

CONTEXT-AWARE UNTUK NATURAL LANGUAGE PROCESSING SERVICES MENGUNAKAN ARSITEKTUR MICROSERVICES

Abdullah Aziz Sembada, DThomas Hatta Fudholi, Raden Teduh Dirgahayu

Fakultas Teknologi Industri, Universitas Islam Indonesia

Email: 18917102@students.uui.ac.id, hatta.fudholi@uui.ac.id, teduh.dirgahayu@uui.ac.id

Abstrak

Berkembangnya era industri 4.0 dan big data membuat Natural Language Processing banyak dibutuhkan terutama saat melakukan preprocessing data. Dengan adanya Natural Language Processing service agar bisa mempermudah peneliti dalam melakukan penelitian karena beberapa kebutuhannya sudah disediakan. Sebelum menyediakan service-service Natural Language Processing terlebih dahulu melakukan perancangan sistem. Sistem di buat menggunakan microservice architecture, microservice dipilih karena memiliki karakteristik flexible, aman, error terisolasi sangat memudahkan dalam melakukan pengembangan sistem. Dalam sistem ini ditambahkan fitur Context-Aware untuk memudahkan pengguna dalam mengolah data. Tujuan dari penelitian adalah mampu mengintegrasikan Context-Aware ke dalam sistem Natural Language Processing service sehingga sistem mampu memberikan rekomendasi algoritma yang paling tepat dari data yang dimiliki pengguna. Hasil pengujian sistem menunjukkan bahwa Natural Language Processing service dapat mempersingkat penelitian tentang natural language processing. Hasil tersebut tidak lepas dari fitur context-aware yang dapat memnentukan jenis file atau data yang di-input pengguna, dengan demikian pengguna langsung diarahkan oleh sistem untuk memproses file atau data tersebut dengan algoritma clustering atau classification. Implementasi microservices juga sangat membantu dalam pengembangan terutama penambah service atau algoritma tidak akan mengganggu service yang sudah ada.

Kata Kunci: Microservice; Natural Language Processing; Context-Aware.

Abstract

The development of the industrial era 4.0 and big data makes Natural Language Processing much needed, especially when preprocessing data. With the Natural Language Processing service, it can make it easier for researchers to do research because some of their needs have been provided. Before providing Natural Language Processing services, first do a system design. The system was created using microservice architecture, microservice was chosen because it has flexible, safe, error isolated characteristics that make it very easy to develop the system. In this system, Context-Aware features are added to facilitate users in processing data. The purpose of the research is to be able to integrate Context-Aware into the Natural Language Processing service system so that the system is able to provide recommendations for the most appropriate algorithm from the data owned by the user. The results of system testing show that the Natural Language Processing service can shorten research on natural language processing. This result cannot be separated from the context-aware feature that can determine the type of file or data inputted by the user, thus the user is directly directed by the system to process the file or data with a clustering or classification algorithm. The implementation of microservices is also very helpful in development, especially adding services or algorithms will not interfere with existing services.

Keywords: Microservice; Natural Language Processing; Context-Aware.

Pendahuluan

Natural Language Processing adalah cabang dari ilmu AI untuk membuat komputer mengerti pengolahan bahasa alami (Trofin et al., 2019) . Natural Language Processing semakin dibutuhkan seiring dengan berkembangnya data teks setiap harinya . Berkembangnya data teks membuat penelitian dengan metode Natural Language Processing semakin dibutuhkan, namun untuk melakukan penelitian menggunakan Natural Language Processing harus bisa programming, tentu ini menjadi hambatan tersendiri terutama bagi yang tidak memiliki latar belakang teknologi informasi.

Dalam hal ini banyak perusahaan sudah menyediakan layanan Natural Language Processing service. Seperti CloudFactory dan prosa.ai, namun layanan dari yang ditawarkan pengguna tidak mengetahui algoritma yang digunakan atau tidak bisa mengubah algoritma yang digunakan. Karena layanan yang digunakan berbasis B2B (business to business) sehingga kurang mendukung untuk para peneliti.

Banyaknya metode Natural Language Processing juga menghambat peneliti untuk memilih metode apa yang akan digunakan. Untuk mengatasi masalah tersebut maka ditambahkan fitur Context-Aware. Context-Aware didefinisikan sebagai informasi yang bisa dirasakan aplikasi dari lingkungan yang dapat digunakan untuk menggambarkan situasi (Biegel & Cahill, 2004) . Dengan Context-Aware sistem mampu membaca file yang di-input oleh pengguna dan bisa mengerti jenis data dari file tersebut apakah termasuk categorical atau uncategorical. Jika uncategorical akan di-training dengan algoritma clustering dan jika categorical akan di-training dengan algoritma klasifikasi, kemudian sistem akan merekomendasikan algoritma dengan akurasi tertinggi. Sistem hanya akan merekomendasikan algoritma - algoritma yang ada pada Natural language processing services.

Microservice adalah gaya arsitektur yang terdiri dari serangkaian aplikasi kecil yang independen, serta masing-masing aplikasi berjalan dalam satu VM dan interaksi antar aplikasi menggunakan HTTP APIs atau asynchronous messaging (Merson & Yoder, 2020) . Microservice sekarang semakin dikenal sebagai gaya arsitektur yang menjanjikan untuk membangun sistem yang kompleks atau bersekala besar (Bao et al., 2019) . Keuntungan menggunakan arsitektur microservice adalah flexible dan mudah dikembangkan (Krämer et al., 2019) , serta sifat yang independen memudahkan untuk melakukan pengembangan sistem (Kargar & Hanifzade, 2018) . Selain microservice ada arsitektur Monolithic, tetapi Aplikasi yang dibuat menggunakan Monolithic memiliki kelemahan ketika basis kode aplikasi tumbuh besar dan perubahan harus dilakukan dengan cepat. Penskalaan kecil juga tidak mungkin dengan Monolithic, karena seluruh aplikasi perlu digunakan setiap saat (Kalske et al., 2017) .

Dari penelitian (Z. Yu et al., 2019) ,(Z. Yu et al., 2019) ,(Hasselbring & Steinacker, 2017) disimpulkan bahwa arsitektur *microservice* sangat memudahkan dalam melakukan pengembangan sistem, pada penelitian (Roca et al., 2020) *microservice* sudah diimplementasikan pada *Natural language processing*. Maka pada penelitian ini akan melakukan implementasi *microservice* untuk sistem *Natural language processing services* diharapkan dapat memudahkan pengembangan sistem serta ditambahkan fitur juga Context-Aware untuk memberikan rekomendasi pengolahan data menggunakan algoritma clustering atau classification.

Kontribusi pada penelitian ini adalah implementasi Context-Aware dan *microservice* untuk Natural Language Processing Services. Untuk bisa menjelaskan kontribusi di atas dan

posisi dari rencana penelitian ini, berikut ini di paparkan state of the art penelitian sebelumnya mengenai Natural Language Processing Services, Context-Aware, Microservices.

Pertama penelitian - penelitian tentang Natural Language Processing Services. Mengusulkan Natural Language Processing untuk sistem yang mengklasifikasikan opini tentang produk dan layanan dari perspektif dan twitt konsumen untuk rekomendasi, sistem ini juga berguna untuk semua produk dan layanan sehingga memudahkan pelanggan. (Eryigit, 2014) Mengembangkan ITU Turkish NLP Web Service, sistem yang menyediakan alat penelitian yang canggih, disebutkan juga karena memberikan kemudahan dalam melakukan pemrosesan bahasa alami, sistem ini mampu menarik banyak pengguna dari berbagai Universitas di Turki.

Kedua penelitian - penelitian tentang Context-Aware. (Kousiouris et al., 2019) Mengembangkan sistem untuk mendeteksi percakapan penting pada telepon seluler, disebutkan juga sistem mampu menunjukkan informasi tentang konteks percakapan dengan menggunakan pendekatan berbasis konten murni. (Voita et al., 2018) Mengembangkan sistem neural machine translation (NMT), merupakan mesin penerjemah yang dilengkapi Context-Aware sehingga NMT mampu mendeteksi kata ganti yang ambigu.

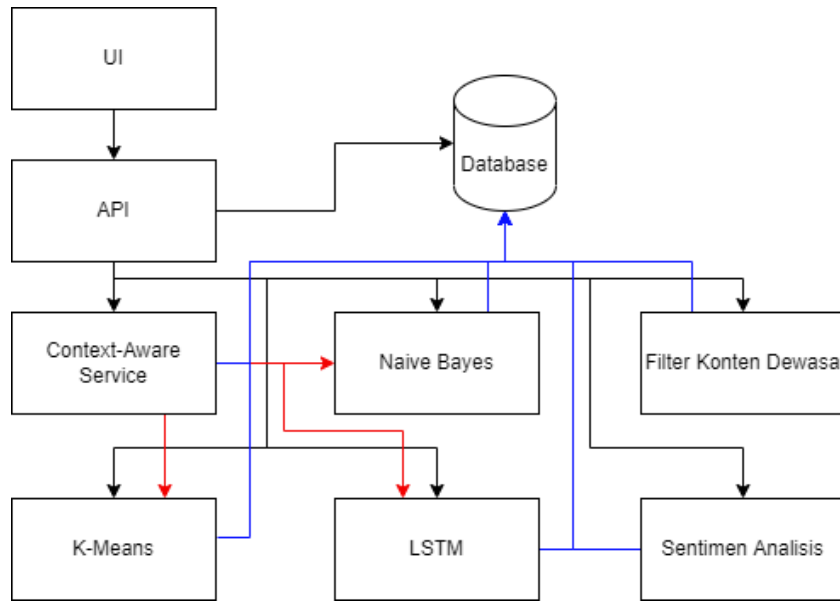
Terakhir penelitian - penelitian tentang Microservice. (Krämer et al., 2019) Memilih microservice sebagai arsitektur dari sistem smart cities, agar sistem menjadi flexible sehingga mudah untuk menambah service ataupun mengganti service yang sudah usang. Dilaporkan juga menggunakan layanan cloud yang memberikan keuntungan skalabilitas dan efektivitas biaya. Microservice diterapkan untuk membangun sistem Internet of Things (IoT) / Mobile, seperti yang dilaporkan (Kousiouris et al., 2019) dan juga (Wang et al., 2020) . Dari paparan tersebut disimpulkan bahwa belum ada penelitian yang melakukan implementasi Context-Aware dan microservice dengan studi kasus Natural Language Processing (NLP) Services. Untuk menangani komunikasi antar service kami menggunakan RESTful API seperti yang sudah dilakukan (Gossett et al., 2018) dan (Liu et al., 2017) .

Metode Penelitian

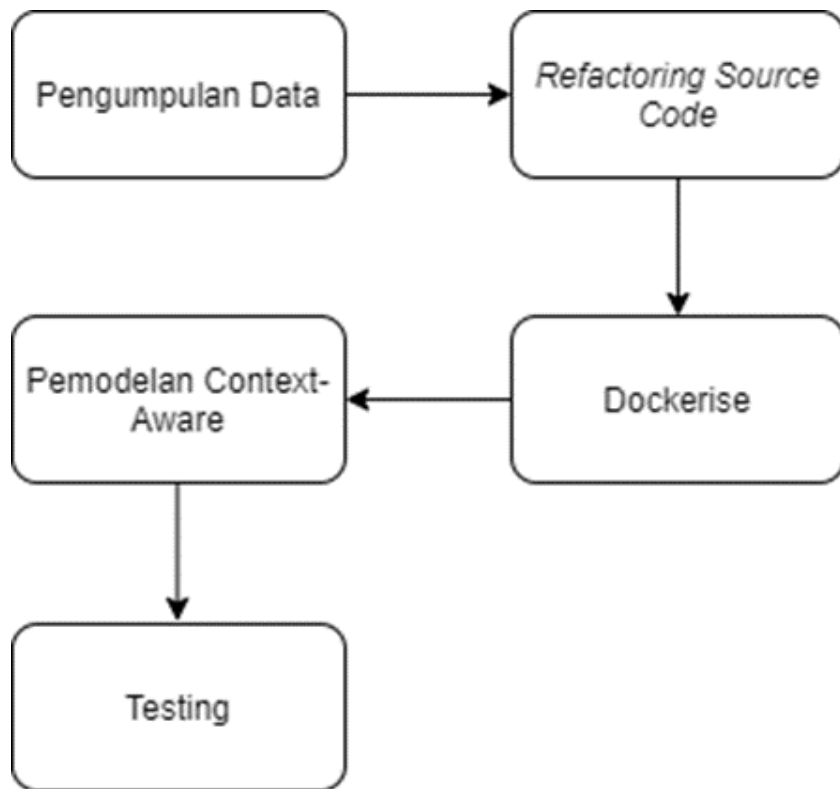
Natural Language Processing Services dibangun menggunakan microservices dengan arsitektur sistem seperti gambar 1. Microservices dijalankan menggunakan docker, docker direkomendasikan karena dapat memfasilitasi manajemen sistem untuk microservices (Ma et al., 2018) . Natural Language Processing Services digunakan untuk melakukan pemrosesan data teks bisa diakses melalui UI ataupun API.

Pengembangan Natural Language Processing Services dilakukan dengan lima langkah. Langkah-langkah yang dilakukan adalah pengumpulan data, refactoring source code, Dockerize, pemodelan context-aware, serta pengembangan dan evaluasi.

Langkah – langkah pengembangan sistem bisa dilihat pada Gambar 2.



Gambar 1. Arsitektur Sistem *Natural Language Processing Services*



Gambar 2. Langkah – langkah penelitian

2.1. Pengumpulan Data

Source code yang digunakan untuk mengembangkan sistem Natural Language Processing Services adalah source code tentang Natural Language Processing yang ditulis menggunakan Bahasa pemrograman Python. Python dipilih karena merupakan solusi

pemrograman lengkap, bahasa yang mudah dipelajari, fleksibilitasnya juga membuat Python menjadi bahasa yang cepat berkembang dan banyak tersedia library. Terutama library data analytics yang membuat Python merupakan bahasa yang direkomendasikan digunakan untuk analisis data (Nagpal & Gabrani, 2019) . Python

juga lebih sederhana jika dibandingkan bahasa pemrograman lain seperti Ruby, Perl, Java, C++, dll (Prashant et al., 2019) .

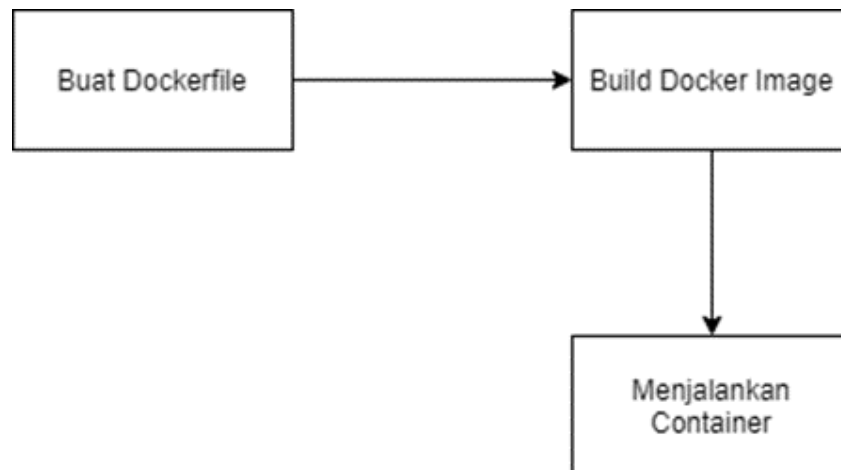
2.2. Refactoring Source Code

Refactoring diperlukan untuk menyesuaikan source code dengan Natural Language Processing Services sehingga source code bisa diintegrasikan. Refactoring dilakukan dengan mengambil kode algoritmanya, serta melakukan penyesuaian input dan output. Kemudian kode algoritma tersebut dipasang pada controller Django Rest Framework. Django dipilih karena terbukti powerful dalam penggunaan kembali kode dan ekstensibilitas fungsional dan perkembangan yang cepat serta Django juga bisa

meningkatkan digitalisasi dan datasharing (Q. Yu & Yang, 2019) .

2.3. Dockerize

Dockerize diperlukan supaya source code bisa dijalankan sebagai docker container, dockerize dilakukan dengan tiga langkah seperti pada Gambar 3 yang pertama adalah pembuatan Dockerfile, Dockerfile berfungsi untuk mendefinisikan root dari sistem file image kontainer, berbagai informasi seperti environment variables, perubahan yang akan dilakukan pada image base, penambahan beberapa aplikasi dan port serta menentukan perintah yang harus dijalankan setelah kontainer dimulai (Santoro et al., 2018) . kedua build docker image dan ketiga menjalankan container.



Gambar 3. Langkah – langkah Dockerize

2.4. Pemodelan *context-aware*

Untuk memberikan rekomendasi analisis, sistem membaca file excel atau csv yang diunggah pengguna. Jika jumlah kolom pada file lebih dari dua, maka kolom dengan jumlah data unik paling sedikit akan digunakan sebagai label dan kolom yang mempunyai jumlah unik paling banyak akan digunakan sebagai fitur. Jika jumlah data unik pada fitur kurang atau sama dengan tiga (≤ 3) maka file tersebut dikategorikan categorical dan jika lebih dari tiga maka dikategorikan uncategorical. Kondisi ini

dipilih karena semakin sedikit data unik pada fitur semakin mudah menentukan kategori pada setiap baris datanya. Jika file input seperti Gambar 4 maka sistem akan mengenali file tersebut sebagai file categorical karena jumlah data unik pada kolom fitur adalah tiga. Setelah kategori file diketahui, sistem memberikan rekomendasi algoritma yang sesuai dengan kategorinya Misalnya jika kategori categorical sistem merekomendasikan untuk menggunakan algoritma naïve bayes.

country	child_mor	exports	health	imports	income	inflation	life_exp	total_fer	gdp
Afghanistan	30.2	10	7.38	44.7	3000	22.4	60.1	2	353
Albania	16.6	28	6.55	48.1	19100	1.44	76.8	1	4090
Algeria	27.3	38.4	4.17	31.1	19100	1.44	76.8	2	4460
Angola	119	62.3	2.85	42.9	3000	22.4	60.1	6	3530
Antigua and Barb	10.3	45.5	6.03	58.9	19100	1.44	76.8	2	12200
Argentina	3.1	16	18700	20.9	75.8	2	10300		
Armenia	4.4	45.3	73.8	1	3220				
Australia	7.3	20.9	82	1	51900				
Austria	4.3	51.3	11	47.8	80.6	1	46900		
Azerbaijan	39.2	54.3	5.88	20.7	16000	13.8	69.1	1	5840
Bahamas	13.8	35	7.89	43.7	22900	-0.393	73.8	1	28000
Bahrain	8.6	69.5	4.97	50.9	41100	7.44	76.8	2	20700
Bangladesh	49.4	16	3.52	21.8	2440	7.14	70.8	2	758
Barbados	14.2	39.5	7.97	48.7	15300	0.321	76.7	1	16000
Belarus	5.5	51.4	5.61	64.5	16200	15.1	70.8	1	6030
Belgium	4.5	76.4	10.7	74.7	41100	1.88	80.1	1	44400
Belize	18.8	58.2	5.2	57.5	7880	1.14	71.8	2	4340
Bhutan	111	23.8	4.1	37.2	1820	0.885	61.8	2	758
Bhutan	42.7	42.5	5.2	70.7	6420	5.99	72.1	2	2180

Gambar 4. File input

2.5. Testing

Pengujian akan dilakukan secara manual pada *context-aware service* yaitu dengan cara melakukan input file, kemudian hasil dari *context-aware service* dicocokkan dengan kategori file yang sudah dikategorikan secara manual sebelumnya.

Hasil Penelitian dan Pembahasan

3.1. Pengumpulan Data

Penelitian ini menggunakan source code yang diperoleh dengan mengumpulkan hasil penelitian tentang Natural Language Processing dan source code yang diunduh dari github. Tabel 1 adalah aplikasi dan algoritma yang sudah dipasang pada Natural Language Processing Services adalah sebagai berikut:

Tabel 1. Daftar Source Code

Nama	sumber
Adult content classification on Indonesian tweets using LSTM neural network	(Hidayatullah et al., 2019)
Sentimen Analisis	(Hidayatullah, 2019)
Text Preprocessing	(Sembada, 2020b)
Long Short-Term Memory	(Sembada, 2020a)
Naïve Bayes	(Hidayatullah, 2021)
K-Means	(Vanderplas, 2018)

Workflow dari projects Source code Adult content classification on Indonesian tweets using LSTM neural network dan Long Short-Term Memory adalah user memberikan input berupa text lewat frontend atau API dan

output dari backend adalah hasil analisis dari data input.

Source code Sentimen Analisis, Naïve Bayes dan K-Means menggunakan bahasa pemrograman python yang ditulis menggunakan google colab. Input dari source

code ini adalah file excel serta output-nya berupa teks.

Source code text preprocessing sudah menggunakan Django rest framework serta sudah bisa diakses melalui API. Input dan output berupa JSON.

3.2. Refactoring Source Code

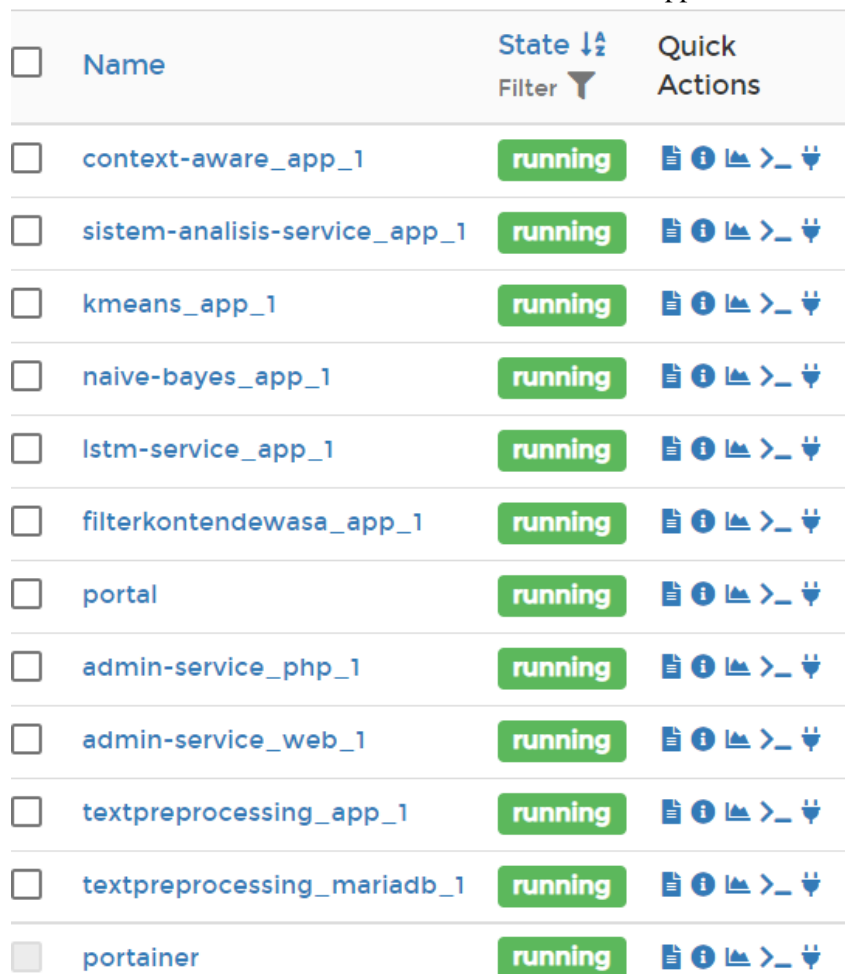
Refactoring yang dilakukan dari source code yang didapat berbeda-beda untuk Adult content classification on Indonesian tweets using LSTM neural network diambil backend-nya saja serta dilakukan penyesuaian database serta input dan output.






























































Untuk source code Sentimen Analisis, Naïve Bayes dan K-Means dilakukan implementasi code ke Django rest framework yang juga dilakukan dilakukan penyesuaian database serta input dan output.

Serta untuk Source code text preprocessing dan Long Short-Term Memory sudah siap diimplementasikan pada natural language processing services karena kedua source code ini dibuat menggunakan Django rest framework. Recode yang dilakukan pada bagian setting koneksi database.

3.3. Dockerize

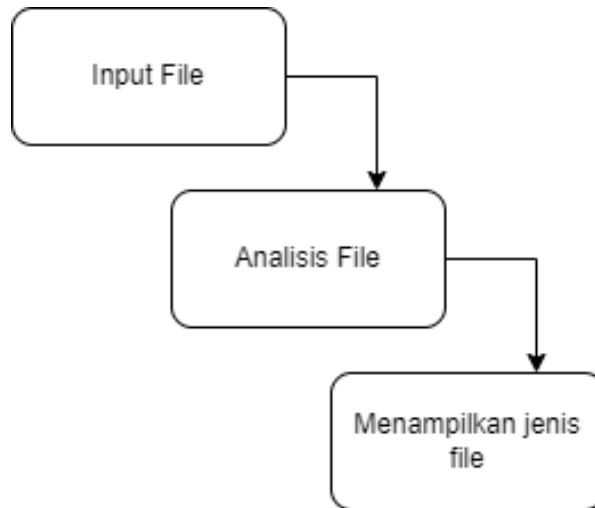
Seperti pada Gambar 3 ada 3 langkah-langkah yang harus dilakukan untuk melakukan dockerize adalah membuat dockerfile, build docker image dan menjalankan container. Gambar 5 adalah hasil dari proses dockerize terdapat 12 container. Dari 12 container yang digunakan untuk service context-aware ada 5 container yaitu textpreprocessing_app, lstm-service_app, naïve-bayes_app, k-means_app dan context-awarw_app.



<input type="checkbox"/>	Name	State 	Quick Actions
<input type="checkbox"/>	context-aware_app_1	running	    
<input type="checkbox"/>	sistem-analisis-service_app_1	running	    
<input type="checkbox"/>	kmeans_app_1	running	    
<input type="checkbox"/>	naive-bayes_app_1	running	    
<input type="checkbox"/>	lstm-service_app_1	running	    
<input type="checkbox"/>	filterkontendewasa_app_1	running	    
<input type="checkbox"/>	portal	running	    
<input type="checkbox"/>	admin-service_php_1	running	    
<input type="checkbox"/>	admin-service_web_1	running	    
<input type="checkbox"/>	textpreprocessing_app_1	running	    
<input type="checkbox"/>	textpreprocessing_mariadb_1	running	    
<input type="checkbox"/>	portainer	running	    

Gambar 5. Daftar Container

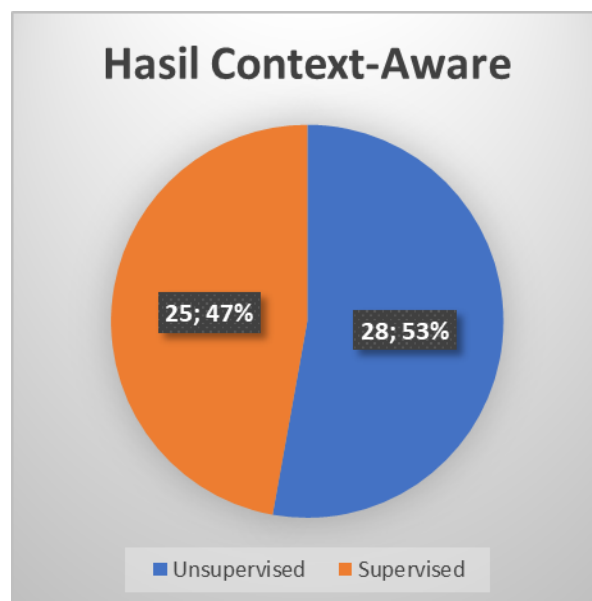
3.4. Pemodelan context-aware



Gambar 6. Proses context-Aware

Gambar 6 merupakan alur proses context-aware yang terdiri 3 tahap. Pertama pengguna mengunggah file csv/xlsx, kemudian file dirubah mejadi JSON dan dikirim ke service context-aware. Kedua service menentukan jenis file tersebut termasuk categorical atau uncategorical. Ketiga service menampilkan jenis file apakah categorical dan uncategorical.

service context-aware dapat mengenali file apakah supervised atau unsupervised adalah dengan menghitung keunikan data pada setiap kolom file tersebut. Dari gambar 7 bisa dilihat hasil dari 53 dataset yang diuji sistem dapat mendeteksi 28 file unsupervised dan 25 file supervised.

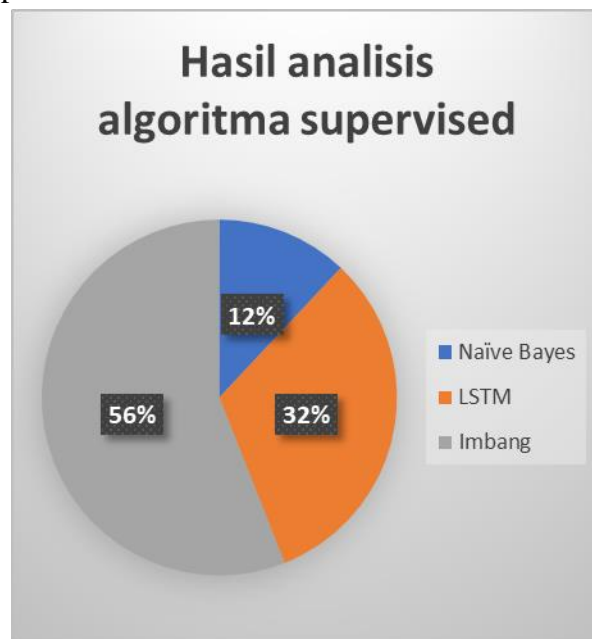


Gambar 7. Hasil context-Aware

Karena algoritma unsupervised tidak bisa dibandingkan akurasi, sehingga service context-aware hanya menampilkan rekomendasi algoritma unsupervised. Sedangkan untuk algoritma supervised, service menghitung akurasi dari masing-masing algoritma untuk dibandingkan tingkat akurasi.

Gambar 8 merupakan hasil dari analisis algoritma-algoritma supervised dari 25 file

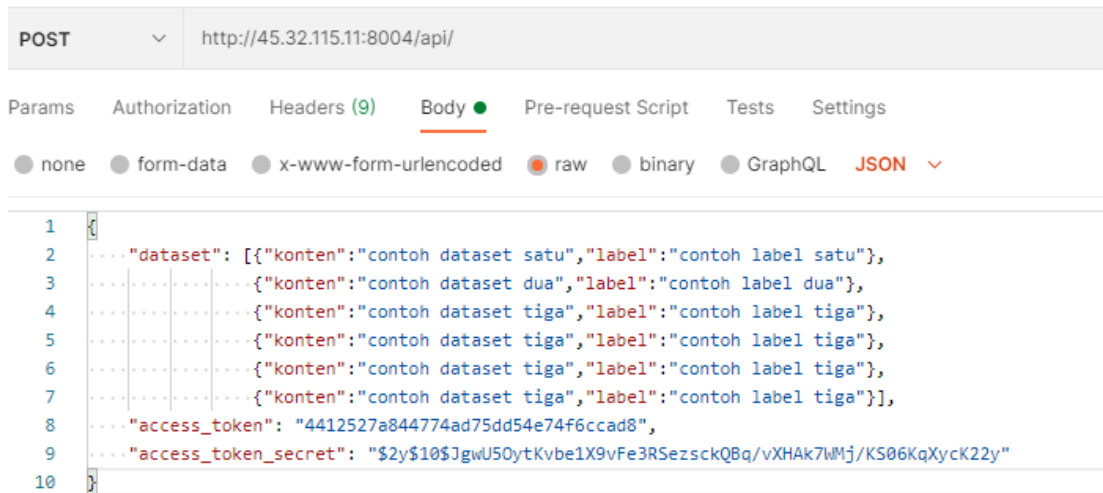
dataset yang dikategorikan file supervised hanya ada 3 file yang tingkat akurasi lebih besar Naïve Bayes sedangkan untuk LSTM ada 8 file yang tingkat akurasi lebih tinggi dari Naïve Bayes dan 14 file sisanya keduanya mempunyai tingkat akurasi yang sama.



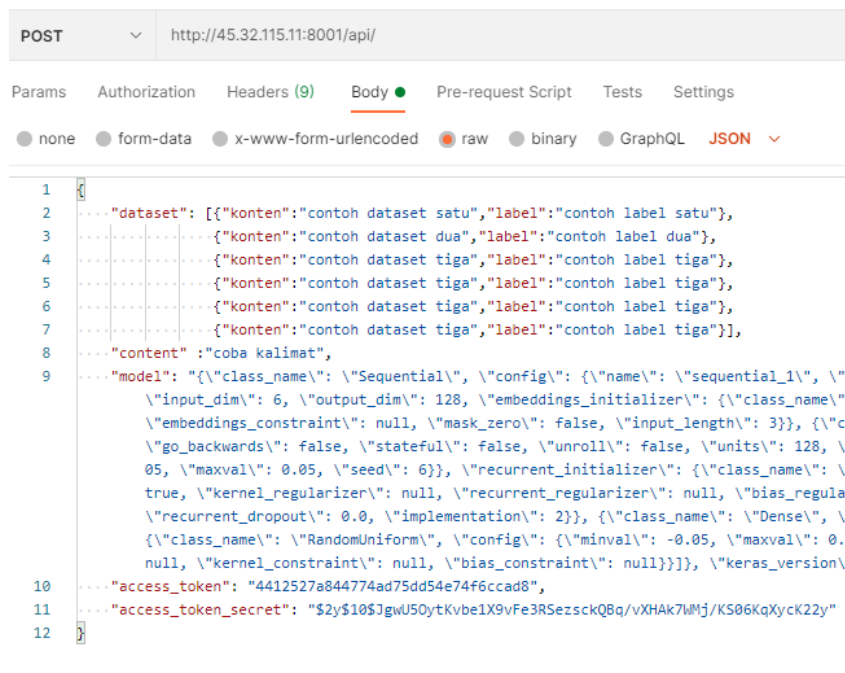
Gambar 8. Hasil Analisis Algoritma Supervised

Natural Language Processing Services bisa diakses menggunakan API dengan contoh penggunaan seperti pada gambar 9 format JSON untuk mengakses API naïve bayes, gambar 10 format JSON untuk mengakses API LSTM, gambar 11 format JSON untuk mengakses API filter konten dewasa dan gambar 12 format JSON untuk mengakses API Text Preprocessing.

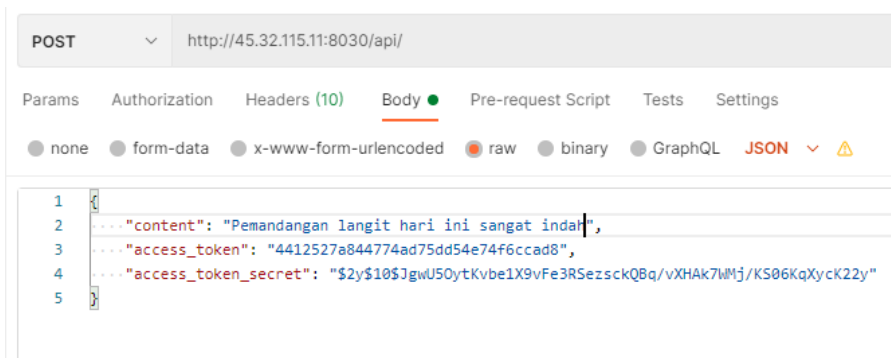
Akses API diharapkan bisa membantu untuk mengakses Natural Language Processing Services dari notebook atau aplikasinya sendiri. Untuk dapat mengakses API pengguna harus mendaftar terlebih dahulu ke Natural Language Processing Services kemudian membuat token dan setiap pengguna akan mendapatkan `access_token` dan `access_token_secret`.



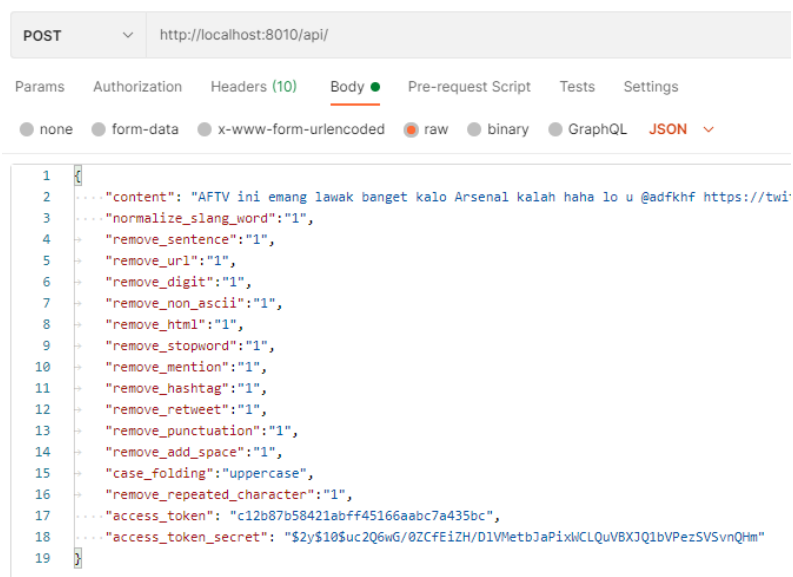
Gambar 9. Akses API Service Naïve Bayes



Gambar 10. Akses API Service LSTM



Gambar 11. Akses API Service Filter Konten Dewasa



Gambar 12. Akses API Service Text Preprocessing

Simpulan

Penelitian ini dapat mempersingkat penelitian tentang natural language processing terutama pada penelitian data teks. Karena sistem mampu menunjukan algoritma yang tepat untuk digunakan pada file yang diujikan ke sistem.

Implementasi microservices juga sangat membantu dalam pengembangan terutama penambah service atau algoritma tidak akan mengganggu service yang sudah ada. Dengan ini juga sangat dimungkinkan menggunakan algoritma dengan bahasa pemrograman yang berbeda karena dengan microservices memungkinkan aplikasi dengan bahasa pemrograman yang berbeda bisa saling berkomunikasi satu sama lain.

Service context-aware dapat mengenali file dan mengkatagorikanya menjadi supervised atau unsupervised seperti pada gambar 7. Dari hasil pada gambar 8 bisa disimpulkan secara keseluruhan LSTM ataupun Naïve bayes memiliki akurasi yang sama. Namun pada pengujian yang hasilnya berbeda LSTM lebih unggul dibandingkan naïve bayes.

Saran untuk penelitian selanjutnya adalah melakukan penambahan aloritma serta

melakukan komparasi algoritma yang sama dengan bahasa yang berbeda.

Saran

Saran merupakan usulan atas perbaikan dari penyelesaian masalah yang dihadapi dalam penelitian ini maupun ide usulan penelitian lanjutan atau revisiting penelitian yang sama dengan sudut pandang yang berbeda

Daftar Pustaka

- Bao, L., Wu, C., Bu, X., Ren, N., & Shen, M. (2019). Performance Modeling And Workflow Scheduling Of Microservice-Based Applications In Clouds. *Ieee Transactions On Parallel And Distributed Systems*, 30(9), 2114–2129. <https://doi.org/10.1109/TPDS.2019.2901467>
- Biegel, G., & Cahill, V. (2004). A Framework For Developing Mobile, Context-Aware Applications. 361–365. <https://doi.org/10.1109/Percom.2004.1276875>
- Eryiğit, G. (2014). Itu Turkish Nlp Web Service. *Proceedings Of The Demonstrations At The 14th*

- Conference Of The European Chapter Of The Association For Computational Linguistics*, 1–4.
- Gossett, E., Toher, C., Oses, C., Isayev, O., Legrain, F., Rose, F., Zurek, E., Carrete, J., Mingo, N., & Tropsha, A. (2018). Aflow-Ml: A Restful Api For Machine-Learning Predictions Of Materials Properties. *Computational Materials Science*, 152, 134–145. <https://doi.org/10.1016/j.commatsci.2018.03.075>
- Hasselbring, W., & Steinacker, G. (2017). Microservice Architectures For Scalability, Agility And Reliability In E-Commerce. *2017 Ieee International Conference On Software Architecture Workshops (Icsaw)*, 243–246. <https://doi.org/10.1109/Icsaw.2017.11>
- Kalske, M., Mäkitalo, N., & Mikkonen, T. (2017). Challenges When Moving From Monolith To Microservice Architecture. *International Conference On Web Engineering*, 32–47.
- Kargar, M. J., & Hanifzade, A. (2018). Automation Of Regression Test In Microservice Architecture. *2018 4th International Conference On Web Research (Icwr)*, 133–137. <https://doi.org/10.1109/Icwr.2018.8387249>
- Kousiouris, G., Tsarsitalidis, S., Psomakelis, E., Koloniaris, S., Bardaki, C., Tserpes, K., Nikolaidou, M., & Anagnostopoulos, D. (2019). A Microservice-Based Framework For Integrating Iot Management Platforms, Semantic And Ai Services For Supply Chain Management. *Ict Express*, 5(2), 141–145. <https://doi.org/10.1016/j.icte.2019.04.002>
- Krämer, M., Frese, S., & Kuijper, A. (2019). Implementing Secure Applications In Smart City Clouds Using Microservices. *Future Generation Computer Systems*, 99, 308–320. <https://doi.org/10.1016/j.future.2019.04.042>
- Liu, S., Li, Y., Sun, G., Fan, B., & Deng, S. (2017). Hierarchical Rnn Networks For Structured Semantic Web Api Model Learning And Extraction. *2017 Ieee International Conference On Web Services (Icws)*, 708–713. <https://doi.org/10.1109/Icws.2017.85>
- Ma, S.-P., Fan, C.-Y., Chuang, Y., Lee, W.-T., Lee, S.-J., & Hsueh, N.-L. (2018). Using Service Dependency Graph To Analyze And Test Microservices. *2018 Ieee 42nd Annual Computer Software And Applications Conference (Compsac)*, 2, 81–86. <https://doi.org/10.1109/Compsac.2018.10207>
- Merson, P., & Yoder, J. (2020). Modeling Microservices With Ddd. *2020 Ieee International Conference On Software Architecture Companion (Icsa-C)*, 7–8. <https://doi.org/10.1109/Icsa-C50368.2020.00010>
- Nagpal, A., & Gabrani, G. (2019). Python For Data Analytics, Scientific And Technical Applications. *2019 Amity International Conference On Artificial Intelligence (Aicai)*, 140–145. <https://doi.org/10.1109/Aicai.2019.8701341>
- Prashant, P., Tickoo, A., Sharma, S., & Jamil, J. (2019). Optimization Of Cost To Calculate The Release Time In Software Reliability Using Python. *2019 9th International Conference On Cloud Computing, Data Science & Engineering (Confluence)*, 471–474. <https://doi.org/10.1109/Confluence.2019.8776620>
- Roca, S., Sancho, J., García, J., & Alesanco,

- Á. (2020). Microservice Chatbot Architecture For Chronic Patient Support. *Journal Of Biomedical Informatics*, 102, 103305. <https://doi.org/10.1016/j.jbi.2019.103305>
- Santoro, C., Messina, F., D'urso, F., & Santoro, F. F. (2018). Wale: A Dockerfile-Based Approach To Deduplicate Shared Libraries In Docker Containers. *2018 Ieee 16th Intl Conf On Dependable, Autonomic And Secure Computing, 16th Intl Conf On Pervasive Intelligence And Computing, 4th Intl Conf On Big Data Intelligence And Computing And Cyber Science And Technology Congress (Dasc/Picom/Datacom/Cybercitech)*, 785–791. <https://doi.org/10.1109/Dasc/Picom/Datacom/Cybercitech.2018.00135>
- Trofin, R. S., Chiru, C., Vizitiu, C., Dinculescu, A., Vizitiu, R., & Nistorescu, A. (2019). Detection Of Astronauts' Speech And Language Disorder Signs During Space Missions Using Natural Language Processing Techniques. *2019 E-Health And Bioengineering Conference (Ehb)*, 1–4. <https://doi.org/10.1109/Ehb47216.2019.8969950>
- Voita, E., Serdyukov, P., Sennrich, R., & Titov, I. (2018). Context-Aware Neural Machine Translation Learns Anaphora Resolution. *Arxiv Preprint Arxiv:1805.10163*. <https://doi.org/10.48550/Arxiv.1805.10163>
- Wang, R., Imran, M., & Saleem, K. (2020). A Microservice Recommendation Mechanism Based On Mobile Architecture. *Journal Of Network And Computer Applications*, 152, 102510. <https://doi.org/10.1016/j.jnca.2019.102510>
- Yu, Q., & Yang, W. (2019). The Analysis And Design Of System Of Experimental Consumables Based On Django And Qr Code. *2019 2nd International Conference On Safety Produce Informatization (Iicspi)*, 137–141. <https://doi.org/10.1109/Iicspi48186.2019.9095914>
- Yu, Z., Han, J., Zhao, T., Tian, N., & Wang, J. (2019). Research And Implementation Of Online Judgment System Based On Micro Service. *2019 Ieee 10th International Conference On Software Engineering And Service Science (Icsess)*, 475–478. <https://doi.org/10.1109/Icsess47205.2019.9040684>