JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

VOL. 11. NO. 2 NOVEMBER 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i2.7166

SENTIMENT ANALYSIS OF IT WORKERS ON NO CODE AND LOW CODE TRENDS: COMPARISON OF LSTM AND SVM MODELS

Yoga Handoko Agustin[1]*; Nabil Nur Afrizal[1]

Information System[1]
Institut Teknologi Garut, Garut, Indonesia[1]
https://www.itg.ac.id/ [1]
yoga.handoko@itg.ac.id*, 2107003@itg.ac.id

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract— *This research explores the sentiment of IT professionals toward the growing trend of No Code and Low Code technologies by comparing the performance of Support Vector Machine (SVM) and Long Short-Term Memory (LSTM) algorithms. Using the SEMMA methodology and automatic labeling with ChatGPT, a total of 4,238 comments were collected from Reddit and Twitter and categorized into positive, neutral, and negative sentiments. The analysis showed that neutral sentiment dominates on both platforms (47.9% on Reddit and 48.8% on Twitter), followed by positive sentiment (41.3% and 43.1%, respectively), indicating cautious but optimistic attitudes toward LCDPs. In terms of model performance, SVM outperformed LSTM with 87% accuracy and a weighted F1-score of 0.87, compared to LSTM's 80% accuracy and a weighted F1-score of 0.80. These findings confirm that classical machine learning methods remain highly effective for short-text sentiment analysis in social media, particularly when combined with TF-IDF feature representation, SMOTE balancing, and LLM-based automatic labeling, while also offering new insights into IT community perceptions of disruptive technologies.*

Keywords: *artificial intelligence, lowcode development, no-code development, sentiment analysis, social media.*

Intisari— *Penelitian ini mengeksplorasi sentimen para profesional TI terhadap tren teknologi No Code dan Low Code dengan membandingkan kinerja algoritma Support Vector Machine (SVM) dan Long Short-Term Memory (LSTM). Menggunakan metodologi SEMMA dan pelabelan otomatis berbasis ChatGPT, sebanyak 4.238 komentar berhasil dikumpulkan dari Reddit dan Twitter, kemudian dikategorikan ke dalam tiga sentimen: positif, netral, dan negatif. Hasil analisis menunjukkan bahwa sentimen netral mendominasi pada kedua platform (47,9% di Reddit dan 48,8% di Twitter), disusul sentimen positif (41,3% dan 43,1%). Distribusi ini mencerminkan sikap hati-hati namun optimis dari komunitas TI terhadap LCDP, di mana manfaat efisiensi dan aksesibilitas diakui, namun kekhawatiran terkait keamanan, reliabilitas sistem, dan dampak terhadap pekerjaan masih muncul. Dari sisi performa model, SVM menunjukkan hasil terbaik dengan akurasi 87% dan nilai F1 tertimbang 0,87, melampaui LSTM yang hanya mencapai akurasi 80% dan nilai F1 tertimbang 0,80. Temuan ini menegaskan bahwa pendekatan machine learning klasik masih sangat relevan untuk analisis sentimen teks pendek di media sosial, khususnya jika dikombinasikan dengan representasi fitur TF-IDF, penyeimbangan data SMOTE, serta pelabelan otomatis berbasis LLM. Selain memberikan kontribusi metodologis, penelitian ini juga menawarkan wawasan empiris terkait persepsi komunitas TI terhadap adopsi teknologi disruptif dalam ekosistem digital*.

Kata Kunci: *kecerdasan buatan, pengembangan kode rendah, pengembangan tanpa kode, analisis sentimen, media sosial.*

INTRODUCTION

The development of information technology in the last decade has changed the paradigm of software development. One of the rapidly growing innovations is the No Code and Low Code platform, which is an application development approach that allows the creation of systems without in-depth programming skills, by utilizing visual interfaces and drag-and-drop features. According to Bordicherla (2025) [1], No Code and Low Code platforms play a crucial role in democratizing the application development process by enabling non-technical users to design digital solutions independently. As noted in Parimi (2025) [2], Low Code platforms not only accelerate development but also reduce technical barriers for non-programmers in building digital solutions. This approach has gained popularity for addressing challenges related to time constraints, budget limitations, and shortages of skilled personnel in digital transformation processes.

Globally, the trend in the use of No Code and Low Code shows significant growth. It is estimated that 85% of new applications will be developed using this approach by 2024 As stated by Taunk (2025) [3]. In the Asia-Pacific region, around 68% of companies have started adopting modern development platforms, including Low Code and Generative AI technology integration findings from Outsystems (2024) [4]. In Indonesia itself, the adoption of this technology continues to increase in various sectors, but it still faces challenges such as integration limitations, security risks, and a lack of workers who are technically proficient in this platform as reported by CIO Insight Hub (2023) [5]. Similarly from Ajimati et al(2025) [6] highlights that the main challenges in adopting LCDP include vendor lock-in and limitations in flexibly customizing systems.

Responses to this technology among IT workers are also diverse. Most welcome it as an innovation in efficiency, while others view it as a threat to the stability of technical professions. Therefore, it is important to map the IT community's perceptions of this trend more systematically.

Previous studies have applied sentiment analysis in software engineering with promising results, but their focus has remained narrow. For instance, Obaidi et al. (2022) [7] demonstrated the effectiveness of advanced models such as BERT, achieving 94% accuracy with an F1-score of 83%, while Ahmed et al. (2024) [8] revealed that positive sentiment (64–68%) among developers correlates with higher code quality. Similarly, Zhang et al. (2024) [9] highlighted the strength of large language models (LLMs) in low-data scenarios. Although these studies underline important methodological advances, they do not specifically explore perceptions of disruptive technologies. Other works concentrated more on algorithmic comparisons. Asnawiyah and Putra (2024) [10] reported that BiLSTM slightly outperformed LSTM (60% vs. 58% accuracy) in multi-class sentiment classification on Twitter, while Ula and Fachrurrazi (2023) [11] found SVM to be more accurate than Naïve Bayes (72% vs. 69%) for detecting cyberbullying sentiment. However, these contexts were limited to general social media data or social issues rather than professional IT communities. Taken together, these findings indicate that despite technical progress in sentiment analysis, there remains a significant gap in examining IT workers' perceptions of disruptive platforms such as No Code and Low Code, especially through a comparative evaluation of classical models (SVM) and deep learning models (LSTM).

Building on these gaps, this study focuses on comparing two widely used classification algorithms—Support Vector Machine (SVM) and Long Short-Term Memory (LSTM)—to evaluate how effectively they capture IT community perceptions of No Code and Low Code technologies. The comparison is particularly relevant because SVM, as a classical machine learning model, often performs well on short and sparse text such as tweets, while LSTM, as a deep learning model, is designed to capture sequential dependencies and contextual nuances in longer discussions such as those found on Reddit. By combining both platforms, the study incorporates spontaneous and concise opinions from Twitter alongside more detailed and technical perspectives from Reddit.

To ensure reliable training data, this research employs automatic labelling with ChatGPT, which has been shown to provide consistent annotations and capture complex linguistic patterns more efficiently than manual or rule-based methods by Wang et al. (2024) [12]. Sentiments are categorized into positive, neutral, and negative classes, providing a balanced representation of community attitudes.

The research methodology follows the SEMMA framework (Sample, Explore, Modify, Model, Assess), which allows systematic handling of unstructured text data. Model evaluation is conducted using the F1-score metric, as it balances precision and recall and is particularly suitable for imbalanced sentiment classes. The findings are expected to provide both practical insights into IT workers' perceptions of disruptive platforms and

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 2 NOVEMBER 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i2.7166

academic contributions to the methodological debate on when classical approaches outperform deep learning in text-based sentiment analysis.

MATERIALS AND METHODS

This study uses the SEMMA (Sample, Explore, Modify, Model, Assess) methodological approach, which consists of five main stages that are carried out systematically and structurally. The SEMMA approach has also been used in previous studies to analyse text-based emotions, as done by Vuyyuru (2025) [13], which emphasizes its effectiveness in managing unstructured data from social media. This methodology was chosen because it provides a comprehensive framework for processing text-based data from social media. A study by Firas (2025) [14] shows that the SEMMA method is highly flexible and can be combined with other approaches such as CRISP-DM to handle big data, and has proven effective in the context of prediction, classification, and visualization of large-scale data patterns. Each phase in this approach is interconnected and iterative, supporting the construction of predictive classification models based on unstructured data, especially in the context of sentiment analysis of technology trends.

The Sampling stage was carried out by collecting data from two popular social media platforms, Reddit and Twitter. Data scraping was conducted using Twikit for Twitter and Instant Data Scraper for Reddit. Keywords such as "nocode", "lowcode", "bubble.io", and "microsoft power apps" were used to filter comments relevant to the research topic. The scraping process was conducted between April 17, 2024 and March 1, 2025, resulting in a total of 4,238 comments, consisting of 2,127 comments from Twitter and 2,111 from Reddit.

The Explore stage aimed to understand the initial characteristics of the dataset. Word frequency analysis was conducted to identify dominant terms appearing in the comments. The results were visualized in a word cloud to facilitate the interpretation of common discussion topics and the general sentiment of the IT community.

The Modify stage consisted of several preprocessing steps:

a. Cleaning irrelevant elements (URLs, emojis, numbers, punctuation).
b. Removing stop words.
c. Applying stemming to normalize words.
d. Tokenization to segment sentences into tokens.
e. Filtering to remove promotional content or non-genuine opinions.

Since the dataset was in English, no explicit mixed-language handling was required, but comments containing slang were preserved as they represent authentic community expressions.

For automatic sentiment labelling, ChatGPT was employed using the following prompt:

"Classify the following opinion as positive, negative, or neutral based on the implied attitude or sentiment in the text. Focus on the context of the opinion rather than only on positive/negative words."

This approach leveraged ChatGPT's ability to capture nuanced meanings and informal technical language, consistent with Belal et al (2023) [22] , who found that ChatGPT provided higher consistency and efficiency compared to manual labelling.

At the Model stage, two classification models were trained: Support Vector Machine (SVM) and Long Short-Term Memory (LSTM).

Support Vector Machine (SVM), For SVM, features were extracted using TF-IDF weighting. Kamalanathan (2021) [15], emphasized the role of word frequency and rarity in text representation, while Cahyani et al. (2021) [16] detailed its mathematical formulation:

$$TF - IDF(t,d) = TF(t,d)xlog(\frac{\mathrm{N}}{DF(t)}) \quad (1)$$

Formula 1. The TF-IDF calculation assigns weights to words based on their frequency of occurrence in documents and the number of documents containing those words.

The decision function in SVM is given by:

$$f(x) = sign\,(w.x + b) \quad (2)$$

Formula 2. The decision function for determining the optimal separation boundary between two classes in a high-dimensional feature space. Previous studies validate the relevance of SVM in social media sentiment classification. Arsi et al. (2021) [17] achieved strong results on Twitter data, while Zuriel (2021) [18] showed that the RBF kernel improved performance in classifying public opinion on policy.

Long Short-Term Memory (LSTM)

LSTM was employed to capture sequential dependencies in text. Its architecture includes memory cells and gating mechanisms by Okut (2021) [19]. This structure is mathematically described through activation functions as follows:

$$ft = \sigma(Wf \cdot [ht-1, xt] + bf) \quad (3)$$

Formula 3. (Forget gate) : determines which past information should be discarded, for example ignoring irrelevant tokens such as filler words or platform-specific hashtags.

$$it = \sigma(Wi \cdot [ht-1, xt] + bi) \quad (4)$$

Formula 4. (Input gate) : regulates which new information should be stored, such as recognizing emotionally charged words like "frustrating" or "efficient."

$$Ct = tanh(WC \cdot [ht-1, xt] + bC) \quad (5)$$

Formula 5. (Candidate value) : creates candidate representations of sentiment-related terms before they are updated into the memory cell.

$$Ct = ft \cdot Ct - 1 + it \cdot C{\sim}t \quad (6)$$

Formula 6. (Cell state update) : integrates past memory with newly relevant information, ensuring that important context—such as whether "easy" was used positively or sarcastically—is preserved.

$$ot = \sigma(Wo \cdot [ht-1, xt] + bo) \quad (7)$$

Formula 7. (Output gate) : decides which parts of the current state contribute to the output, e.g., determining that the presence of "time-saving" reflects a positive sentiment.

$$ht = ot \cdot tanh(Ct) \quad (8)$$

Formula 8. (Hidden state output) : produces the final hidden representation passed to the classifier, encoding both word meaning and contextual sentiment flow across a sentence.

This mechanism makes LSTM highly effective in sentiment analysis tasks where context and word order are important, such as distinguishing between *"no code tools save time"* (positive) and *"no code tools save time but lack flexibility"* (mixed/negative). The LSTM model in this study was configured with two stacked layers consisting of 128 hidden units. A dropout rate of 0.2 was applied to reduce overfitting, and the input was represented using an embedding dimension of 100. The model was trained with a batch size of 32 using the Adam optimizer (learning rate 0.001) for 15 epochs. The effectiveness of LSTM in capturing contextual meaning and dynamic public opinion has also been confirmed by prior studies. For example, study by Dewi et al (2023) [20], applied LSTM to classify sentiment related to COVID-19 vaccination in Indonesia and achieved significant results in understanding complex public opinion. Similarly, Shao (2025) [21] demonstrated that combining CNN and LSTM architectures improved accuracy in film sentiment classification, reinforcing the role of LSTM in handling long-term dependencies.

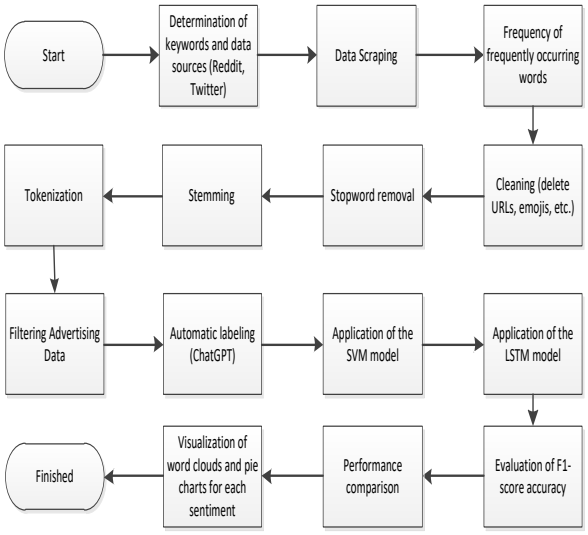
In this study, automatic labelling was conducted using ChatGPT due to its capability in recognizing semantic context and emotional nuances in informal technical language. The following prompt was applied to guide the labelling process: *"Classify the following opinion as positive, negative, or neutral based on the implied attitude or sentiment in the text. Focus on the context of the opinion rather than only on positive/negative words."* Similar to Belal et al. (2023) [22], ChatGPT provided higher consistency and efficiency compared to manual labelling.

The final stage, Assess, focused on evaluating the performance of both classification models. Evaluation was carried out using precision, recall, and the F1-score metric, with the latter chosen as the primary indicator because it balances accuracy and completeness, particularly under imbalanced class distributions. This aligns with the findings of Sitarz (2023) [23], who emphasized F1-score as the most reliable metric in such contexts. The formula is expressed as follows:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (9)$$

Formula 9. Calculation of the F1-score as an evaluation metric that balances precision and recall for imbalanced data.

In addition to numerical evaluation, results were visualized using pie charts to illustrate sentiment proportions and word clouds to highlight dominant words within each sentiment class. To summarize the SEMMA process implemented in this research, a flowchart is presented in Figure 1.



Source: (Research Results, 2025)

Figure 1. SEMMA-Based Research Flowchart

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 2 NOVEMBER 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i2.7166

This diagram illustrates the overall research pipeline, starting from data source identification and keyword selection, followed by scraping, preprocessing (cleaning, stop word removal, stemming, tokenization), data filtering, and automatic labelling using ChatGPT. The flow then continues to the application of two classification models (SVM and LSTM), their performance evaluation using the F1-score, and finally, sentiment visualization through pie charts and word clouds. Each stage is interconnected to provide a holistic sentiment analysis of IT community discussions.

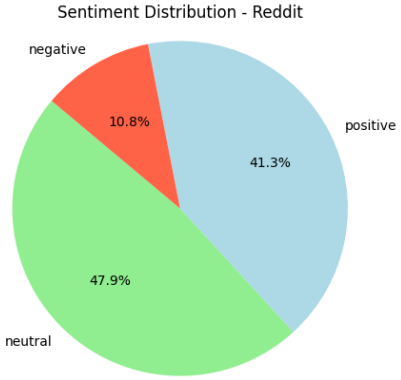
By comprehensively following SEMMA, this research not only ensures reliable sentiment classification but also provides a methodological comparison between machine learning and deep learning approaches, contributing to the understanding of socio-technical trends such as No Code and Low Code.

RESULTS AND DISCUSSION

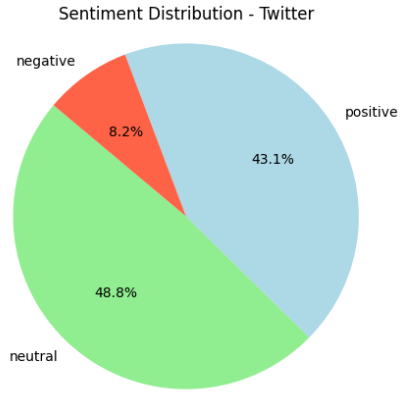
Sentiment analysis was conducted on a total of 4,238 comments from Reddit and Twitter, which had previously undergone preprocessing and automatic labelling using ChatGPT. Based on the classification results, visualization and model performance evaluation were carried out to understand how the IT worker community responds to No Code and Low Code trends more comprehensively.

On the Reddit platform (Figure 1), neutral sentiment was the most dominant with a percentage of 47.9%, followed by positive sentiment at 41.3%, and negative sentiment at 10.8%. Meanwhile, on the Twitter platform (Figure 2), the sentiment distribution showed a fairly similar pattern but with a slight shift. Neutral sentiment remains the highest at 48.8%, followed by positive sentiment at 43.1%, and negative sentiment at 8.2%.

This indicates that most comments are informative or descriptive, not showing strong emotional expressions toward the LCDP topic. The relatively high positive sentiment also indicates acceptance and enthusiasm for this technological development, although a small number of negative comments still appear, which are generally related to concerns about security, feature limitations, and the potential impact on developer jobs. Overall, this distribution suggests that the IT community does not immediately reject innovation but instead demonstrates a cautious and critical attitude in evaluating the benefits and challenges of new technology.



Source: (Research Results, 2025)
Figure 2. Reddit Sentiment Distribution



Source: (Research Results, 2025)
Figure 3. Twitter Sentiment Distribution

This dominance of neutral sentiment indicates that many IT workers are still in an observation phase regarding the adoption of Low-Code/No-Code platforms. According to the Diffusion of Innovation theory, such a pattern is common in the early stages of technology adoption, where users prefer to monitor and evaluate before expressing strong support or resistance. The relatively high proportion of positive sentiment reflects optimism toward the efficiency and accessibility offered by LCDPs, aligning with Bodicherla (2025) [2] and Parimi (2025) [3], who emphasized their role in democratizing application development. On the other hand, the small proportion of negative sentiment is consistent with Ahmed et al. (2024) [8], which highlighted that concerns about security, vendor lock-in, and job displacement often drive resistance to new software tools. These results suggest that the IT community demonstrates cautious optimism: while the potential benefits of LCDPs are widely acknowledged, sustainable adoption will depend on how effectively risks and limitations are addressed.

To evaluate the performance of the classification models, both Support Vector Machine (SVM) and Long Short-Term Memory (LSTM) were tested using precision, recall, and F1-score metrics. The results are presented in Table 1.

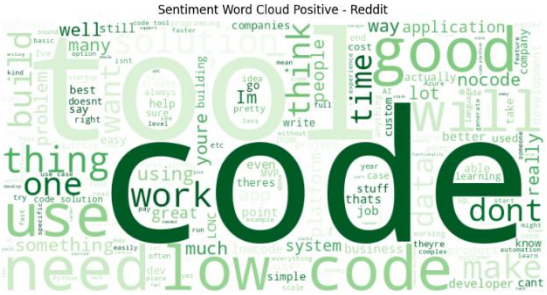
Table 1. SVM and LSTM Model Evaluation

Model	Sentiment	Precision	Recall	F1-Score	Support
SVM	Negative	0.98	0.99	0.98	318
	Neutral	0.83	0.80	0.81	317
	Positive	0.82	0.83	0.82	318
	Weighted Avg.	0.87	0.87	0.87	-
	Accuracy	-	-	-	**87%**
LSTM	Negative	0.95	0.92	0.93	318
	Neutral	0.69	0.80	0.74	317
	Positive	0.78	0.68	0.73	318
	Weighted Avg.	0.81	0.80	0.80	-
	Accuracy	-	-	-	**80%**

Source: (Research Results, 2025)

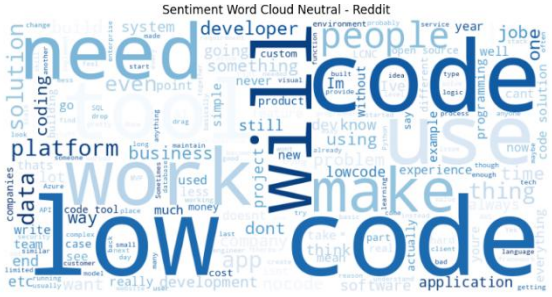
As shown in Table 1, SVM achieved the best overall performance with an accuracy of 87% and a weighted F1-score of 0.87. The model demonstrated particular strength in detecting negative sentiment, reaching an F1-score of 0.98. LSTM, while competitive in the negative sentiment class (F1 = 0.93), performed less consistently for neutral and positive sentiments, resulting in a lower overall accuracy of 80%. These findings indicate that SVM, supported by TF-IDF feature representation and SMOTE balancing, is more effective for short and sparse social media texts such as tweets, while LSTM's contextual advantages are less pronounced in this dataset.

Word cloud visualizations provide deeper insights into the context behind the sentiment distribution. In Reddit's positive sentiment (Figure 4), words such as *"tool," "use,"* and *"platform"* dominate, suggesting that IT workers perceive LCDPs as practical instruments for simplifying development. This reinforces the democratization aspect highlighted by Bodicherla (2025) [2], where ease of use and reduced technical barriers become primary attractions.

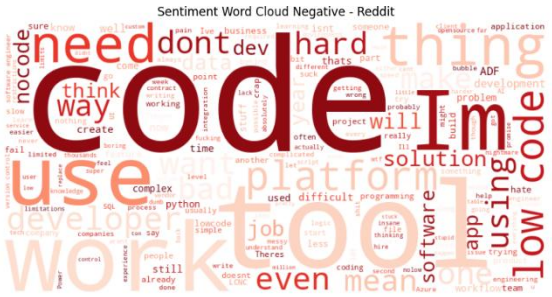


Source: (Research Results, 2025)
Figure 4 Positive Reddit Sentiment Word Cloud

Conversely, the negative sentiment cloud (Figure 6) displays words such as *"problem," "complex,"* and *"fail,"* which represent recurring concerns about technical stability and workflow disruption. These concerns echo Ajimati et al. (2025) [6], who emphasized vendor lock-in and customization limits as major challenges. Neutral sentiment (Figure 5), with frequent terms like *"tool"* and *"data,"* reflects exploratory discussions, suggesting that many users are still in an evaluative stage, consistent with the early adoption phase in the Diffusion of Innovation theory.



Source: (Research Results, 2025)
Figure 5 Reddit Neutral Sentiment Word cloud



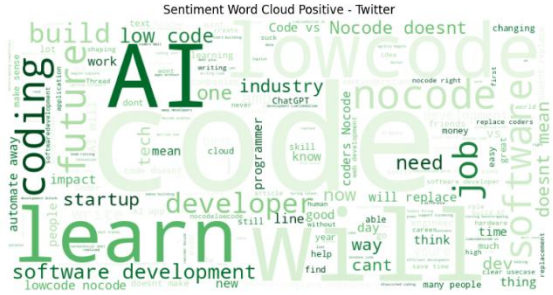
Source: (Research Results, 2025)
Figure 6 Reddit Negative Sentiment Word cloud

On Twitter, positive sentiment (Figure 7) highlights words such as *"AI"* and *"future,"* which illustrate enthusiasm about the integration of LCDPs with emerging technologies and optimism about their impact on developer productivity. This reflects how many practitioners view LCDPs not only as tools for simplifying coding but also as enablers of innovation in the era of artificial intelligence. Neutral sentiment (Figure 8) includes descriptive terms such as *"development"* and *"application,"* indicating information-sharing behavior rather than evaluative stances, for example through the exchange of platform comparisons or the dissemination of related articles. Negative sentiment (Figure 9), dominated by terms like *"job"* and *"impact,"* reflects anxieties about automation and job displacement, aligning with findings by Ahmed et al. (2024) [8] who identified workforce concerns as a recurring barrier

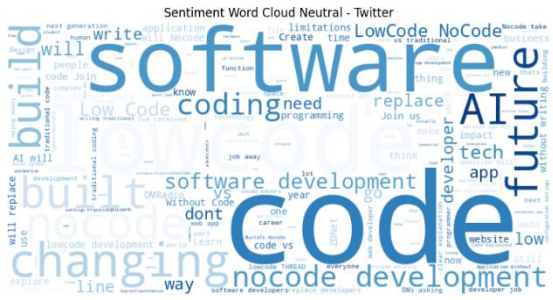
Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 2 NOVEMBER 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i2.7166

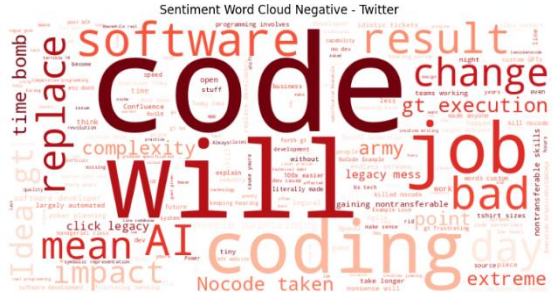
to adoption. These results suggest that while Twitter users are generally optimistic about the role of LCDPs in shaping the future of development, there remains an undercurrent of concern regarding long-term implications for employment stability and the technical maturity of these platforms.



Source: (Research Results, 2025)
Figure 7 Word cloud of Positive Twitter Sentiment

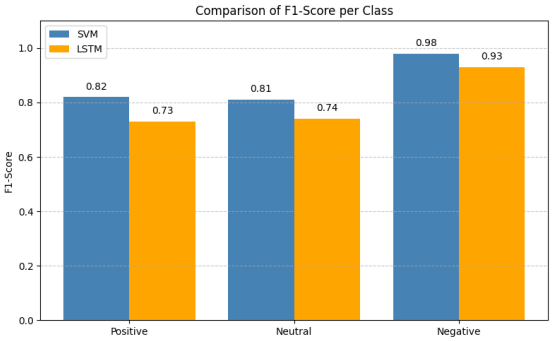


Source: (Research Results, 2025)
Figure 8 Neutral Twitter Sentiment Word cloud



Source: (Research Results, 2025)
Figure 9 Negative Twitter Sentiment Word cloud

Figure 10 illustrates a direct comparison of F1-scores between SVM and LSTM across sentiment classes. The results show that SVM consistently outperforms LSTM, particularly in the positive and neutral categories. For positive sentiment, SVM achieved an F1-score of 0.82, while LSTM only reached 0.73. A similar pattern is observed in the neutral class, where SVM obtained 0.81 compared to LSTM's 0.74. The gap between the two models is smaller in the negative sentiment category, yet SVM still leads with 0.98 versus 0.93 for LSTM.



Source: (Research Results, 2025)
Figure 10. Comparison of F1-Score for Each Class of SVM vs. LSTM

These findings confirm that although LSTM is theoretically designed to capture sequential dependencies in text, its advantage is less evident when handling short and sparse comments commonly found on social media platforms. In contrast, SVM combined with TF-IDF representation and SMOTE balancing demonstrates superior stability and precision, especially for moderate sentiments where contextual cues are limited.

The results highlight that classical approaches like SVM remain highly competitive for short-text sentiment analysis, while deep learning models such as LSTM may require richer context or longer text inputs to fully realize their potential. Moreover, the use of ChatGPT in automatic labelling contributed to the consistency of training data, ensuring that both emotional tone and technical content were effectively captured. Overall, this integration of classical machine learning, deep learning, and LLM-based labelling within the SEMMA framework provides a reliable approach to understanding IT workers' perceptions of disruptive technologies such as Low-Code and No-Code platforms.

CONCLUSION

This study concludes that the IT worker community's perceptions of Low-Code/No-Code platforms are dominated by neutral and positive sentiments, indicating a stage of cautious evaluation alongside optimism for the efficiency and accessibility offered by LCDPs. The Support Vector Machine (SVM) model demonstrated the best performance with 87% accuracy and a weighted F1-score of 0.87, surpassing the Long Short-Term Memory (LSTM) model with 80% accuracy and a weighted F1-score of 0.80. These findings highlight the continued relevance of classical machine learning approaches, particularly when supported

by TF-IDF and SMOTE, in handling short-text sentiment analysis on social media, where data tends to be sparse and fragmented. Practically, this implies that the successful adoption of LCDPs will depend on addressing persistent concerns such as system reliability, job displacement, and security risks, while theoretically contributing to the literature by showing that online discourse reflects cautious consideration rather than polarized acceptance or rejection. Nonetheless, the study is limited by its reliance on Reddit and Twitter as data sources within a restricted period and the use of automatic labelling without extensive manual validation, which may introduce demographic or linguistic bias. Future research should expand to multilingual and cross-platform datasets, apply more advanced approaches such as transformer-based or ensemble models, and explore the use of different large language models for labelling to enhance generalizability.

REFERENCE

[1] B. Bodicherla, "The Rise of Low-Code/No-Code Development: Democratizing Application Development," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 11, no. 2, Art. no. 2, Mar. 2025, doi: 10.32628/CSEIT251112398.

[2] S. Parimi, "Impact of Low-Code/No-Code Platforms," ResearchGate. Accessed: Apr. 19, 2025. [Online]. Available: https://www.researchgate.net/publication/388931508_Impact_of_Low-CodeNo-Code_Platforms

[3] M. Taunk, "No Code Statistics - Market Growth & Predictions (Updated 2025)." Accessed: Apr. 15, 2025. [Online]. Available: https://codeconductor.ai/blog/no-code-statistics/

[4] OutSystems, "New Research Highlights AI and Low-Code Synergy Accelerating Application Development in Asia-Pacific - PR Newswire APAC." Accessed: May 30, 2025. [Online]. Available: https://en.prnasia.com/story/467249-0.shtml

[5] CIO Insight Hub, "The Hidden Dangers of Low-Code/No-Code Platforms: Understanding Security Risks," CIO Insight Hub. Accessed: Apr. 15, 2025. [Online]. Available: https://ciohub.org/post/2023/03/low-code-no-code-platform-security-risks/

[6] M. O. Ajimati, N. Carroll, and M. Maher, "Adoption of low-code and no-code development: A systematic literature review and future research agenda," *Journal of Systems and Software*, vol. 222, p. 112300, Apr. 2025, doi: 10.1016/j.jss.2024.112300.

[7] M. Obaidi, L. Nagel, A. Specht, and J. Klünder, "Sentiment Analysis Tools in Software Engineering: A Systematic Mapping Study," *Information and Software Technology*, vol. 151, p. 107018, Nov. 2022, doi: 10.1016/j.infsof.2022.107018.

[8] M. S. Ahmed, D. Park, and N. U. Eisty, "Sentiment Analysis of ML Projects: Bridging Emotional Intelligence and Code Quality," Sep. 26, 2024, *arXiv*: arXiv:2409.17885. doi: 10.48550/arXiv.2409.17885.

[9] T. Zhang, I. C. Irsan, F. Thung, and D. Lo, "Revisiting Sentiment Analysis for Software Engineering in the Era of Large Language Models," Sep. 07, 2024, *arXiv*: arXiv:2310.11113. doi: 10.48550/arXiv.2310.11113.

[10] Asnawiyah and R. E. Putra, "Perbandingan Algoritma LSTM dan BiLSTM Untuk Analisis Sentimen Multi-Class Media Sosial Twitter," *Journal of Informatics and Computer Science (JINACS)*, pp. 778–786, Jan. 2024.

[11] M. Ula and S. Fachrurrazi, "Analisis Sentimen Cyberbullying pada Media Sosial Twitter menggunakan Metode Support Vector Machine dan Naïve Bayes Classifier," *techsi*, vol. 14, no. 2, p. 91, Dec. 2023, doi: 10.29103/techsi.v14i2.12103.

[12] Z. Wang, Q. Xie, Y. Feng, Z. Ding, Z. Yang, and R. Xia, "Is ChatGPT a Good Sentiment Analyzer? A Preliminary Study," Feb. 17, 2024, *arXiv*: arXiv:2304.04339. doi: 10.48550/arXiv.2304.04339.

[13] M. Vuyyuru, "Emotion Detection from Text: A SEMMA Methodology Approach," Medium. Accessed: Apr. 19, 2025. [Online]. Available: https://medium.com/@moukthikareddy.vuyyuru/title-emotion-detection-from-text-a-semma-methodology-approach-eddf2797a67a

[14] O. Firas, "A combination of SEMMA & CRISP-DM models for effectively handling big data using formal concept analysis based knowledge discovery: A data mining approach." Accessed: Apr. 19, 2025. [Online]. Available: https://www.researchgate.net/publication/367543011_A_combination_of_SEMMA_CRISP-DM_models_for_effectively_handling_big_dat

VOL. 11. NO. 2 NOVEMBER 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i2.7166

a_using_formal_concept_analysis_based_knowledge_discovery_A_data_mining_approach
[15] M. Das, S. Kamalanathan, and P. Alphonse, "A Comparative Study on TF-IDF feature Weighting Method and its Analysis using Unstructured Dataset," 2021.
[16] D. E. Cahyani and I. Patasik, "Performance comparison of TF-IDF and Word2Vec models for emotion text classification," *Bulletin EEI*, vol. 10, no. 5, pp. 2780–2788, Oct. 2021, doi: 10.11591/eei.v10i5.3157.
[17] P. Arsi and R. Waluyo, "Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan Algoritma Support Vector Machine (SVM)," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 8, no. 1, pp. 147–156, Feb. 2021, doi: 10.25126/jtiik.0813944.
[18] H. P. P. Zuriel and A. Fahrurozi, "IMPLEMENTASI ALGORITMA KLASIFIKASI SUPPORT VECTOR MACHINE UNTUK ANALISA SENTIMEN PENGGUNA TWITTER TERHADAP KEBIJAKAN PSBB," *Jurnal Ilmiah Informatika Komputer*, vol. 26, no. 2, Art. no. 2, Aug. 2021, doi: 10.35760/ik.2021.v26i2.4289.
[19] H. Okut, "Deep Learning for Subtyping and Prediction of Diseases: Long-Short Term Memory," in *Deep Learning Applications*, P. Luigi Mazzeo and P. Spagnolo, Eds., IntechOpen, 2021. doi: 10.5772/intechopen.96180.
[20] K. K. Dewi and S. Winiarti, "Analisis Sentimen Menggunakan Long Short-Term Memory Terkait Vaksinasi Covid-19 Di Indonesia," *JSTIE*, vol. 11, no. 3, p. 124, Oct. 2023, doi: 10.12928/jstie.v11i3.27188.
[21] S. Shao, "Enhancing Sentiment Analysis with a CNN-Stacked LSTM Hybrid Model," *ITM Web Conf.*, vol. 70, p. 02002, 2025, doi: 10.1051/itmconf/20257002002.
[22] M. Belal, J. She, and S. Wong, "Leveraging ChatGPT As Text Annotation Tool For Sentiment Analysis," Jun. 18, 2023, *arXiv*: arXiv:2306.17177. doi: 10.48550/arXiv.2306.17177.
[23] M. Sitarz, "Extending F1 metric, probabilistic approach," *AAIML*, vol. 03, no. 02, pp. 1025–1038, 2023, doi: 10.54364/AAIML.2023.1161.