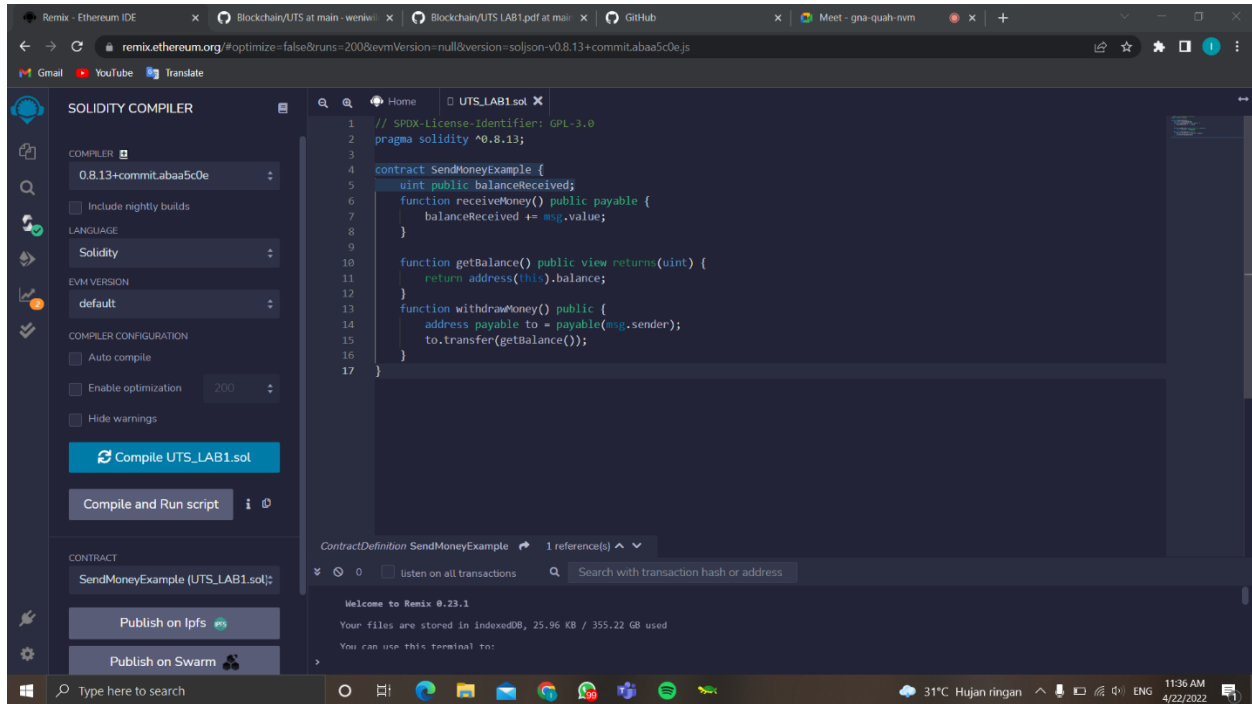


Nama: Irfan Ainul Afif Refnaldi

Nim: 1103184025

## Penjelasan UTS LAB 1



```
1 // SPDX-License-Identifier: GPL-3.0
2 pragma solidity ^0.8.13;
3
4 contract SendMoneyExample {
5     uint public balanceReceived;
6     function receiveMoney() public payable {
7         balanceReceived += msg.value;
8     }
9
10    function getBalance() public view returns(uint) {
11        return address(this).balance;
12    }
13    function withdrawMoney() public {
14        address payable to = payable(msg.sender);
15        to.transfer(getBalance());
16    }
17 }
```

1. Uint public balanceReceived merupakan sebuah variabel penyimpanan yang bisa bersifat umum dan publik yang akan membuat fungsi pengambil secara otomatis di solidity. Sehingga developer bisa memonitor konten pada varibel tersebut.
2. blanceReceived += msg.value adalah objek global yang berisi tentang sekumpulan informasi transaksi yang sedang berlangsung. Value dan sender merupakan properti yang paling penting.
3. Function getBalance() public view returns(uint) adalah fungsi yang tidak bisa mengubah penyimpanan dan dapat mengembalikan informasi.
4. Adress(this).balance adalah variabel tipe alamat yang selalu memiliki properti yang disebut dengan balance, memberikan sejumlah ether yang disimpan pada alamat tersebut. Dapat diakses dengan mudah tetapi hanya menampilkan beberapa yang tersimpan disana. Adress(this) mengonversi Smart Contract ke alamat.

The image displays two sequential screenshots of the Remix Ethereum IDE interface, demonstrating the process of deploying and interacting with a smart contract.

**Top Screenshot: Deployment Phase**

- Left Panel (DEPLOY & RUN TRANSACTIONS):** The "ENVIRONMENT" is set to "JavaScript VM (London)". The "ACCOUNT" is "0x5B3...eddC4 (99.999999%)". The "GAS LIMIT" is "3000000". The "VALUE" is "0 Wei". The "CONTRACT" is "SendMoneyExample - UTS\_LAB1.sol". The "Deploy" button is highlighted.
- Center Panel (Code Editor):** Displays the Solidity code for the `SendMoneyExample` contract. The code includes a constructor, `receiveMoney()`, `getBalance()`, and `withdrawMoney()` functions.
- Bottom Panel (Console/Debug):** Shows the deployment transaction log: `[vm] from: 0x5B3...eddC4 to: SendMoneyExample.(constructor) value: 0 wei data: 0x608...d0033 logs: 0 hash: 0x04b...8fce3`. The "Debug" button is visible.

**Bottom Screenshot: Interaction Phase**

- Left Panel (DEPLOY & RUN TRANSACTIONS):** The "Deploy" button is no longer visible. The "Deployed Contracts" section shows the `SENDMONEYEXAMPLE AT 0x091...` contract. Below it, there are buttons for `receiveMoney`, `withdrawMoney`, `balanceReceiv...`, and `getBalance`. The "Low level interactions" section shows the "CALLDATA" field and a "Transact" button.
- Center Panel (Code Editor):** The same Solidity code for the `SendMoneyExample` contract is displayed.
- Bottom Panel (Console/Debug):** The same deployment transaction log is shown, indicating the contract has been successfully deployed.

## Deploy and Run Transactions

The image displays two screenshots of the Remix Ethereum IDE, illustrating the execution of a Smart Contract named `SendMoneyExample`.

**Top Screenshot:** The IDE shows the contract code and the execution results. The contract has three functions: `receiveMoney()`, `withdrawMoney()`, and `getBalance()`. The execution results show three successful transactions:

- `[vm] from: 0x5B3...eddC4 to: SendMoneyExample.withdrawMoney() 0xd91...39138 value: 0 wei data: 0xac4...46002 logs: 0 hash: 0x1a4...1b816`
- `[call] from: 0x5B380a6a701c568545dCfC8B75F56beddC4 to: SendMoneyExample.balanceReceived() data: 0x52a...90c42`
- `[call] from: 0x5B380a6a701c568545dCfC8B75F56beddC4 to: SendMoneyExample.getBalance() data: 0x120...65fe0`

**Bottom Screenshot:** The IDE shows the same contract code, but the execution results show two failed transactions:

- `[vm] from: 0x5B3...eddC4 to: SendMoneyExample.(constructor) value: 1000000000000000000 wei data: 0x608...d0033 logs: 0 hash: 0xbdc...afc99`
- `[vm] from: 0x5B3...eddC4 to: SendMoneyExample.(constructor) value: 1000000000000000000 wei data: 0x608...d0033 logs: 0 hash: 0xbdc...afc99`

The error messages indicate that the transactions failed due to a `value` error, likely because the value provided was not 1 ether as intended.

Mengirim beberapa Ether ke Smart Contract dengan menukar nilai value dengan 1