

RANCANG BANGUN SISTEM KEAMANAN PADA KOTAK PENYIMPANAN UANG (*MONEYBOX PLUS*) BERBASIS MIKROKONTROLER ARDUINOAchmad Annabawi<sup>1</sup>, Firman Hadi Sukma Pratama\*<sup>2</sup><sup>1</sup>Program Studi Informatika, Universitas Wijaya Kusuma Surabaya, annabawi32@gmail.com<sup>2</sup>Program Studi Informatika, Universitas Wijaya Kusuma Surabaya, firmanpratama@uwks.ac.id

\*)Korespondensi : firmanpratama@uwks.ac.id

## Abstrak

Meskipun sudah banyak yang menyediakan alat penyimpan uang secara praktis, namun sebagian masih ada beberapa orang yang lebih memilih untuk menyimpan uangnya secara pribadi di rumah, dikarenakan tidak semua Bank memiliki banyak cabang atau mesin setor tunai yang banyak, serta adanya ketidak seimbangan dalam jarak tempuh yang jauh menuju alat setor tunai. Perancangan alat *MoneyBox Plus* dengan sistem keamanan yang menggunakan pengunci magnet dan dilengkapi dengan alarm serta penggunaan kode pengaman (*password*) sebagai akses masuk pada *MoneyBox Plus*. Metode yang digunakan dalam pengumpulan dan analisis data serta langkah-langkah yang akan dilakukan pada penelitian ini yaitu metode *prototype*. Pada penelitian ini, Algoritma *Fisher Yates* diimplementasikan menggunakan Fungsi yang terdapat pada program Arduino yaitu *random()*. Fungsi *random()* merupakan Fungsi yang digunakan untuk menghasilkan nilai secara acak. Hasil yang dituju dalam penelitian ini adalah sistem keamanan tersebut dapat terpantau lebih baik lagi dan memastikan bahwa hanya orang berhak saja yang memiliki akses, serta dapat mengantisipasi dalam menyimpan uang.

**Kata Kunci:** Kotak Penyimpanan Uang, Arduino Mega 2560, *Solenoid Lock*

## Abstract

*Even though many have provided practical means of saving money, some people still prefer to keep their money privately at home, because not all banks have many branches or cash deposit machines, and there is an imbalance in the distance traveled. far to the cash deposit tool. The design of the MoneyBox Plus tool with a security system that uses a magnetic lock and is equipped with an alarm and the use of a security code (password) as access to MoneyBox Plus. The method used in data collection and analysis as well as the steps to be carried out in this study is the prototype method. In this study, the Fisher Yates Algorithm is implemented using the function contained in the Arduino program, namely random(). The random() function is a function that is used to generate random values. The aim of this research is that the security system can be monitored better and ensure that only entitled people have access, and can anticipate saving money.*

**Keywords:** Money Storage Box, Arduino Mega 2560, *Solenoid Lock*

## I. PENDAHULUAN

Bank melengkapi pelayanannya dengan menyediakan mesin pengambil uang otomatis atau lebih dikenal dengan ATM (Anjungan Tunai Mandiri). ATM ini tersebar dari kota sampai ke desa. Selain itu bank juga melengkapi layanannya dengan menyediakan mesin setor tunai. Akan tetapi mesin setor tunai ini masih terbatas jumlahnya, sehingga seringkali nasabah menempuh jarak jauh untuk menyetor uang belum lagi jumlah nominal uang yang sedikit turut menurunkan minat nasabah untuk mendatangi mesin setor tunai tersebut. Hal ini menyebabkan para nasabah lebih memilih untuk menyimpan sendiri uangnya di rumah.

Penyimpanan uang secara mandiri di rumah juga mempunyai resiko yang besar khususnya masalah keamanan. Nasabah perlu memiliki tempat penyimpanan yang memadai seperti *MoneyBox Plus*. Alat penyimpanan ini menggunakan mikrokontroler Arduino yang berfungsi untuk memproses output sesuai dengan yang di input kan. Mikrokontroler merupakan

sistem komputer yang berfungsi pada sebuah *chip*, dan juga terdapat sebuah inti prosesor, memori, serta perlengkapan *input/output* [1].

Pembuatan alat *MoneyBox Plus* ini tentunya berkaitan dengan uang yang dimana sistem keamanan sangat diperlukan untuk menjaga uang yang dimiliki [2]. Diharapkan dengan implementasi sistem keamanan tersebut, dapat mengantisipasi dalam menyimpan uang.

Sistem ini dilengkapi dengan alarm jika berhasil masuk ke dalam akses *MoneyBox Plus*. Masalah utama dalam sistem keamanan ini adalah penggunaan *password* untuk akses masuk, karena dengan *password* itulah uang dapat diambil. Implementasi dari *password* tersebut dilakukan secara mengacak terus menerus menggunakan Algoritma *Fisher Yates*. Algoritma ini bekerja dengan melakukan pengacakan atau penyusunan kembali acak dari elemen-elemen dalam himpunan yang berhingga, seperti *array* atau daftar. Proses algoritma ini sangat sederhana dan efisien dalam melakukan pengacakan, dan dapat diimplementasikan

dengan baik dalam pemrograman [3]. Dengan implementasi sistem keamanan ini, diharapkan keamanan terpantau dengan lebih baik, dan memastikan bahwa hanya orang berhak yang memiliki akses [4].

## II. METODE

### 2.1. Tahapan Penelitian

Metode penelitian yang digunakan untuk membuat sistem keamanan pada kotak penyimpanan uang (*Moneybox Plus*) menggunakan metode *prototype* berbasis mikrokontroler Arduino [5]

*Prototype* dalam pengembangan perangkat lunak adalah metode yang digunakan untuk membuat model atau contoh awal dari perangkat lunak yang akan dibuat. *Prototype* merupakan model versi awal atau contoh dari tahapan sistem yang digunakan untuk menyajikan gambaran dari desain yang telah dirancang sebelumnya [6]. Tujuan utama dari pembuatan *prototype* adalah untuk menguji dan memvalidasi konsep, mendapatkan masukan dari pengguna atau pemangku kepentingan, serta mencari dan mengatasi masalah yang mungkin muncul sebelum mengembangkan perangkat lunak secara keseluruhan [7].

Penelitian yang akan dilakukan oleh peneliti meliputi sebagai berikut.



Gambar 1. Diagram Alur Penelitian

### 2.2. Alat dan Bahan Penelitian

Peneliti menggunakan beberapa alat dan komponen untuk penelitian ini, seperti: komponen perangkat keras (*hardware*) dan perangkat lunak (*software*).

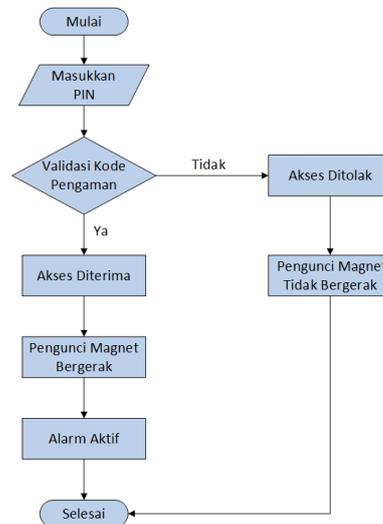
#### 1. Perangkat keras (*hardware*)

- Laptop Lenovo ThinkPad X240
- Mikrokontroler Arduino Mega 2560
- TFT LCD Display Shield
- Solenoid Lock
- Modul Relay
- Buzzer
- Modul GSM SIM800L
- Regulator LM2596
- Kabel USB

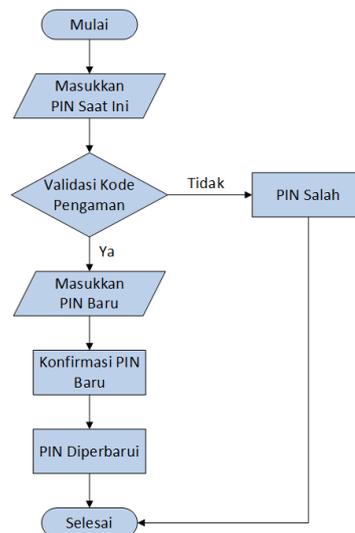
- Power Adaptor 12V
- DC Jack Female
- Kabel Jumper
- Breadboard

#### 2. Perangkat lunak (*software*)

- Sistem Operasi Windows 10
- Arduino IDE
- Microsoft Office Visio 2021
- Adobe Photoshop CS6



Gambar 2. Flowchart Akses Masuk



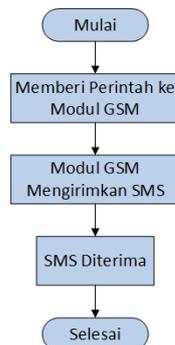
Gambar 3. Flowchart Ubah PIN

### 2.3. Flowchart

*Flowchart* adalah diagram untuk menggambarkan langkah-langkah dan urutan kegiatan dalam sebuah program atau proses [8]. *Flowchart* menggambarkan alur logika atau algoritma dari suatu program dalam bentuk diagram, yang terdiri dari simbol-simbol dan juga panah-panah yang menghubungkannya. *Flowchart* dapat membantu menyelesaikan masalah yang memerlukan penyelidikan khusus dan evaluasi lebih lanjut. *Flowchart* dapat digunakan untuk menyajikan kegiatan manual, kegiatan pemrosesan,

atau keduanya [9]. *Flowchart* adalah alat visual yang serbaguna dan fleksibel untuk menggambarkan berbagai jenis proses atau langkah-langkah dalam bentuk diagram [10]. Adapun *flowchart* yang digunakan pada sistem ini adalah sebagai berikut.

1. Menu Akses Masuk, merupakan menu yang digunakan untuk melakukan akses ke dalam *MoneyBox Plus*, ditunjukkan oleh Gambar 2.
2. Menu Ubah PIN, merupakan menu yang digunakan untuk melakukan perubahan PIN lama menjadi PIN baru seperti pada Gambar 3.
3. Menu Lupa PIN, merupakan menu yang digunakan untuk memberikan perintah terhadap sistem untuk mengirimkan kode pengaman melalui SMS ke nomor tujuan.



Gambar 4. *Flowchart* Lupa PIN

4. Algoritma *Fisher Yates*, digunakan untuk mengacak atau merubah *layout* pada kode pengaman, sehingga *layout* pada kode pengaman tidak selalu menampilkan *layout* yang sama.



Gambar 5. *Flowchart* Algoritma *Fisher Yates*

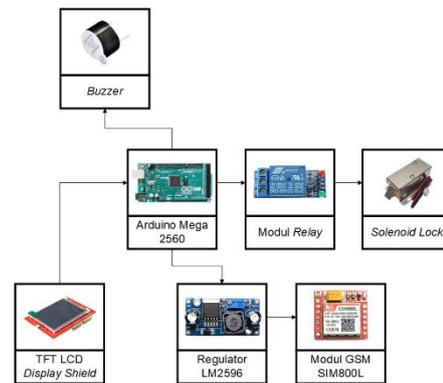
## 2.4. User Interface Design

Tahapan *UI Design (user interface design)* ini merupakan rancangan model sistem keamanan kotak penyimpanan yang bisa diakses oleh pengguna.

Rancangan UI ini berupa tampilan *TFT LCD Display Shield*. Pengguna dapat mengetahui fitur-fitur sistem pada layar tersebut. Untuk mengoperasikan kotak penyimpanan uang berbasis mikrokontroler Arduino (*MoneyBox Plus*) pengguna dapat memilih menu-menu yang tersedia [11].

## 2.5. Modeling Quick Design

Pada tahap *modeling quick design* menggambarkan rancangan umum dari fungsional sistem dalam bentuk blok diagram yang menggambarkan sistem kerja secara keseluruhan [12]. Adapun pemodelan desain cepat yang digunakan ditunjukkan oleh Gambar 6.



Gambar 6. *Modeling Quick Design*

1. *TFT LCD Display Shield* merupakan modul layar yang terdiri dari empat buah LED putih atau biasa disebut dengan *bright backlight* [13]. Digunakan sebagai *input* kode pengaman berupa PIN dengan ukuran layar sebesar 2.4 inci [14].
2. *Arduino Mega 2560* merupakan *board* mikrokontroler berbasis *Arduino Mega 2560* yang memiliki 54 pin digital *input* atau *output* [15]. Digunakan sebagai kontrol utama untuk semua sistem yang ada.
3. *Buzzer* berfungsi untuk menghasilkan bunyi atau suara tertentu ketika diberi tegangan atau sinyal listrik [16]. Digunakan sebagai alarm yang akan bekerja saat berhasil masuk ke dalam akses *MoneyBox Plus*.
4. *Modul Relay* merupakan sebuah rangkaian elektronis sederhana yang menggunakan prinsip elektromagnetisme untuk mengontrol aliran arus listrik pada beban eksternal [17]. *Modul relay* terdiri dari beberapa komponen utama, termasuk saklar, kumparan (medan elektromagnet), dan poros besi (*plunger*) [16]. Digunakan untuk memutus atau menghubungkan *solenoid lock*.
5. *Solenoid Lock* adalah jenis *solenoid* yang dirancang khusus untuk digunakan sebagai mekanisme kunci pada pintu elektronik atau sistem keamanan pintu otomatis [18]. Digunakan sebagai alat pengunci magnet pada *MoneyBox Plus*.
6. *Regulator LM2596* merupakan sebuah modul yang berfungsi sebagai konverter arus searah

(DC) dengan *stepdown*, memiliki rating arus (*current*) sebesar 3A [19]. Digunakan untuk menurunkan tegangan DC dari sumber daya yang lebih tinggi menjadi tegangan yang sesuai dengan Modul GSM SIM800L.

7. Modul GSM SIM800L merupakan komponen yang digunakan dalam proyek-proyek elektronik dan *Internet of Things* (IoT) untuk mengirim dan menerima pesan singkat (SMS), panggilan telepon, serta mengakses data melalui jaringan GSM [20]. Digunakan untuk mengirimkan kode pengaman melalui SMS ke nomor tujuan saat mengalami lupa kode pengaman.

### III. HASIL DAN PEMBAHASAN

#### 3.1. Hasil Implementasi

Sistem keamanan kotak penyimpanan yang dihasilkan adalah *Prototype* kotak penyimpanan uang (*Moneybox Plus*) dengan berbagai sistem. Terdapat berbagai macam modul yang telah diterapkan sebagai kebutuhan untuk perancangan alat ini, sistem keamanan mencakup implementasi terhadap layar LCD, pengunci magnet, alarm, dan modul GSM. Berikut merupakan hasil implementasi pada kotak penyimpanan uang (*Moneybox Plus*).



Gambar 7. Alat MoneyBox Plus

##### 3.1.1. Penggunaan Layar LCD

TFT LCD *display shield* dihubungkan secara langsung pada *board* Arduino Mega 2560. Layar ini menampilkan informasi yang tersedia pada sistem kotak penyimpanan uang (*Moneybox Plus*).



Gambar 8. Penggunaan Layar LCD

##### 3.1.2. Penggunaan Pengunci Magnet

Untuk menjaga keamanan uang yang disimpan digunakan *solenoid lock* yang berfungsi sebagai kunci pintu elektronik. Pengunci magnet ini dapat membuka dan mengunci secara otomatis dengan tambahan modul *relay* yang berfungsi memutuskan atau menghubungkan rangkaian elektronik.

Diperlukan power adaptor 12V untuk menggunakan *solenoid lock* sehingga dapat berfungsi dengan baik. Beberapa modul tersebut dihubungkan pada *board* Arduino Mega 2560 melalui kabel *jumper*, dan

*breadboard*, serta DC *jack female* untuk menghubungkan power adaptor 12V.



Gambar 9. Modul Relay Tidak Aktif

Pada Gambar 9. menjelaskan bahwa ketika tidak ada akses masuk ke dalam sistem atau melakukan kesalahan saat inputan kode pengaman akses masuk, maka modul *relay* tidak aktif dan *solenoid lock* tetap dalam keadaan terkunci.



Gambar 10. Modul Relay Aktif

Sedangkan pada Gambar 10. menunjukkan cara membuka kotak penyimpanan *Moneybox Plus*. Untuk membuka kotak dengan cara memasukkan kode pengaman. Sistem akan mencocokkan kode pengaman tersebut dengan PIN yang tersimpan dalam sistem. Bila kode pengaman sesuai maka modul *relay* aktif dan *solenoid lock* dalam keadaan terbuka.

##### 3.1.3. Penggunaan Alarm

Penggunaan alarm pada kotak penyimpanan uang (*Moneybox Plus*) menggunakan *buzzer* yang berfungsi untuk mengetahui apakah ada yang melakukan akses masuk ke dalam sistem atau tidak dan juga saat melakukan lupa PIN pada sistem. Modul tersebut dihubungkan pada *board* Arduino Mega 2560 melalui kabel *jumper*, dan *breadboard*.



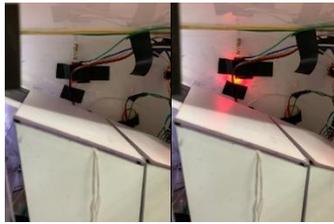
Gambar 11. Penggunaan Buzzer

##### 3.1.4. Penggunaan Modul GSM

Pada kotak penyimpanan uang (*Moneybox Plus*), Modul GSM SIM800L dipergunakan untuk mengirim kode pengaman yang tersimpan di dalam sistem ke nomor tujuan yang telah ditentukan berupa SMS. Saat menggunakan Modul GSM SIM800L diperlukan

modul tambahan seperti regulator LM2596 yang berfungsi untuk menyesuaikan tegangan yang dibutuhkan Modul GSM SIM800L, sehingga dapat berfungsi dengan baik.

Diperlukan power adaptor 12V untuk menggunakan Modul GSM SIM800L melalui regulator LM2596 terlebih dahulu yang dimana tegangannya telah diatur sesuai dengan yang dibutuhkan Modul GSM SIM800L. Beberapa modul tersebut dihubungkan pada *board* Arduino Mega 2560 melalui kabel *jumper*, dan *breadboard*, serta *DC jack female* untuk menghubungkan power adaptor 12V.



Gambar 12. Penggunaan Modul GSM SIM800L

Pada Gambar 12. menjelaskan bahwa ketika modul menerima sinyal, lampu indikator akan berkedip secara perlahan. Sedangkan Saat modul tidak menerima sinyal, lampu indikator akan berkedip secara cepat.

### 3.2. Pengujian Sistem

Pengujian sistem bertujuan untuk mengetahui apakah sistem keamanan Moneybox Plus yang telah dibuat dapat berfungsi dengan baik. Terdapat beberapa sistem yang akan dilakukan pengujian, termasuk beberapa menu pada sistem keamanan antara lain akses masuk, ubah PIN, dan lupa PIN. Sebelum tampilan pilihan menu pada sistem, terdapat menu utama yang digunakan sebagai tampilan awal pada kotak penyimpanan uang (*MoneyBox Plus*).



Gambar 13. Tampilan Menu Utama

#### 3.2.1. Menu Akses Masuk

Menu utama ini merupakan pintu masuk ke dalam sistem. Layar LCD menampilkan menu utama sistem keamanan *Moneybox Plus* seperti yang ditunjukkan oleh Gambar 13. Kotak penyimpanan uang ini dilengkapi sistem keamanan yang hanya dapat diakses oleh orang yang berhak dan mempunyai kode akses yang benar. Uji coba menu akses masuk ini adalah sebagai berikut.

1. Langkah Pertama adalah masuk ke menu utama sistem kemudian pilih menu akses masuk. Selanjutnya masukkan PIN atau kode pengaman

yang sesuai dengan PIN dengan kode pengaman yang tersimpan dalam sistem, seperti yang ditunjukkan oleh Gambar 14.



Gambar 14. Tampilan Menu Akses Masuk

2. Jika PIN atau kode pengaman yang dimasukkan benar maka layar menampilkan pemberitahuan Akses diterima seperti yang ditunjukkan oleh Gambar 15 dan diikuti dengan terbukanya kunci kotak penyimpanan.



Gambar 15. Tampilan Akses Masuk Diterima

3. Sedangkan jika PIN atau kode pengaman yang dimasukkan tidak sesuai maka layar akan menampilkan pesan akses ditolak, seperti pada Gambar 16, dan kotak penyimpanan uang (*MoneyBox Plus*) tetap terkunci.



Gambar 16. Tampilan Akses Masuk Ditolak

#### 3.2.2. Menu Ubah PIN

Menu ubah PIN merupakan menu yang digunakan untuk melakukan perubahan PIN atau kode pengaman. Sebelum melakukan perubahan PIN lama menjadi PIN baru, diharuskan untuk memasukkan PIN lama terlebih dahulu sebagai verifikasi. Adapun pengujian menu ubah PIN ini adalah sebagai berikut.

1. Buka menu utama dan pilih menu ubah PIN. Selanjutnya masukkan PIN atau kode pengaman lama sebagai verifikasi untuk melakukan perubahan PIN.



Gambar 17. Tampilan Menu Ubah PIN

2. Bila PIN atau kode pengaman yang dimasukkan tidak sesuai dengan PIN atau kode pengaman yang lama, maka akses ditolak dan layar menampilkan pesan PIN Salah, seperti pada Gambar 18.



Gambar 18. Tampilan PIN Salah

3. Sedangkan bila PIN atau kode pengaman yang dimasukkan benar, maka layar akan menampilkan pesan “Masukkan PIN baru”. Kemudian masukkan PIN atau kode pengaman baru.



Gambar 19. Tampilan Ubah PIN Baru

4. Setelah memasukkan PIN atau kode pengaman yang baru sesuai dengan keinginan, maka lakukan konfirmasi sehingga PIN akan diperbarui dan dapat tersimpan ke dalam sistem.



Gambar 20. Tampilan PIN Diperbarui

### 3.2.3. Menu Lupa PIN

Menu lupa PIN merupakan menu yang digunakan untuk melakukan perintah terhadap sistem untuk mengirimkan kode pengaman lewat SMS ke nomor tujuan ketika mengalami lupa PIN atau kode pengaman. Adapun pengujian menu lupa PIN ini adalah sebagai berikut.

1. Setelah masuk ke menu lupa PIN, secara otomatis melakukan perintah terhadap sistem untuk

mengirimkan kode pengaman lewat SMS ke nomor tujuan yang telah ditentukan.



Gambar 21. Tampilan SMS Telah Dikirim



Gambar 22. Tampilan SMS Telah Diterima

### 3.3. Hasil Penelitian

Hasil penelitian disini merupakan hasil dari beberapa pengujian fitur yang terdapat pada sistem keamanan yang diantaranya yaitu, perubahan *layout* pada kode pengaman, melakukan pengujian dengan Fungsi *random()*, dan Fungsi *randomSeed()*, serta melakukan pengujian pada fitur lupa kode pengaman.

#### 3.3.1. Layout Kode Pengaman

Dari hasil penelitian yang dilakukan, peneliti membuat tampilan awal pada kode pengaman ini secara berurutan yang dapat dilihat pada gambar berikut.

```
62 String symbol[4][3] = {
63     { "1", "2", "3"},
64     { "4", "5", "6"},
65     { "7", "8", "9"},
66     { "x", "0", "OK"}
67 };
```

Gambar 23. Source Code Layout Kode Pengaman



Gambar 24. Layout Kode Pengaman

Kemudian dilakukan untuk merubah atau mengacak *layout* pada kode pengaman, sehingga *layout* pada kode pengaman tidak selalu menampilkan *layout* yang sama yaitu dengan menggunakan Algoritma *Fisher Yates*. Pada penelitian ini, Algoritma *Fisher Yates* diimplementasikan menggunakan Fungsi yang terdapat pada program Arduino yaitu *random()*. Fungsi *random()* merupakan Fungsi yang digunakan untuk menghasilkan nilai secara acak. Hasil perubahan atau

pengacakan *layout* pada kode pengaman dapat dilihat pada gambar berikut.

```

527 void random_BoxButtons()
528 {
529     for (int j=0; j<3; j++){
530         for (int i=0; i<3; i++){
531             int j1 = random(3);
532             int i1 = random(3);
533
534             String temp = symbol[j][i];
535             symbol[j][i] = symbol[j1][i1];
536             symbol[j1][i1] = temp;
537         }
538     }
539 }
    
```

Gambar 25. Source Code Layout Kode Pengaman Acak



Gambar 26. Layout Kode Pengaman Acak

Pada Gambar 25. terdapat perulangan kurang dari 3 pada variabel *j* dan variabel *i* yang bertujuan untuk melakukan perubahan atau pengacakan dari variabel *symbol*[4][3] menjadi *symbol*[3][3]. Kemudian Fungsi *random()* tersebut diberikan nilai 3 supaya dapat menghasilkan perubahan atau pengacakan yang maksimal dan sesuai dengan variabel *symbol*. Sehingga *layout* pada kode pengaman dapat menghasilkan perubahan atau pengacakan hanya dari angka 1 sampai dengan 9.

### 3.3.2. Pengujian Fungsi $random() \leq 3$

Setelah melakukan perubahan atau pengacakan *layout* pada kode pengaman, peneliti melakukan pengujian Fungsi *random()* yang diberikan nilai kurang dari atau sama dengan 3 sebanyak 3 kali pengujian. Hasil pengujian ini dapat dilihat pada tabel berikut.

Tabel 1. Pengujian Fungsi  $random() \leq 3$

Skenario Pengujian	Hasil Dari Pengujian		
	Memanggil Fungsi <i>random()</i> ke-1	Memanggil Fungsi <i>random()</i> ke-2	Memanggil Fungsi <i>random()</i> ke-3
<i>random</i> (1) <i>random</i> (1)			
<i>random</i> (1) <i>random</i> (2)			
<i>random</i> (1) <i>random</i> (3)			

Skenario Pengujian	Hasil Dari Pengujian		
	Memanggil Fungsi <i>random()</i> ke-1	Memanggil Fungsi <i>random()</i> ke-2	Memanggil Fungsi <i>random()</i> ke-3
<i>random</i> (2) <i>random</i> (1)			
<i>random</i> (2) <i>random</i> (2)			
<i>random</i> (2) <i>random</i> (3)			
<i>random</i> (3) <i>random</i> (1)			
<i>random</i> (3) <i>random</i> (2)			
<i>random</i> (3) <i>random</i> (3)			

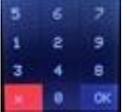
Berdasarkan hasil pengujian pada Tabel 1. dapat disimpulkan bahwa semakin kecil nilai yang diberikan pada Fungsi *random()*, maka semakin minim juga untuk menghasilkan perubahan atau pengacakan *layout* pada kode pengaman. Jika semakin besar nilai yang diberikan pada Fungsi *random()*, maka dapat menghasilkan perubahan atau pengacakan *layout* pada kode pengaman secara maksimal.

### 3.3.3. Pengujian Fungsi $random() > 3$

Setelah melakukan pengujian Fungsi  $random() \leq 3$ , peneliti melakukan pengujian Fungsi *random()* yang diberikan nilai lebih dari 3 sebanyak 3 kali pengujian. Pengujian ini dilakukan untuk mengetahui apakah terdapat kesalahan (*error*) atau tidak saat menghasilkan perubahan atau pengacakan *layout* pada kode pengaman. Hasil pengujian ini dapat dilihat pada tabel berikut.

Tabel 2. Pengujian Fungsi  $random() > 3$

Skenario Pengujian	Hasil Dari Pengujian		
	Memanggil Fungsi <i>random()</i> ke-1	Memanggil Fungsi <i>random()</i> ke-2	Memanggil Fungsi <i>random()</i> ke-3
<i>random</i> (4) <i>random</i> (3)			

Skenario Pengujian	Hasil Dari Pengujian		
	Memanggil Fungsi <i>random()</i> ke-1	Memanggil Fungsi <i>random()</i> ke-2	Memanggil Fungsi <i>random()</i> ke-3
<i>random(3)</i> <i>random(4)</i>			
<i>random(4)</i> <i>random(4)</i>			
<i>random(5)</i> <i>random(5)</i>			

Berdasarkan hasil pengujian pada Tabel 2. dapat disimpulkan bahwa Fungsi *random()* yang diberikan nilai lebih dari 3 terdapat kesalahan (*error*) saat menghasilkan perubahan atau pengacakan *layout* pada kode pengaman. Kesalahan (*error*) terjadi karena nilai yang diberikan pada Fungsi *random()* lebih dari variabel *symbol[3][3]* yang dapat dilihat pada Gambar 25. sehingga menyebabkan perubahan atau pengacakan *layout* pada kode pengaman menjadi tidak sesuai dengan yang diharapkan.

### 3.3.4. Pengujian Fungsi *randomSeed()*

Setelah melakukan beberapa pengujian Fungsi *random()*, peneliti melakukan pengujian Fungsi lain yang terdapat pada program Arduino yaitu *randomSeed()*. Penggunaan Fungsi *randomSeed()* dapat dilihat pada gambar berikut.

```

106 void setup()
107 {
108   Serial.begin(9600); //Use serial monitor for debugging
109   randomSeed(analogRead(1023));

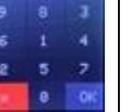
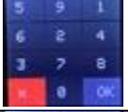
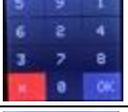
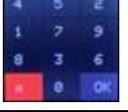
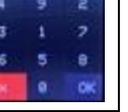
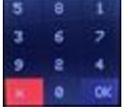
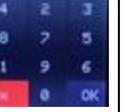
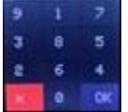
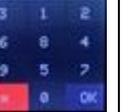
```

Gambar 27. Penggunaan Fungsi *randomSeed()*

Fungsi *randomSeed()* menggunakan hasil dari Fungsi *analogRead()* pada pin analog yang tidak terhubung ke *board* mikrokontroler, sehingga diperoleh nilai secara acak. Fungsi *randomSeed()* dapat dibidang sebagai Fungsi tambahan dari Fungsi *random()* karena dengan menggunakan Fungsi *randomSeed()* akan menghasilkan nilai secara acak terus menerus meskipun sistem telah diatur ulang (*reset*).

Berbeda jika hanya menggunakan Fungsi *random()* saja tanpa menggunakan Fungsi *randomSeed()* yang dimana akan menghasilkan nilai secara acak, tetapi saat sistem telah diatur ulang (*reset*) maka perubahan atau pengacakan tersebut akan sama seperti sebelumnya. Hasil pengujian ini dapat dilihat pada tabel berikut.

Tabel 3. Pengujian Fungsi *randomSeed()*

Skenario Pengujian	Hasil Dari Pengujian		
	Memanggil Fungsi <i>random()</i> ke-1	Memanggil Fungsi <i>random()</i> ke-2	Memanggil Fungsi <i>random()</i> ke-3
Tanpa Fungsi <i>randomSeed()</i> ke-1			
Tanpa Fungsi <i>randomSeed()</i> ke-2			
Tanpa Fungsi <i>randomSeed()</i> ke-3			
Menggunakan Fungsi <i>randomSeed()</i> ke-1			
Menggunakan Fungsi <i>randomSeed()</i> ke-2			
Menggunakan Fungsi <i>randomSeed()</i> ke-3			

Berdasarkan hasil pengujian pada Tabel 3. dapat disimpulkan bahwa memanggil Fungsi *random()* tanpa menggunakan Fungsi *randomSeed()* dapat menghasilkan nilai secara acak, tetapi perubahan atau pengacakan tersebut menjadi sama seperti sebelumnya setelah diatur ulang (*reset*) sebanyak 3 kali.

Sedangkan jika memanggil Fungsi *random()* dengan menggunakan Fungsi *randomSeed()* dapat menghasilkan nilai secara acak terus menerus meskipun telah diatur ulang (*reset*) sebanyak 3 kali.

### 3.3.5. Pengujian Fitur Lupa Kode Pengaman

Pengujian ini dilakukan untuk mengetahui apakah PIN atau kode pengaman yang tersimpan di dalam sistem dapat terkirim dengan baik dan sesuai lewat SMS ke nomor tujuan. Pengujian ini dilakukan sebanyak 3 kali dan setiap pengujian tersebut dilakukan perubahan PIN atau kode pengaman. Hasil pengujian ini dapat dilihat pada gambar berikut.



Gambar 28. Perubahan PIN ke-1



Gambar 29. Pengujian Lupa PIN ke-1



Gambar 30. Perubahan PIN ke-2



Gambar 31. Pengujian Lupa PIN ke-2



Gambar 32. Perubahan PIN ke-3



Gambar 33. Pengujian Lupa PIN ke-3

Berdasarkan hasil pengujian pada Gambar 28. sampai dengan Gambar 33. dapat disimpulkan bahwa fitur Lupa PIN atau kode pengaman pada sistem ini berjalan sesuai dengan yang diharapkan, yang dimana juga dilakukan perubahan PIN atau kode pengaman pada sistem sebanyak 3 kali. Kode pengaman pada sistem yang telah dirubah juga dapat terkirim dengan baik dan sesuai lewat SMS ke nomor tujuan yang telah ditentukan.

#### IV. PENUTUP

##### 4.1. Kesimpulan

Dari hasil pengujian rancang bangun sistem keamanan pada kotak penyimpanan uang berbasis mikrokontroler

arduino (*MoneyBox Plus*), yaitu dapat disimpulkan sebagai berikut :

1. Pengunci magnet dapat berfungsi dengan baik dan sesuai dengan kondisi modul *relay* saat menerima proses dari sistem.
2. *Layout* pada kode pengaman akan selalu mengalami pengacakan atau perubahan, kecuali sistem telah diatur ulang (*reset*).
3. Notifikasi SMS dapat bekerja dengan baik dan digunakan sebagai notifikasi saat menerima kode pengaman dari sistem.

##### 4.2. Saran

Pengembangan sistem keamanan kotak penyimpanan uang berbasis mikrokontroler arduino (*MoneyBox Plus*) supaya alat atau sistem dapat dikembangkan menjadi lebih baik ke depannya, yaitu sebagai berikut :

1. Lebih ditingkatkan untuk pengacakan atau perubahan *layout* pada kode pengaman, sehingga semua angka dari 0 – 9 dapat merubah atau mengacak dengan lebih baik.
2. Ditambahkan fitur untuk mengubah nomor tujuan yang digunakan untuk menerima kode pengaman saat mengalami lupa PIN pada menu utama, sehingga dapat memudahkan pengguna untuk melakukan perubahan langsung tanpa melalui *software* Arduino IDE.

#### DAFTAR PUSTAKA

- [1] A. Virginia, P. Handoko, and H. Hermawan, "Rancang Bangun Sistem Kontrol Lampu Berbasis Arduino Mega 2560," vol. 5, no. 2, 2018.
- [2] R. S. Brahmana, "Penerapan Keamanan Keuangan Dalam Aplikasi E-Money," *J. Manaj. dan Inov.*, vol. 3, no. 1, pp. 46–52, 2020, doi: 10.15642/manova.v3i1.201.
- [3] M. Hasbiyalloh and D. A. Jakaria, "APLIKASI PENJUALAN BARANG PERLENGKAPAN HAND PHONE DI ZILDAN CELL SINGAPARNA KABUPATEN TASIKMALAYA," *JUMANTAKA*, vol. 1, no. 1, pp. 61–70, 2018.
- [4] M. Akram, N. Kurniati, and Salim. Yulita, "Penerapan Algoritma Fisher Yates Shuffle Pada Aplikasi TOEFL Preparation Berbasis Web," *J. Buana Inform.*, vol. 11, no. 2, pp. 112–122, 2020, doi: 10.24002/jbi.v11i2.3622.
- [5] D. M. Santika, F. Fauziah, and B. Rahman, "Algoritma Fisher Yates Pada Aplikasi Pembelajaran Teknik Pengambilan Gambar Menggunakan Kamera DSLR Berbasis Android," *J. Comput. Syst. Informatics*, vol. 4, no. 4, pp. 789–796, 2023, doi:

- 10.47065/josyc.v4i4.3867.
- [6] Tijaniyah and Sabda Alam Arzenda, "Rancang Bangun Prototype Alat Pengusir Tikus Dengan Pemanfaatan Gelombang Ultrasonik Berbasis Internet Of Things," *J. JEETech*, vol. 3, no. 2, pp. 57–63, 2022, doi: 10.48056/jeetech.v3i2.194.
- [7] E. W. Fridayanthie and T. Tsabitah, "Penerapan Metode Prototype Pada Perancangan Sistem Informasi Penggajian Karyawan ( Persis Gawan ) Berbasis Web," vol. 23, no. 2, pp. 151–157, 2021.
- [8] P. Zhang, W. Dou, and H. Liu, "Hierarchical data structures for flowchart," *Sci. Rep.*, vol. 13, no. 1, pp. 1–15, 2023, doi: 10.1038/s41598-023-31968-z.
- [9] W. Wang, "Flowchart," in *A Beginner's Guide to Informatics and Artificial Intelligence*, 2024, pp. 13–19. doi: 10.1007/978-981-97-1477-3\_2.
- [10] I. Budiman, S. Saori, N. A. Anwar, F. Fitriani, and M. Y. Pangestu, "ANALISIS PENGENDALIAN MUTU DI BIDANG INDUSTRI MAKANAN (Studi Kasus: UMKM Mochi Kaswari Lampion Kota Sukabumi)," vol. 1, no. 10, 2021.
- [11] K. Kurniati, "Penerapan Metode Prototype Pada Perancangan Sistem Pengarsipan Dokumen Kantor Kecamatan Lais," *J. Softw. Eng. Ampera*, vol. 2, no. 1, pp. 16–27, 2021, doi: 10.51519/journalsea.v2i1.89.
- [12] U. Ependi and M. Nasir, "Peningkatan Kapasitas Penggunaan Teknologi Informasi Dan Komunikasi Dalam Pelaksanaan ANBK di," vol. 2, no. 1, pp. 1–11, 2024.
- [13] M. Rijali and R. . Khana, "BANGUN SISTEM PEMANTAUAN DAN PENGENDALIAN CAIRAN INFUS MELALUI DISPLAY KONTROL DAN APLIKASI MOBILE DI MASA PANDEMIC COVID-19," vol. 5, no. 1, pp. 1–21, 1945.
- [14] Hendri, D. Meisak, and S. R. Agustini, "Penerapan Metode Prototype Pada Perancangan Sistem Informasi Penjualan Mediatama Solusindo Jambi," *STORAGE J. Ilm. Tek. dan Ilmu Komput.*, vol. 1, no. 4, pp. 1–11, 2022, doi: 10.55123/storage.v1i4.1066.
- [15] J. W. Simatupang, B. Prasetyo, and M. I. A. Galina, "Prototipe Mesin Penjual Air Mineral Otomatis berbasis Arduino Mega 2560 dan RFID-RC522," vol. 10, no. 2, pp. 484–499, 2022.
- [16] R. Rahardi, D. Triyanto, and Suhardi, "PERANCANGAN SISTEM KEAMANAN SEPEDA MOTOR DENGAN SENSOR FINGERPRINT, SMS GATEWAY, DAN GPS TRACKER BERBASIS ARDUINO DENGAN INTERFACE WEBSITE," vol. 06, no. 03, pp. 118–127, 2018.
- [17] M. Danny, "Keywords : Android Applications , Arduino UNO ( microcontroller ATmega328 )," *J. Teknol. Pelita Bangsa*, vol. 9, pp. 85–90, 2018.
- [18] R. Suwartika and G. Sembada, "Perancangan Sistem Keamanan Menggunakan Solenoid Door Lock Berbasis Arduino Uno pada Pintu Laboratorium di PT. XYZ," *J. E-KOMTEK*, vol. 4, no. 1, pp. 62–74, 2020.
- [19] F. A. Lutfi, "PERANCANGAN PURWARUPA SISTEM PERINGATAN KEBOCORAN GAS LIQUEFIED PETROLEUM GAS ( LPG ) PROGRAM STUDI TEKNIK ELEKTRO PERANCANGAN PURWARUPA SISTEM PERINGATAN KEBOCORAN GAS LIQUEFIED PETROLEUM GAS ( LPG ) PERANCANGAN PURWARUPA SISTEM PERINGATAN KEBOCORAN GAS," pp. 1–7, 2018.
- [20] I. W. E. Prastia *et al.*, "Rancang Bangun Monitoring Level Muka Air Tanah Di Perkebunan Lahan Gambut Menggunakan SMS Sebagai Pengirim Informasi Data Berbasis Mikrokontroler," vol. 6, no. 1, pp. 62–68, 2019.