

Penerapan Algoritme Base 64 dalam Rancang Bangun Aplikasi *Lihat.In Short URL*

Ade Husni M^{1*}, Made Hanindia Prami S², dan Retno Mumpuni³

Informatika, UPN "Veteran" Jawa Timur, Surabaya, Indonesia

*e-mail *Corresponding Author*: adehusnim@gmail.com

Abstract

The Short URL Application is a program that allows users to convert long URLs into shorter URLs with fewer characters. This program is highly useful in facilitating users to share links through social media, email, or messaging applications. This research aims to develop a short URL model for URL shortening. The system development follows the common stages in the Base64 method, where the user's input URL is transformed into Base64 encoding and converted into a short ID. The functional features of URL shortening will be tested using Kappa Cohen. The test results show that the features perform well and provide high accuracy.

Keywords: Base64; Short URL application; QR-Code; Application Design; URL shortening

Abstrak

Aplikasi Short URL adalah sebuah program yang memungkinkan pengguna untuk mengonversi URL panjang menjadi URL pendek dengan karakter yang lebih sedikit. Program ini sangat berguna dalam memudahkan pengguna dalam berbagi link melalui media sosial, email, atau aplikasi pesan singkat. Penelitian ini bertujuan untuk mengembangkan model *short url* dalam melakukan pemendekan url. Pengembangan sistem mengikuti tahapan-tahapan umum dalam metode Base64, dimana *url* yang diinput *user* akan dirubah menjadi *base64 encoding* dan diubah menjadi *short id*. Fitur fitur fungsional dalam pemendekkan url akan diujikan menggunakan *Kappa Cohen*. Hasil uji menunjukkan fitur berjalan baik, dan memberikan akurasi yang tinggi

Kata kunci: Base64; Aplikasi Short URL; QR-Code; Rancang Bangun Aplikasi; Pemendekan URL

1. Pendahuluan

Saat ini penggunaan teknologi sudah menjadi bagian penting dari kehidupan manusia sehari-harinya seperti penggunaan teknologi *web* yang salah satu kegunaannya yaitu untuk membagikan informasi. Penggunaan teknologi *web* saat ini semakin meningkat karena adanya pandemi *Covid-19* yang mengharuskan penerapann *physical distancing* [1]. *Web* yang merupakan singkatan dari *World Wide Web* memberikan pengguna akses ke *array* yang luas dari dokumen yang terhubung satu sama lain melalui *hypertext* atau *hypermedia link*.

Karena seringnya *link* URL digunakan pada kehidupan sehari-hari apalagi saat masa pandemi ini yang membuat semua kalangan baik anak kecil hingga lansia harus mengakses *link* URL maka kebutuhan akan *link* URL yang efektif dan efisien terus meningkat karena tingkat keinginan individu untuk menggunakan teknologi *online* termasuk *web* dipengaruhi salah satunya oleh kemudahan penggunaan [2]. Termasuk di dalamnya adalah kemudahan menuju *web* yang dimaksud. Itulah mengapa saat ini terdapat *link shortening service* contohnya seperti *bit.ly* maupun *s.id* yang memberikan solusi atas panjangnya *link* URL dari *web* sehingga membantu mempercepat proses pengetikan dari URL yang panjang dan semua proses yang melibatkan *link* URL lainnya. Kebutuhan akan *link shortening service* yang tinggi terbukti oleh data dari *homepage web s.id* yang menyatakan bahwa per harinya ada 900 ribu orang yang mengunjungi *web s.id* [3].

Fakultas Ilmu Komputer Universitas Pembangunan Negara (UPN) Veteran Jawa Timur telah memiliki *link shortening service* sendiri yaitu aplikasi *Lihat.in* yang berbasis *web*. Namun pada awal tahun 2020 aplikasi *Lihat.in* sudah tidak lagi *dimanage* oleh mahasiswa Fakultas Ilmu Komputer UPN Veteran Jawa Timur karena sudah tidak sesuai dengan kebutuhan. Dengan demikian dipandang perlu mengembangkan sistem baru.

Algoritme *Base64* adalah metode pengkodean yang dapat digunakan untuk mengubah data biner (file, teks, dll.) menjadi teks yang terdiri dari karakter-karakter ASCII. *Base64* merupakan representasi data biner dalam bentuk teks yang dapat dengan mudah ditransmisikan melalui protokol yang hanya menerima teks biasa, seperti email atau protokol HTTP [4]. Algoritme ini telah banyak diuji penggunaannya dalam desain sistem *url* seperti dalam [5]-[7]

Penelitian ini bertujuan untuk membangun kembali aplikasi *Lihat.in* menggunakan algoritme *Base64* untuk menyempurnakan aplikasi sejenis yang saat ini sudah tidak dapat memenuhi kebutuhan.

2. Tinjauan Pustaka

Sebelum melakukan penelitian ini, telah dilakukan peninjauan beberapa penelitian terdahulu untuk dijadikan referensi yang akan dihimpun menjadi sebuah data yang digunakan untuk membantu dalam penulisan ini

Jurnal pertama [8] pada penelitian tersebut dilakukan analisis terhadap keamanan data dengan menggunakan 2 metode Algoritme yaitu *Base64* dan *Ha* dengan perbedaan karakter yang signifikan. Yang dimana dalam enkripsi sebuah gambar yaitu, 2600 bit karakter yang dihasilkan oleh *base64* sedangkan 256 bit yang dihasilkan oleh *HAVAL*.

Jurnal kedua [9] pada penelitian tersebut dilakukan analisis terhadap beberapa file yang dikonversi menjadi teks menggunakan Algoritme *base64*, beberapa file yang telah dikonversi kemudian digabungkan dengan cover media dengan *EOF* method, size dalam media yang telah dikonversi menjadi bertambah, yang dimana *base64* dapat digabungkan dengan metode *EOF*.

Jurnal ketiga [10] pada penelitian ini bertujuan untuk memfilter *Image Spam* dengan menggunakan *base64* dan *N-Gram*, dalam permasalahan tersebut difokuskan dalam pesan spam yang berbentuk gambar di dalam email, yang dimana pada waktu itu sangat susah untuk mendeteksi pesan spam berbentuk gambar. Dari permasalahan tersebut digunakan metode *base64* untuk mengubah "image" menjadi *base64* string lalu diproses dengan *N-Gram String Vector* dan tahap terakhir dilakukan klasifikasi *SVM* dengan 2 parameter spam dan non-spam. Dari tahap pemrosesan tersebut diperoleh akurasi 97,5% dan 98,4%.

Jurnal keempat [11] pada penelitian ini metode integrasi *base64* yang digunakan untuk mendeteksi iklan dalam saluran acara tv. Dalam proses klasifikasi iklan, dengan cara encode setiap pixel yang telah dipilih dan dibandingkan secara berurutan untuk mengurangi komputasi waktu. Dalam penelitian tersebut, berhasil menunjukkan klasifikasi yang telah dibandingkan, dalam 152 detik video file yang telah di tes terdapat 10 detik iklan kfc. Akurasi dalam Teknik *base64* dengan 60% presisi.

Penelitian-penelitian terdahulu menyoroti penggunaan metode *Base64* dalam berbagai konteks, termasuk keamanan data, penggabungan teks dengan media, deteksi spam berbasis gambar, dan deteksi iklan dalam video. Pada penelitian ini melihat pengaruh kualitas encoding *base64*, pada aplikasi *lihatin* serta kepuasan pengguna berdasarkan kebutuhan pengguna sehingga pengukuran yang dilakukan ini akan membantu pengelola aplikasi untuk menyesuaikan kualitas aplikasi sesuai dengan kebutuhan dan keinginan pengguna.

3. Metodologi

3.1 Algoritme *base64*

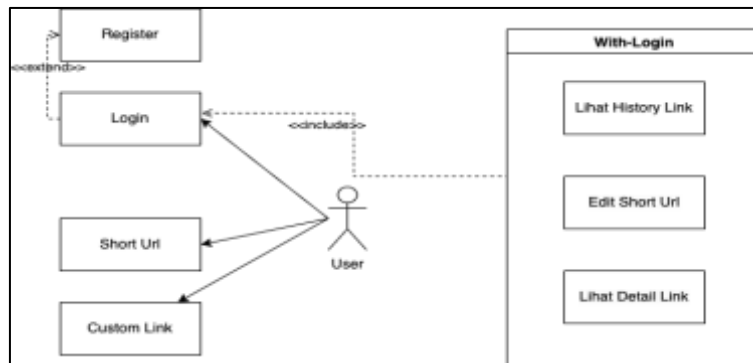
Algoritme *base64* disajikan berikut:

- 1) Ubah alfanumerik menjadi kode ASCII.
- 2) Ubah kode ASCII menjadi bilangan biner.
- 3) Bagi bilangan biner menjadi 3 blok byte.
- 4) Bagi data menjadi 4 Grup, yang dimana setiap grup bernilai 6bits.
- 5) Masing masing pecahan diubah ke dalam nilai desimal, dimana maksimal nilai 6 bit adalah 63.
- 6) Terakhir, jadikan nilai-nilai desimal tersebut menjadi index untuk memilih maksimal index ke 64 atau karakter ke 63 dari penyusun *base64*.
- 7) Apabila dalam proses encoding terdapat sisa pembagi, maka tambahkan karakter pad(=) sebagai penggenap sisa tersebut

3.2 Model Fungsional Sistem

Sebelum dilakukan pembuatan dan implementasi sistem perlu adanya rancangan dan gambaran umum sistem dengan tujuan agar sistem sesuai dengan kebutuhan dari pelaksana.

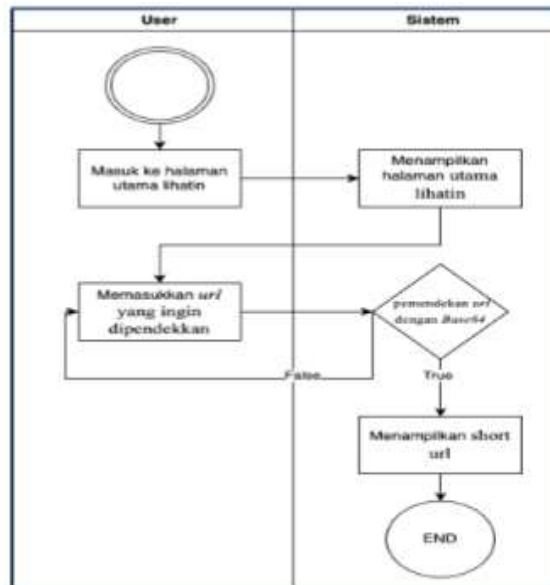
Dalam perancangan sistem yang akan dibuat digunakan *Use Case Diagram* untuk menjelaskan semua rancangan yang telah dibuat.



Gambar 1. *Use Case Diagram* aplikasi Lihat.in

Gambar 1 merupakan *Use case diagram* lihat-in yang menyajikan fitur fungsional utama dalam sistem. Salah satu fitur fungsional akan divalidasi menggunakan *Kappa Cohen*.

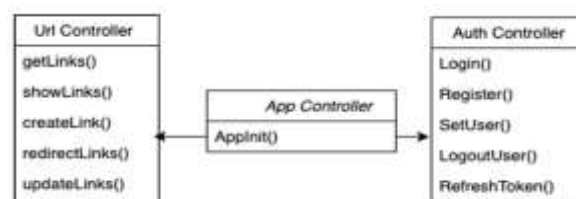
3.3 Model Proses Sistem



Gambar 2. *Activity Diagram Short URL*

Gambar 2 merupakan alur dari aktivitas short url yang memiliki beberapa proses untuk dapat memendekkan url pada sistem. Pertama user masuk ke halaman utama lihatin, kemudian user memasukkan url yang ingin dipendekkan lalu menekan button pendekkan. Setelah itu sistem akan melakukan validasi masukan dari user. Jika masukan salah maka sistem akan memunculkan pesan kesalahan, namun jika masukan benar dan valid maka sistem akan menampilkan halaman detail short url.

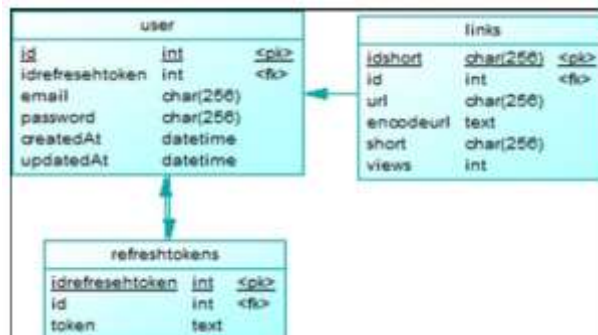
3.4 Struktur Sistem



Gambar 3. *Class Diagram Short URL*

Gambar 3 menjelaskan tentang rangkaian dari class diagram controller yang terdiri dari 3 class yaitu: App Controller merupakan fitur dari Node JS yang berguna untuk menjalankan kode frontend dan backend; Auth controller berfungsi untuk mengelola validasi akun yang mengakses sistem; dan Url Controller berfungsi untuk mengelola URL yang telah dibuat oleh user.

3.5 Model Data



Gambar 4. Physical Data Modeling Short URL

Gambar 4 merupakan bentuk PDM (*Physical Data Modeling*) database sistem yang berasal dari konversi CDM. Proses konversi CDM ke PDM ini menggunakan salah satu fitur pada aplikasi power designer. PDM memiliki bentuk yang lebih spesifik karena memperlihatkan relasi tabel dengan identitas yang lebih jelas daripada CDM.

4. Hasil dan Pembahasan

4.1 Antarmuka Sistem Aplikasi

1) Tampilan User Interface Home



Gambar 5. Tampilan Halaman Home

Pada halaman Gambar 5, user dapat memasukkan *url* yang ingin dipendekkan melalui aplikasi, apabila user telah melakukan input *url*, user dapat menekan button pendekan, atau user dapat menggunakan custom link yang berada dibawah inputan url. Setelah user memasukkan url dan menekan button pendekan, maka hasil dari link menjadi pendek.

Gambar 6 merupakan tampilan home yang telah melakukan input url maupun short id yang ingin *dicustom*. Pengguna dapat bebas memilih *short id* yang ingin dibuat. Pada hasil output Gambar 6 merupakan hasil dari inputan user, user memasukkan short id yang dijadikan hasil dalam short url yang dibuatnya. Contoh: User memasukkan short id dengan karakter "u1" maka hasilnya akan sama persis dengan inputan short id yaitu "u1".

Setelah user mendapatkan hasil short, user dapat memilih ikon sesuai operasi yang akan dilakukan. Salah satu icon yang dapat dipilih adalah icon untuk menampilkan pop-up "QR-Code" pada halaman detail short url, seperti Gambar 7.



Gambar 6. Tampilan Inputan User



Gambar 7. Tampilan Hasil QR-Code Halaman Lihatin

2) Penerapan Algoritme Base64 dan QR-Code

Penerapan metode Base64 dan QR-Code pada aplikasi lihatin berfungsi untuk membuat dan generate short id secara otomatis, sedangkan QR-Code berfungsi untuk menampilkan hasil short url yang telah dibuat dengan menggunakan Algoritme Base64. Dimulai dengan mempersiapkan data url user yang akan dijadikan input system pada Algoritme Base64. Dalam hal ini untuk memberikan hasil short id yang berbeda di setiap pengguna menggunakan aplikasi lihatin.

```
let { url, short } = req.body;

if (short) {
  const oldLink = await Link.findOne({
    where: { short: short },
  })

  if (oldLink) {
    return res.status(404).json({
      status: 'error',
      message: [{
        field: 'short',
      }],
    });
  }
}
```

```

    message: 'short link already exists'
  })
});
}
} else {
  hasil = Base64.encode(url)
  for (var i = 0; i < 1; i++) {
    var hasilAlgo = `${StringGen(Math.random() * (5 - 1) + 1)}`;
  }
  short = hasilAlgo;
  encodeurl = hasil
}

```

Gambar 8. Kode untuk Pencaeian dan Pencocokan *short id*

Didalam kode tersebut, mencari short url yang telah dibuat, dicocokkan dengan short id yang telah ada didalam database, jika ditemukan maka akan muncul error "Short Already Exist". Jika tidak ditemukan didalam database, maka url dari inputan user diubah menjadi Base64, setelah itu akan diacak hasil tersebut dengan fungsi `math.random` yang dimana maks karakter yang dihasilkan adalah 5 karakter short id, dan hasil encode base64 dimasukkan kedalam database.



Gambar 8. Base64 dalam Aplikasi Lihatin

Gambar 8 merupakan hasil pemrosesan short url dengan menggunakan Algoritme base64, yang dimana dalam aplikasi tersebut menampilkan sebuah pop-up message.

3) Fungsi QR-Code

Penerapan QR-Code Pada Aplikasi lihatin bertujuan untuk mempermudah pengguna untuk, melihat serta menggunakan short id yang telah dibuat. QR-Code pada aplikasi lihatin, menggunakan library javascript yang bernama "qrcode".

```

QRCode.toFile(`./public/images/qr/${link.id}.png`,
`${process.env.BASE_URL}/${link.short}`,
{ type: 'png', errorCorrectionLevel: 'M', width: 300, margin: 1 });

```

Gambar 9. QR-Code pada aplikasi lihatin menggunakan library javascript

Kode pada Gambar 9 merupakan implementasi qr-code didalam aplikasi lihatin, salah satu fungsi `QRCode.toFile`, berfungsi sebagai membuat image QR-Code yang akan disimpan didalam folder public, serta menyimpan dengan ekstistensi file .png dengan ukuran 300 x 300 px

dengan margin 1. Didalam gambar png tersebut, berisikan parameter url base serta short id yang telah dibuat.

4.2 Validasi *Kappa Cohen*

Pada penelitian ini digunakan metode Kappa Cohen untuk memvalidasi chatbot ini dengan aturan sebagai berikut:

- 1) Pengujian dilakukan oleh dua mahasiswa UPN Veteran Jatim. Penguji 1 merupakan seorang Mahasiswa di Fakultas FIK. Sedangkan, penguji 2 seorang Mahasiswa Sistem Informasi.
- 2) Pengujian dilakukan pada 15 url untuk menilai Shortener url tersebut.
- 3) Dengan ketentuan Aplikasi dapat melakukan short url terhadap inputan user. Jika hasil short setelah dilakukan pengecekan oleh penguji 1 dan penguji 2 pada realtime memang benar maka bernilai 1, jika tidak maka bernilai 0.

Tabel 1. Tabel Pengujian Kappa Cohen

URL	Rater	
	Penguji 1	Penguji 2
https://pelatihan-ui.com/uji-konsistensi-cohens-kappa/	1	1
https://www.youtube.com/watch?v=z4CiQPV0Mgw&t=206s	1	1
https://www.upnjatim.ac.id/	1	1
https://fasilkom.upnjatim.ac.id/	1	1
https://febis.upnjatim.ac.id/	1	1
https://faperta.upnjatim.ac.id/	1	1
https://ft.upnjatim.ac.id/	1	1
https://fisip.upnjatim.ac.id/	1	1
https://fad.upnjatim.ac.id/	1	1
http://fhukum.upnjatim.ac.id/	1	1
http://repository.upnjatim.ac.id/8400/1/18081010072_Bab1.pdf	1	1
http://repository.upnjatim.ac.id/11766/3/18081010133_BAB%202.pdf	0	0
http://repository.upnjatim.ac.id/8455/2/18081010021_Bab1.pdf	1	1
http://repository.upnjatim.ac.id/10393/4/18081010055-bab2.pdf	1	1
http://repository.upnjatim.ac.id/7668/3/18081010043-Bab2.pdf	1	1

Dari 15 URL yang telah di pendek-kan pada tabel 1, menghasilkan persentase sebesar 93,4% dari penguji 1 dan penguji 2 dalam pemendekan url.

Tabel 2. Tabel Hasil Pengujian Kappa Cohen

		Penguji 1	
		Positif	Negatif
Penguji 2	Positif	14	0
	Negatif	0	1

Perhitungan *Kappa Cohen* menggunakan rumus

$$E_{11} = \frac{(a+b).(a+c)}{n}$$

$$E_{11} = \frac{(14+0).(14+0)}{15}$$

$$E_{11} = 13,067$$

$$E_{12} = \frac{(c+d).(b+d)}{n}$$

$$E_{12} = \frac{(0+1).(0+1)}{15}$$

$$E_{12} = 0,133$$

$$P_o = \frac{a-d}{n}$$

$$P_o = \frac{14+1}{15}$$

$$P_o = 1$$

$$P_e = \frac{E_{11}+E_{12}}{n}$$

$$P_e = \frac{13,067+0,133}{15}$$

$$P_e = 0,88$$

$$K = \frac{P_o - P_e}{1 - P_e}$$

$$K = \frac{1 - 0,88}{1 - 0,88}$$

$$K = 1$$

Hasil tersebut merupakan hasil perbandingan kappa kohen terhadap penguji 1 dan 2, bahwa nilai koefisien realibitas antar rater dengan uji Kappa mendapatkan nilai atau value 0,87. Dari Perhitungan tersebut, koefisien nilai Kappa dapat diinterpretasikan dalam kategori Sempurna karena nilai koefisien Kappa antara $\kappa = 1$. Dengan demikian terdapat kesepakatan antar ahli yang kuat, maka 15 data url dikatakan telah valid dan reliabel. Artinya, aplikasi lihatin ini layak digunakan dikarenakan memberikan hasil short url yang valid terhadap user.

5. Simpulan

aplikasi Lihat.in, yang menggunakan teknologi web dan Algoritme Base64, telah berhasil memberikan layanan enkripsi URL yang aman serta efisien. Sistem yang dikembangkan juga membantu meningkatkan proses transformasi URL menjadi short URL dengan baik dan aman. Pengujian yang dilakukan menunjukkan bahwa sistem Lihat.in berjalan dengan baik dan memberikan hasil akurasi yang tinggi.

Daftar Referensi

- [1] Adam Volle, "URL," *Encyclopedia Britannica*, Jan. 27, 2023.
- [2] Utami Pengertian Node JS. Dicoding, 2021, <https://www.dicoding.com/blog/apa-itu-node-js/>
- [3] PT Aidi Digital Global, *Penyingkat tautan dan platform Microsite terpendek*, <https://home.s.id/id>, diakses pada 19 April 2022

- [4] E. Gunadhi, & A.P. Nugraha, "Penerapan Kriptografi Base64 Untuk Keamanan URL (Uniform Resource Locator) Website Dari Serangan SQL Injection". *Jurnal Algoritma*, vol. 13, no. 2, pp. 391-398, 2016.
- [5] D. Muliawan, & R.R. Astanti, "Mencegah Exploit URL Pada Model Business to Customer Pada Toko Citra Ponsel Ketapang". In *ENTER*. Vol. 1, No. 1, pp. 24-35, 2018.
- [6] N. P. B. Mese, & V. Wibowo, "Perancangan Website UTI-PRO KALBAR Dengan Menggunakan Algoritma Base64 Untuk Mengamankan Database". In *ENTER*, vol. 2, No. 1, pp. 136-150, 2019.
- [7] A. Latif, A.I. Warnilah, & S.K. Wildah, "Implementation Of The Rijndael Algorithm On Web-Based Whistleblowing System". *Jurnal Techno Nusa Mandiri*, vol. 19, no. 2, pp. 141-148, 2022.
- [8] M. Mesran *et al.*, "Combination Base64 and Hashing Variable Length for Securing Data," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jun. 2018. doi: 10.1088/1742-6596/1028/1/012056.
- [9] R. Rahim *et al.*, "Combination Base64 Algorithm and EOF Technique for Steganography," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Apr. 2018. doi: 10.1088/1742-6596/1007/1/012003.
- [10] C. Xu, Y. Chen, and K. Chiew, "An approach to image spam filtering based on Base64 encoding and N-gram feature extraction," in *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2010, pp. 171–177. doi: 10.1109/ICTAI.2010.31.
- [11] M. Faraz Hyder, M. Andleeb Siddiqui, and M. Mukarram, "TV Ad Detection Using the Base64 Encoding Technique," 2021. [Online]. Available: www.etasr.com