

Pengembangan *Library* Sistem Pemeringkat Otomatis Berbasis Kata Sifat

Faisal Rahutomo¹, Yushintia Pramitarini², Muhammad Elfa Rodhian Putra³

¹Teknik Informatika, Teknologi Informasi, Politeknik Negeri Malang
e-mail: faisal@polinema.ac.id

²Teknik Informatika, Teknologi Informasi, Politeknik Negeri Malang
e-mail: yushintia@polinema.ac.id

³Teknik Informatika, Teknologi Informasi, Politeknik Negeri Malang
e-mail: elfanrodhian@gmail.com

Abstrak

Saat ini khususnya di Indonesia banyak beredar ide, opini, kritik atau saran di internet dalam bentuk *website* atau *blog* pribadi. Hal-hal tersebut dapat dinilai menggunakan analisis semantik berdasarkan kata sifat sehingga menghasilkan sebuah rating dari objek yang dinilai. Penelitian tentang pemeringkatan dengan menggunakan kata sifat yang berasal dari tulisan / teks dari internet (Khususnya Bahasa Indonesia) sudah pernah dilakukan menggunakan metode sistem pemeringkat otomatis berbasis kata sifat. Namun, penelitian tersebut hanya dibuat spesifik untuk suatu permasalahan tertentu. Ketika permasalahan yang akan diselesaikan berbeda, sistem harus dibuat ulang untuk menyesuaikan dengan permasalahan yang baru. Maka penelitian ini menyusun sebuah *library* agar dapat digunakan untuk berbagai macam objek tanpa harus menulis fungsi dari awal. Adapun hasil pengujian dari fungsi *library* yang menggunakan objek wisata sebagai data pengujian adalah memperoleh presentase error sebesar 15% dan PCC sebesar 0,99. Hasil dari pengujian komparabilitas adalah *library* dapat berjalan pada Python versi 3.x.x. *Library* yang dibuat telah diunggah di PyPI (Python Package Index) dan GitHub dengan Lisensi MIT sehingga dapat digunakan secara publik oleh pengguna Python.

Keywords: *Library, Analisa Semantic, Sistem Pemeringkat Otomatis Berbasis Kata Sifat*

Abstract

Recently through social media platform, Indonesian citizen able to express their idea, opinion, critics, and suggestion toward everything. Previous research has developed an automatic rating system for the specific issue from raw text. The researches develop a system called adjective based rating system. But, each research is developed for a specific issue. When another issue is needed to be solved, another new system has to be developed. To fulfill this open problem, this paper explains the development of Python library of the adjective-based automatic rating system in Indonesian. The experiment results show the library is working well as described by the algorithm. Experiment results of a case study of tourism object show 15% percentage error and 0.99 Pearson correlation. The library has been uploaded into PyPI as well as GitHub with MIT license, therefore publicly available.

Keywords: *Library, Semantic Analysis, Adjective-based Rating System*

1. Pendahuluan

Perkembangan teknologi informasi yang sangat pesat mempengaruhi cara masyarakat dalam berkomunikasi. Masyarakat lebih suka menyalurkan ide, opini, kritik atau saran lewat media *online*. Saat ini khususnya di Indonesia banyak beredar ide, opini, kritik atau saran di internet dalam bentuk *website* atau *blog* pribadi. Hal-hal tersebut dapat dinilai

menggunakan analisis *semantic* berdasarkan kata sifat yang ada pada tulisan masyarakat yang beredar di media *online*. Kata sifat atau adjektiva (bahasa Latin: *adjectivum*) adalah kelas kata yang mengubah kata benda atau kata ganti, biasanya dengan menjelaskannya atau membuatnya menjadi lebih spesifik. Kata sifat dapat menerangkan kuantitas,

kecukupan, urutan, kualitas, maupun penekanan suatu kata.

Pemeringkatan adalah proses menyusun urutan berdasarkan tolok ukur tertentu (Bahasa, 2016). Kedudukan dalam urutan itu disebut peringkat atau *rank*. Pemeringkatan terhadap suatu objek umumnya terdapat pada *website* tertentu yang memang terdapat fitur pemeringkat di dalamnya. Namun, *website* yang menyediakan fitur pemeringkatan hanya terbatas pada objek tertentu (Misal : makanan, tempat wisata, aplikasi) sedangkan banyak objek yang tidak memiliki sistem peratingan (Misal : universitas, tokoh masyarakat, barang tertentu).

Berdasarkan paparan diatas, penilaian masyarakat yang berupa ide, opini, kritik atau saran terhadap sesuatu dapat dijadikan tolok ukur untuk melakukan pemeringkatan berdasarkan kata sifatnya. Penelitian tentang pemeringkatan dengan menggunakan kata sifat yang berasal dari tulisan / *text* dari internet (khususnya Bahasa Indonesia) pernah dilakukan pada penelitian (Ramadhani, Rahmad, & Rahutomo, 2017), (Faisal Rahutomo, Adi, Rozi, & Saputra, 2018), dan (F Rahutomo et al., 2018). Namun, penelitian tersebut hanya dibuat spesifik untuk suatu permasalahan tertentu, ketika permasalahan yang akan diselesaikan berbeda, sistem harus dibuat ulang untuk menyesuaikan dengan permasalahan yang baru.

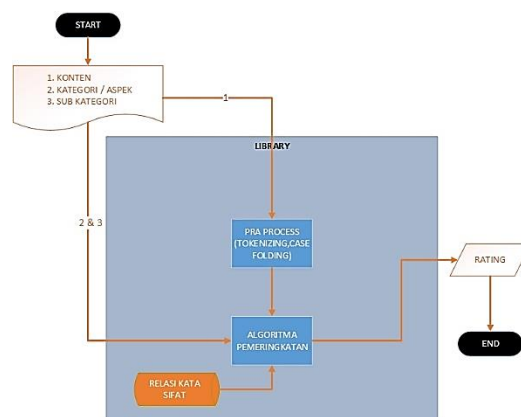
Oleh sebab itu perlu dibuat sebuah *library* yang memiliki fungsi sama seperti Sistem Pemeringkat Otomatis Berbasis Kata Sifat namun dapat menerima *input* secara dinamis agar dapat digunakan tanpa ada batasan permasalahan tertentu. Sehingga penelitian ini diberi judul "Pengembangan *Library* Sistem Pemeringkat Otomatis Berbasis Kata Sifat". Diharapkan dengan *Library* ini akan mempermudah pembuatan aplikasi serupa tanpa harus membuat dari awal.

2. Metode Penelitian

2.1. Sistem Pemeringkat Otomatis

Proses peratingan di dalam makalah ini menggunakan metode Sistem Pemeringkat Otomatis Berbasis Kata Sifat. Sistem Pemeringkat Otomatis Berbasis Kata Sifat adalah metode untuk mencari *rating* dari sebuah objek dengan menggunakan data berupa teks yang berkaitan dengan objek tersebut dengan cara menilai setiap

kata sifat yang mewakili nilai kualitas dari objek yang dinilai (Ramadhani et al., 2017), (F Rahutomo et al., 2018), (Faisal Rahutomo et al., 2018).



Gambar 1. *Workflow Diagram* Sistem Pemeringkat Otomatis Berbasis Kata Sifat

Tahapan awal dari sistem ini adalah *preprocessing*. Penelitian ini menggunakan beberapa tahap *preprocessing* yaitu *Case Folding*, *Tokenizing*, dan *Adjective Pass List*. *Case folding* adalah tahap dalam *text processing* yang bertujuan untuk mengubah semua huruf menjadi huruf kecil (*lowercase*) yang mana hanya huruf a sampai z yang diterima. *Tokenizing* adalah memotong sebuah kalimat menjadi bagian-bagian terkecil dari kalimat, yang disebut dengan *tokens* atau term dan di waktu yang sama menghapus atau menghilangkan tanda baca. *Adjective Pass List* merupakan tahapan dimana token-token yang dihasilkan proses sebelumnya di bandingkan dengan data antonim set dan sinonim setnya sehingga dapat dilakukan perhitungan selanjutnya.

Hasil dari *preprocessing* dicocokkan berdasarkan sinonim dan antonimnya dengan kata sifat pada sub kategori. Setelah didapat jumlah sinonim dan antonim dari sebuah sub kategori (SK), kemudian jumlah sinonim yang cocok dikurangkan dengan jumlah antonim yang cocok, perhitungan ini dapat dilihat pada Persamaan (1). Hasil dari perhitungan ini kemudian dilakukan *thresholding* (TH), yaitu jika nilai ≥ 1 maka nilai = 1, jika nilai ≤ -1 maka nilai = -1, jika nilai = 0 maka nilai = 0, proses ini dapat dilihat pada Persamaan (2). Setelah proses *thresholding* maka didapatkan nilai setiap

sub kategori (SK) yang kemudian dijumlahkan dari SK_1 sampai SK_n , dan menghasilkan nilai jumlah kategori, seperti pada Persamaan (3). Setelah mendapatkan hasil penjumlahan dari tiap sub kategori, kemudian hasil tersebut dibagi dengan banyaknya sub kategori, dan menghasilkan nilai rata-rata kategori, seperti pada Persamaan (4). Nilai rata-rata tersebut kemudian dikali dengan rating maksimal sistem, dalam penelitian ini rating maksimal sistem adalah 5 (bintang), seperti pada Persamaan (5). Proses terakhir adalah menghitung nilai *rating* akhir, yaitu menambahkan semua nilai *rating* kategori kemudian dibagi dengan banyaknya kategori.

$$\text{Nilai Kategori}(SK) = \sum \text{Sinonim} - \sum \text{Antonim} \quad (1)$$

$$\text{Threshold}(TH_{jn}) = \begin{cases} (-1), & \text{if } SK \leq (-1) \\ 0, & \text{if } SK = 0 \\ 1, & \text{if } SK \geq 1 \end{cases} \quad (2)$$

$$\sum \text{Nilai Kategori} = SK_1 + SK_2 + \dots + SK_n \quad (3)$$

$$\text{Rata - rata Kategori} = \frac{\sum \text{Nilai Kategori}}{\sum \text{Kategori}} \quad (4)$$

$$\text{Rating Kategori} = \text{Rata - rata Kategori} \times \sum \text{Bintang} \quad (5)$$

$$\text{Rating} = \frac{RK_1 + RK_2 + \dots + RK_n}{\sum \text{Kategori}} \quad (6)$$

2.2. Pengembangan *Library* Python

Dalam penelitian ini *library* dibuat menggunakan Bahasa pemrograman Python. *File* Python dapat didistribusikan dan digunakan kembali asalkan *file* tersebut ditulis untuk versi Python yang tepat dan hanya menggunakan *library* standar (Python Software Foundation, 2019).

Tahapan awal dari pembuatan *library* ini adalah mendefinisikan *class*. *Class* adalah suatu *frame* yang merupakan definisi yang memuat data dan *method* pengolah data. *Class* juga dapat diartikan sebagai tempat untuk membuat obyek. Di dalam *class* dideklarasikan *variable* dan *method* yang dimiliki oleh obyek. Proses pembuatan obyek dari sebuah *class* disebut dengan *instantiation* (Pressman, 2010). *Library* yang dibuat merupakan sebuah *class* dengan spesifikasi yang dijelaskan pada Tabel 1.

Kemudian dilakukan definisi *method*. Fungsi (*Function*) disebut juga *method* adalah suatu program terpisah dalam blok sendiri yang berfungsi

Tabel 1. Spesifikasi *Class*

		Keterangan
Nama	Spoks	Class
Atribut	maks (integer)	maksimum <i>rating</i> yang diinginkan
	pre (array)	Hasil <i>preprocessing</i>
	sc (array)	<i>Value</i> dari tiap sub kategori (menghitung jumlah kata sifat yang cocok antara data konten dengan sub kategori beserta sinset dan antset)
	th (array)	<i>Threshold</i> (normalisasi hasil sc)
	c (array)	<i>Value</i> dari tiap kategori (menjumlahkan th dari tiap kategori)
	jumlah (array)	Jumlah kategori (jumlah kategori yang didefinisikan)
	avg (array)	<i>Average</i> tiap kategori (membagi c dengan jumlah)
	cr (array)	<i>Rating</i> tiap kategori (megalikan avg dengan maks)
	fr (integer)	<i>Rating</i> akhir (rate-rata cr)

sebagai sub-program (modul program) untuk memproses sebagian dari pekerjaan program utama. *Library* yang dibuat memiliki beberapa fungsi dengan *input* dan *output* yang dijelaskan pada Tabel 2.

Kemudian penggunaan *library* seperti menggunakan *class* biasa di dalam bahasa pemrograman Python. Langkah-langkah penggunaan fungsi dalam *class* adalah *import*, instansiasi, dan pemanggilan fungsi. *Import* bertujuan untuk memanggil *file* dimana *class* tersebut disimpan sehingga *class* dapat digunakan. Contohnya, "from spoks import Spoks". Instansiasi adalah proses menciptakan *instance class* baru. Instansiasi berfungsi untuk mengubah *class* menjadi objek, contohnya, "spoks = Spoks()". Pemanggilan fungsi dilakukan dengan cara memanggil objek yang telah dibuat diikuti nama fungsi yang dipanggil. Contohnya, "pre = spoks.preproses(konten)". Contoh pemanggilan fungsi *preproses* yang ada pada *class*

Tabel 2. Fungsi *Library*

Fungsi	: preproses()
Input	: data konten (text)
Output	: token (array)
Ket	: Memproses input berupa text dengan casefolding dan tokenizing sehingga menghasilkan keluaran berupa token text huruf kecil
Fungsi	: Input_kategori()
Input	: kategori (array)
Output	: kategori (array)
Ket	: memproses input berupa array kategori kemudian menyimpannya kedalam file kategori.json
Fungsi	: Input_sub()
Input	: kategori (array) sub_kategori (array)
Output	: kategori (array) sub_kategori (array)
Ket	: Memproses input berupa array sub kategori untuk setiap kategori yang berbeda dan menyimpannya kedalam file kategori.json
Fungsi	: show_kategori()
Input	: null
Output	: kategori (array) sub_kategori (array)
Ket	: Menampilkan kategori dan sub kategori yang tersimpan
Fungsi	: spoks()
Input	: token (array)
Output	: sc value (array) threshold (array) c value (array) average (array) c rating (array) final rating (text) sub kategori (array)
Ket	: Memproses <i>input</i> berupa <i>token</i> hasil preproses menggunakan Metode Sistem Pemeringkat Otomatis Berbasis Kata Sifat sehingga menghasilkan <i>sc</i> , <i>th</i> , <i>c</i> , <i>avg</i> , <i>cr</i> yang merupakan <i>variable</i> yang dibutuhkan untuk perhitungan <i>final rating</i> Lalu hasil <i>output</i> akhirnya adalah <i>final rating</i>

Spoks yang telah diinstansikan menjadi objek spoks. Hasil *output* dari fungsi *preprocess()* dengan *input* berupa text "Aku suka makan Soto" ditampung dalam suatu *variable*. Hasil *output* dari contoh baris kode diatas adalah array {"aku, suka, makan, soto"}.

Library yang dibuat harus diunggah kedalam repositori agar dapat diinstall secara *public* oleh pengguna Python lainnya. Dalam penelitian ini repositori yang digunakan adalah *PyPI*. Untuk mengunggah *library* ke dalam *PyPI* perlu membuat direktori dimana *file* yang akan diunggah sebagai *library* berada di direktori tersebut. Penelitian ini mendefinisikannya di dalam "packaging_tutorial>spoks>__init__.py". Setelah membuat direktori dan *file*, masuk ke direktori "packaging_tutorial", kemudian edit file "spoks/__init__.py" untuk menambahkan nama paket, "name = "PySPOKS"".

Selanjutnya membuat beberapa *file* untuk mengemas proyek dan menyiapkannya untuk distribusi. *File-file* tersebut yaitu: *setup.py*, *README.md*, dan *LICENSE*. Semuanya masuk ke dalam folder "packaging_tutorial".

File "setup.py" adalah skrip *build* yang berisi informasi tentang paket kepada *setuptools* (seperti nama dan versi) serta *file* kode mana yang akan disertakan. Nama paket dibuat unik agar tidak bentrok dengan paket yang diunggah oleh orang lain. Isi kode terdapat di dalam Gambar 2. *File* "README.md" berisi deskripsi dari *library* yang diunggah. *File* ini juga dapat diedit sesuai dengan kebutuhan. *File* "LICENSE" sangat penting untuk setiap paket yang diunggah ke *PyPI*. *Library* yang dibangun di dalam makalah ini memilih lisensi MIT.

```
import setuptools

with open("README.md", "r") as fh:
    long_description = fh.read()

setuptools.setup(
    name="PySpoks",
    version="0.0.1",
    author="ElfanRodh",
    author_email="elfarodhian@gmail.com",
    description="Sistem Pemeringkat Otomatis Berbasis Kata Sifat",
    long_description=long_description,
    long_description_content_type="text/markdown",
    url="https://github.com/pypa/sampleproject",
    packages=setuptools.find_packages(),
    classifiers=[
        "Programming Language :: Python :: 3",
        "License :: OSI Approved :: MIT License",
        "Operating System :: OS Independent",
    ],
)
```

Gambar 2. Setting *setup.py*

```

Copyright (c) 2018 The Python Packaging
Authority
Permission is hereby granted, free of charge,
to any person obtaining a copy of this
software and associated documentation files
(the "Software"), to deal in the Software
without restriction, including without
limitation the rights to use, copy, modify,
merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to
permit persons to whom the Software is
furnished to do so, subject to the following
conditions:
The above copyright notice and this
permission notice shall be included in all
copies or substantial portions of the
Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT
WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE AND NONINFRINGEMENT. IN NO EVENT
SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR
IN CONNECTION WITH THE SOFTWARE OR THE USE
OR OTHER DEALINGS IN THE SOFTWARE.

```

Gambar 3. License

Selanjutnya adalah membuat paket distribusi yang diunggah ke *PyPI* dan dapat diinstal oleh pip. Versi terbaru dari komponen Python "setuptools" dan "wheel" dipastikan terpasang dengan perintah, "python -m pip install --user --upgrade setuptools wheel". Lalu menjalankan perintah *generate* distribusi ini dari direktori yang sama di mana *file* "setup.py" berada dengan perintah "python3 setup.py sdist bdist_wheel". Perintah ini akan menghasilkan banyak teks dan setelah selesai akan menghasilkan dua *file* di direktori dist, yaitu "PySpoks-0.0.1-py3-none-any.whl" dan "PySpoks-0.0.1.tar.gz".

Setelah membuat akun di <https://pypi.org/> *file* paket distribusi diunggah ke *PyPI* dengan menggunakan komponen "twine". Komponen "twine" telah dipastikan versi terbaru dengan perintah "python3 -m pip install --user --upgrade twine". Mengunggah semua arsip *file* distribusi menggunakan perintah "python3 -m twine upload dist/*". Proses mengunggah membutuhkan nama pengguna dan kata sandi yang telah didaftarkan sebelumnya di *PyPI*. Gambar 4 menunjukkan layar setelah perintah selesai dijalankan. Setelah diunggah, paket dapat dilihat di *link* berikut: <https://pypi.org/project/PySpoks> (Putra, Rahutomo, & Pramitarini, 2019). Kemudian library dapat digunakan siapapun dengan menggunakan perintah "pip install PySpoks". Library juga dapat diakses di GitHub (Putra, Rahutomo, & Pramitarini, 2019a).

```

Uploading distributions to
https://pypi.org/legacy/
Enter your username: [your username]
Enter your password:
Uploading PySpoks-0.0.1-py3-none-
any.whl
100%|████████████████████|
4.65k/4.65k [00:01<00:00, 2.88kB/s]
Uploading PySpoks-0.0.1.tar.gz
100%|████████████████████|
4.25k/4.25k [00:01<00:00, 3.05kB/s]

```

Gambar 4. Hasil Upload File Distribusi

2.3. Metode Pengujian

Pengujian digunakan untuk mengetahui apakah sistem yang dibangun sudah benar dan sesuai dengan yang dibutuhkan. Langkah-langkah pengujian dilakukan dengan 3 cara yaitu: *black box*, *perhitungan*, dan *performa*.

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini berusaha menemukan kesalahan fungsi-fungsi yang kurang benar, kesalahan *input* dan *output*, atau kesalahan struktur data.

Pengujian perhitungan dilakukan dengan cara membandingkan hasil dari perhitungan manual setiap proses dengan menggunakan data sampel. Sampel data yang telah dihitung secara manual diterapkan pada sistem dan melihat setiap prosesnya apakah sudah sesuai dengan perhitungan manual.

Pengujian performa dilakukan untuk menjamin dan memastikan bahwa sistem yang dirancang dapat berjalan seperti yang diharapkan. Pengujian performa terdiri dari 2 tahap pengujian, yaitu pengujian *Pearson Coefficient of Correlation* (PCC) dengan Persamaan (7) dan Persentase *Error* dengan Persamaan (8). Penelitian ini mengambil *rating* dari *Google Maps* sebagai nilai pembanding dalam pengujian. Data *rating* dari *Google Maps* terdapat di dalam Tabel 3. Tabel 3 menunjukkan *rating* yang dikumpulkan dengan cara menginputkan kata kunci pada kolom pencarian Aplikasi *Google Maps*.

$$r = \frac{\sum xy}{\sqrt{\sum x^2 \sum y^2}} \quad (7)$$

$$\% \text{ Error} = \frac{|x - y|}{x} \times 100\% \quad (8)$$

Keterangan:

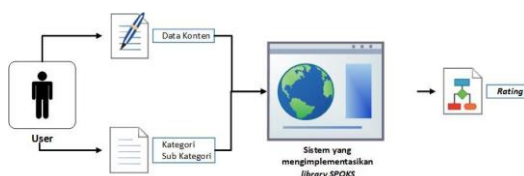
x = nilai data 1

y = nilai data 2

Tabel 3. *Rating* dari Google Maps

No	Wisata	Rating Google Maps (diakses 8 Juni 2019)
1	Pantai Ngudel	4.5
2	Pantai Goa Cina	4.4
3	Coban Rondo	4.4
4	Coban Rais	4.1
5	Batu Night Spectacular	4.4
6	Museum Angkut	4.6
7	Paralayang	4.6
8	Jawa Timur Park 3	4.4
9	Selecta	4.4
10	Omah Kayu	4.3

2.4. Implementasi pada Kasus Wisata



Gambar 5. Gambaran Umum Sistem

Pada penelitian ini dibuat sebuah aplikasi berbasis web untuk mengimplementasikan *library* Sistem Pemeringkat Otomatis Berbasis Kata Sifat. Peneliti menggunakan tempat wisata sebagai objek penelitian. Sistem yang dibuat dapat menerima *input* berupa aspek penilaian dan konten kemudian menghasilkan *output* berupa *rating*.

2.4.1. Pengambilan Data

Terdapat tiga tahap dalam pengumpulan data penilaian objek wisata. Yang pertama adalah mengumpulkan data lemma kata sifat. Kata sifat yang digunakan dalam penelitian ini adalah kata sifat yang telah terdaftar di dalam 8.005 lemma kata sifat yang telah dibuat sebelumnya.

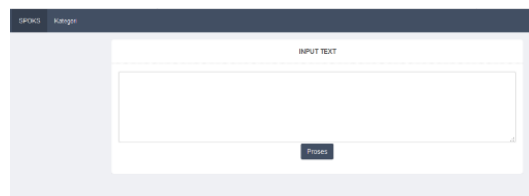
Tahap kedua adalah mengumpulkan 10 nama tempat wisata secara acak, pada penelitian ini tempat wisata yang dikumpulkan adalah yang berada di daerah Malang Raya. Data ini yang kemudian akan dijadikan acuan untuk penilaian kinerja sistem yang dibuat.

Tahap ketiga adalah *crawling data* menggunakan *Google Custom Search API* berdasarkan pada kata kunci nama objek wisata yang dikumpulkan pada tahap kedua. Hasil *crawling* akan disimpan kedalam file

.json untuk selanjutnya menjadi *input* dari sistem yang dibuat.

2.4.2. Input Konten

Halaman pada Gambar 6 adalah halaman utama dari sistem sekaligus menjadi halaman *input* dari sistem. Pada halaman ini terdapat beberapa *item*, dipaparkan pada Tabel 4.

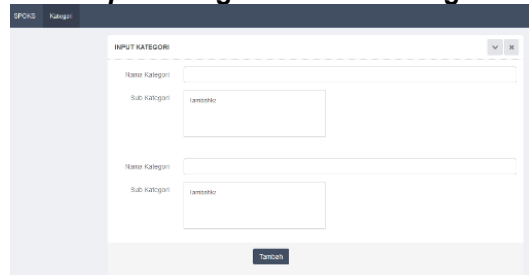


Gambar 6. Antarmuka Halaman Utama

Tabel 4. *Item* Halaman Utama

No	Jenis	Nama	Fungsi
1	Label	Input Text	Text Petunjuk
2	Text Area	Input Dataset	Input <i>dataset</i> untuk proses selanjutnya
3	Button	Proses	Aksi pemrosesan <i>dataset</i>
4	Menu	SPOKS	Pindah ke halaman SPOKS
5	Menu	Kategori	Pindah ke halaman kategori

2.4.3. Input Kategori dan Sub Kategori



Gambar 7. Tambah Kategori dan Sub Kategori

Tabel 5. *Item* Halaman Tambah Kategori dan Sub Kategori

No	Jenis	Nama	Fungsi
1	Label	Nama Kategori	Text Petunjuk
2	Input Text	Kategori	Input nama kategori
3	Input Multi Tags	Sub Kategori	Input sub kategori untuk tiap kategori yang berbeda
3	Button	Proses	Aksi pemrosesan selanjutnya
4	Menu	SPOKS	Pindah ke halaman SPOKS
5	Menu	Kategori	Pindah ke halaman kategori

Halaman pada Gambar 7 merupakan tampilan untuk menginputkan kategori dan sub kategori yang akan digunakan sebagai acuan penilaian. Pada halaman ini terdapat beberapa *item*, dijabarkan pada Tabel 5.

Halaman pada Gambar 8 merupakan tampilan hasil penilaian dari *input dataset* berdasarkan kategori dan sub kategori yang telah ditentukan. Pada halaman ini terdapat beberapa *item* yang ada pada Tabel 6.

2.4.4. Hasil Rating

Kategori	C	Average	CR
Akan Jajan	2	1.8	0.0
Paket	9	1.8	0.0
Kategori	3	0.5	2.5
Jasa Pengantar	1	0.25	1.25
Raja Matak	3	1.8	0.0
Emosional	3	0.42857142857142855	2.142857142857143

Gambar 8. Hasil Rating

Tabel 6. *Item* Halaman Hasil Rating

No	Jenis	Nama	Fungsi
1	Label	Hasil Peratingan	Text Petunjuk
2	Label	Hasil	Jumlah rating yang didapatkan dari perhitungan menggunakan metode
3	Tabel	Tabel Detail Penilaian	Tabel perhitungan langkah-langkah penilaian menggunakan <i>library</i>
4	Menu	SPOKS	Pindah ke halaman SPOKS
5	Menu	Kategori	Pindah ke halaman kategori

3. Hasil dan Pembahasan

3.1. Pengujian Black Box

Pengujian black box memberikan hasil *library* yang dikembangkan berjalan sebagaimana yang direncanakan. *Library* berfungsi baik di tahap instalasi. Semua *method* di dalam *library* yang didefinisikan di Tabel 2 juga telah berfungsi sebagaimana yang direncanakan pada Python versi 3.7.x. Akan tetapi beberapa fungsi tidak berjalan pada versi yang lain. *Library* yang dibuat diuji menggunakan beberapa versi Python yang berbeda untuk melihat kompatibilitasnya. Hasil pengujian kompatibility ditampilkan dalam Tabel 7. Maka dapat disimpulkan

bahwa *library* ini kompatibel pada Python versi 3 ke atas.

Tabel 7. Uji Kompatibilitas

No	Versi Python	Tgl. Rilis	Hasil	Ket.
1	2.7.14	16/09/2017	Tidak Kompatibel	Beberapa fungsi Python tidak support versi ini
2	2.7.15	01/05/2018	Tidak Kompatibel	Beberapa fungsi Python tidak support versi ini
3	3.7.2	24/12/2018	Kompatibel	Semua fungsi berjalan dengan baik
4	3.7.3	25/03/2019	Kompatibel	Semua fungsi berjalan dengan baik

Tabel 8. Rating dari Sistem

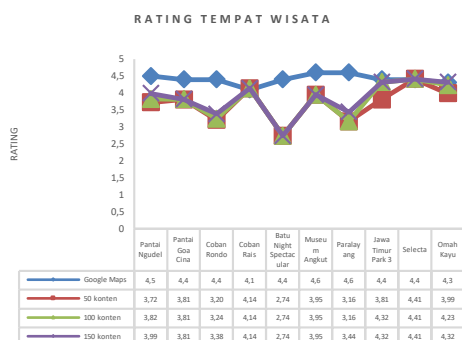
No	Wisata	Nilai Rating Sistem		
		50 konten	100 konten	150 konten
1	Pantai Ngudel	3.72	3.82	3.99
2	Pantai Goa Cina	3.81	3.81	3.81
3	Coban Rondo	3.20	3.24	3.38
4	Coban Rais	4.14	4.14	4.14
5	Batu Night Spectacular	2.74	2.74	2.74
6	Museum Angkut	3.95	3.95	3.95
7	Paralayang	3.16	3.16	3.44
8	Jawa Timur Park 3	3.81	4.32	4.32
9	Jodipan	4.41	4.41	4.41
10	Omah Kayu	3.99	4.23	4.23

3.2. Pengujian Perhitungan Rating

Hasil pengujian dengan membandingkan proses rating satu teks sederhana yang dihitung secara manual dengan fungsi yang dikembangkan menunjukkan hasil yang konsisten. Kemudian penelitian mengambil hasil penilaian 10 tempat wisata yang sudah ditentukan dengan mengatur pengambilan konten dalam 3 varian jumlah, yaitu 50 konten, 100 konten dan 150 konten. Data yang telah dikumpulkan terdapat di dalam Tabel 8.

Pada Gambar 9 dapat dilihat bahwa terdapat beberapa perbedaan pada *rating* dari *Google Maps* dan *rating* dari sistem. Namun, pada beberapa konten memiliki *rating* yang sama dengan *rating* di *Google Maps*. nilai *rating* pada konten tergantung pada jumlah konten yang dinilai serta isi konten tersebut dimana semakin banyak ditemukan kata sifat yang termasuk kedalam sub kategori penilaian serta sinonim set dan antonim setnya, maka *rating*-nya akan semakin banyak.

Beberapa *rating* yang didapat dari sistem memiliki nilai yang hampir sama dengan *rating* pada *Google Maps* seperti Coban Rais, Jatim Park 3, Selecta dan Omah Kayu. Hal ini dapat disebabkan karena banyak opini masyarakat yang mengulas tentang tempat wisata tersebut secara detail dan memuat banyak kata sifat positif begitu juga sebaliknya untuk tempat wisata yang lain.



Gambar 9. Perbandingan *Rating* Tempat Wisata

3.3. Pengujian Performa

3.3.1. Presentase Error

Data *rating* dari hasil perhitungan menggunakan sistem dibandingkan dengan data *rating* yang diperoleh dari *Google Maps* dengan menggunakan Persamaan (8). Perhitungan presentase *error* untuk setiap variasi data berbeda. Hasil perhitungan presentase *error* untuk 50 konten dijelaskan pada Tabel 9, untuk 100 konten dijelaskan pada Tabel 10, untuk 150 konten dijelaskan pada Tabel 11.

Perhitungan nilai *error* dilakukan sesuai dengan Persamaan (8) yang dijelaskan sebelumnya. Contoh Pada tempat wisata Pantai Ngudel memiliki nilai *error* antara *rating* *Google Maps* dengan sistem (50 konten) adalah 17%, lalu dengan sistem (100 konten) adalah 15%, dan

dengan sistem (150 konten) adalah 11%. Perhitungan ini dilakukan di 10 tempat wisata, sehingga peneliti mendapatkan nilai rata-rata *error* sistem (50 konten) adalah 16%, sistem (100 konten) adalah 14%, dan sistem (150 konten) adalah 13%. Kemudian rata-rata seluruh *error* adalah 14.333%

Tabel 9. Presentase *Error* 50 konten

No	Wisata	Google Maps	Sistem	
			50 konten	
			<i>rating</i>	<i>error</i>
1	Pantai Ngudel	4.5	3.72	17%
2	Pantai Goa Cina	4.4	3.81	13%
3	Coban Rondo	4.4	3.20	27%
4	Coban Rais	4.1	4.14	1%
5	Batu Night Spectacular	4.4	2.74	38%
6	Museum Angkut	4.6	3.95	14%
7	Paralayang	4.6	3.16	31%
8	Jawa Timur Park 3	4.4	3.81	13%
9	Selecta	4.4	4.41	0%
10	Omah Kayu	4.3	3.99	7%
Σ <i>error</i>				16%

Tabel 10. Presentase *Error* 100 konten

No	Wisata	Google Maps	Sistem	
			100 konten	
			<i>rating</i>	<i>error</i>
1	Pantai Ngudel	4.5	3.82	15%
2	Pantai Goa Cina	4.4	3.81	13%
3	Coban Rondo	4.4	3.24	26%
4	Coban Rais	4.1	4.14	1%
5	Batu Night Spectacular	4.4	2.74	38%
6	Museum Angkut	4.6	3.95	14%
7	Paralayang	4.6	3.16	31%
8	Jawa Timur Park 3	4.4	4.32	2%
9	Selecta	4.4	4.41	0%
10	Omah Kayu	4.3	4.23	2%
Σ <i>error</i>				14%

Tabel 11. Presentase *Error* 150 konten

No	Wisata	Google Maps	Sistem	
			50 konten	
			rating	error
1	Pantai Ngudel	4.5	3.99	11%
2	Pantai Goa Cina	4.4	3.81	13%
3	Coban Rondo	4.4	3.38	23%
4	Coban Rais	4.1	4.14	1%
5	Batu Night Spectacular	4.4	2.74	38%
6	Museum Angkut	4.6	3.95	14%
7	Paralayang	4.6	3.44	25%
8	Jawa Timur Park 3	4.4	4.32	2%
9	Selecta	4.4	4.41	0%
10	Omah Kayu	4.3	4.32	0%
∑ error				13%

Tabel 12. *PCC* pada 50 konten

No	Wisata	Google Maps		Sistem		
		x	x2	50 konten		
				y	y2	xy
1	Pantai Ngudel	4.5	20.3	3.72	13.84	16.74
2	Pantai Goa Cina	4.4	19.4	3.81	14.52	16.76
3	Coban Rondo	4.4	19.4	3.20	10.27	14.1
4	Coban Rais	4.1	16.8	4.14	17.14	16.97
5	Batu Night Spectacular	4.4	19.4	2.74	7.508	12.06
6	Museum Angkut	4.6	21.2	3.95	15.6	18.17
7	Paralayang	4.6	21.2	3.16	9.986	14.54
8	Jawa Timur Park 3	4.4	19.4	3.81	14.52	16.76
9	Selecta	4.4	19.4	4.41	19.45	19.4
10	Omah Kayu	4.3	18.5	3.99	15.92	17.16
∑			195		138.7	162.7
PCC = $\frac{\sum xy}{\sqrt{\sum x^2 \sum y^2}}$				0.989777996		

3.3.2. PCC

Pengujian ini menggunakan data dari hasil *crawling* sejumlah 50, 100, dan 150 konten. Hasil perhitungan PCC untuk 50, 100, dan 150 konten dijelaskan berturut-turut ke dalam Tabel 12, 13, dan 14.

Pada Tabel 12, tempat wisata Pantai Ngudel memiliki nilai rating 4.5 dari Google Maps, nilai tersebut sebagai nilai x. Kemudian di kolom selanjutnya adalah x2 yang berarti perkalian kuadrat dari x, dengan hasil 20.3. Setelah mendapat nilai x, selanjutnya menentukan nilai y. Hasil nilai *rating* tempat wisata tersebut dari sistem aplikasi adalah 3.72, nilai ini adalah nilai y. Serta dilakukan perkalian kuadrat dari y pada kolom y2, yang bernilai 13.84. Setelah mendapat nilai x dan y, maka selanjutnya adalah menentukan perkalian dari keduanya di kolom xy, yaitu bernilai 16.74. Perhitungan ini dilakukan pada seluruh tempat wisata, sehingga di didapatkan jumlah nilai dari x2 yaitu 195, y2 sejumlah 138.7 dan jumlah nilai xy yaitu 162.7. Kemudian perhitungan PCC dapat dilihat di bagian akhir Tabel 12, nilai PCC pada program yang menggunakan data 50 konten yaitu 0.989777996.

Tabel 13. *PCC* pada 100 konten

No	Wisata	Google Maps		Sistem		
		x	x2	100 konten		
				y	y2	xy
1	Pantai Ngudel	4.5	20.3	3.82	14.59	17.19
2	Pantai Goa Cina	4.4	19.4	3.81	14.54	16.78
3	Coban Rondo	4.4	19.4	3.24	10.52	14.27
4	Coban Rais	4.1	16.8	4.14	17.14	16.97
5	Batu Night Spectacular	4.4	19.4	2.74	7.51	12.06
6	Museum Angkut	4.6	21.2	3.95	15.60	18.17
7	Paralayang	4.6	21.2	3.16	9.99	14.54
8	Jawa Timur Park 3	4.4	19.4	4.32	18.66	19.01
9	Selecta	4.4	19.4	4.41	19.45	19.40
10	Omah Kayu	4.3	18.5	4.23	17.89	18.19
∑			195		145.9	166.6
PCC = $\frac{\sum xy}{\sqrt{\sum x^2 \sum y^2}}$				0.988438106		

Pada Tabel 13., tempat wisata Pantai Ngudel memiliki nilai x dan x^2 yang sama dengan Tabel 12. Hasil nilai *rating* tempat wisata tersebut dari sistem aplikasi adalah 3.82, nilai ini adalah nilai y . Serta dilakukan perkalian kuadrat dari y pada kolom y^2 , yang bernilai 14.59. Setelah mendapat nilai x dan y , maka selanjutnya adalah menentukan perkalian dari keduanya di kolom xy , yaitu bernilai 17.19. Perhitungan ini dilakukan pada seluruh tempat wisata, sehingga di didapatkan jumlah nilai dari x^2 yaitu 195, y^2 sejumlah 145.9 dan jumlah nilai xy yaitu 166.6. Kemudian perhitungan PCC dapat dilihat di bagian akhir Tabel 13, nilai PCC pada program yang menggunakan data 100 konten yaitu 0.988438106.

Tabel 14. PCC pada 150 konten

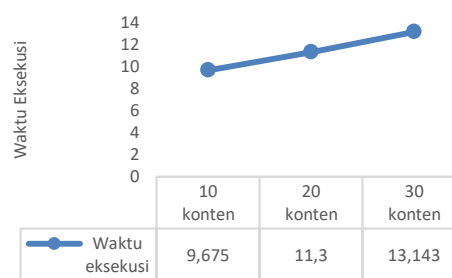
No	Wisata	Google Maps		Sistem		
		x	x ²	y	y ²	xy
1	Pantai Ngudel	4.5	20.3	3.99	15.92	17.96
2	Pantai Goa Cina	4.4	19.4	3.81	14.54	16.78
3	Coban Rondo	4.4	19.4	3.38	11.42	14.87
4	Coban Rais	4.1	16.8	4.14	17.14	16.97
5	Batu Night Spectacular	4.4	19.4	2.74	7.508	12.06
6	Museum Angkut	4.6	21.2	3.95	15.6	18.17
7	Paralayang	4.6	21.2	3.44	11.83	15.82
8	Jawa Timur Park 3	4.4	19.4	4.32	18.66	19.01
9	Selecta	4.4	19.4	4.41	19.45	19.4
10	Omah Kayu	4.3	18.5	4.32	18.66	18.58
		Σ		195	150.7	169.6
		$PCC = \frac{\Sigma xy}{\sqrt{\Sigma x^2 \Sigma y^2}}$		0.990156941		

Pada Tabel 14, tempat wisata Pantai Ngudel memiliki nilai x dan x^2 yang sama dengan Tabel 12. Hasil nilai *rating* tempat wisata tersebut dari sistem aplikasi adalah 3.99, nilai ini adalah nilai y . Serta dilakukan perkalian kuadrat dari y pada kolom y^2 , yang bernilai 15.92. Setelah mendapat nilai x dan y , maka selanjutnya

adalah menentukan perkalian dari keduanya di kolom xy , yaitu bernilai 17.96. Perhitungan ini dilakukan pada seluruh tempat wisata, sehingga di didapatkan jumlah nilai dari x^2 yaitu 195, y^2 sejumlah 150.7 dan jumlah nilai xy yaitu 169.6. Kemudian perhitungan PCC dapat dilihat di bagian akhir Tabel 14, nilai PCC pada program yang menggunakan data 150 konten yaitu 0.990156941.

3.3.3. Pengujian Waktu

Uji performa dilakukan untuk melihat seberapa handal *library* yang dibuat untuk memproses sebuah inputan menjadi output yang diinginkan. Hasil dari pengujian ini digambarkan dalam bentuk grafik.



Gambar 10. Grafik Uji Performa Waktu

Dari Gambar 10 dapat dilihat bahwa waktu eksekusi untuk setiap variasi jumlah konten yang berbeda waktunya juga berbeda. Semakin banyak konten yang diproses maka waktu eksekusi juga semakin lama. Jika digambarkan dalam bentuk grafik dapat dilihat bahwa *library* yang dibuat memiliki performa linear.

4. Kesimpulan

Berdasarkan hasil analisis, perancangan, implementasi dan pengujian yang dilakukan, diperoleh kesimpulan bahwa *library* yang dibuat sudah memiliki fungsi yang sama dengan Sistem Pemeringkat Otomatis Berbasis Kata Sifat dengan nilai presentase *error* sebesar 14.333% dan PCC sebesar 0.990156941 jika dibandingkan dengan nilai *rating* dari Google Maps, maka dapat disimpulkan bahwa *library* yang dibuat memiliki performa yang baik.

Penelitian ini dapat dikembangkan lebih lanjut menuju beberapa arah. Antara lain semacam pembuatan *library* dengan Bahasa pemrograman lain atau mengoptimalkan sumber data relasi kata sifat Bahasa Indonesia.

Referensi

- Bahasa, B. P. dan P. (2016). *Kamus Besar Bahasa Indonesia*. Kementerian Pendidikan dan Kebudayaan Republik Indonesia. Retrieved from <https://kbbi.kemdikbud.go.id/>
- Pressman, R. (2010). *Software Engineering: A Practitioner's Approach* (7th ed.). New York, NY, USA: McGraw-Hill, Inc.
- Putra, M. E. R., Rahutomo, F., & Pramitarini, Y. (2019a). Library Sistem Peringkat Otomatis Berbasis Kata Sifat. Retrieved from <https://github.com/ElfanRodh/PySpoks>
- Putra, M. E. R., Rahutomo, F., & Pramitarini, Y. (2019). Sistem Peringkat Otomatis Berbasis Kata Sifat. Retrieved from <https://pypi.org/project/PySpoks/>
- Python Software Foundation. (2019). Python 3.7.4 documentation. Retrieved from <https://docs.python.org/3/>
- Rahutomo, F., Hidayati, I. P., Saputra, P. Y., Rozi, I. F., Nurhasan, U., & Ramadhani, D. M. (2018). Automatic Rating System for National Delivery Services. In *2018 International Conference on Applied Science and Technology (iCAST)* (pp. 226–230). <https://doi.org/10.1109/iCAST1.2018.8751575>
- Rahutomo, Faisal, Adi, Z. H. R., Rozi, I. F., & Saputra, P. Y. (2018). Implementasi Text Mining Pada Website/Blog Di Internet Untuk Menilai Kinerja Suatu Organisasi. *Inovtek Polbeng Seri Informatika*, 3(2), 101–109. <https://doi.org/https://doi.org/10.35314/isi.v3i2.462>
- Ramadhani, D. M., Rahmad, C., & Rahutomo, F. (2017). Tourism destination rating system based on social media analysis (proposal and dataset development in Indonesian language). In *2017 International Conference on Sustainable Information Engineering and Technology (SIET)* (pp. 41–46). <https://doi.org/10.1109/SIET.2017.8304106>