
2. Dasar Teori.....	59

3. Tugas Pendahuluan	62
Contoh Switch Case	62
4. Percobaan	63
2. Laporan Praktikum	64
MODUL 7 STUKTUR KONTROL LOOP	65
1. Tujuan	65
2. Dasar Teori	65
Loop While	66
Loop For	67
Loop Do..While	68
Nested Loop	70
3. Tugas Pendahuluan	71
Contoh Loop while	71
Contoh Loop for	72
Contoh Loop do while	74
Contoh Nested Loop	76
4. Percobaan	78
5. Laporan Praktikum	80
MODUL 8 ARRAY	81
1. Tujuan	81
2. Dasar Teori	81
Array Dimensi 1	81
Syntax	81
3. Tugas Pendahuluan	82
4. Percobaan	85
5. Laporan Praktikum	86
MODUL 9 SORTING	88
1. Tujuan	88
2. Dasar Teori	88
3. Tugas Pendahuluan	89
Bubble Sort	89
Flowchart Bubble Sort	93
Coding C++ Bubble Sort	94

4. Percobaan	95
5. Laporan Praktikum	95
MODUL 10 SEARCHING	96
1. Tujuan	96
2. Dasar Teori	96
3. Tugas Pendahuluan	97
4. Percobaan	100
5. Laporan Praktikum	101
Referensi	102

Daftar Gambar

Gambar 1 Logo Dev C++	9
Gambar 2 IDE Dev C++	9
Gambar 3 Sistem Kerja Compiler C++.....	11
Gambar 4 Struktur Bangun C++	12
Gambar 5 Program C++ Hello World.....	13
Gambar 6 Output Teks: Sistem Informasi	19
Gambar 7 Output Program Menampilkan Ukuran Penggunaan Memory Untuk Data	26
Gambar 8 Flowchart Hasil Ujian Operator AND	27
Gambar 9 Output Program Functionand, mid=70 dan uas=100 Hasil LULUS	29
Gambar 10 Flowchart Hasil Ujian Operator OR	29
Gambar 11 Output Program FunctionOR, mid=55 dan uas=90 Hasil LULUS	31
Gambar 12 Simbol Dasara Program Flowchart	34
Gambar 13 IF-Logika Percabangan	38
Gambar 14 Structure If else atau Nested If	40
Gambar 15 Flowhart Ujian Algritma	41
Gambar 16 Output FunctionHasilTes, nilai uas =100.....	43
Gambar 17 Output FunctionHasilTes, nilai mid=61.....	43
Gambar 18 IF ..ELSE Hasil Ujian	44
Gambar 19 Output Nilai mid = 97, Keterangan Lulus.....	46
Gambar 20 Output Nilai mid = 63 dan nilai uas = 60, Keterangan Gagal.....	46
Gambar 21 Function	48
Gambar 22 Flowchart Hitung Usia	49
Gambar 23 Output Function HITUNGUSIA	52
Gambar 24 Flowchart Procedure Keliling Segi 3	55
Gambar 25 Procedure KELILINGSEGI3, Parameter sisi bernilai 5	57
Gambar 26 Struktur IF	59
Gambar 27 Struktur IF..ELSE	60
Gambar 28 Flowchart Switch-Case	61
Gambar 29 Flowhart Loop While	66
Gambar 30 FFlowchart Loop For	68
Gambar 31 Flowhart do-while	69
Gambar 32 Nested Loop While	70
Gambar 33 Flowchart Loop-While Cetak Angka 1..5	71
Gambar 34 Output CETAKANGKA: 1 2 3 4 5.....	72
Gambar 35 Flowchart Loop-For, Cetak Angka 1 sd 5.....	73
Gambar 36 Output Angka 1 s.d 5	74
Gambar 37 Flowchart Loop-Do-While, Cetak Angka 1 sd 5	75
Gambar 38 Output Cetak Angka 1 sd 5	76
Gambar 39 Flowchart Nested Loop While, Cetak Pola Bintang 5 Baris dan 5 Kolom	77
Gambar 40 Output Cetak * Pola 5 Baris 5 Kolom.....	78

Gambar 41 Array String Menetak Nama Buah Tropis	83
Gambar 42 Array Numeric Mencetak Nilai Algoritma	85
Gambar 43 Data Acak.....	91
Gambar 44 Terjadi Data Swap Indeks 2 dan indeks 1	91
Gambar 45 Terjadi Data Swap Indeks 3 dan indeks 2	91
Gambar 46 Tidak Terjadi Data Swap	91
Gambar 47 Program Flowchart Ascending menggunakan metode Bubble Sort	93

MODUL 1 PENGANTAR ALGORITMA

1. Tujuan

- Mengetahui IDE Dev C++
- Mengetahui bahasa pemrograman C++
- Memahami struktur bahasa pemrograman C++
- Mampu menuliskan bahasa pemrograman C++

2. Dasar Teori

Bahasa C++ adalah bahasa pemrograman berorientasi objek & merupakan kombinasi dari bahasa tingkat rendah & tingkat tinggi – Bahasa Tingkat Menengah. Bahasa pemrograman dibuat, dirancang & dikembangkan oleh Ilmuwan Komputer Denmark – Bjarne Stroustrup di Bell Telephone Laboratories (sekarang dikenal sebagai Nokia Bell Labs) di Murray Hill, New Jersey. Karena dia menginginkan bahasa yang fleksibel & dinamis yang mirip dengan C dengan semua fiturnya, tetapi dengan tambahan pemeriksaan tipe aktif, basic inheritance, default functioning argument, classes, inlining, dll. dan karenanya C dengan Kelas (C++) diluncurkan.

C++ awalnya dikenal sebagai "C dengan kelas," dan berganti nama menjadi C++ pada tahun 1983. ++ adalah singkatan untuk menambahkan satu ke variasi dalam pemrograman; oleh karena itu C++ secara kasar berarti bahwa "satu lebih tinggi dari C."

IDE Dev C++

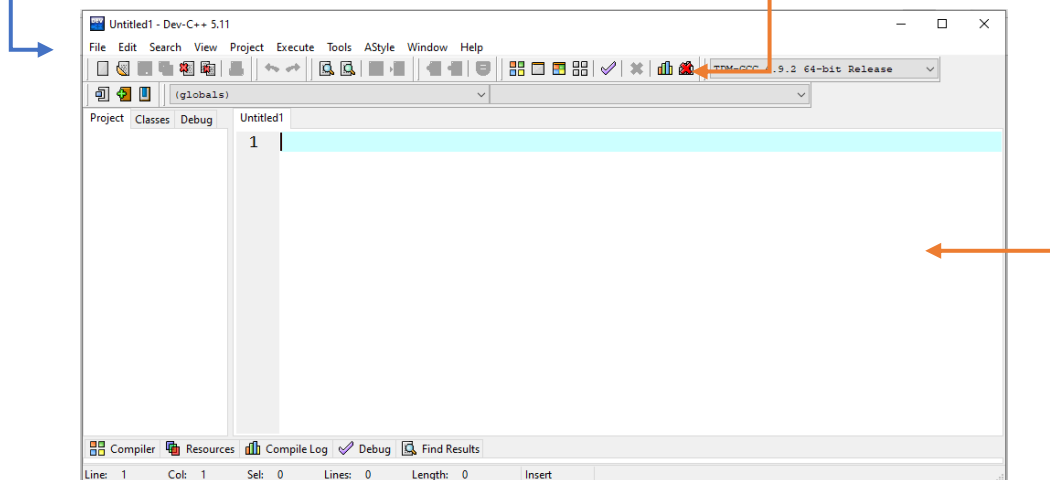
Dev C++ adalah sebuah IDE (*Integrated Development Environment*) C/ C++ yang sudah dilengkapi dengan **TDM-GCC Compiler** (bagian dari *GNU Compiler Collection / GCC*). **Dev C++** merupakan IDE gratis dan *full featur* yang didistribusikan dibawah lisensi *GNU General Public License* untuk pemrograman C dan C++.



Gambar 1 Logo Dev C++

IDE (*Interface Development Enviroment*) dibawah dapat dilihat beberapa bagian pada lingkungan pengembangan program.

1. Top Menu Menu *dropdown* yang terdapat semua menu yang dibutuhkan pada aplikasi IDE.
2. Menu Bar Bagian menu yang bersifat jalan pintas berbentuk gambar. Untuk membuat file baru, menjalankan program tersedia pada bagian ini.
3. Teks editor Bagian programmer menuliskan kode program C++



Gambar 2 IDE Dev C++

Teks Editor

Teks **editor** adalah jenis program yang digunakan untuk mengedit atau menulis teks. Di modul Algoritma ini menggunakan teks editor untuk mengetik program C++. Ekstensi normal file teks adalah (.txt) tetapi file teks yang berisi program C++ harus disimpan dengan ekstensi '.CPP' atau '.C'. File yang diakhiri dengan ekstensi '.CPP' dan '.C' disebut file *source code* dan seharusnya berisi *source code* yang ditulis dalam bahasa pemrograman C++. Ekstensi ini membantu kompiler untuk mengidentifikasi bahwa file tersebut berisi program C++.

Sebelum memulai pemrograman dengan C++, seseorang harus menginstal editor teks untuk menulis program.

CTRL +

Pengoperasian Text Editor selain menggunakan menu Bar atau icon, bisa menggunakan tombol CTRL + pada keyboard. Cara penggunaanya tekan bersamaan tombol CTRL + tombol tertentu, dibawah ini ada beberapa fungsi tombol CTRL + tombol tertentu

1. Save → CTRL + s, berfungsi untuk menyimpan source code, kode program C++ kedalam disk
2. Copy → CTRL + c, berfungsi untuk mengkopi source code (menandai *source code* yang akan di kopi
3. Paste → CTRL + v, berfungsi untuk menggandakan source code
4. Find → CTRL + f, berfungsi untuk mencari teks
5. Find and replace → CTRL + f, pilih tab *replace*, berfungsi untuk mencari teks dan menggantikannya dengan teks lain secara otomatis

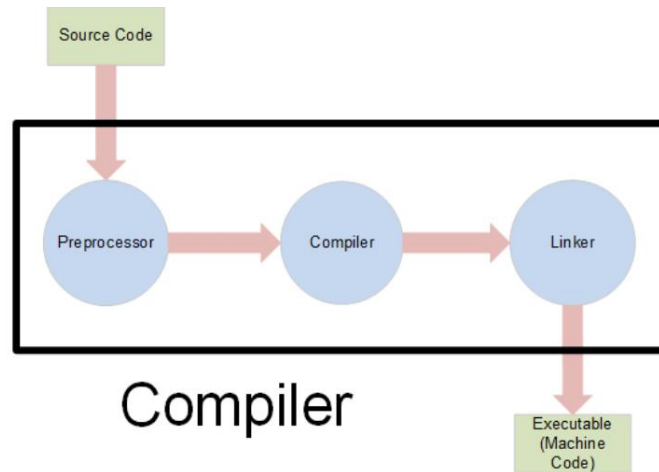
Compiler C++

Setelah menginstal teks editor dan mengetik dan menyimpan program C++ dalam file dengan ekstensi '.CPP', selanjutnya diperlukan compiler C++ untuk mengkompilasi file ini. **Compiler** adalah program komputer yang mengubah bahasa tingkat tinggi menjadi bahasa tingkat rendah yang dapat dimengerti mesin. Dengan kata lain, *programmer* dapat mengatakan bahwa itu mengubah *source code* yang ditulis dalam bahasa pemrograman menjadi bahasa komputer lain yang dimengerti komputer. Untuk mengkompilasi program C++ diperlukan compiler C++ yang akan mengubah *source code* yang ditulis dalam C++ menjadi kode mesin.

Bagaimana Compiler C++ Bekerja

Proses menerjemahkan *source code* C++ ke dalam program yang dapat dieksekusi disebut "mengkompilasi program" atau hanya "mengkompilasi." Sekilas terlihat seperti proses kompilasi

sebagai tindakan tunggal dan umumnya menyebutnya seperti itu. Namun demikian, kompiler C++ modern terdiri dari setidaknya tiga program terpisah:



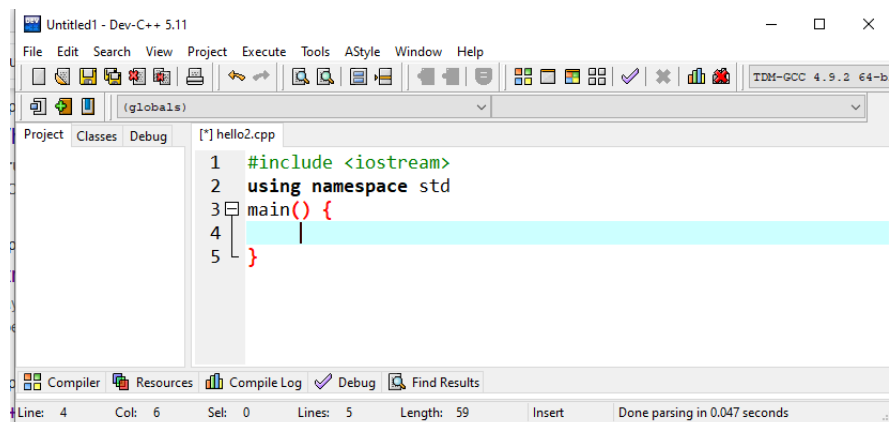
Gambar 3 Sistem Kerja Compiler C++

1. **Source code** atau kode program C++ ditulis menggunakan teks editor
2. **Preprocessor**, menangani statement atau baris kode yang dimulai dengan karakter "#", yang disebut "directive preprocessor." Perhatikan bahwa arahan bukanlah pernyataan C++ (dan karenanya tidak diakhiri dengan titik koma) melainkan menginstruksikan preprosesor untuk melakukan beberapa tindakan. **Preprocessor** membaca dan memproses setiap isi file satu per satu dari atas ke bawah.
3. **Compiler**, menerjemahkan kode sumber (*source code*) C++ ke dalam kode mesin (*machine code*) yang "dipahami" dan dapat dieksekusi oleh komputer tertentu. Program bersumber dalam berbagai ukuran, dari beberapa puluh baris kode hingga jutaan baris
4. **Linker**, adalah komponen ketiga dan terakhir dari sistem kompiler lengkap. Dibutuhkan file objek yang dibuat oleh *compiler* dan menghubungkannya bersama-sama, bersama dengan *library code* dan file *runtime*, untuk membentuk program yang lengkap dan dapat dieksekusi. Nama file yang dapat dieksekusi tergantung pada sistem operasi hosting: Pada komputer Windows, tautan menghasilkan file yang namanya diakhiri dengan ekstensi ".exe".

Struktur Bahasa Program C++

Struktur bangun program C++, paling sedikit terdiri dari 3 bagian lihat gambar 4 sebagai berikut:

1. Menyertakan library, tergantung kebutuhan aplikasi ingin menggunakan fungsi apa, lihat baris 1
2. Menyertakan using namespace std, lihat baris ke 2
3. Menggunakan *keyword* main() dan diikuti tanda {, sebagai tanda awal program dan }, akhir dari program C++



Gambar 4 Struktur Bangun C++

Gambar 4 diatas tidak menghasilkan output apa pun karena belum ada perintah untuk mencetak. Perlu ditambah statement (perintah) *cout* sehingga diubah menjadi gambar agar bisa menampilkan output, luaran Hello World

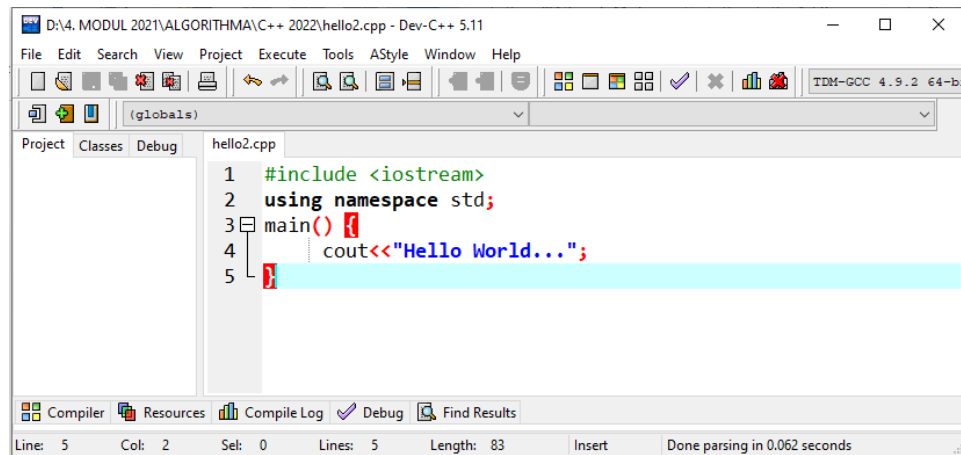
Keyword C++

Keyword adalah kosakata. Istilah *keyword* digunakan dalam Bahasa pemrograman. Bagaimana *keyword* digunakan agar memenuhi syarat dalam pemrograman disebut *syntax* atau tata Bahasa. *Keyword* yang ditulis membentuk satu perintah diakhiri tanda titik koma (;) disebut *statement*. Makna yang tertulis dalam satu statement disebut *semantic*.

Contoh

cout << " Hello World "; → cout=keyword

statement ini berfungsi cetak teks Hello World, diujung statement ini diakhiri tanda titik koma



Gambar 5 Program C++ Hello World

Compiler

Proses menterjemahkan source code menjadi machine code yang dimengerti computer disebut kompilasi, sedangkan alat penterjemahnya disebut Compiler

Untuk melaksanakan kompilasi pada IDE Dev C++ bisa dipilih tombol dibawah ini atau menggunakan menu bar, icon.

F9 → compile, menterjemahkan source code menjadi machine code

F10 → run, menjalankan, eksekusi program yang telah di-compile agar tampak output program

F11 → compile and run, menterjemahkan dan menjalankan program C++

Algoritma

Algoritma berasal dari kata **algorism** merupakan nama penulis buku Arab yang terkenal yaitu Abu Ja'far Muhammad ibnu Musa al-Khuwarizmi, kata al-Khuwarizmi dibaca oleh orang Barat menjadi algorism. Al-Khuwarizmi menulis buku berjudul Kitab al jabar wal-muqabala, yang artinya Buku Pemugaran dan Pengurangan = *the Book of Restoration and Reduction*.

Kata *algorism* sering dikelirukan dengan *arithmetic*, sehingga akhiran *-sm* berubah menjadi *-thm*. Akhirnya kata *algorithm* berangsur-angsur dipakai sebagai metoda perhitungan (komputasi) secara umum, sehingga kehilangan makna aslinya. Dalam bahasa Indonesia, kata *algorithm* diserap menjadi *algoritma* (Munir, 2000). Kata Algoritma dalam bahasa inggris ditulis Algorithm.

Algoritma adalah urutan langkah-langkah logis penyelesaian masalah disusun secara sistematis. Menurut Kamus Besar Bahasa Indonesia terbitan Balai Pustaka 1998 Algoritma adalah urutan logis pengambilan putusan untuk pemecahan masalah.

Ciri-ciri Algoritma

Ciri-ciri algoritma menurut **Donald E. Knuth** dalam bukunya *The Art of Computer Programming*:

1. Algoritma harus berhenti setelah mengerjakan sejumlah langkah terbatas.
2. Setiap langkah harus didefinisikan dengan tepat dan tidak berarti-dua (*ambiguous*).
Algoritma memiliki nol atau lebih masukan (*input*).
3. Algoritma memiliki nol atau lebih keluaran (*output*).
4. Algoritma harus *effective*.
5. Setiap langkah harus sederhana sehingga dapat dikerjakan dalam sejumlah waktu yang masuk akal.

Program dan Bahasa Pemrograman

Program adalah kata, ekspresi, pernyataan atau kombinasinya yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah yang diimplementasikan menggunakan bahasa pemrograman sehingga dapat dieksekusi oleh komputer.

Bahasa Pemrograman atau *programming language* merupakan prosedur/tata cara penulisan program. Pada bahasa pemrograman ada dua faktor penting, yaitu: **Syntax** dan **Semantik**.

Syntax adalah aturan-aturan gramatikal yang mengatur tata cara penulisan kata, ekspresi dan pernyataan. Syntax analog dengan tata bahasa

Semantik adalah aturan-aturan untuk menyatakan suatu arti.

Pemrograman merupakan proses pengimplementasian program ke dalam bahasa pemrograman.

Pemrograman Terstruktur merupakan proses pengimplementasian program yang memiliki rancang bangun terstruktur dan tidak berbelit-belit sehingga mudah ditelusuri, dipahami dan dikembangkan oleh siapa saja.

Programmer adalah orang yang bekerja menyusun suatu program.

Kriteria programmer yang baik yaitu

1. Mampu menyusun algoritma dengan baik.
2. Menguasai bahasa dan teknik penulisan program yang baik.
3. Dapat bekerjasama dalam suatu tim kerja.
4. Dapat bekerjasama secara efisien dan tepat waktu.

Langkah-langkah Membangun Program

1. Definisikan masalah
2. Analisis kebutuhan
3. Penyusunan algoritma
4. Pengkodean/pemrograman
5. Testing dan Debugging
6. Dokumentasi
7. Pemeliharaan

Untuk masalah yang sederhana, pemrogram tidak perlu melalui ketujuh tahap tersebut, tapi cukup dengan:

1. Mengidentifikasi masalah (input, proses dan output).
2. Menyusun algoritma
3. Pengkodean/pemrograman
4. Testing dan Debugging

Defenisi Masalah

Bertujuan mendapatkan pengertian atau pemahaman yang lebih mendalam tentang masalah yang ada.

Langkah-langkah:

1. Pahami **masalah dengan tepat** (berusahalah memahami persepsi permasalahan dari pihak pemesan program).
2. Identifikasi **dan definisi masalah** (definisikan masalah secara rinci dan ungkapkan kemungkinan masalah lain lain yang dapat muncul di seputar masalah utama, batasi masalah)

Penyusunan Algoritma

Algoritma dibuat dengan tujuan menyelesaikan masalah.

Pembuatan algoritma tidaklah sekali dan langsung jadi, harus dikaji terus-menerus sehingga didapat algoritma yang paling lengkap, tepat, benar dan relevan.

Algoritma yang telah tersusun harus dikoreksi kembali, jika terdapat kesalahan maka harus direvisi.

Dokumentasi

Dalam ilmu *programming*, dokumentasi memegang peranan penting. Dokumentasi merupakan salah satu alasan utama yang membedakan *programmer* amatir dengan *programmer* yang profesional. Programmer amatir biasanya menulis program hanya untuk bisa dimengerti oleh dirinya sendiri.

Programmer profesional menulis program untuk dimengerti oleh banyak orang. Pemakai dokumentasi yaitu

1. Programmer, dokumentasi yang dibutuhkannya:
 - a. Internal: gunakan nama prosedur/fungsi, nama variabel yang menggambarkan kegunaannya; berikan keterangan-keterangan di dalam program.
 - b. Eksternal: listing program; gambaran modul-modul program secara hirarkis; spesifikasi aliran data (bagaimana data diterima, file-file yang berhubungan, struktur data, dll).
2. Operator, dokumentasi yang dibutuhkan: tata cara pengoperasian program, biasanya meliputi:
 - a. Nama file yang akan diakses
 - b. Cara mengeksekusi program
 - c. Cara memahami tampilan layar

- d. Tata cara memasukkan data dan mencetak hasil
 - e. Penjelasan detail tentang pesan-pesan kesalahan.
3. Pemakai **biasa**, merupakan orang di dalam organisasi yang memanfaatkan hasil pengolahan data untuk menjalankan tugasnya. Dokumentasi yang dibutuhkan: buku manual yang berisi penjelasan tentang fungsi program dan hasil pengolahan data serta cara membaca hasil tersebut

Penyajian Algoritma

Algoritma bisa disajikan dengan cara sebagai berikut:

- a. Program flowchart
- b. Pseudo code atau english structure

Contoh soal algoritma

Misalkan, anda memiliki teman yang tiba di bandara, dan teman anda perlu pergi dari bandara ke rumah anda.

Bagaimana ia bisa sampai dirumah anda ?

Berikut adalah empat algoritma berbeda yang dapat anda berikan kepada teman anda untuk sampai ke rumah anda !

1. The taxi algorithm
 - ☐ Pergi ke tempat taksi.
 - ☐ Naik taksi.
 - ☐ Berikan alamat saya pada pengemudi
2. The call me algorithm
 - ☐ Ketika pesawat Anda tiba, hubungi ponsel saya.
 - ☐ Temui saya di luar klaim bagasi.
3. The rent a car algoritm
 - ☐ Naik shuttle ke tempat rental mobil.
 - ☐ Menyewa mobil.
 - ☐ Ikuti petunjuk untuk sampai ke rumah saya.
4. The bus algorithm
 - ☐ Di luar klaim bagasi, naik bus nomor 70.
 - ☐ Transfer ke bus 14 di Main Street.
 - ☐ Turun di Elm street.

- ❑ Berjalanlah dua blok ke utara ke rumah saya.

Analisis jawaban

- Keempat algoritma ini mencapai tujuan yang persis sama, tetapi masing-masing algoritma melakukannya dengan cara yang sangat berbeda.
- Setiap algoritma juga memiliki biaya dan waktu perjalanan yang berbeda.
- Naik taksi, misalnya, mungkin adalah cara tercepat, tetapi juga yang paling mahal. Naik bus jelas lebih murah, tetapi jauh lebih lambat.
- Anda memilih algoritma berdasarkan keadaan.

3. Tugas Pendahuluan

Buatlah sebuah program sederhana yang dapat menampilkan teks, masukkan data dan menampilkan data pada C++

Algoritma

- Buat variable `programStudi` berisi data nama program studi "Sistem Informasi"
- cetak teks

C++

Nama program yaitu **cetakTeks.cpp**

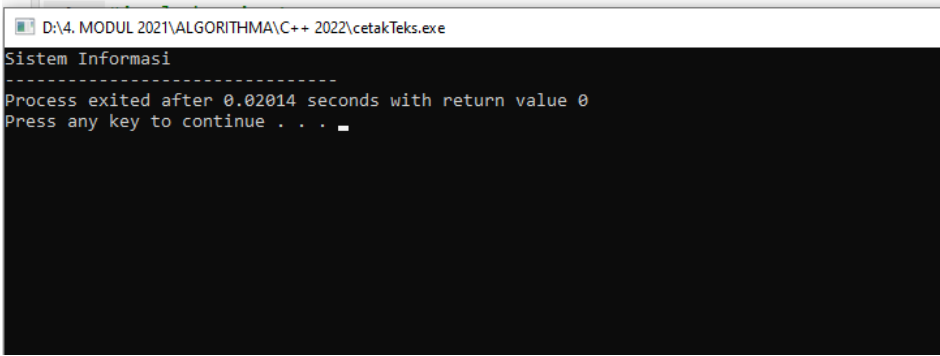
```
cetakTeks.cpp
1  #include <iostream>
2  using namespace std;
3
4  main() {
5      string programStudi ="Sistem Informasi";
6      cout<<programStudi;
7  }
```

Perhatikan coding dari program **cetakTeks.cpp** diatas! coding C++ diatas terdiri dari 7 baris, berikut penjelasan baris per baris, sbb

1. menyertakan library `iostream` pada program **cetakTeks.cpp**, tujuan agar perintah menggunakan keyword terkait input dan output bisa digunakan, terkait ke baris 6

2. menggunakan namespace standard
3. blank, baris kosong
4. menyatakan program utama, setiap program C++ mesti menggunakannya, tanda { menunjukkan awal blok program, ditutup pada baris ke 7
5. string adalah type data teks, sedangkan programStudi adalah variable diberi nilai data “Sistem Informasi”, assignment. Ditugakan berikan nilai “Sistem Informasi” ke variable programStudi
6. keyword **cout**, berfungsi untuk cetak di screen / layar nilai varibel programStudi, hasilnya tampak pada bagian output dibawah
7. }, tanda penutup blok program

Output



```

D:\4. MODUL 2021\ALGORITHMMA\C++ 2022\cetakTeks.exe
Sistem Informasi
-----
Process exited after 0.02014 seconds with return value 0
Press any key to continue . . .

```

Gambar 6 Output Teks: Sistem Informasi

Output program mencetak teks Sistem Informasi

4. Percobaan

Mahasiswa diminta melakukan percobaan soal-soal latihan dibawah ini, tulis *flowchart* dan program C++. Kemudian di praktekan

1. Menampilkan teks nama dan nobp mahasiswa
2. Mencetak teks STMIK Indonesia
3. Menjumlahkan 2 buah bilangan dan mencetak hasilnya
4. Hitunglah usia anda
5. Hitung keliling bangun persegi panjang

6. Hitung luas bangun segitiga

5. Laporan Praktikum

Kumpulkan hasil percobaan di atas, tambahkan dalam laporan *flowchart* dari setiap percobaan tsb !

Beri Penjelasan dan kesimpulan singkat !

MODUL 2 OPERATOR, TYPE DATA DAN VARIABLE

1. Tujuan

- Memahami berbagai macam operator, type data dan variabel dan cara penggunaanya

2. Dasar Teori

Operator

Operator atau pengenal diperlukan dalam menulis program komputer, operator berperan untuk mengekspresikan program terkait: aritmatika, relational dan logika

Operator Aritmatika

Komputer dapat melakukan kalkulasi matematika lebih cepat dari yang dilakukan manusia. Namun bagaimanapun computer tidak dapat melakukan perhitungan matematika tanpa menerima instruksi dari user. Dalam C++, dapat ditulis instruksi komputer untuk melakukan kalkulasi matematika seperti penambahan, pengurangan, perkalian, pembalian, perhitungan sisa bagi atau modulus dan operasi aritmatik yang lain. Agar C++ dapat melakukan kalkulasi aritmatika diperlukan penulisan code yang menggunakan operator aritmatika. Operator aritmatika sangat mirip dengan operator biasa, hanya saja sedikit bervariasi. Operator plus dan minus sama sedangkan operator perkalian menggunakan simbol `*` (bintang) dan pembagian hanya menggunakan simbol `/` dan operator sisa bagi menggunakan simbol `%`. Operator aritmatik ditampilkan pada table dibawah ini.

Asumsi variable A bernilai 10, variable B bernilai 20.

Operator Aritmatika diatas dipakai juga dalam PHP programming, namun beberapa simbol dirubah pemakaiannya tetapi tetap dalam maksud yang sama. Simbol yang dimaksud yaitu untuk modulus, perkalian dan pembagian, perhatikan contoh dibawah ini!

Tabel 2.1: Operator Aritmatika

No	Math	C++ Programming	Contoh
1.	+	+	$10 + 2 = 12$
2.	-	-	$10 - 2 = 8$
3.	/, ÷, :	/	$10 / 2 = 5$
4.	x, .	*	$10 * 2 = 20$
5.	Modulus (sisa bagi)	%	$10 \% 2 = 0$ $10 \% 4 = 2$ $4 \% 2 = 0$ $50 \% 20 = 10$

6.	Increment (Penambahan 1)	++	A=5 A++, maka A bernilai 6 b=2 b=b+10 dapat tulis b += 10, maka b=12
7.	Decrement (Pengurangan 1)	--	A=5 A--, maka A bernilai 5 b=2 b=b-10 dapat tulis b -= 10, maka b=-8

Operator Relational

Operator *Relational* adalah simbol yang sangat penting menyusun operator-operator matematika. Operator tersebut dimungkinkan untuk diprogram untuk membandingkan nilai data kemudian diputuskan untuk melakukan aksi, ketika program dieksekusi atau program dihentikan. Operator ini juga dikenal sebagai pembanding numeric. Secara normal operator digunakan untuk membandingkan dua nilai untuk mendapatkan apakah nilai bernilai sama, lebih besar atau bernilai lebih kecil satu dari yang lainnya. Hasil pembandingan akan mengembalikan hasil nilai *true* atau *false*. Operator tersebut ditampilkan pada table dibawah ini.

Tabel 2.2: Operator Relational

NO	SIMBOL OPERATOR	RELASI OPERASI	CONTOH	DESKRIPSI
1.	==	Sama dengan	A==B	Mengembalikan nilai true jika A sama dengan B, jika tidak maka akan mengembalikan false .
2.	!=	Tidak sama	A != B	Mengembalikan nilai true jika A tidak sama dengan B, jika tidak maka akan mengembalikan false .
3.	>	Lebih besar	A > B	Mengembalikan nilai true jika A lebih besar dari B, jika tidak maka akan mengembalikan false .
4.	<	Lebih kecil	A < B	Mengembalikan nilai true jika A lebih Kecil dari B, jika tidak maka akan mengembalikan false .
5.	>=	Lebih Besar atau sama	A >= B	Mengembalikan nilai true jika A lebih besar dari atau sama dengan B, jika tidak maka akan mengembalikan false .
6.	<=	Lebih Kecil atau sama	A <= B	Mengembalikan nilai true jika A lebih kecil atau sama dengan B, jika tidak maka akan mengembalikan false .

Operator Logika

Kadang-kadang diperlukan lebih dari satu perbandingan sebelum sebuah kondisi diputuskan untuk bisa melakukan aksi. Pada kasus ini, penggunaan pembanding *numeric* menggunakan satu operator tidak cukup, untuk kasus yang kompleks diperlukan operator tambahan dan memakai operator logika. Operator logika ditampilkan pada table dibawah.

Tabel 2.3: Operator Logika

NO	OPERATOR	KETERANGAN	CONTOH
1	&&	Kedua kondisi harus benar	If (A = B) && (P < Q)
2		Satu kondisi atau kedua kondisi harus benar	If (A > B) (P < Q)
3	!	Negative true	If not (A >= B)

Operator Logika dalam pemakaiannya sering dipasangkan dengan statement if untuk menulis ekspresi logika yang lebih kompleks.

Contoh-1.

Ali pergi kuliah jika telah makan dan telah mandi

Analisis Logika

- Ali telah makan logika 1, Ali tidak makan logika 0, begitu juga dengan
- Ali telah mandi logika 1, Ali tidak mandi logika 0
- Untuk pemakaian operator AND, kedua kondisi harus terpenuhi barulah ada aksi
- Untuk pemakaian operator OR, salah satu atau kedua kondisi harus terpenuhi barulah ada aksi

Perhatikan *truth table* dibawah ini menggunakan logika AND

Tabel 2.4 *Truth Table AND*

NO	Makan	Mandi		AND	Hasil
1.	1	1		1 And 1	1 ada aksi=pergi kuliah
2.	1	0		1 AND 0	0 tidak pergi kuliah
3.	0	1		0 AND 1	0 tidak pergi kuliah
4.	0	0		0 AND 0	0 tidak pergi kuliah

Hasil aksi yaitu Ali pergi kuliah, jika telah makan logika 1 dan telah mandi logika 1

Perhatikan truth table dibawah ini OR

Tabel 2.5 *truth table* OR

NO	Makan	Mandi	AND	Hasil
1.	1	1	1 OR 1	1 ada aksi pergi kuliah
2.	1	0	1 OR 0	1 ada aksi pergi kuliah
3.	0	1	0 OR 1	1 ada aksi pergi kuliah
4.	0	0	0 OR 0	0 tidak pergi kuliah

Hasil aksi yaitu Ali pergi kuliah, jika telah makan logika 1 atau telah mandi logika 1 atau salah satu atau kedua keduanya logika 1 terpenuhi

Type data

No	Type Data	Size	Deskripsi	Contoh
1.	int	4 bytes	Menyimpan bilangan bulat, tanpa desimal	int nilaite=90;
2.	float	4 bytes	Menyimpan bilangan pecahan, berisi satu atau lebih desimal. Cukup untuk menyimpan 7 digit desimal	float uas=87,77;
3.	double	8 bytes	Menyimpan bilangan pecahan, berisi satu atau lebih desimal. Cukup untuk menyimpan 15 digit desimal	double saldo=950,00;
4.	boolean	1 byte	Menyimpan nilai true atau false	Bool myBoolean= true;
5.	char	1 byte	Menyimpan satu karakter/huruf/angka, atau nilai ASCII	char nilaiHuruf='A';
6.	string	8 bytes	Menyimpan teks	String myTeks ="Sistem Informasi"
7.	short int	2 bytes	Menyimpan bilangan bulat, tanpa desimal	-255 sd 255
8.	long int	8 bytes	Menyimpan bilangan bulat, tanpa desimal	

Variabel

Variabel adalah wadah untuk menyimpan nilai data.

Di C++, ada berbagai jenis variabel (didefinisikan dengan keyword yang berbeda).

Penulisan nama variabel ikutilah dengan konsisten, artinya jika nama variabel diawali huruf besar maka pemakaian seterusnya harus mengikutinya dalam penulisan. C++ membedakan penulisan variabel Nama dengan variabel nama disebut *case sensitive*. Dimana variabel Nama diawali huruf besar, sedangkan nama semua huruf kecil, dalam pemakaiannya keduanya berbeda. Pastikan penulisan nama variabel mengikuti ketentuan berikut.

1. Awali variabel dengan huruf, tidak boleh angka
2. Jika var dua kata, tidak ada spasi diantaranya
3. Jika var dua kata, boleh diberi tanda sambung symbol under score (garis bawah). Tetap tidak ada spasi diantaranya
4. Nama variabel tidak boleh diawali angka
5. Nama varibel tidak boleh diikuti tanda khusus seperti: %, &, +, dll

NO.	SYAH	TIDAK SYAH
1.	Nama7	7Nama
2.	alamatRumah	Alamat-Rumah
3.	nilaiTes	Nilai%tes
4.	Usia_ku	Usia _ku
5.	sisiAlas	sisi+alas
6.	Nilai	nilai
	NILAI	nilai

Contoh penulisan variabel

```
int myNum = 5;           // Integer (whole number)
float myFloatNum = 5.99; // Floating point number
double myDoubleNum = 9.98; // Floating point number
char myLetter = 'D';     // Character
bool myBoolean = true;   // Boolean
string myText = "Hello"; // String
```

Contoh Program C++ Type Data


Program pengujian untuk berbagai jenis type data dan pendeklarasian var, sizeofdata.cpp sbb

```

functionOR.cpp variable.cpp functifelse.cpp sizeofdata.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      cout << "Size of int : " << sizeof(int) << endl;
6      cout << "Size of float : " << sizeof(float) << endl;
7      cout << "Size of double : " << sizeof(double) << endl;
8      cout << "Size of booblean : " << sizeof(bool) << endl;
9
10     cout << "Size of char : " << sizeof(char) << endl;
11     cout << "Size of string : " << sizeof(string) << endl;
12
13     cout << "Size of short int : " << sizeof(short int) << endl;
14     cout << "Size of long int : " << sizeof(long int) << endl;
15     cout << "Size of wchar_t : " << sizeof(wchar_t) << endl;
16
17     return 0;
18 }

```

Output



```

D:\4. MODUL 2021\ALGORITHMMA\C++ 2022\sizeofdata.exe
Size of int : 4
Size of float : 4
Size of double : 8
Size of booblean : 1
Size of char : 1
Size of string : 8
Size of short int : 2
Size of long int : 4
Size of wchar_t : 2
-----
Process exited after 0.496 seconds with return value 0
Press any key to continue . . .

```

Gambar 7 Output Program Menampilkan Ukuran Penggunaan Memory Untuk Data

3. Tugas Pendahuluan

Contoh logika AND

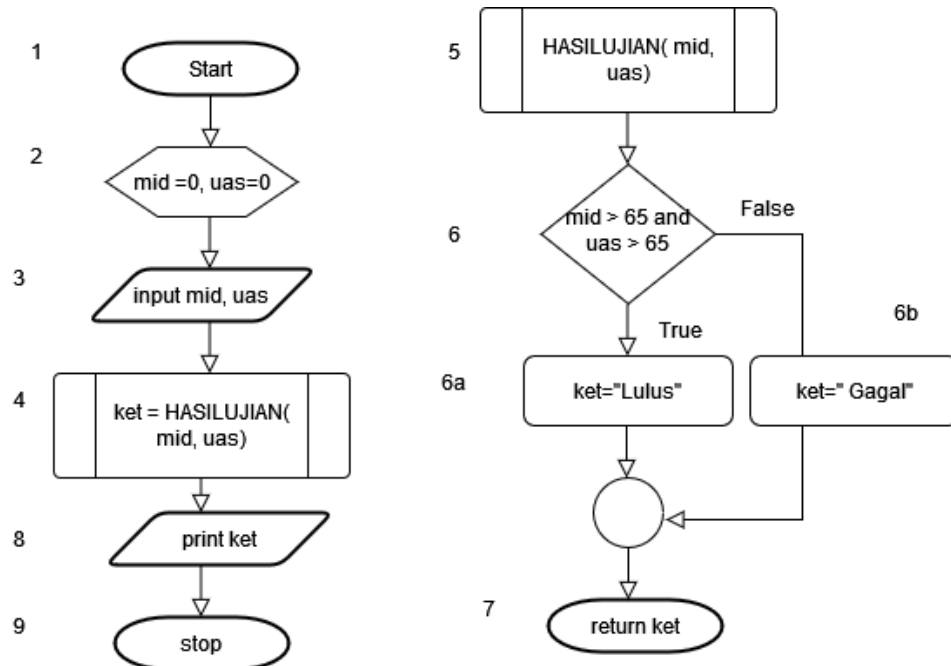
Seorang mahasiswa dinyatakan Lulus jika masing-masing nilai hasil ujian MID dan UAS melebihi 65. Jika tidak dinyatakan GAGAL

Analisis.

Soal diatas menggunakan operator AND maknanya adalah jika kedua kondisi masing-masing nilai MID dan nilai UAS mesti lebih dari 65, maka mahasiswa dinyatakan LULUS, sebaliknya dinyatakan GAGAL

Program Flowchart AND

Flowchart hasil ujian menggunakan operator AND, dapat dilihat pada gambar 8 terdiri dari 9 urutan nomor, penjelasan disampaikan sesuai nomor urut sbb



Gambar 8 Flowchart Hasil Ujian Operator AND

Algoritma AND

1. Start → mulai
2. Preparation → variable mid nilai awal 0, uas nilai awal 0
3. input mid, uas → input mid, uas. Missal mid = 70, uas = 100
4. ket = HASILUJIAN(mid, uas) → panggil function HASILUJIAN, kirim parameter mid dan uas sebagai data input
5. HASILUJIAN(mid, uas) → kedua parameter dipakai pada function ini
6. Jika mid > 65 and uas > 65
 - a. ket = "Lulus" → berikan data Lulus ke var ket. Disebabkan nilai mid = 70 dan uas = 100 maka ket = Lulus
 - b. Jika Tidak memenuhi ket = "Gagal " → berikan data Gagal ke var ket

7. return ket → kembalikan data yang dikandung var ket ke baris 4
8. print ket → cetak data yang dikandung var ket yaitu Lulus
9. stop

Program C++

Program C++ diberi nama **functionand.cpp** terdiri dari 30 baris. Setiap baris atau *statement* diberi penjelasan disebelah kanan *code*, baca dan cermati setiap baris agar paham, dan ditulis code C++ dibawah ini.

```

Untitled1  functionand.cpp
1  #include <iostream> // memberitahu compiler bahwa program menggunakan library iostream
2  using namespace std; // berisi ttg perintah mencetak text atau data ke layar / screen
3
4  string HASILUJIAN(int mid, int uas){ // type data output function ini adalah string
5      // nama function HASILUJIAN, nama parameter nilaites type data int
6      string ket=""; // vari ket dgn type data string=teks
7      if(mid > 65 and uas > 65 ){ // jika nila lebih 65, ada aksi
8          ket="LULUS"; // aksi berikan data LULUS ke var ket
9      } else {
10         ket ="Gagal";
11     }
12     return ket; // kembalikan var ket ke baris 26, pemanggil function
13 }
14
15 main () { // main= program utama harus ada setiap program C++/C
16     int mid =0, uas=0; // var nilaites berisi data 0, type data int= bilangan bulat
17     string keterangan=""; // var keterangan berisi data blank, type data string=teks
18
19     cout<<"Input Nilai MID ="; // cout=perintah mencetak teks di screen
20     cin>>mid; //cin = menerima input data dari keyboard disimpan di var mid
21
22     cout<<"Input Nilai UAS ="; // cout=perintah mencetak teks di screen
23     cin>>uas; //cin = menerima input data dari keyboard disimpan di var uas
24
25     //cout <<endl; // endl=perintah pindah baris
26     keterangan = HASILUJIAN( mid, uas) ; // di panggil function HASILUJIAN
27     // kirim parameter nilaites
28     cout<<"Keterangan = " <<keterangan <<endl ; // cetak teks Keterangan
29     // cetak nilai yang dikandung var keterangan
30 }

```

Output

Percobaan program functionand.exe, di input nilai mid=70 dan uas = 100, tampak output keterangan = LULUS

```

D:\4. MODUL 2021\ALGORITMA\C++ 2022\functionand.exe
Input Nilai MID =70
Input Nilai UAS =100
Keterangan = LULUS

-----
Process exited after 4.13 seconds with return value 0
Press any key to continue . . .

```

Gambar 9 Output Program Functionand, mid=70 dan uas=100 Hasil LULUS

Contoh logika OR

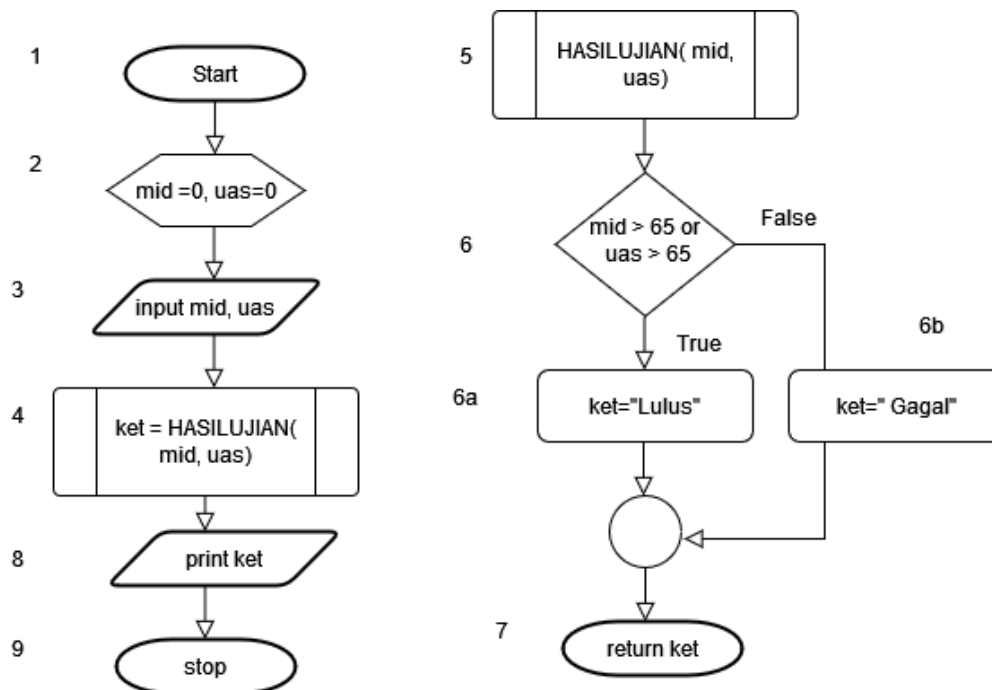
Seorang mahasiswa dinyatakan Lulus jika masing-masing nilai MID atau UAS melebihi 65.

Analisis.

Soal diatas menggunakan operator OR maknanya adalah jika salah satu kondisi atau keduanya terpenuhi, maka mahasiswa dinyatakan LULUS

Program Flowchart OR

Flowchart hasil ujian menggunakan operator OR, dapat dilihat pada gambar 10 terdiri dari 9 urutan nomor, penjelasan disampaikan sesuai nomor urut sbb



Gambar 10 Flowchart Hasil Ujian Operator OR

Algoritma

1. Start → mulai
2. Preparation → variable mid nilai awal 0, uas nilai awal 0
3. input mid, uas → input mid, uas. Missal mid = 70, uas = 100
4. ket = HASILUJIAN(mid, uas) → panggil function HASILUJIAN, kirim parameter mid dan uas sebagai data input
5. HASILUJIAN(mid, uas) → kedua parameter dipakai pada function ini
6. Jika mid > 65 and uas > 65
 - a. ket = “Lulus” → berikan data Lulus ke var ket. Disebabkan nilai mid = 70 atau uas = 100, maka ket = Lulus
 - b. Jika Tidak memenuhi ket = “Gagal “ → berikan data Gagal ke var ket
7. return ket → kembalikan data yang dikandung var ket ke baris 4
8. print ket → cetak data yang dikandung var ket yaitu Lulus
9. stop

Program C++

Program C++ diberi nama **functionOR.cpp** terdiri dari 30 baris. Setiap baris atau *statement* diberi penjelasan disebelah kanan *code*, baca dan cermati setiap baris agar paham, dan ditulis code C++ dibawah ini.

Beda program functionand.cpp dengan functionor.cpp terletak pada pengujian kondisi, dimana pada functionand.cpp harus kedua kondisi terpenuhi barulah ada aksi yaitu nilai mid dan nilai uas mesti lebih dari 65. Sedangkan pada program functionOR.cpp jika salah satu atau kedua nilai melebihi 65 maka ada aksi. Aksi mahasiswa Lulus

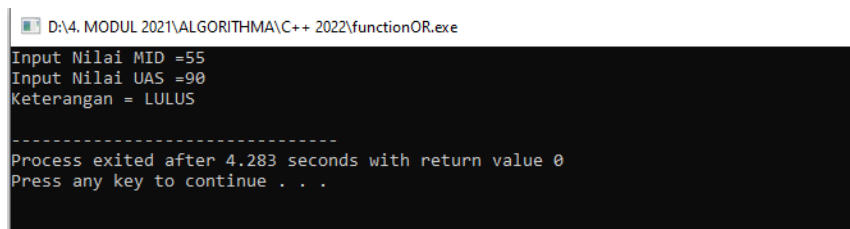
```

Untitled1 [*] functionOR.cpp
1  #include <iostream> // memberitahu compiler bahwa program menggunakan library iostream
2  using namespace std; // berisi ttg perintah mencetak text atau data ke layar / screen
3
4  string HASILUJIAN(int mid, int uas){ // type data output function ini adalah string
5      // nama function HASILUJIAN, nama parameter nilaites type data int
6      string ket=""; // vari ket dgn type data string=teks
7      if(mid > 65 or uas > 65 ){ // jika salah satu nilai lebih 65, ada aksi
8          ket="LULUS"; // aksi berikan data LULUS ke var ket
9      } else {
10         ket ="Gagal";
11     }
12     return ket; // kembalikan var ket ke baris 26, pemanggil function
13 }
14
15 main () { // main= program utama harus ada setiap program C++/C
16     int mid =0, uas=0; // var nilaites berisi data 0, type data int= bilangan bulat
17     string keterangan=""; // var keterangan berisi data blank, type data string=teks
18
19     cout<<"Input Nilai MID ="; // cout=perintah mencetak teks di screen
20     cin>>mid; //cin = menerima input data dari keyboard disimpan di var mid
21
22     cout<<"Input Nilai UAS ="; // cout=perintah mencetak teks di screen
23     cin>>uas; //cin = menerima input data dari keyboard disimpan di var uas
24
25     //cout <<endl; // endl=perintah pindah baris
26     keterangan = HASILUJIAN( mid, uas) ; // di panggil function HASILUJIAN
27     // kirim parameter nilaites
28     cout<<"Keterangan = " <<keterangan <<endl ; // cetak teks Keterangan
29     // cetak nilai yang dikandung var keterangan
30 }

```

Output

Percobaan program functionOR.exe, di input nilai mid=55 dan uas = 90, tampak output keterangan = LULUS, karena salah satu nilai ujian UAS = 90 terpenuhi, lihat gambar 11



```

D:\V4. MODUL 2021\ALGORITMA\C++ 2022\functionOR.exe
Input Nilai MID =55
Input Nilai UAS =90
Keterangan = LULUS

Process exited after 4.283 seconds with return value 0
Press any key to continue . . .

```

Gambar 11 Output Program FunctionOR, mid=55 dan uas=90 Hasil LULUS

4. Percobaan

Mahasiswa diminta melakukan percobaan soal-soal latihan dibawah ini, tulis *flowchart* dan program C++. Kemudian di praktekkan

1. Hasil test matakuliah Pemrograman Visual I berupa nilai 0 s/d 100. Nilai diluar itu ditolak tampilkan warning “Nilai diluar Range”. Dari hasil test itu cetaklah predikat kelulusan seperti ketentuan dibawah ini :
 - a. Jika Nilai 0 s/d 50 Predikat “Fail“
 - b. Jika Nilai 51 s/d 65 Predikat “Good“
 - c. Jika Nilai 66 s/d 85 Predikat “Very Good“
 - d. Jika Nilai 86 s/d 100 Predikat “Excellent”
2. Buat program aplikasi untuk menampilkan Kategori usia. Tentukan **usia** dan **kategori usia** anda berdasarkan ketentuan:
 - a. Usia 0 s.d 5 kategori usia “Balita”
 - b. Usia 6 s.d 10 kategori usia “Anak-anak”
 - c. Usia 11 s.d 17 kategori usia “Remaja”
 - d. Usia 18 keatas kategori usia “Dewasa”

Usia diluar ketentuan diatas tampilkan pesan “Data salah, silahkan perbaiki...!”

5. Laporan Praktikum

Kumpulkan hasil percobaan di atas, tambahkan dalam laporan *flowchart* dan program C++ dari setiap percobaan tsb !

Beri Penjelasan dan kesimpulan singkat !

MODUL 3 PROGRAM FLOWCHART

1. Tujuan

- Mengenal simbol Program *Flowchart* dan cara menggunakannya dengan tepat
- Menyajikan algoritma menggunakan program *flowchart*
- Menterjemahkan program *flowchart* ke program C++

2. Dasar Teori

Flowchart adalah bagan alur suatu rancangan program menggunakan metoda grafik dengan aturannya yang menarik dan mudah untuk dipelajari. Program *Flowchart* adalah simbol-simbol standard yang disusun secara logis untuk mengekspresikan logika penyelesaian masalah *programming*. Program *flowchart* bertujuan untuk melatih programmer pemula (***beginner programmer***) berfikir logis, ***sequence*** dengan cara membangun diagram alur logika sebelum menulisnya ke bahasa pemrograman. Logika program disusun mulai dari atas mengalir kebawah dan dari kiri ke kanan, kemungkinan logika yang disusun mengandung percabangan, perulangan dan mungkin memanggil sub program atau *function*.

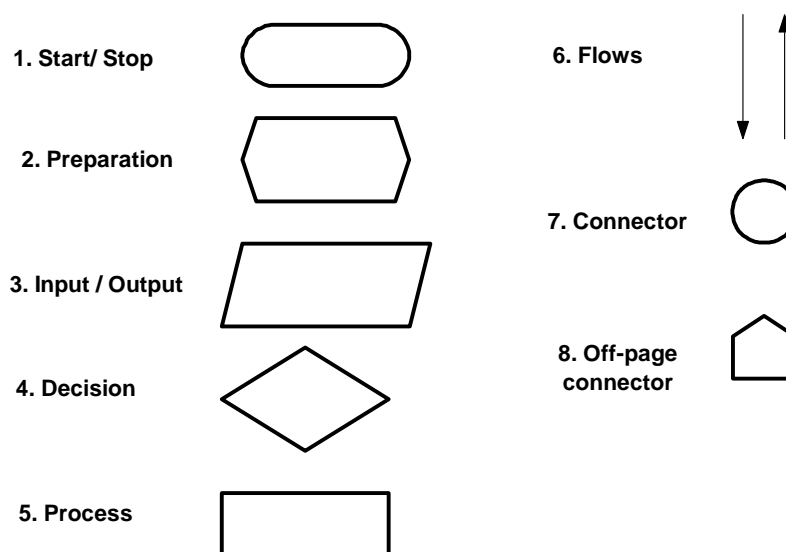
Jadi penyusunan suatu program *flowchart* bukanlah tujuan akhir melainkan suatu metoda bagaimana mendekati masalah *programming language* sebelum menulisnya ke bahasa pemrograman, seperti C++, Java, PHP, dll. Bahasa pemrograman yang ditulis kemudian di terjemahkan ke oleh compiler ke bahasa mesin (executable atau exe), di *run*, barulah diperoleh hasilnya tampak di *screen*. Saat itu programmer mengamati dan berfikir apakah hasil yang tampak sama seperti yang diinginkan ? jika ya programmer telah berhasil menyelesaikan satu masalah atau sebaliknya melakukan perbaikan.

Program *flowchart* disusun sebelum menulis program aplikasi terutama untuk ***beginner programmer*** dan untuk penyelesaian masalah *programming* yang kompleks. Seorang *programmer* yang terlatih bisa menyusun logika program *flowchart* dengan cepat dan tepat. Suatu program *flowchart* yang tepat apabila dapat disusun dengan alur logika yang paling pendek, sederhana, bukan berbelit-belit dan menampilkan *output* yang diinginkan. Untuk mencapai maksud itu seseorang disarankan banyak melakukan latihan, mengerjakan soal-soal *programming*.

Simbol Dasar Program Flowchart

Simbol dasar program flowchart adalah symbol standard yang dikenal dalam ilmu *computer*. Kenali dengan baik sebelum *flowchart* sebelum digunakan untuk menyelesaikan soal-soal program. Berikut ini symbol dasar program flowchart lihat gambar 12!

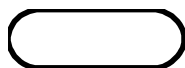
SIMBOL DASAR PROGRAM FLOWCHART



Gambar 12 Simbol Dasara Program Flowchart

Simbol dasar flowchart gambar 12 diatas digunakan sebagai simbol dasar untuk menyusun logika untuk penyelesaian masalah *programming*, penggunaannya susun sesuai kebutuhan, dibawah ini penjelasan masing-masing simbol sebagai berikut

1. Start / Stop



Simbol *start/stop* digunakan untuk memulai dan mengakhiri program. Simbol ini harus ada dalam setiap program *flowchart*, menandakan awal dan akhir dari logika program yang ditulis. Simbol ini sudah baku dan dikenal luas dalam dunia *computer science*. Simbol ini juga digunakan pada *function* untuk mengembalikan kontrol program ke program pemanggil atau caller function.

2. Preparation



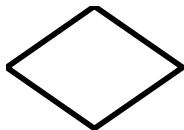
Simbol *preparation* atau persiapan digunakan untuk meng-inisialisasi (menentukan variabel dan memberi nilai awal) variabel sebelum digunakan dalam program. Tujuannya agar program dapat mengenal pasti variabel yang digunakan dan nilai yang dikandungnya. *Programmer* menyiapkan variabel yang diperlukan dengan pasti sebagai persiapan menyusun logika program. Jangan mendeklarasikan variabel yang tidak memiliki peran karena menulis variabel berarti menambah beban *memory* komputer.

3. Input / output



Simbol Input/Output digunakan untuk menetapkan variabel input dan output apa data yang akan diinput dan apa data dicetak dalam program. Apa variabel yang menjadi input dan apa variabel yang menjadi output. Variabel input menetapkan variabel yang diperlukan agar suatu proses dapat terlaksana.

4. Decision



Simbol *decision* digunakan untuk mengekspresikan logika percabangan, suatu kondisi yang memerlukan satu pilihan. Pilihan yang terpenuhi diekspresikan dengan simbol T atau *True* dan sebaliknya F atau *False* yang diikuti oleh simbol flow (alur logika menggunakan simbol garis panah).

Simbol *decision* diperlukan untuk mengekspresikan penentuan pilihan dari kondisi yang ada agar ada satu keputusan tepat yang diambil. Keputusan yang diambil mungkin *True* atau mungkin *False*. Tindak lanjut dari keduanya menentukan ekspresi logika berikutnya, jadi buatlah dengan cermat agar ekspresi logika yang digambarkan mewakili logika yang benar sebagai jawaban dari masalah.

Pastikan penggambaran logika *decision* ini untuk alur logika *True* arah panah menuju ke arah bawah dan alur logika *False* arah panah menuju ke kanan-bawah.

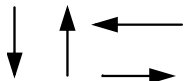
5. Process



Simbol *process* digunakan untuk mengekspresikan logika *assignment*, penugasan, atau ekspresi proses seperti notasi matematika

Pemakaian simbol *process* jumlah ekspresi matematika atau rumus yang dibuat tidak ditentukan, buatlah sesuai keperluan logika yang dibangun.

6. Flow



Simbol *flow*, alur atau arah panah, digunakan untuk menunjukkan alur logika program: ke bawah, ke atas, ke kiri atau ke kanan. *Programmer* dapat menunjukkan alur logika program dengan menggunakan simbol alur logika pada secara tepat.

Secara umum dapat dipahami proses kerja komputer dari atas ke bawah dan dari kiri ke kanan, dalam proses yang terjadi mungkin ditemukan logika percabangan dan logika perulangan semua itu ditunjukkan dengan menggunakan arah alur logika menggunakan *flow*. Pastikan memberi tanda panah pada suatu program flowchart yang ditulis agar dapat dikenali secara pasti alur logika yang dibangun.

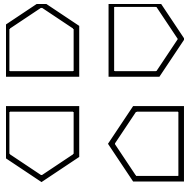
7. Connector



Simbol *connector* digunakan untuk mempertemukan dua arus logika sehingga kembali menyatu. Pemakaian simbol ini berlawanan dengan simbol *decision* yaitu membuat logika percabangan. Pemakaian *connector* ini juga digunakan sebagai tanda sambung flowchart pada halaman yang sama. Flowchart yang kompleks sering digambarkan lebih dari satu bagian, untuk itu diperlukan penyambungan antara agar alur logika dari bagian-bagian itu tetap satuan kesatuan secara keseluruhan.

Jumlah pemakaian *chart*, simbol, *connector* dalam satu gambar flowchart tidak ada ketentuan, jadi silahkan berekspresi.

8. Off-page connector



Simbol *Off-page connector* digunakan untuk menyambung logika ke halaman lain. Penggambaran suatu *flowchart* yang kompleks ada kemungkinan ditulis lebih dari satu bagian *flowchart* dan mungkin disambungkan juga ke halaman berikutnya. Simbol *flow* ini digunakan untuk menyambung alur logika ke halaman berikutnya. Untuk bagian *flowchart* yang ditulis pada halaman yang sama gunakan *chart connector*.

Untuk bisa mengenali tanda sambung dari satu halaman ke halaman lain, tuliskan teks: angka, huruf atau kata pada simbol ini agar jelas alur logika yang tuju menandakan sebagai sambungan dari halaman sebelumnya.

Simbol Lanjutan Program Flowchart

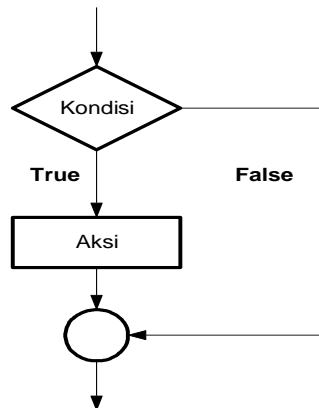
Flowchart IF

Simbol *decision structure*. Simbol ini disusun dari kombinasi simbol Decision dan Flow. Digunakan untuk mengekspresikan **If structure** untuk mengekspresikan logika percabangan, suatu kondisi memerlukan pilihan satu aksi yang tepat yang diseleksi berdasarkan kriteria yang ditentukan dalam soal. **Jika kondisi benar Kerjakan Aksi, jika kondisi salah abaikan**. Tuliskan tanda alur logika yang terpenuhi dengan *True* untuk menegaskan alur logika yang dilalui.

Perhatikan diujung alur logika *True* maupun *False*, kedua alur logika akan bertemu, untuk mempertemukan kedua alur logika tersebut gunakan simbol *connector*, lihat algoritma dibawah ini bagaimana membaca *flowchart* pada gambar 13.

- Jika kondisi *True* (benar) atau terpenuhi
 - Maka ada Aksi
- Jika kondisi *False* (salah) atau tidak terpenuhi
 - Maka tidak ada Aksi

- Setelah melakukan proses, kemudian kedua alur logika bertemu ditandai dengan simbol *connector*
- Lakukan *proses selanjutnya*



Gambar 13 IF-Logika Percabangan

Syntax Statement IF

```

if ( kondisi ) {
    // aksi-1
    ..
}
  
```

Contoh : Hasil nilai tes matakuliah Algoritma dinyatakan Lulus jika nilai melebihi 65

Analisis

Perhatikan dengan cermat soal diatas sebelum menulis flowchart dan program C++. Ikuti proses penyelesaian soal dengan membuat pertanyaan sbb:

1. Apa output yang diinginkan ? Pernyataan Lulus
2. Apa input ? nilai Algoritma
3. Bagaimana proses

```

If (nilaiAlgoritma > 65 ) {
    Keterangan ="Lulus"
}
  
```

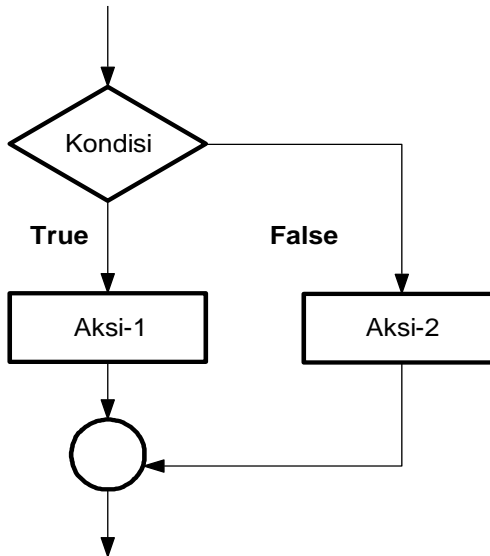
Flowchart IF..ELSE

Simbol decision structure digunakan untuk mengekspresikan terjadinya Aksi untuk kedua kondisi yang terpenuhi atau tidak terpenuhi. Simbol flowchart **if .. else structure**, digunakan untuk mengekspresikan logika percabangan. Suatu kondisi yang memerlukan aksi yang tepat dari hasil seleksi berdasarkan kriteria soal yang ada, **jika kondisi benar kerjakan Aksi-1, jika kondisi salah kerjakan Aksi-2**. Perhatikan setelah dilaksanakan percabangan logika, kedua alur logika bertemu lagi, gunakan simbol *connector* dan terus munculkan alur logika berikutnya hingga logika *stop*, lihat gambar 14. *Flowchart* ini dapat menggambarkan logika percabangan dan memungkinkan dibangun lagi logika percabangan dalam logika percabangan atau disebut nested if. Nested if ini digunakan untuk menyusun logika percabangan dari masalah programming yang kompleks.

Jumlah nested if yang bisa disusun pada sebuah program *flowchart* tidak ada batasan tertentu, buatlah selama diperlukan.

Contoh Soal

Contoh: Mahasiswa peserta tes Algoritma dinyatakan LULUS jika nilai melebihi 65, sebaliknya dinyatakan GAGAL.



Gambar 14 Structure If else atau Nested If

Contoh, buat program flowchart untuk menentukan pernyataan LULUS atau GAGAL dari seorang peserta tes Algoritma. Seorang mahasiswa dinyatakan LULUS jika nilai melebihi 65 atau sebaliknya dinyatakan GAGAL.

Seorang mahasiswa yang memiliki nilai lebih dari 65 maka Aksi yang dilakukan cetak pernyataan “LULUS “ jika tidak memenuhi Aksi yang lakukan cetak pernyataan “GAGAL“

3. Tugas Pendahuluan

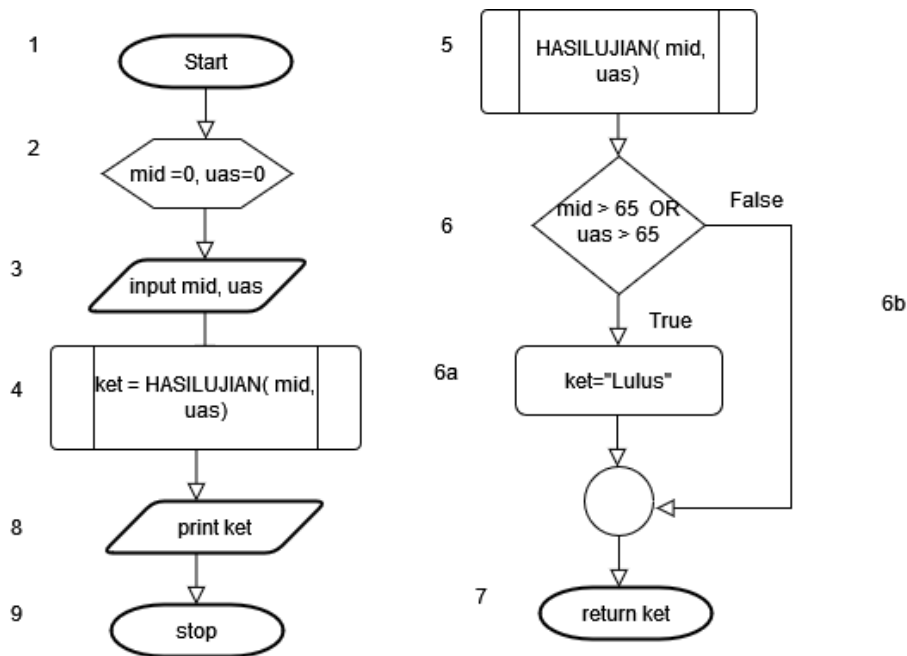
Contoh-1 IF

Contoh-1, buat program *flowchart* untuk menentukan keterangan LULUS dari mahasiswa peserta ujian Algoritma. Seorang mahasiswa dinyatakan LULUS jika nilai MID atau UAS melebihi 65.

Jawaban program flowchart lihat pada gambar 15. Terdapat 11 tahapan untuk menyelesaikan soal diatas menggunakan function. Perhatikan Algoritma, ditulis sesuai nomor urutan pengerjaan sbb:

Program Flowchart IF- Ujian Algoritma

Flowchart dibawah ini dibuat *online* di url <https://app.diagrams.net/>



Gambar 15 Flowhart Ujian Algoritma

Gambar 15 menjelaskan alur logika pengujian hasil ujian Algoritma.

Setelah selesai menulis program flowchart, kegiatan berikutnya adalah coding, menulis program C++, lihat dihalaman berikut! Program C++ ini diberi nama **functhasiltes.cpp** terdiri dari 24 baris. Ada penjelasan setiap *statement*, perintah dalam program functhasiltes.cpp

Algoritma

1. Start, mulai
2. Preparation, nilaimid=0 → deklarasikan var nilaites diberi 0, nilaiuas=0
3. Input nilaimid, nilaiuas → misal nilaimid = 100, nilaiuas=66 boleh angka lain
4. keterangan=HASILUJIAN(nilaimid, nilaiuas) → Panggil function HASILUJIAN, kirim parameter nilaimid =100, nilaiuas=66, hasilnya simpan pada variable **keterangan**
5. HASILUJIAN(nilaimid, nilaiuas) → Pendefenisian function, menerima parameter nilaites =100
6. nilaimid > 65 or nilaiuas > 65 → jika nilaites melebihi 65

- a. Maka ket=" LULUS " → berikan data **LULUS** ke var ket
- Else, sebaliknya, diabaikan
- Connector mempertemukan kedua alur logika true atau false
7. Return ket → nilai hasil proses yang disimpan pad var ket di dikembalikan ke tahap-4 dan disimpan pada variable **ket**
8. print **ket** → cetak nilai keterangan, output LULUS
9. stop, henti

Pada percobaan berikutnya, mahasiswa bisa input nilaites dengan angka berbeda missal 61, 80, 99, dst. Lalu perhatikan output yang ditampilkan program setiap di eksekusi.

Program C++ Ujian Algoritma

Ikuti tahap demi tahap statement yang ada, baca dengan cermat, pahami ! selanjutnya tulis pada teks editor Dev C++ 5.11. program C++ ini diberi nama **functhasiltes.cpp**

```

1 #include <iostream> // memberitahu compiler bahwa program menggunakan library iostream
2 using namespace std; // berisi ttg perintah mencetak text atau data ke layar / screen
3
4 string HASILUJIAN(int mid, int uas ){ // type data output function ini adalah string
5     // nama function HASILUJIAN, nama parameter nilaites type data int
6     string ket=""; // vari ket dgn type data string=teks
7     if(mid > 65 || uas > 65 ){// jika salah satu atau kedua nilai mid atau uas lebih 65 memenuhi
8         ket="LULUS"; // aksi berikan data LULUS ke var ket
9     } // jika tidak memenuhi proses di abaikan
10    return ket; // logika program balik kebaris 21, membawak data ket
11 }
12 main () { // main= program utama harus ada setiap program C++/C
13     int mid=0, uas=0; // var mid, uas berisi data 0, type data int= bilangan bulat
14     string ket =""; // var ket berisi data blank, type data string=teks
15
16     cout<<"Input Nilai MID ="; // cout=perintah mencetak teks Input Nilai MID
17     cin>>mid; //cin = var mid menerima input data dari keyboard
18     cout<<"Input Nilai UAS ="; // cout=perintah mencetak teks Input Nilai UAS
19     cin>>uas; //cin = var uas menerima input data dari keyboard
20
21     ket = HASILUJIAN( mid, uas) ; // panggil function HASILUJIAN baris 4
22     // kirim parameter mid, uas
23     cout<<"Keterangan = " <<ket <<endl ; // cetak teks 'Keterangan ='
24     // cetak data yang dikandung var ket
25 }

```

Output Program Ujian Algoritma

Percobaan ini, program functhasiltes, di input mid=90, uas= 100, tampak output keterangan = LULUS lihat gambar 16. Kedua nilai mid dan nilai uas memenuhi melebihi 65

```

D:\4. MODUL 2021\ALGORITMA\C++ 2022\functhasiltes.exe
Input Nilai MID =90
Input Nilai UAS =100
Keterangan = LULUS

-----
Process exited after 4.753 seconds with return value 0
Press any key to continue . . .

```

Gambar 16 Output FunctionHasilTes, nilai uas =100

Percobaan berikutnya, nilai mid diinput 61 dan nilai uas diinput 100, lihat gambar 17, tampak output LULUS, karena salah satu nilai uas memenuhi. Sebaliknya keterangan akan berisi *blank* jika kedua nilai mid (ujian tengah semester) dan nilai uas (ujian akhir semester) diinput dibawah 65. Coba input nilai masing-masing nilai mid adalah 60 dan nilai uas adalah 64!

Angka 5,193 detik menunjukkan waktu yang diperlukan compiler Dev C++ 5.11 untuk menterjemahkan bahasa program C++ (functhasiltes.cpp) ke bahasa mesin (functhasiltes.exe). Jadi, functhasiltes.exe adalah hasil kompilasi yang bisa langsung di eksekusi (di run), sedangkan functhasiltes.cpp adalah *source code* yang ditulis *programmer*.

```

D:\4. MODUL 2021\ALGORITMA\C++ 2022\functhasiltes.exe
Input Nilai MID =61
Input Nilai UAS =100
Keterangan = LULUS

-----
Process exited after 5.193 seconds with return value 0
Press any key to continue . . .

```

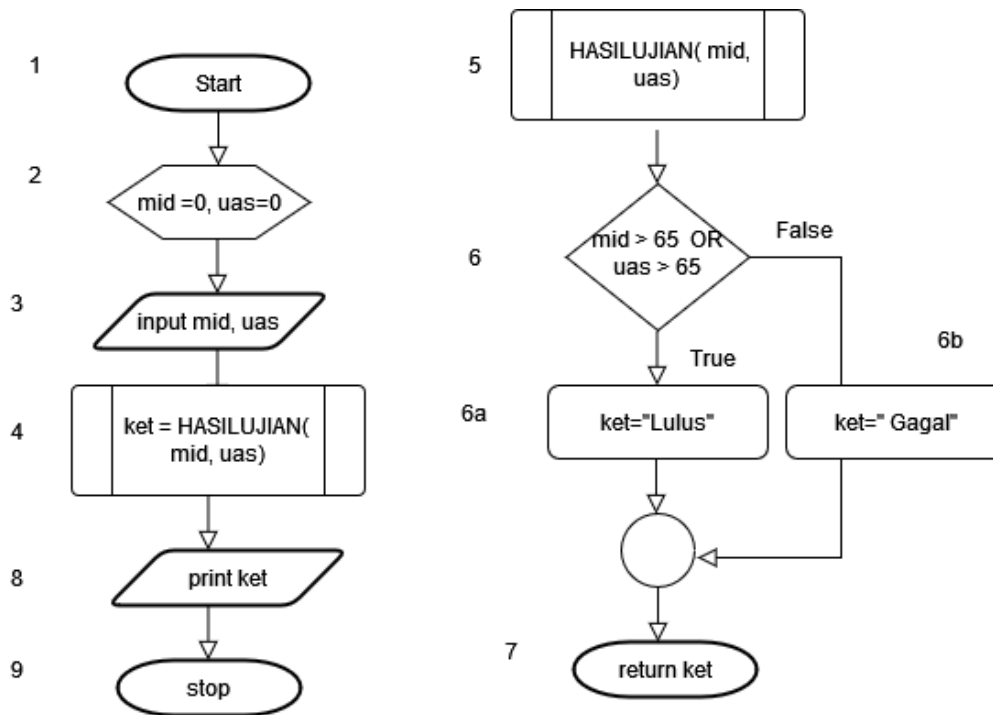
Gambar 17 Output FunctionHasilTes, nilai mid=61

Contoh-2 IF..ELSE

Contoh-1, buat program *flowchart* untuk menentukan keterangan LULUS dari mahasiswa peserta ujian Algoritma. Seorang mahasiswa dinyatakan LULUS jika nilai melebihi 65 sebaliknya dinyatakan GAGAL

Program Flowchart IF..ELSE - Ujian Algoritma

Gambar 18 menggambarkan logika yang digambarkan menggunakan chart disebut *flowchart*. Flowchart ini menseleksi sebuah kondisi dan mengakibatkan ada aksi jika Kondisi Terpenuhi dan ada Aksi jika kondisi tidak terpenuhi. If dibaca jika sedangkan else dibaca jika tidak.



Gambar 18 IF ..ELSE Hasil Ujian

Algoritma

1. Start, mulai
2. Preparation, nilaites=0 → deklarasikan var nilaites diberi 0
3. Input nilaites → misal nilaites = 100, boleh angka lain
4. keterangan=HASILUJIAN(nilaites) → Panggil function HASILUJIAN, kirim parameter nilaites =100, hasilnya simpan pada variable **keterangan**
5. HASILUJIAN(nilaites) → Pendefenisian function, menerima parameter nilaites =100
6. nilaites > 65 → jika nilaites melebihi 65
 - a. Maka ket=" LULUS " → berikan data **LULUS** ke var ket
 - b. Maka ket ="GAGAL" → berikan data **GAGAL** ke var ket

Connector mempertemukan kedua alur logika **true** atau **false**

7. Return ket → nilai hasil proses yang disimpan pad avar ket di kembalikan ke tahap-4 dan disimpan pada variable **keterangan**
8. print **keterangan** → cetak nilai keterangan, output LULUS
9. stop, henti

Pada percobaan berikutnya, mahasiswa bisa input nilaites dengan angka berbeda missal 61, 80, 99, dst. Lalu perhatikan output yang ditampilkan program setiap di eksekusi.

Program C++ IF..ELSE

Nama program ini disebut **functifelse.cpp**

```

functhasiltesLULUS.cpp  functifelse.cpp
1  #include <iostream> // memberitahu compiler bahwa program menggunakan library iostream
2  using namespace std; // berisi ttg perintah mencetak text atau data ke layar / screen
3
4  string HASILUJIAN(int mid, int uas ){ // type data output function ini adalah string
5      // nama function HASILUJIAN, nama parameter nilaites type data int
6      string ket=""; // vari ket dgn type data string=teks
7      if(mid > 65 || uas > 65 ){// jika salah satu atau kedua nilai mid atau uas lebih 65 memenuhi
8          ket="LULUS"; // aksi berikan data LULUS ke var ket
9      } else { // jika tidak memenuhi
10         ket ="GAGAL"; // berikan data GAGAL ke var ket
11     }
12     return ket; // logika program balik kebaris 23, membawa data ket
13 }
14 main () { // main= program utama harus ada |setiap program C++/C
15     int mid=0, uas=0; // var mid, uas berisi data 0, type data int= bilangan bulat
16     string ket =""; // var ket berisi data blank, type data string=teks
17
18     cout<<"Input Nilai MID ="; // cout=perintah mencetak teks Input Nilai MID
19     cin>>mid; //cin = var mid menerima input data dari keyboard
20     cout<<"Input Nilai UAS ="; // cout=perintah mencetak teks Input Nilai UAS
21
22     cin>>uas; //cin = var uas menerima input data dari keyboard
23
24     ket = HASILUJIAN( mid, uas) ; // panggil function HASILUJIAN baris 4
25     // kirim parameter mid, uas
26     cout<<"Keterangan = " <<ket <<endl ; // cetak teks 'Keterangan ='
27     // cetak data yang dikandung var ket
28 }

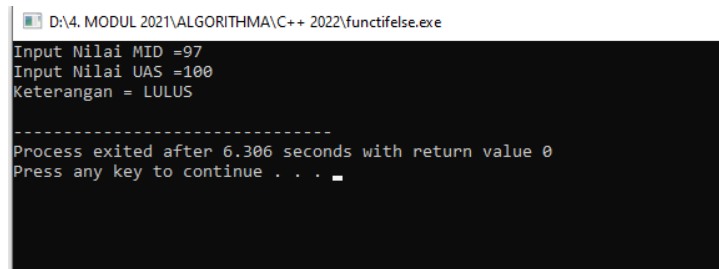
```

Perbedaan antara program **functhasiltes.cpp** dan **functifelse.cpp** terletak pada baris 9 sd 11. Program **functhasiltes.cpp** berfungsi menyeleksi dua var nilai mid dan nilai uas jika memenuhi maka ada aksi yaitu LULUS, jika tidak memenuhi aksi diabaikan. Sedangkan program **functifelse.cpp** berfungsi menyeleksi dua var nilai mid dan nilai uas jika memenuhi maka ada aksi yaitu LULUS, jika tidak memenuhi ada aksi yaitu GAGAL. Ekspresi logika di baris ke-7 pada program **functhasiltes.cpp** dan **functifelse.cpp** termasuk kompleks, menyeleksi dua kondisi menggunakan operator OR sebelum melakukan aksi True yaitu LULUS atau aksi False yaitu GAGAL.

Teruslah berlatih, pahami *statement* pada kedua program C++ diatas, agar anda memiliki *skill*.

Output Program

Pengujian kali ini diinput nilai mid = 97, dan uas = 100, tampak hasil dari program Keterangan LULUS, lihat gambar 19! Karena kedua nilai memenuhi yaitu nilai lebih dari 65.



```

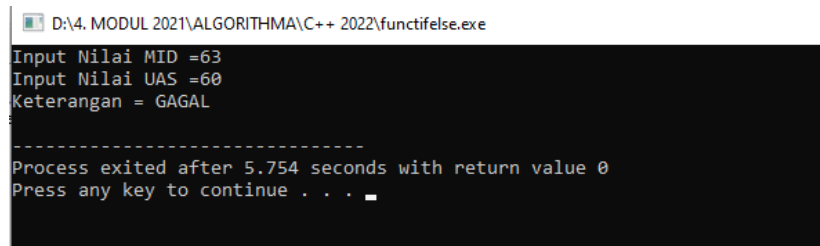
D:\4. MODUL 2021\ALGORITHMA\C++ 2022\functifelse.exe
Input Nilai MID =97
Input Nilai UAS =100
Keterangan = LULUS

-----
Process exited after 6.306 seconds with return value 0
Press any key to continue . . .

```

Gambar 19 Output Nilai mid = 97, Keterangan Lulus

Pengujian kali ini diinput nilai mid diinput 63 dan nilai UAS diinput 60 tampak hasil dari program Keterangan GAGAL. Karena kedua nilai mid dan uas tidak melebihi 65, lihat gambar 20! Silahkan dilanjutkan percobaan, input kombinasi nilai mid dan uas yang berbeda, amati dan simpulkan. Perhatikan dengan cermat bagaimana karakter operator OR dalam program !



```

D:\4. MODUL 2021\ALGORITHMA\C++ 2022\functifelse.exe
Input Nilai MID =63
Input Nilai UAS =60
Keterangan = GAGAL

-----
Process exited after 5.754 seconds with return value 0
Press any key to continue . . .

```

Gambar 20 Output Nilai mid = 63 dan nilai uas = 60, Keterangan Gagal

4. Percobaan

Mahasiswa diminta melakukan percobaan soal-soal latihan dibawah ini, tulis program *flowchart* dan program C++ !

1. Sebuah balok memiliki ukuran panjang 200 cm, lebar 2 cm dan tinggi 10 cm maka berapakah luas permukaan dari balok tersebut?
Diketahui rumus luas permukaan balok yaitu: $L = 2 \times (p \times l) + (p \times t) + (l \times t)$
2. Buat program konversi gram ke kilo gram, misal hitung 800 gram = .. kg. diketahui 1 kg = 1000 gram.
3. Hitung keliling bangun persegi panjang
4. Hitung luas bangun kubus

5. Input 3 bilangan bulat yang berbeda, kemudian tentukan bilangan manakah yang terbesar dari ketiganya. Misal bilangan: 70, 20 dan 100 atau bilangan: 8, 6, dan 1

TENTUKAN BILANGAN TERBESAR	
Bilangan-1	: <u>70</u>
Bilangan-2	: <u>20</u>
Bilangan-3	: <u>100</u>

5. Laporan Praktikum

Kumpulkan hasil percobaan di atas, tambahkan dalam laporan *flowchart* dan program C++ dari setiap percobaan tsb !

Beri Penjelasan dan kesimpulan singkat !

MODUL 4 FUNCTION DAN PROCEDURE BAGIAN-1

1. Tujuan

- Memahami penulisan program per modul sehingga program yang kompleks di buat menjadi sederhana.
- Memahami konsep *modular*
- *Reusable*, sekali buat bisa dipakai berulang-ulang

2. Dasar Teori

Function adalah sebuah anak program yang hanya dapat mengerjakan satu penyelesaian masalah pemrograman. Sebuah function mungkin memiliki nilai parameter yang diterima dari *function caller*, kemudian *function* memprosesnya dan mengembalikan sebuah nilai. Sebuah Function berisi sederetan code atau script program yang disusun berdasarkan logika pemecahan sebuah masalah *programming*.

Nama lain dari Function yaitu sub-routine, procedure atau method. Simbol yang digunakan mirip simbol proses tetapi ditemukan ada garis vertikal ganda dimasing-masing sisi kanan dan kiri, lihat gambar dibawah.



Gambar 21 Function

Konsep Kerja Function

Konsep kerja Function analog dengan Pekerja yang ditugaskan oleh **Bos** dalam satu perusahaan untuk mengerjakan sesuatu. Pekerja diberi bahan dan metoda untuk diproses atau diolah. Pekerja mungkin tidak tahu untuk apa sesuatu hal dikerjakan, dia bekerja tetapi hasil pekerjaannya diberikan kepada **Bos**.

Syntax

```
// pendefenisian fungsi
data type namaFunction( parameter ){

    ..
    ..

    return ..
}

main() {
    ..
    ..
    // pemanggil fungsi
    var = namaFunction ( parameter );
}
```

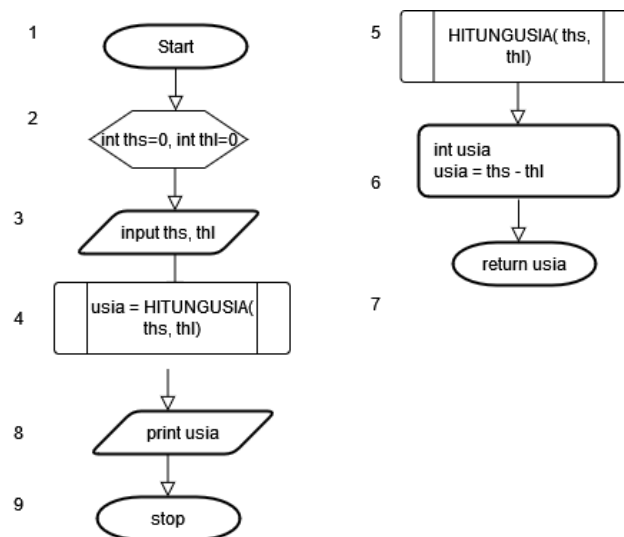
3. Tugas Pendahuluan

Contoh

Buatlah program *flowchart* dan program C++ untuk menghitung usia.

Program Flowchart

Program *flowchart* untuk menghitung usia menggunakan function dan cara kerja programnya sebagai berikut, lihat gambar 22! Terdiri dari 8 tahap



Gambar 22 Flowchart Hitung Usia

Algoritma

1. Start.
2. inisialisasi variabel ths=0, thl =0
input ths, thl

3. input **ths**, **thl**
4. Program utama memanggil function **HITUNGUSIA** dengan mengirim dua nilai parameter **ths** dan **thl**
5. Function bekerja: Function menerima kopian data masing-masing nilai **ths** (tahun sekarang) dan **thl**(tahun lahir)
6. Proses nilai parameter **ths** - **thl**. Logika program kembali ke program utama dan membawa hasil proses yaitu usia dan program utama menerima data disimpan pada variabel **usia**
7. Print usia
8. Stop

Pemakaian simbol Start dan Stop pada function terdapat pada program utama, function atau pekerja ditugaskan oleh program utama, ***caller function***, menghitung atau memproses sesuatu dan hasilnya dikembalikan ke program utama. Program utama memberikan nilai parameter seperti data **ths** dan **thl** ke function untuk diproses. Function bekerja dan mengembalikan nilai **usia** ke program utama. Pemanggil function atau ***caller function*** tidak selalu dilakukan oleh program utama melainkan bisa sesama function tergantung bagaimana *programmer* merancang logika program aplikasi.

Keuntungan pemakaian *function* dalam pembuatan program aplikasi akan tampak pada sebuah penyelesaian masalah yang kompleks, kegiatan itu memerlukan 2 atau lebih programmer. Pada pelaksanaannya suatu *function* yang dibuat programmer ke-1 programmer ke-2, ke-3 dan seterusnya langsung bisa menggunakannya, hal seperti ini disebut ***reusable***. Sekali buat dapat dipakai berulang-ulang. Konsep kerja seperti ini sangat cocok diterapkan dalam ***team work***. Bagaimana program ke-2 dan yang lain bisa menggunakannya ? programmer pertama harus membuat dokumen tentang:

1. Nama function;
2. Kegunaannya, output yang dihasilkan dan
3. Parameter dan type data yang diperlukan.

Contoh penerapan dapat dilihat dari proses pengerjaan pembuatan Aplikasi Sistem Informasi Manajemen seperti Aplikasi Sistem Informasi Kepegawaian, programmer pertama

membuat function HITUNGUSIA dengan menentukan hal dibawah ini dan mendokumentasikannya.

1. Nama function HITUNGUSIA
2. parameter: tahun sekarang dan tahun lahir,
3. Output: usia

Programmer ke-2 yang sedang mengerjakan Laporan Honor Tunjangan Pegawai dapat menggunakannya dengan mengirim paramater: tahun sekarang (ths) dan tahun lahir (thl), begitu juga progammer ke-3 yang sedang mengerjakan Laporan Kepangkatan Pegawai yang memerlukan perhitungan Masa Kerja bisa menggunakan function HITUNGUSIA dengan cara mengirim paramater: tahun sekarang dan tahun masuk kerja. Pembuatan Laporan Pegawai Pensiun, dimana perusahaan telah menetapkan masa usia pensiun adalah 58 tahun, progammer mengirim parameter: tahun lahir (thl) dan tahun sekarang (ths)

Pola kerja seperi ini para programmer menerapkan konsep **modular** yaitu program dibangun per module. Pola kerja seperi ini dalam team work waktu yang digunakan menjadi efektif.

Keuntungan lain yaitu seorang programmer yang ahli menggunakan *function* akan mudah baginya menguasai konsep object oriented programming (OOP). Konsep OOP ini telah bisa diterapkan dalam PHP. Saat ini kemajuan teknologi IT dalam penerapan OOP telah berhasil diciptakan berbagai macam framework seperti CodeIgniter, Yii, Laravel, CakePHP, dll.

Program C++ HITUNGUSIA

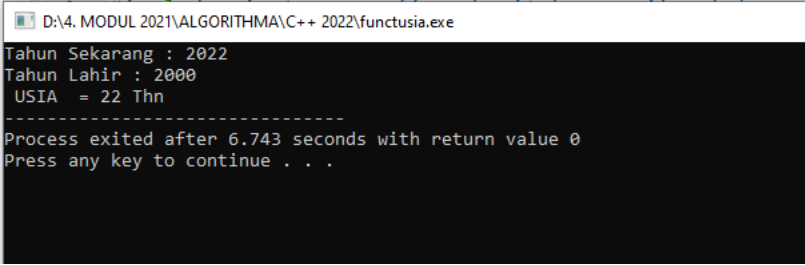
Program C++ disimpan dengan nama **functusia.cpp**. Perhatikan ada penjelasan program disisi kanan setiap baris program C++, ikuti secara cermat, pahami !

```

[*] functusia.cpp
1  #include <iostream> // memberitahu compiler bahwa program menggunakan library iostream
2  using namespace std; // berisi ttg perintah mencetak text atau data ke layar / screen
3
4  int HITUNGUSIA(int ths, int thl) // int = bilangan bulat, parameter ths dan thl
5  { // awal blok program
6      return ths - thl; // dikurangkan kedua var ths dan thl, hasil dikembalikan
7      // ke program utama, ke baris 17
8  } // akhir blok program
9  main () { // program utama ditandai keyword main
10     int ths=0; // var ths type data int=bil bulat
11     int thl=0; // var thl type data int=bil bulat
12     int usia =0; // var usia type data int
13     cout<<"Tahun Sekarang : "; // cetak teks Tahun Sekarang di layar
14     cin>>ths; // input tahun sekarang dari keyboard dan simpan di var ths
15     cout<<"Tahun Lahir : "; // cetak teks Tahun Lahir di layar
16     cin>>thl; // input tahun lahir dari keyboard dan simpan di var thl
17     usia = HITUNGUSIA( ths, thl); //panggil function HITUNGUSIA, kirim parameter ths, thl
18     cout << " USIA = " // cetak teks USIA di layar
19     <<usia << " Thn"; // cetak data usia dan teks Ths
20 }

```

Output



```

D:\4. MODUL 2021\ALGORITHMA\C++ 2022\functusia.exe
Tahun Sekarang : 2022
Tahun Lahir : 2000
USIA = 22 Thn
-----
Process exited after 6.743 seconds with return value 0
Press any key to continue . . .

```

Gambar 23 Output Function HITUNGUSIA

Program functusia.cpp berisi function HITUNGUSIA, hasil data usia yang diperoleh tergantung dari data `ths = 2022` dan `data thl 2000` yang diinput. Setelah data `ths`, data `thl` diproses pada function HITUNGUSIA diperoleh output usia 22 Thn. Data usia yang diperoleh bisa berubah-ubah tergantung dari data `ths` dan `thl` yang diinput. Coba run kembali program diatas, dan input tahun sekarang dan tahu lahir anda! Kemudian perhatikan output yang dihasilkan !

4. Percobaan

Mahasiswa diminta melakukan percobaan soal-soal latihan dibawah ini, tulis program *flowchart* dan program C++ !

1. Buat program untuk penghitungan pemakaian air PDAM per bulan, rumus yang dipakai yaitu

$$\text{Pemakaian} = \text{Meteran Bulan Ini} - \text{Meteran Bulan Lalu}$$

HITUNG PEMAKAIAN AIR	
PDAM TIRTA ZERNIH	
Meteran Bulan Ini (M ³)	: <u>450</u>
Meteran Bulan Lalu (M ³)	: <u>300</u>
Pemakaian Air (M ³)	: <u>150</u>

- Input salah satu angka bulat kemudian tentukan apakah bilangan tersebut bilangan Ganjil atau Genap
- Ketentuan perhitungan Gaji Pokok Karyawan

Gaji Pokok (GP) seorang karyawan mengacu kepada pendidikannya sbb:

Jika pendidikan SD maka Gaji pokok Rp. 1400000,-

Jika pendidikan SLTP maka Gaji pokok Rp. 1750000,-

Jika pendidikan SLTA maka Gaji pokok Rp. 1900000,-

Jika pendidikan DIPLOMA 3 (D3) maka Gaji pokok Rp. 2400000,-

Jika pendidikan S1 maka Gaji pokok Rp. 3200000,-

GAJI KARYAWAN	
NAMA KARYAWAN	: <u>Syfa</u>
PENDIDIKAN	: <u>S1</u>
Gaji Pokok	:Rp. 3200000

5. Laporan Praktikum

Kumpulkan hasil percobaan di atas, tambahkan dalam laporan *flowchart* dan program C++ dari setiap percobaan tsb !

Beri Penjelasan dan kesimpulan singkat !

MODUL 5 FUNCTION DAN PROCEDURE BAGIAN-2

1. Tujuan

- Memahami penulisan program per modul sehingga program yang kompleks di buat menjadi sederhana.
- Memahami konsep *modular*
- *Reusable*, sekali buat bisa dipakai berulang-ulang

2. Dasar Teori

Procedure

Procedure cukup berguna, dan akan menyenangkan untuk dapat menggunakannya kembali di seluruh program, daripada terus menulis kode yang sama. Ini akan memiliki beberapa keuntungan:

- Jika Anda ingin mengubahnya, Anda hanya perlu mengubah prosedurnya
- Menulis kode sekali berarti Anda hanya men-debug sekali
- Hal itu akan menghemat banyak mengetik
- Memisahkan potongan kode ini dan memberinya nama membuat tujuannya lebih jelas

Cara menulis procedure

Syntax untuk *procedure*

```
void namaProcedure() {
    ..
    ..|
}
main() {
    namaProcedure();
}
```

Ciri-ciri procedure sebagai berikut

1. Diawali keyword **void**. Maksud keyword void, dimana sebuah *prodecure* tidak mengembalikan nilai ke program utama atau program lain yang memanggilnya, tidak seperti halnya function
2. Parameter pada function tidak mesti ada, opsi
3. Pada bagian *procedure* **tidak menggunakan keyword return** untuk pengembalian nilai data ke program pemanggil (*caller function*)

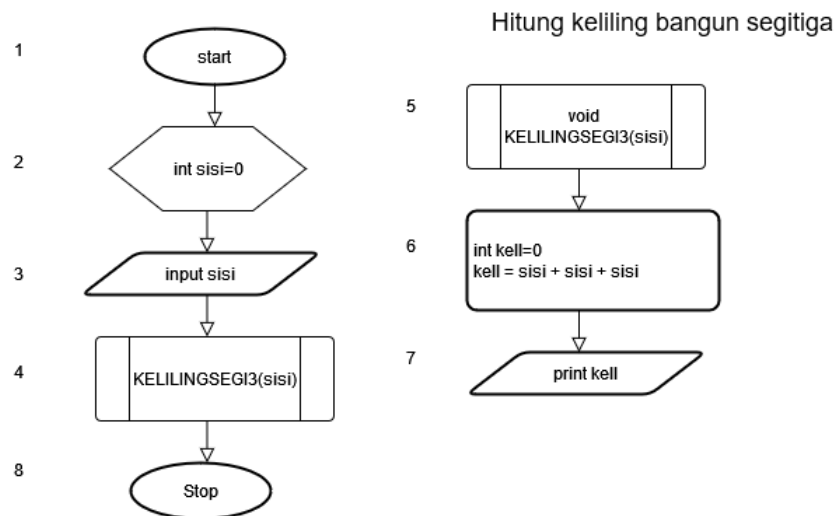
3. Tugas Pendahuluan

Contoh

Hitunglah keliling segitita dimana rumus keliling = sisi + sisi + sisi !

Program Flowchart

Program *flowchart* untuk menghitung keliling bangun segitiga menggunakan procedure dan cara kerja programnya sebagai berikut, lihat gambar 24! Terdiri dari 8 tahap



Gambar 24 Flowchart Procedure Keliling Segi 3

Algoritma

1. Start.
2. inisialisasi variabel sisi=0
3. input sisi
4. Progam utama memanggil procedure KELILINGSEGI3 dengan mengirim 1 nilai parameter yaitu sisi
5. Procedure KELILINGSEGI3, menerima parameter sisi

6. Procedure bekerja
 - a. Deklarasikan var kell, set nilai awal 0
 - b. Hitung $kell = sisi + sisi + sisi$
7. Print kell dan logika program kembali ke program utama
8. Stop

Program C++ Keliling Segi Tiga

Program C++ disimpan dengan nama **procedurekell.cpp**. Perhatikan ada penjelasan program disisi kanan setiap baris program C++, ikuti secara cermat dan pahami !

Tanda **void** pada baris ke 4 artinya sebagai berikut

1. Nama KELILINGSEGI3 adalah *procedure*, bukan *function*
2. Tidak ditemukan keyword **return** pada procedure KELILINGSEGI3, akibatnya *procedure* ini tidak bisa mengembalikan data hasil proses ke program utama (dipanggil pada baris 18)

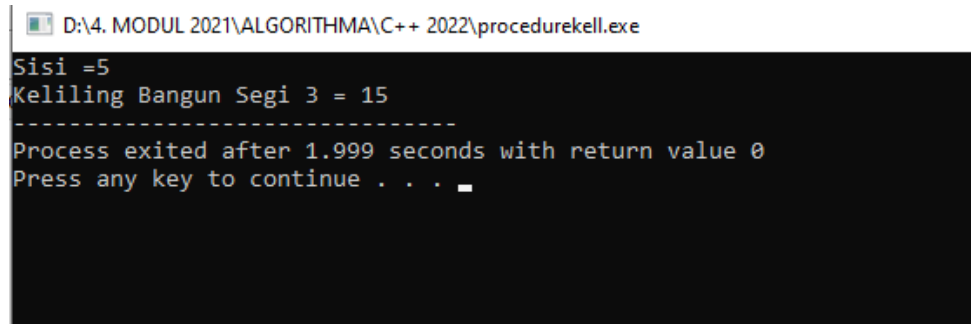
```

Untitled1 procedurekell.cpp
1 #include <iostream> // memberitahu compiler bahwa program menggunakan library iostream
2 using namespace std; // berisi ttg perintah mencetak text atau data ke layar / screen
3
4 void KELILINGSEGI3 (int sisi) { // procedure KELILINGSEGI3 menerima parameter sisi
5     int kell=0; // deklarasikan kell set nilai awal 0
6     kell= sisi + sisi + sisi; // hitung kell, jumlahkan sisi
7     cout<<"Keliling Bangun Segi 3 = " // cetak teks spt disebelah
8     << kell; // print data kell yaitu 15, logika program balik ke program Utama
9
10 }
11
12 main () { // main= program utama, logika program dimulai dari sini
13     int sisi=0; // deklarasikan var sisi set nilai awal 0, type data int
14
15     cout<< "Sisi = " ; // cetak teks 'Sisi' di layar
16
17     cin>> sisi; //input data sisi dari keyboard misal 5
18     KELILINGSEGI3 (sisi); // panggil function KELILINGSEGI3 kirim nilai parameter sisi yaitu 5
19
20 }
  
```

Output

Output program **procedurekell.cpp** adalah 15 bersumber dari data sisi bernilai 5 yang diinput dari keyboard, lihat gambar 25. Anda bisa me- run program ini kembali dan mengubah input data sisi dengan angka lain. Data keliling segitiga diperoleh dengan rumus yang ditemukan pada baris 6

Amati, *output* yang ditampilkan program ini setiap selesai melakukan input data



```

D:\4. MODUL 2021\ALGORITHMMA\C++ 2022\procedurekell.exe
Sisi =5
Keliling Bangun Segi 3 = 15
-----
Process exited after 1.999 seconds with return value 0
Press any key to continue . . .

```

Gambar 25 Procedure KELILINGSEGI3, Parameter sisi bernilai 5

4. Percobaan

Mahasiswa diminta melakukan percobaan soal-soal latihan dibawah ini, tulis program *flowchart* dan program C++ !

1. Diakhir semester dosen mengelola data nilai untuk mendapatkan nilai akhir matakuliah. Nilai akhir dihitung berdasarkan 3 komponen nilai dengan persentasi yang berbeda yaitu
 - a. Persentasi nilai tugas 20%
 - b. Persentasi nilai ujian tengah semester (MID) 30%, dan
 - c. Persentasi nilai ujian akhir semester (UAS) 50%

Buatlah program *flowchart* dan program C++ untuk menghitung nilai akhir dari suatu matakuliah. Lihat *layout*, bentuk tampilan dibawah ini !

Cobalah untuk beberapa *record* data nilai berikut:

90, 80, 100 → (92)

56, 70, 90 → (77,2)

98, 65, 73 → (75,6) dan

57, 65, 77 → (69,4)

NILAI MATAKULIAH	
Nobp	: 2111005
Nama	: Gophar Hadi
Nilai Tugas	: 90
Nilai MID	: 80
Nilai UAS	: 100
Nilai Akhir	: ... 92

2. Pada suatu perusahaan, gaji yang diterima pegawai terdiri dari komponen gaji: gaji pokok, tunjangan dan upah lembur. Dasar untuk menghitung tunjangan 15% dari gaji pokok. Ditetapkan upah lembur sebesar Rp. 40.000 per jam.
- Buatlah program untuk menghitung gaji yang diterima pegawai.
- Masukan (Input) yaitu nomor pegawai, nama, gaji pokok pegawai, jumlah jam lembur.
- Keluaran(Output): Gaji pokok, tunjangan, uang lembur, total gaji

PT MANDIRI K	
SLIP GAJI	
No Pegawai	: 13110
Nama	: Mutiara
Gaji Pokok Rp.	: 2000000
Jumlah Jam Lembur	: 10
Gaji Pokok Rp.	: 2.000.000
Tunjangan Rp.	:
Uang Lembur Rp.	: +
Total Gaji Rp.	:

3.

5. Laporan Praktikum

Kumpulkan hasil percobaan di atas, tambahkan dalam laporan *flowchart* dan program C++ dari setiap percobaan tsb !

Beri Penjelasan dan kesimpulan singkat !

MODUL 6 STUKTUR KONTROL IF DAN IF ELSE

1. Tujuan

- Memahami logika percabangan tunggal dan ganda
- Memahami logika percabangan majemuk
- Mampu menuliskan bahasa pemrograman C++

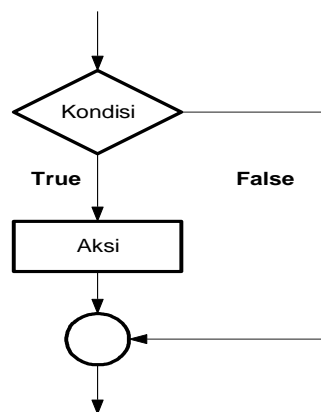
2. Dasar Teori

Struktur IF

Statement **if** digunakan untuk menyeleksi kebenaran suatu kondisi atau logika percabangan tunggal yang diproses. Statement IF digunakan untuk menyelesaikan masalah programming yang sederhana. Pemakaian **if** digunakan bila suatu keadaan programming memerlukan jawaban dari suatu kondisi yakni kondisi Benar atau True atau kondisi Salah atau False.

Jika Kondisi Benar terpenuhi maka ada **Aksi**, sebaliknya jika kondisi Salah maka **aksi diabaikan**. Kondisi logika benar diekspresikan flowchart alur ke bawah menggunakan logika *True*, sebaliknya untuk kondisi tidak terpenuhi digambarkan alur ke arah kanan, gunakan logika *False*.

Ekspresi logika if dapat di ekspresikan dengan symbol flowchart berikut



Gambar 26 Struktur IF

Pseudocode:

1. Jika kondisi memenuhi maka
 - o ada Aksi, tulis code disini
2. jika tidak memenuhi

- o Abaikan

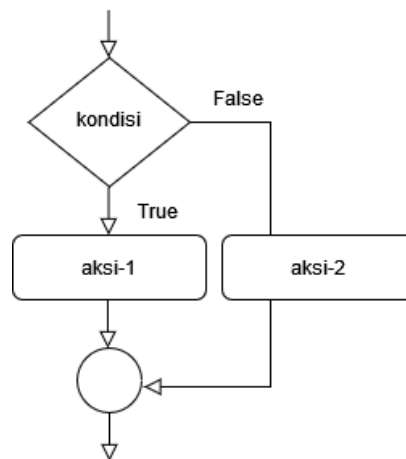
Syntax

```
if (kondisi) {
    //aksi
}
```

Struktur IF.. ELSE

Statement **if..else** digunakan untuk menyeleksi kebenaran suatu kondisi atau logika percabangan jamak yang diproses. Statement IF..ELSE digunakan untuk menyelesaikan masalah programming yang mulai kompleks. Pemakaian **if..else** digunakan bila suatu keadaan programming memerlukan jawaban dari suatu kondisi yakni kondisi Benar atau True atau kondisi Salah atau False.

Gambar 27 menggambarkan logika if .else. Jika kondisi terpenuhi maka kerjakan alur logika True yaitu aksi-1, sebaliknya jika kondisi tidak terpenuhi maka kerjakan alur logika False yaitu aksi-2. Artinya selalu ada Aksi dari jika kondisi terpenuhi atau tidak terpenuhi.



Gambar 27 Struktur IF..ELSE

Pseudocode:

1. Jika kondisi memenuhi maka
 - o ada Aksi-1, tulis code disini
2. jika tidak memenuhi
 - o ada Aksi-2, tulis code disini

Syntax

```

if (kondisi) {
    //aksi-1
}else {
    //aksi-2
}

```

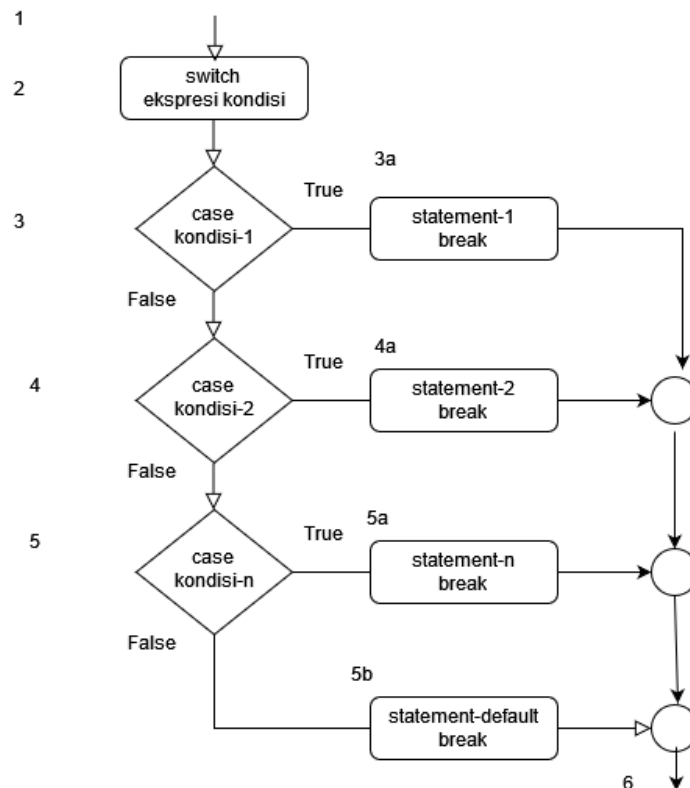
Struktur SWITCH.. CASE

Statement **switch-case** membantu dalam menguji kesetaraan variabel terhadap satu set nilai. Setiap nilai yang dibandingkan dikenal sebagai kasus.

Lihat switch sebagai pernyataan cabang pilihan majemuk. Anda dapat menggeser eksekusi program ke berbagai bagian berdasarkan nilai ekspresi.

Jumlah case dalam struktur switch-case tidak dibatasi

Flowchart switch case bisa dilihat dibawah ini



Gambar 28 Flowchart Switch-Case

Pseudocode:

1. Alur logika dimulai dari arah atas

2. Switch (ekspresi)
3. Case kondisi-1
 - Statement-1
 - Break
4. Case kondisi-2
 - Statement-2
 - Break
5. Case kondisi-n
 - Statement-n
 - Break
- Else
 - default
 - break
6. exit

Syntax

```
switch (variable)
{
    case 1:
        break;
    case 2:
        break;
    default:

```

3. Tugas Pendahuluan

Contoh Switch Case

Data hasil tes Cetaklah nilai huruf (nh) berdasarkan nilai angka (na)

```

1 #include<iostream> //nestedloop-while.cpp
2 using namespace std;
3
4 char NILAIHURUF(int na){
5     char nh;
6     switch( na ){
7         case 100:
8             nh='A';
9             break;
10        case 90:
11            nh='B';
12            break;
13        default:
14            nh='-';
15            break;
16    }
17    return nh;
18 }
19 main(){
20     int na=100;
21
22     char nh = NILAIHURUF(na);
23     cout<< "Nilai Huruf : "<<nh;
24 }

```

Output

```

D:\4. MODUL 2021\ALGORITHMA\C++ 2022\switch-case.exe
Nilai Huruf : A
-----
Process exited after 0.1407 seconds with return value 0
Press any key to continue . . .

```

4. Percobaan

1. Hitunglah upah karyawan pada PT. Oto. Diketahui, jam kerja normal karyawan 40 jam per minggu dan diberikan **upah** sebesar Rp. 200.000 per minggu, jika jam kerja kurang dari normal, maka Upah sebesar Rp. 150.000 per minggu. Untuk karyawan jam kerja diatas 40 jama maka ada **jam lembur**.

Honor lembur diberikan jika jam kerja melebihi jam kerja normal dengan ketentuan honor Lembur sebesar Rp. 10.000,- per jam.

UPAH KARYAWAN	
Nama	: M. Rohiid
Jam Kerja (jam)	: 45
Upah Rp.	:
Jam Lembur (jam):	...
Honor Lembur Rp. :
	----- +
Upah Karyawan Rp.

Ingat karyawan tidak dibolehkan kerja melebihi 52 jam per minggu. Silahkan buat sendiri bentuk luaran / output

Contoh data: nama karyawan: M Rohiid dengan 45 jam kerja.

UpahKaryawan = Upah + HonorLembur

3. Selesaikan persoalan programming dibawah ini !
 Buat Program Flowchart. Cetaklah Slip Pembayaran Rekening Tagihan PDAM seperti output dibawah ini ! Diketahui biaya pemakaian per meter kubik air (HPKA) yaitu Rp. 2000.
 Jawab menggunakan FUNCTION.

Besarnya Pemakaian air per bulan dihitung dengan rumus :

PemakaianAir = Meteran Bulan Ini – Meteran Bulan Lalu

Setiap Pelanggan dikenakan biaya Administrasi (byadm) menurut kelompok pelanggan sebagai berikut:

Kelompok Pelanggan RT byAdm Rp. 20.000

Kelompok Pelanggan IP byAdm Rp. 30.000

Kelompok Pelanggan ID byAdm Rp. 40.000

Sedangkan besarnya pembayaran rekenign tagiha PDAM meggunakan rumus:

Rekening = PemakaianAir x HPKA sedangkan

TagihanRekening = Rekening + byAdm

Note: rt=rumah tangga, ip=instansi pemerintah, id=industri

TAGIHAN REKENING PDAM	
Pelanggan	: Qathrunnada Syfa
Biaya Per m3	Rp. 2000
Meteran Bulan Ini (m3)	: 140
Meteran Bulan Lalu (m3)	: 120 -
Kelompok Pelanggan	: rt

Pemakaian (m3)	20
Biaya Administrasi	: Rp. 20.000,-
Rekening	: Rp. 40.000,-
	----- +
Tagihan Rekening	: Rp. 60.000,-

2. Laporan Praktikum

MODUL 7 STUKTUR KONTROL LOOP

1. Tujuan

- Mampu memahami dan memprediksi perilaku loop: *while*, *for* dan *do-while*
- Mampu menulis *loop while*, *for* dan *do-while* yang *valid*.
- Mampu menjelaskan dan memberikan contoh bagian kondisi dari *loop while*, *for* dan *do-while*.
- Mengetahui representasi flowchart standar untuk *loop while*, *for* dan *do-while*.
- Mengetahui representasi pseudocode standar untuk *loop while*, *for* dan *do-while*.
- Ketahui cara membuat indentasi loop *while*, *for* dan *do-while* agar dapat dibaca.

2. Dasar Teori

Looping atau perulangan. Pernyataan loop memungkinkan untuk dieksekusi pernyataan atau sekelompok pernyataan beberapa kali.

Mungkin ada situasi, ketika Anda perlu mengeksekusi blok kode beberapa kali. Secara umum, pernyataan dieksekusi secara berurutan: Pernyataan pertama dalam suatu fungsi dieksekusi terlebih dahulu, diikuti oleh yang kedua, dan seterusnya disebut *looping*.

Bahasa pemrograman menyediakan berbagai struktur kontrol yang memungkinkan jalur eksekusi yang lebih rumit.

Penggunaan program *flowchart* selain bisa menggambarkan logika *sequence*, decision juga bisa menggambarkan logika perulangan. Logika looping efektif digunakan untuk menyelesaikan masalah yang ada kemiripan penyelesaian masalah antara proses pertama, kedua, dan seterusnya jika dibandingkan dengan logika *sequence*. Simbol utama looping While yaitu *flows*, *Connector*, *Decision* dan *Process*.

Aturan untuk menggambarkan struktur flowchart **Looping While** diperlukan rangkaian beberapa simbol yaitu: *connector*, *decision*, *proses* dan atau *output*. Logika True digambarkan pada arah kanan dari simbol *decision*, beri tanda **True**, proses, output (jika ada) dan panah alur logika berulang menuju kearah *connector*. Perulangan berakhir jika **kondisi** (yang ditulis pada simbol

Decision) tidak terpenuhi, tandai dengan pilihan logika *False*, terus alur logika mengalir menuju proses selanjutnya.

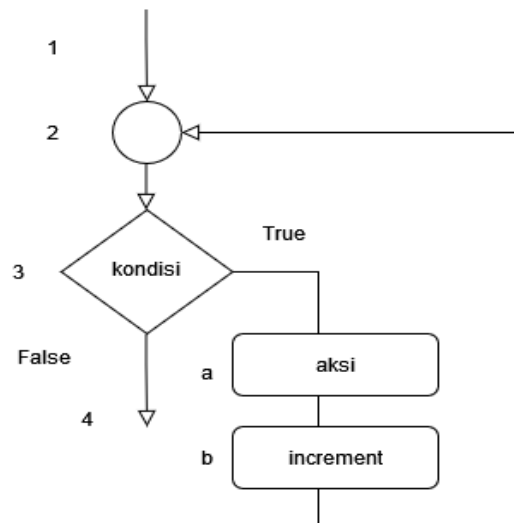
Dalam pemrograman yang kompleks teknik pemrograman *looping* biasa digunakan dalam *array* bertujuan untuk mengenali elemen data yang terkandung dalam suatu *array*. Elemen data tersebut dikenali dengan menggunakan nomor indeks yang ada. Indeks itu bisa diwakili oleh variabel yang dideklarasikan pada decision. Nilai indeks akan terus bertambah bersamaan bertambahnya nilai variabel Pencacah dalam struktur Looping While.

Dalam C++ programming ada 3 jenis looping yaitu: while, for next, dan do .. while.

Berikut ini adalah pernyataan umum dari pernyataan loop di sebagian besar bahasa pemrograman termasuk C++, gambar *flowchart* loop while.

Loop While

Arsitektur Lingkaran



Gambar 29 Flowhart Loop While

Gambar 28 diatas menggambarkan alur logika loop while, mari kita baca alur langkah demi langkah:

Pseudocode

1. Alur logika dimulai dari arah atas

2. Awal perulangan `while`
3. Selama nilai kondisi terpenuhi maka
 - a. Aksi
 - b. *Increment* (penambahan 1, pencacah bertambah 1)
 - c. Menuju awal perulangan `while`
4. End, perulangan berakhir

Penempatan *increment* dalam flowchart tergantung kondisi dari soal kasus yang diselesaikan.

Syntax

Sintaks dasar dari C++ loop `while` adalah sebagai berikut:

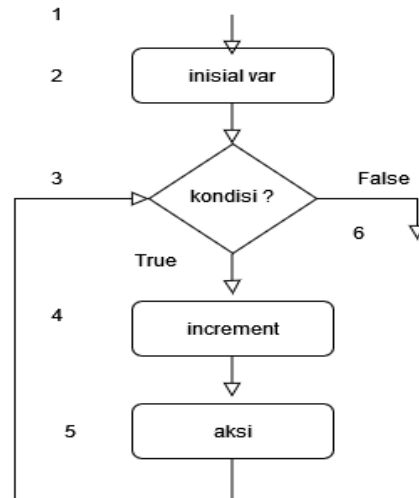
```
while(condition)
{
    // Statements
}
```

Selama kondisi benar kerjakan blok kode (statements), terjadi perulangan atau looping

Loop For

Apa itu Loop For?

Ini adalah struktur kontrol pengulangan yang membantu Anda mengulangi bagian blok code C++ untuk beberapa kali secara tetap. Sebuah blok code *for loop* berjalan asalkan ekspresi pengujian benar. *Loop* menghentikan eksekusi segera ekspresi pengujian menjadi salah, *false*. Ini berarti sebelum eksekusi tubuh loop di setiap iterasi, kondisinya harus dievaluasi. Jika evaluasi mengembalikan nilai *true*, badan loop dieksekusi. Jika evaluasi mengembalikan *false*, eksekusi badan loop dihentikan.



Gambar 30 FLOWchart Loop For

Gambar 29 diatas menggambarkan alur logika loop for, mari kita baca alur langkah demi langkah:

Pseudocode:

1. Alur logika dimulai dari arah atas
2. Set nilai awal var
3. Awal perulangan for, selama kondisi terpenuhi
4. Increment, nilai pencacah bertambah 1
5. Aksi, kerjakan aksi setiap terjadi loop
6. Ulangi step-3
7. Selesai, keluar dari loop for

Syntax

Sintaks dasar dari C++ loop for adalah sebagai berikut:

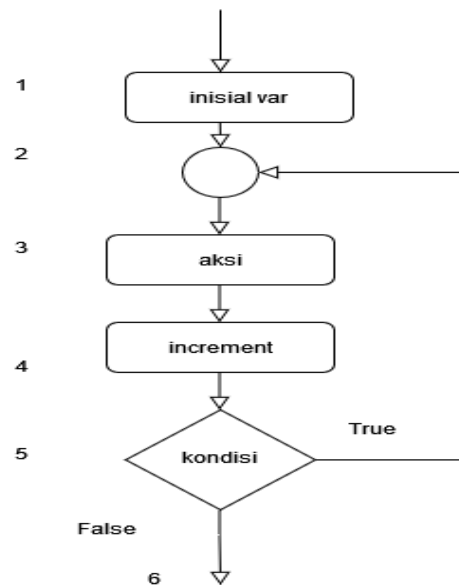
```

for ( initialization;condition;increment ) {
    statement(s);
}
  
```

Loop Do..While

Do-While Loop dapat mengeksekusi blok kode dalam satu loop berdasarkan suatu kondisi. Perulangan do-while mengulang bagian dari program C++ beberapa kali. Dalam perulangan do-while, ekspresi uji ditambahkan di bagian bawah perulangan. Badan loop muncul sebelum ekspresi pengujian. Itu sebabnya badan loop harus dijalankan pertamakali, bahkan ketika ekspresi pengujian bernilai false pada pengujian pertama.

Flowchart



Gambar 31 Flowhart do-while

Gambar 30 diatas menggambarkan alur logika loop do-while, mari kita baca alur langkah demi langkah:

Pseudocode:

7. Inisial var (deklarasikan variable)
8. Alur logika dimulai dari arah atas
9. Aksi: blok code
10. Increment
11. Selama kondisi terpenuhi, ulangi step 1
12. Selesai, keluar dari loop do-while

Syntax

Sintaks dasar dari C++ loop do while adalah sebagai berikut:

```
do{
  //code
}while(condition);
```

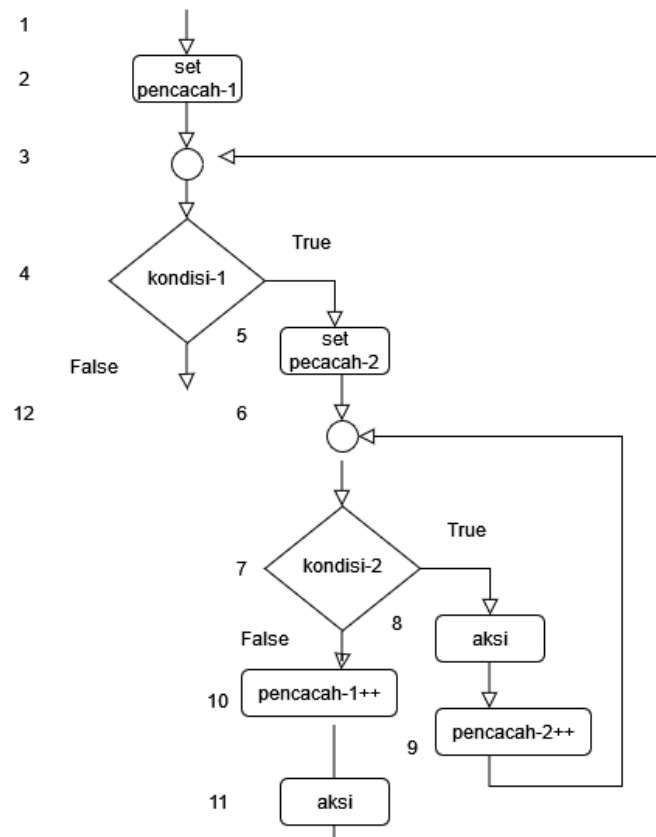
Nested Loop

Dalam modul ini, kita akan belajar tentang *nested loop* atau loop bersarang di C++ dengan bantuan contoh. Dengan kata lain *nested loop* adalah sebuah *loop* dalam *loop*.

Mari kita ambil contoh, misalkan kita ingin mencetak pola 5 baris dan 5 kolom, yang terjadi yaitu mengulang setiap kolom 5 kali, dalam setiap baris selama 5 kali.

Untuk mencapai ini, kita dapat membuat loop untuk diulang lima kali (5 baris). Dan di dalam loop, kita dapat membuat loop lain untuk diulang sebanyak 5 kali (5 kolom). Ini adalah bagaimana kita dapat menggunakan loop bersarang.

Flowchart



Gambar 32 Nested Loop While

Pseudocode

Syntax

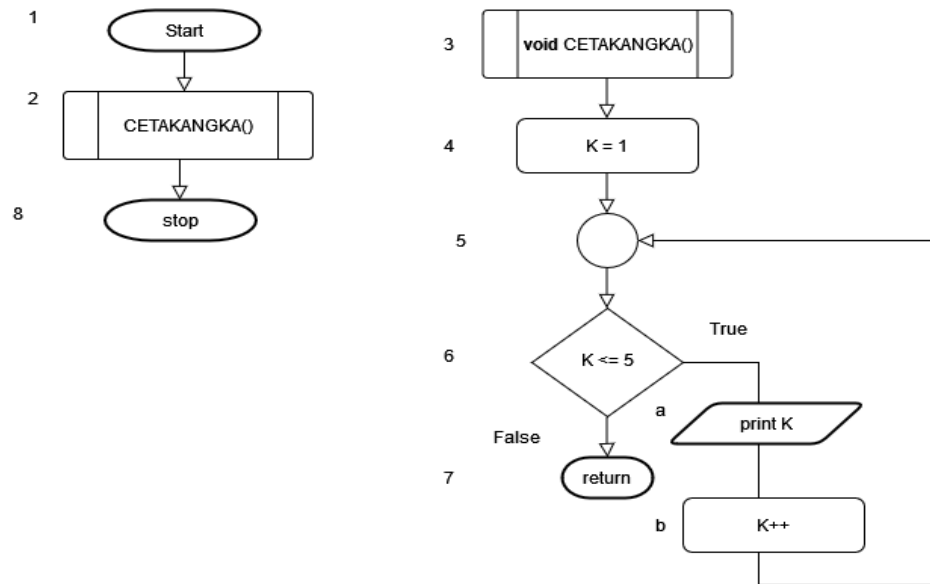
3. Tugas Pendahuluan

Contoh Loop while

Cetaklah angka 1 hingga 5, buat program flowchart dan C++

Output

1 2 3 4 5



Gambar 33 Flowchart Loop-While Cetak Angka 1..5

Gambar 30 diatas menggambarkan alur logika loop while untuk mencetak angka 1 s.d 5, mari kita baca alur langkah demi langkah, seperti *pseudocode* berikut:

Pseudocode

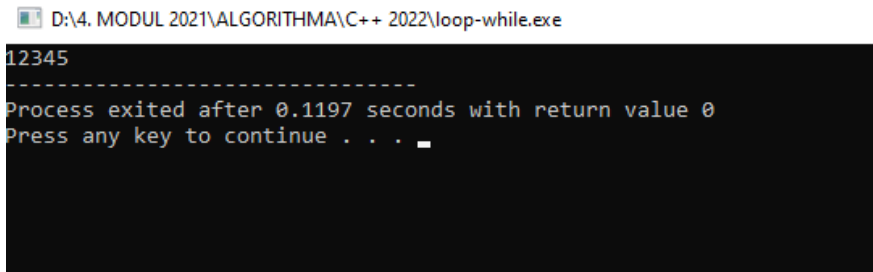
1. Mulai
2. Panggil function CETAKANGKA() (tanpa parameter)
3. Defenisikan function CETAKANGKA()
4. Set nilai var bertipe integer nilai K = 1
5. Mulai loop while
6. While (K <= 5)
 - a. Print K
 - b. K++
 - c. Ulangi proses ke step 5 (loop)
7. Return

8. stop

Program C++ dari contoh soal diatas ditulis dibawah ini. Nama program loop-while.cpp

```
[*] loop-while.cpp
1  #include<iostream>
2  using namespace std;
3
4  void CETAKANGKA(){ // procedure CETAKANGKA ini diawali keyword void
5      int K=1; // nilai K = 1
6      while ( K <= 5){ // selama nilai K <=5, loop
7          cout<<K; // cetak nilai K
8          K++; // nilai K + 1 dan hasilnya simpan di var K
9      }
10     // procedure tidak boleh menggunakan return
11
12 }
13
14 main() {
15     CETAKANGKA(); // panggil function CETAKANGKA
16 }
```

Output



```
D:\4. MODUL 2021\ALGORITHMMA\C++ 2022\loop-while.exe
12345
-----
Process exited after 0.1197 seconds with return value 0
Press any key to continue . . .
```

Gambar 34 Output CETAKANGKA: 1 2 3 4 5

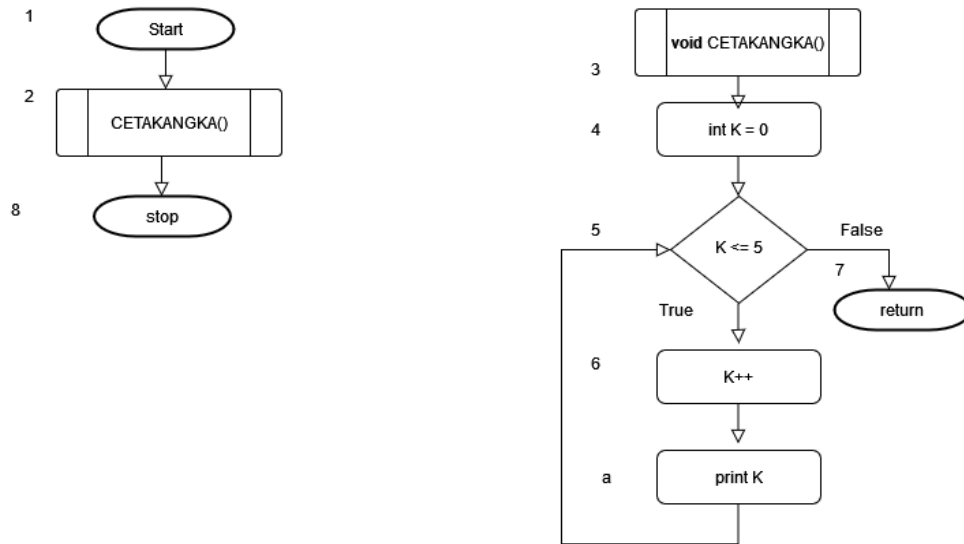
Gambar 31, menampilkan angka yang telah dicetak pada program loop-while.cpp yaitu: 1 2 3 4 5, diproses (diterjemahkan dari source code C++ **loop-while.cpp** menjadi object code **loop-while.exe** dalam wak 0,1197 detik.

Contoh Loop for

Cetaklah angka 1 hingga 5, buat program flowchart dan C++

Output

1 2 3 4 5



Gambar 35 Flowchart Loop-For, Cetak Angka 1 sd 5

Gambar 32 diatas menggambarkan alur logika loop for untuk mencetak angka 1 s.d 5, mari kita baca alur langkah demi langkah, seperti *pseudocode* berikut:

Pseudocode:

1. Mulai
2. Panggil function CETAKANGKA()
3. Defenisikan function CETAKANGKA()
4. Set var K integer, nilai awal 0 → int K =0
5. For (K = 0; K <= 5; K++)
 - a. print K
 - b. Ulangi langkah-5
6. Selesai loop k
7. Alur logika program kembali ke program utama → return
8. Stop

Program C++ dari contoh soal diatas ditulis dibawah ini. Nama program **loop-for.cpp**

Perhatikan pada C++ loop-for.cpp dibawah hasil terjemahan dari flowchart gambar

```

loop-for.cpp
1  #include<iostream>
2  using namespace std;
3
4  void CETAKANGKA(){ // procedure CETAKANGKA ini diawali keyword void
5      // nilai K = 1
6      for(int K=1; K <= 5; K++){ // selama nilai K <=5, Loop. nilai K + 1 dan hasilnya simpan di var K
7          cout<<K<<" "; // cetak nilai K
8
9      }
10     // procedure tidak boleh menggunakan return
11 }
12
13 main() {
14     CETAKANGKA(); // panggil function CETAKANGKA
15 }
16

```

Output

```

1 2 3 4 5
-----
Process exited after 0.2462 seconds with return value 0
Press any key to continue . . .

```

Gambar 36 Output Angka 1 s.d 5

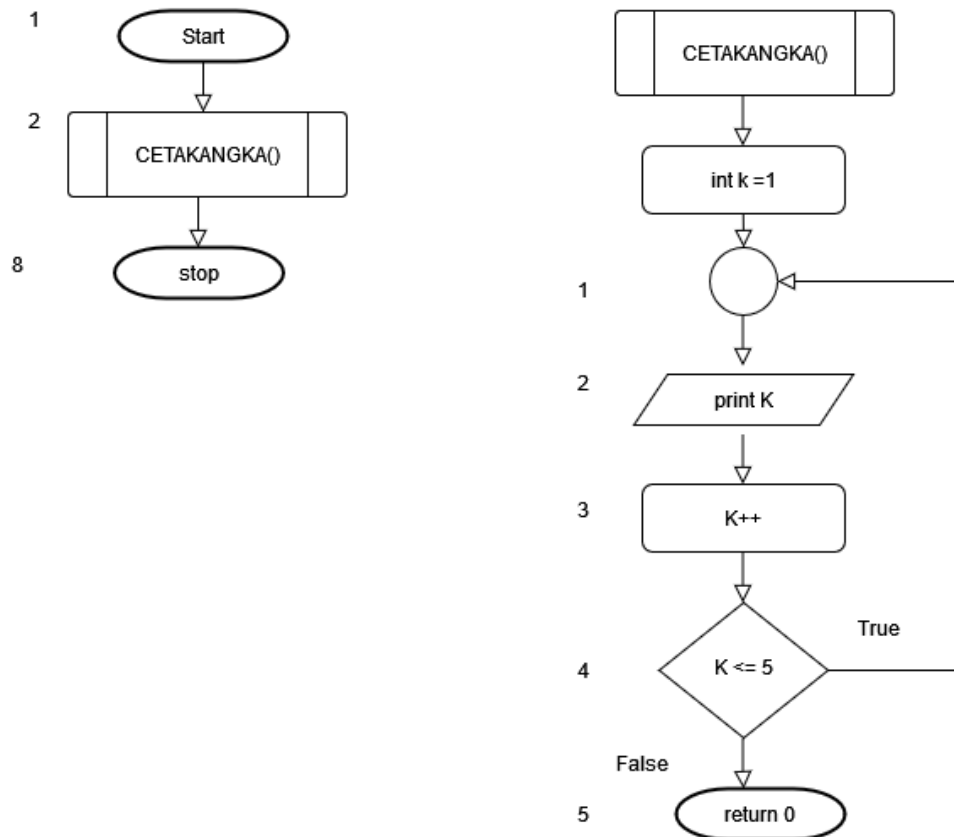
Gambar 34, menampilkan angka yang telah dicetak pada program loop-for.cpp yaitu: 1 2 3 4 5, diproses, diterjemahkan dari source code C++ **loop-for.cpp** menjadi object code **loop-for.exe** dalam wak 0,2462 detik.

Contoh Loop do while

Cetaklah angka 1 hingga 5, buat program flowchart dan C++

Output

1 2 3 4 5



Gambar 37 Flowchart Loop-Do-While, Cetak Angka 1 sd 5

Program C++ dari contoh soal diatas ditulis dibawah ini. Nama program **loop-do-while.cpp**

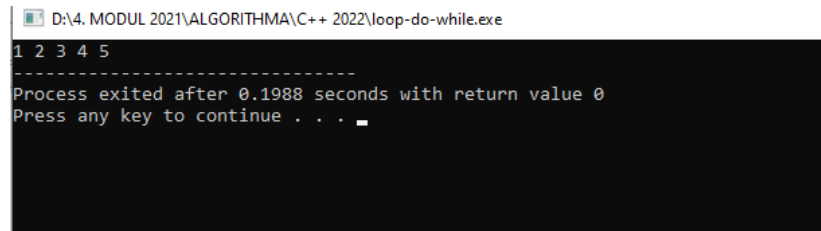
Perhatikan pada C++ loop-do-while.cpp dibawah hasil terjemahan dari *flowchart* gambar

```

loop-do-while.cpp
1  #include<iostream> // loop-do-while.cpp
2  using namespace std;
3
4  // soal: cetaklah angka 1-5
5
6  int CETAKANGKA() {
7      int K=1; //set nilai awal K = 1
8      do {
9          printf("%d ", K); // cetak K
10         K++; // increment K, K = K + 1
11     }while ( K <= 5); // cek selama nila K <= 5, loop
12     return 0; // alur logika kembali ke baris 17
13 }
14
15 main() {
16     CETAKANGKA(); // panggil function baris 7
17 }

```

Output



```
D:\4. MODUL 2021\ALGORITHMMA\C++ 2022\loop-do-while.exe
1 2 3 4 5
-----
Process exited after 0.1988 seconds with return value 0
Press any key to continue . . .
```

Gambar 38 Output Cetak Angka 1 sd 5

Gambar 36, menampilkan angka yang telah dicetak pada program `loop-do-while.cpp` yaitu: 1 2 3 4 5, diproses, diterjemahkan dari source code C++ **loop-do-while.cpp** menjadi object code **loop-do-while.exe** dalam waktu 0,1988 detik.

Contoh Nested Loop

Cetak pola 5 baris dan 5 kolom

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

Flowchart

Flowchart gambar 38 jawaban dari soal diatas *nested loop*. Flowchart tersebut terdiri dari 2 loop while yaitu

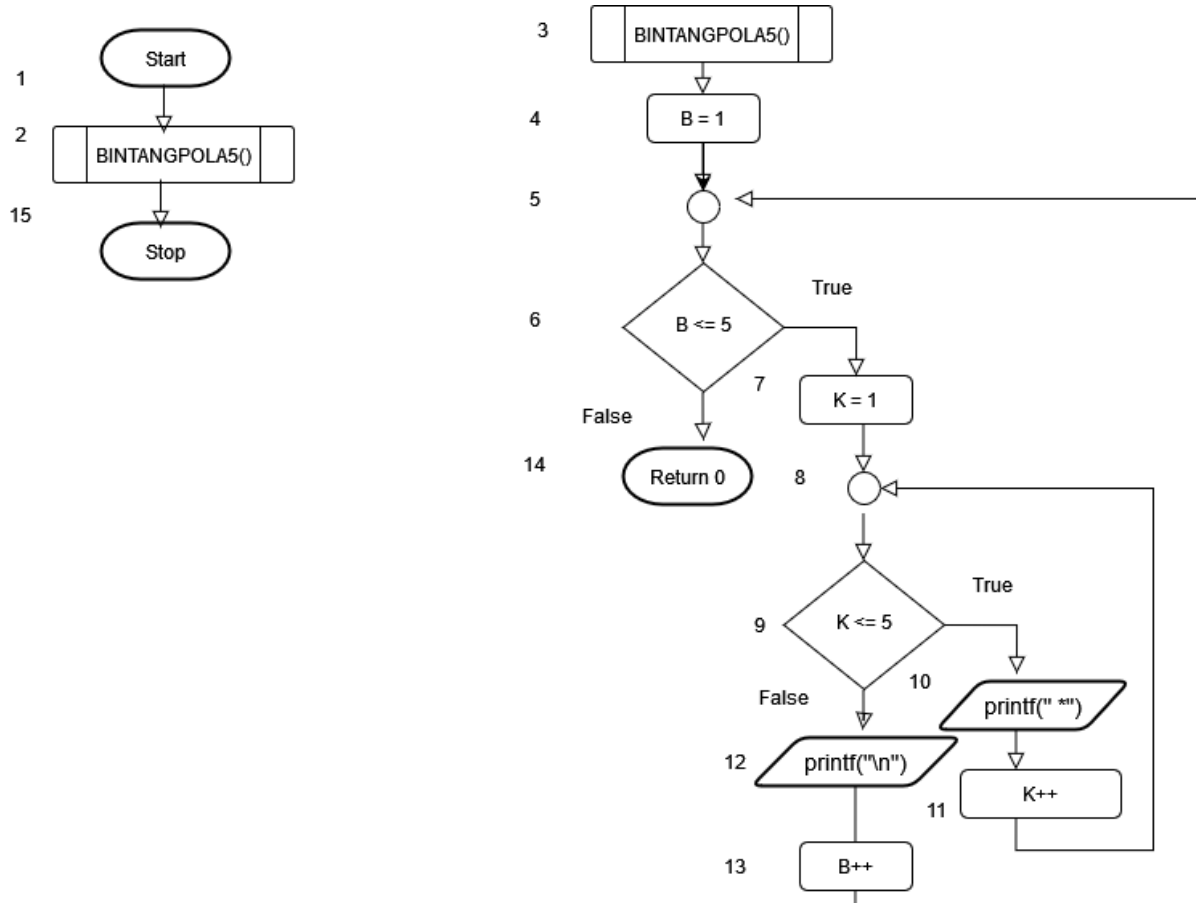
1. Loop while B untuk baris dan
2. Loop while K untuk kolom

Cara kerja *flowchart* ini, flowchart ini diberi nomor 1 s.d 15 menandai urutan alur logika, silahkan ikuti. Alur logika diawali program utama memanggil *function* BITANGPOLA5(), terjadi loop B sebanyak 5 kali untuk membentuk baris, setiap loop B terjadi, maka ada loop K sebanyak 5 kali.

Loop B mulai bekerja dari nilai awal 1 dan diikuti loop K terjadi 5 kali. Setiap terjadi loop K, maka pencacah K bertambah hingga tertinggi bernilai 5. Setiap nilai K bernilai 5 maka:

1. loop K berakhir
2. Terjadi pencacah B yaitu B++

3. Pindah baris baru dan menempatkan symbol * yang baru disana
4. Alur logika terus terjadi hingga membentuk pola 5 baris dan 5 kolom
5. Kendali program kembali ke program utama, ditandai *statement* **return 0**



Gambar 39 Flowchart Nested Loop While, Cetak Pola Bintang 5 Baris dan 5 Kolom

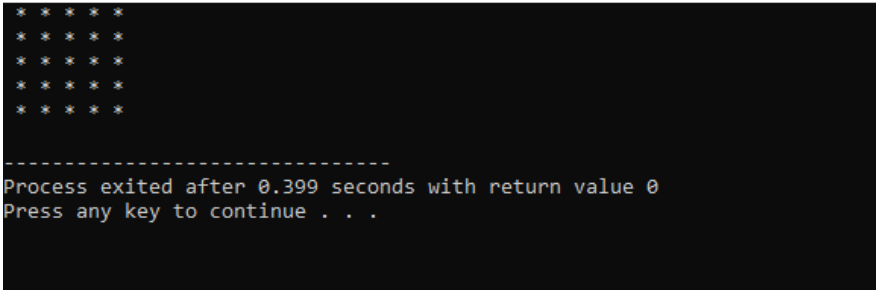
Program C++ dari contoh soal diatas ditulis dibawah ini. Nama program **nestedloop-while.cpp**
 Perhatikan pada C++ **nestedloop-while.cpp** dibawah hasil terjemahan dari *flowchart* gambar

```

nestedloop-while.cpp  Untitled2
1  #include<iostream> //nestedloop-while.cpp
2  using namespace std;
3
4  int BINTANGPOLA5() {
5      int B=1;
6      while ( B <= 5 ){
7          int K = 1;
8          while ( K <= 5 ){
9              printf("*"); //cout<<"*";
10             K++;
11         }
12         printf("\n"); //cout<<endl
13         B++;
14     }
15     return 0;
16 }
17 main(){
18     BINTANGPOLA5();
19 }

```

Output



The screenshot shows a terminal window with the title "D:\4. MODUL 2021\ALGORITHMA\C++ 2022\nestedloop-while.exe". The output displays a 5x5 grid of asterisks (*) followed by a horizontal dashed line. Below the line, it says "Process exited after 0.399 seconds with return value 0" and "Press any key to continue . . .".

Gambar 40 Output Cetak * Pola 5 Baris 5 Kolom

Gambar 39, menampilkan pola 'bintang' 5 baris dan 5 kolom yang telah dicetak pada program nestedloop-while.cpp.

4. Percobaan

Mahasiswa diminta melakukan percobaan soal-soal latihan dibawah ini, tulis program *flowchart* dan program C++ !

Buatlah program flowchart atau pseudocode dan C++ dari soal looping dibawah ini !

1. Cetaklah teks STMIK Indonesia sebanyak 20 kali seperti output ini !

OUTPUT

```
STMIK Indonesia
STMIK Indonesia
STMIK Indonesia
...
```

2. Cetaklah angka 1 sampai dengan 5 seperti output dibawah ini perhatikan output disebelah kanan !

OUTPUT

```
1 2 3 4 5
```

3. Cetaklah semua bilangan genap dari 1 sampai dengan 10 perhatikan output disebelah kanan !

OUTPUT

```
2 4 6 8 10
```

4. Buatlah program flowchart perkalian 2 mulai dari 1 sampai dengan 10, perhatikan output disebelah kanan !

OUTPUT

```
1 x 2 = 2
2 x 2 = 4
3 x 2 = 6
..
..
10 x 2 = 20
```

5. Untuk menampilkan bilangan prima mulai dari 2 sampai dengan 20. Dalam matematika, **bilangan prima** adalah **bilangan asli** yang lebih besar dari 1, yang faktor **pembaginya** adalah 1 dan bilangan itu sendiri. 2 dan 3 adalah bilangan prima. 4 bukan bilangan prima karena 4 bisa dibagi 2. Sepuluh bilangan prima yang pertama adalah 2, 3, 5, 7, 11, 13, 17, 19, 23 dan 29

Bilangan Prima

```
2 3 5 7 11 13 17 19
```

6. **Nested loop.** Buatlah **program PHP** untuk mencetak pola seperti gambar berikut ini, lima baris dan lima kolom: Buat program mencetak pola 5 baris dan 5 kolom seperti berikut ini

a. 5 baris dan 5 kolom	b.	c.	d.	e.
*****	*****	*	*****	* *
*	*****	**	*	* *
*****	***	***	*	*****
*	**	*****	*	* *
*****	*	*****	*	* *

7. **Nested loop** ingin mengulang setiap hari dalam seminggu selama 3 minggu. Misalkan nama hari itu tersebut yaitu hari ke: 1,2,3,4,5,6,7

- Tampilkan semua hari dalam seminggu dalam 3 minggu !
- Tampilkan hari hanya genap. Gunaan *continue* !
- Tampilkan hari hanya minggu ke 1 dan 3. Gunakan *break* !

5. Laporan Praktikum

Kumpulkan hasil percobaan di atas, tambahkan dalam laporan *flowchart* dan program C++ dari setiap percobaan tsb !

Beri Penjelasan dan kesimpulan singkat !

MODUL 8 ARRAY

1. Tujuan

- Memahami Array dimensi 1
- Mampu menulis, mengakses dan menggunakan program C++ Array dimensi 1

2. Dasar Teori

Array Dimensi 1

Suatu array adalah sebuah struktur data yang menyimpan satu atau lebih nilai yang memiliki type data yang sama didalam satu variabel tunggal. Untuk membedakan elemen data antara satu nilai dan nilai yang lain dalam sebuah array diberi indeks. Indeks harus bilangan bulat, angka indeks dimulai dari 0. Pemakaian array erat kaitannya dengan memory komputer, karena setiap pendeklarasian sebuah variabel array, *memory* komputer anda dikapling untuk keperluan pemakaian memory tersebut, semakin besar element yang dideklarasikan pada sebuah variabel array semakin banyak memory komputer anda yang terpakai.

Array digunakan untuk menyimpan beberapa nilai dalam satu variabel, alih-alih mendeklarasikan variabel terpisah untuk setiap nilai.

Syntax

```
typedata var[jumlah elemen data];
```

Sekarang dideklarasikan variabel yang menampung array empat string. Untuk memasukkan nilai ke dalamnya, untuk itu dapat menggunakan literal array - tempatkan nilai dalam daftar yang dipisahkan koma, di dalam kurung kurawal:

Contoh-1

```
string buahTropis[4] = {"Durian", "Mangga", "Salak", "Pepaya"};
```

	0	1	2	3
buahTropis	Durian	Mangga	Salak	Pepaya

Dari gambar array dimensi 1 diatas, dapat dijelaskan sebagai berikut

- **buahTropis** disebut nama variable array
- **durian, manga, salak, pepaya** disebut elemen data
- **0, 1, 3, 4** disebut nomor indeks, lokasi alamat elemen data dalam array buahTropis. Nomor selalu dimulai dari angka 0.

Bagaimana mendapatkan data durian dari array buahTropis ?

Sebutkan nama var, ikuti notasi [, ikuti inkdeks, tutup dengan tanda]

Menjadi sebagai berikut

buahTropis[2] → Salak

buahTropis[0] → Durian

contoh lain, untuk mendeklarasikan array 5 bilangan bulat atau desimal, misal ada nilai matakuliah Algoritma dan Pemrograman untuk 5 orang mahasiswa dapat ditulis dalam sebuah array sbb:

```
int nilai[5] = { 100, 88, 71, 90, 55};
```

Ada perbedaan pendeklarasian untuk data type string (teks) dan int (bilangan bulat) yaitu pendeklarasian untuk type data string, data diapit menggunakan tanda kutip ganda, lihat data “Durian” dari array **buahTropis**. Sedangkan pendeklarasian type data int atau float untuk type data *numeric*, data langsung di tulis tanpa tanda kutip, perhatikan data 100 dan array **nilai**.

3. Tugas Pendahuluan

Contoh-1

Cetak data 5 nama buah tropis

Data buah: **Durian, Mangga, Salak, Pepaya**

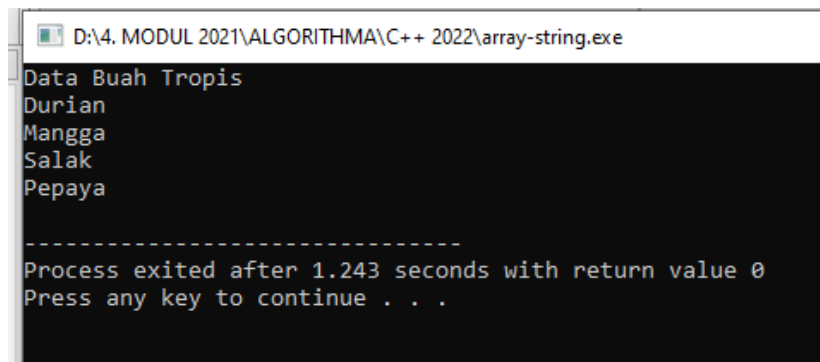
Program C++ dari contoh soal diatas ditulis dibawah ini. Nama program array-string.cpp

```

array-string.cpp array1.cpp
1  #include<iostream> // array-string.cpp
2  using namespace std;
3
4  int CETAKNILAIALGO() {
5
6      //set var array buahTropis dgn 4 elemen data
7      string buahTropis[5]={"Durian", "Mangga", "Salak", "Pepaya"};
8
9      // baca data array buahTropis satu per satu memakai loop while
10     // cetak data buahTropis
11     printf("Data Buah Tropis \n");
12     for (int i=0; i < 4; i++) {
13         cout<< buahTropis[i]<<endl; // cetak data buahTropis yang ke i
14         //var i adalah index, alamat data disimpan
15         // index dimulai dari angka 0
16     }
17     return 0; // alur logika kembali ke baris 21
18 }
19
20 main() {
21     CETAKNILAIALGO(); // panggil function baris 4
22 }
23

```

Output



```

D:\4. MODUL 2021\ALGORITHMMA\C++ 2022\array-string.exe
Data Buah Tropis
Durian
Mangga
Salak
Pepaya

-----
Process exited after 1.243 seconds with return value 0
Press any key to continue . . .

```

Gambar 41 Array String Menetak Nama Buah Tropis

Gambar 41, menampilkan data nama buah tropis: Durian, Manggan, Salak dan Pepaya yang telah dicetak pada program array-string.cpp.

Contoh-2

Data nilai hasil tes matakuliah Algoritma dan Pemrograman untuk sebanyak 5 orang mahasiswa dapat ditulis dalam sebuah array !

Data nilai :100, 88, 71, 90, 55

nilaiAlgoritma = 100, 88, 71, 90, 55

	0	1	2	3	4
nilaiAlgoritma	100	88	71	90	55

Cara mencetak nilaiAlgoritma satu persatu yaitu tulis nama var array ikuti: kurung siku buka, nomor index dan tutup kurung siku tutup. Angka 0,1,2,3 dan 4 disebut *index*. Angka 100, 88, 71, 90 dan 55 disebut elemen data dari array nilaiAlgoritma, perhatikan notasi dibawah ini !

nilaiAlgoritma[0]; → 100

nilaiAlgoritma[1]; → 88

nilaiAlgoritma[2]; → 71

nilaiAlgoritma[3]; → 90

nilaiAlgoritma[4]; → 55

nilaiAlgoritma[0]; → data yang ditampilkan var **nilaiAlgoritma** yang ke 0 adalah 100.

Program C++ dari contoh soal diatas ditulis dibawah ini. Nama program array-1.cpp

```

array1.cpp
1  #include<iostream> // array-1.cpp
2  using namespace std;
3
4  int CETAKNILAIALGO() {
5
6      //set var aaray nilai Algoritma dgn 5 elemen data
7      int nilaiAlgoritma[5]={100, 88, 71, 90, 55};
8
9      // baca data array nilai satu per satu memakai loop for
10     // cetak data nilai
11     printf("Data Nilai Algoritma \n");
12     for (int K=0; K < 5; K++) {
13         printf("%d ", nilaiAlgoritma[K]); // cetak nilai yang ke K
14         //var K adalah indeks, alamat data disimpan
15     }
16     return 0; // alur logika kembali ke baris 20
17 }
18
19 main() {
20     CETAKNILAIALGO(); // panggil function baris 4
21 }

```

Output

```

Data Nilai Algoritma
100 88 71 90 55
-----
Process exited after 0.09288 seconds with return value 0
Press any key to continue . . .

```

Gambar 42 Array Numeric Mencetak Nilai Algoritma

Gambar 42, menampilkan data nilai matakuliah Algoritma: 100, 88, 71, 90, 55 yang telah dicetak pada program array-1.cpp.

4. Percobaan

Mahasiswa diminta melakukan percobaan soal-soal latihan dibawah ini, tulis program program C++ !

Buatlah program C++ dari soal dibawah ini !

OUTPUT

1. Cetaklah nama-nama hari dalam seminggu output ini !

Senin
Selasa
Rabu
Kamis
Jumat
Sabtu
Minggu

2. Tampilkan nama-nama mahasiswa di jurusan Sistem Informasi, lihat output yang diinginkan

Sistem Informasi

1. Rahman
2. Mutiara
3. Qatrunnada
4. Joni Indo
5. Farhan

3. Cetak Kartu rencana studi (KRS) anda seperti gambar dibawah ini. Dari soal dibawah ini kenali variable array dan bukan array, agar bisa didekalarasikan dengan tepat. Ciri-ciri variable array yaitu variabel yang **memiliki nilai data lebih dari satu nilai**, misal kode, matakuliah dan sks, perhatikan variabel kode **memiliki tiga nilai data**: IF010, IF009, dan IF0001. Sedangkan var NIM, Nama dan No hanya memiliki nilai data masing-masing satu nilai data yaitu NIM dengan data 150040 dan Nama dengan data Reski. Khusus variabel No itu adalah pencacah, gunakan variabel dalam loop atau var index untuk dijadikan nomor urut dalam tabel.
Variabel Total SKS dapat dihitung dengan menjumlahkan SKS setiap matakuliah.

KARTU RENCANA STUDI

NIM: 150040 Nama: Reski

NO.	Kode	Matakuliah	SKS
1	IF010	Pemrog Java	3
2.	IF009	Basis Data	3
3.	IF001	Algoritma	4

Total SKS ... ?

Flowchart KRS

4. Buat **program C++** untuk mencetak Kartu Hasil Studi (KHS). Sebelum mengerjakan soal ini, pastikan anda mengenali dengan tepat **variable array** dan bukan variable array!

KARTU HASIL STUDI

NIM: 150040

Nama: Reski

NO.	Kode	Matakuliah	SKS	NH	NM	BOBOT
1	IF010	Pemrog Java	3	A	4	12
2.	IF009	Basis Data	2	C	2	4
3.	IF001	Algoritma	4	B	3	12

Total 9 28
 Indeks Prestasi = 3,11

Aturan

konversi nilai angka (NA) ke nilai huruf (NH), sbb
 jika na 85 s.d 100 maka NH adalah A nilai mutu (NM) 4
 jika na 75 s.d 84 maka NH adalah B nilai mutu (NM) 3
 jika na 65 s.d 74 maka NH adalah C nilai mutu (NM) 2
 jika na 55 s.d 64 maka NH adalah D nilai mutu (NM) 1
 jika na 0 s.d 54 maka NH adalah E nilai mutu (NM) 0

Bobot = sks x NM

Totsks = totsks + sks

TotBobot = totBobot + Bobot

IP = TotBobot / Totsks

5. Laporan Praktikum

Kumpulkan hasil percobaan di atas, tambahkan dalam laporan *flowchart* dan program C++ dari setiap percobaan tsb !

Beri Penjelasan dan kesimpulan singkat !

MODUL 9 SORTING

1. Tujuan

1. Mampu memahami metode mengurut data acak menjadi urut
2. Mampu memahami penggunaan metode *sort*
3. Mampu memahami manfaat metode sorting dalam pengelolaan data
4. Mampu menuliskan salah satu metode *sort* dalam bahasa pemrograman C++

2. Dasar Teori

Sorting adalah proses pengurutan terhadap sekumpulan data yang tersusun secara acak. Data tersebut telah berada dalam memori *computer*, dibaca dan dipindahkan kedalam *array*, kemudian di *sorting* dengan dua cara yaitu dari kecil ke besar disebut ***ascending*** atau dari besar ke kecil disebut ***descending***. Data dapat berupa numeric (angka) seperti: 50, 10, 40, dan 60 atau string (kata) seperti: senin, minggu, jumat dan kamis. Notasi untuk data numeric dan string berbeda yaitu notasi data numeric tidak menggunakan tanda kutip sedangkan notasi data string harus menggunakan tanda kutip.

Sebuah variable array selalu memiliki *index* digunakan untuk membedakan antara satu variable dengan variable yang lain. Sebuah variable *index* memiliki lebih dari satu element data, sedangkan nama dari element data tersebut mempunyai nama yang sama. Index dalam sebuah array selalu integer (bilangan bulat) yang dimulai dari angka nol.

Salah satu metoda yang dapat dipakai untuk melakukan kegiatan *sorting* adalah ***bubble sort*** seperti gelembung air dalam tangki, mengapung keatas karena ringan. Analogi dengan mengurutkan angka dari besar ke kecil dimana ada proses meletakkan bilangan terkecil paling ujung dari sederatan bilangan, diikuti bilangan kedua terkecil, ketiga dan seterusnya, dengan cara membanding bilangan satu persatu mulai dari bilangan yang berada pada urutan pertama dalam array hingga urutan yang terakhir.

Kenapa *sorting* diperlukan dalam *programming* ? karena sangat membantu dalam pengolahan data, membantu mempercepat dalam pencarian data. Bayangkan sebuah kamus Inggris-Indonesia, didalamnya terdapat ribuan bahkan jutaan kosa-kata, dicetak secara acak. Berapa lama waktu yang diperlukan untuk mendapatkan kosa-kata '**sort**' pada kamus tersebut ?

Bandingkan betapa mudahnya mendapatkan kosa-kata '**sort**' pada kamus yang telah disorting secara *ascending*, a sampai z. Itulah salah satu manfaat menggunakan aplikasi yang ditemukan dalam kehidupan sehari-hari. Contoh lain pemanfaatan dari pemakaian *sorting* adalah penempatan nomor telepon dalam buku telepon, katalog buku di perpustakaan, jadwal penerbangan di bandara, nomor contact person pada HP saudara, dll.

Misalkan nama array yang digunakan yaitu bilangan disingkat B,

Misalkan jumlah bilangan ada 4 maka, $N=4$

Indeks $j = 0..3$

Sedangkan variable Temp (*temporary*) adalah variable penampung data sementara ketika terjadi pertukaran tempat data sewaktu proses *sorting* berlangsung.

Ada beberapa metode sort yang dikenali dalam bahasa pemrograman C++ yaitu

1. Bubble sort
2. Selection sort
3. Insertion sort
4. Quick sort

3. Tugas Pendahuluan

Bubble Sort

Diketahui data acak : 50, 10, 40, 60 ingin data diurutkan *ascending*!

Algoritma dari metoda *bubble sort* ini sebagai berikut:

1. Input data
 - a. Deklarasikan sebuah bilangan array misal B
 - b. Tentukan jumlah bilangan dalam array misal $N=4$
 - c. Tentukan index yang digunakan oleh array B misal j

- d. Deklarasikan sebuah variable penampung bilangan sementara misal *temporary* (Temp)
 - e. Buat looping sebanyak N untuk memasukkan semua bilangan satu per satu kedalam array B
2. Sorting Bubble Sort, metode lain *quick sort*, *insertion sort*, *selection sort*. Pada contoh ini menggunakan metode ***bubble sort***, mengurut data *ascending*.

Data acak: 50, 10, 40, 60 jumlah 4 , N= 4

- a. Buat *looping* sebanyak N-1
- b. For j = 1 to N-1 → for(int j=0; j < N-1; j++)
- c. Badingkan data satu persatu, jika Ya lakukan swap (pertukaran tempat data)

50, 10, 40, dan 60 b ← **ascending 10 40 50 60**

J= 0 1 2 3

- d. if B(j) > B(j+1) {
 Temp = B(0) **50**
 B(0) = 10
 B(j) = B(j+1) **10**
 B(j+1)=Temp
 B(1) = 50
 }

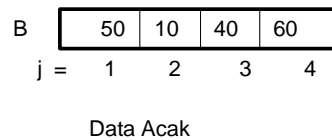
10 50 40 60

- e. jika tidak abaikan

3. Printing

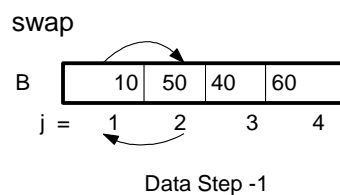
- a. Buat looping sebanyak N
- b. Ulangi langkah 2 sampai semua bilangan telah terurut

Berapa kali proses pengurutan dilakukan ? sebanyak N-1. Perhatikan gambar 43. Pada fase awal data masih acak



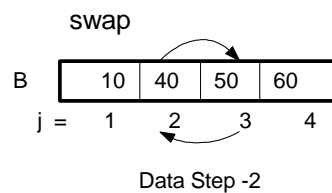
Gambar 43 Data Acak

Kemudian dilakukan sorting, dibandingkan data indeks ke 1 dengan data indeks ke 2, maka pada step-1 terjadi swap-perpindahan tempat data karena data indeks ke 1 lebih besar dari data indeks ke 2. Data pada indeks ke 2 bertukar tempat ke posisi data indeks ke 1 begitu sebaliknya, lihat gambar 44.



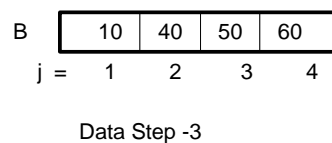
Gambar 44 Terjadi Data Swap Indeks 2 dan indeks 1

Proses berikutnya terjadi looping hingga N-1, akibatnya proses pada Step-1 berulang kejadiannya sampai semua data urut dari kecil sampai besar, lihat gambar 45.



Gambar 45 Terjadi Data Swap Indeks 3 dan indeks 2

Pada Step-3 tidak terjadi data swap disebabkan data telah urut pada Step-2, tetapi proses sorting tetap dilakukan, lihat gambar 46.



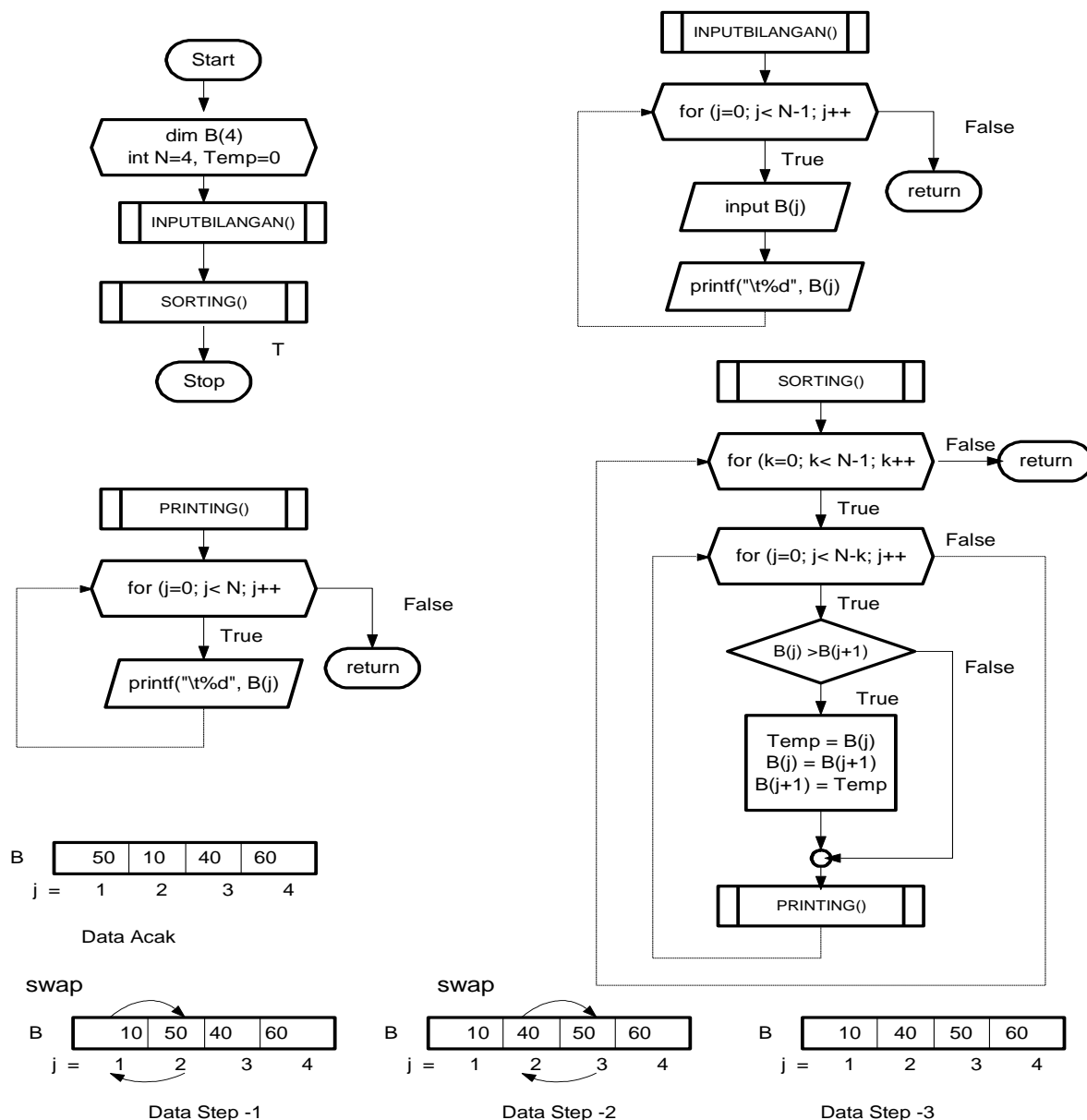
Gambar 46 Tidak Terjadi Data Swap

Program flowchart bubble sort dapat dilihat pada gambar 47. Silahkan konversikan ke dalam bahasa pemrograman C++ untuk menguji logika sorting, amati dan beri kesimpulan ! Flowchart gambar 47 terdiri dari flowchart:

1. Flowchart utama, perhatikan lamabang **start** hingga **stop**
2. Flowchart INPUTBILANGAN → program untuk memasukkan data
3. Flowchart SORTING → proses mengurut data metode bubble sort, ascending
4. Flowchart PRINTING → proses mencetak fase demi fase setiap proses pengururtan data

Flowchart Bubble Sort

Program Flowchart
Ascending Bubble Sort
mengurut data dari kecil ke besar



Gambar 47 Program Flowchart Ascending menggunakan metode Bubble Sort

Dari keadaan terburuk maka proses sorting terhadap sejumlah data acak adalah sebanyak $N-1$ kali. Semakin banyak jumlah data yang akan disorting maka waktu yang digunakan untuk mengurut data akan semakin lama.

Keburukan metoda ini adalah bila pada Step kedua yang terakhir data telah terurut tetapi looping masih tersisa sekali lagi maka proses looping, usaha pengurutan data tetap dilaksanakan.

Keunggulan metoda Bubble Sort ialah mudah dipahami.

Coding C++ Bubble Sort

Program sort ini diberinama bubble_sort_ver2.cpp

```
liniersearch.cpp bubble_sort_ver2.cpp
1  #include <iostream> // bubble_sort_ver2.cpp
2  using namespace std; // 18 mei 2022
3
4  int printing(int B[], int N) {
5
6      for (int a=0; a<N ; a++){
7          cout<<"\t"<<B[a];
8      }
9      cout<<"\n";
10     return 0;
11 }
12 void bubblesort(int B[], int N){
13     int Temp =0;
14
15     for (int k=0; k < (N-1); k++){
16         for (int j=0; j < (N-k-1); j++ ) {
17             // badingkan nilai
18             if (B[j] > B[j+1]) {
19                 // swap=pertukaran tempat
20                 Temp = B[j];
21                 B[j] = B[j+1];
22                 B[j+1] = Temp;
23             }
24             cout<<"Fase " <<k <<": ";
25             printing(B, N); // cetak data setelah di sort
26
27         }
28     }
29 }
```

```

30 void INPUTBILANGAN(int B[], int N){
31     // input data
32     int ke=1;
33
34     cout <<"BUBBLE SORT \nAscending \nJumlah Data ?";
35     cin>>N; // input data dari keyboard.
36     // berapa jumlah data yang akan di sortir mis 5
37     //cout=print=cetak teks atau data
38     // loop untuk input sebanyak 5x --> 2,80,120,15,22
39     for (int i=0; i<N; i++){
40         cout <<"Data ke--> "<<ke<<"? : ";
41         //scanf("%d", &B[i]); versi C
42         cin>> B[i]; // versi C++ : 2,80,120,15,22
43         ke++;
44     }
45
46 }
47
48 main(){
49     int B[70]={}; // jumlah element data = 70
50     // menghitung jumlah elemen data dlm array B
51     int N = sizeof(B)/ sizeof (int);
52     INPUTBILANGAN(B, N);
53     bubblesort(B, N);
54
55     //sort = urut
56     //short = pendek
57     // cari metode sort selai Bubble Sort,
58     // program dan Pahami konsep sort

```

4. Percobaan

Latihan buat program flowchart dan C programming. Pilih salah satu metode sort: selection sort, quick sort atau insertion sort

1. Sorting data berat badan dari 10 mahasiswa ascending dengan data sebagai berikut:
49,81,48,77,57,63,68,77,55.
2. Sorting data nama-nama mahasiswa descending untuk 6 orang: Siraj, Syfa, Rahman, Gopnar, Rudi, dan Tandri

5. Laporan Praktikum

MODUL 10 SEARCHING

1. Tujuan

1. Mampu memahami metode serarch
2. Mampu menulis coding metode searching yaitu *sequence search* dan *binary search*

2. Dasar Teori

Sequence Search

Proses menemukan data yang diinginkan dalam array disebut *searching*. *Searching* sangat berguna ketika menggunakan sebuah array dalam ukuran yang besar.

File adalah kumpulan record-record yang sejenis, *record* merupakan kumpulan dari field-field, field adalah kumpulan character. Character adalah symbol, seperti: A, g, *, ~ (*tilde*), 4,dsb. Satu character sama dengan 8 bit. Jumlah semua simbol yang ada pada *computer* ada sebanyak 256 simbol misal: A..Z, a..z, 0..9, @, #, %, ^, &, , dst dapat dilihat pada daftar [ASCII CODE](#)

Algorithma searching dalam sebuah *Arrray*

1. Setting Jumlah data: N misal jumlah ada 5
2. Setting variable Array. Misal tb[5]
3. isi data (misal tinggi badan) ke dalam Array. Missal : 150,142,171,154,132 cm
4. Setting variable Found bernilai F (menandakan kondisi pertama pencarian belum berhasil, *false*).
5. input data yang akan dicari misal variable caritb, misal 171
6. buat sebuah looping For sebanyak N
7. bandingkan apakah data yang dicari sama dengan data yang tersedia
8. jika ketemu
 - a. setting Found bernilai T (artinya pencarian data berhasil, *true*)
 - b. tampilkan data
 - c. *break* (hentikan looping karena pencarian data sudah berhasil)

9. jika tidak ketemu, teruskan pencarian sampai semua data diperiksa satu persatu dan tampilkan pesan data 'Tidak Ketemu'
10. if Found='T'
 - a. cetak: Ketemu, dan data yang dicari
11. if found ='F'
 - a. cetak: Data tidak ketemu
12. selesai

Binary Search

Anda ingat permainan tebak angka ? dimana respon yang muncul “saya kira angka berada diantara 1 hingga 100” hasilnya “terlalu tinggi”, terlalu rendah”, atau anda tepat menemukannya secara tepat. Strategi yang sering digunakan ketika bermain *game* ini adalah membagi interval diantara angka yang ditebak dan akhir dari *range* menjadi dua bagian. Strategi ini membantu anda membatasi secara cepat bagian angka yang diinginkan.

Ketika pencarian dalam sebuah *array*, proses *binary search* diawali dengan konsep yang sama yakni membagi interval menjadi dua bagian sebagai cara menemukan nilai “key” dengan kemungkinan yang tercepat.

Jika data dalam array telah terurut (*ascending* atau *descending*), anda dapat mencari data yang diinginkan secara cepat menggunakan algoritma *binary search* (mengacu ke pendekatan *divide and conquer*)

3. Tugas Pendahuluan

Sequence Search

Contoh:

Diketahui ada 5 element data Tinggi Badan dalam sebuah Array yakni: 150,142,171,154,132 cm. Lakukan *searching* untuk data 171 !. Buatlah C/C++ Programming !

Perhatikan C Programming dibawah ini !

```

// Author : Ir. M Amrin Lubis, M.Sc
// email : maloebis@gmail.com
// Program: Searching.cpp
// Installation : Lab Komputer STMIK Indonesia, Padang

#include<stdio.h>
#include<conio.h>2

void SEARCHING();

int N=5; // n adalah jumlah data

main(){

    SEARCHING();
    getch();

    return 0;
}

void SEARCHING(){
    int tb[5]={ 150,142,171,154,132};
// index    0  1  2  3  4
    int caritb;

    char found='F';

    printf("\n INPUT TINGGI BADAN = ");
    scanf("%d", &caritb); // 171, 168
    printf("\n TINGGI BADAN YANG DICARI ADALAH ");

    for (int k=0; k<N; k++) { //k=0
        printf("\n data -> ke %d = %d",k, tb[k]);

        if (caritb == tb[k]) {
            found='T';
            printf("\tKetemu");
            break;
        }
    }
    if (found=='T' )
        printf("\t%d" , tb[k] );
    else
        printf("\tTinggi Badan %d Tidak Ketemu", caritb);

```

```
}
```

Penjelasan: uji program dengan mencari data 171, kemudian 180

1. Pertama programmer menjalankan (run) program, anda diminta memasukkan data Tinggi Badan yang akan dicari, misal 171, enter
2. Komputer memprosesnya,
 - a. Disetting logika Found ='F'
 - b. Dilakukan looping sebanyak N data
 - c. Membandingkan data 171 dengan data dalam Array tb[]
3. jika data yang diinput sama dengan data dalam Array tb
 - a. maka logika found di set Found='T',
 - b. kemudian looping dihentikan dengan logika break
4. jika data yang dicari tidak sama
 - a. setingan logika Found masih kondisi semula
5. akhiri looping
6. jika Found ='T' artinya pencarian berhasil kemudian ditampilkan
"Ketemu 171"
7. jika Found ='F' artinya pencarian tidak berhasil kemudian ditampilkan
Tinggi Badan 180 Tidak Ketemu
8. selesai

Binary Search

Perhatikan bilangan interger dalam array dibawah ini:

Array of integers, data yang telah berurut ada dalam array sebagai berikut

10	15	24	36	45	55	64	73	90	98
num[0]	num[1]	num[2]	num[3]	num[4]	num[5]	num[6]	num[7]	num[8]	num[9]

Kita akan mencari key untuk angka 64. Disini dijelaskan bagaimana metoda binary search bekerja: Mula-mula diambil dari posisi awal=1 dan posisi akhir = n

1. Kemudian kita cari posisi data tengah dengan rumus posisi tengah = (posisi awal + posisi akhir) div 2
2. Kemudian data yang di cari dibandingkan dengan data tengah
 - a. Jika sama, data ditemukan, Proses selesai
 - b. Jika lebih kecil, proses dilakukan kembali tetapi posisi akhir dianggap sama dengan posisi tengah -1
 - c. Jika lebih besar , proses dilakukan kembali tetapi posisi awal dianggap sama dengan posisi tengah +1
3. Ulangi langkah kedua hingga data ditemukan , atau tidak ditemukan
4. Pencarian biner ini akan berakhir jika data ditemukan posisi awal lebih besar dari pada posisi akhir. Jika posisi awal sudah lebih besar dari posisis akhir berarti data tidak diketemukan.

Mari perhatikan penjeasan berikut

Pertama, bagian tengah larik ditemukan dengan menambahkan subskrip larik dari nilai pertama ke subskrip dari nilai terakhir dan membaginya dengan dua: $(0 + 9) / 2 = 4$ Pembagian bilangan bulat digunakan untuk sampai pada bilangan ke-4 subscript sebagai tengah. (Tengah matematika yang sebenarnya akan berada di antara subskrip 4 dan 5, tetapi kita harus bekerja dengan subskrip bilangan bulat.)

Subskrip 4 memuat angka 45, yang muncul sebelum 64. Sekarang kita tahu bahwa 64 akan ada di bagian larik di sebelah kanan 45. Sekarang kita cari bagian tengah larik kanan dengan menggunakan pendekatan yang sama: $(5 + 9) / 2 = 7$

Subscript 7 memegang angka 73, yang muncul setelah 64, jadi sekarang kita perlu mencari bagian tengah dari array di sebelah kanan 45, tetapi di sebelah kiri 73: $(5 + 6) / 2 = 5$

Subskrip 5 memegang angka 55, yang datang sebelum 64, jadi sekarang kita bagi lagi $(6 + 6) / 2 = 6$ dan elemen 6 memegang angka 64.

4. Percobaan

Buat flowchart untuk searching menggunakan metode random searching, bisa digunakan data pada bagian Tugas Pendahuluan sebagai berikut :

Diketahui ada 5 element data Tinggi Badan dalam sebuah Array yakni: 150,142,171,154,132 cm. Lakukan *searching* untuk data 171 !. Buatlah C/C++ Programming !

5. Laporan Praktikum

Referensi

1. C++ Geek for Geek <https://www.geeksforgeeks.org>
2. C++ https://icarus.cs.weber.edu/~dab/cs1410/textbook/1.Basics/compiler_op.html
3. Draw io Flowchart <https://app.diagrams.net/>
4. C++ W3Schools <https://www.w3schools.com>
5. Tutorials Point https://www.tutorialspoint.com/cplusplus/cpp_data_types.htm
6. C++ <https://www.dcs.bbk.ac.uk/~roger/cpp/week4.htm>
7. Tutorial Kart <https://www.tutorialkart.com/cpp/cpp-do-while-loop/>
8. Programiz <https://www.programiz.com/cpp-programming/switch-case>
9. Hub Pages (<http://royal52.hubpages.com/hub/Binary-Search-in-C>)