

# Simulasi Penyingkat URL Menggunakan Linear Congruential Generator (LCG)

Ahmad Naufal Ramadan - 13522005<sup>1</sup>

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

<sup>1</sup>13522005@itb.ac.id

**Abstract**— Pembangkit Bilangan Acak Semu (PBAS) memiliki peranan sentral dalam menghasilkan deret angka yang tampak acak. Beberapa aplikasi dari PBAS melibatkan berbagai bidang seperti video game, simulasi, dan analisis numerik. Salah satu metode PBAS yang umum digunakan adalah *Linear Congruential Generator (LCG)*, sebuah algoritma sederhana yang menggunakan persamaan rekursif linier dan operasi modulo untuk menciptakan deret angka acak semu. Dalam konteks ini, makalah ini bertujuan untuk menyelidiki penerapan LCG dalam simulasi penyingkatan URL. LCG dapat memberikan unsur keacakan yang diperlukan dalam proses penyingkatan URL.

**Keywords**— LCG, PBAS, penyingkat URL, simulasi

## I. PENDAHULUAN

Dalam era teknologi saat ini, koneksi internet telah menjadi kebutuhan utama dalam segala aspek kehidupan. Salah satu penggunaan internet yang paling umum adalah untuk mengakses suatu situs. Sering kali kita bertukar informasi dengan mengirimkan tautan situs ke suatu aplikasi komunikasi. Tautan dari suatu situs kadang terlalu panjang. Hal ini mengakibatkan tautan situs sulit untuk diingat sehingga cara untuk mengaksesnya kembali adalah dengan membuka aplikasi komunikasi tersebut kembali atau mencari dalam *search history* dari *browser* yang digunakan. Salah satu cara yang dapat digunakan untuk memudahkan hal tersebut adalah dengan menggunakan penyingkat URL.

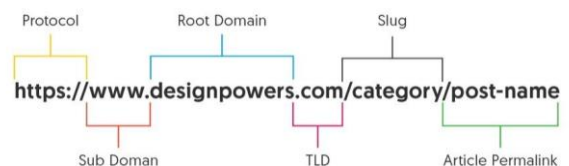
Konsep penyingkat URL berfokus pada pembuatan tautan baru yang lebih sederhana namun tetap mengarahkan pengguna ke halaman yang sama. Tautan yang disingkat juga mengganti domain dari tautan yang disingkat. Aplikasi populer seperti bit.ly bahkan memberikan opsi kepada pengguna untuk memilih apakah mereka ingin menentukan nama untuk tautan yang disingkat atau menggunakan nama acak. Pembangkit nama acak ini dapat diimplementasikan menggunakan *Linear Congruential Generator (LCG)*. Berikut akan diulas simulasi penyingkat URL dengan mengaplikasikan LCG untuk menghasilkan nama acak.

## II. DASAR TEORI

### 2.1 URL

URL (*Uniform Resource Locator*), atau alamat web, merupakan konsep dalam navigasi internet yang menyediakan cara untuk mengakses berbagai halaman web di seluruh dunia.

Sebuah URL terdiri dari enam komponen utama yang membentuk strukturnya, dimulai dengan protocol, yang menentukan metode komunikasi yang akan digunakan, seperti HTTP atau HTTPS. Selanjutnya, sub domain merupakan bagian yang menyediakan informasi tambahan sebelum nama domain utama. Root domain mencakup nama inti dari suatu website. Top level domain (TLD) menentukan jenis atau kategori dari suatu website, seperti .com, .org, atau .net. Kemudian, slug memberikan informasi tambahan tentang halaman atau konten spesifik yang ingin diakses, dan article permalink menunjukkan

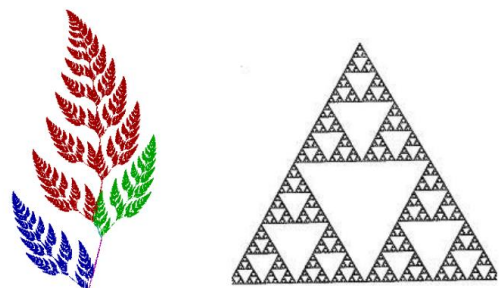


**Gambar 2.2.1** Bagian-bagian dari URL

Sumber: <https://designpowers.com/blog/url-best-practices>

### 2.2 Rekursi dan Relasi Rekurens

Rekursi adalah suatu proses yang mendefinisikan suatu objek dengan dirinya sendiri. Contoh dari rekursi dapat ditunjukkan dari suatu objek fraktal sebagai berikut:



**Gambar 2.2.2** Rekursi dalam bentuk fraktal

Sumber:

[https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/09-Rekursi-dan-relasi-rekurens-\(Bagian1\)-2023.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/09-Rekursi-dan-relasi-rekurens-(Bagian1)-2023.pdf)

Fungsi rekursif adalah fungsi yang didefinisikan dari dua bagian, yaitu:

1. Basis, yaitu bagian yang berisi nilai fungsi yang terdefinisi secara eksplisit sekaligus menghentikan rekursif.
2. Rekurens, yaitu bagian yang mendefinisikan fungsi dalam terminologinya sendiri.

Relasi rekurens adalah suatu barisan yang dapat dijelaskan secara rekursif melalui satu atau lebih elemen sebelumnya. Sebuah relasi rekurens membutuhkan kondisi awal sebagai basis dari rekursi. Kondisi awal ini menjadi kunci dalam menentukan elemen-elemen berikutnya sesuai dengan aturan rekursif yang telah ditetapkan. Kondisi awal yang berbeda dapat menentukan karakteristik dari barisan yang dibentuk. Perbedaan pada kondisi awal dapat menghasilkan barisan yang berbeda sehingga membentuk keragaman dalam sifat dan pola yang muncul. Relasi rekurens memungkinkan pemodelan dan analisis yang mendalam terhadap struktur barisan, menyediakan dasar untuk pemahaman yang lebih baik terhadap pola-pola matematika yang muncul dalam berbagai konteks.

### 2.3 Teori Bilangan

Teori bilangan adalah cabang matematika murni yang berfokus pada bilangan bulat dan fungsi pada bilangan bulat. Beberapa hal yang dipelajari diantaranya adalah:

1. Sifat pembagian pada bilangan bulat
2. Teorema dan algoritma euclidian
3. Pembagi bersama terbesar
4. Bilangan prima
5. Aritmetika modulo
6. Sistem kekongruenan linier

Teori bilangan memiliki banyak penerapan dalam kehidupan sehari-hari. Salah satu contoh penerapannya dapat ditemukan dalam sistem ISBN (*International Standard Book Number*). Konsep teori bilangan digunakan untuk merancang nomor identifikasi unik bagi buku-buku di seluruh dunia sehingga mempermudah klasifikasi dan manajemen inventaris. Selain itu, teori bilangan juga berkontribusi pada fungsi hash untuk keperluan manajemen basis data, penyimpanan password terenkripsi, dan keamanan sistem informasi. Dalam bidang kriptografi, teori bilangan memainkan peran penting dalam merancang algoritma enkripsi seperti protokol kunci publik RSA. Selain itu, teori bilangan memberikan landasan untuk pembangkit bilangan acak semu, seperti *Linear Congruential Generator* (LCG), yang memanfaatkan konsep aritmetika modulo dan relasi rekurens untuk menghasilkan deret bilangan acak semu.

ISBN 817525766-0



**Gambar 2.2.3** ISBN sebagai aplikasi dari teori bilangan  
Sumber:

<https://indonesia.go.id/layanan/kependudukan/sosial/cara-registrasi-isbn-untuk-penerbit-baru>

### 2.4 Linear Congruential Generator (LCG)

*Linear Congruential Generator* (LCG) adalah suatu fungsi rekursif yang dapat menghasilkan bilangan acak semu. Fungsi ini memanfaatkan sistem kekongruenan linier untuk mendapatkan suatu bilangan yang terlihat acak. Persamaan LCG paling sederhana adalah sebagai berikut:

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

$X_n$  = bilangan acak ke-n dari deretnya

$X_{n-1}$  = bilangan acak sebelumnya

a = faktor pengali

b = increment

m = modulus

Karena persamaan LCG merupakan relasi rekurens, maka harus ada kondisi awal untuk menciptakan deret bilangan acak semu. Kondisi awal ini berupa kunci pembangkit  $X_0$  yang disebut umpan (seed).

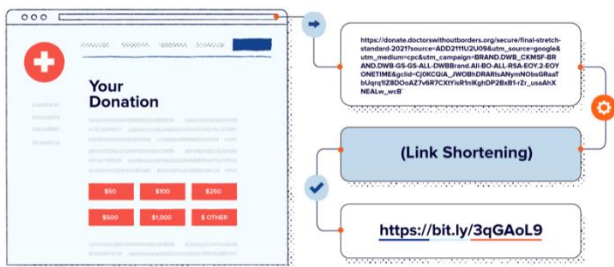
LCG menawarkan kelebihan dalam implementasi yang sederhana dan kemampuan untuk menghasilkan deret bilangan acak dengan cepat. Kebutuhan akan kecepatan tersebut membuat LCG menjadi pilihan utama dalam berbagai aplikasi seperti simulasi, optimasi, dan pemodelan. Selain itu, LCG memiliki keunggulan dalam kemampuannya menghasilkan bilangan acak dengan periode yang besar dan diketahui, menjadikannya cocok untuk penggunaan dalam simulasi jangka panjang atau pengujian algoritma. Selain itu, kelebihan LCG juga terletak pada penggunaannya pada sistem terbenam. Algoritma LCG memerlukan sedikit memori untuk beroperasi sehingga cocok digunakan pada aplikasi terbenam.

Walaupun LCG memiliki kelebihan, terdapat kekurangan yang perlu diperhatikan. LCG tidak cocok untuk kebutuhan kriptografi karena keamanannya dapat dengan mudah dipecahkan jika parameter yang digunakan telah diketahui. Dalam dunia kriptografi, keamanan menjadi prioritas utama, sehingga algoritma PBAS lain yang didesain khusus untuk keamanan kriptografis lebih disarankan. Oleh karena itu, pemilihan algoritma haruslah sesuai dengan kebutuhan spesifik aplikasi, dan LCG sebaiknya digunakan dengan bijak, terutama di lingkungan yang membutuhkan tingkat keamanan yang tinggi.

### 2.5 Penyingkat URL

Penyingkat URL adalah suatu aplikasi website yang dapat membuat suatu URL dari website menjadi lebih pendek sehingga lebih mudah diingat dan dilacak. Beberapa platform terkenal seperti bit.ly, tinyurl.com, dan s.id telah berhasil mempopulerkan layanan ini. Beberapa keuntungan dari penyingkat URL adalah sebagai berikut:

1. Mempermudah untuk berbagi konten
2. Membuat URL yang dibagikan menjadi lebih menarik
3. Memudahkan monitoring traffic



**Gambar 2.2.4** Contoh penggunaan bit.ly

Sumber: <https://bitly.com/blog/url-shortening-101-how-it-works-use-cases-and-examples/>

Pengguna memiliki pilihan untuk memasukkan URL secara manual atau membiarkan sistem menentukan URL secara acak. Apabila tidak memilih untuk memasukkan URL secara manual, *Linear Congruential Generator* (LCG) dapat berperan penting dalam menciptakan deret angka acak semu yang mewakili karakter-karakter yang membentuk URL singkat. Dengan mengandalkan LCG, proses ini tidak hanya menjadi efisien tetapi juga dapat memberikan unsur keacakan yang diperlukan untuk menciptakan URL singkat yang unik dan sulit diprediksi.

### III. IMPLEMENTASI

Program dibuat menggunakan bahasa python. Program dibagi menjadi dua file, yaitu file LCG.py dan main.py File LCG.py memiliki peran utama sebagai tempat inisialisasi dari class LCG yang telah dibuat. Class ini dirancang berdasarkan standar Linear Congruential Generator (LCG) pada bahasa Java. Seed yang digunakan dalam program ini diambil dari waktu saat program dijalankan dalam nano sekon.

```

1  import time
2
3
4  class LCG:
5      def __init__(self) -> None:
6          self.m = 281474976710656
7          self.a = 25214983917
8          self.c = 11
9          self.seed = round(time.time_ns())
10
11     def next(self) -> int:
12         self.seed = (self.a * self.seed + self.c) % self.m
13         return (self.seed - (self.seed & 32767)) >> 16
14

```

**Gambar 3.1** Class LCG dengan seed time dalam nano sekon  
Sumber: Dokumen penulis

File main.py menjadi inti dari simulasi peningkat URL berbasis *Command Line Interface* (CLI) dengan menggunakan file data.json sebagai pengganti database. Program ini mensimulasikan cara kerja peningkat URL dengan memproduksi URL acak dalam format karakter alphanumeric Base62. Data yang diambil dari file data.json diubah menjadi

struktur dictionary. Proses ini terus berlanjut hingga pengguna memutuskan untuk menghentikan penggunaan program, memberikan pengguna kontrol penuh atas durasi dan lingkup penggunaan.

```

5  def main() -> None:
6      # Base 62 characters
7      alphanumeric = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
8      # Initialize lcg
9      lcg = LCG()
10
11     # Open data.json to dictionary
12     with open("data.json", "r") as f:
13         data = json.load(f)
14
15     print(
16         """
17     0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
18     0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
19     0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
20     0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
21     0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
22     0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
23     Selamat datang di shortener.com, tempat untuk memendekkan link agar terlihat lebih rapi dan sederhana.
24     """
25     )
26

```

**Gambar 3.2** Inisialisasi program utama  
Sumber: Dokumen penulis

```

27     while True:
28         link_input = input("Masukkan link yang ingin dipendekkan: ")
29         print()
30
31         done = False

```

**Gambar 3.3** Menerima masukan URL yang ingin dipendekkan dari pengguna  
Sumber: Dokumen penulis

Proses pembuatan URL acak dalam implementasi ini melibatkan langkah-langkah yang terstruktur. Pertama, bilangan acak dihasilkan dengan menggunakan LCG. Kemudian, bilangan acak tersebut dimodulo dengan 62, yang sesuai dengan jumlah karakter pada Base62. Hasil dari operasi modulo ini menghasilkan nilai yang menjadi indeks pada string alphanumeric yang telah diinisialisasi sebelumnya. String ini berfungsi sebagai "pool" karakter yang dapat dipilih untuk membentuk URL acak. Proses ini diulang sebanyak 8 kali, menghasilkan deretan 8 karakter acak yang akan menjadi bagian slug dari URL acak yang dibentuk. Dengan pendekatan ini, program tidak hanya menghasilkan URL acak yang unik secara efektif, tetapi juga memastikan keberagaman dan keamanan dalam karakter yang dihasilkan, menciptakan tautan yang bervariasi dan acak.

```

main.py
32 while not done:
33     print(
34         "Masukkan nama link (jika kosong akan diisi dengan alphanumeric random)"
35     )
36     output = input("shorten.er/")
37
38     if output == "":
39         # Create random sequence of 8 characters using lcg
40         # Chances of the same sequence are 1/62^8
41         rand_val = ""
42         for i in range(8):
43             rand_val += alphanumeric[lcg.next() % 62]
44         link_output = "shorten.er/" + rand_val
45
46     # Check for same sequence just in case if there is one
47     while link_output in data.keys():
48         rand_val = ""
49         for i in range(8):
50             rand_val += alphanumeric[lcg.next() % 62]
51         link_output = "shorten.er/" + rand_val
52
53     print(f"Link berhasil disingkat: {link_output}")
54
55     # Insert key value pair data to dictionary
56     data[link_output] = link_input
57     done = True

```

Snipped

**Gambar 3.4** Pemanfaatan LCG dalam penyingkat URL  
 Sumber: Dokumen penulis

```

main.py
70     print("Terima kasih sudah menggunakan program kami")
71
72     # Overwrite data.json with new data
73     with open("data.json", "w") as f:
74         json.dump(data, f)

```

Snipped

**Gambar 3.7** Menyimpan data dalam data.json  
 Sumber: Dokumen penulis

```

main.py
55     else:
56         link_output = "shorten.er/" + output
57         if link_output in data.keys():
58             print("Nama link sudah ada di database")
59         else:
60             data[link_output] = link_input
61             done = True

```

Snipped

**Gambar 3.5** Fitur mengubah nama URL sesuai masukan pengguna  
 Sumber: Dokumen penulis

```

datajson
1  {
2  "shorten.er/72m4lg1D": "youtube.com",
3  "shorten.er/halo": "instagram.com",
4  "shorten.er/uEa77UjH": "youtube.com"
5  }
6

```

Snipped

**Gambar 3.7** Contoh isi data.json  
 Sumber: Dokumen penulis

```

main.py
65     lanjut = input("Masih ingin lanjut? (Y/n) ")
66     print()
67     if lanjut.lower() == "n":
68         break

```

Snipped

**Gambar 3.6** Menerima masukan dari pengguna jika ingin melanjutkan program atau tidak  
 Sumber: Dokumen penulis

#### IV. HASIL DAN PEMBAHASAN

Hasil pengujian diperoleh dari serangkaian pengujian kasus yang mencakup berbagai skenario yang mungkin terjadi dalam sebuah penyingkat URL. Proses pengujian ini dilakukan untuk menguji fungsionalitas dan fleksibilitas penyingkat URL dalam memenuhi kebutuhan pengguna. Pada penyingkat URL seperti bit.ly, pengguna memiliki kemampuan untuk mengonfigurasi URL sesuai dengan preferensi mereka, memungkinkan personalisasi tautan agar sesuai dengan isi kontennya. Pengguna dapat memasukkan URL yang mudah diingat dan berkaitan dengan isi konten dari halaman website yang dituju oleh URL yang ingin disingkat. Untuk kebutuhan pengguna yang hanya ingin menyederhanakan URL tanpa memikirkan personalisasi, penyingkat URL menyediakan fitur untuk menghasilkan URL singkat secara acak dan pasti unik. Fitur ini berguna untuk menghindari konflik dengan URL yang sudah ada sebelumnya dan memastikan keamanan informasi yang terkandung dalam URL.

```
Masukkan link yang ingin dipendekkan: http://sister21.tech/
Masukkan nama link (jika kosong akan diisi dengan alphanumeric random)
shorten.er/PraktikumOrkom
Masih ingin lanjut? (Y/n) Y
```

**Gambar 4.1** Pengujian penyingkat URL dengan masukan dari pengguna  
Sumber: Dokumen penulis

Program ini memulai dengan meminta pengguna untuk memasukkan URL yang ingin disingkat. Pengujian pertama dilakukan pada URL yang menuju ke situs web praktikum Organisasi dan Arsitektur Komputer. Tujuan utama dari pengujian ini adalah untuk melihat sejauh mana penyingkat URL mampu memberikan hasil yang sesuai dengan keinginan pengguna, yaitu membuat URL lebih mudah diingat. Sebagai contoh, pengguna dapat mengubah URL menjadi shorten.er/PraktikumOrkom untuk menyederhanakan dan mempersonalisasi tautan. Setelah proses singkat dilakukan, URL baru ini akan ditambahkan ke dalam struktur data dictionary yang berisi pasangan antara URL yang disingkat dan URL yang sebenarnya. Langkah ini membantu menciptakan catatan yang rapi dan terstruktur dari URL-URL yang telah disingkat, memudahkan pengelolaan dan pemantauan penggunaan.

```
Masukkan link yang ingin dipendekkan: https://informatika.stei.itb.ac.id/~rinaldi.munir/
Masukkan nama link (jika kosong akan diisi dengan alphanumeric random)
shorten.er/
Link berhasil disingkat: shorten.er/7evV90G2
Masih ingin lanjut? (Y/n) |
```

**Gambar 4.2** Pengujian penyingkat URL dengan pembangkit bilangan acak semu LCG  
Sumber: Dokumen penulis

Selanjutnya akan dilakukan uji coba pada fitur penyingkat URL dengan pembangkit bilangan acak semu LCG. URL yang ingin disingkat adalah URL yang menuju ke situs web Pak Rinaldi Munir. Dapat dilihat URL menuju ke situs web Pak Rinaldi cukup panjang. Hal ini menjadi hal yang sangat tepat untuk menggunakan penyingkat URL agar dapat memudahkan pengguna untuk kembali ke halaman web tersebut. Hasil dari URL yang telah disingkat akan ditampilkan di terminal. Bagian slug dari URL merupakan 8 karakter acak yang merupakan bagian dari karakter Base62 sehingga banyaknya kemungkinan URL acak yang dibuat adalah  $62^8$  atau sebanyak 218.340.105.584.896 kemungkinan. Total populasi manusia di Bumi saat ini adalah sekitar 8.077.653.796. Perbandingan banyaknya kemungkinan dan jumlah populasi manusia di Bumi adalah sekitar 27.030, yang artinya untuk setiap orang dapat membuat sebanyak 27.030 URL acak yang unik. Hal ini sudah cukup aman untuk simulasi yang dibuat mengingat total pengguna pasti akan jauh di bawah total populasi manusia di Bumi, yang artinya setiap orang dapat membuat URL acak lebih banyak.

```
Masukkan link yang ingin dipendekkan: https://informatika.stei.itb.ac.id/~rinaldi.munir/
Masukkan nama link (jika kosong akan diisi dengan alphanumeric random)
shorten.er/KitabDingDong
Masih ingin lanjut? (Y/n)
```

**Gambar 4.3** Pengujian penyingkat URL dengan masukan dari pengguna pada URL yang telah disingkat sebelumnya  
Sumber: Dokumen penulis

Pada tahap uji coba selanjutnya, fokus akan diarahkan pada penyingkatan URL yang telah mengalami proses penyingkatan sebelumnya. Proses penyingkatan URL tetap berlaku karena setiap URL singkat yang dimasukkan unik. Kondisi ini menjadi alasan penggunaan tipe data dictionary dalam menyimpan pasangan URL yang disingkat dan URL sebenarnya. Keunggulan tipe data dictionary terletak pada keunikan key-nya, sehingga cocok sebagai wadah untuk menyimpan URL yang telah disingkat. Sifat unik ini memberikan jaminan bahwa tidak akan ada konflik atau duplikasi dalam penyimpanan. Selain itu, value dari dictionary memiliki fleksibilitas yang tinggi, memungkinkan penyimpanan data duplikat. Hal ini menjadikan value sebagai tempat yang ideal untuk menyimpan URL sebenarnya. Dengan merujuk pada URL yang telah disingkat, pengaksesan langsung terhadap URL asli dapat dilakukan dengan mencari value dari key yang sesuai, yaitu URL singkat tersebut.

```
Masukkan link yang ingin dipendekkan: https://akademik.itb.ac.id/home
Masukkan nama link (jika kosong akan diisi dengan alphanumeric random)
shorten.er/KitabDingDong
Nama link sudah ada di database
Masukkan nama link (jika kosong akan diisi dengan alphanumeric random)
shorten.er/
Link berhasil disingkat: shorten.er/m1NnCqTA
Masih ingin lanjut? (Y/n) n
```

**Gambar 4.3** Pengujian penyingkat URL dengan masukan dari pengguna yang telah ada di database  
Sumber: Dokumen penulis

Pada pengujian terakhir, fokus ditujukan pada situasi di mana pengguna mencoba memasukkan URL singkat yang telah ada dalam database. Dalam konteks ini, program didesain untuk menolak masukan pengguna, mengingat hal ini dapat menciptakan ambiguitas terkait dengan URL singkat yang akan merujuk pada dua alamat website yang berbeda. Keberlanjutan program mengharuskan pengguna untuk memasukkan URL singkat lain hingga ditemukan URL yang benar-benar unik dan belum ada dalam database.

```
datajson
1 {
2   "shorten.er/72m4lg1D": "youtube.com",
3   "shorten.er/halo": "instagram.com",
4   "shorten.er/uEa7UjH": "youtube.com",
5   "shorten.er/PraktikumOrkom": "http://sister21.tech/",
6   "shorten.er/7evV90G2": "https://informatika.stei.itb.ac.id/~rinaldi.munir/",
7   "shorten.er/KitabDingDong": "https://informatika.stei.itb.ac.id/~rinaldi.munir/",
8   "shorten.er/m1NnCqTA": "https://akademik.itb.ac.id/home"
9 }
10
Snipped
```

**Gambar 4.4** Isi data.json setelah ditambahkan data baru  
Sumber: Dokumen penulis

Ketika program sudah selesai digunakan, URL singkat yang telah dihasilkan akan disimpan pada data.json, seperti terlihat pada gambar di atas. Dapat dilihat bahwa terdapat dua URL yang sama tetapi memiliki URL singkat yang berbeda. Penyingkat URL hanya memastikan bahwa URL singkat yang dihasilkan unik sehingga setiap URL singkat yang dihasilkan hanya menuju pada satu alamat website yang unik.

## V. KESIMPULAN DAN SARAN

Penggunaan LCG untuk membangkitkan bilangan acak semu secara efektif dan efisien dapat digunakan dalam berbagai hal, salah satunya adalah dalam pembuatan URL acak. Keunggulan utama LCG terletak pada kemampuannya menghasilkan bilangan acak semu dengan cepat, hanya memerlukan satu kali perhitungan saja tiap bilangan yang dibangkitkan. LCG memanfaatkan teori bilangan, hal ini menjadikan nilai yang dihasilkan tampak acak. LCG dapat memberikan variasi dan keunikan yang diperlukan dalam pembuatan URL acak. Proses ini melibatkan pemanfaatan bilangan acak yang dihasilkan oleh LCG untuk membentuk urutan karakter yang terlihat acak. Proses ini kemudian diaplikasikan dalam pembangkit URL acak, menciptakan tautan yang unik dan sulit ditebak.

Untuk penelitian selanjutnya, dapat mencoba metode pembangkit bilangan acak semu atau metode lain untuk menghasilkan urutan karakter acak pada penyingkat URL.

## VI. UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya yang melimpah dalam proses penulisan makalah ini sehingga dapat diselesaikan tepat waktu. Penulis juga mengucapkan terima kasih kepada Ibu Dr. Nur Ulfa Maulidevi, S.T, M.Sc. sebagai dosen mata kuliah IF2120 Matematika Diskrit kelas K01, yang telah memberikan bimbingan dan ilmu pengetahuan yang berharga untuk menyelesaikan makalah ini. Penulis juga berterima kasih kepada orang tua yang telah memberikan dukungan moril dan spiritual, menjadi sumber inspirasi dan motivasi selama proses penulisan makalah.

## REFERENSI

- [1] <https://www.niagahoster.co.id/blog/apa-itu-url/> (Diakses pada 8 Desember 2023)
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/14-Teori-Bilangan-Bagian1-2023.pdf> (Diakses pada 8 Desember 2023)
- [3] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/15-Teori-Bilangan-Bagian2-2023.pdf> (Diakses pada 8 Desember 2023)
- [4] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/16-Teori-Bilangan-Bagian3-2023.pdf> (Diakses pada 8 Desember 2023)
- [5] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/09-Rekursi-dan-relasi-rekurens-\(Bagian1\)-2023.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/09-Rekursi-dan-relasi-rekurens-(Bagian1)-2023.pdf) (Diakses pada 8 Desember 2023)
- [6] [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 8 Desember 2023)

- [7] <https://sproutsocial.com/glossary/url-shortener/> (Diakses pada 8 Desember 2023)
- [8] <https://designpowers.com/blog/url-best-practices> (Diakses pada 8 Desember 2023)
- [9] <https://indonesia.go.id/layanan/kependudukan/sosial/cara-registrasi-isbn-untuk-penerbit-baru> (Diakses pada 8 Desember 2023)
- [10] <https://bitly.com/blog/url-shortening-101-how-it-works-use-cases-and-examples/> (Diakses pada 8 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, 8 Desember 2023



Ahmad Naufal Ramadan  
13522005