MA3261-Pengantar Matematika Keuangan

Laporan Hasil Analisis Saham dan Opsi
Intel Corporation (INTC)

Oleh

Faza Alisha Ramadina 10820028
Pamella Cathryn 10820033

PROGRAM STUDI SARJANA AKTUARIA
INSTITUT TEKNOLOGI BANDUNG

APRIL 2023

Daftar Isi

Halaman

1 Pendahuluan

1.1 Informasi Perusahaan Intel Corporation (INTC)

Intel adalah perusahaan multinasional yang bergerak dalam bidang teknologi dan berpusat di California, Amerika Serikat. Intel merupakan salah satu produsen *chip* untuk komputer yang terbesar di dunia saat ini. Selayaknya perusahaan yang sudah melakukan *offering* kepada publik, Intel juga menggarap salah satu sumber pemodalannya melalui menjual saham dan menjual produk derivatifnya berupa opsi.

Intel memasarkan sahamnya pada bursa efek NASDAQ dengan kode INTC sejak 13 Oktober 1971. Berikut akan ditampilkan grafik harga saham INTC dalam 5 tahun terakhir.

Terlihat dalam 5 tahun terakhir, harga saham INTC mengalami penurunan yang cukup signifikan. Hal ini tentu sangat disayangkan, terlebih lagi, saham INTC merupakan saham yang digadang-gadang menjadi saham *"Blue Chip"* atau saham yang aman untuk dijadikan investasi jangka panjang dikarenakan fundamental perusahaan yang sudah sangat baik dan prospek jangka panjangnya juga sangat menjanjikan mengingat semakin besarnya teknologi dipakai pada kehidupan zaman sekarang. Hal ini tentu semakin membuat INTC menjadi kurang menarik di mata investor.

Baru-baru ini, Intel menjalankan skema *reverse stock split* yaitu membeli kembali saham yang beredar agar harga saham bisa naik. Namun, hal tersebut tidak bertahan lama dan harga INTC kembali turun.

1.2 Informasi Opsi yang Dipilih

Pada laporan ini, akan dipilih 3 opsi untuk masing-masing periode (1 bulan, 3 bulan, dan 6 bulan) dengan *Strike Price* (K) yang sama dan harga yang berbeda. Opsi yang dipilih untuk periode 1 bulan adalah:

- Opsi Call INTC230414C00020000 seharga \$9.85 dengan $K = \$20$
- Opsi Call INTC230414C00021000 seharga \$6.70 dengan $K = \$21$

- Opsi Put INTC230414P00022000 seharga \$0.03 dengan $K = \$22$

Opsi yang dipilih untuk periode 3 bulan adalah:

- Opsi Call INTC230519C00020000 seharga \$13.83 dengan $K = \$20$
- Opsi Call INTC230519C00021000 seharga \$12.53 dengan $K = \$21$
- Opsi Put INTC230519P00022000 seharga \$0.06 dengan $K = \$22$

Opsi yang dipilih untuk periode 6 bulan adalah:

- Opsi Call INTC230915C00020000 seharga \$12.95 dengan $K = \$20$
- Opsi Call INTC230915C00021000 seharga \$12.80 dengan $K = \$21$
- Opsi Put INTC230915P00022000 seharga \$0.40 dengan $K = \$22$

Data-data tersebut diperoleh dari website finance.yahoo.com yang diakses pada tanggal 14 Maret 2023.

2 Landasan Teori

2.1 Metode Binomial

Misal selang waktu [0,T] dibagi menjadi N subselang yang sama panjang dengan titik-titik bagi $0 = t_0 < t_1 < t_2 < ... < t_N = T$ dengan $t_j = j\Delta t$ $(j = 0, 1, ..., N)$, $\Delta t = \frac{T}{N}$, dan $S_j = S(t_j)$ harga saham pada saat t_j.

Asumsikan

1. Dalam selang waktu Δt harga saham dapat naik atau turun menjadi S $\rightarrow S_u$ atau S $\rightarrow S_d$ dengan 0 < d < u.
2. Peluang harga saham naik *P*(naik) = *p*
3. Ekspektasi *return* harga saham besarnya sama dengan *risk-free interest rate r*. Sehingga untuk harga saham yang bergerak secara acak dari S_j pada saat t_j menjadi S_{j+1} pada saat t_{j+1} yang berarti $E(S_{j+1}) = S_j e^{r\Delta t}$

Nilai *u*, *d*, dan *p* pada ketiga asumsi di atas dapat diketahui dengan membuat sistem persamaan yang menghubungkan ketiganya dengan menggunakan beberapa asumsi, untuk model diskrit, yaitu:

$$\begin{aligned} E(S_{j+1}) &= pS_j u + (1-p)S_j d \\ e^{r\Delta t} &= pu + (1-p)d \qquad (1) \\ Var(S_{j+1}) &= E\left((S_{j+1})^2\right) - \left(E\left(S_{j+1}\right)\right)^2 \\ &= p(S_j u)^2 + (1-p)(S_j d)^2 - S_j^2(pu + (1-p)d)^2 \qquad (2) \end{aligned}$$

Sehingga memberikan $p = \frac{e^{r\Delta t} - d}{u - d}$ dengan $0 \leq p \leq 1$ dan $d \leq e^{r\Delta t} \leq u$.

Untuk model kontinu, dengan $\mu = r$,

$$\begin{aligned} E\left((S_{j+1})^2\right) &= S_j^2 e^{(2r+\sigma^2)\Delta t} \\ Var\left(S_{j+1}\right) &= E\left((S_{j+1})^2\right) - \left(E\left(S_{j+1}\right)\right)^2 \\ &= S_j^2 e^{2r\Delta t}\left(e^{\sigma^2\Delta t} - 1\right) \qquad (3) \end{aligned}$$

Dari persamaan 2 dan 3, akan didapat persamaan yang menyatakan hubungan antara *u*, *d*, dan *p*. Yaitu,

$$\begin{aligned} e^{2r\Delta t}\left(e^{\sigma^2\Delta t} - 1\right) &= pu^2 + (1-p)d^2 - (e^{r\Delta t})^2 \\ e^{(2r+\sigma^2)\Delta t} &= pu^2 + (1-p)d^2 \qquad (4) \end{aligned}$$

Dari persamaan 1 dan 4, sudah didapat 2 persamaan yang menghubungkan nilai *u*, *d*, dan *p*. Dibutuhkan satu persamaan lagi agar ketiga variabel tersebut memiliki nilai tunggal. Yaitu:

$$ud = 1 \tag{5}$$

$$p = \frac{1}{2} \tag{6}$$

Sehingga, nilai *u*, *d*, dan *p* dapat ditentukan yaitu:

$$u = \beta + \sqrt{\beta^2 - 1}$$

$$d = \frac{1}{u} = \beta - \sqrt{\beta^2 - 1} p = \frac{e^{r\Delta t} - d}{u - d}$$

dimana,

$$\beta = \frac{1}{2}\left(e^{-r\Delta t} + e^{(r+\sigma^2)\Delta t}\right) \tag{7}$$

Metode binomial yang akan dipakai pada analisis ini adalah Model Cox-Ross-Rubenstein (CRR) yang mana model tersebut mengaproksimasi persamaan 7 sehingga menjadi bentuk yang lebih sederhana dengan menggunakan $e^x \approx 1 + x + x^2$ dan $\Delta t^2 \to 0$, sehingga:

$$\begin{aligned}
\beta &= \frac{1}{2}\left(e^{-r\Delta t} + e^{(r+\sigma^2)\Delta t}\right) \approx \frac{1}{2}\left(1 - r\Delta t + 1 + (r + \sigma^2)\Delta t\right) \\
&= 1 + \frac{1}{2}\sigma^2 \Delta t \\
\beta^2 &= \frac{1}{4}\left(e^{-2r\Delta t} + 2e^{\sigma^2\Delta t} + e^{2(r+\sigma^2)\Delta t}\right) \approx \frac{1}{4}\left(4 + 4\sigma^2\Delta t + (4r^2 + 3\sigma^4 + 4r\sigma^2)\Delta t^2\right) \\
&= 1 + \sigma^2 \Delta t
\end{aligned}$$

Lalu, akan didapat:

$$u = e^{\sigma\sqrt{\Delta t}} \tag{8}$$

$$d = \frac{1}{u} = e^{-\sigma\sqrt{\Delta t}} \tag{9}$$

Sedangkan, harga saham yang diberikan adalah sebagai berikut:

$$S_{ij} = S_0 u^i d^{j-i}, i = 0, 1, ..., j \tag{10}$$

Dengan S_{ij} menyatakan harga saham pada saat t_j dan telah terjadi kenaikan harga saham sebanyak i kali serta penurunan harga saham sebanyak j-i kali, dihitung sejak saat $t_0 = 0$. Pada saat *expiration date*, $t_N = N\Delta t = T$, terdapat $N + 1$ harga saham yang mungkin yaitu $[S_i N]_{i=0,1,..,N}$.

2.2 Metode Gerak Brown Geometri

Suatu koleksi peubah acak $X(t), t \geq 0$ suatu proses Gerak Brown dengan parameter drift μ dan parameter variansi σ^2, dan

$$S(t) = e^{X(t)}, t \geq 0 \tag{11}$$

Proses $S(t), t \geq 0$, disebut proses Gerak Brown Geometri dengan parameter drift μ dan parameter variansi σ^2. Karena $ln(S(t)), t \geq 0$ adalah Gerak Brown dan $ln(S(t+y)) - ln(S(t)) = ln\left(\frac{S(t+y)}{S(t)}\right)$, maka gerak tersebut saling bebas dari proses sampai waktu y dan memiliki distribusi normal dengan mean μt dan variansi $\sigma^2 t$.

Jika harga saham $S(t)$ dimodelkan sebagai proses GBM dengan $S(0) = 0$,

$$S(t) = se^{X(t)}, t \geq 0,$$

dengan $X(t), t \geq 0$, adalah proses Gerak Brown dengan $X(0) = 0$. Jika X berdistribusi normal,

$$E(e^X) = e^{E(X)+Var(X)/2}.$$

Sehingga $S(t), t \geq 0$, adalah proses GBM dengan drift μ dan volatilitas σ dan $S(0) = s$,

$$E(S(t)) = e^{(\mu+\sigma^2/2)t}.$$

Dalam asumsi GBM, ekspektasi harga saham memiliki laju $\mu + \frac{1}{2}\sigma^2$, yang biasa disebut laju GBM. Oleh karena itu, suatu proses GBM dengan parameter laju μ dan volatilitas σ, akan memiliki parameter drift $\mu - \frac{1}{2}\sigma^2$.

2.3 Penentuan Harga Opsi

2.3.1 Pendahuluan Opsi *Put* dan *Call*

Dalam opsi, terdapat beberapa istilah yang akan digunakan, yaitu:

1. *Call* : hak untuk membeli
2. *Put* : hak untuk menjual
3. C : harga opsi *call*
4. P : harga opsi *Put*
5. K : *strike price* atau harga yang berlaku
6. S_T : harga saham saat *expiration date*

7. T : *expiration date* atau waktu jatuh tempo opsi

Misalkan, sebuah opsi *call* dibeli pemegang dengan harga C dari penerbit, atau opsi *put* dibeli dengan harga P. *payoff* atau pendapatan dari opsi adalah sebagai berikut:

$$PO_{call} = maks(S_T - K, 0) \tag{12}$$

$$PO_{put} = maks(K - S_T, 0) \tag{13}$$

Saat $S_T > K$, opsi *call* dalam kondisi *in the money*, pemegang akan melaksanakan haknya dengan membeli saham dari penerbit seharga K dan menjual saham yang baru seharga S_T. Sedangkan, opsi *put* dalam kondisi *out of the money*, pemegang tidak melaksanakan haknya.

Sedangkan, saat $S_T < K$, opsi *call* dalam kondisi *out of the money*, pemegang tidak melaksanakan haknya. Sedangkan, opsi *put* dalam kondisi *in the money*, pemegang akan melaksanakan haknya.

Payoff atau pendapatan berbeda dengan *Profit* atau keuntungan, *profit* adalah *payoff* yang harus dikurangi dengan *future value* dari nilai harga opsi saat di awal periode, atau dapat ditulis sebagai berikut untuk opsi *call*:

$$Profit = PO - C(1 + r)^T$$

Lalu, berikut *profit* untuk opsi *put*:

$$Profit = PO - P(1 + r)^T$$

2.3.2 Model Binomial Multiperiode

Misal $\{C_{iN}\}_{i=0,1,...,N}$ menyatakan nilai-nilai *payoff* (pendapatan) pada saat jatuh tempo untuk opsi *call* Eropa, maka

$$C_{iN} = max\{S_{iN} - K, 0\}, i = 0, 1, ..., N. \tag{14}$$

Sedangkan, nilai-nilai *payoff* pada saat jatuh tempo untuk sebuah opsi *put* Eropa diberikan oleh

$$P_{iN} = max\{K - S_{iN}, 0\}, i = 0, 1, ..., N. \tag{15}$$

Selanjutnya, perhitungan *payoff* dilakukan dalam arah mundur untuk memperoleh nilai opsi pada saat $t_0 = 0$. misal V_{ij} adalah nilai opsi pada saat t_j yang berkaitan dengan nilai saham S_{ij}, dengan $V_{ij} = C_{ij}$ untuk *call* dan $V_{ij} = P_{ij}$ untuk *put*. Nilai opsi dihitung sebagai rata-rata dari nilai opsi $V_{i,j+1}$ dan $V_{i+1,j+1}$ pada saat t_{j+1}. Untuk $i = 0, 1, ..., j$ dan

$j = N-1, N-2, ..., 1, 0.$

$$V_{i,j} = e^{-r\Delta t(pV_{i+1,j+1}+(1-p)V_{i,j+1})} \tag{16}$$

dengan p peluang harga saham naik. Formulasi opsi *call* Eropa untuk $i = 0, 1, ..., j$, $j = N-1, N-2, ...1, 0$ adalah

$$C_{i,j} = e^{-r\Delta t(pC_{i+1,j+1}+(1-p)C_{i,j+1})} \tag{17}$$

Formulasi opsi *put* Eropa adalah

$$P_{i,j} = e^{-r\Delta t(pP_{i+1,j+1}+(1-p)P_{i,j+1})} \tag{18}$$

2.3.3 Model Gerak Brown Geometri

Pada perhitungan harga opsi dengan model Gerak Brown Geometri, akan difokuskan pada perhitungan payoff dari opsi. Harga saham S_T yang dipakai adalah harga saham dengan perhitungan yang sudah dijelaskan sebelumnya pada bagian 2.2. *Payoff* dari opsi yang akan dibahas pada bagian ini juga sama dengan bagian 2.3.1, yaitu:

$$PO_{call} = maks(S_T - K, 0)$$
$$PO_{put} = maks(K - S_T, 0)$$

Kemudian, *profit*-nya adalah:

$$Profit_{Call} = PO - C(1+r)^T Profit_{Put} = PO - P(1+r)^T$$

2.3.4 Rumus Black-Scholes

Rumus Black-Scholes adalah persamaan diferensial parsial yang dapat mengestimasi harga suatu opsi. Cara untuk menentukan rumus ini dapat diperoleh dari tiga cara yaitu menggunakan asumsi bahwa harga saham berdistribusi lognormal, dengan menyelesaikan persamaan diferensial, dan yang terakhir dengan menggunakan sifat teorema limit pusat pada metode binomial. Pada tugas kali ini akan digunakan cara pertama untuk memperoleh rumus Black-Scholes.

Misalkan harga saham saat *T* dinyatakan dengan *S(T)*, diketahui, model pergerakan harga saham adalah sebagai berikut:

1. $$X = \ln\left(\frac{S(T)}{S(0)}\right) \sim N(\mu T, \sigma^2 T) \tag{19}$$

2.

$$S(0) = e^{-rT}E(S(T)) \tag{20}$$

dengan μ^* = μ - $\frac{1}{2}\sigma^2$ dan *r* adalah suku bunga bebas risiko.

Dari persamaan 19, fungsi kepadatan peluang untuk X dapat dituliskan sebagai berikut:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi T}}exp\left\{-\frac{1}{2}\left(\frac{x-\mu^*T}{\sigma\sqrt{T}}\right)^2\right\} \tag{21}$$

Selanjutnya, fkp untuk *S(T)* akan dicari dengan menggunakan teknik transformasi peubah acak melalui fkp *X*. Jacobian dari fungsi transformasi *X* ke *S(T)* dari persamaan 19 adalah,

$$|J| = |\frac{dX}{dS(T)}| = |\frac{1}{S(T)}| = \frac{1}{S(T)}$$

Dengan demikian, didapat fkp dari *S(T)* yaitu,

$$f(s(T)) = \frac{1}{s(T)\sigma\sqrt{2\pi T}}exp\left\{-\frac{1}{2}\left(\frac{ln\left(\frac{s(T)}{s(0)}\right)-\mu^*T}{\sigma\sqrt{T}}\right)^2\right\} \tag{22}$$

Ekspektasi *E[S(T)]* dapat diturunkan dengan,

$$\begin{aligned} E[\frac{S(T)}{S(0)}] = E[e^{ln\frac{S(T)}{S(0)}}] = E[e^X] &= e^{\mu^*T+\frac{1}{2}\sigma^2T} \\ E[S(T)] &= S(0)e^{\mu^*T+\frac{1}{2}\sigma^2T} \end{aligned} \tag{23}$$

Dari persamaan 23, substitusikan ke persamaan 20,

$$\begin{aligned} S(0) &= e^{-rT}E(S(T)) \\ &= e^{-rT}S(0)e^{\mu^*T+\frac{1}{2}\sigma^2T} \\ &= S(0)e^{-rT+\mu^*T+\frac{1}{2}\sigma^2T} \\ 1 &= e^{-rT+\mu^*T+\frac{1}{2}\sigma^2T} \\ 0 &= -rT+\mu^*T+\frac{1}{2}\sigma^2T \\ \mu^* &= r-\frac{1}{2}\sigma^2 \end{aligned} \tag{24}$$

Substitusikan persamaan 24 ke 22,

$$f(s(T)) = \frac{1}{s(T)\sigma\sqrt{2\pi T}} exp \left\{ -\frac{1}{2} \left(\frac{ln\left(\frac{s(T)}{s(0)}\right) - (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}} \right)^2 \right\} \tag{25}$$

Diketahui harga call adalah $C = e^{-rT}E[maks(S(T) - K), 0]$, sehingga akan didapat,

$$\begin{aligned}
C &= e^{-rT}E[maks(S(T) - K, 0)] \\
&= e^{-rT}\int_0^\infty maks(S(T) - K, 0)f(S(T))\,dS(T) \\
&= e^{-rT}\int_0^K maks(S(T) - K, 0)f(S(T))\,dS(T) + e^{-rT}\int_K^\infty maks(S(T) - K, 0)f(S(T))\,dS(T) \\
&= e^{-rT}\left[\int_0^K 0.f(S(T))\,dS(T) + \int_K^\infty (S(T) - K)f(S(T))\,dS(T)\right] \\
&= e^{-rT}\int_K^\infty (S(T) - K)f(S(T))\,dS(T) \\
&= e^{-rT}\int_K^\infty \frac{(S(T) - K)}{s(T)\sigma\sqrt{2\pi T}} exp \left\{ -\frac{1}{2} \left(\frac{ln\left(\frac{s(T)}{s(0)}\right) - (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}} \right)^2 \right\} dS(T)
\end{aligned} \tag{26}$$

Integrasi pada persamaan 26 dapat dilakukan dengan transformasi sebagai berikut:

$$\begin{aligned}
z &= \frac{ln\left(\frac{s(T)}{s(0)}\right) - (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}} \\
ln\left(\frac{s(T)}{s(0)}\right) &= \sigma\sqrt{T}z + (r - \frac{1}{2}\sigma^2)T
\end{aligned}$$

Dengan demikian, jika S_0 = S(0),

$$\begin{aligned}
S(T) &= S_0 e^{\sigma\sqrt{T}z + (r - \frac{1}{2}\sigma^2)T} \\
\frac{dz}{dS(T)} &= \frac{1}{S(T)\sigma\sqrt{T}} \\
dz &= \frac{dS(T)}{S(T)\sigma\sqrt{T}}
\end{aligned} \tag{27}$$

Selanjutnya untuk transformasi pada integral, batas bawah integrasi untuk S(T) yaitu K, diubah menjadi batas bawah integrasi untuk z menjadi A, sehingga:

$$A = \frac{ln\left(\frac{K}{S_0}\right) - (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}} \tag{28}$$

Dengan transformasi tersebut, persamaan 26 dapat ditulis seperti berikut:

$$
\begin{aligned}
C &= e^{-rT}\frac{1}{\sqrt{2\pi}}\int_A^{\infty}\left[S_0 e^{z\sigma\sqrt{T}} + (r-\frac{1}{2}\sigma^2)T - K\right]e^{-\frac{1}{2}z^2}dz \\
&= \frac{1}{\sqrt{2\pi}}\int_A^{\infty} S_0 e^{z\sigma\sqrt{T}-\frac{1}{2}\sigma^2 T-\frac{1}{2}z^2}dz - \frac{Ke^{-rT}}{\sqrt{2\pi}}\int_A^{\infty} e^{-\frac{1}{2}z^2}dz \\
&= \frac{S_0}{\sqrt{2\pi}}\int_A^{\infty} e^{-\frac{1}{2}(z-\sigma\sqrt{T})^2}dz - Ke^{-rT}N(-A) \\
&= \frac{S_0}{\sqrt{2\pi}}\int_{A-\sigma\sqrt{T}}^{\infty} e^{-\frac{1}{2}u^2}du - Ke^{-rT}N(-A) \\
&= S_0 N(-A+\sigma\sqrt{T}) - Ke^{-rT}N(-A)
\end{aligned} \tag{29}
$$

Dimana pada sebelumnya sudah dimisalkan bahwa $u = z - \sigma\sqrt{T}$.
Bila dimisalkan $d_1 = -A + \sigma\sqrt{T}$ dan $d_2 = -A = d_1 - \sigma\sqrt{T}$, persamaan 29 dapat ditulis sebagai berikut:

$$C = S_0 N(d_1) - Ke^{-rT}N(d_2) \tag{30}$$

Dimana:

$$d_1 = -A + \sigma\sqrt{T} = -\frac{ln\left(\frac{K}{S_0}\right) - (r-\frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}} + \sigma\sqrt{T}$$

$$d_1 = \frac{ln\left(\frac{S_0}{K}\right) + \left(r+\frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}} \tag{31}$$

$$d_2 = d_1 - \sigma\sqrt{T} = \frac{ln\left(\frac{S_0}{K}\right) + \left(r+\frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}} - \sigma\sqrt{T}$$

$$d_2 = \frac{ln\left(\frac{S_0}{K}\right) + \left(r-\frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}} \tag{32}$$

Lalu, dengan Teorema *Put Call Parity*, akan didapat persamaan harga Put yaitu:

$$
\begin{aligned}
P &= C - S_t + Ke^{-rT} \\
&= Ke^{-rT}N(-d_2) - S_t N(d_1)
\end{aligned} \tag{33}
$$

3 Hasil Perhitungan dan Analisis

3.1 Perhitungan Harga Opsi dengan Periode 1 Bulan

Pertama-tama, akan dilakukan pengambilan data harga saham INTC bulanan dari periode Februari 2022 sampai periode Februari 2023 (13 data). Data harga saham ini diperoleh dari website finance.yahoo.com melalui *library* yfinance pada Python. Selanjutnya, akan dihitung *return* harga saham menggunakan rumus:

$$r_t = \begin{cases} 0, & i = 0 \\ \ln\left(\frac{S_i}{S_{i-1}}\right), & i = 1, 2, \cdots, 12 \end{cases}$$

dengan S_i menyatakan data harga Close ke-i. Berikut adalah data *Close* harga saham INTC yang diperoleh dan hasil perhitungan *return*-nya:

Tabel 3.1. *Return* Harga Saham INTC Periode 1 Bulan

Date	Close	$\ln\left(\frac{S_i}{S_{i-1}}\right)$
2022-02-01	45.32675	0.00000
2022-03-01	47.45295	0.04584
2022-04-01	41.73676	-0.12836
2022-05-01	42.53148	0.01886
2022-06-01	36.10265	-0.16388
2022-07-01	35.04110	-0.02984
2022-08-01	30.80451	-0.12886
2022-09-01	25.12050	-0.20398
2022-10-01	27.71345	0.09823
2022-11-01	29.31212	0.05608
2022-12-01	26.11183	-0.11561
2023-01-01	27.91980	0.06695
2023-02-01	24.62989	-0.12538

Setelah itu, akan dihitung nilai parameter *drift* (μ), parameter volatilitas (σ), dan suku bunga *risk-neutral* (r). Perhatikan bahwa:

$$r_t \sim \mathcal{N}(\tilde{\mu}, \tilde{\sigma}^2) = \mathcal{N}\left(\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t, \sigma^2 \Delta t\right)$$

Sehingga, nilai-nilai tersebut dapat dihitung menggunakan rumus:

$$\tilde{\mu} = \frac{1}{n}\sum_{t=0}^{n} r_t = -0.04692$$
$$\tilde{\sigma}^2 = \frac{1}{n-1}\sum_{t=0}^{n}(r_i - \tilde{\mu})^2 = 0.00943$$
$$\mu = \frac{1}{\Delta t}\left(\tilde{\mu} + \frac{1}{2}\tilde{\sigma}^2\right) = -42.20144$$
$$\sigma = \frac{\tilde{\sigma}}{\sqrt{\Delta t}} = 3.07141$$

Nilai suku bunga *risk-neutral* (r) Negara Amerika Serikat diperoleh dari website federalreserve.gov, yaitu sebesar 2.5% per tahun pada tahun 2022 atau 0.21% per bulan.

3.1.1 Metode Binomial

Pertama-tama, akan dibangkitkan data pergerakan harga saham INTC periode 1 bulan dengan $\Delta t = \frac{1}{1000}$ menggunakan metode binomial. Harga saham pada saat t_0 merupakan harga saham INTC pada bulan Februari 2023, yaitu $S_0 = 24.629887$. Sehingga, harga saham yang mungkin terjadi adalah:

$$S_{ij} = S_0 u^i d^{j-i}, \quad i = 0, 1, \cdots, j$$

yang mana j menyatakan banyak *steps* dan i menyatakan banyak kenaikan harga saham. Nilai untuk u dan d ditentukan oleh:

$$\beta = \frac{1}{2}\left(e^{-r\Delta t} + e^{(r+\sigma^2)\Delta t}\right) = 1.00474$$
$$u = \beta + \sqrt{\beta^2 - 1} = 1.10221$$
$$d = \frac{1}{u} = 0.90726$$

Setelah diperoleh nilai-nilai harga sahamnya, akan dihitung harga opsi INTC untuk opsi-opsi yang dipilih sebelumnya menggunakan cara induksi mundur *Payoff*. Rincian dari hasil perhitungan S_{ij} dan harga opsi dapat dilihat pada *code Python* yang terdapat pada Bab Lampiran. Jadi, diperoleh nilai harga opsi:

- Harga Opsi Call 1 dengan K = $20 adalah $21.88
- Harga Opsi Call 2 dengan K = $21 adalah $21.81
- Harga Opsi Put dengan K = $22 adalah $19.12

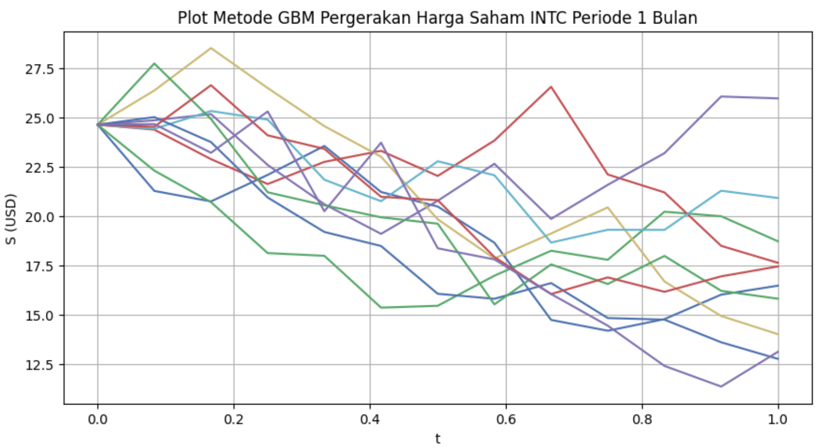
3.1.2 Metode Gerak Brown Geometri

Sekarang, akan dilakukan 1000 simulasi dengan masing-masing 12 *steps* data pergerakan harga saham INTC untuk 1 bulan menggunakan metode gerak brown geometri (jika dilakukan 1000 *steps*, harga saham akan *collapse* ke 0 karena harga saham yang cenderung turun). Harga saham pada saat t_0 merupakan harga saham INTC pada bulan Februari 2023, yaitu $S(t_0) = 24.629887$. Nilai harga pada saat t adalah:

$$S(t) = S(t_0)e^{\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t}Z}$$

yang mana iid $Z \sim \mathcal{N}(0, 1)$.

Hasil simulasi 1000 pergerakan harga saham yang diperoleh dapat dilihat pada *code Python* yang terdapat pada Bab Lampiran. Berikut adalah visualisasi untuk 10 lintasan pertama:



Gambar 3.1. Model Gerak Brown Geometri: Pergerakan Harga Saham INTC Periode 1 Bulan

Setelah diperoleh nilai-nilai harga sahamnya, akan dihitung harga opsi INTC untuk opsi-opsi yang dipilih sebelumnya menggunakan cara *Present Value* dari rata-rata *Payoff*. Hasil perhitungannya dapat dilihat pada *code Python* yang terdapat pada Bab Lampiran. Jadi, diperoleh nilai harga opsi:

- Harga Opsi Call 1 dengan K = $20 adalah $0.44
- Harga Opsi Call 2 dengan K = $21 adalah $0.34
- Harga Opsi Put dengan K = $22 adalah $7.73

3.1.3 Metode Black-Scholes

Terlebih dahulu kita hitung nilai d_1 dan d_2, dengan rumus:

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}$$
$$d_2 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r - \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}$$

Untuk Opsi Call 1 dengan K = $20,

$$\begin{aligned} C &= S_0N(d_1) - Ke^{-r\Delta t}N(d_2) \\ &= 21.86843634591382 \\ &\approx \$21.87 \end{aligned}$$

Untuk Opsi Call 2 dengan K = $21,

$$\begin{aligned} C &= S_0N(d_1) - Ke^{-r\Delta t}N(d_2) \\ &= 21.798451133820155 \\ &\approx \$21.80 \end{aligned}$$

Untuk Opsi Put dengan K = $22,

$$\begin{aligned} P &= Ke^{-r\Delta t}N(-d_2) - S_0N(-d_1) \\ &= 19.100592545364684 \\ &\approx \$19.10 \end{aligned}$$

3.2 Perhitungan Harga Opsi dengan Periode 3 Bulan

Pertama-tama, akan dilakukan pengambilan data harga saham INTC per 3 bulan dari periode Februari 2020 sampai periode Februari 2023 (13 data). Data harga saham ini diperoleh dari website finance.yahoo.com melalui *library* yfinance pada Python. Selanjutnya, akan dihitung *return* harga saham menggunakan rumus:

$$r_t = \begin{cases} 0, & i = 0 \\ \ln\left(\frac{S_i}{S_{i-1}}\right), & i = 1, 2, \cdots, 12 \end{cases}$$

dengan S_i menyatakan data harga Close ke-i. Berikut adalah data *Close* harga saham INTC yang diperoleh dan hasil perhitungan *return*-nya:

Tabel 3.2. *Return* Harga Saham INTC Periode 3 Bulan

Date	Close	$\ln\left(\frac{S_i}{S_{i-1}}\right)$
2020-02-01	54.20736	0.00000
2020-05-01	43.34877	-0.22354
2020-08-01	40.44261	-0.06939
2020-11-01	51.04372	0.23280
2021-02-01	53.28597	0.04299
2021-05-01	50.05906	-0.06247
2021-08-01	45.94196	-0.08582
2021-11-01	46.07064	0.00280
2022-02-01	41.42124	-0.10638
2022-05-01	34.76628	-0.17515
2022-08-01	27.43647	-0.23677
2022-11-01	27.54774	0.00405
2023-02-01	24.62989	-0.11196

Setelah itu, akan dihitung nilai parameter *drift* (μ), parameter volatilitas (σ), dan suku bunga *risk-neutral* (r). Perhatikan bahwa:

$$r_t \sim \mathcal{N}(\tilde{\mu}, \tilde{\sigma}^2) = \mathcal{N}\left(\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t, \sigma^2 \Delta t\right)$$

Sehingga, nilai-nilai tersebut dapat dihitung menggunakan rumus:

$$\tilde{\mu} = \frac{1}{n}\sum_{t=0}^{n} r_t = -0.06068$$

$$\tilde{\sigma}^2 = \frac{1}{n-1}\sum_{t=0}^{n}(r_i - \tilde{\mu})^2 = 0.01422$$

$$\mu = \frac{1}{\Delta t}\left(\tilde{\mu} + \frac{1}{2}\tilde{\sigma}^2\right) = -53.57190$$

$$\sigma = \frac{\tilde{\sigma}}{\sqrt{\Delta t}} = 3.77076$$

Nilai suku bunga *risk-neutral* (r) Negara Amerika Serikat diperoleh dari website federalreserve.gov, yaitu sebesar 2.5% per tahun pada tahun 2022 atau 0.62% per 3 bulan.

3.2.1 Metode Binomial

Pertama-tama, akan dibangkitkan data pergerakan harga saham INTC periode 3 bulan dengan $\Delta t = \frac{1}{1000}$ menggunakan metode binomial. Harga saham pada saat t_0 merupakan harga saham INTC pada bulan Februari 2023, yaitu $S_0 = 24.629887$. Sehingga, harga saham yang mungkin terjadi adalah:

$$S_{ij} = S_0 u^i d^{j-i}, \quad i = 0, 1, \cdots, j$$

yang mana j menyatakan banyak *steps* dan i menyatakan banyak kenaikan harga saham. Nilai untuk u dan d ditentukan oleh:

$$\begin{aligned} \beta &= \frac{1}{2}\left(e^{-r\Delta t} + e^{(r+\sigma^2)\Delta t}\right) = 1.00716 \\ u &= \beta + \sqrt{\beta^2 - 1} = 1.12704 \\ d &= \frac{1}{u} = 0.88728 \end{aligned}$$

Setelah diperoleh nilai-nilai harga sahamnya, akan dihitung harga opsi INTC untuk opsi-opsi yang dipilih sebelumnya menggunakan cara induksi mundur *Payoff*. Rincian dari hasil perhitungan S_{ij} dan harga opsi dapat dilihat pada *code Python* yang terdapat pada Bab Lampiran. Jadi, diperoleh nilai harga opsi:

- Harga Opsi Call 1 dengan K = \$20 adalah \$23.33
- Harga Opsi Call 2 dengan K = \$21 adalah \$23.30
- Harga Opsi Put dengan K = \$22 adalah \$20.64

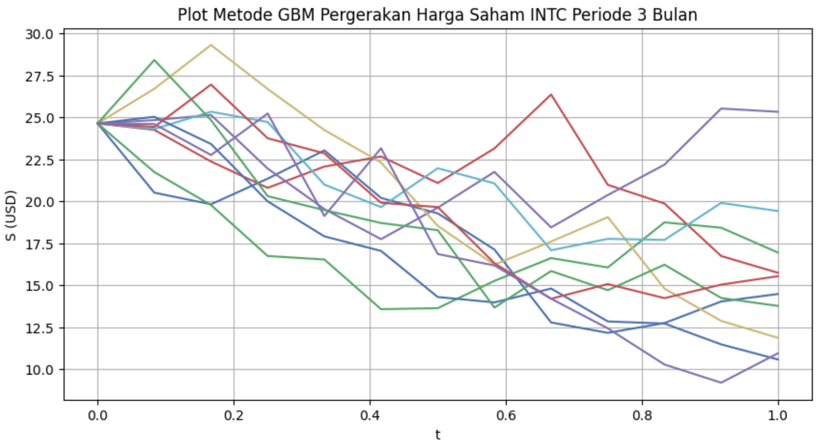
3.2.2 Metode Gerak Brown Geometri

Sekarang, akan dilakukan 1000 simulasi dengan masing-masing 12 *steps* data pergerakan harga saham INTC untuk 3 bulan menggunakan metode gerak brown geometri (jika dilakukan 1000 *steps*, harga saham akan *collapse* ke 0 karena harga saham yang cenderung turun). Harga saham pada saat t_0 merupakan harga saham INTC pada bulan Februari 2023, yaitu $S(t_0) = 24.629887$. Nilai harga pada saat t adalah:

$$S(t) = S(t_0) e^{\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t} Z}$$

yang mana iid $Z \sim \mathcal{N}(0, 1)$.

Hasil simulasi 1000 pergerakan harga saham yang diperoleh dapat dilihat pada *code Python* yang terdapat pada Bab Lampiran. Berikut adalah visualisasi untuk 10 lintasan pertama:



Gambar 3.2. Model Gerak Brown Geometri: Pergerakan Harga Saham INTC Periode 3 Bulan

Setelah diperoleh nilai-nilai harga sahamnya, akan dihitung harga opsi INTC untuk opsi-opsi yang dipilih sebelumnya menggunakan cara *Present Value* dari rata-rata *Payoff*. Hasil perhitungannya dapat dilihat pada *code Python* yang terdapat pada Bab Lampiran. Jadi, diperoleh nilai harga opsi:

- Harga Opsi Call 1 dengan K = $20 adalah $0.37
- Harga Opsi Call 2 dengan K = $21 adalah $0.30
- Harga Opsi Put dengan K = $22 adalah $9.64

3.2.3 Metode Black-Scholes

Terlebih dahulu kita hitung nilai d_1 dan d_2, dengan rumus:

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}$$
$$d_2 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r - \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}$$

Untuk Opsi Call 1 dengan K = $20,

$$\begin{aligned} C &= S_0 N(d_1) - Ke^{-r\Delta t}N(d_2) \\ &= 23.31348186709549 \\ &\approx \$23.31 \end{aligned}$$

Untuk Opsi Call 2 dengan K = $21,

$$\begin{aligned} C &= S_0N(d_1) - Ke^{-r\Delta t}N(d_2) \\ &= 23.280352621815556 \\ &\approx \$23.28 \end{aligned}$$

Untuk Opsi Put dengan K = $22,

$$\begin{aligned} P &= Ke^{-r\Delta t}N(-d_2) - S_0N(-d_1) \\ &= 20.618122891494366 \\ &\approx \$20.62 \end{aligned}$$

3.3 Perhitungan Harga Opsi dengan Periode 6 Bulan

Pertama-tama, akan dilakukan pengambilan data harga saham INTC per 6 bulan dari periode Februari 2017 sampai periode Februari 2023 (13 data). Data harga saham ini diperoleh dari website finance.yahoo.com melalui *library* yfinance pada Python. Selanjutnya, akan dihitung *return* harga saham menggunakan rumus:

$$r_t = \begin{cases} 0, & i = 0 \\ \ln\left(\frac{S_i}{S_{i-1}}\right), & i = 1, 2, \cdots, 12 \end{cases}$$

dengan S_i menyatakan data harga Close ke-i. Berikut adalah data *Close* harga saham INTC yang diperoleh dan hasil perhitungan *return*-nya:

Tabel 3.3. *Return* Harga Saham INTC Periode 6 Bulan

Date	Close	$\ln\left(\frac{S_i}{S_{i-1}}\right)$
2017-02-01	30.23727	0.00000
2017-08-01	38.60631	0.24434
2018-02-01	44.39914	0.13980
2018-08-01	40.83005	-0.08380
2019-02-01	45.00499	0.09736
2019-08-01	50.46901	0.11459
2020-02-01	54.20736	0.07146
2020-08-01	40.44261	-0.29293
2021-02-01	53.28598	0.27579
2021-08-01	45.94196	-0.14829
2022-02-01	41.42123	-0.10359
2022-08-01	27.43647	-0.41192
2023-02-01	24.62989	-0.10791

Setelah itu, akan dihitung nilai parameter *drift* (μ), parameter volatilitas (σ), dan suku bunga *risk-neutral* (r). Perhatikan bahwa:

$$r_t \sim \mathcal{N}(\tilde{\mu}, \tilde{\sigma}^2) = \mathcal{N}\left(\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t, \sigma^2 \Delta t\right)$$

Sehingga, nilai-nilai tersebut dapat dihitung menggunakan rumus:

$$\tilde{\mu} = \frac{1}{n}\sum_{t=0}^{n} r_t = -0.01578$$

$$\tilde{\sigma}^2 = \frac{1}{n-1}\sum_{t=0}^{n}(r_i - \tilde{\mu})^2 = 0.03744$$

$$\mu = \frac{1}{\Delta t}\left(\tilde{\mu} + \frac{1}{2}\tilde{\sigma}^2\right) = 2.93973$$

$$\sigma = \frac{\tilde{\sigma}}{\sqrt{\Delta t}} = 6.11846$$

Nilai suku bunga *risk-neutral* (r) Negara Amerika Serikat diperoleh dari website federalreserve.gov, yaitu sebesar 2.5% per tahun pada tahun 2022 atau 1.24% per 6 bulan.

3.3.1 Metode Binomial

Pertama-tama, akan dibangkitkan data pergerakan harga saham INTC periode 6 bulan dengan $\Delta t = \frac{1}{1000}$ menggunakan metode binomial. Harga saham pada saat t_0 merupakan harga saham INTC pada bulan Februari 2023, yaitu $S_0 = 24.629887$. Sehingga, harga saham yang mungkin terjadi adalah:

$$S_{ij} = S_0 u^i d^{j-i}, \quad i = 0, 1, \cdots, j$$

yang mana j menyatakan banyak *steps* dan i menyatakan banyak kenaikan harga saham. Nilai untuk u dan d ditentukan oleh:

$$\begin{aligned}
\beta &= \frac{1}{2}\left(e^{-r\Delta t} + e^{(r+\sigma^2)\Delta t}\right) = 1.01907 \\
u &= \beta + \sqrt{\beta^2 - 1} = 1.21531 \\
d &= \frac{1}{u} = 0.82284
\end{aligned}$$

Setelah diperoleh nilai-nilai harga sahamnya, akan dihitung harga opsi INTC untuk opsi-opsi yang dipilih sebelumnya menggunakan cara induksi mundur *Payoff*. Rincian dari hasil perhitungan S_{ij} dan harga opsi dapat dilihat pada *code Python* yang terdapat pada Bab Lampiran. Jadi, diperoleh nilai harga opsi:

- Harga Opsi Call 1 dengan K = $20 adalah $24.584
- Harga Opsi Call 2 dengan K = $21 adalah $24.583
- Harga Opsi Put dengan K = $22 adalah $21.95

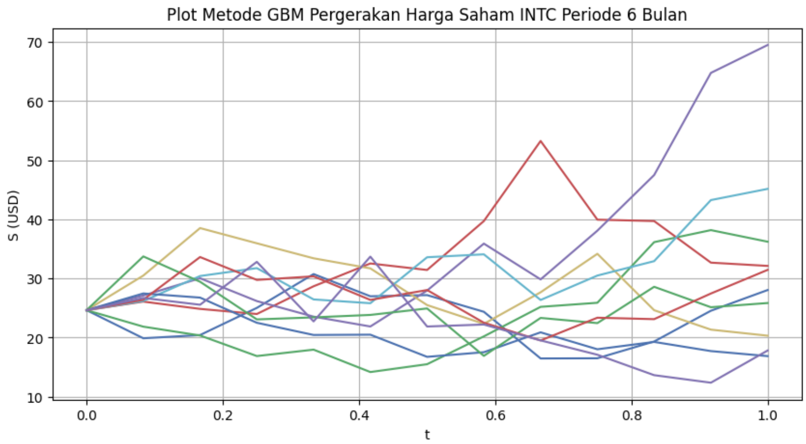
3.3.2 Metode Gerak Brown Geometri

Sekarang, akan dilakukan 1000 simulasi dengan masing-masing 12 *steps* data pergerakan harga saham INTC untuk 3 bulan menggunakan metode gerak brown geometri (jika dilakukan 1000 *steps*, harga saham akan *collapse* ke 0 karena harga saham yang cenderung turun). Harga saham pada saat t_0 merupakan harga saham INTC pada bulan Februari 2023, yaitu $S(t_0) = 24.629887$. Nilai harga pada saat t adalah:

$$S(t) = S(t_0) e^{\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t} Z}$$

yang mana iid $Z \sim \mathcal{N}(0, 1)$.

Hasil simulasi 1000 pergerakan harga saham yang diperoleh dapat dilihat pada *code Python* yang terdapat pada Bab Lampiran. Berikut adalah visualisasi untuk 10 lintasan pertama:



Gambar 3.3. Model Gerak Brown Geometri: Pergerakan Harga Saham INTC Periode 6 Bulan

Setelah diperoleh nilai-nilai harga sahamnya, akan dihitung harga opsi INTC untuk opsi-opsi yang dipilih sebelumnya menggunakan cara *Present Value* dari rata-rata *Payoff*. Hasil perhitungannya dapat dilihat pada *code Python* yang terdapat pada Bab Lampiran. Jadi, diperoleh nilai harga opsi:

- Harga Opsi Call 1 dengan K = \$20 adalah \$8.00
- Harga Opsi Call 2 dengan K = \$21 adalah \$7.52
- Harga Opsi Put dengan K = \$22 adalah \$4.73

3.3.3 Metode Black-Scholes

Terlebih dahulu kita hitung nilai d_1 dan d_2, dengan rumus:

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}$$
$$d_2 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r - \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}$$

Untuk Opsi Call 1 dengan K = \$20,

$$\begin{aligned} C &= S_0 N(d_1) - K e^{-r\Delta t} N(d_2) \\ &= 24.58066001347872 \\ &\approx \$24.581 \end{aligned}$$

Untuk Opsi Call 2 dengan K = $21,

$$\begin{aligned} C &= S_0 N(d_1) - K e^{-r\Delta t} N(d_2) \\ &= 24.579434042435334 \\ &\approx \$24.579 \end{aligned}$$

Untuk Opsi Put dengan K = $22,

$$\begin{aligned} P &= K e^{-r\Delta t} N(-d_2) - S_0 N(-d_1) \\ &= 21.948079371531563 \\ &\approx \$21.95 \end{aligned}$$

3.4 Analisis Perbandingan dengan Harga Opsi pada 14 Maret 2023

Tabel 3.4. Perbandingan Harga Opsi INTC Simulasi dengan Harga pada 14 Maret 2023

	Harga Opsi											
	Periode 1 Bulan				Periode 3 Bulan				Periode 6 Bulan			
Opsi	Aktual	Bin	GBM	BS	Aktual	Bin	GBM	BS	Aktual	Bin	GBM	BS
C_1	$9.85	$21.88	$0.44	$21.87	$13.83	$23.33	$0.37	$23.31	$12.95	$24.584	$8.00	$24.581
C_2	$6.70	$21.81	$0.34	$21.80	$12.53	$23.30	$0.30	$23.28	$12.80	$24.583	$7.52	$24.579
P	$0.03	$19.12	$7.73	$19.10	$0.06	$20.64	$9.64	$20.62	$0.40	$21.95	$4.73	$21.95

Perhatikan bahwa,

- Untuk periode 1 bulan, harga aktual tiap opsi mempunyai perbedaan yang cukup besar dengan harga opsi dari simulasi. Harga opsi yang diperoleh dengan menggunakan metode binomial dan Black-Scholes cukup identik, sedangkan metode Gerak Brown Geometri memiliki harga opsi yang paling berbeda dikarenakan adanya pengaruh stokastik pada perhitungannya. Untuk semua opsi yang ditinjau, diperoleh kesimpulan bahwa harga opsi put lebih murah daripada harga opsi call (kecuali untuk metode GBM), serta opsi call dengan *strike price* lebih besar memiliki harga yang lebih murah.

- Untuk periode 3 bulan, harga aktual tiap opsi juga memiliki perbedaan yang cukup besar dengan harga opsi dari simulasi. Harga opsi yang diperoleh dengan menggunakan metode binomial dan Black-Scholes cukup identik, sedangkan metode Gerak Brown Geometri memiliki harga opsi yang paling berbeda dikarenakan adanya pengaruh stokastik pada perhitungannya. Untuk semua opsi yang ditinjau, diperoleh kesimpulan bahwa harga opsi put lebih murah daripada harga opsi call (kecuali untuk metode GBM), serta opsi call dengan *strike price* lebih besar memiliki harga yang lebih murah.

- Untuk periode 6 bulan, harga aktual tiap opsi juga mempunyai perbedaan yang cukup besar dengan harga opsi dari simulasi. Harga opsi yang diperoleh dengan menggunakan metode binomial dan Black-Scholes cukup identik, sedangkan metode Gerak Brown Geometri memiliki harga opsi yang paling berbeda dikarenakan adanya pengaruh stokastik pada perhitungannya. Untuk semua opsi yang ditinjau, diperoleh kesimpulan bahwa harga opsi put lebih murah daripada harga opsi call, serta harga opsi call dengan *strike price* lebih besar memiliki harga yang lebih murah.
- Untuk keseluruhannya, harga opsi aktual yang paling mahal adalah harga opsi untuk periode 3 bulan. Hal ini bisa saja disebabkan oleh historis harga saham yang cenderung turun dalam periode 3 bulan. Sementara, harga opsi simulasi menunjukkan bahwa semakin lama periodenya, semakin mahal juga harga opsinya. Semua harga opsi aktual dengan harga opsi simulasi metode binomial dan Black-Scholes menunjukkan selisih yang konstan, yaitu $\pm\$12$.

3.5 Analisis Opsi pada 14 April 2023

Harga saham INTC pada 14 April 2023 adalah \$31.91. Sehingga,

1. Untuk periode 1 bulan:
   - Opsi Call dengan *strike price* \$20 akan di-*exercise* karena nilai *payoff*-nya:
     $$PO = max\{S_T - K, 0\} = max\{\$31.91 - \$20, 0\} = \$11.91$$
     Sehingga, keuntungan yang akan diperoleh adalah:
     $$Profit = PO - C(1+r)^T = \$11.91 - \$9.85(1 + 0.21\%)^1 = \$2.04$$
   - Opsi Call dengan *strike price* \$21 akan di-*exercise* karena nilai *payoff*-nya:
     $$PO = max\{S_T - K, 0\} = max\{\$31.91 - \$21, 0\} = \$10.91$$
     Sehingga, keuntungan yang akan diperoleh adalah:
     $$Profit = PO - C(1+r)^T = \$10.91 - \$6.70(1 + 0.21\%)^1 = \$4.20$$
   - Opsi Put dengan *strike price* \$22 tidak akan di-*exercise* karena nilai *payoff*-nya:
     $$PO = max\{K - S_T, 0\} = max\{\$22 - \$31.91, 0\} = 0$$
2. Untuk periode 3 bulan:

- Opsi Call dengan *strike price* \$20 sebaiknya tidak di-*exercise* karena nilai *payoff*-nya:

$$PO = max\{S_T - K, 0\} = max\{\$31.91 - \$20, 0\} = \$11.91$$

Sehingga, keuntungan yang akan diperoleh adalah:

$$Profit = PO - C(1+r)^T = \$11.91 - \$13.83(1 + 0.62\%)^1 = -\$2.01$$

Karena perhitungan keuntungannya negatif (rugi), maka opsi sebaiknya tidak di-*exercise* walaupun dalam keadaan *in the money*.

- Opsi Call dengan *strike price* \$21 sebaiknya tidak di-*exercise* karena nilai *payoff*-nya:

$$PO = max\{S_T - K, 0\} = max\{\$31.91 - \$21, 0\} = \$10.91$$

Sehingga, keuntungan yang akan diperoleh adalah:

$$Profit = PO - C(1+r)^T = \$10.91 - \$12.53(1 + 0.62\%)^1 = -\$1.70$$

Karena perhitungan keuntungannya negatif (rugi), maka opsi sebaiknya tidak di-*exercise* walaupun dalam keadaan *in the money*.

- Opsi Put dengan *strike price* \$22 tidak akan di-*exercise* karena nilai *payoff*-nya:

$$PO = max\{K - S_T, 0\} = max\{\$22 - \$31.91, 0\} = 0$$

3. Untuk periode 6 bulan:

- Opsi Call dengan *strike price* \$20 sebaiknya tidak di-*exercise* karena nilai *payoff*-nya:

$$PO = max\{S_T - K, 0\} = max\{\$31.91 - \$20, 0\} = \$11.91$$

Sehingga, keuntungan yang akan diperoleh adalah:

$$Profit = PO - C(1+r)^T = \$11.91 - \$12.95(1 + 1.24\%)^1 = -\$1.20$$

Karena perhitungan keuntungannya negatif (rugi), maka opsi sebaiknya tidak di-*exercise* walaupun dalam keadaan *in the money*.

- Opsi Call dengan *strike price* \$21 sebaiknya tidak di-*exercise* karena nilai *payoff*-nya:

$$PO = max\{S_T - K, 0\} = max\{\$31.91 - \$21, 0\} = \$10.91$$

Sehingga, keuntungan yang akan diperoleh adalah:

$$Profit = PO - C(1 + r)^T = \$10.91 - \$12.80(1 + 1.24\%)^1 = -\$2.05$$

Karena perhitungan keuntungannya negatif (rugi), maka opsi sebaiknya tidak di-*exercise* walaupun dalam keadaan *in the money*.

- Opsi Put dengan *strike price* \$22 tidak akan di-*exercise* karena nilai *payoff*-nya:

$$PO = max\{K - S_T, 0\} = max\{\$22 - \$31.91, 0\} = 0$$

Lampiran Code Python

In [1]:
```
import yfinance as yf
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from scipy.stats import norm
plt.style.use('seaborn-deep')

stock_code = "INTC"
currency = "USD"
K = [20, 21, 22]
ticker = yf.Ticker(stock_code)
r_0 = 0.025
dt = 1/1000
```

Untuk Periode 1 Bulan

In [ ]:
```
data_1 = ticker.history(start="2022-02-01", end="2023-02-28", interval="1mo")
data_1
```

Out[ ]:

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2022-02-01 00:00:00-05:00	46.353008	47.483803	41.459242	45.326744	756814900	0.365	0.0
2022-03-01 00:00:00-05:00	45.518830	50.277527	42.196356	47.452950	849288700	0.000	0.0
2022-04-01 00:00:00-04:00	47.711474	47.778498	41.650593	41.736767	639472500	0.000	0.0
2022-05-01 00:00:00-04:00	41.861242	44.657097	38.596219	42.531479	773236800	0.365	0.0
2022-06-01 00:00:00-04:00	43.205444	43.359852	35.320955	36.102650	783353000	0.000	0.0
2022-07-01 00:00:00-04:00	35.793836	39.306631	34.008489	35.041096	792918100	0.000	0.0
2022-08-01 00:00:00-04:00	34.790181	36.372868	30.659752	30.804510	899717100	0.365	0.0
2022-09-01 00:00:00-04:00	30.774316	31.563902	25.091254	25.120499	889075300	0.000	0.0
2022-10-01 00:00:00-04:00	25.666383	28.483543	23.970238	27.713455	981043600	0.000	0.0
2022-11-01 00:00:00-04:00	28.074127	30.550110	26.485210	29.312119	788916300	0.365	0.0
2022-12-01 00:00:00-05:00	29.875964	30.113074	25.044831	26.111830	758549700	0.000	0.0
2023-01-01 00:00:00-05:00	26.724365	30.122954	26.052553	27.919800	840843300	0.000	0.0
2023-02-01 00:00:00-05:00	27.554252	30.794768	24.432294	24.629887	880323200	0.365	0.0

In [ ]:
```
tgl = data_1.index.tolist()
date = [i.date() for i in tgl]
stock = data_1["Close"]
dt = 1/1000
ret = []
for i in range(len(stock)):
    if i == 0:
        a = 0
    else:
```

```
        a = np.log(stock[i]/stock[i-1])
    ret.append(a)

tabel_saham = pd.DataFrame(list(zip(date, stock, ret)), columns=['Date', 'Close','ln(St/St-1)
tabel_saham
```

Out[ ]:

	Date	Close	ln(St/St-1)
0	2022-02-01	45.326744	0.000000
1	2022-03-01	47.452950	0.045841
2	2022-04-01	41.736767	-0.128356
3	2022-05-01	42.531479	0.018862
4	2022-06-01	36.102650	-0.163878
5	2022-07-01	35.041096	-0.029845
6	2022-08-01	30.804510	-0.128860
7	2022-09-01	25.120499	-0.203977
8	2022-10-01	27.713455	0.098234
9	2022-11-01	29.312119	0.056083
10	2022-12-01	26.111830	-0.115613
11	2023-01-01	27.919800	0.066948
12	2023-02-01	24.629887	-0.125375

In [ ]:

```
mu_tilde = np.mean(ret)
var_tilde = np.var(ret)
mu = 1/dt * (mu_tilde + 1/2 * var_tilde)
sigma = np.sqrt(var_tilde/dt)
r_1 = (1+r_0)**(1/12)-1
r = np.exp(r_1*dt)-1
print(f'''Return berdistribusi N({mu_tilde}, {var_tilde})
Sehingga,
parameter drift (mu) nya adalah {mu}
parameter volatilitas (sigma) nya adalah {sigma}
Serta, risk-neutral interest rate Amerika (r) adalah {round(100*r_1,2)}% per bulan''')
```

```
Return berdistribusi N(-0.046918202017932474, 0.009433557197861 91)
Sehingga,
parameter drift (mu) nya adalah -42.201423419001515
parameter volatilitas (sigma) nya adalah 3.0714096434474363
Serta, risk-neutral interest rate Amerika (r) adalah 0.21% per bulan
```

Metode Binomial

In [ ]:

```
S_0 = stock[-1]
beta = 0.5*(np.exp(-r*dt)+np.exp((r+sigma**2)*dt))
n = 1000
u = beta+np.sqrt((beta**2)-1)
d = 1/u
p = (np.exp(r*dt)-d)/(u-d)

S = np.ones((n+1, n+1))
for i in range(1, n+2):
    for j in range(1, i+1):
        S[j-1, i-1] = S_0*u**(j-1)*d**(i-j)
```

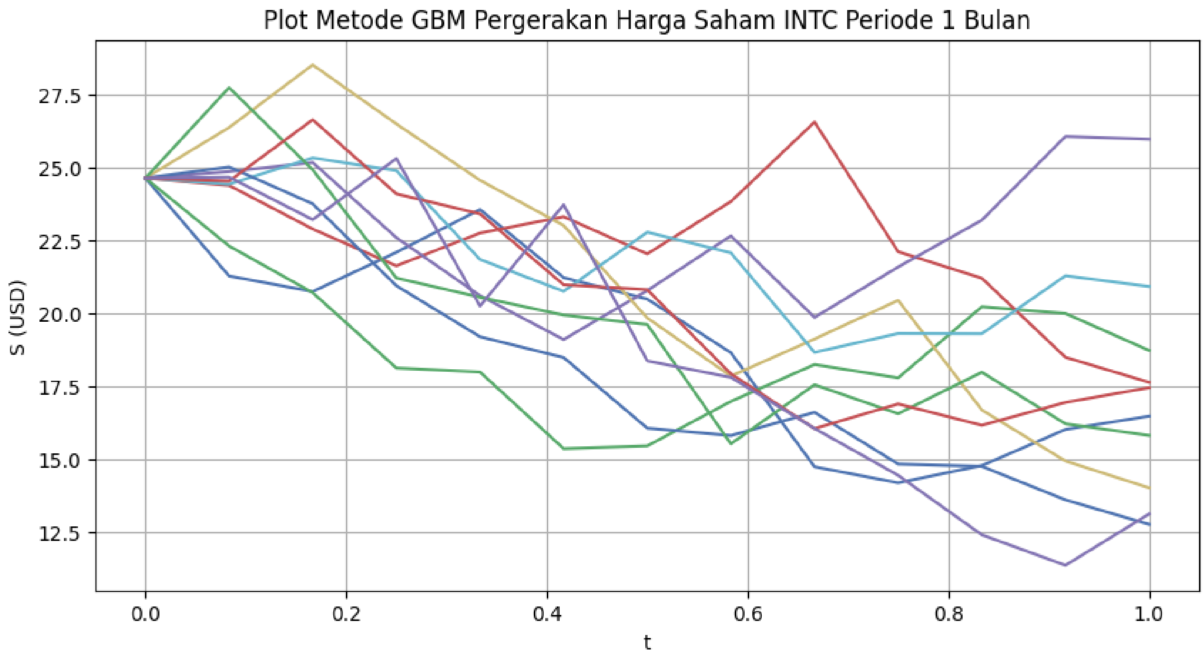
```python
C_1 = np.zeros((n+1, n+1))
C_2 = np.zeros((n+1, n+1))
P_1 = np.zeros((n+1, n+1))

for j in range(1, n+2):
    C_1[j-1, n] = max(S[j-1, n]-K[0], 0)
    C_2[j-1, n] = max(S[j-1, n]-K[1], 0)
    P_1[j-1, n] = max(K[2]-S[j-1, n], 0)

for i in range(n-1, -1, -1):
    for j in range(1, i+2):
        C_1[j-1, i] = max(S[j-1, i]-K[0], np.exp(-r*dt)*(p*C_1[j, i+1]+(1-p)*C_1[j-1, i+1]))
        C_2[j-1, i] = max(S[j-1, i]-K[1], np.exp(-r*dt)*(p*C_2[j, i+1]+(1-p)*C_2[j-1, i+1]))
        P_1[j-1, i] = max(K[2]-S[j-1, i], np.exp(-r*dt)*(p*P_1[j, i+1]+(1-p)*P_1[j-1, i+1]))

V_C1 = C_1[0, 0]
V_C2 = C_2[0, 0]
V_P1 = P_1[0, 0]
print(f'{beta = }, {u = }, {d = }')
print(f'{V_C1 = }, {V_C2 = }, {V_P1 = }')
```

```
beta = 1.0047390967336083, u = 1.1022103874045908, d = 0.9072678060626259
V_C1 = 21.88217096072731, V_C2 = 21.813962961663712, V_P1 = 19.118153888039576
```

Metode GBM

In [ ]:

```python
np.random.seed(108)

# Clear the current figure
plt.figure(figsize = (10, 5))

# Parameters
T = 1
L = 12
M = 1000

# Time steps
t = np.linspace(0, T, L+1)

# Simulate stock prices
log_returns = mu_tilde + np.sqrt(var_tilde) * np.random.randn(M, L)
cumulative_returns = np.cumprod(np.exp(log_returns), axis=1)
stock_prices = S_0 * np.hstack([np.ones((M, 1)), cumulative_returns])
stock_price_t1 = np.sort(stock_prices[:, -1])[::-1]

# Plot stock prices
plt.plot(t, stock_prices[:10].T)
plt.title(f'Plot Metode GBM Pergerakan Harga Saham {stock_code} Periode 1 Bulan')
plt.xlabel('t')
plt.ylabel(f'S ({currency})')
plt.grid()
plt.show()
```



In [ ]:
```
payoff_C1 = []
payoff_C2 = []
payoff_P1 = []
for i in range(len(stock_price_t1)):
  payoff_C1.append(np.max([stock_price_t1[i]-20, 0]))
  payoff_C2.append(np.max([stock_price_t1[i]-K[1], 0]))
  payoff_P1.append(np.max([K[2]-stock_price_t1[i], 0]))

C_1 = np.mean(payoff_C1)*np.exp(-r)
C_2 = np.mean(payoff_C2)*np.exp(-r)
P_1 = np.mean(payoff_P1)*np.exp(-r)
print(f'{C_1 = }, {C_2 = }, {P_1 = }')
```

```
C_1 = 0.4426994433445287, C_2 = 0.340771193314931, P_1 = 7.7287238443533495
```

Metode Black-Scholes

In [ ]:
```
# Untuk opsi call 1 K=20
S_0 = stock[-1]

d1 = (np.log(S_0/K[0])+(r+1/2*sigma**2)*1)/(sigma*np.sqrt(1))
d2 = (np.log(S_0/K[0])+(r-1/2*sigma**2)*1)/(sigma*np.sqrt(1))
C1 = S_0*norm.cdf(d1, loc=0, scale=1)-K[0]*np.exp(-r*1)*norm.cdf(d2, loc=0, scale=1)

# Untuk opsi call 2 K=21
S_0 = stock[-1]

d1 = (np.log(S_0/K[1])+(r+1/2*sigma**2)*1)/(sigma*np.sqrt(1))
d2 = (np.log(S_0/K[1])+(r-1/2*sigma**2)*1)/(sigma*np.sqrt(1))
C2 = S_0*norm.cdf(d1, loc=0, scale=1)-K[1]*np.exp(-r*1)*norm.cdf(d2, loc=0, scale=1)

# Untuk opsi put K=22
S_0 = stock[-1]

d1 = (np.log(S_0/K[2])+(r+1/2*sigma**2)*1)/(sigma*np.sqrt(1))
d2 = (np.log(S_0/K[2])+(r-1/2*sigma**2)*1)/(sigma*np.sqrt(1))
P1 = K[2]*np.exp(-r*1)*norm.cdf(-d2, loc=0, scale=1)-S_0*norm.cdf(-d1, loc=0, scale=1)

print(f'{C1 = }, {C2 = }, {P1 = }')
```

```
C1 = 21.868436673165935, C2 = 21.79845146947277, P1 = 19.100592889146874
```

Untuk Periode 3 Bulan

In [ ]:

```
data_3 = ticker.history(start="2020-02-01", end="2023-02-28", interval="3mo")
data_3
```

Out[ ]:

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2020-02-01 00:00:00-05:00	58.256195	61.536832	39.430932	54.207363	2020100000	0.330	0.0
2020-05-01 00:00:00-04:00	53.429877	59.133422	42.658530	43.348766	1788480500	0.330	0.0
2020-08-01 00:00:00-04:00	44.086841	51.357013	39.830684	40.442619	2086188800	0.330	0.0
2020-11-01 00:00:00-04:00	41.342569	58.804657	40.680502	51.043728	2523308700	0.330	0.0
2021-02-01 00:00:00-05:00	51.822536	63.437450	51.600239	53.285976	1937077900	0.348	0.0
2021-05-01 00:00:00-04:00	53.916928	54.438762	48.745153	50.059063	1524211000	0.348	0.0
2021-08-01 00:00:00-04:00	50.620532	52.570719	44.882473	45.941952	1487763900	0.348	0.0
2021-11-01 00:00:00-04:00	46.617975	53.110516	43.692553	46.070637	2222271300	0.348	0.0
2022-02-01 00:00:00-05:00	46.353007	49.897424	41.335708	41.421230	2245576100	0.365	0.0
2022-05-01 00:00:00-04:00	41.861236	44.657089	33.741765	34.766273	2349507900	0.365	0.0
2022-08-01 00:00:00-04:00	34.790181	36.372868	23.730668	27.436474	2769836000	0.365	0.0
2022-11-01 00:00:00-04:00	28.074128	30.550111	24.711082	27.547739	2388309300	0.365	0.0
2023-02-01 00:00:00-05:00	27.554252	30.794768	24.432294	24.629887	880323200	0.365	0.0

In [ ]:

```
tgl = data_3.index.tolist()
date = [i.date() for i in tgl]
stock = data_3["Close"]
dt = 1/1000
ret = []
for i in range(len(stock)):
    if i == 0:
        a = 0
    else:
        a = np.log(stock[i]/stock[i-1])
    ret.append(a)

tabel_saham = pd.DataFrame(list(zip(date, stock, ret)), columns=['Date', 'Close','ln(St/St-1)
tabel_saham
```

Out[ ]:

	Date	Close	ln(St/St-1)
0	2020-02-01	54.207363	0.000000
1	2020-05-01	43.348766	-0.223539
2	2020-08-01	40.442619	-0.069394
3	2020-11-01	51.043728	0.232799
4	2021-02-01	53.285976	0.042991
5	2021-05-01	50.059063	-0.062470
6	2021-08-01	45.941952	-0.085825
7	2021-11-01	46.070637	0.002797
8	2022-02-01	41.421230	-0.106382
9	2022-05-01	34.766273	-0.175146
10	2022-08-01	27.436474	-0.236774
11	2022-11-01	27.547739	0.004047
12	2023-02-01	24.629887	-0.111960

In [ ]:

```
mu_tilde = np.mean(ret)
var_tilde = np.var(ret)
mu = 1/dt * (mu_tilde + 1/2 * var_tilde)
sigma = np.sqrt(var_tilde/dt)
r_1 = (1+r_0)**(3/12)-1
r = np.exp(r_1*dt)-1
print(f'''Return berdistribusi N({mu_tilde}, {var_tilde})
Sehingga,
parameter drift (mu) nya adalah {mu}
parameter volatilitas (sigma) nya adalah {sigma}
Serta, risk-neutral interest rate Amerika (r) adalah {round(100*r_1,2)}% per 3 bulan''')
```

```
Return berdistribusi N(-0.060681241632016145, 0.01421865563735223)
Sehingga,
parameter drift (mu) nya adalah -53.57191381334003
parameter volatilitas (sigma) nya adalah 3.770763269863574
Serta, risk-neutral interest rate Amerika (r) adalah 0.62% per 3 bulan
```

Metode Binomial

In [ ]:

```
S_0 = stock[-1]
beta = 0.5*(np.exp(-r*dt)+np.exp((r+sigma**2)*dt))
n = 1000
u = beta+np.sqrt((beta**2)-1)
d = 1/u
p = (np.exp(r*dt)-d)/(u-d)

S = np.ones((n+1, n+1))
for i in range(1, n+2):
    for j in range(1, i+1):
        S[j-1, i-1] = S_0*u**(j-1)*d**(i-j)

C_1 = np.zeros((n+1, n+1))
C_2 = np.zeros((n+1, n+1))
P_1 = np.zeros((n+1, n+1))

for j in range(1, n+2):
```

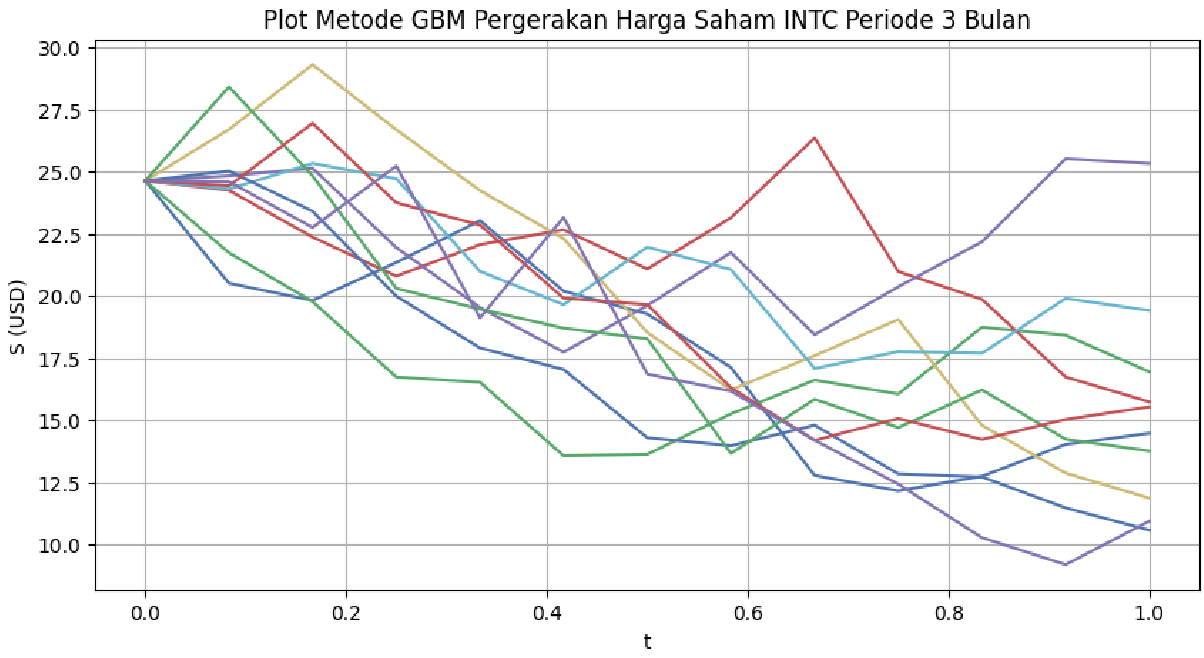
```
    C_1[j-1, n] = max(S[j-1, n]-K[0], 0)
    C_2[j-1, n] = max(S[j-1, n]-K[1], 0)
    P_1[j-1, n] = max(K[2]-S[j-1, n], 0)

for i in range(n-1, -1, -1):
    for j in range(1, i+2):
        C_1[j-1, i] = max(S[j-1, i]-K[0], np.exp(-r*dt)*(p*C_1[j, i+1]+(1-p)*C_1[j-1, i+1]))
        C_2[j-1, i] = max(S[j-1, i]-K[1], np.exp(-r*dt)*(p*C_2[j, i+1]+(1-p)*C_2[j-1, i+1]))
        P_1[j-1, i] = max(K[2]-S[j-1, i], np.exp(-r*dt)*(p*P_1[j, i+1]+(1-p)*P_1[j-1, i+1]))

V_C1 = C_1[0, 0]
V_C2 = C_2[0, 0]
V_P1 = P_1[0, 0]
print(f'{beta = }, {u = }, {d = }')
print(f'{V_C1 = }, {V_C2 = }, {V_P1 = }')
```

```
beta = 1.0071601108079904, u = 1.1270412552918645, d = 0.8872789663241163
V_C1 = 23.328314144899466, V_C2 = 23.29681881782948, V_P1 = 20.635321989812056
```

Metode GBM

In [ ]:

```
np.random.seed(108)

# Clear the current figure
plt.figure(figsize = (10, 5))

# Parameters
T = 1
L = 12
M = 1000

# Time steps
t = np.linspace(0, T, L+1)

# Simulate stock prices
log_returns = mu_tilde + np.sqrt(var_tilde) * np.random.randn(M, L)
cumulative_returns = np.cumprod(np.exp(log_returns), axis=1)
stock_prices = S_0 * np.hstack([np.ones((M, 1)), cumulative_returns])
stock_price_t1 = np.sort(stock_prices[:, -1])[::-1]

# Plot stock prices
plt.plot(t, stock_prices[:10].T)
plt.title(f'Plot Metode GBM Pergerakan Harga Saham {stock_code} Periode 3 Bulan')
plt.xlabel('t')
plt.ylabel(f'S ({currency})')
plt.grid()
plt.show()
```



In [ ]:
```
payoff_C1 = []
payoff_C2 = []
payoff_P1 = []
for i in range(len(stock_price_t1)):
  payoff_C1.append(np.max([stock_price_t1[i]-20, 0]))
  payoff_C2.append(np.max([stock_price_t1[i]-K[1], 0]))
  payoff_P1.append(np.max([K[2]-stock_price_t1[i], 0]))

C_1 = np.mean(payoff_C1)*np.exp(-r)
C_2 = np.mean(payoff_C2)*np.exp(-r)
P_1 = np.mean(payoff_P1)*np.exp(-r)
print(f'{C_1 = }, {C_2 = }, {P_1 = }')
```

```
C_1 = 0.36787838580144244, C_2 = 0.30096938505212706, P_1 = 9.639077548502444
```

Metode Black-Scholes

In [ ]:
```
# Untuk opsi call 1 K=20
S_0 = stock[-1]

d1 = (np.log(S_0/K[0])+(r+1/2*sigma**2)*1)/(sigma*np.sqrt(1))
d2 = (np.log(S_0/K[0])+(r-1/2*sigma**2)*1)/(sigma*np.sqrt(1))
C1 = S_0*norm.cdf(d1, loc=0, scale=1)-K[0]*np.exp(-r*1)*norm.cdf(d2, loc=0, scale=1)

# Untuk opsi call 2 K=21
S_0 = stock[-1]

d1 = (np.log(S_0/K[1])+(r+1/2*sigma**2)*1)/(sigma*np.sqrt(1))
d2 = (np.log(S_0/K[1])+(r-1/2*sigma**2)*1)/(sigma*np.sqrt(1))
C2 = S_0*norm.cdf(d1, loc=0, scale=1)-K[1]*np.exp(-r*1)*norm.cdf(d2, loc=0, scale=1)

# Untuk opsi put K=22
S_0 = stock[-1]

d1 = (np.log(S_0/K[2])+(r+1/2*sigma**2)*1)/(sigma*np.sqrt(1))
d2 = (np.log(S_0/K[2])+(r-1/2*sigma**2)*1)/(sigma*np.sqrt(1))
P1 = K[2]*np.exp(-r*1)*norm.cdf(-d2, loc=0, scale=1)-S_0*norm.cdf(-d1, loc=0, scale=1)

print(f'{C1 = }, {C2 = }, {P1 = }')
```

```
C1 = 23.313480056115253, C2 = 23.28035076494202, P1 = 20.618120990076697
```

Untuk Periode 6 Bulan

In [ ]:
```
data_6 = ticker.history(start="2017-02-01", end="2023-02-28", interval="3mo")
data_6 = data_6[np.arange(len(data_6)) % 2 == 0]
data_6
```

Out[ ]:

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
2017-02-01 00:00:00-05:00	30.797679	31.366458	28.990971	30.237267	1467825300	0.260	0.0
2017-08-01 00:00:00-04:00	30.263823	38.869407	29.177517	38.606319	1562644600	0.273	0.0
2018-02-01 00:00:00-05:00	41.027490	47.985821	36.159239	44.399139	1985708400	0.300	0.0
2018-08-01 00:00:00-04:00	41.857771	44.069976	36.893365	40.830051	1761966400	0.300	0.0
2019-02-01 00:00:00-05:00	41.495588	52.544030	41.451498	45.004990	1317684100	0.315	0.0
2019-08-01 00:00:00-04:00	45.103380	51.102880	39.996662	50.469002	1357593200	0.315	0.0
2020-02-01 00:00:00-05:00	58.256187	61.536823	39.430927	54.207355	2020100000	0.330	0.0
2020-08-01 00:00:00-04:00	44.086837	51.357008	39.830681	40.442616	2086188800	0.330	0.0
2021-02-01 00:00:00-05:00	51.822536	63.437450	51.600239	53.285976	1937077900	0.348	0.0
2021-08-01 00:00:00-04:00	50.620540	52.570728	44.882481	45.941959	1487763900	0.348	0.0
2022-02-01 00:00:00-05:00	46.353011	49.897429	41.335712	41.421234	2245576100	0.365	0.0
2022-08-01 00:00:00-04:00	34.790179	36.372866	23.730666	27.436472	2769836000	0.365	0.0
2023-02-01 00:00:00-05:00	27.554252	30.794768	24.432294	24.629887	880323200	0.365	0.0

In [ ]:
```
tgl = data_6.index.tolist()
date = [i.date() for i in tgl]
stock = data_6["Close"]
dt = 1/1000
ret = []
for i in range(len(stock)):
    if i == 0:
        a = 0
    else:
        a = np.log(stock[i]/stock[i-1])
    ret.append(a)

tabel_saham = pd.DataFrame(list(zip(date, stock, ret)), columns=['Date', 'Close','ln(St/St-1)
tabel_saham
```

Out[ ]:

	Date	Close	ln(St/St-1)
0	2017-02-01	30.237267	0.000000
1	2017-08-01	38.606319	0.244341
2	2018-02-01	44.399139	0.139804
3	2018-08-01	40.830051	-0.083802
4	2019-02-01	45.004990	0.097355
5	2019-08-01	50.469002	0.114586
6	2020-02-01	54.207355	0.071457
7	2020-08-01	40.442616	-0.292933
8	2021-02-01	53.285976	0.275789
9	2021-08-01	45.941959	-0.148294
10	2022-02-01	41.421234	-0.103585
11	2022-08-01	27.436472	-0.411920
12	2023-02-01	24.629887	-0.107913

In [ ]:

```
mu_tilde = np.mean(ret)
var_tilde = np.var(ret)
mu = 1/dt * (mu_tilde + 1/2 * var_tilde)
sigma = np.sqrt(var_tilde/dt)
r_1 = (1+r_0)**(6/12)-1
r = np.exp(r_1*dt)-1
print(f'''Return berdistribusi N({mu_tilde}, {var_tilde})
Sehingga,
parameter drift (mu) nya adalah {mu}
parameter volatilitas (sigma) nya adalah {sigma}
Serta, risk-neutral interest rate Amerika (r) adalah {round(100*r_1,2)}% per 6 bulan''')
```

```
Return berdistribusi N(-0.015778041997122308, 0.03743553242415954)
Sehingga,
parameter drift (mu) nya adalah 2.939724214957463
parameter volatilitas (sigma) nya adalah 6.118458337208772
Serta, risk-neutral interest rate Amerika (r) adalah 1.24% per 6 bulan
```

Metode Binomial

In [ ]:

```
S_0 = stock[-1]
beta = 0.5*(np.exp(-r*dt)+np.exp((r+sigma**2)*dt))
n = 1000
u = beta+np.sqrt((beta**2)-1)
d = 1/u
p = (np.exp(r*dt)-d)/(u-d)

S = np.ones((n+1, n+1))
for i in range(1, n+2):
    for j in range(1, i+1):
        S[j-1, i-1] = S_0*u**(j-1)*d**(i-j)

C_1 = np.zeros((n+1, n+1))
C_2 = np.zeros((n+1, n+1))
P_1 = np.zeros((n+1, n+1))

for j in range(1, n+2):
```

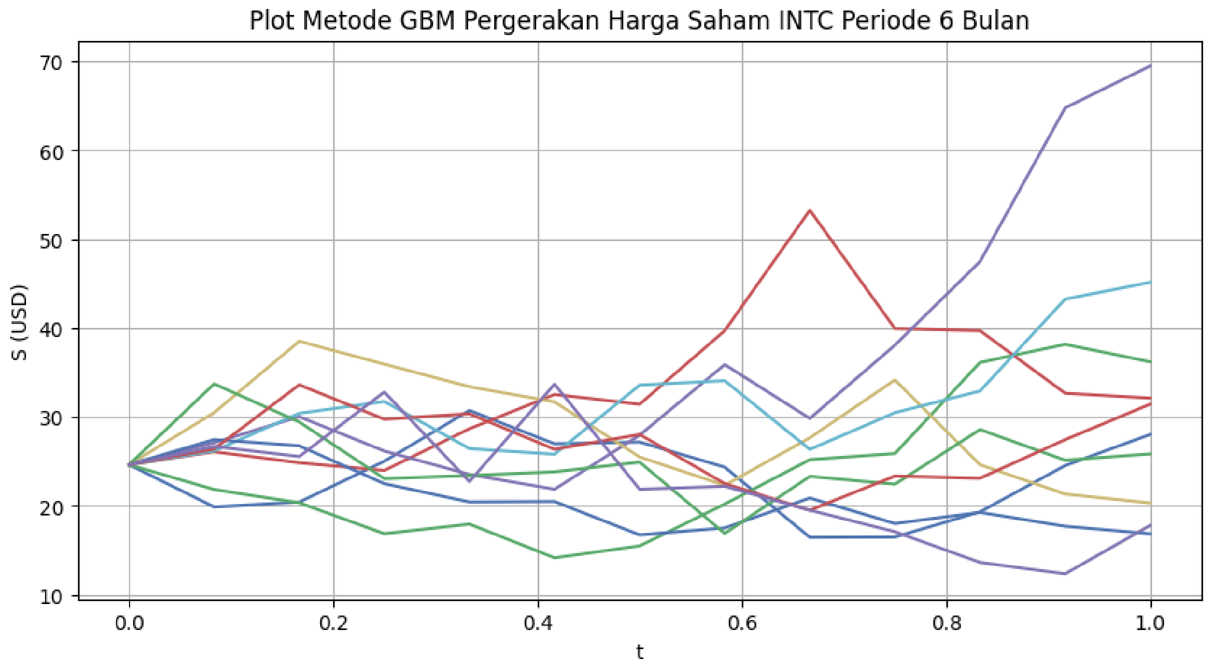
```python
    C_1[j-1, n] = max(S[j-1, n]-K[0], 0)
    C_2[j-1, n] = max(S[j-1, n]-K[1], 0)
    P_1[j-1, n] = max(K[2]-S[j-1, n], 0)

for i in range(n-1, -1, -1):
    for j in range(1, i+2):
        C_1[j-1, i] = max(S[j-1, i]-K[0], np.exp(-r*dt)*(p*C_1[j, i+1]+(1-p)*C_1[j-1, i+1]))
        C_2[j-1, i] = max(S[j-1, i]-K[1], np.exp(-r*dt)*(p*C_2[j, i+1]+(1-p)*C_2[j-1, i+1]))
        P_1[j-1, i] = max(K[2]-S[j-1, i], np.exp(-r*dt)*(p*P_1[j, i+1]+(1-p)*P_1[j-1, i+1]))

V_C1 = C_1[0, 0]
V_C2 = C_2[0, 0]
V_P1 = P_1[0, 0]
print(f'{beta = }, {u = }, {d = }')
print(f'{V_C1 = }, {V_C2 = }, {V_P1 = }')
```

```
beta = 1.0190725343512157, u = 1.2153092033736228, d = 0.8228358653288086
V_C1 = 24.584200891806496, V_C2 = 24.583056256312023, V_P1 = 21.951848529051425
```

Metode GBM

In [ ]:

```python
np.random.seed(108)

# Clear the current figure
plt.figure(figsize = (10, 5))

# Parameters
T = 1
L = 12
M = 1000

# Time steps
t = np.linspace(0, T, L+1)

# Simulate stock prices
log_returns = mu_tilde + np.sqrt(var_tilde) * np.random.randn(M, L)
cumulative_returns = np.cumprod(np.exp(log_returns), axis=1)
stock_prices = S_0 * np.hstack([np.ones((M, 1)), cumulative_returns])
stock_price_t1 = np.sort(stock_prices[:, -1])[::-1]

# Plot stock prices
plt.plot(t, stock_prices[:10].T)
plt.title(f'Plot Metode GBM Pergerakan Harga Saham {stock_code} Periode 6 Bulan')
plt.xlabel('t')
plt.ylabel(f'S ({currency})')
plt.grid()
plt.show()
```



In [ ]:
```
payoff_C1 = []
payoff_C2 = []
payoff_P1 = []
for i in range(len(stock_price_t1)):
  payoff_C1.append(np.max([stock_price_t1[i]-20, 0]))
  payoff_C2.append(np.max([stock_price_t1[i]-K[1], 0]))
  payoff_P1.append(np.max([K[2]-stock_price_t1[i], 0]))

C_1 = np.mean(payoff_C1)*np.exp(-r)
C_2 = np.mean(payoff_C2)*np.exp(-r)
P_1 = np.mean(payoff_P1)*np.exp(-r)
print(f'{C_1 = }, {C_2 = }, {P_1 = }')
```

```
C_1 = 8.003278068676394, C_2 = 7.517533531329767, P_1 = 4.728236783905952
```

Metode Black-Scholes

In [ ]:
```
# Untuk opsi call 1 K=20
S_0 = stock[-1]

d1 = (np.log(S_0/K[0])+(r+1/2*sigma**2)*1)/(sigma*np.sqrt(1))
d2 = (np.log(S_0/K[0])+(r-1/2*sigma**2)*1)/(sigma*np.sqrt(1))
C1 = S_0*norm.cdf(d1, loc=0, scale=1)-K[0]*np.exp(-r*1)*norm.cdf(d2, loc=0, scale=1)

# Untuk opsi call 2 K=21
S_0 = stock[-1]

d1 = (np.log(S_0/K[1])+(r+1/2*sigma**2)*1)/(sigma*np.sqrt(1))
d2 = (np.log(S_0/K[1])+(r-1/2*sigma**2)*1)/(sigma*np.sqrt(1))
C2 = S_0*norm.cdf(d1, loc=0, scale=1)-K[1]*np.exp(-r*1)*norm.cdf(d2, loc=0, scale=1)

# Untuk opsi put K=22
S_0 = stock[-1]

d1 = (np.log(S_0/K[2])+(r+1/2*sigma**2)*1)/(sigma*np.sqrt(1))
d2 = (np.log(S_0/K[2])+(r-1/2*sigma**2)*1)/(sigma*np.sqrt(1))
P1 = K[2]*np.exp(-r*1)*norm.cdf(-d2, loc=0, scale=1)-S_0*norm.cdf(-d1, loc=0, scale=1)

print(f'{C1 = }, {C2 = }, {P1 = }')
```

```
C1 = 24.580659954724073, C2 = 24.579433982215306, P1 = 21.94807930988397
```