Aplikasi Word Alignment Interlinier Bible dari Bahasa Ibrani/Yunani ke Bahasa Jawa dengan Menggunakan Algoritma Markov Chain Monte Carlo

Yunike Christina, Rolly Intan, Andreas Handojo
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
E-Mail: yunikec@gmail.com, rintan@petra.ac.id, handojo@peter.petra.ac.id

ABSTRAK

Alkitab sebagai buku yang paling populer dan berpengaruh, paling banyak dicetak dan diterjemahkan dalam berbagai bahasa di dunia. Ada sekitar 2.527 terjemahan bagian dari teks Alkitab, dan 475 terjemahan penuh teks Alkitab. Tujuan dari semuanya ini adalah agar banyak orang di dunia dapat membaca Alkitab dengan bahasa yang dapat dimengerti. Dalam proses penerjemahan ini, ada kemungkinan terdapat gap atau kesenjangan antara bahasa asli Alkitab dengan bahasa terjemahan.

Untuk menjawab persoalan tersebut, dirancang suatu aplikasi yang menerapkan algoritma Markov Chain Monte Carlo dan dikemas dalam tool efmaral untuk mengkorelasikan bahasa asli Alkitab dengan bahasa lain yang dikenal dengan Interlinear Bible. Intelinear Bible (ITL) merupakan sebuah sarana untuk melihat makna kata atau frasa pada teks asli Alkitab. Dengan Interlinear Bible, gap atau kesenjangan antara bahasa asli Alkitab dengan bahasa terjemahan dapat diatasi. Interlinear bible merupakan salah satu penerapan word alignment dimana memiliki fungsi untuk mengidentifikasi hubungan antar kata-kata dalam teks paralel dengan dua bahasa yang berbeda.

Berdasarkan hasil penelitian ini, dapat diketahui bahwa penambahan kata pertama secara acak pada tiap awal ayat dapat meningkatkan hasil recall, precision dan f-measure dari algoritma baik korelasi antara Bahasa Jawa dengan Hebrew, maupun korelasi antara Bahsasa Jawa dengan Greek. Selain itu, semakin banyak jumlah kata yang dikorelasikan, semakin meningkat nilai recall, precision dan f-measure.

Kata Kunci: *interlinear bible, word alignment,* MCMC

ABSTRACT

The Bible as the most popular and influential book, most printed and translated in various languages in the world. There are about 2,527 translations of parts of the biblical text, and 475 translations of the full text of the Bible. The purpose of all this is so that many people in the world can read the Bible in understandable languages. In this translation process, there is a possibility that there is a gap between the original language of the Bible and the language of translation.

To answer this problem, an application is designed that applies the Markov Chain Monte Carlo algorithm and is packaged in efmaral tools to correlate the original language of the Bible with other languages known as the Interlinear Bible. The Intelinear Bible (ITL) is a means to see the meaning of words or phrases in the original text of the Bible. With the Interlinear Bible, the gap between the original language of the Bible and the translation language can be overcome. Interlinear bible is one application of word alignment which has a function to identify the relationship between words in parallel text in two different languages.

Based on the results of this study, it can be seen that the addition of the first word randomly at the beginning of each verse can increase the recall results, precision and f-measure of the algorithm both the correlation between Javanese and Hebrew, as well as the correlation between Javanese and Greek. In addition, the more number of words correlated, can increasing recall, precision and f-measure value.

Keywords: *interlinear bible, word alignment,* MCMC

1. PENDAHULUAN

Alkitab sebagai buku yang paling populer dan berpengaruh, paling banyak dicetak dan diterjemahkan dalam berbagai bahasa di dunia. Ada sekitar 2.527 terjemahan bagian dari teks Alkitab, dan 475 terjemahan penuh teks Alkitab [3]. Tujuan dari semuanya ini adalah agar banyak orang di dunia dapat membaca Alkitab dengan bahasa yang dapat dimengerti. Meskipun demikian, masih ada orang-orang suku tertentu, khususnya yang berada di daerah terpencil belum pernah membaca Alkitab. Butuh waktu kurang lebih 40 tahun untuk melakukan proses penerjemahan Alkitab ke bahasa baru [10]. Dalam proses penerjemahan ini, ada kemungkinan terdapat gap atau kesenjangan antara bahasa asli Alkitab dengan bahasa terjemahan [1].

Sistem yang dibuat akan melakukan proses korelasi dari dua teks Alkitab yang berbeda bahasa dengan menggunakan *tool efmaral. Tool* ini menerapkan algoritma *Markov Chain Monte Carlo* dalam menghitung probabilitas korelasi tiap kata. Selain itu, sistem ini juga menyediakan fitur untuk melakukan setemming per kata, sehingga *user* dapat membandingkan hasil korelasi dari berbagai teks.

Setelah melalui beberapa proses korelasi teks Alkitab, sistem dapat menampilkan hasil korelasi dengan tampilan yang *user friendly*. Melalui sistem ini, *pengguna* dapat mengetahui makna kata dari bahasa yang dibaca dalam Alkitab berdasarkan bahasa asli Alkitab.

2. DASAR TEORI

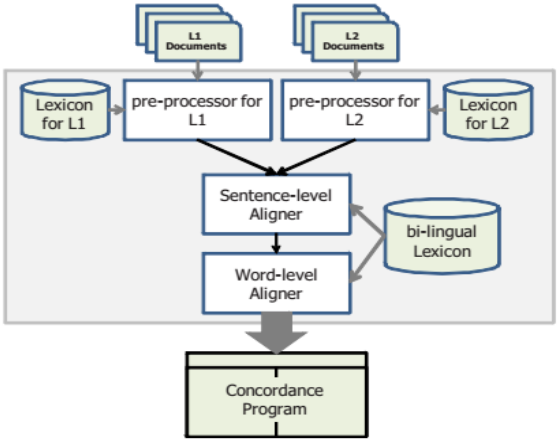
2.1. *Interlinear Bible*

Interlinear Bible adalah suatu cara sederhana untuk memeriksa kata Yunani dan Ibrani yang berada dibalik terjemahan bahasa lain [12]. *Interlinear Bible* dapat menjadi sebuah sarana untuk melihat makna kata atau frasa pada teks asli Alkitab. Dengan *Interlinear Bible*, gap atau kesenjangan antara bahasa asli Alkitab dengan bahasa terjemahan dapat diatasi. Interlinear akan memberikan korelasi yang bermakna antar unsur-unsur kalimat

satu dengan kalimat paralel lain untuk memberi suatu informasi [1]. Dengan interlinear, misionaris tidak membutuhkan waktu yang lama dalam mempelajari makna kata dari bahasa daerah dimana mereka tinggal. Hal serupa juga yang dijelaskan dalam website [11] bahwa manfaat adanya *Interlinear Bible* ini adalah mengurangi waktu dalam menerjemahkan Alkitab.

2.2. *Word Alignment* untuk *parallel corpus*

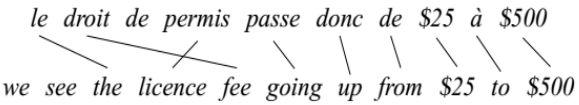
Secara umum, prosedur dalam mempersiapkan *parallel corpus* untuk dua bahasa digambarkan sebagai berikut pada Gambar 1



Gambar 1. Prosedur mempersiapkan paralel corpus

[5]

Pertama, pasangan dokumen paralel ditulis ke dalam bahasa L1 dan L2, L1 dan L2 diproses secara terpisah. *Pre-processor* akan mengelompokkan dokumen ke dalam kalimat-kalimat, dan menglompokkan setiap kata dalam kalimat, menggunakan semua *mono-lingual* lexicon. Penandaan kata kemudian dilakukan, yang diikuti oleh *alignment* pada level kalimat antara dua dokumen. Algoritma *Champollion* digunakan untuk *alignment* kalimat. *Word-level aligner* bertujuan untuk memetakan kata per kata dengan mengacu pada dua bahasa [7]. Salah satu w*ord alignment* model yang banyak digunakan adalah IBM model. IBM model memiliki enam jenis model, namun model pertama dan model kedua yang menjadi landasan utama pengembangan model selanjutnya. Model pertama dapat dianotasikan sebagai $p(f|e)$ yang menunjukkan probabilitas f jika diberikan . Sebagai contoh antara kata dari bahasa *French* yang dianotasikan dengan f dengan kata dari Bahasa *English* yang dianotasikan dengan e, dimana kalimat *French* direpresentasikan dengan sebuah *array* sebanyak I yang menunjukkan jumlah kata dalam kalimat *French*, (f_1, f_2, .., f_I), dan kalimat *English* yang direpresentasikan dengan sebuah *array* sebanyak J yang menunjukkan jumlah kata dalam kalimat *English*, (e_1, e_2, .., e_J). Diasumsikan bahwa setiap kata *French* berkorelasi dengan satu kata *English*. Korelasi antara kata *French* dengan kata *English* dapat direpresentasikan dengan array $\boldsymbol{a}$ dengan panjang I yang digambarkan (a_1, a_2, .., a_I). Nilai variable a_i berada dalam range [0, J] yang menunjukkan index dari korelasi antara kata *English* dengan kata *French* f_i. Jika a_i = 0, ini berarti f_i tidak berkorelasi dengan kata apapun dalam *English*. Gambar 2 menunjukkan contoh korelasi antara kata *French* dengan kata *English*.



Gambar 2. korelasi antara kata *French* dengan kata *English*

[6]

Terdapat satu pasang kalimat (f, e) = (*le droit de permis passe donc de $25 a ` $500, we see the licence fee going up from $25 to $500*). Dimana f menunjuk pada kalimat dalam bahasa *French* dan e menunjukkan kalimat dalam bahasa *English.* Dengan panjang kalimat *French* I = 10 dan kalimat *English* J = 11, f_1 = *le*, f_2 = *droit*, f_3 = *de* dan seterusnya untuk kata *French*, dan e_1 = *we*, e_2 = *see*, e_3 = *the* dan seterusnya untuk kata *English*. Pada contoh di atas, nilai $\boldsymbol{a}$ adalah (3, 5, 0, 4, 6, 7, 8, 9, 10, 11). Dengan notasi ini, dapat dibayangkan bahwa model probabilitas menghasilkan kalimat *French* dari kalimat *English* menggunakan prosedur sederhana. Pertama, panjan I dipilih berdasarkan distribusi $p(I|J)$, dalam hal ini $p(10|11)$. Kemudian, setiap posisi kata *French* dikorelasikan dengan kata *English* (atau null) berdasarkan *uniform distribution* dimana $p(a_i = j \mid J) = \frac{1}{J+1}$, dalam hal ini adalah $\frac{1}{11}$. Pada bagian terakhir, setiap kata *French* f_i diterjemahkan menurut *conditional distribution* pada kata *English* yang terkorelasi. Jadi, untuk *alignment* ini, $p(le|the)$, $p(droit|fee)$, $p(de|null)$, $p(permis|license)$ dan seterusnya dapat dikalikan. Probabilitas gabungan dari kondisi dimana kalimat *French* berkorelasi dengan kalimat *English* dapat dihitung dengan

$$p(f, a|e) = p(I|J) \prod_{i=1}^{I} p(a_i|J).p(f_i|e_{a_i}) \qquad (1)$$

[6]. Salah satu contoh isi corpus dengan Bahasa Jawa sebagai bahasa target dan Bahasa Yunani sebagai bahasa sumber sekaligus bahasa asli Alkitab

Bahasa Jawa:

Iki sarasilahe Gusti Yesus Kritus tedhake Sang Prabu Dawud tedhake Rama Abraham

Bahasa Yunani ditampilkan dengan nomor Strong :

976 1078 2424 5547 5207 1138 5207 11

Biblos genesis Iesous Christos, huios Dabid, huios Abraam.

2.3. Algoritma *Markov Chain Monte Carlo*

Markov Chain Monte Carlo atau MCMC adalah algoritma yang diterapkan dalam *tool efmaral*, digunakan untuk melakukan korelasi dua parallel teks. MCMC adalah sebuah metode sampling *computer-driven* yang memungkinkan untuk mengkarakterisasi sebuah distribusi tanpa memahami semua sifat distribusi matematikanya dengan mengambil value dari distribusi secara acak. MCMC memberikan jawaban untuk masalah simulasi yang sulit dari distribusi dengan dimensi yang tinggi, muncul dalam model yang kompleks [8].

Nama MCMC mengkombinasikan dua properti, yaitu *monte carlo* dan *markov chain. Monte carlo* berarti memperkirakan properti dari distribusi dengan menguji sample acak dari distribusi. Sebagai contoh, daripada menghitung rata-rata dari distribusi normal dengan persamaan distribusi, pendekatan *monte carlo* akan menarik sejumlah besar sampel acak dari distribusi normal dan menghitung rata-rata sampel [9].

Berbeda dengan eksperimen fisik, simulasi *monte carlo* melakukan *random sampling* dan eksperimen pada komputer

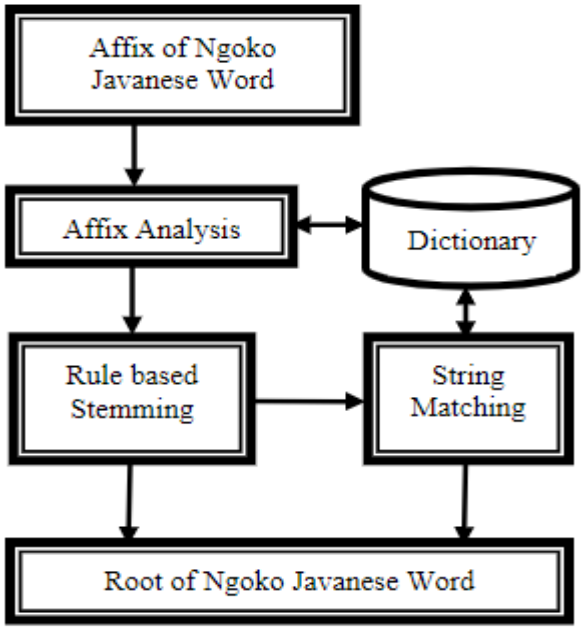
dalam jumlah besar. Kemudian, karakteristik statistikal dari eksperimen (*output model*) diobservasi, dan kesimpulan dari *output model* ditarik berdasarkan eksperimen statistikal. Pada setiap eksperimen, nilai yang mungkin dari variabel acak X yang dimasukkan disample berdasarkan distribusinya. Kemudian nilai dari variabel *output* Y dihitung melalui fungsi $Y = g(X)$ pada sample dari variabel acak yang dimasukkan. Dengan sejumlah percobaan yang dilakukan, sekumpulan sample dari variabel *output* Y tersedia untuk analisis statistikal yang memperkirakan karakteristik dari variabel *output* Y [4].

Sedangkan *markov chain* adalah sebuah gagasan bahwa sampel acak dihasilkan oleh urutan proses khusus. Setiap sampel acak digunakan sebagai batu loncatan untuk menghasilkan sampel acak selanjutnya. Hal khusus dari *chain* adalah setiap sampel baru, hanya tergantung pada sampel sebelumnya. Sample acak pertama atau urutan sample dari sample sebelumnya tidak akan mempengaruhi. [8].
Dapat diketahui bahwa distribusi *posterior* (perubahan nilai probabilitas setelah dilakukan eksperimen) berada pada kisaran distribusi *prior* (nilai probabilitas yang diyakini benar sebelum melakukan eksperimen terhadap sesuatu) dan distribusi probabilitas, namun karena suatu hal (misalnya karena dimensi probabilitas yang tinggi), tidak dapat dihitung secara langsung. Menggunakan metode MCMC akan secara efektif menarik sampel dari distribusi *posterior*, dan menghitung statistik [9].

Untuk memulainya, metode MCMC akan mengambil sebuah nilai parameter acak untuk diperhatikan. Simulasi akan terus berlanjut untuk menghasilkan nilai acak (bagian monte carlo). Kuncinya adalah, untuk sepasang nilai parameter, akan dilakukan perhitungan mana yang merupakan nilai parameter yang lebih baik, dengan menghitung seberapa besar masing-masing nilai menjelaskan data. Jika nilai parameter yang dihasilkan secara acak lebih baik daripada yang terakhir, akan ditambahkan ke rantai nilai parameter dengan probabilitas tertentu yang ditentukan oleh seberapa jauh lebih baik itu (ini adalah bagian rantai Markov) [9].

2.4. *Stemming Bahasa Jawa Formal*

Stemming adalah cabang dari morfologi. Morfologi berarti pembelajaran tentang struktur kata. Stemming merupakan inti dari *natural language processing* untuk pencarian informasi yang efektif dan efisien. Stemming digunakan untuk mengubah variasi kata ke akar kata dengan menerapkan aturan morfologis. Hal yang dilakukan dalam stemming adalah menghilangkan semua imbuhan (*affixes*) baik yang terdiri dari awalan (*prefixes*), sisipan (*infixes*), akhiran (*suffixes*) dan *confixes* (kombinasi dari awalan dan akhiran) pada kata turunan [9]. Stemming Jawa Formal memiliki masalah yang kompleks karena banyaknya imbuhan pada kata. Salah satunya adalah jenis afiks yang berbeda. Awalan dapat berubah tergantung pada huruf pertama dari kata dasar. Misalnya, awalan 'ng' berubah menjadi 'k' saat huruf pertama dari kata dasar adalah 'g', misal 'Ngetok' (root: ketok). Aturan lain di mana awalan dapat diubah menjadi 'ng' adalah ketika huruf pertama dari akarnya adalah 'k', misalnya 'ngethok' (root: kethok). 'Ketok' dan 'kethok' memiliki arti yang berbeda. Dalam bahasa Inggris, 'ketok' berarti terlihat dan 'kethok' berarti terpotong. Jika ada lebih dari satu imbuhan yang dilekatkan pada suatu kata maka urutan langkah untuk menghapus afiks menjadi sangat penting. Stemmer diharapkan dapat mengurangi dimensi data; sehingga akan meningkatkan kinerja proses kategorisasi. Gambar 3 merupakan urutan dalam melakukan stemming pada kata Bahasa Jawa Formal.

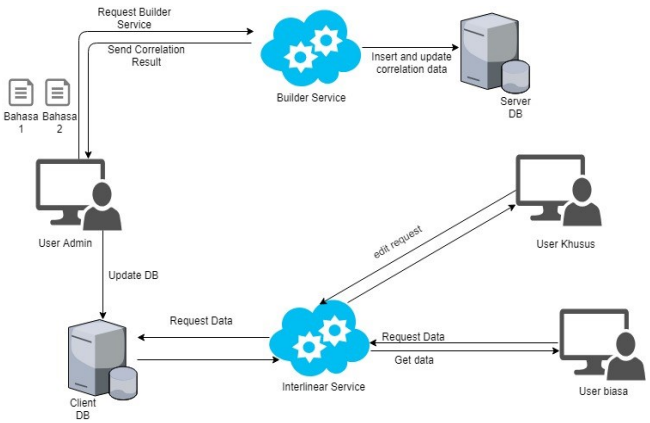


Gambar 3. Diagram Proses Stemming

[2]
Affix dari kata yang diinputkan akan diperiksa dalam kamus. Apabila kata terdapat dalam kamus, maka kata tersebut sudah menjadi akar kata. Apabila kata tidak ada dalam kamus, maka dilakukan proses pemotongan. Jika akar kata tetap tidak ditemukan setelah dilakukan pemotongan, langkah selanjutnya adalah *string matching*. *String matching* dilakukan untuk menemukan kesamaan paling tinggi dengan kata dalam kamus. Jika semua langkah sudah dilakukan dan tetap tidak ditemukan kata yang cocok dengan kamus, maka kata yang memiliki kesamaan paling tinggi dianggap akar katanya [2].

3. DESAIN SISTEM

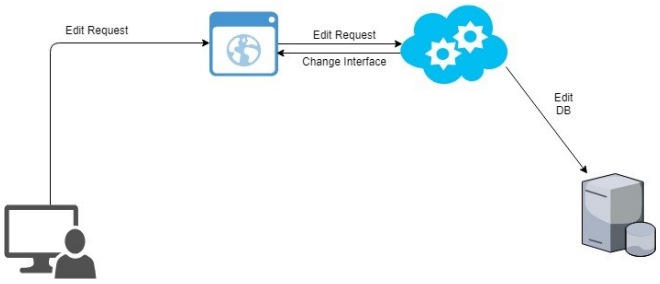
3.1. Arsitektur Sistem *Builder* dan *interlinear Bible*



Gambar 4. Arsitektur Sistem *Builder* dan *interlinear Bible*

Dari Gambar 4 dijelaskan mengenai rancangan sistem arsitektur yang akan di implementasikan pada aplikasi *Interlinear Bible* dan

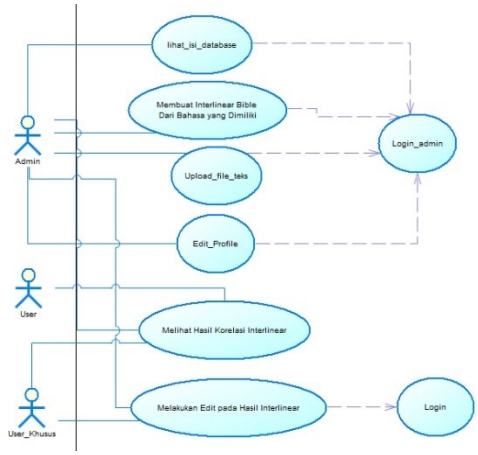
Builder. Sistem arsitektur ini memiliki komponen berupa *builder service*, client yang merupakan *user* admin dan *user* biasa, *database* untuk menyimpan data korelasi, dan *interlinear service*.



Gambar 5. Arsitektur fitur edit *Interlinear Bible*

Gambar 5 menjelaskan rancangan sistem arsitektur yang akan di implementasikan pada fitur edit *Interlinear Bible*. Sistem arsitektur ini memiliki komponen berupa webserver, *database* untuk menyimpan data korelasi, *client* atau *user* yang memiliki hak akses khusus, dan *interface*.

3.2. *Use Case* Diagram

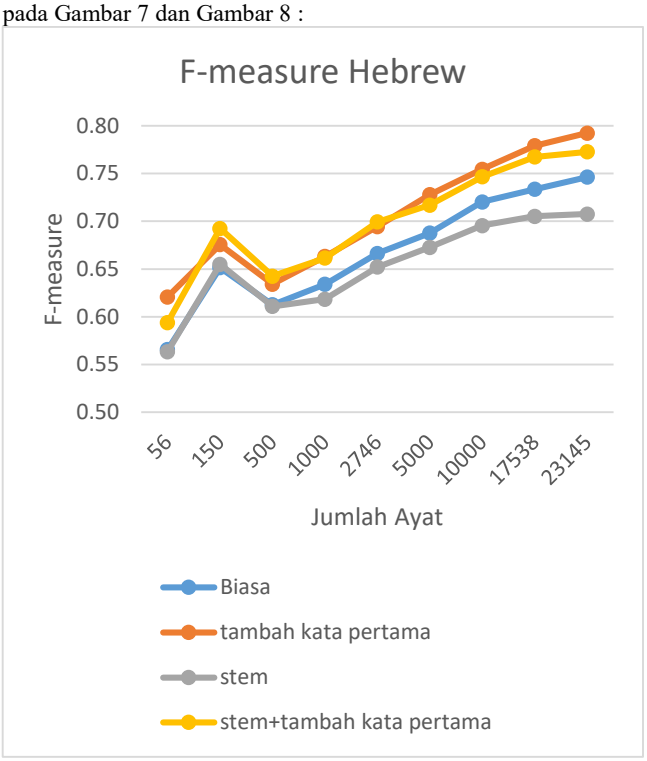


Gambar 6. *Use Case Diagram builder* dan aplikasi *interlinear Bible*

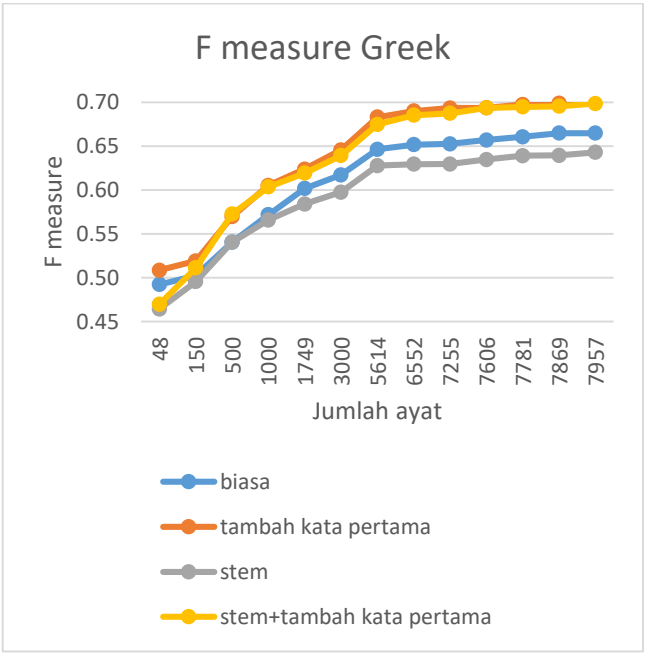
Usecase diagram pada Gambar 6 mendeskripsikan aktor-aktor yang ada di dalam sistem yaitu *User* Admin, *User* biasa, dan *User* khusus. Setiap aktor yang ada memiliki fungsi yang berbeda-beda sesuai dengan kebutuhan. Pertama, *usecase* admin terdiri dari melakukan *upload* file teks ke server, membuat aplikasi *Interlinear Bible*, melihat hasil korelasi atau *Interlinear Bible*, melakukan edit pada hasil interlinear, dan melihat isi database yang dihasilkan. Kedua, usecase *user* biasa hanya terdiri dari melihat hasil korelasi atau *Interlinear Bible*. Ketiga, *user* khusus terdiri dari melihat hasil korelasi atau *Interlinear Bible* dan melakukan edit pada hasil korelasi. Pada usecase diatas, setiap aktor memiliki fungsinya sendiri-sendiri dan tidak berinteraksi antara satu aktor dengan aktor lainnya.

4. PENGUJIAN SISTEM

Selama waktu pengujian, ada tiga macam data test *alignment* yang dipakai untuk masing-masing kitab sebagai hasil dari *tool efmaral*. Data pertama berisi kata-kata yang belum di*stemming*, disebut data biasa, data kedua berisi kata-kata yang sudah di*stemming*, dan data ketiga berisi kata awal acak pada setiap bagian awal ayat. Sedangkan data *alignment gold standard* diperoleh dari Lembaga Sabda. Pengujian dilakukan dengan perbandingan jumlah ayat secara bertahap. Berdasarkan hasil pengujian terhadap masing-masing kitab, diperoleh hasil seperti pada Gambar 7 dan Gambar 8 :



Gambar 7. Perhitungan *F-measure* secara bertahap pada kitab perjanjian lama



Gambar 8. Perhitungan *F-measure* secara bertahap pada kitab perjanjian lama

5. KESIMPULAN

Dari hasil pengujian sistem yang telah dilakukan, dapat diambil beberapa kesimpulan antara lain :

- Semakin banyak kata yang dikorelasikan, maka nilai recall, precision dan F-measure akan semakin meningkat.
- Semakin banyak kata yang dikorelasikan, semakin banyak waktu yang digunakan tool efmaral melakukan perhitungan korelasi.
- Penambahan kata acak pada setiap awal ayat dapat meningkatkan nilai F-measure dari korelasi oleh tool efmaral.
- Berdasarkan tabel pengujian, perbandingan korelasi antara teks yang berisi kata-kata yang sudah distemming dengan teks yang berisi kata-kata yang tidak di*stemming* tidak terlalu signifikan.

6. DAFTAR PUSTAKA

[1] Aji, A. 2016. Retrieved from www.slideshare.net: https://www.slideshare.net/sabda/bible-interlinear

[2] Amin, F., Hadikurniawati, W., Wibisono, S., Februariyanti, H., & Wibowo, J. 2017. A HYBRID METHOD OF RULE-BASED AND STRING MATCHING STEMMER FOR JAVANESE LANGUAGE. Semarang: Faculty of Information Technology.

[3] Christodouloupoulos, C., & Steedman, M. 2014, November 19. *PMC4551210*. Retrieved from www.ncbi.nlm.nih.gov: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4551210/

[4] Du, X. n.d. *me360.* Retrieved from http://web.mst.edu: http://web.mst.edu/~dux/repository/me360/me360_lecture8.html

[5] Kim, J. E., & Lee, K. J. 2013. An automatic maximum word alignment of parallel corpus for ESL learners. *Procedia - Social and Behavioral Sciences 106 (2013) 380 – 386*, 382.

[6] Lopez, A. n.d. Word Alignment and the Expectation-Maximization Algorithm. citeseerx, 1-2.

[7] Madia, M. 2016. *STEMMING BAHASA JAWA UNTUK MENCARI AKAR KATA DALAM BAHASA JAWA DENGAN ATURAN ANALISIS KONTRASIF AFIKSASI VERBA.* Malang.

[8] Ravenzwaaij, D. V., Cassey, P., & Brown, S. D. 2016. A simple introduction to Markov Chain Monte–Carlo. 143.

[9] Shaver, B. 2017, December 22. *a-zero-math-introduction-to-markov-chain-monte-carlo-methods-dcba889e0c50*. Retrieved from towardsdatascience.com: https://towardsdatascience.com/a-zero-math-introduction-to-markov-chain-monte-carlo-methods-dcba889e0c50

[10] about. n.d. Retrieved from wycliffe: https://www.wycliffe.org.uk/about/faq/

[11] Retrieved from http://hebrewbiblescholar.com: http://hebrewbiblescholar.com/avoid-interlinears/

[12] What's an Interlinear Bible. n.d. Retrieved from biblestudytools: https://www.biblestudytools.com/blogs/inside-bst/what-s-an-interlinear-bible.html