

Deteksi Duplikasi *Metadata File* pada Media Penyimpanan menggunakan Metode *Latent Semantic Analysis*

Erlin¹, Bobby Hashbul Fikri², Susanti³, Triyani Arita Fitri⁴
STMIK Amik Riau

Jl. Purwodadi Indah, Km. 10, Pekanbaru, Indonesia

Email: erlin@sar.ac.id¹, boby@sar.ac.id², susanti@sar.ac.id³, triyani@sar.ac.id⁴

Abstract – Metadata files help a user find relevant information, provide digital identification, archives, and conserves stored files so that they are easily found and reused. A large number of data files on the storage media often makes the user unaware of the duplication and redundancy of the files that have an impact on the waste of storage media space, affecting the speed of a computer in the indexing process, finding or backing up data. This study employs the Latent Semantic Analysis method to extract words contained in the metadata file, presented in vector and matrix form; hence, it can be used to detect file duplication and analyze the metadata of various file types in storage media. The findings showed that the Latent Semantic Analysis method could detect duplicate file metadata in various kinds of storage media, thereby further increasing the usability and speed of access to the data storage media.

Keywords – metadata files, duplication detection, storage media, latent semantic analysis

Intisari – Metadata file membantu pengguna menemukan informasi yang relevan, menyediakan identifikasi digital, mengarsipkan dan melestarikan file yang disimpan sehingga mudah ditemukan dan digunakan kembali. Banyaknya data file pada media penyimpanan seringkali membuat pengguna tidak menyadari adanya duplikasi dan redundansi file yang berdampak pada pemborosan ruang media penyimpanan, mempengaruhi kecepatan sebuah komputer dalam proses pengindeksan, proses pencarian data maupun pencadangan data. Penelitian ini menerapkan metode Latent Semantic Analysis untuk mengekstrak kata-kata yang terdapat pada metadata file, mempresentasikan kedalam bentuk vektor dan matrik sehingga bisa digunakan untuk mendeteksi duplikasi file dan menganalisa metadata dari berbagai jenis file dalam media penyimpanan. Hasil penelitian menunjukkan bahwa metode Latent Semantic Analysis mampu mendeteksi metadata file yang duplikasi dalam berbagai jenis media penyimpanan sehingga lebih meningkatkan daya guna dan kecepatan akses dari media penyimpanan tersebut.

Kata Kunci - metadata file, deteksi duplikasi, media penyimpanan, latent semantic analysis

I. PENDAHULUAN

Data dan media penyimpanan merupakan 2 sisi yang tidak bisa dipisahkan dan memegang peran penting di dunia digital saat ini. Para ahli memperkirakan bahwa lebih dari 2.700.000.000.000.000.000.000 KB (2,7 *zettabytes*) data ada di dunia digital dan sekitar 90% dari jumlah tersebut dihasilkan dalam beberapa tahun terakhir. Manusia menghasilkan sekitar 2.500.000.000.000.000 KB (2,5 *quintillion byte*) data baru setiap hari [1]. Tidak diragukan lagi, angka tersebut akan terus bertambah secara eksponensial dan diperkirakan jumlah data pada tahun 2020 mencapai 44 *zettabytes*, yang berarti lebih banyak media penyimpanan data dibutuhkan di masa mendatang.

Media penyimpanan digunakan untuk menempatkan, menyimpan, dan mengambil data elektronik. Teknologi apapun, termasuk perangkat fisik dan bahan atau komponen dalam sistem komputasi yang menerima dan menyimpan informasi yang berkaitan dengan aplikasi dan pengguna berkaitan dengan media penyimpanan.

Evolusi dari media penyimpanan berkembang dengan sangat pesat. Chowdhury [2] menarasikan bentuk awal media penyimpanan adalah *punch cards* (kertas berlubang). Lubang-lubang di kertas berhubungan dengan penyimpanan sedikit data. *Punch cards* digantikan oleh *punch tape* yang selanjutnya berkembang menjadi *magnetic tape*, *compact cassette*, *magnetic drum*, *disket*, *zip disk*, *hard drive*, *hard disk*, *laser disk*, CD, DVD, HD-DVD, *blu-ray optical disk*, *memory card*, *USB flash drive*, *external hard disk*, SSD (*Solid State Drive*), dan terakhir adalah *Cloud Storage*.

Seringkali tanpa sadar, user menyimpan data file yang sama secara berulang kedalam media penyimpanan. Sangat sedikit user yang menyadari bahwa untuk setiap file yang disimpan ada satu set metadata yang terkait, yang memberikan informasi tentang sumber file, penulis dan detail penting lainnya, tergantung pada jenisnya. Data file yang sama bisa tersimpan pada media yang sama namun berbeda path maupun folder dan juga bisa tersimpan pada media yang berbeda. Hal ini menyebabkan berkurangnya *space* penyimpanan karena terisi oleh metadata file yang sama sekaligus memperlambat proses penyimpanan dan pengambilan data dari media penyimpanan tersebut.

Latent Semantic Analysis (LSA) merupakan sebuah metode dalam pemrosesan bahasa alami (*Natural Language Processing*) khususnya semantik distribusi untuk menganalisis hubungan antara seperangkat dokumen dan istilah-istilah yang dikandungnya dengan menghasilkan serangkaian konsep yang berkaitan dengan dokumen dan istilah tersebut [3]. Sejumlah peneliti [4], [5], [6], [7] telah menggunakan metode LSA untuk berbagai bidang dan telah membuktikan keunggulan dari metode LSA tersebut. Walau telah banyak peneliti membuktikan kehebatan dari metode LSA yang sebagian besar berfokus pada analisis sejumlah corpus data, namun belum ditemukan kajian empiris mengenai penggunaan metode LSA untuk mendeteksi penyimpanan 2 (dua) file yang identik sama pada media penyimpanan yang sama maupun berbeda.

Oleh sebab itu, penelitian ini akan menggunakan metode LSA untuk mendeteksi duplikasi metadata file yang identik sama sehingga bisa meningkatkan ruang/*space* kosong media penyimpanan yang bisa digunakan oleh file berbeda lainnya sehingga mempercepat proses penyimpanan dan pengambilan data kembali dari media tersebut.

II. SIGNIFIKANSI STUDI

A. Konsep Dasar

Latent Semantic Analysis (LSA) adalah metode yang *powerful* menggunakan ilmu statistik dan aljabar linier untuk mengekstrak makna kata-kata dari teks [8]. Teks-teks dapat dibandingkan untuk mengukur seberapa mirip teks tersebut dengan yang lain, dan penilaiannya menyerupai penilaian manusia. LSA mengasumsikan bahwa ada beberapa hubungan antara kata-kata, sehingga secara efektif dan akurat mewakili informasi teks.

Fondasi matematika dari LSA adalah *Vector Space Model* (VSM) untuk merepresentasikan dokumen sebagai vektor dalam ruang dimana istilah kamus digunakan sebagai dimensi. Menggunakan notasi matriks, VSM mewakili kumpulan dokumen (*corpus*) dalam ruang istilah kamus dari matriks X . Dalam LSA, istilah frekuensi matrik X disebut dekomposisi nilai singular (*the singular value decomposition*). *Singular Value Decomposition* (SVD) mendekomposisi X menjadi vektor *eigen term*, U , vektor *eigen dokumen* V , dan nilai singular:

$$X = USV^T \quad (1)$$

Dimana, matrik X adalah matriks $m \times n$, matrik U adalah matrik ortogonal dengan ukuran $m \times r$, matrik S adalah matrik diagonal $r \times r$, matrik V adalah matrik ortogonal dengan ukuran $r \times n$.

Di antara matrik tersebut, r adalah pangkat matrik X , dan $r \leq \min(m, n)$. Setelah dekomposisi nilai singular dari matriks X , berikutnya memilih nilai singular k terbesar dari dekomposisi matrik S , melakukan dimensi untuk matriks ($k \leq r$). Melakukan dekomposisi operasi kolom k dari, U , V dan garis k kolom k dari S , membangun matrik perkiraan k - rank dari X_k .

$$X_k = U_k S_k V_k^T \quad (2)$$

Matriks X_k adalah kuadrat-terkecil dari matriks X asli sehingga jumlah perbedaan kuadrat antara masing-masing elemen dalam X dan X_k diminimalkan. Matriks X_k mengubah *term frequencies* asli dengan mempertimbangkan struktur topik tersembunyi di mana term dan dokumen diproyeksikan [9].

B. Studi Literatur

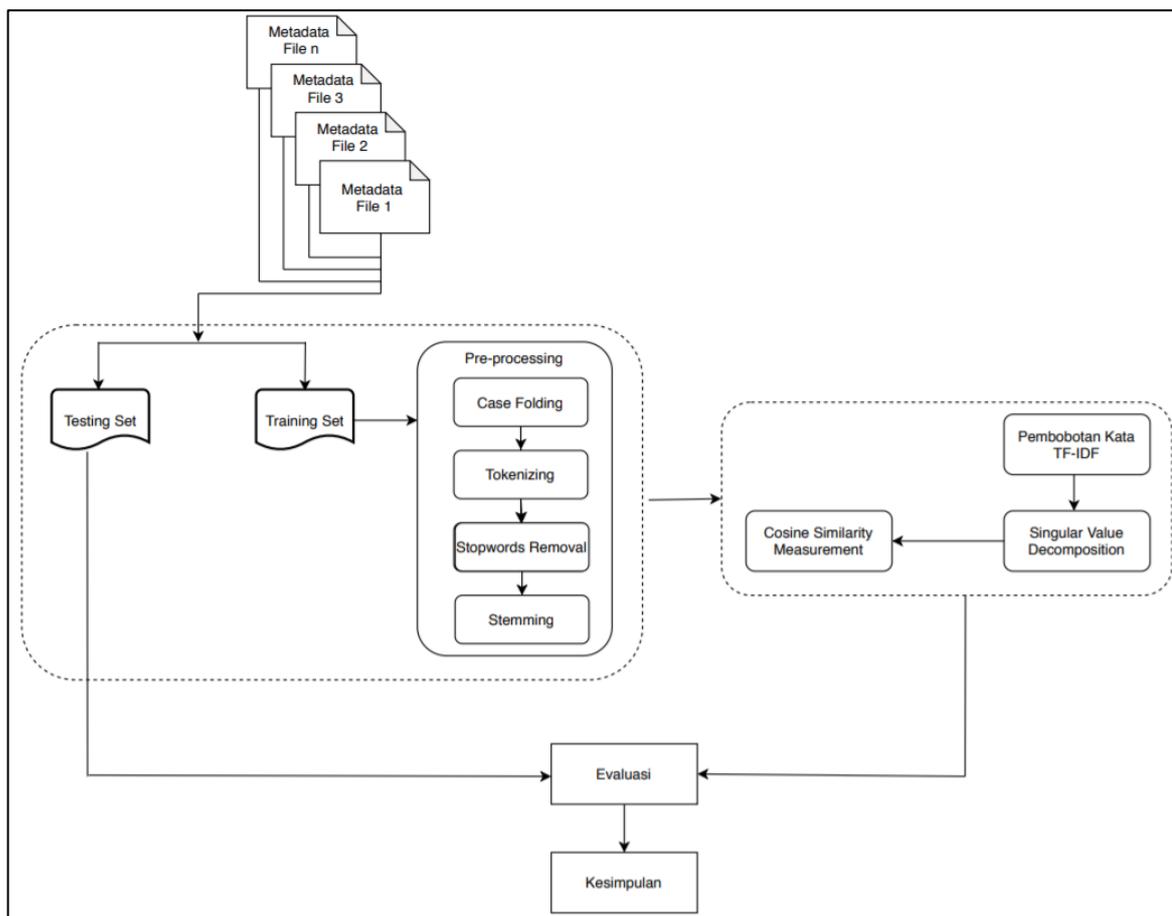
Latent Semantic Analysis diperkenalkan pada akhir tahun 1980-an [10] yang pada tahap awal digunakan untuk meningkatkan sistem pencarian informasi pengindeksan perpustakaan dan kinerja kueri *search engine* [11], [12]. Secara bertahap teknik ini diadopsi oleh para peneliti psikologi dan telah digunakan dalam penelitian kecerdasan buatan, ilmu kognitif, pendidikan, dan sistem informasi [13]. Ide dasar LSA adalah untuk mengekstrak pengetahuan tersembunyi dari serangkaian teks. LSA memproses teks ('dokumen') dari kumpulan file ('corpus') dan mengidentifikasi kata kunci atau istilah. Lebih lanjut, LSA membantu menemukan faktor laten/topik dari istilah yang diekstrak. Fondasi matematika LSA dibangun oleh [14]. Valle-Lisboa dan Mizraji [15] membahas bagaimana LSA menemukan kata-kata laten (istilah) dengan analogi prinsip kerja pikiran manusia. Tinjauan masalah dan teknik solusi dari berbagai tantangan di LSA dilakukan oleh Evangelopoulos et al., [16].

Sejumlah peneliti telah menggunakan metode LSA untuk berbagai bidang. Muller et al., [17] menggunakan LSA untuk memeriksa ribuan koleksi iklan pekerjaan terkait manajemen proses bisnis (MPB). Temuan penelitian mereka menyatakan bahwa LSA mampu mendeteksi penyelarasan individu dan organisasi dengan tipe dan profil ideal yang teridentifikasi menghasilkan keberhasilan kerja yang tinggi dan keberhasilan MPB organisasi. Peneliti lain [18] menggunakan LSA untuk menemukan pasangan objek-set yang sesuai antara gambar dan peta kumpulan data objek poligon. Hasil penelitian menemukan pasangan objek-set yang lebih sesuai dengan akurasi yang sama dalam hal kesamaan bentuk dan informasi bersama dari pasangan yang ditemukan. Penelitian yang mengukur konten emosional dalam ulasan/review suatu produk menggunakan LSA telah dilakukan oleh Ahmad dan Laroche [19]. Penelitian ini berkontribusi pada pemahaman yang lebih baik tentang efek pembingkai emosi diskrit. Selanjutnya Chen et al., [20] mengusulkan pendekatan penggunaan probabilitas LSA berbasis *Transfer Learning* (TL) untuk diagnosis kesalahan *gear* dalam kondisi kerja yang bervariasi yang dapat menyebabkan data pelatihan dan pengujian memiliki distribusi yang berbeda. Hasil eksperimen membuktikan bahwa pendekatan yang diusulkan memiliki keunggulan signifikan dibandingkan diagnosis kesalahan *gear* biasa, terutama dalam berbagai kondisi kerja.

C. Metode Penelitian

Langkah-langkah penyelesaian pada penelitian ini diperlihatkan pada gambar 1 dibawah. Metadata akan diproses melalui tahapan *pre-processing dan filtering*. Pada tahapan ini terdapat 5 (lima) langkah penting yaitu *Cleansing, Case Folding, Tokenizing, Stopword Removal* dan *Stemming*. Hasil dari *pre-processing* akan dilanjutkan dengan tahapan LSA berupa pembobotan term dengan menggunakan perhitungan TF-IDF untuk selanjutnya dilakukan perhitungan *Singular Value Decomposition* dan *Cosine Similarity Measurement*. Hasil LSA akan dievaluasi

dan dibandingkan dengan hasil perhitungan manual yang dilakukan oleh manusia (*human*) dan hasil akhir akan memberikan output dalam bentuk deteksi duplikasi file.



Gambar 1. Tahapan penelitian

Persiapan data yang dilakukan berupa metadata file yang terdiri dari 4 (empat) file dengan berbagai jenis seperti docx, mp3, pdf, jpeg atau jenis file lainnya. Tahapan selanjutnya dilakukan *pre-processing* yang merupakan bagian penting dari setiap algoritma *text mining*. *Case Folding* digunakan untuk mengubah semua huruf besar atau kapital menjadi huruf kecil. Hanya huruf ‘a’ sampai ‘z’ yang diterima. Karakter selain huruf dihilangkan dan dianggap *delimiter*. *Delimiter* adalah urutan satu karakter atau lebih yang dipakai untuk membatasi atau memisahkan *data* yang disajikan dalam *plain Text*.

Selanjutnya *Tokenizing* digunakan untuk memecahkan teks menjadi kata tunggal dan penghapusan tanda baca serta angka. Langkah berikutnya adalah *stopword removal* digunakan untuk menghapus kata-kata umum yang biasanya tidak digunakan karena dianggap kurang penting dan diabaikan untuk proses selanjutnya. Langkah ini melibatkan penghapusan nama, angka, singkatan, bahasa gaul, akronim, dan tanda baca. Kata-kata yang tersisa selanjutnya dilakukan proses *Stemming* untuk menormalkan kata-kata menjadi kata dasar dari kata tersebut seperti “pembatasan”, “batasan”, “batasi”, “keterbatasan” akan dinormalkan kedalam kata dasar menjadi “batas”.

Setelah melalui tahapan *pre-processing*, langkah selanjutnya adalah pemrosesan menggunakan LSA dengan diawali sebuah proses pembobotan terhadap kata-kata dengan cara memberikan nilai pada setiap kata yang muncul menggunakan teknik *term frequency-inverse document frequency* (TF-IDF) yang hasilnya dalam bentuk matrik yang lebih dikenal dengan

term document matrix. Reduksi dimensi matrik perlu dilakukan dengan menggunakan *Singular Value Decomposition* (SVD) yang berfungsi untuk memperkecil nilai kompleksitas dalam pemrosesan *term document matrix*. Hasil dari SVD adalah vektor yang akan digunakan untuk menghitung similaritas dengan pendekatan *cosine similarity* yang akan menghitung nilai kosinus sudut antara vektor dokumen dengan vektor kueri. Besaran sudut yang dihasilkan akan menentukan tingkat similarity dari dokumen yang dibandingkan.

III. HASIL DAN PEMBAHASAN

Pada sebuah media penyimpanan komputer terdapat beberapa file dengan deskripsi sebagai berikut:

File 1 : Ukuran File 2318124 Byte, Lokasi Dan Nama File E:\Hijjaz+selimut+ putih.mp3

File 2 : Ukuran File 3079041 Byte Lokasi Dan Nama File E:\naskah.pdf

File 3 : Ukuran File 3079041 Byte Lokasi Dan Nama File E:\naskah.pdf

File 4 : Ukuran File 3079041 Byte Lokasi Dan Nama File E:\Othernaskah.pdf

Kueri : mp3

Langkah-langkah penyelesaian pendeteksian menggunakan metode *latent semantic analysis* sebagai berikut :

Tahap 1:

Lakukan proses *pre-processing* pada masing-masing *file* dan kueri. Setiap kata akan dijadikan huruf kecil, dan kemudian dilakukan proses penghapusan nama, angka, singkatan, bahasa gaul, akronim dan tanda baca. Berikutnya menjadikan semua kata kedalam bentuk kata dasar.

Tahap 2 :

Susun matriks A dan Q sesuai dengan masing-masing metadata file yang ditemukan dalam semua file (rincian langkah pada Tahap 2a – 2c).

Tahap 2a :

Lakukan perulangan pada masing-masing file dan lakukan pemisahan masing-masing metadata berdasarkan karakter spasi. Kemudian catat semua metadata unik yang belum terdapat pada daftar metadata.

Tahap 2b :

Lakukan pengurutan metadata berdasarkan urutan alfabet. Hal ini hanya dilakukan untuk memudahkan proses pembacaan data.

Tahap 2c :

Lakukan perhitungan pada masing-masing metadata yang ditemukan. Catat jumlah dari masing-masing metadata yang terdapat dalam masing-masing kalimat. Lakukan proses yang sama untuk kueri yang digunakan.

Hasil proses tahap 2 adalah sebuah *Term-Document Matrix* seperti diperlihatkan pada tabel 1.

TABEL I
TERM-DOCUMENT MATRIX

Matrik A	Metadata File				Kueri
	1	2	3	4	
2318124	1	0	0	0	0
2079041	0	1	1	1	0
byte	1	1	1	1	0
dan	1	1	1	1	0
e	1	1	1	1	0
file	2	2	2	2	0
hijjaz	1	0	0	0	0
lokasi	1	1	1	1	0
mp3	1	0	0	0	1
nama	1	1	1	1	0
naskah	0	1	1	0	0
othermaskah	0	0	0	1	0
pdf	0	1	1	1	0
putih	1	0	0	0	0
selimut	1	0	0	0	0
ukuran	1	1	1	1	0

Tahap 3 :

Proses dekomposisi dari *term-document matrix* menjadi 3 (tiga) matrik yaitu matrik *ortogonal* (U), matrik *diagonal* (S) dan *transpose* dari matrik *ortogonal* (V) menggunakan SVD.

Matriks U

0.215193	-0.157259	-0.463331	0.563317
0.068575	-0.360841	0.051547	-0.043949
0.215401	0.346356	-0.017514	0.345204
0.283976	-0.014485	0.034033	-0.062228
0.283976	-0.014485	0.034033	-0.062228
0.567952	-0.028971	0.068066	-0.124457
0.068575	-0.0360841	0.051547	0.025729
0.283976	-0.014485	0.034033	-0.0622228
0.068575	-0.0360841	0.051547	0.025729
0.283975	-0.014485	0.034033	-0.062228
0.146618	0.203582	-0.514847	-0.631442
0.068782	0.142773	0.497364	-0.034062
0.215401	0.346356	-0.017514	0.345204
0.068575	-0.360841	0.051547	0.025729
0.068575	-0.360841	0.051547	0.025729
0.067373	-0.123289	0.492318	-0.031323
0.281231	-0.014444	0.034004	-0.062228

Matriks S

7.039099	0	0	0
0	2.41604	0	0
0	0	1.616739	0
0	0	0	0

Matriks V

0.482708	-0.871807	0.083338	0
0.516028	0.245932	-0.416212	-0.707107
0.516028	0.245932	10.416212	0.707107
0.484171	0.344946	0.804108	0

Tahap 4 :

Aproksimasi derajat 2 dari masing-masing matrik U, S, dan V. Nilai yang disimpan adalah semua baris pada 2 kolom pertama U dan V, dan 2 baris pertama x 2 kolom pertama S. Sehingga menghasilkan matriks Uk, Sk dan Vk.

Matriks Uk

0.215193	-0.157259
0.068575	-.0360841
0.215411	0.346356
0.283976	-0.014485
0.283976	-0.014485
0.283976	-0.014485
0.567952	-0.028971
0.068575	-0.036051
0.287381	-0.013232
0.068111	-0.360831
0.282321	-0.014485
0.146618	0.203581
0.068783	0.148831
0.215231	0.348192
0.068575	-0.368933
0.068575	-0.368933
0.068137	0.142831
0.283976	-0.014485

Matriks Sk

7.039099	0
0	2.41604

Matriks Vk

0.482708	-0.871807
0.516028	0.245932
0.516028	0.245932
0.484171	0.344946

Tahap 5 :

Menghitung nilai matrik Q pada aproksimasi derajat 2 dengan rumus $New Q = QT * Uk * (\frac{1}{Sk})$

Matriks New Q

0.009742	-0.149352
----------	-----------

Tahap 6 :

Hitung nilai similarity masing-masing metadata menggunakan teknik kosinus dan urutkan berdasarkan nilai tertinggi, nilai similarity dihitung dengan rumus: $Sim = (New Q(0) * Vk(0) + New Q(1) * Vk(1)) / (sqrt(New Q(0)^2 + New Q(1)^2) * sqrt(Vk(0)^2 + Vk(1)^2)$

Hasil Akhir perhitungan :

- Peringkat 1 adalah File 1 dengan nilai similarity : 0.904524
- Peringkat 2 adalah File 2 dengan nilai similarity : 0.370553
- Peringkat 3 adalah File 3 dengan nilai similarity : 0.370553
- Peringkat 4 adalah File 4 dengan nilai similarity : 0.526003

Hasil akhir memperlihatkan bahwa tidak ada file yang identik sama dengan kueri yang dicari. Sebaliknya terdeteksi 2 (dua) file yang identik sama yaitu file 2 dan file 3 dengan nilai similarity yang persis sama yaitu 0.370553.

IV. KESIMPULAN

Penelitian ini menguji penggunaan metode Latent Semantic Analysis untuk mendeteksi duplikasi metadata file pada media penyimpanan. Skenario pengujian melalui 6 (enam) tahapan proses sampai memperoleh hasil berupa nilai similarity dari semua file yang diuji. Hasil penelitian membuktikan bahwa metode *Latent Semantic Analysis* mampu mendeteksi file yang identik sama yang terdapat pada media penyimpanan yang sama atau berbeda. Hasil penelitian bisa digunakan untuk meningkatkan efisiensi penggunaan ruang memori serta mempercepat proses akses terhadap data. LSA juga mampu mendeteksi kesamaan beberapa jenis metadata file sehingga lebih meningkatkan akurasi pendeteksian.

REFERENSI

- [1] Lackey, D. *How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read*. Blazon Online. 2019. url: <https://blazon.online/data-marketing/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/>, tanggal akses 1 Mei 2020
- [2] Chowdhury, R. *A Look into the Evolution of Storage Devices [1956-2013]*. Onextrapixel. 2013. url: <https://onextrapixel.com/a-look-into-the-evolution-of-storage-devices-1956-2013/>, tanggal akses 3 Mei 2020
- [3] Sehra, S.S., Singh, J. and Rai, H.S. Using Latent Semantic Analysis to Identify Research Trends in OpenStreetMap. *International Journal of Geo-Information*. 2017; 6, 195: 1-31.
- [4] Ghica, D. R., and Alyahya, K. Latent Semantic Analysis of Game Models Using LSTM. *Journal of Logical and Algebraic Methods in Programming*, 2019; 106, 39–54.
- [5] Sastre, F., Velazquez, A., Sanchez de Leon, L., Montanes, J. L., and Rodrigo, J. Method to Solve Redundant Inverse Problems Based on A Latent Semantic Analysis Approach. Application to An Aerojet Engine. *Aerospace Science and Technology*, 2020;102, 1-10.
- [6] Kim, S., Park, H., and Lee, J. Word2vec-Based Latent Semantic Analysis (W2V-LSA) for Topic Modeling: A Study on Blockchain Technology Trend Analysis. *Expert Systems with Applications*, 2020;113401.
- [7] Santilli, S., Nota, L., & Pilato, G. *The Use of Latent Semantic Analysis in the Positive Psychology: A Comparison with Twitter Posts*. IEEE 11th International Conference on Semantic Computing (ICSC), 2017; 494-498.
- [8] Inrak, P. and Sinthupinyo, S. Applying Latent Semantic Analysis to Classify Emotions in Thai Text. *2nd International Conference on Computer Engineering and Technology*. Chengdu. 2010; 6, 450-454
- [9] Evangelopoulos, N. Latent Semantic Analysis. *WIREs Cognitive Science*. 2013. Vol. 4, John Wiley & Sons. Ltd.
- [10] Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*. 1990;41(6), 391–407
- [11] Cios, K. J., Pedrycz, W., Swiniarski, R. W., and KurganL, A. L. *Data mining: A Knowledge Discovery Approach*. 2007. New York, NY: Springer.
- [12] Manning, C. D., Raghavan, P., and Schütze, H. *An Introduction to Information Retrieval*. 2009. New York, NY: Cambridge University Press.

- [13] Landauer, T. K. LSA as a Theory of Meaning. In T. Landauer & D. S. Mcnamara (Eds.), *Handbook of Latent Semantic Analysis*. 2007
- [14] Martin, D. I., and Berry, M. W. *Mathematical Foundations Behind Latent Semantic Analysis*. In L. Small Bear Technical Consulting (Ed.), 2007. http://dspace.ou.nl/bitstream/1820/966/27/slides_Dian_Martin.pdf.
- [15] Valle-Lisboa, J. C., and Mizraji, E. The Uncovering of Hidden Structures by Latent Semantic Analysis. *Information Sciences*, 2007;177(19), 4122–4147.
- [16] Evangelopoulos, N., Zhang, X., and Prybutok, V. Latent Semantic Analysis: Five Methodological Recommendations. *European Journal of Information Systems*, 21(1), 70–86.
- [17] Oliver Müller, Theresa Schmiedel, Elena Gorbacheva & Jan vom Brocke. Towards A Typology of Business Process Management Professionals: Identifying Patterns of Competences Through Latent Semantic Analysis. *Enterprise Information Systems*, 2014; 50-80.
- [18] Huh, Y., Kim, J., Yu, K., and Cho, S. (2014). M:N Object Matching between Image and Map Object Data Sets by Means of Latent Semantic Analysis. *International Journal of Remote Sensing*, 2014; 35(18), 6799–6814.
- [19] Ahmad, S. N., and Laroche, M. (2015). How Do Expressed Emotions Affect the Helpfulness of a Product Review? Evidence from Reviews Using Latent Semantic Analysis. *International Journal of Electronic Commerce*, 2015; 20(1), 76–111.
- [20] Wenli, C. (2016). *Application Research on Latent Semantic Analysis for Information Retrieval*. Eighth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA).2016; 118-121.