

FRECOMTWEET: APLIKASI REKOMENDASI PRODUK MENGGUNAKAN KEDEKATAN PERTEMANAN PADA TWITTER

Ratih NE Anggraini, Ainatul Maulida, Daniel Oranova Siahaan

Teknik Informatika, Institut Teknologi Sepuluh Nopember
Jalan Teknik Kimia, Kampus ITS, Sukolilo, Surabaya
e-mail: ratih_nea@if.its.ac.id

ABSTRAK

Perkembangan teknologi informasi dan komunikasi saat ini memberikan kemudahan seseorang untuk saling berinteraksi. Kemudahan ini dimanfaatkan untuk bertukar pikiran, seperti menggunakan sosial media Twitter sebagai rekomendasi tentang suatu produk sebelum membelinya. Hal ini melahirkan tren bahwa konsumen mencari rekomendasi produk melalui orang lain, khususnya pada teman di sosial media. Sosial media khususnya Twitter ini memiliki beberapa fitur seperti kicauan, ReTweet dan mentions untuk berinteraksi dengan orang lain. Dalam suatu posting kicauan, pengguna dapat mendeskripsikan produk dan melampirkan tautan, serta memberikan penilaian positif maupun negatif pada produk tersebut. Kicauan dengan penilaian terhadap produk ini dapat digunakan sebagai alternatif rekomendasi atas suatu produk. FrecomTweet merupakan sebuah aplikasi rekomendasi produk berbasis Android yang dapat mendeteksi kedekatan pertemanan berdasarkan interaksi ReTweet dan mentions yang dilakukan oleh pengguna. Aplikasi ini juga mendeteksi sebuah rekomendasi produk yang muncul dalam percakapan antar pengguna. Pendeteksian dilakukan dengan menggunakan metode keyword filtering yaitu mencocokkan isi percakapan dengan penanda yang ada dalam basis data. Jika percakapan tersebut memiliki penilaian positif maka akan dijadikan rekomendasi bagi teman terdekat pengguna. Penelitian ini menggunakan metode crawling dengan filter streaming API Twitter dan dibangun menggunakan kerangka kerja CodeIgniter. Hasil uji coba kotak hitam menunjukkan percakapan pengguna Twitter dapat dijadikan suatu rekomendasi produk dengan nilai presisi sebesar 0.94 dan recall 0.81.

Kata Kunci: Filter kata kunci, kedekatan pertemanan, rekomendasi produk, streaming API Twitter, Twitter4J.

FRECOMTWEET: PRODUCT RECOMMENDATION APPLICATION USING FRIENDSHIP CLOSENESS ON TWITTER

Ratih NE Anggraini, Ainatul Maulida, Daniel Oranova Siahaan

Department of Informatics, Institut Teknologi Sepuluh Nopember
Jalan Teknik Kimia, Kampus ITS, Sukolilo, Surabaya
e-mail: ratih_nea@if.its.ac.id

ABSTRACT

The information and communication technology development makes someone interact with each other easier. This convenience is used to exchange ideas, like using social media Twitter for product recommendations before buying it. It brings up a trend that consumers seek product recommendations through other people on social media. Social media, especially Twitter, has several features such as tweets, ReTweet and mentions to interact with other people. Users can describe the product, attach a link, and give a positive or negative rating in a tweet. These types of tweets can be used as an alternative to product recommendations. FrecomTweet is an Android-based product recommendation application that can detect close friendships based on the user's ReTweet and mentions. This application also detects a product recommendation that appears in a conversation between users. This detection uses the keyword filtering method, which matches the conversation content with the markers in the database. If the conversation has a positive rating, it will recommend the user's closest friends. This research uses a crawling method with the Twitter API streaming filter built using the CodeIgniter framework. The results of the black box test show that Twitter user conversations can be used as a product recommendation with a precision and recall value of 0.94 and 0.81, respectively.

Keywords: Friendship closeness, keyword filtering, product recommendation, streaming Twitter API, Twitter4J.

I. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi saat ini memberikan kemudahan seseorang untuk berhubungan dengan orang lain. Keterhubungan antar orang saat ini membuat percakapan seperti percakapan tentang suatu produk barang dengan mudah dilakukan. Seseorang dapat dengan mudah mengakses aneka kanal informasi. Kemudahan bertukar informasi ini melahirkan suatu tren bahwa konsumen meminta rekomendasi pada orang lain, khususnya teman, sebelum membeli produk barang.

Berdasarkan survei yang dilakukan oleh *Forrester Research* tahun 2021, dari 58.000 responden konsumen Amerika dan Eropa sekitar 70% tersebut lebih percaya pada rekomendasi dari teman sebelum membeli produk barang, 46% responden percaya pada ulasan yang diberikan konsumen, dan sisanya 10% konsumen yang percaya pada iklan. Survei ini juga mengatakan bahwa jejaring sosial seperti Facebook dan Twitter menjadi ruang rekomendasi pilihan dan lebih efektif dibanding dengan iklan banner standar [1]. Hasil survei yang hampir sama terjadi di Indonesia, hasil penelitian *Onbee Marketing Research* pada tahun 2010 menyebutkan sekitar 80% konsumen dari 2000 konsumen Indonesia lebih mempercayai rekomendasi dari teman terdekat atau keluarga pada saat memutuskan untuk membeli suatu produk barang. Sementara itu, iklan hanya menempati peringkat ke-lima sebagai sumber referensi yang dipercaya oleh konsumen [2].

Variabel yang menyusun konsep penilaian konsumen secara online ada tiga, yaitu *Trustworthiness*, *Attractiveness*, dan *Expertise* [3]. *Trustworthiness* ialah suatu hal yang berhubungan dengan kejujuran yang dapat dipercaya. *Attractiveness* merupakan hal yang berhubungan dengan nilai sosial dari seseorang yang meliputi kepribadian tampilan fisik dan memiliki keadaan dengan orang yang menilai. Hal ini bisa dipenuhi salah satunya adalah dengan memberikan rekomendasi produk sesuai dengan preferensi konsumennya [4]. *Expertise* adalah keunggulan atau pengalaman seseorang dalam menilai produk tersebut. Selain itu babirusa testimoni yang diberikan pelanggan pada sebuah toko juga dapat menentukan produk tersebut layak beli atau tidak.

Penelitian sebelumnya tentang keputusan pembelian sudah banyak sekali dilakukan, namun pada variabel dan objek yang berbeda. Hasil penelitian menyatakan bahwa variabel keamanan yang positif dan signifikan tidak berpengaruh terhadap keputusan pembelian [5]. Namun, variabel kualitas pelayanan dan kepercayaan secara signifikan berpengaruh terhadap keputusan pembelian. Pada penelitian lainnya, Novita Ekasari menyatakan bahwa faktor-faktor bauran pemasaran jasa seperti variabel *personal relevance*, *interactivity*, *message*, *brand familiarity* berperan dalam menarik konsumen membeli produk sebesar 80.4% [6]. Variabel *interactivity* memiliki pengaruh yang paling dominan terhadap keputusan pembelian konsumen dibanding variabel lainnya. Hal ini ditunjukkan dengan nilai uji *Fhitung* > *Ftabel* ($102,679 > 2,004$) dan probabilitas kesalahan kurang dari atau sama dengan 10%. Variabel penilaian produk berpengaruh positif dan signifikan terhadap keputusan pembelian [7] [8].

Sosial media dapat menghasilkan pemahaman yang lebih baik dan komprehensif tentang persepsi publik dibanding dengan survei manual. Dari 54 penelitian yang dianalisis, 33 penelitian menggunakan sosial media Twitter untuk menganalisis opini masyarakat [9]. Penelitian berikutnya mengamati tentang perilaku pengguna sosial media menggunakan metode data mining termasuk analisis kluster dan hubungan antar pengguna. Analisis kluster dan aturan asosiasi dapat menentukan pola pengetahuan dari pengguna, motivasi dari kegunaan sosial media dan menciptakan sebuah rekomendasi *social commerce* dalam hal perkembangan teknologi. Mereka menyebutkan beberapa poin yang perlu dipertimbangkan yaitu terkait jumlah data, metode yang digunakan, format dan penggunaan kuesioner, pemilihan beberapa sosial media yakni Facebook, Line, YouTube, Instagram, Twitter, Messenger, WeChat dan Tiktok, dan aspek perbedaan budaya juga menjadi limitasi pada penelitian ini [10]. Penelitian selanjutnya merancang aplikasi Android untuk rekomendasi kuliner yang ada di kota Bandung dengan memanfaatkan metode *Location Based Service* dan *GPS*. Dari hasil kuesioner yang dilakukan, sumber terbanyak untuk mendapat informasi tempat makan yakni berasal dari sosial media, dan penggunaan *search engine* Google juga dinilai masih kurang efektif [11].

Sosial media Twitter adalah layanan jejaring sosial dan *micro blogging* yang dioperasikan oleh Twitter, Inc. Disebut *micro blogging* karena Twitter memungkinkan penggunanya untuk mengirim dan membaca pesan berbasis teks hingga 140 karakter yang dikenal dengan sebutan kicauan [12]. Dengan fitur-fitur yang ada pada Twitter, pengguna dapat berinteraksi dengan orang lain seperti menyelesaikan pertanyaan dan keluhan, mengumumkan produk dan layanan baru, mendapatkan umpan balik serta rekomendasi produk barang. Karena adanya fitur tersebut, telah banyak penelitian yang memanfaatkan Twitter sebagai dataset yang digunakan untuk membangun sistem rekomendasi, seperti untuk merekomendasikan berita dan blog [13], rekomendasi konten [14], URL [15], ataupun selebriti [16].

Pada penelitian sebelumnya, telah dikembangkan beberapa aplikasi rekomendasi produk, yang di antaranya dikembangkan dalam bentuk aplikasi mobile [17]–[19]. Sedangkan penelitian lain menggunakan *click history* dari suatu produk di sebuah *marketplace* serta data penjualannya untuk memberikan rekomendasi produk retail [20].

Berdasarkan latar belakang masalah yang telah dipaparkan di atas, maka penulis tertarik untuk mengangkat judul penelitian “Aplikasi rekomendasi produk menggunakan kedekatan pertemanan pada Twitter”. Penelitian ini bertujuan mendeteksi kedekatan pertemanan berdasarkan interaksi kicauan, *ReTweet* (RT) dan *mentions* yang dilakukan oleh pengguna. Berdasarkan kedekatan pertemanan tersebut, aplikasi akan menilai apakah kicauan yang dilakukan oleh teman di Twitter tersebut mengandung rekomendasi produk yang positif atau tidak.

II. PERANCANGAN DAN IMPLEMENTASI SISTEM

FrecomTweet adalah sebuah aplikasi mobile Android yang dibangun untuk memberikan rekomendasi produk dengan mempertimbangkan kedekatan pertemanan pada jejaring sosial Twitter. Kedekatan pertemanan ini dapat

dideteksi dari interaksi antar pengguna dalam jejaring sosial seperti komunikasi dan *tag* atau menandai seseorang. Interaksi yang dimaksud adalah interaksi pengguna dengan *following* (orang yang diikuti) yang meliputi *mentions* pada kicauan dan *ReTweet*. Kicauan yang dituliskan oleh pengguna dapat diolah untuk mendapatkan informasi tertentu salah satunya informasi suatu produk.

Selanjutnya, kicauan yang ditulis dicocokkan dengan kata kunci yang ada. Pada tahap akhir, kicauan ini dikelompokkan menjadi dua kategori yaitu positif mengandung rekomendasi atau negatif. Kicauan yang positif akan dijadikan rekomendasi untuk teman. Oleh karena itu, pemberian rekomendasi produk ini dimulai dari proses mendeteksi kedekatan pertemanan pada Twitter, dilanjutkan dengan analisis kicauan untuk rekomendasi produk.

Selanjutnya, untuk perancangan aplikasi FrecomTweet ini, akan dijelaskan pada bagian selanjutnya.

A. Perancangan Sistem

Arsitektur dari aplikasi FrecomTweet dapat dilihat pada Gambar 1. Proses rekomendasi diawali dengan melakukan *crawling* yaitu mengambil kicauan dari Twitter dengan menggunakan *filter streaming API* dengan kriteria tertentu. Pada penelitian ini, kriteria yang digunakan adalah *geolocation*, misalkan kota tertentu seperti Surabaya. Oleh karena itu, hanya kicauan dari pengguna dengan profil sesuai dengan *bounding box* lokasi yang dipilih yang akan diambil datanya.

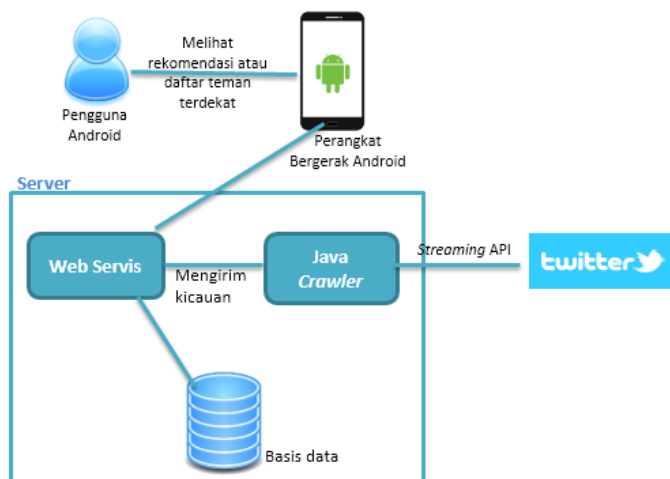
Proses *crawling* ini dilakukan oleh aplikasi Java *console* yang kemudian datanya dikirimkan ke web servis untuk diolah. Pengolahan ini terdiri dari dua proses yaitu proses pengecekan *mentions* atau *ReTweet* dan proses pengecekan rekomendasi produk. Hasil pengecekan inilah yang kemudian disimpan di dalam basis data.

Mula-mula kicauan yang baru masuk diberi nilai *flag*=1. *Flag* 1 ini menunjukkan bahwa kicauan tersebut bukanlah kicauan yang mengandung *ReTweet*. Pemberian *flag* ini untuk menandai sebuah kicauan boleh dilanjutkan pada proses cek rekomendasi atau tidak karena hanya kicauan yang bukan *ReTweet* yang akan diproses cek rekomendasi.

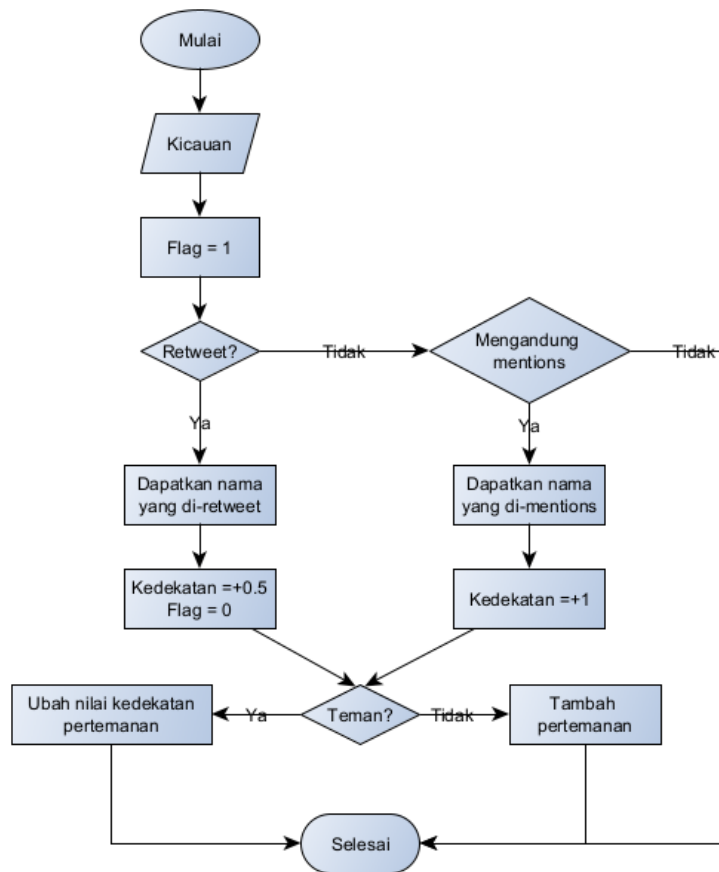
Pada proses pertama kicauan dicek apakah kicauan tersebut adalah kicauan hasil dari *ReTweet* atau bukan. Kicauan yang positif RT ini ditandai dengan awal kata pertama dari kicauan adalah 'RT'. Jika kicauan tersebut terindikasi sebagai *ReTweet*, maka akan dicari dari siapakah kicauan tersebut di*ReTweet*kan. Teman yang kicauannya di*ReTweet*kan ini nilai kedekatannya ditambahkan 0.5 dan nilai update waktu akan diperbaharui dengan waktu kicauan ini dikicaukan serta *flag* diberi nilai 0.

Kicauan yang negatif *ReTweet* akan dicek apakah kicauan tersebut mengandung *mentions* terhadap teman-temannya atau tidak. *Mentions* ini ditandai dengan karakter '@' diikuti nama teman yang di-*mentions*. Jika positif, maka poin kedekatan dengan teman yang di-*mentions* ini akan ditambah 1. Dalam suatu kicauan bisa saja ada lebih dari satu teman yang di-*mentions*. Jika hal ini terjadi, maka nilai kedekatan dengan semua teman yang di-*mentions* tersebut akan ditambahkan poin kedekatan 1 dan pembaruan waktu kedekatan akan diperbaharui dengan waktu kicauan yang mengandung *mentions* ini dikicaukan. Proses pemeriksaan kedekatan pertemanan berdasarkan *Retweet* dan *Mentions* ditunjukkan pada Gambar 2.

Proses yang kedua adalah proses cek rekomendasi. Kicauan yang masih memiliki *flag*=1 akan dicek apakah kicauan tersebut mengandung kata yang cocok dengan salah satu produk yang ada dalam basis data. Jika positif mengandung produk, maka kicauan tersebut akan dicocokkan dengan penanda yang ada dalam database. Ada tiga jenis penanda yang ada yaitu *marker* negasi, *marker* negatif, dan *marker* positif. *Marker* negasi adalah kumpulan penanda yang mengandung kata yang dapat membalikkan arti, misalnya kata "tidak". *Marker* positif adalah kumpulan kata yang jika mengandung penilaian positif. Sebaliknya *marker* negatif adalah kumpulan kata kunci yang akan menunjukkan penilaian yang negatif terhadap suatu produk.



Gambar 1. Arsitektur aplikasi FrecomTweet.



Gambar 2. Diagram alir pemeriksaan kedekatan pertemanan.

Suatu kicauan yang hanya mengandung marker positif, misal “Bebek Sinjay itu enak”. Maka akan dianggap memberikan rekomendasi positif dan akan diberi poin penilaian sebesar 20 yang artinya sangat direkomendasikan. Sedangkan jika mengandung negasi dan diikuti dengan *marker* negative, misalkan “Makan di Bebek Sinjay tidak mahal”, maka dianggap memberi penilaian positif karena pada kalimat tersebut terdapat kata yang menunjukkan negasi “tidak” yang diikuti dengan kata “mahal”.

Kicauan ini direkomendasikan dengan poin penilaian sebesar 20. Jika kicauan tidak mengandung *marker* positif maupun negative, contoh “lagi sama @mirantimia at Bebek Sinjay”, maka dianggap netral. Kalimat ini menjelaskan bahwa penulis kicauan sedang berada di warung Bebek Sinjay. Tidak ada keterangan bagaimana penilaiannya terhadap kuliner Bebek Sinjay sehingga rekomendasi ini diberi poin penilaian rekomendasi sebesar 10 yang artinya kuliner tersebut patut dicoba.

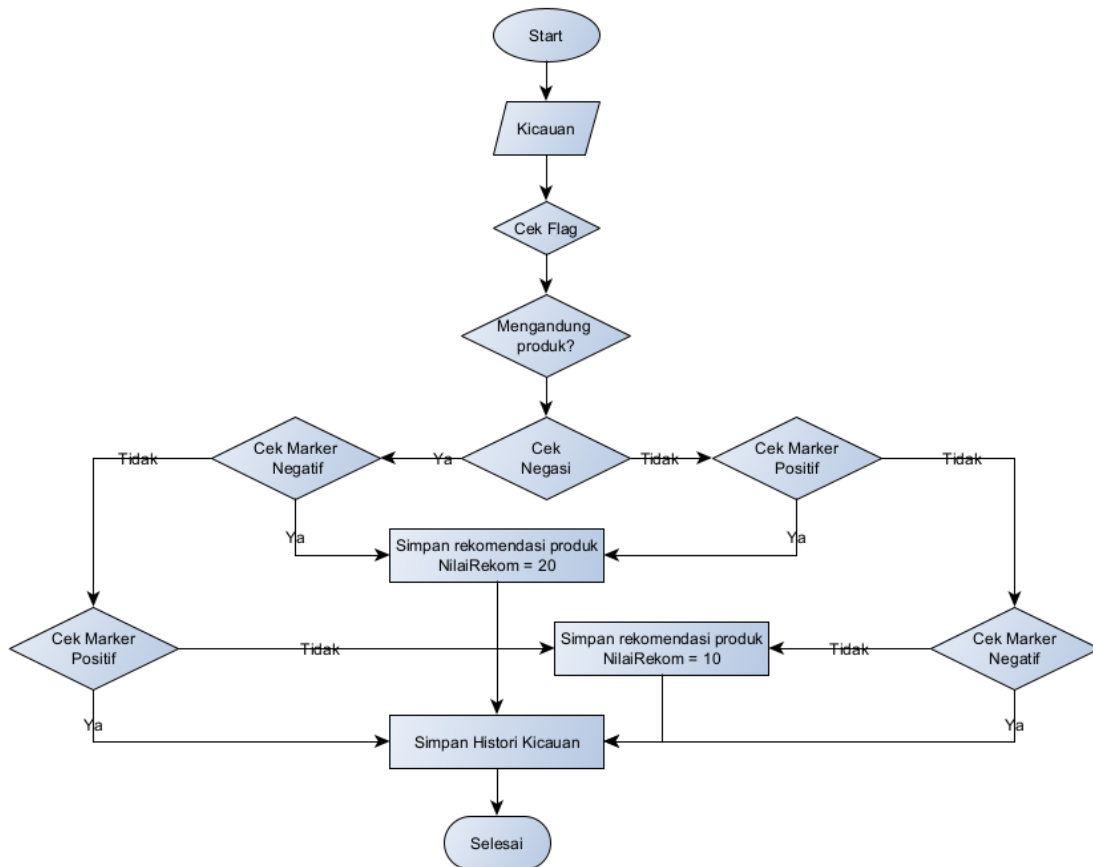
Produk yang telah dinyatakan positif ini kemudian disimpan ke dalam database lengkap dengan nama perekomendasi yaitu penulis kicauan, poin penilaiannya serta waktu kicauan tersebut dikicaukan. Proses rekomendasi produk ini dapat dilihat pada Gambar 3.

B. Implementasi Proses

Dalam merancang perangkat lunak ini digunakan beberapa perangkat pendukung yang terdiri dari perangkat keras dan perangkat lunak yang digunakan. Spesifikasi perangkat keras yaitu komputer Lenovo, prosesor Inter® Core™ i3-3240 CPU @3.40 GHz, dan RAM 4.00 GB. Lalu spesifikasi perangkat lunak yaitu Microsoft Windows 7 64 bit, tool Sublime Text, Eclipse, MySQL sebagai server basis data, Power Designer 15.0 untuk merancang basis data dan Star UML untuk desain aplikasi.

1. Proses *Crawling* Data

Proses *crawling* data Twitter dilakukan oleh mesin *crawling* Java dengan proses. Proses *crawling* ini dibagi menjadi dua mode yaitu mode produksi dan mode pengujian. Proses *crawling* pada mode produksi ini menggunakan *filter streaming API* berdasarkan *geolocation*, dengan *bounding box* kota Surabaya yaitu *latitude* -7.3491667 hingga -7.1491667 dan *longitude* 112.734146 hingga 112.834146. Sedangkan untuk mode pengujian menggunakan *filter streaming API* berdasarkan *userid* yang disimpan dalam basis data tersendiri. Proses *crawling* ini diimplementasikan pada kelas `MainStreaming.java`, `GetUserId.java`, `Streaming.java`, dan `Javapost.java`.



Gambar 3. Diagram alir proses rekomendasi produk.

Tabel I
Contoh potongan kode proses *streaming*.

1	TwitterStream twitterStream = new TwitterStreamFactory(configBuild TwitterStream twitterStream = new TwitterStreamFactory(configBuilder.build()).getInstance());
2	StatusListener listener = new StatusListener() { configBuilder.build()).getInstance();}

Tabel II
Contoh potongan kode *filter* berdasarkan *geolocation*.

1	twitterStream.addListener(listener);
2	double [][] bb = {{long1, lat1}, {long2, lat2}};
3	FilterQuery fq = new FilterQuery();
4	fq.locations(bb);
5	twitterStream.filter(fq);

Tabel III
Contoh potongan kode *filter* berdasarkan *userid*.

1	long [] followArray= user.getids();
2	twitterStream.addListener(listener);
3	FilterQuery fq= new FilterQuery();
4	fq.follow(followArray);
5	twitterStream.filter(fq);

Jika *mode* yang dipilih adalah mode produksi, maka kelas ini akan menjalankan kelas Streaming.java untuk *filter streaming* API berdasarkan *geolocation*. Berikut kode sumber untuk proses *streaming* API dan *filter* berdasarkan *geolocation* seperti Tabel II. Sedangkan jika yang dipilih adalah mode pengujian, maka akan dilakukan *streaming* berdasarkan *userid* seperti Tabel III. Sebelum melakukan *streaming*, maka kelas StramingData.java akan memanggil method yang ada di kelas GetUserid.java untuk mengambil semua *userid* yang akan digunakan dalam *streaming* mode pengujian. Hasil dari proses *crawling* ini kemudian dikirimkan ke web servis melalui kelas Javapost.java untuk kemudian dilakukan proses pengecekan *mentions/ReTweet* serta pengecekan kandungan rekomendasi produk.

2. Proses Klasifikasi Kicauan

Pada web servis, data kicauan yang diterima akan dilakukan pengecekan *mentions/ReTweet* serta pengecekan kandungan rekomendasi produk. Pengecekan ini dilakukan oleh kelas *Home.php*. Mula-mula data yang diterima diberi nilai *flag=1*. Flag 1 ini menunjukkan bahwa kicauan tersebut bukanlah kicauan yang mengandung *ReTweet*. Pemberian flag ini untuk menandai sebuah kicauan boleh dilanjutkan pada proses cek rekomendasi atau tidak, karena hanya kicauan yang bukan *ReTweet* yang akan diproses cek rekomendasi dengan kode yang tertera pada Tabel IV.

Berdasarkan Tabel IV, jika kicauan tersebut mengandung *ReTweet* atau *mentions*, maka kelas *controller* akan memanggil kelas model *follow.php* untuk melakukan proses pembaruan atau tambah data pada tabel *Follow*. Pembaruan jika di dalam tabel *follow* belum ada data pertemanan antara penulis kicauan dengan teman yang di *ReTweet* atau *mentions*. Sebaliknya, jika data pertemanan telah ada, maka akan dijalankan proses tambah data pertemanan pada tabel *follow*. Proses pembaruan atau tambah data ini dilakukan oleh prosedur *update-Follow()* yang ada dalam basis data. Kicauan yang masih memiliki nilai *flag=1* akan diproses cek rekomendasi. Mula-mula kicauan akan dicek apakah kicauan mengandung salah satu nama produk yang ada dalam tabel produk. Jika ada, maka akan dicek kandungan *markernya* seperti pada penjelasan di Gambar 1. Apabila kicauan positif rekomendasi, maka akan dilakukan penambahan data pada tabel rekomendasi. Penambahan data rekomendasi ini melibatkan kelas model rekomendasi.php yang akan memanggil prosedur *insertRekomendasi* pada basis data.

3. Proses Penentuan Teman Terdekat

Proses ini dilakukan ketika ada sebuah *client* mengirimkan permintaan untuk melakukan menghitung kedekatan pertemannya. Pada *client*, proses ini melibatkan tiga kelas yaitu kelas *PencarianTemanActivity.java* berfungsi sebagai kelas untuk memilih jangka waktu kedekatan pertemanan, *JSONParser.java* untuk mengirimkan permintaan pada webservis dengan menyertakan parameter *idpengguna* dan jangka waktu kedekatan serta *AmbilDataTeman.java* untuk menerima data JSON untuk kemudian ditampilkan. Kode sumber mengirikan pertmintaan pada web servis dapat dilihat pada Tabel V.

Pada sisi web servis, permintaan *client* ditangani oleh kelas *KedekatanPertemana.php*. Kelas ini kemudian memanggil kelas model *Follow.php* untuk melakukan kueri data pada tabel *follow* sesuai dengan parameter yang dikirimkan. Data hasil kueri ini kemudian dikembalikan dalam format JSON. Selanjutnya ditangkap oleh *JSONParser.java* pada *client*. Kode sumber untuk implementasi menangkap respon JSON dari web servis dapat dilihat pada Tabel VI.

4. Proses Rekomendasi Produk

Proses rekomendasi produk dilakkukan ketika sebuah *client* mengirimkan permintaan rekomendasi. Mula-mula kelas *PermintaanRekomendasiActivity.java* mengirimkan permintaan ke webservis untuk mengirimkan kategori produk yang tersedia. Hal ini ditangani webservis pada kelas *KategoriProduk.php*. kelas ini akan memanggil model kelas *Ketegori.php* untuk melakukan kueri daftar kategori dari basis data. Ketika pengguna memilih salah satu kategori produk, maka kelas *HasilRekomendasiActivity.java* akan memanggil kelas *JSONParser.java* untuk melakukan permintaan pada webservis. Permintaan ini dengan menyertakan parameter *idpengguna* dan kategori produk. Permintaan ini ditangani oleh kelas *RekomendasiProduk.php* pada sisi server. Selanjutnya kelas ini memanggil kelas model *Rekomendasi.php* untuk melakukan kueri pada tabel rekomendasi sesuai dengan parameter yang ada. Pada proses kueri inilah diimplementasi persamaan 2.1 yaitu perhitungan nilai rata-rata rekomendasi untuk setiap produk. Kueri ini menghasilkan lima produk yang memiliki nilai rata-rata rekomendasi tertinggi sebagai produk yang direkomendasikan bagi pengguna. kode sumber implementasi ini bisa dilihat pada Tabel VII. Selanjutnya hasil kueri ini dikembalikan dalam bentuk JSON yang kemudian diurai dan dimasukkan ke dalam *HashMap* untuk ditampilkan pada halaman *hasilrekomendasi.xml*.

Tabel IV
Kode cek *retweet* pada kicauan.

1	<code>\$cekRT=strpos(\$tweet, 'RT');</code>
2	<code>\$cekMention=strpos(\$tweet, '@');</code>

Tabel V
Kode Mengirim Permintaan pada *Web Service*.

1	<code>DefaultHttpClient httpClient=new DefaultHttpClient()</code>
2	<code>HttpPost httpPost=new HttpPost(url)</code>
3	<code>HttpResponse httpResponse=httpclient.execute(httpPost)</code>
4	<code>HttpEntity httpEntity=httpresponse.getEntity()</code>
5	<code>is=httpEntity.getContent()</code>

Tabel VI
Kode Mengirim Permintaan pada *Web Service*.

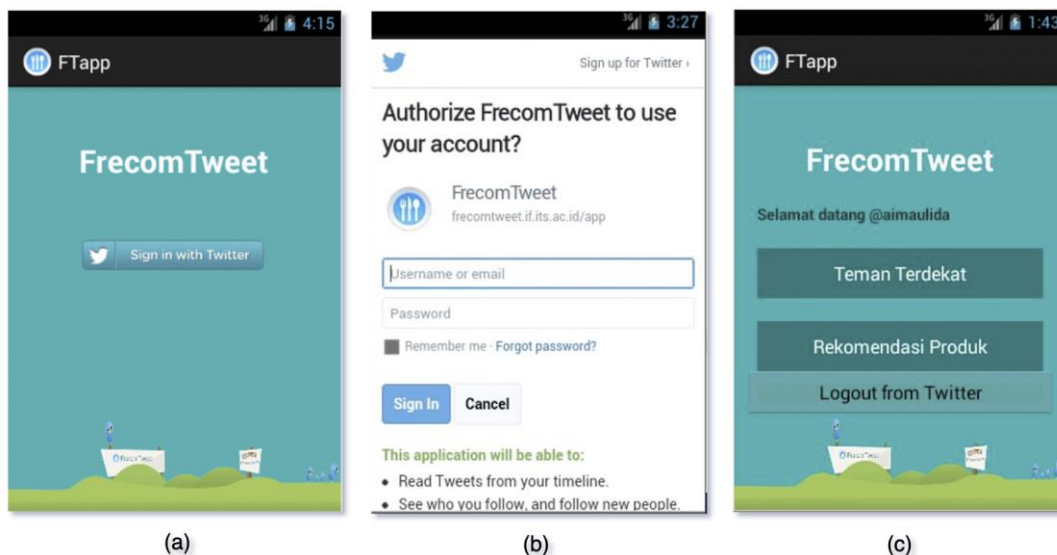
```

1  BufferedReader reader=new BufferedReader(new In-
   putStreamReader(is, "iso-8859-1"),8)
2  StringBuilder sb=new StringBuilder()
3  String line=null
4  while((line=reader.readLine())!=null){
5  sb.append(line) }
6  is.close()
7  json=sb.toString()
    
```

Tabel VII
Kueri Rekomendasi Produk.

```

1  "select b.namaproduk,GROUP_CONCAT(b.idpengguna order by b.nilaikedekatan desc
   separator', @') as perekomendasi, sum(b.hasilkali)as totrek,sum(b.nilaikedeka-
   tan)as Total_NK,ROUND(sum(b.hasilkali)/sum(b.nilaikedekatan),2)as rekomvalue,
   b.alamat
2  from (select rp.idpengguna,a.nilaikedekatan,p.namaproduk, AVG(nilairekomendasi)
   as Rata2_nilai,a.nilaikedekatan*AVG(nilairekomendasi) as hasilkali,p.alamat
3  from (SELECT * FROM follow where idpengguna= '$ids' and updatewaktu >=
   DATE_SUB(CURDATE(),INTERVAL 30 day)) a
4  inner join rekomendasiproduk rp
5  on a.idteman= rp.idpengguna
6  and rp.tanggalrekomendasi >= DATE_SUB(CURDATE(),INTERVAL 30 day)
7  inner join produk p
8  on rp.idproduk = p.idproduk
9  inner join kategori k
10 on p.idkategori = k.idkategori and k.namakategori='$kategori'
11 group by rp.idproduk, rp.idpengguna
12 order by a.nilaikedekatan desc ) b
13 group by b.namaproduk
14 order by rekomvalue desc,Total_NK desc
15 limit 5";
    
```



Gambar 4. Tampilan Antarmuka Halaman Awal (a), Login Twitter (b) dan Halaman Utama (c).

C. Implementasi Antarmuka

Lapisan antarmuka merupakan lapisan yang bertugas mengatur tampilan sistem agar dapat berinteraksi dengan pengguna. Desain halaman antarmuka ini disimpan dengan ekstensi xml. Setiap antarmuka diatur oleh sebuah kelas *controller* yang merupakan *extends* kelas *Activity*.

1. Antarmuka Halaman Awal dan Menu Utama

Halaman ini merupakan tampilan utama yang muncul ketika pengguna membuka aplikasi pertama kali. Untuk dapat menggunakan aplikasi ini pengguna harus masuk menggunakan akun Twitternya. Pada halaman ini terdapat sebuah *button* *sign in* yang akan mengarahkan pengguna untuk melakukan autorisasi *login* Twitter pada *browser*. Antarmuka halaman awal dapat dilihat pada Gambar 4a dan autorisari *login* Twitter pada

Gambar 4b. Sedangkan pada halaman utama aplikasi terdapat nama pengguna yang login, serta tiga buah button yaitu button untuk melihat teman terdekat, permintaan rekomendasi dan keluar dari aplikasi seperti pada Gambar 4c.

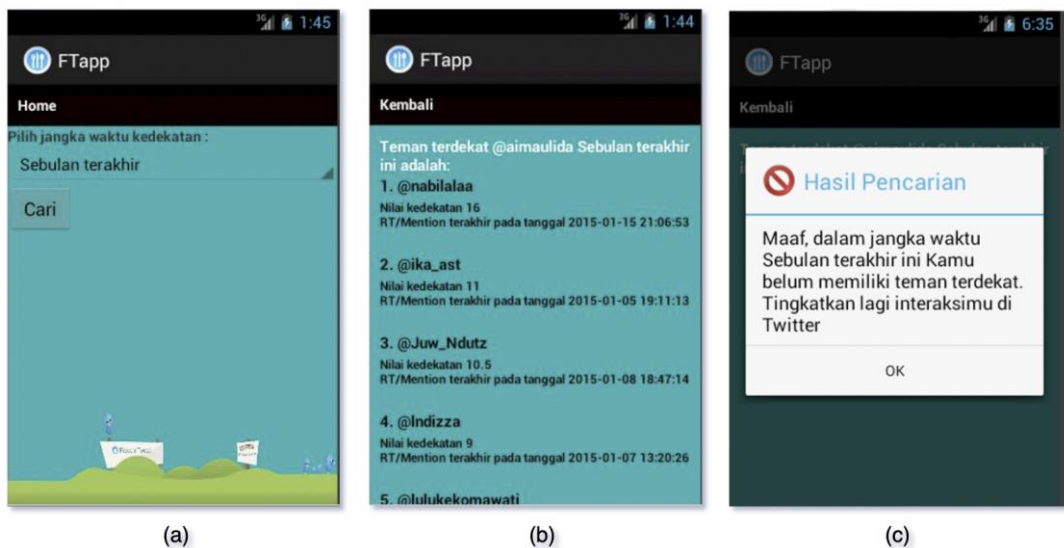
2. Antarmuka Halaman Pencarian Teman Terdekat

Implementasi desain antarmukan halaman pencarian teman terdekat ini disimpan pada `pencarianteman.xml`. Sedangkan kelas `controller` ditangani oleh kelas `Pencarianteman Activity.java`. Pada halaman ini terdapat `textview` yang berisi keterangan untuk memilih jangka waktu kedekatan pertemanan, sebuah `Spinner` jangka waktu kedekatan pengguna dengan `default` pencarian “Sebulan Terakhir”, dan sebuah `button` pencarian.

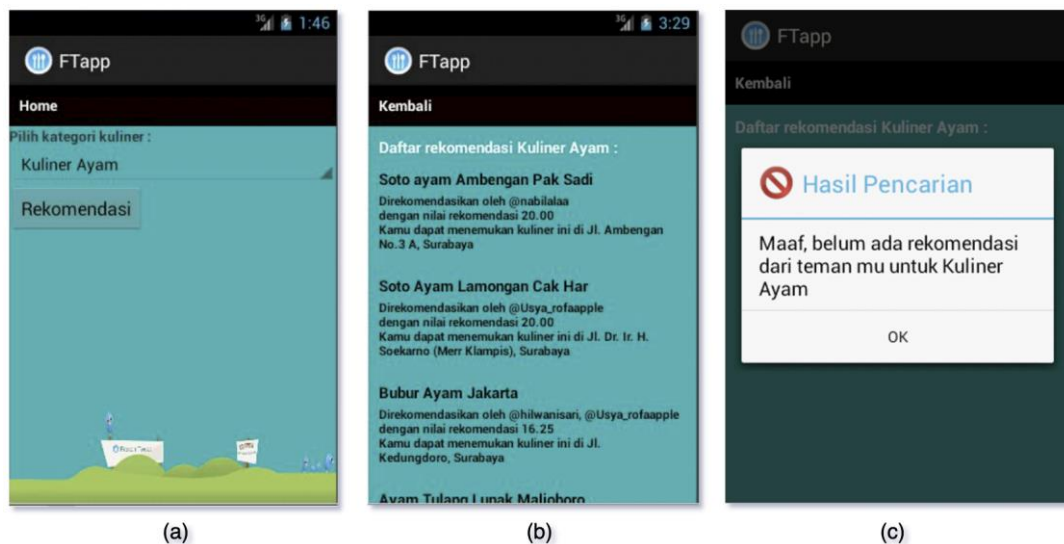
Antarmuka halaman pencarian teman terdekat dapat dilihat pada Gambar 5a berikut ini. Berikutnya pada halaman ini terdapat sebuah `textview` keterangan dari daftar teman terdekat yang ditampilkan `listview` yang ada di bawahnya. Pada `listview` hasil pencarian teman terdekat ditampilkan nama teman terdekat, nilai kedekatannya, serta informasi waktu terjadinya interaksi terakhir dengan teman. Antarmuka halaman daftar teman terdekat dapat dilihat pada Gambar 5b. Dan jika pengguna belum memiliki teman terdekat dalam jangka waktu yang dipilih, maka akan ditampilkan pesan seperti pada Gambar 5c.

3. Antarmuka Halaman Rekomendasi Produk

Implementasi desain antarmukan halaman pencarian teman terdekat ini disimpan pada `pencarianteman.xml`. Sedangkan kelas `controller` ditangani oleh kelas `Pencarianteman Activity.java`. Pada halaman ini terdapat `textview` yang berisi keterangan untuk memilih kategori produk kuliner seperti pada Gambar 6a. Lalu berikutnya halaman hasil pencarian yang berisi sebuah `textview` keterangan daftar rekomendasi yang ditampilkan dalam bentuk list seperti Gambar 6b. Berikutnya jika tidak menemukan data terkait dengan kata kunci yang dipilih maka akan menampilkan pemberitahuan tidak ada rekomendasi seperti pada Gambar 6c.



Gambar 5. Tampilan Antarmuka Halaman Pencarian (a), Daftar (b) dan Pemberitahuan Belum Mempunyai Teman Terdekat (c).



Gambar 6. Tampilan Antarmuka Halaman Perminaan Rekomendasi (a), Hasil Rekomendasi (b) dan Pemberitahuan Tidak Ada Rekomendasi (c).

D. Implementasi Antarmuka

Proses implementasi basis data pada penelitian ini meliputi implementasi struktur basis data, implementasi prosedur, dan implementasi event. Implementasi struktur. Basis data pada server basis data MySQL. Implementasi ini meliputi implementasi pembuatan tabel pengguna, follow, kategori, produk, rekomendasiproduk, serta riwayatkicauan.

Prosedur merupakan salah satu objek *routine* yang tersimpan pada database MySQL dan dapat digunakan untuk menggantikan berbagai kumpulan perintah. Pada sistem ini menggunakan dua buah prosedur yakni *insertRekomendasi* dan *updateFollow*. Prosedur *insertRekomendasi* merupakan prosedur yang dijalankan untuk mengecek apakah pengguna telah terdaftar. Jika belum ada, maka akan dilakukan proses tambah pengguna, namun jika sudah terdaftar akan menambahkan rekomendasi produk dalam basis data. Prosedur kedua adalah prosedur *updateFollow*, yang berfungsi untuk mengubah nilai kedekatan pada tabel *follow*. Sebelum mengubah nilai prosedur akan mengecek apakah idpengguna dan idteman telah terdaftar, serta mengecek apakah sudah ada status pertemanan antar keduanya. Jika belum maka akan ditambahkan. Sebaliknya jika sudah ada, maka akan diperbarui nilai kedekatan pertemanan dan waktunya.

Event Scheduler merupakan sebuah fitur pada MySQL untuk menjalankan perintah kueri SQL secara otomatis dan terjadwal. Pada aplikasi ini *event scheduler* digunakan untuk menghapus data pertemanan yang telah lebih dari 31 hari tidak diperbarui karena dianggap sudah bukan teman terdekat.

III. PENGUJIAN DAN HASIL

Pada bagian ini akan dijelaskan mengenai skenario uji coba dan pembahasan hasil pengujian.

A. Pengujian Sistem

Skenario pengujian pada sistem ini menggunakan pengujian fungsionalitas yakni metode kotak hitam (*black box*). Metode ini menekankan pada kesesuaian hasil keluaran sistem. Pengujian dilakukan dengan akun uji yang di-*crawling* sebanyak 18 akun Twitter dan ditambah dengan 30 akun. Pengujian kebutuhan fungsionalitas pada penelitian ini dilakukan sebanyak dua jenis, pertama pengujian fitur melihat daftar teman terdekat. Selanjutnya pengujian fitur melihat rekomendasi produk dari teman terdekat. Hasil pengujian fitur sistem dapat dilihat pada Tabel VIII.

B. Hasil

Berdasarkan hasil implementasi dan pengujian pada bab sebelumnya, untuk mengukur performa perlu dilakukan perhitungan tingkat persisi dan *recall*. Presisi mengukur tingkat ketepatan antara informasi yang diminta oleh pengguna dan jawaban dari sistem. Sedangkan *recall* merupakan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Hasil perhitungan dan jumlah rekomendasi positif dapat dilihat pada Tabel IX.

Pada rentang waktu 10 hari selanjutnya didapatkan sebanyak 280 kicauan. Dari 280 kicauan, ada 5 kicauan yang terdeteksi rekomendasi menurut sistem. Akan tetapi menurut pengembang seharusnya ada 6 kicauan yang positif rekomendasi. Hal ini berarti ada 1 kicauan yang tidak terdeteksi yaitu kicauan “@kulinersurabaya @foodsurasabaya - Soto Ayam Cak To Undaan Wetan”. Hal ini dikarenakan belum adanya penanda untuk kuliner soto ayam Cak To. Pada rentang waktu 5 hari terakhir *crawling* kicauan didapatkan sebanyak 3 kicauan positif rekomendasi menurut sistem dari 148 kicauan yang terkumpul. Sedangkan menurut pengembang seharusnya ada 4 kicauan yang terdeteksi rekomendasi. Hal ini berarti ada 1 kicauan yang tidak terdeteksi rekomendasi yaitu kicauan “@kulinersurabaya min dah pernah nyobain bebek petemon gak? Menurut aq sambelnya enak. kl rasa bebeknya siy sm aja ky lainx tp sambelx juara”. Ini juga terjadi karena belum tersedianya penanda untuk kuliner bebek Patemon. Dari hasil deteksi yang telah dijabarkan di atas, didapatkan bahwa nilai rata-rata nilai presisi untuk aplikasi ini sebesar 0.94. Sedangkan nilai *recall* sebesar 0.81.

Tabel VIII
Rangkuman Hasil Pengujian.

ID	Nama	Skenario	Hasil
UJ.FT-0001	Pengujian fitur melihat daftar teman terdekat pengguna	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil
UJ.FT-0002	Pengujian fitur untuk melihat rekomendasi produk dari teman terdekat	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil

Tabel IX
Daftar Jumlah Data Hasil Deteksi Rekomendasi.

Rentang waktu	Terdeteksi Rekomendasi	Positif Rekomendasi	Positif dari Pengembangan	presisi	recall
20 hari	22	18	21	0.82	0.85
10 hari	5	5	6	1	0.83
5 hari	3	3	4	1	0.75
Rata-rata				0.94	0.81

IV. KESIMPULAN DAN SARAN

Pada penelitian ini telah dihasilkan sebuah aplikasi FrecomTweet yang merupakan aplikasi berbasis Android dan dapat digunakan untuk memberikan rekomendasi produk berdasarkan tweet dari teman. Proses rekomendasi dilakukan dengan melakukan filtering keyword isi tweet atau mention menggunakan marker positif dan negative. Selanjutnya, akan dideteksi apakah di dalamnya mengandung rekomendasi produk. Hasil pengujian menunjukkan nilai rata-rata presisi hasil deteksinya adalah 94% sedangkan recallnya sebesar 81%.

Nilai presisi dan recall dari aplikasi ini dipengaruhi oleh jumlah marker yang telah disimpan. Oleh karena itu, diperlukan untuk menambahkan jumlah marker positif maupun negatif sehingga nilai presisi dan recall dapat ditingkatkan. Untuk pengembangan sistem selanjutnya, juga dapat dilakukan dengan metode klasifikasi seperti BiLSTM, dengan harapan, nilai presisi dan recallnya akan semakin meningkat. Selain itu, metode pemrosesan bahasa natural dapat dilakukan terlebih dahulu sebelum dilakukan proses klasifikasi rekomendasi produk.

DAFTAR PUSTAKA

- [1] "Sebanyak 70 Persen Konsumen Percaya Merek Via Rekomendasi Teman," 22 Maret 2013. [Online]. Available: <https://www.marketeers.com/sebanyak-70-persen-konsumen-percaya-merek-via-rekomendasi-teman/>. [Accessed 8 October 2021].
- [2] Nurhardianti, "Strategi BRI Unit Hasanuddin Parepare dalam Menjaga Loyalitas Nasabah," M.S. thesis, Abbrev. Studi Perbankan Syariah, IAIN, Parepare, 2018.
- [3] M. Febriana and E. Yulianto, "Pengaruh Online Consumer Review Oleh Beauty Vlogger Terhadap Keputusan Pembelian (Survei Pada Mahasiswa Fakultas Ilmu Administrasi Universitas Brawijaya Angkatan 2014/2015 Dan 2015/2016 Yang Membeli Dan Menggunakan Purbasari Matte Lipstick)," *Jurnal Administrasi Bisnis*, vol. 58, no. 1, pp. 1-9, 2018.
- [4] A. S. Ahmadiyah, F. Aidah, N. Meutia, D. Lumbantobing and R. Anggraini, "Remarketing Media Alternatives Based on Customer Preferences," *JUTI : Jurnal Ilmiah Teknologi Informasi*, vol. 18, no. 2, 2020.
- [5] A. A. Sukma, "Analisis Faktor-Faktor Yang Mempengaruhi Keputusan Pembelian Melalui Social Networking Websites," *undefined*, 2012.
- [6] N. Ekasari, "Pengaruh Promosi Berbasis Sosia Media Terhadap Keputusan Pembelian Produk Jasa Pembiayaan Kendaraan Pada PT. BFI Finance Jambi," *Jurnal Penelitian Universitas Jambi Seri Humaniora*, vol. 16, pp. 81 - 102, Juli 2014.
- [7] A. Surmiandari, "Viral Marketing Sebagai Alternatif Strategi Pemasaran Produk Sariz," *Widya Cipta: Jurnal Sekretari dan Manajemen*, vol. 1, no. 1, pp. 35-43, Maret 2017.
- [8] J. Nieto, R. M. Hernandez-Maestro and P. A. Munoz-Gallego, "Marketing decisions, customer reviews, and business performance: The use of the Toprural website by Spanish rural lodging establishments," *Tourism Management*, vol. 45, pp. 115-123, Dec 20214.
- [9] X. Dong and Y. Lian, "A review of social media-based public opinion analyses: Challenges and recommendations," *Technology in Society*, vol. 67, p. 101724, Nov. 2021.
- [10] S. H. Liao, R. Widowati and Y. C. Hsieh, "Investigating online social media users' behaviors for social commerce recommendations," *Technology in Society*, vol. 66, p. 101655, Aug. 2021.
- [11] R. Fauziyyah and T. Suryana, "Pembangunan Aplikasi Rekomendasi Kuliner Asia Di Kota Bandung Berbasis Mobile Android," *Komputa : Jurnal Ilmiah Komputer dan Informatika*, vol. 8, no. 2, pp. 52-60, Oct. 2019.
- [12] H. Kwak, C. Lee, H. Park and S. Moon, "What is Twitter, a Social Network or a News Media?".
- [13] G. Kazai, I. Yusof and D. Clarke, "Personalised news and blog recommendations based on user location, facebook and twitter user profiling," 2016.
- [14] A.Sharma, J. Jiang, P. Bommanavar, B. Larson and J. lin, "GraphJet: Real-time content recommendations at twitter," in *Proceedings of the VLDB Endowment*, 2015.
- [15] G. Piao and J. G. Breslin, "Analyzing aggregated semantics-enabled user modeling on Google+ and Twitter for personalized link recommendations," 2016.
- [16] M. S. Tajbakhsh and H. Emamgholizadeh, "Multi-agent celebrity recommender system (MACeRS): Twitter use case," *Social Network Analysis and Mining*, vol. 12, no. 1, 2022.
- [17] E. Peltonen, E. Lagerspetz, P. Nurmi and S. Tarkoma, "Constella: Crowdsourced system setting recommendations for mobile devices," *Pervasive and Mobile Computing*, vol. 26, 2016.
- [18] N. Shakhovska, S. Fedushko, M. Gregusml, I. Shvorob and Y. Syeroy, "Development of mobile system for medical recommendations," in *Procedia Computer Science*, 2019.
- [19] P. Alvarez, F. J. Zarazaga-Soria and S. Baldassarri, "Mobile music recommendations for runners based on location and emotions: The DJ-Running system," *Pervasive and Mobile Computing*, vol. 67, 2020.
- [20] H. Hwangbo, Y. S. Kim and K. J. Cha, "Recommendation system development for fashion retail e-commerce," *Electronic Commerce Research and Applications*, vol. 28, pp. 94-101, 2018.