

BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini akan dijelaskan hasil analisis dan perancangan aplikasi yang dilakukan selama pengerjaan tugas akhir.

3.1 Analisis

Bab ini akan memaparkan hasil analisis yang dilakukan selama pengerjaan Tugas Akhir, diawali dengan melakukan analisis perhitungan harga opsi dengan menggunakan metode monte carlo kemudian dilanjutkan dengan melakukan analisis terhadap kebutuhan perangkat lunak.

3.1.1 Analisis Perhitungan Harga Opsi dengan Menggunakan Metode Simulasi Monte Carlo

Metode simulasi Monte Carlo dalam menentukan harga opsi secara garis besar terdiri dari dua buah proses. Pertama, proses pembangkitan harga saham. Kedua, perhitungan harga opsi yang dihitung dari harga saham yang dibangkitkan dari proses pertama.

Proses pertama ialah proses pembangkitan harga saham. Proses pembangkitan harga saham ini dilakukan dengan memasukkan beberapa parameter yang dibutuhkan dalam membangkitkan harga saham di masa depan. Pembangkitan harga saham ini dilakukan dengan asumsi bahwa pergerakan harga saham mengikuti gerakan Brownian motion yang sesuai dengan model Blacks-Scholes. Fungsi tersebut kemudian diturunkan menjadi fungsi stokastik harga saham yakni nilai parameter yang dimasukan diadaptasi dari fakta yang terjadi pada dunia nyata ataupun berupa nilai asumsi yang diinginkan.

Dengan memasukkan variabel – variabel pada persamaan 2.13 pada subbab 2.1.3 kita akan mensimulasikan harga saham tersebut di masa yang akan datang sesuai dengan kondisi yang diinginkan (*risk free rate*, volatilitas, serta jangka waktunya kadaluarsanya). Harga saham hasil simulasi ini bergantung pada hasil pembangkitan bilangan acak. Untuk setiap hasil simulasi, selanjutnya harga saham di masa mendatang itu diperiksa apakah harga saham tersebut lebih besar dari harga tebus atau lebih kecil dari harga tebus.

Jika harga saham tersebut ternyata lebih tinggi dari pada harga tebus berarti harga saham tersebut dijadikan acuan untuk menghitung harga *call option* sebab *call option* hanya mempunyai nilai jika harga saham di masa mendatang lebih tinggi dari harga tebus. Selanjutnya hitung deltaCall yakni delta antara harga saham tersebut dengan harga tebusnya. DeltaCall tersebut merupakan harga *call option* pada waktu T.

Sebaliknya jika harga saham tersebut ternyata lebih kecil dari harga tebus maka harga saham tersebut akan dijadikan acuan untuk menghitung harga *put option* sebab *put option* hanya bernilai jika harga saham di masa mendatang lebih rendah dari harga tebusnya. Selanjutnya hitung harga deltaPut yakni delta antara harga saham tersebut dengan harga tebusnya. DeltaPut tersebut merupakan harga *put option* pada waktu T.

Untuk mendapatkan harga opsi yang lebih akurat simulasi ini dilakukan berulang - ulang. Harga *call option* dihitung dengan cara menjumlahkan seluruh deltaCall yang ada kemudian di rata - rata dengan banyaknya simulasi yang dilakukan. Begitu juga dengan harga *put option*. Seluruh deltaPut dari tiap simulasi dijumlahkan kemudian di bagi dengan jumlah simulasi.

Adapun proses simulasi diatas dapat dijabarkan pada tabel III-1 :

Tabel III - 1 Algoritma Simulasi

```

for I = 1 to NSimulation
    Saham = ProsesPembangkitanHargaSaham()
    If (Saham > HargaExercise)
        Delta = Saham - HargaExercise
        Beli = Beli + Delta
    Else If (HargaExercise > Saham)
        Delta = HargaExercise - Saham
        Jual = Jual + Delta
    End if
End for
OpsiBeli = Beli / NSimulation
OpsiJual = Jual / NSimulation

```

3.1.2 Deskripsi Perangkat Lunak

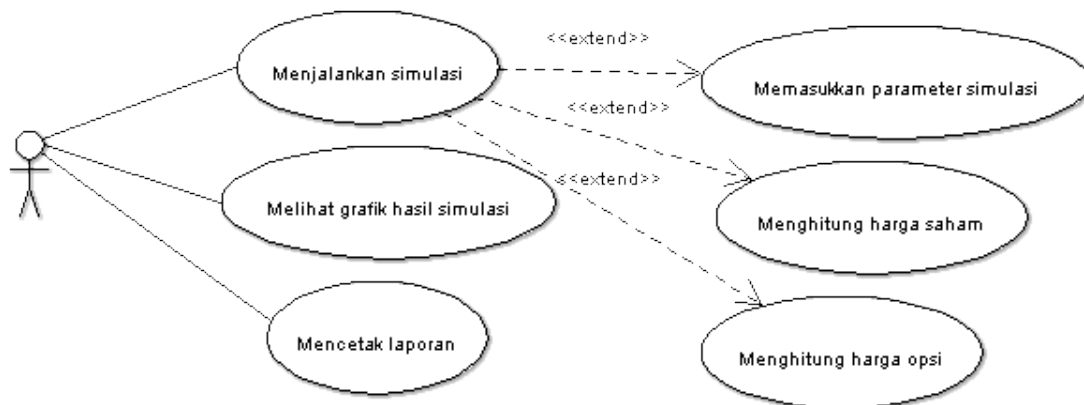
Perangkat lunak yang dikembangkan selanjutnya akan disebut dengan MonteCarlo OpS (*Monte Carlo OPtion Simulator*). MonteCarlo OpS merupakan suatu program untuk menghitung harga opsi suatu saham dengan menggunakan simulasi Monte Carlo. Pengguna MonteCarlo OpS setidaknya paham mengenai dasar teori opsi. Pengguna

dari MonteCarlo OpS tidak dikategorisasikan (administrator, operator, dsb) sehingga tidak dibutuhkan login untuk menggunakannya. Parameter - parameter yang dibutuhkan dari MonteCarlo OpS dimasukkan langsung oleh pengguna dan hasilnya dihitung pada saat dijalankan sehingga MonteCarlo OpS tidak menggunakan basis data dalam penggunaannya.

3.1.3 Kebutuhan Fungsionalitas

Fungsionalitas perangkat lunak MonteCarlo OpS akan tampak pada gambar III-1. Adapun perangkat lunak MonteCarlo OpS akan mengimplementasikan hal - hal berikut :

1. Memasukkan parameter dari pengguna
2. Melakukan simulasi perhitungan harga saham
3. Menampilkan grafik harga saham hasil simulasi
4. Menghitung harga opsi (*put option* dan *call option*)
5. Mencetak laporan hasil perhitungan



Gambar III - 1 Use Case Diagram

Secara rinci, kebutuhan fungsional MonteCarlo OpS dijabarkan pada tabel III-2 :

Tabel III - 2 Kebutuhan Fungsional

Fungsi :	Memasukkan Parameter dari Pengguna
Masukan	Parameter – parameter yang menjadi dasar perhitungan harga saham, yaitu : <i>spot price</i> , <i>strike price</i> , <i>volatility</i> , <i>risk free rate</i> , waktu

Fungsi :	Memasukkan Parameter dari Pengguna
Proses	Melakukan pengesetan pada variabel – variabel kelas MonteCarlo
Keluaran	Kelas Monte Carlo terinisiasi
Fungsi :	Melakukan simulasi perhitungan harga saham
Masukan	Kelas Monte Carlo terinisiasi
Proses	Memproses semua input dengan menggunakan simulasi Monte Carlo untuk menghitung harga saham
Keluaran	Harga saham hasil simulasi
Fungsi :	Menampilkan grafik harga saham hasil simulasi
Masukan	Data - data harga saham hasil simulasi, banyaknya percobaan
Proses	Menampilkan data harga saham hasil simulasi pada grafik
Keluaran	Grafik simulasi
Fungsi :	Menghitung harga opsi
Masukan	Harga saham hasil simulasi serta strike <i>price</i>
Proses	Perhitungan <i>moneyness</i> harga saham hasil simulasi untuk menghitung harga opsi
Keluaran	Harga opsi
Fungsi :	Mencetak laporan hasil perhitungan
Masukan	Hasil perhitungan harga opsi
Proses	Mencetak informasi yang dihasilkan dari perhitungan harga opsi
Keluaran	Laporan harga opsi

3.1.4 Kebutuhan Non Fungsionalitas

Kebutuhan non fungsional menjelaskan fungsi - fungsi pendukung perangkat lunak seperti yang tampak pada tabel III-3 :

Tabel III - 3 Kebutuhan Non-Fungsional

No.	Kebutuhan Non-Fungsional
1.	Antarmuka berbasis window (GUI)
2.	Mudah digunakan (<i>user friendly</i>)
3.	Sistem Operasi : <i>Windows Based</i>

3.1.5 Kebutuhan Data

Masukkan yang dibutuhkan MonteCarlo OpS ialah parameter - parameter untuk menghitung harga saham di masa yang akan datang yakni :

1. Harga saham sekarang (*spot price*).
2. Volatilitas saham
3. Tingkat suku bunga bebas resiko
4. Jangka waktu kadaluarsa.
5. Harga tebus (*Strike price*).

Variabel - variabel inilah yang digunakan untuk menghitung harga opsi. Seluruh data yang dibutuhkan dimasukkan secara langsung oleh pengguna saat menjalankan MonteCarlo OpS.

3.2 Perancangan

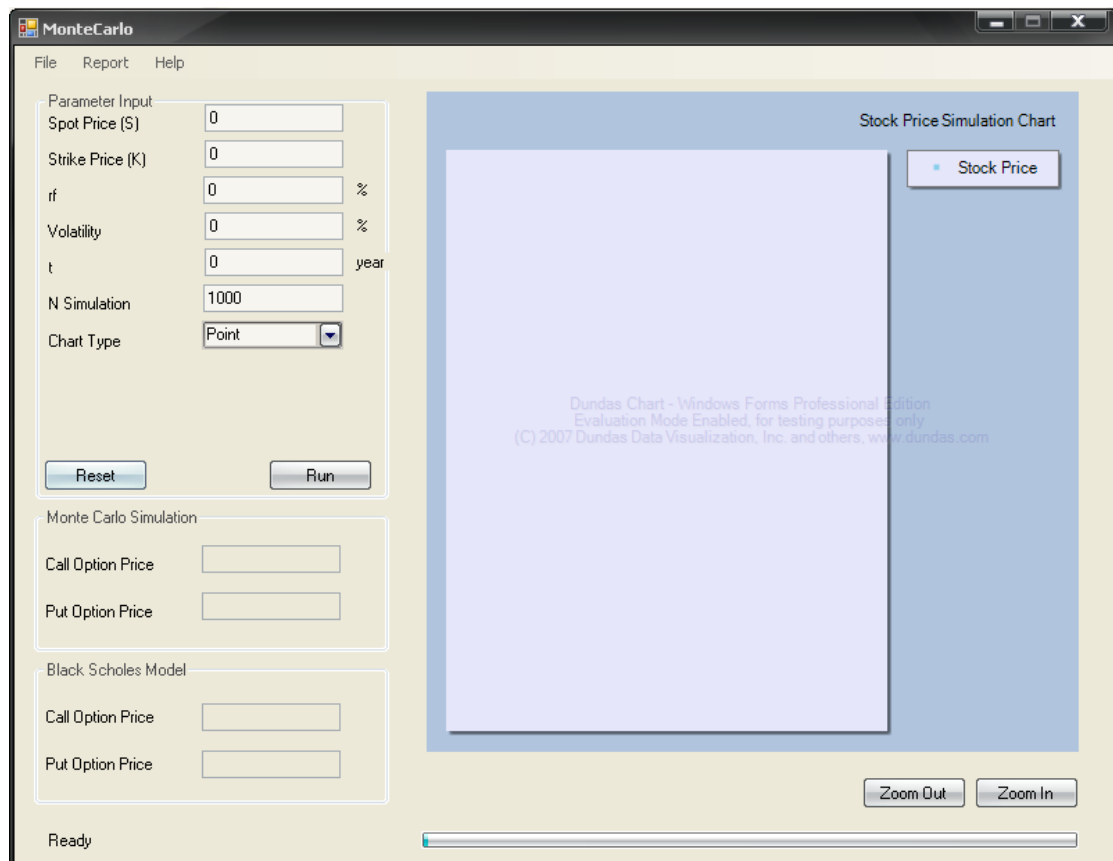
Hasil analisis pada subbab sebelumnya dijadikan dasar untuk merancang perangkat lunak ini. Pada tahap ini akan didefinisikan rancangan antarmuka serta batasan – batasan dalam melakukan perancangan.

3.2.1 Batasan Perancangan

Perangkat lunak MonteCarlo OpS tidak akan menggunakan suatu basis data tertentu sebab data - data yang dibutuhkan untuk melakukan simulasi dimasukkan langsung oleh pengguna.

3.2.2 Perancangan Antarmuka

Perangkat lunak MonteCarlo OpS yang akan dibuat akan mempunyai layar antarmuka sebagai berikut :



Gambar III - 2 Rancangan Tampilan Antarmuka

Menu utama di atas akan menampilkan pilihan sebagai berikut :

1. *File*

Menu ini berisi submenu yang berkaitan dengan program utama seperti keluar dari program.

2. *Report*

Menu ini akan berisi pilihan yang berhubungan dengan pembuatan laporan. Laporan ini terdiri dari dua jenis :

1. Laporan hasil simulasi yang akan berisi hasil perhitungan yang dilakukan perangkat lunak berupa harga opsi (yang dihitung dengan simulasi maupun yang dihitung dengan formula Black-Scholes).
2. Laporan grafik hasil simulasi. Laporan yang disajikan berupa sebuah diagram yang memperlihatkan sebaran harga saham yang dibangkitkan oleh proses simulasi.

3. *Help*

Menu ini akan berisi sub menu tentang program.

Secara umum antarmuka program utama dibagi menjadi dua bagian utama :

1. Sub Panel Kiri

Pada Sub Panel kiri terdiri dari dua bagian utama : bagian parameter input dan bagian output. Pada bagian parameter input pengguna dapat memasukkan parameter - parameter yang dibutuhkan untuk perhitungan simulasi. Selain itu dibagian ini akan disediakan tombol untuk melakukan simulasi serta tombol reset untuk mereset seluruh parameter kembali menjadi 0. Selanjutnya, di bagian bawah sub panel kiri terdapat dua buah sub panel yang berisi hasil perhitungan program. Hasil perhitungan terdiri dari dua jenis yakni hasil simulasi serta hasil perhitungan dengan menggunakan formula Black-Scholes. Dua buah sub panel ini ditaruh berdampingan agar pengguna dapat melihat langsung perbedaan antara hasil simulasi dengan hasil perhitungan Black-Scholes (tingkat keakuratan hasil simulasi).

2. Sub Panel Kanan

Pada bagian Sub Panel Kanan akan berisi grafik hasil simulasi pembangkitan saham yang dapat ditampilkan dalam bentuk kumpulan titik ataupun dalam bentuk garis serta terdapat *progress bar* yang menunjukkan proses berlangsungnya simulasi.

3.2.3 Perancangan Diagram Kelas

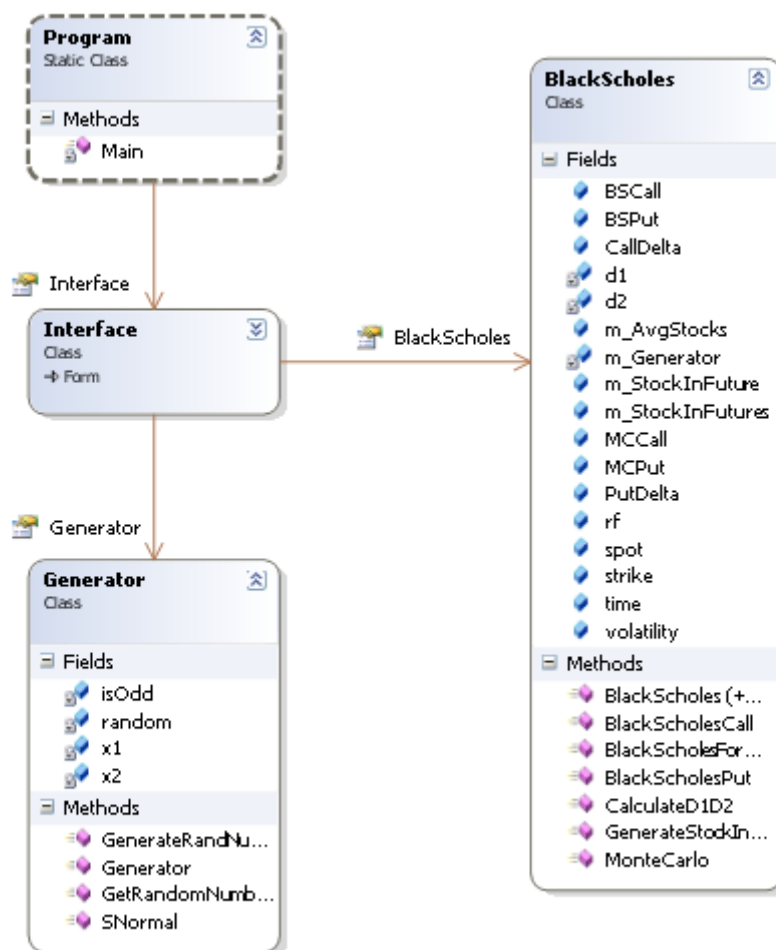
Perincian dari diagram kelas diatas dapat dilihat pada tabel III-4:

Tabel III - 4 Perincian Kelas MonteCarlo OpS

No.	Nama Kelas	Definisi	Operasi
1.	Program	Program utama untuk menjalankan program	Main()
2.	Interface	Antarmuka program utama	InitializeComponent() ButtonResetClicked() ButtonHitungClicked() Show() Exit()
3.	Generator	Kelas pembangkit bilangan acak	Generator() GenerateRandNumbNormDist() SNormal()

No.	Nama Kelas	Definisi	Operasi
3.	Generator	Kelas pembangkit bilangan acak	GetRandomNumber()
4.	BlackScholes	Kelas yang digunakan untuk melakukan simulasi montecarlo serta untuk melakukan perhitungan formula blackscholes	BlackScholes() CalculateD1D2 BlackScholesFormula() BlackScholesCall() BlackScholesPut() MonteCarlo(n); GenerateStockInTheFutures()

Adapun perancangan diagram kelas untuk perangkat lunak ini tampak pada gambar III – 3.



Gambar III - 3 Kelas Diagram Fase Perancangan

3.2.4 Perancangan Struktur Data

Struktur data yang digunakan untuk menyimpan parameter – parameter input untuk melakukan simulasi beserta hasil simulasi dapat dilihat pada tabel III-5.

Tabel III - 5 Perancangan Struktur Data

```
class BlackScholes
{
    double spot;
    double strike;
    double rf;
    double volatility;
    double time;

    double d1;
    double d2;

    double[] m_StockInFutures;

    double BSCall = 0;
    double BSPut = 0;

    double MCCall = 0;
    double MCPut = 0;
}
```