Jurnal Teknik dan Science
JTS Vol 3 No. 3 Oktober 2024 | P-ISSN: 2828-7002 E-ISSN: 2828-6871, Page 01-13

IMPLEMENTASI KONSEP BLOCKCHAIN MENGGUNAKAN PYTHON DAN INTEGRASI DENGAN APLIKASI WEB: STUDI KASUS FLASK DAN HASHLIB

Aulia Intan Shafira[a], Dwi Kartinah[b]
[a]Teknik Informatika, aulins12.ais@gmail.com, Universitas Gunadarma
[b]Program Studi Sistem Informasi, dwi_kartina@staff.gunadarma.ac.id, Universitas Gunadarma

ABSTRACT

The rapid development of the Internet has influenced various fields, from education to business, with more than 5 billion users worldwide. The evolution of web technology, from Web 1.0 to Web 3.0, represents significant changes in the way information is presented and how users interact with data. Web 1.0 only allows one-way access to information, while Web 2.0 allows dynamic interaction between users and content through social media and blogs. Currently, Web 3.0 brings the concept of decentralization by utilizing blockchain technology which allows users to have complete control over their data. This research aims to develop and test Web 3.0-based applications that use blockchain technology for safe, efficient and decentralized data storage and management. Although the prototype developed in this research still uses centralized data storage, the actual decentralized concept will be applied in further development. Storing data on blockchain offers the benefits of greater security and transparency, albeit at a greater cost. This application is expected to be able to address the urgent need for secure and decentralized data storage, especially for data that requires a high level of trust and integrity, such as medical records, financial data, and inventory in warehouses.

Keywords: blockchain, World Wide Web

ABSTRAK

Perkembangan Internet yang pesat telah mempengaruhi berbagai bidang, dari pendidikan hingga bisnis, dengan lebih dari 5 miliar pengguna di seluruh dunia. Evolusi teknologi web, dari Web 1.0 hingga Web 3.0, menunjukkan perubahan signifikan dalam cara informasi disajikan dan bagaimana pengguna berinteraksi dengan data. Web 1.0 hanya memungkinkan akses satu arah terhadap informasi, sedangkan Web 2.0 memungkinkan interaksi dinamis antara pengguna dan konten melalui media sosial dan blog. Saat ini, Web 3.0 membawa konsep desentralisasi dengan memanfaatkan teknologi blockchain yang memungkinkan pengguna memiliki kendali penuh atas data mereka. Penelitian ini bertujuan untuk mengembangkan dan menguji aplikasi berbasis Web 3.0 yang menggunakan teknologi blockchain untuk penyimpanan dan pengelolaan data secara aman, efisien, dan terdesentralisasi. Meski prototipe yang dikembangkan dalam penelitian ini masih menggunakan penyimpanan data terpusat, konsep desentralisasi yang sesungguhnya akan diterapkan pada pengembangan lanjutan. Penyimpanan data dalam blockchain menawarkan keuntungan berupa keamanan dan transparansi yang lebih tinggi, meskipun dengan biaya yang lebih besar. Aplikasi ini diharapkan mampu mengatasi kebutuhan mendesak akan penyimpanan data yang aman dan terdesentralisasi, terutama untuk data yang membutuhkan tingkat kepercayaan dan integritas tinggi, seperti catatan medis, data keuangan, dan persediaan barang di gudang.

Kata Kunci: blockchain, World Wide Web

1. PENDAHULUAN

Hingga kini Internet berkembang dengan sangat pesat. Bahkan, hampir semua bidang membutuhkan kehadiran Internet, mulai dari pendidikan hingga bisnis. Dikutip dari Forbes, saat ini pengguna Internet sudah mencapai lebih dari 5 miliar, setidaknya sekitar 63 persen dari populasi dunia. Semakin hari, internet semakin berevolusi.Perkembangan teknologi web dari Web 1.0 hingga Web 3.0 menunjukkan evolusi yang signifikan

dalam cara informasi disajikan dan bagaimana pengguna berinteraksi dengan data. Web 1.0, sebagai generasi awal dari World Wide Web, hanya menyediakan halaman informasi statis di mana pengguna hanya bisa mengakses konten secara satu arah tanpa adanya interaksi lebih lanjut. Seiring berjalannya waktu, Web 2.0 muncul dan memungkinkan interaksi yang lebih dinamis antara pengguna dan konten melalui platform seperti media sosial dan blog, meskipun kendali atas data pengguna masih berada di tangan platform sentral. Kini, Web 3.0 atau web semantik hadir dengan konsep desentralisasi, yang memberikan pengguna kendali penuh atas data mereka melalui teknologi blockchain.

Penelitian ini dimaksudkan untuk mengembangkan dan menguji aplikasi berbasis Web 3.0 yang memanfaatkan teknologi blockchain untuk menyimpan dan mengelola data dengan cara yang aman, efisien, dan terdesentralisasi. Meskipun pada penelitian ini prototipe yang dikembangkan masih menggunakan penyimpanan data terpusat, konsep desentralisasi yang sesungguhnya akan diterapkan pada pengembangan lanjutan. Penyimpanan data dalam blockchain memang memerlukan biaya yang lebih besar, tetapi memberikan keuntungan berupa keamanan dan transparansi yang lebih tinggi.

Aplikasi ini dirancang untuk mengatasi masalah mendesak terkait penyimpanan data dalam ekosistem Web 3.0. Dalam lingkungan digital saat ini, banyak aplikasi yang membutuhkan penyimpanan data yang aman dan terdesentralisasi, terutama dalam situasi di mana kepercayaan dan integritas data adalah prioritas. Contoh kasus seperti pengelolaan catatan medis, data keuangan,data sensitif lainnya yang memerlukan perlindungan tinggi, atau persediaan barang di gudang. Semua contoh tersebut dapat direpresentasikan dalam bentuk blockchain. Ini menunjukkan bagaimana teknologi blockchain dapat memberikan solusi yang lebih baik dalam pengelolaan data di era Web 3.0.

2. TINJAUAN PUSTAKA

Perancangan Blockchain, Database, dan Sistem Aplikasi Web

Perancangan sistem melibatkan beberapa tahap yang bertujuan untuk menghasilkan blueprint atau cetak biru sistem yang akan diimplementasikan. Tahap-tahap perancangan meliputi:
Sistem yang dirancang terdiri dari tiga komponen utama:

1. **Blockchain:** Implementasi inti dari blockchain yang mencakup pembuatan blok, penambangan (minig), dan validasi blockchain. Perancangan ini menggunakan library *hashlib* pada Python untuk membangun prototipe blockchain. Blockchain ini berfungsi sebagai mekanisme pencatatan transaksi yang aman dan terverifikasi.
2. **Database:** Penyimpanan data transaksi dilakukan menggunakan MySQL. Database ini dirancang untuk menyimpan data transaksi secara efisien dan dapat diakses dengan mudah untuk keperluan pengujian dan pengembangan.
3. **Web Aplication:** Aplikasi web berbasis Flask yang memungkinkan pengguna untuk berinteraksi dengan blockchain, melakukan registrasi, login, transaksi, dan melihat dashboard. Pada perancangan ini membuat alur proses aplikasi dan merancang *interface*.

Komponen Blockchain

Dalam pengembangan sistem blockchain untuk aplikasi transaksi digital ini, terdapat beberapa komponen utama yang digunakan:

1. Library *hashlib*: Library ini digunakan untuk menghasilkan hash SHA-256, yang merupakan elemen kriptografi penting dalam blockchain. Hash ini digunakan untuk mengamankan data dalam setiap blok sehingga data tidak dapat diubah tanpa mengubah hash yang terkait.
2. Kelas *Block*: Kelas ini mewakili node dalam blockchain. Setiap blok dalam rantai berisi data transaksi, hash dari blok sebelumnya, nonce (angka yang digunakan sekali dalam proses mining), dan nomor blok. Struktur ini memastikan bahwa setiap blok terhubung secara kriptografis dengan blok sebelumnya, menciptakan rantai yang aman.
3. kelas *Blockchain*: Kelas ini merupakan struktur data yang mengelola rantai blok. Kelas ini bertanggung jawab atas penambangan (mining) blok baru, serta validasi integritas rantai blok. Proses penambangan melibatkan pencarian nonce yang menghasilkan hash sesuai dengan kriteria tertentu, memastikan bahwa setiap blok baru yang ditambahkan ke rantai telah diverifikasi.

Komponen Web Application

Untuk mengintegrasikan blockchain dengan antarmuka pengguna, beberapa komponen utama yang digunakan dalam pengembangan aplikasi web adalah:

1. Framework *Flask*: Digunakan untuk membangun *interface* web yang memungkinkan pengguna untuk berinteraksi dengan blockchain melalui API dan *interface* web yang ramah pengguna. Flask digunakan karena kesederhanaannya dan kemampuannya untuk mengintegrasikan berbagai komponen dengan mudah.

2. Database MySQL: Database ini digunakan untuk menyimpan informasi pengguna dan data lainnya yang diperlukan oleh aplikasi web. Meskipun blockchain menyimpan data transaksi, MySQL digunakan untuk manajemen data pengguna dan informasi lain yang tidak disimpan dalam blockchain.
3. Library 'passlib': Passlib adalah library untuk hashing password pengguna dengan aman. Ini memastikan bahwa password pengguna disimpan dalam bentuk hash yang aman, meningkatkan keamanan sistem secara keseluruhan.

3. METODE PENELITIAN

Metode pada penelitian ini menggunakan metode pengembangan perangkat lunak waterfall yang terdiri dari beberapa tahapan, yaitu:

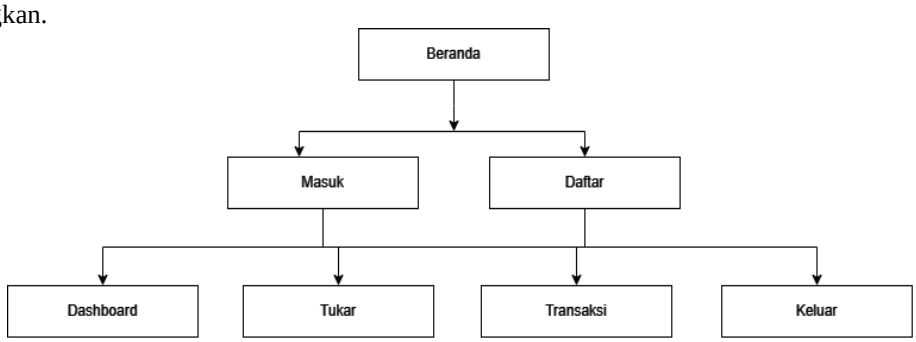
a. Analisis merupakan tahapan untuk menentukan tools pendukung apa saja yang akan digunakan dalam pembuatan aplikasi, mulai dari kebutuhan Kebutuhan Fungsional dan Non-Fungsional.
b. Perancangan blockchain menggunakan library hashlib pada python dan perancangan dari segi sistem aplikasi. Pada tahap ini meliputi perancangan alur proses aplikasi dan rancangan interface aplikasi.
c. Implementasi sistem transaksi dilakukan dengan memanfaatkan API dan sistem blockchain yang telah dirancang sebelumnya, serta diintegrasikan ke dalam aplikasi web untuk memastikan kelancaran dan keamanan proses transaksi.
d. Pengujian sistem berbasis web untuk mengevaluasi fungsionalitas dan efektivitas sistem yang telah diimplementasikan.
e. Evaluasi dan perbaikan sistem berdasarkan hasil pengujian dan evaluasi sebelumnya.

4. PEMBAHASAN

1. Perencanaan

Struktur Navigasi

Struktur navigasi pada sistem aplikasi web ini dirancang untuk memudahkan pengguna dalam mengakses berbagai fitur yang tersedia. Gambar 3.1 menunjukkan struktur navigasi hierarki dari aplikasi web yang telah dikembangkan.



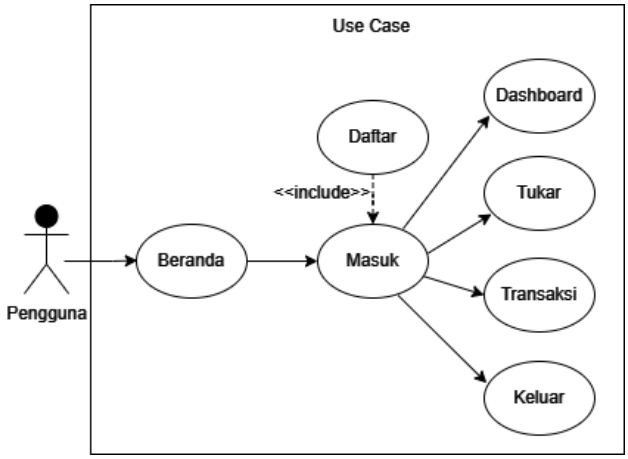
Gambar 4.1 Struktur Navigasi Hierarki

Halaman utama adalah Beranda, yang berfungsi sebagai pintu masuk bagi pengguna. Dari sini, pengguna dapat memilih untuk Masuk atau Daftar. Halaman Masuk digunakan untuk login ke akun, sedangkan halaman Daftar untuk registrasi pengguna baru. Setelah masuk, pengguna diarahkan ke Dashboard, yang merupakan pusat kendali utama aplikasi. Dari Dashboard, pengguna dapat mengakses fitur Tukar untuk pertukaran aset digital dan fitur Transaksi untuk melihat riwayat transaksi. Fitur Keluar memungkinkan pengguna untuk keluar dari akun mereka dengan aman. Struktur navigasi ini dirancang untuk memberikan pengalaman yang mudah dan efisien bagi pengguna dalam mengelola akun dan transaksi mereka.

UML

a. Diagram Use Case

Diagram use case menggambarkan interaksi antara pengguna dan sistem aplikasi web yang dirancang. Pada diagram ini, terdapat beberapa use case yang menjelaskan fungsi-fungsi utama yang dapat diakses oleh pengguna.



Gambar 4.2 Diagram Use Case

Beranda adalah halaman awal yang menampilkan informasi dasar dan navigasi utama. Perancangan Database

Perancangan database merupakan langkah penting dalam pengembangan aplikasi untuk memastikan data dapat disimpan dan diakses dengan efisien. Pada proyek ini, database digunakan untuk menyimpan informasi pengguna yang akan berinteraksi dengan sistem blockchain. Berikut adalah tabel utama yang digunakan dalam perancangan database aplikasi:

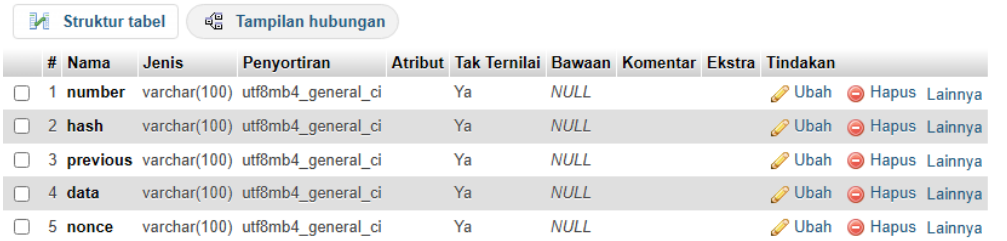
a. Perancangan Tabel Database Blockchain

Tabel blockchain dirancang untuk menyimpan informasi mengenai setiap blok yang ada dalam rantai blockchain. Berikut adalah struktur tabel blockchain beserta deskripsi setiap field:

Table 4.1 Tabel Struktur Tabel Blockchain

Nama field	Type	Length
number	varchar	10
hash	varchar	64
previous	varchar	64
data	varchar	100
nonce	Varchar	15

Setelah melakukan perancangan, implementasikan pada struktur tabel blockchain yang ada pada phpMyAdmin


Struktur tabel Tampilan hubungan

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	number	varchar(100)	utf8mb4_general_ci		Ya	NULL			Ubah Hapus Lainnya
2	hash	varchar(100)	utf8mb4_general_ci		Ya	NULL			Ubah Hapus Lainnya
3	previous	varchar(100)	utf8mb4_general_ci		Ya	NULL			Ubah Hapus Lainnya
4	data	varchar(100)	utf8mb4_general_ci		Ya	NULL			Ubah Hapus Lainnya
5	nonce	varchar(100)	utf8mb4_general_ci		Ya	NULL			Ubah Hapus Lainnya

Gambar 4.3 Tabel blockchain

b. Perancangan Tabel Database Users

Tabel users digunakan untuk menyimpan data pengguna yang terdaftar dalam aplikasi. Berikut adalah struktur tabel users beserta deskripsi setiap field:

Table 4.2 Tabel Struktur Tabel User

Nama field	Type	Length
name	varchar	30
username	varchar	10
email	varchar	25
password	varchar	12

Setelah melakukan perancangan, implementasikan pada struktur tabel users yang ada pada phpMyAdmin

Struktur tabel | Tampilan hubungan

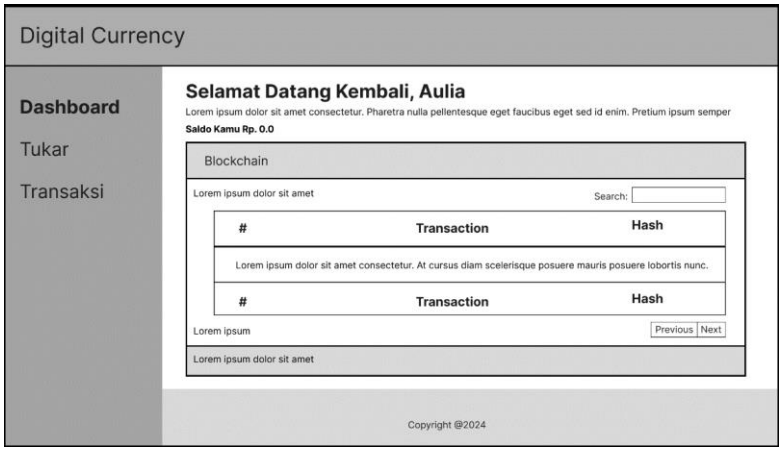
#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	**name**	varchar(30)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
2	**username**	varchar(10)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
3	**email**	varchar(25)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya
4	**password**	varchar(12)	utf8mb4_general_ci		Tidak	*Tidak ada*			Ubah Hapus Lainnya

Gambar 4.4 Tabel users

Perancangan Tampilan Website

a. Rancangan Tampilan Beranda

Tampilan beranda merupakan halaman utama dari sebuah website yang menjadi titik awal interaksi pengguna.



Gambar 4.5 Rancangan Tampilan Website

Implementasi

Implementasi merupakan langkah penting dalam pengembangan sistem, dimana seluruh perancangan dan perencanaan diubah menjadi kode yang dapat dieksekusi. Berikut adalah penjelasan rinci mengenai proses implementasi yang dilakukan dalam proyek ini.

Pembuatan Program

```python
from hashlib import sha256

def updatehash(*args):
    hashing_text = ""; h = sha256()
    for arg in args:
        hashing_text += str(arg)
    h.update(hashing_text.encode('utf-8'))
    return h.hexdigest()

class Block():
    def __init__(self, number=0, previous_hash="0"*64, data=None, nonce=0):
        self.data = data
        self.number = number
        self.previous_hash = previous_hash
        self.nonce = nonce

    def hash(self):
        return updatehash(
            self.number,
            self.previous_hash,
            self.data,
            self.nonce
        )

    def __str__(self):
        return str("Block#: %s\nHash: %s\nPrevious: %s\nData: %s\nNonce: %s\n" %(
            self.number,
            self.hash(),
            self.previous_hash,
            self.data,
            self.nonce
            )
        )
```

Gambar 4.6 Potongan Kode File blockchain.py

Penyusunan Genesis Block

Genesis block adalah blok pertama dalam blockchain dan berfungsi sebagai dasar untuk blok-blok berikutnya.

```
class Blockchain():
    difficulty = 4

    def __init__(self):
        self.chain = []

    def add(self, block):
        self.chain.append(block)

    def mine(self, block):
        try: block.previous_hash = self.chain[-1].hash()
        except IndexError: pass

        while True:
            if block.hash()[:self.difficulty] == "0" * self.difficulty:
                self.add(block); break
            else:
                block.nonce += 1
```

Gambar 4.7 Potongan Kode File blockchain.py

Pembuatan API

Bagian API dari file terdiri dari sejumlah rute (routes) yang ditentukan menggunakan Flask, yang menangani berbagai permintaan HTTP untuk fungsi-fungsi spesifik dalam aplikasi web. Rute-rute ini didefinisikan menggunakan decorator @app.route yang menentukan URL endpoint dan metode HTTP yang diizinkan (GET dan POST).

Halaman Registrasi

Rute untuk halaman registrasi (/register) menangani pendaftaran pengguna baru. Saat permintaan POST diterima dengan data formulir yang valid, data pengguna baru disimpan dalam database, dan sesi pengguna diperbarui untuk mencerminkan status login mereka. Jika pengguna sudah ada, pesan kesalahan dikirimkan kembali.

```
#Registration page
@app.route("/register", methods = ['GET', 'POST'])
def register():
    form = RegisterForm(request.form)
    users = Table("users", "name", "email", "username", "password")

    #if form is submitted
    if request.method == 'POST' and form.validate():
        #collect form data
        username = form.username.data
        email = form.email.data
        name = form.name.data

        #make sure user does not already exist
        if isnewuser(username):
            #add the user to mysql and log them in
            password = sha256_crypt.encrypt(form.password.data)
            users.insert(name,email,username,password)
            log_in_user(username)
            return redirect(url_for('dashboard'))
        else:
            flash('User already exists', 'danger')
            return redirect(url_for('register'))

    return render_template('register.html', form=form)
```

Gambar 4.8 Potongan Kode Halaman Registrasi

Halaman login

Rute login (/login) memproses permintaan masuk pengguna. Pada penerimaan permintaan POST, sistem memverifikasi kredensial pengguna dengan membandingkan kata sandi yang diberikan dengan kata sandi yang disimpan dalam database. Jika berhasil, sesi diperbarui dan pengguna diarahkan ke halaman dashboard. Jika gagal, pesan kesalahan ditampilkan.

```python
#Login page
@app.route("/login", methods = ['GET', 'POST'])
def login():
    #if form is submitted
    if request.method == 'POST':
        #collect form data
        username = request.form['username']
        candidate = request.form['password']

        #access users table to get the user's actual password
        users = Table("users", "name", "email", "username", "password")
        user = users.getone("username", username)
        accPass = user.get('password')

        #if the password cannot be found, the user does not exist
        if accPass is None:
            flash("Username is not found", 'danger')
            return redirect(url_for('login'))
        else:
            #verify that the password entered matches the actual password
            if sha256_crypt.verify(candidate, accPass):
                #log in the user and redirect to Dashboard page
                log_in_user(username)
                flash('You are now logged in.', 'success')
                return redirect(url_for('dashboard'))
            else:
                #if the passwords do not match
                flash("Invalid password", 'danger')
                return redirect(url_for('login'))

    return render_template('login.html')
```

Gambar 4.9 Potongan Kode Halaman Login

1. Halaman Transaksi

Rute transaksi (/transaction) memungkinkan pengguna yang sudah login untuk mengirim uang ke pengguna lain. Endpoint ini dilindungi oleh decorator @is_logged_in yang memastikan hanya pengguna yang sudah login yang dapat mengaksesnya. Ketika permintaan POST diterima, sistem mencoba melaksanakan transaksi dan memberikan umpan balik kepada pengguna mengenai keberhasilan atau kegagalan operasi tersebut.

```python
#Transaction page
@app.route("/transaction", methods = ['GET', 'POST'])
@is_logged_in
def transaction():
    form = SendMoneyForm(request.form)
    balance = get_balance(session.get('username'))

    #if form is submitted
    if request.method == 'POST':
        try:
            #attempt to execute the transaction
            send_money(session.get('username'), form.username.data, form.amount.data)
            flash("Money Sent!", "success")
        except Exception as e:
            flash(str(e), 'danger')

        return redirect(url_for('transaction'))

    return render_template('transaction.html', balance=balance, form=form, page='transaction')
```

Gambar 4.10 Potongan Kode Halaman Transaksi

2. Halaman Beli

Rute pembelian (/buy) mengizinkan pengguna yang sudah login untuk membeli suatu jumlah tertentu, yang melibatkan transfer uang dari akun bank ke akun pengguna. Sama seperti rute transaksi, rute ini juga dilindungi oleh decorator @is_logged_in.

```python
#Buy page
@app.route("/buy", methods = ['GET', 'POST'])
@is_logged_in
def buy():
    form = BuyForm(request.form)
    balance = get_balance(session.get('username'))

    if request.method == 'POST':
        #attempt to buy amount
        try:
            send_money("BANK", session.get('username'), form.amount.data)
            flash("Purchase Successful!", "success")
        except Exception as e:
            flash(str(e), 'danger')

        return redirect(url_for('dashboard'))

    return render_template('buy.html', balance=balance, form=form, page='buy')
```

Gambar 4.11 Potongan Kode Halaman Beli

3. Logout

Rute logout (/logout) menangani permintaan untuk keluar dari sesi pengguna. Setelah menghapus semua data sesi, pengguna diarahkan kembali ke halaman login dengan pesan sukses.

```python
#logout the user. Ends current session
@app.route("/logout")
@is_logged_in
def logout():
    session.clear()
    flash("Logout success", "success")
    return redirect(url_for('login'))
```

Gambar 4.12 Potongan Kode Logout

4. Halaman Dashboard

Rute dashboard (/dashboard) menyajikan informasi mengenai saldo pengguna dan data blockchain yang terkait. Endpoint ini memastikan bahwa pengguna harus login sebelum dapat mengakses data pribadi mereka.

```python
#Dashboard page
@app.route("/dashboard")
@is_logged_in
def dashboard():
    balance = get_balance(session.get('username'))
    blockchain = get_blockchain().chain
    ct = time.strftime("%I:%M %p")
    return render_template('dashboard.html', balance=balance, session=session, ct=ct, blockchain=blockchain, page='dashboard')
```

Gambar 4.13 Potongan Kode Halaman Dashboard

5. Halaman Index

Rute index (/ dan /index) menyajikan halaman utama aplikasi yang tidak memerlukan otentikasi pengguna.

```python
#Index page
@app.route("/")
@app.route("/index")
def index():
    return render_template('index.html')
```

Gambar 4.14 Potongan Kode Halaman index.html

Koneksi Database

Konfigurasi database adalah bagian penting dari aplikasi web berbasis Flask yang menghubungkan aplikasi tersebut dengan database MySQL. Dalam konfigurasi ini, modul flask_mysqldb diimpor dan digunakan untuk menginisialisasi koneksi ke MySQL.

```
from flask_mysqldb import MySQL

app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'penulisan_ilmiah'
app.config['MYSQL_CURSORCLASS'] = 'DictCursor'

mysql = MySQL(app)
```

Gambar 4.15 Koneksi Database

Konfigurasi database pada gambar tersebut adalah bagian dari aplikasi web berbasis Flask yang menghubungkannya dengan MySQL. Pertama, modul flask_mysqldb dan kelas MySQL diimpor. Kemudian, berbagai parameter konfigurasi disetel:

a. MYSQL_HOST diatur ke 'localhost' untuk koneksi ke server MySQL lokal.
b. MYSQL_USER diatur ke 'root', nama pengguna default MySQL.
c. MYSQL_PASSWORD dikosongkan untuk contoh ini.
d. MYSQL_DB diatur ke 'penulisan_ilmiah', nama database yang digunakan.
e. MYSQL_CURSORCLASS diatur ke 'DictCursor' agar hasil query dikembalikan sebagai dictionary.

Setelah itu, objek MySQL diinisialisasi dengan instance aplikasi Flask, mengikat konfigurasi database ke aplikasi sehingga dapat berinteraksi dengan database MySQL secara efisien.

Pembuatan Interface

File dashboard.html adalah bagian dari aplikasi web berbasis Flask yang berfungsi sebagai halaman dashboard pengguna. Template ini menggunakan layout.html sebagai kerangka dasar, memungkinkan konsistensi struktur di seluruh aplikasi.

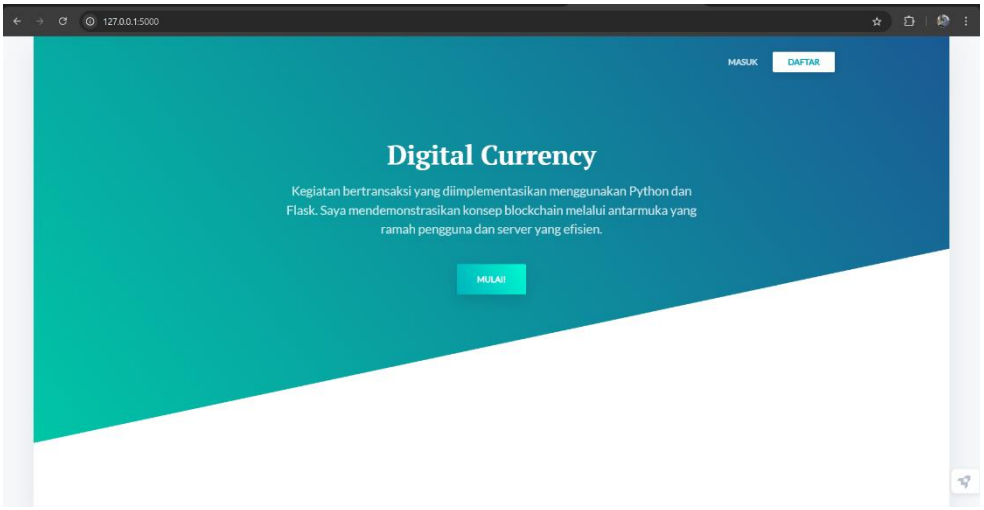
```
{% extends 'layout.html' %} {% block body %}
<h1>Selamat Datang Kembali, {{session.get('name')}}</h1>
{% include 'includes/_messages.html' %}
<p></p>

<p>Saldo Kamu Rp. {{ balance }}</p>

<div class="card mb-3">
    <div class="card-header">
        <i class="fas fa-table"></i>
        Blockchain
    </div>
    <div class="card-body">
        <div class="table-responsive">
            <table class="table table-bordered" id="dataTable" width="100%" cellspacing=
                <thead>
                    <tr>
                        <th>#</th>
                        <th>Transaction</th>
                        <th>Hash</th>
```

Gambar 4.16 Potongan Kode Pembuatan *Interface*

Komponen utama halaman adalah sebuah card Bootstrap yang menampilkan informasi blockchain. Card ini berisi tabel responsif yang menunjukkan nomor blok, data transaksi, dan hash setiap blok dalam blockchain.

Penggunaan Aplikasi

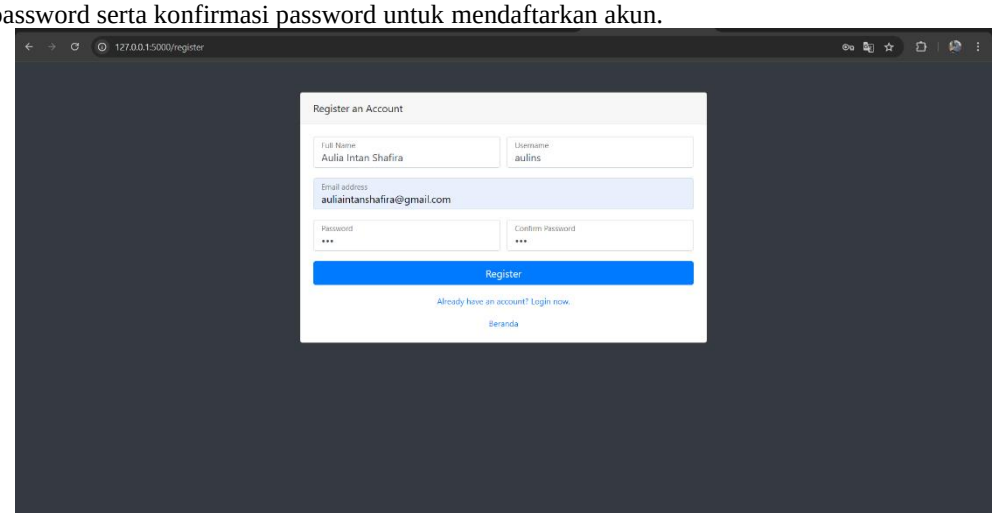
a. Tampilan Awal
   Pada tampilan ini merupakan halaman awal ketika pengguna mengakses sistem. Fitur daftar dan masuk dapat digunakan untuk dapat melanjutkan ke bagian dashboard.



Gambar 4.17 Halaman Utama

b. Fitur Daftar

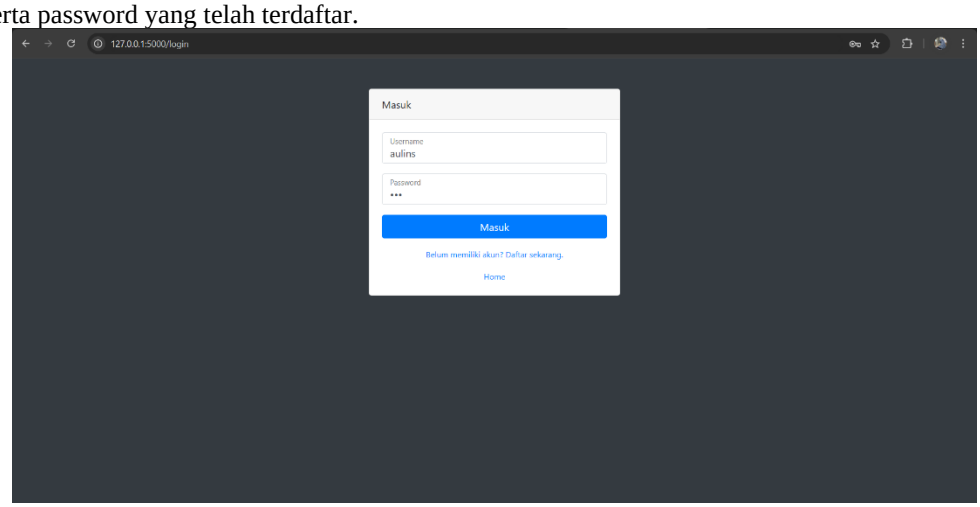
Pengguna dapat mengisikan data yang harus diisi seperti nama lengkap, username, email, password serta konfirmasi password untuk mendaftarkan akun.



Gambar 4.18 Halaman Daftar

c. Fitur Masuk

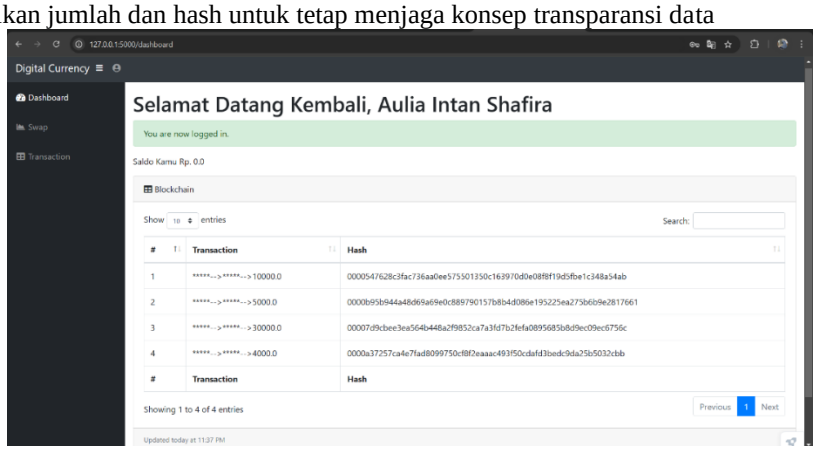
Apabila pengguna telah memiliki akun, fitur masuk dapat digunakan untuk mengisikan username serta password yang telah terdaftar.



Gambar 4.19 Halaman Masuk

d. Dashboard

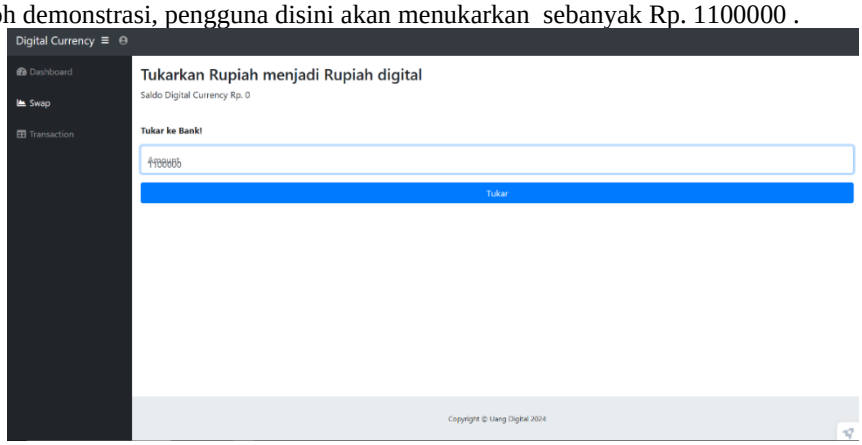
Pada bagian dashboard terlihat bahwa sudah terdapat berbagai transaksi yang sudah terjadi dengan nama disamarkan, karena inti dari transaksi ini yaitu bersifat anonymous, sehingga hanya menampilkan jumlah dan hash untuk tetap menjaga konsep transparansi data



Gambar 4.20 Halaman Dashboard

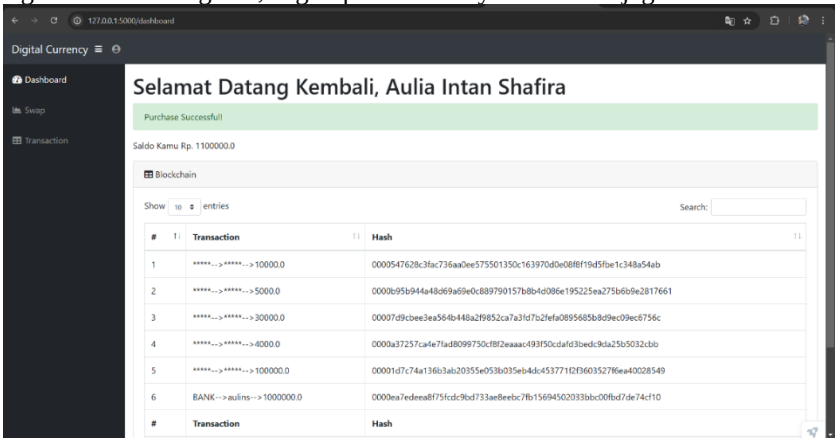
e. Swap

Pada halaman swap, pengguna dapat menukarkan Rupiah nya menjadi Rupiah Digital ke Bank. Saldo awal pengguna akan masih sebanyak Rp. 0 hinga pengguna menukarkan uang nya. Pada contoh demonstrasi, pengguna disini akan menukarkan sebanyak Rp. 1100000 .



Gambar 4.21 Halaman Swap

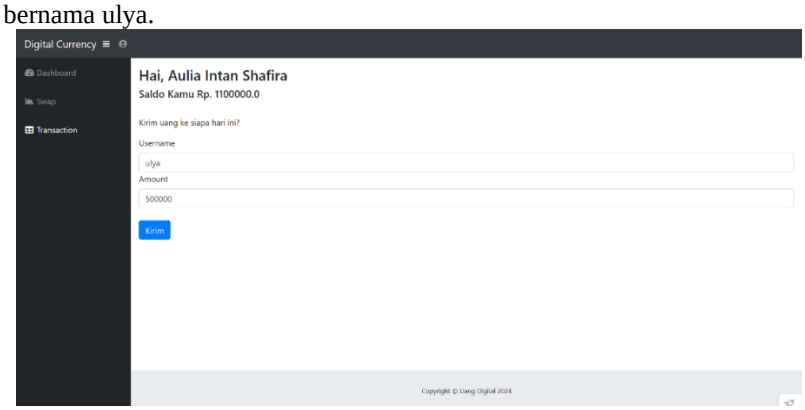
Akan terlihat pada halaman dashboard riwayat transaksi yang dilakukan pengguna seperti pada gambar 3.25 . pengguna hanya akan melihat transaksi diri nya saja dan tidak akan melihat riwayat transaksi yang dilakukan orang lain, begitu pun sebaliknya untuk menjaga kerahasiaan data pengguna



Gambar 4.22 Halaman Dashboard

f. Transaction

Halaman transaksi digunakan untuk bertransaksi dengan pengguna lain yang sudah terdaftar dalam sistem. Pada kasus ini, username aulins akan mengirimkan sejumlah rupiah digital kepada username bernama ulya.



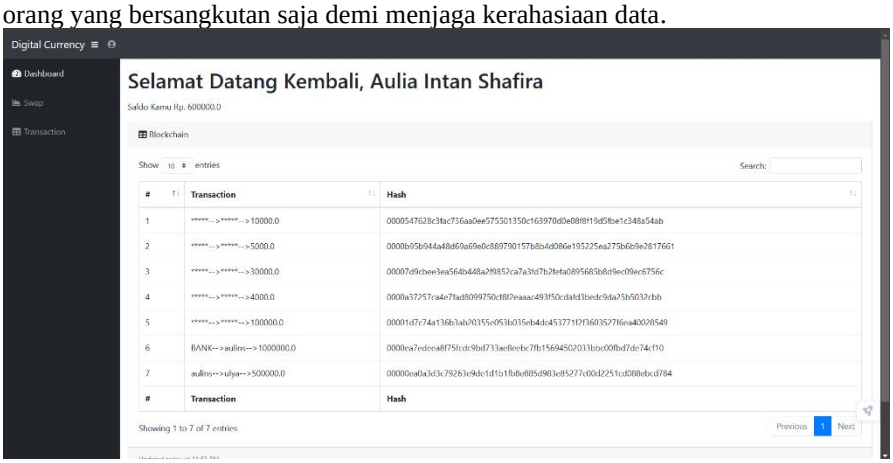
Gambar 4.23 Halaman Transaction

Terlihat pada pesan singkat bahwa uang telah terkirim dan saldo aulins telah berkurang.

Gambar 4.24 Halaman Transaction

g. Penambahan transaksi pada dashboard

Apabila dilihat dari halaman dashboard, akan terlihat penambahan transaksi yang telah dilakukan yaitu pengiriman dari aulins ke ulya. Perlu diingat bahwa transaksi ini hanya akan dilihat oleh kedua orang yang bersangkutan saja demi menjaga kerahasiaan data.



Gambar 4.25 Halaman Dashboard

Pengujian

Pengujian Fungsionalitas Web dengan melakukan pengujian pada halaman registrasi, login, transaksi, dan dashboard untuk memastikan bahwa semua fungsi berjalan dengan baik.

Table 4.3 Tabel Pengujian Fungsionalitas Website

No	Browser	Versi	Hasil
1	Google Chrome	126.0.6478.127 (Official Build) (64-bit)	Tampilan yang dihasilkan: Baik
			Kecepatan: Lancar
			Navigasi: Berfungsi dengan Baik
2	Mozilla Firefox	128.0 (64-bit)	Tampilan yang dihasilkan: Baik
			Kecepatan: Lancar
			Navigasi: Berfungsi dengan Baik
3	Microsoft Edge	126.0.2592.87 (Official build) (64-bit)	Tampilan yang dihasilkan: Baik
			Kecepatan: Lancar
			Navigasi: Berfungsi dengan Baik

5. KESIMPULAN

Berdasarkan hasil penelitian dan implementasi yang telah dilakukan, dapat disimpulkan bahwa penggunaan teknologi blockchain dengan bahasa pemrograman Python dan integrasi dengan aplikasi web menggunakan framework Flask telah berhasil diimplementasikan dengan baik. Proses ini mencakup pembuatan blockchain sederhana yang mampu melakukan transaksi data secara aman dan efisien. Penggunaan pustaka hashlib untuk hashing dan MySQL untuk penyimpanan data memberikan dasar yang kuat untuk pengembangan lebih lanjut.

Implementasi ini menunjukkan bahwa konsep blockchain dapat diterapkan pada sistem transaksi digital dengan memberikan keamanan dan integritas data yang tinggi. Meskipun sistem yang dikembangkan masih menggunakan penyimpanan data terpusat, prototipe ini berhasil menunjukkan potensi teknologi blockchain dalam menjaga kepercayaan dan transparansi dalam transaksi digital. Selain itu, hasil pengujian fungsionalitas menunjukkan bahwa sistem dapat beroperasi dengan lancar pada berbagai platform web browser, memastikan bahwa aplikasi dapat diakses dan digunakan dengan baik oleh pengguna.

DAFTAR PUSTAKA

[1] Antonopoulos, A. M., & Wood, G. (2018). Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications. O'Reilly Media.

[2] Xie, J., & Tang, W. (2021). "A Practical Approach to Blockchain Implementation for Web Applications Using Flask and Python." International Journal of Computer Science and Information Security, 19(5), 135-142.

[3] Li, C., Jiang, P., & Huang, X. (2022). "Integrating Python Blockchain Libraries with Web Frameworks for Secure Transactions." IEEE Access, 10, 49728-49738.

[4] Kim, J., Lee, S., & Kim, Y. (2023). "Evaluating Blockchain Performance in Web Applications: A Case Study

[5] Zheng, Z., Xie, S., Dai, H. N., & Wang, H. (2018). "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends." IEEE International Conference on Big Data, 557-564.