

Penerapan Algoritma *Pattern Matching* pada Aplikasi Berita KURIO dengan Memanfaatkan Teknologi *Machine Learning* dan Mesin Rekomendasi

Candra Heslen Parera - 13515019
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515019@std.stei.itb.ac.id

Abstract—Algoritma *pattern matching* digunakan untuk mencari suatu string (*pattern*) di dalam teks. Algoritma yang termasuk dalam *pattern matching* yaitu Algoritma Brute Force, Algoritma Knuth-Morris-Pratt, dan Algoritma Boyer-Moore. Pengaplikasian algoritma *pattern matching* ini contohnya pada pencarian string di dalam Editor Text, Web Search Engine, Analisis Citra, Bionformatics, dll. Makalah ini akan membahas salah satu pengaplikasian *pattern matching* yaitu pada aplikasi berita Kurio. Aplikasi berita Kurio ini akan menampilkan berita-berita yang sesuai dengan keinginan pengguna. Keinginan pengguna dapat diketahui dari pemanfaatan teknologi machine learning dan mesin rekomendasi yang diterapkan pada aplikasi. Selanjutnya dengan memanfaatkan teknologi machine learning dan mesin rekomendasi tersebut, *pattern matching* digunakan sebagai pengekskusi berita-berita atau informasi apa saja yang harus ditampilkan ke pengguna.

Keywords—*machine learning; pattern matching, mesin rekomendasi, Kurio.*

I. PENDAHULUAN

Seiring berkembangnya zaman, manusia sekarang tidak pernah lepas dari yang namanya smartphone. Dimanapun mereka berada smartphone adalah barang yang harus wajib dibawa. Karena ketergantungan manusia sekarang terhadap smartphone itulah para pengembang teknologi membuat suatu aplikasi yang dirancang untuk dapat di install di dalam smartphone. Contohnya adalah aplikasi berita Kurio. Aplikasi berita ini adalah aplikasi yang menampilkan berita-berita atau informasi yang faktual. Jadi para penikmat pembaca berita tidak perlu membaca manual yaitu membeli koran atau dalam bentuk fisik lainnya, mereka hanya tinggal menginstall aplikasi ini pada smartphone mereka dan bisa mendapatkan berita yang diinginkan. Kelebihan aplikasi ini adalah dilengkapi dengan fitur video channel dan juga aplikasi ini menampilkan berita-berita atau informasi yang sesuai dengan si pengguna. Sesuai disini dalam artian bahwa berita yang ditampilkan merupakan hasil analisis tentang kebutuhan berita apa yang diinginkan pengguna dan berita apa yang tidak diinginkan. Dalam menampilkan berita tersebut, aplikasi ini menerapkan algoritma *pattern matching* yang memanfaatkan teknologi *machine learning* dan mesin rekomendasi.

II. LANDASAN TEORI

A. Algoritma *Pattern Matching*

Algoritma *pattern matching* atau string matching dalam Bahasa Indonesia dikenal dengan istilah algoritma pencocokan string atau disebut juga pencarian string di dalam teks. [1]

Persoalan pencarian string dirumuskan sebagai berikut :
Diberikan :

1. T: teks (text), yaitu (long) string yang panjangnya n karakter
2. P: pattern, yaitu string dengan panjang m karakter ($m < n$) yang akan dicari di dalam teks.

Carilah (find atau locate) lokasi pertama di dalam teks yang bersesuaian dengan pattern.

Contoh 1

Pattern : not
Teks : nobody noticed him
 ↑target

Contoh 2

Pattern : apa
Teks : **Siapa** yang menjemput **Papa** dari kota
Balikpapan?

Dalam algoritma pencocokan string, teks diasumsikan berada di dalam memori, sehingga bila kita mencari string di dalam sebuah arsip, maka semua isi arsip perlu dibaca terlebih dahulu kemudian disimpan di dalam memori. [2]

Pada contoh 2, jika pattern muncul lebih dari sekali di dalam teks, maka pencarian hanya akan memberikan keluaran berupa lokasi pattern ditemukan pertama kali.

Berikut adalah Konsep String :

1. Asumsi S merupakan suatu string dengan ukuran m.

$$S = x_1x_2 \dots x_{m-1}$$

2. Sebuah prefix S merupakan substring $S[0..k]$
3. Sebuah suffix S merupakan substring $S[k..m-1]$
 - Dengan k adalah suatu index antara 0 sampai dengan m-1

Contoh 3

S = a n d r e w ; m = 6 ; dengan i = 0 adalah a

- Semua kemungkinan prefix S : “a”, “an”, “and”, “andr”, “andre”, “andrew”
- Semua kemungkinan suffix S : “w”, “ew”, “rew”, “drew”, “ndrew”, “andrew”

Ada tiga buah algoritma yang umum digunakan dalam pencocokan string yaitu algoritma Brute Force, algoritma Knuth-Morris-Pratt, dan algoritma Boyer Moore.

A.1 Algoritma Brute Force

Dengan asumsi bahwa teks berada di dalam array T[1..n] dan pattern berada di dalam array P[1..m], maka algoritma Brute Force pencocokan string adalah sebagai berikut:

1. Mula-mula pattern P dicocokkan pada awal teks T.
2. Dengan bergerak dari kiri ke kanan, bandingkan setiap setiap karakter di dalam pattern P dengan karakter yang bersesuaian di dalam teks T sampai:
 - a. semua karakter yang dibandingkan cocok atau sama (pencarian berhasil), atau
 - b. dijumpai sebuah ketidakcocokan karakter (pencarian belum berhasil)
3. Bila pattern P belum ditemukan kecocokannya dan teks T belum habis, geser pattern P satu karakter ke kanan dan ulangi langkah 2. [1]

Contoh 4

Teks : nobody noticed him
Pattern : not

nobody **not**iced him
 s=0 not
 s=1 not
 s=2 not
 s=3 not
 s=4 not
 s=5 not
 s=6 not
 s=7 **not**

Kompleksitas algoritma Brute-Force:

- Kompleksitas kasus terbaik adalah O(n). Kasus terbaik terjadi jika yaitu bila karakter pertama pattern P tidak pernah sama dengan karakter teks T yang dicocokkan. Pada kasus ini, jumlah perbandingan yang dilakukan paling banyak n kali.
- Kasus terburuk membutuhkan m(n - m + 1) perbandingan, yang mana kompleksitasnya adalah O(mn).

A.2 Algoritma Knuth-Morris-Pratt

Pada algoritma Knuth-Morris-Pratt (KMP) informasi ketidakcocokan pattern dengan teks digunakan disimpan untuk menentukan jumlah pergeseran. Algoritma KMP melakukan pergeseran lebih jauh sesuai dengan informasi yang disimpan, tidak seperti pada algoritma Brute Force di mana pergeseran dilakukan setiap satu karakter, sehingga waktu pencarian dapat

dikurangi secara signifikan. Algoritma Knuth-Morris-Pratt dikembangkan oleh D. E. Knuth, bersama-sama dengan J.H Morris dan V. R. Pratt. [1]

Persoalan algoritma KMP dapat didefinisikan sebagai berikut:

Misalkan A adalah alfabet dan $x = x_1x_2\dots x_k$, $k \in \mathbb{N}$, adalah string yang panjangnya k yang dibentuk dari karakter-karakter di dalam alfabet A.

Awalan (prefix) dari x adalah upa-string (substring) u dengan

$$u = x_1x_2\dots x_{k-1}, k \in \{1, 2, \dots, k-1\}$$

dengan kata lain, x diawali dengan u.

Akhiran (suffix) dari x adalah upa-string (substring) u dengan

$$u = x_k - b x_{k-b+1} \dots x_k, k \in \{1, 2, \dots, k-1\}$$

dengan kata lain, x diakhiri dengan v.

Pinggiran (border) dari x adalah upa-string r sedemikian sehingga

$$r = x_1x_2\dots x_{k-1} \text{ dan } u = x_k - b x_{k-b+1} \dots x_k, k \in \{1, 2, \dots, k-1\}$$

dengan kata lain, pinggiran dari x adalah upa-string yang keduanya awalan dan juga akhiran sebenarnya dari x. [1]

Contoh

Misalkan $x = abacab$.

Awalan sebenarnya dari x adalah

□, a, ab, aba, abac, abaca
(ket: □ = string kosong)

Akhiran sebenarnya dari x adalah

□, b, ab, cab, acab, bacab

Pinggiran dari x adalah

□, ab

Pinggiran □ mempunyai panjang 0, pinggiran ab mempunyai panjang 2.

Algoritma KMP melakukan proses awal terhadap pattern P dengan menghitung fungsi pinggiran yang mengindikasikan pergeseran s terbesar yang mungkin dengan menggunakan perbandingan yang dibentuk sebelum pencarian string. Hal ini dimaksudkan untuk mencegah pergeseran yang tidak berguna seperti pada algoritma Brute Force. [1]

Fungsi pinggiran hanya bergantung pada karakter-karakter di dalam pattern. Fungsi pinggiran b(j) didefinisikan sebagai ukuran awalan terpanjang dari P yang merupakan akhiran dari P[1..j].

Berikut adalah algoritma untuk menghitung fungsi pinggiran:

```
procedure HitungPinggiran(input m : integer, P : array[1..m] of char,
```

```
output b : array[1..m] of integer)  
{ Menghitung nilai b[1..m] untuk pattern P[1..m] }
```

Deklarasi

k,q : integer

Algoritma:

```
b[1]←0  
q←2  
k←0  
for q←2 to m do  
  while ((k > 0) and (P[q] ≠ P[k+1])) do  
    k←b[k]  
  endwhile  
  if P[q]=P[k+1] then  
    k←k+1  
  endif  
  b[q]=k  
endfor
```

Untuk melakukan pencocokan string, mulamula kita harus menghitung fungsi pinggiran untuk pattern tersebut. Selanjutnya dilakukan pencocokan pertama, samakan ujung kiri pattern dengan ujung kiri teks. Misalkan karakter cocok pada posisi [1..5] dan pada posisi 6 tidak ditemukan kecocokan, jumlah pergeseran selanjutnya ditentukan oleh pinggiran awalan P yang berkesesuaian. [1]

Kompleksitas algoritma KMP:

Untuk menghitung fungsi pinggiran dibutuhkan waktu $O(m)$, sedangkan pencarian string membutuhkan waktu $O(n)$, sehingga kompleksitas waktu algoritma KMP adalah $O(m+n)$.

A.3 Algoritma Boyer Moore

Pada algoritma Boyer Moore untuk menentukan pencocokan string ada dua teknik yang bisa digunakan, yaitu:

1. Teknik looking-glass

Mencari P di T dengan berjalan mundur pada P.

2. Teknik character-jump

Ketika tidak cocok di $T[i] = x$ maka karakter di $P[j]$ tidak sama dengan $T[i]$.

Pencarian string dengan menggunakan algoritma ini berbeda dengan algoritma KMP. Algoritma BM melakukan perbandingan pattern mulai dari kanan / karakter paling belakang dan lompatan maju dilakukan sejauh mungkin.

Kompleksitas algoritma Boyer-Moore:

Pada kasus terburuk $O(nm+A)$. Tapi, ketika alphabet (A) besar, maka waktunya cepat, sedangkan ketika A nya kecil lambat. Jadi Boyer-Moore bagus untuk teks English tapi tidak untuk binary.

B. Aplikasi Berita Kurio

Nama Kurio berasal dari bahasa Latin, *Cura/Curos* yang berarti “to care for” kemudian berkembang menjadi “Curiosity” dalam bahasa Inggris yang berarti “keinginan untuk mempelajari atau rasa ingin tahu akan sesuatu. Kurio sendiri adalah aplikasi untuk membaca konten yang bisa dipersonalisasi sesuai keinginan pengguna. [3]

Kurio merupakan sebuah aplikasi pembaca berita dan content aggregator, yaitu aplikasi yang menghimpun berita, artikel, dan informasi lainnya sesuai topik dan temanya dari berbagai sumber. Semua isi aplikasi Kurio tersebut berasal dari sumber portal bermacam-macam seperti Kompas, Liputan6, BBC, MetroTV, Viva, AntaraNews, Suara, IDNTimes, Beritagar, Kapanlagi, Merdeka, dan masih banyak lagi. [4]

Aplikasi ini sengaja dirancang untuk memberi kemudahan bagi penggunaannya untuk mendapatkan beragam informasi yang diinginkan. Lewat aplikasi Kurio ini, pengguna juga bisa mengeksplore, membaca, atau langsung berbagi berita kepada pengguna lainnya. Pengguna bisa menentukan konten apa yang ingin dibaca, menjelajah berbagai pilihan dan memilih topic – topik tertentu untuk secara rutin dibaca.

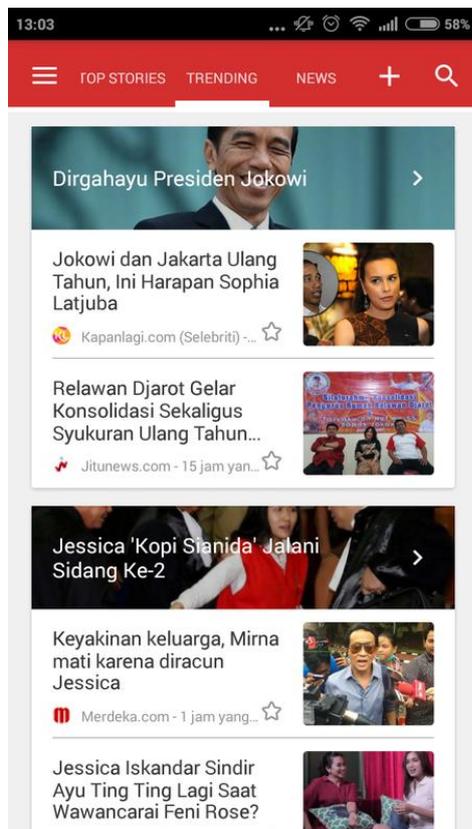
David Wayne Ika selaku Founder sekaligus CEO Kurio menegaskan bahwa aplikasi ini mengombinasikan beragam konten berita terbaru dan topik yang berguna, mulai dari *digital marketing*, media sosial, *healthy living*, hingga *startup*. Topik konten yang disajikan antara lain: News, Business, Economy, Sports, Soccer, Entertainment, Technology, Health hingga K-Pop dan Art Design.

Berikut tampilan antarmuka dari aplikasi Kurio :



(Sumber :

https://commondatastorage.googleapis.com/app-s-nesia-production-001/uploads/googleplay/screenshot_image/5751/27d4a81a-4ea5-4ddd-9644-d2bdf263caf2.png, diakses 18 Mei 2017 pukul 02.12 WIB)



(Sumber :

https://commondatastorage.googleapis.com/apps-nesia-production-001/uploads/entries/pictures/1968/content_Screenshot_2016-06-22-13-03-35.png, diakses 18 Mei 2017 pukul 02.12 WIB)

Aplikasi Kurio dilengkapi dengan teknologi *machine learning* dan mesin rekomendasi untuk menyajikan konten-konten yang diperlukan pengguna. Dari pemanfaatan teknologi tersebut, ketika pengguna semakin banyak menggunakan aplikasi, maka secara otomatis akan memisahkan konten yang diinginkan dan konten yang tidak diinginkan. [5]

C. Machine Learning

Machine learning dalam Bahasa Indonesia dikenal dengan sebutan pembelajaran mesin. Istilah *machine learning* pada dasarnya adalah proses computer untuk belajar dari data (learn from data).

Machine learning merupakan salah satu cabang dari disiplin ilmu Kecerdasan Buatan (*Artificial Intelligence*) yang membahas mengenai pembangunan sistem yang berdasarkan pada data.

Menurut Arthur Samuel, *Machine learning* adalah bidang studi yang memberikan kemampuan program komputer untuk belajar tanpa secara eksplisit diprogram. Sedangkan Tom M. Mitchell memberikan definisi sebagai “Sebuah program komputer yang belajar dari pengalaman (*experience*) E dari tugas yang dibebankan (*Task*) T dengan kinerjanya (*performance*) P yang terukur.

Untuk dapat mengaplikasikan *machine learning*, maka ada hal yang menjadi syarat *machine learning* dapat digunakan yaitu:

1. Terdapat data. Tanpa adanya data, maka computer tidak akan bisa belajar apa-apa. Karena computer tidak akan bisa belajar apa-apa, maka *machine learning* tidak bekerja. Data dibagi menjadi dua kelompok yaitu:
 - a. Data training, digunakan untuk melatih algoritma untuk mencari model yang cocok.
 - b. Data testing, digunakan untuk mengetes dan mengetahui performa model yang didapatkan pada tahapan testing.
2. Terdapat sebuah pola. Jika data yang ada bersifat acak. Maka *machine learning* tidak akan dapat menyelesaikan permasalahan tersebut.
3. Persoalan tersebut tidak bisa diselesaikan secara matematis. *Machine learning* menggunakan teori ilmu statistik (probabilitas dan pendekatan). Sehingga ketika sebuah peristiwa tidak dapat diselesaikan dengan pendekatan perhitungan matematik biasa, maka *Machine learning*-lah solusinya. [7]

Berikut adalah contoh penerapan *machine learning*:

1. Penerapan di bidang kedokteran contohnya adalah mendeteksi penyakit seseorang dari gejala yang ada. Contoh lainnya adalah mendeteksi penyakit jantung dari rekaman elektrokardiogram.
2. Pada bidang computer vision contohnya adalah penerapan pengenalan wajah dan pelabelan wajah seperti pada facebook. Contoh lainnya adalah penterjemahan tulisan tangan menjadi teks.
3. Pada bidang information retrieval contohnya adalah penterjemahan bahasa dengan menggunakan komputer, mengubah suara menjadi teks, dan filter email spam. [6]

D. Sistem Rekomendasi

Sistem rekomendasi (Recommender System) atau biasa disebut juga dengan istilah mesin rekomendasi adalah sebuah perangkat lunak untuk memberikan rekomendasi kepada pengguna mengenai produk yang dapat digunakannya. Produk ini dapat berupa barang elektronik, buku, musik, film, berita, dan sebagainya. Rekomendasi ini dibuat berdasarkan adanya personalisasi sehingga rekomendasi yang dihasilkan mungkin berbeda-beda bagi tiap user. Personalisasi ini dapat dihasilkan dari informasi user, yang berupa rating atau data transaksi user. [8]

Tujuan dari recommender system adalah untuk melakukan scanning secara otomatis dari informasi yang bermacam-macam dan sangat banyak jumlahnya untuk mendapatkan rekomendasi produk yang berpotensi disukai oleh seorang user. Beberapa contoh recommender system online adalah Amazon (www.amazon.com), FireFly (www.firefly.com), dan GroupLens (www.cs.umn.edu/Research/GroupLens). [9]

Jenis-jenis sistem rekomendasi yang sering dipakai yaitu:

1. Collaborative Filtering recommendation
Sistem rekomendasi yang menggunakan metode ini menggunakan data opini user, item-item yang sering

digunakan/dibeli oleh user lain, ataupun semacam rating yang diberikan dari user untuk mendapatkan rekomendasi. Jadi, metode ini menghasilkan rekomendasi untuk user target berdasarkan user-user lain yang sebelumnya memiliki minat yang sama dengan user tersebut.

2. Content-based recommendation

Berbeda dengan *collaborative filtering*, metode content-based filtering tidak menggunakan parameter semacam rating untuk menghasilkan rekomendasi. Melainkan menggunakan deskripsi dari profil pengguna, atau dari deskripsi suatu item-item yang pernah digunakan oleh user tersebut untuk menghasilkan suatu rekomendasi. [8]

III. PENERAPAN *PATTERN MATCHING* PADA APLIKASI BERITA KURIO DENGAN MEMANFAATKAN *MACHINE LEARNING* DAN SISTEM REKOMENDASI

A. Analisis *Pattern Matching* pada *Machine Learning* Aplikasi Kurio

Aplikasi Kurio menyediakan sebuah fitur pemilihan topik yang sesuai keinginan pengguna. Fitur pemilihan topik tersebut secara otomatis diakses ketika pengguna pertama kali menggunakan aplikasi. Selain itu juga pemilihan topik dapat di explore lagi nantinya atau dapat diganti. Tujuan dari adanya pemilihan topik ini adalah agar aplikasi mengetahui topik-topik apa saja yang diinginkan oleh pengguna. Hal tersebut sebagai acuan nantinya untuk memberikan rekomendasi pada pengguna (sistem rekomendasi). Pengaplikasian *machine learning* sendiri pada aplikasi ini yaitu aplikasi mendapatkan banyak atau kumpulan data dari berbagai portal berita. Berita-berita tersebut pastinya memiliki suatu pola-pola yaitu berdasarkan topiknya. Dari data dan pola-pola tersebut, maka *machine learning* dapat bekerja. Selain itu, pada aplikasi Kurio *machine learning* juga digunakan sebagai mesin penjaga akan berita yang hoax atau palsu. Jadi saat ada berita hoax, maka *machine learning* akan memfilter berita-berita tersebut dan ketika ditemukan bahwa berita itu hoax maka tidak akan ditampilkan di aplikasi.

Penggunaan *pattern matching* pada *machine learning* aplikasi Kurio yaitu ketika aplikasi Kurio mendapatkan data-data dari berbagai macam portal berita maka mesin akan mencari beberapa pola atau pattern. Contohnya mesin mendapatkan pattern string "politik". Maka pater tersebut akan dicari pada keseluruhan berita atau artikel dan ketika menghubungkan pattern dengan pattern lain yang cocok untuk dijadikan dalam suatu topik maka berita tersebut dimasukkan kedalam suatu topik misalnya topik "Politik". Jadi *machine learning* tidak hanya mengenali satu pattern saja tapi ada beberapa pattern yang nantinya jika dihubungkan akan membentuk suatu pola baru yang ternyata terdefiniskan dalam satu topik. Pencocokan pattern ini menggunakan algoritma string matching.

B. Analisis *Pattern Matching* pada Sistem Rekomendasi

Sistem rekomandasi ini untuk menguntungkan para pengguna karena mereka bisa secara langsung mendapatkan

berita yang benar-benar diinginkan tanpa harus memfilter-filter. Untuk kedua metode yang digunakan yaitu Collaborative Filtering recommendation dan Content-based recommendation, sistem rekomendasi menggunakan algoritma *pattern matching* untuk mendapatkan hasil rekomendasi.

Pada Collaborative Filtering recommendation, *pattern matching* diaplikasikan dengan pencocokan data berupa rating, opini, dan item-item user lain. Sistemnya sama dengan algoritma pencocokan string yaitu parameter yang digunakan untuk pencocokan dijadikan sebagai pattern atau pola sebagai pembanding untuk dibandingkan dengan pattern user lain. Sehingga ketika ditemukan pattern yang sama maka hasil temuan atau kecocokan tersebut dijadikan sebagai rekomendasi untuk user.

Pada Content-based recommendation, *pattern matching* diaplikasikan dengan pencocokan data dari deskripsi dan profil atau item-item dari user sendiri. Untuk algoritma pencocokan stringnya sama dengan Collaborative Filtering recommendation hanya saja untuk pembandingnya tidak kepada user lain melainkan parameter pattern dari user berupa profil atau item-item yang pernah digunakan dicocokkan dengan data hasil *machine learning*. Sehingga ditemukan rekomendasi yang cocok untuk user tersebut.

C. Analisis *Pattern Matching* pada Fitur "Search" Aplikasi Kurio dengan Memanfaatkan Hasil dari *Machine Learning* dan Sistem Rekomendasi

Pentingnya *machine learning* dan sistem rekomendasi ini juga digunakan pada fitur "Search". Meskipun fitur "Search" algoritma pencocokan stringnya akan mencari data pattern string yang dicari dari semua portal berita baik yang sudah direkomendasikan atau belum namun dengan memanfaatkan *machine learning* dan sistem rekomendasi dari aplikasi, berita yang ditampilkan tetap sesuai dengan keinginan pengguna. Untuk pencocokan pattern pada fitur "Search" nya sendiri dapat menggunakan algoritma Brute Force, Knuth-Morris-Pratt, maupun Boyer Moore.

Contoh

Pada kolom "Search",

Pattern : Jokowi

Maka pattern tersebut akan dicocokkan untuk semua berita yang ada di aplikasi. Dan ketika ditemukan pattern yang sama maka semua berita yang mengandung pattern tersebut ditampilkan dalam aplikasi. Tampilan pattern pada aplikasi yaitu ketika pattern pertama kali ditemukan.

D. Pemanfaatan Teknologi *Machine Learning* dan Mesin Rekomendasi pada Aplikasi Berita Kurio

Aplikasi berita Kurio menampilkan berita-berita sesuai topik yang telah dipilih oleh pengguna. Dalam penentuan berita mana yang ditampilkan ke pengguna tersebut, aplikasi Kurio memanfaatkan teknologi *machine learning* dan mesin rekomendasi. Dari hal tersebut maka pengguna aplikasi mendapatkan berita yang hanya benar-benar diinginkan oleh pengguna sehingga memduhkan pengguna untuk membaca berita sesuai ketertarikannya. Selain itu, *machine learning*

juga menyeleksi berita mana yang hoax mana yang fakta sehingga berita yang didapatkan pengguna benar-benar faktual.

IV. KESIMPULAN

Salah satu pengaplikasian algoritma *pattern matching* yaitu pada aplikasi berita Kurio. Aplikasi berita Kurio memanfaatkan algoritma *pattern matching* untuk mendapatkan berita yang sesuai dengan keinginan pembaca dengan memanfaatkan teknologi *machine learning* dan mesin rekomendasi. Pada aplikasi Kurio juga terdapat tema atau topik yang dapat dipilih oleh pengguna untuk menentukan ketertarikan pengguna di bidang apa dalam memilih berita. Aplikasi berita Kurio memanfaatkan teknologi *machine learning* untuk mengelompokkan berita-berita yang sesuai dengan topik tertentu dalam satu kategori. Hal tersebut dapat dilaksanakan karena adanya data berita dan penerapan algoritma *pattern matching*. Untuk berita yang sesuai keinginan pembaca didapatkan dengan memanfaatkan teknologi mesin rekomendasi. Mesin rekomendasi menerapkan algoritma *pattern matching* untuk mendapatkan berita yang sesuai atau rekomendasi untuk pengguna.

V. UCAPAN TERIMAKASIH

Pertama, penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa karena berkat rahmat dan ridhaNya sehingga penulis dapat menyelesaikan makalah ini. Penulis juga mengucapkan terima kasih kepada orang tua penulis yang selalu membantu secara moral, material, dan doa. Penulis turut mengucapkan terima kasih kepada Ibu Masayu, selaku dosen dari mata kuliah Strategi Algoritma atas bimbingan dan jasa beliau yang selama ini telah mengajar dan memberikan banyak ilmu kepada penulis. Tak lupa juga, penulis mengucapkan terima kasih kepada teman-teman seperjuangan yang telah memberikan masukan kepada penulis.

DAFTAR PUSTAKA

- [1] Munir, Rinaldi, "Diktat Kuliah IF2211 *Strategi Algoritma*", Informatika, Bandung: 2007.
- [2] http://informatika.stei.itb.ac.id/~rinaldi_munir/Stmik/2006-2007/Makalah_2007/MakalahSTMIK2007-012.pdf, diakses 18 Mei 2017 pukul 01.42 WIB.
- [3] <http://lelakibugis.net/kurio-aplikasi-berita-indonesia/>, diakses 18 Mei 2017 pukul 02.03 WIB.
- [4] <http://app-nesia.com/articles/kurio-app-berita-indonesia>, diakses 18 Mei 2017 pukul 02.03 WIB.
- [5] <https://kurio.co.id/>, diakses 18 Mei 2017 pukul 02.55 WIB.
- [6] <https://www.codepolitan.com/mengenal-teknologi-machine-learning-pembelajaran-mesin>, diakses 18 Mei 2017 pukul 07.25 WIB.
- [7] <http://embeddednesia.com/v1/?p=1641>, diakses 18 Mei 2017 pukul 08.00 WIB.
- [8] Lior Rokach Francesco Ricci, *Recommender System Handbook*. London: Springer, 2011.
- [9] Weiyang Lin, "Association Rule Mining for Collaborative Recommender System," 2000.

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, 19 Mei 2017



Candra Hesén Parera / 13515019