



# Modul

## Smart Contract

### A. Pendahuluan

#### 1. Deskripsi Singkat

Modul ini berjudul "Pengantar Smart Contract", yang membahas konsep dasar, bahasa pemrograman, serta praktik implementasi smart contract sederhana di blockchain.

1. Konsep dasar smart contract dan fungsinya di blockchain.
2. Pengenalan bahasa pemrograman Solidity sebagai standar pembuatan smart contract di Ethereum.
3. Langkah-langkah praktikum, mulai dari menulis kode, mengompilasi, hingga melakukan deployment ke blockchain menggunakan Remix IDE.
4. Contoh implementasi smart contract sederhana untuk memahami mekanisme kerja dan interaksi dengan pengguna.

#### 2. Tujuan

Memberikan pemahaman dasar serta keterampilan teknis bagi mahasiswa dalam merancang dan mengimplementasikan smart contract di blockchain Ethereum.

#### 3. Capaian Pembelajaran Mata Kuliah

Setelah menyelesaikan kegiatan belajar ini, mahasiswa diharapkan mampu:

1. Memahami Konsep Dasar Blockchain dan Smart Contract  
Mahasiswa mampu menjelaskan konsep dasar blockchain, fungsi smart contract, serta peran keduanya dalam ekosistem terdesentralisasi.
2. Menganalisis Arsitektur dan Lingkungan Pengembangan Smart Contract  
Mahasiswa mampu mengidentifikasi berbagai platform blockchain yang mendukung smart contract, serta memahami perbedaan di antara bahasa pemrograman smart contract.
3. Merancang dan Mengembangkan Smart Contract Sederhana  
Mahasiswa mampu menulis, menguji, dan melakukan deploy smart contract sederhana menggunakan Solidity dan Remix IDE.
4. Mengimplementasikan Smart Contract dalam Berbagai Kasus Penggunaan  
Mahasiswa mampu menerapkan smart contract untuk berbagai skenario nyata seperti transaksi digital, voting, dan manajemen aset digital (NFT).



## B. Uraian Materi

### 1. Apa Itu Smart Contract?

Smart Contract adalah program komputer yang berjalan di atas blockchain dan dieksekusi secara otomatis saat kondisi tertentu terpenuhi. Konsep ini pertama kali diperkenalkan oleh Nick Szabo pada tahun 1994.

#### a. Karakteristik Smart Contract

1. Self-executing: Eksekusi otomatis tanpa perantara.
2. Immutable: Tidak bisa diubah setelah di-deploy.
3. Transparent: Semua orang bisa melihat kode dan hasil eksekusinya.
4. Trustless: Tidak butuh pihak ketiga, kode adalah kontraknya.

#### b. Manfaat Smart Contract

1. Mengurangi biaya administrasi.
2. Mengurangi risiko manipulasi.
3. Meningkatkan efisiensi dan kecepatan.

### 2. Blockchain Pendukung Smart Contract

#### Contoh Blockchain Pendukung

Blockchain	Bahasa Utama
Ethereum	Solidity
Binance Smart Chain	Solidity
Solana	Rust
Cardano	Plutus (Haskell)
Bitcoin	Script (Terbatas, multisig dan timelock saja)

### 3. Bahasa Pemrograman Smart Contract: Solidity

#### a. Apa Itu Solidity?

Solidity adalah bahasa pemrograman utama untuk menulis smart contract di jaringan Ethereum dan kompatibel dengan EVM (Ethereum Virtual Machine).

#### b. Karakteristik Solidity

- Tipe statis (statically typed).
- Mirip JavaScript/C++.
- Terintegrasi dengan alat-alat seperti Remix, Hardhat, dan Truffle.

## C. Praktikum - Membuat Smart Contract Sederhana

### a. Alat yang Digunakan

1. Remix IDE (<https://remix.ethereum.org>)
2. Metamask Wallet (Testnet Sepolia)
3. Faucet untuk Testnet ETH (<https://sepoliafaucet.com>)

### b. Langkah-langkah

1. Buka **Remix IDE**.
2. Buat file baru: SimpleContract.sol.
3. Salin kode berikut:

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract SimpleContract {
5     string public message;
6
7     constructor(string memory _message) {
8         message = _message;
9     }
10
11     function updateMessage(string memory _newMessage) public {
12         message = _newMessage;
13     }
14 }
15
```

4. Compile file.
5. Pilih **Environment: Injected Provider (MetaMask)**.
6. Hubungkan ke jaringan **Sepolia Testnet**.
7. Klik **Deploy**.
8. Panggil fungsi **updateMessage** di Remix dan lihat hasilnya.



## Modul Blockchain

Ade Kurniawan. S.Pd., M.Pd.T  
Informatika UNP



### D. Referensi

1. Ethereum.org - Smart Contract Documentation
2. Solidity Documentation - <https://docs.soliditylang.org>
3. Remix IDE - <https://remix.ethereum.org>
4. Metamask - <https://metamask.io>