PENGEMBANGAN MODEL PENCARIAN DOKUMEN SKRIPSI MAHASISWA UNIVERSITAS JEMBER DALAM REPOSITORY UNEJ DENGAN MENGGUNAKAN VECTOR SPACE MODEL

Oleh:
Yusran Alindri Dwimajida
NIM 182410101134

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTsER
UNIVERSITAS JEMBER
2022

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam mengerjakan skripsi;
2. Ayahanda Indrio Susherdianto, Ibunda Sri Mulyeni, Kakak pertama Yenaafi Mahdirio, Adik Pertama Aldorio Oktmusyaffa, dan Adik Kedua Janita Saufayuthika;
3. Sahabat – sahabat saya yang selalu mendukung saya;
4. Guru-guru dari Pendidikan formal maupun informal;
5. Almamater Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Jember

MOTTO

"Tanpa ilmu, amal itu tidak ada gunanya. Sedangkan ilmu tanpa amal adalah hal yang sia-sia."

- Abu Bakar Asshidiq -

"Mahkota seseorang adalah akalnya. Derajat seseorang adalah agamanya. Sedangkan kehormatan seseorang adalah budi pekertinya."

- Umar Bin Khattab -

"Aku sudah pernah merasakan semua kepahitan dalam hidup dan yang paling pahit ialah berharap kepada manusia."

- Ali Bin Abi Thalib -

PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Yusran Alindri Dwimajida

NIM : 182410101134

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "PENGEMBANGAN MODEL PENCARIAN DOKUMEN SKRIPSI MAHASISWA UNIVERSITAS JEMBER DALAM REPOSITORY UNEJ DENGAN MENGGUNAKAN VECTOR SPACE MODEL", adalah benar – benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika kemudian hari pernyataan ini tidak benar.

Jember, 24 November 2022

Yang menyatakan,

Yusran Alindri Dwimajida

NIM 182410101134

SKRIPSI

PENGEMBANGAN MODEL PENCARIAN DOKUMEN SKRIPSI MAHASISWA UNIVERSITAS JEMBER DALAM REPOSITORY UNEJ DENGAN MENGGUNAKAN VECTOR SPACE MODEL

Oleh :

Yusran Alindri Dwimajida

NIM 182410101134

Pembimbing

Dosen Pembimbing Utama	: Achmad Maududie, ST., M.Sc
Dosen Pembimbing Pendamping	: Tio Dharmawan, S.Kom.,M.Kom

PENGESAHAN PEMBIMBING

Skripsi berjudul ”Pengembangan Model Pencarian Dokumen Skripsi Mahasiswa Universitas Jember Dalam Repository Unej Dengan Menggunakan Vector Space Model” karya Yusran Alindri Dwimajida telah diuji dan disahkan pada:

hari, tanggal : Rabu, 14 Desember 2022

tempat : Fakultas Ilmu Komputer Universitas Jember.

Disetujui Oleh :

Pembimbing I,

Achmad Maududie, ST., M.Sc

NIP. 197004221995121001

Pembimbing II,

Tio Dfarmawan, S.Kom.,M.Kom

NIP. 199111122022031011

PENGESAHAN PENGUJI

Skripsi ”Pengembangan Model Pencarian Dokumen Skripsi Mahasiswa Universitas Jember Dalam Repository Unej Dengan Menggunakan Vector Space Model” karya Yusran Alindri Dwimajida telah diuji dan disahkan pada:

hari, tanggal : Rabu, 14 Desember 2022

tempat : Fakultas Ilmu Komputer Universitas Jember.

Disetujui Oleh :

Penguji I,	Penguji II,
Anang Andrianto, ST., MT	M. Arief Hidayat, S.Kom., M.Kom.
NIP. 196906151997021002	NIP/NRP. 198101232010121003

RINGKASAN

Pengembangan Model Pencarian Dokumen Skripsi Mahasiswa Universitas Jember Dalam Repository Unej Dengan Menggunakan Vector Space Model; Yusran Alindri Dwimajida; 2022; 68 halaman; program studi sistem informasil fakultas ilmu computer; Universitas Jember.

Banyaknya dokumen skripsi yang diterbitkan oleh mahasiswa Universitas Jember, diperlukan suatu cara untuk dapat mencari dokumen skripsi mahasiswa dengan cepat dan mudah. Salah satu cara yang dapat diterapkan adalah dengan melalui komputerisasi dokumen skripsi mahasiswa Universitas Jember yang dikenal dengan Repository UNEJ. Tetapi, pencarian skripsi pada Repository UNEJ masih berdasarkan kehadiran katanya saja, oleh karena itu sering sekali hasil pencarian tidak sesuai dengan keinginan dari penggunannya.

Salah satu metode untuk melakukan pencarian dokumen adalah dengan menggunakan konsep *information retrieval*. Information retrieval adalah suatu pemrosesan komputer dengan menggunakan kode tertentu dalam upaya mencari dokumen yang ingin dicari sesuai dengan kebutuhan pengguna. Dalam menerapkan konsep *information retrieval,* terdapat berbagai macam cara salah satunya adalah menggunakan *vector space model*. vector space model merupakan suatu konsep pencarian dokumen dengan meperhatikan jarak antar vektor. Vektor space model dipilih karena memiliki tingkat keakuratan yang tinggi.

PRAKATA

Puji syukur kehadirat Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul ”Pengembangan Model Pencarian Dokumen Skripsi Mahasiswa Universitas Jember Dalam Repository Unej Dengan Menggunakan Vector Space Model”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember: Penyusunan skripsi ini tidak lepas dari dukungan berbagai belah pihak. Oleh karena itu penulis menyampaikan terima kasih kepada:

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan proses pengerjaan skripsi;
2. Doa, dukungan, dan harapan dari kedua orangtua saya yaitu, Indrio Susherdianto dan Sri Mulyeni;
3. Achmad Maududie, ST., M.Sc selaku dosen pembimbing utama dan Kepala Prodi Sistem Informasi Fakultas Ilmu Komputer;
4. Tio Dharmawan, S.Kom.,M.Kom selaku dosen pembimbing pendamping yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
5. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Fakultas Ilmu Komputer Universitas Jember;
6. Diri saya sendiri yang mau dan mampu bertahan, berjuang, berusaha dengan sekuat tenaga, dan tidak menyerah walau dalam perjalanannya selalu berhadapan dengan berbagai tantangan secara sabar. Terimakasih untuk diri sendiri sudah bertahan sejauh ini;
7. Arizha Izzul selaku Dospem ketiga saya yang telah membantu, mendampingi, memotivasi dan memberi dukungan moral selama proses skripsi;
8. Ungki Aprilian yang telah membantu, mendukung, memotivasi, dan menjadi teman cerita yang pengertian;
9. Adik - adik saya Tanaya Naila Astari, Carenina Atiyar Pangastuti, Revina Hani Rahmadilla, Mustika Binar Bestari, Rizqi Kamila, dan Rasyiddin Permanaputra yang telah memberikan dukungan moral dalam pengerjaan skripsi ini;
10. Teman - teman inti HIMASIF Periode 2020/2021 yaitu Rio Gunawan, Ungki Aprilian, Hilmy Citra, Dicky Fattah, Karendhika Argiansyah, Muhammad Wahid Ash Shiddiq yang telah memberikan dukungan moral, teman cerita, dan dukungan kepada saya selama proses pengerjaan skripsi ini;
11. Kepengurusan Himpunan Mahasiswa Sistem Informasi periode 2018/2019, 2019/2020, dan 2020/2021 yang memberikan pelajaran hidup berharga selama saya menjadi pengurus;

12. Kakak – Kakak saya Brian Rizqi P.D yang telah memberikan motivasi dan saran dalam proses pengerjaan skripsi;
13. Teman –teman Program Studi Sistem Informasi di semua angkatan 2018;
14. Semua pihak yang tidak dapat disebutkan satu persatu; Dengan harapan bahwa penelitian ini nantinya terus berlanjut dan berkembang kelak, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat bagi semua pihak.

Jember, 24 November 2022

Penulis

BAB 1. Pendahuluan

1.1. LATAR BELAKANG

Perpustakaan UNEJ memiliki wadah penyimpanan hasil penelitian ilmiah yang dilakukan oleh civitas akademika UNEJ secara digital, yang dikenal sebagai Repository UNEJ. Tujuan dari Repository UNEJ adalah untuk memudahkan civitas akademika bisa melestarikan dan membagikan publikasinya. Saat ini diperkirakan ada sekitar 50.000 lebih dokumen yang berada didalam Repository UNEJ. Dengan jumlah dokumen sebanyak itu, diperlukan suatu sistem pencarian dokumen yang dapat memberikan dokumen sesuai dengan *query*/*input* yang dimasukkan oleh pengguna. Saat ini, hasil pencarian pada Repository UNEJ masih berupa pencarian berdasarkan kehadiran suatu kata dari *query* yang *diinputkan* dengan kehadiran kata yang berada didalam dokumen skripsi. Sehingga masih cukup sering terjadi ketidaktepatan dalam memberikan hasil pencarian dokumen skripsi, seperti kesalahan dalam mengahasilkan topik skripsi yang sesuai dengan hasil *input* oleh *user*. Oleh karena itu, perlu adanya model yang dapat melakukan pencarian dokumen yang dapat menghasilkan hasil pencarian sesuai dengan kebutuhan dari penggunanya. Sedangkan tantangan yang menjadi perhatian adalah peneliti tidak memiliki akses secara langsung terhadap dokumen skripsi di repository UNEJ.

Dalam mengimplementasikan pembuatan sistem pencarian dokumen ini, peneliti menggunakan konsep *Information Retrieval*. *Information retrieval* merupakan bidang ilmu yang berkaitan dengan struktur, analisis, atau pengorganisasian, penyimpanan, pencarian, dan pengambilan informasi (Croft et al., 2010). Didalam bidang ilmu *information retrieval* terdapat berbagai macam model dalam melakukan implementasi pencarian dokumen, salah satunya adalah *Vector space model*. *Vector space model* dipilih karena memiliki cara kerja yang efisien, mudah dalam representasi,dan dapat diimplementasikan pada pencocokan dokumen (Croft et al.,2010). Sehingga, penggunaan *vector space model* dapat menjadi solusi untuk membuat sistem pencarian dokumen dengan waktu yang cukup singkat.

Sedangkan untuk mendapatkan database skirpsi melalui Repository UNEJ dilakukan proses web scraping dengan melihat tag-tag website yang sesuai dengan kebutuhan untuk database. Rancangan sistem yang akan dibuat berupa pembuatan sistem pencarian dimana sistem akan melakukan pencarian dokumen berdasarakan *query* yang dimasukkan oleh pengguna dengan mencocokan *query* terhadap judul dan abstrak dari tiap dokumen skripsi. Dengan melakukan dua jenis sumber pencocokan, diharapkan akan memberikan hasil pencarian dokumen yang lebih akurat.

1.2. RUMUSAN MASALAH

Berdasarkan latar belakang diatas. Adapun rumusan masalah pada penelitian ini adalah:

1. Bagaiamana rancangan metode pembacaan data skripsi mahasiswa pada Repository UNEJ tanpa perlu mengakses database repository tersebut secara langsung?
2. Bagaimana tingkat akurasi hasil pencarian dokumen skripsi mahasiswa Universitas Jember menggunakan model *Vector Space Model?*

1.3. TUJUAN

Adapun tujuan yang ingin dicapai dari penelitian ini adalah:

1. Merancangan sistem pencarian dokumen skripsi mahasiswa UNEJ tanpa perlu mengkases database Repository UNEJ secara langsung
2. Mengetahui tingkat akurasi hasil pencarian dokumen skripsi mahasiswa Universitas Jember menggunakan *Vector Space Model*

1.4. MANFAAT

Hasil dari penelitian ini diharapkan dapat memberi manfaat, diantaranya :

1. Bagi Peneliti

   Penelitian ini diharapkan dapat menjadi wadah dalam mengimplementasikan ilmu yang sudah dipelajari selama di kuliah sekaligus belajar ilmu baru yang sebelumnya belum pernah peneliti pelajari

2. Bagi Akademis

   Penelitian ini diharapkan dapat menjadi kontribusi mahasiswa dalam mengimplementasi ilmu yang telah didapatkan selama diperkuliahan dan menjadi refrensi kepada para pembaca, serta dapat menjadi dasar dalam melakukan penelitian lanjutan

3. Bagi Objek penelitian

   Penelitian ini diharapkan dapat menjadi bahan evaluasi dalam sistem pencarian dokumen di Repository UNEJ, sehingga dimasa yang akan datang sistem pencarian dokumen di Repository UNEJ dapat menjadi lebih baik lagi.

BAB 2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Pada penelitian oleh Fauzi et al. (2019) memiliki metode yang serupa yaitu penelitian ini memaparkan penerapana metode *vector space model* pada pencarian dokumen. metode *vector space model* memberikan sebuah kerangka pencocokan parsial dengan menetapkan bobot non-biner pada istilah indeks dalam *query* dan dokumen. Istilah pada dokumen direpresentasikan sebagai dimensi dari ruang vektor. Relevansi sebuah dokumen ke sebuah *query* didasarkan pada similaritas antara vektor dokumen dan *query.* Pada penelitian ini dilalui beberapa tahap yaitu filtering, indexing, perhitungan tf-idf, perhitungan magnitude, perhitungan dot product, dan perhitungan similaritas cosine. Hasil dari penelitian ini berupa urutan dokumen yang ditampilkan dari penerapan vector space model dengan disesuaikan query yang diinputkan dengan melihat banyaknya kata yang mirip (similar) dari perhitungan similaritas cosine.

Selanjutnya, Penelitian yang dilakukan oleh Syahrian et al. (2019) memiliki topik yang serupa. Metode yang digunakan pada penelitian ini adalah *vector space model*. Pada penelitian ini peneliti bertujuan untuk menciptakan aplikasi sistem pencarian data yang dapat membantu pengguna terutama fakultas teknologi informasi perbanas Jakarta dalam pencarian dokumen menjadi lebih cepat, mudah, dan lebih akurat dengan mengaplikasikan metode *vector space model.* Pada penelitian ini juga menggunakan pendekatan deskriptif kualitatif dan data yang digunakan untuk penelitian ini adalah berasal dari data FTI mahasiswa perbanas Jakarta. Sedangkan untuk hasil penelitian ini yaitu, menghasilkan aplikasi pencarian data pada fakultas teknologi perbanas Jakarta dan menghasilkan aplikasi pencarian data yang telah memiliki indeks kesamaan (*similarity index*) dari penerapan *Vector Space Model.*

2.2. Web Scraping

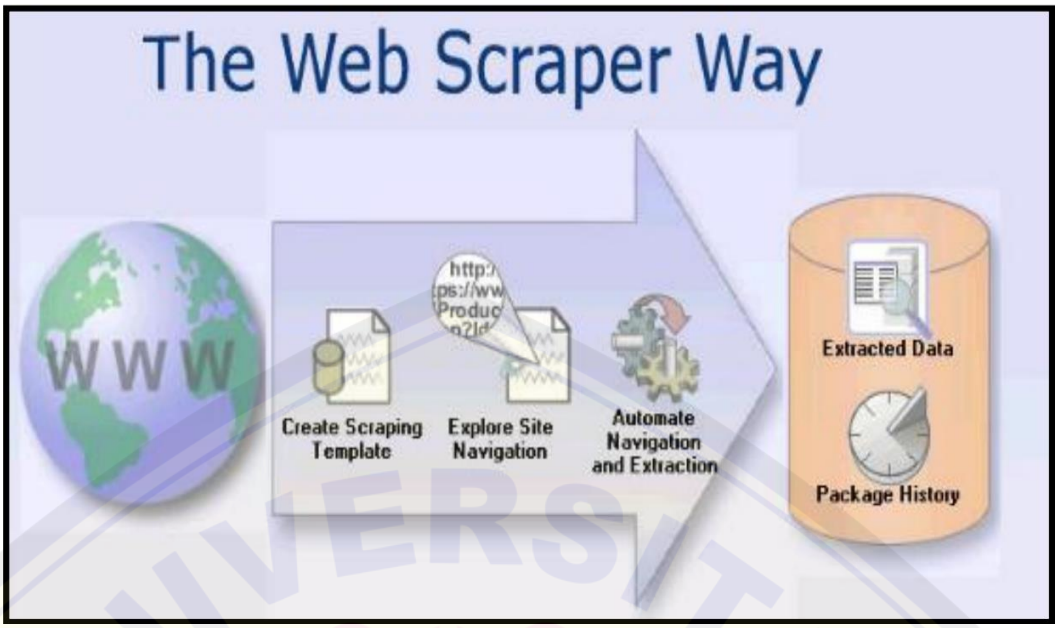
Web scraping merupakan teknik dalam melakukan ekstraksi informasi yang terdapat pada situs website tertentu dan disimpan kedalam sistem atau basis data untuk dijadikan analisa data (Flores et al., 2020). Bentuk dokumen yang diperoleh

biasanya berbentuk bahasa *markup language* seperti HTML atau XHTML. Proses dalam melakukan web scraping dapat dilakukan menjadi dua langkah berurutan, dimulai dari melakukan proses akuisisi sumber data yang dimiliki oleh situs website, lalu ekstraksi data yang diinginkan.

Menurut Turland (2010) Terdapat 4 proses dalam melakukan web scrapping, diantaranya :

1. *Create Scraping Template* ; proses dalam memeriksa struktur web dari dokumen HTML yang terdapat pada situs website tertuju. Berdasarkan tag pada dokumen, informasi penting mungkin didapatkan.
2. *Explore site navigation* ; proses dalam menjelajahi dari situs yang dituju dengan tujuan agar dapat menemukan halaman web yang tepat. Jika halaman yang ditemukan sudah tepat, maka akan dimasukan pada *softwere web scraping.*
3. *Automate navigation and extraction ;* Proses pengambilan informasi pada website yang ditentukan dengan aplikasi *web scrapper* secara otomatis.
4. *Extracted data and package history ;* Proses dalam menyimpan informasi – informasi hasil dari scraping website kedalam database yang akan digunakan pada proses algoritma VSM.

   Untuk lebih jelasnya, berikut ini adalah gambar 2.1 sebagai alur cara pengerjaan proses web sraping,



Gambar 2. 1 Alur proses web scraping

Setelah mendampatkan data yang dibutuhkan untuk melakukan proses penelitian dengan menggunakan teknik web scraping, selanjutnya adalah proses dalam 5melakukan analisa data dengan menguakan konsep ninformation retrieval.

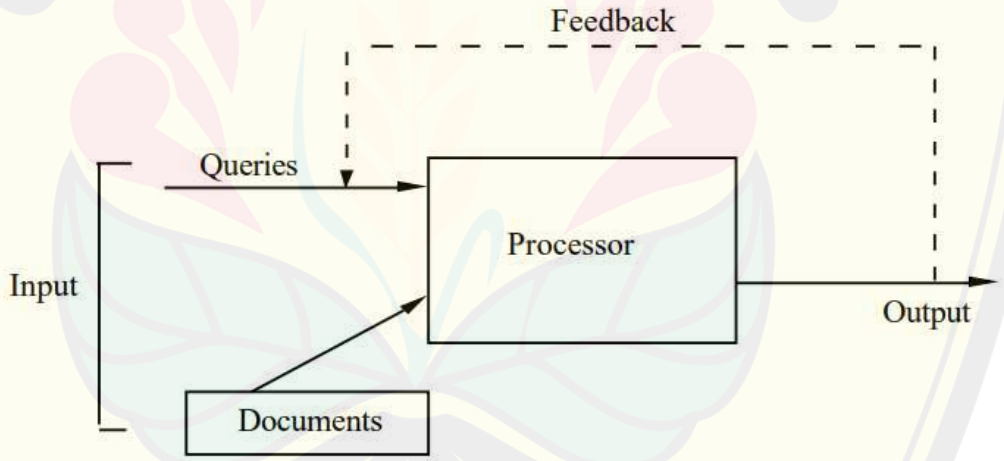
2.3. Information Retrieval

Information retrieval merupakan bidang yang berfokus pada struktur, analisis, ataupun organisasi, penyimpanan, pencarian, dan penemuan kembali dari informasi (Salton, 1968). Information retrieval merupakan sebuah disiplin ilmu dalam mencari dokumen yang relevan dari hasil pencocokan sederhana dengan pola leksikal dalam query (Ceri et al., 2013). Sehingga dapat disimpulkan bahwa information retrieval adalah bidang ilmu yang berfokus dalam menemukan informasi dalam melakukan pencarian dokumen yang relevan berdasarkan kebutuhan pengguna.

Menurut Rijsbergen (1979) menyatakan *Information Retrieval* itu dapat dibagi menjadi tiga kelompok utama yaitu analisis konten, struktur informasi, dan evaluasi. Sehingga, *Information Retrieval* cocok bagi pengguna yang ingin

melakukan pencarian data berupa dokumen yang berasal dari data yang tidak terstruktur.

Rijsbergen (1979) menyatakan Konsep kerja dari sistem *Information Retrieval* cukup simpel, yaitu *user* menginputkan hal yang ingin dicari atau memasukkan *query* ke dalam sistem, kemudian sistem akan memproses dari *query* yang dimasukkan tersebut, lalu sistem akan melakukan pembacaan dokumen secara keseluruhan dengan cara mengubah kata-kata pada dokumen menjadi kata kata dasar dan mengubah isi dokumen menjadi kata-kata yang dianggap penting saja. Lalu, sistem akan mencocokkan tiap-tiap kata pada masing masing dokumen dengan *query* yang dimasukkan oleh pengguna, kemudian sistem akan memberikan jawaban berupa dokumen yang memiliki banyak kata yang sesuai dari *query* yang dimasukkan sebelumnya. Berikut ini adalah gambaran dari proses *Information Retrieval* menurut Rijsbergen (1979).



Gambar 2.1 Alur Proses Information Retrieval menurut C. J. van RIJSBERGEN (1979)

McCaren & Leiter (1973) menyatakan terdapat beberapa kelebihan dari *Information Retrieval* diantaranya adalah pada saat information Retrieval ini *online*, pengguna dapat mengubah isi dari query yang diinputkan dalam tiap satu sesi. Tahapan dalam *information retrieval* dimulai dari melakukan text preprocessing,

yaitu tahapan dalam mempersiapkan dokumen sebelum dapat diolah (Wicaksono et al., 2016).

2.4. Text Preprocessing

Proses *text preprocessing* diperlukan karena dokumen teks tidak dapat diproses langsung oleh algoritma pencarian, sehingga diperlukan persiapan untuk menghasilkan data numerik yang dapat diproses dalam perhitungan (Basmalah Wicaksono et al., 2016). Berdasarkan jurnal oleh Ahmad Fauzi dan Ginabila (2016) menyatakan tahapan *text preprocessing* pada penelitian terdiri dari Case folding, Tokenizing, Stop Word Removal, dan Stemming.

2.4.1. Case Folding

Case folding adalah proses dalam mengkonversikan semua text dokumen menjadi bertulisan kecil/lowercase (Arafah et al., 2018). Tujuan dari melakukan case folding adalah untuk mengurangi dimensi data, sehingga dapat meningkatkan kekuatan statistik data dan tidak mengurangi keabsahan data (Pennebaker, Booth, et al., 2015). Setelah menjalani proses case folding, selanjutnya adalah tahapan dalam mengubah teks menjadi kata atau disebut juga proses tokenizing.

2.4.2. Tokenizing

Tokenizing adalah sebuah proses dalam mengubah bentuk teks menjadi kata, frasa, atau arti kata lainnya, yang disebut dengan token (Uysal & Gunal, 2014). Proses melakukan tokenizing adalah dengen memotong kalimat menjadi kata kata dan menghilangkan karakter tanda baca (Erin et al., 2007). Menurut Vijayarani dan Janani (2016) pada jurnalnya Proses tokenizing itu penting karena secara general data tekstual hanyalah kumpulan kalimat yang masih berada pada tahap awal, sedangkan semua proses dalam melakukan analisis membutuhkan kata-kata yang tersedia pada dataset. Oleh karena itu, diperlukannya penguraian kata dengan melakukan tokenisasi dokumen. Setelah membentuk teks mejadi token- token, langkah selanjutnya adalah proses stopword removal.

2.4.3. Stop Word Removal

Stop word removal adalah proses pemilihan kata-kata penting atau kata-kata yang dapat digunakan untuk mewakili dokumen (Pasnur et al., 2018). Kebanyakan dari kata kata yang dihilangkan merupakan jenis fungsi, Seperti kata penghubung.

Proses stop word removal diperlukan karena kata-kata yang dianggap tidak memiliki arti ini dapat meningkatkan waktu pemrosesan data lebih lama oleh algoritma dan terkadang memiliki efek kontra produktif dalam proses *text mining* (Alvriyanto et al., 2020). Selanjutnya masuk dalam tahapan stemming.

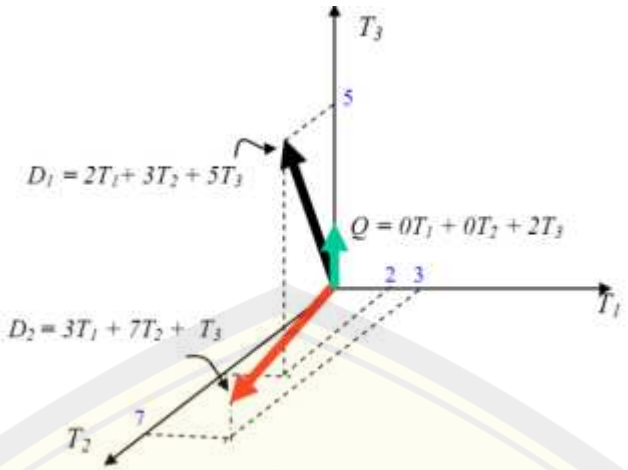
2.4.4. Stemming

Stemming merupakan proses dimana semua kata yang telah terkurasi dari proses sebelumnya, diubah menjadi kedalam bentuk dasarnya dengan menghilangkan *suffix* dan *prefix* nya, contoh katanya adalah 'membuka', 'terbuka', 'pembuka' diubah bentuknya menjadi 'buka' (Alvriyanto et al., 2020). Tujuan dari proses stemming adalah untuk mendapatkan kata dasar (stem), atau bentuk dasar (root), dan bentuk kata turunan dari masing kata (Uysal & Gunal, 2014). Dengan harapan proses *stemming* dapat meningkatkan sensitivitas dan kemampuan untuk mencari pada dokumen yang relevan (Alvriyanto et al., 2020).

Setelah melewati proses *stemming* akan menghasilkan data dokumen baru yang lebih mudah untuk dilakukan analisa pada *Vector Space Model*. Sehingga langkah berikutnya adalah implementasi *vector space model.*

2.5. Vector Space Model

Vector Space Model merupakan sebuah representai dokumen dan query sebagai vektor di dalam ruang dimana setiap dimensi berkorespondensi terhadap tiap term dari masing masin kosakata (Ceri et al., 2013). Pada *Vector Space Model*, setiap bobot kata dalam sebuah dokumen menunjukan seberapa mirip dokumen tersebut dengan *query* yang dimasukkan. Pengukuran relevansi kesamaan dokumen dengan *query* dipandang sebagai pengukuran kesamaan antara vektor dokumen dengan vektor *query* (Wicaksono et al., 2016). Contoh representasi relevansi antara dokumen dengan *query* tergambar pada Gambar 2.2. Q merupakan query pembanding. D_1 dan D_2 merupakan dokumen yang akan dibandingkan. Sedangkan T_1, T_2, dan T_3 adalah *term* pada setiap dokumen tersebut



Gambar 2.2 Representasi dokumen dan query pada ruang vektor (Turney & Patel, 2010)

Menurut Ahmad Fauzi & Ginabila (2018) terdapat empat langkah-langkah dalam melakukan perhitungan *Vector space model* yaitu dimulai dengan menghitung bobot dokumen dengan menggunakan tf-idf, menghitung jarak dokumen dengan query (menghitung magnitude), menghitung Dot Product Sum, dan terakhir menghitung similarity cosine.

2.6. Perhitungan TF-IDF

TF-IDF merupakan salah satu cara untuk memberikan bobot antara suatu kata dengan suatu dokumen (Robertson, 2005). Menurut Alviryanto (2020) terdapat dua konsep perhitungan pada algoritma TF-IDF, yaitu algoritma frekuensi kemunculan sebuah kata didalam suatu dokumen (TF) dan *inverse* frekuensi dokumen yang mengandung kata tersebut (IDF).

TF atau disebut juga *Term Frequency* adalah sebuah metode perhitungan bobot dari setiap *term* pada suatu dokumen. *term* adalah bobot hubungan suatu kata. Perhitungan TF dilihat dari seberapa banyak frekuensi *term* dari suatu dokumen. Semakin banyak frekuensi *term* pada sebuah dokumen, Semakin tinggi nilai TF nya. Rumus dari TF seperti pada Persamaan (1).

$$tf = tf_{ij} \quad (1)$$

Keterangan :

tf_{ij} = banyaknya kemunculan term t_{ij} dalam dokumen d_j

tf = term frequency

Sedangkan IDF adalah *inverse* dari frekuensi dokumen. Pada IDF, kata yang paling sedikit muncul pada suatu dokumen memiliki nilai yang tinggi. Sehingga rumus dari IDF adalah

$$IDF(i) = log\left(\frac{N}{df(i)}\right) \quad (2)$$

Dimana:

N = total seluruh dokumen

$df(i)$ = total dokumen yang memiliki *term* t

Gabungan dari nilai TF dan IDF digunakan untuk menentukan hasil bobot term tiap kata. Tujuannya, agar kombinasi dari perhitungan TF-IDF dapat meningkatkan performa dari algoritma text mining (Alvriyanto et al., 2020). Sehingga menghasilkan rumus yang pembobotan dokumen, adapun rumusnya adalah sebagai berikut.

$$W(i,j) = TF(i,j)x\ log\ \frac{N}{df(i)} \quad (3)$$

Keterangan :

$W(i,j)$ = bobot dokumen

$TF(i,j)$ = banyaknya kemunculan term t_i pada dokumen d_j

N = jumlah dokumen yang terambil oleh sistem

Setelah mengetahui hasil nilai pembobotannya atau nilai TF-IDF nya, selanjutnya adalah tahap dalam melakukan perhitungan jarak atau dikenal dengan perhitungan magnitude.

2.7. Perhitungan Cosine Similarity

Perhitungan *Cosine similarity* dimulai dari tahapan perhitungan *Dot Product*. Perhitungan *Dot Product* adalah perhitungan masing masing kata untuk melihat derajat kemiripan antar dokumen teks yang tersedia dengan query (Fauzi & Ginabila, 2018). Hasil dari perhitungan *Dot Product* akan mempengaruhi dari hasil *cosine similarity* (Fauzi & Ginabila, 2018). Sehingga, perhitungan dot product dapat menentukan hasil *cosine similarity* secara signifikan. Selain itu, perhitungan Cosine Similairity diperlukan karena jika hanya tepaut pada menghitung jarak antara dokumen dengan query saja, akan menimbulkan hasil pengukuran yang bias (Pykes, 2020).

Cosine Similarity adalah metode yang digunakan untuk mengetahui kemiripan antara vektor dokumen dengan vektor query yang berdasarkan dari sudut yang paling kecil (Wicaksono et al., 2016). Dalam melakukan perhitungan *cosine similarity* ukuran kesamaan antara dua vektor dalam ruang dimensi diperoleh dari nilai kosinus sudut dikalikan dengan nilai dua vektor yang dibandingkan (Pasnur et al., 2018). Pada perhitungan vector space model, dokumen dikatakan semakin mirip dengan *query* apabila memiliki nilai sudut yang semakin kecil . Jika semakin kecil nilai sudutnya, semakin mirip nilai antara kedua vektor tersebut (Alvriyanto et al., 2020). Untuk rumus dari *Cosine Similarity* adalah

$$Sim(d\,.d_i) = cos\ cos\ \theta$$

$$Sim\ (q.d_j) = \frac{q.d_j}{|q||d_j|}$$

$$Sim\ (q.d_j) = \frac{\sum_{i=1}^{t} W_{i,q} W_{i,j}}{\sqrt{\sum_{j=1}^{t} (w_{iq})^2}\sqrt{\sum_{i=1}^{t} (W_{ij})^2}} \quad (4)$$

Dengan ketentuan :

q = Jumlah bobot query

d_i = bobot dokumen

$Sim\ (q.d_j)$ = similaritas antara kueri dan dokumen

W_{ij} = Bobot *term* j pada dokumen

W_{iq} = Bobot *query*

Pada akhirnya, hasil yang didapatkan dari perhitungan cosine similarity ini adalah nilai cosinus dari masing masing dokumen yang menjadi nilai untuk mengurutkan kesamaan antara dokumen dengan *query* yang diinputkan. Tetapi, hasil yang dihasilkan oleh dengan menggunakan *vector space model* ini juga perlu dilakukan pengujian kualitas pengukurannya, tujuannya adalah agar hasil yang diberikan pada Penggunaan vector space model bisa diketahui nilai kredibilitasnya. Salah satu metode untuk mengetahui kredibilitas hasil pencarian adalah dengan menggunakan perhitungan *Precision* dan *Recall*

2.8. K-Modes Clustering

K-Modes clustering merupakan bentuk pengembangan dari algoritma K-Means clustering (Badrutaman et al., 2020). K-Modes clustering pertama kali dikenalkan pada tahun 1977 oleh Huang. Algoritma K-Modes clustering berfungsi untuk melakukan clustering untuk data bebentuk kategorik, dikarenakan pada data kategorik tidak dapat diketahui nilai jarak euclidean nya (Badruttaman, 2020). Dalam implementasinya, terdapat 5 teknik dalam melakukan clustering dengan menggunakan metode K-Modes yaitu (Chaturvedi., 2001):

1. Menghitung jarak antar subjek pada kode *dummy,* dan menggunakan prosedur pengelompokan hierarkis seperti hubungan tunggal, lengkap, atau rata-rata pada jarak antarsubjek yang diturunkan
2. Menggunakan konsep k-means dalam mengimplementasikan variable dari data yang berbentuk katagorial.
3. Menggunakan analsis korespondensi untuk mendapantkan koordinat spasial setiap subjek, dan dengan menggunakan K-means untuk menurunkan koodinat spasial.
4. menggunakan prosedur kelas laten yang tersedia untuk melakukan analisis tabel kontingensi
5. Menggunakan Algoritma Ditto hartigan untuk data kategorikal

Setelah mengclustering data yang akan dievaluasi, selanjutnya adalah masuk dalam tahapan evaluasi hasil pencarian.

2.9. Evaluasi hasil pencarian

Confussion matrix merupakan salah satu alat dalam melakukan analisa performa. Cara Kerja Confusion Matrix adalah dengan merangkum classifier dari setiap pengujian klasifikasi dengan berbagai uji coba data (Shultz & Fahlman, 2017). Confusion matrix berbentuk matrix 2 dimensi dimana salah satu dimensinya merupakan kelas yang sebenarnya, dan dimensi yang lainnya terdiri dari kelas yang dihasilkan/diprediksi. Sehingga akan membentuk maktriks sesuai dengan gambar 2.3.

		Assigned class	
		Positive	Negative
Actual class	Positive	TP	FN
	Negative	FP	TN

Gambar 2.3 Bentuk confusion matrix dengan matriks 2 dimensi

Dari gambar 2.3. terdapat 4 cel yang bernama TP, FP, TN, dan FN. Arti dari TP merupakan True Positive (TP), yang berarti hasil prediksi menghasilkan hasil yang benar (positif) dari keadaan yang benar. Pada FP merupakan False Positive (FP) yang berarti hasil prediksi dinyatakan benar (positif) dari keadaan yang salah. Selanjutnya pada TN merupakan True Negative (TN) yang berarti hasil prediksi dinyatakan salah (negatif) dari keadaan yang salah. Sedangkan FN meripakan False Negative (FN) yang berarti hasil prediksi dinyatakan salah dari keadaan yang benar. Setelah mengetahui klasifikasi confusion matriks selanjutnya dapat diketahui nilai dari *precision* dan *recall* yang menjadi nilai untuk mengerahui keakuratan hasil datamining.

Precision merupakan salah satu kuantitas yang biasa digunakan dalam melakukan pengukuran performa dari sistem *Inforomation Retrieval* (Hidayat, 2013). *Precision* digunakan untuk mengukur seberapa baik kinerjanya dalam menolak dokumen yang tidak relevan (Croft et al., 2010). Menurut Basmalah et al. (2016) mengatakan bahwa *Precision* adalah perbandinan antara hasil pencarian yang

relevan terhadap semua pencarian yang berhasil di temukan. Sedangkan, *Recall* adalah seberapa banyak dokumen relevan yang didapatkan pada saat melakukan pencarian (Hidayat, 2013). Menurut Wicaksono et al. (2016) *recall* adalah perbandingan antara hasil pencarian yang relevan dengan seluruh data relevan yang ada pada koleksi *database.* Precision dan recall bertujuan untuk mengukur seberapa besar keberhasilan pencarian yang diperoleh. Semakin tinggi nilai *precision* dan *recall*-nya, semakin bagus strategi pencariannya (Wicaksono et al., 2016).

Dalam melakukan Precision dan Recall, terdapat rumus yang dapat digunakan untuk menghitungnya,yaitu

$$Recall = \frac{|A \cap B|}{|A|} \quad (6)$$

$$Precision = \frac{|A \cap B|}{|B|} \quad (7)$$

Keterangan:

A = Dokumen yang relevan dengan query

B = Dokumen yang diambil

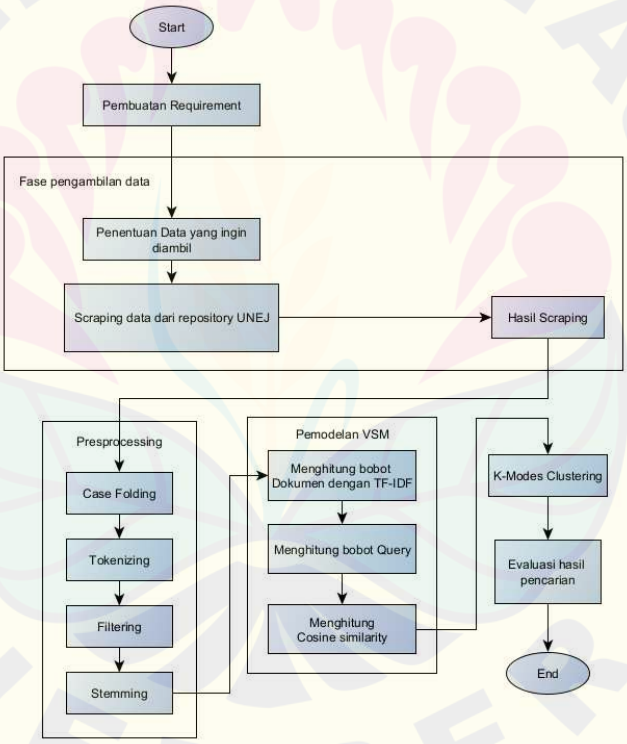
BAB 3. METODOLOGI PENELITIAN

3.1. Jenis Penelitian

Jenis penelitian ini adalah penelitian secara kuantitatif. Terdapat beberapa jenis penelitian kuantitatif, tetapi untuk penelitian ini menggunakan jenis penelitian kuantitatif deskriptif. Jenis tersebut dipilih karena pada penelitian ini akan memaparkan hasil penelitian ini sesuai dengan hasil perhitungannya.

3.2. Tahapan Penelitian

Tahapan penelitian yang akan dilakukan pada penelitian ini tercantum pada gambar berikut



Gambar 3. 1 Tahapan Penelitian

3.3. Penentuan Requirement

Langkah awal sebelum melakukan pembuatan sistem adalah tahapan penentuan *requirement*. Pembuatan *requirement* berfungsi untuk mengetahui apa saja yang

dibutuhkan dan bagaimana kita dapat mengkomunikasikan atau menghubungkan tiap komponen fungsi sistem ke dalam sistem agar terintegrasi secara baik.

Proses penentuan requirement dimulai menganalisis halaman website yang dituju untuk dijadikan bahan penelitian. Hal yang dianalisis dimulai dari elemen apa saja yang terdapat pada halaman website, model apa saja yang terdapat didalam website, dan juga bagaimana cara penulisan link pada halaman website tersebut. Selanjutnya adalah menentukan data apa saja yang akan diteliti.

Setelah menentukan data yang ingin diteliti dan mengetahui hal - hal apa saja yang terdapat didalam website. Selanjutnya adalah menentukan teknik apa yang akan digunakan dalam pengambilan data, pada penelitian ini teknik yang akan digunakan untuk mengambil data adalah dengan menggunakan teknik web scraping. Kemudian menentukan bahasa pemprograman yang akan digunakan dan menentukan aplikasi apa saja yang akan digunakan untuk melakukan penelitian. Pada penelitian ini, bahasa pemrograman yang akan digunakan adalah dengan menggunakan python dan aplikasi yang digunakan yaitu google colab sebagai teks editor.

3.4. Pengambilan Data

Proses pengambilan data dimulai dari mengimport kebutuhan – kebutuhan untuk dilakukan proses web scraping diataranya library *BeautifulSoup*, *request*, dan library *padas*. Library *BeautifulaSoup* digunakan untuk melakukan proses pengambilan data dari file dengan jenis HTML dan XML, sedangkan library *request* digunakan melakukan proses pemanggilan pada format HTTP , dan library *pandas* digunakan untuk melakukan proses pengolahan data pada python.

Setelah mengimport semua kebutuhan untuk melakukan web scraping, tahap selanjutnya adalah tulis atau salin link website yang akan dijadikan target untuk diambil datanya, pada penelitian ini website yang digunakan adalah website Repository UNEJ pada bagian skripsi mahasiswa. Kemudian, hubungkan akses website dengan python menggunakan perintah *requests*. Sehingga website akan terhubung dengan python untuk dilakukan pengolahan data.

Setelah itu, dengan menggunakan perintah *html parser* dari *beatifulsoup*, website akan memberikan teks HTML yang termuat didalam website. Teks HTML inilah yang akan dilakukan pengolahan agar dapat menampilkan data-data yang diinginkan, dengan cara memanggil setiap tag – tag dari data yang dibutuhkan. Pada penelitian ini data – data yang akan diambil diantaranya adalah data judul, abstrak, nama penulis, fakultas, dan tanggal upload. Sedangkan untuk tag – tag yang digunakan untuk mendapantkan data data tersebut diantaranya:

- Data judul menggunakan tag *h2* dengan nama class *page-header first-page-header*
- Data abstrak menggunakan tag *div* dengan nama class *simple-item-view-description item-page-field-wrapper table*
- Data nama penulis menggunakan tag *div* dengan nama class *simple-item-view-authors item-page-field-wrapper table*
- Data fakultas menggunakan tag *ul* dengan nama class *ds-referenceSet-list*
- Data tanggal upload menggunakan tag *div* dengan nama class *simple-item-view-date word-break item-page-field-wrapper table*

Selanjutnya, dilakukan proses pencarian dan pengambilan data dengan menggunakan logika *looping* dan dikombinasikan dengan perintah – perintah dari *beautiufulsoup.* Untuk membudahkan pengambilan data, dapat menggunakan perintah *try* dan *except.* Hasil dari kombinasi tersebut akan menghasilkan database yang berasal dari website yang dituju. Tahapan terkahir adalah export data yang didapatkan kebentuk CSV untuk dilakukan proses *preprocessing*.

3.5. Preprocessing

Tahapan *Preprocessing* dimulai dari tahapan *case folding,* kemudian tahap *tokenizing,* kemudian tahap *stop word removal,*dan terakhir adalah tahap *stemming.*

3.5.1. Case Folding

Seperti yang telah dijelaskan sebelumnya, bahwa tujuan case folding adalah menormalisasikan data dengan cara membentuk satu format yang sama. Proses yang dilakukan pada penelitian ini adalah dengan mengkonversi seluruh teks menjadi huruf kecil (*lowercase*). Contohnya adalah “Universitas Jember memiliki

fasilitas perpustakaan yaitu UPT-perpustakaan." akan berubah menjadi "universitas jember memiliki fasilitas perpustakaan yaitu upt perpustakaan". Pada penelitian ini, cara melakukan konversi teks menjadi huruf kecil adalah dengan menggunkanan perintah *str.lower()* yang digabungkan kedalam pandas dataframe, sehingga menjadi *df['nama kolom yang dituju'].str.lower.*

3.5.2. Tokenizing

Dalam penelitian ini, proses pembuatan token menggunakan library *nltk* (natural languages Toolkit) dimana sudah tersaji perintah *word_tokenize*. Tetapi sebelum dapat melakukan proses tokenisasi hendaknya dilakukan beberapa penyesuaian data diantaranya menghapus *punctutuion*, *whitespace*, dan *single character*. Setelah itu baru dapat melakukan implementasi perintah *word_tokenize.* Diawali dengan memanggil perintah *word_tokenize* melalui library nltk. Setelah itu, membuat sebuah fungsi yang bertujuan untuk menjalankan perintah *word_tokenize* kemudian return hasil fungsi dengan membuat sepuah *looping* yang akan mengecek hasil dari tokenisasi merupakan bentuk alphabet, dengan perintah *isalpha*. Setelah itu baru di *apply* pada dataframe yang dituju. Agar lebih jelas, berikut ini adalah bentuk kode program untuk menjalankan perintah *word_tokenize* menggunakan pandas dataframe*: df['nama kolom']=df.apply(lambda x: tokenize(x['kolom tujuan']), axis=1).*

3.5.3. Stop Word Removal

Pada penelitian ini, Penggunaan stopword removal menggunakan libarary *nltk.* Proses penghpusan kata kata ini dilakukan dengan menggunakan perintah *stopwords* dari library *nltk*. Kemudian memilih jenis bahasa yang akan digunakan pada proses stopword, pada penelitian ini, stopword removal menggunakan bahasa Indonesia, sehigga bentuk kodenya menjadi *stop_words = set(stopwords.words('indonesian')).* Setelah itu, membuat fungsi baru untuk membuat proses stopword removal, pada penelitian ini fungsi tersebut diberi nama *stopwords_removal* dan diakhiri dengan *return* kata kata yang tidak ada didalam list stopword. Selanjutnya, gunakan fungsi *stopwords_removal* untuk diaplikasikan pada pandas dataframe agar proses stopword dapat dilaksanakan. Untuk lebih jelasnya, bentuk kodenya menjadi *df['nama kolom'].apply(stopwords_removal).*

3.5.4. Stemming

Pada penelitian ini proses stemming dilakukan dengan menggunakan library *sastrawi*. Untuk mengimplementasikannya, hal yang pertama dilakukan adalah memanggil library *sastrawi* lalu import perintah *stemmerfactory*, selanjutnya membuat method untuk menempatkan *stemmerfactory* tersebut, pada penelitian ini diberinama method *factory*. Kemudian membuat *stemmer* yang berfungsi untuk membuat perintah *create_stemmer()*. Setelah melakukan proses persiapan untuk *library sastrawi*, selanjutnya membuat persiapan dalam melakukan proses stemming. Dimulai dari membuat fungsi yang bernama *stemmed_wrapper()* yang berfungsi untuk melakukan proses stemming. setelah itu membuat list kosong untuk menampung kata-kata hasil *stemming*. kemudian dengan bantuan fungsi *looping,* gunakan untuk melakukan proses *stemming*.

3.6. Pemodelan Vector Space Models

Pemodelan *Vector Space Model* terdiri dari tahapan perhitungan TF-IDF, Perhitungan nilai *magnitude,* perhitungan dot product, menghitung *similarity cosine.* Selain itu, pada proses perhitungan *Vector Space Model* akan ada *query processing* yang berfungsi untuk menghitung nilai vector query pada dokumen, dimana tujuannya untuk menentukan nilai vector terkecil yang hasilkan dari masing masing dokumen. nilai inilah yang menjadi dasar dalam pemodelan *Vector Space Model.* Langkah pertama adalah menghitung TF-IDF.

3.6.1. Perhitungan TF-IDF

Perhitungan TF-IDF didapatkan dari hasil perkalian antara nilai TF dengan nilai IDF, sehingga dapat sesuai dengan Persamaan 3. Pada penelitian ini, perhitungan TF-IDF dilakukan dengan menggunakan perintah *tfidfvectorizer* dari *sklearn*. Langkah yang dilalui yaitu, membut *vectorizer* terlebih dahulu yang berisi *tfidfvectorizer,* hal ini dilakukan untuk mempermudah dalam melakukan proses perhitungan tf-idf. Selanjutnya panggil data yang akan diubah kebentuk vektor dengan menggunakan *vectorizer.fit_transform.* Selanjutnya buat dataframe baru yang akan berisi data nilai vektori pada masing – masing kata di setiap dokumen. lalu, print databasenya.

3.6.2. Perhitungan *Magnitude*

Pada penelitian ini, perhitungan magnitude dilakukan dengan cara menghitung nilai vector pada query. Langkah yang dilakukan dimulai dari membuat method yang berfungsi untuk memasukkan query. Kemudian query yang telah diinputkan akan dilakukan proses *cleaning* terlebih dahulu dimulai dari proses tokenizing, stopword removal, dan stemming. Untuk proses tokenizing penelti menggunakan perintah *word_tokenize* seperti halnya pada proses tokenizing pada dokumen. selanjutnya, untuk proses stopword peneliti menggunakan perintah *stopwords* dari *nltk* seperti halnya pada proses stopword pada dokumen dokumen. Selanjutnya adalah proses stemming, pada proses stemming disini peneliti menggunakan *sastrawi.* Langkah – langkah yang dilakukan dimulai dari membuat list kosong untuk menyimoan hasil stemming, kemudian membuat *looping* untuk proses *stemming* kata pada query, selanjutnya didalam looping tersebut gunakan perintah *sastrawi* untuk proses stemming. lalu, tambahkan hasil stemming dengan *append* pada list kosong yang sudah dibuat sebelumnya. Sesudah dilakukan proses *cleaning* pada query, selanjutnya adalah proses mengubah kata – kata pada query menjadi bentuk vector. Proses ini menggunakan perintah *tfidfvectorizer* dari *sklearn* maka hasil yang didapatkan adalah berupa bentuk vector dari query yang sebelumnya diinputkan.

3.6.3. Cosine similarity

Perhitungan *cosine similarity* dilakukan juga sekaligus perhitungan *dot product*. Hal ini terjadi karena perhitungan *dot product* merupakan satu bagian dari rumus perhitungan *cosine similarity.* Pada penelitian ini menggunakan perintah *cosine_similarity* dari *sklearn.* Proses nya dimulai dari memanggil perintah *cosine similarity* dari *sklearn*, lalu membuat method untuk melakukan proses perhitungan cosine similarity dan tentukan parameter apa saja yang dibutuhkan, pada penelitian ini untuk parameter yang dibutuhkan terdiri dari data dokumen yang telah dilakukan proses *preprocessing* dan input query yang telah melalui proses *cleaning query*. Sehingga bentuk programnya adalah *cosineSimilarities = cosine_similarity("database dokumen", "query yang telah dibersihkan").flatten()*. Fungsi flatten disini adalah untuk mengubah data kebentuk satu dimensi.

3.7. K-Modes Clustering

Sebelum melakukan evaluasi pencarian, peneliti membuat clustering terlebih dahulu dari data hasil perhitungan *vector space model* dan data aslinya. Untuk implementasinya sendiri, peneliti menggunakan library *kmodes.* Cara implementasinya adalah dengan membuat fungsi yang berisi perintah dari *kmodes* dan masukkan nialai dari clustering yang diinginkan serta jumalah iterasi maksimal yang dikehendaki. Lalu, dengan menggunakan fungsi yang telah kita buat tersebut, kita dapat membuat nilai clusternya pada database yang dituju dengan cara menggunakan perintah *fit predict*. Lalu, agar data yang kita buat tidak hilang dapat dibuatkan *dictionary* yang berisi nilai cluster dari setiap dokumen, gunakan *looping* untuk melakukan proses pemasukan data dengan secara otomatis. Setelah itu dapat dilakukan tahapan evaluasi hasil pencarian.

3.8. Evaluasi hasil pencarian

Pada tahapan evaluasi hasil pencarian, peneliti menggunakan metode *precision* dan *recall*. Pada penelitian ini, proses dimulai dari meng-*import* library yang dibutuhkan, dimulai dari import *matplotlib*, *confusion matrix*, *pandas, dan seaborn*. Setelah mengimport dan meninstall dari library yang dibutuhkan, selanjutnya adalah membuat fungsi untuk men-visualisasi penggunaan *confusion matrix*, kemudian membuat *array* yang berisi hasil prediksi data sesungguhnya. Lalu tampilkan dengan memanggil perintah hasil pendefinisian *confusion matrix*, dan terakhir adalah memanggil perintah *classification report* untuk mengetahui hasil dari nilai *precision* dan *recall* nya.

BAB 4. HASIL DAN PEMBAHASAN

4.1. Pembuatan Requirement

Proses pembuatan requirement dimulai dari menganalisis dari situs tujuan penelitian yaitu Repository UNEJ. Hal yang dianalisis pada repository UNEJ dimulai dari tampilan halaman muka pada website, menu - menu yang tersedia pada website, dan juga bentuk penulisan link pada website. Selanjutnya, menentukan data apa saja yang akan diambil pada Repository UNEJ. Pada penelitian ini, peneliti bertujuan untuk membuat model pencarian dokumen skripsi mahasiswa berdasarkan abstrak dan judul dari skripsi mahasiswa. Setelah melakukan proses persiapan, selanjutnya adalah masuk kedalam tahap web scraping data pada Repository UNEJ.

4.2. Pengambilan data

Pada proses pengambilan data, diawali dengan analisis layout tampilan website repository UNEJ. Hal ini bertujuan untuk mengetahui letak dan posisi dari data yang ingin didapatkan. Berikut ini adalah hasil analisa layout dari dokumen skripsi mahasiswa pada repository UNEJ yang tersaji pada gambar 4.1.

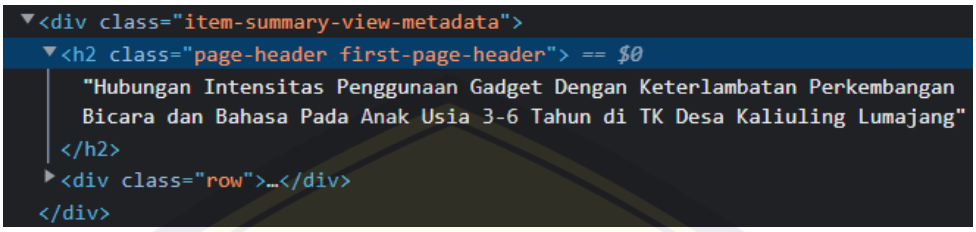
Gambar 4. 1 Analisa tampilan dokumen skripsi pada repository UNEJ

Pada gambar 4.1 terdapat berbagai macam bagian dari layout website repository UNEJ. Setelah mengetahui posisi dari masing – masing data, selanjutnya adalah menganalisis penamaan tag dari masing – masing data tersebut. Perlu diketahui, untuk pengambilan

data penelitian ini hanyalah data berupa teks saja yang akan diambil. Berikut ini adalah nama data dan penamaan tag pada data yang akan diambil.

A. Judul skripsi

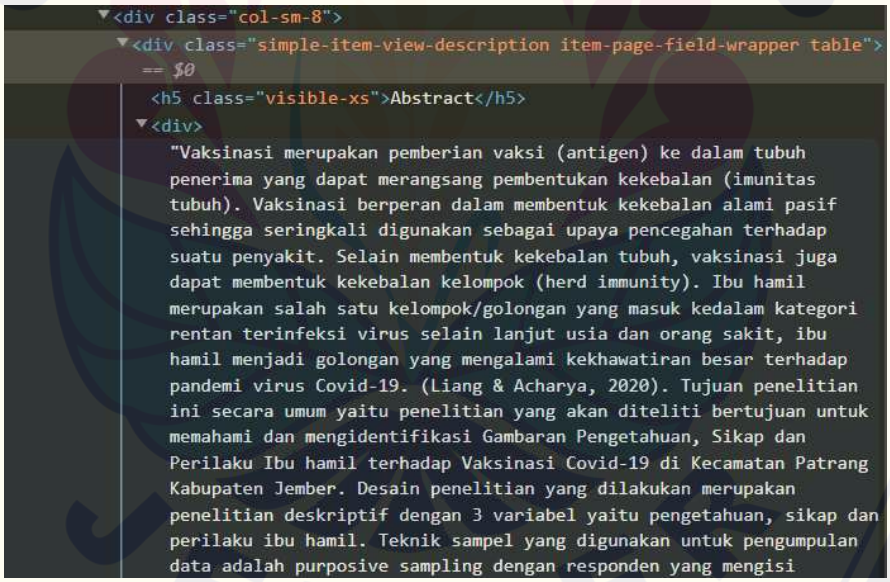
Pada data judul skripsi, penggunaan tag yang digunakan untuk menggambil data judul skripsi adalah tag *h2* dengan nama class *page-header first-page-header*



Gambar 4. 2 Tag pada HTML bagian Judul

B. Abstrak skripsi

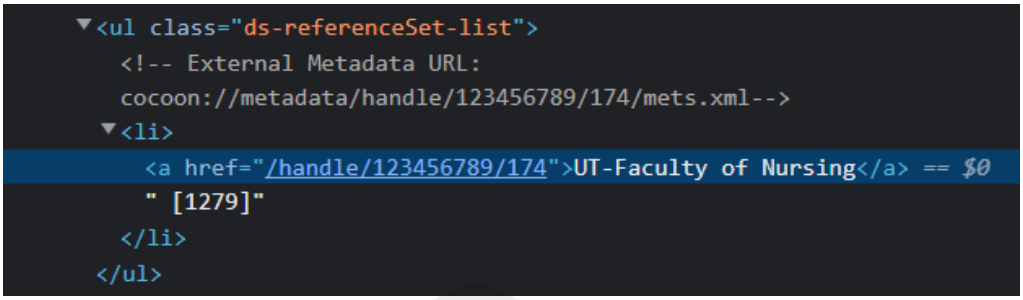
Pada data abstrak skripsi, pengguanan tag yang digunakan untuk menggambil data abstrak adalah tag *div* dengan nama class *simple-item-view-description item-page-field-wrapper table*



Gambar 4. 3 Tag pada HTML bagian Abstrak

C. Nama fakultas

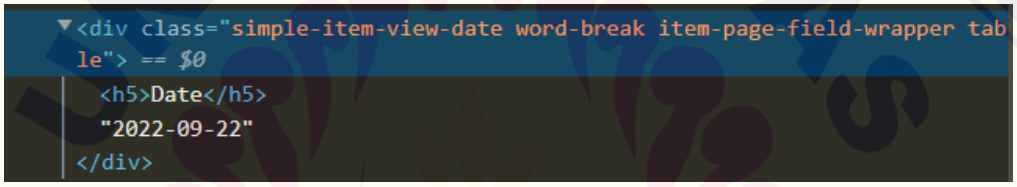
Pada data fakultas, pengguanan tag yang digunakan untuk menggambil data fakultas adalah tag *ul* dengan nama class *ds-referenceSet-list*



Gambar 4. 4 Tag pada HTML bagian nama fakultas

D. Tanggal terbit

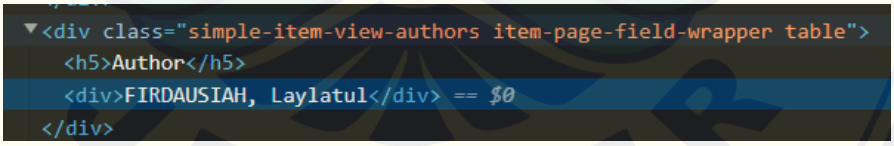
Pada data tanggal terbit, pengguanan tag yang digunakan untuk menggambil data tanggal terbit adalah tag *div* dengan nama class *simple-item-view-date word-break item-page-field-wrapper table*



Gambar 4. 5 Tag pada HTML bagian tanggal terbit

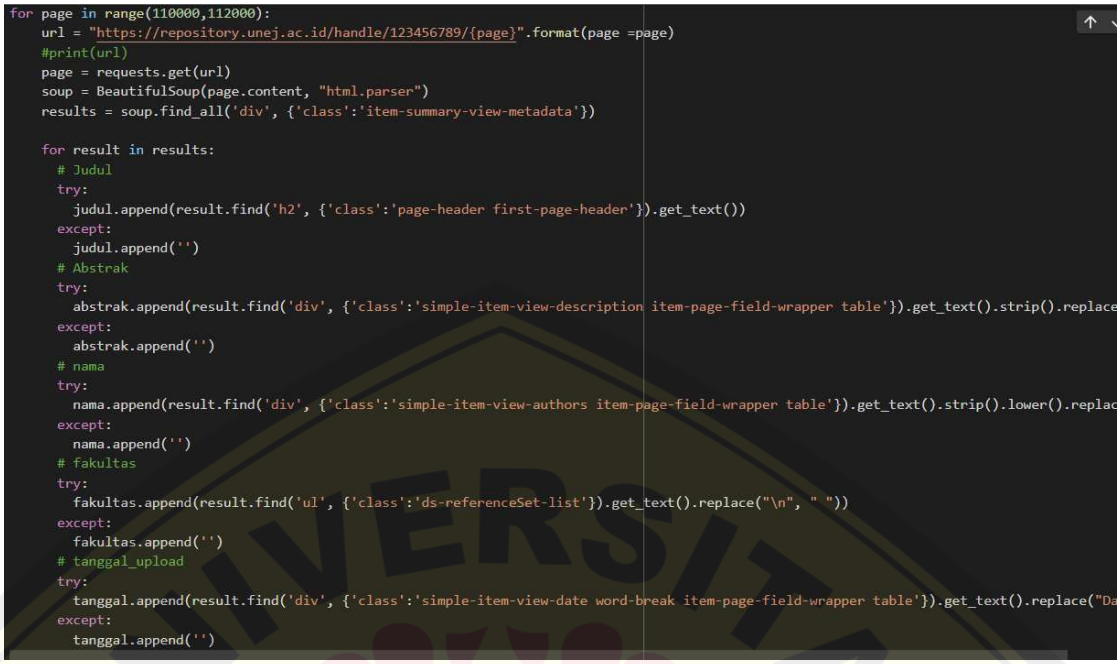
E. Nama penulis

Pada data nama penulis, pengguanan tag yang digunakan untuk menggambil data tanggal terbit adalah tag *div* dengan nama class *simple-item-view-authors item-page-field-wrapper table*



Gambar 4. 6 Tag pada HTML bagian nama penulis

Setelah mendapatkan nama – nama tag dari masing – masing data yang diambil, selanjutnya gunakan logika *looping* untuk proses pengambilan data dan manfaatkan perintah *try* dan *except* untuk manangani kesalahan proses eksekusi kode program. Berikut ini adalah hasil dari bentuk penulisan kode untuk implementasi *try* & *except* dan logika *looping*nya yang tersaji pada gambar 4.7.



```
for page in range(110000,112000):
    url = "https://repository.unej.ac.id/handle/123456789/{page}".format(page =page)
    #print(url)
    page = requests.get(url)
    soup = BeautifulSoup(page.content, "html.parser")
    results = soup.find_all('div', {'class':'item-summary-view-metadata'})

    for result in results:
      # Judul
      try:
        judul.append(result.find('h2', {'class':'page-header first-page-header'}).get_text())
      except:
        judul.append('')
      # Abstrak
      try:
        abstrak.append(result.find('div', {'class':'simple-item-view-description item-page-field-wrapper table'}).get_text().strip().replace
      except:
        abstrak.append('')
      # nama
      try:
        nama.append(result.find('div', {'class':'simple-item-view-authors item-page-field-wrapper table'}).get_text().strip().lower().replac
      except:
        nama.append('')
      # fakultas
      try:
        fakultas.append(result.find('ul', {'class':'ds-referenceSet-list'}).get_text().replace("\n", " "))
      except:
        fakultas.append('')
      # tanggal_upload
      try:
        tanggal.append(result.find('div', {'class':'simple-item-view-date word-break item-page-field-wrapper table'}).get_text().replace("Da
      except:
        tanggal.append('')
```

Gambar 4. 7 Kode program proses web scraping

Berdasarkan pada gambar 4.7 proses web scraping dilakukan dengan rencana awal mengambil sebanyak 2000 data skripsi mahasiswa dari website repository UNEJ, hal ini dapat dilihat pada range halaman ketika proses *looping*. Selanjutnya, dengan memanfaatkan fungsi *try & except* dan pengguan tag – tag html yang susai pada letak data, peneliti berhasil mendapatkan data – data skripsi mahasiswa Universitas Jember.. Pada gambar 4.7 pula, peneliti banyak menggunakan fungsi *get_text* dan *replace*. Penggunaan fungsi *get_text* berguna untuk mendapantkan hanya bagian teks nya saja yang diambil pada proses scraping, dan Penggunaan *replace* berfungsi untuk menghilangkan kata/karakter/symbol yang tidak diinginkan.

	judul	nama	abstrak	fakultas	tanggal upload
0	Strategi Pemasaran Marketing Mix 7P pada Puasi...	fadel mohamad	Meningkatnya berbagai macam usaha bisnis memb...	UT-Faculty of Social and Political Sciences...	2021-06-02
1	Pengaruh Pemberian Pupuk Organik Cair (POC) Ur...	putranto moch. afif dwi	Konsumsi sawi pakcoy di Indonesia pada tahun ...	UT-Faculty of Teacher Training and Educati...	2022-07-12
2	Pengaruh Brand Ambassador, E-promotion, Kualit...	rachman fernanda meidiwanto	Tujuan penelitian ini untuk mengtahui pengaru...	UT-Faculty of Economic and Business [11123]	2022-07-12
3	Stilistika dalam Novel Dua Barista Karya Najha...	layli iva anishatus zihrol	Stilistika dalam novel Dua Barista dikaji kar...	UT-Faculty of Teacher Training and Educati...	2022-07-22
4	Pengaruh Pengungkapan Sustainability Report te...	dewi viska kartika	Penelitian ini bertujuan untuk menguji pengar...	UT-Faculty of Economic and Business [11123]	2022-07-01
...	...	...	...	...	...
469	Pengaruh Implementasi Good Corporate Governanc...	fadilah fina riska	Krisis ekonomi di Indonesia menyebabkan perus...	UT-Faculty of Social and Political Sciences...	2022-06-24
470	Pengembangan E-LKPD Berbasis Kearifan Lokal Ke...	nurhaniah nuri	Materi di dalam buku siswa kelas IV SD pada t...	UT-Faculty of Teacher Training and Educati...	2022-07-25
471	The Effect of Mother's Mindset about Children ...	asrumi asrumi seliyari agustina dewi rasni hanny	The community's view of life and culture is d...	LSP-Jurnal Ilmiah Dosen [5285]	2022-09-26
472	The Medical Traditions of Indonesian- Osing-E...	asrumi asrum sariono agus sudarmaningtyas ana...	Tujuan penelitian ini mengungkap tipe-tipe le...	LSP-Jurnal Ilmiah Dosen [5285]	2022-09-18
473	Penyuluhan Peran Pola Pikir(Mindset) Orang Tua...	asrumi asrumi rasni hanny sundari asri	WHO menargetkan angka stunting tahun 2020 mak...	LSP-Jurnal Ilmiah Dosen [5285]	2022-09-01

474 rows × 5 columns

Gambar 4. 8 Data skripsi mahasiswa yang berhasil didapatkan

Pada awalnya, peneliti ingin mendapantkan data sebesar 2000 data tetapi berdasarkan hasil web scraping, peneliti hanya bisa mendaptkan data sebesar 474 data yang sesungguhnya berhasil didapatkan sesuai dengan pada gambar 4.8. Hal ini dikarenakan pada skripsi mahasiswa, terdapat skripsi yang aksesnya dibatasi untuk dilihat oleh public, sehingga pada penelitian ini data yang di peroleh hanyalah sebesar 474 data skripsi mahasiswa yang bersifat publik.

Ketika data sudah didapatkan, selanjutnya *export* data ke bentuk CSV agar lebih mudah diproses oleh python. Berikut ini adalah Tabel dari database skripsi mahasiswa universitas jember yang tercantum pada gambar 4.9.

	A	B	C	D	E
1	judul	nama	abstrak	fakultas	tanggal upload
2	Strategi Pemasaran M	fadel mohamad	Meningkatnya berbagai macam usaha bisnis	['UT-Faculty', 'of', 'Social', 'and'	2021-06-02
3	Pengaruh Pemberian	putranto moch. afif c	Konsumsi sawi pakcoy di Indonesia pada tah	['UT-Faculty', 'of', 'Teacher', 'Tr	2022-07-12
4	Pengaruh Brand Amb	rachman fernanda m	Tujuan penelitian ini untuk mengtahui penga	['UT-Faculty', 'of', 'Economic', '	2022-07-12
5	Stilistika dalam Nove	layli iva anishatus zih	Stilistika dalam novel Dua Barista dikaji kare	['UT-Faculty', 'of', 'Teacher', 'Tr	2022-07-22
6	Pengaruh Pengungka	dewi viska kartika	Penelitian ini bertujuan untuk menguji penga	['UT-Faculty', 'of', 'Economic', '	2022-07-01
7	Penyusunan Laporan	abidin zainal	Usaha Mikro Kecil dan Menengah (UMKM) s	['UT-Faculty', 'of', 'Economic', '	2022-06-20
8	Tindak Pidana denga	bahari gabriel angelia	Tindak pidana perusakan hutan yang sudah r	['UT-Faculty', 'of', 'Law', '[5521	2021-05-20
9	Pengaruh Independer	amanda frisca ella	Penelitian ini bertujuan untuk menguji penga	['UT-Faculty', 'of', 'Economic', '	2022-06-06
10	Analisis Mutu Fisik Bu	karuniasari dian	Jambu biji merah (Psidium guajava L.) merup	['UT-Faculty', 'of', 'Agricultural'	2022-06-21
11	Pemahaman Akuntar	asyhari aulia rizka	UMKM belum dapat mengikuti prasyarat per	['UT-Faculty', 'of', 'Economic', '	2022-07-15
12	Gambaran Quality of	adisiwi yahtarita ulfia	During the COVID-19 pandemic, the elderly v	['UT-Faculty', 'of', 'Nursing', '[1	2021-09-02
13	Kehidupan Sosial Eko	rahayu yayuk	Penelitian ini membahas Kehidupan Sosial El	['UT-Faculty', 'of', 'Culture', '(Cu	2021-07-19
14	Hubungan Tingkat Str	pusparini yeni	Stress is a physical, psychological, emotional	['UT-Faculty', 'of', 'Nursing', '[1	2021-10-13
15	Kadar Malondialdehi	mahendra yohanes cl	Pemeriksaan radiologi di kedokteran gigi (KG	['UT-Faculty', 'of', 'Dentistry', '[	2021-08-02
16	Pengembangan Elekt	amini rosyida	Fosfor (P) merupakan salah satu unsur hara	['UT-Faculty', 'of', 'Mathematic	2021-07-15
17	Perkembangan Diplo	evita ypriliansi nora	Skripsi ini menelaah perkembangan diploma	['UT-Faculty', 'of', 'Social', 'and'	2021-07
18	Analisis Tegangan Da	nugroho yulianto set	Indonesia adalah negara yang terletak di dae	['UT-Faculty', 'of', 'Engineering'	2021-08-12
19	Analisis Tingkat Kepu	sudhasni yuliavira	Kepuasan pasien merupakan indikator penti	['UT-Faculty', 'of', 'Economic', '	2021-08-31
20	Rancang Bangun Kota	zuhriasa zahra	Sayuran adalah tanaman hortikultura yang d	['UT-Faculty', 'of', 'Agricultural'	2021-09-21
21	Identifikasi dan Uji Se	lutfadaturroifa alya v	Dibalik tingginya angka konsumsi dan produk	['UT-Faculty', 'of', 'Medical', '[1	2022-06-21
22	Perencanaan Perkuat	rahmawati safra dwi	Pihak BBPJN VIII PPK S 02 membangun 3 dinc	['UT-Faculty', 'of', 'Engineering'	2021-07-13
23	Isu Etis Keperawatan	wulandari fitriyah	World Health Organization (WHO) menetapk	['UT-Faculty', 'of', 'Nursing', '[1	2022-05-31
24	Pengaruh Budaya Org	putri elfira ramadhan	Penelitian ini bertujuan untuk mengetahui da	['UT-Faculty', 'of', 'Economic', '	2022-06-28
25	Pengaruh Model Prob	fambudi ridho	Belajar merupakan suatu proses perubahan	['UT-Faculty', 'of', 'Teacher', 'Tr	2022-07-17
26	Analisis Determinan F	laksono adam indra	Penelitian ini bertujuan untuk mengetahui pe	['UT-Faculty', 'of', 'Economic', '	2022-07-19
27	Pemodelan Kemiskin	dewi mita kornilia	Salah satu Provinsi penyumbang kemiskinan	['UT-Faculty', 'of', 'Mathematic	2022-05-25
28	Uji Antelmintik serta	yogisuari ghita	Infeksi cacing (helmintiasis) merupakan peny	['UT-Faculty', 'of', 'Pharmacy', '	2022-07-13

Gambar 4. 9 Tabel Database Skripsi Mahasiswa Universitas Jember

Dari gambar 4.9, database yang dimiliki memuat dari judul, abstrak, nama penulis, fakultas, dan tanggal upload. Tetapi, data yang dibutuhkan berupa judul dan abstrak saja. Oleh karena itu, dilakukan seleksi ulang dari databse yang sudah ada. Sehingga didapatkan hasil seleksi database seperti gambar 4.10 dibawah ini.

	judul	abstrak
1	judul	abstrak
2	Strategi Pemasaran Marketing Mix 7	Meningkatnya berbagai macam usaha bisnis membuat para pelaku usaha wajib m
3	Pengaruh Pemberian Pupuk Organik	Konsumsi sawi pakcoy di Indonesia pada tahun 2015 dan 2016 mengalami kenaik
4	Pengaruh Brand Ambassador, E-pror	Tujuan penelitian ini untuk mengtahui pengaruh brand ambassador, e-promotion,
5	Stilistika dalam Novel Dua Barista Ka	Stilistika dalam novel Dua Barista dikaji karena terdapat gaya bahasa pengarang y
6	Pengaruh Pengungkapan Sustainabili	Penelitian ini bertujuan untuk menguji pengaruh positif Sustainability report yang
7	Penyusunan Laporan Keuangan UMK	Usaha Mikro Kecil dan Menengah (UMKM) sangat berperan penting dalam pereko
8	Tindak Pidana dengan Sengaja Meng	Tindak pidana perusakan hutan yang sudah menjadi perhatian masyarakat karena
9	Pengaruh Independensi, Profesionali	Penelitian ini bertujuan untuk menguji pengaruh independensi, profesionalisme, da
10	Analisis Mutu Fisik Buah Jambu Biji M	Jambu biji merah (Psidium guajava L.) merupakan salah satu komoditas tanaman y
11	Pemahaman Akuntansi, Tingkat Penc	UMKM belum dapat mengikuti prasyarat perbankan dalam hal penyusunan lapora
12	Gambaran Quality of Life Pada Lansi	During the COVID-19 pandemic, the elderly were required to reduce direct intera
13	Kehidupan Sosial Ekonomi Masyarak	Penelitian ini membahas Kehidupan Sosial Ekonomi Masyarakat di Kecamatan Se
14	Hubungan Tingkat Stres Perawat Rua	Stress is a physical, psychological, emotional and mental discomfort that causes
15	Kadar Malondialdehid Mda Darah Ra	Pemeriksaan radiologi di kedokteran gigi (KG) merupakan salah satu pemeriksaan
16	Pengembangan Elektroda Selektif Io	Fosfor (P) merupakan salah satu unsur hara makro yang dibutuhkan tanaman dala
17	Perkembangan Diplomasi Budaya In	Skripsi ini menelaah perkembangan diplomasi budaya Indonesia melalui batik. Tuj
18	Analisis Tegangan Dan Regangan Car	Indonesia adalah negara yang terletak di daerah pertemuan tiga lempeng tektoni
19	Analisis Tingkat Kepuasan Pasien Per	Kepuasan pasien merupakan indikator penting terhadap kualitas pelayanan yang
20	Rancang Bangun Kotak Pendingin Sa	Sayuran adalah tanaman hortikultura yang dapat berguna sebagai sumber pendap
21	Identifikasi dan Uji Sensitivitas Salmo	Dibalik tingginya angka konsumsi dan produksi daging ayam broiler, terdapat pengg
22	Perencanaan Perkuatan Lereng Men	Pihak BBPJN VIII PPK S 02 membangun 3 dinding penahan tanah dengan tipe gravit
23	Isu Etis Keperawatan dalam Penanga	World Health Organization (WHO) menetapkan COVID-19 sebagai Kedaruratan Ke
24	Pengaruh Budaya Organisasi, Kepuas	Penelitian ini bertujuan untuk mengetahui dan menganalisis secara parsial pengar
25	Pengaruh Model Problem Based Lea	Belajar merupakan suatu proses perubahan aktivitas baik secara fisik ataupun psik
26	Analisis Determinan Persistensi Laba	Penelitian ini bertujuan untuk mengetahui pengaruh book-tax differences dan arus

Gambar 4. 10 Database skripsi mahasiswa universitas jember setelah melakukan seleksi ulang oleh peneliti

Untuk melakukan seleksi ulang pada dataframe, peneliti membuat kolom baru bernama *all* yang isinya merupakan gabungan antara kolom judul dan kolom abstrak yang sesuai dengan gambar 4.11.

```
1 df["all"] = df["judul"] + " " + df["abstrak"]
2 df.head()
```

Gambar 4. 11 Kode untuk melakukan penggabungan kolom judul dan kolom abstrak

Pada database skripsi mahasiswa Universitas Jember yang akan diteliti meliputi judul dan abstrak skripsi. pada database judul menggunakan tipe data string, dan pada data abstak skripsi menggunkana tipe data string. Tipe data string dipilih agar dapat memudahkan peneliti dalam melakukan *cleaning data* dan analisis.

4.3. Preprocessing

Preprocessing dimulai dari melakukan proses *Case Folding* dilanjutkan pada proses *tokenizing* selanjutnya porses *stopword removal* dan terakhir adalah *stemming*.

4.3.1. Case Folding

Pada proses Case folding. Peneliti menggunakan perintah *str.lower* dari python. Dengan perintah ini, maka secara otomatis teks yang ada didalam dokumen akan diubah menjadi ke bentuk huru kecil semua. Untuk menjalankan perintah str.lower pada penelitian ini, peneliti juga memanfaatkan dataframe. Berikut ini adalah langkah yang dilakukan peneliti untuk melakukan case folding yang tercantum pada gambar 4.11.

```
# this function is to create case folding to lowercase
df['all'] = df['all'].str.lower()
df.head()
```

Gambar 4. 12 Kode perintah untuk melakukan case folding

Berdasarkan gambar 4.12 peneliti membuat dataframe baru yang bernama *all,* didataframe ini berisi gabungan antara data judul dengan data abstrak skripsi mahasiswa. Hal ini dilakukan agar memperluas *bank of word* saat dilakukan pencarian. Selanjutnya, dengan menggunakan dataframe *all*, peneliti menggabungkan perintah *str.lower* untuk dilakukan *case folding* pada data yang ada di database *all.* Sehingga hasil yang didapatkan sesuai dengan pada gambar 4.12.

	judul	nama	abstrak	fakultas	tanggal upload	all
0	Strategi Pemasaran Marketing Mix 7P pada Puasi...	fadel mohamad	Meningkatnya berbagai macam usaha bisnis memb...	['UT-Faculty', 'of', 'Social', 'and', 'Politic...	2021-06-02	strategi pemasaran marketing mix 7p pada puasi...
1	Pengaruh Pemberian Pupuk Organik Cair (POC) Ur...	putranto moch. afif dwi	Konsumsi sawi pakcoy di Indonesia pada tahun ...	['UT-Faculty', 'of', 'Teacher', 'Training', 'a...	2022-07-12	pengaruh pemberian pupuk organik cair (poc) ur...
2	Pengaruh Brand Ambassador, E-promotion, Kualit...	rachman fernanda meidiwanto	Tujuan penelitian ini untuk mengtahui pengaru...	['UT-Faculty', 'of', 'Economic', 'and', 'Busin...	2022-07-12	pengaruh brand ambassador, e-promotion, kualit...
3	Stilistika dalam Novel Dua Barista Karya Najha...	layli iva anishatus zihrol	Stilistika dalam novel Dua Barista dikaji kar...	['UT-Faculty', 'of', 'Teacher', 'Training', 'a...	2022-07-22	stilistika dalam novel dua barista karya najha...
4	Pengaruh Pengungkapan Sustainability Report te...	dewi viska kartika	Penelitian ini bertujuan untuk menguji pengar...	['UT-Faculty', 'of', 'Economic', 'and', 'Busin...	2022-07-01	pengaruh pengungkapan sustainability report te...

Gambar 4. 13 Data hasil case colding

Pada gambar 4.13 terlihat bahwa data yang awalnya berada di dataframe judul memiliki bentuk yang masih beragam seperti terdapat data yang memiliki huruf besar dan kecil. Tetapi setelah dilakukan *case folding*, maka data diseragamkan kebentuk huruf kecil semua yang sesuai pada dataframe *all*. Untuk lebih jelas, peneliti mengambil beberapa sampel data yang tersaji pada tabel 4.1.

Tabel 4. 1 Tabel data dokumen judul skripsi sebelum dan sesudah case folding

No.	Data sebelum case folding	Data seteleh dilakukan proses case folding
1.	Strategi Pemasaran Marketing Mix 7P pada Puasinjember Katering di Kecamatan Sumbersari Kabupaten Jember	strategi pemasaran marketing mix 7p pada puasinjember katering di kecamatan sumbersari kabupaten jember
2.	Pengaruh Pemberian Pupuk Organik Cair (POC) Urine Kambing yang diperkaya PGPR (Plant Growth Promoting Rhizobacteria) dari Akar Tanaman Bambu terhadap Pertumbuhan Pakcoy (Brassica rapa L.) sebagai Materi Penyusunan Booklet	pengaruh pemberian pupuk organik cair (poc) urine kambing yang diperkaya pgpr (plant growth promoting rhizobacteria) dari akar tanaman bambu terhadap pertumbuhan pakcoy (brassica rapa l.) sebagai materi penyusunan booklet
3.	Pengaruh Brand Ambassador, E-promotion, Kualitas Produk dan Harga terhadap Keputusan Pembelian pada Konsumen Sepatu Ortuseight di Jember	pengaruh brand ambassador, e-promotion, kualitas produk dan harga terhadap keputusan pembelian pada konsumen sepatu ortuseight di jember
4.	Stilistika dalam Novel Dua Barista Karya Najhaty Sharma dan Pemanfaatannya sebagai Alternatif Materi Pembelajaran Sastra di SMA	stilistika dalam novel dua barista karya najhaty sharma dan pemanfaatannya sebagai alternatif materi pembelajaran sastra di sma
5.	Pengaruh Pengungkapan Sustainability Report terhadap Nilai Perusahaan dengan Profitabilitas Sebagai Pemoderasi (Studi Empiris Pada Perusahaan Sektor Keuangan yang terdaftar di Bursa Efek Indonesia Tahun 2017-2020)	pengaruh pengungkapan sustainability report terhadap nilai perusahaan dengan profitabilitas sebagai pemoderasi (studi empiris pada perusahaan sektor keuangan yang terdaftar di bursa efek indonesia tahun 2017-2020)

Berdasarakan tabel 4.1 di setiap dokumen pada kolom pertama terdapat penggunaan kata-kata yang menggunakan huruf kapital. Tetapi setelah dilakukan proses case folding dengan menggunakan perintah *str.lower*, kata–kata yang awalnya terdapat huruf capital semuanya berhasil diseragamkan menjadi menggunakan huruf kecil semua.

4.3.2. Tokenizing

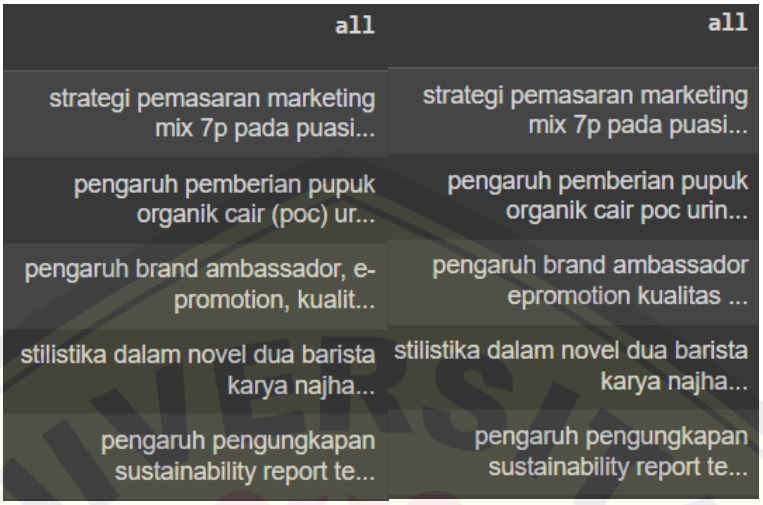
Sebelum masuk kedalam tahap tokenizing, terdapat tahap menghapus *single character*, tanda baca/*puctuation* dan *whitespace*. Hal ini dilakukan agar dokumen yang tersaji bersifat homogen berupa data text saja. Untuk lebih jelasnya, peneliti melampirkan bentuk program yang dibuat untuk melakukan proses penghapusan *single character*, tanda baca/*puctuation* dan *whitespace* yang tersaji pada gmbar 4.14.

```
#remove punctuation
def remove_punctuation(text):
    return text.translate(str.maketrans("","",string.punctuation))

df['all'] = df['all'].apply(remove_punctuation)

#remove whitespace
def remove_whitespace(text):
    return re.sub(' ','',text)

# remove single character
def remove_singl_char(text):
    return re.sub(r"\b[a-zA-Z]\b", "", str(text))

df['all'] = df['all'].apply(remove_singl_char)
df.head()
```

Gambar 4. 14 kode program proses penghapusan punctuation, whitespace, dan single charachter

Pada gambar 4.13 proses penghapusan punctuation, whitespace, dan single character adalah dengan membuat sebuah method baru. Pada proses penghilangan punctuation menggunakan method dengan nama fungsi *maketrans*. Fungsi ini akan karakter dengan karakter lain, pada penelitian ini karakter diubah adalah berada di fungsi *str.punctuation* dan akan dibuah ke bentuk kosong. Selanjutnya pada proses penghapusan *whitespace* menggunakan method dengan nama fungsi *re.sub.* Fungsi ini merupakan salah satu fungsi *regex* yang akan mengubah karakter yang ditentukan, pada penelitian ini karakter yang ditentukan adalah karakter *spasi* dan diubah kebentuk kosong. Selanjutnya untuk proses penghapusan single character adalah dengan method yang memiliki fungsi bernama *re.sub*. seperti halnya penghapusan *whitespace*, disini re.sub akan mengubah karakter diantara a-z dan A-Z yang berdiri sendiri ke bentuk kosong. Berikut ini adalah data hasil dari proses

penghapusan *punctuation*, *whitespace* dan *single character* yang tersaji pada gambar 4.15.



all	all
strategi pemasaran marketing mix 7p pada puasi...	strategi pemasaran marketing mix 7p pada puasi...
pengaruh pemberian pupuk organik cair (poc) ur...	pengaruh pemberian pupuk organik cair poc urin...
pengaruh brand ambassador, e-promotion, kualit...	pengaruh brand ambassador epromotion kualitas ...
stilistika dalam novel dua barista karya najha...	stilistika dalam novel dua barista karya najha...
pengaruh pengungkapan sustainability report te...	pengaruh pengungkapan sustainability report te...

Gambar 4. 15 hasil proses penghilangan punctuation, whitespace, dan single charachter

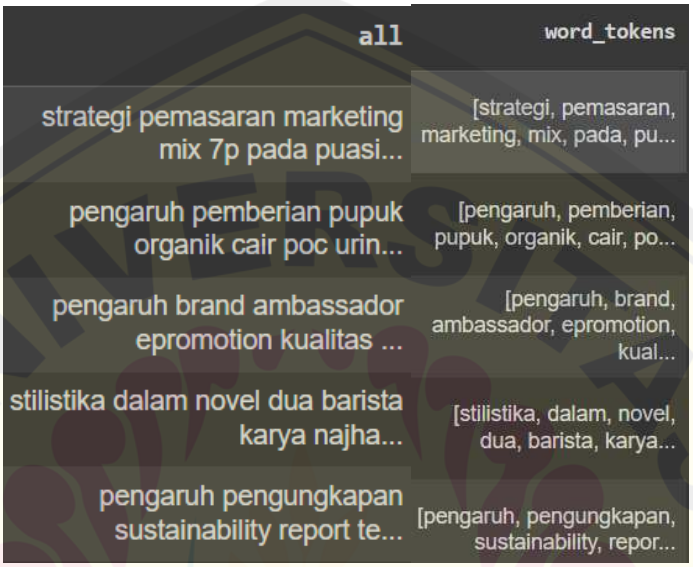
Pada gambar 4.15 adalah beberapa contoh hasil dari proses penghilangan punctuation, whitespace, dan single character. Pada kata “Pupuk organic (POC)” diubah menjadi “pupuk organic poc” tanpa adanya tanda kurung. Selanjutnya pada kata “Brand Ambassador, E-promotion, Kualit..” diubah kata menjadi “brand ambassador e-promotion kualit..” tanpa adanya tanda strip(-) dan kom(,).

Selanjutnya, barulah dapat dilakukan proses tokenisasi. Proses tokenisasi di penelitian ini menggunakan perintah *word_tokenize* dari *nltk*. Untuk implementasinya, peneliti membuat method baru yang berisi fungsi *word_tokenize*. Agar lebih jelas disini peneliti menampilkan bentuk penulisan kodenya yang tersaji pada gambar 4.16.

```
1 # this function returns a list of tokenized and stemmed words of any text
2 def tokenize(column):
3     tokens = nltk.word_tokenize(column)
4     return [w for w in tokens if w.isalpha()]
5
6 df['word_tokens'] = df.apply(lambda x: tokenize(x['all']), axis=1)
7 df.head()
```

Gambar 4. 16 Penulisan kode ptoses tokenizing

Pada gambar 4.16 dapat terlihat bahwa peneliti membuat fungsi baru yang bernama *tokenize.* fungsi ini yang akan menampung proses tokenisasi, yaitu dengan memanggil perintah *word_tokenize* selanjutnya, tinggal di return untuk dapat menjalankan perintah *word_tokenize* dengan membuat sebuah looping sederhana. Sehingga didaptkan hasil tokenisasi seperti pada gambar 4.17.



Gambar 4. 17 hasil tokenisasi

Pada gambar 4.17 adalah hasil dari proses tokenisasi. Berdasarakna gambar 4.16 dapat dilihat pada database judul yang awalnya memiliki satu kesatuan kalimat dipecah menjadi kata – kata. Agar lebih jelas, peneliti membuat beberapa contoh hasil tokenisasi yang tersaji pada tabel 4.2.

Tabel 4. 2 Tabel hasil tokenisasi

No.	Data hasil case folding	Data hasil tokenisasi
1.	strategi pemasaran marketing mix 7p pada puasinjember katering di kecamatan sumbersari kabupaten jember	'strategi', 'pemasaran', 'marketing', 'mix', 'pada', 'puasinjember', 'katering', 'di', 'kecamatan', 'sumbersari', 'kabupaten', 'jember'
2.	pengaruh pemberian pupuk organik cair (poc) urine kambing yang diperkaya pgpr (plant growth promoting rhizobacteria) dari akar tanaman bambu terhadap pertumbuhan pakcoy (brassica rapa l.) sebagai materi penyusunan booklet	'pengaruh', 'pemberian', 'pupuk', 'organik', 'cair', 'poc', 'urine', 'kambing', 'yang', 'diperkaya', 'pgpr', 'plant', 'growth', 'promoting', 'rhizobacteria', 'dari', 'akar', 'tanaman', 'bambu', 'terhadap', 'pertumbuhan', 'pakcoy', 'brassica', 'rapa', 'sebagai', 'materi', 'penyusunan', 'booklet'

3.	pengaruh brand ambassador, e-promotion, kualitas produk dan harga terhadap keputusan pembelian pada konsumen sepatu ortuseight di jember	'pengaruh', 'brand', 'ambassador', 'epromotion', 'kualitas', 'produk', 'dan', 'harga', 'terhadap', 'keputusan', 'pembelian', 'pada', 'konsumen', 'sepatu', 'ortuseight', 'di', 'jember
4.	stilistika dalam novel dua barista karya najhaty sharma dan pemanfaatannya sebagai alternatif materi pembelajaran sastra di sma	'stilistika', 'dalam', 'novel', 'dua', 'barista', 'karya', 'najhaty', 'sharma', 'dan', 'pemanfaatannya', 'sebagai', 'alternatif', 'materi', 'pembelajaran', 'sastra', 'di', 'sma',
5.	pengaruh pengungkapan sustainability report terhadap nilai perusahaan dengan profitabilitas sebagai pemoderasi (studi empiris pada perusahaan sektor keuangan yang terdaftar di bursa efek indonesia tahun 2017-2020)	'pengaruh', 'pengungkapan', 'sustainability', 'report', 'terhadap', 'nilai', 'perusahaan', 'dengan', 'profitabilitas', 'sebagai', 'pemoderasi', 'studi', 'empiris', 'pada', 'perusahaan', 'sektor', 'keuangan', 'yang', 'terdaftar', 'di', 'bursa', 'efek', 'indonesia', 'tahun'

Berdasarkan hasil penelitian pada tabel 4.2, didaptkan hasil tokenisasi berupa pemecahan masing – masing kata pada tiap kalimat dipotong-potong menjadi kata. Tetapi, ada beberapa kata yang tidak dilakukan proses tokenisasi, seperti data yang berupa angka dan *single charachter*. Pada data berupa angka seperti data angka berupa 7,1, 2017, dan 2020. Proses tokenisasi hanya terjadi pada data yang berupa teks saja sehingga angka tidak dapat dilakukan proses tokenisasi. Sedangkan pada single character seperti kata p dari kata 7p, tidak dilakukan proses tokenisasi karena sebelum memulai proses tokenisasi telah dilakukan penyesuaian data terlebih dahulu yaitu menghapus *single character*, *whitespace*, dan *punctuation*. Sehingga secara otomatis tanda baca juga akan terhapus dari proses tokenisasi. Setelah proses tokenisasi selesai selanjutnya masuk kedalam tahapan *stopword removal*.

4.3.3. Stopword Removal

Selanjutnya pada proses stopword, penelitian ini menggunakan perintah *stopwords* dari *nltk*, sedangkan bahasa yang digunakan yaitu bahasa Indonesia. Pemilihan bahasa Indonesia karena data yang tersedia menggunakan bahasa Indonesia. Berikut ini adalah proses penulisan kode untuk melakukan proses stopword removal yang tersaji pada gambar 4.18.

```
from nltk.corpus import stopwords
print(stopwords.words('indonesian'))
stop_words = set(stopwords.words('indonesian'))

def stopwords_removal(words):
    return [word for word in words if word not in stop_words]

df['word_stopword'] = df['word_tokens'].apply(stopwords_removal)
df.head()
```

Gambar 4. 18 Bentuk program proses stopword removal

Dari gambar 4.18 proses stopword removal dimulai dari memanggil fungsi *stopwords* dari nltk, kemudian set ke bahasa Indonesia. Selanjutnya membuat method baru untuk dilakukan proses stopword removal. Proses stopword removal ini menggunakan logika *looping* di setiap kata – kata nya. Selanjutnya peneliti memasukkan data hasil stemming tersebut ke dalam dataframe yang bernama *word_stopword*. Berikut ini adalah hasil stopword removal yang tersaji seperti pada gambar 4.19.

word_tokens	word_stopword
[strategi, pemasaran, marketing, mix, pada, pu...	[strategi, pemasaran, marketing, mix, puasinje...
[pengaruh, pemberian, pupuk, organik, cair, po...	[pengaruh, pemberian, pupuk, organik, cair, po...
[pengaruh, brand, ambassador, epromotion, kual...	[pengaruh, brand, ambassador, epromotion, kual...
[stilistika, dalam, novel, dua, barista, karya...	[stilistika, novel, barista, karya, najhaty, s...
[pengaruh, pengungkapan, sustainability, repor...	[pengaruh, pengungkapan, sustainability, repor...

Gambar 4. 19 hasil stop word removal

Berdasarkan gambar 4.19 terdapat beberapa kata yang hilang seperti kata 7p, pada, dalam, dua. Hal ini terjadi karena proses stopword removal menghapus kata kata yang tidak memiliki arti dan juga menghapis kata penghubung. Untuk lebih

jelasnya, peneliti mengambil beberapa contoh hasil stopword removal yang dibuat pada tabel 4.3.

Tabel 4. 3 Tabel hasil stopword removal

No.	Data hasil tokenisasi	Data hasil stopword removal
1.	'strategi', 'pemasaran', 'marketing', 'mix', 'pada', 'puasinjember', 'katering', 'di', 'kecamatan', 'sumbersari', 'kabupaten', 'jember'	'strategi', 'pemasaran', 'marketing', 'mix', 'puasinjember', 'katering', 'kecamatan', 'sumbersari', 'kabupaten', 'jember'
2.	'pengaruh', 'pemberian', 'pupuk', 'organik', 'cair', 'poc', 'urine', 'kambing', 'yang', 'diperkaya', 'pgpr', 'plant', 'growth', 'promoting', 'rhizobacteria', 'dari', 'akar', 'tanaman', 'bambu', 'terhadap', 'pertumbuhan', 'pakcoy', 'brassica', 'rapa', 'sebagai', 'materi', 'penyusunan', 'booklet'	'pengaruh', 'pemberian', 'pupuk', 'organik', 'cair', 'poc', 'urine', 'kambing', 'diperkaya', 'pgpr', 'plant', 'growth', 'promoting', 'rhizobacteria', 'akar', 'tanaman', 'bambu', 'pertumbuhan', 'pakcoy', 'brassica', 'rapa', 'materi', 'penyusunan', 'booklet'
3.	'pengaruh', 'brand', 'ambassador', 'epromotion', 'kualitas', 'produk', 'dan', 'harga', 'terhadap', 'keputusan', 'pembelian', 'pada', 'konsumen', 'sepatu', 'ortuseight', 'di', 'jember	'pengaruh', 'brand', 'ambassador', 'epromotion', 'kualitas', 'produk', 'harga', 'keputusan', 'pembelian', 'konsumen', 'sepatu', 'ortuseight', 'jember'
4.	'stilistika', 'dalam', 'novel', 'dua', 'barista', 'karya', 'najhaty', 'sharma', 'dan', 'pemanfaatannya', 'sebagai', 'alternatif', 'materi', 'pembelajaran', 'sastra', 'di', 'sma',	'stilistika', 'novel', 'barista', 'karya', 'najhaty', 'sharma', 'pemanfaatannya', 'alternatif', 'materi', 'pembelajaran', 'sastra', 'sma'
5.	'pengaruh', 'pengungkapan', 'sustainability', 'report', 'terhadap', 'nilai', 'perusahaan', 'dengan', 'profitabilitas', 'sebagai', 'pemoderasi', 'studi', 'empiris', 'pada', 'perusahaan', 'sektor', 'keuangan', 'yang', 'terdaftar', 'di', 'bursa', 'efek', 'indonesia', 'tahun'	'pengaruh', 'pengungkapan', 'sustainability', 'report', 'nilai', 'perusahaan', 'profitabilitas', 'pemoderasi', 'studi', 'empiris', 'perusahaan', 'sektor', 'keuangan', 'terdaftar', 'bursa', 'efek', 'indonesia'

Berdasarkan hasil penelitian pada tabel 4.3 terdapat kata kata yang ketika sebelumnya ada pada hasil tokenisasi menjadi hilang setelah dilakukan proses stopword removal. Hal ini dikarenakan pada proses stopword removal terjadi penghapusan kata penghubung dan kata - kata yang dianggap tidak penting atau tidak memiliki arti. Pada tabel 4.3. kata penghubung yang hilang diantaranya : kata dan, pada, di, yang, dari, terhadap, sebagai, dengan, dalam. Sedangkan kata – kata yang tidak memiliki arti yang hilang di antaranya : dalam, dua, dan tahun. Setelah proses stopword removal selesai, selanjutnya adalah ke tahapan stemming.

4.3.4. Stemming

Pada proses Stemming, peneliti menggunakan perintah *stemmerfactory* dari *sastrawi.* Pemilihan library sastrawi dikarenakan salah satu fungsi library *sastrawi* adalah untuk melakukan proses stemming menggunakan bahasa Indonesia. Berikut ini peneliti akan melampirkan proses stemming menggunakan python dimulai dari gambar 4.19.

```
import swifter
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
factory = StemmerFactory()
stemmer = factory.create_stemmer()
```

Gambar 4. 20 import kebutuhan untuk stemming

Pada gambar 4.20 peneliti membuat object bernama factory dari class StemmerFactory(). Object ini selanjutnya berguna untuk membuat object stemmer dengan menggunakan perintah create_stemmer(). Setelah persiapan selesai, selanjutnya adalah tahapan proses melakukan stemming yang termuat pada gambar 4.21.

```
def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}

for document in df['word_stopword']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ' '

print(len(term_dict))
print("------------------------")

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print(term,":" ,term_dict[term])

print(term_dict)
print("------------------------")

# apply stemmed term to dataframe
def get_stemmed_term(document):
    return [term_dict[term] for term in document]

df['stemmed_word'] = df['word_stopword'].swifter.apply(get_stemmed_term)
```

Gambar 4. 21 proses stemming

Setelah membuat persiapan kebutuhan proses stemming, selanjutnya dimulai proses stemmingnya dimulai dari membuat method baru dimana didalamnya akan

terjadi pemanggilan fungsi *stemmer.stem()* yang akan berfungsi melakukan proses stemming setiap termnya. Selanjutnya membuat list baru yang bernama *term_dict* yang berfungsi untuk menampung hasil konversi tiap kata menjadi term. Selanjutnya adalah proses *looping* untuk mengubah tiap kata menjadi term. Setelah mengubah kata menjadi term, selanjutnya adalah melakukan proses looping untuk memproses setiap term kedalam proses stemming dengan memanggil method yang sudah dibuat diawal yaitu method *stemmed_wrapper*. Setelah proses stemming selesai, selanjutnya data hasil stemming disimpan ke dalam dataframe dengan cara membuat method yang berisikan hasil stemming setiap term. Lalu, method tersebut di panggil kembali pada saat pembuatan dataframe baru, pada penelitian ini hasil proses stemming dimasukkan kedalam dataframe *stemmed_word*. Peneliti juga menggunakan perintah *swifter* untuk mempercepat proses pemasukan data stemming pada dataframe. Hasil proses stemming dapat dilihat pada gambar 4.22.

word_stopword	stemmed_word
[strategi, pemasaran, marketing, mix, puasinje...	[strategi, pasar, marketing, mix, puasinjember...
[pengaruh, pemberian, pupuk, organik, cair, po...	[pengaruh, beri, pupuk, organik, cair, poc, ur...
[pengaruh, brand, ambassador, epromotion, kual...	[pengaruh, brand, ambassador, epromotion, kual...
[stilistika, novel, barista, karya, najhaty, s...	[stilistika, novel, barista, karya, najhaty, s...
[pengaruh, pengungkapan, sustainability, repor...	[pengaruh, ungkap, sustainability, report, nil...

Gambar 4. 22 hasil stemming

Dari hasil di gambar 4.22 terdapat perbuahan kata – kata diantara pada kata pemasaran berubah menjadi pasar, kemudian pada kata pemberian berubah menjadi

beri, pada kata pengungkapan berubah menjadi ungkap. Agar lebih jelas peneliti mengambil beberapa contoh data hasil stemming yang tersaji pada tabel 4.4.

Tabel 4. 4 Tabel Haisl proses stemming

No.	Data hasil stopword removal	Data hasil stemming
1.	'strategi', 'pemasaran', 'marketing', 'mix', 'puasinjember', 'katering', 'kecamatan', 'sumbersari', 'kabupaten', 'jember'	'strategi', 'pasar', 'marketing', 'mix', 'puasinjember', 'katering', 'camat', 'sumbersari', 'kabupaten', 'jember'
2.	'pengaruh', 'pemberian', 'pupuk', 'organik', 'cair', 'poc', 'urine', 'kambing', 'diperkaya', 'pgpr', 'plant', 'growth', 'promoting', 'rhizobacteria', 'akar', 'tanaman', 'bambu', 'pertumbuhan', 'pakcoy', 'brassica', 'rapa', 'materi', 'penyusunan', 'booklet'	'pengaruh', 'beri', 'pupuk', 'organik', 'cair', 'poc', 'urine', 'kambing', 'kaya', 'pgpr', 'plant', 'growth', 'promoting', 'rhizobacteria', 'akar', 'tanam', 'bambu', 'tumbuh', 'pakcoy', 'brassica', 'rapa', 'materi', 'susun', 'booklet'
3.	'pengaruh', 'brand', 'ambassador', 'epromotion', 'kualitas', 'produk', 'harga', 'keputusan', 'pembelian', 'konsumen', 'sepatu', 'ortuseight', 'jember'	'pengaruh', 'brand', 'ambassador', 'epromotion', 'kualitas', 'produk', 'harga', 'putus', 'beli', 'konsumen', 'sepatu', 'ortuseight', 'jember'
4.	'stilistika', 'novel', 'barista', 'karya', 'najhaty', 'sharma', 'pemanfaatannya', 'alternatif', 'materi', 'pembelajaran', 'sastra', 'sma'	'stilistika', 'novel', 'barista', 'karya', 'najhaty', 'sharma', 'manfaat', 'alternatif', 'materi', 'ajar', 'sastra', 'sma'
5.	'pengaruh', 'pengungkapan', 'sustainability', 'report', 'nilai', 'perusahaan', 'profitabilitas', 'pemoderasi', 'studi', 'empiris', 'perusahaan', 'sektor', 'keuangan', 'terdaftar', 'bursa', 'efek', 'indonesia'	'pengaruh', 'ungkap', 'sustainability', 'report', 'nilai', 'usaha', 'profitabilitas', 'pemoderasi', 'studi', 'empiris', 'usaha', 'sektor', 'uang', 'daftar', 'bursa', 'efek', 'indonesia'

Pada tabel 4.4. dapat dilihat bahwa kata kata yang sebelumnya memiliki kata awalan (*prefix*) dan akhiran (*suffix*) berubah menjadi kata – kata dasarnya saja. Hal ini dikarenakan pada proses stemming, akan mengubah kata –kata menjadi kebentuk dasarnya dengan menghilangkan *prefix* dan *suffix*. Contohnya adalah kata pemasaran berubah menjadi pasar, kata kecamatan berubah menjadi camat, kata pemberian berubah menjadi beri. Setelah proses stemming selesai selanjutnya adalah masuk kedalam tahapan pemodelan vector space model yang dimulai dari proses perhitungan TF-IDF.

4.4. Pemodelan Vector Space Model

Setelah melakukan proses *preprocessing,* data yang kotor sebelumnya akan berubah menjadi data yang sudah bersih. Data yang sudah bersih ini kemudian akan digunakan untuk melakukan proses pemodelan *vector space model*. Pemodelan

vector space model diawali dari tahap perhitungan tf-idf, perhitungan *magnitude, perhitungan dot product*, dan terakhir adalah perhitungan *cosine similarity*.

4.4.1. Perhitungan TF-IDF

Pada perhitunagan TF-IDF di penelitian ini, menggunakan perintah *tfidfveectorizer* dari *sklearn*. Library ini akan berfungsi untuk mengubah kata – kata pada tiap dokumen menjadi kebentuk vector yang nanti nya akan digunakan untuk perhitungan *cosine similarity*. Berikut ini adalah proses TF-IDF yang tersaji pada gambar 4.23.

```
1 vectorizer = TfidfVectorizer()
2 X = vectorizer.fit_transform(df['cleaned_data'])
3
4 df1 = pd.DataFrame(X.toarray(), columns=vectorizer.get_feature_names())
5 pd.set_option('display.max_rows', 10)
6 df1
```

Gambar 4. 23 proses perhitungan TF-IDF menggunakan tfidfvectorizer

Pada gambar 4.23 proses perhitungan TF-IDF dimulai dengan mebuat fungsi barnama *vectorizer* yang berisi perintah dari *tfidfvectorizer*. Selanjutnya, peneliti memanggil perintah *fit_transform* yang akan mengubah term menjadi bentuk vector. Kemudian nilai vector inilah yang menjadi nilai dari perhitungan TF-IDF. Berikut ini adalah gambar 4.23 yang merupakan hasil dari perhitungan TF-IDF yang diperoleh.

	aa	aas	aba	abad	abadi	abai	abate	abc	abdi	abdoer	...	ziziphus	zizipus	zona	zone	zones	zoom	zpt	zscore	zscoretbu	zuhdi
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
469	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.047574	0.0	0.0
470	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
471	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
472	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
473	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0

474 rows × 10047 columns

Gambar 4. 24 Hasil perhitungan TF-IDF

Pada gambar 4.24 sebagian besar data yang terlihat memiliki nilai 0.0, karena nilai dari TF-IDF nya berada diantara dokumen 5 – 459. Tetapi terdapat salah satu data yang memiliki nilai TF-IDF sebesar 0.047574 yang berada pada kata zscore dan dokumen ke 469 yang menandakan bahwa kata zscore memiliki nilai TF-IDF sebesar 0.047574 pada data ke 469. Agar lebih jelas, peneliti mengambil beberapa hasil TF-IDF yang diperoleh yang tersaji pada Tabel 4.5:

Tabel 4. 5 Tabel dokumen judul dan abstrak skripsi

	abad	abadi	agustus	ahli	adequacy	Konkret	konflik
1.	0.0	0.0	0.0	0.0	0.026995414	0.0	0.0
2.	0.0	0.0	0.0	0.0	0.0	0.0	0.129513557
3.	0.0	0.0	0.0	0.052876423	0.0	0.0	0.0
4.	0.05316563	0.0	0.0	0.0	0.0	0.0	0.0
5.	0.0	0.0	0.040982867	0.0	0.0	0.0	0.0
6.	0.0	0.0	0.0	0.0	0.0	0.034141041	0.0
7.	0.0	0.13889368	0.0	0.0	0.0	0.0	0.0
8.	0.0	0.0	0.0	0.0	0.0	0.075079703	0.0
9.	0.0	0.0	0.0	0.0	0.0	0.0	0.03510065
10.	0.0	0.0	0.0	0.0	0.0	0.0	0.039702104

Pada tabel 4.5 merupakan beberapa contoh hubungan antara nomor vector kata dengan dokumen yang ada. Pada kata abad, memiliki nilai vector 0.05316563 pada dokumen 4, sedangkan pada dokumen lain memiliki nilai 0.0 yang berarti untuk kata abad tidak tersedia pada dokumen selain dokumen 4. Tetapi terdapat kata yang memiliki nilai vector pada lebih dari satu dokumen, seperti kata konkret yang memiliki nilai vector 0.034141041 pada dokumen 6 dan nilai 0.075079703 pada dokumen 8, yang berarti bahwa kata penghargaan terdapat pada dokumen 6 dan dokumen 8, sedangkan pada dokumen selain itu tidak ada sehingga nilai vectornya

0 . begitu juga pada kata konflik yang memiliki tiga nilai vector pada tiga dokumen yang berbeda yaitu nilai 0.129513557 pada dokumen 2, nilai 0.03510065 pada dokumen 9, dan nilai 0.039702104 pada dokmen 10.

4.4.2. Pemrosesan Query

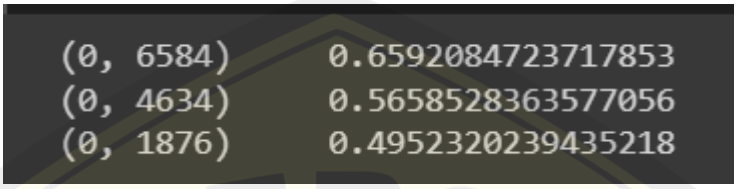
Selanjutnya pada perhitungan magnitude akan dilakukan perhitungan jarak pada query dengan cara menghitung nilai vectorynya. Pada gambar 4.25 merupakan proses perhitungan query.

```
def tokenize(column):
    tokens = nltk.word_tokenize(column)
    return [w for w in tokens if w.isalpha()]

def stopwords_removal(words):
    return [word for word in words if word not in stop_words]

def stemmed_wrapper(term):
    return stemmer.stem(term)


query = 'pemrograman komputer dan database'
query = query.lower()
query = tokenize(query)
query = stopwords_removal(query)
q = []
for w in query:
  stem = stemmed_wrapper(w)
  q.append(stem)
q = ' '.join(q)

query_vector = vectorizer.transform([q])
print(query_vector)
```

Gambar 4. 25 Proses perhitungan query

Langkah pertama yang dilakukan adalah memasukan querynya terlebih dahulu. Pada penelitian ini menggunakan query bernama **"pemrograman komputer dan database"**. Lalu, dengan query yang telah dimasukkan mulai lakukan proses *cleaning* kata dimulai dari tahap case folding, tokenizing, kemudian stopword removal, dan terakhir adalah stemming. Hal ini dilakukan agar pada kata di query

dan dokumen memiliki kesamaan susunan. Setelah itu baru menghitung nilai vectornya dengan menggunakan perintah *tfidfvectorizer* yang berada didalam fungsi vectorizer. Dengan perintah ini maka query akan ke bentuk nilai vector pada query. setelah dilakukan proses perhitungan qury, didapatkan hasil berupa nilai vector pada query yang tercantum pada gambar 4. 26.



Gambar 4. 26 hasil perhitungan query

Pada gambar 4.26 merupakan nilai dari masing – masing vector query. Untuk lebih jelas peneliti membuat sebuah tabel yang berupa hasil perhitungan vector query. Tabel ini tersaji pada tabel 4.6.

Tabel 4.6 Hasil perhitungan query

query	Nilai vector
program	0.6592084723717853
komputer	0.5658528363577056
database	0.4952320239435218

Dari tabel 4.6 dapat disimpulkan bahwa nilai vector query dari **"pemrograman komputer dan database**" adalah 0.6592084723717853 untuk kata program, 0.5658528363577056 untuk kata komputer, dan 0.4952320239435218 pada kata database.

4.4.3. Perhitungan cosine similarity

Pad perhitungan *cosine similarity*, terdapat juga sekaligus proses perhitungan *dot product*, karena proses perhitungan *dot product* menjadi satu bagian pada rumus perhitungan cosine similarity. Untuk menghitung nilai *cosine similarity* nya peneliti menggunakan perintah *cosine similarity* dari *sklearn*. Proses perhitungan cosine similarity tersaji pada gambar 4.27.

```
#  calculate cosine similarities
from sklearn.metrics.pairwise import cosine_similarity
cosineSimilarities = cosine_similarity(X,query_vector).flatten()
```

```
related_docs_indices = cosineSimilarities.argsort()[:-556:-1]

prediksi = []
prediksi_fak = []

for i in related_docs_indices:
    data = df['judul'][i]
    data2 = df['fakultas'][i]
    prediksi.append(data)
    prediksi_fak.append(data2)
    print(data)
    print(data2)
```

Gambar 4. 27 proses perhitungan cosine similarity

Berdasarkan gambar 4.27 Langkah yang dilakukan dimulai dari memanggil perintah *cosine similarity* dari *sklearn* kemudian membuat fungsi untuk melakukan proses perhitungan *cosine similarity* dengan melibatkan antara nilai vector dari dokumen dan nilai vector pada query. Setelah itu akan didapatkan hasil berupa urutan dokumen yang paling sesuai pada query yang dihitung dari derajat kemiripan antara vector dokumen dengan vector query. Fungsi *argsort* adalah berfungsi untuk mengurutkan dokumen berdasarkan indeks kemiripannya. Pembuatan list prediksi dan prediksi_fak digunakan untuk menampung hasil prediksi yang diberikan, data – data ini nanti akan dibuat database baru berupa database prediksi, untuk lebih jelasnya peneliti mencantum kan gambar 4.28 yang berisi kode pembuatan database hasil prediksi.

```
df2 = pd.DataFrame({'judul': prediksi,'fakultas': prediksi_fak})
df2
```

Gambar 4. 28 Proses pembuatan databse hasil predski

Dari gambar 4.28 peneliti membuat kolom judul dan fakultas untuk menampilkan hasil prediksinya, kolom judul berfungsi untuk menampilkan nama judul skripsinya dan kolom fakultas menunjukan kategori fakultas dari skripsi yang terkait. Kolom fakultas nanti juga akan dijadikan acuan untuk perhitungan confusion matrix.

Berikut ini adalah hasil dokumen dari perhitungan vector space dokumen yang tersaji pada gambar 4.29.

	judul	fakultas
0	Implementasi Metode Steganografi Reduced Diffe...	UT-Faculty of Computer Science [606]
1	Pengembangan Chatbot Menggunakan Metode Cosine...	UT-Faculty of Computer Science [606]
2	Efektivitas Terapi Latihan Asertif dalam Menur...	Diploma Programme - Nursing [189]
3	Perlindungan Hukum terhadap Pencipta Aplikasi ...	UT-Faculty of Law [5524]
4	Klasifikasi Kanker Payudara Berdasar Citra Mam...	UT-Faculty of Computer Science [606]
...	...	...
550	Pengaruh Interval Pemberian Nutrisi Pada Siste...	UT-Faculty of Agriculture [3313]
551	College Readers Of English Demotivating Factors	UT-Faculty of Culture (Cultural Knowledge) ...
552	Analisis Perilaku Harga, Integrasi Pasar, dan ...	UT-Faculty of Agriculture [3313]
553	Aksentuasi Produksi Bersih Pada Agroindustri K...	LSP-Jurnal Ilmiah Dosen [5310]
554	Strategi Pemasaran Marketing Mix 7P pada Puasi...	UT-Faculty of Social and Political Sciences...

555 rows × 2 columns

Gambar 4. 29 hasil perurutan dokumen menggunakan vector space model

4.5. K-Modes Clusetering

Untuk mengimplementasikan clustering K-Modes, peneliti menggunakan library *kmodes*. Langkah dimulai dengan memanggil library *kmodes,* lalu panggil database yang akan dilakukan K-Modes Clustering, pada penelitian ini data yang digunakan adalah data hasil prediksi dari *vector space model* dan data aslinya yang bertujuan untuk mengetahui nilai clusteringnya. Berikut ini adalah bentuk proses inisiasi *k-modes clustering* pada gambar 4.30.

```
1 pip install kmodes
```

```
1 from kmodes.kmodes import KModes
```

```
1 df3 = pd.read_csv('/content/drive/MyDrive/DATASET/Predicted_Data (2).csv',encoding='utf8')
2 df3
```

Gambar 4. 30 proses inisiasi k-modes clustering

Setelah mengimport kebutuhan yang diperlukan, selanjutnya adalah membuat fungsi baru yang berisi fungsi kmodes, dan didalamnya terdapat berisi jumlah cluster yang diinginkan dan maksimal iterasi yang dikehendaki, pada penelitian ini peneliti menggunakan nialai clustering sebanyak 2, dan maksimal iterasi nya adalah

555 data (sesuai dengan jumlah data yang dimiliki). Berikut ini adalah proses pembentukan kodenya dan hasilnya yang berada pada gambar 4.31.

```
km=KModes(n_clusters=2,max_iter=555)
clusters = km.fit_predict(df3)
clusters
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

Gambar 4. 31 Proses melakukan clustering pada database

Setelah itu, buatlah *dictionary* dan proses *looping* untuk memasukan nilai dari hasil clsustering tersebut kedalam dictionary. Berikut ini adalah prosesnya yang terdapat pada gambar 4.32.

```
cluster_dict=[]
for c in clusters:
  cluster_dict.append(c)
```

Gambar 4. 32 Proses pembuatan dictionary untuk memasukan nilai cluster

Langkah terakhir adalah masukan dictionary yang berisi nilai cluster tersebut kedalam database dengan cara membuat fungsi dari dataframe yang dituju dan panggil nama *dictionary* nya. Berikut ini adalah proses dan hasil database setelah memiliki nilai cluster yang terdapat pada gambar 4.33.

```
df3['cluster']=cluster_dict
df3
```

	judul	fakultas	cluster
0	Implementasi Metode Steganografi Reduced Diffe...	UT-Faculty of Computer Science [606]	0
1	Pengembangan Chatbot Menggunakan Metode Cosine...	UT-Faculty of Computer Science [606]	0
2	Efektivitas Terapi Latihan Asertif dalam Menur...	Diploma Programme - Nursing [189]	0
3	Perlindungan Hukum terhadap Pencipta Aplikasi ...	UT-Faculty of Law [5524]	0
4	Klasifikasi Kanker Payudara Berdasar Citra Mam...	UT-Faculty of Computer Science [606]	0

Gambar 4. 33 Proses pemasukan dictionary pada database

Proses clustering ini juga dilakukan pada database aslinya. Pada gambar 4.33 merupakan nilai cluster dari database hasil prediksi, sementara untuk database aslinya adalah terletak pada gambar 4.34.

```
df4['cluster']=cluster_dict2
df4
```

	judul	nama	abstrak	fakultas	tanggal upload	cluster
0	Strategi Pemasaran Marketing Mix 7P pada Puasi...	fadel mohamad	Meningkatnya berbagai macam usaha bisnis memb...	UT-Faculty of Social and Political Sciences...	2021-06-02	0
1	Pengaruh Pemberian Pupuk Organik Cair (POC) Ur...	putranto moch. afif dwi	Konsumsi sawi pakcoy di Indonesia pada tahun ...	UT-Faculty of Teacher Training and Educati...	2022-07-12	0
2	Pengaruh Brand Ambassador, E-promotion, Kualit...	rachman fernanda meidiwanto	Tujuan penelitian ini untuk mengtahui pengaru...	UT-Faculty of Economic and Business [11141]	2022-07-12	0
3	Stilistika dalam Novel Dua Barista Karya Najha...	layli iva anishatus zihrol	Stilistika dalam novel Dua Barista dikaji kar...	UT-Faculty of Teacher Training and Educati...	2022-07-22	0
4	Pengaruh Pengungkapan Sustainability Report te...	dewi viska kartika	Penelitian ini bertujuan untuk menguji pengar...	UT-Faculty of Economic and Business [11141]	2022-07-01	0

Gambar 4. 34 Proses dan hasil clsterng pada databse aslinya

4.6. Evaluasi hasil pencarian

Setelah melakukan clustering pada database hasil prediksi dan database aslinya, selanjutnya adalah proses untuk melakukan perhitungan c*onfusion matrix*. Berikut ini adalah gambar 4.35 yang merupakan proses visualisasi diagram *confusion matrix.*

```
# Source code credit for this function: https://gist.github.com/shaypal5/94c53d765083101efc0240d776a23823
def print_confusion_matrix(confusion_matrix, class_names, figsize = (10,7), fontsize=14):
    df_cm = pd.DataFrame(
        confusion_matrix, index=class_names, columns=class_names,
    )
    fig = plt.figure(figsize=figsize)
    try:
        heatmap = sns.heatmap(df_cm, annot=True, fmt="d")
    except ValueError:
        raise ValueError("Confusion matrix values must be integers.")
    plt.ylabel('Truth')
    plt.xlabel('Prediction')
```

Gambar 4. 35 proses pembuatan diagram confusion matrix

Pada gambar 4.35 proses dimulai dari membuat fungsi baru bernama *print_confusion_matrix* yang berisi parameter *confusion_matrix* untuk melakukan proses perhitungan confusion matrix, *class_names* untuk memberi nama pada masing masing kolom, *figsize* untuk mengatur ukuran diagram, dan *fontsize* untuk mengatur ukuran font. Selanjutnya membuat dataframe baru yang berisi perintah *confusion_matrix,* dan index. Dataframe ini berisi informasi terkait pembentukan

diagram *confusion matrix* dan juga fungsi untuk penamaan kolomnya. Setelah itu membuat ukuran dari diagramnya dengan menggunakan *pyplot* dari *matplotlib.* Kemudian membuat hasil isi diagram dengan *try* & *except.* Didalam perintah *try* berisikan perintah untuk mengambarkan pewarnaan pada setiap *cell* nya, jika semakin banyak/besar nilainya maka nilainya akan semakin berbeda. Sedangkan ketika di dalam perintah *except* berikan peringakatan jika terjadi error berupa "data harus berupa integer". Proses berikutnya adalah label kan kolm x dan y agar mudah diingat. Setelah membuat diagaram untuk visualisasi confusion matrix selanjutnya, tentukan nilai *truth* dan *prediction* nya. Berikut ini adalah gambar 4.36 sebagai proses penentuan nilai *truth* dan *prediction*,

```
truth = df4['true_doc']
prediction = df3['predicted_doc']
```

Gambar 4. 36 proses penentuan nilai truth dan prediction

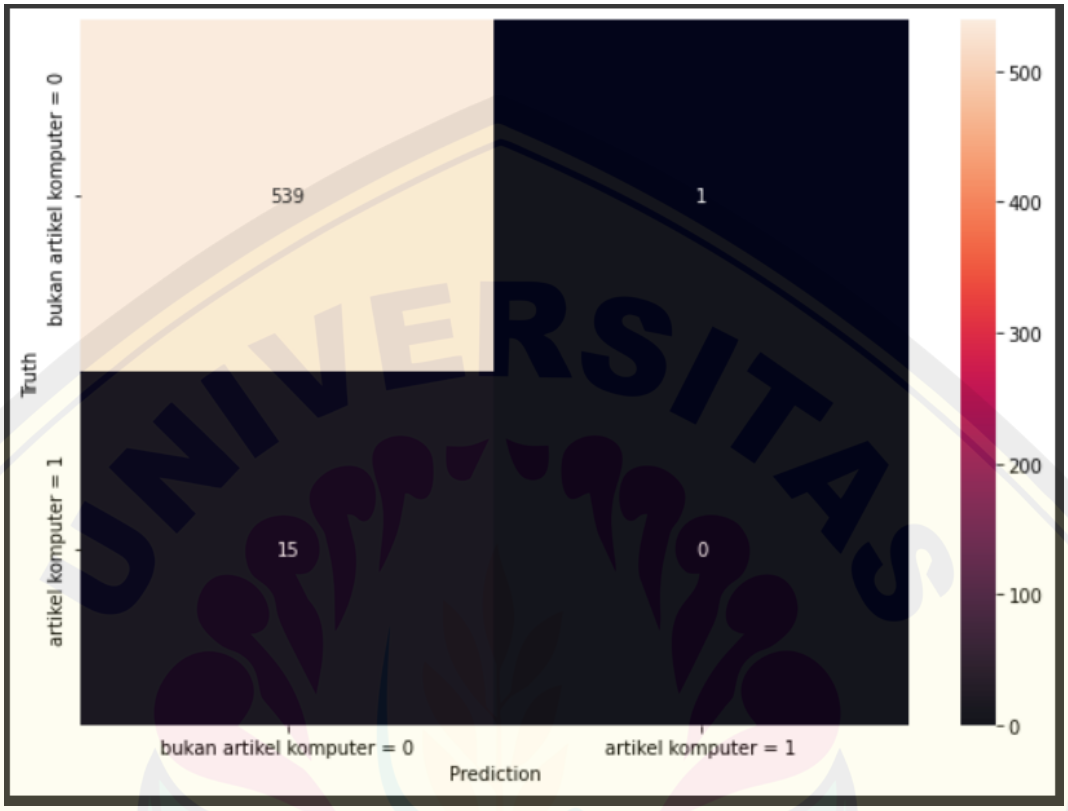
Pada gambar 4.38 nilai truth diperoleh dari database berisikan dokumen asli yang termuat pada df4 dan pada penggunaan kolom *true_doc* sebagai representasi dokumen yang sesuai dengan query yang telah kita inputkan. Pada nilai prediction menggunakan database hasil prediksi yang termuat pada df3 dengan memanggil kolom *predicted_doc* sebagai representasi dokumen yang suai dengan query yang telah diinputkan. Jika proses ini sudah dilakukan, selanjutnya masuk kedalam proses untuk memanggil perintah *truth* dan *prediction* yang telah ditentukan sebelumnya. Berikut ini adalah proses yang dilakukan sesuai pada gambar 4.37

```
cm = confusion_matrix(truth,prediction)
print_confusion_matrix(cm,["bukan artikel komputer = 0","artikel komputer = 1"])
```

Gambar 4. 37 proses pemanggilan nilai truth dan prediction dan menampilkan bentuk visualisasinya

Pada gambar 4.39 peneliti membuat fungsi yang bernama *cm* dimana didalamnya ada perintah *confusion_matrix* dan nilai *truth* dan *prediction.* Jika sudah dilakukan

proses tersebut selajutnya tinggal tampilkan bentuk diagramnaya saja dengan menggunakan perintah *print_confusion_matrix*. Maka akan didaptkan bentuk diagram *confusion matrix* seperti pada gambar 4.38.



Gambar 4. 38 Bentuk diagram confusion matrix

Jika sudah mengetahui nilai confusion matrixnya, selanjutnya dapat diketahui nilai dari precision, recall dan akurasinya. Berikut ini adalah gambar 4.39 yang merupakan proses perhitungan precision dan recall

```
1 print(classification_report(truth, prediction))
```

	precision	recall	f1-score	support
0	0.97	1.00	0.99	540
1	0.00	0.00	0.00	15
accuracy			0.97	555
macro avg	0.49	0.50	0.49	555
weighted avg	0.95	0.97	0.96	555

Gambar 4. 39 proses perhitungan precision dan recall

Dari gambar 4.39 untuk menghitung nilai precision recall dapat menggunakan perintah *classification report* dari *sklearn.* Pada gambar 4.39 dapat diketahui untuk nilai precision pada dokumen yang bukan merupakan artikel dokumen adalah sebesar 0.97 dan untuk dokumen yang merupakan artikel computer adalah 0.00. Sedangkan untuk recall pada dokumen bukan artikel komputer adalah sebesar 1.00 dan 0.00 untuk dokumen yang merupakan artikel komputer. Selain itu juga dapat diketahui akurasi dari hasil pencarian menggunakan vector space model adalah sebesar 0.97.

BAB 5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan analisis dan hasil penelitian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut :

1. Perancangan metode data skripsi mahasiswa Universitas Jember yang terdapat pada repository UNEJ dilakukan dengan menggunakan pendekatan web scraping yang menggunakan tag HTML dan atribut class sebagai acuan pola dalam membentuk pola ekspresi regulernya. Hal ini didasarkan pada hasil temuan dalam proses analisis struktur layout laman repositori UNEJ, yaitu struktur yang digunakan selalu tetap. Elemen judul diekstrak berdasarkan tag *h2* dengan nama class *page-header first-page-header*, elemen abstrak diekstrak berdasarkan tag *div* dengan nama class *simple-item-view-description item-page-field-wrapper table*, elemen nama penulis diekstrak berdasarkan tag *div* dengan nama class *simple-item-view-authors item-page-field-wrapper table*, elemen tanggal penerbit dikstrak berdasarkan tag *div* dengan nama class *simple-item-view-date word-break item-page-field-wrapper table*, dan elemen nama fakultas diekstrak berdasarkan tag *div* dengan nama class *simple-item-view-authors item-page-field-wrapper table*.
2. Hasil evaluasi pada sistem pencarian dokumen skripsi mahasiswa Universitas Jember yang telah dibangun dengan menggunakan tools classification report dari sklearn dengan metode precision & recall menunjukan tingkat akurasi yang tinggi, yaitu sebesar 0.97 atau sebesar 97%.

5.2. Saran

Dari hasil penelitian ini, peneliti memiliki saran yang dapat dijadikan masukan dalam melakukan peningkatan kualitas pencarian pada repository UNEJ.

1. Dalam melakukan pencarian dokumen, hendaknya sistem tidak hanya membaca kehadiran katanya saja tetapi juga menggabungkan antara dua jenis acuan data yaitu pada bagian judul dan juga abstrak. Sehingga, dapat menghasilkan

dokumen yang lebih relevan antara kata kunci yang diberikan dengan hasil yang didapatkan.

2. Penggunaan *vector space model* terbukti mampu memberikan dokumen yang relevan dengan kata kunci yang diberikan. Sehingga dapat dijadikan salah satu acuan dalam melakukan peningkatan kualitas pencarian dokumen pada repository UNEJ.

DAFTAR PUSTAKA

Arafah, M. (2018). Implementation of Generalized Vector Space Model Method At Automatic Assessment of Online Essay Exam. *Journal of Information Technology and Its Utilization*, *1*(2), 43. https://doi.org/10.30818/jitu.1.2.1893

Basmalah Wicaksono, V., Saptono, R., & Widya Sihwi, S. (2016). Analisis Perbandingan Metode Vector Space Model dan Weighted Tree Similarity dengan Cosine Similarity pada kasus Pencarian Informasi Pedoman Pengobatan Dasar di Puskesmas. *Jurnal Teknologi & Informasi ITSmart*, *4*(2), 73. https://doi.org/10.20961/its.v4i2.1768

Erin, L., Schapire, R. E., Banerjee, S., Immorlica, N., & Indyk, P. (2007). Preliminary draft (c) 2007 Cambridge UP Preliminary draft (c) 2007 Cambridge UP. *IEEE Transactions on Knowledge and Data Engineering*, *4209*(c), 304–315.

Flores, V. A., Permatasari, P. A., & Jasa, L. (2020). Penerapan Web Scraping Sebagai Media Pencarian dan Menyimpan Artikel Ilmiah Secara Otomatis Berdasarkan Keyword. *Majalah Ilmiah Teknologi Elektro*, *19*(2), 157. https://doi.org/10.24843/mite.2020.v19i02.p06

Hidayat, W. (2013). Indexing and Retrieval Engine untuk Dokumen Berbahasa Indonesia dengan Menggunakan Inverted Index. *Seminar Nasional Informatika Dan Aplikasinya (SNIA) 2015*, *October*.

https://www.researchgate.net/publication/284492748

Shultz, T. R., & Fahlman, S. E. (2017). Encyclopedia of Machine Learning and Data Mining. In *Encyclopedia of Machine Learning and Data Mining*. https://doi.org/10.1007/978-1-4899-7687-1

Uysal, A. K., & Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing and Management*, *50*(1), 104–112. https://doi.org/10.1016/j.ipm.2013.08.006