

Penerapan Teknologi *Blockchain* Pada Sistem Transaksi Aplikasi *Point of Sale* Berbasis Web

Yoga Dwi Nugroho¹, I Made Suartana²

^{1,2} Teknik Informatika, Universitas Negeri Surabaya

¹yogadwi.20083@mhs.unesa.ac.id

²madesuartana@unesa.ac.id

Abstrak— Aplikasi *point of sale* (POS) memerlukan keamanan dan transparansi yang tinggi untuk meminimalisir perubahan data tidak diinginkan yang berkaitan dengan proses transaksi penjualan dan pembelian. Untuk memenuhi kebutuhan ini, peneliti membuat sebuah aplikasi *point of sale* (POS) berbasis web yang menerapkan teknologi *blockchain* dan menggunakan *smart contract* yang diterapkan pada fitur transaksi. Teknologi *blockchain* yang digunakan memiliki metode konsensus *proof of stake* yang berfungsi melakukan verifikasi semua perubahan pada blok terenkripsi dalam jaringan *blockchain* terdesentralisasi. Selain menggunakan teknologi *blockchain*, website *point of sale* ini dibangun menggunakan *framework* *react.js* untuk pembuatan tampilan *user interface* dan *library* *web3.js* untuk mengintegrasikan teknologi *blockchain* dengan *react.js*. Sistem *point of sale* (POS) ini diuji dengan dua jenis pengujian, yaitu *functional testing* untuk menguji keberhasilan setiap fungsi pada *user interface*, dan *smart contract testing* untuk memastikan semua fungsi pada sistem transaksi berjalan tanpa ada masalah. Dengan hasil pengujian yang telah dilakukan, sistem *point of sale* (POS) yang menerapkan teknologi *blockchain* dapat mengatasi permasalahan yang ada, terutama mengenai transparansi perubahan data transaksi yang dilakukan.

Kata Kunci— Transaksi, *point of sale*, *blockchain*, *smart contract*, *proof of stake*.

I. PENDAHULUAN

Pada era modern seperti sekarang ini, teknologi memiliki peran yang sangat penting dalam kehidupan kita. Teknologi telah menyebar ke berbagai sektor, seperti pendidikan, hiburan, dan bisnis[1]. Pada dunia bisnis, pemanfaatan teknologi yang sedang berkembang pesat saat ini salah satu contohnya adalah penerapan sistem *Point of sale*. *Point of sale* merupakan aplikasi yang digunakan untuk melakukan transaksi penjualan, pembelian, retur, *inventory*, dan lain-lain [2]. Selain itu, aplikasi *Point of sale* juga sudah terhubung dengan cloud computing, sehingga seluruh transaksi di toko tercatat dengan benar.

Sistem transaksi yang manual memakan waktu dan tidak efisien, belum lagi kemungkinan terjadinya kesalahan informasi, perhitungan, dan pencatatan barang sehingga mengakibatkan hilangnya keuntungan. Aplikasi *Point of sale* membantu proses transaksi lebih cepat dan juga efektif dalam mengelola sistem dan persediaan barang[3]. Namun, aplikasi

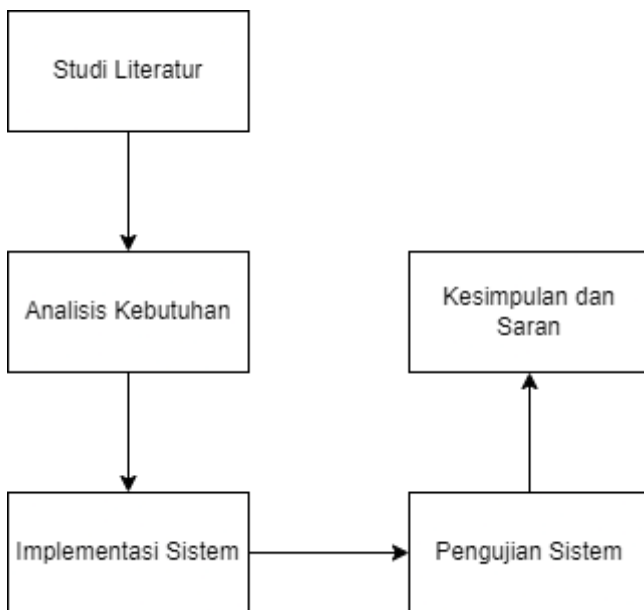
berbasis web merupakan celah besar yang rentan terhadap peretasan dan sabotase data.

Sebagai solusi dari masalah tersebut, penelitian ini berfokus pada penerapan teknologi *blockchain* pada sistem transaksi. Teknologi *Blockchain* adalah teknologi yang digunakan sebagai sistem penyimpanan digital atau database dengan menerapkan sistem enkripsi[4]. *Block* artinya kelompok dan *chain* artinya rantai. Arti di balik nama teknologi ini mencerminkan cara kerja *blockchain*. Dimana teknologi ini menggunakan sumber daya komputer untuk membuat blok-blok yang terhubung (*chain*). Blok yang terhubung kemudian digunakan untuk mengeksekusi transaksi. Teknologi *Blockchain* direalisasikan dengan konsep desentralisasi informasi dalam pengolahan datanya [5]. *Blockchain* mampu berjalan pada algoritma komputer tanpa dikendalikan oleh sistem tertentu[6]. Sifatnya yang terdesentralisasi berarti bahwa teknologi ini tidak mempunyai satu kekuasaan dengan kendali penuh. Teknologi ini menggunakan sistem verifikasi oleh para validator sebelum transaksi dieksekusi. *Blockchain* memiliki keunggulan enkripsi, oleh karena itu, keamanan transaksi tidak menjadi masalah.

Dalam tugas akhir ini, peneliti membuat aplikasi *Point of sale* berbasis website dengan menerapkan teknologi *blockchain* pada sistem transaksinya. Aplikasi ini dibuat dengan menggunakan platform Ethereum dan menggunakan bahasa pemrograman Solidity untuk membuat *smart contract*. *Smart contract* dianalogikan dengan kontrak umum di dunia nyata namun berbentuk digital dan dapat dijalankan sendiri [7]. Untuk memfasilitasi proses pengembangan dan simulasi *smart contract*, peneliti menggunakan *framework* Truffle dan Ganache sebagai jaringan pengujian. Selain itu, peneliti juga menggunakan React.js untuk membuat antarmuka pengguna interaktif. Dengan menggabungkan semua elemen ini, peneliti berharap dapat menciptakan aplikasi yang aman, efisien, dan ramah pengguna untuk transaksi berbasis *blockchain*.

II. METODE PENELITIAN

Rancangan penelitian penerapan teknologi *blockchain* pada sistem transaksi aplikasi *point of sales* memiliki beberapa tahapan yang perlu dilakukan seperti pada gambar 1.



Gbr 1. Rancangan penelitian

A. Studi Literatur

Studi literatur adalah tahap pertama yang dilakukan pada rancangan penelitian. Peneliti mengumpulkan informasi dan data yang relevan dengan penerapan teknologi *blockchain* pada sistem transaksi. Mencakup Solidity sebagai bahasa pemrograman utama untuk pengembangan *smart contract*. Juga mengenai konsep metode konsensus *proof of stake* yang akan diimplementasikan pada penelitian ini. Sumber referensinya meliputi buku-buku, ebook, jurnal penelitian serta sumber-sumber lain seperti internet.

B. Analisis Kebutuhan

Untuk memastikan bahwa penelitian akan berjalan dengan lancar, maka dibutuhkan sistem yang bisa untuk mendukung penelitian ini. Dengan itu analisis kebutuhan menjadi tahapan yang diperlukan dalam rancangan penelitian ini. Adapun kebutuhan yang diperlukan pada penelitian ini yaitu sebagai berikut:

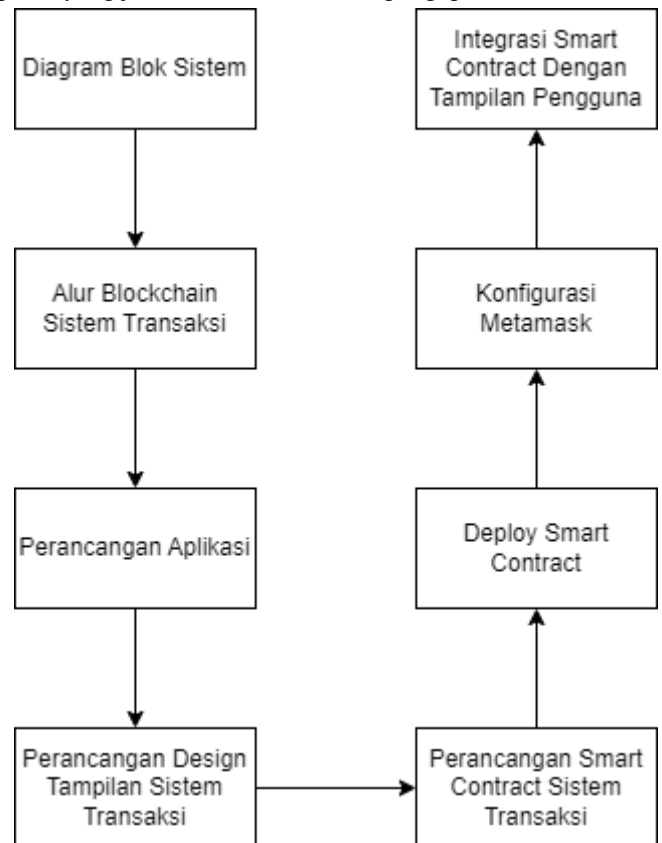
- 1) Kebutuhan fungsional
 - a. Sistem dapat menyimpan data dengan aman
 - b. Sistem dapat menambahkan data
 - c. Sistem dapat mengedit data
 - d. Sistem dapat menampilkan data
- 2) Kebutuhan non fungsional
 - a. Sistem menyediakan tampilan antarmuka yang mudah dipahami oleh pengguna
 - b. Sistem tidak ada *bug* dan berjalan dengan baik

C. Implementasi Sistem

Pada tahap implementasi sistem, peneliti akan merancang dan mengimplementasikan pembangunan sebuah aplikasi *point of sale* yang menerapkan teknologi *blockchain* pada sistem transaksinya.

1) Diagram blok sistem

Diagram blok sistem membantu memetakan arsitektur aplikasi secara keseluruhan, menentukan interaksi antar komponen, dan memastikan bahwa setiap elemen memiliki peran yang jelas dalam keseluruhan pengoperasian sistem.

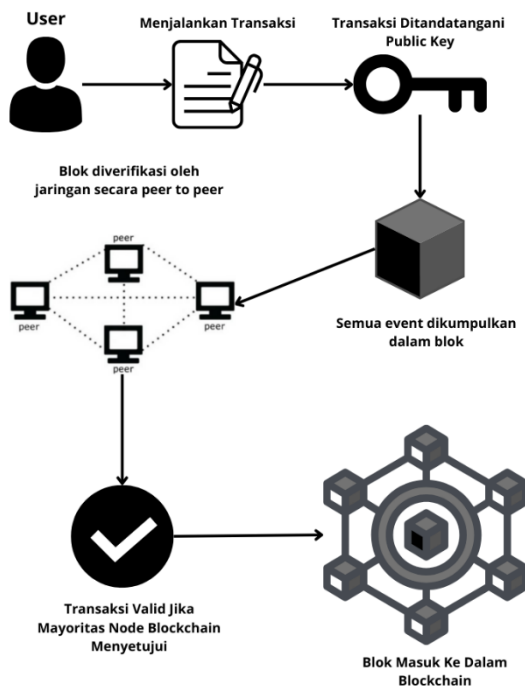


Gbr 2. Alur perancangan sistem

Seperti alur yang ada pada gambar 2, setelah pembuatan diagram blok sistem, selanjutnya akan dijelaskan terlebih dahulu mengenai alur *blockchain* pada sistem transaksi. Setelah diketahui tentang alur *blockchain* yang berjalan pada sistem transaksi, maka akan dilanjutkan dengan tahapan yang akan dilakukan selanjutnya dalam penelitian ini. Tahapan selanjutnya adalah perancangan aplikasi *point of sales* berbasis web. Perancangan aplikasi ini mencakup beberapa tahapan, meliputi perancangan *smart contract* pada fitur transaksi aplikasi, konfigurasi *metamask* untuk mengelola transaksi kripto, dan juga perancangan tampilan pengguna sebagai interaksi antara pengguna dan aplikasi.

2) Alur *blockchain* sistem transaksi

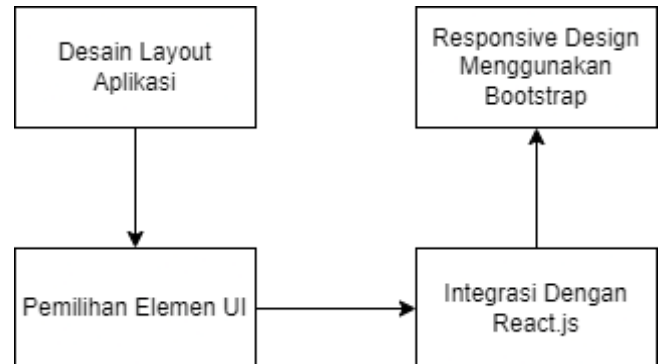
Alur kerja *blockchain* dimulai dengan transaksi yang dilakukan oleh pengguna. Semua event ini dikumpulkan dalam satu blok, dimana ada event lain di waktu yang bersamaan. Ketika sebuah blok terisi penuh, blok tersebut secara kriptografis ditambahkan ke *blockchain*. Setiap blok memiliki hash unik yang mengidentifikasinya dan juga terkait dengan ringkasan blok sebelumnya, membentuk *blockchain* yang tidak terputus[8]. Selanjutnya, jaringan *peer-to-peer* memverifikasi setiap blok yang ditambahkan ke *blockchain*. Kontrol ini dilakukan oleh *node* jaringan atau komputer yang memverifikasi kebenaran transaksi dan memastikan konsistensi data. Jika mayoritas *node* dalam jaringan setuju bahwa suatu blok valid, blok tersebut dianggap valid dan termasuk dalam *blockchain*[9]. Terakhir, setelah suatu transaksi dikonfirmasi dan dimasukkan dalam blok, transaksi tersebut menjadi permanen dan tidak dapat diubah. Data dalam blok tidak dapat dimanipulasi, karena setiap perubahan dalam satu blok akan mempengaruhi semua blok berikutnya, sehingga membuat sistem menjadi sangat aman. Dengan demikian, aliran *blockchain* memberikan keamanan, transparansi, dan keandalan dalam sistem transaksi yang diperlukan dalam aplikasi. Gambar 3 menjelaskan mengenai alur *blockchain* sistem transaksi.



Gbr 3. Alur *blockchain* sistem transaksi

3) Perancangan *design* tampilan sistem transaksi

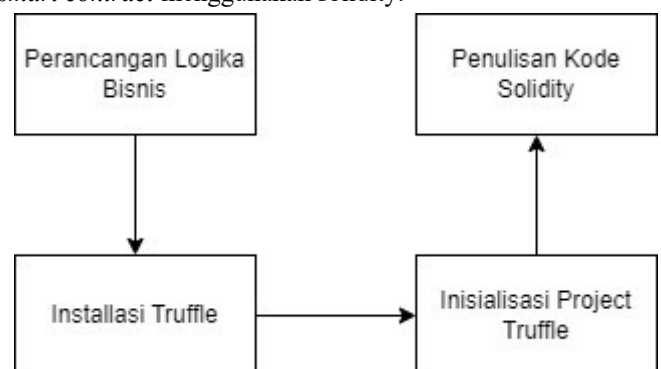
Pada tahap perancangan tampilan pengguna, peneliti akan merancang tampilan antarmuka pengguna yang responsif dan user friendly untuk aplikasi *point of sale* (POS) berbasis website. Proses perancangan meliputi tahapan-tahapan seperti melakukan desain layout aplikasi, pemilihan elemen UI yang sesuai kebutuhan, melakukan integrasi dengan *framework* React.js seperti yang dijelaskan pada gambar 4.



Gbr 4. Alur perancangan design tampilan sistem transaksi

4) Perancangan *smart contract* sistem transaksi

Perancangan *smart contract* untuk fitur transaksi pada aplikasi *point of sales* berbasis website akan dibuat menggunakan platform *blockchain* ethereum dengan bahasa pemrograman solidity dan juga *framework* Truffle untuk mendukung pengembangan serta pengujian *smart contract*. Seperti pada gambar 5, perancangan *smart contract* dimulai dengan perancangan logika bisnis pada fitur transaksi aplikasi *point of sale*. Setelah logika bisnis sudah sesuai dengan kebutuhan, maka selanjutnya adalah penulisan kode program *smart contract* menggunakan solidity.

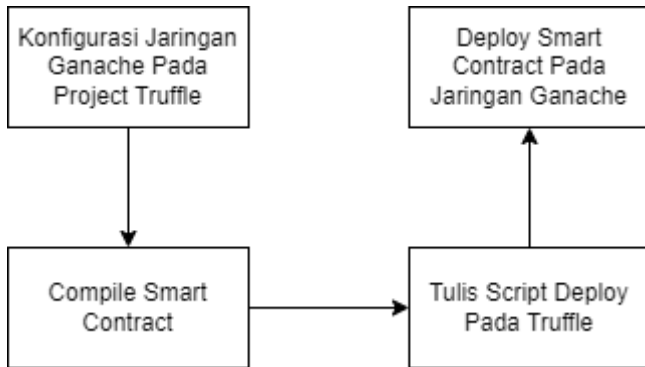


Gbr 5. Alur perancangan smart contract

5) *Deploy smart contract*

Agar *smart contract* dapat berjalan dan diintegrasikan dengan frontend, *smart contract* harus *deploy* terlebih dahulu. Tahap *deploy* diperlukan untuk mengunggah dan mengaktifkan *smart contract* agar berjalan pada jaringan *blockchain*. Gambar 6 menjelaskan mengenai proses *deploy*

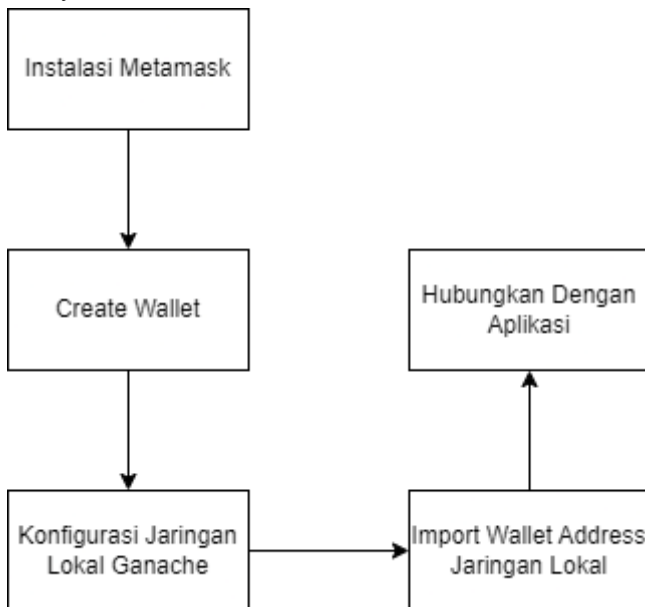
smart contract yang dilakukan dengan menggunakan *framework* truffle pada jaringan ganache.



Gbr 6. Alur deploy smart contract

6) Konfigurasi metamask

Pada tahap ini dilakukan konfigurasi metamask agar wallet kripto yang dibutuhkan sebagai manajemen untuk biaya transaksi *smart contract* dapat berinteraksi dengan aplikasi *Point of sale* (POS) berbasis website. Setiap proses transaksi pada *smart contract* membutuhkan biaya transaksi atau disebut juga gas fee[10], maka diperlukan konfigurasi metamask untuk dapat terkoneksi dengan *address* wallet kripto yang terdapat mata uang ETH di dalamnya. Gambar 7 menjelaskan tentang alur konfigurasi metamask pada aplikasi *point of sale*.

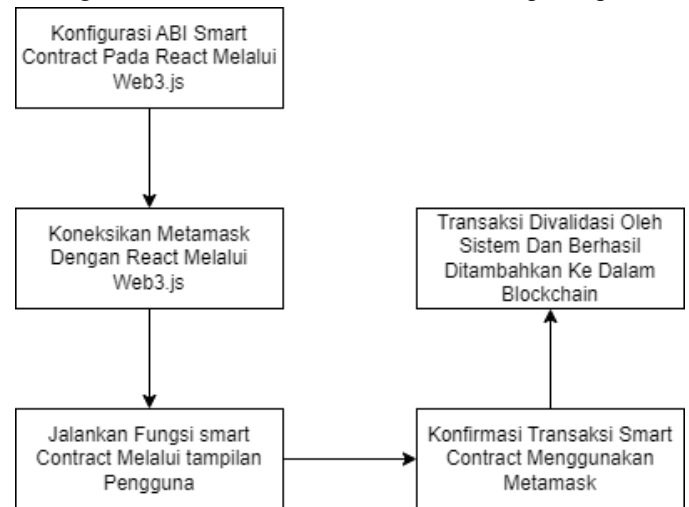


Gbr 7. Alur konfigurasi metamask

7) Integrasi *smart contract* dengan tampilan pengguna

Pada tahap ini, berfokus pada proses integrasi *smart contract* dengan tampilan pengguna agar *user* dapat berinteraksi dengan fungsi-fungsi yang sudah dibuat menggunakan *smart contract* pada aplikasi *point of sale*

(POS). Gambar 8 menjelaskan tentang proses integrasi *smart contract* dengan tampilan pengguna. Tahap integrasi *smart contract* dengan tampilan pengguna diharapkan dapat meningkatkan keamanan dan validitas transaksi pada aplikasi.



Gbr 8. Alur integrasi smart contract dengan tampilan pengguna

D. Pengujian Sistem

Pada tahap pengujian sistem, terdapat 2 jenis pengujian yang akan dilakukan pada penelitian ini yaitu:

1) *Smart contract testing*

Untuk memastikan bahwa fungsi-fungsi yang terdapat pada *smart contract* yang sudah dibuat berjalan dengan baik, diperlukan adanya pengujian untuk *smart contract*. Pengujian ini dilakukan dengan menggunakan *framework* truffle.

2) *Functional testing*

Pengujian fungsional adalah bagian penting dari proses pengembangan aplikasi. Functional testing adalah jenis pengujian aplikasi yang berfokus pada fungsionalitas suatu sistem[11]. Tujuannya adalah memastikan bahwa sistem sudah berjalan sesuai dengan yang diharapkan.

III. HASIL DAN PEMBAHASAN

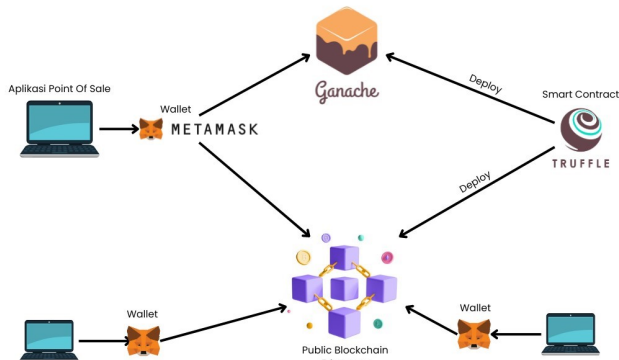
Bab ini akan diuraikan hasil dari rancangan penerapan teknologi *blockchain* pada sistem transaksi aplikasi *point of sale*. Selain itu, hasil pengujian sistem dan analisis mengenai hasil tersebut juga akan dibahas pada bab ini.

A. Implementasi Rancangan Penelitian

Pada tahap ini, dilakukan pembangunan aplikasi *point of sale* berbasis web dengan menerapkan teknologi *blockchain* pada sistem transaksinya.

1) Skema infrastruktur jaringan *blockchain* pada aplikasi

Sebelum masuk ke dalam tahap implementasi, akan dijelaskan dulu mengenai infrastruktur jaringan *blockchain* pada aplikasi.



Gbr 9. Infrastruktur jaringan blockchain pada aplikasi

Seperti yang dapat dilihat pada gambar 9 yang menjelaskan mengenai infrastruktur jaringan *blockchain*, *smart contract* yang dibuat menggunakan *framework* truffle harus *dideploy* terlebih dahulu ke jaringan *blockchain* agar dapat digunakan pada aplikasi *client*, pada penelitian ini menggunakan ganache sebagai jaringan *blockchain* lokal. Aplikasi *client* juga harus menggunakan metamask sebagai penghubung antara aplikasi *client* dengan *smart contract* yang sudah *dideploy*, setiap transaksi yang dilakukan harus melalui verifikasi metamask. Penerapannya pada *real project* yaitu *smart contract* yang dibuat menggunakan *framework* truffle harus *dideploy* terlebih dahulu pada jaringan *public blockchain* yang dapat diakses oleh siapapun.

B. Hasil Implementasi

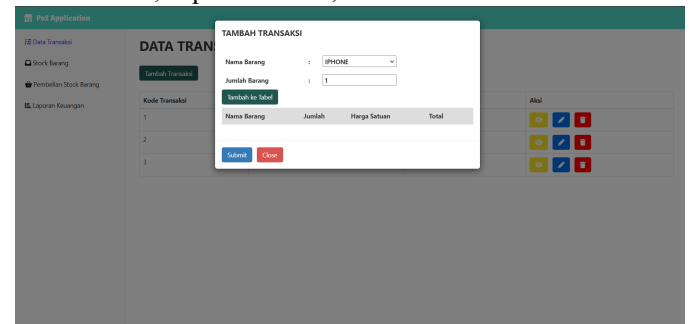
1) Frontend

Frontend merupakan *user interface* aplikasi *point of sale* berbasis website yang menangani interaksi antara *user* dengan sistem. Frontend memudahkan *user* untuk mengetahui fitur, fungsi dan alur dari sistem aplikasi *point of sale*. Berikut hasil dari frontend aplikasi *point of sale* berbasis website.



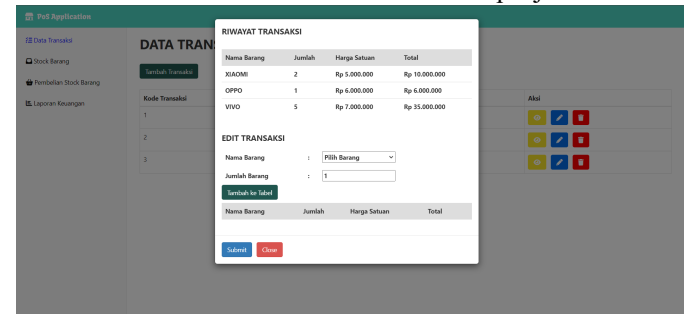
Gbr 10. Halaman utama transaksi penjualan

Seperti yang dapat dilihat pada gambar 10, halaman utama data transaksi penjualan menampilkan tabel riwayat transaksi penjualan yang sudah dibuat oleh *user* dan beberapa aksi yang dapat dilakukan oleh *user* meliputi tambah transaksi, edit transaksi, hapus transaksi, dan lihat detail transaksi.



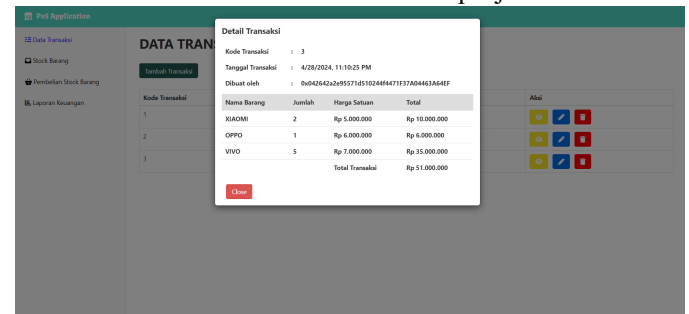
Gbr 11. Modal tambah transaksi penjualan

Gambar 11 merupakan tampilan modal yang muncul ketika *user* melakukan aksi tambah transaksi penjualan.



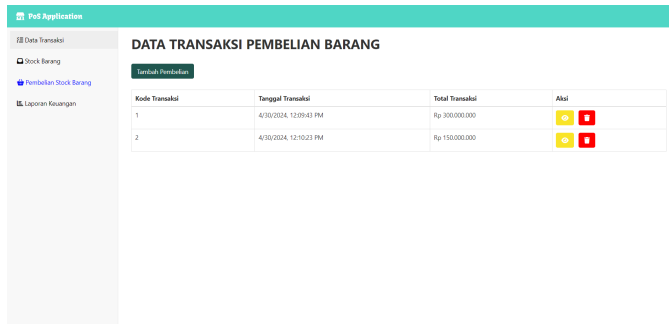
Gbr 12. Modal edit transaksi penjualan

Gambar 12 merupakan tampilan modal yang muncul ketika *user* melakukan aksi edit transaksi penjualan.



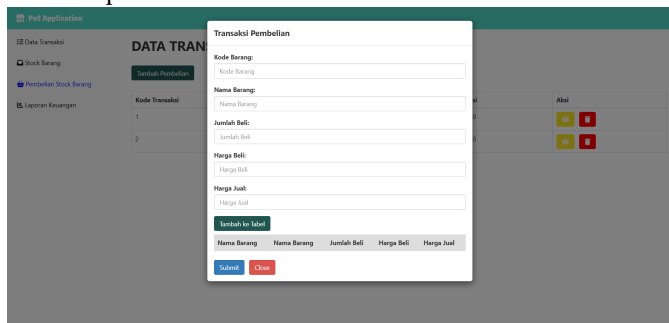
Gbr 13. Modal detail transaksi penjualan

Gambar 13 merupakan tampilan modal yang muncul ketika *user* melakukan aksi lihat detail transaksi penjualan.



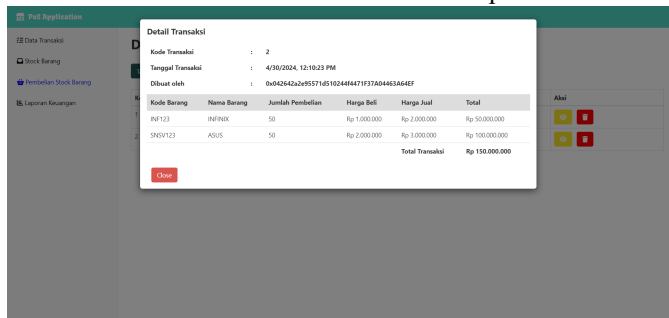
Gbr 14. Halaman utama transaksi pembelian

Seperti yang dapat dilihat pada gambar 14, halaman utama data transaksi pembelian barang menampilkan tabel riwayat transaksi pembelian yang sudah dibuat oleh *user* dan beberapa aksi yang dapat dilakukan oleh *user* meliputi tambah transaksi pembelian, hapus transaksi pembelian, dan lihat detail transaksi pembelian.



Gbr 15. Modal tambah transaksi pembelian

Gambar 15 merupakan tampilan modal yang muncul ketika *user* melakukan aksi tambah transaksi pembelian.



Gbr 16. Modal detail transaksi pembelian

Gambar 16 merupakan tampilan modal yang muncul ketika *user* melakukan aksi lihat detail transaksi pembelian.

2) Backend

Backend pada sistem ini berupa *smart contract* yang memiliki fungsi yaitu menambahkan transaksi, mengedit transaksi, menghapus transaksi, dan melihat data transaksi. Fungsi-fungsi pada *smart contract* yang dijalankan oleh *user* akan terekam pada ganache. Berikut merupakan hasil *smart contract* yang terekam pada ganache.



Gbr 17. Tambah transaksi ganache

Gambar 17 merupakan hasil *smart contract* yang berisikan rekaman informasi dari fungsi tambah transaksi yang sudah dijalankan oleh *user*.



Gbr 18. Edit transaksi ganache

Gambar 18 merupakan hasil *smart contract* yang berisikan rekaman informasi dari fungsi edit transaksi yang sudah dijalankan oleh *user*.



Gbr 19. Hapus transaksi ganache

Gambar 19 merupakan hasil *smart contract* yang berisikan rekaman informasi dari fungsi hapus transaksi yang sudah dijalankan oleh *user*.

C. Pengujian Sistem

Pengujian sistem transaksi aplikasi *point of sale* pada penelitian ini dilakukan dengan menggunakan metode *smart contract testing* dan fungsional *testing*.

1) Smart contract testing

Smart contract testing merupakan pengujian berbasis *unit testing* yang berfungsi untuk memastikan bahwa setiap fungsi

yang dibuat pada *smart contract* berjalan sesuai dengan yang diharapkan tanpa adanya kesalahan sistem.

```
it("should add and complete a transaction", async () => {
  const itemIds = [1, 2];
  const amounts = [3, 2];
  await transaksiInstance.tambahDanSelesaikanTransaksi(itemIds, amounts, {
    from: accounts[0],
  });
  const transaction = await transaksiInstance.getDataTransaksi(
    accounts[0],
    1
  );
  assert.equal(transaction[2].length, 2, "Transaction items not added");
});
```

Gbr 20. Smart contract testing tambah transaksi penjualan

Gambar 20 merupakan kode pengujian fungsi tambah transaksi, data transaksi baru diambil untuk verifikasi bahwa item telah ditambahkan dalam transaksi dengan benar dan memastikan bahwa panjang array *transaction* sesuai dengan jumlah item yang ditambahkan, sehingga dapat dikonfirmasi keberhasilan penambahan dan penyelesaian transaksi.

```
it("should edit a transaction", async () => {
  const newAmounts = [4, 1];
  await transaksiInstance.editTransaksi(1, [1, 2], newAmounts, {
    from: accounts[0],
  });
  const transaction = await transaksiInstance.getDataTransaksi(
    accounts[0],
    1
  );
  assert.equal(
    transaction[3][0].jumlah,
    newAmounts[0],
    "Transaction not edited"
  );
});
```

Gbr 21. Smart contract testing edit transaksi penjualan

Gambar 21 merupakan kode pengujian edit transaksi, data transaksi diperbarui untuk memeriksa apakah jumlah item pertama dalam transaksi sesuai dengan nilai baru yang telah diubah. Hal ini dilakukan untuk memastikan bahwa transaksi telah berhasil diedit sesuai dengan yang diharapkan.

```
it("should delete a transaction", async () => {
  await transaksiInstance.deleteTransaksi(1, { from: accounts[0] });
  try {
    await transaksiInstance.getDataTransaksi(accounts[0], 1);
  } catch (error) {
    assert(error.message.includes("Transaksi tidak ditemukan"));
    return;
  }
  assert(false, "Transaction still exists");
});
```

Gbr 22. Smart contract testing hapus transaksi penjualan

Gambar 22 merupakan kode pengujian hapus transaksi, dilakukan verifikasi bahwa transaksi benar-benar telah dihapus dengan mengambil data transaksi. Jika data transaksi tidak ditemukan maka pengujian *delete transaction* berhasil.

```
it("should return transaction data with item details", async () => {
  const transactionId = 1;
  try {
    const transactionData = await transaksiInstance.getDataTransaksi(
      accounts[0],
      transactionId
    );
    assert.fail("Expected transaction not found error");
  } catch (error) {
    assert(error.message.includes("Transaksi tidak ditemukan"), "Unexpected error message");
  }
});
```

Gbr 23. Smart contract testing get data transaksi penjualan

Gambar 23 merupakan kode pengujian pengambilan data transaksi, tujuannya adalah untuk memverifikasi bahwa pengambilan data transaksi dengan *detail item* berhasil dilakukan. Jika tidak ada pesan *error* yang muncul maka pengujian pada fungsi *getDataTransaksi* berhasil.

```
it("should add and finalize a purchase transaction", async () => {
  const _kodes = ["Kode1", "Kode2"];
  const _nimas = ["Nama1", "Nama2"];
  const _jumlahs = [2, 3];
  const _hargaBelis = [100, 150];
  const _hargaJuals = [120, 180];

  await transaksiInstance.tambahDanSelesaikanTransaksiPembelian(
    _kodes,
    _nimas,
    _jumlahs,
    _hargaBelis,
    _hargaJuals,
    { from: accounts[0] }
  );

  const pembelianDetails = await transaksiInstance.getPembelianDetails(
    accounts[0],
    1
  );

  assert.equal(pembelianDetails.length, _kodes.length, "Jumlah detail transaksi tidak sesuai");
});
```

Gbr 24. Smart contract testing tambah transaksi pembelian

Gambar 24 merupakan kode pengujian tambah transaksi pembelian, tujuannya adalah untuk menambahkan dan menyelesaikan sebuah transaksi pembelian. Setelah fungsi ini dijalankan, selanjutnya diambil *detail* pembelian, kemudian diperiksa apakah jumlah *detail* transaksi sesuai dengan jumlah data yang disediakan, seperti kode barang dan nama barang. Jika jumlah *detail* transaksi sesuai dengan data yang diharapkan maka pengujian akan berhasil.

```
it("should delete a purchase transaction", async () => {
  const kodeTransaksi = 1;
  const transactionBefore = await transaksiInstance.getPembelianData(
    accounts[0],
    kodeTransaksi
  );
  expect(transactionBefore.toNumber()).to.not.equal(0);

  await transaksiInstance.deletePurchase(kodeTransaksi, { from: accounts[0] });

  const transactionAfter = await transaksiInstance.getPembelianData(
    accounts[0],
    kodeTransaksi
  );
  expect(transactionAfter.toNumber()).to.equal(0);
});
```

Gbr 25. Smart contract testing hapus transaksi pembelian

Gambar 25 merupakan kode pengujian hapus transaksi pembelian, data transaksi pembelian akan diperiksa menggunakan fungsi *getPembelianData* dan diharapkan nilai transaksi setelah penghapusan sama dengan 0, yang menunjukkan bahwa transaksi pembelian telah berhasil dihapus dan pengujian berhasil.

```
it("should return all purchases for a buyer", async () => {
  const result = await transaksiInstance.getAllPurchases(accounts[0]);

  assert.isArray(result[0], "transactionCodes should be an array");
  assert.isArray(result[1], "transactionDates should be an array");
  assert.isArray(result[2], "transactionTotals should be an array");
  assert.isArray(result[3], "transactionItemIds should be an array");
  assert.isArray(result[4], "transactionItemDetails should be an array");

  assert.equal(
    result[0].length,
    result[1].length,
    "Length of transactionCodes and transactionDates should be the same"
  );
  assert.equal(
    result[0].length,
    result[2].length,
    "Length of transactionCodes and transactionTotals should be the same"
  );
  assert.equal(
    result[0].length,
    result[3].length,
    "Length of transactionCodes and transactionItemIds should be the same"
  );
  assert.equal(
    result[0].length,
    result[4].length,
    "Length of transactionCodes and transactionItemDetails should be the same"
  );
});
```

Gbr 26. Smart contract testing get data transaksi pembelian

Gambar 26 merupakan kode pengujian mengambil data transaksi pembelian yang bertujuan untuk memeriksa apakah fungsi `getAllPurchases` mengembalikan semua pembelian yang terkait dengan *address user* yang dijadikan parameter. Setelah *script test* untuk *smart contract* sistem transaksi selesai dibuat, maka langkah selanjutnya adalah melakukan *smart contract test* dengan menuliskan *truffle test* pada terminal. Jika test berjalan lancar dan tidak ada error maka akan menghasilkan output sebagai berikut.

```
Contract: Transaksi
✓ should add and complete a transaction (410ms)
✓ should edit a transaction (699ms)
✓ should delete a transaction (521ms)
✓ should return transaction data with item details
✓ should return item details for a transaction
✓ should add and finalize a purchase transaction (1642ms)
✓ should delete a purchase transaction (483ms)
✓ should return all purchases for a buyer
```

Gbr 27. Output smart contract testing

Gambar 27 menunjukkan bahwa seluruh asersi pada pengujian yang telah dilakukan berhasil dipenuhi, yang berarti semua fungsi pada *smart contract* sistem transaksi yang diuji berjalan dengan normal dan tidak ada kendala yang menyebabkan kesalahan sistem.

2) Functional testing

Tujuan utama dari fungsional *testing* adalah untuk menguji apakah sistem tersebut melakukan apa yang seharusnya dilakukan, sesuai dengan kebutuhan dan harapan pengguna[12]. Pengujian dilakukan pada *browser* google chrome yang sudah *terintegrasi* dengan metamask dengan menggunakan jaringan Ethereum lokal.

TABEL I

HASIL PENGUJIAN FITUR CONNECT WALLET APLIKASI DENGAN METAMASK

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian
1	Password	Password:	Tombol buka	Tombol

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian
	metamask tidak diisi	(kosong)	pada metamask akan tidak dapat di klik	pada metamask tidak dapat di klik (berhasil)
2	Password metamask diisi dengan password yang salah	Password: (salah)	Sistem akan menolak koneksi aplikasi dengan metamask dan menampilkan pesan kesalahan password	Metamask gagal terkoneksi dengan menampilkan pesan password salah (berhasil)
3	Password metamask diisi dengan benar	Password: (benar)	Metamask akan sukses terkoneksi dengan aplikasi dan semua fitur aplikasi dapat dijalankan	Metamask terkoneksi dengan aplikasi dan fitur pada aplikasi dapat dijalankan (berhasil)

Hasil pengujian pada tabel I menunjukkan bahwa hasil pengujian berhasil pada seluruh skenario pengujian fitur *connect wallet* aplikasi dengan metamask.

TABEL III

HASIL PENGUJIAN FITUR TAMBAH TRANSAKSI PENJUALAN

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian
1	Tidak memilih produk dan mengosongi jumlah produk	Produk: (kosong) Jumlah: (kosong)	Sistem tidak akan menanggapi <i>request</i> dari <i>user</i>	Sistem tidak menanggapi <i>request user</i> (berhasil)
2	Tidak memilih produk dan mengisi jumlah produk	Produk: (kosong) Jumlah: (kosong)	Sistem tidak akan menanggapi <i>request</i> dari <i>user</i>	Sistem tidak menanggapi <i>request user</i> (berhasil)
3	Memilih produk dan tidak mengisi jumlah produk	Produk: (dipilih) Jumlah: (kosong)	Sistem akan berjalan dan transaksi penjualan akan terbuat dengan total harga 0 dan stok produk tidak berkurang	Transaksi penjualan terbuat dengan total harga 0 dan stok produk pada inventaris tidak berkurang (berhasil)
4	Memilih produk dan mengisi jumlah	Produk: (dipilih) Jumlah: (diisi)	Sistem akan berjalan dan transaksi penjualan	Transaksi penjualan terbuat dengan

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian
	produk		akan terbuat sesuai dengan produk dan jumlah yang dipilih	produk dan jumlah yang telah dipilih (berhasil)

Hasil pengujian pada tabel II menunjukkan bahwa hasil pengujian berhasil pada seluruh skenario pengujian fitur tambah transaksi penjualan.

TABEL IIIII
HASIL PENGUJIAN FITUR EDIT TRANSAKSI PENJUALAN

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian
1	Tidak memilih produk dan mengosongi jumlah produk	Produk: (kosong) Jumlah: (kosong)	Sistem tidak akan menanggapi <i>request</i> dari <i>user</i>	Sistem tidak menanggapi <i>request user</i> (berhasil)
2	Tidak memilih produk dan mengisi jumlah produk	Produk: (kosong) Jumlah: (kosong)	Sistem tidak akan menanggapi <i>request</i> dari <i>user</i>	Sistem tidak menanggapi <i>request user</i> (berhasil)
3	Memilih produk dan mengisi jumlah produk	Produk: (dipilih) Jumlah: (kosong)	Sistem akan berjalan dan transaksi penjualan akan diubah dengan total harga 0 dan stok produk tidak berkurang	Transaksi penjualan diubah dengan total harga 0 dan stok produk tidak berkurang (berhasil)
4	Memilih produk dan mengisi jumlah produk	Produk: (dipilih) Jumlah: (diisi)	Sistem akan berjalan dan transaksi penjualan akan diubah sesuai dengan produk dan jumlah yang dipilih	Transaksi penjualan diubah dengan produk dan jumlah yang telah dipilih (berhasil)

Hasil pengujian pada tabel III menunjukkan bahwa hasil pengujian berhasil pada seluruh skenario pengujian fitur edit transaksi penjualan.

TABEL IVV
HASIL PENGUJIAN FITUR HAPUS TRANSAKSI PENJUALAN

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian
1	Memilih dan menghapus data transaksi dengan menekan tombol hapus	Klik tombol hapus pada data transaksi yang	Sistem akan melakukan penghapusan data transaksi penjualan	Sistem menghapus data transaksi penjualan (berhasil)

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian
		dipilih		

Hasil pengujian pada tabel IV menunjukkan bahwa hasil pengujian berhasil pada seluruh skenario pengujian fitur hapus transaksi penjualan.

TABEL V
HASIL PENGUJIAN FITUR LIHAT DETAIL TRANSAKSI PENJUALAN

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian
1	Melihat detail transaksi penjualan dengan menekan tombol detail	Klik tombol detail pada transaksi penjualan yang dipilih	Sistem akan menampilkan detail transaksi penjualan	Sistem menampilkan detail transaksi penjualan (berhasil)

Hasil pengujian pada tabel V menunjukkan bahwa hasil pengujian berhasil pada seluruh skenario pengujian fitur lihat detail transaksi penjualan.

TABEL VI
HASIL PENGUJIAN FITUR TAMBAH TRANSAKSI PEMBELIAN

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian
1	Mengosongi semua <i>form input</i>	<i>Form input:</i> (kosong)	Sistem tidak akan menanggapi dan tombol tambah akan <i>disable</i>	Sistem tidak menanggapi dan tombol tambah <i>terdisable</i> (berhasil)
2	Mengisi sebagian <i>form input</i>	<i>Form input:</i> (diisi sebagian)	Sistem tidak akan menanggapi dan tombol tambah akan <i>disable</i>	Sistem tidak menanggapi dan tombol tambah <i>terdisable</i> (berhasil)
3	Mengisi semua <i>form input</i> dan kolom jumlah beli, harga beli, dan harga jual dengan tipe data selain <i>number</i>	Kode barang: (string) Nama barang: (string) Jumlah: (string) Harga beli: (string) Harga jual: (string)	Sistem akan memberikan output NaN pada kolom jumlah beli, harga beli, dan harga jual dan transaksi pembelian tidak dapat dijalankan	Sistem memberikan output NaN pada kolom jumlah beli, harga beli, dan harga jual dan transaksi pembelian tidak dapat dijalankan (berhasil)
4	Mengisi semua <i>form input</i> dengan tipe data yang sesuai	<i>Form input:</i> (diisi semua dengan tipe data yang sesuai)	Sistem akan berjalan dan transaksi pembelian akan terbuat sesuai dengan <i>input user</i>	Transaksi pembelian terbuat sesuai dengan <i>input user</i> (berhasil)

Hasil pengujian pada tabel VI menunjukkan bahwa hasil pengujian berhasil pada seluruh skenario pengujian fitur tambah transaksi pembelian.

TABEL VII
HASIL PENGUJIAN FITUR EDIT TRANSAKSI PEMBELIAN

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian
1	Memilih dan menghapus data transaksi pembelian dengan menekan tombol hapus	Klik tombol hapus pada data transaksi pembelian yang dipilih	Sistem akan melakukan penghapusan data transaksi pembelian	Sistem menghapus data transaksi pembelian (berhasil)

Hasil pengujian pada tabel VII menunjukkan bahwa hasil pengujian berhasil pada seluruh skenario pengujian fitur hapus transaksi pembelian.

TABEL VIII
HASIL PENGUJIAN FITUR LIHAT DETAIL TRANSAKSI PEMBELIAN

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian
1	Melihat detail transaksi pembelian dengan menekan tombol detail	Klik tombol detail pada transaksi pembelian yang dipilih	Sistem akan menampilkan detail transaksi pembelian	Sistem menampilkan detail transaksi pembelian (berhasil)

Hasil pengujian pada tabel VIII menunjukkan bahwa hasil pengujian berhasil pada seluruh skenario pengujian fitur lihat detail transaksi pembelian.

IV. KESIMPULAN

Penelitian ini berhasil menjelaskan bahwa penerapan teknologi *blockchain* pada sistem transaksi *point of sale* (POS) menunjukkan keamanan dan transparansi data transaksi. Ketika *smart contract* dijalankan, semua perubahan pada data transaksi secara otomatis dicatat dalam blok terenkripsi, yang kemudian disimpan dalam jaringan *blockchain* terdesentralisasi. Hal ini memastikan bahwa setiap transaksi memiliki rekaman jejak yang jelas dan tidak dapat diubah, sehingga mengurangi risiko manipulasi data. Selain itu, catatan transaksi yang disimpan dalam perangkat lunak Ganache menunjukkan efektivitas *blockchain* dan memastikan transparansi penuh atas perubahan transaksi.

Penerapan metode konsensus *proof of stake* pada teknologi *blockchain* pada aplikasi *point of sale* (POS) ini memberikan transparansi dan keamanan pada sistem transaksi. Metode

proof of stake memungkinkan untuk memverifikasi setiap perubahan transaksi antar *node* jaringan tanpa konsumsi energi yang tinggi. Oleh karena itu, penelitian ini menyimpulkan bahwa penerapan teknologi *blockchain* yang menggunakan metode konsensus *proof of stake* sangat cocok untuk meningkatkan keamanan dan transparansi sistem transaksi pada aplikasi POS.

REFERENSI

- [1] Saputra, A. M. A., Kharisma, L. P. I., Rizal, A. A., Burhan, M. I., & Purnawati, N. W. (2023). *TEKNOLOGI INFORMASI: Peranan TI dalam berbagai bidang*. PT. Sonpedia Publishing Indonesia.
- [2] Prayogi, B. S., Fitri, I., & Nuraini, R. (2022). Aplikasi Point of sale Berbasis Website pada Toko Sembako Tegar. *Jurnal Teknologi Informasi dan Komunikasi*, 6(2), 2022. <https://doi.org/10.35870/jti>
- [3] Siddik, M., & Samsir, S. (2020). Rancang Bangun Sistem Informasi Pos (Point of Sale) Untuk Kasir Menggunakan Konsep Bahasa Pemrograman Orientasi Objek. *JOISIE (Journal Of Information Systems And Informatics Engineering)*, 4(1), 43-48.
- [4] Sunarya, P. A. (2022). Penerapan sertifikat pada sistem keamanan menggunakan teknologi *blockchain*. *Jurnal MENTARI: Manajemen, Pendidikan dan Teknologi Informasi*, 1(1), 58-67.
- [5] Chandra Nugraha, A. (2020). PENERAPAN TEKNOLOGI BLOCKCHAIN DALAM LINGKUNGAN PENDIDIKAN. 4(1).
- [6] Ahmad, J. A., & Dirgahayu, T. (2023). The Role Of Blockchain To Solve Problems Of Digital Right Management (DRM). *Jurnal Teknik Informatika (Jutif)*, 4(1), 85-95.
- [7] Panda, S. K., & Satapathy, S. C. (2021). An Investigation into Smart Contract Deployment on Ethereum Platform Using Web3.js and Solidity Using Blockchain (hlm. 549-561). https://doi.org/10.1007/978-981-16-0171-2_52
- [8] Santoso, J. T. (2023). Teknologi Keamanan Siber (Cyber Security). Penerbit Yayasan Prima Agus Teknik, 1-173.
- [9] Lukita, C., & Faturahman, A. (2022). Perkembangan fintech terhadap crowdfunding dan *blockchain* di era disrupsi 4.0. *Jurnal MENTARI: Manajemen, Pendidikan dan Teknologi Informasi*, 1(1), 9-19.
- [10] Kartiko, H. S., Rismawan, T., & Ruslianto, I. (2023). Implementasi IPFS untuk Mengurangi Gas Fee Smart Contract Ethereum pada Aplikasi Penggalangan Dana. *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 9(2), 195-203.
- [11] Utomo, A., Sutanto, Y., Tiningrum, E., & Susilowati, E. M. (2020). Pengujian Aplikasi Transaksi Perdagangan Menggunakan Black Box Testing Boundary Value Analysis. *Jurnal Bisnis Terapan*, 4(2), 133-140.
- [12] Gunawan, D., Raniri, I. A. A., Setyawan, R. N., & Prasetya, Y. D. (2021). Web-Based Library Information System in Madrasah Ibtidaiyah Negeri Surakarta. *Jurnal Teknik Informatika (Jutif)*, 2(1), 33-41.