

Analisis Perbandingan Algoritma FP-Growth dan CP-Tree untuk Data Teks

Dian Sa'adillah Maylawati¹

Jurnal Informatika
Sekolah Tinggi Teknologi Garut
Jl. Mayor Syamsu No. 1 Jayaraga Garut 44151 Indonesia
Email : jurnal@sttgarut.ac.id

¹dsaadillah@sttgarut.ac.id

Abstrak – *Frequent Pattern Growth* (FP-Growth) dan *Compact Pattern Tree* (CP-Tree) adalah algoritma *Frequent Itemset Mining* (FIM) yang menghasilkan *frequent itemset* dari transaksi *database*. *Frequent itemset* dapat digunakan sebagai representasi terstruktur untuk data teks yang merupakan data tidak terstruktur atau semi terstruktur. CP-Tree adalah algoritma FIM yang dikembangkan dari algoritma FP-Growth. Namun, CP-Tree melakukan proses data secara inkremental sedangkan FP-Growth non-inkremental. Artikel ini membahas analisis terhadap algoritma FP-Growth dan CP-Tree dalam menghasilkan representasi terstruktur dari data teks. Berdasarkan hasil analisis dan evaluasi terhadap algoritma FP-Growth CP-Tree diperoleh bahwa *frequent itemset* yang dihasilkan dari representasi pohon kedua algoritma tersebut sama. Secara proses algoritma FP-Growth lebih sederhana dibandingkan algoritma CP-Tree. Namun, algoritma CP-Tree lebih fleksibel terhadap penambahan transaksi baru dibandingkan algoritma FP-Growth. Hal ini dikarenakan CP-Tree tidak mengulang dari awal untuk proses *scanning* data dan membuat struktur pohon seperti FP-Growth apabila ada data transaksi baru.

Kata Kunci – Algoritma, Compact Pattern Tree, Frequent Pattern Growth, Frequent Itemset, Frequent Pattern Mining, Teks Mining.

I. PENDAHULUAN

Perangkat teknologi dirancang untuk meningkatkan kualitas hidup manusia [1], salah satunya adalah teknologi sistem informasi yang dapat mengefektifkan dan mengefisiensikan berbagai aktifitas. Sosial media adalah salah satu media yang memanfaatkan teknologi informasi. Setiap menitnya menghasilkan data teks dalam jumlah besar, dimana dari data teks tersebut dapat dicari informasi dan pengetahuan yang lebih penting [2]–[4]. Sistem informasi adalah kombinasi berbagai komponen teknologi informasi dan aktifitas manusia yang terdiri dari prosedur atau bisnis proses [5], secara umum digunakan untuk mendukung kegiatan manajemen dan operasional [6]. Sistem informasi mengelola proses terhadap data [7], level tertinggi dari sistem informasi adalah fleksibilitas dan skalabilitas dalam membangun sistem [8]. Berdasarkan beberapa penelitian, sistem informasi memiliki kapabilitas yang tinggi dalam mendukung keputusan, sistem memiliki akses data yang akurat dan berjalan dengan efisien [9], memiliki tingkat akurasi yang tinggi [10], mendukung keputusan untuk menyelesaikan berbagai masalah [11], rendah biaya [12], akses yang luas [13], menerima dan memberi pengetahuan terhadap penggunanya [14], meningkatkan produktivitas [15], menyediakan data dan informasi dengan lebih baik [16], dan dalam kasus khusus sangat berpotensi untuk menjadikan penyimpanan data sebagai media pengetahuan [17].

Teks merupakan salah satu data tidak terstruktur yang perlu dilakukan penanganan khusus sebelum diproses lebih lanjut [4], [18], misalnya sebelum dilakukan proses *teks mining*, *information retrieval*, maupun *natural language processing*. Di era digital dan sosial media saat ini, data teks yang mengalir setiap harinya dapat dimanfaatkan untuk memperoleh informasi penting, bahkan pengetahuan di dalamnya. *Teks mining* adalah sebuah teknik untuk menemukan nilai-nilai penting yang sebelumnya tidak diketahui secara otomatis dengan mengekstrak data teks sehingga diperoleh

pengetahuan yang bermanfaat [19]–[21]. Teks mining atau yang dikenal juga dengan teks data mining adalah bagian dari data mining [22], [23]. Perbedaan keduanya adalah pada data mining, data yang diolah adalah data terstruktur, sedangkan pada teks mining, data yang diolah adalah data teks yang tidak terstruktur atau semi terstruktur [4], [24], [25]. Oleh karena itu, data teks perlu direpresentasikan dalam bentuk data terstruktur supaya dapat dilakukan proses mining.

Representasi terstruktur dari sebuah teks secara umum terdapat dua jenis, yaitu bentuk *single word* atau yang lebih dikenal dengan *bag of words* dan bentuk *multiple words*. *Bag of word* adalah bentuk representasi terstruktur dari teks dengan mengumpulkan seluruh kata dalam dokumen teks tanpa melihat keterkaitan antar kata [26]–[28], sedangkan representasi *multiple word* adalah representasi teks yang mengumpulkan kata-kata pada dokumen teks dengan memperhatikan keterkaitan antar kata sehingga dengan representasi *multiple word* makna semantik pada dokumen teks dapat terjaga dengan lebih baik, karena dapat menangkap hubungan antar kata/frasa, bahkan hingga klausa dan kalimat [29]. *Frequent pattern* adalah salah satu bentuk representasi *multiple of words*, sehingga representasi terstruktur teks yang dihasilkan akan menjaga makna dari teks tersebut [3], [30]–[32]. *Frequent Pattern Mining* atau *Frequent Itemset Mining* (FIM) adalah salah satu teknik data mining yang menghasilkan pola berupa himpunan *item* yang *frequent* [3], [4], [33], [34]. Sejak tahun 1993 awal kemunculan FIM hingga awal tahun 2018 setidaknya terdapat 57 algoritma FIM [35], termasuk algoritma FP-Growth dan CP-Tree. Pada artikel ini akan dibahas analisis terhadap algoritma FP-Growth dan CP-Tree dalam menghasilkan representasi terstruktur dari data teks berupa *frequent itemsets*.

II. FREQUENT ITEMSET MINING

FIM yang dahulu dikenal *large itemset mining* [33], [36], bekerja untuk menemukan kumpulan item (itemset) yang frequent dari transaksi pada database [35], [37], [38]. Item dikatakan frequent ketika memenuhi nilai batas ambang (threshold value) atau minimum support. Dimana, minimum support menandakan banyaknya kemunculan itemset yang harus dipenuhi dari keseluruhan transaksi yang terdapat pada database. Berbeda dengan *sequential pattern mining*, FIM menghasilkan pola dengan item yang muncul selalu bersamaan tanpa memperhatikan urutan kemunculan. FIM pada teks dikenal pula dengan frequent word itemset (FWI) [3], [4], sebagai salah satu representasi teks terstruktur. FWI memiliki struktur dengan memandang dokumen atau kumpulan data teks sebagai himpunan pola itemset. Struktur FWI tersebut dapat diilustrasikan dengan $\{(w_1, w_2), (w_3, w_4), \dots\}$ dimana (w_1, w_2) adalah FWI_i , (w_3, w_4) adalah FWI_{i+1} , etc. Urutan kemunculan himpunan FWI sesuai dengan urutan kemunculannya dalam dokumen atau kumpulan data teks, tetapi kemunculan elemen pada FWI tidak harus urutan kemunculan kata pada dokumen atau kumpulan data teks. Hal ini bermakna dalam koleksi dokumen FWI_i sering muncul diikuti dengan FWI_{i+1} dan seterusnya. Elemen atau item dalam FWI_i , w_1 selalu muncul bersamaan dengan w_2 tanpa harus berurutan w_1 diikuti w_2 , jika w_2 muncul lebih dulu dari w_1 akan dianggap sebagai FWI yang sama, begitupun kemunculan elemen pada FWS_{i+1} dan seterusnya.

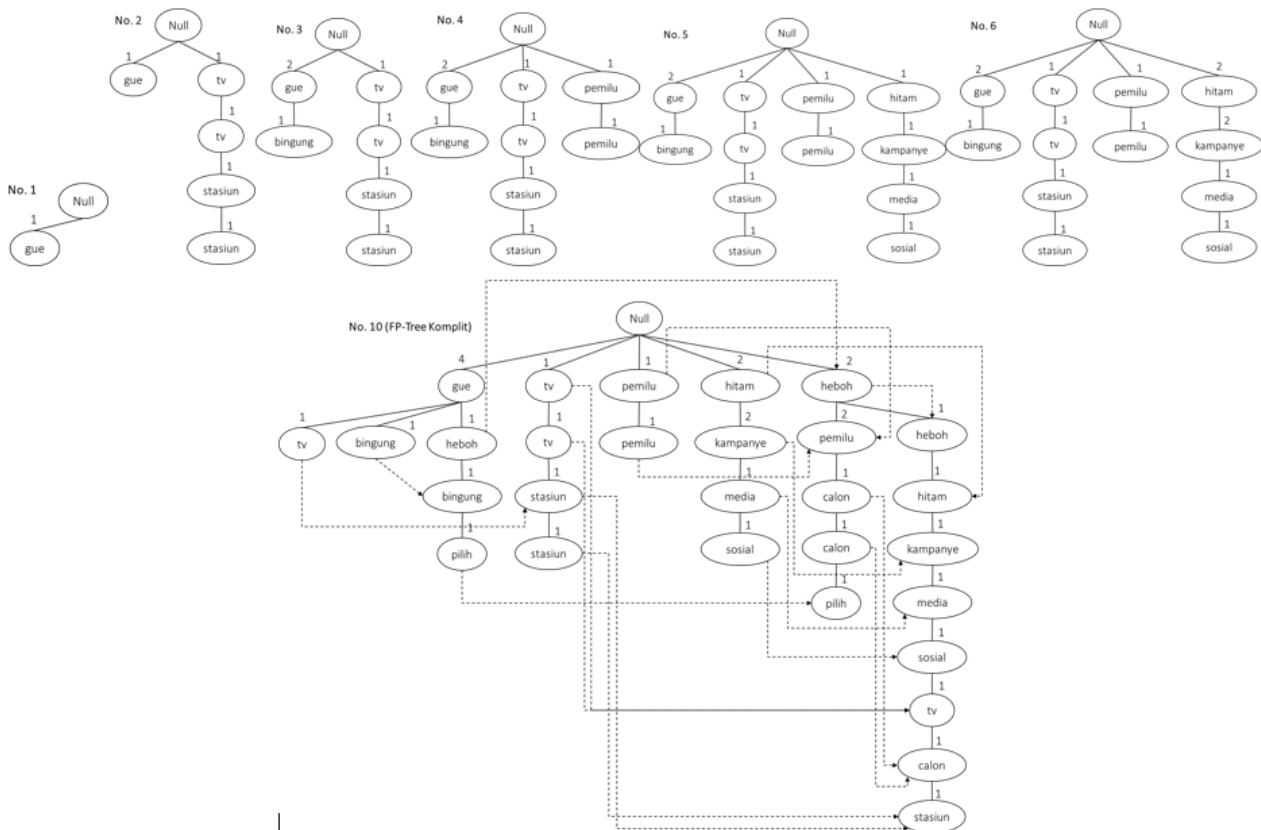
III. ALGORITMA FP-GROWTH UNTUK TEKS

Algoritma FP-Growth adalah algoritma yang dikembangkan dari algoritma Apriori. FP-Growth adalah metode dalam menghasilkan *frequent itemset* tanpa melakukan pembangkitan kandidat fitur [24][39]. Pada FP-Growth, database dikompresi menjadi struktur pohon. Ada dua langkah utama dalam menghasilkan frequent itemset tanpa membangkitkan kandidat fitur, yaitu membangun struktur data yang disebut dengan FP-Tree dan mengekstraksi frequent itemset langsung dari FP-Tree yang sudah terbentuk tersebut. Langkah-langkah tersebut antara lain: melakukan scan database dan menemukan support dari masing-masing item; mengeliminasi semua item yang tidak frequent; mengurutkan frequent items berdasarkan support-nya. Mulai dari yang support yang tertinggi hingga terendah; setiap transaksi yang dibaca dan disimpan petakan pada tree; root pertama dilabeli dengan null, kemudian item dari setiap transaksi akan disimpan

dalam node yang bersisi item-name, count, dan node-link; mengkesktrak frequent itemsets dari FP-Tree; mMem baca dari daun ke akar (algoritma *buttom-up*); dan melakukan *divide and conquer*. Misalnya terdapat 3 buah dokumen teks seperti pada Tabel 1, setelah dilakukan pra-proses terhadap data teks, seperti *tokenizing*, *stopwords removal*, *stemming*, dan lain sebagainya [4], setiap kata hasil pra-proses tersebut akan dimasukkan dalam struktur pohon pada Gambar 1.

Tabel 1: Contoh Koleksi Dokumen

No	Isi Dokumen	Hasil Pra-proses
1	Gue nonton tv berita ga ada yang netral apa? Stasiun tv M dukung nomer 1, stasiun tv O dukung nomer 2. Gue bingung mana yang bener mana engga. Pemilu oh pemilu.	gue nonton tv berita netral stasiun tv dukung nomer stasiun tv dukung nomer gue bingung bener engga pemilu oh pemilu
2	Sosial media sekarang jadi ajang sebar-sebar kampanye hitam. Stop kampanye hitam!! Ayo ciptakan pemilu damai, ga usah pake heboh belain calon pilihannya sampe nyebar isu ga jelas tentang calon lainnya.	sosial media jadi ajang sebar sebar kampanye hitam stop kampanye hitam ayo cipta pemilu damai usah pake heboh belain calon pilih sampe sebar isu jelas calon
3	Pemilu presiden taun ini heboh dengan kampanye hitam di sosial media, heboh dengan stasiun tv yang cuma mihak satu calon. Gue? Mau pilih yang mana aja gue masing bingung, gimana mau ikutan heboh?	pemilu presiden taun heboh kampanye hitam sosial media heboh stasiun tv mihak satu calon gue pilih aja gue bingung gimana ikutan heboh



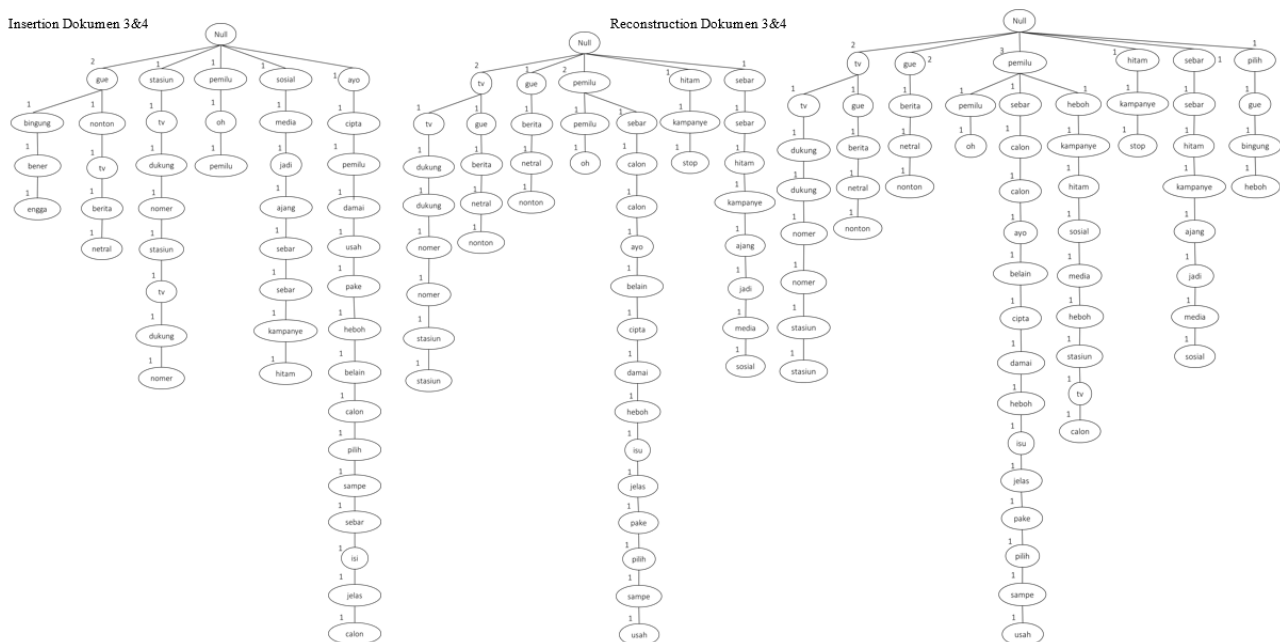
Gambar 1: Struktur Pohon hasil algoritma FP-Growth

IV. ALGORITMA CP-TREE UNTUK TEKS

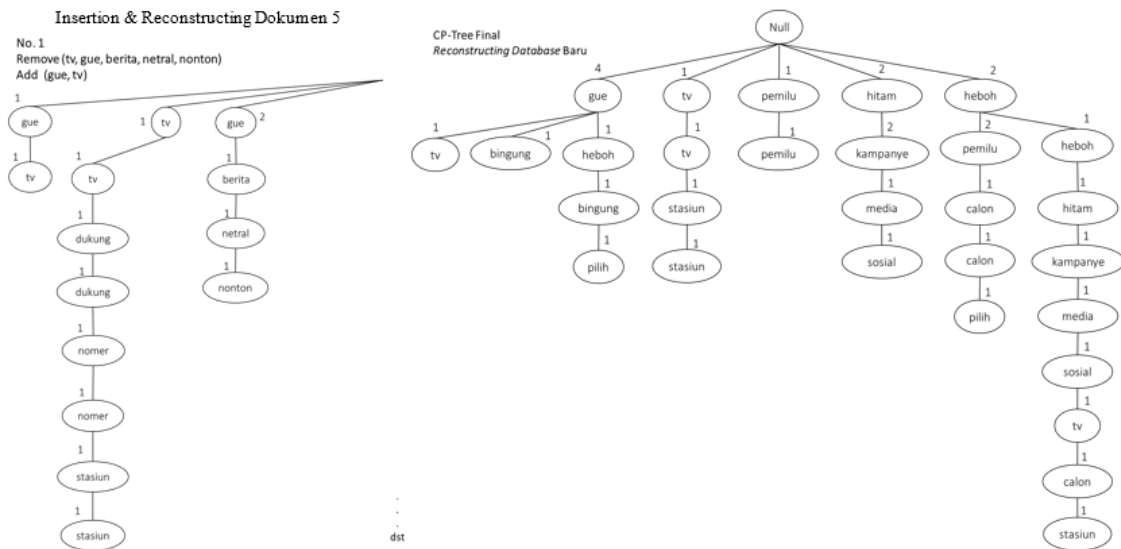
CP-Tree atau *Compact Pattern Tree* adalah algoritma yang dikembangkan dari algoritma FP-Growth sehingga dapat digunakan untuk *interactive* dan *incremental mining* [40], [41]. Dimana *incremental data mining* adalah salah satu teknik untuk menemukan *knowledge* dari data yang besar

[24]. Teknik ini dilakukan untuk mendapatkan pola baru saat *database* yang ada di tambahkan dengan transaksi baru. Misalnya DB adalah *database* yang asli dan *minSupp* adalah *support* minimum dari DB. Kemudian db adalah *increment database* dari transaksi baru yang ditambahkan pada DB. $U = DB \cup db$, dimana U adalah *database update* yang berisi semua transaksi dari DB dan db. Jika L^{DB} adalah *frequent pattern* dari DB, maka pada *incremental mining* yang dicari adalah L^U yaitu *frequent pattern* dari U dengan memanfaatkan L^{DB} yang sudah terbentuk. Sebuah *pattern* A dikatakan *frequent* pada *database update* U jika *support* dari A dalam U lebih besar sama dengan *minSupp* dikali |U|. Jika DB dan U memiliki minimal *support* yang sama, *frequent pattern* dari DB belum tentu *frequent* dalam U. Di sisi lain *pattern* yang tidak *frequent* dalam DB bisa menjadi *frequent* dalam U.

FP-Tree yang digunakan pada algoritma FP-Growth membaca *database* sebanyak dua kali sehingga tidak sesuai dengan kebutuhan *data stream processing*. Algoritma CP-Tree melakukan modifikasi pada FP-Tree sehingga *database* cukup dibaca sekali. Algoritma ini cukup sederhana dengan dua tahap, yaitu tahap *insertion* dan *reconstructing*. Tahap *insertion* adalah tahap membaca *database* kemudian memasukkan daftar *items* pada I-List (*Item List*) berdasarkan *lexicographical item order*, setiap transaksi dimasukkan ke dalam CP-Tree berdasarkan I-List. Pembentukan *node* pada CP-Tree juga harus memenuhi minimal *support* seperti pada FP-Tree. Sedangkan tahap *reconstructing* adalah tahap mengurutkan kembali *items* pada I-List berdasarkan *frequency-decending order* (I-List sort), kemudian CP-Tree yang terbentuk dari tahap *insertion* diurutkan kembali berdasarkan I-List sort. Tahap *reconstructing* ini mirip dengan teknik konstruksi pada FP-Tree. Pada dasarnya modifikasi yang dilakukan algoritma CP-Tree hanya pada struktur *tree* yang dibangun saja, sedangkan *frequent pattern* yang dibentuk sama seperti FP-Growth dengan membangun *prefix tree* dan mengambilnya secara *bottom-up*. Pada dasarnya proses mendapatkan *frequent itemset* dari algoritma CP-Tree sama dengan FP-Growth. Perbedaanya CP-Tree memiliki tahap *insertion* dan *reconstructing* yang memungkinkan struktur *tree* yang dihasilkan dapat di-update jika ada *database* baru yang ditambahkan sesuai dengan prinsip *incremental mining*. Sehingga *scanning* pada *database* original hanya dilakukan satu kali. Selanjutnya secara berulang, sebanyak *database* yang ditambahkan, tahap *insertion* dan *reconstructing* diulang. Jika sudah tidak ada *database* yang ditambahkan maka *frequent pattern* akan diekstrak dari *tree* hasil tahap *reconstructing* terakhir. Masih menggunakan contoh dokumen yang sama pada Tabel 1, Gambar 2 adalah proses algoritma CP-Tree untuk *incremental database* dengan original *database* adalah dokumen 3 dan 4, sedangkan *database* yang ditambahkan adalah dokumen 5 (Gambar 3).



Gambar 2: Struktur Pohon hasil algoritma CP-Tree untuk Dokumen 3 dan 4



Gambar 3: Struktur Pohon hasil algoritma CP-Tree untuk penambahan Dokumen 5

V. KESIMPULAN/RINGKASAN

Algoritma FP-Growth dan CP-Tree adalah algoritma FIM yang menghasilkan *frequent itemset* dari transaksi *database*. Pada dasarnya seperti algoritma FIM lain, FP-Growth dan CP-Tree digunakan untuk data terstruktur, tetapi tidak menutup kemungkinan digunakan untuk data tidak terstruktur seperti teks. FP-Growth dan CP-Tree sama-sama tidak membangkitkan kandidat fitur, melainkan merepresentasikan data teks dalam struktur pohon dan *feature itemsets* dihasilkan dari pohon tersebut. Pada hasil analisis algoritma FP-Growth dan CP-Tree untuk teks diketahui bahwa struktur pohon dan *feature itemsets* yang dihasilkan sama. Namun, FP-Growth dan CP-Tree berbeda dari teknik dan prinsip yang dijalankan pada proses pembentukan struktur pohonnya. FP-Growth tidak menggunakan teknik inkremental sehingga apabila ada data baru yang masuk, struktur pohon harus dibuat dari awal. Sedangkan CP-Tree menggunakan teknik inkremental yang memungkinkan memproses data baru tanpa mengulang membuat struktur pohon dari awal.

DAFTAR PUSTAKA

- [1] M. A. Ramdhani, H. Aulawi, A. Ikhwana, and Y. Mauluddin, "Model of green technology adaptation in small and medium-sized tannery industry," *J. Eng. Appl. Sci.*, vol. 12, no. 4, pp. 954–962, 2017.
- [2] Jumadi, D. S. A. Maylawati, B. Subaeki, and T. Ridwan, "Opinion mining on Twitter microblogging using Support Vector Machine: Public opinion about State Islamic University of Bandung," in *Proceedings of 2016 4th International Conference on Cyber and IT Service Management, CITSM 2016*, 2016.
- [3] D. Sa' Adillah Maylawati and G. A. Putri Saptawati, "Set of Frequent Word Item sets as Feature Representation for Text with Indonesian Slang," in *Journal of Physics: Conference Series*, 2017, vol. 801, no. 1.
- [4] D. S. A. Maylawati, "PEMBANGUNAN LIBRARY PRE-PROCESSING UNTUK TEXT MINING DENGAN REPRESENTASI HIMPUNAN FREQUENT WORD ITEMSET (HFWI) Studi Kasus: Bahasa Gaul Indonesia," Bandung, 2015.
- [5] A. Pamoragung, K. Suryadi, and M. A. Ramdhani, "Enhancing the implementation of e-Government in indonesia through the high-quality of virtual community and knowledge portal," in *Proceedings of the European Conference on e-Government, ECEG*, 2006, pp. 341–348.
- [6] M. A. Ramdhani, *Metodologi Penelitian untuk Riset Teknologi Informasi*. Bandung: UIN Sunan Gunung Djati Bandung, 2013.
- [7] D. S. Maylawati, W. Darmalaksana, and M. A. Ramdhani, "Systematic Design of Expert System Using Unified Modelling Language," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 288, no. 1, p. 12047, 2018.
- [8] H. Aulawi, M. A. Ramdhani, C. Slamet, H. Ainissyifa, and W. Darmalaksana, "Functional Need Analysis of Knowledge Portal Design in Higher Education Institution," *Int. Soft Comput.*, vol. 12, no. 2, pp. 132–141, 2017.
- [9] C. Slamet, A. Rahman, A. Sutedi, W. Darmalaksana, M. A. Ramdhani, and D. S. Maylawati, "Social Media-Based Identifier for Natural Disaster," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 288, no. 1, p. 12039, 2018.

- [10] C. Slamet, R. Andrian, D. S. Maylawati, W. Darmalaksana, and M. A. Ramdhani, "Web Scraping and Naïve Bayes Classification for Job Search Engine," vol. 288, no. 1, pp. 1–7, 2018.
- [11] Y. A. Gerhana, W. B. Zulfikar, A. H. Ramdani, and M. A. Ramdhani, "Implementation of Nearest Neighbor using HSV to Identify Skin Disease," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 288, no. 1, p. 012153 1234567890 Implementation, 2018.
- [12] A. Rahman, C. Slamet, W. Darmalaksana, Y. A. Gerhana, and M. A. Ramdhani, "Expert System for Deciding a Solution of Mechanical Failure in a Car using Case-based Reasoning," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 288, no. 1, p. 12011, 2018.
- [13] C. Slamet, A. Rahman, M. A. Ramdhani, and W. Darmalaksana, "Clustering the Verses of the Holy Qur'an Using K-Means Algorithm," *Asian J. Inf. Technol.*, vol. 15, no. 24, pp. 5159–5162, 2016.
- [14] D. S. Maylawati, M. A. Ramdhani, W. B. Zulfikar, I. Taufik, and W. Darmalaksana, "Expert system for predicting the early pregnancy with disorders using artificial neural network," in *2017 5th International Conference on Cyber and IT Service Management, CITSM 2017*, 2017.
- [15] W. B. Zulfikar, Jumadi, P. K. Prasetyo, and M. A. Ramdhani, "Implementation of Mamdani Fuzzy Method in Employee Promotion System," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 288, no. 1, p. 12147, 2018.
- [16] D. S. A. Maylawati, M. A. Ramdhani, A. Rahman, and W. Darmalaksana, "Incremental technique with set of frequent word item sets for mining large Indonesian text data," in *2017 5th International Conference on Cyber and IT Service Management, CITSM 2017*, 2017.
- [17] A. Taofik, N. Ismail, Y. A. Gerhana, K. Komarujaman, and M. A. Ramdhani, "Design of Smart System to Detect Ripeness of Tomato and Chili with New Approach in Data Acquisition," in *IOP Conference Series: Materials Science and Engineering*, 2018, vol. 288, no. 1, p. 12018.
- [18] H. Mahgoub, D. Rösner, N. Ismail, and F. Torkey, "A Text Mining Technique Using Association Rules Extraction," *Int. J. Comput. Intell.*, vol. 4, no. 1, pp. 21–28, 2008.
- [19] V. Gupta and G. S. Lehal, "A survey of text mining techniques and applications," *Journal of Emerging Technologies in Web Intelligence*, vol. 1, no. 1, pp. 60–76, 2009.
- [20] V. Gupta and G. S. Lehal, "A Survey of Text Summarization Extractive techniques," in *Journal of Emerging Technologies in Web Intelligence*, 2010, vol. 2, no. 3, pp. 258–268.
- [21] C. J. Torre, M. J. Martin-Bautista, D. Sanchez, and I. Blanco, "Text Knowledge Mining: And Approach To Text Mining," *ESTYLF08*, vol. 17–19, 2008.
- [22] A.-H. Tan, "Text Mining: The state of the art and the challenges," in *Proceedings of the PAKDD 1999 Workshop on Knowledge Discovery from Advanced Databases*, 1999, vol. 8, pp. 65–70.
- [23] H. Jiawei, M. Kamber, J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. 2006.
- [24] H. Jiawei, M. Kamber, J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. 2012.
- [25] S. M. Weiss, N. Indurkha, T. Zhang, and F. J. Damerau, "Information Retrieval and Text Mining," *Springer Berlin Heidelberg*, no. Fundamentals of Predictive Text Mining, pp. 75–90, 2010.
- [26] H. M. Wallach, "Topic Modeling: Beyond Bag-of-Words," *ICML*, no. 1, pp. 977–984, 2006.
- [27] A. Sethy and B. Ramabhadran, "Bag-of-word normalized n-gram models," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2008, pp. 1594–1597.
- [28] W. Pu, N. Liu, S. Yan, J. Yan, K. Xie, and Z. Chen, "Local word bag model for text categorization," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2007, pp. 625–630.
- [29] A. Doucet and H. Ahonen-Myka, "An efficient any language approach for the integration of phrases in document retrieval," *Lang. Resour. Eval.*, vol. 44, no. 1–2, pp. 159–180, 2010.
- [30] A. Doucet and H. Ahonen-Myka, "Non-contiguous word sequences for information retrieval," *MWE '04 Proc. Work. Multiword Expressions*, vol. 26, no. July, pp. 88–95, 2004.
- [31] H. Ahonen-Myka, "Discovery of Frequent Word Sequences in Text," *Proc. ESF Explor. Work. Pattern Detect. Discov.*, vol. {LNCS} (24, no. Teollisuuskatu 23, pp. 180–189, 2002.
- [32] H. Ahonen-Myka, "Finding All Maximal Frequent Sequences in Text," *Proc. ICML Work. Mach. Learn. Text Data Anal.*, pp. 11–17, 1999.
- [33] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," *J. Comput. Sci. Technol.*, vol. 15, no. 6, pp. 487–499, 1994.
- [34] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: Current status and future directions," *Data Min. Knowl. Discov.*, vol. 15, no. 1, pp. 55–86, 2007.
- [35] P. Fournier-Viger, J. C. W. Lin, B. Vo, T. T. Chi, J. Zhang, and H. B. Le, "A survey of itemset mining," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 7, no. 4, 2017.
- [36] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and a I. Verkamo, "Fast discovery of association rules," *Advances in knowledge discovery and data mining*, vol. 12, pp. 307–328, 1996.
- [37] F. Kovács and J. Illés, "Frequent itemset mining on hadoop," *Comput. Cybern. (ICCC), 2013 IEEE 9th Int. Conf.*, pp. 241–245, 2013.
- [38] S. Moens, E. Aksehirli, and B. Goethals, "Frequent Itemset Mining for Big Data," in *2013 IEEE International Conference on Big Data*, 2013, pp. 111–118.
- [39] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD '00*, 2000, pp. 1–12.
- [40] S. K. Tanbeer, C. F. Ahmed, B.-S. Jeong, and Y.-K. Lee, "Efficient frequent pattern mining over data streams," in *Proceeding of the 17th ACM conference on Information and knowledge mining - CIKM '08*, 2008, p. 1447.
- [41] S. K. Tanbeer, C. F. Ahmed, B. S. Jeong, and Y. K. Lee, "Efficient single-pass frequent pattern mining using a prefix-tree," *Inf. Sci. (Ny)*, vol. 179, no. 5, pp. 559–583, 2009.