

LESSON 17 : HARDHAT DAOS

Voting Contract

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS E:\Blockchain> mkdir hardhat-dao-fcc

Directory: E:\Blockchain

Mode                LastWriteTime         Length Name
----                -
d-----          7/6/2022   9:34 PM             hardhat-dao-fcc

PS E:\Blockchain> cd hardhat-dao-fcc\
PS E:\Blockchain\hardhat-dao-fcc> code .
```

PERTAMA BUAT FOLDER HARDHAT-DAO-FCC DAN MASUK KE DALAMANYA

```
PS E:\Blockchain\hardhat-dao-fcc> git --version  
git version 2.37.0.windows.1
```

```
PS E:\Blockchain\hardhat-dao-fcc> node --version  
v16.9.1
```

```
PS E:\Blockchain\hardhat-dao-fcc> yarn --version  
1.22.19
```

CEK VERSI YANG KITA PUNYA DARI
MULAI GIT, NODE, DAN YARN

KITA TAMBAHKAN HARDHAT KE YARN

```
PS E:\Blockchain\hardhat-dao-fcc> yarn add --dev hardhat
yarn add v1.22.19
info No lockfile found.
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
success Saved lockfile.
success Saved 207 new dependencies.
```

BUAT B
BERNA

```
hardhat-dao-fcc > contracts > Box.sol
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) {
```

```
PS E:\Blockchain\hardhat-dao-fcc> yarn add --dev @openzeppelin/contracts
yarn add v1.22.19
warning package.json: No license field
warning No license field
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...

warning No license field
success Saved 1 new dependency.
info Direct dependencies
└─ @openzeppelin/contracts@4.7.0
info All dependencies
└─ @openzeppelin/contracts@4.7.0
```

KITA TAMBAHKAN OPEN ZEPPELIN CONTRACTS

```
PS E:\Blockchain\hardhat-dao-fcc> yarn hardhat compile
yarn run v1.22.19
warning package.json: No license field
$ E:\Blockchain\hardhat-dao-fcc\node_modules\.bin\hardhat compile
888 888 888 888 888
888 888 888 888 888
888 888 888 888 888
8888888888 888b. 888d888 .d88888 88888b. 8888b. 888888
888 888 "88b 888p" d88" 888 888 "88b "88b 888
888 888 .d888888 888 888 888 888 .d888888 888
888 888 888 888 888 Y88b 888 888 888 888 Y88b.
888 888 "Y888888 888 "Y88888 888 888 "Y888888 "Y888

Welcome to Hardhat v2.9.9

✓ What do you want to do? · Create a basic sample project
✓ Hardhat project root: · E:\Blockchain\hardhat-dao-fcc
✓ Do you want to add a .gitignore? (Y/n) · y
✓ Help us improve Hardhat with anonymous crash reports & basic usage data? (Y/n) · true

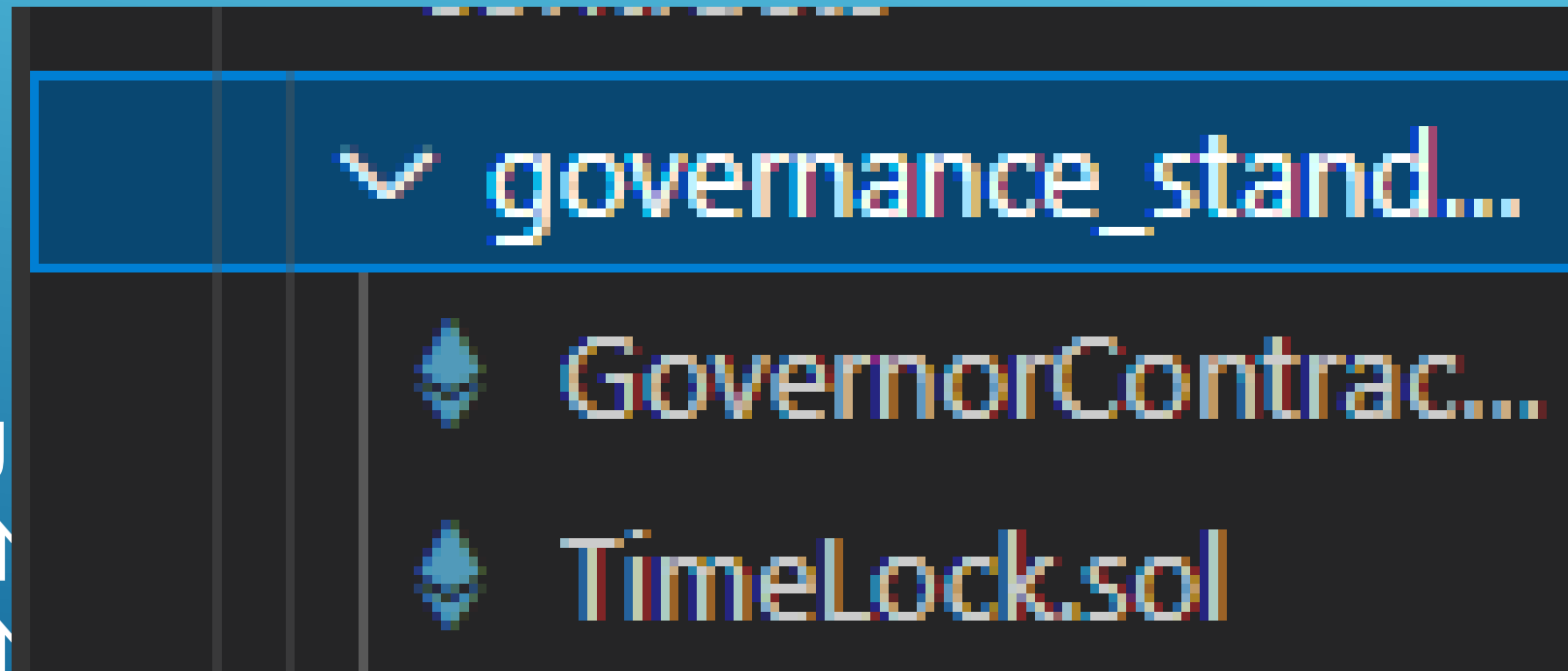
You need to install these dependencies to run the sample project:
  yarn add --dev "hardhat@^2.9.9" "@nomiclabs/hardhat-waffle@^2.0.0" "ethereum-waffle@^3.0.0" "chai@^4.2.0" "@nomiclabs/hardhat-ethers@^2.0.0" "ethers@^5.0.0"

Project created
See the README.md file for some example tasks you can run.
Done in 65.59s.
```

LALU KITA COMPILE HARDHAT TERSEBUT

```
GovernanceToken.sol X
hardhat-dao-fcc > contracts > GovernanceToken.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) {
```


LALU BU
GOVERN
GOVERN
TIMELOCK.SOK



```
GovernorContract.sol X
hardhat-dao-fcc > contracts > governance_standard > GovernorContract.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
10 contract GovernorContract is
11     Governor,
12     GovernorCountingSimple,
13     GovernorVotes,
14     GovernorVotesQuorumFraction,
15     GovernorTimelockControl
16 {
17     uint256 public s_votingDelay;
18     uint256 public s_votingPeriod;
19
20     constructor(
21         ERC20Votes _token,
22         TimelockController _timelock,
23         uint256 _quorumPercentage,
24         uint256 _votingPeriod,
25         uint256 _votingDelay
26     )
27         Governor("GovernorContract")
28         GovernorVotes(_token)
29         GovernorVotesQuorumFraction(_quorumPercentage)
```

GOVERNANCECONTRACT.SOL BERISI
HAMPIR SEMUA LOGIKA UNTUK SISTEM
VOTING INI

```
TimeLock.sol X
hardhat-dao-fcc > contracts > governance_standard > TimeLock.sol
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 import "@openzeppelin/contracts/governance/TimeLockController.sol";
5
6 contract TimeLock is TimeLockController {
7     // minDelay is how long you have to wait before executing
8     // proposers is the list of addresses that can propose
9     // executors is the list of addresses that can execute
10    constructor(
11        uint256 minDelay,
12        address[] memory proposers,
13        address[] memory executors
14    ) TimeLockController(minDelay, proposers, executors) {}
15 }
16
```

TIME LOCK BERISI TENTANG ADDITIONAL
CONTRACT SI OWERNYA

```

PS E:\Blockchain\hardhat-dao-fcc> yarn add --dev @nomiclabs/hardhat-ethers ethers
yarn add v1.22.19
warning package.json: No license field
warning No license field
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Saved lockfile.
warning No license field
success Saved 10 new dependencies.
info Direct dependencies
├─ @nomiclabs/hardhat-ethers@0.3.0-beta.13
└─ ethers@5.6.9
info All dependencies
├─ @ethersproject/contracts@5.6.2
├─ @ethersproject/json-wallets@5.6.1
├─ @ethersproject/providers@5.6.8
├─ @ethersproject/solidity@5.6.1
├─ @ethersproject/units@5.6.1
├─ @ethersproject/wallet@5.6.2
├─ @nomiclabs/hardhat-ethers@0.3.0-beta.13
├─ aes-js@3.0.0
├─ bech32@1.1.4
└─ ethers@5.6.9
Done in 13.18s.

```

```

Done in 13.18s.
> yarn add --dev hardhat-deploy
yarn add v1.22.19
warning package.json: No license field
warning No license field
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
warning "hardhat-deploy > zksync-web3@0.7.9" has incorrect peer dependency "ethers@~5.5.0".
[4/4] Building fresh packages...
success Saved lockfile.
warning No license field
success Saved 15 new dependencies.
info Direct dependencies
├─ hardhat-deploy@0.11.11
info All dependencies
├─ @types/qs@6.9.7
├─ async@3.2.4
├─ axios@0.21.4
├─ combined-stream@1.0.8
├─ delayed-stream@1.0.0
├─ encode-utf8@1.0.3
├─ fmix@0.1.0
├─ form-data@4.0.0
├─ fs-extra@10.1.0
├─ hardhat-deploy@0.11.11
├─ match-all@1.2.6
├─ mime-db@1.52.0
├─ mime-types@2.1.35
├─ murmur-128@0.2.1
└─ zksync-web3@0.7.9
Done in 8.08s.
PS E:\Blockchain\hardhat-dao-fcc>

```

KITA INSTALL YARN ADD --DEV TYPESCRIPT TYPECHAIN

TS
@
UN

```
PS E:\Blockchain\hardhat-dao-fcc> yarn add --dev typescript typechain ts-nod
e @typechain/ethers-v5 @typechain/hardhat @types/chai @types/node
yarn add v1.22.19
warning package.json: No license field
warning No license field
[1/4] Resolving packages...
[2/4] Fetching packages...
warning Pattern ["@types/node@^18.0.3"] is trying to unpack in the same dest
ination "C:\Users\Fath\AppData\Local\Yarn\Cache\lv6\npm-@types-node-1
8.0.3-463fc47f13ec0688a33aec75d078a0541a447199-integrity\node_modules\@typ
es\node" as pattern ["@types/node@*", "@types/node@*", "@types/node@*", "@type
s/node@*", "@types/node@*"]. This could result in non-deterministic behavior,
skipping.
[3/4] Linking dependencies...
warning "hardhat-deploy > zkSync-web3@0.7.9" has incorrect peer dependency "
ethers@~5.5.0".
warning " > @typechain/ethers-v5@10.1.0" has unmet peer dependency "@ethersp
roject/abi@^5.0.0".
warning " > @typechain/ethers-v5@10.1.0" has unmet peer dependency "@ethersp
roject/bytes@^5.0.0".
warning " > @typechain/ethers-v5@10.1.0" has unmet peer dependency "@ethersp
roject/providers@^5.0.0".
warning " > @typechain/hardhat@6.1.2" has unmet peer dependency "@ethersproj
ect/abi@^5.4.7".
warning " > @typechain/hardhat@6.1.2" has unmet peer dependency "@ethersproj
ect/providers@^5.4.7".
[4/4] Building fresh packages...
success Saved lockfile.
warning No license field
success Saved 38 new dependencies.
info Direct dependencies
├─ @typechain/ethers-v5@10.1.0
└─ @typechain/hardhat@6.1.2
```

@TYPES/NODE

B
FILL

▼ deploy	
TS	01-deploy-govern...
TS	02-deploy-time-lo...
TS	03-deploy-govern...
TS	04-setup-governan...
TS	05-deploy-box.ts

01-deploy-governor-token.ts
02-deploy-time-lock.ts
03-deploy-governor-
contract.ts
04-setup-governance-
contract.ts
05-deploy-box.ts

DENEGATE \$ 5

```
TS 01-deploy-governor-token.ts X
hardhat-dao-fcc > deploy > TS 01-deploy-governor-token.ts > ...
1 import { HardhatRuntimeEnvironment } from "hardhat/types"
2 import { DeployFunction } from "hardhat-deploy/types"
3 import verify from "../helper-functions"
4 import { networkConfig, developmentChains } from "../helper-hardhat-config"
5 import { ethers } from "hardhat"
6
7 const deployGovernanceToken: DeployFunction = async function (hre: HardhatRuntimeEnvironment) {
8   // @ts-ignore
9   const { getNamedAccounts, deployments, network } = hre
10   const { deploy, log } = deployments
11   const { deployer } = await getNamedAccounts()
12   log("-----")
13   log("Deploying GovernanceToken and waiting for confirmations...")
14   const governanceToken = await deploy("GovernanceToken", {
15     from: deployer,
16     args: [],
17     log: true,
18     // we need to wait if on a live network so we can verify properly
19     waitConfirmations: networkConfig[network.name].blockConfirmations || 1,
20   })
21   log(`GovernanceToken at ${governanceToken.address}`)
22   if (!developmentChains.includes(network.name) && process.env.ETHERSCAN_API_KEY) {
23     await verify(governanceToken.address, [])
24   }
25   log(`Delegating to ${deployer}`)
26   await delegate(governanceToken.address, deployer)
27   log("Delegated!")
```

FILE 01-DEPLOY-GOVERNOR-TOKEN.TS UNTUK MENDEPLOY TOKEN, MEMBERI TIAP AKUN PRIVATE KEY

```
TS 02-deploy-time-lock.ts X
hardhat-dao-fcc > deploy > TS 02-deploy-time-lock.ts > ...
1 import { HardhatRuntimeEnvironment } from "hardhat/types"
2 import { DeployFunction } from "hardhat-deploy/types"
3 import verify from "../helper-functions"
4 import { networkConfig, developmentChains, MIN_DELAY } from "../helper-hardhat-config"
5
6 const deployTimeLock: DeployFunction = async function (hre: HardhatRuntimeEnvironment) {
7   // @ts-ignore
8   const { getNamedAccounts, deployments, network } = hre
9   const { deploy, log } = deployments
10  const { deployer } = await getNamedAccounts()
11  log("-----")
12  log("Deploying TimeLock and waiting for confirmations...")
13  const timeLock = await deploy("TimeLock", {
14    from: deployer,
15    args: [MIN_DELAY, [], []],
16    log: true,
17    // we need to wait if on a live network so we can verify properly
18    waitConfirmations: networkConfig[network.name].blockConfirmations || 1,
19  })
20  log(`TimeLock at ${timeLock.address}`)
21  if (!developmentChains.includes(network.name) && process.env.ETHERSCAN_API_KEY) {
22    await verify(timeLock.address, [])
23  }
24 }
25
26 export default deployTimeLock
27 deployTimeLock.tags = ["all", "timelock"]
```

FILE 02-DEPLOY-TIME-LOCK.TS
UNTUK MENDEPLOY TIME LOCKS,
ADDITIONAL CONTRACT OWNER


```
TS 03-deploy-governor-contract.ts X
hardhat-dao-fcc > deploy > TS 03-deploy-governor-contract.ts > ...
1  import { HardhatRuntimeEnvironment } from "hardhat/types"
2  import { DeployFunction } from "hardhat-deploy/types"
3  import verify from "../helper-functions"
4  import {
5    networkConfig,
6    developmentChains,
7    QUORUM_PERCENTAGE,
8    VOTING_PERIOD,
9    VOTING_DELAY,
10 } from "../helper-hardhat-config"
11
12 const deployGovernorContract: DeployFunction = async function (hre: HardhatRuntimeEnvironment) {
13   // @ts-ignore
14   const { getNamedAccounts, deployments, network } = hre
15   const { deploy, log, get } = deployments
16   const { deployer } = await getNamedAccounts()
17   const governanceToken = await get("GovernanceToken")
18   const timeLock = await get("TimeLock")
19
20   log("-----")
21   log("Deploying GovernorContract and waiting for confirmations...")
22   const governorContract = await deploy("GovernorContract", {
23     from: deployer,
24     args: [
25       governanceToken.address,
26       timeLock.address,
27       QUORUM_PERCENTAGE,
```

FILE 03-DEPLOY-GOVERNOR-
CONTRACT.TS UNTUK MENDEPLOY
SEMUA LOGIC DALAM VOTING.

```
TS 04-setup-governance-contracts.ts X
hardhat-dao-fcc > deploy > TS 04-setup-governance-contracts.ts > ...
1  import { HardhatRuntimeEnvironment } from "hardhat/types"
2  import { DeployFunction } from "hardhat-deploy/types"
3  import verify from "../helper-functions"
4  import { networkConfig, developmentChains, ADDRESS_ZERO } from "../helper-hardhat-config"
5  import { ethers } from "hardhat"
6
7  const setupContracts: DeployFunction = async function (hre: HardhatRuntimeEnvironment) {
8    // @ts-ignore
9    const { getNamedAccounts, deployments, network } = hre
10   const { log } = deployments
11   const { deployer } = await getNamedAccounts()
12   const governanceToken = await ethers.getContract("GovernanceToken", deployer)
13   const timeLock = await ethers.getContract("TimeLock", deployer)
14   const governor = await ethers.getContract("GovernorContract", deployer)
15
16   log("-----")
17   log("Setting up contracts for roles...")
18   // would be great to use multicall here...
19   const proposerRole = await timeLock.PROPOSER_ROLE()
20   const executorRole = await timeLock.EXECUTOR_ROLE()
21   const adminRole = await timeLock.TIMELOCK_ADMIN_ROLE()
22
23   const proposerTx = await timeLock.grantRole(proposerRole, governor.address)
24   await proposerTx.wait(1)
25   const executorTx = await timeLock.grantRole(executorRole, ADDRESS_ZERO)
26   await executorTx.wait(1)
27   const revokeTx = await timeLock.revokeRole(adminRole, deployer)
```

FILE 04-SETUP-GOVERNANCE-
CONTRACTS.TS UNTUK MENGATUR
CARA KERJA VOTING YANG
NANTI AKAN BERJALAN

```
TS 05-deploy-box.ts X
hardhat-dao-fcc > deploy > TS 05-deploy-box.ts > ...
1 import { HardhatRuntimeEnvironment } from "hardhat/types"
2 import { DeployFunction } from "hardhat-deploy/types"
3 import verify from "../helper-functions"
4 import { networkConfig, developmentChains } from "../helper-hardhat-config"
5 import { ethers } from "hardhat"
6
7 const deployBox: DeployFunction = async function (hre: HardhatRuntimeEnvironment) {
8   // @ts-ignore
9   const { getNamedAccounts, deployments, network } = hre
10  const { deploy, log } = deployments
11  const { deployer } = await getNamedAccounts()
12  log("-----")
13  log("Deploying Box and waiting for confirmations...")
14  const box = await deploy("Box", {
15    from: deployer,
16    args: [],
17    log: true,
18    // we need to wait if on a live network so we can verify properly
19    waitConfirmations: networkConfig[network.name].blockConfirmations || 1,
20  })
21  log(`Box at ${box.address}`)
22  if (!developmentChains.includes(network.name) && process.env.ETHERSCAN_API_KEY) {
23    await verify(box.address, [])
24  }
25  const boxContract = await ethers.getContractAt("Box", box.address)
26  const timeLock = await ethers.getContract("TimeLock")
27  const transferTx = await boxContract.transferOwnership(timeLock.address)
```

FILE 05-DEPLOY-BOX.TS UNTUK
MENDEPLOY SMARTCONTRACT
YANG SUDAH KITA BUAT DIAWAL

```
PS E:\Blockchain\hardhat-dao-fcc> yarn hardhat deploy
yarn run v1.22.17
warning package.json: No license field
$ E:\Blockchain\hardhat-dao-fcc\node_modules\.bin\hardhat deploy
Nothing to compile
No need to generate any newer typings.
Deploying Governance Token...
deploying "GovernanceToken" (tx: 0x6755c235273eae4735db03594be3c08fbe59f167b3ae4be2d1112a9973388561)...: de
ployed at 0x5FbDB2315678afecb367f032d93F642f64180aa3 with 3376648 gas
Deployed governance token to address 0x5FbDB2315678afecb367f032d93F642f64180aa3
Checkpoints 1
Delegated!
Deploying Timelock...
deploying "TimeLock" (tx: 0x8230faa079de6656f3ae1b4199c107d9ddd7e0623823701fcac09dc05d634370)...: deployed
at 0x9fE46736679d2D9a65F0992F2272dE9f3c7fa6e0 with 2684060 gas
Deploying governor
deploying "GovernorContract" (tx: 0x6bd36bd7451cb0e8099364cb9165b7d659695efba040718452de48599a7e48d1)...: d
eployed at 0xCf7Ed3AccA5a467e9e704C703E8D87F634fB0Fc9 with 4690056 gas
Setting up roles...
Deploying Box...
deploying "Box" (tx: 0xfbbc0cce71d18a6e32d4e789bd30caeab46cebe24ba7d50924afd8a870c2429e)...: deployed at 0x
a513E6E4b8f2a923D98304ec87F64353C4D5C853 with 445808 gas
YOU DUN IT!!!
```

LALU KITA DEPLOY DENGAN YARN
HARDHAT DEPLOY, MAKA AKAN KELUAR
SEPERTI INI