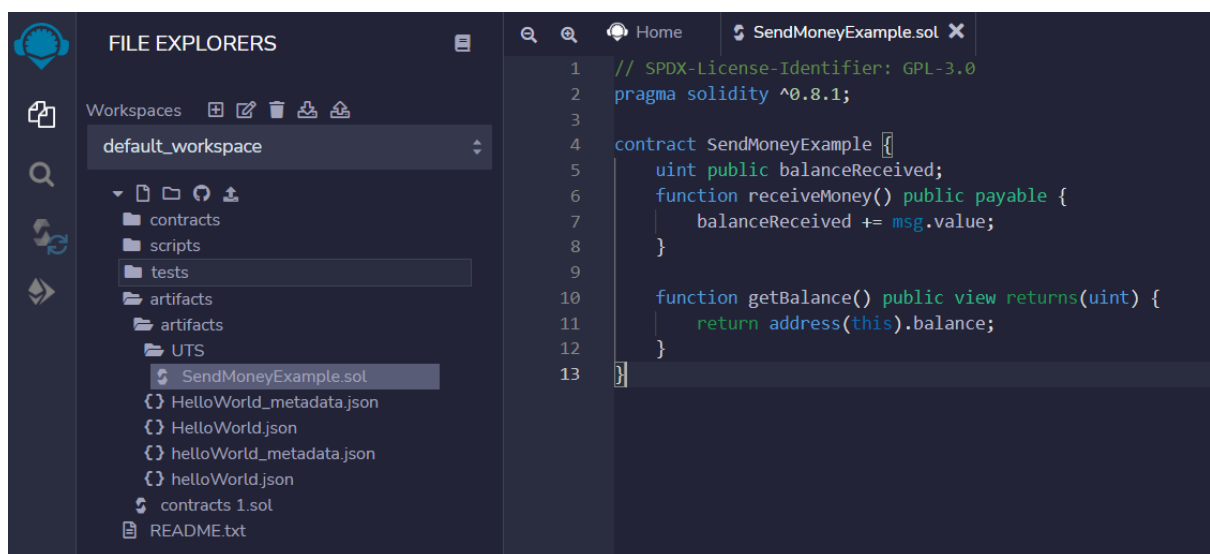


```
// SPDX-License-Identifier: GPL-3.0
pragma solidity ^0.8.1;

contract SendMoneyExample {
    uint public balanceReceived;
    function receiveMoney() public payable {
        balanceReceived += msg.value;
    }

    function getBalance() public view returns(uint) {
        return address(this).balance;
    }
}
```



Penjelasan per line

uint public balanceReceived : adalah variable penyimpanan umum / public. Variabel public akan membuat fungsi pengambil secara otomatis di Solidity. Jadi developer selalu dapat memonitor konten yang sedang ada pada variable tersebut.

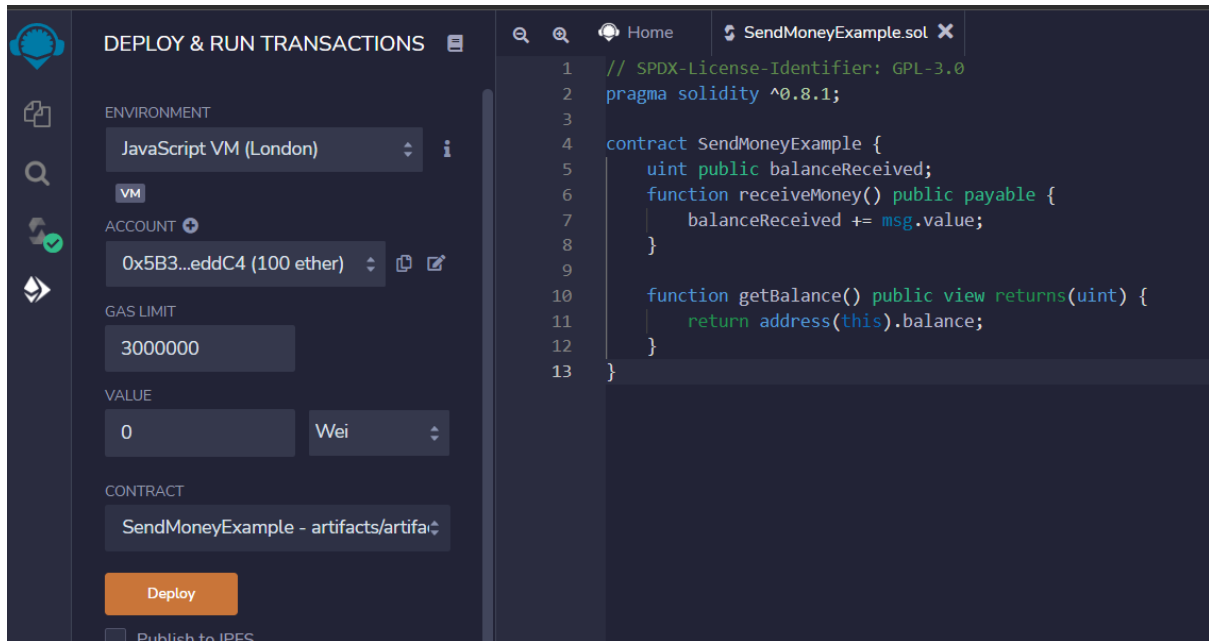
balanceReceived += msg.value : msg adalah objek global yang selalu ada yang berisi beberapa informasi tentang transaksi yang sedang berlangsung. Property yang paling penting disini adalah *.value* dan *.sender*.

function getBalance() public view returns(uint) : fungsi view adalah fungsi yang tidak mengubah penyimpanan (read-only) dan dapat mengembalikan informasi. Tidak perlu ditambang.

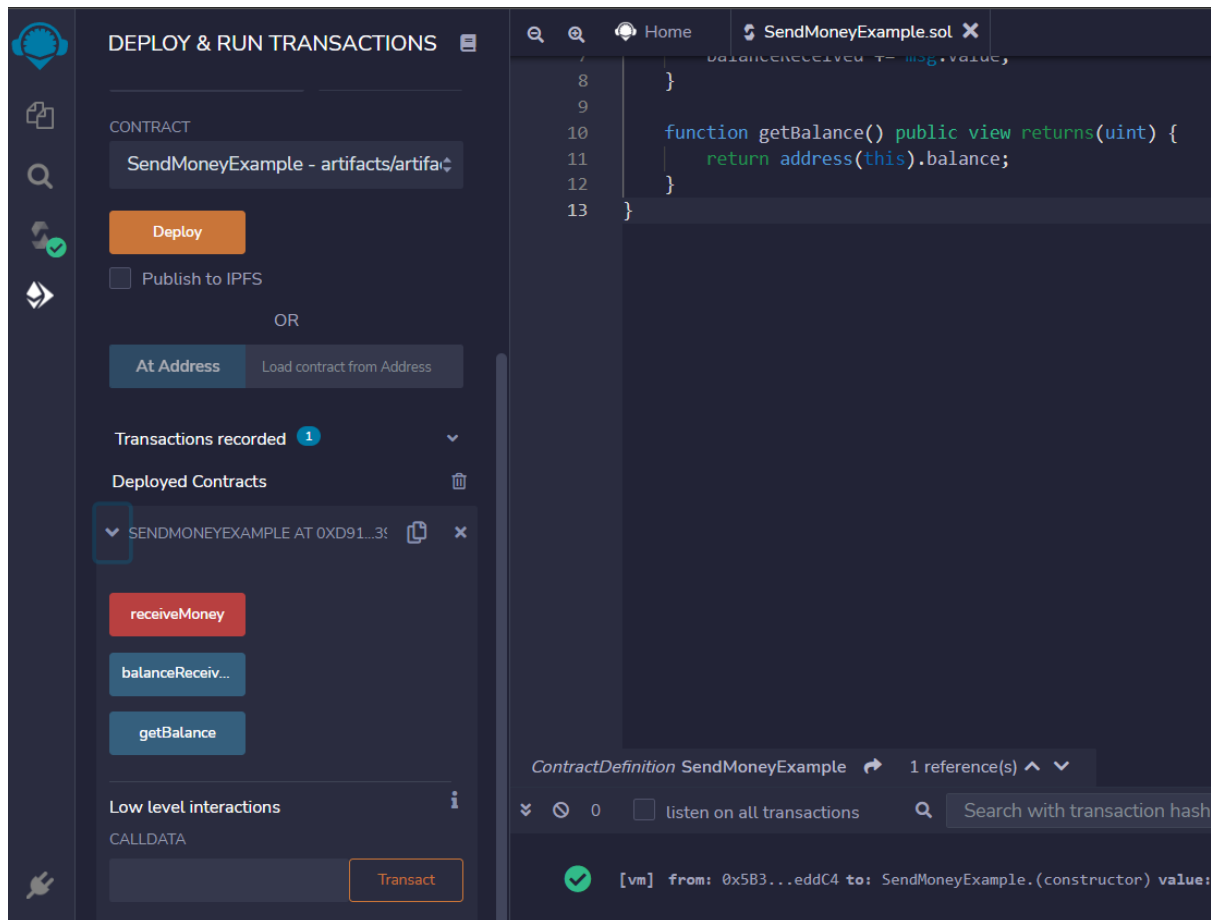
address(this).balance : variable dari tipe alamat selalu memiliki property yang disebut *.balance* yang memberi sejumlah Ether yang disimpan di alamat tersebut. Hal ini bukan berarti kita dapat mengakses secara mudah, namun hanya memberitahu berapa banyak yang disimpan disana. *address(this)* mengonversi Smart Contract ke alamat. Jadi, line ini pada dasarnya mengembalikan jumlah Ether yang disimpan di Smart Contract itu sendiri.

Deploy the Smart Contract

Buka Plugin Deploy dan Run Transactions dan terapkan Smart Contract ke dalam JavaScript VM:

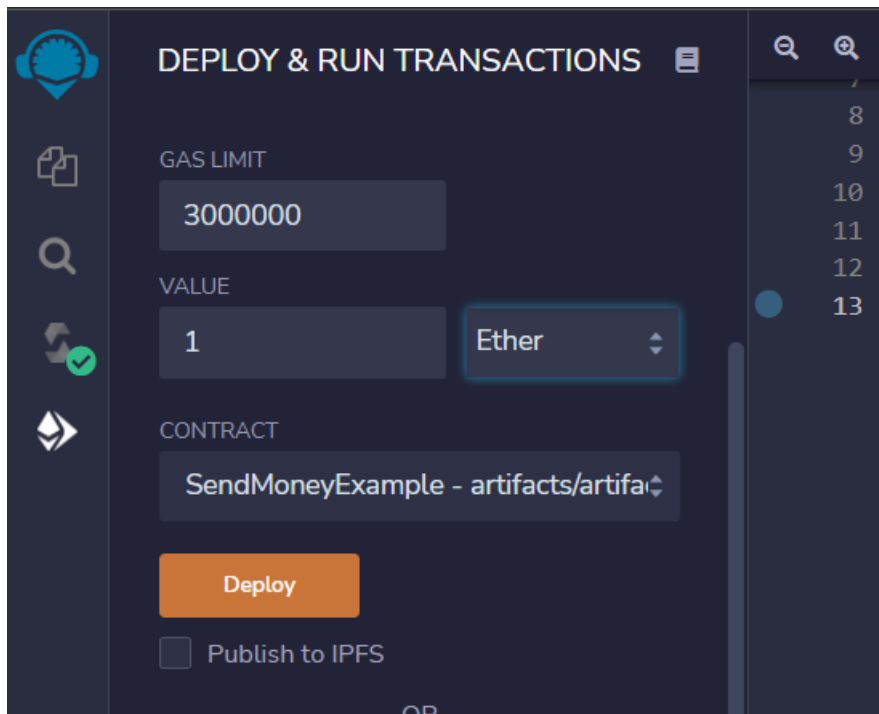


Klik button Deploy pojok kiri bawah

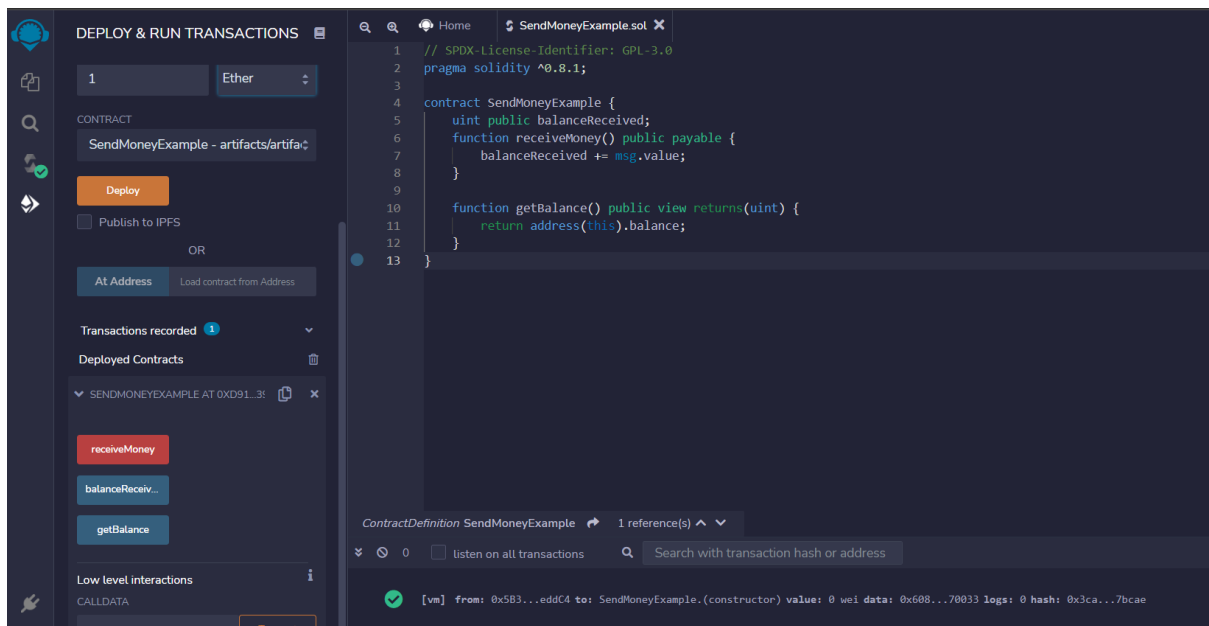


Mengirim beberapa Ether ke Smart Contract

Scroll up ke bagian "value" dan masukkan "1" ke bagian input setelah itu pilih "Ether" dari dropdown



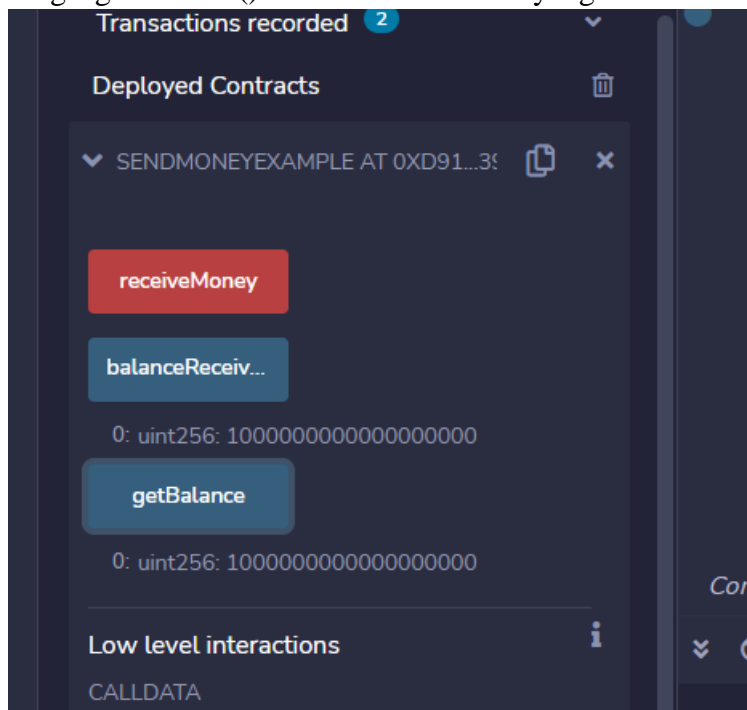
Kemudian scroll ke bawah ke Smart Contract dan tekan tombol merah “receiveMoney”



Amati juga terminal, lihat ada transaksi baru yang dikirim ke “The Network” (Ini hanya simulasi, namun sama dengan blockchain nyata).

Cek Balance

Untuk dapat mengirim Ether atau Wei ke Smart Contract, variable `balanceReceived` dan fungsi `getBalance()` harus memiliki nilai yang sama.



Bagaimana cara mengeluarkan Ether lagi?? Mari kita tambahkan metode Withdrawal sederhana.

Withdraw Ether dari Smart Contract

Tambahkan fungsi Withdraw

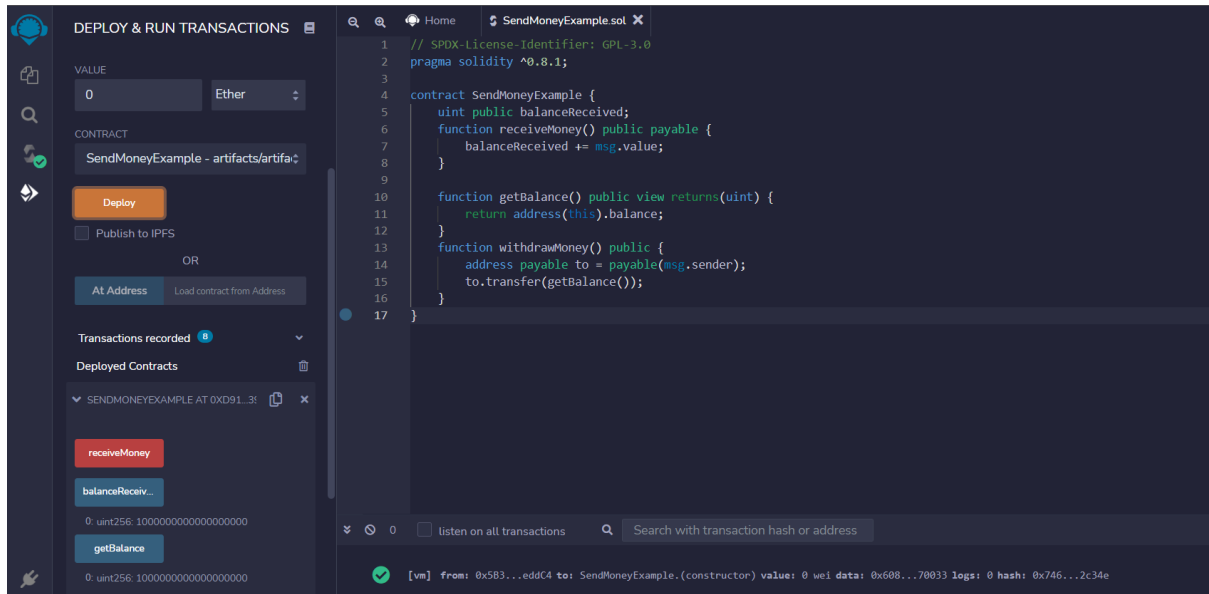
```
// SPDX-License-Identifier: GPL-3.0
pragma solidity ^0.8.1;

contract SendMoneyExample {
    uint public balanceReceived;
    function receiveMoney() public payable {
        balanceReceived += msg.value;
    }

    function getBalance() public view returns(uint) {
        return address(this).balance;
    }
    function withdrawMoney() public {
        address payable to = payable(msg.sender);
        to.transfer(getBalance());
    }
}
```

Fungsi ini akan mengirim semua danayang tersimpan di Smart Contract ke orang yang memanggil fungsi “*withdrawMoney()*”

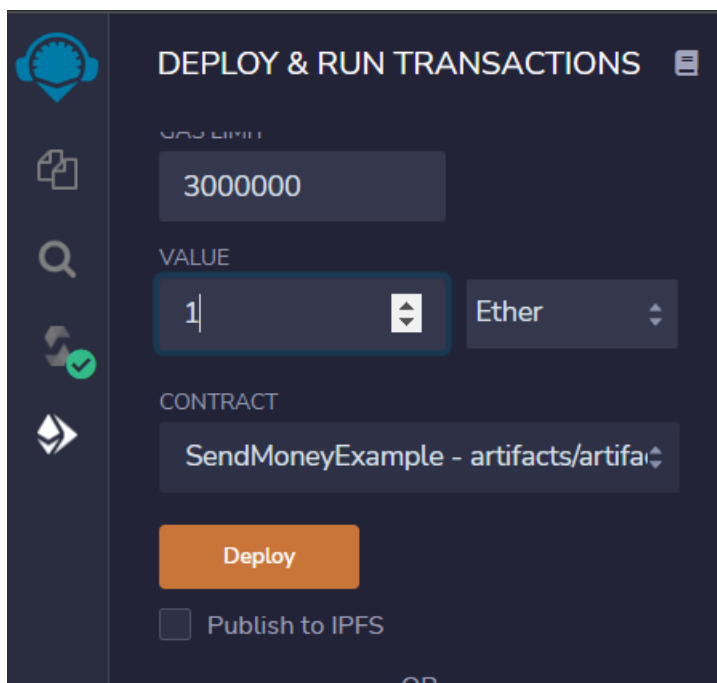
Deploy Smart Contract baru

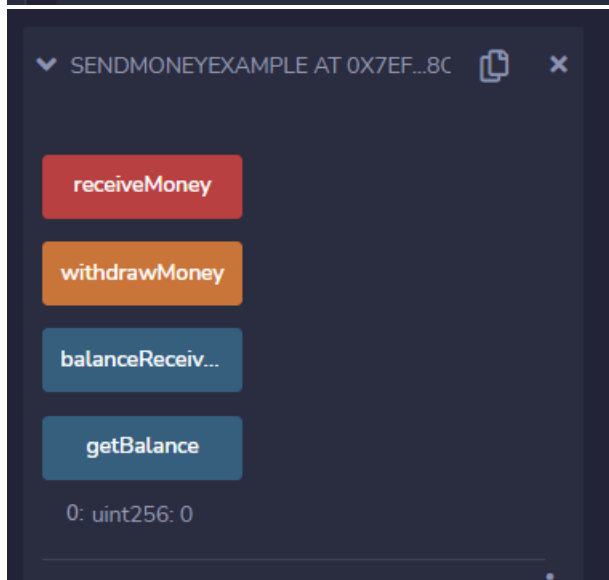
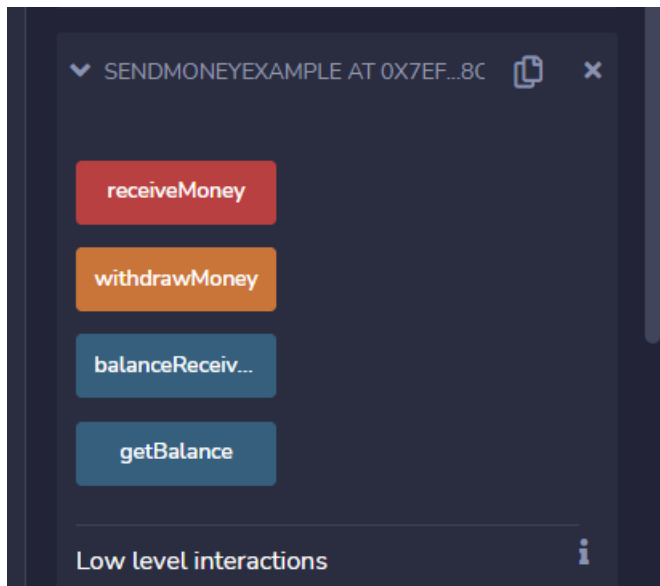


Lakukan hal yang sama dengan tahap sebelumnya namun dengan code yang baru (dengan fungsi withdraw)

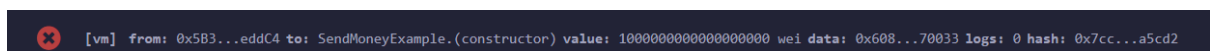
Masukkan "1 Ether" ke value input box

Tekan "receiveMoney" di Smart Contract baru





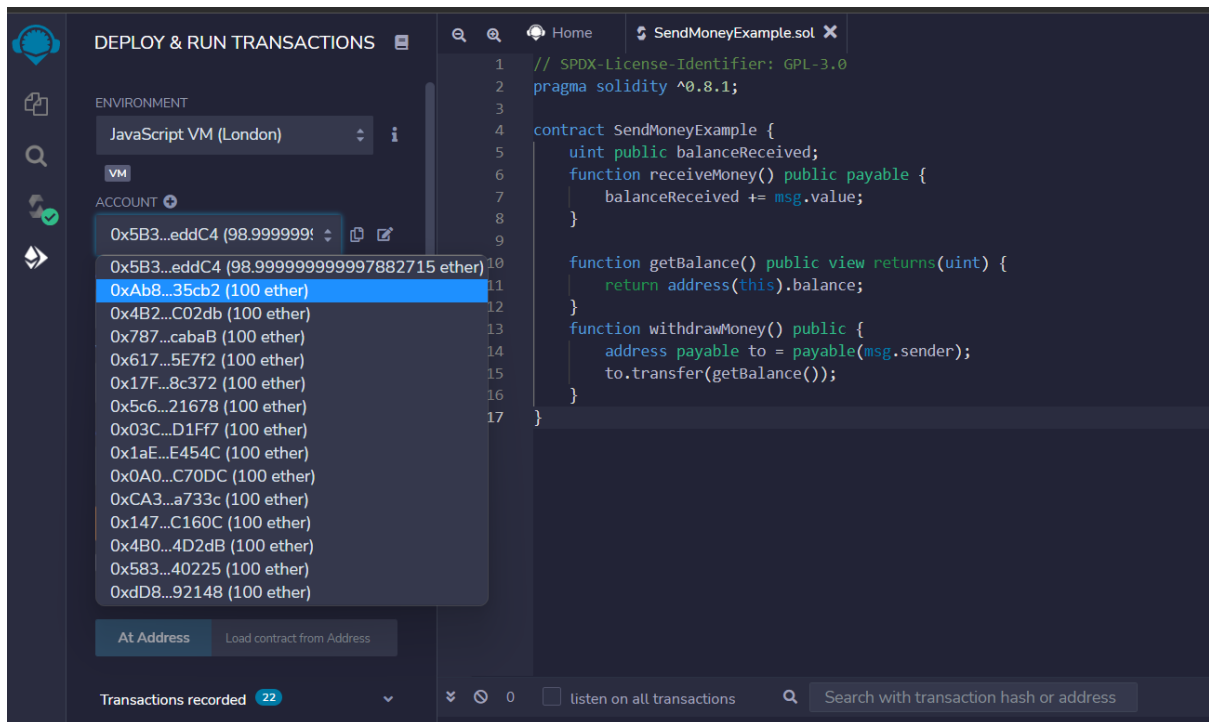
Tidak muncul jumlah saldo yang harusnya 100xxxxx seperti pada transaksi sebelumnya, karena pada saat saya input value 1 Ether, muncul error



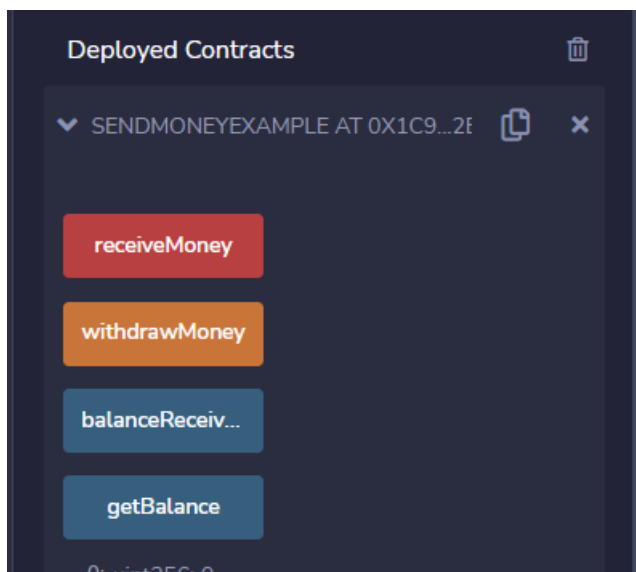
Namun setidaknya seperti itulah cara cek balance withdraw-nya.

Withdraw Dana dari Smart Contract

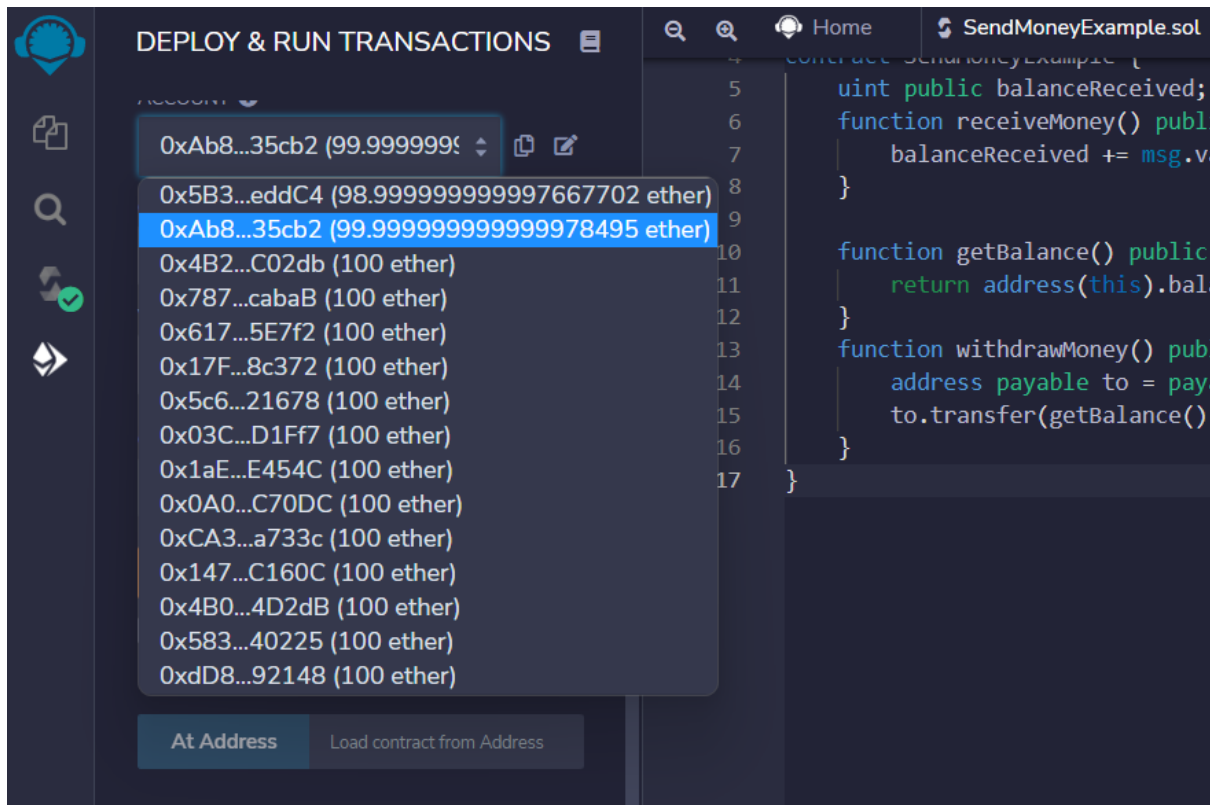
Pilih akun kedua dari dropdown akun untuk membuat perbedaan dengan akun yang sebelumnya kita pakai



Tekan button “withdrawMoney”:



Amati jumlah Ether yang terdapat pada akun kedua yang anda miliki



Mengapa tidak terdapat 101 Ether di akun ini? Padahal sebelumnya terdapat 100 Ether dan sekarang sudah saya tambahkan 1 Ether, bukankah harusnya 101 Ether?

Mari kita bahas !!!

Kita tahu bahwa dalam transaksi Ethereum terdapat Gas Fee. Berapa Gas yang Anda bayar, Anda bertanya-tanya? Kami membahas ini - secara mendalam.

Mari kita ke fungsi lain, yang memungkinkan kita mengirim jumlah penuh ke Alamat tertentu! Itu akan tetap tidak aman, tapi setidaknya mengajarkan konsep baru - satu per satu!

Withdraw ke Akun Tertentu

Sebelumnya kami dapat secara bebas mengirim Ether ke Smart Contract yang memiliki fungsi “withdrawMoney”. Bagaimana membuat dana dapat ditarik ke akun / rekening tertentu.

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity ^0.8.1;

contract SendMoneyExample {
    uint public balanceReceived;
    function receiveMoney() public payable {
        balanceReceived += msg.value;
    }

    function getBalance() public view returns(uint) {
```

```

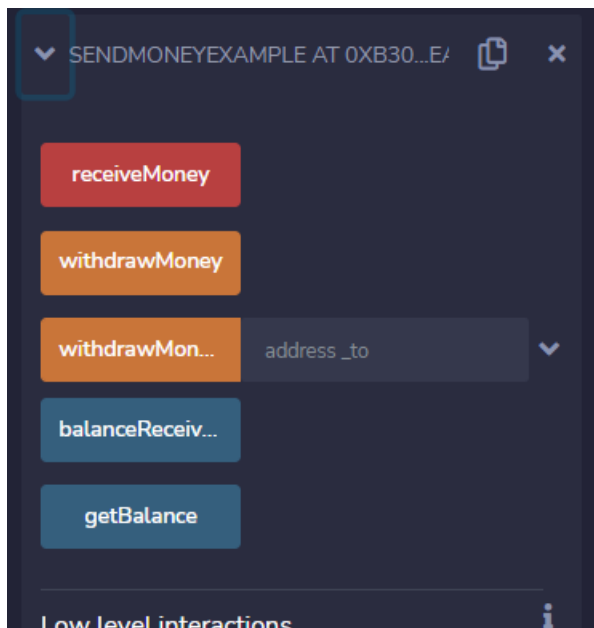
    return address(this).balance;
}
function withdrawMoney() public {
    address payable to = payable(msg.sender);
    to.transfer(getBalance());
}
function withdrawMoneyTo(address payable _to) public {
    _to.transfer(getBalance());
}
}

```

Sekarang kami dapat menentukan alamat tujuan pengiriman dana.

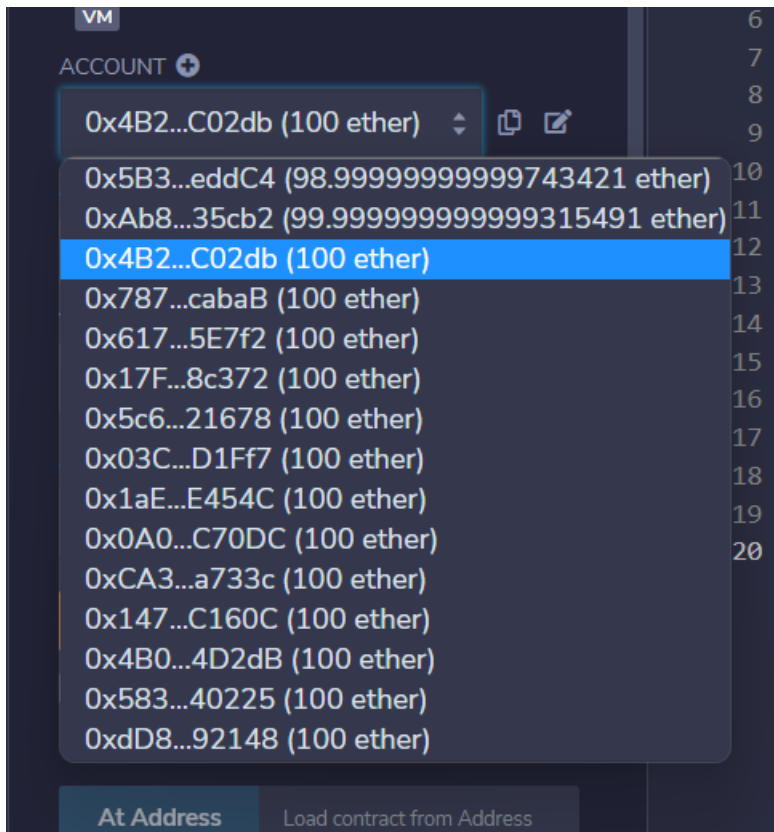
Deploy ulang Smart Contract kita:

1. Deploy Smart Contract
2. Tutup Instance sebelumnya
3. Kirim 1 Ether ke Smart Contract
4. Pastikan Balance menunjukkan nilai yang benar

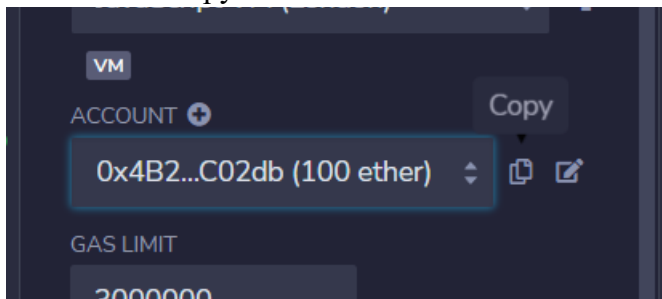


Tes fungsi “withdrawMoneyTo”

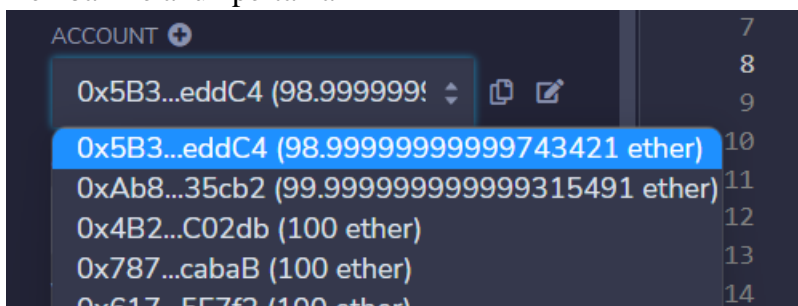
1. Pilih akun ketiga dari dropdown akun



2. Tekan ikon “copy”



3. Kembali ke akun pertama



4. Paste akun yang sudah di copy ke bagian input sebelah “withdrawMoneyTo”


```

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

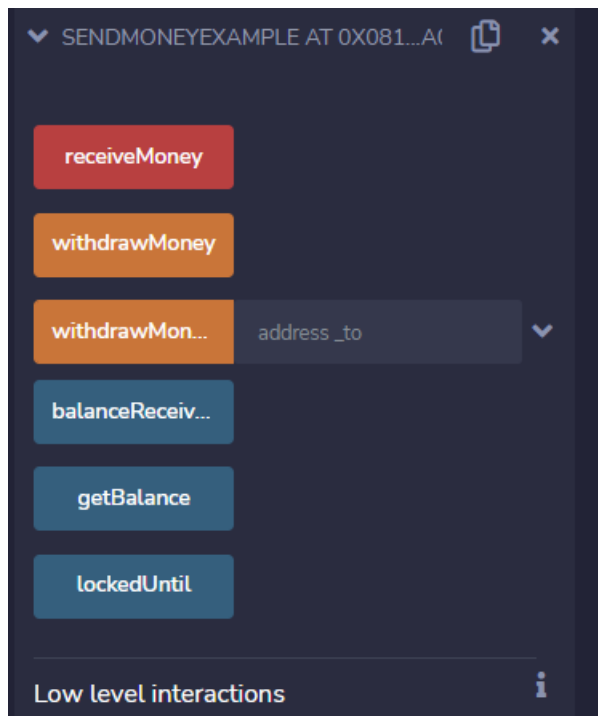
```

Deploy dan tes Smart Contract

1. Deploy instance versi baru
2. Hapus instance lama
3. Kirim 1 Ether ke Smart Contract (jangan lupa bagian value) dengan menekan "receiveMoney"
4. Cek balance

Praktik

Klik "withdrawMoney" - dan tidak ada yang terjadi. Saldo tetap sama sampai 1 Menit berlalu sejak Anda menekan "receiveMoney".



SENDMONEYEXAMPLE AT 0X081...A0

receiveMoney

withdrawMoney

withdrawMon...

address_to

balanceReceiv...

getBalance

0: uint256: 0

lockedUntil

0: uint256: 0

Low level interactions