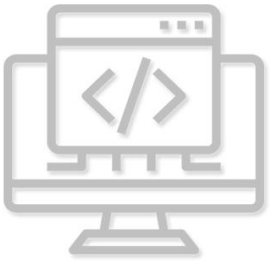
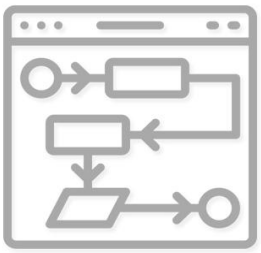
Mengenal Bahasa C

Algoritma Pemrograman





Susanti

Susanti - Mengenal Bahasa C

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa atas segala rahmat dan nikmat-Nya sehingga penulis dapat menyelesaikan modul Algoritma Pemrograman Bahasa C ini.

Algoritma Pemrograman merupakan matakuliah yang mengajarkan tentang bagaimana suatu masalah dapat diselesaikan dengan cara berdasarkan atas tahapan-tahapan yang sistematis dan logis pada sebuah komputer. Suatu algoritma dapat dibuktikan kebenarannya dengan computer setelah algoritma tersebut diubah/ ditranslasikan ke dalam bahasa pemrograman. Bahasa pemrograman yang digunakan dalam modul ini yaitu bahasa pemrograman C.

Modul ini membahas mengenai pembahasan pada bahasa pemrograman C sebagai panduan bagi mahasiswa dalam mempelajari algoritma pemrograman bahasa C. adapun kenapa memilih bahasa C yaitu karena bahasa C adalah bahasa pemrograman yang fleksible, bersifat moduler dan fungsi-fungsi yang telah kita buat, bisa kita gunakan kembali dalam program ataupun aplikasi lain. Selain itu, pengaruh positif lain adalah semakin banyaknya kompiler yang dikembangkan untuk berbagai platform (berpengaruh ke portabilitas).

Akhir kata, penulis mengucapkan syukur Alhamdulillah modul ini dapat diselesaikan dengan baik dan terimakasih kepada semua pihak yang turut membantu dalam kelancaran pembuatan modul ini, walaupun masih banyak kekurangan disana-sini yang perlu perbaikan dimasa yang akan datang. Oleh karena itu penulis menerima masukan dan saran dari rekan-rekan dosen pengajar matakuliah Algoritma Pemrograman maupun masukan dari mahasiswa yang menggunakan modul ini. Mudah-mudahan modul ini dapat bermanfaat baik bagi penulis sendiri maupun bagi pengguna lainnya.

Tanjungpinang, Mei 2020

PENULIS

DAFTAR ISI

BAB 1: ALGORITMA PEMROGRAMAN BAHASA C

A. KONSEP ALGORITMA BAHASA C

Algoritma adalah metode efektif yang diekspresikan sebagai rangkaian terbatas. Algoritma juga merupakan kumpulan perintah untuk menyelesaikan suatu masalah. Masalah yang akan dipecahkan perlu diperjelas misalnya mengenai lingkup dan asumsi. Masalah ini dipecahkan dengan berbantuan computer berupa algoritma yang dipecahkan menjadi program computer (*source code*), kemudian diubah menjadi *executable code* yang dapat dijalankan (*run*). Keluaran hasil *run* merupakan jawaban atas masalah.

Manfaat belajar pemrograman bahasa C antara lain:

- *C adalah bahasa pemrograman yang fleksibel*
  Dengan menguasai bahasa C, kita bisa menulis dan mengembangkan berbagai jenis program mulai dari operating system, word processor, graphic processor, spreadsheets, ataupun kompiler untuk suatu bahasa pemrograman.
- *C adalah bahasa pemrograman yang bersifat moduler*
  Program C ditulis dalam routine yang biasa dipanggil dengan fungsi. Fungsi-fungsi yang telah kita buat, bisa kita gunakan kembali (reuse) dalam program ataupun aplikasi lain.
- *C adalah bahasa pemrograman yang paling populer saat ini*
  Dengan banyaknya programmer bahasa C, membawa pengaruh semakin mudahnya kita menemukan pemecahan masalah yang kita dapatkan ketika menulis program dalam bahasa C. Pengaruh positif lain adalah semakin banyaknya kompiler yang dikembangkan untuk berbagai platform (berpengaruh ke portabilitas).

B. ALGORITMA

1. DEFINISI ALGORITMA

*"Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis".*Kata *Logis* merupakan kata kunci dalam Algoritma. Langkah-langkah dalam Algoritma harus logis dan harus dapat ditentukan bernilai salah atau benar.

2. ALGORITMA MERUPAKAN JANTUNG ILMU INFORMATIKA

Algoritma adalah jantung ilmu komputer atau informatika. Banyak cabang ilmu komputer yang diacu dalam terminologi algoritma. Namun, jangan beranggapan algoritma selalu identik dengan ilmu komputer saja. Dalam kehidupan sehari-haripun banyak terdapat proses yang dinyatakan dalam suatu algoritma. Cara-cara membuat kue atau masakan yang dinyatakan dalam suatu resep juga dapat disebut sebagai algoritma. Pada setiap resep selalu ada urutan langkah-lankah membuat masakan. Bila

langkah-langkahnya tidak logis, tidak dapat dihasilkan masakan yang diinginkan. Secara umum, pihak (benda) yang mengerjakan proses disebut pemroses (*processor*). Pemroses tersebut dapat berupa manusia, komputer, robot atau alat-alat elektronik lainnya. Pemroses melakukan suatu proses dengan melaksanakan atau "mengeksekusi" algoritma yang menjabarkan proses tersebut.

Melaksanakan Algoritma berarti mengerjakan langkah-langkah di dalam Algoritma tersebut. Pemroses mengerjakan proses sesuai dengan algoritma yang diberikan kepadanya. Juru masak membuat kue berdasarkan resep yang diberikan kepadanya, pianis memainkan lagu berdasarkan papan not balok.Karena itu suatu Algoritma harus dinyatakan dalam bentuk yang dapat dimengerti oleh pemroses. Jadi suatu pemroses harus :

1. Mengerti setiap langkah dalam Algoritma
2. Mengerjakan operasi yang bersesuaian dengan langkah tersebut.

3. STRUKTUR DASAR ALGORITMA

Adapun struktur dasar pada algoritma adalah sebagai berikut:

a. Sekuensial (runtunan)

Pada struktur sekuensial ini langkah-langkah yang dilakukan dalam algoritma diproses secara berurutan. Dimulai dari langkah pertama, kedua, dst. Pada dasarnya suatu program memang menjalankan suatu proses dari yang dasar seperti struktur ini.

b. Struktur seleksi

Struktur seleksi menyatakan pemilihan langkah yang didasarkan oleh suatu kondisi atau pengambilan suatu keputusan.Struktur ini ditandai selalu dengan bentuk flowcart decision (flowcart yang berbentuk belah ketupat).

c. Struktur perulangan

Struktur ini memberikan suatu perintah atau tindakkan yang dilakukan beberapa kali. Misalnya jika teman mau menuliskan kata "belajar c" sebanyak sepuluh kali. Akan lebih efisien jika teman menggunakan sturktur ini dari pada sekedar menuliskannya berturut-turut sebanyak sepuluh kali.

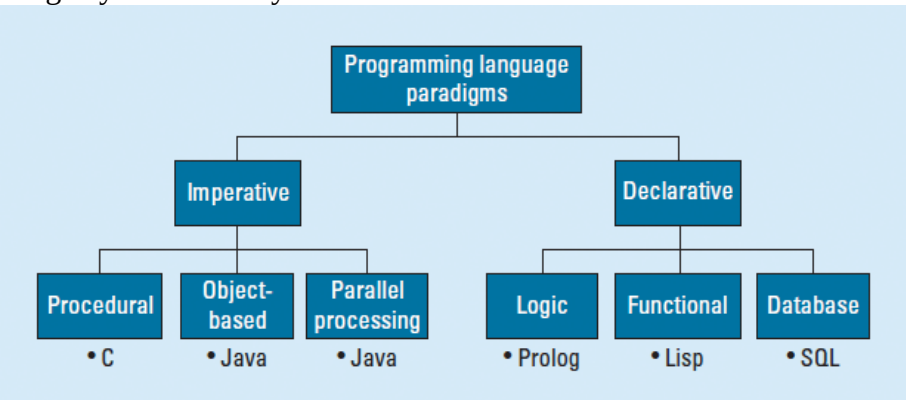
4. MEKANISME PELAKSANAAN ALGORITMA OLEH PEMROSES

Komputer hanyalah salah satu pemroses. Agar dapat dilaksanakan oleh komputer, algoritma harus ditulis dalam notasi bahasa pemrograman sehingga dinamakan program. Jadi program adalah perwujudan atau implementasi teknis Algoritma yang ditulis dalam bahasa pemrogaman tertentu sehingga dapat dilaksanakan oleh komputer.

5. BELAJAR MEMPROGRAM DAN BELAJAR BAHASA PEMOGRAMAN

Belajar memprogram tidak sama dengan belajar bahasa pemrograman. Belajar memprogram adalah belajar tentang metodologi pemecahan masalah, kemudian menuangkannya dalam suatu notasi tertentu yang mudah dibaca dan dipahami. Sedangakan belajar bahasa

pemrograman berarti belajar memakai suatu bahasa aturan-aturan tata bahasanya, instruksi-instruksinya, tata cara pengoperasian *compiler*-nya, dan memanfaatkan instruksi-instruksi tersebut untuk membuat program yang ditulis hanya dalam bahasa itu saja. Sampai saat ini terdapat puluhan bahasa pemrogram.Yang dapat dibedakan berdasarkan tujuan dan fungsinya. Diantaranya adalah :



Gambar 1.1 Perbedaan Bahasa Program Berdasarkan Tujuan dan Fungsinya

Sumber: google, https://www.researchgate.net/figure/A-hierarchy-of-programming-language-paradigms-based-on-Doris-Appleby-and-Julius_fig2_230556215

6. PRODUK YANG DIHASILKAN PEMROGRAM

- Program dengan rancangan yang baik (metodologis, sistematis)
- Dapat dieksekusi oleh mesin
- Berfungsi dengan benar
- Sanggup melayani segala kemungkinan masukan
- Disertai dokumentasi
- Belajar memprogram, titik berat : designer program

7. ALGORITMA

Aksi:

- Kejadian yang terjadi pada selang waktu terbatas (dimulai saat T0 dan berakhir pada saat T1)
- Menghasilkan efek netto yang terdefinisi dengan baik dan direncanakan

  Contoh :

  - Ibu Tati **mengupas kentang** untuk mempersiapkan makan malam. (luas ruang lingkupnya)
  - Karena ruang lingkup luas, maka harus didefinisikan keadaan awal dan efek netto yang direncanakan (Initial State dan Final State)
  - Initial State (keadaan awal): T0 kentang sudah ada dikantong kentang, dan ditaruh di rak dapur dimana ibu Tati akan mengupasnya

- Final State (keadaan akhir): T1 kentang dalam keadaan terkupas di panci, siap untuk dimasak dan kantong kertasnya harus dikembalikan ke rak lagi.
- Kejadian: urut-urutan dari beberapa aksi yang terjadi secara berurutan.
- Efek kumulatif dari semua aksi yang terjadi menjadi efek netto dari kejadian
- Penggolongan suatu kejadian menjadi aksi adalah relatif tergantung darisudut pandang. Contoh mengupas kentang dapat dijelaskan:
    a. Ambil kantong kentang dari rak
    b. Ambil panci dari almari
    c. Kupas kentang
    d. Kembalikan kantong kentang ke rak

  Contoh lain:
    a. Ambil kantong kentang dari rak dan ambil panci dari almari
    b. Kupas kentang
    c. Kembalikan kantong kentang ke rak

  Jika esok hari ibu Tati mengupas kentang lagi untuk makan malam juga, dan kita mengamati hal-hal yang sama, apakah hal tsb bisa disebut sama ?

  Ini tergantung jawabannya bisa sama bisa tidak.

  Tidak karena ibu Tati tidak mungkin mengupas kentang yang sama dengan kemarin **Sama** karena kemiripan pola yang dilakukan.

C. NOTASI ALGORITMA

Algoritma mempunyai aturan penulisan sendiri yang disebut dengan notasi algoritma. Notasi algoritma ini tidak tergantung dari spesifikasi bahasa pemrograman tertentu dan komputer yang mengeksekusinya. Notasi algoritma merupakan bahasa universal yang dapat diterima oleh semua bahasa pemrograman yang ada. Oleh sebab itu algoritma yang baik harus dapat diterjemahkan ke dalam bentuk source code dari semua bahasa pemrograman yang ada. Untuk membuat algoritma dari suatu permasalahan, biasanya digunakan salah satu dari tiga buah notasi algoritma yang dikenal, yaitu uraian kalimat deskriptif, flow chart, atau pseudo code. Sebagai contoh permasalahan, jika diinginkan sebuah program komputer yang dapat mengetahui bilangan terbesar dari tiga buah bilangan yang dimasukkan.Bagaimanakah algoritmanya?

1. Deskriptif Algoritma

Algoritma dengan uraian kalimat deskriptif adalah notasi algoritma yang paling sederhana karena algoritma ini menggunakan bahasa sehari-hari. Untuk permasalahan yang sederhana penggunaan notasi ini sangatlah mudah, akan tetapi untuk permasalahan yang lebih komplek dan rumit, penggunaan notasi ini akan lebih sulit dan sering kali terjadi ambigu

dalam langlah-langkah penyelesaian masalah. Oleh karena itulah untuk kasus-kasus yang lebih komplek, penggunaan notasi ini jarang sekali bahkan tidak digunakan. Permasalahan di atas, yaitu mencari bilangan terbesar dari tiga buah bilangan yang dimasukkan, tergolong permasalahan yang sederhana, jadi algoritmanya masih mudah dan dapat dijelaskan dengan uraian kalimat deskriptif sebagai berikut:

1. Masukkan sembarang bilangan sebanyak tiga buah.
2. Ambil bilangan pertama dan set maksimumnya sama dengan bilangan pertama.
3. Ambil bilangan kedua dan bandingkan dengan maksimum.
4. Apabila bilangan kedua lebih besar dari maksimum maka ubah maksimumnya menjadi sama dengan bilangan kedua.
5. Ambil bilangan ketiga dan bandingan dengan maksimum.
6. Apabila bilangan ketiga lebih besar dari maksimum maka ubah lagi maksimumnya menjadi sama dengan bilangan ketiga.
7. Variabel maksimum akan berisi bilangan yang terbesar dan tampilkan hasilnya.

Algoritma dengan uraian kalimat deskriptif seperti di atas sudah jarang sekali kita temukan karena kadang kala agak sulit untuk memahaminya. Yang paling banyak kita temukan adalah algoritma (dengan uraian kalimat deskriptif) yang ditulis secara lebih sistematis dan efisien sehingga lebih mudah untuk memahaminya. Algoritma tersebut adalah sebagai berikut:

Gambar 1.2 Algoritma Deskriptif

1. Masukkan a, b, dan c.
2. mak ← a.
3. Jika b > mak, kerjakan langkah ke-4. Jika tidak, kerjakan langkah ke-5.
4. mak ← b.
5. Jika c > mak, kerjakan langkah ke-6. Jika tidak, kerjakan langkah ke-7.
6. mak ← c.
7. Tulis mak.

Sumber : Buku Bahasa Pemrograman Lengkap

Dalam notasi algoritma, baik menggunakan notasi algoritma dengan uraian kalimat deskriptif, flow chart maupun pseudo code, kita tidak menggunakan tanda = (sama dengan) tetapi menggunakan simbol anak panah ke arah kiri (<-) seperti yang terlihat pada langkah ke-2, 4, dan 6. Sebagai contoh pada langkah ke-2, arti dari notasi tersebut adalah nilai variabel a (yang ada di sebelah kanan anak panah) diberikan kepada variabel mak (yang ada di sebelah kiri anak panah). Dengan demikian jika nilai variabel a adalah 10, maka nilai mak juga 10 atau dalam penulisan secara matematis mak = a. Penggunaan anak panah ini dikarenakan, seperti yang telah dikemukakan sebelumnya, algoritma tidak diperuntukkan untuk suatu bahasa pemrograman tertentu, tetapi dapat diaplikasikan atau diterjemahkan ke dalam bentuk source code dari semua bahasa pemrograman yang ada. Dalam pascal misalnya, notasi yang digunakan

untuk tanda = (sama dengan) adalah := (titik dua dilanjutkan dengan tanda sama dengan) sehingga langkah ke-2 akan diterjemahkan menjadi mak := a. Akan tetapi dalam bahasa C++ maupun Java, tanda = (sama dengan) masih tetap digunakan sehingga penerjemahannya adalah mak = a.

2. Flow Chart

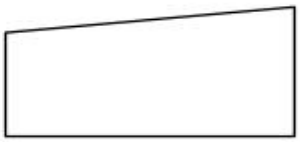
Notasi algoritma yang paling banyak digunakan adalah flow chart karena bentuknya yang sederhana dan mudah dipahami.Flow chart (diagram alir) adalah penggambaran secara grafik dari langkah-langkah pemecahan masalah yang harus diikuti oleh pemroses.Flow chart terdiri atas sekumpulan simbol, dimana masing-masing simbol menggambarkan suatu kegiatan tertentu. Flow chart diawali dengan penerimaan masukan (input), pemrosesan masukan, dan diakhiri dengan menampilkan hasilnya (output). Adapun simbol-simbol yang sering digunakan untuk menyusun flow chart (dalam microsoft visio) adalah sebagai berikut :

1) Masukan
   Masukan merupakan kegiatan penerimaan data yang disimbolkan dengan jajaran genjang. Kita dapat menuliskan masukan yang diperlukan pada suatu waktu secara satu per satu maupun secara keseluruhan, akan tetapi untuk alasan efisiensi ruang gambar biasanya masukan dituliskan bersamaan secara keseluruhan.

Gambar 1.3 Simbol Masukan

Sumber : Buku Bahasa Pemrograman Lengkap

2) Masukan manual
   Untuk masukan secara manual yang dimasukkan melalui keyboard, atau perangkat input lainnya seperti barcode reader, kita dapat menggunakan simbol masukan secara manual. Sama dengan simbol masukan, pada simbol masukan manual ini untuk alasan efisiensi ruang gambar biasanya masukan juga dituliskan bersamaan secara keseluruhan.



Gambar 1.4 Simbol masukan manual

Sumber : Buku Bahasa Pemrograman Lengkap

3) Proses

Data yang dimasukan kemudian diproses untuk menghasilkan jawaban atas persoalan yang ingin dipecahkan.Kegiatan memproses data ini disimbolkan dengan persegi panjang.Sama seperti simbol pada masukan, penulisan operasi-operasi pada data dapat dilakukan secara satu per satu maupun secara keseluruhan.

Gambar 1.5 Simbol proses

Sumber : Buku Bahasa Pemrograman Lengkap

4) Keluaran

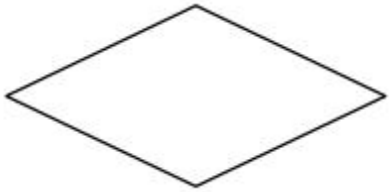
Keluaran adalah hasil dari pemrosesan data dan merupakan jawaban atas permasalahan yang ada. Keluaran ini harus ditampilkan pada layar monitor agar dapat dibaca oleh pengguna program. Sama seperti aturan pada simbol-simbol sebelumnya, penulisan hasil pemrosesan data dapat dilakukan secara satu per satu maupun secara keseluruhan.

Gambar 1.6 Simbol Keluaran

Sumber : Buku Bahasa Pemrograman Lengkap

5) Percabangan

Yang dimaksud dengan percabangan disini adalah suatu kegiatan untuk mengecek atau memeriksa suatu keadaan apakah memenuhi suatu kondisi tertentu atau tidak. Jadi dalam percabangan ini, kita harus menuliskan



Gambar 1.7 Simbol Percabangan

Sumber : Buku Bahasa Pemrograman Lengkap

kondisi apa yang harus dipenuhi oleh suatu keadaan. Hasil dari pemeriksaan keadaan ini adalah YA atau TIDAK. Jika pemeriksaan keadaan menghasilkan kondisi yang benar, maka jalur yang dipilih adalah jalur yang berlabel YA, sedangkan jika pemeriksaan keadaan menghasilkan kondisi yang salah, maka jalur yang dipilih adalah jalur yang berlabel TIDAK.Berbeda dengan aturan pada simbol-simbol sebelumnya, penulisan kondisi harus dilakukan secara satu per satu (satu notasi percabangan untuk satu kondisi).

6) Sub rutin

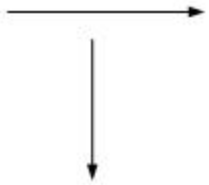
Sub rutin adalah suatu bagian dalam program yang dapat melakukan (atau diberi) tugas tertentu. Jadi sub rutin merupakan "program kecil" yang menjadi bagian dari suatu program yang besar. Sub rutin ada dua macam, yaitu prosedur (procedure) dan fungsi (function). Perbedaan antara keduanya adalah setelah dipanggil prosedur tidak mengembalikan suatu nilai sedangkan fungsi selalu mengembalikan suatu nilai. Dalam bahasa C++ kedua sub rutin tersebut dijadikan satu yaitu function, sedangkan untuk Java menggunakan class dimana keduanya bisa diatur untuk dapat mengembalikan nilai atau tidak dapat mengembalikan nilai. Sub rutin ini akan didiskusikan pada bab tersendiri. Sub rutin memiliki suatu flow chart yang berdiri sendiri diluar flow chart utama. Jadi dalam simbol sub rutin, kita cukup menuliskan nama sub rutinnya saja, sama seperti jika kita melakukan pemanggilan terhadap suatu sub rutin pada program utama (main program) yang akan anda pelajari pada bagian atau bab lain pada

Gambar 1.8 Simbol Sub Rutin

Sumber : Buku Bahasa Pemrograman Lengkap

buku ini. Aturan penulisan simbol sub rutin sama dengan simbol percabangan, yaitu penulisan nama sub rutin dilakukan secara satu per satu.

7) Arah aliran
Arah aliran merupakan jalur yang harus diikuti dan merupakan garis penghubung yang menghubungkan setiap langkah pemecahan masalah yang ada dalam flow chart.Arah aliran ini disimbolkan dengan anak panah.



Gambar 1.9 Simbol Arah Aliran

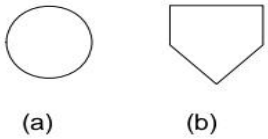
Sumber : Buku Bahasa Pemrograman Lengkap

8) Terminator
Terminator berfungsi untuk menandai titik awal dan titik akhir dari suatu flow chart. Simbol terminator ini diberi label MULAI untuk menandai titik awal dari flow chart dan label SELESAI untuk menandai titik akhir dari flow chart. Jadi dalam sebuah flow chart harus ada dua simbol terminator, yaitu simbol terminator untuk MULAI dan SELESAI.

Gambar 1.10 Simbol Terminator

Sumber : Buku Bahasa Pemrograman Lengkap

9) Konektor
Konektor berfungsi untuk menghubungkan suatu langkah dengan langkah lain dalam msebuah flow chart dengan keadaan on page atau off page. Yang dimaksud dengan konektor on page adalah konektor yang digunakan untuk menghubungkan suatu langkah dengan langkah lain dalam satu halaman. Sedangkan konektor off page adalah konektor untuk menghubungkan suatu langkah dengan langkah lain dalam halaman yang berbeda. Konektor ini digunakan apabila ruang gambar yang kita gunakan untuk menggambar flow chart tidak cukup luas untuk memuat flow chart secara utuh, jadi perlu dipisahkan atau digambar di halaman yang berbeda.



Gambar 1.11 Simbol konektor on page (a) dan off page (b)

Sumber : Buku Bahasa Pemrograman Lengkap

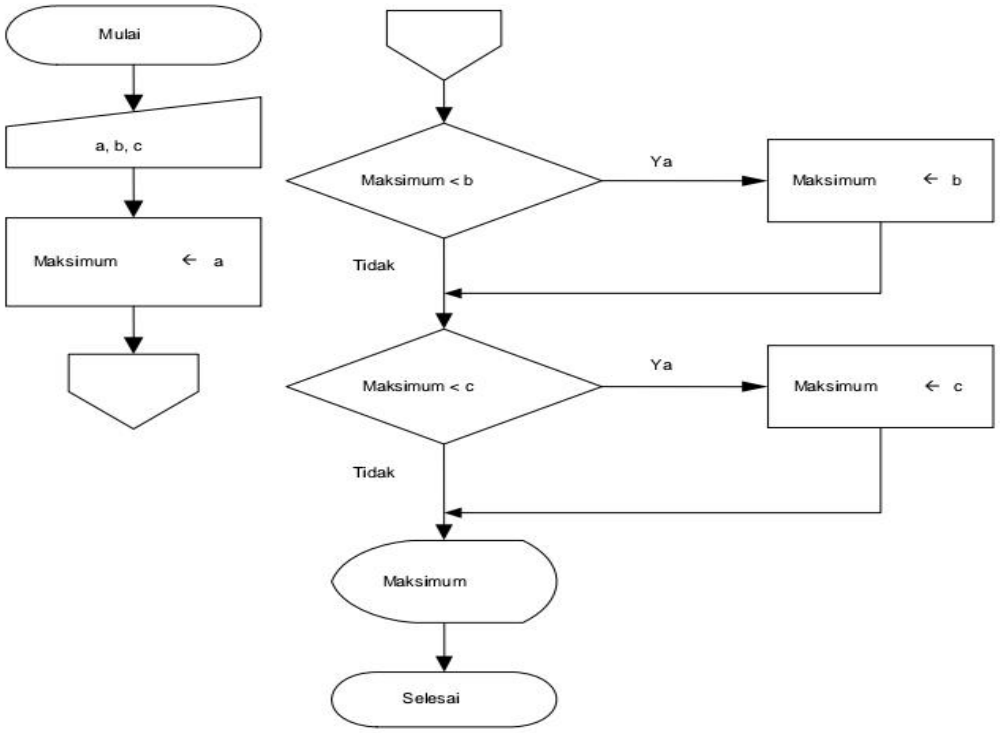
10) Dokumen

Gambar 1.12 Simbol Dokumen

Sumber : Buku Bahasa Pemrograman Lengkap

Dokumen merupakan tampilan data secara fisik yang dapat dibaca oleh manusia. Data ini biasanya merupakan hasil pemecahan masalah (informasi) yang telah dicetak (print out).

Berikut adalah flow chart untuk permasalahan yang diberikan (mencari bilangan terbesar dari tiga bilangan acak yang dimasukkan).



Gambar 1.13 Flowchart

Sumber : Buku Bahasa Pemrograman Lengkap

3. Pseudo code

Pseudo code adalah algoritma yang bentuknya (strukturnya) sangat mirip dengan bahasa pemrograman khususnya bahasa pemrograman terstruktur seperti pascal. Kemiripan ini merupakan keuntungan dari pseudo code karena implementasi atau penerjemahan algoritma ke dalam source code suatu bahasa pemrograman sangatlah mudah meskipun penggunaannya tidak sepopuler flow chart. Dalam penulisannya, pseudo code harus terdiri dari tiga bagian, yaitu :

1. Judul algoritma
   Bagian yang terdiri atas nama algoritma dan penjelasan(spesifikasi) dari algoritma tersebut. Nama sebaiknya singkat dan menggambarkan apa yang dapat dilakukan oleh algoritma tersebut.
2. Deklarasi
   Bagian untuk mendefinisikan semua nama yang digunakan di dalam program. Nama tersebut dapat berupa nama tetapan, peubah atau variabel, tipe, prosedur, dan fungsi.
3. Deskripsi
   Bagian ini berisi uraian langkah-langkah penyelesaian masalah yang ditulis dengan menggunakan aturan-aturan yang akan dijelaskan selanjutnya. Algoritma untuk permasalahan di atas yaitu mencari bilangan terbesar dari tiga bilangan acak yang dimasukkan dengan menggunakan pseudo code adalah:

   Algoritma bilangan terbesar

```
{algoritma ini mencari bilangan terbesar dari tiga bilangan yang dimasukkan}
deklarasi
        a,b,c,mak : integer
deskripsi
        read(a,b,c)
        mak ← a
        if (mak<b)
                mak ← b
        else if(mak<c)
                mak ← c
        end if
        write(mak)
```

Gambar 1.14 Algoritma bilangan terbesar

Sumber : Buku Bahasa Pemrograman Lengkap

Dalam pseudo code, garis bawah harus digunakan untuk kata algoritma (yang diikuti oleh judul dari algoritma), kata deklarasi, kata deskripsi, tipe data, read, write, if, else, end if, for, end for, while, end while, do while, dan end do while.

D. SYARAT - SYARAT ALGORITMA

Syarat-Syarat Algoritma menurut Donald E. Knuth, yaitu:

1. Finiteness (Keterbatasan)
   Algoritma harus berakhir setelah melakukan sejumlah langkah proses

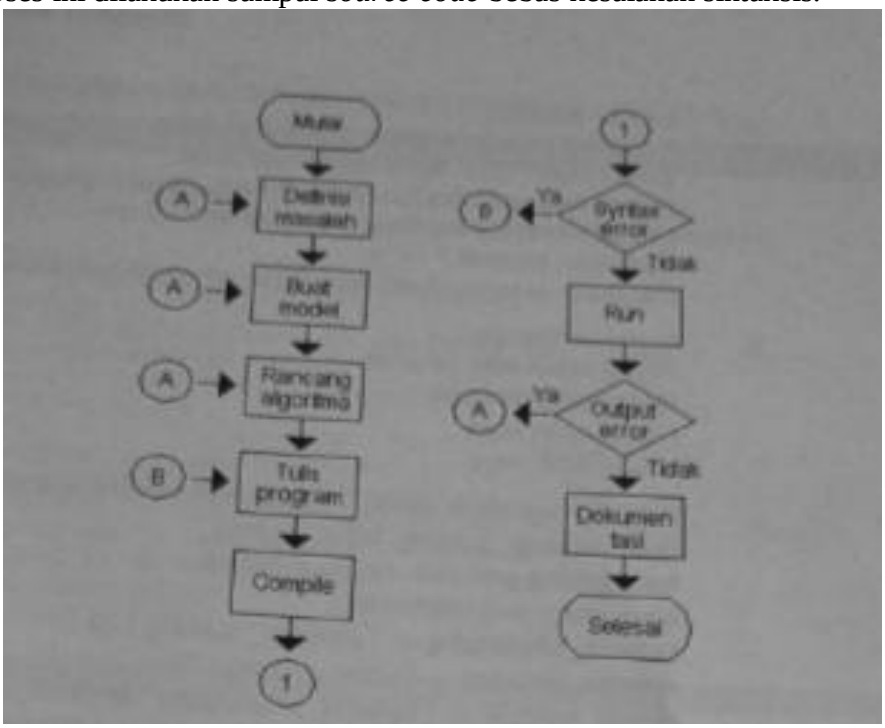
2. Definiteness (Kepastian)
   Setiap langkah algoritma harus didefinisikan dengan tepat dan tidak menimbulkan makna ganda
3. Input (Masukan)
   Sebuah algoritma memiliki nol atau lebih masukan (input) yang diberikan kepada algoritma sebelum dijalankan
4. Output (Keluaran)
   Setiap algoritma memberikan satu atau beberapa hasil keluaran
5. Effectiveness (Efektivitas)
   Langkah-langkah algoritma dikerjakan dalam waktu yang "wajar"

E. PEMECAHAN MASALAH DENGAN KOMPUTER

Algoritma yang dirancang bukan merupakan produk akhir.Karena yang kita inginkan adalah pemecahan masalah dengan bantuan komputer maka algoritma harus diubah menjadi program komputer yang dapat dijalankan perangkat keras komputer. Gambar 1.15 memperlihatkan tahapan proses pemecahan masalah dengan bantuan komputer.

a. Definisi masalah
   Masalah yang akan dipecahkan perlu diperjelas misalnya mengenai lingkup dan asumsi. Perhitungan luas dan keliling segitiga misalnya perlu memastikan bentuk segitiga: siku-siku, sama kaki, sama sisi, atau segitiga sembarang. Bentuk segitiga yang berbeda memerlukan data masukan dan rumus perhitungan yang berbeda.
b. Membuat model
   Yang dimaksud dengan model di sini adalah model perhitungan atau model matematika untuk memcahkan masalah.Untuk menghasilkan daftar ranking prestasi akdemik mahasiswa diperlukan pengurutan (sorting) indeks prestasi.Untuk menghitung panjang kabel minimum yang diperlukan dalam menghubungkan sejumlah computer pada lokasi yang berbeda misalnya harus menggunakan model *minimum spanning tree.*
c. Merancang algoritma
   Setelah model ditentukan rincian proses dibuat dalam bentuk algoritma. Algortima yang dirancang menggunakan ketiga jenis algoritma, disajikan dalam bentuk *pseudocode, flowchart,* atau kode maya.
d. Menulis program
   Rancangan algoritma diubah menjadi program computer (*source code*) sesuai dengan *compiler* yang akan digunakan. *Source code* ini diketikkan ke dalam komputer dengan menggunakan editor teks seperti notepad atau editor yang disediakan IDE *compiler.*
e. Kompilasi program (*Compile*) dan Kesalahan Sintaksis
   Compile adalah proses menerjemahkan *source code* menjadi bentuk yang dapat dijalankan mesin komputer. Proses penerjemahan tersebut memerlukan sejenis piranti lunak yang disebut *compiler. Compiler Borland* C++, GCC (GNU C Compiler), dan Visual C++ adalah contoh *compiler* bahasa C.
   Apabila *source code* tidak sesuai dengan kaidah bahasa pemrograman yang digunakan *compiler* maka akan menimbulkan *syntax error.*Banyak hal dapat menyebabkan *syntax error. Compiler* akan memberitahukan

instruksi yang menyebabkan kesalahan.Bagian *source code* yang menyebabkan *syntax error* diperbaiki dan program di-*compile* ulang. Proses ini dilakukan sampai *source code* bebas kesalahan sintaksis.



Gambar 1.15 Tahapan Pemecahan Masalah Dengan Komputer

Sumber: Buku Algoritma dan struktur data bahasa C, 2009.

f. Run dan Kesalahan Output

Apabila *source code* yang di-*compile* bebas kesalahan sintaksis maka oleh *compiler* akan dibentuk *file* yang disebut *executable code. Executable code* berupa kode mesin komputer yang dapat dieksekusi (dijalankan, di-*run)* perangkat keras komputer.*Executable code* dijalankan berulang kali dan diberi data yang berbeda.Proses ini disebut program *testing.*Hasil keluaran (*output*) diperiksa. Sebuah program disebut benar apabila selalu memberikan hasil yang benar.

Kesalahan *output* menyatakan kesalahan logika program yang dapat disebabkan banyak hal: masalah yang tidak terdefinisi lengkap, model yang salah, salah rancang algoritma, atau salah menerjemahkan algortima menjadi *source code*. Lakukan perbaikan terhadapa *source code* dan di-*compile* kembali. Proses ini diulangi sampai program menghasilkan keluaran yang benar.

g. Dokumentasi

Terhadap program (*source code*) yang kita tulis mungkin pada suatu saat akan dilakukan perubahan (modifikasi). Seorang pemrogram hanya bisa melakukan perubahan terhadap sebuah program apabila dia memahami alur logika program tersebut.Memahami alur logika program dengan membaca *source code* sangat menyita waktu terutama jika program panjang atau program ditulis secara tidak terstruktur.Pada tahap

pengembangan program perlu dilakukan dokumentasi. untuk menjelaskan alur logika program dan sangat membantu proses modifikasi program.

F. KESIMPULAN

Algoritma adalah metode efektif yang diekspresikan sebagai rangkaian terbatas.Algoritma juga merupakan kumpulan perintah untuk menyelesaikan suatu masalah.Perintah-perintah ini dapat diterjemahkan secara bertahap dari awal hingga akhir. Masalah tersebut dapat berupa apa saja, dengan syarat untuk setiap permasalahan memiliki kriteria kondisi awal yang harus dipenuhi sebelum menjalankan sebuah algoritma. Jadi, algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis.Kata lgis merupakan kata kunci dalam algoritma.Langkah-langkah dalam algoritma harus logis dan harus dapat ditentukan bernilai salah atau benar. Untuk membuat algoritma dari suatu permasalahan, biasanya digunakan salah satu dari tiga buah notasi algoritma yang dikenal, yaitu

- Deskriptif Algoritma
- Flow Chart
- Pseudo Code

G. SOAL LATIHAN

1. Jelaskan tentang algoritma, noatsi algoritma dan syarat-syarat algoritma!
2. Sebutkan alasan mengapa harus menggunakan bahasa C ?
3. Sebutkan 3 buah notasi algoritma yang dikenal !
4. Mengapa notasi algoritma flowchart paling banyak digunakan? Jelaskan!
5. Jelaskan mengenai pemecahan masalah algoritma dengan bantuan computer!
6. Apa yang dimaksud dengan pemrograman, bahasa pemrograman dan program ?
7. Jelaskan dengan contoh kasus algoritma yang memiliki struktur perulangan for !
8. Apa yang dimaksud dengan kompilasi program (compile)?
9. Apa yang dimaksud kompilasi program?
10. Apa yang kamu ketahui tentang bahasa C?
11. Jelaskan apa itu pseudo code!
12. Jelaskan apa yang dimaksud dengan deskriptif algoritma!
13. Sebutkan langkah-langkah pemecahan masalah dengan bantuan komputer!
14. Apa yang akan terjadi jika source code tidak sesuai dengan kaidah bahasa pemrograman ?
15. Jelaskan dengan contoh kasus algoritma yang memiliki struktur runtunan !

BAB 2: STRUKTUR PROGRAM BAHASA C

A. KONSEP STRUKTUR PROGRAM

Dengan kemajuan tekhnologi di dunia, membuat akal pikiran manusia semakin mahir dalam menciptakan sesuatu seperti menciptakan sebuah computer. Dimana pada awalnya computer merupakan mesin yang tidak bisa apa-apa. Manusia pun memberikan serangkaian intruksi kepada komputer, salah satunya dalam pembuatan program – program yang dapat membantu manusia dalam menyelesaikan pekerjaan dengan cepat, baik dan memuaskan.

B. PROGRAM BAHASA C

Bahasa C adalah bahasa pemograman yang dapat dikatakan berada di antara bahasa beraras rendah dan beraras tinggi. Bahasa beraras rendah artinya bahasa yang berorientasi pada mesin dan beraras tinggi berorientasi pada manusia. Bahasa beraras rendah, misalnya bahasa assembler, bahasa ini ditulishanya dimengerti oleh mesin saja. Sedangkan bahasa beraras tinggi relative digunkan, karena ditulis dengan bahasa manusia sehingga mudah dimengerti dan tidak tergantung pada mesin.

Bahasa C mempunyai ciri khas tersendiri dari bahasa pemograman sebelumnya seperti Pascal. Ciri khas inilah yang membuat bahasa C menjadi populer dari bahasa pemograman yang lain. Ciri khas bahasa C, yakni:

- Berukuran kecil
- Penggunaan lebih leluasa pada pemanggilan fungsi.
- Gaya penulisan lebih bebas tidak seperti Pascal.
- Bahasa pemograman terestruktur.
- Dapat menggunakan bahasa pemograman tingkat rendah (paada operasi bitwise) dan tetap dapat mudah dibaca.

Dalam bahasa C fungsi dapat dibagi menjadi dua, yairu fungsi pustaka atau fungsi yang telah tersedia dalam Turbo C dan fungsi yang didefinisikan atau dibuat oleh programmer.

C. KOMPONEN BAHASA C

8. Karakter

- *Huruf, angka, dan garis bawah*
  Huruf adalah A s/d Z dan a s/d z, angka adalah 0 s/d 9, dan garis bawah adalah _.
- *White-space*
  Karakter-karakter *space, tab, line-feed, carriage-return, form-feed, vertical-tab,* dan *new-line* disebut *white-space character* karena mereka mempunyai fungsi sebagai spasi antara kata-kata atau baris-baris. Setiap konstanta dan identifier selalu dipisahkan oleh karakter-karakter ini.
- *Tanda baca dan karakter khusus*

Tanda baca dan karakter khusus dalam bahasa C mempunyai kegunaan yang bervariasi, dari pengorganisasian teks program sampai pendefivisian tugas-tugas yang akan dilakukan oleh compiler.
Karakter-karakter yang termasuk ke dalam kelompok ini adalah:
, . ; : ? ‘ ” ( ) [ ] { } < > ! | / \ ~ + # % & ^ * – =

- *Escape Sequence*

  Karakter-karakter escape-sequence adalah suatu urutan karakter yang digunakan untuk mewakili ekspresi suatu karakter lain. Daftar karakter-karakter escape-sequence adalah sebagai berikut.

\n	new-line
\t	horizontal-tab
\v	vertical-tab
\b	backspace
\r	carriage-reurn
\f	form-feed
\a	bell
\’	tanda-petik
\”	tanda-kutip
\\	backslash
\ddd	karakter ASCII dalam notasi oktal
\xddd	karakter ASCII dalam notasi heksadesimal

9. Operator

Operator (lambang operasi) adalah simbol-simbol, baik berupa satu atau beberapa karakter, yang menetapkan bagaimana suatu nilai dimanipulasi. Itu adalah:
! ~ – + * / % < > = | ^ ‘ ++ — -= += /= %= << >> == != <= >= |= && || ?: &= ^= <<= >>=

10. Konstanta

- *Integer Constant*

  Suatu integer-constant adalah angka desimal, oktal, atau heksadesimal yang mewakili suatu nilai bilangan bulat. Suatu konstanta-desimal (berbasis-sepuluh) dibentuk oleh sekumpulan angka 0 s/d 9, dengan ketentuan tidak dimulai oleh angka 0. Suatu konstanta-oktal (berbasis-delapan) dibentuk oleh sekumpulan angka 0 s/d 7, dengan ketentuan dimulai oleh angka 0. Suatu konstanta-heksadesimal (berbasis-enambelas) dibentuk oleh sekumpulan angka 0 s/d 9 dan atau huruf A s/d F atau a s/d f, dengan ketentuan dimulai oleh angka-huruf 0X atau 0x. Contoh:

Desimal	Oktal	Heksadesimal
10	012	0xA
132	0204	0x84
32179	076663	0x7DB3

- *Floating-point Constant*

  Suatu floating-point constant adalah angka desimal yang mewakili suatu nilai bilangan nyata. Nilai dari konstanta ini meliputi porsi-porsi bilangan bulat, pecahan, dan eksponen. Aturan penulisan konstanta ini adalah seperti dalam format berikut.

[digit][.digit][E|e[+|-]digit]

Di mana digit adalah berupa sekumpulan angka 0 s/d dan E atau e sebagai simbol eksponen. Contoh:

12E-3 berarti 12 kali 10 pangkat -3
6.25E+4 berarti 6.25 kali 10 pangkat 4

- *Character Constant*

  Suatu character-constant dibentuk dengan menempatkan suatu karakter di antara dua tanda-petik (‘ ‘). Suatu escape-character dipandang sebagai satu karakter, oleh karena itu dapat dijadikan suatu character constant. Contoh:

  ‘A’
  ‘\n’
  ‘\0x41’

- *String-Literal*

  Suatu string-literal dibentuk oleh satu atau sekumpulan karakter yang ditempatkan di antara dua tanda-kutip. Contoh:

  “BANDUNG”
  “Hari Jum\â€™at”
  “Tit\a\a”
  “Baris-1\nBaris-2”

11. Identifier

Identifier adalah nama yang diberikan untuk *konstanta-bernama, variabel, jenis-data, fungsi,* dan *label* di dalam program. Setiap identifier yang akan digunakan harus didefinisikan sebelumnya. Suatu identifier dibentuk oleh satu atau beberapa buah karakter, sebanyak-banyaknya 31 karakter, yang dimulai oleh suatu huruf atau garis-bawah dan dapat diikuti oleh huruf, angka, atau garis-bawah. Catatlah, bahasa C membedakan penggunaan huruf-besar dan huruf-kecil dalam suatu identifier. Contoh:

ClrScr
H2o
Tempat Tinggal
Nama_Siswa
Kota
Pilihan

12. Keyword

Untuk kegunan dalam pemrograman, bahasa C mencadangkan sejumlah identifier yang telah didefinisikan, disebut keyword, bagi pemrogram. Kata-kata ini merupakan instruksi terhadap C untuk mengerjakan/menyatakan suatu hal tertentu. Oleh karena itu semua keyword tidak dapat didefiniskan-ulang dan hanya digunakan sesuai dengan peruntukannya.

Berikut adalah daftar keyword dalam bahasa C:

auto break case char const continue default do double else enum extern float for goto if int long register return short signed sizeof static struct switch typedef union unsigned void volatile while

D. OPERASI DAN OPERATOR

1. Operator

Ditinjau berdasarkan sifatnya, operator dapat dibedakan menjadi tiga macam, yaitu :

1) unary : operator yang melibatkan sebuah operand
2) binary : operator yang melibatkan dua operand
3) ternary : operator yang melibatkan tiga operand

Perhatikan :

- **Operator Unary**

  Operator unary adalah operator yang hanya melibatkan sebuah operand. kategorinya sebagai berikut:

Operator	Jenis Operator	Contoh
+	Membuat nilai positif	+7
-	Membuat nilai negatif	-7
++	Increment	C++
--	Decrement	C--

- **Operator Binary**

  Operator binary adalah operator yang digunakan dalam operasi yang melibatkan dua buah operand.

  1) Operator Aritmetika

  Operator aritmatika adalah operator yang digunakan untuk melakukan operasi-operasi aritmetika. Adapun yang termasuk dalam operator aritmetika di dalam C++ adalah seperti berikut ini.

Operator	Jenis Operasi	Contoh
+	Penjumlahan	1 + 1 = 2
-	Pengurangan	2 – 1 = 1
*	Perkalian	1 * 1 = 1
/	Pembagian	9 / 3 = 3
%	Sisa bagi (modulus)	10 % 3 = 1

Prioritas dalam penggunaan harus diperhatikan Contoh : Menghitung Diskriminan

D = b2
– 4 AC

Cara menulis di program : D = b*b - 4 * A * C atau
D = (b*b) – (4*A*C)

Penulisan Programnya menjadi :

```
# include <iostream.h> // tampilkan di layar
# include <conio.h>// melibatkan pembersihan layar
void main ( )
{
```

```
int a, b, c, d;
clrscr ( ); // digunakan untuk membersihkan layar □ <conio.h>
a = 5;
b = 600;
c = 8;
d = b*b – 4 * a * c;
cout << ”Nilai d = “ << d << ‘\n’;
}
```

2) Operator Logika

Operator logika adalah operator yang digunakan untuk melakukan operasi dimana nilai yang dihasilkan dari operasi tersebut hanya berupa nilai benar (true) dan salah (false). Adapun yang termasuk dalam operator logika adalah seperti berikut.

Operator	Jenis Operasi	Contoh
&&	AND (dan)	1 && 1 = 1
||	OR (atau)	1 || 0 = 1
!	NOT (negasi)	! 0 = 1

3) Operator Relasional

Operator relasional adalah operator yang digunakan untuk menentukan relasi atau hubungan dari dua buah operand. Operator ini ditempatkan di dalam sebuah ekspresi, yang kemudian akan menentukan benar atau tidaknya sebuah ekspresi.

Operator Jenis Operasi Contoh

Operator	Jenis Operasi	Contoh
>	Lebih besar	(3 > 2) = 1
<	Lebih Kecil	(3 < 2) = 0
>=	Lebih besar atau sama dengan	(3 >= 3) = 1
<=	Lebih kecil atau sama dengan	(3 <= 2) = 0
==	Sama dengan	(5 == 2) = 0
!=	Tidak sama dengan	(5 != 2) = 1

4) Operator Bitwise

Operator yang digunakan untuk melakukan operasi-operasi yang berhubungan dengan pemanipulasian bit. Operator bitwise ini hanya dapat dilakukan pada operand yang bertipe char dan int saja karena ini berkoresponden dengan tipe byte atau word di dalam bit. Yang termasuk dalam operator bitwise dalam C++ adalah sebagai berikut:

Operator	Jenis Operasi	Contoh
&	AND	1 & 0 = 0
|	OR	1 | 0 = 1
^	EXCLUSIVE OR (XOR)	1 ^ 1 = 0
~	NOT	~1 = 0
>>	Shift Right	16 >> 1 = 8
<<	Shift Left	1 << 2 = 4

5) Operator Penugasan (asigment)

Misal : a = 5; A = 2 + b;

- Penugasan dalam ungkapan a = 2 + (b+1)
- Penugasan berganda → a = b = c = d = e = 1;

6) Increment

Increment adalah suatu penambahan nilai yang terjadi pada sebuah variabel. Adapun yang digunakan untuk melakukan incrementadalah operator ++. Operator ini akan menambahkan nilai dari suatu variabel dengan nilai 1. Terdapat dua buah jenis increment yang terdapat dalam bahasa C++, yaitu pre-increment dan post-increment. Arti pre-increment yaitu melakukan penambahan nilai sebelum suatu variabel itu diproses, sedangkan post-increment yaitu melakukan proses terlebih dahulu sebelum dilakukan penambahan nilai. Adapun bentuk umum dari pre-increment dan post-increment dapat dilihat dibawah ini

```
// Melakukan pre-increment
++nama_variabel;
// Melakukan post-increment
Nama_variabel++;
```

Berikut diperlihatkan contoh program untuk melakukan pre-increment

```
#include <iostream>
void main()
{
 int C;              // Mendeklarsikan variabel C
// Mengisikan nilai ke dalam variabel C dengan nilai 5
 C = 5;
 // Melakukan pre-increment
 cout<<"Nilai C awal : "<<C<<endl;
 cout<<"Nilai ++C : "<<++C<<endl;
 cout<<"Nilai C akhir : "<<C<<endl;
 cout<<'\n';
          // Mengubah nilai yang terdapat dalam variabel C dengan nilai 10
 C = 10;
          // Melakukan post-increment
 cout<<"Nilai C awal : "<<C<<endl;
 cout<<"Nilai C++ : "<<C++<<endl;
 cout<<"Nilai C akhir : "<<C<<endl;
}
```

Hasil dari program diatas adalah

Pre increment	Post-increment
Nilai C awal : 5	Nilai C awal : 10
Nilai ++C : 6	Nilai C++ : 10
Nilai C akhir : 6	Nilai C akhir : 11

7) Decrement

Decrement adalah suatu pengurangan nilai yang terjadi pada sebuah variabel. Adapun bentuk umum dari pre-increment dan post-increment dapat dilihat dibawah ini.

```
#include <iostream>
void main()
```

```
{
 int C;                  // Mendeklarsikan variabel C
                         // Mengisikan nilai ke dalam variabel C dengan
nilai 5
 C = 5;
                         // Melakukan pre-increment
 cout<<"Nilai C awal : "<<C<<endl;
 cout<<"Nilai --C : "<<--C<<endl;
 cout<<"Nilai C akhir : "<<C<<endl;
 cout<<'\n';
                         // Mengubah nilai yang terdapat dalam variabel C
                         // dengan nilai 10
  C = 10;
                         // Melakukan post-decrement
  cout<<"Nilai C awal : "<<C<<endl;
  cout<<"Nilai C-- : "<<C--<<endl;
  cout<<"Nilai C akhir : "<<C<<endl;
 }
```

Hasil dari program diatas adalah

Pre increment Nilai C awal : 5 Post-increment Nilai C awal : 10

Nilai ++C : 4 Nilai C++ : 10

Nilai C akhir : 4 Nilai C akhir : 9

8) Operator Majemuk

Berikut program increment dan decriment majemuk:

X = X + 2 → X +=2;

Y = Y * 4 → Y *=4;

Contoh yang lain :

-= → Y -=2 identik dengan Y = Y-2;

/= → X /=2 identik dengan X = X/2;

%= → Y %=2 identik dengan Y = Y%2;

9) Manipulator Setw(n)

Manipulator setw (n), digunakan u/ mengatur lebar tampilan data. Misalnya : bilangan 145, dengan mengatur setw(6); → dampaknya akan diperoleh enam ruang spasi:

1	2	3	4	5	6
.	.	.	1	4	5

Contoh program :

```
 // Menampilkan 3 buah barang menggunakan manipulator setw(4)
# include <iostream.h>
# include <iomanip.h>>//lab untuk manipulasi
# include <conio.h>
void main ()
{
 int jumbar1 = 150,
jumbar2 = 23,
jumbar3 =1406;
 clrscr( );
```

```
cout << “barang 1 = “ << setw (4)<<jumbar1<<endl;
cout << “barang 2 = “ << setw (4)<<jumbar2<<endl;
cout << “barang 3 = “ << setw (4)<<jumbar3<<endl;
}
```

E. KONVERSI TIPE DATA IMPLISIT

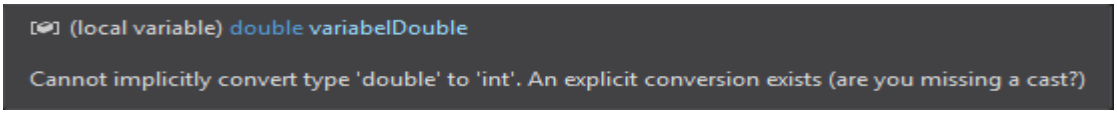
Yang dimaksud dengan konversi secara implisit adalah pada saat *compiler* mampu melakukan konversi tanpa perlu diberi instruksi untuk melakukan konversi. Tidak semua konversi bisa dilakukan secara implisit. C# akan melaakukan konversi secara implisit selama dimungkinkan. Kondisi yang memungkinkan terjadinya konversi tipe data secara implicit:

Pertama, konversi secara implisit dimungkinkan apabila tipe data yang akan dikonversikan memiliki ukuran atau rentang nilai yang lebih kecil daripada tipe data hasil konversi. Contohnya, tipe data int memiliki ukuran **4 Byte** sedangkan tipe data double memiliki ukuran **8 Byte**. Oleh karena itu tipe data int bisa dikonversi menjadi tipe data double secara implisit. Namun, tidak sebaliknya.

```
int variabelInt = 2147483647;
double variabelDouble = variabelInt;
```

Pada potongan kode program di atas, compiler akan mengkonversi nilai variabel variabelInt yang bertipe data int secara implisit. Sedangkan potongan kode program di bawah, akan menghasilkan eror yang menyatakan bahwa compiler tidak bisa melakukan konversi secara implisit dari double ke int.

```
double variabelDouble = 2147483647;
int variabelInt = variabelDouble;
```

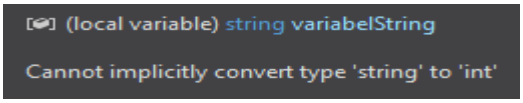


Compiler Tidak Bisa Mengkonversi Tipe double Ke Tipe int Secara Implisit.

Kedua, konversi secara implisit hanya bisa dilakukan antar tipe data yang sepadan. string dan int bukan merupakan tipe data yang sepadan. Oleh karena itu, *compiler* tidak bisa mengkonversi tipe data string menjadi int secara implisit.

```
string variabelString = Console.ReadLine();
int variabelInt = variabelString;
```

Potongan kode program di atas akan menghasilkan eror seperti:



Eror Yang Dihasilkan Ketika Anda Mencoba Mengkonversi Tipe string ke Tipe int Secara Implisit

F. TYPE CONVERSION (TYPE CASTING)

Type conversion atau type casting yaitu mengubah dari satu tipe data ke tipe data yang lain. Type conversion dapat menggunakan mekanisme casting

yang berfungsi untuk memaksakan suatu variable untuk dikonversi menjadi tipe data sesuai dengan casting yang diberikan . Tujuan dari konversi data adalah :

1. Convert dari String ke Integer, untuk melakukan perhitungan biasanya data dari user yang melakukan input.
2. Convert dari Integer ke String, Float ke String, Double ke string, untuk memasukan angka kedalam suatu kalimat.
3. Convert dari Integer ke Float, atau Integer ke Double, untuk perhitungan apabila hasil memiliki nilai decimal.

Jenis tipe-tipe data

Type	Size (bits)	Range
Bit	1	0, 1
bool	8	0, 1
Chair	8	-128 to 127
Unsigned char	8	0 to 255
Signed char	8	-128 to 127
Int	16	-32768 to 32767
short int	16	-32768 to 32767
unsigned int	16	0 to 65535
signed int	16	-32768 to 32767
long int	32	-2147483648 to 2147483647
unsigned long int	32	0 to 4294967295
signed long int	32	-2147483648 to 2147483647
float	32	±1.175e-38 to ±3.402e38
double	32	±1.175e-38 to ±3.402e38

Untuk memasukkan data diperlukan juga operator yang dapat mengcompilenya. Adapun operator antara lain :

Contoh dari pengkonversian data yaitu :

```
Double nilai;
sttring kalimat;
nilai = 9.57;
kalimat = "Nilai UTS anda adalah " + Convert.toString(nilai);
```

G. PRE-PROCESSOR DIRECTIVE

Preprocessor directive adalah suatu pernyataan yang akan diikutsertakan dalam program, dimana pernyataan tersebut akan decompile sebelum proses compilasi yang sebenarnya dilakukan. Preprocessor directive ditandai dengan simbol #. Bertugas untuk mengarahkan prepocessor yang akan digunakan untuk membaca file header atau bisa dikatakan sebagai pengatur proses kompilasi.

Manfaat dari preprocessor directive yang bisa kita dapatkan yaitu :

1. Mengganti token dengan karakter tertentu

2. Menyisipkan file lain ke dalam file yang sedang aktif
3. Memberikan kondisi pada saat program di compile
4. Mengubah nomor baris dari program sumber dan mengubah nama file
5. Menerapkan aturan mesin untuk menentukan kode-kode tertentu

Penggunaan directives dapat bekerja dengan project. Project adalah suatu program yang di dalamnya beberapa file sumber dengan berbagai tipe. Suatu pekerjaan yang besar akan dapat diselesaikan dengan membagi unit-unit kecil. Contoh preprocessor directive : #include, #define, dan lainnya. Contoh file header : , , , dan lainnya. Contoh penulisannya : #include

using namespace std;

Preprocessor directive define merupakan suatu pengarah untuk mendefenisikan suatu preprocessor macro, konstanta dan variabel. Penulisan preprocessor directive define ini ialah #define. Contohnya :

#define PI 3.14

#define KUADRAT(x) (x*x)

H. KESIMPULAN

Didalam elemen dan struktur program terdapat 6 bagian antara lain: tipikal program c, elemen Bahasa c, operasi dan operator , konversi data secara implisit, type conversion, dan processor directive. Bahasa C sendiri pengertiannya adalah bahasa pemograman yang dapat dikatakan berada di antara bahasa beraras rendah dan beraras tinggi. Bahasa beraras rendah artinya bahasa yang berorientasi pada mesin dan beraras tinggi berorientasi pada manusia. Didalam elemen bahasa c, terdapat juga : Karakter , Operator , Konstanta, Identifier dan Keyword. Dalam operator juga terdapat unary, binary, serta ternary.

I. SOAL LATIHAN

1. Sebutkan 4 pembagian dari operator binary !
2. Jelaskan dua jenis decrement yang terdapat dalam bahasa C++ !
3. Buatlah operator, jenis operator dan contoh dari operator bitwise
4. Sebutkan jenis-jenis tipe data di dalam pemograman ?
5. Sebutkan tujuan convert dari integer ke double ?

BAB 3: INTRUKSI *INPUT* DAN *OUTPUT* BAHASA C

A. KONSEP *INPUT* DAN *OUTPUT*

Dalam sebuah Bahasa pemrograman terdapat suatu masukan (input) dan keluaran (output). Masukan dan keluaran ini digunakan sebagai interaksi pengguna terhadap program yang akan dieksekusinya. Masukan adalah suatu inputan yang diberikan oleh pengguna suatu program dimana masukan tersebut akan diteruskan kedalam proses dengan aturan-aturan yang disepakati oleh bahasa pemrograman dan akan ditampilkan hasil berupa keluaran (output) dari proses yang telah dilakukan. Redirection adalah pembelokkan, yaitu membelokkan standar output suatu program ke file atau membelokkan standar input suatu program dari suatu file.

B. INSTRUKSI MASUKAN DAN INSTRUKSI KELUARAN

Instruksi Masukan dalam bahasa pemrograman berupa interaksi pengguna berupa masukan yang diteruskan sesuai dengan aturan-aturan yang disepakati oleh bahasa pemrograman dalam sebuah proses, sedangkan instruksi keluaran yakni berupa tampilan atau hasil dari proses yang telah dilakukan

Kita akan membahas bagian-bagian yang terdapat dalam instruksi masukan dan instruksi keluaran, sehingga kita dapat membedakan dengan jelas mengenai proses masukan (Input) dan juga keluaran (output).

1. Cout (Standard Output)

   Dalam pemrograman Cout berfungsi untuk meletakan suatu informasi ke dalam bentuk *Standard output* biasa dalam bentuk layar. Pada bahasa C++ Cout merupakan turunan dari kelas *Iostream*. Tampilan data akan ditampilkan dalam bentuk layar. Penggunaan Cout menyisipkan operator (<<).

   Contoh:

```
#include <iostream>

using namespace std;

int main()
{
    char sample[] = "Matematika";

    cout << sample << " - Sangat Menyenangkan";

    return 0;
}
```

Output : `Matematika - Sangat Menyenangkan`

https://ide.geeksforgeeks.org/iv8eLRxSjD (klik untuk melihat)

Pada penggunaan program diatas penyisipan operator (<<) dan memasukan nilai sampel variabel string “Sangat Menyenangkan” dalam Cout standard output dan kemudian ditampilkan pada layar.

2. Cin (Standard Input)

Cin berfungsi sebagai pembaca data dari *Standard Input*, perangkat *Standard input* biasanya berupa keyboard. Ekstraksi operator yang digunakan (>>) untun mengekstraksi data dari objek Cin.

Contoh :

```
#include <iostream>
using namespace std;

int main()
{
    int age;

    cout << "Enter your age:20";
    cin >> age;
    cout << "\nYour age is:20 " << age;

    return 0;
}
```

Output

```
:20

Your age is : 20
```

https://ide.geeksforgeeks.org/niAptfHHLT

a. Cin dengan sebuah variabel

Dalam bentuk `Cin >> Var`

Cin yang menggunakan satu variabel yakni *int* dan *float.*

Contoh	Hasil

//*---------- *
 //* Contoh : Membaca data bertipe int dan *
 //* float dari keyboard *
 //*---------- *
 #include <iostream.h>
 void main ()
 {
 int bil_x ; // Definisi bilangan bulat
 float bil_y ; // Definisi bilangan pecahan
 cout << "Masukkan sebuah bilangan bulat = " ;
 cin >> bil_x ;
 cout << "Masukkan sebuah bilangan pecahan = " ;
 cin >> bil_y;
 	
 cout << " Bilangan Bulat = " << bil_x << endl;
 cout << " Bilangan Pecahan = " << bil_y << endl;
 	
 return 0;
 }
	Masukkan sebuah bilangan bulat = 123
	 Masukkan sebuah bilangan pecahan = 23.1
	 Bilangan Bulat = 123
	 Bilangan Pecahan 23.1

Penggunaan int dan float memiliki fungsi sama dengan (Char, int, long, float ataupun doubel) yakni untuk membaca program data yang berubah ubah.

b. Cin dengan dua Variabel

Dalam bentuk `Cin >> var 1 >>var`

Penggunaan cin dua variabel dapat dilakukan dengan menggunakan tanda pisah berupa spasi, tab, dan enter.

Contoh	Hasil

//*----------*
//* Contoh : Untuk membaca data dengan lebih *
//* dari satu variabel *
//*----------*
#include <iostream.h>
void main ()
{
int bil_x ; // Definisi bilangan bulat
float bil_y ; // Definisi bilangan pecahan
cout << "Masukkan sebuah bilangan bulat dan " << endl;
cout << "Masukkan sebuah bilangan pecahan " << endl ;
cin >> bil_x >> bil_y;
 	
cout << " Bilangan Bulat = " << bil_x << endl;
cout << " Bilangan Pecahan = " << bil_y << endl;
return 0;
}
	Masukkan sebuah bilangan bulat dan
	Masukkan sebuah bilangan pecahan
	20 23.2 ↵
	Bilangan Bulat = 20
	Bilangan Pecahan 23.2

c. Cin untuk membaca karakter

Penggunaannya ialah dengan menekan tanda enter , seperti yang ditunjuk oleh panah merah. Jika pada penekanan enter tidak membuahkan hasil maka dapat menggunakan fungsi getch() atau getche().

d. Fungsu getch() dan getche()

Fungsi ini digunakan untuk membaca sebuah karakter tanpa haru menekan enter, fungsi ini dapat membaca tombol spasi, tab dan juga enter.

Bentuk penggunaan

```
Karakter = getch()

Karakter = getche()
```

Adapun perbedaan getch() dan getche() adalah:

Getch() : Tidak menampilkan tombol yang ditekan

Contoh	Hasil

//*----------*
//* Contoh : Membaca karakter dengan getch() *
//*----------*
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
void main()
{
char karakter;
clrscr();
cout << "Masukkan sebuah karakter ";
karakter = getch();
cout << "Anda mengetik " << karakter << endl;
cout << "Masukkan sebuah karakter ";
karakter = getch();
cout << "Anda mengetik " << karakter << endl;
return 0;
}
	Masukkan sebuah karakter a
	Tidak ditampilkan
	Anda mengetik a
	Masukkan sebuah karakter b
	Tidak ditampilkan
	Anda mengetik b

Getch() : Menampilkan karakter dari tombol yang ditekan

Pada penggunaan getch() dapat digunakan dengan menunggu sembarang tombol ditekan, dan tidak perlu meletakan variabel.

Contoh	Hasil

//*----------*
//* Contoh : getch() untuk memebaca sembarang *
//* tombol *
//*----------*
#include <iostream.h >
#include <conio.h>
void main()
{
cout << "Tekanlah sembarang tombol" << endl;
cout << "Untuk mengakhiri program ini " ;
getch();
}
	Tekanlah sembarang tombol
	Untuk mengakhiri program ini

3. Un-buffered standard error stream (cerr)

C++ cerr merupakan kesalahan standard yang akan menampilkan kesalahan, ceer merupakan bagian turunan dari *Iostream.* Cerr dalam C++ sebagai un-buffered sehingga dapat langsung menampilkan kesalahan, karena Cerr tidak memiliki buffer untuk menyimpan kesalahan dan akan ditampilkan nanti. Maka dalam penggunaannya cerr akan langsung menampilkan kesalahan.

```
#include <iostream>

using namespace std;

int main()
{
    cerr << "An error occurred";
    return 0;
}
```

Output:

An error occurred

4. Buffered standard error stream (clog)

Clog merupakan turunan dari *Iostream*, kegunaan Clog sama seperti Cerr yankni menampilkan kesalahan, namun pada Clog kesalahan pertama dimasukan ke dalam buffer dan di simpan sampai tidak terlalu penuh. Pada Clog pesan kesalahan juga akan ditampilkan pada layar.

```
#include <iostream>

using namespace std;

int main()
{
    clog << "An error occured";

    return 0;
}
```

Output

An error occurred

C. PENGATURAN LETAK DI LAYAR

IDE Borland C++ menyediakan sejumlah instruksi yang berhubungan dengan layar dan tampilan seperti pembersihan layar, pengaturan posisi cursor, dan pewarnaan teks. Instruksi-instruksi ini berbentuk *function* yang *function prototype*-nya disimpen dalam *header file conio.h.* jadi apabila *function-function* tersebut mau digunakan maka pada program harus ditambahkan instruksi #include<conio.h>. layar tampilan monitor computer pada modus teks terdiri dari 25 baris dan 80 kolom.

D. KESIMPULAN

Kesimpulan yang kita dapat kan ialah, Function prototype merupakan pernyataan dari deklarasi function, yang terdiri dari retun type, identitas, parameter dan tanpa badan dari function (definisi). Karena merupakan pernyataan maka mendirikan function prototype harus di akhiri dengan semicolon. Selanjutnya, Instruksi Masukan dalam bahasa pemrograman

berupa interaksi pengguna berupa masukan yang akan diteruskan sesuai dengan aturan-aturan yang disepakati oleh bahasa pemrograman dalam sebuah proses, sedangkan instruksi keluaran yakni berupa tampilan atau hasil dari proses yang telah dilakukan. Ada jenis instruksi input dan juga output yang dapat kita gunakan sesuai dengan kebutuhan.

E. SOAL LATIHAN

1. Apa yang dimaksud *function prototype*?
2. Apa yang dimaksud perintah Echo, cat, more, sort, grep, wc, cut dan uniq?
3. Apa yang dimaksud dengan *pipeline*?
4. Apa yang dimaksud dengan perintah masukan menggunakan cin()?
5. Sebutkan 2 cara menggunakan fungsi!

BAB 4: STRUKTUR KENDALI PEMILIHAN DAN PENGULANGAN

A. KONSEP STRUKTUR KENDALI

Struktur Kendali Pemilihan adalah suatu struktur dasar algoritma yang memiliki satu atau lebih kondisi tertentu dimana sebuah instruksi dilaksanakan jika sebuah kondisi/ persyaratan terpenuhi. Dalam membuat program, pengambilan keputusan seringkali dimanfaatkan. Pemanfaatan ini dapat dijumpai misalnya pada waktu:

1. Menentukan besarnya tunjangan yang diterima seseorang pegawai yang disesuaikan dengan jumlah anak yang ditanggung.
2. Menentukan nilai huruf mahasiswa berdasarkan score yang diperoleh
3. Menentukan suatu bilangan merupakan prima atau tidak.

Sedangkan, Instruksi perulangan digunakan untuk menjalankan satu atau beberapa instruksi sebanyak beberapa kali jika suatu kondisi terpenuhi hanya dengan menuliskan intruksi tersebut satu kali saja. Proses perulangan biasanya digunakan untuk pemasukan data, perhitungan, dan proses penampilan hasil pengolahan data.

Struktur perulangan terdiri dari empat bagian :

1. Kondisi perulangan, yaitu ekspresi boolean yang harus dipenuhi untuk melaksanakan pengulangan.
2. Badan perulangan, yaitu satu atau lebih instruksi yang akan diulang.
3. Inisialisasi, yaitu aksi yang dilakukan sebelum perulangan dilakukan pertama kali.

Terminasi, yaitu aksi yang mengakibatkan perulangan dihentikan.

B. STRUKTUR KENDALI PEMILIHAN

1. HUBUNGAN ANTAR NILAI

Hubungan antar nilai digunakan untuk membandingkan antara dua buah operand. Yang dimaksud operand adalah sebuah nilai atau variabel. Seperti dalam matematika yaitu kurang dari, lebih dari, sama dengan, dan sebagainya. Operator hubungan dalam bahasa C dapat dilihat pada Tabel berikut :

Tabel 2.1: Tabel macam-macam hubungan antar nilai dalam Bahasa C

Operator	Arti
<	Kurang dari
<=	Kurang dari atau sama dengan
>	Lebih dari
>=	Lebih dari atau sama dengan
==	Sama dengan
!=	Tidak sama dengan

2. INSTRUKSI IF dan IF ELSE

Pernyataan if adalah pernyataan yang dapat dipakai untuk mengambil keputusan berdasarkan suatu kondisi. Bentuk pernyataan ini ada dua macam:
a. if
b. if else**,** Bentuk Umumnya Satu Kasus:

```
if (kondisi)
  pernyataan ;
```

Pernyataan dilaksanakan jika dan hanya jika kondisi yang diinginkan terpenuhi, jika tidak program tidak memberikan hasil apa-apa. Berikut contoh program if satu kasus.

```
// Contoh Penggunaan if

# include <iostream.h>

void main ()
{
   int usia;

   cout << "Berapa usia Anda : ";
   cin >> usia;

   if (usia < 17)
      cout << "Anda tidak diperkenankan menonton" << endl;
}
```

Gambar 2.1: Penggunaan if satu kasus

Tampak bahwa kalau usia yang dimasukkan lebih dari atau sama dengan 17, program tidak menghasilkan apa-apa.

Pernyataan1 dilaksanakan jika dan hanya jika kondisi yang diinginkan terpenuhi, jika tidak, lakukan pernyataan2. Jika Anda tidak mempergunakan pernyataan else program tidak akan error, namun jika anda mempergunakan pernyataan *else* tanpa didahului pernyataan *if*, maka program akan error. Jika pernyataan1 atau pernyataan2 hanya terdiri dari satu baris, maka tanda { } tidak diperlukan, namun jika lebih maka diperlukan.

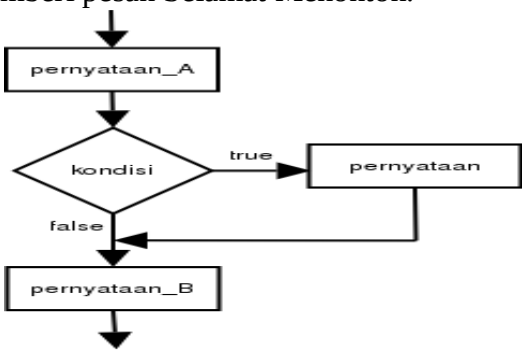
Bentuk Umumnya banyak kasus:

```
if (kondisi)
{

        pernyataan1;
        pernyataan1a;
        pernyataan1b;

}
else
{

        pernyataan2;
        pernyataan2a;
        pernyataan2b;

}
```

```
# include <iostream.h>

void main ()
{
   int usia;

   cout << "Berapa usia Anda ? ";
   cin >> usia;

   if (usia < 17)
      cout << "Anda tidak diperkenankan menonton" << endl;
   else
      cout << "Selamat Menonton" << endl;
}
```

Gambar 2.2: penggunaan if banyak kasus

Terlihat bahwa kalau usia yang dimasukkan lebih dari 17, program akan memberi pesan Selamat Menonton.



Gambar2.3: flowchart instruksi if secara umum

3. NESTED IF

Nested if atau if bersarang merupakan struktur if yang paling komplek, karena merupakan perluasan dan kombinasi dari berbagai struktur if lainnya. Konsep dari percabangan ini adalah terdapat Struktur If yang berada didalam Struktur If lainnya. Artinya dalam pernyataan If bersarang jika kondisi If yang paling luar (paling atas) bernilai benar, maka kondisi If yang berada didalamnya baru akan dilihat (di cek).

C. STRUKTUR KENDALI PENGULANGAN

1. Instruksi FOR

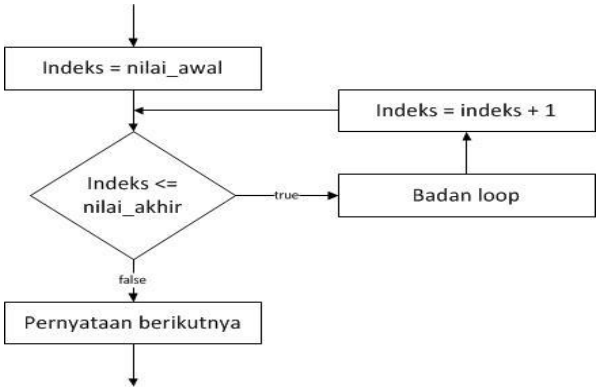
Instruksi ini digunakan apabila kita tahu secara pasti banyaknya perulangan yang akan dilakukan. Pernyataan FOR mempunyai 3 parameter:

a. Nilai awal (initial value)
b. Test kondisi yang menentukan akhir LOOP
c. Penentu perubahan nilai.

Bentuk umum pseudocode FOR format naik:

for **indeks=nilai_awal** to **nilai_akhir** do
<instruksi/blok instruksi>
endfor

Bentuk umum flowchart :



Gambar 1 Flowchart Instruksi FOR Format Naik

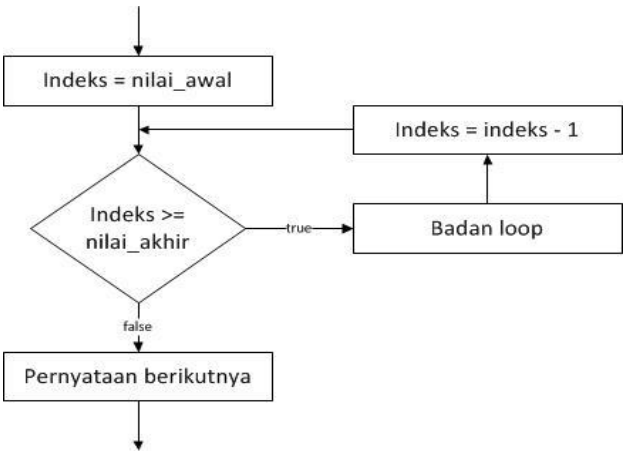
Cara kerjanya:

a. Indeks diassign dengan nilai awal.
b. Indeks dibandingkan dengan nilai akhir.
c. Jika indeks <= nilai akhir maka
    - Badan loop dikerjakan.
    - Secara otomatis nilai indeks ditambah 1.
    - Indeks dibandingkan dengan nilai akhir.
d. Jika indeks > nilai akhir maka akan dikerjakan statemen pertama sesudah "endfor" (badan loop).

Bentuk umum pseudocode FOR format turun:

for **indeks=nilai_awal down** to **nilai_akhir** do
        <instruksi/blok instruksi>
    endfor

Bentuk umum flowchart :



Gambar 2 Flowchart Instruksi FOR Format Turun

Cara kerjanya:

a. Indeks diassign dengan nilai awal.
b. Indeks dibandingkan dengan nilai akhir.
c. Jika indeks >= nilai akhir maka
    - Badan loop dikerjakan.
    - Secara otomatis nilai indeks dikurangi 1.

- Indeks dibandingkan dengan nilai akhir.

d. Jika indeks < nilai akhir maka akan dikerjakan statemen pertama sesudah "endfor" (badan loop).

Yang perlu diperhatikan di sini adalah penggunaan tanda titik koma (;) sebagai pemisah parameter dalam sintaks FOR, dalam sintaks tersebut memiliki :

a. Ekspresi inisialisasi yang memberi nilai awal pada variabel kontrol LOOP FOR dan for (initial value; condition_expr; incremental_expr) memberitahu program di mana memulai LOOP. Sebagai contoh a = 1 menginisialkan variabel kontrol a dengan 1 dan memulai LOOP pada a = 1.

b. Ekspresi kondisi. Bila test kondisi bernilai salah maka LOOP akan berhenti. Sebagai contoh, a <= 3 program akan melakukan test apakah variabel a lebih kecil atau sama dengan 3, jika a lebih besar dari 3 maka LOOP akan dihentikan.

c. Ekspresi perubahan nilai yang berfungsi untuk menaikkan atau menurunkan nilai dari variabel kontrol. Ekspresi perubahan nilai ini dapat berupa nilai positif (penaikan) atau nilai negatif(penurunan). Operasi penaikan nilai dapat menggunakan operator ++ yang artinya setiap loop operator ++ akan menambah nilai 1 ke variabel kontrol. Sedangkan untuk penuruna nilai dapat menggunakan operator --.

2. Instruksi WHILE

Instruksi perulangan ini dapat digunakan apabila kita belum tahu secara pasti berapa kali banyaknya perulangan yang akan dilakukan.

Bentuk umum pseudocode:
while (kondisi) do
(instruksi/blok istruksi)
endwhile
Bentuk umum flowchart :
Gambar 3 Flowchart Instruksi WHILE

Cara kerjanya:

a. Sebelum masuk ke "while-loop" kondisi yang merupakan ekspresi boolean harus sudah mempunyai nilai.
b. Jika kondisi bernilai true maka seluruh badan loop dikerjakan.
c. Dicek kembali apakah kondisi bernilai true atau false. Jika kondisi bernilai true maka maka tidak ada perubahan, artinya kembali mengerjakan badan loop. Jika kondisi bernilai false maka langsung mengerjakan statemen pertama sesudah loop WHILE.
d. Looping berhenti setelah kondisi bernilai false, sehingga harus ada statemen yang mengakibatkan kondisi bernilai false. Tetapi jika kondisi

tetap true maka terjadi infinite true, artinya jika tidak ada statemen yang mengakibatkan kondisi bernilai false maka terjadi infinite loop.
Sintaks statemen WHILE:

```
while (condition_expr)
{ statemen-statemen }
```

3. Intruksi While Do

Konstruksi while digunakan untuk melakukan perulangan terhadap baris kode tertentu selama suatu kondisi terpenuhi. Jika kondisi sudah tidak terpenuhi, maka program akan keluar dari perulangan.

Penulisan perulangan di algoritma While - Do

Algoritma	contoh
While kondisimengulang do Aksi End	Password : string Write (‘masukkan password’) Read(password) While password < > ‘123’ do Write (‘password salah ‘) Write(‘masukkan password’) Read(password) End

D. KESIMPULAN

Struktur Kendali Pemilihan adalah suatu struktur dasar algoritma yang memiliki satu atau lebih kondisi tertentu dimana sebuah instruksi dilaksanakan jika sebuah kondisi/persyaratan terpenuhi. Dalam membuat program, pengambilan keputusan seringkali dimanfaatkan.

Beberapa struktur dapat dipakai untuk menyelesaikan masalah yang sama tetapi ada struktur perulangan yang hanya cocok dipakai untuk masalah tertentu. Macam-macam struktur perulangan :

- Instruksi FOR
- Instruksi WHILE
  Instruksi WHILE-DO

E. SOAL LATIHAN

1. Buatlah program untuk memeriksa apakah pemasukan lebih besar / kecil dari pengeluaran ?
2. Buatlah program untuk memeriksa apakah suatu bilangan habis dibagi 5 dan habis dibagi 3 sekaligus atau tidak ?
3. Buatlah program untuk mencari bilangan terkecil dari 4 bilangan yang diinputkan ?
4. Buatlah konversi nilai huruf dari nilai angka yang diinputkan user !
   a. 80-100
   b. 65-79

BAB 5: FUNCTION BAHASA C

A. KONSEP *FUNCTION*

Fungsi adalah kumpulan pernyataan yang melakukan tugas tertentu. Sejauh ini fungsi digunakan dengan dua cara :

1. Fungsi utama yaitu fungsi yang ada dalam setiap program yang telah ditulis
2. Fungsi library disebut juga dengan “sqrt” dan “pow”

Fungsi dapat juga digunakan seperti fungsi library pada C++. Salah satu alasan fungsi digunakan untuk memecahkan program kedalam sebuah program yang lebih kecil sehingga mudah dikelola. Setiap unit modul, deprogram sebagai fungsi terpisah. Misalnya pada sebuah buku yang memiliki seribu halaman,tetapi tidak dibagi kedalam bab atau bagian. Jika ingin mencoba untuk menemukan satu topic dalam buku ini akan sangat sulit. Real-world program dapat dengan mudah ada ribuan basis kode, dan kecuali modularized. Fungsi juga digunakan untuk menyederhanakan program. Jika tugas tertentu dilakukan di beberapa tempat sebuah program, sebuah fungsi dapat ditulis sekali saja untuk melakukan tugas itu, dan kemudian akan dijalankan kepan saja dibutuhkan.

Ketika membuat sebuah fungsi yang harus dilakukan adalah menulis defenisi. Semua defenisi mempunyai bagian – bagian di bawah.

a. Name
b. Parameter list
c. Body
d. Return type.

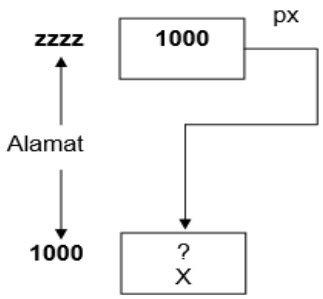
B. VARIABEL LOKAL, GLOBAL, DAN POINTER

Variabel lokal atau variabel internal, artinya variabel yang hanya dikenali secara lokal dalam sebuah subprogram (fungsi atau prosedur). Variabel lokal tidak dapat dipanggil, diakses dan diubah oleh prosedur atau fungsi yang lain, bahkan oleh program utama sekalipun. karena hanya dapat dikenali oleh prosedur atau fungsi dimana variabel ini didefinisikan. Lingkup dari variable local terbatas. Hanya berlaku dimana variable tersebut dideklarasikan. Jika dideklarasikan diawal fungsi (seperti dalam main) maka lingkup dari variable tersebut adalah untuk seluruh fungsi main. Variabel lokal dapat dikatakan sebagai variable yang aman dan tersembunyi dari fungsi lain, tetapi variable ini tidak menyediakan cara yang mudah untuk berbagi data.

Variabel Global adalah variabel yang didefinisikan dalam program utama dan dapat digunakan di program utama maupun sub-sub program lainnya. Nilai dari variabel ini dapat dipanggil, diakses dan diubah oleh prosedur atau fungsi apapun yang terdapat dalam program tersebut. Variabel global dapat digunakan untuk setiap bagian dari program, maupun fungsi, walaupun

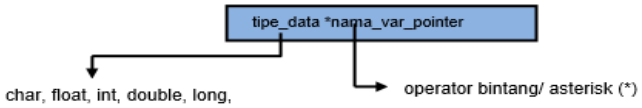
dideklarasikan diakhir program. Jika sejumlah besar data harus dapat diakses oleh semua fungsi dalam program, variabel global dapat digunakan sebagai alternatif mudah. Sebuah variabel global merupakan variabel yang ditetapkan diluar oleh semua fungsi dalam sebuah program.

Suatu variable yang bertipe pointer (variabel pointer) tidak berisi data, melainkan berisi alamat suatu data. Variabel pointer sering dikatakan sebagai variabel yang menunjuk ke obyek lain. Pada kenyataan yang sebenarnya, variabel pointer berisi alamat dari suatu obyek lain (yaitu obyek yang dikatakan ditunjuk oleh pointer). Sebagai contoh, px adalah variable pointer dan x adalah variabel yang ditunjuk oleh px. Kalau x berada pada alamat memori (alamat awal) 1000, maka px akan berisi 1000. Sebagaimana diilustrasikan pada gambar di bawah ini:



Gambar 13.4. Variabel pointer px menunjuk ke variabel x

Suatu variabel pointer dideklarasikan dengan bentuk sebagai berikut :



C. MENDEKLARASIKAN DAN PENGIRIMAN DATA PADA FUNGSI

Pertama kali program dijalankan, komputer akan mencari Fungsi main() dan melaksanakan instruksi- instruksi yang ada didalamnya. Suatu fungsi mempunyai **'judul'** yang minimal berisi Nama dan Tipe fungsi. Menulis **'judul'** sebuah Fungsi sebagai awal dari suatu Fungsi disebut men-**DEFINISIKAN** Fungsi. Bila fungsi ditulis dibawah fungsi main(), maka fungsi tersebut harus '**didaftarkan**' atau **DIDEKLARASIKAN** terlebih dahulu sebelum dapat digunakan. Dimana pen-deklarasian ini ditulis sebelum program induk main(). Deklarasi fungsi diakhiri dengan titik koma. Tipe_data dapat berupa segala tipe data yang dikenal C ataupun tipe data buatan, namun tipe data dapat juga tidak ada dan digantikan dengan void yang berarti fungsi tersebut tidak mengembalikan nilai apapun. Nama fungsi adalah nama yang unik. Argumen dapat ada atau tidak (opsional) yang digunakan untuk menerima argumen/parameter. Antar argumen-argumen dipisahkan dengan menggunakan tanda koma.

Nilai-nilai yang akan dikirim pada sebuah fungsi disebut dengan argument (arguments). Programmer yang ahli biasanya sudah akrab dengan cara menggunakan argumen dalam pemanggilan fungsi. Pernyataan berikut ini

merupakan fungsi pow dengan dua argumen yang sedang melakukan panggilan 2 dan 4.

```
result = pow(2, 4);
```

Sebuah parameter merupakan variabel khusus yang menangani nilai yang dilewatkan sebagai argumen menuju sebuah fungsi. Dengan menggunakan parameter, kita dapat merancang sendiri fungsi yang dapat menerima data. Program berikut merupakan definisi sebuah fungsi yang menggunakan parameter:

```
void TampilNilai(int num)

{ cout << "Nilainya adalah " << num << endl;  }
```

Perhatikan integer num yng digunakan untuk mendefinisikan variabel yang berada dalam kurung (int num). Variabel num adalah parameter. Integer Ini digunakan untuk membuat fungsi Tampil Nilai supaya menerima nilai integer sebagai argumen. Program 7.6 dibawah merupakan contoh yang menggunakan fungsi TampilNilai. Perhatikan program dibawah ini:

```
#include <conio.h>
#include <iostream>
using namespace std;
//Prototipe Fungsi  void TampilNilai(int);
int main()
{cout << "Saya sedang memasukan 5 ke fungsi TampilNilai.\n";
 TampilNilai(5);  // Call TampilNilai dengan argument 5
 cout << "Sekarang saya sudah kembali ke program utama.\n";
 getch(); return 0;  }
void TampilNilai(int num)
{cout << "Besar nilainya adalah: " << num << endl;  }
```

Keluaran programnya adalah sebagai berikut:

- Saya sedang memasukan 5 ke fungsi TampilNilai.
- Besar nilainya adalah: 5
- Sekarang saya sudah kembali ke program utama.

Dalam prototype fungsi tersebut diatas yang perlu diperhatikan adalah pada Tampil Nilai:

```
void TampilNilai(int);   // function prototype
```

Dalam fungsi tersebut kita tidak perlu memasukan daftar parameter yang merupakan variabel dalam tanda kurung, dan hanya dengan memasukan type data yang diperlukan saja. Fungsi ini dapat prototype ini juga dapat ditulis sebagai berikut:

```
void TampilNilai(int num);
```

Fungsi tersebut diatas sangat mudah digunakan walaupun, compiler mengabaikan nama variabel pada parameter fungsi prototype tersebut. Dalam program utama, fungsi TampiNilai disebut dengan argumen 5, dimana hal

tersebut berada dalam tanda kurung. Nomor 5 dimasukan kedalam variabel num, yang mana num merupakan parameter TampilNilai.

D. REKURSIF

Rekursif adalah salah satu metode dalam dunia matematika dimana definisi sebuah fungsi mengandung fungsi itu sendiri. Dalam dunia pemrograman, rekursi diimplementasikan dalam sebuah fungsi yang memanggil dirinya sendiri. Atau rekursif merupakan satu teknik pemrograman dengan cara memanggil sebuah fungsi dari dirinya sendiri, baik itu secara langsung maupun tidak langsung. Pemanggilan fungsi rekursif secara langsung berarti dalam fungsi tersebut terdapat statement untuk memanggil dirinya sendiri sedangkan secara tidak langsung berarti fungsi rekursif tersebut memanggil 1 atau lebih fungsi lain sebelum memanggil dirinya sendiri.

Fungsi rekursif tidak langsung merupakan realisasi fungsi yang dapat cross-recursif yaitu jika realisasi fungsi f mengandung fungsi g yang realisasinya adalah aplikasi terhadap f. Fungsi merupakan sub program yang sangat bermanfaat dalam pemrograman, terutama untuk program atau proyek yang besar. Manfaat penggunaan subprogram antara lain adalah :

1. Meningkatkan readibility, yaitu mempermudah pembacaan program.
2. Meningkatkan modularity, yaitu memecah sesuatu yang besar menjadi modul-modul atau bagianbagian yang lebih kecil sesuai dengan fungsinya, sehingga mempermudah pengecekan, testing dan lokalisasi kesalahan.
3. Meningkatkan reusability, yaitu suatu sub program dapat dipakai berulang kali dengan hanya memanggil sub program tersebut tanpa menuliskan perintahperintah yang semestinya diulangulang.

Sub Program Rekursif adalah sub program yang memanggil dirinya sendiri selama kondisi pemanggilan dipenuhi. Dengan melihat sifat sub program di atas maka sub program rekursif harus memiliki :

1. kondisi yang menyebabkan pemanggilan dirinya berhenti (disebut kondisi khusus atau special condition).
2. pemanggilan diri sub program (yaitu bila kondisi khusus tidak dipenuhi)

Sub program rekursif umumnya dipakai untuk permasalahan yang memiliki langkah penyelesaian yang terpola atau langkah-langkah yang teratur.

E. PERPUSTAKAAN FUNGSI

Standar library function merupakan suatu fungsi standart yang sudah disediakan oleh progam Bahasa C, dimana untuk menggunakannya, maka kita harus mencantumkan suatu header file dari fungsi tersebut yaitu dengan menggunakan perintah #include dalam suatu program. Ada beberapa penggunaan library function adalah sebagai berikut: Library function yang berhubungan dengan penanganan tanggal dan waktu.

1. Dalam penanganan tanggal dan waktu pada Bahasa C mempunyai suatu pustaka sendiri, yaitu time.h. dimana fungsi tersebut memiliki beberapa kumpulan fungsi sebagai berikut:

Fungsi	Deskripsi
Difftime	Untuk menghitung perbedaan waktu yang terjadi dari dua nilai time_t
Time	Untuk mengembalikan suatu nilai waktu yang ada saat ini
Clock	Untuk mengembalikan suatu nilai perhitungan clock dari sebuah prosesor
Ctime	Untuk melakukan konversi nilai time_t ke representasi tekstual
Strftime	Digunakan untuk melakukan konversi suatu objek struct tm pada representasi tekstual
Wcsftime	Digunakan untuk melakukan konversi objek struct tm ke suatu representasi custom wide string
Gmtime	Digunakan untuk melakukan suatu konversi nilai time_t ke suatu ekspresi kalender menggunakan koordinat universal GMT

2. Library function yang berhubungan dengan fungsi matematika Library function matematika digunakan untuk operasi matematika. Library function yang digunakan dalam matematika adalah math.h. sehingga dengan penggunaan library ini, seorang programmer harus menuliskan suatu instruksi yaitu #include <math.h>. adapun beberapa fungsi yang digunakan dalam fungsi matematika ini adalah sebagai berikut:

Fungsi	Deskripsi
Fabs()	Digunakan untuk memberikan nilai kembalian yang berupa suatu nilai yang absolut
Ceil()	Digunakan untuk memperoleh suatu bilangan bulat yang paling kecil dimana nilai yang muncul tidak boleh kurang dari nilai argumennya
Floor()	Digunakan untuk memperoleh suatu bilangan bulat yang muncul tidal lebih besar dari nilai argumennya
Round()	Digunakan untuk memperoleh suatu nilai bilangan bulat yang terdekat dari nilai argumennya
Trunc()	Digunakan untuk memperoleh suatu nilai bilangan bulat dari argumennya
Pow()	Digunakan untuk menghitung perpangkatan
Pow10()	Digunakan untuk menghitung perpangkatan 10
Sqrt()	Digunakan untuk menghitung akar
Hypot()	Digunakan untuk menghitung sisi miring dari segitiga siku - siku
Log()	Digunakan untuk menghitung nilai algoritma

F. MACRO

Makro dalam ilmu komputer adalah aturan atau pola yang menentukan bagaimana urutan input tertentu (sering kali urutan karakter) harus dipetakan ke urutan output pengganti (juga sering merupakan urutan karakter) sesuai dengan prosedur yang ditentukan. Proses pemetaan yang instantiate (mengubah) penggunaan makro menjadi urutan tertentu dikenal sebagai ekspansi makro . Fasilitas untuk menulis makro dapat disediakan sebagai bagian dari aplikasi perangkat lunak atau sebagai bagian dari bahasa pemrograman . Dalam kasus sebelumnya, makro digunakan untuk membuat tugas menggunakan aplikasi kurang berulang. Dalam kasus terakhir, mereka adalah alat yang memungkinkan seorang programmer untuk mengaktifkan penggunaan kembali kode atau bahkan untuk merancang bahasa khusus domain .

G. KESIMPULAN

Fungsi dapat juga digunakan seperti fungsi library pada C++. Salah satu alasan fungsi digunakan untuk memecahkan program kedalam sebuah program yang lebih kecil sehingga mudah dikelola. Setiap unit modul, deprogram sebagai fungsi terpisah.

H. SOAL LATIHAN

1. Tulis function yang rekursif untuk menampilkan bilangan decimal dalam bentuk biner
2. Selain variable local dan global di dalam bahasa C juga terdapat static variable. Apa perbedaan variable statis dan variable global? Kapan variable statis cocok untuk digunakan?
3. Apa kekurangan program yang menggunakan variable global dibandingkan dengan pengiriman parameter?
4. Apa keluaran program berikut:

```
#include <stdio.h>
Int a,b;
Int main () {
        Int a, b;
        Print (“%d %d”, a, b);
}
```

5. Apa keluaran program berikut:

```
#include <stdio.h>
Int a=10, b=20;
Int main () {
        Int a=30, b=40;
        Print (“%d %d”, a, b);
}
```

BAB 6: ARRAY BAHASA C

A. KONSEP ARRAY

Dalam ilmu komputer, Array (larik) ialah penampung sejumlah data sejenis (homogen) yang menggunakan satu identifier (pengenal). Masing-masing elemen larik diakses menggunakan indeks (subscript) dari nol sampai n-1 (n menyatakan jumlah elemen larik). Pengolahan data larik harus per elemen. Elemen Larik dapat diakses langsung (acak), maksudnya untuk memanipulasi elemen ke-4 tidak harus melalui elemen ke-1, ke-2 dan ke-3. Berdasarkan banyaknya indeks larik dibagi menjadi satu dimensi dan multi dimensi (dua dimensi, tiga dimensi). Larik dapat diakses berdasarkan indeksnya. Indeks larik umumnya dimulai dari 0 dan ada pula yang dimulai dari angka bukan 0. Pengaksesan larik biasanya dibuat dengan menggunakan perulangan (looping).

B. DEKLARASI ARRAY

Variabel array dideklarasikan dengan mencantumkan tipe dan nama variabel yang diikuti dengan banyaknya lokasi memori yang ingin dibuat.

Contoh :

int c[5];

C++ secara otomatis akan menyediakan lokasi memori yang sesuai dengan yang dideklarasikan, dimana nomor indeks selalu dimulai dari 0.

int c $[5] = \{-12, 0, 20, 85, 1551\}$;

Nilai suatu variabel dapat juga diinisialisasi secaar langsung seperti yang terfapat di dalam tanda kurung kurawal pada saat deklarasi di atas.

int x$[5] = [0]$;

deklarasi variable; array sekaligus mengisi setiap lokasi memorinya dengan nilai 0.

C. LARIK SATU DIMENSI

Larik satu dimensi merupakan jenis larik dasar dan jenis larik yang sering digunakan, Pemakaian larik satu dimensi terutama dipakai dalam tipe data string (terutama dalam bahasa Bahasa Pemrograman C). Meskipun C dibuat untuk memprogram sistem dan jaringan komputer namun bahasa ini juga sering digunakan dalam mengembangkan software aplikasi. C juga banyak dipakai oleh berbagai jenis platform sistem operasi dan arsitektur komputer, bahkan terdapat beberapa compiler yang sangat populer telah tersedia. C

secara luar biasa memengaruhi bahasa populer lainnya, terutama C++ yang merupakan extensi dari C.

Bentuknya (Pseudocode) :

```
nama_var : array
```

Bentuknya (C++) :

```
Tipe nama_var          ;
```

Dengan :

tipe : menyatakan jenis elemen array (int, char dan lain-lain)

range indeks : menyatakan indeks awal sampai dengan indeks akhir dari var array yang dibuat

ukuran : menyatakan jumlah maksimal elemen array

contoh algoritma :

```
Algoritma PangkatDua

Deklarasi
  square : array [1..100] of integer
  i,k : integer

Deskripsi
  for i ← 1 to 10 do
     k ← i + 1
     square[i] ← k * k
     output("Pangkat dari ",k," adalah ", square[i])
  endfor
```

Program C++

```
#include<iostream.h>
main()
{
  int square[100];
  int i,k;

  for(i=0; i<10; i++)
  {
   k = i + 1;
   square[i] = k * k;
   cout<<"\nPangkat dari "<<k<<" adalah "<< square[i];
  }
}
```

Output :

```
Pangkat dari 1 adalah 1
Pangkat dari 2 adalah 4
Pangkat dari 3 adalah 9
Pangkat dari 4 adalah 16
Pangkat dari 5 adalah 25
Pangkat dari 6 adalah 36
```

```
Pangkat dari 7 adalah 49
Pangkat dari 8 adalah 64
Pangkat dari 9 adalah 81
Pangkat dari 10 adalah 100
```

D. LARIK DIMENSI DUA

Larik dua dimensi merupakan tipe larik yang lain. Larik dua dimensi sering dipakai untuk mempresentasikan tabel dan matriks da;am pemrograman. Larik dua dimensi terdiri dari m buah baris dan n buah kolom.

Deklarasi array (Pseudocode) :

```
nama_var : array [rangeindeks1,rangeindeks2] of tipe
```

Deklarasi array (C++):

```
tipe nama_var[ukuran1][ukuran2];
```

Contoh: Sebuah matrik A berukuran 2x3 dapat didklarasikan sebagai berikut :

Algoritma : a : array [1..2,1..3] of integer

$a_1 \leftarrow 11$

$a_2 \leftarrow 7$

$a_3 \leftarrow 4$

$a_4 \leftarrow 12$

C++ :

```
int a[2][3] = {{11, 7, 4},{12, 3, 9}};
```

yang akan menempati lokasi memori dengan susunan berikut :

	0	1	2
0	11	7	4
1	12	3	9

dan definisi variabel untuk setiap elemen tersebut adalah :

	0	1	2
0	a[0][0]	a[0][1]	a[0][2]
1	a[1][0]	a[1][1]	a[1][2]

Contoh Program 1 :

```
#include<iostream.h>
main()
{
  int A[3][3]={{1,2,3},{4,5,6},{7,8,9}};
  int i,j;

  for(i=0; i<3; i++)
  {
      for(j=0; j<3; j++)
        cout<<A[i][j]<<"  ";
      cout<<endl;
  }
}
```

Output :

```
1 2 3
4 5 6
7 8 9
```

Contoh Program 2 :

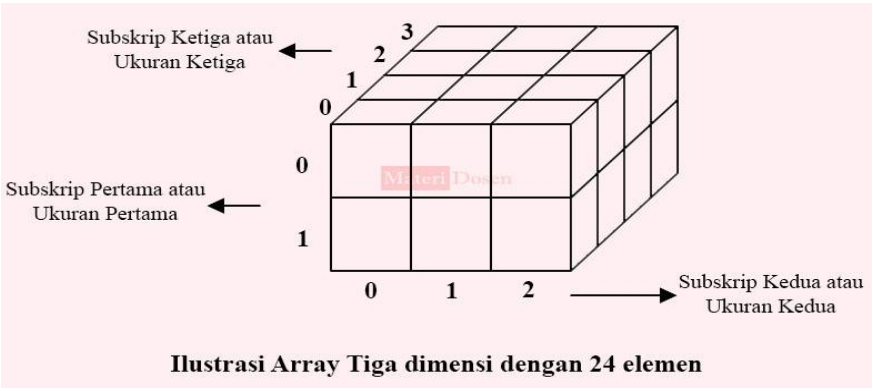
```
#include<stdio.h>
void printArray(int [][3]);
main()
{
      int matrik1 [2][3] = { {1, 2, 3}, {4, 5, 6}};
       int matrik2 [2][3] = { {1, 2, 3}, {4, 5}};
       int matrik3 [2][3] = { {1, 2}, {4} };
       printArray(matrik1);
       printArray(matrik2);
       printArray(matrik3);
       return 0;
}
void printArray(int a[][3])
{    int i, j;

       for(i=0; i<=1; i++)
            {  for(j=0; j<=2; j++)
                    printf("%d ", a[i][j]);
                 printf("\n");
              }
}
```

Output :

```
1 2 3
4 5 6
1 2 3
4 5 0
1 2 0
4 0 0
```

E. ARRAY TIGA DIMENSI

Array Multidimensi merupakan array yang serupa dengan array satu dimensi maupun array dua dimensi, namun array multidimensi dapat memiliki memori yang lebih besar. Biasanya array multidimensi digunakan untuk menyebut array dengan dimensi lebih dari dua atau array yang mempunyai lebih dari dua subskrip, seperti untuk menyebut array tiga dimensi, empat dimensi, lima dimensi dan seterusnya. Berikut gambar yang dapat mengilustrasikan sebuah array multidimensi, dimana pada gambar dibawah ini kami menggunakan array tiga dimensi sebagai contoh dari array multidimensi.



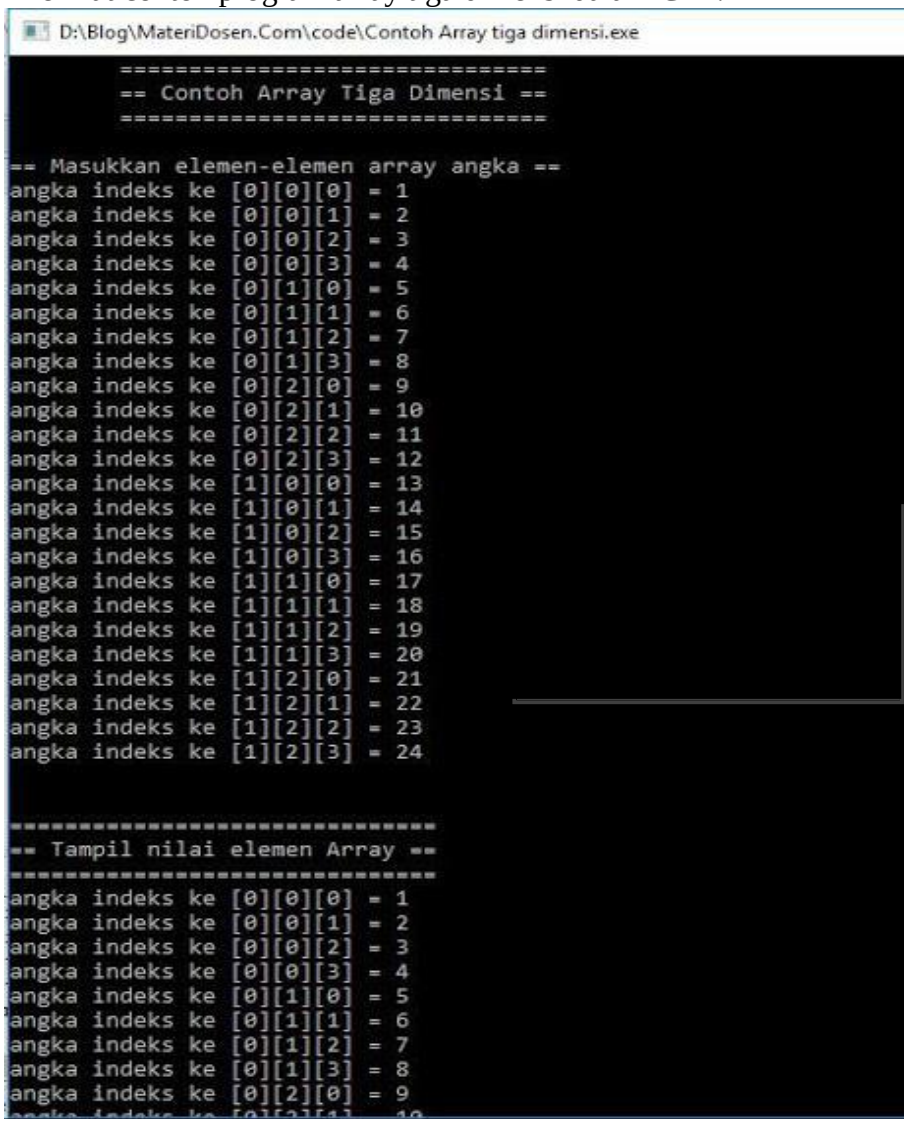
Ilustrasi Array Tiga dimensi dengan 24 elemen

Sumber: http://www.materidosen.com/2017/06/array-multidimensi-dimensi-c-lengkap.html

* Pada ilustrasi array tiga dimensi diatas, array tersebut memiliki besar subskrip pertama / besar ukuran pertama sebanyak 2, besar ukuran kedua sebanya 3 dan besar ukuran ketiga sebanyak 4.

Cara yang digunakan untuk mengakses elemen array multidimensi (dimisalkan array 3 dimensi) adalah dengan menuliskan indeks ukuran pertama / subskrip pertama, indeks ukuran kedua dan indeks ukuran ketiga.

Berikut contoh program array tiga dimensi dalam C++:



Sumber: http://www.materidosen.com/2017/06/array-multidimensi-dimensi-c-lengkap.html

F. MANFAAT LARIK

Array atau larik merupakan struktur data yang sering digunakan dalam pemrograman untuk menyimpan data yang akan diolah atau diproses seperti proses sorting. Array adalah struktur data yang terdiri dari kumpulan variabel yang bertipe sama. Bberapa bahasa pemrograman mendukung struktur array statis dan dinamis. Pada CC++ array yang didukung adalaharray statis. Pada CC++ adday merupakan pointer yang mempunyai alokasi memory tetap (pointeconstant). Array adalah suatu tipe data terstruktur yang berupa

sejumlah data sejenis (bertipe data sama) yang jumlahnya tetap dan diberi suatu nama tertentu. Elemen-elemen array tersusun secara sekuensial didalam memori sehingga memiliki alamat yang berdekatan. Adday dapat berupa array 1 dimensi, 2 dimensi, bahkan n-dimensi. Elemen-elemen array bertipe data sama tapi bisa bernilai sama atau berbeda-beda. Array yang digunakan untuk menyimpan data-data yang diinputkan masing-masing kedalam memory komputer. Jadi, jumlah datanya banyak namun satu jenis. Kegunaannya ialah untuk menyimpan dara yang cukup banyak namun memiliki tipe yang sama.

G. KESIMPULAN

Array (larik) ialah penampung sejumlah data sejenis (homogen) yang menggunakan satu identifier (pengenal). Masing-masing elemen larik diakses menggunakan indeks (subscript) dari nol sampai n-1 (n menyatakan jumlah elemen larik). Pengolahan data larik harus per elemen. Elemen Larik dapat diakses langsung (acak), maksudnya untuk memanipulasi elemen ke-4 tidak harus melalui elemen ke-1, ke-2 dan ke-3. Berdasarkan banyaknya indeks larik dibagi menjadi satu dimensi dan multi dimensi (duadimensi, tiga dimensi).

Array atau larik merupakan struktur data yang sering digunakan dalam pemrograman untuk menyimpan data yang akan diolah atau diproses seperti proses sorting. Array adalah struktur data yang terdiri dari kumpulan variabel yang bertipe sama. Bberapa bahasa pemrograman mendukung struktur array statis dan dinamis. Pada CC++ array yang didukung adalaharray statis. Pada CC++ adday merupakan pointer yang mempunyai alokasi memory tetap (pointeconstant). Array adalah suatu tipe data terstruktur yang berupa sejumlah data sejenis (bertipe data sama) yang jumlahnya tetap dan diberi suatu nama tertentu.

H. SOAL LATIHAN

1. Diketahui matriks A dan matriks B sebagai berikut :

$$A = \begin{pmatrix} 2 & 3 & 6 \\ 4 & 7 & 8 \end{pmatrix} \quad B = \begin{pmatrix} 3 & 5 \\ 2 & 4 \\ 8 & 6 \end{pmatrix}$$

   Buatlah program untuk menghitung matriks C=matriks A*matriks B

2. Diketahui matriks $A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$, buatlah program untuk menghitung matriks B = matriks A^3!
3. Diketahui matriks $A = \begin{bmatrix} 8 & -5 \\ 3 & -2 \end{bmatrix}$, $B = \begin{bmatrix} x & 2 \\ 3 & 2 \end{bmatrix}$, buatlah program untuk menghitung matriks C = matriks A + matriks B!

BAB 7: *STRUCTURE* BAHASA C

A. KONSEP *STRUCTURE*

Struktur merupakan kumpulan dari beberapa variable dengan beragam tipe data yang dikemas dalam satu variable. Struktur dapat kita gunakan untuk menyimpan banyak data. Struktur membantu data-data yang rumit khususnya dalam program yang besar karena struktur memberikan kelompok variable yang diperlukan dalam satu unit.

B. RECORD

Record dapat dikatakan sebagai suatu kumpulan data item yang masing-masing mempunyai jenis data berbeda. Data item yang merupakan elemen record biasanya disebut dengan FIELD.

1.1. Cara Mendeklarasikan Record

Bentuk umum deklarasi suatu variabel berjenis record adalah sbb :

```
TYPE identifier = RECORD
                    Nama_field_1 : jenis;
                    Nama_field_2 : jenis;
                    ……..
                    ……..
                    nama_field_n : jenis;
   END;
```

Contoh :

```
1.  VAR nilai : RECORD
              Nilai_1 : integer;
              Nilai_2 : integer;
    END
2.  TYPE date = RECORD
                  Tanggal : 1..31;
                  Bulan   : 1…12;
                  Tahun   : 1900..2000;
   END;
VAR event1,event2 : ARRAY [1..10] OF date;
3. TYPE  account = RECORD
                  cust_no      : integer;
                  cust_type    : char;
                  cust_balance : real;
END;
VAR customer : account;
```

1.2. Memroses Variabel Jenis Record

Perhatikan deklarasi variabel berikut :

```
TYPE nilai : RECORD
Nilai1 : real;
Nilai2 : real;
END;
VAR x,y : nilai;
```

Untuk memproses variabel x dan / atau y dilakukan dengan cara menyebutkan field designatornya, yg terdiri dari atas :

Nama_record**.**nama_field

Pada deklarasi diatas yang dimaksud dengan field designator-nya adalah :

x**.**nilai1
x**.**nilai2
y**.**nilai1
y**.**nilai2

Jadi jika ingin membaca variabel x atau y atau keduanya, maka bentuk statement-nya adalah :

READ (x.nilai1, x.nilai2, y.nilai1, y.nilai2);

Selanjutnya, misal ingin dibuat program sederhana untuk menjumlahkan dua bilangan kompleks a dan b yang hasilnya disimpan di c.

Secara aljabar penjumlahan dua bilangan kompleks adalah sebagai berikut :

```
a = x1 + iy1
b = x2 + iy2      +
c = (x1 + x2 ) + I(y1 + y2)
```

Maka bentuk garis besar programnya adalah sebagai berikut :

```
Program contoh ;
Type bk = record
        Bag_nyata       : integer;
        Bag_imajiner    : integer;
                End;
         Var a,b,c : bk;
Begin
  Read (a.bag_nyata, a.bag_imajiner,
  b.bag_nyata, b.bag_imajiner); c.bag_nyata
  := a.bag_nyata + b.bag _imajiner;
  c.bag_imajiner := a.bag_imajiner +
  b.bag_imajiner;
  writeln(c.bag_nyata,'    +','i',c.bag_imajiner);
  End.
```

C. PENDEKLARASIAN TIPE DATA

2.1 Tipe Data

Tipe data menentukan nilai yang dapat disimpan variabel tersebut dan operator-operator apa saja yang dapat dikenakan padanya. Suatu tipe menyatakan pola penyajian data dalam komputer. Mendefinisikan tipe data berarti:

1. menentukan nama tipe data itu
2. mendefinisikan domain nilai yang dapat dimiliki
3. perjanjian tentang cara menulis tetapan bertipe tersebut
4. operator yang dapat dioperasikan terhadap data bertipe tersebut

Tipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap perintah yang akan dilaksanakan oleh komputer. Sebagai contoh, variabel **x** yang bertipe data integer mempunyai nilai 13 akan dibagi oleh **y** yang bernilai 4 yang bertipe data integer pula, maka hasil pembagiannya adalah 4. Mengapa 13 dibagi 4 bukan 3.25? Karena variabel **x** dan **y** keduanya bertipe integer maka hasil operasinya akan menghasilkan nilai yang bertipe integer pula, namun jika keduanya bertipe float maka akan menghasilkan nilai pecahan yaitu 3.25.

Tipe data secara umum dapat dikelompokan menjadi dua macam:

1. Tipe dasar / sederhana :
   a) Bilangan bulat (*integer*)

      Tipe bilangan bulat adalah tipe yang memiliki keterurutan. Ini artinya, bila sebuah nilai bilangan bulat diketahui, nilai sebelumnya (*predecessor*) dan nilai sesudahnya (successor) dapat ditentukan. Contohnya, *predecessor* dari 8 adalah 7, sedangkan *successor*-nya adalah 9. Konstanta untuk nilai bertipe bilangan bulat harus ditulis tanpa mengandung titik desimal.

   b) Bilangan pecahan (*floating point*)

      Bilangan pecahan atau bilangan *real* adalah bilangan yang mengandung pecahan desimal, misalnya 3.65, 0.0003, 2.60240000E-6, .24, dan lain-lain.

      Contoh pendeklarasian bilangan pecahan dalam Bahasa C:

      float y = 2.37;

      float x =1.23E2

      keterangan: 1.23E2 dapat dibaca $1.23 * 10^2 = 123$

   c) Karakter (*character*)

      Tidak seperti tipe bilangan bulat yang digunakan untuk menyimpan data numeris, tipe karakter digunakan untuk menyimpan data alfanumeris, seperti A, Z, @, $, 1, 9, &, *, dan semua kode ASCII. Pemberian nilai untuk tipe data karakter harus diberi tanda petik tunggal (berada diantara tanda petik tunggal).Tipe data ini hanya dapat menampung satu karakter. Seperti halnya pada tipe bilangan bulat, tipe karakter juga mempunyai ketentuan (*successor* dan *predecessor*) yang

ditentukan oleh cara pengkodeannya di dalam komputer, misalnya pengkodean ASCII.

d) Bilangan logika (*boolean*)

Bilangan logika hanya mengenal dua buah nilai: benar (true) atau salah (false). Istilah "bilangan" pada "bilangan logika" muncul karena nilai "benar" dan " salah" dapat dinyatakan dengan angka 1 dan 0 (atau sebaliknya tergantung konvensi yang digunakan). Karena jangkauan nilai tipe boolean hanya beranggotakan dua buah nilai, maka konstanta (constant) atau tetapan yang terdapat pada tipe ini adalah true dan false

2. Tipe bentukan :

a) Larik (*array*)

Tipe larik memungkinkan pembuat program dapat mendeklarasikan kumpulan variabel yang bertipe sama (homogen). Karena larik dapat menyimpan lebih dari satu nilai dengan nama dan tipe yang sama maka untuk membedakan antara data satu dengan data lainnya digunakan indeks, pada bahasa C indeks dimulai dari 0. larik selain satu dimensi, juga dapat dideklarasikan sebagai larik multidimensi.

b) Pointer

Pointer adalah variabel yang menunjukkan lokasi memori (alamat dari suatu data yang disimpan) tertentu.

c) String

Tipe string digunakan untuk menyimpan data yang berupa untaian karakter dengan panjang tertentu. Tipe string sebenarnya merupakan pengembangan tipe karakter.

d) Rekaman (struct)

Tipe data rekaman (dalam bahasa C dinamakan struct) adalah suatu struktur data yang menggabungkan beberapa data yang mempunyai tipe data yang berbeda (heterogen).

Tipe data dalam Bahasa C disediakan lima macam tipe data dasar, yaitu:

No	Tipe Data	Ukuran	Range (jangkauan)`	Format	Keterangan
1	Char	1 byte	- 128 s/d 127	%c	Karakter/string
2	Int	2 byte	- 32768 s/d 32767	%i , %d	Integer/bilangan bulat
3	Float	4 byte	- 3.4E-38 s/d 3.4E+38	%f	Float/bilangan pecahan
4	Double	8 byte	- 1.7E-308 s/d 1.7+308	%lf	Pecahan presisi ganda
5	Void	0 byte	-	-	Tidak bertipe

Tipe data erat hubungannya dengan konstanta, variabel dan bentuk format penampilan data dalam program. Konstanta yang digunakan oleh program juga dapat ditentukan sesuai dengan tipe dari datanya, apakah termasuk konstanta numerik (integer atau float), konstanta karakter ataukah konstanta string.

2.2. Konstanta

Konstanta adalah suatu nilai yang tetap (tidak berubah) dalam sebuah program. Contoh sebuah konstanta adalah saat kita menetukan sebuah tetapan dengan sebuah nilai, misalnya kita tetapkan pi adalah bernilai 3.14, dalam Bahasa C penetapan konstanta tersebut dapat dilakukan dengan menggunakan preprocessor directive #define.

1. Konstanta numerik integer (*integer constant*)

   Konstanta numerik integer adalah sebuah bilangan bulat. Konstanta numerik integer dapat berupa bilangan basis 10 (desimal) yaitu menggunakan digit 0 sampai dengan 9, basis 8 (oktal) menggunakan digit 0 sampai dengan 7, basis 16 (heksadesimal) menggunakan digit 0 sampai dengan 9 dan ditambah huruf A atau a sampai dengan F atau f (huruf A sebagai pengganti 10 sampai dengan F sebagai pengganti 15). Konstanta oktal dengan penulisan diawali "0" (nol).

   Konstanta heksa desimal diawali dengan "0x" (nol dan x). Nilai konstanta heksadesimal dan nilai konstanta oktal tidak dapat bernilai negatif dan tidak dapat bernilai pecahan. Berikut ini contoh konstanta numerik ineteger: 23, -6, 0x7b (bernilai 123 dalam decimal), 015 (bernilai 13 dalam desimal).

2. Konstanta numerik pecahan (*floating-point constant*)

   Konstanta numerik pecahan adalah sebuah bilangan pecahan. Tanda desimal dalam bahasa C adalah tanda titik . Konstanta numerik pecahan dapat bernilai ketepatan tunggal (*float*), ketepatan ganda (*double*) atau ketepatan ganda panjang (*long double*). Konstanta numerik pecahan hanya dapat ditulis dalam bentuk nilai desimal. Berikut ini contoh konstanta numerik pecahan: 3.14, 123. (ketepatan tunggal), 23.65E-78 (ketepatan ganda), 1.23E-7890 (ketepatan ganda panjang).

3. Konstanta karakter (*character constant*) dan karakter string (*string constant*)

   Konstanta karakter adalah merupakan nilai sebuah karakter yang ditulis di antara tanda petik tunggal. Konstanta string merupakan nilai sebuah karakter atau lebih yang ditulis di antara petik ganda. Konstanta string sering pula disebut dengan literal string. Berikut ini adalah contoh konstanta karakter dan karakter string:

   'Z' (konstanta karakter huruf Z bernilai ASCII 90).

   "Algoritma" (konstanta string).

4. Konstanta karakter escape (*escape sequence*)

   Konstanta karakter escape adalah karakter yang diawali dengan tanda *backslash* (\), konstanta karakter escape banyak digunakan di statemen-statemen untuk menampilkan hasil, misalnya untuk menempatkan kursor di baris berikutnya.

2.3 Variabel

Variabel adalah suatu pengenal yang digunkan untuk mewakili suatu nilai tertentu di dalam program. Nama variabel dapat dibuat sendiri oleh pemrogram dengan ketentuan penamaan.

Semua variabel yang akan digunakan di dalam program C harus dideklarasikan terlebih dahulu. Deklarasi variabel dimaksudkan untuk memberitahukan compiler C nama variabel dan tipe data yang digunakan sehingga bahasa C dapat mempersiapkan tempat variabel tersebut diletakkan di memori. Bentuk umu pendeklarasian variabel dalam Bahasa C adalah :

Tipedata nama_variabel;

Contoh:

int n1;//mendeklarasikan n1 yang dapat menampung nilai integer (bilangan bulat) float x = 12.3;//mendeklarasikan **x** yang bertipe float dan diberikan nilai awal 12.3 char kar = 'X'//mendeklarasikan variabe l**kar** bertipe karakter, yang berarti dapat menampung Nilai karakter, variabel **kar** diberi nilai awal **X**.

D. STRUKTUR BERTINGKAT

Struktur dapat dideklarasikan secara bertingkat. Yang berarti bahwa salah satu elemen suatu struktur merupakan struktur juga. Untuk struktur bertingkat kita memerlukan dua buah operator titik. Jika suatu elemen dari struktur dapat diakses seperti berikut:

nama_variabel.nama_elemen;

Maka sub-elemen dari struktur dapat diakses dengan menuliskan seperti berikut:

nama_variabel.nama_elemen.nama_sub_elemen;

```
#include<stdio.h>
void main()
{
structtanggal{
unsignedinthari;
unsignedintbulan;
unsignedinttahun;
};
structalamat{
charjalan[30];
charkota[20];
};
struct{
charnama[40];
structtanggaltgl_masuk;
structalamatrumah;
floatgaji;
```

```
}pegawai={"Nana Sutrisna", 16, 12, 1985,
"Joglo Raya No.52", "Jakarta Barat", 2500000};
printf("NamaPegawai = %s", pegawai.nama);
printf("\nTanggalmasuk = %d-%d-%d",
pegawai.tgl_masuk.hari,
pegawai.tgl_masuk, pegawai.tgl_masuk.tahun);
printf("\nAlamatrumah = Jalan %s, %s",
pegawai.rumah.jalan,
pegawai.rumah.kota);
printf("\nGajiPokok = %5.2f", pegawai.gaji);
}
```

Hasilnya:

```
NamaPegawai = Nana Sutrisna
Tanggalmasuk = 16-12-1985
Alamatrumah = JalanJoglo Raya No.52, Jakarta
Barat
GajiPokok = 2500000.00
```

E. *BIT FIELD*

Bit field adalah struktur data yang digunakan dalam pemrograman komputer . Terdiri dari sejumlah lokasi memori komputer yang berdekatan yang telah dialokasikan untuk menampung urutan bit , disimpan sehingga bit atau kelompok bit apa pun dalam himpunan dapat diatasi. Bidang bit paling umum digunakan untuk mewakili tipe integral dari lebar bit tetap yang diketahui. Arti dari bit individu dalam bidang ditentukan oleh programmer; misalnya, bit pertama dalam bidang bit (terletak di alamat dasar bidang) kadang-kadang digunakan untuk menentukan keadaan atribut tertentu yang terkait dengan bidang bit.

Untuk bahasa yang tidak memiliki bitfield asli, atau di mana programmer menginginkan kontrol ketat atas representasi bit yang dihasilkan, dimungkinkan untuk memanipulasi bit secara manual dalam jenis kata yang lebih besar.

Contoh

Bahasa pemrograman C

Mendeklarasikan bidang bit dalam C dan C ++ :

```
// buram dan tunjukkan
# tentukan YA 1
#define NO 0

// gaya garis
#define SOLID 1
#define DOTTED 2
#define DASHED 3
```

```
// warna primer
#define BLUE 4 / * 100 * /
#define GREEN 2 / * 010 * /
#define RED 1 / * 001 * /

// warna campuran
#define BLACK 0 / * 000 * /
#define YELLOW (RED | GREEN) / * 011 * /
#define MAGENTA (RED | BLUE) / * 101 * /
#define CYAN (GREEN | BLUE) / * 110 * /
#define WHITE (RED | GREEN | BLUE) / * 111 * /

const char * colors [ 8 ] = { "Hitam" , "Merah" , "Hijau" , "Kuning" ,
"Biru" , "Magenta" , "Cyan" , "Putih" };

// bit properti kotak bidang
struct BoxProps
{
  buram int unsigned : 1 ;
  uns_color int fill_color : 3 ;
  unsigned int : 4 ;  // isi hingga 8 bit
  show_border int unsigned : 1 ;
  unsigned int border_color : 3 ;
  unsigned int border_style : 2 ;
  unsigned char : 0 ;  // isi hingga byte terdekat (16 bit)
  unsigned char width : 4 , // Membagi byte menjadi 2 bidang 4 bit
            tinggi : 4 ;
};
```

Tata letak bidang bit dalam struktur C ditentukan oleh implementasi . Untuk perilaku yang tetap dapat diprediksi di kompiler, mungkin lebih baik untuk meniru bidang bit dengan operator bit dan primitif:

```
 / * Masing-masing arahan preprocessor ini mendefinisikan sedikit,
sesuai dengan satu tombol pada controller.  Urutan tombol
cocok dengan Sistem Hiburan Nintendo.  * /
#define KEY_RIGHT (1 << 0) / * 00000001 * /
#define KEY_LEFT (1 << 1) / * 00000010 * /
#define KEY_DOWN (1 << 2) / * 00000100 * /
#define KEY_UP (1 << 3) / * 00001000 * /
#define KEY_START (1 << 4) / * 00010000 * /
#define KEY_SELECT (1 << 5) / * 00100000 * /
#define KEY_B (1 << 6) / * 01000000 * /
#define KEY_A (1 << 7) / * 10000000 * /
int gameControllerStatus = 0 ;
/ * Mengatur gameControllerStatus menggunakan OR * /
membatalkan KeyPressed ( int key ) { gameControllerStatus | = key ;  }
```

```
/ * Mematikan kunci di gameControllerStatus menggunakan AND dan ~ (binary NOT) * /
membatalkan KeyReleased ( kunci int ) { gameControllerStatus & = ~ kunci ;  }
/ * Menguji apakah bit diatur menggunakan AND * /
int IsPressed ( int key ) { return gameControllerStatus & key ;  }
```

F. UNION DAN ENUMERATION

Sama seperti struct, union juga merupakan tipe data yang dibangkitkan, dimana anggota-anggotanya menggunakan secara bersama-sama ruang penyimpanan memori yang sama, berbeda dengan struktur yang masing-masing variabel menempati lokasi memori yang berbeda. Jumlah bytes yang digunakan untuk menyimpan union adalah sedikitnya cukup untuk menyimpan data terbesar yang ditangani. Oleh karena itu, tipe union ini umumnya digunakan utuk menangani satu, dua, atau tiga variabel dengan tipe yang mirip. Sebagai contoh:

```
union nilaiUjian {
int uts, uas;
float akhir;
}
```

Inisialisasi, deklarasi, dan pengolahan terhadap tipe union ini sama dengan struct yang telah dijelaskan pada bagian sebelumnya.

Sistem angka (enumeration) adalah sistem penulisan untuk mengekspresikan angka; yaitu, notasi matematika untuk mewakili angka dari himpunan yang diberikan, menggunakan angka atau simbol lainnya secara konsisten. Urutan simbol yang sama dapat mewakili angka yang berbeda dalam sistem angka yang berbeda. Misalnya, "11" mewakili angka *sebelas* dalam sistem angka desimal (digunakan dalam kehidupan bersama), angka *tiga* dalam sistem angka biner (digunakan dalam komputer), dan nomor dua dalam sistem angka unary (misalnya digunakan dalam penghitungan angka skor). Angka yang dilambangkan dengan angka disebut nilainya.

Dalam bidang ilmu komputer tertentu, sistem posisikan dasar *k* *yang* dimodifikasi digunakan, yang disebut penomoran bijektif , dengan angka 1, 2, ..., *k* ($k \geq 1$), dan nol diwakili oleh string kosong. Ini menetapkan suatu penghilangan antara himpunan semua digit-string seperti itu dan himpunan bilangan bulat non-negatif, menghindari non-keunikan yang disebabkan oleh nol di depan. Bilangan basis- *k* bijektif juga disebut notasi *k* -adic, jangan dikacaukan dengan angka-angka *p* -adik . Basa 1 objektif sama dengan unary.

Dalam basis *b* posisi posisi sistem *b* (dengan *b* bilangan asli lebih besar dari 1 dikenal sebagai radix), *b* simbol dasar (atau digit) yang sesuai dengan *b* pertama bilangan alami termasuk nol digunakan. Untuk menghasilkan sisa angka, posisi simbol pada gambar digunakan. Simbol di posisi terakhir memiliki nilainya sendiri, dan saat bergerak ke kiri nilainya dikalikan dengan *b* . Misalnya, dalam sistem desimal (basis 10), angka 4327 berarti $(4 \times 10^{3}) + (3 \times 10^{2}) + (2 \times 10^{1}) + (7 \times 10^{0})$, mencatat bahwa $10^{0} = 1$.

Secara umum, jika *b* adalah basis, seseorang menulis angka dalam sistem bilangan basis *b* dengan menyatakannya dalam bentuk $a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \ldots + a_0 b^0$ dan menulis digit yang disebutkan $a_n a_{n-1} a_{n-2} \ldots a_0$ dalam urutan menurun. Digit adalah bilangan alami antara 0 dan *b* - 1 , inklusif.

G. KESIMPULAN

Record dapat dikatakan sebagai suatu kumpulan data item yang masing-masing mempunyai jenis data berbeda..Data item yang merupakan elemen record biasanya disebut dengan FIELD. Tipe data merupakan bagian program yang paling penting karena tipe data mempengaruhi setiap perintah yang akan dilaksanakan oleh komputer.

Union juga merupakan tipe data yang dibangkitkan, dimana anggota-anggotanya menggunakan secara bersama-sama ruang penyimpanan memori yang sama, berbeda. Numeration istem angka (atau sistem angka) adalah sistem penulisan untuk mengekspresikan angka; yaitu, notasi matematika untuk mewakili angka dari himpunan yang diberikan, menggunakan angka atau simbol lainnya secara konsisten.

H. SOAL LATIHAN

1. Deklarasikan sebuah variable struktur (missalkan namanya – sample) yang didefinikan memiliki tipe struktur record!
2. Tampilkan ke layar (menggunkana fungsi *printf()*) string yang tersimpan dalam array word dari struktur sample!
3. Definisikan sebuah struktur (misalkan namanya - date) yang memiliki tiga *field* bertipe *int* (misalkan namanya – day, month dan year). Kemudian tuliskan potongan program untuk memasukkan lima buah tanggal yang disimpan dalam sebuah array struktur yang bertipe date!
4. Buatlah sebuah program untuk menghitung luas lingkaran, nilai jari-jari dimasukkan dari keyboard, sedangkan nilai muncul secara otomatis!
5. Buatlah program untuk membantu kasir swalayan untuk memisahkan pecahan uang kembalian menjadi Rp50.000,- , Rp20.000,- , Rp10.000,- , Rp5.000,- , Rp2.000,- , RP1000,- , Rp500,- , dan Rp100,-

BAB 8: BERKAS DATA BAHASA C

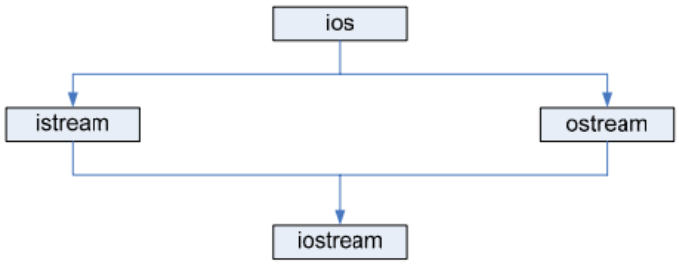
A. KONSEP BERKAS

Piranti lunak didalam computer disimpan dalam bentuk file (berkas). Sejumlah berkas dapat ditampung di dalam satu unit yang sama yaitu *folder* atau *directory*. Didalam dunia pengolahan data, berkas dibagi menjadi dua kelompok besar yaitu: berkas program (*program file*) dan berkas data (*data file*). Berdasarkan format data direkam ke dalam media penyimpanan sekunder, berkas dikelompokkan menjadi dua: berkas teks (*text file/* ASCII *file*) dan berkas biner (*binary file*). Berkas program yang kita ketik, disebut source kode, berformat teks. *Source code* di-*compile* menjadi *executable code*, berformat biner.

B. STEAM DAN FILE

File adalah sekumpulan data yang disimpan dalam media penyimpanan luar seperti disket/harddisk. Dalam C++ file adalah sebuah stream yang disimpan dalam media penyimpanan luar. Karena merupakan sebuah stream, operasi yang berlaku pada stream berlaku juga untuk file. Stream adalah suatu logika device yang menghasilkan dan menerima informasi atau wadah yang digunakan untuk menampung keluaran dan menampung aliran data.

Hirarki I/O class :



Penjelasan :

- ios adalah virtual base class untuk class istream dan ostream. Berisi fasilitas dasar untuk melakukan proses input/output. Dideklarasikan untuk pointer ke buffer untuk tempat penyimpanan data sementara.
- Istream (input stream) mendefinisikan fasilitas untuk melakukan input suatu informasi. Di dalamnya didefinisikan fungsi get ( ), getline ( ), extractor operator >>
- Ostream (output stream) mendefinisikan fasilitas untuk melakukan setting terhadap output.
- Iostream : berisi semua fasilitas dari ios, istream, ostream ditambah beberapa fungsi untuk menyempurnakan kerja dari fungsi yang dideklarasikan pada base class.

Untuk melakukan proses file I/O, diperlukan file header fstream.h didalam program. Dalam file fstream.h didefinisikan beberapa class/object yang berhubungan dengan pemrosesan file, yaitu : ifstream, ofstream, fstream yang diturunkan dari istream dan ostream.

Jika mendeklarasikan suatu stream, dapat menghubungkan stream tersebut dengan file, dimana proses ini berhubungan dengan operasi terhadap file.

Tiga proses utama dalam mengelola file adalah :

1. membuka file
2. melakukan proses terhadap file
3. menutup file

Sebelum membuka file, harus mengetahui keadaan mode filenya. Keadaan yang perlu diketahui adalah :

1. untuk membuka file dengan tujuan output, digunakan ofstream.
2. untuk membuka file dengan tujuan input, digunakan ifstream.
3. untuk membuka file dalam keadaan input maupun output, gunakan stream.

Fungsi open ( ) digunakan untuk membuka file. Bentuk umumnya adalah :

```
Void open (char* file_name, int mode, int access) ;
```

Keterangan :

- dengan fungsi open ( ), menhubungkan stream dengan file yang bernama file_name.
- nilai dari var mode akan menentukan bagaimana keadaan file jika dibuka.
- Variable access akan menentukan bagaimana metode pengaksesan terhadap file tersebut.

Nilai ini berhubungan dengan atribut file yang ada pada DOS. Nilai pada variable mode adalah :

1. ios : : app semua informasi yang ditulis ke dalam file (output) akan ditambahkan dibagian akhir file tersebut.
2. ios : : ate file akan dibuka dengan pointer file menunjuk pada akhir file.
3. ios : : in file akan dibuka sebagai input.
4. ios : : out file akan dibuka sebagai output
5. ios : : nocreate akan mengakibatkan kesalahan jika file tidak ada.
6. ios : : noreplace file yang dibuka tidak dapat diganti, atau mengakibatkan kesalahan jika file yang akan dibuka sudah ada.
7. ios : : trunc menyebabkan isi dari file yang sudah ada akan hilang

Nilai pada variable access adalah :

0 : normal file
1 : read only file
2 : hidden file
4 : system file
8 : archive bit-set file

Contoh penggunaan open ( ) :

Akan dibuka sebuah file sebagai output, dimana atribut dari file tersbut adalah read only dan nama file tersebut adalah tes.

Jawab :
Ofstream fout ;
Fout.open (“tes”, ios : : out, 1) ;

Jika dalam pembukaan file terdapat kesalahan, maka fungsi open ( ) akan mengembalikan nilai NULL.

Fungsi close ( ) untuk menutup file yang telah dibuka.
Bentuk umumnya :
Void close ( ) ;
Contoh penggunaan close ( ) :
Fout.close ( ) ;
Fin.close ( ) ;
Fio. Close ( ) ;

Didalam class ios terdapat pendefinisian fungsi : int eof ( ) digunakan untuk menentukan apakah pointer pada file telah mencapai akhir darti file tersebut.

Dalam class ios ada 4 (empat) buah fungsi untuk melakukan tes terhadap error yang terjadi didalam stream. Fungsi tersebut adalah :

1. int good ( )
2. int eof ( )
3. int bad ( )
4. int fail ( )

Keempat fungsi itu akan menghasilkan nilai 0 (nol) jika kondisi tersebut salah dan nilai bukan nol jika kondisi tersebut benar. Jika error terjadi, maka stream harus dibersihkan dahulu dari kesalahan sebelum melanjutkan ke proses berikutnya. Fungsi yang digunakan untuk membersihkan kesalahan pada stream jika terjadi kesalahan : clear ( )

Bentuk umumnya : Void clear (int flags = 0) ;

Fungsi clear ( ) dalam keadaan default akan membersihkan seluruh flag. Dapat ditentukan flag mana yang akan dibersihkan, dengan memberi argument yang sesuai.

Fungsi int rdstate ( ) digunakan untuk menentukan jenis kesalahan yang terjadi, dengan mengembalikan nilai integer kesalahan tersebut. Nilai enum yang dikembalikan oleh fungsi rdstate ( ) adalah :

1. goodbit
   0 jika tidak ada error dan 1 jika terjadi error
1. eofbit
   1 jika eof ditemukan dan 0 jika eof tidak ditemukan
2. failbit
   1 jika non fatal error ditemukan dan 0 jika tidak
3. badbit
   1 jika fatal error ditemukan dan 0 jika tidak.

Input/output pada file binary digunakan fungsi get ( ), put ( )

Bentuk umum: I stream & putback (char ch) ;

C. FILE I/O

Nama berkas yang digunakan di dalam program(logicalfile name) harus diasosiasikan dengan nama berkas pada media penyimpanan sekunder (physical file name). sebuah berkas harus di aktifkan terlebih dahulu sebelum data dibaca darinya atau ditulis ke dalamnya. Setelah selesai digunakan berkas harus ditutup.

a) Mengaktifkan Berkas

Berkas data diaktifkan dengan mengunakan instruksi topen(), intruksi topen() mengasosiasikan nama berkas (physical file) dengan buffer area, dan mengembalikan alamat awal buffer area. Jika proses ini gagal, misalnya berkas yang dimaksud secara fisik belum ada atau tidak ditemukan, maka function ini akan megembalikan NULL.

FILE*topen (const char*filename,const char*mode);

Mode pasa sintaks topen() adalah modus mengaktofkan berkas, apakah berkas di aktifkan untuk dibaca (“r”,read),ditulisi(“w”,Write). Ditambahkan pada ujung berkas (“a”,append),dan diubah (“r+”,”w+”,”a+’). Tabel berikut menampilkan sebelum seluruh modus yang dikenal. Berkas yang diaktifkan dengan modus “r” harus sudah ada , jika tidak demikian maka akan menimbulkan kesalahan.

Mode	Kegunaan
“r”	Buka berkas yang telah ada untuk dibaca
“w”	Buku berkas baru unyuk ditulisi
“a”	Buku berkas yang telah ada untuk ditambah record
“r+’	Buku berkas yang telag ada untuk dibaca dan ditulis
“w+”	Buku berkas baru untuk ditulis dan dibaca
“a+”	Buku berkas yang telah ada untuk dibaca dan ditambah record

b) Menutup berkas

Setelah berkas selesai digunakan maka harus ditutup dengan menggunakan intruksi fclose() atau fcliseall().

int **fclose**(FILE*stream);

instruksi dclose() menutup berkas stream tertentu. Apabila proses berhasil maka sikembalikan nol. Apabila terjadi kesalahan saat menutup berkas maka dikembalikan EOF.

int **fcloseall** (void);

int **_fcloseall (void);**

intsruksi fclose() menutup seluruh stream yang aktif kecuali stalin,stdout,stdpen,stderr, dan stdaux. Apabila proses berhasil maka dikembalikan sebuah bilangan yang menyatakan jumlah streamyang berhasil ditutup. Apabila terjadi kesalahan makan dikembalikan EOF,

intruksi _fcloseall() pada xompiler GCC berfungsi dengan fcloseall() pada compiler Borland.

c) Deteksi End File

Intruksi scant() digunakan untuk key-in data. Intruksi ini akan mengembalikan angka yang menyatakan jumlah data yang diketikkan melalui keyboard. Jika tidak ada data yang diketikkan melainkan dilakukan penekanan tombol ctrl Z maka akan dikembalikan bilangan negatif.

```
Deteksi jumlah data yang diketikkan
# include <stdio.h>
int main () {
  int i, j, jmldata;
  char c;

  jmldata = scanf("%d %c", &i, &c);
  while (jmldata == 2) {
    for (j = 0; j < i; j++) putchar(c);
    putchar('\n');
    jmldata = scanf("%d %c", &i, &c);
  } ;
  return 0;
}
```

```
***** hasil run *****
5 #
#####
8 *
********
^Z
```

Int **feof** (FILE*stream);

Intruksi feof() mendetekdi akhir sebuah stream. Apabila kondisi end-offile terpenuhi maka intruksi ini akan mengembalikan bilangan selain nil. Pada pemasukan data melalui pegerikan keyboard(stdin) snd-of-file terjadi apabila pemakaina menekan tombol ctrl Z secara bersamaan

d) File Reopen

FILE*__freopen__(const char*filename, const char*mode,FILE*stream):

Function freopen() mengasosikan sebuah berkas baru dengan sebuah stream yang sudah aktif. Dengan perkataan lain freopen lain freopen() menggunakan stream yang telah ada untuk mengakseskan berkas lain.

D. PENGOLAHAN DATA TEKS

Berkas teks adalah berkas yang direkam dengan format yang sama dengan tampilan di layar monitor tanpa dilakukan enskripsi. Berkas teks dapat diolah dengan editor teks seperti Notepad, IDE,Turbo, C++ 3.0, IDE Dev-C++. Pada Dos prompt sistem operasi Windows berkas teks dapat dilihat isinya dengan menggunakan instruksi type dan isinya sama seperti saat disunting dengan editor teks.

a. Olah Data Karakter

Instruksi yang berhubungan dengan pengolahan data karakter dan berkas teks adalah putc(), fputc(), getc(), dan fgetc().

Int putc (int c, FILE + stream);

Int fgetc (FILE +stream);

Instruksi putc() dan fputc() menulis karakter c ke berkas stream. instruksi ini mengembalikan karakter C jika proses penulisan berhasil atau mengembalikan EOF jika gagal.

Int getc (int c, FILE + stream);

Int fgetc (FILE +stream);

Instruksi getc() dan fgetc() membaca satu karakter dari berkas stream dan memimndahkan penunjuk posisi berkas stream ke karakter berikutnya. Instruksi ini mengembalikan karakter terbaca jika proses berhasil atau mengembalikan EOF jika terjadi kesalahan.

Instruksi fgetc() menyebabkan sebuah karakter dibaca dari berkas fin. Jika karakter tidak berhasil dibaca maka status berkas menjadi end-of-file atau feof() akan bernilai true. Bayangkan keadaan isis berkas sudah terbaca semua. Pada saat akan dibaca lagi untuk mendapatkan karakter berikutnya maka menyebabkan feof(). Pada penggalan yang diubah, setelah karakter dibaca dari berkas fin lalu karakter ini ditulis ke berkas fout tanpa terlebih dahulu memeriksa apakah sebenarnya ada karakter yang berhasil dibaca. Hal ini menyebabkan sebuah karakter yang tidak didefinisikan ikut tertulis ke dalam berkas fout. Instruksi fgetc() dan getc() membaca data masukan dari file stream. Karena stdin adalah input stream maka stdn dapat digunakan sebagai parameter fgetc() dan getc() seperti diperlihatkan program berikut.

Program: Sayang Semuanya

```
1  #include <studio.h>
2
3   int main() (
4.          FILE *fout;
5           char nama [21] , kar ;
6
7           printf (*nama berkas?  ‘’ ) ; scanf (‘’&s’’ , nama) ;
8           fout = fopen (nama,  ‘’w’’) ;
9           fflush(stdin) ;
10          kar = fgetc (stdin) ;
11          while (l feof (stdin) )   (
12                fputc (kar, fout) ;
13                kar = getc (stdin) ;
14      }
15        fclose (fout) ;
16        retrun (fout) ;
17   }
```

```
Nama berkas ? sayang 123.
Txt
1  1 Aku sayang ibu
2  2  Juga sayang ayah
3  3 Sayang adik kakak
Satu dua tiga sayang
semuanya
^z
```

```
D: \ > type sayang123. Txt
1 1 Aku sayang ibu
2 2 Juga sayang ibu
3 3 Sayang adik kakak
Satu dua tiga sayang
semuanya
```

b. Olah Data String

Instruksi yang berhubungan dengan pengolahan data string dan berkas teks adalah fputs() untuk menulis strig ke dalam berkas teks dan fgets() untuk membaca string dari berkas teks .

Int fputs (contst char *s, FILE *stream);

Instruksi fputs () menulis string s yang diakhiri Null ke berkas stream, tetapi karakter NULL tersebut tidak ikut ditulis, juga tidak ditambahkan karakter newline. Apabila proses penulisan berhasil maka instruksi ini mengembalikan sebuah bilangn positif, sebaliknya mengembalikan EOF.

Instruksi fgets() berusaha membaca n karakter dari berkas stream ke dalam string s. Instruksi ini berhenti membaca setelah terbaca n karakter atau terbaca karakter newline. Karakter newline yang terbaca akan ditambahkan pada akhir string. Apabila poses pembacaan berhasil maka instruksi ini mengembalikan string tersebut (ditunjuk oleh pointer), selain itu akan dikembalikan NULL.

E. PENGOLAHAN DATA BINER DAN RECORD BINER

Bit field adalah struktur data yang digunakan dalam pemrograman komputer . Terdiri dari sejumlah lokasi memori komputer yang berdekatan yang telah dialokasikan untuk menampung urutan bit , disimpan sehingga bit atau kelompok bit apa pun dalam himpunan dapat diatasi. Bidang bit paling umum digunakan untuk mewakili tipe integral dari lebar bit tetap yang diketahui. Arti dari bit individu dalam bidang ditentukan oleh programmer; misalnya, bit pertama dalam bidang bit (terletak di alamat dasar bidang) kadang-kadang digunakan untuk menentukan keadaan atribut tertentu yang terkait dengan bidang bit.

Sistem bilangan biner disusun dari angka-angka, sama seperti sistem bilangan desimal (sistem bilangan 10) yang sering digunakan saat ini.

Tetapi untuk desimal menggunakan angka 0 sampai 9, sistem bilangan biner hanya menggunakan angka 0 dan 1. Semua kode program dan data pada komputer disimpan serta dimanipulasi dalam format biner yang merupakan kode-kode mesin komputer.Sehingga semua perhitungannya diolah menggunakan aritmatik biner, yaitu bilangan yang hanya memiliki nilai dua kemungkinan yaitu 0 dan 1 dan sering disebut sebagai bit (binary digit) atau dalam arsitektur elektronik biasa disebut sebagai digital logic.

Untuk bilangan biner, kalikan bilangan paling kanan terus ke kiri dengan 2 0 , 2 1 , 2 2 , dst. Contoh : 101102 = (1 X 2 4) + (0 X 2 3) + (1 X 2 2) + (1 X 2 1) + (0 X 2 0) = (16 + 0 + 4 + 2 +0) = 22 Dari contoh diatas, menunjukkan bahwa bilangan biner 10110 sama dengan bilangan desimal 22. Dari dua sistem bilangan diatas, dapat dibuat rumus umum untuk mendapatkan nilai desimal dari radiks bilangan tertentu : (N)r = [(d0 x r 0) + (d1 x r 1) + (d2 x r 2) + … + (dn x r n)]10 dimana; N = Nilai r = Radiks d0, d1, d2 = digit dari yang terkecil (paling kanan) untuk d0 Untuk mengkonversi bilangan desimal kebiner ada dua cara, perhatikan contoh berikut:

Cara I : 16810 kurangkan dengan pangkat terbesar dari 2 yang mendekati 16810 yaitu 128 (2 7).

a. 128 (2 7) lebih kecil dari 168, maka bilangan paling kiri adalah 1. 168 – 128 = 40.
b. 64 (2 6) lebih besar dari 40, maka bilangan kedua adalah 0.
c. 32 (2 5) lebih kecil dari 40, maka bilangan ketiga adalah 1. 40 – 32 = 8
d. 16 (2 4) lebih besar dari 8, maka bilangan keempat adalah 0.
e. 8 (2 3) lebih kecil/sama dengan 8, maka bil. kelima adalah 1. 8 – 8 = 0.
f. Karena sisa 0, maka seluruh bit dikanan bil. kelima adalah 0. 16810 = 101010002.

Cara II :

168 / 2 = 84 sisa 0
84 / 2 = 42 sisa 0 4
2 / 2 = 21 sisa 0
21 / 2 = 10 sisa 1
10 / 2 = 5 sisa 0
5 / 2 = 2 sisa 1
2 / 2 = 1 sisa 0
1 / 2 = 0 sisa 1

Bit biner terbesar dimulai dari bawah, sehingga 16810 = 101010002

Berkas biner (binary file) dibedakan dari berkas teks dengan menambahkan karakter b pada modus pengaktifan berkas. Intruksi topen (“bilangan.bin”.”) mengaktifan berkas biner bilangan.bin untuk dibaca isinya. Intruksi topen (“buku.dok”,”wb”) mengaktifan berkas biner buku.dok untuk ditulisi record ke dalamnya. Intruksi topen (“ponsel.dat”,”ab”) mengaktifkan

berkas biner ponsel.dat untuk ditambahi record. Pada berkas biner data yang bertipe bilangan (integer dan floaf) akan deirekam dengan format biner.

Sizw_t **fwrite** (const coid*ptr,size_t size_f n, FILE *stream);

Intruksi fwrite() menambahkan n item data yang masing-masing berukuran size ke berkas stream. Data yang akan ditulis akan ditunjuk pointer ptn. Apabila proses berhasil maka intruksi ini mengembalikan jumlah data yang tertulis. Apabila terjadi kesalahan makan akan dikembalikan angka lain.

Fwrite data katakter

```
 Include <stdio.h>
Int main() {
        FILE *f;
        Char karakter = 'A' ;
        char string (10) = "algo";
        F = fopen("karakter.bin", "wb");
        fweitw(&karakter, sizeof(char), 1, f);
        fweitw(&string[0], sizeof(char[10]) , 1, f);
        fclose(f) ;
       Return 0 ;
}
```

Jika isi berkas karakter.bin ditampilkan dari MSDOS prompt dengan menggunakan instruksi type karakter.bin maka hasilnya adlah Aalgo. Jika berkas karakter.bin dibuka dengan menggunakan editor teks yang dapat menampilkan dengan format heksadesimal maka isinya adlah 41 61 6C 6F 00 00 00 00 00 00. Nilai 41 ASCII untuk karakter algo, dan enam bilangan nol selanjutnya untuk melengkapi format 10 karakter.

Pada DevC++ fwrite() akan menulis integer dalam bentuk format biner 4byte. Misalnya bilangan 64 dengan format biner adalah 01000000 00000000 00000000 00000000 atau dengan bentuk heksadesimal menjadi 40 00 00 00, sedangkan pada turbo C++ 3.0 yang menggunakan format 2 byte menjadi 40 00.

Fwrite data numeric

```
#Include <stdio.h>
Int main() {
        FILE *f;
        int bilangan = 64 ;
        float pecahan = 12.5 ;
        F = fopen("karakter.bin", "wb");
        fweitw(&bilangan, sizeof(int) ,  1, f);
        fweitw(&pecahan, sizeof(char(float) , 1, f);
        fclose(f) ;
       Return 0 ;
}
```

Data numeric direkam kedalam berkas biner dengan menggunakan format biner, sehingga jika isi berkas bilangan.bin ditampilkan dari MSDOS prompt degan mengunakan intruksi type bilangan.pin maka hasilnya tidak mencerminkan bilangan 64 dan 12.5

Jika berkas bilangan.bin dibuka dengan menggunakan editor terks yang dapat menampilakan isi berkas dengan format heksadesimal maka isinya adalah 40 00 00 00 00 00 48 41. Empat bilangan bulat 64 dan empat bilangan selanjutnya adalah representasi bilangan pecahan 12.

Intruksi fwrite() juga dapat digunakan unttuk merekam sejumlah data sekaligus. Pada program berikutnya lima elemen data pertama larik diberi nilai kemudian direkam sekaligus ke dalam berkas biner larik.bin

Intruksi fread{} membaca n elemen data yang masing-masing berukuran size dari berkas stream ke dalam suatu blok memori yang ditunjuk printer ptn. Apabila proses berhasil maka intruks ini mengembalikan jumlah elemen data yang terbaca. Apabila tidak ada data yang terbaca maka intrksi ini mengembalikan nol.

Intruksi fread() juga dapat digunakan untuk mrmbaca sejumlah data sekaligus. Pada program berikut kita membaca 200 interger sekaligus dari berkas biner larik.bin. karena berkas ini hanya mempunyai lima data maka kelima data yang terbaca, yang di tunjukkan oleh isi variable n.

Berbeda dengan intruksi fprintf() yang harus menyertakan format setiap data yang akan dorekam ke dalam berkas, intruksi fwrite() meminta ukuran data(dalam satuan byte yang akan direkam dan alamat awal data tersebut. Dengan demikian intruksi fwite() selain dapat merekam data tunggal(interger,character,floating point number) juga dapat digubakan untuk merekam data intruksi seperti larik dan recor(strukture). Keempat data dijadikan dield-field dari sebuah structure. Berkas ponsel.dat diaktifkan dengan modus"ab" agar data lama tidal hilang dan data baru ditambahkan di akhir.

intrukis dscant() yang harus menyertakan format setiap elemen data yang akan dibaca dari berkas , intruksi fread() meminta ukuran data (dalam satuan byte) yang akan dibaca dan alamat awa; data tersebut. Dengan demikian intriksi fread() selain dapat membaca data tunggal dapat digunakan untuk membaca data terstruktur.

F. PENGOLAHAN DATA TEKS BERFORMAT

Intruksi fprintf() dan fscanf() berfungsi untuk mengolah sejumlah data yang berbeda tipe sekaligus, sehingga dapat digunakam untuk menggantikan *function-function* manipulasi berkas sebelumnya.

1. Function fprintf() menulis data berformat ke berkas stream sesuai dengan format masing-masing argumen. Apabila proses berhasil maka intruksi ini mengembalikan bilangan yang menyatakan jumlah byte yang dituliskan,sebaliknya mengembalikan EOF. Setiap record mempunyai panjang(jumlah karakter) yang berbeda. Struktur record seperti ini disebut variabel length record.
2. Intruksi fscanf() membaca data dari berkas stream ke dalam field yang sesuai format. Apabila proses berhasil maka intruksi ini mengembalikan bilangan yang menyatakan jumlah field yang berhasil dibaca. Apabila tidak ada field yang berhasil dibaca maka maka akan dikembalikan angka nol. Apabila end-of-file maka akan dikembalikan EOF.

G. PERCETAKAN KE PRINTER

Pada lingkungan compiler turbo C++ 3.0 selain ketiga stream baku yang otomatis diaktifkan pada saat sebuah program berjalan (stdin,sdout,stderr), juga diaktifkan standard printing stream (stdprn). Stream ini dapat digunakan untuk mengirim keluaran printer untuk dicetakkan melalui pararell port LPT1(bukan Port DOT yang banyak digunakan printer sekarang, yang menggunakan table USB). Pada percetakan printer/n akan memindai posisi percetakan ke kolom yang sama berikutnya, sehingga perlu/r yang akan memindai posisi ke kolom pertama atau terkini

Program daftar perkalian

```
1 #include <stdio.h>
2 #include <conio.h>
3
4 int main() {
5       int dasar, mulai ,sampai, I;
6
7  clrscr() ;
8     printf(“bilangan dasar ? “);scanf(“%d”, &dasar) ;
9     printf(“mulai dari  ? “);scanf(“%d”, &mulai) ;
10   printf(“sampai dengan  ? “);scanf(“%d”, &sampai) ;
11    printf(stdprn,”TABEL PERKALIAM %d-AN   /n/n/r”, dasar) ;
12    for (I = mulai ; I <= sampai ; i++)
13    fprintf(stdprn,”%2d*%2d=%3d/n/r, dasar, I,dasar*i) ;
14    return 0 ;
15  }
```

```
Bilangan dasar ? 5
Mulai dari ? 1
Sampai dengan ?10
```

```
°   TABEL PERKALIAN 5-AN                                    °
°                                                           °
°   5*1=5                                                   °
°   5*2=10                                                  °
°   5*3=15                                                  °
°   5*4=20                                                  °
°   5*5=25                                                  °
°   5*6=30                                                  °
°   5*7=35                                                  °
°   5*8=40                                                  °
°   5*9=45                                                  °
°   5*10=50                                                 °
°                                                           °
```

H. KESIMPULAN

File adalah sekumpulan data yang disimpan dalam media penyimpanan luar seperti disket/hardisk. Stream adalah suatu logika device yang menghasilkan dan menerima informasi atau wadah yang digunakan untuk menampung keluaran dan menampung aliran data. Fungsi clear ( ) dalam keadaan default akan membersihkan seluruh flag. Dapat ditentukan flag mana yang akan dibersihkan, dengan memberi argument yang sesuai. Fungsi int rdstate ( ) digunakan untuk menentukan jenis kesalahan yang terjadi, dengan mengembalikan nilai integer kesalahan tersebut. Berkas teks adalah berkas yang direkam dengan format yang sama dengan tampilan di layar monitor tanp dilakukan enskripsi. Berkas biner (binary file) dibedakan dari berkas teks dengan menambahkan karakter b pada modus pengaktifan berkas.

I. SOAL LATIHAN

1. Stream dapat di bedakan berdasarkan jenis data yang di alirkan pada stream, stream karakter dan byte. Jelaskan masing masing kegunaan dari jenis stream tsb! Sebutkan alasan mengapa harus menggunakan bahasa C ?
2. Jelaskan perbedaan field dengan file dan berikan contohnya.!
3. Jelaskan cara yang dapat digunakan untuk mereduksi jumlah akses ke disk dalam kinerja sistem file yang termasuk dalam manajemen file I/O!
4. Tentukan bilangan biner dari 580.
5. Buatlah contoh program file I/O!

DAFTAR PUSTAKA

Aan Junior. 2016. https://www.academia.edu/26254918/MODUL_I_PENGETAHUAN_DASAR_PEMROGRAMAN

Amborowati Armadyah. Array Konsep. *Array Konsep*, 1-14.

Alex Budiyanto. (2003). Pengantar_Algoritma_dan_Pemrograman_1. In *Pengantar Algoritma dan Pemrograman*.

Ardiansyah, Hendri, dkk; 2019; *Algoritma dan Pemrograman 1*; Banten; Unpam Press.(http://eprints.unpam.ac.id/8617/1/TPL0022_ALGORITMA%20DAN%20PEMROGRAMAN%201.pdf)

Dan, A., Prawira, D., & Informasi, J. S. (2015). *Modul perkuliahan algoritma dan pemrograman*.

Data, T. (n.d.). *Konsep tipe data 2.1* (Issue 1, pp. 3–6).

Dian Prawira; modul algoritma dan pemrograman.(http://hmsiuntan.com/wp-content/uploads/2019/09/modul-algoritma-pemrograman-revisi-des-2016-1.pdf)

File, F. (n.d.). *FILE & STREAM*. *2*(5), 1–5.

http://staffnew.uny.ac.id/upload/131872515/pendidikan/Bab+II+Keg+Pemb+3+_Oprator.pdf

https://www.google.com/url?sa=t&source=web&rct=j&url=https://repository.unikom.ac.id/33444/1/MODUL%25201%2520dan%25202.pdf&ved=2ahUKEwjE77esvLnoAhUJ63MBHffMDIUQFjAEegQIBxAB&usg=AOvVaw3cPIBaxlnzXnQ3iMW1c5Yx

https://www.google.com/url?sa=t&source=web&rct=j&url=http://julio.staff.ipb.ac.id/files/2011/12/14-directives.pdf&ved=2ahUKEwi_qrvevLnoAhUh6nMBHZc3ANwQFjAAegQIBRAB&usg=AOvVaw0kNoiwYJVQT7iXoVCqoxRS

https://www.geeksforgeeks.org/basic-input-output-c/

https://en.wikipedia.org/wiki/Bit_field

https://en.m.wikipedia.org/wiki/Numeral_system

https://translate.google.com/translate?u=https://en.wikipedia.org/wiki/Macro_(computer_science)&hl=id&sl=en&tl=id&client=srp

https://translate.google.com/translate?hl=id&sl=en&u=https://en.wikipedia.org/wiki/Variadic_function&prev=search

Komputer, D., & Bilangan, S. (n.d.). *2. Dasar dari Komputer, Sistem Bilangan, dan Gerbang logika 2.1. Data*. 1–15. https://doi.org/10.1109/ISCBI.2017.8053544

Maulana, G. G. (2017). Pembelajaran Dasar Algoritma Dan Pemrograman Menggunakan El-Goritma Berbasis Web. *Jurnal Teknik Mesin*, *6*(2), 8.

https://doi.org/10.22441/jtm.v6i2.1183

Mukidin. S. Kom., 2009. Array(Larik). *Array(larik)*, 3-8.

Munir, Rinaldi, Leony Lidya; 2015; *Algoritma dan Pemrograman*; Bandung; Informatika Bandung.

(PRAWIRA 2015)Pemilihan, V Struktur. 2008. "STRUKTUR PEMILIHAN Suatu." *2008*: 1–7. PRAWIRA, DIAN. 2015. *MODUL PERKULIAHAN ALGORITMA DAN PEMROGRAMAN*. 1st ed. ed. DIAN PRAWIRA. Pontianak.

(Pemilihan 2008) Pemilihan, V Struktur. 2008. "STRUKTUR PEMILIHAN Suatu." *2008*: 1–7.

Rahma Aulia. 2015. Praktikum 2: Operasi Input Output. Sistem Operasi

Record, C. M. (n.d.). *Record dapat dikatakan sebagai suatu kumpulan data item mempunyai jenis data berbeda. yang masing-masing Data item yang merupakan elemen record biasanya disebut dengan FIELD*. (pp. 1–9).

Scarlet, D. (2013). 済無No Title No Title. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9). https://doi.org/10.1017/CBO9781107415324.004

Suprapto, Kadarisma Tejo Yuwono dkk. 2008. Bahasa Pemrograman. Jakarta: Direktorat Pembina Sekolah Menengah Kejuruan

Suprapto; 2008; *Bahasa Pemrograman untuk Sekolah Menengah Kejuruan*; Jakarta; Direktorat Jenderal Manajemen Pendidikan Dasar dan Menengah Departemen Pendidikan Nasional

Syaifudin, Y. ., Rozi, I. F., Mentari, M., & Lestari, V. A. (2018). Dasar Pemrograman. *DASAR PEMROGRAMAN*, *1*, 1–170.

Thomson Susabda Ngoen. (2009). *Algoritma dan Struktur Data Bahasa C*. Mitrawacanamedia.

Umum, T. I., Khusus, T. I., Materi, P., & Bertingkat, S. (n.d.). *(Struktur) 14* (pp. 116–121).

Ulti Desi Arni. (2018). Logika dan Algoritma Backpropagation. In *Garuda Cyber*. https://garudacyber.co.id/artikel/635-logika-dan-algoritma-backpropagation.

Utami, E. (n.d.). Struktur Pemilihan. *Struktur Pemilihan*, *1*, 1–7.

Windarto Agus Perdana, Henny Harumy, Indri Sulistiningsih. 2016. Belajar Dasar Algoritma dan Pemrograman C++

Wilis Kaswidjanti, S.Si.,M.Kom; Modul Algoritma dan Pemrograman function pertemuan ke-11

Biodata Penulis

Susanti, menyelesaikan S1 di Program Studi Pendidikan Matematika Fakultas Keguruan dan Ilmu Pendidikan STKIP "Tapanuli Selatan" Padangsidimpuan tahun 2014, dan S2 di Program Pascasarjana jurusan Pendidikan Matematika Universitas Negeri Padang (UNP) tahun 2017. Saat ini adalah dosen di Program Studi Pendidikan Matematika Fakultas Keguruan dan Ilmu Pendidikan Universitas Maritim Raja Ali Haji (FKIP UMRAH) Tanjungpinang, Kepulauan Riau. Mulai aktif mengajar di Program Studi Pendidikan Matematika pada tahun 2019.