Web Service for Search Engine Bahasa Indonesia (SEBI)

Husni1, Ika Oktavia Suzanti2, Yoga Dwitya Pramudita3, Sigit Susanto Putro4, Lukman Heryawan5

1,2,3,4Informatics Engineering Department, Faculty of Engineering, University of Trunojoyo Madura
5Department of Social Informatics Graduate School of Informatics Kyoto University, Japan

husni@trunojoyo.ac.id

Abstract. The purpose of this research is to produce an Indonesian language search engine (SEBI) that is in accordance with the objectives of the 2016 KKNI-based curriculum. SEBI 2017 is one of the complete search engines that is able to collect web pages from the internet, do preprocessing, build indexes, handle query and categorize documents. SEBI 2017, as a prototype has shortcomings, namely time and access. Based on this, the researchers built a new architecture called the SEBI API (2018). In the SEBI API architecture (2018), the ease and breadth of access, as well as the integration of data from various applications and different owners. RESTful API added to SEBI allows SEBI to be accessed from various access devices using various programming languages, libraries or frameworks that support REST API client (consumer) functions. Broadly speaking, the 2018 SEBI architecture has been equipped with RESTful API (provider) and client (consumer) functions. After testing by entering keywords, the results show that the SEBI API architecture (2018) has been able to correct the weaknesses contained in the SEBI architecture (2017)

1. Introduction
In accordance with the objectives of the 2016 KKNI-based curriculum that refers to ACM / IEE Computing Curricula [1] and APTIKOM [2], then in 2017, through a learning media innovation grant program, the Informatics Study Program (PSIF) of the University of Trunojoyo Madura (UTM) has produced a computational concept "teaching aid" called the SEBI [3]. SEBI is short of Search Engine Bahasa Indonesia. As a search engine that is quite complete, SEBI is able to collect the web pages of the Internet, perform preprocessing, build a system of index (inverted index), deal with queries or request documents from users, categorize the documents to the class that is appropriate and recommends a document that is relevant to user needs. The system architecture of SEBI I is shown in Figure 1.

As a prototype, SEBI 2017 also has several shortcomings. Although it has been useful as a teaching aid that is able to demonstrate the implementation of many algorithms from many courses in Information, SEBI 2017 still does not really represent the system in the real world (industry). The focus of SEBI is still on implementing the computational method, just reviewing how the method is realized in the program code. Important things in the current industry, are still not represented in the 2017 SEBI draft. The Industrial World prioritizes calculations that are realistic, practical and
minimalist in cost. Whereas in universities, lecturers and students tend to be idealistic, and see the form of computational algorithms only in terms of accuracy [4].

Search Engines like Google and Bing do not provide the latest and most relevant documents to Queries. But this Search Engine provides relevant documents in a very short time. From this description, we can understand that in the real world, time is very important in preparing search results.

Figure 1. Architecture of Systems Search Engine Languages Indonesia in 2017 (SEBI 2017)

Another factor that also includes the gap between SEBI 2017 and the industry is accessibility, which is how easy SEBI can be accessed by users. The [web] application architecture now at least divides the application into 2 main sides, namely : Backend and Frontend, although in its development can be put on in a single device. Backend is walked in on a web server and provides business and data that is required by the user [5]. Frontend is the interface for the user and can be built separately from the Backend. The two industry requirements above, namely speed and accessibility, do not get much attention in lectures even though the world of work is expecting experience even from fresh graduates. How to provide the search results are relevant to the query as quickly (no more 1 sec), while the Search Engine collects the document in a number of large? How to make the Search Engine that is built can be accessed through many Frontend? Broadly speaking, the right answer is to change the SEBI design and open access to many frontends. In accordance with the latest trends, a system that can be accessed from many devices must provide an API (application programming interface) [6].

API is the name of another of the web service. In general, web services are divided into two, namely SOAP and RESTful. Trends show that RESTful API is more widely used lately, one of which is because of its ease and flexibility [7, 8]. The updated SEBI design with fundamental changes and accommodating RESTful API technology, an implementation strategy that also considers speed in addition to accuracy, is expected to be a quality props.

2. Methodology

SEBI plus API (2018) seeks to emulate the needs of the industrial world, including adjusting software architecture and applying cache techniques. Heterogeneity of programming languages (web [9], desktop, mobile [10-12], appliance) and access to information devices (Laptops, Tablets, clock hands, smartphones) also need to be considered. How heterogeneity is accommodated in the SEBI? Trends in the application when it is the ease and breadth of
access, as well as the integration of data from various applications and the owner is different. RESTful API added to SEBI allows SEBI to be accessed from various access devices using various programming languages, libraries or frameworks that support REST API client (consumer) functions. Broadly speaking, the 2018 SEBI architecture that has been equipped with RESTful API (provider) is shown in Figure 2.

Figure 2. SEBI Plus RESTful API Architecture (2018)

Figure 2 shows that various clients (consumers) from accessing SEBI through the API Provider. The API Provider will retrieve documents that are relevant to the user's cached queries (keywords). If no relevant documents are obtained, the API Provider immediately writes the new keywords to the New Keyword list and informs the user that there are no documents relevant to the Query. This is implemented to speed up the arrival of information to the user. Similarity Calculator or Query Processor only processes new keyword lists, not all Queries from users. The result of calculating the similarity is not immediately submitted to the user but must be stored in the cache (as in SEBI 2017). The results in this cache will be accessed by the API Provider to handle new Queries from users (via the Consumer API).

3. Result and Discussion

This research produced several software products, as shown in Figure 2, namely:

1. SEBI: Indonesian search engines that have experienced slight changes, especially in handling user queries.
2. RESTful API Provider to handle Queries sent by the Consumer API. This API is integrated with SEBI. The combination of these two products can be called SEBI 2018.
3. RESTful API Consumer to access the RESTful API Provider. The results in the form of relevant documents can be further processed on this client side.

Figure 3 shows an example of access to the REST API Provider from the web browser for the keyword "English league". In addition to the Provider side, REST API has a client or consumer side. Users can create consumers with a specific programming language, library or framework. Figure 4 shows the results of the execution of accessing the API Provider and requesting 5 documents with the highest relevance for the keyword "English league". While
Figure 5 shows the results obtained when providing the same keywords in the 2017 SEBI interface. Figure 4 and Figure 5 clearly show that SEBI access through the API provides results faster than direct access to the SEBI interface.

After seeing changes in the SEBI Architecture and integrated with the RESTful API that was built using the Slim framework, the results show that the SEBI architecture that is integrated with the RESTful API: (1) simpler than the previous SEBI architecture, (2) easier to understand, (3) able to provide short search results to users, (4) allows search results to be accessed from and displayed on a variety of computer devices, (5) can be accessed with a programming language that is mastered, (6) more worthy of release than the previous SEBI, and (7) there are 2 application side, namely: Backend and Frontend.

Figure 3. Results (JSON format) returned by the RESTful API Provider for the keyword "English league"

```
{...
  "title": "Tadbirkan Tim Terbaik di Inggris, Chelsea pun Balik ke Jalur Kemenangan"
  "timestamp": "2018-12-09 15:20:00"
}
```

Figure 4. REST API Consumer Execution made with PHP

```
Waktu proses: 0.00186112258911133
2018-12-09 15:20:00 Tadbirkan Tim Terbaik di Inggris, Chelsea pun Balik ke Jalur Kemenangan
2018-12-09 18:00:00 PSSMS Medan, Sriwijaya FC, dan Mitra Kukar Tendang Kepi...n
2018-12-06 18:10:00 Proliga 2019 Start di Yogya, Panitia Sedangkan 6.000 Tiket Percantik
2018-12-09 23:08:00 Babi Budi Shabra Gamar & Persija
2018-12-10 04:46:00 Hasil Liga Italia, AC Milan Diredam Torino Tanpa Gol
```
Figure 5. Results for the "English league" keyword given by SEBI 2017

4. Conclusion

The SEBI API Architecture (2018) is an architectural renewal of the previous architecture, SEBI (2017). The SEBI API architecture (2018) has been equipped with RESTful API (provider) and REST API client (consumer) functions. SEBI access through the API provides results faster than direct access to the SEBI interface. In addition, the SEBI API (2018) allows search results to be accessed from and displayed on various computer devices.

References

[1] Anonim. (2013). ACM/IEEE Computing Curricula [Online]. Available: https://www.acm.org/education/CS2013-final-report.pdf.
[2] APTIKOM, "Draft KKNI bidang Ilmu Komputer (Informatika)," 2015.
[3] Achmad Jauhari, S. S. Putro, and E. Rahmatani, "Laporan Pelaksanaan Program Hibah Inovasi Pembelajaran Berbasis Program Studi," 2017.
[4] Husni. (2017). Handout Temu-Kembali Informasi [Online]. Available: husni.trunojoyo.ac.id.
[5] V. Devanathan and S. Sundaramurthy, "A Novel Method and Environment For Scalable Web Services Orchestration," in IEEE World Congress on Services Computing, 2016, pp. 128-129.
[6] Y. Li, F. Zhu, P. Li, and R. Wang, "Design of Higher Education Quality Monitoring and Evaluation Platform Based on Big Data," presented at the The 12th International Conference on Computer Science & Education (ICCSE 2017), University of Houston, USA, 2017.
[7] Y. Liu, X. Liu, Y. M. Y. Liu, Z. Zheng, G. Huang, and M. B. Blake, "Characterizing RESTFul Web Services Usage On Smartphones: A Tale of Native Apps and Web Apps," in IEEE International Conference on Web Services, 2015.

[8] D. M. Rathod, D. M. S. Dahiya, and D. S. y. M. Parikh, "Towards Composition of RESTful Web Services," in IEEE 6th ICCCNT Denton, U.S.A, 2015.

[9] A. MISBAH and A. ETTALBI, "Towards a Standard WSDL Implementation of Multiview Web Services,"

[10] I. Malavolta, "Web-based Hybrid Mobile Apps: State of the Practice and Research Opportunities," in IEEE/ACM International Conference on Mobile Software Engineering and Systems, Austin, TX, USA, 2016, pp. 241-242.

[11] M. Ali, M. E. Joorabchi, and A. Mesbah, "Same App, Different App Stores: A Comparative Study," in IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESof), 2017, pp. 79-90.

[12] N. Serrano, J. Hernantes, and G. Gallardo, "Mobile Web Apps," presented at the IEEE SoftwarE, 2013.