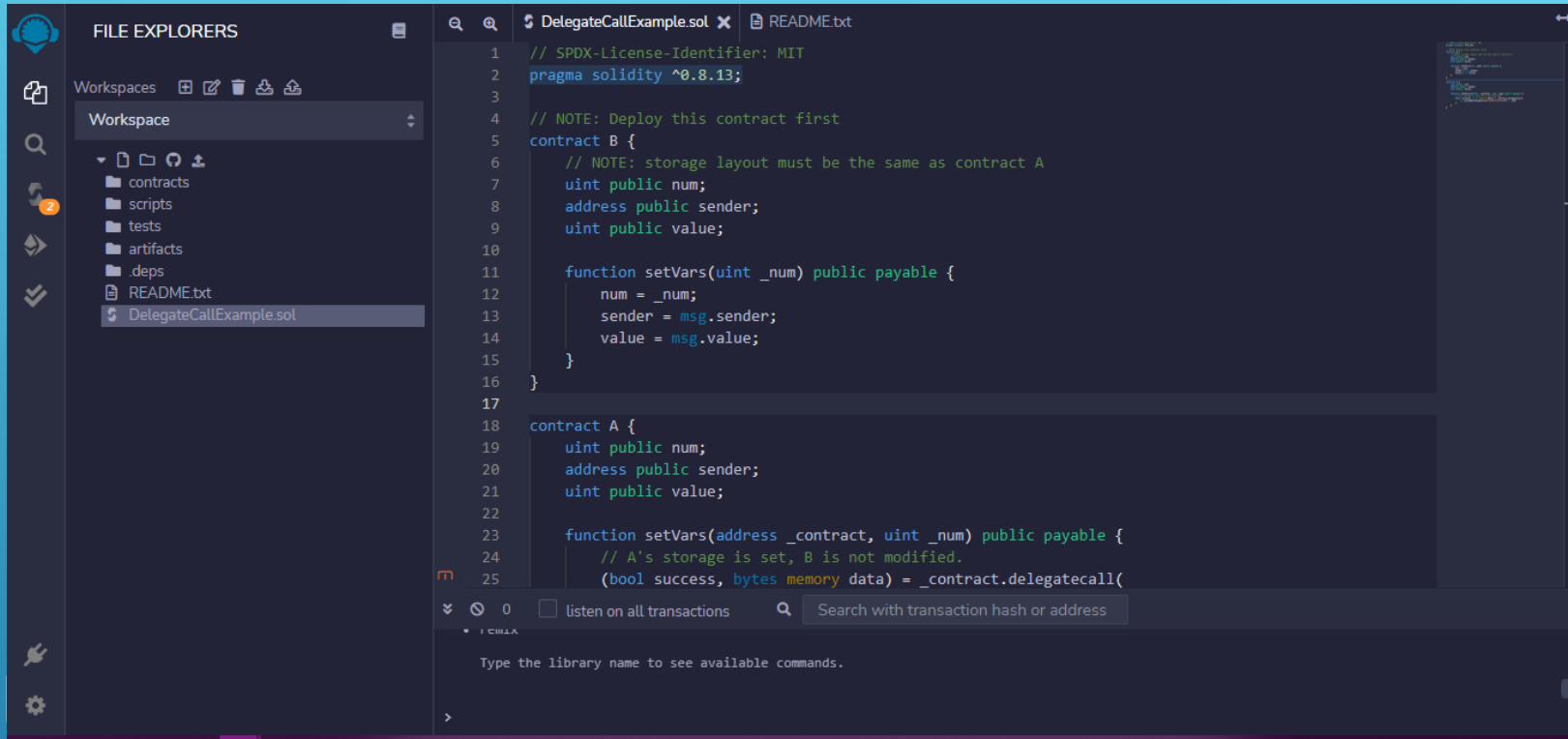


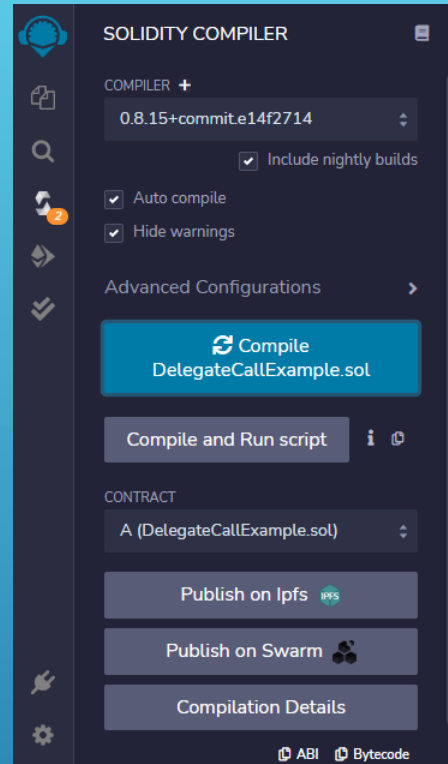
LESSON 16: HARDHAT UPGRADES

Delegate Call

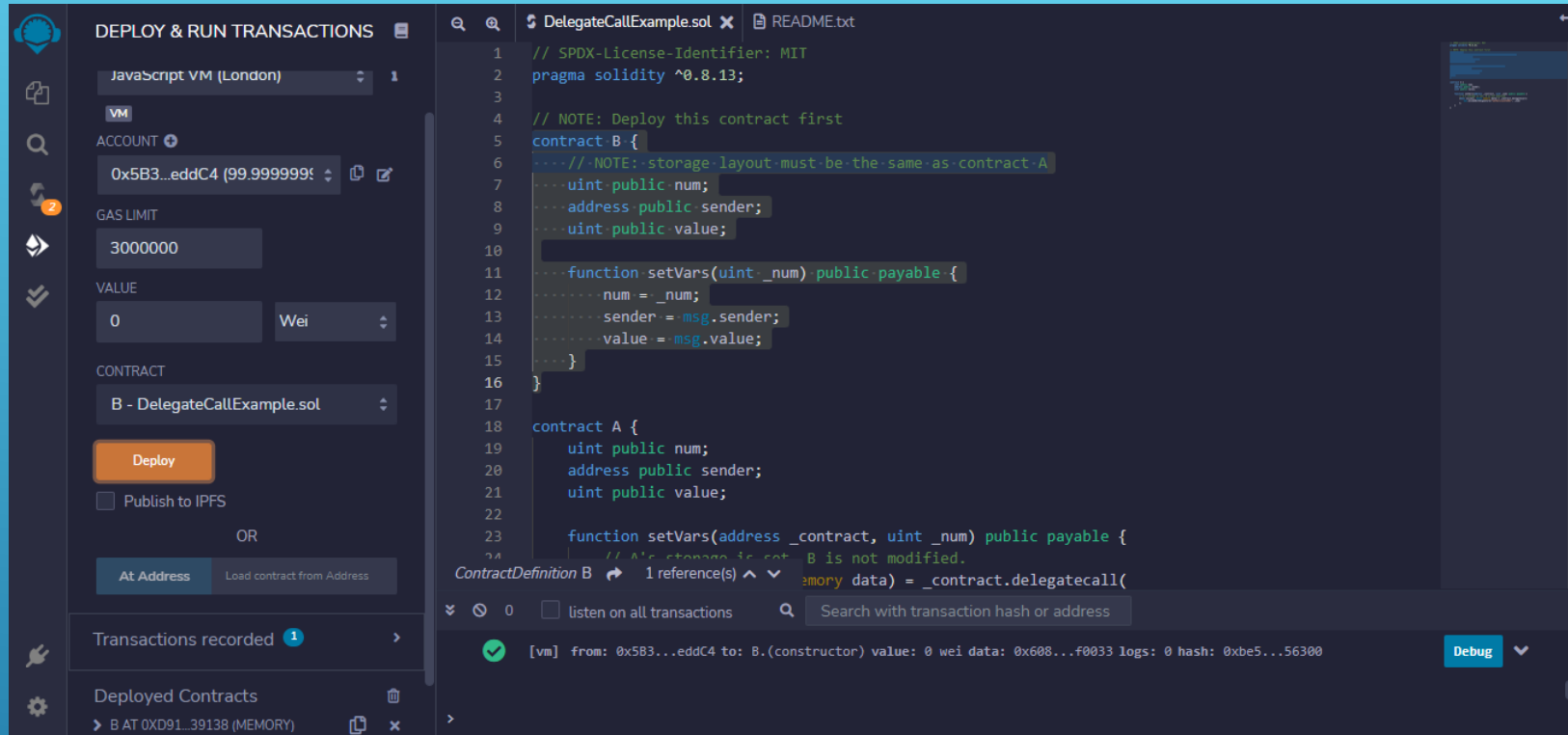
Several thin, parallel white lines are drawn diagonally across the right side of the slide, starting from the bottom left and extending towards the top right.



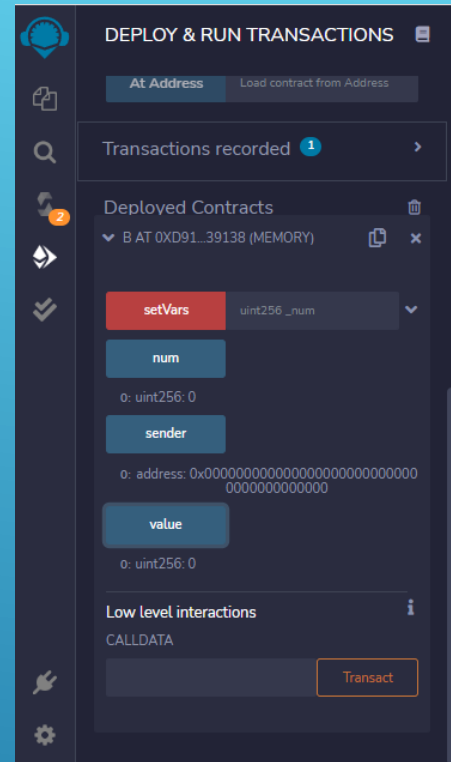
B
DELEGATECALLEXAMPLE.SOL
DENGAN ISI SEPERTI DI LINK
[HTTPS://SOLIDITY-BY-
EXAMPLE.ORG/DELEGATECALL](https://solidity-by-example.org/delegatecall)



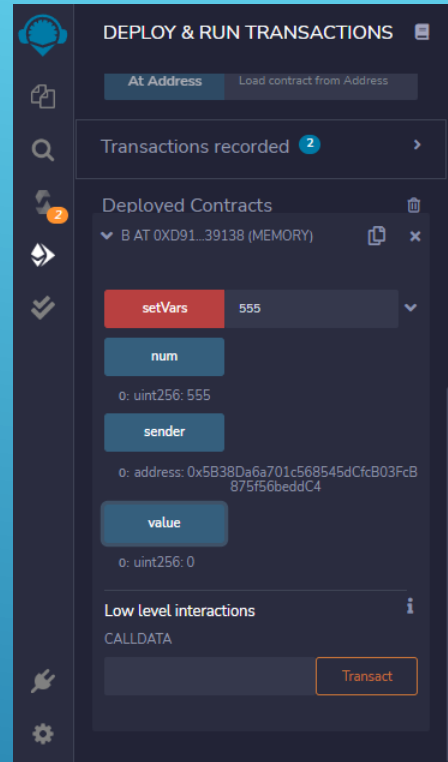
COMPILE FILE
DELEGATECALLEXAMPLE.SOL
DENGAN VERSI 0.8.13 KEATAS



LALU KITA DEPLOY CONTRACT B, MAKA
AKAN MUNCUL TAMPILAN SEPERTI INI



KLIK NUM, SENDER, DAN VALUE MAKA
ISINYA AKAN KOSONG SEPERTI DIBAWAH
INI PADA CONTRACT B.



LALU KITA SETVARS 555 MAKA IA AKAN MENGISI NUM 555 DAN AKAN MENGIRIM SENDER.

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT
JavaScript VM (London)

ACCOUNT
0x5B3...eddC4 (99.999999%)

GAS LIMIT
3000000

VALUE
0 Wei

CONTRACT
A - DelegateCallExample.sol

Deploy

☐ Publish to IPFS

OR

At Address

Load contract from Address

Transactions recorded 3

DelegateCallExample.sol

12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

```
function setVars(uint _num) public payable {
    num = _num;
    sender = msg.sender;
    value = msg.value;
}

contract A {
    uint public num;
    address public sender;
    uint public value;

    function setVars(address _contract, uint _num) public payable {
        //A's storage is set, B is not modified.
        (bool success, bytes memory data) = _contract.delegatecall(
            abi.encodeWithSignature("setVars(uint256)", _num)
        );
    }
}
```

ContractDefinition A 1 reference(s)

0 ☐ listen on all transactions

Search with transaction hash or address

✓ [vm] from: 0x5B3...eddC4 to: A.(constructor) value: 0 wei data: 0x608...f0033 logs: 0 hash: 0xa9c...7e878

Debug

DEPLOY & RUN TRANSACTIONS

value
0: uint256: 0

Low level interactions
CALLDATA
Transact

A AT 0XF8E...9FBEB (MEMORY)

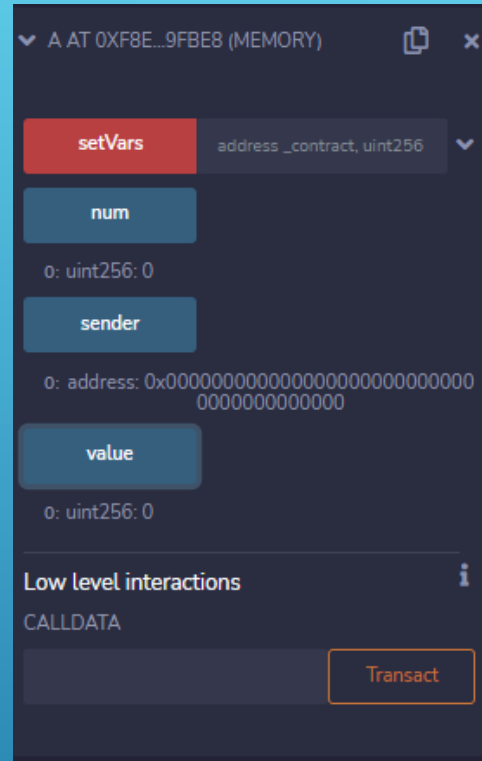
setVars address _contract, uint256

num

sender

value

Low level interactions
CALLDATA
Transact



KLIK NUM, SENDER, DAN VALUE MAKA
ISINYA AKAN KOSONG SEPERTI DIBAWAH
INI, PADA CONTRACT A.

LALU KITA PINJAM FUNCTION DARI CONTRACT B RUN SEKALI, LALU HAPUS KEMBALI FUNCTION TERSEBUT, DAN INI ADALAH PERUMPAMAAN SISTEM DELEGATE BERKERJA

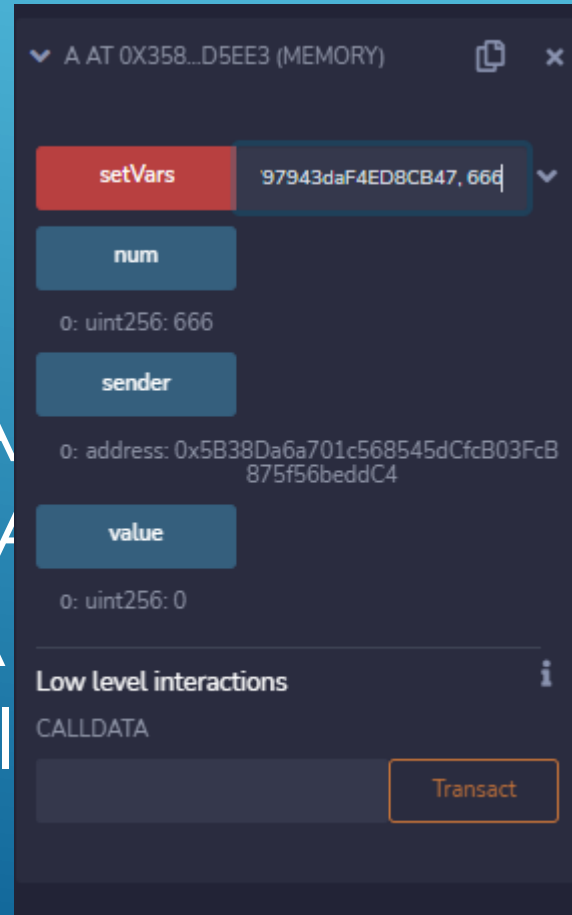
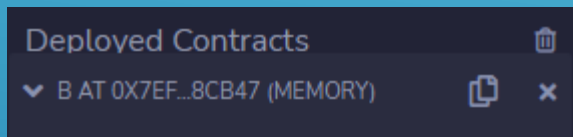
This screenshot shows the Remix IDE interface. On the left, the 'DEPLOY & RUN TRANSACTIONS' panel displays the contract address 0x58380a6a701c568545dCfc803Fc8875f56beddC4. The 'Low level interactions' section shows a 'Transact' button. The main editor displays the Solidity code for 'DelegateCallExample.sol', which includes a 'setVars' function and a 'contract A' that uses 'delegatecall' to call a function from another contract.

```
11 function setVars(uint _num) public payable {
12     num = _num;
13     sender = msg.sender;
14     value = msg.value;
15 }
16
17
18 contract A {
19     uint public num;
20     address public sender;
21     uint public value;
22
23     function setVars(address _contract, uint _num) public payable {
24         // A's storage is set, B is not modified.
25         (bool success, bytes memory data) = _contract.delegatecall(
26             abi.encodeWithSignature("setVars(uint256)", _num)
27         );
28     }
29
30     function setVars(uint _num) public payable {
31         num = _num;
32         sender = msg.sender;
33         value = msg.value;
34     }
35 }
```

This screenshot shows the same Remix IDE interface after a transaction has been executed. The 'Low level interactions' section now displays the transaction details, including the 'setVars' function call. The main editor shows the updated code for 'contract A', where the 'setVars' function is now a simple assignment, and the 'delegatecall' function has been removed. The bottom panel shows the transaction details, including the 'CALL' data and the 'Debug' button.

```
11 function setVars(uint _num) public payable {
12     num = _num;
13     sender = msg.sender;
14     value = msg.value;
15 }
16
17
18 contract A {
19     uint public num;
20     address public sender;
21     uint public value;
22
23     function setVars(address _contract, uint _num) public payable {
24         // A's storage is set, B is not modified.
25         (bool success, bytes memory data) = _contract.delegatecall(
26             abi.encodeWithSignature("setVars(uint256)", _num)
27         );
28     }
29
30     function setVars(uint _num) public payable {
31         num = _num;
32         sender = msg.sender;
33         value = msg.value;
34     }
35 }
```

CALL [call] from: 0x58380a6a701c568545dCfc803Fc8875f56beddC4 to: A.value() data: 0x3fa...4f245



KITA COPY ADDRESS PA
LALU ISI SETVARS DENGAN
DI CONTRACT A, MAKA
BERJALAN DENGAN BAIK
CONTRACT B

A screenshot of a Solidity IDE with a dark theme. The main editor window shows a Solidity contract named 'A'. It has two public variables: 'Angka' of type 'uint' and 'Pengirim' of type 'address'. There is a public payable function 'setVars' that takes a 'uint _num' parameter and sets 'num', 'sender', and 'value' to the provided values. The file explorer on the left shows 'DelegateCallExample.sol' and 'README.txt'. The background of the slide has large, semi-transparent white text that reads: 'JIKA KITA MENGUBAH NAMA VARIABLE PADA CONTRACT A, MAKA TIDAK AKAN BERPENGARUH PADA CONTRACT B'. The text is partially obscured by the IDE window.

```

11     function setVars(uint _num) public payable {
12         num = _num;
13         sender = msg.sender;
14         value = msg.value;
15     }
16 }
17
18 contract A {
19     uint public Angka;
20     address public Pengirim;
21     uint public Nilai;
22
23     function setVars(address _contract, uint _num) public payable {
24         // A's storage is set, B is not modified.
25         (bool success, bytes memory data) = _contract.delegatecall(
26             abi.encodeWithSignature("setVars(uint256)", _num)
27         );
28     }
29 }
30

```

LALU KITA D
A DAN B

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT
JavaScript VM (London)

ACCOUNT
0x5B3...eddC4 (99.999999%)

GAS LIMIT
3000000

VALUE
0 Wei

CONTRACT
B - DelegateCallExample.sol

Deploy

☐ Publish to IPFS

OR

At Address Load contract from Address

Transactions recorded 11

ENVIRONMENT
JavaScript VM (London)

ACCOUNT
0x5B3...eddC4 (99.999999%)

GAS LIMIT
3000000

VALUE
0 Wei

CONTRACT
A - DelegateCallExample.sol

Deploy

☐ Publish to IPFS

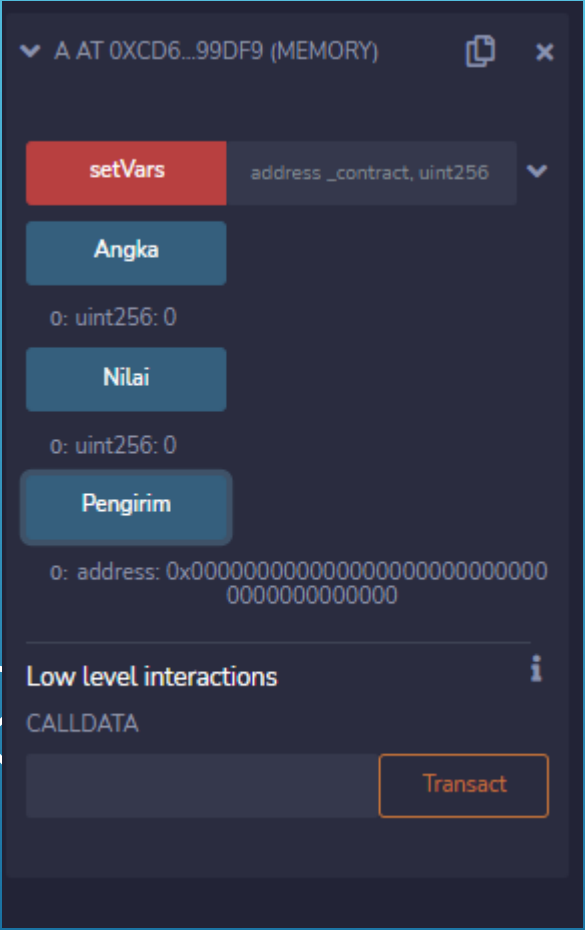
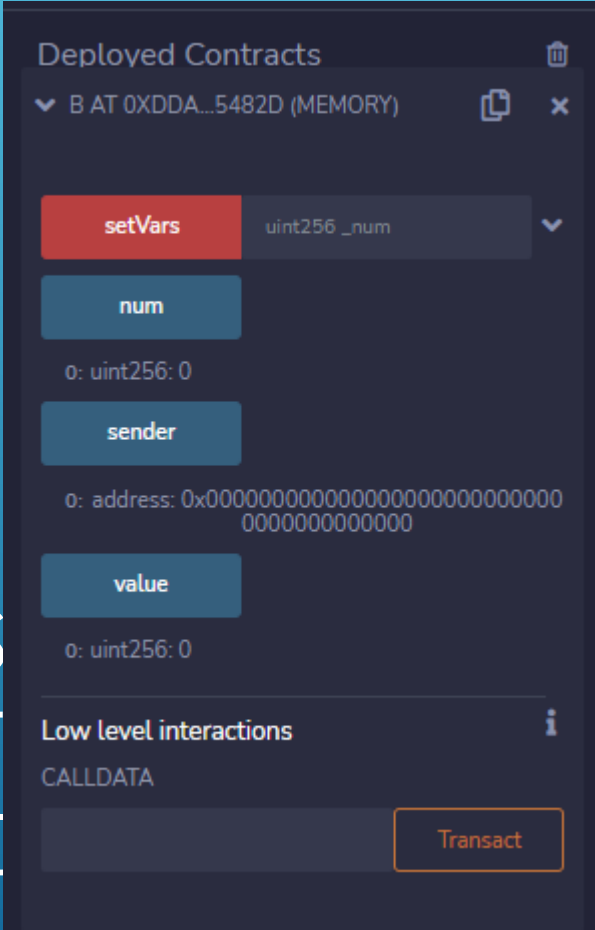
OR

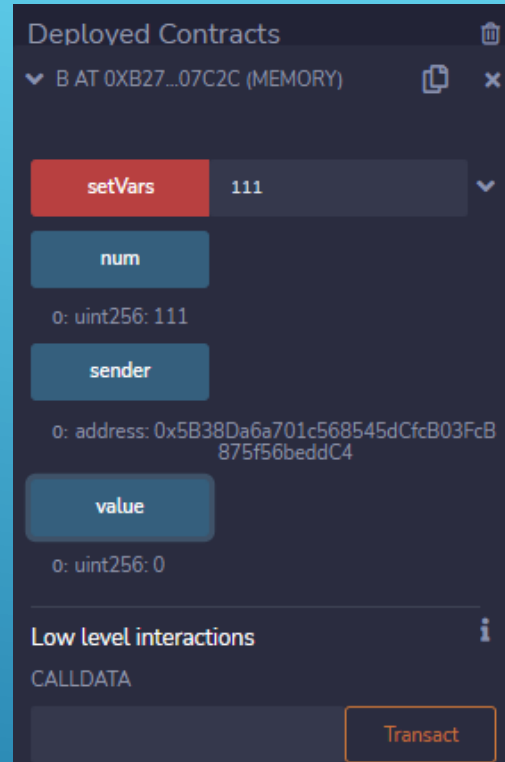
At Address Load contract from Address

Transactions recorded 11

ACT

KLIK NUM, S
 CONTRACT
 DAN NILAI D





KITA SETVARS LAGI PADA CONTRACT
B

COPY ALAMAT CONTRACT B DAN MASUKAN DI
SETVARS PADA A BESERTA NILAI YANG
DIINGINKAN, M
PERSIS SAMA D
DENGAN NAM
TRACT A AKAN BEKERJA
NTRACT B, WALAUPUN
YANG BERBEDA

▼ A AT 0XCD6...99DF9 (MEMORY)

setVars

0xb27A31f1b0AF2946B71 ▼

Angka

0: uint256: 222

Nilai

0: uint256: 0

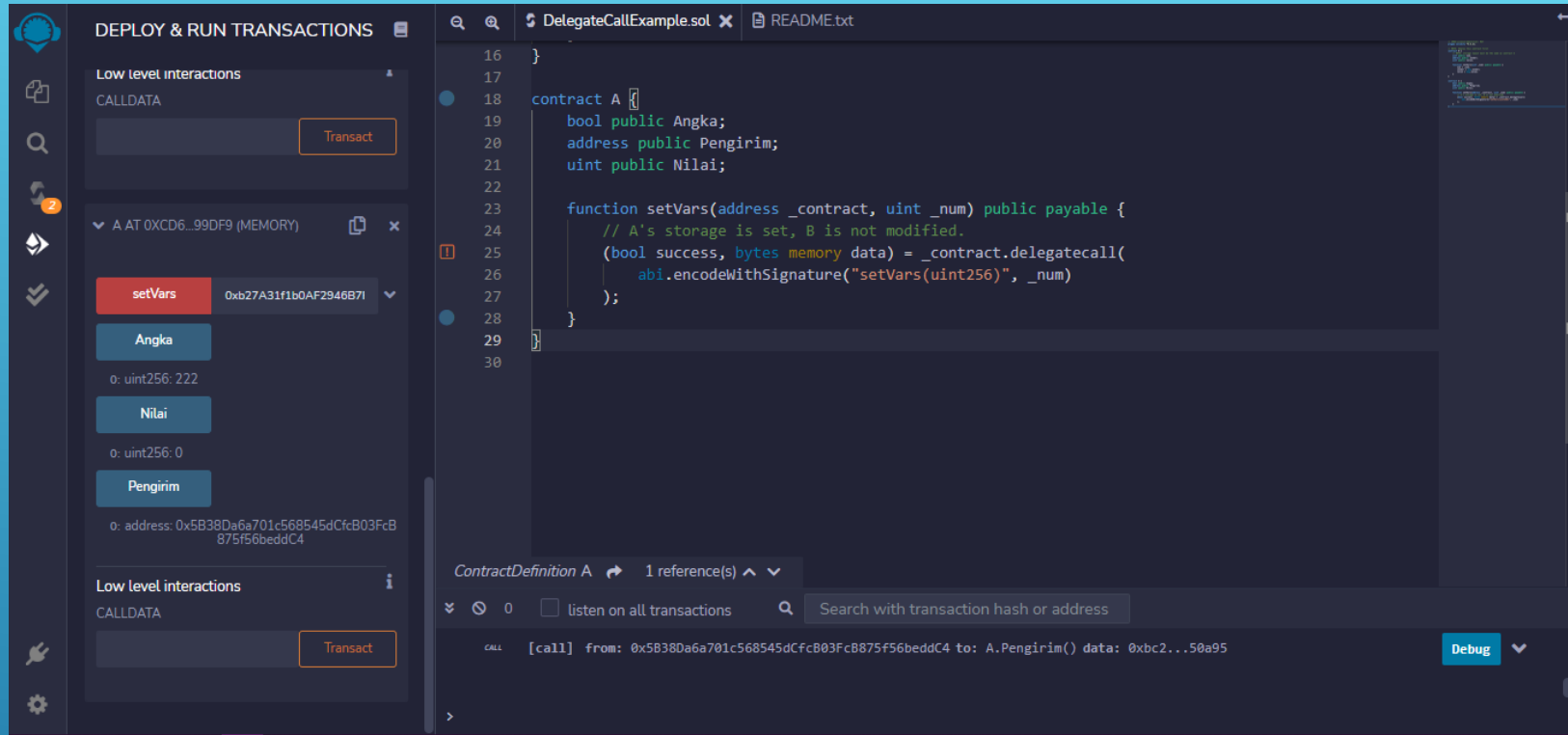
Pengirim

0: address: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

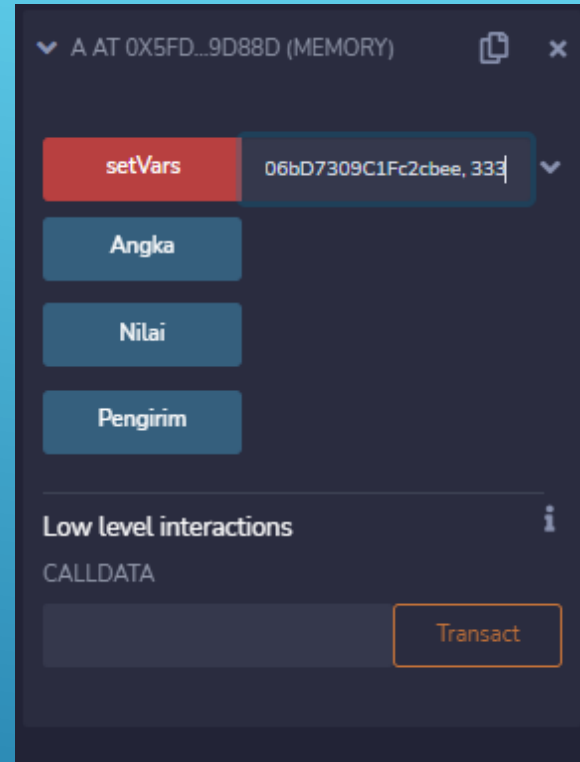
Low level interactions

CALLDATA

Transact

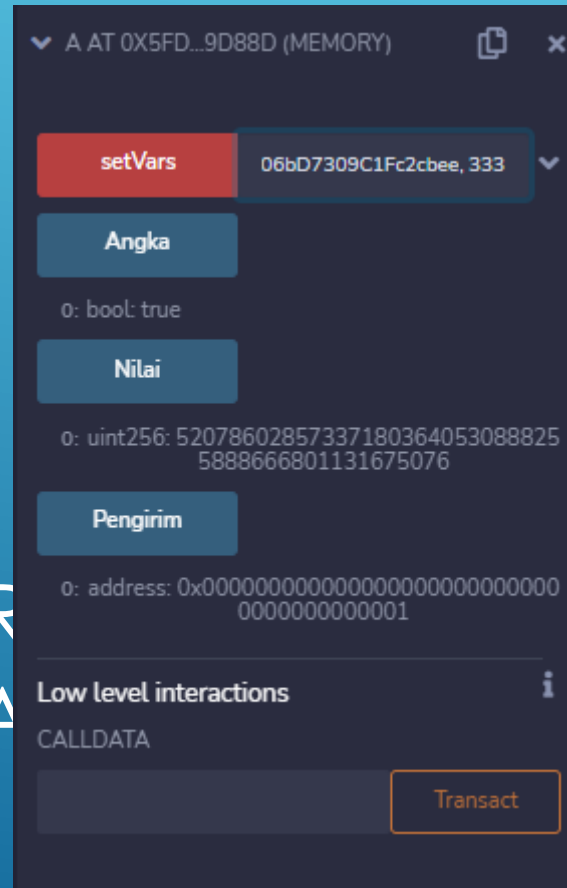


SEKARANG COBA KITA UBAH TIPE
VARIABLE ANGKA PADA CONTRACT A
JADI BOOLEAN

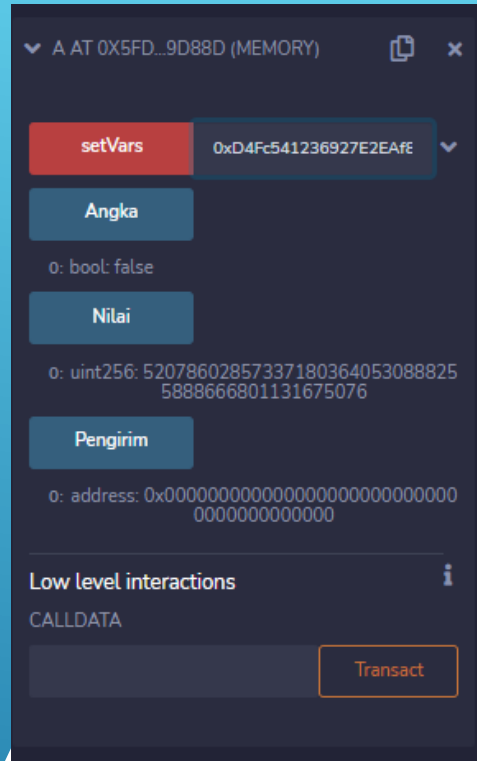


KITA DEPLOY KEMBALI CONTRACT A DAN
ISI KEMBALI ADDRESS CONTRACT B SERTA
SETVARS YANG KITA INGINKAN

KLIK ANGKA, PENGIRIM
CONTRACT A, MAKA
KARENA BOOLEAN



DI
AN TRUE



JIKA KITA ISI 0 PADA SETVARSNYA, MAKA
AKAN BERNILAI FALSE KARENA PADA
CONTRACT B VALUENYA ADALAH
INTEGER/ANGKA JADI SELAIN 0 AKAN
BERNILAI TRUE, SEDANGKAN 0 FALSE