Bilangan Acak

- Haahr, Mads. "Introduction to Randomness and Random Numbers". http://random.org/randomness/. Retrieved 2009-04-08.
- Achmad Basuki, Lab. Computer Vision, EEPIS-ITS Surabaya.
- Wikipedia.org

Random Number

- Terdapat dua pendekatan untuk men-generate random number:
  - Pseudo-Random Number Generators (PRNGs).

    contoh: Linear Congruent Method
  - True Random Number Generators (TRNGs).

Characteristic	Pseudo-Random Number Generators	True Random Number Generators
Efficiency	Excellent	Poor
Determinism	Determinstic	Nondeterministic
Periodicity	Periodic	Aperiodic

True Random Number Generators (TRNGs).

- TRNGs extract randomness from physical phenomena and introduce it into a computer.
- Contoh:
  - penguapan radioactive.
  - atmospheric noise.

Thunderstorms generate atmospheric noise

Pembangkitan Bilangan Pseudo-Acak

- Awalnya dihasilkan dengan teknik mesin pemintal, melempar dadu, mengocok kartu.
- Saat ini menggunakan komputer untuk menghasilkan bilangan pseudo-acak.
- Bilangan acak yang dibangkitkan oleh komputer merupakan bilangan acak semu, karena pembangkitannya menggunakan operasi-operasi aritmatika.
- Banyak algoritma atau metode yang dapat digunakan untuk membangkitkan bilangan acak.

Linear Congruent Method

- Linear Congruent Method (LCM) merupakan metode pembangkitkan bilangan acak yang banyak digunakan dalam program komputer.

The generator is defined by the recurrence relation:

$$X_{n+1} = (aX_n + c) \mod m$$

where X_n is the sequence of pseudorandom values, and

m, $0 < m$ — the "modulus"
a, $0 < a < m$ — the "multiplier"
c, $0 \leq c < m$ — the "increment" (the special case of $c = 0$ corresponds to Park–Miller RNG)
X_0, $0 \leq X_0 < m$ — the "seed" or "start value"

are integer constants that specify the generator.

Contoh

- Membangkitkan bilangan acak sebanyak 8 kali dengan a=2, c=7, m = 10 dan x(0)=2

  x(1) = (2 (2) + 7) mod 10 = 1

  x(2) = (2 (1) + 7) mod 10 = 9

  x(3) = (2 (9) + 7) mod 10 = 5

  x(4) = (2 (5) + 7) mod 10 = 7

  x(5) = (2 (7) + 7) mod 10 = 1

  x(6) = (2 (1) + 7) mod 10 = 9

  x(7) = (2 (9) + 7) mod 10 = 5

  x(8) = (2 (5) + 7) mod 10 = 7

Bilangan acak yang dibangkitkan adalah: 1 9 5 7 1 9 7

Terjadi pengulangan bilangan secara periodik(4)

Contoh 2

- Membangkitkan bilangan acak sebanyak 8 kali dengan a=4, c=7, m = 15 dan x(0)=3

  x(1) = (4 (3) + 7) mod 15 = 4

  x(2) = (4 (4) + 7) mod 15 = 8

  x(3) = (4 (8) + 7) mod 15 = 5

  x(4) = (4 (5) + 7) mod 15 = 12

  x(5) = (4 (12) + 7) mod 15 = 10

  x(6) = (4 (10) + 7) mod 15 = 2

  x(7) = (4 (2) + 7) mod 15 = 0

  x(8) = (4 (0) + 7) mod 15 = 7

Bilangan acak yang dibangkitkan a dalah: 4 8 5 12 10 2 0 7

Tidak terlihat pengulangan bilangan secara periodik

Contoh 3

- Membangkitkan bilangan acak sebanyak 16 kali dengan a=4, c=7, m = 15 dan x(0)=3

x(1) = (4 (3) + 7) mod 15 = 4	x(9) = (4 (7) + 7) mod 15 = 13
x(2) = (4 (4) + 7) mod 15 = 8	x(10) = (4 (13) + 7) mod 15 = 14
x(3) = (4 (8) + 7) mod 15 = 5	x(11) = (4 (14) + 7) mod 15 = 3
x(4) = (4 (5) + 7) mod 15 = 12	x(12) = (4 (3) + 7) mod 15 = 4
x(5) = (4 (12) + 7) mod 15 = 10	x(13) = (4 (4) + 7) mod 15 = 8
x(6) = (4 (10) + 7) mod 15 = 2	x(14) = (4 (8) + 7) mod 15 = 5
x(7) = (4 (2) + 7) mod 15 = 0	x(15) = (4 (5) + 7) mod 15 = 12
x(8) = (4 (0) + 7) mod 15 = 7	x(16) = (4 (12) + 7) mod 15 = 10

Bilangan acak yang dibangkitkan adalah: 4 8 5 12 10 2 0 7 13 14 3 4 8 5 12 10

Terlihat pengulangan bilangan secara periodik(10)

Kesimpulan

- Terjadi pengulangan pada periode waktu tertentu atau setelah sekian kali pembangkitan, hal ini adalah salah satu sifat dari metode ini, dan *pseudo random generator pada umumnya.*
- Penentuan konstanta LCM (a, c dan m) sangat menentukan baik tidaknya bilangan acak yang diperoleh dalam arti memperoleh bilangan acak yang seakan-akan tidak terjadi pengulangan.

rand() functions in runtime libraries of various compilers

Source	m	a	c	output bits of seed in *rand()* / *Random(L)*
Numerical Recipes	2^{32}	1664525	1013904223	
Borland C/C++	2^{32}	22695477	1	bits 30..16 in *rand()*, 30..0 in *lrand()*
glibc (used by GCC)[4]	2^{32}	1103515245	12345	bits 30..0
ANSI C: Watcom, Digital Mars, CodeWarrior, IBM VisualAge C/C++	2^{32}	1103515245	12345	bits 30..16
Borland Delphi, Virtual Pascal	2^{32}	134775813	1	bits 63..32 of *(seed * L)*
Microsoft Visual/Quick C/C++	2^{32}	214013	2531011	bits 30..16
Apple CarbonLib	$2^{31} - 1$	16807	0	see Park–Miller RNG
MMIX by Donald Knuth	2^{64}	6364136223846793005	1442695040888963407	
VAX's **MTH$RANDOM**,[5] old versions of glibc	2^{32}	69069	1	
Random class in Java API	2^{48}	25214903917	11	bits 48...17

[*citation needed*]

Park-Miller random number generator

- The **Park–Miller random number generator** disebut juga sebagai **Lehmer random number generator** adalah variant dari linear congruential generator.

$$X_{k+1} = X_k \cdot g \mod n$$

Dimana,

n adalah bil. Prima atau pangkat bil. Prima;

n = 2^{31}-1 = 2.147.483.647(Mersenne prime: M_{31})

g adalah multiplier (multiplicative order);

g = 16.807 (multiplicative order dari M_{31}.)

Tugas

- Cari metode lain untuk generator random number.