Text Mining untuk Pengelompokan Skripsi di Prodi Pendidikan Informatika Universitas Trunojoyo Madura

Laili Cahyani[1], Muchamad Arif[2]

[1,2] *Prodi Pendidikan Informatika, Universitas Trunojoyo Madura, Indonesia*
email: [1]laili.cahyani@trunojoyo.ac.id, [2]arif@trunojoyo.ac.id

Abstrak

Skripsi merupakan karya tulis ilmiah mahasiswa sebagai syarat kelulusan atau perolehan gelar sarjana. Meskipun demikian, dosen juga berperan aktif dalam mengarahkan penentuan topik skripsi sebagai pembimbing. Skripsi yang ideal mengacu pada topik – topik yang up to date. Hendaknya skripsi juga mendukung keberhasilan rencana induk penelitian (RIP) Universitas bahkan rencana induk riset nasional (RIRN). Selain itu, topik skripsi harus selaras dengan bidang minat sesuai kurikulum program studi. Untuk itu, perlu adanya penyelarasan antara kondisi universitas, kebutuhan masyarakat, dan tujuan nasional. Sehingga, analisis data skripsi diperlukan untuk tujuan tersebut. Selama ini data dokumen skripsi di Program Studi Pendidikan Informatika Universitas Trunojoyo Madura belum terorganisir dengan baik di lingkup Program Studi. Sedangkan, jumlah data terus meningkat. Hal itu menjadi tantangan dalam pencarian dan penentuan topik skripsi sebagai bahan referensi selanjutnya. Selain itu, hingga saat ini belum dilakukan analisis terkait berkembangan skripsi yang sudah ada. Belum dilakukan juga pemetaan atau pengelompokan skripsi. Sehingga, dapat memberikan peluang adanya kemiripan skripsi. Oleh karena itu, dikembangkan sistem pengelompokan skripsi menggunakan text mining (studi kasus Program Studi Pendidikan Informatika Universitas Trunojoyo Madura). Dengan adanya sistem tersebut, diharapkan dapat membantu manajemen pengelolaan data skripsi bagi bidang skripsi di Program Studi. Sehingga dapat membantu dalam penentuan kebijakan dosen pembimbing dan meminimalisir peluang adanya kemiripan topik skripsi. Hasil penelitian ini menunjukkan bahwa metode clustering menggunakan K-Means dapat melakukan pengelompokan topic skripsi secara optimal dengan nilai akurasi sebesar 0,972972973, nilai presisi sebesar 916666667, nilai recall sebesar 0,9849199722, dan F-Measure sebesar 0,949199722 dalam skala 0 – 1.

Kata Kunci: Skripsi, Text Mining, Clustering, K-Means Clustering.

Abstract

Thesis is a student's scientific writing as a requirement for graduation or obtaining a bachelor's degree. However, the lecturer also plays an active role in directing the determination of the thesis topic as a supervisor. The ideal thesis refers to topics that are up to date. The thesis should also support the success of the university's research master plan (RIP) and even the national research master plan (RIRN). In addition, the thesis topic must be in line with the field of interest according to the curriculum of the study program. For this reason, there needs to be an alignment between university conditions, community needs, and national goals. Thus, thesis data analysis is needed for this purpose. So far, the thesis document data in the Informatics Education Study Program, Trunojoyo University, Madura has not been well organized in the scope of the Study Program. Meanwhile, the amount of data continues to increase. This becomes a challenge in finding and determining the thesis topic as the next reference material. In addition, until now there has not been an analysis related to the development of an existing thesis. The mapping or grouping of theses has not yet been carried out. So, it can provide an opportunity for the existence of similar scripts. Therefore, a thesis grouping system was developed using text mining (a case study of the Informatics Education Study Program, Trunojoyo University, Madura). With this system, it is hoped that it can help the management of thesis data management for the thesis field in the Study Program. So that it can help in determining the policy of the supervisor and minimize the chances of a similar thesis topic. The results of this study indicate that the clustering method using K-Means can optimally group thesis topics with an accuracy value of 0.972972973, a precision value of 916666667, a recall value of 0.9849199722, and an F-Measure of 0.949199722 on a scale of 0 – 1.

Keywords: thesis, clustering, K-means clustering, text mining.

PENDAHULUAN

Skripsi merupakan karya tulis ilmiah mahasiswa sebagai syarat kelulusan atau perolehan gelar sarjana. Meskipun demikian, dosen juga berperan aktif dalam mengarahkan penentuan topik skripsi sebagai pembimbing. Skripsi yang ideal mengacu pada topik – topik yang *up to date* (Mustikasari, 2017). Hendaknya skripsi juga mendukung keberhasilan rencana induk penelitian (RIP) Universitas bahkan rencana induk riset nasional (RIRN). Selain itu, topik skripsi harus selaras dengan bidang minat sesuai kurikulum program studi. Untuk itu, perlu adanya penyelarasan antara kondisi universitas, kebutuhan masyarakat, dan tujuan nasional. Sehingga, analisis data skripsi diperlukan untuk tujuan tersebut.

Berdasarkan observasi, selama ini data dokumen skripsi di Program Studi Pendidikan Informatika Fakultas Ilmu Pendidikan Universitas Trunojoyo Madura belum terorganisir dengan baik di lingkup Program Studi. Pengumpulan dokumen diserahkan secara manual ke program studi oleh mahasiswa yang telah menempuh skripsi. Sedangkan, lulusan dari tahun ke tahun semakin bertambah dan tentu berdampak pada semakin besarnya data skripsi yang terkumpul. Hal itu menjadi tantangan dalam pencarian dan penentuan topik skripsi sebagai bahan referensi selanjutnya. Selain itu, hingga saat ini belum dilakukan analisis terkait perkembangan skripsi yang sudah ada. Belum dilakukan juga pemetaan atau pengelompokan skripsi secara spesifik berdasarkan topik, permasalahan, data, materi, atau solusi yang diangkat. Sehingga, dapat memberikan peluang adanya kemiripan topik, permasalahan, data, materi, bahkan solusi yang diangkat.

Text mining merupakan salah satu teknologi yang dapat mengolah data teks untuk menghasilkan informasi tertentu. Dengan adanya *text mining*, data semi terstruktur maupun data yang tidak terstruktur dapat dikenali dengan baik. Dengan bertambahnya dokumen yang terkumpul dalam periode waktu yang cukup lama, maka *text mining* dibutuhkan untuk membantu dalam pengelolahan dan ekstraksi informasi penting yang dapat diperoleh (Salloum, Al-Emran, Monem, & Shaalan, 2018). *Text mining* dapat melakukan analisis, pengelompokan data, atau ekstraksi informasi dengan berbagai metode clustering (Allahyari et al., 2017).

Pada penelitian sebelumnya di tahun 2017, telah dilakukan pengelompokan skripsi menggunakan *self organizing maps clustering* (studi kasus: Prodi Teknik Informatika Universitas Nusantara PGRI Kediri) oleh Ika Zulaikah. Penelitian tersebut melakukan pengelompokan menggunakan text mining dengan metode SOM. Hasil menunjukkan bahwa metode SOM dapat digunakan untuk mengelompokkan skripsi dalam beberapa cluster (Zulaikah, 2017).

Penelitian lainnya pada tahun 2016, dilakukan oleh Lynda Rahmawati, dkk dengan judul Analisa Clustering Menggunakan Metode K-Means dan *Hierarchical Clustering* (Studi Kasus: Dokumen Skripsi Jurusan Kimia, FMIPA Universitas Sebelas Maret) (Rahmawati, Widya Sihwi, & Suryani, 2016). Penelitian tersebut melakukan pengelompokan menggunakan *text mining* dengan metode kombinasi antara *Hierarchical Clustering* dan *K-Means Clustering*. Hasil penelitian mendapatkan 16 cluster dokumen. Kemudian dianalisis keterkaitan antar dokumen di tiap cluster serta perkiraan temanya. Diperoleh juga keterkaitannya dengan dosen serta pengaruh dari keahlian dosen terhadap tema yang ada. Selain itu, jumlah suatu tema penelitian/skripsi juga terkait dengan minat mahasiswa serta adanya proyek dosen (Rahmawati et al., 2016).

Penelitian lain terkait pengelompokan skripsi mahasiswa dilakukan oleh Herny F dan Dwi Budi S pada tahun 2017 dengan judul Hierarchical Agglomerative Clustering untuk Pengelompokan Skripsi Mahasiswa (Februariyanti & Santoso, 2017). Penelitian tersebut juga melakukan pengelompokan skripsi melalui data judul menggunakan *text mining* dengan metode *Hierarchical Agglomerative Clustering*. Hasil penelitian menunjukkan bahwa metode tersebut dapat melakukan pengelompokan skripsi menjadi 5 buah cluster.

Berdasarkan uraian di atas, diperlukan sebuah analisis data terkait skripsi mahasiswa. Oleh karena itu, dikembangkan sistem pengelompokan skripsi menggunakan *text mining* (studi kasus Program Studi

Pendidikan Informatika Fakultas Ilmu Pendidikan Universitas Trunojoyo Madura). Adapun metode *clustering* yang digunakan adalah K-Means sebab terbukti baik dan handal untuk permasalahan *clustering* (Aditya & Fitrianah, 2021). Dengan adanya sistem ini, diharapkan dapat membantu manajemen pengelolahan data skripsi bagi bidang skripsi di Program Studi. Selain itu, dengan adanya sistem ini dapat membantu terkait penentuan kebijakan dosen pembimbing ataupun Program Studi dalam penentuan topik skripsi selanjutnya. Serta dapat meminimalisir peluang adanya kemiripan skripsi atau *issue* penelitian yang kurang *up to date*.

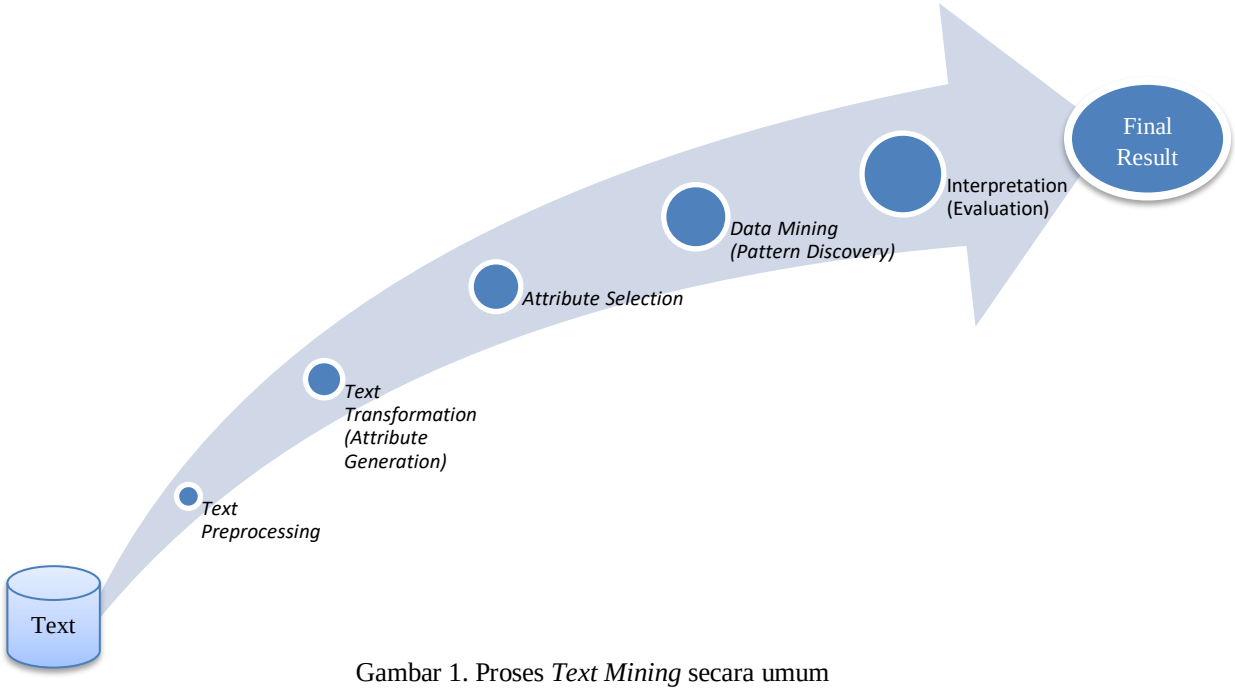
METODE PENELITIAN

Metode Text Mining

Text mining merupakan salah satu teknologi yang mengelola data teks untuk memperoleh informasi tertentu. Dengan *text mining*, informasi bisa didapatkan melalui hasil analisis pada data teks semi terstruktur maupun tidak terstruktur. Hal ini tentu sangat membantu pekerjaan manusia seiring dengan semakin banyaknya data teks ataupun dokumen yang ada pada web, aplikasi digital, maupun media sosial. Tentunya data-data tersebut memiliki jumlah besar dengan kurang terstruktur sehingga perlu waktu lama untuk menganalisis informasi di dalamnya.

Secara umum, *data mining* dan *text mining* dianggap serupa dengan persepsi bahwa teknik yang digunakan dalam *data mining* dapat juga digunakan dalam *text mining*. Namun, keduanya berbeda. Sebab *data mining* melibatkan data terstruktur, sedangkan *text mining* memerlukan ekstraksi ciri tertentu yang dimiliki teks sehingga diperlukan pemrosesan awal (Salloum et al., 2018).

Text mining pada umumnya memiliki beberapa tahapan sebagai berikut: praproses, pembangkitan atribut, pemilihan atribut, penemuan pola, dan interpretasi (Februariyanti & Santoso, 2017).



Gambar 1. Proses *Text Mining* secara umum

Gambar 1 menunjukkan tahapan atau proses yang ada pada Text Mining. Diawali dengan data masukan berupa teks (*text*), baik berupa kumpulan dokumen ataupun data teks dengan karakteristik tertentu. Diakhiri dengan perolehan hasil informasi yang diinginkan sesuai kebutuhan (*final result*). Adapun rincian tahapannya adalah:

a. Praproses (*Text Preprocessing*)

   Pada tahap pra proses, terdapat langkah-langkah berikut: *case folding, tokenization, filtering* dan *stemming*. *Case folding* digunakan untuk menghilangkan karakter selain huruf a-z dan mengubah semua huruf ke bentuk *lower case*. *Tokenization* dilakukan untuk memisahkan

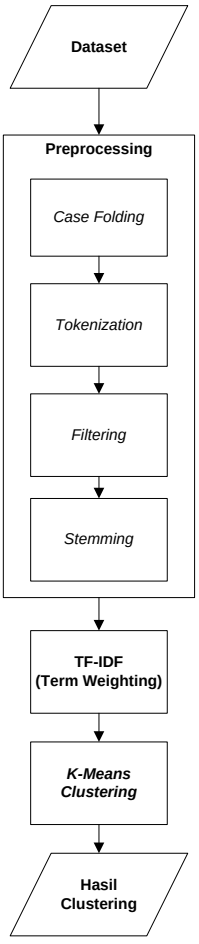
kalimat menjadi kata tunggal dan menghapus spasi, enter, tabulasi, dan tanda baca. *Filtering* dilakukan untuk memperoleh kata penting dalam teks. Kata penting diperoleh dengan menghilangkan *stopword* (kata sambung, kata depan, dan lainnya). *Stemming* merupakan proses yang dilakukan untuk mengubah kata menjadi bentuk kata dasar.

b. Pembangkitan Atribut (*Attribute Generation*)
Pada tahap pembangkitan atribut, dilakukan pemilihan fitur yang dapat digunakan sebagai atribut untuk merepresentasikan sebuah teks dalam memperoleh informasi. Ada beberapa pendekatan yang digunakan untuk pembangkitan atribut ini, yaitu pembobotan setiap kata, *penggunaan vector space document*, dan *stemming*. Meskipun pada artikel lain stemming masuk ke dalam pra proses, namun hakikatnya proses tersebut dilakukan untuk pembangkitan fitur atau atribut penting.

c. Pemilihan Atribut (*Attribute Selection*)
Pada tahap pemilihan atribut, dilakukan proses pengurangan dimensi fitur atau atribut jika terlalu besar dan menghilangkan atribut yang tidak relevan dengan tujuan perolehan informasi.

d. Penemuan Pola (*Data Mining* / *Pattern Discovery*)
Pada tahap penemuan pola, diperoleh *database* yang sudah terstruktur untuk diproses menggunakan teknik data mining seperti *clustering*, klasifikasi, analisis asosiasi, dan lainnya sesuai kebutuhan untuk mendapatkan informasi yang diharapkan.

e. Interpretasi (*Interpretation/Evaluation*)
Pada tahap interpretasi atau evaluasi, dilakukan pengujian pada hasil yang diperoleh dari proses data mining/pattern discovery. Apakah proses dapat dihentikan ataukah perlu diulang kembali sebab hasil belum sesuai harapan.



Gambar 2. Alur Pengelompokan Skripsi Menggunakan Clustering

Adapun proses pengelompokan dokumen skripsi dilakukan berdasarkan alur pada Gambar 2 sebagai berikut. Secara garis besar ada 3 kelompok proses. Pertama adalah *preprocessing* yang terdiri dari tahapan *case folding*, *tokenizing*, *filtering*, dan *stemming*. Kedua adalah proses pembobotan dengan menggunakan metode TF-IDF. Ketiga adalah proses pengelompokan atau *clustering* menggunakan metode K-Means.

Dataset yang digunakan adalah data abstrak dari dokumen skripsi mahasiswa program studi Pendidikan Informatika. Pada tahap awal, dilakukan *preprocessing* dengan berbagai tahapan, antara lain: *case folding*, *tokenizing*, *filtering*, dan *stemming*. Tahap tersebut dilakukan untuk menyiapkan data agar bebas *text* yang tidak berarti. Pada *case folding*, proses yang dilakukan adalah mengubah semua huruf menjadi *lower case*, menghapus angka, dan menghapus tanda baca. Pada tahap *tokenizing*, dilakukan proses pemisahan setiap kata. Pada tahap *filtering*, dilakukan proses pemilihan kata yang dianggap penting. Sedangkan pada tahap *stemming*, dilakukan proses pengubahan kata menjadi kata dasar.

Setelah kata dasar didapatkan, selanjutnya dilakukan pembobotan *term* untuk setiap kata pada dokumen menggunakan TF-IDF. Kemudian dilakukan pengelompokan dengan menggunakan algoritma *K-Means Clustering* sehingga diperoleh hasil *clustering*.

Metode K-Means

Berikut adalah langkah-langkah algoritma *K-Means Clustering* (Shanie, Suprijadi, & Zulhanif, 2017):

1. Tentukan jumlah cluster k minimum yang diinginkan.
2. Tentukan nilai ambang batas (*Threshold*).
3. Alokasikan setiap objek data ke dalam cluster terdekat untuk membentuk partisi baru.
4. Hitung centroid atau rata-rata dari data partisi baru.
5. Hitung jarak antara setiap objek ke setiap centroid menggunakan jarak *Euclidean* dengan persamaan berikut.

   $$d(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2} \quad (1)$$

   $d(x,y)$ merupakan jarak Euclidean objek ke centroid. x_k merupakan bobot kata ke-k pada cluster yang akan dihitung jaraknya ke centroid. y_k merupakan bobot kata ke-k pada centroid.
6. Untuk setiap cluster dihitung nilai SSE (*sum square error within group*) dengan fungsi sebagai berikut:

   $$SS_w = \sum_{k=1}^{k}\sum_{x_i \in C_k}\|x_i - c_k\| \quad (2)$$

   SS_w merupakan nilai SSE masing-masing cluster. x_i merupakan bobot kata ke-i pada cluster yang akan dihitung SSE nya. c_k merupakan centroid pada cluster ke-k.
7. Bandingkan nilai SSE masing-masing klaster mitra. Jika selisih nilai SSE setiap pasangan cluster melebihi nilai threshold, akan menambah jumlah cluster, dan objek yang paling jauh dalam cluster akan terpilih sebagai cluster baru.
8. Ulangi langkah 4-7.
9. Perhitungan selesai jika selisih antara nilai SSE setiap pasangan cluster lebih kecil dari nilai *threshold* untuk seluruh pasangan *cluster*, dan merupakan jumlah cluster yang optimal.

Metode Elbow

Metode Elbow digunakan untuk menentukan jumlah cluster yang optimal dalam dataset. Metode ini memilih nilai K yang tepat dengan menggunakan pengukuran WCSS (*Within Cluster Sum of Squares*). Untuk menentukan nilai K yang optimal, nilai K akan diperiksa satu per satu dan SSE (*Sum*

Square Error) nilai akan dicatat. SSE (Jumlah Kuadrat *Error*) adalah rumus yang digunakan untuk mengukur perbedaan antara data yang diperoleh dengan model ramalan yang telah dilakukan sebelumnya(Aditya & Fitrianah, 2021).

HASIL DAN PEMBAHASAN

Jumlah dataset yang diperoleh adalah 387 dokumen dengan rincian yang dapat dilihat pada Tabel 1. pengambilan data dilakukan melalui *repository* tugas akhir dan skripsi mahasiswa pada perpustakaan Universitas Trunojoyo Madura melalui https://library.trunojoyo.ac.id/elib/.

Tabel 1. Dataset yang digunakan

Tahun Lulusan	Jumlah
2017	60
2018	117
2019	112
2020	61
2021	37
Jumlah Data	387

Pada tahap desain sistem, dilakukan pemahaman bentuk data pada dataset yang diperoleh. Kemudian, dilakukan *preprocessing* dengan berbagai tahapan, antara lain: case folding, tokenizing, filtering, dan stemming. Tahap ini dilakukan dengan menggunakan software jupyter notebook 6.3.0. Tahap tersebut dilakukan untuk menyiapkan data agar bebas text yang tidak berarti. Berikut ini adalah hasil tahapan preprocessing, antara lain:

Input

Matakuliah Pemrograman Berorientasi Objek (PBO) adalah salah satu matakuliah wajib yang harus ditempuh mahasiswa Prodi Pendidikan Informatika. Terdapat beberapa permasalahan yang terjadi dalam pelaksanaannya, yaitu kurangnya media pembelajaran, kurangnya keaktifan mahasiswa, dan perlunya ilustrasi terhadap materi yang disampaikan. Hal ini menyebabkan kurang maksimalnya proses penyerapan materi oleh mahasiswa pada matakuliah PBO. Dengan memperhatikan karakteristik peserta didik dan kebutuhan pengguna, penelitian ini bertujuan untuk mengembangkan, menguji kelayakan, dan mengetahui respon pengguna pada media yang akan dikembangkan yaitu Pengembangan Game Edukasi 3D Factory Developer Menggunakan Metode Forward Chaining Dan Pendekatan Metafora Pada Pemrograman Berorientasi Objek. Penelitian ini merupakan penelitian pengembangan yang menggunakan model pengembangan Kemp dan Dayton yang terdiri dari sembilan tahapan yaitu (1) Tujuan umum, (2) Tujuan khusus, (3) Karakteristik peserta didik, (4) Isi materi, (5) Treatment, (6) Storybo ard, (7), Naskah, (8) Developing, editing, and mixing, dan (9) Testing and revising. Produk yang dihasilkan penelitian ini berupa game edukasi berbasis android untuk matakuliah PBO. Penilaian yang diperoleh dari uji ahli materi menunjukkan kelayakan sebesar 99% memiliki kualifikasi sangat layak, uji ahli media sebesar 93% memiliki kualifikasi sangat layak, uji coba perorangan sebesar 87% memiliki kualifikasi sangat layak, uji coba kelompok kecil sebesar 85% memiliki kualifikasi sangat layak, dan uji coba kelompok besar sebesar 85% memiliki kualifikasi sangat layak. Sehingga dapat disimpulkan bahwa game edukasi Factory Developer menggunakan metode Forward Chaining dan pendekatan Metafora layak digunakan sebagai media pembelajaran pada matakuliah Pemrograman Berorientasi Objek Program Studi Pendidikan Informatika.

case folding "Lower Case"

```
lower_case = kalimat.lower()
print(lower_case)
```

```
matakuliah pemrograman berorientasi objek (pbo) adalah salah satu matakuliah wajib yang harus ditempuh mahasiswa prodi pendidik
an informatika. terdapat beberapa permasalahan yang terjadi dalam pelaksanaannya, yaitu kurangnya media pembelajaran, kurangnya
keaktifan mahasiswa, dan perlunya ilustrasi terhadap materi yang disampaikan. hal ini menyebabkan kurang maksimalnya proses pen
yerapan materi oleh mahasiswa pada matakuliah pbo. dengan memperhatikan karakteristik peserta didik dan kebutuhan pengguna, pen
elitian ini bertujuan untuk mengembangkan, menguji kelayakan, dan mengetahui respon pengguna pada media yang akan dikembangkan
yaitu pengembangan game edukasi 3d factory developer menggunakan metode forward chaining dan pendekatan metafora pada pemrogram
an berorientasi objek. penelitian ini merupakan penelitian pengembangan yang menggunakan model pengembangan kemp dan dayton yan
g terdiri dari sembilan tahapan yaitu (1) tujuan umum, (2) tujuan khusus, (3) karakteristik peserta didik, (4) isi materi, (5)
treatment, (6) storyboard, (7), naskah, (8) developing, editing, and mixing, dan (9) testing and revising. produk yang dihasilk
an penelitian ini berupa game edukasi berbasis android untuk matakuliah pbo. penilaian yang diperoleh dari uji ahli materi menu
njukkan kelayakan sebesar 99% memiliki kualifikasi sangat layak, uji ahli media sebesar 93% memiliki kualifikasi sangat layak,
uji coba perorangan sebesar 87% memiliki kualifikasi sangat layak, uji coba kelompok kecil sebesar 85% memiliki kualifikasi san
gat layak, dan uji coba kelompok besar sebesar 85% memiliki kualifikasi sangat layak. sehingga dapat disimpulkan bahwa game edu
kasi factory developer menggunakan metode forward chaining dan pendekatan metafora layak digunakan sebagai media pembelajaran p
ada matakuliah pemrograman berorientasi objek program studi pendidikan informatika.
```

case folding "Removing Punctuation"

```
hasilRemovePunctuation = hasilReNum.translate(str.maketrans("","",string.punctuation))
print(hasilRemovePunctuation)
```

```
matakuliah pemrograman berorientasi objek pbo adalah salah satu matakuliah wajib yang harus ditempuh mahasiswa prodi pendidikan
informatika terdapat beberapa permasalahan yang terjadi dalam pelaksanaannya yaitu kurangnya media pembelajaran kurangnya keakt
ifan mahasiswa dan perlunya ilustrasi terhadap materi yang disampaikan hal ini menyebabkan kurang maksimalnya proses penyerapan
materi oleh mahasiswa pada matakuliah pbo dengan memperhatikan karakteristik peserta didik dan kebutuhan pengguna penelitian in
i bertujuan untuk mengembangkan menguji kelayakan dan mengetahui respon pengguna pada media yang akan dikembangkan yaitu pengem
bangan game edukasi d factory developer menggunakan metode forward chaining dan pendekatan metafora pada pemrograman berorienta
si objek penelitian ini merupakan penelitian pengembangan yang menggunakan model pengembangan kemp dan dayton yang terdiri dari
sembilan tahapan yaitu tujuan umum tujuan khusus karakteristik peserta didik isi materi treatment storyboard naskah dev
eloping editing and mixing dan testing and revising produk yang dihasilkan penelitian ini berupa game edukasi berbasis android
untuk matakuliah pbo penilaian yang diperoleh dari uji ahli materi menunjukkan kelayakan sebesar memiliki kualifikasi sangat l
ayak uji ahli media sebesar memiliki kualifikasi sangat layak uji coba perorangan sebesar memiliki kualifikasi sangat layak u
ji coba kelompok kecil sebesar memiliki kualifikasi sangat layak dan uji coba kelompok besar sebesar memiliki kualifikasi san
gat layak sehingga dapat disimpulkan bahwa game edukasi factory developer menggunakan metode forward chaining dan pendekatan me
tafora layak digunakan sebagai media pembelajaran pada matakuliah pemrograman berorientasi objek program studi pendidikan infor
matika kata kunci game forward chaining metafora pbo
```

Hasil *Tokenizing*

```
hasilRemovingWhitespace = hasilRemovingWhitespace.translate(str.maketrans('','',string.punctuation)).lower()

tokens = nltk.tokenize.word_tokenize(hasilRemovingWhitespace)
print(tokens)
```

```
['matakuliah', 'pemrograman', 'berorientasi', 'objek', 'pbo', 'adalah', 'salah', 'satu', 'matakuliah', 'wajib', 'yang', 'haru
s', 'ditempuh', 'mahasiswa', 'prodi', 'pendidikan', 'informatika', 'terdapat', 'beberapa', 'permasalahan', 'yang', 'terjadi',
'dalam', 'pelaksanaannya', 'yaitu', 'kurangnya', 'media', 'pembelajaran', 'kurangnya', 'keaktifan', 'mahasiswa', 'dan', 'perlun
ya', 'ilustrasi', 'terhadap', 'materi', 'yang', 'disampaikan', 'hal', 'ini', 'menyebabkan', 'kurang', 'maksimalnya', 'proses',
'penyerapan', 'materi', 'oleh', 'mahasiswa', 'pada', 'matakuliah', 'pbo', 'dengan', 'memperhatikan', 'karakteristik', 'pesert
a', 'didik', 'dan', 'kebutuhan', 'pengguna', 'penelitian', 'ini', 'bertujuan', 'untuk', 'mengembangkan', 'menguji', 'kelayaka
n', 'dan', 'mengetahui', 'respon', 'pengguna', 'pada', 'media', 'yang', 'akan', 'dikembangkan', 'yaitu', 'pengembangan', 'gam
e', 'edukasi', 'd', 'factory', 'developer', 'menggunakan', 'metode', 'forward', 'chaining', 'dan', 'pendekatan', 'metafora', 'p
ada', 'pemrograman', 'berorientasi', 'objek', 'penelitian', 'ini', 'merupakan', 'penelitian', 'pengembangan', 'yang', 'mengguna
kan', 'model', 'pengembangan', 'kemp', 'dan', 'dayton', 'yang', 'terdiri', 'dari', 'sembilan', 'tahapan', 'yaitu', 'tujuan', 'u
mum', 'tujuan', 'khusus', 'karakteristik', 'peserta', 'didik', 'isi', 'materi', 'treatment', 'storyboard', 'naskah', 'developin
g', 'editing', 'and', 'mixing', 'dan', 'testing', 'and', 'revising', 'produk', 'yang', 'dihasilkan', 'penelitian', 'ini', 'beru
pa', 'game', 'edukasi', 'berbasis', 'android', 'untuk', 'matakuliah', 'pbo', 'penilaian', 'yang', 'diperoleh', 'dari', 'uji',
'ahli', 'materi', 'menunjukkan', 'kelayakan', 'sebesar', 'memiliki', 'kualifikasi', 'sangat', 'layak', 'uji', 'ahli', 'media',
'sebesar', 'memiliki', 'kualifikasi', 'sangat', 'layak', 'uji', 'coba', 'perorangan', 'sebesar', 'memiliki', 'kualifikasi', 'sa
ngat', 'layak', 'uji', 'coba', 'kelompok', 'kecil', 'sebesar', 'memiliki', 'kualifikasi', 'sangat', 'layak', 'dan', 'uji', 'cob
a', 'kelompok', 'besar', 'sebesar', 'memiliki', 'kualifikasi', 'sangat', 'layak', 'sehingga', 'dapat', 'disimpulkan', 'bahwa',
'game', 'edukasi', 'factory', 'developer', 'menggunakan', 'metode', 'forward', 'chaining', 'dan', 'pendekatan', 'metafora', 'la
yak', 'digunakan', 'sebagai', 'media', 'pembelajaran', 'pada', 'matakuliah', 'pemrograman', 'berorientasi', 'objek', 'program',
'studi', 'pendidikan', 'informatika']
```

Perhitungan frekuensi kemunculan hasil token

```
tokens = nltk.tokenize.word_tokenize(hasilRemovingWhitespace)
kemunculan = nltk.FreqDist(tokens)
print(kemunculan.most_common())
```

```
[('yang', 8), ('dan', 8), ('layak', 6), ('matakuliah', 5), ('uji', 5), ('sebesar', 5), ('memiliki', 5), ('kualifikasi', 5), ('s
angat', 5), ('media', 4), ('materi', 4), ('ini', 4), ('pada', 4), ('penelitian', 4), ('pemrograman', 3), ('berorientasi', 3),
('objek', 3), ('pbo', 3), ('mahasiswa', 3), ('yaitu', 3), ('pengembangan', 3), ('game', 3), ('edukasi', 3), ('menggunakan', 3),
('coba', 3), ('pendidikan', 2), ('informatika', 2), ('kurangnya', 2), ('pembelajaran', 2), ('karakteristik', 2), ('peserta',
2), ('didik', 2), ('pengguna', 2), ('untuk', 2), ('kelayakan', 2), ('factory', 2), ('developer', 2), ('metode', 2), ('forward',
2), ('chaining', 2), ('pendekatan', 2), ('metafora', 2), ('dari', 2), ('tujuan', 2), ('and', 2), ('ahli', 2), ('kelompok', 2),
('adalah', 1), ('salah', 1), ('satu', 1), ('wajib', 1), ('harus', 1), ('ditempuh', 1), ('prodi', 1), ('terdapat', 1), ('beberap
a', 1), ('permasalahan', 1), ('terjadi', 1), ('dalam', 1), ('pelaksanaannya', 1), ('keaktifan', 1), ('perlunya', 1), ('ilustras
i', 1), ('terhadap', 1), ('disampaikan', 1), ('hal', 1), ('menyebabkan', 1), ('kurang', 1), ('maksimalnya', 1), ('proses', 1),
('penyerapan', 1), ('oleh', 1), ('dengan', 1), ('memperhatikan', 1), ('kebutuhan', 1), ('bertujuan', 1), ('mengembangkan', 1),
('menguji', 1), ('mengetahui', 1), ('respon', 1), ('akan', 1), ('dikembangkan', 1), ('d', 1), ('merupakan', 1), ('model', 1),
('kemp', 1), ('dayton', 1), ('terdiri', 1), ('sembilan', 1), ('tahapan', 1), ('umum', 1), ('khusus', 1), ('isi', 1), ('treatmen
t', 1), ('storyboard', 1), ('naskah', 1), ('developing', 1), ('editing', 1), ('mixing', 1), ('testing', 1), ('revising', 1),
('produk', 1), ('dihasilkan', 1), ('berupa', 1), ('berbasis', 1), ('android', 1), ('penilaian', 1), ('diperoleh', 1), ('menunju
kkan', 1), ('perorangan', 1), ('kecil', 1), ('besar', 1), ('sehingga', 1), ('dapat', 1), ('disimpulkan', 1), ('bahwa', 1), ('di
gunakan', 1), ('sebagai', 1), ('program', 1), ('studi', 1)]
```

Langkah selanjutnya yaitu dilakukan pembobotan term untuk setiap kata pada dokumen menggunakan TF-IDF. Python memiliki library khusus untuk melakukan perhitungan TF-IDF. Library yang digunakan yaitu TfidfVectorizer dari sklearn.feature_extraction.text. Berikut ini implementasi Tf-Idf untuk data uji. Adapun hasilnya dapat dilihat pada Tabel 2.

```
#Menghitung TF-IDF
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(hasil_stem)
namaFitur=vectorizer.get_feature_names()
print(namaFitur, type(namaFitur))
len(namaFitur)
dense = X.todense()
denselist = dense.tolist()
print(denselist[1])
#print(X.shape)
len(denselist)

dflatih = pd.DataFrame(denselist, columns = namaFitur)
dflatih
dflatih.to_excel("tfidlatih.xlsx")
```

Tabel 2 Hasil perhitungan Tf-idf

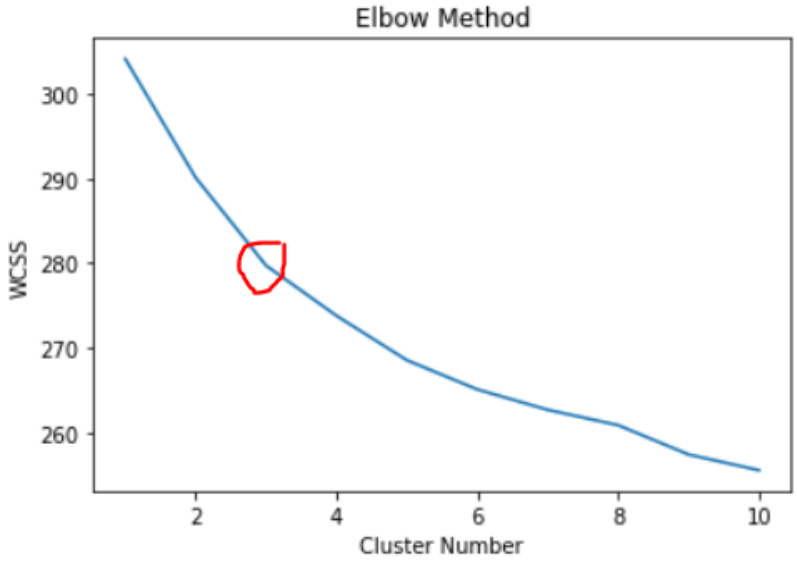
i	abad	absah	abstrak	acak	acu
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0,057909	0	0	0
4	0	0,044533	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0,080964
10	0	0	0	0	0
11	0	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0
18	0	0	0	0	0
19	0	0,064901	0	0	0
20	0	0	0	0,076899	0
21	0	0	0	0	0
22	0	0	0	0	0
23	0	0	0	0	0
24	0	0	0	0	0
25	0	0	0	0	0
26	0	0	0	0	0
27	0,074361	0	0	0	0
28	0	0	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0
31	0	0	0	0	0
32	0	0	0,080598	0	0
33	0	0	0	0	0
34	0	0	0	0	0
35	0	0	0	0	0
36	0	0	0	0	0

Untuk penentuan nilai k secara manual yang digunakan adalah sesuai pembagian topik skripsi yang ada di program studi pendidikan informatika, yaitu 3 cluster. Selain itu juga dilakukan penentuan k optimal menggunakan metode elbow. Dari hasil optimasi penentuan jumlah K menggunakan metode elbow, didapatkan bahwa jumlah K optimal adalah 3 dengan nilai WCSS sebesar 279,7108 yang dapat dilihat pada Tabel 3. Grafik perhitungan WCSS untuk 350 data latih dapat dilihat pada Gambar 3. Semakin kecil skor WCSS, semakin baik. Sumbu x adalah jumlah kluster, sumbu y adalah skor WCSS.

Bisa dilihat bahwa saat K=1, nilai WCSS sangat tinggi. Kemudian menurun terus sampai K=3 terlihat membentuk seperti sebuah siku.

Tabel 3 Hasil perhitungan WCSS

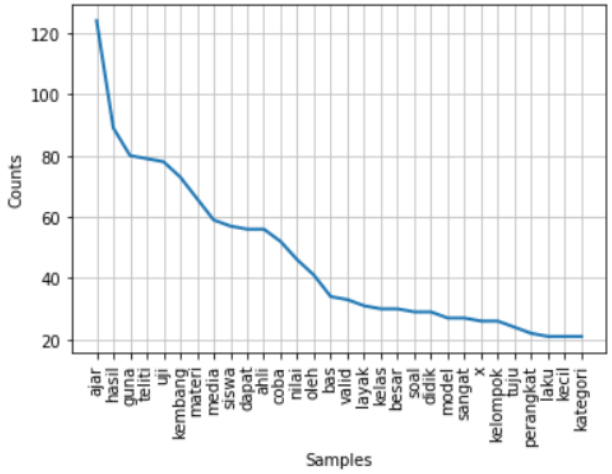
Jumlah K	WCSS
1	304,1357
2	290,0975
3	279,7108
4	273,7813
5	268,5261
6	265,0761
7	262,667
8	260,8525
9	257,3866
10	255,5285



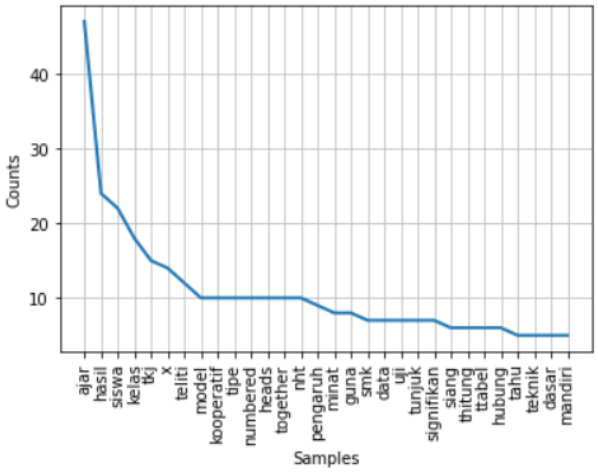
Gambar 3 Grafik WCSS metode Elbow

Setelah diketahui jumlah K yang optimal, langkah berikutnya adalah pembentukan model clustering K-Means dengan menggunakan 350 data latih yang menghasilkan 3 cluster yaitu cluster 0, cluster 1, dan cluster 2. Model tersebut kemudian diujikan kepada 37 data uji. Hasil yang didapatkan dari proses clustering dengan model yang telah dibentuk terhadap 37 data uji adalah sebagai berikut.

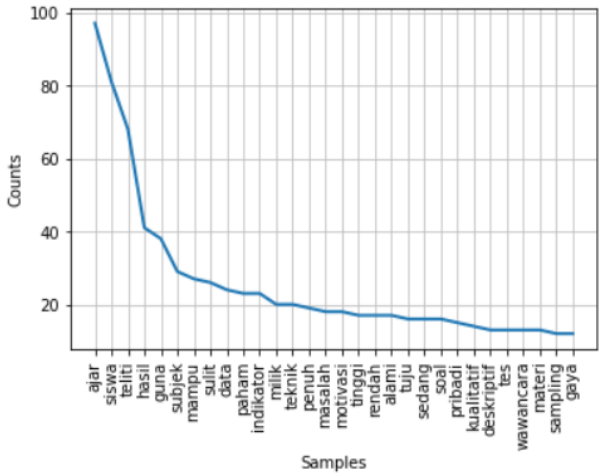
Dari proses clustering yang dilakukan terhadap data uji, dapat dilakukan analisis hasilnya. Rincian jumlah term untuk setiap cluster atau kelompok adalah 3235 term pada cluster 0, 491 term pada cluster 1, dan 1806 term pada cluster 2. Sedangkan frekuensi tertinggi kemunculan term untuk tiap cluster dapat dilihat pada Gambar 4, 5, dan 6.



Gambar 4 Frekuensi kemunculan term untuk kelompok pertama (cluster 0)



Gambar 5 Frekuensi kemunculan term untuk kelompok kedua (cluster 1)



Gambar 6 Frekuensi kemunculan term untuk kelompok ketiga (cluster 2)

Evaluasi

Evaluasi dilakukan terhadap hasil performa algoritma yang digunakan untuk pengelompokan skripsi. Pada pengujian ini, digunakan acuan data berupa sekumpulan dokumen skripsi yang telah dikelompokan secara manual. Pengujian ini menggunakan *Confusion Matrix* pada Tabel 4. Pengukuran

dengan metode ini dapat menggunakan persamaan akurasi, presisi, sensitivitas (recall), dan F-Measure terhadap data cluster hasil pengujian *k-means clustering* dan data cluster secara manual. Prosedur yang digunakan dapat dilihat pada Tabel 5. adapun hasil evaluasinya adalah sebagai berikut (Husni, Negara, & Syarief, 2015)(Wahyuni, Arifiyanti, & Afandi, 2020):

Tabel 4. *Confusion Matrix* (Wahyuni, Arifiyanti, & Afandi, 2020)

Confusion Matrix		Prediksi	
		Benar	Salah
Aktual	Benar	TP (True Positive)	FP (False Positive)
	Salah	FN (False Negative)	TN (True Negative)

TP = *True positive* (cluster prediksi adalah benar namun kelas sebenarnya adalah benar). TN= *True negative* (cluster prediksi adalah salah namun kelas sebenarnya adalah benar). FP = *False positive* (cluster prediksi adalah benar namun kelas sebenarnya adalah salah). FN= *False negative* (cluster prediksi adalah salah namun kelas sebenarnya adalah benar). Sedangkan untuk metric pengukuran dapat dilihat pada Tabel 5.

Tabel 5. Metric pengukuran untuk evaluasi performa algoritma (Wahyuni, Arifiyanti, & Afandi, 2020)

Metric	**Rumus**
Akurasi	$akurasi = \frac{TP + TN}{TP + TN + FP + FN}$
Recall/ Presisi	$Recall = \frac{TP}{TP + FN}$
Precision	$Precision = \frac{TP}{TP + FP}$
F-Measure	$F - Measure = 2 * \frac{Recall * Precision}{Recall + Precision}$

Tabel 6. *Confusion Matrix* pada Hasil Clustering Data Uji

Confusion Matrix		Prediksi		
		Cluster 0	Cluster 1	Cluster 2
Aktual	Cluster 0	20	0	0
	Cluster 1	1	3	0
	Cluster 2	0	0	13

Nilai TP adalah 20 + 3 + 13 = 36. Sedangkan jumlah data uji adalah 37. Sehingga diperoleh nilai Akurasi = TP/Jumlah Data = 36/37 = 0,972972973. Sedangkan untuk perhitungan nilai Precision dapat dilakukan dengan tabel penolong pada Tabel 7. Nilai Precision diperoleh dari perhitungan (1+0,75+1)/3. Nilai 3 adalah jumlah cluster. Sehingga nilai Precision yang didapatkan adalah 0,916666667. Kemudian dilakukan nilai Recall melalui tabel penolong pada Tabel 8. Nilai Recall diperoleh dari perhitungan (0,952380952+1+1)/3. Sehinggan nilai *Recall* yang didapatkan adalah 0,9849199722. Setelah nilai Precision dan Recall didapatkan, kemudian dapat dilakukan perhitungan nilai F (*F-Measure/F-Score*). Nilai *F-Measure* yang didapatkan adalah 0,949199722.

Tabel 7. Tabel Penolong Perhitungan Nilai *Presisi*

	Cluster 0	Cluster 1	Cluster 2
TP	20	3	13
FP	0+0	1+0	0+0
Precision	20/(20+0)=1	3/(3+1) = 0,75	13/(13+0) = 1

Tabel 8. Tabel Penolong Perhitungan Nilai *Recall*

	Cluster 0	Cluster 1	Cluster 2
TP	20	3	13
FN	1+0	0+0	0+0
Precision	20/(20+1)= 0,952380952	3/(3+0) = 1	13/(13+0) = 1

KESIMPULAN

Pada penelitian ini, telah diimplementasikan metode K-means untuk pengelompokan dokumen skripsi. Adapun nilai K Optimal yang didapatkan yaitu 3 cluster. Hasil penelitian ini menunjukkan bahwa metode clustering menggunakan K-Means dapat melakukan pengelompokan skripsi secara optimal dengan nilai akurasi sebesar 0,972972973, nilai presisi sebesar 916666667, nilai *recall* sebesar 0,9849199722, dan *F-Measure* sebesar 0,949199722 dalam skala 0 – 1.

DAFTAR PUSTAKA

Aditya, D. L., & Fitrianah, D. (2021). Comparative Study of Fuzzy C-Means and K-Means Algorithm for Grouping Customer Potential in Brand Limback. *Jurnal Riset Informatika*, *3*(4), 327–334. https://doi.org/10.34288/jri.v3i4.241

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques. *ArXiv*.

Februariyanti, H., & Santoso, D. B. (2017). Hierarchical Agglomerative Clustering untuk Pengelompokan Skripsi Mahasiswa. *Prosiding SINTAK*, 33–40.

Husni, Negara, Y. D. P., & Syarief, M. (2015). Clusterisasi Dokumen Web (Berita) Bahasa Indonesia Menggunakan Algoritma K-Means (Clustering Web Documents (News) Indonesian Language Using K-Means Algorithm). *Jurnal Simantec*, *4*(3), 159–166.

Mustikasari, D. (2017). Analisis Tema Skripsi Mahasiswa Menggunakan Document Clustering Dengan Algoritma Lingo. *Kinetik*, *2*(2), 131–140. https://doi.org/10.22219/kinetik.v2i2.180

Rahmawati, L., Widya Sihwi, S., & Suryani, E. (2016). Analisa Clustering Menggunakan Metode K-Means Dan Hierarchical Clustering (Studi Kasus : Dokumen Skripsi Jurusan Kimia, Fmipa, Universitas Sebelas Maret). *Jurnal Teknologi & Informasi ITSmart*, *3*(2), 66. https://doi.org/10.20961/its.v3i2.654

Salloum, S. A., Al-Emran, M., Monem, A. A., & Shaalan, K. (2018). Using text mining techniques for extracting information from research articles. *Studies in Computational Intelligence*, *740*, 373–397. https://doi.org/10.1007/978-3-319-67056-0_18

Shanie, T., Suprijadi, J., & Zulhanif. (2017). Text grouping in patent analysis using adaptive K-means clustering algorithm. *AIP Conference Proceedings*, *1827*, 1–9. https://doi.org/10.1063/1.4979457

Wahyuni, E. D., Arifiyanti, A. A., & Afandi, M. I. (2020). *Klasifikasi Teks Dengan Python*. Indomedia Pustaka.

Zulaikah, I. (2017). *Pengelompokan Skripsi Menggunakan Self Organizing Maps Clustering (Studi Kasus : Prodi Teksnik Informatika Universitas Nusantara PGRI Kediri)*. 13–14.