

Komputasi *Grid* Menggunakan *Globus* untuk Menghitung Opsi *Put* Amerika dengan Simulasi *Monte Carlo*

Aji Purwinarko¹ & Reza Pulungan²

¹ Jurusan Ilmu Komputer, FMIPA UNNES, Semarang

² Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta
Email: ¹ajienator@yahoo.com & ²pulungan@ugm.ac.id

Abstrak. *Internet* dan teknologi komputasi *grid* mengubah cara kita mengatasi masalah yang kompleks. Komputasi *grid* terus menjanjikan untuk memberikan kemampuan yang tinggi dari berbagai sistem dan teknik komputasi. Kemampuan mendistribusikan aplikasi pada beberapa mesin adalah salah satu aspek kunci dari komputasi *grid*. Salah satu penyedia librari komputasi *grid* adalah *Globus Toolkit*. Komputasi *grid* ini dapat dimanfaatkan untuk menjalankan aplikasi opsi *put* Amerika dengan menggunakan simulasi *Monte Carlo*. Simulasi *Monte Carlo* dapat meramalkan harga saham yang akan terjadi. Dari hasil penelitian yang dilakukan, menunjukkan bahwa semakin banyak simulasi yang dilakukan maka semakin akurat nilai rata-rata harga saham. Hasil penelitian ini menunjukkan bahwa semakin banyak simulasi yang dilakukan, akan menghasilkan nilai opsi *put* yang konvergen dengan *standard error* yang kecil dan proses komputasi dengan menggunakan jumlah prosesor yang besar akan lebih cepat.

Kata kunci: *Grid komputer; globus; mpi; monte carlo; opsi put Amerika.*

1. PENDAHULUAN

Di dalam bursa saham, dikenal adanya kontrak opsi sebagai sarana alternatif dalam berinvestasi. Kontrak opsi atau opsi didasarkan pada suatu perjanjian untuk membeli atau menjual saham pada suatu tingkat harga dan periode waktu tertentu. Opsi dapat digunakan untuk menentukan besarnya keuntungan yang ingin dicapai maupun meminimalkan tingkat kerugian. Penggunaan hak untuk menjual atau membeli saham dalam kontrak opsi dikatakan sebagai tindakan eksekusi.

Berdasarkan waktu eksekusinya, kontrak opsi dibedakan atas opsi Amerika yakni kontrak opsi yang dapat dieksekusi kapanpun antara tanggal pembelian sampai dengan tanggal jatuh tempo (*expiration date*) dan opsi Eropa yakni opsi yang hanya dapat dieksekusi pada saat tanggal jatuh tempo. Harga opsi Eropa dapat ditentukan dengan menggunakan formula *Black-Scholes* yang merupakan solusi analitik dari persamaan *Black-Scholes*, namun untuk opsi Amerika

belum terdapat solusi analitik, sehingga penelitian-penelitian yang selama ini dilakukan untuk menentukan harga opsi Amerika adalah [1, 2, 3]:

Bila ditinjau dari jenis hak yang diberikan kepada *option holder*, ada dua jenis opsi, yaitu: opsi *call* (*call option*) dan opsi *put* (*put option*). Opsi *call* memberi hak kepada *option holder* untuk membeli sejumlah tertentu *optioned stock* pada (atau sebelum) *expiration date* dan dengan *exercise price* tertentu. Di lain pihak, opsi *put* memberi hak kepada *option holder* untuk menjual sejumlah tertentu *optioned stock*, dengan *exercise price* dan pada (atau sebelum) *expiration date* tertentu. Metode perhitungan opsi *put* antara lain [1, 4]:

Semakin banyak simulasi yang dilakukan, maka akan didapatkan harga opsi *put* Amerika yang semakin konvergen, akan tetapi proses ini memerlukan *resource* yang besar dan mahal, sehingga perlu diupayakan adanya *resource* yang besar dan murah. *Resource* yang besar dan murah dapat diciptakan dengan memanfaatkan komputasi *parallel* ataupun *grid*. Dibandingkan dengan teknologi *parallel* tradisional, komputasi *grid* memiliki keuntungan yang lebih besar [5, 6, 7].

Berdasarkan uraian di atas, penentuan harga opsi *put* Amerika menggunakan metode *Monte Carlo* dengan memanfaatkan komputasi *grid* masih belum banyak dilakukan, sehingga dalam penelitian ini akan dieksplorasi penentuan harga opsi *put* Amerika dengan metode *Monte Carlo* menggunakan komputasi *grid* yang dibangun dengan *Globus Toolkit 4.2.1*.

2. METODE

Dalam menghitung opsi *put* Amerika, metode yang digunakan adalah metode *Monte Carlo*. Model simulasi *Monte Carlo* merupakan bentuk simulasi probalistik di mana solusi dari suatu masalah diberikan berdasarkan proses acak. Opsi *put* Amerika dapat ditentukan dengan membangkitkan harga saham yang terjadi disepanjang interval $[0, T]$ dan T menyatakan waktu jatuh tempo. Dengan mengasumsikan bahwa harga saham S mengikuti model gerak *Brown geometrik* sehingga memenuhi persamaan 1 sebagai berikut [1]:

$$dS(t) = rS(t)dt + \sigma S(t)dW(t) \quad (1)$$

dengan $S(t)$ adalah harga saham pada waktu t . $rS(t)$ merupakan nilai harapan dari perubahan $S(t)$. Parameter r menyatakan tingkat rata-rata pertumbuhan harga saham dan $rS(t)dt$ sendiri disebut sebagai komponen deterministik. Mengingat harga saham juga dipengaruhi oleh faktor ketidakpastian maka komponen stokastiknya adalah $S(t)dW(t)$, dengan menyatakan volatilitas harga saham. $dW(t)$ adalah peubah acak dengan *drift rate* 0 dan *variance rate* 1, yang

mana $W(t)$ (distribusi normal) merupakan proses stokastik yang mengikuti gerak *Brown*. Dengan demikian, perubahan harga yang terjadi di dalam saham tidak secara langsung dipengaruhi oleh $W(t)$, melainkan oleh $dW(t)$.

Nilai S_t pada persamaan (1) dapat diperoleh dengan cara menentukan harga saham awal S_0 dan membangkitkan secara acak faktor pengganggu (*Brownian noise*), sehingga didapatkan persamaan (2) sebagai berikut [8]:

$$S(t) = S_0 e \left(\left(r - \frac{\sigma^2}{2} \right) t + \sigma W(t) \right) = S_0 e \left(\left(r - \frac{\sigma^2}{2} \right) t + \sigma \sqrt{t} N(0,1) \right) \quad (2)$$

$N(0, 1)$ adalah standar variabel acak normal.

Payoff opsi *put American* untuk setiap $t \in [0, T)$ dapat ditentukan sebagai berikut:

$$\Lambda(S(t)) = \max[E - S(t), 0] \quad (3)$$

Sehingga, nilai opsi *put American* pada keadaan bebas resiko dapat diasumsikan sebagai berikut [8]:

$$P^{AM}(S_0, 0) = \sup_{0 \leq \tau \leq T} E[e^{-r\tau} \Lambda(S(\tau))] \quad (4)$$

τ adalah waktu penghentian (*stopping time*) dan $e^{-r\tau} \Lambda(S(\tau))$ merupakan nilai eksekusi (*payoff*) yang terdiskon dari opsi tersebut.

Hubungan antara $P^{Am}(S, t)$ dan $\Lambda(S(\tau))$ dapat dikelompokkan menjadi tiga kategori, yaitu:

a. Nilai $P^{Am}(S, t)$ memenuhi ketaksamaan:

$$P^{Am}(S, t) < \Lambda(S(t)) \quad (5)$$

Jika investor membeli kontrak opsi tersebut dengan harga $P^{Am}(S, t)$ dan kontrak opsi segera dieksekusi, maka investor akan memperoleh keuntungan bebas resiko sebesar $= E - S - P^{Am}(S, t)$. Hal ini berarti bahwa terdapat peluang terjadinya tindakan *arbitrase*, maka kemungkinan pertama tidak berlaku.

b. Nilai $P^{Am}(S, t)$ memenuhi ketaksamaan:

$$P^{Am}(S, t) = \Lambda(S(t)) \quad (6)$$

Maka akan terdapat dua reaksi investor, pertama adalah investor tidak tertarik membeli karena investasi yang impas dan yang kedua adalah investor tertarik untuk membeli opsi karena adanya harapan bahwa nilai pengembalian opsi (*return*) pada saat opsi dieksekusi akan meningkat. Untuk mengantisipasi kedua kemungkinan tersebut, maka investor pemegang kontrak opsi lebih memilih mengeksekusi opsinya. Dengan demikian, Persamaan (6) memberikan keadaan kepada investor untuk mengeksekusi kontrak opsi *put American*.

c. Nilai $P^{Am}(S, t)$ memenuhi ketaksamaan:

$$P^{Am}(S, t) > \Lambda(S(t)) \quad (7)$$

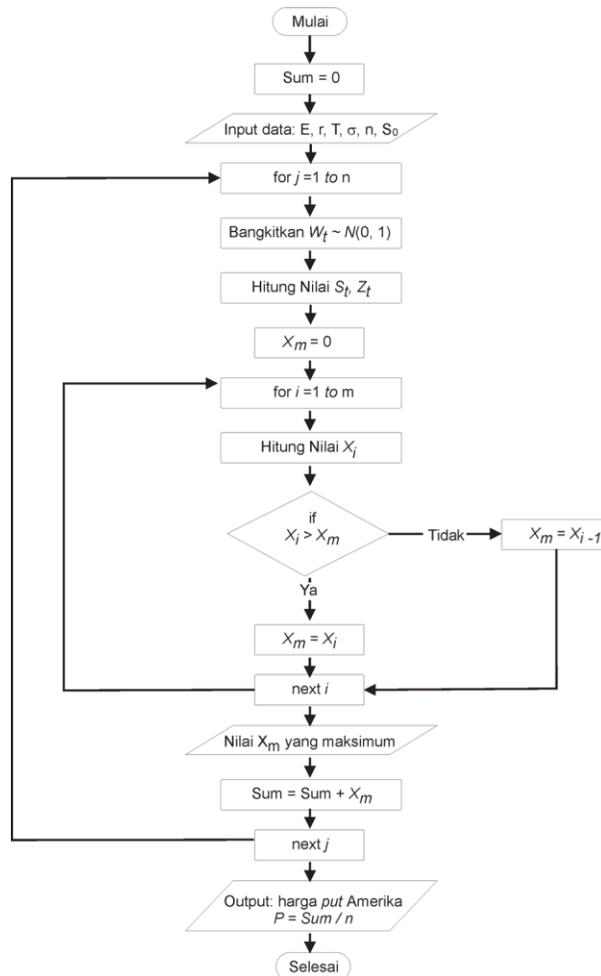
Hal ini berarti bahwa tindakan eksekusi opsi akan merugikan karena nilai keuntungan opsi lebih kecil dari nilai kontrak opsinya. Akibatnya investor pemegang kontrak opsi lebih memilih untuk menjual kontrak opsi dengan harga

$P^{Am}(S, t)$ kepada pihak lain. Dengan demikian, ketaksamaan (7) akan menghasilkan aksi jual kontrak opsi.

Dengan demikian, dapat disimpulkan bahwa nilai opsi *put* Amerika harus memenuhi persamaan (8).

$$P^{Am}(S, t) \geq \Lambda(S(t)) \quad (8)$$

Flowchart dari simulasi *Monte Carlo* dapat diperlihatkan melalui Gambar 1.



Gambar 1 Flowchart of Monte Carlo simulation.

Legend:

N: banyak dari hari eksekusi yang mungkin
 r: tingkat bunga bebas resiko satu periode

i: indeks hari

S_0 : harga saham awal

n: banyak path sampel

S_t : harga saham pada waktu t

m: indeks sampel

$\Lambda(S(t)) = zt$: nilai intrinsik (payoff) opsi pada waktu t

E: harga eksekusi

σ : volatilitas saham

T: Waktu

X_m : nilai rata-rata dari setiap X maksimum

2.1 Kebutuhan Masukan

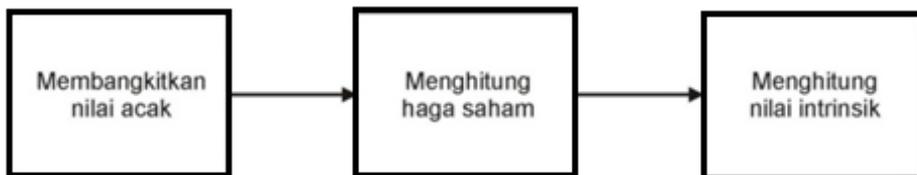
Masukan data yang terdiri dari 7 variabel, yaitu banyaknya hari yang mungkin untuk melakukan eksekusi (N), jumlah *path* sampel (n), harga eksekusi (E), horizon waktu (T), tingkat bunga bebas resiko satu periode (r), harga saham awal (S_0), *vola-tilitas* saham (σ).

2.2 Kebutuhan Proses

Hasil akhir dari sistem ini adalah harga opsi terbaik. Namun sebelumnya, terdapat langkah-langkah untuk mendapatkan opsi tersebut, yaitu:

1. Proses membangkitkan bilangan acak
Proses membangkitkan bilangan acak. Bilangan acak ini akan digunakan untuk membangkitkan *Brownian noise*.
2. Proses menghitung harga saham *gayut waktu* S_t
Harga saham S_t dipengaruhi oleh hasil perhitungan *Brownian noise* setiap waktu (Wt). S_t juga ditentukan oleh S_{t-1} , proses ini merupakan bentuk rantai *Markoff*.
3. Proses menghitung nilai intrinsik Z_t (*payoff*)
Nilai intrinsik ini didapatkan dengan menghitung selisih antara nilai eksekusi dengan harga saham setiap waktu (S_t).

Pada Gambar 2 menunjukkan tahapan-tahapan proses yang harus dilakukan, yang terdiri dari membangkitkan nilai acak, menghitung harga saham setiap waktu dan menghitung nilai intrinsik setiap waktu.



Gambar 2 Tahapan proses.

2.3 Kebutuhan Keluaran

Setiap proses yang telah dilakukan akan menghasilkan keluaran masing-masing:

1. Proses membangkitkan bilangan acak
Proses ini akan menghasilkan menghasilkan bilangan acak.
2. Proses menghitung harga saham *gayut waktu* S_t
Proses ini akan menghasilkan harga saham setiap waktu (S_t).
3. Proses menghitung nilai intrinsik Z_t (*payoff*)

Proses ini akan menghasilkan *payoff*, yang mana *payoff* ini adalah hasil akhir dari sistem.

3. HASIL DAN PEMBAHASAN

Berdasarkan metode yang dipaparkan di atas, setiap tugas yang diberikan oleh *user* akan diterima oleh *server*. *Server* akan mendistribusikan setiap perhitungan kepada masing-masing *node*. Setiap hasil perhitungan yang telah dilakukan oleh masing-masing *node* dikembalikan kepada *server*. Sehingga, *server* disini mempunyai peran menerima tugas yang telah diberikan oleh *user*, kemudian membagi tugas tersebut kepada setiap *node* yang terhubung dengan *server*, selanjutnya mengumpulkan kembali hasil perhitungan yang telah dilakukan oleh *user* dan pada akhirnya mengirimkan hasil perhitungan tersebut kepada *user*. Sehingga didapatkan hasil sebagai mana berikut:

3.1 Membandingkan Hasil Referensi dengan Hasil Percobaan Menggunakan Bilangan Acak 154.748

Simulasi percobaan pertama ini menggunakan prosesor tetap 12 dengan $rate=0.0488$, $\sigma= \{0.2, 0.4\}$, $T= \{0.0833, 0.5833\}$, $stock= 40$, $strike= \{35, 40, 45\}$, $M= 40.000$ dan $d= 14$ (jumlah bilangan acak 154.748). Perbandingan hasil referensi dan percobaan dapat dilihat pada tabel 1.

Tabel 1 Percobaan 1.

No	K	σ	T	Referensi	Percobaan	SD	SE	Waktu (s)
1	35	0,2	0,0833	0,0062	0,0377	1,08	0,01	8,53
2	35	0,2	0,5833	0,4328	0,4124	1,62	0,01	8,73
3	40	0,2	0,0833	0,8522	0,858	1,91	0,01	8,39
4	40	0,2	0,5833	1,9904	1,8705	3,54	0,02	8,49
5	45	0,2	0,0833	5	4,9057	5,51	0,03	8,33
6	45	0,2	0,5833	5,267	4,8811	6,65	0,03	8,54
7	35	0,4	0,0833	0,2466	0,2527	1,35	0,01	8,63
8	35	0,4	0,5833	2,1549	2,0666	4,33	0,02	8,96
9	40	0,4	0,0833	1,7681	1,7608	3,19	0,02	8,66
10	40	0,4	0,5833	4,3526	4,2225	2,2	0,01	8,51
11	45	0,4	0,0833	5,2868	5,2963	6,63	0,03	8,62
12	45	0,4	0,5833	7,383	7,1935	10,17	0,05	8,95

Dari hasil percobaan 1 di atas, dapat diketahui bahwa nilai opsi *put* yang dihasilkan pada kolom percobaan dibandingkan dengan kolom Referensi terdapat selisih [4]. Selisih masing-masing perhitungan disebabkan karena

perbedaan *Brownian noise*. Sementara, *Brownian noise* itu sendiri dipengaruhi oleh bilangan acak yang dibangkitkan oleh *random generator* yang digunakan. *Standard error* (SE) yang didapatkan dari setiap percobaan berbeda-beda dan rata-rata di bawah 5%. Dengan nilai SE yang di bawah 5% menunjukkan bahwa opsi *put* yang di hasilkan dalam simulasi ini masih relevan.

Berdasarkan percobaan 1, semakin besar volatilitas harga saham yang diberikan, akan memberi dampak pada peningkatan nilai opsi *put*. Sehingga, untuk waktu eksekusi yang sama, volatilitas yang besar akan menghasilkan opsi *put* yang semakin besar. Hal ini akan berdampak pada selisih antara P^{AM} dengan $\Lambda(S(t))$ yang akan semakin kecil.

Berikut adalah contoh hasil perbandingan opsi *put* dengan *payoff*, untuk nilai K dan yang sama dengan nilai T berbeda.

- 1) Nilai P^{AM} pada percobaan 1 lebih kecil dibandingkan dengan simulasi 2, sehingga selisih antara P^{AM} dengan $\Lambda(S(t))$ pada simulasi 1 lebih besar dibandingkan dengan simulasi 2. Hal ini menunjukkan bahwa simulasi 1 akan lebih menguntungkan dibandingkan dengan simulasi 2.
- 2) Nilai P^{AM} pada percobaan 3 lebih kecil dibandingkan dengan simulasi 4, sehingga selisih antara P^{AM} dengan $\Lambda(S(t))$ pada simulasi 3 lebih besar dibandingkan dengan simulasi 4. Hal ini menunjukkan bahwa simulasi 3 akan lebih menguntungkan dibandingkan dengan simulasi 4.

Berikut adalah contoh hasil perbandingan opsi *put* dengan *payoff*, untuk nilai K dan T yang sama dengan nilai berbeda.

- 1) Nilai P^{AM} pada simulasi 1 lebih kecil dibandingkan dengan simulasi 7, sehingga selisih antara P^{AM} dengan $\Lambda(S(t))$ pada simulasi 1 lebih besar dibandingkan dengan simulasi 7. Hal ini menunjukkan bahwa simulasi 1 akan lebih menguntungkan dibandingkan dengan simulasi 7.
- 2) Nilai P^{AM} pada simulasi 2 lebih kecil dibandingkan dengan simulasi 8, sehingga selisih antara P^{AM} dengan $\Lambda(S(t))$ pada simulasi 2 lebih besar dibandingkan dengan simulasi 8. Hal ini menunjukkan bahwa simulasi 2 akan lebih menguntungkan dibandingkan dengan simulasi 8.

3.2 Membandingkan Hasil Referensi dengan Hasil Percobaan Menggunakan Bilangan Acak 1.004.748

Simulasi percobaan kedua ini menggunakan prosesor 12 dengan rate = 0.0488, sigma = 0.2, 0.4, T = 0.0833, 0.5833, stock = 40, strike = 35, 40, 45, M = 5.000 dan d = 14 (jumlah bilangan acak 1.004.748). Perbandingan hasil referensi dan percobaan dapat dilihat pada tabel 2.

Tabel 2 Percobaan 2.

No	K	σ	T	Referensi	Percobaan	SD	SE	Waktu (s)
1	35	0,2	0,0833	0,0062	0,0321	0,99	0,01	1,41
2	35	0,2	0,5833	0,4328	0,4126	1,57	0,02	1,35
3	40	0,2	0,0833	0,8522	0,8574	1,85	0,03	1,43
4	40	0,2	0,5833	1,9904	1,8806	3,53	0,05	1,3
5	45	0,2	0,0833	5	4,8849	5,47	0,08	1,32
6	45	0,2	0,5833	5,267	4,8492	6,63	0,09	1,37
7	35	0,4	0,0833	0,2466	0,2501	1,28	0,02	1,45
8	35	0,4	0,5833	2,1549	2,097	4,34	0,06	1,31
9	40	0,4	0,0833	1,7681	1,7617	3,17	0,04	1,37
10	40	0,4	0,5833	4,3526	4,2259	6,99	0,1	1,22
11	45	0,4	0,0833	5,2868	5,2611	6,6	0,09	1,26
12	45	0,4	0,5833	7,383	7,1564	10,18	0,14	1,48

Berbeda halnya antara percobaan 1 dan percobaan 2, pada percobaan 2 ini menggunakan bilangan acak yang lebih besar (1.004.748) dan perulangan lebih sedikit (5000), dibandingkan dengan percobaan 1 yang menggunakan bilangan acak dalam jumlah kecil (154.748) dan perulangan lebih banyak (40.000). *Standard error* (SE) yang dihasilkan pada percobaan 2 lebih besar dibandingkan dengan percobaan 1, nilai SE yang dihasilkan pada percobaan 2 berkisar antara 2-14 %. Hal ini menunjukkan juga bahwa besarnya perulangan dan jumlah bilangan acak yang digunakan mempengaruhi hasil yang di dapat.

Berdasarkan hukum bilangan besar, maka dari hasil percobaan 1 dapat disimpulkan bahwa banyaknya rata-rata sampel sesuai dengan banyaknya simulasi yang dilakukan dan rata-rata yang konvergen pada nilai tengah akan selalu berlaku meskipun banyaknya simulasi ditambah sampai menuju tak hingga. Dengan demikian hukum bilangan besar merupakan penduga dari rata-rata populasi yang jumlahnya sangat besar.

3.3 Membandingkan Waktu Eksekusi Opsi Put untuk Jumlah Prosesor yang Berbeda

Simulasi percobaan ketiga ini menggunakan prosesor yang berjumlah 3, 5, 7, 9 dan 11 dengan jumlah bilangan acak 154.748. Perbandingan waktu eksekusi opsi put untuk jumlah prosesor yang berbeda dapat dilihat pada tabel 3.

Tabel 3 Percobaan 3.

No	K	Σ	T	PUT	Waktu Eksekusi				
					n = 3	n = 5	n = 7	n = 9	n = 11
1	35	0,2	0,0833	0,0377	20,50	15,90	10,92	9,57	8,11
2	35	0,2	0,5833	0,4124	19,90	15,78	10,52	9,86	8,54
3	40	0,2	0,0833	0,858	20,30	16,72	11,35	9,63	8,56
4	40	0,2	0,5833	1,8705	19,85	15,63	10,39	9,44	8,32
5	45	0,2	0,0833	4,9057	19,93	15,66	11,31	9,97	8,55
6	45	0,2	0,5833	4,8811	19,20	14,80	11,16	10,32	8,50
7	35	0,4	0,0833	0,2527	20,10	15,66	10,78	9,37	8,15
8	35	0,4	0,5833	2,0666	19,98	16,10	11,32	9,51	8,04
9	40	0,4	0,0833	1,7608	20,00	16,14	11,17	9,59	8,13
10	40	0,4	0,5833	4,2225	19,30	15,66	11,42	9,35	8,17
11	45	0,4	0,0833	5,2963	19,80	16,29	11,31	9,69	8,13
12	45	0,4	0,5833	7,1935	19,00	15,75	11,28	9,72	8,85

Dari percobaan 3 diketahui waktu komputasi dengan menggunakan prosesor 3, 5, 7, 9 dan 11. Waktu yang diperlukan untuk melakukan komputasi dengan menggunakan jumlah prosesor yang semakin besar akan semakin kecil dibandingkan proses komputasi dengan menggunakan prosesor sebanyak 3. Selain jumlah prosesor dan banyaknya simulasi, kecepatan ini juga dipengaruhi oleh komunikasi antar komputer. Komunikasi disini adalah komunikasi internet yang menghubungkan *grid* tersebut. Sebagaimana kita ketahui bahwa komunikasi internet ini sangat dipengaruhi oleh *bandwidth*, semakin besar *bandwidth* maka hambatan akan semakin kecil dan waktu tunda akan semakin kecil.

4. SIMPULAN

Berdasarkan uraian diatas dapat disimpulkan sebagai berikut:

1. Pemilihan jumlah bilangan acak yang digunakan akan mempengaruhi nilai opsi *put*, semakin banyak bilangan acak yang digunakan, akan memerlukan waktu yang semakin besar. Namun, untuk mendapatkan nilai opsi *put* yang sepadan, jumlah simulasi yang dibutuhkan akan lebih sedikit.
2. Semakin banyak simulasi yang dilakukan, maka nilai opsi *put* yang didapatkan akan semakin konvergen. Selain itu, akan di dapatkan *standard error* yang kecil.

3. Nilai ekstrinsik akan mempengaruhi besarnya opsi put. Untuk nilai ekstrinsik di bawah atau sama dengan nilai *stock*, akan didapatkan nilai opsi *put* yang lebih kecil dibandingkan dengan nilai *payoff*. Sehingga, investor akan mendapatkan keuntungan jika segera mengeksekusi opsi tersebut.
4. Nilai volatilitas harga saham akan mempengaruhi besarnya opsi *put*. Untuk waktu eksekusi yang sama, nilai volatilitas harga saham yang lebih kecil akan lebih menguntungkan, karena akan didapatkan nilai opsi put yang lebih kecil.
5. Jumlah prosesor mempengaruhi proses komputasi, dengan menggunakan jumlah prosesor yang lebih besar, maka waktu yang diperlukan untuk menghitung sejumlah sampel yang sama akan semakin kecil. Sedangkan nilai opsi *put* tidak dipengaruhi oleh jumlah prosesor yang terlibat dalam komputasi.

Banyaknya simulasi yang dilakukan mempengaruhi waktu komputasi, semakin banyak simulasi dan jumlah prosesor tetap, maka waktu yang diperlukan untuk melakukan proses komputasi akan semakin besar. Sedangkan nilai opsi *put* dipengaruhi oleh banyaknya simulasi yang dilakukan. Semakin banyak simulasi yang dilakukan, nilai rata-rata opsi *put* akan bersifat konvergen. Berikut ini merupakan saran penulis kepada peneliti yang akan mengembangkan topik ini pada penelitian mendatang.

1. Sumber daya komputer.

Salah satu tantangan yang harus dihadapi penulis adalah minimnya sumber daya komputer yang tersedia, sehingga kurang memperlihatkan heterogenitas dalam arsitektur, sistem yang digunakan di dalam *grid*.

2. Perancangan antarmuka.

Penulis dalam penelitian ini menggunakan *command line interface* untuk menjalankan aplikasi, sehingga jika memungkinkan peneliti berikutnya dapat mengembangkan *interface* dan menjalankannya di dalam bahasa pemrograman java, delphi dsb.

REFERENSI

- [1] Syazali, M., *Penentuan Harga Opsi Put Amerika dengan Simulasi Monte Carlo*, Tesis, Program Studi Matematika Terapan Pascasarjana, IPB, Bogor, 2011.
- [2] Han, G., Kim, B. & Lee., *Kernel-based Monte Carlo simulation for American option pricing*, *Expert Systems with Applications*, Elsevier, Netherlands, 2009.
- [3] Cvetanoska, V. & Stojanovski, T., *Using High Performance Computing and Montecarlo Simulation For Pricing American Options*,

<http://arxiv.org/ftp/arxiv/papers/1205/1205.0106.pdf>, 1 May 2012, diakses 28 September 2012.

- [4] Huang, J., Subrahmanyam, M. G. & Yu, G. G., Pricing and Hedging American Options: A Recursive Integration Method, *The Review of Financial Studies*, Volume 9, No. 1, pp 277-300, Oxford University, 1996.
- [5] Xu, H. & Wu, G., *Parallel programming in Grid: Using MPI*, Proceedings of the Third International Symposium on Electronic Commerce and Security Workshops, Academy Publisher, Guangzhou, 2010.
- [6] Pahlevi, S. M., *Komputasi Grid dan Paralel, Risalah Lokakarya Komputasi dalam Sains dan Teknologi Nuklir*, Badan Tenaga Nuklir Nasional, Jakarta, 2008.
- [7] Doan, V. D., *Grid computing for Monte Carlo based intensive calculations in financial derivative pricing applications*, Dissertation, University of Nice, Nice, 2010.
- [8] Higham, D. J., *An Introduction to Financial Option Valuation*, Cambridge University Press, Cambridge, 2004.