

Penggunaan Association Rule Learning pada Real-time Business Intelligence dengan Data Stream Mining

The Use of Association Rule Learning at Real-time Business Intelligence with Data Stream Mining

¹Faizal Hendyansyah Khrisdian, ²Shaufiah, ST., MT., ³Shinta Yulia Puspitasari, ST.

^{1,2,3}Prodi SI Teknik Informatika, Fakultas Teknik, Universitas Telkom

¹fkhrisdian@gmail.com, ²shaufiah@gmail.com, ³shinta1907@gmail.com

Abstrak

Dengan semakin berkembangnya teknologi, kebutuhan pemrosesan data menjadi semakin cepat. Semakin banyaknya data yang mengalir, menuntut proses analisis data yang cepat sehingga dapat menghasilkan keputusan dengan tingkat keterlambatan *minimum*. Tidak hanya pada level managerial atau strategis saja sebuah sistem business intelligence dibutuhkan, bahkan pada level operasional pun sudah mulai membutuhkan sistem *business intelligence* untuk meningkatkan efektifitas dan efisiensi kerja. Oleh sebab itu, dibangunlah sebuah sistem *real-time business intelligence* yang dituntut bekerja cepat untuk membantu kinerja pada level operasional. Dalam tugas akhir ini, dilakukan penelitian tentang sebuah sistem *real-time business intelligence* untuk memprediksi *delay* pada penerbangan pesawat pada PT Garuda Indonesia menggunakan metode *association rule learning* dengan algoritma *apriori*. Data penerbangan akan dilakukan *preprocessing* sebelumnya dengan cara menghilangkan data yang memiliki *missing value* dan melakukan perubahan data untuk merubah data yang berbentuk string menjadi integer agar dapat dihitung. Data tersebut nantinya akan diolah dan digali agar membentuk informasi atau pengetahuan yang dapat digunakan sebagai dasar pengambilan keputusan. Hasil dari proses *real-time analytic* akan dimasukkan ke dalam beberapa skenario pengujian dengan jumlah *minimum support* dan *minimum confident* yang berbeda, dan juga dengan proses penggunaan *apriori* yang berbeda.

Kata Kunci : Real-time Business Intelligence, Delay, Association Rule Learning, Apriori.

Abstract

With the development of the technology, the needs of data processing becomes faster. The manner of streaming data, forcing a faster data analyzing process so that can produce a decision with *minimum* latency. Not only on managerial or strategic level a business intelligence is needed, but also on operational level is starting need a business intelligence system to enhance their work effectivity and efficiency. Therefore, a real-time business intelligence that required to work faster to enhance the operational level performance is constructed. In this final project, conducted a research about a real-time business intelligence to predict a flight *delay* at PT Garuda Indonesia using association rule method with apriori algorithm. The flight data will go through a preprocessing process by removing data with a missing value and changing some variable to change string data in to integer so that it can be calculated. Those data will be processed and mined in order to forming some informations or knowledges as the base of decision making. The result of real-time analytic process will be conducted in several testing scenarios with different *minimumsupport* and *minimum confident*, and also with different use of apriori.

Keywords : Real-time Business Intelligence, Delay, Association Rule Learning, Apriori.

Pendahuluan

Sudah sekian lama *business intelligence (BI)* digunakan oleh organisasi untuk mendapatkan pengetahuan dari operasi bisnis mereka. *BI* dirumuskan kedalam rencana bisnis strategis dan taktis, dan biasanya diprakarsai dengan cara menganalisis data *historical*. Saat ini, bisnis berkembang menjadi lebih kompetitif dan dinamis dibandingkan dengan masa lalu dimana permintaan untuk sistem pendukung yang menunjang dan kemampuan dalam pengambilan keputusan dengan cepat meningkat. Dengan permintaan pasar bisnis yang baru ini, sebuah studi baru-baru ini, menganjurkan bahwa *BI* harus dispesifikasikan kedalam empat dimensi, yaitu : strategis, taktis, operasional dan *real-time*[1].

Pada abad ke-21, organisasi berubah menjadi bentuk baru berbasis pengetahuan dan jaringan dalam menanggapi lingkungan yang dikarakteristikan dengan batas-batas organisasi yang tidak jelas dan dengan perubahan yang sangat cepat [2]. Perusahaan mengalami perubahan lingkungan yang dihasilkan dari ekonomi baru dari informasi [4], dan kompetisi yang meningkat secara dinamis dan semakin global [3].

Sebuah temuan baru dari *BI*, yang bernama *real-time Business Intelligence (rt-BI)* muncul dan memiliki fungsi untuk *managing*, *monitoring* dan optimasi operasi bisnis harian secara real-time dan mendekati real-time. Selain itu, *rt-BI* diperkuat dengan analisis prediksi yang canggih melebihi data stream yang terus-menerus, *real-time monitoring* dan kecepatan dari *in-memory technology*. Keterlambatan (keterlambatan data, keterlambatan analisis, keterlambatan keputusan) idealnya harus bernilai

nol. Pendekatan utamanya adalah waktu respon dari sistem harus selalu berada di bawah ambang batas waktu (*time Threshold*) pengambilan aksi dan tingkat pemrosesan data harus lebih cepat dibandingkan dengan tingkat produksi data [4].

Pada karya tulis ini, salah satu metode yang digunakan untuk melakukan proses *data mining* adalah *stream mining*. Pada dasarnya, metode ini memiliki tujuan untuk mengatasi masalah dari pemrosesan jumlah data yang sangat banyak dan diproduksi secara terus menerus sehingga dapat memakan waktu yang lama atau bisa mengabiskan *memory* komputer pada saat pemrosesan berlangsung [5]. Dengan metode *stream mining*, diharapkan sistem *rt-BI* dapat mengeluarkan hasil analisis *BI* dengan tingkat keterlambatan yang sangat rendah atau mendekati nol. Data yang akan digunakan pada karya tulis ini adalah data yang dapat mempengaruhi penjadwalan penerbangan. Maka dari itu, diperlukanlah sebuah metode yang dapat melakukan analisis keterhubungan antar atribut yang dapat mempengaruhi penjadwalan dimana nantinya hasil analisis ini akan diubah menjadi pengetahuan atau aturan baru dalam penjadwalan.

1. Dasar Teori dan Perancangan

1.1 Real-time Business Intelligence

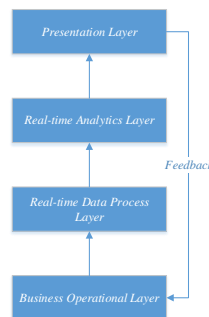
Pengertian dari *rt-BI* sangat bergantung pada pengertian dari apa itu “*real-time*” untuk sebuah bisnis. Tidak mengejutkan jika tidak ada definisi yang ditetapkan tentang *rt-BI*.

“*Real-time*” dapat berarti :

- Kebutuhan untuk mendapatkan proses dengan tingkat keterlambatan nol
- Bahwa sebuah proses menyediakan informasi kapanpun dibutuhkan oleh manajemen atau proses lain

Sebuah kemampuan untuk memperoleh pengukuran performansi kunci yang berhubungan dengan situasi pada masa sekarang dan tidak hanya pada beberapa situasi *historic*[7].

Sistem *rt-BI* berhubungan dengan banyak teknologi dan peralatan, dan berevolusi dari *strategic BI* dan *tactical BI*. Sebuah empat lapis *framework* diajukan untuk sebuah sistem *rt-BI* pada gambar 2.1 berikut. Perkembangan utamanya adalah sebuah pemrosesan secara *real-time* dari seluruh proses penemuan pengetahuan [4].



Gambar 1 Real-time Processing

Lapisan analisis menggunakan metode *data mining* yang cepat untuk merubah dari data ke informasi. Sejuah ini sudah banyak algoritma dan metodologi real-time data mining[8].

1.2 Data Stream Mining

Perhatian *data stream mining* adalah mengesktrak struktur pengetahuan yang direpresentasikan ke dalam model atau *pattern* dari *stream* informasi yang tak henti-henti [9].

Paradigma *data stream* timbul baru baru ini untuk menanggapi masalah data yang datang secara terus menerus. Algoritma yang ditulis untuk *data stream* dapat menanggulangi masalah ukuran data yang berkali lipat lebih besar dibandingkan ukuran *memory* komputer. Asumsi utama dari pemrosesan *data stream* adalah melatih contoh dapat diperiksa secara singkat hanya pada satu waktu. Contoh-contoh tersebut datang secara cepat dan berentetan kemudian harus dibuang untuk menyediakan ruang pada *memory* komputer untuk menampung contoh-contoh yang akan masuk berikutnya [5].

Data stream [10] memberikan solusi yang baik untuk membangun sebuah *real-time data warehouse*, dimana akan dapat meningkatkan frekuensi dari pembaruan data. Dengan kata lain, semakin cepat datawarehouse mengalami pembaruan data maka akan semakin cepat proses *real-time analytics* dieksekusi.

1.3 Association Rule Learning

Pada data mining, *association rule learning* adalah metode yang populer dan sudah diriset dengan baik untuk mencari hubungan antar variabel-variabel di dalam database yang besar [11]. *Association rule learning* mengeksplorasi data dengan cara mencari hubungan antar variabel, seperti produk-produk yang dibeli secara bersamaan, disebut juga *market basket analysis*.

Algoritma yang digunakan pada penelitian ini adalah algoritma *Apriori*. *Apriori* adalah sebuah algoritma seminal untuk menggali *frequent itemsets* untuk *Boolean Association Rule*. *Apriori* menggunakan pendekatan *iterative* yang dikenal dengan *level-wise search*. Hal yang paling pertama dilakukan adalah dengan cara mencari *frequent 1-itemset* dengan cara melakukan *scanning* pada *database* untuk menghitung *count* dari setiap *item* dan mengumpulkan *item* tersebut untuk memenuhi *minimum support*. Hasil dari pencarian tersebut dilambangkan dengan L_1 . Kemudian, L_1 akan digunakan untuk mencari L_2 . L_2 adalah sekumpulan *frequent 2-itemsets* dimana nantinya L_2 ini akan digunakan untuk mencari L_3 . Hal ini akan terus menerus diulang hingga tidak ada lagi *frequent k-itemsets*. Pencarian L_k membutuhkan satu proses *scanning database* secara menyeluruh [1].

Berikut adalah algoritma dari *apriori* :

```

Apriori(T, ε)
  L1 ← {large 1 – itemsets}
  k ← 2
  while Lk-1 ≠ ∅
    Ck ← {a ∪ {b} | a ∈ Lk-1 ∧ b ∉ a} – {c | {s | s ⊆ c ∧ |s| = k – 1} ⊄ Lk-1}
    for transactions t ∈ T
      Ct ← {c | c ∈ Ck ∧ c ⊆ t}
      for candidates c ∈ Ct
        count[c] ← count[c] + 1
    Lk ← {c | c ∈ Ck ∧ count[c] ≥ ε}
    k ← k + 1
  return ∪k Lk

```

Gambar 2 Algoritma Apriori [1]

Ketika semua *frequent k-itemsets* ditemukan dari transaksi di dalam database, tahap selanjutnya adalah untuk menghasilkan *association rule* yang kuat dari *frequent k-itemsets* tersebut (*association rule* yang kuat harus memenuhi *minimum support* dan *minimum confidence*). *Minimum confidence* dapat didapatkan menggunakan formula berikut [1] :

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support_count(A \cup B)}{support_count(A)}$$

Gambar 3 Formula untuk mencari Confidence[1]

Support_count (A ∪ B) adalah jumlah transaksi yang mengandung *itemset A ∪ B*, dan *support_count A* adalah jumlah transaksi yang mengandung *itemset A* [1].

Untuk mendapatkan *rulebase* yang terbaik, dalam artian *rulebase* memiliki tingkat akurasi kecocokan yang tinggi dan tidak memakan waktu lama dalam pembentukannya, maka diperlukan *minimum support* dan *minimum confidence* yang terbaik. *Minimum support* adalah jumlah minimal sebuah *itemset* muncul pada sebuah *dataset*, sedangkan *minimum confidence* adalah jumlah persentase terkecil dari sebuah *rule* agar dapat masuk ke dalam *rulebase*.

Efek dari *minimum support* adalah untuk mengefektifkan proses *apriori*. Jika *minimum support* terlalu rendah, maka proses dari penggalian data menggunakan *apriori* akan semakin memakan waktu lama. Hal ini dikarenakan oleh cara kerja *apriori* yang mencari seluruh *itemset* dan dikombinasikan menjadi beberapa kombinasi level *itemset*. Maka dari itu jika *minimum support* terlalu kecil, maka akan banyak *itemset* yang memiliki keterhubungan yang kecil akan ikut masuk ke dalam proses *apriori*. Namun, efek positif dari semakin kecilnya *minimum support* adalah akan semakin banyaknya variasi kombinasi *itemset* yang akan masuk ke dalam *rulebase*. Dengan semakin banyaknya variasi *rule*, maka akan meningkatkan kemungkinan akurasi kecocokan karena jumlah variasi yang banyak akan memungkinkan untuk mengantisipasi variasi data latih yang semakin banyak pula. Jika *minimum support* terlalu besar, maka efek negatifnya adalah akan semakin banyaknya *itemset* yang akan hilang dan tidak masuk ke dalam proses *apriori* karena tidak mampu memenuhi *minimum support*. Hal ini akan berakibat kebalikan dari *minimum support* yang terlalu kecil, yaitu tingkat akurasi akan semakin kecil karena variasi dari kombinasi *itemset* akan semakin sedikit dikarenakan jumlah *itemset* yang lolos *Threshold* menjadi semakin sedikit. Namun, efek positifnya adalah waktu pemrosesan *apriori* akan menjadi lebih cepat karena semakin sedikitnya jumlah *itemset* yang harus dikombinasikan menjadi beberapa level *itemset*. Maka dari itu jumlah *minimum support* yang optimum bisa menghasilkan *rulebase* yang memiliki tingkat akurasi yang tinggi namun tidak memakan waktu lama untuk menyelesaikan proses *apriori*.

Jika sudah ditemukan *minimum support* yang optimum, maka diperlukan *minimum confidence* yang sesuai agar *rule* yang sudah dihasilkan oleh algoritma *apriori* memiliki tingkat kecocokan yang tinggi dengan data uji yang akan diujikan di dalam sistem ini. *Minimum confidence* adalah persentase minimal sebuah *rule* agar bisa masuk ke dalam *rulebase*. Semakin tingginya persentase *confidence* sebuah *rule*, menunjukkan semakin besar kemungkinannya *rule* tersebut akan terjadi. Idealnya, *minimum confidence* harus bernilai tinggi agar kualitas *rule* yang dihasilkan semakin akurat dan semakin sering persentase terjadinya. Namun dengan terlalu tingginya *minimum confidence* akan membuat *rule* yang masuk ke dalam *rulebase* menjadi

sedikit dan jika terlalu rendah maka kualitas *rulebase* akan menjadi jelek. Maka dari itu dibutuhkan *minimum confidence* yang optimum agar *rulebase* dapat menangani banyak variasi dataset namun memiliki kualitas yang baik.

1.4 Flight Delay

Menurut Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 89 Tahun 2015 Tentang Penanganan Keterlambatan (*Delay Management*) Pada Badan Usaha Angkutan Udara Niaga Berjadwal Di Indonesia, Keterlambatan Penerbangan (*Flight Delay*) adalah terjadinya perbedaan waktu antara waktu keberangkatan atau kedatangan yang dijadwalkan dengan realisasi waktu keberangkatan atau kedatangan.

Menurut peraturan tersebut, keterlambatan penerbangan dikelompokkan menjadi 6 kategori keterlambatan :

- a. Kategori 1: keterlambatan 30 menit s/d 60 menit;
- b. Kategori 2 : keterlambatan 61 menit s/d 120 menit;
- c. Kategori 3 : keterlambatan 121 menit s/d 180 menit;
- d. Kategori 4 : keterlambatan 181 menit s/d 240 menit;
- e. Kategori 5 : keterlambatan lebih dari 240 menit;
- f. Kategori 6 : pembatalan penerbangan.

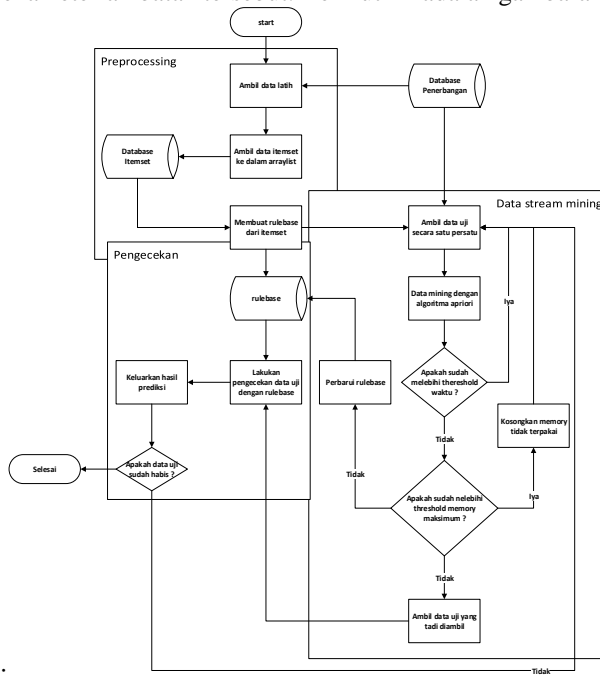
Namun berdasarkan data penerbangan PT Garuda Indonesia tahun 2014, keterlambatan penerbangan dibagi menjadi 4 kategori keterlambatan (tidak termasuk dengan kategori pembatalan penerbangan) :

- a. Kategori 1 : Keterlambatan 15 menit s/d 30 menit;
- b. Kategori 2 : Keterlambatan 30 menit s/d 120 menit;
- c. Kategori 3 : Keterlambatan 120 menit s/d 240 menit;
- d. Kategori 4 : Keterlambatan lebih dari 240 menit.

Menurut peraturan PT Garuda Indonesia, penerbangan dianggap *delay* jika perbedaan antara waktu keberangkatan atau waktu kedatangan yang sudah dijadwalkan dengan realisasi terlambat 15 menit atau lebih, sehingga jika selisihnya kurang dari 15 menit maka penerbangan tersebut tidak dihitung terlambat atau termasuk dalam Kategori 0.

1.5 Perancangan Sistem

Dalam tugas akhir ini, sistem yang dibangun dengan menggunakan metode *real-time business intelligence* dan *data stream mining*. Penggunaan *real-time business intelligence* bertujuan untuk menganalisa prediksi dari sebuah penerbangan tidak hanya menggunakan data yang telah lampau, tetapi juga menggunakan *current data*. Sedangkan penggunaan *data stream mining* digunakan untuk memastikan proses *mining* dilakukan dengan cepat agar tingkat keterlambatan bernilai nol dan tidak ada data yang terbuang karena keterlambatan tersebut. Berikut ini adalah gambaran umum dari sistem



Gambar 4 Gambaran Umum Sistem

Dari gambaran umum di atas, terdapat dua proses utama yang dilakukan :

1. Preprocessing

Pada proses ini, data uji yang sudah ditentukan sebelumnya dilakukan pemecahan menjadi *itemset*. Setiap satu *record* data dipecah menjadi beberapa *itemset* berdasarkan tiap atribut yang akan dipakai dari data tersebut. Setelah seluruh *itemset* didapat, maka dilakukan proses *data mining* dengan metode *apriori* untuk mendapatkan *rulebase* awal sebelum dilakukannya *data stream mining*.

2. Data stream mining

Pada proses ini, data uji yang sudah ditentukan diambil satu persatu untuk dilakukan *mining* dengan menggunakan metode *apriori*. Skema *data stream mining* digunakan dengan cara menggunakan *Threshold* waktu dan ukuran *memory* yang digunakan. Jika *threshold* waktu sudah terlewati (terlambat), maka data yang sedang diproses akan langsung dibuang walaupun belum selesai diproses dan belum menghasilkan hasil prediksi dan data selanjutnya diambil untuk dilakukan proses *mining*. Jika *Threshold memory* sudah terlewati, maka akan dilakukan penghapusan *memory* yang sudah tidak terpakai agar bisa dipakai lagi, hal ini akan memungkinkan untuk tidak terjadinya *out of memory* karena *memory* sudah penuh dan tidak bisa lagi menampung proses selanjutnya.

3. Pengecekan

Pada proses ini, data uji akan dicocokkan dengan *rulebase* yang sudah ada. Proses pengecekan dilakukan dengan cara mencocokkan model data uji dengan model yang ada di *rulebase*. Jika ada model pada *rulebase* yang cocok, maka hasil prediksi akan disesuaikan dengan model tersebut. Jika tidak ada model yang cocok, maka data tersebut akan dimasukkan ke dalam kategori *not found* (data yang tidak memiliki kesamaan model pada *rulebase*).

2. Pembahasan

Pengukuran performansi dari sistem yang dibangun akan dilihat dari beberapa bagian, meliputi akurasi prediksi, tingkat *data not found* dan tingkat kecepatan sistem dalam mengeluarkan hasil prediksi penerbangan seperti berikut :

- Akurasi ADA (Actual Delta Arrival) adalah penghitungan total data prediksi “da” yang didapat dari model *rulebase* yang sesuai dengan selisih kedatangan pesawat sebenarnya dibagi dengan total data pesawat yang memiliki kesamaan dengan salah satu model *rulebase*.
- Akurasi *estimateddelay* adalah penghitungan total data prediksi *delay* yang didapatkan dari model *rulebase* yang sesuai dengan kondisi *delay* pesawat sebenarnya dibagi dengan total data pesawat yang memiliki kesamaan dengan salah satu model *rulebase*.
- Frekuensi *data not found* adalah total data yang tidak sesuai sama sekali dengan model *rulebase* dibagi dengan total semua data uji. Hal ini berguna untuk menganalisis *Thresholdminimumsupport* dan *minimumconfidence* agar bisa didapatkan *minimumsupport* dan *minimumconfidence* yang optimal.
- Rata-rata waktu yang dibutuhkan untuk mengeluarkan sebuah prediksi *delay* untuk setiap *record* data uji. Hal ini berfungsi untuk melihat berapa *Threshold* optimum yang sesuai untuk mendapatkan hasil yang terbaik.

3.1 Skenario

Skenario pengujian ini berdasarkan pemilihan variasi dari *Threshold* yang akan digunakan pada sistem ini. *Threshold* yang akan diubah-ubah nilainya adalah *minimumconfidence*, *minimumsupport*, maksimal waktu pemrosesan *data stream mining* dan maksimal *memory* yang digunakan untuk melakukan proses *data stream mining*. Sehingga ketentuan skenario yang akan dilakukan sebagai berikut:

- a. Untuk *minimumconfidence* pada setiap pengujian akan digunakan nilai 0%, 10%, 20%, 30%, 40%, 50%, 60% dan 70%.
- b. Untuk *minimumsupport* pada setiap pengujian akan digunakan nilai 0, 5 dan 7.

3.2 Analisis

Gambar 5 Grafik Akurasi Prediksi Estimated Delay

Dari Gambar 8 di atas dapat terlihat semakin tinggi *minimumconfidence* maka semakin tinggi tingkat akurasi prediksi *delay*. Hal ini disebabkan oleh dengan semakin tingginya *minimumconfidence*, maka semakin tinggi pula kemungkinan prediksi dari *rulebase* akan terjadi atau bisa dikatakan kualitas *rule* yang ada pada *rulebase* semakin bagus karena semakin mendekati pada kenyataan yang terjadi.

Gambar 6 Gradik Data Not Found

Dari Gambar 9 di atas dapat dilihat bawasanya semakin tingginya *minimumsupport* dan *minimumconfidence* maka *data not found* semakin tinggi. Hal ini disebabkan oleh dengan berkurangnya ragam *itemset* dan juga *rule* yang ada pada *rulebase*. Dengan jumlah *itemset* yang sedikit ragamnya maka akan berpengaruh dengan ragam *rule* yang dihasilkan, ditambah dengan tingginya *minimumconfidence* dari sebuah *rulebase*, maka *rule* pun menjadi semakin sedikit dan mengakibatkan data yang tidak memiliki kecocokan sama sekali dengan *rule* yang ada pada *rulebase*. Algoritma apriori menggunakan teknik pencocokan untuk data uji yang akan diprediksi dengan *rule* yang ada pada *rulebase*. Jika tidak memiliki kecocokan sama sekali, maka data tersebut tidak mengalami prediksi melainkan hanya mengalami proses mining saja sehingga tidak keluar hasil prediksi penerbangannya.

Gambar 7 Grafik Total Running Time

Dari Gambar 10 diatas dapat dilihat bahwa semakin tinggi *minimumsupport* maka semakin kecil total running time untuk melakukan analisis data stream mining. Hal ini disebabkan jumlah ragam *itemset* yang berkurang sehingga proses apriori menjadi semakin cepat. Dengan berkurangnya jumlah ragam *itemset*, maka proses apriori yang mengkombinasikan *itemset* menjadi beberapa level akan menjadi semakin cepat. *Itemset* yang tidak memenuhi thershold tidak akan dimasukkan ke dalam proses mining apriori sampai *itemset* tersebut dapat memenuhi *minimumsupport* yang sudah ditentukan.

Gambar 8 Grafik Penggunaan Memory Computer

Pada Gambar 11 ditunjukkan grafik penggunaan *memory computer* untuk pemrosesan setiap record data uji yang masuk pada scenario pengujian dengan *minimum confidence* sebesar 30%, *minimum support* sebesar 7 dan *time threshold* selama 1 detik. Dari bentuk grafik bias dilihat bawasanya penggunaan *memory computer* akan menanjak dan pada titik tertentu pneggunaan *memory computer* akan turun kembali seperti semula tetapi sedikit lebih tinggi dibandingkan posisi awal, lalu kembali naik lagi dan kembali turun lagi dan begitu seterusnya sampai akhir.

Grafik penggunaan *memory computer* tidak menanjak terus karena ada pengimplementasian *data stream mining* dimana memory yang sudah tidak dipakai akan dikosongkan agar bias dipakai lagi untuk pemrosesan selanjutnya. Jika tidak diimplementasikan *metode data stream* mining, maka aka nada kemungkinan penggunaan *memory computer* akan terus menanjak tanpa ada pengosongan *memory computer* yang sudah dipakai dan akan mengaikabatkan *out of memory*.

4. Kesimpulan

Berdasarkan pengujian serta analisis yang telah dilakukan pada tugas akhir ini, dapat diambil beberapa kesimpulan, yaitu :

1. Semakin tinggi *minimumconfidence* dan *minimumsupport* dapat mengakibatkan :
 - meningkatkan akurasi dari prediksi *delay*.
 - meningkatnya data yang tidak memiliki kecocokan sama sekali dengan *rule* yang ada pada *rulebase* (*data not found*).
 - semakin cepatnya total waktu analisis semua data uji dan waktu pemrosesan setiap data *record*.
2. Data stream mining cocok untuk diimplementasikan dengan sistem real-time business intelligence karena dengan metode data stream mining memaksa sistem untuk melakukan analisis dengan cepat. Pada tugas akhir ini dengan

memberikan dan mengatur *Threshold* waktu dan memori, sistem dapat melakukan proses analytic tanpa mengalami latency dan tanpa mengalami out of *memory*.

3. Algoritma apriori dapat diimplementasikan pada data stream mining dengan cara membatasi waktu dan memori pada saat proses mining dilakukan. Pada tugas akhir ini, dengan menggunakan data penerbangan Garuda Indonesia pada tahun 2014, dapat dibuktikan bawasanya apriori dapat mengemban tugas sebagai pencari basis aturan dengan range akurasi pada 70%-82% dengan batasan waktu permosesan maksimal selama satu detik dan penggunaan *memory* maksimal sebesar 100 Mb.

Daftar Pustaka :

- [1] S. Asghar, S. Fong dan T. Hussain, "Business Intelligence Modelling : A Case Study of Disaster Management Organization in Pakistan," dalam *The 4th International Conference of Computer Sciences and Convergence Information Technology*, Seoul, 2009.
- [2] J. Schiefer dan A. Seufert, "Enhanced Business Intelligence - Supporting Business Processes with Real-time Business Analytics," dalam *The 16th International Workshop on Dexa '05*, 2005.
- [3] R. M. D'Aveni, "Hypercompetition," dalam *The Free Press*, New York, 1994.
- [4] Y. Hang dan S. Fong, *Real-time Business Intelligence System Architecture with Stream Mining*, Macau: IEEE, 2010.
- [5] A. Bifet dan R. Kirkby, *Data Stream Mining : A Practical Approach*, New Zealand: Center for Open Software Innovation, 2009.
- [6] Y. Malhotra, *From Information Management to Knowledge Management : Beyond "Hi-Tech Hidebound"*, Medford: NJ, 2000.
- [7] B. Azvine, Z. Cui, D. D. Nauck dan B. Majeed, "Real-time Business Intelligence for the Adaptive Enterprise," dalam *the 8th IEEE CEC/EEE '06*, 2006.
- [8] G. Dong, J. Han, L. Lakshmanan, J. Pei, H. Wang dan P. Yu, "Online Mining of Changes from Data Stream : Research Problem and Preliminary Results," dalam *ACM SIGMOD Workshop on Management and Processing of Data Streams*, San Diego, 2004.
- [9] M. M. Gaber, A. Zaslavsky dan S. Krinashwami, "Mining Data Stream : A Review," vol. 34, no. 2, p. 18, 2005.
- [10] I. Botan, Y. Cho, R. Derakhsan, N. Dindar, L. Haas, K. Kim dan N. Tatbul, "Federated Stream Processing Support for Real-time Business Intelligence Applications," dalam *VLDB International Workshop on BIRTE '09*, Lyon, 2009.
- [11] G. Piatetsky-Shapiro, "Discovery, Analysis dan Presentation of Strong Rules," dalam *AAA/MIT Press*, Cambridge, 1991.
- [12] Evans dan Wurster, "Blown to Bits," dalam *Harvard Business School Press*, Boston, 2000.
- [13] S. Asghar, S. Fong dan T. Hussain, "Business Intelligence Modelling : A Case Study of Disaster Management Organization in Pakistan," dalam *The 4th International Conference of Computer Sciences and Convergence Information Technology*, Seoul, 2009.
- [14] J. Schiefer dan A. Seufert, "Enhanced Business Intelligence - Supporting Business Processes with Real-time Business Analytics," dalam *The 16th International Workshop on Dexa '05*, 2005.
- [15] R. M. D'Aveni, "Hypercompetition," dalam *The Free Press*, New York, 1994.