

# Klasifikasi Berita Clickbait Menggunakan K-Nearest Neighbor (KNN)

Riska Sagita<sup>1</sup>, Ultach Enri<sup>2</sup>, Aji Primajaya<sup>3</sup>

Teknik Informatika, Universitas Singaperbangsa Karawang

Jl. HS. Ronggo Waluyo, Telukjambe Timur, Karawang, Jawa Barat, Indonesia - 41361

E-mail: <sup>1</sup>riskasagita1999@gmail.com, <sup>2</sup>ultach@staff.unsika.ac.id,

<sup>3</sup>aji.primajaya@staff.unsika.ac.id

Diterima: 24 Juni 2020; Direvisi: 14 Oktober 2020; Disetujui: 30 Oktober 2020

## Abstrak

Clickbait menjadi salah satu cara untuk mencari pendapatan dengan meningkatkan traffic pembaca dan pengunjung. Praktik clickbait ini memberikan pengaruh besar terhadap penyedia situs berita karena rasa keingintahuan dari para pembaca serta sulitnya para pembaca memilih berita clickbait atau bukan clickbait. Praktik clickbait ini sendiri sangat di andalkan oleh penyedia situs berita yang kerap menggunakan judul–judul yang menjebak untuk menarik para pembaca. Untuk mendapatkan hasil berita clickbait atau bukan clickbait digunakanlah K-Nearest Neighbors (K-NN) dimana waktu proses pada KKN lebih cepat dari pada waktu proses yang lainnya. Dari hasil penelitian yang dilakukan untuk pengklasifikasian berita clickbait, didapatkan hasil terbaik pada jumlah  $k=11$  dengan menggunakan skenario 1 pada pembagian data dengan jumlah data sebanyak 800 data latih dan 200 data uji yang menghasilkan akurasi sebesar 71%, Precision 72%, dan Recall 71%. Hal ini menunjukkan bahwa klasifikasi berita clickbait dapat di klasifikasikan menggunakan K-Nearest Neighbor.

**Kata kunci:** klasifikasi, text mining, clickbait, k-nearest neighbor

## Abstract

Clickbait is a way to generate income by increasing traffic to readers and visitors. This clickbait practice has a major influence on news site providers because of the curiosity of the readers and the difficulty of readers choosing clickbait or non-clickbait news. The clickbait practice itself is heavily relied on by news site providers who often use tricky titles to attract readers. K-Nearest Neighbors (K-NN) is used to get clickbait or non-clickbait news results, where the processing time for KKN is faster than the other processing times. From the results of research conducted for the classification of clickbait news, the best results were obtained at the number  $k = 11$  by using scenario 1 on data sharing with 800 training data and 200 test data which resulted in an accuracy of 71%, 72% Precision, and 71 Recall. %. This shows that the classification of clickbait news can be classified using K-Nearest Neighbor.

**Keywords:** classification, text mining, clickbait, k-nearest neighbor

## 1. PENDAHULUAN

Internet adalah salah satu teknologi informasi yang sangat diminati oleh banyak kalangan, karena internet telah menyentuh berbagai aspek kebudayaan manusia mulai dari gaya hidup, pendidikan, penelitian hingga ke dunia bisnis. Salah satu manfaat dari internet adalah sebagai penyedia informasi yang cepat dan fleksibel, namun terkadang informasi yang diberikan tidak

---

sesuai dengan judul artikel di media elektronik. Judul yang menjebak (*Clickbait*) merupakan modus media *online* untuk meningkatkan *traffic* pengunjung [1]

*Clickbait* menjadi salah satu cara untuk mencari pendapatan dengan meningkatkan *traffic* pembaca dan pengunjung. Semakin banyak pengunjung situs, semakin banyak pula kemungkinan untuk mendapatkan pendapatan dari situs tersebut. Faktor pendorong maraknya *clickbait* di media *online* adalah persaingan yang semakin ketat antar media untuk mendapatkan pembaca [2] dan pendapatan yang melimpah pada *viewer*, iklan, memberikan keuntungan berupa materi walaupun pada dasarnya antara judul dan isi tidak memiliki maksud yang sama atau judul yang bombastis tersebut hanya dijelaskan secara singkat atau tidak utuh [3]

Praktik *clickbait* yang marak terjadi membuat para pembaca kesulitan untuk memilih berita *clickbait* atau bukan *clickbait* karena pada dasarnya praktik *clickbait* ini dijadikan sebagai daya tarik para penyedia situs untuk memancing para pembaca yang penasaran akan judul-judul berita yang di buat oleh penyedia situs sehingga praktik *clickbait* ini memberikan pengaruh yang sangat besar terhadap penyedia situs.

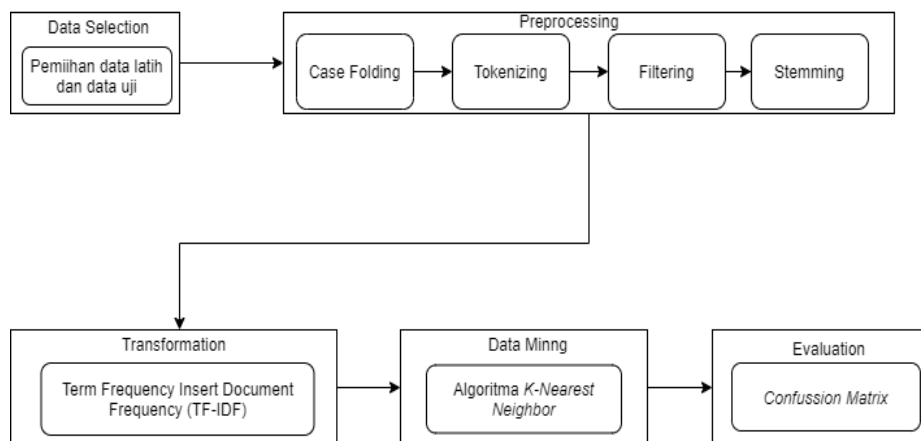
Berdasarkan praktik *clickbait* yang marak terjadi saat ini untuk pengklasifikasian *clickbait* atau bukan *clickbait* pada berita pengelompokan berita dilakukan dengan teknik *text mining*. *Text Mining* dapat didefinisikan penemuan serta ekstraksi pengetahuan yang menarik dari sebuah teks yang bebas maupun tidak struktur dan mencakup segala sesuatu mulai dari pengambilan informasi yaitu pengambilan dokumen atau pengambilan web untuk klasifikasi. Dalam penelitian [4] di dapatkan hasil bahwa KNN mendapatkan nilai akurasi yang lebih tinggi dibandingkan dengan SVM. Pada penelitian [5] didapatkan bahwa waktu proses pada KNN lebih cepat dari pada waktu proses SVM maka dapat di silmpulkan bahwa KNN merupakan algoritma yang terbaik. Dari penjelasan yang sudah dipaparkan sebelumnya dapat di lihat menggunakan *clickbait* pada judul berita memberikan pengaruh besar terhadap situs berita. Tetapi untuk para pembaca hal itu sangat membuat para pembaca resah karena para pembaca tidak dapat membedakan berita *clickbait* atau bukan *clickbait*.

Pada penelitian ini menggunakan metode text mining dimana pada metode text mining, dengan 3 skenario untuk pembagian dan algoritma K-Nearest Neighbor ini diharapkan dapat membantu pembaca atau pengunjung situs dalam memilih berita *clickbait* atau bukan *clickbait*.

## 2. METODOLOGI PENELITIAN

Metode yang digunakan pada penelitian ini adalah metode text mining, dimana metode ini terdapat beberapa proses diantaranya adalah, 1) *data Selection*, 2) *pre-processing*, 3) *transformation*, 4) *data mining*, dan 5) *evaluation*, berikut alur penelitian pada gambar 1.

### 2.1 Data Selection



Gambar 1. Alur penelitian

Pada tahap ini merupakan tahap persiapan dan pemilihan data. Pemilihan data dilakukan dengan mengumpulkan 1000 berita, artikel berita yang akan menjadi data training dan data testing didapat dari situs berita dengan data berita terbit pada bulan januari 2020. Artikel berita yang telah dikumpulkan akan di kelompokkan dan dilabeli secara manual sesuai dengan kategorinya masing-masing yakni *clickbait* dan bukan *clickbait*. Pembagian data digunakan 3 skenario pembagian antara data training berbanding dengan data testing yaitu, 80%:20%, 50%:50%, 20%:80%. Dari 3 skenario pembagian data bertujuan untuk mendapatkan nilai performa sistem jika jumlah data training lebih banyak dari data testing, jumlah data training sama dengan jumlah data testing dan jika jumlah data training lebih sedikit dibandingkan dengan data testing.[6]

## 2.2 Pre-processing

Pada tahap *preprocessing* terdapat beberapa tahapan untuk mentransformasikan data ke dalam bentuk representasi format data lain, berikut tahapan-tahapan pada *pre-processing*:

1. *Case folding* adalah transformasi huruf pada teks, semua huruf diubah menjadi huruf kecil. Karakter huruf yang dihasilkan harus berupa 'a' sampai karakter huruf 'z'. Selain karakter huruf tersebut harus dianulir [7].
2. *Tokenizing* merupakan proses untuk memisahkan kata-kata dari sebuah kalimat atau memutus urutan *string* menjadi potongan-potongan seperti kata-kata berdasarkan tiap kata yang menyusunnya [8].
3. *Stopwords Removal atau Filtering*, *stoplist* atau *stopword* adalah proses menghilangkan kata-kata yang tidak penting dengan pendekatan *bag-of- words*. Hasil dari *stoplist* adalah *wordlist* yang berisi kata penting. [9].
4. *Stemming*, merupakan proses pemetaan dan penguraian berbagai bentuk (*variants*) dari suatu kata menjadi bentuk kata dasarnya (*stem*) [10].

## 2.3 Transformation

Pada tahap ini metode dipresentasikan dalam dokumen yang berisi bobot kata. Bobot tersebut menyatakan kepentingan atau kontribusi kata terhadap suatu dokumen dan kumpulan dokumen. Tahap pembobotan kata pada dokumen menggunakan metode TF-IDF. Metode TF-IDF ini menggabungkan dua konsep yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen dan inverse frekuensi dokumen yang mengandung kata tersebut [11].

Rumus dalam menentukan pembobot dengan TF-IDF seperti pada persamaan (1):

$$IDF (Word) = \log \frac{td}{df} \quad (1)$$

IDF (*word*) adalah nilai IDF dari setiap kata yang akan di cari, *td* adalah jumlah keseluruhan dokumen yang ada, *df* adalah jumlah kemunculan kata pada semua dokumen.

## 2.4 Data Mining

Pada tahap ini menggunakan *K-Nearest Neighbor (KNN)*, Dalam pengklasifikasian terdapat 2 proses yang dilakukan yakni proses *training* dan proses *testing*. Proses pengklasifikasian dilakukan dengan menentukan nilai K terlebih dahulu, menghitung kuadrat jarak euclid(*query instance*) masing-masing objek terhadap *training* data yang diberikan, lalu mengumpulkan label class Y (klasifikasi *Nearest Neighbor*) dengan persamaan (2).

$$D(x_i, y_i) = \sqrt{\sum_{i=1}^n (x_i, y_i)^2} \quad (2)$$

Keterangan:  $X_i = \text{data training}$

$Y_i = \text{data testing}$

$D(x_i, y_i) = \text{jarak}$

$i = \text{variabel data}$

$n = \text{Dimensi data}$

## 2.5 Evaluation

Pada tahap terakhir ini dilakukan evaluasi nilai *akurasi*, *precision* dan *recall* yang

merupakan tahapan dalam penelitian dengan tujuan untuk mengetahui pengaruh model yang dihasilkan dari pengklasifikasian yang telah dilakukan. Pada penelitian ini hasil yang disajikan berupa grafik dengan metode validasi yang digunakan adalah *Confusion matrix* yang memberikan keputusan berdasarkan hasil dari *training* dan *testing* serta memberikan penilaian performansi klasifikasi berdasarkan data dengan benar atau salah.

Tabel 1. *Confusion matrix*

| Classification | Predicted Class         |                         |
|----------------|-------------------------|-------------------------|
|                | Class = Yes             | Class = No              |
| Class = Yes    | a (True Positive – TP)  | b (False Negative – FN) |
| Class = No     | c (False Positive – FP) | d (True Negative – TN)  |

Terdapat beberapa cara untuk mengukur performa, beberapa cara yang sering digunakan adalah dengan menghitung akurasi, *recall*, dan *precision*[8].

$$Akurasi = \frac{TP+TN}{Tp+TN+FP+FN} \quad (3)$$

$$Precision = \frac{TP}{Tp+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

Evaluasi dan validasi hasil dihitung dengan menggunakan rumus *accuracy* pada persamaan (3), *precision* pada persamaan (4), *sensitifity* pada persamaan (5), dan *spesificity*. Nilai akurasi dinyatakan dalam persentase. Jika akurasi mencapai angka 100%, hal tersebut berarti bahwa semua kasus yang diprediksi diklasifikasikan seluruhnya dengan benar [12].

### 3. HASIL DAN DISKUSI

Bagian ini memaparkan hasil dari proses klasifikasi berita *clickbait*. Proses dimulai dengan membagi data berita yang dikumpulkan pada situs berita dengan data terbit pada bulan Januari 2020 ke dalam data latih dan data uji. Kedua, melaksanakan setiap langkah *pre-processing*, kemudian menghitung bobot dengan TF-IDF dan implementasi algoritma *K-Nearest Neighbor (KNN)*. Terakhir melakukan evaluasi dan uji validasi dengan *Confusion Matrix*

#### 3.1 Data Selection

Pada penelitian ini data berita yang digunakan adalah 1000 berita yang telah dilakukan labelling dengan 2 kategori yaitu kategori *clickbait* dan bukan *clickbait*. Data berita dilakukan pelabelan dengan cara manual tiap berita diberi label *clickbait* (C) atau bukan *clickbait* (BC) yang ditampilkan pada gambar 2. Berita yang dilabeli *clickbait* (C) mengandung kata yang bersifat hiperbola pada judul dan ketidaksesuaian isi berita dengan judul, sedangkan berita yang dilabeli bukan *clickbait* (BC) berita yang tidak mengandung kata hiperbola dan berita sudah sesuai antara judul dan isi berita. Data yang digunakan sudah di periksa kembali dengan membaca isi berita, berita yang sesuai dengan judul atau yang tidak sesuai dengan judul, berita yang tidak sesuai ialah yang nantinya akan mendapatkan label dengan kategori *clickbait*. Proses pelabelan memakan waktu 2 bulan, mulai dari tanggal 16 Maret sampai 16 Mei 2020.

Setelah dilakukan pelabelan pembagian data training dan data testing berdasarkan penelitian [6] dilakukan dengan 3 skenario rasio yaitu 80%:20%, 50%:50%, dan 20%:80% untuk melihat performa dari sistem jika jumlah data training lebih banyak dari data testing dan data training sama dengan data testing begitupun sebaliknya. Data input pada penelitian menggunakan format ekstensi *xlsx*.

| Judul  | Kategori |
|--|----------|
| Yasonna Copot Dirjen Imigrasi Ronny Sompie Terkait Kasus Harun Masiku, Ini Respons KPK               | C        |
| 2 Bukti Ditemukan, Polisi Bekuk Pegawai Kedai Kopi Lempar Susu ke Driver Ojol sampai Berdarah        | BC       |
| 2 Orang Kembali Dilaporkan Terinfeksi Virus Corona di Thailand                                       | BC       |
| 2 Pengasuh PAUD Jadi Tersangka Kasus Mayat Balita Tanpa Kepala, Ayah Korban_ Kami Tak Terlalu Senang | BC       |
| 3 Hari Cari Lutfi Alfiandi, Ibunda Ungkap Telpon Terakhir Saat Anak Ditangkap_ Ampun Ampun ke Polisi | BC       |
| 3 Minuman Tradisional di Bandung yang Bisa Menghangatkan Tubuh Saat Musim Hujan                      | BC       |
| 3 Restoran Legendaris di Jakarta Pusat Sejak 1960-an, Cocok Bersantap Bersama Keluarga               | BC       |
| 3 Zodiak Pendiam yang Tidak Suka Membuka Obrolan dengan Orang Baru, Kamu Termasuk_                   | BC       |
| 4 Cara Mudah untuk Mengurangi Kebiasaan Menunda Pekerjaan, Coba Yuk!                                 | BC       |
| 4 Fakta Jelang Hasil Autopsi Lina Diumumkan_ Ada Video CCTV Sebelum Lina Meninggal, Kekecewaan Teddy | BC       |
| 5 Fakta Pasukan Quds, Pasukan Elit Iran yang Dipimpin Mayjen Qasem Soleimani, Pahlawan atau Teroris  | C        |
| 5 Fakta Penangkapan Kurir Narkoba di Serpong, dari Kode 'Rahasia' hingga Sabu Senilai 864 Miliar     | BC       |
| 5 FAKTA TERBARU Kasus Medina Zein_ Akui Tak Beri ASI pada Anaknya, hingga Dukungan Zaskia Sungkar    | BC       |
| 5 Hantu Populer yang Dikenal Masyarakat Indonesia, dari Tuyul hingga Kuntilanak                      | BC       |
| 5 Wanita Cantik 'Korban' Valentino Rossi, Ada Mantan Istri Musuh Bebuyutannya di MotoGP              | BC       |

Gambar 2. Data berita hasil pelabelan

### 3.2 Pre-processing

Pada tahanan *pre-procesing* terdapat beberapa tahap yang harus dilakukan yaitu tahap *case folding*, *tokening*, *filtering*, dan *stemming*. Tahapan ini menggunakan bahasa pemrograman Python pada Google Colab dengan memanfaatkan library Sastrawi dengan hasil pada tabel 2.

1. *Case folding*, tahap ini data berita dirubah menjadi huruf kecil, menghapus angka, menghapus tanda baca dan menghapus spasi.
2. *Tokenizing*, tahap ini dilakukan pemecahan kalimat judul berita menjadi kata sehingga setelah *tokenizing* judul berita terpisah-pisah perkata.
3. *Filtering*, tahap ini dilakukan penghilangan kata sambung atau kata yang tidak penting dengan menggunakan teknik *removes stopwords*. Pada penelitian *library* yang digunakan untuk teknik *removes stopwords* adalah *library* Sastrawi.
4. *Stemming*, Pada tahap ini mengubah kata imbuhan menjadi kata dasar. proses ini juga menggunakan *library* yang sama yaitu *library* Sastrawi

Tabel 2. Contoh proses *pre-processing*

|                     |   |
|---------------------|---|
| Data                | 3 Zodiak Pendiam yang Tidak Suka Membuka Obrolan dengan Orang Baru, Kamu Termasuk         |
| <i>Case Folding</i> | zodiak pendiam yang tidak suka membuka obrolan dengan orang baru kamu termasuk            |
| <i>Tokenizing</i>   | zodiac, pendiam, yang, tidak, suka, membuka, obrolan, dengan, orang, baru, kamu, termasuk |
| <i>Filtering</i>    | zodiac, pendiam, suka, membuka, obrolan, orang, baru, kamu, termasuk                      |
| <i>Stemming</i>     | zodiac, diam, suka, membuka, obrolan, orang, baru, kamu, termasuk                         |

### 3.3 Transformation

Pada tahap ini dilakukan proses *Term Frequency* dilakukan untuk pembobotan kata dengan menggunakan TF-IDF. Pemberian bobot nilai pada tiap kata akan dihitung probabilitasnya, perhitungan pembobotan kata dilakukan dengan cara menentukan nilai *Term Frequency* (TF) seperti pada gambar 3.

|           | tfidf    |
|-----------|----------|
| sompie    | 0.345173 |
| ronny     | 0.345173 |
| respons   | 0.312015 |
| dirjen    | 0.312015 |
| copot     | 0.301340 |
| ...       | ...      |
| intip     | 0.000000 |
| investasi | 0.000000 |
| ipb       | 0.000000 |
| iphone    | 0.000000 |
| zombie    | 0.000000 |

3242 rows x 1 columns

Gambar 3 Hasil Pembobotan Kata TF-IDF

### 3.4 Data Mining

Pada tahap ini dilakukan pemodelan dengan *K-Nearest Neighbor* (KNN). Pada pemodelan ini pembagian *data training* dan *data testing* dibagi menjadi 3 skenario yang ditampilkan pada tabel 3 yaitu pembagian antara data training berbanding dengan data testing yaitu, 80%:20%, 50%:50%, 20%:80%.

Tabel 3 Pembagian *Data Training* dan *Testing*

| Skenario   | Data Training (%) | Data Testing (%) |
|------------|-------------------|------------------|
| Skenario 1 | 80%               | 20%              |
| Skenario 2 | 50%               | 50%              |
| Skenario 3 | 20%               | 80%              |

Pengujian KNN pada penelitian ini dilakukan pemodelan dengan menggunakan *K-Nearest Neighbor* dengan parameter nilai *k* pada KNN yaitu 1,3,5,7,9,11,13,15 dimana untuk melihat apakah parameter nilai *k* ini mempengaruhi hasil dari klasifikasi dokumen berita *clickbait* tersebut. Berikut merupakan hasil dari modek klasifikasi KNN dengan menggunakan data training.

Tabel 4 Tabel Hasil Nilai Akurasi

|       | Skenario 1 | Skenario 2 | Skenario 3 |
|-------|------------|------------|------------|
| 1-NN  | 58%        | 55%        | 57%        |
| 3-NN  | 57%        | 54%        | 60%        |
| 5-NN  | 66%        | 58%        | 58%        |
| 7-NN  | 68%        | 56%        | 59%        |
| 9-NN  | 69%        | 59%        | 59%        |
| 11-NN | 71%        | 57%        | 56%        |
| 13-NN | 69%        | 55%        | 56%        |
| 15-NN | 68%        | 56%        | 55%        |

Berdasarkan tabel 4 dapat dilihat bahwa hasil model KNN untuk nilai akurasi dengan nilai *k*=11 mendapatkan nilai terbaik pada skenario 1 yang menghasilkan nilai akurasi sebesar 71%.

Tabel 5 Tabel Hasil Nilai *Precision*

|       | Skenario 1 | Skenario 2 | Skenario 3 |
|-------|------------|------------|------------|
| 1-NN  | 58%        | 55%        | 57%        |
| 3-NN  | 57%        | 54%        | 60%        |
| 5-NN  | 66%        | 58%        | 58%        |
| 7-NN  | 68%        | 56%        | 59%        |
| 9-NN  | 69%        | 59%        | 59%        |
| 11-NN | 72%        | 57%        | 56%        |
| 13-NN | 69%        | 55%        | 56%        |
| 15-NN | 68%        | 56%        | 55%        |

Berdasarkan tabel 5 dapat dilihat bahwa hasil model KNN untuk nilai *Precision* dengan nilai  $k=11$  mendapatkan nilai terbaik pada skenario 1 yang menghasilkan nilai akurasi sebesar 72%.

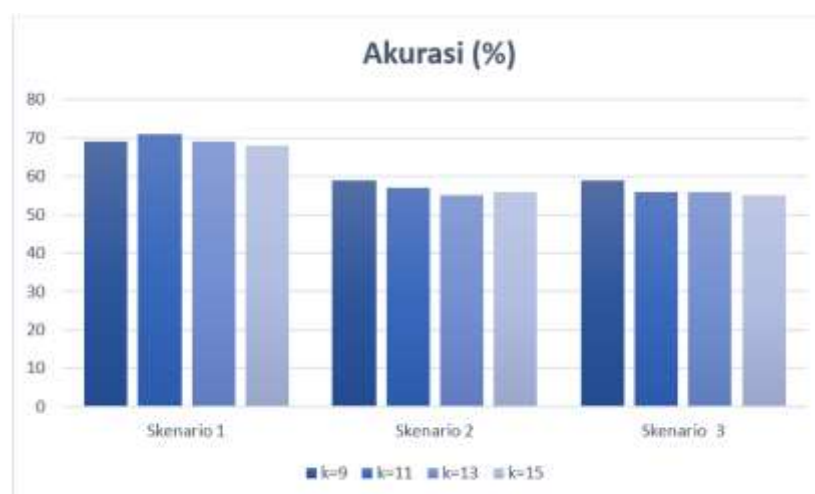
Tabel 6 Tabel Hasil Nilai *Recall*

|       | Skenario 1 | Skenario 2 | Skenario 3 |
|-------|------------|------------|------------|
| 1-NN  | 58%        | 55%        | 57%        |
| 3-NN  | 57%        | 54%        | 60%        |
| 5-NN  | 66%        | 58%        | 58%        |
| 7-NN  | 68%        | 56%        | 59%        |
| 9-NN  | 69%        | 59%        | 59%        |
| 11-NN | 71%        | 57%        | 56%        |
| 13-NN | 69%        | 55%        | 56%        |
| 15-NN | 68%        | 56%        | 55%        |

Berdasarkan tabel 6 dapat dilihat bahwa hasil model KNN untuk nilai *Recall* dengan nilai  $k=11$  mendapatkan nilai terbaik pada skenario 1 yang menghasilkan nilai akurasi sebesar 71%.

### 3.5 Evaluation

Hasil yang diperoleh keseluruhan semua tes yang dilakukan pemodelan KNN menghasilkan nilai tingkat akurasi, precision, dan recall. Berikut ini merupakan hasil grafik pemodelan KNN menggunakan 3 skenario.



Gambar 4 Grafik Akurasi Pemodelan KNN

Pada gambar 4 menunjukkan perbandingan nilai akurasi dengan skenario pembagian dataset. Dari hasil tersebut menunjukkan bahwa jumlah data training mempengaruhi tingkat

performa, terlihat pada skenario pembagian dataset 80%-20% dengan nilai  $k=9$  sampai  $k=15$  mendapatkan nilai akurasi tertinggi dengan nilai mencapai 71% dan pada splitting dataset 20%:80% mendapatkan nilai akurasi terendah. Hal ini disebabkan karena semakin banyak data training akan menambah informasi dari karakteristik setiap kelas berita.

#### 4. KESIMPULAN

Dari hasil penelitian yang telah dilakukan dapat disimpulkan Pengklasifikasian berita *clickbait* dilakukan beberapa proses seperti *case folding*, *tokenizing*, *filtering*, *stemming*, *term weight* (TF-IDF) dan melakukan pemodelan dengan menggunakan algoritma *K-Nearest Neighbor* dan melakukan pemilihan nilai  $K$  untuk mendapatkan model terbaik. Rasio perbandingan data latih dan uji pada skenario 1 dengan rasio perbandingan sebesar 80%:20% memiliki performa paling tinggi, sehingga dapat disimpulkan bahwa jumlah data latih yang semakin banyak akan menambah ketepatan klasifikasi dikarenakan sistem akan banyak mendapatkan informasi dari data latih. Berdasarkan hasil uji coba parameter pada model pengklasifikasian dengan menggunakan algoritma *K-Nearest Neighbor* (KNN) menunjukkan bahwa model pengklasifikasian terbaik adalah model dengan nilai  $k=11$  pada skenario 1 yang menghasilkan nilai akurasi sebesar 71%, dengan di dapatkannya nilai akurasi terbaik maka dapat diketahui bahwa pengaruh *clickbait* terhadap tingkat pengunjung pada situs berita sangat berperan sehingga situs berita mendapatkan traffic cukup bagus untuk setiap berita-berita nya

#### 5. SARAN

Ada beberapa saran yang perlu diperhatikan untuk dapat melakukan penelitian lebih lanjut agar mendapatkan hasil yang lebih baik lagi. Diantaranya menambahkan jumlah dataset yang digunakan, dengan menambahkan jumlah dataset diharapkan mampu menambah jumlah dan keragaman informasi sehingga dapat menambah informasi dari metode *k-nearest neighbors* dan meningkatkan performa sistem. Menggunakan model yang beragam serta pemilihan parameter nilai  $K$  yang lebih banyak, agar semakin jelas seberapa berpengaruh parameter nilai  $K$  terhadap sistem. Menggunakan algoritma lainya untuk mendapatkan model pengklasifikasian terbaik yang dapat memungkinkan meningkatkan nilai akurasi yang rendah.

#### DAFTAR PUSTAKA

- [1] A. F. Yavi, "Klasifikasi Artikel Berbahasa Indonesia untuk Mendeteksi Clickbait menggunakan Metode Naïve Bayes," *J. Inf. Technol.*, vol. 06, no. 01, pp. 141–147, 2018.
- [2] M. Rizky and M. R. Kertanegara, "Penggunaan Clickbait Headline pada Situs Berita dan Gaya Hidup Muslim Dream . co . id," *Mediat. J. Komun.*, vol. 11, no. June 2017, pp. 31–43, 2018.
- [3] Y. Yamlean, "Clickbait Journalism Dan Pelanggaran Etika Jurnalistik (Studi Kasus Pelanggaran Etika Jurnalistik Dalam Praktik Clickbait Pada Media Online Jogja.Tribunnews.Com Periode 1 Maret 2019 - 30 April 2019)," Universitas mercu buana yogyakarta, 2019.
- [4] L. A. Utami, "Analisis Sentimen Opini Publik Berita Kebakaran Hutan Melalui Komparasi Algoritma Support Vector Machine Dan K-Nearest Neighbor Berbasis Particle Swarm Optimization," *J. Pilar Nusa Mandiri*, vol. 13, no. 1, pp. 103–112, 2017.
- [5] S. kartika Lidya, O. S. Sitompul, and S. Efendi, "Sentiment Analysis Pada Teks Bahasa Indonesia Menggunakan Support Vector Machine (Svm) Dan K-Nearest Neighbor (K-Nn)," *Semin. Nas. Teknol. Inf. dan Multimed. 2015*, no. maret, pp. 1–8, 2015.



- [6] A. A. Irfa and M. S. Mubarak, "Klasifikasi Topik Berita Berbahasa Indonesia Menggunakan k-Nearest Neighbor," *e-Proceeding Eng.*, vol. 5, no. 2, pp. 3631–3640, 2018.
  - [7] F. N. Rozi and D. H. Sulistyawati, "Klasifikasi Berita Hoax Pilpres Menggunakan Metode Modified K-Nearest Neighbor Dan Pembobotan Menggunakan Tf-Idf," *KONVERGENSI*, vol. 15, no. Januari, 2019.
  - [8] S. Nur and K. Fithriasari, "Klasifikasi Berita Online Menggunakan Metode Support Vector Machine dan K- Nearest," *J. Sains dan Seni ITS*, vol. 5, no. 2, pp. 2337–3520, 2016.
  - [9] A. R. Prasetyo, Idriati, and P. P. Adikara, "Klasifikasi Hoax Pada Berita Kesehatan Berbahasa Indonesia Dengan Menggunakan Metode Modified K-Nearest Neighbor," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 12, pp. 7466–7473, 2018.
  - [10] Z. U. Siregar, R. R. A. Siregar, and R. Arianto, "Klasifikasi Sentiment Analysis Pada Komentar Peserta Diklat Menggunakan Metode K-Nearest Neighbor," *J. Kilat*, vol. 8, no. 1, pp. 81–92, 2019.
  - [11] B. Herwijayanti, D. E. Ratnawati, and L. Muflikhah, "Klasifikasi Berita Online dengan menggunakan Pembobotan TF-IDF dan Cosine Similarity," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 1, pp. 306–312, 2018.
  - [12] A. Khoirunnisa, B. Irawan, and R. M Rumani, "Analisis dan implementasi perbandingan algoritma c.45 dengan naïve bayes untuk prediksi penawaran produk," *e-Proceeding Eng.*, vol. 3, no. 3, pp. 5029–5035, 2016.
-