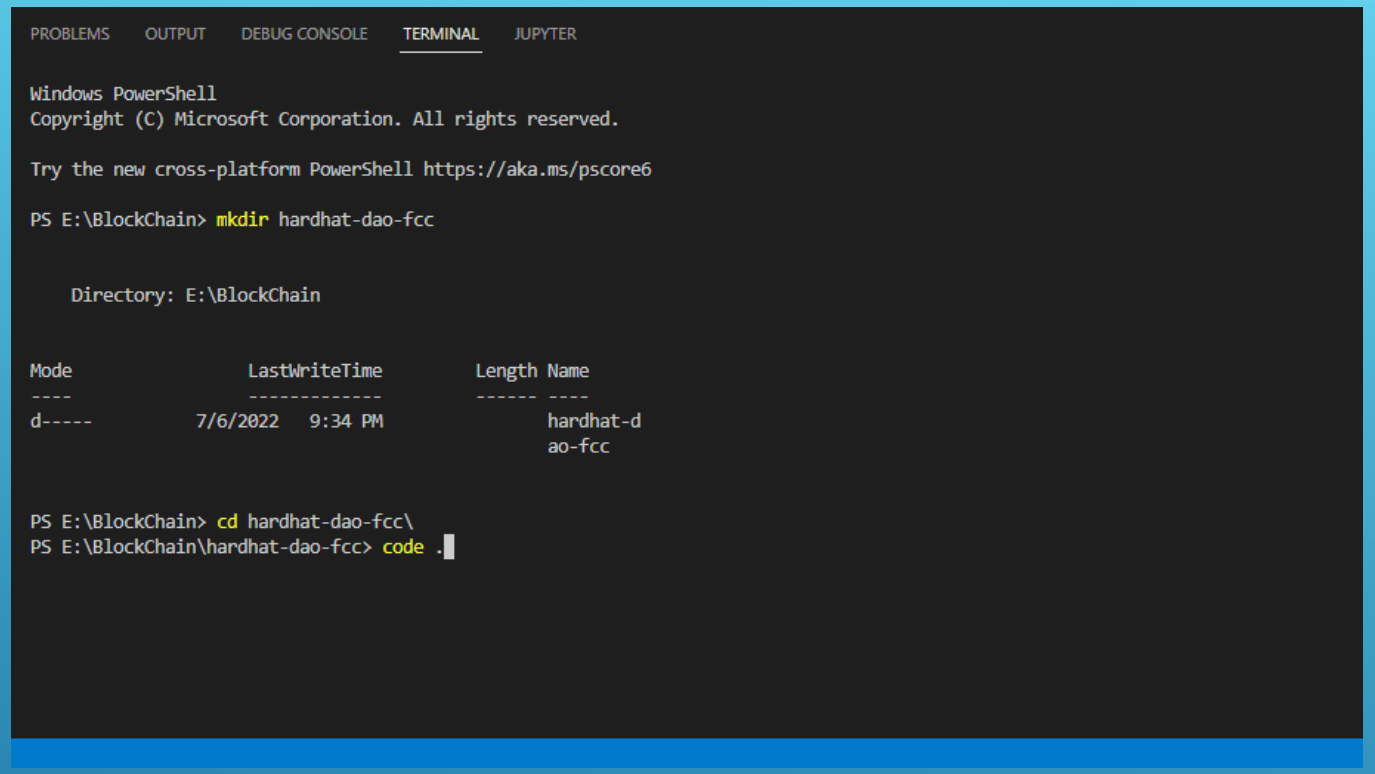
LESSON 17 : HARDHAT DAOS

Voting Contract



PERTAMA BUAT FLODER 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.
```

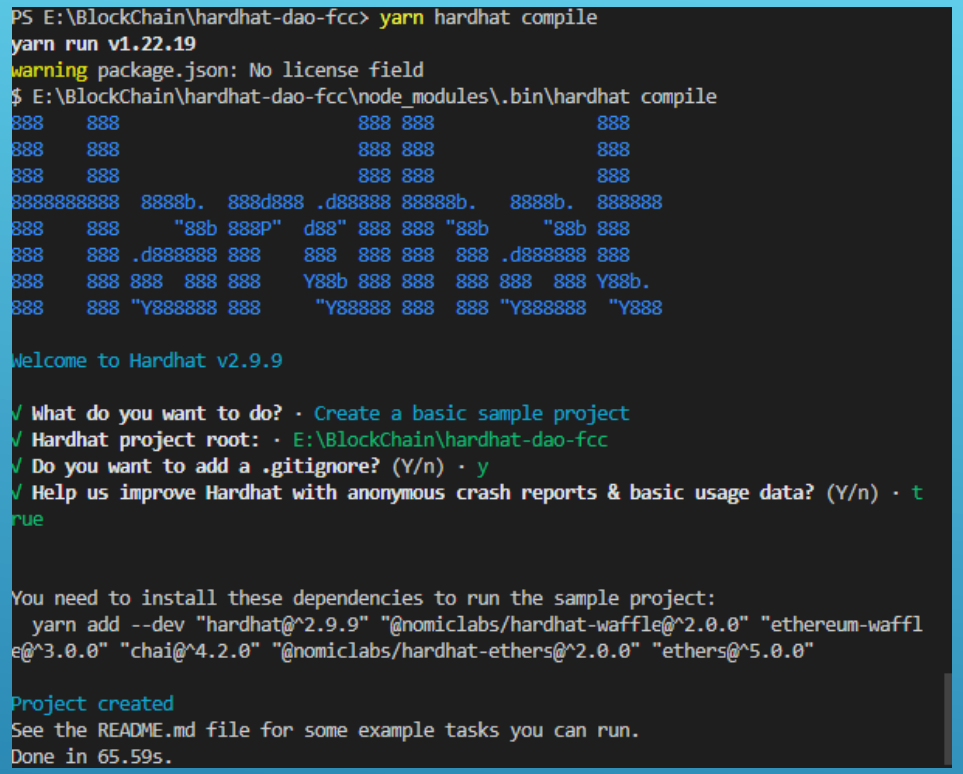
BUAT B
BERNA

hardhat-dao-fcc > contracts > Box.sol

```sol
// contracts/Box.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/access/Ownable.sol";

contract Box is Ownable {
  uint256 private value;

  // Emitted when the stored value changes
  event ValueChanged(uint256 newValue);

  // Stores a new value in the contract
  function store(uint256 newValue) public onlyOwner {
    value = newValue;
    emit ValueChanged(newValue);
  }

  // Reads the last stored value
  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



LALU KITA COMPILE HARDHAT TERSEBUT

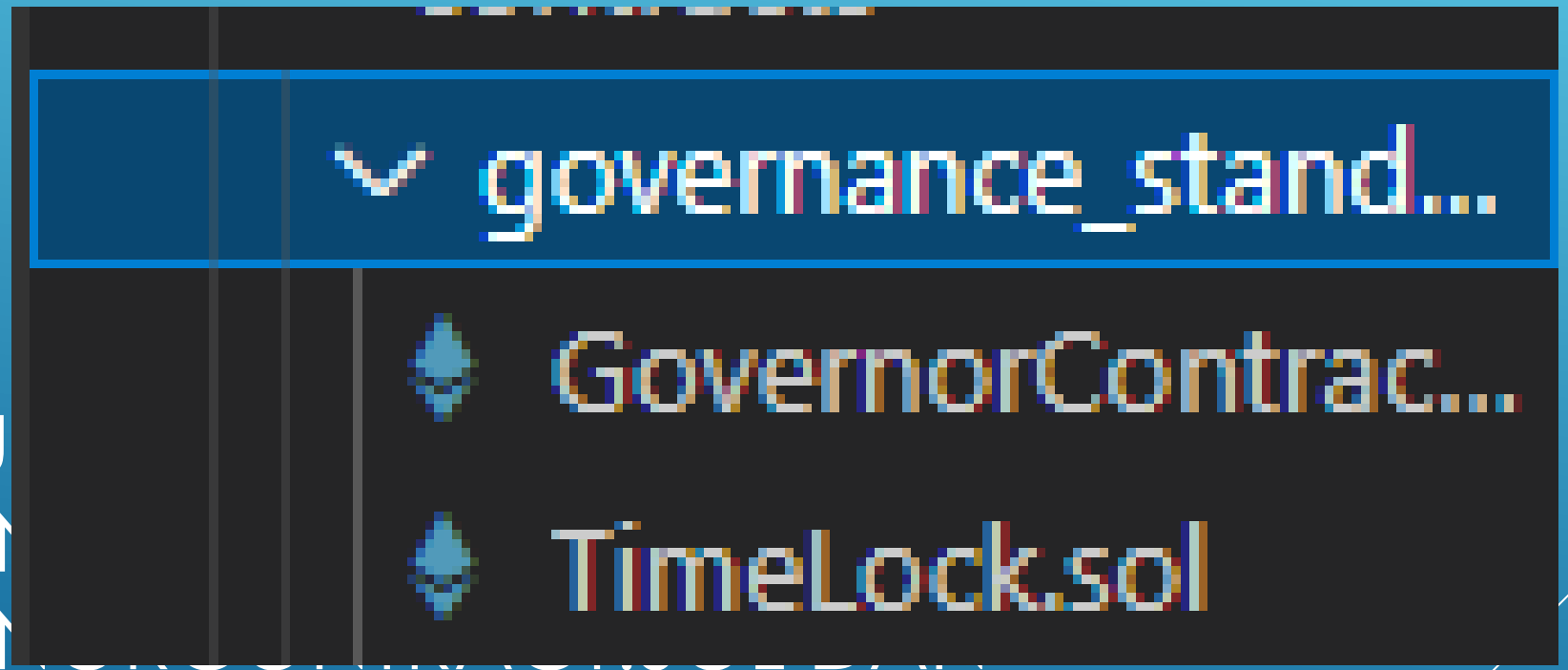
BU
GO
TO
TO

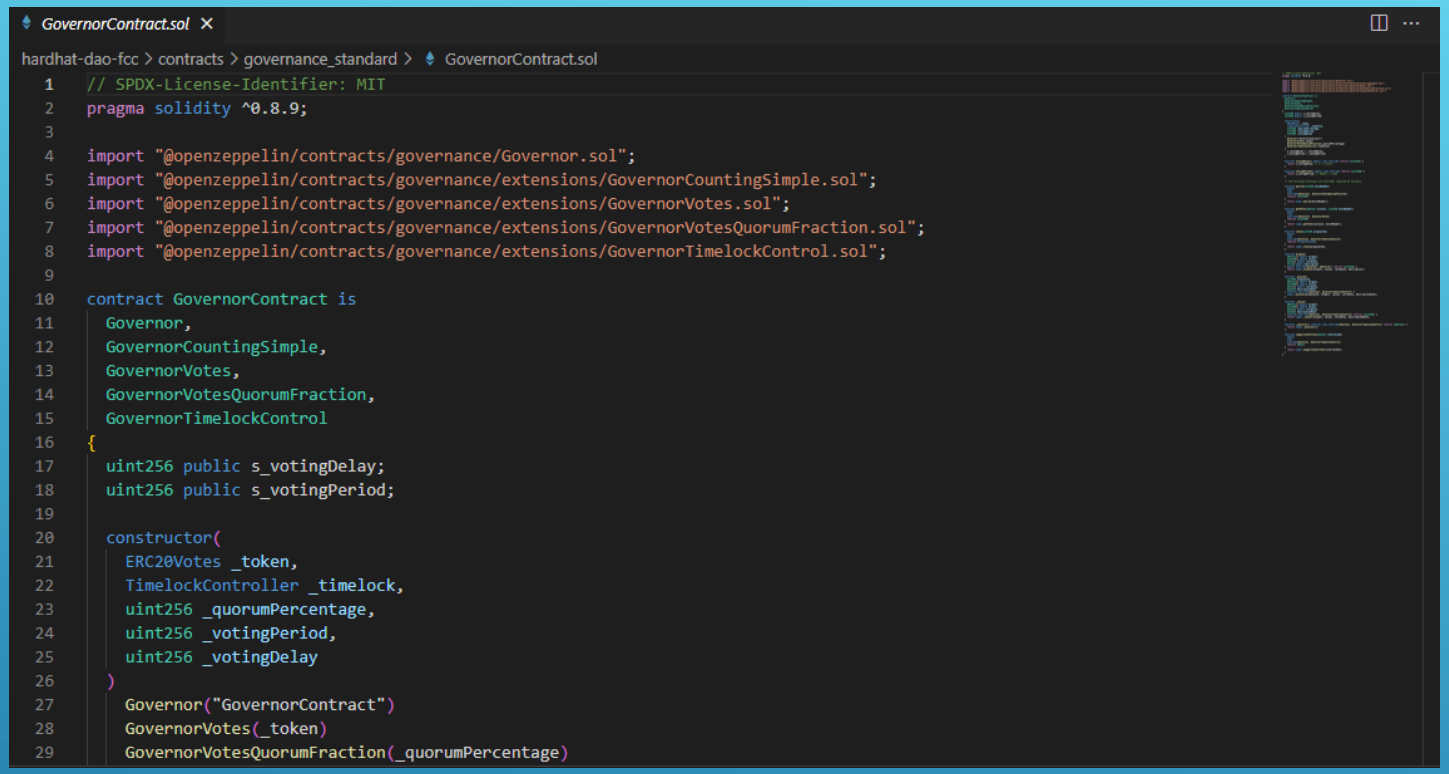
GovernanceToken.sol

hardhat-dao-fcc > contracts > GovernanceToken.sol

```sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.9;

import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Votes.sol";

contract GovernanceToken is ERC20Votes {
  uint256 public s_maxSupply = 1000000000000000000000000;

  constructor() ERC20("GovernanceToken", "GT") ERC20Permit("GovernanceToken") {
    _mint(msg.sender, s_maxSupply);
  }

  // The functions below are overrides required by Solidity.

  function _afterTokenTransfer(
    address from,
    address to,
    uint256 amount
  ) internal override(ERC20Votes) {
    super._afterTokenTransfer(from, to, amount);
  }

  function _mint(address to, uint256 amount) internal override(ERC20Votes) {
    super._mint(to, amount);
  }

  function _burn(address account, uint256 amount) internal override(ERC20Votes) {
```

LALU BU
GOVERN
GOVERN
TIMELOCK.SOK





GOVERNANCECONTRACT.SOL BERISI HAMPIR SEMUA LOGIKA UNTUK SISTEM VOTING INI

TimeLock.sol

hardhat-dao-fcc > contracts > governance_standard > TimeLock.sol

```sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/governance/TimelockController.sol";

contract TimeLock is TimelockController {
  // minDelay is how long you have to wait before executing
  // proposers is the list of addresses that can propose
  // executors is the list of addresses that can execute
  constructor(
    uint256 minDelay,
    address[] memory proposers,
    address[] memory executors
  ) TimelockController(minDelay, proposers, executors) {}
}

```

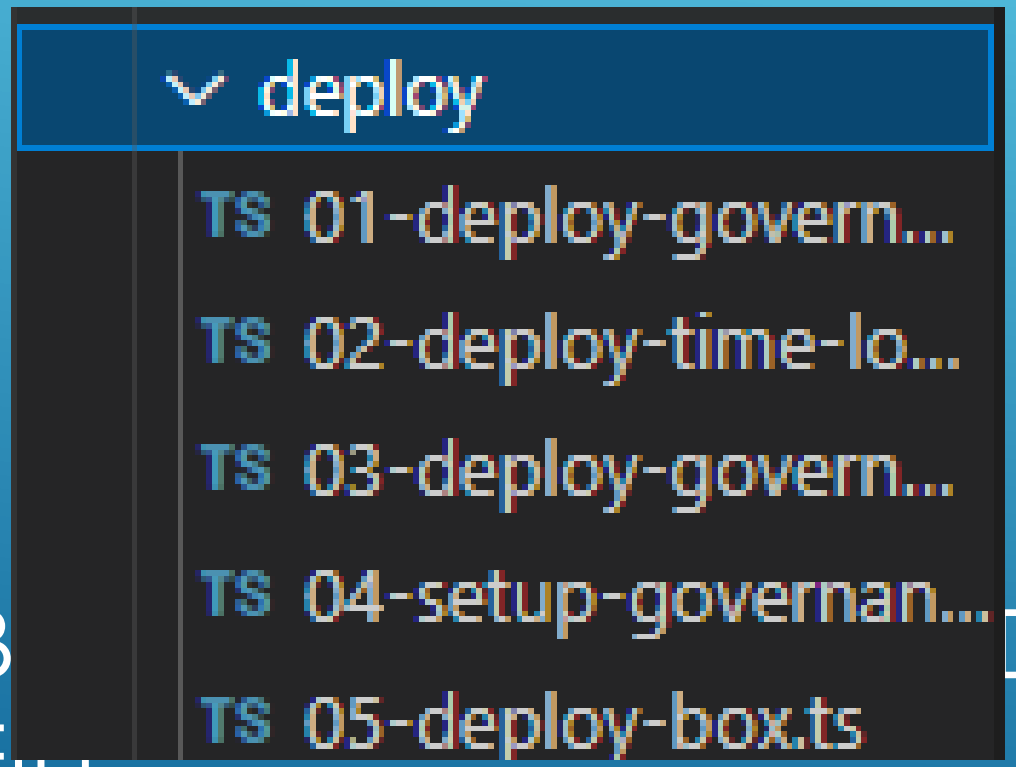
TIME LOCK BERISI TENTANG ADDITIONAL CONTRACT SI OWNERNYA

```
PS E:\BlockChain\hardhat-dao-fcc> yarn add --dev @nomiclabs/hardh
at-ethers@npm:hardhat-deploy-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
├─ asynckit@0.4.0
├─ 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>
```

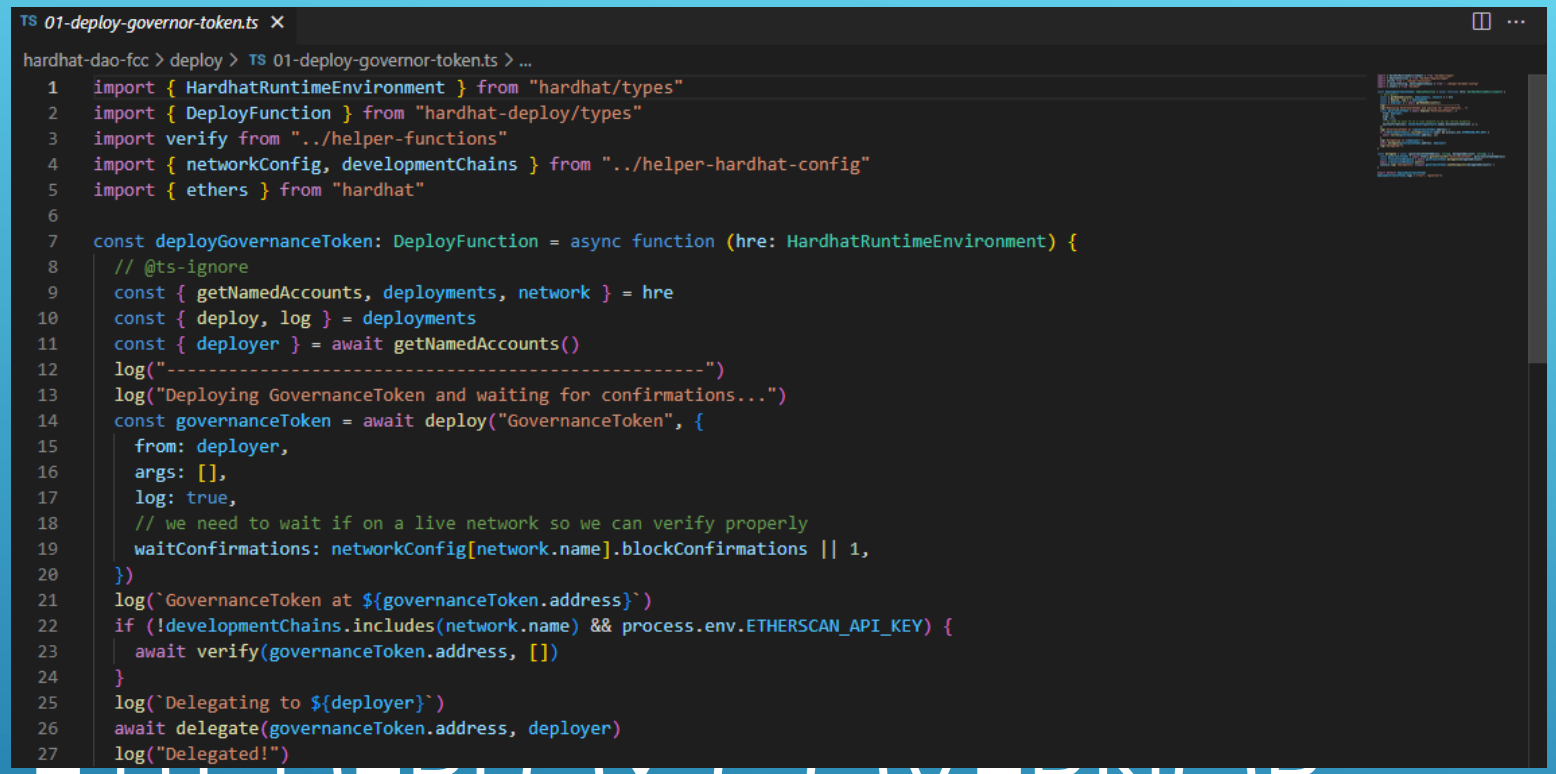
KITA INSTALL YARN ADD --DEV TYPESCRIPT TYPECHAIN TS- @- @TYPES/NODE UN

```
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\\v6\\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
```

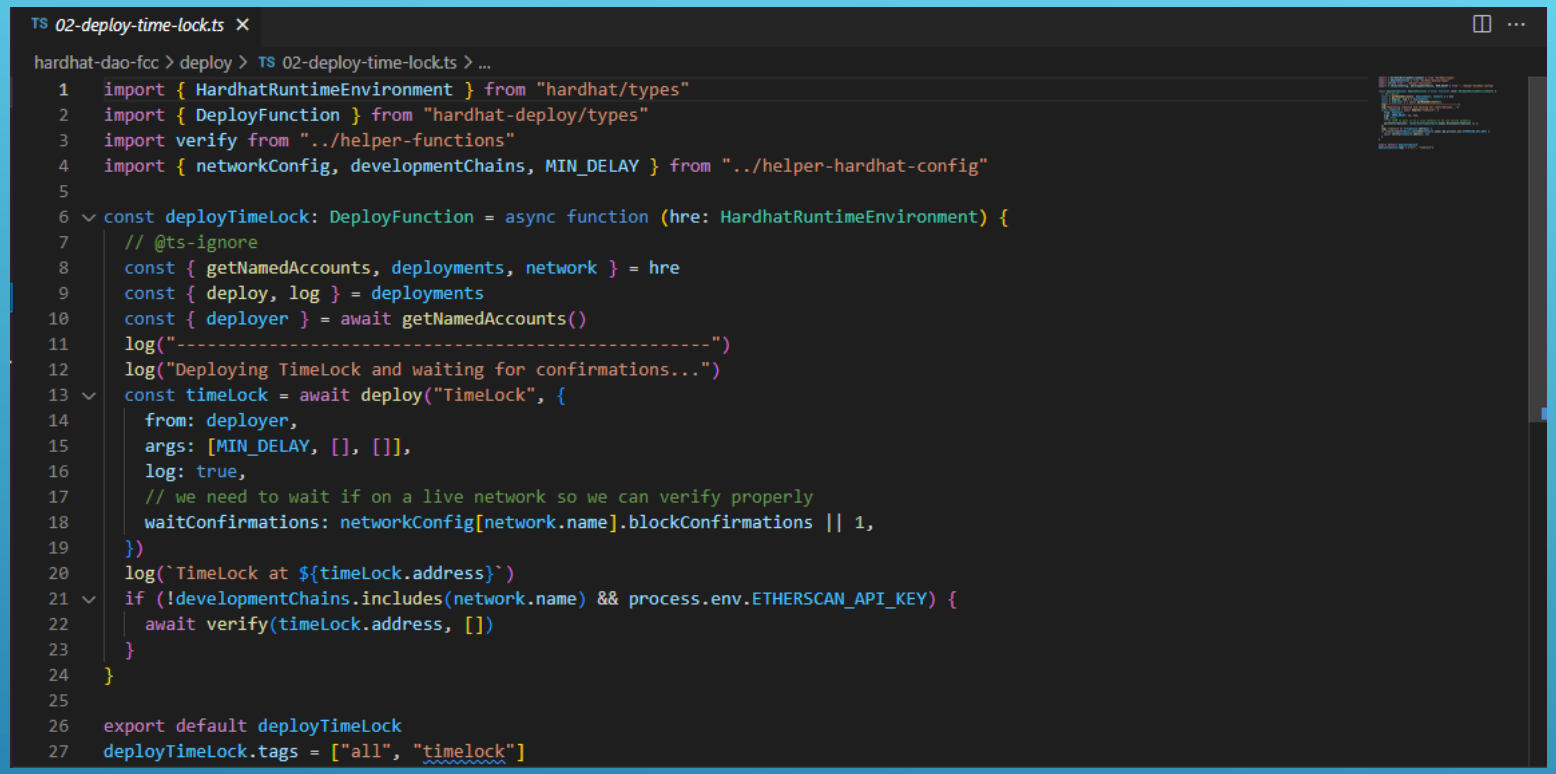


01-deploy-governor-token.ts
02-deploy-time-lock.ts
03-deploy-governor-contract.ts
04-setup-governance-contracts.ts
05-deploy-box.ts

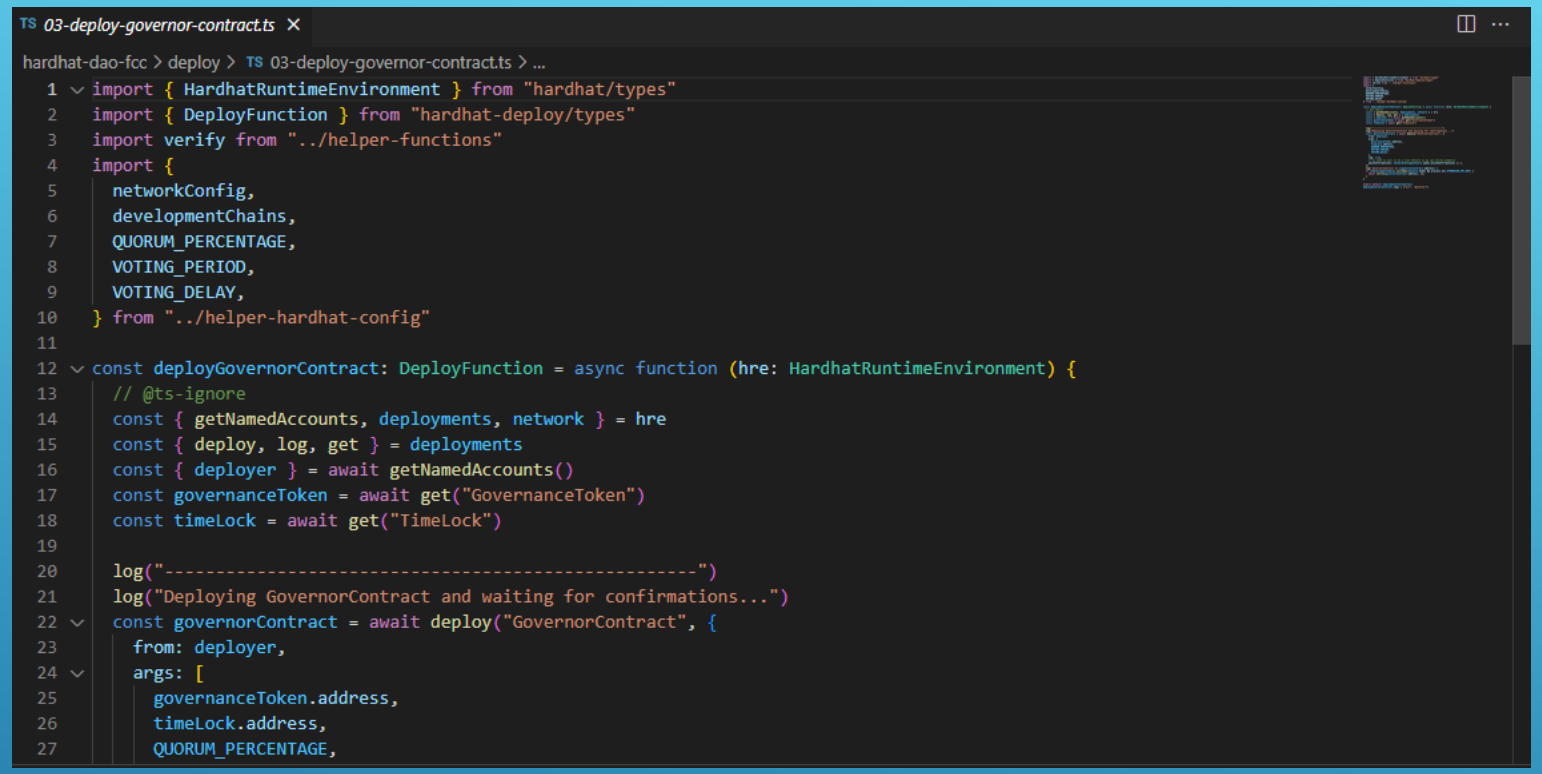
B DENGAN ISI 5
FILE



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



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



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

TS 04-setup-governance-contracts.ts

hardhat-dao-fcc > deploy > TS 04-setup-governance-contracts.ts > ...

```ts
import { HardhatRuntimeEnvironment } from "hardhat/types"
import { DeployFunction } from "hardhat-deploy/types"
import verify from "../helper-functions"
import { networkConfig, developmentChains, ADDRESS_ZERO } from "../helper-hardhat-config"
import { ethers } from "hardhat"

const setupContracts: DeployFunction = async function (hre: HardhatRuntimeEnvironment) {
  // @ts-ignore
  const { getNamedAccounts, deployments, network } = hre
  const { log } = deployments
  const { deployer } = await getNamedAccounts()
  const governanceToken = await ethers.getContract("GovernanceToken", deployer)
  const timeLock = await ethers.getContract("TimeLock", deployer)
  const governor = await ethers.getContract("GovernorContract", deployer)

  log("----------------------------------------------------")
  log("Setting up contracts for roles...")
  // would be great to use multicall here...
  const proposerRole = await timeLock.PROPOSER_ROLE()
  const executorRole = await timeLock.EXECUTOR_ROLE()
  const adminRole = await timeLock.TIMELOCK_ADMIN_ROLE()

  const proposerTx = await timeLock.grantRole(proposerRole, governor.address)
  await proposerTx.wait(1)
  const executorTx = await timeLock.grantRole(executorRole, ADDRESS_ZERO)
  await executorTx.wait(1)
  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 ×

hardhat-dao-fcc > deploy > TS 05-deploy-box.ts > ...

```ts
import { HardhatRuntimeEnvironment } from "hardhat/types"
import { DeployFunction } from "hardhat-deploy/types"
import verify from "../helper-functions"
import { networkConfig, developmentChains } from "../helper-hardhat-config"
import { ethers } from "hardhat"

const deployBox: DeployFunction = async function (hre: HardhatRuntimeEnvironment) {
  // @ts-ignore
  const { getNamedAccounts, deployments, network } = hre
  const { deploy, log } = deployments
  const { deployer } = await getNamedAccounts()
  log("----------------------------------------------------")
  log("Deploying Box and waiting for confirmations...")
  const box = await deploy("Box", {
    from: deployer,
    args: [],
    log: true,
    // we need to wait if on a live network so we can verify properly
    waitConfirmations: networkConfig[network.name].blockConfirmations || 1,
  })
  log(`Box at ${box.address}`)
  if (!developmentChains.includes(network.name) && process.env.ETHERSCAN_API_KEY) {
    await verify(box.address, [])
  }
  const boxContract = await ethers.getContractAt("Box", box.address)
  const timeLock = await ethers.getContract("TimeLock")
  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