IMPLEMENTASI *VECTOR SPACE MODEL* PADA SISTEM PENCARIAN MESIN KARAOKE

[1)] Anna, [2)] Ade Hendini

[1)] Komputerisasi Akuntansi, AMIK BSI Pontianak
anna.nnz@bsi.ac.id
[2)] Manajemen Informatika, AMIK BSI Pontianak
ade.aee@bsi.ac.id

Abstract *- The growing business of karaoke now demands businessmen to compete to provide karaoke service with convenient facilities for visitors. One of them is a song search based on title, singer, or country category on karaoke machine. This can be solved by information retrieval system (Information Retrieval System). One of the models in the information retrieval system is the Vector Space Model. The dataset used in this study is all genres of songs in pre-processing which includes tokenizing, filtering, the formation of inverted index and stemming. While the search process using modeling based Vector Space Model. The result is a design of a search system on a karaoke machine that can give weighting of each song that matches the keywords the song is looking for. With this system can also be seen the value of the weight of each song is relevant to provide recommendations or the title of the most popular song / often sought so as to suit the needs.*

Keywords: Vector Space Model, Search System, Relevant, Retrieval System.

Abstrak - Semakin berkembangnya bisnis karaoke sekarang ini menuntut para pebisnis untuk berlomba-lomba menyediakan layanan tempat karaoke dengan fasilitas yang nyaman bagi pengunjung. Salah satunya adalah pencarian lagu berdasarkan judul, penyanyi, ataupun kategori negara pada mesin karaoke. Hal ini bisa dipecahkan dengan sistem temu kembali informasi (*Information Retrieval System*). Salah satu model dalam information retrieval system yakni *Vector Space Model*. Dataset yang digunakan pada penelitian ini adalah semua genre lagu di pre-processing yang meliputi tokenizing, filtering, pembentukan inverted index dan stemming. Sedangkan proses pencariannya menggunakan pemodelan berbasis *Vector Space Model*. Hasilnya adalah sebuah rancangan sebuah sistem pencarian pada mesin karaoke yang dapat memberikan pembobotan dari masing-masing lagu yang sesuai dengan kata kunci lagu yang dicari. Dengan sistem ini pula dapat dilihat nilai bobot dari masing-masing lagu yang relevan dengan memberikan rekomendasi atau judul lagu yang paling populer/sering dicari sehingga sesuai dengan kebutuhan.

Kata Kunci: Vector Space Model, Sistem Pencarian, Relevan, Sistem Temu Kembali.

A. PENDAHULUAN

Pesatnya dunia hiburan yang ditawarkan perlahan-lahan akan mengurangi titik jenuh yang dialami oleh konsumen. Seperti yang diketahui tanggung jawab dan kewajiban dalam keseharian yang kerap kali tidak berimbang mampu membuat tiap individu mengalami titik jenuh. Menurut pakar, manusia harus mengimbangi kinerja otak kanan dan otak kiri mereka untuk menjaga kestabilan kesehatan tubuh, yang mana otak kiri lebih dominan digunakan dalam kegiatan berpikir logika, sedangkan otak kanan digunakan dalam kegiatan seni atau hiburan. Berkaraoke merupakan salah satu hiburan yang digunakan kebanyakan orang untuk menghabiskan waktu liburan mereka.

Karaoke adalah sebuah bentuk hiburan di mana seseorang menyanyi diiringi dengan musik dan teks lirik yang ditunjukkan pada sebuah layar televisi. Di Indonesia termasuk salah satu negara yang banyak mendirikan usaha karaoke hampir diseluruh pelosok tanah air, sehingga sangat mudah sekali bagi para pelanggan untuk mencari tempat-tempat karoke untuk mereka bernyanyi. Perkembangan bisnis karaoke keluarga makin hari makin meningkat dan semuanya mempunyai tujuan yang sama, yaitu untuk memberikan hiburan bagi anggota keluarga atau sejenak melepas penat dari rutinitas keseharian. Semakin berkembangnya tempat penyedia layanan karaoke, semakin banyak pula para pebisnis berlomba-lomba untuk menciptakan fasilitas yang nyaman bagi pengunjung, salah satunya adalah sistem pencarian lagu yang ingin dinyanyikan pada mesin karaoke. Kualitas sistem pencarian lagu baik berdasarkan judul, penyanyi, ataupun kategori negara sangat mempengaruhi kenyamanan pengunjung. Terkadang pengunjung kesulitan dalam mencari dengan tepat dan cepat lagu yang diinginkan. Hal ini dirasakan merugikan bagi mereka yang terkendala menemukan lagu dalam waktu yang lama. Beberapa diantaranya

ada lagu yang tidak dapat ditemukan jika dicari berdasarkan judul, tetapi berhasil ditemukan ketika dicari berdasarkan penyanyi dengan judul lagu yang sama diawal. Selain itu, ada beberapa sistem pencarian belum terkelompokkan dengan baik baik dalam kategori penyanyi solo, featuring, band, maupun kategori penyanyi pria dan wanita, ini menunjukkan rendahnya kualitas pencarian lagu pada mesin karaoke tersebut. Hal tersebut sedikit banyaknya berdampak terhadap kualitas pelayanan tempat karaoke itu sendiri.

Tujuan penelitian ini adalah untuk merancang sebuah sistem pencarian pada mesin karaoke yang dapat memberikan pembobotan dari masing-masing lagu yang sesuai dengan kata kunci lagu yang dicari yaitu berdasarkan judul, asal negara ataupun penyanyi. Sehingga akan dapat memilih yang mana lagu-lagu populer yang paling sering dicari.

Penerapan *Vector Space Model* (VSM) merupakan salah satu alternatif untuk memecahkan masalah ini. Dengan *Vector Space Metode* dapat dilihat tingkat kedekatan atau kesamaan dengan cara pembobotan term. Pada proses stemming atau mencari kata dasar pada kata, sistem akan menggunakan algoritma tala. Dengan adanya penelitian ini para pebisnis dan pengunjung akan sangat terbantu dalam mempermudah pencarian lagu dengan cepat dan paling populer sesuai yang diinginkan.

B. TINJAUAN PUSTAKA

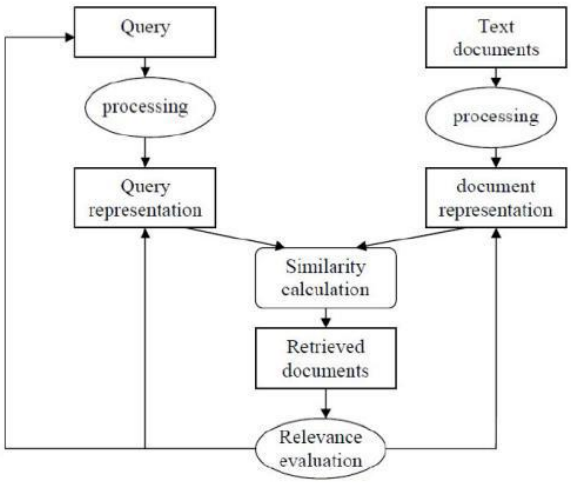
1. Sistem Temu Kembali (Information Retrieval System)

Temu Kembali Informasi (*Information Retrieval*) adalah menemukan bahan (biasanya dokumen) dari sesuatu yang tidak terstruktur (biasanya teks) dalam jumlah yang besar untuk menghasilkan informasi yang dibutuhkan (biasanya tersimpan dalam komputer. Prinsip kerja sistem temu kembali informasi jika ada sebuah kumpulan dokumen dan seorang *user* yang memformulasikan sebuah pertanyaan request atau query. Sistem Temu Kembali Informasi bertujuan untuk menjawab kebutuhan informasi *user* dengan sumber informasi yang tersedia dalam waktu yang singkat dan kondisi seperti sebagai berikut (Fatkhul Amin, 2012):

a) Mempresentasikan sekumpulan ide dalam sebuah dokumen menggunakan sekumpulan konsep.
b) Terdapat beberapa pengguna yang memerlukan ide, tapi tidak dapat mengidentifikasikan dan menemukannya dengan baik.
c) Sistem temu kembali informasi bertujuan untuk mempertemukan ide yang dikemukakan oleh penulis dalam dokumen dengan kebutuhan informasi pengguna yang dinyatakan dalam bentuk key word query/istilah penelusuran.

2. Arsitektur Sistem Temu Kembali Informasi

Sistem memiliki dua pekerjaan yaitu, yaitu melakukan pre-processing terhadap database dan kemudian menerapkan metode tertentu untuk menghitung kedekatan relevansi atau similarity antara dokumen di dalam database yang telah dipreprocess dengan query pengguna. Query yang dimasukkan pengguna dikonversi sesuai aturan tertentu untuk mengekstrak term-term penting yang sejalan dengan term-term yang sebelumnya telah diekstrak dari dokumen dan menghitung relevansi antara query dan dokumen berdasarkan pada term-term tersebut. Sebagai hasilnya, sistem mengembalikan suatu daftar dokumen terurut sesuai nilai kemiripannya dengan query pengguna.



Gambar 1. Proses Temu Kembali Sistem

3. Text Mining

Text mining adalah proses mengolah data yang berupa teks yang didapatkan dari dokumen untuk mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa hubungan antar dokumen. Proses penganalisisan teks guna menyarikan informasi yang bermanfaat untuk tujuan tertentu.

Text mining dapat didefinisikan sebagai suatu proses menggali informasi dimana seorang *user* berinteraksi dengan sekumpulan dokumen menggunakan *tools* analisis yang merupakan komponen-komponen dalam data mining yang salah satunya adalah kategorisasi.

4. Tokenizing

Tokenisasi secara garis besar dapat diartikan sebagai peoses pememecah sekumpulan karakter dalam suatu teks ke dalam satuan kata. Proses untuk membagi teks yang dapat berupa kalimat, paragraf atau dokumen, menjadi token-token atau bagian-bagian tertentu. Tokenisasi merupakan proses pemisahan suatu rangkaian

karakter berdasarkan karakter spasi, dan mungkin pada waktu yang bersamaan dilakukan juga proses penghapusan karakter tertentu, seperti tanda baca. Token seringkali disebut sebagai istilah term atau kata, sebagai contoh sebuah token merupakan suatu urutan karakter dari dokumen tertentu yang dikelompokkan sebagai unit semantik yang berguna untuk diproses.

5. Filtering

Filtering adalah proses membuang kata-kata yang dianggap sebagai nois atau kata yang dianggap tidak penting dan tidak berpengaruh terhadap makna kata. Tahap filtering adalah tahap pengambilan kata-kata yang penting dari hasil tokenizing. Tahap filtering ini menggunakan daftar stoplist atau wordlist. Stoplist yaitu penyaringan filtering terhadap kata-kata yang tidak layak untuk dijadikan sebagai pembeda atau sebagai kata kunci dalam pencarian dokumen sehingga kata-kata tersebut dapat dihilangkan dari dokumen. Sedangkan wordlist adalah daftar kata yang mungkin digunakan sebagai kata kunci dalam pencarian dokumen, dengan demikian maka tentu jumlah kata yang termasuk dalam wordlist akan lebih banyak daripada stoplist.

6. Stemming

Stemming dalam sistem temu kembali informasi digunakan untuk membatasi varian bentuk kata yang berbeda menjadi bentuk dasarnya, sehingga nantinya dapat meningkatkan kemampuan sistem dalam menemukan dokumen relevan sesuai query yang ada.

Teknik stemming dikembangkan untuk alasan mereduksi term menjadi bentuk dasarnya. Term yang ada pada dokumen dan query memiliki banyak varian morfologik maka akan sulit term-term tersebut dianggap ekivalen. Namun dalam beberapa kasus tertentu varian morfologik term-term memiliki interpretasi semantik yang sama dan dapat dikategorikan ekivalen. Algoritma Stemming untuk bahasa yang satu berbeda dengan algoritma stemming untuk bahasa lainnya. Proses stemming pada teks berbahasa Indonesia lebih rumit/kompleks karena terdapat variasi imbuhan yang harus dibuang untuk mendapatkan root word dari sebuah kata.

7. Indexing

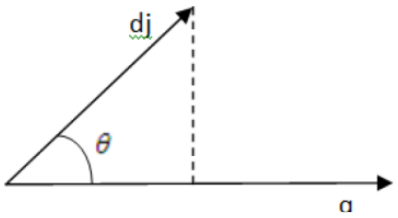
Proses Indexing adalah tahap pengindeksan kata dari koleksi teks yang digunakan untuk mempercepat proses pencarian. Seluruh dokumen dalam koleksi disimpan dalam satu file dengan format tertentu sehingga antara dokumen satu dengan dokumen yang lain bisa dibedakan. Setelah kata telah dikembalikan dalam bentuk kata dasar, kemudian disimpan dalam tabel. Proses indexing menghasilkan database index.

C. METODE PENELITIAN

Vector Space Model (VSM) adalah salah satu metode atau algoritma yang sering digunakan untuk sebuah sistem temu kembali informasi. Algoritma ini merupakan sebuah model yang digunakan untuk mengukur kemiripan atau kesamaan (similarity term) antar suatu dokumen dengan suatu query dengan cara pembobotan term. Dalam Information Retrieval System, kemiripan antar dokumen didefinisikan berdasarkan representasi bag-of-words dan dikonversi ke suatu model ruang vector. Relevansi sebuah dokumen ke sebuah query didasarkan pada similaritas diantara vektor dokumen dan vektor query.

Konsep dasar dari *Vector Space Model* adalah menghitung jarak antar dokumen kemudian mengurutkan berdasarkan tingkat kedekatannya. Cara kerja *Vector Space Model* dimulai dengan case folding , cleaning data, indexing, filtering, stemming, dan tokenisasi yaitu tahap pemotongan string input berdasarkan tiap kata yang meyusunnya dan memecah dokumen ke dalam tabel frekuensi kata. Seluruh kata dalam dokumen dibentuk menjadi satu yang disebut sebagai term. Tiap dokumen ditampilkan sebagai vektor yang akan dibandingkan dengan term yang telah dibentuk. Similarity Analysis untuk mengukur kemiripan dokumen dilakukan dengan menghitung cosinus jarak antara dokumen tersebut.

Sebuah dokumen dj dan sebuah query q direpresentasikan sebagai vektor dimensi seperti pada gambar 2.7.



Gambar 2. Representasi Dokumen dan Query pada Ruang Vektor

Proses perhitungan *Vector Space Model* melalui tahapan perhitungan *term frequency (tf)* menggunakan persamaan (1)

$$tf = tfij$$

Dengan *tf* adalah *term frequency*, dan *tfij* adalah banyaknya kemunculan term *ti* dalam dokumen *dj, Term frequency (tf)* dihitung dengan menghitung banyaknya kemunculan term *ti* dalam dokumen *dj.*

Perhitungan *Inverse Document Frequency (idf),* menggunakan persamaan (2)

$$idfi = logN/df_i$$

Dengan *idfi* adalah *inverse document frequency*, *N* adalah jumlah dokumen yang terambil oleh sistem, dan *dfi* adalah banyaknya dokumen dalam koleksi dimana term *ti* muncul di dalamnya, maka Perhitungan *idfi* digunakan untuk mengetahui banyaknya term yang dicari *(dfi)* yang muncul dalam dokumen lain yang ada pada *database*.

Perhitungan *term frequency Inverse Document Frequency* (*tfidf*), menggunakan persamaan (3)

$$Wij = tfij \ .log \ N/df_i$$

Dengan *Wij* adalah bobot dokumen, *N* adalah Jumlah dokumen yang terambil oleh sistem, *tfij* adalah banyaknya kemunculan term *ti* pada dokumen *dj*, dan *dfi* adalah banyaknya dokumen dalam koleksi dimana term *ti* muncul di dalamnya. Bobot dokumen *(Wij)* dihitung untuk didapatkannya suatu bobot hasil perkalian atau kombinasi antara *term frequency (tfij)* dan *Inverse Document Frequency (idf).*

Perhitungan Jarak *query,* menggunakan persamaan (4)

$$|q| = \sqrt{\sum_{i=1}^{t} (W_{i,q})^2}$$

Dengan |*q*| adalah Jarak query, dan *Wiq* adalah bobot *query* dokumen ke-*i*, maka Jarak *query (|q|)* dihitung untuk didapatkan jarak *query* dari bobot *query* dokumen *(Wiq)* yang terambil oleh sistem. Jarak *query* bisa dihitung dengan persamaan akar jumlah kuadrat dari *query*.

Perhitungan Jarak Dokumen, menggunakan persamaan (5)

$$|d_j| = \sqrt{\sum_{i=1}^{t} (W_{i,j})^2}$$

Dengan |*dj*| adalah jarak dokumen, dan *Wij* adalah bobot dokumen ke-*i*, maka Jarak dokumen *(|dj |)* dihitung untuk didapatkan jarak dokumen dari bobot dokumen dokumen *(Wij)* yang terambil oleh sistem. Jarak dokumen bisa dihitung dengan persamaan akar jumlah kuadrat dari dokumen. Menghitung *index terms* dari dokumen dan query *(q,dj).* menggunakan persamaan (6)

$$q,d_j = \sum_{i=1}^{t} W_{i,q} \ . W_{i,j}$$

Dengan *Wij* adalah bobot *term* dalam dokumen, *Wiq* adalah bobot *query*. Pengukuran *Cosine Similarity* menghitung nilai *kosinus* sudut antara dua *vector* menggunakan persamaan (7)

$$Sim(q, d_j) = \frac{q \cdot d_j}{|q| * |d_j|}$$

Similaritas antara *query* dan dokumen atau *Sim(q,dj)* berbanding lurus terhadap jumlah bobot *query (q)* dikali bobot dokumen *(dj)* dan berbanding terbalik terhadap akar jumlah kuadrat *q (|q|)* dikali akar jumlah kuadrat dokumen *(|dj|).* Perhitungan similaritas menghasilkan bobot dokumen yang mendekati nilai 1 atau menghasilkan bobot dokumen yang lebih besar dibandingkan dengan nilai yang dihasilkan dari perhitungan *inner product*.

D. HASIL DAN PEMBAHASAN

1. *Vector Space Model*

Untuk kepentingan analisis data di dalam melakukan penelitian ini dibutuhkan beberapa sampel data yang diambil dari enam buah judul lagu yaitu:

D1 : Kesempurnaan Cinta
D2 : Seindah Biasa
D3 : Cinta dan Benci
D4 : Bukan Cinta Biasa
D5 : Cinta Sejati
D6 : Sahabat Sejati

Jadi total dokumen ada 6. Apabila dilakukan pencarian dokumen dengan kata kunci:

Q : Cinta
dokumen manakah yang paling relevan ?

Untuk menjawab pertanyaan di atas perlu dilakukan suatu proses perhitungan pembobotan atau perankingan dari masing-masing judul lagu tersebut sehingga dari hasil pembobotan atau perangkingan tersebut akan terlihat judul lagu yang mana yang relevan atau yang paling diprioritaskan. Di dalam penerapan metode *Vector Space Model* ada beberapa tahapan proses pengolahan data terlebih dahulu.

Dari tahapan pembobotan atau perangkingan dengan *Vector Space Model* agar mempermudah di dalam proses perhitungan dengan tahapan-tahapan persamaan yang telah dijelaskan sebelumnya, dibuat sebuah tabel ilustrasi perhitungan *Vector Space Model* seperti pada tabel di bawah ini.

Tabel 1. Ilustrasi Perhitungan *Vector Space Model*

	Jumlah $tf = tf_{ij}$							
Terms	**Q**	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	***dfi***
Sempurna	0	1	0	0	0	0	0	1
Cinta	1	1	0	1	1	1	0	4
Indah	0	0	1	0	0	0	0	1

Biasa	0	0	1	0	1	0	0	2
Benci	0	0	0	1	0	0	0	1
Bukan	0	0	0	0	1	0	0	1
Sejati	0	0	0	0	0	1	1	2
Sahabat	0	0	0	0	0	0	1	1

Tabel 2. Ilustrasi Perhitungan *Vector Space Model*(Lanjutan)

		idfi = logN/df$_i$
Terms	***N/df$_i$***	***idfi***
Sempurna	6/1 = 6	0,7782
Cinta	6/4 = 1,5	0,1761
Indah	6/1 = 6	0,7782
Biasa	6/2 = 3	0,4771
Benci	6/1 = 6	0,7782
Bukan	6/1 = 6	0,7782
Sejati	6/2 = 3	0,4771
Sahabat	6/1 = 6	0,7782

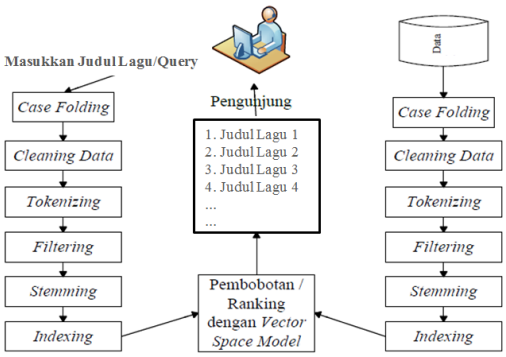
Tabel 3. Ilustrasi Perhitungan *Vector Space Model*(Lanjutan)

	Bobot ***Wij = tfij .logN/dfi***						
Terms	**Q**	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**
Sempurna	0	0,7782	0	0	0	0	0
Cinta	0,1761	0,1761	0	0,1761	0,1761	0,1761	0
Indah	0	0	0,7782	0	0	0	0
Biasa	0	0	0,4771	0	0,4771	0	0
Benci	0	0	0	0,7782	0	0	0
Bukan	0	0	0	0	0,7782	0	0
Sejati	0	0	0	0	0	0,4771	0,4771
Sahabat	0	0	0	0	0	0	0,7782

Keterangan: ***Terms*** adalah hasil indexing, ***Q*** adalah jumlah kemunculan *trems* pada *Query, (**D1, D2, D3, D4, D5, dan D6**)* jumlah kemunculan *terms,* ***dfi*** adalah banyaknya dokumen dalam koleksi di mana term *ti* muncul di dalamnya, ***N/dfi*** total dokumen dibagi banyaknya dokumen dalam koleksi di mana term *ti* muncul di dalamnya, ***idfi*** adalah nilai dari ***log*** dan ***(Q, D1,D2,D3,D4,D5 dan D6)*** kolom terakhir adalah bobot dokumen dan bobot *query*. Setelah itu dihitung jarak dokumen dengan menggunakan persamaan(5). Proses terakhir yaitu menghitung similaritas dokumen dan meranking menggunakan persamaan(7). Dari proses similarities di atas dapat diambil ranking dari setiap lagu. Jadi, dokumen yang paling relevan dengan kata kunci adalah yang paling tinggi tingkat kemiripannya.

2. Skema Pencarian Data Lagu

Skema rancangan pencarian data lagu menjelaskan bagaimana alur pemrosesan sistem pencarian data lagu yang nantinya akan dibangun. Dari gambar tersebut dapat dilihat proses diawali dari memasukkan judul lagu atau *query* pada sistem oleh Pengunjung, kemudian query tersebut dilakukan beberapa proses yaitu *case folding, cleaning data, tokenizing, filtering, stemming* dan *indexing* sehingga akan dapat menghasilakan bobot atau ranking dari judul lagu yang relevan. Pengunjung akan mendapatkan informasi berupa sejumlah lagu yang relevan dan dapat dilihat berapa bobot untuk masing masing lagu yang direkomendasikan oleh sistem.



Gambar 2. Skema Pencarian Data Lagu

3. Hasil

Dari hasil perhitungan pembobotan di atas maka sistem memberikan jawaban atau hasil perhitungan yang menunjukkan judul lagu manakah yang paling relevan dan yang mana yang paling diprioritaskan.

E. KESIMPULAN

Dari pembahasan dan penjelasan di atas, dapat ditarik kesimpuan bahwa penerapan *Vector Space Model* pada sistem pencarian lagu berdasarkan judul dapat mempercepat proses pencarian lagu yang relevan sesuai dengan lagu yang ingin dicari pengunjung. Dengan sistem ini pula dapat dilihat nilai bobot dari masing-masing lagu yang relevan menjadi urutan teratas untuk dipilih atau yang sedang populer.

DAFTAR PUSTAKA

[1] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schutze, 2008, Introduction to Information Retrieval, Cambridge University Press.
[2] Fatkhul Amin "Sistem Temu Kembali Informasi dengan Metode *Vector Space Model*." JSIB, Semarang, 2012
[3] Giat Karyono dan Fandy Setyo Utomo " Temu Balik Informasi pada Dokumen Teks Berbahasa Indonesia dengan Metode Vector Space Retrieval Model." Semantik, Semarang, 2012.
[4] http://lib.unnes.ac.id/18705/1/2503406561.pdf, diakses pada tanggal 16 September 2016 pukul 13.30