# Studi Analisis Perbandingan Pembangkit Bilangan Acak Berbasis Kuantum Dan Algoritma Konvensional

Andika Pratama – NIM 13507005
*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
*If17005@students,if.itb.ac.id*

***Abstrak*— Bilangan acak merupakan salah satu faktor yang sangat penting dalam kriptografi. Akan etapi membangkitkan bilangan acak yang baik adalah masalah yang ternyata cukup sulit. Pembangkit bilangan acak (RNG) yang banyak dipakai adalah pembangkit bilangan acak semu (PRNG) dimana bilangan yang dihasilkannya tidak benar-benar acak, melainkan mengikuti sebuah aturan atau algoritma tertentu. Berdasarkan teori fisika, mekanika fisika makro tidak dapat membangkitkan bilangan acak sejati, dan satu-satunya cara membuat pembangkit bilangan acak sejati (TRNG) adalah dengan memanfaatkan fenomena fisika dalam level atomik, yaitu mekanika kuantum. Dalam makalah ini penulis akan membahas teknik-teknik pembuatan RNG, membedah teknologi RNG berbasis mekanika fisika kuantum (QRNG) yang menjanjikan keacakan sejati, dan melakukan studi analisis perbandingan performa dan kualitas QRNG dan PSRNG biasa yang menggunakan algoritma konvensional dengan menggunankan metode pengujian statistik.**

***Kata kunci*— pembangkit bilangan acak, kriptografi, mekanika kuantum, PRNG, TRNG, QRNG, pengujian statistik.**

## I. PENDAHULUAN

Bilangan acak merupakan salah satu faktor yang sangat penting dalam kriptografi. Hal ini disebabkan karena bilangan acak menjadi dasar perhitungan dalam kriptografi, yang kemudian menentukan kekuatan dari kriptografi itu sendiri. Misalnya untuk pembangkitan parameter kunci pada algoritma kunci-publik, pembangkitan initialization vector (IV) pada algoritma kunci-simetri, dan sebagainya.

Tidak ada cara konvensional yang bisa benar-benar menghasilkan deret bilangan acak secara sempurna. Umumnya cara yang digunakan dalam membangkitkan bilangan acak adalah berdasarkan suatu algoritma atau fungsi tertentu yang deterministik, sehingga sebenarnya bilangan yang dibangkitkan tersebut bersifat pseudorandom, karena pembangkitan bilangannya dapat diulang kembali. Apabila bilangan acak yang menjadi dasar dalam kriptografi tersebut bersifat pseudorandom, akan memudahkan bagi kriptanalis untuk memecahkan enkripsi / dekripsi. Oleh karena itu, dibutuhkan suatu pembangkit bilangan yang benar-benar acak.

Pembangkit bilangan acak berbasis kuantum adalah teknologi yang masih sangat baru, membangkitkan bilangan acak berdasarkan fenomena fisika kuantum. Menurut teori fisika modern, teknologi ini menjanjikan pembangkitan bilangan yang sepenuhnya acak: tak bisa diprediksi ataupun direplikasi. Teknologi Pembangkit bilangan acak berbasis kuantum ini menarik dan perlu dianalisa lebih lanjut.

## II. PEMBANGKITAN BILANGAN ACAK

Bilangan acak dibutuhkan dibanyak kegiatan, misalnya program kriptografi, permainan, simulasi komputer, sampling statisik, dan lain-lain. Pembangkit bilangan acak, atau RNG(*Random Number Generator*) adalah alat yang dapat memberikan bilangan secara acak. Pembangkit bilangan acak sendiri dapat dibagi menjadi dua jenis, yaitu:

1. Pembangkit Bilangan Acak Sejati, atau TRNG (*True Random Number Generator)*, dimana bilangan yang dibangkitkannya benar-benar acak.
2. Pembangkit Bilangan Acak Semu, atau PRNG (*Psedo Random Number Generator)*, dimana bilangan yang dibangkitkannya tidak benar-benar acak, hanya kelihatannya saja acak.

Secara prinsip oleh teori matematika, sebuah pembangkitan bilangan acak yang sejati(TRNG) adalah pembangkit yang dimana sekuens bilangan yang dibangkitkan tidak dapat diprediksi maupun direplikasi dengan cara apapun. Pembangkit sekuens bilangan yang tidak memnuhi syarat tersebutdinyatakan sebagai PRNG. Selain perbedaan prinsip dasar tersebut, terdapat juga beberapa perbedaan karakteristik dari TRNG dan PRNG. Berikut adalah tabel perbedaan umum antara TRNG dan PRNG:

| Karakteristik | TRNG | PRNG |
|---|---|---|
| **Efisiensi** | Relatif jelek | Relatif bagus |
| **Deterministik** | tidak | ya |
| **Periodik** | tidak | ya |

*Tabel 1Perbandingan karakteristik PRNG dan TRNG*

PRNG umumnya efisien, yang berarti dapat memproduksi banyak bilangan dalam waktu singkat, serta bersifat deterministik, yang berarti sebua sekuen bilangan dapat direproduksi lagi apabila kondisi awalnya

diketahui. PRNG ujuga umumnya bersifat periodik, yang berarti bahwa sekuens bilangan pada akhirnya akan berulang. Sedangkan TRNG bersifat non-deterministik, yang berarti suatu bilangan sekuens bilangan tidak akan bisa direproduksi kembali, juga sekuens bilangan yang dihasilkan tidak memiliki periode pengulangan. Kelemahan dari TRNG yang ada adalah, dibandingkan PRNG, efisiennya relatif masih kalah jauh.

Walau PRNG secara prinsip adalah cacat karena dapat diprediksi atau direplikasi, RNG jenis ini tetapi tetap banyak dipakai. Hal ini disebabkan karena secara umum, PRNG jauh lebih mudah untuk diimplementasikan, lebih mudah digunakan, dan untuk beberapa kasus kebutuhan, seperti simulasi komputer, sifat deterministik memiliki keuntungan tersendiri.

Dalam implementasi RNG secara umum, terdapat dua pendekatan untuk membangkitkan bilangan acak, yaitu:

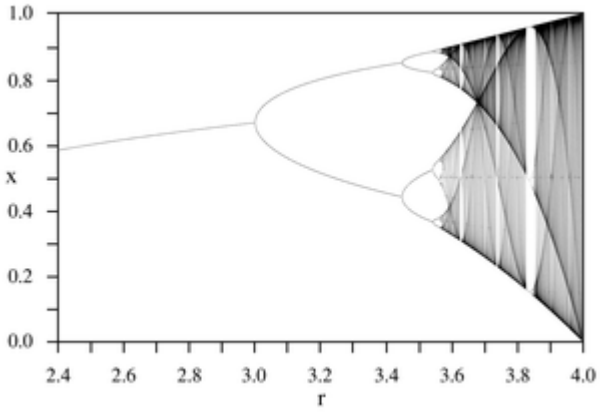
### *A. Algoritma Pembangkit Sekuens Bilangan*

Pendekatan ini menggunakan sebuah algoritma yang bisa membangkitkan sekuens bilangan dari sebuah *seed*, atau sebuah variabel awal. Karena komputasi di komputer konvensional secara inherennya merupakan proses yang deterministik, tidak ada algoritma yang dapat membangkitkan bilangan yang benar-benar acak, sehingga pendekatan ini adalah termasuk PRNG.

Kualitas pembangkitan bilangan acak oleh setiap algoritma PRNG berbeda-beda, dan dapat dibagi menjadi dua jenis, yaitu:

1. *Cryptographically Secure Pseudorandom Generator* (CSPRNG) yaitu PRNG yang cocok digunakan dalam aplikasi kriptografi. Sebuah PRNG dikatakan CSPRNG apabila lolos uji keacakan statistik dan tahan terhadap serangan prediksi bilangan acak.
   *Contoh algoritma:*
   - Blum Blum Shut (BBS)
   - Berbasis RSA
   - Persamaan Berbasis Teori Chaos
   - dll.

2. *Cryptographically Unsecure Pseudorandom Generator* (CUPRNG) yaitu PRNG yang tidak cocok digunakan dalam aplikasi kriptografi. RNG ini mudah diprediksi, dan hanya digunakan untuk kebutuhan dimana kualitas keacakan tidak begitu perlu diperhatikan.
   *Contoh algoritma:*
   - Linear Congruential Generator(LCG)
   - Mersenne twister
   - dll.




*Gambar 1Contoh Diagram Bifurkasi sebuah persamaan chaos, darimana PRNG membangkitkan bilangan acak*

### *B. Pengukuran Fenomena Fisika*

Dalam kebutuhan dimana PRNG dengan pendekatan algoritan dianggap tidak cukup sesuai, pilihan lainnya adalah dengan hardware RNG fisik yang membangkitkan nilai bilangan berdasarkan dari pengukuran sebuah fenomena fisika. Contoh sederhananya dalam kehidupan sehari-hari adalah mengocok dadu, dimana secara acak dadu akan menampilkan bilangan antara satu sampai enam bila dilempar. Dalam implementasi seriusnya, RNG yang menggunakan pendekatan ini biasanya merupakan sebuah hardware yang menghasilkan sekuens bit berdasarkan pengukuran fenomena fisika dari sebuah sensor.

Masalah utama dari RNG dengan mengukur fenomena fisika adalah masalah bias. Sebuah RNG biner akan dianggap bias apabila probabilitas suatu kemungkinan kejadian tidak sama dengan probabilitas alternatif kejadian. Kesulitannya adalah membuat sensor yang dapat 100% tidak bias terhadap fenomena yang hendak diamati. Namun apabila tidak bisa, maka biasanya digunakan algoritma *post-processing* untuk menghilangkan bias, seperti prosedur *Von Neumann*, tetapi proses ini akan memendekkan data yang dihasilkan.

Selain masalah bias, kualitas keacakan yang dihasilkan dari RNG melalui pendekatan ini sangatlah bergantung pada sifat fundamental proses fenomena fisika yang diobservasi sensor. Fenomena fisika yang bisa dipilih adalah fenomena yang hasil pengukuran akan berfluktuasi tanpa adanya pattern yang bisa diprediksi atau direplikasi, juga sensor yang digunakan harus cukup akurat dan meminimasi bias. Fenomena fisika acak yang umumnya digunakan oleh RNG digunakan dapat dikategorikan menjadi dua jenis, yaitu:

***Chaotic Systems***

Adalah sebuah sistem proses fenomena fisika dimana perubahan kecil di kondisi awal akan dapat menghasilkan perubahan besar prilaku sistem. Sebenarnya menurut teori, proses fisika yang terjadi dalam fenomena ini masih bersifat deterministik, sehingga menurut teori fisika bilangan acaknya masih dapat diprediksi dan berarti termasuk PRNG.

Walaupun secara teknik *Chaotic Systems* merupakan PRNG, prilaku *Chaotic Systems* tetaplah sangat sulit untuk diprediksi, karena untuk bisa memprediksi dengan akurat, memerlukan pengetahuan posisi dan kecepatan dari setiap molekul dalam sistem dan kekuatan komputasi yang teramat besar. Hal ini hampir mustahil untuk dilakukan, sehingga beberapa orang menganggap RNG yang berbasis *Chaotic Systems* bisa digolongkan sebagai TRNG.

Contoh *Chaotic Systems* antara lain:

- Johnson–Nyquist noise
- Avalanche noise
- Atmospheric noise

*Gambar 2Contoh Chaotic Systems: Atmospheric Noise yang disebabkan oleh Petir*

***Quantum Events***

Adalah sebuah kejadian fisika yang dihasilkan dari proses mekanika quantum. Mekanika quantum sendiri adalah cabang keilmuan dari fisika teoritis yang menjelaskan proses alam dalam level atomik dan subatomik. RNG berbasis quantum, atau QRNG, menggunakan fakta bahwa pengamatan dari prilaku partikel subatomik akan bersifat acak dalam keadaan tertentu.

Menurut teori fisika terkini, dipercaya bahwa kejadian-kejadian kuantum dalam level partike l dan atom ini sepenuhnya non-deterministik, sehingga mustahil untuk diprediksi. Karena itu keacakan dari fenomena quantum dapat dibuktikan adalah keacakan sejati, dan RNG yang menggunakan fenomena ini adalah TRNG. RNG ini menjanjikan kualitas bilangan acak yang terbaik yang mungkin dicapai secara teoritis. RNG yang dibangun didasari pada fenomena ini akan dibahas lebih lanjut pada bab III.

## III. TRNG BERBASIS QUANTUM (QRNG)

### *A. Teknik QRNG yang telah ada*

Berdasarkan fenomena *Radioactive Decay*

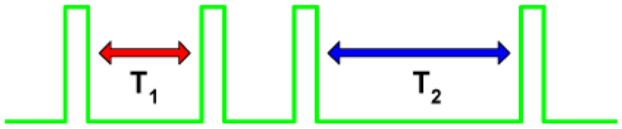
Teknik ini menggunakan ketidakpastian dari radiasi beta pada materi yang bersifat radioaktif. materi yang bersifat radioaktif (Contoh: uranium, caesium,dll) akan cenderung melepaskan radiasi untuk berubah menjadi elemen yang lebih stabil. Proses inilah yang disebut *Radioactive Decay*. Kecepatan dari elemen tersebut melepaskan radiasi dinyatakan dalam *half-life*, yaitu ukuran rata-rata waktu kapan jumlah elemen radioaktif suatu materi akan tinggal setengahnya. *Half-life* setiap elemen berbeda, misalnya materi Cæsium-137 memiliki *half-life* 30.17 tahun, sedangkan Sodium-35 memiliki *half-life* kurang dari sedetik.

$$^{137}\mathrm{Cs} \xrightarrow{30.17\mathrm{y}} {}^{137\mathrm{m}}\mathrm{Ba} + \beta^- + \overline{\nu_e} \xrightarrow{156\mathrm{s}} {}^{137}\mathrm{Ba} + \gamma$$

*Gambar 3 Reaksi Radiactive Decay pada Caesium-137*

Hal yang menarik dalam fenomena ini adalah, walaupun dapat diprediksi laju radioaktif suatu elemen, secara prinsip tidak dapat diketahui kapan tepatnya suatu atom akan melepaskan partikel radiasi tersebut. Apakah sebuah atom akan melepaskan partikel radiasinya sedetik lagi, atau sepuluh tahun lagi, tak bisa diketahui secara pasti.

Keacakan dari waktu radiasi terjadilah yang dieksploitasi untuk membangkitkan bilangan acak.[1] karena waktu sebuah atom melepaskan radiasi bersifat acak, maka rentang waktu dari dua kejadian *decay* berturut-turut juga akan bersifat acak. Maka metode yang dilakukan adalah mengukur rentang waktu tersebut dan membandingkannya dengan rentang waktu berikutnya, lalu menghasilkan bit 1 atau 0 berdasarkan hasil perbandingan.




*Gambar 4Pengukuran Rentang Waktu Radioactive Decay*

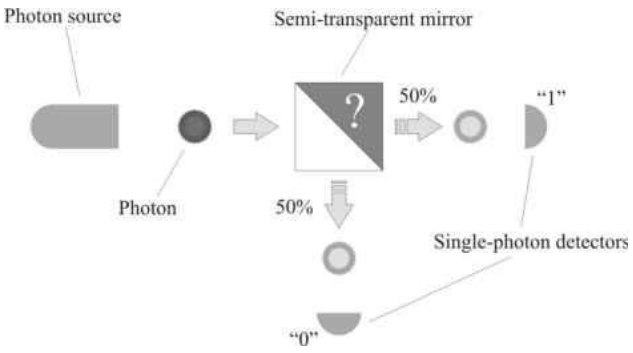
Contoh aplikasi teknik ini adalah pada QRNG Fourmilab HotBits[1], dmana metode pengukuran merakan terlihat seperti pada gambar diatas. Apa bila rentang T1 lebih kecil dari rentang T2, maka dihasilkan bit 0. Jika rentang T1 lebih besar dari T2, dihasilkan bit 1. Apabila rentang T1 dan T2 sama, maka data ini diabaikan.

Teknik ini adalah teknik lama, yang semakin jarang digunakan. Hal ini antara lain karena kecepatan pembangkitan bilangan acak dengan teknik ini kurang dapat diandalkan, serta karena sumber radioaktif yang dibutuhkan adalah bahan yang berbahaya bagi kesehatan manusia.

Berdasarkan prilaku Foton

Optik adalah ilmu mengenai cahaya. Dari pandangan fisika kuantum, cahaya terbentuk dari partikel-partikel yang disebut foton. Foton dalam keadaan tertentu akan memberikan prilaku yang bersifat acak. Salah satu conotoh keadaan untuuk itu, yangh cocok untuk kebutuhan pembangkitan bilangan acak, adalah trasnmisi pada cermin semi-transparan. Adalah fakta fisika bahwa kejadian apakah foton tersebut dipantulkan atau tidak

bersifat acak dan tidak dapat dipengaruhi oleh faktor internal[3].




*Gambar 5Sistem Optik pembangkitan bilangan acak*

Penerapan QRNG menggunakan cara ini ini telihat pada gambar diatas. Sebuah digunakan sebuah sumber foton, misalnya *light emitting diode* yang foton yang dihasilkan diaarahkan kepada cermin semi transparan dan dua sensor untuk mendeteksi apakan foton dipantulkan atau memnembus cermin. Dalam gambar diatas, bit 1 dihasilkan apabila foton menembus cermin, sedangkan bit 0 dihasilkan bila foton dipantulkan.

Teknik ini adalah teknologi baru yang umurnya baru beberapa tahun, dan lebih handal dibandingkan dengan teknik *radioative decay*.Telah muncul beberapa produk QRNG komersil dan hampir semuanya didasari pada teknik ini.

## IV. EKSPERIMEN PENGUJIAN QNRG

### *A. Metode Pengujian*

Tidak ada metode pengujian yang dapat menentukan apakah sebuah sekuens bilangan hasil sebuah RNG adalah acak atau tidak dengan pasti. Hal ini karena untuk TRNG, setiap kemungkinan sekuens bilangan memilki probabilitas yang sama. Fakta ini berarti sebuah TRNG juga dapat membangkitkan sekuens bilangan yang bagi manusia tampak tidak acak.

Karena tidak ada cara yang dapat menentukan dengan pasti, maka akan dilakukan pendekatan pragmatis berupa uji statistik yang dapat memberikan nilai kepercayaan mengenai keacakan RNG tersebut. Terdapat banyak pengujian statistik yang tersedia untuk mengecek keacakan deret bilangan, dan akan digunakan beberapa diantaranya.

Untuk makalah ini, eksperimen dilakukan pada sampel-sampel sekuens byte acak sebesar satu mega byte yang dibangkitkan oleh RNG-RNG berikut:

- CSPRNG biasa dengan menggunakan Kelas RNGCryptoServiceProvider dalam library System.Security.Cryptography di framework .NET 4.0 oleh Microsoft. RNG ini menggunakan algoritma proprietary yang telah teruji *cyrptographically secure*. RNG ini dipilih karena merupakan CPRNG standar untuk program-program dalam OS Windows.
- QRBG121, yaitu hardware QRNG yang dibuat oleh Mario Stipcevic, Rudjer Boskovic Institute. QRBG121 menghasilkan bilangan random menggunakan fenomena emisi foton dalam semikonduktor, dan pendeteksinya dengan efek photoelektrik. QRNG ini tersedia melalui internet service di url http://random.irb.hr/
- QRNG Berlin, yaitu service QRNG oleh Departemen Fisika Universitas Berlin yang dapat diakses pada url http://qrng.physik.hu-berlin.de/. QRNG ini menggunakan fenomena keacakan waktu kedatangan foton. QRNG ini masih dalam penelitian lanjut, dan kelebihan utama dari QRNG ini dalah kecepatan pembangkitannya yang jauh lebih cepat dari QRNG biasa[9].

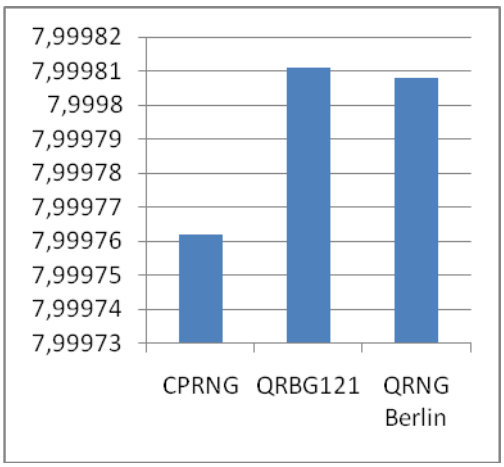
Sedangkan melakukan pengujian statistik digunakan program ENT oleh John Walker. Progam ini dapat mengukur nilai-nilai ukuran keacakan sekuens byte sebuah file *binary* berdasarkan teknik pengujian keacakan standar[2].

### *B. Hasil Pengujian dan Pembahasan*

Hasil pengujian adalah sebagai berikut, dikelompokkan berdasarkan jenis tes yang dilaksanakan.

**Entropy**

Adalah nilai kepadatan informasi suatu file, yang diekspresikan sebagai jumlah bit per karakter. Data yang acak akan memilki nilai entropi yang tinggi. Hasil pengujian adalah sebagai berikut:




*Gambar 6Entropi RNG (semakin besar lebih baik)*

Rentang entropi adalah dari nol hingga delapan, dan dari hasil tes terlihat bahwa semua RNG menghasilkan sekuens byte dengan entropi yang baik. Namun, dalam presisi hitungan yang tinggi, terlihat bahwa entropi data acak yang dihasilkan oleh sistem QNRG masih lebih tinggi dari CSPRNG.

**Chi-square Test**

Pengujian Chi-square adalah tes yang paling umum digunakan untuk menguji keacakan data, dan sangat sensitif pada adanya error dalam RNG. Distribusi Chi-square dikalkulasikan pada sekuens byte dan akan menghasilkan sebuah nilai absolute dan persentase seberapa sering sekuens bilangan acak sejati akan melewati nilai tersebut. Persentase ini diinpretasikan apakah data patut dicurigai keacakannya atau tidak. Sekuens yang menghasilkan persentase lebih dari 99% atau kurang dari 1% adalah hampir pasti tidak acak. Persentase antara 99%-95% dan 1%-5% menunjukan sekuens patut dicurigai tidak acak, dan persentase antara 90%-95% serta 5%-10% mengindikasikan bahwa sekuens ini 'hampir dicurigai'. Hasil pengujian adalah sebagai berikut:

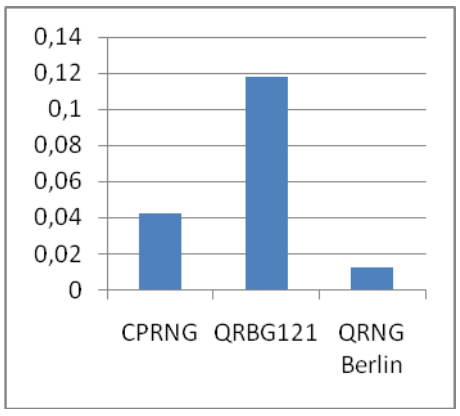
| RNG | Persentase Chi-square | Arti persentase |
|---|---|---|
| **CSPRNG** | 7,84% | Lolos, dengan catatan 'hampir dicurigai' |
| **QRBG121** | 18,38% | Lolos, dianggap acak. |
| **QRNG Berlin** | 29,95% | Lolos, dianggap acak. |

*Tabel 2 Hasil pengujian Chi-square test*

Sampel data acak dari CSPRNG standar Windows ternyata tidak lolos dengan sempurna dari tes chi-square, menghasilkan status 'hampir dicurigai'. Hal ini menunjukkan adanya kelemahan dalam CSPRNG, yang perlu diperbaiki. Sedangkan kedua QRNG yang dipakai lolos tanpa ada masalah.

**Arithmetic Mean**

Nilai ini dihasilkan dengan menjumlahkan semua byte dalam data dan membaginya dengan panjang data. Jika data mendekati random, nilainya akan mendekati 127.5. Kualitas dapat dilihat dari seberapa besar deviasi nilai terhadap 127.5 Hasil pengujian adalah sebagai berikut:

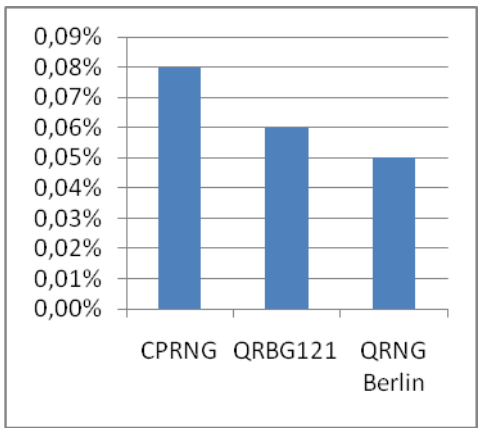



*Gambar 7Deviasi ArithmaticMean RNG (semakin kecil semakin baik)*

Semua RNG yang dicobakan membeikan deviasi yang kecil, menunjukkan bahwa nilai sekuens bytenya bersifat acak. Hal yang menarik adalah karena deviasi QRNG QBRG121 lebih besar dari CSPRNG, hal ini tetapi masih bisa dianggap normal karena variasi dari data yang acak.

**Monte Carlo Value for Pi**

Setiap sekuens sepanjang 6 byte digunakan sebagian kordinat X dan Y 24 bit dalam sebuah bujur sangkar. Apabila jarak dari titik yang dihasilkan secara random lebih kecil dari setengah panjang sisi bujursangkar, sekuens ini dinyatakan "kena". Persentase titik yang "kena" dapat digunakan untuk menghitung nilai π. Untuk data yang besar, nilai yang dihasilkan akan mendekati nilai π sebenarnya jika data tersebut bersifat acak.

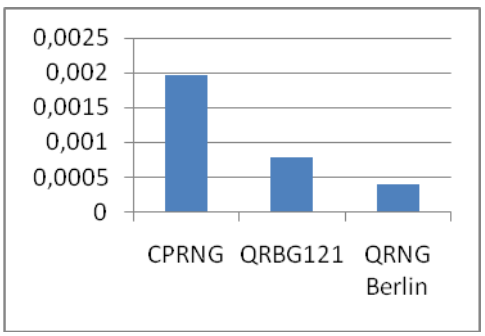



*Gambar 8Deviasi dari π (semakin kecil semakin baik)*

Semua RNG yang dicobakan memberikan nilai π yang akurat, semua deviasinya lebih kecil dari 0,1% dan menandakan data tersebut bersifat acak. Tetapi dapat terlihat bahwa RNG yang berbasis kuantum masih memberikan hasil yang lebih baik.

**Serial Correlation Coefficient**

Nilai ini mengukur seberapa besar pengaruh suatu byte pada byte setelahnya. Untuk data yang acak, nilai ini akan mendekati nol. Berikut adalah hasil pengujian, berupa nilai selisihnya dari nol:




*Gambar 9Deviasi dari nol (semakin kecil semakin baik)*

Semua RNG yang dicobakan memberikan nilai *Serial Correlation Coefficient* yang amat kecil, mengindikasikan data tersebut bersifat acak. Tetapi dapat terlihat bahwa

RNG yang berbasis kuantum masih memberikan hasil yang lebih baik.

**Perbandingan Efisiensi QNRG dengan CSPRNG**

Selain dari hasil pengujian statistik, pemilihan RNG juga perlu dipertimbangkan berdasarkan efisiensinya. Dalam makalah ini saya membandingkan QNRG dengan CSPRNG berdasarkan kecepatan pembangkitan bilangan acak dan biayanya.

- Efisiensi CSPRNG
  Dari hasil eksperimen penulis, algoritma CSPRNG standar yang tersedia di sistem operasi Windows dapat menghasilkan data acak dalam kecepatan sekitar 235 Mb/s di sebuah komputer dengan CPU Athlon 64 X2 dan memori 2GB, tanpa membutuhkan biaya tambahan.
- Efisiensi QRNG
  Produk QRNG komersil yang tersedia saat makalah ini dibuat dapat menghasilkan data acak dalam kecepatan antara 4 Mb/s hingga 16 Mb/s tergantung dari model produk, yang harganya antara 990 € hingga 2230 € [4].

Dapat dilihat bahwa dalam aspek efisiensi, untuk saat ini CSPRNG masih lebih baik dari QRNG.

## V. KESIMPULAN

Dari hasil pengujian statistik pada pembangkit bilangan acak berbasis fenomena fisika quantum dan pembangkit bilangan acak berbasis algoritma, keduanya memberikan hasil yang baik dan menunjukkan layak dipakai untuk aplikasi kriptografi.

Tetapi walaupun keduanya layak, hasil pengujian dalam presisi tinggi menunjukkan sampel data acak dari QNRG memberikan kualitas keacakan yang lebih baik dari CSPRNG dalam hampir semua tes uji yang dilaksanakan. Hal ini mengkonfirmasikan dasar teori fisika quantum yang menjadi dasar QRNG, yang menjanjikan QNRG adalah TRNG yang murni.

Sedangkan mengenai efisiensi QRNG sendiri, pada saat makalah ini ditulis masih jauh dibawah efisiensi dari CSPRNG yang biasa digunakan, hingga masih menyulitkan pengadopsian QRNG secara luas. Tetapi untuk kedepan dapat diekspektasi bahwa efisiensi QRNG akan terus meningkat, sesuai perkembangan teknologi. Penelitian terbaru telah melaporkan QRNG yang kecepatannya mencapai 150Mb/s, yang efisiensinya mulai menyaingi kecepatan CSPRNG biasa, walau QRNG ini masih dalam tahap penelitian dan belum menjadi produk komersil [9].

Dari hasil analisis, penulis merekomendasikan untuk memakai tetap CSPRNG untuk kebanyakan aplikasi kriptografi dimana faktor kecepatan adalah prioritas yang tinggi, disebabkan keuntungan dari efisiensi CSPRNG masih lebih besar dari kelebihan bilangan acak sejati oleh QRNG. Sedangkan QRNG adalah pilihan yang patut dipertimbangkan untuk kebutuhan kriptografi dimana kecepatan proses tidak kritikal dan dibutuhkan keamanan data setinggi mungkin.

## REFERENSI


[1] http://www.fourmilab.ch/hotbits/how3.html.
[2] http://www.fourmilab.ch/random/
[3] http://www.idquantique.com/images/stories/PDF/quantis-random-generator/quantis-whitepaper.pdf
[4] http://www.idquantique.com/ordering/shop.html
[5] http://www.random.org/randomness/
[6] http://www.informatika.org/~rinaldi/Kriptografi/2010-2011/Pembangkit%20Bilangan%20Acak.ppt
[7] http://www.cp.eng.chula.ac.th/~piak/teaching/dmath/random/test-random.ppt
[8] http://www.irb.hr/users/stipcevi/research/index.html.
[9] http://qrng.physik.hu-berlin.de/
[10] http://www.random.org/analysis/


## 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, 8 Mei 2010

Andika Pratama, NIM : 13507005