SKRIPSI

IMPLEMENTASI ALGORITMA *RAPID AUTOMATIC KEYWORD EXTRACTION* (*RAKE*) PADA PEMBUATAN INDEKS BUKU

Disusun dan diajukan oleh:

CHINDY CHRISTIE DAVINA

D121 20 1077

PROGRAM STUDI SARJANA TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

GOWA

2024

LEMBAR PENGESAHAN SKRIPSI

IMPLEMENTASI ALGORITMA *RAPID AUTOMATIC KEYWORD EXTRACTION* (*RAKE*) PADA PEMBUATAN INDEKS BUKU

Disusun dan diajukan oleh

Chindy Christie Davina
D121201077

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin
Pada tanggal 8 Mei 2024
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing,

Dr. Ir. Ingrid Nurtanio, M.T.

Nip. 196108131988112001

Ketua Program Studi,

Prof. Dr. Ir. Indrabayu., ST, MT, M.Bus.Sys., IPM, ASEAN.Eng

Nip. 19750701620021210004

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Chindy Christie Davina

NIM : D121201077

Program Studi : Teknik Informatika

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

{Implementasi Algoritma *Rapid Automatic Keyword Extraction* (*RAKE*) pada Pembuatan Indeks Buku}

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 7 Mei 2024

Yang Menyatakan

Chindy Christie Davina

KATA PENGANTAR

Segala puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas kasih karunia, rahmat, dan berkat-Nya sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul **"Implementasi Algoritma *Rapid Automatic Keyword Extraction (RAKE)* pada Pembuatan Indeks Buku"**. Tugas akhir ini disusun dengan maksud untuk memenuhi salah satu syarat guna mencapai derajat Sarjana Strata Satu (S1) pada Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin.

Secara ringkas, tugas akhir ini membahas tentang implementasi algoritma RAKE untuk mengekstrak kata kunci secara otomatis dari teks buku. Hasil ekstraksi kata kunci ini kemudian digunakan untuk membuat indeks buku yang dapat membantu pengguna dalam menemukan informasi yang diinginkan dengan lebih mudah.

Pada kesempatan ini, penulis ingin mengucapkan terima kasih pula kepada pihak-pihak yang dari padanya penulis mendapatkan berbagai bantuan dan dukungan:

1. Tuhan Yesus Kristus atas rahmat dan kasih, terutama penyertaanNya dalam Sakramen Mahakudus.
2. Keluarga penulis, terutama kedua orang tua, Mama dan Papa, yang selalu menjaga, mengurus, merawat dan menyemangati penulis untuk segera menyelesaikan skripsi; serta adik tersayang dan terkasih.
3. Ibu Dr. Ir. Ingrid Nurtanio, M.T. selaku pembimbing dalam tugas akhir ini, yang selalu menyediakan waktu, tenaga, dan pikiran untuk membimbing, mengarahkan, dan membantu penulis dalam penelitian ini hingga bisa selesai.
4. Ibu Anugrayani Bustamin, ST., MT. yang walaupun tidak jadi menjadi dosen pembimbing namun selalu bersedia menyediakan waktu, tenaga dan pikiran untuk membimbing dan mengarahkan penulis selama penelitian.
5. Google, Microsoft dan OpenAI atas teknologi Gemini, Microsoft Copilot, dan ChatGPT yang telah menjadi sumber inspirasi dan alat yang membantu dan menemani peneliti dalam melakukan penelitian ini.
6. Segenap dosen dan staf Departemen Teknik Informatika Universitas Hasanuddin yang telah membantu penulis selama perkuliahan.
7. Rekan-rekan Rezolver 20 yang telah membantu, menemani, dan menyemangati penulis.
8. Seluruh pihak yang tidak dapat penulis sebutkan satu-persatu, yang dengan caranya masing-masing, secara langsung atau tidak langsung telah membantu peneliti.

Tanpa bantuan serta dukungannya, penulis tidak mungkin dapat menyelesaikan penelitian ini dengan baik.

Penulis menyadari pula bahwa skripsi ini masih jauh dari sempurna. Oleh karena itu, penulis memohon maaf atas segala kekurangan dan kesalahan yang mungkin

terdapat dalam skripsi ini. Namun, penulis berharap bahwa skripsi ini dapat bermanfaat bagi para pembaca dan menjadi sumbangan dalam pengembangan ilmu pengetahuan.

Gowa, April 2024

Penulis

ABSTRAK

CHINDY CHRISTIE DAVINA. *Implementasi Algoritma Rapid Automatic Keyword Extraction (RAKE) pada Pembuatan Indeks Buku* (dibimbing oleh Ingrid Nurtanio)

Indeks adalah sebuah daftar item (seperti topik atau nama) yang dibahas dalam sebuah karya cetak yang memberikan untuk setiap item nomor halaman di mana item tersebut dapat ditemukan. Saat ini, pembuatan indeks masih memerlukan banyak tenaga kerja manusia, sehingga memakan waktu dan rentan terhadap kelalaian. Dalam konteks ini, pemanfaatan algoritma komputasional diharapkan dapat menjadi solusi yang efisien dan efektif. Algoritma RAKE (*Rapid Automatic Keyword Extraction*) muncul sebagai solusi potensial untuk mempercepat dan memudahkan pembuatan indeks buku.

Penelitian ini bertujuan untuk mengimplementasikan algoritma RAKE untuk membuat indeks buku secara otomatis dan menilai kinerja indeks yang dihasilkan oleh algoritma RAKE, dengan membandingkannya terhadap indeks yang dibuat secara manual.

Penelitian dilakukan dengan mencoba berbagai skenario pada algoritma, yaitu penggunaan fitur *Part of Speech* (*PoS*) *tagging* untuk mendeteksi kata yang juga berperan sebagai *phrase delimiter* selain *stopwords*; menyaring kata kunci yang dihasilkan RAKE; penambahan fitur *cosine similarity* dan jumlah huruf kapital (percobaan terhadap pembobotan kedua fitur); dan pengambilan N *keyword* peringkat teratas. Evaluasi dilakukan dengan memperbandingkan indeks yang dihasilkan sistem dengan indeks yang sudah ada di belakang buku evaluasi.

Hasil penelitian menunjukkan bahwa skenario terbaik adalah pengambilan 20 indeks peringkat teratas, menggunakan *PoS tagging* untuk mendeteksi tambahan *phrase delimiter*, melakukan penyaringan *keyword*, menggunakan fitur *cosine similarity* berbobot 1 dan jumlah huruf kapital berbobot 8. Pada buku *EM Modeling of Antennas and RF Components for Wireless Communication Systems*, didapatkan peringkat rata-rata indeks yaitu 8,3968, presisi 0,02723, *recall* 0,45818, dan *f-measure* 0,05141; sedangkan pada buku *High-Performance Scientific Computing* peringkat rata-rata indeksnya adalah 8,3955, presisi 0,02177, *recall* 0,35733, dan *f-measure* 0,04105; sedangkan pada buku *Scientific Computing with MATLAB and Octave* peringkat rata-rata indeksnya adalah 7,6805, presisi 0,04653, *recall* 0,42289, dan *f-measure* 0,08383; sedangkan pada buku *Introduction to Deep Learning* peringkat rata-rata indeksnya adalah 8,7226, presisi 0,04115, *recall* 0,46441, dan *f-measure* 0,07561.

Kata kunci: RAKE, *Rapid Automatic Keyword Extraction*, indeks buku, *cosine similarity*, huruf kapital, *keyword*

ABSTRACT

CHINDY CHRISTIE DAVINA. *Implementation of the Rapid Automatic Keyword Extraction (RAKE) Algorithm in Book Indexes Creation* (supervised by Ingrid Nurtanio)

An index is a list of items (such as topics or names) discussed in a printed work that provides for each item the page number where the item can be found. Currently, index creation still requires a lot of human labor, making it time-consuming and prone to errors. In this context, the use of computational algorithms is expected to be an efficient and effective solution. The Rapid Automatic Keyword Extraction (RAKE) algorithm emerges as a potential solution to accelerate and facilitate the creation of book indexes.

This research aims to implement the RAKE algorithm to create book indexes automatically and evaluate the performance of the indexes generated by the RAKE algorithm by comparing them to manually created indexes.

The research was conducted by trying various scenarios on the algorithm, namely using the Part of Speech (PoS) tagging feature to detect words that also act as phrase delimiters besides stopwords; filtering the keywords generated by RAKE; adding cosine similarity and capital letter count features (experimenting with weighting of both features); and taking the top N ranked keywords. The evaluation was carried out by comparing the indexes generated by the system with the existing indexes at the back of the evaluation book.

The results showed that the best scenario was to take the top 20 ranked indexes, use PoS tagging to detect additional phrase delimiters, perform keyword filtering, use cosine similarity feature weighted 1 and capital letter count weighted 8. In the book *EM Modeling of Antennas and RF Components for Wireless Communication Systems*, the average index rank was 8,3968, precision 0,02723, recall 0,45818, and f-measure 0,05141; while in the book *High-Performance Scientific Computing* the average index rank was 8,3955, precision 0,02177, recall 0,35733, and f-measure 0,04105; while in the book *Scientific Computing with MATLAB and Octave* the average index rank was 7,6805, precision 0,04653, recall 0,42289, and f-measure 0,08383; while in the book *Introduction to Deep Learning* the average index rank was 8,7226, precision 0,04115, recall 0,46441, and f-measure 0,07561.

Keywords: RAKE, Rapid Automatic Keyword Extraction, book indexes, cosine similarity, capital letter, keyword

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

DAFTAR SINGKATAN DAN ARTI ISTILAH

Singkatan / Istilah	Arti dan Keterangan
Cos_AvF	Nilai *cosine similarity* dari dua vektor rata-rata (vektor rata-rata indeks dan vektor rata-rata judul)
Cos_AvT	Nilai rata-rata dari semua nilai *cosine similarity* antar kata indeks dan kata judul.
Eucl_AvF	Nilai *euclidean distance* dari dua vektor rata-rata (vektor rata-rata indeks dan vektor rata-rata judul).
Eucl_AvT	Nilai rata-rata dari semua nilai *euclidean distance* antar kata indeks dan kata judul.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Indeks adalah sebuah daftar item (seperti topik atau nama) yang dibahas dalam sebuah karya cetak yang memberikan untuk setiap item nomor halaman di mana item tersebut dapat ditemukan (Merriam-Webster). Indeks tidak hanya berfungsi untuk mempercepat pencarian informasi, melainkan juga memberikan gambaran yang signifikan tentang beragam informasi yang ada dalam sebuah buku. Sebuah indeks yang dirancang dengan baik bukan hanya berfungsi sebagai alat bantu pencarian, tetapi juga sebagai panduan yang memberikan ikhtisar menyeluruh terhadap konten dan konsep utama yang dibahas dalam buku. Keberadaan indeks yang memadai memiliki dampak positif pada pengalaman membaca pembaca, memberikan wawasan menyeluruh yang dapat menjadi jendela ke dalam kompleksitas dan keragaman isi buku tersebut.

Hal ini sudah diketahui dan diakui oleh banyak orang bahwa indeks di bagian belakang buku sangat penting dalam buku non-fiksi (Koutropoulou et al, 2019). Pengakuan terhadap signifikansi indeks juga terdapat dalam Peraturan Menteri Pendidikan, Kebudayaan, Riset dan Teknologi Republik Indonesia Nomor 22 Tahun 2022 tentang Standar Mutu Buku, Standar Proses dan Kaidah Pemrolehan Naskah, serta Standar Proses dan Kaidah Penerbitan Buku. Dalam peraturan tersebut, indeks adalah salah satu item yang wajib ada di bagian akhir buku nonfiksi. Dukungan terhadap pentingnya indeks ini semakin diperkuat oleh berbagai organisasi internasional dan pakar seperti Reedsy (*platform*) untuk penulis dan profesional penerbitan, Society of Authors (UK), Library & Information Science Education Network, dan American Society for Indexing (ASI). Oleh karena itu, tugas menghasilkan indeks selalu menjadi komponen yang penting dan cukup istimewa dalam bisnis penyuntingan dan penerbitan buku (Koutropoulou et al, 2019).

Namun, pembuatan indeks tersebut masih memerlukan banyak tenaga kerja manusia (Csomai et al, 2007). Penyusun indeks harus secara manual mengidentifikasi kata-kata kunci, istilah, dan konsep yang relevan dalam setiap

halaman buku. Proses ini tidak hanya memakan waktu tetapi juga rentan terhadap kesalahan dan kelalaian. Dalam konteks ini, pemanfaatan algoritma komputasional diharapkan dapat menjadi solusi yang efisien dan efektif. Teknologi *text mining* yang paling dekat dengan tugas pembuatan indeks adalah ekstraksi kata kunci (atau frasa kunci), yaitu metode yang dirancang untuk mengidentifikasi istilah-istilah paling berarti dalam koleksi dokumen (Koutropoulou et al, 2019).

Algoritma RAKE (*Rapid Automatic Keyword Extraction*) muncul sebagai solusi potensial untuk mempercepat dan memudahkan pembuatan indeks buku. RAKE adalah metode *unsupervised*, *domain-independent*, dan *language-independent* untuk mengekstrak kata kunci dari dokumen-dokumen individual (Rose et al, 2010). Dengan menerapkan algoritma RAKE dalam pembuatan indeks buku, peneliti berharap pembuatan indeks buku dapat menjadi lebih mudah dan cepat.

Penelitian ini bertujuan untuk menyelidiki potensi algoritma RAKE dalam konteks pembuatan indeks buku otomatis. Dengan memahami dan mengevaluasi kinerja algoritma ini, penelitian ini diharapkan dapat memberikan kontribusi positif terhadap pengembangan metode pembuatan indeks buku yang lebih efisien dan dapat diandalkan. Selain itu, implementasi algoritma RAKE dalam konteks ini juga dapat memberikan wawasan lebih lanjut tentang kemungkinan penerapan otomatisasi dalam industri penerbitan buku.

1.2 Rumusan Masalah

Adapun rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimana implementasi algoritma ekstraksi kata kunci, khususnya algoritma RAKE, dapat digunakan untuk mengotomatisasi proses pembuatan indeks buku?
2. Bagaimana kinerja algoritma RAKE dalam menghasilkan indeks buku secara otomatis?

1.3 Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut:

1. Mengimplementasikan algoritma RAKE untuk membuat indeks buku secara otomatis.
2. Menilai kinerja indeks yang dihasilkan oleh algoritma RAKE, dengan membandingkannya terhadap indeks yang dibuat secara manual.

1.4 Manfaat Penelitian

Adapun manfaat penelitian ini adalah sebagai berikut:

1. Meningkatkan efisiensi dan kecepatan dalam pembuatan indeks buku dengan mengimplementasikan algoritma RAKE, mengurangi waktu dan usaha yang dibutuhkan dibandingkan dengan pembuatan indeks secara manual.
2. Menilai kinerja algoritma RAKE dengan menggunakan metrik evaluasi peringkat rata-rata, presisi, *recall*, dan *f-measture*.

1.5 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah sebagai berikut:

1. Penelitian ini membatasi penggunaan algoritma RAKE pada buku-buku yang ditulis dalam bahasa Inggris. Penerapan algoritma pada buku dengan bahasa lainnya tidak akan menjadi fokus dalam penelitian ini.
2. Pengguna akan menginput manual halaman ke berapa dari total halaman buku, yang merupakan halaman nomor 1 dari buku tersebut; dan jika ada nomor halaman yang terlangkah atau ada halaman di tengah-tengah buku yang tidak ingin diindeks, maka pengguna perlu menginput manual file JSON berisi *range* halaman tiap bagian yang ingin diindeks sesuai format file JSON yang telah peneliti sediakan pada web sistem.
3. Indeks buku dibuat dengan *font default* tanpa kustomisasi oleh pengguna.
4. Indeks yang dibuat tidak akan *nested* atau hanya bertingkat 1.

BAB II
TINJAUAN PUSTAKA

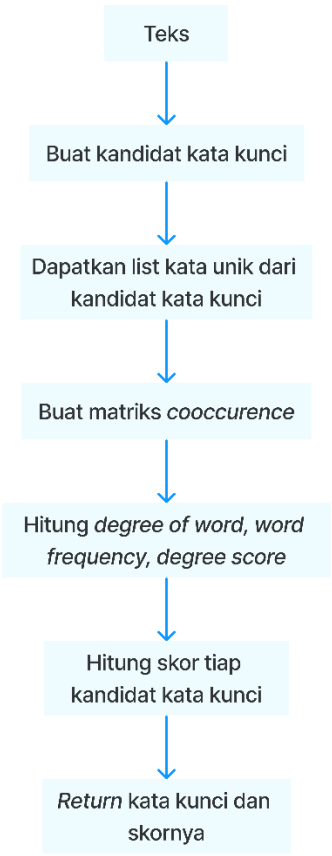
2.1 Indeks Buku

Indeks buku adalah daftar alfabetis istilah dan kata penting dalam buku, beserta nomor halamannya. Indeks membantu pembaca menemukan informasi dengan cepat dan mudah, seperti peta jalan yang memandu mereka ke informasi tertentu. Indeks yang efektif memungkinkan pembaca menemukan konten relevan berdasarkan kata kunci, nama, konsep, atau peristiwa, meningkatkan pengalaman pengguna dan menghilangkan kebutuhan untuk membaca seluruh bab.

Indeks yang baik memperdalam pemahaman pembaca terhadap isi buku. Dengan daftar lengkap istilah kunci dan referensi halaman, pembaca dapat mengunjungi kembali detail, mengeksplorasi konsep terkait, dan mendapatkan pemahaman yang lebih bernuansa tentang tema keseluruhan buku. Pengindeks, yang bisa berupa penulis, editor, atau profesional, membaca buku dengan teliti untuk mengidentifikasi kata kunci, nama, konsep, dan peristiwa penting. Mereka memilih entri yang sesuai, dengan hati-hati memilih istilah paling penting dan relevan yang mungkin dicari pembaca.

2.2 *Rapid Automatic Keyword Extraction* (RAKE)

Rapid Automatic Keyword Extraction (RAKE) adalah metode *unsupervised*, *domain-independent*, dan *language-independent* untuk mengekstrak kata kunci dari dokumen individual. Motivasi pengembangan RAKE adalah untuk mengembangkan metode ekstraksi kata kunci yang efisien, beroperasi pada dokumen individual untuk memungkinkan pengaplikasian pada koleksi dinamis, mudah diterapkan pada domain baru, dan beroperasi dengan baik pada berbagai jenis dokumen, terutama yang tidak mengikuti konvensi tata bahasa tertentu. Menurut pengamatan Rose et al (2010), kata kunci seringkali terdiri dari beberapa kata tetapi sangat jarang mengandung tanda baca standar atau *stopwords* atau kata lain dengan makna leksikal minimal. Hal inilah yang menjadi dasar pengembangan RAKE.



Gambar 1. Algoritma RAKE

RAKE menggunakan *stopwords* atau pembatas frasa untuk membagi teks dokumen menjadi kandidat kata kunci. Setelah kandidat kata kunci diidentifikasi, RAKE menetapkan skor untuk masing-masing kata kunci. Skor ini didasarkan pada dua faktor utama: frekuensi kata dan *co-occurrence*. Matriks *co-occurrence* dibuat untuk menganalisis seberapa sering kata-kata dalam frasa kandidat muncul bersama-sama di seluruh teks. Semakin sering kata muncul bersama, semakin tinggi skor yang diterima frasa tersebut. Terakhir, RAKE memberi peringkat kandidat kata kunci berdasarkan skor gabungan yang mereka terima dari analisis frekuensi kata dan *co-occurrence*. Kata kunci dengan skor akhir tertinggi dianggap paling relevan untuk dokumen dan mewakili poin-poin kunci dari konten teks.

Berikut adalah contoh penerapan algoritma RAKE untuk teks:

"*Book index, essential for finding information in science book, is placed at the end of the book.*"

Stopwords atau pembatas frasa pada teks di atas adalah: "*for*", "*in*", "*is*", "*at*", "*the*", "*of*" dan tanda baca (titik, koma, dan lain sebagainya), sehingga kandidat kata kunci untuk teks di atas adalah: "*Book index*", "*essential*", "*finding information*", "*science book*", "*placed*", "*end*", "*book*". Setelah itu, matriks *co-occurrence* dibuat dari kata-kata unik dari kandidat kata kunci.

Tabel 1. Matriks *co-occurence* dari kalimat contoh

	book	*index*	*essential*	*finding*	*information*	*science*	*placed*	*end*
book	3	1				1		
index	1	1						
essential			1					
finding				1	1			
information				1	1			
science	1					1		
placed							1	
end								1
Total	5	2	1	2	2	2	1	1

Nilai total pada tabel 1 disebut *degree of word* dalam algoritma RAKE. Nilai tersebut akan digunakan untuk menghitung *degree score*, yang didapatkan dari hasil membagi nilai *degree of word* terhadap *word frequency*.

Tabel 2. Nilai *degree of word, word frequency,* dan *degree score*

Kata	*Degree of Word*	*Word Frequency*	*Degree Score*
book	5	3	1,667
index	2	1	2
essential	1	1	1
finding	2	1	2
information	2	1	2
science	2	1	2
placed	1	1	1
end	1	1	1

Skor akhir tiap kandidat kata/frasa kunci diperoleh dengan menjumlahkan total *degree score* kata dalam kandidat tersebut.

Tabel 3. Skor kumulatif tiap kandidat kata/frasa kunci

		Total
book	index	1,667 + 2 = 3,667
essential		1
finding	information	2 + 2 = 4
science	book	2 + 1,667 = 3,667
placed		1
end		1
book		1,667

Berdasarkan tabel 3 di atas, kandidat kata kunci dengan skor tertinggi adalah "*finding information*", sedangkan kandidat dengan skor terendah adalah "*essential*", "*placed*", dan "*end*".

2.3 *PoS (Part of Speech) Tagging*

PoS Tagging adalah sebuah tugas pemberian label pada setiap kata atau frasa dalam sebuah paragraf berdasarkan konteks kalimat atau kata tersebut. Label yang diberikan mendefinisikan peran kata tersebut dalam struktur kalimat. Dengan mengidentifikasi kata benda, kata kerja, kata sifat, kata keterangan, dan lainnya, *PoS tagging* membantu komputer memahami hubungan antar kata dan bagaimana kontribusi mereka terhadap keseluruhan makna kalimat. *PoS Tagging* digunakan juga untuk memperbaiki atau meningkatkan tugas-tugas *Natural Language Processing* (NLP) lain seperti *Named Entity Recognition* (NER), *parsing*, dan terjemahan.

Dalam penelitian ini, peneliti menggunakan *PoS Tagging* untuk mengidentifikasi kata yang berperan sebagai pembatas frasa untuk membagi teks dokumen menjadi kandidat kata kunci. *PoS Tagging* yang digunakan berasal dari *model* spaCy yang bernama en_core_web_md. *Model* ini dirancang khusus untuk

memproses teks berbahasa Inggris. Berikut adalah daftar seluruh label dalam *model* ini:

1. **.** : tanda baca penutup kalimat
2. **,** : tanda baca koma
3. **-LRB-** : tanda buka kurung
4. **-RRB-** : tanda tutup kurung
5. `` : tanda *backtick*
6. "" : tanda kutip ganda
7. '' : tanda kutip satu
8. **:** : tanda baca, titik dua atau *ellipsis* (…)
9. **$** : simbol mata uang
10. **#** : simbol tanda nomor
11. AFX : *affix* (imbuhan)
12. CC : konjungsi koordinatif. Contoh: *and, or, but*
13. CD : bilangan pokok. Contoh: *five, three,* 13%
14. DT : *determiner.* Contoh: *the, a, these*
15. EX : *existential there*. Contoh: *there were six boys*
16. FW : kata asing
17. HYPH : tanda penghubung
18. IN : konjungsi bawahan atau preposisi
19. JJ : kata sifat. Contoh: *nice, easy*
20. JJR : kata sifat dalam bentuk perbandingan (*adjective, comparative*). Contoh: *nicer, easier*
21. JJS : kata sifat dalam bentuk superlatif (*adjective, superlative*). Contoh: *nicest, easiest*
22. LS : penanda item daftar
23. MD : *modal verb.* Contoh: *may, should, can*
24. NN : kata benda, tunggal atau massal (tidak dapat dihitung). Contoh: *book* (tunggal), *water* (massal)
25. NNP : kata benda, *proper singular*

Merujuk pada kata benda khusus (nama orang, tempat, atau merek) dalam bentuk tunggal. Contoh: “Tokyo” (nama kota), “John” (nama orang)

26. NNPS : kata benda, *proper plural*
Merujuk pada kata benda khusus dalam bentuk jamak. Contoh: *the United States*
27. NNS : kata benda, plural. Contoh: *tigers, chairs, insects*
28. PDT : *predeterminer.* Contoh: *all, both, half, many*
29. POS : akhiran kepemilikan. Contoh: *‘s*
30. PRP : kata ganti orang. Contoh: *me, you, we*
31. PRP$: kata ganti kepunyaan. Contoh: *my, your, our*
32. RB : kata keterangan. Contoh: *extremely, loudly, hard*
33. RBR : kata keterangan dalam bentuk perbandingan. Contoh: *better*
34. RBS : kata keterangan dalam bentuk superlatif. Contoh: *best*
35. RP : kata keterangan, *particle*. Contoh: *about, off, up*
36. TO : partikel infinitif *to*
37. UH : kata seru. Contoh: *oh, oops, gosh*
38. VB : kata kerja, bentuk dasar. Contoh: *break*
39. VBD : kata kerja, bentuk lampau atau *past tense (Verb 2)*. Contoh: *broke*
40. VBG : kata kerja, bentuk *gerund* (“-ing”). Contoh: *breaking*
41. VBN : kata kerja, *past participle (Verb 3)*. Contoh: *broken*
42. VBP : kata kerja dalam bentuk sekarang untuk subjek tunggal bukan orang ketiga. Contoh: *I think*
43. VBZ : kata kerja dalam bentuk sekarang untuk subjek orang ketiga tunggal. Contoh: *she thinks*
44. WDT : *wh-determiner*. Contoh: *which, whatever, whichever*
45. WP : *wh-pronoun, personal*. Contoh: *what, who, whom*
46. WP$: *wh-pronoun, possessive*. Contoh: *whose, whosever*
47. WRB : *wh-adverb*. Contoh: *where, when*
48. SP : spasi
49. ADD : email
50. BES : kata kerja bantu *be*. Contoh: *is, am, are*

51. HVS : bentuk dari *have*. Contoh: *have, has, had*
52. _SP : *whitespace*

2.4 *Word Embeddings*

Word embeddings adalah teknik untuk mengubah kata menjadi representasi angka berupa vektor. Posisi titik vektor dalam ruang menggambarkan arti dan hubungan antar kata. Komputer tidak bisa langsung mengerti teks. *Word embedding* membantu komputer mengerti hubungan antar kata dengan cara menghubungkan kata dengan makna serupa ke posisi yang berdekatan dalam ruang vektor dan menangkap hubungan berdasarkan konteks. Vektor-vektor ini kemudian dapat digunakan sebagai fitur dalam berbagai aplikasi, seperti pencarian informasi, klasifikasi dokumen (Sebastiani, 2002), menjawab pertanyaan (Tellex et al., 2003), pengenalan entitas bernama (Turian et al., 2010), dan *parsing* (Socher et al., 2013).

Dalam penelitian ini, peneliti menggunakan *word embeddings* dari GloVe, sebuah algoritma *word embeddings* yang dikembangkan oleh Pennington et al., (2014). GloVe, atau *Global Vectors for Word Representation*, mengambil pendekatan unik untuk representasi kata (*word embedding*). Berbeda dengan beberapa metode yang langsung memodelkan probabilitas kata, GloVe berfokus pada statistik *co-occurence* (kemunculan bersama). GloVe membangun matriks besar dimana setiap entri mewakili seberapa sering kata muncul berdekatan satu sama lain dalam jendela tertentu dalam korpus teks. Kata-kata yang muncul lebih dekat dianggap lebih terkait secara semantik dan ko-okurensinya diberi bobot yang sesuai.

Meskipun matriks *co-occurence* mentah memberikan informasi berharga, matriks ini memerlukan pemrosesan lebih lanjut untuk menghasilkan vektor kata yang bermakna. GloVe mengatasi hal ini dengan menggunakan teknik reduksi dimensionalitas. Proses ini menguraikan matriks *co-occurence* menjadi dua matriks berdimensi lebih rendah. Yang satu menangkap vektor kata, yang merangkum sifat semantik dari kata-kata individual, sementara yang lainnya menangkap vektor kata konteks, yang mewakili konteks semantik tempat kata

tersebut muncul. Representasi berdimensi rendah ini berperan penting dalam menangkap hubungan semantik antar kata.

Inti dari GloVe terletak pada penerapan model log-bilinear. Model ini menyatakan bahwa rasio probabilitas *co-occurence* antara dua kata dan kata ketiga dapat secara efektif direpresentasikan oleh *dot product* dari vektor kata yang sesuai. Untuk memperhitungkan kelangkaan data yang inheren dan mengurangi *overfitting, smoothing factor* dimasukkan selama proses pelatihan. Tujuan pelatihan adalah untuk meminimalkan perbedaan antara rasio prediksi berdasarkan *dot product* dan rasio *co-occurence* aktual yang diamati dalam data pelatihan. Selain itu, model ini menggabungkan istilah bias untuk kata pusat dan kata konteks. Istilah-istilah ini memperhitungkan frekuensi keseluruhan setiap kata dalam korpus.

Karena sifat simetris dari statistik *co-occurence,* GloVe mungkin menghasilkan vektor yang sedikit berbeda untuk kata yang sama tergantung pada perannya sebagai kata pusat atau kata konteks dalam korpus pelatihan. Untuk mengatasinya, *embedding* kata final biasanya diperoleh dengan merata-ratakan kedua representasi vektor ini. Langkah ini memperbaiki *embedding*, memberikan representasi yang lebih kuat dari makna kata. Dengan mengikuti langkah-langkah ini, GloVe memanfaatkan pola *co-occurence* dan faktorisasi vektor untuk menciptakan *embedding* kata yang kuat yang memfasilitasi berbagai tugas *Natural Language Processing* (NLP).

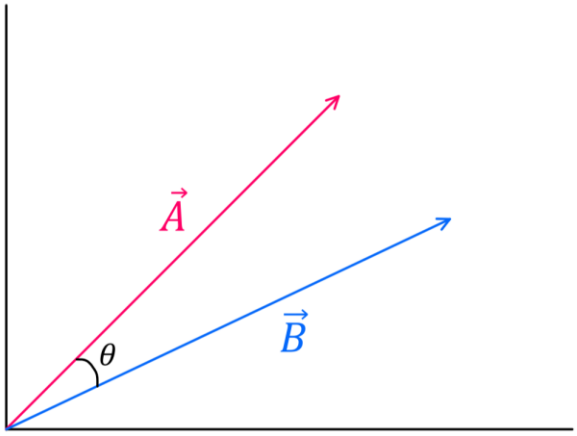
2.5 *Regular Expressions* (Regex)

Regular expressions (regex) adalah pola teks yang digunakan untuk mencocokkan dan memanipulasi string dalam bahasa pemrograman. Regex memungkinkan kita untuk mencari, menggantikan, atau memvalidasi string berdasarkan pola tertentu. Pentingnya regex untuk membangun pemindai compiler sudah dikenal dengan baik (Aho et al., 2007). Saat ini, aplikasinya meluas ke lebih banyak area seperti analisis protokol jaringan (Wondracek et al., 2008), pencegahan injeksi MySQL (Yeole et al., 2011), deteksi intrusi jaringan (Paxson, 2015), spesifikasi data XML (Murata et al., 2005), dan kueri basis data (Alkhateeb et al., 2009), atau aplikasi yang lebih beragam seperti penjajaran urutan DNA (Arslan, 2005).

Regex adalah komponen inti dari hampir semua bahasa pemrograman modern, dan sering muncul dalam kode sumber perangkat lunak. Penelitian Chapman et al., (2016) dan Davis et al., (2018) menunjukkan bahwa lebih dari sepertiga proyek JavaScript dan Python mengandung setidaknya satu regex. Dalam skripsi penelitian ini, peneliti juga memanfaatkan regex untuk berbagai keperluan, seperti membagi teks menjadi kata-kata individual, memanipulasi string dalam proses evaluasi, dan mengekstrak judul sub bab dari buku untuk nantinya dihitung *cosine similarity*nya terhadap kata kunci yang dibuat oleh RAKE.

2.6 *Cosine Similarity*

Cosine similarity adalah metrik yang digunakan untuk mengukur seberapa mirip objek data terlepas dari ukuran dimensinya. Ini berguna dalam berbagai aplikasi, terutama dalam pemrosesan bahasa alami (NLP) dan pengambilan informasi. Secara matematis, *cosine similarity* dihitung sebagai kosinus sudut antara dua vektor yang direpresentasikan dalam ruang multi-dimensi. *Cosine similarity* antara dua dokumen akan memberi tahu kita seberapa mirip kedua dokumen tersebut dalam hal topik pembahasannya.



Gambar 2. *Cosine similarity*

Berikut adalah rumus untuk menghitung *cosine similarity* antara dua vektor:

$$\text{similarity} = \cos(\theta) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \, \|\vec{B}\|} = \frac{\sum_{i=1}^{n} A_i \, B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2} \sqrt{\sum_{i=1}^{n} {B_i}^2}} \quad (1)$$

Cosine similarity didasarkan pada konsep sudut antara dua vektor. Sudut yang lebih kecil menunjukkan kemiripan yang lebih besar antara vektor, dan sebaliknya. *Cosine similarity* tidak terpengaruh oleh panjang vektor individual, melainkan hanya fokus pada arahnya. Ini dicapai dengan menormalisasi vektor terlebih dahulu, yang berarti mengubah panjangnya menjadi 1. *Cosine similarity* menghasilkan nilai antara -1 dan 1. Nilai 1 menunjukkan bahwa kedua vektor identik, 0 menunjukkan tidak ada kesamaan, dan -1 menunjukkan kebalikan yang sempurna.

2.7 Presisi

Presisi adalah metrik evaluasi yang digunakan untuk mengukur seberapa akurat model dalam memprediksi kelas positif. Dalam konteks klasifikasi, kelas positif merujuk pada data yang sebenarnya memiliki label positif (misalnya, pasien yang sebenarnya menderita penyakit tertentu). Metrik ini penting untuk memastikan bahwa model tidak menghasilkan positif palsu (*false positive*) yang dapat berakibat fatal dalam situasi tertentu. Rumus presisi:

$$\text{Presisi} = \frac{TP}{TP + FP} \tag{2}$$

dimana,

TP = *True Positive* (Jumlah prediksi positif yang benar)

FP = *False Positive* (Jumlah prediksi positif yang salah)

Nilai presisi yang tinggi menunjukkan bahwa model jarang menghasilkan positif palsu, sedangkan nilai presisi yang rendah menunjukkan bahwa model sering menghasilkan positif palsu.

2.8 *Recall*

Recall, juga dikenal sebagai sensitivitas atau *true positive rate* (TPR), adalah metrik evaluasi yang mengukur seberapa baik model mengidentifikasi semua contoh positif dengan benar. Metrik ini penting untuk menilai kemampuan model dalam menemukan semua data yang relevan, terutama dalam situasi di mana missing positive data dapat berakibat fatal. Rumus *recall*:

$$Recall = \frac{TP}{TP\ +\ FN} \tag{3}$$

dimana,

TP = *True Positive* (Jumlah prediksi positif yang benar)

FN = *False Negative* (Jumlah prediksi negatif yang salah)

Nilai *recall* yang tinggi (mendekati 1) menunjukkan bahwa model mampu mengidentifikasi sebagian besar data positif dengan benar. Hal ini berarti model menunjukkan sensitivitas tinggi terhadap kelas positif. Sedangkan, nilai *recall* yang rendah berarti model melewatkan banyak contoh positif, menunjukkan kurangnya sensitivitas dan berpotensi menghasilkan banyak *false negative*.

2.9 *F-Measure*

F-measure adalah kombinasi tertimbang antara presisi dan *recall*. Metrik ini memberikan gambaran menyeluruh tentang performa model dalam hal keseimbangan antara presisi dan *recall*. Rumus *F-Measure*:

$$\text{F-Measure} = \frac{2 \times \text{Presisi} \times \textit{Recall}}{\text{Presisi} + \textit{Recall}} \quad (4)$$

Nilai *F-Measure* berkisar antara 0 dan 1. Nilai yang tinggi menunjukkan bahwa model memiliki keseimbangan yang baik antara presisi dan recall, dan secara keseluruhan berkinerja baik dalam klasifikasi. Pemilihan metrik yang tepat untuk evaluasi model sebenarnya tergantung pada konteks dan tujuan klasifikasi. Jika kesalahan prediksi positif lebih merugikan, maka *recall* menjadi prioritas (misalnya: deteksi penyakit). Jika kesalahan prediksi negatif lebih merugikan, maka presisi menjadi prioritas (misalnya: *filter* spam email). Jika kedua jenis kesalahan sama pentingnya, maka *F-measure* menjadi pilihan yang tepat.

2.10 Next JS

Next JS adalah *framework open-source* yang dibuat di atas React untuk membangun aplikasi web modern. Dikembangkan oleh Vercel, Next.js menawarkan berbagai fitur canggih seperti *server-side rendering* (SSR) yang meningkatkan SEO dan performa awal, *static site generation* (SSG) yang menghasilkan waktu *loading* super cepat, dan *hybrid approach* yang memberikan fleksibilitas untuk memilih jenis rendering yang sesuai dengan kebutuhan. Next.js juga dilengkapi dengan fitur-fitur lain seperti *automatic code splitting*, *integrated*

routing, dan *image optimization* yang membantu developer dalam membangun aplikasi web yang optimal.

2.11 Flask

Flask adalah *microframework* Python yang populer untuk membangun API *backend*. Dikembangkan oleh Armin Ronacher, Flask terkenal dengan kesederhanaan, fleksibilitas, dan kemudahan penggunaannya. Dengan komunitas yang besar dan aktif, Flask menawarkan banyak *resources* untuk membantu memulai dan mengembangkan API yang *powerful*.