

Pemanfaatan Teori Bilangan Dalam Penentuan Penghuni Kamar di Asrama Kidang Pananjung

Muhamad Rifki Virziadeili Harisman - 13522120¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13522120@std.stei.itb.ac.id

Abstract—Asrama Kidang Pananjung ITB adalah salah satu asrama yang disediakan ITB untuk para mahasiswanya. Pada saat proses pendaftaran, pendaftar asrama akan mendapatkan nomor kamar secara acak. Penggunaan konsep teori bilangan dapat membantu dalam melakukan pengacakan nomor kamar tersebut. Pada teori bilangan kita mengenal Aritmatika Modulo dan Pembangkit Bilangan Acak (LCG). Selain itu, *knuth shuffle* juga merupakan algoritma yang dapat menghasilkan bilangan acak dengan baik. Makalah ini akan membahas bagaimana penerapan konsep teori bilangan pada penentuan anggota tiap kamar di Asrama Kidang Pananjung ITB

Keywords—LCG, Fisher-Yates Algorithm, Asrama Kidang Pananjung ITB

I. PENDAHULUAN

Asrama Kidang Pananjung ITB adalah salah satu fasilitas tempat tinggal yang disediakan oleh ITB. Berlokasi di Jalan Cisitu Lama XI No. 12, Dago, Coblong, Kota Bandung, Jawa Barat. Asrama ini adalah asrama khusus mahasiswa putra berkapasitas 222 orang. Terdiri dari 6 gedung dengan 1 gedung sebagai ruang pengelola dan ruang diskusi. Total kamar ada sebanyak 74 kamar yang masing masing diperuntukkan untuk 3 orang setiap kamarnya.



Asrama Kidang Pananjung

Gambar 1: Asrama Kidang Pananjung ITB

(sumber: <https://asrama.itb.ac.id/asrama/4>)

Asrama ini memiliki fasilitas di setiap kamarnya berupa kasur busa dan tempat tidur, lemari pakaian, rak buku, meja dan kursi. Sedangkan fasilitas umum terdapat wifi, ruang diskusi, kamar mandi dan tempat cuci bersama, aula, mushola, perpustakaan,

lapangan olahraga dan area parkir.

Dalam penentuan penghuni tiap kamar di Asrama Kidang Pananjung ITB, diperlukan suatu cara yang dapat mengacak pendaftar secara merata. Selain melakukan pengacakan, kita juga perlu membagi-bagi pendaftar menjadi 3 orang per kamar. Penggunaan konsep teori bilangan dapat membantu dalam menentukan penghuni tiap kamar di Asrama Kidang Pananjung ITB.

Teori Bilangan merupakan cabang dari ilmu Matematika Diskrit yang mempelajari bilangan bulat dan sifatnya. Pada makalah ini akan dibahas bagaimana konsep yang ada di Teori Bilangan dapat membantu petugas Asrama Kidang Pananjung ITB dalam menentukan penghuni tiap kamar.

II. LANDASAN TEORI

A. Aritmatika Modulo

Aritmatika modulo adalah salah satu metode aritmatika untuk menyelesaikan permasalahan mengenai pada bilangan bulat. Ide awal dalam penyelesaian masalah menggunakan aritmatika modulo adalah dengan memanfaatkan informasi dari hasil bagi pembagian suatu bilangan, bukan dengan bilangan itu sendiri.

Misalkan a dan m bilangan bulat ($m > 0$). Operasi

$a \bmod m$ dibaca (“ a modulo m ”)

Notasi: $a \bmod m = r$,
sedemikian sehingga $a = mq + r$, dengan $0 \leq r < m$.

B. Relasi Rekurens

Relasi rekurens adalah bentuk persamaan yang dinyatakan dalam bentuk rekursif. Bila persamaan yang mengepresikan a_n dinyatakan secara rekursif dalam satu atau lebih *term* elemen sebelumnya, yaitu $a_0, a_1, a_2, \dots, a_{n-1}$, maka persamaan tersebut dinamakan relasi rekurens [2].

Contoh :

$$a_n = 2a_{n-1} + 1$$

$$a_n = a_{n-1} + 2a_{(n-2)}$$

$$a_n = 2a_{n-1} - a_{n-2}$$

Dalam menghitung elemen ke n dari suatu barisan rekursif, perlu ada nilai awal untuk menghitung elemen selanjutnya. Bagian ini dinamakan *basis*, yaitu nilai awal dari relasi rekurens yang berguna untuk menghentikan proses rekursi dari persamaan rekurens.

Berikut ini adalah barisan Fibonacci, barisan yang dapat dibentuk menggunakan konsep relasi rekurens.

Basis:

$$\begin{aligned} Fib_0 &= 0 \\ Fib_1 &= 1 \end{aligned}$$

Relasi rekurens:

$$Fib_n = Fib_{n-1} + Fib_{n-2}, \text{ untuk } n > 1$$

Barisan yang terbentuk:

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \dots$$

Kondisi awal secara unik menentukan elemen-elemen barisan. Kondisi awal yang berbeda akan menghasilkan elemen-elemen barisan yang berbeda pula.

C. Pembangkit Bilangan Acak Berbasis Kekongruenan Lanjar

Pembangkit bilangan acak berbasis kekongruenan lanjar biasa juga disebut LCG (*Linear Congruential Generator*). LCG merupakan sebuah algoritma yang menghasilkan urutan dari bilangan acak semu dari persamaan diskontinu lanjar piecewise. Metode ini merupakan salah satu metode tertua dan terbaik yang dapat menghasilkan bilangan acak semu.

Berikut adalah persamaan LCG:

$$X_n = (aX_{n-1} + b) \bmod m$$

Dengan,

X_n = bilangan acak ke - n dari deretnya

X_{n-1} = bilangan acak sebelumnya

a = faktor pengali, $0 < a < m$

b = increment, $0 \leq b < m$

m = modulus, $m > 0$

X_0 = seed atau nilai awal

Contoh penggunaan:

Akan dibangkitkan lima bilangan acak semu dari LCG dengan $a = 11, c = 7, m = 50, \text{ dan } x_0 = 1$.

Berikut adalah hasilnya

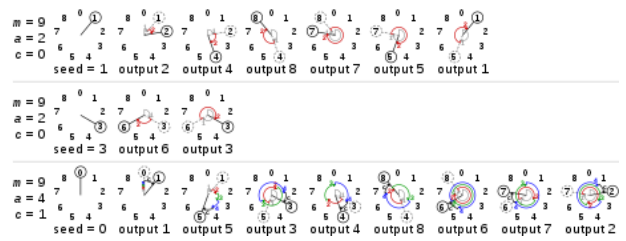
- $X_1 = (11(1) + 7) \bmod 50 = 18$
- $X_2 = (11(18) + 7) \bmod 50 = 5$
- $X_3 = (11(5) + 7) \bmod 50 = 12$
- $X_4 = (11(12) + 7) \bmod 50 = 39$
- $X_5 = (11(39) + 7) \bmod 50 = 36$

Lebih lanjut akan menghasilkan 20 bilangan acak sebagai berikut:

$$18, 5, 12, 39, 36, 3, 40, 47, 24, 21, 38, 25, 32, 9, 6, 23, 10, 17, 44, 42.$$

Pada contoh kali ini, nilai parameter yang dipilih sesuai untuk menghindari perulangan yang memunculkan nilai berulang. Pada dasarnya LCG memiliki karakteristik akan terjadi pengulangan pada periode waktu tertentu atau setelah sekian kali membangkitkan bilangan acak.

Penentuan konstanta LCG ($a, b, \text{ dan } m$) sangat menentukan baik atau tidaknya bilangan acak yang diperoleh



Berikut adalah algoritma sederhana dalam membangkitkan bilangan acak berbasis kekongruenan lanjar yang ditulis dalam notasi algoritmik.

```

procedure LCG (input a, m, c, seed : integer,
                output rand : array of integer)
{ I.S. m, c, dan seed terdefinisi
  F.S. rand berisi bilangan-bilangan bulat yang
  dapat dihasilkan oleh LCG sesuai dengan
  urutannya }

KAMUS
{ Variabel }
curr : integer
{ Fungsi Antara }
function isIn (L : array of integer,
               x : integer) → boolean
{ Mengembalikan true jika x merupakan member
  dari L dan false jika bukan merupakan
  member dari L }
procedure append (input/output L : array of
                  integer,
                  x : integer)
{ I.S. L terdefinisi dan mungkin kosong;
  x terdefinisi
  F.S. x ditambahkan sebagai elemen
  terakhir pada L }

ALGORITMA
curr ← seed
while (not isIn(rand, curr)) do
  append(rand, curr)
  curr ← (a * curr + c) mod m
  
```

Gambar 2: Algoritma LCG dalam notasi Algoritmik

(sumber:

[https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Makalah2021/Makalah-Matdis-2021%20\(116\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Makalah2021/Makalah-Matdis-2021%20(116).pdf))

D. Knuth Shuffle

Knuth Shuffle, juga dikenal sebagai algoritma Fisher-Yates, merupakan suatu algoritma pengacakan (shuffling) elemen-elemen dalam suatu himpunan. Algoritma ini pertama kali diperkenalkan oleh Ronald A. Fisher dan Frank Yates pada tahun 1938.

Knuth Shuffle merupakan versi modern dari Fisher-Yates shuffle, desain ulang dari algoritma ini diperkenalkan oleh Durstenfield pada tahun 1964 untuk dijalankan oleh komputer. Kemudian dipopulerkan oleh Donald E. Knuth di dalam bukunya "The Art of Computer Programming".

Berikut adalah algoritma dari Knuth Shuffle atau Fisher-Yates Algorithm,

```
To shuffle an array a of n elements (indices 0..n-1):
for i from n - 1 downto 1 do
    j = random integer with 0 <= j <= i
    exchange a[j] and a[i]
```

Gambar 3: Knuth Shuffle a.k.a Fisher Yates Algorithm

(Sumber: <https://www.geeksforgeeks.org/shuffle-a-given-array-using-fisher-yates-shuffle-algorithm/>)

III. ANALISIS DAN IMPLEMENTASI

A. Pemanfaatan LCG dalam Pembangkitan Bilangan Acak

Seperti yang sudah disebutkan di awal, pemilihan penghuni tiap kamar di Asrama Kidang Pananjung ITB dilakukan secara acak. Untuk memudahkan proses pengacakan tersebut, akan digunakan pembangkitan bilangan acak menggunakan LCG (*Linear Congruential Generator*).

Dalam penggunaannya pemilihan nilai dari parameter seperti a, b, m, x sangat mempengaruhi hasil angka acak yang dihasilkan. Umumnya, digunakan angka yang cukup besar dalam menggunakan algoritma ini. Pada kali ini saya akan mencoba menggunakan beberapa nilai a, b, m, x sebagai berikut,

```
import time

x = int(time.time() * (10**7))
a = 22695477
b = 1013904223
m = 2**32

def lcg(min, max):
    global x, a, b, m
    x = (a*x+b)%m
    return min+(((max-min+1)*x)//m)
```

Gambar 4: Linear Congruential Generator Algorithm

(Sumber: Dokumentasi Pribadi)

Pada algoritma LCG di atas, terdapat beberapa penyesuaian. Seperti, fungsi LCG menerima parameter berupa nilai min dan nilai max dari angka acak yang akan dihasilkan. Kemudian, hasil dari LCG menggunakan angka yang cukup besar dinormalisasi menjadi berada di rentang [min ... max].

Berikut adalah hasil uji untuk menghasilkan dua puluh angka acak pertama dalam rentang [0..20]

```
Berikut adalah 20 bilangan random pertama hasil lcg
Dengan rentang 0 sampai 20 :
3
5
4
3
6
18
20
11
9
6
14
9
17
10
1
18
10
8
7
9
PS C:\Users\Acer\Documents\Pemrogramman\Python\matdis>
```

Gambar 5: Hasil uji LCG

(Sumber: Dokumentasi Pribadi)

B. Pemanfaatan Knuth Shuffle

Setelah berhasil membangun fungsi LCG, fungsi tersebut akan digunakan untuk melakukan pencarian angka acak pada algoritma *knuth_shuffle*.

Berikut adalah implementasi dari algoritma *knuth_shuffle*,

```
from lcg import *

def knuth_shuffle(items):
    for i in range(len(items) - 1, 0, -1):
        pick = lcg(0,i)
        items[pick], items[i] = items[i], items[pick]
    return items
```

Gambar 6: Knuth Shuffle a.k.a Fisher-Yates Algorithm

(Sumber: Dokumentasi Pribadi)

Untuk menguji apakah program berjalan dengan baik, maka dilakukan uji test menggunakan list integer sebagai berikut,

```
PS C:\Users\Acer\Documents\Pemrogramman\Python\matdis> & C:/Python312/
Berikut adalah contoh hasil pengacakan dengan algoritma knuth shuffle
List awal: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
List Setelah: [8, 12, 5, 9, 13, 0, 6, 4, 7, 1, 11, 10, 3, 2]
PS C:\Users\Acer\Documents\Pemrogramman\Python\matdis>
```

Gambar 7: Hasil Uji knuth_shuffle

(Sumber: Dokumentasi Pribadi)

C. Membaca data

Dalam proses pendaftaran penghuni asrama, biasanya data akan disimpan dalam ekstensi *.xlsx*, *.xls* ataupun *.csv* (File excel atau spreadsheet). Data tersebut akan berisi informasi-informasi tentang pendaftar penghuni asrama.

Dalam penentuan penghuni tiap kamar asrama, data tersebut dapat digunakan sebagai bahan acuan. Pada makalah ini, penulis membuat model dari data pendaftaran

tersebut. Berikut adalah pemodelan data pendaftar asrama,

	A	B	C	D
1	No	Nama	Asal	No. Telp
2	1	rasya	Bandung	0821389293
3	2	lukman	Lampung	0821389294
4	3	yusuf	Bandung	0821389295
5	4	mahardika	Karawang	0821389296
6	5	bahtiar	Sorong	0821389297
7	6	sumanto	Makasar	0821389298
8	7	bejo	Medan	0821389299
9	8	bokir	Semarang	0821389300
10	9	danto	Bali	0821389301
11	10	wagyu	Bandung	0821389302
12	11	sudarto	Bekasi	0821389303
13	12	eka	Bandung	0821389304
14	13	munhas	Bogor	0821389305
15	14	sulaiman	Jakarta	0821389306

Gambar 8: Model data pendaftar Asrama Kidang Pananjung ITB

(Sumber: Dokumentasi Pribadi)

Model data yang telah dibuat dalam file excel tersebut akan diakses dalam program python. Pada makalah ini, proses membaca file excel dilakukan dengan bantuan modul openpyxl. Berikut implementasi programnya,

```

from openpyxl import load_workbook

wb = load_workbook("Data_Penghuni_Asrama.xlsx")
sheet = wb.active

data_list = []

for row in sheet.iter_rows(min_row=2, max_col=3, values_only=True):
    tmp = []
    tmp.append(row[0])
    tmp.append(row[1])
    data_list.append(tmp)

for data in data_list:
    print(data)

```

Gambar 9: Implementasi baca file excel menggunakan openpyxl

(Sumber: Dokumentasi Pribadi)

Untuk pemodelan sederhana, penulis hanya mengambil data nomor pendaftaran dan nama dari model data pendaftar penghuni asrama.

Setelah dilakukan testing terhadap program, ternyata program berhasil menyimpan data nomor pendaftaran dan nama dari tiap-tiap pendaftar penghuni asrama. Berikut adalah hasil bacaan dari program di atas,

```

PS C:\Users\Acer\Documents
[1, 'rasya']
[2, 'lukman']
[3, 'yusuf']
[4, 'mahardika']
[5, 'bahtiar']
[6, 'sumanto']
[7, 'bejo']
[8, 'bokir']
[9, 'danto']
[10, 'wagyu']
[11, 'sudarto']
[12, 'eka']
[13, 'munhas']
[14, 'sulaiman']
PS C:\Users\Acer\Documents

```

Gambar 10: Hasil bacaan model data pada program python

(Sumber: Dokumentasi Pribadi)

D. Implementasi Program

Setelah semua kebutuhan program telah terpenuhi dan dapat berjalan dengan baik. Sesi berikutnya adalah membuat program menjadi satu program utuh yang dapat membaca data model dari file excel dan melakukan proses pengacakan untuk penghuni tiap kamar asrama.

Program diawali dengan membaca file "Data_Penghuni_Asrama.xlsx", file ini berisi model data yang telah dibuat sebelumnya. Kemudian dengan bantuan modul openpyxl, kita akan mengakses *sheet* dari file excel tersebut.

Setelah *sheet* dapat diakses, kita dapat membuat list kosong yang nantinya akan diisi oleh data hasil bacaan dari file model data.

Proses pengisian list dilakukan dengan mengambil kolom ke-0 dan ke-1 dari excel. Data tersebut kemudian dimasukkan ke dalam list kosong yang telah dibuat.

Setelah seluruh data telah dibaca, saat ini list telah berisi kumpulan nama dan nomor pendaftaran penghuni asrama yang telah dibaca dari file model data

Kita akan memanggil fungsi `input_to_room` yang bertujuan mengacak susunan dari data pendaftar asrama, kemudian membagi setiap tiga anggota penghuni kamar. Mengingat kapasitas penghuni tiap kamar di Asrama Kidang Pananjung ITB adalah tiga orang per kamar.

Dalam prosesnya, setelah setiap data diacak. Program akan melakukan iterasi pada isi list. Jika isi list memiliki index yang habis dibagi tiga, maka akan menampilkan di kamar mana penghuni berikutnya akan tinggal.

IV. KESIMPULAN

Matematika diskrit adalah cabang ilmu matematika yang sangat berguna dalam kehidupan sehari-hari. Dalam keinformatikaan matematika diskrit menjadi ilmu dasar yang perlu dikuasai dengan baik. Banyak sekali pengaplikasian ilmu ini.

Makalah ini sebagai bentuk nyata, ada beberapa ilmu dari matematika diskrit yang dipakai dalam menyelesaikan permasalahan di kehidupan nyata. Dalam penentuan penghuni kamar Asrama Kidang Pananjung ITB, kita dapat mengimplementasikan konsep teori bilangan.

Konsep teori bilangan yang dipakai pada makalah ini ada, Aritmaika modular, Pembangkit bilangan acak berbasis kekongruenan lanjar, serta Algoritma Fisher-Yates a.k.a Knuth Shuffle yang didapat dari bahan bacaan lain.

Dengan menggunakan konsep matematika diskrit tersebut, dapat dibuat sistem yang dapat diaplikasikan secara langsung untuk penentuan penghuni kamar Asrama Kidang Pananjung ITB.

VI. SARAN

Dalam pemodelan sederhana kali ini masih ada beberapa hal yang dirasa masih bisa dimaksimalkan agar program dapat menjadi lebih bermanfaat. Salah satunya adalah memaksimalkan proses baca tulis dari program ke dalam suatu file yang dapat dibaca oleh orang awam juga.

Jumlah data pada kondisi nyata pasti akan jauh lebih banyak dan lebih variatif. Keterbacaan data hasil komputasi seharusnya dapat dibaca juga oleh orang awam yang tidak mengerti tentang pemrograman. Oleh karena itu perlu adanya peningkatan dari proses pemodelan sederhana ini agar dapat dipakai secara nyata.

VII. UCAPAN TERMA KASIH

Di akhir penghujung makalah ini, penulis ingin mengucapkan terima kasih sebanyak-banyaknya kepada beberapa pihak yang telah membantu penulis dalam menyelesaikan makalah ini :

- 1). Tuhan yang Maha Esa atas anugerah dan rahmat-Nya, penulis dapat menyelesaikan makalah ini dengan baik.
- 2). Dr. Fariska Zakhralativa Ruskanda, S.T., M.T. selaku dosen kelas K2 karena sudah membimbing penulis dengan sangat baik selama perkuliahan Matematika Diskrit.
- 3). Seluruh tim dosen mata kuliah Matematika Diskrit Sem. I Tahun 2023/2024 telah berperan dalam memberi wadah pembelajaran kepada penulis dan mahasiswa lainnya..

```
from lcg import *
from knuth_shuffle import *
from openpyxl import load_workbook

wb = load_workbook("Data_Penghuni_Asrama.xlsx")
sheet = wb.active

data_list = []

for row in sheet.iter_rows(min_row=2, max_col=3, values_only=True):
    tmp = []
    tmp.append(row[0])
    tmp.append(row[1])
    data_list.append(tmp)

print("-----Data penghuni asrama-----")
def input_to_room(list, room_capacity):
    room = 1
    for i in range(0, len(list)):
        if(i%room_capacity==0):
            print(f"Kamar {room}: ")
            room+=1
            print("-|",list[i])

input_to_room(knuth_shuffle(data_list), 3)
```

Gambar 11: Program utama

(Sumber: Dokumentasi Pribadi)

Berikut adalah tampilan hasil dari program utama. Terlihat bahwa data penghuni asrama telah teracak dan menempati kamar dengan masing masing kamar berjumlah maksimal tiga penghuni

```
PS C:\Users\Acer\Documents\Pemrograman>
-----Data penghuni asrama-----
Kamar 1:
-| [11, 'sudarto']
-| [3, 'yusuf']
-| [8, 'bokir']
Kamar 2:
-| [5, 'bahtiar']
-| [13, 'munhas']
-| [14, 'sulaiman']
Kamar 3:
-| [7, 'bejo']
-| [1, 'rasya']
-| [12, 'eka']
Kamar 4:
-| [10, 'wagyu']
-| [4, 'mahardika']
-| [6, 'sumanto']
Kamar 5:
-| [9, 'danto']
-| [2, 'lukman']
PS C:\Users\Acer\Documents\Pemrograman>
```

Gambar 12: Hasil dari Program Utama

(Sumber: Dokumentasi Pribadi)

REFERENCES

- [1] Direktorat Sarana dan Prasarana Institut Teknologi Bandung, Data Asrama ITB, “Asrama Kidang Pananjung”, <https://asrama.itb.ac.id/asrama/4> (Diakses pada 10 Desember 2023)
- [2] Munir, Rinaldi. 2023. “Rekursi dan relasi rekurens (Bagian 2)”. [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/10-Rekursi-dan-relasi-rekurens-\(Bagian2\)-2023.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/10-Rekursi-dan-relasi-rekurens-(Bagian2)-2023.pdf) (Diakses pada 10 Desember 2023).
- [3] Munir, Rinaldi. 2023. “Teori Bilangan (Bagian 3)”. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/16-Teori-Bilangan-Bagian3-2023.pdf> (Diakses pada 10 Desember 2023)
- [4] Brianaldo. Phandiarta. 2021. “Aplikasi Pembangkit Bilangan Acak Berbasis Kekongruenan Lanjar pada Mesin Slot”, [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Makalah2021/Makalah-Matdis-2021%20\(116\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2021-2022/Makalah2021/Makalah-Matdis-2021%20(116).pdf) (Diakses pada 10 Desember 2023)
- [5] Mesran. 2013. “Metode LCM (Linear Congruent Method)”. <https://mesran.blogspot.com/2013/06/metode-lcm-linear-congruent-method.html> (Diakses pada 10 Desember 2023)
- [6] Fangasorla. “LCG (linear congruential generator): a simple random number generation algorithm” <https://www.cnblogs.com/vancasola/p/9942583.html> (Diakses pada 10 Desember 2023)
- [7] Openpyxl, Docs, “openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files”, <https://openpyxl.readthedocs.io/en/stable/> (diakses pada 11 Desember 2023)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Desember 2023



Muhamad Rifki Virziadeili Harisman
(13522120)