

# Rancang Bangun Aplikasi UMN Library Catalog Menggunakan Metode Rocchio Relevance Feedback

Marcel Bonar Kristanda<sup>1</sup>, Seng Hansun<sup>2</sup>, Albert<sup>3</sup>

Program Studi Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia

Diterima 30 November 2017

Disetujui 8 Juni 2018

**Abstract**— Library catalog is a documentation or list of all library collections. Unfortunately, there is a problem identified in the process of searching a book inside library catalog in Universitas Multimedia Nusantara's library information system regarding the relevant result based on user query input. This research aims to design and build a library catalog application on Android platform in order to increase the relevancy of searching result in a database using calculated Rocchio Relevance Feedback method along with user experience measurement. User experience analysis result presented a good respond with 91.18% score based by all factor and relevance value present 71.43% precision, 100% recall, and 83.33% F-Measure. Differences of relevant results between the Senayan Library Information system (SLiMS) and the new Android application ranged at 36.11%. Therefore, this Android application proved to give relevant result based on relevance rank.

**Index Terms**—Rocchio, Relevance, Feedback, Pencarian, Buku, Aplikasi, Android, Perpustakaan.

## I. PENDAHULUAN

Perpustakaan dalam sebuah institusi pendidikan tinggi dapat dilihat sebagai sebuah tempat untuk mahasiswa mencari bahan pustaka dalam proses pembelajarannya atau sebuah unit kerja yang melakukan pengelolaan dan pelayanan dalam pemanfaatan koleksi bahan pustaka [1]. Dalam menjalankan peranannya dan pelayanan yang berkualitas, pemanfaatan teknologi informasi menjadi suatu hal yang vital di dalam perpustakaan [2]. Salah satu tantangan umum yang dihadapi perpustakaan adalah penyediaan fasilitas sistem informasi pencarian buku yang mengeluarkan hasil secara efektif dan relevan. Beberapa sistem informasi perpustakaan yang bersifat *open source* telah dipublikasikan untuk diunduh dan digunakan secara gratis untuk menjawab tantangan tersebut, salah satunya adalah Senayan Library Management System (SLiMS) [3].

Studi pendahuluan yang telah dilakukan pada SLiMS milik Perpustakaan Universitas Multimedia Nusantara (UMN) telah berhasil mengimplementasikan algoritma Levenshtein Distance pada fitur *autocomplete* SLiMS dengan

tingkat keberhasilan dalam memprediksi *keyword* sebesar 73,33% [4]. Penelitian tersebut bertujuan untuk mengurangi kesalahan masukan pengguna pada mesin pencarian buku di SLiMS sehingga memperkecil kemungkinan pengguna tidak menemukan bahan pustaka yang dicari.

Meskipun fitur tersebut telah diimplementasikan, temuan dalam studi kelayakan lanjutan menunjukkan proses pencarian masih belum menghasilkan keluaran yang relevan dengan kata kunci masukan [5]. Namun, studi tersebut juga menunjukkan bahwa pencarian buku di SLiMS UMN termasuk mudah untuk dilakukan dan mudah untuk dipelajari. SLiMS yang merupakan sistem informasi berbasis *web* ini juga berpotensi untuk didukung dengan aplikasi *mobile* dalam melakukan pencarian buku [6]. Hal ini diperkuat dengan hasil survey terhadap ketertarikan responden dalam melakukan pencarian buku lewat *smartphone* [5].

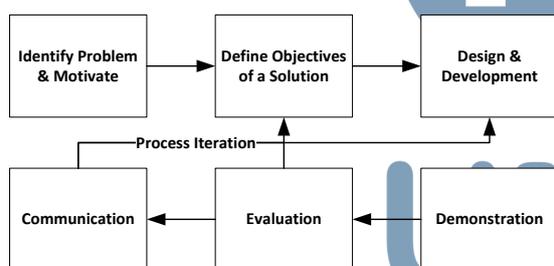
Proses pencarian informasi dapat dievaluasi dan dioptimalkan lewat beberapa algoritma seperti *k-means clustering*, *widrow-hoff* dan *support vector machine* [7]. Metode *relevance feedback* juga menjadi salah satu metode yang telah terbukti dapat mengurangi waktu pemrosesan [7] dan menghasilkan dokumen yang akurat dengan tingkat akurasi yang tinggi berdasarkan *query* dari pengguna [8]. Keunggulan *roocchio relevance feedback* dalam mengolah dengan waktu pemrosesan yang lebih cepat dan presisi yang lebih tinggi dari *support vector machine* telah berhasil diidentifikasi [9]. Hal yang serupa juga ditemukan dalam proses penelusuran informasi ketika metode tersebut dibandingkan dengan algoritma *widrow-hoff* [10]. Temuan pada kedua kajian tersebut juga sejalan dengan perbandingan hasil pencarian kalimat dimana hasil keluaran *roocchio relevance feedback* lebih baik daripada metode *k-means clustering* [11]. Faktor-faktor tersebut yang memotivasi penelitian oleh Yugianus, Harry, dan Rini dalam mengembangkan sistem berbasis web untuk penelusuran katalog perpustakaan dengan metode *roocchio relevance feedback* [12].

Pengembangan sistem yang dilakukan oleh Yugianus,dkk direalisasikan dengan pembangunan sistem berbasis web dan diujicoba fitur-fiturnya dengan *blackbox testing* dengan penggunaan *corpus data* yang belum terdefiniskan dengan jelas [12]. Penelitian tersebut belum mencakup pengujian terhadap *user experience* dari sistem yang digunakan. Sedangkan *user experience* berperan penting dalam pembangunan sebuah sistem untuk mengevaluasi pengalaman pengguna yang memakai langsung sistem tersebut [13].

Berdasarkan studi kelayakan yang telah dilakukan dan keunggulan metode *rocchio relevance feedback* berdasarkan penelitian terkait, maka terbentuklah motivasi penelitian ini. Data yang digunakan dalam penelitian ini merupakan data asli dari Perpustakaan UMN dan mengikuti pola data yang terdapat di dalam SLiMS. Penelitian akan ditinjau dari dua sisi pengukuran yaitu tingkat relevansi [14] dan juga *user experience* [15].

## II. METODOLOGI PENELITIAN

Metodologi penelitian yang dilaksanakan berdasarkan pada proses model Design Science Research Methodology (DSRM) untuk riset sistem informasi [16] yang runtutannya dapat dilihat pada Gambar 1.



Gambar 1. Proses Model Penelitian DSRM.

### A. Identifikasi Masalah dan Motivasi

Keluhan-keluhan terhadap aktivitas pencarian buku yang menggunakan SLiMS telah disampaikan secara lisan dan bersifat non-formal oleh dua tipe pengguna sistem yaitu mahasiswa dan bagian Perpustakaan UMN. Sebagai upaya untuk melihat masalah dari cara pandang yang lebih objektif, maka diselenggarakan suatu studi kelayakan yang bertujuan untuk verifikasi dan melakukan penilaian terhadap keluhan [5]. Studi kelayakan tersebut dilakukan menggunakan dua teknik yaitu:

- Wawancara: Wawancara terhadap kepala perpustakaan dan bagian *information literacy* di Perpustakaan UMN dengan menggali masalah sistem pencarian secara lebih mendalam sebagai stakeholder yang paling erat kaitannya dengan pengelolaan data di dalam sistem SLiMS UMN.

- Survey: Survey terhadap mahasiswa UMN untuk memberikan penilaian terhadap pencarian buku di SLiMS yang dilihat dari faktor *usability*, *functionality*, *user satisfaction* dan *user experience*. Sedangkan, faktor *operability* dan *inovativeness* ditanyakan untuk mengetahui kelayakan aplikasi smartphone sebagai media pencarian buku.

### B. Definisi Tujuan terhadap Solusi

Hasil dari identifikasi masalah dan motivasi kemudian diolah dengan menelaah literatur-literatur pendukung yang hasilnya dijabarkan pada Bab Pendahuluan. Terdapat dua hal yang dianalisa pada tahap ini yaitu:

- 1) *Metode Pencarian Data*: Metode pencarian data dicari dengan fokus kepada metode yang mempunyai banyak keunggulan dari metode lainnya dan dengan performa yang telah teruji.
- 2) *Platform Aplikasi*: Platform aplikasi disaring dengan hasil wawancara dari bagian Perpustakaan UMN dalam konteks kebutuhan dan pengembangan perpustakaan untuk mengadakan fasilitas layanan pencarian buku via *mobile apps*. Kemudian, hal tersebut diverifikasi dengan hasil survey mahasiswa UMN terutama pada penilaian faktor *inovativeness* dan *operability* untuk pengembangan berbasis *mobile*.

Kedua hasil tersebut kemudian membentuk tujuan penelitian untuk merancang dan membangun aplikasi Android pencarian buku perpustakaan menggunakan metode *rocchio relevance feedback*. Perancangan awal berupa *breakdown* fitur-fitur aplikasi juga telah dibuat pada tahap ini sebagai tindak lanjut dari diskusi terhadap pihak Perpustakaan akan hasil studi kelayakan dan studi literatur yang dilakukan [5].

### C. Perancangan dan Pembangunan

Berdasarkan dari hasil perancangan awal sebagai suatu solusi yang diajukan, perancangan keseluruhan aplikasi kemudian dibuat dengan menghasilkan tiga jenis perancangan, yaitu: arsitektur sistem, struktur data buku perpustakaan, dan *flowchart*. Fase pembangunan aplikasi dijalankan berdasarkan arsitektur sistem yang membagi pembangunan aplikasi menjadi dua bagian yaitu dengan pemrograman web dan pemrograman sistem mobile.

### D. Demonstrasi

Dalam tahap ini, dilakukan uji coba terhadap hasil dari pembangunan aplikasi dengan melakukan uji fungsionalitas yang dijabarkan dalam bentuk screenshots. Selain itu, simulasi juga dilakukan terhadap perhitungan *precision*, *recall* dan *F-measure* pada hasil pencarian melalui aplikasi mobile dan kemudian dibandingkan dengan perhitungan pada hasil pencarian melalui SLiMS.

### E. Evaluasi

Evaluasi aplikasi ditujukan kepada pengguna aplikasi utama yaitu mahasiswa UMN dengan menyelenggarakan survey melalui pengisian kuesioner. Kuesioner tersebut berisikan pertanyaan untuk mengevaluasi aplikasi dari segi *user experience* yang kemudian dianalisa untuk menilai kelayakan aplikasi untuk dipublikasikan baik sebagai hasil penelitian maupun aplikasi yang dapat digunakan langsung oleh mahasiswa UMN.

### F. Komunikasi

Tahap komunikasi untuk menyampaikan masalah dan kepentingan penelitian dilakukan dengan 3 tahapan: mempublikasikan aplikasi ke portal pencarian aplikasi, memaparkan dan mempertanggungjawabkan penelitian sebagai sebuah makalah Sarjana Komputer dalam program studi Informatika yang berjudul "Rancang Bangun Aplikasi Android Pencarian Buku Perpustakaan Menggunakan Metode Rocchio Relevance Feedback (Studi Kasus: Perpustakaan UMN)" [17], dan mempublikasikan manuskrip kegiatan dan hasil penelitian ke dalam jurnal nasional.

## III. HASIL STUDI KELAYAKAN

Studi kelayakan melibatkan 100 orang mahasiswa sebagai partisipan untuk survey dan 2 orang karyawan perpustakaan sebagai partisipan untuk wawancara [5].

### A. Hasil Wawancara

Beberapa kekurangan dari SLiMS UMN disampaikan oleh bagian literasi informasi Perpustakaan UMN, salah satunya adalah hasil pencarian buku yang dirasakan masih kurang. Sedangkan, kepala perpustakaan menuturkan bahwa sistem yang sedang berjalan belum sepenuhnya memenuhi kebutuhan. Rencana pengembangan fasilitas perpustakaan ke arah pencarian, peminjaman dan stok opname buku via mobile apps juga disampaikan. Hasil wawancara ini kemudian menjadi dasar pembentukan kuesioner untuk melakukan survey terhadap mahasiswa UMN.

### B. Hasil Survey

Hasil survey menunjukkan bahwa hanya terdapat 40-responden dari total 100-orang yang mengetahui tentang SLiMS sebagai sistem pencarian buku. Kemudian 40 orang tersebut ditanyakan mengenai pengalamannya dalam melakukan pencarian di SLiMS, hasilnya adalah sebenarnya pencarian cukup mudah dilakukan, namun relevansi dan keakuratan yang dihasilkan masih dirasakan kurang. Pengembangan layanan pencarian via aplikasi *mobile* juga direspons dengan sangat positif dari keseluruhan responden.

## IV. METODE ROCCHIO RELEVANCE FEEDBACK

Metode *roocchio relevance feedback* adalah strategi reformulasi *query* yang digunakan untuk membantu

*user* pemula dalam *information retrieval systems*. *User* disajikan dengan hasil pencarian dokumen yang relevan [18]. *Information retrieval* atau *document retrieval* adalah teknik untuk menghasilkan hasil dokumen relevan dengan *request* dari user melalui perbandingan *request* dengan indeks yang menghasilkan tingkat *similarity* dari konten yang dipilih [19].

Teknik *relevance feedback* digunakan setelah *user* memasukkan *keyword* yang data sudah didapat dari pencarian di database. *Relevance feedback* mendekati *query* ke rata-rata dokumen relevan dengan tingkat perhitungan similaritas [12]. Berikut tahapan perhitungan metode Rocchio.

### A. Tokenizing

*Tokenizing* adalah tahap pemrosesan sebuah kata kunci menjadi unit kecil. Pembuatan token dilakukan pada kata kunci dan dokumen yang didapat.

### B. Filtering

*Filtering* adalah tahap mengambil kata-kata penting dari hasil *tokenizing*. Pada tahap ini kata-kata yang tidak deskriptif dibuang, contoh "yang", "dan", "di".

### C. Stemming

*Stemming* adalah tahap mencari akar kata sesuai dengan kata kunci. Pada tahap ini dilakukan proses pengembalian berbagai kata ke dalam suatu representasi.

### D. Term Weighting

*Term weighting* adalah tahap pembobotan tiap term yang dicari pada setiap dokumen sehingga dapat diketahui ketersediaan dan kemiripan suatu term dalam dokumen.

Perhitungan *term frequency* dan *inverse document frequency* akan dilakukan bersamaan dengan *query* yang merupakan fitur di dalam database. *Term frequency* (*tf*) adalah frekuensi dari kemunculan sebuah *term* dalam dokumen yang bersangkutan. *Inverse document frequency* (*idf*) merupakan sebuah perhitungan dari bagaimana *term* didistribusikan secara luas pada koleksi dokumen yang bersangkutan. Rumusan nilai *idf* sebuah *term* dapat dijabarkan menjadi persamaan berikut [20].

$$idf = \log \frac{n}{df} \quad (1)$$

Keterangan:

*idf* : nilai *inverse document frequency*.  
*n* : jumlah dokumen di dalam koleksi.  
*df* : nilai *document frequency*.

Jika pencarian pada dokumen yang cukup besar, skema yang dapat digunakan untuk pemberian bobot adalah *term frequency* dikalikan dengan *inverse document frequency* yang disebut sebagai nilai bobot *term* atau  $\beta$  [12].

$$\beta = (tf) * idf \quad (2)$$

Keterangan:

$\beta$ : nilai bobot term.

$tf$ : nilai term frequency.

$idf$ : nilai inverse document frequency.

Selanjutnya, tingkat kemiripan term dengan judul dokumen akan dimasukkan ke dalam sebuah fungsi perhitungan similaritas yaitu metode Rocchio yang dapat dilihat dari rumusan manual sebagai berikut [12].

$$R = N + \beta \left( \left( \frac{Dp}{Np} \right) - \left( \frac{Dn}{Nn} \right) \right) \quad (3)$$

Keterangan:

R : tingkat kemiripan term.

N : jumlah term tiap dokumen.

$\beta$  : nilai bobot term.

Dp : term dari dokumen relevan.

Np : jumlah term keseluruhan dokumen relevan.

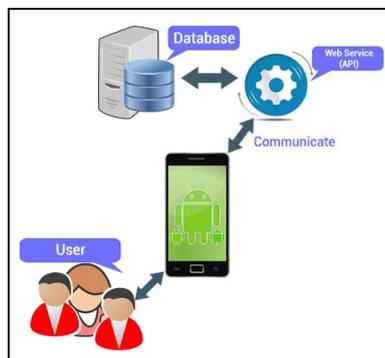
Dn : term dari dokumen tidak relevan.

Nn : jumlah term keseluruhan dokumen tidak relevan

## V. PERANCANGAN APLIKASI

### A. Arsitektur Sistem

Perancangan aplikasi dibuat dengan menggunakan perangkat Android yang melakukan akses database menggunakan Apache Client sebagai web service (API). API merupakan perantara database agar dapat berkomunikasi dengan perangkat mobile. Struktur API yang digunakan untuk berkomunikasi berupa file-file PHP yang terhubung melakukan pertukaran data menggunakan JSON sehingga aplikasi dapat saling berkomunikasi untuk mendapatkan data. Gambaran arsitektur sistem agar aplikasi dapat berjalan dapat dilihat pada Gambar 2.



Gambar 2. Arsitektur Sistem Aplikasi UMN Library Catalog

Pada saat user melakukan *request* pencarian data buku, pengiriman data kata kunci dimulai. Perangkat mobile mengirimkan data *request* dan selanjutnya *request* diterima oleh perantara berupa file PHP yang

di-encode ke dalam JSON, lalu data *request* dieksekusi ke database melalui MySQL *query*. Setelah *query* berhasil dieksekusi, maka database memberikan data hasil. Hasil tersebut dikirimkan kembali ke API untuk segera berkomunikasi dan dikirimkan ke perangkat *mobile* lalu dihitung nilai similaritas dan ditampilkan kepada user.

### B. Struktur Data Buku Perpustakaan

Tabel merupakan sebuah struktur data yang biasanya digunakan untuk menyimpan keseluruhan data yang akan digunakan. Dalam pencarian buku saat ini pada perpustakaan UMN menggunakan database MySQL dengan versi 5.6.24 dan mempunyai 16.628 koleksi. Struktur tabel yang digunakan dapat dijabarkan sebagaimana terlihat pada Tabel 1.

Tabel 1. Struktur Data Buku Perpustakaan

Nama Tabel	Fungsi
Buku	Menyimpan data buku seperti judul, nomor fisik, nomor isbn.
Identitas Pengarang	Menyimpan identitas pengarang atau penulis buku.
Identitas Topik	Menyimpan identitas ulasan buku.
Pengarang	Menyimpan data pengarang tau penulis buku.
Tipe	Menyimpan penjelasan tipe buku.
Penerbit	Menyimpan nama penerbit.
Topik	Menyimpan keterangan topik dari koleksi buku.
Item	Menyimpan kode masing-masing buku.
Loan	Menyimpan data peminjaman.

### C. Alur Menu Aplikasi

Alur aplikasi dimulai dari *splash screen*, yaitu sebuah gambar sesaat setelah *user* membuka aplikasi UMN Library Catalog. Setelah *splash screen* selesai, aplikasi menampilkan menu yang dapat diakses oleh user diantaranya pencarian buku, tutorial aplikasi, tentang pembuat dan informasi aplikasi, informasi perpustakaan UMN, survei *online*, dan sosial media perpustakaan UMN.

Pada menu pencarian buku, *user* dapat melakukan pencarian buku yang diinginkan baik *normal search* maupun *advanced search*. Selanjutnya adalah tutorial aplikasi. Di dalam tutorial ini *user* disajikan gambar-gambar yang dapat digeser ke kiri dan ke kanan untuk panduan menggunakan aplikasi pencarian UMN Library Catalog.

Menu selanjutnya adalah biografi pembuat dan informasi aplikasi atau *about*. Pada menu ini terdapat informasi tentang pembuat aplikasi dan hal-hal yang

menyangkut tentang aplikasi pencarian buku seperti tambahan *external library* yang digunakan.

Menu informasi tentang perpustakaan UMN menyediakan informasi tentang perpustakaan UMN, mulai dari visi dan misi, tata cara peminjaman buku, *website*, dan kontak pihak perpustakaan.

Menu survei adalah tempat *user* dapat menilai aplikasi atau mengevaluasi aplikasi via *browser* dengan mengisi pertanyaan dari kuesioner.

Menu terakhir adalah menu sosial media perpustakaan UMN, dalam bagian ini terdapat sosial media yang digunakan oleh pihak perpustakaan untuk menginformasikan hal-hal baru yang ada pada perpustakaan. Jika *user* selesai dengan pencarian dan menemukan buku yang sesuai, maka *user* dapat langsung menekan tombol *back* dua kali untuk keluar dari aplikasi.

## VI. PEMBANGUNAN APLIKASI

Pembangunan aplikasi dilakukan berdasarkan arsitektur sistem. Tahapan pertama pemrograman web ditujukan untuk membangun Web Service yang bertugas untuk melakukan *encoding* dari bentuk balikan hasil *query* menjadi JSON Format. Web Service dibangun di dalam lingkungan PHP 5.6.8. Proses *term weighting* dan *stemming* juga dilakukan saat *request* data diterima dari aplikasi Android.

Fungsi *stemming* dilakukan pada saat *keyword LIKE* dipakai di dalam *query* untuk pencarian berdasar judul buku. Pada saat kata kunci yang dimasukan merupakan kata dasar, seperti “narasi”, maka judul buku yang mengandung kata dasar “narasi” tersebut dapat diambil seperti “menarasikan”. Selanjutnya, pada perhitungan *term weighting* dieksekusi menggunakan *query* dengan mode *natural language*. Penggunaan *natural language* merupakan model *vector space* yang dapat menghitung nilai pembobotan *term weighting*.

Selanjutnya dibangun aplikasi Android untuk minimal target Android JellyBean 4.1.2 dengan Android Standard Development Kit (SDK) yang menggunakan bahasa pemrograman Java. Terdapat 3 implementasi kode terkait dengan implementasi kode untuk metode *rochio*. Proses *tokenizing* dan *filtering* diimplementasikan di sini sebelum dikirim ke dalam Web Service untuk masuk ke dalam proses *stemming* dan *term weighting*.

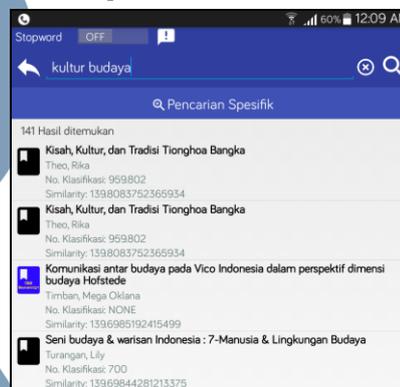
Setiap kata kunci yang dimasukkan akan diproses dengan cara melakukan pemisahan terhadap separator seperti tanda petik (‘), tanda *dash* (-), tanda titik (.), dan lain sebagainya. Setelah kata kunci berhasil menjadi *token*, maka tahap selanjutnya adalah *filter* kata kunci yang mengandung *stopword*. Setiap kata kunci akan di-*filter* jika menggunakan fitur *stopword* pada aplikasi.

Implementasi kode dilanjutkan dengan perhitungan rumus (1), (2), dan (3). Hasil perhitungan yang merupakan nilai tingkat similaritas antara judul dengan kata kunci yang dimasukkan. Pada judul dan kata kunci dihitung tingkat kesamaannya, semakin tinggi nilainya maka semakin baik tingkat similaritas antara judul dengan kata kunci yang dicari dan nantinya akan diurutkan menggunakan metode pengurutan berupa komparator.

Bagian terakhir dari pembangunan aplikasi adalah implementasi kode pengurutan yang digunakan untuk mengurutkan judul, nomor klasifikasi, nama pengarang, dan nilai similaritas. Data yang sudah ditemukan, diklasifikasi berdasarkan nilai similaritas. Seluruh nilai dibandingkan dengan mengimplementasikan komparator dan hasilnya adalah pengurutan nilai tertinggi berdasarkan tingkat similaritas antara judul dengan kata kunci yang dimasukkan.

## VII. DEMONSTRASI APLIKASI

### A. Screenshots Aplikasi



Gambar 3. Screenshot halaman pencarian

Gambar 4 merupakan halaman pencarian buku, bagian ini merupakan bagian utama dalam pencarian dimana *user* dapat mencari buku yang diinginkan. Pada bagian paling atas terdapat fitur *stopword* yang dapat menghilangkan kata hubung jika diaktifkan, dan di samping fitur *switch stopwords* terdapat sebuah tombol informasi untuk memberitahukan kegunaan dari fitur ini.

Selanjutnya di bawah fitur *stopword* terdapat bagian pencarian dengan memasukan kata kunci ke dalam kolom yang sudah disediakan. Lalu ada sebuah tombol (x) kecil untuk menghapus keseluruhan *input* yang diketikkan.

Selain itu, terdapat opsi untuk melakukan pencarian spesifik berdasarkan 5 *fields*, yaitu judul buku, ISBN, nama pengarang, tahun buku, dan nama dosen.

Setelah melakukan pencarian, maka aplikasi

memberitahukan berapa keluaran data yang dihasilkan dan tepat pada bagian bawah panel pencarian terdapat sebuah *list* untuk menampung semua hasil. Untuk melihat detail buku dapat dilakukan dengan cara mengklik bagian buku yang ingin dilihat.

### B. Simulasi Perhitungan Hasil

Berikut rumusan yang digunakan untuk melakukan perhitungan *precision* dan *recall* [14].

$$\text{Recall} = \frac{\text{Jumlah Dokumen Relevan pada Aplikasi}}{\text{Jumlah Dokumen Relevan pada Database}} \quad (4)$$

$$\text{Precision} = \frac{\text{Jumlah Dokumen Relevan dalam Aplikasi}}{\text{Jumlah Dokumen yang Didapat (Hasil)}} \quad (5)$$

Dalam simulasi ini dilibatkan 30 orang mahasiswa untuk memasukkan kata kunci pencarian. Pengambilan sampel kata kunci ini dilakukan dengan teknik *purposive sampling* pada mahasiswa yang sedang melakukan kegiatan pencarian buku di perpustakaan untuk melakukan pencarian lewat UMN Library Catalog. Dari proses ini dihasilkanlah 30 kata pencarian judul buku yang berbeda-beda untuk dihitung nilai *precision* dan *recall*-nya berdasarkan hasil pencarian. Tabel 2 memperlihatkan hasil perhitungan *precision* dan *recall* aplikasi yang dibangun.

Tabel 2. Hasil Perhitungan Precision-Recall UMN Library Catalog

No	K	A	B	C	D	P	R
1	Upacara ritual	15	13	2	13	87	100
2	Forward Chaining	13	11	2	11	85	100
3	Calculus	17	17	0	17	100	100
4	Photoshop	120	120	0	120	100	100
5	Perceived valence	2	2	0	2	100	100
6	Altman z	1	1	0	1	100	100
7	Pemahaman wajib pajak	301	58	243	58	19	100
8	Customer relationship	134	54	80	54	40	100
9	Probabilitas dan statistika	19	19	0	19	100	100
10	Science	79	76	0	76	100	100
11	Production	173	173	0	173	100	100
12	Cultures and communication	495	57	438	57	12	100
13	Delivering happiness	3	3	0	3	100	100
14	Filmmaker	21	21	0	21	100	100
15	Sociology (art)	11	3	8	3	27	100
16	Return on asset	91	63	28	63	69	100
17	Setting up scenes	14	8	6	8	57	100
18	3d modeling	241	143	98	143	59	100

No	K	A	B	C	D	P	R
19	Typography	14	14	0	14	100	100
20	Communicative toic	16	16	0	16	100	100
21	Investment	32	32	0	32	100	100
22	Manipulasi foto	95	26	69	26	27	100
23	Manajerial	13	13	0	13	100	100
24	Berjuta rasanya	1	1	0	1	100	100
25	Practical filmmaking	72	31	41	31	43	100
26	Sosiologi komunikasi	370	111	259	111	30	100
27	Investing	4	4	0	4	100	100
28	Kultur budaya	141	51	90	51	36	100
29	Financial accounting	526	167	359	167	32	100
30	Plagiarisme pelanggaran hak cipta	137	28	113	28	20	100
	Rata-Rata					71.4	100

Keterangan:

K : Keyword.

A : Hasil keseluruhan dokumen yang didapatkan pada aplikasi.

B : Hit (Hasil relevan pada aplikasi).

C : Miss (Hasil tidak relevan pada aplikasi).

D : Hit Database (dokumen relevan pada *database*).

P : Precision dalam %.

R : Recall dalam %.

Perhitungan juga dilakukan terhadap hasil pencarian pada SLiMS UMN sebagaimana terlihat pada Tabel 3.

Tabel 3. Hasil Perhitungan Precision-Recall SLiMS UMN

No	K	A	B	C	D	P	R
1	Upacara ritual	1	1	0	13	100	8
2	Forward Chaining	0	0	0	11	0	0
3	Calculus	13	13	0	17	100	76
4	Photoshop	109	109	0	120	100	91
5	Perceived valence	0	0	0	2	0	0
6	Altman z	245	0	245	1	0	0
7	Pemahaman wajib pajak	0	0	0	58	0	0
8	Customer relationship	8	8	0	54	100	15
9	Probabilitas dan statistika	5	5	0	19	100	26
10	Science	46	41	5	76	89	54
11	Production	48	48	0	173	100	28
12	Cultures and communication	4	4	0	57	100	7
13	Delivering happiness	3	3	0	3	100	100
14	Filmmaker	11	11	0	21	100	52

No	K	A	B	C	D	P	R
15	Sociology (art)	8	3	5	3	37	100
16	Return on asset	6	2	4	63	33	3
17	Setting up scenes	1	1	0	8	100	13
18	3d modeling	2	2	0	16	100	13
19	Typography	13	13	0	32	100	41
20	Communicative toeic	2	2	0	26	100	8
21	Investment	6	6	0	13	100	46
22	Manipulasi foto	1	1	0	1	100	100
23	Manajerial	8	3	5	3	37	100
24	Berjuta rasanya	6	2	4	63	33	3
25	Practical filmmaking	2	2	0	31	100	6
26	Sosiologi komunikasi	4	4	0	111	100	4
27	Investing	3	3	0	4	100	75
28	Kultur budaya	0	0	0	51	0	0
29	Financial accounting	27	26	1	167	96	16
30	Plagiarisme pelanggaran hak cipta	9	9	0	28	100	32
	Rata-Rata					78.5	33.77

Setelah perhitungan pada kedua sistem pencarian, dicarilah nilai F-Measure sebagai nilai rata-rata harmonis antara perhitungan *precision* dan *recall*. Nilai F-Measure dapat dihitung dari 0 hingga 1 atau dalam persentase 0% - 100%.

$$F - Measure = 2 * \frac{precision + recall}{precision + recall} \quad (6)$$

F-Measure untuk UMN Library Catalog menghasilkan nilai sebesar 83,33%.

$$F - Measure = \frac{2 * (71.43 * 100)}{(71.43 + 100)} = 83,33\%$$

Sedangkan F-Measure untuk SLiMS UMN menghasilkan nilai yang lebih rendah, yaitu 47,22%.

$$F - Measure = \frac{2 * (78.5 * 33.77)}{(78.5 + 33.77)} = 47,22\%$$

Dari sini dapat terlihat perbedaan relevansi berdasarkan F-Measure antara aplikasi Android UMN Library Catalog dengan SLiMS UMN yang berbasis web sebesar 36,11%. Perbedaan ini membuktikan bahwa UMN Library Catalog mampu menyajikan pencarian yang lebih relevan dari SLiMS UMN.

#### VIII. EVALUASI APLIKASI

Pada tahap simulasi perhitungan nilai hasil pencarian, telah dilibatkan 30 orang mahasiswa. Setelah melakukan simulasi tersebut, para mahasiswa dijadikan responden dalam survey *user experience*

sebagai proses evaluasi aplikasi secara menyeluruh. Terdapat 12 pertanyaan dimana 8 pertanyaan kuesioner merupakan penilaian terhadap faktor-faktor *user experience*, 3 pertanyaan terkait pengalaman pengguna terhadap hasil pencarian yang dirasakan dan 1 pertanyaan mengenai kata kunci yang dimasukkan. Tabel 4 memperlihatkan rekapitulasi hasil survey terhadap *user experience* aplikasi yang dibangun.

Tabel 4. Hasil Survey User Experience Aplikasi

No	Faktor	1	2	3	4	5	%
1	Attractiveness	0	0	3	14	13	87
2	Perspiciuity#1	0	0	3	14	13	79
3	Perspiciuity#2	0	0	0	3	27	86
4	Efficiency	0	0	0	2	28	95
5	Dependability#1	0	0	1	9	20	87
6	Dependability#2	0	0	1	11	18	90
7	Stimulation	√	√	√	√	√	100
8	Novelty	0	0	0	9	21	95
9	Relevansi	0	0	0	5	25	96
10	Akurasi	0	0	0	5	25	96
11	Detail	0	0	0	13	17	92

Dari analisa faktor *user experience* pada masing-masing faktor, nilai terendah didapatkan pada faktor *perspicuity* pada pertanyaan pertama mengenai kefamiliaran menggunakan aplikasi. Hasil didapatkan hanya sebesar 79% yang berarti responden belum sepenuhnya merasa familiar dengan aplikasi pencarian buku yang dirancang dan dibangun. Selanjutnya, nilai tertinggi didapatkan pada faktor *stimulation* yang menanyakan tentang keinginan untuk menggunakan aplikasi, mendapat tanggapan sangat baik yaitu 100% responden ingin menggunakan aplikasi ini untuk melakukan pencarian, baik buku maupun dokumen skripsi ataupun jurnal-jurnal koleksi perpustakaan UMN.

Setelah didapatkan hasil perhitungan masing-masing faktor *user experience*, selanjutnya adalah perhitungan nilai rata-rata dari keseluruhan faktor yang dapat dijabarkan sebagai berikut.

$$Rata - rata \ user \ experience = \frac{87 + 79 + 86 + 95 + 87 + 90 + 100 + 95 + 96 + 96 + 92}{11} = 91,18\%$$

Hasil perhitungan ini menunjukkan evaluasi aplikasi dari segi pengalaman user mendapatkan nilai yang sangat baik. Aplikasi telah mampu menyediakan layanan yang tidak hanya memberikan hasil yang relevan namun juga dapat memenuhi kebutuhan pengalaman pengguna dalam melakukan pencarian bahan pustaka di Perpustakaan UMN. Aplikasi telah

dipublikasikan lewat Google Playstore dengan nama UMN Library Catalog sehingga seluruh *civitas academica* UMN dapat menggunakan aplikasi tersebut dengan gratis.

#### IX. KESIMPULAN

Aplikasi UMN Library Catalog untuk pencarian buku pada koleksi perpustakaan UMN dengan metode *rocchio relevance feedback* berhasil dirancang dan dibangun dengan bagian arsitektur yang meliputi *database*, API, *platform* Android dan dapat mengeluarkan hasil pencarian yang lebih relevan sesuai dengan rancangan serta dapat berjalan dengan spesifikasi minimum Android 4.1.2.

Analisa *user experience* terhadap aplikasi yang meliputi faktor *attractiveness*, *perspicuity*, *efficiency*, *dependability*, *stimulation*, *novelty*, dan beberapa faktor umum seperti tingkat relevansi, akurasi, dan detail informasi buku menggunakan teknik pengambilan survei dengan kuesioner berhasil dilaksanakan. Proses pengujian terhadap aplikasi disimpulkan bahwa responden menyukai menggunakan aplikasi pencarian dan mendapatkan persentase faktor secara menyeluruh yaitu 91.18%. Hal ini menunjukkan bahwa aplikasi pencarian berdasarkan tingkat relevansi menggunakan *platform mobile* khususnya Android disukai oleh para responden.

Pengujian aplikasi pada data koleksi buku perpustakaan UMN menggunakan perhitungan *precision*, *recall*, dan *F-Measure* menghasilkan nilai *precision* 71.43%, *recall* 100%, dan *F-Measure* 83.33%. Hal ini bertujuan untuk mengukur tingkat relevansi yang dihasilkan oleh aplikasi pencarian UMN Library Catalog. Selain itu, *F-Measure* atau rata-rata relevansi yang dihasilkan pada aplikasi UMN Library Catalog unggul sebesar 36.11% dari sistem pencarian SLiMS. Dapat disimpulkan bahwa UMN Library Catalog mampu menyajikan pencarian yang relevan sesuai dengan kata kunci yang dimasukkan oleh *user*.

#### TERIMA KASIH

Terima kasih kami haturkan kepada Ibu Stefani Orisa Mahardhini sebagai *library superintendent* dan juga Bapak Dony Pratidana sebagai *library information literacy officer* yang telah menyediakan waktu untuk bekerjasama, menjadi narasumber, berdiskusi dan memberikan izin untuk menyelenggarakan penelitian di dalam Perpustakaan UMN.

#### DAFTAR PUSTAKA

- [1] Mesran, Implementasi Algoritma Brute Force Dalam Pencarian Data Katalog Buku Perpustakaan. *Majalah Ilmiah Informasi dan Teknologi Ilmiah (INTI)*. 2014. Vol. III, No.1.
- [2] Tjandarini, A.B., Lemantara, J., Pembuatan Prototipe Tempat dan Aplikasi Pengembalian Koleksi Perpustakaan

- Secara Mandiri, *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*. 2015. Vol. IV, No.1.
- [3] Barve, S., & Dahibhate, N. B., "Open source software for library services", *DESIDOC Journal of Library & Information Technology* 32(5) [Online]. 2012. <https://search.proquest.com/docview/1413408771>, Tanggal Akses : 14 Februari 2017.
- [4] Sudewo, L., Implementasi Algoritma Levenshtein Distance Pada Fitur Autocomplete Senayan Library Management System (Studi Kasus: SLiMS Universitas Multimedia Nusantara), *Skripsi Program Sarjana Teknik Informatika Universitas Multimedia Nusantara*. 2015, Tidak Dipublikasikan.
- [5] Albert, Kristanda, M.B., dan Hansun, S., Studi Kelayakan dan Perancangan Aplikasi Pencarian Buku pada Katalog Perpustakaan Menggunakan Rocchio Relevance Feedback. *Jurnal ULTIMATICS*, 2016. Vol.VIII, No. 1, Hal. 37-43.
- [6] Fatmawati, E., Trend Terkait M-Library Untuk Perpustakaan Masa Depan. *Visi Pustaka*, 2012, Vol. 14, No. 3.
- [7] Tari, Luis., Tu, Phan Huy., Hakenberg, Jorg., Chen, Yi., Son, Tran Cao., Gonzalez, Graciela., Baral, Chitta. "Incremental Information Extraction Using Relational Databases". *Journal IEEE Transactions on Knowledge and Data Engineering*, IEEE Educational Activities Department, Piscataway, New Jersey, USA. 2012. Volume 24, Issue 1, pp. 86-99.
- [8] Cooper, James W., Prager, John M. "Anti-Serendipity: Finding Useless Documents and Similar Documents". *Proceedings of the 33rd Hawaii International Conference on System Sciences*. 2012. Volume 3, Page 3008.
- [9] Onoda, Takashi., Murata, H., Yamada, S. "Comparison of Performance for SVM based Relevance Feedback Document Retrieval in several Vector Space Models". *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, IEEE Computer Society, Washington DC, USA. 2011. Volume 03, Pages 169-172.
- [10] Shankar, Shrikanth., Karypis, George. "Weight adjustment schemes for a centroidbased classifier". *Proceedings of KDD Workshop on Text Mining*. 2012. Volume 6440 pp. 86-97.
- [11] Park, Sun., Cha, Byung Rae. "Query based Personalized Summarization Agent using NMF and Relevance Feedback". *Proceedings of the 4th International Conference on Ubiquitous Information Management and Communication*, ICUIMC, New York, USA. 2011. Article No. 61.
- [12] Yugianus, P., Dachlan, H. S., dan Hasanah, R. N., Pengembangan Sistem Penelusuran Katalog Perpustakaan dengan Metode Rocchio Relevance Feedback. *Jurnal Electrics Communications Controls Informatics Systems (ECCIS)*. 2013. Vol. 7 No. 1.
- [13] Kusuma, W.A., Noviasari, V., & Marthasari, G.I., Analisis Usability dalam User Experience pada Sistem KRS-Online UMM menggunakan USE Questionnaire, *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*. 2016. Vol.5, No. 4.
- [14] Jizba, R., "Measuring Search Effectiveness". *Creighton University Health Science Library and Learning Resource Center*. 2006.
- [15] Lestari, P. N., Santosa, P. I., dan Ferdiana, R., Pengukuran Pengalaman Pengguna Dengan Menggunakan Sistem Informasi Akademik. *Seminar Nasional Teknologi Informasi dan Komunikasi*. 2016.
- [16] Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S. A design science research methodology for information systems research. *Journal of management information systems*. 2007 Dec 1;24(3):45-77.
- [17] Albert, Rancang Bangun Aplikasi Android Pencarian Buku Perpustakaan Menggunakan Metode Rocchio Relevance Feedback (Studi Kasus: Perpustakaan UMN), *Skripsi Program Sarjana Teknik Informatika Universitas Multimedia Nusantara*. 2016, Tidak Dipublikasikan.
- [18] Selberg, E. W., "Information Retrieval Advances Using Relevances Feedback". *Department of Computer Science and Engineering University of Washington*, 2011.

- [19] Liddy, E. D., "Automatic Document Retrieval. Encyclopedia of Language and Linguistic. 2<sup>nd</sup> Ed". 2011. Philadelphia,
- [20] Uden dan Mark, V., "Rocchio Relevance Feedback in Learning Classification Algorithms". *Department of Computing Science University of Nijmegen*. 2011.

