

Pembuatan Sistem Frequently Asked Questions pada Website Ruparupa

Tamara Violeta¹, Desi Arisandi², Novario Jaya Perdana³

^{1,3}Sistem Informasi, Fakultas Teknologi Informasi, Universitas Tarumanagara

²Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara

Jl. Letjen S. Parman, Tomang, Daerah Khusus Ibukota Jakarta 11440

E-mail: ¹tamaravioleta14@gmail.com, ²desi@fti.untar.ac.id, ³novariojp@fti.untar.ac.id

ABSTRAK

Teknologi membawa pengaruh besar terhadap hidup masyarakat tanpa dapat diprediksi setiap generasinya. Pengaruh yang didatangkan dari teknologi mengarahkan masyarakat untuk memulai bisnis dengan membuat toko *online*, sebagai bagian dari *e-commerce*. PT Omni Digitama Internusa (Ruparupa) merupakan salah satu *e-commerce* yang menjual berbagai peralatan rumah tangga, furnitur, dan gaya hidup dibawah naungan Kawan Lama Group. Sebagai *e-commerce*, dibutuhkan wadah informasi dengan penyediaan FAQ (*Frequently Asked Questions*) untuk efisiensi waktu dan peningkatan *traffic website*, serta lebih membangun kepercayaan dan profesionalisme. FAQ yang efektif dapat dilihat dari penyajian informasi yang ringkas, detil, dan mudah diakses. FAQ Ruparupa saat ini masih menggunakan kata statis, dimana belum dapat menampilkan pertanyaan berulang dan akses yang terhitung masih cukup sulit. Muncul ide sebagai solusi untuk mengatasi permasalahan tersebut, yaitu pembuatan sistem FAQ terbaru dengan adanya pengkategorian FAQ dengan pendukung fitur-fitur tambahan. Tujuannya membantu konsumen dengan memusatkan perhatian mereka pada kasus yang lebih spesifik, sehingga meningkatkan efisiensi waktu, baik dari sisi konsumen, maupun tim Ruparupa. Sistem dibuat menggunakan metode SDLC (*System Development Life Cycle*) dengan model *Scrum*. Bahasa pemrograman yang digunakan yaitu *ReactJS* dengan *database* MongoDB. FAQ ini akan meningkatkan kualitas UI/UX Ruparupa dan memberikan kemudahan untuk mengelola FAQ dengan adanya *user-friendly dashboard*.

Kata Kunci: *FAQ, E-commerce, ReactJS, Content Management System*

ABSTRACT

Digital technology brings a impact on people's life which cannot be predicted everyeach generation. Technology directs people to start a business by creating online stores, as a part of *e-commerce*. PT Omni Digitama Internusa (Ruparupa) is an *e-commerce* that sells various household appliances, furniture and lifestyle under the auspices of the Kawan Lama Group. As an *e-commerce*, an information forum is needed by providing FAQs (*Frequently Asked Questions*) for time efficiency and increasing website traffic, as well as building more trust and professionalism. An effective FAQ can be seen from the concise information, detailed, and easy to access. Current FAQ on Ruparupa still uses the static word, which cannot display repeated questions and quite difficult access. An idea emerged as a solution to overcome the existing problems, by creating a new FAQ system with FAQ categorization and supporting features. The aim is to help consumers by focusing their attention on more specific cases, thus increasing time efficiency. This system was created using the SDLC (*System Development Life Cycle*) method with the *Scrum* model. The programming language used is *ReactJS* and *MongoDB* database. This FAQ will improve Ruparupa's UI/UX quality and make it easier to manage FAQs with an *easy-to-use dashboard*.

Keywords: *FAQ, E-commerce, ReactJS, Content Management System*

1. PENDAHULUAN

Perkembangan teknologi yang berkembang pesat sesuai perkembangan zaman membawa dampak besar bagi kehidupan masyarakat. Masyarakat kini banyak memanfaatkan teknologi untuk berbagai hal, bahkan untuk kegiatan usaha dan bisnis. Hal tersebut menjadi peluang bisnis bagi masyarakat dengan membuat toko *online* yang merupakan bagian dari *e-commerce*.

E-commerce merupakan suatu media *online* untuk melakukan aktivitas yang berkaitan dengan

penjualan, pembelian, dan pemasaran barang atau jasa (Sulistiyawati dan Nursiam, 2019: 161). Terdapat beberapa jenis *e-commerce*, antara lain yaitu *Business to Business*, *Business to Consumer*, *Consumer to Consumer*, *Consumer to Business*, *Business to Public Administration*, dan *Consumer to Public Administration* (Investopedia, 2022). Dengan adanya *e-commerce*, transaksi jual beli dapat dilakukan dengan sangat mudah, dimana pembeli dan penjual tidak diharuskan untuk bertemu secara langsung.

PT Omni Digitama Internusa (Ruparupa) merupakan salah satu *e-commerce* yang menyediakan peralatan rumah tangga, furnitur, dan gaya hidup dibawah naungan Kawan Lama Group. Sebagai *e-commerce*, dibutuhkan wadah informasi dan kenyamanan bagi pengunjung dengan penyediaan FAQ (*Frequently Asked Questions*).

FAQ merupakan bentuk implementasi *Knowledge Management System* untuk mengelola pengetahuan yang menjelaskan data ataupun informasi penting (Setyadi & Johan, 2022). FAQ berarti kumpulan data pertanyaan dan jawaban paling sering diajukan (Arthana, dkk., 2021). Kefektifannya dapat dilihat dari penyajian informasi yang ringkas dan detil, serta mudah diakses.

Sebagai keputusan untuk data pertanyaan apa saja yang ingin dimunculkan pada FAQ, dibutuhkan data nyata melalui penelusuran *website* ataupun interaksi langsung dengan pelanggan. Dibutuhkan juga pertimbangan atas informasi yang kemungkinan akan konstan untuk beberapa waktu, karena pada umumnya FAQ tidak perlu diperbarui terlalu sering (Rheny, 2022).

FAQ Ruparupa saat ini masih menggunakan kata statis dimana belum otomatis menampilkan pertanyaan berulang dan akses yang terhitung cukup sulit. Maka dari itu, dibuatlah sistem FAQ terbaru dengan adanya pengkategorian FAQ dan pendukung fitur tambahan. Tujuannya yaitu memusatkan perhatian pengunjung pada kasus yang lebih spesifik, sehingga meningkatkan efisiensi waktu.

Rancangan pembuatan sistem FAQ akan terbagi menjadi beberapa *section*, yaitu kategori, artikel FAQ, diskusi, dan Hubungi Kami. Kategori menyediakan sejumlah menu yang dibedakan berdasarkan tema. Artikel FAQ menyediakan sejumlah pertanyaan paling sering diajukan secara berurutan. Diskusi berupa fitur tanya jawab antara pengunjung *website*. Terakhir, Hubungi Kami sebagai layanan komunikasi.

Adapun beberapa fitur pendukung lainnya seperti FAQ *history*, *searching* FAQ, dan *feedback customer* untuk informasi *rating* yang detil. Selain itu, Adapun pembuatan *Content Management System* untuk mengelola data supaya dapat membuat dan memodifikasi data yang akan dimunculkan pada FAQ *website* (Cabot, J (2018)). Sistem dibuat dengan bahasa pemrograman React JS dengan DBMS MongoDB.

Dalam penelitian *Sistem Frequently Asked Questions* menggunakan metode pengembangan SDLC (*System Development Life Cycle*). SDLC adalah sistem berjalan yang digunakan oleh suatu organisasi yang dikembangkan untuk memperbaiki kekurangan sistem (Sri, 2017). Metode SDLC ditunjukkan pada Gambar 1. Adapun beberapa tahapan dalam SDLC, yaitu:

1. *Planning*

Tahapan ini merupakan tahapan awal perencanaan indentifikasi dan penentuan *scope*

yang akan dilakukan selama proses pengembangan proyek. Pada tahap ini juga akan dilaksanakan pengumpulan informasi yang dibutuhkan, setelah itu dilakukan perencanaan *budget*, *time frame*, dan lainnya yang akan membawa proyek lebih teratur dan terorganisir.

2. *Analysis*

Analysis merupakan tahapan kedua untuk analisis kebutuhan sistem dari rencana yang telah dibuat pada tahap awal. Analisa yang dilakukan berupa analisa kelebihan dan kekurangan sistem, masalah, kegunaan, serta tujuan dan target yang ingin dicapai. Hal ini digunakan guna mencapai hasil yang sesuai ekspektasi.

3. *Design*

Design merupakan tahap membuat rencana atau spesifikasi desain berdasarkan hasil analisis yang sudah ada pada tahap kedua. Tahap ini menghasilkan desain, gambaran, dan dokumen yang akan diwujudkan dalam proyek. Aspek desain yang ditentukan meliputi *Architecture*, *User Interface*, *Platform*, dan *Security*.

4. *Coding*

Coding merupakan tahap implementasi berdasarkan hasil rancangan desain yang telah dibuat pada tahap sebelumnya. Pada tahap ini, tim pengembang mulai membangun sistem dengan menulis kode dengan bahasa pemrograman yang ditentukan. Implementasi yang dibuat berupa pembuatan *database* dan aplikasi. Tahapan ini menjadi fase terpanjang selama pengembangan sistem dalam SDLC.

5. *Testing*

Testing merupakan tahap kelima yang melibatkan QA (*Quality Assurance*) untuk melakukan *testing*. *Testing* yang dilakukan berupa pengecekan apakah sistem bekerja sesuai fungsionalitas yang diharapkan, tanpa ada *error* ataupun *bug*. Jika terjadi kendala sistem, maka tim pengembang yang akan melakukan perbaikan terlebih dahulu.

6. *Maintenance*

Maintenance menjadi tahapan akhir untuk pemeliharaan sistem supaya tetap optimal. Aktivitas yang biasanya dilakukan berupa perbaikan *bug* dan *upgrade* sistem, serta peningkatan fitur.



Gambar 1. Metode SLDC

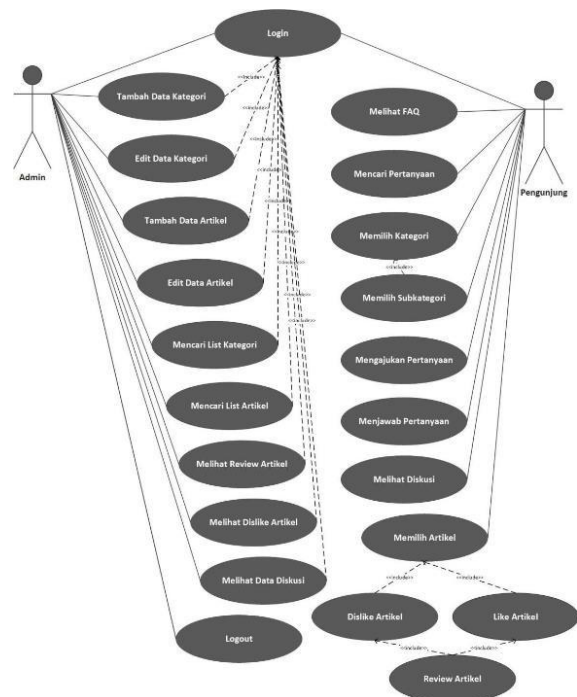
Model yang digunakan dalam perancangan sistem FAQ ini yaitu Agile Scrum, yang merupakan model SDLC dengan proses ramping namun mampu menghasilkan proyek akhir dengan kualitas tinggi (KYEREMEH, 2019). Agile Scrum mementingkan adanya kekuatan kolaborasi tim, inkremental produk, dan proses iterasi untuk mencapai tujuan (Srivastava dkk (2017)).

2. PEMBAHASAN

Berdasarkan hasil penelitian, perancangan Sistem *Frequently Asked Questions* diawali dengan perancangan diagram UML (*Unified Modeling Language*) yang merupakan metodologi pengembangan beberapa metode *Object Modeline Technique* dan *Object Oriented Software Engineering* (Nugroho, 2009). Beberapa jenis diagram UML yang paling sering digunakan yaitu *Use Case Diagram*, *Use Case Scenario*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, dan *Entity Relationship Diagram*.

1. Use Case Diagram

Perancangan *use case* dilaksanakan untuk melihat proses bisnis dan aktivitas yang dilakukan pada sistem. Dalam sistem *Frequently Asked Questions*, terdapat 2 aktor, yaitu admin dan pengunjung, dengan total 21 *use case*. Admin berupa aktor yang mengatur *Content Management System* untuk menginput ataupun memodifikasi data. Sedangkan, pengunjung adalah aktor yang akan melihat hasil FAQ pada *website* Ruparupa. *Use Case Diagram* ditunjukkan pada Gambar 2.



Gambar 2. Use Case Diagram

2. Use Case Scenario

Use Case Scenario berisi deskripsi mengenai seluruh aktor dalam melakukan aktivitas pada sistem dan cara menanggapi respon oleh sistem.

3. Activity Diagram

Activity Diagram merupakan lanjutan dari *Use Case Diagram* yang menggambarkan alur dan proses yang terjadi dalam sistem.

4. Sequence Diagram

Sequence Diagram merupakan diagram yang menjelaskan secara rinci mengenai interaksi antar objek, seperti mengirimkan dan menampilkan pesan, serta proses masuk ke dalam *database*.

5. Class Diagram

Class Diagram merupakan diagram yang menggambarkan deskripsi pada *class*, atribut, dan hubungan antar objek. Tujuannya untuk pemahaman skema sistem yang akan dirancang.

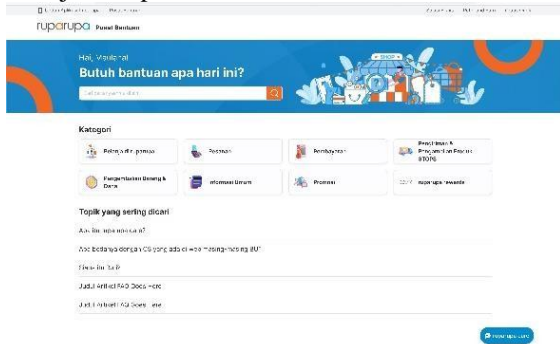
6. Entity Relationship Diagram

Entity Relationship Diagram merupakan diagram yang menggambarkan hubungan antar entitas secara rinci, dengan tujuan mempermudah penyusunan *database*. Dari ERD ini, Adapun keinginan untuk penggunaan *database* NoSQL yang akan mengubah strukturnya menjadi JSON.

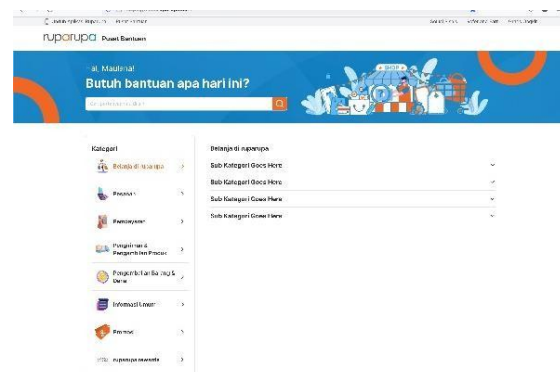
7. User Interface Design

User Interface Design merupakan proses pembuatan tampilan antarmuka yang nyaman dan mudah digunakan. Perancangan *User Interface Design* pada sistem *Frequently Asked Questions* terbagi menjadi 2 sisi, yaitu *UserInterface* Pengguna (*Website* Ruparupa) dan *User Interface* Admin (*Dashboard*). *User Interface* pengguna terbagi menjadi halaman utama FAQ, halaman kategori dengan setiap subkategori, halaman artikel dengan

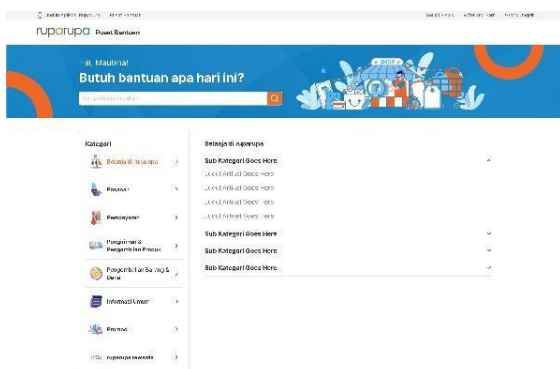
bagian review, hubungi kami, halaman diskusi, halaman pencarian beserta fitur terakhir dilihat. Hasil dari perancangan *User Interface* Pengguna ditunjukkan pada Gambar 3 - Gambar 7.



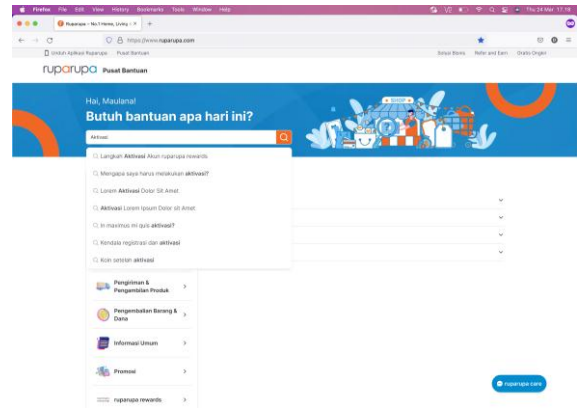
Gambar 3. Halaman Utama FAQ



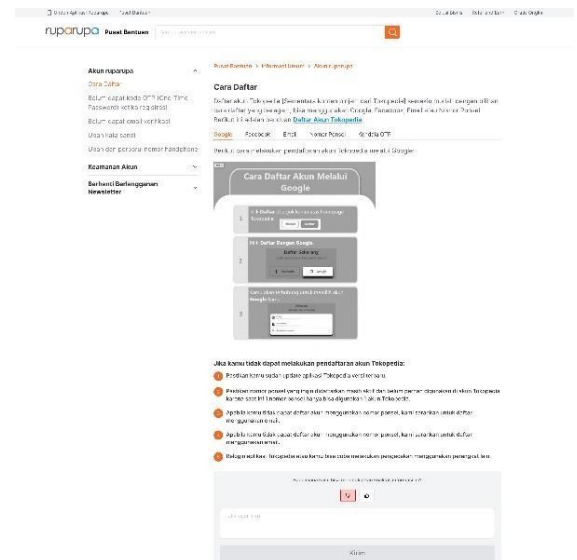
Gambar 4. Halaman Kategori FAQ



Gambar 5. Halaman Subkategori FAQ

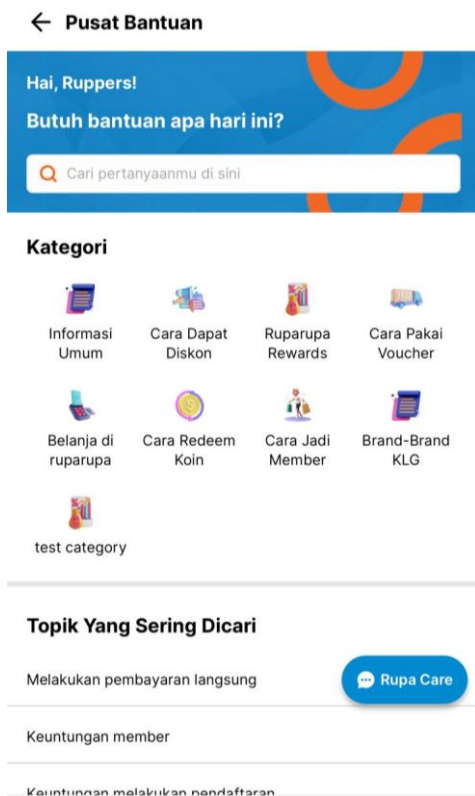


Gambar 6. Fitur Pencarian FAQ



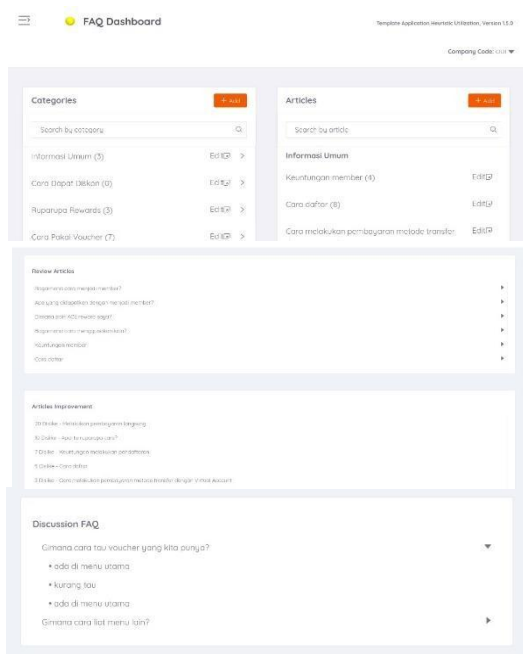
Gambar 7. Halaman Artikel FAQ

User Interface Design pada sistem FAQ ini juga dapat dilihat pada *website mobile* dengan tampilan utama seperti yang ditunjukkan pada Gambar 8.

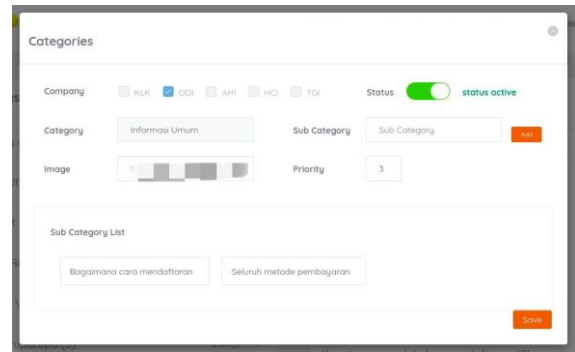


Gambar 8. Halaman Utama FAQ (Mobile)

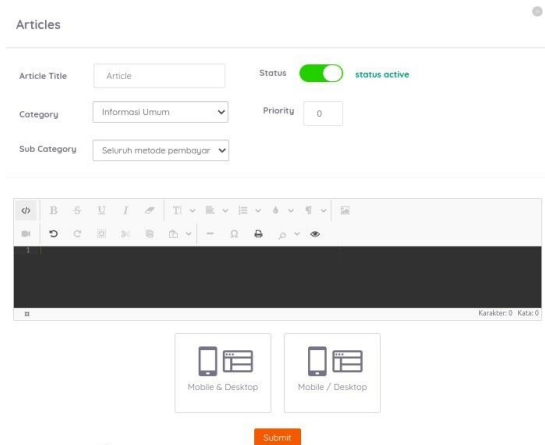
User Interface admin terbagi menjadi halaman utama, form membuat dan edit kategori, form membuat dan edit artikel, form melihat articles improvement, melihat diskusi, dan fitur search. Hasil dari perancangan User Interface Admin ditunjukkan pada Gambar 9 – Gambar 11.



Gambar 9. Halaman Utama FAQ (Admin)



Gambar 10. Form Kategori FAQ



Gambar 11. Form Artikel FAQ

Setelah tahapan perancangan dibuat, dilanjutkan ke tahap implementasi, antara lain yaitu:

1. Implementasi Basis Data

Hasil rancangan Entity Relationship Diagram diimplementasikan ke dalam bentuk basis data. Basis data yang digunakan pada sistem ini adalah database NoSQL yaitu MongoDB, yang berbasis dokumen dengan format file berupa JSON (Javascript Object Notation). Adapun perbedaan antara database jenis SQL dan NoSQL yang dijabarkan pada Tabel 1.

Tabel 1. Perbedaan SQL dan NoSQL

| SQL | NoSQL |
|--|--|
| Tipe relasional yang terstruktur | Tipe non-relasional yang tidak terstruktur |
| Skema standar untuk data terstruktur | Skema dinamis untuk data tidak terstruktur |
| Cenderung lambat dalam transformasi data | Cenderung cepat dalam transformasi data |

2. Implementasi Front-End

Hasil implementasi Front-End dibuat berdasarkan hasil rancangan User Interface yang sudah dibuat. Implementasi dilakukan dengan menggunakan bahasa pemrograman HTML, CSS, dan Javascript dengan ReactJS.

Tahapan implementasi melibatkan kegiatan pembuatan program aplikasi dengan penjabaran proses lengkap sebagai berikut:

1. Setup MongoDB dan Membuat Collection untuk kategori, artikel, review artikel, dan diskusi
2. Penulisan kode pemrograman dengan Visual Studio Code menggunakan ReactJS
3. Implementasi JSON
4. Menyambungkan *Dashboard* dan *Database*
5. Membuat *Query* untuk *Dashboard*
6. Integrasi API dan *Frontend*
7. Penulisan Kode Pemrograman Frontend

Setelah tahap implementasi dilakukan, Adapun pengujian unit yang harus dilakukan untuk memastikan bahwa sistem dalam program berhasil dijalankan sesuai ekspektasi tanpa adanya kesalahan secara keseluruhan. Pengujian dilakukan terhadap setiap fungsi yang bersangkutan dengan *database*. Pengujian dilakukan menggunakan metode *Black Box Testing* dan *User Acceptance Test*. *Black Box Testing* berfokus pada fungsional *input* dan *output* aplikasi. Sedangkan, UAT dilakukan untuk mengetahui apakah sistem telah memenuhi kebutuhan pengguna. Hasil dari pengujian Dashboard Admin ditunjukkan pada Gambar 12 dan pengujian *Website* ditunjukkan pada Gambar 13.

| No | Function | Kondisi diuji | Harapan hasil | Hasil |
|----|--------------------------|--|--|-------|
| 1 | getCategoriesContent | FIND seluruh data kategori dengan <i>filter keyword</i> dan <i>company</i> | Data kategori didapatkan sesuai dengan <i>filter</i> | Pass |
| 2 | getArticleCategories | FIND seluruh data artikel sesuai kategori yang diklik | Data artikel didapatkan sesuai dengan kategori | Pass |
| 3 | getReviewArticlesContent | FIND seluruh data review artikel | Data review artikel didapat | Pass |
| 4 | getDiscussionContent | FIND seluruh data diskusi | Data diskusi didapat | Pass |
| 5 | insertCategoriesContent | INSERTONE data kategori | Menyimpan data kategori | Pass |
| 6 | updateCategoriesContent | UPDATEONE data kategori yang ada | Data kategori terupdate | Pass |
| 7 | insertArticlesContent | INSERTONE data artikel | Menyimpan data artikel | Pass |
| 8 | updateArticlesContent | UPDATEONE data artikel yang ada | Data artikel terupdate | Pass |

Gambar 12. Unit Testing Dashboard Admin

| No | Function | Kondisi diuji | Harapan hasil | Hasil |
|----|--------------------------|---|--|-------|
| 1 | getCategoriesContent | GET seluruh data kategori dengan <i>filter keyword</i> dan <i>company</i> | Data kategori didapatkan sesuai dengan <i>filter</i> | Pass |
| 2 | getArticleCategories | GET seluruh data artikel sesuai kategori yang diklik | Data artikel didapatkan sesuai dengan kategori | Pass |
| 3 | getArticlesContent | GET seluruh data artikel dengan <i>filter keyword</i> dan <i>company</i> | Data artikel didapatkan sesuai dengan <i>filter</i> | Pass |
| 4 | getDiscussionContent | GET seluruh data diskusi | Data diskusi didapat | Pass |
| 5 | getReviewArticlesContent | GET seluruh data review artikel | Data review artikel didapat | Pass |
| 6 | insertReviewArticles | POST data review artikel | Menyimpan data review | Pass |
| 7 | insertDiscussionContent | POST data diskusi | Menyimpan data diskusi | Pass |

Gambar 13. Unit Testing Website

Setelah melalui *Unit Testing*, dilakukanlah pengintegrasian *unit* menjadi satu sistem secara keseluruhan, yang dilanjutkan dengan pengujian tahap akhir oleh *end-user* dengan UAT (*User Acceptance Testing*). *End-user* disini adalah pengguna yang akan berinteraksi dengan sistem yang dibuat. Pengujian UAT dilakukan untuk memastikan sistem berjalan sebagai solusi untuk pengguna (Vibiola, Octariadi & Alkadri). Adapun kuisioner *User Interface* guna mengetahui nilai dan saran sistem yang dibuat.

3. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang dilakukan untuk perancangan sistem *Frequently Asked Questions* pada *website* Ruparupa, maka dapat diambil beberapa kesimpulan.

1. Sistem *Frequently Asked Questions* mempermudah *user* dalam mengelola data FAQ melalui tampilan *Content Management System* yang mudah digunakan.
2. Perancangan sistem *Frequently Asked Questions* berbasis *website* dirancang untuk memenuhi kebutuhan pengguna dengan penyediaan berbagai fitur, seperti pencarian, pengkategorian artikel berdasarkan kategori dan subkategori, *review* artikel, serta diskusi.
3. *Frequently Asked Questions* yang saat ini ada pada Ruparupa tidak jarang membuat konsumen terus mengajukan pertanyaan berulang dan kebingungan saat mencari informasi. Sehingga, sistem FAQ yang baru membantu meringankan permasalahan yang ada saat ini.
4. Adanya sistem *Frequently Asked Questions* ini mempermudah konsumen dalam mencari berbagai informasi.

5. Penggunaan metode *scrum* bermanfaat sebagai komunikasi antar anggota tim dengan menerapkan *daily scrum* setiap harinya.

PUSTAKA

- Arthana, I., Dantes, G., Dewi, L., Setemen, K., & Marti, N. (2021). *PENGEMBANGAN PROTOTYPE FREQUENTLY ASKED QUESTION (FAQ) UNDIKSHA DENGAN PENDEKATAN USER CENTERED DESIGN*.
- Cabot, J. (2018). WordPress: A Content Management System to Democratize Publishing. *IEEE Software*, vol. 35, no. 3, pp. 89-92, doi: 10.1109/MS.2018.2141016.
- Investopedia. (2022, Juli 6). *Ecommerce Defined: Types, History, and Examples*. Investopedia. <https://www.investopedia.com/terms/e/ecommerce.asp>
- KYEREMEH, K. (2019). OVERVIEW OF SYSTEM DEVELOPMENT LIFE CYCLE MODELS. Available at SSRN 3448536, *Papers.Srn.Com*.
- Mulyani, Sri, Ak., CA. (2017). Metode Analisis dan Perancangan Sistem. Abdi Sistemika
- Nugroho, A. (2009). *Rekayasa Perangkat Lunak Menggunakan UML dan Java*. Penerbit Andi.
- Rheny, S. (2022, Februari 24). *Kenali apa saja fungsi FAQ dan tips membuat FAQ yang efektif*. <https://www.ekrut.com/media/faq-adalah>
- Setyadhi, E. M., & Johan, M. C. (2022, Mei). *View /of Aplikasi Berbasis Web untuk Fitur Frequently Asked Question Bagi Mahasiswa dan Dosen Koordinator Tugas Akhir*. Aplikasi Berbasis Web untuk Fitur Frequently Asked Questions.
- Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017, May). *SCRUM model for agile methodology*. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 864-869). IEEE.
- Suprpto, "User Acceptance Testing (UAT) Refreshment PBX Outlet Site BNI Kanwil Padang," *J. Civronlit Unbari*, vol. 6, no. 2, p. 54, 2021, doi: 10.33087/civronlit.v6i2.85.
- Vibiola, Vera., Ocatriadi, B. C & Alkadri S. P. (2022). *SISTEM INFORMASI GEOGRAFIS PERSEBARAN PERGURUAN TINGGI DI KOTA PONTIANAK MENGGUNAKAN METODE HAVERSINE FORMULA*. JOUTICA Volume 7 No 2 Tahun 2022.