

Implementation of PDDIKTI Neo Feeder Web Service in Recording of Independent Campus Activities

Herlambang Brawijaya^{-1*}, Slamet Widodo⁻², Samudi⁻³

Sistem Informasi
Universitas Bina Sarana Informatika
Jakarta, Indonesia
<http://www.bsi.ac.id>

herlambang.hba@bsi.ac.id⁻¹, slamet.smd@bsi.ac.id⁻²

Sistem Informasi
Universitas Nusa Mandiri
Jakarta, Indonesia
<http://www.nusamandiri.ac.id>
samudi.smx@nusamandiri.ac.id⁻³

(*) Corresponding Author

Abstract

Independent Learning-Independent Campus Program (MBKM) is a policy of granting the right for students to be able to take study activities outside the study program as many as three semesters with the division of two semesters of study outside the college and one semester in different study programs in one college. As well as teaching and learning activities, universities must report Independent Learning-Independent Campus activities to DIKTI every semester through the Neo Feeder PDDIKTI application. The Neo Feeder PDDIKTI application has a feature to enter the activities the operator will carry out. The operator enters this data individually on the Neo Feeder PDDIKTI application. This is a significant problem because the data entry process will take quite a lot of time, even though PDDIKTI has provided web service access to universities to optimize the data reporting process using the Neo Feeder PDDIKTI application. Building an application that can be used for recording MBKM activities by utilizing web services provided by PDDIKTI is the primary purpose of this study. The development of an application certainly requires a method as a framework or guide in facilitating the manufacturing process. The extreme Programming (XP) method becomes essential for applications with variable or non-fixed needs. This method has four working elements: planning, designing, coding, testing and software increment. The output generated by this study is an application for recording MBKM activities that use web service restful API technology so that data entry can be done en masse and not one by one.

Keywords: Extreme Programming; Independent Learning Independent Campus; Web Service; Information System Design, Feeder PDDIKTI

Abstrak

Program Merdeka Belajar Kampus Merdeka (MBKM) merupakan kebijakan pemberian hak bagi mahasiswa untuk dapat menempuh kegiatan belajar di luar program studi sebanyak tiga semester dengan pembagian dua semester belajar di luar perguruan tinggi dan satu semester di program studi yang berbeda dalam satu perguruan tinggi. Seperti halnya kegiatan belajar mengajar, kegiatan merdeka belajar kampus merdeka ini wajib dilaporkan oleh perguruan tinggi kepada DIKTI setiap semester melalui aplikasi Neo Feeder PDDIKTI. Aplikasi Neo Feeder PDDIKTI ini memiliki fitur untuk memasukan aktifitas kegiatan ini yang nantinya dilakukan oleh operator PDDIKTI perguruan tinggi. Operator PDDIKTI perguruan tinggi memasukan data aktifitas ini secara satu per satu pada aplikasi Neo Feeder PDDIKTI. Hal ini menjadi permasalahan utama karena proses pemasukan data ini akan menghabiskan waktu yang cukup banyak padahal PDDIKTI sudah memberikan akses web service kepada perguruan tinggi untuk dapat digunakan dalam mengoptimalkan proses pelaporan data menggunakan aplikasi Neo Feeder PDDIKTI. Membangun sebuah aplikasi yang dapat digunakan untuk pencatatan kegiatan aktifitas MBKM dengan memanfaatkan web service yang diberikan oleh PDDIKTI merupakan tujuan utama dalam penelitian ini. Pembangunan sebuah aplikasi tentunya membutuhkan metode sebagai kerangka kerja atau panduan dalam memudahkan proses pembuatannya.

Metode Extreme Programming (XP) menjadi unsur penting sebagai metode yang digunakan untuk aplikasi yang memiliki kebutuhan berubah-ubah atau tidak tetap. Metode ini memiliki empat unsur kerja yaitu perencanaan, perancangan, pengkodean, pengujian dan peningkatan perangkat lunak. Keluaran yang dihasilkan oleh penelitian ini berupa aplikasi pencatatan aktifitas kegiatan MBKM yang menggunakan teknologi web service restful API sehingga dapat dilakukan pemasukan data secara masal dan tidak satu per satu.

Kata kunci: Extreme Programming; Merdeka Belajar Kampus Merdeka; Web Service; Perancangan Sistem Informasi; Feeder PDDIKTI

INTRODUCTION

Education has a vital role in forming students who can compete in a society where culture, society, the world of work, and technology are constantly changing and developing (Tohir, 2020). Permendikbud No 3 of 2020 concerning National Standards for Higher Education exists in carrying out the duties and roles of forming students who are ready to fight in the world of work and the community environment (Faiz & Purwati, 2021; Fuadi & Aswita, 2021; Rusli Baharuddin, 2021). By issuing the Merdeka Belajar Kampus Merdeka (MBKM) policy, tertiary institutions must provide flexibility for students to voluntarily choose to study outside the study program with a maximum of three semesters—two semesters studying outside of college and one semester studying in different study programs in one tertiary institution. The maximum number of credits is 20 per semester (Laga, Nona, Langga, & Jamu, 2021; Meke, Astro, & Daud, 2021; Sulistiyan et al., 2021). Students can choose the form of activity in the MBKM program according to their preferred field (Tohir, 2020). There are nine forms of activities in the MBKM program, including student exchanges, internships/work practices, teaching assistance in education units, research/research, humanitarian projects, entrepreneurial activities, independent studies/projects, building villages/thematic actual work courses, and defending the country (Fuadi & Aswita, 2021; Tohir, 2020).

Reporting higher education teaching and learning activities to DIKTI is a state mandate in the RI Minister of National Education Decree Number 184/U/2001 Article 5 (Widodo et al., 2018). As implementers of teaching and learning activities, tertiary institutions must submit reports on the teaching and learning activities process to the Directorate General of Higher Education and the Higher Education Database when the semester ends (Gilvy Langgawan Putra et al., 2021). In making it easier for universities to submit reporting on the teaching and learning activities process, the Directorate General of Higher Education presented

a reporting submission application called Feeder DIKTI which later transformed into Neo Feeder PDDIKTI in February 2022 (Hariyanto & Hernawati, 2020). With the Feeder application, universities are more controlled in submitting their reports because this application provides various features and settings that can be used by universities in the process of entering reporting data. This application also provides several features that universities can use as a reminder (early warning system) and the history of processes that universities have carried out. One of the features related to recording independent campus learning activities is the activity feature of students and supervisors (Handrianus Pranatawijaya, 2020). The data entry process in the feeder application is carried out one by one. The Directorate General of Higher Education provides access to a restful API web service that universities can develop into the information system entering data on the Feeder application (Perwira & Santosa, 2017; Usman, 2020). Having access to the restful API web service makes it easier for universities to develop a better system because one of the advantages of this restful API is that it can exchange data across platforms and has an independent language (Amin, 2018; Sitorus, Sri Wahyuni, & Batubara, 2018; Somya & Michelle Esmeralda Nathanael, 2019). This makes it easier for universities to enter data en masse or in large quantities.

With such a policy issued by the Directorate General of Higher Education regarding submitting reports on teaching and learning activities, there is still a problem: the PDDIKTI feeder application does not provide a feature to import data entering data into the feeder application. Of course, this is homework for universities in inputting data one by one into the PDDIKTI Feeder application. This is also a significant obstacle because the data that universities have to report is not tiny, and it becomes inefficient with the length of the data entry process (Widodo et al., 2018).

The policy regarding data collection on MBKM activities is dynamic, which means the policy

is not fixed or constantly changing. Determining the system development method can refer to how often the system will experience changes based on the system's needs. One of the advantages of the Extreme Programming (XP) method is that it facilitates the development of systems where requirements are unclear or are likely to experience swift changes in requirements (Supriyatna, 2018). The changes to the PDDikti policy will also impact the adjustment of the web service used to synchronize with the PDDikti neo feeder application.

MATERIALS AND METHODS

Extreme Programming (XP)

Agile software development is one of the techniques used in developing fast and dynamic software (Trisnadoli, 2021). The concept of an agile framework is an incremental and iterative approach. One of the methods used in implementing this agile concept is Extreme programming (XP) (Supriyatna & Puspitasari, 2021). This method offers more efficient, adaptive, flexible steps and the focus desired by the developer in a relatively short time (Budiarti & Risyanto, 2020; Carolina & Supriyatna, 2019; Supriyatna, 2018).

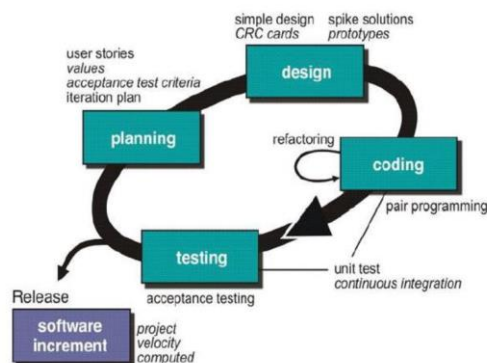


Figure 1. Extreme Programming Framework

The steps taken in designing an application for recording the activities of the Merdeka Campus using the web service on the PDDIKTI Neo Feeder are as follows:

1. **Planning.** This initial stage begins with identifying and analyzing user needs and ensuring users understand business processes and get an overview of recording the activities of the Merdeka Campus.
2. **Design.** Continuing from the planning stage, this stage starts designing the system using the Unified Modeling Language (UML), including use case diagrams and activity diagrams.

3. **Coding.** Implementing the two stages above is creating program code concerning the previous stage. The primary coding at this stage uses the PHP programming language combined with HTML, CSS and JavaScript with the bootstrap display format.
4. **Testing.** This stage is a critical element of Extreme Programming. This stage uses Blackbox Testing to ensure the system can work correctly and as expected.
5. **Software Increment.** This stage is the final stage, where the system has been developed in stages and implemented by adding several features to increase system functionality.

The Extreme Programming (XP) method is the choice in system development because of several advantages compared to other methods. As the costs incurred in the system development process are more affordable because it does not focus on requirements for too long but focuses on coding, does not require a large team in system development, and is flexible in dealing with rapid changes in requirements (Carolina & Supriyatna, 2019; Supriyatna, 2018). Unlike the Agile Modeling method, which focuses on fixed requirements, so it takes effort when changes occur, Extreme Programming (XP) can be immediately repaired, and these changes are tested (Handrianus Pranatawijaya, 2020).

RESULTS AND DISCUSSION

Identification of problems

The problems obtained from the results at the planning stage are as follows:

1. The data entry process is carried out individually through the PDDIKTI Feeder application.
2. The process of entering large amounts of data takes a long time.

Requirement Analysis

Based on the problems at the problem identification stage above, the need for an independent campus activity recording system is obtained as follows:

1. The system has a feature to provide a data format that the user must prepare.
2. The system can perform data parsing through the web service according to the data provided.
3. The system must be able to process data quickly even though the amount of data is up to thousands.

- The system can display the response results from the data parsing, whether it was successful or failed to be sent.
- The system can display the amount of data successfully sent and received by the PDDIKTI Feeder application.

Web Service Interoperability Analysis

Interoperability analysis at the planning stage is critical because this system will interact with data (data sharing) with other systems, such as the PDDIKTI Neo Feeder Application. Web service interoperability of this system can be seen in Figure 2 below.



Figure 2. Interoperability Schema

In Figure 2, this system acts as a client-side, making requests and responses in parsing data to the PDDIKTI Neo Feeder Application. From the above, it can be seen that this system will receive data originating from SIAKAD and map the data imported into this system to be forwarded to the data parsing process.

Web Service Module Analysis

Access to the web service used in this system is obtained from supporting documentation provided by the Director General of Higher Education through the pddikti-admin.kemdikbud.go.id page—module structure to be used, such as Table 1 to Table 6.

- Get Student Activity Type (GetJenisAktivitasMahasiswa)

Table 1. Get Student Activity Type

Activity_type_id	Activity_type_name	Flags_MBKM
13	Internships/work practices	1
14	Teaching Assistance in Education Units	1
15	Research	1
16	Humanitarian Projects	1
17	Entrepreneurial Activities	1
18	Independent Studies/Projects	1
19	Building Thematic Real Work Villages/Lectures	1
20	Defending the State	1

- Get Activity Category (GetKategoriKegiatan)

Table 2. Get Activity Category

Activity_category_id	Activity_category_name
110300	Guiding Real Work Lectures, Field Work Practices, including guiding student training, student exchanges, internships, research-based lectures, entrepreneurship, and other forms of community service and the like

- Insert Student Activities (InsertAktivitasMahasiswa)

Table 3. Insert Student Activities

Method	Type Data	Nullable	Description
Token			
Jenis_anggota	Numeric	Not null	0: Personal, 1: Group
Id_jenis_aktivitas	Numeric	Not null	Activity Type ID. Web Service: GetJenisAktivitasMahasiswa
Id_prodi	Uuid	Not null	ID SMS. Web Service: GetProdi
Id_semester	Character	Not null	ID Semester. Web Service: GetSemester
Judul	Character varying	Not null	
Keterangan	Text		
Lokasi	Character varying		
Sk_tugas	Character varying		
Tanggal_sk_tugas	Character varying		

- Insert Student Activity Members (InsertAnggotaAktivitasMahasiswa)

Table 4. Insert Student Activity Members

Method	Type Data	Nullable	Description
Token			
Id_aktivitas	Uuid	Not null	Web service: GetListAktivitasMahasiswa
Id_registrasi_mahasiswa	Uuid	Not null	Web Service: GetListRiwayatPendidikanMahasiswa
Jenis_pernikahan	Character	Not null	1: Leader, 2: Member, 3: Personal

5. Insert Student Advisor (InsertBimbingMahasiswa)

Table 5. Insert Student Advisor

Method	Type Data	Nullable	Description
Token			
Id_aktivitas	Uuid	Not null	Web service: GetListAktivitasMahasiswa
Id_kategori_kegiatan	Integer	Not null	Web Service: GetKategoriKegiatan
Id_dosen	Uuid	Not null	ID Dosen. Web service: GetListDosen

6. Insert Konversi Kampus Merdeka (InsertKonversiKampusMerdeka)

Table 6. Insert Independent Campus Conversion

Method	Type Data	Nullable	Description
Token			
Id_aktivitas	Uuid	Not null	Web service: GetListAktivitasMahasiswa
Id_matkul	Uuid	Not null	Web Service: GetListMataKuliah
Id_anggota	Uuid	Not null	Student activity member id. Web Service: GetListAktivitasMahasiswa
Sks_mata_kuliah	Numeric(5,0)	Not null	
Nilai_angka	Numeric	Not null	
Nilai_indeks	Numeric	Not null	
Nilai_huruf	Character(3)	Not null	

Design

The design stage describes several diagrams representing the system and system architecture. Based on the analysis stages, several diagrams were decided in this study: Use Case Diagrams, Activity Diagrams, and Entity Relationship Diagrams.

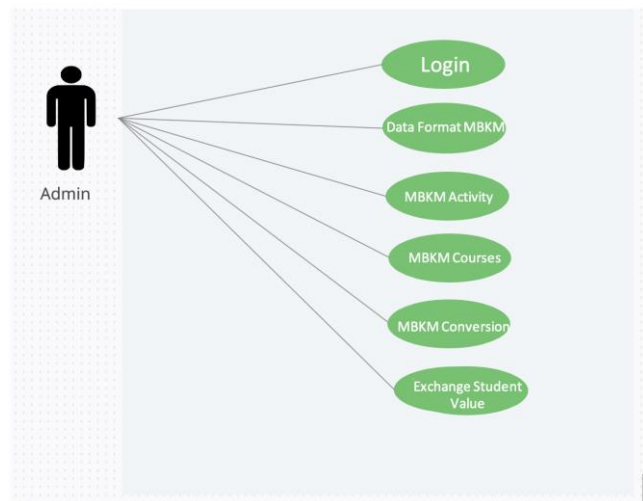


Figure 3. Use Case Diagram Admin

The admin use case diagram in Figure 3 illustrates what actions the admin can take advantage of in recording the activities of MBKM students. To record MBKM activities, the admin must log in to enter the system. After logging in, the system will display several features, which are sequential according to the stages of recording MBKM student activities.

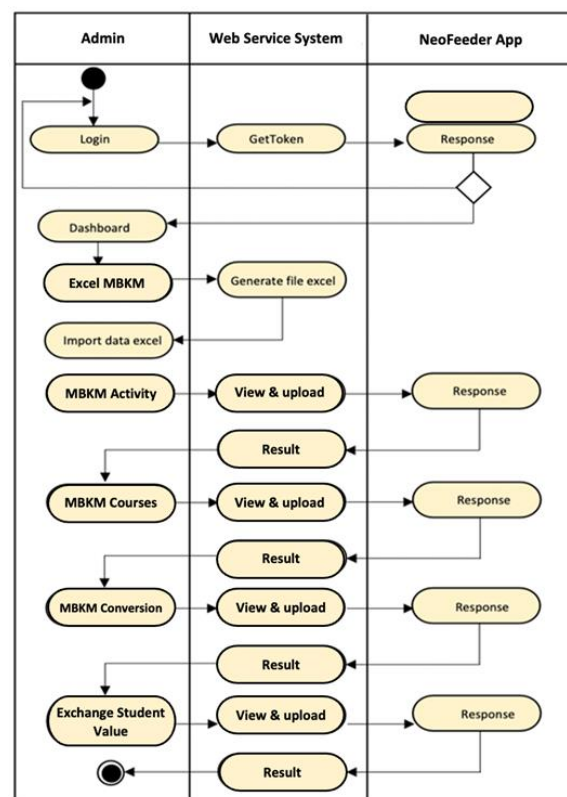


Figure 4. Activity Diagram

In the activity diagram Figure 4. Represents the flow of activities contained in the MBKM student activity recording system, from the beginning of the activity to the process that occurs until the activity ends.

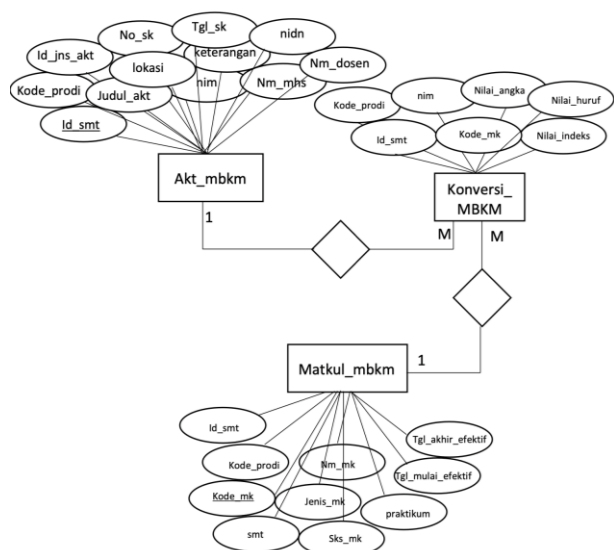


Figure 5. Entity Relationship Diagram

In the entity diagram Figure 5. Represents the data model used in storing data in the database, as for recording the activities of MBKM students using three data tables.

Coding

In the coding stage, the author represents the system using the PHP programming language coupled with HTML and jquery and CSS and uses the MySQL database. The following display is generated at this stage.



Figure 6. Login Page

This page is the initial view that will appear when the program starts to be accessed.

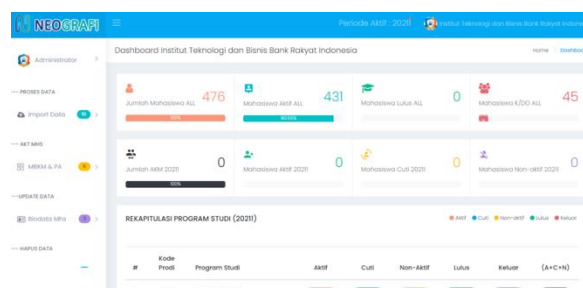


Figure 7. Dashboard Page

In Figure 7, a dashboard page will appear when the user successfully logs in.

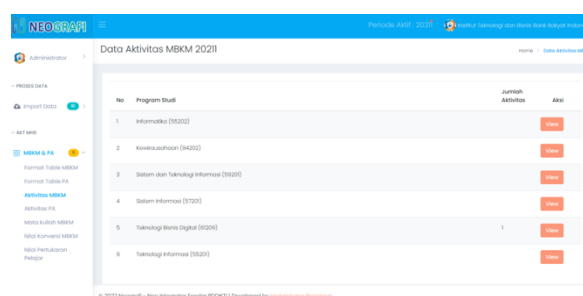


Figure 8. Activity MBKM Menu

Figure 8 is a menu for processing MBKM activity data into the neo feeder application via a web service.

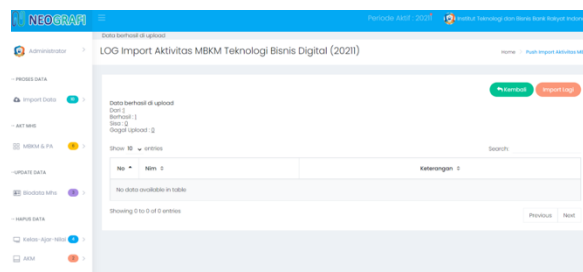


Figure 9. Activity MBKM Log Process

Figure 9 shows the log of the MBKM activity data import process. The number of successful and failed messages sent to the web service is listed.

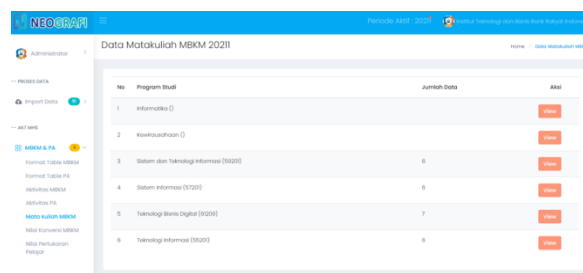


Figure 10. MBKM Subject Menu

Figure 10 is a menu for processing MBKM subject data into the neo-feeder application via a web service.

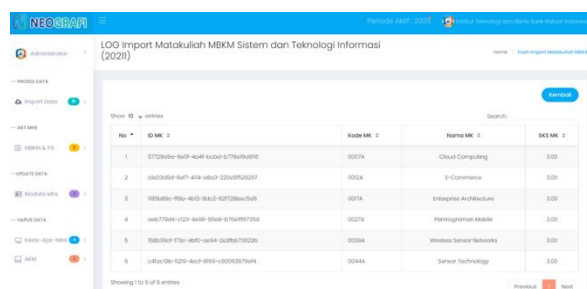


Figure 11. Subject MBKM Log Process

Figure 11 displays the successful and failed MBKM data import process logs in sending data.

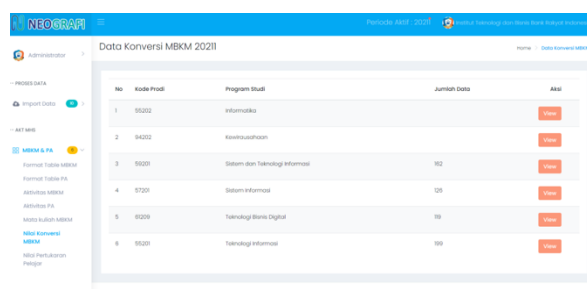


Figure 12. MBKM Konversion Value Menu

Figure 12 is a menu for processing MBKM value conversion data into the neo-feeder application via a web service.

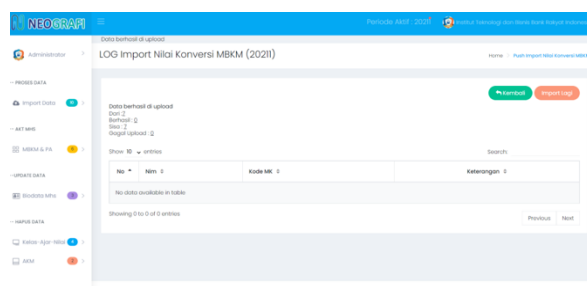


Figure 13. MBKM Konversion Value Log Process

Figure 13 shows the log on importing the MBKM value conversion data. The number of successful and failed messages sent to the web service is listed.

Testing

The testing phase is carried out using black box testing by testing the input and output processed on the system.

Table 7. Login Pagi Testing

scenario	Test case	expectation	Testing result	conclusion
Choose an institution that is not suitable for the neo-feeder application	Pts : universitas nusa mandiri	The system will refuse	According to expectations	valid
Choose the appropriate institution with the neo feeder	Pts: institute BRI	The login is booming, and the dashboard page appears	According to expectations	valid

CONCLUSIONS AND SUGGESTIONS

Conclusion

Based on the research and design results conducted in building an independent campus activity recording system using this web service. The application of web services in this system makes the data entry process for independent campus activities efficient and effective because data entry does not need to be done one by one but can be done massively. In subsequent research, it can be developed by creating an integrated system with higher education SIAKAD to make the process more seamless.

REFERENCES

- Amin, M. M. (2018). Interoperabilitas Perangkat Lunak Menggunakan Restful Web Service. *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, 4(1), 14–22. <https://doi.org/10.26594/register.v4i1.1129>
- Budiarti, Y., & Risyanto, R. (2020). Implementasi Metode Extreme Programming Untuk Merancang Sistem Informasi Pendaftaran Siswa Baru Berbasis Web Pada SMK Multimedia Mandiri Jakarta. *Informatika: Jurnal Ilmiah Fakultas Sains Dan Teknologi*, 8(1), 2–9. Retrieved from www.bps.go.id
- Carolina, I., & Supriyatna, A. (2019). Penerapan Metode Extreme Programming Dalam Perancangan Aplikasi Perhitungan Kuota SKS Mengajar Dosen. *Jurnal IKRA-ITH Informatika*, 3(1), 106–113.
- Faiz, A., & Purwati, P. (2021). Koherensi Program Pertukaran Pelajar Kurikulum Merdeka

- Belajar Kampus Merdeka dan General Education. *Edukatif: Jurnal Ilmu Pendidikan*, 3(3), 649–655. <https://doi.org/10.31004/edukatif.v3i3.378>
- Fuadi, T. M., & Aswita, D. (2021). Merdeka Belajar Kampus Merdeka (MBKM): Bagaimana Penerapan Dan Kendala Yang Dihadapi Oleh Perguruan Tinggi Swasta Di Aceh. *Jurnal Dedikasi Pendidikan*, 5(2), 603–614. Retrieved from <http://jurnal.abulyatama.ac.id/index.php/dedikasi>
- Gilvy Langgawan Putra, M., Sandriansyah, S., Maulana Fikri, A., Maulana Jaya Saputra, H., Rovi Harjanto, B., & Saputra, D. (2021). Pembuatan Aplikasi Sinkronisasi Data Akademik Dengan Feeder PDDIKTI Berbasis Web Service. *SPECTA Journal of Technology*, 5(1), 13–21. Retrieved from <https://journal.itk.ac.id/index.php/sjt>
- Handrianus Pranatawijaya, V. (2020). Implementasi Pencatatan Aktivitas Mahasiswa Menggunakan Web Service Pada Feeder PDDIKTI Dengan Metode Extreme Programming. *Jurnal Teknologi Informasi*, 14(2), 179–188. <https://doi.org/10.47111/JTI>
- Hariyanto, M., & Hernawati, H. (2020). Rancang Bangun Aplikasi Web Service Client Feeder PDDIKTI Versi 3.0 Universitas Tarumanagara. *Jurnal Rekayasa Perangkat Lunak*, 1(2), 103–109. Retrieved from <http://jurnal.bsi.ac.id/index.php/reputasi>
- Laga, Y., Nona, R. V., Langga, L., & Jamu, M. E. (2021). Persepsi Mahasiswa Terhadap Kebijakan Merdeka Belajar Kampus Merdeka (MBKM). *Edukatif: Jurnal Pendidikan*, 4(1), 699–706. <https://doi.org/10.31004/edukatif.v4i1.1951>
- Meke, K. D. P., Astro, R. B., & Daud, M. H. (2021). Dampak Kebijakan Merdeka Belajar Kampus Merdeka (MBKM) pada Perguruan Tinggi Swasta di Indonesia. *Edukatif: Jurnal Ilmu Pendidikan*, 4(1), 675–685. <https://doi.org/10.31004/edukatif.v4i1.1940>
- Perwira, R. I., & Santosa, B. (2017). Implementasi Web Service Pada Integrasi Data Akademik Dengan Replika Pangkalan Data Dikti. *Telematika*, 14(1), 1–11. Retrieved from <http://ejurnal.its.ac.id/index.php/teknik/article/view/1258/618>
- Rusli Baharuddin, M. (2021). Adaptasi Kurikulum Merdeka Belajar Kampus Merdeka (Fokus: Model MBKM Program Studi). *Jurnal Studi Guru Dan Pembelajaran*, 4(1). <https://doi.org/10.30605/jsdp.4.1.2021.591>
- Sitorus, Z., Sri Wahyuni, M., & Batubara, S. (2018). Kebutuhan Web Service untuk Sinkronisasi Data Antar Sistem Informasi dalam Universitas. *Jurnal Teknik Dan Informatika*, 5(2), 87–90. Retrieved from <http://pancabudi.ac.id>
- Somya, R., & Michelle Esmeralda Nathanael, T. (2019). Pengembangan Sistem Informasi Pelatihan Berbasis Web Menggunakan Teknologi Web Service Dan Framework Laravel. *Jurnal Techno Nusa Mandiri*, 16(1), 51–58.
- Sulistiyani, E., Khamida, K., Soleha, U., Amalia, R., Hartatik, S., Putra, R. S., ... Andini, A. (2021). Implementasi Merdeka Belajar Kampus Merdeka (MBKM) pada Fakultas Kesehatan dan Non Kesehatan. *Edukatif: Jurnal Ilmu Pendidikan*, 4(1), 686–698. <https://doi.org/10.31004/edukatif.v4i1.1943>
- Supriyatna, A. (2018). Metode Extreme Programming Pada Pembangunan Web Aplikasi Seleksi Peserta Pelatihan Kerja. *Jurnal Teknik Informatika*, 11(1), 1–18. <https://doi.org/10.15408/jti.v11i1.6628>
- Supriyatna, A., & Puspitasari, D. (2021). Implementation of Extreme Programming Method in Web Based Digital Report Value Information System Design. *International Journal of Information System & Technology Akreditasi*, 5(1), 67–75.
- Tohir, M. (2020). Buku Panduan Merdeka Belajar Kampus Merdeka (kesatu). *Direktorat Jenderal Pendidikan Tinggi Kementerian Pendidikan Dan Kebudayaan*.
- Trisnadoli, A. (2021). Implementasi Extreme Programming (XP) Agile Software Development pada Pengembangan Sistem Informasi Keluargaku. *Jurnal Informatika Universitas Pamulang*, 6(2), 305–311. <https://doi.org/10.32493/informatika.v6i2.10088>
- Usman, S. (2020). Implementasi Sistem Informasi Akademik Dengan Feeder PDDIKTI Berbasis Android. *Journal of System and Computer Engineering (JSCE)*, 1(1), 28–37.
- Widodo, S., Brawijaya, H., Retnoningsih, E., Mandiri Jakarta, N., Damai No, J., & Jati Barat Pasar, W. (2018). Integrasi Data Akademik Dengan Aplikasi Feeder PDDIKTI Berbasis Web service. *Bina Insani Ict Journal*, 5(2), 153–162.