

Implementasi *String Matching* dalam Mendeteksi Pola Komunikasi pada Mahasiswa

Diyah Susan Nugrahani - 13523080

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: diyhannugrahani04@gmail.com , 13523080@std.stei.itb.ac.id

Abstract— *Stress is a common condition caused by the demands of modern life, affecting individuals across all ages, including students. Academic stress arises from the pressure to achieve optimal results within limited timeframes, often observed during exam periods. Behavioral changes, especially in communication patterns such as word choice and tone, can indicate stress levels. This study analyzes chat communication patterns as an early stress detection method among students using string matching and regular expression algorithms. The findings highlight the potential of pattern recognition in identifying stress-related behaviors in digital interactions.*

Keywords—*stress, communication, regular expression, string matching, pattern.*

I. PENDAHULUAN

Stress adalah suatu kondisi yang umum dialami terutama dalam dinamika kehidupan saat ini. Perkembangan dunia yang begitu cepat membuat kehidupan menjadi lebih kompleks dan menuntut manusia untuk terus menyesuaikan diri dengan perkembangan yang ada. Tuntutan dan tekanan yang berlebihan dapat menjadi pemicu munculnya *stress* dalam diri. *Stress* dapat menimpa siapa saja termasuk anak-anak, remaja, dewasa, atau orang lanjut usia. [2] Lin Huang (2014) menyatakan bahwa *stress* yang jumlahnya banyak bisa membahayakan diri, termasuk mahasiswa.

Lingkungan akademik bisa menjadi potensi yang menimbulkan stres akademik pada mahasiswa. [1] Busari (2012) mengemukakan bahwa stres akademik merupakan suatu tekanan yang muncul karena siswa memiliki keinginan untuk memberikan hasil yang terbaik dalam ujian, namun adanya Batasan waktu yang membuat lingkungan akademik siswa menegang. Hal ini mendasari alasan mahasiswa cenderung mengalami stres ketika menjelang musim ujian.

Orang yang mengalami stres memiliki perubahan perilaku dalam kesehariannya. Cara berkomunikasi mereka juga mengalami perubahan dari aspek pemilihan kata dan intonasi nada bicara. Perubahan perilaku tersebut jika diamati lebih dalam terkadang memiliki pola-pola tertentu. Perubahan pola perilaku tersebut bisa menjadi salah satu tanda deteksi dini seseorang sedang tidak dalam keadaan baik-baik saja.

Pola perlu diamati untuk dapat dianalisis lebih lanjut. Setiap pola memiliki keterkaitan dengan pola lain yang apabila dihubungkan akan dapat memberikan *insight* tertentu. Salah

satu pola yang dapat diamati dari perilaku komunikasi seseorang adalah pola komunikasi dalam *chat*. Pemilihan kata yang digunakan untuk berkomunikasi dengan *chat* bisa dianalisis untuk pendeteksi dini yang mengidentifikasi seseorang sedang dalam kondisi stres atau tidak. Makalah ini dibuat dengan tujuan melakukan analisis terhadap pola komunikasi dalam *chat* untuk mendeteksi stres dini pada mahasiswa. Metode analisis yang digunakan dalam makalah ini adalah *string matching* dan *pattern recognition* menggunakan algoritma pencocokan *string* dan *regular expression*.

II. LANDASAN TEORI

A. String Matching

String Matching atau pencocokan *string* merupakan algoritma yang digunakan untuk menemukan pola tertentu pada serangkaian *string* panjang dari teks. Pencarian pola ini dapat dilakukan dengan cara *exact matching* atau *fuzzy matching*. *Exact matching* merupakan pencarian yang dilakukan dengan menemukan pola pada suatu teks. Pencarian ini dilakukan sesuai dengan pola yang dicari. *Fuzzy matching* merupakan pencarian yang menggunakan aproksimasi sehingga memiliki toleransi terhadap kesalahan salah eja dan sejenisnya. Salah satu algoritma yang digunakan untuk menghitung tingkat kemiripan adalah dengan *levenshtein distance*. Terdapat dua algoritma yang dapat digunakan untuk melakukan *pattern matching*, yaitu *KMP algorithm* dan *Boyer-Moore Algorithm*.

B. KMP Algorithm

The Knut-Morris-Pratt (KMP) algorithm merupakan algoritma pencarian pola dalam teks yang dilakukan dari kiri ke kanan. Algoritma ini memiliki konsep yang hampir sama dengan algoritma *brute force* namun hal yang membedakan adalah pergeseran pola yang lebih baik dibandingkan *brute force*. Pada algoritma ini, Ketika terdapat *mismatch* antara pola P pada P[j] di mana T[i] ≠ P[j], maka pergeseran pola yang dilakukan adalah sejauh panjang pola dikurangi panjang *prefix* yang merupakan *suffix*. Secara matematis pergeseran yang dilakukan dapat dituliskan sebagai berikut :

$$S = \text{length}(\text{pattern}) - \text{length}(\text{prefix\&suffix}) \quad (1)$$

Untuk mempermudah perhitungan pergeseran, dalam algoritma ini terdapat fungsi pinggir KMP (*KMP Border*

Function) yang digunakan untuk melakukan preprocessing untuk menemukan prefix dalam pola.

j	0	1	2	3	4	5
P[j]	a	b	a	a	b	a
k	0	1	2	3	4	
b(k)	0	0	1	1	2	

Tabel 1. KMP Border Function

Kompleksitas algoritma KMP adalah sebesar $O(m+n)$. $O(m)$ untuk menghitung fungsi pinggir dan $O(n)$ untuk pencarian string. Kompleksitas algoritma ini lebih cepat jika dibandingkan dengan brute force.

C. Boyer-Moore Algorithm

Boyer-Moore algorithm merupakan algoritma pencarian pola yang menggunakan dua teknik utama, yaitu looking-glass dan character-jump technique. Teknik looking-glass merupakan teknik yang digunakan untuk mencari pola P dalam teks T yang dilakukan secara mundur, dimulai dari akhir.

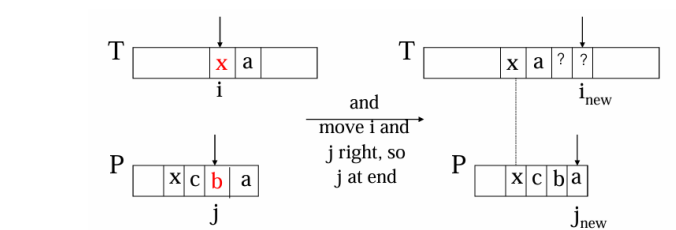
s	e	p	a	t	u
		s	a	t	u

Tabel 2. Ilustrasi Pencarian Backward

Teknik character-jump adalah teknik pergeseran pola yang dilakukan berdasarkan kondisi yang terpenuhi. Terdapat tiga kemungkinan kondisi dalam melakukan pergeseran, yaitu :

1. Kondisi 1

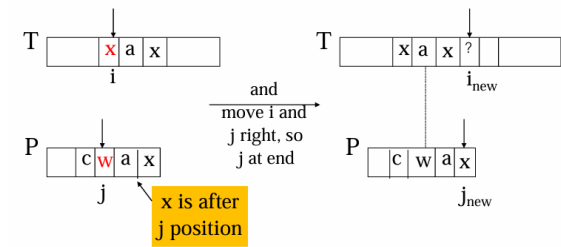
Ketika P mengandung x, maka geser P ke kanan hingga sejajar dengan kemunculan terakhir x.



Gambar 1. Boyer-Moore Case 1 (sumber : [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-\(2025\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-(2025).pdf))

2. Kondisi 2

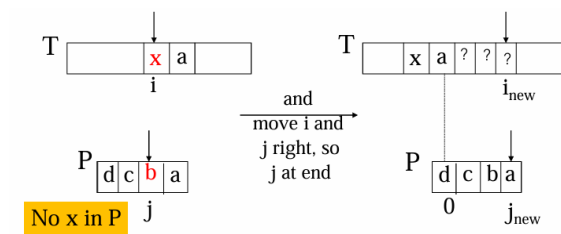
Ketika P mengandung x, namun kondisi tidak memungkinkan pergeseran seperti kondisi 1, maka geser P satu karakter ke kanan.



Gambar 2. Boyer-Moore Case 2 (sumber : [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-\(2025\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-(2025).pdf))

3. Kondisi 3

Ketika kasus 1 dan 2 tidak terpenuhi, maka geser P sehingga P[0] sejajar dengan T[i+1]



Gambar 3. Boyer-Moore Case 3 (sumber : [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-\(2025\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-(2025).pdf))

Untuk mempermudah perhitungan pergeseran, algoritma ini memiliki last occurrence function yang digunakan untuk preprocessing dan menemukan kemunculan terakhir pada pola P.

A = {a, b, m, u}

P = "bambu"

x	a	b	m	u	etc
L(x)	1	3	2	4	-1

Tabel 3. Last Occurrence Function

D. Regular Expression

Regular expression atau regex adalah notasi khusus yang digunakan untuk melakukan pencocokan pola dalam teks. Pencarian pola ini dilakukan dengan cara membuat formula ekspresi untuk mencari kata-kata dengan karakteristik tertentu. Formula dibuat menggunakan perpaduan beberapa simbol yang telah tersedia. Beberapa basic symbol untuk regex yang sering digunakan antara lain :

expression	matches...
abc	abc (that exact character sequence, but anywhere in the string)
^abc	abc at the <i>beginning</i> of the string
abc\$	abc at the <i>end</i> of the string
a b	either of a and b
^abc abc\$	the string abc at the beginning or at the end of the string
ab(2,4)c	an a followed by two, three or four b's followed by a c
ab(2,c)	an a followed by at least two b's followed by a c
ab*c	an a followed by any number (zero or more) of b's followed by a c
ab+c	an a followed by one or more b's followed by a c
ab?c	an a followed by an optional b followed by a c; that is, either abc or ac
a.c	an a followed by any single character (not newline) followed by a c
a\\.c	a.c exactly
[abc]	any one of a, b and c
[Aa]bc	either of Abc and abc
[abc]+	any (nonempty) string of a's, b's and c's (such as a, abba, acbabacaaa)
[^abc]+	any (nonempty) string which does <i>not</i> contain any of a, b and c (such as defg)
\\d\\d	any two decimal digits, such as 42; same as \\d{2}
\\w+	a "word": a nonempty sequence of alphanumeric characters and low lines (underscores), such as foo and 12bar8 and foo_1
100\\s*mk	the strings 100 and mk optionally separated by any amount of white space (spaces, tabs, newlines)
abc\\b	abc when followed by a word boundary (e.g. in abc! but not in abcd)
perl\\B	perl when <i>not</i> followed by a word boundary (e.g. in perlert but not in perl stuff)

Gambar 4. *Regular Expression* (sumber: <https://devopedia.org/images/article/173/6028.1557317770.jpg>)

III. ANALISIS DAN PEMBAHASAN

Disclaimer, penulis bukanlah ahli dalam konsep psikologi. Penulisan makalah ini ditujukan untuk mendeteksi kemunculan pola-pola kalimat negatif dalam pesan. Kemunculan pola ini akan dihitung untuk mengetahui seberapa sering kalimat negatif muncul dan dalam makalah ini penulis mencoba untuk menghubungkan korelasi antara kemunculan kalimat negatif tersebut sebagai tanda deteksi dini stres pada mahasiswa. Tentu hasil dari perhitungan ini tidak bisa dijadikan sebagai sumber diagnosis karena makalah yang dibuat tidak berfokus pada masalah tersebut, namun lebih berfokus pada penerapan konsep *pattern matching* untuk menemukan pola kalimat negatif dalam chat.

Teks adalah salah satu jenis data yang dapat diolah untuk mendapatkan informasi tertentu. Dalam suatu obrolan *chat*, seseorang akan menggunakan teks untuk mengirim pesan kepada pihak lain. Pesan yang dikirim berupa teks *string* panjang yang dapat diolah untuk mendapatkan *insight* tertentu, seperti salah satunya adalah untuk mendeteksi munculnya tanda-tanda stres dari pola komunikasi yang terdeteksi dari pesan yang dikirim.

Dalam penulisan makalah ini, penulis melakukan survei singkat kepada beberapa mahasiswa Insititut Teknologi Bandung untuk mengetahui kalimat atau kata apa yang sering mereka gunakan dalam obrolan *chat* ketika sedang merasa tertekan atau tidak baik-baik saja. Survei ini telah diisi oleh 10 orang responden. Survei ini tidak dapat mewakili keseluruhan mahasiswa dan hanya mewakili sebagian sampel saja. Survei ini ditujukan untuk mengumpulkan informasi *keyword* yang akan digunakan untuk perhitungan kemunculan pola dengan menggunakan regex. Hasil survei yang dilakukan adalah sebagai berikut :

Apakah kamu seorang mahasiswa/mahasiswi?

10 jawaban

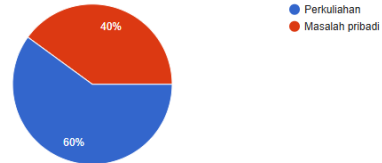
Salin diagram



Penyebab mengalami stress atau merasa tidak baik-baik saja

10 jawaban

Salin diagram



Kalimat atau kata-kata apa yang sering kamu gunakan dalam obrolan (chat) ketika sedang merasa stress atau tidak baik-baik saja. (pisahkan dengan tanda koma, contoh : malas, lelah, bosan, ingin tidur)

10 jawaban

bosan, lelah, gatau mau gimana lagi
Cape
capek, nangis, main yuk, mau ngobrol, stres
cape, hhh, tolong
lelah, capek, ingin tidur, demotivasi
cape, ketiduran, stres cuy, gila gw, huaaa
Lelah, capek, stress, jenuh, dll.
mati, meninggal, mau mati, nyerah, malas, mls, loss, nangis, mau nangis, gila, mau gila, fak, cuki, fakmen
cape, ingin tidur, tolong, skip
tidur, istirahat

Gambar 5. Hasil Survei Pola Komunikasi (sumber : [Survei Pola Komunikasi Saat Stress](#) milik penulis)

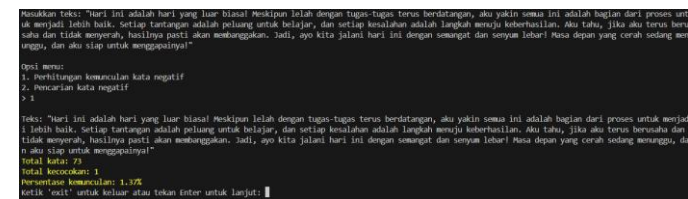
Hasil survei menunjukkan bahwa 60% penyebab stres yang dialami mahasiswa adalah masalah perkuliahan dan 40% karena masalah pribadi. Jawaban responden pada pertanyaan ketiga akan menjadi dasar dalam perhitungan kemunculan kalimat negatif pada obrolan. Kalimat atau kata hasil survei akan digunakan untuk *string matching* dengan regex. Perhitungan kemunculan pola yang sesuai akan dibandingkan dengan jumlah kata yang ada pada teks. Hasil akhirnya berupa persentase berapa *matches pattern* yang muncul dalam teks tersebut.

Teks yang digunakan untuk pengujian adalah teks fiktif yang digunakan sebagai pembandingan antara pesan yang memiliki *tone* negatif dan yang memiliki *tone* positif.

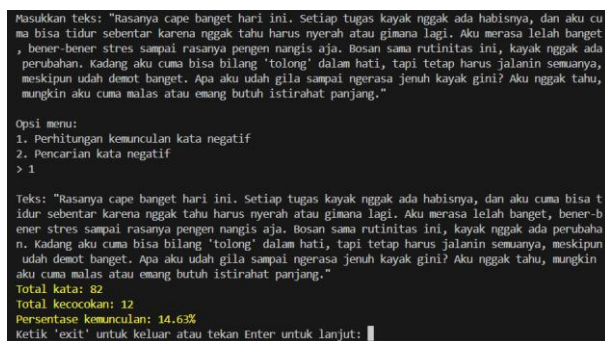
pesan positif	pesan negatif
"Hari ini adalah hari yang luar biasa! Meskipun lelah dengan tugas-tugas terus berdatangan, aku yakin	"Rasanya cape banget hari ini. Setiap tugas kayak nggak ada habisnya, dan aku cuma bisa tidur sebentar karena nggak

semua ini adalah bagian dari proses untuk menjadi lebih baik. Setiap tantangan adalah peluang untuk belajar, dan setiap kesalahan adalah langkah menuju keberhasilan. Aku tahu, jika aku terus berusaha dan tidak menyerah, hasilnya pasti akan membanggakan. Jadi, ayo kita jalani hari ini dengan semangat dan senyum lebar! Masa depan yang cerah sedang menunggu, dan aku siap untuk menggapainya!"	tahu harus nyerah atau gimana lagi. Aku merasa lelah banget, bener-bener stres sampai rasanya pengen nangis aja. Bosan sama rutinitas ini, kayak nggak ada perubahan. Kadang aku cuma bisa bilang 'tolong' dalam hati, tapi tetap harus jalanin semuanya, meskipun udah demot banget. Apa aku udah gila sampai ngerasa jenuh kayak gini? Aku nggak tahu, mungkin aku cuma malas atau emang butuh istirahat panjang."
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabel 4. Teks Uji Coba



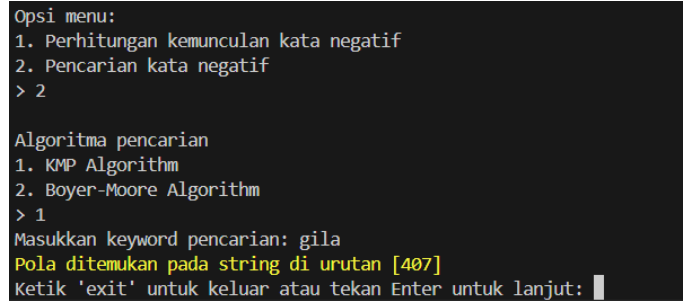
Gambar 6. Hasil Uji Coba Teks Positif



Gambar 7. Hasil Uji Coba Teks Negatif

Hasil persentase kemunculan pola menunjukkan bahwa pada teks dengan *tone* positif, hanya ditemukan 1,37% kata dari hasil survei yang menggambarkan perasaan negatif. Sedangkan pada teks dengan *tone* negatif ditemukan 14,63% kata dari hasil survei yang menggambarkan perasaan negatif. Hal ini menunjukkan bahwa kecenderungan penggunaan kata bernada negatif dilakukan ketika mengirim pesan dengan *tone* negatif juga. Pesan dengan *tone* negatif memang tidak dapat langsung disimpulkan sebagai bentuk tanda-tanda stres, akan tetapi banyaknya kemunculan kata negatif dapat mengindikasikan bahwa seseorang sedang dalam kondisi tidak baik-baik saja.

Selain melakukan perhitungan persentase kemunculan kata negatif, program yang penulis buat juga dapat digunakan untuk mencari kata tertentu dalam pesan. Fitur ini dapat digunakan untuk pencarian secara *exact match* untuk menemukan apakah dalam obrolan ini terdapat *keyword* yang dicari.



Gambar 8. Hasil Uji Coba Pencarian *Keyword*

IV. IMPLEMENTASI PROGRAM

Implementasi program pada makalah ini dibagi menjadi dua bagian secara umum. Program perhitungan persentase kemunculan menggunakan regex dan program pencarian kata menggunakan algoritma *string matching*. Untuk perhitungan persentase, pengguna cukup menginputkan teks pesan yang ingin dicek dan *output* persentasenya akan dihitung berdasarkan *keyword* hasil survei yang telah dilakukan. Perhitungan dihitung dengan menggunakan formula :

$$\text{persentase} = \text{matches word} / \text{length word}$$



Gambar 9. Program *Regex Search*


```
# pola berdasarkan hasil survei
my_patterns = {
    'cape_pattern': r'\bcape\.\?\\w*\b',
    'lelah_pattern': r'\blelah\.\?\\w*\b',
    'tidur_pattern': r'\btidur\.\?\\w*\b',
    'nangis_pattern': r'\bnangis\.\?\\w*\b',
    'bosan_pattern': r'\bbosan\.\?\\w*\b',
    'gatau_pattern': r'\bgatau\.\?\\w*\b',
    'stres_pattern': r'\bstres\.\?\\w*\b',
    'tolong_pattern': r'\btolong\.\?\\w*\b',
    'demot_pattern': r'\bdemot\.\?\\w*\b',
    'gila_pattern': r'\bgila\.\?\\w*\b',
    'jenuh_pattern': r'\bjenuh\.\?\\w*\b',
    'mati_pattern': r'\bmati\.\?\\w*\b',
    'meninggal_pattern': r'\bmeninggal\.\?\\w*\b',
    'nyerah_pattern': r'\bnyerah\.\?\\w*\b',
    'malas_pattern': r'\bmalas\.\?\\w*\b',
    'tolong_pattern': r'\btolong\.\?\\w*\b',
    'kasar_pattern': r'\b(fak|cuki)\.\?\\w*\b'
}
```

Gambar 10. *Pattern* Hasil Survei

Implementasi program *string match* menggunakan algoritma kmp dan Boyer-Moore. Untuk algoritma KMP dimulai dengan mencari `compute_lps()`. Fungsi ini bertujuan untuk mencari fungsi *longest prefix suffix*. Setelah itu teks akan dicek secara iteratif dari kiri ke kanan dan ketika ditemukan *string match* maka akan di return indeks yang sesuai.

```
def kmp_search(text, pattern):

    def compute_lps(pattern):
        """Menghitung Longest Proper Prefix yang juga Suffix"""
        m = len(pattern)
        lps = [0] * m
        length = 0
        i = 1

        while i < m:
            if pattern[i] == pattern[length]:
                length += 1
                lps[i] = length
                i += 1
            else:
                if length != 0:
                    length = lps[length - 1]
                else:
                    lps[i] = 0
                    i += 1

        return lps

    n = len(text)
    m = len(pattern)

    if m == 0:
        return []

    lps = compute_lps(pattern)

    matches = []
    i = 0 # index untuk text
    j = 0 # index untuk pattern

    while i < n:
        if pattern[j] == text[i]:
            i += 1
            j += 1

        if j == m:
            matches.append(i - j)
            j = lps[j - 1]
        elif i < n and pattern[j] != text[i]:
            if j != 0:
                j = lps[j - 1]
            else:
                i += 1

    return matches
```

Gambar 11. Program Algoritma KMP

Implementasi algoritma Boyer-Moore dimulai dengan mencari fungsi `bad_char_table()`. Fungsi ini bertujuan untuk mencari posisi terakhir kemunculan setiap karakter. Setelah itu teks akan dicek secara iteratif dari kanan ke kiri dan ketika ditemukan *string match* maka akan di return indeks yang sesuai.

```
def boyer_moore_search(text, pattern):

    def bad_char_table(pattern):
        """Membuat tabel bad character"""
        table = {}
        m = len(pattern)

        # Inisialisasi semua karakter dengan -1
        for i in range(m):
            table[pattern[i]] = -1

        # Isi tabel dengan posisi terakhir setiap karakter
        for i in range(m):
            table[pattern[i]] = i

        return table

    n = len(text)
    m = len(pattern)

    if m == 0:
        return []

    bad_char = bad_char_table(pattern)

    matches = []
    s = 0

    while s <= n - m:
        j = m - 1

        # Cocokkan pattern dari kanan ke kiri
        while j >= 0 and pattern[j] == text[s + j]:
            j -= 1

        if j < 0:
            matches.append(s)
            s += (m - bad_char.get(text[s + m], -1) - 1) if s + m < n else 1
        else:
            char = text[s + j]
            shift = max(1, j - bad_char.get(char, -1))
            s += shift

    return matches
```

Gambar 12. Program Algoritma Boyer-Moore

V. KESIMPULAN

Pola komunikasi yang dilakukan orang memiliki kaitan dengan keadaan dan kondisi mereka. Orang yang berada dalam kondisi kurang baik secara emosional memiliki kecenderungan untuk memiliki pola komunikasi yang bernada negatif. Pemilihan kata yang memiliki nada negatif memberikan indikasi bahwa seseorang sedang tidak baik-baik saja. Hasil survei yang dilakukan untuk mendapatkan informasi kata atau kalimat yang sering digunakan saat merasa stres dijadikan sebagai bahan pembandingan dan perhitungan persentase kemunculan kata negatif. Hasil tes perbandingan teks positif dan negatif pada bab III menunjukkan bahwa teks dengan nada negatif memiliki persentase kemunculan kata negatif yang lebih besar dibandingkan dengan teks bernada positif. Hal ini menunjukkan bahwa pola komunikasi seseorang dapat menggambarkan keadaan mereka. Akan tetapi, nilai persentase ini tidak dapat dijadikan patokan pasti terkait *range* berapa yang masuk kategori stres atau tidak. Nilai persentase ini hanya menggambarkan seberapa sering seseorang menggunakan pola komunikasi dengan nada negatif. Nilai tersebut dapat

digunakan sebagai bahan evaluasi dan indikasi awal kondisi seseorang apakah sedang tidak baik-baik saja.

VIDEO LINK AT YOUTUBE

<https://youtu.be/wpO9YOhbKss?feature=shared>

LAMPIRAN LINK GITHUB

<https://github.com/DiyahSusan/MakalahStima2025>

ACKNOWLEDGMENT

Pertama-tama penulis ingin mengucapkan rasa syukur kepada Tuhan Yang Maha Esa sehingga penulis dapat menyelesaikan makalah ini dengan tepat waktu dan sesuai dengan yang diharapkan. Penulis juga ingin mengucapkan terima kasih kepada dosen matakuliah Strategi Algoritma IF2211 karena telah memberikan banyak ilmu baru dan membuat penulis mengeksplor banyak hal baru di mata kuliah ini. Penulis ingin memberikan ucapan terima kasih untuk orang-orang yang karya tulisannya telah menginspirasi penulis dalam menyelesaikan makalah ini. Penulis juga ingin memberikan ucapan terima kasih kepada seluruh pembaca yang telah meluangkan waktu untuk membaca makalah ini. Penulis berharap makalah ini dapat bermanfaat bagi pembaca secara khusus dan masyarakat secara umum.

REFERENCES

- [1] Hakiki, M. I., & Mariyati, L. I. (2017/2018). Gambaran stres akademik mahasiswa prodi psikologi FPIP yang memprogram mata kuliah skripsi. Universitas Muhammadiyah Sidoarjo. Retrieved from <https://doi.org/10.21070/acopen.3.2020.955>
- [2] Lin, S. H., & Huang, Y. C. (2014). Life stress and academic burnout. *Active Learning in Higher Education*, 15(1), 77-90.
- [3] Lumban Gaol, N. T. (2016). Teori stres: Stimulus, respons, dan transaksional. *Buletin Psikologi*, 24(1), 1-11. Retrieved from <https://doi.org/10.22146/bpsi.11224>
- [4] Munir, Rinaldi. (2025). Pencocokan string dengan algoritma brute force, KMP, Boyer-Moore. Retrieved from [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-\(2025\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-(2025).pdf)
- [5] Munir, Rinaldi. (2025). Pencocokan string dengan *Regular Expression*. Retrieved from [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/24-String-Matching-dengan-Regex-\(2025\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/24-String-Matching-dengan-Regex-(2025).pdf)
- [6] Rezki, A. N., Naryoso, A., & Pradekso, T. (n.d.). Hubungan intensitas penggunaan media sosial dengan tingkat stres. Program Studi S1 Ilmu Komunikasi, Fakultas Ilmu Sosial dan Ilmu Politik, Universitas Diponegoro. Retrieved from <https://ejournal3.undip.ac.id/index.php/interaksi-online/article/download/43531/30858>

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, 24 Juni 2025



Diyah Susan Nugrahani 13523080