

Nama : Raihan Puspa Anggita Putri

NIM : 1103184065

Kelas : TK-42-PIL

Lesson 13 : Hardhat DeFI & Aave

```
interface AggregatorV3Interface {
    function decimals() external view returns (uint8);

    function description() external view returns (string memory);

    function version() external view returns (uint256);

    // getRoundData and latestRoundData should both raise "No data present"
    // if they do not have data to report, instead of returning unset values
    // which could be misinterpreted as actual reported values.
    function getRoundData(uint80 _roundId)
        external
        view
        returns (
            uint80 roundId,
            int256 answer,
            uint256 startedAt,
            uint256 updatedAt,
            uint80 answeredInRound
        );

    function latestRoundData()
        external
        view
        returns (
            uint80 roundId,
            int256 answer,
            uint256 startedAt,
            uint256 updatedAt,
            uint80 answeredInRound
        );
};
```

```
interface IERC20 {
    function allowance(address owner, address spender) external view returns (uint256 remaining);

    function approve(address spender, uint256 value) external returns (bool success);

    function balanceOf(address owner) external view returns (uint256 balance);

    function decimals() external view returns (uint8 decimalPlaces);

    function decreaseApproval(address spender, uint256 addedValue) external returns (bool success);

    function increaseApproval(address spender, uint256 subtractedValue) external;

    function name() external view returns (string memory tokenName);

    function symbol() external view returns (string memory tokenSymbol);

    function totalSupply() external view returns (uint256 totalTokensIssued);

    function transfer(address to, uint256 value) external returns (bool success);

    function transferFrom(
        address from,
        address to,
        uint256 value
    ) external returns (bool success);
}
```

```

function getReserveNormalizedIncome(address asset) external view returns (uint256);

/**
 * @dev Returns the normalized variable debt per unit of asset
 * @param asset The address of the underlying asset of the reserve
 * @return The reserve normalized variable debt
 */
function getReserveNormalizedVariableDebt(address asset) external view returns (uint256);

/**
 * @dev Returns the state and configuration of the reserve
 * @param asset The address of the underlying asset of the reserve
 * @return The state of the reserve
 */
function getReserveData(address asset) external view returns (DataTypes.ReserveData memory);

function finalizeTransfer(
    address asset,
    address from,
    address to,
    uint256 amount,
    uint256 balanceFromAfter,
    uint256 balanceToBefore
) external;

function getReservesList() external view returns (address[] memory);

function getAddressesProvider() external view returns (ILendingPoolAddressesProvider);

function setPause(bool val) external;

function paused() external view returns (bool);
}

```

```
interface IWeth {
    function allowance(address owner, address spender) external view returns (uint256 remaining);

    function approve(address spender, uint256 value) external returns (bool success);

    function balanceOf(address owner) external view returns (uint256 balance);

    function decimals() external view returns (uint8 decimalPlaces);

    function name() external view returns (string memory tokenName);

    function symbol() external view returns (string memory tokenSymbol);

    function totalSupply() external view returns (uint256 totalTokensIssued);

    function transfer(address to, uint256 value) external returns (bool success);

    function transferFrom(
        address from,
        address to,
        uint256 value
    ) external returns (bool success);

    function deposit() external payable;

    function withdraw(uint256 wad) external;
}
```