

# Meningkatkan Proses Pemahaman Operasi Perhitungan Bilangan Bulat Menggunakan Pseudocode Dan Program Sederhana Pada Bahasa Pemrograman C++

**Leonurdin Aziz O L**

Universitas Amikom Yogyakarta

**Alvin Adwitya R R**

Universitas Amikom Yogyakarta

**Abstract:** *This research aims to improve understanding on integer calculations using integer calculation process through C++ programming language on students. This research is the research to complete coursework Math diskreit and as Individual Research. The research comprises several stages – stages: planning, implementation, the writing source code, trial and error code and the conclusion of the program. Success in research can be measured from (1) index increased student achievement and a minimum of 80% of students respond well to practical or theory in the category either (2) understanding on integer data types and writing code in a language improved C++ programming (3) students can easily recognize the looping line and look for the error code. Results showed that the use of integer calculations through the programming language C++ may improve the understanding of the students in computing whole numbers. It proved to be where students achieve all of the criteria listed in the research. This research is evidenced by (1) index increased student achievement in practical or theory (2) median value – a class average is increasing and shows significant results for some percent of results before its derived classes and (3) the students more easily respond to learning and understanding the operation of integer calculations.*

**Keywords:** *understanding the calculations, the integers, the programming language C++.*

**Abstrak:** Penelitian ini bertujuan untuk meningkatkan pemahaman pada perhitungan bilangan bulat dengan menggunakan proses perhitungan bilangan bulat melalui Bahasa pemrograman C++ pada mahasiswa. Penelitian ini merupakan penelitian untuk menyelesaikan tugas kuliah Matematika diskreit dan sebagai Penelitian Individual. Penelitian terdiri beberapa tahapan – tahapan yaitu: perencanaan, penerapan, penulisan source code, trial dan error code dan kesimpulan program. Keberhasilan pada penelitian bisa diukur dari (1) indeks prestasi mahasiswa meningkat dan minimal 80% mahasiswa memberikan respon baik terhadap praktikum ataupun teori dalam kategori baik (2) pemahaman pada tipe data integer dan penulisan code dalam Bahasa pemrograman C++ meningkat (3) mahasiswa dapat dengan mudah mengenali baris perulangan dan mencari error code. Hasil pada penelitian menunjukkan bahwa penggunaan perhitungan bilangan bulat melalui Bahasa pemrograman C++ dapat meningkatkan pemahaman mahasiswa dalam perhitungan bilangan bulat. Hal itu terbukti dimana mahasiswa mencapai semua kriteria yang telah tertera dalam penelitian. Penelitian ini terbukti (1) Indeks prestasi mahasiswa yang meningkat dalam praktikum ataupun teori (2) nilai rata – rata pada kelas meningkat dan menunjukkan hasil yang signifikan untuk beberapa persen dari hasil sebelumnya yang diperoleh kelas sebelumnya dan (4) mahasiswa lebih mudah merespon pembelajaran dan memahami operasi perhitungan bilangan bulat.

**Kata kunci:** pemahaman perhitungan, bilangan bulat, Bahasa pemrograman C++.

## PENDAHULUAN

Tujuan dalam penelitian ini tidak lebih untuk mengikuti perkembangan teknologi dan mengembangkan pemahaman dalam Bahasa pemrograman C++ dan Bahasa OOP (Object Oriented Programming) lainnya seperti C#, Java, Pascal, Python, PHP dan sejenisnya. Di masa

berkembang pesatnya bidang teknologi dan Pemrograman ini tak lepas dan dilandasi oleh pemahaman akan matematika seperti Logika dan Kalkulus. Maka dari itu matematika akan sangat membantu dan menjadi tumpuhan untuk melakukan perhitungan dalam Bahasa pemrograman seperti C++ ataupun pada Bahasa Pemrograman lainnya.

Bahasa Pemrograman C++ merupakan Bahasa yang sudah lama dan populer di dalam pembelajaran pada computer dan Programmer yang ingin mengembangkan system software ataupun membuat game. Oleh sebab itu pemahaman akan *bahasa pemrograman C++* ini akan sangat penting terlebih lagi ketika mahasiswa pada jurusan – jurusan yang berkaitan dengan teknologi ataupun programming. Pembelajaran Bahasa Pemrograman C++ secara teori maupun praktikum ini mempunyai peranan penting untuk meningkatkan pengetahuan mahasiswa tentang bahasa pemrograman disamping mahasiswa belajar bahasa pemrograman mahasiswa secara otomatis mempelajari tata cara penulisan Bahasa pemrograman dimana salah satu dari tipe data dari Bahasa pemrograman tersebut yaitu integer (atau bilangan bulat). Bilangan bulat bertanda ini dapat mepresentasikan nilai bilangan bulat negatif sedangkan yang tidak bertanda adakah kebalikan dari bilangan bulat yang bertanda. Bilangan bulat bertanda (positif) di dalam komputer adalah rantaiian bit, yg dimana rantaiian ini menggunakan bilangan biner. Urutan pada bit-bit yang muncul pun bervariasi tergantung dari jumlah bit – bit yang di presentasikan atau yang keluar. Bilangan bulat atau dalam Bahasa pemrograman C++ disebut tipe data integer ini mempunyai ukuran tetap atau batas, Ukuran dari int sendiri yaitu 4 byte dengan jangkauan -2,147,483,64... hingga 2,147,483,64... tetapi pada PC nilai yang diperbolehkan adalah niali -32768 sampai 32767, Selain C++ ada beberapa tipe data yang masing - masing bisa menampung bilangan bulat pada bahasa pemrograman C++ contohnya tipe data char tipe data ini bias digunakan untuk penulisan karakter dalam kode *ASCII*, tetapi tipe data ini juga bisa digunakan untuk menyimpan integer dari 0 hingga 255, kemudian tipe data *short int*, integer dengan jangkauan pendek ukuran 2 byte dengan jangkauan -32,76... hinnga 32,76... , tipe data *long int (long)* integer dengan jangkauan panjang ukuran 4 byte dengan jangkauan -2,147,483,64... hingga 2,147,483,64... ,maka untuk meningkatkan pembelajaran dan pemahaman akan operasi perhitungan bilangan bulat mahasiswa minimal mengetahui tipe data manakah yang dapat digunakan untuk proses perhitungan, proses perhitugan penjumlahan, pengurangan, pembagian dan perkalian . Pada bilangan integer mahasiswa bisa menggunakan proses perhitungan ini untuk menuliskan code dan memahami cara logika dari bilangan bulat. Pembelajaran pada proses perhitungan tak selalu dapat menghasilkan bilangan bulat positif atau selalu berbasis positif mahasiswa bisa saja membuat program sederhana dengan basis

bilangan bulat nya adalah sebuah bilangan bulat negatif atau dalam penulisan code mahasiswa membuat perulangan dimana hasil dari perulangan terus menerus mendapatkan hasil bilangan ganjil ataupun bilangan genap dalam proses ini memahami sebuah logika dari sebuah perulangan juga dapat meningkatkan pemahaman akan operasi hitung pada bilangan bulat .

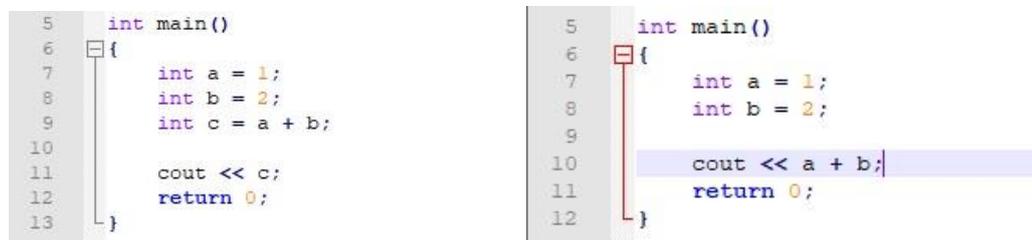
### ***Proses Penerapan bilangan bulat pada Code Sederhana***

Proses pada operasi hitung ini diterapkan dalam baris code yaitu dengan membuat variabel dengan tipe data yang sama kemudian membuat proses perhitungan bisa (+), (-), (\*) dan (/) atau (%). Proses perhitungan bilangan bulat pada code akan membuat mahasiswa bisa berfikir lebih banyak lagi tentang penulisan code dalam bahasa pemrograman c++ karena dalam menuliskan source code tidak hanya dapat menuliskan dalam satu logika saja mahasiswa bisa menganalisis penulisan code dan dengan metode yang seperti ini kemampuan mahasiswa akan lebih meningkat dalam operasi hitung bilangan bulat atau operasi hitung menggunakan tipe data integer (int).

Dalam Penelitian ini untuk meningkatkan kemampuan operasi perhitungan bilangan bulat dengan membuat code sederhana menggunakan tipe data integer pada software untuk menulis code seperti sublime text 3, notepad++, notepad, dev dan lain sebagainya untuk mengetahui tata cara penulisan dan mengetahui compiler ketika code ingin dijalankan dalam proses ini mahasiswa akan lebih banyak istilah – istilah pada Bahasa pemrograman , dan dengan ini juga mahasiswa akan lebih mudah menerima pembelajaran pada operasi hitung bilangan bulat , berikut cth operasi perhitungan pada code sederhana pada Bahasa C++

### ***Penjumlahan***

Gambar 1. Proses Penjumlahan bilangan bulat pada code sederhana



```
5 int main()
6 {
7     int a = 1;
8     int b = 2;
9     int c = a + b;
10
11     cout << c;
12     return 0;
13 }
```

```
5 int main()
6 {
7     int a = 1;
8     int b = 2;
9
10     cout << a + b;
11     return 0;
12 }
```

Hasil code setelah di compile = 3 , integer a = 1 dan integer b = 2 ,kemudian integer c = a + b Cout sebagai code untuk menampilkan integer c yang dimana c berisi code penjumlahan antara integer a dengan integer b.

Gambar 3 - 4. Proses Penjumlahan bilangan bulat pada code sederhana

```

4
5     int a = 1;
6     int b = 2;
7
8     int main()
9     {
10
11         cout << a + b;
12         return 0;
13     }

```

```

4
5     int a = 2;
6     int b = -5;
7
8     int main()
9     {
10
11         cout << a + b;
12         return 0;
13     }

```

```

D:\>a
3
D:\>

```

```

D:\>a
-3
D:\>

```

```

D:\>a
7
D:\>

```

Hasil output dari gambar keempat setelah di compile adalah  $-3$ , karena integer  $a = 2$ , dan integer  $b = -5$  cout pada code adalah  $a + b$ , jika pada cout  $a - b$  maka hasil dari penjumlahan gambar keempat adalah  $7$ .

*Int main ()* pernyataan ini mendeklarasikan fungsi utama, bahwa suatu program C++ dapat berisi banyak fungsi, tetapi harus selalu memiliki sebuah fungsi utama (main function). Fungsi adalah modul yang berisi kode-kode untuk menyelesaikan masalah-masalah tertentu. Kata *int* menandakan fungsi integer main. Kurung kurawal buka menandakan awal program. Dan kurung kurawal tutup menandakan akhir program. *Cout* adalah sebuah objek dari library standar C++ yang digunakan untuk mencetak string ke piranti output standar, yang biasanya adalah layar komputer kita. Compiler menghubungkan kode dari library standar itu dengan kode yang telah kita tuliskan untuk mendapatkan hasil yang executable. Return 0 digunakan untuk mengembalikan nilai dari 0 tersebut atau suatu nilai integer. Intruksi return membuat fungsi main () akan mengembalikan kode yang mengikuti instruksi tersebut. Ini merupakan cara yang sudah sering digunakan untuk mengekskusi program.

### Pengurangan

Gambar 5 – 6. Proses Pengurangan bilangan bulat pada code sederhana

```
4  
5     int b = -5;  
6     int c = 10;  
7  
8     int main()  
9     {  
10  
11         cout << b - c;  
12         return 0;  
13     }
```

```
4  
5     int b = 5;  
6     int c = -10;  
7  
8     int main()  
9     {  
10  
11         cout << b - c;  
12         return 0;  
13     }
```

### Perkalian

Gambar 7 – 8. Proses Perkalian bilangan bulat pada code sederhana

```
4  
5     int b = 2;  
6     int c = -5;  
7  
8     int main()  
9     {  
10  
11         cout << b * c;  
12         return 0;  
13     }
```

```
4  
5     int b = -2;  
6     int c = -5;  
7  
8     int main()  
9     {  
10  
11         cout << b * c;  
12         return 0;  
13     }
```

### Pembagian

Gambar 9 – 10. Proses pembagian bilangan bulat pada code sederhana

```
4  
5     int b = 8012;  
6     int c = 806;  
7  
8     int main()  
9     {  
10  
11         cout << b / c;  
12         return 0;  
13     }
```

```
4  
5     int b = 8012;  
6     int c = -806;  
7  
8     int main()  
9     {  
10  
11         cout << b / c;  
12         return 0;  
13     }
```

Gambar 11– 12. Proses pembagian bilangan bulat pada code sederhana

```

4
5     int b = -8012;
6     int c = -806;
7
8     int main()
9     {
10
11         cout << b / c;
12         return 0;
13     }

```

```

4
5     int b = -8012;
6     int c = -806;
7     int d = -6;
8     int a = b / c;
9     int e = a / d;
10
11     int main()
12     {
13
14         cout << a << "\n";
15         cout << e;
16         return 0;
17     }

```

Pada beberapa code sederhana di atas telah menggunakan library yang sesuai untuk digunakan dalam proses perhitungan bilangan bulat, pada header dari code sederhana ini library yang digunakan adalah library <iostream> ini adalah objek dari kelas ostream yang dapat mewakili aliran dari output standar yang berorientasi untuk mempersempit karakter (char jenis). Dan ostream sendiri berkaitan dengan C stream stdout. Aliran output standar adalah tujuan default karakter yang ditentukan oleh lingkungan. Tujuan ini dapat dipergunakan Bersama dengan benda-benda yang lebih standar (seperti cerr atau yang menyumbat). Sebagai objek kelas ostream, karakter dapat ditulis untuk itu data baik sebagai diformat menggunakan operator penyisipan (operator <<) atau sebagai data terformat, menggunakan fungsi anggota seperti menulis. Objek dinyatakan di header <iostream> dengan hubungan eksternal dan durasi statis: berlangsung seluruh durasi program. Dalam hal perintah inisialisasi statis, cout dijamin dengan baik dibangun dan diinisialisasi selambat – lambatnya pertama kalinya objek jenis (ios base:: Init) dibangun, dengan dimasukkannya <iostream> menghitung sebagai setidaknya satu inisialisasi seperti objek dengan durasi statis. Cout tidak terikat dengan output lain, Secara default cout disinkronkan dengan stdout. Sebuah program sebaiknya tidak dicampuri operasi output pada cout dengan output operasi pada wcout (atau dengan operasi lain output lebar yang berorientasi pada stdout) setelah operasi output telah dilakukan antara keduanya, aliran output standar mengakuisisi orientasi (baik sempit atau lebar) yang hanya dapat dengan aman diubah oleh freopen memanggil stdout.

Integer dapat langsung di inputkan atau di berikan nilai negatif karena bilangan bulat pada C++ (integer) mempunyai batasan nilai yang cukup Panjang tetapi integer adalah bilangan yang satu bilangan integer tidak dapat menghasilkan angka dengan titik/bilangan cacah kendati

integer mempunyai Batasan yang cukup Panjang bilangan bulat pada C++ (integer) berbeda dengan tipe data float ataupun double yang hasil dari operasi perhitungan bisa berupa bilangan cacah dengan titik mengambang, bilangan cacah dengan ketelitian ganda dan bilangan cacah dengan ketelitian ganda dan panjang (float and double,long double), walau pada semua tipe data ini mempresentasikan angka pada Bahasa Pemrograman C++.

*Latar Belakang* membuat sebuah penelitian tentang meningkatkan operasi bilangan bulat pada Bahasa pemrograman tak lebih untuk meningkatkan pemahaman terhadap bilangan bulat itu sendiri. Banyak dari mahasiswa mungkin belum terlalu mengenal Bahasa pemrograman C++ terlebih lagi ketika mahasiswa yang sama sekali tidak mempunyai dasar programming atau membuat sebuah program akan lebih sulit ketika mahasiswa diharuskan membuat sebuah program sederhana dengan menggunakan Bahasa C++, maka dari itu penelitian ini dimaksudkan untuk meningkatkan pemahaman mahasiswa tentang macam – macam Bahasa Pemrograman dan salah satu nya Bahasa pemrograman tingkat tinggi yaitu C++, sangatlah penting pemahaman akan Bahasa ini terlebih di masa yang sekarang ini banyak perusahaan – perusahaan besar yang mengembangkan game dengan Bahasa pemrograman C++, salah satunya yaitu Dota 2 (Defense of the Ancients 2) game dikembangkan oleh *Valve Corporation* dan perancang nya yaitu *IceFrog*.

## **METODE PENELITIAN**

Dalam penelitian ini ada beberapa tahapan yang dapat meningkatkan pemahaman mahasiswa baik dalam pengetahuan penulisan program aritmatika sederhana pada C++ dan pada operasi hitung bilangan bulat. Tahapan pada penelitian ini berupa Dasar Pemrograman untuk membuat sebuah code program pada Bahasa Pemrograman C++ menggunakan bilangan bulat (integer). Penelitian ini menggunakan Tahapan Pseudocode yaitu menuliskan algoritma sederhana, serta membuat program sederhana. Berikut tahapan Pseudocode membuat operasi hitung

Gambar 13. Algoritma membuat sebuah operasi hitung

```

1  Algoritma Pejumlahan bilangan bulat
2
3  //pada tahap ini menentukan tipe data dan nama variabel
4  tentukan variabel pertama = input angka pertama
5  tentukan variabel kedua = input angka kedua
6
7  Jalankan
8
9  //membuat variabel penampung
10 tentukan variabel penampung variabel 1 dan variabel 2
11 //menampilkan variabel penampung
12 tampilkan variabel penampung
13
14 selesai
15
16

```

Algoritma adalah urutan langkah – langkah logis untuk menyelesaikan satu permasalahan. Ketika kita memutuskan untuk membuat sesuatu maka kita memerlukan langkah – langkah yang logis untuk bisa membuatnya atau misalnya kita mempunyai masalah maka kita butuh langkah yang tidak sembarangan tetapi butuh pertimbangan yang logis untuk mengatasinya. Algoritma ini adalah yang harus dipahami oleh seorang programmer dan kita bisa mendapatkan matakuliah seperti ini pada jurusan yang berkaitan dengan pemrograman seperti Struktur Data dan Pemrograman. Ada 3 bentuk dasar dari Algoritma yaitu: *Algoritma Sekuensial (Sequence Algorithm)* algoritma yang langkah nya berurutan dari awal sampai akhir, *Algoritma Perulangan (Looping Algorithm)* algoritma ini akan menjalankan langkah – langkah tertentu berulang – ulang, *Algoritma percabangan / Bersyarat (Conditional Algorithm)* algoritma ini menjalankan langkah setelahnya apabila ada syarat sudah terpenuhi. Algoritma ini sangat penting dalam dunia komputer dan pemrograman dan seperti halnya matematika yang membutuhkan rumus untuk menyelesaikan suatu permasalahan algoritma juga membutuhkan langkah – langkah logis yang untuk menyelesaikan suatu masalah. Algoritma haruslah ada input, proses, output memiliki instruksi – instruksi yang spesifik jelas dan tidak ambigu dan juga mempunyai stopping role.

Gambar 14. Pseudocode

```
1  PSEUDECODE
2
3  start
4
5  Deklarasi
6  input nilai1 : integer
7  input nilai2 : integer
8
9  do
10
11  read n <- nilai1 + nilai2
12
13  printf n
14
15  end
```

Pseudocode tidak terikat oleh Bahasa pemrograman tertentu. Pseudocode digunakan untuk menuliskan algoritma dari suatu masalah serta berisikan langkah-langkah untuk menyelesaikan suatu masalah tetapi pseudocode berbeda dari algoritma. Pseudocode dapat memudahkan penulisan code program serta di dalam pseudocode kita dapat bebas menuliskan Bahasa yang mudah dimengerti oleh orang lain seperti: if(jika), then(maka), while (ketika), do(jalankan), repeat(ulangi), for (untuk menampilkan tulisan sesuai angka yg telah ditentukan), dan lain sebagainya. Pseudocode adalah suatu Bahasa yang memungkinkan programmer untuk berfikir bagaimana cara untuk menyelesaikan suatu permasalahan tanpa harus memikirkan syntax dari Bahasa pemrograman. Dengan mempelajari tahapan ini kita akan lebih tahu cara membuat sebuah pseudocode dengan baik di tambah dengan pemahaman akan algoritma, di tahapan ini juga kita menganalisis penggunaan operasi hitung dalam suatu code, bilangan bulat (integer) adalah salah satu tipe data yang dapat di gunakan untuk melakukan proses perhitungan pada Bahasa pemrograman C++, tahapan mempelajari pseudocode pada mahasiswa ini dapat meningkatkan pemahaman untuk menuliskan sebuah langkah-langkah sederhana untuk mempermudah membuat suatu program dari pemilihan library, penggunaan variabel dan tipe data, dan bagaimana cara menampilkan output dari suatu program.

Dari tahapan-tahapan seperti inilah yang bisa menguatkan ingatan mahasiswa pada operasi bilangan pada suatu program sederhana, mahasiswa akan lebih baik dalam memilih sebuah bilangan apa yang akan di operasi kan, karena mahasiswa sudah sering di berikan pemahaman seperti ini pada matakuliah yang diambilnya dan pada prakteknya pun mahasiswa akan lebih memahami konsep dari suatu bilangan dan kedepannya pun hal seperti inilah juga

sangat diinginkan oleh seorang dosen, mahasiswa mendapatkan materi yang jelas dan mahasiswa yang cukup untuk menanyakan berbagai macam program sederhana untuk dikembangkan ataupun mahasiswa belajar demi meningkatkan pengetahuan Bahasa C++.

Tahapan setelah pseudocode adalah membuat program operasi hitung sederhana dengan Bahasa C++, misalnya membuat suatu program kalkulator, membuat program untuk menampilkan angka ganjil dan genap, mencari angka terbesar Max dan angka terkecil Min dan program sederhana lainnya yg berkaitan dengan bilangan bulat dan operasi hitung.

Gambar 13. Membuat fungsi Operasi Perhitungan

```

1  #include <iostream>
2
3  using namespace std;
4
5  int Tambah(int a,int b)
6  {
7      int fungsi;
8      fungsi = a+b;
9      return fungsi;
10 }
11 int Kurang(int a,int b)
12 {
13     int fungsi;
14     fungsi = a-b;
15     return fungsi;
16 }
17 int Kali(int a,int b)
18 {
19     int fungsi;
20     fungsi = a*b;
21     return fungsi;
22 }
23 int Bagi(int a,int b)
24 {
25     int fungsi;
26     fungsi = a/b;
27     return fungsi;
28 }

```

Membuat fungsi ini berfungsi untuk memecahkan program ketika sebuah program yang cukup rumit dan besar menjadi program-program lebih sederhana. Jika dilihat memang tidak ada perubahan yang mecolok/signifikan diantara prosedur dan fungsinya karena, hanya di bedakan pada *return value*; tetapi dengan menggunakan fungsi bisa kita ketahui bahwa Pada fungsi sendiri,

1. nilai akan dipanggil kembali terdapat nama fungsinya (prosedur ada pada parameter yang dipanggil secara acuan),
2. fungsi tersebut bisa langsung digunakan untuk dicetak hasilnya atau bisa kita pindahkan ke variable,
3. selain fungsi yang dapat di gunakan dari sebuah prosedur yaitu parameternya mengandung nilai balik.

Dengan demikian dari program diatas bisa kita ketahui bahwa jika sub program memiliki nilai balik (*return value*) maka sub program ini disebut *Function(fungsi)*.

Gambar 14. Memanggil masing – masing dari fungsi

```
29
30  int main()
31  {
32      int c,d;
33
34      cout << " masukkan nilai 1 = ";
35      cin >> c;
36      cout << " masukkan lebar 2 = ";
37      cin >> d;
38      cout << "\n";
39      cout << " Operasi hitung pada bilangan bulat" << "\n";
40      cout << "\n";
41      cout << " Penambahan   : " << Tambah(c,d) << "\n";
42      cout << " Pengurangan  : " << Kurang(c,d) << "\n";
43      cout << " Perkalian    : " << Kali(c,d) << "\n";
44      cout << " Pembagian    : " << Bagi(c,d) << "\n";
45      return 0;
46  }
```

Gambar 15. Hasil dari output program

```
D:\>g++ a.cpp -o a
D:\>a
masukkan nilai 1 = 80
masukkan lebar 2 = 10

Operasi hitung pada bilangan bulat

Penambahan   : 90
Pengurangan  : 70
Perkalian    : 800
Pembagian    : 8
D:\>_
```

## HASIL PENELITIAN

Dari penelitian ini berdasarkan tanggapan dan wawancara dari beberapa mahasiswa, diperoleh hasil sebagai berikut. (1) Mahasiswa biasanya telah mendapatkan seperti ini dan pseudocode tetapi sebagian juga masih belum karena faktor mengajar yang berbeda-beda dari masing -masing dosen tetapi masih mungkin bila materi ini bisa dijadikan acuan untuk pembahasan program sederhana yang berkaitan dengan operasi perhitugan dan bilangan bulat, (2) proses pembelajaran materi operasi hitung bilangan bulat ini masih belum mendapatkan

hasil yang maksimal, dikarenakan mahasiswa yang belum paham betul akan bahasa pemrograman dan terlebih lagi mahasiswa yang cenderung lebih mudah paham sulit untuk memberikan sedikit pengetahuan kepada rekan – rekan menjadikan materi seperti ini akan sedikit lebih lama dipahami, dan (3) masih banyak mahasiswa yang pada jurusan yang berkaitan dengan komputer dan Bahasa pemrograman C++ sudah cukup paham dengan materi seperti ini karena prakteknya sebagian mengatakan paham dan mengerti untuk membuat sebuah program sederhana tetapi pada kenyataan mahasiswa tidak tahu fungsi – fungsi dan perintah yang terdapat pada Bahasa pemrograman C++ dan ditambah lagi masih bingung untuk membuat sebuah program sederhana yang menyebabkan materi panjang seperti ini hanya menambahkan tingkat kesulitan mahasiswa untuk mempelajari dan meningkatkan pemahaman akan operasi hitung dan bilangan tetapi disisi lain mahasiswa lebih menerima pembahasan akan pseudocode yang menarik dan mudah untuk diterima.

Penelitian ini dalam hal pratikum bisa dikatakan kurang berhasil karena pada dasar nya dalam mempelajari operasi bilangan bulat Bahasa pemrograman akan lebih baik ketika langsung membuat membuat program dari materi yg telah disampaikan oleh masing-masing, selain itu mahasiswa yang aktif bertanya tentang program juga akan lebih mudah paham ketika dilakukan nya pratek untuk membuat program, tetapi dari hasil wawancara tidak menutup kemungkinan bahwa mahasiswa yg belum terlalu fasih akan Bahasa pemrograman tidak membutuhkan materi tersebut, jika pratek saja maka mahasiswa tidak sepenuhnya akan paham bisa dikatakan mahasiswa yang sudah pernah memakai dan menggunakan Bahasa pemrograman akan lebih mudah memahami tetapi tidak dengan yang belum pernah memakai atau menggunakan jadi bisa disimpulkan mahasiswa jika hanya dengan pratek saja belum tentu akan memahami program dengan baik maka dari itu harus lah ada pemahaman terhadap Bahasa pemrograman itu sendiri yaitu, membaut materi tentang macam-macam tipe data dan ukuran dari masing-masing tipe data, kemudian pemahaman operasi perhitungan dan logika akan sangat membantu dalam membuat program, pemahaman algoritma dan pseudocode dan terakhir barulah membuat sebuah program, ketika mahasiswa sudah fasih akan tahapan-tahapan ini maka membuat program yg beragam dan berkaitan dengan operasi perhitungan

## **KESIMPULAN**

Dari metode dan penelitan yang sudah dejelaskan bisa kita lihat bahwa operasi perhitungan pada bilangan bulat pada Bahasa pemrograman C++ adalah dasar yang harus diketahui oleh mahasiswa karena dengan dasar dasar seperti ini mahasiswa akan lebih mudah

memahami pembelajaran akan operasi perhitungan untuk menentukan bilangan bulat, maka mahasiswa yang telah memahami akan program sederhana seperti tidak menutup kemungkinan mahasiswa bisa mencoba atau membuat program perhitungan lainnya yang membutuhkan perhitungan yang rumit dan cukup panjang maka dari itu materi dasar pemrograman C++ bisa membantu dan membuat akan mahasiswa lebih fasih lagi dalam membangun suatu program, telah mendapatkan sebuah pembelajaran algoritma dan pseudocode ini tidak akan membingungkan bagi mahasiswa yang belum pernah mencoba membuat sebuah program sederhana ataupun program yang rumit sekalipun karena telah dibekali pengetahuan tersebut.

Dan harapan dari penelitian ini mahasiswa lebih bijak lagi dalam mengolah suatu materi agar tidak terjadi salah paham ataupun kurangnya pemahaman dalam, dan terlebih lagi materi seperti ini jangan hanya dijadikan acuan untuk membuat satu program saja, jadikan materi seperti ini untuk mendapatkan ide/gagasan lain membuat program perhitungan sederhana yang dapat membantu mahasiswa lainnya untuk mengatasi ketidaktahuan akan Bahasa pemrograman c++.

## **DAFTAR PUSTAKA**

- Muginah & Widjajanti, Djamilah Bondan (2014), Peningkatan Keterampilan Hitung Bilang Bulat Melalui Metode Electrical Charges Pada Siswa Kelas V SD, 2(2). Yogyakarta: Jurnal Prima Edukasi
- Sinaga, Hendro Pramana (2012). Program Modulo Dengan C++ (Online), diakses pada tanggal 31 Oktober 2018
- Karim, Damas Amirul. Contoh Program C++ Sederhana Menggunakan Fungsi & Prosedur (Online), (<https://www.kodedasar.com/fungsi-cpp/>). Diakses pada tanggal 1 November 2018
- Wikipedia. C++ (Online), (<https://id.wikipedia.org/wiki/C%2B%2B>). Diakses pada tanggal 1 November 2018
- Karim, Damas Amirul, Belajar Bahasa Pemrograman C++ Lengkap dari Awal untuk Pemula (Online). (<https://www.kodedasar.com/belajar-cpp/>). Diakses 1 November 2018