

Design and Implementation of Database Prototype for A Portable Electronic Transaction Device

<https://doi.org/10.3991/ijoe.v18i09.29691>

Syifaul Fuada¹(✉), Akhmad Alfaruq², Trio Adiono³

¹Program Studi Sistem Telekomunikasi, Universitas Pendidikan Indonesia, Bandung, Indonesia

²Electrical Engineering Department, School of Electrical Engineering and Informatics,
Institut Teknologi Bandung, Bandung, Indonesia

³University Center of Excellence on Microelectronics, Institut Teknologi Bandung,
Bandung, Indonesia

syifaulfuada@upi.edu

Abstract—In recent times, electronic money (e-money) has gained significant popularity in the form of smartphone applications, barcodes, and smart card systems. In previous research, we have developed a smartcard-based device equipped with a dual interface to facilitate electronic transactions using contact or contactless technology. These methods involve initiating a connection with a cloud server. This device also supports payment activities, as well as checking or topping up account balances by utilizing contact and contactless smart cards, which are produced in Indonesia by PT. Xirka Silicon Technology. The aforementioned research did not explain the significance of a database which actually plays a vital role in the system's data storage. Therefore, this research aims to provide detailed information on the design of this database. Furthermore, five web pages are designed including, token generator, token list, partner list, payments, and transactions to provide specific services requested by users via a web URL through the database that stores transaction activities carried out by the device. Token verification serves as a mechanism for credit transfer and verification. Additionally, details were provided for a web API and a prototype web user interface. Sample transactions were demonstrated using Postman platform to illustrate the developed API and database storage results.

Keywords—smart card, database, contactless and contact smart card, electronic transaction device

1 Introduction

E-money (Electronic money) has become an integral part of transaction processes in various walks of life around the world. The use of cashless payments is well disseminated into various applications. Furthermore, the traditional payment method, which usually requires physical (conventional) money, has been transformed to e-money in Indonesia [1], [2]. The emergence of the Covid-19 pandemic has further encouraged people to make digital transactions due to the several restrictions imposed on movement and gatherings [3]–[6] and later, it will be predicted to become a dependency on the

consumer goods sector in Indonesia [7]. The payment methods that support transactions using e-money include debit cards, credit cards, electronic wallets, person-to-person (P2P) payment methods, and electronic payments at specially designated kiosks. Based on the technology used, e-money is divided into two types: the chip-based, which is in the form of a rectangular card that connects with a server, and the e-Wallet, which generally uses smartphone applications as a medium for transactions. Popular e-wallets in Indonesia include T-cash, OVO, GOPAY, i.Saku, Sakuku, LinkAja, and others. These e-wallets have their unique mobile application developed by specific merchants for different platforms such as Android, iOS, and Windows, for their end-users. The chip-based and e-wallet systems are widely used in Indonesia, especially at toll gates, banks, and modern canteens; it is a secure technology that uses contact or contactless smart cards in facilitating transactions. However, many companies still do not incorporate the use of this technology in their devices [8].

In previous research, we developed a device for electronic transactions using a SBC (Single Board Computer) as an application processor, a microcontroller as a smart card reader, and a touch screen as the user interface [9]. The unique feature of this device is its ability to carry out transactions in both contact and contactless methods. Verified applications used for electronic transactions are able to check and top-up the fund balances, and also make payments. This proposed system stores transaction data in a cloud database with full encryption. Furthermore, this device is portable, easy to use, and connects with thermal printers via the ethernet port to immediately print the proof of transaction (transaction receipt).

This research succeeded in proving that electronic and internet-based transactions do not necessarily require the use of imported devices since they can be developed in Indonesia. The BIC (Business Innovation Center) recognized this device by issuing it an award as one of the 112 most prospective innovations for the year 2020 (explore this following link: www.bic.web.id). In addition, in terms of mass procurement, this device is certainly more affordable because it is locally produced, especially the card reader, which is manufactured by the first and only chip manufacturing company in Indonesia known as PT. Xirka Silicon Technologies. The continuation of the mass deployment of this proposed solution is determined by how much support the stakeholders have in it [10] and Indonesian government policies [11]. The following are some of the features of this device [9]:

- 1) It performs basic transaction features such as making payments and checking or topping up balance.
- 2) The smart card used follows the ISO/IEC 14443 A/MIFARE standard for the contactless method.
- 3) The device can be operated without connecting to a $220V_{AC}$ mains supply because it uses a built-in rechargeable battery.
- 4) It has been equipped to connect with a thermal printer to print proof of transactions.

Although more details of this system have been presented in previous research [8], [9], the server details have not been thoroughly explained. Therefore, this research also aims to examine the design process of the database and its correlation with previous research. It should be noted that the proposed payment system device is still a

prototype, where the study was carried out at the black box testing stage. It starts from reading contact/contactless smartcards, touchscreen displays function, thermal printer function, and device communication with servers. For this reason, database development is still simple. However, it does not eliminate the main tasks in the application menus within the device, e.g., the balance check, payment, and balance fill menus. The database design is also accompanied by a log table, which aims to determine any transactions done by the system. Moreover, it can support audit data if there are differences in the data between the card and the server. In this report, database storage clusters will be discussed in acceptable detail, including the transaction, involved parties, credit, and debit information.

2 System design

2.1 Overview of previous work

Figure 1 illustrates a block diagram of an electronic transaction system, divided into three main parts including, hardware, software, and server. The system created is capable of performing the general functions of an electronic transaction machine. In addition, about 90% of products and services needed for manufacturing this device are locally sourced, including the smart card readers. The hardware containing Raspberry Pi, Smart Card Reader, and Power supply board is packaged using an appropriate box created using PLA material.

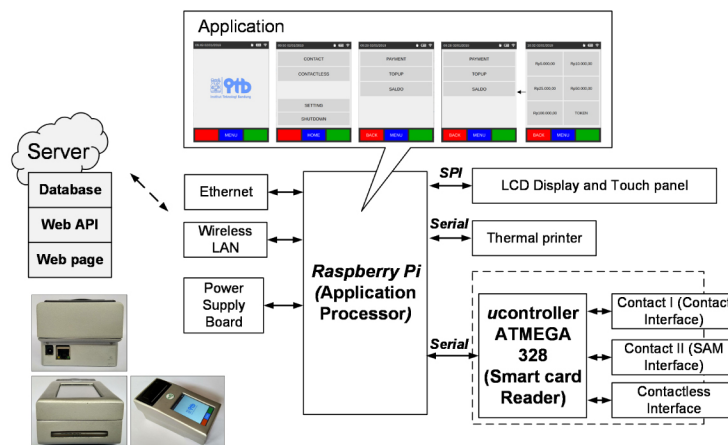


Fig. 1. Block diagram of a portable electronic transaction systems

This device has a graphic user interface for the contact method where the card is inserted into the machine, such as transactions at a bank, and another interface for the contactless method where the card is attached to a machine, such as transactions at a toll gate. Furthermore, it is equipped with a battery charger, which enables it to be used anywhere without plugging it into a $220V_{AC}/50\text{Hz}$ (Indonesia electricity specification) power source until the battery needs recharging. The smart cards can be integrated to

take advantage of several possibilities, including their potential in replacing physical money. To print out the payment proof, the users should connect this device to a thermal printer through LAN connector. Therefore, this device can be used for a wide range of applications, such as payments on toll roads, general financial transactions, and campus canteens to support smart campus concept.

The sharp increase in online and touchless transactions has the potential to leverage this advancement in technology on an industrial and commercial scale, especially during the era COVID-19 pandemic, where several restrictions on movements and gatherings have been imposed. In addition, this smart card system is also be used in the transportation sector to ensure the punctuality of public transport vehicles in terms of their arrival and departure times. Therefore, this device is expected to not only improve the safety and comfort of passengers but to also improve accessibility to vehicles serving routes because it stores electronic records of vehicle and passenger movement data at the terminals provided by the system. The discussion on the hardware design has been elaborated in [8], and the transaction workflow has been discussed in [9]. This research will also examine details on the server.

2.2 Design of a simple server

The sequential process of creating a server is divided into three main parts namely, database, web API, and an interfacing web page. The figures generated by the token is stored in the database, which consists of four tables titled “log”, “partner”, “*partner_activity*”, and “token”. The log table stores data relating to transactions on the smart card, such as payments and checking or topping-up balance. The partner table stores the list of registered partners, the *partner_activity* table stores payment activities, and the token table stores the generated tokens.

Furthermore, the following are the titles assigned to the rows on the log table: “*Id_card*” (stores the smart card ID of the user), “*DateTime*” (stores activity dates), “*debit*” (stores debit nominal), and “*credit*” (stores credit nominal). The Partner table requires “*partner_id*” (stores partner ID), “*name*” (stores partner names), “*addresses*” (stores partner addresses), and “*device_id*” (stores a list of devices of the partner). The *partner_activity* table requires data such as *DateTime* (stores the dates and the time of any activity), “*Id_mitra*” (stores the ID of the partner receiving payment), “*id_card*” (stores ID of the smart card used for payments), and “*total*” (stores nominal payments). For the token table, what needs to be stored is “*dt_generated*” (stores the date the token code was generated), “*id_token*” (stores the nominal value of the balance), “*dt_activated*” (stores the date the token was used) and “*is_valid*” (stores the usage status of the token; the column will be 1 if the token is valid and 0 if the token has been used). The database structures for the log, partner, *partner_activity* and token tables are shown in Tables 1–4, respectively. Tables also present the size of a stored row data.

After the database was successfully created, a web API (Application Programming Interface) was created to provide specific services requested by users via a web URL (Universal Resource Locator). Furthermore, the web API is used to provide data

required by transaction devices, store data sent from these devices, as well as update the data requested by the device. It was built with PHP post using JavaScript Object Notation (JSON) as a lightweight data exchange format which is translated by the computer in order to make it readable for the end-users of the website. This format is partly based on the JavaScript Programming Language, ECMA-262 Standard 3rd Edition. The details of the web API and its functions can be seen in Table 5.

Table 1. Database structure for the “log”

No.	Colom Name	Data Type	Length of Data	Storage (Bytes)
1	No	int	11	4
3	Datetime	datetime	0	8
4	Id_card	varchar	16	16
5	Debt	varchar	64	64
6	Credit	varchar	64	64
7	Saldo	varchar	64	64
Total of Storage				220

Table 2. Database structure for the “partner”

No.	Colom Name	Data Type	Length of Data	Storage (Bytes)
1	No	int	16	4
3	id_mitra	int	16	4
4	Name	varchar	160	160
5	Address	text	0	Maximum 65.535 (Depends on the numbers of characters)
6	id_perangkat	varchar	64	64
7	is_deleted	enum	0	1
Total of Storage				233–65768

Table 3. Database structure for the “partner_activity”

No.	Colom Name	Data Type	Length of Data	Storage (Bytes)
1	no	int	11	4
3	datetime	datetime	0	8
4	id_mitra	varchar	16	16
5	id_card	varchar	16	16
6	Total	varchar	64	64
Total of Storage				108

Table 4. Database structure for the “token”

No.	Colom Name	Data Type	Length of Data	Storage (Bytes)
1	No	int	11	4
3	dt_generated	datetime	0	8
4	id_user	varchar	16	16
5	id_token	varchar	20	20
6	nominal	int	16	4
7	dt_activated	datetime	0	8
8	is_valid	enum	0	1
Total of Storage				61

Table 5. List of web APIs created for communication between servers and tools

Web API Name	Description
<i>submit_log</i>	<p>It is used to send transaction data from the device to the server. If the transaction is a payment, the credit value is inputted with the number 0. If the transaction is a balance top-up, the debit value is inputted with the number 0. The parameters that need to be sent are:</p> <ul style="list-style-type: none"> • debit: debit nominal sent to the server. • credit: credit nominal sent to the server. • balance: balance nominal sent to the server.
<i>partner_activity</i>	<p>It is used to send the history of transactions made by the device. For partner_activity, the type is filled with the value “log”. The parameters that need to be sent are:</p> <ul style="list-style-type: none"> • type: the type of activity on the partner. • <i>id_mitra</i>: id of the partner who received the payment. • <i>id_card</i>: id of the smart card used for payment. • total: the total amount of payments made.
<i>partner_list</i>	<p>It is used to request a list of partners registered on the transaction device. For partner_list, the type is filled with the value “list”. The parameters that need to be sent are:</p> <ul style="list-style-type: none"> • type: the type of activity on the partner. • <i>id_device</i>: id of the device used.
<i>token_check</i>	<p>It is used to check whether the token entered in the device is valid or not. For token_check, the type is filled with the value “check”. The parameters that need to be sent are:</p> <ul style="list-style-type: none"> • type: activity type on token • <i>id_token</i>: token id entered in the device. • <i>id_user</i>: user id used to access the data. • <i>key_user</i>: the key used to access data.
<i>token_update</i>	<p>It is used to change the token on the server from valid to invalid because it has already been used. For token_update, the type is filled with the value “update”. The parameters that need to be sent are:</p> <ul style="list-style-type: none"> • type: activity type on token • <i>id_token</i>: the token id entered in the device. • <i>id_user</i>: the user id used to access the data. • <i>key_user</i>: the key used to access data.

Web API testing is carried out using the postman application, whose primary function is to perform queries. These queries are compared with the database in order to ascertain the success or failure of the queries. The web API test results are shown in Figures 2–10, it is concluded that all queries performed using the Web API were successful; this allows clients to interact with the created web API with ease.

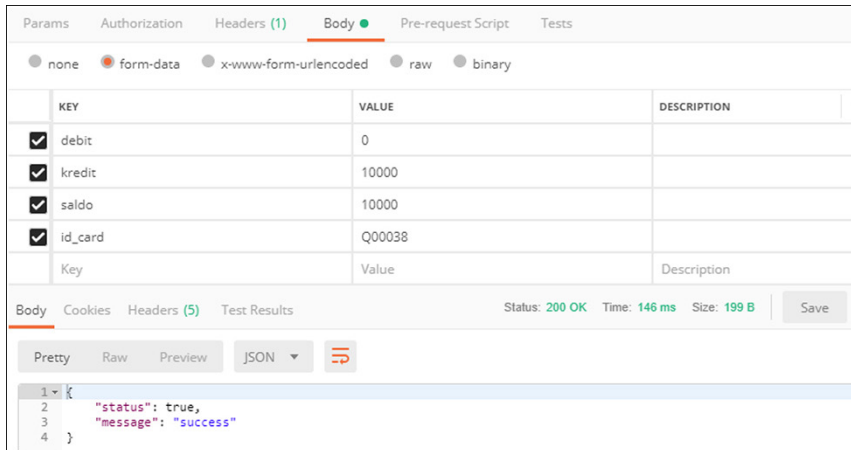


Fig. 2. A screenshot of Web API for submitting log

no	datetime	id_card	debit	kredit	saldo
208	2019-01-07 23:21:29	Q00038	0	10000	10000
207	2019-01-02 22:37:03	Q00002	24000	0	867062

Fig. 3. A screenshot of the results of submitting logs using the Web API

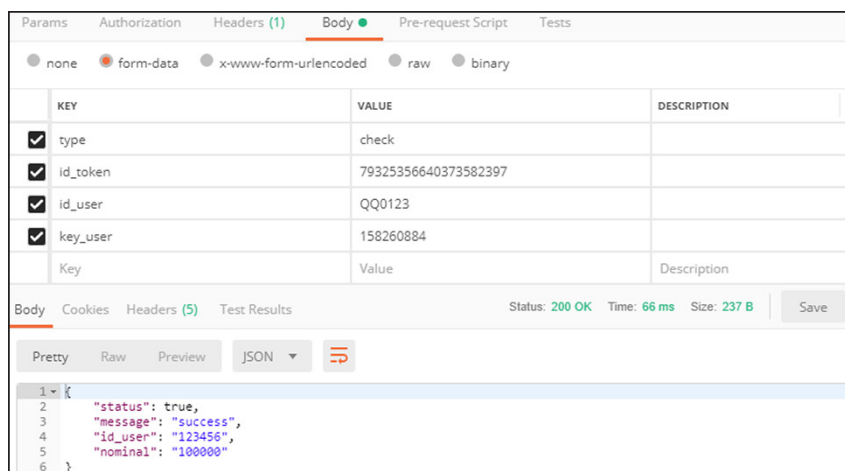


Fig. 4. A screenshot of Web API to check token

no	dt_generated	id_user	id_token	nominal	dt_activated	is_valid
10	2019-01-02 22:13:5	123456	79325356640373582397	100000	(Null)	1
9	2019-01-02 10:43:5	123456	94162187945439930499	50000	2019-01-02 10:52:52	0
8	2019-01-02 10:43:2	123456	40772747338159274781	50000	(Null)	1

Fig. 5. A screenshot data of tokens checked by Web API

Params Authorization Headers (1) **Body** Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> type	update	
<input checked="" type="checkbox"/> id_token	79325356640373582397	
<input checked="" type="checkbox"/> id_user	QQ0123	
<input checked="" type="checkbox"/> key_user	158260884	
Key	Value	Description

Body Cookies Headers (5) Test Results Status: 200 OK Time: 74 ms Size: 199 B Save

Pretty Raw Preview JSON

```

1 {
2   "status": true,
3   "message": "success"
4 }
    
```

Fig. 6. A screenshot of update token status using Web API

no	dt_generated	id_user	id_token	nominal	dt_activated	is_valid
10	2019-01-02 22:13:5	123456	79325356640373582397	100000	2019-01-07 23:28:50	0
9	2019-01-02 10:43:5	123456	94162187945439930499	50000	2019-01-02 10:52:52	0
8	2019-01-02 10:43:2	123456	40772747338159274781	50000	(Null)	1

Fig. 7. A screenshot of the result of updating the token status using the Web API

Params Authorization Headers (1) **Body** Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> type	log	
<input checked="" type="checkbox"/> id_mitra	112244	
<input checked="" type="checkbox"/> id_card	Q00038	
<input checked="" type="checkbox"/> total	3800	
Key	Value	Description

Fig. 8. A screenshot of Web API for sending payment activity

no	datetime	id_mitra	id_card	total
24	2019-01-07 23:24:41	112244	Q00038	3800
23	2019-01-02 22:37:03	112244	Q00002	24000
22	2019-01-02 22:34:53	112243	Q00002	70000

Fig. 9. A screenshot of the results of submitting payment activities using the Web API

The screenshot shows a web API client interface with the following details:

- Params:** type: list, id_perangkat: 38
- Status:** 200 OK, Time: 141 ms, Size: 510 B
- JSON Response:**

```

1 {
2   "type": "list",
3   "data": [
4     {
5       "id_mitra": "112233",
6       "nama": "Rumah Bundo",
7       "alamat": "Bandung, Jawa Barat",
8       "is_deleted": "false"
9     },
10    {
11     "id_mitra": "112234",
12     "nama": "Warteg Bahari",
13     "alamat": "Bandung, Jawa Barat",
14     "is_deleted": "false"
15    },
16    {
17     "id_mitra": "112243",
18     "nama": "Sate Maranggi",
19     "alamat": "Bandung, Jawa Barat",
20     "is_deleted": "false"
21    }
22  ]
23 }
```

Fig. 10. A screenshot of the results of partner list queries performed with Web API

3 Results and analysis

As aforementioned before, the design and implementation of a transaction storage database for online transactions serves as the major focus point for this report. This work continues previous research which demonstrates a hardware interface capable of dual interface transactions, contract and contactless. After the Web API has been created, a web page that will be used to display the data in the database was built. The five pages created include, token generator, token list, partner list, payments, and transactions. The user interface, which is still in a prototype stage consists of two parts namely: the navigation section, which is used to place buttons, combo boxes, textboxes,

dropdown lists, labels, links, menus, images, etc.; and the view section is used to display the contents of the created web page. The navigation section contains five main menus: token generator, token logs, partners, transaction partners, and transaction logs. Furthermore, the token generator page is used to generate the token codes which are used by the device (Figure 11). The token list page contains a list of generated tokens and their status (Figure 12), and the partner page contains a list of registered partners (Figure 13). To avoid placing any burden on the server, the database was made scalable in order to easily add or remove resources if there is an increase in the number of users accessing the web page. The payment page contains the list of payments that have been made with this device (Figure 14), and the transaction page contains smart card activities, which includes both payments and balance top-ups (Figure 15).

Figures 16a and b show a flowchart that explains the relationship between the web UI and API in the mechanism of how the system works in Figure 1 in more detail. This flowchart also describes the data exchanged between the hardware and the database (including the web UI). For example, when the menu button is pressed, and the token generator buttons are selected, it will enter the token generator menu. Then, there is a query to the token table in the database. Regarding the encryption process for data packets, we also realize that it is essential for user privacy. Token creation techniques, database security, and the immutability of stored data are exciting things to study. As a form of security against data exchange, we used Secure REST API. Every sending or requesting data from the server to the hardware or vice versa always uses an SSL (Secure Sockets Layer) connection. The data stored in the database is not encrypted since there is only limited to hardware testing devices.

The token in the database was used to top up the balance. The token uses a 20-digit random number obtained using the `mt_rand()` function served by PHP. When the submit button is pressed, the random generator program will create numbers employed for the tokens, then be concurrently stored with the top up balance value in the database (Figure 17a). Finally, the token is entered into the “top up” menu displayed on the hardware (Figure 17b). If the token number is the same as displayed in the database, the token is considered valid, and the balance will be added according to the registered nominal. If they are not the same, the process will fail.

Several critical metrics, including database size requirements based on user base, are also essential. Tables 1–4 provide storage information by default. For example, our system offers 220, 108, and 61 bytes of storage for Log, Partner activity, and Token, respectively. Meanwhile, for the partner table, 233 to 65768 bytes are available depending on the address variable. More detailed calculations are required for larger-scale requirements by referring to the default settings. Our system is modular according to needs, where the scaling-up process can be done by calculating the target users who use the system. The considerations include how many transactions per day and how long the data is stored.

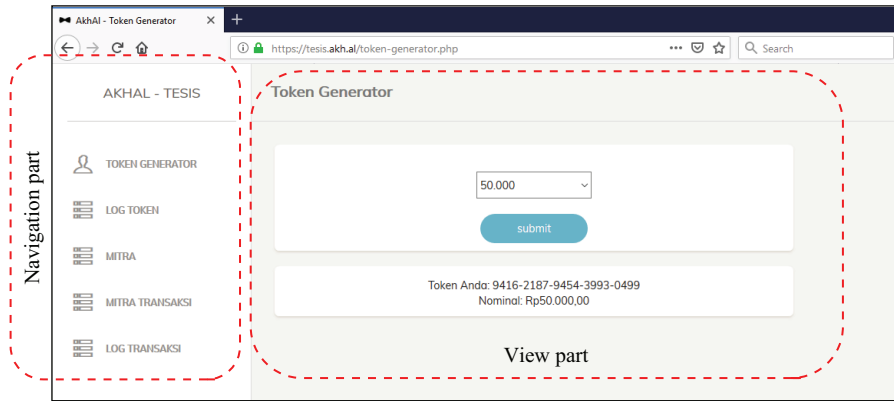


Fig. 11. Token generator page

	Dihasilkan	Nominal	ID Token	Digunakan	Status
TOKEN GENERATOR					
TOKEN	2018-12-25 23:21:53	Rp30.000,00	8938-4431-8928-4221-4067	2018-12-25 23:43:06	Belum digunakan
MITRA	2018-12-25 23:24:11	Rp50.000,00	6741-4058-3136-1341-4640	2018-12-25 23:49:36	Sudah digunakan
PEMBAYARAN	2018-12-25 23:45:33	Rp5.000,00	2127-4663-9133-7266-3061		Belum digunakan
TRANSAKSI	2019-01-02 10:38:23	Rp50.000,00	6254-3038-7104-6607-4437		Belum digunakan
	2019-01-02 10:41:31	Rp50.000,00	7333-3617-4841-1585-2605		Belum digunakan
	2019-01-02 10:41:41	Rp50.000,00	1281-5955-9425-4296-1712		Belum digunakan
	2019-01-02 10:41:57	Rp50.000,00	9956-2676-0286-6882-8639		Belum digunakan
	2019-01-02 10:43:26	Rp50.000,00	4077-2747-3381-5927-4781		Belum digunakan
	2019-01-02 10:43:50	Rp50.000,00	9416-2187-9454-3993-0499	2019-01-02 10:52:52	Sudah digunakan

Fig. 12. Token list page

	ID Mitra	Nama Mitra	Alamat Alamat	ID Perangkat
TOKEN GENERATOR				
TOKEN	112233	Rumah Bundo	Bandung, Jawa Barat	38
MITRA	112234	Warteg Bahari	Bandung, Jawa Barat	38
PEMBAYARAN	112243	Sate Maranggi	Bandung, Jawa Barat	38
	112244	Baso Pelangi	Bandung, Jawa Barat	

Fig. 13. Partner list page display

TOKEN GENERATOR					
	Tanggal	ID Mitra	Nama Mitra	Alamat Mitra	Total
TOKEN	2019-01-07 23:24:41	112244	Baso Pelangi	Bandung, Jawa Barat	Rp3.800,00
MITRA	2019-01-02 22:37:03	112244	Baso Pelangi	Bandung, Jawa Barat	Rp24.000,00
PEMBAYARAN	2019-01-02 22:34:53	112243	Sate Maranggi	Bandung, Jawa Barat	Rp70.000,00
TRANSAKSI	2019-01-02 10:16:51	112244	Baso Pelangi	Bandung, Jawa Barat	Rp26.000,00
	2019-01-02 10:13:26	112233	Rumah Bundo	Bandung, Jawa Barat	Rp56.000,00
	2019-01-02 10:06:46	112233	Rumah Bundo	Bandung, Jawa Barat	Rp68.000,00
	2019-01-02 10:04:33	112233	Rumah Bundo	Bandung, Jawa Barat	Rp86.000,00
	2019-01-02 10:00:38	112233	Rumah Bundo	Bandung, Jawa Barat	Rp86.000,00
	2018-12-28 17:09:12	112244	Baso Pelangi	Bandung, Jawa Barat	Rp9.000,00

Fig. 14. Payment page display

TOKEN GENERATOR					
	Time	ID Kartu	Debit	Kredit	Saldo
TOKEN	2019-01-07 23:21:29	Q00038	0	10000	10000
MITRA	2019-01-02 22:37:03	Q00002	24000	0	867062
PEMBAYARAN	2019-01-02 22:34:53	Q00002	70000	0	891062
TRANSAKSI	2019-01-02 22:24:42	Q00002	0	100000	961062
	2019-01-02 22:21:24	Q00002	0	50000	861062
	2019-01-02 22:18:41	Q00002	0	100000	811062
	2019-01-02 22:12:16	Q00002	0	50000	711062
	2019-01-02 10:52:52	Q00001	0	50000	307478
	2019-01-02 10:46:19	Q00001	0	50000	257478

Fig. 15. Transaction page display

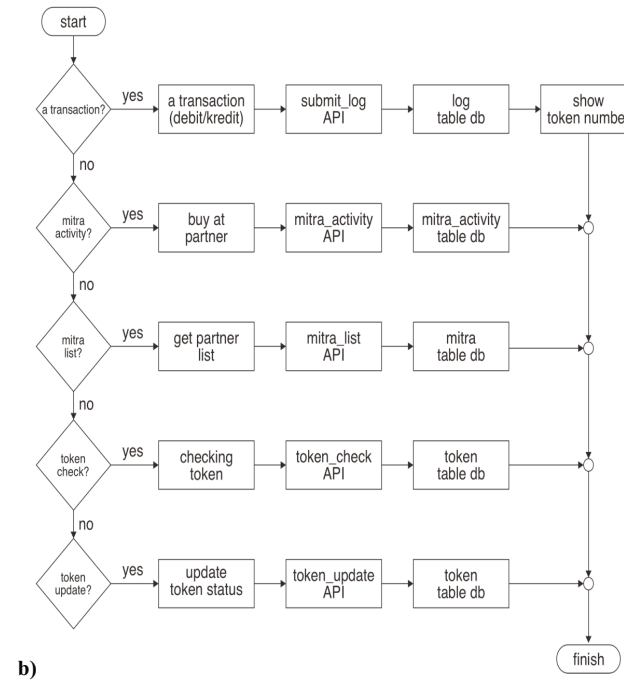
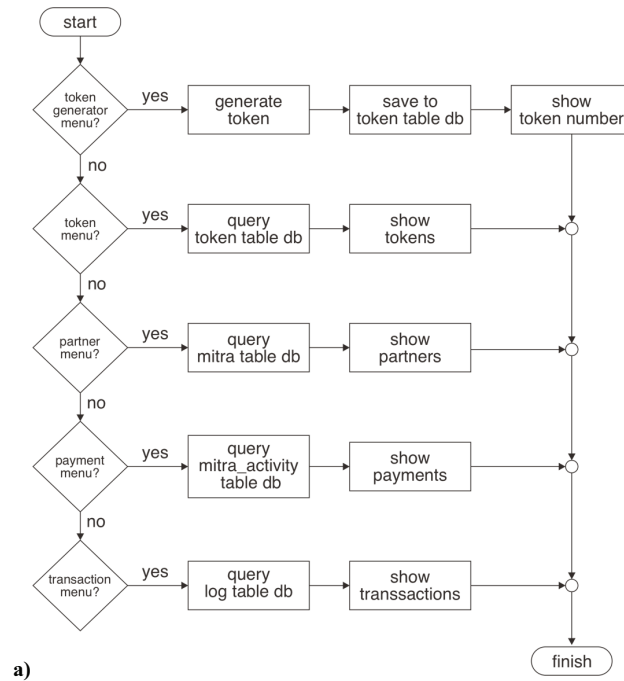


Fig. 16. Flowcharts of: (a) UI web; (b) API

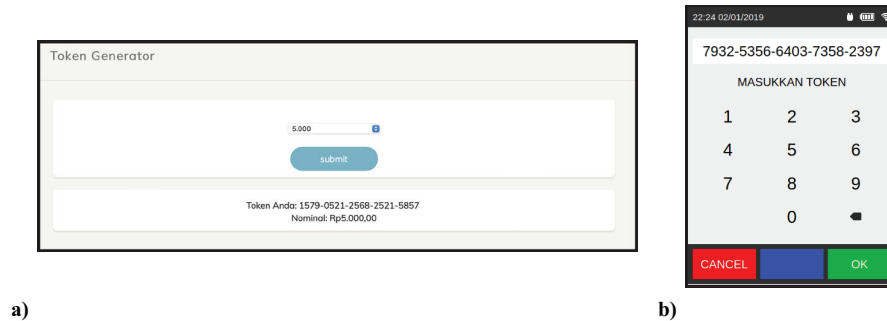


Fig. 17. Token generated from: (a) UI web; (b) Applications

4 Conclusion

The use of e-money in conducting financial transactions has provided efficiency and comfort to users. As in previous work, we have developed a device to read smart cards in order to enable users check their balance, details of financial transactions as well as make payments. Based on database test data, the device has proved successful in sending transaction data to the server using an internet network connected via a WLAN (Wireless Local Area Network). Furthermore, this code generated by the token was successfully saved in a database which comprises of four tables including, the log table, used to store data related to transactions on the smart card specifically for payment and balance top-up activities; the partner table, used to store the list of registered partners; the *partner_activity* table, used to store payment activities; and the token table, used to store the codes generated by the token. The database at this research stage is still quite simple, so further complexity development is needed. Overall, this report captures the design elements of a simple database and the interface API. Hence, data for testing comprehensive metrics, such as transaction latency on a single user, two users, three users, Etc., is required to build a database in further development. Furthermore, scalability and costs factor will be also included.

5 References

- [1] D. Octabriyantiningtyas, E. Suryani, and A. R. Jatmiko, "Modeling Customer Satisfaction with the Service Quality of E-Money in Increasing Profit of PT. Telekomunikasi Indonesia," *Procedia Computer Science*, vol. 161, pp. 943–950, Jan. 2019, <https://doi.org/10.1016/j.procs.2019.11.203>
- [2] N. Saraswati and I. Mukhlis, "The Influence of Debit Card, Credit Card, and E-Money Transactions Toward Currency Demand in Indonesia," *Quantitative Economics Research*, vol. 1, no. 2, Dec. 2018, <https://doi.org/10.17977/um051v1i22018p87-94>
- [3] D. N. Aryani et al., "A Study on Consumer Behaviour: Transition from Traditional Shopping to Online Shopping During the COVID-19 Pandemic," *International Journal of Applied Business and International Management (IJABIM)*, vol. 6, no. 2, Aug. 2021, <https://doi.org/10.32535/ijabim.v6i2.1170>
- [4] D. Mustomi, A. Puspasari, A. Azizah, and D. Wijayanti, "Analisis Belanja Online di Kalangan Mahasiswa Pada Masa Pandemi Covid 19," *Jurnal Akrab Juara*, vol. 5, no. 4, Nov. 2020.

- [5] N. Hanifah and D. R. Rahadi, “Analisis Perilaku Konsumen dalam Memutuskan Pembelian Secara Online pada Masa Pandemi Covid-19,” *SULTANIST: Jurnal Manajemen dan Keuangan*, vol. 8, no. 2, Dec. 2020, <https://doi.org/10.37403/sultanist.v8i2.206>
- [6] J. Koch, B. Frommeyer, and G. Schewe, “Online Shopping Motives during the COVID-19 Pandemic—Lessons from the Crisis,” *Sustainability*, vol. 12, no. 24, Jan. 2020, <https://doi.org/10.3390/su122410247>
- [7] M. Novethesia, R. Sanjaya, and F. Margaretha, “Kendala Keuangan dan Saling Ketergantungan Dalam Keputusan Keuangan Pada Sektor Consumer Goods di Indonesia,” *Jurnal Muara Ilmu Ekonomi dan Bisnis*, vol. 4, no. 2, Jun. 2020, <https://doi.org/10.24912/jmieb.v4i2.7797>
- [8] T. Adiono, R. I. Malingkas, A. C. Swastika, and S. Fuada, “Electronic Transaction Device Based on Contact Smart Card using Programmable System-on-Chip,” in *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*, Pangkal Pinang, Oct. 2018, pp. 51–56, <https://doi.org/10.1109/ICECOS.2018.8605217>
- [9] S. Fuada, A. Alfaruq, and T. Adiono, “A Portable Electronic Transaction Device Based on Dual Interface Smart Card,” *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 16, no. 3, pp. 27–45, Mar. 2020, <https://doi.org/10.3991/ijoe.v16i03.11639>
- [10] T. Yuliandani, “Integrasi Kepentingan Stakeholder Dalam Pembangunan Berkelanjutan di Kabupaten Bojonegoro,” Skripsi, Ilmu Politik, Fakultas Ilmu Sosial dan Ilmu Politik, Universitas Airlangga, Surabaya, 2017. Accessed: Apr. 26, 2021. [Online]. Available: <http://repository.unair.ac.id/67979/>
- [11] G. R. Redu, “Tindak Lanjut Pengambilan Kebijakan Luar Negeri Oleh Indonesia Dalam Menghadapi Kerjasama ACFTA (ASEAN-China Free Trade Agreement),” Skripsi, Universitas Brawijaya, Malang, Indonesia, 2013.

6 Authors

Syifaul Fuada is with the Program Studi Sistem Telekomunikasi Universitas Pendidikan Indonesia (UPI) as a young Lecturer. His research interests include analog circuit design and instrumentation, circuit simulation, engineering education, IoT, multimedia learning development and VLC. The author can be found at syifaulfuada@upi.edu. Ruang Dosen Sistel, Gd. Baru Lt. 1, UPI Purwakarta campus, Jl. Veteran No.8, Nagri Kaler, Kec. Purwakarta, Kabupaten Purwakarta, Jawa Barat 41115, Indonesia.

Akhmad Alfaruq received a B.Eng. in Electrical Engineering from Telkom University, Indonesia, and he is with the M.Sc. in Electrical Engineering with Microelectronics option from the School of Electrical Engineering and Informatics, Institut Teknologi Bandung (ITB), Indonesia. His research interests include embedded system, hardware assembly, and internet of things. The author can be found at alfaruq@students.itb.ac.id. Gd. Achmad Bakrie (LABTEK VIII) Lt. III, ITB campus, Jln. Ganesha No.10, Bandung city 40132, West Java, Indonesia.

Trio Adiono is a Full professor and a senior lecturer at the School of Electrical Engineering and Informatics. His research interests include VLSI design, signal and image processing, artificial intelligence, VLC, smart cards, and electronics solution design and integration. The author can be found at tadiono@stei.itb.ac.id. IC Design Laboratory, Gd. Riset dan Inovasi ITB (Eks. PAU) Lt. IV, ITB Campus, 4J66+M2V, Lebak Siliwangi, Coblong, Bandung City, West Java 40132, Indonesia.

Article submitted 2022-01-23. Resubmitted 2022-04-28. Final acceptance 2022-05-13. Final version published as submitted by the authors.