Nama

Nyoman Tri Darma W.

NIM

: 2305551052

Kelompok

: 09

**MODUL I**

**DASAR PEMROGRAMAN, FUNGSI, DAN PROSEDUR**

**Tujuan**

1. Untuk mengetahui dan memahami tipe dasar dalam bahasa C/C++ dan keperluan memori dari masing-masing tipe data.
2. Untuk mengetahui dan memahami pengertian, kegunaan, serta fungsi simbol-simbol dalam flowchart.
3. Untuk mengetahui dan memhami konsep variabel lokal dan variabel global dan juga perbedaan antara keduanya.
4. Untuk mengetahui dan memahami konsep fungsi dan prosedur dalam bahasa C/C++.
5. Untuk mengetahui dan memhami konsep passing by value dan passing by reference dalam bahasa C/C++.
6. Untuk mengetahui dan memahami seni penulisan program.

**Tugas Pendahuluan**

1. Sebutkan dan jelaskan secara rinci tipe data dasar yang digunakan dalam bahasa C/C++ dan keperluan memori untuk masing-masing tipe data serta berikan contoh pendeklarasian tipe data tersebut!
2. Jelaskan pengertian dan kegunaan dari *flowchart!* Lalu gambarkan dan jelaskan fungsi masing-masing simbol yang terdapat dalam *flowchart!*
3. Jelaskan tentang konsep variabel lokal dan variabel global serta perbedaannya!
4. Jelaskan tentang konsep fungsi dan prosedur, perbedaan keduanya, serta aplikasinya dalam pemrograman dengan C/C++!
5. Jelaskan tentang konsep *passing by value, passing by reference* dan perbedaannya, serta buatlah contoh program dengan Bahasa C/C++!
6. Jelaskan yang dimaksud dengan seni penulisan program beserta contohnya!
7. Buatlah *flowchart* dan *pseudocode* (notasi algoritma) dari program pada soal di bawah ini, lalu lakukan *trace* terhadap *flowchart* yang Anda buat sehingga yakin bahwa *flowchart* tersebut sudah benar.

**Jawaban**

# Tipe Data Dasar pada Bahasa Pemrograman C/C++

Tipe data adalah kategori data yang digunakan untuk menyimpan informasi dari suatu variabel dan digunakan untuk menggambarkan jenis karakter yang ada dalam sebuah program. Selain itu, tipe data juga dapat menentukan operator yang dapat digunakan dalam operasi program. Oleh karena itu, ketika kita membuat program, penting untuk menentukan tipe data yang akan digunakan sebelum mendefinisikan variabel atau operator yang akan digunakan. Dalam Bahasa C, terdapat berbagai jenis tipe data yang dapat digunakan, termasuk tetapi tidak terbatas pada:

## 1.1 Tipe data Interger

Tipe data integer adalah jenis data yang digunakan untuk menyimpan nilai numerik dalam rentang tertentu. Tipe data ini memungkinkan penyimpanan angka bulat, baik positif maupun negatif. Ada tiga varian tipe data integer yang dapat digunakan, yaitu Short, Int, dan Long. Tipe data Short memiliki ukuran 2 byte dan dapat menampung nilai dari -32768 hingga 32767. Tipe data Int memiliki ukuran 4 byte dan dapat menampung nilai dari -2147483648 hingga 2147483647. Sedangkan tipe data Long juga memiliki ukuran 4 byte dan dapat menampung nilai dari -9223372036854775808 hingga 9223372036854775807. Dengan menggunakan tipe data integer, kita dapat mendeklarasikan variabel untuk menyimpan nilai numerik dengan batasan tertentu. Berikut adalah contoh pendeklarasian tipe data integer :

#include <stdio.h>

int main(){

int umur;

umur = 20;

return(0);

}

**Kode Program 1. 1** source code tipe data interger

Kode program 1.1 merupakan *source code* dalam program tipe data *integer*. Dalam program terdapat tipe data yang digunakan yaitu int atau integer yang digunakan untuk menyimpan data dalam bentuk bilangan bulat pada variabel tanggal.

## 1.2 Tipe data Char

Char atau Character adalah jenis data dasar dalam bahasa pemrograman C yang digunakan untuk menyimpan variabel karakter. Tipe data ini hanya dapat menampung angka dari 0 hingga 9, huruf alfabet (baik huruf kecil maupun huruf kapital), dan tanda baca atau simbol seperti (!, ?, @). Setiap variabel tipe data character hanya dapat memuat satu karakter tunggal, dan membutuhkan satu byte memori. Penulisan variabel tipe data char menggunakan tanda petik tunggal (‘) atau ganda (“) sebelum dan sesudah karakter yang ingin disimpan. Contohnya dapat dilihat dalam pendeklarasian tipe data character berikut.

**Kode program 1. 2** Source Code Program Type Data Character

#include <stdio.h> int main(){

char alphabet;

alphabet = 'T';

return(0);

}

Kode program 1.2 Merupakan *source code* dalam program tipe data *char*. Dalam program terdapat tipe data yang digunakan yaitu karakter ‘M’ yang disimpan pada variabel huruf.

## 1.3 Single Floating-Point

Tipe data Single Floating-Point atau Float digunakan untuk menyimpan data numerik atau angka dengan presisi lebih tinggi daripada tipe data integer. Float umumnya digunakan untuk menyimpan bilangan desimal atau pecahan dalam bahasa pemrograman C. Selain Float, dalam C juga terdapat tipe data Double dan Long Double yang dapat digunakan untuk menyimpan bilangan pecahan. Perbedaan utama antara ketiganya terletak pada tingkat presisi angka yang dapat disimpan. Float dapat menyimpan 6 angka desimal setelah koma, Double dapat menyimpan 15 angka desimal setelah koma, dan Long Double dapat menyimpan 19 angka desimal setelah koma. Berikut adalah contoh pendeklarasian tipe data float:

#include <stdio.h> int main(){

float nilai\_uas;

nilai\_uas = 91.45;

return(0);

}

**Kode program 1. 3** Source Code Program Type Data Float

Kode program 1.3 Merupakan *source code* dalam program tipe data *float*. Dalam program terdapat tipe data yang digunakan yaitu *float* digunakan untuk menyimpan data dalam bentuk bilangan desimal atau pecahan pada variabel jumlah.

## 1.4 Tipe Data *Double*

Tipe data double memiliki kemiripan dengan tipe data float, namun memiliki presisi yang lebih tinggi, dikenal sebagai double precision. Tipe data double mampu menyimpan hingga 15 angka di belakang koma, menjadikannya lebih akurat daripada tipe data float. Ada juga variasi dari tipe data double, yaitu long double, yang memiliki presisi lebih tinggi dibandingkan float dan double. Tipe data double memerlukan 8 byte memori, sedangkan tipe data long double memerlukan 12 byte. Pendeklarasiannya menggunakan kata kunci double, seperti contoh berikut:

**Kode program 1. 4** Source Code Program Type Data Double

#include <stdio.h>

int main(){

double nilai\_praktikum;

nilai\_praktikum = 95.955678;

return(0);

}

Kode Program 1.4 merupakan contoh pendeklarasian tipe data *double* dalam bahasa C. Tipe data *double* digunakan untuk menyimpan data dalam bentuk bilangan desimal atau pecahan dengan banyak angka di belakang koma pada variabel latitude.

# Flowchart

Flowchart adalah representasi visual dari urutan proses menggunakan simbol-simbol khusus. Flowchart ini memetakan setiap langkah dengan rinci, menggambarkan hubungan antar instruksi atau proses dalam suatu program, atau menyajikan langkah-langkah yang diperlukan untuk mengeksekusi suatu proses dalam program. Tujuan dari flowchart adalah menjelaskan proses secara terperinci, membantu mengilustrasikan hubungan antara langkah-langkah dalam suatu masalah, dan menyederhanakan proses tersebut sehingga lebih mudah dimengerti. Secara umum, setiap perancangan flowchart melibatkan tiga elemen utama: input, proses, dan output, yang diwakili oleh simbol-simbol tertentu.

**Tabel 2. 1** Simbol-Simbol dalam *Flowchart*

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Simbol** | **Nama** | **Fungsi** |
| 1. |  | *Terminator* | Sebagai simbol awal (*start*) dan akhir (*end*) dalam flowchart. |
| 2. |  | *Process* | Sebagai simbol aliran proses dalam sistem. |
| 3. |  | *Input Output* | Sebagai simbol yang menyatakan proses input atau output. |
| 4. |  | *Preparation* | Sebagai penyedia tempat penyimpanan dalam pemberian nilai awal. |
| 5. |  | *Manual Input* | *Flowchart* Sebagai simbol proses yang tidak dilakukan computer. |
| 6. |  | *Decision* | Sebagai simbol yang menunjukkan percabangan dalam aliran proses. |
| 7. |  | *Flow Direction* | Sebagai penghubung simbol satu dengan yang lainnya. |
| 8 |  | *Connector* | Sebagai simbol untuk keluar-masuk atau penyambungan proses pada lembar / halaman yang sama. |
| 9 |  | *Predefine proses* | Sebagai simbol untuk pelaksanaan suatu bagian (sub-program)/procedure |
| 10 |  | *Connector* | Sebagai symbol untuk keluar-masuk atau  penyambung proses pada lembar / halaman yang berbeda. |

# Konsep Variable Lokal dan Global

Variabel adalah suatu penunjuk yang digunakan untuk menyimpan nilai dalam bahasa pemrograman C. Nilai dalam variabel dapat diubah sesuai dengan kebutuhan pengguna selama program berjalan. Dalam mendefinisikan variabel dalam bahasa pemrograman C, terdapat dua tipe penempatan yang dapat digunakan, yaitu variabel lokal dan variabel global. Variabel lokal dan variabel global memiliki makna sebagai berikut:

## 3.1 Variabel Lokal

Variabel lokal adalah variabel yang dinyatakan di dalam suatu fungsi tertentu. Variabel ini hanya dapat diakses oleh fungsi di mana variabel tersebut dideklarasikan, karena variabel tersebut hanya akan dikenali dalam batasan fungsi itu sendiri. Variabel lokal juga dikenal dengan istilah Automatic Duration, yang berarti variabel tersebut diciptakan ketika kompilator menjalankan deklarasinya di dalam suatu lingkup tertentu dan dihapus saat keluar dari lingkup tersebut.

|  |
| --- |
| #include <stdio.h>  void cetakusia()  {  int usia = 17 ;  printf("USIA = %d\n", usia);  }  int main()  {  cetakusia(); return 0;  } |

**Kode Program 3. 1** Contoh penggunaan variabel lokal.

Kode program 3.1 merupakan contoh penggunaan variabel lokal yang terdapat pada fungsi cetakusia(). Dalam kode program tersebut, terdapat variabel usia dengan tipe data int, dimana variabel usia tersebut hanya terdefinisi pada fungsi cetakusia(), sementara pada fungsi lain tidak terdefinisi variabel usia.

## 3.2 Variabel Global

Variabel global adalah variabel yang bisa diakses dan dipanggil oleh fungsi serta prosedur di seluruh program. Variabel ini memiliki cakupan yang meluas, memungkinkan penggunaannya dalam berbagai fungsi atau prosedur tanpa perlu mendeklarasikannya berulang kali. Hal ini membantu mengurangi repetisi kode, karena programmer tidak perlu menuliskan variabel yang sama berulang kali dalam berbagai fungsi atau prosedur. Variabel global memiliki sifat Static Duration, artinya variabel ini diciptakan saat program dimulai dan dihapus saat program berakhir.

|  |
| --- |
| #include <stdio.h> int usia = 17;  void cetakusia()  {  printf("USIA = %d\n", usia);  }  int main()  {  cetakusia();  printf("USIA = %d", usia);  return 0;  } |

**Kode Program 3. 2** Contoh penggunaan variabel global

Kode program 3.2 merupakan contoh penggunaan variabel global. Dalam kode program tersebut, terdapat variabel usia dengan tipe data int, dimana variabel usia tersebut terdefinisi pada semua fungsi, baik pada fungsi cetakusia(), maupun pada fungsi main().

## 3.3 Perbedaan Variabel Lokal dan Global

Variabel lokal dan global adalah dua tipe variabel dalam pemrograman dengan perbedaan utama dalam cakupan (scope) penggunaannya. Variabel lokal hanya dapat diakses di dalam fungsi atau blok kode di mana variabel tersebut dideklarasikan. Dengan kata lain, variabel lokal hanya ada dalam lingkup fungsi tempat ia dibuat, dan tidak dapat diakses oleh fungsi-fungsi lain dalam program. Di sisi lain, variabel global dapat diakses oleh semua fungsi dan blok kode dalam program. Variabel ini dideklarasikan di luar fungsi-fungsi dan dapat digunakan di mana saja dalam program. Namun, karena variabel global dapat diakses oleh banyak bagian dalam program, perubahan nilainya bisa sulit dijejaki dan memahami, sehingga perlu dikelola dengan hati-hati agar menghindari konflik dan kesalahan yang tidak terduga dalam program.

# Konsep Fungsi dan Prosedur

Fungsi dan prosedur dalam bahasa pemrograman C adalah segmen program yang dibuat secara terpisah dari blok utama atau fungsi main, dan memiliki tujuan khusus untuk mengeksekusi tugas tertentu. Biasanya, fungsi dan prosedur ini digunakan untuk menyelesaikan masalah spesifik berdasarkan parameter yang diberikan. Keberadaan fungsi dan prosedur bertujuan untuk menghindari duplikasi kode, mengurangi kompleksitas program, serta membuat program lebih terstruktur dan mudah dimengerti. Dengan pendekatan ini, keuntungannya adalah memungkinkan penghematan waktu dalam pengembangan program.

## 4.1 Fungsi

Fungsi adalah serangkaian perintah yang dikelompokkan dalam suatu ruang lingkup untuk menjalankan tugas spesifik. Penggunaan fungsi memungkinkan kita mengurangi pengulangan penulisan kode yang seringkali muncul, karena kode-kode tersebut dapat ditempatkan dalam sebuah fungsi dan dipanggil sesuai kebutuhan. Dengan fungsi, program menjadi lebih terorganisir dan efisien, membuat alur program menjadi lebih mudah dipahami. Selain itu, fungsi juga dapat menerima input dan menghasilkan output sesuai dengan kebutuhan program. Dalam bahasa C, fungsi pertama kali dieksekusi saat program mulai berjalan. Berikut adalah contoh penggunaan fungsi dalam bahasa pemrograman C.

|  |
| --- |
| #include <stdio.h>  float luas\_lingkaran ( int r){ float luas;  luas = 3.14 \* r \* r; return luas;  } |

**Kode Program 4. 1** Contoh Fungsi

Kode program, 4.1 Merupakan *Source Code* penggunaan fungsi dalam bahasa pemrograman C. Terdapat fungsi luas lingkaran yang didalamnya berisi parameter bertipe data *float* yaitu int r, serta nilai yang dikembalikan yang juga bertipe data *float* yaitu hasil.

## 4.2 Prosedur

Prosedur dalam sebuah program merupakan blok kode terpisah yang bertindak sebagai subprogram atau bagian kecil dari program utama. Dalam prosedur, perintah-perintah dapat dikelompokkan tetapi prosedur tidak dapat menghasilkan nilai yang dapat diproses oleh perintah-perintah tersebut. Oleh karena itu, prosedur dideklarasikan dengan tipe data void, yang berarti tidak mengembalikan nilai. Penggunaan prosedur umumnya terbatas pada program-program terstruktur, mengikuti prinsip modularitas, di mana program dibagi ke dalam bagian-bagian yang lebih kecil untuk mempermudah pembacaan dan pemahaman kode serta untuk menyembunyikan kompleksitas detail program. Contoh penggunaan prosedur dalam bahasa pemrograman C dapat dilihat di bawah ini.

|  |
| --- |
| #include <stdio.h>  void luas(int r){  int hasil = 3.14 \* r \* r;  printf(“Luas lingkaran = %d\n”, hasil);  }  int main(){  int jari\_jari;  printf(“Masukan jari-jari lingkaran:”);  scanf(“%d,&jari-jari);  luas(jari\_jari) return 0;  } |

**Kode Program 4. 2** Contoh Prosedur

Kode program, 4.2 Merupakan *Source Code* penggunaan prosedur dalam bahasa pemrograman C dalam fungsi luas lingkaran menggunakan prosedur.

# Passing By Value dan Passing By Refrence

*Passing By value* dan *Passing By Referensi* merupakan proses dalam penggunaan parameter dari fungsi ataupun yang terdapat dalam Bahasa pemrograman C ataupun bahasa pemrograman lainnya. Parameter adalah sebutan untuk nilai pada saat pendefinisian. Pada umumnya, parameter yang berlaku pada Bahasa pemograman C/C++ dapat menjadi dua jenis diantaranya adalah sebagai berikut.

## 5.1 Passing By Value

*Passing by value* merupakan proses penyalinan dari parameter *actual* (dari pemanggil atau *celler*) ke parameter *formal* (*function*). Perubahan yang terjadi di dalam *function* tidak akan mempengaruhi nilai asli pemanggilnya. Berikut merupakan contoh pengaplikasian *Passing By Value* pada bahasa pemrograman C/C++.

|  |
| --- |
| #include <stdio.h>  void swap (int x, int y){ int z = x;  x = y; y = z;  }  int main(){  int x = 10, y = 20; swap (x,y);  printf("x = %d dan y = %d\n", x, y); return 0;  } |

**Kode Program 5. 1** Passing By Value

Kode program di atas merupakan contoh implementasi penggunaan *passing by value* pada bahasa pemrograman C/C++. Pada program ini sebuah fungsi *swap* yang menerima 2 parameter yakni *int* a dan *int* b, yang bertujuan untuk menukar isi dari a dan b. Bila kita jalankan program ini, maka *output* yang dihasilkan program ini adalah (x = 10 dan y = 20). Hal ini terjadi dikarenakan perubahan nilai hanya terjadi di dalam fungsi *swap*. Dimana bila kita mencetak data a dan b di sana, nilai memang benar sudah terganti. Namun dikarenakan *copy value* dari nilai tersebut, maka tidak mempengaruhi x dan y dari sisi pemanggil.

## 5.2 Passing By Reference

*Passing by reference* merupakan upaya untuk melewatkan alamat dari suatu variabel ke dalam fungsi, jadi yang diberikan kepada fungsi lain tersebut adalah alamat (*memory addres*) dari variabel yang dilewatkan. Dalam *passing by reference* saat fungsi dipanggil maka nilai *variable* yang diinputkan akan berubah

|  |
| --- |
| #include <stdio.h>  void swap (int\* x, int\* y){ int z = \*x;  \*x = \*y;  \*y = z;  }  int main(){  int x = 10, y = 20; swap (&x,&y);  printf("x = %d dan y = %d\n", x, y); return 0;  } |

**Kode Program 5. 2** Passing By Reference

Kode program di atas merupakan contoh implementasi penggunaan *passing by reference* pada bahasa pemrograman C/C++. Hasil *output* dari contoh program ini adalah (x = 20 dan y = 10).

# Seni Penulisan Program

Seni penulisan program merujuk pada keterampilan dan keahlian dalam menciptakan kode komputer yang efisien, efektif, dan mudah dimengerti. Ini melibatkan kemampuan untuk merencanakan dan merancang solusi untuk masalah, mengorganisasikan logika program dengan cara yang terstruktur dan mudah dipahami, serta menuliskan kode dengan cara yang bersih dan efisien. Seni penulisan program juga mencakup kemampuan untuk membuat algoritma yang efisien, memilih struktur data yang sesuai, dan mengoptimalkan kinerja program. Selain itu, seni penulisan program juga mencakup keterampilan dalam mendokumentasikan kode agar orang lain dapat dengan mudah memahaminya, serta kemampuan untuk menguji dan memperbaiki kesalahan (debugging). Dengan seni penulisan program yang baik, seorang pengembang dapat menciptakan perangkat lunak yang stabil, andal, dan mudah dimengerti oleh pengguna dan pengembang lainnya.

# Flowchart, Pseudocode, dan Trace Program Modul 1

Pada modul ini membahas mengenai dasar-dasar dalam bahasa pemrograman. Program dibahas adalah program kalkulator, mengitung luas dan keliling bangun datar yang sudah ditentukan serta menghitung volume dan luas permukaan bangun ruang yang sudah ditentukan juga sebelumnya. Berikut merupakan penjelasan dari flowchart dan pseudocodenya.

## Program Kalkulator

Program kalkulator merupakan sebuah program sederhana yang dimana terdapat lima jenis operasi dasar matematika, yaitu penjumlahan, pengurangan, perkalian, pembagian, dan modulus. Program ini diawali dengan melakukan input dua buah nilai, kemudian proses operasi matematika yang ingin dilakukan, dan output berupa hasil dari operasi yang dilakukan pada kedua nilai tersebut. Berikut flowchart, pseudocode, dan trace dari program kalkulator.

### Flowchart Program Kalkulator

Flowchart adalah urutan atau langkah-langkah dari proses program yang digambarkan dalam bentuk simbol. Flowchart digunakan untuk mempermudah dalam menyelesaikan suatu masalah hingga tuntas. Berikut flowchart dari program kalkulator.

|  |
| --- |
|  |

**Gambar 7.** 1 *Flowchart* Sub-Program *open()* Kalkulator Sederhana

Pada Gambar 7.1 merupakan flowchart sub-program open()dari bagian program kalkulator. Pada sub-program ini ditampilkan daftar nama anggota dari kelompok 9 dan kemudin sub-program akan diakhiri dengan perintah end. Setelah dimulai, akan di ditampilkan nama program dan nama kelompok kemudian selesai.

|  |
| --- |
|  |

**Gambar 7. 2** Flowchart Int Main() Program Kalkulator

Gambar 7.2. merupakan flowchart sub-program main() dari program kalkulator sederhana. Pada bagian awal program terdapat pendeklarasian variabel menu, kemudian menampilkan sub-program open(), lalu akan ditampilkan pilihan menu perhitungan. Pada bagian ini, user diminta untuk menginputkan pilihan program yang ingin dijalankan. Bagian ini pula sudah dilengkapi dengan validasi, dimana jika user menginputkan karakter diluar angka 1 sampai 5, maka sistem akan meminta inputan secara berulang hingga inputan sesuai dengan yang diminta. Apabila angka yang diinputkan sudah sesuai, maka akan ditampilkan sub-program close(). Selanjutnya, jika user menginputkan angka 1, maka sistem akan menampilkan sub-program penjumlahan(), Begitupun selanjutnya.

|  |
| --- |
|  |

**Gambar 7. 3** Flowchart Penjumlahan Program Kalkulator

Pada Gambar 7.3. merupakan flowchart sub-program penjumlahan() dari bagian program kalkulator. Setelah dimulai, akan dideklarasikan variabel angka1, angka2, dan hasil1. user diminta untuk menginputkan angka, jika yang diinputkan merupakan huruf, maka sistem akan menampilkan input salah lalu mengulang proses input angka1 hingga yang diinput sesuai dengan yang diminta. Begitu pula saat menginput angka2, jika yang diinputkan merupakan huruf, maka sistem akan menampilkan input salah lalu mengulang proses input angka2 hingga yang diinput sesuai dengan yang diminta. Setelah itu akan dilakukan proses aritmatika penjumlahan hasil1 = angka1+angka2. Kemudian ditampilkan hasil1 dan sub- program penjumlahan() selesai.

|  |
| --- |
|  |

**Gambar 7. 4** Flowchart Pengurangan Program Kalkulator

Pada Gambar 7.4 merupakan flowchart sub-program pengurangan() dari bagian program kalkulator. Setelah dimulai, akan dideklarasikan variabel angka1, angka2, dan hasil. user diminta untuk menginputkan angka1, jika yang diinputkan merupakan huruf, maka sistem akan menampilkan input salah lalu mengulang proses input angka2 hingga yang diinput sesuai dengan yang diminta. Begitu pula saat menginput angka2, jika yang diinputkan merupakan huruf, maka sistem akan menampilkan input salah lalu mengulang proses input angka2 hingga yang diinput sesuai dengan yang diminta. Setelah itu akan dilakukan proses aritmatika penjumlahan hasil = angka1– angka2. Kemudian ditampilkan hasil dan sub- program pengurangan()selesai.

|  |
| --- |
|  |

**Gambar 7. 5** Flowchart Perkalian Program Kalkulator

Pada Gambar 7.5 merupakan flowchart sub-program perkalian() dari bagian program kalkulator. Setelah dimulai, akan dideklarasikan variabel angka1, angka2, dan hasil. user diminta untuk menginputkan angka2, jika yang diinputkan merupakan huruf, maka sistem akan menampilkan input salah lalu mengulang proses input angka1 hingga yang diinput sesuai dengan yang diminta. Begitu pula saat menginput angka2, jika yang diinputkan merupakan huruf, maka sistem akan menampilkan input salah lalu mengulang proses input angka2 hingga yang diinput sesuai dengan yang diminta. Setelah itu akan dilakukan proses aritmatika penjumlahan hasil = angka1\*angka2. Kemudian ditampilkan hasil3 dan sub- program perkalian() selesai.

|  |
| --- |
|  |

**Gambar 7. 6** Flowchart Pembagian Program Kalkulator

Pada Gambar 7.6 merupakan flowchart sub-program pembagian() dari bagian program kalkulator. Setelah dimulai, akan dideklarasikan variabel angka1, angka2, dan hasil. user diminta untuk menginputkan angka1, jika yang diinputkan merupakan huruf, maka sistem akan menampilkan input salah lalu mengulang proses input angka1 hingga yang diinput sesuai dengan yang diminta. Begitu pula saat menginput angka2, jika yang diinputkan merupakan huruf, maka sistem akan menampilkan input salah lalu mengulang proses input angka2 hingga yang diinput sesuai dengan yang diminta. Setelah itu akan dilakukan proses aritmatika penjumlahan hasil=angka1/angka2. Kemudian ditampilkan hasil4 dan sub- program pembagian() selesai.

|  |
| --- |
|  |

**Gambar 7. 7** Flowchart Modulus Program Kalkulator

Pada Gambar 7.7 merupakan flowchart sub-program modulus()dari bagian program kalkulator. Setelah dimulai, akan dideklarasikan variabel angka1, angka2, dan hasil. user diminta untuk menginputkan angka1, jika yang diinputkan merupakan huruf, maka sistem akan menampilkan input salah lalu mengulang proses input angka1 hingga yang diinput sesuai dengan yang diminta. Begitu pula saat menginput angka2, jika yang diinputkan merupakan huruf, maka sistem akan menampilkan input salah lalu mengulang proses input angka2 hingga yang diinput sesuai dengan yang diminta. Setelah itu akan dilakukan proses aritmatika penjumlahan hasil = angka1 % angka2. Kemudian ditampilkan hasil5 dan sub- program modulus() selesai.

|  |
| --- |
|  |

**Gambar 7. 8** Repeat Program Kalkulator

Pada Gambar 7.8 merupakan flowchart sub-program ulang() dari bagian program kalkulator. Setelah dimulai, akan dideklarasikan variabel menu, lalu akan ditampilkan pilihan menu, apakah 1 Ulang atau 0 Exit. user diminta untuk menginputkan angka 1 atau 0. Sistem akan memvalidasi input menu, jika yang diinput bukan angka 1 atau 0, maka akan ditampilkan input salah lalu sistem akan mengulang input menu hingga sesuai dengan yang diminta. Jika menu yang diinputkan adalah 1, maka akan berlanjut ke sub-program close() selanjutnya kembali ke sub-program main(). Namun, jika menu yang diinput adalah 0, maka akan berlanjut ke sub-program close() selanjutnya ke sub-program end(). Dan sub-program ulang() selesai.

|  |
| --- |
|  |

**Gambar 7. 9** Close Program Kalkulator

Gambar 7.9 merupakan flowchart sub-program close() dari program kalkulator sederhana. Jika dimulai, sistem akan menampilkan tulisan selanjutnya kemudian jika user menekan keyboard maka sistem akan membersihkan layar dari tampilan sebelumnya, yang merupakan fungsi dari perintah system (“cls”) dan sub-program close() berakhir.

|  |
| --- |
|  |

**Gambar 7. 10** End Program Kalkulator

Pada Gambar 7.10. merupakan flowchart sub-program end()dari bagian program kalkulator. Setelah dimulai, sistem akan menampilkan Terimakasih Telah Menggunakan Program Kami, Sampai Jumpa Kembali, Salam Hangat: Kelompok 9. Kemudian sub-program end() selesai

### Pseudocode Program Kalkulator

Pseudocode adalah suatu istilah dalam pemrograman untuk menuliskan sintaks, statement, algoritma, dan lain sebagainya, dalam bahasa yang mudah dipahami. pseudocode juga dapat mempermudah seorang programmer dalam menyusun sebuah program secara terstruktur dan sekuensial. Berikut pseudocode dari program kalkulator sederhana.

|  |
| --- |
| plaintext  Procedure Penjumlahan      Declare angka1, angka2, hasil as Float      Display "1. Penjumlahan"      Display "Masukan bilangan pertama: "      Input angka1      While Not IsNumeric(angka1)          Display "Yang anda masukan bukan angka!"          Display "Masukan nilainya kembali dengan benar: "          Input angka1      End While      Display "Masukan bilangan kedua: "      Input angka2      While Not IsNumeric(angka2)          Display "Yang anda masukan bukan angka!"          Display "Masukan nilainya kembali dengan benar: "          Input angka2      End While      hasil <- angka1 + angka2      Display "------------------------------"      Display "Hasil:            ", hasil  End Procedure  Procedure Pengurangan      Declare angka1, angka2, hasil as Float      Display "2. Pengurangan"      Display "Masukan bilangan pertama: "      Input angka1      While Not IsNumeric(angka1)          Display "Yang anda masukan bukan angka!"          Display "Masukan nilainya kembali dengan benar: "          Input angka1      End While      Display "Masukan bilangan kedua: "      Input angka2      While Not IsNumeric(angka2)          Display "Yang anda masukan bukan angka!"          Display "Masukan nilainya kembali dengan benar: "          Input angka2      End While      hasil <- angka1 - angka2      Display "------------------------------"      Display "Hasil:            ", hasil  End Procedure  Procedure Perkalian      Declare angka1, angka2, hasil as Float      Display "3. Perkalian"      Display "Masukan bilangan pertama: "      Input angka1      While Not IsNumeric(angka1)          Display "Yang anda masukan bukan angka!"          Display "Masukan nilainya kembali dengan benar: "          Input angka1      End While      Display "Masukan bilangan kedua: "      Input angka2      While Not IsNumeric(angka2)          Display "Yang anda masukan bukan angka!"          Display "Masukan nilainya kembali dengan benar: "          Input angka2      End While      hasil <- angka1 \* angka2      Display "------------------------------"      Display "Hasil:            ", hasil  End Procedure  Procedure Pembagian      Declare angka1, angka2, hasil as Float      Display "4. Pembagian"      Display "Masukan bilangan pertama: "      Input angka1      While Not IsNumeric(angka1)          Display "Yang anda masukan bukan angka!"          Display "Masukan nilainya kembali dengan benar: "          Input angka1      End While      Display "Masukan bilangan kedua: "      Input angka2      While Not IsNumeric(angka2)          Display "Yang anda masukan bukan angka!"          Display "Masukan nilainya kembali dengan benar: "          Input angka2      End While      hasil <- angka1 / angka2      Display "------------------------------"      Display "Hasil:            ", hasil  End Procedure  Procedure Modulus      Declare angka1, angka2, hasil as Float      Display "5. Modulus"      Display "Masukan bilangan pertama: "      Input angka1      Display "Masukan bilangan kedua: "      Input angka2      hasil <- Modulo(angka1, angka2)      Display "------------------------------"      Display "Hasil:            ", hasil  End Procedure |

Kode Program 7. 1 Pseudocode Program Kalkulator

Kode program 7.1 adalah pseudocode dari program kalkulator. Pseudocode pada kode program 7.1 berkaitan dengan flowchart dari program kalkulator. Pseudocode ini akan diimplementasikan menjadi sebuah kode program dengan versi computer readable yang akan menjalankan program kalkulator tersebut.

### Trace Program Kalkulator

Trace program kalkulator merujuk pada langkah-langkah atau jejak eksekusi yang diambil oleh program kalkulator saat dijalankan. Ini melibatkan pemantauan bagaimana program mengambil input, melakukan operasi matematika, dan menghasilkan output. Dalam konteks ini, "trace" mengacu pada pencatatan langkah-langkah yang dilakukan oleh program selama proses eksekusi.

**Tabel 7. 1** Trace Program kalkulator

|  |  |
| --- | --- |
| Input | menu = 1 (penjumlahan)  angka1 = 10  angka2 = 20  Menu = 0 (Exit) |
| Proses | hasil1 = angka1 + angka2  hasil = + 10 + 20  hasil1 = 30 |
| Output | Print hasil kalkulasi = 30  Terimakasih Telah Menggunakan Program Kami |

Tabel 7.1 merupakan trace dari program kalkulator. Program kalkulator diawali dengan pemilihan menu operasi yang ingin dilakukan, misalnya adalah penjumlahan, dilanjutkan dengan melakukan input data dua nilai yang ingin dioperasikan, yaitu 10 dan 20. Kedua nilai tersebut akan dioperasikan dan menghasilkan output dari operasi penjumlahan, yaitu 30.

## Program Bangun Datar

Program menghitung luas dan keliling pada bangun datar adalah program sederhana yang menyediakan lima pilihan bangun datar, yaitu segitiga sembarang, belah ketupat, jajargenjang, trapesium, dan lingkaran. Program ini juga menyediakan dua jenis operasi perhitungan, yaitu menghitung keliling dan luas. Program ini mengharuskan user untuk melakukan input nilai masing-masing atribut dan pilihan operasi yang diinginkan pada bangun datar yang dipilih. Output hasil akan ditampilkan setelah proses perhitungan dalam program selesai. Berikut flowchart, pseudocode, dan trace dari program keliling luas bangun datar.

### Flowchart Program Bangun Datar

Flowchart merupakan urutan atau langkah-langkah dari proses program yang digambarkan dalam bentuk simbol. Flowchart digunakan untuk mempermudah dalam menyelesaikan suatu masalah, khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. Berikut flowchart dari program keliling dan luas bangun datar

|  |
| --- |
|  |

**Gambar 7. 11** Flowchart Sub Program open() Program Bangum Datar

Pada Gambar 7.11 merupakan flowchart sub-program open() dari bagian program Bangun Datar. Setelah dimulai, akan di ditampilkan nama program dan nama kelompok kemudian selesai.

|  |
| --- |
|  |

Gambar 7. 12

|  |
| --- |
|  |

Gambar 7. 13

|  |
| --- |
|  |

Gambar 7. 14

|  |
| --- |
|  |

Gambar 7. 15

|  |
| --- |
|  |

**Gambar 7. 16** Flowchart Sub-program repeat() Program Bangun Datar

Pada Gambar 7.16 merupakan flowchart sub-program ulang() dari bagian program menghitung keliling dan luas bangun datar. Setelah dimulai, akan dideklarasikan variabel menu, lalu akan ditampilkan pilihan menu, apakah 1 Ulang atau 0.Exit. Pengguna diminta untuk menginputkan angka 1 atau 0. Sistem akan memvalidasi input menu, jika yang diinput bukan angka 1 atau 0, maka akan ditampilkan input salah lalu sistem akan mengulang input menu hingga sesuai dengan yang diminta. Jika menu yang diinputkan adalah 1, maka akan berlanjut ke sub-program close() selanjutnya kembali ke sub-program main(). Namun, jika menu yang diinput adalah 0, maka akan berlanjut ke sub-program close() selanjutnya ke sub-program end(). Dan sub-program ulang() selesai.

|  |
| --- |
|  |

**Gambar 7. 17** Flowchart Sub-program close() Program Bangun Datar

Gambar 7.17 merupakan flowchart sub-program close() dari program Bangun Datar. Jika dimulai, sistem akan menampilkan tulisan selanjutnya kemudian jika user menekan keyboard maka sistem akan membersihkan layar dari tampilan sebelumnya, yang merupakan fungsi dari perintah system (“cls”) dan sub-program close() berakhir.

|  |
| --- |
|  |

**Gambar 7. 18** Flowchart Sub-program end()Program Bangun Datar

Pada Gambar 718. merupakan *flowchart* sub-program end()dari bagian program menghitung keliling dan luas bangun datar. Setelah dimulai, sistem akan menampilkan Terimakasih. Kemudian sub-program end()selesai.

* + 1. **Pseudocode Program Bangun Datar**

Pseudocode merupakan suatu istilah dalam pemrograman untuk menuliskan sintaks, statement, algoritma, dan sebagainya, dalam bahasa yang mudah dipahami manusia. Pseudocode dapat dikatakan sebagai representasi kode program dengan versi human readable (mudah dipahami manusia). Berikut pseudocode dari program menghitung luas dan keliling bangun datar.

|  |
| --- |
|  |

**Kode Program 7. 2** Pseudocode Program Bangun Datar

* + 1. **Trace** **Program Bangun Datar**

Trace program merupakan sebuah proses untuk melacak input yang dimasukkan oleh pengguna ke dalam sebuah program serta menganalisa hasil input tersebut serta memahami proses pengolahan yang ada pada sebuah program untuk mendapatkan output yang sesuai. Berikut ini adalah trace program dari program menghitung keliling dan luas bangun datar yang dijabarkan melalui tabel di bawah ini.

**Tabel 7. 2** Trace Program Bangun Datar

|  |  |
| --- | --- |
| Input | menu = 3  operasi = 2  alas = 7  tinggi = 20 |
| Proses | float luas = alas \* tinggi  float luas = 7 \* 20  float luas = 140.00 |
| Output | Luas jajar genjang : 140.00 cm2 |

Tabel 7.2 merupakan tracing dari program menghitung luas dan keliling bangun datar. Pada tabel di atas, program melakukan pencarian nilai hasil dari luas bangun datar jajar genjang. *User* melakukan input nilai alas dan tinggi dari bangun datar jajar genjang dan akan diproses oleh program. Proses tersebut akan menghasilkan output berupa luas jajar genjang.

## Program Bangun Ruang

Program bangun ruang adalah sebuah program komputer yang dirancang untuk menghitung dan melakukan berbagai operasi terkait dengan bangun ruang tiga dimensi, seperti kubus, bola, tabung, limas, dan bangun ruang lainnya. Tujuan utama dari program ini adalah untuk membantu pengguna menghitung parameter dan properti dari bangun ruang, termasuk luas permukaan, volume, dan sifat-sifat lainnya.

### Flowchart Program Bangun Ruang

Program menghitung luas permukaan dan volume bangun ruang merupakan program yang menyediakan operasi perhitungan luas permukaan dan volume dari berbagai bangun ruang. Beberapa pilihan bangun ruang yang tersedia, yaitu tabung, bola, limas segiempat, prisma segitiga, dan kerucut. Program diawali dengan melakukan input pilihan bangun ruang. Output berupa hasil dari perhitungan luas permukaan atau volume dari bangun ruang. Berikut flowchart, pseudocode, dan trace dari program menghitung luas permukaan dan volume bangun ruang.

|  |
| --- |
|  |

**Gambar 7. 19** Flowchart Sub Program open() Program Bangum Ruang

Pada Gambar 7.19 merupakan flowchart sub-program open() dari bagian program Bangun Ruang. Setelah dimulai, akan di ditampilkan nama program dan nama kelompok kemudian selesai.

|  |
| --- |
|  |

**Gambar 7. 20** Flowchart Sub Program Tabung Program Bangum Ruang

Gambar 7.20 merupakan *flowchart* dari Sub-Program fungsi tabung yang ada pada program menghitung volume dan luas permukaan bangun ruang. Untuk *flowchart* dari fungsi lp tabung dapat dilihat pada gambar.

|  |
| --- |
|  |

**Gambar 7. 21** Flowchart Sub Program Bola Program Bangum Ruang

Gambar 7.21 merupakan flowchart dari Sub-Program fungsi lp bola yang ada pada program menghitung volume dan luas permukaan bangun ruang. Untuk flowchart dari fungsi lp bola dapat dilihat pada gambar.

|  |
| --- |
|  |

**Gambar 7. 22** Flowchart Sub Program Limas Segiempat Program Bangum Ruang

Gambar 7.22 merupakan flowchart dari Sub-Program fungsi limas yang ada pada program menghitung volume dan luas permukaan bangun ruang. Untuk flowchart dari fungsi lp limas dapat dilihat pada gambar.

|  |
| --- |
|  |

**Gambar 7. 23** Flowchart Sub Program Prisma segitiga Program Bangum Ruang

Gambar 7.23 merupakan flowchart dari Sub-Program fungsi prisma\_segitiga yang ada pada program menghitung volume dan luas permukaan bangun ruang. Untuk flowchart dari fungsi prisma\_segitiga dapat dilihat pada gambar.

|  |
| --- |
|  |

**Gambar 7. 24** Flowchart Sub Program Kerucut Program Bangum Ruang

Gambar 7.24 merupakan flowchart dari Sub-Program fungsi volume\_kerucut yang ada pada program menghitung volume dan luas permukaan bangun ruang. Untuk flowchart dari fungsi volume\_kerucut dapat dilihat pada gambar.

|  |
| --- |
|  |

**Gambar 7. 25** Flowchart Sub-program repeat() Program Bangun Ruang

Pada Gambar 7.25 merupakan flowchart sub-program ulang() dari bagian program menghitung keliling dan luas bangun ruang. Setelah dimulai, akan dideklarasikan variabel menu, lalu akan ditampilkan pilihan menu, apakah 1 Ulang atau 0.Exit. Pengguna diminta untuk menginputkan angka 1 atau 0. Sistem akan memvalidasi input menu, jika yang diinput bukan angka 1 atau 0, maka akan ditampilkan input salah lalu sistem akan mengulang input menu hingga sesuai dengan yang diminta. Jika menu yang diinputkan adalah 1, maka akan berlanjut ke sub-program close() selanjutnya kembali ke sub-program main(). Namun, jika menu yang diinput adalah 0, maka akan berlanjut ke sub-program close() selanjutnya ke sub-program end(). Dan sub-program ulang() selesai.

|  |
| --- |
|  |

**Gambar 7. 26** Flowchart Sub-program close() Program Bangun Ruang

Gambar 7.26 merupakan flowchart sub-program close() dari program Bangun Ruang. Jika dimulai, sistem akan menampilkan tulisan selanjutnya kemudian jika user menekan keyboard maka sistem akan membersihkan layar dari tampilan sebelumnya, yang merupakan fungsi dari perintah system (“cls”) dan sub-program close() berakhir.

|  |
| --- |
|  |

**Gambar 7. 27** Flowchart Sub-program end()Program Bangun Ruanag

Pada Gambar 7.27 merupakan *flowchart* sub-program end()dari bagian program menghitung keliling dan luas bangun Ruang. Setelah dimulai, sistem akan menampilkan Terimakasih. Kemudian sub-program end()selesai.

* + 1. **Pseudocode Program Bangun Ruang**

Pseudocode untuk program bangun ruang adalah deskripsi tingkat tinggi dari algoritma atau logika program dalam bentuk mirip kode pemrograman, tetapi lebih berfokus pada logika algoritma daripada sintaksis bahasa pemrograman tertentu. Ini memberikan panduan tentang cara menyelesaikan masalah dan dapat diimplementasikan dalam berbagai bahasa pemrograman.

|  |
| --- |
| plaintext  Procedure VolumeDanLuasPermukaanTabung      Declare jari\_jari, tinggi, luas\_permukaan, volume as Float      Display "Menghitung Volume dan Luas Permukaan Tabung"      Display "Masukkan jari-jari dari tabung : "      Input jari\_jari      While Not IsNumeric(jari\_jari)          Display "Karakter yang Anda inputkan salah!"          Display "Mohon inputkan angka yang benar."          Display "Masukkan jari-jari dari tabung : "          Input jari\_jari      End While      Display "Masukkan tinggi dari tabung :"      Input tinggi      While Not IsNumeric(tinggi)          Display "Karakter yang Anda inputkan salah!"          Display "Mohon inputkan angka yang benar. "          Display "Masukkan tinggi dari tabung : "          Input tinggi      End While      luas\_permukaan <- 2 \* PHI \* jari\_jari \* (jari\_jari + tinggi)      volume <- PHI \* jari\_jari \* jari\_jari \* tinggi      Display "Volume tabung adalah : ", volume      Display "Luas permukaan tabung adalah : ", luas\_permukaan  End Procedure  Procedure VolumeDanLuasPermukaanBola      Declare radius, luas\_permukaan, volume as Float      Display "Menghitung Volume dan Luas Permukaan Bola"      Display "Inputkan radius bola : "      Input radius      While Not IsNumeric(radius)          Display "Yang anda masukan bukan angka!"          Display "Inputkan nilainya lagi :"          Input radius      End While      Luas\_permukaan <- 4 \* PHI \* radius \* radius      volume <- (4/3) \* PHI \* pow(radius, 3)      Display "Volume bola adalah : ", volume      Display "Luas Permukaan bola adalah : ", luas\_permukaan  End Procedure  Procedure VolumeDanLuasPermukaanLimasSegiEmpat      Declare tinggi\_limas, panjang, lebar, luas, keliling, luas\_permukaan, volume as Float      Display "Menghitung Volume dan Luas Permukaan Limas Segi Empat"      Display "Masukkan panjang sisi alas : "      Input panjang      While Not IsNumeric(panjang)          Display "Karakter yang Anda inputkan salah!"          Display "Mohon inputkan angka yang benar. "          Display "Masukkan panjang sisi alas : "          Input Panjang      End While      Display "Masukkan tinggi limas : "      Input tinggi\_limas      While Not IsNumeric(tinggi\_limas)          Display "Karakter yang Anda inputkan salah!"          Display "Mohon inputkan angka yang benar. "          Display "Masukkan tinggi limas : "          Input tinggi\_limas      End While          Display "Masukkan lebar : "          Input lebar      While Not IsNumeric(lebar)          Display "Karakter yang Anda inputkan salah!"          Display "Mohon inputkan angka yang benar. "          Display "Masukkan lebar limas  : "          Input lebar      End While      luas <- Panjang \* lebar      keliling <- 2 \* (panjang + lebar)      luas\_permukaan <- luas + (keliling \* tinggi\_limas)/2      volume <- (luas \* tinggi\_limas) / 3      Display "Volume limas segi empat adalah : ", volume      Display "Luas permukaan limas segi empat adalah adalah : ", luas permukaan  End Procedure  Procedure VolumeDanLuasPermukaanPrismaSegitiga      Declare alas\_segitiga, tinggi\_segitiga, tinggi\_prisma, sisi\_miring, luas\_segitiga, keliling\_segitiga, luas\_permukaan, volume  as Float      Display "Menghitung Volume dan Luas Permukaan Prisma Segitiga"      Display "Masukkan sisi alas segitiga : "      Input alas\_segitiga      While Not IsNumeric(alas\_segitiga)          Display "Karakter yang Anda inputkan salah!"          Display "Mohon inputkan angka yang benar."          Display "Masukkan sisi alas segitiga : "          Input alas\_segitiga      End While      Display "Masukkan tinggi alas prisma : "      Input tinggi\_segitiga      While Not IsNumeric(tinggi\_segitiga)          Display "Karakter yang Anda inputkan salah!"          Display "Mohon inputkan angka yang benar."          Display "Masukkan tinggi alas prisma : "          Input tinggi\_segitiga      End While      Display "Masukkan sisi miring segitiga : "      Input sisi\_miring      While Not IsNumeric(sisi\_miring)          Display "Karakter yang Anda inputkan salah!"          Display "Mohon inputkan angka yang benar."          Display "Masukkan sisi miring segitiga : "          Input sisi\_miring      End While      Display "Masukkan tinggi prisma : "      Input tinggi\_prisma      While Not IsNumeric(tinggi\_prisma)          Display "Karakter yang Anda inputkan salah!"          Display "Mohon inputkan angka yang benar."          Display "Masukkan tinggi prisma : "          Input tinggi\_prisma      End While      luas\_segitiga <- 0.5 \* alas\_segitiga \* tinggi\_segitiga      keliling <- alas\_segitiga + tinggi\_segitiga + sisi\_miring      luas\_permukaan <- (2 \* luas\_segitiga) + (keliling\_segitiga \* tinggi\_prisma)      volume <- luas\_segitiga \* tinggi\_prisma      Display "Volume prisma adalah : ", volume      Display "Luas permukaan prisma adalah : ", luas\_permukaan  End Procedure  Procedure VolumeDanLuasPermukaanKerucut      Declare radius, tinggi, sisi\_miring, volume, luas\_permukaan as Float      Display "Menghitung Volume dan Luas Permukaan Kerucut"      Display "Inputkan panjang radius : "      Input radius      While Not IsNumeric(radius)          Display "Yang Anda masukkan bukan angka!"          Display "Inputkan nilainya lagi : "          Input radius      End While      Display "Inputkan sisi miring kerucut : "      Input sisi\_miring      While Not IsNumeric(sisi\_miring)      Display "Yang Anda masukkan bukan angka!"          Display "Inputkan nilainya lagi : "          Input tinggi      End While      Display "Inputkan tinggi kerucut : "      Input tinggi      While Not IsNumeric(tinggi)          Display "Yang Anda masukkan bukan angka!"          Display "Inputkan nilainya lagi : "          Input tinggi      End While      sisi\_miring <- sqrt(pow(radius, 2) + pow(tinggi, 2))      luas\_permukaan <- PHI \* radius \*(radius + sisi\_miring)      volume <- (PHI \* pow(radius,2) \* tinggi) / 3      Display "Volume kerucut adalah : ", volume      Display "Luas permukaan kerucut adalah : ", luas\_permukaan  End Procedure |

**Kode Program 7. 3** Program Bangun Ruang

Program di atas adalah serangkaian prosedur untuk menghitung volume dan luas permukaan dari beberapa bangun ruang, yaitu tabung, bola, limas segi empat, prisma segitiga, dan kerucut. Setiap prosedur meminta input dari pengguna. memvalidasi input, menghitung volume dan luas permukaan sesuai rumus yang berlaku untuk setiap bangun ruang, dan menampilkan hasilnya kepada pengguna. Program ini membantu pengguna menghitung sifat-sifat geometris dari berbagai bangun ruang tiga dimensi.

* + 1. **Trace Program Bangun Ruang**

*Trace* program bangun ruang merujuk pada urutan langkah-langkah eksekusi yang dilakukan oleh program komputer ketika dijalankan untuk menghitung volume dan luas permukaan bangun ruang tertentu.

**Tabel 7. 3** Trace Program Bangun Ruang

|  |  |
| --- | --- |
| Input | menu = 2 (tabung) radius\_tabung = 14  tinggi\_tabung = 10 menu9 = 0 (exit) |
| Proses | volume\_tabung = 3.14\*radius\_tabung\*radius  \_tabung\*tinggi\_tabung volume\_tabung = 3.14\*14\*14\*10  volume\_tabung = 6154,4  la\_tabung = 3.14\*radius\_tabung\*radius  \_tabung  la\_tabung = 3.14\*14\*14 la\_tabung = 615,44  ls\_tabung = 2\*3.14\*radius\_tabung\*ting gi\_tabung  ls\_tabung = 2\*3.14\*14\*10 ls\_tabung = 3516,8  lp\_tabung = la\_tabung + ls\_tabung  lp\_tabung = 615,44 + 3516,8  lp\_tabung = 4132,24 |
| Output | print Volume tabung adalah 6154,4  print Luas permukaan tabung adalah 4132,24  print Apakah anda ingin mengulang? (0)  print Terimakasih |

Tabel 7.3 merupakan hasil trace dari operasi menghitung volume dan luas permukaan tabung yang berada pada program menghitung volume dan luas permukaan bangun ruang. Pada saat menjalankan program, pengguna akan diminta untuk memilih bangun ruang yang ingin dihitung volume dan luas permukaannya pada menu awal program. Bangun tabung dipilih dengan menginputkan angka 1. Selanjutnya pengguna akan diminta untuk memasukkan panjang radius tabung dan tinggi tabung, dimana apabila pengguna sudah menginputkannya, maka kedua buah bilangan tersebut akan dilakukan pemrosesan sesuai dengan rumus yang sudah dideklarasikan. Kemudian program akan mengirimkan output hasil proses berupa volume dan luas permukaan tabung kepada pengguna.

DAFTAR PUSTAKA

[1] A. Wibowo, "Pass By Value dan pass By Reference," Pass By Value dan pass By Reference, 12 Maret 2018. [Online]. Available: https://www.wapwibowo.com/2018/12/pass-by-value-dan-pass-by-reference.html.

[Accessed 2023 Oktober 10].[2] Ahmad, "Variabel-global-dan-variabel-lokal-dalam-bahasa-c," 10 November 2016. [Online]. Available: https:www.mahirhoding.com/variabel-global-dan-variabel-lokal-dalam-bahasa-c/.[3] Andhika, "tipedatapemrograman," tipedatapemrograman, 10 Oktober 2023. [Online]. Available: https://andhikanurafian.web.id/tipe-data-pemorgraman-c-dasar/. [Accessed 20 Oktober 2022].[4] Andre, "BelajarC," BelajarC, 04 Februari 2018. [Online]. Available: https://www.duniailkom.com/tutorial-belajar-c-jenis-jenis-tipe-data-dalam-bahasa-c/. [Accessed 11 Oktober 2023].[5] Andre, "duniailkom," duniailkom, 11 Oktober 2023. [Online]. Available: https://www.duniailkom.com/tutorial-belajar-c-plus-plus-jenis-jenis-tipe-data-dalam-bahasa-c-plus-plus/. [Accessed 20 Oktober 2022].[6] Fajar, "belajarcpp," belajarcpp, 12 February 2020. [Online]. Available: https://www.belajarcpp.com/tutorial/cpp/tipe-data/. [Accessed 20 oktober 2022].[7] Heri, "Simbol Flowchart," Simbol Flowchart, 27 April 2017. [Online]. Available: https://salamadian.com/simbol-simbol-flowchart/. [Accessed 11 Oktober 2023].[8] Joko, "Variabel Lokal dan Variabel Global," Variabel Lokal dan Variabel Global, 16 Mei 2020. [Online]. Available: https://pakkoding.id/2020/05/16/variabel-lokal-dan-global-dalam-bahasa-pemrograman-c/. [Accessed 11 Oktober 2023].[9] A. Muhardian, "Seni Penulisan Program Bahasa C," Seni Penulisan Program Bahasa C, 20 Januari 2020. [Online]. Available: https://www.petanikode.com/c-syntak/. [Accessed 11 Oktober 2023].