

Undiksha Virtual Assistant (Shavira) : Integration Frequency Asked Question with Rasa Framework

I Ketut Resika Arthana¹, Luh Joni Erawati Dewi², Ketut Agus Seputra³, Ni Wayan Marti⁴

^{1,2} Computer Science, Universitas Pendidikan Ganesha
^{3,4} Informatics Management, Universitas Pendidikan Ganesha
Singaraja, Indonesia

e-mail: resika@undiksha.ac.id, joni.erawati@undiksha.ac.id, agus.seputra@undiksha.ac.id,
wayan.marti@undiksha.ac.id

Abstract

Nowadays, the implementation of chatbots in information systems is getting popular to increase user experience. In our previous research, we developed Undiksha Virtual Assistant (Shavira). Shavira is a knowledge management system (KMS) as Frequency Asked Question (FAQ) system to facilitate users to get an answer about their questions. The challenge on the existing Shavira system is that the users must open a website and type queries to get a list of answers. The text query must match with the keyword in systems. To address this problem and to avoid cognitive load, in this research we implemented a chatbot in Shavira. The purpose of the chatbot implementation in Shavira is to make information easier to be accessed and to increase user experience. The users can ask a question in natural language and get an answer immediately on their chat application such as: Telegram, Facebook, or Messenger. We used Rasa Framework as a chatbot engine in Shavira. Rasa Framework is an open-source virtual assistant engine based on artificial intelligence. The challenge in conducting this research is in integrating the existing FAQ system with Rasa Framework. This research consists of four phases, they are: adaptation data structure in the existing FAQ with data structure requested by Rasa framework, data integration and mapper, and evaluation. The result of this research is a new Shavira that is integrated with a chatbot and can be accessed from chat applications such as Telegram. We evaluated the accuracy of Shavira powered by Rasa Framework with ten topics about academic information. The results showed that the accuracy of Shavira is 90%. We also evaluated the usability of Shavira with Software Usability Scale (SUS) Questionnaire. The usability evaluation showed that Shavira satisfied users with value 81 (threshold 68) and categorized Excellent, Acceptable and Prometer.

Keywords: Shavira, Chatbot, Rasa Framework

INTRODUCTION

A good information quality in terms of time, accuracy and relevancy is very important to be given to users as a part of a service. Information quality helps users to achieve their goals by utilizing the services provided. In the previous research, we developed a web-based Undiksha FAQ system to help users get answers of the frequently asked questions. The FAQ stores data on frequently asked questions and answers to those questions. With the FAQ, it is hoped that users will get answers easily to the questions.

Searching information through website requires more effort by users to find the exact information. The user has to open the browser, then open the website address, type the question keyword and finally get the answer. These steps are less efficient if the user wants to get a faster answer. One of the most popular technological innovations today to provide automatic answers to users is Chatbot technology.

Currently, many organizations use technology to provide services to customers automatically, namely Chatbot technology (Almansor & Hussain, 2020). According to Sandbank, Shmueli-Scheuer, Herzig,

Konopnicki, Richards & Piorkowski (2018), about 80% company plan to implement chatbot. The main benefit of using a chatbot in an organization is that it can automatically answer customer questions quickly. In addition, the use of chatbots also reduces the operating costs of providing customer service. Chatbots have been widely used in both the private and public sectors such as: banks, government, tourism, the stock market and so on. Chatbot also can use in education area (Clarizia et al., 2018). Chatbot technology is now equipped with voice technology including Siri, Alexa, Google Assistant and Cortana (Androutsopoulou et al., 2019). The method to search for answers by the chatbot utilizing machine learning technology to perform a sophisticated retrieval process, where responses are generated based on analysis from web searches (Cahn, 2017). Chatbot also can used in Veterinary consultation (Huang & Chueh, 2021).

The popularity of chatbot implementations is supported by various chatbot technologies. There are various technologies that can be used to develop chatbots such as: Artificial Intelligence Markup Language (AIML) with interpreter. Another examples are Dialogflow, Microsoft Bot and RASA Framework (Singh et al., 2019). In addition, developers can also develop from scratch by utilizing NLP and Machine learning technologies. The choice of chatbot technology is tailored to the needs or design of the chatbot application.

In this research, the FAQs that have been developed previously will be integrated with Rasa Framework. RASA is an open source framework consists of stack machine learning consists of libraries for developers to create contextual chatbots. The FAQ serves as a data source that will be linked to the RASA Framework via the webservice

API, then the data will be trained on the RASA framework to get a chatbot model. The model is used as a service that is connected to the presentation layer such as: Telegram and Facebook.

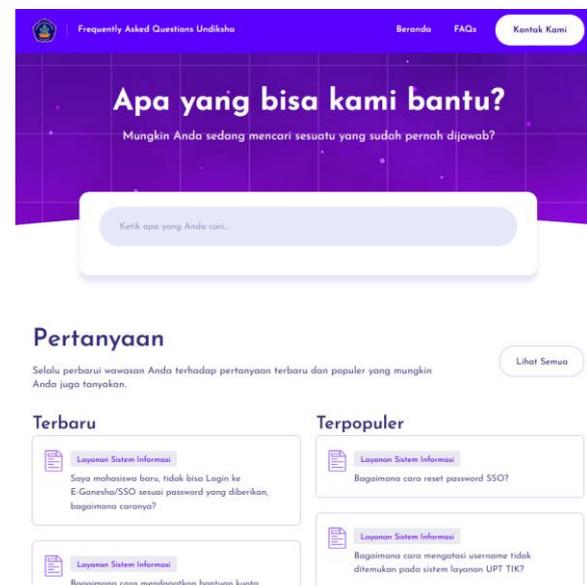


Figure 1. FAQ Undiksha User interface

Undiksha Virtual Assistant (Shavira) is an intelligence system based on natural language processing (NLP) that can help users to get information with natural language. On our previous research, we have developed FAQ Undiksha as a part of Shavira with User Centered Design approach (Resika Arthana et al., 2021). FAQ Undiksha is a web-based application used to facilitate the users to get answer related to the questions they have (Figure 1). FAQ Undiksha is developed with PHP programming language and MySQL database. FAQ Undiksha can be used by any organizations and any activities at Undiksha. The organizations that have an activity can manage their lists of questions and answers. The overall workflow of FAQ Undiksha depicted in Figure 2.

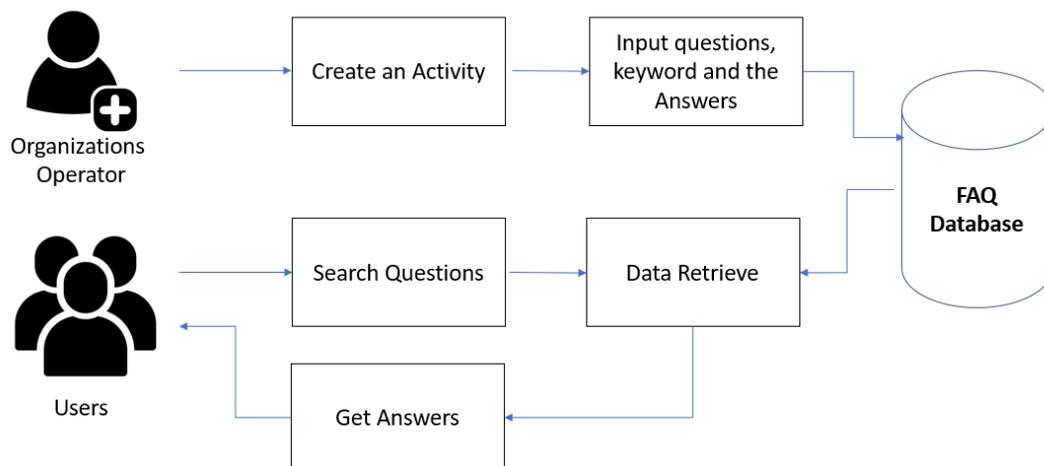


Figure 2. User flow on existing FAQ Undiksha (Resika Arthana et al., 2021)

Design for chatbots represents a transition from the design of visual layout and interaction mechanisms to the design of conversation (Følstad & Brandtzaeg, 2017). In this research, we proposed improvisation capability of Shavira with the implementation of chatbot. The purposed implementation chatbot in Shavira is providing access information with online chat conversation in natural language. With Shavira powered by chatbot, the users will be easier to access information.

In this research, we use RASA Framework as the engine of chatbot. Rasa is an open source machine learning framework for automated text and voice-based conversations. Rasa capable to understand messages, hold conversations, and connect to messaging channels and APIs (Rasa, 2021).

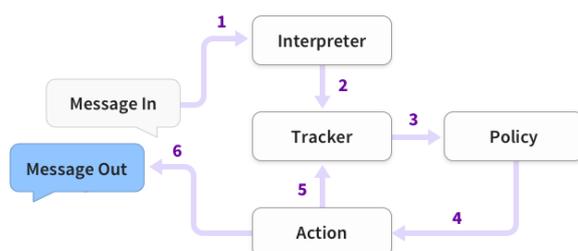


Figure 3. Rasa Open Source architecture (Rasa, 2021)

The message that is sent to Rasa will be received by Natural Language Understanding (NLU). NLU consists of Interpreter. Interpreter correspond to convert text to dictionary consist of text, the intent and any entities. Then, the message will be passed to the Tracker. Tracker correspond to keep track of conversation state. The action based on text chooses by policy, then logged to the tracker and send response to the user by web page or other service. In this research, we use telegram as interface chatbot by users.

There are many previous researches that implemented virtual assistant with rasa Framework. For example, Deepika at all design the implementation of intelligent virtual laboratory using RASA framework (Deepika et al., 2021). The purpose of this research is to help students to get answer about mistakes in C programming language with virtual assistant. (Jiao, 2020) integrated RASA NLU with neural network (NN) methods implements the system based on entity extraction after intent recognition. (Windiatmoko et al., 2021) using RASA Framework for University Enquiries.

METHODS

In this research, we improved Shavira with the implementation of Chatbot technology based on the existing FAQ

Undiksha. FAQ stored frequently asked questions and the answers. Figure 4 shows the implementations of the research workflow. Due to different data format between FAQ Undiksha and Rasa Framework, we must make data mapping from FAQ Undiksha and NLU. After data mapping, we trained the data in Rasa Framework to get Chatbot Models. The Chatbot Models used when there is an interaction in Chatbot. Then we integrated with Telegram Application to have an easier chatbot access. So, the Chatbot implementation on FAQ Undiksha consists of 3 steps. First we developed Data Integrator dan Mapper, second we developed Chatbot with Rasa Framework, third we integrated Rasa Framework with Telegram and the last we evaluated the accuracy and usability of Shavira.

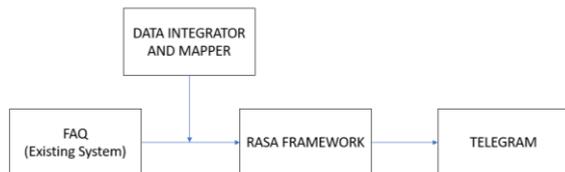


Figure 4. Proposed System Workflow

1. Data Integrator and Mapper

In the existing FAQ Undiksha system, we stored data as single pair question and answer. Due to Rasa Framework needs more than one questions as alternatives and allowed to give alternative answers, we modified existing schema database in existing

FAQ Undiksha. The new schema database depicted in Figure 5.

Before		After	
question_answer		question_answer	
id	int	id	int
activity_id	int	activity_id	int
questions	varchar	context	varchar
keyword	varchar	questions	varchar
answer	varchar	questions_alt1	varchar
created_at	timestamp	questions_alt2	varchar
		questions_alt3	varchar
		keyword	varchar
		answer	varchar
		answer_alt1	varchar
		answer_alt2	varchar
		answer_alt3	varchar
		created_at	timestamp

Figure 5. Modification data structure FAQ to comply Rasa Framework

Developing chatbot in Rasa Framework required config at least 3 files in ".yml" format. These files consists of nlu.yml, domain.yml, rules.yml. The nlu.yml file store list of questions and intent as category (Figure 6. nlu.yml). Every intent consist of one or more questions. The domain.yml file store responses based on the intents as action from questions (Figure 7. domain.yml). We also can put image as responses. In the domain.yml file, we can put one or more texts answer on every intents. The rules.yml file store rules or paths between intent and action (Figure 8. rules.yml). If we need to the conversation between user and chatbot in sequential ask and action, we can configure in stories.yml.

```

1 version: "2.0"
2 nlu:
3   - intent: salam
4     examples: |
5       - halo
6       - selamat pagi
7       - selamat siang
8       - selamat malam
9       - om swastiatu
10      - hi
11
12    - intent: jadwalkrs
13      examples: |
14        - kapan jadwal KRSan
15        - kapan jadwal krs
16        - KRS kapan sih
17        - kapan menyusun kartu rencana studi
18
19    - intent: syaratkrs
20      examples: |
21        - Apa syarat KRSan
22        - syarat KRS
23        - langkah-langkah menyusun KRS
24        - langkah menyusun Kartu Rencana Studi
25
26    - intent: kalender
27      examples: |
28        - Rungsa kalender akademik 2021
29        - Minta kalender akademik
30        - Ada kalender akademik
31
32    - intent: metodekuliah
33      examples: |
34        - Kuliah ganjil 2021 online atau offline
35        - Kuliah ganjil 2021 daring atau luring
  
```

Figure 6. nlu.yml

```

1 version: "2.0"
2
3 intents:
4   - salam
5   - jadwalkrs
6   - syaratkrs
7   - kalender
8   - metodekuliah
9
10 responses:
11   utter_salam:
12     - text: "Halo, ada yang bisa saya bantu?"
13
14   utter_jadwalkrs:
15     - text: "Sesuai kalender akademik, Pengisian KRS dimulai dari 2-14 Agu"
16
17   utter_syaratkrs:
18     - text: "Syarat bisa menyusun KRS Ganjil 2021 meliputi : (1) Sudah mem
19
20   utter_kalender:
21     - text: "Kalender akademik 2021 bisa dilihat di https://akademik.undik
22
23   utter_metodekuliah:
24     - text: "Terkait pandemi covid19, perkuliahan ganjil 2021/2022 dilaksa
25
  
```

Figure 7. domain.yml

```

1 version: "2.0"
2
3 rules:
4   - rule: jawab salam
5     steps:
6       - intent: salam
7       - action: utter_salam
8
9   - rule: jawab jadwalkrs
10    steps:
11      - intent: jadwalkrs
12      - action: utter_jadwalkrs
13
14   - rule: jawab syaratkrs
15    steps:
16      - intent: syaratkrs
17      - action: utter_syaratkrs
18
19   - rule: jawab kalender
20    steps:
21      - intent: kalender
22      - action: utter_kalender
23
24   - rule: jawab metodekuliah
25    steps:
26      - intent: metodekuliah
27      - action: utter_metodekuliah
28
  
```

Figure 8. rules.yml

Due to different technology and format between FAQ system and Rasa Framework, we developed integrator and mapper

module. Integration module developed based on web service technology. Integration module corresponding to

synchronize data from FAQ to file config on Rasa Framework.

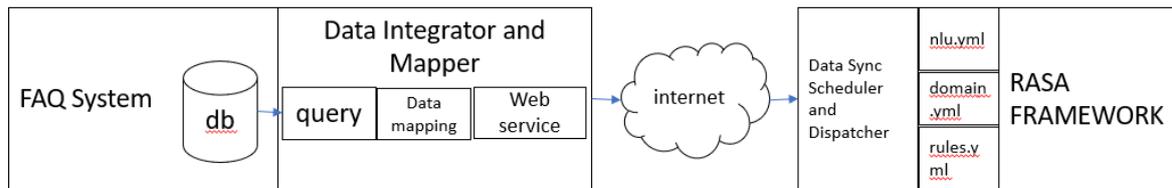


Figure 9. Architecture Data Integrator and Mapper

2. Chatbot Development

We provide access to Shavira chatbot via Telegram application. Telegram is free and opensource instant messaging (IM) system. We use Telegram as interaction layer because Telegram is one of the most popular messaging applications and most of our staff and students use that application to communicate.

The integration of Rasa Framework with Telegram consists of several steps. First, we published service Rasa via https then we created Bot on Telegram via botFather. The final step is setup a webhook between telegram with Rasa using webhook URL. After that, we can communicate to chatbot via Telegram as show in Figure 10.

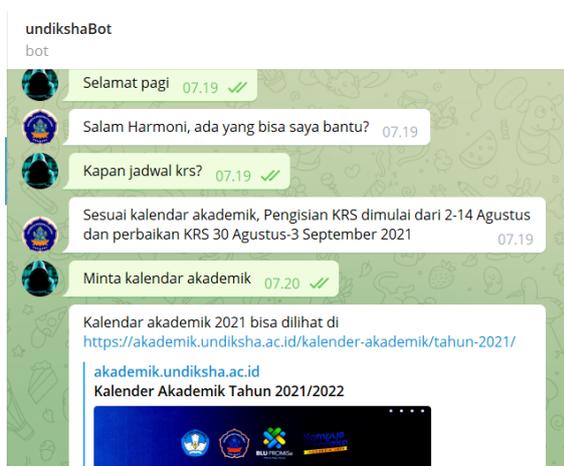


Figure 10. Access Chatbot via Telegram

3. Integration

As show on Figure 11, we modified form input on FAQ Undiksha to support more than one alternative questions. The alternative questions is useful to make training data more reliable. Users input data of question and answer then the data stored in database.

Figure 11. Input Question Answer FAQ Undiksha

Integrator and Mapper publish data based on data on database as a web service. The webservice can be accessed by endpoint url. As depicted at Figure 12, the response from web service formatted as json.

```

1 [{"intents": [
2     {"tag": "jadwalkrs",
3       "patterns": ["Kapan jadwal KRSan", "kapan jadwal krs", "KRS kapan sih","Kapan menyusun kartu rencana studi"],
4       "responses": ["Sesuai kalender akademik, Pengisian KRS dimulai dari 2-14 Agustus dan perbaikan KRS 30 Agustus-3
5     ]},
6     {"tag": "kalendarAkademik",
7       "patterns": ["Punya kalender akdemik 2021", "Minta kalender akademik", "Ada kalender akademik"],
8       "responses": ["Kalender akademik 2021 bisa dilihat di https://akademik.undiksha.ac.id/kalender-akademik/tahun-20
9     ]},
10    {"tag": "metodeperkuliahan",
11      "patterns": ["Kuliah ganjil 2021 online atau offline", "Kuliah ganjil 2021 daring atau luring"],
12      "responses": ["Terkait pandemi covid19, perkuliahan ganjil 2021/2022 dilaksanakan secara daring"]
13    },
14    {"tag": "apasyaratkrs",
15      "patterns": ["Apa syarat KRSan", "syarat KRS", "langkah-langkah menyusun KRS", "langkah menyusun Kartu Rencana Stu
16      "responses": ["Syarat bisa menyusun KRS Ganjil 2021 meliputi : (1) Sudah membayar UKT, (2) Mengisi link survey,
17    ]},
18    {"tag": "orangtuakhs",
19      "patterns": ["Bagaimana cara orang tua konfirmasi khs", "dimana melihat akun orang tua", "apa yang harus dilakuka
20      "responses": ["Untuk konfirmasi bahwa orang tua sudah melihat KHS : (1) Lihat akun orang tua di profil SSO lalu
21    ]},
22  ],
23  {"tag": "akunSSO",
24    "patterns": ["Dimana melihat akun SSO"],
25    "responses": ["Mahasiswa baru bisa melihat akun SSO E-Ganesha bisa dilihat di https://daftarkembali.undiksha.ac
26  },
27  {"tag": "akunSSO",
28    "patterns": ["Saya lupa password SSO", "SSO tidak bisa login", "cara reset password SSO"],
29    "responses": ["Reset akun SSO E-Ganesha dilakukan dengan klik 'Lupa Password' pada https://sso.undiksha.ac.id/
30  ]}
31 ]}

```

Figure 12. JSON output from web service

Data sync and scheduler retrieve data from web service. Scheduler retrieve data based on the time that has been set before. Data JSON received by dispatcher and write nlu.yml (Figure 13), domain.yml (Figure 6) and rules .yml (Figure 7)

On Rasa framework, we configure some parameter to make chatbot as FAQ Assistants (Figure 14). First one, we updated config.yml in policies parameter. We used RulePolicy as policies, so the chatbot will respond the same way every time the same type of question is being asked. Next, we set ResponseSelector in NLU pipeline. The ResponseSelector will build a single retrieval model for all retrieval intents

```

1 # Configuration for Rasa NLU.
2 # https://rasa.com/docs/rasa/nlu/components/
3 language: en
4
5 pipeline:
6 - name: ResponseSelector
7   epochs: 100
8   retrieval_intent: faq
9
10 policies:
11 - name: RulePolicy

```

Figure 14. FAQ Configuration

4. Evaluation

We design evaluation chatbot by developing interactive conversation to confirm that the answer of chatbot is correct or not. The users ask something to chatbot,

then the chatbot will answer and ask to the user whether the answer is correct or not. Correct answer means that answer of chatbot appropriate with context of questions. With this method, we can get the accuracy of chatbot.

We also evaluated usability of chatbot with Software Usability Scale (SUS) to get users satisfaction level. SUS consists of 10 question with 5 options, from strongly agree to strongly disagree (Sharfina & Santoso, 2016) (Arthana et al., 2019).

RESULTS AND DISCUSSION

We use 15 sample questions as data in this research. The questions are about academic information in Undiksha. We input the data in FAQ system, then we synchronize to RASA Framework. After synchronization has done, then the RASA train data and save as model. The model loaded by RASA CORE and provide access to telegram. The users can interact with bot via telegram.

Data from FAQ Undiksha is extracted via web service formatted as JSON Format ((shows in Figure 12). Data from JSON mapped and saved as files that required by RASA Framework. The files are "nlu.yml", "domain.yml" and "rules.yml" (Figure 6, Figure 7, Figure 8).

the context due to misleading term. For example, when users asked about Kartu Rencana Studi (KRS) the users use term 'Mata Pelajaran'. They use synonym term that is not defined in data training (nlu.yml). In the future, the collection of questions on each topic should be more varied to suit the user's questions.

improve interaction between Shavira and users. This limitation potential to reduce quality of conversations. On future research, we must implement Indonesian language in chatbot like what have been done in previous research by (Elcholiqi & Musdholifah, 2020).

Rasa Framework not supported Indonesian language yet. We cannot utilize *spacy_sklearn* pipelines in config.json to

Table 3. SUS Data Respondent

Respondent	Value answer from Respondent										Total	Final Value (Total x 2.5)
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10		
R1	4	3	4	4	4	4	4	3	3	4	37	93
R2	4	2	4	2	4	2	4	2	4	2	30	75
R3	4	3	3	3	2	3	2	2	4	3	29	73
R4	4	2	4	2	4	3	4	2	3	2	30	75
R5	4	2	3	4	3	3	4	4	3	3	33	83
R6	3	3	3	3	4	4	4	3	4	4	35	88
R7	4	3	3	4	4	3	3	3	3	3	33	83
R8	4	2	4	4	4	3	4	4	2	3	34	85
R9	3	3	3	4	4	2	4	2	3	3	31	78
R10	4	3	4	3	4	3	3	3	4	2	33	83
R11	2	3	4	3	3	3	3	3	4	3	31	78
R12	4	0	3	3	3	3	3	3	3	3	28	70
R13	3	3	4	3	4	4	4	3	3	4	35	88
R14	4	3	4	3	4	3	4	3	3	4	35	88
R15	4	2	3	3	3	3	4	2	3	3	30	75
Average	4	2	4	3	4	3	4	3	3	3,1		
Average												81

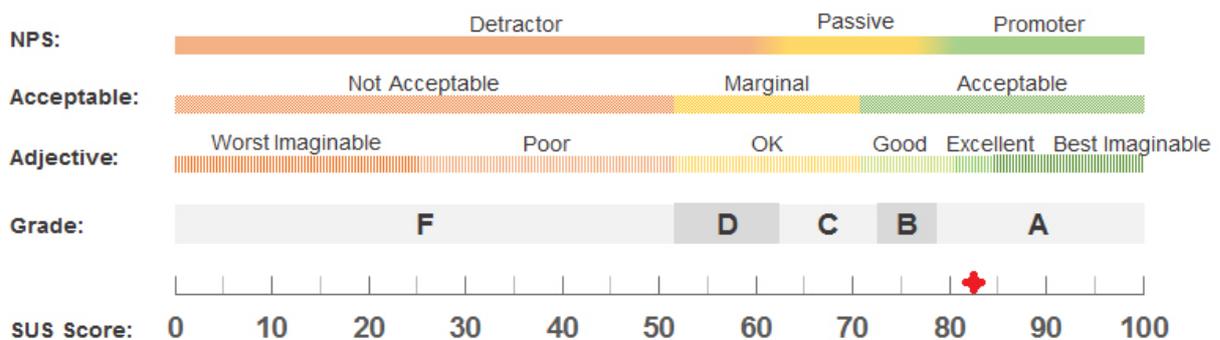


Figure 17. SUS Scale Position (modified from (Sauro, 2018))

In addition to evaluation of accuracy, we also evaluate usability. We evaluate

user satisfaction with Software Usability Scale (SUS). The Questions of SUS consists of.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

The respondent of SUS Questionnaire is 15 respondents as shown at Table 3. The results of SUS questionnaire analysis showed that users satisfaction level in Shavira interaction is 81. According to (Jeff Sauro, 2011), the threshold of results SUS questionnaire is 68. So, the user satisfaction users in Shavira categorized as Excellence, Acceptable and Promoter (Figure 17).

CONCLUSION

We have improved capability of Shavira from web-based FAQ to implementation with chatbot powered by Rasa Framework. The challenge in this research are data integration between existing FAQ and Rasa framework done by data Integrator and Mapper. Data from FAQ that is provided by web service and caught by Data Scheduler and Dispatcher then saved as file that required by RASA Framework. To make access to chatbot easier, we integrated with Telegram Application. The evaluation show that accuracy of Shavira is 90% and the

users satisfied with Excellence, Acceptable and Promoter category.

ACKNOWLEDGEMENT

This research funded by Lembaga Penelitian dan Pengabdian kepada Masyarakat Universitas Pendidikan Ganesha.

REFERENCES

- Almansor, E. H., & Hussain, F. K. (2020). Survey on Intelligent Chatbots: State-of-the-Art and Future Research Directions. *Advances in Intelligent Systems and Computing*, 993(August 2019), 534–543. https://doi.org/10.1007/978-3-030-22354-0_47
- Androutsopoulou, A., Karacapilidis, N., Loukis, E., & Charalabidis, Y. (2019). Transforming the communication between citizens and government through AI-guided chatbots. *Government Information Quarterly*, 36(2), 358–367. <https://doi.org/10.1016/j.giq.2018.10.001>
- Arthana, I. K. R., Pradnyana, I. M. A., & Dantes, G. R. (2019). Usability testing on website wadaya based on ISO 9241-11. *Journal of Physics: Conference Series*, 1165(1), 12012.
- Cahn, J. (2017). *CHATBOT: Architecture, Design, & Development*. University of Pennsylvania.
- Clarizia, F., Colace, F., Lombardi, M., Pascale, F., & Santaniello, D. (2018). Chatbot: An education support system for student. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 11161 LNCS*. Springer International Publishing. https://doi.org/10.1007/978-3-030-01689-0_23
- Deepika, N. M., Bala, M. M., & Kumar, R. (2021). Design and implementation of intelligent virtual laboratory using RASA framework. *Materials Today: Proceedings*, xxx, 1–5. <https://doi.org/10.1016/j.matpr.2021.01.226>

- Elcholiqi, A., & Musdholifah, A. (2020). Chatbot in Bahasa Indonesia using NLP to Provide Banking Information. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 14(1), 91. <https://doi.org/10.22146/ijccs.41289>
- Følstad, A., & Brandtzaeg, P. B. (2017). Chatbots and the New World of HCI. In *Interactions* (Vol. 24, Issue 4, pp. 38–42). <https://doi.org/10.1145/3085558>
- Huang, D.-H., & Chueh, H.-E. (2021). Chatbot usage intention analysis: Veterinary consultation. *Journal of Innovation & Knowledge*, 6(3), 135–144. <https://doi.org/10.1016/j.jik.2020.09.002>
- Jeff Sauro. (2011). *Measuring Usability with the System Usability Scale (SUS)*. Measuringu. <https://measuringu.com/sus/>
- Jiao, A. (2020). An Intelligent Chatbot System Based on Entity Extraction Using RASA NLU and Neural Network. *Journal of Physics: Conference Series*, 1487(1). <https://doi.org/10.1088/1742-6596/1487/1/012014>
- Rasa. (2021). *Introduction to Rasa Open Source*. <https://rasa.com/docs/rasa/>
- Resika Arthana, I. K., Dantes, G. R., Dewi, L. J. E., Setemen, K., & Marti, N. W. (2021). Pengembangan Prototype Frequently Asked Question (Faq) Undiksha Dengan Pendekatan User Centered Design. *Jurnal Pendidikan Teknologi Dan Kejuruan*, 18(1), 77. <https://doi.org/10.23887/jptk-undiksha.v18i1.32141>
- Sauro, J. (2018). *5 Ways to Interpret a SUS Score*. Measuringu. <https://measuringu.com/interpret-sus-score/>
- Sharfina, Z., & Santoso, H. B. (2016). An Indonesian adaptation of the System Usability Scale (SUS). *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 1566(1), 145–148. <https://doi.org/10.1109/ICACSIS.2016.7872776>
- Singh, A., Ramasubramanian, K., & Shivam, S. (2019). Introduction to Microsoft Bot, RASA, and Google Dialogflow. In *Building an Enterprise Chatbot* (pp. 281–302). Springer.
- Windiatmoko, Y., Rahmadi, R., & Hidayatullah, A. F. (2021). Developing Facebook Chatbot Based on Deep Learning Using RASA Framework for University Enquiries. *IOP Conference Series: Materials Science and Engineering*, 1077(1), 012060. <https://doi.org/10.1088/1757-899x/1077/1/012060>