Penggunaan String Matching dengan Algoritma KMP dalam Mengidentifikasi Kemungkinan Kerusakan pada Hardware Komputer

Mohamad Hilmi Rinaldi - 13520149
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
Email: aldi280402@gmail.com

***Abstrak*—Komputer merupakan perangkat teknologi informasi yang sedang banyak digunakan oleh berbagai kalangan umur di masa pandemi COVID-19. Perangkat ini dapat digunakan untuk berbagai kegiatan mulai dari menulis dokumen, mengedit video, menjelajah web, dan kegiatan lainnya. Alasan masih banyak orang untuk memilih perangkat komputer dibandingkan lainnya yaitu terdapat pada komponen *hardware*-nya yang fleksibel untuk diganti ketika terjadi kerusakan atau sekedar ingin meningkatkan performanya. Namun, ketika terjadi kerusakan pada perangkat komputer terdapat kebingungan bagian komponen mana yang memiliki kerusakan. Di dalam makalah ini, akan dibahas salah satu program yang dapat digunakan untuk mengidentifikasi kemungkinan kerusakan pada *hardware* komputer menggunakan *string matching* dengan algoritma KMP sehingga dapat mengurangi kebingungan pada pengguna dalam menentukan komponen *hardware* yang perlu diganti ketika terjadi kendala pada komputer.**

Kata kunci—Komputer; Hardware; String Matching; Algoritma KMP

I. PENDAHULUAN

Pada masa pandemi COVID-19, seluruh kegiatan dan aktivitas manusia sangat dibatasi. Pembatasan ini membuat perubahan yang sangat drastis di berbagai bidang mulai dari pendidikan, ekonomi, hingga perkantoran. Kegiatan yang awal mulanya mengharuskan tatap muka mulai berpindah menjadi daring. Akibat dari perubahan ini, kebutuhan perangkat teknologi informasi untuk menunjang kegiatan daring semakin meningkat. Peningkatan ini terjadi karena perangkat tersebut dibutuhkan oleh berbagai kalangan mulai dari anak-anak yang masih sekolah hingga kalangan dewasa.

Perangkat yang digunakan dapat berbagai macam mulai dari komputer, laptop, *smartphone*, hingga tablet. Salah satu perangkat yang banyak digunakan yaitu komputer. Komputer dapat digunakan untuk berbagai macam kebutuhan mulai dari menjelajah web, menulis dokumen, mengedit video, bermain *game*, dan masih banyak lagi aktivitas yang dapat dilakukan dengan perangkat ini.

Terdapat beberapa alasan orang-orang masih memilih untuk menggunakan komputer dibandingkan perangkat yang lain. Diantaranya yaitu jika dibandingkan dengan laptop yang dapat mengerjakan aktivitas yang hampir sama, performa komputer lebih baik dibandingkan dengan laptop untuk harga yang sama karena pada laptop komponen yang digunakan dibuat versi kecilnya sehingga performanya kurang maksimal. Selain itu, komputer lebih mudah untuk di-*upgrade* karena hanya perlu mengganti beberapa bagian komponen pada *hardware*-nya seperti mengganti *processor* atau kartu grafisnya saja.

Fleksibilitas dalam komponen *hardware* pun dapat menguntungkan ketika terdapat kerusakan pada komputer. Untuk mengatasi kerusakannya, bisa dengan mengganti beberapa bagian komponen *hardware*-nya saja. Namun, terkadang terdapat kebingungan untuk menentukan bagian komponen yang bermasalah. Hal ini terlihat pada forum diskusi di suatu *social media* yang menanyakan komponen *hardware* mana yang harus diganti ketika terdapat suatu kerusakan berdasarkan suatu ciri-ciri yang dialami seperti pada gambar di bawah.

Gambar 1 Tangkapan Layar di *Social Media*
Sumber : Arsip Penulis

Di dalam forum tersebut, terdapat diskusi untuk menjelaskan kemungkinan komponen *hardware* yang mengalami kerusakan berdasarkan kendala yang dialami. Oleh karena itu, dibuatlah program sederhana untuk membantu mengidentifikasi bagian komponen *hardware* yang rusak berdasarkan masukan ciri-ciri yang dialami dengan *string matching* melalui data yang sudah dikumpulkan.

II. LANDASAN TEORI

A. Komputer

Komputer merupakan sebuah perangkat atau mesin yang dapat melakukan proses, perhitungan, dan operasi berdasarkan instruksi yang diberikan oleh program. Perangkat ini dapat menerima *input* lalu memprosesnya dan akan menghasilkan suatu *output*.

Berdasarkan komponennya, komputer memiliki suatu perangkat keras yang disebut dengan *hardware*. *Hardware* merupakan suatu bagian komputer yang dapat dilihat fisiknya. Komponen ini terbagi lagi menjadi beberapa bagian yaitu :

1. *Motherboard*

Gambar 2 Komponen *Motherboard*
Sumber : https://www.crucial.com/articles/pc-builders/what-is-computer-hardware

Motherboard berfungsi sebagai tempat untuk menyimpan dan menyambungkan berbagai komponen di komputer. Di dalam *motherboard* berisi soket *processor*, soket RAM, soket *VGA card*, dan soket tambahan lainnya.

2. *Processor* (CPU)

Gambar 3 Komponen *Processor*
Sumber : https://www.crucial.com/articles/pc-builders/what-is-computer-hardware

*Processor (*CPU) berfungsi sebagai otak sentral dalam komputer. CPU dapat mengerjakan berbagai perintah yang sudah terprogram dan tersimpan dalam *harddisk*.

3. *Hard disk* (HDD)

Gambar 4 Komponen *Hard disk*
Sumber : https://www.easytechjunkie.com/what-is-a-hard-disk.htm

Hard disk (HDD) berfungsi sebagai media penyimpanan data. HDD ini bersifat *non volatile* sehingga data sudah tersimpan tidak akan hilang jika listrik dimatikan.

4. *Random Access Memory* (RAM)

Gambar 5 Komponen *Random Access Memory*
Sumber : https://www.crucial.com/articles/pc-builders/what-is-computer-hardware

Random Access Memory (RAM) berfungsi sebagai media penyimpanan data sementera ketika komputer sedang digunakan. RAM ini bersifat *volatile* sehingga data akan hilang jika listrik dimatikan.

5. VGA *Card* (Kartu Grafis)

Gambar 6 Komponen VGA *Card*
Sumber : https://www.crucial.com/articles/pc-builders/what-is-computer-hardware

VGA merupakan singkatan dari *Video Graphics Array*. VGA *card* berfungsi untuk mengeluarkan *output* grafis yang akan ditampilkan pada monitor.

6. *Power Supply*

Gambar 7 Komponen *Power Supply*
Sumber : https://www.crucial.com/articles/pc-builders/what-is-computer-hardware

Power supply berfungsi untuk menyuplai daya berupa

arus listrik untuk menyalakan komputer dan perangkat lainnya. Arus listrik yang diterima oleh *power supply* akan diubah menjadi arus listrik searah (DC) yang awalnya berasal dari daya arus listrik berlawanan arah (AC).

B. String Matching

String matching merupakan pencarian sebuah *pattern* di dalam sebuah teks. *Pattern* dan teks memiliki tipe data string dengan panjang dari *pattern* akan lebih kecil dari panjang teks. Suatu string terdiri dari dua bagian, yaitu *prefix* dan *suffix*. *Prefix* merupakan suatu huruf atau beberapa huruf yang berada di awal string. Sedangkan *suffix* berupa huruf yang berada di akhir string. Misalkan terdapat string S yang berukuran m :

$$S = x_0 x_1 \ldots x_{m-1}$$

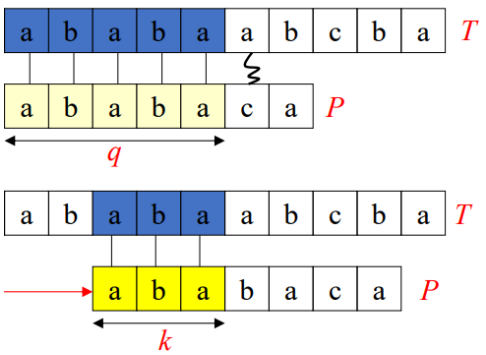
Prefix dari string S yaitu substring dari S[0..k] dan *suffix* dari string S yaitu substring dari S[k..m-1], k merupakan indeks yang berada di antara 0 hingga m-1.

Dalam *string matching*, terdapat beberapa algoritma yang dapat digunakan. Diantaranya yaitu seperti algoritma Knuth-Morris-Pratt (KMP) dan algoritma Boyer-Moore (BM).

C. Algoritma Knuth-Morris-Pratt (KMP)

Algoritma KMP merupakan algoritma dalam *string matching* yang melakukan pencarian dengan urutan dari kiri ke kanan (*left-to-right order*). Konsep algoritma ini mirip dengan algoritma *string matching* dengan brute force karena kedua algoritma melakukan teknik pencocokan karakter yang sama. Perbedaannya yaitu pada algoritma ini jauh lebih cerdik dalam mengakali karakter mana saja yang tidak perlu diperiksa lagi sehingga pergeseran yang dilakukan bisa lebih sedikit.

Misalkan terdapat dua buah string, yaitu *pattern* yang ingin dicari (ABABAABCBA) dan string yang diharapkan mengandung pattern (ABABACA). Langkah pencocokan diilustrasikan seperti ini:



Gambar 8 Algoritma KMP
Sumber : https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf

Berdasarkan ilustrasi di atas, dapat dilihat bahwa ketika karakter di T dan P tidak sama, maka dilakukan pergeseran *pattern* P ke kanan sebanyak 2 karakter. Hal ini dilakukan karena algoritma KMP menggeser *pattern* sejauh *prefix* terbesar dari P[0..j-1] yang merupakan *suffix* dari P[1..j-1].

Untuk mengetahui jumlah pergeseran ketika terjadi *missmatch*, Algoritma KMP menggunakan informasi yang dihasilkan oleh *border function* seperti pada gambar di bawah :

(k = j-1)

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

Gambar 9 *Border Function*
Sumber : https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf

j = Indeks ketika terjadi *missmatch*

k = j - 1

b(k) = Ukuran terbesar *prefix* P[0..k] dan *suffix* P[1..k]

Nilai b(k) dijadikan dasar algoritma KMP untuk mengetahui sejauh mana pergeseran harus dilakukan melalui pengurangan panjang *pattern* yang sudah valid dengan b(k).

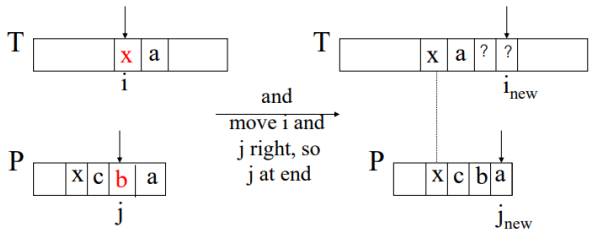
D. Algoritma Boyer-Moore (BM)

Algoritma Boyer-Moore merupakan algoritma dalam *string matching* yang melakukan pencocokan dengan 2 teknik, yaitu *looking glass technique* yang melakukan pencocokan dari bagian kanan pattern ke kiri dan *character-jump technique*.

Character-jump technique dibagi menjadi tiga bagian yaitu :

1. Kasus 1

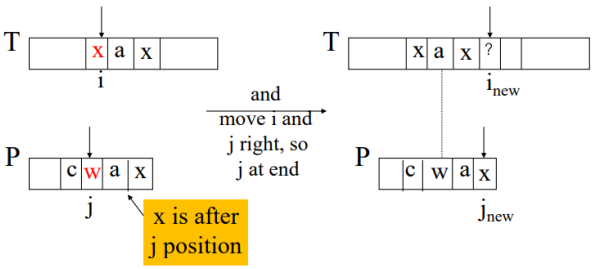
   Pattern P mengandung x, maka dilakukan pergeseran sampai x pada P bertemu dengan x pada T.



Gambar 10 Kasus 1 pada Algoritma BM
Sumber : https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf

2. Kasus 2

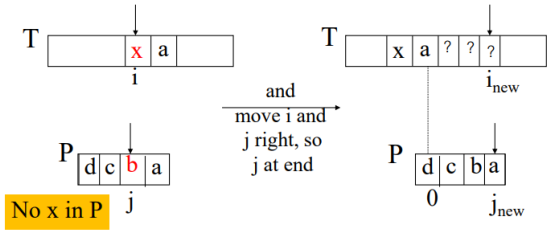
   Pattern P mengandung x, tetapi tidak mungkin melakukan shifting ke kiri (karena x tidak ada di kiri indeks j), maka geser P ke kanan sebanyak 1 karakter.



Gambar 11 Kasus 2 pada Algoritma BM
Sumber : https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf

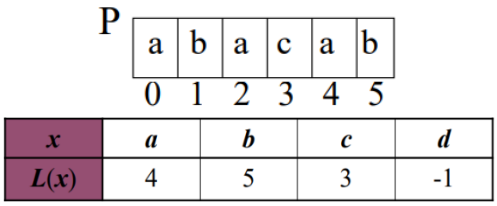
3. Kasus 3

   Tidak memenuhi kasus 1 dan 2 sehingga langsung geser P[0] ke T[i+1].



Gambar 12 Kasus 3 pada Algoritma BM
Sumber : https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf

Untuk mengetahui kemunculan terakhir dari tiap karakter teks di dalam patter, algoritma Boyer-Moore menggunakan informasi yang dihasilkan oleh *last occurrence function* seperti pada gambar di bawah :



Gambar 13 *Last Occurrence Function*
Sumber : https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf

III. PENERAPAN STRING MATCHING DENGAN ALGORITMA KMP DALAM MENGIDENTIFIKASI KEMUNGKINAN KERUSAKAN PADA HARDWARE KOMPUTER

A. Pengumpulan Data

Dalam mengidentifikasi kerusakan pada *hardware* komputer, diperlukan data uji sebagai teks yang nantinya akan dicocokkan dengan *input* dari pengguna sebagai *pattern*-nya. Pengumpulan data uji dilakukan dengan informasi yang berada di internet. Dari kendala-kendala komponen *hardware* komputer yang ada, diambil 6 komponen yang akan diambil menjadi sebuah *database* yang akan diproses dengan algoritma KMP. Berikut merupakan kendala-kendala dari berbagai komponen *hardware* komputer.

1. *Motherboard*
   a. Tidak bisa masuk BIOS
   b. Tidak ada respon di monitor ketika dinyalakan
   c. Processor mati total
   d. Sering terjadi blue screen
2. *Processor*
   a. Terjadi blue screen dengan status error (Process...)
   b. Terjadi hang ketika membuka aplikasi berat
   c. Kinerja komputer melambat
3. *RAM*
   a. Terjadi blue screen dengan status error (Memory Management)
   b. Tidak ada respon di monitor ketika dinyalakan
   c. Tidak bisa masuk BIOS
4. *VGA Card*
   a. Gambar yang dihasilkan monitor tidak sesuai
   b. Crash ketika bermain game
   c. Crash ketika render video
   d. Tidak dapat booting
5. *Hard disk*
   a. Tidak bisa booting
   b. Membuka aplikasi lambat
   c. Booting windows lambat
6. *Power supply*
   a. Komputer mati mendadak
   b. Tidak dapat menyala
   c. Kipas menjadi lebih lambat

B. Pemrosesan Data

Berdasarkan data kendala yang sudah dikumpulkan, kemudian data tersebut diproses ke dalam bentuk array dua dimensi yang akan dijadikan sebagai *database*. Kolom pertama untuk setiap barisnya diisi dengan elemen *string* dari nama komponen *hadrware*-nya dan kolom kedua diisi dengan elemen *string* dari kumpulan kendalanya.

C. Penggunaan Algoritma KMP dalam Program

Kendala-kendala yang sudah diproses sebelumnya ke dalam array dua dimensi akan digunakan dalam program utama. Alur jalannya program yaitu diawali menerima *input* dari pengguna untuk kendala yang dialami. Lalu, *input* tersebut akan dicocokkan dengan teks dari *database* yang sudah diproses menjadi array dua dimensi dengan algoritma KMP. Jika terdapat kecocokkan dengan kendala yang terdapat pada *database*, maka program akan mengeluarkan *output* berupa kemungkinan komponen yang mengalami kerusakan dan ciri-ciri kerusakan yang terdapat pada komponennya. Berikut merupakan algoritma KMP dalam bahasa python:

```
def kmp(strinput, strpattern):
    n = len(strinput)
    m = len(strpattern)
    i = 0
```

```
    j = 0
    fail = failure(strpattern)
    while(i < n):
       if strinput[i] == strpattern[j] :
            if j == m - 1:
                return i - m + 1
            i += 1
            j += 1
       else:
            if j > 0:
                j = fail[j - 1]
            else:
                i += 1
    return -1

def failure(strpattern):
    fail = [0] * len(strpattern)
    i = 1
    j = 0
    while(i < len(strpattern)):
      if strpattern[i] == strpattern[j]:
            fail[i] = j + 1
            i += 1
            j += 1
      else:
            if j > 0:
                j = fail[j - 1]
            else:
                fail[i] = 0
                i += 1
    return fail
```

IV. HASIL PENGUJIAN DAN PEMBAHASAN

Berikut merupakan hasil pengujian program dari beberapa kasus uji:

1. Masukan pengguna tidak valid

```
Kendala : as

Input kendala tidak valid
```

Gambar 14 Komponen *Power Supply*
Sumber : Arsip Penulis

Ketika masukan tidak membentuk suatu kata (kurang dari 3 karakter), maka program memberikan *output* pesan kesalahan "Input kendala tidak valid".

2. Masukan pengguna valid dan terdiri dari satu kendala

```
Kendala : tidak dapat menyala

Kemungkinan komponen yang mengalami kerusakan:
- Power Supply
Kendala : Komputer mati mendadak, tidak dapat menyala,
 kipas menjadi lebih lambat
```

Gambar 15 Komponen *Power Supply*
Sumber : Arsip Penulis

Ketika masukan valid dan kendala yang dimasukkan terdapat pada *database* maka program akan mengeluarkan *output* berupa kemungkinan komponen yang mengalami kerusakan dan ciri-ciri kerusakan yang terdapat pada komponennya.

3. Masukan pengguna valid dan terdiri dari dua kendala

```
Kendala : tidak bisa masuk bios, terjadi blue screen

Kemungkinan komponen yang mengalami kerusakan:
- Motherboard
Kendala : Tidak bisa masuk BIOS, tidak ada respon di mon
itor ketika dinyalakan, processor mati total, sering ter
jadi blue screen

- RAM
Kendala : Terjadi blue screen dengan status error (Memor
y Management), tidak ada respon di monitor ketika dinyal
akan, tidak bisa masuk BIOS
```

Gambar 16 Komponen *Power Supply*
Sumber : Arsip Penulis

Ketika masukan valid dan kendala yang dimasukkan terdapat pada *database* maka program akan mengeluarkan *output* berupa kemungkinan komponen yang mengalami kerusakan dan ciri-ciri kerusakan yang terdapat pada komponennya.

4. Masukan pengguna valid dan tidak terdapat di *database*

```
Kendala : tidak dapat membuka aplikasi

Tidak ada kerusakan sesuai kendala yang dimasukkan
```

Gambar 17 Komponen *Power Supply*
Sumber : Arsip Penulis

Ketika masukan valid dan kendala yang dimasukkan tidak terdapat pada *database* maka program akan mengeluarkan *output* pesan kesalahan berupa "Tidak ada kerusakan sesuai kendala yang dimasukkan"

V. KESIMPULAN

Aplikasi *string matching* dapat menyelesaikan berbagai persoalan di kehidupan nyata. Salah satunya yaitu mengidentifikasi kemungkinan kerusakan *hardware* komputer berdasarkan kendala yang dialami. Dengan algoritma KMP, program berhasil mencari kemungkinan kerusakan komponen tersebut berdasarkan kendala yang dimasukan oleh pengguna dari *database* yang telah dibuat.

Dengan mengetahui kemungkinan kerusakan *hardware* komputer tersebut, diharapkan pengguna komputer atau teknisi dapat mengurangi kebingungan dan meminimalisir waktu yang dibutuhkan dalam menentukan perangkat *hardware* yang perlu diganti ketika terjadi kendala pada komputer.

PRANALA VIDEO

Pranala : https://youtu.be/NegS9w8vEpA

UCAPAN TERIMA KASIH

Pertama-tama penulis ingin menyampaikan terima kasih kepada Allah SWT. yang telah menganugerahkan rahmat, hidayah, dan karunia-Nya sehingga penyusunan makalah ini dapat selesai pada waktunya. Ucapan terima kasih juga penulis sampaikan kepada orang tua penulis yang telah mendukung dari segala bidang baik itu doa hingga dukungan moral. Selain itu, penulis juga ingin menyampaikan terima kasih kepada para dosen mata kuliah IF2211 Strategi Algoritma terutama kepada Ibu Ulfa selaku pengajar K02 atas ilmu yang telah disampaikan selama satu semester ini.

DAFTAR PUSTAKA

[1] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf, diakses tanggal 22 Mei 2022
[2] http://saifulrahman.lecture.ub.ac.id/files/2012/02/Hardware-Komputer-dan-Fungsinya.pdf diakses tanggal 22 Mei 2022
[3] https://www.ashadin.com/2018/08/ciri-ciri-komponen-cpu-komputerlaptop.html, diakses tanggal 22 Mei 2022

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, 22 Mei 2022

Mohamad Hilmi Rinaldi, 13520149