Rancang Bangun Sistem Pencarian Teks dengan Menggunakan Model *Continuous-Bag-of-Words* dan Model *Continuous Skip-Gram* pada Koleksi Dokumen

Muhammad Zalghornain
Program Studi Ilmu Komputer
Universitas Negeri Jakarta
Jakarta, Indonesia
MuhammadZalghornain_3145153000
@mhs.unj.ac.id

Dr. Ria Arafiyah, M.Si
Program Studi Ilmu Komputer
Universitas Negeri Jakarta
Jakarta, Indonesia
riaarafiah@unj.ac.id

Muhammad Eka Suryana, M.Kom
Program Studi Ilmu Komputer
Universitas Negeri Jakarta
Jakarta, Indonesia
eka-suryana@unj.ac.id

***Abstraksi*— Dengan banyaknya data yang dibutuhkan, diperlukan suatu alat untuk mensortir data yang diinginkan secara cepat dan mudah. *Search engine* merupakan salah satu cara untuk mensortir data sesuai keinginan yang dibutuhkan *user* berdasarkan kata yang di-*input user* secara cepat dan mudah. Sistem *search engine* tradisional hanya menggunakan jumlah frekuensi kata pada dokumen untuk mencari kata yang relevan. Diperlukan cara untuk mengerti kueri *user* agar bisa didapatkan hasil pencarian yang sesuai keinginan *user* terlepas keterbatasan kueri *user*. Akan digunakan metode *Continuous-Bag-of-Words* dan *Continuous Skip-Gram* untuk mencari kata di sekitar kata yang di-*input user* yang lalu akan digunakan untuk melakukan pencarian dokumen. Hasil menunjukkan bahwa *neutral network* tidak cocok digunakan untuk mesin pencarian karena waktu *training*-nya yang lama untuk mendapatkan hasil yang diinginkan. Sedangkan untuk hasil relevansinya, metode *Continuous-Bag-of-Words* dapat menghasilkan hasil relevan dengan hasil kesesuaian 86.67% terhadap hasil *ranking* user.**

Keywords*— *Continuous-Bag-of-Words, Continuous-Skip-Gram, pencarian, pencarian teks, ranking dokumen, relevansi dokumen

I. PENDAHULUAN

Web merupakan tempat penyimpanan informasi dalam bentuk teks, gambar, audio dan video. Dengan banyaknya informasi yang ada di web, diperlukan alat untuk mencari dan mendapatkan sesuatu yang diperlukan pengguna dengan mudah.

Information Retrieval (IR) merupakan pencarian material (biasanya dokumen) tak berstruktur (biasanya teks) yang memenuhi kebutuhan informasi dari koleksi besar (biasanya disimpan di komputer) (Manning et al., 2010) [1]. IR dibutuhkan untuk mensortir data yang relevan secara otomatis terhadap *input* yang dimasukkan *user*, sehingga *user* tidak perlu mensortir data sendiri, terutama jika data tersebut ada dalam jumlah yang besar.

Pe-*ranking*-an merupakan hal yang penting dalam *Information Retrieval*, mengembalikan dokumen yang diinginkan *user* merupakan bagian yang penting dalam *search engine*. Sistem pencarian tradisional biasanya menggunakan jumlah frekuensi kata pada dokumen untuk mencari dokumen yang relevan terhadap kueri *user*. Model ini memiliki kelemahan pada keterbatasan kata.

Query Expansion merupakan teknik yang biasa di gunakan pada *Information Retrieval* untuk meningkatkan performa pengambilan data dengan memodifikasi kueri original, dengan menambahkan kata baru atau pembobotan ulang kata original. Menurut Vechtomova and Wang (2006), "*Query Expansion* idealnya harus memiliki beberapa karakteristik, salah satunya merupakan hubungan semantik dengan kueri original" [2]. Penimbangan semantik kata bertujuan untuk mengerti maksud *user* untuk menghasilkan hasil pencarian yang lebih relevan. Dengan mempertimbangkan makna semantik diharapkan dapat meningkatkan kualitas pencarian dan mengurangi waktu *user* untuk memilah-milah dokumen yang relevan.

Word Embedding merupakan representasi kata-kata dalam bentuk vektor, dimana kata yang mirip akan berdekatan pada ruang vektor. *Word Embedding* dapat menangkap makna semantik dari kata. Salah satu cara untuk menghasilkan *Word Embedding* merupakan *Word2vec*. Menurut Al-Saqqa and Awajan (2019), *Word2vec* merupakan salah satu model paling umum yang digunakan dalam *Word Embedding* [3].

Word2vec menggunakan *neural network* yang terdiri dari dua model yaitu *Continuous Bag-of-Words* atau CBOW dan *Continuous Skip-Gram* atau *Skip-Gram*. CBOW menggunakan kata konteks (kata di sekitar kata target) untuk memprediksi kata target, sedangkan *Skip-Gram* menggunakan kata target untuk memprediksi kata di sekitarnya atau kata konteks. Pada penelitiannya, Mikolov et al. (2013) menemukan bahwa CBOW memiliki hasil akurasi sintaksis lebih tinggi daripada *Skip-Gram* terhadap kata yang sering muncul, sedangkan Skip-Gram memiliki akurasi semantik yang lebih tinggi dibanding CBOW [4]. Pendekatan *Word2vec* telah banyak digunakan di berbagai eksperimen dan menjadi pijakan dalam meningkatkan ketertarikan pada *Word Embedding* sebagai teknologi.

Proses pencarian pada *search engine* terdiri dari *crawling*, *indexing*, dan *searching*. Pada penelitian kali ini peneliti akan fokus terhadap bagian *searching*. Peneiltian akan dimulai dengan pengumpulan data. Data yang dikumpulkan akan digunakan untuk melatih sistem dan juga sebagai target pencarian sistem. Sistem akan dilatih untuk menghasilkan data kata relevan dengan model CBOW dan *Skip-Gram*. Setelah sistem selesai di latih, *user* akan melakukan pencarian dokumen pada sistem. *User* akan memasukkan *keyword* pada mesin pencarian. Sistem akan

mencari kata terdekat dari *input* kata *user* dengan menggunakan model CBOW dan *Skip-Gram*. Sistem menggunakan gabungan kata relevan dan kata *input user* untuk mencari dokumen. Dokumen dengan jumlah kemunculan kata *input* dan kata relevan yang tinggi akan dikeluarkan oleh sistem. Lalu di akhir *user* akan menilai relevansi dokumen.

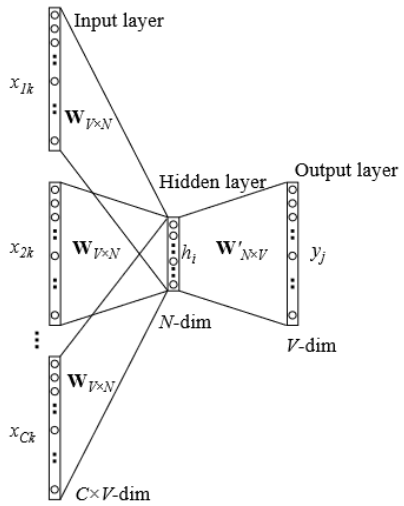
II. KAJIAN PUSTAKA

A. *Word2Vec*

Mikolov et al. (2013), mengajukan dua model baru untuk mempelajari representasi distribusi kata dengan fokus meminimalisasi kompleksitas komputasi [4]. Dari model model sebelumnya, kompleksitas disebabkan oleh *hidden layer* yang tidak linear di model model tersebut. Walaupun ini yang membuat *neural network* menarik, Mikolov memilih untuk menjelajahi model yang lebih sederhana yang mungkin tidak dapat merepresentasikan data seakurat *neural network* tetapi dapat di latih dengan lebih efisien.

1) Continuous Bag-of-Words

Arsitektur ini memprediksi kata target dengan menggunakan kata konteks di sekitarnya. Berikut akan dijelaskan proses dari model CBOW yang dipaparkan oleh Rong (2014) [5].



Gambar 1. Model CBOW

Dalam gambar 1, V menunjukkan ukuran kosakata, dan N menunjukkan ukuran *hidden layer*. *Input* merupakan vektor *one-hot encoded*.

Weight diantara *input layer* dan *output layer* direpresentasikan dengan $V \times N$ matriks W. Untuk mendapatkan *output* pada *hidden layer*, model CBOW mengambil rata-rata vektor dari kata *input* konteks dan mengalikannya dengan *weight matrix input->hidden*.

$$h = \frac{1}{C} W^T (X_1 + X_2 + \ldots + X_C) \quad (1)$$

$$= \frac{1}{C} \left(V_{w_1} + V_{w_2} + \ldots + V_{w_C}\right)^T \quad (2)$$

dimana h merupakan *hidden layer*.
C merupakan jumlah kata konteks.
$w_1, \ldots, w_C$ merupakan kata di konteks.
Vw merupakan vektor *input* dari kata w.
Dari *hidden layer* ke *output layer*, ada *weight matrix* berbeda, W'. Dengan *weight* tersebut kita bisa mendapatkan u_j

$$u_j = {V'_{w_j}}^T h \quad (3)$$

V'_{w_j} merupakan kolom ke j dari matriks W'.
Lalu kita menggunakan *softmax* untuk mendapatkan distribusi posterior kata.

$$p(w_j|w_I) = y_j = \frac{\exp(u_j)}{\sum_{j'=1}^{V} \exp(u_{j'})} \quad (4)$$

dimana y_j merupakan *output* dari j unit di *output layer*.

Update matrix weight hidden->output

Dengan menggunakan *stochastic gradient descent*, kita mendapatkan persamaan *update weight hidden->output*

$$\mathrm{V}'^{(\mathrm{new})}_{\mathrm{w_j}} = \mathrm{V}'^{(\mathrm{old})}_{\mathrm{w_j}} - \eta \cdot \mathrm{e_j} \cdot \mathbf{h}$$

$$\text{for j} = 1, 2, \ldots, \text{V}. \quad (5)$$

$$e_j = y_j - t_j \quad (6)$$

dimana η merupakan *learning rate*.
e_j merupakan *prediction error* kata ke-j dari *output layer*.
h merupakan *hidden layer*.
V'_{w_j} merupakan *output* dari vektor w_j.
V_w *input* vektor dan V'_w *output* vektor, mereka merupakan dua representasi vektor berbeda dari kata w.

Update matrix weight input->hidden

$$V^{(\mathrm{new})}_{w_{I,c}} = V^{(\mathrm{old})}_{w_{I,c}} - \frac{1}{C} \cdot \eta \cdot \mathrm{EH}^T$$

$$\text{for } c = 1,2,\ldots,C. \quad (7)$$

dimana $V_{w_{I,c}}$ merupakan *input* vektor kata ke c di *input* konteks.
η merupakan *learning rate*.
EH sama dengan :

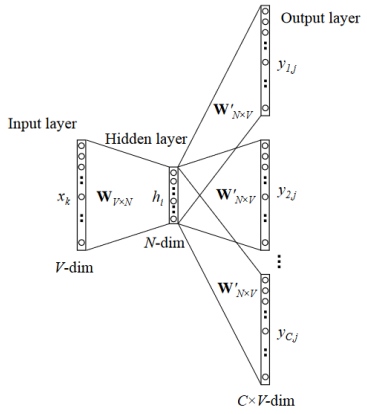
$$\mathrm{EH} = \sum_{j=1}^{V} e_j \cdot w'_{ij}$$

(8)

e_j, sama seperti persamaan (6), merupakan prediksi error kata ke-j di *output layer*.
EH, N-dimensi vektor merupakan jumlah dari *output* vektor dari semua kata di kosakata yang dikalikan *weight error* prediksi.

2) Continuous Skip-Gram

Model ini memprediksi kata konteks berdasarkan kata target yang di masukkan. Berikut akan dijelaskan proses dari model *Skip-Gram* yang dipaparkan oleh Rong (2014) [5].



Gambar 2. Model *Skip-Gram*

Model ini, kebalikan dari CBOW, memiliki kata target di *input layer* dan kata konteks di *output layer*. *Hidden layer* memiliki *output* yang sama dengan CBOW, yang berarti h hanya menyalin (dan transpos) baris dari *weight input->hidden matrix*, W, dengan *input* kata w_i

$$\mathrm{h} = \mathrm{v}_{w_I}^T$$

(9)

untuk *output layer*, karena *Skip-Gram* memiliki satu *input* kata dan konteks kata yang berbeda, maka hasil *output layer*-nya semua sama

$$\mathrm{u}_{c,j} = \mathrm{u}_j = \mathrm{v}_{w_j}^{\mathrm{T}} \cdot \mathbf{h} \qquad \text{for } c = 1, 2, \ldots, C$$

(10)

dimana $u_{c,j}$ merupakan hasil keluaran perkalian antara **h**, *hidden layer* dan $\mathrm{v}_{w_j}^T$.
$\mathrm{v}_{w_j}^T$ merupakan hasil perkalian *input* vektor dan *input weight* matriks ke-j di panel *output layer* ke-c.

Untuk *output layer* :

$$y_{c,j} = \frac{\exp(u_{c,j})}{\sum_{j'=1}^{V} \exp(u_j')}$$

(11)

sama seperti CBOW, *Skip-Gram* menggunakan *softmax* untuk *output layer*.
Dimana $y_{c,j}$ merupakan *output* dari *input* kata ke-j dengan konteks ke-c.
$y_{c,j}$ memiliki hasil *output* yang sama semua karena kata *input*-nya hanya satu. Masing-masing *output* akan dibandingkan dengan vektor konteks kata C yang sebenarnya, yang akan menghasilkan vektor-vektor yang berbeda, yang pada akhirnya akan dijumlahkan untuk meng-*update weight matrix* model ini.

Update matrix weight hidden->output

$$V_{w_j}'^{(new)} = V_{w_j}'^{(old)} - \eta \cdot EI_j \cdot \mathbf{h}$$

$$for\ j = 1, 2, \ldots, V.$$

(12)

Dimana EI merupakan V-dimensional vektor EI = EI_1, ... , EI_V , yang merupakan total dari $e_{c,j}$.
$e_{c,j}$ merupakan *prediction error* semua kata konteks

$$\mathrm{EI}_j = \sum_{c=1}^{C} e_{c,j}$$

(13)

$$e_{c,j} = y_{c,j} - t_{c,j}$$

(14)

$e_{c,j}$ merupakan selisih antara probabilitas prediksi dan *true vector* (vektor yang sebenarnya).
True vector merupakan *one-hot encode* kata konteks ke-c.

Update matrix weight input->hidden

$$V_{w_I}^{(\mathrm{new})} = V_{w_I}^{(\mathrm{old})} - \eta \cdot \mathrm{EH}^T$$

(15)

Dimana EH merupakan N-dimension vektor yang setiap isinya didefinisikan dengan

$$\mathrm{EH}_i = \sum_{j=1}^{V} \mathrm{EI}_j \cdot w'_{ij} \quad (16)$$

Persamaan EI sama dengan persamaan (13).

B. Euclidean Distance

Euclidean distance merupakan rumus untuk mengukur jarak antara dua vektor. Nilai yang semakin kecil berarti vektor semakin dekat. Euclidean distance akan digunakan untuk mengukur jarak antara hasil prediksi dengan vektor yang sebenarnya.

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} \quad (17)$$

dimana p merupakan vektor pertama dan q merupakan vektor kedua

C. Hit Rate

Hit Rate merupakan pembagian dari *user* yang dimana jawaban benarnya termasuk dalam daftar rekomendasi terhadap total *user*. *Hit Rate* akan digunakan untuk mengukur kesesuaian sistem pencarian terhadap *user*

$$HR = \frac{|U_{hit}^{L}|}{|U^{all}|} \quad (18)$$

dimana U_{hit}^{L} merupakan jumlah user yang memberikan jawaban yang benar.

dalam top L rekomendasi.

U^{all} merupakan total *user* dalam tes.

III. METODOLOGI PENELITIAN

A. Metodologi Pengembangan Sistem

Berikut akan dijelaskan proses yang digunakan untuk melakukan pencarian dengan menggunakan Model CBOW dan *Skip-Gram*. Sistem akan dibuat dalam bentuk terminal dengan bahasa pemrograman Phyton versi 3.9.1 dan local web server Apache dengan penyimpanan data dalam bentuk database MySQL, dengan menggunakan aplikasi XAMPP.

Gambar 3. Proses Sistem

Perancangan sistem akan dimulai dengan pengumpulan data. Data akan dikumpulkan dengan *web crawl* lalu akan dirapihkan dengan menghapus hal-hal yang tidak berkaitan dengan artikel, contohnya teks iklan, rekomendasi judul artikel lain, dan yang lainnya. Data yang sudah dikumpulkan dan diraphikan akan disimpan ke dalam *database*. Data yang dikumpulkan merupakan data yang akan dijadikan sebagai target pencarian pada sistem *search engine*.

Selanjutnya *Training* Sistem. Sistem akan dilatih dengan menggunakan kumpulan dokumen yang telah dikumpulkan, dengan metode CBOW dan *Skip-Gram*. Masing-masing pelatihan data dilakukan untuk menemukan kata relevan terhadap kata *input* berdasarkan arsitektur model, CBOW untuk menemukan kata target dari kata *input* sekitar, dan *Skip-Gram* untuk menemukan kata konteks sekitar dari kata *input*. Setelah sistem di latih sistem dapat digunakan untuk pencarian.

Proses selanjutnya *Input* Kata, *user* memasukkan kata *input* pada *search engine* untuk mencari dokumen yang relevan. Kata *input* dibatasi maksimal tiga kata untuk metode *Skip-Gram* dan dua atau empat kata untuk metode CBOW dan metode gabungan *Skip-Gram* dan CBOW. Setelah kata di *input*, sistem akan mencari kata relevan terhadap kata *input* berdasarkan metode yang digunakan.

Kata relevan yang ditemukan sistem akan digunakan untuk pencarian dokumen oleh sistem. Dokumen yang memiliki kata *input* dan kata relevan dengan frekuensi kemunculan tinggi akan diberikan *ranking* tinggi pada *output* pencarian sistem. Lima hasil dokumen dengan *ranking* tertinggi yang dikeluarkan sistem akan dinilai relevansinya oleh *user*.

B. Analisis Data

Tahap pertama dalam rancangan eksperimen adalah pengumpulan dataset. Dataset yang penulis pakai merupakan kumpulan artikel yang didapat dari situs : *https://www.indosport.com*, yang di *crawl* dengan menggunakan *tools* yang dirancang oleh Muhammad Fathan Qoriiba pada skripsinya PERANCANGAN *CRAWLER* SEBAGAI PENDUKUNG PADA *SEARCH ENGINE*.

C. Penggunaan Sistem Untuk Melakukan Pencarian

Setelah proses *training* selesai sistem dapat digunakan untuk melakukan pencarian. *User* akan meng-*input* kata kedalam mesin pencarian dan sistem akan memproses data. Kata yang di-*input user* akan diproses mirip dengan cara *training* sistem tetapi dengan menggunakan *weight matrix input* dan *weight matrix output* yang telah disimpan di *database* setelah proses *training* selesai. Proses akan berjalan hingga ke *output layer* yang hasil di *output layer*-nya akan di cek, dimana bagian vektor yang paling mendekati satu merupakan indeks dari kata yang paling relevan terhadap kata *input*. Hal ini bisa didapatkan dari proses *backpropagation* yang berusaha memperbaiki hasil *output layer* sesuai dengan *true vektor (one-hot encode)* kata di sekitar kata *input*. Akan dicari beberapa kata relevan dari kata *input user* untuk memperluas hasil pencarian dari *search engine*. Di akhir, kata relevan yang ditemukan sistem tersebut akan digabungkan dengan kata *input user* untuk mencari dokumen yang relevan.

D. Pe-ranking-an Dokumen

Pe-*ranking*-an hasil pencarian akan dilakukan dengan mencari jumlah frekuensi kemunculan kata pada dokumen dengan menggunakan kata *input user* dan kata relevan yang telah ditemukan sistem. Pada pencarian dengan menggunakan CBOW, dua kata *input* yang dimasukkan *user* akan di cari kata tengahnya, lalu tiga kesatuan kata tersebut akan digabungkan lalu dicari di semua dokumen. Begitu juga dengan Skip-Gram, kata yang di *input user* akan digabungkan dengan dua kata keluaran sistem lalu akan digabungkan menjadi satu kesatuan lalu di cari di seluruh dokumen. Karena Skip-Gram dan CBOW hanya dapat memprediksi kemungkinan kata yang keluar berdasarkan modelnya tanpa mengetahui tetangga yang mana yang dibagian kiri atau kanan, maka pencarian dimodifikasi dengan mencoba dua kombinasi. Untuk CBOW kata satu dan dua yang di *input user* juga akan dicari dengan cara dibalik. Contohnya kata *input* saya dan bermain dengan kata tengah suka akan dicari saya suka bermain dan bermain suka saya, sedangkan Skip-Gram dua kata keluarannya yang akan dibalik, contohnya kata *input* suka dengan keluaran saya dan bermain akan dicari saya suka bermain dan bermain suka saya.

E. Data Keluaran dan Evaluasi Sistem

Data yang dihasilkan dari mesin pencarian merupakan dokumen yang diharapkan relevan dengan pencarian *user*. Sistem pencarian akan mengeluarkan lima hasil dokumen. *User* akan diminta untuk memberi *ranking* pada lima hasil dokumen tersebut terhadap kesesuaiannya dengan kata *input*. Lalu penulis di akhir akan menilai kesesuaian *ranking* dokumen keluaran sistem terhadap *ranking* dokumen yang diberikan *user*.

F. Rancangan Eksperimen

Akan dilakukan tiga eksperimen dalam penelitian ini :

1. Skenario pertama (Menggunakan metode *Skip-Gram*)
   - *Tester* memasukkan kata *input* yang sama ke dalam *search engine*
   - Sistem mencari kata dengan kemiripan tinggi (kata konteks) dari kata *input user*
   - Sistem melakukan pencarian dokumen terhadap kata *input* dan kata konteks yang terkait
   - Pe-*ranking*-an dilakukan terhadap dokumen dengan menghitung frekuensi kemunculan kata *input* dan kata konteks
   - Sistem mengembalikan *ranking* dokumen berdasarkan hasil tersebut
   - *Tester* menilai relevansi yang dihasilkan oleh sistem

2. Skenario kedua (Menggunakan metode *Continuous Bag-of-Words*)
   - *Tester* memasukkan kata *input* yang sama ke dalam *search engine*
   - Sistem mencari kata dengan kemiripan tinggi (kata target) dari kata *input user*
   - Sistem melakukan pencarian dokumen terhadap kata *input* dan kata target yang terkait
   - Pe-*ranking*-an dilakukan terhadap dokumen dengan menghitung frekuensi kemunculan kata *input* dan kata target
   - Sistem mengembalikan *ranking* dokumen berdasarkan hasil tersebut
   - *Tester* menilai relevansi yang dihasilkan oleh sistem

3. Skenario ketiga (Menggunakan gabungan metode *Skip-Gram* dan *Continous Bag-of-Words*)
   - *Tester* memasukkan kata *input* yang sama ke dalam *search engine*
   - Sistem mencari kata konteks dan kata target dengan kemiripan tinggi dari kata *input user*
   - Sistem melakukan pencarian dokumen terhadap kata *input*, kata target, dan kata konteks terkait
   - Pe-*ranking*-an dilakukan terhadap dokumen dengan menghitung frekuensi kemunculan kata *input* dan kata target
   - Pe-*ranking*-an juga ditemukan dengan menggunakan frekuensi kata *input* dan kata konteks
   - Sistem mengembalikan *ranking* dokumen berdasarkan hasil tersebut
   - *Tester* menilai relevansi yang dihasilkan oleh sistem

IV. HASIL DAN PEMBAHASAN

A. *Data dan Training Parameter*

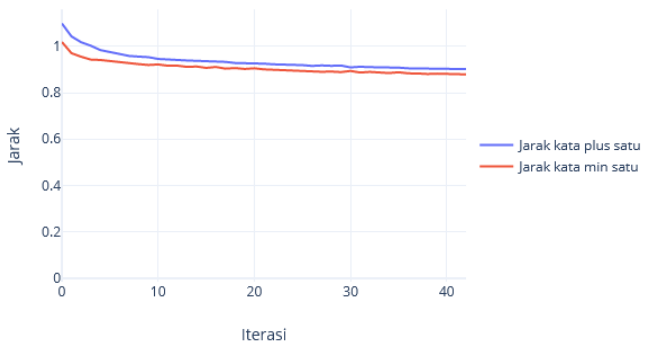
Parameter yang digunakan untuk *training* data merupakan *learning rate* sebesar 0.05 dan 400 *hidden layer* (400 dimensi vektor). Dikarenakan lamanya waktu *training* data, penulis memilih untuk menghapus *stopwords* yang ada dalam teks. Daftar kata *stopwords* yang dihapus didapat dari "*A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*", Tala (2003) [6]. Kata yang hanya berisi angka juga dihapus karena dianggap tidak memiliki makna (contohnya kata "10:43").

B. *Analisis Hasil*

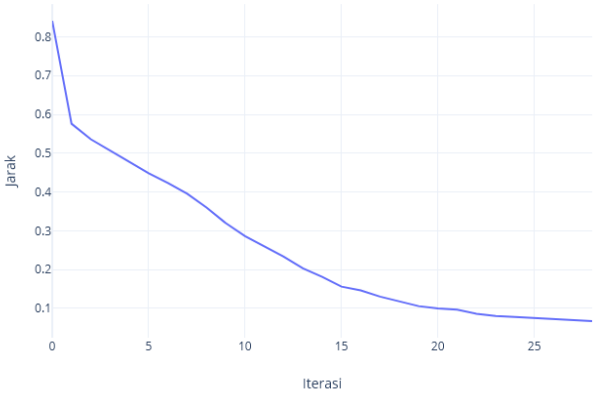
Bagian ini akan membahas hasil dari tiga eksperimen yang telah dilakukan. Hasil skenario percobaan yang pertama, pada saat pelatihan proses *training Skip-Gram*, setelah tiga jam *training*, hasil yang di dapat terlalu jauh dengan hasil yang seharusnya. Hasil yang di dapat merupakan jarak antara hasil prediksi dan hasil yang sebenarnya yang di hitung dengan menggunakan *Euclidean Distance*. Jarak akhir yang di dapat merupakan 0.87 untuk tetangga kiri dan 0.9 untuk tetangga kanan setelah 43 iterasi selama tiga jam. Berdasarkan landainya grafik dan hasil jarak pada *Skip-Gram* yang dianggap terlalu jauh, maka sistem dianggap tidak cocok untuk skenario pengetesan.

Untuk skenario CBOW, *training* sistem *input* dua kata dengan waktu tiga jam menghasilkan hasil jarak 0.1 pada iterasi 50, sedangkan *training* sistem *input* empat kata menghasilkan jarak akhir 0.067 dengan total 29 iterasi. Dengan hasil tersebut sistem dianggap layak untuk di tes ke *user*.

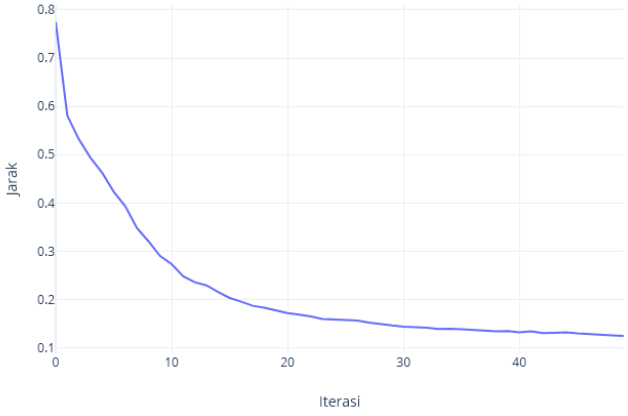
.



Gambar 4. Jarak Antara Prediksi dengan Hasil Sebenarnya *Skip-Gram.*



Gambar 5. Jarak Antara Prediksi dengan Hasil Sebenarnya CBOW Dua Kata.



Gambar 6. Jarak Antara Prediksi dengan Hasil Sebenarnya CBOW Empat Kata.

Untuk pengetesan relevansi pencarian terhadap *user* dengan menggunakan metode CBOW, *user* akan diberikan lima pasang kata untuk metode CBOW dua kata, dan lima dua pasang kata untuk metode CBOW empat kata. *User* akan diminta untuk memasukkan kata tersebut ke dalam mesin pencarian. Mesin pencarian akan mengeluarkan lima hasil dokumen lalu *user* diminta untuk menilai *ranking* dari dokumen keluaran sistem tersebut. Untuk hasil pencarian teks yang hanya menghasilkan jumlah dokumen yang kurang dari lima, sisa data akan diisi dengan dokumen random untuk dinilai oleh *user*, dan untuk dokumen yang memiliki jumlah hasil kemunculan kata yang sama, dokumen akan diberikan *ranking* yang sama.

Berikut akan dipaparkan hasil pencarian dengan menggunakan metode CBOW dua kata.

Tabel 1. Pencarian Pertama CBOW dengan Menggunakan Kata "hasil" dan "italia"

Dokumen	Ranking				Kesesuaian
	Sistem	User 1	User 2	User 3	
https://www.indosport.com/tag/13322/lautaro-martinez	1 / 2	1	4	2	66.67%
https://www.indosport.com/sepakbola/20221110/hasil-liga-italia-inter-milan-vs-bologna-sempat-tertinggal-nerazzurri-pesta-gol	1 / 2	2	1	1	100%
https://www.indosport.com/sportainment/20200324/persija-libur-ini-yang-dilakukan-marko-simic-dan-manohara	x	3	2	5	x
https://www.indosport.com/tag/5059/kenny-sansom	x	4	3	4	x
https://www.indosport.com/multisport/20190930/naik-ring-november-2019-daud-yordan-jalani-latihan-khusus-di-bali	x	5	5	3	x

Pencarian pertama dengan kata hasil dan italia menghasilkan kata tengah liga, dan kata "hasil liga italia" ditemukan sekali pada dua *ranking* teratas dengan nilai yang sama, yaitu masing-masing dokumen ditemukan kata tersebut hanya sekali. Sedangkan tiga *ranking* dibawahnya merupakan data random karena tidak ditemukan kata tersebut pada dokumen lain. Karena hasil pertama dan hasil kedua ditemukan dengan nilai yang sama, maka hasil pertama atau kedua diberikan *ranking* 1 atau 2. Jika *user* menilai *ranking* satu atau dua pada dokumen tersebut maka hasil akan dianggap sesuai.

Pada hasil kueri pertama, didapatkan dua *hit* dari total tiga user, maka didapatkan *hit rate* sebesar $\frac{2}{3}$, yang menghasilkan kesesuaian 66.67%. Hasil kesesuaian rata-rata yang di dapat dari semua kueri merupakan 83.34%

Berikut merupakan hasil rata-rata kesesuaian dari semua pencarian.

Tabel 2. Tabel Rata-Rata Kesesuaian Pencarian dengan Metode CBOW Dua Kata

Pencarian	Kesesuaian
Pencarian 1	83.34%
Pencarian 2	100%
Pencarian 3	66.67%
Pencarian 4	100%
Pencarian 5	83.34%

Dengan itu dari lima pertanyaan total yang diberikan, hasil kesesuaian pe-*ranking*-an sistem pencarian dengan menggunakan metode CBOW dua kata terhadap pe-*ranking*-an *user* merupakan 86.67%.

Berikut merupakan hasil pencarian dengan menggunakan metode CBOW empat kata.

Tabel 3. Pencarian Pertama CBOW dengan Menggunakan Kata "pemain", "kunci", "juara", dan "piala"

Dokumen	Ranking				Kesesuaian
	Sistem	User 1	User 2	User 3	
https://www.indosport.com/tag/13322/lautaro-martinez	1 / 2	3	3	2	33.33%
https://www.indosport.com/sepakbola/20221117/termasuk-lionel-messi-dan-lisandro-martinez-ini-5-pemain-kunci-argentina-juara-piala-dunia-2022	1 / 2	1	1	1	100%
http://www.indosport.com/raket/20220502/	x	2	2	3	x

Dokumen	Ranking				Kesesuaian
	Sistem	User 1	User 2	User 3	
sederhana-namun-tetap-khidmat-begini-suasana-perayaan-idulfitri-tim-bulutangkis-indonesia-di-manila					
https://www.indosport.com/otomotif/20181031/muncul-bunyi-berisik-ini-solusi-sempurna-untuk-cvt-motor-matik	x	5	5	4	x
https://www.indosport.com/multi-event/20220506/breaking-news-asian-games-hangzhou-2022-resmi-ditunda-gara-gara-covid-19	x	4	4	5	x

Pencarian kata pertama menggunakan kata pemain, kunci, juara dan piala, menghasilkan kata tengah argentina. Pencarian dengan kata "pemain kunci argentina juara piala" menghasilkan jumlah temu satu kali pada dokumen pertama dan kedua, dan tidak ditemukan pada dokumen lainnya. Hasil rata-rata kesesuaian merupakan 66.67%

Berikut merupakan hasil rata-rata kesesuaian dari lima pencarian.

Tabel 4. Tabel Rata-Rata Kesesuaian Pencarian dengan Metode CBOW Empat Kata

Pencarian	Kesesuaian
Pencarian 1	66.67%
Pencarian 2	100%
Pencarian 3	100%
Pencarian 4	100%
Pencarian 5	66.67%

Dengan itu rata-rata kemunculan pada sistem CBOW empat kata merupakan 86.67%. Kebanyakan pencarian dari CBOW empat kata hanya menghasilkan satu dokumen karena kata yang dicari perlu banyak.

V. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Berdasarkan tiga eksperimen yang telah dikerjakan, kesimpulan dari analisis hasil, implementasi, dan evaluasi program dipaparkan sebagai berikut.

1. Hasil eksperimen pertama dianggap gagal, dikarenakan hasil akhir jarak antara vektor prediksi dan vektor kata yang sebenarnya merupakan 0.87 untuk tetangga kiri dan 0.9 untuk tetangga kanan, yang dianggap tidak layak untuk dijadikan sistem pencarian.
2. Hasil eksperimen kedua menunjukkan bahwa *training* sistem berhasil, yang ditunjukkan oleh hasil jarak 0.1 pada CBOW dua kata dan 0.067 pada CBOW empat kata setelah tiga jam training. Lalu berdasarkan pencarian dengan penggabungan kata *input* dan kata keluaran sistem, sistem menghasilkan rata-rata kesesuaian 86.67% untuk kedua sistem, CBOW dua kata dan CBOW empat kata. Dengan ini dapat disimpulkan bahwa pencarian menggunakan metode *Continuous-Bag-of-Words* dapat menghasilkan pencarian yang relevan.
3. Hasil eksperimen ketiga dianggap gagal karena sistem *Skip-Gram* tidak dapat digunakan.
4. Dikarenakan *training* waktu yang lama (tiga jam untuk mencapai dekat dengan nol pada CBOW dengan total data 38925 kata, 70 artikel), maka *neural network* ini dapat dianggap tidak bagus untuk di implementasikan pada mesin pencarian, apalagi ditambah dengan banyaknya data dokumen yang harus di proses dalam kumpulan data yang selalu mengekspansi.

B. *Saran*

Berdasarkan kesimpulan di atas, saran yang dapat diberikan penulis untuk penelitian selanjutnya yaitu sebagai berikut.

1. Menggunakan metode lain untuk mesin pencarian.
2. Menambahkan cara untuk membedakan *ranking* dari hasil pencarian dengan jumlah kata temu yang sama.
3. Merapihkan data *crawl* lebih baik lagi, karena terlihat dari hasil pencarian, data yang di *crawl* masih ada kalimat dan kata yang di proses yang tidak merepresentasikan judul artikel (contohnya judul rekomendasi *link* ke artikel lain).
4. Menggunakan gabungan kata *input* dan kata mirip dengan banyak variasi untuk ekspansi percobaan pencarian.
5. Menggunakan banyak kategori tema untuk melihat efeknya pada hasil pencarian.

DAFTAR PUSTAKA

[1] C. Manning, P. Raghavan and H. Schütze, "Introduction to information retrieval," *Natural Language Engineering*, vol. 16, p. 100–103, 2010.

[2] O. Vechtomova and Y. Wang, "A study of the effect of term proximity on query expansion," *Journal of Information Science*, vol. 32, pp. 324-333, 2006.

[3] S. Al-Saqqa and A. Awajan, "The Use of Word2vec Model in Sentiment Analysis: A Survey," in *Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control*, New York, NY, USA, 2019.

[4] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *Proceedings of Workshop at ICLR*, vol. 2013, January 2013.

[5] X. Rong, *word2vec Parameter Learning Explained*, arXiv, 2014.

[6] F. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," December 2003.