# Laporan Tugas Pemrograman - Diffie Hellman

**Rakina Zata Amni, 1306398951**
**Kelas A - Asdos Dinda Susanti**

## Diffie Hellman

Algoritma Diffie Hellman adalah algoritma yang digunakan untuk mendapatkan *shared key*/kunci rahasia bersama. Kunci yang didapatkan dari algoritma ini dapat digunakan untuk enkripsi. Sang pengirim dapat mengenkripsi dengan kunci yang didapat dan sang penerima dapat mendekripsi dengan kunci yang didapat.

## Cara Kerja Algoritma Diffie Hellman

- Kedua pihak (sebut saja Alice dan Bob) pertama-tama harus menyetujui sebuah bilangan prima $p$ dan bilangan bulat $g$ yang merupakan *primitive root* modulo $p$.
- Setelah itu, Alice akan memilih sebuah bilangan bulat rahasia $a$, dan Bob akan memilih sebuah bilangan bulat rahasia $b$. Alice lalu akan menghitung $A = g^a \bmod p$ dan Bob akan menghitung $B = g^b \bmod p$.
- Alice lalu akan mengirimkan $A$ ke Bob dan Bob akan mengirimkan $B$ ke Alice.
- Alice akan menghitung *shared key*nya, yaitu $B^a \bmod p$. Bob akan menghitung *shared key*nya, $A^b \bmod p$.
- Perhatikan bahwa $A^b \bmod p = B^a \bmod p = g^{ab} \bmod p$.
- Alice dan Bob kini memiliki *shared key* yang sama dan rahasia.

## Mengapa algoritma Diffie Hellman dapat dibilang aman?

Perhatikan bahwa informasi-informasi yang bisa kita dapatkan secara publik hanyalah nilai $p$, $g$, $A$, dan $B$. Untuk mendapatkan *shared key,* kita harus menemukan salah satu dari $a$ atau $b$. Ini sama saja dengan *discrete logarithm problem,* dimana kita harus mencari $y$ dan kita hanya diberitahu $x$ dan $x^y$. Kita harus mencoba semua kemungkinan untuk nilai $a$ (atau $b$), dimana banyak kemungkinan nilainya sangat besar. Jika nilai $p$ sangat besar, tentu saja hal ini akan memakan waktu yang lama. Ini berarti algoritma Diffie Hellman susah dipecahkan dan aman.

## Cryptanalysis algoritma Diffie Hellman

Di algoritma *cryptanalysis* saya, saya mencari nilai $a$ dengan cara mencoba semua kemungkinan nilai $a$. Karena g adalah *primitive root* modulo p, nilai $a$ yang harus kita cek adalah 0..$p$-1. Setelah kita dapatkan nilai $a$ (dengan mengecek apakah $g^a \bmod p = A$), kita bisa mendapatkan *shared key*nya dengan menghitung $B^a \bmod p$. Agar dapat menghitung pangkat dengan cepat,

saya menggunakan algoritma perpangkatan modular yang membutuhkan waktu *O(log N).* Maka kompleksitas algoritma saya adalah *O(P log P)*. Program saya dapat berjalan di bawah 1 detik untuk nilai P yang relatif kecil (dibawah 10 juta).

## Program Saya

Saya membuat program untuk menyimulasikan algoritma Diffie Hellman dan *cryptanalysis* terhadap algoritma Diffie Hellman yang menggunakan bilangan prima yang kecil. Program saya menggunakan bahasa C++ dan menempatkan anda sebagai Alice. Saya juga membuat versi *web-app* dalam bahasa PHP. Inti dari kedua program tersebut sama saja.

## Halaman di kawung

mahasiswa.cs.ui.ac.id/~rakina.zata/md2/diffie.html

## Referensi

Dalam mengerjakan tugas ini, saya membaca dan menonton berbagai sumber di internet yang menjelaskan algoritma Diffie Hellman. Sumber-sumber tersebut adalah:

Wikipedia

Khan Academy