# Penerapan Algoritma Brute Force dalam mencari Faktor Prima pada suatu Bilangan

Widhaprasa Ekamatra Waliprana - 13508080
*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
*If18080@students.if.itb.ac.id*

***Abstract*—Makalah ini akan membahas penerapan dari algoritma brute force. Penerapan yang dibahas adalah pemecahan masalah dalam pencarian faktor prima dari suatu bilangan. Faktor prima memiliki banyak kegunaan diantaranya adalah dalam menentukan FPB dan KPK. Proses pemecahan masalah ini merupakan pengembangan dari algoritma pencarian faktor dari sebuah bilangan dan uji keprimaan. Dalam penerapannya penulis menggunakan iterasi dalam mencari faktor prima tersebut. Ditampilkan juga hasil eksekusi program pencarian faktor prima ini.**

***Index Terms*—algoritma brute force, faktor, bilangan prima**

## I. PENDAHULUAN

Bilangan adalah suatu konsep matematika yang digunakan untuk pencacahan dan pengukuran. Simbol ataupun lambang yang digunakan untuk mewakili suatu bilangan disebut sebagai angka atau lambang bilangan. Dalam matematika, konsep bilangan selama bertahun-tahun lamanya telah diperluas untuk meliputi bilangan nol, bilangan positif, bilangan negatif, bilangan ganjil, bilangan genap, bilangan asli, bilangan cacah, bilangan pecahan, bilangan prima, bilangan rasional, bilangan irasional, bilangan kompleks dan bilangan lainnya.

| | |
|---|---|
| **Natural** | (0), 1, 2, 3, 4, 5, 6, 7, ..., *n* |
| **Integers** | −*n*, ..., −5, −4, −3, −2, −1, 0, 1, 2, 3, 4, 5, ..., *n* |
| **Positive integers** | 1, 2, 3, 4, 5, ..., *n* |
| **Rational** | $^{a}/_{b}$ where *a* and *b* are integers and *b* is not zero |
| **Real** | The limit of a convergent sequence of rational numbers |
| **Complex** | *a* + *bi* where *a* and *b* are real numbers and *i* is the square root of −1 |

**Gambar 1. Jenis-jenis Bilangan**

Setiap bilangan pasti mempunyai faktor. Faktor adalah bilangan yang habis membagi bilangan lainnya. Misalkan bilangan A habis dibagi B maka B merupakan faktor dari A. Faktor dari suatu bilangan merupakan sesuatu yang penting, karena jika kita mengetahui suatu faktor dari suatu bilangan maka proses pembagian akan lebih efisien.

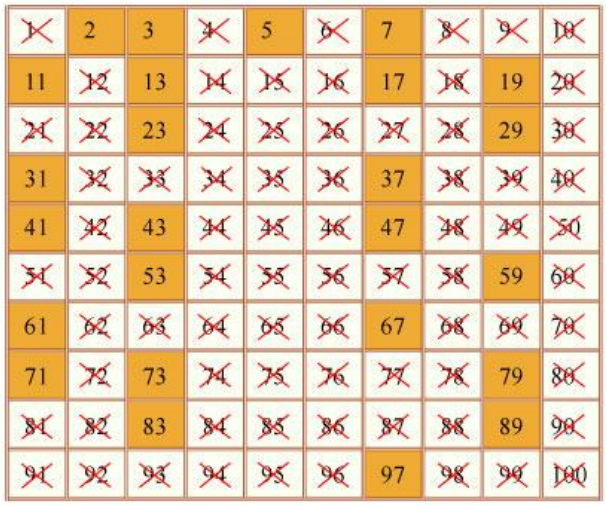
Pencarian faktor suatu bilangan merupakan salah satu solusi yang dapat memecahkan permasalahan. Permasalahan yang dapat dipecahkan dengan pencarian faktor suatu bilangan adalah dalam menentukan suatu faktor dari persamaan baik persamaan kuadrat, persamaan pangkat tiga dan persamaan polinomial lainnya.

Salah satu bilangan yang berkaitan dengan faktor bilangan adalah bilangan prima. Bilangan prima merupakan bilangan yang hanya habis dibagi 1 dan bilangan itu sendiri. Kegunaan bilangan prima sangatlah banyak, namun kita sangat kesulitan untuk mengecek apakah suatu bilangan merupakan bilangan prima, karena bilangan prima merupakan himpunan bilangan yang tidak memiliki pola. Oleh karena itu diperlukan suatu cara untuk mengecek apakah suatu bilangan

Dibawah ini akan dijelaskan lebih rinci mengenai bilangan prima:

### A. Pengertian Bilangan Prima

Bilangan prima adalah bilangan asli yang lebih besar dari 1 dengan faktor pembaginya adalah 1 dan bilangan itu sendiri. 2 dan 3 merupakan bilangan prima. 4 bukan bilangan prima karena 4 bisa dibagi 1,2, dan 4. Sepuluh bilangan prima yang pertama adalah 2, 3, 5, 7, 11, 13, 17, 19, 23 dan 29. Bilangan prima terkecil adalah 2 dan bilangan prima terbesar adalah tak hingga.



**Gambar 2. Bilangan Prima antara 1 sampai 100**

**B. Aplikasi Bilangan Prima**

Aplikasi dari bilangan prima diantaranya adalah:

1. Aritmatik modulo a prime p
2. Public-key Cryptography
3. Sylow Theorems
4. dan aplikasi lainnya.

**C. Faktor Prima**

Faktor Prima merupakan faktor dari sebuah bilangan yang faktornya adalah bilangan prima. Faktor prima merupakan faktor yang lebih sederhana dari faktor positif. Aplikasi dari faktor prima diantaranya adalah salah satu langkah dalam mencari FPB (Faktor Persekutuan Terbesar) dan KPK (Kelipatan Persekutuan Terkecil) dari banyak bilangan. Faktor Prima juga digunakan untuk mengenkripsi public key dari RSA Algorithm.

## II. DASAR TEORI

**Algoritma Brute Force**

Brute force adalah sebuah pendekatan yang lempang (straightforward) untuk memecahkan suatu masalah (problem statement) dan definisi konsep yang dilibatkan Algoritma brute force memecahkan masalah dengan sangat sederhana, langsung, dan dengan cara yang jelas (obvious way) meskipun bukan merupakan solusi yang paling mangkus.

Karakteristik Algoritma Brute Force

1. Algoritma Brute Force umumnya tidak "cerdas" dan tidak mangkus karena ia membutuhkan jumlah langkah yang besar dalam penyelesaiannya. Kadang pula algoritma Brute Force disebut juga algoritma naïf (*naïve algorithm*).
2. Algoritma Brute Force lebih cocok untuk masalah yang berukuran kecil.
3. Meskipun bukan metode yang mangkus, hampir semua masalah dapat diselesaikan dengan algoritma Brute Force.

Kekuatan dan Kelemahan Metode Brute Force

Kekuatan:

1. Metode brute force dapat digunakan untuk memecahkan hampir sebagian besar masalah (*wide applicability*).
2. Metode brute force sederhana dan mudah dimengerti.
3. Metode brute force menghasilkan algoritma yang layak untuk beberapa masalah penting seperti pencarian, pengurutan, pencocokan string, perkalian matriks.
4. Metode brute force menghasilkan algoritma baku (standard) untuk tugas-tugas komputasi seperti penjumlahan/perkalian n buah bilangan, menentukan elemen minimum atau maksimum di dalam tabel (list).

Kelemahan:

1. Metode brute force jarang menghasilkan algoritma yang mangkus.
2. Beberapa algoritma brute force lambat sehingga tidak dapat diterima.
3. Tidak sekontruktif/sekreatif teknik pemecahan masalah lainnya.

## III. PEMECAHAN MASALAH

Di bawah ini akan dipaparkan tahap-tahap dalam mencari faktor prima yang dipecahkan dengan Algoritma Brute force. Perlu diingat bahwa pencarian faktor prima ini adalah pengembangan dari algoritma pencarian faktor dari suatu bilangan dan uji keprimaan yang telah dipelajari pada bahasan algoritma brute force. Pertama-tama cari faktor positif dari bilangan yang ingin dicari faktor primanya. Kemudian setiap faktor positif dari bilangan tersebut dicek keprimaannya. Setelah di cek, maka faktor prima dari bilangan tersebut telah ditemukan. Di bawah ini akan dipaparkan secara teknis cara pencarian faktor positif, cek keprimaan dan pencarian faktor prima.

**A. Algoritma Brute Force dalam pencarian Faktor Positif suatu Bilangan**

Penerapan Algoritma Brute Force dalam pencarian Faktor Positif dapat digambarkan sebagai berikut:

1. Masukkan angka yang ingin dicari faktornya, kita tuliskan dengan nama input.

2. Inisialisasi suatu array of integer dengan nama faktor dan sebuah integer dengan nama faktorcount sebagai tempat penyimpanan himpunan faktor dari bilangan yang dicari.
3. Inisialisasi variabel i sebagai iterator dengan nilai i sama dengan 1.
4. Hitung input dimoduluskan dengan i (input mod i) dengan hasilnya kita namakan result.
5. Jika result sama dengan 0, maka masukkan i ke dalam array of integer faktor dengan faktorcount sebagai indeksnya dan nilai dari faktorcount ditambah 1.
6. Ulangi langkah 4 dan 5 dengan nilai i bertambah 1 sampai dengan nilai i sama dengan input.
7. Array of integer faktor sudah berisi faktor positif dari input dan faktorcount berisikan jumlah faktor positif yang dimiliki input.

Berikut adalah *source code* dari pemecahan masalah pencarian faktor positif dalam C++:

```
void factorgenerator(int input, int
factor[], int &factorcount)
{
  for(int i=1; i <= input; i++){
       if(input % i == 0){
             factor[factorcount] = i;
             factorcount++;
       }
  }
}
```

**B. Algoritma Brute Force dalam pengecekan Bilangan Prima**

Penerapan Algoritma Brute Force dalam pengecekan Bilangan Prima dapat digambarkan sebagai berikut:

1. Masukkan angka yang ingin dicek apakah bilangan itu bilangan prima atau bukan, kita tuliskan dengan nama input.
2. Jika input sama dengan 1 maka input bukan bilangan prima dan program selesai. Jika i lebih besar dari 1 maka lanjutkan.
3. Inisialisasi variabel i sebagai iterator dengan nilai i sama dengan 2.
4. input dimoduluskan dengan i (input mod i) dengan hasilnya kita namakan result.
5. Jika result sama dengan 0, maka input bukan bilangan prima dan program selesai, namun jika result tidak sama dengan 0 maka ulangi langkah 4 dan 5 dengan nilai i bertambah 1.
6. Jika nilai i sudah sama dengan input dan program belum selesai, maka input merupakan bilangan prima dan program selesai.
7. Program di atas menggunakan representasi Boolean dalam menentukan bilangan prima atau bukan.

Berikut adalah *source code* dari pemecahan masalah pencarian faktor positif dalam C++:

```
bool isprimenumber(int input){
  if(input == 1){
       return false;
  }
  for(int i=2; i < input; i++){
       if(input % i == 0){
             return false;
       }
  }
  return true;
}
```

**C. Algoritma Brute Force dalam pencarian Faktor Prima suatu Bilangan**

Penerapan Algoritma Brute Force dalam pencarian Faktor Prima dapat digambarkan sebagai berikut:

1. Dari solusi permasalahan A didapat faktor positif dari bilangan yang dicari.
2. Inisialisasi suatu array of integer dengan nama faktor dan sebuah integer dengan nama faktorcount sebagai tempat penyimpanan himpunan faktor dari bilangan yang dicari.
3. Inisialisasi variabel i sebagai iterator dengan nilai i sama dengan 0.
4. Cek keprimaan isi dari array of integer faktor positif dengan solusi permasalahan B dengan indeks i.
5. Jika faktor positif dengan indeks i adalah bilangan prima, maka masukkan ke dalam faktor dan faktorcount ditambah dengan 1.
6. Ulangi langkah 4 dan 5 dengan nilai i bertambah 1 sampai dengan i bernilai jumlah faktor positifnya.
7. Array of integer faktor sudah berisi faktor prima dari input dan faktorcount berisikan jumlah faktor prima yang dimiliki input.

Berikut adalah *source code* dari pemecahan masalah pencarian faktor positif dalam C++:

```
void
primefactorgenerator(int
input, int factor[], int
&factorcount)
{

  int factortemp[200];
  int counttemp = 0;

  factorgenerator(input,factor,factor
count);
```

```
  for (int i=0; i < factorcount; i++)
      if(isprimenumber(factor[i])){
            factortemp[counttemp] =
factor[i];
            counttemp++;
      }
  }

  factorcount = counttemp;
  for (int i=0; i < counttemp; i++){
      factor[i] = factortemp[i];
  }

}
```

**D. Program Utama**

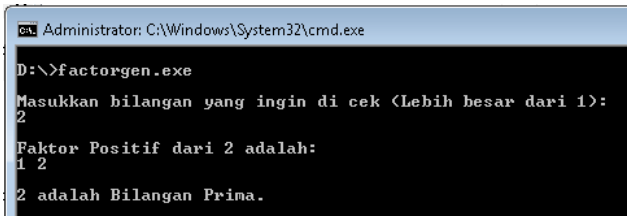
Program utama ini berisikan proses menampilkan hasil eksekusi di layar. Berikut ini adalah *source code* dari program utama dalam bahasa C++:

```
int main(){

  int factor[200];
  int indexfactor = 0;

  int primefactor[200];
  int indexprimefactor = 0;

  cout << "\n";

  int buff;
  cout << "Masukkan bilangan yang
ingin di cek (Lebih besar dari 1):"
<< endl;
  cin >> buff;
  cout << "\n";

  if(buff == 1 || buff < 0){
      cout <<"Bilangan harus lebih
besar dari 1."<< endl;
      return -1;
  }

  factorgenerator(buff,factor,indexf
actor);
  cout << "Faktor Positif dari "<<
buff << " adalah:" << endl;
  for(int i=0; i<= indexfactor-1;
i++){
  cout << factor[i] <<" ";
  }
  cout << "\n" << endl;

  if(isprimenumber(buff)){
      cout << buff << " adalah
Bilangan Prima." << endl;
      return 0;
  }

  primefactorgenerator(buff,p
rimefactor,indexprimefactor)
;
  cout << "Faktor Prima dari
"<< buff << " adalah:" <<
endl;
  for(int i=0; i<=
indexprimefactor-1; i++){
      cout << primefactor[i]
<<" ";
  }

  cout << "\n" << endl;
  cout << buff << " bukan Bilangan
Prima." << endl;

  return 0;

}
```

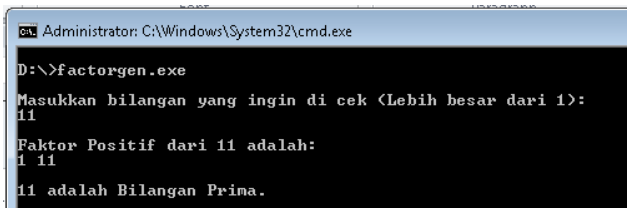
## IV. PENGUJIAN

Dengan Algoritma Brute Force masalah pencarian faktor prima dapat diselesaikan. Semakin besar angka yang hendak dicari, semakin lama pula proses pencarian faktor primanya. Berikut akan dipaparkan hasil eksekusi program pencarian faktor prima ini.

Dibawah ini adalah hasil eksekusi program dengan masukan bilangan prima.



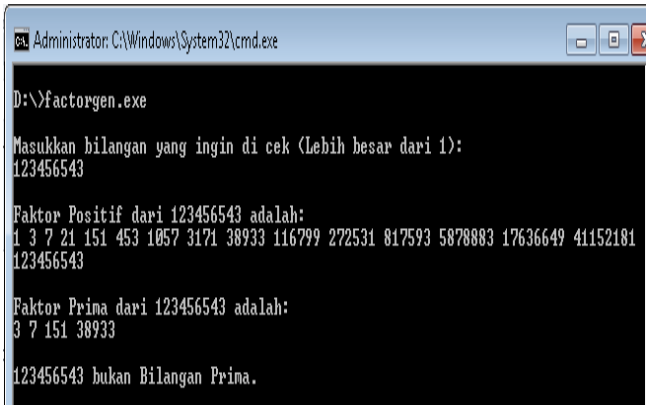

**Gambar 3. Hasil Eksekusi Program dengan input sama dengan 2**



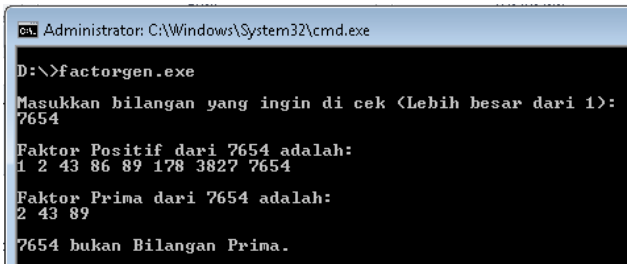

**Gambar 4. Hasil Eksekusi Program dengan input dengan 11**

Jika masukan merupakan bilangan prima, maka hasil faktor primanya tidak akan ditampilkan.

Dibawah ini adalah hasil eksekusi program dengan masukan bukan bilangan prima.




**Gambar 5. Hasil Eksekusi Program dengan input sama dengan 123456543**




**Gambar 6. Hasil Eksekusi Program dengan input sama dengan 7654**

Dapat dilihat bahwa faktor prima dari bilangan masukan di atas dapat dicari. Dari gambar 5 dapat dilihat walaupun faktor positifnya banyak, namun faktor primanya tidak banyak.

## V. KESIMPULAN

Kesimpulan yang dapat diambil dari pengujian di atas antara lain adalah:

1. Algoritma Brute Force dapat menyelesaikan permasalahan pencarian faktor prima suatu bilangan.
2. Algoritma Brute Force ini merupakan algoritma yang mampu menyelesaikan masalah dengan cara sederhana, langsung, dan jelas namun algoritma ini bukanlah algoritma yang mangkus.
3. Pencarian faktor prima ini adalah pengembangan dari algoritma pencarian faktor dari suatu bilangan dan uji keprimaan.

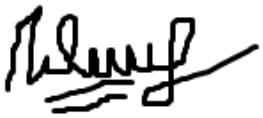
## REFERENCES

[1] Rinaldi Munir, 2005, *Diktat Kuliah IF2251 Strategi Algoritmik*, Program Studi Teknik Informatika ITB.
[2] en.wikipedia.org/number
[3] en.wikipedia.org/prime_number

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 10 Desember 2010



Widhaprasa Ekamatra Waliprana
13508080