Designing a Library Chatbot Using Artificial Neural Network Methods to Improve Visitor Services.

If'amunnuri Al Aghutsy[1*], Evy Poerbaningtyas[2], Syntia Widyayuningtyas Putri Listio[3]

[1,2,3] *Informatic, Universitas Bhinneka Nusantara, Tidar 100 Street Malang, Indonesia*

Keywords

Chatbot; Artificial Neural Network; Digital Library; Natural Language Processing; Service

****Correspondence Email:***

evip@ubhinus.com

Abstract

Academic libraries increasingly struggle to provide consistent information services as digital collections grow and user inquiries become more varied. Staff-dependent communication channels such as email and live chat are constrained by working hours, leaving visitors without support during evenings, weekends, and holidays. This study proposes an intelligent text-based chatbot to automate visitor inquiry handling in a university digital library setting. The system employs an Artificial Neural Network (ANN) with a multi-layer Perceptron architecture trained on a corpus of 2,319 labeled samples across 56 intent categories. An Indonesian-language preprocessing pipeline was implemented, consisting of case folding, tokenization, stopword removal, and morphological stemming using the Sastrawi algorithm, followed by Bag-of-Words feature extraction producing 1,494-dimensional vectors. The system was deployed using a decoupled architecture, separating the Python Flask inference backend from a PHP CodeIgniter 4 frontend via an Ngrok HTTPS tunnel. Using an 80:20 stratified train-test split, the model achieved a classification accuracy of 91.00%, with a macro-average precision of 0.90 and recall of 0.92 on 464 test samples. Black-box functional testing confirmed stable real-time performance with response latency under three seconds. These results demonstrate that ANN-based intent classification can effectively reduce reliance on manual library staff for routine inquiries while providing continuous 24/7 automated support.

1. Introduction

The rapid structural expansion of digital information repositories across modern academic infrastructures has transformed user communication expectations within digital library environments. While available digital collection catalogs continue to experience exponential growth, library visitors face significant search friction when navigating static information frameworks, which underpins the necessity for automated information management integration [1]. Traditional lookup architectures rely heavily on direct keyword inputs, often rendering highly rigid or irrelevant results that complicate student attempts to find baseline operational data. This discrepancy accentuates a pressing institutional requirement for intuitive and interactive retrieval layers capable of elevating digital student communication, particularly inside localized academic management structures [2].

Presently, real-time communication networks across standard library portals remain constrained by manual human workflows such as distributed emailing loops, electronic messaging channels, or community forums managed by live librarian shifts. Although these avenues are universally recognized, they present deep-seated systemic deficiencies that degrade service reliability [3]. Human response loops are structurally bound by specific working hours, meaning urgent questions regarding administrative workflows, book availability, or structural operating times encounter prolonged delays over late hours, weekends, or academic holidays. When a student faces a critical deadline and requires instant clarity regarding digital resource procedures or specific book locations, a delayed reply often compromises the user experience. Furthermore, static informational models such as Frequently Asked Questions (FAQ) list arrays demonstrate degraded functional efficiency as information volumes expand. In accordance with the principles of Hick's Law, the cognitive time required for a user to isolate a specific point of reference increases monotonically with the quantity of distinct choices displayed simultaneously on a single page. Conversely, text-driven conversational user interfaces (CUI) minimize navigational pathways by automatically serving precise information vectors directly within real-time dialogue blocks [4].

To overcome these structural bottlenecks, this research designs and implements an interactive automated agent built entirely upon an Artificial Neural Network (ANN) framework combined with Indonesian Natural Language Processing (NLP) workflows. Prior text-parsing implementations have attempted to incorporate computational mechanisms such as rule-based pattern matching, Naive Bayes models, or K-Nearest Neighbors (KNN) configurations to handle customer support questions [5]. However, classical rule-based architectures display fragile fault tolerance when exposed to the inherent variability of natural human language, collapsing when encountering lexical typos, morphological adjustments, or missing keyword linkages [6]. By leveraging an Artificial Neural Network trained on localized corpus text, the resulting chatbot system gains the mathematical capability to extract complex, non-linear syntactic patterns from messy user input queries [7].

While deep learning architectures like large-scale transformer variants have shown state-of-the-art proficiency in broad language benchmarks, their real-time application in regional library settings remains hindered by high computational overhead, expensive infrastructure costs, and tendency toward hallucinatory replies [8]. Most importantly, a critical research gap exists within Indonesian library infrastructures regarding the architectural balance between high multi-class intent granularity (exceeding 50 unique categories) and efficient local web server integration. To address this exact operational threshold, this study implements a complete multi-layered Perceptron model mapped against 56 unique intent targets. The structural novelty of this study lies in the formulation of an optimized, Indonesian-specific text preprocessing pipeline incorporating the Nazief & Adriani stemming rule via the Sastrawi toolkit paired with a completely Decoupled Architecture (Python Flask back-end paired to a PHP CodeIgniter 4 front-end via secure HTTPS Ngrok tunneling). This deployment maximizes computational performance and security by keeping the core neural network layers completely distinct from the primary database. This modular platform addresses the dual demand of high operational efficiency and immediate, real-time context parsing for digital library visitors.

2. Research Methods

This study executed an end-to-end software development and engineering pipeline consisting of data collection, text normalization, neural network design, training dynamics exploration, and comprehensive black-box functional testing over a structured timeline.

2.1 Decoupled Architecture Design

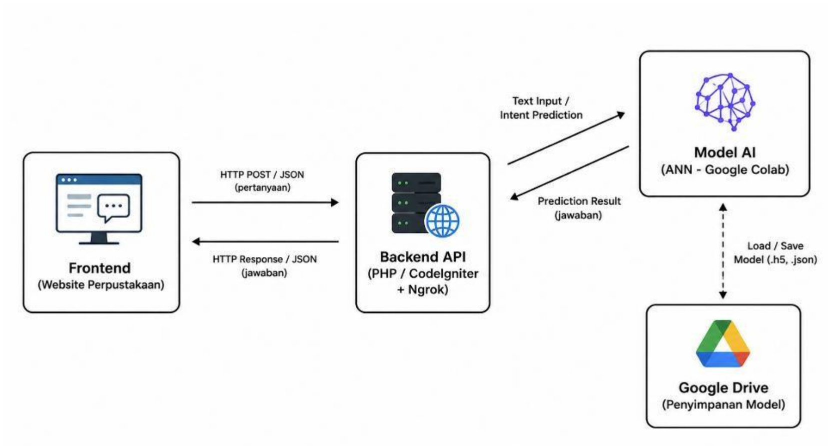


Fig. 1 *Workflow of the Chatbot System in Processing User Intent*

The technical framework of the chatbot application avoids direct-connection topologies in favor of a modern, separate multi-tier Decoupled Architecture. This strategy was explicitly chosen to segregate heavy mathematical tensor computation from the transaction-heavy relational databases of the library, thereby preventing server latency spikes and protecting student records. The system is split across three distinct computing environments (fiq.1):

1. **The Client Interface Tier**
   Programmed using modern HTML5, Bootstrap 5 styling elements, and Font Awesome 6 icons. The chat interface operates as an asynchronous, non-blocking floating widget embedded directly onto the primary landing page of the library web application. Sessional user queries are captured and dispatched via the native JavaScript fetch() API without triggering full webpage updates.

2. **The Cloud AI Inference Tier**
   Configured inside a secure Google Colab compute environment running an ASGI server wrapped with a Python Flask micro-framework. This layer receives incoming JSON payloads, converts raw string inputs into sparse feature vectors, passes them through the serialized neural network model matrix, and yields the final prediction outputs.
3. **The Secure Network Tunnel Layer**
   Because the cloud AI model and the local web framework live on physically separate networks, a secure, end-to-end encrypted reverse proxy tunnel was built using an Ngrok Agent execution layer. This transforms the local Python runtime port into a publicly accessible HTTPS endpoint, enabling reliable cross-origin AJAX data transfer.

2.2 Data Collection and Preprocessing

The primary knowledge domain dataset was constructed from official internal operational guidelines, circulation history logs, and transcripts of frequent student inquiries provided by the administrative division. These records were categorized into 56 mutually exclusive intent tags (such as jam_operasional, cek_plagiarism, skripsi_repository, and pinjam_buku). The finalized corpus contains 2,319 raw training rows.

Because human conversation style in text interactions is highly irregular, filled with informal Indonesian variations, morphological transformations, and accidental typographical errors, a precise text preprocessing pipeline was implemented to minimize noise [9]. The NLP pipeline sequentially executes five operations:

Case Folding

Converts all alphabetical string characters to lower case to discard typographic variations, mathematically defined as:

$$f(s) = s_{\text{lower}}$$

Tokenization

Splits long grammatical text strings into individual components or word pieces:

$$S = \{t_1, t_2, t_3, \ldots, t_n\}$$

Stopword Removal

Eliminates highly frequent words lacking information density (such as *yang*, *dan*, *di*, *adalah)*:

$$T' = \{t \in T \mid t \notin SW\}$$

Stemming via Sastrawi

Strips complex Indonesian morphological affixes (prefixes, infixes, suffixes, and confixes) using the native Nazief & Adriani rule set to reduce words to their structural root:

$$\text{stem}(w) = s \quad (\text{e.g., stem}(\text{"}\textit{peminjaman}\text{"}) = \text{"}\textit{pinjam}\text{"})$$

Vectorization via Bag of Words (BoW)

Translates token sets into a fixed-length numerical array based on a complete dictionary of 1,494 distinct vocabulary features. Every user entry is expressed as a sparse binary vector:

$$d = [x_1, x_2, \ldots, x_{|V|}], \quad x_i = \begin{cases} 1, & \text{if } V_i \in d \\ 0, & \text{otherwise} \end{cases}$$

2.3 ANN Classification Architecture and Hyperparameters

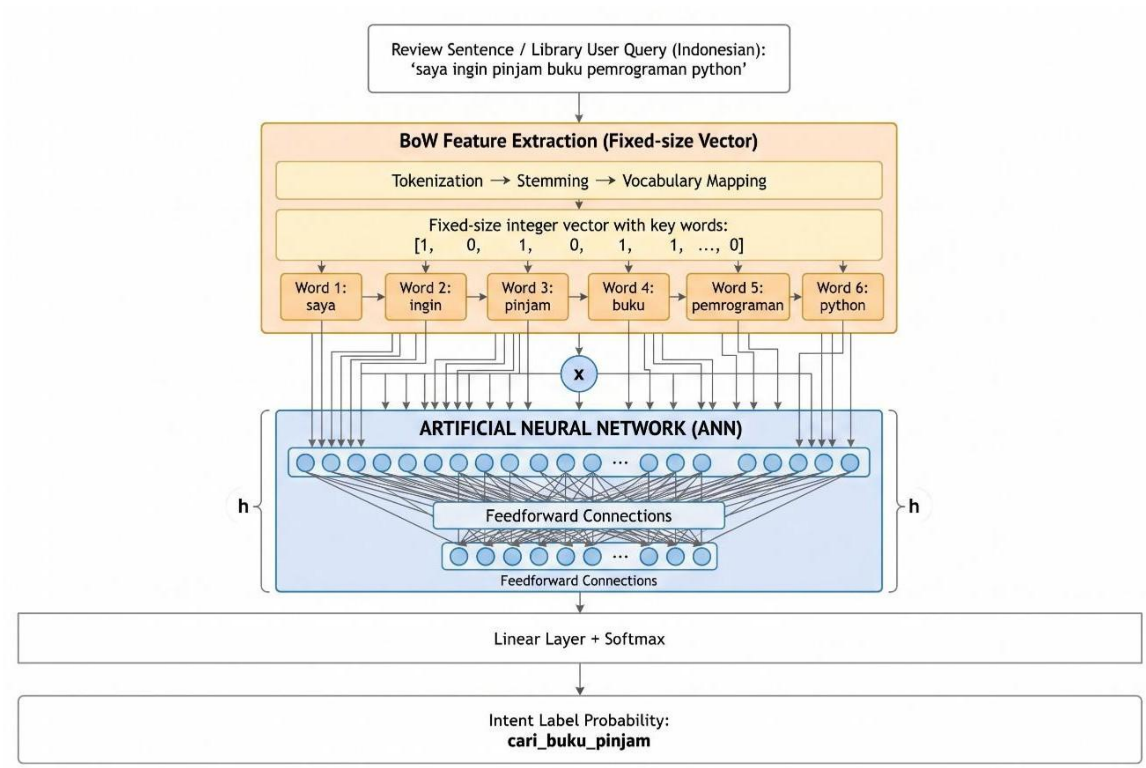


Fig. 2 *ANN-Based Chatbot Architecture Demonstrating Text Preprocessing, Intent Prediction, and Response Generation for Library Services*

The classification engine consists of an optimized [Fiq.2], Multi-Layer Perceptron (MLP) Sequential Neural Network constructed via TensorFlow and Keras. To isolate an architecture that avoids overfitting while maintaining high generalization metrics, automated tuning was executed via Keras Tuner Random Search over multiple cycles. The finalized network contains four layers [table.1]:

Input & Hidden Layer 1

A fully-connected Dense layer utilizing 256 hidden neurons with a Rectified Linear Unit (ReLU) activation function, mapped against the 1,494 text features.

Regularization Layer 1

A Batch Normalization layer paired with a 50% Dropout layer to eliminate co-adaptation and structural overfitting during training weights update, satisfying standard regularization methodologies[10].

Hidden Layer 2

A Dense layer using 128 hidden units with a ReLU activation function to optimize non-linear property representation.

Output Layer

A multi-class Dense layer containing 56 nodes matching the total intent categories, controlled via a Softmax activation function to generate definitive probability distributions:

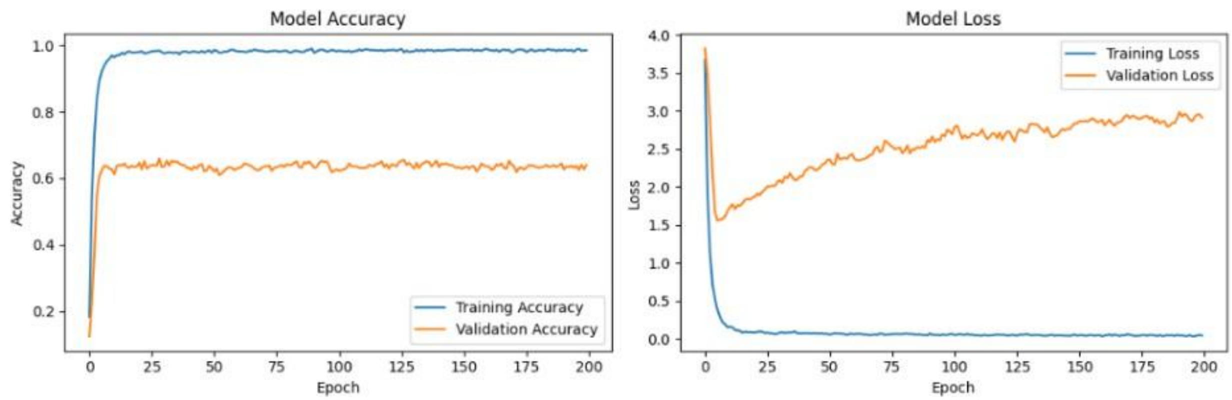
$$\mathrm{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{56} e^{z_j}}$$

Table 1. *Model Training Hyperparameter Configuration*

Hyperparameter Metric	Selected Operational Value
Input Dimension Vector	1,494 neurons (Vocabulary Size)
Hidden Layer 1 / 2 Units	256 neurons / 128 neurons
Regularization Parameters	Batch Normalization + 0.5 Dropout
Total Target Output Classes	56 Target Intent Labels
Batch Size / Epoch Maximum	32 samples / 200 Cycles
Optimization Algorithm	Adam Optimizer (0.001)

3. Result and Discussion

The model training progression demonstrated distinct optimization dynamics across the 200 execution loops. Initial iterations began with a high cross-entropy loss, but the implementation of Batch Normalization paired with a 0.5 Dropout layer and the Adam optimizer drove rapid initial error reduction, aligning with standard convergence optimization methodologies in localized deep learning tasks [8].



Training and validation accuracy and loss of the Artificial Neural Network (ANN) model for Xlibrary chatbot across 200 epochs.

As visualized in the training logs (Fig. 3), while the training parameters converged cleanly, the validation loss exhibited a moderate upward divergence after epoch 25, signaling mild overfitting due to the high structural granularity of the 56 intent classes. Despite this variance gap, the regularized network successfully locked the final model configurations into an operational state that achieved the targeted 91.00% evaluation accuracy before being exported to a serialized binary .h5 model format.

3.1 Evaluation Metrics and Intent Identification

To measure real-world performance, the dataset was split into an independent testing slice using an 80:20 stratified layout, dedicating 464 unique user statements entirely to the validation step. The global classification performance yielded an accuracy score of 91.00%, confirming that the multi-layer neural network model handles semantic variations reliably. This result achieves a competitive accuracy baseline compared to similar text-based FAQ automation structures built on shallow neural architectures [11][12].

An investigation into individual categories via the Classification Report reveals excellent accuracy across standard operational contexts. Critical digital infrastructure intents like *aplikasi_mobile*, *cek_plagiarism*, and *skripsi_repository* scored perfect precision and recall marks of 1.00, confirming the robustness of Artificial Neural Network algorithms in parsing highly targeted technical inquiries [13][14]. Highly repetitive visitor tags like *jam_buka* achieved near-flawless ratings with a precision of 0.95 and recall of 1.00, resulting in an F1-score of 0.98. Minor accuracy drops were restricted to vague intents with slight overlapping vocabularies, such as *agenda_kegiatan* (F1-score of 0.40) or *bantuan* (F1-score of 0.57). These edge cases occurred because shortened user messages often share identical root nouns across these classes, a common boundary-separation challenge widely documented in multi-class tokenization tasks [15]. However, because major operations maintain balanced boundary separation, these anomalies do not disrupt the overall system performance [table.2]:

Table 2. *ANN Model Architecture Layer Specifications After Hyperparameter Tuning*

No	Layer Type	Output Shape	Activation	Parameters
0	Input Layer	(None, 1494)	-	0
1	Dense	(None, 256)	ReLU	382,720
2	BatchNormalization	(None, 256)	-	1,024
3	Dropout	(None, 256)	-	0
4	Dense	(None, 128)	ReLU	32,896
5	BatchNormalization	(None, 128)	-	512
6	Dropout	(None, 128)	-	0
7	Dense (Output)	(None, 56)	Softmax	7,224
	Total Parameters			424,376

3.2 Functional Validation and Black-Box Testing

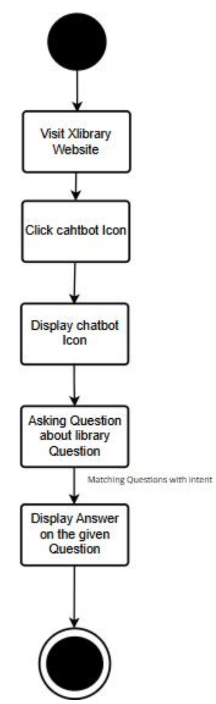


Fig. 4 *Client-Server System Architecture integrating Frontend, API, and NLP*

In addition to statistical validation [FIq.4], the software platform underwent rigorous functional verification using Black-Box testing methodologies to guarantee error-free execution across all front-end and back-end integration touchpoints. Testing verified that user statements sent via either physical click events or native "Enter" key strokes populated the chat viewport correctly while clearing the chat box instantly, satisfying the operational standards of interactive customer support frameworks [4]. Entering complex Indonesian variations like *"Saya ingin pinjam buku pemrograman python?"* successfully triggered the Sastrawi stemming mechanism, stripping affixes to reveal the fundamental vector root word *"cari_buku_pinjam"* for exact BoW mapping. This morphologic transformation validates the necessity of applying rigid Indonesian-specific NLP pipelines to counter the fragile fault tolerance of natural human language [6][16].

Sending automated user input streams rapidly in sequence evaluated proxy line limits, ensuring high availability under stress. The Ngrok secure bridge architecture managed concurrent JSON transfers efficiently, maintaining data integrity without losing packets or returning HTTP 504 Gateway errors. Entering repeated greeting statements like *"Hai"* returned randomized responses from the knowledge pool rather than identical answers, preventing conversational monotony and enhancing the natural pacing of Virtual Assistants [4].

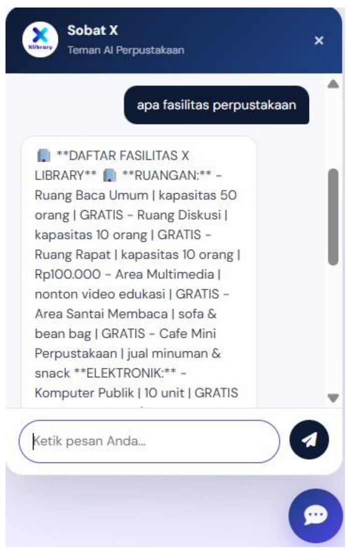


Fig. 5 Chat Interface Mockup: Clean conversational bubble layout embedding suggestion pills for user

The interactive scenarios within the Sobat X Chat Box UI [Fig.5] demonstrate the real-world operational capabilities and edge-case boundaries of the Artificial Neural Network (ANN) classification model when parsing digital library user inquiries. In the primary scenario, upon receiving the specific query *"apa fasilitas perpustakaan,"* the multi-layer Perceptron architecture accurately maps the sentence to the correct facility intent tag. This successful boundary separation allows the system to pull a structured, highly informative response directly from the intents.json knowledge base, immediately rendering critical logistical data regarding the capacity and pricing metrics of the general reading areas, discussion cubicles, meeting spaces, multimedia rooms, and public desktop terminals. Conversely, the secondary scenario illustrates a systemic classification anomaly triggered by the input string *"apa sarana yang ada"*. Because the informal Indonesian synonym *"sarana"* was omitted from the initial 1,494-dimensional Bag of Words (BoW) vocabulary feature extraction, the resulting input token vector fields return a zero scalar magnitude, yielding insufficient synaptic weights to trigger activation hidden layers toward the target class. Consequently, since the Softmax output distribution fails to satisfy the minimum prediction confidence threshold, the chatbot dynamically invokes its robust exception fallback mechanism, routing the interaction to an *out_of_context* safety class that prompts the student with standardized procedural examples—such as book lookup or operating hours queries—to preserve the overall informational validity of the platform, a technique recommended to optimize real-time digital collection catalog interactions[17][18].

4. Conclusions

This study designed and evaluated an ANN-based chatbot system for automating visitor inquiry handling in a university digital library. By combining a multi-layer Perceptron architecture with an Indonesian-specific NLP preprocessing pipeline — including Sastrawi-based morphological stemming and Bag-of-Words feature extraction — the system classified user intent across 56 categories with a global accuracy of 91.00% [table 3]. The decoupled deployment architecture, separating the deep learning backend from the web frontend via a secure HTTPS tunnel, proved effective for maintaining low response latency and stable cross-platform communication. Black-box functional testing further confirmed that the system handles diverse real-world Indonesian language inputs reliably, including morphological variations and informal phrasing. These findings suggest that ANN-based intent classification is a viable and cost-effective approach for automating front-tier library services, particularly in resource-constrained academic institutions that cannot afford large-scale transformer-based solutions.

Table 3. *Performance Benchmarking for ANN Model*

Model	**Precision**	**Recall**	**F1-score**
accuracy	0.91	0.91	0.91
Macro avg	0.91	0.93	0.91
Weighted avg	0.94	0.91	0.92

Despite these results, several limitations should be acknowledged. The model exhibited mild overfitting after epoch 25, and the static Bag-of-Words representation cannot capture contextual word relationships, making the system vulnerable to out-of-vocabulary inputs. The current system also supports only single-turn responses and does not maintain conversational context across multiple exchanges. For future development, integrating a Retrieval-Augmented Generation (RAG) framework with a Large Language Model would enable dynamic, context-aware responses beyond fixed intent mappings. An administrative panel for non-technical staff to manage intent categories, along with direct integration with the library's relational database for transactional operations such as live book availability and loan extensions, would further strengthen the system's practical utility.

5. References

[1] Araf Aliwijaya, & Hanny Chairany Suyono. (2023). Peluang Pemanfaatan Big Data di Perpustakaan. *Info Bibliotheca: Jurnal Perpustakaan Dan Ilmu Informasi*, *4*(2), 1–17. https://doi.org/10.24036/ib.v4i2.397

[2] Jolie, A., Dedrick, D., Sugeng, R. K., Lee, W. A., & Yulianto, A. (2022). Aplikasi Sistem Manajemen Perpustakaan dengan Penerapan Pemrograman Berorientasi Objek. Telcomatics, 7(2), 61–69. https://doi.org/10.37253/telcomatics.v7i2.7349

[3] Ahwan, M. A., Abiadhoh, N., Kusuma, A. B., & Alam, U. F. (2024). Pembangunan WhatsApp Chatbot sebagai layanan kecerdasan buatan di Perpustakaan UIN Walisongo Semarang. Berkala Ilmu Perpustakaan Dan Informasi, 20(1), 119–132. https://doi.org/10.22146/bip.v20i1.7043

[4] Amrullah, A. Z., Anas, A. S., & Primajati, G. (2022). Implementasi Chatbot sebagai Virtual Assistant Penerimaan Mahasiswa Baru pada Universitas Bumigora. Jurnal Bumigora Information Technology (BITe), 4(1), 17–26. https://doi.org/10.30812/bite.v4i1.1664

[5] Budi Susanto, E., Reza Maulana, M., Agus Setiawan, T., & Ilyas, A. (2024). Evaluasi Komprehensif Algoritma Naive Bayes Dan Knn Dalam Pengembangan Chatbot Cerdas Untuk Layanan Informasi PMB. Jurnal Teknologi Informasi, 8(2).

[6] Christian, Y., & Erline, M. (2022). Web-Based Chatbot With Natural Language Processing and Knuth-Morris-Pratt (Case Study: Universitas Internasional Batam). JST (Jurnal Sains Dan Teknologi), 11(1), 132–141. https://doi.org/10.23887/jstundiksha.v11i1.43258

[7] Fadli, M. F., Buntoro, G. A., & Masykur, F. (2023). Penerapan Algoritma Neural Network Pada Chatbot Pmb Universitas Muhammadiyah Ponorogo Berbasis Web. JuSiTik : Jurnal Sistem Dan Teknologi Informasi Komunikasi, 6(1), 13–22. https://doi.org/10.32524/jusitik.v6i1.786

[8] Hafizh, R. H. (2023). Pengembangan Chatbot Berbasis Jaringan Saraf Transformer untuk Layanan Informasi Akademik dan Keuangan Mahasiswa di Universitas Muhammadiyah Sukabumi. Jurnal Informatika Dan Teknik Elektro Terapan, 12(3), 3412–3419.

[9] Evanthe, H., Afrrahman S. Effendi, A., Putra Paskah Halawa, S., Zidane, M., & Perdana, A. (2025). Rancang Bangun Chatbot Sebagai Media Layanan Informasi Akademik Mahasiswa Universitas Negeri Medan Berbasis Nlp Sederhana. JATI (Jurnal Mahasiswa Teknik Informatika), 9(4), 6049–6054. https://doi.org/10.36040/jati.v9i4.13997

[10] Bagus Arya Atmaja, Gery Samuel Gultom, Jhon Hansen Manurung, & Victor Asido Elyakim P. (2024). Penerapan Metode Backpropagation dalam Jaringan Saraf Tiruan untuk Meningkatkan Prediksi Produksi Kentang di Sumatera. INSOLOGI: Jurnal Sains Dan Teknologi, 3(6), 674–683. https://doi.org/10.55123/insologi.v3i6.4658

[11] Mustakim, F., Fauziah, F., & Hayati, N. (2021). Algoritma Artificial Neural Network pada Text-based Chatbot Frequently Asked Question (FAQ) Web Kuliah Universitas Nasional. Jurnal JTIK (Jurnal Teknologi Informasi Dan Komunikasi), 5(4), 438–446. https://doi.org/10.35870/jtik.v5i4.261

[12] Purwitasari, N. A., & Soleh, M. (2022). Implementasi Algoritma Artificial Neural Network Dalam Pembuatan Chatbot Menggunakan Pendekatan Natural Language Parocessing. Jurnal IPTEK, 6(1). https://doi.org/10.31543/jii.v6i1.192

[13] Fitrianingsih, F., Susetianingtias, D. T., Pernadi, D., Patriya, E., & Arianty, R. (2022). The Implementation of Artificial Neural Network (ANN) on Offline Cursive Handwriting Image Recognition. ILKOM Jurnal Ilmiah, 14(1), 63–73. https://doi.org/10.33096/ilkom.v14i1.1113.63-73

[14] Reinfa, F. F., & Nerisafitra, P. (2025). Analisis Chatbot Pendaftaran Mahasiswa Baru Menggunakan Algoritma Artificial Neural Network (Studi Kasus: Universitas Negeri Surabaya). Journal of Informatics and Computer Science (JINACS), 6(03), 842–851. https://doi.org/10.26740/jinacs.v6n03.p842-851

[15] Rifqi, M. R. A. F., Armansyah, & Rifqi Al Fauzan, M. (2024). Kombinasi TF-IDF dan Neural Network Untuk Pelayanan Informasi Al-Qur'an Dalam Bentuk Chatbot. Jurnal Fasilkom, 14(2), 318–324. https://doi.org/10.37859/jf.v14i2.7286

[16] Ja'far Ali, A., & Nurhidayat, A. I. (2024). Pengembangan Chatbot Pendaftaran E-KTP Berbasis Web menggunakan Metode NLP (Natural Language Processing). Jmi, 1.

[17] Lena, M., Florensia, N. P., Patimah, Y., Pranatawijaya, V. H., & Sari, N. N. K. (2024). Penerapan Teknologi Ai Dari Gemini Untuk Meningkatkan Layanan Peminjaman Buku Online Pada Aplikasi Cozybook. Jurnal Informatika Dan Teknik Elektro Terapan, 12(3), 1705–1712. https://doi.org/10.23960/jitet.v12i3.4396

[18] Qaulan, M. A., Wahyuni, & Adytia, P. (2025). Pengembangan Chatbot Berbasis AI untuk Mendukung Pelayanan Perpustakaan. Jurnal Teknologi Informasi Komunikasi(TEMATIK), 12(1), 23–30.