Random Number Generator

Tadya Rahanady Hidayat (13509070)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
tadya.rahanady@students.itb.ac.id

ABSTRAK

Makalah ini membahas tentang *Random Number Generator(RNG)* dan pengaplikasiannya. Metode penghitungan utama dalam sistem RNG adalah modulo, maka dari itu RNG berkaitan dengan salah satu materi yang ada pada kuliah Struktur Diskrit, yaitu Teori Bilangan.

Kata Kunci : ***Random Number Generation***, Modulo, Teori Bilangan

1. PENDAHULUAN

Random Number Generator(RNG) adalah sebuah program atau alat untuk menghasilkan urutan angka atau simbol secara tidak teratur. Sistem ini diaplikasikan ke dalam banyak bidang, seperti sampel statistika, simulasi komputer, kriptografi, bahkan untuk desain.

Makalah ini akan membahas beberapa aplikasi RNG, seperti randomisasi di dalam game,sains, kriptografi, dan seni.

2. TEORI BILANGAN

Secara tradisional, teori bilangan adalah cabang dari matematika murni yang mempelajari sifat-sifat bilangan bulat dan mengandung berbagai masalah terbuka yang dapat mudah mengerti sekalipun bukan oleh ahli matematika. Dalam teori bilangan dasar, bilangan bulat dipelajari tanpa menggunakan teknik dari area matematika lainnya.

2.1 Modulo

Operasi modulo merupakan sisa pembagian dari satu bilangan oleh bilangan yang lain. Jika diberikan dua bilangan a dan b, a modulo b (disingkat sebagai a mod b) dapat disamakan dengan sisa dari pembagiannya. Misalnya, "5 mod 4" akan menghasilkan 1, karena 5 dibagi dengan 4 bersisa 1, sedangkan "9 mod 3" akan menghasilkan 0 karena pembagian 9 oleh 3 tidak meninggalkan sisa. Ketika a atau b adalah negatif, definisi ini menjadi memiliki celah dan banyak bahasa pemrograman memberikan definisi yang berbeda-beda.

Meskipun biasanya a dan n keduanya adalah bilangan bulat, banyak sistem penghitungan yang memungkinkan penggunaan jenis operan numerik lainnya.

$$r = a - n \left\lfloor \frac{a}{n} \right\rfloor .$$

Gambar 1. Rumus modulo

Beberapa hasil operasi dengan operator modulo:

(i) 56 mod 9 = 2 (56 = 9 · 6 + 2)
(ii) 32 mod 2 = 0 (32 = 2 · 16 + 0)
(iii) 5 mod 7 = 5 (5 = 7 · 0 + 5)
(iv) 0 mod 7 = 0 (0 = 7 · 0 + 0)
(v) – 21 mod 9 = 6 (–21 = 9 (–3) + 6)
(vi) – 28 mod 14 = 0 (–28 = 14(–2) + 0)

2.2 Kekongruenan

Misalkan 38 mod 5 = 3 dan 13 mod 5 = 3, maka dikatakan 38 ≡ 13 (mod 5) (baca: 38 kongruen dengan 13 dalam modulo 5).

Misalkan *a* dan *b* bilangan bulat dan *m* adalah bilangan > 0, maka ***a* ≡ *b*** (mod *m*) jika *m* habis membagi *a* – *b*.

Jika *a* tidak kongruen dengan *b* dalam modulus *m*, maka ditulis *a* ≡/ *b* (mod *m*).

Contoh :

17 ≡ 2 (mod 3)

(3 habis membagi 17 – 2 = 15)

–7 ≡ 15 (mod 11)

(11 habis membagi –7 – 15 = –22)

12 ≡/ 2 (mod 7)

(7 tidak habis membagi 12 – 2 = 10)

$-7 \not\equiv 15 \pmod 3$

(3 tidak habis membagi –7 – 15 = –22)

2.3 Modulo Invers

Inversi di dalam modulo berbeda dengan inversi bilangan biasa. Untuk mendapatkan suatu invers dari sebuah modulo, beberapa syarat harus terpenuhi, yaitu :

a. Jika *a* dan *m* relatif prima dan *m* > 1, maka balikan (*invers*) dari *a* modulo *m* ada.
b. Balikan dari *a* modulo *m* adalah bilangan bulat *x* sedemikian sehingga

$$xa \equiv 1 \pmod m$$

Pembuktian: *a* dan *m* relatif prima, jadi PBB(*a*, *m*) = 1, dan terdapat bilangan bulat *x* dan *y* sedemikian sehingga

$$xa + ym = 1$$

yang mengimplikasikan bahwa

$$xa + ym \equiv 1 \pmod m$$

Karena $ym \equiv 0 \pmod m$, maka

$$xa \equiv 1 \pmod m$$

Kekongruenan yang terakhir ini berarti bahwa *x* adalah balikan dari *a* modulo *m*.

Contoh inversi modulo :

Tentukan balikan dari 4 (mod 9), 17 (mod 7), dan 18 (mod 10).

Solusi:

(a) Karena PBB(4, 9) = 1, maka balikan dari 4 (mod 9) ada. Dari algoritma Euclidean diperoleh bahwa

$$9 = 2 \cdot 4 + 1$$

Susun persamaan di atas menjadi

$$-2 \cdot 4 + 1 \cdot 9 = 1$$

Dari persamaan terakhir ini kita peroleh –2 adalah balikan dari 4 modulo 9.

Periksa bahwa $-2 \cdot 4 \equiv 1 \pmod 9$

3. RANDOM NUMBER GENERATOR

3.1 Sejarah

Banyaknya aplikasi dari *randomness* atau sifat ketidakteraturan, menyebabkan dikembangkannya berbagai metode untuk menghasilkan data yang tidak teratur. Banyak dari metode ini sudah ada sejak zaman kuno, termasuk dadu , melempar koin, penggunaan kartu, dan teknik lainnya.Karena sifatnya,untuk menghasilkan nomor yang tidak teratur dalam jumlah yang besar membutuhkan banyak pekerjaan dan / atau waktu. Saat ini, setelah munculnya *Random Number Generator*, metode tradisional,cenderung untuk ditinggalkan. Beberapa metode untuk RNG sering memberikan hasil yang kurang memuaskan dari tujuannya, meskipun mungkin sesuai dengan yang diinginkan. Dengan keberhasilan yang bervariasi, beberapa uji statistik tidak teratur dimaksudkan untuk mengukur hasil yang tak terduga. Pada tahun 2010 ditemukan RNG yang sempurna, berlandaskan prinsip-prinsip fisika kuantum.

3.2 Metode

Metode-metode untuk menghasilkan angka acak yang dilakukan terbagi menjadi beberapa cara, dari yang paling sederhana, hingga metode yang cukup rumit. Beberapa contohnya adalah :

3.2.1 Metode Fisik

Metode paling pertama untuk menghasilkan angka secara acak adalah dengan menggunakan dadu, koin, rolet, dan lain sebagainya. Sampai saat ini, metode ini masih cukup sering digunakan, terutama di dalam game dan perjudian. Karena metode ini dianggap terlalu lambat, pengaplikasiannya untuk statistika dan kriptografi kurang begitu popular saat ini.

Dasar dari metode fisik adalah fenomena fisika atomik atau subatomik acak yang tidak bisa diprediksi dapat dilacak dengan menggunakan mekanika kuantum.

3.2.2 Metode Distribusi Probabilitas

Metode ini menggunakan fungsi densitas probabilitas. Metode ini bekerja cukup baik untuk menghasilkan *pseudo-random* dan *true random number*. Salah satu metode, yaitu metode inverse, mengintegralkan area lebih dari sama dengan bilangan acak.

Metode kedua, *acceptance-rejection*, memilih antara nilai x dan y, lalu membandingkan apakah fungsi x lebih besar dari nilai y. Apabila fungsi x lebih dari nilai y, maka nilai x akan diterima. Jika sebaliknya, maka nilai x akan ditolak dan algoritmanya akan mencoba ulang.

3.2.3 Metode Komputasi

Metode ini menggunakan algoritma bernama *Pseudo-random number generator* yang secara otomatis menghasilkan serangkaian angka acak yang memiliki kualitas baik. Nilai yang dihasilkan oleh algoritma tersebut secara umum ditentukan dengan sebuah konstanta yang disebut *seed.* Salah satu PRNG yang umum adalah *linear congruential generator*, yang menggunakan rekurens dari persamaan

$$X_{n+1} = (aX_n + b) \bmod m$$

Untuk menghindari sifat non-acak yang muncul dari *linear congruential generator,* beberapa *random number generator* dengan koefisin nilai pengali yang berbeda-beda dapat digunakan secara paralel.

Beberapa bahasa pemrograman memiliki fungsi yang bersifat *random number generator.* Fungsi-fungsi ini biasanya digunkan untuk menghasilkan angka, kata, atau bilangan real yang tersebar diantara 0 dan 1. Fungsi-fungsi tersebut biasanya memiliki sifat statistika yang buruk. Biasanya fungsi-fungsi tersebut diinisialisasi menggunakan *real time clock* sebagai *seed* menyebabkan perhitungan yang dilakukan di dalam millisecond dan sangat jauh jika dibandingkan dengan presisi manusia.

Fungsi-fungsi tersebut memberikan hasil yang cukup untuk beberapa tugas (contohnya video game), tetapi tidak cocok digunakan saat tingkat acak yang dibutuhkan sangat tinggi, seperti aplikasi untuk kriptografi dan analisis numerik dalam statistik.

Salah satu contoh sederhana *pseudo-random number generator* adalah metode *Multiply-with-carry* yang ditemukan oleh George Marsaglia. Program ini memiliki kecepatan dan sifat acak yang cukup baik.

Contoh *random number generator* :

```
m_w = <choose-initializer>;    /* must
not be zero */
m_z = <choose-initializer>;    /* must
not be zero */

uint get_random()
{
    m_z = 36969 * (m_z & 65535) + (m_z
>> 16);
    m_w = 18000 * (m_w & 65535) + (m_w
>> 16);
    return (m_z << 16) + m_w;  /* 32-
bit result */
}
```

Contoh fungsi sederhana dalam Bahasa C :

```
int rand()
{
  random_seed = random_seed *
1103515245 +12345;
  return (unsigned int)(random_seed /
65536) % 32768;
}
```

4. APLIKASI

Random Number Generator dapat diaplikasikan untuk berbagai hal yang membutuhkan sifat acak, seperti judi, statistika, kriptografi, seni, dan lain sebagainya.

4.1 Games

Kegunaan dari angka acak pertama kali diteliti di dalam konteks perjudian. Banyak alat yang menghasilkan angka acak seperti dadu dan rolet dikembangkan untuk digunakan di dalam *games of chance*. *Games* elektronik modern biasanya memuat lebih dari satu *random number generation* untuk menghasilkan angka acak di dalam *game.* Sistem *random number generator* diterapkan juga untuk mekanisme *loot system* dalam *massively multiplayer online role-playing games(MMORPG).* Salah satu aspek di dalam *online game* adalah *loot* (baran yang dijatuhkan monster). Mekanisme *random number generator* mengabaikan semua *input* dari pemain dan memberikan statistika yang sama untuk semua pemain, sehingga membuat terciptanya sebuah system yang ‘adil’. Karena adanya kesamaan antara menggunakan *random number generator* dan melempar (*rolling*) dadu, proses ini juga dikenal sebagai “***rolling***”, dan “***rolling for loot*** ” adalah proses dimana setiap pemain diberikan nomor acak dengan *random number generator* yang menentukan apakah mereka akan mendapatkan barang yang diinginkan atau tidak.

4.2 Politik

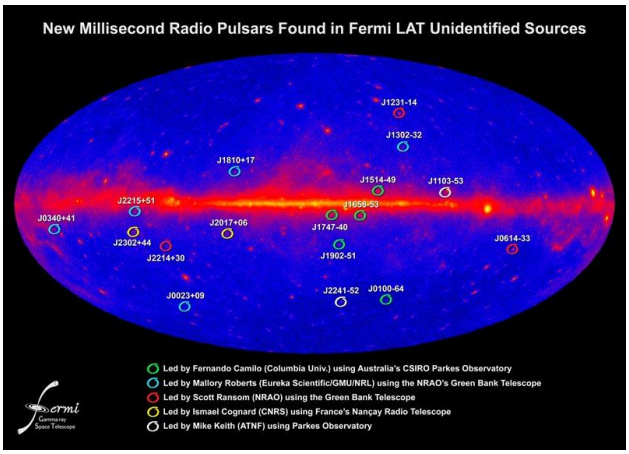
Pada abad ke 5 Sebelum Masehi, Demokrasi Athena mengembangkan gagasan *isonomia*(kesetaraan dalam hak politik), dan pemilihan acak adalah cara untuk mendapatkan hak ini. Pemilihan di Athena dikatakan tidak demokratis, karena penduduk yang terpilih karena popularitas bertentangan dengan kesetaraan demokratis yang dituju. Karena itu, pelaksanaan korupsi tidak akan bisa dilakukan karena tidak ada seorang pun yang tahu siapa yang akan terpilih.

4.3 Sains

Angka tidak teratur memiliki peran yang cukup banyak, salah satunya adalah fisika. Banyak metode analisa statistika, seperti Metode Bootstrap, membutuhkan angka tidak teratur. Metode Monte Carlo dalam fisika dan ilmu komputer membutuhkan angka tidak terbantur. Angka tidak teratur biasa digunakan di dalam *parapsychology* untuk menguji *precognition*(firasat).

Statistika berdasarkan teori statistika berakar dari konsep *randomness*(ketidakteraturan). Banyak elemen dari statistika bergantung pada konsep ketidakteraturan dari *random number*. Ketika angka tidak teratur tersebut 'gagal' untuk menjadi tidak teratur, analisa statistic dapat mengalami bias. Elemen statistika yang bergantung pada angka tidak teratur termasuk : memilih sampel, menyembunyikan protokol yang dipelajari dari subjek, dan simulasi Monte Carlo.

Dalam fisika, banyak eksperimen bergantung pada analisa statistika untuk hasil keluarannya. Sebagai contoh, sebuah eksperimen mengumpulkan sinar-x yang bersumber dari benda astronomis dan menganalisa hasilnya untuk sinyal periodik. Karena *random noise* dapat muncul, analisa statistika dibutuhkan untuk mendeteksi sinyal yang menyerupai sinyal yang sesungguhnya.



Gambar 2. Sinar-X dari pulsar

Saat fenomena nyata terpengaruhi oleh proses yang *unpredictable*, seperti *radio noise* atau cuaca, proses tersebut dapat disimulasikan menggunakan *random* atau *pseudo-random number*. *Random number generator* otomatis pada awalnya dibuat untuk mensimulasikan fenomena fisika, terutama untuk transfer neutron dalam reaksi fisi nuklir.

Pseudo-random numbers sering kali digunkan di dalam simulasi kejadian statistika, contoh yang sederhana adalah hasil dari melempar koin. Situasi yang lebih kompleks adalah simulasi populasi genetika atau sifat dari partikel sub-atom. Metode simulasi seperti itu sering disebut metode *stochastic*, yang memiliki banyak aplikasi di dalam simulasi komputer.

4.4 Kriptografi

Serangkaian angka tidak teratur digunakan di dalam kriptografi yang mendukung sistem keamanan di dalam komunikasi modern. Sebagai contoh, jika pengguna ingin menggunakan *encryption algorithm*, lebih baik jika mereka memilih angka tidak teratur sebagai kunci. Angka tersebut harus memiliki 'tingkat' yang tinggi, sehingga menambah tingkat keamanan.

Angka yang benar-benar tidak teratur diperlukan untuk menjamin keamanan yang dibentuk dari *one-time pad*, satu-satunya algoritma enkripsi yang terbukti tidak dapat dipecahkan. Terlebih lagi rangkaian tidak teratur tersebut tidak dapat digunakan ulang dan tidak akan pernah bisa dibuka oleh penyerang.

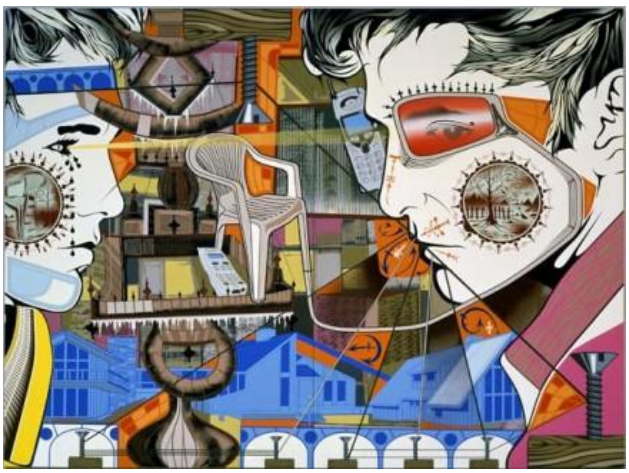
Karena persyaratan dalam kriptografi adalah perbedaan yang bervariasi, setiap urutan tidak teratur yang dipublikasikan adalah pilihan yang buruk. Di dalam kriptografi, angka tidak teratur haruslah tidak hanya acak, tetapi juga harus rahasia dan sulit diprediksi.

Gambar 3. Contoh kode kriptografi

4.5 Literatur, Musik, dan Seni

Beberapa teori estetika berdasar dari sifat tidak teratur. Salah satu contoh kebutuhan akan sifat tidak teratur yaitu untuk penempatan barang di dalam pameran seni. Meski biasanya dihindari dengan menggunakan tema. Dadaisme telah berusaha mengakui sifat ketidakteraturan dalam

berbagai cara. Biasanya orang-orang menyalahartikan sifat tidak teratur dikarenakan kurangnya informasi, padahal di dalam beberapa teori seni, semua seni adalah tidak teratur dikarenakan itu “hanya cat dan kanvas”.



Gambar 4. Salah satu lukisan abstrak

4.6 Aplikasi Lainnya

Angka tidak teratur juga biasa digunakan dimana ‘keadilan’ dapat dicapai dengan randomisasi seperti juri dan draft lotre militer. Contoh lainnya termasuk, memilih “*Random Quote of the Day*” untuk situs web. Bentuk lebih lemah dari ketidakteraturan juga terikat dengan algoritma hash, pencarian armortisasi, dan algoritma sorting.

5. KESIMPULAN

Salah satu aplikasi dari materi Teori Bilangan, yaitu modulo, memiliki beberapa kegunaan yang cukup bermanfaat. Salah satunya adalah sebagai dasar untuk pembuatan sebuah mesin atau program *random number generator*.

Random number generator memilik banyak aplikasi, seperti *game*, politik, sains, kriptografi, seni, dan lain sebagainya. *Random number generator* sering digunakan untuk sistem yang membutuhkan serangkaian angka tidak teratur untuk masukan atau keluaran dari sistem tersebut.

REFERENSI

[1] http://en.wikipedia.org/wiki/Random_number_generation. Tanggal akses : 14 Desember 2010 pukul 20:00 WIB

[2] Munir, Rinaldi, Slide Kuliah IF2091, Struktur Diskrit, bagian Teori Bilangan, 2010

[3] http://en.wikipedia.org/wiki/Applications_of_randomness. Tanggal akses : 14 Desember 2010 pukul 20:00 WIB

[4] http://id.wikipedia.org/wiki/Teori_bilangan. Tanggal akses : 14 Desember 2010 pukul 20:00 WIB