

IMPLEMENTASI METODE *THESAURUS* DALAM SISTEM TEMU BALIK INFORMASI BERBASIS WEB UNTUK PENINGKATAN AKURASI Pencarian DENGAN PENGURUTAN *BUBBLE SORT*

Lala Aqila Putri Arliantha¹, Safrizal², Ade Tia Kesuma³, Silva Davira⁴, Aminarti⁵, Mutia Dinda Isnaini⁶
Sistem Informasi, STMIK Kaputama^{1,3,4,5,6}, Universitas Muhammadiyah Asahan², Sumatera Utara

Email: *llalakk.a21@gmail.com¹, rizalsyl75@gmail.com², adetiakusuma6@gmail.com³,
davirasilva738@gmail.com⁴, amiinartii7@gmail.com⁵, mutiadinda25@gmail.com⁶

ABSTRAK

Pesatnya perkembangan informasi digital menyebabkan meningkatnya jumlah data yang tersedia, sehingga pengguna sering mengalami kesulitan dalam memperoleh informasi yang relevan dan akurat. Permasalahan ini mendorong perlunya pengembangan sistem temu balik informasi yang mampu meningkatkan kualitas hasil pencarian. Penelitian ini bertujuan untuk mengimplementasikan metode *thesaurus* dalam sistem temu balik informasi berbasis web guna memperluas kata kunci pencarian serta meningkatkan akurasi hasil pencarian. Metode *thesaurus* digunakan untuk melakukan ekspansi *query* dengan menambahkan istilah-istilah yang memiliki hubungan makna dengan kata kunci utama. Selanjutnya, algoritma *bubble sort* diterapkan untuk mengurutkan hasil pencarian berdasarkan tingkat relevansi yang diperoleh dari proses pembobotan. Sistem ini diimplementasikan menggunakan bahasa pemrograman PHP dengan antarmuka berbasis web. Hasil pengujian menunjukkan bahwa penerapan metode *thesaurus* mampu meningkatkan relevansi pencarian dengan menampilkan informasi yang lebih sesuai dengan kebutuhan pengguna. Pengurutan menggunakan *bubble sort* membantu menyajikan hasil pencarian secara sistematis dari tingkat relevansi tertinggi hingga terendah, sehingga memudahkan pengguna dalam memperoleh informasi yang diinginkan. Dengan demikian, kombinasi metode *thesaurus* dan algoritma *bubble sort* terbukti efektif dalam meningkatkan akurasi dan kualitas sistem temu balik informasi.

Kata kunci

Sistem Temu Balik Informasi, *Thesaurus*, Ekspansi Query, *Bubble Sort*, Relevansi Pencarian

ABSTRACT

The rapid growth of digital information has led to an increasing volume of data, making it difficult for users to obtain relevant and accurate information. This condition highlights the need for an information retrieval system capable of improving search accuracy and relevance. This study aims to implement a thesaurus method in a web-based information retrieval system to expand search queries and enhance search accuracy. The thesaurus method is used to perform query expansion by adding terms that are semantically related to the main keywords. Furthermore, the bubble sort algorithm is applied to sort search results based on relevance scores obtained from the weighting process. The system is implemented using PHP as the programming language with a web-based interface. The results show that the application of the thesaurus method improves search relevance by presenting information that better matches user needs. The use of the bubble sort algorithm helps organize search results from the highest to the lowest relevance, making the information easier to

understand and access. Therefore, the combination of the thesaurus method and the bubble sort algorithm is proven to effectively enhance the accuracy and quality of information retrieval systems.

Keywords

Information Retrieval System, Thesaurus, Query Expansion, Bubble Sort, Search Relevance

1. PENDAHULUAN

Pesatnya pertumbuhan informasi digital menyebabkan dimensi data semakin besar sehingga informasi penting sering tersebar luas dan menjadi kurang informatif. Kondisi ini melatarbelakangi munculnya sistem temu balik informasi sebagai cabang ilmu dalam teknologi informasi yang bertujuan memperoleh informasi relevan dari kumpulan data (Hasan, 2018). Namun, akurasi pencarian masih dipengaruhi oleh variasi penulisan, kesalahan ketik, perbedaan istilah, serta keterbatasan sistem dalam memahami relasi antar kata, sehingga hasil pencarian belum optimal.

Thesaurus merupakan salah satu komponen pendukung yang bernilai dalam memanfaatkan korelasi antar dokumen (Informatika *et al.*, 2004). *Thesaurus* berisi kumpulan kata yang dikelompokkan berdasarkan hubungan makna dan digunakan sebagai model domain untuk mengidentifikasi kesamaan antara sumber informasi dan query (BRUNDY, 2019). Proses digitalisasi *thesaurus* memungkinkan pencarian sinonim dilakukan secara otomatis dan lebih efisien (Zamzami, Rahutomo and ..., 2016).

Menurut Tambunan (Kamariah Tambunan, 2012), *thesaurus* menyajikan istilah yang tersusun sistematis dan saling berhubungan secara semantik dari umum hingga spesifik. Dalam pengolahan berbasis komputer, *thesaurus* dimanfaatkan dengan mengekstraksi lema dan relasi sinonim untuk membentuk struktur data terorganisasi (Dozan, 2019). Penelitian yang dilakukan oleh Dilah Kencono dan Fransiska Wiratikusuma (Kencono and Wiratikusuma, 2022), menunjukkan bahwa penggunaan *thesaurus* dapat meningkatkan efektivitas temu balik informasi melalui perluasan kata kunci pencarian, mengingat perbedaan pilihan kata antara pengguna dan penulis (Paiki, 2012).

Pengurutan data merupakan salah satu operasi dasar yang sering digunakan dalam berbagai aplikasi komputer, seperti pencarian, penyimpanan, dan pemrosesan data (Yusuf *et al.*, 2024). Dalam sistem temu balik informasi, *bubble sort* dapat dimanfaatkan untuk mengurutkan hasil pencarian berdasarkan tingkat relevansi atau bobot kemunculan data. Pengurutan ini punya keunggulan yaitu mekanisme yang sederhana sehingga mudah dipahami, karena ide dasar dari algoritma ini adalah membandingkan setiap elemen dalam *array*, dan jika urutannya salah, maka akan ditukar hingga tidak diperlukan lagi pertukaran (Ghifari, 2025).

2. METODE PENELITIAN

Jenis penelitian yang digunakan adalah penelitian eksperimen, yang di mana bertujuan untuk menguji bagaimana penerapan konsep *thesaurus* pada sistem temu balik informasi berbasis web yang dapat meningkatkan akurasi dan relevansi hasil pencarian dengan pengurutan *bubble sort*. Pada teknik pengumpulan data, data yang digunakan dalam sistem diperoleh melalui pengumpulan informasi artis dan band internasional yang umum dicari oleh pengguna, yang di mana setiap entri data terdiri dari nama artis atau band, jenis entitas, deskripsi singkat dari data terkait, serta daftar kata kunci terkait yang berfungsi sebagai perluasan makna pencarian.

2.1 Ekspansi Query

Ekspansi *query* dengan analisis lokal hanya menggunakan *query* dan dokumen yang sudah ditemu dikembalikan pada pencarian awal. Pada tahap awal, pengguna memasukkan sebuah *query* awal yang dinyatakan sebagai himpunan kata kunci. *Query* ini kemudian diproses oleh sistem untuk dilakukan perluasan menggunakan *thesaurus*. *Query* awal akan ditambahkan dengan beberapa *term* atau kata pada *query* untuk memberikan kemudahan dalam proses *information retrieval*. Teknik ini digunakan untuk memodifikasi *query* agar memenuhi kebutuhan informasi yang dibutuhkan oleh *user*. Proses penambahan *query* dengan mencari sinonim dalam bentuk *unstemmed-term* dari *query* (Dwi Laxmi and Ali Fauzi, 2019). Secara matematis, *query* awal dapat dinyatakan sebagai:

$$Q = \{q_1, q_2, q_3, \dots, q_n\}$$

Keterangan:

Q = Query awal

q_n = Kata ke – n dalam *query*

Setiap kata kunci q_1 memiliki sekumpulan kata sinonim atau istilah terkait yang diperoleh dari *thesaurus*, yang dinyatakan sebagai:

$$T(q_i) = \{s_1, s_2, \dots, s_m\}$$

Keterangan:

$T(q_i)$ = Himpunan istilah dari kata q_i

s_m = Sinonim atau kata yang berhubungan

Hasil dari proses *query expansion* adalah *query* baru yang merupakan gabungan antara *query* awal dan seluruh istilah hasil *thesaurus*, sehingga dapat dituliskan sebagai:

$$Q' = Q \cup T(q_1) \cup T(q_2) \cup \dots \cup T(q_n)$$

Keterangan:

Q' = Query hasil ekspansi

\cup = Operasi gabungan (union)

Setelah proses perluasan *query*, sistem memberikan bobot yang berbeda pada kata kunci asli dan kata kunci hasil *thesaurus*. Bobot ini bertujuan untuk menjaga agar kata kunci utama tetap memiliki pengaruh yang lebih besar dibandingkan kata kunci tambahan. Bobot *term* didefinisikan sebagai:

$$w'(t) = 1 \text{ jika } t \in Q, \alpha \text{ jika } t \in T(Q)$$

Keterangan:

$w'(t)$ = Bobot *term*

t = Kata(*term*)

α = Bobot kata hasil *thesaurus* ($0 < \alpha < 1$)

Nilai merupakan konstanta dengan nilai kurang dari satu, yang menyatakan tingkat kontribusi istilah hasil *thesaurus* terhadap relevansi pencarian. Dengan pembobotan ini, sistem tetap memprioritaskan kata kunci asli, namun tetap mempertimbangkan kata kunci yang memiliki hubungan makna.

2.2 Bubble Sort

Algoritma *bubble sort* merupakan urutan atau struktur yang diterapkan pada bahasa komputer atau pemrograman untuk membantu memecahkan masalah yang datanya tersedia sebagai input dan output sebagai hasil dari proses yang dijalankan (Abdullah *et al.*, 2023). Cara kerja algoritma *bubble sort* adalah mengulang proses, melakukan perbandingan antara setiap elemen *array* dan melakukan penggantian posisi jika urutannya sudah benar, memproses elemen dari kiri ke kanan (Rizki Saputra, Andryana and Sholihati, 2021), setiap elemen yang berdekatan nilainya akan dibandingkan dan prosesnya akan terus berulang sampai telah sesuai dengan urutannya. Pengujian data pada algoritma ini akan sangat efisien jika jumlah datanya sedikit (Andi Astuti, 2023),

sedangkan jika jumlah datanya besar, pengujian ini akan memakan waktu yang lama karena merupakan algoritma sederhana.

3. HASIL DAN PEMBAHASAN

Pada penelitian kali ini, sistem temu balik informasi berbasis web menyimpan data artikel tentang artis dan band internasional.

Tabel 1. Dataset Artis/Band Internasional

ID	Nama Artis/Band	Deskripsi
D1	Taylor Swift	Pop singer and songwriter
D2	One Direction	British pop boy band
D3	Ariana Grande	Pop and R&B Singer
D4	Coldplay	British rock band
D5	Justin Bieber	Pop singer and artist

Lalu pada studi kasus ini, pengguna memasukkan *query* pencarian berupa: “Justin Bieber”, maka:

Query awal dinyatakan sebagai himpunan kata kunci

$$Q = \{q_1, q_2, q_3, \dots, q_n\}$$

$$Q = \{justin, beiber\}$$

Setiap kata dalam *query* memiliki istilah lain yang maknanya berhubungan.

$$T(q_i) = \{s_1, s_2, \dots, s_m\}$$

$$T(justin\ beiber) = \{artist, pop, singer\}$$

Query diperluas menjadi:

$$Q' = Q \cup T(q_1) \cup T(q_2) \cup \dots \cup T(q_n)$$

$$Q' = \{justin, beiber, artist, pop, singer\}$$

Bobot diberikan untuk membedakan kepentingan kata asli dan kata hasil *thesaurus*.

$$w'(t) = 1 \text{ jika } t \in Q, \alpha \text{ jika } t \in T(Q)$$

yang di mana, pada penelitian ini nilai α yang digunakan adalah 1, maka:

Tabel 2. Pembobotan Dokumen D1 (Taylor Swift)

Term	Bobot
Justin	0
Bieber	0
Artist	0
Pop	1
singer	1

Tabel 3. Pembobotan Dokumen D2 (One Direction)

Term	Bobot
Justin	0
Bieber	0
Artist	0
Pop	1
singer	0

Tabel 4. Pembobotan Dokumen D3 (Arian Grande)

Term	Bobot
Justin	0
Bieber	0

Artist	0
Pop	1
singer	1

Tabel 5. Pembobotan Dokumen D4 (Coldplay)

<i>Term</i>	Bobot
Justin	0
Bieber	0
Artist	0
Pop	0
singer	0

Tabel 6. Pembobotan Dokumen D5 (Justin Bieber)

<i>Term</i>	Bobot
Justin	1
Bieber	1
Artist	1
Pop	1
singer	1

Tabel 7. Rekap Bobot Dokumen

Dokumen	Bobot
D1	2
D2	1
D3	2
D4	0
D5	5

Selanjutnya adalah pengurutan dengan menggunakan *bubbel sort*

$$W_d(i) < W_d(i + 1)$$

Dengan menggunakan data dokumen bobot, maka:

$$\{D1=2, D2=1, D3=2, D4=0, D5=5\}$$

Iterasi I = {D1=2, D3=2, D2=1, D5=5, D4=0}

Iterasi II = {D1=2, D3=2, D5=5, D2=1, D4=0}

Iterasi III = {D1=2, D5=5, D3=2, D2=1, D4=0}

Iterasi IV = {D5=5, D1=2, D3=2, D2=1, D4=0}

Tabel 8. Hasil Akhir Perankingan Bobot Dokumen

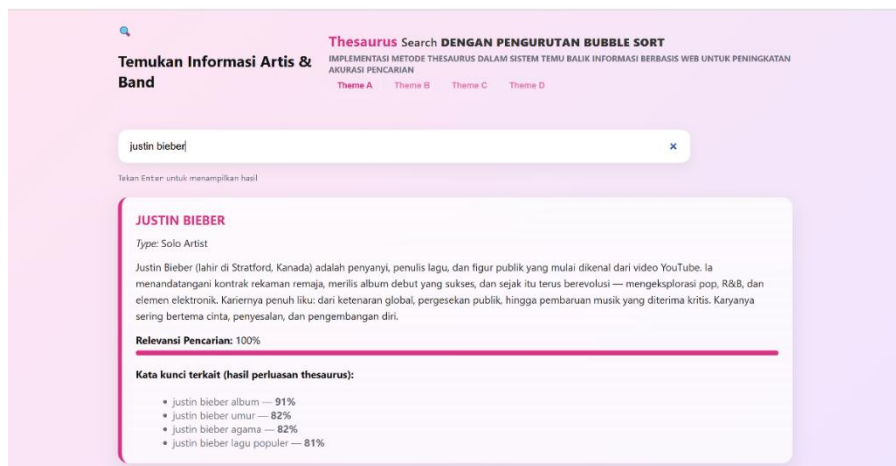
Peringkat	Artis/Band	Bobot
D5	Justin Bieber	5
D1	Taylor Swift	2
D3	Ariana Grande	2
D2	One Direction	1
D4	Coldplay	0

Berdasarkan hasil perhitungan bobot yang dilakukan, setiap dokumen memperoleh nilai bobot akhir yang menunjukkan tingkat relevansinya terhadap *query* pencarian “Justin Bieber”. Nilai bobot ini diperoleh dari kecocokan antara kata kunci pada *query* awal dan hasil perluasan *query* dengan menggunakan metode *thesaurus* dengan *term* yang terdapat pada masing-masing dokumen.

3.1 Implementasi Sistem

Sistem temu balik informasi diimplementasikan dalam bentuk aplikasi web dengan menggunakan PHP sebagai bahasa pemrograman dan dijalankan pada XAMPP sebagai

server lokal, sementara proses pengembangan dilakukan menggunakan Visual Studio Code. Tampilan sistem dirancang menggunakan HTML dan CSS, serta didukung JavaScript untuk mengelola interaksi pengguna seperti saran pencarian dan eksekusi pencarian. Metode *thesaurus* digunakan untuk memperluas kata kunci pencarian, sedangkan algoritma *bubble sort* diterapkan untuk mengurutkan hasil pencarian berdasarkan nilai relevansi yang ditampilkan dalam bentuk persentase, sehingga hasil yang diperoleh lebih akurat dan mudah dipahami.



Gambar 1. Tampilan Hasil Pencarian

Tampilan hasil pencarian pada sistem menunjukkan deskripsi utama dari kata kunci yang dicari oleh pengguna. Di bagian bawah hasil pencarian, sistem menampilkan saran kata kunci terkait yang diperoleh melalui metode *thesaurus* sebagai bentuk perluasan istilah. Setiap saran pencarian dilengkapi dengan nilai bobot dalam bentuk persentase yang merepresentasikan tingkat relevansi terhadap kata kunci awal. Nilai bobot tersebut kemudian diurutkan menggunakan algoritma *bubble sort* dari persentase tertinggi hingga terendah, sehingga saran pencarian yang paling relevan ditampilkan pada urutan teratas. Dengan mekanisme ini, tampilan sistem tidak hanya menyajikan hasil pencarian, tetapi juga menunjukkan hubungan keterkaitan antar kata kunci secara terstruktur dan mudah dipahami oleh pengguna.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penerapan metode *thesaurus* dalam sistem temu balik informasi mampu memperluas kata kunci pencarian sehingga hubungan antar istilah menjadi lebih terstruktur dan relevan. Penggunaan algoritma *bubble sort* berperan dalam mengurutkan hasil pencarian berdasarkan bobot relevansi yang dihasilkan dari proses perluasan kata kunci tersebut, sehingga informasi dengan tingkat kecocokan tertinggi ditampilkan pada urutan teratas. Kombinasi antara metode *thesaurus* dan algoritma *bubble sort* menghasilkan tampilan hasil pencarian yang lebih sistematis, mudah dipahami, serta membantu pengguna dalam memperoleh informasi yang sesuai dengan kebutuhan pencarian.

5. DAFTAR PUSTAKA

Abdullah, M.F. *et al.* (2023) "Penggunaan Algoritma Bubble Sort dalam Pengurutan Nomor Induk Mahasiswa," *Hello World Jurnal Ilmu Komputer*, 2(1), pp. 14–19. Available at:

- <https://doi.org/10.56211/helloworld.v2i1.206>.
- Andi Astuti, Y. (2023) "Analisis Pengujian Data Algoritma Bubble Sort," *Remik: Riset dan E-Jurnal Manajemen Informatika Komputer*, 7(3), pp. 1413–1420. Available at: <http://doi.org/10.33395/remik.v7i3.12594>.
- Azzahrawaani, Z. and Agustina, S. (2023) "Pemanfaatan Tesaurus dalam Pembelajaran : Sebuah," *Jurnal Pustaka Ilmiah*, 9(2), pp. 59–66. Available at: <https://jurnal.uns.ac.id/jurnalpustakailmiah/article/view/75001>.
- BRUNDY, R. (2019) "Pengembangan Sistem Temu Kembali Berbasis Web Menggunakan Algoritma Tolerance Value Generator," *Katalog.Ukdw.Ac.Id* [Preprint]. Available at: http://katalog.ukdw.ac.id/id/eprint/515%0Ahttps://katalog.ukdw.ac.id/515/1/71150004_bab1_bab5_daftarpustaka.pdf.
- Dozan, D.J. (2019) "Pengembangan Sistem Information Retrieval Untuk Bahasa Indonesia Berbasis Web Menggunakan Vector Space Model," pp. 1–33. Available at: https://katalog.ukdw.ac.id/443/1/71130027_bab1_bab5_daftarpustaka.pdf.
- Dwi Laxmi, M. and Ali Fauzi, M. (2019) "Query Expansion Pada Sistem Temu Kembali Informasi Berbahasa Indonesia Dengan Metode Pembobotan TF-IDF Dan Algoritme Cosine Similarity Berbasis Wordnet," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(1), pp. 2548–964. Available at: <http://j-ptiik.ub.ac.id>.
- Ghifari, H. Al (2025) "Penggunaan algoritma bubble sort pada Bahasa pemrograman Java," 3, pp. 681–687.
- Hasan, M.I. (2018) "Information Retrieval System artikel kesehatan menggunakan pembobotan tf. idf dan Latent Semantic Indexing." Available at: <http://etheses.uin-malang.ac.id/12546/>.
- Informatika, T. et al. (2004) "Pemilihan Tesaurus Online," pp. 1–4.
- Kamariah Tambunan (2012) "Tesaurus Bioteknologi: Sebagai Alat Bantu Pengindeksan Dokumen," *Baca: Jurnal Dokumentasi Dan Informasi*, 33(2). Available at: <https://doi.org/http://dx.doi.org/10.14203/j.baca.v33i2.99>.
- Kencono, D. and Wiratikusuma, F. (2022) "Tesaurus Sebagai Sarana Pendukung Pembelajaran Kosakata Bahasa Indonesia," *Jurnal Pendidikan Bahasa Indonesia*, 10(1), p. 13. Available at: <https://doi.org/10.30659/jpbi.10.1.13-25>.
- Paiki, F.F. (2012) "Evaluasi penggunaan Similarity Thesaurus terhadap ekspansi kueri dalam sistem temu kembali informasi berbahasa Indonesia," *Fridolin Paiki-Paper Evaluasi Penggunaan Similarity ...*, (March), pp. 20–29. Available at: https://www.researchgate.net/profile/Fridolin-Paiki/publication/296638505_Evaluasi_Penggunaan_Similarity_Thesaurus_Terhadap_Ekspansi_Kueri_pada_Sistem_Temu_Kembali_Informasi_Berbahasa_Indonesia/links/56d7074308aee1aa5f75c1fd/Evaluasi-Penggunaan-Similarity.
- Rizki Saputra, D.Y., Andryana, S. and Sholihati, I.D. (2021) "Analisis Perbandingan Algoritma Bubble Sort Dan Selection Sort Pada Sistem Pendukung Keputusan Pemilihan Tempat Kost Berbasis Ios (Iphone Operating System)," *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 6(2), pp. 318–324. Available at: <https://doi.org/10.29100/jupi.v6i2.2015>.
- Yusuf, A. et al. (2024) "Analisis Algoritma Bubble Sort Ascending / Descending dan Implementasinya Menggunakan Bahasa Pemrograman Python," 2, pp. 53–57.
- Zamzami, D., Rahutomo, F. and ... (2016) "Aplikasi wordnet indonesia berdasarkan kamus thesaurus bahasa indonesia menggunakan algoritma rule based text parsing," *Seminar Informatika ...* [Preprint]. Available at: <http://jurnalti.polinema.ac.id/index.php/SIAP/article/download/42/40>.