

Penerapan Algoritma Boyer-Moore Untuk Analisis Sentimen Berita-Berita di Indonesia Lewat Aplikasi Berbasis Android

Ahmad Faiz Sahupala

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
faiz.sahupala29@gmail.com

Abstract— Makalah ini menjelaskan sedikit tentang bagaimana menerapkan algoritma *Boyer-Moore* untuk menganalisis sentimen dari berita-berita di Indonesia dengan menunjukkan persentase konten yang positif, negatif, dan netral untuk berita yang dicari lewat aplikasi berbasis android. Algoritma *Boyer-Moore* merupakan salah satu algoritma untuk pencocokan *string*. Pembuatan Makalah ini bertujuan untuk menerapkan algoritma *Boyer-Moore* untuk mencocokkan berita-berita yang sudah di ambil dari internet dengan keyword konten yang telah di inisiasi terlebih dahulu. Penulis memilih Algoritma *Boyer-Moore* karena akan lebih efektif dalam pencocokan *string*.

Keywords— Algoritma *Boyer-Moore*, berita, pencocokan *string*, analisis sentimen, android

I. PENDAHULUAN

Berita merupakan informasi baru mengenai sesuatu yang terjadi, disajikan lewat bentuk cetak, siaran, Internet, atau dari mulut kepada orang ketiga atau orang banyak. Seiring berjalannya waktu, berita kini lebih sering dipublikasi lewat internet dan banyak portal berita yang sudah menyediakan aplikasi untuk mengakses konten-konten berita lewat aplikasi *handphone*.



GAMBAR 1. Contoh Portal Berita Indonesia

Di Indonesia, sudah cukup banyak berita – berita yang tersebar luas tentang suatu entitas. Sebagai contoh, jika

mencari suatu berita tentang “jokowi” akan banyak berita berita yang muncul yang menceritakan tentang “jokowi”. Tetapi, pembaca terkadang hanya melihat dari satu sisi berita , sehingga langsung menyimpulkan apakah entitas yang dicari sering diberitakan baik, buruk, atau netral saja. Padahal masih banyak berita tentang entitas tersebut yang dapat dicari dan berbeda sudut pandang. Oleh karena itu, penulis ingin mencoba menyelesaikan permasalahan tersebut, agar orang-orang di Indonesia tidak menyimpulkan secara cepat sebuah entitas yang diberitakan. Penulis akan membantu lewat teknologi yang akan menghasilkan analisis sentimen berita yang dapat memvisualisasikan data berapa banyaknya konten baik/buruk/netral suatu entitas yang diberitakan.

Dalam hal ini, penulis membuat sebuah aplikasi berbasis android untuk melakukan analisis sentimen berita-berita di Indonesia dan memisahkannya berdasarkan sifat konten berita tersebut, seperti konten positif, konten negatif, dan netral. Berita-berita tersebut didapatkan dari newsapi.org yang membuka API News gratis. Lalu, untuk pencocokan stringnya penulis memilih untuk menggunakan Algoritma *Boyer-Moore* dikarenakan algoritma *Boyer-Moore* merupakan salah satu algoritma yang efektif untuk melakukan pencocokan *string*. Aplikasi ini akan membantu pengguna melihat statistik seberapa banyak berita yang mengeluarkan konten positif, negatif, dan netral pada suatu *keyword* yang ada di berita-berita Indonesia.

II. TEORI DASAR

A. Algoritma Boye-Moore

Pencarian menggunakan algoritma Boyer-Moore dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Tidak seperti algoritma pencocokan *string* pada umumnya, algoritma *Boyer-Moore* memulai pencocokan *string* dari sebelah kanan *pattern*. Oleh karena itu, akan lebih banyak informasi yang didapat.

Misalnya ada sebuah usaha pencocokan yang terjadi pada teks $[i..i+n-1]$, dan anggap ketidakcocokan pertama terjadi di antara teks $[i+j]$ dan $pattern[j]$, dengan $0 < j < n$. Berarti, teks $[i+j+1..i+n-1]=pattern[j+1..n-1]$ dan $a=teks[i+j]$ tidak

sama dengan $b = \text{pattern}[j]$. Jika u adalah akhiran dari pattern sebelum b dan v adalah sebuah awalan dari pattern , maka penggeseran-penggeseran yang mungkin adalah:

1. Penggeseran *good-suffix* yang terdiri dari menyejajarkan potongan teks $\text{teks}[i+j+1..i+n-1] = \text{pattern}[j+1..n-1]$ dengan kemunculannya paling kanan di pattern yang didahului oleh karakter yang berbeda dengan $\text{pattern}[j]$. Jika tidak ada potongan seperti itu, maka algoritme akan menyejajarkan akhiran v dari teks $\text{teks}[i+j+1..i+n-1]$ dengan awalan dari pattern yang sama.
2. Penggeseran *bad-character* yang terdiri dari menyejajarkan teks $\text{teks}[i+j]$ dengan kemunculan paling kanan karakter tersebut di pattern . Bila karakter tersebut tidak ada di pattern , maka pattern akan disejajarkan dengan teks $\text{teks}[i+n+1]$.

Secara sistematis, langkah-langkah yang dilakukan algoritme Boyer-Moore pada saat mencocokkan string adalah:

1. Algoritme Boyer-Moore mulai mencocokkan pattern pada awal teks.
2. Dari kanan ke kiri, algoritme ini akan mencocokkan karakter per karakter pattern dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
 1. Karakter di pattern dan di teks yang dibandingkan tidak cocok (mismatch).
 2. Semua karakter di pattern cocok. Kemudian algoritme akan memberitahukan penemuan di posisi ini.
3. Algoritme kemudian menggeser pattern dengan memaksimalkan nilai penggeseran *good-suffix* dan penggeseran *bad-character*, lalu mengulangi langkah 2 sampai pattern berada di ujung teks.

B. Android Studio

Android Studio adalah sebuah IDE untuk Android Development yang diperkenalkan google pada acara Google I/O 2013. Android Studio merupakan pengembangan dari Eclipse IDE dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android. *Android Studio* dipakai untuk penulis mengembangkan dan mengimplementasikan algoritma *Boyer-Moore* dan menjadi penghubung antara *User* dan aplikasi yang akan penulis buat.



Gambar 2. IDE Android Studio

C. JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, standar ECMA-262 Edisi ke-3 – Desember 1999. JSON merupakan format text yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat itu, menjadikan JSON ideal sebagai bahasa pertukaran-data.

Dalam hal ini, JSON digunakan untuk java mengambil data dari newsapi.org agar dapat memiliki data data berita sesuai dengan yang diminta oleh program.

```
[
  {
    "name": "Shona",
    "location": "LA",
    "id": 8
  },
  {
    "name": "Shona",
    "location": "LA",
    "id": 9
  }
]
```

Gambar 3. Contoh Potongan Kode JSON

D. Java Language

Java adalah bahasa pemrograman yang sudah ada dari tahun 1990-an. Bahasa pemrograman java kian berkembang dan mendominasi di berbagai bidang teknologi. Bahasa pemrograman java merupakan bahasa pemrograman berorientasi objek (*Object Oriented Programming*). Java pun akrab dengan dunia saintifik dan akademik. Cukup banyak akademisi di Indonesia yang menggunakan bahasa Java sebagai alat menyelesaikan skripsi atau tugas akhir dengan berbagai topik seperti kecerdasan buatan, *data mining*, *architecture*, dan yang penggunaannya paling terbesar adalah pembuatan aplikasi android native. Oleh karena itu, penulis ingin menggunakan bahasa java sebagai acuan untuk menerapkan algoritma *booyer-moore* untuk pencocokan

string. Karena android sendiri dibentuk dengan bahasa pemrograman Java sebagai bahasa utamanya. Bahasa pemrograman java mudah dipelajari, karena hampir mirip dengan bahasa pemrograman lainnya seperti Free Pascal, C, C++, C#, dan lain lain. Yang membedakan adalah bahasa ini merupakan bahasa pemrograman berorientasi objek. Oleh karena itu, dalam bahasa java kita akan sering mendengar istilah public, class, static, void, dan lain lain. Berikut adalah contoh potongan program dalam bahasa pemrograman Java :

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World...");
    }
}
```

Gambar 4. "Hello World" dalam bahasa java

E. NEWS API from newsapi.org

newsapi.org adalah sebuah web yang menyediakan data data berita dari seluruh dunia dan membuka API yang dapat diakses secara gratis dan mudah dengan melalui file JSON. Newsapi.org dapat memberikan data data berita yang penulis minta, yaitu berita berita di Indonesia seperti kompas, detik, liputan6, dan banyak lagi yang akan diolah oleh penulis untuk menghasilkan *output* yang penulis inginkan

III. ANALISIS

A. Pengaturan retrofit untuk teknis request internet

Pada Android Studio, kita harus melakukan *request* untuk mengambil data ke Internet. Pengambilan data ke Internet, bisa dilakukan oleh banyak hal. Untuk kasus ini, penulis memakai *retrofit* service untuk membantu penulis mengimplementasikannya di Android Studio.

Retrofit adalah *library* REST API Client untuk Android dan java dari squareup. Untuk pertama-tama *library* harus di deklarasikan di file *build.gradle*, di dalam block *dependencies* seperti ini,

```
implementation
'com.google.code.gson:gson:2.8.1'
implementation
'com.squareup.retrofit2:converter-
gson:2.2.0'
implementation
'com.squareup.retrofit:retrofit:1.9.0'
```

Block diatas akan meng-*import* kelas kelas dari retrofit agar dapat dipakai oleh penulis. Setelah itu, membuat sebuah kelas *RetrofitClientUtils.java* untuk mengatur REST API *client* yang akan dipanggil nantinya. Dan membuat sebuah *static* method untuk pemanggilannya.

```
public static Retrofit client() {
    Gson gson = new GsonBuilder().
        setLenient().create();
    HttpLoggingInterceptor logging =
```

```
new HttpLoggingInterceptor()
.setLevel(HttpLoggingInterceptor.Level.BODY);

OkHttpClient.Builder okHttpClient =
new
OkHttpClient.Builder().addInterceptor(log
ging);

retrofit = new Retrofit.Builder().
baseUrl(BASE_URL).
client(okHttpClient.build()).

addConverterFactory(GsonConverterFactory.
create(gson)).
build();
return retrofit;
}
```

B. Pengaturan newsapi.org di Android Studio

Pada Android Studio, hal pertama yang harus dilakukan adalah mengenal *endpoint* dari newsapi.org yang akan mengirimkan data data berita ke Android Studio. Penulis membuat suatu kelas *interface* untuk mengambil *endpoint* apa yang akan di *request* ke newsapi.org. Oleh karena itu, penulis membuat kelas *NewsService.java*

```
public interface NewsService {
    @GET("everything?")
    Call<NewsModel>
callNews(@QueryParam(encoded = true)
Map<String, String> query,
@Query("apiKey") String apiKey,
@Query("language") String language,
@Query("page") int page);
}
```

Kelas tersebut akan mengembalikan *endpoint* yang di inginkan oleh pengguna. Sebagai contoh *endpoint* yang akan dibalikan adalah "BASE_URL/everything?" dengan BASE_URL nya adalah newsapi.org. *endpoint* tersebut akan mengembalikan data data berita secara random. Agar penulis dapat menyaring berita apa saja yang akan didapatkan, penulis menambah *query* language untuk menyatakan berita sesuai id regional negara yang mengisi *query* tersebut. Dalam kasus ini, penulis menggunakan id "ID" yang menyatakan regional negara Indonesia. Lalu, *query* apiKey harus di isi dengan sebuah string unik yang diberikan oleh newsapi.org ketika penulis mendaftarkan akun. Terakhir, *query* page yang menyatakan mau mengambil data berita di halaman ke berapa.

Method *callNews* akan dipanggil ketika program ingin mengirim *request* untuk mendapatkan data berita yang diinginkan setelah program memberikan *query - query* yang diminta method *callNews*.

C. Pemanggilan API newsapi.org

Setelah kita melakukan pengaturan *retrofit* dan juga interface *news*. Kita dapat mengimplementasikannya di dalam Suatu Activity dimana kita ingin menampilkan datanya. Untuk pemanggilan kita dapat membuat suatu method *callDataNews()* seperti berikut

```
private void callDataNews() {
    String query =
    getIntent().getStringExtra("query");
    Map<String, String> map = new
    HashMap<>();
    map.put("q", query);
    Call<NewsModel> call =
    sendNewsRequest().callNews(map,
    "cf74c3ea94174f62a677fcf4355900af", "id",
    10);
    call.enqueue(new
    Callback<NewsModel>() {
        @Override
        public void
        onResponse(Call<NewsModel> call,
        Response<NewsModel> response) {
            if (response.body() != null) {
                calculateData(response.body());
            }
        }
        @Override
        public void
        onFailure(Call<NewsModel> call, Throwable
        t) {
            Toast.makeText(ResultActivity.this,
            "something was error",
            Toast.LENGTH_SHORT).show();
            Log
            .d("ResultActivityDebug", "onFailure: "
            + t.toString());
        }
    });
}
```

Pada pemanggilan api, hal yang pertama dilakukan adalah mengambil query yang diisi oleh pengguna (diperlihatkan di baris pertama). Lalu, membuat suatu object yang berisi data data *query* yang ingin di kirimkan. Lalu *me-request* data ke endpoint yang telah dibuat. Dan menunggu hasil dari *request* ke server yang akan di kembalikan lagi ke program. Pengembalian *request* dibagi menjadi 2, yang pertama jika hasil *request* gagal, maka program akan melakukan *logging* dan menuliskan pesan error ke *log* program. Jika hasil *request* berhasil, maka data yang telah diambil (*respond.body*) akan dilanjutkan ke perhitungan data yang lebih lanjut akan dijelaskan di *point* berikutnya.

D. Perhitungan Data dan Pengimplementasian Algoritma *booyer-moore*

Setelah mendapatkan data dari API newsapi.org, penulis mengolahnya untuk menghasilkan analisis sentimen yang baik. Pertama-tama penulis membuat suatu array of string yang berisi perkataan perkataan baik yang penulis beri nama variabelnya *positive*, dan juga array of string perkataan perkataan buruk yang penulis beri nama *negative*.

String[] positive = { "berani", "bangga", "hebat", "pintar", "bagus", "baik", "blusukan", };	String[] negative = { "jelek", "turun", "pergi", "komunis", "menolak", };
--	---

Kedua array tersebut akan dipakai untuk pencocokan string dengan Algoritma *booyer-moore*. Oleh karena itu, penulis mengimplementasikan Algoritma *booyer-moore* sebagai salah satu method di Java yang akan mengembalikan index dari pattern yang sama.

Pertama, penulis membuat method *buildLast* yang bertujuan untuk mengembalikan sebuah array of integer yang menyatakan indeks kemunculan terakhir tiap *char* dari pattern yang ada pada parameter input.

```
public static int[] buildLast(String
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;
}
```

Setelah itu, membuat method *bmMatch* yang bertujuan untuk mengembalikan indeks dari pattern yang dicari, dan akan mengembalikan -1 jika pattern tidak ditemukan.

```
private 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;

    int j = m-1;
    do {
```

```

        if (pattern.charAt(j) ==
text.charAt(i)) {
            if (j == 0)
                return i; // match

            else { // looking-glass
technique

                i--;
                j--;
            }
        } else { // character jump
technique

            int lo =
last[text.charAt(i)]; //last occ
            i = i + m - Math.min(j,
1+lo);
            j = m - 1;
        }
    } while (i <= n-1);

    return -1; // no match
}

```

Setelah Algoritma *booyer-moore* di implementasikan, kita menginisiasi variabel positif, negatif, netral yang menyatakan jumlah berita tiap kontennya, yang semulanya di inisiasi 0 semua. Lalu, melakukan iterasi dari tiap deskripsi berita yang di dapatkan. Tiap melakukan iterasi deskripsi berita, pertama penulis melakukan iterasi *positive* yang mana merupakan kumpulan kata baik dan melakukan pencocokan string dengan algoritma *booyer moore* dengan tiap kata di array *positive*. jika, ditemukan kata yang sama, maka iterasi diberhentikan dan menambahkan jumlah ke variabel positif, dan melanjutkan iterasi dengan deskripsi berita selanjutnya. Jika tidak ditemukan kata positif, maka dilanjutkan melakukan iterasi dengan variabel *negative* yang mana merupakan kumpulan kata kata buruk dan melakukan pencocokan string dengan algoritma *booyer-moore* dengan tiap kata di array *negative*. Jika ditemukan kata yang sama, maka iterasi diberhentikan dan menambahkan jumlah ke bariabel negatif, dan melanjutkan iterasi dengan deskripsi berita selanjutnya. Jika tidak ditemukan kata negatif, maka penulis menambahkan jumlah variabel netral, karena berita tersebut tidak positif dan juga tidak negatif.

```

private void calculateData (NewsModel
body) {
    int positif = 0;
    int negatif = 0;
    int netral = 0;

    //pengecekan
    for (int i = 0; i < 500; i++){
        boolean found = false;

        //positif
        for (String pattern : positive){

```

```

        if
        (bmMatch(body.getArticles().get(i).getDes
cription().toLowerCase(), pattern) != -
1){

            positif++;
            found = true;
            break;
        }

        //negatif
        if (!found){
            for (String pattern :
negative){

                if
                (bmMatch(body.getArticles().get(i).getDes
cription().toLowerCase(), pattern) != -
1){

                    negatif++;
                    found = true;
                    break;
                }
            }

            //netral
            if (!found) netral++;
        }

        setResultNews(positif, negatif,
netral);
    }
}

```

Setelah itu, kita sudah mendapatkan data data berapa banyak berita positif, negatif, dan netral. Yang selanjutnya akan ditampilkan ke layar Android.

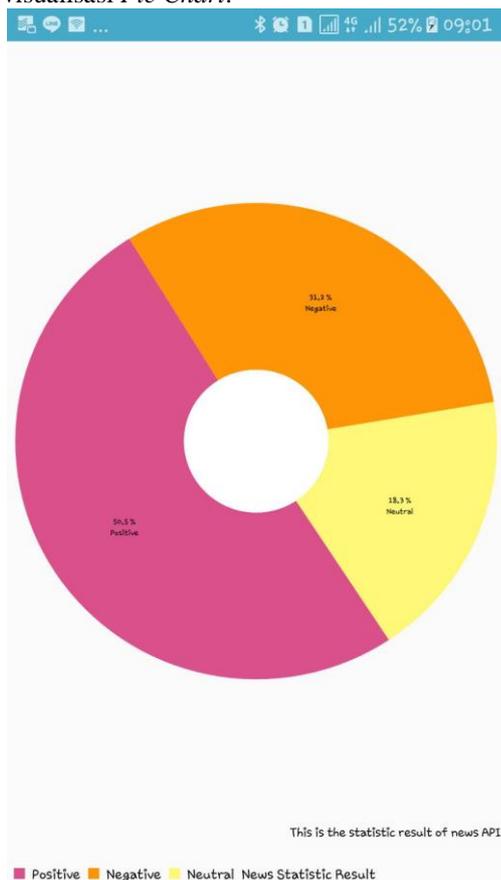
IV. MASUKAN DAN KELUARAN

Setelah program selesai dibuat, penulis melakukan percobaan dengan menggunakan *keyword* "jokowi" seperti pada gambar berikut



Gambar 5. Contoh Pengisian *keyword*

Setelah itu, program akan langsung mengirim dan melakukan eksekusi sesuai dengan yang telah dijelaskan pada bab sebelumnya. Program akan menampilkan data statistiknya dengan visualisasi *Pie Chart*.



V. KESIMPULAN

Pengiriman Berita sudah sangat banyak di Internet. Tetapi, sebagai manusia yang baik, kita tidak boleh dengan mudah menyimpulkan suatu entitas yang diberitakan buruk atau tidak sebelum kita memiliki data yang jelas. Oleh karena itu, kita

data memanfaatkan teknologi dengan baik untuk mengatasi masalah tersebut. Algoritma *booyer-moore* bekerja dengan baik sebagai teknologi yang membantu penulis menyelesaikan masalah tersebut dengan mangkus dan sangkil.

REFERENCES

- [1] <https://www.codepolitan.com/rest-api-client-sederhana-dengan-retrofit-pada-android-studio-58986d62c46ae> , diakses pada 06 Mei 2018 pukul 00:24
- [2] <https://developer.android.com/studio/intro/index.html?hl=id> , diakses pada 24 April 2018 pukul 18:47.
- [3] <https://www.codepolitan.com/mengenal-dan-memulai-pemrograman-java-belajar-java> , diakses pada 24 April 2018 pukul 18:48
- [4] <http://ngodingz.blogspot.co.id/2016/03/pengenalan-android-studio.html> , diakses pada 24 April 2018 pukul 19:01
- [5] <https://www.json.org/json-id.html> , diakses pada 24 April 2018 pukul 19:45
- [6] <https://cms-assets.tutsplus.com/uploads/users/1469/posts/27871/image/filter.png> , diakses pada 24 April 2018 pukul 20:10
- [7] Munir, Rinaldi. 2006. *Diktat Strategi Algoritma*.

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, 25 April 2018

Ahmad Faiz Sahupala, 13516065