// 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;

    }

}

A screenshot of a computer

Description automatically generated with medium confidence

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:

A screenshot of a computer

Description automatically generated with medium confidence

Klik button Deploy pojok kiri bawah

A screenshot of a computer

Description automatically generated with medium confidence

**Mengirim beberapa Ether ke Smart Contract**

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

Graphical user interface, application, Teams

Description automatically generated

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

A screenshot of a computer

Description automatically generated with medium confidence

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. A screenshot of a computer

Description automatically generated with medium confidence

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**

A screenshot of a computer

Description automatically generated with medium confidence

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

Graphical user interface, application

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence Graphical user interface, application, website

Description automatically generated

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

A screenshot of a computer

Description automatically generated with medium confidence

Tekan button “withdrawMoney”:

A screenshot of a computer

Description automatically generated with medium confidence

Amati jumlah Ether yang terdapat pada akun kedua yang anda miliki

Graphical user interface, text, application, chat or text message

Description automatically generated

Mengapa tidak terdapat 101 Ether di akun ini? Padahal sebelumya 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 dapar 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 menunjukan nilai yang benar

Graphical user interface, application

Description automatically generated

**Tes fungsi “withdrawMoneyTo”**

1. Pilih akun ketiga dari dropwdown akun

A screenshot of a computer

Description automatically generated with medium confidence

1. Tekan ikon “copy”

A screenshot of a computer

Description automatically generated with medium confidence

1. Kembali ke akun pertama

Graphical user interface, text, application, chat or text message

Description automatically generated

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

Graphical user interface, diagram

Description automatically generated

1. Tekan button withdrawMoneyTo yang sudah diisi akun ketiga
2. Cek balance di akun ketiga

Graphical user interface, text

Description automatically generated

Seharusnya saldo pada akun ketiga 101 Ether, disini tetap 100 karena terdapat error yang sudah saya jelaskan sebelumnya.

**Withdrawal Locking**

Extend Smart Contract

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.

// SPDX-License-Identifier: GPL-3.0

pragma solidity ^0.8.1;

contract SendMoneyExample {

    uint public balanceReceived;

    uint public lockedUntil;

    function receiveMoney() public payable {

        balanceReceived += msg.value;

        lockedUntil = block.timestamp + 1 minutes;

    }

    function getBalance() public view returns(uint) {

        return address(this).balance;

    }

    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".

Graphical user interface, application

Description automatically generated

Graphical user interface

Description automatically generated