

Pencocokan Menu Berbasis Keywords pada Chatbot dengan Metode Jaccard

I Gede Suarnata^{a1}, I Made Sukarsa^{a2}, Kadek Suar Wibawa^{b3}

^aProgram Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana, Bali
e-mail: 1gsuarnata99@gmail.com, 2sukarsa@unud.ac.id, 3suar_wibawa@unud.ac.id

Abstrak

Tantangan dari mengembangkan chatbot adalah mampu berkomunikasi secara interaktif dan memberikan respons yang akurat sesuai keinginan pengguna. Salah satu metode *pattern-matching* yang sederhana dan dapat diterapkan sebagai pencocokan menu pada chatbot adalah metode Jaccard. Penelitian ini bertujuan untuk mengetahui pengaruh Algoritma Jaccard terhadap pencocokan menu berbasis keywords pada chatbot. Penelitian ini membahas pengaruh proses pencocokan menu berbasis keywords dengan menggunakan metode Jaccard pada chatbot ketika mencocokkan pesan yang dikirim pengguna dengan menu yang tersedia pada basis data chatbot. Berdasarkan implementasi yang dilakukan pada PHP, penggunaan metode Jaccard mampu menangani beberapa kondisi ketika mencocokkan menu. Beberapa kondisi tersebut antara lain, pesan yang terdiri dari keseluruhan kata kunci, pesan yang terdiri dari sebagian kata kunci, maupun pesan yang terdiri dari beberapa kata kunci dengan posisi yang ditukar.

Kata kunci: Pencocokan Menu, Keywords, Chatbot, Jaccard, PHP

Abstract

The challenge in developing a chatbot is being able to communicate interactively while also delivering correct responses based on the user's wants. The Jaccard method is a simple pattern matching method that can be implemented as a menu matching in a chatbot. The goal of this study was to know what effect using the Jaccard algorithm that is applied in a chatbot which the matching menu process based on keywords. This study explains the process of matching a menu based on keywords using the Jaccard method on the chatbot when comparing messages given by users with the menu available in the chatbot database. Based on the chatbot implementation in PHP, the Jaccard method can handle several conditions while matching menu. Some of these conditions include messages consisting of all keywords, messages consisting of several keywords, and messages consisting of several keywords with changes positions.

Keywords: Menu Matching, Keywords, Chatbot, Jaccard, PHP

1. Introduction

Perkembangan teknologi yang cepat memberikan dampak yang banyak pada sektor komunikasi modern. Proses komunikasi yang awalnya dilakukan oleh sesama manusia sekarang dapat digantikan dengan sebuah teknologi yang bernama *chatbot*. Sistem *chatbot* difungsikan untuk dapat membalas otomatis pesan pengguna berdasarkan basis pengetahuan yang disediakan. Respons yang dihasilkan merupakan hasil pencocokan kata kunci yang paling sesuai dengan pesan yang dikirimkan oleh pengguna.

Tantangan dari perkembangan sistem *chatbot* adalah mampu berkomunikasi secara interaktif dan memberikan respons yang akurat sesuai keinginan pengguna. Hal tersebut menyebabkan terjadinya banyak perkembangan pada proses pembuatan *chatbot*. Proses pembuatan *chatbot* dapat dilakukan dengan berbagai cara. *Pattern* (pola) dan *template* (respons) merupakan skema umum dari pembuatan *chatbot*. Proses pencocokan *pattern* pada

chatbot dapat dilakukan dengan memanfaatkan *framework* AIML ataupun memanfaatkan *pattern-matching* [1]. Salah satu metode *pattern-matching* yang sederhana dan dapat diterapkan sebagai pencocokan menu pada *chatbot* adalah metode Jaccard.

Jaccard merupakan sebuah algoritma yang digunakan untuk membandingkan kemiripan ataupun perbedaan dari sebuah dokumen ataupun *dataset* berdasarkan kata yang dimiliki [2]. Penelitian sebelumnya menjelaskan bahwa pencocokan teks dengan Jaccard memberikan hasil yang baik dan lebih cocok diterapkan pada dokumen teks yang sangat pendek [3]. Penelitian lainnya membahas manfaat Jaccard sebagai *query suggestion* untuk membandingkan *keywords* dari pengguna dengan data pada basis data pengetahuan sistem dan memberikan hasil metode Jaccard akurat sebagai *query suggestion* sesuai dengan *keywords* yang dimasukkan oleh pengguna [2].

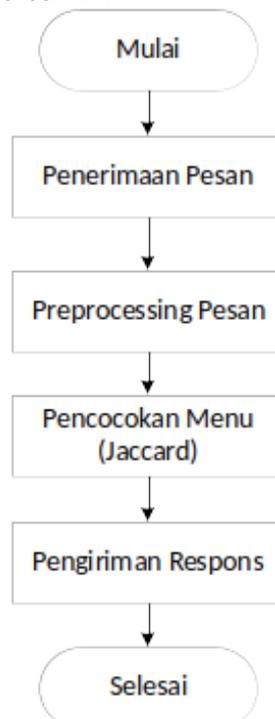
Penelitian ini berfokus pada pengaruh proses pencocokan menu berbasis *keywords* dengan menggunakan metode Jaccard pada *chatbot* ketika mencocokkan pesan yang dikirim pengguna dengan menu yang tersedia. Algoritma Jaccard dipilih karena dapat melakukan perbandingan tanpa melihat posisi penulisan dari sebuah dokumen atau kalimat [4]. Hal tersebut dinilai baik untuk digunakan sebagai pencocokan menu berbasis *keywords* pada *chatbot* karena dapat memberikan opsi yang lebih banyak terhadap menu yang akan dipilih pengguna. Proses pencocokan menu *chatbot* dilakukan dengan membandingkan setiap kata pada pesan yang dikirim pengguna dengan menu yang disediakan. Sistem pencocokan menu berbasis *keywords* pada *chatbot* dibangun menggunakan Bahasa Pemrograman PHP dengan studi kasus menggunakan Bahasa Indonesia.

2. Research Method

Pada penelitian ini, terdapat beberapa tahapan mulai dari analisa permasalahan, pengumpulan data, perancangan sistem *chatbot*, dan uji coba sistem *chatbot*. Berikut merupakan rancangan sistem pencocokan menu berbasis *keywords* pada *chatbot* yang ditampilkan dalam diagram alur.

2.1. Diagram Alur Umum

Diagram alur umum merupakan alur secara umum mengenai proses pencocokan menu berbasis *keywords* pada *chatbot*. Diagram alur umum menggambarkan aliran pesan dari dikirim oleh pengguna hingga pengguna menerima kembali pesan respons dari *chatbot*. Berikut merupakan diagram alur umum penelitian ini.

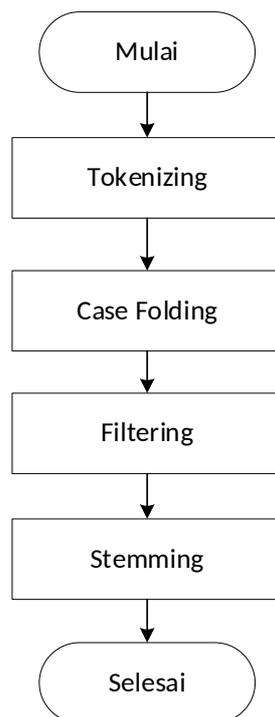


Gambar 1. Diagram Alur Umum Sistem *Chatbot*

Diagram alur umum sistem *chatbot* terdiri dari empat proses. Proses pertama adalah diterimanya pesan yang dikirim pengguna dengan aplikasi Telegram. Proses kedua adalah *preprocessing* pesan yang berfungsi untuk mengubah teks pesan yang dikirim pengguna menjadi sesuai dengan format pemrosesan. Proses ketiga adalah pencocokan menu yang bertujuan untuk mencocokkan pesan yang dikirim pengguna dengan menu yang telah disediakan pada *database chatbot* sehingga memperoleh pesan balasan berupa menu. Proses terakhir adalah pengiriman respons kepada pengguna melalui *chatbot* yang disediakan.

2.2. Diagram Alur Preprocessing

Diagram alur *preprocessing* menjelaskan mengenai pemrosesan tahap awal dari pesan yang akan disesuaikan dengan format yang diperlukan pada tahapan selanjutnya. Berikut merupakan diagram alur *preprocessing* yang digunakan.

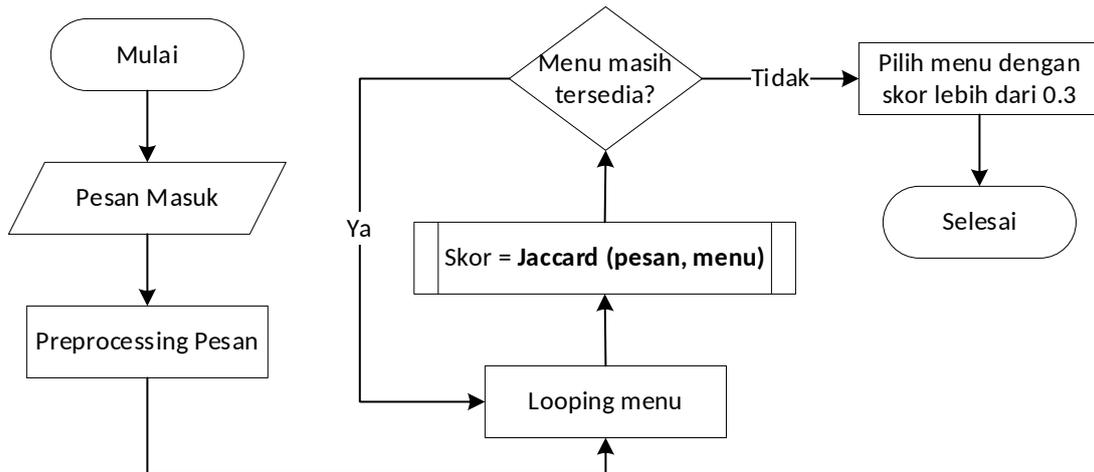


Gambar 2. Diagram Alur Preprocessing

Diagram alur *preprocessing* terdiri dari proses *tokenizing*, proses *case folding*, proses *filtering*, dan proses *stemming*. Proses *tokenizing* berfungsi untuk mengelompokkan kata dari suatu kalimat untuk menjadi token. Proses *case folding* bertujuan untuk menjadikan teks menjadi non-kapital serta penghapusan karakter yang tidak diperlukan. Proses *filtering* berfungsi untuk menghapus *stopwords* atau kata yang tidak memiliki makna. Proses *stemming* berfungsi untuk menjadikan kata menjadi kata dasar untuk kata berimbuhan dengan menggunakan sebuah *library* yaitu Sastrawi.

2.3. Diagram Alur Pencocokan Menu

Diagram alur proses pencocokan menu menjelaskan proses pencocokan pesan yang dikirim oleh pengguna dengan menu yang disediakan pada *database chatbot*. Berikut merupakan diagram alur pencocokan menu yang digunakan pada penelitian ini.



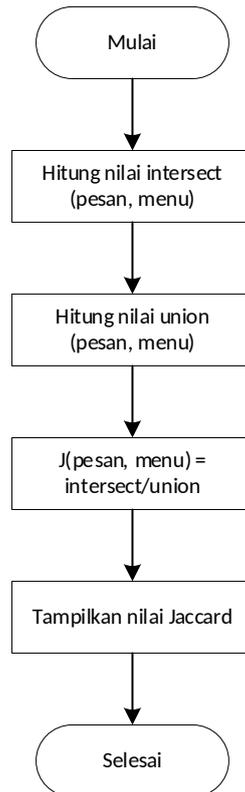
Gambar 3. Diagram Alur Pencocokan Menu

Diagram alur pencocokan menu terdiri dari beberapa tahapan. Penjelasan dari diagram alur pencocokan menu adalah sebagai berikut:

- Masukan dari proses pencocokan menu adalah pesan yang dikirim oleh pengguna.
- Proses selanjutnya yang dijalankan adalah *preprocessing* pesan.
- Sistem kemudian mengakses menu pada *database chatbot*.
- Sistem kemudian membandingkan pesan pengguna dengan semua menu yang tersedia pada *database chatbot*. Proses perbandingan dilakukan dengan menggunakan metode Jaccard.
- Menu yang dipilih adalah menu dengan skor lebih dari *threshold* yang telah ditentukan yaitu 0.3 dan kemudian akan diurutkan berdasarkan menu dengan skor tertinggi.
- Poin (d) akan terus dijalankan hingga semua menu yang tersedia telah dicocokkan dengan pesan pengguna.

2.4. Diagram Alur Jaccard

Diagram alur Jaccard merupakan diagram yang menjelaskan bagaimana proses Algoritma Jaccard bekerja ketika mencocokkan pesan yang dikirim pengguna dengan menu yang tersedia. Berikut merupakan diagram alur Jaccard yang digunakan pada penelitian ini.



Gambar 4. Diagram Alur Jaccard

Diagram alur jaccard terdiri dari beberapa proses. Penjelasan dari diagram alur Jaccard adalah sebagai berikut:

- a. Proses pertama adalah menghitung nilai *intersect* atau jumlah kata yang tidak sama dari kedua *string* atau kalimat.
- b. Proses kedua adalah menghitung nilai *union* atau jumlah kata yang sama dari kedua *string* atau kalimat.
- c. Selanjutnya hitung dengan menggunakan rumus dari Algoritma Jaccard (membagi *intersect* dengan *union*) dan skor sudah didapatkan.

3. Literature Study

Studi literatur dilakukan berdasarkan kajian pustaka yang bersumber dari jurnal, hasil penelitian terdahulu yang dipublikasikan, dan sumber lainnya seperti internet, surat kabar, serta media lainnya yang sesuai dengan pengembangan sistem pencocokan menu berbasis *keywords* pada penelitian ini.

3.1 Teks Preprocessing

Tahap *text preprocessing* memiliki tujuan untuk menyiapkan teks menjadi data siap proses untuk proses selanjutnya dalam rangkaian pemrosesan NLP [5]. Beberapa operasi yang dilakukan pada tahap teks preprocessing adalah *tokenizing*, *case folding*, *filtering*, dan *stemming*.

1) Tokenizing

Sebagian besar proses NLP yang dilakukan beroperasi pada tingkat kata atau kalimat, sehingga *tokenizing* menjadi langkah penting dalam pemrosesan bahasa alami. Beberapa penelitian terkait pemrosesan NLP menyertakan proses *tokenizing* sebagai pemrosesan awal. Tokenisasi bertugas membagi kata dari teks atau kalimat menjadi *set morfem* yang berurutan [6].

2) Case Folding

Case folding pada penelitian ini berfungsi untuk mengganti seluruh huruf pada kata atau kalimat menjadi huruf kecil [5]. Pada tahap ini hasil dari tokenisasi yang telah dilakukan

sebelumnya diubah menjadi huruf non-kapital. Tujuan dilakukannya adalah untuk mempermudah proses selanjutnya ketika data pesan dari pengguna akan dicocokkan dengan data menu yang disimpan pada *database*.

3) *Filtering*

Filtering adalah tahapan untuk memilih kata-kata penting dari token yang tersedia. Terdapat dua pendekatan umum yang dapat digunakan yaitu *stoplist* (menghilangkan kata yang tidak digunakan) atau *wordlist* (menggambil kata yang digunakan). *Stoplist* atau *stopword* adalah kata-kata yang tidak deskriptif [5]. Pada penelitian ini digunakan pendekatan *stoplist* karena *stopwords* yang dideteksi akan dihapus dari kalimat.

4) *Stemming*

Stemming merupakan proses menghilangkan imbuhan yang terdapat pada kata sehingga kata tersebut menjadi kata dasar. Digunakan *Stemmer* Sastrawi yang merupakan sebuah *library* PHP yang menyediakan *stemming* dalam Bahasa Indonesia. Proses *stemming* pada *Stemmer* Sastrawi bergantung pada kamus kata dasar yang digunakan, sumber kata dasar yang digunakan pada *Stemmer* Sastrawi berasal dari *kateglo.com* [7].

3.2 Jaccard Index

Jaccard Index merupakan sebuah algoritma yang digunakan untuk membandingkan kemiripan dari sebuah dokumen ataupun *dataset* [2]. Perhitungan koefisien Jaccard Index antara dua kumpulan data didapatkan berdasarkan hasil pembagian dari jumlah data yang sama yang dibandingkan dengan keseluruhan jumlah data yang ada. Jaccard index dapat dirumuskan sebagai berikut.

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad ()$$

Persamaan 1 merupakan rumus perhitungan menggunakan metode Jaccard Index. Penjelasan dari rumus tersebut adalah A dan B merupakan kumpulan set yang akan dihitung tingkat kemiripannya, $|A \cap B|$ merupakan irisan (*intersect*) dari kumpulan set A dan B , dan $|A \cup B|$ merupakan kesatuan (*union*) dari kumpulan set A dan B .

4. Result and Discussion

Bagian ini berisi hasil dan pembahasan dari sistem pencocokan menu berbasis *keywords* pada *chatbot* dengan metode Jaccard.

4.1 Implementasi Jaccard pada PHP

Metode Jaccard dipilih karena dinilai paling sesuai dengan pencocokan *string* yang dapat dirubah posisi karakternya, dimana dalam konteks ini adalah menu. Pencocokan menu sangat efektif dilakukan tanpa memperhitungkan posisi *string* pada suatu kalimat. Berikut merupakan kode program dari metode Jaccard yang diimplementasikan dengan Bahasa Pemrograman PHP.

```
function jaccard($item1, $item2)
{
    $arr_intersection = array_intersect( $item2, $item1 );
    $arr_union = array_unique(array_merge( $item1, $item2 ));
    $coefficient = count( $arr_intersection ) / count( $arr_union );

    return $coefficient;
}
```

Kode Program 1. Jaccard Index

Kode Program 1 merupakan pengaplikasian dari metode Jaccard Index. Sesuai dengan rumus Jaccard Index yaitu membagi antara jumlah data yang sama dengan keseluruhan jumlah data yang ada. Fungsi PHP *array intersect* digunakan untuk memperoleh *string* atau data yang sama diantara dua kalimat dan untuk mendapatkan keseluruhan data dengan catatan data yang sama dihitung satu digunakan fungsi PHP *array marge* yang kemudian dikombinasikan dengan fungsi *array unique*.

4.2 Perhitungan Jaccard Terhadap Menu

Proses perhitungan yang dilakukan berdasarkan rumus Jaccard Index. Berikut merupakan proses perhitungan menggunakan metode Jaccard Index antara pesan yang dikirim pengguna terhadap beberapa menu *chatbot* yang disediakan.

Tabel 1. Contoh Pesan Pengguna dan Menu untuk Perhitungan Jaccard

No	Pesan Setelah Preprocessing (A)	Menu Teratas (B)	
1	tambah aktivitas	1	Tambah Aktivitas
		2	Tambah Departemen
		3	Hapus Aktivitas
2	aktivitas	1	Tambah Aktivitas
		2	Hapus Aktivitas
		3	Tampilkan Foto Awal Aktivitas
3	aktivitas tambah	1	Tambah Aktivitas
		2	Tambah Departemen
		3	Hapus Aktivitas

Tabel 1 merupakan contoh pesan yang dikirim pengguna dan menu yang didapatkan *chatbot* berdasarkan hasil pencocokan. Perhitungan kemiripan dengan menggunakan metode Jaccard pada contoh kedua “aktivitas” dengan menu “tambah aktivitas” dengan menggunakan spasi sebagai *delimiter* maka A = [aktivitas] dan B = [tambah, aktivitas]. Irisan (*intersect*) himpunan A dan B ($A \cap B$) = [aktivitas] dengan jumlah satu. Kesatuan (*union*) himpunan A dan B ($A \cup B$) = [tambah, aktivitas] dengan jumlah dua. Hasil perhitungan kemiripan kedua kalimat dengan perhitungan Jaccard dapat dilihat sebagai berikut.

$$Jaccard(A, B) = \frac{|1|}{|2|} = 0.5$$

Gambar 5. Contoh Hasil Perhitungan Jaccard

Gambar 5 merupakan hasil perhitungan Jaccard dari kata “aktivitas” dengan kalimat “tambah aktivitas”. Hasil perhitungan rumus Jaccard dari kedua data tersebut menghasilkan skor 0,5. Hasil keseluruhan dari perhitungan Jaccard antara himpunan A dan B ditampilkan pada Tabel 2.

Tabel 2. Hasil Perhitungan Jaccard

No.	Pesan Setelah Preprocessing (A)	Menu Teratas (B)		Hasil Perhitungan Jaccard
1	tambah aktivitas	1	Tambah Aktivitas (Harapan)	1
		2	Tambah Departemen	0.33
		3	Hapus Aktivitas	0.33
2	aktivitas	1	Tambah Aktivitas (Harapan)	0.5
		2	Hapus Aktivitas	0.5
		3	Tampilkan Foto Awal Aktivitas	0.25
3	aktivitas tambah	1	Tambah Aktivitas (Harapan)	1
		2	Tambah Departemen	0.33
		3	Hapus Aktivitas	0.33

Hasil perhitungan pada Tabel 2 menunjukkan bahwa penggunaan metode Jaccard pada pencocokan menu berbasis *keywords* dapat memberikan hasil yang baik. Metode Jaccard yang digunakan mampu menangani beberapa kondisi seperti menggunakan keseluruhan kata kunci, sebagian kata kunci, maupun beberapa kata kunci dengan posisi yang ditukar untuk dapat menentukan menu yang diharapkan.

5. Conclusion

Pencocokan menu berbasis *keywords* pada *chatbot* dengan metode Jaccard dapat diterapkan pada bahasa pemrograman PHP. Penggunaan metode Jaccard mampu menangani beberapa kondisi seperti ketika pesan terdiri dari keseluruhan kata kunci, sebagian kata kunci, maupun beberapa kata kunci dengan posisi yang ditukar. Pencocokan menu dengan memanfaatkan metode Jaccard dinilai sangat efektif karena tidak memperhitungkan posisi *string* pada suatu menu. Misalnya ketika ada sebuah menu dengan nama "Tambah aktivitas", namun pengguna mengetikkan kalimat "aktivitas tambah" maka hasil yang diberikan sistem *chatbot* masih memiliki nilai sempurna atau maksimal.

Sistem pencocokan menu ini diharapkan dapat dikembangkan dengan menggunakan algoritma NLP lainnya agar dapat diketahui kelebihan ataupun kekurangan dari masing-masing algoritma NLP tersebut. Diharapkan juga sistem pencocokan menu ini dapat dikombinasikan dengan sistem lainnya seperti perbaikan salah ketik ataupun pencocokan berbasis *semantik* (kemiripan antara teks atau dokumen berdasarkan makna kontekstualnya).

References

- [1] I. N. S. Paliwahet, I. M. Sukarsa, and I. K. Gede Darma Putra, "Pencarian Informasi Wisata Daerah Bali Menggunakan Teknologi Chatbot," *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 8, no. 3, p. 144, 2017, doi: 10.24843/lkjiti.2017.v08.i03.p01.
 - [2] K. Rinatha, "Simple Query Suggestion Untuk Pencarian Artikel Menggunakan Jaccard Similarity," *J. Ilm. Rekayasa dan Manaj. Sist. Inf.*, vol. 3, no. 1, pp. 30–34, 2017.
 - [3] M. Ermawati and J. L. Buliali, "Text Based Approach For Similar Traffic Incident Detection from Twitter," *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 9, no. 2, p. 63, 2018, doi: 10.24843/lkjiti.2018.v09.i02.p01.
 - [4] O. Nurdiana, J. Jumadi, and D. Nursantika, "Perbandingan Metode Cosine Similarity Dengan Metode Jaccard Similarity Pada Aplikasi Pencarian Terjemah Al-Qur'an Dalam Bahasa Indonesia," *J. Online Inform.*, vol. 1, no. 1, p. 59, 2016, doi: 10.15575/join.v1i1.12.
 - [5] N. P. Putra and Sularno, "Penerapan Algoritma Rabin-Karp Dengan Pendekatan Synonym Recognition Sebagai Antisipasi Plagiarisme Pada Penulisan Skripsi," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 1, no. 2, pp. 49–58, 2019.
 - [6] A. H. Aliwy, "Tokenization as Preprocessing for Arabic Tagging System," *Int. J. Inf. Educ. Technol.*, vol. 2, no. 4, pp. 348–353, 2012, doi: 10.7763/ijiet.2012.v2.149.
 - [7] N. Saputra, T. B. Adji, and A. E. Permanasari, "Analisis Sentimen Data Presiden Jokowi dengan Preprocessing Normalisasi dan Stemming Menggunakan Metode Naive Bayes dan SVM," *J. Din. Inform.*, vol. 5, no. November, p. 12, 2015.
-