

**LAPORAN OBSERVASI ANALISIS, DESAIN, DAN IMPLEMENTASI *GENETIC ALGORITHM* TERHADAP DATA SAHAM**

Ditujukan untuk memenuhi salah satu tugas mata kuliah:  
Sistem Cerdas (**CTI-2G3**)

oleh:

Bagas Alfito Prismawan (1303193027)  
Kevin Antonio Fajrin (1303193123)  
Sultan Kautsar (1303194010)



**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS INFORMATIKA  
UNIVERSITAS TELKOM  
BANDUNG  
2021**

## **ABSTRAK**

Jual beli saham merupakan hal yang sangat menarik. Karena saham bisa membuat para investor memperoleh keuntungan yang besar namun bisa sebaliknya. Untuk mendapatkan keuntungan yang besar, investor perlu melakukan analisis dalam memprediksi harga saham. Namun, memprediksi harga saham adalah hal yang sulit dilakukan karena harga saham mengalami fluktuasi setiap waktu dengan cepat. Sehingga investor perlu memprediksi harga saham sesingkat mungkin. Salah satu teknik yang dapat dipakai untuk memprediksi adalah menggunakan pendekatan Algoritma Genetika. Algoritma Genetika sendiri memiliki ruang pencarian yang sangat luas sehingga bisa mendapatkan solusi terbaik untuk berbagai macam permasalahan. Dalam mengimplementasikan algoritma genetika ini, representasi kromosom yang digunakan adalah real coded, proses crossover yang digunakan adalah extended intermediate, random mutation pada proses mutasi dan metode seleksi replacement selection.

**Kata Kunci:** Algoritma Genetika, Prediksi Harga Saham, Regresi

## **ABSTRACT**

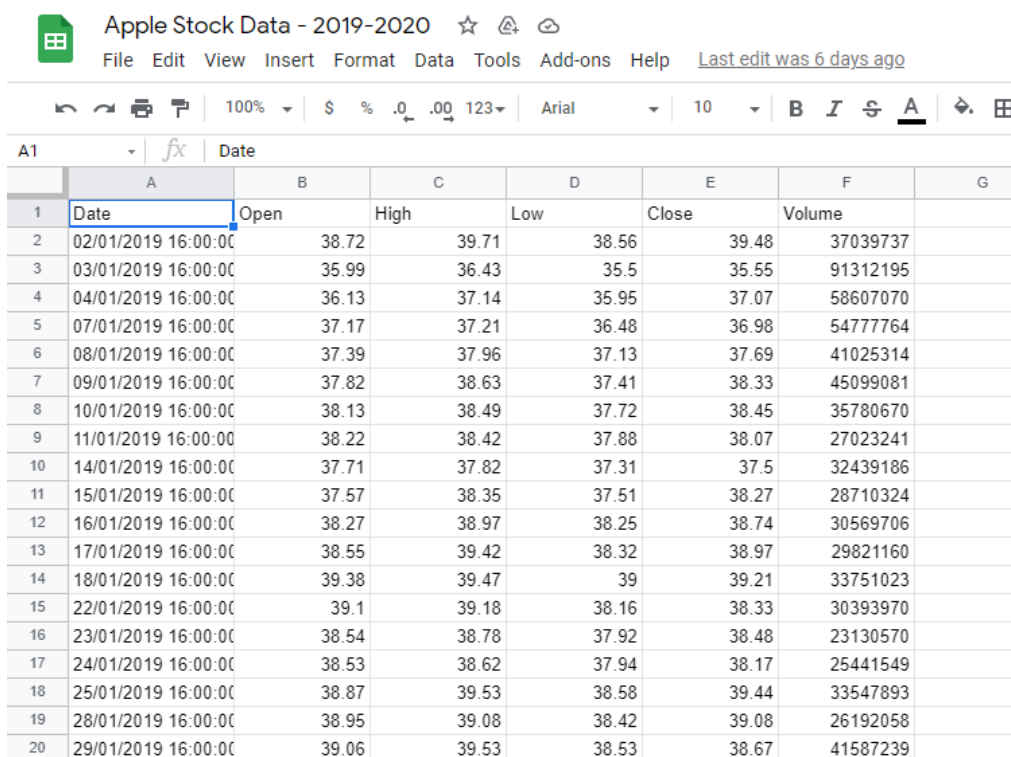
*Buying and selling stocks is a very interesting thing. Because stocks can make investors get big profits, but vice versa. To get a big profit, investors need to carry out analysis in predicting stock prices. However, predicting stock prices is a difficult thing to do because stock prices fluctuate rapidly all the time. So that investors need to predict the stock price as short as possible. One of the techniques that can be used to predict is using the Genetic Algorithm approach. The Genetic Algorithm itself has a very broad search space so that it can get the best solution for various problems. In implementing this genetic algorithm, the chromosome representation used is real coded, the crossover process used is extended intermediate, random mutation in the mutation process and the replacement selection method.*

## 1. PENDAHULUAN

Saham merupakan tanda bukti penyertaan kepemilikan modal/dana suatu perusahaan (Tambunan, 2006). Keuntungan memiliki saham sangatlah menjanjikan. Dengan memperjualbelikan saham investor bertujuan mendapatkan keuntungan yang maksimal dan resiko yang minimal. Keuntungan bisa diperoleh dari dividen (pembagian laba/keuntungan kepada para pemegang saham perusahaan), memperoleh keuntungan modal saat saham dijual kembali dengan bunga yang lebih mahal (capital gain) dan masih banyak lagi (Fahmi, 2013). Selain mendapatkan keuntungan yang besar, saham juga merupakan bisnis yang sangat beresiko dan menimbulkan kerugian (Torben, 2007). Sehingga, para investor harus berhati-hati dalam memperjual belikan sahamnya.

## 2. PERMASALAHAN

1. Menganalisis, desain, dan implementasi algoritma *Genetic Algorithm* (GA) untuk menemukan nilai konstanta dari suatu fungsi peramalan saham Apple. Stock Data Realtime: <https://finance.yahoo.com/quote/AAPL/history>.



	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Volume	
2	02/01/2019 16:00:00	38.72	39.71	38.56	39.48	37039737	
3	03/01/2019 16:00:00	35.99	36.43	35.5	35.55	91312195	
4	04/01/2019 16:00:00	36.13	37.14	35.95	37.07	58607070	
5	07/01/2019 16:00:00	37.17	37.21	36.48	36.98	54777764	
6	08/01/2019 16:00:00	37.39	37.96	37.13	37.69	41025314	
7	09/01/2019 16:00:00	37.82	38.63	37.41	38.33	45099081	
8	10/01/2019 16:00:00	38.13	38.49	37.72	38.45	35780670	
9	11/01/2019 16:00:00	38.22	38.42	37.88	38.07	27023241	
10	14/01/2019 16:00:00	37.71	37.82	37.31	37.5	32439186	
11	15/01/2019 16:00:00	37.57	38.35	37.51	38.27	28710324	
12	16/01/2019 16:00:00	38.27	38.97	38.25	38.74	30569706	
13	17/01/2019 16:00:00	38.55	39.42	38.32	38.97	29821160	
14	18/01/2019 16:00:00	39.38	39.47	39	39.21	33751023	
15	22/01/2019 16:00:00	39.1	39.18	38.16	38.33	30393970	
16	23/01/2019 16:00:00	38.54	38.78	37.92	38.48	23130570	
17	24/01/2019 16:00:00	38.53	38.62	37.94	38.17	25441549	
18	25/01/2019 16:00:00	38.87	39.53	38.58	39.44	33547893	
19	28/01/2019 16:00:00	38.95	39.08	38.42	39.08	26192058	
20	29/01/2019 16:00:00	39.06	39.53	38.53	38.67	41587239	

### **3. DASAR TEORI**

#### **3.1. Saham**

Saham adalah bukti kepemilikan nilai sebuah perusahaan atau bukti penyertaan modal. Pemilik saham juga memiliki hak untuk mendapatkan dividen sesuai dengan jumlah saham yang dimilikinya. Dengan memegang saham, maka individu maupun badan bisa mengklaim kepemilikan pada suatu perusahaan terbuka. Artinya, pemegang saham berapapun jumlah lembar yang dimilikinya berhak hadir dalam Rapat Umum Pemegang Saham (RUPS). Salah satu cara untuk memiliki saham perusahaan, seseorang harus membelinya di pasar modal.

#### **3.2. Algoritma Genetika**

Sebuah algoritma genetika (GA) adalah teknik pencarian yang digunakan dalam komputasi untuk mencari solusi yang tepat atau perkiraan untuk mencari dan masalah optimasi. Algoritma genetik adalah kelas khusus dari perhitungan evolusi yang menggunakan teknik terinspirasi oleh biologi evolusioner seperti sebagai warisan, mutasi, seleksi, dan crossover. Sebuah algoritma genetika menemukan solusi potensial untuk masalah tertentu sebagai kromosom sederhana seperti struktur data sehingga dapat menjaga informasi penting. Pelaksanaannya dimulai dengan pemilihan populasi kromosom, yang merupakan seperangkat solusi untuk masalah yang bisa terjadi untuk skenario tertentu. Satu mengevaluasi fitness dan kemudian melakukan reproduksi untuk mendapatkan solusi yang lebih baik sehubungan dengan masalah sasaran. Kromosom, yang merupakan solusi yang lebih baik, diberi lebih banyak kesempatan untuk reproduksi daripada mereka yang mewakili solusi yang lebih miskin. Proses ini berlanjut selama beberapa generasi setelah kita mendapatkan solusi yang optimal. Operator digunakan untuk percobaan ini adalah dua-titik crossover dan mutasi merayap. Crossover adalah operator genetika digunakan untuk berbagai struktur gen kromosom mana informasi gen dipertukarkan antara orang tua yang dipilih dengan memilih dua poin dalam struktur gen dari setiap orangtua.

## 4. IMPLEMENTASI

### 4.1. Regresi Linear

Dalam laporan ini, menggunakan data saham dari Apple dan akan dilakukan analisis, desain, dan implementasi algoritma *Genetic Algorithm* (GA) ke dalam suatu program komputer untuk menemukan nilai konstanta dari suatu fungsi peramalan saham, dengan asumsi fungsi Regresi Linear adalah:

$$f(x)=a_0 + a_1.y_1 + a_2.y_2 + a_3.y_3 + .... + a_{10}.y_{10}$$

Dimana:

$f(x)$	menyatakan harga saham saat ini
$a_0, a_1 \dots a_{10}$	suatu konstanta yang dicari
$y_1, y_2, \dots y_n$	nilai saham hari ke 1 sampai hari ke 10

Setelah nilai dari suatu prediksi, metode yang digunakan untuk mengevaluasi nilai fitness dari suatu individu adalah dengan menggunakan metode Mean Squared Error (MSE). MSE dihitung dengan menjumlahkan kuadrat semua kesalahan peramalan pada setiap periode dan membaginya dengan jumlah periode peramalan . Proses dari MSE itu sendiri dapat dihitung menggunakan persamaan

$$MSE = \frac{\sum (X_n - Y_n)^2}{n}$$

Kemudian kita menentukan nilai fitness dari tiap individu untuk melihat seberapa baik satu individu dengan individu lainnya, nilai fitness dapat dihitung menggunakan persamaan sebagai berikut :

$$\text{Nilai Fitness} = \frac{1}{MSE}$$

#### 4.2. Parameter Genetic Algorithm

Sekarang mengatur ukuran populasi, jumlah generasi yang kode akan berjalan, tingkat mutasi dalam kromosom, nama saham dalam bentuk ticker perusahaan, NumReturn untuk mengembalikan jumlah kromosom terbaik yaitu sebanyak 10 kromosom, jumlah hari perdagangan stok yang secara default disimpan sebagai 50 hari sesuai ketentuan.

```
PopulationSize = 500 #ukuran populasi
NumberOfGenerations = 10 #Loncatan Generasi
MutationRate = 10 #laju crossover
MutationChange = 1 #laju mutasi
NumReturn = 10 # mengambil dari 10 hari terakhir
NumberOfDay = 50 #Prediksi untuk 50 hari kedepan
```

#### 4.3. Mutasi Kromosom

Mendefinisikan fungsi untuk memutasi kromosom. Mutasi dilakukan dengan memasukkan variabel toChange, yang akan bermutasi maksimal 10 kromosom dalam iterasi dengan mean dan distribusi standar masing-masing 0,0.15.

```
# Mendefinisikan fungsi untuk memutasi kromosom.
# Mutasi dilakukan dengan memasukkan variabel toChange, yang akan bermutasi maksimal 10 kromosom.
# dalam iterasi dengan mean dan distribusi standar masing-masing 0,0.15.
def mutate(self):
    mu, sigma = 0, 0.15 # Mean dan standard deviation.
    s = numpy.random.normal(mu, sigma, 1) # edit-1.
    x = iter(s)
    toChange = random.randint(0, 5)
    if toChange == 0:
        self.buy = random.randint(0, 999) % 2
    if toChange == 1:
        self.min = next(x)
    if toChange == 2:
        self.max = next(x)
    if toChange == 3:
        self.prev_min = next(x)
    if toChange == 4:
        self.prev_max = next(x)
    if self.min > self.max:
        self.min, self.max = self.max, self.min
    if self.prev_min > self.prev_max:
        self.prev_min, self.prev_max = self.prev_max, self.prev_min
```

#### 4.4. Crossover

Menghasilkan individu atau populasi baru (anak dari populasi awal) dengan gen yang berbeda dari populasi awal. Jumlah individu yang dihasilkan tergantung pada tingkat crossover (cr) pengguna.

```
#Uniform Crossover
#The below function performs Uniform crossover in the current population and swaps if the child has a better score than the previous minimum.
def uniformCross(self, z):
    children = []
    for i in range(PopulationSize-len(self.nextGeneration)):
        child = Chromosome(0,0,0,0,0)
        chromosome1 = self.nextGeneration[random.randint(0,999999) % len(self.nextGeneration)]
        chromosome2 = self.nextGeneration[random.randint(0,999999) % len(self.nextGeneration)]
        if(random.randint(0,999) %2):
            child.min = chromosome1.min
        else:
            child.min = chromosome2.min

        if(random.randint(0,999) %2):
            child.max = chromosome1.max
        else:
            child.max = chromosome2.max

        #Check to make sure the swap yields viable chromosome
        if child.max < child.min:
            child.max, child.min = child.min, child.max

        if(random.randint(0,999) %2):
            child.prev_min = chromosome1.prev_min
        else:
            child.prev_min = chromosome2.prev_min

        if(random.randint(0,999) %2):
            child.prev_max = chromosome1.prev_max
        else:
            child.prev_max = chromosome2.prev_max

        #Check if swap is needed
        if child.prev_max < child.prev_min:
            child.prev_max, child.prev_min = child.prev_min, child.prev_max

        if(random.randint(0,999) %2):
            child.buy = chromosome1.buy
        else:
            child.buy = chromosome2.buy

        #Append
        children.append(child)
```

#### 4.5. Training Data

Untuk menghasilkan data pelatihan dan melatih algoritma. Daftar populasi, kromosom generasi berikutnya, dayChange untuk perubahan harga saham pada suatu hari. Lalu hari berikutnya adalah perubahan pada hari berikutnya, keuntungan adalah perubahan positif yang terjadi saat membeli saham jika perubahannya meningkat secara positif dari hari ke hari.

```
class TrainingData(object):
    population = []
    nextGeneration = []
    dayChange = []
    nextDayChange = []
    profit = []

    def __init__(self, stockName='', popSize=None, mRate=None, mChange=None):
        self.stockName = stockName
        self.popSize = popSize
        self.mRate = mRate
        self.mChange = mChange

    # Generate Data from chosen stock
    # Here in the below code, the following generates the data from yahoo finance to be specific
    # and leaves behind the dividend column for the analysis
    def generateData(self):
        global DataSize

        # Download the data from yahoo finance for last 365 days
        data = []
        url = "https://finance.yahoo.com/quote/" + Stock_name + "/history/"
        rows = bs(urllib2.urlopen(url).read(), "lxml").findAll('table')[0].tbody.findAll('tr')
```



#### 4.6. Generate Data

Hasilkan Data dari saham yang dipilih. Berikut ini untuk menghasilkan data dari yahoo finance untuk lebih spesifik, dan meninggalkan kolom dividen untuk analisis.

```
def generateData(self):
    global DataSize

    # Download the data from yahoo finance for last 365 days
    data = []
    url = "https://finance.yahoo.com/quote/" + Stock_name + "/history/"
    rows = bs(urllib2.urlopen(url).read(), "lxml").findAll('table')[0].tbody.findAll('tr')

    for each_row in rows:
        divs = each_row.findAll('td')
        if divs[1].span.text != 'Dividend': # Ignore this row in the table
            # I'm only interested in 'Open' price; For other values, play with divs[1 - 5]
            data.append({'open': divs[1].span.text, 'Adj close': float(divs[5].span.text.replace(',', ''))})
    var = data[:NumberOfDay]
    print(data)

    file = open('stock_data', 'w')
    closes = [c['Adj close'] for c in data]
    opens = [o['open'] for o in data]
    oArray = []
    cArray = []

    for c in closes:
        cArray.append(c)

    for o in opens:
        oArray.append(o)

    for x in range(len(data) - 2):
        # %Difference, Next Day %Difference, Money Made Holding for a Day
        file.write(str((float(cArray[x]) - float(oArray[x + 1])) / 100) + ' ' + str(
            (float(cArray[x + 1]) - float(oArray[x + 2])) / 100) + ' ' + str(
                (float(oArray[x]) - float(oArray[x + 1])) + '\n')))
```

#### 4.7. Inisiasi Populasi

Menginisialisasi populasi kromosom acak Populasi kromosom berasal dari seragam acak distribusi normal dengan mean 0 dan deviasi standar 0,15. Diberikan kode diatas, menginisialisasi populasi. Buat N Kromosom dengan N menjadi Ukuran Populasi. Setiap variabel Kromosom diberi nomor dari distribusi normal dengan mean 0 dan Standar Deviasi 1,5.

```
def populationInit(self):  
  
    # Buat N Kromosom dengan N menjadi Ukuran Populasi  
    # Setiap variabel Kromosom diberi nomor dari distribusi normal  
    # dengan mean 0 dan Standar Deviasi 1,5  
  
    mu, sigma = 0, 0.15 # mean dan standard deviation  
    s = numpy.random.normal(mu, sigma, 4 * PopulationSize)  
    x = iter(s)  
    for i in range(PopulationSize):  
        temp = Chromosome(next(x), next(x), next(x), next(x), random.randint(0, 999) % 2, 0)  
  
        # Jika minimum diberi nilai lebih tinggi dari max swap mereka, ehingga masuk akal.  
        if temp.min > temp.max:  
            temp.min, temp.max = temp.max, temp.min  
        if temp.prev_min > temp.prev_max:  
            temp.prev_min, temp.prev_max = temp.prev_max, temp.prev_min  
  
    # Push Chromosome ke population array.  
    self.population.append(temp)]
```

#### 4.8. Fungsi Fitness

Menentukan skor untuk setiap kromosom dalam `self.population` Tugas yang sulit adalah memutuskan apa yang bisa menjadi Fungsi kebugaran atau skor, dihitung seperti perubahan harga hari ini lebih dari sebelumnya minimum, perubahan hari ini kurang dari maksimum sebelumnya juga jika perubahan hari berikutnya lebih dari minimum dan kurang dari maksimum hari ini. Jadi, jika kondisi ini memenuhi dan jika beli memiliki nilainya sama dengan 1, maka itu akan menambah keuntungan ke dalam skor jika tidak dikurangi darinya dan jika tidak satupun dari kasus ini memenuhi maka skor diberikan sebagai 0 untuk memulai.

```
def fitnessFunction(self):
    for i in range(len(self.population)):
        match = False
        for j in range(DataSize):

            #print(self.population[i].min, self.nextDayChange[j], self.population[i].max)
            #If match is found we BUY
            if(self.population[i].prev_min < self.dayChange[j] and self.population[i].prev_max > self.dayChange[j]):
                if(self.population[i].min < self.nextDayChange[j] and self.population[i].max > self.nextDayChange[j]):
                    if(self.population[i].buy == 1):
                        match = True
                        self.population[i].score += self.profit[j]

            #Match is found and we short
            if(self.population[i].prev_min < self.dayChange[j] and self.population[i].prev_max > self.dayChange[j]):
                if(self.population[i].min < self.nextDayChange[j] and self.population[i].max > self.nextDayChange[j]):
                    if(self.population[i].buy == 0):
                        match = True
                        self.population[i].score -= self.profit[j]

            #We have not found any matches = 0
            if match == False:
                self.population[i].score = 0
        #print(self.population[i].score)
```

#### 4.9. Pemilihan Pilihan Acak Berbobot

Pemilihan pilihan acak berbobot Fungsi di atas menghitung nilai kebugaran suatu populasi dan kemudian secara acak memilih nilai (pilih) dari kisaran seragam (0, maks (skor)), di mana maks (score adalah skor dari total kromosom digabungkan dalam populasi dan di putaran berikutnya ia memilih skor populasi dan menambahkannya variabel 'saat ini' dan jika mata uang melebihi 'pilih', maka kita pergi ke generasi berikutnya dan menambahkan populasi saat ini di generasi berikutnya [].

```
def weighted_random_choice(self):
    self.fitnessFunction()
    max = self.population[0].score
    for i in self.population[1:]:
        max += i.score
    pick = random.uniform(0, max)
    current = 0
    for i in range(len(self.population)):
        current += self.population[i].score
        if current > pick:
            self.nextGeneration.append(self.population[i])
```

#### 4.10. Penghapusan Indeks self.population Skor Nol

Menghapus indeks di self.population yang tidak memiliki skor.

```
def exists(self):
    i = 0
    while i < len(self.population):
        if self.population[i].score is None:
            del self.population[i]
        else:
            i += 1
```

#### 4.11. Uniform Crossover

Uniform Crossover Fungsi di bawah ini melakukan persilangan Uniform dalam populasi saat ini dan bertukar jika anak memiliki skor lebih baik dari minimum sebelumnya.

```
def uniformCross(self, z):
    global i
    children = []
    for i in range(PopulationSize - len(self.nextGeneration)):
        child = Chromosome(0, 0, 0, 0, 0)
        chromosome1 = self.nextGeneration[random.randint(0, 999999) % len(self.nextGeneration)]
        chromosome2 = self.nextGeneration[random.randint(0, 999999) % len(self.nextGeneration)]
        if (random.randint(0, 999) % 2):
            child.min = chromosome1.min
        else:
            child.min = chromosome2.min

        if (random.randint(0, 999) % 2):
            child.max = chromosome1.max
        else:
            child.max = chromosome2.max

        # Check to make sure the swap yields viable chromosome
        if child.max < child.min:
            child.max, child.min = child.min, child.max

        if (random.randint(0, 999) % 2):
            child.prev_min = chromosome1.prev_min
        else:
            child.prev_min = chromosome2.prev_min

        if (random.randint(0, 999) % 2):
            child.prev_max = chromosome1.prev_max
        else:
            child.prev_max = chromosome2.prev_max
```

#### 4.12. Print Kromosom

Cetak skor kromosom Fungsi cetak kromosom beserta semua kromosomnya memilih nilai NumReturn teratas darinya karena mereka sudah diurutkan menurut skor mereka. Dan memberi saran apakah akan long (beli) saham atau short (jual).

```
def printChromosomes(self):
    buyRec = []
    shortRec = []
    for i in range(len(self.population)):
        if (self.population[i].buy == 1):
            buyRec.append(self.population[i])
        if (self.population[i].buy == 0):
            shortRec.append(self.population[i])

    print("The Best %d Chromosomes When Buying" % (NumReturn))
    outputBuy = []
    outputShort = []
    fieldnames = ["Score"]
    i = 1
    size = len(buyRec)
    while i < NumReturn + 1:
        index = size - i
        print("min: %f | max: %f | previous min: %f | previous max: %f | score: %f" % (
            buyRec[index].min, buyRec[index].max, buyRec[index].prev_min, buyRec[index].prev_max,
            buyRec[index].score))
        outputBuy.append(buyRec[index].score)
        i += 1

    print("The Best %d Chromosomes When Shorting" % (NumReturn))
    i = 1
    size = len(shortRec)
    while i < NumReturn + 1:
        index = size - i
        print("min: %f | max: %f | previous min: %f | previous max: %f | score: %f" % (
            shortRec[index].min, shortRec[index].max, shortRec[index].prev_min, shortRec[index].prev_max,
```

#### 4.13. Mengambil Data Historikal

Berfungsi untuk mendapatkan data historis selama beberapa hari untuk saham tertentu.

```
def get_historical_data(name, number_of_days):
    data = []
    url = "https://finance.yahoo.com/quote/" + name + "/history/"
    rows = bs(urllib2.urlopen(url).read(), "lxml").findAll('table')[0].tbody.findAll('tr')

    for each_row in rows:
        divs = each_row.findAll('td')
        if divs[1].span.text != 'Dividend': # Abaikan baris ini pada tabel
            # I'm only interested in 'Open' price; For other values, play with divs[1 - 5]
            data.append({'Date': divs[0].span.text, 'Adj close': float(divs[1].span.text.replace(',', ''))})
    # data.append({'open': divs[1].span.text, 'Adj close': float(divs[5].span.text.replace(',', ''))})
    return data[:number_of_days]
```

## 5. HASIL DAN PENGUJIAN

Untuk mendapatkan nilai prediksi yang maksimal atau mendekati dengan data aktual, maka diperlukan pengujian sebelumnya untuk parameter kontrol algoritma genetika, karena pada dasarnya parameter kontrol algoritma genetika itu sendiri akan berbeda-beda tergantung dengan permasalahan yang dihadapi. Pada Penelitian ini, pengujian akan dilakukan untuk setiap elemen dari parameter kontrol algoritma genetika.

### 5.1. Pengujian Terhadap Ukuran Populasi

Pada pengujian ini diberikan nilai probabilitas crossover sebesar 0.10, nilai probabilitas mutasi sebesar 0.1, dengan jumlah generasi sebanyak 100. Jumlah individu yang diuji mulai dari 10 hingga 100, dengan 3 kali pengujian maka hasil yang didapatkan adalah

```
[{'open': '129.80', 'Adj close': 133.0}, {'open': '128.95', 'Adj close': 130.36}, {'open': '125.83', 'Adj close': 127.9}, {'open': '126.50', 'Adj close': 126.21}, {'open': '123.87', 'Adj close': 125.9}, {'open': '123.66', 'Adj close': 123.0}, {'open': '121.65', 'Adj close': 122.15}, {'open': '120.11', 'Adj close': 119.9}, {'open': '121.65', 'Adj close': 121.39}, {'open': '120.35', 'Adj close': 121.21}, {'open': '119.54', 'Adj close': 120.59}, {'open': '122.82', 'Adj close': 120.09}, {'open': '123.33', 'Adj close': 122.54}, {'open': '120.33', 'Adj close': 123.39}, {'open': '119.90', 'Adj close': 119.99}, {'open': '122.88', 'Adj close': 120.53}, {'open': '124.05', 'Adj close': 124.76}, {'open': '125.70', 'Adj close': 125.57}, {'open': '121.41', 'Adj close': 123.99}, {'open': '120.40', 'Adj close': 121.03}, {'open': '122.54', 'Adj close': 121.96}, {'open': '121.69', 'Adj close': 119.98}, {'open': '119.03', 'Adj close': 121.09}, {'open': '120.93', 'Adj close': 116.36}, {'open': '120.98', 'Adj close': 121.42}, {'open': '121.75', 'Adj close': 120.13}, {'open': '124.81', 'Adj close': 122.06}, {'open': '128.41', 'Adj close': 125.12}, {'open': '123.75', 'Adj close': 127.79}, {'open': '122.59', 'Adj close': 121.26}, {'open': '124.68', 'Adj close': 120.99}, {'open': '124.94', 'Adj close': 125.35}, {'open': '123.76', 'Adj close': 125.86}, {'open': '128.01', 'Adj close': 126.0}, {'open': '130.24', 'Adj close': 129.87}, {'open': '129.20', 'Adj close': 129.71}, {'open': '131.25', 'Adj close': 130.84}, {'open': '135.49', 'Adj close': 133.19}, {'open': '134.35', 'Adj close': 135.37}, {'open': '135.90', 'Adj close': 135.13}, {'open': '136.48', 'Adj close': 135.39}, {'open': '136.62', 'Adj close': 136.01}, {'open': '136.03', 'Adj close': 136.91}, {'open': '137.35', 'Adj close': 136.76}, {'open': '136.30', 'Adj close': 137.18}, {'open': '135.76', 'Adj close': 133.74}, {'open': '135.73', 'Adj close': 134.79}, {'open': '133.75', 'Adj close': 133.94}, {'open': '135.83', 'Adj close': 131.76}, {'open': '139.52', 'Adj close': 136.89}, {'open': '143.43', 'Adj close': 141.85}, {'open': '143.60', 'Adj close': 142.95}, {'open': '143.07', 'Adj close': 142.71}, {'open': '136.28', 'Adj close': 138.86}, {'open': '133.80', 'Adj close': 136.67}, {'open': '128.66', 'Adj close': 131.83}, {'open': '127.78', 'Adj close': 127.64}, {'open': '128.78', 'Adj close': 126.95}, {'open': '130.80', 'Adj close': 128.72}, {'open': '128.76', 'Adj close': 130.69}, {'open': '128.50', 'Adj close': 128.61}, {'open': '129.19', 'Adj close': 128.79}, {'open': '132.43', 'Adj close': 131.85}, {'open': '128.36', 'Adj close': 130.72}, {'open': '127.72', 'Adj close': 126.41}, {'open': '128.89', 'Adj close': 130.81}, {'open': '133.52', 'Adj close': 129.22}, {'open': '134.08', 'Adj close': 132.49}, {'open': '135.58', 'Adj close': 133.52}, {'open': '138.05', 'Adj close': 134.67}, {'open': '133.99', 'Adj close': 136.49}, {'open': '131.32', 'Adj close': 131.77}, {'open': '132.16', 'Adj close': 130.76}, {'open': '131.61', 'Adj close': 131.68}, {'open': '125.02', 'Adj close': 128.04}, {'open': '128.96', 'Adj close': 126.47}, {'open': '128.90', 'Adj close': 128.51}, {'open': '127.41', 'Adj close': 127.62}, {'open': '124.34', 'Adj close': 127.69}, {'open': '122.60', 'Adj close': 121.6}, {'open': '122.43', 'Adj close': 122.23}, {'open': '120.50', 'Adj close': 123.06}, {'open': '124.53', 'Adj close': 121.6}, {'open': '124.37', 'Adj close': 124.19}, {'open': '122.31', 'Adj close': 123.57}, {'open': '122.60', 'Adj close': 122.07}, {'open': '123.52', 'Adj close': 122.76}, {'open': '122.02', 'Adj close': 122.9}, {'open': '121.01', 'Adj close': 122.54}, {'open': '116.97', 'Adj close': 118.87}, {'open': '116.57', 'Adj close': 116.42}, {'open': '115.55', 'Adj close': 115.86}, {'open': '113.91', 'Adj close': 115.0}, {'open': '117.18', 'Adj close': 113.68}, {'open': '118.64', 'Adj close': 117.16}, {'open': '117.59', 'Adj close': 118.46}, {'open': '118.61', 'Adj close': 117.85}, {'open': '119.55', 'Adj close': 119.21}, {'open': '118.92', 'Adj close': 120.12}]
```



## 5.2. Pengujian untuk mendapatkan Kromosom Terbaik

Program mengambil kromosom terbaik untuk sepuluh kromosom. Di mana diukur dalam selisih atau *range* nilai minimum dan maksimumnya. Jika *range* berdekatan maka semakin baik untuk kromosom tersebut. Lalu diberikannya *score* untuk mengurutkan mulai kromosom terbaik pada urutan pertama hingga urutan ke sepuluh.

The Best 10 Chromosomes When Buying				
min: -0.185149	max: 0.071623	previous min: -0.000609	previous max: 0.082818	score: 174.180000
min: -0.102101	max: 0.075452	previous min: -0.000793	previous max: 0.095693	score: 174.180000
min: -0.022848	max: 0.170493	previous min: -0.013293	previous max: 0.199068	score: 172.320000
min: 0.009403	max: 0.138991	previous min: -0.180344	previous max: 0.162746	score: 158.310000
min: 0.009403	max: 0.138991	previous min: -0.180344	previous max: 0.162746	score: 158.310000
min: -0.011073	max: 0.072587	previous min: -0.015111	previous max: 0.158679	score: 141.860000
min: -0.083139	max: 0.130752	previous min: -0.015769	previous max: 0.194086	score: 140.740000
min: -0.006290	max: 0.056289	previous min: -0.246298	previous max: 0.110883	score: 132.360000
min: -0.006290	max: 0.056289	previous min: -0.246298	previous max: 0.110883	score: 132.360000
min: -0.113056	max: 0.260033	previous min: 0.015908	previous max: 0.221060	score: 128.920000
The Best 10 Chromosomes When Shorting				
min: -0.206269	max: 0.197615	previous min: -0.132466	previous max: -0.005291	score: 76.950000
min: -0.104859	max: 0.049029	previous min: -0.305196	previous max: -0.006392	score: 75.260000
min: -0.188802	max: 0.028025	previous min: -0.278741	previous max: 0.008950	score: 69.320000
min: -0.050812	max: 0.101227	previous min: -0.185170	previous max: 0.004791	score: 67.860000
min: -0.190900	max: 0.000702	previous min: -0.181074	previous max: 0.002776	score: 61.310000
min: -0.128049	max: 0.042645	previous min: -0.153235	previous max: -0.015489	score: 60.120000
min: -0.128049	max: 0.042645	previous min: -0.153235	previous max: -0.015489	score: 60.120000
min: -0.321812	max: 0.040065	previous min: -0.096208	previous max: -0.017395	score: 59.180000
min: -0.103114	max: 0.100737	previous min: -0.077651	previous max: -0.017558	score: 57.470000
min: -0.042013	max: 0.128564	previous min: -0.094671	previous max: 0.001364	score: 51.390000

## 5.3. Pengujian Probabilitas Mutasi dan Crossover

Pada pengujian ini diberikan jumlah individu sebanyak 100, nilai probabilitas mutasi sebesar 0.1, dengan jumlah generasi sebanyak 100. Nilai probabilitas crossover yang diuji mulai dari 0.5 sampai dengan 0.10, dengan 5 kali pengujian pada setiap kelipatannya. Hasil yang didapat ialah :

[174.18000000000018, 174.18000000000018, 172.3200000000001, 158.30999999999995, 158.30999999999995, 141.86000000000013, 140.74000000000012, 132.36000000000013, 132.36000000000013, 128.92000000000007]

#### 5.4. Kromosom Terbaik

Kromosom dikategorikan terbaik ketika *score* yang didapatkan oleh kromosom itu sendiri bernilai tinggi dibanding kromosom yang lainnya. Untuk kromosom terbaik diambil sepuluh dengan terurut descending sesuai dengan *score* yang didapat.

```
[{"Date": "Apr 09, 2021", "Adj close": 311.4}, {"Date": "Apr 08, 2021", "Adj close": 314.85}, {"Date": "Apr 07, 2021", "Adj close": 306.34}, {"Date": "Apr 06, 2021", "Adj close": 308.84}, {"Date": "Apr 05, 2021", "Adj close": 300.89}, {"Date": "Apr 01, 2021", "Adj close": 298.4}, {"Date": "Mar 31, 2021", "Adj close": 289.99}, {"Date": "Mar 30, 2021", "Adj close": 289.83}, {"Date": "Mar 29, 2021", "Adj close": 285.77}, {"Date": "Mar 26, 2021", "Adj close": 278.3}, {"Date": "Mar 25, 2021", "Adj close": 280.98}, {"Date": "Mar 24, 2021", "Adj close": 291.0}, {"Date": "Mar 23, 2021", "Adj close": 293.15}, {"Date": "Mar 22, 2021", "Adj close": 290.45}, {"Date": "Mar 19, 2021", "Adj close": 281.22}, {"Date": "Mar 18, 2021", "Adj close": 279.87}, {"Date": "Mar 17, 2021", "Adj close": 275.71}, {"Date": "Mar 16, 2021", "Adj close": 276.08}, {"Date": "Mar 15, 2021", "Adj close": 269.08}, {"Date": "Mar 12, 2021", "Adj close": 269.14}, {"Date": "Mar 11, 2021", "Adj close": 268.11}, {"Date": "Mar 10, 2021", "Adj close": 268.64}, {"Date": "Mar 09, 2021", "Adj close": 261.18}, {"Date": "Mar 08, 2021", "Adj close": 265.55}, {"Date": "Mar 05, 2021", "Adj close": 260.67}, {"Date": "Mar 04, 2021", "Adj close": 256.47}, {"Date": "Mar 03, 2021", "Adj close": 260.29}, {"Date": "Mar 02, 2021", "Adj close": 265.78}, {"Date": "Mar 01, 2021", "Adj close": 260.82}, {"Date": "Feb 26, 2021", "Adj close": 256.47}, {"Date": "Feb 25, 2021", "Adj close": 262.3}, {"Date": "Feb 24, 2021", "Adj close": 262.33}, {"Date": "Feb 23, 2021", "Adj close": 259.5}, {"Date": "Feb 22, 2021", "Adj close": 257.95}, {"Date": "Feb 19, 2021", "Adj close": 269.86}, {"Date": "Feb 18, 2021", "Adj close": 269.57}, {"Date": "Feb 17, 2021", "Adj close": 271.24}, {"Date": "Feb 16, 2021", "Adj close": 270.8}, {"Date": "Feb 12, 2021", "Adj close": 270.52}, {"Date": "Feb 11, 2021", "Adj close": 271.89}, {"Date": "Feb 10, 2021", "Adj close": 272.45}, {"Date": "Feb 09, 2021", "Adj close": 266.44}, {"Date": "Feb 08, 2021", "Adj close": 268.75}, {"Date": "Feb 05, 2021", "Adj close": 266.8}, {"Date": "Feb 04, 2021", "Adj close": 267.01}, {"Date": "Feb 03, 2021", "Adj close": 265.62}, {"Date": "Feb 02, 2021", "Adj close": 264.0}, {"Date": "Feb 01, 2021", "Adj close": 259.52}, {"Date": "Jan 29, 2021", "Adj close": 265.3}, {"Date": "Jan 28, 2021", "Adj close": 277.18}, {"Date": "Jan 27, 2021", "Adj close": 282.53}, {"Date": "Jan 26, 2021", "Adj close": 278.14}, {"Date": "Jan 25, 2021", "Adj close": 278.14}, {"Date": "Jan 22, 2021", "Adj close": 272.01}, {"Date": "Jan 21, 2021", "Adj close": 269.26}, {"Date": "Jan 20, 2021", "Adj close": 268.93}, {"Date": "Jan 19, 2021", "Adj close": 256.9}, {"Date": "Jan 15, 2021", "Adj close": 247.9}, {"Date": "Jan 14, 2021", "Adj close": 253.4}, {"Date": "Jan 13, 2021", "Adj close": 251.55}, {"Date": "Jan 12, 2021", "Adj close": 256.63}, {"Date": "Jan 11, 2021", "Adj close": 260.48}, {"Date": "Jan 08, 2021", "Adj close": 268.31}, {"Date": "Jan 07, 2021", "Adj close": 265.9}, {"Date": "Jan 06, 2021", "Adj close": 262.0}, {"Date": "Jan 05, 2021", "Adj close": 268.29}, {"Date": "Jan 04, 2021", "Adj close": 274.78}, {"Date": "Dec 31, 2020", "Adj close": 272.0}, {"Date": "Dec 30, 2020", "Adj close": 277.95}, {"Date": "Dec 29, 2020", "Adj close": 276.95}, {"Date": "Dec 28, 2020", "Adj close": 268.74}, {"Date": "Dec 24, 2020", "Adj close": 268.88}, {"Date": "Dec 23, 2020", "Adj close": 266.89}, {"Date": "Dec 22, 2020", "Adj close": 271.5}, {"Date": "Dec 21, 2020", "Adj close": 272.98}, {"Date": "Dec 18, 2020", "Adj close": 275.77}, {"Date": "Dec 17, 2020", "Adj close": 277.07}, {"Date": "Dec 16, 2020", "Adj close": 274.76}, {"Date": "Dec 15, 2020", "Adj close": 274.83}, {"Date": "Dec 14, 2020", "Adj close": 273.37}, {"Date": "Dec 11, 2020", "Adj close": 274.53}, {"Date": "Dec 10, 2020", "Adj close": 275.54}, {"Date": "Dec 09, 2020", "Adj close": 283.66}, {"Date": "Dec 08, 2020", "Adj close": 286.01}, {"Date": "Dec 07, 2020", "Adj close": 279.19}, {"Date": "Dec 04, 2020", "Adj close": 280.3}, {"Date": "Dec 03, 2020", "Adj close": 286.25}, {"Date": "Dec 02, 2020", "Adj close": 285.36}, {"Date": "Dec 01, 2020", "Adj close": 279.16}, {"Date": "Nov 30, 2020", "Adj close": 276.03}, {"Date": "Nov 27, 2020", "Adj close": 277.39}, {"Date": "Nov 25, 2020", "Adj close": 278.14}, {"Date": "Nov 24, 2020", "Adj close": 268.49}, {"Date": "Nov 23, 2020", "Adj close": 270.89}, {"Date": "Nov 20, 2020", "Adj close": 272.56}, {"Date": "Nov 19, 2020", "Adj close": 271.02}, {"Date": "Nov 18, 2020", "Adj close": 274.52}, {"Date": "Nov 17, 2020", "Adj close": 277.68}, {"Date": "Nov 16, 2020", "Adj close": 275.05}, {"Date": "Nov 13, 2020", "Adj close": 277.72}]
```

#### 6. Aktivitas Kontribusi

Nama	Kontribusi
Bagas Alfito Prismawan	Menyusun sebagian implementasi, menyusun pengujian terhadap ukuran populasi, Menyusun pengujian Probabilitas Mutasi dan Crossover, mencari referensi kodingan untuk pengkodean kromosom.
Kevin Antonia Fajrin	Menyusun Pendahuluan, Menyusun Permasalahan, Mencari data, Menyusun sebagian dari implementasi, Menyusun sebagian dari hasil dan pengujian.
Sultan Kautsar	Menyusun Abstrak; Menyusun dasar teori dari Saham dan Algoritma Genetika; Menyusun sebagian dari implementasi; Menyusun pengujian untuk mendapatkan kromosom terbaik.

#### 7. Link Video Presentasi

YouTube Link: <https://youtu.be/qFAhO-bg33c>