Jurnal Teknik Industri, Vol. 26, No. 2, December 2024
ISSN 1411-2485 print / ISSN 2087-7439 online
DOI: https://doi.org/10.9744/jti.26.2.131-144

A Text Mining Approach to Analyzing the Omnichannel Retail Business Performance of the KlikIndomaret App

Akhmad Ghiffary Budianto[1*], Arief Trisno Eko Suryo[1], Andry Fajar Zulkarnain[1], Gunawan Rudi Cahyono[1], Rusilawati[1], Siti Fatimah Az-Zahra[1]

[1)] Faculty of Engineering, Electrical Engineering Department, Universitas Lambung Mangkurat
Jl. Achmad Yani Km. 33,5 Banjarbaru, Kalimantan Selatan 70714, Indonesia
Email: ghiffaryb04@gmail.com*, arief.suryo@ulm.ac.id, andry.zulkarnain@ulm.ac.id, gunawan.cahyono@ulm.ac.id, habsyi.sila@gmail.com, aazzahraa615@gmail.com
*Corresponding author

Abstract: The evolution of Web 2.0 technology has significantly influenced the use of Android applications, enabling users to provide feedback through reviews and star ratings. In managing omnichannel retail businesses, this user-generated content serves as a valuable source of information for performance evaluation and strategic management of both online and offline operations. Large-scale user review data is well-suited for analysis through text mining, particularly in sentiment analysis, when combined with topic and keyword filtering in the business domain. This study utilizes the RoBERTa Transformer model for the sentiment classification of user reviews. Among the 520 user reviews, 211 displayed good emotion, while 309 showed negative sentiment. By applying filtering processes to topics and keywords within the omnichannel retail business domain, the study identifies "economic value" and "delivery and CRM" as priority areas for improvement. This conclusion is drawn based on the significant disparity between positive and negative sentiments. As a result, management can formulate strategies to enhance the performance and user experience of the KlikIndomaret Android application.

Keywords: Text mining, sentiment analysis, omnichannel, transformer, web scrapping.

Introduction

The rapid growth of digital information presents new challenges across various industries. Mobile applications have become prevalent in many aspects of life and are widely used by consumers [1]. These applications serve numerous functions, including payment processing [1], checking product availability online [2], and other retail-related activities. Specific Web 2.0 applications within the retail sector offer unique features and functions that align with the opportunities, risks, goals, and operational and strategic needs of omnichannel management. Retailers can identify and adapt to these opportunities and risks to enhance their omnichannel retailing strategies [3]. Omnichannel retailing refers to the use of various channels to interact with customers and fulfill their orders. The interaction between a customer and a retail channel is typically based on three flows: information, products, and funds. The retailer provides product and pricing information to the customer, who then places an order. The order information is used by the retailer to deliver the product to the customer, and payment is transferred from the customer to the retailer. The use of different channels for each flow helps categorize the components of omnichannel retailing [4].

With the advancement of Web 2.0 technology, application users have become a primary source of information, particularly through online reviews. These reviews, such as those found on Amazon, can reveal an individual's opinion about a product [5]. User-generated reviews can assist others in making informed purchase decisions and serve as valuable feedback for future application development. However, mobile application development often fails to align with user expectations due to a lack of data on consumer preferences [6].

Text mining is a knowledge-based process that employs analytical tools to extract meaningful information from natural language text. It is used to examine and explore unstructured data, uncovering significant patterns from textual data sources [7]. In a highly competitive business landscape, text mining supports enterprise business intelligence processes [8]. One application of text mining is sentiment analysis, which helps businesses understand user opinions about products and services. This process, known as Voice of Customer (VOC) data [9], enables the analysis of public opinions, sentiments, evaluations, assessments, attitudes, and emotions toward entities such as products, services, organizations, individuals, issues, and events. Sentiment analysis categorizes reviewer attitudes into positive, negative, or neutral sentiments [10].

Previous research on sentiment analysis in mobile applications has been conducted on several e-commerce websites, including Amazon [11], [12] and eBay [13]. Sentiment analysis using text mining on social media platforms primarily gauges public opinions about specific organizations [14]. User reviews posted on an application can provide valuable insights for developers, highlighting features that are either appreciated, criticized, or recommended. These reviews not only express opinions about application features but also convey users' feelings or sentiments toward these features [15]. In sentiment analysis, limitations can arise from the word count of user reviews. When reviews exceed 150 words, feature extraction becomes necessary to enhance classification accuracy.

Conversely, shorter reviews tend to be more accurate in semantic classification [16]. The precision of text mining in translating user reviews into actionable information is crucial. Additionally, users may sometimes give high star ratings but express negative sentiments in their reviews. Transformer algorithms are effective in filtering such discrepancies [17]. Implicit reviews provided by users offer insights into consumer preferences, often inferred from general statements or behaviors rather than direct feedback. Web 2.0 technology allows users to provide feedback through comments and star ratings. These comments serve as data sources for sentiment analysis [18], while explicit reviews offer direct and specific assessments or evaluations of a product, service, or experience [10]

Improving service systems and enhancing usability are vital in intelligent retail. Satisfied customers are more likely to recommend a product or service to others, thereby increasing the likelihood of future purchases [19]. This is known as behavioral loyalty, which refers to the readiness for repeat purchases or maintaining a relationship with a brand or company [20]. Data analysis shows a positive correlation between after-sales service quality and customer satisfaction, and a further positive correlation between customer satisfaction and customer loyalty [21].

In Indonesia, KlikIndomaret is one of the most popular applications for shopping for everyday goods. The KlikIndomaret app has already garnered over 5 million downloads and received reviews from more than 140,000 users, with an overall rating of 4.3 stars. The app offers an omnichannel retail system, allowing users to shop for goods online and either pick them up at physical stores or have them delivered. KlikIndomaret also features a membership system to foster user loyalty. The app's account system provides flexible payment options, including e-wallets, bank transfers, and cash on delivery (COD). These features have attracted a wide range of user feedback, leading to diverse comments on the application.

RoBERTa (Robustly optimized BERT approach) is a pre-trained language model based on the Transformer architecture, similar to BERT (Bidirectional Encoder Representations from Transformers). Developed by the Facebook AI team, RoBERTa offers several advantages over BERT, including longer training with larger batches, training over more data, the removal of the next sentence prediction objective, training on longer sequences, and dynamically changing the masking pattern applied to the training data [22]. RoBERTa can be fine-tuned for text classification tasks, such as sentiment analysis, effectively using its contextual embeddings to categorize text into predefined categories [23]. This capability is particularly useful for understanding both explicit and implicit reviews from users.

This research focuses on understanding user reviews of mobile applications in the retail sector. Text mining can be leveraged to develop intelligent systems that enhance customer service. These systems, designed to interpret user sentiments, can provide valuable insights into consumer behavior and improve application usability [24]. Therefore, this study addresses the following research questions: (1) By utilizing text mining with RoBERTa, what is the overall sentiment of KlikIndomaret users? (2) Based on the literature on omnichannel performance, what are the main drivers of negative sentiment among KlikIndomaret users? The use of Transformers combined with RoBERTa pre-training is expected to efficiently classify positive and negative sentiments. The classified data can then be filtered by keywords relevant to user review topics. This analysis of user comments will help identify areas for improvement in both mobile application features and offline service aspects in stores. Consequently, management can develop strategies to enhance user satisfaction and loyalty in innovative retail services [15].

Methods

Web Scraping on the Google Play Store for the KlikIndomaret Application

For data collection, web scraping was conducted using the Python library (google-play-scraper). The token used in this web scraper was 'com.indomaret.klikindomaret' [25]. User review data was collected only from January

1 to April 15, 2024. Data collection should be conducted periodically throughout the year, with intervals of every 3-4 months to align with the application's bug fix plan and feature adjustments. According to the latest datalog, updates for the KlikIndomaret application were released on April 3, 2024 (version 2404100), May 9, 2024 (version 2405100), and June 11, 2024 (version 2406100). Since bug fixes are implemented monthly, a period of 3-4 months is considered sufficient for gathering feedback from app users.

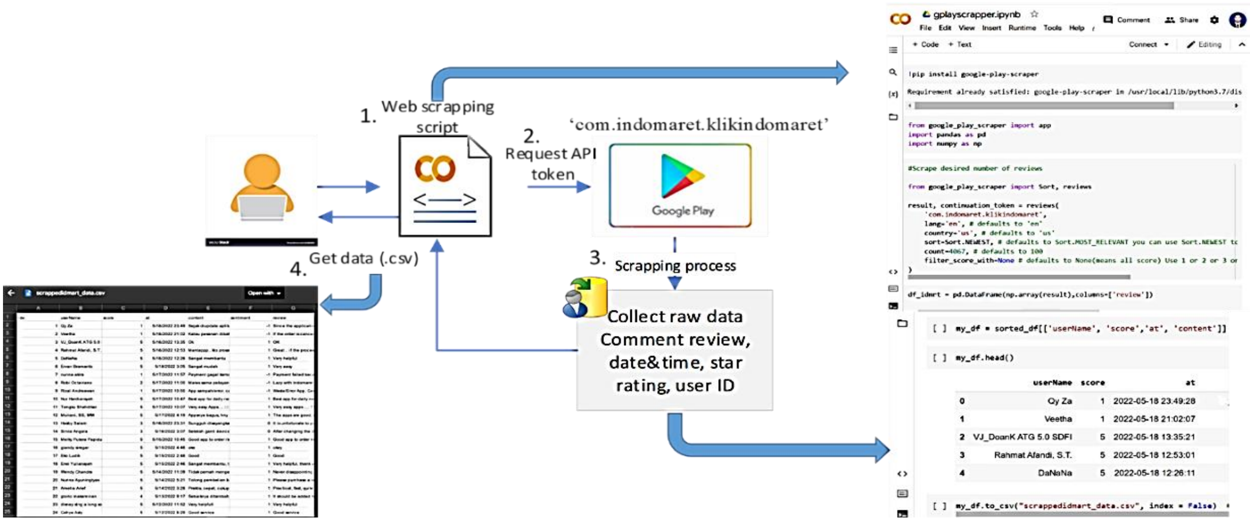


Figure 1. Web scraping process

The collected data was transformed into .csv format for further preprocessing. The dataset was sorted from newest to oldest. Star ratings were converted into sentiment labels, where negative sentiments were labeled with 1-2 stars, neutral with 3 stars, and positive sentiments with 4-5 stars. This labeling was done to structure the data but depends on the subjectivity of the commenter and the star rating. The script for web scraping on Google Colab can be found in Appendix A, and the web scraping results are available in Appendix B or Figure 5.

Data Preprocessing and Data Cleaning

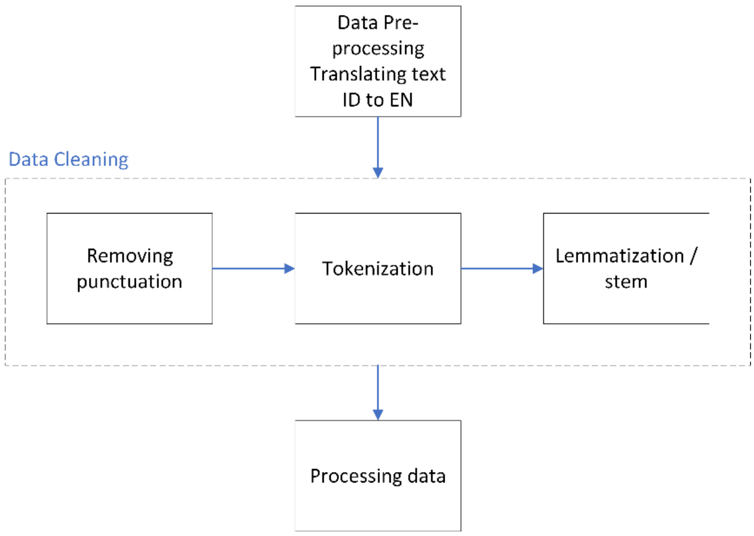


Figure 2. Data preprocessing and data cleaning

During the data preprocessing stage, user review data undergoes a data cleaning process. This involves manually translating the 'reviews content' column from Indonesian to English using Google Sheets with the formula "=GOOGLETRANSLATE(text, ["id", "en"])". Although the web scraping results are initially set to the "en" language, this step is necessary to ensure all data is consistently in English. By ensuring that the data is clean, consistent, and in English, the subsequent data processing and text analysis will yield more accurate information and deeper insights. This step is crucial to ensure the entire data processing is conducted in English.

Removing punctuations. Punctuation marks often do not carry significant meaning or add value to NLP models. In Python, punctuation can be removed using the following code:

```
import string
string.punctuation
```

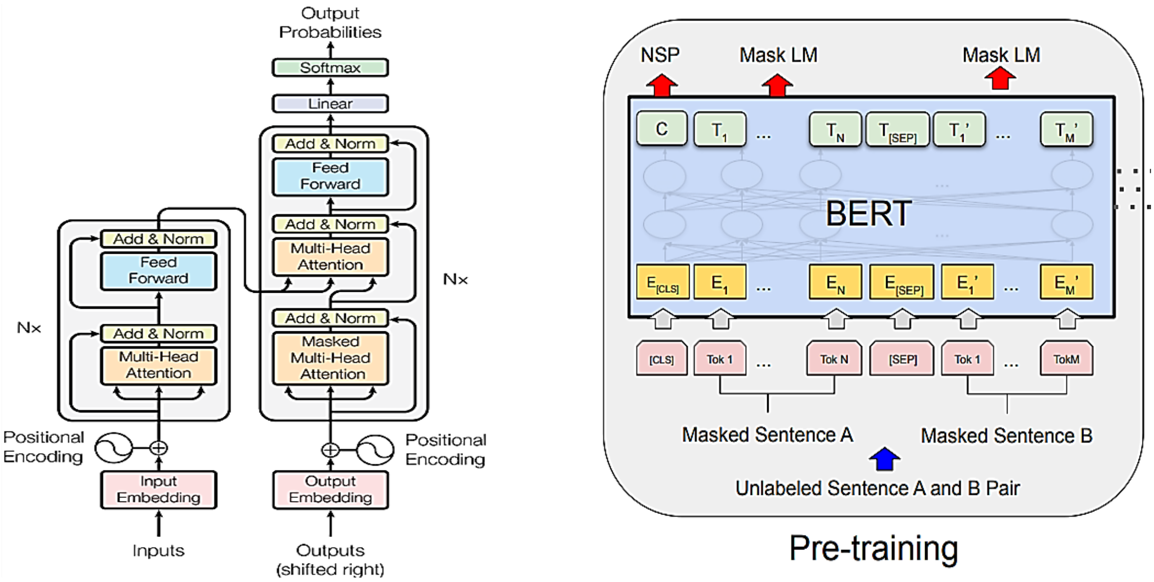
This will print a list of 32 punctuation marks, including: !"#$%&'()*+,-./:;<=>?@[]^_`{|}~
Tokenization. Tokenization is the process of splitting a string into a list of words. In Python, the NLTK library can be used to tokenize text and classify the part-of-speech (POS) tags for each word. Here is a breakdown of common POS tags:

DT (determiner): Refers to a specific noun.
JJ (adjective): Describes a noun.
NNS (noun, plural): Indicates more than one object.
VBZ (verb, third-person singular present): Describes an action performed by a third-person singular subject.
VBN (verb, past participle): Indicates a past or completed action.
IN (preposition): Shows spatial or temporal relationships between words.
NNP (proper noun, singular): Refers to a specific person, place, or thing.
Understanding these POS tags helps in analyzing the structure and meaning of a sentence. More details on POS tags can be found in Appendix C.

Lemmatization/stem. Lemmatization and stemming are techniques used in text processing to reduce words to their base forms. For example, "goes" is reduced to "go," and "stocks" to "stock." The script for lemmatization can be found in Appendix D.

Transformer and RoBERTa Model

The Transformer model was initially designed for machine translation and is now widely used for various Natural Language Processing (NLP) tasks, such as text classification, document summarization, and question answering [26]. It employs a 64-layer transformer network and causal attention to predict the next character. Despite being limited by a fixed input size of 512 characters, this model segments the input data and learns from each segment [27]. The architecture of the Transformer model is depicted in Figure 3 (A), while the BERT pre-training process is shown in Figure 3 (B).



(a) (b)

Figure 3. (a) Transformer model architecture [28], (b) BERT pre-training achitecture [29]

The Robustly Optimized BERT Pre-training Approach (RoBERTa) is a variant of BERT (Bidirectional Encoder Representations from Transformers) that enhances BERT by adjusting several parameters from the original version. RoBERTa utilizes a larger pre-training dataset, consisting of 160GB of English-based data. It employs dynamic masking and complete sentences without losing the Next Sentence Prediction (NSP) functionality [27]. An illustration of the RoBERTa model is provided in Figure 4(a).

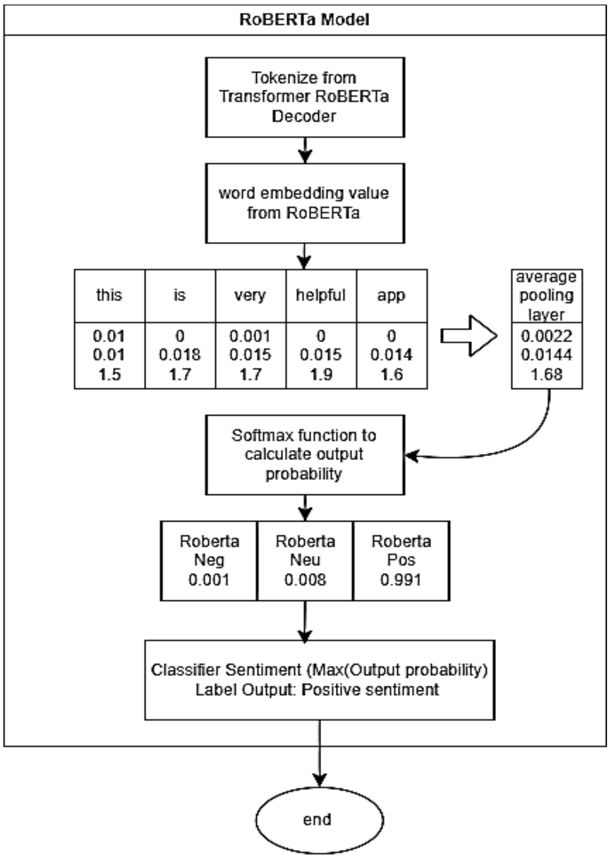


Figure 4. (a) Illustration of the RoBERTa model

Figure 4(a) presents a simplified illustration of how RoBERTa models perform sentiment classification on text. Pre-trained RoBERTa models assign vector values to tokenized words, a process known as word embedding. This process is performed using a neural network trained on vast amounts of data. Consequently, word embeddings in BERT and RoBERTa pre-trained models may differ. For example, in the RoBERTa model, word embeddings are categorized into three sentiment types: negative, neutral, and positive. For the sentence "this is a very helpful app," the word embedding values might be ["this" (0.01, 0.01, 1.5), "is" (0, 0.018, 1.7), "very" (0.001, 0.015, 1.7), "helpful" (0, 0.015, 1.5), "app" (0, 0.014, 1.6)]. These vector values are then averaged to produce the average pooling layer, which serves as input to the softmax calculation shown in equation (1) [28.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

Attention is a function used to map queries, track key-value pairs, and produce data in the form of vectors. In this formula, Q is the matrix that constructs the query (containing a vector for each word), is the key vector, and V is the value vector. d_k is the dimension of the key vector K. $\frac{QK^T}{\sqrt{d_k}}$ represents the step for calculating the attention weight. Softmax, an activation function in deep learning, is useful for determining attention weight, which leads to output probabilities between 0 and 1.

The softmax values are normalized to obtain the output probability values for the sentiment labels. The sentiment label is determined by selecting the maximum value from the output probability distribution. The RoBERTa model follows a similar process to BERT, but with distinct word embedding values. The prediction process for sentiment labels is also carried out in a similar manner.

During pre-training, BERT performs language modeling by predicting a certain percentage of masked tokens. However, a limitation of the original implementation is that the tokens selected to mask a given text sequence in different batches are sometimes the same. Figure 4(b) illustrates that even though BERT and RoBERTa models may have the same sentence structure, their word embedding vector values can differ, which may impact the probability values for predicting sentiment labels.

This paper employs pre-trained BERT and RoBERTa models for sentiment analysis tasks. These models were selected due to their robust language representation capabilities and exceptional performance in NLP tasks, particularly sentiment analysis. BERT and RoBERTa are trained on extensive text datasets, allowing them to effectively capture word-level and inter-word relationships within sentences. The flexibility of these pre-trained models also makes them well-suited for sentiment analysis tasks. In this study, the specific pre-trained models used are "bert-base-uncased" and "roberta-base-sentiment," both obtained from the Hugging Face library [29].

```
model_id="bert-base-uncased"
tokenizer = BertTokenizerFast.from_pretrained(model_id)
```

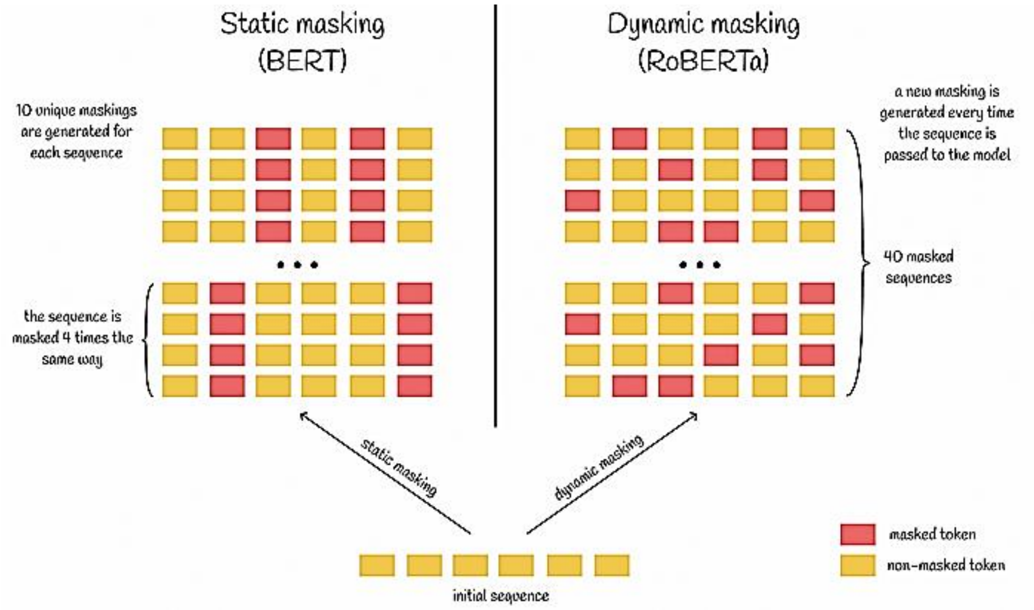


Figure 4. (b) Static masking (BERT) and dynamic masking (RoBERTa) [30]

Additionally, the RoBERTa model utilized in this study is "cardiffnlp/twitter-roberta-base-sentiment," also obtained from the Hugging Face library [31]

```
MODEL = f"cardiffnlp/twitter-roberta-base-sentiment"
tokenizer = AutoTokenizer.from_pretrained(MODEL)
model = AutoModelForSequenceClassification.from_pretrained(MODEL)
```

Filtering Using Keywords

Using Object-Oriented Programming (OOP) in Python, sentiment analysis results can be classified into positive and negative labels and filtered using specific keywords. This approach provides a benchmark for identifying user preferences in the retail mobile application. The categories and keywords, summarized from omni-channel retailing research [32], are presented in Table 1. The N-grams script used to filter the text using the keywords in Table 1 can be found in Appendix E.

Table 1. Topics and keywords from omni-channel retailing [32]

Topic Number	Topic Name	Keywords
1	Delivery and CRM	Delivery, offline, event, distribution, community, coupon, address, shipping, service
2	Economic value	Purchase, product, discount, price, online payment, point
3	Reviews and user experience	Review, recommendation, gift, easy, use, stock, call, app
4	Product quality and brand reputation	Quality, brand, sold, store, sales, order, shop, good or bad, excellent or worse

Delivery: Within the omnichannel retail landscape, efficient and reliable delivery processes are paramount. These processes encompass the entire journey from order placement to customer receipt of the product and include options like in-store pickup, home delivery, and even same-day delivery for select items. A seamless delivery experience is a key driver of customer satisfaction and repeat business.

Customer Relationship Management (CRM): Strong customer relationship management practices facilitate connections with customers across various channels. This enables the personalization of the customer experience and the effective addressing of their needs. By leveraging data from online purchases, browsing behavior, and past interactions, retailers can offer targeted promotions, loyalty programs, and exceptional customer service.

Economic Value: In the context of omnichannel retail, economic value refers to the perceived worth of a product or service relative to its cost to the customer.

Reviews: Online reviews from previous customers significantly influence consumer purchase decisions. Encouraging positive reviews and promptly addressing negative ones demonstrate transparency and a commitment to customer satisfaction.

User Experience: A positive user experience across all channels is crucial in omnichannel retail. This includes a user-friendly website, a functional mobile app, and a smooth in-store buying experience. Customers should be able to locate desired products easily, complete purchases efficiently, and receive helpful assistance when needed.

Product Quality: Consistent high-quality products foster trust and cultivate customer loyalty. Customers expect products to meet their expectations regarding performance, durability, and functionality.

Brand Reputation: A strong brand reputation extends beyond product quality. It encompasses customer service, ethical practices, and the overall brand image. Positive brand recognition fosters customer trust and incentivizes customers to choose the brand over competitors.

Results and Discussions

Web scraping from the Google Play Store provided the user review data for the KlikIndomaret Android application. The collected data includes the username (reviewer identity), score (star rating from 1 to 5, with 1 being very poor and 5 being excellent), date and time of review submission, and content (user comments). Data was collected from January 1 to April 15, 2024. For clarity, the web scraping results in DataFrame format are shown in Figure 5.

	ide	userName	score	at	content
0	1	Damayanti Satyaputri	1	4/15/2024 11:32	The service depends on the branch shop service...
1	2	Fransiskus Xaverius	5	4/15/2024 8:57	Excellent
2	3	robby cokro	5	4/15/2024 7:58	nice apps
3	4	Intan Resya Adelia	5	4/15/2024 5:21	Good
4	5	Kekeu Kirani Firdaus	1	4/15/2024 3:52	Stock is not updated and when you contact CS t...
...	...	...	...	...	...
515	516	Hasna M	1	1/1/2024 9:47	I ordered on schedule. When it was time, he sa...
516	517	yn (ms_yeona)	1	1/1/2024 8:10	Oh man, this is "free shipping"... Pay for the...
517	518	Ferany Ramdhani	5	1/1/2024 8:06	Satisfied with shopping at Click Indomaret, ea...
518	519	Tua Uttu	1	1/1/2024 6:09	There's no need to shop here... it's taking to...
519	520	Chamn Choi	1	1/1/2024 1:20	The stock application was available but after ...

520 rows × 5 columns

Figure 5. Web scraping results

Descriptive statistics of user ratings are presented in Figure 6 to highlight issues within the KlikIndomaret Android application. User ratings are distributed as follows: 1 (very poor) by 202 users, 2 (poor) by 26 users, 3 (fair) by 38 users, 4 (good) by 36 users, and 5 (excellent) by 218 users. A significant number of users still give poor ratings, nearly balancing those who provide satisfactory ratings. Previous studies have highlighted intriguing cases where app users have given high ratings but included reviews and comments with negative sentiment. Conversely, some users have given low ratings while providing positive comments. These findings cannot be fully captured by merely examining the descriptive statistics of app ratings [17]. Therefore, strategic steps are required to identify user complaints through the analysis of reviews and to devise strategies based on these insights.

Data cleaning involves removing unnecessary data, such as emoticons and other symbols that cannot be processed. The results of data cleaning are shown in Figure 7. The dataset comprises 520 user reviews categorized as 'STR' or String.

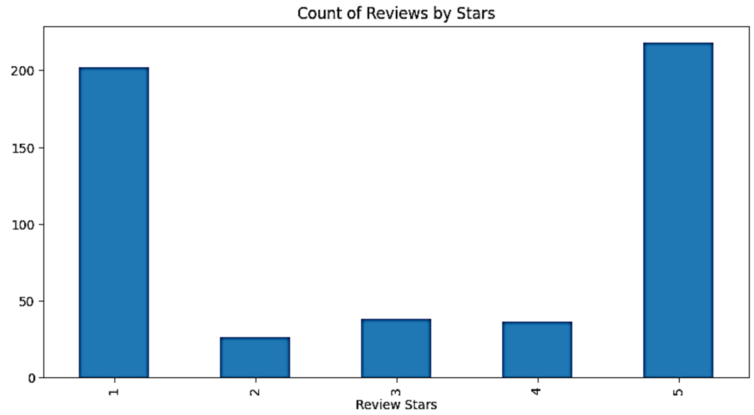
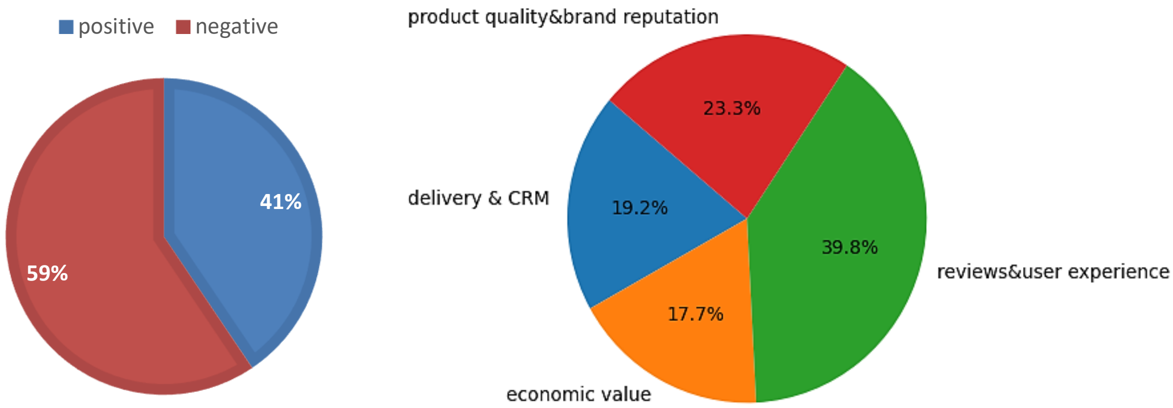


Figure 6. User rating statistics

```
0      The service depends on the branch shop service...
1                                             Excellent
2                                             nice apps
3                                                  Good
4      Stock is not updated and when you contact CS t...
                             ...
515    I ordered on schedule. When it was time, he sa...
516    Oh man, this is "free shipping"... Pay for the...
517    Satisfied with shopping at Click Indomaret, ea...
518    There's no need to shop here... it's taking to...
519    The stock application was available but after ...
Name: content, Length: 520, dtype: object
```

Figure 7. Data cleaning results for 520 rows of data

Figure 8 (A) presents statistics from the 520 user reviews processed with the RoBERTa Transformer model. The RoBERTa Transformer model classifies 211 reviews (49%) as positive sentiments and 309 reviews (59%) as negative sentiments. Meanwhile, Figure 8 (B) shows the review statistics based on Omni-Channel retailing topics and keywords [23]. Most users provided reviews related to general use at 39.8%, followed by product quality and brand reputation at 23.3%, delivery and Customer Relationship Management (CRM) at 19.2%, and economic value at 17.7%.



(a) (b)

Figure 8. (a) Sentiment percentage distribution, (b) Negative sentiment percentage by topic

Figure 9 displays word clouds of user reviews, where the size of each word indicates its frequency in the reviews. Larger words appear more frequently, highlighting aspects of application performance and omni-channel services in KlikIndomaret. Figure 9 (A) shows the word cloud for positive sentiment, with users praising the KlikIndomaret Android application using words such as helpful, good, fast app, easy, time, service, delivery, application, and excellent. Conversely, Figure 9 (B) shows the word cloud for negative sentiment, where user complaints frequently mention application issues, time, delivery, order, long wait times, payment issues, service, and refunds. For the word frequency details in the word cloud, refer to Appendix F in Table 2.

(a) (b)

Figure 9. (a) Word cloud for positive sentiments, (b) Word cloud for negative sentiments

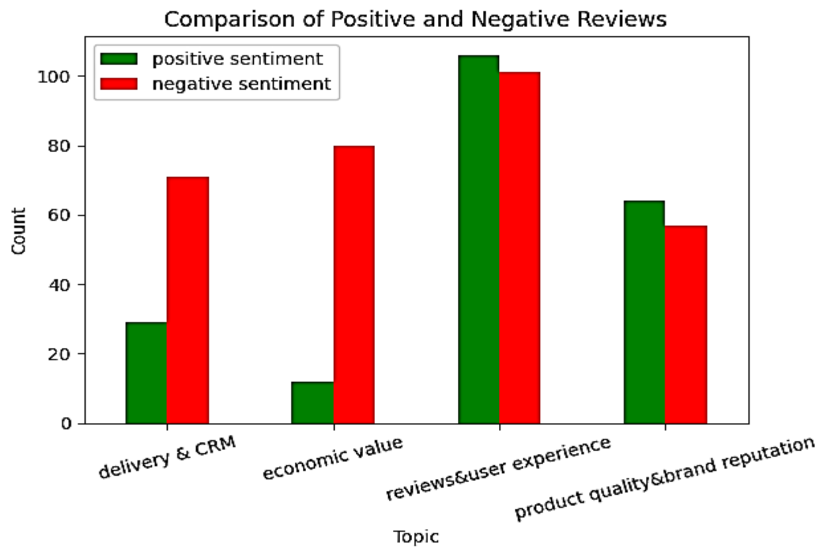


Figure 10. Comparison of positive and negative sentiments regarding omni-channel retail topics

After classifying the 520 user reviews of the KlikIndomaret Android application into positive and negative sentiments, these reviews were filtered based on topics and keywords from Table 1. Figure 10 compares positive and negative sentiments across various omni-channel retail topics. The topics of general use and product quality & brand reputation have balanced sentiments between positive and negative. However, the topics of economic value and delivery & CRM exhibit more negative sentiments than positive ones.

Regarding economic value, there are many complaints with negative sentiments, such as:

"Checkout takes a very long time (from selecting shipping options to payment). Pressed repeatedly, but there was no response from the application."

"Disappointing. Every time there is a promotion, the goods are never available. Even if they are, the payment process doesn't include the promos."

"Technically, the app has flaws. First, on the main screen, a popup said, 'Yay, there's a new coupon for you,' but when tapped, it turned out to be nothing."

"The coupon didn't come with the points; the wait was more than a day."

"I can't click the 'select payment' button, and it takes a long time to load. How about it, Indomaret? Please improve the system."

"'Select Payment' can't be clicked. What kind of application is this? You can only buy once, and the second purchase ends up with the same issue."

Regarding delivery & CRM, there are many negative sentiments in comments such as:

"The app said it could deliver, but the delivery notification failed because the address was not found."

"I made an order at 8 p.m., chose regular delivery for the next day at 9-10 a.m., but it still wasn't delivered by 1 p.m. Then, I was told via chat that the address was too far."

"The attitude of the shop and drivers needs improvement. Delivery often goes over schedule."

"This application is very helpful, but the delivery process and duration are very long. The location is so close, but I'm waiting for more than three hours!"

This indicates a need for improvement, beginning with economic value issues such as problematic payment processes, invalid discounts and promotions, and unredeemable loyalty points. In terms of delivery and CRM, improvements are needed in delivery processes related to timeliness and flexibility, offline store services, uneven product distribution, free shipping promotions, and the use of discount coupons in offline stores. The sentiment classification and text mining analysis results aim to provide management with a method to identify and formulate strategies for enhancing omnichannel retail business processes.

Future Work

Future work should explore the integration of text classification with business concepts, which holds considerable promise. Further investigation is needed into the performance of text classification algorithms, such as logistic regression and support vector machines. By comparing these algorithms with pre-trained BERT and RoBERTa models, it may be possible to refine text classification models based on user reviews for greater accuracy. Additionally, incorporating the Indonesian language into word embedding using the TF-IDF (Term Frequency-Inverse Document Frequency) concept could improve the transparency and clarity of vector value calculations.

Conclusions

Sentiment analysis can be used for various purposes, including understanding user review comments on Android applications in the omnichannel retail domain. The proposed model employs text mining for sentiment analysis using the Transformer model RoBERTa. This model is chosen for its ability to provide accurate compound score assessments that are contextually relevant, as well as for its pre-trained capabilities that eliminate the need for training and testing data against Bag-of-Words (BOW) models.

The user review data for the KlikIndomaret Android application, collected through web scraping, amounted to 520 reviews. The RoBERTa model classified 211 reviews as positive and 309 as negative. Filtering by topics and keywords revealed that the Economic Value and Delivery & CRM topics had the highest number of negative sentiments. The Economic Value topic is particularly highlighted for improvement due to frequent complaints about problematic payment processes, invalid discounts, unredeemable loyalty points, and refund processes. For the Delivery & CRM topic, improvements are needed in delivery processes concerning timeliness and flexibility, offline store services, uneven product distribution, free shipping promotions, and the use of discount coupons in offline stores. Text mining, combined with sentiment analysis and topic filtering, can accurately and precisely capture user feedback in the retail domain, helping management identify significant issues from user reviews on a large scale and devise appropriate strategies to enhance the KlikIndomaret Android application.

References

[1] A. Badia, D. Martín, and M. Gómez, "Teachers' perceptions of the use of Moodle activities and their learning impact in secondary education," *Technol. Knowl. Learn.*, vol. 24, pp. 483–499, 2019, doi: https://doi.org/10.1007/s10758-018-9354-3.

[2] S. J. Hartati and N. Sayidah, "The use of CIPP model for evaluation of computational algorithm learning program," in *Journal of Physics: Conference Series*, 2018, vol. 1088, no. 1, p. 12081, doi: 10.1088/1742-6596/1088/1/012081.

[3] D. Möhlenbruch, D. E. Blunck, and F. Ritschel, "Interactive Web 2.0 applications in the multi-channel marketing for retailers," *Eur. Retail Res.*, pp. 1–25, 2010, doi: https://doi.org/10.1007/978-3-8349-8938-3_1.

[4] S. Chopra, "The evolution of omni-channel retailing and its impact on supply chains," *Transp. Res. procedia*, vol. 30, pp. 4–13, 2018, doi: https://doi.org/10.1016/j.trpro.2018.09.002.

[5] S.-W. Huang, P.-F. Tu, W.-T. Fu, and M. Amanzadeh, "Leveraging the crowd to improve feature-sentiment analysis of user reviews," in *Proceedings of the 2013 international conference on Intelligent user interfaces*, 2013, pp. 3–14, doi: https://doi.org/10.1145/2449396.2449400.

[6] G. Basilaia and D. Kvavadze, "Transition to online education in schools during a SARS-CoV-2 coronavirus (COVID-19) pandemic in Georgia.," *Pedagog. Res.*, vol. 5, no. 4, 2020.

[7] S. De, S. Dey, S. Bhatia, and S. Bhattacharyya, "An introduction to data mining in social networks," in *Advanced data mining tools and methods for social computing*, Elsevier, 2022, pp. 1–25.

[8] A. Abbas, L. Zhang, and S. U. Khan, "A literature review on the state-of-the-art in patent analysis," *World Pat. Inf.*, vol. 37, pp. 3–13, 2014, doi: https://doi.org/10.1016/j.wpi.2013.12.006.

[9] C. Gallagher, E. Furey, and K. Curran, "The application of sentiment analysis and text analytics to

customer experience reviews to understand what customers are really saying," *Int. J. Data Warehous. Min.*, vol. 15, no. 4, pp. 21–47, 2019, doi: 10.4018/IJDWM.2019100102.
[10] B. Liu, *Sentiment analysis and opinion mining*. Springer Nature, 2022.
[11] K. L. S. Kumar, J. Desai, and J. Majumdar, "Opinion mining and sentiment analysis on online customer review," in *2016 IEEE international conference on computational intelligence and computing research (ICCIC)*, 2016, pp. 1–4, doi: 10.1109/ICCIC.2016.7919584.
[12] R. Ireland and A. Liu, "Application of data analytics for product design: Sentiment analysis of online product reviews," *CIRP J. Manuf. Sci. Technol.*, vol. 23, pp. 128–144, 2018, doi: https://doi.org/10.1016/j.cirpj.2018.06.003.
[13] S. Moghaddam, "Beyond sentiment analysis: mining defects and improvements from customer feedback," in *Advances in Information Retrieval: 37th European Conference on IR Research, ECIR 2015, Vienna, Austria, March 29-April 2, 2015. Proceedings 37*, 2015, pp. 400–410, doi: https://doi.org/10.1007/978-3-319-16354-3_44.
[14] M. McGuire and C. Kampf, "Using social media sentiment analysis to understand audiences: A new skill for technical communicators?," in *2015 IEEE International Professional Communication Conference (IPCC)*, 2015, pp. 1–7, doi: 10.1109/IPCC.2015.7235801.
[15] N. M. Rizk, A. Ebada, and E. S. Nasr, "Investigating mobile applications' requirements evolution through sentiment analysis of users' reviews," in *2015 11th International Computer Engineering Conference (ICENCO)*, 2015, pp. 123–130, doi: 10.1109/ICENCO.2015.7416336.
[16] L. Zhang, K. Hua, H. Wang, G. Qian, and L. Zhang, "Sentiment analysis on reviews of mobile users," *Procedia Comput. Sci.*, vol. 34, pp. 458–465, 2014, doi: https://doi.org/10.1016/j.procs.2014.07.013.
[17] A. G. Budianto, B. Wirjodirdjo, I. Maflahah, and D. Kurnianingtyas, "Sentiment analysis model for klikindomaret android app during pandemic using vader and transformers NLTK library," in *2022 IEEE international conference on industrial engineering and engineering management (IEEM)*, 2022, pp. 423–427, doi: 10.1109/IEEM55944.2022.9989577.
[18] S. Hu, A. Kumar, F. Al-Turjman, S. Gupta, and S. Seth, "Reviewer credibility and sentiment analysis based user profile modelling for online product recommendation," *IEEE Access*, vol. 8, pp. 26172–26189, 2020, doi: 10.1109/ACCESS.2020.2971087.
[19] E. W. Anderson and V. Mittal, "Strengthening the satisfaction-profit chain," *J. Serv. Res.*, vol. 3, no. 2, pp. 107–120, 2000, doi: https://doi.org/10.1177/109467050032001.
[20] I. Khan, "Impact of customers satisfaction and customers retention on customer loyalty," *Int. J. Sci. Technol. Res.*, vol. 1, no. 2, pp. 106–110, 2012.
[21] D. Wahjudi, T. Kwanda, and R. Sulis, "The impact of after-sales service quality on customer satisfaction and customer loyalty of middle-upper class landed housings," *J. Tek. Ind.*, vol. 20, no. 1, pp. 65–72, 2018, doi: https://doi.org/10.9744/jti.20.1.65-72.
[22] Y. Liu *et al.*, "Roberta: A robustly optimized bert pretraining approach," *arXiv Prepr. arXiv1907.11692*, 2019, doi: https://doi.org/10.48550/arXiv.1907.11692.
[23] Y. Sun *et al.*, "Ernie: Enhanced representation through knowledge integration," *arXiv Prepr. arXiv1904.09223*, 2019, doi: https://doi.org/10.48550/arXiv.1904.09223.
[24] S.-C. Chen and S. S. C. Shang, "Sustaining user experience in a smart system in the retail industry," *Sustainability*, vol. 13, no. 9, p. 5090, 2021, doi: https://doi.org/10.3390/su13095090.
[25] "KlikIndomaret on Google Play Store." https://play.google.com/store/apps/details?id=com.indomaret.klikindomaret&hl=en_US&gl=US (accessed Apr. 15, 2024).
[26] R. Al-Rfou, D. Choe, N. Constant, M. Guo, and L. Jones, "Character-level language modeling with deeper self-attention," in *Proceedings of the AAAI conference on artificial intelligence*, 2019, vol. 33, no. 01, pp. 3159–3166, doi: https://doi.org/10.1609/aaai.v33i01.33013159.
[27] F. A. Acheampong, H. Nunoo-Mensah, and W. Chen, "Transformer models for text-based emotion detection: a review of BERT-based approaches," *Artif. Intell. Rev.*, vol. 54, no. 8, pp. 5789–5829, 2021, doi: https://doi.org/10.1007/s10462-021-09958-2.
[28] A. Vaswani *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, doi: https://doi.org/10.48550/arXiv.1706.03762.
[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv Prepr. arXiv1810.04805*, 2018, doi: https://doi.org/10.48550/arXiv.1810.04805.
[30] V. Efimov, "Large Language Models: RoBERTa — A Robustly Optimized BERT approach," 2023. https://towardsdatascience.com/roberta-1ef07226c8d8 (accessed Jun. 28, 2024).
[31] F. Barbieri, J. Camacho-Collados, L. Neves, and L. Espinosa-Anke, "Tweeteval: Unified benchmark and comparative evaluation for tweet classification," *arXiv Prepr. arXiv2010.12421*, 2020, doi: https://doi.org/

10.48550/arXiv.2010.12421.
[32] S.-H. Kim and B.-K. Yoo, "Topics and sentiment analysis based on reviews of omni-channel retailing," *J. Distrib. Sci.*, vol. 19, no. 4, pp. 25–35, 2021, doi: https://doi.org/10.15722/jds.19.4.202104.25

Appendix

A. The script for web scrapping data KlikIndomaret on Google Play Store

```
!pip install google-play-scraper
from google_play_scraper import app
import pandas as pd
import numpy as np
#Scrape desired number of reviews

from google_play_scraper import Sort, reviews

result, continuation_token = reviews(
    'com.indomaret.klikindomaret',
    lang='en', # defaults to 'en'
    country='us', # defaults to 'us'
    sort=Sort.NEWEST, # defaults to Sort.MOST_RELEVANT you can use Sort.NEWEST to get newst
reviews
    count=524, # defaults to 100
    filter_score_with=None # defaults to None(means all score) Use 1 or 2 or 3 or 4 or 5 to
select certain score
df_idmrt = pd.DataFrame(np.array(result),columns=['review'])
df_idmrt = df_idmrt.join(pd.DataFrame(df_idmrt.pop('review').tolist()))
df_idmrt.head()
len(df_idmrt.index) #count the number of data we got
df_idmrt[['userName', 'score','at', 'content']].head()  #preview userName, rating, date-time,
and reviews only

#Run This Code to Sort the Data By Date
new_df = df_idmrt[['userName', 'score','at', 'content']]
sorted_df = new_df.sort_values(by='at', ascending=False) #Sort by Newst, change to True if
you want to sort by Oldest.
sorted_df.head()
my_df = sorted_df[['userName', 'score','at', 'content']] #get userName, rating, date-time,
and reviews only
my_df.to_csv("scrappedidmart_data.csv", index = False)  #Save the file as CSV , to download:
click the folder icon on the left. the csv file should be there.
```

B. Web scrapping result

	userName	score	at	content
0	Damayanti Satyaputri	1	2024-04-15 11:32:54	The service depends to the branch shop service...
1	Fransiskus Xaverius	5	2024-04-15 08:57:05	Excellent
2	robby cokro	5	2024-04-15 07:58:49	nice apps

C. Data cleaning for tokenization

```
import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')
nltk.download('words')
nltk.download('vader_lexicon')
tokens = nltk.word_tokenize(example)
tokens[:10]
```

['Fast', ',', 'good', 'goods', ',', 'normal', 'prices']

```
tagged = nltk.pos_tag(tokens)
tagged[:10]
```

[('Fast', 'NNP'),
(',', ','),

```
('good', 'JJ'),
('goods', 'NNS'),
(',', ','),
('normal', 'JJ'),
('prices', 'NNS')]
```

D. Data cleaning for lemmatization/stem in English

```
# import these modules for lemmatization/stem
import nltk
nltk.download('wordnet')
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

print("goes :", lemmatizer.lemmatize("goes"))
print("stocks :", lemmatizer.lemmatize("stocks"))

# a denotes adjective in "pos"
print("better :", lemmatizer.lemmatize("better", pos="a"))
goes : go
stocks : stock
better : good
```

E. N-grams filter OOP using python on Google Colab

```
import nltk

def generate_filtered_ngrams(text, keywords, n=2):
    # Preprocess text
    preprocessed_text = text.lower()
    preprocessed_text = re.sub(r'[^\w\s]', '', preprocessed_text)
    tokens = nltk.word_tokenize(preprocessed_text)

    # Generate n-grams
    ngrams = []
    for i in range(len(tokens) - n + 1):
        ngrams.append(tokens[i:i+n])

    # Filter n-grams based on keywords
    filtered_ngrams = []
    for ngram in ngrams:
        if any(keyword in ngram for keyword in keywords):
            filtered_ngrams.append(ngram)

    return filtered_ngrams

# Example usage using keyword on [24] list
text = "This is an example text with specific keywords to filter."
keywords = ["delivery", "courier"]

filtered_ngrams = generate_filtered_ngrams(text, keywords)
print(filtered_ngrams)
```

F. Word frequency for positive and negative word cloud

Table 2. Word frequency for word cloud

No.	Positive sentiment		Negative sentiment		No.	Positive sentiment		Negative sentiment	
	Word	Count	Word	Count		Word	Count	Word	Count
1	helpful	34	application	92	16	shop	8	want	28
2	good	28	even	55	17	lots	8	still	27
3	easy	23	indomaret	54	18	excellent	7	please	26
4	thank	23	time	53	19	great	7	stock	24
5	app	18	long	49	20	please	7	often	24
6	indomaret	18	order	44	21	always	6	service	23
7	shopping	17	delivery	42	22	buy	6	use	23
8	fast	14	really	41	23	best	6	will	23
9	time	13	app	40	24	use	6	goods	21
10	service	10	though	36	25	needs	6	payment	21
11	delivery	10	shop	34	26	apps	5	ordered	20
12	really	10	click	31	27	fast	5	via	20
13	nice	9	takes	31	28	especially	5	courier	20
14	application	8	slow	30	29	people	5	make	19
15	makes	8	items	29	30	courier	5	customer	19

The script for word cloud plot

```
# Import libraries

import pandas as pd
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt


# Load your DataFrame
from google.colab import drive
drive.mount('/content/drive')
# reading the data
data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/dataset/negatif sentimen.csv')
text_column = 'content' # Replace with the name of your text column

# Create a list of text from the DataFrame column
text_list = data[text_column].tolist()

# Join the text into a single string
text = ' '.join(text_list)

# Create a dictionary to store word frequencies
word_counts = {}
for word in text.lower().split():
  if word not in STOPWORDS and word not in ['the', 'a', 'an']: # Remove common words if
needed
    if word not in word_counts:
      word_counts[word] = 0
    word_counts[word] += 1

# Create the word cloud
wordcloud = WordCloud(width=800, height=600).generate_from_frequencies(word_counts)

# Create a plot of the word cloud
plt.figure(figsize=(8, 6))
plt.imshow(wordcloud, interpolation='nearest')
plt.axis('off')

# Display the word cloud
plt.show()

import csv

# Write word counts to CSV file
with open("word_counts.csv", "w", newline="") as csvfile:
    writer = csv.writer(csvfile)
    writer.writerow(["Word", "Count"])  # Header row
    for word, count in word_counts.items():
        writer.writerow([word, count])
```