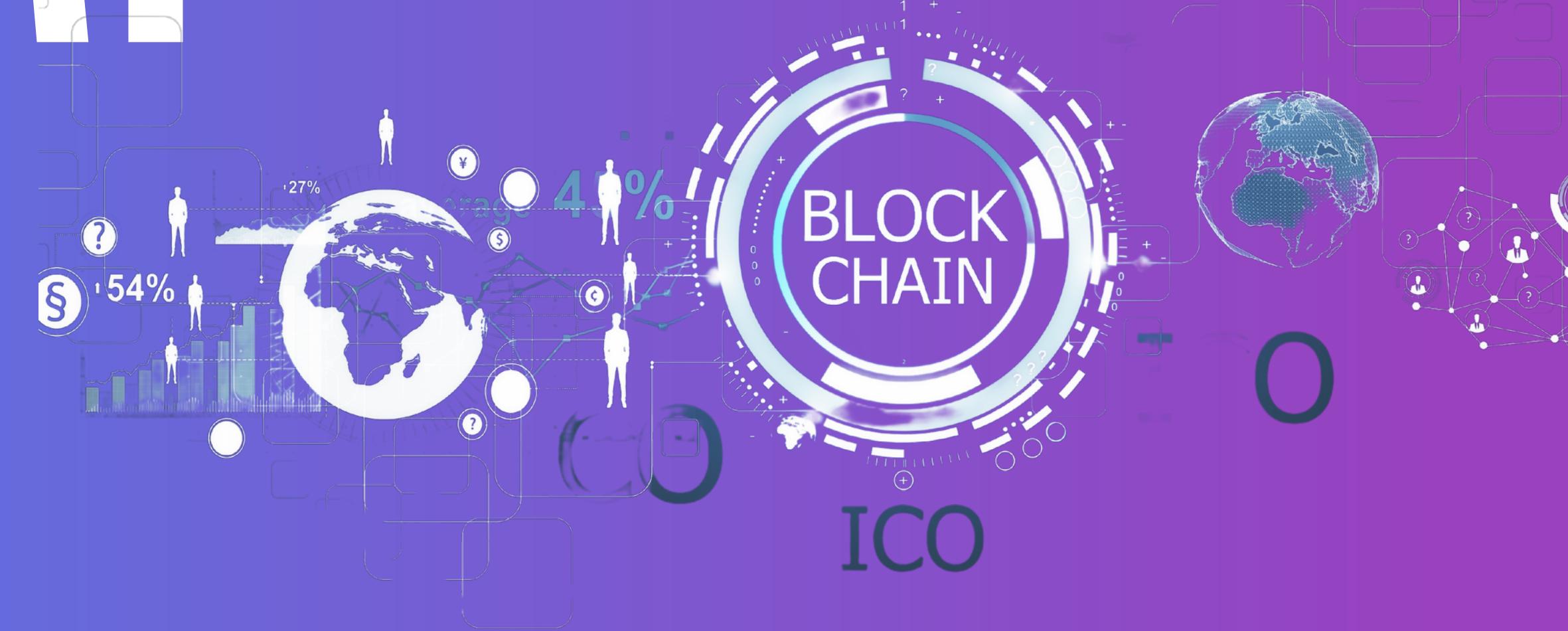
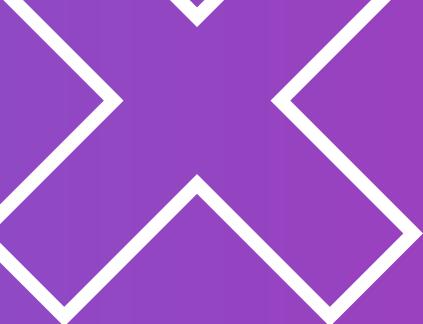


# HARDHAT

# ERC20

Get Started





```
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window  
darma@Darma-MacBook-Pro:~/Desktop/hardhat-erc20  
~/desktop  
mkdir hardhat-erc20  
~/desktop  
cd /Users/darma/Desktop/hardhat-erc20  
~/Desktop/hardhat-erc20  
yarn add --dev hardhat  
yarn add v1.22.17  
warning package-lock.json found. Your project contains lock files generated by  
s advised not to mix package managers in order to avoid resolution inconsisten  
d lock files. To clear this warning, remove package-lock.json.  
[1/4] 🔎 Resolving packages...  
warning ethereum-waffle > @ethereum-waffle/provider > ganache-core > web3-provi  
.4.2: Deprecated in favor of '@metamask/eth-sig-util'  
[2/4] 🚀 Fetching packages...  
[3/4] 🔁 Linking dependencies...  
warning " > bootstrap@5.1.3" has unmet peer dependency "@popperjs/core@^2.10.2"  
warning "react-moralis > use-immer@0.6.0" has incorrect peer dependency "react@  
warning "ethereum-waffle > @ethereum-waffle/compiler > typechain > ts-essential  
endency "typescript@>=3.7.0".  
[4/4] 🏗️ Building fresh packages...  
success Saved lockfile.  
success Saved 3 new dependencies.  
info Direct dependencies  
└── hardhat@2.10.0  
info All dependencies  
└── hardhat@2.10.0  
   └── mocha@10.0.0  
      └── workerpool@2.1
```

# CREATE NEW PROJECT

01.

Create Folder

mkdir hardhat-erc20

02.

Framework hardhat

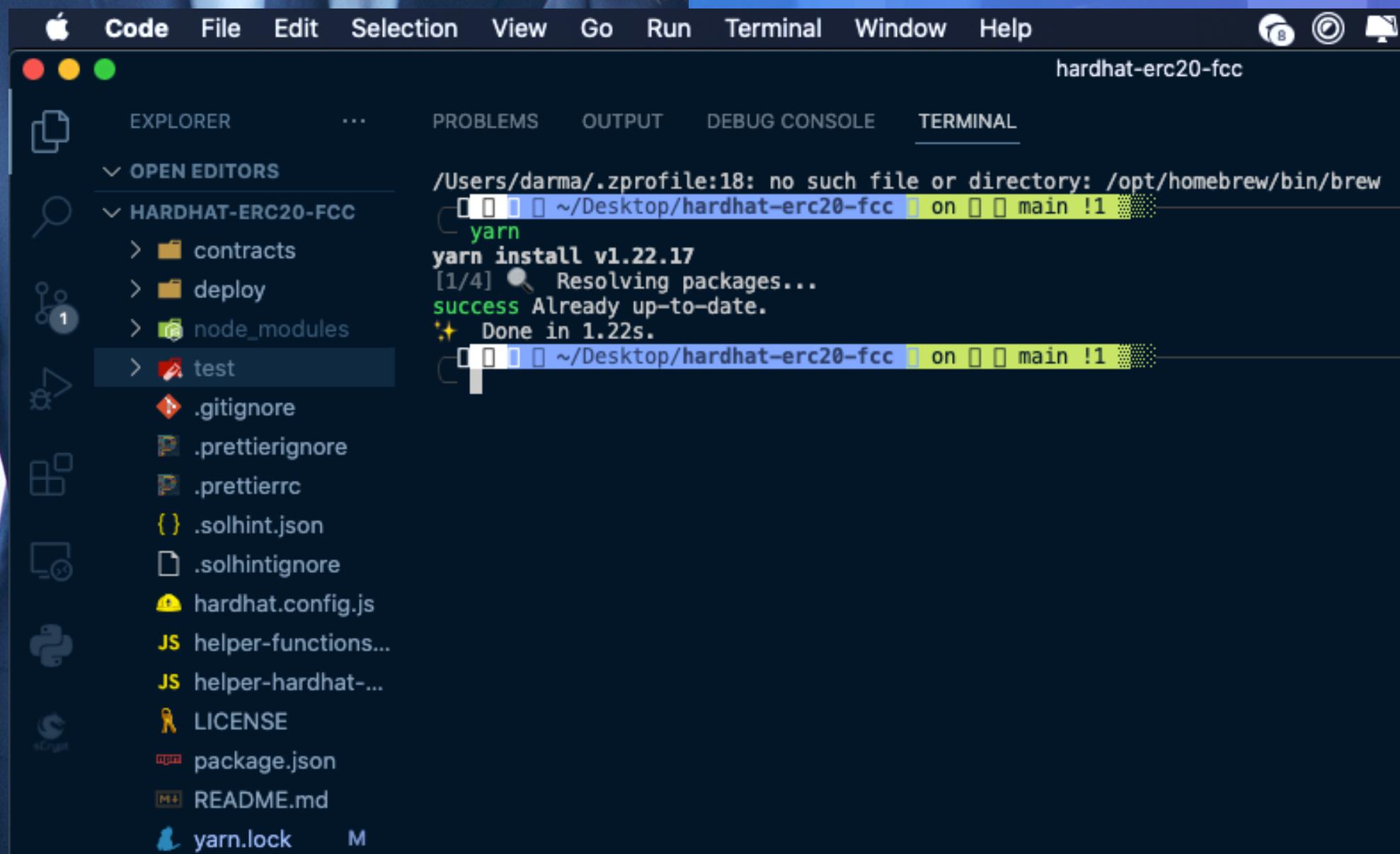
yarn add --dev hardhat

# Install Open Zeppelin

Instal dengan cara "npm install @openzeppelin/contract" pada terminal anda

```
~ Desktop/hardhat-erc20-fcc on main !1
  npm install @openzeppelin/contracts

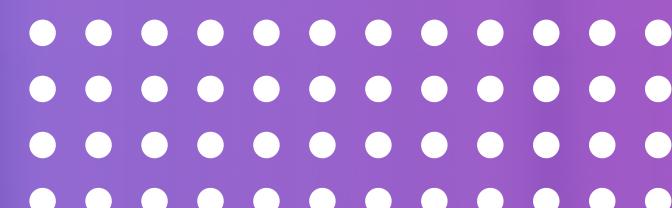
npm notice
npm notice New minor version of npm available! 8.11.0 => 8.13.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.13.2
npm notice Run npm install -g npm@8.13.2 to update!
npm notice
```



```
hardhat-erc20-fcc
Code File Edit Selection View Go Run Terminal Window Help
EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
/Users/darma/.zprofile:18: no such file or directory: /opt/homebrew/bin/brew
yarn install v1.22.17
[1/4] Resolving packages...
success Already up-to-date.
Done in 1.22s.
```



# Install Dependency



# OpenZeppelin-Contract



```
OurToken.sol — hardhat-erc20-fcc
contracts > OurToken.sol
1 // contracts/OurToken.sol
2 // SPDX-License-Identifier: MIT
3 pragma solidity ^0.8.7;
4
5 import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
6
7 contract OurToken is ERC20 {
8     constructor(uint256 initialSupply) ERC20("OurToken",
9         _mint(msg.sender, initialSupply);
10    }
11 }
12

ManualToken.sol 8
contracts > ManualToken.sol
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.7;
3
4 interface tokenRecipient {
5     function receiveApproval(
6         address _from,
7         uint256 _value,
8         address _token,
9         bytes calldata _extraData
10    ) external;
11 }
12
13 contract TokenERC20 {
14     // Public variables of the token
15     string public name;
16     string public symbol;
17     uint8 public decimals = 18;
18     // 18 decimals is the strongly suggested default,
19     uint256 public totalSupply;
20
21     // This creates an array with all balances
22     mapping(address => uint256) public balanceOf;
23     mapping(address => mapping(address => uint256)) p
24
25     // This generates a public event on the blockchain
26     event Transfer(address indexed from, address inde
27
28     // This generates a public event on the blockchain
29 }
```

PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL

/Users/darma/.zprofile:18: no such file or directory: /opt/homebrew/bin/brew

Ln 1, Col 1 Spaces: 2 UTF-8 LF Solidity ⚡ Go Live 🛡️ ✅ Pret

# Create code with openzeppelin

Blockchain



# Deploying Token

The screenshot shows a code editor interface with a dark theme. The main area displays a JavaScript file named `01-deploy-token.js`. The code is used to deploy a Solidity contract named `OurToken` on a Hardhat network. It requires the `hardhat` package and the `helper-hardhat-config` package. The deployment process involves specifying the initial supply and using the `deployer` to interact with the network. If the deployment is successful on a live network, it will verify the contract and log its address.

```
const { getNamedAccounts, deployments, network } = require("hardhat")
const {
  developmentChains,
  INITIAL_SUPPLY,
} = require("../helper-hardhat-config")
const { verify } = require("../helper-functions")

module.exports = async ({ getNamedAccounts, deployments }) => {
  const { deploy, log } = deployments
  const { deployer } = await getNamedAccounts()
  const ourToken = await deploy("OurToken", {
    from: deployer,
    args: [INITIAL_SUPPLY],
    log: true,
    // we need to wait if on a live network so we can verify properly
    waitConfirmations: network.config.blockConfirmations || 1,
  })
  log(`ourToken deployed at ${ourToken.address}`)

  if (
    !developmentChains.includes(network.name) &&
    process.env.ETHERSCAN_API_KEY
  ) {
```

The terminal pane at the bottom shows the command `yarn hardhat deploy` being run, which compiles the Solidity code and deploys the `OurToken` contract to the local network. The output includes the transaction hash, the deployed address, and the gas used for the deployment.

```
yarn hardhat deploy
yarn run v1.22.17
$ /Users/darma/Desktop/hardhat-erc20-fcc/node_modules/.bin/hardhat deploy
Compiling 6 files with 0.8.7
Solidity compilation finished successfully
deploying "OurToken" (tx: 0x167069b2c1ff4ced4891bcc9964738e61f35af2c80ac024bd4a5aee3b3e6eb0)...: deployed at 0x5FbDB2315678afecb367f032d93F642f64180aa3 with 1225998 gas
ourToken deployed at 0x5FbDB2315678afecb367f032d93F642f64180aa3
✨ Done in 8.36s.
```



# Deploying Token

```
~/Desktop/hardhat-erc20-fcc on main !1 ?1
yarn hardhat deploy
yarn run v1.22.17
$ /Users/darma/Desktop/hardhat-erc20-fcc/node_modules/.bin/hardhat deploy
Compiling 6 files with 0.8.7
Solidity compilation finished successfully
deploying "OurToken" (tx: 0x167069b2c1ff48ced4891bcc9964738e61f35af2c80ac024bd4a5aee3b3e6eb0)...
f032d93F642f64180aa3 with 1225998 gas
ourToken deployed at 0x5FbDB2315678afecb367f032d93F642f64180aa3
+ Done in 8.36s.
```