

Penerapan Teknologi Blockchain pada Manajemen Stok Barang Aplikasi Point of Sale

Aji Ngafifudin¹, I Made Suartana²

^{1,3} Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya

¹aji.20054@mhs.unesa.ac.id

²madesuartana@unesa.ac.id

Abstrak— Sistem kasir dengan menggunakan aplikasi atau website yang lebih sering dikenal dengan sistem point of sale (POS) dalam perkembangannya memerlukan tingkat keamanan serta transparansi dalam setiap perubahan aksi pada sistem berupa proses masuk keluarnya barang dalam rantai pasok stok barang. Untuk mengatasi permasalahan ini, peneliti membangun sebuah *website point of sale* (POS) dengan mengimplementasikan fungsi logic control pada smart contract yang tersedia dalam teknologi blockchain. Teknologi blockchain dengan konsensus proof of stake dipilih dikarenakan teknologi ini dapat memverifikasi setiap adanya perubahan-perubahan setiap aksi dalam proses masuknya stok barang dalam blok-blok yang telah terenkripsi dalam *ledger* yang terdistribusi pada jaringan *blockchain*. Hasil dari implementasi *blockchain* dapat dilihat dengan menggunakan software *ganache* yang digunakan untuk merekam setiap perubahan aksi pada stok barang. Selain itu, *point of sale* (POS) telah diuji dengan dua jenis *testing*, yaitu *functional testing* untuk menguji keberhasilan setiap aksi yang dilakukan pada *user interface*-nya, dan *unit testing* untuk menguji kelayakan spesifikasi dengan melakukan validasi setiap kode pada smart contract.

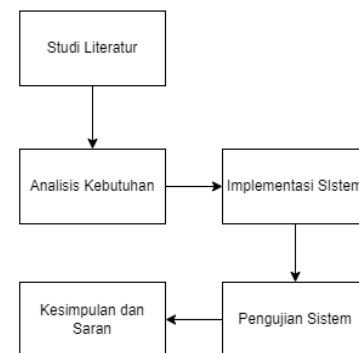
Kata Kunci— Stok barang, point of sale, blockchain, proof of stake, smart contract.

I. PENDAHULUAN

Perkembangan dunia bisnis pada saat ini berkembang pesat, dan salah satu pendorong utamanya adalah adopsi teknologi internet. Fenomena ini tercermin dengan meningkatnya jumlah toko dalam menjual kebutuhan pokok guna memenuhi tuntutan masyarakat modern. Selain itu, perkembangan teknologi internet telah mengakibatkan persaingan bisnis yang semakin ketat di kalangan berbagai toko, yang secara aktif berupaya meningkatkan standar kualitas produk dan layanan yang mereka tawarkan kepada pelanggan [1]. Salah satu contoh penerapan teknologi internet adalah melalui penggunaan sistem *point of sale* (POS), atau yang lebih umum dikenal sebagai sistem kasir. Sistem ini digunakan untuk melakukan segala macam kegiatan yang ada dalam toko tersebut, dimulai dari transaksi penjualan, pembelian, dan retur, khususnya dalam konteks penjualan produk secara langsung di toko fisik [2]. Sistem *point of sale* (POS) tidak hanya berperan dalam memudahkan transaksi penjualan, pembelian, retur, dan sebagainya, tetapi juga membantu secara efektif dalam mengelola stok barang toko. Dalam perkembangannya, *point of sale* (POS) membutuhkan suatu sistem yang dapat memberikan tingkat keamanan dan transparansi tambahan dalam rantai pasok toko. Salah satu teknologi yang dapat diadopsi dalam meningkatkan keamanan *point of sale* (POS) adalah teknologi *blockchain*. *Blockchain* adalah sejenis teknologi buku besar atau *ledger* yang

terdistribusi dan ditingkatkan menjadi sistem penyimpanan yang lengkap dengan menambahkan fungsi *logic control* seperti *smart contract*. Dengan adanya perubahan klasifikasi, mode layanan dan persyaratan 2 aplikasi, bentuk teknologi inti dari *blockchain* menunjukkan perkembangan yang beragam. Selanjutnya, masing-masing lapisan *blockchain* dianalisis dari perspektif prinsip dasar, teknologi terkait dan kemajuan penelitian secara mendalam. Selain itu, pemilihan teknologi dan karakteristik dari proyek-proyek *blockchain* dirangkum secara sistematis [3]. Teknologi *blockchain* dapat memverifikasi adanya perubahan-perubahan keluar masuknya barang dalam blok-blok yang telah terenkripsi di antara partikel-partikel jaringan. Adanya penggunaan *blockchain* dikarenakan kebutuhan yang mendesak dalam meningkatkan keamanan manajemen stok barang dalam sistem POS [4]. Oleh karena itu, peneliti tertarik untuk mengimplementasikan teknologi *blockchain* ke dalam pengembangan manajemen stok barang pada sistem *point of sale* dengan harapan dapat meningkatkan keamanan dalam pengelolaan stok barang.

II. METODOLOGI PENELITIAN



Gbr .1 Alur Rancangan Penelitian

Penelitian ini bertujuan untuk melakukan eksplorasi terhadap implementasi teknologi *blockchain* ke dalam sistem manajemen stok barang aplikasi *point of sale* (POS). Seperti pada gbr. 1, Pertama studi literatur dilakukan guna memahami konsep dasar *blockchain*, penerapan aplikasi di industri manufaktur, dan terintegrasi dengan sistem POS. Lalu, Analisis kebutuhan yang melibatkan identifikasi kebutuhan-kebutuhan dalam manajemen stok yang dapat diatasi dengan *blockchain*, seperti akurasi inventaris, dan transparansi rantai pasok barang [5]. Berikutnya, Implementasi sistem aplikasi *point of sale* mencakup pengembangan *prototype* yang terintegrasi *blockchain* dengan aplikasi POS untuk melacak pergerakan stok barang. Kemudian, pengujian sistem dilakukan guna melakukan evaluasi efisiensi solusi yang

diusulkan. Dan, terdapat kesimpulan dari penelitian ini adalah yang dimana implementasi teknologi *blockchain* dapat meningkatkan transparansi manajemen stok barang di aplikasi POS [6]. Saran untuk penelitian-penelitian berikutnya yaitu melanjutkan eksperimen sistem dengan skala jauh lebih kompleks dan mempertimbangkan beberapa faktor regulasi dalam implementasi *blockchain* ke dalam lingkungan bisnis.

Pada gbr.1 dapat diketahui tahapan-tahapan pada penelitian ini diantaranya, yaitu:

1. Studi Literatur

Dalam tahap studi literatur, penelitian ini melibatkan eksplorasi mendalam terhadap sejumlah konsep kunci yang terkait dengan pengembangan aplikasi *point of sale* (POS) berbasis *website* menggunakan teknologi *blockchain*. Studi literatur mencakup analisis tentang konsep dan prinsip dasar teknologi *blockchain*, termasuk pemahaman mendalam tentang Solidity sebagai bahasa pemrograman utama untuk pengembangan *smart contract*, serta konsep *proof of stake* sebagai metode konsensus yang mendasari jaringan *blockchain* [7]. Selain itu, penelitian ini juga memperhatikan integrasi teknologi terkait, seperti Metamask untuk manajemen dompet kripto, React sebagai framework frontend untuk pengembangan antarmuka pengguna yang dinamis, serta penggunaan web3.js untuk interaksi dengan jaringan Ethereum. Melalui studi literatur ini, pemahaman mendalam tentang prinsip-prinsip teknis dan praktis yang mendasari pengembangan aplikasi POS berbasis *website* dengan menggunakan teknologi *blockchain* telah diperoleh, memandu penelitian lebih lanjut dalam tahap perancangan dan pengujian sistem

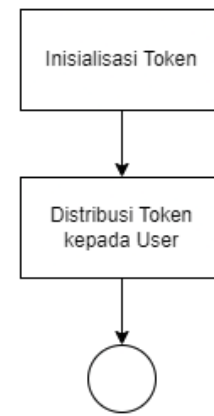
2. Analisis Kebutuhan

1) Kebutuhan Fungsional dan Non-Fungsional

TABEL I
KEBUTUHAN FUNGSIONAL DAN NON-FUNGSIONAL

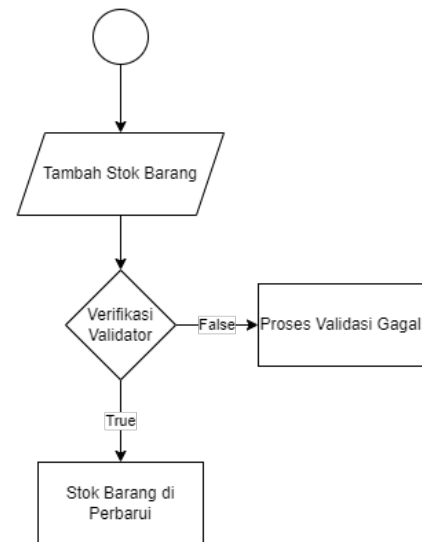
Fungsional	Non-Fungsional
Sistem dapat menambahkan data stok barang	Sistem dapat menampilkan <i>User Interface</i> yang menarik
Sistem dapat mengedit data stok barang	
Sistem dapat menghapus data stok barang	Tidak adanya <i>bug system</i>
Sistem dapat menampilkan data stok barang	

3. Implementasi Sistem



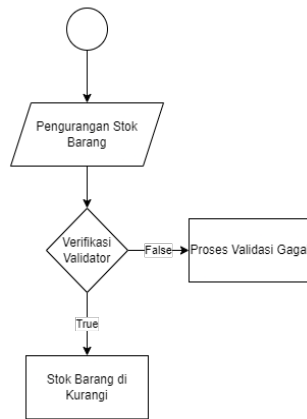
Gbr .2 Inisialisasi Token

Dalam proses Inisialisasi seperti pada gbr. 2, sistem membuat token untuk mewakili stok barang, lalu token tersebut didistribusikan kepada semua user yang ada dalam jaringan sistem.



Gbr .3 Penambahan Stok Barang

Dalam proses penambahan stok barang seperti pada gbr. 3, *user* akan melakukan input terlebih dahulu, lalu meminta verifikasi oleh validator, jika permintaan tersebut valid, maka stok barang akan ditambahkan ke blok baru dalam *blockchain* yang berisikan informasi penambahan stok barang. Namun, jika permintaan gagal maka stok tersebut tidak dapat ditambahkan.



Gbr .4 Pengurangan Stok Barang

Dalam proses pengurangan stok barang seperti pada gbr. 4, *user* melakukan permintaan kepada validator untuk mengurangi stok barang, jika permintaan valid, maka blok baru akan ditambahkan dalam *blockchain* yang berisikan informasi pengurangan stok barang. Namun, jika permintaan barang, blok tersebut tidak dapat ditambahkan ke dalam *blockchain*.

1) Diagram Blok Sistem

Perancangan sistem dalam penelitian ini dimulai dari pembuatan diagram blok sistem, dengan tujuan untuk dapat memberikan gambaran secara visual pada komponen-komponen utama yang akan ada dalam infrastruktur sistem aplikasi [8]. Diagram blok sistem nantinya akan berguna dalam pemetaan arsitektur keseluruhan sistem aplikasi, menentukan interaksi antara komponen-komponen yang ada, dan memastikan bahwa setiap elemennya memiliki peran yang jelas dalam fungsionalitas keseluruhan sistem. Dalam konteks penelitian ini, diagram blok sistem akan mencakup beberapa komponen-komponen utama seperti aplikasi *web frontend*, *smart contract*, Metamask sebagai *wallet* dari *user*, dan koneksi dengan jaringan lokal *blockchain* Ethereum. Diagram ini akan memudahkan peneliti untuk dapat memahami relasi antara komponen-komponen tersebut dan juga dapat merencanakan proses integrasi secara efisien.

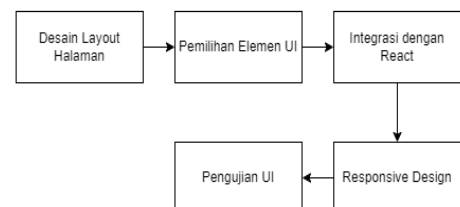
Gbr .5 Alur Implementasi Sistem

Setelah diagram blok sistem dibuat, seperti gbr. 5 langkah selanjutnya adalah merancang aplikasi berbasis web. Ini mencakup beberapa tahapan, termasuk perancangan *smart contract* untuk logika bisnis, konfigurasi Metamask untuk mengelola transaksi kripto, serta perancangan tampilan

pengguna yang responsif dan mudah digunakan. Perancangan ini akan memastikan bahwa aplikasi dapat berfungsi secara optimal dan sesuai dengan kebutuhan pengguna, sambil memanfaatkan teknologi yang dipilih, seperti React untuk pengembangan *frontend* dan Solidity untuk pengembangan *smart contract* dalam lingkungan *blockchain* Ethereum. Dengan menggunakan teknologi seperti Bootstrap untuk desain responsif, pengembang dapat menghasilkan antarmuka pengguna yang estetik dan intuitif.

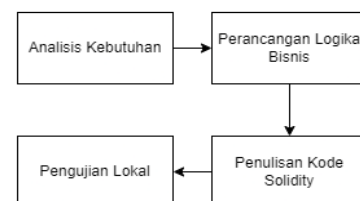
2) Perancangan Tampilan Pengguna

Dalam tahap perancangan tampilan pengguna, seperti pada gbr .6 fokusnya adalah merancang *user interface* (UI) yang responsif dan intuitif untuk aplikasi *point of sale* (POS) berbasis *website*. *Tech stack* yang digunakan meliputi react sebagai *framework frontend* untuk pengembangan UI yang dinamis, bootstrap untuk *responsive design* yang menyesuaikan dengan berbagai perangkat dan ukuran layar, serta javascript untuk logika interaksi UI. Proses perancangan tampilan pengguna mencakup langkah-langkah seperti desain *layout* halaman, pemilihan elemen UI yang sesuai, integrasi dengan komponen frontend react, dan pengujian tampilan guna memastikan kesesuaian dengan kebutuhan dari pengguna. Dengan menggunakan kombinasi react, bootstrap, dan javascript, tahap perancangan tampilan pengguna diharapkan dapat menghasilkan antarmuka yang menarik dan mudah digunakan bagi pengguna aplikasi POS berbasis website yang sedang dikembangkan.



Gbr .6 Alur Perancangan Tampilan Pengguna

3) Perancangan *Smart Contract* dalam Manajemen Stok Barang



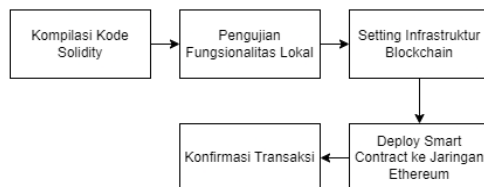
Gbr .7 Alur Perancangan Smart Contract

Dalam tahap perancangan smart contract, seperti pada gbr. 7 fokus utamanya adalah merancang kode kontrak cerdas (*smart contract*) yang akan

diimplementasikan dalam jaringan *blockchain* Ethereum [9]. *Tech stack* yang digunakan untuk ini meliputi Solidity sebagai bahasa pemrograman utama untuk *smart contract*, serta penggunaan Ganache sebagai simulator jaringan *blockchain* lokal untuk pengembangan dan pengujian. Solidity akan digunakan untuk menulis kode *smart contract* yang memuat logika bisnis yang tepat, termasuk fungsi-fungsi untuk manajemen stok barang seperti validasi stok barang, penambahan stok barang, dan manajemen harga dalam aplikasi *Point of Sale* (POS) berbasis *website*. Selain itu, penggunaan Ganache akan memfasilitasi pengembangan dan pengujian *smart contract* secara lokal sebelum di *deploy* ke jaringan Ethereum yang sebenarnya. Dengan menggunakan kombinasi Solidity dan Ganache, tahap perancangan *smart contract* diharapkan dapat menghasilkan kode yang efektif dan dapat diandalkan untuk memenuhi kebutuhan manajemen stok barang dalam konteks aplikasi POS berbasis *website* yang sedang diteliti.

4) Deploy Smart Contract

Dalam tahap *deploy smart contract*, seperti gbr. 8 fokusnya adalah mengunggah dan mengaktifkan kontrak cerdas (*smart contract*) yang telah dirancang ke dalam jaringan *blockchain* Ethereum yang sesuai. *Tech stack* yang digunakan meliputi Truffle sebagai *development environment* dan pengujian *smart contract*, serta ada Ganache yang digunakan sebagai simulator jaringan *blockchain* Ethereum untuk pengembangan lokal [10]. Proses *deploy smart contract* meliputi langkah-langkah seperti *code compilation* solidity menjadi *bytecode* yang dapat dijalankan di jaringan *blockchain*, pengujian fungsionalitas *smart contract* menggunakan Ganache untuk *local development*, dan akhirnya, pengunggahan dan aktivasi *smart contract* ke jaringan Ethereum yang sesungguhnya [11]. Dengan menggunakan perpaduan antara Truffle dan Ganache, tahap *deploy smart contract* diharapkan dapat dilaksanakan dengan lancar, memastikan bahwa *smart contract* dapat diakses dan digunakan oleh aplikasi *point of sale* (POS) berbasis *website* yang sedang dikembangkan.

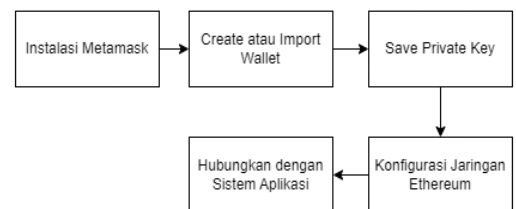


Gbr .8 Alur Deploy Smart Contract

5) Konfigurasi Metamask

Dalam tahapan konfigurasi metamask, seperti gbr. 9 fokusnya adalah mempersiapkan dan mengatur

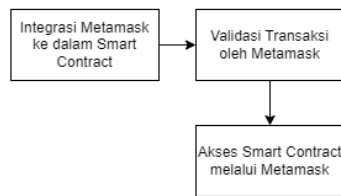
wallet kripto metamask agar dapat berinteraksi dengan aplikasi *point of sale* (POS) berbasis *website* yang akan dikembangkan. *Tech stack* yang digunakan yaitu react untuk *frontend development* aplikasi web, serta *library* web3 untuk memungkinkan komunikasi antara aplikasi web dan jaringan *blockchain* Ethereum. Proses konfigurasi dari metamask meliputi langkah-langkah seperti menginstal metamask *extension* pada *browser*, mengkonfigurasi dan mengelola *private key* serta *wallet address*, serta memperbarui pengaturan jaringan untuk berinteraksi dengan jaringan *blockchain* Ethereum yang tepat. Dengan memanfaatkan react sebagai *framework frontend* dan web3 sebagai *library* javascript untuk melakukan proses integrasi dengan *blockchain*, tahap konfigurasi metamask diharapkan dapat dilaksanakan dengan lancar, yang memungkinkan pengguna untuk melakukan transaksi kripto dengan aman dan efisien dalam aplikasi POS berbasis *website* yang sedang dikembangkan.



Gbr .9 Alur Konfigurasi Metamask

6) Integrasi Smart Contract dengan Tampilan Pengguna

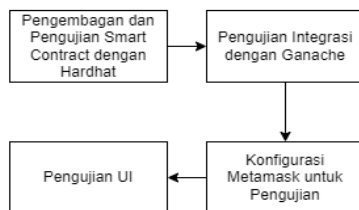
Dalam tahap integrasi *smart contract*, seperti gbr. 10 fokusnya adalah mengintegrasikan *smart contract* ke dalam aplikasi *point of sale* (POS) berbasis *website* untuk proses validasi transaksi yang melibatkan *smart contract*. *Tech stack* yang digunakan mencakup React sebagai *framework frontend* untuk pengembangan *user interface*, Web3.js sebagai *library* JavaScript untuk berinteraksi dengan jaringan Ethereum, dan Metamask sebagai *wallet* kripto yang digunakan user untuk melakukan konfirmasi transaksi. Proses integrasi *smart contract* meliputi langkah-langkah seperti menambahkan koneksi ke metamask ke dalam aplikasi web, mengintegrasikan tampilan Metamask untuk mengkonfirmasi transaksi, serta memastikan bahwa *smart contract* dapat diakses dan divalidasi melalui Metamask. Dengan menggunakan kombinasi react, web3, dan metamask, tahap integrasi metamask diharapkan dapat memperkuat keamanan dan validitas transaksi yang melibatkan *smart contract* dalam manajemen stok barang pada aplikasi POS berbasis *website* yang sedang dikembangkan.



Gbr .10 Alur Integrasi Smart Contract dengan Tampilan Pengguna

4. Pengujian Sistem

Dalam tahap pengujian sistem dengan *tech stack* truffle, ganache, dan metamask, seperti gbr. 11 fokusnya adalah pada testing menyeluruh pada fungsionalitas dan kinerja sistem aplikasi *point of sale* (POS) berbasis *website* yang menggunakan teknologi *blockchain*. Menggunakan truffle sebagai *development framework* ethereum, pengujian *smart contract* dapat dilakukan dengan cermat untuk memastikan bahwa logika bisnis yang tertanam dalam kontrak berfungsi dengan benar [12]. Ganache sebagai simulator jaringan *blockchain* Ethereum memungkinkan pengujian integrasi antara aplikasi *frontend* dan *smart contract* secara lokal, sehingga mempercepat siklus *development* dan *testing* [13]. Selanjutnya, dengan metamask sebagai *crypto wallet*, pengujian dapat mencakup interaksi antara user dan aplikasi, memvalidasi proses transaksi, dan memastikan antarmuka pengguna berfungsi dengan baik. Dengan menggunakan kombinasi ini, pengujian sistem dapat dilakukan secara menyeluruh untuk memastikan bahwa aplikasi POS berbasis *website* yang menggunakan teknologi *blockchain* dapat beroperasi dengan lancar, aman, dan sesuai dengan kebutuhan pengguna.



Gbr .11 Alur Pengujian Sistem

5. Kesimpulan dan Saran

Hasil dari seluruh tahap implementasi dan pengujian sistem dievaluasi untuk menyimpulkan keberhasilan proyek serta memberikan saran untuk perbaikan atau development lebih lanjut. Kesimpulan menyajikan rangkuman temuan utama dari penelitian ini, termasuk penilaian terhadap fungsionalitas, kinerja, dan keamanan aplikasi POS berbasis *website* yang menggunakan teknologi *blockchain* dengan menggunakan truffle, ganache, dan metamask. Selain itu, kesimpulan juga dapat mencakup analisis terhadap kesesuaian aplikasi dengan kebutuhan pengguna serta potensi dampaknya dalam meningkatkan efisiensi manajemen stok barang. Di samping itu, saran-saran diberikan untuk perbaikan atau pengembangan lebih lanjut, seperti peningkatan

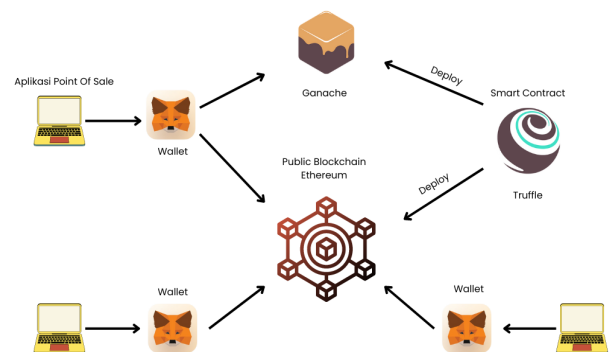
fitur, penambahan fungsionalitas baru, atau pemecahan masalah keamanan yang teridentifikasi selama pengujian. Kesimpulan dan saran ini memberikan gambaran menyeluruh tentang hasil proyek dan arahan untuk langkah-langkah selanjutnya dalam pengembangan aplikasi POS berbasis *website* yang menggunakan teknologi *blockchain*.

III. HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan terkait dengan hasil dari proses implementasi *blockchain* ke dalam manajemen stok barang pada sistem *point of sale*. Dimulai dari hasil implementasi *framework* React.js pada sisi *user interface* dan *Solidity* pada sisi *backend development* hingga pembahasan mengenai *unit testing* yang dilakukan untuk pengujian sistem *point of sale*.

A. Arsitektur Sistem Jaringan Blockchain pada Aplikasi

Pada bagian ini, menjelaskan mengenai tahapan-tahapan dalam mengimplementasikan *tech stack* yang telah dijelaskan pada bab sebelumnya ke dalam sistem *point of sale*.



Gbr .12 Infrastruktur Jaringan Blockchain pada Aplikasi

Seperti yang terlihat pada gbr. 12 mengenai infrastruktur jaringan *blockchain*, *smart contract* yang dibuat dengan kerangka kerja Truffle harus disebar ke jaringan *blockchain* sebelum dapat digunakan dalam aplikasi klien. Pada penelitian ini, Ganache digunakan sebagai jaringan *blockchain* lokal. Aplikasi klien yang dikembangkan dengan *framework* React.js akan diintegrasikan dengan *smart contract* yang telah di-deploy ke jaringan Ganache dengan menggunakan *library* Web3.js. Aplikasi klien juga membutuhkan Metamask untuk menghubungkan aplikasi klien dengan *smart contract* yang telah di-deploy, dimana setiap transaksi harus diverifikasi melalui Metamask. Dalam aplikasi dunia nyata, *smart contract* yang dibuat dengan kerangka kerja Truffle harus disebar ke jaringan *blockchain* publik yang dapat diakses oleh semua orang [14]. Metamask juga digunakan sebagai penghubung antara aplikasi klien dengan *smart contract* yang telah disebar di jaringan *blockchain* publik untuk memverifikasi setiap transaksi yang dilakukan.

mulai dari kode barang, nama barang, jumlah barang, dan harga satuan.

3) Modal Edit Stok Barang

Edit Stok Barang

Kode Barang:

Nama Barang:

Jumlah Barang:

Harga Satuan:

Gbr .18 Modal Edit Stok Barang

Modal edit stok barang pada gbr. 20 memperlihatkan beberapa parameter yang dapat diedit oleh user, mulai dari kode barang, nama barang, jumlah barang, dan harga satuan.

C. Pengujian Sistem

Pengujian sistem *point of sale* pada fitur *inventory*, dilakukan dengan menggunakan *unit testing* pada *code smart contract* dan *functional testing* pada *user interface*.

1. Unit Testing

Unit testing dilakukan untuk memastikan bahwa setiap *component* pada *smart contract* sesuai dengan spesifikasi yang diharapkan dan dengan tujuan utamanya yaitu untuk melakukan validasi setiap kode dalam *smart contract* beroperasi dengan benar tanpa adanya *bug system*. Beberapa indikator yang harus dipenuhi dalam *unit testing* pada *smart contract* fitur *inventory* antara lain yaitu fungsi *add item*, *get item*, *edit item*, *delete item*, *increase stock item*, dan *decrease stock item*.

```
it("should add an item", async () => {
  const itemId = await inventarisInstance.nextItemId();
  await inventarisInstance.addItem("TEST001", "Test Item", 50, 1000);
  const newItemId = await inventarisInstance.nextItemId();
  assert.equal(newItemId.toNumber(), itemId.toNumber() + 1, "Item was not added successfully");
});
```

Gbr .19 Source Code Add Item

Dalam pengujian *add item* pada kode diatas, pertama-tama mengambil ID item terakhir pada *inventory* menggunakan fungsi *nextItemId()*, lalu ditambahkan *new item* menggunakan parameter kode, name, stok, dan harga dengan fungsi *addItem()*. Berikutnya, ID item terakhir diambil kembali untuk memastikan fungsi *add item* beroperasi dengan benar.

```
it("should get item by index", async () => {
  const newItemIndex = await inventarisInstance.getItemCount() - 1;
  const item = await inventarisInstance.getItem(newItemIndex);
  assert.equal(item[1], "TEST001", "Item kode mismatch");
  assert.equal(item[2], "Test Item", "Item nama mismatch");
});
```

```
assert.equal(item[3].toNumber(), 50, "Item stok mismatch");
assert.equal(item[4].toNumber(), 1000, "Item harga mismatch");
});
```

Gbr .20 Source Code Get Item by Index

Pengujian sistem *smart contract* selanjutnya seperti pada kode diatas, yaitu *get item by index*. Yang dilakukan pertama kali yaitu menghitung *index* item terakhir menggunakan fungsi *getItemCount()* dan melakukan pengurangan pada *index*. Setelah itu, detail item diambil dengan menggunakan fungsi *getItem()*. Pengujian ini dilakukan dengan membandingkan kode, nama, stok, dan harga dengan nilai yang sesuai dengan item yang sebelumnya telah dilakukan penambahan. Namun, bila terdapat ketidakcocokan pada *smart contract*, maka pengujian ini dianggap gagal.

```
it("should edit item", async () => {
  const itemId = 1;
  await inventarisInstance.editItem(itemId, "TEST002", "Edited Item", 75, 2000);
  const editedItem = await inventarisInstance.getItem(itemId - 1);
  assert.equal(editedItem[1], "TEST002", "Item code should be updated");
  assert.equal(editedItem[2], "Edited Item", "Item name should be updated");
  assert.equal(editedItem[3], 75, "Item stock should be updated");
  assert.equal(editedItem[4], 2000, "Item price should be updated");
});
```

Gbr .21 Source Code Edit Item by ID

Pengujian selanjutnya yaitu edit item pada *smart contract inventory*. Pada pengujian ini, dipilih item menggunakan ID item, lalu digunakan fungsi *editItem()* untuk mengubah beberapa atribut item yang telah dipilih. Setelah itu, item yang telah di edit diambil kembali menggunakan fungsi *getItem()* dan disimpan pada *variable* *editedItem*. Seperti pada pengujian *get item by index*, apabila terdapat ketidaksesuaian pada nilai yang diharapkan, maka pengujian ini dianggap gagal.

```
it("should delete an item", async () => {
  const itemCountBefore = await inventarisInstance.getItemCount();
  const newItemIndex = itemCountBefore - 1;
  await inventarisInstance.deleteItem(newItemIndex);
  const itemCountAfter = await inventarisInstance.getItemCount();
  assert.equal(itemCountAfter.toNumber(), itemCountBefore.toNumber() - 1, "Item was not deleted successfully");
});
```

Gbr .22 Source Code Delete Item

Pengujian *smart contract* selanjutnya yaitu *delete item* pada fitur *inventory*. Pertama, sebelum melakukan penghapusan, di ambil jumlah total item dengan fungsi *getItemCount()* dan disimpan dalam *variable* *itemCountBefore*. Setelah itu, dilakukan penghitungan *index* item terakhir dengan cara mengurangi satu dari total jumlah item yang tersedia. Berikutnya, untuk melakukan

penghapusan item pada *index* tersebut diberikan fungsi `deleteItem()`. Setelah itu, total jumlah item yang tersedia setelah penghapusan diambil kembali dengan menggunakan `getItemCount()` dan disimpan pada *variable* `itemCountAfter`. Dan seperti pengujian sebelumnya, apabila asersi tidak terpenuhi, maka pengujian *delete item* dianggap gagal.

```
it("should decrease stock of an item", async () => {
  const newItemIndex = await
  inventarisInstance.getItemCount() - 1;
  const initialStock = await
  inventarisInstance.getStokBarang(newItemIndex);
  await inventarisInstance.kurangiStokBarang(newItemIndex,
  10);
  const updatedStock = await
  inventarisInstance.getStokBarang(newItemIndex);
  assert.equal(updatedStock.toNumber(),
  initialStock.toNumber() - 10, "Stock was not decreased successfully");
});
```

Gbr. 23 Source Code Decrease Stock Item

Dan, pengujian sistem pada *smart contract* yang terakhir, yaitu *decrease stock item*. Pertama, *index item* terakhir pada *inventory* diambil dengan fungsi `getItemCount()` dan dikurangi satu dikarenakan *index* dimulai dari nol. Setelah itu, menggunakan fungsi `getStokBarang()` untuk mengambil stok item dan disimpan pada *variable* `initialStock`. Berikutnya, untuk mengurangi stok item digunakan fungsi `kurangiStokBarang()`. Setelah stok dikurangi, digunakan fungsi `getStokBarang()` untuk mengambil stok item terbaru dan disimpan pada *variable* `updatedStock`. Dan, seperti pengujian sebelumnya, apabila asersi dalam pengujian *decrease stock item* tidak terpenuhi, maka pengujian ini dianggap gagal. Dan berikut merupakan hasil dari pengujian yang dilakukan dengan menggunakan beberapa indikator yang telah dibuat.

```
Contract: Inventaris
✓ should add an item (86ms)
✓ should get item by index
✓ should edit item (113ms)
✓ should delete an item (159ms)
✓ should increase stock of an item (64ms)
✓ should decrease stock of an item (81ms)
```

Gbr. 24 Hasil Unit Testing Inventory

2. Functional Testing

Functional Testing dilakukan untuk memastikan bahwa setiap *component* pada *user interface* sesuai dengan spesifikasi yang diharapkan dan dengan tujuan utamanya yaitu untuk melakukan validasi setiap *element*, *button*, dan *action* yang dilakukan oleh pengguna dalam *user interface* beroperasi dengan benar tanpa adanya *bug system*.

TABEL I
HASIL FUNCTIONAL TESTING FITUR CONNECT WALLET SISTEM POINT OF SALE DENGAN EKSTENSI METAMASK

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1	Kata sandi metamask tidak diisi	Kata sandi: (kosong)	Tombol buka pada ekstensi metamask tidak akan tidak dapat diklik	Tombol buka pada ekstensi metamask tidak tidak dapat diklik (berhasil)
2	Kata sandi metamask diisi dengan kata sandi yang salah	Kata sandi: (salah)	Sistem akan menolak koneksi website dengan ekstensi metamask dan menampilkan pesan kesalahan kata sandi	Sistem menolak koneksi website dengan ekstensi metamask dan menampilkan pesan kesalahan kata sandi (berhasil)
3	Kata sandi metamask diisi dengan benar	Kata sandi: (benar)	Ekstensi metamask akan berhasil terkoneksi dengan sistem	Ekstensi metamask akan berhasil terkoneksi dengan sistem (berhasil)

Pada tabel II, hasil *functional testing* fitur *connect wallet* sistem *point of sale* menggunakan ekstensi metamask dengan rincian pengujian terdiri dari tiga skenario, semuanya telah berhasil untuk dilakukan pengujian secara fungsional.

TABEL III
HASIL FUNCTIONAL TESTING FITUR TAMBAH BARANG PADA SISTEM POINT OF SALE

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1	Tidak mengisi kode barang, nama barang, jumlah barang, dan harga satuan	Kode barang: (kosong) Nama barang: (kosong) Jumlah barang: (kosong), Harga satuan: (kosong)	Sistem akan tidak menanggapi action submit yang dilakukan oleh pengguna dan sistem akan meminta untuk mengisi kolom selanjutnya yang belum terisi	Sistem tidak menanggapi action submit yang dilakukan oleh pengguna dan sistem akan meminta untuk mengisi kolom selanjutnya yang belum terisi (berhasil)
2	Mengisi kode barang, tetapi tidak mengisi nama barang, dan jumlah	Kode barang: (terisi) Nama barang: (kosong) Jumlah	Sistem akan tidak menanggapi action submit yang dilakukan oleh pengguna dan sistem	Sistem tidak menanggapi action submit yang dilakukan oleh pengguna

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
	jumlah barang, dan harga satuan	barang: (kosong), Harga satuan: (kosong)	akan meminta untuk mengisi kolom selanjutnya yang belum terisi	dan sistem akan meminta untuk mengisi kolom selanjutnya yang belum terisi (berhasil)
3	Mengisi kode barang, dan nama barang, tetapi tidak mengisi jumlah barang, dan harga satuan	Kode barang: (terisi) Nama barang: (terisi) Jumlah barang: (kosong), Harga satuan: (kosong)	Sistem akan tidak menanggapi action submit yang dilakukan oleh pengguna dan sistem akan meminta untuk mengisi kolom selanjutnya yang belum terisi	Sistem tidak menanggapi action submit yang dilakukan oleh pengguna dan sistem akan meminta untuk mengisi kolom selanjutnya yang belum terisi (berhasil)
4	Mengisi kode barang, nama barang, dan jumlah barang, tetapi tidak mengisi harga satuan	Kode barang: (terisi) Nama barang: (terisi) Jumlah barang: (terisi), Harga satuan: (kosong)	Sistem akan tidak menanggapi action submit yang dilakukan oleh pengguna dan sistem akan meminta untuk mengisi kolom selanjutnya yang belum terisi	Sistem akan tidak menanggapi action submit yang dilakukan oleh pengguna dan sistem akan meminta untuk mengisi kolom selanjutnya yang belum terisi
5	Mengisi kode barang, nama barang, jumlah barang, dan harga satuan	Kode barang: (terisi) Nama barang: (terisi) Jumlah barang: (terisi), Harga satuan: (terisi)	Sistem akan menanggapi action submit yang dilakukan dengan memunculkan modal ekstensi metamask untuk melakukan konfirmasi tambah barang Sistem akan menolak tambah barang yang dilakukan	Sistem akan menanggapi action submit yang dilakukan oleh pengguna modal ekstensi metamask untuk melakukan konfirmasi tambah barang Sistem akan menolak

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
				tambah barang yang dilakukan

Pada tabel III, hasil *functional testing* fitur tambah barang pada sistem *point of sale* dengan rincian pengujian terdiri dari tujuh skenario, semuanya telah berhasil untuk dilakukan pengujian secara fungsional.

TABEL III
HASIL FUNCTIONAL TESTING FITUR EDIT BARANG PADA SISTEM POINT OF SALE

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1	Mengedit kode barang	Kode barang: (diubah)	Sistem akan menanggapi action submit pengguna dan sistem memunculkan modal ekstensi metamask untuk konfirmasi edit barang	Sistem menanggapi action submit pengguna dan sistem memunculkan modal ekstensi metamask untuk konfirmasi edit barang (berhasil)
2	Mengedit nama barang	Nama barang: (diubah)	Sistem akan menanggapi action submit pengguna dan sistem memunculkan modal ekstensi metamask untuk konfirmasi edit barang	Sistem menanggapi action submit pengguna dan sistem memunculkan modal ekstensi metamask untuk konfirmasi edit barang (berhasil)
3	Mengedit jumlah barang	Jumlah barang: (diubah)	Sistem akan menanggapi action submit pengguna dan sistem memunculkan modal ekstensi metamask untuk konfirmasi edit barang	Sistem menanggapi action submit pengguna dan sistem memunculkan modal ekstensi metamask untuk konfirmasi edit barang (berhasil)
4	Mengedit harga satuan	Harga satuan: (diubah)	Sistem akan menanggapi action submit pengguna dan sistem memunculkan modal ekstensi metamask untuk konfirmasi edit barang	Sistem menanggapi action submit pengguna dan sistem memunculkan modal ekstensi metamask untuk konfirmasi edit barang (berhasil)

Pada tabel IV, hasil *functional testing* fitur edit barang pada sistem *point of sale* dengan rincian pengujian terdiri dari enam skenario, semuanya telah berhasil untuk dilakukan pengujian secara fungsional.

TABEL V
HASIL FUNCTIONAL TESTING FITUR HAPUS BARANG PADA SIS-TEM POINT OF SALE

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1	Memilih salah satu data stok barang dan menghapusnya dalam tabel stok barang	Klik tombol hapus pada tabel stok barang yang dipilih	Sistem akan menanggapi action hapus yang dilakukan oleh pengguna dan sistem memunculkan modal ekstensi metamask untuk konfirmasi hapus stok barang yang dipilih	Sistem menanggapi action hapus yang dilakukan oleh pengguna dan sistem memunculkan modal ekstensi metamask untuk konfirmasi hapus stok barang yang dipilih (berhasil)
2	Menolak konfirmasi hapus stok barang pada ekstensi metamask	Klik tombol tolak pada ekstensi metamask	Sistem akan menolak menghapus stok barang yang dilakukan	Sistem menolak menghapus stok barang yang dilakukan (berhasil)
3	Menyetujui konfirmasi hapus stok barang pada ekstensi metamask	Klik tombol konfirmasi pada ekstensi metamask	Sistem akan menghapus stok barang yang dilakukan	Sistem menghapus stok barang yang dilakukan (berhasil)

Pada tabel V, hasil *functional testing* fitur hapus barang pada sistem *point of sale* dengan rincian pengujian terdiri dari tiga skenario, semuanya telah berhasil untuk dilakukan pengujian secara fungsional.

IV. KESIMPULAN

Penelitian ini berhasil menerapkan teknologi *blockchain* pada sistem manajemen stok dengan menunjukkan keamanan dan transparansi data stok. Ketika *smart contract* dieksekusi, semua perubahan pada data stok secara otomatis direkam dalam blok terenkripsi, yang kemudian disimpan dalam jaringan *blockchain* yang terdesentralisasi. Hal ini memastikan bahwa setiap perubahan pada stok memiliki rekam jejak yang jelas dan tidak dapat diubah, sehingga mengurangi risiko manipulasi data. Pengujian fungsional menunjukkan bahwa aplikasi manajemen stok bekerja dengan baik dan menyediakan antarmuka pengguna yang responsif sambil memastikan bahwa setiap perubahan stok dicatat

secara akurat dan aman. Selain itu, catatan perubahan stok yang disimpan dalam perangkat lunak Ganache menunjukkan keefektifan *blockchain* dan memastikan transparansi penuh dari perubahan stok.

Penerapan metode konsensus *proof of stake* pada teknologi *blockchain* pada sistem manajemen stok ini terbukti efisien dan aman. Metode *proof of stake* memungkinkan untuk memverifikasi setiap perubahan stok antara node jaringan tanpa konsumsi energi yang tinggi. *Proof of stake* memberikan keamanan tambahan dengan memastikan bahwa hanya node yang memiliki kepentingan yang signifikan dalam jaringan yang dapat melakukan verifikasi, sehingga mengurangi risiko serangan dari pihak-pihak yang berniat jahat. Analisis ini menunjukkan bahwa teknologi *blockchain* dengan bukti konsensus kepemilikan tidak hanya meningkatkan efisiensi otentikasi perubahan saham, tetapi juga menjaga integritas dan keamanan data saham. Pengujian kontrak pintar memastikan bahwa setiap fungsi sistem manajemen stok bekerja dengan lancar seperti yang diinstruksikan, yang memperkuat keandalan seluruh sistem. Oleh karena itu, penelitian ini menyimpulkan bahwa penerapan teknologi *blockchain* dengan menggunakan metode *proof of stake* consensus cocok untuk meningkatkan keamanan, transparansi, dan efisiensi sistem manajemen stok.

REFERENSI

- [1] 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>
- [2] Putri, S. Z., Hariadi, M., & Rachmadi, R. F. (2023). Rekomendasi Barang pada E-Commerce Menggunakan Content Based Recommender System Berbasis Ethereum Blockchain. JURNAL TEKNIK ITS, 12(2), A145–A152. <https://doi.org/10.12962/j23373539.v12i2.115263>
- [3] Zeng, S., Ru, H., Huang, T., Jiang, L., Wang, S., & Feng, W. (2020). Survey of blockchain: principle, progress and application.
- [4] Rahardja, U., Aini, Q., Yusup, M., Edliyanti, A., & Raharja JI Jend, U. (2020). PENERAPAN TEKNOLOGI BLOCKCHAIN SEBAGAI MEDIA PENGAMANAN PROSES TRANSAKSI E-COMMERCE (Vol. 5, Nomor 1).
- [5] Nurul Hidayat, S. E., Astuti, S. D., SE, M., Munir Azhari, S. E., SS, M., Malaikosa, E. J., ... & Muhammad Fauzan, S. T. (2024). MANAJEMEN LOGISTIK DAN RANTAI PASOK TERINTEGRASI. CV Rey Media Grafika.
- [6] Syamil, A., Dania, R. D. M., Saori, S., Waty, E., Fahmi, M. A., Hartati, V., ... & Haryadi, R. M. (2023). Buku Ajar Manajemen Rantai Pasok. PT. Sonpedia Publishing Indonesia.
- [7] Sumampouw, E. G. J., & Sembiring, I. (2024). Analisis Verifikasi Proof of Stake (POS) NFT dengan Teknologi Smart Contract. Edukit: Jurnal Pendidikan Teknologi Informasi dan Komunikasi, 4(1), 15-28.
- [8] Arifin, N. Y., Kom, S., Kom, M., Tyas, S. S., Sulistiani, H., Kom, M., ... & Kom, M. (2022). Analisa Perancangan Sistem Informasi. Cendikia Mulia Mandiri.
- [9] Dewanto, A. (2021). Potensi Maksimal Dari Harga Value Pada Sebuah Transaksi Yang Menggunakan Teknologi Blockchain Di Jaringan Ethereum (Doctoral dissertation, Univeristas Komputer Indonesia).
- [10] Fardana, M. I. A. (2022). Sistem penilaian tempat wisata di Kabupaten Kediri berbasis web menggunakan Blockchain Ethereum (Doctoral dissertation, Univrsitas Islam Negeri Maulana Malik Ibrahim).
- [11] Darmawan, A. M. (2022). Sistem data sharing untuk rating dan preference user menggunakan ethereum blockchain pada game tempat wisata di Malang (Doctoral dissertation, Universitas Islam Negeri Maulana Malik Ibrahim).
- [12] Yafi, A., Arhandi, P. P., Firdaus, V. A. H., Ismail, A., & Batubulan, K. S. (2023). Sistem Keamanan E-Voting Menggunakan Arsitektur Publik

- Blockchain Ethereum. KLIK: Kajian Ilmiah Informatika dan Komputer, 4(3), 1313-1322.
- [13] Baiquni, M. F., & Dirgahayu, R. T. (2023). Aplikasi Terdesentralisasi Berbasis Blockchain dan Smart Contract untuk Pengelolaan Zakat. AUTOMATA, 4(2).
- [14] Fernando, A. (2022). IMPLEMENTASI SMART CONTRACT UNTUK E-VOTING PADA JARINGAN BLOCKCHAIN (Doctoral dissertation, KODEUNIVERSITAS041060# UniversitasBuddhiDharma).