

## **BAB II**

### **LANDASAN TEORI**

#### ***Marketplace* NFT**

*Marketplace* merupakan *platform* yang menyediakan sebuah tempat dan juga fasilitas untuk mempertemukan para penjual dan pembeli untuk menawarkan produk yang dihasilkan atau dikelola. [3]

NFT atau *non-Fungible Token* merupakan identifikasi *asset digital* unik yang tidak dapat disalin ataupun dibagi, karena token yang sudah diidentifikasi sudah tercatat dalam sistem *blockchain* sebagai pengesahan dari pemilik aslinya. Maka dari itu *marketplace* NFT merupakan *platform* yang menyediakan tempat serta fasilitas untuk *creator* atau *owner* sebagai penjual dan pengguna sebagai pembeli, yang didalamnya *creator* menciptakan karya berupa NFT lalu menjualnya, dan para konsumen membelinya melalui transaksi dompet mata uang kripto. [8]

#### **Sistem *Blockchain* dan jenis service**

*Blockchain* adalah sebuah mekanisme basis data secara lanjutan dengan menyimpan data ke dalam blok yang terhubung dengan rantai yang berfungsi untuk memberikan informasi secara transparan dalam jaringan.

Solusi teknologi untuk *marketplace* MetanesiaNFT adalah menggunakan teknologi *Blockchain*. Alasan menggunakan *Blockchain* karena NFT hanya bisa digunakan atau terhubung ke sistem *blockchain* yang sebagai sistem pendukung perancangan dan pembangunan *asset digital* atau token NFT.

Teknologi *Blockchain* memiliki sifat transparan, terautentikasi, terdistribusi dan permanen didalam infrastrukturnya, maka dapat memberikan rasa aman dari pemalsuan data informasi, memberikan data informasi yang valid terhadap pembaca, dan masih banyak lagi manfaat yang bisa didapatkan. [9]

a. *Private Blockchain*

*Private Blockchain* merupakan layanan yang memudahkan perusahaan Anda untuk membangun chain infrastruktur secara tertutup di dalam satu perusahaan, atau dalam satu grup perusahaan. [7]

b. *Public Blockchain*

*Public Blockchain* merupakan layanan yang memudahkan perusahaan Anda untuk membangun chain infrastruktur secara publik dengan menggunakan jaringan *Blockchain* publik yang terpercaya. [7]

### **Tools Teknologi yang digunakan**

a. NestJS

NestJS adalah kerangka kerja untuk membangun aplikasi sisi server Node.js yang efisien dan dapat diskalakan. Ini menggunakan JavaScript progresif, dibangun dengan dan sepenuhnya mendukung TypeScript (namun masih memungkinkan pengembang untuk membuat kode dalam JavaScript murni) dan menggabungkan elemen OOP (Pemrograman Berorientasi Objek), FP (Pemrograman Fungsional), dan FRP (Pemrograman Reaktif Fungsional). Di bawah tenda, Nest memanfaatkan kerangka kerja Server HTTP yang kuat seperti Express (default) dan secara opsional dapat dikonfigurasi untuk menggunakan Fastify juga! Nest menyediakan level abstraksi di atas framework Node.js umum ini (Express/Fastify), tetapi juga memaparkan API mereka langsung ke *developer*. Ini memberi pengembang kebebasan untuk menggunakan banyak sekali modul pihak ketiga yang tersedia untuk *platform* yang mendasarinya. [10]

b. VueJS

VueJS adalah framework JavaScript untuk membangun antarmuka pengguna. dibangun dengan HTML standar, CSS, dan JavaScript dan menyediakan model pemrograman deklaratif dan berbasis komponen. framework dan ekosistem ini mencakup sebagian besar fitur umum yang

dibutuhkan dalam pengembangan frontend. Namun web sangat beragam - hal yang kami buat di web dapat sangat bervariasi dalam bentuk dan skala. Dengan pemikiran tersebut, Vue dirancang agar fleksibel dan dapat diadopsi secara bertahap. Vue dapat digunakan dengan berbagai cara:

- Meningkatkan HTML statis tanpa langkah build
- Menyematkan sebagai Komponen Web di halaman mana pun
- Aplikasi Satu Halaman (SPA)
- Fullstack / Rendering Sisi Server (SSR)
- Jamstack / Pembuatan Situs Statis (SSG)
- Menargetkan desktop, seluler, WebGL, dan bahkan terminal [11]

#### c. Database NoSQL MongoDB

Database NoSQL dibuat dengan tujuan khusus untuk model data spesifik dan memiliki skema fleksibel untuk membuat aplikasi modern. Database NoSQL dikenal secara luas karena kemudahan pengembangan, fungsionalitas, dan kinerja dalam berbagai skala. Halaman ini menyertakan sumber daya untuk membantu Anda memahami lebih baik database NoSQL dan mulai menggunakannya. Database NoSQL sangat cocok untuk digunakan dengan berbagai aplikasi modern seperti aplikasi seluler, web, dan gaming yang memerlukan database yang fleksibel, dapat diskalakan, berkinerja tinggi, dan memiliki fungsionalitas tinggi untuk memberikan pengalaman pengguna yang baik. [12]

- **Fleksibilitas:** Database NoSQL umumnya menyediakan skema fleksibel yang memungkinkan pengembangan yang lebih cepat dan lebih berulang. Model data fleksibel membuat database NoSQL ideal untuk data yang semi terstruktur dan tidak terstruktur.
- **Skalabilitas:** Database NoSQL umumnya didesain untuk meningkatkan skala dengan menggunakan kluster perangkat keras yang terdistribusi alih-alih meningkatkan skala dengan menambah server yang mahal dan robust. Beberapa penyedia layanan cloud menangani aktivitas di balik operasi ini sebagai layanan yang dikelola sepenuhnya.

- Kinerja tinggi: Database NoSQL dioptimalkan untuk model data spesifik dan pola akses yang memberikan kinerja yang lebih tinggi dibandingkan jika Anda mencoba mendapatkan fungsionalitas yang mirip dengan database relasional.
- Fungsionalitas tinggi: Database NoSQL menyediakan API dan jenis data fungsional yang dibuat secara khusus untuk setiap model data yang sesuai. [12]

MongoDB adalah sistem basis data berorientasi dokumen lintas *platform*. Diklasifikasikan sebagai basis data "NoSQL", MongoDB menghindari struktur basis data relasional tabel berbasis tradisional yang mendukung JSON seperti dokumen dengan skema dinamis (MongoDB menyebutnya sebagai format BSON), membuat integrasi data dalam beberapa jenis aplikasi lebih mudah dan lebih cepat. Dirilis di bawah Server Side Public License, MongoDB adalah perangkat lunak bebas dan sumber terbuka. [13]

### **Metode Pengembangan Perangkat Lunak**

Dalam mengimplementasikan solusi tersebut divisi/departemen perusahaan memajemen proyek tersebut menggunakan metode pengembangan perangkat lunak menggunakan metode *scrum* dengan melakukan pendekatan mindset agile.

Di dalam metode *scrum* terdapat *Scrum team* yang terdiri dari *product owner*, *scrum master* dan *team development*. Didalam *scrum* tim bersifat swakelola dan lintas fungsi yang artinya swakelola ialah mengerjakan pekerjaan dengan cara terbaik tanpa diperintah orang lain diluar tim dan lintas fungsi ialah seseorang yang memiliki semua keahlian dan bisa mengerjakan semua tanpa bergantung dengan orang lain. [14]

#### **a. Definisi Metode *Scrum***

Dilansir dari Buku panduan penggunaan *Scrum*, *Scrum* merupakan Sebuah kerangka kerja dimana orang-orang dapat mengatasi masalah kompleks

adaptif, dimana pada saat bersamaan mereka juga menghantarkan produk dengan nilai setinggi mungkin secara produktif dan kreatif. Kerangka Kerja atau disebut framework tersebut melakukan pendekatan mindset agile. Pada metode pengembangan menggunakan kerangka kerja *scrum* mempunyai beberapa control empiris yang berfungsi untuk mengoptimalkan kemampuan prediksi dan mengendalikan risiko, diantaranya ada :

a) Transparansi

Aspek signifikan dari sebuah proses harus dapat dilihat oleh orang-orang yang bertanggung jawab terhadap dampaknya. Transparansi membutuhkan aspek-aspek tersebut ditentukan oleh standar baku sehingga para pengamat memiliki pemahaman yang sama terhadap apa yang sedang ditinjau.

b) Inspeksi

Pengguna *Scrum* harus sering menginspeksi artefak *Scrum* dan perkembangan menuju *Sprint* Goal agar mereka dapat mendeteksi adanya variansi hasil yang tidak diharapkan.

c) Adaptasi

Jika pemeriksa menemukan bahwa ada satu hal atau lebih dari proses yang menyimpang di luar ambang batas yang bisa diterima yang dapat menyebabkan produk tidak bisa diterima, maka proses atau materi yang sedang diproses harus diubah. Pengubahan harus dilakukan secepatnya untuk meminimalkan penyimpangan yang semakin jauh. [14]

b. *Scrum Team*

Di dalam *Scrum* terdapat *scrum team* dan *event sprint*. *Scrum team* diantaranya

a) *Product Owner*

*Product Owner* adalah orang yang bertanggung jawab untuk memaksimalkan nilai bisnis dari produk yang dihasilkan oleh

*Development Team*. Cara melakukannya sangat bervariasi antar organisasi, *Scrum Team* dan individu

b) *Scrum Master*

*Scrum Master* bertanggung jawab untuk mengenalkan dan menyokong penggunaan *Scrum* sebagaimana dijelaskan di dalam Panduan *Scrum* ini. *Scrum Master* melakukan ini dengan membantu orang-orang agar dapat memahami teori, praktik praktik, aturan-aturan dan tata nilai *Scrum*.

c) *Development Team*

*Development Team* terdiri dari para ahli profesi yang bekerja untuk menghantarkan Increment “Selesai” yang berpotensi untuk dirilis di setiap akhir *Sprint*. Increment “Selesai” wajib tersedia pada saat *Sprint Review*. Hanya anggota dari *Development Team* yang membuat Increment ini. [14]

c. Acara *Sprint*

*Sprint* merupakan detak jantung dalam *scrum* yang didalam penggunaan metode pengembangan *scrum* terdapat ketentuan Batasan waktu satu bulan atau kurang yang diatur sesuai ketentuan. *Sprint* memiliki empat acara formal untuk melakukan inspeksi dan adaptasi sebagai berikut:

- *Sprint Planning*, merupakan *event sprint* yang didalamnya bekerja untuk membuat perencanaan tugas yang akan dikerjakan pada *sprint* yang akan dijalankan. Perencanaan ini dilakukan secara kolaboratif Bersama seluruh anggota *scrum team*
- *Daily scrum*, merupakan *event sprint* yang dilakukan setiap harinya selama *sprint* berlangsung dengan timebox yang disediakan adalah 15 menit. Tujuan event ini untuk mengoptimalkan kerjasama tim dengan melakukan inspeksi pada pekerjaan yang dilakukan.

- *Sprint Review*, merupakan *event sprint* yang dilakukan oleh *scrum team* di penghujung *sprint*, didalam event ini akan dilakukan inspeksi increment dan peninjauan *product backlog* oleh pemegang kepentingan.
- *Sprint Retrospective*, merupakan *event sprint* yang dilakukan oleh *scrum team* untuk menginspeksi dirinya sendiri dan memperbaiki untuk meningkatkan kualitas kinerja di *sprint* berikutnya [14]

d. Artefak *Scrum*

Artefak *Scrum* merepresentasikan pekerjaan atau nilai bisnis guna terciptanya transparansi dan kesempatan untuk menginspeksi dan mengadaptasi. Artefak dirancang untuk memaksimalkan transparansi informasi utama agar setiap orang memiliki pemahaman yang sama mengenai artefak tersebut, artefak *scrum* diantaranya

- a. *Product backlog*, adalah daftar terurut semua hal yang telah diketahui hingga saat ini harus ada di dalam produk. *Product backlog* adalah satu-satunya sumber kebutuhan untuk semua perubahan yang perlu diberlakukan terhadap produk. *Product Owner* bertanggung jawab terhadap *Product backlog*, termasuk isi, ketersediaan dan urutannya.
- b. *Sprint Backlog* adalah daftar *Product backlog* item yang terpilih untuk *Sprint* ditambah perencanaan untuk menghantarkan Increment dan mencapai *Sprint Goal*. *Sprint Backlog* adalah prakiraan dari *Development Team* mengenai fungsionalitas yang akan masuk ke dalam Increment berikutnya dan pekerjaan yang perlu dikerjakan untuk menghantarkan fungsionalitasnya menjadi Increment yang “Selesai”.
- c. Increment adalah manifestasi dari *Product backlog* item yang diselesaikan dalam *Sprint* dan total nilai bisnis Increment dari seluruh *Sprint* yang lalu. [14]