

Integrasi Sistem Question Answering (QA) sebagai Model Berbasis Layanan Web menggunakan Django Rest Framework

Nurjayanti^{1*}, Utami Kusuma Dewi²

^{1,2} Program Studi Rekayasa Perangkat Lunak, Fakultas Informatika, Universitas Telkom

*Email: jayanti@telkomuniversity.ac.id

Abstrak

Penelitian ini bertujuan untuk mengembangkan sistem *Question Answering (QA)* Islami berbasis layanan web dengan memanfaatkan *RESTful Web Services*. Sistem *QA* ini dirancang untuk memberikan jawaban akurat dan relevan terhadap pertanyaan dalam domain agama Islam, menggunakan sumber daya teks seperti *Al-Quran*, *hadis*, dan literatur keagamaan lainnya sebagai basis pengetahuan. Tahap perancangan melibatkan pembuatan rancangan basis data dan endpoint *REST API*, sementara tahap implementasi menggunakan *Django* dan *Django REST Framework (DRF)* untuk mengintegrasikan model *QA* dengan *REST API*. Pengujian sistem dilakukan menggunakan aplikasi *Postman* untuk memastikan bahwa semua endpoint berfungsi sesuai dengan yang diharapkan. Hasil penelitian menunjukkan bahwa sistem *QA* Islami ini tidak hanya fleksibel dan mudah diimplementasikan, tetapi juga mampu diintegrasikan dengan berbagai platform lainnya.

Kata kunci: *Sistem Question Answering (QA)*, *RESTful Web Services*, *Django*, *Django REST Framework (DRF)*, *Islam*, *Layanan Web*

Abstract

This study aims to develop an Islamic *Question Answering (QA)* system based on *RESTful Web Services*. The *QA* system is designed to provide accurate and relevant answers to questions within the domain of Islamic studies using textual resources such as the *Qur'an*, *Hadith*, and other religious literature as its knowledge base. The design phase involves the development of a database schema and *REST API* endpoints, while the implementation phase employs *Django* and the *Django REST Framework (DRF)* to integrate the *QA* model with the *REST API*. System testing was conducted using the *Postman* application to ensure that all endpoints functioned as expected. The results of the study indicate that the Islamic *QA* system is not only flexible and easy to implement but also capable of integration with various other platforms.

Keywords: *Question Answering (QA)*, *RESTful Web Services*, *Django*, *Django REST Framework (DRF)*, *Islam*, *web services*

PENDAHULUAN

Sistem *Question Answering (QA)* merupakan salah satu alat efektif untuk memperoleh jawaban akurat terhadap pertanyaan dalam bahasa alami dari pengguna sistem. Penelitian pada sistem *QA* banyak dilakukan dengan mengkombinasikan berbagai teknik seperti pemrosesan bahasa alami, pengambilan informasi, ekstraksi informasi, pemrosesan grafik pengetahuan, pembelajaran mesin, dan pembelajaran mendalam (Momtazi & Abbasiantaeb, 2022). Pada umumnya, studi mengenai *QA* dibagi berdasarkan domain pertanyaan, seperti keagamaan (Hamed & Aziz, 2016), pendidikan (Lende & Raghuwanshi,

2016), kesehatan (Bae et al., 2021), hukum (Thomas & Sangeetha, 2022), dan domain lainnya.

Salah satu domain keagamaan yaitu Islam memiliki banyak sumber daya teks yang tersedia secara daring, di antaranya kitab suci *Al-Quran*, *hadis*, *tafsir*, dan literatur keagamaan lainnya yang digunakan sebagai basis pengetahuan dalam sistem *QA*. Ketersediaan data ini memungkinkan para peneliti untuk mengembangkan dan melatih sistem *QA* yang spesifik untuk menjawab pertanyaan mengenai Islam (Hamed & Aziz, 2016) (Nurjayanti et al., 2022) (Ahmed et al., 2022).

Penelitian sebelumnya telah dilakukan pada pengambilan pertanyaan dalam sistem QA dengan menggunakan metode Siamese LSTM pada tahun 2022. Studi tersebut secara khusus berfokus pada pertanyaan dalam domain agama Islam. Penyematan kata yang sudah dilatih sebelumnya digunakan untuk meningkatkan pemahaman kontekstual dan diintegrasikan ke dalam model Siamese LSTM. Lapisan akhir model menggunakan fungsi Manhattan untuk memprediksi kesamaan antara pertanyaan dari pengguna dan pertanyaan dalam basis data (Nurjayanti et al., 2022). Meskipun peneliti sudah melakukan evaluasi pada model yang dilatih, proses implementasi (*deployment*) dari sistem QA tersebut belum dilakukan. Oleh karena itu, diperlukan penelitian lebih lanjut mengenai bagaimana proses implementasi sistem QA Islam tersebut dari mulai perancangan sampai tahap pengujian.

Penelitian ini bertujuan untuk membangun sistem QA Islami berupa layanan web (*web service*) yaitu dengan melakukan pembungkusan model sehingga dapat diintegrasikan dengan web, aplikasi bergerak, atau perangkat lunak lainnya. Salah satu jenis layanan pembungkus yang digunakan dalam penelitian ini adalah *RESTful Web Service*. REST (*Representational State Transfer*) adalah gaya arsitektur dari *Application Programming Interface* (API) yang saat ini banyak digunakan dalam pengembangan web modern. Dalam REST dibuat serangkaian batasan untuk meminimalkan latensi dan komunikasi jaringan, serta memaksimalkan independensi dan skalabilitas dari implementasi komponennya (Fielding & Taylor, 2002). Dengan REST API, model dapat mudah berkomunikasi dan berinteraksi dengan aplikasi web maupun aplikasi bergerak (C Hillar, 2018).

Dalam metode penelitian ini, akan dilakukan analisa terhadap desain basis data dan arsitektur, implementasi, pengujian dan evaluasi dari REST API yang dibuat. Tujuannya adalah menyediakan pendekatan terstruktur bagi pengembang dan peneliti lainnya untuk membangun sistem QA sebagai model berbasis layanan web.

TINJAUAN PUSTAKA

Penelitian terkait implementasi sistem QA telah dilakukan dengan memanfaatkan Flutter (Permana et al., 2024). Penelitian tersebut menghasilkan sebuah aplikasi *mobile* berbasis

chatbot yang mengakses data eksternal melalui layanan web dari aplikasi *backend* E-Mufassir. Penelitian serupa mengembangkan *chatbot* untuk sistem QA yang terintegrasi dengan layanan kecerdasan buatan (AI) untuk mendukung proses ekstraksi dan pencocokan kata kunci dari korpus (Alhassan et al., 2022).

Berbeda dengan pendekatan yang digunakan dalam kedua penelitian tersebut dimana menitikberatkan pada penggunaan *chatbot* sebagai antarmuka utama sistem QA, penelitian ini mengadopsi pendekatan berbasis QA model terlatih yang secara langsung memproses pertanyaan dan menemukan jawaban dari basis data. Oleh karena itu, fokus penelitian ini diarahkan pada perancangan dan implementasi model berbasis layanan web.

Selaras dengan penelitian yang dilakukan oleh Alhassan et al. (2022), penelitian ini juga menggunakan bahasa pemrograman Python yang memiliki kemampuan unggul dalam pemrosesan teks serta menyediakan pustaka-pustaka yang mendukung kebutuhan teknis implementasi model QA.

Application Programming Interface (API) adalah suatu cara bagaimana komputer berkomunikasi secara langsung dengan komputer lainnya. Salah satu arsitektur dari API web yang banyak digunakan adalah *Representational State Transfer* (REST). REST menyediakan akses terprogram pada fungsionalitas berupa data yang diserialkan (JSON atau XML) bagi pengguna dan aplikasi (Baker, 2022). Dalam penelitian sebelumnya, sistem QA dikembangkan dengan menggunakan bahasa pemrograman Python. Oleh karena itu, pada pengembangan REST API dalam penelitian ini akan digunakan *Django* dan *Django REST Framework* (DRF) yang juga mendukung bahasa pemrograman Python.

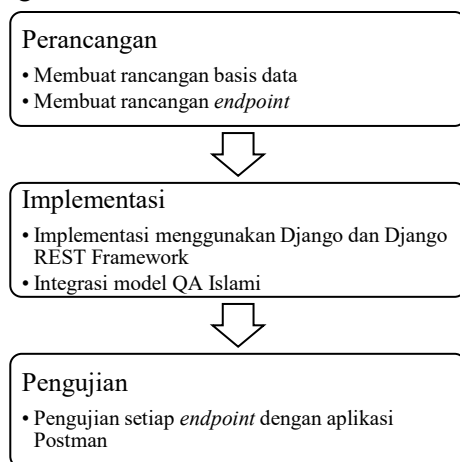
DRF merupakan seperangkat alat untuk mengembangkan web API dalam Django, yang merupakan web framework tingkat tinggi dari Python. DRF menyediakan banyak fitur untuk layanan web seperti model basis data, serialisasi data, autentikasi untuk keamanan dan perizinan pada web API (C Hillar, 2018).

Integrasi sistem QA Islami dengan DRF memungkinkan pembuatan layanan web yang skalabel, efisien dan ramah pengguna. Dengan memanfaatkan kemampuan dari model QA modern dengan arsitektur DRF, diharapkan dapat membuat sistem QA Islami sebagai model berbasis layanan web yang handal serta mudah

diimplementasi. Integrasi ini juga menjanjikan peningkatan akses dan penggunaan dari sistem QA dalam berbagai aplikasi atau domain lainnya.

METODE

Dalam proses pengembangan sistem QA Islami dengan menerapkan REST API dibagi menjadi tiga tahap utama yaitu perancangan, implementasi dan pengujian. REST API dikembangkan menggunakan Django dan Django REST Framework.



Gambar 1. Tahapan Metodologi

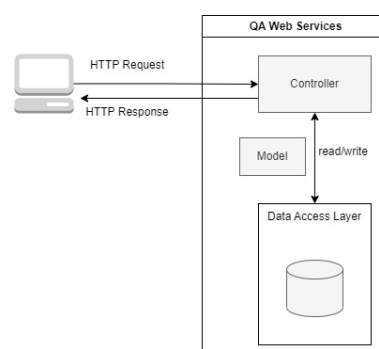
Pada tahap perancangan dibuat rancangan basis data untuk kebutuhan implementasi sistem QA Islami. Pertanyaan dari pengguna sistem akan disimpan dan kemudian diproses untuk memperoleh jawaban dari model QA. Hasil keluaran dan evaluasi dari model juga disimpan untuk meningkatkan basis pengetahuan dari model QA. Berikutnya dibuat rancangan setiap *endpoint* untuk memudahkan pembuatan REST API dengan menentukan nama, metode dan parameter dari *endpoint*.

Kemudian ditahap implementasi, rancangan yang sudah dibuat menjadi acuan untuk diimplementasikan ke dalam kerangka kerja web dari bahasa pemrograman Python yaitu Django dan DRF. Model yang dihasilkan dari penelitian Nurjayanti et al. (2022) diimpor ke dalam proyek untuk diintegrasikan dengan REST API yang dibuat. Selain itu, kode pemrograman untuk pembuatan masukan model, pra pemrosesan data, dan konfigurasi model juga diimpor ke dalam proyek yang dibuat.

Pada tahap akhir pengujian dilakukan terhadap setiap *endpoint* guna memastikan setiap

fungsi berjalan dengan baik dan sesuai rancangan. Pengujian dilakukan dengan menggunakan aplikasi Postman yang banyak digunakan untuk pengujian web API. Pengujian dilakukan dengan memperhatikan kode status dan balasan serialisasi data yang diberikan dari *endpoint* yang diuji.

Pada Gambar 2 dapat dilihat rancangan arsitektur dari layanan web sistem QA Islami, pengguna mengirimkan permintaan ke REST API melalui *controller* dan menerima balasan berupa serialisasi data.

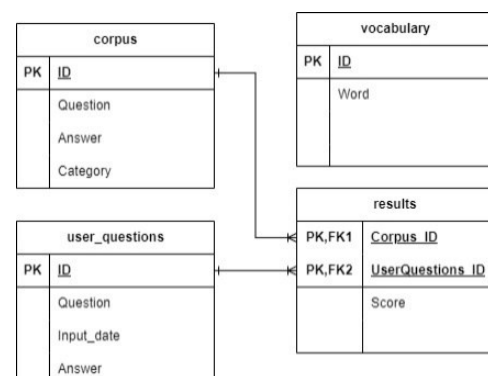


Gambar 2. Arsitektur Layanan Web Sistem QA Islami

HASIL DAN PEMBAHASAN

Proses pengembangan sistem QA Islami berbasis layanan web ini dimulai dengan analisa dan perancangan basis data. Basis data yang dirancang adalah untuk menampung parameter masukan dari pengguna dan luaran dari sistem. Selain itu terdapat basis pengetahuan dari model QA dalam rancangan. Pada

Gambar 3 dapat dilihat rancangan basis data untuk sistem QA Islami berbasis layanan web.



Gambar 3. Rancangan Basis Data Layanan Web Sistem QA

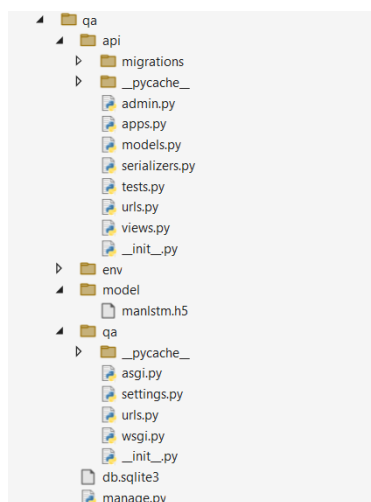
Berikutnya dibuat rancangan setiap *endpoint* seperti pada

Tabel 1 yang berfungsi untuk memudahkan proses pembuatan REST API dan memastikan semua kebutuhan pada sistem QA Islami terpenuhi.

Tabel 1. Daftar *Endpoint* REST API

Method HTTP	Endpoint	Keterangan
GET	/api/v1/question	Get all previous questions
PUT	/api/v1/question/[q]	Insert a new question
DELETE	/api/v1/question/[q]	Delete a question
GET	/api/v1/corpus	Get all corpus
GET	/api/v1/vocab	Get all vocabulary
GET	/api/v1/result/[q]	Get the model result for the question
DELETE	/api/v1/result/[q]	Delete the model result to the question
POST	/api/v1/predict	Get model prediction

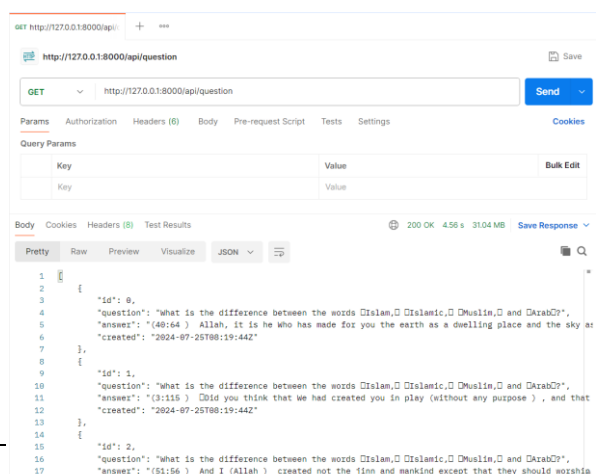
Pada tahap implementasi, rancangan dari tahap sebelumnya diimplementasikan ke dalam kerangka kerja web. Model dari sistem QA Islami disimpan dengan format **.h5** untuk diintegrasikan dengan REST API yang dibuat. Adapun hasil implementasi dari integrasi sistem QA Islami sebagai layanan web dapat dilihat pada Gambar 4.



Gambar 4. Implementasi Layanan Web Sistem QA Islami

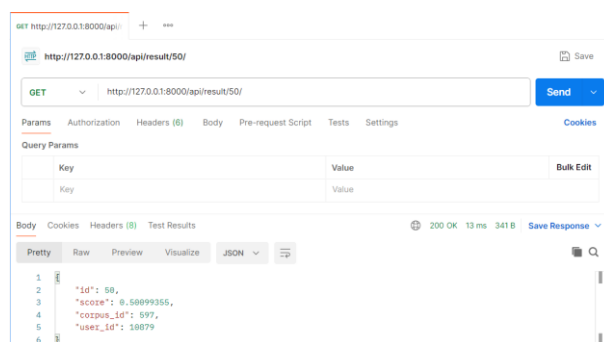
Selanjutnya dilakukan pengujian untuk evaluasi dan memastikan kelayakan serta kesesuaian REST API yang dibuat. Dapat dilihat pada

Gambar 5 merupakan layanan yang digunakan untuk menampilkan semua pertanyaan pengguna pada sistem QA Islami berhasil mengirim balasan berupa sumber daya (*resource*) dalam format JSON dengan kode status 200 yang berarti sukses.

Gambar 5. Pengujian *Endpoint* Question

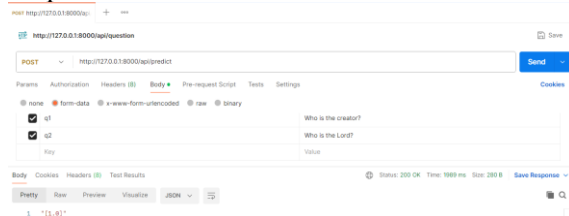
Pada

Gambar 6 merupakan layanan yang digunakan untuk menampilkan jawaban dari pertanyaan pengguna dalam sistem QA berhasil mengirim balasan dengan kode status 200 yang berarti berhasil atau sukses.

Gambar 6. Pengujian *Endpoint* Result

Implementasi integrasi model yang sudah dilatih dari sistem QA dapat dilihat pada

Gambar 7. Balasan dari *endpoint* adalah keluaran fungsi prediksi dari model dengan parameter masukan berupa teks pertanyaan. Metode impor model kedalam proyek memungkinkan sistem QA menggunakan langsung model yang sudah dilatih dan dioptimasi.



Gambar 7. Pengujian *Endpoint Predict*

SIMPULAN

Penerapan REST API menjadikan sistem QA Islami ini lebih fleksibel, mudah diimplementasi, dan dapat diintegrasikan dengan berbagai *platform* lainnya. Hal ini dapat memperluas cakupan penggunaan dari sistem QA Islami, memungkinkan akses yang lebih luas bagi masyarakat yang memerlukan pengetahuan agama Islam. Penelitian ini tidak hanya memberikan kontribusi penting dalam pengembangan sistem QA sebagai model berbasis layanan web, tetapi juga membuka peluang signifikan untuk penelitian lebih lanjut dalam berbagai domain lainnya.

Adapun penelitian selanjutnya akan difokuskan pada pengembangan antarmuka pengguna (*user interface*) yang terhubung langsung dengan layanan web pada sistem QA sehingga sistem ini dapat digunakan secara lebih interaktif dan ramah pengguna. Kemudian, penerapan teknik pemrosesan bahasa alami (*Natural Language Processing*, NLP) lanjutan seperti pemahaman kontekstual dan pencarian semantik juga mampu meningkatkan akurasi serta relevansi jawaban atas pertanyaan-pertanyaan keagamaan yang kompleks. Selain itu, pada tahap implementasi melalui uji coba langsung di lingkungan pendidikan atau platform komunitas daring diharapkan dapat memberikan umpan balik yang berharga untuk pengembangan sistem QA secara berkelanjutan dan berbasis kebutuhan pengguna.

DAFTAR PUSTAKA

Ahmed, B. H., Saad, M. K., & Refaee, E. A. (2022). QQATeam at Qur'an QA 2022:

Fine-Tuning Arabic QA Models for Qur'an QA Task. *5th Workshop Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection, OSACT 2022 - Proceedings at Language Resources and Evaluation Conference, LREC 2022*.

Alhassan, N. A., Saad Albarrak, A., Bhatia, S., & Agarwal, P. (2022). A Novel Framework for Arabic Dialect Chatbot Using Machine Learning. *Computational Intelligence and Neuroscience*, 2022. <https://doi.org/10.1155/2022/1844051>

Bae, S., Kim, D., Kim, J., & Choi, E. (2021). Question Answering for Complex Electronic Health Records Database using Unified Encoder-Decoder Architecture. *Proceedings of Machine Learning Research*, 158.

Baker, M. (2022). Secure Web Application Development: A Hands-On Guide with Python and Django. In *Secure Web Application Development: a Hands-On Guide with Python and Django*. <https://doi.org/10.1007/978-1-4842-8596-1>

C Hillar, G. (2018). Django Restful Web Services The Easiest Way to Build Python Restful Api and Web Services With Django. In *Django RESTful Web Services*.

Fielding, R. T., & Taylor, R. N. (2002). Principled Design of the Modern Web Architecture. *ACM Transactions on Internet Technology*, 2(2). <https://doi.org/10.1145/514183.514185>

Hamed, S. K., & Aziz, M. J. A. (2016). A question answering system on Holy Quran translation based on question expansion technique and Neural Network classification. *Journal of Computer Science*, 12(3). <https://doi.org/10.3844/jcssp.2016.169.177>

Lende, S. P., & Raghuwanshi, M. M. (2016). Question answering system on education acts using NLP techniques. *IEEE WCTFTR 2016 - Proceedings of 2016 World Conference on Futuristic Trends in Research and Innovation for Social*

Welfare.

<https://doi.org/10.1109/STARTUP.2016.7583963>

- Momtazi, S., & Abbasiantaeb, Z. (2022). Question Answering over Text and Knowledge Base. In *Question Answering over Text and Knowledge Base*. <https://doi.org/10.1007/978-3-031-16552-8>
- Nurjayanti, Adiwijaya, & Al Faraby, S. (2022). LEARNING CONTEXTUAL MEANING FOR QUESTION RETRIEVAL USING SIAMESE LSTM ON ISLAMIC QUESTION ANSWERING SYSTEM. *ICIC Express Letters*, 16(9). <https://doi.org/10.24507/icicel.16.09.1011>
- Permana, L. A., Safaat Harahap, N., Handayani, L., & Affandes, M. (2024). Implementasi Fitur Question Answering pada Aplikasi Qur'an Berbasis Mobile Menggunakan Framework Flutter. *JUPITER: Jurnal Penelitian Ilmu Dan Teknologi Komputer*, 16(1), 301–312.
- Thomas, A., & Sangeetha, S. (2022). Knowledge Graph Based Question-Answering System for Effective Case Law Analysis. *Smart Innovation, Systems and Technologies*, 267. https://doi.org/10.1007/978-981-16-6616-2_27