



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1. Resep Masakan

Menurut Borghini, A. (2015), resep adalah alat utama dalam budaya kuliner apa pun, yang memberikan instruksi kepada orang mengenai cara menyiapkan bahan-bahan dengan cara yang aman, bergizi, menyenangkan. Meningkatnya pentingnya resep juga dipamerkan dan berkembang melalui media. Buku memasak sendiri telah menjadi sektor utama industri penerbitan.

2.2. Jaro-Winkler Distance

Menurut Christian Wibisono (2018), algoritma Jaro-Winkler Distance memiliki kompleksitas waktu *quadratic runtime complexity* yang sangat efektif pada *string* pendek dan dapat bekerja lebih cepat dari algoritma *edit distance*. Pada algoritma Jaro digunakan rumus untuk menghitung jarak (*dj*) antara dua *string* yaitu *s1* dan *s2* adalah

$$dj = \frac{1}{3} * \left(\frac{m}{s1} + \frac{m}{s2} + \frac{m-t}{m} \right) \quad \dots(2.1)$$

dengan:

m = jumlah karakter yang sama persis

|s1| = panjang string 1

|s2| = panjang string 2

t = jumlah transposisi

Jarak teoritis dua karakter dihitung sebagai sama apabila tidak melebihi rumus:

$$\left\lceil \frac{\max(|s1|, |s2|)}{2} \right\rceil - 1 \quad \dots(2.2)$$

Akan tetapi bila mengacu kepada nilai yang akan dihasilkan oleh algoritma Jaro-Winkler maka nilai jarak maksimalnya adalah satu yang menandakan kesamaan *string* yang dibandingkan mencapai seratus persen atau sama persis. Biasanya s1 digunakan sebagai acuan untuk urutan di dalam mencari transposisi. Yang dimaksud transposisi di sini adalah karakter yang sama dari *string* yang dibandingkan akan tetapi tertukar urutannya. Sebagai contoh, dalam membandingkan kata CRATE dengan TRACE, bila dilihat seksama maka dapat dikatakan semua karakter yang ada di s1 ada dan sama dengan karakter yang ada di s2 tetapi dengan urutan yang berbeda. Dengan mengganti C dan T, dapat dilihat perubahan kata CRATE menjadi TRACE. Pertukaran dua elemen string inilah adalah contoh nyata dari transposisi yang dijelaskan.

Dalam pencocokkan DwAyNE dan DuANE memiliki urutan yang sama D-A-N-E, jadi tidak ada transposisi.. Jaro-Winkler distance menggunakan *prefix scale* (p) yang memberikan tingkat penilaian yang lebih, dan *prefix length* (l) yang menyatakan panjang awalan yaitu panjang karakter yang sama dari *string* yang dibandingkan sampai ditemukannya ketidaksamaan. Bila *string* s1 dan s2 yang diperbandingkan, maka Jaro-Winkler Distance-nya (dw) adalah:

$$dw = dj + (lp(1 - dj)) \quad \dots(2.3)$$

Keterangan :

dj = Jaro Distance untuk *string* s1 dan s2

l = panjang prefiks umum di awal *string* nilai maksimalnya 4 karakter (panjang karakter yang sama sebelum ditemukan ketidaksamaan max 4)

p = konstanta *scaling factor*. Nilai standar untuk konstanta ini menurut Winkler adalah $p = 0,1$.

Contoh kasus:

1. Diberikan *string* yang memiliki panjang yang berbeda yaitu, s_1 adalah JONES dan s_2 adalah JOHNSON.

$$m = 4$$

$$s_1 = 5$$

$$s_2 = 7$$

Dari kata JONES dan JOHNSON diketahui bahwa tidak terjadi transposisi maka nilai $t = 0$.

$$dj = \frac{1}{3} * \left(\frac{4}{5} + \frac{4}{7} + \frac{4-0}{4} \right) = 0.790$$

2. *String* JONES dan JOHNSON hanya cocok pada dua karakter pertamanya yaitu JO, sehingga Jaro Winkler distancenya adalah:

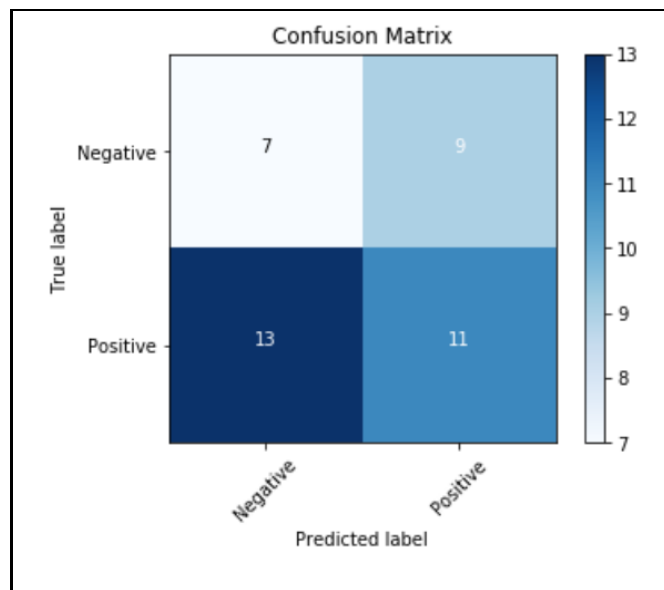
$$dw = 0.790 + 0.1 \times 2 \times (1.0 - 0.790) = 0.832$$

2.3. Ketepatan (Accuracy)

Akurasi adalah metrik evaluasi yang digunakan untuk mengukur jumlah prediksi yang diterima model. Formula untuk akurasi di bawah ini:

$$Accuracy = \frac{True\ Negative + True\ Positive}{True\ Positive + False\ Positive + True\ Negative + False\ Negative} \dots(2.4)$$

Akurasi dapat menghitung berapa persen dari prediksi yang benar. Akurasi melihat True Positive dan True Negatives. *Confusion matrix* menampilkan jumlah *True Positive*, *False Positive*, *True Negative*, dan *False Negative* yang diproduksi oleh model. Menggunakan *Confusion matrix* bisa mendapatkan nilai yang dibutuhkan untuk menghitung akurasi. Melihat *confusion matrix* di bawah ini (Gambar 1.1), kita dapat menggunakan rumus di atas untuk menghitung akurasi.



Gambar 2.1 Confusion Matrix (Erika D, 2018)

$$Accuracy = \frac{11 + 7}{11 + 7 + 9 + 13} = 0.45$$

Jadi, hasil *accuracy* dari data/*confusion matrix* tersebut adalah 0.45 (Erika D, 2018).

2.4. Technology Acceptance Model (TAM)

Pada waktu yang sama ketika *usability researchers* membuat kuesioner terstandarisasi pertama untuk menilai tingkat *usability*, banyak penelitian mempelajari adopsi dari sistem informasi yang menangani masalah yang serupa,

dan salah satu yang paling berpengaruh adalah *Technology Acceptance Model* (TAM). Sejumlah penelitian mendukung keabsahan dari TAM dan penjelasan yang memuaskan dari *end-user system usage*. Berdasarkan *Technology Acceptance Model* (TAM), faktor yang paling utama yang berpengaruh dalam penggunaan teknologi adalah *Perceived Usefulness* (PU) dan *Perceived Ease of Use* (PEU). Dalam TAM, PU merupakan tingkat yang mana orang percaya menggunakan teknologi tersebut akan meningkatkan performa pekerjaannya, dan PEU adalah tingkat yang mana orang percaya menggunakan teknologi tersebut akan mengurangi usaha/pekerjaannya (effortless).

Kuesioner TAM terdiri dari 12 pernyataan, enam untuk PU (Gambar 2.2) dan enam lainnya untuk PEU (Gambar 2.3), pengisi kuisisioner akan menanggapi dengan tingkat setuju.

<i>Perceived Usefulness (PU)</i>	Likely						Unlikely
	<i>Extremely</i>	<i>Quite</i>	<i>Slightly</i>	<i>Neither</i>	<i>Slightly</i>	<i>Quite</i>	<i>Extremely</i>
1. Using [this product] in my job would enable me to accomplish tasks more quickly.							
2. Using [this product] would improve my job performance.							
3. Using [this product] in my job would increase my productivity.							
4. Using [this product] would enhance my effectiveness on the job.							
5. Using [this product] would make it easier to do my job.							
6. I would find [this product] useful in my job.							

Gambar 2.2 Kuesioner Perceived Usefulness (PU) dalam TAM (James R. Lewis, 2019)

Perceived Ease-of-Use (PEU)	Likely						Unlikely	
	Extremely	Quite	Slightly	Neither	Slightly	Quite	Extremely	
7. Learning to operate [this product] would be easy for me.								
8. I would find it easy to get [this product] to do what I want it to do.								
9. My interaction with [this product] would be clear and understandable.								
10. I would find [this product] would be clear and understandable.								
11. It would be easy for me to become skillful at using [this product].								
12. I would find [this product] easy to use.								

Gambar 2.3 Kuesioner Perceived Ease-of-Use (PEU) dalam TAM (James R. Lewis, 2019)

Dalam perhitungan nilai TAM, digunakan skala poin 0-100, kemudian mencari rata-rata PU dan PEU (James R. Lewis, 2019).

$$PU = (AVERAGE(TAM01, TAM02, TAM03, TAM04, TAM05, TAM06) - 1) \left(\frac{100}{6} \right) \dots (2.5)$$

$$PEU = (AVERAGE(TAM07, TAM08, TAM09, TAM10, TAM11, TAM12) - 1) \left(\frac{100}{6} \right) \dots (2.6)$$