ACC
21 Mei 2021

SISTEM PEMBANGKIT KATA KUNCI DOKUMEN WEB MENGGUNAKAN METODE TEXTRANK (STUDI KASUS: IMAJI SOCIOPRENEUR)

SKRIPSI

Oleh:

Mokhamad Asif

NIM 172410102039

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER

UNIVERSITAS JEMBER

2021

Daftar Isi

Daftar Tabel

Daftar Gambar

A. Judul

Sistem Pembangkit Kata Kunci Dokumen Web Menggunakan Metode Textrank (Studi Kasus: Imaji Sociopreneur).

B. Latar Belakang

Web adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, video) yang di dalamnya mengunakan protokol *HTTP* (*hypertext transfer protokol*) dan untuk mengaksesnya dibutuhkan perangkat lunak berupa *browser* (Wibisono & Susanto, 2015). Dalam perkembangan *web* atau *website* penggunaan kategori membantu dalam melakukan klasifikasi halaman lebih lanjut. Hal ini memungkinkan pembuat situs *website* untuk menggunakan kata kunci secara tepat di setiap halaman dan subkategori yang membantu Google Spider mengindeks setiap halaman secara akurat dan dengan demikian membuat peringkat pencarian yang lebih baik (Fernandes & Vidyasagar, 2015). Google Spider sendiri merupakan program yang digunakan untuk menemukan dan memindai situs secara otomatis dengan mengikuti link dari satu halaman web ke halaman web lainnya (Google Developer, 2021). Pengklasifikasian halaman untuk membantu pengoptimalan dalam pencarian di suatu *website* inilah yang biasa disebut Search Engine Optimization atau SEO.

Imaji Sociopreneur adalah startup sosial yang berjalan di pemberdayaan masyarakat dan teknologi pertanian. Salah satu kendala yang dihadapi oleh Imaji Sociopreneur yaitu belum optimalnya pemodelan *SEO* dari *Website* Imaji Sociopreneur yang disebabkan karena tidak adanya *tag*, kategori, ataupun kata kunci terkait konten-konten. Pembuatan *tag*, kategori, ataupun kata kunci membutuhkan usaha lebih berupa penambahan fitur pada website yang membutuhkan riset dan juga usaha penulis untuk menyimpulkan kata-kata yang sering muncul pada sebuah paragraf yang akan ditulis pada Website Imaji Sociopreneur. Oleh sebagai itu, dibutuhkan pendekatan ektraksi kata kunci yang

dapat membantu menyimpulkan setiap tulisan yang ada di *Website* Imaji Sociopreneur.

Textrank adalah salah satu algoritma dalam *Natural Language Processing* (NLP) dengan pemodelan berbasis graf ranking model yang dikembangkan dari algoritma *Pagerank*. Algoritma *Texrank* digunakan untuk menentukan kepentingan suatu teks ataupun kalimat berdasarkan paragraf atau keseluruhan teks tertentu. *Textrank* mengadopsi pemodelan pemeringkatan kepentingan setiap *node* yang akan dibangun sebuah graf (sesuai dengan *Pagerank*) yang diimplementasi pada data tekstual (Wongchaisuwat, 2019). *Textrank* melakukan pemeringkatan dengan menghitung skor kepentingan setiap *node* atau *vertex* yang mewakili kata atau kalimat berdasarkan informasi dari keseluruhan data.

Berdasarkan penjelasan di atas, maka dalam penelitian ini peneliti mencoba mengimplementasikan Algoritma *Textrank* untuk melakukan ektraksi kata kunci pada *Website* Imaji Sociopreneur untuk menggantikan peran *tag*, kategori, ataupun kata kunci yang dibuat penulis untuk memaksimalkan *SEO* dari *Website* Imaji Sociopreneur.

C. Rumusan Masalah

Berdasarkan uraian yang telah disampaikan dalam latar belakang maka permasalahan yang harus diselesaikan dalam penelitian ini adalah bagaimana rancangan system ekstraksi tag, kategori, atau kata kunci otomatis untuk website Imaji Sociopreneur menggunakan algoritma Textrank.

D. Tujuan & Manfaat

D.1 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah mengembangkan sistem pada *Website* Imaji Sociopreneur utamanya penentuan *tag*, kategori, atau kata kunci dengan menggunakan algoritma *Textrank.*

D.2 Manfaat

Manfaat penelitian ini diharapkan dapat memberi manfaat sebagai berikut:

1. Bagi Peneliti

   Menerapkan ilmu pengetahuan yang telah didapat selama masa perkuliahan di Fakultas Ilmu Komputer untuk mengembangkan *Website* Imaji Sociopreneur menggunakan algoritma *Textrank*.

2. Bagi Objek Penelitian

   Hasil penelitian ini diharapkan dapat membantu Imaji Sociopreneur dalam peningkatan kualitas *Website* Imaji Sociopreneur pada bagian otomatisasi kata kunci.

3. Bagi Akademis

   Hasil penelitian ini diharapkan dapat menambah informasi yang berkaitan dengan judul penelitian bagi peneliti lain terutama pada Fakultas Ilmu Komputer Universitas Jember. Selain itu penelitian ini bertujuan untuk mendorong peneliti lain khususnya di Fakultas Ilmu Komputer Universitas Jember untuk mendorong pembaca dengan minat yang sama untuk mengambil topik penelitian yang serupa.

E. Batasan Masalah

Beberapa hal yang menjadi batasan masalah dalam penelitian ini sebagai berikut.

1. *Website* Imaji Sociopreneur yang dikembangkan adalah *Website* official dari Imaji Sociopreneur yang sudah digunakan hingga sekarang.
2. Pengembangan yang dilakukan pada penelitian ini hanya pengembangan otomatisasi ekstraksi kata kunci pada halaman-halaman *Website* Imaji Sociopreneur pada fitur blog, project dan event.
3. Responden dalam penelitian ini adalah jajaran dari Imaji Sociopreneur terutama pada yang menulis konten di *Website* Imaji Sociopreneur.

F. Tinjauan Pustaka

F.1 Penelitian Terdahulu

Penelitian yang dilakukan oleh Papis Wongchaisuwat pada tahun 2019 dengan judul "*Automatic Keyword Extraction Using TextRank*". Algoritma yang diusulkan bertujuan untuk memproses dan mengekstrak kata kunci secara otomatis dari data tekstual yang panjang. Algoritma ini didasarkan pada *Texrank* dengan penyesuaian lebih lanjut pada bobot *edge* dalam graf yang dibangun dari teks input. Secara khusus, setiap *node* mewakili unit teks dasar sementara setiap *edge* diberi bobot dengan ukuran kesamaan *Word2Vec* yang disesuaikan dengan skor pentingnya kalimat. Performa superior dari algoritma yang diusulkan mungkin dapat dicapai jika bobot *edge* lebih mewakili hubungan sebenarnya di antara unit teks yang berdekatan. Bobot *edge* didasarkan pada skor kepentingan kalimat dan kata yang berasal dari nilai hubungan 1 kata ataupun kalimat dengan kata ataupun kalimat lainnya. Karenanya, kesamaan kalimat akan berpotensi meningkatkan performa algoritma secara keseluruhan.

Penelitian yang dilakukan oleh Zhou Qingyun, Fang Yuansheng, Shang Zhenlei, dan Zhong Wanli pada tahun 2020 dengan judul "*Keyword Extraction Method for Complex Nodes Based on TextRank Algorithm*". Algoritma yang diusulkan adalah *TextRank* yang dikembangkan untuk ekstraksi pada tulisan huruf Cina. Metode ekstraksi kata kunci ini berdasarkan penghapusan *node* yang memiliki hubungan yang sangat kompleks, hal ini dilakukan untuk meningkatkan efektivitas algoritma ekstraksi kata kunci. Sementara itu, jumlah iterasi *TextRank* berkurang selama operasi berlangsung, dan waktu dibutuhkan untuk ekstraksi kata kunci teks juga dipersingkat. Dalam penelitian ini menunjukkan bahwa algoritma *Textrank* untuk ektraksi kata kunci dengan melakukan penghapusan pada *node* yang memiliki hubungan komplek dalam dokumen itu dapat mempengaruhi pentingnya kata dalam teks sampi batas tertentu. Kesimpulan dalam penelitian ini peneliti mengatakan metode penghapusan *node* dalam jaringan yang kompleks dapat mengekstrak kata kunci dengan lebih baik.

Penelitian yang dilakukan oleh Eris, Viny Christanti M dan Jeanny Pragantha pada tahun 2017 dengan judul "*Penerapan Algoritma Textrank Untuk Automatic Summarization Pada Dokumen Berbahasa Indonesia*"(Eris et al., 2017). Dalam penelitian ini algoritma *Textrank* digunakan untuk *automatic summarization* yaitu sistem yang digunakan untuk meringkat dokumen secara otomatis. *Textrank* diambil sebagai algorima untuk melakukan *automatic summarization* dikarenakan tidak diperlukannya pelatihan menggunakan data training. Perumusan kesimpulan diambil dari melakukan preprosessing, menghitung nilai kesamaan konten yang tumpang tindih, menghitung nilai *TextRank* pada setiap kalimat, dan membuat graf. Hasil dari penelitian ini menunjukan bahwa, algoritma ini mampu memberikan ringkasan dengan konten informatif hingga 82,48% untuk teks ringkasan 50% dan konten informatif 93,76% untuk teks ringkasan yang dirangkum 75%. Kesimpulan dalam penelitian ini peniliti mengatakan Algortima TextRank dapat mengambil kalimat menjadi hasil ringkasan jika kalimat tersebut mempunyai nilai content overlap similarity yang tinggi dibandingkan dengan kalimat-kalimat yang lainnya sehingga kalimat yang direpresentasikan sebagai vertex tersebut mempunyai banyak edge dan bernilai tinggi

F.2 *Website* Imaji Sociopreneur

Website Imaji Sociopreneur adalah *website* official Imaji Sociopreneur yang berfungsi sebagai media exposure yang dapat dicari di *search engine* seperti google. *Website* ini memuat beberapa fitur yang dapat menunjang *SEO* dari *Website* Imaji Sociopreneur yaitu blog, *event*, dan projek. Ketiga fitur ini berisi konten-konten mengenai Imaji Sociopreneur itu sendiri. Dalam perkembangannya *Website* Imaji Sociopreneur membutuhkan *SEO* untuk meningkatkan jumlah kunjungan pada *website* ini. Pengoptimalan jumlah kunjungan ini diharapkan dapat meningkatkan minat masyarakat untuk bersama mewujudkan visi dan misi Imaji Sociopreneur.

F.3 *Natural Language Processing*

Natural Language Processing (Pemrograman Bahasa Alami) adalah bidang penelitian dalam ilmu komputer dan kecerdasan buatan (AI) yang berkaitan dengan pemrosesan bahasa alami seperti bahasa Inggris atau Mandarin. Pemrosesan ini umumnya melibatkan penerjemahan bahasa alami menjadi data (angka) yang dapat digunakan komputer untuk mempelajari dunia. Dan pemahaman tentang dunia ini terkadang digunakan untuk menghasilkan teks bahasa alami yang mencerminkan pemahaman tersebut (Lane et al., 2019). Hal ini dapat dilakukan secara umum dengan mencari 5W+1H. *NLP* biasanya membuat penggunaan konsep-konsep linguistik seperti kata benda, kata kerja, kata sifat, dan lainnya dan struktur gramatikal (baik direpresentasikan sebagai ungkapan-ungkapan seperti frase nomina atau frase preposisional, atau hubungan ketergantungan seperti subjek dari- atau objek-dari) (Wangsanegara & Subaeki, 2015).

F.4 *Textrank*

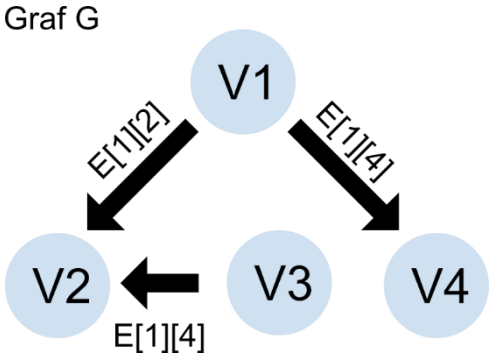
Texrank merupakan *graph-based ranking algorithm* (pemeringkatan dengan model graf), yaitu algoritma yang dapat menentukan kepentingan suatu teks ataupun kalimat berdasarkan paragraf atau keseluruhan teks tertentu. *Textrank* mengadopsi pemodelan pemeringkatan kepentingan setiap *node* yang akan dibangun sebuah graf sesuai dengan *Pagerank* yang diimplementasi pada data tekstual (Wongchaisuwat, 2019). Pemrosesan teks *Textrank* sangatlah fleksibel karena dapat digunakan pada berbagai bahasa tanpa mengubah algoritmanya. Hal ini dikarenakan *Textrank* tidak memerlukan *data training* untuk proses pengelolaan dokumen (Mihalcea & Tarau, 2004). Terdapat dua jenis pengelolahan bahasa dalam *TextRank*, yaitu *TextRank for keyword extraction* (ekstraksi kata kunci) dan *TextRank for sentence extraction* (ekstraksi kalimat) (Eris et al., 2017).

Kata kunci adalah kata atau frasa penting dalam judul, judul subjek (pendeskripsian), catatan konten, abstrak, atau teks catatan dalam katalog online

atau *database* bibliografi yang dapat digunakan sebagai istilah pencarian dalam pencarian teks bebas untuk mengambil semua catatan yang berisi itu (Reitz, 2020).

Kata kunci digunakan untuk memahami dengan cepat konten dokumen dan subjek pemahaman. Teknologi ekstraksi kata kunci adalah cara yang signifikan untuk mendapatkan makna inti dari informasi teks dengan cepat, dan dapat aplikasikan diberbagai bidang seperti untuk intelijen, jurnalisme, pencarian informasi, dan pemahaman bahasa alami (Qingyun et al., 2020).

Model *Textrank* menerapkan algoritma *Pagerank* yang terkenal dengan data Tekstual. Algoritma *Pagerank* berbasis graf digunakan untuk mengukur kepentingan relatif halaman situs *website* dalam kumpulan *hyperlink* (Wongchaisuwat, 2019).



Gambar 1. Ilustrasi Graf G

Dilambangkan sebuah graf sebagai G(V,E), V merupakan himpunan *vertex* graph G dan E merupakan himpunan *edge*, dimana E merupakan subset dari V*V. Untuk *vertex* V_i, In(V_i) merupakan himpunan *vertex* yang terhubung dan mengarah masuk ke dalam *vertex* V_i (predecessor), dan Out(V_i) merupakan himpunan *vertex* yang terhubung dan mengarah keluar *vertex* V_i (successor). Nilai V_i dinyatakan dalam persamaan

$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \qquad \text{(F.1)}$$

Dimana d merupakan nilai *damping factor* yang dapat diambil nilainya mulai dari 0 hingga 1 (Mihalcea & Tarau, 2004).

Koneksi antar *node* dalam graf dapat ditangani secara berbeda dengan memasukkan kekuatan koneksi ke dalam model. Secara khusus, bobot *edge* W_{ij}

yang sesuai dengan *node* V_i dan *node* V_j dipertimbangkan saat menghitung skor kepentingan. Rumus untuk skor tertimbang didefinisikan sebagai berikut

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ij}}{\sum_{V_k \in Out(V_j)} w_{jl}} S(V_j) \qquad \text{(F.2)}$$

Dalam graf yang akan dibangun, alur algoritma yang digunakan terdiri dari: A) Perhitungan skor kalimat; B) Perhitungan skor kata kunci. Fase pertama menerapkan algoritma *Textrank* pada level kalimat yang menghasilkan skor penting untuk setiap kalimat. Pada tahap kedua, variasi dari algoritma *Textrank* di tingkat kata diimplementasikan dan dikombinasikan dengan representasi kata yang disematkan. Skor kalimat dari frasa pertama diperhitungkan saat menghitung skor kata di frasa kedua. Akhirnya, kata-kata yang terkait dengan skor tertinggi diambil sebagai kata kunci yang diekstrak. Semua implementasi menggunakan python dengan implementasi Word2Vec dari genisme. Model Word2Vec terlatih yang digunakan dalam pekerjaan ini dilatih sebagai bagian dari kumpulan data Google Berita (Wongchaisuwat, 2019). Berikut alur *Textrank* dalam pendekatan Wongchaisuwat:

A. *Sentence Scores Computation*

Untuk menghitung skor kepentingan setiap kalimat, dokumen asli awalnya dipecah menjadi beberapa kalimat. Graf yang sepenuhnya terhubung dibangun dari *node* dan *edge* yang mewakili kalimat dan skor kesamaan antara 2 *node* yang berdekatan. Graf kalimat G_s = (V, E) adalah graf yang tidak berarah dengan sekumpulan kalimat V dan sisi-sisi E dengan V adalah *vertex* (*node*) dan E adalah *edge*. Setiap *edge* diberi bobot sesuai dengan skor kesamaan kalimat. Kesamaan skor antara 2 kalimat didasarkan pada kesamaan kata dan panjang kalimat seperti yang didefinisikan dalam. algoritma *Textrank,* kemudian diimplementasikan pada graf G_s. skor kepentingan untuk kalimat V_i didefinisikan sebagai WS(V_i) diambil dari algoritma.

B. *Keyword Scores Computation*

Pada tahap ini bobot pada algoritma *Textrank* diterapkan pada tingkatan kata untuk mengekstraksi kata atau frasa. Setelah *tokenizing*

dokumen asli, graf tidak berarah G_w akan dibangun ketika token (kata) dianggap sebagai *node*. Hubungan sesama antarkata ditambahkan ke *edge* yang menghubungkan antara *node* yang berdekatan. Ukuran kedekatan dari kata-kata yang terbentuk dipertimbangkan dalam hubungan ini. Secara khusus, *edge* antara 2 *node* apapun ditambahkan jika dan hanya jika jarak antara 2 kata yang sesuai kurang dari ukuran jendela yang ditentukan sebelumnya.

Bobot *edge* W_{ij} sebagian diperoleh dari kesamaan antara kata V_i dan V_j. Untuk meningkatkan kinerja algoritma *Textrank,* kemiripan semantik antara 2 kata dipertimbangkan. Untuk lebih spesifik, representasi vektor untuk setiap kata diambil dari model Word2Vec. Kesamaan antara vektor kata dihitung dan dimasukkan ke dalam rumus TextRank berbobot. *Out-Of-Vocabulary* yang dilambangkan sebagai kata-kata *OOV* adalah kata-kata yang tidak terlihat yang diamati hanya dalam set tes. Ini menyiratkan bahwa vektor kata untuk kata-kata *OOV* tidak dapat diambil dari model Word2Vec. Untuk menangani kata-kata *OOV,* nilai kesamaan yang telah ditentukan digunakan sebagai default.

Penelitian ini dibangun dengan asumsi bahwa kepentingan setiap kata diperoleh dari kata itu sendiri dan kalimat dari mana kata itu diambil. Skor kalimat juga berisi wawasan yang berguna untuk membantu meningkatkan kinerja algoritma. Menurut algoritma yang diusulkan, skor kalimat WS_s yang dihitung dari algoritma *Textrank* pada langkah sebelumnya dinormalisasi ke rentang 0 dan 1. Bobot *edge* yang dihitung dari model Word2Vec selanjutnya disesuaikan dengan skor kalimat ini. Pada dasarnya, vektor kata yang sesuai dengan 2 *node* yang berdekatan ini V_i danV_j diambil dari model Word2Vec. Kesamaan kosinus dihitung antara 2 vektor kata ini. Selain itu, sekumpulan skor kalimat yang sesuai dengan semua kalimat tempat V_i danV_j diambil, dikumpulkan. Skor kalimat rata-rata di seluruh set ini kemudian dihitung. Bobot *edge* akhir W_{ij} adalah perkalian dari skor kalimat rata-rata dan kesamaan kata. Terakhir, rumus

Textrank berbobot dengan bobot *edge* akhir diulangi hingga konvergensi. Skor akhir WS_w untuk setiap kata diambil.

Setelah mengurutkan skor kata terakhir dalam urutan terbalik (besar ke kecil), kata-kata yang sesuai dengan skor teratas dikumpulkan sebagai kata kunci potensial. Kata kunci potensial ini diproses pasca untuk mencari kata kunci multi-kata. Secara khusus, kata kunci potensial yang berdekatan yang terdapat dalam dokumen asli digabungkan menjadi kata kunci frase tunggal.

F.5 *Text Preprocessing*

Text Preprocessing adalah suatu tahapan mengubah teks asli sebagai masukan dan menerapkan beberapa rutinitas dasar untuk mengubah atau menghilangkan unsur tekstual yang tidak berguna dalam pengolahan lebih lanjut (Najjichah et al., 2019). Dalam penelitian ini digunakan beberapa metode *text preprocessing* sebelum dilakukan pembangunan graf *Textrank* yaitu diantaranya.

F.5.1 *Case Folding*

Case Folding merupakan proses pengubahan data menjadi format yang sesuai. Hal ini bertujuan mengurangi redudansi data yang akan digunakan dalam proses pengklasifikasian sehingga proses perhitungan menjadi optimal. Contohnya mengubah format data menjadi *lowercase* atau *uppercase* sesuai dengan kebutuhan yang dibutuhkan dalam proses pengklasifikasiannya (Muttaqin & Bachtiar, 2016). Berikut contoh hasil dari proses *case folding*:

Tabel 1. Hasil *Case Folding*

Sebelum *case folding*	**Hasil *case folding***
Melihat keadaan tersebut kami dari Imaji Sociopreneur Bersama Yayasan Mimpi Indonesia menggagas sebuah Gerakan yang kami beri nama Menanam Buku	melihat keadaan tersebut kami dari imaji sociopreneur bersama yayasan mimpi indonesia menggagas sebuah gerakan yang kami beri nama menanam buku

F.5.2 *Tokenizing*

Tokenizing merupakan tahapan penguraian string teks menjadi *term* atau kata. Tujuan dari *Tokenizing* yaitu memisahkan kata-kata dalam sebuah paragraf, kalimat atau halaman ke dalam kata tunggal (Najjichah et al., 2019). Berikut contoh hasil dari proses *tokenizing*:

Tabel 2. Hasil *Tokenizing*

Sebelum *tokenizing*	**Hasil *tokenizing***
melihat keadaan tersebut kami dari imaji sociopreneur bersama yayasan mimpi indonesia menggagas sebuah gerakan yang kami beri nama menanam buku	['melihat', 'keadaan', 'tersebut', 'kami', 'dari', 'imaji', 'sociopreneur', 'bersama', 'yayasan', 'mimpi', 'indonesia', 'menggagas', 'sebuah', 'gerakan', 'yang', 'kami', 'beri', 'nama', 'menanam', 'buku']

F.5.3 *Filtering*

Tahap *Filtering* adalah tahap mengambil kata-kata penting dari hasil *tokenizing*. Dalam *Filtering* dapat dilakukan dengan algoritma *Stopword removal*. *Stopword removal* merupakan penghapusan kata-kata yang tidak relevan dalam penentuan topik sebuah dokumen dan yang sering muncul pada dokumen, misalnya “dan”, “atau”, “sebuah”,“adalah”,pada dokumen berBahasa Indonesia (Najjichah et al., 2019). Berikut contoh hasil dari proses filtering:

Tabel 3. Hasil *Filtering*

Sebelum filtering	**Hasil filtering**
['lihat', **'ada'**, 'sebut', **'kami'**, **'dari'**, 'imaji', 'sociopreneur', **'sama'**, 'yayasan', 'mimpi', 'indonesia', 'gagas', 'buah', 'gera', 'yang', 'kami', 'beri', 'nama',	['lihat', 'sebut', 'imaji', 'sociopreneur', 'yayasan', 'mimpi', 'indonesia', 'gagas', 'buah', 'gera', 'beri', 'nama', 'tanam', 'buku']

'tanam', 'buku']	

F.5.4 *Stemming*

Stemming merupakan tahapan pengubahan suatu kata menjadi akar katanya dengan menghilangkan imbuhan awal atau akhir pada kata tersebut (Najjichah et al., 2019). Dalam *Stemming* Bahasa Indonesia dilakukan beberapa tahapan sebagai berikut

1. Pengecekan kata tersebut apakah merupakan kata dasar.
2. Menghilangkan inflection suffix lalu dikakukan proses nomer 1.
3. Menghilangkan derivational suffix lalu dilakukan proses nomer 1.
4. Menghilangkan derivational prefix lalu dilakukan proses nomer 1.
5. Bila keempat proses tidak menemukan kata dasarnya. Maka dilakukan analisis kata tersebut masuk dalam tabel diambiguitas kolom terakhir atau tidak.
6. Bila semua proses di atas gagal, maka algoritma mengembalikan kata aslinya.

Berikut merupakan contoh dari proses *stemming*:

Tabel 4. Hasil *Stemming*

Sebelum *stemming*	**Hasil *stemming***
['melihat', 'keadaan', 'tersebut', 'kami', 'dari', 'imaji', 'sociopreneur', 'bersama', 'yayasan', 'mimpi', 'indonesia', 'menggagas', 'sebuah', 'gerakan', 'yang', 'kami', 'beri', 'nama', 'menanam', 'buku']	['lihat', 'ada', 'sebut', 'kami', 'dari', 'imaji', 'sociopreneur', 'sama', 'yayasan', 'mimpi', 'indonesia', 'gagas', 'buah', 'gera', 'yang', 'kami', 'beri', 'nama', 'tanam', 'buku']

F.5.5 *Parts-of-Speech Tagging*

Part-of-speech (POS) tagging atau secara singkat dapat ditulis sebagai *tag*ging merupakan proses pemberian penanda *POS* atau kelas sintaktik pada tiap kata di dalam corpus. Dikarenakan *tag* secara umum juga diaplikasikan pada tanda baca, maka dalam proses *tag*ging, tanda baca seperti tanda titik, tanda koma, dll perlu dipisahkan dari kata-kata. Oleh sebab itu, proses tokenisasi biasanya dilakukan sebelum *POS-Tagging*. Selain itu beberapa *preprocessing* juga dilakukan seperti pemisahan koma, tanda petik, dll dari kata serta dilakukan juga disambiguitas pada tanda baca penanda akhir kalimat seperti tanda titik dan tanda tanya agar dapat dibedakan dari tanda yang digunakan untuk singkatan (seperti contohnya: e.g. dan etc.) (Suhartono, 2019). Berikut contoh hasil dari *POS-Tagging*:

Tabel 5. Hasil *POS-Tagging*

Kata	**Keterangan Label**
Saya	PRON
Dan	CCONJ
Dia	PRON
Kemarin	ADJ
Pergi	VERB
Ke	ADP
Pasar	NOUN
Bersama	ADP
Untuk	ADP
Membeli	VERB
Jeruk	NOUN

Dimana keterangan label sebagai berikut:

ADJ: kata sifat
ADP: preposisi
ADV: keterangan
AUX: kata bantu
CCONJ: kata penghubung
INTJ: kata seru
NOUN: kata benda
NUM : angka

PART : partikel	SYM : simbol
PRON : kata ganti	VERB : kata kerja
PUNCT : tanda baca	X : lainny

G. Metodologi Penelitian

G.1 Jenis Penelitian

Jenis penelitian ini menggunakan penelitian kuantitatif. Penelitian kuantitatif adalah metode yang digunakan untuk melakukan uji pada masalah penelitian yang berhubungan dengan data angka yang dapat di kalkulas. Wahidmurni (2017) mengatakan "Metode penelitian kuantitatif merupakan suatu cara yang digunakan untuk menjawab masalah penelitian yang berkaitan dengan data berupa angka dan program statistik. Untuk dapat menjabarkan dengan baik tentang pendekatan dan jenis penelitian, populasi dan sampel, instrumen penelitian, teknik pengumpulan data, dan analisis data dalam suatu proposal dan/atau laporan penelitian diperlukan pemahaman yang baik tentang masing-masing konsep tersebut". Dalam penilitian ini data yang akan dikalukulasi adalah data testing tingkat keakurasian dari metode yang akan didapat dari testing crawling data beberapa jurnal yang memiliki kata kunci dan beberapa wawancara kepada penulis di *Website* Imaji Sociopreneur.

G.2 Objek Penelitian

Objek penelitian merupakan *Website* Imaji Sociopreneur. Data yang didapat diperoleh dari penulis di *Website* Imaji Sociopreneur sebagai narasumber dan Jurnal Bahasa Indonesia yang tersebar di internet sebagai uji testing.

G.3 Tempat dan Waktu Penelitian

Tempat dilaksanakan penelitian yaitu kantor kantor Imaji Sociopreneur yang berlokasi Kelurahan/Desa Kecamatan Tegal Gede, Kecamatan Sumbersari

Kabupaten Jember. Waktu penelitian dilakukan selama empat bulan, dimulai dari bulan Maret sampai dengan bulan April 2021.

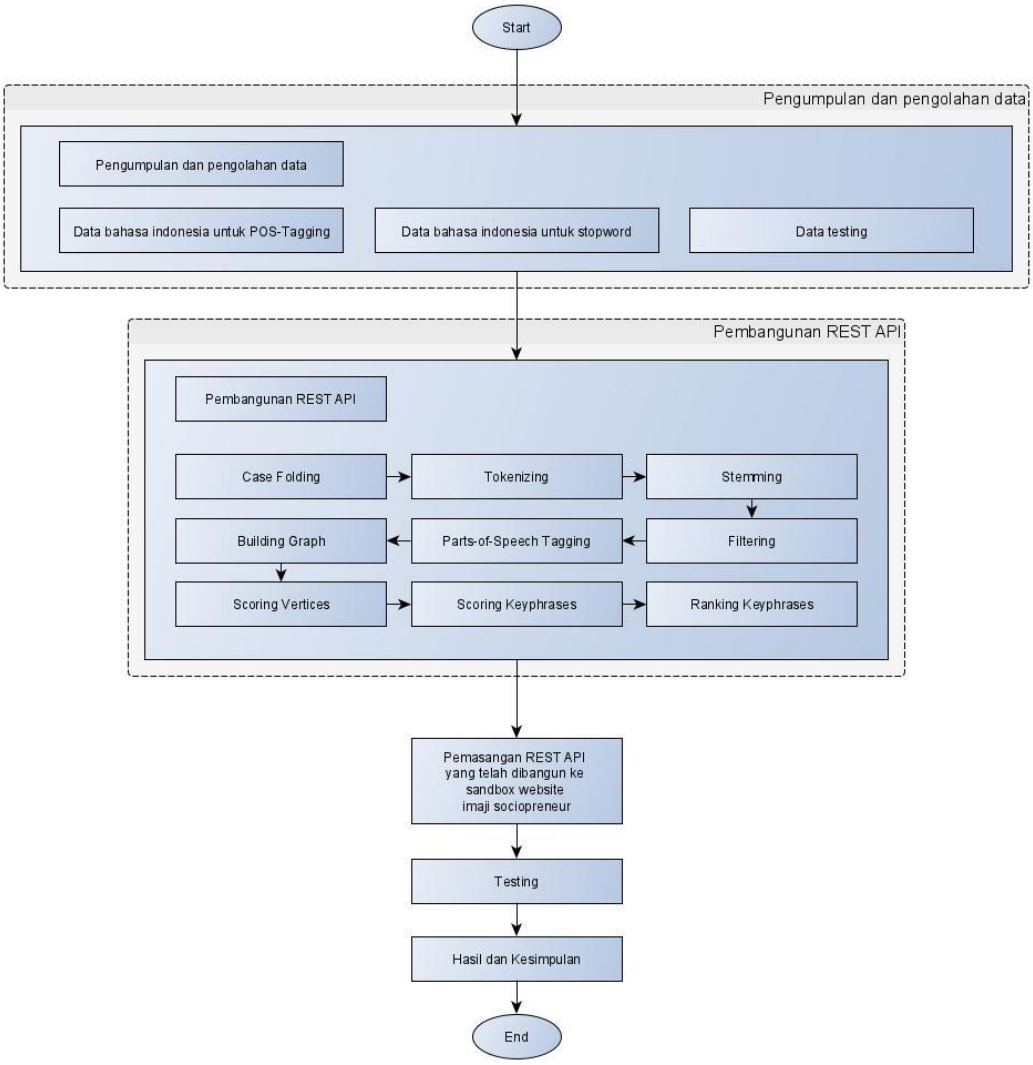
G.4 Gambaran Sistem

Sistem akan dibangun berdasarkan pada fungsionalitas yang ada pada batasan masalah dan mengembangkan *Website* Imaji Sociopreneur yang telah ada. Data yang akan diproses oleh *Website* Imaji Sociopreneur akan dikirim melalui *RESTful API* yang dibangun khusus untuk ekstraksi kata kunci. Pembangunan *RESTful API* dilakukan agar *Website* Imaji Sociopreneur tidak terbebani oleh proses ekstraksi kata kunci yang dilakukan. Selain itu, ekstraksi kata kunci dilakukan dalam pemrosesan dengan python, sedangkan Website Imaji Sociopreneur dibangun menggunakan *Framework* Laravel.

Dalam pemrosesan pada *RESTful API* ekstraksi kata kunci setelah data diterima akan dilakukan proses mulai dari cleaning data hingga *ranking keyphrases* dari hasil *ranking keyphrases* akan dikembalikan sebagai *string* yang telah berupa *keyword* yang telah di ekstraksi.

G.5 Tahapan Penlitian

Berikut merupakan alur dari tahapan penelitian dalam sistem :



Gambar 2. Tahapan Penelitian

G.5.1 Pengumpulan dan pengolahan data

Pengumpulan dan pengolahan data dibagi menjadi 3 bagian. Pertama, pengumpulan dan pengolahan data bahasa Indonesia untuk *POS-Tagging* dimana kata akan diubah olah untuk didapatkan jenis-jenisnya, misalnya kata benda, kata kerja kata hubung dan lain-lain. Kedua, pengumpulan dan pengolahan data bahasa Indonesia untuk *stopword* dimana kata akan dibersihkan dari kata yang tidak diperlukan. Ketiga, data untuk testing yang dilakukan

dengan cara crawling data pada website yang menyediakan hasil penelitian yang telah ada kata kuncinya yang akan dijadikan untuk mengukur keberhasilan serta data dari penulis di Website Imaji Sociopreneur.

G.5.2 Pembangunan *RESTful API*

Dalam tahap pembangunan *RESTful API* terdapat beberapa step dalam pemrosesan teks untuk menghasilkan kata kunci yang telah didapat dari teks yang dikirim dari website imajisociopreneur maupun postman untuk testing. Dimana tahap ini melakukan

1. *Case Folding*

   Case Folding adalah tahapan awal dari *text processing* yaitu merubah semua karakter yang ada data menjadi huruf kecil (*lowercase*). Dalam proses ini implementasinya akan menggunakan fungsi python *lower*() untuk mengubah menjadi huruf kecil.

2. *Tokenizing*

   Tokenizing adalah proses pemecahan dokumen yang terdiri dari kumpulan kalimat menjadi beberapa bagian kata yang disebut token. Dalam pada implementasinya akan menggunakan fungsi *word_tokenize(word)* dari library NLTK yang ada pada python.

3. *Stemming*

   Stemming merupakan tahapan yang diperlukan untuk memperkecil jumlah indeks dari suatu dokumen, berdasarkan kata penyusun dari dokumen tersebut. Pada proses stemming juga digunakan library Sastrawi untuk menemukan kata dasar.

4. Filtering

   Filtering merupakan tahapan pengambilan sejumlah kata penting dari hasil token yang telah didapatkan. Dalam hal ini tahapan algoritma yang dipakai adalah stopword, dimana data kata yang telah ada akan dicocokan dengan list stopword dan yang ada dalam kamus stopword maka data kata akan dihilangkan. Data stopword yang digunakan berasal dari https://www.ranks.nl/stopwords/indonesian.

5. *Parts-of-Speech Tagging*

Parts-of-Speech Tagging merupakan tahapan pemberian *tag* pada setiap *corpus*, *Parts-of-Speech Tagging* tidak hanya memberi *tag* pada kata namun juga pada symbol ataupun tanda baca. *Parts-of-Speech Tagging* yang digunakan menggunakan Flair *NLP* library yang dikembangkan oleh Puspita Kaban, Untuk melakukan *POS-tagging*, kita perlu membuat sebuah *POS-Tagger* yang terdiri atas *word embedding* dan *dictionary*. Sederhananya, *word embedding* adalah representasi dari kata-kata ke dalam sebuah vektor. Adapun library pada *tag*ger ini dibangun dari sebuah *corpus* (kumpulan kata-kata) yang sudah ditandai. Flair *NLP* sudah menyediakan corpus bahasa Indonesia yang dapat digunakan untuk *POS*-Tagging (Kaban, 2019). Namun Flair tidak menyediakan secara langsung *POS-tagging* berbahasa Indonesia maka diperlukan train library *POS-tagging*, yang dibuat dengan cara sebagai berikut

```
# 1. get the corpus
corpus = NLPTaskDataFetcher.load_corpus(NLPTask.UD_INDONESIAN)

# 2. what tag do we want to predict?
tag_type = 'upos'

# 3. make the tag dictionary from the corpus
tag_dictionary = corpus.make_tag_dictionary(tag_type=tag_type)
print(tag_dictionary.idx2item)

# 4. initialize embeddings
embedding_types: List[TokenEmbeddings] = [
    WordEmbeddings('id-crawl'),
    WordEmbeddings('id'),
    #WordEmbeddings('glove'),
    #BertEmbeddings('bert-base-multilingual-cased')
]

embeddings: StackedEmbeddings = StackedEmbeddings(embeddings=embedding_types)

# 5. initialize sequence tagger
from flair.models import SequenceTagger
tagger: SequenceTagger = SequenceTagger(hidden_size=256,
                                        embeddings=embeddings,
                                        tag_dictionary=tag_dictionary,
                                        tag_type=tag_type,
                                        use_crf=True)

from flair.trainers import ModelTrainer

trainer: ModelTrainer = ModelTrainer(tagger, corpus)

# 7. start training
trainer.train('resources/taggers/example-universal-pos',
              learning_rate=0.1,
              mini_batch_size=32,
              max_epochs=10)
```

Gambar 3. Pembentukan *corpus POS-Tagging* Bahasa Indonesia

Maka setelah itu library flair untuk *POS-Tagging* berbahasa Indonesia dapat digunakan.

6. *Building Graph*

Building Graph adalah tahapan pertama dalam *textrank*. *Building graph* dilakukan karena *textrank* adalah model berbasis graf. Setiap kata dalam kosakata akan berfungsi sebagai simpul untuk graf. Kata-kata tersebut akan direpresentasikan di simpul oleh *index*nya dalam daftar kosakata.

Building graph dilakukan dengan cara pemanfaatan library math dan numpy pada python untuk membantu pembangunan graf.

7. S*coring Vertices*

   Scoring Vertices adalah tahapan node atau simpul yang telah dibuat pada tahap 6 akan di hitung menggunakan persamaan (F.1) yang akan didapatkan nilai tiap *vertex* yang akan digunakan untuk penentuak *keyphrases* atau frasa unik.

8. *Scoring Keyphrases*

   Scoring Keyphrases adalah tahapan menilai frasa (frasa kunci kandidat) dan membangun daftar frasa kunci dengan membuat daftar versi frasa tokenized \ kandidat-frasa kunci. Frasa dinilai dengan menambahkan skor anggotanya (kata \ unit teks yang diberi peringkat oleh algoritma graf).

9. *Ranking Keyphrases*

   Ranking Keyphrases adalah tahapan memberi peringkat frasa kunci berdasarkan skor yang telah dihitung pada proses sebelumnya. *Ranking Keyphrases* dilakukan dengan menggunakan numpy untuk melakukan *sorting*.

G.5.3 Penghubungan *RESTful API* Textrank yang telah dibangun ke *Website* Imaji Sociopreneur

Pemasangan *RESTful API* yang telah dibangun pada proses G.4.2 dilakukan menggunakan *request RESTful API* milik *Website* Imaji Sociopreneur untuk menambahkan data ke *database* tulisan yang bersangkutan.

H. LUARAN YANG DIHARAPKAN

Luaran yang diharapkan dari penelitian ini yaitu:

1. Digunakan sebagai seminar proposal
2. Skripsi sebagai tugas akhir
3. Rekomendasi bagi objek penelitian
4. Jurnal yang dipublikasikan
5. SEO Support untuk *Website* Imaji Sociopreneur

I. JADWAL KEGIATAN

Pengerjaan skripsi ini diperlukan beberapa tahap untuk menyelesaikan, berupa jadwal kegiatan sebagai berikut :

Tabel 6. Jadwal Kegiatan

No	Tahapan Penelitian	Februari				Maret				April				Mei			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Penyusunan dan pengajuan proposal																
2	Seminar Proposal																
3	Pengumpulan dan pengolahan data																
4	Pembangunan REST API																
5	Pemakaian REST API yang telah dibuat ke sandbox Website imaji sociopreneur																
6	Testing																
7	Hasil dan kesimpulan																
8	Penulisan laporan skripsi																

Daftar Pustaka

Eris, Mawardi, V. C., & Pragantha, J. (2017). PENERAPAN ALGORITMA TEXTRANK UNTUK AUTOMATIC SUMMARIZATION PADA DOKUMEN BERBAHASA INDONESIA. *Jurnal Ilmu Teknik Dan Komputer*, *1*(1), 71–78. https://publikasi.mercubuana.ac.id

Fernandes, S., & Vidyasagar, A. (2015). Digital Marketing and Wordpress. *Indian Journal of Science and Technology*, *8*(12), 83–89. https://doi.org/10.17485/ijst/2015/v8i

Google Developer. (2021). *Ringkasan tentang crawler Google*. https://developers.google.com/search/docs/advanced/crawling/overview-google-crawlers?hl=id

Kaban, P. (2019). *POS-Tagging Bahasa Indonesia dengan Flair NLP*. Medium. https://puspitakaban.medium.com/pos-tagging-bahasa-indonesia-dengan-flair-nlp-c12e45542860

Lane, H., Howard, C., & Hapke, H. M. (2019). *Natural Language Processing in Action(Understanding,analyzing, and generating text with python)*.

Mihalcea, R., & Tarau, P. (2004). TextRank : Bringing Order into Texts. *ResearchGate - Conference Paper July 2004, May 2014*.

Muttaqin, F. A., & Bachtiar, A. M. (2016). Implementasi Teks Mining Pada Aplikasi Pengawasanpenggunaan Internet Anak "Dodo Kids Browser." *Jurnal Ilmiah Komputer Dan Informatika*.

Najjichah, H., Syukur, A., & Subagyo, H. (2019). Pengaruh Text Preprocessing Dan Kombinasinya Pada Peringkas Dokumen Otomatis Teks Berbahasa Indonesia. *Jurnal Teknologi Informasi*, *15*(1), 1–11.

Qingyun, Z., Yuansheng, F., Zhenlei, S., & Wanli, Z. (2020). Keyword Extraction Method for Complex Nodes Based on TextRank Algorithm. *Proceedings - 2020 International Conference on Computer Engineering and Application, ICCEA 2020*, 359–363. https://doi.org/10.1109/ICCEA50009.2020.00084

Reitz, J. M. (2020). *Online Dictionary for Library and Information Science*. ABC-CLIO, LLC. http://www.abc-clio.com/ODLIS/

Suhartono, D. (2019). *Part of speech tagging*. Binus. https://socs.binus.ac.id/2019/12/31/part-of-speech-tagging/

Wangsanegara, N. K., & Subaeki, B. (2015). Implementasi Natural Language Processing Dalam Pengukuran Ketepatan Ejaan Yang Disempurnakan (Eyd) Pada Abstrak Skripsi Menggunakan Algoritma Fuzzy Logic. *Jurnal Teknik Informatika*, *8*(2). https://doi.org/10.15408/jti.v8i2.3185

Wibisono, G., & Susanto, W. E. (2015). Perancangan Website Sebagai Media Informasi Dan Promosi Batik Khas Kabupaten Kulonprogo. *Jurnal Evolusi*, *3*(2), 64–69. https://ejournal.bsi.ac.id/ejurnal/index.php/evolusi/article/view/630

Wongchaisuwat, P. (2019). Automatic Keyword Extraction Using TextRank. *2019 IEEE 6th International Conference on Industrial Engineering and Applications, ICIEA 2019*, 377–381. https://doi.org/10.1109/IEA.2019.8714976