



## Application of the Rational Unified Process Method in Web Service Development Payment System Integration with Multibank Virtual Accounts

Sopingi<sup>1</sup> , Ety Meikati<sup>2</sup>, Wijiyanto<sup>3</sup>

<sup>1</sup>Department of Information System, Universitas Duta Bangsa, Surakarta, Indonesia, 57154

<sup>2</sup>Department of Accounting, Universitas Duta Bangsa, Indonesia, Surakarta, 57154

<sup>3</sup>Department of Informatics Management, Universitas Duta Bangsa, Surakarta, Indonesia, 57154

 [sopingi@udb.ac.id](mailto:sopingi@udb.ac.id)

 <https://doi.org/10.37339/e-komtek.v5i1.565>

Published by Politeknik Dharma Patria Kebumen

### Abstract

#### Artikel Info

Submitted:

18-04-2021

Revised:

09-06-2021

Accepted:

09-06-2021

Online first :

30-06-2021

The Payment System is a series of interconnected components consisting of rules, institutions, and procedures used for money transactions in fulfilling the obligations of economic activities. The Duta Bangsa University of Surakarta is a case study in this research where the student payment system is still a billing system. The faculty finance department gives bills tuition fees to students one by one. Moreover, it determines the bank used for payments. It resulted in quite many queues when filling out the study plan card (KRS) and exams. This research aims to produce a web service application and standardization with good performance in the payment process at Duta Bangsa University through virtual accounts from various banks and payment channels. The method used to achieve the authors' goals in this study is the Rational Unified Process (RUP) method. The research result is a web service application that can make invoices, inquiries, payments, and reversals of various bank payment channels. The User Acceptance Test results show a very agreeable value of 94.17%, so that the web service is feasible for the production version application. While testing the response time to measure performance for 7 (seven) days, the average response time was 0.49809 seconds.

**Keywords:** *Web service, Virtual account, Payment, User acceptance test, Response time*

### Abstrak

Sistem pembayaran adalah rangkaian komponen yang saling berhubungan yang terdiri dari aturan, lembaga, dan tata cara yang digunakan untuk transaksi uang dalam memenuhi kewajiban kegiatan ekonomi. Universitas Duta Bangsa Surakarta menjadi studi kasus dalam penelitian ini dimana sistem pembayaran mahasiswa masih menggunakan sistem billing. Bagian keuangan fakultas memberikan tagihan biaya kuliah kepada mahasiswa satu per satu. Dan itu menentukan Bank yang digunakan untuk pembayaran. Hal ini mengakibatkan cukup banyak antrian saat pengisian Kartu Rencana Studi (KRS) dan ujian. Penelitian ini bertujuan untuk menghasilkan sebuah aplikasi web service dan standarisasi dengan kinerja yang baik dalam proses pembayaran di Universitas Duta Bangsa melalui virtual account dari berbagai bank dan payment channel. Metode yang digunakan untuk mencapai tujuan penulis dalam penelitian ini adalah metode Rational Unified Process (RUP). Hasil penelitian berupa aplikasi web service yang dapat membuat invoice, inquiry, pembayaran, dan reversal berbagai channel pembayaran bank. Hasil User Acceptance Test menunjukkan nilai sangat setuju sebesar 94,17% sehingga web service ini layak untuk aplikasi versi produksi. Sedangkan pengujian response time untuk mengukur kinerja selama 7 (tujuh) hari, rata-rata response time adalah 0,49809 detik.

**Kata-kata kunci:** *Layanan web, Akun virtual, Pembayaran, Uji penerimaan pengguna, Waktu respons*



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

## 1. Introduction

The Payment System is a series of interconnected components consisting of rules, institutions, and procedures used for money transactions in fulfilling the obligations of economic activities. The purpose of the payment system is to transfer money from one bank to another, where the transfer process can be done offline using the bookkeeping transfer slip form or online using ATM, i-banking, m-banking, and e-wallet. For example, one of the agencies that implement a payment system is an educational institution with several types of payments whose nominal bills can change with the time of billing, which can be at any time.

Duta Bangsa University of Surakarta (UDB) is a university case study in this research. The number of active students in the 2020/2021 *Genap* more than 4000 students spread over four faculties. UDB cooperates with 2 (two) banks in a student payment system with a billing system. Namely, the faculty finance department bills tuition fees to students one by one and determines the bank used for payments according to each student's faculty. After invoicing, students must come directly to the bank teller by providing a payment deposit slip. This payment system resulted in many queues during the filling out the study plan card (KRS) and exams. This problem can be overcome if online payment facilities cannot queue to tellers to make online payments by providing a virtual account for each student. A virtual account is an identification number given by a bank to an agency's customers as an account number for payment purposes [1].

The payment process using a virtual account can be done anywhere with any channel, but it still requires a billing process that the finance department must do. This billing process certainly makes it difficult for the finance department because each student will have a different bill with a virtual account from another bank.

Previous research entitled "*Pengembangan Sistem Pembayaran Mahasiswa dengan Mobile Payment BTN Syariah*" by Eko Purwanto, et al. [2] using the prototype method has resulted in a student payment system with mobile payments. The weakness of this system is that it is not directly integrated with the financial system and can only be integrated with 1 (one) bank. Furthermore, research conducted by Eko Purwanto has not tested the performance of the payment system.

The research entitled "*Virtual Account dan Tracer Digital Untuk Pembayaran Uang Sekolah Siswa Berbasis Android*" by Lusiana Efrizoni et al. [3] using the waterfall model development

method resulted in an application used as a track record of student payment transactions. Research conducted by Lusiana Efrizoni et al. is not integrated with banks. Therefore, students must upload proof of payment from the bank to the finance department's authorization process.

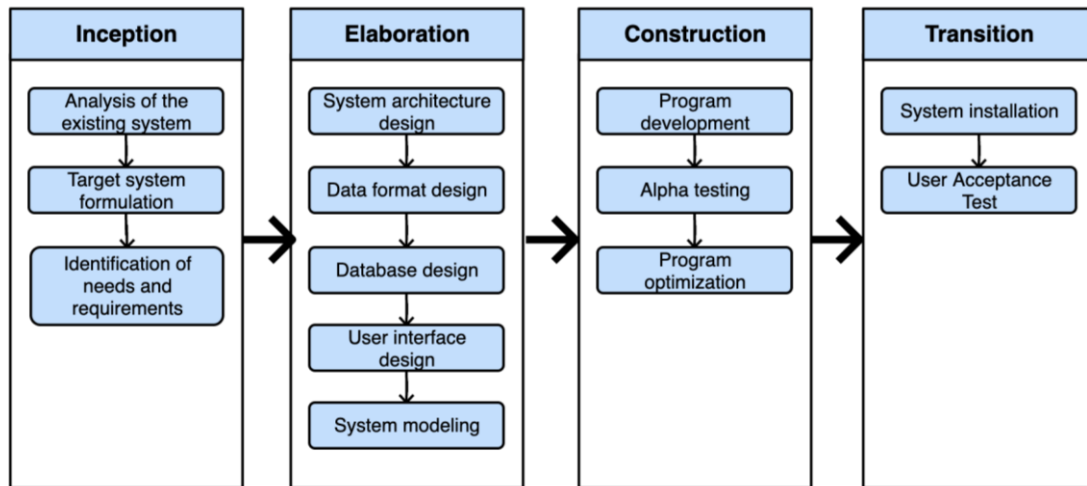
The research entitled "Design and Research of Virtual Payment Systems in Colleges and Universities" by Sun Yang and Lixia Wen [4] stated that traditional payment systems resulted in various problems, especially in service to slow students. This research produces architecture and standardization of payment system development using a virtual payment system.

Based on previous research, the authors researched the development of a web payment system integration with virtual accounts from several banks in UDB where the web service can be accessed with multiplatform from web, mobile, and applications from the bank. The use of web services in data communication between database applications and client applications has a high level of performance with a response time value of less than 1 second [5]. The web service developed in this study will be standardized to meet the transaction processing needs of various teller channels, ATMs, i-banking, m-banking, and e-wallets. The author records the response time using the system log on each request made to determine the level of performance of the developed web service. To get a good understanding, the writer will create a web service that runs natively using the Golang programming language, a programming language capable of building applications with high-performance levels [6].

This research aims to produce a web service application and standardization with good performance in the payment process at UDB through virtual accounts from various banks and transaction channels. The method used to achieve the authors' goals in this study is the Rational Unified Process (RUP) method. This method is used because system development is required for a limited time, and improvements will be made during the development process. In addition, RUP provides the best simulation process framework for system development [7].

## 2. Method

Develop a web integrated payment system web service with virtual accounts using the Rational Unified Process (RUP) method approach. RUP is an iterative approach to application development. The development process focuses on architecture (architecture-centric) and is oriented towards use cases (use case driven) [8]. The stages of the research carried out with the RUP approach are presented in [Figure 1](#).



**Figure 1.** Research Framework with the RUP Approach

Steps at the Inception stage:

- a. Analysis of the existing system is carried out by analyzing the payment system that is currently running so that a list of services that are already available and that need to be developed is generated.
- b. The target system is formulated by mapping the process in the payment system.
- c. Identification of needs and requirements is carried out by carrying out technical specifications and specifications for payment system requirements.

The steps in the Elaboration stage:

- a. The system architecture design describes an overview of the payment system as a whole.
- b. The data format design describes the standardization of data structures for request and response processing.
- c. Database design to describe a relational database structure.
- d. The user interface design is used to adjust the appearance of an existing system to send or receive data from the developed payment system web service.
- e. System modelling using the Unified Modeling Language (UML).

Steps at the Construction stage:

- a. Making web service programs using the Golang programming language and MySQL database.
- b. Alpha testing is done to find program bugs.
- c. Program optimization is an improvement from the alpha testing stage.

Step through the Transition stage:

- a. System installation on a cloud server with Linux Centos 8 and MySQL version 8 operating systems.
- b. User Acceptance Testing involves the financial admin, financial audit, partner banks, and students to test the system based on technical specifications and requirements specifications.

### 3. Result and Discussion

#### 3.1. System Analysis

After conducting observations and interviews with the finance department, financial audit, and the bank, a list of services that are already available and services will be developed is obtained.

The benefits are presented in **Table 1**.

**Table 1.** Services Already Available and Services that need to be developed

Process	Service Available	Service Developed
Bill	Master of tuition fees Another fee master	Invoice creation Invoice cancellation Invoice email notification
<i>Inquiry</i> Bill reading	Check manually in the finance section	Inquiry Process from Teller channel, ATM, i-Banking, m-Banking, and e-wallet
<i>Payment</i> Payment Transactions	Transaction manually or offline	Payment process from Teller channels, ATMs, i-Banking, m-Banking, and e-wallets Validate the nominal payment with the nominal invoice Payment process email notification is successful.
<i>Reversal</i> Transaction Cancellation	It cannot be cancelled	Manual Reversal process from Teller channel Automatic reversal process from ATM, i-Banking, m-Banking, and e-wallet channels
Reconciliation	Manually matching transactions from bank deposit evidence with transaction data	Automatic matching of transactions from the bank to the payment system

Meanwhile, the mapping of the process flow on the web service payment system integrated with virtual accounts is presented in **Figure 2**.

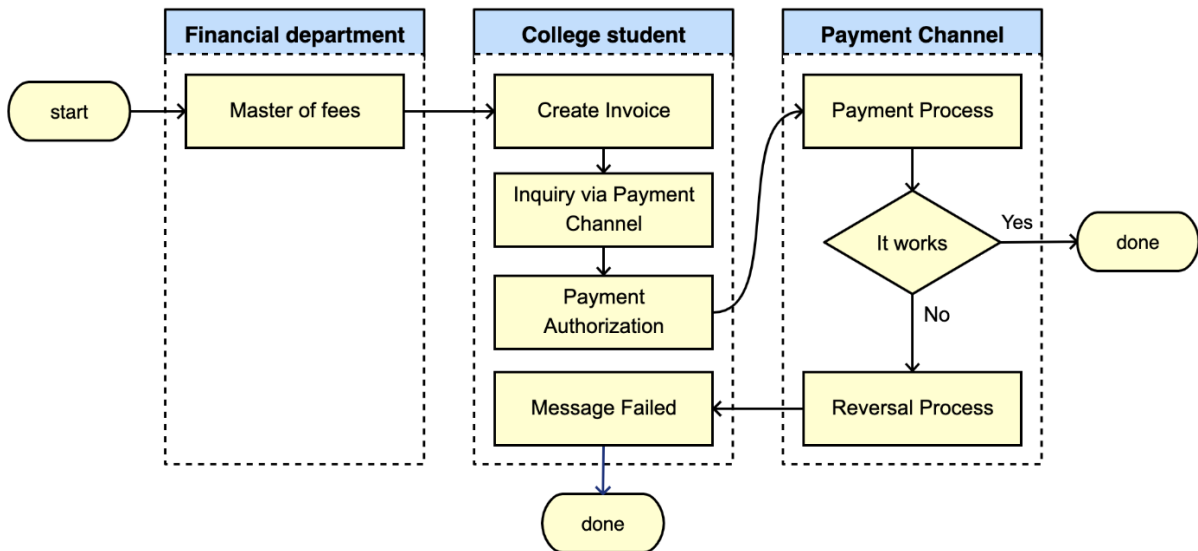


Figure 2. Mapping of the Web Payment System Process Flow Integrated with Virtual Account

### 3.2. System Planning

For the payment system web service to be integrated with virtual accounts from various banks, the authors have designed a web service architecture in Figure 3.

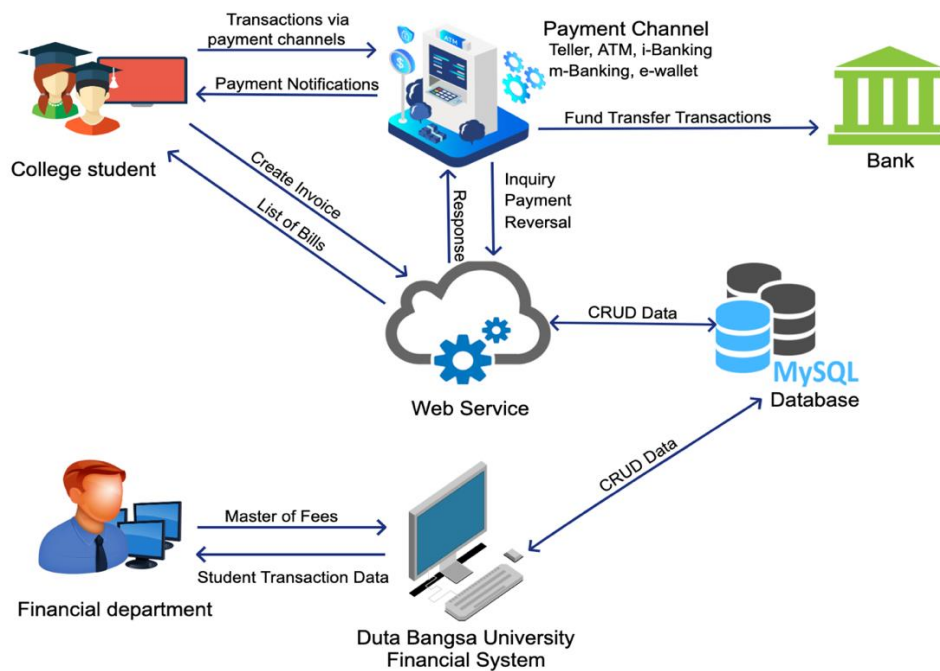


Figure 3. The web service architecture of the payment system is integrated with a virtual account

In developing the web service, the author uses the Representative State Transfer (REST) method is a style of software architecture for distributed hypermedia systems such as the World Wide Web [9], while for data communication via URIs (Universal Resource Identifiers) between student payment systems, payment channels and financial databases in the JavaScript Object

Notation (JSON) format. JSON uses JSON as a data object in the request and response process with a standardized structure to be accessed and read from different payment channels and different banks. The JSON structure is presented in **Table 2**. Request data is sent from the teller payment channel, ATM, i-banking, m-banking, and e-wallet to the payment system web service and will receive a response sent from the payment system web service to the payment channel. Every request will always be validated in the JSON format whether it is following the set standards and little validation that is filled in whether it matches the nominal in the invoice, if there is a mismatch, the web service will give a failed response and provide a message that causes the process to fail so that the validation of the request can prevent the occurrence payment transaction error.

**Table 2.** Structure of the JSON Request and Response Process


Process	Request	Response
<i>Inquiry</i>	<pre>{   "action": "inquiry",   "kodeBank": "",   "kodeBiller": "",   "kodeChannel": "",   "kodeTerminal": "",   "nomorPembayaran": "",   "tanggalTransaksi": "",   "idTransaksi": "",   "checksum": "" }</pre>	<pre>{   "rc": "00",   "msg": "Inquiry Success",   "nomorPembayaran": "",   "idPelanggan": "",   "nama": "",   "idTagihan": "",   "totalNominal": "",   "informasi": [     { "label_key": "", "label_value": "" },     { "label_key": "", "label_value": "" }   ],   "rincian": [     { "kode_rincian": "", "nominal": "" },     { "kode_rincian": "", "nominal": "" }   ] }</pre>
<i>Payment</i>	<pre>{   "action": "payment",   "kodeBank": "",   "kodeBiller": "",   "kodeChannel": "",   "kodeTerminal": "",   "nomorPembayaran": "",   "tanggalTransaksi": "",   "idTransaksi": "",   "idTagihan": "",   "totalNominal": "",   "nomorJurnalPembukuan": "",   "checksum": "" }</pre>	<pre>{   "rc": "00",   "msg": "Payment Success",   "nomorPembayaran": "",   "idPelanggan": "",   "nama": "",   "idTagihan": "",   "totalNominal": "",   "informasi": [     { "label_key": "", "label_value": "" },     { "label_key": "", "label_value": "" }   ],   "rincian": [     { "kode_rincian": "", "nominal": "" },     { "kode_rincian": "", "nominal": "" }   ] }</pre>

Process	Request	Response
Reversal	<pre>{   "action": "reversal",   "kodeBank": "",   "kodeBiller": "",   "kodeChannel": "",   "kodeTerminal": "",   "nomorPembayaran": "",   "tanggalTransaksi": "",   "tanggalTransaksiAsal": "",   "idTransaksi": "",   "idTagihan": "",   "totalNominal": "",   "nomorJurnalPembukuan": "",   "checksum": "" }</pre>	<pre>{   "rc": "00",   "msg": "Reversal Success",   "nomorPembayaran": "",   "idPelanggan": "",   "nama": "",   "idTagihan": "",   "totalNominal": "",   "informasi": [     { "label_key": "", "label_value": "" },     { "label_key": "", "label_value": "" }   ],   "rincian": [     { "kode_rincian": "", "nominal": "" },     { "kode_rincian": "", "nominal": "" }   ] }</pre>

### 3.3. System Development

System development is carried out by upgrading the system that has been running previously. Namely, the payment module for the students' information system and the finance department's online payment transaction module access. Development ran by conducting a request and response process on the web payment system. The User Interface is built with the Webix Framework in the Javascript programming language.

The paying bills data display is presented in [Figure 4](#).

 Refresh Tagihan Program Studi				
Biaya	Total	Potongan	Sudah	Kekurangan
Atribut Dan KMB	1,100,000	0	1,100,000	0
Dana Pengembangan	7,500,000	3,750,000	3,750,000	0
SPP 1	425,000	0	425,000	0
SPP 2	425,000	0	425,000	0
SPP 3	425,000	0	425,000	0
SPP 4	425,000	0	425,000	0
SPP 5	425,000	0	425,000	0

**Figure 4.** Payment Bill



The invoices display that students have independently made is presented in **Figure 5**.

Ada Invoice Aktif		Cetak Invoice	Batalan Invoice
Biaya	Jumlah Bayar		
Kerja Praktik 2020/2021 Genap	500,000		
tes swab khusus kelas A	100,000		
Admin	1,500		
	601,500		
* Pembayaran bisa melalui Teller, ATM, m-banking, atau i-banking			
<b>CARA PEMBAYARAN VIA TELLER</b>			
<ol style="list-style-type: none"> <li>Silahkan datang ke teller <b>BTN Syariah</b></li> <li>Sampaikan nama: ARVIAN SOFYAN MAJID dan nomor pembayaran: <b>70610470201711194</b></li> <li>Teller akan melakukan konfirmasi identitas dan jumlah yang akan dibayarkan</li> </ol>			

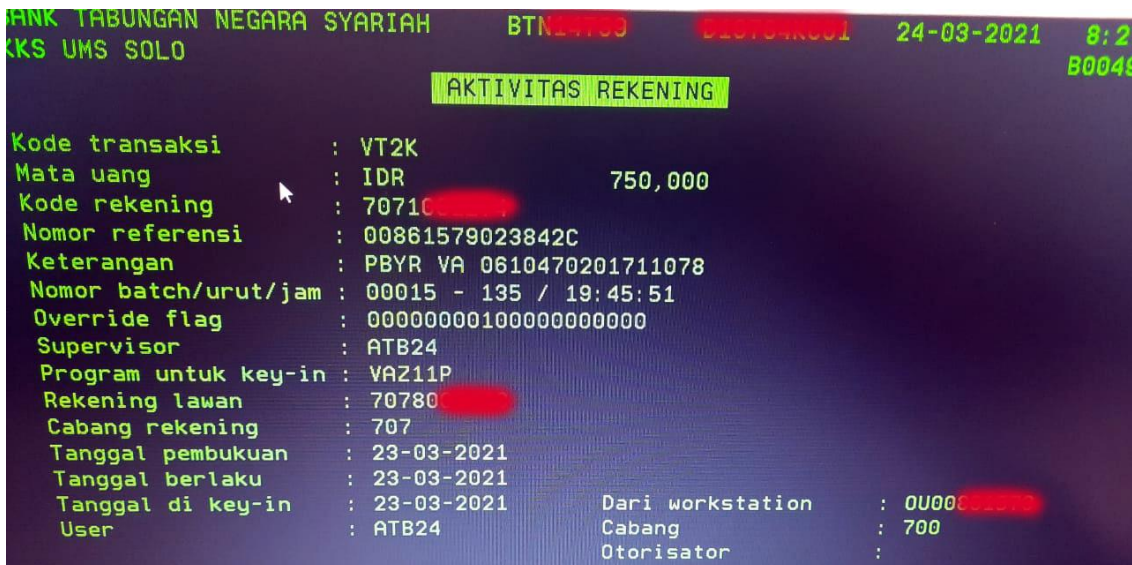
**Figure 5.** Active Payment Invoice Notification

Students can print invoices sent via email to facilitate the payment process, as shown in **Figure 6**.

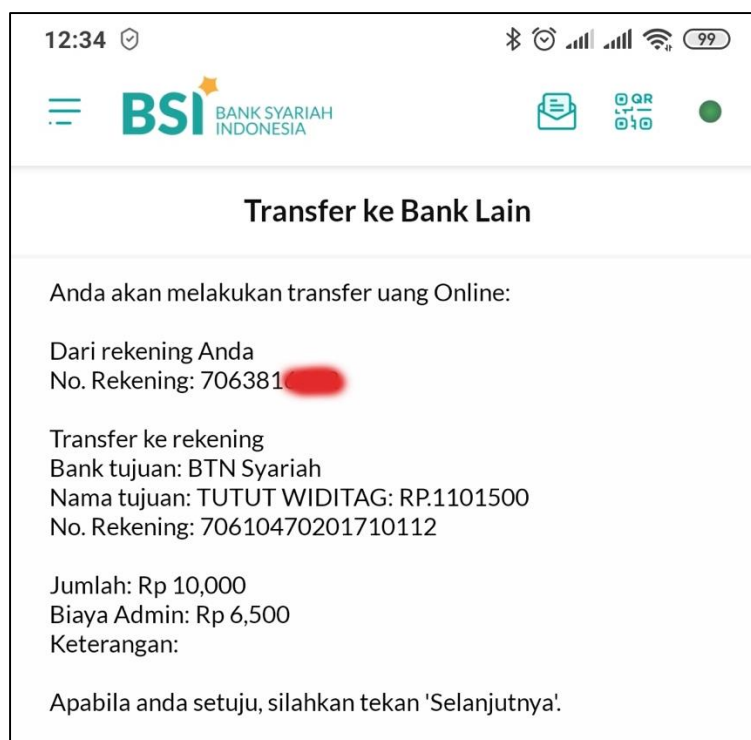
Invoice Pembayaran		Cetak	Tutup
No. Daftar	: 201711194		
Nama	: ARVIAN SOFYAN MAJID		
NIM	: 170103034		
Program Studi	: S1-Teknik Informatika		
<b>INVOICE BELUM DIBAYAR</b>			
No	Jenis Pembayaran	Jumlah	
1	Kerja Praktik 2020/2021 Genap	500.000	
2	tes swab khusus kelas A	100.000	
3	Admin	1.500	
		Total Rp.	601.500
Terbilang: Enam Ratus Satu Ribu Lima Ratus Rupiah			
* Pembayaran bisa melalui Teller, ATM, m-banking, atau i-banking			
<b>CARA PEMBAYARAN VIA TELLER</b>			
<ol style="list-style-type: none"> <li>Silahkan datang ke teller <b>BTN Syariah</b></li> <li>Sampaikan nama: ARVIAN SOFYAN MAJID dan nomor pembayaran: <b>70610470201711194</b></li> <li>Teller akan melakukan konfirmasi identitas dan jumlah yang akan dibayarkan</li> </ol>			
<b>CARA PEMBAYARAN VIA ATM BTN Syariah</b>			
<ol style="list-style-type: none"> <li>Pilih <b>Pembayaran</b></li> <li>Pilih <b>MultiPayment</b></li> <li>Pilih <b>Virtual Account</b></li> <li>Masukkan nomor virtual account: <b>70610470201711194</b> dan tekan <b>Bayar</b></li> </ol>			

**Figure 6.** Print Version of Payment Invoice

Payments with virtual accounts can be made in several channels, and the following are examples of transactions made from several channels in [Figure 7](#), [Figure 8](#), [Figure 9](#), and [Figure 10](#).



**Figure 7.** Payment Transactions through Teller



**Figure 8.** Payment Transactions via m-Banking



Figure 9. Payment Transactions through ATMs

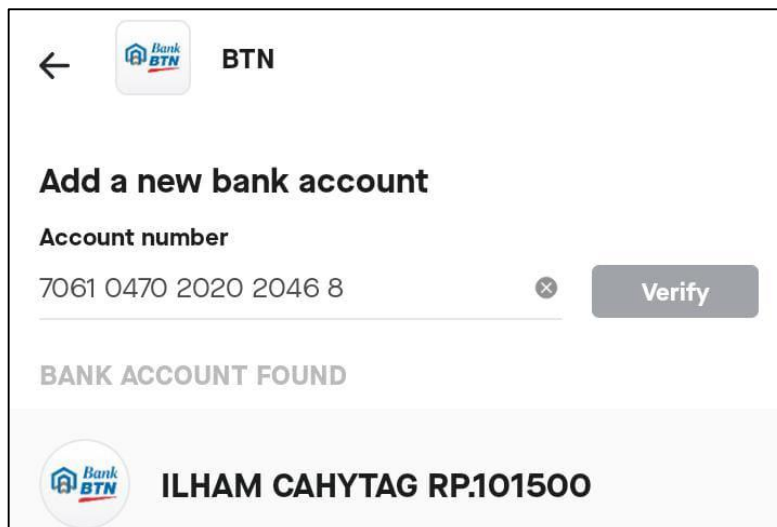


Figure 10. Payment Transactions through GoPay

### 3.4. System Testing

Testing uses the UAT (User Acceptance Test) method, a test carried out by the user with the output in the form of a test result document used as evidence that the application can be accepted and has met needs [10]. UAT testing involved 30 respondents, with the results presented in Table 3.

**Table 3.** UAT Test Results

Statement	Choice	Respondent
The invoice creation process went smoothly	Strongly agree	30
	Agree	0
	Simply Agree	0
	Disagree less	0
	Disagree	0
The Inquiry process went smoothly	Strongly agree	25
	Agree	4
	Simply Agree	1
	Disagree less	0
	Disagree	0
Payment process running smoothly	Strongly agree	28
	Agree	0
	Simply Agree	2
	Disagree less	0
	Disagree	0
The Reversal process went smoothly	Strongly agree	30
	Agree	0
	Simply Agree	0
	Disagree less	0
	Disagree	0

The UAT test results in **Table 3** can be calculated as follows

$$\text{Strongly agree} : \frac{113}{120} \times 100\% = 94,17\%$$

$$\text{Agree} : \frac{4}{120} \times 100\% = 3,33\%$$

$$\text{Simply Agree} : \frac{3}{120} \times 100\% = 2,50\%$$

$$\text{Disagree less} : \frac{0}{120} \times 100\% = 0\%$$

$$\text{Disagree} : \frac{0}{120} \times 100\% = 0\%$$

The next test is the response time test, a test to measure the performance of the request and response processes on the web payment system service. The method used is recording the response time using the log system in the Golang programming language. The recording results for 7 (seven) days can the authors present the average response time each day in **Table 4**.

**Table 4.** Average Response Time for Request and Response Process

Log Date	Number of requests	Average Response Time (ms)
April 1, 2021	765	487,12
April 2, 2021	735	538,29
April 3, 2021	351	521,66
April 4, 2021	271	596,66
April 5, 2021	1.482	463,59
April 6, 2021	671	436,42
April 7, 2021	611	442,87
<b>Average for 7 days</b>		<b>498,09</b>

Based on Table 4, it can be seen that for 7 (seven) days, an average response time of 498.09 milliseconds was obtained where 1000 milliseconds were equal to 1 second, so the average response time was 0.49809 seconds.

#### 4. Conclusion

The web service that has been developed for an integrated payment system with a multibank virtual account is suitable for the production version application based on the results of the UAT test, which states strongly agree with 94.17% for the process of making invoices, inquiries, payments, and reversals running smoothly. While testing the response time to measure performance for 7 (seven) days, the average response time was 0.49809 seconds.

The next researcher can test Quality of Service (QoS), including accessibility, reliability, and performance.

#### References

- [1] R. Hartanto and J. P. Ramli, "Hubungan Hukum Para Pihak Dalam Peer To Peer Lending," *J. Huk. Ius Quia Iustum*, vol. 25, no. 2, pp. 320–338, 2018, doi: 10.20885/iustum.vol25.iss2.art6.
- [2] E. Purwanto and S. Sopingi, "Pengembangan Sistem Pembayaran Mahasiswa Dengan Mobile Payment Btn Syariah," *J. Inkofar*, vol. 1, no. 2, 2019, doi: 10.46846/jurnalinkofar.v1i2.71.
- [3] L. Efrizoni, R. K. Soemanov, and Y. Efendi, "Virtual Account dan Tracer Digital Untuk Pembayaran Uang Sekolah Siswa Berbasis Android," *Semin. Nas. APTIKOM*, vol. 0, no. 0, p. 2019, 2019, [Online]. Available: <http://publikasi.dinus.ac.id/index.php/semnastik/article/view/2842>
- [4] S. Yang and L. Wen, "Design and Research of Virtual Payment System in Colleges and Universities," *Open J. Soc. Sci.*, vol. 08, no. 06, pp. 455–464, 2020, DOI: 10.4236/jss.2020.86035.
- [5] Sopingi, R. Setyowati, and S. Purnomo, "Pengembangan Web Service Digital Assessment Test of English for International Communication (TOEIC)," *J. E-Komtek*, vol. 4, no. 1, pp.

- 75–90, 2020, doi: 10.37339/e-komtek.v4i1.232.
- [6] J. Meyerson, “The go programming language,” *IEEE Softw.*, vol. 31, no. 5, 2014, DOI: 10.1109/MS.2014.127.
- [7] A. Anwar, “A Review of RUP (Rational Unified Process),” *Int. J. Softw. Eng.*, vol. 5, no. 2, pp. 8–24, 2014, [Online]. Available: <http://www.cscjournals.org/library/manuscriptinfo.php?mc=IJSE-142>
- [8] S. Shafiee, Y. Wautelet, L. Hvam, E. Sandrin, and C. Forza, “Scrum versus Rational Unified Process in facing the main challenges of product configuration systems development,” *J. Syst. Softw.*, vol. 170, 2020, DOI: 10.1016/j.jss.2020.110732.
- [9] R. Richards, “Representational State Transfer (REST),” in *Pro PHP XML and Web Services*, Berkeley, CA: Apress, 2006, pp. 633–672.
- [10] B. Hambling and P. van Goethem, *User Acceptance Testing a Step-By-Step Guide*. 2013.