

Pembuatan Aplikasi *Electronic Guided Bible Study Process* dengan *Biblical Recommendation System*

Agus Setiawan Kuswandi¹, Iwan Njoto Sandjaja², Alexander Setiawan³
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121-131, Surabaya, 60236

Telp. (031) – 2983452

wawan.park.95@gmail.com¹, iwanns@petra.ac.id², alexander@petra.ac.id³

ABSTRAK

Dengan kemajuan teknologi, berbagai kebutuhan sehari-hari manusia dapat dipenuhi dengan mudah. Kebutuhan komunikasi, transportasi, hingga makanan dapat dipenuhi dengan bantuan teknologi. Dalam cara pandang Kristen, manusia memiliki kebutuhan utama untuk mengenal dan memiliki hubungan yang dekat dengan Penciptanya. Tetapi, teknologi belum digunakan dengan maksimal untuk menolong pemenuhan kebutuhan ini. Orang Kristen percaya bahwa Alkitab adalah Firman Allah yang sanggup untuk menjawab kebutuhan utama manusia. Oleh karena itu, penulis membuat suatu aplikasi web untuk menolong orang Kristen memahami dan mendalami Alkitab. Aplikasi ini menyediakan fitur pencarian ayat dan informasi dengan prinsip *relevant search* menggunakan *Elasticsearch*. Dengan fitur tersebut, hasil pencarian dapat ditampilkan menurut apa yang menjadi kebutuhan pengguna yang melakukan pencarian. Selain itu, aplikasi juga dapat digunakan untuk melakukan Pendalaman Alkitab dan memberikan rekomendasi topik Pendalaman Alkitab bagi pengguna dengan menggunakan algoritma *Item-based Collaborative Filtering with Item Category Distance and Interestingness Measure*. Hasil dari pengujian aplikasi menunjukkan bahwa fitur-fitur dalam aplikasi ini telah berjalan dengan baik dan dapat menolong penggunaannya untuk dapat mendalami Alkitab dengan lebih baik.

Kata Kunci: Aplikasi Web, *Elasticsearch*, *Item-based Collaborative Filtering with Item Category Distance and Interestingness Measure*, *Relevant Search*, Kekristenan, Pendalaman Alkitab.

ABSTRACT

With technological advancements, human needs can be fulfilled easily. The needs of communication, transportation, and even food can be fulfilled with the help of technology. From the Christian Worldviews, human has basic needs of knowing and having an intimate relationship with his Creator. However, technology isn't yet used to its full potential to help the fulfillment of this necessity. Christians believe that the Bible is the Word of God which can give answers to the human's basic needs. Therefore, the writer try to develop a Web Application to help Christians study and understand the Bible. The Application provides verse and information search features with relevant search principle using Elasticsearch. With that features, search results shown based on

user needs. On the other hand, user can use application to do Bible Study and give Bible Study material recommendation to the user using Item-based Collaborative Filtering with Item Category Distance and Interestingness Measure algorithm. The testing results of the application shows that the features in this application are working properly and able to help the users to understand the Bible better.

Keywords: *Web Application, Elasticsearch, Item-based Collaborative Filtering with Item Category Distance and Interestingness Measure, Relevant Search, Christianity, Bible Study.*

1. LATAR BELAKANG

Seiring kemajuan teknologi, kebutuhan sehari-hari manusia semakin dapat dipenuhi dengan mudah. Khususnya di era digital, semakin besar kebutuhan manusia akan berinteraksi dan berkomunikasi. Media sosial menjadi jawaban bagi kebutuhan ini. Manusia juga membutuhkan transportasi yang mudah digunakan. Saat ini beragam aplikasi taksi atau ojek online dapat memenuhi kebutuhan ini.

Dalam cara pandang dunia Kristen, manusia memiliki kebutuhan utama dalam hidup yaitu untuk mengenal dan memiliki hubungan yang dekat dengan Tuhan (Mazmur 73:25-26¹, Filipi 3:10²). Hal itu dapat dicapai dengan persekutuan dengan Firman Tuhan, yaitu melalui membaca dan mendalami Alkitab (Matius 4:4³) [1].

Akan tetapi, pendalaman Alkitab sering dinilai sulit dan membosankan. Hal ini disebabkan karena ketika mendalami Alkitab seseorang harus membuka berbagai buku yang tebal-tebal

¹ “Siapa gerangan ada padaku di sorga selain Engkau? Selain Engkau tidak ada yang kuingini di bumi. Sekalipun dagingku dan hatiku habis lenyap, gunung batuku dan bagianku tetaplah Allah selama-lamanya.”

² “Yang kukehendaki ialah mengenal Dia dan kuasa kebangkitan-Nya dan persekutuan dalam penderitaan-Nya, di mana aku menjadi serupa dengan Dia dalam kematian-Nya.”

³ “Tetapi Yesus menjawab, ‘Ada tertulis: Manusia hidup bukan dari roti saja, tetapi dari setiap Firman yang keluar dari mulut Allah.’”

seperti kamus Alkitab dan tafsiran sebagai bahan untuk menolong Pendalaman Alkitab.

Dengan realita kemajuan teknologi yang sudah dibahas di atas, Pendalaman Alkitab pun juga dapat dipermudah dengan memanfaatkan teknologi. Maka, dibuatlah sebuah aplikasi web untuk menolong kegiatan Pendalaman Alkitab. Aplikasi ini akan menyediakan bahan-bahan yang berkualitas dan dilengkapi dengan fitur *relevant search* yang akan dibuat dengan menggunakan *Elasticsearch*. Fitur ini akan menolong pengguna mencari bahan sesuai dengan kebutuhan ketika melakukan Pendalaman Alkitab.

2. RELEVANT SEARCH

Dalam kehidupan sehari-hari, manusia sering melakukan pencarian. Dalam ranah pemanfaatan teknologi, misalnya seseorang berbelanja di toko *online*. Tentu kata kunci yang dicari diharapkan dapat memberikan hasil sesuai dengan apa yang dicari. Inilah yang disebut dengan prinsip *relevant search*.

Relevant Search adalah pencarian yang dilakukan oleh *search engine* yang menampilkan hasil yang paling sesuai dengan keinginan pengguna. *Relevant search* bukanlah pencarian yang ajaib ataupun dapat membaca pikiran pengguna. Akan tetapi, perlu ada proses yang sangat teliti, bagaimana dokumen-dokumen atau data-data disimpan sehingga memungkinkan untuk dicari menggunakan kata kunci yang juga diproses sedemikian rupa tepat setelah pengguna mengetikkan kata kunci pencarian [4].

Apabila suatu aplikasi hendak menyediakan fitur pencarian, aplikasi tersebut harus mempertimbangkan tiga hal penting : hasil yang ditampilkan relevan, menampilkan statistik atau informasi berkaitan dengan hasil yang ditampilkan, dan keduanya dapat dilakukan dengan cepat. Di sinilah *search engine* seperti *Elasticsearch* dapat dimanfaatkan karena memang *Elasticsearch* diciptakan untuk memenuhi kebutuhan tersebut [2].

2.1 Elasticsearch

Elasticsearch adalah suatu *open-source search engine library* yang dibangun berdasarkan *Apache Lucene* yang juga merupakan *search engine sejenis* yang dibangun untuk digunakan pada aplikasi Java. *Elasticsearch* menggunakan kemampuan dari *Apache Lucene* dan meningkatkannya untuk menyediakan proses penyimpanan, *indexing*, dan pencarian yang lebih mudah dan cepat. Aplikasi yang memanfaatkan *Elasticsearch* tidak harus menggunakan bahasa pemrograman Java seperti ketika memanfaatkan *Apache Lucene*. Aplikasi dapat mengirim data melalui HTTP dalam format JSON [2].

2.2 Inverted Index

Pencarian yang cepat dengan *Elasticsearch* dimungkinkan dengan mengindeks setiap data atau yang tersedia. *Index* adalah struktur data yang memungkinkan data dapat ditemukan dengan cepat. Contoh dari struktur data seperti ini dapat ditemukan pada halaman *index* dalam hampir semua buku. Membuat data *index* dapat dilakukan dengan menambah *field* baru pada *database* yang berisi posisi data itu berada dalam dokumen. Dalam sistem

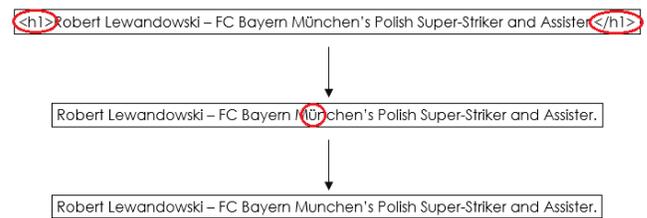
Elasticsearch, proses ini disebut dengan proses *inverted indexing*. Dengan *inverted indexing*, ketika pengguna mencari suatu kata kunci, hasil pencarian dapat langsung didapatkan dari struktur data *index* tanpa harus melakukan pencarian dengan mencocokkan setiap kata satu demi satu dari setiap dokumen.

2.3 Komponen Analisis

Proses analisis data terdiri dari 3 langkah, yaitu *character filtering*, *tokenization*, dan *token filtering* [4]. Sebagai contoh kalimat “<h1>Robert Lewandowski – FC Bayern München’s Polish Super-Striker and Assister.</h1>” akan dianalisis dengan 3 langkah yang telah disebutkan di atas.

2.3.1 Character Filtering

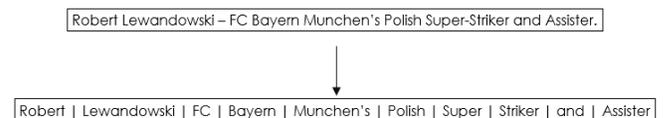
Dengan *character filtering*, tag HTML dalam suatu kalimat akan dihilangkan sehingga menyisakan isinya, lalu *ASCII folding filter* mengubah karakter bertanda dalam beberapa bahasa asing menjadi huruf biasa. Proses ini ditunjukkan dalam Gambar 1.



Gambar 1. Proses *character filtering*

2.3.2 Tokenization

Dalam proses *tokenization*, kalimat dipisahkan sehingga membentuk beberapa kata (seperti dalam penjelasan *inverted index* pada subbab 2.2.). Dasar pemisahan kata pada umumnya adalah spasi. Tetapi contoh dalam kalimat ini, karakter setrip dan titik juga dipakai sebagai pemisah atau dengan kata lain dihilangkan. Misalnya frase “super-striker” apabila disimpan dalam bentuk yang demikian maka sulit untuk mendapatkan kecocokan dengan kata kunci yang dicari pengguna. Mengingat, pengguna kemungkinan besar akan mencari dengan kata kunci “super striker”, sehingga kata tersebut dipisahkan. Proses *tokenization* ditunjukkan dalam Gambar 2.

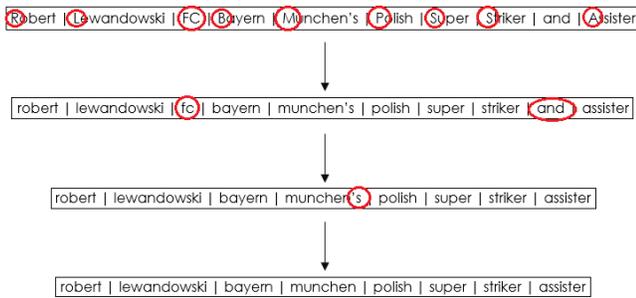


Gambar 2. Proses *tokenization*

2.3.3 Token Filtering

Proses *token filtering* akan menormalisasi token-token yang didapatkan dari proses *tokenization*. Proses yang dilakukan misalnya mengubah token dalam bentuk *lowercase*. Lalu, menghilangkan *stop-word* seperti “and” dan “fc” yang artinya adalah *football club* (klub sepak bola). Kata ini sangat umum digunakan dalam nama klub sepak bola. Terakhir, tanda *possession*

(kepemilikan) “'s” dihilangkan. Proses ini ditunjukkan dalam Gambar 3.



Gambar 3. Proses token filtering

3. RECOMMENDATION SYSTEM

Recommendation system adalah sistem yang dibuat untuk menyediakan bahan yang tepat di antara banyaknya kumpulan bahan sebagai solusi bagi usaha manusia yang melakukan pekerjaan tersebut secara manual [3]. Dalam Recommendation system, terdapat 3 metode yang umum digunakan :

- Content-based Recommendation.** Dalam metode ini, informasi mengenai suatu objek menjadi dasar untuk menyediakan rekomendasi bagi pengguna. Suatu objek yang memiliki informasi yang mirip dengan minat pengguna akan menjadi objek yang direkomendasikan bagi pengguna tersebut. Oleh karena itu, kemiripan antara objek-objek menjadi informasi yang penting bagi metode ini. Sebagai contoh, untuk merekomendasikan film kepada pengguna, maka diperlukan informasi film seperti apa yang disukai pengguna lalu sistem akan menampilkan film-film yang mirip dengan minat pengguna.
- Collaborative Filtering.** Dalam metode ini, fungsi rekomendasi bekerja dengan cara mencari informasi mengenai minat pengguna lain yang memiliki minat yang mirip dengan pengguna atau dua item yang diminati oleh beberapa pengguna yang sama.
- Hybrid Approach.** Metode ini menggabungkan kedua metode sebelumnya. Metode ini menjadi solusi bagi kelemahan yang muncul jika hanya menggunakan metode Content-based atau Collaborative filtering [3].

3.1 Algoritma Item-based Collaborative Filtering with Item Category Distance and Interestingness Measure (ICF - ICDIM)

Pada dasarnya, item-based collaborative filtering menekankan perbandingan antar item. Apabila item I diminati oleh user A, B, C, dan D, maka item J yang diminati oleh user A, C, dan D kemungkinan besar juga akan diminati oleh user B. Dengan demikian, item J dapat direkomendasikan kepada user B. Ini adalah contoh dari item-based collaborative filtering. Untuk melakukan fungsi tersebut antara item yang satu dengan item yang lain perlu diukur kemiripannya (similarity). Dalam skripsi ini, akan digunakan Item-based Collaborative Filtering yang sudah

dimodifikasi yang disebut dengan Item-based Collaborative Filtering with Item Category Distance and Interestingness Measure (ICF - ICDIM) Recommendation Algorithm. Umumnya perhitungan yang dilakukan untuk mengukur kemiripan adalah menggunakan rumus cosinus sebagai berikut :

$$sim(u,v) = \cos(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \times \|\vec{v}\|} = \frac{\sum_{i=1}^m r_{iu} \times r_{iv}}{\sqrt{\sum_{i=1}^m r_{iu}^2} \sqrt{\sum_{i=1}^m r_{iv}^2}} \quad (1)$$

Dimana, r_{iu} adalah nilai yang diberikan user untuk item u, r_{iv} adalah nilai yang diberikan user untuk item v, dan m adalah banyaknya user. Akan tetapi, rumus ini akan kehilangan nilai kemiripan apabila user tidak memberikan nilai/rating pada suatu item yang diukur (atau dengan kata lain rating = 0). Perhitungan kemiripan antara 2 item dapat dilakukan dengan menggunakan alternatif rumus probabilitas bersyarat, yaitu sebagai berikut :

$$sim(u,v) = P(v|u) = \frac{Freq(uv)}{Freq(u)} \quad (2)$$

Dimana, $freq(uv)$ adalah banyaknya user yang memberikan nilai (secara praktis misalnya user tersebut membeli item tersebut) untuk kedua item yaitu item u dan v, $freq(u)$ adalah banyaknya user yang memberikan nilai untuk item u (atau membeli item u). Akan tetapi, nilai similarity dengan perhitungan ini akan menjadi tinggi apabila item v mendapatkan nilai yang tinggi dari banyak user (atau banyak user yang membeli item v), bukan karena item u dan item v diberi rating secara oleh user yang sama. Solusinya, rumus ini dikontrol dengan membagnya dengan banyaknya item v yang diberi rating oleh user :

$$sim(v,u) = \frac{Freq(uv)}{Freq(v) \times (Freq(u))^\alpha} \quad (3)$$

Dengan α adalah variabel kontrol. Selain itu, nilai ini juga dipertajam dengan menghitung kemiripan kedua item dalam hal kategorinya $d(u,v)$.

$$d(u,v) = \begin{cases} 0 & u = v \\ l(u,v)/H & u \neq v \end{cases} \quad (4)$$

Dimana, apabila kedua item ada pada kategori yang sama ($u = v$), maka nilai $d(u,v) = 0$. Sedangkan, apabila kedua item memiliki kategori yang berbeda, maka nilai $d(u,v)$ adalah jarak terjauh terhadap akar sub-kategori yang sama ($l(u,v)$) dibagi tinggi categories tree (H).

Lalu, penting juga untuk menghitung rule interestingness (RI) antar item yang dirumuskan sebagai berikut :

$$RI(u,v) = P(u,v) - P(u) * P(v) \\ = \frac{Freq(uv)}{m} - \frac{Freq(u) \times Freq(v)}{m^2} \quad (5)$$

Dimana, $\text{freq}(uv)$, seperti di atas adalah banyaknya *user* yang membeli/memberi rating untuk *item* u dan *item* v , serta m adalah banyaknya *user*. Sehingga, rumus *similarity* yang telah dipertajam dirumuskan seperti gambar berikut :

$$\text{sim}(u,v) = \frac{\text{Freq}(uv)}{\text{Freq}(u) + (\text{Freq}(v))^\alpha + \sum_{i=1}^m |r_{iu} - r_{iv}|} + \beta \times RI(u,v) + \gamma \times s_{uv} \quad (6)$$

Dimana, $s_{uv} = 1 - d(u,v)$, sedangkan α , β , dan γ adalah variabel kontrol yang telah diuji memberikan hasil yang maksimal pada nilai berturut-turut $0.6 \geq \alpha \geq 0.3$, $0.9 \geq \beta \geq 0.4$, $0.8 \geq \gamma \geq 0.4$. Pada akhirnya, untuk memprediksi rating/nilai yang diberikan *user* u untuk *item* i , ditunjukkan dengan rumus sebagai berikut :

$$P_{ui} = \bar{r}_i + \frac{\sum_{j \in I_{nei}} \text{sim}(i,j)(r_{uj} - \bar{r}_j)}{\sum_{j \in I_{nei}} \text{sim}(i,j)} \quad (7)$$

Dimana, \bar{r}_i adalah rata-rata rating untuk *item* i , I_{nei} adalah *top-N nearest neighbor* dari *item* i , dan r_{uj} adalah nilai rating dari *user* u untuk *item* j . Dalam tulisan ini, rumus ini digunakan sebagai algoritma *item-based collaborative filtering with item category distance and interestingness measure* (ICF-ICDIM) [5].

4. DESAIN SISTEM

4.1 Penggunaan *Elasticsearch*

Penggunaan *Elasticsearch* dalam aplikasi ini diawali dengan melakukan migrasi data dari *database* MySQL menuju *Elasticsearch*. Proses migrasi data menuju *Elasticsearch* akan dikerjakan dengan Proses *Indexing*. Sedangkan, proses mendapatkan data sekaligus nilai relevansi berdasarkan proses analisisnya akan dikerjakan dengan Proses *Query*.

4.1.1 Proses Mapping dan Indexing

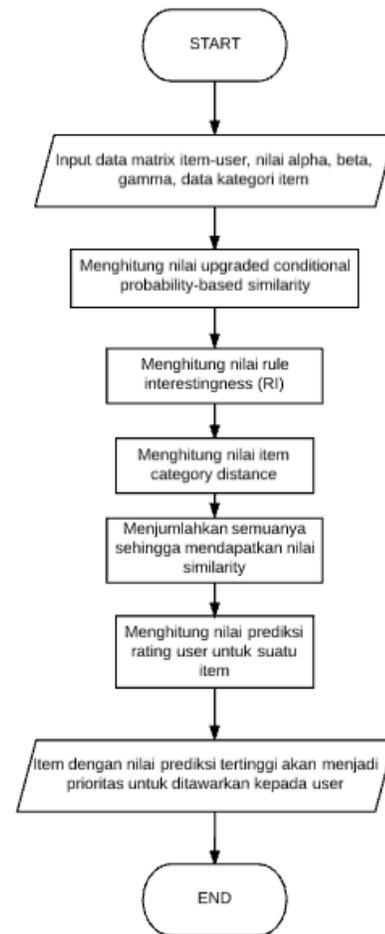
Kedua proses ini adalah proses yang tidak dapat dipisahkan. *Mapping* adalah proses pengaturan sebelum membuat *index* (atau *database* dalam terminologi MySQL) termasuk mengatur *analyzer* sebagai bagian yang penting dari proses ini. *Indexing* adalah proses memasukkan data pada *index Elasticsearch* (proses *insert*). Apabila proses mapping mempersiapkan *analyzer*-nya, maka pada proses *indexing* inilah data sesungguhnya dianalisis yang pada intinya diuraikan dalam bentuk *token-token* yang diperlukan untuk proses *search*.

4.1.2 Proses Query

Fitur kunci dalam aplikasi ini adalah fitur *relevant search*. Sehingga, proses *query* adalah proses yang penting untuk menghasilkan pencarian yang relevan. Proses *query* memasukkan proses analisis yang sama dengan proses *mapping* dan *indexing*. Ini bertujuan supaya hasil analisis *keyword* yang dicari dapat cocok dengan hasil analisis data yang sudah di-*index*.

4.2 Proses ICF - ICDIM

Item-based collaborative filtering akan dimanfaatkan untuk memberikan rekomendasi topik pendalaman Alkitab bagi *user*. Prinsipnya, setiap *item* direlasikan dengan setiap *user* sehingga membentuk *item-user matrix*. Isinya adalah nilai *rating* yang *user* berikan kepada *item* yang bersangkutan. *Item-item* yang belum diberi *rating* oleh *user-user* yang bersangkutan (*rating* = 0) dapat diprediksi nilai *rating*-nya dengan rumus P_{ui} seperti yang sudah dijelaskan di Subbab 3.1. *Item* dengan nilai prediksi *rating* tertinggi untuk suatu *user* akan direkomendasikan kepada *user* tersebut seperti fitur “*you may also like ...*” dalam beberapa *web e-commerce*. Gambar 4 adalah *flowchart* langkah-langkah untuk menghitung nilai prediksi *rating item*.



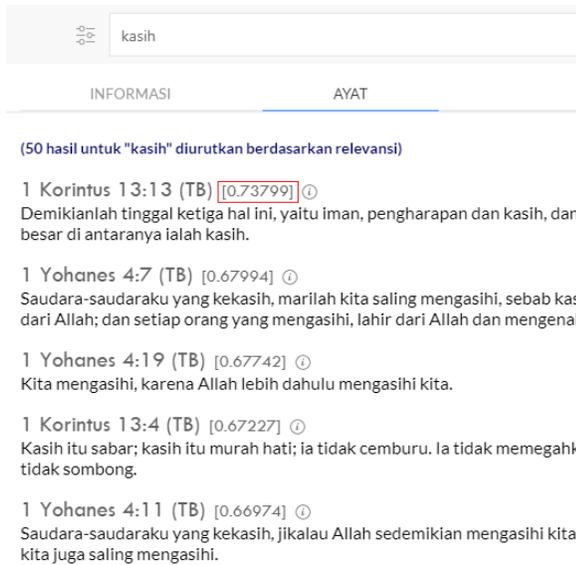
Gambar 4. *Flowchart* proses rekomendasi dengan ICF - ICDIM

5. PENGUJIAN

Pengujian akan dilakukan terhadap fitur-fitur utama aplikasi.

5.1 Fitur Pencarian Ayat

Gambar 5 menampilkan pengujian fitur pencarian ayat dengan kata kunci pencarian “kasih”.



Gambar 5. Hasil pencarian ayat “kasih”

Hasil pencarian ayat ditampilkan dengan urutan berdasarkan bobot nilai tertinggi (tanda merah). Bobot nilai ini ditentukan dari kata “kasih” yang ditemukan dalam isi Alkitab TB, TL, AYT, dan BIS, kata “kasih” ditemukan dalam judul perikop TB, TL, AYT, dan BIS, kata “kasih” dalam nama topik yang mengandung ayat tersebut, dan kata “kasih” yang berasal dari bahasa asli yang ada dalam ayat tersebut.

Pencarian ayat menghasilkan hasil paling akurat apabila melakukan *Elasticsearch query* terhadap seluruh *data* (31102 *record* untuk isi Alkitab dan judul perikop, 5835 *record* untuk sistem topik), tetapi membutuhkan proses *query* yang lama. Tanpa pengaturan, aplikasi akan menampilkan 50 hasil ayat dengan nilai tertinggi dalam fitur pencarian ayat ini. Untuk mempercepat proses *query*, tetapi dengan konsekuensi mengorbankan akurasi penilaian, maka digunakan variabel *limit*. Nilai *limit* akan dikalikan dengan berapa hasil yang pengguna ingin tampilkan sehingga mendapatkan nilai berapa banyak *record* yang harus diproses oleh *Elasticsearch*. Misalnya, *limit* = 3, maka banyak *record* yang harus diproses *Elasticsearch* untuk 50 hasil ayat yang ditampilkan adalah $3 \times 50 = 150$ *record*. Tabel 1 menampilkan perbandingan waktu pencarian ayat dengan nilai *limit* yang berbeda.

Tabel 1. Perbandingan waktu pencarian (dalam detik)

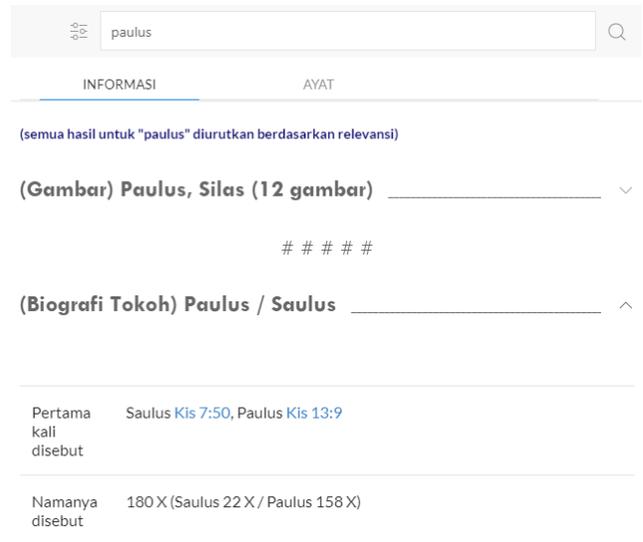
Kata kunci	limit = 3	limit = 5	semua record
kasih	6	8	11
tritunggal	1	1	2
ucapan bahagia	5	6	10
tulisan ... ⁴	10	10	> 250
keselamatan ... ⁵	11	12	24

⁴ tulisan yang dari allah bermanfaat untuk mengajar dalam kebenaran

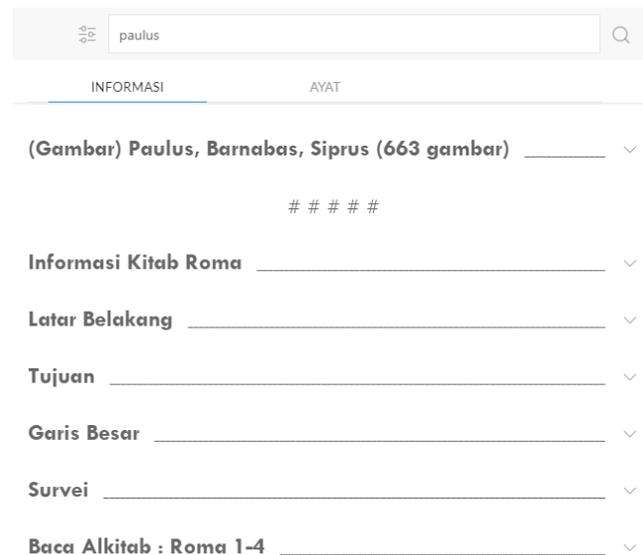
⁵ keselamatan oleh iman

5.2 Fitur Pencarian Informasi

Pada fitur pencarian informasi, hasil pencarian adalah berupa bahan seperti informasi kitab, biografi tokoh, kamus, artikel, gambar, video, dan peta. Apabila mencari nama tokoh maka biografi akan muncul paling atas, diikuti dengan bahan lain yang berkaitan dengan nama tokoh tersebut. Apabila mencari nama tempat, maka peta akan muncul paling atas. Sebagai contoh, hasil pencarian kata kunci “paulus” ditampilkan pada Gambar 6 dan 7. Informasi surat Roma muncul karena Paulus adalah penulis surat Roma.



Gambar 6. Hasil pencarian informasi “paulus” (bagian 1)



Gambar 7. Hasil pencarian informasi “paulus” (bagian 2)

5.3 Fitur Rekomendasi Topik PA

Fitur PA akan menolong pengguna untuk melakukan PA secara digital. Sebelum PA dimulai, pengguna dapat mencari dan memilih

topik PA yang ingin digunakan. Salah satu fitur untuk menolong pengguna memilih topik PA adalah fitur rekomendasi topik PA. Sebagai contoh terdapat pengguna dengan topik PA yang pernah mereka pilih sebagai berikut :

- Sorey : Topik 3 dan 5
- Rose : Topik 1, 3, dan 5
- Laila : Topik 1, 2, 3, dan 5
- Mikleo : Topik 1, 3, 4, dan 5
- Zaveid : Topik 2, 3, dan 5

Keterangan :

- 1 = Karunia Rohani
- 2 = Dewasa di Dalam Doa
- 3 = Dewasa di Dalam Karakter
- 4 = Melangkah Melawan Pencobaan
- 5 = Melangkah Dalam Iman

Hasil rekomendasi topik PA ditampilkan pada Gambar 8.

Pilih Topik PA

CARI
TAMPILKAN SEMUA
REKOMENDASI TOPIK

(70 rekomendasi topik PA untuk Anda)

Karunia Rohani

Dewasa Di Dalam Doa

Melangkah Melawan Pencobaan

Mengenal Yesus Sang Allah Sejati

Mengenal Yesus Yang Berkuasa Atas Alam

Mengenal Yesus Sahabat Orang-orang Berdosa

Mengenal Yesus Sang Terang Dunia

Gambar 8. Rekomendasi topik PA bagi Sorey

6. KESIMPULAN

Berdasarkan hasil pengujian, dapat ditarik beberapa kesimpulan sebagai berikut :

1. Fitur pencarian ayat sudah dapat memberikan hasil yang membantu mempelajari dan mendalami Alkitab dengan lebih baik.
2. Fitur pencarian informasi sudah dapat menampilkan bahan-bahan yang diperlukan seperti informasi kitab, biografi tokoh, peta, kamus, artikel, gambar, dan video untuk mendalami Alkitab dengan lebih baik.

3. Fitur Pendalaman Alkitab (PA) sudah berjalan dengan baik dan membantu pengguna untuk mendalami Alkitab dengan bahan PA yang berkualitas.
4. 50% Responden sangat setuju bahwa seluruh fitur dalam aplikasi ini telah berjalan dengan baik, 50% lainnya mengatakan setuju.
5. 70% Responden mengatakan bahwa tampilan aplikasi menarik, 30% sisanya mengatakan tampilannya biasa saja (netral).
6. Seluruh responden tidak mengalami kesulitan untuk memahami tampilan dan penggunaan aplikasi ini. 40% Mengatakan bahwa aplikasi sangat mudah dipahami, 60% lainnya mengatakan mudah dipahami.
7. 40% Responden menilai bahwa aplikasi ini sangat baik, 50% mengatakan baik, dan 10% sisanya mengatakan netral.
8. 70% Responden menilai bahwa aplikasi ini sangat menolong mereka untuk mendalami Alkitab (baik secara pribadi atau kelompok) dengan lebih baik, 30% sisanya menilai aplikasi ini menolong.

7. REFERENSI

- [1] Alkitab Terjemahan Baru. 2016. Lembaga Alkitab Indonesia.
- [2] Gheorghe, R., Hinman, M. L., dan Russo, R. 2016. *Elasticsearch in Action*. Manning Publications Co., Shelter Island, NY.
- [3] Ozgobek, O., Gulla, J. A., dan Erdur, R. C. 2014. A Survey on Challenges and Methods in News Recommendation. In *10th International Conference on Web Information System and Technologies*, 2 (Barcelona, Spain, April 03 – 05, 2014), 278 – 285. DOI = 10.5220/0004844202780285
- [4] Turnbull, D. and Berryman, J. 2016. *Relevant Search*. Manning Publications Co., Shelter Island, NY.
- [5] Wei, S., Ye, N., Zhang, S., Huang, X., dan Zhu, J. 2012. Item Based Collaborative Filtering Recommendation Algorithm Combining Item Category with Interestingness Measure. In *International Conference on Computer Science and Service System*, 1 (Nanjing, China, August 11 – 13, 2012), 2038 – 2041. DOI = 10.1109/CSSS.2012.507