

## ▼ No 2 - Opsi Put Amerika

### ▼ n Langkah

```
S0_1 = float(input('Masukkan nilai S0, jika dalam pecahan atau % mohon ubah ke bentuk desimal = '))
K_1 = float(input('Masukkan nilai K, jika dalam pecahan atau % mohon ubah ke bentuk desimal = '))
sigma_1 = float(input('Masukkan nilai sigma (per tahun), jika dalam pecahan/persen mohon ubah ke bentuk desimal = '))
r_1 = float(input('Masukkan nilai r (per tahun), jika dalam pecahan/persen mohon ubah ke bentuk desimal = '))
T_1 = float(input('Masukkan nilai T (waktu jatuh tempo, DALAM TAHUN), jika dalam pecahan/persen mohon ubah ke bentuk desimal = '))
n_1 = int(input('Masukkan nilai n (banyak langkah), jika dalam pecahan mohon bulatkan ke bilangan bulat terdekat = '))

Masukkan nilai S0, jika dalam pecahan atau % mohon ubah ke bentuk desimal = 100
Masukkan nilai K, jika dalam pecahan atau % mohon ubah ke bentuk desimal = 100
Masukkan nilai sigma (per tahun), jika dalam pecahan/persen mohon ubah ke bentuk desimal = 0.35
Masukkan nilai r (per tahun), jika dalam pecahan/persen mohon ubah ke bentuk desimal = 0.04
Masukkan nilai T (waktu jatuh tempo, DALAM TAHUN), jika dalam pecahan/persen mohon ubah ke bentuk desimal = 0.5
Masukkan nilai n (banyak langkah), jika dalam pecahan mohon bulatkan ke bilangan bulat terdekat = 100
```

### Fungsi Waktu Generik

```
import numpy as np

from functools import wraps
from time import time

def timing(f):
    @wraps(f)
    def wrap(*args, **kw):
        ts = time()
        result = f(*args, **kw)
        te = time()
        print('func:%r args:[%r, %r] took: %2.4f sec' % \
              (f.__name__, args, kw, te-ts))
        return result
    return wrap
```

### Cabang Pohon Put Amerika

```
import numpy as np

opttype = 'P' # Tipe Opsi 'C' untuk Call, dan 'P' untuk Put

delta_t_1 = T_1/n_1
u_1 = np.exp(sigma_1*(np.sqrt(delta_t_1)))
d_1 = np.exp(-sigma_1*(np.sqrt(delta_t_1)))
p_1 = ((np.exp(r_1*delta_t_1)) - d_1)/(u_1 - d_1)

print("Nilai delta t untuk metode Binomial", n_1, "langkah sebesar: ", delta_t_1)
print("Nilai parameter harga saham naik (u) adalah: ", u_1)
print("Nilai parameter harga saham turun (d) adalah: ", d_1)
print("Nilai peluang harga saham naik (p) adalah: ", p_1)

def put_amerika (K_1,T_1,S0_1,r_1,n_1,u_1,d_1,opttype='P'):
    #precompute values
    disc = np.exp(-r_1*delta_t_1)

    # Inisialisasi harga saham saat waktu jatuh tempo
    S_1 = np.zeros(n_1+1)
    for j in range(0, n_1+1):
        S_1[j] = S0_1 * u_1**j * d_1**(n_1-j)

    # PO dari Opsi
    C = np.zeros(n_1+1)
    for j in range(0, n_1+1):
        if opttype == 'P':
            C[j] = max(0, K_1 - S_1[j])
        else:
            C[j] = max(0, S_1[j] - K_1)

    # Perhitungan mundur dengan diagram cabang pohon
```

```

for i in np.arange(n_1-1,-1,-1):
    for j in range(0,i+1):
        S_1 = S0_1 * u_1**j * d_1**(i-j)
        C[j] = disc * ( p_1*C[j+1] + (1-p_1)*C[j] )
        if opttype == 'P':
            C[j] = max(C[j], K_1 - S_1)
        else:
            C[j] = max(C[j], S_1 - K_1)

return C[0]

Nilai delta t untuk metode Binomial 100 langkah sebesar: 0.005
Nilai parameter harga saham naik (u) adalah: 1.0250575294843356
Nilai parameter harga saham turun (d) adalah: 0.9755550017793236
Nilai peluang harga saham naik (p) adalah: 0.4978537332249116

n_langkah = put_amerika(K_1,T_1,S0_1,r_1,n_1,u_1,d_1,opttype='P')
print("Hasil perhitungan nilai opsi untuk metode Binomial", n_1, "langkah sebesar: ", n_langkah)

Hasil perhitungan nilai opsi untuk metode Binomial 100 langkah sebesar: 8.949710382983117

```

## ▼ No 3 - Opsi Call Eropa

### ▼ n langkah

```

import numpy as np

N = int(input('Masukkan nilai n (banyak langkah), jika dalam pecahan mohon bulatkan ke bilangan bulat terdekat = '))
t = float(input('Masukkan nilai T (waktu jatuh tempo, DALAM TAHUN), jika dalam pecahan/persen mohon ubah ke bentuk desimal = '))
t = t / (N - 1)
S0 = float(input('Masukkan nilai S0, jika dalam pecahan atau % mohon ubah ke bentuk desimal = '))
K = float(input('Masukkan nilai K, jika dalam pecahan atau % mohon ubah ke bentuk desimal = '))
r = float(input('Masukkan nilai r (per tahun), jika dalam pecahan/persen mohon ubah ke bentuk desimal = '))

sigma = float(input('Masukkan nilai sigma (per tahun), jika dalam pecahan/persen mohon ubah ke bentuk desimal = '))
u = np.exp(sigma * np.sqrt(t))
d = 1/u
p = (np.exp(r * t) - d) / (u - d)

Masukkan nilai n (banyak langkah), jika dalam pecahan mohon bulatkan ke bilangan bulat terdekat = 250
Masukkan nilai T (waktu jatuh tempo, DALAM TAHUN), jika dalam pecahan/persen mohon ubah ke bentuk desimal = 1
Masukkan nilai S0, jika dalam pecahan atau % mohon ubah ke bentuk desimal = 50
Masukkan nilai K, jika dalam pecahan atau % mohon ubah ke bentuk desimal = 50
Masukkan nilai r (per tahun), jika dalam pecahan/persen mohon ubah ke bentuk desimal = 0.05
Masukkan nilai sigma (per tahun), jika dalam pecahan/persen mohon ubah ke bentuk desimal = 0.3

print("Nilai delta t untuk metode Binomial", N, "langkah sebesar: ", t)
print("Nilai parameter harga saham naik (u) adalah: ", u)
print("Nilai parameter harga saham turun (d) adalah: ", d)
print("Nilai peluang harga saham naik (p) adalah: ", p)

Nilai delta t untuk metode Binomial 250 langkah sebesar: 0.004016064257028112
Nilai parameter harga saham naik (u) adalah: 1.019193601156315
Nilai parameter harga saham turun (d) adalah: 0.9811678555138699
Nilai peluang harga saham naik (p) adalah: 0.5005284588042681

```

```

# Membuat matriks kosong berisi harga saham dan nilai opsi call
stock_prices = np.zeros( (N, N) )
call_prices = np.zeros( (N, N) )

# Memasukkan harga saham awal ke matriks
stock_prices[0,0] = S0

# Memasukkan harga saham yang lain
for i in range(1, N ):
    M = i + 1
    stock_prices[i, 0] = d * stock_prices[i-1, 0]
    for j in range(1, M ):
        stock_prices[i, j] = u * stock_prices[i - 1, j - 1]

# Hitung nilai opsi call pada waktu jatuh tempo. Jika nilai opsi <0, akan diisi menjadi 0 pada matriks.
expiration = stock_prices[-1,:] - K
expiration.shape = (expiration.size, )

```

```
expiration = np.where(expiration >= 0, expiration, 0)

# Baris terakhir matriks berisi nilai opsi call saat waktu jatuh tempo
call_prices[-1,:] = expiration

# Perhitungan mundur dengan diagram cabang pohon
for i in range(N - 2,-1,-1):
    for j in range(i + 1):
        call_prices[i,j] = np.exp(-r * t) * ((1-p) * call_prices[i+1,j] + p * call_prices[i+1,j+1])

hrg_call = call_prices[0,0]
print("Hasil perhitungan nilai opsi untuk metode Binomial", N, "langkah sebesar: ", hrg_call)

    Hasil perhitungan nilai opsi untuk metode Binomial 250 langkah sebesar: 7.121142700538418
```

---

✓ 0s completed at 7:27 PM

