Nama : L Yasril Imam
Nim : 201710370311051
Matkul : Simulasi data

1. Bangkitkan 500 bilangan acak dengan menggunakan metode linear congruent generator dengan a = 8, c = 47, m = 100, dan X 0 = 27

   a. Kerjakan perhitungan manual dengan excel

	A	B	C	D
1	NO	Xn	nilai U	
2	1	27	00.27	
3	2	63	0,04375	
4	3	51	00.51	
5	4	55	00.55	
6	5	87	0,060416667	
7	6	43	00.43	
8	7	91	0,063194444	
9	8	75	0,052083333	
10	9	47	00.47	
11	10	23	00.23	
12	11	31	00.31	
13	12	95	0,065972222	
14	13	7	00.07	
15	14	3	00.03	
16	15	71	0,049305556	
17	16	15	00.15	
18	17	67	0,046527778	
19	18	83	0,057638889	
20	19	11	00.11	
21	20	35	00.35	
22	21	27	00.27	
23	22	63	0,04375	

   b. Kerjakan perhitungan dengan python/ R/ matlab

   Sourcode :

```
import random as rd
import numpy as np
import pandas as pd
import csv
import matplotlib.pyplot as plt
import collections
from scipy.stats import norm
from scipy.stats import uniform
import seaborn as sns
from math import ceil, floor, sqrt

a=8
c=47
m=100
x = [27]
U =[]
total =[]
head = [["NO","Xn","nilai U"]]
```

```
with open("UASPython.csv", "w", newline="") as f:
        writer = csv.writer(f)
        writer.writerows(head)
        for iterasi in range(500):
            total.append([])
            for i in range(1):
                data=[]
                no = iterasi+1
                data.append(no)
                data.append(x[-1])
                U.append(x[-1]/m)
                data.append(U[-1])
                x.append((a*x[-1]+c)%m)

                print(data)
                total[iterasi].append(data)
                writer.writerows(total[iterasi])

df = pd.read_csv("pythonnomor2.csv", encoding ='utf-8')
df.head(5)
```

:

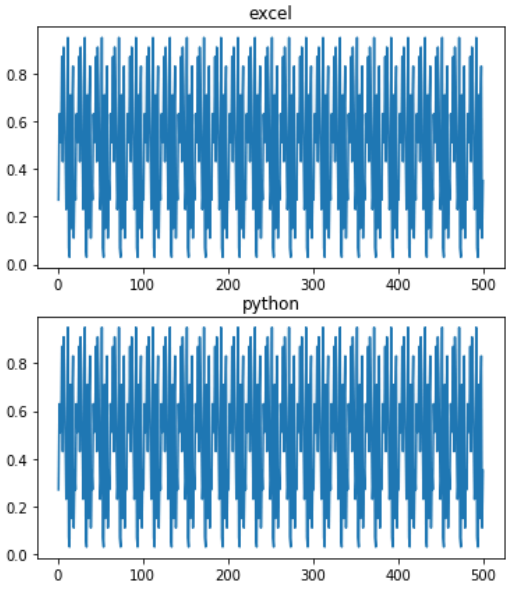
	NO	Xn	nilai U
0	1	27	0.27
1	2	63	0.63
2	3	51	0.51
3	4	55	0.55
4	5	87	0.87

```
df2 = pd.read_excel("pythonuas1.csv", encoding ='utf-8')
df2.head(5)
```

	no	X0	nilai U
0	1	27	0.27
1	2	63	0.63
2	3	51	0.51
3	4	55	0.55
4	5	87	0.87

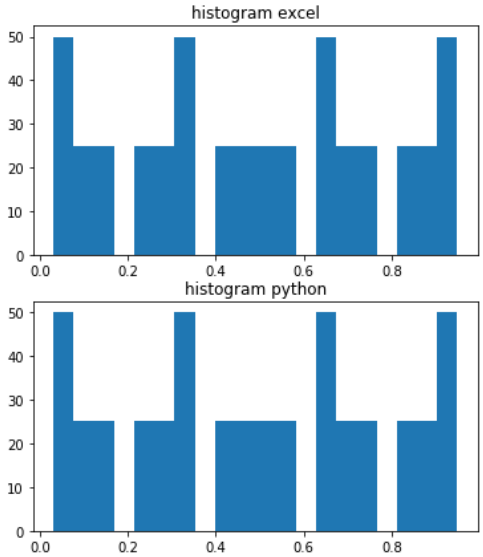
c. Buat plot garis (grafik) dari kedua hasil tsb

```
f,ax = plt.subplots(2,1,figsize=(6,7))
ax[0].plot(df['NO'],df['nilai U'])
ax[0].set_title("excel")
ax[1].plot(df2['no'],df2['nilai U'])
ax[1].set_title("python")
```

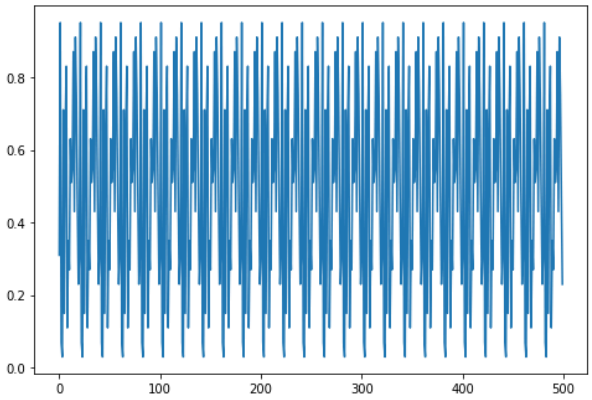


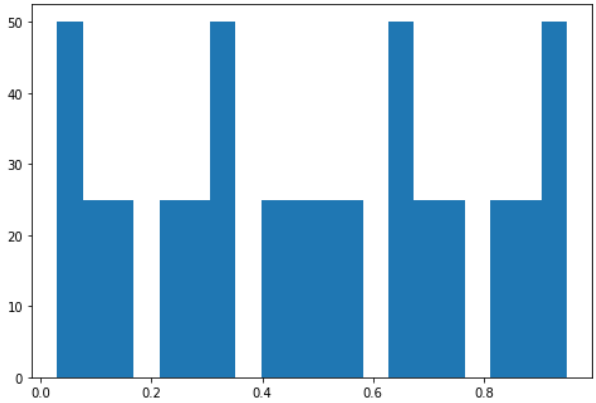
d. Buat histogram dari kedua hasil tsb

```
f,ax = plt.subplots(2,1,figsize=(6,7))
ax[0].hist(df['nilai U'],bins=20)
ax[0].set_title("histogram excel")
ax[1].hist(df2['nilai U'],bins=20)
ax[1].set_title("histogram python")
```



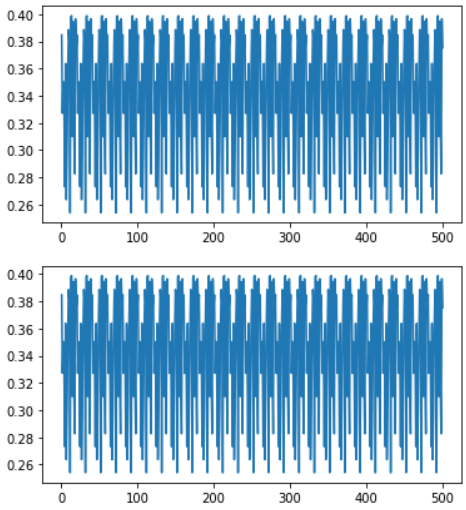
```
def seedLCG(initVal):
    global rand
    rand = initVal

def lcg(a,c,m):
    global rand
    rand = (a*rand + c) % m
    return rand/float(m)

def my_rand(n,a,c,m):
    rand_num = np.random.rand(n)
    for i in range(n):
        rand_num[i] =  lcg(a,c,m)
    return rand_num

a1  = 8
c1  = 47
m1  = 100
n = 500
seedLCG(123)
x1 = my_rand(n,a1,c1,m1)
# print(x1)
f,ax = plt.subplots(2,1,figsize=(8,12))
ax[0].plot(range(n),x1)
ax[1].hist(x1,bins=20)
```



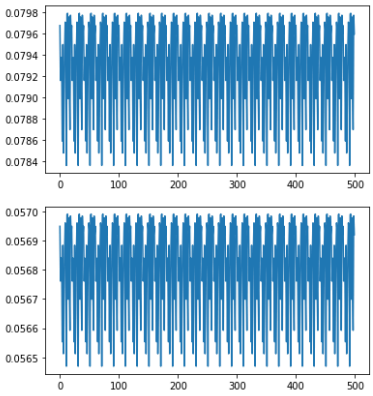


e. Buat Probability Density Function (PDF) Plot dari kedua hasil tsb

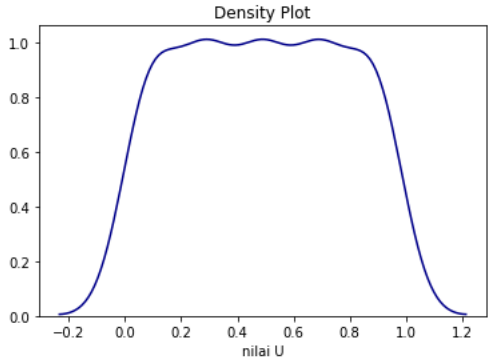
```
f,ax = plt.subplots(2,1,figsize=(6,7))
ax[0].plot(df['NO'],norm.pdf(df['nilai U']))
ax[1].plot(df2['no'], norm.pdf(df2['nilai U']))
```




```
def pdf(x, mu=0, sigma=1):
    """
    Calculates the normal distribution's probability density
    function (PDF).

    """
    term1 = 1.0 / ( sqrt(2*np.pi) * sigma )
    term2 = np.exp( -0.5 * ( (x-mu)/sigma )**2 )
    return term1 * term2

f,ax = plt.subplots(2,1,figsize=(6,7))
pdf1 = pdf(df['nilai U'], mu=0, sigma=5)
pdf2 = pdf(df2['nilai U'], mu=0, sigma=7)
ax[0].plot(range(500), pdf1)
ax[1].plot(range(500), pdf2)

plt.show()
```

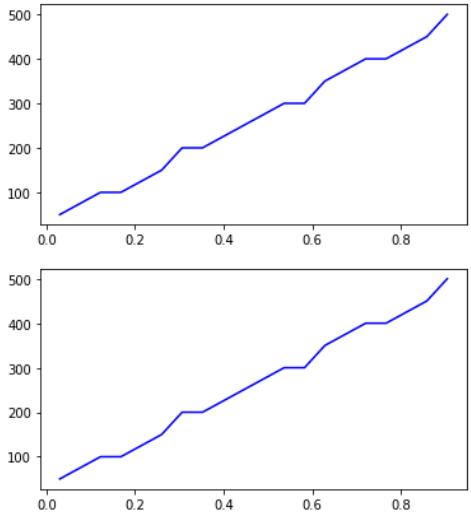


```
sns.distplot(df['nilai U'], hist=False, kde=True, color = 'darkblue')
plt.title('Density Plot')
```



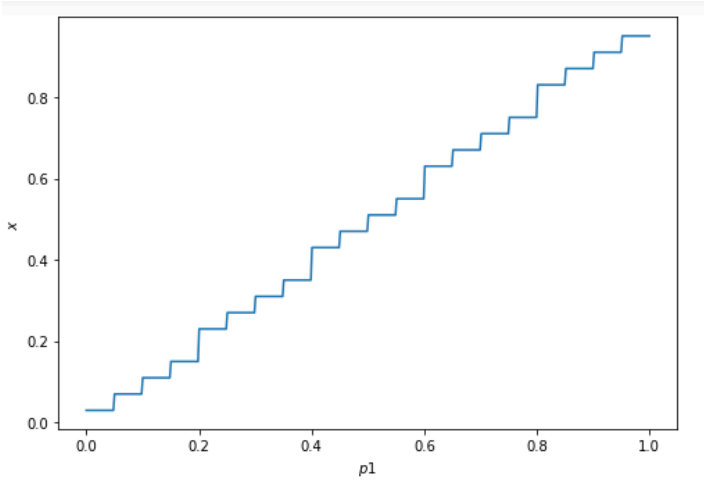
f. Buat Cumulative Density Function (CDF) Plot dari kedua hasil tsb

```
f,ax = plt.subplots(2,1,figsize=(6,7))
values, base = np.histogram(df['nilai U'], bins=20)
cumulative = np.cumsum(values)
ax[0].plot(base[:-1], cumulative, c='blue')

values2, base2 = np.histogram(df2['nilai U'], bins=20)
cumulative2 = np.cumsum(values2)
ax[1].plot(base2[:-1], cumulative2, c='blue')
```



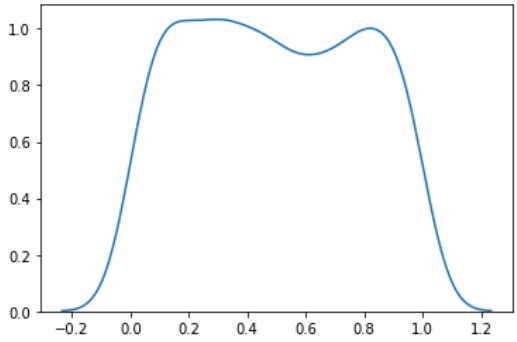

```
data1 = df["nilai U"]
data2 = df2['nilai U']


data_sorted1 = np.sort(data1)
data_sorted2 = np.sort(data2)


p1 = 1. * np.arange(len(data1)) / (len(data1) - 1)
p2 = 1. * np.arange(len(data2)) / (len(data2) - 1)
```

```
f,ax = plt.subplots(2,1,figsize=(8,12))
ax[0].plot(p1, data_sorted1)
ax[0].set_xlabel('$p1$')
ax[0].set_ylabel('$x$')

ax[1].plot(p2, data_sorted2)
ax[1].set_xlabel('$p2$')
ax[1].set_ylabel('$x$')
```

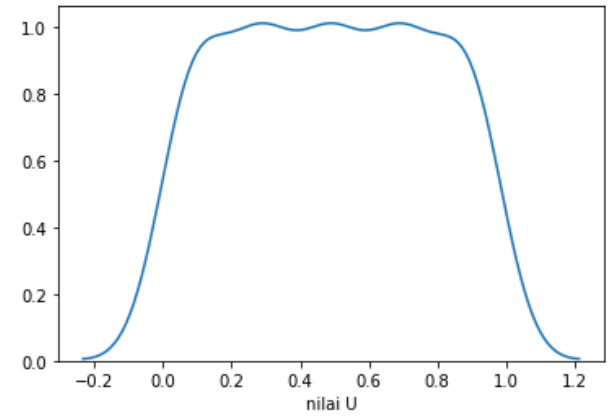


g. Lakukan analissis apakah data acak hasil excel dan python menunjukkan distribusi uniform?

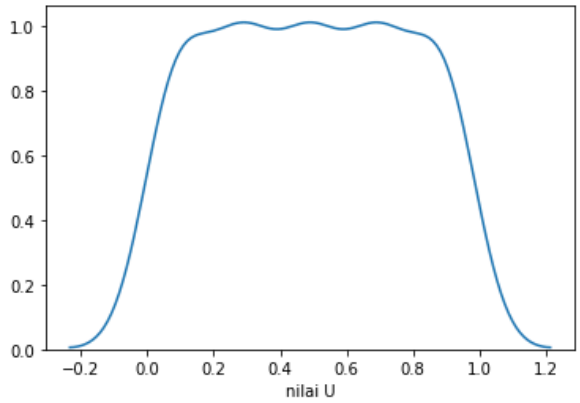
```
sns.distplot(np.random.uniform(size=1000), hist=False)
```



sns.distplot(df['nilai U'], hist=False)



sns.distplot(df2['nilai U'], hist=False)



Dari hasil data acak dari excel dan python menunjukan data tersebut merupakan distribusi uniform karena hasil peluangnya sama dan bentuk grafiknya juga sama.

2. Cari jurnal bahasa inggris tentang Random Number Generation (pilih salah satu metode misal : Psuedo random generation, linear congruent generator dan lain-lain).

    a. Tulis judul, author, resume, metode yang digunakan, dan conclusion

Judul : Inferring Sequences Produced by Pseudo-Random Number Generators

Author : JOAN BOYAR

Resume : Pseudo generator merupakan metode yang dianggap aman secara kriptograpy. Hasil dari metode pseudo ini dapat berupa angka acak dan mampu memecahkan skema enkripsi. Generator dari pseudo ini memberikan nilai sebanyak 1 bit dan dalam segi peforma sangat lambat namun secara kriptograpy sangat kuat. Metode ini melakukan setidaknya satu perkalian pada dua angka yang besar untuk setiap bitnya. Untuk menghasilkan output bit dari $log_2 n$ untuk mendapatkan hasil operasi bit ke n. Selain itu juga Pseudo nilai acak linear ini sering digunakan di dalam simulasi monte carlo dan algoritma probabilistik, Rumus yang digunakan untuk mendapatkan hasul yang lebih cepat $X_{i+1} = aX_i + b\ (mod\ m)$ namun dalam segi keamanan kurang baik (enkripsi). Namun metode infrensi ini sangat efisien untuk memprediksi urutan dari sebuah generator yang dihasilkan algoritma antara lain:

1. Dapat diasumsikan nilai dari $\emptyset_j's$ dan dihitung melalui bilang bulat tanpa mengurai modulo dari m dalam polynomial waktu dalam sebuah log m dan k
2. Tidak ada koefisien $\propto_j$ dalam modulus m
3. Membuat prediksi pada setiap elemenya secara urut dan satu persatu dari sebuah pengetahuan nilai sebelumnya
4. Memiliki kesalahan yang dibatasi polynomial dalam log m dan k dan n_0
5. Pada sebuah elemen meghasilkan sebuah prediksi dalam urutan waktu yang dibatasi oleh polynomial di log m dan k dengan demikian dapat diasumsikan bahwa setiap algoritma membuat infrensi yang salah

Metode : General Method to Linear and Quadratic Congruences

Kesimpulan : Pseudo ini sering digunakan dalam kasus enkripsi atau kriptograpy untuk sebuah keamanan sebab hasil dari pseudo ini berupa angka acak dan memiliki sebuah generator property ekstrapolasi. Selain itu metode infrensi yang efisien untuk memprediksi urutan apapun namun juga ada kekurangan terdapat sebuah metode infrensi yang tidak efisien untuk memprediksi urutan apapun atau peforma yang cukup lambat.

b. Apa yang dapat Anda simpulkan mengenai implementasi dari Random Number Generation

Random number generator ini sangat berperan penting dalam proses simulasi sebab dengan menggunakan bilang acak untuk memprediksi berbagai macam simulasi bisa juga dengan data lama bisa proses kembali menggunakan random number generator ini. Selain itu pseudo juga menggunakan bilangan acak ini untuk melakukan keamanan atau enkripsi dengan memanfaatkan generator random number ini.

3. Hasil pengamatan 100 hari terakhir diperoleh data permintaan daging per kg harian sebesar

a. Dengan informasi diatas simulasikan permintaan daging selama 15 hari dan tentukan rata-rata permintaan harian (kerjakan dengan excel)

	A	B	C	D
1	Hari	Angka Acak	Permintaan Daging Setiap harinya	
2	1	26	26	
3	2	24	24	
4	3	28	28	
5	4	24	24	
6	5	20	20	
7	6	24	24	
8	7	28	28	
9	8	26	26	
10	9	20	20	
11	10	24	24	
12	11	20	20	
13	12	28	28	
14	13	24	24	
15	14	20	20	
16	15	24	24	
17				
18	Total Permintaan Daging	360	360	
19	Rata rata Permintaan Daging	24	24	
20				
21				
22	Permintaan Daging (Kg)	Frekuensi	Probabilitas	Probabilitas Komulatif
23	20	10	10%	0
24	22	10	10%	0,1
25	24	20	20%	0,2
26	26	25	25%	0,4
27	28	30	30%	0,65
28	30	5	5%	0,95
29				

b. Cari jurnal mengenai simulasi monte carlo. Tulis judul, author, dan resume mengenai simulasi monte carlo

Judul : ANALISA SIMULASI MONTE CARLO UNTUK MEMPREDIKSI TINGKAT KEHADIRAN MAHASISWA DALAM PERKULIAHAN

Author : Harvei Desmon Hutahaean

Resume : Simulasi Monte Carlo merupakan simulasi probabilistik dimana suatu solusi. dari suatu masalah diberikan berdasarkan proses randomisasi. Proses acak ini melibatkan suatu distribusi probabilitas dari variabel data yang dikumpulkan berdasarkan data masa lalu maupun distribusi probabilitas teoritis. Simulasi bukan hanya solusi dengan menggunakan model (data atau miniatur) yang dibuat sedemikian rupa untuk menghasilkan nilai tertentu. Selain itu simulasi dapat menduga perilaku suatu sistem yang diamati dengan menggunakan data hasil pengamatan yang dilakukan dalam waktu tertentu. Smulasi juga memiliki model yang dapat di definisikan sebagai representasi dari sistem baik secara kualitatif yang mewakili suatu proses atau kejadian, dimana dapat menggambarkan secara jelas hubungan interaksi antar berbagai faktor-faktor penting yang akan diamati. Selain itu ada beberapa model dalam simulasi antara lain :

1. Model simulasi deterministik, mengasumsikan tidak ada variabilitas dalam parameter model dan, oleh karenanya, tidak melibatkan variabel random. Jika model deterministik dijalankan atas nilai masukan yang sama, maka akan selalu menghasilkan nilai yang sama. Keluaran dari sekali menjalankan model simulasi deterministik merupakan nilai nyata dari performasi model.
2. Model simulasi stokastik, berisikan satu atau beberapa variabel random untuk menjelaskan proses dalam sistem yang diamati. Keluaran dari model simulasi stokastik adalah random dan oleh karenanya hanya merupakan perkiraan dari karaterisktik sesungguhnyadari model.
3. Model simulasi kontinyu, kondisi variabel berubah secara kontinyu, sebagai contoh, aliran fluida dalam pipa atau terbangnya pesawat udara, kondisi variabel posisi dan kecepatan berubah secara kontinyu terhadap satu dengan lainnya.
4. Model simulasi diskrit, kondisi variabel berubah hanya pada beberapa titik (tertentu, yang dapat dihitung) dalam waktu. Kebanyakan dari sistem manufaktur dimodelkan sebagai simulasi kejadian dinamis, diskrit, stokastik dan menggunakan variabel random untuk memodelkan rentang kedatangan, antrian, proses, dan sebagainya.

Simulasi : Simulasi Monte Carlo didefinisikan sebagai semua teknik sampling statistik yang digunakan untuk memperkirakan solusi terhadap masalahmasalah kuantitatif (Monte Carlo Method, 2008). Dalam simulasi Monte Carlo sebuah model dibangun berdasarkan sistem yang sebenarnya. Setiap variabel dalam model tersebut memiliki nilai yang memiliki probabilitas yang berbeda, yang ditunjukkan oleh distribusi probabilitas atau biasa disebut dengan probability distribution function (pdf) dari setiap variabel. Metode Monte Carlo mengsimulasikan sistem tersebut berulangulang kali, ratusan bahkan sampai ribuan kali tergantung sistem yang ditinjau, dengan cara memilih sebuah nilai random untuk setiap variabel dari distribusi probabilitasnya. Hasil yang didapatkan dari simulasi tersebut adalah sebuah distribusi probabilitas dari nilai sebuah sistem secara keseluruhan. Simulasi Monte Carlo telah diaplikasikan pada berbagai bidang antara lain; manajemen proyek, transportasi, desain komputer, finansial, meteorologi, biologi dan biokimia.

Implementasi : Penggunana metode Monte Carlo membutuhkan sejumlah besar angka acak sehingga seiring dengan berkembangnya metode ini, berkembang pula random number generator yang ternyata lebih efektif digunakan untuk tabel angka acak yang sebelumnya sering digunakan untuk pengambilan contoh statistik. Metode ini terbagi dalam 5 tahapan :

1. Membuat distribusi kemungkinan untuk variabel penting
2. Membangun distribusi kemungkinan kumulatif untuk tiap-tiap variabel di tahap pertama
3. Menetukan interval angka random
4. Membuat simulasi dari rangkaian percobaan

Melakukan suatu prediksi untuk meramalkan suatu variable di masa mendatang dengan berdasarkan pertimbangan data pada masa lampau. Tahapan perhitungan Monte Carlo dengan cara membuat distribusi probabilitas dari variable untuk menentukan distribusi probabilitas dapat diperoleh dengan rumus

$$Probabilitas - i = \frac{Jumlah\ hadir}{Total\ hadir}$$

$$Probabilitas - i = \frac{Jumlah\ hadir}{Total\ hadir}$$

Menghitung distribusi kemungkinan kumulatif untuk variabel pada tahap pertama Konversi dari distribusi probabilitas biasa menjadi distribusi kumulatif yaitu dengan cara menjumlahkan tiap angka kemungkinan dengan jumlah sebelumnya. Dan memiliki beberapa tabel seperti tabel kehadiran, tabel probabilitas kumulatif tidak hadir, tabel kumulatif hadir dll. Dari data tabel tersebut di simulasikan dari sebuah rangkaian percobaan dengan

mengambil angka secara random dari bilangan acak untuk menentukna hasil prediksi hadir dan tidak hadir dengan menggunakan angka random.