

Lesson 17

HARDHAT DAOS

Hikmah Nisya - 1103184094

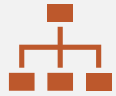
Radzis Araaf Jaya Jamaludin - 1103184234

Raudhatul Rafiqah Assyahiddini - 1103180225

Apa itu DAO ?

Decentralized Autonomous Organization (DAO) adalah sebuah organisasi yang dirancang untuk otomatis dan terdesentralisasi. Ini bertindak sebagai bentuk dana modal ventura, berdasarkan kode sumber terbuka dan tanpa struktur manajemen atau dewan direksi yang khas. Maka terjadinya membuat organisasi tersebut pengembang DAO percaya bahwa mereka dapat menghilangkan kesalahan manusia atau manipulasi dana investor dengan menempatkan kekuatan pengambilan keputusan ke tangan sistem otomatis dan proses crowdsourced.

DAO ini diluncurkan pada tahun 2016 bulan April akhir berkat adanya crowdsale token selama sebulan yang dapat mengumpulkan lebih dari \$150 juta dana.



Decentralized



Autonomous



Organizations

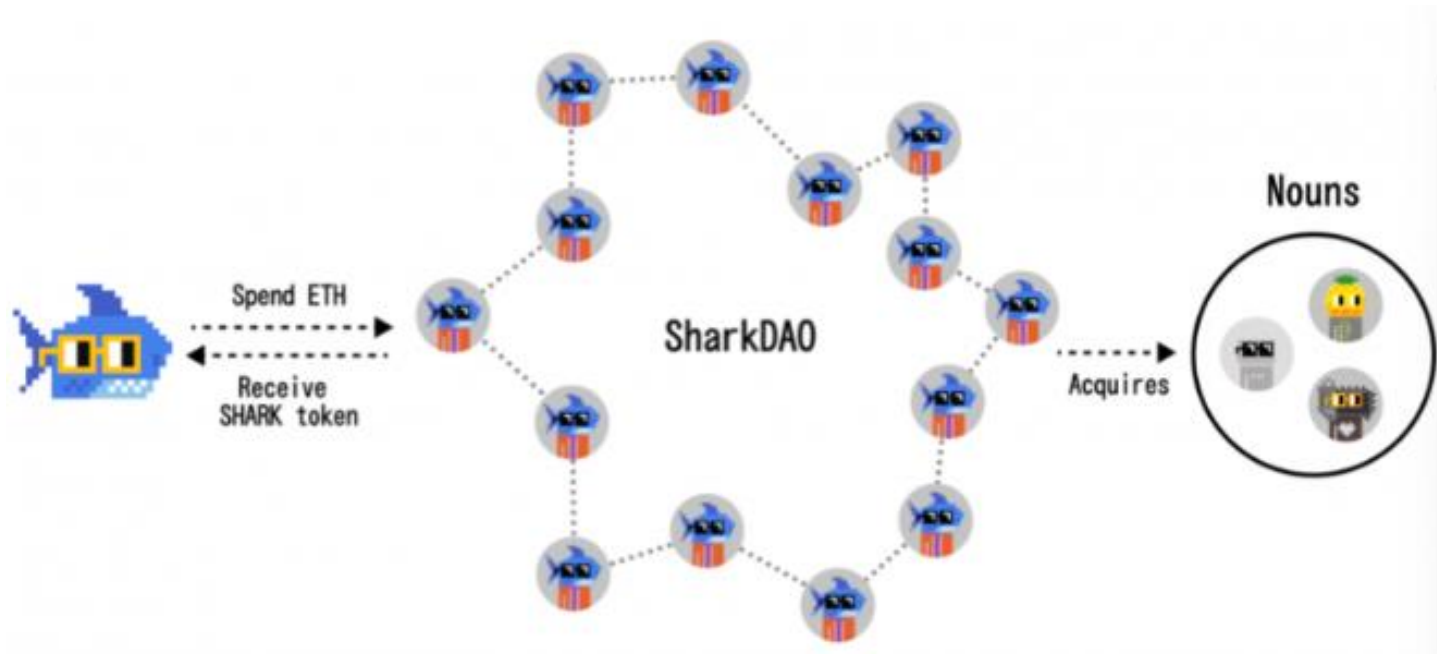
What is DAO?

Any group that is governed by a transparent set of rules found on a blockchain or smart contract.
And It's overloaded because some people say bitcoin is down because the miners can choose whether or not to upgrade their software. Other people think that Dows must use transparent smart contracts, which have the rules ingrained right into them.

DAO ini memiliki sebuah tim inti anggota komunitas melalui penggunaan smart contracts, smart contracts ini menjabarkan kerangka dasar yang digunakan DAO untuk beroperasi. Lalu mereka dapat diverifikasi dan dapat diaudit secara public sehingga setiap anggota potensial dapat sepenuhnya memahami bagaimana protocol berfungsi di setiap langkah.

Setelah aturan ini secara resmi ditulis ke dalam blockchain, langkah selanjutnya yaitu pedanaan dimana DAO perlu mencari cara untuk menerima dana dan bagaimana memberikan tata kelola . Ini biasanya dicapai melalui penerbitan token dimana protocol menjual token untuk mengumpulkan dana dan mengisi penbendaharaan DAO. Lalu sebagai imbalan fiat mereka, pemegang token diberikan hak suara tertentu, biasanya sebanding dengan kepemilikan mereka. Setelah pedanaan selesai DAO siap untuk diterapkan

Cara Kerja DAO



Setelah kode didorong ke dalam produksi, dan tidak dapat lagi diubah dengan cara selain consensus yang dicapai melalui pemungutan suara anggota dengan artian tidak ada otoritas khusus yang dapat mengubah aturan DAO sepenuhnya.

Cara Terlibat DAO?

Setelah Anda menemukan proyek yang menarik, ada beberapa cara berbeda untuk terlibat langsung. Saya merasa penting untuk dicatat bahwa tidak semua DAO beroperasi dengan tujuan yang sama sehingga langkah pertama adalah mencari tahu fungsi inti dari setiap DAO. Untuk DAO yang berfokus pada tata kelola teknis, penting untuk memahami jenis hak suara apa yang diberikan kepada pemegang token dan jenis proposal apa yang dipertaruhkan. Dalam beberapa kasus seperti Uniswap, pemegang token dapat memilih untuk mendistribusikan sebagian dari biaya yang dikumpulkan oleh protokol di antara mereka sendiri.

Dalam protokol lain seperti Compound, pemegang token dapat memberikan suara untuk mendistribusikan biaya protokol ini untuk perbaikan bug dan peningkatan sistem. Pendekatan ini juga memungkinkan pekerja lepas dan mereka yang umumnya tertarik dengan proyek untuk dapat bergabung secara ad hoc dan menerima kompensasi atas pekerjaan mereka melalui proyek yang didanai hibah DAO (DAO secara teratur memposting proyek ad hoc semacam ini di server Discord mereka)

DAO LANDSCAPE

Curated by @Cooopahtroopa • Pixels by Carlos/

DAO Operating Systems

ARAGON DaoStack DaoHaus
COLONY Syndicate Orca CW

Investment DAOs

MetaCartel theLAO Flamingo
Komorebi UdacityFund BitDAO
Free Company Duck DAO

Grants DAO

MetaCartel Ventures MolochDAO
Audius Grants Uniswap Grants
Mint Fund Sevens Foundation
Compound Grants Aave Grants

Collector DAOs

PleasrDAO Flamingo SquiggleDAO
FingerprintsDAO BRRDAO Whale
MUSE0 herstoryDAO BeetsDAO
JennyDAO MeebitsDAO Gremlins

Protocol DAOs

MAKER Compound UNISWAP AAVE Yearn Sushi
SYNTHETIX Curve Index Coop PieDAO tornado GITCOIN
LIDO pool KeeperDAO Badger RaribleDAO Olympus
hDAO Rari Capital Balancer Cream Finance AUDIUS
Universe Inverse finance Aavegotchi INSTADAPP OceanDAO

Service DAOs

DXdao PartyDAO
MetaFactory Fire Eyes DeepDAO
Reverie NeptuneDAO Llama
MetaverseDAO LexDAO dOrg
DaoHaus VitaDAO ShinyDAO
OPOLIS DoinGud DEEP WORK YGG
HoneyDAO Myco UniWhales
Yam DAO AladdinDAO

Media DAOs

FOREFRONT BanklessDAO GCR DarkstarDAO rekt

Social DAOs

FWB Seed Club
Radicale FiatLuxDAO
Metafam KrausHaus
ProsperDAO Bright Moments
Meta Gamma Delta
SongCamp CabinDAO
PROOF OF HUMANITY BAYC
TheWIPmeetup

DAO sering meminta komunitas untuk membangun ide-ide menarik melalui proyek-proyek yang didanai hibah, dan individu dengan pikiran kewirausahaan dapat dengan bebas mengajukan proposal untuk membantu memimpin pengembangan protokol di masa depan.

Tingkat partisipasi DAO bervariasi. Anda dapat memilih untuk menukar ke token tata kelola dan memperhatikan suara Snapshot; Anda dapat bergabung dengan DAO's Discord dan mengambil proyek aktual di mana Anda mendapat kompensasi atas kontribusi Anda; Anda bahkan dapat berinvestasi ke DAO yang menarik dengan berjejaring di konferensi. Anda memilih seberapa terlibat Anda ingin menjadi.

Tools:

snapshot : UI for off-chain voting / sentiment analysis

Tally : UI for on-chain voting

Gnosis Safe : Multi-sig

Openzeppelin : DAO code tools

Zodiac : More DAO code tools

Openzeppelin Defender : A tool to propose governance and other contract functions.

```

1 // contracts/Box.sol
2 // SPDX-License-Identifier: MIT
3 pragma solidity ^0.8.0;
4
5 import "@openzeppelin/contracts/access/Ownable.sol";
6
7 contract Box is Ownable {
8     uint256 private value;
9
10    // Emitted when the stored value changes
11    event ValueChanged(uint256 newValue);
12
13    // Stores a new value in the contract
14    function store(uint256 newValue) public onlyOwner {
15        value = newValue;
16        emit ValueChanged(newValue);
17    }
18
19    // Reads the last stored value
20    function retrieve() public view returns (uint256) {
21        return value;
22    }
23 }

```

Box.sol

```

1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.9;
3
4 import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Votes.sol";
5
6 contract GovernanceToken is ERC20Votes {
7     uint256 public s_maxSupply = 1000000000000000000000000;
8
9     constructor() ERC20("GovernanceToken", "GT") ERC20Permit("GovernanceToken") {
10         _mint(msg.sender, s_maxSupply);
11     }
12
13    // The functions below are overrides required by Solidity.
14
15    function _afterTokenTransfer(
16        address from,
17        address to,
18        uint256 amount
19    ) internal override(ERC20Votes) {
20        super._afterTokenTransfer(from, to, amount);
21    }
22
23    function _mint(address to, uint256 amount) internal override(ERC20Votes) {
24        super._mint(to, amount);
25    }
26
27    function _burn(address account, uint256 amount) internal override(ERC20Votes) {
28        super._burn(account, amount);
29    }
30 }

```

GovernanceToken.sol

```

1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.9;
3
4 import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
5 import "@openzeppelin/contracts/token/ERC20/extensions/draft-ERC20Permit.sol";
6 import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Votes.sol";
7 import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Wrapper.sol";
8
9 contract MyToken is ERC20, ERC20Permit, ERC20Votes, ERC20Wrapper {
10     constructor(IERC20 wrappedToken)
11         ERC20("MyToken", "MTK")
12         ERC20Permit("MyToken")
13         ERC20Wrapper(wrappedToken)
14     {}
15
16     // The functions below are overrides required by Solidity.
17
18     function _afterTokenTransfer(
19         address from,
20         address to,
21         uint256 amount
22     ) internal override(ERC20, ERC20Votes) {
23         super._afterTokenTransfer(from, to, amount);
24     }
25
26     function _mint(address to, uint256 amount) internal override(ERC20, ERC20Votes) {
27         super._mint(to, amount);
28     }
29
30     function _burn(address account, uint256 amount) internal override(ERC20, ERC20Votes) {
31         super._burn(account, amount);
32     }
33
34     function decimals() public view override(ERC20, ERC20Wrapper) returns (uint8) {
35         return super.decimals();
36     }
37 }

```

GovernanceTokenWrapper.sol

```

1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.9;
3
4 import "@openzeppelin/contracts/governance/Governor.sol";
5 import "@openzeppelin/contracts/governance/extensions/GovernorCountingSimple.sol";
6 import "@openzeppelin/contracts/governance/extensions/GovernorVotes.sol";
7 import "@openzeppelin/contracts/governance/extensions/GovernorVotesQuorumFraction.sol";
8 import "@openzeppelin/contracts/governance/extensions/GovernorTimelockControl.sol";
9 import "@openzeppelin/contracts/governance/extensions/GovernorSettings.sol";
10
11 contract GovernorContract is
12     Governor,
13     GovernorSettings,
14     GovernorCountingSimple,
15     GovernorVotes,
16     GovernorVotesQuorumFraction,
17     GovernorTimelockControl
18 {
19     constructor(
20         IVotes _token,
21         TimelockController _timelock,
22         uint256 _quorumPercentage,
23         uint256 _votingPeriod,
24         uint256 _votingDelay
25     )
26         Governor("GovernorContract")
27         GovernorSettings(
28             _votingDelay, /* 1 block */ // voting delay
29             _votingPeriod, // 45818, /* 1 week */ // voting period
30             0 // proposal threshold
31         )
32         GovernorVotes(_token)
33         GovernorVotesQuorumFraction(_quorumPercentage)
34         GovernorTimelockControl(_timelock)
35     {}
36
37     function votingDelay()
38         public
39         view
40         override(IGovernor, GovernorSettings)
41         returns (uint256)
42     {
43         return super.votingDelay();
44     }
45 }

```

GovernorContract.sol