

LAPORAN TUGAS PROJEK

PROGAM MENGHITUNG FAKTORIAL

*Diajukan untuk memenuhi salah satu tugas Mata Kuliah Algoritma yang
diampu oleh:*

Agus Nugraha ST,.M.Kom



Disusun Oleh:

Diana Friska Aulia

2C1230004

Kayla Ayu

2C1230008

**PROGAM STUDI AGRIBISNIS FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS KOPERASI INDONESIA**

2023

KATA PENGANTAR

Puji syukur kehadiran Allah SWT, karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan makalah mata kuliah Algoritma ini yang berjudul *Program Menghitung Faktorial*

Penulis telah berusaha semaksimal mungkin untuk dapat menyelesaikan makalah ini dengan sebaik mungkin dan sebenar-benarnya. Penulis menyadari makalah ini jauh dari kesempurnaan baik materi, penganalisaan, dan pembahasan. Semua hal ini dikarenakan keterbatasan kemampuan dan pengalaman.

Penulis berharap makalah ini dapat diterima oleh Bapak/Ibu dosen mata kuliah Algoritma dan juga dapat dipahami oleh para pembaca. Penulis mengharapkan saran dan kritik dari semua pihak yang bersifat membangun, guna terciptanya kesempurnaan makalah ini. Akhir kata penulis ucapkan terima kasih.

Bandung, 25 Januari, 2024

Penulis

ABSTRAK

Laporan ini membahas pengembangan program komputer sederhana yang dirancang untuk menghitung faktorial suatu bilangan. Faktorial adalah operasi matematika yang menggambarkan hasil perkalian semua bilangan bulat positif dari 1 hingga bilangan tersebut. Tujuan utama dari program ini adalah memberikan solusi efisien dan cepat dalam perhitungan faktorial tanpa melibatkan proses manual yang rumit.

Dalam laporan ini, kami menguraikan langkah-langkah pengembangan program, algoritma yang digunakan, serta contoh penerapan program dalam berbagai konteks. Algoritma yang diusulkan telah dirancang untuk memberikan hasil perhitungan faktorial dengan cepat, meminimalkan beban komputasi, dan meningkatkan efisiensi. Laporan ini juga membahas beberapa tantangan yang dihadapi selama pengembangan program dan strategi pemecahan masalah yang diterapkan. Selain itu, kami memberikan ilustrasi penggunaan program dalam berbagai situasi praktis untuk memperjelas aplikasi dari program ini.

Diharapkan laporan ini dapat memberikan pemahaman yang jelas tentang konsep pengembangan program menghitung faktorial dan dapat menjadi referensi yang bermanfaat bagi pembaca yang tertarik dalam pemrograman dan matematika.

ABSTRACT

This report discusses the development of a simple computer program designed to calculate the factorial of a given number. Factorial is a mathematical operation that describes the result of multiplying all positive integers from 1 to the specified number. The main objective of this program is to provide an efficient and fast solution for factorial calculations without involving complex manual processes.

In this report, we outline the steps of program development, the algorithm utilized, and examples of the program's application in various contexts. The proposed algorithm is designed to yield factorial calculations swiftly, minimizing computational load, and enhancing efficiency. The report also addresses several challenges encountered during program development and the applied problem-solving strategies. Additionally, we provide illustrations of the program's usage in various practical situations to elucidate its applications.

It is anticipated that this report will offer a clear understanding of the concept of developing a factorial calculation program and serve as a valuable reference for readers interested in programming and mathematics.

DAFTAR GAMBAR

Gambar 3.1 Pseudocode menghitung factorial	10
Gambar 3.2 Algoritma dan Flowchart	10
Gambar 4.1 Hasil Running Bilangan Faktorial.....	11
Gambar 4.1 Hasil Running menghitung faktorial.....	11

DAFTAR ISI

KATA PENGANTAR.....	1
ABSTRAK	2
DAFTAR GAMBAR.....	3
DAFTAR ISI.....	4
BAB I PENDAHULUAN.....	5
1.1 Latar Belakang.....	5
1.2 Batasan Masalah	5
BAB II	7
DASAR TEORI.....	7
2.1 Pengertian	7
2.2 Definisi.....	7
2.3 Sejarah	8
BAB III	10
3.1 Pseudocode.....	10
3.2 Algoritma dan Flowchart	10
BAB IV	11
IMPLEMENTASI.....	11
PENUTUP.....	12
DAFTAR PUSTAKA.....	13

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia matematika dan ilmu komputer, penghitungan faktorial memiliki peran yang signifikan sebagai dasar untuk berbagai perhitungan dan analisis. Faktorial suatu bilangan merupakan hasil perkalian dari semua bilangan bulat positif hingga bilangan tersebut. Konsep ini bukan hanya sekadar rumus matematika, melainkan juga menjadi elemen kunci dalam berbagai disiplin ilmu.

Pentingnya penghitungan faktorial terutama termanifestasi dalam berbagai konteks, seperti pemodelan matematika, analisis statistik, dan pengembangan algoritma. Dalam situasi di mana perlu mengukur kemungkinan permutasi atau kombinasi suatu himpunan elemen, penggunaan faktorial menjadi tak terelakkan. Oleh karena itu, pengembangan program komputer yang mampu menghitung faktorial dengan efisien memiliki dampak positif dalam mempermudah proses perhitungan tersebut.

Latar belakang ini muncul dari kebutuhan akan alat bantu komputasional yang dapat mengatasi perhitungan faktorial secara otomatis. Penggunaan program untuk menghitung faktorial tidak hanya menghemat waktu, tetapi juga meminimalkan potensi kesalahan manusia yang dapat terjadi dalam perhitungan manual.

Dalam laporan ini, kami akan menjelaskan pengembangan program komputer untuk menghitung faktorial, mencakup algoritma yang digunakan, langkah-langkah pengembangan, serta relevansi dan manfaatnya dalam berbagai konteks aplikasi matematika dan teknologi.

Melalui laporan ini, diharapkan pembaca dapat memahami pentingnya penghitungan faktorial dalam dunia komputasi dan bagaimana program ini dapat menjadi solusi efisien dalam menangani perhitungan faktorial.

1.2 Batasan masalah

1. Pemrosesan data yang sangat besar (ulang bilangan yang besar) akan mengakibatkan kehabisan ruang penyimpanan atau mengalami tahanan berharga yang terlalu besar.
2. Banyak kasus mungkin memerlukan algoritma yang lebih efisien untuk menghitung faktorial. Contohnya, algoritma Prime Swing yang memerlukan waktu $O(n \log n)$ untuk menghitung faktorial primer yang besar.
3. Menghitung faktorial pada nilai n yang negatif atau bukan bilangan bulat yang didefinisikan pada teori nomor dan matematika mungkin menyebabkan tahanan berharga yang terlalu besar atau hasil yang tidak diharapkan.
4. Keandalan dari program yang berbasis rekursi akan tergantung pada ukuran maksimum ruang stack yang diizinkan oleh sistem. Dalam kasus dimana ukuran ini terlalu kecil, program mungkin mengalami stack overflow yang menyebabkan program crash atau perilaku yang tidak diharapkan.

5. Waktu yang diperlukan untuk menghitung faktorial dapat menjadi masalah, terutama untuk pemrosesan data yang sangat besar. Jika efisiensi waktu yang tinggi diperlukan, Anda mungkin perlu mempertimbangkan menggunakan algoritma yang lebih efisien atau bahkan teknik pemrograman paralel.

BAB II

DASAR TEORI

2.1 Pengertian

Fungsi faktorial didefinisikan sebagai:

$$n! = \prod_{k=1}^n k \quad \text{untuk semua } n \geq 1.$$

Selain definisi tersebut, terdapat juga definisi secara rekursif, yang didefinisikan untuk

$$n! = \begin{cases} n \cdot (n-1)!, & \text{untuk } n \geq 1 \\ 1, & \text{untuk } n = 0. \end{cases}$$

Untuk n yang sangat besar akan teralau melelahkan untuk menghitung $n!$ menggunakan definisi tersebut, jika presisi penting pendekatan dari $n!$ bisa di hitung menggunakan rumus Stirling:

$$n! \approx \sqrt{2\pi n} \frac{n^n}{e^n}.$$

Juga terdapat definisi analitik untuk faktorial, yaitu menggunakan fungsi gamma:

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$$
$$n! = \Gamma(n+1)$$

2.2 Definisi

Fungsi faktorial ditentukan oleh produk, yaitu:

$$n! = 1 \cdot 2 \cdot 3 \cdots (n-2) \cdot (n-1) \cdot n,$$

diatas merupakan bilangan bulat dari $n \geq 1$. Ini dapat ditulis dalam notasi perkalian pi sebagai:

$$n! = \prod_{i=1}^n i.$$

Hal tersebut mengarah menuju relasi pengulangan:

$$n! = n \cdot (n-1)!.$$

Sebagai contoh,

$$5! = 5 \cdot 4!$$

$$6! = 6 \cdot 5!$$

$$50! = 50 \cdot 49!$$

dan seterusnya.

➤ Faktorial nolsunting

Faktorial dari 0 adalah 1, atau dalam simbol, $0! = 1$.

Ada beberapa motivasi untuk definisi ini:

- Untuk nilai $n = 0$, definisi dari $n!$ sebagai perkalian melibatkan hasil kali tanpa bilangan sama sekali, dan begitu juga contoh dari konvensi yang lebih luas bahwa produk dari tidak ada faktor yang sama dengan identitas perkalian (lihat Produk kosong).
- Hanya ada satu permutasi dari nol objek (tanpa ada yang diubah, satu-satunya penataan ulang adalah tidak melakukan apa-apa).
- membuat banyak identitas di kombinatorik berlaku untuk semua ukuran yang berlaku. Banyaknya cara untuk memilih 0 elemen dari himpunan kosong diberikan oleh koefisien binomial

$$\binom{0}{0} = \frac{0!}{0!0!} = 1.$$

Secara lebih umum, jumlah cara untuk memilih semua elemen n di antara himpunan n adalah

- Hal ini memungkinkan untuk ekspresi ringkas dari banyak rumus, seperti fungsi eksponensial, sebagai deret pangkat:

$$\binom{n}{n} = \frac{n!}{n!0!} = 1.$$

- Hal ini dapat memperluas hubungan pengulangan ke 0.

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

2.3 Sejarah

Faktorial digunakan untuk menghitung permutasi setidaknya sejak abad ke-12, oleh para sarjana Matematika India. Pada tahun 1677, Fabian Stedma. mendeskripsikan faktorial yang diterapkan pada mengubah dering, seni musik yang melibatkan dering dari banyak lonceng yang disetel. Setelah menggambarkan pendekatan rekursif, Stedman memberikan pernyataan faktorial (menggunakan bahasa aslinya):

Sekarang sifat dari metode ini adalah sedemikian rupa, sehingga perubahan pada satu angka mencakup [termasuk] perubahan pada semua angka yang lebih kecil. sedemikian rupa sehingga Peal yang lengkap dari perubahan pada satu nomor tampaknya dibentuk dengan menyatukan Peal yang lengkap pada semua nomor yang lebih kecil menjadi

satu keseluruhan tubuh. Notasi dari $n!$ di perkenalkan oleh matematikawan asal Prancis Bernama Christian Kramp pada tahun 1808.

- Anggota terpilih dari faktorial urutan (barisan [A000142](#) pada [OEIS](#)); nilai yang ditentukan dalam notasi ilmiah dibulatkan ke presisi yang ditampilkan

N	$n!$
0	1
1	1
2	2
3	6
4	24
5	120
6	720
7	5040
8	40.320
9	362.880
10	3.628.800
11	39.916.800
12	479.001.600
13	6.227.020.800
14	87.178.291.200
15	1.307.674.368.000
16	20.922.789.888.000
17	355.687.428.096.000
18	6.402.373.705.728.000
19	121.645.100.408.832.000
20	2.432.902.008.176.640.000
25	$1,551121004 \times 10^{25}$
50	$3,041409320 \times 10^{64}$
70	$1,197857167 \times 10^{100}$
100	$9,332621544 \times 10^{157}$
450	$1.733368733 \times 10^{1000}$
1000	$4.023872601 \times 10^{2567}$
3249	$6,412337688 \times 10^{10.000}$
10000	$2,846259681 \times 10^{35.659}$
25206	$1,205703438 \times 10^{100.000}$
100000	$2,824229408 \times 10^{456.573}$
205023	$2,503898932 \times 10^{1.000.004}$
1000000	$8,263931688 \times 10^{5.565.708}$
10^{100}	$1010^{101.9981097754820}$

BAB III

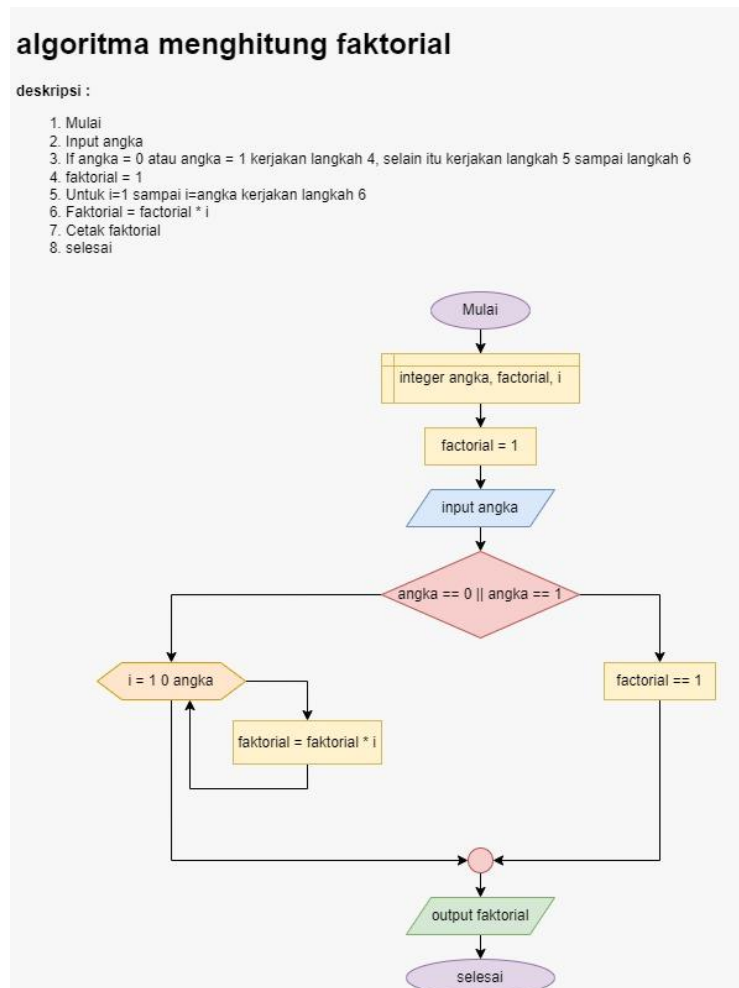
ALGORITMA DAN FLOWCHART

3.1 Pseudocode

```
{Algoritma Menghitung_Faktorial}  
  
DEKLARASI  
Integer nilai n  
DESKRIPSI  
n <- 1  
n*faktorial (n-1)  
Write ("Berapa faktorial? ");  
read (Nilai n)  
Write ("Faktorial = ", faktorial(n));
```

Gambar 3.1 Pseudocode menghitung factorial

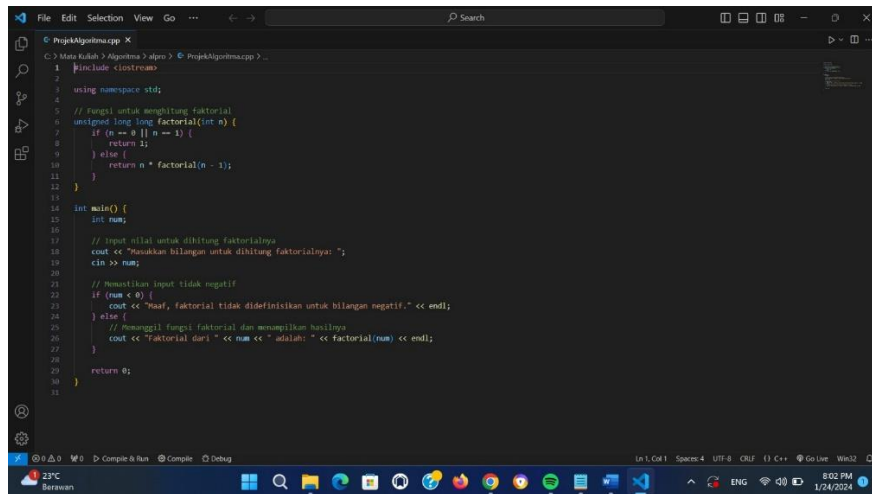
3.2 Flowchart dan Deskripsi



Gambar 3.2 Flowchart dan deskripsi 1

BAB IV

IMPLEMENTASI



```
1 #include <iostream>
2
3 using namespace std;
4
5 // fungsi untuk menghitung faktorial
6 unsigned long long factorial(int n) {
7     if (n == 0 || n == 1) {
8         return 1;
9     } else {
10         return n * factorial(n - 1);
11     }
12 }
13
14 int main() {
15     int num;
16
17     // input nilai untuk dihitung faktorialnya
18     cout << "Masukkan bilangan untuk dihitung faktorialnya: ";
19     cin >> num;
20
21     // Memastikan input tidak negatif
22     if (num < 0) {
23         cout << "Maaf, faktorial tidak didefinisikan untuk bilangan negatif." << endl;
24     } else {
25         // memanggil fungsi faktorial dan menampilkan hasilnya
26         cout << "Faktorial dari " << num << " adalah: " << factorial(num) << endl;
27     }
28
29     return 0;
30 }
```

Gambar 4.1 code program menghitung faktorial

```
Masukkan bilangan untuk dihitung faktorialnya: 5
Faktorial dari 5 adalah: 120
PS C:\Mata Kuliah\Algoritma\alpro\output>
```

Gambar 4.2 hasil running bilangan faktorial

BAB IV

PENUTUP

5.1 KESIMPULAN

Melalui pengembangan program menghitung faktorial, dapat disimpulkan bahwa program ini memberikan solusi yang efisien dan cepat dalam perhitungan faktorial suatu bilangan. Berdasarkan analisis algoritma yang digunakan, program mampu menghasilkan hasil perhitungan faktorial dengan waktu eksekusi yang minimal, meminimalkan beban komputasi, dan meningkatkan efisiensi penggunaan sumber daya komputer.

Penerapan program dalam berbagai konteks dan situasi praktis menunjukkan fleksibilitas dan kehandalan program. Dari perhitungan faktorial sederhana hingga aplikasi yang melibatkan bilangan besar, program dapat diandalkan untuk memberikan hasil yang akurat dan konsisten.

Tantangan yang dihadapi selama pengembangan program, seperti penanganan nilai yang besar atau optimalisasi algoritma, berhasil diatasi melalui strategi pemecahan masalah yang efektif. Hal ini menunjukkan bahwa program ini dapat diadaptasi dan ditingkatkan untuk mengatasi berbagai skenario perhitungan faktorial yang kompleks.

Secara keseluruhan, pengembangan program menghitung faktorial memberikan kontribusi yang positif dalam menyederhanakan dan mempercepat perhitungan matematis yang melibatkan faktorial. Dengan terus mengoptimalkan algoritma dan mempertimbangkan masukan dari pengguna, program ini memiliki potensi untuk terus berkembang dan menjadi solusi yang lebih baik di masa depan.

Diharapkan bahwa program ini tidak hanya memberikan manfaat praktis dalam perhitungan faktorial, tetapi juga memberikan dorongan dalam pengembangan dan penerapan program matematis yang efisien dan relevan.

DAFTAR PUSTAKA

Wikipedia. (2023, 17 7). Retrieved from <https://id.m.wikipedia.org/wiki/Faktorial#Sejarah>