

## 5. PENGUJIAN SISTEM

Pengujian sistem menunjukkan beberapa hal, yaitu hasil *knowledge graph*, pengujian kecepatan pembuatan vektor, pengujian efektivitas filter properti Wikidata, pengujian tiga algoritma *Jaccard Similarity*, *Euclidean Distance*, dan *Cosine Similarity*, perbandingan dengan hasil sistem perpustakaan yang digunakan saat ini, perbandingan metrik antar kata kunci, serta analisis hasil pencarian.

### 5.1 Knowledge Graph

Setiap koleksi memiliki atribut-atribut yang masing-masing terhubung dengan atribut lain melalui hubungan yang terdefinisi dalam *graph*. *Class Collection* terhubung dengan *class Creator* melalui relasi *createdBy*, *class Contributor* melalui relasi *contributedBy*, *class Theme* melalui relasi *hasTheme*, *class Category* melalui relasi *hasCategory*, *class Subcategory* melalui relasi *hasSubcategory*, *class Subject* melalui relasi *hasSubject*, dan *class Keyword* melalui relasi *hasKeyword*. *Class Keyword* terhubung dengan *class WikidataKeyword* melalui relasi *relatedTo*.

*Node Keyword*, *WikidataKeyword*, serta relasi antara keduanya diambil dari Wikidata. Data koleksi dan Wikidata diintegrasikan melalui hubungan *hasKeyword* antara *node Collection* dan *node Keyword*, dilanjutkan dengan hubungan antara *node Keyword* dan *WikidataKeyword* melalui relasi *relatedTo*. Relasi *relatedTo* antara *node Keyword* dan *WikidataKeyword* memiliki subrelasi yang menunjukkan hubungan lebih spesifik. Subrelasi yang menghubungkan antar *keyword* itu total berjumlah 1070 dan total *related keywords* sebanyak 144021. Contoh beberapa subrelasi yang ada dapat dilihat pada Tabel 5.1. Tabel tersebut menunjukkan 20 subrelasi terbanyak serta jumlah subrelasi tersebut digunakan sebagai relasi antara dua *node keyword*. Dengan relasi ini, data koleksi dan entitas terkait dapat terkoneksi sehingga dapat memberikan kemampuan untuk memperkaya pencarian.

Tabel 5.1

Tabel Contoh Properti Wikidata

ID	Nama Properti	Jumlah Properti
P31	instance of	138802
P279	subclass of	106792
P1343	described by source	80695
P1889	different from	77074

P910	topic's main category	44023
P527	has part(s)	43033
P361	part of	30456
P5008	on focus list of Wikimedia project	30004
P407	language of work or name	25113
P1552	has characteristic	24683
P2860	cites work	23443
P1433	published in	22859
P460	said to be the same as	22512
P17	country	21843
P921	main subject	18305
P1424	topic's main template	17575
P150	contains the administrative territorial entity	17502
P2579	studied in	15252
P50	author	13683
P366	has use	13528

Contoh representasi *graph* untuk salah satu koleksi dapat dilihat pada Gambar 5.1. *Node* berwarna merah menunjukkan *class Collection*, *node* berwarna hijau cerah menunjukkan *class Subject*, *node* berwarna ungu lavender menunjukkan *class Subcategory*, *node* berwarna pink menunjukkan *class Theme*, *node* berwarna biru menunjukkan *class Contributor*, *node* berwarna oranye menunjukkan *class Creator*, *node* berwarna ungu kebiruan menunjukkan *class Category*, *node* berwarna hijau lime menunjukkan *class Keyword*, dan *node* berwarna tosca menunjukkan *keyword* dari Wikidata.

Dalam contoh tersebut ada koleksi dengan ID 9018 terhubung dengan *node-node* seperti *creator*, *contributor*, *theme*, *subject*, *category*, *subcategory* yang memberikan informasi terkait koleksi tersebut. Koleksi ini memiliki relasi *createdBy* dengan *node* berwarna oranye yang menunjukkan nama pencipta koleksi. Koleksi tersebut juga terhubung dengan *contributor* dengan *node* berwarna biru melalui relasi Pembimbing dan Penguji. Kedua relasi tersebut merupakan anak dari relasi *contributedBy* yang menunjukkan adanya pihak-pihak dengan peran khusus yang terlibat dalam koleksi tersebut. Terdapat juga relasi dengan *node theme*, *category*,



Lima *keyword* telah ditentukan yang digunakan dalam pengujian *search engine*. Lima *keyword* ini terdiri dari “Artificial Intelligence”, “Furniture Design”, “Purchase Intention”, “Mass Communication”, dan “Consumer Satisfaction”. Mahasiswa jurusan yang sesuai dengan bidang kata kunci tersebut diberikan 1000 data judul dan deskripsi koleksi. Lalu, mahasiswa tersebut sebagai evaluator memberikan nilai dalam skala 0-10 untuk menunjukkan tingkat relevansi antara *keyword* dan setiap koleksi. Masing-masing *keyword* dinilai oleh 3 evaluator, sehingga total ada sebanyak 15 evaluator untuk semua *keyword* yang diuji. 1000 data yang digunakan merupakan koleksi digital S1 yang diambil secara acak. Koleksi digital S1 dipilih sebagai fokus dalam skripsi ini karena jenis koleksi ini merupakan yang paling sering dicari oleh pengguna. Selain itu, pengujian dilakukan pada 1000 data karena setiap koleksi harus dinilai secara manual oleh evaluator dan jika seluruh data koleksi digital S1 yang berjumlah sekitar 30.000 digunakan dalam pengujian, proses evaluasi ini akan menjadi sangat tidak praktis dan memakan waktu lama. Selain itu, dengan jumlah data yang terlalu banyak, terdapat risiko bahwa penilaian relevansi menjadi tidak konsisten atau kurang akurat akibat kelelahan evaluator. Oleh karena itu, diputuskan untuk menggunakan 1000 data sebagai sampel representatif yang memungkinkan evaluasi dilakukan dengan lebih efisien dan tetap menjaga kualitas penilaian. Hasil dari penilaian tersebut menjadi *ground truth* yang merupakan acuan untuk mengukur tingkat kesesuaian antara hasil yang dihasilkan algoritma dengan relevansi sebenarnya.

### 5.3 Perbandingan Kecepatan Pembuatan Vektor

Pengujian ini dilakukan untuk membandingkan kecepatan proses pembuatan vektor menggunakan dua cara berbeda. Cara pertama yaitu dengan melakukan query SPARQL langsung ke GraphDB setiap kali pencarian pertama dilakukan. Cara kedua yaitu dengan menyimpan hasil query SPARQL dari GraphDB ke database MySQL sebagai penyimpanan bantuan, yang kemudian diakses saat pencarian pertama dilakukan. Data yang diambil dari GraphDB untuk disimpan di database MySQL adalah *keyword*, *wikidata keyword*, *subject*, *creator*, dan *contributor*. Hasil pengujian dengan 1000 data menunjukkan bahwa pembuatan vektor menggunakan cara pertama membutuhkan waktu sebanyak 1.736 detik, sedangkan cara kedua hanya memerlukan waktu sebanyak 0.356 detik. Perbedaan ini semakin mencolok seiring bertambahnya jumlah data, seperti terlihat pada 5000 data di mana metode pertama memerlukan 70.640 detik, sementara metode kedua hanya membutuhkan 2.227 detik. Berdasarkan pengujian ini, didapatkan bahwa cara kedua jauh lebih cepat untuk pembuatan vektor meskipun sedikit lebih tidak efektif. Detail perbandingan waktu untuk kedua metode dapat dilihat pada Tabel 5.2, yang memperlihatkan hasil pengujian pada jumlah data yang berbeda.

Tabel 5.2

Tabel Perbandingan Kecepatan Pembuatan Vektor

Jumlah Data	Cara Pertama (detik)	Cara Kedua (detik)
1000	13.736	0.356
2500	29.020	1.013
5000	70.640	2.227

#### 5.4 Pengujian Filter Properti Wikidata

Wikidata memiliki banyak properti yang mewakili hubungan antara satu kata dengan kata lain. Dalam 1000 data koleksi yang digunakan, terdapat 685 properti yang berbeda. Dilakukan perhitungan jumlah properti menggunakan query seperti pada Gambar 5.2 dan hasilnya dapat dilihat pada Gambar 5.3 ditampilkan sepuluh properti Wikidata yang digunakan terbanyak pada data yang ada. Dapat dilihat, terdapat properti “described by source” yang menghubungkan suatu kata dengan sumber dokumen yang memberikan penjelasan lebih lanjut. Contohnya terdapat entitas dengan value “design” memiliki relasi “described by source” dengan entitas dengan value “Encyclopædia Britannica 11th edition” yang dimana menunjukkan yang menunjukkan bahwa “design” dijelaskan atau dirujuk secara khusus dalam dokumen tersebut sebagai sumber referensi. Namun, properti seperti ini tidak sesuai dengan tujuan penggunaan Wikidata dalam skripsi ini, yaitu untuk memperluas makna atau hubungan semantik dari suatu kata. Maka dari itu, diidentifikasi beberapa properti yang dianggap kurang cocok dan dapat dilihat pada Tabel 5.3.

Pada Gambar 5.4, ditampilkan contoh pemetaan keyword melalui properti “instance of”. Dari gambar tersebut dapat dilihat beberapa *keyword* seperti “acquisition price” dan “consumer loyalty” yang terhubung dengan *keyword* “scholarly article”, namun *keyword* tersebut cenderung kurang memberikan informasi yang relevan untuk memperluas konsep karena lebih bersifat deskriptif dibandingkan memberikan hubungan semantik yang signifikan. Dan setelah diidentifikasi, ditemukan bahwa cukup banyak *keyword* yang terhubung dengan “scholarly article”. Oleh karena itu, diterapkan juga filter secara menyeluruh untuk semua properti dengan target entitas “scholarly article”.

```

select ?property ?property_label (count(?property) as ?property_count) where {
  ?a ?property ?b .
  ?property rdfs:label ?property_label .
  ?b a lib:WikidataKeyword .
  filter (?p ≠ lib:relatedTo )
}
group by ?property ?property_label
order by desc(?property_count)
limit 20

```

Gambar 5.2 Query SPARQL Menghitung Jumlah Properti

	property	property_label	property_count
1	wdt:P31	"instance of"	"7644"
2	wdt:P279	"subclass of"	"5935"
3	wdt:P1889	"different from"	"4204"
4	wdt:P1343	"described by source"	"4116"
5	wdt:P910	"topic's main category"	"2430"
6	wdt:P527	"has part(s)"	"2257"
7	wdt:P5008	"on focus list of Wikimedia project"	"1718"
8	wdt:P361	"part of"	"1689"
9	wdt:P2860	"cites work"	"1371"
10	wdt:P1433	"has characteristic"	"1306"
11	wdt:P1433	"published in"	"1280"
12	wdt:P460	"said to be the same as"	"1202"
13	wdt:P921	"main subject"	"1110"
14	wdt:P17	"country"	"1044"
15	wdt:P407	"language of work or name"	"970"
16	wdt:P1424	"topic's main template"	"906"
17	wdt:P50	"author"	"821"
18	wdt:P150	"contains the administrative territorial entity"	"818"
19	wdt:P2579	"studied in"	"809"
20	wdt:P366	"has use"	"697"

Gambar 5.3 Hasil Query 20 Properti Dengan Jumlah Terbanyak

	keyword	property	wikidata_keyword
1	"accordance"	"instance of"	"Biblical software"
2	"acquisition price"	"instance of"	"scholarly article"
3	"adequate assets"	"instance of"	"economics terminology"
4	"borland delphi"	"instance of"	"file format"
5	"businesses"	"instance of"	"Wikinews article"
6	"celebrity endorsers"	"instance of"	"master's thesis"
7	"classical assumption test"	"instance of"	"scholarly article"
8	"classical assumption tests"	"instance of"	"comic strip"
9	"communication"	"instance of"	"type of process"
10	"communication"	"instance of"	"type of interaction"
11	"communication"	"instance of"	"intentional human activity"
12	"communication"	"instance of"	"social skill"
13	"communication"	"instance of"	"social behavior"
14	"companies"	"instance of"	"company register"
15	"companies"	"instance of"	"registration authority"
16	"companies"	"instance of"	"Executive agency"
17	"consumer loyalty"	"instance of"	"scholarly article"

Gambar 5.4 Contoh Pemetaan Keyword Melalui Properti Instance Of

Tabel 5.3

Tabel Properti Wikidata yang Tidak Sesuai

Properti	Keterangan	Contoh
Publisher (P123)	Penerbit dari suatu karya	Advertisements - Publisher - Macmillan Publishers
Described by source (P1343)	Sumber yang digunakan untuk mendeskripsikan suatu entitas	Data Processing - Described by source - Open Science Thesaurus
Category for people who died here (P1465)	Kategori orang-orang yang meninggal di suatu tempat tertentu	Jakarta - Category for people who died here - Category:Deaths in Jakarta
Given name version for other gender (P1560)	Nama yang digunakan untuk versi jenis kelamin lain dari nama seseorang	Christian - Given name version for other gender - Krista
Driving side (P1622)	Menunjukkan sisi jalan tempat kendaraan seharusnya dikendarai	Indonesia - Driving side - Left
Name day (P1750)	Hari yang diperingati sebagai hari nama, seringkali berhubungan dengan kalender agama atau budaya.	Rose - Name day - August 23
Twinned administrative body (P190)	Merujuk pada badan administrasi yang memiliki hubungan kembar (sister city, kota atau wilayah yang memiliki hubungan)	Semarang - Twinned administrative body - Samarinda
Highest judicial authority (P209)	Otoritas pengadilan tertinggi di suatu negara atau wilayah	Indonesia - Highest judicial authority - Supreme Court of the Republic of Indonesia
Genetic association (P2293)	Hubungan antara suatu entitas atau sifat dengan genetik, seperti mutasi atau pola pewarisan	Depression - Genetic association - CACNA1C
Writing system (P282)	Sistem tulisan yang digunakan untuk merepresentasikan bahasa tertentu	Sales - Writing system - Latin script

Instance of (P31)	Menunjukkan jenis atau kategori umum dari entitas	Consumer loyalty - Instance of - Scholarly article
Has tense (P3103)	Menunjukkan hubungan dengan bentuk waktu tertentu dalam bahasa	Studying german - Has tense - Perfect tense
Shares border with (P47)	Menunjukkan entitas atau wilayah yang berbatasan dengan entitas lain	South surabaya - Shares border with - Sidoarjo
On focus list of Wikimedia project (P5008)	Daftar proyek Wikimedia yang difokuskan	Data analysis - On focus list of Wikimedia project - WikiProject Zika Corpus
Has grammatical gender (P5109)	Jenis kelamin gramatikal	Spa - Has grammatical gender - Feminine
Head of government (P6)	Pemimpin eksekutif suatu negara atau wilayah	Surabaya - Head of government - Eri Cahyadi
Level of description (P6224)	Tingkat kedalaman atau rincian dalam dokumentasi atau deskripsi suatu entitas	Employee work - Level of description - Series
Access restriction status (P7228)	Status pembatasan akses terhadap suatu entitas	Applications - Access restriction status - Unrestricted access
Use restriction status (P7261)	Pembatasan penggunaan entitas	Tax invoices - Use restriction status - Unrestricted access
Given name (P735)	Nama depan seseorang	Fuzzy - given name - AI
Significant event (P793)	Peristiwa penting yang terkait dengan entitas atau kejadian sejarah	Conflict handling - Significant event - Filing
Indexed in bibliographic review (P8875)	Ulasan bibliografi atau referensi pustaka	Machines - Indexed in bibliographic review - Scopus

Dari properti-properti yang ada diatas, dilakukan perbandingan hasil pencarian menggunakan filter properti diatas dan tanpa menggunakan filter. Perbandingan dilakukan dengan menggunakan 5 kata kunci sebagai input yaitu “Artificial Intelligence”, “Furniture

Design”, “Purchase Intention”, “Mass Communication”, “Consumer Satisfaction”. Pengujian ini dilakukan menggunakan dua metode yaitu *Jaccard Similarity* dan *Euclidean Distance* dengan evaluasi hasil pencarian berdasarkan beberapa metrik yaitu *Precision@k*, *Precision*, *Recall*, *F1-Score*, dan MAP.

Hasil dari perbandingan dapat dilihat pada Tabel 5.4 yang dimana menunjukkan bahwa penggunaan filter pada properti Wikidata menghasilkan peningkatan performa pada metode *Jaccard Similarity*. Hasil *precision@k* tetap stabil pada beberapa *query* seperti *Purchase Intention*, dan *Consumer Satisfaction*, namun meningkat pada *query Furniture Design* dan *Mass Communication*. Hal itu menunjukkan bahwa *filtering* membantu memperbaiki *Precision@K* dengan cara mengurangi *noise* yang dihasilkan oleh properti atau relasi yang kurang relevan. Metrik yang lain juga mengalami peningkatan yang cukup signifikan, kecuali metrik *recall* yang sedikit menurun. Penurunan *recall* ini kemungkinan terjadi karena koleksi yang sebelumnya dianggap relevan mungkin tidak lagi terdeteksi karena hilangnya relasi yang tidak sesuai dengan kriteria filter. Akibatnya, meskipun kualitas hasil pencarian teratas meningkat, cakupan keseluruhan koleksi relevan yang ditemukan menjadi lebih terbatas, sehingga *recall* sedikit menurun. Meskipun *recall* sedikit menurun, hal ini dapat dianggap sebagai kompromi yang dapat diterima, mengingat tujuan utama dari sistem pencarian ini adalah untuk meningkatkan kualitas hasil pencarian teratas.

Sedangkan pada metode *Euclidean Distance* yang dapat dilihat pada Tabel 5.5, hasil perbandingan menunjukkan bahwa penggunaan filter tidak memberikan dampak yang signifikan, bahkan cenderung menurun. Sebagian besar nilai *Precision@k* tetap nol atau tidak berubah, seperti pada *query Artificial Intelligence* dan *Purchase Intention*. Sedangkan, pada *query* sedangkan untuk *query Consumer Satisfaction*, nilai *Precision@20* bahkan turun dari 0.050 menjadi 0. Namun, nilai *Precision@k* pada *query Furniture Design* dan *Mass Communication* sedikit meningkat walaupun tidak signifikan.

Secara keseluruhan, penggunaan filter properti Wikidata ini mampu meningkatkan skor kemiripan di kedua algoritma. Namun, perbandingan ini menunjukkan bahwa *filtering* properti Wikidata sangat efektif untuk metode *Jaccard Similarity* karena dapat meningkatkan relevansi hasil pencarian teratas.

Tabel 5.4

Tabel Hasil Perbandingan Filter Properti Wikidata Menggunakan Metode *Jaccard Similarity*

Metrik	Tanpa Filter	Dengan Filter
Query: Artificial Intelligence		

Precision@20	0.583	0.583
Precision@10	0.633	0.567
Precision	0.021	0.126
Recall	0.844	0.774
F1-Score	0.039	0.216
MAP	0.441	0.457
Query: Furniture Design		
Precision@20	0.450	0.567
Precision@10	0.733	0.833
Precision	0.057	0.347
Recall	0.715	0.703
F1-Score	0.106	0.452
MAP	0.429	0.535
Query: Purchase Intention		
Precision@20	0.950	0.950
Precision@10	1.000	1.000
Precision	0.059	0.082
Recall	0.944	0.712
F1-Score	0.112	0.146
MAP	0.469	0.479
Query: Mass Communication		
Precision@20	0.416	0.567
Precision@10	0.366	0.767
Precision	0.209	0.598
Recall	0.667	0.657
F1-Score	0.314	0.616
MAP	0.251	0.488
Query: Consumer Satisfaction		
Precision@20	0.750	0.750
Precision@10	0.767	0.767
Precision	0.052	0.052
Recall	0.896	0.896

F1-Score	0.098	0.098
MAP	0.561	0.582
<b>Average</b>		
Precision@20	0.630	0.683
Precision@10	0.700	0.787
Precision	0.079	0.241
Recall	0.813	0.749
F1-Score	0.134	0.306
MAP	0.430	0.508

Tabel 5.5

Tabel Hasil Perbandingan Filter Properti Wikidata Menggunakan Metode *Euclidean Distance*

Metrik	Sebelum	Sesudah
Query: Artificial Intelligence		
Precision@20	0.000	0.000
Precision@10	0.000	0.000
Precision	0.021	0.016
Recall	0.294	0.236
F1-Score	0.038	0.030
MAP	0.052	0.014
Query: Furniture Design		
Precision@20	0.016	0.017
Precision@10	0.033	0.033
Precision	0.018	0.022
Recall	0.215	0.266
F1-Score	0.033	0.040
MAP	0.024	0.024
Query: Purchase Intention		
Precision@20	0.000	0.000
Precision@10	0.000	0.000
Precision	0.075	0.076
Recall	0.361	0.368

F1-Score	0.124	0.125
MAP	0.044	0.044
Query: Mass Communication		
Precision@20	0.100	0.100
Precision@10	0.033	0.067
Precision	0.029	0.030
Recall	0.134	0.134
F1-Score	0.047	0.049
MAP	0.088	0.088
Query: Consumer Satisfaction		
Precision@20	0.050	0.000
Precision@10	0.000	0.000
Precision	0.072	0.321
Recall	0.504	0.541
F1-Score	0.126	0.141
MAP	0.027	0.026
<b>Average</b>		
Precision@20	0.033	0.023
Precision@10	0.013	0.020
Precision	0.043	0.093
Recall	0.302	0.309
F1-Score	0.074	0.077
MAP	0.047	0.039

### 5.5 Perbandingan Algoritma Jaccard Similarity, Euclidean Distance, dan Cosine Similarity

Dalam praktiknya, pengguna biasanya hanya memperhatikan hasil pada urutan awal dari hasil pencarian. Sehingga metrik *Precision@K* menjadi fokus utama karena sistem pencarian dirancang untuk memberikan hasil yang relevan di posisi teratas kepada pengguna. Berdasarkan hasil rata-rata pengujian dari tiga evaluator yang terdapat pada Tabel 5.6, terlihat bahwa *Jaccard Similarity* secara konsisten menunjukkan performa yang lebih baik dibandingkan dengan *Euclidean Distance* dan *Cosine Similarity* dalam berbagai metrik evaluasi. Nilai rata-rata *Precision@K* pada *Jaccard Similarity* jauh lebih tinggi daripada kedua algoritma lainnya, bahkan ada *query* yang mencapai nilai 1. Selain itu, nilai MAP dengan rata-rata 0.508 menunjukkan

bahwa koleksi relevan yang ditemukan oleh *Jaccard Similarity* tersebar secara merata di posisi strategis dalam hasil pencarian. Dengan kata lain, algoritma ini tidak hanya mencakup koleksi relevan tetapi juga berhasil memprioritaskannya untuk muncul lebih awal di daftar hasil pencarian.

Di sisi lain, *Euclidean Distance* menunjukkan performa yang paling rendah dalam semua metrik, terutama pada *Precision@K*, yang sering kali memiliki nilai 0 untuk banyak kata kunci. Hanya pada kata kunci "Furniture Design" dan "Mass Communication", nilai *Precision@K* lebih dari nol yang mungkin dikarenakan kata kunci tersebut sedikit lebih umum daripada kata kunci lainnya walaupun tetap sangatlah rendah. Nilai rata-rata *Recall* sebesar 0.309 menunjukkan bahwa algoritma ini tidak mampu mencakup banyak koleksi relevan. Nilai MAP yang rendah mengonfirmasi bahwa koleksi relevan jarang muncul di posisi strategis. Hal ini mungkin disebabkan oleh sifat algoritma *Euclidean Distance* yang cenderung lebih sensitif terhadap perbedaan skala antar atribut dibandingkan kesamaan atribut itu sendiri. Oleh karena itu, metode ini tampaknya kurang cocok untuk masalah yang membutuhkan penilaian relevansi berbasis kesamaan atribut seperti dalam pencarian koleksi perpustakaan.

*Cosine Similarity* berada di posisi tengah, dengan rata-rata metrik lebih baik daripada *Euclidean Distance* namun tetap kalah dari *Jaccard Similarity*. Pada beberapa kata kunci tertentu, nilai rata-rata *Precision@K* sangatlah rendah. Pada kata kunci "Purchase Intention" dan "Consumer Satisfaction", nilainya cukup lebih baik namun masih jauh lebih rendah dibandingkan dengan *Jaccard Similarity*. Nilai rata-rata precision yang sedikit lebih tinggi daripada *Jaccard Similarity* menunjukkan bahwa *Cosine Similarity* sedikit lebih baik dalam menghasilkan rekomendasi yang relevan di seluruh set hasil, bukan hanya bagian awal. Hal ini dapat terjadi karena *Cosine Similarity* fokus pada kemiripan proporsional sehingga elemen yang relevan tetapi hanya sedikit berbeda dari input tidak selalu mendapatkan prioritas tinggi. Akibatnya, elemen relevan mungkin terdistribusi lebih jauh dalam daftar hasil.

Dari nilai *precision@k* serta hasil top 10 dan 20 pencarian teratas untuk masing-masing algoritma, *Jaccard Similarity* terbukti berhasil mengembalikan koleksi yang lebih relevan dibandingkan *Euclidean Distance* dan *Cosine Similarity*. Hal ini terlihat dari adanya beberapa koleksi yang relevan dengan query muncul di posisi teratas serta metrik *Precision@20* sebesar 0.683 dan *Precision@10* sebesar 0.787 yang dimana jauh lebih tinggi, mampu memprioritaskan item relevan pada urutan awal hasil pencarian. Sementara itu, *Cosine Similarity* gagal memberikan hasil yang relevan di posisi teratas. Diikuti dengan *Euclidean Distance* yang memiliki performa terburuk sehingga kurang efektif dalam memprioritaskan item relevan. Secara

keseluruhan, *Jaccard Similarity* adalah pilihan algoritma yang paling efektif untuk aplikasi pencarian yang mengutamakan relevansi hasil pada urutan awal.

Tabel 5.6

Tabel Hasil Metrik Evaluasi *Jaccard Similarity*, *Euclidean Distance*, dan *Cosine Similarity*

Kata Kunci	Algoritma	Prec@20	Prec@10	Precision	Recall	F1-Score	MAP
Artificial Intelligence	Jaccard Similarity	0.583	0.567	0.126	0.774	0.216	0.457
	Euclidean Distance	0.000	0.000	0.016	0.236	0.030	0.014
	Cosine Similarity	0.100	0.200	0.400	0.118	0.181	0.129
Furniture Design	Jaccard Similarity	0.567	0.833	0.347	0.703	0.452	0.535
	Euclidean Distance	0.017	0.033	0.022	0.266	0.040	0.024
	Cosine Similarity	0.483	0.667	0.219	0.676	0.326	0.479
Purchase Intention	Jaccard Similarity	0.950	1.000	0.082	0.712	0.146	0.479
	Euclidean Distance	0.000	0.000	0.076	0.368	0.125	0.044
	Cosine Similarity	0.583	0.833	0.493	0.482	0.481	0.420
Mass Communication	Jaccard Similarity	0.567	0.767	0.598	0.657	0.616	0.488
	Euclidean Distance	0.100	0.067	0.030	0.134	0.049	0.088
	Cosine Similarity	0.250	0.000	0.254	0.185	0.211	0.108
Consumer Satisfaction	Jaccard Similarity	0.750	0.767	0.052	0.896	0.098	0.582

	Euclidean Distance	0.000	0.000	0.321	0.541	0.141	0.026
	Cosine Similarity	0.567	0.800	0.342	0.733	0.466	0.488
<b>Average</b>	Jaccard Similarity	0.683	0.787	0.241	0.749	0.306	0.508
	Euclidean Distance	0.023	0.020	0.093	0.309	0.077	0.039
	Cosine Similarity	0.397	0.500	0.342	0.439	0.333	0.325

Untuk memberikan gambaran yang lebih jelas mengenai perbedaan performa ketiga algoritma, berikut adalah ilustrasi proses perhitungan masing-masing algoritma. Misalnya, *keyword* yang dicari adalah “artificial intelligence” dan setelah melalui proses *keyword extraction* dan mencari *related keywords* di Wikidata, ditemukan kata-kata “machine learning” dan “neural network”. Dengan demikian, elemen yang dimiliki oleh *keyword* adalah *artificial intelligence*, *machine learning*, dan *neural network*. Lalu terdapat koleksi A berjudul “Perancangan dan pembuatan aplikasi klasifikasi citra observasi bintang matahari menggunakan metode probabilistic neural network dan self-organized maps” yang telah melalui proses *keyword extraction* dan pencarian *related keywords* di Wikidata sehingga memiliki elemen *neural network* dan *classification* dan koleksi B yang berjudul “Penerapan 3D human pose estimation indoor area untuk motion capture dengan menggunakan YOLOv4-Tiny, EfficientNet Simple Baseline, dan VideoPose3D” memiliki elemen *artificial intelligence* dan *neural network*. Ilustrasi elemen-elemen *keyword* serta koleksi dapat dilihat pada Gambar 5.5.

*Keyword* = {*artificial intelligence, machine learning, neural network*}

*Koleksi A* = {*neural network, classification*}

*Koleksi B* = {*artificial intelligence, neural network*}

Gambar 5.5 Contoh Elemen *Keyword* dan Koleksi

Pada Gambar 5.6, ditunjukkan proses perhitungan nilai *Jaccard Similarity* antara elemen *keyword* dengan koleksi A dan koleksi B. Nilai *Jaccard Similarity* dihitung sebagai rasio antara jumlah elemen dalam *intersection* kedua himpunan dengan jumlah elemen dalam *union* kedua himpunan. Untuk Koleksi A, *intersection* antara elemen *keyword* dan elemen koleksi A adalah

*neural network*, sehingga jumlah elemen *intersection* adalah 1. Gabungan kedua himpunan menghasilkan *artificial intelligence*, *machine learning*, *neural network*, *machine learning*, dan *classification* dengan total jumlah elemen sebanyak 4. Oleh karena itu, nilai *Jaccard Similarity* untuk Koleksi A adalah 0,25 yang didapat dari 1 sebagai *intersection* dibagi 4 sebagai *union*. Sementara itu, untuk Koleksi B, *intersection* elemen keyword dan elemen koleksi B menghasilkan dua elemen, yaitu *artificial intelligence* dan *neural network*. Gabungan kedua himpunan menghasilkan *artificial intelligence*, *neural network*, dan *machine learning*, dengan jumlah elemen sebanyak 3. Dengan demikian, nilai *Jaccard Similarity* untuk Koleksi B adalah 0,67 yang dimana lebih besar dari koleksi B.

$$\begin{aligned}
 J(\text{Keyword, Koleksi A}) &= \frac{\{\text{artificial intelligence, machine learning, neural network}\} \cap \{\text{neural network, classification}\}}{\{\text{artificial intelligence, machine learning, neural network}\} \cup \{\text{neural network, classification}\}} \\
 &= \frac{\{\text{neural network}\}}{\{\text{artificial intelligence, machine learning, neural network, classification}\}} \\
 &= \frac{1}{4} = 0,25 \\
 \\
 J(\text{Keyword, Koleksi B}) &= \frac{\{\text{artificial intelligence, machine learning, neural network}\} \cap \{\text{artificial intelligence, neural network}\}}{\{\text{artificial intelligence, machine learning, neural network}\} \cup \{\text{artificial intelligence, neural network}\}} \\
 &= \frac{\{\text{artificial intelligence, neural network}\}}{\{\text{artificial intelligence, machine learning, neural network}\}} \\
 &= \frac{2}{3} = 0,67
 \end{aligned}$$

Gambar 5.6 Contoh Perhitungan *Jaccard Similarity*

Dengan algoritma *Euclidean Distance*, elemen-elemen dianggap sebagai titik dalam ruang vektor, di mana setiap elemen dalam *keyword* dan koleksi direpresentasikan sebagai dimensi dalam ruang tersebut. *Euclidean Distance* dihitung dengan mengambil akar kuadrat dari jumlah kuadrat perbedaan nilai pada setiap elemen antara dua vektor. Dalam konteks ini, elemen-elemen yang dibandingkan adalah *artificial intelligence*, *neural network*, *machine learning*, dan *classification*. Untuk menghitung *Euclidean Distance*, langkah pertama adalah mengonversi *keyword* dan koleksi menjadi vektor, di mana setiap elemen dikodekan dengan angka 1 jika elemen ada dan 0 jika elemen tidak ada. Selanjutnya, perhitungan dilakukan dengan menghitung selisih kuadrat untuk setiap elemen yang sesuai dari kedua vektor, menjumlahkan hasilnya, dan mengambil akar kuadrat dari jumlah tersebut.

Untuk Koleksi A, selisih nilai antara elemen *keyword* dan koleksi A dihitung untuk setiap atribut. Perhitungannya adalah  $(1 - 0)^2$  untuk *artificial intelligence*,  $(1 - 1)^2$  untuk *neural network*,  $(1 - 0)^2$  untuk *machine learning*, dan  $(0 - 1)^2$  untuk *classification*. Jumlah kuadrat dari nilai-nilai ini adalah 3, sehingga jaraknya adalah akar dari 3 yaitu 1,73. Untuk Koleksi B, proses yang sama diterapkan. Selisih nilai dihitung sebagai  $(1 - 1)^2$  untuk *artificial intelligence*,

$(1 - 1)^2$  untuk *neural network*,  $(1 - 0)^2$  untuk *machine learning*, dan  $(0 - 0)^2$  untuk *classification* karena tidak ada pada keyword maupun koleksi B. Jumlah kuadratnya adalah 1, sehingga jaraknya adalah akar dari 1 yaitu 1. Perhitungan lebih detail dapat dilihat pada Gambar 5.7.

	Artificial intelligence	Neural network	Machine learning	Classification
Keyword	1	1	1	0
Koleksi A	0	1	0	1
Koleksi B	1	1	0	0

$$Dist(\text{Keyword}, \text{Koleksi A}) = \sqrt{(1 - 0)^2 + (1 - 1)^2 + (1 - 0)^2 + (0 - 1)^2} = \sqrt{3} = 1.73$$

$$Dist(\text{Keyword}, \text{Koleksi B}) = \sqrt{(1 - 1)^2 + (1 - 1)^2 + (1 - 0)^2 + (0 - 0)^2} = \sqrt{1} = 1$$

Gambar 5.7 Contoh Perhitungan *Euclidean Distance*

Dalam algoritma *Cosine Similarity*, *keyword* dan koleksi dianggap sebagai vektor dalam ruang multidimensi, di mana setiap dimensi mewakili adanya elemen tertentu. Sama seperti *Euclidean Distance*, *keyword* dan koleksi juga dikonversi menjadi vektor, dengan cara memberikan nilai 1 untuk elemen yang ada, dan 0 untuk elemen yang tidak ada. Untuk Koleksi A, *Cosine Similarity* dihitung dengan mencari *dot product* antara vektor *keyword* dan Koleksi A. Hal itu dilakukan dengan menjumlahkan hasil perkalian elemen-elemen yang sesuai, yaitu 0 untuk “artificial intelligence”, 1 untuk “neural network”, 0 untuk “machine learning”, dan 0 untuk “classification” sehingga menghasilkan 1. Selanjutnya, panjang vektor dihitung dengan untuk *keyword*, panjangnya adalah akar kuadrat dari jumlah kuadrat setiap elemen yaitu akar 3, sementara untuk koleksi A panjangnya akar kuadrat dari 2. Hasil *Cosine Similarity* dihitung dengan membagi *dot product* dengan hasil kali panjang kedua vektor, yang menghasilkan nilai sekitar 0,4.

Proses serupa dilakukan untuk Koleksi B. *Dot product* dihitung dari hasil penjumlahan perkalian elemen-elemen yang sesuai, yaitu 1 untuk “artificial intelligence”, 1 untuk “neural network”, 0 untuk “machine learning”, dan 0 untuk “classification” sehingga menghasilkan 2. Panjang vektor untuk *keyword* tetap sama seperti sebelumnya, sedangkan panjang vektor untuk Koleksi B dihitung dari elemen-elemen  $(1^2 + 0^2 + 1^2 + 0^2)$ . *Cosine Similarity* dihitung dengan membagi *dot product* dengan hasil kali panjang kedua vektor, menghasilkan nilai sekitar 0,81.

	Artificial intelligence	Neural network	Machine learning	Classification
Keyword	1	1	1	0
Koleksi A	0	1	0	1
Koleksi B	1	1	0	0

$$\begin{aligned} \text{Cos}(\text{Keyword}, \text{Koleksi A}) &= \frac{(1 \times 0) + (1 \times 1) + (1 \times 0) + (0 \times 1)}{\sqrt{(1^2) + (1^2) + (1^2) + (0^2)} \times \sqrt{(0^2) + (1^2) + (0^2) + (1^2)}} \\ &= \frac{1}{\sqrt{3} \times \sqrt{2}} = \frac{1}{\sqrt{6}} = \frac{1}{2.449} = 0,4 \end{aligned}$$

$$\begin{aligned} \text{Cos}(\text{Keyword}, \text{Koleksi B}) &= \frac{(1 \times 1) + (1 \times 1) + (1 \times 0) + (0 \times 0)}{\sqrt{(1^2) + (1^2) + (1^2) + (0^2)} \times \sqrt{(1^2) + (1^2) + (0^2) + (0^2)}} \\ &= \frac{2}{\sqrt{3} \times \sqrt{2}} = \frac{2}{\sqrt{6}} = \frac{2}{2.449} = 0,81 \end{aligned}$$

Gambar 5.8 Contoh Perhitungan *Cosine Similarity*

## 5.6 Perbandingan Hasil Dengan Sistem Perpustakaan Saat Ini

Pada bagian ini, dilakukan perbandingan hasil antara metode *Jaccard Similarity*, yang merupakan algoritma terbaik diantara tiga algoritma yang dibandingkan sebelumnya dengan metode yang digunakan perpustakaan saat ini yaitu *Full Text Search* dari PostgreSQL. Contoh query *Full Text Search* di PostgreSQL dapat dilihat pada Gambar 5.9. Dapat dilihat pada Tabel 5.7, hasil *Precision@20* dari *Full Text Search* cukup tinggi pada query *Purchase Intention* dan *Consumer Satisfaction*, walaupun tetap lebih rendah jika dibandingkan dengan metode *Jaccard Similarity*. Hal ini kemungkinan terjadi karena query-query seperti *Purchase Intention* dan *Consumer Satisfaction* merupakan kata yang lebih general sehingga lebih mudah bagi sistem *Full Text Search* untuk menemukan koleksi yang memiliki kata kunci tersebut secara literal.

Sebaliknya, nilai *Precision@20* dari *Full Text Search* cenderung lebih rendah pada query-query seperti *Artificial Intelligence*, *Furniture Design*, atau *Mass Communication*. Sebagai contoh, terdapat salah satu koleksi yang dikembalikan berjudul “Pengaruh financial attitude, emotional intelligence, dan locus of control terhadap money management di komunitas gamers” yang dimana koleksi ini tidak terkait dengan *Artificial Intelligence* sama sekali. Namun karena judulnya mengandung kata “intelligence”, sistem menganggapnya relevan. Selain itu, hasil dari query *Artificial Intelligence* hanya mengembalikan 5 koleksi saja yang menunjukkan keterbatasan dalam memahami konteks yang lebih luas dari kata kunci, terutama jika kata kunci

yang diinputkan spesifik seperti *Artificial Intelligence*. Misalnya, terdapat koleksi berjudul “Penerapan algoritma particle swarm optimization (PSO) untuk optimasi pembangunan negara dalam turn based strategy game” yang relevan dengan *Artificial Intelligence* tetapi tidak secara eksplisit mengandung kata kunci tersebut secara literal. *Full Text Search* gagal mendeteksi relevansi koleksi ini karena bergantung pada pencocokan kata literal, sedangkan *Jaccard Similarity* berhasil mengenali hubungan konseptual dan mengembalikannya sebagai hasil yang relevan.

Sedangkan untuk query *Furniture Design*, meskipun kata kunci tersebut sudah cukup spesifik namun masih mengandung kata yang umum yaitu “design”. Kata “design” tergolong sebagai kata yang sangat umum digunakan dalam berbagai konteks koleksi, sehingga dapat mempengaruhi sistem pencarian yang cenderung mengembalikan koleksi yang mengandung kata “design” tanpa memperhatikan relevansi konteksnya dengan “furniture”. Contohnya, koleksi seperti “Penerapan metode Modified-Partial Capacity Design (M-PCD) pada bangunan persegi 6 lantai” dikembalikan meskipun tidak memiliki keterkaitan dengan “furniture” di urutan yang cukup tinggi. Hal ini yang kemungkinan menyebabkan nilai Precision@20-nya rendah.

Begitu pula dengan query *Mass Communication*, kata “communication” yang juga sangat umum menghasilkan koleksi dari berbagai subjek yang tidak selalu relevan dengan “mass communication”. Misalnya, koleksi seperti “A Study of the word-formation processes used in SMS text” atau “The Functions of questions produced by Petra Christian University's male and female students” dikembalikan meskipun hanya berhubungan dengan komunikasi secara umum, bukan komunikasi massa. Sistem *Full Text Search* tidak mampu memahami kombinasi kata “mass” dan “communication” sebagai sebuah konsep gabungan. Sistem cenderung menganggap kata “communication” sebagai unit pencarian independen, yang akhirnya menyebabkan koleksi yang relevan tidak selalu muncul di posisi teratas.

Secara keseluruhan, hasil perbandingan antara metode *Jaccard Similarity* dan *Full Text Search* dari PostgreSQL menunjukkan bahwa *Jaccard Similarity* memiliki performa yang lebih baik dalam mengembalikan koleksi yang relevan. Hal ini terlihat dari nilai Precision@20 dari *Full Text Search* adalah 0.33 dan Precision@10 bernilai 0.307, yang dimana lebih rendah. Selain itu, *Jaccard Similarity* mampu menangkap hubungan semantik yang lebih mendalam antara kata kunci dan koleksi, sehingga menghasilkan daftar koleksi yang lebih relevan di posisi teratas. Sebaliknya, *Full Text Search* yang bergantung pada pencocokan literal sering kali mengembalikan koleksi yang tidak relevan karena kurangnya pemahaman konteks kata kunci dan ketidakmampuannya dalam menangkap keterkaitan antar koleksi.

Tabel 5.7

Tabel Hasil Metrik Evaluasi Full Text Search dari PostgreSQL

Kata Kunci	Precision @20	Precision @10	Precision	Recall	F1-Score	MAP
Artificial Intelligence	0.2	0.4	0.8	0.236	0.363	0.248
Furniture Design	0.133	0.2	0.226	0.308	0.256	0.38
Purchase Intention	0.7	0.1	0.517	0.488	0.496	0.718
Mass Communication	0.167	0.233	0.185	0.086	0.116	0.484
Consumer Satisfaction	0.45	0.6	0.343	0.339	0.34	0.481
<b>Average</b>	<b>0.33</b>	<b>0.307</b>	<b>0.414</b>	<b>0.292</b>	<b>0.314</b>	<b>0.462</b>

```

1 SELECT id, title,
2     ts_rank(to_tsvector('english', coalesce(title, '')) || ' ' || coalesce(description, '')),
3     to_tsquery('english', 'artificial | intelligence')) AS rank
4 FROM koleksi_digital_1000
5 WHERE to_tsvector('english', coalesce(title, '')) || ' ' || coalesce(description, '')
6     @@ to_tsquery('english', 'artificial | intelligence')
7 ORDER BY rank DESC;

```

Gambar 5.9 Contoh Query Full Text Search di PostgreSQL

### 5.7 Perbandingan Kata Kunci

Variasi nilai antar query kata kunci menunjukkan bahwa performa algoritma *Jaccard Similarity* juga dipengaruhi oleh karakteristik query. Query spesifik cenderung memiliki nilai *Precision@k* yang lebih tinggi karena relevansi koleksi lebih mudah diidentifikasi. Contohnya, *Purchase Intention* dan *Consumer Satisfaction* masing-masing memiliki nilai *Precision@K* sebesar 1.0 dan 0.8, menunjukkan bahwa algoritma mampu menghasilkan koleksi relevan di posisi teratas untuk query ini. Namun, pada query *Artificial Intelligence* yang juga cukup spesifik, nilai *Precision@K* justru lebih rendah. Hal ini kemungkinan disebabkan oleh jumlah koleksi dengan topik *Artificial Intelligence* di database perpustakaan relatif lebih sedikit dibandingkan *Purchase Intention* dan *Consumer Satisfaction*, yang merupakan topik umum dalam studi bisnis dan cenderung memiliki lebih banyak koleksi relevan di database.

Sebaliknya, query yang mengandung kata umum menghasilkan lebih banyak koleksi yang tidak relevan, yang menurunkan *Precision@K*. Contohnya, query *Furniture Design* dan *Mass Communication* menghadapi lebih banyak *noise* karena kata-kata seperti “design” dan “communication” yang dimana sering muncul dalam konteks berbeda. *Noise* ini membuat algoritma kesulitan memprioritaskan koleksi relevan di posisi teratas, sehingga *Precision@K* cenderung lebih rendah. Namun, query yang mengandung kata umum cenderung memiliki nilai *Precision* keseluruhan yang lebih tinggi dibandingkan query spesifik. Hal ini kemungkinan terjadi meskipun koleksi relevan berada di top k, banyak koleksi tidak relevan muncul di luar k.

## 5.8 Analisis Hasil Pencarian

Dari hasil pencarian pada *search engine* yang dibuat menggunakan *Jaccard Similarity* masih ditemukan beberapa koleksi yang tidak relevan dengan kata kunci yang dimasukkan. Hal ini mungkin disebabkan karena kata kunci yang dimasukkan terlalu umum. Sebagai contoh, dari kata kunci “*Mass Communication*”, *search engine* menampilkan koleksi berjudul “*Persuasive strategies used by Agung Sedayu Group in the Infomercial, Metro TV*” yang dimana tidak terkait dengan topik *Mass Communication* secara keseluruhan. Namun karena koleksi tersebut memiliki topik mengenai *persuasive strategy* yang dimana merupakan bagian dari topik *communication*, sehingga sistem juga mengembalikan koleksi tersebut. Hal ini terjadi karena terdapat proses yang melakukan *split* pada kata kunci yang diinputkan dan setiap kata tersebut akan dicari *related keywords*-nya juga di Wikidata. Proses ini menyebabkan koleksi yang seharusnya tidak terkait, namun karena memiliki kata-kata individual yang sama, dianggap relevan oleh sistem.

Selain itu, ambiguitas kata kunci juga menjadi faktor signifikan. Kata kunci dengan makna ganda, seperti “*design*”, dapat merujuk pada desain dalam konteks seni atau desain dalam arti perancangan. Hal ini sering kali membingungkan algoritma pencarian dan menghasilkan koleksi yang mungkin relevan dengan satu interpretasi tetapi tidak sesuai dengan maksud pengguna. Contohnya saat mencari dengan kata kunci “*Furniture Design*”, *search engine* mengembalikan koleksi dengan judul “*Perancangan dan pembuatan sistem informasi manajemen fakultas teknologi industri berbasis web dengan content management system*” yang dimana sangatlah jauh dari topik *Furniture Design*. Namun, karena arti kata perancangan yang dianggap sebagai desain dalam arti lebih umum, koleksi yang tidak sesuai konteks ini dapat muncul.

Faktor penyebab lain adalah pada proses *keyword extraction* yang menghasilkan frasa terlalu umum, seperti “websites” yang ketika diproses melalui Wikidata memunculkan *related keywords* seperti “online service” atau “digital media”. *Related keywords* yang diambil dari Wikidata terkadang terlalu luas sehingga tidak sesuai dengan konteks query. Hal ini dapat memperluas cakupan pencarian secara tidak terkendali sehingga koleksi yang sebenarnya tidak relevan ikut terambil.

Terlepas dari kemunculan beberapa koleksi yang tidak relevan, *search engine* yang dibuat juga menunjukkan beberapa keunggulan dalam memberikan hasil pencarian yang sesuai dengan kata kunci yang dicari. Sistem ini mampu memperhitungkan hubungan semantik antar kata, tidak hanya cocok dengan kata kunci secara langsung, tetapi juga dapat menilai kedekatan makna antar berbagai istilah yang berhubungan. Hal itu karena pengintegrasian *related keywords* dari Wikidata yang menjadi salah satu keunggulan signifikan. Dengan Wikidata, pencarian diperkaya dengan istilah atau konsep yang tidak selalu tercakup dalam kata kunci asli, tetapi memiliki keterkaitan yang erat dengan topik pencarian.

*Search engine* juga berhasil memposisikan hasil pencarian secara urut dari yang sangat relevan ke yang tidak seberapa relevan berkat perhitungan *similarity*. Hal ini sangat bagus mengingat pengguna biasanya hanya melihat 1-2 halaman pertama, sehingga koleksi yang tidak relevan berada diluar 2 halaman pertama dan koleksi yang paling sesuai berada di urutan teratas. Sebagai contoh, saat dilakukan pencarian dengan kata kunci “Consumer Satisfaction”, koleksi pertama yang ditampilkan berjudul “Analisis pengaruh persepsi harga dan citra merek terhadap keputusan pembelian melalui kepuasan konsumen sebagai variabel mediasi pada Mixue di Surabaya” yang dimana sesuai dengan kata kunci. Koleksi ini membahas hubungan kepuasan konsumen dengan berbagai faktor yang memengaruhi keputusan pembelian, sehingga sangat relevan dengan kata kunci. Pada posisi diluar 20 teratas, tepatnya posisi ke-46, koleksi yang ditampilkan berjudul “Pengaruh brand image dan online store image terhadap purchase intention melalui perceived risk pada produk sneakers Adidas”. Meskipun koleksi ini membahas membahas faktor-faktor seperti brand image dan perceived risk yang dapat memengaruhi kepuasan konsumen secara implisit. Namun, fokus utama koleksi tersebut adalah pada niat beli produk, bukan pada analisis langsung tentang tingkat kepuasan konsumen terhadap produk atau layanan tertentu. Sehingga koleksi tersebut tidak terlalu relevan secara langsung dan wajar berada pada posisi tersebut. Pada urutan ke-127, ditampilkan koleksi berjudul “Motif dan kepuasan subscriber menonton channel ‘Kok Bisa’ di Youtube”. Koleksi ini membahas kepuasan tetapi dalam konteks yang sama sekali berbeda, yaitu kepuasan konsumen

media digital. Penempatan koleksi tersebut jauh di belakang mengingat konteksnya tidak secara langsung terkait dengan topik “Consumer Satisfaction” dalam arti pemasaran atau pembelian, tetapi lebih relevan pada konteks media digital.

Hal ini menunjukkan bahwa sistem secara bertahap menurunkan peringkat hasil yang kurang relevan, sehingga koleksi yang benar-benar sesuai dengan konteks kata kunci tetap mendominasi halaman pertama. Sehingga *search engine* yang dibuat tidak hanya membantu pengguna menemukan hasil yang relevan secara cepat, tetapi juga memastikan bahwa hasil yang kurang sesuai tidak mengganggu pengalaman pencarian pengguna di awal proses eksplorasi informasi.