

Aplikasi Kompresi Pesan Teks Pada Platform Android Menggunakan Algoritma Lz77

Richard Simanjuntak

Sistem Informasi, Universitas Tjut Nyak Dhien, Sumatera Utara, Indonesia

Email: richardsimanjuntak@gmail.com

Email Penulis Korespondensi: richardsimanjuntak@gmail.com

Abstrak—Perkembangan teknologi komunikasi pada era ini sudah sangat pesat dan maju hal itu dibuktikan dengan semakin banyak produksi peralatan-peralatan komunikasi baru dengan berbagai macam sistem operasi. Dengan kecanggihan alat-alat komunikasi pada saat ini membuat handphone atau alat komunikasi lainnya mempunyai fungsi yang hampir sama dengan komputer, bahkan perangkat keras yang tertanam didalam peralatan komunikasi sekarang juga sudah mendekati spesifikasi komputer pada umumnya. Android adalah jenis sistem operasi handphone yang paling digemari pada era ini dikarenakan tampilan yang menarik dan juga relatif murah serta mudah untuk digunakan. Walaupun Android adalah sistem operasi yang pintar dan sangat canggih, android masih memakai fasilitas sms untuk mengirim pesan teks, sama seperti sms konvensional yaitu karakter pesan masih saja terbatas dan pesan dikirim berdasarkan layar yaitu sms dikirim tiap 160 karakter, jika sms berkapasitas 2 layar atau lebih dari 160 karakter, maka akan dikenakan tarif 2 kali sms. Tentu hal ini akan memakan biaya yang lebih besar karena sama saja dengan mengirim sms dua kali. Algoritma kompresi Lz77 adalah sebuah algoritma kompresi teks yang dapat memperkecil ukuran teks dengan mengurangi komponen teks yang sama, sehingga ukuran teks akan lebih ringkas.

Kata Kunci: Perkembangan teknologi komunikasi, Sistem operasi Android, Kompresi teks Lz77, Kompresi pesan teks, Efisiensi pengiriman SMS

Abstract—The development of communication technology in this era has been very rapid and advanced, as evidenced by the increasing production of new communication equipment with various operating systems. With the sophistication of communication tools at this time, mobile phones or other communication tools have almost the same function as computers, even the hardware embedded in communication equipment is now close to computer specifications in general. Android is the most popular type of mobile phone operating system in this era because of its attractive appearance and also relatively cheap and easy to use. Even though Android is a smart and very sophisticated operating system, Android still uses the SMS facility to send text messages, just like conventional SMS, namely message characters are still limited and messages are sent based on the screen, namely SMS is sent every 160 characters, if the SMS has a capacity of 2 screens or more than 160 characters, it will be charged 2 times the rate of SMS. Of course, this will cost more because it is the same as sending an SMS twice. The Lz77 compression algorithm is a text compression algorithm that can reduce the size of text by subtracting the same text components, so that the text size will be more compact.

Keywords: Communication technology development, Android operating system, Lz77 text compression, Text message compression, SMS delivery efficiency

1. PENDAHULUAN

Perkembangan teknologi komunikasi pada era ini sudah sangat pesat dan maju hal itu dibuktikan dengan semakin banyak produksi peralatan-peralatan komunikasi baru dengan berbagai macam sistem operasi. Dengan kecanggihan alat-alat komunikasi pada saat ini membuat handphone atau alat komunikasi lainnya mempunyai fungsi yang hampir sama dengan komputer, bahkan perangkat keras yang tertanam didalam peralatan komunikasi sekarang juga sudah mendekati spesifikasi komputer pada umumnya. Android adalah jenis sistem operasi handphone yang paling digemari pada era ini dikarenakan tampilan yang menarik dan juga relatif murah serta mudah untuk digunakan. Walaupun Android adalah sistem operasi yang pintar dan sangat canggih, android masih memakai fasilitas sms untuk mengirim pesan teks, sama seperti sms konvensional yaitu karakter pesan masih saja terbatas dan pesan dikirim berdasarkan layar yaitu sms dikirim tiap 160 karakter, jika sms berkapasitas 2 layar atau lebih dari 160 karakter, maka akan dikenakan tarif 2 kali sms. Tentu hal ini akan memakan biaya yang lebih besar karena sama saja dengan mengirim sms dua kali.[1]

Algoritma kompresi Lz77 adalah sebuah algoritma kompresi teks yang dapat memperkecil ukuran teks dengan mengurangi komponen teks yang sama, sehingga ukuran teks akan lebih ringkas. Algoritma kompresi Lz77 melakukan kompresinya dengan mengganti porsi data dengan suatu referensi untuk menyamakan data yang sudah dilewati oleh encoder dan decoder. Algoritma ini menggunakan suatu “sliding window” (Muller 2008) yang terdiri dari search buffer dan look-ahead buffer. Search buffer digunakan sebagai kamus, sedangkan look-ahead buffer merupakan buffer yang berisi string yang akan dikompresi. Search buffer merupakan buffer yang sudah melalui tahap kompresi, namun digunakan sebagai kamus. Penggunaan search buffer dan look-ahead buffer dapat diatur sesuai dengan kemampuan sistem. Yaitu dengan menentukan panjang array masing-masing dari search buffer dan look-ahead buffer sehingga menjadikan algoritma kompresi Lz77 sangat cocok digunakan pada mobile programming seperti Android. Selain untuk

mengompresi teks algoritma lz77 juga dapat digunakan untuk mengkompresi file karena lz77 dapat menkompresi biner ataupun byte, sehingga algoritma ini sangat dinamis dan praktis[2].

Dengan penerapan kompresi pada handphone Android, maka metode yang cocok untuk pengkompresian dan dekompresi pada handphone Android adalah algoritma kompresi lz77. Alasan pemilihan algoritma lz77 adalah bahwa algoritma lz77 memberikan performa terbaik untuk beberapa jenis file dibandingkan algoritma-algoritma modifikasinya[3].

2. METODOLOGI PENELITIAN

2.1 Analisa Perancangan

Untuk mendapatkan hasil yang maksimal haruslah dimulai dengan melakukan sebuah analisa tentang perancangan. Perancangan bertujuan untuk membuat konsep dari sistem yang dibuat. Dari sebuah konsep yang jelas akan mempermudah tahap lanjutan untuk membuat design nyata yang akan digunakan untuk pembuatan sebuah sistem. Untuk proses tahapan lanjut pembuatan tulisan ini, penulis terlebih dahulu menganalisa tentang perancangan yang dibuat. Adapun analisa dari perancangan dari tulisan ini adalah : analisa mengenai perancangan sistem, perancangan input, perancangan output dan menganalisa perancangan tampilan sistem[4].

2.2 Analisa Sistem

Analisa sistem bertujuan untuk mendapatkan sebuah gambaran atau design yang tepat mengenai cara kerja sebuah sistem. Dengan adanya gambaran yang bagus mengenai sebuah sistem, berarti juga menghasilkan sebuah sistem yang baik juga. Analisa sistem ini membahas mengenai bagaimana proses kerja dari sebuah sistem dari awal sampai akhir, yang dijelaskan dengan bahasa sederhana dan juga jelas[5]. Analisa sebuah sistem juga sangat mempengaruhi kinerja dari sebuah sistem yang dihasilkan. Karena kinerja sebuah sistem sangat dipengaruhi oleh seberapa jauh permasalahan yang akan diselesaikan oleh sebuah sistem. Semakin luas cakupan masalah sistem, maka semakin tidak maksimal juga kinerja sebuah sistem[6].

2.3 Analisa dan Logika Metode

Pada pembahasan analisa dan logika metode, penulis membahas mengenai bagaimana teknik kompresi dan dekompresi pesan teks (string). Pada bagian kompresi penulis menerangkan cara merancang search buffer dan look-ahead buffer, cara mencari string yang sama dan bagaimana menyiapkan output data menjadi string kembali. Pada bagian dekompresi dibahas cara memisahkan offset, panjang karakter dan data karakter yang berbentuk biner serta cara mendekompresikan kembali pesan seperti semula[7].

2.3.1 Kompresi Pesan

Dalam pengkompresian pesan mempunyai empat tahapan yaitu :

- Merancang Search Buffer dan Look-ahead Buffer
Metode ini melakukan pencarian satu (atau lebih) simbol pada look-ahead buffer yang sama dengan simbol pada search buffer. Pencarian dilakukan dari kanan ke kiri. Praktek nya, perbandingan ukuran search buffer dengan look-ahead buffer kira-kira adalah beberapa ribu banding 10[8].
Algoritma ini melakukan pencarian terhadap kumpulan simbol terpanjang yang sama pada awal look-ahead buffer terhadap search buffer dan mengeluarkan sebuah pointer terhadap kesamaan tersebut. Secara umum, sebuah token lz77 memiliki tiga buah bagian yakni : offset, panjang dan simbol selanjutnya pada look-ahead buffer. Offset dan panjang merupakan pointer yang menunjukan posisi dan jumlah kecocokan. Apabila ditemukan kecocokan, lz77 menambahkan suatu pointer dengan suatu simbol dibelakangnya. Apabila tidak ada yang cocok, maka dihasilkan null pointer dan simbol unik. Selanjutnya sliding window digeser sebanyak simbol yang terkompresi berdasarkan pointernya[9].
Berikut ilustrasi metode kompresi lz77 :
Misalkan kata yang ingin dikompresi adalah “beberapabek” dapat dilihat pada tabel 1 berikut[9].

Tabel 1. Simulasi Lz77 terhadap string

7	6	5	4	3	2	1	(OUTPUT)
							b e b e r apabek

Search buffer

look-ahead buffer

2. Mencari karakter sama

Misalkan kata yang akan dikompresi adalah “beberapabek”, maka karakter akan dicari karakter yang sama pada tabel berikut :

Tabel 2. Simulasi Lz77 terhadap string

7	6	5	4	3	2	1		(OUTPUT)
							b e b e r apabebek	(0,0,b)
					b		e b e r a pabebek	(0,0,e)
				b	e		b e r a p abebek	(2,2,r)
	b	E	b	e	r		a p a b e b ek	(0,0,a)
	b	e	B	e	r	a	p a b e b Ek	(0,0,p)
B	e	b	E	r	a	p	a b e b e K	(2,1,b)
B	e	r	A	p	a	b	e b e k	(6,1,b)
R	a	p	A	b	e	b	e k	(2,1,k)
P	a	b	e	b	e	k		

Search buffer

look-ahead buffer

3. Mengubah Hasil Menjadi Biner

Untuk mengubah hasil dari tabel 2 menjadi bilangan biner, maka dapat dilihat pada tabel berikut:./

Tabel 3. Hasil Menjadi Biner

(output)	Output Dijadikan Biner		
	OFSET	PANJANG	Karakter
(0,0,b)	000	000	01100010
(0,0,e)	000	000	01100101
(2,2,r)	010	010	01110010
(0,0,a)	000	000	01100001
(0,0,p)	000	000	01110000
(2,1,b)	010	001	01100010
(6,1,b)	110	001	01100010
(2,1,k)	010	001	01101011

Untuk membedakan bit mana yang menunjukkan posisi dan panjang dengan simbol, diberikan tanda dengan menambahkan 1 bit. Bit 0 menunjukkan bahwa 8 bit selanjutnya dari perogram adalah simbol, sedangkan bit 1 menunjukkan bahwa 6 bit selanjutnya berisi offset dan panjang simbol yang diikuti 8 bit isi simbolnya. Maka dari tabel 3 akan ditemukan hasil sebagai berikut :

0 01100010 0 01100101 1 010 010 01110010 0 01100001 0 01110000 1 010 001 01100010 1 110
001 01100010 1 010 001 01101011

4. Mendapatkan Karakter Kompresi

Setelah mendapatkan hasil yang telah di Biner kan, maka bilangan Biner dipisah menjadi 8 bit, sehingga menghasilkan karakter yang terkompresi.

0 01100010 0 01100101 1 010 010 01110010 0 01100001 0 01110000 1 010 001 01100010 1 110
001 01100010 1 010 001 01101011

Bilangan dipisah menjadi 8 bit.

00110001 00011001 01101001 00111001 00011000 01001110 00010100
01 011000 10111000 10110001 01010001 01101011

Biner diatas dirobah menjadi karakter, yaitu mendapatkan hasil bilangan desimal pada tabel berikut :

Tabel 4 hasil bilangan desimal dari bilangan biner

49	25	105	57	24
78	20	88	184	177
81	107			

Maka karakter yang dihasilkan dari tabel 4 yaitu sebagai berikut :

Tabel 5 hasil karakter dari bilangan desimal

1	f	i	9	↑
N	q	X	.	±
Q	k			

Jadi karakter pada tabel 2.5 adalah pesan yang akan dikirim

5. Mendapatkan nilai rasio kompresi

Rasio atau dalam istilah perbandingan adalah sebuah nilai selisih antara ukuran baru dengan ukuran lama, sehingga diketahui jelas angka dan nilai perbedaan antara dua ukuran tersebut. Didalam kompresi hamper dipastikan bahwa ukuran bit data sebelum dikompresi dengan ukuran bit setelah dikompresi berbeda atau tidak sama, maka sangat diperlukan untuk membuat nilai rasio atau perbandingan ini. Dibawah ini adalah rumus untuk mencari nilai rasio dalam persen :

$$\text{ratio} = (\text{ukuran lama} - \text{ukuran baru}) / (\text{ukuran lama}) \times 100$$

Jadi jika dikembalikan kepada data hasil kompresi diatas maka akan didapat nilai-nilai sebagai berikut :

Ukuran lama = 104 bit

Ukuran baru = 96 bit

$$\text{ratio} = (104 - 96) / 104 \times 100 = 7,69\%$$

Maka nilai ratio dari rumus di atas dapat ditentukan = 7,69%

2.3.2 Dekompresi Pesan

1. Merubah Pesan Menjadi Biner

Pesan yang diterima akan dirobah karakter menjadi biner yaitu dengan cara sebagai berikut :

Tabel 6 Pesan diterima

1	f	i	9	↑
N	q	X	.	±
Q	k			

Dari karakter pada tabel 6, maka akan dirobah terdahulu ke bilangan desimal yaitu pada tabel berikut :

Tabel 7 hasil bilangan desimal dari karakter

49	25	105	57	24
78	20	88	184	177
81	107			

Dengan bilangan desimal pada tabel 3.7 maka, bilangan biner dapat ditentukan dengan mudah, yaitu sebagai berikut :

00110001	00011001	01101001	00111001	00011000
01001110	00010100	01 011000	10111000	10110001
01010001	01101011			

1. Memisahkan offset, panjang dan karakter

Untuk memisahkan offset, panjang dan karakter dengan biner yang sudah didapatkan, maka dapat ditentukan dengan memisahkan tanda satu bit, yaitu sebagai berikut :

00110001	00011001	01101001	00111001	00011000
01001110	00010100	01 011000	10111000	10110001
01010001	01101011			

Hasil dari pemisahan tanda sebagai berikut :

0 01100010 0 01100101 1 010 010 01110010 0 01100001 0 01110000 1 010 001 01100010 1 110 001 01100010 1 010 001 01101011

Jika bit pertama ditemukan angka 0, maka 8 bit selanjutnya adalah karakter dan selanjutnya jika bit pertama adalah angka 1, maka 6 bit selanjutnya adalah biner dari offset dan panjang, 8 bit selanjutnya adalah karakter.

Tabel 8 biner dari offset, panjang dan simbol menjadi karakter

OFFSET	PANJANG	SIMBOL	(output)
0	0	1100010	(0,0,b)
0	0	1100101	(0,0,e)
10	10	1110010	(2,2,r)
0	0	1100001	(0,0,a)
0	0	1110000	(0,0,p)
10	1	1100010	(2,1,b)
110	1	1100010	(6,1,b)
10	1	1101011	(2,1,k)

2. Mengambil Pesan

Cara untuk mengambil pesan, maka pesan yang terkompresi akan dikembalikan lagi ke pesan semula (pesan aslinya), dengan langkah-langkah sebagai berikut ini :

Tabel 9 pengembalian pesan

(0,0,b)	b
(0,0,e)	be
(2,2,r)	beber
(0,0,a)	bebera
(0,0,p)	beberap
(2,1,b)	beberapab
(6,1,b)	beberapabeb
(2,1,k)	beberapabebek

Jadi pesan dapat dikembalikan dengan proses pada tabel 3.9, sehingga pesan bisa kembali lagi menjadi karakter semula atau sebelum dikompresikan “beberapabebek”, dan pesan tersebut tidak mengalami kecacatan atau kerusakan sedikit pun. Algoritma lz77 salah satu algoritma kompresi pesan yang bisa mengembalikan pesan pada semula atau sebelum dikompresi.

3. HASIL DAN PEMBAHASAN

3.1 Algoritma

Algoritma adalah urutan langkah-langkah penyelesaian masalah menurut logika masing-masing orang dalam berfikir. Yang terpenting adalah dalam pembuatan algoritma langkah-langkah yang dibuat haruslah sesuai dengan logika, karena komputer tidak dapat memproses sesuatu yang berhubungan dengan perasaan atau perkiraan semata.

Dalam tahap ini dibahas algoritma mengenai pembentukan pesan terkompresi dan dekompresi. Pembahasan ini mencakup algoritma input, output dan proses serta return atau nilai kembali terhadap proses.

Input membahas mengenai proses atau variabel yang digunakan sebagai proses atau masukkan, sedangkan output membahas mengenai hasil yang akan dikeluarkan dan proses membahas mengenai algoritma perhitungan atau pembahasan.

3.1.1.Algoritma Kompresi

3.1.1.1 Merancang Search Buffer Dan Look-Ahead Buffer

Input : karakter
Output : searchBuffer(),lookBuffer()
Proses :
 searchBuffer() = nothing
 For i = 0 to len(karakter)
 lookBuffer = karakter
 next i
 lookBuffer,searchBuffer()

3.1.1.2 Mencari Karakter Sama

Input : searchBuffer(),lookBuffer()
Output : hasil()
Proses :
 varJlhSama = 0
 varIndeks
 For i = 0 to Ubound(lookBuffer)
 For j = 0 to Ubound(searchBuffer)
 If(left(lookBuffer(),i)) = left(searchBuffer(),j) then
 varJlhSama = i
 add.searchBuffer() = left(lookBuffer(),i)
 hasil(varIndeks)(0) = mid(lookBuffer(),1,i)
 hasil(varIndeks)(1) = mid(lookBuffer(),1,i)
 hasil(varIndeks)(2) = mid(lookBuffer(),1,i)
 varIndeks=varIndeks+1
 endif
 Next J
 next i
 hasil()

3.1.1.3 Mengubah Hasil Menjadi Biner

Input : hasil()
Output : dataBiner
Proses :
 For i = 0 to uBound(hasil)
 dataBiner = dataBiner & biner(ascii(hasil(i)(0))) & _
 biner(ascii(hasil(i)(1))) & _
 biner(ascii(hasil(i)(0)))
 next i
 dataBiner

3.1.1.4 Mendapatkan Karakter Kompresi

Input : dataBiner
Output : strKarakter
Proses :
 For i = 0 to len(dataBiner) step 8
 strKarakter = strKarakter & char(binToInt(mid(dataBiner,8,i)))
 next i
 strKarakter

3.1.1.5 Mendapatkan Nilai Rasio Kompresi

Input : karakter, strKarakter
Output : rasioKompresi
Proses :
 rasioKompresi = (len(karakter)-len(strkarakter))/len(karakter)*100
 rasioKompresi

3.1.2 Algoritma Dekompresi

3.1.2.1 Merubah Pesan Menjadi Biner

```
Input      : karakter
Output     : dataBiner
Proses     :
  For i = 0 to len(karakter)
    dataBiner = dataBiner & biner(asc(mid(karakter,1,i)))
  next i
dataBiner
```

3.1.2.2 Memisahkan Offset, Panjang Dan Karakter

```
Input      : dataBiner
Output     : hasil()
Proses     :
  Var i = len(dataBiner)
  Var indeks = 0
  Var Jlh = 0
  While(i > 0)
    If(mid(dataBiner,1,varIndeks) = 0 then
      Hasil(Jlh)(0) = bintToInt(mid(dataBiner,indeks,1))
      Hasil(Jlh)(1) = bintToInt(mid(dataBiner,indeks+1,1))
      Hasil(Jlh)(2) = char(bintToInt(mid(dataBiner,indeks+2,8)))
      Indeks = indeks + 10
      i = i - 10
    Else
      Hasil(Jlh)(0) = bintToInt(mid(dataBiner,indeks,3))
      Hasil(Jlh)(1) = bintToInt(mid(dataBiner,indeks+3,3))
      Hasil(Jlh)(2) = char(bintToInt(mid(dataBiner,indeks+6,8)))
      Indeks = indeks + 14
      i = i - 14
    Endif
    Jlh = jlh+1
  loop
  hasil()
```

3.1.2.3 Mengambil Pesan

```
Input      : hasil()
Output     : pesan
Proses     :
  For i = 0 to uBound(hasil)
    Pesan = pesan & mid(pesan,hasil(i)(0),(hasil(i)(1))
    Pesan = pesan & hasil(i)(2)
  Next i
  pesan
```

3.2 Hasil

Hasil adalah proses penerapan dari sistem baru yang dibuat. Agar dapat berjalan dengan sempurna implementasi memerlukan penjelasan tentang cara kerja sistem dari awal hingga akhir. Selain itu implementasi juga sebagai langkah untuk pengujian suatu sistem apakah sistem mempunyai kekurangan, yang nantinya dari kekurangan itu akan muncul saran untuk memperbaiki sistem menjadi lebih baik.

Untuk menunjang kelancaran implementasi sistem perlu juga dijelaskan adalah bagaimana spesifikasi sistem dan perangkat yang dibutuhkan untuk menjalankan sistem. Untuk lebih jelasnya berikut ini adalah penjelasan dari spesifikasi perangkat keras dan perangkat lunak serta cara kerja sistem :

3.2.1. Spesifikasi Perangkat Keras dan Perangkat Lunak

Sistem ini telah diuji coba, sehingga baik dan lancar apabila perangkat keras dan perangkat lunak yang digunakan mempunyai spesifikasi minimal sebagai berikut :

- Prosesor Intel Pentium Android 600 GHz

- b. Memory 512 Mb
- c. Harddisk minimal 80 Gb
- d. VGA 128 Mb
- e. Monitor dengan resolusi 1.920 x 1.080 pixel
- f. Keyboar dan Mouse
- g. Sistem Operasi berbasis Linux

3.2.2. Cara Kerja Sistem

Pada tahapan ini dibahas mengenai langkah-langkah cara penggunaan sistem yang dibuat. Hal ini berfungsi untuk memandu pengguna untuk menggunakan program, langkah-langkah tersebut adalah sebagai berikut :

3.2.2.1. Proses Kirim Sms

Langkah-langkah untuk kirim pesan dalam program yang dibuat adalah sebagai berikut :

- a. Klik Menu Utama
- b. Klik kirim sms jendela form kirim sms akan terbuka
- c. Masukkan karakter atau pesan
- d. Masukan nomor tujuan
- e. Klik kirim

3.2.2.2. Proses Pesan Masuk

Langkah-langkah untuk proses pesan masuk dalam program yang dibuat adalah sebagai berikut :

- a. Klik Menu Utama
- b. Klik Pesan Masuk jendela form pesan masuk akan terbuka
- c. Klik tombol Pilih untuk membuka pesan masuk yang akan dilihat
- d. Klik keluar untuk keluar dari form Pesan Masuk

3.2.2.3. Proses pesan keluar

Langkah-langkah untuk proses pesan keluar dalam program yang dibuat adalah sebagai berikut :

- a. Klik Menu Utama
- b. Klik Pesan Keluar jendela form pesan keluar akan terbuka
- c. Klik tombol Pilih untuk membuka pesan keluar yang akan dilihat
- d. Klik keluar untuk keluar dari form Pesan Keluar

3.2.3.4 Proses Pesan Tersimpan

Langkah-langkah untuk proses pesan tersimpan dalam program yang dibuat adalah sebagai berikut :

- a. Klik Menu Utama
- b. Klik Pesan Tersimpan jendela form pesan tersimpan akan terbuka
- c. Klik tombol Pilih untuk membuka pesan tersimpan yang akan dilihat
- d. Klik keluar untuk keluar dari form Pesan Tersimpan

4. KESIMPULAN

Berdasarkan pembahasan dan evaluasi dari bab-bab sebelumnya dan analisa terhadap pengujian pesan teks maka dapat diambil beberapa kesimpulan diantaranya proses algoritma kompresi lz77 sebagai salah satu metode mengkompresikan pesan teks yaitu dengan mengkompres pesan dengan menggunakan search buffer dan look-ahead buffer, menerapkan algoritma lz77 dalam aplikasi kompresi pada platform adroid dengan membuat aplikasi melalui android studio, Besar rasio dapat dikatakan tinggi terbukti dari hasil pengujian yang dilakukan

REFERENCES

- [1] T. S. B. Hadi and C. Darujati, "Analisis dan Implementasi Toko Online From. Munch: Studi Kasus Pengembangan Platform E-Commerce," *DIKE: Jurnal Ilmu Multidisiplin*, vol. 1, no. 2, pp. 49–52, 2023.
- [2] M. H. Mahendra, D. T. Murdiansyah, and K. M. Lhaksmana, "Analisis Sentimen Tweet COVID-19 menggunakan K-Nearest Neighbors dengan TF-IDF dan Ekstraksi Fitur CountVectorizer," *DIKE: Jurnal Ilmu Multidisiplin*, vol. 1, no. 2, pp. 37–43, 2023.
- [3] G. Setiawan and G. S. Budi, "Implementasi Metode Forward Chaining Pada Sistem Pakar Untuk Penyakit DBD," *DIKE: Jurnal Ilmu Multidisiplin*, vol. 1, no. 2, pp. 44–48, 2023.

- [4] M. M. Hidayat, "Inovasi Sistem Pembayaran SPP Online untuk Efisiensi Administrasi di SMP Hangtuh 1 Surabaya," *DIKE: Jurnal Ilmu Multidisiplin*, vol. 2, no. 1, pp. 30–36, 2024.
- [5] P. Simanjuntak Richard and M. Sijabat Ramson, Rikson, "Meningkatkan Keamanan Siber dalam Lingkungan Internet of Things (IoT) dengan Menggunakan Sistem Deteksi Intrusi Berbasis Pembelajaran Mesin," *DIKE: Jurnal Ilmu Multidisiplin*, vol. 2, no. 2, pp. 62–68, 2024.
- [6] E. Panggabean and J. R. Sagala, "Analisa Perbandingan Metode Jaringan Syaraf Tiruan Dengan Metode Sistem Pendukung Keputusan Untuk Penerimaan Tenaga Kerja," *JUMIN*, vol. 2, no. 2, pp. 41–44, Jun. 2021, doi: 10.55338/jumin.v2i2.697.
- [7] N. Hafizar, E. R. Syahputra, and D. Irwan, "Desain Dan Penerapan Sistem Informasi Untuk Pemasaran Biji Kopi Dan Bubuk Kopi Arabika Berbasis Android," vol. 4, 2022.
- [8] R. L. Sianturi, "Eksperimen dan simulasi Transien Suhu Pahat intan pada pemesian Titanium (Ti-64l-4V)," *Jurnal Kolaborasi Sains dan Ilmu Terapan*.
- [9] M. A. M. Muda, "Pengaruh Pemikiran Jabir Bin Hayyan terhadap Ilmu Alkimia," *Jurnal Kolaborasi Sains dan Ilmu Terapan*, vol. 2, 2023.