

Penerapan Algoritma Not So Naive Pada Aplikasi Kumpulan Cerita Motivasi

Winda Triana

Program Studi Teknik Informatika, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Budi Darma,
Jalan Sisingamangaraja No. 338, Medan, Sumatera Utara, Indonesia
Email: windatriana349@gmail.com

Abstrak- Manusia memiliki kelemahan dan juga memiliki kelebihan yang tercipta berbagai macam corak dan ragamnya. Dalam posisi hilang semangat hidup kita membutuhkan motivasi untuk membantu kita berdiri lagi. Dengan motivasi, seorang individu bisa bertindak dengan tepat untuk mencapai tujuannya. Tidak hanya itu, keterbatasan untuk menemukan suatu cerita motivasi diinternet memiliki beberapa masalah diantaranya, susah menemukan sesuai dengan yang dibutuhkan, membutuhkan waktu yang lama dan gangguan jaringan. Dengan demikian, perlunya sebuah informasi yang praktis yang dapat diakses kapan saja dan dimana saja oleh semua masyarakat. Hal itu guna untuk mempermudah dan mempercepat seseorang untuk mencari motivasi yang sesuai dengan yang dibutuhkan untuk mencapai tujuan. Sehingga penulis membuat aplikasi yang dapat membantu individu lain menemukan motivasi yang tepat berdasarkan apa yang mereka cari dan mendorong mereka untuk memperbaiki dan menemukan jati dirinya serta dapat membantu meminimalkan biaya dalam pembelian buku yang membahas tentang cerita motivasi. Untuk melakukan proses pencarian judul pada aplikasi kumpulan cerita motivasi membutuhkan algoritma not so naive. Dengan metode ini dapat mempermudah dalam pencarian judul yang diinginkan pada aplikasi kumpulan cerita motivasi.

Kata Kunci: Cerita Motivasi, Algoritma Not So Naive, Pattern, Android

Abstract- Humans have weaknesses and also have advantages that are created by various types and varieties. In a position of losing our enthusiasm for life we need motivation to help us stand up again. With motivation, an individual can act appropriately to achieve his goals. Not only that, the limitations of finding a motivational story on the internet have several problems including difficulty finding what is needed, it takes a long time and network disruptions. Thus, the need for practical information that can be accessed anytime and anywhere by all people. This is in order to make it easier and faster for someone to find the appropriate motivation needed to achieve goals. So that the authors make applications that can help other individuals find the right motivation based on what they are looking for and encourage them to improve and find their identity and can help minimize costs in purchasing books that discuss motivational stories. To perform the title search process in a motivational story collection application requires a not so naive algorithm. With this method, it can make it easier to find the desired title in the motivational story collection application.

Keywords: Motivation Stories, Not So Naive Algorithm, Pattern, Android

1. PENDAHULUAN

Perilaku manusia tidak terlepas dari perilaku dalam bermasyarakat yang sering disebut dengan perilaku sosial, perilaku sosial yang secara khusus ditujukan kepada orang lain untuk memberi atau menerima yang berkaitan satu sama lain secara sistematis, perilaku tersebut merupakan salah satu kebutuhan manusia yang bersifat fundamental dimana manusia tidak dapat hidup sendiri. Perilaku manusia sebenarnya hanyalah cermin motivasi dasar mereka, agar perilaku manusia sesuai dengan tujuan hidup mereka. Beberapa orang memandang motivasi sebagai sesuatu yang terkandung pada perseorangan dan memandang bahwa motivasi timbul dari sumber diluar perseorangan. Manusia juga memiliki kelemahan dan juga memiliki kelebihan yang tercipta berbagai macam corak dan ragamnya. Dalam kehidupan manusia pasti pernah merasakan keputusasaan dimana kita terjatuh dalam posisi yang sangat terpuruk, saat semua semangat dan mimpi telah lenyap seketika. Dalam posisi hilang semangat hidup kita membutuhkan motivasi untuk membantu kita berdiri lagi. Manusia membutuhkan dorongan positif yang berasal dari luar dirinya sehingga dia bisa menyadari keberadaan dan segala potensi yang ada pada dirinya. Dengan motivasi, seorang individu bisa bertindak dengan tepat untuk mencapai tujuannya. Tidak hanya itu, keterbatasan untuk menemukan suatu cerita motivasi diinternet memiliki beberapa masalah diantaranya, susah menemukan sesuai dengan yang dibutuhkan, membutuhkan waktu yang lama dan gangguan jaringan

Dengan demikian, perlunya sebuah informasi yang praktis yang dapat diakses kapan saja dan dimana saja oleh semua masyarakat. Hal itu guna untuk mempermudah dan mempercepat seseorang untuk mencari motivasi yang sesuai dengan yang dibutuhkan untuk mencapai tujuan. Sehingga penulis membuat aplikasi yang dapat membantu individu lain menemukan motivasi yang tepat berdasarkan apa yang mereka cari dan mendorong mereka untuk memperbaiki dan menemukan jati dirinya serta dapat membantu meminimalkan biaya dalam pembelian buku yang membahas tentang cerita motivasi. Berdasarkan penelitian terdahulu yang dilakukan oleh D. Cantone dan S.Faro, dalam *Proceeding of the International Conference on with Algorithms*, yang berjudul *Searching For Substring With Contant Extra Space Complexity*. Menyimpulkan bahwa algoritma not so naive merupakan variasi sederhana dari algoritma naive. Cara kerjanya adalah dengan memindai teks dan pola dari kiri ke kanan. Pada fase pencarian algoritma not so naive akan mencocokkan 2 karakter sekaligus, jika di akhir pencocokan pada kedua karakter tersebut mengalami kecocokan maka pola akan digeser sebanyak 2 posisi ke kanan, jika tidak maka pola akan digeser 1 posisi ke kanan. Namun bisa juga sebaliknya, tergantung hasil dari fase preprocessing yang telah ditetapkan terlebih dahulu. Berbeda dengan algoritma naive yang dimana pergeserannya tetaplah sebanyak 1 posis[1].

Pada penelitian sebelumnya telah dilakukan sebuah penelitian tentang penerapan algoritma not so naive pada pencarian jurnal berbasis mobile. Dimana dalam penelitian ini membahas bahwa algoritma not so naive tidak dapat

mencari 1 huruf saja. Karena pada saat pra-proses algoritma ini membutuhkan minimal 2 huruf awal yang sesuai untuk menentukan nilai variabel k dan ell pada proses pencarian [2]. Algoritma *string matching* merupakan suatu algoritma pencocokan *string* yang sifatnya mencari sebuah *string* yang terdiri dari beberapa *pattern* (karakter) dalam jumlah teks yang besar. Untuk aplikasi kumpulan cerita motivasi dirancang menggunakan algoritma *Not So Naïve*. Algoritma *Not So Naïve* adalah algoritma yang digunakan untuk melakukan proses pencocokan *string*. Algoritma ini merupakan variasi simpel dari algoritma *naïve* namun menjadi lebih efisien untuk beberapa kasus.

Untuk mengatasi masalah pencarian judul cerita motivasi, maka digunakan algoritma *string matching* salah satunya yaitu algoritma *Not So Naïve*. Dengan menggunakan algoritma *Not So Naïve*, pencarian judul cerita motivasi akan dilakukan dengan cara memasukan judul cerita motivasi pada kotak pencarian, jika judul cerita motivasi ditemukan maka akan muncul isi cerita motivasi tersebut

2. METODOLOGI PENELITIAN

2.1 String Matching

String Matching adalah proses pencarian semua kemunculan *query* yang selanjutnya disebut *pattern* ke dalam *string* yang lebih panjang (teks). *Pattern* dilambangkan dengan $x=x[0..m-1]$ dan panjangnya adalah m . Teks dilambangkan dengan $y=y[0..n-1]$ dan panjangnya adalah n . Kedua *string* terdiri dari sekumpulan karakter yang disebut alfabet yang dilambangkan dengan Σ (sigma) dan mempunyai ukuran σ (tao). *String matching* dibagi menjadi dua, yakni *exact matching* dan *heuristic* atau *statistical matching* [5].

2.2 Algoritma Not So Naïve

Algoritma *Not So Naïve* merupakan variasi turunan dari algoritma *naïve* atau yang sering disebut dengan algoritma *Brute Force*. Cara kerja algoritma ini adalah dengan memiliki fase pencarian mengecek teks dan pola dari kiri ke kanan. Lalu, algoritma *Not So Naïve* akan mengidentifikasi terlebih dahulu dua kasus yang dimana di setiap akhir fase pencocokan pergeseran dapat dilakukan sebanyak 2 posisi ke kanan, tidak seperti algoritma *naïve* yang dimana pergeseran tetaplah sebanyak 1 posisi ke kanan. Misal di asumsikan bahwa $P[0] \neq P[1]$. Jika $P[0] = T[s]$ dan $P[1] = T[s+1]$, maka di akhir fase pencocokan pergeseran s bisa dilakukan sebanyak 2 posisi, karena $P[0] \neq P[1] = T[s+1]$. Dan jika $P[0] = P[1]$ dan $P[0] = P[s]$ tapi $P[1] \neq T[s+1]$, maka sekali lagi pergeseran s dapat dilakukan sebanyak 2 posisi dimana P adalah *pattern*, T adalah teks dan S adalah nilai posisi [6].

2.3 Aplikasi

Aplikasi bisa dikatakan suatu perangkat lunak yang siap pakai dengan menjalankan instruksi-instruksi dari pengguna (user), aplikasi banyak diciptakan untuk membantu berbagai keperluan, seperti contoh: membuat laporan, percetakan dan lain-lain aplikasi merupakan sistem yang dirancang dan disusun sedemikian rupa untuk menghasilkan informasi yang terpadu dengan menggunakan sarana komputer sebagai sarana penunjangnya [7].

2.4 Kumpulan Cerita Motivasi

Kumpulan adalah sebuah homonim karena arti-artinya memiliki ejaan dan pelafalan yang sama tetapi maknanya berbeda. Kumpulan memiliki arti dalam kelas nomina atau kata benda sehingga kumpulan dapat menyatakan nama dari seseorang, tempat, atau semua benda dan segala yang dibendakan [8]. Cerita motivasi adalah dorongan untuk mendapatkan sesuatu tujuan dengan lebih giat, misalnya kita mempunyai motivasi atau dorongan untuk memiliki mobil baru, tentu kita akan melakukan segala usaha dalam pekerjaan agar dapat terkumpul uang untuk membeli mobil tersebut dan itulah yang disebut dengan motivasi. Namun pada biasanya kita lebih sering menggunakan kata motivasi untuk mendapatkan hidup yang lebih baik yaitu dengan mencapai tujuan tujuan hidup kita yang lebih luas. Sering kita temui orang yang sudah pupus harapannya dikarenakan masalah yang mereka alami, dan saat kita menghadapi orang yang sedang dalam posisi kehilangan arah dan mulai pupus harapan, disaat itulah kita harus bisa mendorong dan mengarahkannya ke jalan yang lebih baik (memotivasi) [9].

3. HASIL DAN PEMBAHASAN

Algoritma *Not So Naïve* dapat diterapkan dalam pencarian cerita motivasi berbasis android. Hal itu guna untuk mempermudah pencarian karakter dalam pencocokan *pattern* berdasarkan dengan teks. Proses pencarian cerita motivasi dilakukan dengan memasukan judul cerita motivasi pada kolom pencarian yang telah tersedia pada aplikasi tersebut. Jika cerita motivasi yang dicari pada kolom pencarian tersebut ditemukan, maka akan ditampilkan isi dari cerita motivasi tersebut. Cerita motivasi dikumpulkan dengan cara mencari sebuah referensi yang ada pada internet kemudian dikumpulkan dan disimpan pada database.

Pencarian cerita motivasi ini dirancang dengan menggunakan algoritma *Not So Naïve*, dimana algoritma ini sebagai solusi dalam pencarian cerita motivasi, karena banyaknya macam-macam cerita motivasi sehingga sangat sulit untuk menampilkan penyajian cerita motivasi itu menurut dengan abjadnya. Tidak hanya itu, dengan terbatasnya ukuran layar smartphone yang dimiliki, maka sangat diperlukan *string matching* sebagai suatu proses dalam pencarian. Dimana

dalam algoritma string matching penyajian cerita motivasi akan lebih sedikit karena adanya proses pengecekan berdasarkan pattern.

Algoritma *Not So Naive* dapat diterapkan dalam pencarian cerita motivasi berbasis android. Hal itu guna untuk mempermudah pencarian karakter dalam pencocokan *pattern* berdasarkan dengan teks. Proses pencarian cerita motivasi dilakukan dengan memasukan judul cerita motivasi pada kolom pencarian yang telah tersedia pada aplikasi tersebut. Jika cerita motivasi yang dicari pada kolom pencarian tersebut ditemukan, maka akan ditampilkan isi dari cerita motivasi tersebut.

Penggunaan algoritma *Not So Naive* untuk pencarian *pattern* dalam teks, seperti pada contoh dibawah ini:

Tabel 1. Kumpulan Data Cerita Motivasi

Judul Cerita Motivasi
Nilai Kehidupan
Taman Kehidupan
Jadilah Pelita
Puncak Kesuksesan
Hidup Harmonis

Pada contoh kasus ini cerita yang akan dicari adalah cerita Nilai Kehidupan, pencarian dilakukan berdasarkan judul cerita, misalkan dalam pencarian digunakan kata/pattern “hidup”, maka proses pencariannya adalah sebagai berikut:

Pattern : Hidup
Teks : Nilai Kehidupan

Tabel 2. Fase Preprocessing

Index	0	1	2	3	4
Pattern	H	I	D	U	P

Berdasarkan tabel 2., karen $P[0] = H$ dan $P[1] = I$ pada pattern “HIDUP” tidak mengalami kecocokkan maka nilai $k = 1$ dan nilai $ell = 2$. Dimana kedua variabel tersebut akan digunakan untuk nilai pergeseran pada fase pencarian.

Fase Pencarian 1

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Teks	N	I	L	A	I	-	K	E	H	I	D	U	P	A	N
Pattern	H	I	D	U	P										

Pada fase pencarian pertama, perbandingan karakter pada index ke 0 mengalami ketidakcocokkan namun index ke 1 mengalami kecocokkan ($P[0] \neq T[s]$ dan $P[1] = T[s+1]$). Walaupun pada index ke 0 mengalami ketidakcocokkan, namun itu tidak berpengaruh jika pada index selanjutnya mengalami kecocokkan. Maka akan dilakukan percobaan selanjutnya dengan posisi *pattern* digeser sebanyak 1 posisi sesuai dengan nilai variabel k .

Fase Pencarian 2

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Teks	N	I	L	A	I	-	K	E	H	I	D	U	P	A	N
Pattern		H	I	D	U	P									

Pada fase pencarian kedua, perbandingan karakter pada index ke 1 dan 2 mengalami ketidakcocokkan ($P[0] \neq T[s]$ dan $P[1] \neq T[s+1]$). Maka akan dilakukan percobaan selanjutnya dengan posisi *pattern* digeser sebanyak 1 posisi sesuai dengan nilai variabel k .

Fase Pencarian 3

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Teks	N	I	L	A	I	-	K	E	H	I	D	U	P	A	N
Pattern			H	I	D	U	P								

Pada fase pencarian ketiga, perbandingan karakter pada index ke 2 dan 3 mengalami ketidakcocokkan ($P[0] \neq T[s]$ dan $P[1] \neq T[s+1]$). Maka akan dilakukan percobaan selanjutnya dengan posisi *pattern* digeser sebanyak 1 posisi sesuai dengan nilai variabel k .

Fase Pencarian 4

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Teks	N	I	L	A	I	-	K	E	H	I	D	U	P	A	N
Pattern				H	I	D	U	P							

Pada fase pencarian keempat, perbandingan karakter pada index ke 3 mengalami ketidakcocokkan namun index ke 4 mengalami kecocokkan ($P[0] \neq T[s]$ dan $P[1] = T[s+1]$). Walaupun pada index ke 3 mengalami ketidakcocokkan, namun itu tidak berpengaruh jika pada index selanjutnya mengalami kecocokkan.. Maka akan dilakukan percobaan selanjutnya dengan posisi *pattern* digeser sebanyak 1 posisi sesuai dengan nilai variabel k .

Fase Pencarian 5

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Teks	N	I	L	A	I	-	K	E	H	I	D	U	P	A	N
Pattern					H	I	D	U	P						

Pada fase pencarian kelima, perbandingan karakter pada index ke 4 dan 5 mengalami ketidakcocokan ($P[0] \neq T[s]$ dan $P[1] \neq T[s+1]$). Maka akan dilakukan percobaan selanjutnya dengan posisi *pattern* digeser sebanyak 1 posisi sesuai dengan nilai variabel k.

Fase Pencarian 6

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Teks	N	I	L	A	I	-	K	E	H	I	D	U	P	A	N
Pattern						H	I	D	U	P					

Pada fase pencarian keenam, perbandingan karakter pada index ke 5 dan 6 mengalami ketidakcocokan ($P[0] \neq T[s]$ dan $P[1] \neq T[s+1]$). Maka akan dilakukan percobaan selanjutnya dengan posisi *pattern* digeser sebanyak 1 posisi sesuai dengan nilai variabel k.

Fase Pencarian 7

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Teks	N	I	L	A	I	-	K	E	H	I	D	U	P	A	N
Pattern							H	I	D	U	P				

Pada fase pencarian ketujuh, perbandingan karakter pada index ke 6 dan 7 mengalami ketidakcocokan ($P[0] \neq T[s]$ dan $P[1] \neq T[s+1]$). Maka akan dilakukan percobaan selanjutnya dengan posisi *pattern* digeser sebanyak 1 posisi sesuai dengan nilai variabel k.

Fase Pencarian 8

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Teks	N	I	L	A	I	-	K	E	H	I	D	U	P	A	N
Pattern								H	I	D	U	P			

Pada fase pencarian kedelapan, perbandingan karakter pada index ke 7 dan 8 mengalami ketidakcocokan ($P[0] \neq T[s]$ dan $P[1] \neq T[s+1]$). Maka akan dilakukan percobaan selanjutnya dengan posisi *pattern* digeser sebanyak 1 posisi sesuai dengan nilai variabel k.

Fase Pencarian 9

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Teks	N	I	L	A	I	-	K	E	H	I	D	U	P	A	N
Pattern									H	I	D	U	P		

Pada fase pencarian kesembilan, perbandingan karakter pada index ke 8 dan 9 mengalami kecocokkan ($P[0] = T[s]$ dan $P[1] = T[s+1]$). Maka dilanjutkan pada index berikutnya, dan semua mengalami kecocokkan, oleh sebab itu teks akan dikeluarkan. Namun, algoritma *Not So Naive* akan melakukan percobaan terus sampai sisa teks lebih kecil dari pada *pattern*. Untuk percobaan selanjutnya, posisi *pattern* akan digeser sebanyak 2 posisi sesuai dengan nilai variabel *ell*.

Fase Pencarian 10

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Teks	N	I	L	A	I	-	K	E	H	I	D	U	P	A	N
Pattern											H	I	D	U	P

Pada fase pencarian kesepuluh, perbandingan karakter pada index ke 10 dan 11 mengalami ketidakcocokan ($P[0] \neq T[s]$ dan $P[1] \neq T[s+1]$). Maka akan dilakukan percobaan selanjutnya dengan posisi *pattern* digeser sebanyak 1 posisi sesuai dengan nilai variabel k. Namun karena sisa teks telah lebih kecil daripada *pattern* maka fase pencarian berhenti disini.

4. KESIMPULAN

Berdasarkan hasil kesimpulan dari hasil dan pembahasan pada penelitian ini dimana penerapan algoritma not so naive dapat diterapkan dalam pencarian judul cerita motivasi. Hal itu guna untuk mempermudah pencarian karakter dalam pencocokan pattern berdasarkan dengan teks. Serta Algoritma not so naive dapat diimplementasikan sebagai solusi dalam pencarian cerita motivasi. Karena banyaknya judul cerita motivasi yang ada di internet dan berhubung keterbatasan layar dari masing-masing smartphone, maka diperlukan string matching dalam proses pencarian judul cerita motivasi Dimana dalam algoritma ini penyajian cerita motivasi menjadi lebih sedikit karena adanya proses pengecekan berdasarkan pattern.

REFERENCES

- [1] D. Cantone dan S.Faro, "Proceeding of The International Conference on FUN with Algorithm".
- [2] M. Alfarizi and S. R. Saragih, "Penerapan Algoritma Not So Naive Pada Pencarian Jurnal Berbasis Mobile," pp. 490–494, 2019.
- [3] Nurdin Usman, *Konteks Implementasi Berbasis Kurikulum*. Jakarta: PT.Raja Grafindo Persada, 2002.
- [4] E. Rinaldi Munir, *Algoritma dan Pemrograman dalam Pascal dan C, Revisi Ed.* Bandung: Informatika Bandung, 2011.
- [5] E. Yeni Anistyasari, dan Rahmi Fitri Riyanto Sarno, *Semantic Search*. Yogyakarta: Andi Offset, 2012.
- [6] G. I. Webb, J. R. Boughton, and Z. Wang, "Not so naive Bayes: Aggregating one-dependence estimators," *Mach. Learn.*, 2005.
- [7] J. Hartono, *Pengenalan Komputer Aplikasi*. 2004.
- [8] T. P. KKBI, *Kumpulan*. 1997.
- [9] F. V. Sheila, *CARA MENDORONG, MENGARAHKAN DAN MEMOTIVASI ORANG LAIN*. 2014