

# Turnitin Originality Report

Processed on: 10-Sep-2020 9:50 AM WIB  
 ID: 1383340766  
 Word Count: 8802  
 Submitted: 1

Similarity Index  
**20%**

## Similarity by Source

Internet Sources: 20%  
 Publications: 1%  
 Student Papers: 1%

Buku Algoritma dan  
 Pemrograman C By Patmi  
 Kasih

1% match (Internet from 19-Jun-2019)

<http://rppsilabusmasmpsdkelas.blogspot.com/2013/03/cara-program-mencetak-matriks-dengan.html>

1% match (Internet from 19-Jan-2017)

[http://comlab.ilkom.unsri.ac.id/wp-content/uploads/2015/03/Modul-Praktikum-Alpro-I\\_MI.docx](http://comlab.ilkom.unsri.ac.id/wp-content/uploads/2015/03/Modul-Praktikum-Alpro-I_MI.docx)

1% match (Internet from 25-Apr-2019)

<http://patriciadianitawijaya.blogspot.com/2012/04/binary-search.html>

1% match (Internet from 01-Apr-2019)

<http://feri-saptono.blogspot.com/2009/>

1% match (Internet from 17-Jun-2020)

[https://mafiadoc.com/algoritma-pemrograman-sub-pokok-bahasan\\_59fca35f1723dd82f6a48824.html](https://mafiadoc.com/algoritma-pemrograman-sub-pokok-bahasan_59fca35f1723dd82f6a48824.html)

1% match (Internet from 05-Feb-2020)

<http://brontoz.blogspot.com/2011/>

1% match (Internet from 03-Mar-2019)

<http://hendrazon3.blogspot.com/2009/05/>

1% match (Internet from 28-Dec-2019)

<https://fr.scribd.com/doc/296393748/251160880-Modul-Program-C-Codeblocks-Revisi-1>

1% match (Internet from 26-Jul-2016)

<https://zainsvg.wordpress.com/page/2/>

1% match (Internet from 24-Aug-2020)

<https://zhainnet.blogspot.com/2013/10/makalah-dasar-dasar-pemrograman.html>

1% match (Internet from 10-Jun-2020)

<http://fasilkom.mercubuana.ac.id/wp-content/uploads/2017/10/Pemrograman-Algoritma-dan-Struktur-Data-TI.pdf>

1% match (Internet from 23-Mar-2011)

<http://staff.unud.ac.id/~agungcahyawan/wp-content/uploads/2010/11/array.pdf>

1% match (Internet from 15-Mar-2020) <a href="http://haidaralv.blogspot.com/2014/03/algoritma-operator-dan-seleksi-beserta.html">http://haidaralv.blogspot.com/2014/03/algoritma-operator-dan-seleksi-beserta.html</a>
1% match (Internet from 01-Aug-2018) <a href="http://iwan-ihmas.blogspot.com/2015/02/">http://iwan-ihmas.blogspot.com/2015/02/</a>
1% match (Internet from 04-Jul-2020) <a href="https://id.123dok.com/document/zwv2wxgg-laporan-individu-kegiatan-praktik-pengalaman-lapangan-ppl-uny-di-smk-negeri-2-wonosari-1.html">https://id.123dok.com/document/zwv2wxgg-laporan-individu-kegiatan-praktik-pengalaman-lapangan-ppl-uny-di-smk-negeri-2-wonosari-1.html</a>
1% match (Internet from 16-Apr-2019) <a href="http://kev.pulo.com.au/pp/RESOURCES/cplusplus/tutorial/tutorial/tut3-2.html">http://kev.pulo.com.au/pp/RESOURCES/cplusplus/tutorial/tutorial/tut3-2.html</a>
1% match (Internet from 18-Sep-2017) <a href="http://atit.staff.gunadarma.ac.id/Downloads/files/38105/Pertemuan+2+C%2B%2B.pdf">http://atit.staff.gunadarma.ac.id/Downloads/files/38105/Pertemuan+2+C%2B%2B.pdf</a>
1% match (Internet from 12-Jun-2019) <a href="https://docplayer.info/33859502-Lsp-telematika-cover.html">https://docplayer.info/33859502-Lsp-telematika-cover.html</a>
1% match (Internet from 10-Feb-2020) <a href="https://id.scribd.com/doc/253367722/modul-C">https://id.scribd.com/doc/253367722/modul-C</a>
1% match (Internet from 12-Nov-2019) <a href="http://therezna.blogspot.com/2010/">http://therezna.blogspot.com/2010/</a>
1% match (Internet from 09-Jun-2020) <a href="https://www.coursehero.com/file/47260447/Pertemuan-9pptx/">https://www.coursehero.com/file/47260447/Pertemuan-9pptx/</a>
1% match (Internet from 17-Oct-2019) <a href="http://yogapratama1205.blogspot.com/">http://yogapratama1205.blogspot.com/</a>
1% match (Internet from 20-Apr-2020) <a href="https://jhuzmatulufliana.blogspot.com/2014/01/laporan-algoritma-dan-pemrograman.html">https://jhuzmatulufliana.blogspot.com/2014/01/laporan-algoritma-dan-pemrograman.html</a>
1% match (Internet from 29-Nov-2018) <a href="http://www.teorimaya.com/">http://www.teorimaya.com/</a>
1% match (Internet from 11-Jan-2020) <a href="https://hayatsetiawanblog.wordpress.com/blog/">https://hayatsetiawanblog.wordpress.com/blog/</a>
1% match (Internet from 01-Sep-2016) <a href="https://ml.scribd.com/doc/309983679/soalbahasac8saberan">https://ml.scribd.com/doc/309983679/soalbahasac8saberan</a>
MODUL PRAKTIKUM ALGORITMA PEMROGRAMAN DENGAN C++ Untuk Kalangan Sendiri Dosen Pengampu : Patmi Kasih, M.Kom Rina Firliana, M.Kom FAKULTAS TEKNIK UNIVERSITAS NUSANTARA PGR KEDIRI 2016 ALGORITMA DAN PEMROGRAMAN <u>Algoritma adalah urutan aksi-aksi yang dinyatakan dengan jelas dan tidak rancu untuk memecahkan suatu masalah dalam rentang waktu tertentu. Setiap aksi harus dapat dikerjakan dan mempunyai efek tertentu. Algoritma</u> merupakan logika, metode, dan

tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan. Algoritma dapat dituliskan dengan banyak cara, mulai dari menggunakan bahasa alami yang digunakan sehari-hari, simbol grafik bagan alir (flowchart), sampai menggunakan bahasa pemrograman seperti bahasa C atau C++.

Program adalah kumpulan instruksi komputer, sedangkan metode dan tahapan sistematis dalam program adalah algoritma. Program ini ditulis dengan menggunakan bahasa pemrograman. Jadi bisa kita sebut bahwa program adalah suatu implementasi dari bahasa pemrograman. Beberapa pakar memberi formula bahwa : program = struktur data + algoritma Bagaimanapun juga struktur data dan algoritma berhubungan sangat erat pada sebuah program. Algoritma yang baik tanpa pemilihan struktur data yang tepat akan membuat program menjadi kurang baik, demikian juga sebaliknya. Struktur data disini bisa berupa list, tree, graph, dsb.

• Tujuan Praktikum Tujuan praktikum ini adalah mengenalkan mahasiswa pada bahasa pemrograman C++. Setelah mempelajari algoritma pemrograman, diharapkan mahasiswa dapat menerapkannya pada sejumlah bahasa pemrograman tanpa terpaku pada salah satu bahasa pemrograman.

• Tata Tertib Praktikum 1. Praktikum dilaksanakan sebagai salah satu point dalam penilaian nilai akhir sehingga dituntut kedisiplinan di dalam pelaksanaan praktikum, sehingga dalam praktikum pemrograman ini dibuat beberapa aturan sebagai berikut :

2. Peserta praktikum wajib datang tepat waktu.
3. 10 menit sebelum praktikum dimulai, akan diadakan evaluasi atau pengecekan peserta oleh pengajar praktikum.
4. Penilaian yang dilakukan adalah penilaian pada saat berlangsungnya praktikum, dengan skala 1-10.
5. Ketertiban, kesopanan, dan keaktifan saat mengikuti praktikum dijadikan salah satu point penilaian oleh pengajar praktikum.
6. Peserta praktikum diwajibkan menunjukkan jalannya program yang sudah dibuat sendiri kepada pengajar praktikum.

• Perangkat dan Kompiler Sebagai penghargaan terhadap hak kekayaan intelektual, dalam hal ini penggunaan perangkat lunak, maka program aplikasi/tool kompiler untuk praktikum ini menggunakan perangkat lunak yang bersifat free.

### PENGENALAN BAHASA C++

A. Sekilas C++

Bahasa pemrograman computer terdiri atas dua bagian, yaitu bahasa pemrograman tingkat tinggi (high level language) dan bahasa pemrograman tingkat rendah (low level language).

Penggolongan ini didasarkan pada kemudahan untuk mempelajari bahasa pemrograman komputer tersebut dan kecepatan eksekusinya. Makin tinggi tingkat suatu bahasa pemrograman komputer, maka bahasa pemrograman komputer tersebut akan makin mudah dipelajari. Dan sebaliknya, makin rendah tingkat suatu bahasa pemrograman komputer, maka makin tinggi kecepatan eksekusinya.

Bahasa pemrograman C++ adalah salah satu bahasa pemrograman komputer tingkat tinggi. Hal ini dikarenakan bahasa pemrograman C++ termasuk bahasa pemrograman komputer yang mudah dipelajari. Tetapi bahasa pemrograman C++ lebih banyak digunakan daripada bahasa pemrograman lainnya, karena memiliki kecepatan eksekusi yang cukup tinggi bila dibandingkan dengan bahasa pemrograman komputer tingkat tinggi lainnya.

Bahasa pemrograman C++ memiliki keunikan tersendiri dari bahasa pemrograman komputer yang lain, yaitu bersifat case sensitive (membedakan antara huruf kecil dan huruf kapital), dimana hampir semua perintah dalam bahasa pemrograman C++ menggunakan huruf kecil.

Sejarah C++ C++ adalah pengembangan dari bahasa C, yang merupakan pengembangan dari dua bahasa pemrograman generasi sebelumnya, yaitu BCPL dan B (Basic). BCPL dibuat pada tahun 1967 oleh Martin Richards sebagai bahasa untuk menulis sistem operasi dan compiler. Ken Thompson membuat banyak fitur pada

bahasa B yang dibuatnya dan menggunakan B untuk membuat versi awal dari sistem operasi UNIX di Bell Laboratories pada tahun 1970 pada komputer DEC PDP-7. Bahasa C dikembangkan dari bahasa B oleh Dennis Ritchie di Bell Laboratories dan pada awalnya diimplementasi pada komputer DEC PDP-11 pada tahun 1972. C menggunakan banyak konsep penting dari BCPL dan B sekaligus ada tambahan jenis-jenis data dan fitur lainnya. C kemudian dikenal sebagai bahasa pengembang sistem operasi UNIX. Pada masa sekarang, kebanyakan sistem operasi ditulis dengan menggunakan C dan/atau C++. C tersedia untuk hampir semua komputer. Pada akhir dekade 1970-an, C telah berkembang dengan menjadi sesuatu yang sekarang disebut "C tradisional", "C klasik", atau "C Kernighan dan Ritchie". C++ adalah penambahan dari C, dikembangkan oleh Bjarne Stroustrup pada awal dekade 1980-an di Bell Laboratories. C++ memberikan tambahan fitur yang meningkatkan kekuatan bahasa C, dan yang lebih penting lagi, kemampuan untuk pemrograman berbasis object ([Object Oriented Programming](#)).

B. [Kelebihan dan Kekurangan Bahasa C/C++](#) : ? [Bahasa C++ tersedia hampir di semua jenis komputer.](#) ? [Kode bahasa C/C++ sifatnya adalah portable dan fleksibel untuk semua jenis komputer.](#) ? [Proses executable program bahasa C/C++ lebih cepat.](#) ? [Dukungan pustaka yang banyak.](#) ? [C adalah bahasa yang terstruktur.](#) ? [C++ sudah mendukung OOP \(Object Oriented Programming\).](#) [Kekurangan Bahasa C/C++](#) : ? [Banyaknya Operator serta fleksibilitas penulisan program kadang-kadang membingungkan pemakai.](#) ? [Bagi pemula pada umumnya akan kesulitan menggunakan pointer dan penerapan konsep OOP.](#) Editor Bahasa C/C++ [Untuk memulai membuat program, tersedia berbagai editor yang dapat digunakan](#) , diantaranya : [Turbo C++](#), [Borland C++](#), [C++ builder](#), Microsoft Visual C++, dlsb.

C. Langkah-langkah menuliskan program dalam Borland C++

1. Bukalah editor Borland C++ melalui START menu, atau double click icon turbo C++
2. Maka akan muncul tampilan aplikasi sebagai berikut, Gambar : User Interface Borland C++ 5.02
3. [Source Code program C/C++ dapat ditulis di text editor Borland C++.](#) [File](#) → [New](#) → [Text Edit](#), maka akan muncul layar putih tempat untuk menulis program C++.
4. Program dapat disimpan dengan Pilih Menu [File](#), [Save As](#) atau [Save](#), [Save All](#).
5. Untuk mengeksekusi dan mengcompile (menjalankan) program C++ yang telah kita buat kita mengklik tombol petir berikut: [Kompilasi file dapat juga dilakukan dengan \(ALT + F9 atau pilih submenu Compile\)](#). [Compiler dijalankan untuk mengubah source code menjadi sebuah bahasa program.](#) [Setelah kita kompilasi file yang berisi source code, maka sebagai hasil kompilasi, kita akan mendapatkan suatu file yang bisa dijalankan \(executable file\).](#) [Menjalankan program yang kita buat berarti menjalankan file hasil proses kompilasi tersebut.](#)

D. Struktur program C++ Struktur dasar program bahasa C++ terdiri dari tiga bagian, sebagai berikut:

- o Preprocessor Directives Preprocessor directives merupakan perintah dalam bahasa C++ yang akan membuat dan mengkompilasi file perantara dengan program utama
- o Deklarasi Global Deklarasi global merupakan bagian program C++ tempat pendeklarasian konstanta, variabel, fungsi atau header fungsi, tipe data baru, atau struktur data yang memiliki sifat global.
- o Fungsi Utama Fungsi utama adalah fungsi pertama yang akan dieksekusi oleh kompiler C++. Dan kompiler C++ hanya akan mengeksekusi fungsi-fungsi lain yang dipanggil pada fungsi utama atau yang memiliki hubungan dengan fungsi utama secara tidak langsung. Bagian ini digunakan untuk meletakkan seluruh instruksi program. Instruksi-instruksi yang akan diberikan untuk dikerjakan ditulis di dalam main() diantara "{" dan "}". Yang perlu diperhatikan bahwa setiap instruksi atau baris perintah diakhiri dengan

tanda baca titik koma (;) atau semicolon. Definisi fungsi merupakan tempat pendefinisian header fungsi yang telah dideklarasikan pada bagian deklarasi global. #include ? preprocessor directives tipe\_data nama\_variabel; ? deklarasi variable global tipe\_data main() { ? fungsi utama } Blok Contoh: 1 //preprocessor directive 2 #include //file header 3 //deklarasi global 4 int TES = 1000; 5 //fungsi utama 6 int main() 7 { 8 printf("Nilai TES= %d", TES); 9 return 0; 10 } #include adalah pengarah preprosesor yang berfungsi menginstruksikan kepada kompiler untuk menyisipkan file lain saat program dikompilasi (yaitu file-file yang berisi stream dan fungsi-fungsi input/output). Biasanya file-file yang disisipkan adalah file-file header, Pustaka input/output merupakan pustaka. Pustaka ini telah ada di header stdio.h dan iostream.h. void didepan main(), untuk menyatakan bahwa fungsi main() tidak memiliki nilai balik. main() menjadi awal dan akhir eksekusi program C++, sehingga sebuah program dalam C++ mengandung sebuah fungsi main() Main →nama judul fungsi { →awal tubuh fungsi/awal eksekusi program →tubuh fungsi/blok } →akhir tubuh fungsi/akhir eksekusi program Tanda () digunakan untuk mengapit argumen fungsi, yaitu nilai yang akan dilewatkan ke fungsi. Blok pernyataan : merupakan satu atau beberapa buah statemen/pernyataan yang pada setiap akhir baris pernyataan diakhiri dengan titik koma (;). Contoh program : #include #include void main() { clrscr(); //membersihkan layar cout <<"Selamat Belajar C++\n"; getch(); } Hasil eksekusi : Selamat Belajar C++ Mengenal cout Pengenal cout (baca : cout) merupakan objek dalam C++ yang digunakan untuk mengarahkan data ke standar output (layar). Atau dapat juga di tuliskan dengan printf. Tanda << (dua tanda kurang dari berurutan) adalah operator "penyisipan/peletakan" yang akan mengarahkan operand (data) yang terletak di sebelah kanannya ke objek yang terletak di sebelah kirinya. Pustaka input/output merupakan pustaka yang berisi stream dan fungsi-fungsi input/output. Pustaka ini telah ada di header stdio.h dan iostream.h. Dalam bahasa C pustaka I/O yang digunakan adalah stdio.h yang berisi fungsi- fungsi seperti printf (cout dan scanf. Scanf(cin) digunakan untuk memasukkan sebuah nilai ke variable dan printf(cout) digunakan untuk mencetak suatu nilai dari variable maupun konstanta. Pada contoh di atas "Selamat Belajar C++\n" diarahkan ke cout, yang memberikan hasil berupa tampilan string tersebut ke layar. \n, dapat juga dituliskan </karakter pindah baris (new line). #include <iostream.h> #include <iostream.h> menginstruksikan kepada kompiler untuk menyisipkan file iostream.h pada saat program dikompilasi tanpa diakhiri titik koma. File iostream.h perlu disertakan pada program yang melibatkan cout. Tanpa #include <iostream.h> akan terjadi kesalahan saat program dikompilasi. Sebab file iostream.h berisi deklarasi yang diperlukan oleh cout dan berbagai objek yang berhubungan dengan operasi masukan-keluaran. clrscr(); Pernyataan yang diperlukan untuk menghapus layar. Apabila menggunakan pernyataan ini maka harus disertakan file header conio.h. Komentar Komentar diperlukan untuk menjelaskan mengenai program atau bagian-bagian dalam program. Isi penjelasan berupa: - Tujuan/fungsi program - Saat program dibuat/direvisi, bias berupa tanggal dibuatnya program tersebut. - Keterangan-keterangan lain tentang kegunaan sejumlah pernyataan dalam program. Tanda awal komentar dalam program C++ ada dua cara : ? Diawali tanda // (dua tanda garis miring/double slash), untuk komentar yang hanya satu baris. Semua tulisan setelah tanda // dianggap sebagai komentar dan tidak akan dieksekusi oleh C++. ? /\*mulai blok komentar pada baris ini tidak akan dieksekusi sampai ditemui akhir blok komentar \*/. Contoh program :

```
//Program yang mengandung komentar #include #include void main() {
clrscr(); //membersihkan layar /*mulai blok komentar pada baris ini tidak
akan dieksekusi sampai ditemui akhir blok komentar */ cout <<"Selamat
Belajar C++\n"; getch(); } Note : Sebelum mulai melakukan coding
program, sebaiknya diingat bahwa C/C++ bersifat "case sensitive", yang
berarti huruf besar dan huruf kecil nantinya akan dianggap berbeda oleh
program. Tugas Praktikum 1 1. Coba anda buat program tampilan, dengan
menampilkan: a. Nama: ..... b. Fakultas: ..... c. Universitas:
..... d. Kota : ..... e. .... Jawaban : Telah diperiksa
Tanggal : Modul Dengan ini mahasiswa telah menyetujui dan menyatakan
bahwa aplikasi telah berjalan serta selesainya pengisian jawaban dari tiap
pertanyaan yang ada. Pengajar Praktikum Mahasiswa Nilai BAB I
PERNYATAAN, TIPE DATA, VARIABEL & KONSTANTA A. Tujuan Percobaan -
Memahami pengertian dan penggunaan pernyataan, tipe data, variabel dan
konstanta dalam bahasa C++ - Memahami penggunaan tipe data standar
pada bahasa C++ - Memahami cara pendefinisian tipe data baru pada
bahasa C++ B. Pernyataan, Tipe Data, Variabel, dan Konstanta Pernyataan
dalam bahasa C++ merupakan serangkaian atau kelompok rangkaian
perintah/symbol standar C++, perintah/symbol yang didefinisikan
pemrogram, variabel/ konstanta, dan atau operator yang diakhiri dengan
tanda ; (titik koma/semi-colon). pernyataan; Variabel pada bahasa C++
adalah bagian dari memori yang hanya dapat menampung satu
data/informasi pada satu waktu, dimana data/informasi tersebut dapat
berubah setelah dideklarasikan selama pengeksekusian program. Variabel
dilambangkan dengan simbol yang didefinisikan oleh pemrogram dan
variabel harus dideklarasikan terlebih dahulu sebelum dapat digunakan.
Pendeklarasian variabel pada bahasa C++ adalah sebagai berikut :
tipe_data nama_variabel; Variabel dapat langsung diberi nilai awal pada
saat pendeklarasiannya. Pemberian nilai tersebut dapat dilakukan dengan
cara sebagai berikut, tipe_data nama_variabel=nilai_awal; Konstanta
merupakan variabel yang data/nilainya tidak dapat diubah setelah
dideklarasikan selama pengeksekusian program. Cara mendeklarasikan
konstanta adalah sebagai berikut, const tipe_data
nama_konstanta=nilai_konstanta; Cara lain untuk mendeklarasikan
konstanta adalah dengan mendeklarasikannya sebagai makro, yaitu
sebagai berikut, #define nama_makro definisi_makro; Besarnya bagian
memori yang digunakan oleh variabel atau konstanta untuk menampung
data/informasi tergantung dari tipe data/informasi yang digunakan. Tipe-
tipe data standar yang terdefiniskan pada bahasa C++ dapat dilihat dalam
tabel berikut : Tabel : Tipe-tipe data standar pada bahasa C++ Tipe data
Ukuran Presisi Jangkauan nilai Jumlah digit presisi 1. char 2. int 3. long int
4. float (desimal) 5. double 6. long double 1 byte 2 byte 4 byte 4 byte 8
byte 10 byte -128 sampai +127 -32768 sampai +32767 -2147438648
sampai +2147438647 3.4x10-38 hingga 3.4x10+38 1.7x10-308 hingga
1.7x10+308 3.4x10-4932 hingga 1.1x10+4932 - - - 6-7 15-16 19
Pemodifikasian tipe data unsigned dan signed Untuk pemodifikasian tipe
unsigned diterapkan pada data bertipe bilangan bulat (char, short, int dan
long). Variable yang ditambah unsigned akan menyebabkan nilai yang
terkandung didalamnya selalu bernilai positif, sehingga jangkauannya akan
berubah. Untuk pemodifikasian tipe signed merupakan default dari tipe
data dasar, yang menyatakan data bernilai positif maupun negative.
Pemodifikasian tipe data dapat dilihat pada tabel berikut Tabel 1.2. Tipe-
tipe data pemodifikasian pada bahasa C++ 1. char/signed char 2. unsigned
char 3. int/signed int 4. short int/signed short int 5. unsigned int 6. long
int/signed long int 7. unsigned long int Untuk mendefinisikan suatu tipe
```

data baru selain tipe-tipe data standar pada bahasa C++ digunakan perintah berikut: typedef tipe\_data tipe\_data\_baru; Tabel : Notasi Bahasa C++ Kelompok Notasi C++ Ket. 1. Tipe Dasar Boolean Byte Shortint Word Integer 1 byte 1 byte 2 byte 2 byte Longint Real Double Extended Char String string[N] Record field1: type; field2: type; . . . fieldN: type; end; 4 byte 2. Operator a. Aritmatika + \* / Div Mod Penjumlahan Pengurangan Perkalian Pembagian Hasil bagi Sisa bagi b. Perbandingan <<= >>= = <> Lebih kecil dari Lebih kecil atau sama dengan Lebih besar dari Lebih besar atau sama dengan Sama dengan Tidak sama dengan c. Logika not and or xor 3. Komentar { komentar } (\* komentar \*) 4. Lain-lain Const Type True False Contoh penggunaan Input Output 1. Contoh penggunaan printf dan scanf #include int data1; int main(void) { printf("nilai awal variabel adalah %d",data1); printf("\n sebelum proses memasukkan nilai \n masukkan nilai"); scanf("%d",&data1); printf("sesudah proses memasukkan nilai \n"); printf("maka nilai input menjadi %d\n",data1); } 2. Contoh penggunaan cout dan cin #include int data1; int main(void) { cout<<"sebelum proses memasukkan nilai, data1=0"<>data1; cout<<"<a href="#include <iostream.h> 2. #include <conio.h> 3. const float phi=3.14159; 4. int main() 5. { 6. Float phi = 3.14; 7. cout<<"Nilai Phi adalah "<>phi; 10. cout<<"\nNilai phi yang baru adalah "< #include const float mewah=0.005; const float menengah=0.002; float pajak; int harga; int main() { cout<<"Masukkan harga barang\n"; cin>>harga; if

(30000)Operator merupakan symbol yang biasa dilibatkan atau dipakai dalam program untuk melakukan sesuatu operasi atau manipulasi. Operand adalah objek dari operator. Operator dan operand akan membentuk ekspresi. Ekspresi ini dapat membentuk suatu pernyataan. Berikut ini penggolongan operator- operator standar yang terdapat dalam bahasa pemrograman C++. a. Operator Aritmatika Operator aritmatika yang disediakan oleh C++ antara lain perkalian, pembagian, modulus, penjumlahan, pengurangan, yang berturut turut diwakili oleh symbol \*,/,%,+,-. Contoh program : #include Int a=12, b=24, c; void main() { c=a+b; cout<<"a + b = "<Operator penugasan yang berupa symbol sama dengan (=) berfungsi untuk memberikan suatu nilai ke suatu variable. c. Operator Penaikan dan Penurunan Operator ini berhubungan dengan operator aritmatika. Operator penaikan (increment) dan operator penurunan (decrement) berturut-turut diwakili oleh symbol ++ dan --. Operator ini bisa diletakkan di belakang atau didepan operand. Contoh program : #include Int a=12; void main() { a++; cout<<"Nilai dari a++ : "<>, &, |, ^, ~ Contoh program : #include Int a=12, b; void main() { b = a >> 1; cout<<"Nilai dari a + b = "<lebih dari >= lebih dari atau sama dengan != tidak sama dengan <kurang dari == sama dengan <= kurang dari atau sama dengan Contoh program : #include Int a; void main() { a = 12>3; cout<<"Nilai dari 12 > 3 = "< 3 = "< 3 -? 1, yang berarti benar bahwa 12 lebih besar dari 3. 12 3 ? 0, yang berarti salah kalau 12 lebih kecil dari 3. f. Operator Logika Operator logika digunakan untuk menghubungkan dua buah operand menjadi sebuah ungkapan kondisi. Operator yang digunakan: &&(and), ||(or), !(not). Contoh program : #include int a=25, b; void main() { b =(a>25)&&(a<30); cout<<"=(a>25)&&(a<30)"<Penyeleksian Satu Kasus, dengan menggunakan struktur if : Pada penyeleksian satu kasus, kondisi akan diseleksi oleh statemen if. Bila kondisi bernilai benar (true), maka aksi yang akan dikerjakan dapat dituliskan pada baris pernyataan dalam cout<<"pernyataan";. Bila kondisi bernilai salah (false), maka tidak ada aksi yang akan dikerjakan dan program selesai. Gambar diagram alir penyelesaian satu kasus untuk struktur IF ditunjukkan dalam dan struktur

penulisan dalam bahasa C++ dibawah ini. IF kondisi? benar salah Aksi  
 Gambar : Diagram Alir Struktur Penyeleksian Satu Kasus (IF-) if kondisi aksi; contoh program : void `main() { int angka; cout<<" angka yang dimasukkan : ";cin>>angka; if (angka> 0) cout<<" Angka anda adalah positif"; } b. Penyeleksian Dua Kasus, menggunakan struktur if - else : Dalam struktur if-else, aksi1 akan dilaksanakan jika kondisi bernilai benar (true), dan jika kondisi bernilai salah (false) maka aksi2 yang akan dilaksanakan. Statemen else menyatakan ingkaran (negation) dari kondisi. Gambar diagram alir penyelesaian dua kasus untuk struktur if-else ditunjukkan dalam gambar dan struktur penulisan dalam bahasa C++ dapat dilihat dibawah ini. IF kondisi? benar salah aksi2 ELSE aksi1 Gambar : Diagram Alir Struktur IF-ELSE if kondisi then aksi1 else aksi2 contoh program : #include void main() { int angka; cout<<"angka yang dimasukkan : ";cin>>angka; if (angka>0) cout<<"Angka anda adalah positif"; else cout<<"Angka anda adalah negatif"; } Gambar : Struktur Bahasa C Untuk Penyeleksian Dua Kasus \(IF-ELSE\) c. Penyeleksian Tiga Kasus atau Lebih \(Penyeleksian Tersarang\) Untuk penyeleksian tiga kasus atau lebih juga menggunakan struktur if-else sebagaimana halnya permasalahan dua kasus. Gambar diagram alir penyelesaian tiga kasus untuk struktur if-else dan struktur penulisan dalam bahasa C dapat dilihat dari gambar berikut. IF benar kondisi1 salah aksi1 IF benar kondisi2 aksi2 salah aksi3 Gambar : Diagram alir dari struktur tiga kasus if-else if (tersarang) if kondisi1 then { aksi1 } else if (kondisi2) { aksi2 } else if (kondisi3) { aksi3 } else { aksi..N } contoh program : #include <iostream.h> #include <conio.h> void main() { int nilai; cout<<" Nilai Anda : ";cin>>nilai; if (nilai>= 85) { cout<<"Nilai Anda A"; } else if (nilai>= 66) { cout<<"Nilai Anda B"; } else if (nilai>= 50) { cout<<"Nilai Anda C"; } else if (nilai>= 40) { cout<<"Nilai Anda D"; } else { cout<<"Nilai Anda E" } getch(); } Gambar : Struktur Bahasa C untuk Penyeleksian Tiga Kasus IF-ELSE (tersarang) d. Struktur case. Struktur ini dapat digunakan untuk menganalisa penyeleksian dua kasus atau lebih dan bentuknya adalah lebih sederhana daripada struktur if- else ataupun bentuk if-else bersarang. Jika dalam program, dibutuhkan penyeleksi kondisi dengan banyak alternative (lebih dari dua), biasanya digunakan struktur case yang terdapat di dalam pernyataan switch (nama_variabel_yang_diseleksi. Pernyataan switch dengan case mempunyai format sebagai berikut: switch (variabel) { case nilai1_variabel : pernyataan_1;break; case nilai2_variabel : pernyataan_2;break; . . default : pernyataan_n; break; } Gambar : Struktur Bahasa C Untuk Seleksi CASE Pernyataan_1 hanya akan dilaksanakan apabila nilai variabel sama dengan nilai1_variabel. Pernyataan_2 hanya akan dilaksanakan apabila nilai variabel sama dengan nilai2_variabel. Pernyataan_n hanya akan dilaksanakan apabila nilai variabel tidak ada yang sesuai dengan nilai-nilai variabel penguji di atasnya. Jadi : pernyataan_1 hanya akan dilaksanakan apabila nilai variabel=nilai1_variabel. Pernyataan_2 hanya akan dilaksanakan apabila nilai variabel=nilai2_variabel. Pernyataan_n hanya akan dilaksanakan apabila nilai variabel tidak ada yang sesuai dengan nilai-nilai variabel penguji di atasnya. contoh program : #include #include void main() { int golongan, status; float gaji_pokok, tunjangan, total; cout<<"Golongan Anda [1,2,3,4] : ";cin>>golongan; cout<<"Status [0=tidak kawin,1=kawin]: ";cin>>status; switch(golongan) { case 1 : gaji_pokok = 2000000; break; case 2 : gaji_pokok = 1500000; break; case 3 : gaji_pokok = 1000000; break; case 4 : gaji_pokok = 7000000; break; } if (status=1) tunjangan=gaji_pokok*2%; else tunjangan=0; total=gaji_pokok+tunjangan; cout<<"Gaji Pokok : "<30 Kegemukan`



```
<< "\nrata2 = " << rata2; } Tugas Praktikum 4a. 1. Tulislah kode program
berikut ke dalam file Faktorial.cpp #include <iostream.h> #include
<conio.h> //Program penghitung faktorial int main() { //kamus int i = 1;
int hasilfaktorial =1; int faktorial; //Algoritma cout<<" Masukkan nilai
faktorial\n"; cin>>faktorial; while (i<=faktorial) {
hasilfaktorial=hasilfaktorial*i; i=i+1; } cout<<"Hasilnya adalah <adalah
tipe terstruktur yang terdiri dari sejumlah elemen-elemen yang bertipe
sama. Banyaknya elemen dalam suatu larik ditunjukkan oleh suatu indeks
yang harus merupakan tipe data yang menyatakan keterurutan, misalnya
integer atau karakter (karena ia menyatakan posisi data). Tiap-tiap elemen
di larik dapat diakses langsung melalui indeksnya. Suatu larik memiliki
jumlah elemen yang jumlahnya tetap, sehingga jumlah elemen larik tidak
dapat diubah selama pelaksanaan program. Larik dapat berupa larik satu
dimensi ataupun lebih dari satu dimensi. Matriks merupakan contoh larik
yang memiliki dimensi lebih dari satu. Struktur penyimpanan matriks di
dalam memori utama yang setiap individu elemennya diakses dengan
menggunakan dua buah indeks (yang biasanya dikonotasikan dengan baris
dan kolom). Karena matriks sebenarnya adalah larik, konsep umum dari
larik juga berlaku untuk matriks, yaitu: • Kumpulan elemen yang bertipe
sama, dapat berupa tipe dasar (integer, real, boolean, char, dan
string), atau tipe terstruktur seperti record. • Setiap elemen data dapat
diakses secara langsung jika indeksnya (baris dan kolom) diketahui. •
Merupakan struktur data yang statik, artinya jumlah elemennya sudah
dideklarasikan terlebih dahulu di dalam bagian DEKLARASI dan tidak bisa
diubah selama pelaksanaan program Elemen matriks diakses melalui
indeks baris dan indeks kolomnya. Jika indeks baris dinyatakan dengan i
dan indeks kolom dinyatakan dengan j, maka notasi algoritmik untuk
mengakses elemen pada baris i dan kolom j adalah nama_matriks[i,j]. 1.
Array satu dimensi Yaitu kumpulan elemen-elemen identik yang tersusun
dalam satu baris. Elemen- elemen tersebut memiliki tipe data yang sama,
tetapi isi dari elemen itu boleh berbeda. Berikut ini adalah konsep array
satu dimensi: 0 123 456789 17 21 33 1 48 0 2 16 72 9 Bentuk umum :
tipeData NamaArray [JumlahElemen] = {}; int Bola [10] = { 17, 21, 33,
dst}; Contoh: unsigned int TinggiBadan [100];//deklarasi array bool Hadir
[5] = {true, true, false, false};//pemberian nilai array pada inisialisasi
Pendeklarasian array diawali dengan penyebutan tipe data diikuti nama
variabel array, diikuti jumlah elemen. Jika array hendak diberi nilai awal,
nilai-nilai awal dituliskan dalam inisialisasi. Jika inisialisasi kurang dari
jumlah elemen array, sisanya akan diinisialisasikan dengan nol. Elemen
array diakses dengan menyebutkan indeks posisi elemen tersebut dalam
array. Elemen pertama array memiliki indeks 0. unsigned int TinggiBadan
[100] //tinggi badan memiliki 100 elemen TinggiBadan [0] = 7;
//mengeset elemen pertama TinggiBadan TinggiBadan [5] = 16;
//mengeset elemen keenam temp = TinggiBadan[99]; //mengopi elemen
terakhir ke variabel temp Contoh program 1 Result = result + billy[n] 2.
Array dua dimensi Sering digambarkan sebagai sebuah matriks dan
merupakan perluasan dari sebuah array satu dimensi. Jika array satu
dimensi hanya terdiri dari sebuah baris dengan beberapa kolom elemen
maka array dua dimensi terdiri dari beberapa baris dan beberapa kolom
elemen yang bertipe sama sehingga dapat digambarkan sebagai berikut: 0
123 456789 1 87 65 4 6 7 23 81 10 0 17 21 33 1 48 0 2 16 72 9 2 3 Int
Matriks[4][10] Bentuk umum : tipeData NamaArray [JumlahBaris]
[JumlahKolom] = { {}, {}, ... {}, }; contoh double Matrix [4][4];//
pemesanan tempat array bool Papan [2][2] = { {true, false}, {false, true}
}; contoh program 2 Output yang diinginkan : Tabel 1 : Data Kelulusan
```

Jurusan 1992 1993 1994 1995 1. Teknik Informatika 35 45 80 120 2. Manajemen Informatika 100 110 70 101 3. Teknik Komputer 10 15 20 17

Tugas Praktikum 5 1. Salinlah program 1 2. Pada program 1 pengisian array melalui inisialisasi, buatlah program seperti program 1 tetapi pengisian array melalui fungsi utama/pengesean array. 3. Buatlah program menggunakan for yang menghasilkan output:  $A[1, 1] = 1$   $A[1, 2] = 2$   $A[2, 1] = 3$  subject: tugas5/kelas/nama  $A[2, 2] = 4$   $B[1, 1] = 5$   $B[1, 2] = 6$   $B[2, 1] = 7$   $B[2, 2] = 8$  Jawaban Telah diperiksa Tanggal : Modul 2

Dengan ini mahasiswa telah menyetujui dan menyatakan bahwa aplikasi telah berjalan serta selesainya pengisian jawaban dari tiap pertanyaan yang ada. Pengajar Praktikum Mahasiswa Nilai BAB VI METHOD, FUNGSI & PROSEDUR Tujuan Praktikum : [Setelah Mengikuti praktikum ini, Mahasiswa diharapkan mampu untuk:](#) 1. [Mengenal dan memahami fungsi dan procedure dalam Bahasa C++](#) 2. [Mengerti dan memahami](#) tentang deklarasi,return, parameter dan bentuk 3. Dapat menggunakan dalam pemrograman C++

A. Method [Method adalah fungsi atau prosedur yang dibuat oleh seorang programmer](#) di dalam [suatu unit /class. Method dapat dibagi menjadi fungsi dan prosedur. Fungsi adalah bagian atau sub dari program yang mempunyai algoritma tertentu dalam menyelesaikan suatu masalah dengan mengembalikan hasil. Prosedur adalah bagian atau sub dari program yang mempunyai algoritma tertentu dalam menyelesaikan suatu masalah tanpa mengembalikan suatu nilai hasil. Secara umum method dalam C adalah sebuah fungsi.](#) Deklarasi sebuah [method](#) Method terdiri atas dua bagian yakni : 1. Method declaration 2. Method body ? Contoh sebuah method : ? Penggambaran deklarasi method adalah sebagai berikut : [Return type Setiap fungsi menghasilkan suatu nilai dengan tipe data tertentu. Apabila terdapat suatu fungsi yang tidak menghasilkan suatu nilai apapun maka bagian return type ini diganti dengan void.](#) Berikut adalah contoh fungsi yang tidak menghasilkan apapun : void CetakLayar() { printf("Program Sarjana Teknik Informatika"); } Method Name [Setiap fungsi dalam pemrograman mempunyai identitas nama fungsi. Penamaan fungsi / method mengikuti aturan](#) : a. [Tidak menggunakan spasi atau menggunakan spasi dengan diganti "\\_"](#) (misal method Hitung\_Integral). b. Tidak menggunakan kata – kata yang telah dipakai oleh compiler (reserved words). Parameter Bagian parameter diisi dengan parameter – parameter fungsi yang diperlukan. Parameter apabila lebih dari satu akan dipisahkan dengan tanda koma (","). [Parameter](#) dapat dibagi menjadi 2 : a. Parameter Aktual, adalah nilai yang dimasukkan ke dalam pemanggilan fungsi atau prosedur. b. Parameter Formal, adalah deklarasi variable yang digunakan sebagai parameter pada suatu fungsi atau prosedur. B. Fungsi Apabila kita membuat program yang terlalu panjang, membuat kita akan kesulitan membaca dan memahami program. Sehingga kita perlu memecah program tersebut menjadi beberapa bagian (modul) yang memudahkan kita untuk mencari kesalahan program, memperbaiki, dan membuat dokumentasi program. Untuk membuat modul tersebut C++ menyediakan fungsi. Fungsi berguna untuk mengumpulkan statemen yang dapat dijalankan menjadi satu dalam suatu blok dan menjalankannya kembali hanya dengan menuliskan nama fungsi yang menampungnya. Fungsi juga dapat dipakai untuk menampung baris-baris perintah yang sering dipakai dalam sebuah program. Deklarasi fungsi dapat dilakukan dengan dua cara yaitu : ? Header fungsi tanpa parameter atau fungsi tanpa return value ? Header fungsi dengan parameter atau fungsi dengan return value

1. Header fungsi tanpa parameter atau fungsi tanpa return value Bentuk Umum : TipeData namaFungsi Contoh : Int JumlahIsi() double Random() void Clear() Blok fungsi tanpa parameter :

Tipe data namaFungsi () { ..... } Contoh program 1 : #include <int a,b; int tambah() { return a+b; }; int kali() { return a\*b; }; int main() { cout<<"masukkan nilai a = ";cin>>a; cout<<"masukkan nilai b = ";cin>>b; cout<<"hasil a+b = "> ) Contoh : Int Jumlahkan(int a, int b) double ArcCos(double x) Blok fungsi tanpa parameter Tipe data namaFungsi (parameter) { ..... } Contoh program 2 : #include int tambah(int a, int b) { return a+b; }; int kali(int a, int b) { return a\*b; }; int main() { cout<<"masukkan nilai a = ";cin>>a; cout<<"masukkan nilai b = ";cin>>b; cout<<"hasil a+b = "> <#include <iostream.h> #include <conio.h> //Program untuk menampilkan hasil //Perhitungan segitiga menggunakan prosedur //Kamus int Alas, Tinggi; void hitungSegitiga (int A, int T) { float hasil; hasil=0.5\*A\*T; cout<<" Luas Segitiga : ">>hasil<<"\n"; } //Algoritma void main() { Alas=5; Tinggi=6; cout<<" Alas : ">>Alas<<"\n"; cout<<"Tinggi:>>Tinggi<<"\n"; hitungSegitiga(Alas,Tinggi); getch(); } Tampilkan keluaran program : Jelaskan bagaimana jalannya prosedur serta parameter – parameter yang terlibat : 2. Buatlah file cobaFungsi.cpp lalu jalankan, amati, dan jelaskan. #include <iostream.h> #include <conio.h> //Program untuk menampilkan hasil //Perhitungan segitiga menggunakan fungsi int Alas, Tinggi; float hitungSegitiga (int A, int T) { float hasil; hasil=0.5\*A\*T; return hasil; } //Algoritma void main() { Alas=5; Tinggi=6; cout<<" Alas : ">>Alas<<"\n"; cout<<"Tinggi:>>Tinggi<<"\n"; cout<<"Luas Segitiga : ">>hitungSegitiga(Alas,Tinggi)<<"\n"; getch(); } Tuliskan hasil keluaran program : Jelaskan bagaimana jalannya Fungsi serta parameter – parameter yang terlibat : Jelaskan perbedaan fungsi dan prosedur dari contoh kode di atas. Telah diperiksa Tanggal : Modul 1, 2 Dengan ini mahasiswa telah menyetujui dan menyatakan bahwa aplikasi telah berjalan serta selesainya pengisian jawaban dari tiap pertanyaan yang ada. Pengajar Praktikum Mahasiswa Nilai Tugas Praktikum 6b : 1. Buatlah fungsi tanpa parameter menghitung luas lingkaran dan volume tabung. 2. Buatlah fungsi dengan parameter menghitung luas lingkaran dan volume tabung. Output: Menu 1. Hitung Luas Lingkaran 2. Hitung Volume tabung Pilihan Anda = 1 Menghitung Luas Lingkaran Masukkan phi = 3.14 Masukkan jari-jari lingkaran = 10 Luas lingkaran = 314 atau Pilihan Anda = 2 Menghitung Volume Tabung Masukkan phi = 3.14 Masukkan jari-jari tabung = 10 Masukkan tinggi tabung = 10 Luas lingkaran = 3140 Jawaban : Telah diperiksa Tanggal : Modul 1, 2 Dengan ini mahasiswa telah menyetujui dan menyatakan bahwa aplikasi telah berjalan serta selesainya pengisian jawaban dari tiap pertanyaan yang ada. Pengajar Praktikum Mahasiswa Nilai BAB VII POINTER & REFERENCE Tujuan Praktikum : • Mahasiswa dapat menggunakan Pointer • Mahasiswa dapat menggunakan Pointer dengan argument fungsi call by reference • Memahami penggunaan pointer ke fungsi A. Dasar Teori Pointer adalah suatu variabel yang fungsinya menyimpan alamat (address). Jadi pointer merupakan fasilitas untuk mengakses suatu variabel dengan memakai address. Pada bab sebelumnya kita sudah mengenal variabel, bedanya variabel hanya berisi nilai, dan bukan alamat. Jika kita mempunyai sebuah variabel dengan tipe data tertentu, maka untuk mendapatkan alamat dari variabel tersebut adalah dengan menggunakan operator &. Alamat inilah yang kemudian akan disimpan kedalam variabel yang bertipe pointer. tipe data \*nama\_pointer; Untuk mendeklarasikan variabel sebagai sebuah pointer, kita harus menambahkan tanda asterisk (\*) di depan nama variabel. int A; //deklarasi variabel biasa int \*ptrA; //deklarasi variabel pointer Tipe data diatas berguna untuk menyatakan bahwa pointer yang kita deklarasikan akan ditempati oleh data dengan tipe tertentu. Misal kita punya variabel X

dengan tipe long, dan punya pointer P dengan tipe long. Kita dapat memerintahkan pointer P untuk menunjuk ke alamat yang ditempati oleh variabel X. long X; long \*P; P=&X; //deklarasi variabel X dengan tipe long //pointer P //memerintahkan pointer P untuk menunjuk alamat dari variabel X. // Contoh program, [Nama File](#) : pointer .cpp #include <stdio.h> #include <conio.h> void main() { int A; //deklarasi var biasa int \*ptrA; //deklarasi var pointer clrscr(); A=23; ptrA=&A; printf("\n A = %d",A); //isi dari var A printf("\n&A = %u",&A); //alamat memory var A printf("\n\*(&A) = %u",\*(&A)); //isi di alamat pada var A printf("\nptrA = %u",ptrA); //isi dari var pointer ptrA printf("\n\*ptrA = %d",\*ptrA); //isi di alamat pada var ptrA printf("\n&ptrA = %u",&ptrA); //alamat memory var ptrA getch(); } B. Kegunaan Pointer

- [Calling by reference](#)
- [Mengembalikan lebih satu nilai dari satu fungsi](#)
- [Mengirimkan array dan string dari suatu fungsi ke fungsi lain](#)
- [Memudahkan dalam manipulasi array](#)
- [Manipulasi memory dalam system](#)
- [Membuat struktur data dinamis seperti linklist, tree, graph dan sebagainya](#)

• Nama array adalah alamat memory (pointer) pertama dari elemen array tersebut. Jika A adalah array berdimensi satu maka alamat memory dari elemen pertama adalah &A[0] atau A saja. Sehingga &A[i] sama dengan (A+i). Maka A[i] pun sama dengan \*(A+i)

- Berikut contoh program tentang bagaimana mengakses data array menggunakan konsep pointer : #include #include void main() { int Nilai[5]={55,75,60,90,80}; clrscr(); for(int i=0;i<5;i++) { //akses dengan indeks array printf("\nNilai[%d] = %d",i,Nilai[i]); //akses dengan pointer printf("\n\*(Nilai+%d)= %u",i,\*(Nilai+i)); } getch ();

• Contoh program penerapan pointer untuk argumen fungsi call by reference :

```
/*Nama File : tukar2.c Prinsip Call by reference*/ #include #include void tukar(int *a,int *b) { int temp; temp=*a; *a=*b; *b=temp;} void main() { clrscr(); int x=3, y=5; printf("\nNilai x dan y semula"); printf("\nx = %d y = %d",x,y); tukar(&x,&y); printf("\nNilai x dan y kemudian"); printf("\nx = %d y = %d",x,y); getch();}
```

Tugas Praktikum 7 : 1. Salinlah contoh program dengan nama Pointer.Cpp diatas. Buatlah dengan versi C++ yang lama, yaitu versi cout dan cin. 2. Tunjukkan dimana letak perbedaannya setelah program dijalankan dan menampilkan hasil. Jawaban : Telah diperiksa Tanggal : Modul 1, 2 Dengan ini mahasiswa telah menyetujui dan menyatakan bahwa aplikasi telah berjalan serta selesainya pengisian jawaban dari tiap pertanyaan yang ada. Pengajar Praktikum Mahasiswa Nilai BAB VIII STRUCTURE Tujuan Praktikum : ? Mahasiswa dapat memahami struktur structure ? Mahasiswa dapat membuat pengolahan data dengan structure A. Dasar Teori [Structure/struktur adalah kumpulan elemen-elemen data yang digabungkan menjadi satu kesatuan. Masing-masing elemen data dikenal dengan sebutan field/ kolom. Field data tersebut dapat memiliki tipe data yang sama ataupun berbeda. Walaupun field-field tersebut berada dalam satu kesatuan, masing-masing field tersebut dapat diakses secara individual. Field-field tersebut digabungkan menjadi satu dengan tujuan untuk kemudahan dalam operasinya. Misalnya kita ingin mencatat data-data mahasiswa dan pelajar dalam sebuah program, untuk membedakannya dapat membuat sebuah record mahasiswa yang terdiri dari field- field NIM, nama, alamat, dan IPK serta sebuah record pelajar yang terdiri dari field-field Nama, NomorUrut, alamat dan JumlahNilai. Dengan demikian akan lebih mudah membedakan keduanya.](#) B. Deklarasi [struktur](#) Pendeklarasian structure selalu diawali oleh kata baku struct diikuti nama structure dan deklarasi field-field di antara kurung kurawal. Bentuk Umum: struct namaStruct Contoh: struct Mahasiswa { tipeData1 Field1; { char NIM[10]; tipeData2 Field2; char Nama[20]; ..... char Alamat[10]; tipeDataN FieldN;}; float IPK; }; Contoh

program : Tugas Praktikum 8 : 1. Salinlah contoh program diatas 2.  
 Buatlah program sehingga memberikan output sebagai berikut : Form data karyawan ===== Nama : Agus Hermanto Gaji : 2000000 Status (0 belum menikah, 1 sudah, 2 janda): 1 Jumlah anak : 3  
 ===== Masukkan data lagi? (Y/T) : y =====  
Nama : Lia Ilawan Gaji : 1500000 Status (0 belum menikah, 1 sudah, 2 janda): 2 Jumlah anak : 2  
 ===== Masukkan data lagi? (Y/T) : t =====

Jawaban : Telah diperiksa Tanggal : Modul 1, 2 Dengan ini mahasiswa telah menyetujui dan menyatakan bahwa aplikasi telah berjalan serta selesainya pengisian jawaban dari tiap pertanyaan yang ada. Pengajar Praktikum Mahasiswa Nilai BAB IX STRING DAN CHARACTER Tujuan Praktikum Memahami penggunaan string dan character dalam C++, serta mengetahui penggunaan fungsi-fungsi untuk manipulasi string. A. Dasar Teori Pada C++ tidak ada tipe variable elemen yang spesifik untuk menyimpan string. Untuk keperluan ini dapat digunakan array dengan tipe char, dimana berisi elemen dengan tipe char. Perlu di ingat bahwa tipe char digunakan untuk menyimpan 1 karakter, karena itu array dari char digunakan untuk menyimpan string. Contoh : char jenny [20]; Dapat menyimpan sampai dengan 20 karakter : Penyimpanan karakter-nya dapat direpresentasikan seperti dibawah ini : Perhatikan, karakter NULL ('\0') selalu disertakan diakhir string untuk indikasi akhir dari string. B. Inisialisasi string Sama halnya seperti array-array sebelumnya, inisialisasi pada string sbb : char mystring[] = { 'H', 'e', 'l', 'l', 'o', '\0' }; Contoh diatas, merupakan inisialisasi 6 buah elemen bertipe char, yaitu Hello dan karakter null '\0'. Untuk menentukan nilai konstan, pada string digunakan tanda kutip ganda ("), sedangkan untuk karakter kutip tunggal ('). String yang diapit oleh kutip ganda sudah mengandung karakter Null pada akhir string, contoh : char mystring [] = { 'H', 'e', 'l', 'l', 'o', '\0' }; char mystring [] = "Hello"; // contoh diatas merupakan deklarasi array mystring yang berisi 6 elemen. Pemberian nilai pada string Sama halnya seperti pemberian nilai pada array-array sebelumnya, untuk array dengan tipe char dapat dituliskan : mystring[0] = 'H'; mystring[1] = 'e'; mystring[2] = 'l'; mystring[3] = 'l'; mystring[4] = 'o'; mystring[5] = '\0'; C. Fungsi-Fungsi dalam String (cstring.h) 1. Fungsi strcpy Cara pemberian nilai seperti contoh diatas sangat tidak praktis. Umumnya untuk pemberian nilai pada array bertipe char digunakan fungsi strcpy. strcpy (string copy) mendefinisikan cstring (string.h) library dan dapat dipanggil dengan cara : strcpy (string1, string2); Baris perintah diatas menyebabkan isi daristring2 di-copy ke string1. string2 dapat berupa array, pointer, atau konstanta string. Contoh : Output : // setting value to string J. Soulie Perhatikan, header harus disertakan agar bisa menggunakan fungsi strcpy. Bisa juga menggunakan fungsi sederhana seperti misalkan diberi nama setstring, dengan operasi yang sama seperti strcpy. Contoh : // setting value to string Output : J. Soulie Metode lain yang dapat digunakan untuk inisialisasi nilai yaitu input stream (cin). Dalam kasus ini, nilai string ditentukan oleh user saat eksekusi program. Ketika menggunakan cin, biasanya digunakan metode getline, Pemanggilannya sebagai berikut: cin.getline (char buffer[], int length, char delimiter = '\n'); dimana, buffer adalah alamat untuk menyimpan input, length adalah maksimum panjang buffer,dan delimiter adalah karakter yang digunakan untuk menentukan input akhir, dengan default-atau dengan ('\n'). Contoh : // cin with strings

Output : [What's your name? Juan Hello Juan. Which is your favourite team? Inter Milan I like Inter Milan too.](#) Perhatikan kedua pemanggilan `cin.getline`, menggunakan identifier yang sama (`mybuffer`). Sama halnya seperti penggunaan operator extraction, sehingga dapat dituliskan : `cin >> mybuffer`; Instruksi diatas dapat berjalan, hanya saja mempunyai keterbatasan bila dibandingkan dengan `cin.getline`, diantaranya : ? Dapat menerima 1 kata saja (bukan kalimat lengkap). ? Tidak diperkenankan untuk memberikan ukuran buffer. Akan menyebabkan program tidak stabil jika user meng-input lebih besar dari kapasitas array yang ada.

2. Fungsi `Strlen` Merupakan fungsi manipulasi string yang digunakan untuk menghitung panjang string. Sehingga fungsi ini memberikan nilai balik panjang string. Sintaks: `size_t strlen (const char* string)`; Contoh cara pengoperasiannya, yaitu: Panjang = `strlen(bunga)`;

3. Fungsi `Strcat` Merupakan fungsi yang digunakan untuk menggabungkan string. Fungsi ini memberikan nilai balik string yang sudah digabung. Sintaks: `char* strcat (char* dest, const char* src)`; Contoh cara pengoperasiannya, yaitu: `Strcat(str1, str2) Strcat("aku", "belajar")`

4. Fungsi `strcmp` Merupakan fungsi yang digunakan untuk membandingkan dua buah string yaitu `string1` dan `string2`. Sintaks: [`strcmp: int strcmp \(const char\* string1, const char\* string2\)`](#); Contoh: `Strcmp (str1, str2) Strcmp("aku", "aku")= 0`

5. `Strrev` Merupakan string yang digunakan untuk membalik string. Contoh: `Strrev(str)`; Str ?kamu `Strrev(str) Str ? umak` Cttn : `char*` sama dengan `char[]`

D. Konversi string ke tipe lainnya String dapat berisi data dengan tipe lain seperti angka. Contoh "1977". `cstdlib (stdlib.h)` library menyediakan 3 fungsi yang dapat menangani hal tersebut: ? [`atoi: converts string to int type.`](#) ? [`atol: converts string to long type.`](#) ? [`atof: converts string to float type.`](#) Fungsi-fungsi ini menerima 1 parameter dan mengembalikan nilainya kedalam tipe yang diminta (`int`, `long` or `float`). Fungsi ini dikombinasikan dengan metode `getline` pada `cin`. Contoh : // `cin` and atau\* functions Output : Enter price: 2.75 Enter quantity: 21 Total price: 57.75

Tugas Praktikum 9 1. Salinlah contoh program dibawah ini. Jalankan dan perhatikan hasilnya. //contoh program penggunaan fungsi string dengan `getline`.

```

Datagetline.cpp #include #include int main() {
string nama,kota,telp,alamat; cout<<"Nama Mahasiswa cout<<"Kota
Mahasiswa cout<<"Alamat cout<<"Nomer Telepon : "; getline(cin,nama);
: "; getline(cin,kota); : "; getline(cin,alamat); : "; getline(cin,telp);
clrscr(); cout<<"< , gantilah tipe variable yang semula string dengan
char, hilangkan penggunaan getline gunakan struktur cin biasanya.
Jalankan program dan amati hasilnya.
3. Pada program Datapribadi.cpp, isikan data isian dengan lebih dari 2 suku kata. Misalkan nama mahasiswa : Yudha Bimantara Setyawan, lanjutkan untuk isian berikutnya. Amati hasilnya. Apa yang terjadi? Mengapa demikian? Jawaban : Telah diperiksa
Tanggal : Modul 1, 2 Dengan ini mahasiswa telah menyetujui dan menyatakan bahwa aplikasi telah berjalan serta selesainya pengisian jawaban dari tiap pertanyaan yang ada.
Pengajar Praktikum Mahasiswa Nilai BAB X SEARCHING Tujuan Praktikum : ? Mahasiswa mengerti dan memahami binary search pada bahasa C++ ? Dapat menggunakan binary search dalam pemrograman Dalam Pencarian Binary Search, Data yang ada harus diurutkan terlebih dahulu berdasarkan suatu urutan tertentu yang dijadikan kunci pencarian. Adalah teknik pencarian data dalam dengan cara membagi data menjadi dua bagian setiap kali terjadi proses pencarian. Prinsip pencarian biner adalah: Data diambil dari posisi 1 sampai posisi akhir N Kemudian cari posisi data tengah dengan rumus: \(posisi awal + posisi akhir\) / 2. Kemudian data yang dicari dibandingkan dengan data yang di tengah, apakah sama atau lebih kecil, atau lebih

```

besar? Jika lebih besar, maka proses pencarian dicari dengan posisi awal adalah posisi tengah + 1 Jika lebih kecil, maka proses pencarian dicari dengan posisi akhir adalah posisi tengah - 1 Jika data sama, berarti ketemu contoh program [binary search : //binary searching, //program bisa jalan jika data sudah terurut #include <iostream.h> #include <conio.h> int data\[10\]={1,3,4,7,12,25,40,65,78,90}; int binary\\_search\(int cari\) { int l,r,m; int n=10; l=0; r=n-1; int ketemu=0; while \(l<=r && ketemu==0\) { m=\(l+r\)/2; if \(data\[m\]==cari\) ketemu=1; else if\(cari<data\[m\]\) r=m-1; else l=m+1; } if\(ketemu==1\) return 1; else return 0; } void main\(\) { clrscr\(\); int cari, hasil; cout<<" masukan data yang ingin di cari= "; cin>>cari; hasil = binary\\_search\(cari\); if\(hasil==1\) { cout<<" data ada !"<<endl; } else if\(hasil==0\) cout<<" data tidak ada !"<<endl; getch\(\); }](#)

Tugas Praktikum 10. 4. Salinlah contoh program diatas, tuliskan hasil outputnya. 5. Dari program diatas, kesimpulan apa yang anda dapat mengenai searching? Jawaban : Telah diperiksa Tanggal : Modul Dengan ini mahasiswa telah menyetujui dan menyatakan bahwa aplikasi telah berjalan serta selesainya pengisian jawaban dari tiap pertanyaan yang ada.

Pengajar Praktikum Mahasiswa Nilai BAB XI SORTING Tujuan Instruksional: ? Mengerti dan memahami tentang pengurutan dalam bahasa C++ ? Dapat menggunakan sorting bubble sort dan selection short dalam pemrograman C++ A. Dasar Teori Sorting adalah proses menyusun kembali data yang sebelumnya telah disusun dengan suatu pola tertentu ataupun secara acak, sehingga menjadi tersusun secara teratur menurut aturan tertentu.

Pada umumnya ada 2 macam pengurutan, yaitu: ? Pengurutan secara ascending (urut naik). ? Pengurutan secara descending (urut turun).

Algoritma untuk melakukan sorting juga ada berbagai macam, antara lain:

? Teoretis : Computational complexity theory, Big O notation, Total order, Stability, Comparison sort. ? Exchange sorts : Exchange sort, Bubble sort, Cocktail sort, Comb sort, Gnome sort, Quick sort. ? Selection sorts :

Selection sort, Heap sort, Smooth sort. ? Insertion sorts : Insertion sort, Shell sort, Tree sort, Library sort, Patience sorting. B. Sorting Algorithm ?

Bubble Sort • Metode sorting paling mudah, namun paling lambat

dibandingkan dengan yang lain. • Bubble sort mengurutkan data dengan cara membandingkan elemen sekarang dengan elemen berikutnya. • Bisa dilakukan baik dari kepala array maupun ekor array. • Proses yang berlangsung, jika: ? Ascending: jika elemen sekarang lebih besar daripada

elemen berikutnya, maka kedua elemen tersebut ditukar. ? Descending: jika elemen sekarang lebih kecil daripada elemen berikutnya, maka kedua

elemen tersebut ditukar. • Hal ini akan terlihat seperti penggeseran angka, perbandingan, kemudian jika memenuhi syarat kemudian tukar. • Proses

penukaran ini akan terus dilakukan hingga seluruh array telah diperiksa. • Contoh fungsi bubble sort: //Bubble Sort void bubble (int a[], int n) { int i,j; for (i=n;i>=1;i--) { for (j=2;j<a[j]) tukar (a,j-1,j); } } ? Selection Sort

• Kombinasi sorting dan searching. • Untuk setiap proses, akan dilakukan dengan mencari elemen dari posisi yang belum diurutkan dan kemudian

memilih elemen yang memiliki nilai terkecil atau terbesar yang akan ditukarkan ke posisi yang tepat di dalam array. • Misalnya untuk putaran

pertama, akan dicari data dengan nilai terkecil dan data ini akan ditempatkan pada indeks terkecil, pada putaran kedua akan dicari data

kedua terkecil, dan akan ditempatkan di indeks kedua, begitu seterusnya hingga tidak ada data yang dicari lagi. • Selama proses, perbandingan

dan perubahan hanya dilakukan pada indeks perbandingan saja, pertukaran data secara fisik terjadi pada akhir proses. • Contoh fungsi

selection sort: //Selection Sort void selection (int a[],int n) { int i,j,pos; for (i=1;i

Tujuan Instruksional: Setelah Mengikuti praktikum ini, Mahasiswa

diharapkan mampu untuk: 1. Mengenal dan memahami matriks dalam Bahasa C++ 2. Dapat menggunakan matriks dalam pemrograman C++ ? ? ? Adalah struktur penyimpanan data di dalam memori utama Setiap individu elemennya diacu dengan menggunakan dua buah indeks (baris dan kolom) Merupakan sebuah array yang memiliki dimensi lebih dari satu Strukturnya praktis dan mudah digunakan, jika indeksnya diketahui (baris dan kolom), maka akan diketahui pula posisi relatif elemen di dalam kumpulannya ? Gambar dibawah menunjukkan sebuah matriks yang terdiri dari 5 buah baris dan 4 buah kolom ? Elemen yang diarsir menyatakan elemen matrik untuk baris 2 dan kolom 4 ? ? ? Jika matriks pada gambar sebelumnya adalah M, maka elemen yang diarsir adalah M[2,3] Cara mengacu dengan peubah indeks hanya benar apabila nilai dari peubah indeks tersebut sudah terdefinisi. Misalkan : M[p,2] , M[1,k] , M[p,q] Hanya benar dengan syarat p, k, q sudah terdefinisi Elemen Matriks Matriks di Memori ? Representasi matriks di dalam memori tetap sebagai deretan sel berurutan ? Contoh representasi matriks 4x4 di memori ? Program Mencetak Matriks dengan bahasa C++ ini, ini dia tampilannya ; ? Kemudian dibawah ini adalah source code/codingnya : #include <iostream> using namespace std; void baca\_matriks(int mat[10][10], int baris, int kolom) { int i, j; for( i =0;i<baris; i++) for(j=0;j<kolom;j++) { cout <<"Data [" <<(i+1) <<" , " <<(j+1) <<" ] = "; cin >> mat[i][j]; } } void cetak\_matriks(const int A[10][10], int baris, int kolom) { for(int i = 0;i<baris;i++) { for(int j = 0;j<kolom;j++) cout <<" " << A[i][j] ; cout << endl; } } int main() { int m, n; int matriks[10][10]; int jumlah[10][10]; cout <<"Banyak baris : " ; cin >> m; cout <<"Banyak kolom : " ; cin >> n; cout <<"Data matriks \n"; baca\_matriks(matriks,m,n); cout <<"Cetak Matriks \n"; cetak\_matriks(matriks,m,n); system(" PAUSE"); return 0; } Tugas Praktikum 12 . + Buat algoritma dengan pseudocode untuk penjumlahan matriks diatas! Dan diterapkan dalam pemrograman! Jawaban : Telah diperiksa Tanggal : Modul Dengan ini mahasiswa telah menyetujui dan menyatakan bahwa aplikasi telah berjalan serta selesainya pengisian jawaban dari tiap pertanyaan yang ada. Pengajar Praktikum Mahasiswa Nilai DAFTAR PUSTAKA Kadir, A. 2003. Pemrograman C++, Membahas Pemrograman Berorientasi Obyek Menggunakan Turbo C++ dan Borland C++. Penerbit Andy Yogyakarta. Mahmudy, W.M. 2004. Modul Praktikum Semester Ganjil 2004–2005 Pemrograman C++. Jurusan Matematika Fakultas MIPA, Universitas Brawijaya Malang. NN. 2009. Laporan Praktikum Dasar Komputer dan Pemrograman. Jurusan Elektro, Fakultas Teknik Universitas Brawijaya Malang. NN. 2010. Modul Praktikum Algoritma dan Pemrograman, Teknik Informatika, Fakultas Teknik Universitas Nusantara Kediri. Raharjo, Budi, 2010. Pemrograman C++ Revisi Pertama, Mahir dan Cepat menjadi master C++, Penerbit Informatika, Bandung. Sanjaya, D. 2005. Asyiknya Belajar Struktur Data di Planet C++. Penerbit PT. Elex Media Komputindo Kelompok Gramedia. Jakarta 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66