FREQUENT ITEMSET MINING PADA ARTIKEL COVID-19 MENGGUNAKAN WEB CRAWLING DAN ALGORITMA FP-GROWTH

(*Frequent Itemset Mining On Covid-19 Articles Using Web Crawling And Fp-Growth Algorithm*)

Rizky Dwi Hadisaputro, I Gde Wirarama Wedashwara W.*, Ariyan Zubaidi
Dept Informatics Engineering, Mataram University
Jl. Majapahit 62, Mataram, Lombok NTB, Indonesia
Email: rizkydwihadisaputro@gmail.com [wirarama, ariyan13]@unram.ac.id

Abstract

Virus Corona COVID-19 merupakan penyakit yang telah menjadi pandemi di seluruh Dunia. Khususnya Indonesia yang berada di posisi 20 besar negara yang menyumbang kasus terbanyak COVID-19. Hal ini menyebabkan banyaknya pemberitaan tentang virus ini oleh berbagai media massa. Salah satu cara penyampaian informasi yang cukup populer adalah melalui portal berita daring. Dalam mengekstraksi kata yang mengandung dampak serta bahasan virus corona dapat menggunakan teknik data mining. Data mining akan memudahkan dalam mengekstraksi informasi yang bermanfaat dan pengetahuan terkait dari berbagai basis data besar. Dalam mendapatkan basis data berita yang besar pada penelitian ini digunakan teknik web. Hasil crawling selanjutnya akan diolah dan dicari kombinasi kata yang sering muncul atau dikenal dengan istilah frequent itemset. Teknik Frequent Patten Growth (FP-Growth) adalah salah satu algoritma dalam mencari frequent itemset yang merupakan pengembangan dari algoritma Apriori. Data yang digunakan sebanyak 7857 berita dari 10 kategori berita dengan kata kunci pencarian "Corona Indonesia". Nilai ambang batas yang digunakan untuk studi kasus ini berada pada nilai 0,8 untuk support dan 0,7 untuk confidence yang menghasilkan frequent itemset sebanyak 246869. Dalam penelitian ini strong rule association yang dihasilkan adalah kombinasi kata (Baca, Indonesia) dengan kata (Corona, Orang, Covid) yang memiliki nilai confidence 1,0, adapun untuk nilai rule terendah berada pada kombinasi kata (Baca, Indonesia, Video) dengan kata (Gambas, Laku, Corona, Sebar, Orang, Covid, Detik) dengan nilai confidence yang dihasilkan 0,8. Penelitian berkontribusi sebagai sistem terpadu untuk melakukan rangkaian web crawling hingga frequent item set mining untuk topik wabah COVID-19. Sistem yang telah dibangun dapat digunakan selanjutnya untuk menelusuri informasi tentang topik ini kedepannya.

Keywords: *Data Mining, Web Crawling, Corona, Frequent Itemset, FP-Growth*

**Penulis Korespondensi*

1. PENDAHULUAN

Pada bulan Desember 2019, bermunculan sejumlah kasus pneumonia dengan penyebab tidak diketahui yang memiliki gejala demam, rasa letih, batuk, dan kesulitan bernapas sebagai gejala utama terjadi di Wuhan dalam waktu singkat. Pemerintah Tiongkok dan departemen kesehatan di semua tingkat memberikan prioritas utama pada penyakit ini dan segera memberlakukan tindakan untuk pengendalian penyakit dan perawatan medis, dan mengarahkan lembaga penelitian untuk memulai investigasi, perawatan, dan kolaborasi penelitian. *World Health Organization* (WHO) menamakan virus itu 2019-nCoV sementara *International Committee on Taxonomy of Viruses* (ICTV) menyebutnya SARS-Cov-2; dan pneumonia yang disebabkan oleh infeksi virus disebut pneumonia *corona*virus baru (COVID-19) oleh WHO [1].

Sejak itu, terdapat banyak portal berita daring nasional yang merilis berita mengenai perkembangan virus *corona* di Indonesia maupun internasional. Salah satunya adalah situs berita detikcom. Sama seperti media cetak pada umumnya, berita daring disusun atas kata yang berisi informasi yang bernilai dan spesifik atas beberapa kategori. Kata atau term pada berita mengandung dampak serta bahasan yang berbeda tentang virus *corona* di Indonesia pada setiap kategori.

Dalam menggali informasi dampak serta bahasan yang terkait dengan *corona* Indonesia di berbagai kategori berita diperlukan teknik *data mining*. Adapun *data mining* adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan

machine learning untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai basis data besar [2]. Penggunaan *data mining* akan memudahkan dalam mengekstraksi informasi dalam berita sehingga kita dapat mengetahui *keyword* apa saja yang berhubungan dengan kata *corona* serta dampak corona di Indonesia di berbagai kategori. Pada proses untuk mengumpulkan berita yang akan melewati tahapan *data mining,* diperlukan teknik *web crawling.*

Web crawling sendiri merupakan sistem untuk mengunduh laman *web* secara massal. *Web crawling* juga digunakan untuk berbagai macam tujuan [3]. Umumnya seperti mesin pencari *web* dan pembuatan korpus laman *web*. Maka dari itu, *web crawling* dapat digunakan untuk mengumpulkan segala informasi terkait berita virus *corona* dalam bentuk korpus. *Web crawling* bekerja dengan menelusuri HTML pada sebuah situs dan memilih informasi apa saja yang akan dikumpulkan dengan mengacu pada *tag* pada berkas HTML.

Hasil *crawling* dari situs berita akan diolah dan dicari *term* yang paling sering muncul atau dikenal dengan istilah *frequent itemset*. Algoritma yang digunakan pada penelitian ini yakni algoritma *Frequent Pattern Growth* (*FP-Growth*). Pada algoritma *FP-Growth*, *generate candidate* tidak dilakukan seperti pada Apriori karena *FP-Growth* menggunakan konsep pembangunan *tree* dalam pencarian *frequent itemset*. *FP-Growth* menjadi lebih unggul daripada *Apriori* jika dilihat dari segi komputasi yang lebih cepat [4].

Pengumpulan data akan dilakukan menggunakan *web crawling* dan pengolahan data menggunakan algoritma *FP-Growth.* Keluaran dari penelitian ini berupa model yang mampu mengetahui keterkaitan antar term dengan *keyword* pencarian yakni *corona* pada *frequent itemset* serta frekuensi kata yang muncul di setiap kategori berita tertentu.

2. TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Telah dilakukan penelitian untuk penerapan sistem pengawasan infeksi penyakit melalui penambangan situs berita dan *Twitter* [5]. Penelitian dilakukan untuk mengembangkan teknik *text mining* untuk mengekstraksi informasi tentang penyakit menular dari *tweet* dan berita di media sosial. Metode yang digunakan adalah *Fuzzy Algorithm for Extraction, Monitoring, and Classification of Infectious Diseases*. Data yang digunakan adalah sebanyak 10.000 data yang terdiri dari berita dan *tweet*. Kemudian dilakukan *preprocessing* dan menghasilkan 1100 kata kunci dan 9 kelas. Hasil dari penelitian ini adalah nilai *recall* sebesar 88,41%.

Dilakukan juga penelitian untuk sistem ektraksi topik terkait penyakit melalui sumber berbasis internet berbasis *web* [6]. Sistem ini memantau topik penting terkait penyakit dan menyediakan informasi terkait penyakit tersebut. Dalam penelitian ini dilakukan penerapan Pemrosesan Bahasa Alami dan Algoritma *Ranking*. Pengumpulan data dilakukan sebanyak 100 data artikel dan *tweet* per jam dari pencarian melalui API *NAVER* dan *Twitter*.

Penelitian untuk memeriksa stigma terhadap penyakit Alzheimer di *Twitter* [7]. *Dataset* pada penelitian ini dikumpulkan berdasarkan 9 kata kunci dan menghasilkan 31.150 *tweet* terkait penyakit Alzheimer yang bersumber dari *Twitter* API. Dilakukan pemberian label tingkatan stigma sebanyak 6 pada *tweet* acak untuk data *training*. Hasil dari penelitian ini adalah mengidentifikasi 21,13% *tweet* yang menggunakan kata kunci penyakit Alzheimer berisi stigma publik yang negatif.

Penelitian untuk mengklasifikasikan *tweet* menggunakan fitur analisis sentimen dan pembobotan TF-IDF untuk memperbaiki deteksi tren penyakit flu [8]. Penelitian menggunakan berbagai algoritma pembelajaran mesin untuk mencari akurasi yang lebih baik. *Dataset* yang digunakan sebanyak 10.592 *tweet* yang terdiri atas 5.249 *tweet* yang berhubungan dengan flu dan 5.343 *tweet* yang tidak berhubungan dengan flu. Dari berbagai algoritma yang digunakan, klasifikasi menggunakan *Random Forest* menghasilkan akurasi 90,16% dan *F-Measure* sebesar 90,1%.

Penelitian untuk melakukan pencarian *frequent itemset* pada analisis keranjang belanja menggunakan algoritma *FP-Growth* [9]. Penelitian ini menggunakan dataset Supermarket dan pengolahan data menggunakan perangkat lunak Rapid Miner. Nilai *minimum support* ditetapkan sebesar 10% dan nilai *minimum confidence* sebesar 70% Hasil dari penelitian ini adalah menemukan korelasi beberapa produk seperti *beer wine spirit* dan *frozen foods* dengan *snack foods* dengan kekuatan korelasi 2.477.

Telah dilakukan pencarian *frequent itemset* dalam *association rule mining* menggunakan algoritma *Apriori* dengan studi kasus *supermarket* [10]. Dataset yang digunakan terdiri dari 12 atribut dan 108.131 transaksi. Penelitian ini menggunakan perangkat lunak Tanagra. Nilai *support* yang dihasilkan sebesar 15,489% dan nilai *confidence* yang dihasilkan sebesar 83,719% menghasilkan nilai rasio *lift* sebesar 2,47766.

Penelitian untuk menganalisa dan membandingkan metode algoritma *Apriori* dan *FP-Growth* untuk mencari pola daerah strategis pengenalan kampus studi kasus di STKIP Adzkia Padang [11]. Dalam penelitian ini digunakan data mahasiswa 1 angkatan dengan nilai *minimum support* 0,05% dan nilai *minimum confidence* 0,7%. Perangkat lunak yang digunakan untuk menunjang penelitian ini adalah Tanagra dan Rapid Miner. Hasil dari penelitian ini adalah terdapat 19 *association rule* dengan 8 *association rule* yang mempunyai kombinasi daerah di pesisir selatan.

Penelitian tentang teknik *data mining* untuk penentuan paket hemat sembako dan kebutuhan harian dengan menggunakan algoritma *FP-Growth* [12]. Penelitian ini dilakukan pada transaksi di *minimarket* Ulfamart dengan jumlah data transaksi sebanyak 60 data sampel. Nilai *minimum support* ditentukan sebesar 0,1 dan *minimum confidence* sebesar 0,5 menghasilkan 8 *rules.* Hasil dari penelitian ini mendapatkan 5 rekomendasi paket hemat dengan ketentuan paket yang telah ditentukan *minimarket* Ulfamart.

Berdasarkan berbagai penelitian yang telah dijelaskan sebelumnya, dapat disimpulkan bahwa metode *FP-Growth* memiliki hasil yang lebih baik dari algoritma *Apriori* dari segi komputasi. Oleh karena itu, penelitian untuk pencarian *frequent itemset* pada berita virus *corona* di Indonesia dapat dilakukan dengan menggunakan metode tersebut.

2.2 Teori Penunjang

2.2.1 Teks

Teks merupakan data tidak terstruktur yang disusun oleh kumpulan kata. Kata – kata perlu memiliki arti tersendiri serta disusun berdasarkan aturan tertentu untuk dapat membentuk sebuah teks. Aturan yang digunakan dalam penyusunan kata dalam teks ini disebut grammar [13].

2.2.2 *Web Crawling*

Web crawling atau *web scraping* merupakan istilah yang umum digunakan sebagai nama teknik maupun teknologi yang digunakan untuk mengumpulkan informasi yang tersedia untuk umum dari internet untuk tujuan tertentu. Informasi yang didapatkan dari internet sering kali berbeda, tetapi akan menjadi berharga bila dikumpulkan dalam satu paket menggunakan teknik ini. Salah satu kasus penggunaan web crawling adalah menentukan bagaimana perasaan orang tentang subjek tertentu, yang dikenal sebagai analisis sentimen [14].

2.2.3 *Data Mining*

Data Mining merupakan suatu proses otomatis atau semi otomatis untuk menemukan informasi (*knowledge*) baru yang memiliki potensi dari sekumpulan data [15]. Secara sederhana *data mining* adalah penambangan atau penemuan informasi baru dengan mencari pola atau aturan tertentu dari sejumlah data yang sangat besar. *Data mining* juga disebut sebagai serangkaian proses untuk menggali nilai tambah berupa pengetahuan yang selama ini tidak diketahui secara manual dari suatu kumpulan data. *Data mining* sering juga disebut sebagai *knowledge discovery in database* (KDD). KDD adalah kegiatan yang meliputi pengumpulan, pemakaian data, historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar.

2.2.4 *Text preprocessing*

Text preprocessing merupakan proses untuk mentransformasikan teks ke dalam kumpulan kata. Teks merupakan data yang tidak terstruktur, yang mana hampir tidak mungkin bentuk raw-nya untuk diproses langsung menggunakan program komputer. Operasi numerik pun tidak dapat diaplikasikan pada data teks. Oleh karena itu, perlu dilakukan preprocessing pada teks untuk mendapatkan data yang dapat diolah menggunakan komputer. Terdapat 3 langkah mendasar yang dilakukan dalam text preprocessing, yaitu tokenization, stemming, dan stop-word removal [13].

a. *Tokenization*

Tokenization adalah proses untuk memotong teks menjadi kata/*token* yang dipisahkan oleh spasi atau tanda baca. Proses *tokenization* menerima teks sebagai *input* dan menghasilkan kumpulan *token* sebagai *output*. Selanjutnya, *token* yang mengandung karakter spesial atau angka akan dihilangkan, lalu token akan diubah menjadi *lowercase* [13].

b. *Stemming*

Proses selanjutnya *dalam text preprocessing* adalah *stemming*. Pada tahap ini, *token* yang didapatkan dari proses *tokenization* diubah menjadi bentuk dasarnya. Proses *stemming* biasanya dilakukan pada kata benda, kata kerja, dan kata sifat [13].

c. *Stop-word removal*

Pada proses *stop-word removal*, dilakukan penghapusan *stop-word dari* daftar *token* atau kata yang sudah diproses dengan tahap *stemming*. *Stop-*

word merupakan kata yang tidak berhubungan dengan konteks dari teks, sehingga perlu dihilangkan untuk meningkatkan efisiensi dari proses *training* atau klasifikasi [13]. Contoh dari *stop-word* dalam bahasa Indonesia adalah "di" dan "ke". Kata – kata tersebut tidak dapat mewakili konteks dari dokumen karena terdapat pada hampir seluruh dokumen [13].

2.2.5 *Association Rule*

Analisis asosiasi atau *association rule mining* adalah teknik *Data Mining untuk* menemukan aturan asosiatif antara suatu kombinasi *item* [11]. Algoritma aturan asosiasi akan menggunakan data latihan, sesuai dengan pengertian *Data Mining*, untuk menghasilkan pengetahuan. Pengetahuan untuk mengetahui *item-item* belanja yang sering dibeli secara bersamaan dalam suatu waktu. Aturan asosiasi yang berbentuk "*if…then…*" atau "jika…maka…" merupakan pengetahuan yang dihasilkan dari fungsi aturan asosiasi. Aturan ini dihitung dari data yang sifatnya probabilistik.

Metodologi dasar analisis asosiasi terbagi menjadi dua tahap:

a. Analisa pola frekuensi tinggi

Tahap ini mencari kombinasi item yang memenuhi syarat minimum dari nilai *support* dalam *database*. Nilai *support* sebuah *item* diperoleh dengan Persamaan (1)

$$Support\,(\mathrm{X}) = \frac{\sum transaksi\ mengandung\ X}{\sum transaksi} \quad (1)$$

Kemudian, untuk mendapatkan nilai *support* dari dua *item* diperoleh dengan Persamaan (2)

$$Support(\mathrm{X,Y}) = \mathrm{P}(\mathrm{X} \cap \mathrm{Y})$$
$$Support(\mathrm{X,Y}) = \frac{\sum transaksi\ mengandung\ X\ dan\ Y}{\sum transaksi} \quad (2)$$

b. Pembentukan aturan asosiasi

Setelah semua pola frekuensi tinggi ditemukan, barulah dicari aturan asosiatif yang memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan asosiasi "*jika A maka B*". Nilai *confidence* dari aturan "*jika A maka B*" diperoleh dengan Persamaan (3)

$$Confidence = \mathrm{P}(\mathrm{Y} \mid \mathrm{X})$$

$$Confidence\,(\mathrm{X,Y}) = \frac{\sum transaksi\ mengandung\ X\ dan\ Y}{\sum transaksi\ mengandung\ X} \quad (3)$$

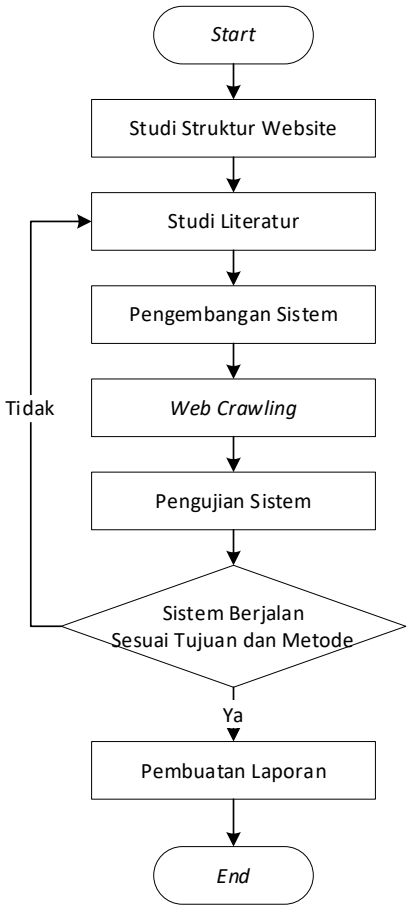
2.2.6 *Frequent Pattern Growth* (*FP-Growth*)

Algoritma *FP-Growth* merupakan pengembangan dari algoritma *Apriori*. Algoritma *Frequent Pattern Growth* adalah salah satu alternatif algoritma yang dapat digunakan untuk menentukan himpunan data yang paling sering muncul (*frequent itemset*) dalam sebuah kumpulan data [16]. Pengembangan dari algoritma *Apriori* ini terletak dalam *scanning database* dan akurasi *rules*-nya. FP- *Growth* lebih memberikan keuntungan karena hanya dilakukan satu atau dua kali saja *scanning database* sedangkan *Apriori* perlu melakukan *scanning database* berulang ulang.

Pada algoritma *FP-Growth* menggunakan konsep pembangunan *tree*, yang biasa disebut FP-*Tree*, dalam pencarian *frequent itemsets* bukan menggunakan *generate candidate* seperti yang dilakukan pada algoritma *Apriori*. Dengan menggunakan konsep tersebut, algoritma *FP-Growth* menjadi lebih cepat daripada algoritma *Apriori.*

3. METODE PENELITIAN

3.1 Diagram Alir Penelitian



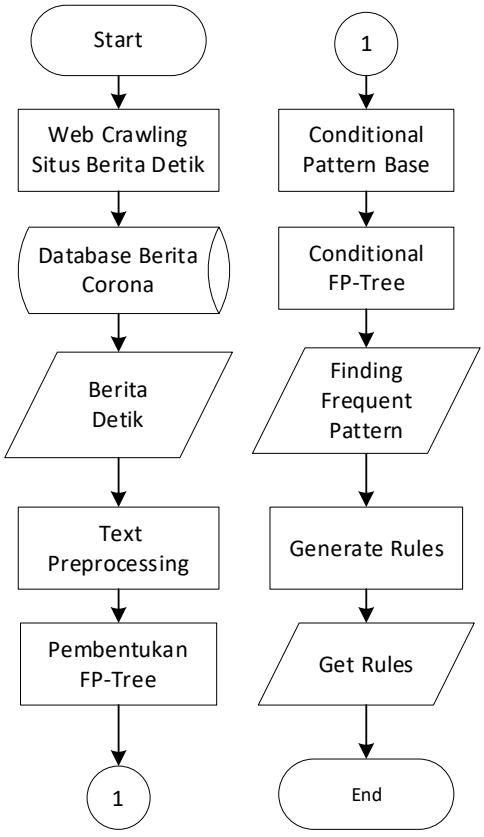
Gambar 1. Diagram Alir Penelitian.

3.2 Pengumpulan Data

Bahan penelitian yang digunakan dalam penelitian tentang *frequent itemset mining* pada artikel *corona* menggunakan *web crawling* dan algoritma *FP-Growth* ini adalah berita daring tentang virus *corona* di Indonesia yang dikumpulkan dari situs daring *www.detik.com*. Dari 10.000 data berita yang terkumpul, terdapat 105 kategori berita yang didapatkan dan hanya 10 kategori dengan berita terbanyak yang digunakan sebagai *dataset*.

3.3 Rancangan Sistem

Rancangan dari sistem *frequent itemset mining* pada artikel *corona* menggunakan *web crawling* dan algoritma *FP-Growth* terdiri dari beberapa tahapan, yang dapat dilihat pada Gambar 4.



Gambar 2. Alur Pembangunan Sistem

3.3.1 *Web Crawling* Situs Berita

Berita dikumpulkan menggunakan kata kunci "*Corona* Indonesia" pada mesin pencari Detik. Dari hasil pencarian ditemukan sebanyak 105 subdomain yang mewakili kategori dari berita tersebut, kemudian masing-masing *link* berita diberikan label sesuai dengan *subdomain*-nya. Berdasarkan jumlah berita terbanyak, diambil 10 *subdomain* dengan jumlah 7857 berita.

3.3.2 *Input* Berita

Pada tahap ini, berita yang telah dikumpulkan dari situs berita Detik telah diberi label kategori sebelum dimasukan ke dalam sistem untuk diproses.

3.3.3 *Text Preprocessing*

Pada penelitian ini dilakukan 3 tahap *preprocessing* untuk mentransformasikan berita utuh menjadi *terms* yang akan digunakan pada proses selanjutnya.

a. *Tokenization*

TABEL I. CONTOH BERITA HASIL *TOKENIZATION*.

Teks Berita	Hasil *Tokenization*
Berbagai pembatasan mulai dari penentuan jarak aman, aktivitas di luar rumah, hingga anjuran untuk tidak berkerumun semuanya ditetapkan demi menekan penyebaran coronavirus yang masih sulit dikendalikan.	berbagai, pembatasan, mulai, dari, penentuan, jarak, aman, aktivitas, di, luar, rumah, hingga, anjuran, untuk, tidak, berkerumun, semuanya, ditetapkan, demi, menekan, penyebaran, coronavirus, yang, masih, sulit, dikendalikan

Berita utuh dipecah menjadi kata per kata. Semua kata juga mengalami *case folding* dan selain huruf dihilangkan dan dianggap *delimiter*.

b. *Stemming*

TABEL II. CONTOH BERITA HASIL *STEMMING*.

Hasil *Tokenization*	Hasil *Stemming*
berbagai, pembatasan, mulai, dari, penentuan, jarak, aman, aktivitas, di, luar, rumah, hingga, anjuran, untuk, tidak, berkerumun, semuanya, ditetapkan, demi, menekan, penyebaran, coronavirus, yang, masih, sulit, dikendalikan	bagai, batas, mulai, dari, tentu, jarak, aman, aktivitas, di, luar, rumah, hingga, anjur, untuk, tidak, kerumun, semua, tetap, demi, tekan, sebar, coronavirus, yang, masih, sulit, kendali

Hasil dari *tokenization* berupa kata diubah menjadi kata dasar dengan menghilangkan imbuhan. Algoritma yang digunakan adalah algoritma Nazief dan Adriani.

c. *Stopword Removal*

TABEL III. CONTOH BERITA HASIL *STOPWORD REMOVAL.*

Hasil *Stemming*	Hasil *Stopword Removal*
bagai, batas, mulai, dari, tentu, jarak, aman, aktivitas, di, luar, rumah, hingga, anjur, untuk, tidak, kerumun, semua, tetap, demi, tekan, sebar, coronavirus, yang, masih, sulit, kendali	bagai, batas, mulai, tentu, jarak, aman, aktivitas, luar, rumah, hingga, anjur, tidak, kerumun, semua, tetap, tekan, sebar, coronavirus, masih, sulit, kendali

Semua kata hasil dari proses *stemming* yang merupakan *stopword* (kata umum) dihilangkan. Hal ini dilakukan untuk meningkatkan efisiensi beban komputasi. Algoritma yang digunakan pada tahap ini adalah algoritma Sastrawi.

3.3.4 *Frequent Pattern Growth* (FP-*Growth*)

Tahap pertama dalam algoritma ini adalah menyiapkan dataset. Berikut contoh data hasil *preprocessing* terkait berita corona di setiap kategori.

TABEL IV. BERITA HASIL *PREPROCESSING* DI BERBAGAI KATEGORI

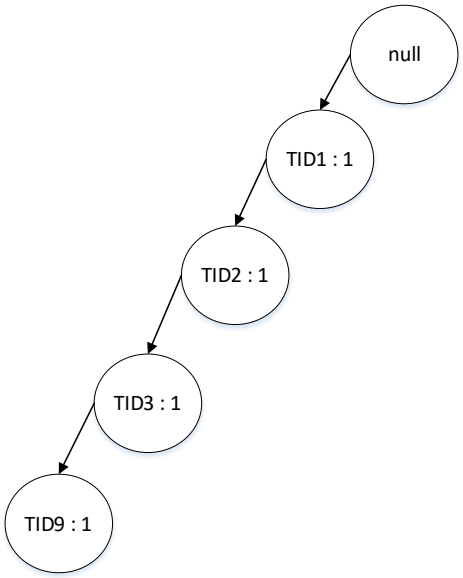
No	Kategori Berita	Daftar Kata
1	Berita Umum	pemerintah, positif, kasus, isolasi
2	Kesehatan	pemerintah, positif, masker, prokes, sehat, sosialisasi
3	Ekonomi dan Bisnis	*corona*, virus, isolasi, lockdown
4	Travel	pemerintah, *corona*, virus, sehat
5	Kuliner	positif, kasus, masker, prokes
6	Industri	pemerintah, positif, kasus, sosialisasi
7	Pendidikan	pemerintah, positif, kasus, masker, prokes, lockdown
8	Sepakbola	kasus, *corona*, virus, sehat
9	Teknologi	positif, masker, prokes, *corona*, virus, sosialisasi
10	Hiburan	pemerintah, masker, prokes, isolasi, lockdown

Tahap kedua mendaftarkan seluruh kata beserta frekuensinya di keseluruhan kategori. Tahapan ini disebut juga pencarian *frequent itemset.* Data diurutkan berdasarkan jumlah frekuensi terbesar dengan ketentuan *nilai minimum support* sebanyak 2.

TABEL V. DAFTAR KATA SERTA FREKUENSI KEMUNCULANNYA.

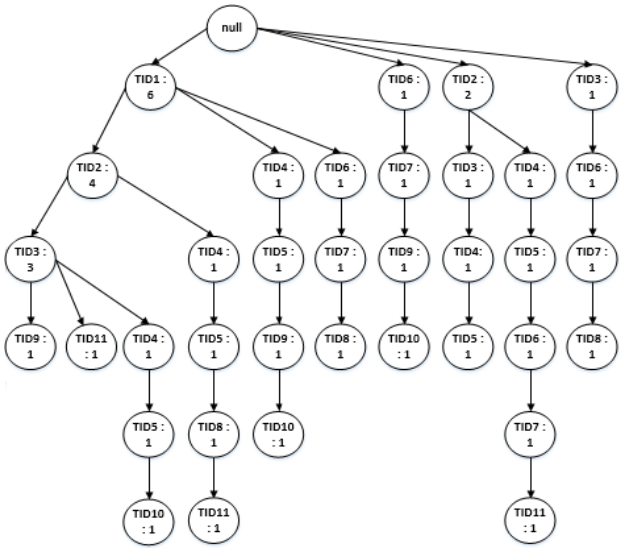
No	*itemset*	*node*	*Support count*
1	pemerintah	TID1	6
2	positif	TID2	6
3	kasus	TID3	5
4	masker	TID4	5
5	prokes	TID5	5
6	*corona*	TID6	4
7	virus	TID7	4
8	sehat	TID8	3
9	isolasi	TID9	3
10	lockdown	TID10	3
11	sosialisasi	TID11	3

Tahap ketiga adalah membangun *frequent pattern tree*. Pada tahap ini hasil dari Tabel V dengan *minimum support* 3 di implementasikan dalam bentuk *tree* berdasarkan urutan frekuensi dari terbesar ke terkecil. Berikut ini contoh *tree* yang terbentuk dari Berita Umum dengan daftar kata pemerintah, positif, kasus dan isolasi.



Gambar 3. Proses pembentukan *FP-Tree* pada Berita Umum

Jika terdapat kata sudah terdapat dalam *tree* maka nilai setelah *node* akan bertambah. *Child* baru akan terbentuk apabila terdapat kata yang sebelumnya belum ada.



Gambar 4. Hasil penerapan seluruh berita pada *tree*

Tahap keempat adalah membuat *conditional pattern base*. Pembangkitan *conditional pattern base* didapatkan melalui hasil FP-*Tree*, dengan mencari *support count* terkecil sesuai dengan hasil pengurutan prioritas.

TABEL V. DAFTAR *CONDITIONAL PATTERN BASE*.

Item	*Conditional Pattern Base*
TID11	{{TID1 TID2 TID3 : 1}, {TID1 TID2 TID4 TID5 TID8 : 1}, {TID2 TID4 TID5 TID6 TID7 : 1}}
TID10	{{TID1 TID2 TID3 TID4 TID5 : 1}, {TID1 TID4 TID5 TID9 : 1}, {TID6 TID7 TID9 : 1}}
TID9	{{TID1 TID2 TID3 : 1}, {TID1 TID4 TID5 : 1}, {TID6 TID7 : 1}}
TID8	{{TID1 TID2 TID4 TID5 : 1}, {TID1 TID6 TID7 : 1}, {TID3 TID6 TID7 : 1}}
TID7	{{TID1 TID6: 1}, {TID6 : 1}, {TID2 TID4 TID5 TID6 : 1}, {TID3 TID6 : 1}}
TID6	{{TID1 : 1}, {TID2 TID4 TID5 : 1}, {TID3 : 1}}
TID5	{{TID1 TID2 TID3 TID4 : 1}, {TID1 TID2 TID4 : 1}, {TID1 TID4 : 1}, {TID2 TID3 TID4 : 1}, {TID2 TID4 : 1}}
TID4	{{TID1 TID2 TID3 : 1}, {TID1 TID2 : 1}, {TID1 : 1}, {TID2 TID3 : 1}, {TID2 : 1}}
TID3	{{TID1 TID2 : 3}, {TID2 : 1}}
TID2	{TID1 : 4}

Pada tahap ini, *support count* dari setiap *item* pada setiap *conditional pattern base* dijumlahkan, lalu setiap *item* yang memiliki *support count* lebih besar atau sama dengan *minimum support* akan dibangkitkan dengan *conditional* FP-*Tree*.

Tahap kelima adalah membuat *conditional fp-tree*. Tahap ini dilakukan dengan cara melihat frekuensi item dari tiap *conditional pattern base*.

TABEL VI. DAFTAR *CONDITIONAL FP-TREE*.

Item	*Conditional Pattern Base*	*Conditional FP-Tree*
TID11	{{TID1 TID2 TID3 : 1}, {TID1 TID2 TID4 TID5 TID8 : 1}, {TID2 TID4 TID5 TID6 TID7 : 1}}	<TID1 : 2, TID2 : 2>
TID10	{{TID1 TID2 TID3 TID4 TID5 : 1}, {TID1 TID4 TID5 TID9 : 1}, {TID6 TID7 TID9 : 1}}	<TID1 : 2>
TID9	{{TID1 TID2 TID3 : 1}, {TID1 TID4 TID5 : 1}, {TID6 TID7 : 1}}	<TID1 : 2>
TID8	{{TID1 TID2 TID4 TID5 : 1}, {TID1 TID7 TID6 : 1}, {TID3 TID6 TID7 : 1}}	<TID1 : 2>
TID7	{{TID1 : 1}, {TID6 : 1}, {TID2 TID4 TID5 TID6 : 1}, {TID3 TID6 : 1}}	-
TID6	{{TID1 TID7 : 1}, {TID2 TID4 TID5 : 1}, {TID3 : 1}}	-
TID5	{{TID1 TID2 TID3 TID4 : 1}, {TID1 TID2 TID4 : 1}, {TID1 TID4 : 1}, {TID2 TID3 TID4 : 1}, {TID2 TID4 : 1}}	<TID1 : 3, TID2 : 2, TID4 : 3>, <TID2 : 2, TID4 :2>
TID4	{{TID1 TID2 TID3 : 1}, {TID1 TID2 : 1}, {TID1 : 1}, {TID2 TID3 : 1}, {TID2 : 1}}	<TID1 : 3, TID2 : 2>, <TID2 : 2>
TID3	{{TID1 TID2 : 3}, {TID2 : 1}}	<TID2 : 3>
TID2	{TID1 : 4}	<TID1 : 4>

Pada tahap ini frekuensi item yang muncul dalam *conditional pattern base* dengan jumlah lebih dari nilai *support* yakni 2, maka akan masuk ke dalam *conditional fp-tree*.

Tahap keenam adalah membuat *frequent pattern*. Apabila *conditional fp-tree* merupakan lintasan tunggal (*single path*), maka didapatkan *frequent pattern* dengan melakukan kombinasi ke item untuk setiap *conditional fp-tree.*

TABEL VI. DAFTAR *FREQUENT PATTERN*.

Item	*Conditional FP-Tree*	*Frequent Pattern*
TID11	<TID1 : 2, TID2 : 2>	{TID1, TID11 : 2}, {TID2, TID11 : 2} , {TID1,TID2,TID11 :2}

TID10	<TID1 : 2, TID4 : 2, TID5 : 2>	{TID1, TID10 : 2}, {TID4, TID10 :2}, {TID5, TID10 :2}, {TID1, TID4, TID10 : 2}, {TID1, TID5, TID10 :2}, {TID4, TID5, TID10 :2}, {TID1, TID4, TID5, TID10 : 2}
TID9	<TID1 : 2>	{TID1, TID9 :2}
TID8	<TID1 : 2>	{TID1, TID8 :2}
TID7	-	-
TID6	-	-
TID5	<TID1 : 3, TID2 : 2, TID4 : 3>, <TID2 : 2, TID4 :2>	{TID1, TID5 : 3}, {TID2, TID5 : 4}, {TID4, TID5 : 5}, {TID1, TID2, TID5 : 2}, {TID1, TID4, TID5 : 3}, {TID2, TID4, TID5 : 4}, {TID1, TID2, TID4, TID5 : 2}
TID4	<TID1 : 3, TID2 : 2>, <TID2 : 2>	{TID1, TID4 : 3}, {TID2, TID4 :4}, {TID1, TID2, TID4 : 2}
TID3	<TID2 : 3>	{TID2, TID3 : 3}
TID2	<TID1 : 4>	{TID1, TID2 : 4}

Tahap terakhir adalah menentukan nilai *support* dan *confidence*. Nilai *support* digunakan untuk menentukan persentase jumlah transaksi yang mengandung item. Sedangkan nilai *confidence* digunakan dalam menentukan *strong rule association*.

TABEL VI. HASIL PERHITUNGAN SUPPORT DAN CONFIDENCE DARI FREQUENT PATTERN.

Aturan Asosiasi	*Support*	*Confidence*
(Pemerintah) => (Sosialisasi)	0,2	0,67
(Positif) => (Sosialisasi)	0,2	0,67
(Pemerintah, Positif) => (Sosialisasi)	0,2	0,67
(Pemerintah) => (Lockdown)	0,2	0,67
(Masker) => (Lockdown)	0,2	0,67
(Prokes) => (Lockdown)	0,2	0,67
(Pemerintah, Masker) => (Lockdown)	0,2	0,67
(Pemerintah, Prokes) => (Lockdown)	0,2	0,67
(Masker, Prokes) => (Lockdown)	0,2	0,67
(Pemerintah, Masker, Prokes) => (Lockdown)	0,2	0,67
(Pemerintah) => (Isolasi)	0,2	0,67
(Pemerintah) => (Sehat)	0,2	0,67
(Pemerintah) => (Prokes)	0,3	0,6
(Positif) => (Prokes)	0,4	0,8
(Masker) => (Prokes)	0,5	1
(Pemerintah, Positif) => (Prokes)	0,2	0,4
(Pemerintah, Masker) => (Prokes)	0,3	0,6
(Positif, Masker) => (Prokes)	0,4	0,8
(Pemerintah, Positif, Masker) => (Prokes)	0,2	0,4
(Pemerintah) => (Masker)	0,3	0,6
(Positif) => (Masker)	0,4	0,8
(Pemerintah, Positif) => (Masker)	0,2	0,4
(Positif) => (Kasus)	0,3	0,6
(Pemerintah) => (Positif)	0,4	0,67

Berdasarkan hasil Tabel VI, dapat dilihat bahwa aturan asosiasi dengan nilai *confidence* tertinggi sebesar 1 adalah jika terdapat item "Masker" maka kemunculannya akan bersamaan dengan item "Prokes". Aturan lainnya dengan nilai *confidence* sebesar 0,8 berturut-turut adalah jika terdapat item "Positif" maka kemunculannya akan bersamaan dengan item "Prokes", jika terdapat item "Positif" dan "Masker" maka kemunculannya akan bersamaan dengan item "Prokes" dan jika terdapat item "Positif" maka kemunculannya akan bersamaan dengan item "Masker".

4. HASIL DAN PEMBAHASAN

4.1 Pembangunan *Associaton Rule*

Dalam proses membangun *association rule* dibutuhkan data yang sesuai agar *rule* yang didapatkan bisa lebih baik. Pada proses sebelumnya dilakukan proses *crawling* yang menghasilkan berita dari 10 kategori dengan total 7857 berita dan setelah melewati proses *preprocessing* didapatkan 1513 kombinasi kata.

TABEL VII. DAFTAR KATA HASIL CRAWLING DAN PREPROCESSING.

No.	Kata	Frekuensi	*Support*
1.	indonesia	79	100%
2.	corona	61	100%
3.	covid	53	100%
4.	vaksin	42	50%
5.	makan	41	60%
…	…	…	…
1509.	yogyakarta	1	10%
1510.	youtube	1	10%
1511.	yuliani	1	10%
1512.	yulianto	1	10%
1513.	yusran	1	10%

Hasil kata pada Tabel VII didapatkan dari menggabungkan 7587 berita dari 10 kategori. Setelah menghitung frekuensi tahapan selanjutnya dalam

algoritma FP-*Growth* yakni membuat *frequent itemsets* dari data yang ada pada Tabel VII.

TABEL VIII. HASIL PEMBUATAN FREQUENT ITEMSETS.

No.	*Itemsets*	*Support*
1.	baca, indonesia, corona, orang, covid	1,0
2.	corona, orang, indonesia, covid	1,0
3.	orang, baca, indonesia	1,0
4.	corona, indonesia, covid	1,0
5.	corona, baca, indonesia	1,0
6.	corona, orang, baca, indonesia	1,0
7.	jakarta, indonesia, covid	0,9
8.	detik, corona, indonesia, covid	0,9
9.	corona, baca, indonesia, banyak	0,9
10.	baca, indonesia, covid, banyak	0,9
11.	corona, baca, covid, banyak	0,9
12.	corona, orang, gambas, covid	0,9
13.	corona, indonesia, gambas, covid	0,9
14.	baca, indonesia, video, gambas, corona, orang, covid, detik	0,9
15.	indonesia, gambas, banyak, detik, orang	0,8
16.	gambas, banyak, corona, orang, covid	0,8
17.	baca, orang, covid, detik, Jakarta, gambas	0,8
18.	indonesia, gambas, banyak, corona, orang, covid	0,8
19.	baca, indonesia, banyak, corona, bagai, orang, covid	0,8
20.	baca, indonesia, video, gambas, laku, corona, sebar, orang, covid, detik	0,8

Dilakukan pembuatan *frequent itemsets* dengan penerapan nilai *minimum support* yakni 0,8. Nilai *support* adalah persentase kombinasi item tersebut dalam kumpulan data.

Setelah nilai *support*, selanjutnya adalah nilai *confidence* yang merupakan nilai yang menunjukkan kuatnya hubungan antar item dalam *association rule*. Dalam penelitian ini nilai *minimum confidence* yang digunakan sebesar 0,7.

TABEL IX. HASIL PEMBUATAN FREQUENT ITEMSETS.

No	*Antecedents*	*Consequents*	*Confidence*
1	baca, indonesia	corona, orang, covid	1,0
2	corona, orang, indonesia	covid	1,0
3	orang, baca	indonesia	1,0
4	corona, indonesia	covid	1,0
5	corona, baca	indonesia	1,0
6	corona, orang	baca, indonesia	1,0
7	jakarta	indonesia, covid	1,0
8	detik, corona, indonesia	covid	1,0
9	corona, baca	indonesia, banyak	0,9
10	baca, indonesia	covid, banyak	0,9
11	corona, baca	covid, banyak	0,9
12	corona, orang, gambas	covid	1,0
13	corona, indonesia	gambas, covid	0,9
14	baca, indonesia, video	gambas, corona, orang, covid, detik	1,0
15	indonesia, gambas, banyak	detik, orang	1,0
16	gambas, banyak, corona, orang	covid	1,0
17	baca, orang, covid, detik, jakarta	gambas	1,0
18	indonesia, gambas, banyak, corona, orang	covid	1,0
19	baca, indonesia, banyak, corona, bagai	orang, covid	1,0
20	baca, indonesia, video	gambas, laku, corona, sebar, orang, covid, detik	0,8

Tahapan ini menghasilkan *rules* dari proses yang telah dilakukan dengan panjang bervariasi mulai dari 2-11 kata. Adapun untuk nilai maksimal yang didapatkan yakni dengan nilai *support* 1,0 dan *confidence* 1,0, salah satunya adalah *rule* (Baca, Indonesia)=>(Corona, Orang Covid). Sedangkan untuk nilai *minimum* yang didapatkan yakni dengan nilai *support* 0,8 dan *confidence* 0,8, salah satunya adalah *rule* (Baca, Indonesia, Video)=>(Gambas, Laku, Corona, Sebar, Orang, Covid, Detik).

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang sudah dilakukan, dapat disimpulkan bahwa:

1. Dari hasil *crawling* artikel corona indonesia di *website* Detik.com yang telah dilakukan menghasilkan 7857 berita yang terdiri dari 10 kategori berita. Adapun kategori berita dengan

berita terbanyak yakni pada kategori Berita Umum.

2. Dari hasil penelitian yang telah dilakukan nilai ambang batas yang digunakan untuk studi kasus ini berada pada nilai 0,8 untuk *support* dan 0,7 untuk *confidence* yang menghasilkan *frequent itemset* sebanyak 246869.
3. Dalam penelitian ini *rule* yang paling kuat yang dihasilkan adalah kombinasi kata (Baca, Indonesia) dengan kata (Corona, Orang, Covid) yang memiliki nilai *confidence* 1,0, adapun untuk nilai rule terendah berada pada kombinasi kata (Baca, Indonesia, Video) dengan kata (Gambas, Laku, Corona, Sebar, Orang, Covid, Detik) dengan nilai *confidence* yang dihasilkan 0,8.
4. Dalam menjalankan perhitungan terdapat kendala perangkat keras dalam menentukan *frequent itemset* dengan *support* dibawah 0,8.

5.2 Saran

Berdasarkan hasil penelitian yang sudah didapatkan kemudian terdapat beberapa catatan saran untuk dapat diperbaiki serta dikembangkan pada penelitian serupa yaitu:

1. Penelitian dengan menggunakan algoritma FP-*Growth* dengan data yang berjumlah banyak lebih disarankan untuk melakukan komputasi pada *device* yang memiliki prosesor yang tinggi dengan RAM yang lebih besar dari pada yang digunakan peneliti.
2. Mencoba *frequent itemset mining* menggunakan PySpark atau *platform* pemrosesan paralel lainnya.

DAFTAR PUSTAKA

[1] W. Zhou, *Buku Panduan Pencegahan Coronavirus 101 Tips Berbasis Sains Yang Dapat Menyelamatkan Hidup Anda*. Wuhan: Skyhorse Publishing, 2020. [Online]. Available: https://fin.co.id/wp-content/uploads/2020/03/Buku-Panduan-Pencegahan-Coronavirus-101-Tips-Berbasis-Sains.pdf

[2] E. Turban, J. E. Aronson, and T.-P. Liang, *Decision Support Systems and Intelligent Systems*. 2008. doi: 10.1002/9780470755891.ch11.

[3] C. Olston and M. Najork, "Web Crawling," *Found. Trends Inf. Retr.*, vol. 4, no. 3, p. 86, 2010, [Online]. Available: https://books.google.co.id/books?hl=en&lr=&id=CRS_GMjF5gwC&oi=fnd&pg=PA1&dq=web+crawling&ots=_mYioi27TM&sig=MEOaRVIrHqLx55u96he5ZnvO080&redir_esc=y#v=onepage&q=web crawling&f=false

[4] D. Widiastuti and N. Sofi, "Analisis Perbandingan Algoritma Apriori Dan Fp-Growth Pada Transaksi Koperasi," *UG J. Vol.*, vol. 8, no. 01, pp. 21–24, 2014.

[5] A. S. Jahanbin, K., Rahmanian, F., Rahmanian, V., & Jahromi, "Application of Twitter and web news mining in infectious disease surveillance systems and prospects for public health Anwendung von Twitter und Web News Mining in Überwachungssystemen für Infektionskrankheiten und Perspektiven der öffentlichen Gesundheit," *GMS Hyg. Infect. Control*, vol. 14, pp. 1–12, 2019.

[6] J. Yoon, J. W. Kim, and B. Jang, "DiTeX: Disease-related topic extraction system through internet-based sources," *PLoS One*, vol. 13, no. 8, pp. 1–16, 2018, doi: 10.1371/journal.pone.0201933.

[7] N. Oscar, P. A. Fox, R. Croucher, R. Wernick, J. Keune, and K. Hooker, "Machine learning, sentiment analysis, and tweets: An examination of Alzheimer's disease stigma on Twitter," *Journals Gerontol. - Ser. B Psychol. Sci. Soc. Sci.*, vol. 72, no. 5, pp. 742–751, 2017, doi: 10.1093/geronb/gbx014.

[8] A. Alessa and M. Faezipour, "Tweet Classification Using Sentiment Analysis Features and TF-IDF Weighting for Improved Flu Trend Detection," vol. 2, no. Cdc, pp. 174–186, 2018, doi: 10.1007/978-3-319-96136-1.

[9] L. I. Prahartiwi, "Pencarian Frequent Itemset pada Analisis Keranjang Belanja Menggunakan Algoritma FP-Growth," *Inf. Syst. Educ. Prof.*, vol. 2, no. 1, pp. 1–10, 2017.

[10] L. I. Prahartiwi and W. Dari, "Algoritma Apriori untuk Pencarian Frequent itemset dalam Association Rule Mining," *PIKSEL Penelit. Ilmu Komput. Sist. Embed. Log.*, vol. 7, no. 2, pp. 143–152, 2019, doi: 10.33558/piksel.v7i2.1817.

[11] D. Sepri and M. Afdal, "Analisa Dan Perbandingan Metode Algoritma Apriori Dan Fp-Growth Untuk Mencari Pola Daerah Strategis," *J. Sist. Inf. Kaputama*, vol. 1, no. 1, pp. 47–55, 2017.

[12] A. Nastuti, "Teknik Data Mining Untuk Penentuan Paket Hemat Sembako Dan Kebutuhan Harian Dengan Menggunakan Algoritma Fp-Growth (Studi Kasus Di Ulfamart Lubuk Alung)," *Inform. J. Ilm. Fak. Sains dan Teknol. Univ. Labuhanbatu*, vol. 7, no. 3, pp. 111–119, 2019.

[13] T. Jo, *Text Mining : Concept, Implementation and Big Data Challenge*, no. 2. Seoul: Springer, 2019. doi: 10.17308/sait.2020.2/2924.

[14] V. Smith, *Go Web Scraping Quick Start Guide*. Birmingham: Packt Publishing, 2019.

[15] F. Kurniasih, N. Kumaladewi, and L. Katjong, "Analisa Dan Perancangan Data Mining Dengan

Metode Market Basket Analysis Untuk Analisa Pola Belanja Konsumen pada Tendencies Store," *Sist. Inf.*, vol. 5, no. 1, pp. 1–10, 2012, [Online]. Available: http://journal.uinjkt.ac.id/index.php/sisteminformasi/article/view/280

[16] G. Djati *et al.*, "Algoritma Frequent Pattern Growth pada Sistem Rekomendasi Film," vol. 3, pp. 19–24, 2021.