Media of Computer science (MCs)
Vol.2, No.2, December 2025, pp. 101~112
ISSN (print): 3063-4822, ISSN (online): 3063-4997
DOI: 10.69616/mcs.v2i2.249

Implementation of Blockchain Technology for Securing Data Point Transactions in an IoT-Based Waste Sorting System

Naufal Raihan Elriza*[1], Kasliono[2], Hirzen Hasfani[3]
[1,2,3]Department of Computer Systems Engineering, Tanjungpura University, Pontianak, Indonesia
e-mail: ***[1]h1051211094@student.untan.ac.id**, [2]kasliono@siskom.untan.ac.id
[3]hirzenhasfani@siskom.untan.ac.id

Abstract

An Internet of Things (IoT) based waste sorting device awards points to users who dispose of waste through it, as long as the user is registered in the system. However, despite these benefits, this system is vulnerable to various forms of cybercrime. One such challenge is the rise of data manipulation and cyberattacks such as sql injection and threats from internal parties (Insider Threats). This research aims to secure point data transactions in an Internet of Things (IoT) based waste sorting system integrated with blockchain technology to improve security in recording user point data. Tests were conducted to ensure that point data sent from IoT devices were successfully recorded on the blockchain network permanently and verified through transaction hashes in etherscan and to prevent sql Injection and Insider Threat attacks in attempts to illegally alter data. The results of the data transmission test to the blockchain network, which was carried out 30 times, showed that each transaction was successfully recorded and provided a transaction hash. In addition, the attack test, which was carried out 30 times, each attack resulted in a notification with the text "[PERINGATAN] Terjadi Percobaan Pengubahan Poin " in red. Using the blockchain network, both attacks failed to alter user points.

Keywords— *Blockchain, IoT, Transaction Security, Smart Contract, SQL Injection, Waste Sorting System*

Abstrak

Alat pemilah sampah berbasis Internet of Things (IoT) memberikan poin kepada pengguna yang membuang sampah melalui alat tersebut, selama pengguna telah terdaftar dalam sistem. Namun, di balik manfaat tersebut, sistem alat ini rentan terhadap berbagai bentuk kejahatan siber. Salah satu tantangan tersebut adalah maraknya manipulasi data dan serangan siber seperti SQL injection maupun ancaman dari pihak internal (Insider Threat). Penelitian ini bertujuan untuk mengamankan transaksi data poin pada sistem pemilah sampah berbasis Internet of Things (IoT) yang terintegrasi dengan teknologi blockchain guna meningkatkan keamanan dalam pencatatan data poin pengguna. Pengujian dilakukan untuk memastikan data poin yang dikirim dari perangkat IoT berhasil tercatat di jaringan blockchain secara permanen dan terverifikasi melalui transaction hash di etherscan dan mencegah serangan SQL Injection dan Insider Threat dalam percobaan pengubahan data secara ilegal. Hasil pengujian pengiriman data ke jaringan blockchain yang dilakukan sebanyak 30 kali menunjukkan bahwa setiap transaksi berhasil dicatat serta memberikan hash transaction. Selain itu, pengujian serangan yang dilakukan sebanyak 30 kali tiap serangan akan memberikan pemberitahuan berupa teks

"[PERINGATAN] Terjadi Percobaan Pengubahan Poin" yang berwarna merah. Dengan menggunakan jaringan blockchain, kedua serangan tersebut tidak berhasil mengubah poin pengguna.

Kata kunci— *Blockchain, IoT, Keamanan Transaksi, Smart Contract, SQL Injection, Sistem Pemilah Sampah*

1. PENDAHULUAN

Alat pemilah sampah berbasis *Internet of Things* (IoT) merupakan salah satu inovasi dalam mendukung pengelolaan sampah yang efisien, terotomatisasi, serta memberikan insentif kepada masyarakat dalam bentuk poin sebagai hadiah. Pengguna yang ingin menggunakan alat ini harus terlebih dahulu terdaftar dalam sistem. Setelah terdaftar, pengguna dapat membuang sampah ke alat tersebut, dan sistem akan memproses data sampah serta memberikan poin yang sesuai berdasarkan jenis dan berat sampah yang dibuang. Alat pemilah sampah menggunakan NodeMCU Esp 32 sebagai mikrokontroller untuk mengelola proses dari awal hingga akhir, mulai dari deteksi identitas pengguna, klasifikasi jenis sampah, penghitungan berat, pengiriman data ke server, hingga pemberian poin kepada pengguna melalui sistem backend yang terhubung [1].

Untuk identifikasi pengguna, alat ini menggunakan sensor RFID untuk mendeteksi pengguna berdasarkan nomor kartu pengguna melalui trtansmisi gelombang radio. Setiap kartu RFID memiliki nomor ID unik yang digunakan untuk mengenali pengguna secara individual. Ketika kartu didekatkan ke alat, ID pengguna akan dibaca dan dikirim ke mikrokontroler untuk diverifikasi. Setelah itu, sistem akan mengaktifkan sensor *proximity* kapasitif dan sensor *proximity* induktif untuk mengidentifikasi jenis sampah yang dimasukkan. Sensor *proximity* kapasitif untuk mendeteksi dan memilah sampah yang berbahan plastik dan sensor *proximity* indukif untuk mendeteksi sampah yang berbahan logam [2], [3]. Alat ini juga menggunakan sensor Load Cell untuk menimbang berat sampah dengan mendeteksi perubahan berat melalui strain gauge serta modul HX711 sebagai penguat sinyal dan mengonversinya menjadi poin digital [4], [5]. Selain itu, digunakan motor power window untuk menggerakkan sampah yang menggunakan arus listrik DC agar masuk ke tempat sampah yang sesuai berdasarkan jenis sampah dengan bantuan motor servo yang bekerja berdasarkan sistem kontrol tertutup (closed loop) [6], [7]. LCD TFT 3.5 shield digunakakan untuk menampilkan proses pemilahan sampah dan kalkulasi poin. Namun, sistem ini juga berisiko terhadap berbagai jenis serangan siber yang dapat membahayakan integritas dan kerahasiaan data, khususnya data poin pengguna. Kejahatan yang dapat terjadi pada sistem pemilah sampah berbasis IoT adalah pencuiran data poin pengguna. Pencurian tersebut dapat dilakukan dengan teknik sql *injection* dan *insider threat*.

Serangan sql *injection* merupakan salah satu bentuk serangan siber yang berbahaya karena menyerang sistem dengan cara menyisipkan perintah sql berbahaya ke dalam input pengguna [8]. Tujuannya adalah untuk mengakses, memanipulasi, atau bahkan menghapus data pada sistem basis data yang menjadi target. Teknik ini sering digunakan oleh peretas untuk mengeksploitasi celah keamanan dalam sistem basis data. Serangan *insider trheat* dapat terjadi secara sengaja maupun tidak sengaja, di mana karyawan menyalahgunakan wewenang yang mereka miliki. Dengan akses yang diberikan, mereka dapat memperoleh informasi atau memasuki sistem yang seharusnya tetap terlindungi [9]. Kedua serangan tersebut dapat menjadi permasalah yang besar pada alat pemilah sampah berbasis IoT jika tidak mempunyai keamanan yang dapat mencegah dari serangan tersebut. Maka dari itu diperlukan teknologi *blockchain* untuk memperkuat keamanan serta pencegahan serangan sql *injection* dan *insider threat*.

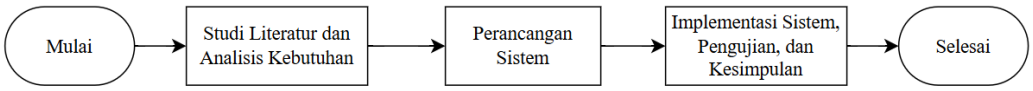
Teknologi *blockchain* dapat digunakan sebagai solusi atas dari permasalahan dalam sistem digital yang berkaitan dengan keamanan dan transparansi data. *Blockchain* menjadi solusi yang lebih aman dan transparan karena bersifat desentralisasi, sehingga tidak bergantung pada satu otoritas tunggal [10]. *Blockchain* memfasilitasi pembentukan jaringan terdistribusi tanpa memerlukan kepercayaan antar pengguna (*trustless network*), serta menggunakan kriptografi untuk menjamin validitas dan keamanan setiap interaksi dalam jaringan [11]. Pengiriman data ke

jaringan blockchain tidak hanya tercatat secara permanen dan tidak dapat diubah, tetapi juga dapat dengan mudah diverifikasi oleh semua pihak yang terlibat, serta mengurangi potensi manipulasi atau penipuan [12].

Penelitian ini bertujuan untuk mengintegrasikan teknologi *blockchain* ke dalam proses pengiriman data poin, sehingga setiap transaksi yang mencatat jumlah dan jenis sampah serta poin yang diterima pengguna, dapat disimpan dalam bentuk blok yang terenkripsi dan terdistribusi. Hal ini memastikan bahwa data poin yang dikirim dari perangkat IoT tetap aman dan akurat, serta tidak dapat dimanipulasi baik oleh pihak internal maupun eksternal.

2. METODE PENELITIAN

Metode penelitian pada implementasi teknologi *blockchain* untuk keamanan transaksi pengiriman data poin pada sistem pemilah sampah berbasis IoT terdiri dari beberapa tahapan, yaitu studi literatur dan analisis kebutuhan, perancangan sistem, dan implementasi sistem, pengujian, serta kesimpulan. Tahapan penelitian diawali dengan studi literatur dan analisis kebutuhan perangkat keras dan lunak, seperti Alat Pemilah Sampah Berbasis IoT, *Blockchian*, dan NodeMCU ESP32. Tahapan selanjutnya meliputi perancangan sistem, implementasi sistem, pengujian sistem menggunakan serangan Sql *Injection* dan *Insider Threat* dan menarik kesimpulan. Adapun Gambar 1 yang memperlihatkan diagram alir penelitian.



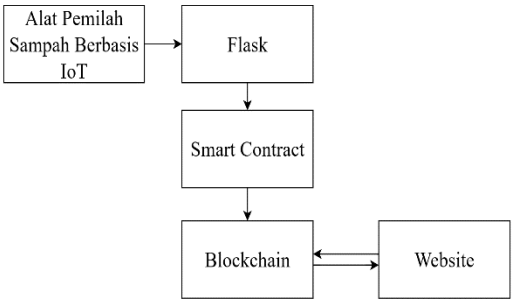
Gambar 1 Diagram Alir Penelitian

2.1 Studi Literatur dan Analisis Kebutuhan

Studi literatur melibatkan pengumpulan informasi dan kajian literatur terkait penggunaan teknologi *blockchain*, metode penyerangan *database*, dan meningkatkan keamanan penyimpanan data. Dalam tahap analisis kebutuhan ditentukan komponen utama, yaitu alat pemilah sampah berbasis IoT dengan NodeMCU ESP32 sebagai mikrokontroler yang berfungsi mengatur poin pengguna sebelum diteruskan ke Flask dan dicatat di *blockchain*. *Blockchain Ethereum* digunakan untuk mencatat transaksi secara aman melalui smart contract yang ditulis dengan *Solidity* dan dikembangkan menggunakan *Remix* IDE, sementara integrasi aplikasi dilakukan dengan *Flask* dan Web3.py.

2. 2 Perancangan Sistem

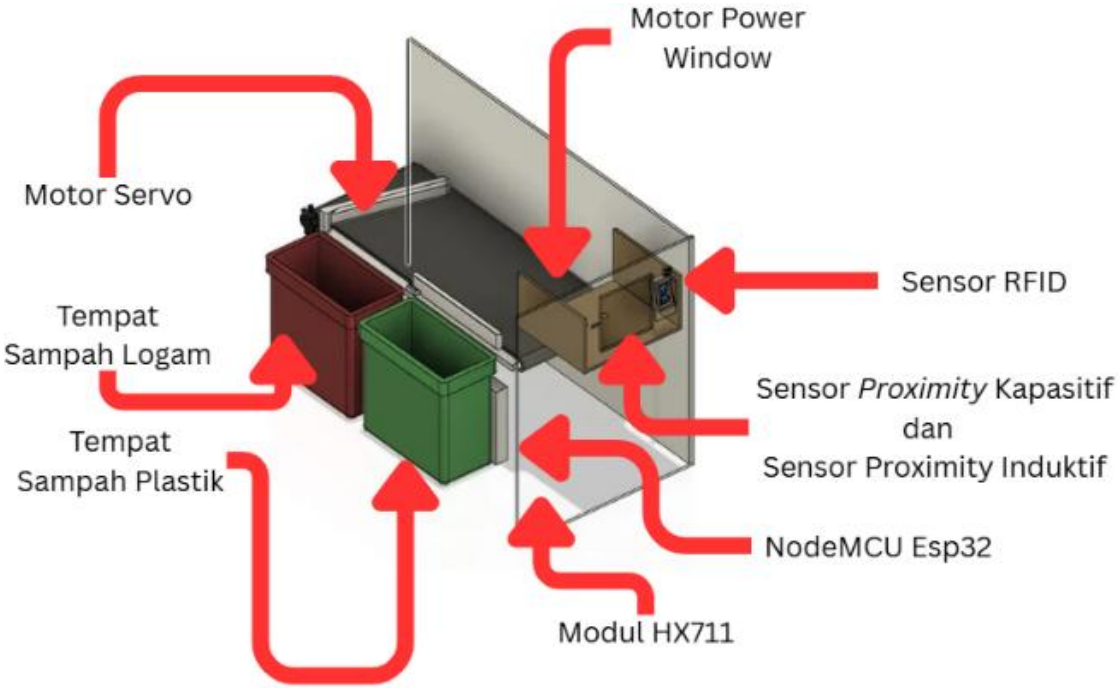
Perancangan sistem mencakup perancangan alat pemilah sampah berbasis iot dan perancangan blockchain. Perangkat keras berupa alat pemilah sampah berbasis iot. *Blockchain* menggunakan *Ethereum testnet Sepolia* dengan *smart contract Solidity* yang dideploy melalui *Remix* IDE untuk mencatat dan memvalidasi transaksi secara otomatis. Adapun perancangan arsitektur sistem secara umum dapat dilihat pada Gambar 2.



Gambar 2 Rancangan Arsitektur Sistem

2. 2.1 Perancangan Alat Pemilah Sampah Berbasis IoT

Perangkat keras yang digunakan pada alat pemilah sampah berbasis IoT terdiri dari beberapa komponen. Modul RFID digunakan untuk mengidentifikasi pengguna berdasarkan nomor kartu RFID yang bersifat unik, sehingga sistem dapat mencatat aktivitas penyetoran sampah secara individu. Untuk mendeteksi jenis sampah, alat ini dilengkapi dengan dua sensor proximity, yaitu kapasitif dan induktif. Setelah jenis sampah dikenali, alat menimbangnya menggunakan sensor load cell, dan hasil pengukuran berat tersebut dikonversi menjadi poin digital melalui modul HX711. Selanjutnya, sistem secara otomatis memindahkan sampah ke tempat yang sesuai menggunakan motor power window sebagai pendorong, serta motor servo untuk mengatur arah penutup atau katup pemilah. Untuk memberikan umpan balik secara visual kepada pengguna, seluruh proses ditampilkan melalui layar LCD TFT 3.5 shield. Layar ini menyajikan informasi seperti jenis sampah, berat, dan poin yang diperoleh dari setiap transaksi penyetoran. Gambar rancangan alat pemilah sampah berbasis IoT dapat dilihat pada Gambar 3.



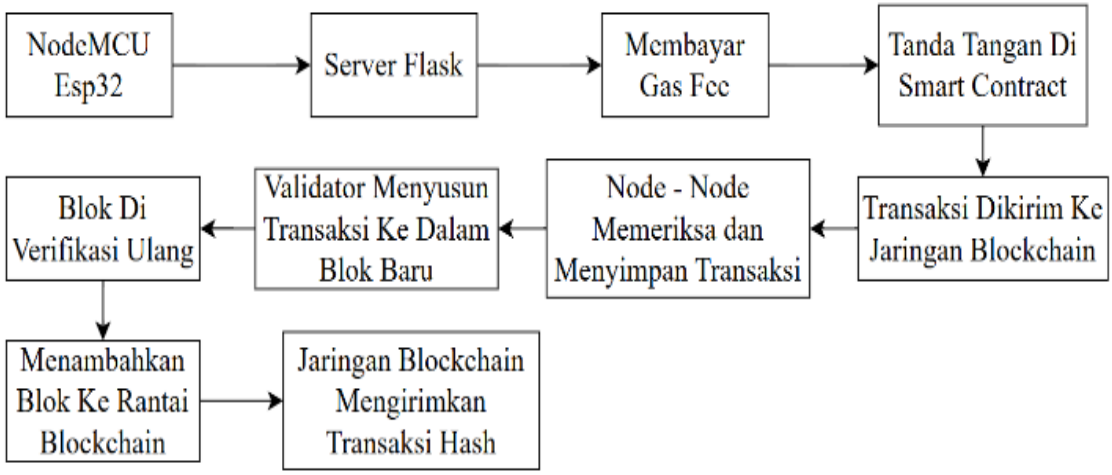
Gambar 3 Rancangan Alat Pemilah Sampah Berbasis IoT

2. 2.2 Perancangan Pengiriman Data ke Blockchain

Blockchain adalah sistem terdistribusi yang bekerja berdasarkan konsep desentralisasi dan enkripsi, memungkinkan setiap pengguna dalam jaringan untuk memverifikasi transaksi dan berpartisipasi dalam proses validasi data, sehingga menciptakan lingkungan yang transparan di mana seluruh transaksi dapat dipantau dan diakses secara global [13], [14]. Hal penting dalam teknologi ini adalah *smart contract*, yaitu program komputer yang secara otomatis memverifikasi dan mengeksekusi dirinya sendiri tanpa memerlukan pihak ketiga, seperti yang pertama kali dikemukakan oleh Nick Szabo pada tahun 1994 [15]]. Platform yang paling umum digunakan untuk mengembangkan dan menjalankan *smart contract* adalah *Ethereum* karena dilengkapi dengan *Ethereum Virtual Machine* (EVM) untuk memfasilitasi eksekusi kode secara terdesentralisasi [16]. Pengembangan smart contract ini dilakukan menggunakan bahasa pemrograman *Solidity* yang dirancang untuk jaringan *Ethereum*. *Remix* IDE digunakan sebagai alat berbasis web yang dapat diakses langsung melalui browser tanpa instalasi tambahan [17].

Adapun mekanisme pengiriman data transaksi poin pengguna dari alat pemilah sampah berbasis IoT ke jaringan *blockchain* adalah sebagai berikut. Setelah NodeMCU Esp32 mengirimkan data yang berisikan id pengguna, jenis sampah, dan poin yang diperoleh ke server

Flask, data tersebut akan diproses oleh *smart contract* yang terintegrasi dalam sistem backend. Sebelum transaksi dikirim ke jaringan, *backend* akan memberikan *gas fee*. Transaksi ini kemudian dikirim ke jaringan *Ethereum* untuk divalidasi. *Smart contract* akan mengelola dan meneruskan data transaksi ke jaringan *blockchain*. Setelah dikirim, transaksi akan disebar ke seluruh node dalam jaringan. *Node-node* tersebut akan memeriksa dan menyimpan transaksi dalam antrian transaksi. Validator kemudian memilih transaksi dari antrian ini untuk disusun ke dalam blok baru. Setiap transaksi dalam blok akan dihash dan disusun menggunakan struktur pohon hash. Jika blok berhasil divalidasi, maka blok tersebut akan disebarluaskan ke seluruh *node* dalam jaringan lalu diverifikasi ulang dan ditambahkan secara permanen ke rantai *blockchain*. Setelah ini, jaringan akan mengembalikan *transaction hash* sebagai bukti bahwa transaksi telah dicatat secara permanen. *Transaction hash* tersebut kemudian disimpan di sistem dan ditampilkan pada antarmuka website bersama dengan riwayat transaksi serta informasi pengguna yang melakukan penyetoran. Gambar mekanisme pengiriman data ke jaringan *blockchain* dapat dilihat pada Gambar 4.



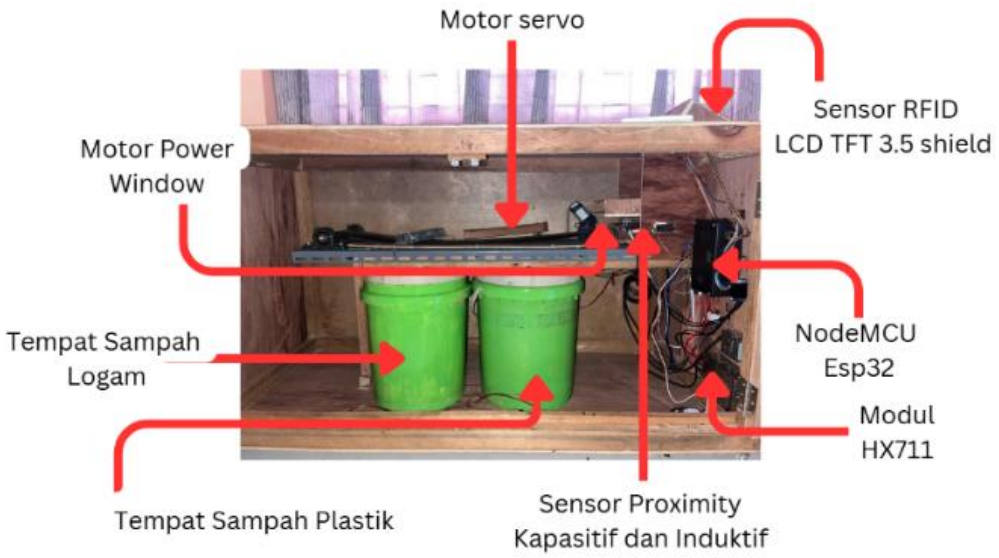
Gambar 4 Mekanisme Pengiriman Data Ke Jaringan Blockchain

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Alat Pemilah Sampah Berbasis IoT

Perangkat keras yang digunakan pada alat pemilah sampah berbasis IoT terdiri dari beberapa komponen yang mendukung proses identifikasi pengguna, pendeteksian jenis sampah, penimbangan berat, pemilahan otomatis, serta visualisasi proses. Untuk proses identifikasi, sistem memanfaatkan sensor RFID yang membaca nomor kartu RFID milik pengguna. Setiap pengguna memiliki ID RFID yang unik, sehingga memungkinkan sistem untuk mencatat, memantau, dan melacak aktivitas penyetoran sampah secara individu. Setelah proses identifikasi selesai, sistem melanjutkan dengan deteksi jenis sampah menggunakan dua jenis sensor proximity, yaitu sensor induktif dan sensor kapasitif. Sensor induktif digunakan untuk mendeteksi material logam, sedangkan sensor kapasitif digunakan untuk mengenali bahan non-logam seperti plastik.

Setelah jenis sampah dikenali, sampah ditimbang menggunakan sensor Load Cell yang mampu mengukur berat berdasarkan deformasi fisik pada strain gauge. Data berat tersebut kemudian diperkuat dan dikonversi menjadi data digital menggunakan modul HX711, yang memungkinkan kalkulasi poin secara akurat berdasarkan bobot sampah. Setelah kalkulasi selesai, sistem secara otomatis mengarahkan sampah ke wadah yang sesuai menggunakan motor power window sebagai aktuator pendorong, dan motor servo yang mengatur arah katup pemilah berdasarkan jenis sampah. Keseluruhan proses ditampilkan melalui layar LCD TFT 3.5 shield yang memberikan visualisasi proses pemilahan dan perolehan poin secara langsung kepada pengguna. Gambar alat pemilah sampah berbasis IoT dapat dilihat pada Gambar 5.



Gambar 5 Alat Pemilah Sampah Berbasis IoT

3.2 Implementasi Pengiriman Data ke Blockchain

Pengujian pengiriman data poin ke jaringan *blockchain* dilakukan untuk memverifikasi integritas sistem dalam mencatat setiap aktivitas pengguna, khususnya transaksi poin, secara permanen dan aman. Dalam sistem pemilah sampah berbasis iot ini, proses pengujian menjadi penting karena menyangkut validasi data yang berasal dari perangkat keras (sensor dan RFID) hingga disimpan secara digital dalam jaringan *blockchain Ethereum* melalui *smart contract*. Tujuan utama dari pengujian ini adalah memastikan bahwa setiap perhitungan poin yang berasal dari berat dan jenis sampah yang dimasukkan oleh pengguna benar-benar tersimpan dalam jaringan blockchain tanpa bisa diubah atau dihapus oleh pihak mana pun.

```
@app.route('/add_points', methods=['POST'])
def add_points():
    token = request.headers.get("X-API-Key")
    if token != ESP32_API_KEY:
        log_event("[PERINGATAN] Terjadi Percobaan Pengubahan Poin")
        return jsonify({"status": "error", "message": "Unauthorized access"}), 403

    data = request.get_json()
    nomor_kartu = data.get("nomor_kartu", "").strip().upper()
    jenis = data.get("jenis")
    berat = float(data.get("berat", 0))
    poin = int(data.get("poin", 0))

    log_event(f"[ADD POINTS] Request kartu: {nomor_kartu}, Jenis: {jenis}, Berat: {berat}, Poin: {poin}")

    if len(nomor_kartu) != 8:
        return jsonify({"status": "error", "message": "Nomor kartu harus 8 digit."})

    try:
        nama = get_nama_by_kartu(nomor_kartu)
        if not nama:
            return jsonify({"status": "error", "message": "Kartu tidak ditemukan dalam database."})

        poin_sebelumnya = contract.functions.getUserPoints(nomor_kartu).call()
        total_poin = poin_sebelumnya + poin

        nonce = w3.eth.get_transaction_count(FROM_ADDRESS)
        tx = contract.functions.addPoints(nomor_kartu, poin).build_transaction({
            'chainId': 11155111,
            'from': FROM_ADDRESS,
            'gas': 2000000,
            'gasPrice': w3.to_wei('20', 'gwei'),
            'nonce': nonce,
        })

        signed_tx = w3.eth.account.sign_transaction(tx, private_key=PRIVATE_KEY)
        tx_hash = w3.eth.send_raw_transaction(signed_tx.raw_transaction)
        w3.eth.wait_for_transaction_receipt(tx_hash)

        save_transaction_to_file(nama, jenis, berat, poin, total_poin, tx_hash.hex())

        return jsonify({
            "status": "success",
            "nama": nama,
            "tx_hash": tx_hash.hex(),
            "poin_didapat": poin,
            "total_poin": total_poin
        })
    except Exception as e:
        log_event(f"[ERROR] Add points: {e}")
        return jsonify({"status": "error", "message": str(e)})
```

Gambar 6 Kode Program Pengiriman Data Poin Ke Blockchain

Program perhitungan poin dibagi menjadi tiga tahap. Pada tahap pertama, sistem menunggu input dari sensor kapasitif atau induktif untuk mendeteksi jenis sampah. Selanjutnya, sistem masuk ke tahap kedua untuk mengukur berat menggunakan sensor load cell. Setelah itu, tahap ketiga dijalankan, yaitu menunggu pengguna menempelkan kartu RFID. Jika kartu terbaca, sistem mengirim permintaan validasi ke endpoint /validate_card pada server Flask. Apabila kartu valid, poin dihitung berdasarkan jenis dan berat sampah (logam: 3 poin/gram, plastik: 2 poin/gram), kemudian dikirim ke endpoint /add_points dalam format JSON. Jika server merespons sukses, sistem mencatat transaksi, menampilkan hash transaksi, serta menggerakkan servo dan motor untuk menyalurkan sampah ke tempat yang sesuai.

Server Flask memuat konfigurasi koneksi ke node *Ethereum* dari *smart contract*, serta pengaturan koneksi ke *database* MySQL yang menyimpan data pengguna. Fungsi get_nama_by_kartu() digunakan untuk memverifikasi apakah kartu RFID terdaftar dalam basis data, sedangkan save_transaction_to_file() mencatat transaksi ke dalam file log pengguna. Flask menyediakan beberapa endpoint: /validate_card untuk memeriksa keabsahan kartu, /add_points untuk menambahkan poin ke *blockchain*, /check_balance untuk melihat saldo poin, dan /redeem_points untuk menukar poin. Kode Program flask dapat dilihat pada Gambar 6. *Smart contract Solidity* bernama PointSystem bertugas mengelola poin pengguna berdasarkan nomor kartu. Fungsi getUserPoints digunakan untuk melihat saldo poin, addPoints untuk menambahkan poin, dan redeemPoints untuk menukar poin jika jumlahnya mencukupi. Jika saldo tidak cukup, transaksi akan gagal dengan pesan error "Not enough points". Kode program *Solidity* dapat dilihat pada Gambar 7.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract PointSystem {
    // Menyimpan poin berdasarkan nomor kartu
    mapping(string => uint256) public points;

    // Ambil poin pengguna berdasarkan nomor kartu
    function getUserPoints(string memory nomorKartu) public view returns (uint256) {
        return points[nomorKartu];
    }

    // Tambah poin untuk nomor kartu tertentu
    function addPoints(string memory nomorKartu, uint256 amount) public {
        points[nomorKartu] += amount;
    }

    // Tukar poin dari nomor kartu
    function redeemPoints(string memory nomorKartu, uint256 pointsToRedeem) public {
        require(points[nomorKartu] >= pointsToRedeem, "Not enough points.");
        points[nomorKartu] -= pointsToRedeem;
    }
}
```

Gambar 7 Kode Program *Solidity*

Setiap transaksi yang berhasil dikirim ke jaringan *blockchain* akan menghasilkan output berupa *transaction hash*, yaitu string hash kriptografik yang menjadi bukti digital bahwa transaksi tersebut telah diterima dan diproses oleh jaringan Ethereum. *Transaction hash* ini dapat diverifikasi secara publik melalui layanan seperti *Etherscan*, sehingga menjamin keaslian, keabsahan, dan keterbukaan terhadap siapa pun yang ingin memantau transaksi tersebut. Adapun tabel data transaksi dapat dilihat pada Tabel 1.

Tabel 1 Pengiriman Data ke *Blockchain*

No.	Waktu	Nama	Jenis	Berat	Poin	Jumlah Poin	Hash
1.	2025-05-15 20:11:14	Nopal	Plastik	11.23g	22	22	7d7...914
2.	2025-05-15 20:13:00	Nopal	Plastik	2.51g	5	27	72b... e63
3.	2025-05-15 20:13:12	Nopal	Logam	48.39g	145	172	edf...574
4.	2025-05-15 20:13:37	Nopal	Plastik	10.89g	22	194	1cc... 989
5.	2025-05-15 20:34:25	Yessa	Logam	63.78g	191	191	0f1... 9b1
6.	2025-05-15 20:35:12	Yessa	Plastik	6.14g	12	203	372... c2e
7.	2025-05-15 20:36:24	Yessa	Penukaran	0	-56	147	b16... d04
8.	2025-05-15 20:40:25	Yessa	Plastik	8.63g	17	164	e07... 818
9.	2025-05-15 20:41:00	Yessa	Plastik	1.59g	3	167	447... b75
10.	2025-05-23 15:58:16	Nopal	Logam	120.5g	361	555	84c...fb6

Hasil pengujian menunjukkan bahwa sebanyak 10 transaksi yang dikirimkan dari alat pemilah sampah berbasis iot diteruskan ke jaringan *blockchain Ethereum* berhasil diproses. Tabel 1 menampilkan data hasil pengujian tersebut, yang mencakup waktu transaksi, nama pengguna, jenis sampah, berat sampah, poin yang didapat, total saldo poin setelah transaksi, dan *transaction hash*. Sebagai contoh, pada transaksi pertama yang dilakukan pada tanggal 15 Mei 2025 pukul 20:11:14 oleh pengguna bernama Nopal, sistem mendeteksi sampah plastik seberat 11,23 gram, menghasilkan 22 poin, dan tercatat dalam *blockchain* dengan hash transaksi 7d7...914. Transaksi berikutnya menunjukkan konsistensi dalam pencatatan dan penghitungan poin yang akurat.

3.3 Pengujian Serangan Sql Injection dan Insider Threat

Pengujian ini dilakukan untuk mengevaluasi tingkat keamanan sistem pemilah sampah berbasis IoT terhadap dua jenis ancaman siber yang umum dan berbahaya, yaitu sql *injection* dan *Insider Threat*. Tujuan dari pengujian ini adalah untuk menguji sejauh mana sistem dapat mendeteksi, menolak, dan melaporkan upaya manipulasi data poin pengguna yang dilakukan oleh pihak eksternal maupun internal.

Langkah awal pengujian serangan sql *injection* dimulai dari pengiriman permintaan HTTP POST ke endpoint /add_points, di mana penyerang menyisipkan perintah sql ke dalam parameter seperti nomor_kartu, jenis, berat, dan poin. Sqlmap dikonfigurasi dengan level dan risiko injeksi yang berbeda-beda untuk menguji berbagai kemungkinan eksploitasi. Perintah sql berbahaya yang disisipkan bertujuan untuk mengubah nilai poin pengguna secara langsung melalui perintah seperti OR '1'='1', UNION SELECT, atau manipulasi kueri UPDATE. Setelah menjalankan proses serangan, output dari sqlmap ditampilkan pada command prompt yang menunjukkan apakah sistem rentan terhadap injeksi dan apakah data dapat diubah. Tabel pengujian dari serangan sql *injection* yang telah dilakukan dapat dilihat pada Tabel 2

Tabel 2 Pengujian Dengan Melakukan Serangan Sql Injection

No.	Waktu	Keterangan	Hasil
1.	2025-07-17 14:50:54	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah

No.	Waktu	Keterangan	Hasil
2.	2025-07-17 14:50:54	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
3.	2025-07-17 14:50:56	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
4.	2025-07-17 14:50:57	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
5.	2025-07-17 14:50:59	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
6.	2025-07-17 14:50:59	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
7.	2025-07-17 14:51:01	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
8.	2025-07-17 14:51:03	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
9.	2025-07-17 14:51:03	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
10.	2025-07-17 14:51:05	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah

Hasil pengujian pada Tabel 2 menunjukkan bahwa seluruh upaya serangan yang dilakukan tidak berhasil mengubah data poin pengguna. Sistem secara otomatis mendeteksi adanya aktivitas mencurigakan dan memunculkan pesan peringatan pada terminal, berupa teks: "[PERINGATAN] Terjadi Percobaan Pengubahan Poin", yang ditampilkan dalam warna merah sebagai tanda bahaya. Respons sistem menunjukkan bahwa backend telah dilengkapi dengan validasi input, sanitasi data, dan perlindungan terhadap perintah sql berbahaya.

Ancaman *insider threat* merujuk pada situasi di mana individu yang memiliki akses legal ke sistem menyalahgunakan hak akses tersebut untuk melakukan tindakan ilegal. Dalam pengujian ini, skenario difokuskan pada seorang administrator sistem yang mencoba mengirimkan data palsu secara langsung ke backend server untuk menambah poin pengguna. Serangan dilakukan menggunakan perintah curl melalui terminal, yang mengirim permintaan HTTP POST ke endpoint /add_points dengan menyisipkan data dalam format JSON. Data yang dikirim meliputi nomor_kartu, jenis sampah, berat, dan poin, yang diisi secara manual oleh admin tanpa proses validasi dari sistem sensor IoT. Contoh data yang dikirim antara lain adalah: nomor_kartu: "B3EE112D", jenis: "Logam", berat: 120, dan poin: 360. Dengan menyertakan header "Content-Type: application/json", server Flask diharapkan akan menerima data dan memprosesnya sebagaimana data sah dari perangkat.Tabel pengujian dari serangan insider threat yang telah dilakukan dapat dilihat pada Tabel 3.

Tabel 3 Pengujian Dengan Melakukan Serangan Insider threat

No.	Waktu	Keterangan	Hasil
1.	2025-07-17 14:53:05	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
2.	2025-07-17 14:53:07	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
3.	2025-07-17 14:53:07	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
4.	2025-07-17 14:53:09	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
5.	2025-07-17 14:53:09	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
6.	2025-07-17 14:53:11	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah

No.	Waktu	Keterangan	Hasil
7.	2025-07-17 14:53:11	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
8.	2025-07-17 14:53:13	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
9.	2025-07-17 14:53:15	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah
10.	2025-07-17 14:53:17	[PERINGATAN] Terjadi Percobaan Pengubahan Poin	Poin tidak berubah

4. KESIMPULAN

Hasil pengujian menunjukkan bahwa sistem berhasil mendeteksi adanya upaya penyisipan data tidak sah. Pesan peringatan yang sama seperti pada serangan sql *injection* juga ditampilkan pada terminal, yaitu "[PERINGATAN] Terjadi Percobaan Pengubahan Poin". Tabel 3 menunjukkan hasil 10 kali percobaan serangan *insider threat* dengan hasil yang konsisten bahwa sistem tidak memproses data palsu dan saldo poin pengguna tetap tidak berubah

Kedua jenis pengujian, baik SQL Injection maupun Insider Threat, dilakukan dengan total 20 percobaan (masing-masing 10 kali). Hasil dari kedua pengujian tersebut menunjukkan bahwa sistem berhasil menggagalkan seluruh upaya serangan dan mempertahankan integritas data poin pengguna. Kedua pengujian ini bertujuan untuk mengevaluasi ketahanan sistem teknologi *blockchain* terhadap dua jenis serangan, yaitu sql *injection* dan *insider threat*. Pengujian dilakukan dengan penyerang mencoba mengubah data poin pengguna dengan teknik sql *injection* yang menggunakan alat penetrasi sqlmap dan menggunakan teknik *insider threat* dengan melalui pengiriman data manual langsung ke endpoint sistem oleh seorang admin menggunakan command curl. Hasil pengujian menunjukkan bahwa dalam kedua jenis serangan, data poin pengguna tidak mengalami perubahan dan sistem berhasil menggagalkan upaya seraangan tersebut. Dengan pencatatan data poin ke dalam jaringan *blockchain Ethereum* melalui smart contract, setiap transaksi yang sah akan memiliki hash unik dan bersifat immutable. Secara keseluruhan, hasil pengujian memberikan bukti kuat bahwa sistem yang dibangun telah memenuhi aspek keamanan digital dengan baik.

5. SARAN

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa hal yang dapat direkomendasikan untuk pengembangan penelitian dan sistem di masa mendatang. Pertama, penelitian selanjutnya disarankan untuk memperluas skenario pengujian keamanan dengan melibatkan jenis serangan siber yang lebih beragam, seperti replay attack, distributed denial-of-service (DDoS), dan manipulasi smart contract, sehingga ketahanan sistem dapat dievaluasi secara lebih komprehensif. Kedua, sistem dapat dikembangkan dengan menerapkan mekanisme autentikasi dan otorisasi yang lebih kuat, misalnya penggunaan multi-factor authentication (MFA) atau digital signature, guna meningkatkan perlindungan terhadap ancaman insider threat. Ketiga, optimalisasi performa blockchain perlu dilakukan, khususnya pada aspek latensi transaksi dan biaya gas (gas fee), melalui pemanfaatan teknik layer-2, sidechain, atau private blockchain agar sistem lebih efisien untuk implementasi skala besar. Keempat, penelitian berikutnya dapat mengintegrasikan mekanisme analisis data dan visualisasi yang lebih lanjut untuk memantau pola transaksi poin pengguna, sehingga sistem tidak hanya berfungsi sebagai pengaman data, tetapi juga sebagai dasar pengambilan keputusan dalam pengelolaan program insentif pengelolaan sampah. Terakhir, pengujian sistem pada lingkungan nyata dengan jumlah pengguna dan perangkat IoT yang lebih besar perlu dilakukan untuk menguji skalabilitas, reliabilitas, dan keberlanjutan sistem dalam konteks implementasi yang lebih luas.

DAFTAR PUSTAKA

[1] E. Prihantoro, Sulistiyanto, and M. F. Efendi, "Perancangan Smart Klinik Berbasis Mikrokontroler Nodemcu ESP32," *JEECOM*, vol. 3, no. 2, 2021, https://doi.org/10.33650/jeecom.v3i2.6234.

[2] I. Al Rasyid, "Pengembangan Sistem Informasi Absensi Berbasis Radio Frequency Identification (RFID) Terintegrasi Dengan Sistem Informasi Akademik," *Ilmudata.org*, vol. 3, no. 2, p. 1, 2023.

[3] I. D. Rahman, M. A. Auliq, and Sutikno, "Prototype Alat Pemilah dan Penghancur Sampah Berbasis Mikrokontroler Arduino UNO R3 Sebagai Bahan Pupuk Organik," *Jurnal Teknik Elektro dan Komputasi (ELKOM)*, 2024, https://doi.org/10.32528/elkom.v6i2.22318.

[4] Z. Arifin, M. Zaenudin, and Y. Saleh, "Perancangan Kontroler Pada Konveyor Pendeteksi Berat Menggunakan Load Cell Berbasis PLC," *TECHNOPEX-2023 Institut Teknologi Indonesia*, 2023.

[5] A. Kurniawan and A. Rizky, "Perancangan Alat Timbang Untuk Rekapitulasi Pemakaian Zat Pewarna Kain Di Pt. Indo-Rama Synthetic Tbk. Menggunakan Arduino Wemos Lolin S2 Mini Dengan Modul Load Cell HX711," *Jurnal Informatika dan Komputer (INFOKOM) Volume 12 Nomor 1 Tahun 2024*, 2024, https://doi.org/10.56689/infokom.v12i1.1137

[6] B. Nughroho, E. Prasetiyo, and G. Marausna, "Rancang Bangun Dual Axis Sun Tracker Menggunakan Motor DC Power Window CSD60-B," *JURNAL TEKNOLOGI TERPADU VOL. 10. NO. 1*, vol. 10, no. 1, 2022, https://doi.org/10.32487/jtt.v10i1.1422.

[7] R. Ramdan, L. Lasmadi, and P. Setiawan, "Sistem Pengendali On-Off Lampu dan Motor Servo sebagai Penggerak Gerendel Pintu Berbasis Internet Of Things (IoT)," *AVITEC*, vol. 4, no. 2, p. 211, Aug. 2022, https://doi.org/10.28989/avitec.v4i2.1317.

[8] F. T. Anugrah, S. Ikhwan, and J. Gusti, "Implementasi Intrusion Prevention System (IPS) Menggunakan Suricata Untuk Serangan SQL Injection," *Techné Jurnal Ilmiah Elektroteknika*, 2022, https://doi.org/10.31358/techne.v21i2.320.

[9] Y. A. Setiawan, "Analisis Ancaman Orang Dalam dan Strategi Intrusi Sistem Proteksi Fisik Reaktor Nuklir dengan Pendekatan Stokastik," *Jurnal Pengembangan Energi Nuklir*, vol. 22, no. 1, pp. 1–7, 2020, https://doi.org/10.17146/jpen.2020.22.1.5460.

[10] M. Yeni and D. Kumala, "Teknologi Blockchain untuk Transparansi dan Keamanan pada Era Digital," *Unmuha Repository*, 2020.

[11] Budianto, R. Mumpuni, and H. E. Wahanani, "Pembuatan Smartcontract Aplikasi Fundraising Berbasis Blockchain," *Jurnal Informatika Dan Tekonologi Komputer (JITEK)*, vol. 3, no. 3, pp. 204–212, Nov. 2023, https://doi.org/10.55606/jitek.v3i3.2001.

[12] T. W. E. Suryawijaya, "Memperkuat Keamanan Data melalui Teknologi Blockchain: Mengeksplorasi Implementasi Sukses dalam Transformasi Digital di Indonesia," *JKSP: Jurnal Studi Kebijakan Publik*, vol. 2, no. 1, pp. 55–67, 2023, https://doi.org/10.21787/jskp.2.2023.55-68.

[13] Z. Munawar, N. I. Putri, I. Iswanto, and D. Widhiantoro, "Analisis Keamanan Pada Teknologi Blockchain," *Infotronik : Jurnal Teknologi Informasi dan Elektronika*, vol. 8, no. 2, p. 67, Dec. 2023, https://doi.org/10.32897/infotronik.2023.8.2.2062.

[14] K. Sinha and M. Verma, "The Detection of SQL Injection on Blockchain-Based Database," 2021, pp. 234–262. https://doi.org/10.4018/978-1-7998-7589-5.ch011.

[15] I. Parmitasari, "EKSISTENSI SMART CONTRACT MENURUT HUKUM KONTRAK DI INDONESIA," *Prosiding Seminar Nasional Hasil Penelitian dan Pengabdian Masyarakat*, 2022.

[16] A. M. Mabruroh, F. Dewanta, and A. A. Wardana, "Implementasi Ethereum Blockchain dan Smart Contract pada Jaringan Smart Energy Meter," *JURNAL MULTINETICS*, vol. 7, no. 1, p. 82, 2021, https://doi.org/10.32722/multinetics.v7i1.4122 .

[17] S. Singh, V. Tiwari, and V. Vadi, "Smart Contract Using Solidity (Remix-Ethereum IDE)," *International Journal of Advanced Research in Computer and Communication Engineering ISO*, vol. 3297, no. 2, 2023, https://doi.org/10.17148/IJARCCE.2023.12253.