

String Matching dengan Knuth-Morris Pratt pada Aplikasi Pengecekan Kemiripan Judul Project PAK

String Matching with Knuth-Morris Pratt on Similarity Checking Application of PAK Project Title

Santi^{a,1,*} dan Komang Aryasa^{a,2}

^aTeknik Informatika, Universitas Dipa Makassar, Makassar, Indonesia

¹santi@undipa.ac.id; ²komang.aryasa@undipa.ac.id

*corresponding author

Informasi Artikel	ABSTRAK
Diserahkan : 24 Februari 2025 Diterima : 12 Maret 2025 Direvisi : 20 Maret 2025 Diterbitkan : 20 Maret 2025 Kata Kunci: Pencocokan karakter Algoritma KMP Judul proyek Aplikasi	Pembuatan Aplikasi Konsentrasi (PAK) merupakan salah satu mata kuliah wajib pada Universitas Dipa Makassar. Pada mata kuliah ini, setiap mahasiswa diwajibkan membuat satu <i>project</i>/aplikasi yang tidak boleh sama dengan mahasiswa lainnya. Penilaian judul <i>project</i> dilakukan secara langsung oleh dosen pengampu mata kuliah. Hal ini rentan terhadap terjadinya kesalahan atau pun kekeliruan dan bahkan bisa menghabiskan banyak waktu karena dosen masih berpikir dan berusaha mengingat judul-judul apa saja yang sudah pernah dipamerkan pada semester-semester sebelumnya. Selain dosen, mahasiswa juga terkadang kebingungan mengenai judul yang akan mereka ajukan, sehingga banyak diantara mereka yang mengajukan judul lebih dari satu untuk menghindari penolakan, terutama penolakan karena judul tersebut telah banyak diaplikasikan oleh mahasiswa-mahasiswa sebelumnya. Oleh karena itu, penelitian ini bertujuan untuk merancang sebuah aplikasi berbasis <i>web</i> yang dapat mengecek kesamaan/kemiripan judul <i>project</i> mahasiswa dengan menerapkan algoritma Knuth- Morris Pratt (KMP). Algoritma ini melakukan pencocokan karakter (<i>string matching</i>) yang ada pada judul yang diajukan dengan judul-judul yang sudah ada di dalam <i>database</i> sistem. Hasil pengujian menunjukkan bahwa penerapan algoritma KMP dapat dengan cepat dan akurat mendeteksi tingkat kemiripan judul <i>Project</i> sehingga, baik dosen maupun mahasiswa, dapat mengevaluasi sebuah judul <i>Project</i> yang akan dikerjakan.
Keywords: String matching KMP algorithm Project title Application	ABSTRACT The Creation of Application Concentration Course is one of the mandatory courses at Dipa University Makassar. In this course, each student is required to create one project/application that cannot be the same as other students. The assessment of the project title was carried out directly by the lecturer in charge of the course. This was prone to errors or mistakes and can even take a lot of time because the lecturer was still thinking and trying to remember what titles have been exhibited in previous semesters. In addition to lecturers, students were also sometimes confused about the title they will submit, so many of them submitted more than one title to avoid rejection, especially rejection because of the title has been widely applied by previous students. Therefore, this study aimed to design a web-based application that can check the similarity of students' project titles by applying the Knuth-Morris Pratt (KMP) algorithm. This algorithm matches the strings/characters in the submitted title with the titles that already exist in the system database. The test results show that the application of the KMP algorithm can quickly and accurately detect the level of similarity of Project titles, so that both lecturers and students can evaluate a Project title that will be done.

This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



I. Pendahuluan

Pembuatan Aplikasi Konsentrasi (PAK) merupakan salah satu mata kuliah (MK) wajib bagi mahasiswa Program Studi (Prodi) Teknik Informatika (TI) dan Sistem Informasi (SI) pada Universitas Dipa Makassar. Persyaratan lulus pada MK ini adalah mahasiswa harus membuat sebuah program aplikasi (*project*) yang akan dipamerkan pada akhir semester. Sebelum pembuatan aplikasi *project*, mahasiswa mengajukan judul terlebih dahulu ke dosen pengampu MK tersebut. Judul harus berbeda di antara mahasiswa. Untuk penilaian atau

pengecekan sebuah judul dilakukan secara langsung dengan dosennya. Hal ini tentunya rentan terhadap terjadinya kesalahan atau pun kekeliruan dan bahkan bisa menghabiskan banyak waktu karena dosen masih berpikir dan berusaha mengingat judul-judul *project* PAK apa saja yang telah dipamerkan pada semester-semester sebelumnya. Selain itu, mahasiswa juga terkadang kebingungan mengenai judul yang akan mereka ajukan, sehingga banyak di antara mereka yang mengajukan judul lebih dari satu untuk menghindari penolakan, terutama penolakan karena judul tersebut sudah banyak diaplikasikan oleh mahasiswa-mahasiswa sebelumnya.

String matching merupakan sebuah algoritma yang digunakan dalam pencocokan suatu pola kata tertentu terhadap suatu kalimat atau teks panjang [1], seperti pada pengecekan plagiarisme [2] dan kamus penerjemah Bahasa Indonesia-Korea [3] dan Bahasa Aceh-Bahasa Indonesia [4]. *Knuth-Morris Pratt* (KMP) merupakan sebuah algoritma *string matching* yang melakukan perbandingan antara karakter *text* dan karakter *pattern* [5]. Algoritma KMP ini akan mencocokkan pola atau susunan kata yang akan dicari dari kiri ke kanan pada awal teks dan kemudian menggeser susunan kata sampai susunan kata tersebut berada di ujung teks [6]. Dengan memanfaatkan algoritma KMP ini, maka proses pencarian dapat dilakukan dengan mudah [7] dan dalam waktu yang begitu singkat, antara 61 – 80% pada pencarian istilah SOP LPM [8] misalnya. Kurniawan dan Indrianti telah membuktikan bahwa rata-rata waktu yang dibutuhkan algoritma KMP untuk melakukan pencarian data buku dalam sistem informasi perpustakaan di SMPN 2 Depok adalah 0.014 detik, waktu tercepat adalah 0.012 detik dan waktu terlamanya adalah 0.019 detik [9]. Peneliti lain, juga menemukan bahwa, dengan sekitar 0.505 detik waktu yang dibutuhkan untuk melakukan pencarian di web jurnal dengan menggunakan 5 kata kunci [10]. Algoritma KMP ini juga lebih cepat 10,41% dalam proses pencarian berkas dibanding dengan pencarian biasa [11].

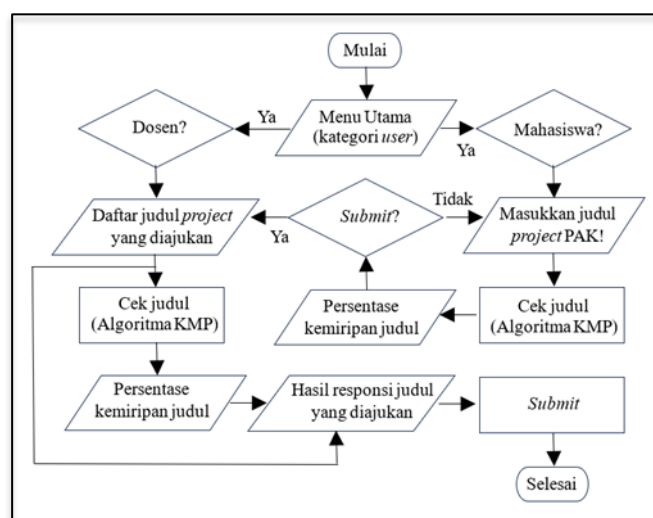
Selain dapat diandalkan karena kecepatan waktu pencariannya, algoritma KMP juga diketahui dapat menghitung tingkat persentase kemiripan antara dua atau lebih barisan teks atau kalimat. Salah satu contohnya adalah pada sistem pendeteksi tingkat kemiripan dua atau lebih judul skripsi [12] yang menemukan bahwa hasil dari proses *string matching* antara karakter pada teks dengan karakter pada *pattern* dapat menampilkan jumlah kemiripan kata dan persentasenya (hasil persentase tidak dipublish) [13].

Berdasarkan pokok permasalahan dan temuan peneliti terkait yang telah dikemukakan di atas, maka tujuan utama dari penelitian ini adalah merancang sebuah aplikasi berbasis web yang dapat membantu dosen dan mahasiswa dalam hal pengecekan kemiripan judul *project* PAK sehingga proses pengajuan judul dapat berjalan sesuai dengan Rencana Pembelajaran Semester (RPS).

II. Metode

Penelitian ini dilaksanakan di kampus Universitas Dipa Makassar. Adapun bahan penelitian berupa data judul *project* yang diajukan oleh mahasiswa, yang diuji persentase kemiripannya dengan judul-judul *project* PAK yang telah mengikuti pameran aplikasi (mulai dari semester ganjil 2019 sampai dengan semester ganjil 2023; diambil di Prodi TI dan SI).

Untuk memudahkan perancangan program aplikasi pengecekan kemiripan judul ini (secara spesifik pada proses penerapan algoritma KMP), penulis terlebih merancang *flowchart* sistem seperti yang terlihat pada Gambar 1. *Flowchart* merupakan representasi grafis atau skematis dari suatu proses atau algoritma dengan menggunakan beberapa simbol yang saling terhubung melalui sebuah garis [14]. *Flowchart* memiliki keuntungan besar karena dapat mengakomodasi berbagai konsep bahasa pemrograman dalam kerangka kerja yang sama [15].



Gambar 1. *Flowchart* Aplikasi

Gambar 1 memperlihatkan bahwa, *user* atau pengguna dari aplikasi (khususnya *interface* yang menjadi wadah penerapan algoritma KMP) adalah dosen dan mahasiswa.

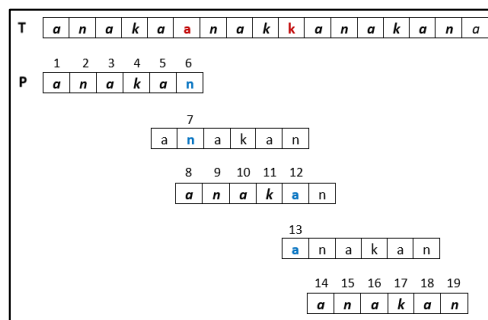
- *User* dosen memiliki hak akses untuk melihat daftar judul-judul project PAK yang diajukan oleh mahasiswa. Apabila sebuah judul dianggap layak untuk dikerjakan, maka judul tersebut dapat langsung disetujui melalui pengisian *form* pada *interface* hasil responsi judul kemudian mengirimkannya (*submit*) kembali kepada mahasiswa sebagai *feedback*. Sebaliknya, ketika dosen merasa belum yakin, maka judul tersebut dapat dicek persamaan atau persentase tingkat kemiripannya dengan judul-judul lain.
- *User* mahasiswa memiliki hak akses dalam pemasukan judul. Namun, sebelum mengajukan sebuah judul, mahasiswa harus mengecek kesamaan/kemiripan judul tersebut dengan judul-judul yang sudah ada. Dari tingkat persentase kemiripan judul, mahasiswa dapat memutuskan apakah judul tersebut layak diajukan atau memasukkan judul lain.

Penerapan algoitma KMP dalam pengecekan judul *project* PAK memiliki tahapan proses yang dimulai dengan pencocokan karakter *pattern* dengan karakter pada awal teks (*text*), dari kiri ke kanan. Pecocokan ini dilakukan per karakter *pattern* dengan per karakter *text* yang bersesuaian. Jika terjadi ketidakcocokan atau pun kecocokan antara karakter *pattern* dan karakter *text*, maka pencocokan berhenti. Jika karakter tidak cocok, maka indeks *text* yang diperiksa bergeser satu karakter, dan pemeriksaan diulang kembali sampai *pattern* berada di ujung *text*. Namun, sebelum proses pencocokan, terlebih dahulu mencari nilai dari fungsi pinggiran *F* yang akan digunakan untuk melakukan pergeseran ketika terdapat karakter yang tidak cocok. Sebagai contoh, perbandingan antara karakter *pattern* *P* “anakan” dengan karakter *text* *T* “anakaanakana”, memiliki nilai fungsi pinggiran seperti yang terlihat pada Tabel 1.

Tabel 1. Fungsi Pinggiran KMP “anakan”

J	0	1	2	3	4	5
P(J)	a	n	a	k	a	n
F(J)	0	0	1	0	1	2

Setelah mendapatkan nilai fungsi pinggiran untuk setiap karakter *pattern* *P*, maka langkah berikutnya adalah pencocokan karakter dengan algoritma KMP seperti yang terlihat pada Gambar 2.



Gambar 2. Contoh Proses Pencocokan Karakter dengan Algoritma KMP

Gambar 2 memperlihatkan proses pencocokan karakter *text* *T*= “anakaanakana” dengan karakter masukan, *pattern* *P*= “anakan” menggunakan fungsi pinggiran *F* pada Tabel 1.

III. Hasil dan Pembahasan

A. User Interface

Desain *user interface* aplikasi menggunakan *software* Visual Studio Code 1.84.2 dengan Bahasa Pemrograman Hypertext Preprocessor (PHP) 8 dan Basis Data MySQL. Berikut adalah tampilan dari *user interface* aplikasi:

- *Interface* Pengajuan Judul *project* PAK

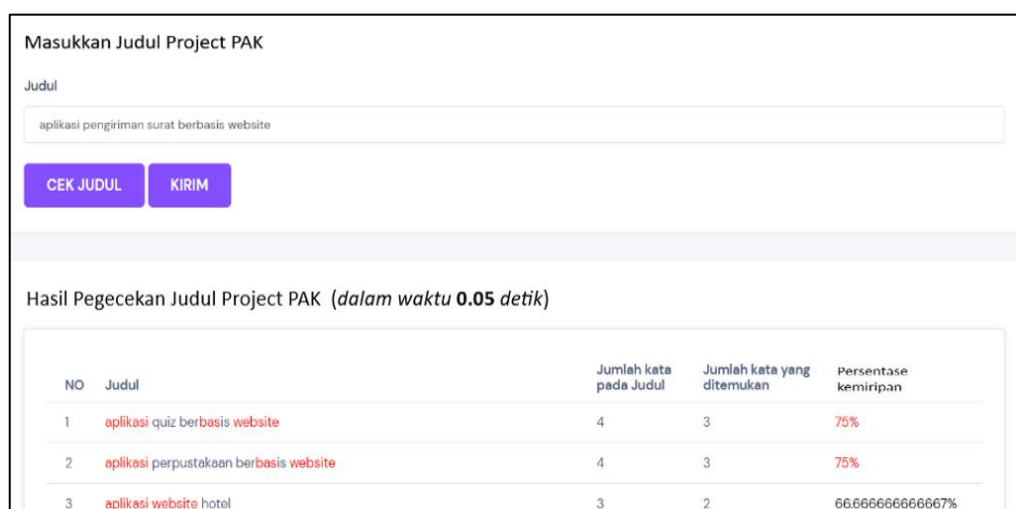
Interface ini digunakan oleh mahasiswa ketika hendak mengecek kemiripan judul yang akan diajukan dengan judul-judul yang sudah ada.


Gambar 3. *Interface* Pengajuan Judul

Gambar 3 memperlihatkan *interface* dari pengajuan judul. Mahasiswa akan diminta untuk memasukkan sebuah judul, misalnya ”aplikasi pengiriman surat berbasis website”. Setelah itu, mahasiswa menekan tombol “CEK JUDUL” untuk mengetahui apakah judul yang diajukan tersebut memiliki kesamaan atau ada kemiripan dengan judul-judul terdahulu atau judul baru yang sudah disetujui oleh dosen.

- *Interface* Hasil Pengecekan Judul

Interface yang menampilkan beberapa judul yang sama atau mirip dengan judul yang diajukan beserta dengan tingkat persentasenya.



NO	Judul	Jumlah kata pada Judul	Jumlah kata yang ditemukan	Persentase kemiripan
1	aplikasi quiz berbasis website	4	3	75%
2	aplikasi perpustakaan berbasis website	4	3	75%
3	aplikasi website hotel	3	2	66.666666666667%

Gambar 4. *Interface* Hasil Pengecekan Judul

Gambar 4 menampilkan persentase kemiripan antara judul yang diajukan, ”aplikasi pengiriman surat berbasis website” dengan yang sudah dipamerkan atau yang terdapat di dalam *database*, “aplikasi quiz berbasis website (75%), aplikasi perpustakaan berbasis website (75%), aplikasi website hotel (66,66%)” dengan waktu pengecekan 0.05 detik. Persentase tersebut dapat menjadi bahan pertimbangan bagi mahasiswa apakah judul tersebut sudah layak diajukan atau memasukkan judul yang lain. *Interface* yang sama juga bisa diakses oleh dosen ketika hendak mengecek sebuah judul.

- *Interface* Hasil Responsi Judul

Setelah mahasiswa melakukan *submit* judul, maka judul tersebut akan masuk ke dalam daftar judul yang diajukan sesuai dengan kelasnya. Selanjutnya, dosen pengampu MK akan memeriksa satu per satu judul yang masuk, seperti yang terlihat pada Gambar 5.


Gambar 5. *Interface* Hasil Responsi Judul

Gambar 5 memperlihatkan salah satu judul, “aplikasi pengiriman surat berbasis website”, yang sedang dicek oleh dosen. Judul tersebut memiliki persentase kemiripan tertinggi dengan judul lain sebesar 75%.

B. Pengujian Aplikasi Secara Manual

Ketika seorang mahasiswa hendak melakukan pengajuan judul *project* PAK, terlebih dahulu akan mengecek apakah judul yang akan diajukan tersebut, misalnya “aplikasi penjualan helm berbasis android”, memiliki kesamaan atau kemiripan dengan judul mahasiswa lain. Untuk mengatasi hal ini, algoritma KMP diterapkan di dalam perancangan aplikasi dengan tahapan sebagai berikut:

- *Explode*: memecah judul yang dimasukkan tersebut menjadi kata, sehingga diperoleh “aplikasi”, “penjualan”, “helm”, “berbasis”, “Android”.
- *Stemming*: menemukan kata dasar yang berbahasa Indonesia dari setiap kata menggunakan *library*, sehingga kata dasar dari judul tersebut adalah “aplikasi”, “jual”, “helm”, “basis”, “Android”.
- KMP: pencarian kata yang sama. Secara sistematis, tahapan *string matching* dimulai pada kata pertama yaitu “aplikasi”, dapat dilihat pada Tabel 2.

Tabel 2. *String matching* untuk kata pertama “aplikasi”

No	Judul (di database)	Hasil Pencarian
1	Aplikasi penjualan berbasis web	1
2	Aplikasi parkir	1
3	Aplikasi pengkasan	1

Tabel 2 memperlihatkan proses *string matching* untuk kata pertama dari judul *project* yang diajukan. Dengan mengambil tiga contoh judul dari dalam *database*, dapat dilihat bahwa kata “aplikasi” juga terdapat pada ketiga judul tersebut sehingga hasil pencarian bernilai satu (1).

Setelah *string matching* untuk kata pertama selesai, proses yang sama selanjutnya dilakukan untuk kata kedua “jual”, seperti yang terlihat pada Tabel 3.

Tabel 3. *String matching* untuk kata kedua “jual”

No	Judul	Hasil Pencarian
1	Aplikasi penjualan berbasis web	1+1
2	Aplikasi parkir	1+0
3	Aplikasi pengkasan	1+0

Tabel 3 memperlihatkan proses *string matching* untuk kata kedua dari judul *project*. Seperti pada proses sebelumnya, jika terdapat kecocokan kata di antara judul *project* yang diajukan dengan judul-judul *project* yang terdapat di dalam *database*, maka hasil pencarian diberi nilai 1 (satu), sebaliknya, jika tidak terdapat kecocokan maka hasil pencarian diberi nilai nol (0). Seperti yang terlihat pada Tabel 3, kata “jual” ini hanya terdapat pada judul pertama sehingga hasil pencarian sebelumnya ditambah dengan satu (1), sementara pada judul kedua dan ketiga tidak ditemukan sehingga hasil pencarian sebelumnya ditambah dengan nol (0). Proses *string matching* ini terjadi berulang hingga kata terakhir pada judul yang diajukan.

Tabel 4. Total hasil *string matching* kata

No	Judul	Hasil Pencarian	Total
1	Aplikasi penjualan berbasis web	1+1+1+0	3
2	Aplikasi parkir	1+0	1
3	Aplikasi pengkasan	1+0	1

Tabel 4 menunjukkan banyaknya kata-kata yang sama antara judul *project* yang diajukan dengan judul-judul yang terdapat di dalam *database*.

- Menentukan jumlah persentase kemiripan. Setelah didapatkan total kata yang sama, langkah berikutnya adalah menghitung persentase kemiripan dari antar judul yang diajukan dengan judul-judul yang terdapat di dalam *database*, seperti yang terlihat pada Tabel 5.

Tabel 5. Perhitungan persentase kemiripan

No	Judul	Jumlah kata pada judul	Jumlah kata yang sama	Persentase
1	Aplikasi penjualan berbasis web	4	3	$(3/4) \times 100\% = 75\%$
2	Aplikasi parkir	2	1	$(1/2) \times 100\% = 50\%$
3	Aplikasi pengkasan	2	1	$(1/2) \times 100\% = 50\%$

Tabel 5 memperlihatkan persentase kemiripan judul. Persentase ini didapatkan dengan cara membagi jumlah kata pada judul dengan jumlah kata yang ditemukan sama melalui proses *string matching*, kemudian dikalikan dengan 100%. Dari tabel ini dapat disimpulkan bahwa, judul yang diajukan oleh mahasiswa “Aplikasi penjualan helm berbasis Android”, memiliki kemiripan dengan judul-judul yang telah dipamerkan dengan tingkat persentase tertinggi sebesar 75%.

Dengan merujuk pada contoh judul yang diajukan di atas, dengan hanya mengambil kata “Aplikasi”, maka proses perhitungan algoritma KMP secara sistematis adalah sebagai berikut (dimulai dengan penentuan nilai fungsi pinggiran setiap karakter; dimana J = panjang karakter yang diberikan *pattern*; $P(J)$ = karakter ke- j pada *pattern*; $F(J)$ = nilai fungsi pinggiran dari *pattern*):

- $F(0) = 0$ (karena awal dari fungsi pinggiran ditetapkan dengan nilai 0)

Tabel 6. Tahap $f(0)$ untuk *pattern* “aplikasi”

J	0	1	2	3	4	5	6	7
P(J)	a	p	l	i	k	a	s	i
F(J)	0							

- $F(1)$ = pencocokan $P(0)$ dengan $P\text{-next}(1)$
 $P(0) = a$ dan $P(1) = p$
 $P(0) \neq P(1)$ tidak terjadinya kecocokan
Maka, $F(1) = 0$

Tabel 7. Tahap $f(1)$ untuk *pattern* “aplikasi”

J	0	1	2	3	4	5	6	7
P(J)	a	p	l	i	k	a	s	i
F(J)	0	0						

- $F(2)$ = pencocokan $P(0)$ dengan $P\text{-next}(2)$
 $P(0) = a$ dan $P(2) = l$
 $P(0) \neq P(2)$ tidak terjadinya kecocokan
Maka, $F(2) = 0$

Tabel 8. Tahap $f(2)$ untuk *pattern* “aplikasi”

J	0	1	2	3	4	5	6	7
P(J)	a	p	l	i	k	a	s	i
F(J)	0	0	0					

- $F(3)$ = pencocokan $P(0)$ dengan $P\text{-next}(3)$
 $P(0) = a$ dan $P(3) = i$
 $P(0) \neq P(3)$ tidak terjadinya kecocokan
Maka, $F(3) = 0$

Tabel 9. Tahap $f(3)$ untuk *pattern* “aplikasi”

J	0	1	2	3	4	5	6	7
P(J)	a	p	l	i	k	a	s	i
F(J)	0	0	0	0	0			

- $F(4)$ = pencocokan $P(0)$ dengan $P\text{-next}(4)$
 $P(0) = a$ dan $P(4) = k$
 $P(0) \neq P(4)$ tidak terjadinya kecocokan
Maka, $F(4) = 0$

Tabel 10. Tahap f(4) untuk *pattern* “aplikasi”

J	0	1	2	3	4	5	6	7
P(J)	a	p	l	i	k	a	s	i
F(J)	0	0	0	0	0			

- F(5)= pencocokan P(0) dengan P-next(5)
P(0)= a dan P(1)= a
P(0)= P(5) terjadi kecocokan (sama)
Maka, F(5)= 1

Tabel 11. Tahap f(5) untuk *pattern* “aplikasi”

J	0	1	2	3	4	5	6	7
P(J)	a	p	l	i	k	a	s	i
F(J)	0	0	0	0	0	1		

- F(6)= pencocokan P(0) dengan P-next(6)
P(0)= a dan P(6)= s
P(0)= P(6) tidak terjadinya kecocokan
Maka, F(6)= 0

Tabel 12. Tahap f(6) untuk *pattern* “aplikasi”

J	0	1	2	3	4	5	6	7
P(J)	a	p	l	i	k	a	s	i
F(J)	0	0	0	0	0	1	0	

- F(7)= pencocokan P(0) dengan P-next(7)
P(0)= a dan P(7)= i
P(0)= P(7) tidak terjadinya kecocokan
Maka, F(7)= 0

Tabel 13. Tahap f(7) untuk *pattern* “aplikasi”

J	0	1	2	3	4	5	6	7
P(J)	a	p	l	i	k	a	s	i
F(J)	0	0	0	0	0	1	0	0

Setelah mendapatkan nilai fungsi pinggiran dari masing-masing karakter *pattern*, maka langkah selanjutnya adalah *string matching* algoritma KMP dengan langkah-langkah sebagai berikut:

- *String matching* antara *pattern* (P) “aplikasi” dengan teks (T) “aplikasi penjualan” dimulai pada ujung kiri teks dan ujung kiri *pattern*.
- *String matching* dilakukan per karakter *pattern* dari kiri, (P[1], T[1]) ke kanan hingga ujung teks, T(13).
 - 1) Jika antara karakter pada *pattern* dan karakter pada teks terjadi ketidakcocokan (*mismatch*), maka karakter pada *pattern* bergeser ke kanan sebanyak satu kali. Sebaliknya, jika terjadi kecocokan, maka *string matching* berhenti dan melakukan pencocokan kembali untuk karakter berikutnya sampai karakter pada *pattern* berada di ujung teks.
 - 2) Jika semua karakter di *pattern* cocok dengan semua karakter pada teks, Algoritma kemudian memberitahu penemuan posisi ini, dapat dilihat pada Tabel 14.

Tabel 14. *String matching* antara *pattern* “aplikasi” dan teks “aplikasi penjualan”

I	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
T	a	p	l	i	k	a	s	i		p	e	n	j	u	a			
P	a	p	l	i	k	a	s	i										

Tabel 14 memperlihatkan proses *string matching* antara karakter di kata pertama pada judul yang diajukan “aplikasi” dengan karakter di kata pertama dan sebagian di kata kedua dari judul yang sudah ada di dalam database “aplikasi penju”. Karena ditemukan kecocokan keseluruhan pola karakter pada *pattern* dengan teks, sehingga proses *string matching* selesai.

IV. Kesimpulan dan saran

Dari hasil penelitian yang telah dijabarkan di atas, maka penulis dapat menyimpulkan bahwa, baik dosen maupun mahasiswa dapat dengan mudah mengetahui kemiripan judul *project* PAK yang diajukan dengan judul-judul yang terdapat di dalam database atau yang telah dipamerkan. Penerapan algoritma KMP pada penelitian ini dapat menemukan dengan baik judul yang mirip dan menghitung persentase tingkat kemiripannya. Namun, meskipun persentase kemiripan tinggi, 75%, seorang dosen dapat menyetujui pengajuan sebuah judul dengan pertimbangan lain seperti, dari segi arti kalimat secara keseluruhan. Tingkat persentase kemiripan judul hanya merupakan bahan pertimbangan, bukan sebagai penentu disetujui atau ditolaknya sebuah judul. Persamaan atau kemiripan sebuah judul seorang mahasiswa dengan mahasiswa lainnya sangat sulit dihindari mengingat jumlah mahasiswa yang memprogramkan MK PAK ini sekitar 600-700 per semester. Alasan lain, meskipun judul sama akan tetapi konten aplikasi akan dirancang berbeda. Dengan demikian, penelitian ini dapat dilanjutkan dengan meningkatkan fungsionalitas fitur dimana tidak hanya menampilkan kata yang mirip tetapi juga dapat menampilkan kalimat yang memiliki arti yang mirip. Selain itu, penelitian ini juga dapat dikembangkan sampai pada tahap pengambilan keputusan layak tidaknya judul tersebut diterima dan berbasis *mobile* agar aksesibilitas lebih mudah.

Ucapan Terima Kasih

Penulis mengucapkan banyak terima kasih kepada Christabella Adeline Pakiding, S.Kom. dan Mila Exselina Peda, S.Kom. atas bantuannya dalam proses pembuatan aplikasi dan pengambilan data judul-judul *Project* PAK yang telah dipamerkan pada semester-semester sebelumnya di Program Studi Teknik Informatika dan Sistem Informasi Universitas Dipa Makassar. Terima kasih yang sebesar-besarnya juga penulis sampaikan kepada Universitas Dipa Makassar atas *Research grand* yang telah diberikan, semoga sukses dan jaya selalu.

Daftar Pustaka

- [1] S. Baase and A. Van Gelder, "Computer Algorithms Introduction to Design and Analysis, 3rd Ed.," 2000, Addison-Wesley, Boston.
- [2] D. Krisbiantoro, S. F. Rohim, and I. Santiko, "Perbandingan Algoritma N-gram dan Algoritma Knuth Morris Pratt untuk Mengukur Tingkat Akurasi Plagiarisme pada Dokumen Abstrak Skripsi Berbasis Website," *JITU J. Inform. Technol. Commun.*, vol. 5, no. 1, pp. 30–39, 2021, doi: 10.36596/jitu.v5i1.390.
- [3] R. Kurniawan R, A. H. Lubis, and S. A. Hadisa, "Implementasi Algoritma Knuth Morris Pratt Dalam Pencocokan String Pada Kamus Indonesia–Korea," *J. Sist. Komput. dan Inform.*, vol. 5, no. 1, p. 79, 2023, doi: 10.30865/json.v5i1.6149.
- [4] Nuraini and B. Firmansyah, "Implementasi Algoritma Knuth Morris Prath Untuk Kamus Terjemahan Digital Aceh – Bahasa Indonesia Berbasis Web," *J. Nas. Inform.*, vol. 1, no. 1, p. 66, 2020.
- [5] M. Ilham and A. H. Mirza, "Penerapan Algoritma Knuth Morris Pratt Dalam Fitur Pencarian Pengarsipan Dokumen Pada Sma Plus Negeri 17 Palembang," *J. Softw. Eng. Ampera*, vol. 1, no. 2, pp. 110–121, 2020, doi: 10.51519/journalsea.v1i2.49.
- [6] M. Rossaria, B. Susilo, and Ernawati, "Implementasi Algoritma Pencocokan String Knuth Morris-Pratt dalam Aplikasi Pencarian Dokumen Digital Berbasis Android," 2015, *Jurnal Rekursif*, Vol. 3 No. 2, Bengkulu.
- [7] Nursobah and P. Pahrudin, "Penerapan Algoritma Pencarian Knuth-Morris-Pratt (KMP) Dalam Sistem Informasi Perpustakaan SMK TI Pratama," 2019, *Sebatik, Samarinda*. doi: DOI:10.46984/SEBATIK.V23I1.451.
- [8] K. A. Khairan and H. Ahmadian, "Penerapan Algoritma Knuth-Morris-Pratt Pada Fitur Pencarian Definisi Istilah Standar Operasional Prosedur (Sop) Pada Lembaga Penjaminan Mutu Uin Ar-Raniry," *Cybersp. J. Pendidik. Teknol. Inf.*, vol. 3, no. 1, p. 26, 2019, doi: 10.22373/cj.v3i1.4723.
- [9] K. Hendra and I. Veri, "Implementasi Algoritma Knuth Morris Pratt untuk Pencariaan Data Buku Pada Sistem Informasi Perpustakaan," 2022, *KLIK: Kajian Ilmiah Informatika dan Komputer*, Vol. 3 No. 3, Bandung.
- [10] D. S. Rizky and T. K. S. Ratih, "String Matching Pada Pencarian Di Web Jurnal Menggunakan Algoritma Knuth Morris Pratt (KMP)," 2023, *PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika*, Vol. 16, No. 2. doi: https://doi.org/10.33322/petir.v16i2.2062.
- [11] K. Aryasa, R. D. Likliwatil, Yosep, and R. Prirendi, "Implementasi Algoritma Knuth Morris Pratt Dalam Pencarian Berkas Berbasis Web (Studi Kasus: Dinas Pariwisata Kota Makassar)," *e-Jurnal JUSITI (Jurnal Sist. Inf. dan Teknol. Informasi)*, vol. 11, no. 01, pp. 1–12, 2022, doi:

- 10.36774/jusiti.v11i1.906.
- [12] W. Astuti, "Analisis String Matching Pada Judul Skripsi Dengan Algoritma Knuth-Morris Pratt (Kmp)," *Ilk. J. Ilm.*, vol. 9, no. 2, pp. 167–172, 2017, doi: 10.33096/ilkom.v9i2.136.167-172.
 - [13] D. S. Islamiyati and A. Fikri, "View of Penerapan Algoritma Knuth-Morris-Pratt dalam Mendeteksi Tingkat Kemiripan Judul Skripsi Berbasis Web.pdf."
 - [14] A. Prasetyo, "Pedoman Pembelajaran Algoritma Pemrograman Dasar 1," *Univ. Politek. Purbaya Tegal*, vol. 01, no. December, pp. 1–32, 2018, doi: 10.5281/zenodo.2213705.
 - [15] T. Yokoyama, H. B. Axelsen, and R. Glück, "Fundamentals of reversible flowchart languages," *Theor. Comput. Sci.*, vol. 611, pp. 87–115, 2016, doi: 10.1016/j.tcs.2015.07.046.