

LAB 1: Deposit/Withdraw Ether

1.1 LAB: Smart contract Self Managing Funds

Pada lab ini kita akan mempelajari tentang bagaimana cara membuat smart contract yang akan mengatur keuangan anda. Anda akan mengirim Ether pada smart contract anda, lalu smart contract akan mengatur Ethernya dan dapat dikirim kepada siapapun. Ini seperti akun bank dengan code programming di dalamnya

Ini pun dapat digunakan sebagai escrow ether kedalam smart contract. Pertama yang kita perlukan adalah contoh setor dan Tarik tunai secara simple, lalu saya akan menunjukan kepada anda bagaimana smartcontract dapat mengunci sebuah dana menggunakan aktivasi waktu.

1.1.1 Yang anda ketahui pada akhir lab ini

Mengetahui contract address dan sebuah global msg-object

Bagaimana smart contract mengatur pendanaan

Bagaimana untuk mengirim dan mengambil Ether dari dan ke smart contract

1.1.2 Yang anda butuhkan

1. Browser
2. Koneksi koneksi
3. Waktu

1.2 Smart Contract

Mari kita buat smart contract yang simple dengan membuat file di remix dan paste kode dibawah ini

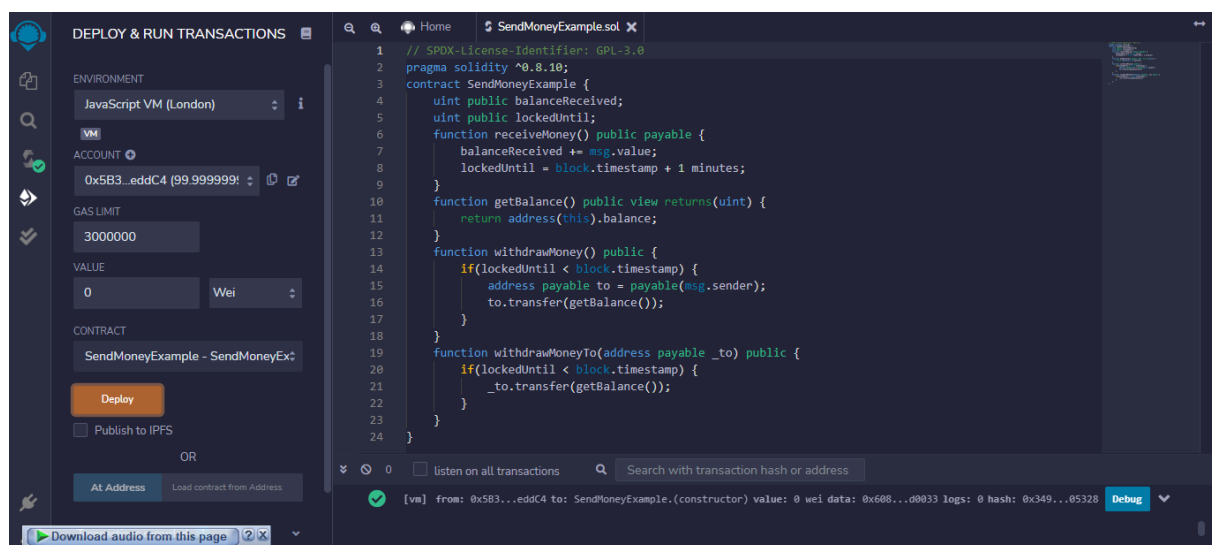
(SendMoneyExample.sol)

uint public balanceReceived : sebuah variable public storage

balanceReceived += msg.value : msg-object adalah object global yang selalu ada dengan berisi beberapa informasi tentang transaksi yang berlangsung

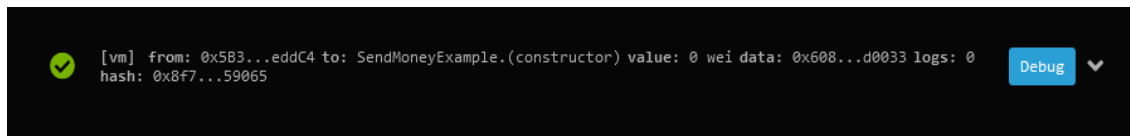
function getBalance() public view returns(uint): view function adalah sebuah fungsi yang tidak alter terhadap penyimpanan (read-only) dan dapat mengembalikan informasi

address(this).balance : sebuah variable pada tipe alamat yang selalu mempunyai property yang disebut .balance yang dimana memberikan anda jumlah dari ether yang disimpan pada alamat tersebut.



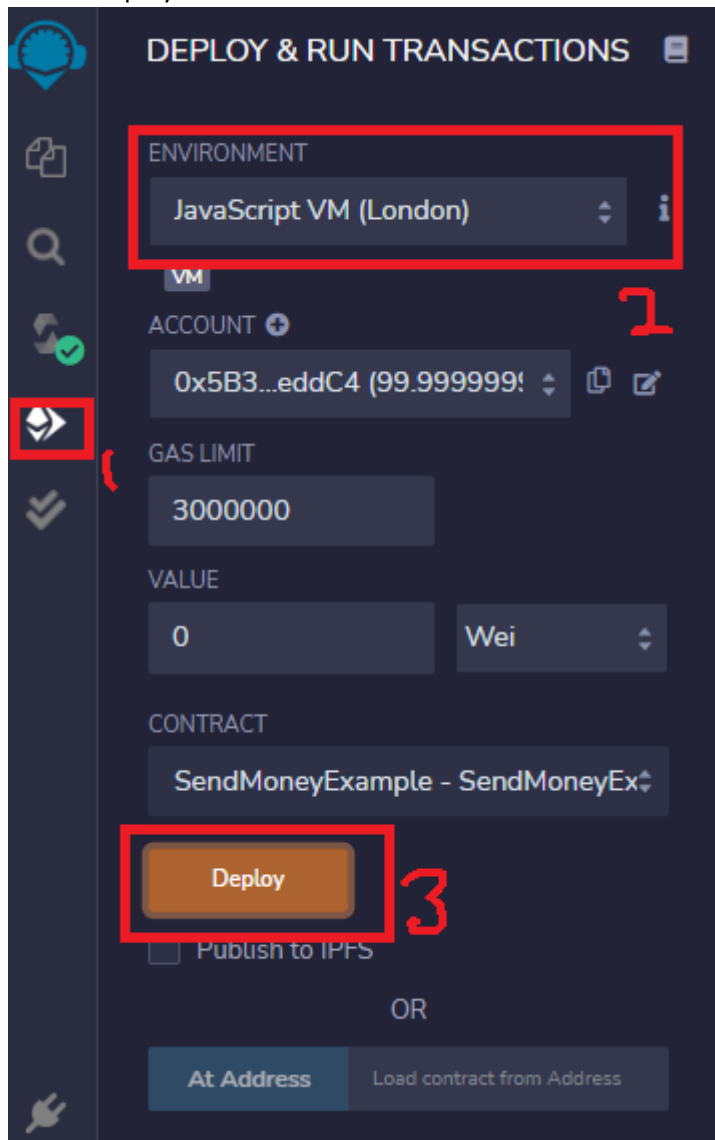
1.3 Mendeploy dan Menggunakan Smart Contract

Pertama-tama kita harus mendeploy smart contract kita. Lalu kita dapat melihat jika kita dapat menyimpan ether dan mendapatkan balance ether kita dari smart contract.



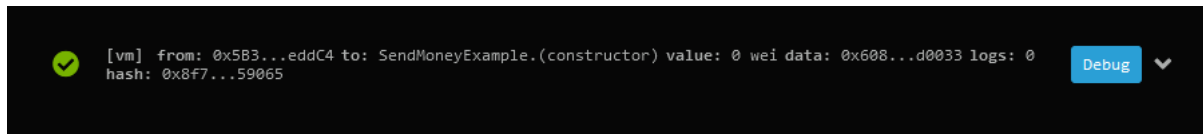
1.3.1 Mendeploy Smart Contract

Klik ke Deploy and Run Transaction



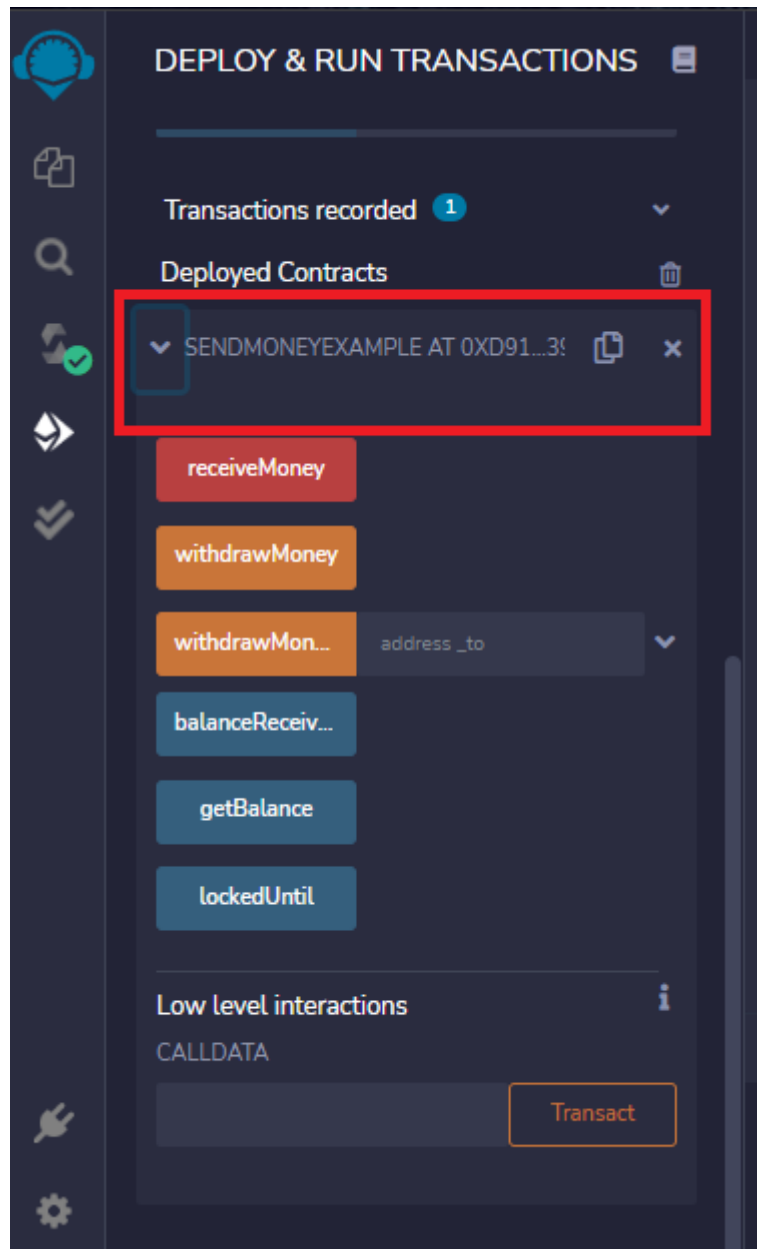
Pada tab terminal kita dapat lihat status deploy smart contract kita apakah berhasil atau gagal.

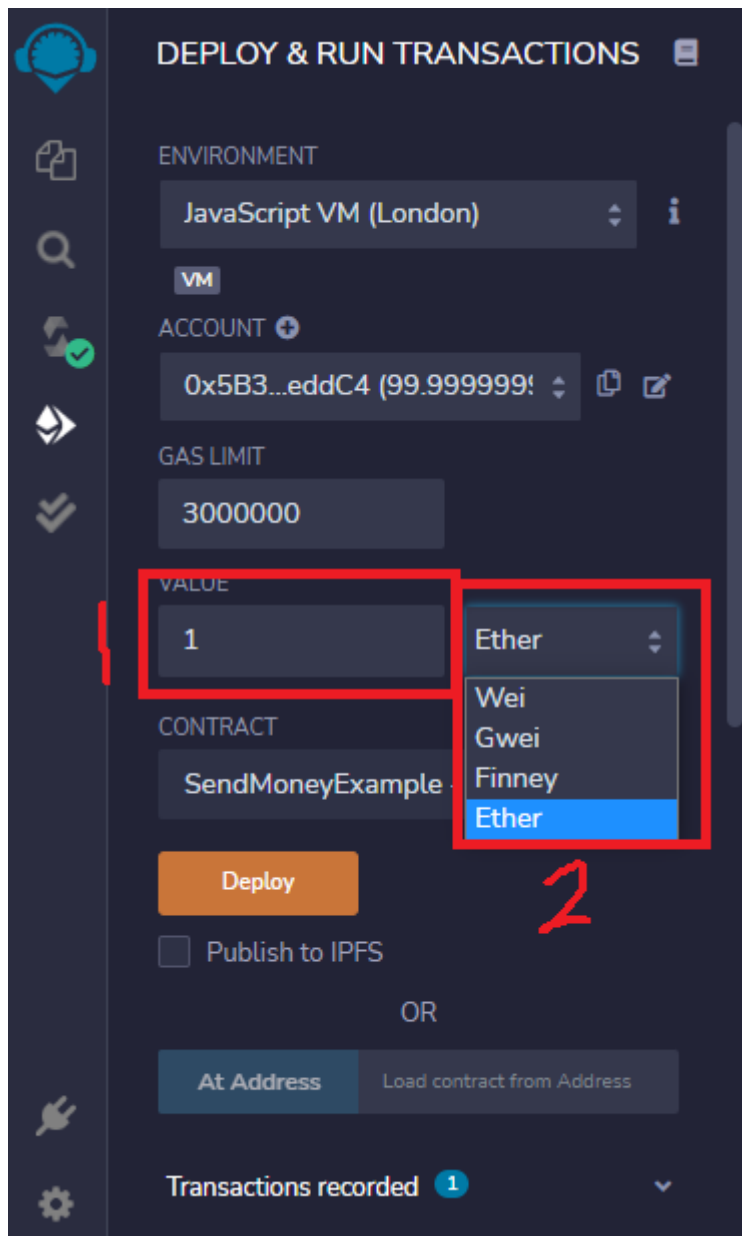
Untuk melihat smart contract yang telah terdeploy kita klik pada tab deployed contracts



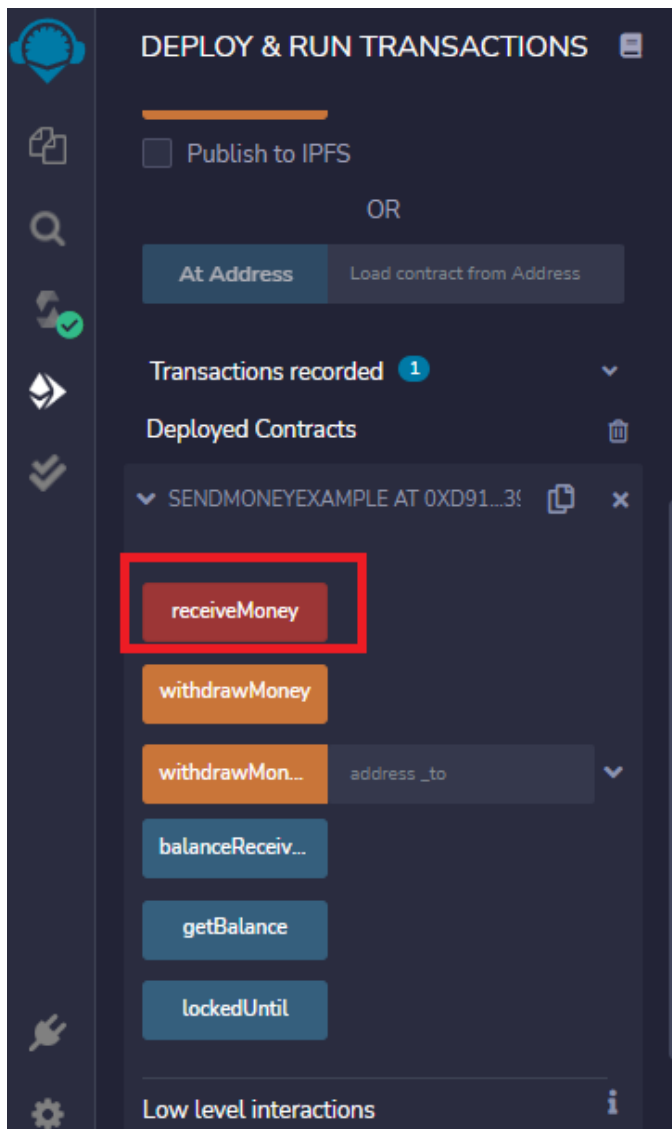
1.3.2 Mengirim Ether ke Smart Contract

Untuk dapat mengirim ether ke dalam Smart Contract kita harus memasukan value yang akan kita kirim ke alamat Ethereum kita. Pertama scroll keatas dan kita dapat lihat value , lalu rubah nilai value menjadi 1 dan rubah menjadi ether.

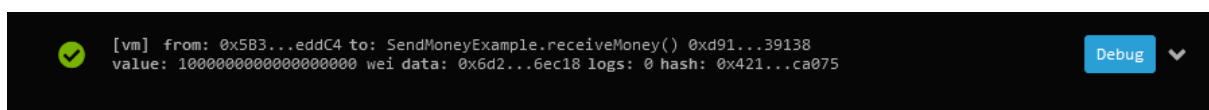




Lalu scroll kebawah ke deployed contract dan kita dapat melihat dan klik tombol merah "receiveMoney"

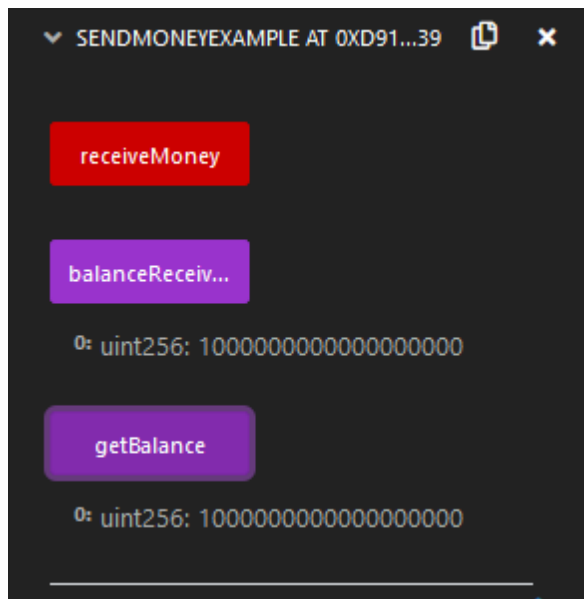


Pada kolom terminal kita dapat lihat keberhasilan dari transaksi kita



1.3.3 Cek Saldo di Smart Contract

Pada tahapan sebelumnya kita mengirim 1 Ether atau 10^{18} Wei ke address kita, berdasarkan smartcontract yang kita telah buat sebelumnya, untuk mendapatkan nilai yang telah di terima kita dapat mengklik "balanceReceive" dan untuk melihat total saldo kita dapat mengklik "getBalance"



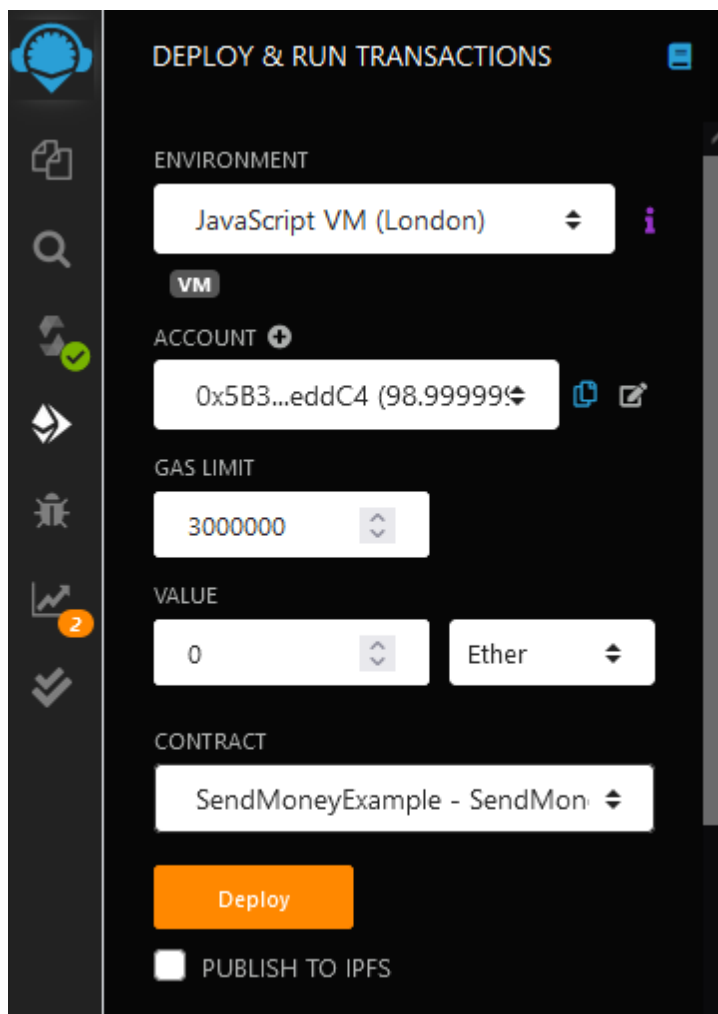
1.4 Menarik Ether dari Smart Contract

Untuk dapat menarik saldo Ether dari address kita, maka tahapan selanjutnya adalah menambahkan line baru untuk fungsi penarikan ether. Tambahkan kode dibawah ini kedalam kode smartcontract yang sebelumnya kita buat.

```
function withdrawMoney() public {  
    if(lockedUntil < block.timestamp) {  
        address payable to = payable(msg.sender);  
        to.transfer(getBalance());  
    }  
}
```

fungsi ini bertugas untuk mengirim semua saldo kita ke smart contract ke orang yang diberinama sebagai fungsi “withdrawMoney()”

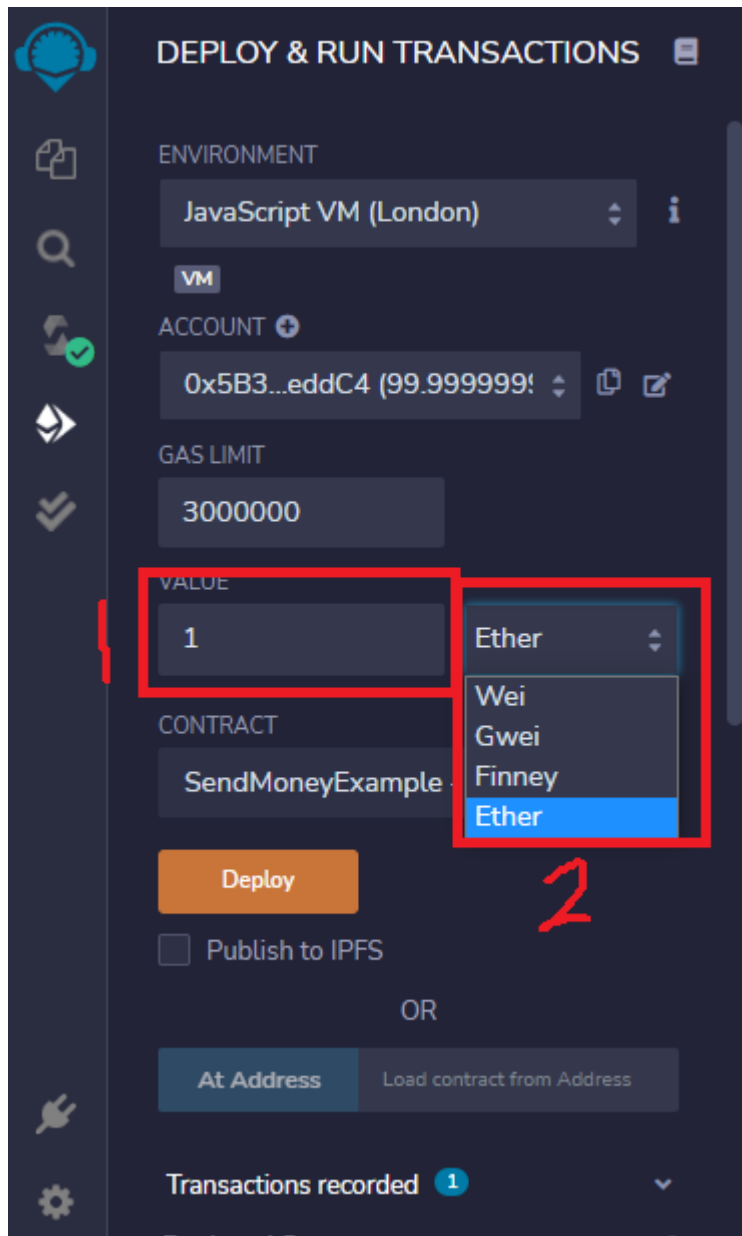
1.4.1 Deploy Smart Contract yang Terbaru



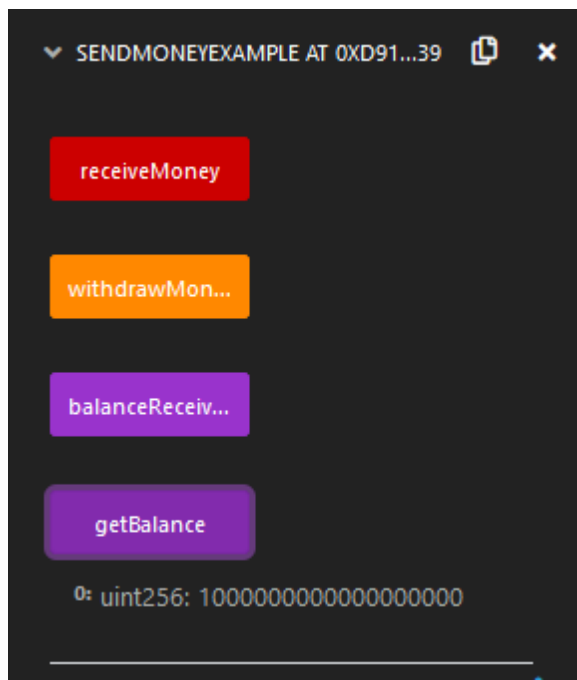
setelah smart contract kita terdeploy, Langkah selanjutnya adalah memastikan di terminal status deploy smart contract kita

```
[vm] from: 0x5B3...eddC4 to: SendMoneyExample.(constructor) value: 0 wei data: 0x608...d0033 logs: 0  
hash: 0x7d9...716d8
```

lalu kita masukan lagi 1 Ether ke address kita seperti sebelumnya

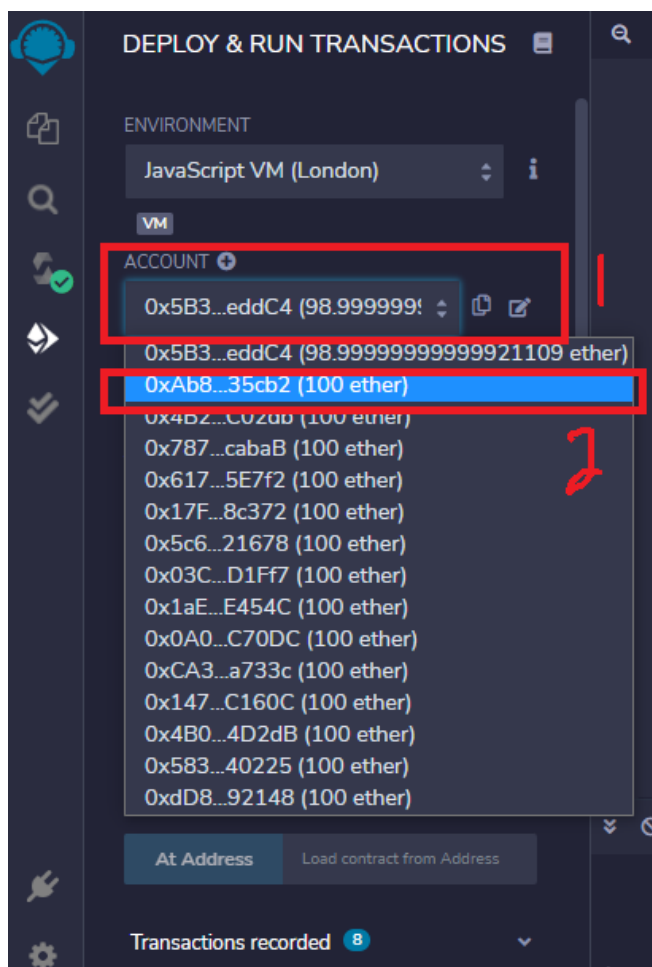


```
[vm] from: 0x5B3...eddC4 to: SendMoneyExample.receiveMoney() 0xd91...39138  
value: 1000000000000000000 wei data: 0x6d2...6ec18 logs: 0 hash: 0x421...ca075
```

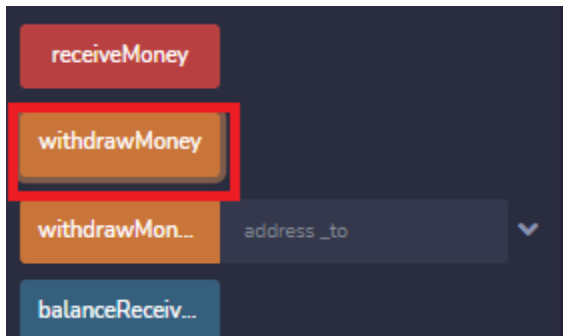



1.4.2 Menarik Saldo dari Smart Contract

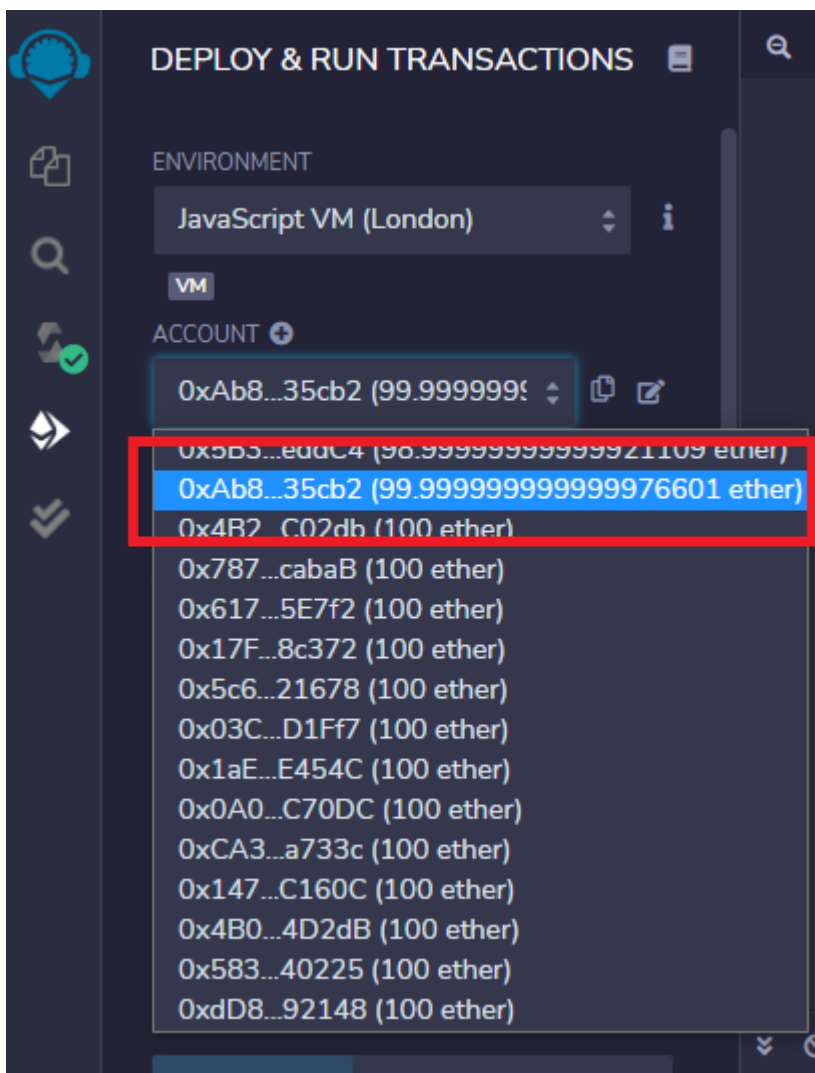
Pada tahapan ini kita akan menarik saldo dari alamat akun yang berbeda



Lalu kita klik tombol withdrawMoney



Disini sini kita dapat cek pada account Ethereum kita bahwa akun ke dua kita bertambah nilai ethernya



Pada gambar diatas terdapat benar bahwa ether kita terkirim ke akun ke du akita, tetapi mengapa tidak menjadi 101 ether di akun kita? Pada jaringan Ethereum dikenal dengan konsep gas, konsep ini dipakai karena setiap ada transaksi keluar dan kedalam, kita membutuhkan miner untuk memining block kita, maka dari itu gas ini adalah sebagai upah kepada miner yang sudah memining block , untuk harga dari gas sendiri sangatlah fluktuaktif, dikarenakan menyesuaikan banyak transaksi, kepadatan jaringan, dan jam transaksi pada saat itu

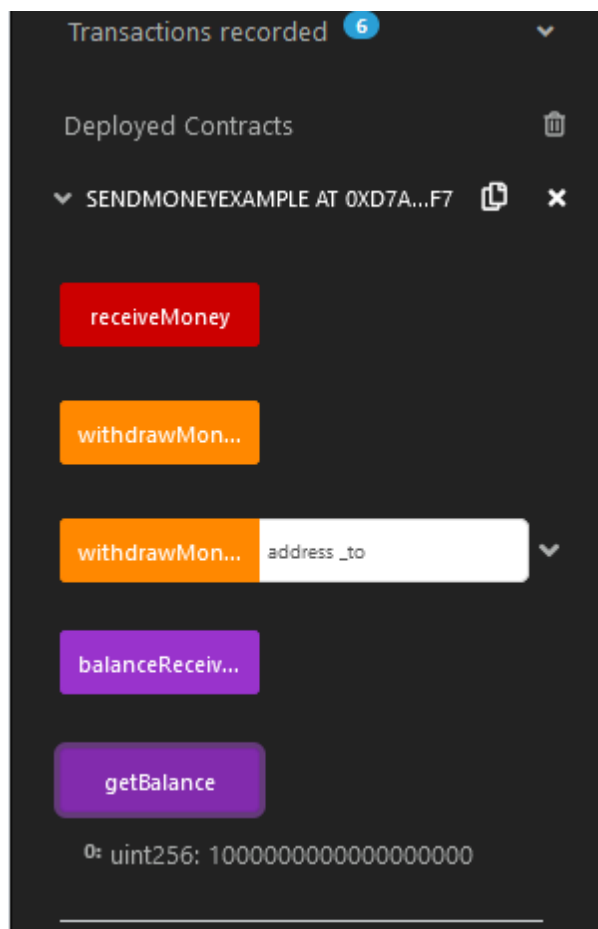
1.5 Menarik Saldo ke Akun Spesifik

Sebelumnya kita mempelajari tentang smartcontract kita dengan mengirim ether degan menggunakan fungsi “withdrawMoney”. Pada kali ini kita akan menambahkan satu fungsi “withdrawMoneyTo(address payable_to) public” dan dapat menambahkan line pada dibawah ini ke kode smartcontract kita.

```
function withdrawMoneyTo(address payable _to) public {
    if(lockedUntil < block.timestamp) {
        _to.transfer(getBalance());
    }
}
```

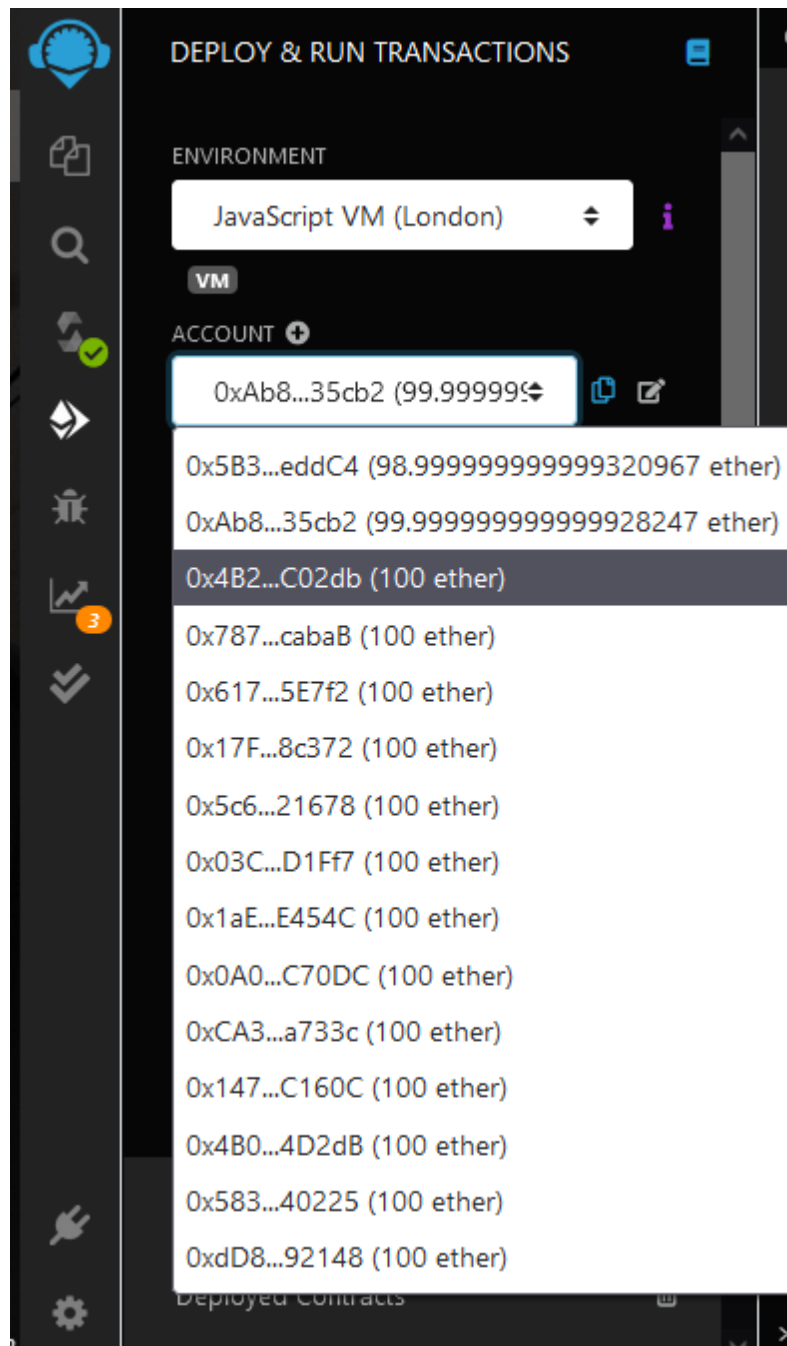
1.5.1 Deploy Kembali Smart Contract Kita

Deploy Kembali smart contract kita dan kirim satu ether ke akun dan cek Kembali apakah ether sudah di terima

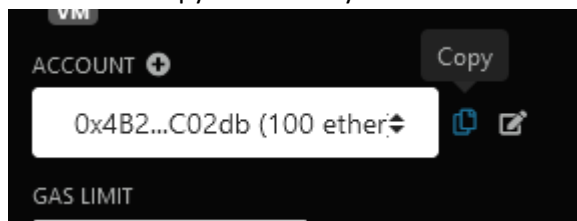


1.5.2 Test fungsi “withdrawMoneyTo”

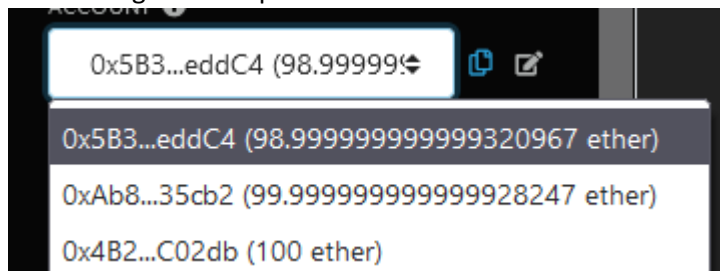
- Gunakan akun yang ketiga pada list akun kita



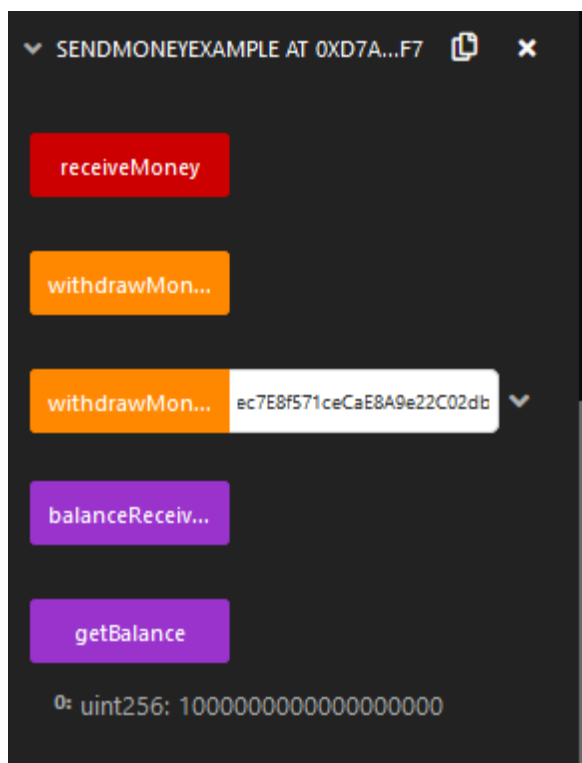
2. Klik tombol copy untuk menyalin address akun kita



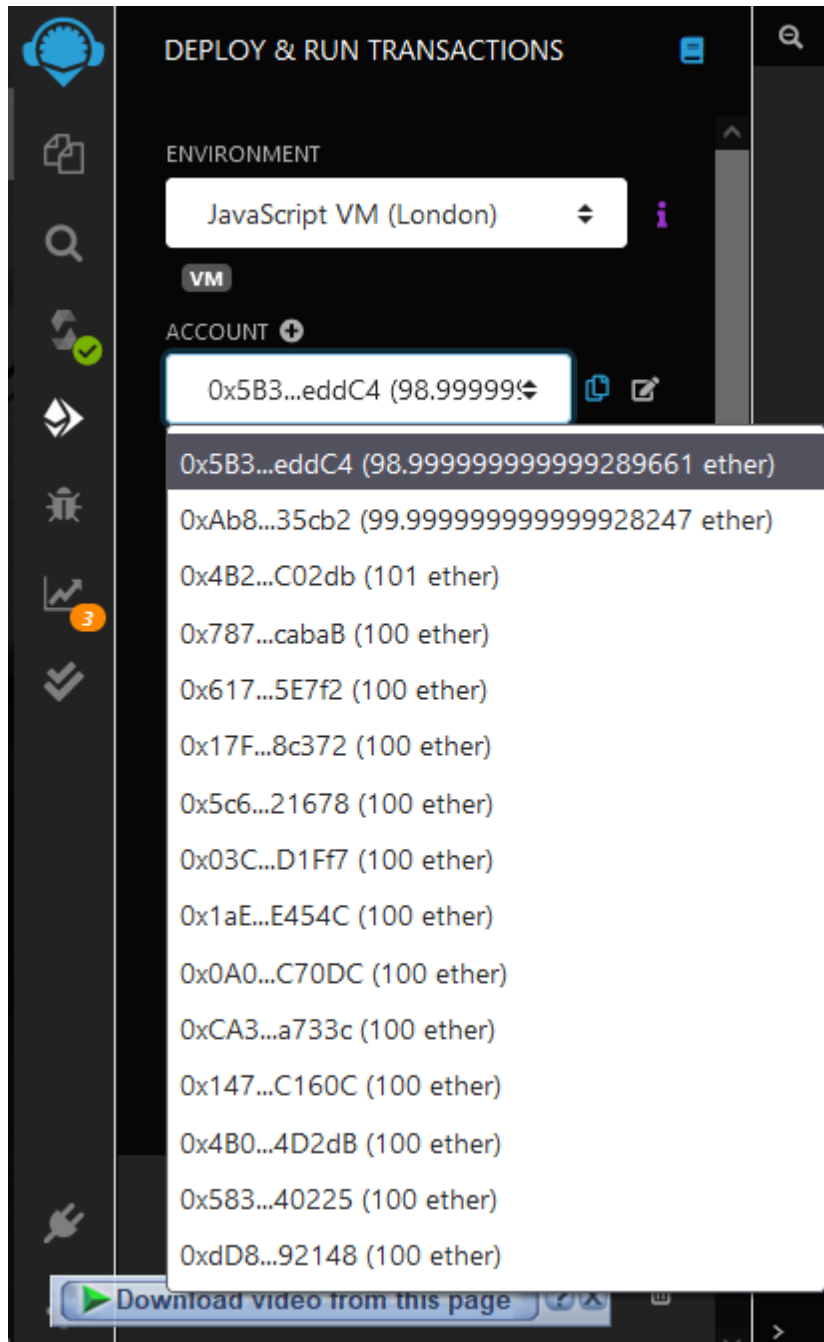
3. Kembali lagi ke akun pertama kita



4. Masukkan alamat akun yang kita telah salin sebelumnya ke kolom "withdrawMoneyTo" dan klik tombol "withdrawMoneyTo"



5. Lihat pada bagian akun dan lihat ada kejanggalan bahwa nilai ether yang diterima pas 101 ether



Mengapa bisa 101 ether? Dikarenakan kita mengirim dari akun 1 ke smartcontract, dengan menginstruksikan smartcontract untuk mengirimkan semua saldo ke alamat smartcontract ke akun ke tiga anda di list akun. Pembayaran fees dibayarkan oleh akun pertama jadinya akun ke 3 menerima 1 full ether.

1.6 Penguncian Pengambilan Saldo

Yang kita butuhkan adalah menyimpan "block.timestamp" di suatu tempat. Ada beberapa metode untuk melakukan ini, saya lebih suka membiarkan pengguna tahu berapa lama itu terkunci. Jadi, alih-alih menyimpan stempel waktu setoran, saya akan menyimpan stempel waktu yang terkunci.

```
1 // SPDX-License-Identifier: GPL-3.0
2 pragma solidity ^0.8.10;
3 contract SendMoneyExample {
4     uint public balanceReceived;
5     uint public lockedUntil;
6     function receiveMoney() public payable {
7         balanceReceived += msg.value;
8         lockedUntil = block.timestamp + 1 minutes;
9     }
10    function getBalance() public view returns(uint) {
11        return address(this).balance;
12    }
13    function withdrawMoney() public {
14        if(lockedUntil < block.timestamp) {
15            address payable to = payable(msg.sender);
16            to.transfer(getBalance());
17        }
18    }
19    function withdrawMoneyTo(address payable _to) public {
20        if(lockedUntil < block.timestamp) {
21            _to.transfer(getBalance());
22        }
23    }
24 }
```

1.6.1 Deploy dan Test Smart Contract