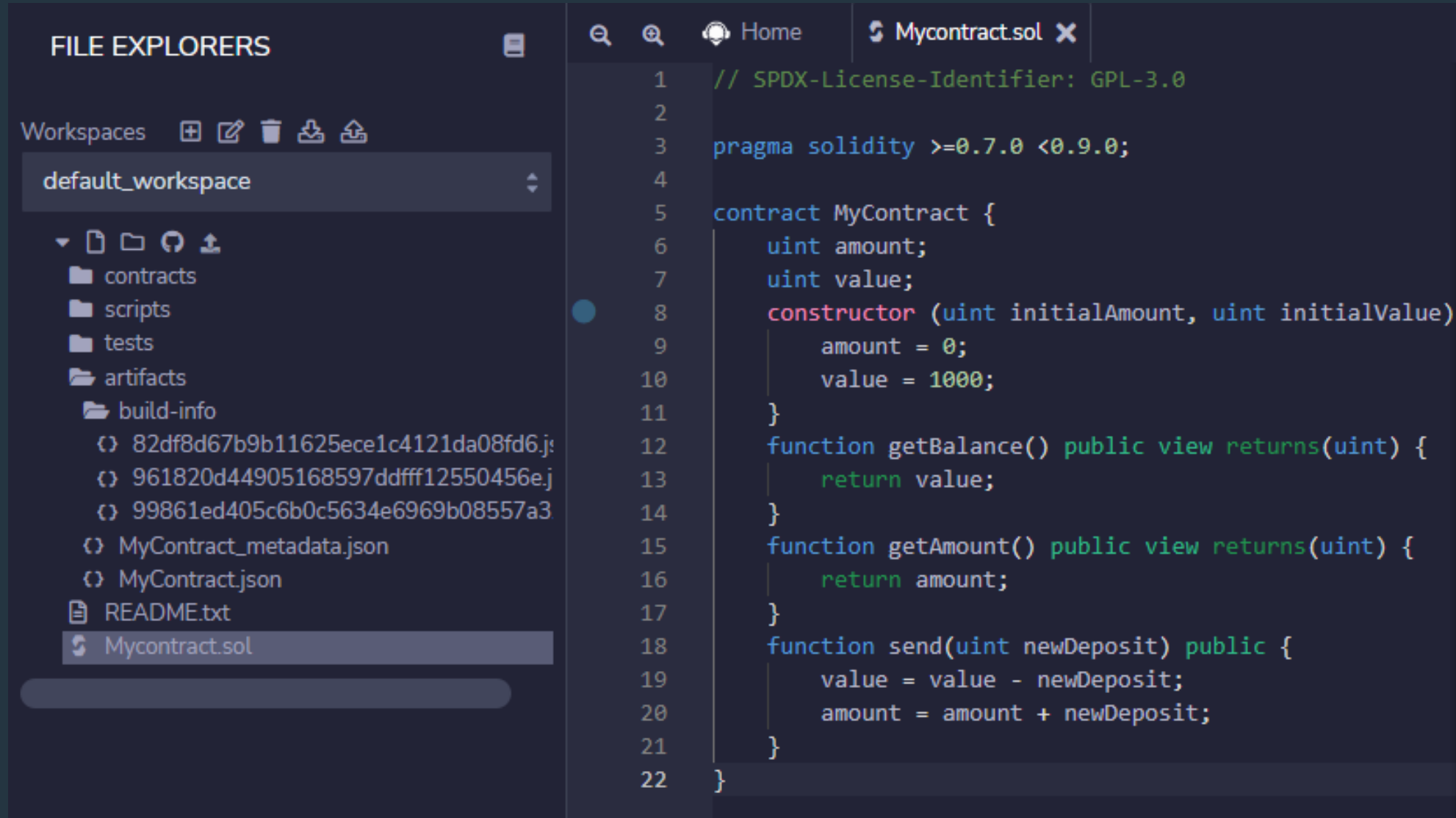


Ethereum Tutorial

Hanif Shafwan Mahib -- 1103194150

Membuat Kontrak



The image shows a screenshot of the Visual Studio Code (VS Code) interface. On the left, the 'FILE EXPLORERS' sidebar is open, showing a workspace named 'default_workspace'. Inside this workspace, there are several folders: 'contracts', 'scripts', 'tests', 'artifacts', and 'build-info'. Below these folders, a list of files is displayed, including '82df8d67b9b11625ece1c4121da08fd6.js', '961820d44905168597ddfff12550456e.js', '99861ed405c6b0c5634e6969b08557a3.js', 'MyContract_metadata.json', 'MyContract.json', 'README.txt', and 'Mycontract.sol'. The 'Mycontract.sol' file is selected and highlighted.

On the right, the editor window shows the content of 'Mycontract.sol'. The code is written in Solidity and includes a license header, pragma statement, and a contract definition for 'MyContract'. The contract has two state variables, 'amount' and 'value', both of type 'uint'. It features a constructor that initializes 'amount' to 0 and 'value' to 1000. There are three public functions: 'getBalance()' which returns 'value', 'getAmount()' which returns 'amount', and 'send()' which decreases 'value' by 'newDeposit' and increases 'amount' by 'newDeposit'.

```
1 // SPDX-License-Identifier: GPL-3.0
2
3 pragma solidity >=0.7.0 <0.9.0;
4
5 contract MyContract {
6     uint amount;
7     uint value;
8     constructor (uint initialAmount, uint initialValue)
9     {
10         amount = 0;
11         value = 1000;
12     }
13     function getBalance() public view returns(uint) {
14         return value;
15     }
16     function getAmount() public view returns(uint) {
17         return amount;
18     }
19     function send(uint newDeposit) public {
20         value = value - newDeposit;
21         amount = amount + newDeposit;
22     }
23 }
```

Melakukan Compile

The image shows the Solidity Compiler web interface. On the left, the 'COMPILER' section is set to '0.7.1+commit.f4a555be'. Below this, there are checkboxes for 'Include nightly builds', 'Auto compile', and 'Hide warnings', all of which are currently unchecked. An 'Advanced Configurations' link is also present. A large blue button labeled 'Compile Mycontract.sol' is prominent. Below the compiler settings, the 'CONTRACT' section shows 'MyContract (Mycontract.sol)' selected. At the bottom of the left panel, there are buttons for 'Publish on Ipfs', 'Publish on Swarm', and 'Compilation Details', along with links for 'ABI' and 'Bytecode'.

The main editor on the right displays the Solidity code for 'MyContract.sol'. The code includes a license header, a pragma statement for Solidity version, and a contract definition with a constructor, two getter functions, and a send function. A red warning icon is visible on line 8, indicating a compiler warning.

```
1 // SPDX-License-Identifier: GPL-3.0
2
3 pragma solidity >=0.7.0 <0.9.0;
4
5 contract MyContract {
6     uint amount;
7     uint value;
8     constructor (uint initialAmount, uint initialValue)
9     {
10         amount = 0;
11         value = 1000;
12     }
13     function getBalance() public view returns(uint) {
14         return value;
15     }
16     function getAmount() public view returns(uint) {
17         return amount;
18     }
19     function send(uint newDeposit) public {
20         value = value - newDeposit;
21         amount = amount + newDeposit;
22     }
23 }
```

Melakukan Deploy

The screenshot displays the Remix IDE interface during the deployment of a Solidity contract. The left sidebar contains the 'DEPLOY & RUN TRANSACTIONS' panel, which is divided into several sections:

- ENVIRONMENT:** Set to 'JavaScript VM (London)'.
- ACCOUNT:** A VM account with address '0x5B3...eddC4' and a balance of '99.99999999'.
- GAS LIMIT:** Set to '3000000'.
- VALUE:** Set to '0' Wei.
- CONTRACT:** Selected as 'MyContract - Mycontract.sol'.
- DEPLOY:** Fields for 'INITIALAMOUNT: 10000' and 'INITIALVALUE: 10'. A 'transact' button is visible.
- Publish to IPFS:** A checkbox that is currently unchecked.
- OR:** A section with 'At Address' and 'Load contract from Address' buttons.
- Transactions recorded:** Shows '1' transaction.
- Deployed Contracts:** Lists 'MYCONTRACT AT 0XD91...39138 (MEM)'.

The main editor area shows the Solidity code for 'MyContract.sol':

```
1 // SPDX-License-Identifier: GPL-3.0
2
3 pragma solidity >=0.7.0 <0.9.0;
4
5 contract MyContract {
6     uint amount;
7     uint value;
8     constructor (uint initialAmount, uint initialValue) public {
9         amount = 0;
10        value = 1000;
11    }
12    function getBalance() public view returns(uint) {
13        return value;
14    }
15    function getAmount() public view returns(uint) {
16        return amount;
17    }
18    function send(uint newDeposit) public {
19        value = value - newDeposit;
20        amount = amount + newDeposit;
21    }
22 }
```

Below the code editor, the 'ContractDefinition MyContract' section shows '1 reference(s)'. The bottom panel displays the transaction details for the deployment:

ContractDefinition MyContract 1 reference(s) ^ v

0 ☐ listen on all transactions Search with transaction hash or address

The following libraries are accessible:

- web3 version 1.5.2
- ethers.js
- remix

Type the library name to see available commands.
creation of MyContract pending...


[vm] from: 0x5B3...eddC4 to: value: 0 wei data: 0x608...0000a logs: 0 hash: 0x3d5...2a802



Melakukan transaksi ether

OR

At Address Load contract from Address

Transactions recorded 2 >

Deployed Contracts 

▼ MYCONTRACT AT 0XD91...39138 (ME)  


send 100 ▼

getAmount

o: uint256: 100

getBalance

o: uint256: 900

Low level interactions 


CALLDATA



Transact

OR

At Address Load contract from Address

Transactions recorded 1 >

Deployed Contracts 

▼ MYCONTRACT AT 0XD91...39138 (ME)  


send uint256 newDeposit ▼

getAmount

o: uint256: 0

getBalance

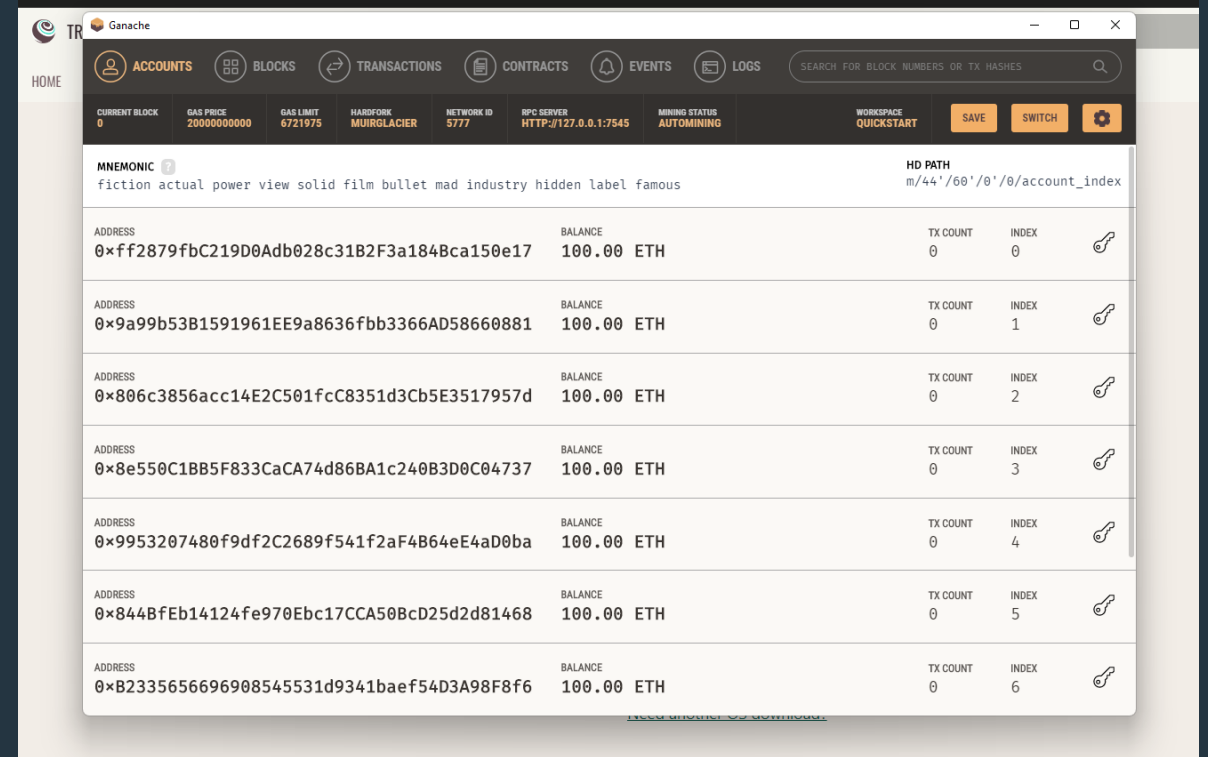
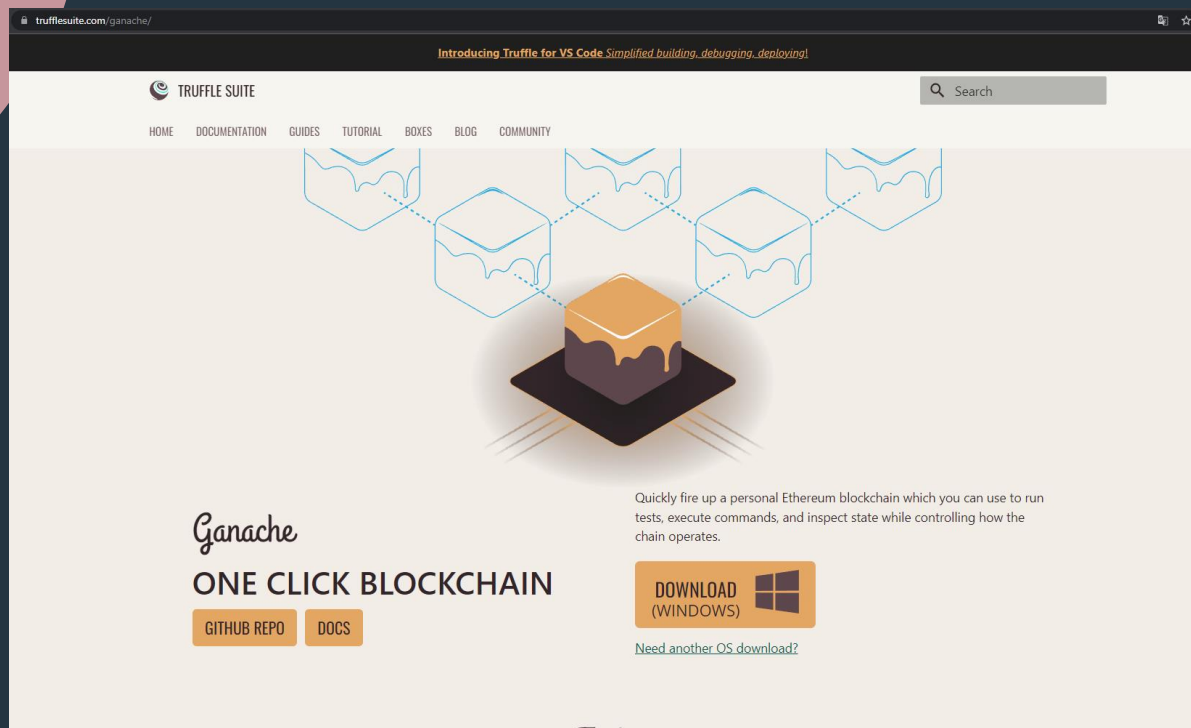
o: uint256: 1000

Low level interactions 

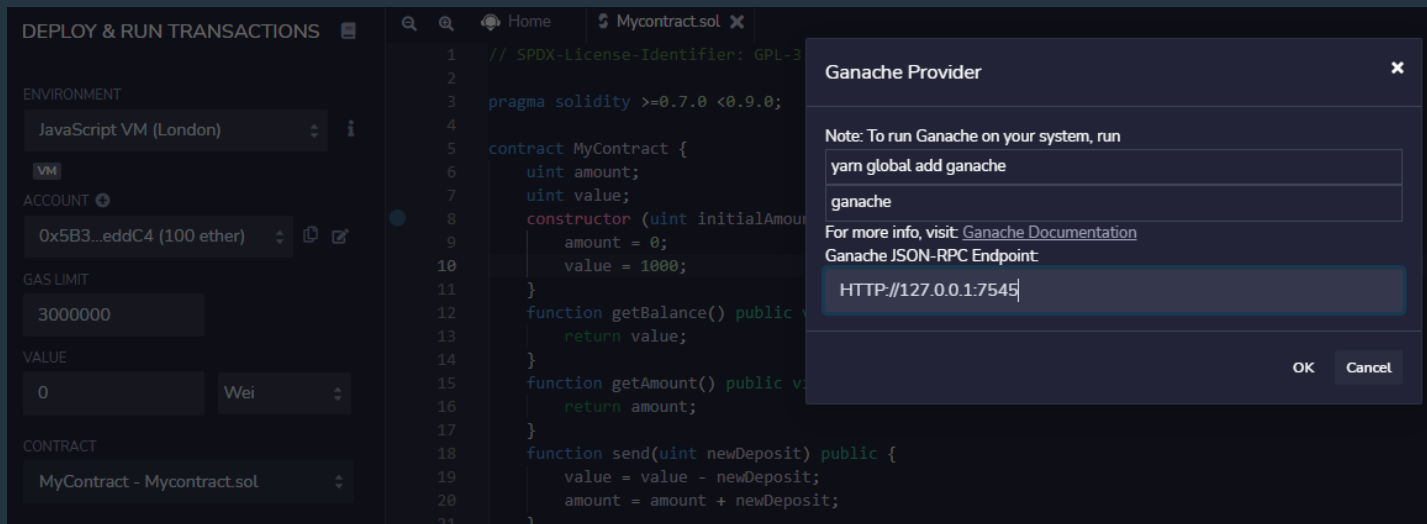
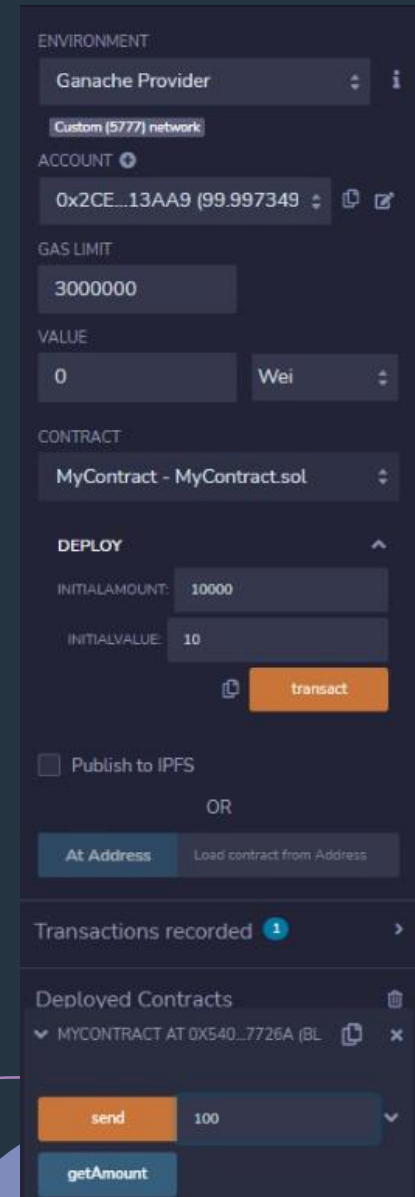
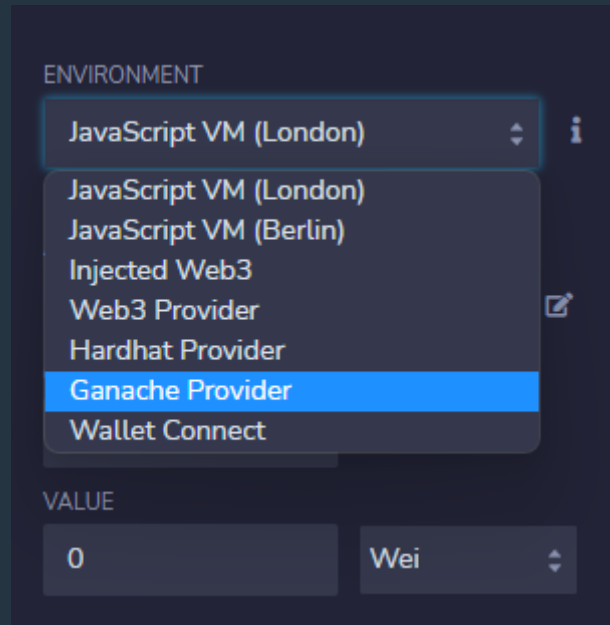
CALLDATA

Transact

Mendownload ganache lalu install



Mengubah Environment menjadi Ganache



Mencoba transaksi Ether menggunakan ganache

Setelah Deploy

TX HASH

0x098bd46863ef9c9567fec3fd204b4542f95f9fb48763363e3351310dec13e147

FROM ADDRESS

0x2CE8fe5cA270274f35CA470878840D3Accd13AA9

CREATED CONTRACT ADDRESS

0x5404B10A60E6679cAaEBF5e56Bbd9Db04b97726A

GAS USED

132502

VALUE

0

CONTRACT CREATION

Setelah sending money

TX HASH

0xd2c1288394f24ba7d321747a45eb392aa361ae286b7abc16633cebdde340fe96

FROM ADDRESS

0x2CE8fe5cA270274f35CA470878840D3Accd13AA9

TO CONTRACT ADDRESS

0x5404B10A60E6679cAaEBF5e56Bbd9Db04b97726A

GAS USED

48072

VALUE

0

CONTRACT CALL

TX HASH

0x098bd46863ef9c9567fec3fd204b4542f95f9fb48763363e3351310dec13e147

FROM ADDRESS

0x2CE8fe5cA270274f35CA470878840D3Accd13AA9

CREATED CONTRACT ADDRESS

0x5404B10A60E6679cAaEBF5e56Bbd9Db04b97726A

GAS USED

132502

VALUE

0

CONTRACT CREATION



Thanks