# SKRIPSI

# SISTEM REKOMENDASI SALURAN YOUTUBE EDUKASI SECARA SEMANTIK MENGGUNAKAN NEURAL NETWORK WORD EMBEDDINGS

Disusun oleh :
**Ahmad Farhan Setiawan**
**(065118240)**

**PROGRAM STUDI ILMU KOMPUTER**
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**
**UNIVERSITAS PAKUAN**
**BOGOR**
**2023**

# SKRIPSI

# SISTEM REKOMENDASI SALURAN YOUTUBE EDUKASI SECARA SEMANTIK MENGGUNAKAN NEURAL NETWORK WORD EMBEDDINGS

Diajukan sebagai salah satu syarat untuk memperoleh Gelar Sarjana Komputer Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam


Disusun oleh :
**Ahmad Farhan Setiawan**
**(065118240)**




**PROGRAM STUDI ILMU KOMPUTER**
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**
**UNIVERSITAS PAKUAN**
**BOGOR**

**2023**

# HALAMAN PENGESAHAN

**Judul :** Sistem Rekomendasi Saluran Youtube Edukasi Secara Semantik Menggunakan Neural Network Word Embeddings
**Nama :** Ahmad Farhan Setiawan
**NPM :** 065118240

Mengesahkan,

| Pembimbing Pendamping | Pembimbing Utama |
|---|---|
| FMIPA - UNPAK | FMIPA - UNPAK |
|  |  |
| Siska Andriani, M.Kom. | Dr. Prihastuti Harsani, M.Si. |

Mengetahui,

| Ketua Program Studi Ilmu Komputer | Dekan |
|---|---|
| FMIPA - UNPAK | FMIPA - UNPAK |
|  |  |
| Arie Qur'ania, M.Kom. | Asep Denih, S.Kom., M.Sc., Ph.D. |

# PERNYATAAN KEASLIAN KARYA TULIS SKRIPSI

Dengan ini saya menyatakan bahwa:

Sejauh yang saya ketahui, karya tulis ini bukan merupakan karya tulis yang pernah dipublikasikan atau sudah pernah dipakai untuk mendapatkan gelar sarjana di Universitas lain, kecuali pada bagian-bagian di mana sumber informasinya dicantumkan dengan cara referensi yang semestinya.

Demikian pernyataan ini saya buat dengan sebenar-benarnya. Apabila kelak dikemudian hari terdapat gugatan, penulis bersedia dikenakan sanksi sesuai dengan peraturan yang berlaku

Bogor, Januari 2023

(Ahmad Farhan Setiawan)

**PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA**

==========

Saya yang bertandatangan di bawah ini :

| | |
|---|---|
| Nama | : Ahmad Farhan Setiawan |
| NPM | : 065118240 |
| Judul Skripsi | : Sistem Rekomendasi Saluran Youtube Edukasi Secara Semantik Menggunakan Neural Network Word Embeddings |

Dengan ini saya menyatakan bahwa Paten dan Hak Cipta dari produk Skripsi dan Tugas Akhir di atas adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apapun kepada perguruan tinggi manapun.

Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan Paten, hak cipta dari karya tulis saya kepada Universitas Pakuan.

Bogor, Januari 2023

Ahmad Farhan Setiawan
065118240

# RIWAYAT HIDUP

Penulis dilahirkan di Bogor pada tanggal 5 Mei 2001 dari pasangan Bapak Mansyur Nurhaman dan Ibu Pupu Marpu'ah sebagai anak ketiga dari tiga bersaudara.

Penulis memulai pendidikan di Sekolah Dasar yang bertempat di SDN Siliwangi, kemudian pada tahun 2012 masuk SMP Negeri 9 Bogor dan Penulis adalah Alumni dari SMA Al-Ghazaly Bogor.

Pada tahun 2018 penulis meneruskan pendidikan ke Universitas Pakuan Bogor, Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam. Pada bulan Desember tahun 2022 penulis menyelesaikan penelitian dengan judul Sistem Rekomendasi Saluran Youtube Edukasi Secara Semantik Menggunakan *Neural Network Word Embeddings*.

# RINGKASAN

Ahmad Farhan Setiawan. 2022. Sistem Rekomendasi Saluran Youtube Edukasi Secara Semantik Menggunakan *Neural Network Word Embeddings* dibawah bimbingan Dr. Prihastuti Harsani, M.Si. dan Siska Andriani, M.Kom.

Sistem Rekomendasi Saluran Youtube Edukasi Secara Semantik Menggunakan *Neural Network Word Embeddings* merupakan sistem yang dapat merekomendasikan beberapa saluran Youtube edukasi berdasarkan nama saluran lain ataupun kata kunci yang diinputkan pengguna secara semantik. Sistem ini dapat merekomendasikan data serupa tidak hanya berdasarkan kemunculan kata pada atribut di masing-masing data, namun juga dapat merekomendasikan data melalui kesamaan semantik dari masing-masing kata yang ada pada atribut di masing-masing data berkat metode *Neural Network Word Embeddings*.

Diharapkan penelitian ini dapat memberikan manfaat, pengetahuan dan ilmu baru bagi para pembaca. Selain itu, penelitian ini juga diharapkan dapat membantu para pengguna Youtube yang menjadikan Youtube sebagai media pembelajaran dalam mencari saluran-saluran Youtube edukasi lain yang belum diketahui sebelumnya agar dapat mendapatkan ilmu dan pengetahuan baru dari saluran lainnya.

# KATA PENGANTAR

Assalamualaikum Wr Wb.,

Puji dan syukur saya panjatkan kehadirat Allah SWT., karena rahmat dan hidayahnya, saya dapat menyelesaikan laporan skripsi yang berjudul "Sistem Rekomendasi Saluran Youtube Edukasi Secara Semantik Menggunakan Neural Network Word Embeddings".

Atas banyaknya dukungan moril dan materil yang diberikan dalam penyusunan skripsi ini, maka saya mengucapkan banyak-banyak terima kasih kepada:

1. Dr. Prihastuti Harsani, M.Si. selaku dosen pembimbing utama yang telah memberikan bantuan berupa bimbingan, saran, ide yang sangat berperan penting dalam penyelesaian laporan skripsi ini.
2. Siska Andriani, M.Kom. selaku dosen pembimbing pendamping yang telah memberi bantuan berupa bimbingan, saran, ide yang sangat berperan penting dalam penyelesaian laporan skripsi ini.
3. Kedua orang tua yang selalu memberikan semangat yang tiada henti-hentinya untuk dapat menyelesaikan skripsi ini.
4. Teman-teman yang telah mendukung dan membantu saya dalam proses pembuatan laporan skripsi.

Saya menyadari bahwa laporan skripsi ini belum sempurna. Oleh karena itu, segala kritik dan saran yang membangun akan diterima dengan senang hati. Mudah-mudahan Allah SWT akan membalas semua kebaikan kepada semua pihak yang membantu. Akhir kata, semoga laporan ini dapat bermanfaat bagi kita semua. Aamiin. Wassalamualaikum Wr Wb.

Bogor, Januari 2023

Ahmad Farhan Setiawan

# DAFTAR ISI

# DAFTAR GAMBAR

# DAFTAR TABEL

# DAFTAR LAMPIRAN

# BAB I
# PENDAHULUAN

## 1.1. Latar Belakang Masalah

Youtube, merupakan sarana berbagi video online dan media sosial yang didirikan oleh Steve Chan, Chad Hurley, dan Jawed Karim pada bulan Februari tahun 2005. Menurut data yang dikeluarkan pada bulan Mei 2021, saat ini, Youtube merupakan situs web terpopuler nomor dua setelah Google dengan lebih dari satu miliar pengguna di setiap bulannya (Semrush, 2021). Dengan Youtube, pengguna dapat melihat, mengunggah, serta membagikan video dengan berbagai macam tema serta kategori. Beberapa kategori diantaranya adalah video blog (*vlog*), serial televisi, musik, cuplikan film, video instruksi, bahkan edukasi.

Youtube juga dikenal sebagai sarana edukasi yang dapat memberi banyak pengetahuan kepada para pengguna. Pandemi Covid-19 yang mulai mengguncang dunia pada tahun 2020 silam mengharuskan orang-orang untuk melakukan aktivitasnya di dalam rumah, termasuk aktivitas belajar mengajar. Karena hal itu, banyak orang yang menjadikan Youtube sebagai salah satu sumber pembelajaran mereka. *Official Youtube*, melalui *Youtube Creator Academy* telah merilis pernyataan bahwa konten edukasi adalah satu dari fokus utama yang perlu dikembangkan secara serius, "Setiap hari, lebih dari ratusan juta penonton mencari dan menonton berbagai tayangan edukasi melalui aplikasi Youtube" (Youtube Creator Academy, 2020).

Terdapat beberapa permasalahan yang didapat pengguna saat ingin mendapatkan edukasi dari Youtube, salah satunya adalah pengguna seringkali mengalami kesulitan pada saat ingin mencari saluran-saluran edukasi yang ingin dicari. Terkadang, pengguna hanya akan mendapati video-video yang sama ataupun saluran-saluran yang sudah memiliki nama besar saja, hal ini tentunya akan menyulitkan pengguna ketika ingin mencari saluran kecil yang memiliki konten edukasi yang juga dapat memberi banyak manfaat. Permasalahan lainnya yaitu pengguna juga sering mengalami kebingungan untuk menonton saluran Youtube edukasi mana yang ingin ditonton karena banyaknya jumlah saluran edukasi yang ada. Pengguna juga sering mengalami kesulitan untuk mendapatkan nama saluran lain yang mungkin memiliki konten yang serupa dengan saluran edukasi favoritnya.

Youtube sendiri sebenarnya sudah memiliki sistem rekomendasinya sendiri, hanya saja, sistem rekomendasi pada aplikasi Youtube hanya memberikan pengguna rekomendasi mengenai video-video berdasarkan riwayat video yang disukai oleh pengguna, bukan merekomendasikan saluran-saluran serupa yang mungkin akan disukai pengguna. Kekurangan lain yang terdapat pada sistem rekomendasi milik Youtube adalah, pengguna tidak dapat mencari daftar saluran-saluran Youtube edukasi sesuai dengan kata kunci yang ingin dicari, melainkan Youtube hanya akan sebisa mungkin merekomendasikan saluran Youtube yang memiliki nama dengan padanan kata seperti pada kata kunci yang dimasukkan oleh pengguna di kolom pencarian, bukan menyediakan saluran-saluran Youtube yang sering mengunggah konten yang berkaitan dengan kata kunci. Selain itu, Youtube juga tidak memiliki fitur untuk mencari daftar saluran yang memiliki konten serupa dengan saluran-saluran favorit pengguna.

Penelitian serupa mengenai sistem rekomendasi menggunakan *neural network word embeddings* (word2vec) telah beberapa kali dilakukan sebelumnya. Salah satunya dilakukan oleh Aldiansyah (2019) yang mengembangkan sistem

rekomendasi lagu *cross language* berdasarkan lirik menggunakan *word2vec*. Penelitian selanjutnya dilakukan oleh Laili (2019) yang mengembangkan sistem rekomendasi film berdasarkan sinopsis menggunakan metode *word2vec*. Sistem-sistem yang telah dikembangkan sebelumnya tersebut cukup berhasil merekomendasikan beberapa data relevan secara semantik kepada pengguna, namun sistem-sistem rekomendasi tersebut masih memiliki nilai *precision* yang kurang memuaskan.

Berdasarkan uraian tersebut, dibutuhkan sebuah sistem rekomendasi yang dapat merekomendasikan pengguna Youtube beberapa saluran edukasi yang sering mengunggah konten berdasarkan kata kunci secara semantik yang diberikan pengguna yang juga mengandung kesamaan makna dengan kata kunci, sehingga pengguna dapat dengan mudah mendapatkan referensi baru mengenai saluran yang dapat dinikmati sesuai dengan kata kunci yang diminati oleh pengguna meskipun saluran tersebut tidak memiliki atribut yang mengandung kata kunci tersebut. Dibutuhkan juga sistem rekomendasi yang dapat merekomendasikan daftar beberapa saluran yang memiliki tingkat kemiripan paling tinggi dengan saluran yang diinputkan oleh pengguna yang dapat menangkap kesamaan makna walaupun masing-masing atributnya mengandung kata-kata yang berbeda satu sama lain.

Untuk dapat menangkap kesamaan makna semantik dari masing-masing kata, dibutuhkan sebuah metode terkini dan terbarukan yang mampu menangkap kesamaan makna kata secara semantik dengan sangat baik. Dengan menggunakan metode *neural network word embeddings*, metode tersebut merupakan metode yang dapat merepresentasikan kata-kata menjadi vektor berdimensi tinggi yang juga dapat menangkap konteks dan makna semantik dari masing-masing kata yang dapat diukur kemiripannya dengan perhitungan *cosine similarity* (Perone, C. 2018). Dengan mengukur kemiripan dari setiap saluran, sistem dapat memberikan skor yang menjadi acuan dalam memberikan rekomendasi sesuai dengan deskripsi, judul video, dan kata kunci dari setiap saluran. Dengan sistem rekomendasi ini, penikmat konten edukasi Youtube diharapkan dapat menambah referensi baru mengenai saluran-saluran edukasi yang ada di Youtube yang sebelumnya tidak diketahui oleh pengguna sehingga akan mendatangkan banyak manfaat, baik untuk pemilik saluran, maupun untuk penikmat konten edukasi di Youtube.

### 1.2. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut :

a. Membangun sebuah sistem rekomendasi yang dapat merekomendasikan beberapa saluran Youtube edukasi sesuai dengan kata kunci yang diinputkan oleh pengguna ataupun berdasarkan saluran lain yang memiliki konten serupa berdasarkan kesamaan makna satu sama lain dari masing-masing atribut.

### 1.3. Ruang Lingkup

Ruang lingkup dari penelitian adalah sebagai berikut :

a. Dataset yang digunakan pada sistem rekomendasi ini adalah data-data saluran Youtube yang sering mengunggah konten-konten edukasi yang didapat dari hasil *scraping*.
b. Saluran-saluran Youtube yang terdapat pada data adalah saluran Youtube yang mengunggah konten-konten edukasi dengan bahasa Indonesia dan bahasa Inggris.

c. Saluran-saluran Youtube yang digunakan adalah saluran-saluran Youtube yang memiliki kategori *Education, How To & Style, Pets & Animals,* dan *Science & Technology.*
d. Data berisi saluran-saluran Youtube dengan atribut berupa nama saluran, jumlah *subscriber,* jumlah *view,* deskripsi saluran, negara, *keywords* saluran, dan sampel judul video.
e. Sistem dikembangkan dengan menggunakan bahasa pemrograman Python dan beberapa kerangka kerja didalamnya.
f. Kata-kata yang akan menjadi kamus kata (*corpus*) untuk dilatih pada proses *neural network word embeddings* adalah gabungan kata-kata yang terdapat pada atribut deskripsi saluran, *keywords* saluran, dan sampel judul video
g. Proses evaluasi dilakukan dua kali, evaluasi pertama adalah mengevaluasi model *word2vec* dengan menggunakan *pearson correlation*, dan evaluasi kedua adalah mengevaluasi sistem rekomendasi dengan menggunakan *precision*.
h. Tahap evaluasi sistem dilakukan dengan dua kriteria berbeda, kriteria pertama adalah pengujian pada kata-kata di bidang edukasi, dan kriteria kedua adalah pengujian pada kata-kata yang tidak berhubungan dengan edukasi.
i. Perhitungan kemiripan antara nilai *word embeddings* dari masing-masing data dihitung menggunakan *cosine similarity.*
j. Hasil penelitian berupa sebuah sistem rekomendasi berbasis website.
k. Sistem dapat menerima input berupa kata kunci ataupun nama saluran Youtube edukasi lain yang tersedia pada dataset sistem.
l. Sistem juga memiliki fitur untuk eksplorasi model *word embeddings* yang digunakan, seperti untuk mencari kesamaan antar kata, mencari kata terdekat, dan melakukan perhitungan aritmatika dari kata yang dimasukkan.

### 1.4. Manfaat Penelitian

Manfaat dari penelitian ini diantaranya adalah :

a. Memudahkan pengguna Youtube saat ingin mencari saluran berkonten edukasi yang sesuai dengan preferensi pengguna.
b. Meningkatkan minat belajar masyarakat melalui konten-konten edukasi yang diunggah oleh para saluran Youtube.
c. Menambah cakupan ilmu dan referensi baru kepada para pengguna Youtube mengenai saluran-saluran Youtube edukasi yang ada di platform Youtube.
d. Menambah kemampuan dan pengetahuan peneliti mengenai *Natural Language Processing* dan *Word Embeddings*.

# BAB II
# TINJAUAN PUSTAKA

## 2.1. Sistem Rekomendasi

Sistem rekomendasi adalah sebuah turunan spesifik dari sistem penunjang keputusan yang membantu pengguna untuk mendapatkan objek yang menarik atau berguna dari sebuah pilihan dokumen atau data berjumlah banyak yang dapat menghasilkan output berupa data-data relevan. Sistem rekomendasi dibutuhkan ketika seseorang atau sekelompok orang memiliki gagasan tentang sebuah keadaan yang diinginkan, namun kesulitan untuk memutuskan. Sistem rekomendasi dapat memberikan bantuan dalam konteks ini dengan mencoba menemukan item-item yang sesuai yang dapat membantu untuk mencapai target yang diinginkan (Felfernig, Boratto, Stettinger & Tkalcic, 2018). Proses penyaringan biasanya digunakan oleh sistem rekomendasi untuk memecahkan masalah kelebihan informasi atau data dengan memberikan saran berupa item kepada pengguna tertentu sesuai dengan preferensi yang dimiliki pengguna. Sistem rekomendasi telah terbukti sangat berguna di banyak bidang dan telah banyak digunakan di berbagai bidang, seperti di bidang perbelanjaan (Amazon), musik (Pandora), film (Netflix), perjalanan (TripAdvisor), restoran (Yelp), media sosial (Facebook), dan artikel (TED) (Ghuribi & Noah, 2019).

Empat pendekatan yang umum digunakan oleh sistem rekomendasi akhir-akhir ini yaitu *Collaborative Filtering*, *Content-Based*, *Knowledge-Based*, dan *Hybrid*. *Collaborative Filtering* merupakan pendekatan yang memanfaatkan sebuah komunitas pengguna untuk memberikan rekomendasi kepada sebuah pengguna berdasarkan preferensi yang dimiliki oleh pengguna lain yang berhubungan, oleh karena itu, *Collaborative Filtering* juga disebut sebagai pendekatan 'korelasi orang-ke-orang'. *Content-Based* adalah pendekatan sistem yang mengaitkan konten turunan dari item dengan profil atau karakteristik pengguna. Dalam pendekatan ini, sistem belajar untuk merekomendasikan item yang mirip dengan yang disukai pengguna di masa lalu atau sesuai preferensi pengguna tanpa campur tangan pengguna lain, jadi output yang akan dihasilkan oleh *Content-Based* didasarkan pada kesamaan antar item satu sama lain. *Knowledge-Based* merupakan pendekatan yang bergantung pada beberapa jenis pengetahuan eksternal dari masing-masing item. Kelemahan utama dari *Knowledge-Based* adalah sulitnya untuk mendapatkan pengetahuan eksternal tersebut. Maka dari itu, *Knowledge-Based* hanya dapat digunakan pada skenario tertentu saja. Sedangkan *Hybrid Filtering* merupakan pendekatan yang menggabungkan cara kerja antara *Collaborative Filtering* dan *Content-Based* (Tarnowska, Ras & Daniel, 2020).

## 2.2. Saluran Youtube

Saluran Youtube adalah tempat para pengguna Youtube untuk mengelompokkan video yang dibuat dan diunggah, video yang ditonton dan disukai, serta daftar putar video yang dibuat. Setiap saluran Youtube memiliki alamat web (URL) yang dapat menjadi sebuah identitas. Para pengguna yang memiliki akun Youtube dapat berlangganan ke beberapa saluran Youtube, itu berarti, ketika pengguna yang berlangganan ke sebuah saluran tertentu membuka YouTube, maka video yang diunggah oleh saluran yang berlangganan akan hadir di halaman beranda YouTube mereka, serta pengguna akan mendapatkan notifikasi setiap kali saluran yang berlangganan mengunggah video baru (*Queensland Government*, 2020).

## 2.3. Neural Network

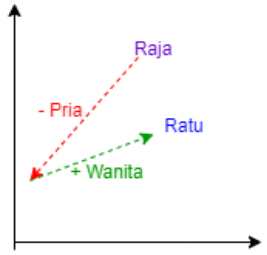
*Neural Network*, merupakan sebuah tipe dari *Machine Learning* yang memiliki karakteristik utama berupa memiliki dua atau lebih lapisan tersembunyi dalam prosesnya. *Neural Network* terinspirasi dari jaringan otak manusia yang diterjemahkan ke dalam komputer. Lapisan padat, lapisan yang paling umum, terdiri dari *neuron* yang saling berhubungan. Pada lapisan padat, setiap *neuron* pada lapisan tertentu terhubung ke setiap *neuron* pada lapisan berikutnya, yang berarti nilai output dari *neuron* sebelumnya akan menjadi input untuk *neuron* berikutnya. Setiap koneksi antar *neuron* memiliki bobot yang dimilikinya, yang nantinya, bobot-bobot tersebut akan dilakukan perhitungan dengan nilai input untuk menghasilkan nilai bobot baru (Kinsley & Kukiela, 2020).

*Neural Network* telah terbukti berguna untuk memecahkan berbagai masalah sulit, seperti mengenali hal-hal dalam gambar dan dalam pemrosesan bahasa. *Neural Network* mengambil pendekatan yang berbeda untuk pemecahan masalah dibandingkan dengan program komputer konvensional. Untuk memecahkan masalah, program konvensional menggunakan pendekatan algoritmik, yaitu komputer mengikuti serangkaian instruksi untuk memecahkan masalah. Sebaliknya, *Neural Network* menyelesaikan masalah dengan mencoba meniru cara kerja *neuron* di otak manusia (Taylor, 2017).

## 2.4. Word Embeddings

*Word Embeddings* adalah alat yang sangat berguna dan serbaguna yang telah berguna untuk menyelesaikan banyak masalah mendasar dalam bidang penelitian *Natural Language Processing*. *Word Embeddings* juga sering diterapkan secara luas dalam pencarian informasi, sistem rekomendasi, deskripsi gambar, penemuan hubungan, dan terjemahan tingkat kata. Selain itu, banyak aplikasi penting dibangun di atas *Word Embeddings*. Beberapa contoh diantaranya adalah *Long Short Term Memory* (LSTM) yang digunakan untuk pemodelan bahasa, terjemahan mesin, peringkasan teks dan pembuatan keterangan gambar (Yin & Shen, 2018).

*Word Embeddings* dapat mewakili kata semantik melalui vektor numerik dengan cara mengamati pola kemunculannya. *Word Embeddings* dapat digunakan untuk mengukur kesamaan kata dan mengubah praktik studi berbasis korpus yang telah berlangsung sejak lama dalam bidang linguistik (Hellrich, 2017). Kesamaan linguistik memiliki banyak segi. Misalnya, dua kata mungkin memiliki kesamaan dalam hal semantik, sintaksis, maupun morfologi satu sama lain. *Word Embeddings* telah terbukti dapat menangkap sebagian besar kesamaan-kesamaan tersebut hingga ke tingkat tertentu. *Word Embeddings* biasanya dilatih untuk menghasilkan representasi yang dapat menangkap kesamaan linguistik. Misalnya, “es” dan “dingin” memiliki kesamaan topik, “es” dan “api” memiliki kesamaan secara sintaksis karena keduanya merupakan kata benda, dan “kedinginan” dan “dingin” memiliki kesamaan morfologis karena keduanya berasal dari akar yang sama. Dengan *Word Embeddings*, kata-kata tersebut dapat ditangkap oleh komputer sebagai kata-kata yang memiliki kesamaan satu sama lain (Cotterell & Schutze, 2019). Melalui penggunaan kamus kata yang sangat besar, *Word Embeddings* biasanya menghasilkan ruang vektor dengan ratusan dimensi untuk memahami tingkat kesamaan yang berbeda antara kata-kata.




Gambar 1. Perhitungan Aritmatika Vektor (Di Gennaro, et al. 2021)

Proporsi kesamaan seperti "Pria" dengan "Wanita" dan "Raja" dengan "Ratu" dapat direproduksi melalui perhitungan aritmatika vektor, memungkinkan untuk mengekspresikan hubungan antara kata-kata ke dalam sebuah perhitungan matematika. Misalnya, persamaan "Raja" - "Pria" + "Wanita" akan menghasilkan vektor "Ratu" sebagai tetangga terdekat, persamaan tersebut jelas sangat berguna dalam *Natural Language Processing,* khususnya pada sistem rekomendasi, karena dapat menghasilkan kesamaan kata tidak hanya berdasarkan intensitas kemunculannya saja, namun juga berdasarkan kesamaan linguistik dan makna dari masing-masing kata (Gennaro, Buonnano & Palmieri, 2021).

## 2.5. Word2Vec

*Word Embeddings,* hingga saat ini sudah memiliki cukup banyak model algoritma yang dikembangkan, tetapi model yang menggunakan arsitektur *Neural Network* hanya ada sedikit, salah satunya adalah model yang dikenal dengan nama *Word2Vec*. Produksi *Word Embeddings* melalui *Word2Vec* dapat dilakukan dengan dua cara berbeda: yaitu *Continuous Bag-of-Words* (CBOW) dan *Skip-Gram* (SG). Kedua pendekatan memiliki perbedaan pada variabel input dan output, tetapi pada dasarnya kedua pendekatan tersebut menggunakan struktur jaringan yang sama. CBOW bekerja dengan cara memprediksi sebuah kata dari kata-kata di sekitar berdasarkan jumlah *window size* yang ditentukan, sedangkan *Skip-Gram* merupakan kebalikannya, *Skip-Gram* bekerja dengan cara memprediksi kata-kata di sekitar sebuah kata sesuai dengan jumlah *window size* yang ditentukan (Gennaro, Buonnano & Palmieri, 2021).

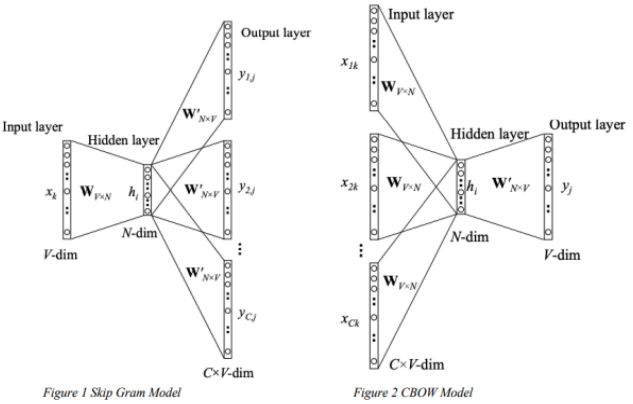
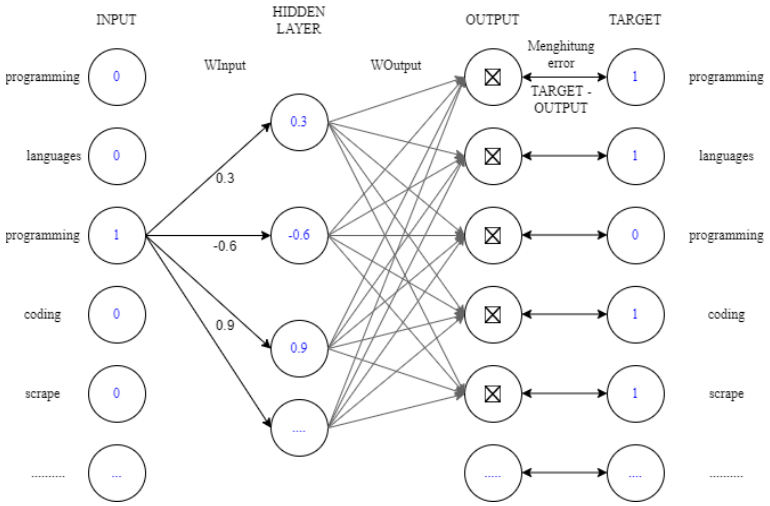



Figure 1 Skip Gram Model  Figure 2 CBOW Model

Gambar 2. Perbedaan model Skip Gram dan CBOW (Di Gennaro, et al. 2021)

*Word2Vec* bekerja menggunakan arsitektur *Neural Network* dengan cara mengambil kamus kata yang besar sebagai input, lalu menghasilkan representasi vektor berupa *hidden layer* dari setiap kata yang ada pada kamus kata tersebut sebagai output. Setiap kata akan melihat, memprediksi, dan menghasilkan nilai

vektor yang hampir sama dengan nilai vektor dari kata sebelum dan sesudah kata sesuai jumlah *window size* yang diinisialisasi secara berulang-ulang. Semakin sering berjumpa, maka nilai kesamaan dari dua kata akan semakin besar. Walaupun dua kata tidak selalu berdekatan, maka akan ada kemungkinan kedua kata tersebut memiliki nilai kesamaan besar, jika kedua kata tersebut sama-sama sering menjumpai kata sekitar yang mirip ataupun sama. Gambaran proses pelatihan *Word2Vec* dengan arsitektur *Neural Network* dapat dilihat pada Gambar 3.




Gambar 3. Proses pelatihan *Word2Vec* dengan arsitektur *Neural Network*

File vektor yang dihasilkan dapat digunakan untuk penelitian pada pemrosesan bahasa alami dan aplikasi pembelajaran mesin seperti sistem rekomendasi. Vektor kata tersebut juga dapat digunakan untuk mengukur jarak kedekatan antar vektor kata satu sama lain. Output dari proses *training Word2Vec* akan menghasilkan vektor berukuran jumlah kata * jumlah dimensi yang ditentukan. Gambaran vektor yang dihasilkan dari proses *training Word2Vec* dapat dilihat pada Gambar 4.

| Input | |
|---|---|
| d1, d2, d3 | … the quick brown … The quick brown and over cunny fox … the fast … the fox is brown and quick … |

| Output | f1 | f2 | f3 | f4 | f5 | f6 | f7 | … | fn |
|---|---|---|---|---|---|---|---|---|---|
| the | num | num | num | num | num | num | num | … | num |
| quick | num | num | num | num | num | num | num | … | num |
| brown | num | num | num | num | num | num | num | … | num |
| and | num | num | num | num | num | num | num | … | num |
| over | num | num | num | num | num | num | num | … | num |
| cunny | num | num | num | num | num | num | num | … | num |
| fox | num | num | num | num | num | num | num | … | num |
| fast | num | num | num | num | num | num | num | … | num |
| … | num | num | num | num | num | num | num | … | num |

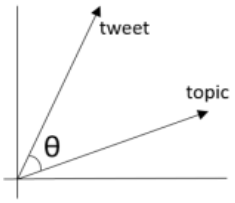
Gambar 4. Gambaran Output Word2Vec (Jansen, 2017)

*Word2Vec* dapat memahami hubungan semantik antar kata. Kata-kata yang memiliki kesamaan secara semantik, akan dipetakan ke dalam sebuah ruang vektor yang berdekatan. *Word2Vec* biasa digunakan untuk mengolah kamus kata dalam jumlah besar dan termasuk model prediktif yang sangat efisien untuk mempelajari pola dari kamus kata besar yang tidak berlabel. Dengan menggunakan kamus kata yang baik, *Word2Vec* dapat memprediksi secara akurat arti kata berdasarkan riwayat kemunculannya. Prediksi tersebut dapat digunakan untuk menentukan makna semantik sebuah kata dengan kata-kata lainnya yang mirip.

*Word2Vec* merupakan metode *unsupervised learning* yang dapat menghasilkan representasi terdistribusi dari kata dan frasa dalam ruang vektor berdimensi tinggi. *Word2Vec* menggunakan *Neural Network* yang dilatih untuk dapat menangkap hubungan antara elemen bahasa dan konteks di mana mereka terjadi (Jansen, 2017).

## 2.6. Cosine Similarity

*Cosine Similarity* adalah perhitungan yang menghitung sudut *cosine* diantara dua vektor. Teknik perhitungan ini dapat menunjukkan derajat kemiripan antar dokumen yang diwakili oleh vektor, ketika dua vektor memiliki kesamaan makna, maka nilai kesamaannya pun akan tinggi dan memperoleh nilai hampir 1. Gambaran dari *Cosine Similarity* dapat dilihat pada Gambar 5.




Gambar 5. Gambaran Cosine Similarity (Carolina & Jorge, 2017)

Kata "topik" dan "tweet" direpresentasikan sebagai vektor, di mana setiap vektor memiliki frekuensi kata, dan kemudian, rumus perhitungan kesamaan diterapkan sebagai berikut.

$$cos = \frac{\sum_i^n A_i \times B_i}{\sqrt{\sum_i^n (A_i)^2} \times \sqrt{\sum_i^n (B_i)^2}}, \tag{2.1}$$

Vektor "topik" diwakili oleh A, dan vektor "tweet" diwakili oleh B. Jika hasil dari perhitungan *cosine similarity* menunjukkan angka yang besar, maka kedua kata tersebut memiliki kemiripan, begitupun sebaliknya (Carolina & Jorge, 2017).

## 2.7. Pearson Correlation

Formula *correlation* digunakan untuk mencari seberapa kuat hubungan diantara data. Formula tersebut akan menghasilkan nilai antara 0 hingga 1, dimana angka disekitar 0 merepresentasikan hubungan yang lemah, dan angka disekitar 1 merepresentasikan hubungan yang kuat. Korelasi antara data menghitung seberapa erat hubungan antara data. Salah satu jenis perhitungan korelasi yang banyak digunakan yaitu *Pearson Correlation. Pearson Correlation* menunjukkan hubungan linier antara dua buah data. Berikut adalah rumus dari *Pearson Correlation* :

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[\,n\Sigma x^2 - (\Sigma x)^2\,]\,[\,n\Sigma y^2 - (\Sigma y)^2\,]}} \tag{2.2}$$

*Pearson Correlation* dapat menjawab berapa nilai relasi dari data model yang dimiliki dengan data *benchmark*. *Pearson Correlation* juga dapat menjawab performa dari model *word embeddings* dengan cara menghitung korelasi model dengan data *benchmark* (Glen, 2021).

## 2.8. Precision

*Precision* dalam sistem rekomendasi yaitu perhitungan jumlah item yang relevan dalam daftar yang direkomendasikan dengan jumlah total data yang ada. *Precision* mengukur nilai dari masing-masing data dalam daftar rekomendasi yang disukai oleh pengguna dan relevan dengan preferensi yang dimiliki pengguna.

$$Precision@k = \frac{Jumlah\ data\ relevan\ pada\ top\ K}{K} \tag{2.3}$$

$$AveragePrecision@k = \frac{Jumlah\ precision@k\ dari\ n\ query}{jumlah\ sampel\ query} \tag{2.4}$$

$$MeanAveragePrecision@k = \frac{Jumlah\ AveragePrecision@k\ dari\ n\ partisipan/penguji}{jumlah\ partisipan/penguji} \tag{2.5}$$

*Precision@k* adalah fraksi dari item yang relevan pada top K hasil yang direkomendasikan dengan jumlah K. *Average Precision@k* adalah fraksi dari hasil *precision@k* dari jumlah sampel *query* yang diuji dengan jumlah sampel *query* keseluruhan. *Mean Average Precision@k* adalah fraksi dari hasil *Average Precision@k* dari jumlah partisipan/penguji yang menguji dengan jumlah partisipan/penguji secara keseluruhan (Thiago, 2019).

### 2.9. Penelitian Terkait

Penelitian mengenai sistem rekomendasi menggunakan *neural network word embeddings* dengan model *word2vec* beberapa kali telah dilakukan sebelumnya oleh peneliti lain. (Aldiansyah, 2019) melakukan penelitian berupa sistem rekomendasi lagu *cross language* berdasarkan lirik menggunakan *word2vec*. Pada penelitian tersebut, digunakan data 400 lirik lagu yang terdiri dari 200 lirik lagu berbahasa Indonesia dan 200 lirik lagu berbahasa Inggris. Data tersebut lalu dibersihkan dan diolah dengan metode yang telah disebutkan. Model yang digunakan adalah model *word2vec*. Nilai *precision* yang didapat dari sistem tersebut sebesar 0.388. Sistem tersebut memiliki nilai precision yang kurang memuaskan dikarenakan data yang digunakan hanya berjumlah sedikit.

Penelitian selanjutnya dilakukan oleh (Laili, 2019) yang melakukan penelitian tentang sistem rekomendasi film berdasarkan sinopsis menggunakan metode *word2vec*. Penelitian tersebut menggunakan model *word2vec* untuk mengetahui kesamaan semantik. Data yang digunakan sebagai data latih terdiri dari 150 data berbagai film. Nilai precision yang didapat dari proses pengujian adalah sebesar 0.726.

### 2.10. Tabel Perbandingan Penelitian

Tabel 1 menampilkan garis besar mengenai perbandingan antara penelitian yang dilakukan dengan penelitian-penelitian sebelumnya.

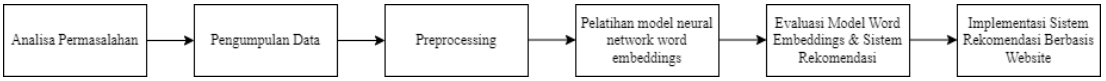
Tabel 1. Perbandingan dengan penelitian sebelumnya

| Peneliti | Judul | Studi Kasus | Dataset |
|---|---|---|---|
| (Aldiansyah, 2021) | Sistem Rekomendasi Lagu *Cross Language* Berdasarkan Lirik Menggunakan *word2vec*. | Lirik Lagu | 400 data lirik lagu, 200 lagu berbahasa Indonesia, 200 lagu berbahasa Inggris |
| (Laili, 2019) | Sistem Rekomendasi Film Berdasarkan Sinopsis Menggunakan Metode *word2vec* | Film | 150 data berbagai *genre* film |
| (Farhan, 2022) | Sistem Rekomendasi Saluran Youtube Edukasi Secara Semantik Menggunakan Neural Network Word Embeddings | Saluran Youtube Edukasi | 12551 data saluran Youtube berkonten edukasi |

# BAB III
# METODE PENELITIAN

## 3.1. Metode Penelitian

Penelitian ini membutuhkan beberapa tahapan yang perlu dilakukan, tahapan-tahapan tersebut diantaranya adalah tahap analisa permasalahan, pengumpulan data, *preprocessing*, implementasi dan pelatihan kamus kata menggunakan *word embeddings*, evaluasi model, penerapan sistem rekomendasi, pembuatan website & penghitungan kemiripan, serta tahapan evaluasi sistem. Garis besar dari tahapan yang akan dilalui dalam penelitian dapat dilihat pada gambar 6.




Gambar 6. Alur Metode Penelitian

### 3.1.1. Analisa Permasalahan

Tahap analisa permasalahan merupakan tahapan yang berfokus kepada permasalahan-permasalahan yang terjadi di sekitar. Pada penelitian ini, terdapat permasalahan mengenai sulitnya mencari saluran Youtube yang memiliki konten-konten dengan tema edukasi secara spesifik berdasarkan kata kunci ataupun kesamaan konten dengan saluran lain. Permasalahan tersebut datang karena saat ini, Youtube sudah menjadi salah satu sumber yang sering dijadikan tempat untuk mendapatkan ilmu maupun keterampilan bagi banyak orang, namun saja, Youtube tidak memiliki fitur rekomendasi saluran secara spesifik pada edukasi saja, melainkan hanya merekomendasikan video-video berdasarkan yang disukai pengguna di masa lalu. Sistem pencarian saluran pada Youtube pun tidak dapat menampilkan saluran-saluran yang memiliki konten berdasarkan kata kunci, melainkan hanya menampilkan saluran-saluran yang mengandung nama yang berada di kata kunci. Penelitian mengenai sistem rekomendasi di bidang edukasi sudah beberapa kali dilakukan sebelumnya, namun metode yang digunakan pada penelitian-penelitian tersebut tidak bisa menangkap kesamaan makna semantik antar kata. Dari permasalahan tersebut, dapat disimpulkan bahwa dibutuhkan sebuah sistem yang dapat merekomendasikan saluran-saluran Youtube berdasarkan saluran Youtube lain dengan konten yang serupa ataupun kata kunci yang diinputkan pengguna yang dapat menangkap makna semantik dari masing-masing kata yang berada di dokumen.

Tahap analisa permasalahan juga adalah tahap penentuan sebuah metode yang harus digunakan oleh sistem agar dapat menghasilkan sistem yang bekerja dengan baik dan benar sesuai permasalahan. Oleh karena itu, digunakanlah metode *neural network word embeddings* dengan model *word2vec* dan pendekatan *content based filtering* untuk membangun sistem tersebut. Pada tahap ini, juga dibutuhkan banyak sumber referensi yang digunakan untuk menunjang penelitian ini, sumber-sumber referensi tersebut dikumpulkan dari berbagai jenis, seperti jurnal, skripsi, buku, ataupun literatur dari internet.

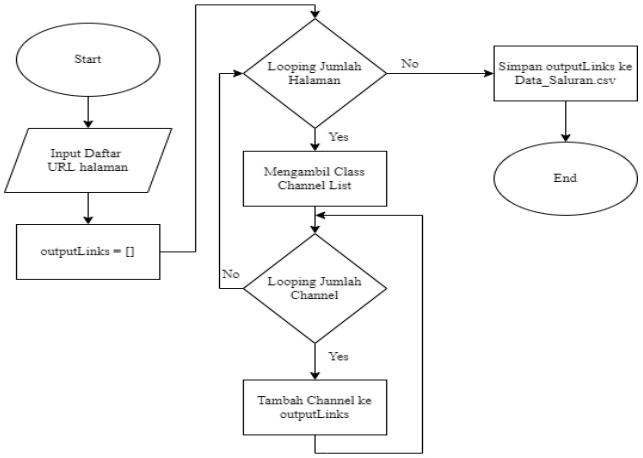
### 3.1.2. Pengumpulan Data

Data berupa saluran-saluran Youtube edukasi yang akan digunakan pada sistem rekomendasi saluran youtube edukasi ini merupakan data-data saluran youtube yang

sering mengunggah konten dengan kategori *How To & Style, Pets & Animals,* dan *Science & Technology* karena kategori-kategori tersebut termasuk kedalam kategori mengenai ilmu pengetahuan serta kreator dengan kategori tersebut seringkali mengunggah video-video yang bertujuan untuk mengedukasi para penonton.. Untuk mendapatkan data-data saluran yang memiliki kategori tersebut, digunakan sebuah alat berupa website bernama *The Youtube Channel Crawler* (https://channelcrawler.com) yang memiliki data-data tersebut. Data-data tersebut lalu di *scraping* untuk melakukan proses pengekstrakan url dari masing-masing saluran kedalam file csv. Dari proses *scraping* tersebut, terkumpul data berjumlah 12551 data berupa kumpulan url. Setelah url terkumpul, proses selanjutnya adalah mengumpulkan atribut-atribut dari masing-masing url dengan menggunakan youtube *Application Programming Interface* (API). Atribut-atribut yang diambil dari API adalah atribut berupa nama saluran, jumlah pelanggan, deskripsi, jumlah penonton, negara, url foto profil, kata kunci, dan judul video-video yang terakhir diunggah.

Proses pengumpulan data pada penelitian ini seluruhnya dilakukan menggunakan metode *web scraping*. *Web scraping*, juga dikenal sebagai *web extraction* adalah teknik untuk mengekstrak data dari *World Wide Web* (WWW) dan menyimpannya ke dalam file sistem ataupun database. Umumnya, data web diekstrak menggunakan *Hypertext Transfer Protocol* (HTTP) atau melalui browser web. Proses tersebut dapat dilakukan baik secara manual oleh pengguna atau secara otomatis oleh bot atau *web crawler* (Zhao, 2017).

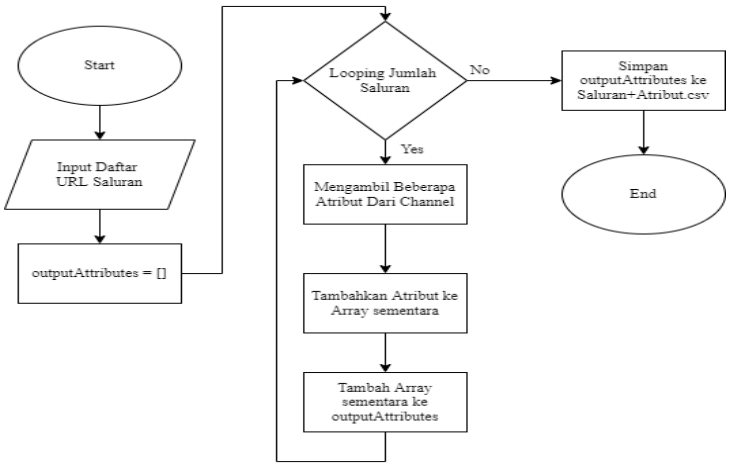
Proses pengumpulan data pada penelitian ini terbagi menjadi dua proses, yaitu proses pengambilan url dan proses pengambilan atribut. Gambaran tahapan dari proses pengambilan url dapat dilihat pada Gambar 7.




Gambar 7. Flowchart Proses Pengambilan URL

Proses pengekstrakan url dimulai dengan menginputkan daftar url dari Channel Crawler sesuai dengan kategori dan nomor halaman. Lalu program akan otomatis mengulang dan mengekstrak url dari halaman web hingga index lebih dari sama dengan jumlah halaman dari *channel crawler*. Setelah pengulangan selesai, output berupa url dari masing-masing saluran yang disimpan dalam variabel output akan ditambahkan ke file csv.

Tahap selanjutnya setelah pengambilan url adalah mengambil data-data atribut menggunakan Youtube API berdasarkan url-url yang telah diekstrak sebelumnya. Gambaran proses pengambilan atribut dapat dilihat pada Gambar 8.

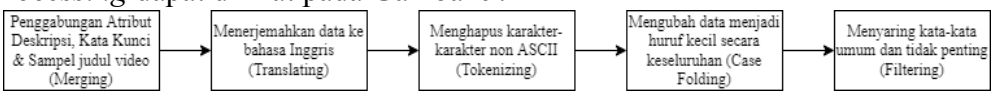



Gambar 8. Flowchart Proses Pengambilan Atribut

Proses pengambilan atribut dimulai dengan menginputkan daftar URL yang telah dilakukan sebelumnya. Lalu program akan otomatis mengulang dan mengekstrak atribut-atribut melalui Youtube API hingga index lebih dari sama dengan jumlah daftar url. Setelah pengulangan selesai, output berupa atribut-atribut yang disimpan dalam variabel output akan ditambahkan ke file CSV.

### 3.1.3. Preprocessing

*Text Preprocessing* menjadi langkah pertama dalam alur sistem rekomendasi, karena proses ini dapat berpotensi untuk mempengaruhi dampak ataupun keakuratan dari sistem rekomendasi (Camacho, 2017). Tujuan utama dari *text preprocessing* adalah untuk mendapatkan fitur kunci dari kumpulan data dokumen teks dan untuk meningkatkan relevansi antara kata dan dokumen (Kadhim, 2018). Pada penelitian ini, tahapan *preprocessing* dibagi kembali menjadi beberapa tahapan. Tahapan dari *preprocessing* dapat dilihat pada Gambar 9.




Gambar 9. Tahapan Preprocessing

**Merging**

Tahapan *preprocessing* yang pertama dilakukan adalah *merging, merging* dilakukan untuk menggabungkan atribut-atribut pada *dataset* yang akan dijadikan parameter pada sistem rekomendasi. Dari atribut-atribut yang dihasilkan dari proses *web scraping* di tahap sebelumnya, atribut yang akan digunakan untuk dilatih pada proses *word embeddings* adalah atribut deskripsi, kata kunci, dan sampel judul video. Gambaran contoh dari proses *merging* dapat dilihat pada Tabel 2.

Tabel 2. Tabel Gambaran Merging

| **Sebelum** | | | **Sesudah** |
|---|---|---|---|
| **Deskripsi Saluran** | **Kata Kunci** | **Judul Video** | **Atribut Gabungan** |
| Channel yang akan mengajarkan anda tentang bahasa pemrograman | “pemrograman”, ”koding” | Cara scraping web dengan Python | Channel yang akan mengajarkan anda tentang bahasa pemrograman “pemrograman”, ”koding” Cara scraping web dengan |

| | | | |
|---|---|---|---|
| | | | Python |
| Channel yang berisi tutorial tentang bahasa pemrograman Python | “python”, ”pemrograman” | Web programming dengan Python | Channel yang berisi tutorial tentang bahasa pemrograman Python “python”, ”pemrograman” Web programming dengan Python |

**Translating**

Tahapan selanjutnya yang akan dilakukan adalah tahap *translating* atau menerjemahkan atribut gabungan ke dalam bahasa Inggris. Hal ini dilakukan karena *dataset* yang akan digunakan pada penelitian ini terdiri dari saluran-saluran Youtube edukasi berbahasa Indonesia dan Inggris, sehingga atribut perlu diterjemahkan agar sistem dapat menangkap kesamaan dua saluran yang serupa walaupun berbeda bahasa. Pada tahapan ini, data diterjemahkan dengan menggunakan *Google Translate API*. Gambaran dari proses *translating* dapat dilihat pada Tabel 3.

Tabel 3. Tabel Gambaran Proses Translating

| Sebelum | Sesudah |
|---|---|
| Channel yang akan mengajarkan anda tentang bahasa pemrograman “pemrograman”, ”koding” Cara scraping web dengan Python | A channel that will teach you about programming languages "programming", "coding" How to scrape the web with Python |
| Channel yang berisi tutorial tentang bahasa pemrograman Python “python”, ”pemrograman” Web programming dengan Python | Channel that contains tutorials on the Python programming language "python", "programming" Web programming with Python |

**Tokenizing**

Tahapan *preprocessing* yang selanjutnya dilakukan pada penelitian ini adalah *tokenizing, tokenizing* dilakukan untuk menghapus karakter-karakter non alphabet agar menghasilkan sebuah teks bersih yang hanya berisi kata-kata, karakter-karakter yang akan dihapus pada tahapan ini yaitu karakter-karakter Non-ASCII. Perbandingan dokumen sebelum dilakukan *tokenizing* dengan setelah dilakukan *tokenizing* dapat dilihat pada Tabel 4.

Tabel 4. Tabel Perbandingan Tokenizing

| Sebelum | Sesudah |
|---|---|
| A channel that will teach you about programming languages "programming", "coding" How to scrape the web with Python | A channel that will teach you about programming languages programming coding How to scrape the web with Python |
| Channel that contains tutorials on the Python programming language "python", "programming" Web programming with Python | Channel that contains tutorials on the Python programming language python programming Web programming with Python |

**Case Folding**

*Case Folding* merupakan tahapan *preprocessing* yang bertujuan untuk membuat semua huruf alphabet menjadi huruf kecil tanpa terkecuali. Hal ini dilakukan agar komputer dapat menganggap bahwa dua kata adalah sama walaupun kedua kata tersebut memiliki perbedaan dari segi huruf besar atau huruf kecil. Tahapan ini sangat diperlukan untuk menambah performa dan relevansi sistem. Perbandingan dokumen sebelum dan setelah dilakukan *case folding* dapat dilihat pada Tabel 5.

Tabel 5. Tabel Perbandingan Case Folding

| Sebelum | Sesudah |
|---|---|
| A channel that will teach you about programming languages programming coding How to scrape the web with Python | a channel that will teach you about programming languages programming coding how to scrape the web with python |
| Channel that contains tutorials on the Python programming language python programming Web programming with Python | channel that contains tutorials on the python programming language python programming web programming with python |

**Filtering**

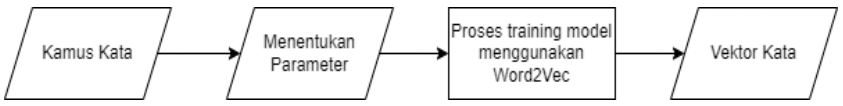
*Filtering* merupakan tahapan penyaringan kata-kata tidak penting yang terdapat pada dokumen. Pada tahapan ini, dilakukan penghapusan pada kata-kata yang sering diulang yang muncul di setiap dokumen, kata-kata konjungsi seperti "*or*", "*and*", "*the*" dan kata ganti "*he*", "*they*", "*that*", dan sebagainya perlu dihilangkan karena tidak akan memiliki pengaruh dan kata-kata tersebut hanya akan menambah nilai kesamaan pada dokumen yang tidak memiliki kesamaan sama sekali. Perbandingan dokumen sebelum dengan setelah dilakukan *filtering* dapat dilihat pada Tabel 6.

Tabel 6. Tabel Perbandingan Filtering

| Sebelum | Sesudah |
|---|---|
| a channel that will teach you about programming languages programming coding how to scrape the web with python | channel teach programming language programming coding scrape web python |
| channel that contains tutorials on the python programming language python programming web programming with python | channel tutorials python programming language python programming web programming python |

### 3.1.4. Pelatihan Model Neural Network Word Embeddings

Tahap selanjutnya adalah tahap implementasi *neural network word embeddings.* Pada dasarnya, tahap ini terbagi kedalam 4 tahapan yang dapat dilihat pada Gambar 10.




Gambar 10. Tahapan Word Embeddings

Kata-kata dari atribut gabungan yang telah dibersihkan akan dijadikan input berbentuk kamus kata yang akan di *training* menggunakan metode *Word Embeddings*

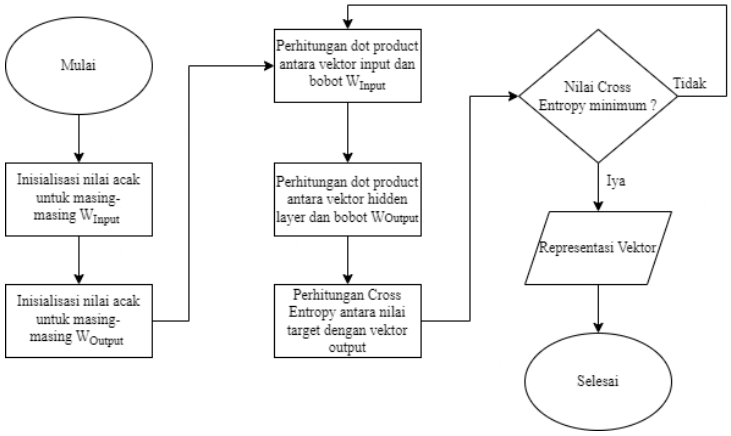
dengan bantuan model *word2vec* yang akan menghasilkan representasi vektor berupa angka yang dapat dilakukan perhitungan matematika serta perhitungan kesamaan.

Tahap selanjutnya adalah tahap penentuan parameter yang akan digunakan pada proses *training*. Penentuan parameter adalah tahap yang cukup penting karena bisa mempengaruhi performa dari model *word embeddings* yang dilatih. Berikut adalah beberapa parameter yang akan menjadi acuan pada proses evaluasi :

a. Algoritma : Algoritma yang digunakan pada proses *training*, terdiri dari *Skip-Gram* dan *CBOW*
*b.* *Window Size* : Jumlah maksimum kata sebelum dan sesudah kata target yang digunakan untuk memprediksi konteks, terdiri dari 2, 5, 7, dan 10 *window size.*
*c.* *Dimension Size* : Nilai panjang dari masing-masing vektor kata, terdiri dari 100, 200, dan 300 jumlah *dimension size.*

Setelah parameter ditentukan, proses selanjutnya adalah proses *training*, proses ini membutuhkan waktu yang cukup lama karena kamus kata yang besar dan proses yang dilakukan berulang-ulang.

Proses ini akan dilakukan berulang-ulang hingga mencapai nilai error minimum, sehingga nilai pada vektor *hidden layer* akan dapat merepresentasikan kata berdasarkan makna semantik sebaik mungkin. Setelah proses *training* selesai, maka akan dihasilkan sebuah output berupa representasi vektor berjumlah jumlah kata input * jumlah dimensi yang ditentukan yang contohnya bisa dilihat pada Gambar 4. Vektor tersebutlah yang akan digunakan pada proses sistem rekomendasi dengan cara menghitung kemiripannya menggunakan *cosine similarity*. Gambaran terperinci dari proses training *Word Embeddings* dapat dilihat pada Gambar 11.




Gambar 11. *Flowchart* terperinci proses *training Word2Vec*

Proses perhitungan kemiripan dilakukan untuk menghitung tingkat kemiripan antara dua dokumen pada ruang vektor yang dihasilkan dari proses *training*. Dengan perhitungan ini, sistem dapat menentukan angka kemiripan antara satu data saluran dengan data saluran yang lainnya. Proses ini dilakukan dengan menggunakan *cosine similarity,*

Terdapat dua dokumen yang akan digunakan sebagai contoh, yang berisi atribut gabungan dari dua saluran Youtube yang berbeda yang telah dilakukan proses *preprocessing*. Selanjutnya, kata-kata tersebut dijadikan kamus kata untuk dilakukan proses *Word Embeddings* yang akan menghasilkan representasi vektor numerik dengan masing-masing berjumlah vektor sesuai dengan yang diinisialisasikan pada

parameter *dimension size*. Misalnya, jumlah dimensi yang diinisialisasikan berjumlah 5, maka akan menghasilkan representasi vektor yang dapat dilihat pada Tabel 7.

Tabel 7. Representasi Vektor

| **Kata Input** | **Vektor 1** | **Vektor 2** | **Vektor 3** | **Vektor 4** | **Vektor 5** |
|---|---|---|---|---|---|
| channel (A,B) | -0.052 | 0.085 | -0.002 | 0.056 | -0.021 |
| teach (A) | 0.067 | -0.035 | -0.081 | 0.096 | -0.097 |
| programming (A,B) | -0.007 | -0.085 | -0.026 | -0.025 | 0.090 |
| language (A,B) | -0.056 | 0.028 | 0.072 | -0.006 | 0.064 |
| coding (A) | -0.086 | 0.002 | 0.072 | -0.092 | -0.089 |
| scrape (A) | 0.078 | -0.055 | 0.096 | -0.051 | -0.015 |
| web (A,B) | -0.085 | -0.076 | 0.032 | -0.056 | -0.058 |
| python (A,B) | 0.009 | 0.081 | 0.023 | -0.096 | -0.089 |
| tutorials (B) | -0.077 | 0.088 | 0.014 | 0.040 | -0.046 |

Masing-masing bobot dari kata-kata unik yang terdapat pada saluran akan dijumlahkan lalu diambil rata-rata untuk dilakukan perhitungan *cosine similarity*. Maka, hasil representasi vektornya adalah sebagai berikut :

a. Saluran A ("*channel teach programming language programming coding scrape web python*") = [-0.015, -0.015, 0.016, -0.022, -0.014].

b. Saluran B ("*channel tutorials python programming language python programming web programming python*") = [-0.026, 0.011, 0.009, 0.033, -0.006].

Angka-angka dalam vektor tersebut lalu dimasukkan ke dalam persamaan (2.1) seperti berikut ini:

$$Cosine\ Similarity(A,B) = \frac{0.047}{\sqrt{0.001386} + \sqrt{0.0013496}}$$

$$Cosine\ Similarity(A,B) = \frac{0.047}{0.037+0.037}$$

$$Cosine\ Similarity(A,B) = \frac{0.047}{0.074} = 0.6351$$

Hasil kemiripan antara saluran A dan saluran B yang diperoleh dari perhitungan *cosine similarity* tersebut hasilnya cukup besar, yakni 0.6351. Hasil tersebut mengisyaratkan bahwa saluran A dan saluran B memiliki kemiripan yang cukup besar.

### 3.1.5. Evaluasi Model & Sistem

Proses evaluasi merupakan proses yang dilakukan untuk mengukur performa dari sistem. Pada penelitian ini, proses evaluasi dilakukan dua kali, evaluasi pertama adalah evaluasi model *word embeddings* dan yang kedua adalah evaluasi sistem. Evaluasi model *word embeddings* dilakukan dengan menghitung korelasi menggunakan *pearson correlation* dari model yang telah dilatih dengan data ukur bernama *Simlex-999,* yang merupakan data ukur yang digunakan untuk menghitung seberapa baik model dalam menangkap kesamaan dari beberapa kata. *Simlex-999* merupakan data ukur yang dikembangkan oleh Felix Hill yang melibatkan setidaknya 500 orang yang menggunakan bahasa Inggris sebagai bahasa utama untuk memberi nilai kesamaan dari beberapa kata. Evaluasi model akan dilakukan pada 24 skenario pada model dengan parameter berbeda. Parameter yang akan dijadikan acuan pada evaluasi model adalah parameter algoritma, *window,* dan *dimension size*. Model dengan nilai korelasi terbaiklah yang akan digunakan pada sistem.

Proses evaluasi kedua adalah proses evaluasi sistem. Evaluasi sistem akan dilakukan menggunakan *precision* sebagai ukuran ketepatan antara data saluran yang diminta oleh pengguna dengan jawaban yang akan diberikan oleh sistem. Proses inilah yang akan menentukan performa sistem dalam merekomendasikan data yang relevan kepada pengguna.

#### 3.1.6. Implementasi User Interface

Sistem rekomendasi yang telah dibuat selanjutnya akan diimplementasikan dalam bentuk *Graphical User Interface*(GUI) berbasis website. Pembuatan website bertujuan untuk memudahkan dalam melakukan proses evaluasi sistem serta dapat menunjukkan bagaimana sistem rekomendasi bekerja.

### 3.2. Komponen Alat dan Bahan

Komponen alat dan bahan sangat dibutuhkan untuk digunakan pada penelitian ini. Komponen-komponen tersebut diantaranya adalah sebagai berikut:

#### 3.2.1. Alat

1. Perangkat Keras :
   a. *Laptop* dengan *Processor* AMD Athlon Gold, *Random Access Memory* 12GB
2. Perangkat Lunak :
   a. Google Collab
   b. Python 3

#### 3.2.2. Bahan

a. Data daftar url saluran Youtube bertema edukasi
b. Data atribut-atribut dari masing-masing saluran Youtube

# BAB IV
# PERANCANGAN DAN IMPLEMENTASI

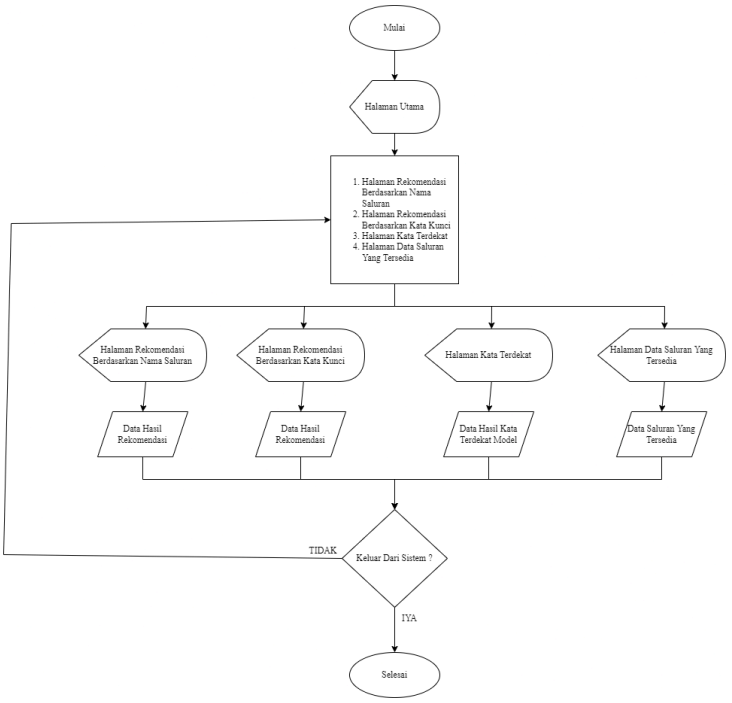
## 4.1. Tahap Proses Analisis

Tahap analisis dan perencanaan merupakan tahapan paling awal yang akan dilakukan pada penelitian ini. Tahapan ini adalah tahapan yang penting karena akan memudahkan dan mempercepat proses penelitian, terutama pada proses implementasi. Proses yang pertama kali dilakukan adalah menganalisis beberapa permasalahan yang ada dengan cara mencari berbagai sumber literatur, seperti jurnal, skripsi, buku, ataupun sumber-sumber literatur terpercaya yang terdapat di internet. Setelah proses analisa selesai, maka akan ditarik kesimpulan berupa perencanaan sistem.

## 4.2. Tahap Perancangan

Tahapan ini bertujuan untuk memberikan gambaran mengenai rancangan sistem yang akan dikembangkan, mencakup perancangan sistem secara umum dan secara detail, serta perancangan *user interface*.

Gambaran *flowchart* dari alur pada sistem rekomendasi untuk pengguna dapat dilihat pada gambar 12.




Gambar 12. Flowchart Sistem Untuk Pengguna

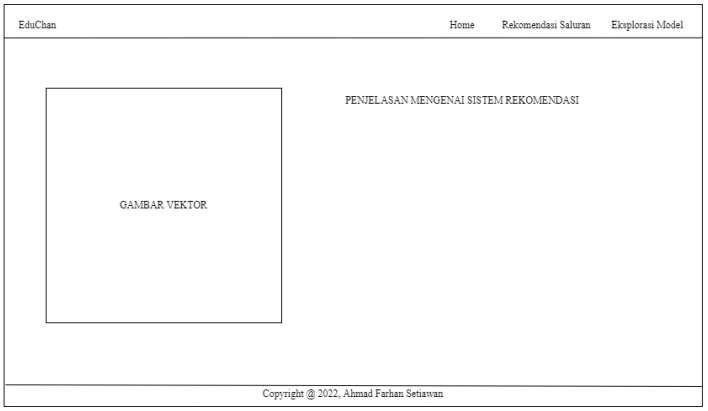
Pengguna dapat memilih untuk mencari rekomendasi saluran Youtube edukasi dengan menggunakan kata kunci yang diinputkan atau dengan menginputkan saluran Youtube edukasi favoritnya. Setelah pengguna menginputkan kata kunci ataupun nama saluran, sistem akan menghitung kemiripan data yang diinputkan dengan data-data saluran yang terdapat didalam *dataset* berdasarkan vektor *word embeddings* yang dimiliki dari kata yang terdapat pada masing-masing atribut, setelah itu sistem akan mengurutkan data berdasarkan nilai kesamaan yang paling besar, data-data saluran yang memiliki nilai kesamaan tersebut akan menjadi output dan dapat dilihat oleh pengguna.

### 4.2.1. *User Interface*

Berikut adalah beberapa rancangan *User Interface* untuk sistem rekomendasi saluran Youtube edukasi yang akan dibuat.

**Halaman Utama Sistem**

Halaman Utama merupakan halaman pertama yang akan dijumpai oleh pengguna ketika pertama kali mengakses sistem rekomendasi. Halaman ini juga sekaligus menjadi halaman awal agar pengguna bisa menggunakan sistem rekomendasi. Perancangan Halaman Utama pada sistem terdapat pada Gambar 13.

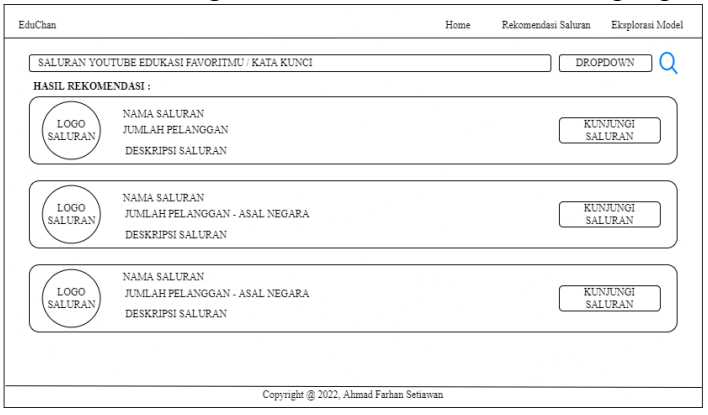



Gambar 13. Rancangan Halaman Utama

Halaman ini berisi penjelasan mengenai bagaimana cara menggunakan sistem dan bagaimana sistem bekerja dalam merekomendasikan saluran Youtube edukasi serupa.

**Halaman Rekomendasi**

Halaman rekomendasi merupakan halaman inti yang akan memberikan pengguna beberapa rekomendasi saluran Youtube edukasi lain yang serupa dengan saluran yang diinputkan oleh pengguna. Pada halaman ini, pengguna dapat memilih untuk mencari rekomendasi baik berdasarkan nama saluran edukasi favoritnya atau berdasarkan kata kunci. Rancangan halaman rekomendasi terdapat pada Gambar 14.

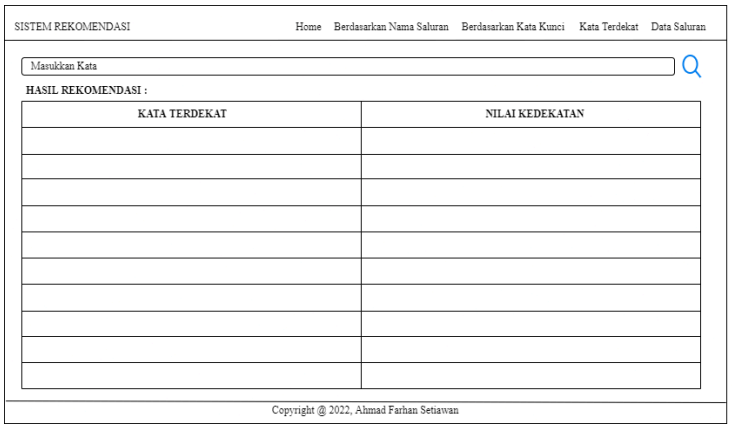



Gambar 14. Rancangan Halaman Rekomendasi

Halaman ini memungkinkan pengguna untuk mendapatkan rekomendasi saluran Youtube edukasi lain yang memiliki kesamaan dan kemiripan dengan saluran ataupun kata kunci yang diinputkan oleh pengguna. Pengguna juga dapat langsung mengunjungi saluran dengan menekan langsung tombol “Kunjungi Saluran” yang akan mengarah ke URL dari masing-masing saluran.

**Halaman Hasil Eksplorasi Model**

Halaman hasil eksplorasi model merupakan halaman yang memungkinkan pengguna untuk mengeksplorasi model *word embeddings* yang digunakan pada sistem rekomendasi yang telah melalui proses *training*. Pada halaman ini, pengguna dapat mengetahui beberapa kata-kata semantik dari kata yang diinputkan, mengetahui nilai kesamaan dari dua buah kata, serta melakukan perhitungan aritmatika kata. Rancangan Halaman Hasil Eksplorasi Model dapat dilihat pada Gambar 15.




Gambar 15. Rancangan Halaman Hasil Eksplorasi Model

Halaman ini bertujuan agar pengguna dapat mengeksplor model *word embeddings* yang digunakan pada sistem rekomendasi. Pengguna dapat melihat bagaimana performa model yang digunakan dalam menangkap kesamaan kata berdasarkan makna semantik.

## 4.3. Tahap Implementasi

Tahap implementasi adalah tahapan untuk mengimplementasikan perencanaan dan perancangan menjadi sebuah sistem sungguhan yang dapat digunakan. Tahapan ini terbagi menjadi dua tahapan, yaitu tahap pelatihan model *word embeddings* dan tahap pembuatan website.

### 4.3.1. Tahap Pelatihan Model *Word Embeddings*

Pada tahap ini, dilakukan proses pelatihan *word embeddings* dengan tahap-tahap yang telah dijelaskan pada metode penelitian. Tahapan ini dilakukan dengan menggunakan bahasa pemrograman *Python* dengan bantuan *Gensim*. Tahapan ini akan menghasilkan output berupa representasi vektor dari kata-kata yang dijadikan input pada model. Vektor tersebut dapat merepresentasikan makna semantik yang akan digunakan pada sistem rekomendasi. Pada tahap ini juga dilakukan evaluasi terhadap beberapa model dengan parameter berbeda dengan menggunakan *Pearson Correlation,* model yang mendapatkan nilai terbesarlah yang akan digunakan pada sistem rekomendasi.

```
def vectors():

    word_embeddings = []

    # Reading the each 'Description'
    for line in data['Cleaned_Fix']:
        avgword2vec = None
        count = 0
        for word in line.split():
            if word in model.wv.vocab:
                count += 1
                if avgword2vec is None:
                    avgword2vec = model[word]
                else:
                    avgword2vec = avgword2vec + model[word]

        if avgword2vec is not None:
            avgword2vec = avgword2vec / count
            word_embeddings.append(avgword2vec)
        else:
            word_embeddings.append(np.zeros(300))

    return word_embeddings

word_embeddings = vectors()
```

Gambar 16. Penggalan Kode Proses Pelatihan *Word Embeddings*

### 4.3.2. Tahap Pembuatan Website Sistem Rekomendasi

Tahapan ini merupakan tahapan yang memungkinkan sistem rekomendasi dapat digunakan oleh pengguna. Pada tahap ini, digunakan bantuan kerangka kerja *React.JS* untuk bagian *frontend*, dan kerangka kerja *Flask* untuk bagian *backend*. Pada tahap ini juga dilakukan evaluasi terhadap sistem rekomendasi dalam memberikan hasil rekomendasi kepada pengguna dengan menggunakan *Precision*.

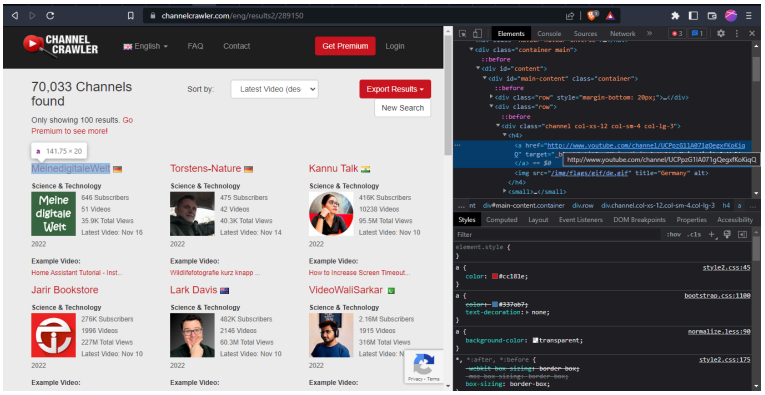
# BAB V
# HASIL DAN PEMBAHASAN

## 5.1. Hasil

Bab ini akan mengulas hasil jadi serta hasil evaluasi dari sistem rekomendasi saluran Youtube edukasi menggunakan *Word Embeddings*, yang meliputi hasil dari proses pelatihan model *word embeddings* dan juga hasil sistem rekomendasi yang berbasis website. Berikut adalah penjelasan mengenai beberapa hasil dari penelitian.

### 5.1.1. Pengumpulan Data

Pengumpulan data merupakan tahap untuk mengumpulkan data-data saluran Youtube edukasi untuk digunakan pada sistem rekomendasi dan untuk dijadikan input pada proses pelatihan model *Word Embeddings.* Proses pengumpulan data pada penelitian ini seluruhnya menggunakan metode *web scraping.* Proses *web scraping* dibagi kedalam dua tahapan, tahap pertama adalah mengekstrak *url* dari saluran-saluran yang memiliki kategori yang telah dijelaskan dari sebuah *website* yang memiliki data lengkap yang berisi *url-url* saluran Youtube bernama *The Youtube Channel Crawler.* Berikut adalah gambaran umum dari proses *scraping url* yang dilakukan, dapat dilihat pada Gambar 17.

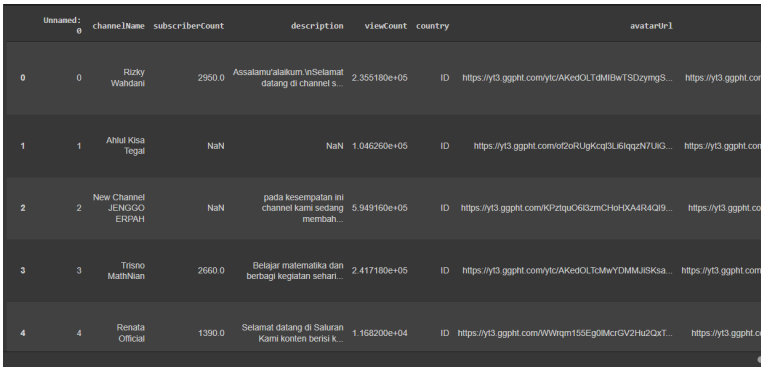



Gambar 17. Proses pengambilan *url* saluran

Proses tersebut menghasilkan *url* saluran Youtube dengan konten edukasi berjumlah 12551 data. Setelah dilakukan tahap pengumpulan *url*, langkah selanjutnya adalah proses pengambilan atribut-atribut yang dimiliki oleh masing-masing saluran. Proses pengambilan atribut ini dilakukan dengan menggunakan bantuan Youtube API yang merupakan *tool* resmi yang dimiliki Youtube untuk mengambil atribut-atribut yang dimiliki oleh para saluran Youtube. Pada tahap ini, *url-url* yang telah diambil sebelumnya akan dijadikan input untuk menghasilkan output berupa atribut-atribut yang dimiliki. Berikut adalah gambaran dari data berupa atribut yang dimiliki oleh masing-masing saluran, dapat dilihat pada Gambar 18.


Gambar 18. Contoh atribut yang dimiliki masing-masing saluran

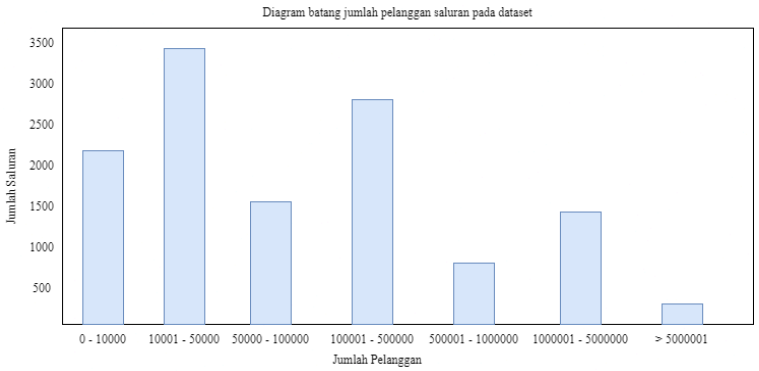
Terdapat banyak sekali atribut-atribut yang dimiliki oleh masing-masing saluran yang dapat disimpan kedalam format CSV. Namun hanya beberapa atribut saja yang disimpan kedalam *file* dengan format CSV, atribut-atribut tersebut yaitu nama saluran, jumlah pelanggan, deskripsi saluran, jumlah penonton, negara, url foto profil, url foto sampul, kata kunci, dan beberapa judul video-video yang terakhir diunggah. Gambaran dari data CSV yang berisi atribut-atribut dari masing-masing saluran dapat dilihat pada Gambar 19.



| | Unnamed: 0 | channelName | subscriberCount | description | viewCount | country | avatarUrl | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | Rizky Wahdani | 2950.0 | Assalamu'alaikum.\nSelamat datang di channel s... | 2.355180e+05 | ID | https://yt3.ggpht.com/ytc/AKedOLTdMtBwTSDzymgS:... | https://yt3.ggpht.co |
| 1 | 1 | Ahlul Kisa Tegal | NaN | NaN | 1.046260e+05 | ID | https://yt3.ggpht.com/of2oRUgKcql3Li6lqqzN7UiG:... | https://yt3.ggpht.co |
| 2 | 2 | New Channel JENGGO ERPAH | NaN | pada kesempatan ini channel kami sedang membah... | 5.949160e+05 | ID | https://yt3.ggpht.com/KPztquO6l3zmCHoHXA4R4Ql9... | https://yt3.ggpht.co |
| 3 | 3 | Trisno MathNian | 2660.0 | Belajar matematika dan berbagi kegiatan sehari... | 2.417180e+05 | ID | https://yt3.ggpht.com/ytc/AKedOLTcMwYDMMJlSKsa... | https://yt3.ggpht.co |
| 4 | 4 | Renata Official | 1390.0 | Selamat datang di Saluran Kami konten berisi k... | 1.168200e+04 | ID | https://yt3.ggpht.com/WWrqm155Eg0lMcrGV2Hu2QxT... | https://yt3.ggpht.co |

Gambar 19. Data Saluran Youtube Edukasi

Data yang dikumpulkan melalui *web scraping* tersebut terdiri dari saluran-saluran Youtube bertema edukasi yang memiliki tema yang beragam serta jumlah pelanggan yang beragam pula. Hal tersebut sangat baik, mengingat manfaat penelitian ini juga ingin merekomendasikan saluran-saluran edukasi yang belum populer dan memiliki jumlah pelanggan yang sedikit. Ragam jumlah pelanggan dari saluran-saluran Youtube yang terkumpul berupa grafik dapat dilihat pada Gambar 20.




Gambar 20. Grafik Sebaran Jumlah Pelanggan Pada *Dataset*

Gambar 20 menunjukkan bahwa data saluran-saluran yang digunakan sangat beragam dari segi kepopuleran saluran berdasarkan jumlah pelanggan yang dimiliki oleh masing-masing saluran. Bahkan, saluran yang mendominasi pada *dataset* yang digunakan merupakan saluran dengan jumlah pelanggan dari 10001 hingga 50000.

### 5.1.2. Preprocessing

Tahap *preprocessing* dibagi kembali menjadi beberapa tahapan. Tahapan-tahapan yang ada pada proses *preprocessing* meliputi tahap *merging, translating, tokenizing, case folding,* dan *filtering.* Berikut adalah contoh hasil dari tahap *preprocessing* yang dilakukan pada sebuah saluran, dapat dilihat pada Gambar 21.

| description | keywords | videoTitle | cleaned |
|---|---|---|---|
| Neuron adalah media edukasi seputar fakta, berita, dan apapun yang berkaitan dengan dunia kesehatan. Untuk menghubungi demi kepentingan apapun, email ke: neuron.animation@gmail.com Kamu juga dapat mendukung Neuron (dan Hipotesa) melalui KaryaKarsa di: https://karyakarsa.com/neuronmedia Merchandise kami dapat dibeli di https://www.tokome.id/neuronmedia | "animasi edukasi" animasi edukasi kedokteran kesehatan "informasi kesehatan" | Apakah Gigi Bungsu Harus Dicabut? Apa itu Gangguan Depersonalisasi-Derealisasi? Apa itu Sexual Fetish? Dan Dari Mana Datangnya? Kembar yang Terpisah \| Apakah Gen lebih berpengaruh daripada Lingkungan? Bagaimana Kerokan Dapat Menyembuhkan Penyakit? Mengapa Anda Tidak Berjenggot Bagaimana Epilepsi Bisa Terjadi? The Hidden Risks of LASIK Surgery What Is Amnesia? Secrets from Acne-Free Communities What is Sleep Paralysis? Should We Give Up Masturbation? What is a Psychopath? What are Phobias? What is Body Dysmorphia? Mengapa Tes Perawan Tidak Masuk Akal What is Anxiety Disorder? What is ADHD? Dying to Save The World What is Obsessive-Compulsive Disorder (OCD) ? What is Post-Traumatic Stress Disorder (PTSD)? The 4 Most Dangerous Covid Variants Right Now How Zoom is Making You Sick What is Autism? Are We Born Smart? What is Dissociative Identity Disorder (DID)? Vasectomy and Sterilization \| Understanding Contraceptives Part 3 What is Bipolar Disorder? Menggunakan Hormon untuk Mencegah Kehamilan \| Mengenal Kontrasepsi Part 2 What is the Difference Between These 5 Corona Vaccines? | neuron educational medium news facts anything related world health contact interest email neuron animation gmail com also support neurons hypotheses work https karyakarsa com neuronmedia merchandise purchased https www tokome id neuronmedia animation educational animation health education health health information whether wisdom teeth revoked depersonalization disorder sexual fetish twins come separately whether genes influential environment scrapings cure diseases bearded epilepsy occur hidden risks lasik surgery amnesia secrets acne free communities sleep paralysis give masturbation psychopath phobias body dysmorphia mengapa tes perawan tidak masuk akal anxiety disorder adhd dying save world obsessive compulsive disorder ocd post traumatic stress disorder ptsd 4 angerous covid variants right zoom making sick sick autism born smart dissociative identity disorder vasectomy sterilization understanding contraceptives part 3 bipolar disorder using hormones prevent pregnancy knowing contraception part 2 difference 5 corona vaccines |

Gambar 21. Hasil Proses *Preprocessing*

Tahapan ini adalah tahapan yang akan menghasilkan sebuah baris baru berupa kata-kata yang sudah bersih yang akan menjadi kamus kata pada proses *training word embeddings*. Kata-kata yang akan dibersihkan dan dijadikan kamus kata adalah kata-kata yang berasal dari atribut deskripsi, *keywords* saluran, dan sampel judul video.

### 5.1.3. Pelatihan Model Word Embeddings

Tahap implementasi *training word embeddings* tepat dilakukan setelah proses *preprocessing*. Atribut yang telah melalui proses *preprocessing* akan dijadikan input berupa kamus kata untuk dilakukan *training word embeddings* dengan arsitektur *neural network*.

Kamus kata yang dijadikan input pada proses *training word embeddings* adalah kata-kata dari beberapa atribut yang ada pada *dataset* yang telah dilakukan *preprocessing*. Hal ini dilakukan agar model dapat mengerti kata-kata spesifik dari tema tertentu yang hanya ada pada *dataset* sehingga sistem rekomendasi dapat mengetahui kesamaan beberapa kata yang spesifik hanya terdapat pada *dataset* yang digunakan. Pada penelitian ini, terdapat sekitar 100000 kata yang ada pada kamus

kata yang akan menjadi input pada proses pelatihan *neural network word embeddings*.

Proses selanjutnya adalah langsung ke proses inti dari penelitian ini, yakni proses *training neural network word embeddings* dengan menggunakan *Word2Vec* untuk menghasilkan representasi vektor berdimensi tinggi dari setiap kata yang terdapat pada kamus kata. Pada proses *training word embeddings,* ditentukan juga beberapa parameter yang akan menjadi pembeda dan akan berpengaruh terhadap hasil evaluasi. Pada penelitian ini, dilakukan beberapa kali proses *training* dengan parameter berbeda-beda, hal ini bertujuan untuk menemukan model terbaik yang akan digunakan pada sistem rekomendasi setelahnya. Parameter-parameter yang akan dijadikan acuan pada proses *training* adalah sebagai berikut :

a. Algoritma : Digunakan 2 algoritma berbeda yang akan diterapkan pada proses *training,* yakni algoritma CBOW dan *Skip-Gram*
b. Jumlah Dimensi : Digunakan 3 jumlah dimensi dari masing-masing kata berbeda yang akan diterapkan pada proses *training*, yakni 100,200, dan 300 jumlah dimensi
c. *Window Size* : Digunakan 4 nilai *window size* yang akan diterapkan pada proses *training,* yakni 2,5,7, dan 10 nilai *window size*.

Proses *training* akan dilakukan sebanyak 2*3*4 kali atau sebanyak 24 kali dengan nilai parameter berbeda-beda berdasarkan keterangan tersebut.

### 5.1.4. Evaluasi Model *Word Embeddings*

Setiap kali proses *training* selesai dilakukan, maka model berupa representasi vektor akan dilakukan perhitungan evaluasi dengan menggunakan *Pearson Correlation* yang akan menghitung nilai korelasi dengan data *benchmark* bernama *Simlex-999* yang merupakan data *benchmark* yang dibuat dan diteliti oleh Felix Hill yang melibatkan sekitar 500 orang di Britania Raya untuk menentukan nilai kesamaan dari beberapa kata dalam bahasa Inggris. *Simlex-999* terdiri dari 999 kata dalam bahasa Inggris beserta nilai kesamaannya yang didapat dari 500 orang responden tersebut, data kesamaan tersebut akan dibandingkan dengan data model yang telah dilatih pada proses *word embeddings*. Contoh sampel data berupa nilai kesamaan dari beberapa kata pada *Simlex-999* dapat dilihat pada Tabel 8.

Tabel 8. Sampel data pada *Simlex-999*

| | | |
|---|---|---|
| *Old* | *New* | 1.58 |
| *Smart* | *Intelligent* | 9.2 |
| *Big* | *Broad* | 6.73 |
| *New* | *Ancient* | 0.23 |
| *School* | *Grade* | 4.42 |

Data *Simlex-999* memberikan nilai kesamaan semantik yang cukup besar pada kata-kata seperti kata sinonim ataupun pada dua kata yang memiliki kesamaan semantik. Sebaliknya, *Simlex-999* tidak memberi nilai yang besar pada kata-kata yang memiliki makna berlawanan. Hal ini memang sangat dibutuhkan pada penelitian ini agar sistem dapat menganggap sama dua saluran yang memiliki makna kesamaan secara menyeluruh dan tidak akan menganggap sama dua saluran yang memiliki data dengan kata-kata yang memiliki makna berlawanan.

Proses evaluasi akan menghitung nilai kedekatan yang didapat dari beberapa pasang kata hasil proses *training* dengan nilai kedekatan dari beberapa pasang kata

yang ada pada *Simlex-999* mengacu pada persamaan (2.2). Misalnya, nilai kedekatan yang didapat dari proses pelatihan terhadap model dengan algoritma *skip-gram*, jumlah dimensi vektor 100 dan jumlah *window size* 5 pada lima kata di tabel 8 adalah 2.4 pada pasangan kata *old* dan *new*, 8.71 pada pasangan kata *smart* dan *intelligence*, 5.13 pada pasangan kata *big* dan *broad*, 1.17 pada pasangan kata *new* dan *ancient*, serta 5.64 pada pasangan kata *school* dan *grade*. Mengacu pada persamaan (2.2), maka nilai tersebut akan dihitung sebagai berikut.

$$r = \frac{5*(23.05*22.16) - (23.05)(22.16)}{\sqrt{((5*141.12)-(23.05)^2)((5*152)-(22.16)^2)}} = 0.26322$$

Proses *training* model *word embeddings* dilakukan sebanyak 24 kali dengan parameter yang berbeda-beda, Tabel 9 adalah tabel yang memperlihatkan nilai dari *Pearson Correlation* dari seluruh model yang telah dilakukan proses pelatihan.

Tabel 9. Hasil *Pearson Correlation* model *word embeddings*

| **Algoritma** | **Jumlah Dimensi Vektor** | ***Window Size*** | | | |
|---|---|---|---|---|---|
| | | **2** | **5** | **7** | **10** |
| ***Skip Gram*** | **100** | 0.263224177 | 0.255381088 | 0.243202750 | 0.202387573 |
| | **200** | 0.283415676 | 0.263165571 | 0.262463068 | 0.234886013 |
| | **300** | 0.286847576 | **0.303474794** | 0.264772695 | 0.253895858 |
| **CBOW** | **100** | 0.212183079 | 0.213740622 | 0.210370121 | 0.197188209 |
| | **200** | 0.225560995 | 0.217191017 | 0.218051736 | 0.210520791 |
| | **300** | 0.219178875 | 0.227473140 | 0.219690857 | 0.198394230 |

Nilai *Pearson Correlation* yang didapat memiliki nilai yang sangat bervariasi, namun tidak terlalu berbeda jauh. Dari segi algoritma yang digunakan, terlihat bahwa model yang menggunakan *skip-gram* memiliki nilai yang lebih besar dari model yang menggunakan algoritma CBOW terlepas dari jumlah dimensi dan *window size* yang diterapkan. Dari segi jumlah dimensi vektor yang diterapkan, terlihat jelas bahwa semakin tinggi jumlah dimensi yang ditentukan maka semakin tinggi pula nilai *Pearson Correlation* yang didapat. Sedangkan dari segi jumlah *window size*, nilai *Pearson Correlation* paling tinggi didapat dari model dengan jumlah *window size* 2 ataupun 5, sedangkan model dengan jumlah *window size* 10 selalu mendapatkan nilai *Pearson Correlation* terkecil dibanding dengan model lainnya yang memiliki jumlah *window size* yang lebih kecil.

Model yang memiliki nilai *Pearson Correlation* terbesar adalah model yang menggunakan algoritma *skip-gram* dengan jumlah dimensi vektor berjumlah 300 dan jumlah *window size* bernilai 5 dengan nilai *Pearson Correlation* sebesar **0.30347479435806907**. Dari hasil tersebut, dapat ditentukan bahwa model tersebutlah yang akan lanjut ke proses selanjutnya untuk diterapkan pada sistem rekomendasi saluran Youtube edukasi pada proses selanjutnya.

### 5.1.5. Evaluasi Sistem Rekomendasi

Evaluasi sistem rekomendasi dilakukan dengan menggunakan *precision*. Terdapat tiga metode *precision* yang akan diukur pada proses evaluasi ini, yaitu

pengujian *Precision @K*, *Average Precision @K (AP@K)*, dan *Mean Average Precision @K (MAP@K)*. *Precision @K* merupakan metode yang mengukur nilai *precision* semua hasil rekomendasi dari *query* yang dicoba oleh partisipan, *Average Precision @K* akan mengukur nilai rata-rata *precision* dari masing-masing *query* yang dicoba, dan *Mean Average Precision @K* mengukur nilai rata-rata *AP@K* dari masing-masing partisipan. Sistem diuji terhadap empat partisipan dengan setiap partisipan akan menguji empat kali sistem dengan empat kueri yang berbeda, dua kali mencari rekomendasi berdasarkan nama saluran lain, dan dua kali berdasarkan kata kunci. Selain itu, sistem juga dilakukan pengujian dengan menggunakan kata kunci yang tidak berhubungan dengan edukasi, hal ini dilakukan untuk mengetahui bagaimana sistem menghadapi kasus tersebut, dan juga untuk mengetahui apakah data-data saluran yang dikumpulkan benar merupakan saluran-saluran Youtube yang sering mengunggah konten edukasi.

***Precision@K***

Berikut adalah hasil dari nilai *Precision@K* (nilai relevan pada top K hasil yang direkomendasikan dari jumlah K) mengacu pada persamaan (2.3), dapat dilihat pada Tabel 10.

Tabel 10. Tabel Hasil *Precision@K*

| **Partisipan** | ***Query*** | ***Ranking*** | **Keterangan** | **P@K** | **Nilai** |
|---|---|---|---|---|---|
| **Partisipan 1** | *freecodecamp.org* | 1 | Relevan | P@1 | 1 |
| | | …….. | …….. | …… | …… |
| | | 10 | Relevan | P@10 | 1 |
| | …… | …….. | …….. | …… | …… |
| | …… | …….. | …….. | …… | …… |
| | …… | …….. | …….. | …… | …… |
| **Partisipan 2** | *Web Programming UNPAS* | 1 | Relevan | P@1 | 1 |
| | | …….. | …….. | …… | …… |
| | | 10 | Tidak | P@10 | 0.80 |
| | …… | …….. | …….. | …… | …… |
| | …… | …….. | …….. | …… | …… |
| | …… | …….. | …….. | …… | …… |
| **Partisipan 3** | *Belajar edit foto* | 1 | Relevan | P@1 | 1 |
| | | …….. | …….. | …… | …… |
| | | 10 | Relevan | P@10 | 0.8 |
| | …… | …….. | …….. | …… | …… |
| | …… | …….. | …….. | …… | …… |
| | …… | …….. | …….. | …… | …… |

Tabel 10 menunjukkan tiga peringkat teratas sebagian besar memberikan hasil rekomendasi yang relevan. Beberapa nilai precision cukup konsisten pada peringkat teratas dan mulai menunjukkan penurunan pada peringkat 4 hingga peringkat 10. Walaupun ada beberapa *query* yang menunjukkan data tidak relevan pada peringkat 3, namun 2 teratas bisa dikatakan merupakan data yang sudah pasti merupakan saluran Youtube yang relevan dengan *query* yang diberikan.

***Average Precision@K (AP@K)***

Berikut adalah hasil dari nilai *Average Precision@K* mengacu pada persamaan (2.4) dari sistem yang diuji oleh tiga partisipan, dapat dilihat pada Tabel 11.

Tabel 11. Tabel Hasil *Average Precision@K*

| **Partisipan** | ***Query*** | **P@10** | **AP@4** |
|---|---|---|---|
| **Partisipan 1** | *freecodecamp.org* | 1 | 0.875 |
| | *Neuron* | 0.9 | |
| | *Belajar menggambar dan melukis* | 0.8 | |
| | *Olahraga dan Kesehatan* | 0.8 | |
| **Partisipan 2** | *Web Programming UNPAS* | 0.8 | 0.85 |
| | *GadgetIn* | 1 | |
| | *Data Science dan Machine Learning* | 0.8 | |
| | *Belajar bahasa Inggris* | 0.8 | |
| **Partisipan 3** | *Hipotesa* | 0.9 | 0.80 |
| | *Kelas Terbuka* | 0.7 | |
| | *Belajar Microsoft Excel* | 0.8 | |
| | *Belajar edit foto* | 0.8 | |

Tabel 11 menunjukkan nilai rata-rata paling rendah didapat oleh partisipan 3 dengan nilai rata-rata 0,80 dan tertinggi di angka 0,875 berdasarkan *query* yang ditentukan oleh partisipan 1. Dari Tabel 11 juga dapat dilihat bahwa *query* "*Kelas Terbuka*" memiliki nilai *P@10* yang paling rendah dari seluruh *query* yang dicoba, hal tersebut dapat diasumsikan karena saluran tersebut mengunggah konten dengan judul yang tidak fokus pada satu pembahasan, sehingga mempengaruhi hasil rekomendasi yang didapat. Dari tabel tersebut dapat disimpulkan bahwa nilai *AP@4* yang didapat oleh masing-masing partisipan sangatlah besar dengan nilai diatas 0.8.

***Mean Average Precision@K (MAP@K)***

Berikut adalah hasil dari nilai *Mean Average Precision@K* mengacu pada persamaan (2.5) dari sistem yang diuji oleh tiga partisipan, dapat dilihat pada Tabel 12.

Tabel 12. Tabel Hasil *Mean Average Precision@K*

| **Partisipan** | **AP@4** | **MAP@3** |
|---|---|---|
| **Partisipan 1** | 0.875 | 0.842 |
| **Partisipan 2** | 0.85 | |
| **Partisipan 3** | 0.80 | |

Tabel 12 menunjukkan bahwa dari tiga partisipan yang menguji sistem, hasil pengujian *precision* keseluruhan mendapatkan nilai 0,842 untuk hasil rekomendasi yang relevan terhadap *query* yang diberikan oleh masing-masing partisipan dan sisanya sebesar 0.158 tidak relevan terhadap *query*.

**Nilai Precision Pada Kata Kunci Non Edukasi**

Terdapat empat *query* yang merupakan kata kunci yang tidak memiliki hubungan dengan dunia edukasi. Berikut adalah nilai *precision* sistem pada pengujian terhadap kata kunci non edukasi, dapat dilihat pada Tabel 13.

Tabel 13. Nilai *precision* pada kata kunci non edukasi

| | | | | | |
|---|---|---|---|---|---|
| **Partisipan 4** | Sepakbola | 1 | Tidak | P@1 | 0 |
| | | …….. | …….. | …… | …… |
| | | 10 | Relevan | P@10 | 0.4 |
| | *Video Dancing* | 10 | Tidak | P@10 | 0.2 |
| | Drama Korea | 10 | Tidak | P@10 | 0.1 |
| | Musik rock | 10 | Tidak | P@10 | 0.3 |

Berikut adalah nilai rata-rata atau *average precision* dari pengujian pada kata kunci non edukasi, dapat dilihat pada Tabel 14.

Tabel 14. Nilai *Average Precision* pada pengujian kata kunci non edukasi

| **Partisipan** | ***Query*** | **P@10** | **AP@4** |
|---|---|---|---|
| **Partisipan 4** | Sepakbola | 0.4 | 0.25 |
| | *Video Dancing* | 0.2 | |
| | Drama Korea | 0.1 | |
| | Musik rock | 0.3 | |

Nilai rata-rata menunjukkan bahwa sistem tidak dapat bekerja dengan baik jika kata kunci yang diinputkan tidak berhubungan dengan dunia edukasi. Namun, hal itu juga berarti baik, karena secara tidak langsung mengisyaratkan bahwa data-data yang telah dikumpulkan hanyalah data-data berupa saluran Youtube yang berkaitan dengan edukasi saja, sehingga sistem tidak dapat menemukan saluran yang sesuai jika kata kunci yang diinputkan tidak berhubungan dengan edukasi.

### 5.1.6. Implementasi *Graphical User Interface* (GUI)

Sistem rekomendasi yang akan dibuat pada penelitian ini berupa sebuah website dengan beberapa tampilan *Graphical User Interface* (GUI) yang akan memudahkan pengguna ketika sedang menggunakan sistem. Sistem dibuat dengan menggunakan kerangka kerja *Flask* dan *React.JS* untuk bagian *frontend*.

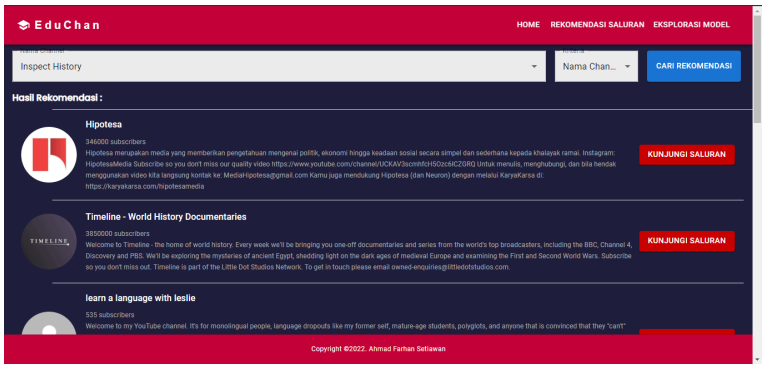
**Tampilan Halaman Utama**

Tampilan halaman utama sistem rekomendasi saluran Youtube edukasi ini merupakan tampilan yang menampilkan penjelasan mengenai metode yang digunakan pada sistem rekomendasi. Hasil tampilan halaman utama dapat dilihat pada Gambar 22.


Gambar 22. Tampilan Halaman Utama

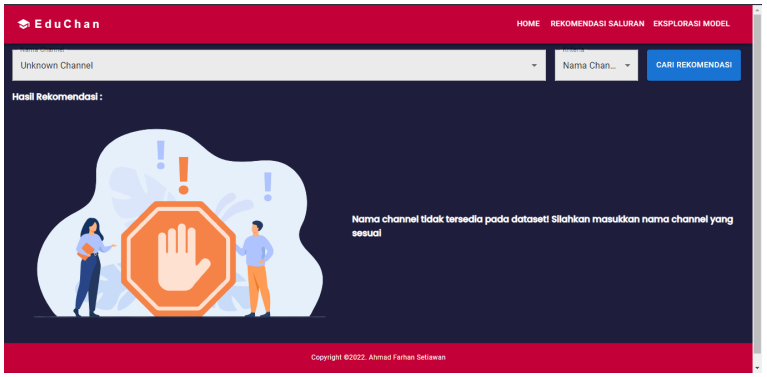
**Tampilan Halaman Rekomendasi**

Halaman rekomendasi merupakan halaman yang memungkinkan pengguna untuk menginputkan nama saluran Youtube edukasi atau kata kunci yang diminati oleh pengguna. Setelah pengguna menekan tombol "Cari Rekomendasi", maka sistem akan menampilkan hasil rekomendasi berdasarkan tingkat kemiripan dari setiap saluran Youtube edukasi yang ada pada dataset yang mengacu pada model *word embeddings*. Hasil tampilan halaman rekomendasi dapat dilihat pada Gambar 23.




Gambar 23. Tampilan Halaman Rekomendasi

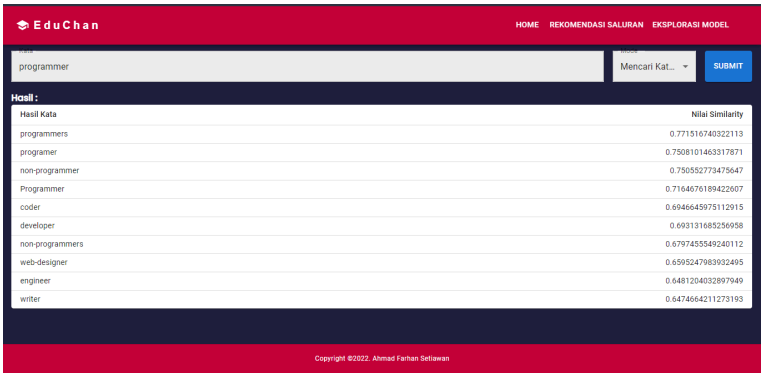
Jika nama saluran Youtube edukasi yang diinputkan tidak tersedia di dalam dataset, maka sistem akan memberikan pesan berupa peringatan seperti pada Gambar 24.




Gambar 24. Tampilan Pesan Error

**Tampilan Halaman Eksplorasi Model**

Halaman ini merupakan halaman tambahan yang memungkinkan pengguna untuk mengeksplorasi model *word embedding* yang digunakan pada sistem rekomendasi. Pada halaman ini, pengguna hanya tinggal menginputkan sebuah kata, lalu sistem akan otomatis memberikan daftar kata yang memiliki makna semantik terdekat dengan kata yang diinputkan. Jika kata yang diinputkan oleh pengguna merupakan kata yang tersedia pada model, maka akan keluar hasilnya yang bisa dilihat pada Gambar 25.




Gambar 25. Halaman Eksplorasi Model

## 5.2. Pembahasan

Sistem rekomendasi saluran Youtube edukasi ini merupakan sistem yang dapat merekomendasikan para pengguna Youtube yang menjadikan Youtube sebagai media pembelajaran untuk mencari beberapa rekomendasi berupa saluran Youtube lain yang memiliki konten serupa dengan saluran Youtube edukasi favoritnya ataupun yang memiliki konten serupa dengan kata kunci yang dimiliki. Sistem ini dibuat dengan menggunakan metode *Neural Network Word Embeddings* yang dapat menangkap kata serupa berdasarkan makna semantik. Pada sistem yang berbasis *website*, pengguna juga dapat mengeksplorasi model yang digunakan pada sistem. Dengan menggunakan model tersebut, sistem akan mencari saluran-saluran terdekat berdasarkan *query* yang diinputkan oleh pengguna.

Model *neural network word embeddings* yang digunakan pada sistem juga dapat dilakukan proses eksplorasi, terdapat beberapa hal menarik yang dapat dieksplorasi dari model yang akan digunakan, seperti mencari kata dengan kesamaan makna terdekat, melihat nilai kesamaan antara dua kata, dan perhitungan aritmatika antar kata. Berikut adalah hasil berupa daftar kata yang memiliki kesamaan semantik dari beberapa kata yang diinputkan, dapat dilihat pada Gambar 26.

```
find_word_similarity('math')
[('mathematics', 0.8272228240966797),
 ('maths', 0.8168162703514099),
 ('mathematic', 0.7626342177391052),
 ('math-science', 0.7625195980072021),
 ('math-', 0.7619606256484985),
 ('non-math', 0.7588698267936707),
 ('math-related', 0.748168408870697),
 ('math-based', 0.7443567514419556),
 ('mathy', 0.7362400889396667),
 ('science', 0.7097405195236206)]

find_word_similarity('laptop')
[('laptops', 0.8158079385757446),
 ('laptop.', 0.7200470566749573),
 ('Laptop', 0.7093129754066467),
 ('lap-top', 0.703281581401825),
 ('palmtop', 0.703200101852417),
 ('laptop-sized', 0.6984363198280334),
 ('desktop', 0.6970775127410889),
 ('computer', 0.6961240172386169),
 ('netbook', 0.6835439205169678),
 ('notebook', 0.6819139122962952)]

find_word_similarity('indonesia')
[('indonesian', 0.7991378307342529),
 ('indonesians', 0.73257976770401),
 ('malaysia', 0.7202091217041016),
 ('indo', 0.6834501624107361),
 ('australia', 0.672518253326416),
 ('singapore', 0.6692454814910889),
 ('Indonesia', 0.66309654712677),
 ('jakarta', 0.6613821387290955),
 ('Indonesia.', 0.6556822061538696),
 ('cambodia', 0.628200113773346)]

find_word_similarity('blockchain')
[('blockchains', 0.8700655698776245),
 ('Blockchain', 0.8020414710044861),
 ('blockchain-based', 0.7852085828781128),
 ('cryptocurrency', 0.712134599685669),
 ('bitcoin', 0.6961855292320251),
 ('Bitcoin', 0.6618550419807434),
 ('cryptocurrencies', 0.6430795192718506),
 ('Blockchain.info', 0.6398552656173706),
 ('crypto-currency', 0.623888373374939),
 ('altcoin', 0.6205140352249146)]
```

Gambar 26. Hasil Kesamaan Kata Semantik

Beberapa kata telah dicoba, terlihat bahwa model dapat memberikan hasil berupa kata yang memiliki makna semantik dengan cukup memuaskan. Contohnya, model dapat menganggap bahwa kata "*math*" dan "*science*" memiliki nilai kesamaan yang besar, begitu juga pada kata "*laptop*" dengan "*computer*", "indonesia" dengan "malaysia" dan "*blockchain*" dengan "*cryptocurrency*"

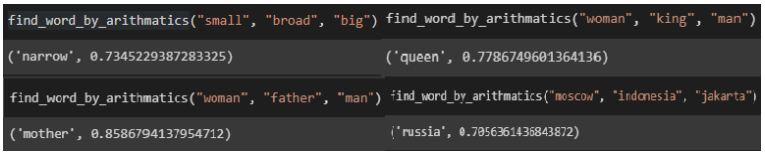
Model *word embeddings* yang telah dilatih juga memiliki fungsi untuk melihat nilai kesamaan antara dua kata. Dengan fungsi tersebut, dapat diketahui nilai kesamaan antara dua buah kata berbeda. Berikut adalah hasil berupa nilai kesamaan dari dua buah kata yang diinputkan, dapat dilihat pada Gambar 27.

```
count_similarity_between_words('html', 'css')
0.6768152

count_similarity_between_words('mouse', 'keyboard')
0.6149152

count_similarity_between_words('windows', 'linux')
0.5049116

count_similarity_between_words('smart', 'clever')
0.78505135
```

Gambar 27. Hasil Nilai Kesamaan Antara Dua Kata

Model juga dapat menilai kesamaan makna dengan performa yang cukup baik. Dari Gambar 27, dapat dilihat bahwa model dapat menganggap sama dua kata sinonim seperti "*smart*" dengan "*clever*" karena memiliki nilai kesamaan yang besar, dan model juga dapat menangkap kesamaan dari dua kata dengan kesamaan konteks seperti "html" dengan "css", "*mouse*" dengan "*keyboard*", dan "*windows*" dengan "*linux*" karena memiliki nilai kesamaan diatas 0.5.

Model *word embeddings* dengan menggunakan *word2vec* merupakan model yang juga dapat dilakukan fungsi perhitungan aritmatika. Dengan fungsi ini, dapat diketahui hasil kata berdasarkan pertambahan maupun pengurangan dari beberapa kata. Berikut adalah hasil berupa kata dari beberapa perhitungan aritmatika, dapat dilihat pada Gambar 28.

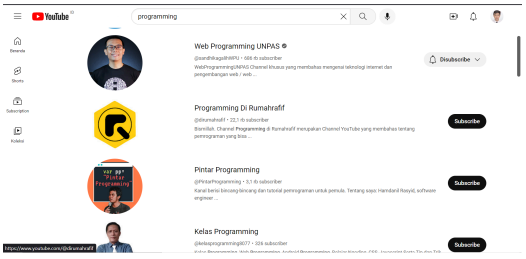
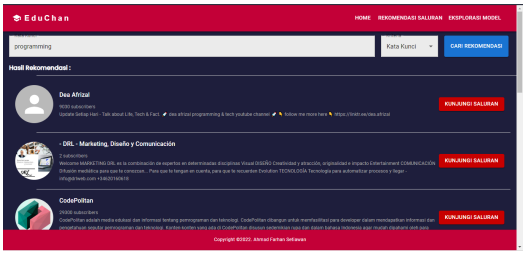
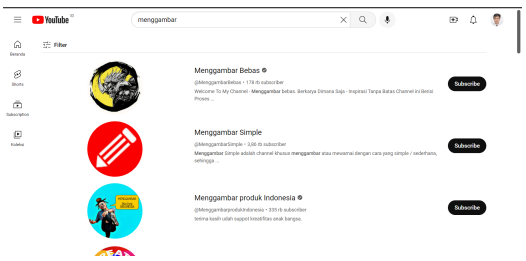
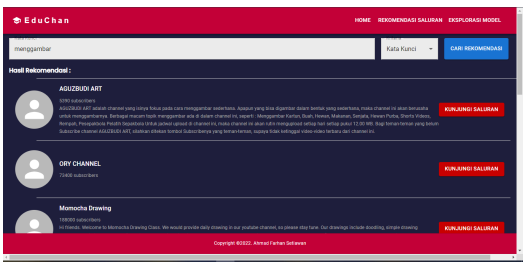



Gambar 28. Hasil Model Berdasarkan Perhitungan Aritmatika

Model ternyata sangat baik ketika melakukan perhitungan aritmatika dari beberapa kata. Perhitungan aritmatika ini mengisyaratkan jika jarak antara kata parameter kedua dan ketiga dengan kata parameter satu dan output memiliki jarak yang hampir sama atau dengan kata lain, model dapat menangkap hubungan antar kata berdasarkan sebuah kriteria dengan baik. Seperti contoh pada Gambar 28, model dapat menangkap hubungan antara jenis kelamin dengan jabatan pada "*king*" dan "*man*" yang memiliki nilai kesamaan yang hampir serupa dengan "*queen*" dan "*woman*", begitu juga dengan hubungan negara dan ibu kota seperti "*Indonesia*" dengan "*Jakarta*" yang memiliki nilai kesamaan yang hampir serupa dengan "*Russia*" dan "*Moscow*".

Sistem rekomendasi saluran Youtube edukasi ini juga terbukti dapat merekomendasikan beberapa data saluran serupa sesuai yang diinputkan oleh pengguna dengan melihat kesesuaian isi atau konten dari masing-masing saluran secara semantik, atau dengan kata lain sistem ini dapat merekomendasikan beberapa saluran kepada pengguna tidak hanya berdasarkan saluran-saluran yang memiliki padanan kata sesuai dengan kata kunci yang diinputkan pengguna seperti yang dilakukan oleh aplikasi Youtube saat ini. Berikut adalah perbandingan hasil dari sistem rekomendasi saluran yang dimiliki oleh Youtube dengan sistem yang dibuat pada penelitian ini, dapat dilihat pada Tabel 15.

Tabel 15. Perbandingan Hasil Rekomendasi Youtube Dengan Sistem Yang Dibuat

| **Youtube** | **Sistem Yang Dibuat** |
|---|---|
|  |  |
|  |  |



Pada tabel perbandingan tersebut, terbukti bahwa Youtube hanya merekomendasikan saluran-saluran yang namanya mengandung padanan kata yang diinputkan oleh pengguna. Hal tersebut tidak buruk, namun alangkah baiknya

Youtube juga dapat merekomendasikan saluran-saluran serupa dengan memperhatikan isi konten secara semantik agar saluran-saluran yang memiliki konten sesuai yang diinputkan oleh pengguna namun namanya tidak mengandung padanan kata khusus juga dapat terekomendasi dengan baik agar dapat dikenal oleh lebih banyak pengguna.

Selain itu, penelitian ini juga menghasilkan nilai *precision* yang sedikit lebih besar dibanding dengan penelitian-penelitian terdahulu yang juga menggunakan metode *neural network word embeddings* dengan model *word2vec.* Tabel perbandingan nilai *precision* dengan penelitian terdahulu dapat dilihat pada Tabel 16.

Tabel 16. Perbandingan Nilai *Precision* Dengan Penelitian Terdahulu

| **Peneliti** | **Judul** | **Nilai Precision** |
|---|---|---|
| (Aldiansyah, 2021) | Sistem Rekomendasi Lagu *Cross Language* Berdasarkan Lirik Menggunakan *word2vec*. | 0.388 |
| (Laili, 2019) | Sistem Rekomendasi Film Berdasarkan Sinopsis Menggunakan Metode *word2vec* | 0.726 |
| (Farhan, 2022) | Sistem Rekomendasi Saluran Youtube Edukasi Secara Semantik Menggunakan Neural Network Word Embeddings | 0.842 |

Nilai *precision* yang cenderung lebih besar dari penelitian-penelitian terdahulu bisa disebabkan oleh beberapa faktor, faktor pertama yang mempengaruhi adalah karena data yang digunakan oleh penelitian ini lebih banyak dari penelitian-penelitian terdahulu, karena data yang banyak berarti jumlah kata yang dijadikan input untuk proses pelatihan *word embeddings* juga lebih banyak, hal itu dapat menyebabkan sistem dapat lebih mengerti dengan baik makna semantik atau hubungan dari masing-masing kata secara lebih luas.

Faktor selanjutnya adalah karena pada penelitian ini model *word embeddings* yang digunakan adalah model dengan nilai *pearson correlation* terbesar atau model terbaik dari sekian banyak model yang dilatih. Oleh karena itu sistem pada penelitian ini memiliki nilai *pearson correlation* yang lebih besar dibanding sistem yang dibuat pada penelitian terdahulu yang menggunakan model yang tidak melalui proses uji coba dengan berbagai parameter yang ditentukan untuk didapat model dengan performa terbaik.

# BAB VI
# KESIMPULAN DAN SARAN

## 6.1. Kesimpulan

Sistem rekomendasi saluran Youtube edukasi secara semantik menggunakan metode *neural network word embeddings* dengan menggunakan metode *neural network word embeddings* dapat memberikan rekomendasi yang relevan kepada pengguna sesuai dengan kemiripan makna semantik dari masing-masing kata berdasarkan model terbaik yang didapat dari proses pengujian.

Pengujian model *word embeddings* dilakukan terhadap 24 model dengan parameter yang berbeda-beda dengan cara melihat nilai *Pearson Correlation* yang didapat dengan membandingkan model dengan data *benchmark* bernama *Simlex-999*. Pada proses pengujian, model yang menggunakan algoritma *Skip-Gram* memiliki nilai *Pearson Correlation* yang lebih besar dari model yang menggunakan algoritma CBOW, model yang memiliki jumlah dimensi banyak juga lebih unggul dari model dengan jumlah dimensi lebih kecil, serta model yang menggunakan parameter *window size* lebih kecil ternyata lebih unggul dari model dengan *window size* lebih besar. Hasilnya, model yang dilatih menggunakan *Skip-Gram* dengan jumlah dimensi 300 dan *window size* 5 menjadi model terbaik mengalahkan model-model lainnya dengan nilai *Pearson Correlation* 0.30347479435806907.

Sistem rekomendasi diuji terhadap 3 partisipan dengan masing-masing partisipan menguji sistem dengan 4 *query* yang berbeda. Sistem dapat memberi hasil rekomendasi yang relevan pada peringkat 3 teratas dan akan mulai menghasilkan data tidak relevan pada peringkat 4 kebawah, meski ada beberapa kasus yang menunjukkan bahwa sistem tidak merekomendasikan data yang relevan pada peringkat 3. Dari nilai *AP@K* dari masing-masing partisipan, didapat nilai *precision* keseluruhan (*MAP@K*) bernilai 0.842 atau dengan kata lain, sistem dapat merekomendasikan 84.2% data yang relevan. Dilakukan juga pengujian dengan menginputkan kata kunci yang tidak berhubungan dengan edukasi, didapat hasil 0.25 atau sistem hanya dapat merekomendasikan 25% data relevan jika kata kunci yang diinputkan tidak berhubungan dengan edukasi.

## 6.2. Saran

Berdasarkan penelitian yang telah dilakukan tersebut, terdapat beberapa saran yang dapat dipertimbangkan untuk penelitian selanjutnya, yaitu:

a. Data yang digunakan pada sistem dapat ditingkatkan dari segi jumlah agar kata yang ada pada kamus kata yang dijadikan input pada *word embeddings* lebih banyak, sehingga akan menghasilkan model dengan performa yang lebih bagus dalam menangkap makna semantik dari masing-masing kata.
b. Pengujian model *word embeddings* dilakukan lebih banyak lagi dengan cakupan parameter yang lebih luas lagi seperti penggunaan nilai *dimension size* dan *window size* yang lebih banyak agar dapat mengetahui hubungan antara parameter yang digunakan dengan performa model secara komprehensif.

# DAFTAR PUSTAKA


**Al-Ghuribi, S. M., & Noah, S. A. M**. 2019. *Multi-criteria review-based recommender system–the state of the art*. IEEE Access **7** : 169446-169468.

**Aldiansyah, G. W., Adikara, P. P., & Wihandika, R. C.** 2019. Rekomendasi Lagu Cross Language Berdasarkan Lirik Menggunakan Word2VEC. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN, **2548**, 964X.

**Burke, R., Felfernig, A., & Göker, M. H**. 2018. *Recommender systems: An overview. Ai Magazine*, ***32*(3)** : 13-18.

**Camacho-Collados, J., & Pilehvar, M. T**. 2017. *On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis*. arXiv preprint arXiv:1707.01780.

**Cotterell, R., & Schütze, H**. 2019. *Morphological word embeddings*. arXiv preprint arXiv : 1907.02423.

**Creators, Youtube**. 2017. *Ready To Make Educational Content On Youtube?*. https://www.youtube.com/watch?v=-FcWxXdq2lk . 16 Mei 2022.

**Di Gennaro, G., Buonanno, A., & Palmieri, F. A**. 2021. *Considerations about learning Word2Vec*. The Journal of Supercomputing, **77(11)**, 12320-12335.

**Fócil-Arias, C., Ziiniga, J., Sidorov, G., Batyrshin, I., & Gelbukh, A**. 2017. *A tweets classifier based on cosine similarity*. Conference and Labs of the Evaluation Forum, (pp. 11-14).

**Glen**, **Stephanie**. 2021. *Correlation Coefficient: Simple Definition, Formula, Easy Steps*. https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/ . 28 Juni 2022.

**Hashim, H**. 2018. *Application of technology in the digital era education*. International Journal of Research in Counseling and Education, ***2*(1)** : 1-5.

**Hellrich, J**. 2019. *Word embeddings: reliability & semantic change*. IOS Press : 347.

**Jansen, S**. 2017. *Word and phrase translation with word2vec*. arXiv preprint arXiv:1705.03127.

**Kadhim, A. I**. 2018. *An evaluation of preprocessing techniques for text classification*. International Journal of Computer Science and Information Security (IJCSIS), **16(6)**, 22-32.

**Khufa, Bryan** 2021. Sistem Rekomendasi pada Forum Kesehatan dengan Pemeringkatan Pertanyaan Serupa Menggunakan Pendekatan Deep Learning. *The Journal on Machine Learning and Computational Intelligence (JMLCI)* 1.1.

**Kinsley, H., Kukiela, D**. 2020. *Neural Network From Scratch In Python*. Kinsley, Szczecin.

**Nugroho, Fajar**. 2019. Sistem Rekomendasi Kata Kunci Untuk Website Menggunakan Hybrid Semantic Relatedness Dan Associative Neural Network.. Skripsi. Universitas Komputer Indonesia, Bandung.

**Perone, C. S., Silveira, R., & Paula, T. S**. 2018. *Evaluation of sentence embeddings in downstream and linguistic probing tasks*. arXiv preprint arXiv : 1806.06259.

**Semrush**. 2021. *Top 100 : The Most Visited Website In The US [2021 Top Website Edition]*. https://www.semrush.com/blog/most-visited-websites/ . 16 Mei 2022.

**Silveira, T., Zhang, M., Lin, X., Liu, Y., & Ma, S**. 2019. *How good your recommender system is? A survey on evaluations in recommendation*. International Journal of Machine Learning and Cybernetics, **10(5)**, 813-831.

**Tarnowska, K., Ras, Z. W., & Daniel, L**. 2020. *Recommender system for improving customer loyalty*. Springer International Publishing.

**Taylor, Michael**. 2017. *Neural Network : A Visual Introduction For Beginners*. Blue Windmill Media, Vancouver.

**Yin, Z., & Shen, Y**. 2018. *On the dimensionality of word embedding*. Advances in neural information processing systems : 31.

**Zhao, B**. 2017. *Web scraping*. Encyclopedia of big data, 1-3.

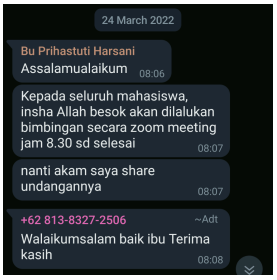
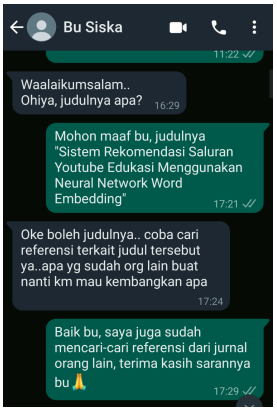
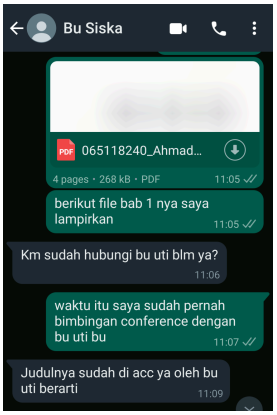
**Zhang, S., Yao, L., Sun, A., & Tay, Y**. 2018. *Deep Learning based Recommender System: A Survey and New Perspectives*. ACM Computing Surveys: 1-35.

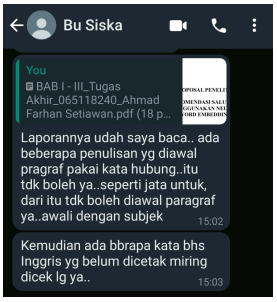
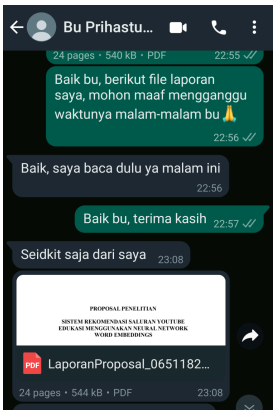
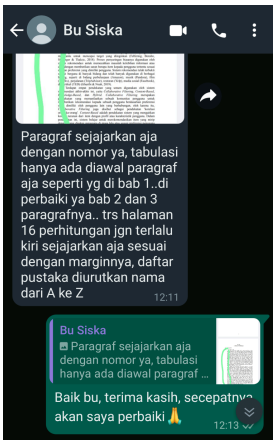
# LAMPIRAN

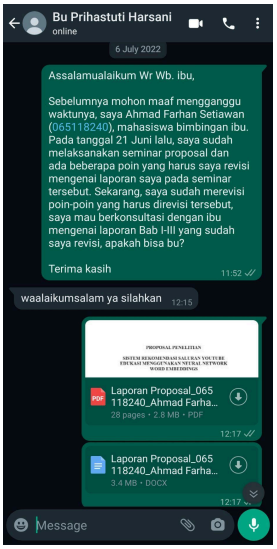
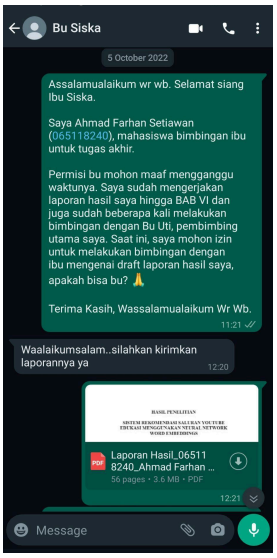
Lampiran 1. Kartu Bimbingan Mahasiswa

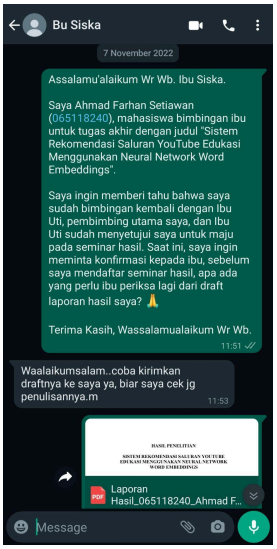
## Kartu Bimbingan Mahasiswa
Program Studi Ilmu Komputer - FMIPA UNPAK

Nama Mahasiswa : Ahmad Farhan Setiawan
NPM : 065118240
Judul Skripsi : Sistem Rekomendasi Saluran Youtube Edukasi Secara Semantik Menggunakan Neural Network Word Embeddings
Pembimbing 1 : Dr. Prihastuti Harsani, M.Si
Pembimbing 2 : Siska Andriani, M.Kom.

| No. | Tanggal | Dosen Pembimbing | Catatan | Bukti Bimbingan |
|---|---|---|---|---|
| 1. | 25 Maret 2022 | Dr. Prihastuti Harsani, M.Si | Persetujuan judul dan bab 1 |  |
| 2. | 26 April 2022 | Siska Andriani, M.Kom. | Pengajuan judul |  |
| 3. | 17 Mei 2022 | Siska Andriani, M.Kom. | Penyerahan bab 1 |  |

| | | | | |
|---|---|---|---|---|
| 4. | 3 Juni 2022 | Siska Andriani, M.Kom | Revisi bab 1, 2, 3 |  |
| 5. | 8 Juni 2022 | Dr. Prihastuti Harsani, M.Si | Revisi bab 1,2,3,4 |  |
| 6. | 15 Juni 2022 | Siska Andriani, M.Kom | Revisi Laporan Bab 1 - Daftar Pustaka |  |

| | | | | |
|---|---|---|---|---|
| 7. | 6 Juli 2022 | Dr. Prihastuti Harsani, M.Si. | Revisi Laporan Bab 1 - Bab 3 |  |
| 8. | 25 September 2022 | Dr. Prihastuti Harsani, M.Si. | Revisi Laporan Hasil |  |
| 9. | 5 Oktober 2022 | Siska Andriani, M.Kom. | Revisi Laporan Hasil |  |

| | | | | |
|---|---|---|---|---|
| 10. | 7 November 2022 | Siska Andriani, M.Kom. | Revisi Laporan Hasil |  |

Bogor, Januari 2023

Program Studi Ilmu Komputer
Fakultas MIPA UNPAK

Arie Qur'ania, M.Kom

Lampiran 2. Hasil *Precision* Lengkap

| **Partisipan** | ***Query*** | ***Ranking*** | **Keterangan** | **P@K** | **Nilai** |
|---|---|---|---|---|---|
| **Partisipan 1** | *Neuron* | 1 | Relevan | P@1 | 1 |
| | | 2 | Relevan | P@2 | 1 |
| | | 3 | Relevan | P@3 | 1 |
| | | 4 | Relevan | P@4 | 1 |
| | | 5 | Relevan | P@5 | 1 |
| | | 6 | Relevan | P@6 | 1 |
| | | 7 | Tidak | P@7 | 0.85 |
| | | 8 | Relevan | P@8 | 0.88 |
| | | 9 | Relevan | P@9 | 0.89 |
| | | 10 | Relevan | P@10 | 0.9 |
| | *freecodecamp.org* | 1 | Relevan | P@1 | 1 |
| | | 2 | Relevan | P@2 | 1 |
| | | 3 | Relevan | P@3 | 1 |
| | | 4 | Relevan | P@4 | 1 |
| | | 5 | Relevan | P@5 | 1 |
| | | 6 | Relevan | P@6 | 1 |
| | | 7 | Relevan | P@7 | 1 |
| | | 8 | Relevan | P@8 | 1 |
| | | 9 | Relevan | P@9 | 1 |
| | | 10 | Relevan | P@10 | 1 |
| | *Belajar menggambar dan melukis* | 1 | Relevan | P@1 | 1 |
| | | 2 | Relevan | P@2 | 1 |
| | | 3 | Relevan | P@3 | 1 |
| | | 4 | Tidak | P@4 | 0.75 |
| | | 5 | Relevan | P@5 | 0.8 |
| | | 6 | Tidak | P@6 | 0.67 |
| | | 7 | Relevan | P@7 | 0.71 |
| | | 8 | Relevan | P@8 | 0.75 |
| | | 9 | Relevan | P@9 | 0.78 |
| | | 10 | Relevan | P@10 | 0.8 |
| | *Olahraga dan kesehatan* | 1 | Relevan | P@1 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| | | 2 | Relevan | P@2 | 1 |
| | | 3 | Relevan | P@3 | 1 |
| | | 4 | Relevan | P@4 | 1 |
| | | 5 | Relevan | P@5 | 1 |
| | | 6 | Tidak | P@6 | 0.83 |
| | | 7 | Relevan | P@7 | 0.86 |
| | | 8 | Tidak | P@8 | 0.75 |
| | | 9 | Relevan | P@9 | 0.78 |
| | | 10 | Relevan | P@10 | 0.8 |
| **Partisipan 2** | *Web Programming UNPAS* | 1 | Relevan | P@1 | 1 |
| | | 2 | Relevan | P@2 | 1 |
| | | 3 | Relevan | P@3 | 1 |
| | | 4 | Relevan | P@4 | 1 |
| | | 5 | Relevan | P@5 | 1 |
| | | 6 | Relevan | P@6 | 1 |
| | | 7 | Relevan | P@7 | 1 |
| | | 8 | Relevan | P@8 | 1 |
| | | 9 | Tidak | P@9 | 0.89 |
| | | 10 | Tidak | P@10 | 0.80 |
| | *GadgetIn* | 1 | Relevan | P@1 | 1 |
| | | 2 | Relevan | P@2 | 1 |
| | | 3 | Relevan | P@3 | 1 |
| | | 4 | Relevan | P@4 | 1 |
| | | 5 | Relevan | P@5 | 1 |
| | | 6 | Relevan | P@6 | 1 |
| | | 7 | Relevan | P@7 | 1 |
| | | 8 | Relevan | P@8 | 1 |
| | | 9 | Relevan | P@9 | 1 |
| | | 10 | Relevan | P@10 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| | *Data Science dan Machine Learning* | 1 | Relevan | P@1 | 1 |
| | | 2 | Relevan | P@2 | 1 |
| | | 3 | Relevan | P@3 | 1 |
| | | 4 | Relevan | P@4 | 1 |
| | | 5 | Tidak | P@5 | 0.8 |
| | | 6 | Relevan | P@6 | 0.83 |
| | | 7 | Tidak | P@7 | 0.71 |
| | | 8 | Relevan | P@8 | 0.75 |
| | | 9 | Relevan | P@9 | 0.78 |
| | | 10 | Relevan | P@10 | 0.8 |
| | *Belajar bahasa Inggris* | 1 | Relevan | P@1 | 1 |
| | | 2 | Relevan | P@2 | 1 |
| | | 3 | Tidak | P@3 | 0.66 |
| | | 4 | Relevan | P@4 | 0.75 |
| | | 5 | Relevan | P@5 | 0.8 |
| | | 6 | Relevan | P@6 | 0.83 |
| | | 7 | Relevan | P@7 | 0.86 |
| | | 8 | Relevan | P@8 | 0.88 |
| | | 9 | Tidak | P@9 | 0.78 |
| | | 10 | Relevan | P@10 | 0.8 |
| **Partisipan 3** | *Hipotesa* | 1 | Relevan | P@1 | 1 |
| | | 2 | Relevan | P@2 | 1 |
| | | 3 | Relevan | P@3 | 1 |
| | | 4 | Relevan | P@4 | 1 |
| | | 5 | Tidak | P@5 | 0.8 |
| | | 6 | Relevan | P@6 | 0.83 |
| | | 7 | Relevan | P@7 | 0.86 |
| | | 8 | Relevan | P@8 | 0.88 |
| | | 9 | Relevan | P@9 | 0.89 |

| | | | | | |
|---|---|---|---|---|---|
| | | 10 | Relevan | P@10 | 0.9 |
| | *Kelas Terbuka* | 1 | Relevan | P@1 | 1 |
| | | 2 | Relevan | P@2 | 1 |
| | | 3 | Relevan | P@3 | 1 |
| | | 4 | Relevan | P@4 | 1 |
| | | 5 | Tidak | P@5 | 0.8 |
| | | 6 | Relevan | P@6 | 0.83 |
| | | 7 | Tidak | P@7 | 0.71 |
| | | 8 | Tidak | P@8 | 0.62 |
| | | 9 | Relevan | P@9 | 0.67 |
| | | 10 | Relevan | P@10 | 0.7 |
| | *Belajar Microsoft Excel* | 1 | Relevan | P@1 | 1 |
| | | 2 | Relevan | P@2 | 1 |
| | | 3 | Relevan | P@3 | 1 |
| | | 4 | Tidak | P@4 | 0.75 |
| | | 5 | Relevan | P@5 | 0.8 |
| | | 6 | Relevan | P@6 | 0.83 |
| | | 7 | Relevan | P@7 | 0.86 |
| | | 8 | Tidak | P@8 | 0.75 |
| | | 9 | Relevan | P@9 | 0.78 |
| | | 10 | Relevan | P@10 | 0.8 |
| | *Belajar edit foto* | 1 | Relevan | P@1 | 1 |
| | | 2 | Relevan | P@2 | 1 |
| | | 3 | Tidak | P@3 | 0.67 |
| | | 4 | Relevan | P@4 | 0.75 |
| | | 5 | Relevan | P@5 | 0.8 |
| | | 6 | Relevan | P@6 | 0.83 |
| | | 7 | Relevan | P@7 | 0.86 |
| | | 8 | Tidak | P@8 | 0.75 |

| | | | | | |
|---|---|---|---|---|---|
| | | 9 | Relevan | P@9 | 0.78 |
| | | 10 | Relevan | P@10 | 0.8 |

Lampiran 3. Hasil *precision* pada kata kunci non edukasi

| | | | | | |
|---|---|---|---|---|---|
| **Partisipan 4** | Sepakbola | 1 | Tidak | P@1 | 0 |
| | | 2 | Relevan | P@2 | 0.5 |
| | | 3 | Relevan | P@3 | 0.66 |
| | | 4 | Tidak | P@4 | 0.5 |
| | | 5 | Tidak | P@5 | 0.4 |
| | | 6 | Tidak | P@6 | 0.33 |
| | | 7 | Tidak | P@7 | 0.28 |
| | | 8 | Tidak | P@8 | 0.25 |
| | | 9 | Relevan | P@9 | 0.33 |
| | | 10 | Relevan | P@10 | 0.4 |
| | *Video Dancing* | 1 | Relevan | P@1 | 1 |
| | | 2 | Relevan | P@2 | 1 |
| | | 3 | Tidak | P@3 | 0.66 |
| | | 4 | Tidak | P@4 | 0.5 |
| | | 5 | Tidak | P@5 | 0.4 |
| | | 6 | Tidak | P@6 | 0.33 |
| | | 7 | Tidak | P@7 | 0.28 |
| | | 8 | Tidak | P@8 | 0.25 |
| | | 9 | Tidak | P@9 | 0.22 |
| | | 10 | Tidak | P@10 | 0.2 |
| | Drama Korea | 1 | Tidak | P@1 | 0 |
| | | 2 | Tidak | P@2 | 0 |
| | | 3 | Tidak | P@3 | 0 |
| | | 4 | Tidak | P@4 | 0 |
| | | 5 | Tidak | P@5 | 0 |
| | | 6 | Tidak | P@6 | 0 |

| | | | | | |
|---|---|---|---|---|---|
| | | 7 | Tidak | P@7 | 0 |
| | | 8 | Relevan | P@8 | 0.12 |
| | | 9 | Tidak | P@9 | 0.11 |
| | | 10 | Tidak | P@10 | 0.1 |
| | Musik rock | 1 | Relevan | P@1 | 1 |
| | | 2 | Tidak | P@2 | 0.5 |
| | | 3 | Tidak | P@3 | 0.33 |
| | | 4 | Relevan | P@4 | 0.5 |
| | | 5 | Relevan | P@5 | 0.6 |
| | | 6 | Tidak | P@6 | 0.5 |
| | | 7 | Tidak | P@7 | 0.42 |
| | | 8 | Tidak | P@8 | 0.37 |
| | | 9 | Tidak | P@9 | 0.33 |
| | | 10 | Tidak | P@10 | 0.3 |