BAB II

LANDASAN TEORI

VOTING

Voting adalah cara pengambilan keputusan dengan cara mengambil suara terbanyak dari jumlah anggota seluruhnya. Voting akan mengambil paling tidak 50% + 1 dari jumlah keseluruhan anggota. Voting biasanya dilakukan setelah melakukan sebuah musyawarah dimana musyawarah yang dilakukan tidak menemukan solusi/ belum mendapatkan sebuah keputusan bersama, maka dilakukanlah sistem voting Mulai dari pemilihan RT,RW,Kepala Desa sampai tingkat Presiden semuanya dilakukan dengan sistem voting yaitu pencoblosan surat suara (pemilu).Bahkan dalam ruang lingkup kecil seperti sebuah kelas di sekolah / kampus pun terkadang memakai sistem voting. Sistem voting dianggap suatu cara yang cukup efektif karena seluruh anggota dapat menyaksikan sendiri bagaimana pemilihan terlaksana seingga meminimalisir sebuah kecemburuan antar pihak dan saling bisa lebih menerima sebuah hasil keputusan[18].

TEKNOLOGI WEB

Teknologi Web adalah teknologi yang berhubungan dengan antarmuka untuk menghubungkan server dan client. Web aplikasi membutuhkan tool utama sebagai jembatan untuk mulai berselancar (browsing) pada jaringan internet yaitu Browser. Banyak beberapa jenis web browser diantaranya yang paling sering digunakan adalah Chrome, Firefox, Opera, dan IE. Dasar – dasar pada teknologi web diantaranya:

1. HTML

   Hypertext Markup Language adalah bahasa yang digunakan untuk mendeskripsikan dan menentukan konten dari halaman web.

2. CSS

   Cascading Style Sheet digunakan untuk menentukan tampilan/gambar pada konten web

3. HTTP

   Hypertext Transfer Protocol digunakan untuk mengantar HTML dan komponen konten lainnya

4. Web API

   Web Application Programming Interface digunakan untuk melaksanakan tugas - tugas beragam seperti memanipulasi DOM, dan memainkan audio maupun video.

Dalam artian abstrak, Web adalah kumpulan dokumen yang sangat banyak, beberapa di antaranya dihubungkan oleh tautan link. Dokumen-dokumen ini diakses oleh browser Web. Web berisikan semua kumpulan perangkat lunak dan protokol yang telah diinstal pada sebagian besar[19].

E-VOTING

E-voting merupakan sebuah sistem yang memanfaatkan perangkat elektronik dan mengolah informasi digital untuk membuat surat suara, memberikan suara, menghitung perolehan suara, menayangkan perolehan suara dan memelihara serta menghasilkan jejak audit. Dibandingkan dengan pemungutan suara konvensional, e-voting menawarkan beberapa keuntungan. Sebuah proses e-voting harus dirancang sedemikian rupa untuk menjamin terpenuhinya asas-asas pemilu yaitu langsung, umum, bebas, rahasia (luber) dan .jujur dan adil (jurdil) dalam pemilihan langsung. Dalam konteks demokrasi, sistem pemungutan suara elektronik (e-voting) juga harus menghormati dan menjamin atribut dan sifat dari pemilihan langsung tersebut seperti transparansi, kepastian, keamanan akuntabilitas, dan akurasi. Selain kesiapan teknologi, tentunya harus didukung dengan kesiapan masyarakat dalam melaksanakan sistem e-voting ini kedepannya. Ketidaksiapan yang juga ditambah dengan kurangnya sosialisasi pemerintah terhadap e-voting juga dapat menjadi faktor pemicu kegagalan dalam penerapan sistem ini, Sistem e-voting yang diharapkan mampu mengakomodasi seluruh asas-asas pemilu secara efektif dan efisien[20].

BLOCKCHAIN

Pada tahap ini akan diterangkan mengenai definisi Blockchain, sejarah Blockchain, dan cara kerja Blockchain.

Definisi Blockchain

Blockchain merupakan struktur data yang memungkinkan membuat buku besar digital (hyperledger) dan membagikannya diantara jaringan dengan pihak independen[21]. Setiap transaksi dalam buku besar publik diverifikasi oleh konsensus mayoritas peserta dalam sistem. Setelah dimasukan, informasi tidak akan pernah bisa diubah atau dihapus. Blockchain berisi catatan tertentu dan dapat diverifikasi dari setiap transaksi yang pernah dilakukan. tetapi teknologi Blockchain yang mendasarinya telah bekerja dengan sempurna dan menemukan berbagai aplikasi baik di dunia finansial maupun non-finansial.

Sejarah Blockchain

Pada tahun 2008, seorang individu (grup) yang menulis dengan nama Satoshi Nakamoto menerbitkan sebuah makalah berjudul "Bitcoin: A Peer-To-Peer Electronic Cash System". Makalah ini menjelaskan versi peer-to-peer dari uang tunai elektronik yang akan memungkinkan pembayaran online untuk dikirim langsung dari satu pihak ke pihak lain tanpa melalui lembaga keuangan. Bitcoin adalah realisasi pertama dari konsep ini. Sekarang "cryptocurrency" adalah label yang digunakan untuk menggambarkan semua jaringan dan media pertukaran yang menggunakan kriptografi untuk mengamankan transaksi seperti terhadap sistem-sistem di mana transaksi disalurkan melalui entitas terpercaya yang terpusat. Penulis makalah ingin tetap menjadi anonim dan karenanya tidak ada yang tahu Satoshi Nakamoto sampai hari ini. Beberapa bulan kemudian, sebuah program open source yang mengimplementasikan protokol baru dirilis. Siapa pun dapat menginstal program sumber terbuka ini dan menjadi bagian dari jaringan peer-to-peer bitcoin. Ini telah tumbuh dalam popularitas sejak itu.

Cara Kerja Blockchain

Bitcoin menggunakan bukti kriptografi alih-alih mekanisme trust-in-the-third-party untuk dua pihak yang bersedia melakukan transaksi online melalui Internet. Setiap transaksi dilindungi melalui tanda tangan digital, dikirim ke "Public key" penerima, dan ditandatangani secara digital menggunakan "private key" pengirim. Untuk menghabiskan uang, pemilik cryptocurrency perlu membuktikan kepemilikannya atas "private key". Entitas yang menerima mata uang digital kemudian memverifikasi tanda tangan digital, yang menyatakan kepemilikan "private key" yang sesuai, dengan menggunakan "private key" pengirim pada transaksi masing-masing. Kriptografi di belakang protokol didasarkan pada modulo enkripsi simetris matematika dimana 'kunci' untuk mengenkripsi pesan atau transaksi berbeda dari 'kunci' untuk mendeskripsi[22]. Sistem desentralisasi Blockchain berjalan dengan jaringan peer-to-peer dimana semua pengguna dengan node block berperan penting dalam penyimpanan dan distribusi data pada satu jaringan tanpa memanfaatkan distribusi server secara sentral. Selain itu, tiap block melakukan verifikasi dan dikelola menggunakan otomatisasi dan protokol tata kelola Bersama[23].

Setiap transaksi dikirim ke setiap node yang ada dalam jaringan Blockchain dan kemudian dicatat dalam buku besar publik (Hyperledger) setelah diverifikasi. Setiap transaksi perlu diverifikasi untuk validasi sebelum dicatat dalam buku besar. Node verifikasi perlu memastikan dua hal sebelum merekam transaksi apa pun :

1. Pengirim memiliki cryptocurrency melalui verifikasi tanda tangan digital pada transaksi.
2. Pengirim memiliki jumlah cryptocurrency yang cukup dalam akunnya untuk melakukan pengiriman, melalui pemeriksaan transaksi terhadap akun pengirim atau “Public key”, yang terdaftar di hyperledger (buku besar). Langkah tersebut memastikan bahwa ada saldo yang cukup di akunnya sebelum menyelesaikan transaksi.

ETHEREUM

Ethereum adalah proyek yang mencoba untuk membangun teknologi umum; teknologi dimana semua konsep mesin berbasis transaksi dapat dibangun. Selain itu Ethereum juga bertujuan untuk memberikan pengembang akhir (end developer) sistem end-to-end yang terintegrasi secara ketat untuk membangun perangkat lunak.

Block

Block dalam Ethereum adalah kumpulan potongan informasi yang relevan (dikenal sebagai header block), bersama dengan informasi yang berkaitan dengan transaksi yang terdiri, dan satu set header block U lainnya yang diketahui memiliki induk sama dengan induk block sekarang. Header block mengandung beberapa informasi yaitu:

1. ParentHash
   Hash Keccak 256-bit dari header block induk, secara keseluruhan.
2. OmmersHash
   Hash Keccak 256-bit dari bagian daftar ommers dari block ini.
3. Beneficiary
   Alamat 160-bit di mana semua biaya yang dikumpulkan dari keberhasilan penambangan block ini akan ditransfer
4. stateRoot
   Hash Keccak 256-bit dari node root dari state trie, setelah semua transaksi dieksekusi dan finalisasi diterapkan.
5. transactionsRoot
   Hash Keccak 256-bit dari simpul akar dari struktur trie diisi dengan setiap transaksi di bagian daftar transaksi block.
6. receiptsRoot
   Hash Keccak 256-bit dari node root dari struktur trie diisi dengan penerimaan dari setiap transaksi di bagian daftar transaksi block.
7. logsBloom

Filter Bloom terdiri dari informasi yang dapat diindeks (alamat log dan topik log) yang terkandung dalam setiap entri log dari penerimaan setiap transaksi dalam daftar transaksi.

8. Difficulty

   Nilai skalar yang sesuai dengan tingkat kesulitan block ini. Ini dapat dihitung dari tingkat kesulitan block sebelumnya dan cap waktu.

9. Number

   Nilai skalar sama dengan jumlah block leluhur. Block genesis memiliki angka nol.

10. gasLimit

    Nilai skalar sama dengan batas pengeluaran gas saat ini per block.

11. gasUsed

    Nilai skalar sama dengan total gas yang digunakan dalam transaksi di block ini.

12. Timestamp

    Nilai skalar sama dengan output pada waktu Unix () pada awal block ini.

13. extraData

    Byte array acak yang berisi data yang relevan dengan block ini. Harus 32 byte atau kurang.

14. mixHash

    Hash 256-bit yang terbukti dikombinasikan dengan notice bahwa jumlah komputasi yang cukup telah dilakukan pada block ini.

15. Nonce

    Hash 64-bit yang terbukti dikombinasikan dengan campuran-hash bahwa jumlah komputasi yang cukup telah dilakukan pada block ini.

World State

World State (state) adalah pemetaan antara alamat (identifier 160 bit) dan state akun (struktur data yang direalisasikan). Meskipun tidak disimpan di Blockchain, diasumsikan bahwa implementasi akan mempertahankan pemetaan ini. State akun terdiri dari 4 bidang berikut:

a. Nonce

   Nilai skalar sama dengan jumlah transaksi yang dikirim dari alamat ini atau, dalam kasus akun dengan kode terkait, jumlah contract creation yang dibuat oleh akun.

b. Balance

   Nilai skalar sama dengan jumlah Wei yang dimiliki oleh alamat ini.

c. storageRoot

   Hash 256-bit dari simpul akar pohon Merkle Patricia yang mengkodekan konten penyimpanan akun (pemetaan antara nilai-nilai integer 256-bit), dikodekan ke dalam trie sebagai pemetaan dari hash Keccak 256-bit dari 256 kunci integer bit ke nilai integer 256-bit yang dikodekan RLP.

d. codeHash

   Hash dari kode EVM akun ini adalah kode yang dieksekusi jika alamat ini menerima panggilan pesan; itu tidak berubah dan dengan demikian, tidak seperti semua bidang lainnya, tidak dapat diubah setelah konstruksi. Semua fragmen kode seperti itu terkandung dalam database negara di bawah hash yang sesuai untuk pengambilan nanti.

Transaksi

Suatu transaksi adalah instruksi tunggal yang ditandatangani secara kriptografis yang dibangun oleh seorang aktor secara eksternal dengan ruang lingkup Ethereum. Meskipun diasumsikan bahwa faktor eksternal utama akan bersifat manusia, alat perangkat lunak akan digunakan dalam konstruksi dan penyebarannya. Ada dua jenis transaksi: transaksi yang menghasilkan panggilan pesan dan transaksi yang menghasilkan akun baru dengan kode terkait (dikenal secara informal sebagai ‘contract creation’).

SMART CONTRACT

Pada tahap ini akan diterangkan mengenai definisi smart contract, jenis – jenis smart contract, syarat keabsahan kontrak, dan perlindungan hukum penggunaan smart contract dalam sestem keamana e-voting.

Definisi Smart Contract

Smart contract merupakan suatu bentuk perjanjian elektronik yang berkaitan erat dengan teknologi Blockchain. Kim mendefinisikan Blockchain sebagai "*a distributed database that maintains a continuously-growing list of data records secured from tampering and revision. It consists of blocks, holding batches of individual transactions. Each block contains a timestamp and a link to a previous block*"[24].

Jenis – Jenis Smart Contract

Smart contract terbagi menjadi 5 (lima) macam bentuk dengan fungsi dan penerapan yang berbeda[28]. Kelima macam bentuk tersebut adalah basic token contract, crowd sale contract, mintable contract, refundable contract, dan terminable contract. Dari kelima macam bentuk smart contract, empat bentuk pertama merupakan smart contract yang biasa digunakan dalam jual beli cryptocurrency. Sedangkan Terminable Contract merupakan bentuk smart contract yang dapat digunakan untuk sistem Blockchain dalam jual beli barang secara online dan eksekusi program Blockchain dalam jasa keuangan dan perbankan.

Smart Contract dan Solidity

Di dasar platform Ethereum berdiri dua jenis entitas, yang disebut akun; akun dapat dimiliki secara eksternal, yang biasanya dikendalikan oleh aktor manusia melalui kunci kriptografi pribadi, atau dapat berupa akun kontrak. Akun kontrak dikendalikan oleh kode yang akan dijalankan di mesin virtual, yang hanya dapat diaktifkan oleh akun yang dimiliki secara eksternal .Akun kontrak menerapkan kontrak pintar, sistem yang biasanya berisi token nilai yang akan dibuka kuncinya hanya jika kondisi tertentu terpenuhi[31].

```
//inisialisasi contract set user yang mendeploy sebagai admin
  function Evote_contract() public {
     manager = msg.sender;
  }

  // pembatasan pada fungsi yang hanya bisa digunakan oleh admin
  modifier restricted() {
     require(msg.sender == manager);
     _;
  }
```

Solidity adalah bahasa tingkat tinggi yang berorientasi kontrak, diketik secara statis, yang digunakan untuk mengimplementasikan kontrak pintar pada mesin virtual Ethereum .Kontrak dalam Solidity didefinisikan, mirip dengan kelas dalam pemrograman berorientasi objek tradisional, sebagai kumpulan fungsi dan data. Pemanggilan fungsi, serta sejarah nilai data yang disimpan, disimpan di Blockchain yang mendasarinya, membuat eksekusi kontrak pintar sepenuhnya dapat dibaca.

```
//fungsi untuk vote kandidat
   function voteCandidate(uint256 index) public {
      bool valid = true;
      Candidate storage candidate = candidates[index];
      if (candidate.voters[msg.sender] || candidate.complete == true) {
         valid = false;
      } else {
         for (uint256 i = 0; i < getCandidatesCount(); i++) {
            if (candidates[i].complete == false) {
               if (candidates[i].voters[msg.sender]) {
                  valid = false;
               }
            }
         }
      }
      require(valid);
      require(validate_voters(msg.sender));

      candidate.voters[msg.sender] = true;
      candidate.voteCount++;
   }
```

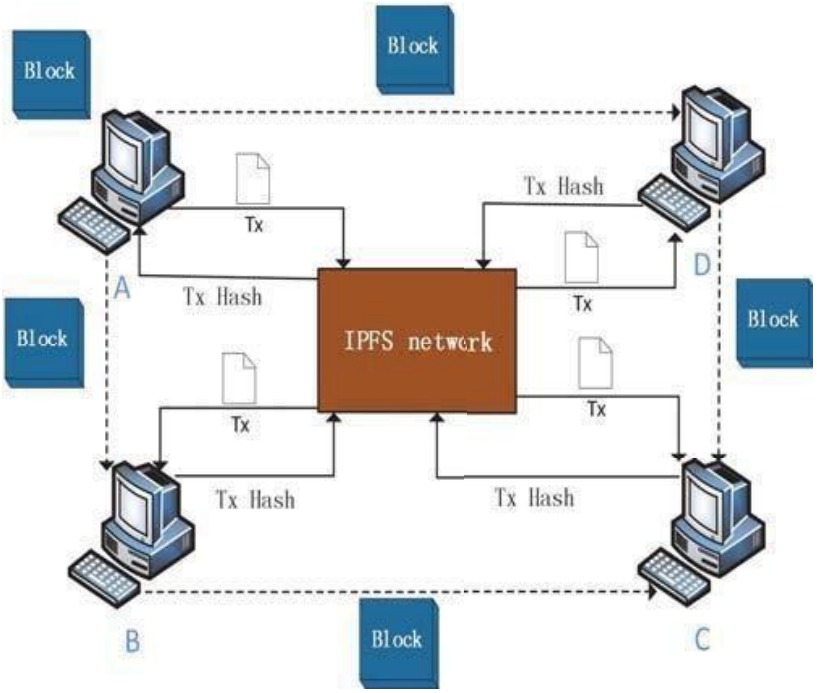
```
function validate_voters(address _sender) public view returns (bool) {
    bool validate = false;

    Audience[] memory newAudience = audiences;
    for (uint256 i = 0; i < newAudience.length; i++) {
        if (
            newAudience[i].pesertaAddress == _sender &&
            newAudience[i].active
        ) {
            validate = true;
            i = newAudience.length;
        }
    }
    return validate;
}
```

INTERPLANETARY FILE SYSTEM

Interplanetry File System (IPFS) adalah distribusi file sistem berbasis protokol peer-to-peer untuk menyimpan file. Tiap node pada IPFS menyimpan objek IPFS dalam penyimpanan lokal. Node terhubung satu sama lain dan mentransfer objek. Objek – objek ini mewakili file dan struktur data lainnya. IPFS merupakan platform untuk menulis, menggunakan aplikasi, dan sistem baru untuk mendistribusikan dan membuat versi data yang besar.

Interplanetary File mendistribusikan protokol penyimpanan. Konsepnya mempunyai mekanisme deduplikasi tanpa batasan dari server pusat yang tersentralisasi. Selain itu, data yang telah diupload ke sistem dapat disimpan secara permanen. Untuk data dengan request yang tinggi, IPFS akan membuat duplikasi data di sepanjang request path, yang berarti dapat langsung dibaca secara lokal pada request selanjutnya. IPFS terjamin aman, hasil yang tinggi, memiliki model penyimpanan *content addressed block* yang mendukung kapasitas tinggi dan akses bersamaan.



Gambar 2. 1 Model Penyimpanan IPFS

2.7.1. Contract Penyimpanan IPFS

Kontrak ini menyediakan fungsi utama berikut:

addFile: Pemilik atau pembuat kontrak dapat menjalankan fungsi ini. Peristiwa dipicu ketika file baru diunggah ke server IPFS, yang mengembalikan hash IPFS ke pemilik dan disimpan dalam kontrak pintar. Konten yang diunggah hanya diminta oleh pemohon yang sah kecuali yang masuk daftar hitam oleh pemiliknya

```
//Add file
function addFile(bytes32 id, bool isVisible) public {
require(!blacklist[msg.sender] && FileMap[id].timestamp == 0);
.
.
.
emit confirmFileIndexID( FileList.length-1, id);
}
```

deleteFile: Fungsi ini hanya dijalankan ketika pemilik ingin menghapus file tertentu dari server. Saat penghapusan file diperlukan, argumen indeks dan alamat adalah diteruskan ke fungsi ini.

```
// File deletion
function deleteFile(uint index, bytes32 id, address[] memory addr)
public {
.
.
.
FileList[index] = 0;
emit confirmFileDeletion( index, id);
}
```

DECENTRALIZES APP

DApps atau aplikasi terdesentralisasi (Decentralized App) adalah generasi baru dari aplikasi yang tidak dikendalikan atau dimiliki oleh otoritas tunggal, tidak dapat dimatikan atau tidak dapat mengalami downtime. Konsep DApp masih pada tahap awal. Berikut ini merupakan 4 karakteristik utama yang harus ada pada DApp yaitu[33]:

1. Open source

   Open source merupakan karakteristik paling penting karena aplikasi tersebut harus membuat core source code (kode inti) bisa tersedia untuk semua orang.

2. Lingkungan terdesentralisasi

   Seperti namanya, aplikasi yang terdesentralisasi menyimpan semuanya di Blockchain yang terdesentralisasi atau teknologi kriptografi apa pun untuk menyelamatkan aplikasi dari bahaya otoritas yang terpusat (sentralisasi) dan menekankan pada sifat otonom atau independen.

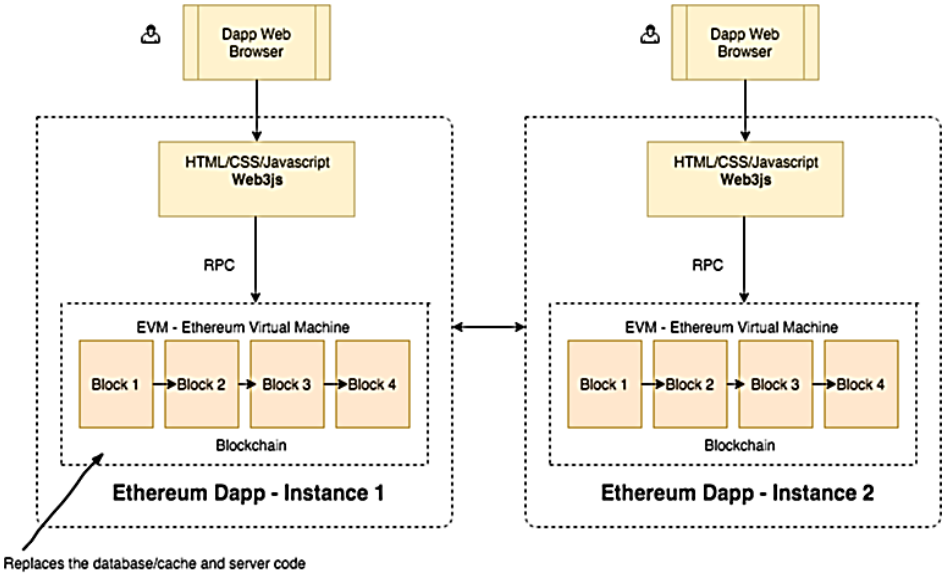
3. Algoritma

   Dapp perlu memiliki mekanisme konsensus yang menggambarkan bukti nilai dalam sistem kriptografi. Pada dasarnya, hal itu memberikan nilai pada token kriptografi dan membuat protokol konsensus yang disetujui pengguna untuk menghasilkan token kripto yang berharga.

4. Intensif

   Karena aplikasi ini didasarkan pada Blockchain yang didesentralisasi, validator catatan di jaringan harus dihargai / diberi insentif dengan token kriptografi atau segala bentuk aset digital yang memiliki nilai.

   Arsitektur Dapps dapat dilihat pada **Gambar 2.1**



Gambar 2. 2 Decentralized App

Sumber: https://www.commonlounge.com/

Setiap client-browser berkomunikasi dengan sistem aplikasi itu sendiri. Tidak ada server utama yang terkoneksi dengan browser client. Artinya bahwa, setiap orang yang ingin berinteraksi dengan Dapps harus mempunyai full copy Blockchain untuk menjalankan pada perangkat masing-masing. Namun dalam kenyataannya hal tersebut tidak diperlukan karena banyak beberapa solusi dan optimasi agar Dapps bisa selalu berinteraksi dengan cepat dan mudah. Dalam mekanismenya, Dapps pada Blockchain terdiri dari semacam 2 hal ini yaitu:

1. Database

   Setiap transaksi yang muncul pada jaringan Ethereum dikemas menjadi block dan tiap block terhubung satu sama lain.

2. Code

   Logika untuk beli, jual, cancel, pengembalian uang dibuat melalui smart contract dengan bahasa pemrograman Solidity.

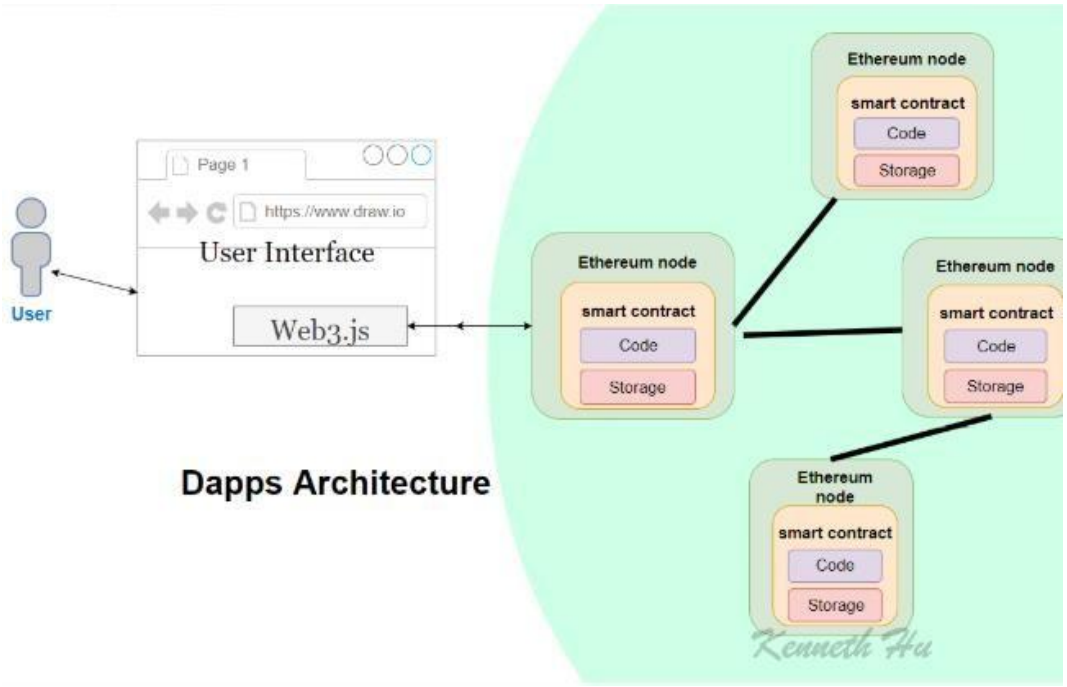
Dapps memiliki dua mekanisme secara umum dimana dalam tahap membangun pelaku consensus: mekanisme Proof of Work dan mekanisme Proof of Stake.

Mekanisme Proof of Work, keputusan tentang perubahan dalam DApp dibuat berdasarkan jumlah pekerjaan yang berkontribusi masing-masing stakeholder terhadap operasi DApp.

Dengan mekanisme Proof of Stake, keputusan tentang perubahan DApp dibuat berdasarkan persentase kepemilikan yang dimiliki berbagai stakeholder atas aplikasi tersebut. Sebagai contoh, suara pemegang saham yang mengontrol 10% token yang dikeluarkan oleh DApp, memiliki bobot 10%.

Web3.js

Web3.js adalah kumpulan library Javascript yang memungkinkan program untuk berinteraksi dengan node Ethereum lokal menggunakan HTTP, WebSocket, atau IPC. Dengan API Web3.js, front-end bisa berinteraksi dengan Smart Contract. Web3.js merupakan jembatan bagi JSON RPC Ethereum menggunakan bahasa pemrograman Javascript sebagai antarmukanya, yang memuatnya langsung dapat digunakan dalam aplikasi web karena Javascript mendukung secara native hamper di semua web browser. Web3.js juga digunakan pada server-side dengan Node.js dan pada desktop berbasis Electron. Web3.js dapat digunakan untuk terhubung ke jaringan Ethereum melalui node Ethereum yang memungkinkan akses melalui HTTP.



Gambar 2. 3 Arsitektur Dapps dengan Web3js

Sumber:https://medium.com/coinmonks/web3-js-ethereum-javascript-api-72f7b22e2f0a

Contoh menginisiasi penggunaan Web3:

```
If (window.ethereum){
        App.Web3Provider = window.ethereum;
        Try{
                Await window.ethereum.enable();
        } catch (error){
                Console.error("User denied account access")
        }
} else if (window.web3){
        App.web3Provider = window.web3.currentProvider;
} else {
        App.web3Provider = new

        Web3.provider.HttpProvider('http://localhost:7545');

}

Web3 = new Web3(App.web3Provider);
```

Pertama, kita cek dulu apakah menggunakan browser dapp modern atau sesuai versi dari Metamask dimana ethereum provider di-inject ke dalam objek window. Jika iya, maka kita buat objek web3nya dan juga kita membutuhkan akses

khusus ke akun kita dengan (ethereum.enable). Jika objek ethereumnya tidak ada, maka kita mengecek apakah web3nya sudah di-inject atau belum. Jika objek ethereumnya ada, maka terindikasi bahwa kita sedang menggunakan browser dap pang versi lama (seperti versi lama Metamask). Maka dari itu, kita menggunakan providernya dan pakai untuk membuat objek web3nya. Apabila web3 yang di-inject tidak muncul, kita buat iobjek web3nya pada local provider.

Rinkeby

Rinkeby adalah seorang Ethereum testnet (atau jaringan pengujian). Testnets biasanya digunakan oleh pengembang untuk menjalankan "tes" untuk aplikasi atau perangkat lunak mereka. Mata uang di testnet tidak berharga. Rinkeby testnet, tidak seperti mainnet Ethereum, adalah jaringan bukti otoritas, berlawanan dengan jaringan bukti kerja seperti mainnet Ethereum. Statistik untuk testnet Rinkeby dapat dilihat pada gambar 2.4 Tesnet Rinkeby.

Gambar 2. 4 Tesnet Rinkeby

OBJECT – ORIENTED PROGRAMMING (OOP)

OOP (Object Oriented Programming) atau yang dikenal dengan Pemrograman Berorientasi Objek merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus ke dalam kelas-kelas atau objek-objek. Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program,

dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

UML

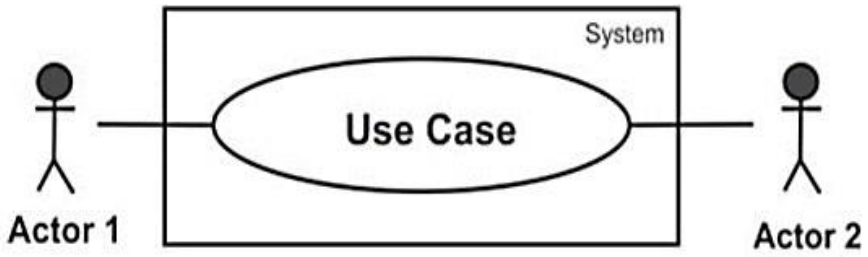
UML (Unified Modeling Language) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem. UML adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP[34].

Komponen UML

Pada bagian ini akan dijelaskan mengenai definisi use case, activity diagram, class diagram, dan sequence diagram.

a. Use Case Diagram

Diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem yang akan dibuat. Use Case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat . Syarat penamaan pada use case adalah nama didefinisikan sesederhana mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case.



Gambar 2. 5 Diagram Use Case

1. Aktor

   Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat di luar sistem yang akan dibuat itu sendiri, walaupun dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

2. Use Case

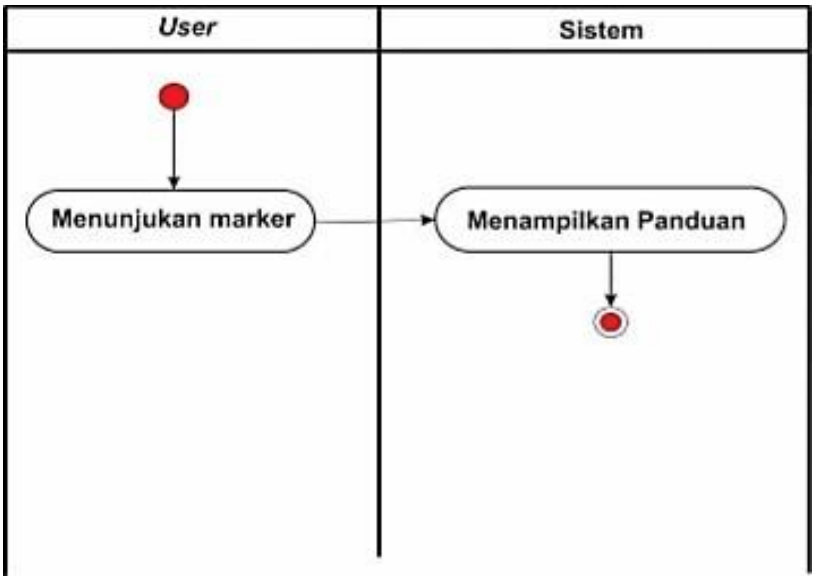
   Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang yang saling bertukar pesan antar unit maupun aktor. Berikut adalah simbol-simbol yang ada pada diagram use case.

Use case digunakan untuk mendeskripsikan interaksi antara actor dan mendeskripsikan fungsi – fungsi yang ada pada sistem.

b. Activity Diagram

Menggambarkan rangkaian aliran kerja (workflow) atau aktivitas dari sebuah sistem atau proses bisnis yang ada pada perangkat lunak. Bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor. Diagram aktivitas banyak digunakan untuk mendefinisikan hal-hal:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang akan didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.



Gambar 2. 6 Skema Relasi

c. Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Adapun susunan pada diagram kelas adalah memiliki jenis-jenis sebagai berikut :

1. Kelas main

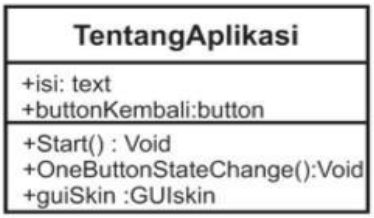
   Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
2. Kelas yang menangani tampilan sistem (view)

   Kelas yang mendefinisikan dan mengatur tampilan ke pemakai
3. Kelas yang diambil dari pendefinisian use case

   Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian use case, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.
4. Kelas yang diambil dari pendefinisian data

Kelas yang digunakan untuk memegang dan menghubungkan data menjadi sebuah yang diambil maupun akan disimpan ke basis data.

Pada sistem yang dibangun class diagram digunakan mendefinisikan kelas-kelas yang dibuat dalam membangun sistem. Semua class beserta dengan atribut dan methodnya akan didefinisikan dengan menggunakan class diagram.

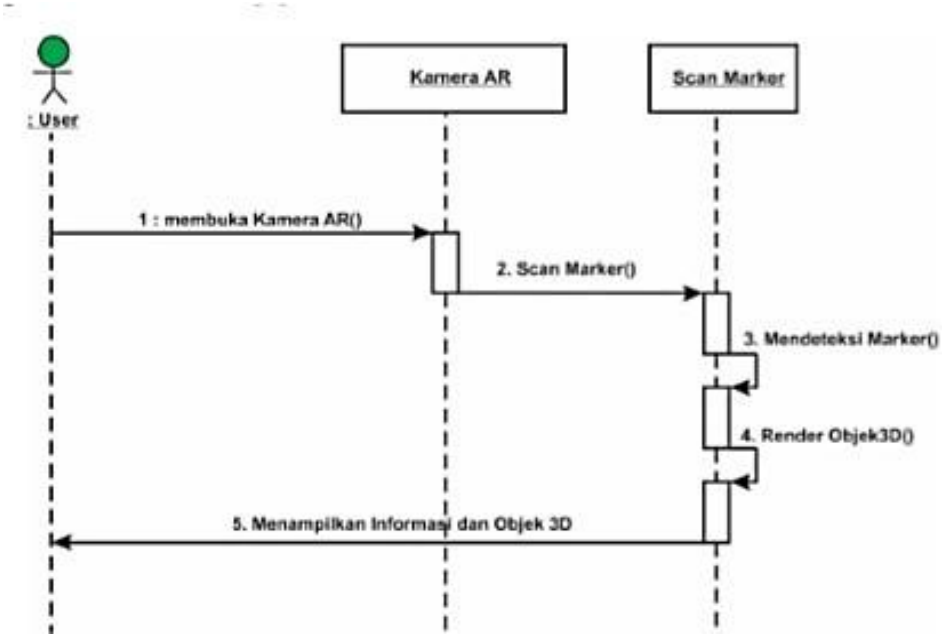


Gambar 2. 7 Diagram Class

d. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram.



Gambar 2. 8 Diagram Sequence

METODE PENGUJIAN SISTEM

Metode pengujian sistem untuk mengetahui efektifitas dari software yang dibangun selain memberikan kesempatan pada para pengguna untuk mengoperasikan dan melakukan pengecekan pada software. Metode pengujian sistem terdiri dari White-Box dan Black-Box.

Pengujian Black-Box

Pengujian kotak hitam atau Black-Box testing yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian Black-box dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian Black-box harus dibuat dengan kasus benar dan kasus salah[35].

Pengujian black-box berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya:

1. Fungsi – fungsi yang salah atau hilang,
2. Kesalahan interface,
3. Kesalahan dalam struktur data atau database eksternal,
4. Kesalahan performa,
5. Kesalahan inisialisasi dan terminasi.

Skala Data yang Digunakan

Skala data yang digunakan untuk mengukur variabel yang didapat, menggunakan skala likert. Skala likert adalah skala respon psikometri terutama digunakan dalam kuesioner untuk mendapatkan preferensi responden atas sebuah pernyataan atau serangkaian laporan. Skala Likert mempunyai dua bentuk pertanyaan dalam penggunaannya. Yakni bentuk pertanyaan positif untuk mengukur skala positif dan bentuk pertanyaan negatif untuk mengukur skala negatif. Untuk pertanyaan positif diberi skor 5, 4, 3, 2 dan 1. Lalu untuk pertanyaan negatif diberi skor 1, 2, 3, 4, dan 5 atau -2, -1, 0, 1, dan 2. Contoh dari kategori penilaian dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Kategori Penilaian

NO	Pertanyaan	SS	S	N	TS	STS
1	**Apakah Anda Setuju Jika Pemilihan Ketua Himpunan Dilakukan Secara Online (E-Voting)?**		✓			

Ket : SS = Sangat Setuju

S = Setuju

N = Netral

TS = Tidak Setuju

STS = Sangat Tidak Setuju

Rumus Skala Likert

T x Pn

T : Total jumlah responden yang memilih

Pn : Pilihan angka skore likert

Contoh :

Dari 50 orang mahasiswa jurusan Teknik informatika yang menjadi responden, diketahui rincian jumlah data seperti berikut:

- Sangat setuju: 20 orang
- Setuju: 5 orang
- Netral: 5 orang
- Tidak setuju: 15 orang
- Sangat tidak setuju: 5 orang

Penyelesaian :

- Sangat setuju: 20 x 5 = 100
- Setuju: 5 x 4 = 20
- Netral: 5 x 3 = 15
- Tidak setuju: 15 x 2 = 30
- Sangat tidak setuju: 5 x 1 = 5

Total Skor = 170

Interpretasi Skor Perhitungan

Jika ingin mengetahui hasil interpretasi harus mengetahui terlebih dahulu skor maksimum (X) dan minimum (Y), maka rumusnya adalah

X = jumlah responden x skor tertinggi.

Y = jumlah responden x skor terendah.

Skor Maksimum = Jumlah Peserta x Pn(Max)

Skor Minimum = Jumlah Peserta x Pn(Min)

Dari perhitungan ini kita bisa mengetahui interval penilaian Dengan rumus :

I = 100/ Jumlah Skor

Ini merupakan interval dari jarak terendah 0% hingga tertinggi 100%. Berikut adalah kriteria interpretasi skor berdasarkan intervalnya:

- Angka 0% – 19,99% = Sangat tidak setuju/buruk/kurang sekali
- Angka 20% – 39,99% = Tidak setuju / Kurang baik

- Angka 40% – 59,99% = Cukup / Netral
- Angka 60% – 79,99% = Setuju/Baik/suka
- Angka 80% – 100% = Sangat setuju/Baik/Suka

BAHASA PEMROGRAMAN

Pada tahap ini akan menjelaskan mengenai bahasa pemrograman yang dipakai. Terdapat 2 bahasa pemrograman yaitu Javascript dan Solidity. Javascript menurut (Sunyoto,2007:17) adalah bahasa scripting yang populer di internet dan dapat bekerja di sebagian besar browser populer seperti Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode Javascript dapat disisipkan dalam halaman web menggunakan tag SCRIPT.

Beberapa hal tentang Javascript:

1. Javascript didesain untuk menambah interaktif suatu web
2. Javascript merupakan sebuah bahasa scripting.
3. Bahasa scripting merupakan bahasa pemrograman yang ringan.
4. Javascript berisi baris kode yang dijalankan di computer (web browser).
5. Javascript biasanya disisipkan (embedded) dalam halaman HTML.
6. Javascript adalah bahasa interpreter (yang berarti sintaks dieksekusi tanpa proses kompilasi).

React Js

React.js adalah library (pustaka) UI yang merupakan bagian dari proyek dan dikembangkan salah satu perusahaan teknologi terbesar di dunia yaitu Facebook untuk berfokus pada kompatibilitas dan kecepatan kinerja pada bagian tampilan aplikasi Front-End[36]. React dapat melakukan rendering user interface (UI) yang kompleks dengan kinerja yang tinggi[37]. Hal yang mendasar dibalik React adalah Virtual DOM. React JS secara efektif menggunakan Virtual DOM yang dapat membuat fungsi baik dari client-side atau server-side. Virtual DOM membuat subtree dari node berdasarkan perubahan status dan dapat memanipulasi DOM sesedikit mungkin untuk menjaga tiap komponen tetap di-update.

React memanfaatkan ekstensi berbasis XML yang nyaman untuk JavaScript, yang dikenal sebagai JSX, untuk menghasilkan komponen sebagai kombinasi dari beberapa node XML[38].

Solidity

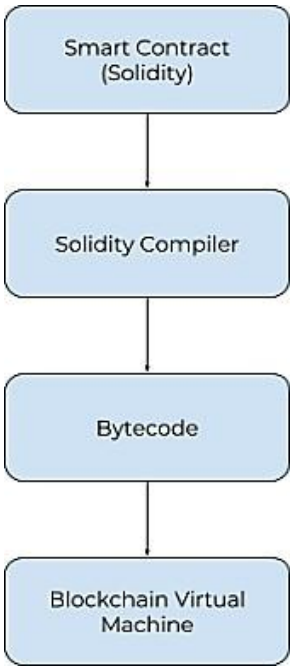
Solidity adalah bahasa berorientasi objek (Object Oriented Programming) yang dirancang untuk menulis Smart Contract dalam Ethereum dan merupakan bahasa tingkat tinggi serta bisa dikatakan sebagai bahasa berorientasi kontrak (Contract Oriented Programming) dengan eksperimental pertama[39]. Solidity ditulis secara statis, mendukung inheritance (pewarisan) library, dan tipe kompleks yang ditentukan pengguna di antara fitur-fitur lainnya. Bahasa pemrograman Solidity mirip dengan javascript karena pada javascript terdapat class, class yang terdapat pada soldity disebut contract. Syntax dan function nya pun banyak kemiripan dengan javascript. Contoh penggunaan syntax pada Solidity:

```
pragma solidity >=0.4.0 < 0.7.0;
contract SimpleStorage{
        uint storedData;
        function set(uint x) public {
                storedData = x;
        }
        function get() public view returns (uint){
                return storedData;
        }
}
```

Penulisan syntax solidity pada baris pertama source code dimulai dengan syntax “pragma solidity” untuk memulai dengan bahasa pemrograman solidity dilanjutkan dengan range versi dari solidity “0.4.0 < 0.7.0”.

Solidity ditulis dengan format file .sol. Sebenarnya solidity bukanlah bahasa yang sebenarnya yang dieksekusi pada Blockchain Virtual Machine.

Solidity adalah bahasa yang bertujuan untuk mempermudah pembuatan smart contract. Saat akan compile ataupun deploy smart contract, dibutuhkan yang namanya Solidity Compiler. Melalui solidity compiler tersebut smart contract akan di compile ke dalam bytecode yang nantinya bytecode tersebut lah yang akan dieksekusi oleh virtual machine.



Gambar 2. 9 Alur Solidity

Sumber: https://medium.com/@emurgo_id/bahasa-pemrograman-solidity-6cd597503a6f

Ada beberapa tipe data basic pada solidity, seperti hal nya bahasa pemrograman lain nya. Dalam solidity ada string, integer, array, dan lain-lain.

1. String

   Sama seperti bahasa lain nya, tipe data string digunakan untuk membuat variabel yang berupa sekumpulan karakter.
2. Bool

   Bool adalah tipe data Boolean.
3. Int

Int tau biasa disebut integer. Dalam solidity, tipe data ini harus berupa angka positif atau negatif. Tidak ada desimal untuk tipe data ini. Pada tipe data **int** ini dibagi menjadi int8**,** int16**,** int32**, …,** int256. Angka pada tipe data int tersebut adalah sebuah angka bit yang akan digunakan untuk menyimpan angka yang dispesifikasikan seberapa besar angka yang dimiliki oleh variable.

Tabel 2. 2 Tipe Integer

Integer		
Nama	Batas Bawah	Batas Akhir
Int8	-128	127
Int16	-32.768	32.767
Int32	-2.147.48.648	-2.147.483.647
…	…	…
Int256	Angka yang sangat kecil	Angka yang sangat besar

4. Uint

Uint atau biasa disebut unsigned integer. Dalam solidity, tipe data ini harus berupa angka positif kebalikan dari int. Tidak ada angka negatif dan desimal untuk tipe data ini. Sama seperti tipe data **int**, pada tipe data **int** ini dibagi menjadi **uint8**, **uint16**, **uint32**, **…**, **uint256**. Angka pada tipe data uint tersebut adalah sebuah angka bit yang akan digunakan untuk menyimpan angka yang dispesifikasikan seberapa besar angka yang dimiliki oleh variabel.

Tabel 2. 3 Tipe Unit

Unit		
Nama	Batas Bawah	Batas Akhir

uint8	0	255
uint16	0	65.535
uint32	0	4.294.967.295
…	…	…
uint256	0	Angka yang sangat besar

5. Fixed / unfixed

   Fixed / fixed adalah tipe data untuk variabel angka yang mempunyai nilai dan bilangan desimal.

6. Address

   Address dalam solidity untuk membuat membuat variabel yang berisi address account ataupun address contract account. Solidity dibuat khusus untuk mengembangkan smart contract sehingga tipe data address dibuat. Address account dibuat dalam bentuk angka acak dengan tipe hexadecimal. Contoh address dalam hexadecimal adalah "0x2a9054870cBB655ae3cd5F8231199dA7aBb6b0b4".

Node Js

Node.js adalah teknologi I / O asinkron open source dan event-driven untuk membangun server web yang scalable dan efisien[40]. Node.js merupakan platform yang tangguh untuk mengembangkan aplikasi yang tak terhitung jumlahnya. Developer dapat menulis aplikasi dalam satu bahasa yaitu Javascript yang mampu berjalan di browser, server, dan berbentuk mobile. Javascript hanya berjalan pada client-side, makan Node Js ada untuk melengkapi peran Javascript pada sisi server-side. Node Js memiliki library server HTTP sendiri sehingga memungkinkan untuk menjalankan server web tanpa menggunakan program server seperti Apache atau Nginx.

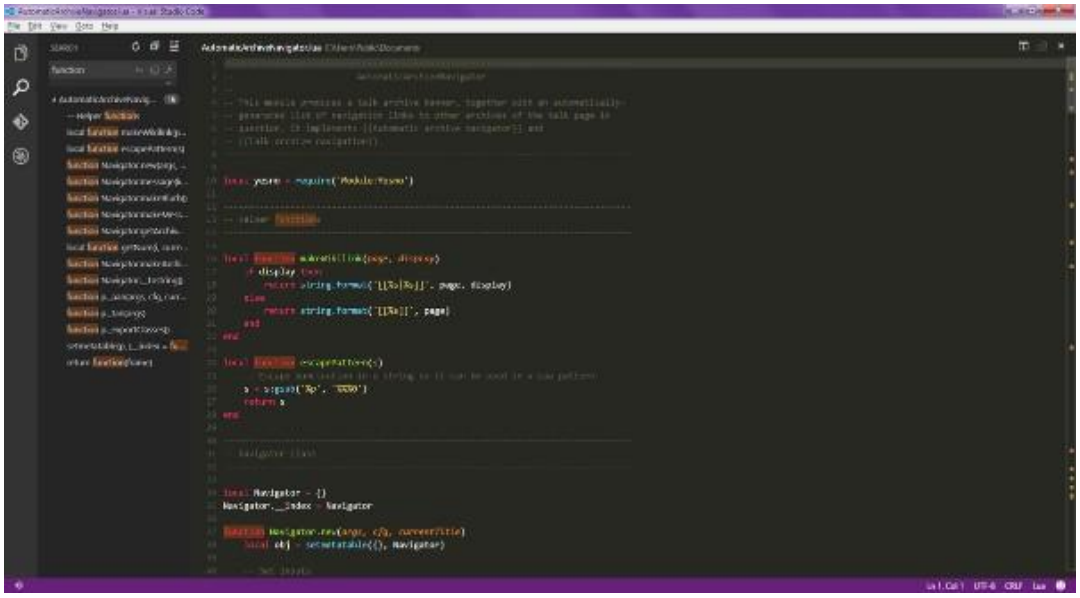
Berbeda dengan bahasa pemrograman sisi server pada umumnya yang bersifat blocking, Node.js bersifat non-blocking, sebagaimana halnya JavaScript bekerja. Node.js berjalan dengan basis event (event-driven). Maksud dari Blocking secara sederhana adalah, bahwa suatu kode program akan dijalankan hingga selesai, baru kemudian beralih ke kode program selanjutnya.

VISUAL STUDIO CODE

Microsoft Visual Studio Code adalah one-stop shop yang memungkinkan kita fokus pada proses pengembangan dan melupakan tools baru. Belakangan tersedia banyak tools yang membantu developer menulis, mengetes, debug, dan deploy. Kita bisa berganti-ganti tools tergantung pada tugas yang dikerjakan, seperti Eclipse untuk Java, IDE untuk Python, Git Bash untuk source code control. Semua tools ini ada untuk membuat hidup kita jadi lebih mudah. Tapi terkadang demi memahami, melakukan konfigurasi, dan menggunakan tool baru, kita menghabiskan banyak waktu. Lihat nih beberapa fitur Visual Studio Code:

1. Cross platform – tersedia di macOS, Linux dan Windows artinya Anda dapat bekerja pada sistem operasi manapun tanpa khawatir belajar coding tools yang sama untuk sistem yang berbeda-beda.
2. Lightweight – tak perlu menunggu lama untuk memulai. Anda mengontrol sepenuhnya bahasa, tema, debugger, commands dan lain-lainnya sesuai keinginan. Ini dapat dilakukan melalui extensions untuk bahasa populer seperti python, node.js, java dan lain-lainnya di Visual Studio Code.
3. Powerful editor – memfungsikan fitur untuk source code editing yang sangat produktif, seperti membuat code snippets, IntelliSense, autocorrect, dan formatting.
4. Code Debugging – salah satu fitur terkeren yang ditawarkan Visual Studio Code adalah membantu Anda melakukan debug pada kode dengan cara mengawasi kode, variabel, call stack dan expression yang mana saja.
5. Source control – Visual Studio Code memiliki integrated source control termasuk Git support in-the-box dan penyedia source code control lainnya di pasaran. Ini meningkatkan siklus rilis proyek Anda secara signifikan.

Integrated terminal – Tiada lagi multiple windows dan alt-tabs. Anda dapat melakukan command-line task sekejap dan membuat banyak terminal di dalam editor.



Gambar 2. 10 Tampilan Visual Studio Code

METAMASK CHROME EXTENSION

Metamask adalah dompet (wallet) yang berbasis extension yang berjalan pada browser (Chrome, Firefox, Opera, atau Brave Browser) dapat dengan mudah digunakan dan nyaman untuk pengujian karena dapat terhubung ke berbagai node Ethereum untuk menguji Blockchain. Metamask merupakan dompet (wallet) Ethereum berbasis web. Metamask bisa juga disebut portal bagi web2 dan web3 sehingga biasa disebut "web3 provider". Dengan kata lain, metamask memungkinkan penggunanya untuk menyimpan data terkait Ethereum seperti alamat Public dan private key seperti dompet Ethereum lainnya, dan memungkinkan pengguna untuk berinteraksi dengan situs web yang menjalankan aplikasi berbasis Ethereum dan smart contract (web browser menjadi Ethereum browser).