Program Pendeteksian Bahasa menggunakan Algoritma Pencocokan String

Aisyah Nurul Izzah Adma
Program Studi Teknik Informatika
Sekolah Teknik Informatika dan Elektro
Institut Teknologi Bandung, Jl.Ganesha No.10 Bandung 40132
13517046@std.ste.itb.ac.id

***Abstract*—Makalah ini dituliskan dengn tujuan untuk melakukan analisa terhadap aplikasi pendeteksi bahasa. Analisa yang dilakukan adalah melakukan perbandingan antara algoritma knutt-morris-pratt, algoritma boyer-moore, dan regular expression dalam melakukan pencocokan string untuk mendeteksi bahasa yang dituliskan. Pada akhir makalah, didapatkan kesimpulan bahwa pencocokan string pada pendeteksian bahasa lebih tepat menggunakan algoritma knutt-morris-pratt.**

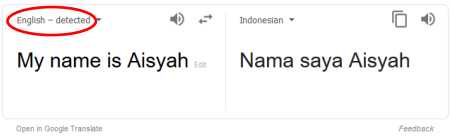
Keywords—Pencocokan string; Pendeteksian; Bahasa

I. PENDAHULUAN

Bahasa menjadi sebuah kebutuhan makhluk hidup untuk dapat berinteraksi dengan sesamanya. Bahasa tercipta secara alamiah untuk menjelaskan sesuatu. Keberagaman pola interaksi di setiap kelompok sosial dapat menyebabkan bahasa yang berbeda dengan kelompok sosial lainnya. Sama halnya dengan bahasa di seluruh dunia, setiap negara bahkan daerah di negara tersebut memiliki bahasa yang berbeda-beda. Menurut referensi, sebuah penelitian terbaru menyimpulkan bahwa terdapat sebanyak 7.000 bahasa di seluruh dunia dan digunakan hampir tujuh miliar orang.

Keberagaman bahasa ini tidak menjadi masalah ketika masyarakat antar negara tidak perlu melakukan interaksi. Namun, dimulai dengan era globalisasi, memungkinkan adanya kebutuhan interaksi masyarakat di negara yang berbeda. Selain itu, memungkinkan pula kita mendapat informasi yang dituliskan menggunakan bahasa yang tidak kita ketahui. Informasi tersebut menjadi tidak ada manfaatnya untuk kita jika kita tidak memahami bahasa tersebut.

Kini, perkembangan teknologi melahirkan berbagai aplikasi yang mampu mendeteksi suatu bahasa. Contohnya *google translate*. Saat kita menuliskan bahasa yang ingin kita terjemahkan, sebelumnya, *google translate* mampu mendeteksi jenis bahasa yang kita tuliskan.



Gambar 1. Google translate mendeteksi “My name is Aisyah” adalah Bahasa Inggris

Sebagai mahasiswi Teknik Informatika, saya merasa tertarik untuk melakukan penelitian tentang algoritma yang digunakan dalam melakukan pendeteksian bahasa tersebut. Dalam makalah ini, akan dilakukan analisis mengenai penggunaan dan perbandingan berbagai algoritma pencocokan string untuk mendeteksi suatu bahasa yang berbentuk tulisan.

II. DASAR TEORI

A. *Bahasa*

Dikutip dari Kridalaksana (1923), bahasa adalah sistem lambang bunyi yang arbitrer yang digunakan oleh anggota kelompok sosial untuk bekerja sama, berkomunikasi dan mengidentifikasikan diri.

Menurut Anderson dan Douglas Brown bahwa bahasa memiliki ciri atau sifat bahasa. Ciri-ciri bahasa itu antara lain bahasa itu adalah sebuah sistem, berwujud lambang, berupa bunyi, bersifat arbitrer, bermakna, bersifat konvensional, unik, universal, dan produktif, bervariasi, dinamis,digunakan sebagai alat komunikasi, dan merupakan identitas penuturnya.

Bahasa memiliki berbagai jenis bentuk. Contohnya bahasa lisan, bahasa tubuh, bahasa batin, bahasa isyarat ataupun bahasa pemrograman. Secara konsep umum, bahasa bisa mengacu pada kemampuan kognitif untuk belajar, dan menggunakan sistem komunikasi yang kompleks, atau untuk menjelaskan sekumpulan aturan yang membentuk sistem tersebut, atau sekumpulan pengucapan yang dapat dihasilkan dari aturan-aturan tersebut. Hal ini mengakibatkan setiap orang dapat memiliki interpretasi bahasa yang berbeda.

Di seluruh dunia, terdapat 6000-7000 jenis bahasa. Bahasa itu sendiri ada yang dituliskan kedalam alfabet ataupun menggunakan huruf-huruf yang tidak terdaftar dalam ASCII. Contoh bahasa yang dituliskan dalam alfabet adalah Bahasa Indonesia dan Bahasa Inggris. Sedangkan bahasa yang dapat dituliskan dalam huruf bukan alfabet adalah Bahasa Korea dan Bahasa Jepang.

B. *String*

String adalah urutan simbol atau nilai yang berdekatan seperti string karakter (urutan karakter) atau string biner (urutan biner).

Asumsikan sebuah string memiliki panjang m, string didefinisikan sebagai berikut :

$$S = x_o x_1 x_2 \dots x_{m-1}$$

Prefix dari string tersebut ialah sebuah bagian dari string (substring) yaitu $S = [0..k]$. Suffix dari string tersebut ialah sebuah bagian dari string (substring) yaitu $S = [k..m-1]$. Dengan k adalah sebuah index apapun bernilai dari 0 sampai 1.

Berikut ini contoh untuk dapat lebih memahami konsep string :

- S :

A	I	S	Y	A	H

- Semua kemungkinan prefix :
  "a", "ai", "ais", "aisy "aisya", "aisyah"
- Semua kemungkinan suffix :
  "h", "ah", "yah", "syah", "isyah", "aisyah"

C. Pencocokan String

Pencocokan string adalah suatu permasalahan untuk menemukan kemunculan string yang disebut pola dalam sebuah string lain yang disebut teks. Secara umum, pencocokan string dilakukan sehingga menemukan kecocokan yang tepat sesuai pola atau disebut dengan istilah *exact matching.*

Dalam algoritma pencocokan string, digunakan simbol Tdan P. Simbol tersebut didefinisikan sebagai berikut.

- T : *Text*, yaitu string yang panjangnya n karakter
- P : *Pattern*, yaitu string yang panjangnya m karakter yang akan dicari di dalam *text*. Diasumsikan bahwa $m << n$.

D. Algoritma Pencocokan String

Terdapat beberapa macam algoritma yang digunakan dalam melakukan pencocokan string. Berikut ini algoritma pencocokan string termasuk penjelasannya.

a) Algoritma Brute-Force

Melakukan pemeriksaan pada setiap posisi di dalam teks T untuk melihat kemungkinan pola P dimulai dari posisi tersebut.

Diberikan contoh sebagai berikut :

- T :

A	I	S	Y	A	H

- P :

Y	A	H

Pada saat pemeriksaan, P bergerak satu karakter pada T.

A	I	S	Y	A	H

↓

Y	A	H

Ketika menemukan ketidakcocokan pada indeks tersebut pada T, P kembali bergeser satu karakter pada T. Pemeriksaan terus dilakukan dengan menggerakan P setiap satu karakter pada T hingga solusi ditemukan, yaitu terdapat pola P pada teks T.

A	I	S	Y	A	H

↓

Y	A	H

Pada algoritma brute-force, kasus terburuk menyebabkan jumlah perbandingan yaitu sebanyak $m * (n - m + 1) = O(mn)$. Pada kasus terbaik yang terjadi ketika karakter pertama pada pola P tidak pernah sama dengan karakter teks T yang dicocokan, kompleksitas yang terjadi dalam notasi big-O ialah $O(n)$. Sedangkan, pada kasus rata-rata, kompleksitas yang terjadi dalam notasi big-O ialah $O(m + n)$.

b) Algoritma Knuth-Morris-Pratt

Algoritma Knuth-Morris-Pratt atau KMP merupakan algoritma pencocokan string yang mencari sub-string dengan urutan dari kiri ke kanan. Algoritma ini memiliki kemiripan dengan algoritma brute-force, tetapi dalam melakukan pergeseran pola P dilakukan secara lebih efisien. Pada algoritma ini, disimpan informasi yang digunakan untuk melakukan jumlah pergeseran sehingga pola P dapa bergeser sebesar lebih dari satu karakter pada teks T.

Pada algoritma KMP, dilakukan proses awal terhadap pola P untuk menemukan kecocokan prefix pada pola P itu sendiri. Proses tersebut merupakan proses kalkulasi fungsi pinggiran (*The border function)*. Fungsi pinggiran mengindikasikan pergeseran terbesar yang mungkin dengan menggunakan perbandingan yang dibentuk sebelum pencarian string. Fungsi pinggiran $b(k)$ didefinisikan sebagai prefix terpanjang dari $P[0..k]$ yang merupakan akhiran darisuffix $P[1..k]$. Fungsi ini bergantung pada karakter dalam pattern saja, bukan pada karakter dalam teks.

Algoritma KMP adalah sebagai berikut :

```
if a mismatch occurs at P[j]
(i.e. P[j] != T[i]), then
     k = j-1;
     j = b(k);      // obtain the new j
```

Berikut ini contoh dari pencocokan string menggunakan algoritma KMP :

- P :

a	b	a	a	b	a

1. Kalkulasi Fungsi Pinggiran

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

Misal diambil $b[4]$. $b[4]$ artinya ukuran terbesar prefix $P[0..4]$ yang juga merupakan suffix $P[1..4]$.

2. Pencocokan String

- T :

a	B	a	A	B	x	.	.

- P :

a	b	a	A	B	a

$j = 5$

A	B	A	a	b	a

$j_{new} = 2$

Tidak perlu mengulang perbandingan

Memulai perbandingan pada indeks b[4] = 2 pada P

Dalam melakukan perhitungan fungsi pinggiran, diperlukan sebesar $O(m)$. Sedangkan pencocokan string dilakukan sebesar $O(n)$. Sehingga, algoritma KMP memiliki kompleksitas waktu $O(m + n)$.

c) Algoritma Boyer-Moore

Algoritma Boyer-Moore atau BM adalah salah satu algoritma untuk mencari suatu string di dalam teks. Ide utama algoritma ini adalah mencari string dengan melakukan pembandingan karakter mulai dari karakter paling kanan dari string yang dicari. Algoritma ini juga menyimpan informasi fungsi pinggiran. Dengan mengunakan algoritma ini, secara rata-rata proses pencarian akan menjadi lebih cepat jika dibandingkan dengan algoritma lainnya.

Algoritma BM didasari dengan dua teknik, yaitu :

1) *The Looking-glass Technique*

Melakukan pencarian pola P pada teks T dengan melakukan pergeseran dari belakang ke depan dimulai dari karakter terakhir P.

2) Char*acter Jump Technique*

Melompati karakter ketika ada karkter pada pola P dan teks T yang tidak sama. Pada saat melakukan peloncatan karakter, terdapat 3 jenis kemungkinan kasus sebagai berikut.

1. Kasus I

Jika P mengandung karakter ‘x’ di sebelah kiri, maka geser P sehingga karakter di T[i] saat ini, yaitu ‘x’, sejajar dengan karakter ‘x’ dipola P (kemunculan terakhir karakter ‘x’ di pola).

2. Kasus II

Jika P mengandung karakter ‘x’, tetapi sudah berada di sebelah kanan dari karakter di P yang saat ini diperiksa, maka geser P sebanyak satu karakter sehingga sejajar dengan T[i+1] dengan T[i] adalah karakter di teks yang sedang diperiksa.

3. Kasus III

Jika kasus satu dan dua tidak berlaku, maka geser pola sehingga P[0] (karakter pertama pola) sejajar dengan T[i+1] dengan T[i] adalah karakter di teks yang sedang diperiksa.

d) Regular Expression

Regular expression adalah deretan karakter yang mendefinisikan sebuah pola dalam pencarian string. Algoritma regular expression mencari suatu pola dengan *parser*, sehingga algoritma ini adalah jenis algoritma yang tidak *exact-matching*.

Berikut ini beberaapa notasi umum yang digunakan dalam regular expression :

Notasi	Penjelasan
.	Karakter apapun selain *newline*
\.	Periode
^	Awal dari string
$	Akhir dari string
\d, \w, \s	Sebuah digit, karakter huruf ataupun spasi
\D, \W, \S	Apapun selain sebuah digit, karakter huruf ataupun spasi
[abc]	Karakter a,b, atau c
[a-z]	Karakter a sampai z
[^abc]	Selain karakter a, b, atau c
aa|bb	aa ataupun bb
?	Nol atau satu elemen berturut
*	Nol atau lebih elemen berturut
+	Nol atau satu elemen berturut
{n}	Tepat n elemen berturut
{n,}	n atau lebih elemen berturut
{m,n}	Antara m dan n elemen berturut

III. PEMBAHASAN

Untuk mendeteksi bahasa, sebuah aplikasi membutuhkan data yang berisi informasi berbagai bahasa dari berbagai negara. Secara statistik, setiap bahasa memiliki spesifik pola karakter dan frekuesi setiap karakternya. Namun, untuk teks yang panjang, hal tersebut sama sekali tidak dapat dijadikan acuan untuk melakukan pendeteksian bahasa. Oleh karena itu, dibutuhkan data berupa kamus bahasa sebagai acuan dalam melakukan pencocokan string untuk mendeteksi bahasa.

Terdapat beberapa prosedur yang disiapkan untuk aplikasi agar mampu mendeteksi bahasa. Berikut ini penjelasan dari langkah-langkah tersebut :

1) Membentuk data besar yang beirisi bahasa dari semua negara dan daerah sebagai sumber acuan

2) Membuat algoritma untuk mendeteksi bahasa

3) Ketika aplikasi dijalankan, aplikasi menerima inputan string. String tersebut dikirimkan ke dalam algoritma pendeteksi.

4) Pendeteksian bahasa dilakukan dengan mencocokan string inputan dengan data yang ada di dalam kamus berdasarkan algoritma tertentu.

5) Bahasa dengan bentukan string dalam kamus yang paling cocok dengan string input menjadi solusi untuk ditampilkan sebagai jawaban dari aplikasi.

Dalam makalah ini, dilakukan analisa untuk menentukan algoritma yang paling tepat untuk digunakan dalam mendeteksi bahasa. Algoritma yang akan dibandingkan adalah sesuai dengan pada teori dasar yaitu algoritma knut-morris-pratt, algoritma boyer-moore, dan regular expression.

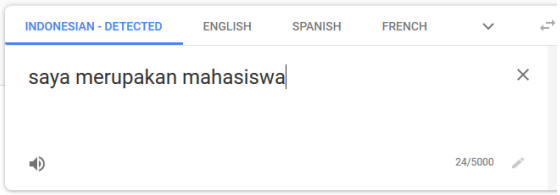
Berdasarkan teori dasar yang sudah dijelaskan, penulis memiliki hipotesis bahwa diantara ketiga algoritma, regular expression adalah yang paling efisien dan tepat digunakan dalam program. Kamus yang disimpan dalam data adalah kumpulan kata yang dimiliki dalam bahasa tertentu. Sedangkan string inputan berupa kalimat sehingga diperlukan pemecahan kalimat menjadi kata perkata untuk melakukan *exact-matching* jika pencocokan menggunakan algoritma knutt-morris-pratt ataupun algoritma boyer-moore.

Berikut ini ilustrasi proses yang dirancang untuk dilakukan oleh program :

1) Melakukan input string

```
Input kalimat : “Saya seorang mahasiswa”
```

“Saya seorang mahasiswa” adalah kalimat yang akan dideteksi bahasanya.



Gambar 2. Google translate mendeteksi “Saya merupakan mahasiswa” adalah Bahasa Indonesia

2) Program membagi kalimat menjadi kata-per-kata

```
Word[0] = “Saya”
Word[1] = “seorang”
Word[2] = “mahasiswa”
```

3) Untuk setiap kata, lakukan proses pencocokan string

Misalkan proses dilakukan terhadap word[0].

```
Word[0] = “Saya”
```

Dalam proses ini, pencarian kata “Saya” dalam kamus akan menggunakan algoritma pencocokan string. Masing-masing kode program dari prosedur atau fungsi algoritma adalah berikut ini :

a) Algoritma KMP

```
public static int kmpMatch(String text,String
pattern) {
    int n = text.length();
    int m = pattern.length();
    int fail[] = computeFail(pattern);
    int i = 0;int j = 0;

    while (i<n) {
       if (pattern.charAt(j) == text.charAt(i)){
          if (j==m-1)
             return i-m+1;i++;j++;
       }else if (j>0)
             j = fail[j-1];
       else
             i++;
    }

    return -1; //no match
} // end of kmpMatch

public static int[] computeFail(String pattern)
{
    int fail[] = new
    int[pattern.length()];
    fail[0] = 0;
    int m = pattern.length();
    int j = 0;
    int i = 1;

    while (i<m) {
       if (pattern.charAt(j) ==
       pattern.charAt(i)) { // j+1 chars match
          fail[i] = j+1;
          i++;
          j++;
    } else if (j>0) { // j followsmatching
          prefixj = fail[j-1];
    } else {
          fail[i] = 0;
          i++;
    }

    return fail;
} // end of computeFail()}
```

b) Algoritma Boyer-Moore

```
public static int bmMatch(String text, String
pattern) {
   int last[] = buildLast(pattern);
   int n = text.length();
   int m = pattern.length();
   int i = m-1;

   if (i > n-1)
      return -1; // no match if pattern is longer
than text
      int j = m-1;
      do {
         if (pattern.charAt(j) == text.charAt(i))
            if (j == 0)
               return i;
            else {
               i--;
               j--;
            }
         else {
            int lo = last[text.charAt(i)];
            //last occur
            i = i+m-Math.min(j, 1+lo);
            j = m-1;
         }
      }

   while (i<=n-1);
      return -1; //no match
}

public static int[] buildLast(String pattern)
/* Return array storing index of last occurrence
ofeach ASCII char in pattern. */{
   int last[] = new int[128];
   // ASCII char set
   for (int i=0; i<128; i++)
       last[i] = -1;
       //initialize array
   for (int i=0; i<pattern.length(); i++)
       last[pattern.charAt(i)] = i;
       return last;
} // end of buildLast()
```

Dalam proses pencocokan, sebelumnya menentukan algoritma yang tepat, penulis mengusulkan cara sebagai berikut:

1. Misalkan kata “saya” sudah berada di dalam Kamus Bahasa Indonesia.
2. Ketika huruf pertama yaitu “S” dicocokkan dengan awal kamus. Karakter akan menemukan ketidakcocokan dengan karakter “A”
3. Dilakukan pencarian, bagian kamus dengan mencocokan huruf awal kata “Saya” dengan huruf awal setiap kata pada kamus.
4. Ketika huruf dengan awalan S sudah ditemukan, pencocokan akan dilanjutkan ke karakter kedua pada “Saya”.
5. Huruf kedua pada kata “Saya” melakukan langkah yang sama seperti langkah 2-4 dengan mencocokan karakter pada urutan yang sama pada “Saya” dan pada huruf dalam kamus.
6. Pencarian berhenti ketika “Saya” sudah mencapai akhir kata dan berhasil menemukan kecocokan dengan kata pada kamus. Jawaban dari bahasa yang dideteksi adalah Bahasa Indonesia.
7. Misalkan, kata “Saya” bukan merupakan bahasa Indonesia, maka tidak ditemukan kecocokan sehingga pencocokan string akan dilakukan pada kamus bahasa lain yang ada di dalam data.

Untuk lebih jelas, diberikan ilustrasi algoritma yang penulis uraikan sebelumnya sebagai berikut.

- Word[0] :

s	a	y	a

- Mencocokan setiap kata dengan membandingkan huruf pertamanya saja.

  Kata dalam kamus :

a

  Tidak cocok

s	a	y	a

  Kata dalam kamus :

S

  cocok

s	a	y	a

- Mencocokan setiap kata dengan membandingkan huruf keduanya saja.

  Kata dalam kamus :

s	a

  cocok

s	a	y	a

- Mencocokan setiap kata dengan membandingkan huruf ketiganya saja.

  Kata dalam kamus :

s	a	a

  tidak cocok

s	a	y	a

  Kata dalam kamus :

s	a	b

  tidak cocok

s	a	y	a

  … seterusnya hingga

  Kata dalam kamus :

s	a	y

  cocok

s	a	y	a

- Mencocokan setiap kata dengan membandingkan huruf keempatnya saja.

  Kata dalam kamus :

s	a	y	a

  cocok

S	a	y	a

- Kata “Saya” ditemukan dalam kamus Bahasa Indonesia.

Setelah kata pertama dalam kalimat berhasil ditemukan bahasanya, kata selanjutnya dicari kembali menggunakan algoritma yang serupa. Pada akhir program, akan dihitung presentase dari bahasa yang disimpulkan dari tiap kata pada kalimat. Bahasa dengan presentase terbesar menjadi jawaban bahasa dari kalimat yang diinput.

Jika kita telaah, algoritma yang penulis tawarkan memiliki beberapa kemiripan dengan algoritma Knutt-Morris-Pratt. Modifikasi yang dilakukan terhadap algoritma adalah ketika ditemukan ketidakcocokan pada suatu karakter, pemeriksaan akan dilakukan pada kata yang berbeda tanpa memerika ulang awal kata yang telah divalidasi cocok sebelumnya.

IV. PENUTUPAN

a) Kesimpulan

Pendeteksian bahasa dapat dilakukan dengan menggunakan algoritma pencocokan string. Rancangan umum program adalah program menerima string inputan, kemudian dikirimkan sebagai parameter kedalam fungsi atau prosedur pencocokan string. Fungsi atau prosedur tersebut melakukan pencocokan kalimat dengan kumpulan kata pada kamus bahasa. Algoritma yang paling efisien dan tepat adalah algoritma knutt-morris-pratt atau KMP.

b) Saran

Kesimpulan yang didapatkan adalah berdasarkan analisis terhadap proses dan teori dasar. Untuk mendapatkan kesimpulan yang lebih tepat, perlu dilakukan eksperimen dengan melakukan rancangan program secara sederhana.

UCAPAN TERIMA KASIH

Pertama-tama penulis mengucapkan ucapan terima kasih kepada Allah SWT karena dengan izin dan rahmatnya, penulis memiliki kesempatan untuk membuat makalah ini dengan keadaan sehat wal'afiat.

Terima kasih pula dengan kedua orang tua penulis Bapak Admarius Sirjon dan Ibu Eliya Azis dengan segala dorongan dari segi materil maupun moril yang mendorong penulis untuk terus giat dalam menuntut ilmu dan menjadi seorang yang bertanggung-jawab dan menjadi insan yang bermanfaat kedepannya.

Tidak dilupakan, terima kasih kepada Ibu Dr. Nur Ulfa Maulidevi selaku dosen pengajar penulis dalam mata kuliah Strategi Algoritma ini yang selalu bersemangat dan menginspirasi dalam berbagi ilmu-ilmunya yang bermanfaat untuk penulis dan juga memberikan pengalaman-pengalaman unik yang penulis rasakan selama pengajaran beliau ini.

REFERENCES

[1] Munir, Rinaldi.Diktat Kuliah IF2251 Strategi Algoritmik. Institut Teknologi Bandung. 2007.

[2] A

[3] https://www.techopedia.com/definition/8801/pattern-matching. Diakses pada tanggal 26 April 2019 pada pukul 00.46 WIB.

[4] https://whatis.techtarget.com/definition/string. Diakses pada tanggal 26 April 2019 pada pukul 07.06 WIB.

[5] https://www.scribd.com/document/94376225/Algoritma-Boyer-Moore. Diakses pada tanggal 26 April 2019 pada pukul 09.06 WIB.

[6] https://piptools.net/algoritma-boyer-moore-search/. Diakses pada tanggal 26 April 2019 pada pukul 09.10 WIB.

[7] https://piptools.net/algoritma-kmp-knuth-morris-pratt/. Diakses pada tanggal 26 April 2019 pada pukul 09.10 WIB.

[8] http://blog.unnes.ac.id/novitasetiasih/2015/11/27/pengertian-dan-konsep-bahasa/. Diakses pada 26 April 2019 dengan pukul 09.37 WIB.

[9] http://oktonion.blogspot.com/2015/04/pengertian-bahasa-dan-jenis-jenis-bahasa.html. Diakses pada 26 April 2019 dengan pukul 09.42 WIB.

[10] http://portalsatu.com/read/budaya/terpetakan-jumlah-bahasa-di-seluruh-dunia-di-mana-posisi-indonesia-5960. Diakses pada 26 April 2019 dengan pukul 09.53 WIB.

[11] https://blog.algorithmia.com/introduction-language-identification/. Diakses pada 26 April 2019 dengan pukul 10.51 WIB.

[12] https://stackoverflow.com/questions/7670427/how-does-language-detection-work. Diakses pada 26 April 2019 dengan pukul 10.51 WIB.

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, 26 April 2019

Aisyah Nurul Izzah Adma
13517046