JNATIA Volume 4, Nomor 2, Februari 2026 p-ISSN: 2986-3929
Jurnal Nasional Teknologi Informasi dan Aplikasinya e-ISSN: 3032-1948

Implementasi Model Poisson Laplace untuk IR pada E-Skripsi Universitas Udayana

I Putu Andhika Ardianta Putra[a1], Cokorda Pramartha[a2]

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana, Bali
Jalan Raya Kampus Udayana, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]andhikaardianta1407@gmail.com
[2]cokorda@unud.ac.id

Abstract

At Udayana University, a digital repository system is available to store books and student theses. However, the current repository system only categorizes documents based on the year of publication, without any grouping based on topics, fields of science, or other data. Although a search feature is available, the function has not been optimized. To overcome this problem, research was conducted on the Probabilistic Information Retrieval Model, namely the Poisson Model with Laplace Smoothing and Normalization 2. Research was conducted on student thesis title data in 2024 as many as 4,671 titles and evaluation using 10 queries. The research resulted in a Mean Average Precision value, normalized Discounted Cumulative Gain, and recall of 0.715 and a precision value of 0.1421. From this value, further experiments need to be held because the precision value is far different from the recall, indicating the number of False Positive values.

Keywords: *Information Retrieval, PL2, Text Mining, Theses, Poisson Distribution*

1. Pendahuluan

Perkembangan teknologi informasi telah membawa perubahan signifikan di berbagai bidang, termasuk dalam dunia pendidikan. Salah satu bentuk penerapannya adalah penggunaan sistem berbasis web untuk memfasilitasi berbagai aktivitas akademik, seperti pengisian Kartu Rencana Studi (KRS), input nilai mahasiswa, serta pengelolaan repositori kampus. Repository ini menjadi wadah penting dalam menyimpan, mengelola, dan menyebarluaskan karya ilmiah seperti skripsi dan buku digital secara online [1].

Di Universitas Udayana, telah tersedia sistem repositori digital untuk menyimpan buku dan skripsi mahasiswa. Namun, sistem repositori saat ini hanya mengelompokkan dokumen berdasarkan tahun terbit, tanpa adanya pengelompokan berdasarkan topik, bidang ilmu, atau data lainnya. Meskipun tersedia fitur pencarian, fungsinya belum berjalan optimal. Hal ini menyebabkan mahasiswa kesulitan dalam menemukan dokumen yang relevan, karena harus menelusuri secara manual dari ratusan hingga ribuan skripsi yang tersedia.

Permasalahan ini berkaitan erat dengan bidang *Information Retrieval* (IR), yaitu proses pengambilan informasi dari sekumpulan dokumen berdasarkan permintaan pengguna. Salah satu pendekatan paling sederhana dalam IR adalah *Boolean Model*, yang hanya mempertimbangkan keberadaan atau ketiadaan kata kunci dalam dokumen. Namun, pendekatan ini memiliki keterbatasan dalam menangani *partial matching*, karena *query* dari user harus sama persis dengan dokumen yang dicari.

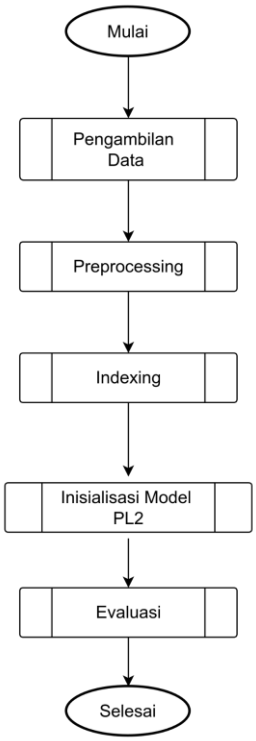
Untuk mengatasi keterbatasan tersebut, dikembangkan model-model IR lain seperti *algebraic model* dan *probabilistic model*. Model probabilistik bekerja dengan memanfaatkan kehadiran dan frekuensi istilah dalam dokumen untuk menentukan relevansi terhadap query pengguna. Salah satu model probabilistik yang unggul dan modern adalah *Poisson Model with Laplace Smoothing and Normalization 2* **(PL2)** [2]. Model ini merupakan bagian dari framework *Divergence from*

Randomness (DFR), yang mengukur ketidakteraturan distribusi istilah dalam dokumen terhadap distribusi acak menggunakan distribusi Poisson [3].

PL2 memiliki keunggulan dalam mendukung *partial match* dan penilaian relevansi berbasis probabilitas, sehingga mampu memberikan hasil pencarian yang lebih fleksibel dibandingkan metode pencocokan kata kunci secara langsung. Keunggulan ini berguna dalam sistem pencarian e-skripsi karena judul dan isi skripsi sering menggunakan variasi istilah, sinonim, atau perbedaan gaya penulisan, yang menyebabkan dokumen relevan tidak selalu memiliki kecocokan kata yang sama persis dengan kueri pengguna. Model ini akan diimplementasikan pada dataset yang terdiri dari 4.671 judul skripsi mahasiswa tahun 2024 yang tersedia di e-perpustakaan Universitas Udayana. Dengan penerapan PL2, diharapkan proses pencarian skripsi dapat menjadi lebih efektif dan efisien, serta meningkatkan pengalaman mahasiswa dalam mengakses informasi ilmiah.

2. Metode Penelitian

Tahapan dari pelaksanaan penelitian ini dapat dilihat pada gambar 1



Gambar 1. Diagram alur penelitian

Penelitian dimulai dengan (1) mengambil judul skripsi dari e-skripsi Universitas Udayana secara manual dengan mengeksekusi kode javascript pada console chrome dan memindahkan hasilnya ke file .txt. Dari proses ini didapatkan dataset berupa 4.671 judul skripsi tahun 2024, dimana setiap data merepresentasikan satu judul skripsi dalam bentuk teks dengan campuran bahasa Indonesia dan istilah bahasa inggris, disertai dengan singkatan-singkatan. Setiap data memiliki panjang yang bervariasi dengan topik penelitian dari berbagai program studi. Dataset ini dibatasi hanya pada informasi judul tanpa menyertakan abstrak atau isi dokumen, sehingga berfokus pada efektivitas metode pencarian berbasis judul. (2) Selanjutnya data judul skripsi akan melalui tahap preprocessing. Tahap preprocessing ini terdiri dari *lowercasing*, penghapusan karakter selain alfabet, penghapusan *stopword*, dan *stemming*. Setelah preprocessing, selanjutnya dilakukan *indexing* dan hasil indexnya akan digunakan untuk pencarian berdasarkan *query user* menggunakan model PL2.

$$PL2(t,d) = \frac{1}{tfn+1}[tfn \cdot log_2(\frac{tfn}{\lambda}) + (\lambda - tfn) \cdot log_2(e) + 0.5 \cdot log_2(2\pi tfn)] \quad (1)$$

Keterangan:

t = Istilah term pada dokumen
d = dokumen yang dievaluasi
Tfn = Term frequency normalized (Frekuensi kemunculan istilah dalam dokumen, yang telah dinormalisasi berdasarkan panjang dokumen)
λ = Expectation (mean) dari distribusi Poisson untuk istilah *t*

Indexing dan pencarian menggunakan model ini akan menggunakan *library* Pyterrier. Kemudian, hasil pencarian akan dievaluasi menggunakan Precision, Recall, *normalized Discounted Cumulative Gain (nDCG)* dan *Mean Average Precision (MAP).*

$$Precision = \frac{|TP|}{|TP| + |FP|} \quad (2)$$

$$Recall = \frac{|TP|}{|TP| + |FN|} \quad (3)$$

Keterangan:

TP = True Positive (benar dinilai benar)
FP = False Positive (salah dinilai benar)
FN = False Negative (salah dinilai benar)

$$MAP = \frac{1}{|Q|} \sum_{q \in Q} AP(q) \quad (4)$$

Keterangan:

Q = Jumlah total query
AP(q) = Average Precision untuk query ke-q

$$nDCGp = \frac{DCGp}{IDCGp} \quad (5)$$

Keterangan
:
$DCGp$ = Discounted Cumulative Gain
$IDCGp$ = Ideal $DCGi$ (DCG pada urutan relevansi terbaik)

Metrik – metrik ini merupakan standar dalam evaluasi sistem IR modern [5].

3. Hasil dan Diskusi

Implementasi PL2 dilakukan menggunakan library PyTerrier, yaitu library Python yang memungkinkan integrasi berbagai model information retrieval (IR), termasuk model dari metode *Divergence from Randomness* (*DFR*) seperti PL2. Dataset yang digunakan dalam penelitian ini adalah kumpulan 4.671 judul skripsi tahun 2024 dari sistem e-skripsi Universitas Udayana. Setelah melalui tahap *preprocessing*, dilakukan proses *indexing* menggunakan fungsi *DFIndexer.index()*. Setelah indexing, untuk mencari dokumen berdasarkan satu *query* akan menggunakan fungsi *search ()*, dan untuk scoring berdasarkan beberapa query akan menggunakan fungsi *transform ()*.

3.1. Preprocessing Text

Tahap preprocessing terdiri dari *lowercasing*, penghapusan karakter selain alfabet, penghapusan *stopword*, dan *stemming*. Untuk stopword dan stemming akan dilakukan berdasarkan dictionary,

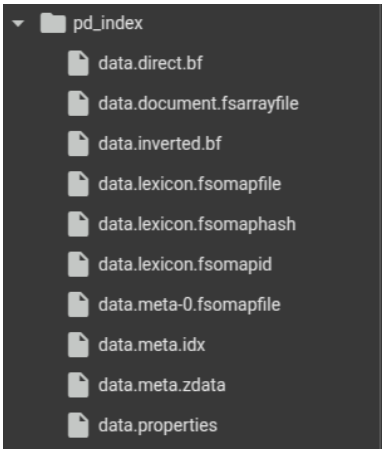
menggunakan library sastrawi. Pada umumnya, tahap preprocessing mencakup proses tokenisasi. Namun, dalam penelitian ini, proses tokenisasi tidak dilakukan secara manual karena telah ditangani secara otomatis oleh PyTerrier saat proses indexing.

Tabel 1. Hasil Preprocessing

Sebelum Preprocessing	Setelah Preprocessing
Pengaruh Kompensasi dan Kepuasan Kerja Terhadap Kinerja Karyawan Pada Perusahaan Optimate Holistic Cabang Kebo Iwa	pengaruh kompensasi kepuasan kerja kinerja karyawan perusahaan optimate holistic cabang kebo iwa
DETERMINAN PENERIMAAN APLIKASI ATLAS (AUDIT TOOLS AND LINKED ARCHIVE SYSTEM) (STUDI PADA KAP BALI)	determinan penerimaan aplikasi atlas audit tools and linked archive system studi kap bali
Analisis Faktor Yang Mempengaruhi Produksi Batu Bata Di Kabupaten Tabanan	analisis faktor mempengaruhi produksi batu bata kabupaten tabanan
FAKTOR-FAKTOR YANG MEMPENGARUHI MAHASISWA DALAM MENGIKUTI MERDEKA BELAJAR KAMPUS MERDEKA	faktorfaktor mempengaruhi mahasiswa mengikuti merdeka belajar kampus merdeka
Perencanaan Lanskap Kaki Rinjani Camp di Desa Sajang, Kecamatan Semablun, Kabupaten Lombok Timur, Nusa Tenggara Barat	perencanaan lanskap kaki rinjani camp desa sajang kecamatan semablun kabupaten lombok timur nusa tenggara barat

3.2. Indexing

Untuk indexing, akan digunakan fungsi bawaan *PyTerrier*, yaitu *DFIndexer.index()* [4]. Fungsi akan menerima data text sebagai parameter untuk membangun index, yang akan disimpan dalam bentuk file.



Gambar 2. File hasil DFIndexer

3.3. Pemanggilan Model

Pertama, model PL2 diinisialisasi menggunakan fungsi *terrier.Retriever()* [4]. Fungsi ini menerima 1 parameter wajib, yaitu *index* dari daftar dokumen. Untuk membuat model menggunakan model PL2, ditambahkan parameter *wmodel*='PL2' ketika pemanggilan fungsi. Fungsi juga dapat menerima parameter *num_results* untuk membatasi jumlah dokumen yang dikembalikan model ketika melakukan pencarian.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

```
pd_indexer = pt.DFIndexer("./pd_index", overwrite=True, tokeniser="UTFTokeniser", blocks=True)
data["docno"] = data["docno"].astype(str)

indexref = pd_indexer.index(data["title"], data['docno'])

data['qid'] = data['docno']

PL2 = pt.terrier.Retriever(indexref, wmodel="PL2", num_results=15)
```

Gambar 3. Inti code model PL2

Setelah inisialisasi, model sudah dapat melakukan pencarian dokumen dengan fungsi *search ()*. Fungsi search memerlukan 1 parameter wajib, yaitu query user dalam bentuk string.

qid	docid	docno	rank	score	query
1	1389	1390	0	9.525128	machine learning
1	1011	1012	1	7.991900	machine learning
1	4076	4077	2	7.991900	machine learning
1	2341	2342	3	7.689219	machine learning
1	2100	2101	4	7.409468	machine learning
1	1646	1647	5	6.907461	machine learning
1	1647	1648	6	6.907461	machine learning
1	993	994	7	6.680805	machine learning
1	2211	2212	8	6.468047	machine learning
1	213	214	9	6.267774	machine learning
1	214	215	10	6.267774	machine learning
1	215	216	11	6.267774	machine learning
1	217	218	12	6.267774	machine learning
1	2260	2261	13	5.014367	machine learning
1	2163	2164	14	4.762564	machine learning

Gambar 4. Hasil *search()* dari model PL2

3.4. Evaluasi Model

Untuk evaluasi, pertama-tama digunakan fungsi *transform ()* terhadap model [4]. Fungsi ini menerima data yang berisi kumpulan query dalam bentuk string.

qid	docid	docno	rank	score	query
1	1389	1390	0	9.525128	machine learning
1	1011	1012	1	7.991900	machine learning
1	4076	4077	2	7.991900	machine learning
1	2341	2342	3	7.689219	machine learning
1	2100	2101	4	7.409468	machine learning
..	...	...	...	...	...
10	1037	1038	1	5.038112	k-nearest neighbor
10	196	197	2	4.526291	k-nearest neighbor
10	3910	3911	3	4.526291	k-nearest neighbor
10	2647	2648	4	4.240172	k-nearest neighbor
10	896	897	5	3.465425	k-nearest neighbor

Gambar 5. Hasil *transform()* dari model PL2

Dari hasil fungsi *transform (),* kemudian akan ditambahkan *label* pada tiap barisnya. Label ini melambangkan benar atau tidaknya hasil dokumen yang diberikan dokumen. Label dapat berupa biner (0 = tidak relevan, 1=relevan), atau angka bertingkat 0 = tidak relevan, 1=cukup relevan, 2=sangat relevan). Untuk menambah label, pertama-tama hasil *transform ()* dapat diekspor menjadi file .csv terlebih dahulu. File .csv kemudian diberi label secara manual, kemudian diimpor kembali untuk tahap selanjutnya.

Tabel 2. Sample Hasil *transform ()* yang diberi label

qid	docid	docno	score	query	text	label
214	215	10	6.267773912682396	machine learning	capstone project rancang bangun robot smart toy berbasis sensor teknologi machine learning pemantauan stimulasi kognitif motorik balita	1
215	216	11	6.267773912682396	machine learning	capstone project rancang bangun robot smart toy berbasis sensor teknologi machine learning pemantauan stimulasi kognitif motorik balita	1
217	218	12	6.267773912682396	machine learning	capstone project rancang bangun robot smart toy berbasis sensor teknologi machine learning pemantauan stimulasi kognitif motorik balita	1
2260	2261	13	5.0143668061298605	machine learning	creative learning hub pontianak utara	0
2163	2164	14	4.762563911948022	machine learning	ruang kreatif digital learning hub denpasar	0

Untuk melakukan evaluasi, akan digunakan fungsi *Evaluate ()* yang ada pada library *PyTerrier*. Fungsi ini menerima dua parameter wajib, yaitu *res* dan *qrels*, dimana *res* berupa *DataFrame* dengan kolom qid, docno, serta score dan qrels berupa *DataFrame* dengan kolom qid, docno, serta label. Untuk metrik yang akan dinilai, dapat dimasukkan pada parameter *metrics*, yang berupa array berisi string metrik yang akan digunakan.

Berikut hasil percobaan model PL2 pada 10 query terkait informatika, yang memunculkan 129 dokumen:

Tabel 3. Hasil Evaluasi

MAP	nDCG	R@5	P@5
0.7105	0.7105	0.7105	0.1421

Berdasarkan nilai diatas, dapat dilihat bahwa nilai recallnya jauh lebih besar dibanding nilai precision. Ini menandakan besarnya nilai FP yang artinya banyak dokumen yang dinilai relevan oleh model, padahal sebenarnya tidak relevan.

Berdasarkan hasil dari penelitian lain yang menggunakan metode BM25, diperoleh nilai R@5 sebesar 0,928 dan nilai P@5 sebesar 0,241 [5]. Hal ini kemungkinan disebabkan oleh jumlah query yang lebih banyak serta ukuran data yang lebih kecil, yaitu hanya 700 dokumen. Temuan tersebut dapat dijadikan pertimbangan untuk melakukan eksperimen lanjutan dengan jumlah query yang lebih banyak dan tidak terbatas pada kata kunci seputar informatika saja.

4. Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa metode PL2 memiliki potensi untuk diimplementasikan pada sistem e-skripsi Universitas Udayana. Hasil evaluasi menunjukkan nilai MAP dan nDCG masing-masing sebesar 0.7105, yang mengindikasikan bahwa PL2 mampu menghasilkan pemeringkatan dokumen yang relevan dan konsisten. Selain itu, nilai Recall@5

p-ISSN: 2986-3929
e-ISSN: 3032-1948

sebesar 0.7105 menunjukkan bahwa sebagian besar dokumen relevan berhasil ditemukan pada lima peringkat teratas. Namun, nilai Precision@5 yang diperoleh masih relatif rendah, yaitu sebesar 0.1421. Hal ini dipengaruhi oleh keterbatasan jumlah dan variasi kueri yang digunakan dalam pengujian, serta penggunaan dataset yang hanya berupa judul skripsi. Oleh karena itu, hasil penelitian ini menunjukkan bahwa PL2 memiliki potensi yang baik, namun masih memerlukan pengujian lanjutan untuk memperoleh evaluasi yang lebih komprehensif.

Daftar Pustaka

[1] Maha, R. N., Widiyaningrum, A. R., & Tupan, T. (2023). Implementasi Sistem Integrasi Repositori Karya Ilmiah di Badan Riset dan Inovasi Nasional (BRIN) Berbasis Eprints. *Media Pustakawan*, *30*(2), 132–142. https://doi.org/10.37014/medpus.v30i2.3701

[2] Kulkarni, H., Goharian, N., Frieder, O., & MacAvaney, S. (2024). LexBoost: Improving Lexical Document Retrieval with Nearest Neighbors (Vol. 33, pp. 1–10). https://doi.org/10.1145/3685650.3685658

[3] Amati, G., & Van Rijsbergen, C. J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Transactions on Office Information Systems, 20(4), 357–389. https://doi.org/10.1145/582415.582416

[4] Macdonald, C., & Tonellotto, N. (2020). Declarative Experimentation in Information Retrieval using PyTerrier. https://doi.org/10.1145/3409256.3409829

[5] Fahmi Ilmi, M., Pandu Adikara, P., & Indriati. (2022). Pencarian Dokumen Skripsi menggunakan BM25 dan Faceted Search berdasarkan Kata Kunci Abstrak (Studi Kasus: Universitas Muhammadiyah Sidoarjo). Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 6(9), 4175–4180. https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/11531

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Halaman ini sengaja dibiarkan kosong