

BUILDING WITH IDRX STABLECOIN ON LISK BOOTCAMP

Pertemuan ke – 3

Interacting with IDRX Stablecoin

IDRX Stablecoin is a coin that pegged into real world asset (Rupiah). IDRX Stablecoin benefits including Cheap, Fast, Transparent, No paperwork, No middleman, Inclusive.

Why Now ?

Indonesia is at 3rd rank of global crypto adoption, First-Mover Advantage, De-dollarization trend for cross-border transaction (same as the goal from brics idea), Foreign exchange volatility & high slippage, timing stablecoin is becoming the new financial rails, other countries already have their own stablecoin ecosystem.

Dalam konteks **blockchain**, **faucet** adalah layanan atau situs web yang memberikan sejumlah kecil cryptocurrency secara gratis kepada pengguna, biasanya untuk tujuan **pengujian** atau **pembelajaran**.

Jenis Faucet:

1. **Testnet Faucet**

Digunakan untuk mengirim token testnet (bukan uang sungguhan) ke dompet pengguna agar bisa menguji smart contract atau aplikasi blockchain tanpa menggunakan uang asli.

Contoh: *Sepolia Faucet, Lisk Sepolia Faucet, Polygon Mumbai Faucet.*

2. **Mainnet Faucet** (jarang)

Dulu ada faucet yang memberikan koin sungguhan dalam jumlah kecil (seperti Bitcoin atau Ethereum), tetapi sekarang sudah sangat langka karena tingginya nilai aset kripto.

Fungsi Faucet dalam Pengembangan:

- Mengisi dompet developer dengan **test token**.
- Digunakan di **testnet** untuk menguji fitur seperti transaksi, smart contract deployment, atau gas fee.
- Aman karena tidak menggunakan token riil.



Contoh Kasus:

Jika kamu membuat aplikasi voting atau crowdfunding di blockchain, dan kamu ingin mencobanya di **Lisk Sepolia testnet**, kamu bisa:

1. Minta token testnet dari **Lisk Faucet**.
2. Gunakan token itu untuk **deploy smart contract** atau **melakukan transaksi** tanpa biaya nyata.





⚙️ 1. Mainnet (Main Network)

Mainnet adalah **jaringan blockchain utama** tempat semua transaksi, kontrak pintar (smart contract), dan aset kripto bersifat **nyata** dan memiliki **nilai ekonomi**.

-  Token yang digunakan: **asli dan berharga** (misalnya ETH di Ethereum mainnet).
-  Transaksi: **permanen dan tercatat di blockchain publik**.
-  Digunakan oleh **pengguna akhir** dan **aplikasi resmi**.
-  Contoh: Ethereum Mainnet, Bitcoin Mainnet, Binance Smart Chain Mainnet.

🧪 2. Testnet (Test Network)

Testnet adalah jaringan blockchain **simulasi** untuk **pengujian**. Semua token dan transaksi di sini **tidak memiliki nilai riil**.

-  Token: **gratis** dari faucet, digunakan hanya untuk pengujian.
-  Transaksi: tidak nyata, bisa dihapus atau di-reset.
-  Tujuan: untuk **developer menguji** aplikasi, kontrak pintar, dan integrasi sebelum meluncurkannya ke mainnet.
-  Contoh: Sepolia Testnet (Ethereum), Lisk Sepolia Testnet, Polygon Mumbai Testnet.

🔍 Perbandingan Singkat

| Fitur | Mainnet | Testnet |
|--------------|----------------------------|----------------------------|
| Tujuan | Penggunaan nyata | Pengujian dan pengembangan |
| Nilai token | Bernilai (nyata) | Tidak bernilai (gratis) |
| Resiko biaya | Ada resiko kehilangan aset | Tidak ada resiko finansial |

| Fitur | Mainnet | Testnet |
|--------------|------------------------|---------------------------------|
| Stabilitas | Sangat stabil dan aman | Bisa tidak stabil atau di-reset |
| Contoh token | ETH, BTC, BNB, IDRXX | SepoliaETH, TestBNB, TestIDRX |

Testnet sangat penting untuk **mencegah kesalahan fatal** sebelum aplikasi blockchain diluncurkan secara resmi.

ERC-20 adalah **standar token** di jaringan **Ethereum** yang digunakan untuk membuat dan mengelola **token digital** (seperti stablecoin, token proyek, atau token voting) agar **kompatibel** dengan ekosistem Ethereum.

Pengertian Singkat:

ERC-20 (Ethereum Request for Comment 20) adalah protokol standar untuk **token yang bisa dipertukarkan (fungible)**, artinya setiap token memiliki **nilai dan fungsi yang sama**.

Contohnya:

1 token USDT = 1 token USDT

1 token ABC = 1 token ABC (selama dari smart contract yang sama)

Fungsi ERC-20 Token:

- Representasi **aset digital** (misalnya: USDT, DAI, IDRXX)
 - **Crowdfunding (ICO/IDO)**
 - **Governance token** untuk voting
 - **Utility token** untuk akses fitur di aplikasi
 - Digunakan di DApps, DeFi, game, NFT marketplace, dll.
-

Fitur Standar ERC-20:

Smart contract token ERC-20 harus punya fungsi seperti:

| Fungsi | Penjelasan |
|-------------|--|
| totalSupply | Jumlah total token yang ada |
| balanceOf | Melihat saldo token dari address tertentu |
| transfer | Mengirim token ke address lain |
| approve | Memberi izin address lain untuk menggunakan token kita |

| Fungsi | Penjelasan |
|--------------|--|
| transferFrom | Mengirim token dari akun yang sudah disetujui |
| allowance | Melihat berapa banyak token yang sudah disetujui untuk digunakan |

✂ Contoh Implementasi ERC-20 Sederhana (Solidity):

solidity

CopyEdit

```
function transfer(address to, uint256 amount) public returns (bool);
function balanceOf(address owner) public view returns (uint256);
```

📦 Contoh ERC-20 Token Populer:

- **USDT** (Tether)
 - **USDC** (USD Coin)
 - **DAI**
 - **IDRX** (Indonesia Rupiah Stablecoin)
 - **LINK** (Chainlink)
-

🔄 Fungible vs Non-Fungible:

- ERC-20 → **Fungible Token**
- ERC-721 → **Non-Fungible Token (NFT)**

1. ✅ Function Visibility (Aksesibilitas)

Menentukan **siapa** yang bisa memanggil fungsi.

| Keyword | Artinya |
|----------|--|
| public | Bisa dipanggil dari luar dan dalam kontrak. |
| external | Hanya bisa dipanggil dari luar kontrak (bukan dari dalam). |
| internal | Hanya bisa dipanggil dari dalam kontrak itu sendiri atau turunan. |
| private | Hanya bisa dipanggil dari dalam kontrak tempat dideklarasikan. |

Contoh:

solidity

CopyEdit

```
function doSomething() public { ... }
function onlyExternal() external { ... }
```

2. Function State Mutability (Status Mutasi State)

Menentukan apakah fungsi **mengubah atau membaca data blockchain**.

| Keyword | Fungsi | Keterangan |
|---------|--------------------------------|--|
| view | Membaca data dari state | Tidak mengubah state |
| pure | Tidak membaca atau mengubah | Hanya hitung-hitungan, tanpa akses storage |
| payable | Bisa menerima ETH/token | Wajib untuk fungsi terima ETH |
| (tanpa) | Mengubah state | Fungsi default yang bisa menulis ke blockchain |

Contoh:

```
solidity
CopyEdit
function getBalance() public view returns (uint) { ... }
function add(uint a, uint b) public pure returns (uint) { ... }
function deposit() public payable { ... }
```

3. Return Types (Tipe Pengembalian)

Menentukan jenis data yang dikembalikan oleh fungsi.

Contoh:

```
solidity
CopyEdit
function getName() public view returns (string memory) {
    return name;
}
```

4. Modifier (Akses Tambahan)

Modifier adalah jenis khusus yang bisa digunakan untuk menambahkan **syarat tambahan** sebelum fungsi dijalankan.

Contoh:

```
solidity
CopyEdit
modifier onlyOwner() {
    require(msg.sender == owner, "Not authorized");
    _;
}

function withdraw() public onlyOwner {
```

```
// hanya owner yang bisa withdraw  
}
```

5. 📦 Function Type sebagai Tipe Variabel

Fungsi juga bisa dijadikan **variabel** atau **parameter**.

Contoh:

```
solidity  
CopyEdit  
function apply(uint x, function (uint) external returns (uint) func) public returns (uint) {  
    return func(x);  
}
```

Kesimpulan:

Function types di Solidity mengatur:

- Siapa yang bisa memanggil (visibility)
- Apakah bisa ubah data atau tidak (mutability)
- Bisa menerima ETH atau tidak (payable)
- Mengembalikan data atau tidak (returns)
- Modifier untuk kontrol akses
- Fungsi sebagai variable

Fungsi-fungsi yang kamu tulis adalah bagian dari standar **ERC-20** pada Ethereum, yaitu antarmuka standar untuk token yang dapat dipertukarkan (fungible tokens). Berikut adalah penjelasan dari masing-masing fungsi:

1. function name() public view returns (string)

- **Fungsi:** Mengembalikan nama token.
 - **Contoh:** "MyToken"
 - **Kegunaan:** Untuk menampilkan nama token di wallet seperti MetaMask atau block explorer.
-

2. function symbol() public view returns (string)

- **Fungsi:** Mengembalikan simbol token.

- **Contoh:** "MTK"
 - **Kegunaan:** Seperti ticker saham, memudahkan pengguna mengenali token (misalnya BTC, ETH, DAI).
-

3. function decimals() public view returns (uint8)

- **Fungsi:** Menentukan jumlah angka di belakang koma desimal yang didukung token.
 - **Contoh:** 18 artinya 1 token = 10^{18} unit dasar.
 - **Kegunaan:** Untuk menghitung dan menampilkan nilai token secara manusiawi.
-

4. function totalSupply() public view returns (uint256)

- **Fungsi:** Mengembalikan total jumlah token yang beredar.
 - **Contoh:** $1000000 * 10^{18}$ jika total supply adalah satu juta token dengan 18 desimal.
 - **Kegunaan:** Mengetahui jumlah maksimal token dalam sistem.
-

5. function balanceOf(address _owner) public view returns (uint256 balance)

- **Fungsi:** Mengembalikan saldo token dari alamat _owner.
 - **Kegunaan:** Untuk mengecek berapa banyak token yang dimiliki oleh suatu alamat.
-

6. function transfer(address _to, uint256 _value) public returns (bool success)

- **Fungsi:** Mengirim sejumlah _value token dari pengirim (msg.sender) ke alamat _to.
 - **Kegunaan:** Untuk melakukan pengiriman token antar user secara langsung.
-

7. function transferFrom(address _from, address _to, uint256 _value) public returns (bool success)

- **Fungsi:** Mengirim token dari _from ke _to atas izin sebelumnya.
 - **Kegunaan:** Digunakan dalam mekanisme delegasi atau kontrak pihak ketiga seperti DEX.
-

8. function approve(address _spender, uint256 _value) public returns (bool success)

- **Fungsi:** Memberi izin kepada `_spender` untuk membelanjakan hingga `_value` token dari pemilik (`msg.sender`).
 - **Kegunaan:** Digunakan bersama `transferFrom()` untuk otorisasi.
-

9. function allowance(address _owner, address _spender) public view returns (uint256 remaining)

- **Fungsi:** Mengembalikan sisa jumlah token yang diizinkan `_spender` untuk dibelanjakan dari `_owner`.
- **Kegunaan:** Untuk melihat berapa banyak token yang masih bisa digunakan oleh pihak ketiga (seperti smart contract) dari suatu alamat.