



Research Article

Open Access (CC-BY-SA)

Development of academic information system using webassembly technology

Poetri Lestari Lokapitasari Belluano ^{a,1}; Purnawansyah ^{a,2}; La Saiman ^{a,3}; Benny Leonard Enrico Panggabean ^{a,4,*}

^a Universitas Muslim Indonesia, Urip Sumohardjo Makassar 90231, Indonesia

¹ poetrilestari@umi.ac.id; ² purnawansyah@umi.ac.id; ³ lasaiman.labfik@umi.ac.id; ⁴ blep@unpacti.ac.id

* Corresponding author

Article history: Received March 05, 2021; Revised July 04, 2021; Accepted July 04, 2021; Available online August 07, 2021

Abstract

The Academic Information System (in Indonesian often abbreviated as SIAKAD) is a system developed to manage student data that aims at facilitating online academic administration activities. It aims to provide academic information services in the form of web applications, where teachers can independently create student academic reports to synchronize data into the DAPODIK system, a primary education data system, and develop academic information systems using web assembly according to user experience (UX) and developer experience (DX). The research method consists of field studies and literature related to web assembly, primary education data (DAPODIK), and the Academic Information System (SIAKAD). This information system and database were built using the Convention Over Configuration paradigm. The design phase used a prototyping model to graphically represent the system workflow and used an experimental research approach. Moreover, the study used an integrated modeling language (UML), and a Database Management System using PostgreSQL, and alpha testing for model testing. The Client Application was built using the C# programming language for users to generate student academic reports every semester. Processing data transactions using web assembly took less time than the traditional web, which was less than 300 milliseconds.

Keywords: SIAKAD; Xsia; Webassembly; UX; DX

Introduction

The Academic Information System (SIAKAD) is one of system applications to manage information data for higher education institutions as well as for schools from elementary to high school levels. It is especially important to facilitate the students' academics needs. Many schools in Indonesia have been using SIAKAD, but in South Sulawesi Province, not many schools have started implementing SIAKAD as an information center, for both public and private schools, in managing all academic and administrative data in an integrated manner.

Currently, schools still tend to use the Primary Education Data (DAPODIK) application in reporting the results of students' academic activities. DAPODIK itself is an application created by the Ministry of Education and Culture of the Republic of Indonesia which is used as a data collection system to capture all data related to institutions, school curricula, student data, teacher and employee data, and school infrastructure data. The DAPODIK application is operated by a school operator or a specially appointed DAPODIK operator from administrative staff or teachers. This seems to be an obstacle in itself, because the operators need to carry out the duties and responsibilities that very much cover all aspects of school data both data of student and teacher activities. In addition, all forms of computer-based reporting, both independent and global, are the responsibility of the operator.

When observing operators' complaints from an online media such as beritasampit.co.id (12 March 2020) it was found that they were always required to complete reports every semester within the scheduled time. Their work became more complicated and less effective due to some online technical controls.

The case would be different if every school has SIAKAD which can be accessed directly by each teacher to carry out reporting on the performance and results of their students' academic at any time. Moreover, when the SIAKAD data is in accordance with the DAPODIK reporting format, the reporting work can be implemented and monitored systematically by the school DAPODIK operator so that it can minimize the accumulation of data processing by only one or two operators, and the reporting can be completed in time.

The need for an academic information system is directly related to the performance interests of each education provider from schools. In this case, all data processing activities that have been planned, defined and arranged in such a way, especially in the academic sphere, are required to be reported continuously. Thus, independence is needed in carrying out DAPODIK reporting performed by each teacher.

Reinforced by the roadmap from previous research that provides an overview of the current SIAKAD implementation, it is revealed that the implementation is undergoing a significant development. There are some additions of features according to the needs of business rules, both internal and external systems. The development of certain modules or features needs to be adjusted to the needs and internal policies of the school.

Therefore, it is necessary to develop a web-based data processing system application that is able to work automatically and consistently to carry out academic reporting as required by DAPODIK. The application of a web application with a good user interface design and speed of data transactions according to the needs of each school device, especially teaching teachers, are also important concerns so that independent reporting performance can run easily.

The use of web applications is getting wider and the computing needs are getting bigger and they demand a better performance. This need can be supported by the construction of xSIA, namely the application of an Academic Information System Using Webassembly Technology.

Currently, web applications are required to be able to do things that are getting bigger in terms of computing, such as image rendering or image processing. Quoted from the WebAssembly.org page, WebAssembly (Wasm) is an instruction / code in binary form for stack-based virtual machines (applications) designed for compiling high-level languages such as C # so that they are good to build client-server web.

Web Assembly technology can be run at near native speed so that it can be efficient and fast in accessing data. It is open and easy to debug which can be converted into easy-to-read text.

In a previous study, "Development of Academic Information System Interoperability with the Dikti Feeder Using Web Service Technology", it is concluded that the xSIA Academic Information System sought to realize academic system interoperability to carry out data/information exchange through a web service protocol that is mutually agreed upon through various communication channels with various systems that work together and interact with other applications, from the Study Program level to the University PDPT which is then forwarded to the PDDIKTI Feeder.[1] The same researcher who discussed "Development of Single Page Application in Academic Information Systems" concluded that Single Page Application (SPA) is a web application architecture concept where the goal is to make the workings of a web work like a desktop application and have efficient data usage and are more interactive for users to use. .[2]

In the research entitled "Weakening WebAssembly", it is stated that WebAssembly (Wasm) is a portable virtual instruction device that can be used safely in various environments, such as Web browsers. It is a low-level language whose instructions are to compile hardware directly, focusing on single-threaded computing, with the capability of atomic instructions for synchronized access to the principle of Shared Memory. Wasm supports correct compilation of various applications, and provides precise specification of the memory model.[3]

Based on related research references, it has been explained that the development of a web-based academic information system is important to implement WebAssembly technology that can maintain system security when interaction with other devices is required flexibly, using a more careful design, as well as a simple and easy-to-develop interface.

The xSIA Academic Information System with webassembly technology provides academic information system services in the form of a web application where the teacher is an actor who includes reporting on student academic results in the form of assessments. The assessment is in mark format for data synchronization needs in DAPODIK. Also, the xSIA web application is not only for the needs of an academic reporting system at the Elementary Education level, but can also be used by secondary schools and universities, based on the User Experience (UX) and Developer Experience (DX) specifications.

Method

A. Academic Information System

An information system can be defined as a collection of interrelated elements that form a single unit to integrate data, process and store, and distribute information. Academic Information System is software that is used to present information and organize administration related to academic activities. With the use of software like this, it is hoped that academic administrative activities can be managed properly and the required information can be obtained easily and quickly.[4]

B. Basic Education Data

Basic Education Data Information System (Dapodik) is an information system that handles school data such as education staff, students, periodicals, sanitation, study groups and school infrastructure data. The DAPODIK system will later determine the policy of the central government in terms of the ratio of education personnel, the requirements for determining the professional allowance for educators (TFG), School Operational Costs (BOS) and many other central policies. Therefore, the educational unit must input data and update data according to the existing physical evidence into the Dapodik application to get accurate, fast, complete, valid and up to date data. School Operators play an important role in collecting data as well as users of the Dapodik application.[5]

C. Web Application

A web application is an application that uses browser technology to run applications and is accessed via a computer network. A web application is a program that is stored on a server and sent over the internet and accessed through a browser interface. A web application is an application that is accessed using a web browser via the internet or intranet. A web application is also a computer software that is coded in a programming language that supports web-based software.[6]

It is argued that Web Applications are usually built using a combination of programming languages for a variety of different purposes and benefits, namely managing data and information in the internet network. The web application has very light acceleration and minimal resource usage. **Figure 1** is a detailed form of a web application in the implementation of desktop web and mobile web:

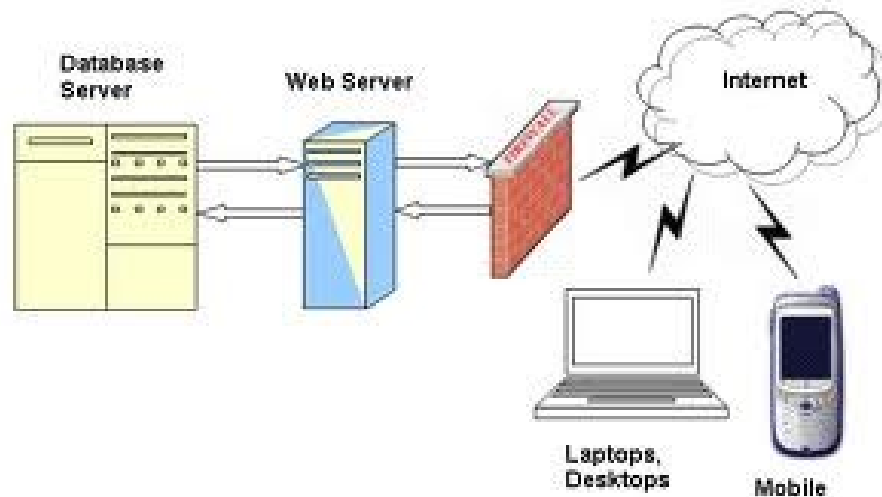


Figure 1. Web application detail

D. Webassembly

Webassembly (Wasm) is to make the new bytecode (set of instructions) read faster for the browser to execute in a high-level language. Therefore, it becomes more efficient both when implemented on desktop and mobile browsers. The entire code is read separately from the web application page. Wasm was developed by 4 well-known browsers, namely: Chrome, Edge, Firefox, and WebKit.[7]

Browsers currently use JavaScript to run code and enable functions on websites, such as forms and dynamic content. However, progress has been made with better execution time performance through asm.js, but the use of bytecode-based systems such as .NET will support faster execution time performance. Wasm is a low-level language whose instructions are used to compile hardware directly, focusing on single-threaded computing, with the capability of atomic instructions for synchronized access on the principle of Shared Memory.[3]

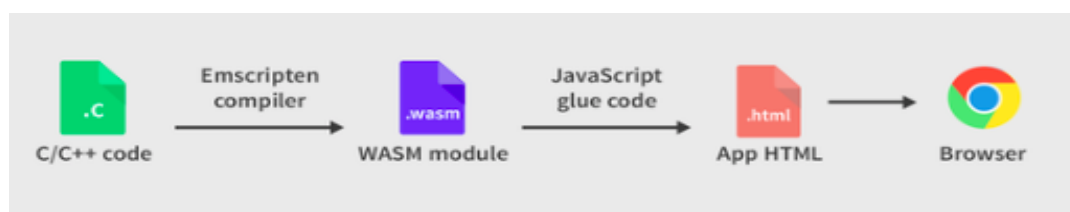


Figure 2. Wasm architecture

Figure 2. shows the binary code compiled or converted into a WASM file and downloaded by javascript to display web pages through the browser engine. So far, Web Browsers use JavaScript to run code and activate functions on websites, but they often have problems. For example, websites can run well in the Mozilla Firefox browser but may not work in other browsers. JavaScript has the potential to cause rendering blocks which can cause websites to take longer to load. Rendering block is a delay condition where all files including JavaScript must load perfectly, then display. At the user level, the security is not guaranteed because the program runs in a separate sandboxed environment from the main runtime. Meanwhile, at the developer level, the lack of security is in the bug memory safety.

E. Programming Language

Programming language is a formal language consisting of a set of instructions to produce various outputs by implementing an algorithmic structure.

Most programming languages consist of computer instructions, where a specific set of instructions is also used by programming languages in general. In general, programming languages are divided into two components, namely syntax (form) and semantics (meaning). Some languages are defined by specification documents such as C language defined by ISO while other languages such as Perl have an implementation domain used as a reference.[8]

The language used to instruct computers is called a programming language. There are 2 types of programming languages consisting of high-level languages & low-level languages. Most programming languages are classified as high-level languages, including C, which uses instructions derived from human words such as begin, end, if, for, while, and, or. And Machine Language (Assembly) is a Low-Level Language that provides instructions to the computer using a language in the form of binary code, for example 011001100101.[9]

F. C Sharp (C#) Mono Framework

In a previous study, it is stated "C# is a modern object-oriented, general-purpose programming language, created and developed by Microsoft together with the .NET platform. There is highly diverse software developed with C# and on the .NET platform: office applications, web applications, websites, desktop applications, mobile applications, games and many others." C Sharp language (C#) is a modern object-oriented programming language created and developed by Microsoft in conjunction with the .NET platform.[10]

C# language is a strong combination of C++ and Java, so it seems to have similarities between C# with C++ and Java, as explained by Miles in his book about the similarity of C Sharp programming language "C# bears a strong resemblance to the C++ and Java programming languages, having borrowed (or improved) features provided by these languages." While C# Mono-framework is a code technique that is applied to develop open source-based applications.[11]

G. User Experience (UX)

According to the definition of ISO 9241-210, user experience is a person's perception and response to the use of a product, system, or service. According to Mendiola, User Experience (UX) assesses how satisfied and comfortable a person is with a product, system, and service. A principle in building UX is that the audience has the power to determine their own level of satisfaction (customer rule). No matter how good the features of a product, system, or service, if the intended audience are not being able to feel satisfaction, rules, and comfort in interacting, the UX level will be low. The development of the digital and mobile world makes UX more complicated and multidimensional.[12]

H. Developer Experience (DX)

Developer Experience (DX) is the same as User Experience but the main user of the application is the development team (Developers) who are very concerned about the application of application products, documentation, frameworks, open-source solutions, general tools, APIs, and so on in connection with application development. DX will provide technical recommendations to support application performance starting from the application of technology, functions, to the user interface.

I. Microsoft User Interface

According to Karpen, the interface is the part of the computer system that is in direct contact with the user. There are two types of interfaces, namely Command Line Interface (CLI) and Graphical User Interface (GUI). Command Line Interface (CLI) is a type of interface where the user interacts with the computer system via a text-terminal. Microsoft User Interface uses a GUI-based interface (Graphic User Interface) where users are presented with icons, buttons, forms, lists. GUI is in great demand by computer users because of its easiness, accessibility, and attractive appearance. The GUI interface is in the form of a graphical mode with a more attractive appearance and does not seem monotonous so that it can make users more comfortable and tend to be able to change the user's feelings psychologically.[13]

J. System Development Stages

The research stages using the Software Development Life Cycle (SDLC) model that will be carried out show in **Figure 3**, including:

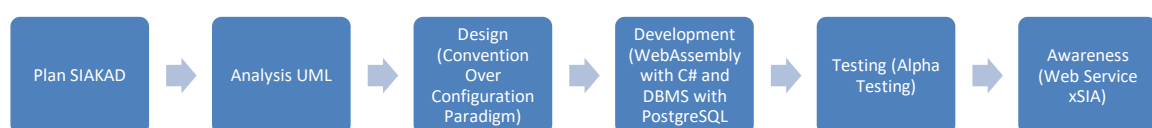


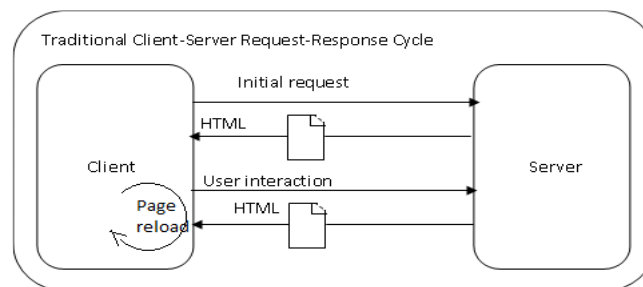
Figure 3. Stages of SDLC model development

1. Planning, developing academic information systems in schools to universities that require webassembly technology to facilitate student academic reporting data transactions.
2. Needs Analysis, modeled using Unified Modeling Language (UML) diagram tools to determine the work flow of the system being built.
3. System Design, applying the Convention Over Configuration paradigm with the aim of specifying function modules that are created using the framework while maintaining the flexibility of system performance.
4. Development, using webassembly technology built with C# tools and PostgreSQL DBMS.
5. System testing, using the Alpha Testing technique carried out by researchers to test the speed performance of the system in data transactions both on the user client side and on the server side.

K. Objects and Model Development

This research was conducted at SDN No. 133 Inpres, Takalar district, and at the National Tri Tunas Institute of Technology and Health. In this study, the observed variables were adjusted to the user's needs, like students' academic data. The population of academic data used as a sample is academic transaction data for the 2019 period.

The model used in the system development is the Prototyping Model which is a graphical representation of the workflow of the DAPODIK system that describes the content and flow of information (data and control) both functionally and behaviorally. It describes the essence of what should be built on the xSIA academic information system using Webassembly technology. The experimental research design for the implementation of web apps with WebAssembly technology, is based on the manipulation of independent variables including dynamic information presentation so that users can make changes to information data at any time, and measure the effect on the dependent variable.



Gambar 4. Traditional Client-Server Request-Response Cycle

Figure 4. Shows the application of Webassembly technology using a traditional model for the data communication process between client – server where the client makes a request to open a site or application and the server responds via an HTML page.

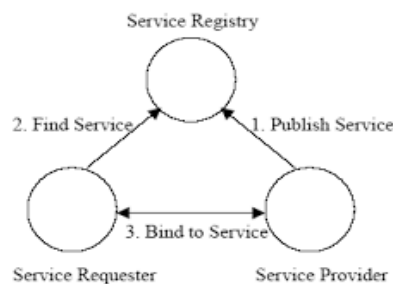


Figure 5. Webservice Architectural Design

Figure 5. The webservice architecture is used in the system design that shows the form of Request and Response transaction communication. The sequences in which specific information must be specified in a particular order. As content and processing becomes more complex, alternative content can be submitted as a diversion to get the content to appear more complete. WebApp content can be organized categorically in two (or more) dimensions, either vertically or horizontally.

Results and Discussion

A. Requirement Specification

Technically, the results of this research were implemented with several hardware and software specifications, including:

1. Server hardware in the form of a 1.8 GHz Processor, 8 GB Memory
2. Server Software in the form of Operating System macOS Mojave 10.14.3, RDBMS PostgreSQL 13, and HTTP server Nginx 1.19.1
3. C# programming language
4. Client software, i.e., Browser Latest Chrome 88

The data variables used included master data:

1. Lecturer
2. Student
3. Institution
4. Authentication
5. Courses
6. Lecturing activities

They were used in the xSIA WebAssembly simulation for data management of Student Academic Values. While the prototype interface designed in localhost consisted of:

1. webassembly.test for client side
2. xsia_v6.test for server side

In the system there were many actors, one of which was the teacher user. One of the tasks of the teacher was to enter the value of the student's academic results. Teachers were given special access rights to be able to enter the final assessment of all subjects taught through the webassembly.test page.

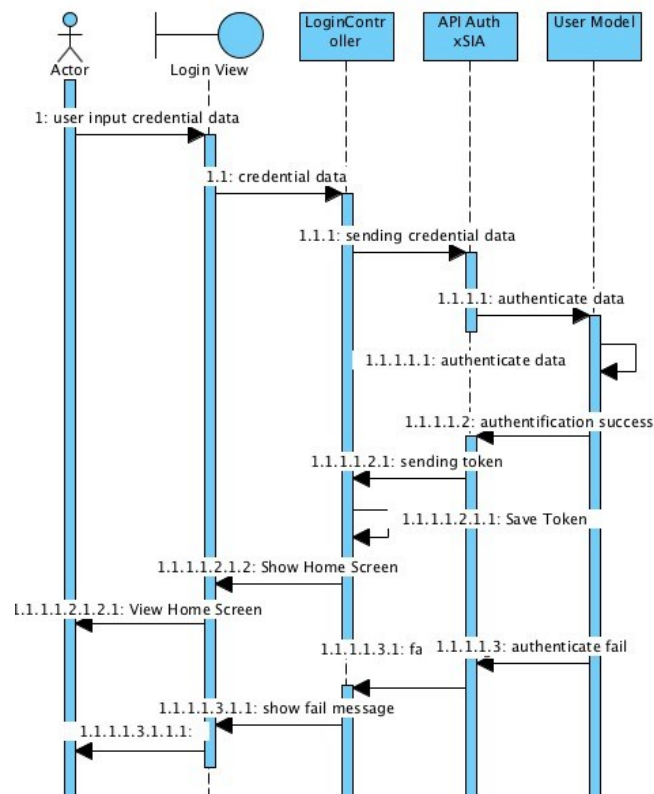


Figure 6. User Login Activity on WebAssembly

Authentication used the OAuth 2 method, where authentication was the industry standard for authorization in web-based application development. **Figure 6.** Shows the user entered login data sent via the http protocol. After the login data was recognized by the server, the server returned a success notification message and a token. Token served as an identifier to identify a user when conducting data transactions.

m. User Interface Implementation

The user interface was built using 3 (three) programming languages, namely HTML 5, CSS 3, and C# as a substitute for Javascript which is generally used to build web apps. The choice of C# as one of the programming languages for Web assembly development was for library completion to build an application. Webassembly is a technology that is prepared as a next-generation replacement for javascript which has a large size at each render time in the browser. Webassembly has a small file size and is quickly managed and executed by the browser. One of the

distinguishing features between web applications using Webassembly technology and traditional web applications is that there are files created by compiling the C# language into binary files which usually have the extension. wasm as shown in **Figure 7**. Whereas for traditional webs, the extension is usually .js.

dotnet.5.0.1.js	26 November 2020 02.04	242 KB	JavaScript
dotnet.5.0.1.js.br	Today 11.52	55 KB	Document
dotnet.5.0.1.js.gz	Today 11.54	63 KB	gzip co...archive
dotnet.timezones.blst	26 November 2020 02.04	654 KB	Document
dotnet.timezones.blst.br	Today 11.52	67 KB	Document
dotnet.timezones.blst.gz	Today 11.54	145 KB	gzip co...archive
dotnet.wasm	26 November 2020 02.04	2,8 MB	Document
dotnet.wasm.br	Today 11.53	883 KB	Document
dotnet.wasm.gz	Today 11.54	1,1 MB	gzip co...archive

Figure 7. Hasil Compile C# menjadi webassembly

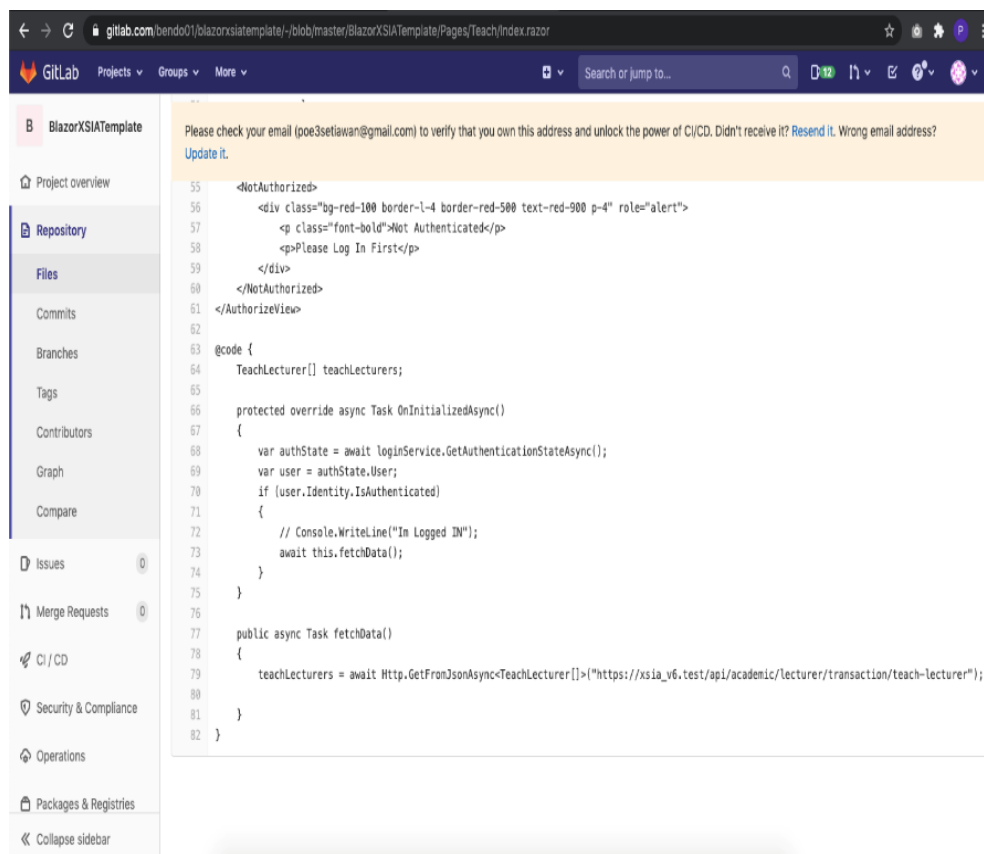


Figure 8. Code Webassembly

Figure 8 is a form of implementation of the C # programming language which was compiled into a Webassembly, so that it could be managed and used in the browser. As a simulation of the use of Webassembly on the xsia system, it is shown when the user (teacher) opened the start page of the application, which was likely to be accessed instantly and on a scheduled basis by other users (teacher). As in **Figure 8** the entry of student academic results was accessed simultaneously by all users (Teachers).

The user entered value data in the form of numbers. The data sent to the server was combined with a token as a differentiator between a user and other users. If the transaction was successful, the server would send a successful message or vice versa.

Building an xSIA academic information system for reporting student academic scores where teachers could carry out student academic reporting independently for DAPODIK data synchronization. The teacher entered grades on a scheduled basis, where data transactions were a learning activity for a number of students that must be filled in which required fast system performance. It could be seen from the average amount of execution time in the value entry process, the data sending process (request) was 163ms for 243kb of data. And the process of data reply (response) from the server was 55ms.

By implementing webassembly technology to evaluate the performance of data transaction speed on the xSIA academic information system, it could be observed that the difference in transaction time speed between the two same activities was also applied to traditional web applications in **Figure 9** and webassembly in **Figure 10**.

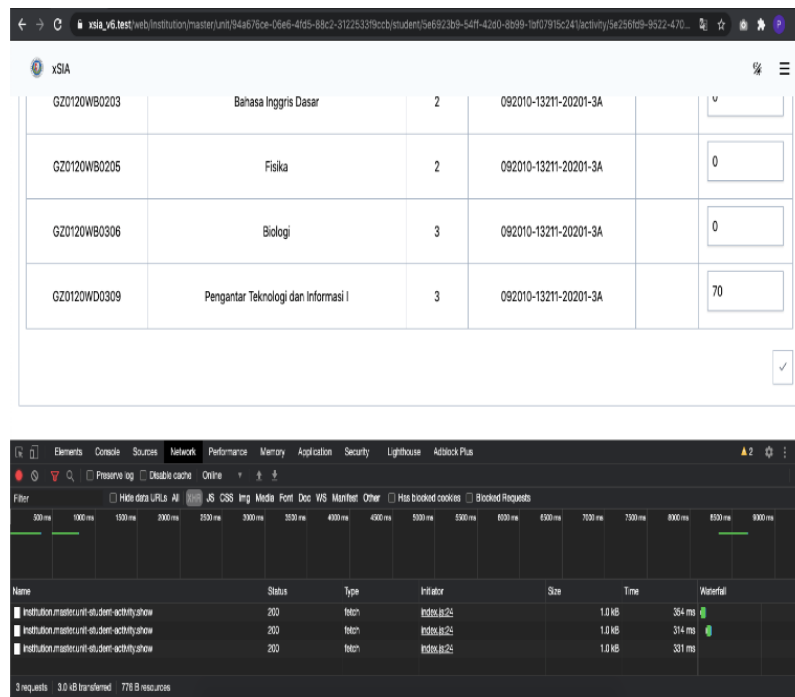


Figure 9. Traditional Web Access Speed

On the traditional web, a rendering process is needed, which is re-drawing all the components of web elements starting from the syntax and data in the browser, thus requiring additional processing time. As can be seen in **Figure 8**, there were 3 (three) request activities with a status code of 200 on the http protocol, which means that the request was successfully processed by the server, but with an average speed of 300ms.

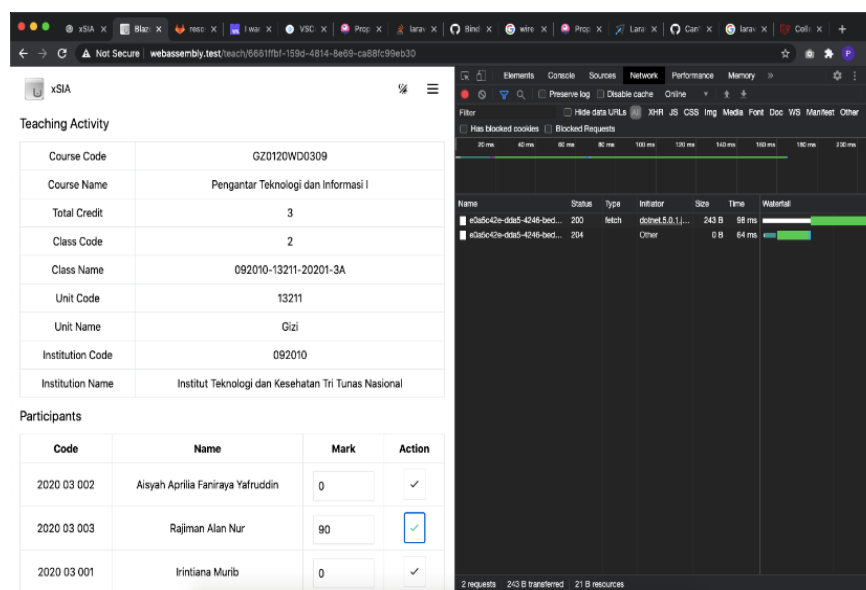


Figure 10. Webassembly

Meanwhile, the use of webassembly technology with the same activity simulation for 3 (three) requests showed a status code of 204 which means the request was successfully processed without the browser needing to re-render, which was indicated by the content size of 0kb. All request and response activities were less than 300ms. With a fast response time, users did not need to wait longer, resulting in a good User Experience (UX). From the Developer Experience (DX) point of view, the use of concrete syntax could make it easier for application developers to use a more descriptive and easy-to-understand syntax.

Conclusion

From the results of research in respect to the development of the xSIA academic information system using webassembly technology, it can be concluded that the development of academic information system services in the form of a web application supported the users (teachers) to carry out students' academic score reports independently for synchronizing DAPODIK data. Meanwhile, the use of WebAssembly technology for web-based application development met the specification criteria in terms of User Experience (UX) and Developer Experience (DX).

Acknowledgement

Thank you to the members of the Indonesian Muslim University, Tri Tunas National ITEKES, and SDN Inpres No.133 Takalar South Sulawesi, who supported all the research activities.

References

- [1] P. L. L. Belluano, "Pengembangan Interoperabilitas Sistem Informasi Akademik Dengan Feeder Dikti Menggunakan Teknologi Web Service," *Proceedings SNRIK Vol.1 Desember 2016*, ISSN : 2443-048X, vol. 1, 2016, [Online]. Available: <https://docplayer.info/78758463-Pengembangan-single-page-application-pada-sistem-informasi-akademik.html>.
- [2] P. L. Lokapitasari Belluano, "Pengembangan Single Page Application Pada Sistem Informasi Akademik," *ILKOM Jurnal Ilmiah*, vol. 10, no. 1, pp. 38–43, 2018, doi: 10.33096/ilkom.v10i1.204.38-43.
- [3] C. Watt, A. Rossberg, and J. Pichon-Pharabod, "Weakening WebAssembly," *Proceedings of the ACM on Programming Languages*, vol. 3, no. OOPSLA, pp. 1–28, 2019, doi: 10.1145/3360559.
- [4] Satoto and Kodrat Iman, "Analisis Keamanan Sistem Informasi Akademik Berbasis Web Di Fakultas Teknik Universitas Diponegoro," *Artikel Ilmiah Terpublikasi Universitas Diponegoro*, 2008.
- [5] N. Wahyuni, F. Mardiana Nugraha, and S. Borneo Internasional Balikpapan Batu Ampar, "Analisis Perbandingan Sistem Data Pokok Pendidikan (Dapodik) Tingkat Smk," *Jurnal Sistem Informasi*, vol. 1, no. April, pp. 2597–3827, 2018.
- [6] M. Arifin, "Aplikasi Web Dengan Simulasi Kredit Menggunakan Codeigniter Framework Pada Toko Langgeng Elektronik," 2011, [Online]. Available: <https://media.neliti.com/media/publications/207585-none.pdf>. JADI NO.6
- [7] W. N. K. Laurensius Adi, Rizky Januar Akbar, "Platform E-Learning untuk Pembelajaran Pemrograman Web Menggunakan Konsep Progressive Web Apps," *JURNAL TEKNIK ITS*, vol. Vol. 6, No, pp. 2337-3520 (2301–928X Print), 2017, [Online]. Available: <https://core.ac.uk/download/pdf/289794241.pdf>.
- [8] T. Koetsier, "On the prehistory of programmable machines: Musical automata, looms, calculators," *Mechanism and Machine Theory*, vol. 36, no. 5, pp. 589–603, 2001, doi: 10.1016/S0094-114X(01)00005-2.
- [9] Yusuf Wahyudi and R. Berlian, "Instruksi Bahasa Pemrograman ADT (Abstract Data Type) Pada Virus dan Loop Batch," *Jurnal Media Infotama*, vol. 9, no. 3–2, 2013.
- [10] Svetlin Nakov, *Fundamentals of Computer Programming with C# (English, 2013)*. 2013.
- [11] R. Miles, *C# Programming Yellow Book*. Cheese Edition. 2016.
- [12] M. B. Wiryawan, "User Experience (Ux) sebagai Bagian dari Pemikiran Desain dalam Pendidikan Tinggi Desain Komunikasi Visual," *Humaniora*, vol. 2, no. 2, p. 1158, 2011, doi: 10.21512/humaniora.v2i2.3166.
- [13] Karpen, "Antarmuka Sebagai Media Komunikasi Dengan Sistem," *Jurnal Sains dan Teknologi Informasi*, vol. Vol 1, 2012.
- [14] Rizka Liatmaja, "Sistem Informasi Akademik Pada Lembaga Bimbingan Belajar Be Excellent Pacita", *Indonesian Jurnal on Networking and Security (IJNS)*, Vol 2 No.2, 2013
- [15] Akbarullah, M.Suyanto, Sukoco, "Evaluasi User Experience pada Sistem Informasi Akademik Di STKIP PGRI Pacitan", *Jurnal Penelitian Pendidikan*, Vol 11 No.11, 2019