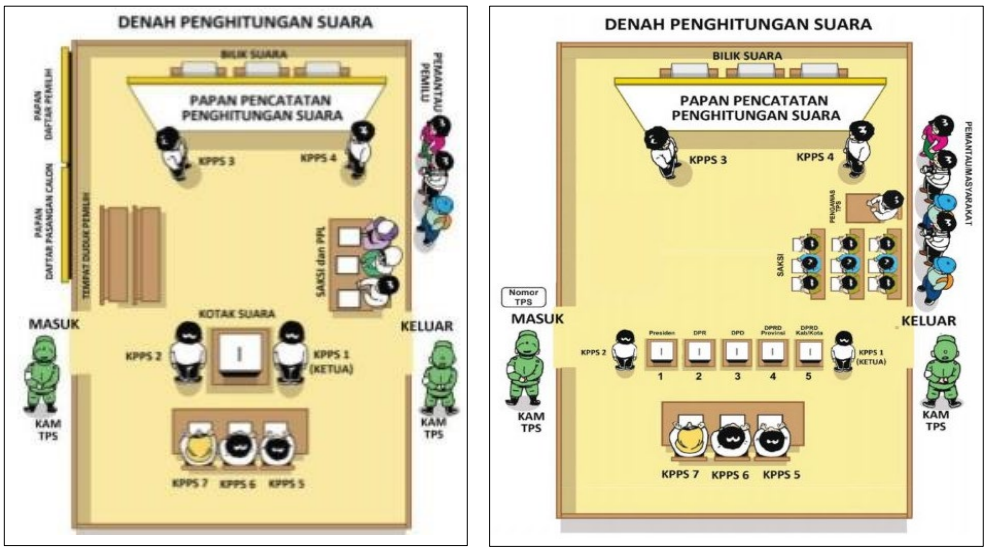
BAB II

TINJAUAN PUSTAKA

Pemilihan Umum adalah proses memilih seseorang untuk mengisi jabatan tertentu yang diselenggarakan seharusnya sesuai dengan asas langsung, umum, bebas, rahasia, jujur, dan adil dalam Negara Kesatuan Republik Indonesia berdasarkan Pancasila dan Undang – Undang Dasar Negara Republik Indonesia tahun 1945. Berikut ini merupakan beberapa contoh mekanisme pemungutan suara yang sudah terdapat dan sering terjadi di Indonesia:

1. Tradisional yaitu datang ke tempat pemungutan suara

   Contoh paling mudah dan sangat terlihat adalah pemilihan umum kepala pemerintahan. Hingga saat ini, proses pemungutan suara dilakukan secara *onsite* di mana para pemilih datang ke tempat pemungutan suara sesuai yang tertera pada daftar pemilih tetap, melakukan verifikasi dengan menyerahkan surat undangan hak pilih (seperti model C-6) ataupun dengan tanda pengenal lainnya, lalu menunggu giliran untuk memilih pada ruangan yang disediakan. Gambar 2.1 merupakan denah diberlakukannya pemilu saat ini.

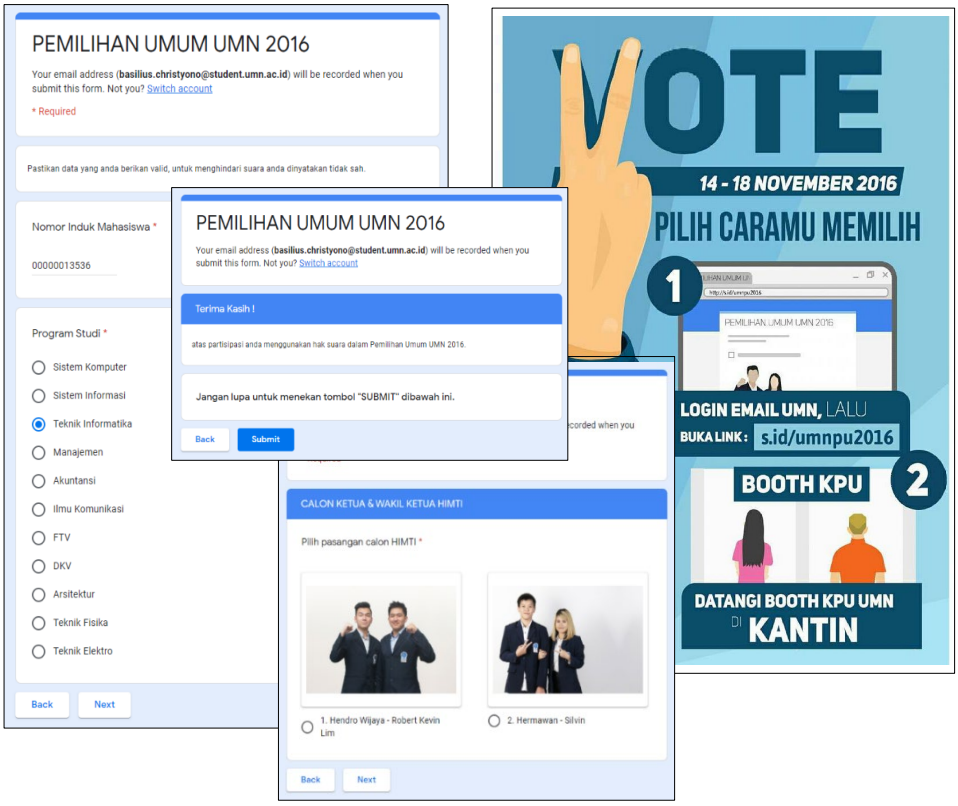


Gambar 2.1 Komponen Denah Tempat Pemungutan Suara (KPU RI, 2019)

2. Elektronik dengan *server* terpusat

Beberapa pemungutan suara yang sering ditemui dengan sistem seperti ini yaitu misalnya survei, ujian, petisi, bahkan pemilihan ketua kelas hingga organisasi lembaga dalam kampus. Cara ini memang memiliki keuntungan efisien dan lebih hemat biaya, namun siapa pun yang memiliki akses ke *server* tersebut dapat melihat atau mungkin juga mengubah data. Sebagai contoh yang ada di lingkungan sekitar misalnya pemilihan ketua himpunan yang ada di lingkungan kampus Universitas Multimedia Nusantara. Pada pemilihan ini, sistem yang digunakan sudah dilakukan secara elektronik yaitu dengan menggunakan Google *Form* yang nantinya data hasil akan tersimpan ke dalam Google *Sheet* hanya dengan persyaratan sebagai berikut:

- Pemilihan dilakukan menggunakan akun surel khusus yang dimiliki oleh masing-masing mahasiswa
- Hanya mahasiswa aktif yang suaranya akan dianggap sah
- Setiap akun hanya dapat satu kali memilih
- Setiap akun memilih kandidat dari daftar program studinya, apabila sesuai, maka akan dianggap sebagai suara yang sah



Gambar 2.2 Cara Pengisian Surat Suara (KPU UMN, 2016)

Gambar 2.2 merupakan penggunaan *form* untuk pengumpulan data. Penggunaan *Google Form* selain memudahkan dalam penggunaan, juga cepat dalam proses persiapan hingga pembuatan, hanya saja sistem ini kurang tepat apabila digunakan untuk pemungutan suara yang bersifat

rahasia yang bahkan pembuatnya saja tidak boleh tahu apa yang dipilih oleh pengguna karena cara ini biasanya digunakan untuk mengumpulkan data yang sifatnya informatif dan memungkinkan untuk digunakan sebagai keperluan statistik misalnya kuesioner. Setelah pengguna memberikan suaranya, akan ada periode rekapitulasi suara di mana seluruh data surat suara yang masuk akan di cek untuk menentukan apakah akun pemilih tersebut merupakan mahasiswa aktif dan memilih sesuai dengan jurusannya sebagai suara sah kemudian dilakukan penghitungan suara.

Hingga saat ini penggunaan *blockchain* pada umumnya digunakan hanya terbatas pada *cryptocurrency*. Maka pada penelitian kali ini, akan merancang sistem pemungutan suara elektronik yang menggunakan *blockchain* dengan *server* yang terdesentralisasi sehingga memungkinkan pemilih untuk dapat memberikan hak suaranya kapan pun dan di mana saja selama periode pemungutan berlangsung.

2.1. Blockchain

Blockchain dapat diartikan sebagai struktur data yang memungkinkan untuk membuat buku digital dari data dan berbagi data dalam jaringan secara terdesentralisasi. Setiap blok yang terkandung dalam teknologi blockchain selalu dikaitkan dengan satu blok sebelum dan satu blok sesudahnya. Saat ini ada beberapa jenis blockchain (Laurence, 2017), yaitu:

- *Public Blockchains*

Blockchain publik, seperti Bitcoin, adalah jaringan terdistribusi besar yang dijalankan secara terbuka bagi siapa saja untuk berpartisipasi ditingkat mana pun dan biasanya bersifat *open-source* yang dikelola oleh komunitas mereka.

- *Permissioned Blockchains*

  Blockchain dengan izin, seperti Ripple, mengontrol peran yang dapat dimainkan individu dalam jaringan. Mereka masih tergolong cukup besar. Untuk jenis ini biasanya sudah tidak lagi *open-source*.

- *Private Blockchains*

  Blockchain secara tertutup, cenderung lebih kecil. Keanggotaan mereka dikendalikan dengan ketat. Jenis *blockchain* ini disukai oleh suatu lembaga atau organisasi tertentu yang memiliki anggota terpercaya dan mendistribusikan informasi rahasia.

Ketiga jenis *blockchain* menggunakan kriptografi agar memungkinkan setiap peserta di jaringan tertentu dapat mengelola buku besar dengan cara yang aman tanpa perlu otoritas pusat untuk menegakkan aturan. Teknologi baru ini bekerja melalui empat fitur utama (Cachin dan Vukolić, 2017) yaitu:

1. Buku besar ada di banyak lokasi berbeda, tidak ada titik kegagalan dalam pemeliharaan buku besar yang didistribusikan.
2. Adanya kontrol terdistribusi atas siapa yang dapat menambahkan transaksi baru ke buku besar.

3. Setiap blok baru yang diusulkan untuk buku besar harus merujuk ke versi buku besar sebelumnya, menciptakan rantai yang tidak dapat diubah dari mana *blockchain* mendapatkan namanya, dan dengan demikian mencegah gangguan dengan integritas entri sebelumnya.
4. Mayoritas *node* jaringan harus mencapai konsensus sebelum blok entri yang diusulkan menjadi bagian permanen dari buku besar.

2.2. Genesis Blok

Ethereum, secara keseluruhan dapat dilihat sebagai *transaction-based state machine* di mana jaringan blockchain dimulai dengan keadaan asal (disebut sebagai *genesis*) dan secara bertahap melakukan transaksi untuk berubah keadaan tersebut menjadi beberapa kondisi terkini (Gavin, 2014). Kondisi *state* dapat mencakup informasi seperti saldo akun, reputasi, pengaturan, maupun data yang berkaitan dengan informasi dunia fisik; singkatnya, apa pun yang saat ini dapat direpresentasikan oleh komputer, dapat diterima dan dimasukkan ke dalam *blockchain*. Jaringan Ethereum dikatakan sebagai jaringan *private* jika *node* tidak terhubung ke *node* jaringan utama *blockchain* ethereum yang umum di internet. Dalam konteks ini privat berarti terbatas atau terisolasi, dan memiliki perlakuan khusus. Adapun tampilan blok *genesis* merupakan data yang berbentuk JSON (*JavaScript Object Notation*) seperti Gambar 2.3.

Gambar 2.3 Pengaturan Awal Blockchain

Semua pengaturan angka atau nilai menggunakan hexadesimal. Beberapa pengaturan yang perlu diperhatikan:

- ***config***: File dimulai dengan blok "*config*" yang berisi semua parameter konfigurasi dan ambang batas yang mengontrol operasi dasar jaringan.
  - ***chainId***: Digunakan untuk memisahkan jaringan *blockchain* yang satu dengan lainnya.
  - ***clique***: Algoritma dari *Proof-of-Authority* (PoA) menentukan waktu untuk melakukan *mining* dan *commit*.

- ***difficulty***: Ini menentukan seberapa sulit untuk menambang di jaringan *blockchain*. Dilakukan penyetelan ke nilai serendah mungkin sehingga tidak perlu menunggu terlalu lama hingga blok berhasil ditambang.
- ***gasLimit***: Digunakan untuk menentukan nilai batas maksimum biaya *gas* setiap satu blok (total *gas* seluruh transaksi yang akan berada di dalam blok).
- ***nonce & mixhash***: *Nonce* dan *mixhash* adalah nilai yang ketika digabungkan, memungkinkan untuk memverifikasi bahwa suatu blok telah ditambang secara kriptografi, dan dengan demikian dianggap valid. *Mixhash* adalah *hash* 256-bit yang membuktikan, ketika dikombinasikan dengan 64-bit *nonce*, bahwa jumlah komputasi yang cukup telah dilakukan pada blok ini: *Proof-of-Work* (PoW). Namun karena pada penelitian ini menggunakan PoA maka nilainya di nol kan saja.
- ***extraData***: Menentukan daftar dompet atau akun dapat melakukan *mining* menciptakan blok baru.
- ***alloc***: Menentukan daftar dompet yang sudah diisi koin saat *blockchain* ini dimulai.

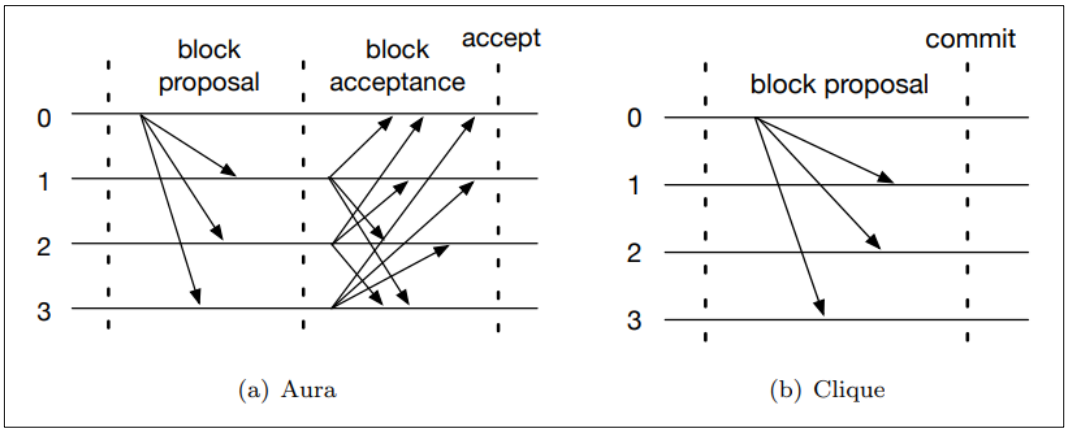
2.3. Proof-of-Authority Consensus

Proof-of-Authority (PoA) adalah sebuah algoritma konsensus berbasis reputasi yang merupakan sebuah solusi praktis dan efisien untuk jaringan *blockchain*. PoA memiliki struktur yang mirip dengan *Delegated Proof of Stake*

(DPoS), di mana jumlah *validator* sangat terbatas. Perbedaannya adalah bahwa dalam DPoS mereka dipilih oleh masyarakat / *node*, yaitu semua pemegang *token*, sedangkan di PoA mereka dipilih setelah pemeriksaan menyeluruh, biasanya dari entitas yang mengendalikan proyek blockchain ataupun dari kelompok tertentu yang dapat dipercaya. Algoritma konsensus PoA menggunakan nilai identitas, yang berarti *validator* menggunakan reputasinya untuk dapat membuat atau menulis blok data baru ke dalam rantai. Maka dari itu *blockchain* PoA diamankan dengan *node* validasi yang terpilih sebagai entitas yang dapat dipercaya. Model bergantung pada jumlah blok validator yang terbatas dan ini membuat sistem yang lebih dapat berskala. Blok dan transaksi diverifikasi oleh peserta yang sudah disetujui, yang berlaku sebagai moderator sistem.

Algoritma PoA mengandalkan seperangkat jumlah N *node* terpercaya yang disebut otoritas misalnya lembaga Komisi Pemilihan Umum dan lembaga Badan Pengawas Pemilu. Setiap otoritas diidentifikasi oleh *id* unik dan mayoritas dari jumlah mereka akan dianggap jujur, yaitu setidaknya N/2+1. Pihak berwenang menjalankan konsensus untuk menjalankan transaksi yang diminta oleh klien. Konsensus dalam algoritma PoA bergantung pada skema rotasi *mining*, sebuah pendekatan yang banyak digunakan untuk mendistribusikan pembuatan blok secara adil di antara pihak berwenang. Waktu dibagi menjadi beberapa langkah, yang masing-masing memiliki otoritas terpilih sebagai pemimpin *mining*. Pada Gambar 2.5 ada beberapa mekanisme yang digunakan untuk membuat sebuah blok baru yaitu *Authority Round* (atau disebut juga *AuRa*) dan *Clique*. Kedua implementasi PoA ini bekerja dengan cara yang sangat berbeda, keduanya memiliki babak

pertama di mana blok baru diusulkan oleh pemimpin saat ini (*block proposal*). *AuRa* membutuhkan putaran lebih lanjut (*block acceptance*), sedangkan *Clique* tidak.



Gambar 2.5 Pertukaran Pesan *AuRa* dan *Clique*. (DPoS, 2018)

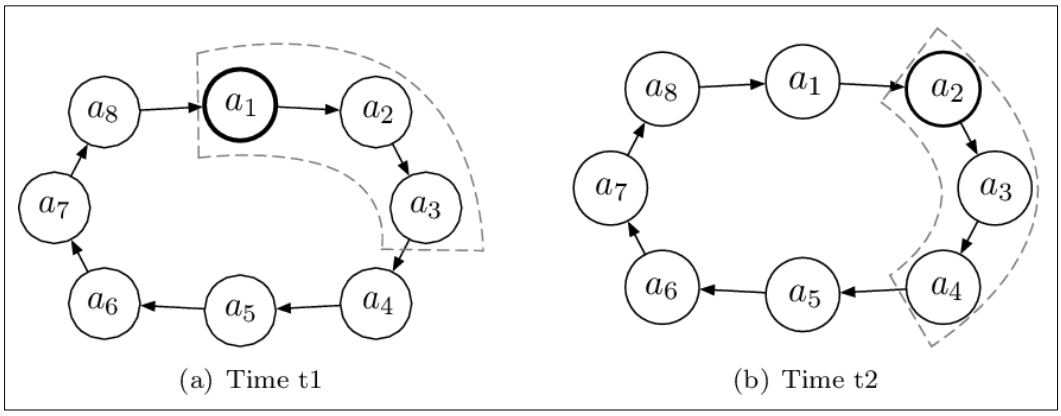
- *Authority Round* (*AuRa*)

  AuRa merupakan algoritma PoA yang diimplementasikan dalam Parity, sebuah *Ethereum Client* yang menggunakan bahasa Rust.

  1. Pemimpin blok tersebut kemudian mencoba membuat blok baru dari daftar transaksi yang ada (disebut juga *block proposal*), lalu mendistribusikannya kepada seluruh *validator* lainnya.
  2. Seluruh *validator* menerima proposal dari blok tersebut sebagai *block acceptance.*
  3. Jika seluruh *validator* lainnya (kecuali pemimpinnya) menerima blok yang sama, maka blok tersebut dinyatakan diterima dan akan didistribusikan ke dalam jaringan sebagai blok permanen. Jika tidak menerima blok yang sama, seluruh *validator* akan ambil

voting untuk menentukan apakah blok yang dikirimkan dari pemimpin ini dianggap tidak sah dan berbahaya atau tidak.

4. Jika *block leader* dianggap sebagai aktor jahat oleh mayoritas *validator* lain, mereka akan dikeluarkan atau ditahan dari daftar otoritas dan seterusnya kehilangan hak untuk membuat blok. Atau jika tidak, proses pembuatan blok akan diulang dari awal dengan memilih pemimpin yang baru sebagai pengganti ronde yang tidak sah tersebut.



Gambar 2.6 Seleksi *Validator* Untuk Membuat Blok, *Clique* (DPoS, 2018)

- *Clique* (*epochs*)

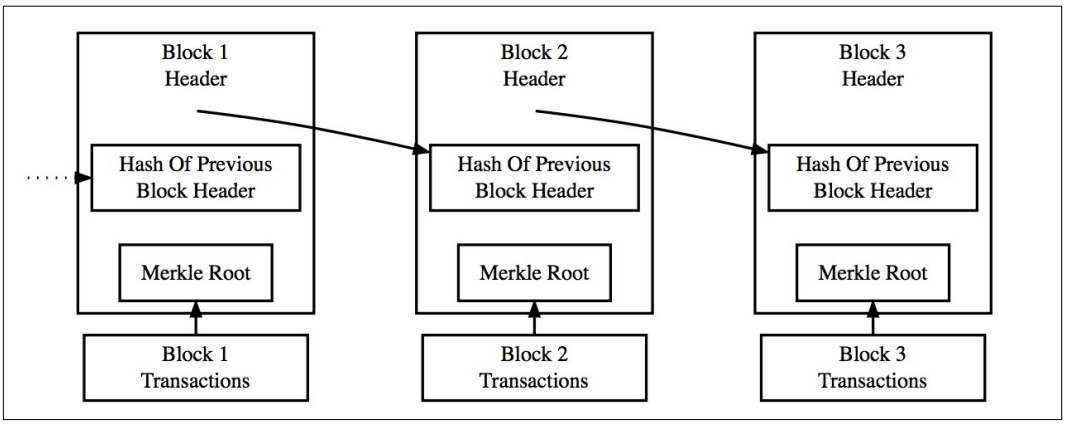
  Clique merupakan algoritma PoA yang diimplementasikan dalam Geth, sebuah *Ethereum Client* yang menggunakan bahasa GoLang. Dalam *Clique*, sebuah blok dapat diusulkan tidak hanya oleh pemimpin blok, tetapi juga oleh sejumlah *validator* lainnya (setengah dari jumlah seluruh *validator* - 1) seperti yang terdapat pada Gambar 2.6. Sebagai contoh, katakanlah jumlah *validator* ada 8 dan terdaftar sebagai angka

1 hingga 8. Formula di atas memberi informasi bahwa maksimal tiga *validator* dapat mengajukan / membuat blok baru.

1. Untuk blok pertama, 1 adalah pemimpin blok, dan *validator* 2 dan 3 juga dapat mengusulkan blok.
2. Blok yang dibuat oleh pemimpin blok memiliki prioritas di atas yang lain, tetapi jika karena alasan apa pun pemimpin blok gagal untuk mengusulkan yang lain akan tetap mengusulkan dan blok yang diusulkan pertama akan dianggap menang.
3. Setelah blok di *commit* ke blockchain secara permanen, *clique* akan berubah di periode pembuatan blok berikutnya, selanjutnya 2 adalah pemimpin blok dan *validator* 3 dan 4 juga dapat mengusulkan blok.
4. Proses ini berulang sampai semua *validator* menjadi pemimpin blok, dan berputar lagi dari awal.

2.4. Kriptografi Antar Blok

Hashing merupakan salah satu teknik kriptografi dalam menghitung nilai unik (Eastlake dan Hansen, 2001) yang dapat diibaratkan sebagai sidik jari dari sebuah data dan merupakan sebuah identitas dari suatu blok data yang digunakan untuk menghubungkan blok baru dengan blok sebelumnya seperti yang dijelaskan pada Gambar 2.7. Contohnya adalah *Secure Hash Algorithms* (SHA).



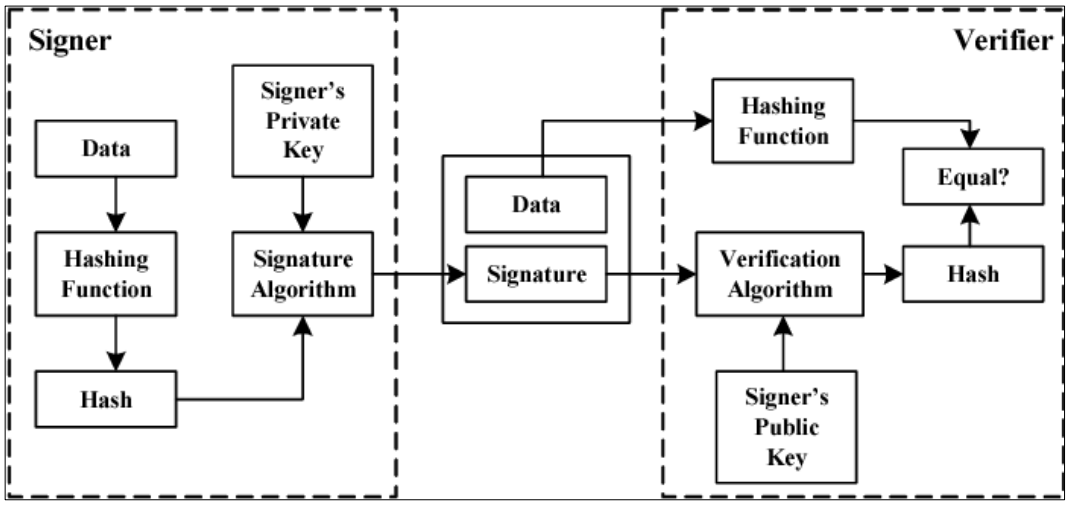
Gambar 2.7 Ilustrasi *Blockchain* (http://www.blockchain.org)

2.5. Kriptografi Transaksi Data Pemilih

Elliptic Curve Digital Signature Algorithm (ECDSA) adalah sebuah tanda tangan digital yang digunakan untuk autentikasi konten secara digital (Johnson *et al*., 1992) sebagai cara untuk mengetahui keaslian konten ataupun kepemilikan. Algoritma ini akan menghasilkan kunci yang saling terkait secara matematis. Gambar 2.8 merupakan rangkaian proses tanda tangan dan verifikasi suatu pesan secara matematis.

- ***Private Key***: Nomor rahasia, hanya diketahui oleh orang yang membuatnya. Kunci pribadi pada dasarnya adalah nomor yang dibuat secara acak.
- ***Public Key***: Nomor yang terkait dengan kunci pribadi, tetapi tidak perlu dirahasiakan. Kunci publik dapat dihitung dari kunci pribadi, tetapi tidak sebaliknya. Kunci publik dapat digunakan untuk menentukan apakah tanda tangan asli (dengan kata lain, diproduksi dengan kunci yang tepat) tanpa memerlukan kunci pribadi untuk diungkapkan.

- ***Signature***: Nomor yang membuktikan bahwa operasi penandatanganan berlangsung. Tanda tangan secara matematis dihasilkan dari hash sesuatu yang akan ditandatangani, ditambah dengan kunci pribadi. Lalu dengan adanya kunci publik, algoritma matematika dapat digunakan pada tanda tangan untuk menentukan bahwa itu awalnya dihasilkan dari hash dan kunci pribadi, tanpa perlu mengetahui kunci pribadi.



Gambar 2.8 Proses *Sign* Dan Verifikasi ECDSA (APMediaCast, 2016)

2.6. Ethereum Smart Contract

Ethereum merupakan jaringan *peer-to-peer* publik atau *blockchain* dengan mata uang digitalnya sendiri yang disebut Ether. Ethereum diciptakan oleh Vitalik Buterin pada tahun 2014 dan tujuan Ethereum adalah untuk menjadi *platform* di mana *smart contracts* dapat diciptakan dan dijalankan. Karakteristik Blockchain Ethereum bersifat *statefull* dan *Turing Completeness* (Dickerson *et al*., 2017) sehingga dimungkinkan untuk menyimpan data selain transaksi dan membuat program yang kompleks. Saat ini ethereum memiliki dua metode konsensus yakni

Proof-of-Work di mana konsensus ini merupakan proses pembuatan blok dengan cara menyelesaikan perhitungan komputasi secara matematis pada tingkat kesulitan tertentu dan siapa pun yang berhasil dengan cepat menyelesaikannya atau membuat blok barunya akan mendapatkan koin sebagai bentuk imbalan karena telah memproses transaksi menjadi blok baru, metode konsensus yang lainnya adalah *Proof-of-Authority* di mana hanya *node* tertentu saja yang dapat berpartisipasi dalam proses pembuatan blok baru, namun dalam hal ini tidak ada tingkat kesulitan tersendiri dan proses pembuatan blok barunya sudah ditargetkan pada waktu tertentu. *Smart contract* adalah sebuah fitur yang ditawarkan oleh Ethereum dan merupakan protokol komputer yang berfungsi untuk memfasilitasi, memverifikasi, atau menegakan negoisasi secara digital yang ditulis melalui kode program. *Smart contract* bekerja tanpa melalui pihak ketiga dan memiliki proses transaksi yang kredibel sehingga tidak bisa di manipulasi ataupun diubah. Dengan menggunakan *smart contracts*, pengguna dapat melakukan pertukaran uang, properti, saham atau apa pun secara transparan, tanpa konflik dan tanpa perantara.

Semua pengguna yang terlibat dalam pemungutan suara dapat dibagi menjadi tiga jenis antara lain pemilih, penyelenggara dan *smart contract* itu sendiri.

- Pemilih, merupakan orang yang telah diberikan hak sebagai partisipan untuk memilih kandidat yang ada dalam suatu pemilu.
- Penyelenggara, adalah pihak yang membuat atau mengadakan pemilu.
- *Smart Contract*, berisi fungsi prosedur yang digunakan selama pemilu.

Sehingga dapat disimpulkan bahwa:

$$Users = Pemilih \cup Penyelenggara \cup Smart\ Contract \quad \text{(i)}$$

Seluruh pengguna memiliki akun Ethereum masing-masing dan semua aktivitas yang dilakukan akan dianggap sebagai transaksi yang kemudian direkam dan disimpan dalam *blockchain*, dapat direpresentasikan sebagai berikut:

$$a \xrightarrow[memo]{c} b \quad \text{di mana} \quad a, b \in Users \qquad \text{(ii)}$$

dan c merupakan data objek yang dikirimkan, memo merupakan data opsional misalnya catatan, pesan atau deskripsi yang ingin dilampirkan dalam transaksi.

2.7. The Tallying Vote Model

Saat ini, Indonesia belum menerapkan pemilihan umum secara elektronik, namun di beberapa negara lain telah menerapkan pemilihan secara elektronik, negara – negara Eropa, Negara Jerman misalnya, dalam pemilu yang dilakukan, secara umum kegiatan ini memiliki prinsip yang sama (Spyros *et al*., 2002) seperti berikut ini:

- ***Generality***: Semua warga negara, kecuali dinyatakan sebaliknya dengan ajudikasi, di atas usia tertentu memiliki hak untuk memilih. Ini berarti bahwa setiap partisipasi dalam proses pemungutan suara selalu dapat dikonfirmasi.
- ***Freedom***: Setiap orang bebas memilih sesuai dengan keinginannya.
- ***Equality***: Semua suara dianggap setara. Proses pemungutan suara diatur sedemikian rupa sehingga mengamankan:

- ***Eligibility***: Hanya yang memiliki hak pilih yang dapat berpartisipasi dan memberikan suaranya.
- ***Un-reusability***: Setiap pemilih yang memenuhi syarat hanya dapat memilih satu kali.
- ***Un-changeability / Integrity***: Tidak ada yang bisa menduplikasi suara orang lain, atau mengubah suara orang lain.
- ***Verifiability***: Pemilih harus bisa memverifikasi bahwa suaranya masuk dan dihitung.
- ***Accessibility***: Pemilih harus memiliki akses tanpa pandang bulu ke infrastruktur pemilihan.

- ***Secrecy***: Bersifat rahasia. Registrasi, otentikasi, dan pemungutan suara dilakukan secara terpisah. Surat suara divalidasi secara independen dari pihak otentikasi pemilih.
- ***Directness***: Tidak ada pihak ketiga yang terlibat selama proses pemilihan dan setiap surat suara dicatat dan dihitung secara langsung.

Atribut tersebut berkaitan erat dengan kasus penggunaan bisnis untuk proses pemilihan umum di mana:

1. Mendata pemilih, proses ini sangat penting untuk menentukan pemilih dalam proses pemungutan suara saat ini. Secara umum, semua orang di atas usia tertentu memiliki hak / kewajiban untuk berpartisipasi dalam proses pemilihan.

2. Memberikan otentikasi, proses ini dilakukan untuk memungkinkan mereka mengidentifikasi diri mereka sendiri selama proses pemungutan suara. Tanggung jawab untuk penyediaan sarana otentikasi dapat berada di tangan negara. Di beberapa negara proses ini tidak dilakukan, karena pemilih dapat menggunakan kartu identitas atau paspor mereka untuk memilih.
3. Voting, proses ini dilakukan setelah otentikasi. Pemilih memberikan suaranya, dengan cara yang melindungi kerahasiaan, dan surat suara individu yang berwenang dicatat. Dan sesuai prinsip dari kesetaraan adalah bahwa pemilih tidak diperbolehkan memilih lagi untuk pemilihan itu.
4. Penghitungan suara, proses ini dilakukan untuk memvalidasi suara dan menentukan jumlah suara yang diterima oleh masing-masing pihak.
5. Verifikasi integritas hasil, Proses ini terjadi jika pemilih atau pihak lain yang berkepentingan, meminta untuk memverifikasi bahwa salah satu dari prosedur pemilihan tersebut telah dilakukan dengan benar.

2.8. Transaction Speed

Secara umum, total kecepatan transaksi dapat diukur dengan menghitung banyaknya jumlah transaksi yang dapat dikerjakan dalam satuan waktu. Namun dalam *blockchain* ini, seluruh kegiatan transaksi yang diproses akan dimasukkan ke dalam sebuah blok di mana blok ini merupakan hasil catatan dari seluruh kegiatan transaksi yang telah disetujui. Dalam implementasinya, setiap blok ini memiliki

beberapa faktor yang dapat mempengaruhi jumlah transaksi yang akan dimuat ke dalamnya. Faktor tersebut di antaranya adalah, blok akan dibuat dalam kurun waktu tertentu dan untuk setiap transaksi akan memiliki biayanya masing-masing, di mana dalam setiap blok memiliki nilai batasan maksimum dari total biaya transaksi secara kumulatif. Transaksi maksimum per detik (TPS) secara teori dapat dihitung dengan menggunakan persamaan seperti berikut:

$$TPS\ (Transaction\ per\ second) = \frac{GasLimit}{TxGas * BlockTime} \qquad \text{(iii)}$$

Di mana *GasLimit* adalah batas maksimum *gas* untuk setiap bloknya, *TxGas* adalah *gas* (biaya) yang diperlukan untuk melakukan transaksi dan *BlockTime* adalah waktu untuk setiap blok baru dibuat.