

Technical Assessment: Tarik Tambang

A. Deskripsi Tugas

Buatlah sebuah Smart Contract berbasis Solidity yang mensimulasikan permainan "Tarik Tambang" dengan taruhan uang asli (Native Token seperti ETH/MATIC/BNB).

B. Spesifikasi Fungsional (Game Logic)

1. Inisialisasi Permainan

- Hanya **Admin** (owner contract) yang dapat memulai permainan.
- Saat inisialisasi, Admin menentukan **durasi permainan** (batas waktu/deadline).

2. Fase Permainan (Betting)

- Selama waktu belum habis, user (alamat wallet) dapat mengirimkan sejumlah coin/token ke dalam smart contract.
- User harus memilih untuk mendukung **Tim A** atau **Tim B**.
- Smart Contract harus mencatat jumlah kontribusi masing-masing user secara akurat.

3. Penyelesaian (Game End)

- Setelah waktu habis, Admin dapat menutup permainan/menjalankan fungsi finalisasi.
- **Kondisi Menang:**
 - Jika **Total Dana Tim A > Total Dana Tim B**, maka **Tim A Menang**.
 - Jika **Total Dana Tim B > Total Dana Tim A**, maka **Tim B Menang**.
 - Total pot (dana Tim A + Tim B) menjadi hadiah bagi pemenang.
- **Kondisi Seri (Draw):**
 - Jika **Total Dana Tim A == Total Dana Tim B**.

4. Distribusi Hadiah (Claim/Withdraw Mechanism)

- **Penting:** Untuk menghindari masalah Gas Limit, distribusi hadiah **TIDAK BOLEH** dikirim otomatis (airdrop) oleh kontrak.
- User dari tim pemenang harus melakukan **Withdraw/Claim** secara mandiri.
- **Perhitungan Hadiah:** Distribusi harus **proporsional** berdasarkan kontribusi user terhadap timnya.
 - **Rumus:** $(\text{Kontribusi User} / \text{Total Dana Tim Pemenang}) * \text{Total Pot Kedua Tim}$
- **Refund (Jika Seri):**

- Jika permainan berakhir seri, user dari kedua tim dapat melakukan claim untuk mengambil kembali uang mereka (100% refund).

5. Spesifikasi Teknis

- **Bahasa:** Solidity (Gunakan versi pragma yang stabil, e.g., `^0.8.0`).
- **Framework:** Project wajib dibuat menggunakan **Foundry**.
- **Testing:** Sertakan Unit Test script di dalam Foundry untuk menguji skenario:
 - Tim A menang (cek kalkulasi hadiah).
 - Kondisi Seri (cek refund).
 - User mencoba withdraw sebelum waktu habis (harus revert).
 - User dari tim kalah mencoba withdraw (harus revert).

6. Ketentuan Deployment & Submission

A. Deployment

- Deploy Smart Contract ke **Testnet** (Base Sepolia). Token base sepolia dapat didapatkan dari faucet berikut <https://www.alchemy.com/faucets/base-sepolia> atau <https://docs.base.org/base-chain/tools/network-faucets#ethfaucet-com>
- Source code contract **wajib diverifikasi** di Block Explorer Basescan (<https://sepolia.basescan.org/>)

B. Repository

- Upload kode ke **GitHub** (Repository Public).
- Repository harus bersih dan terstruktur.

C. Dokumentasi (README.md) Wajib menyertakan file

README.md yang berisi:

- **Panduan Instalasi & Testing:** Cara menjalankan test suite di environment local menggunakan Foundry.
- **Contract Address:** Alamat smart contract yang sudah dideploy.
- **Link Block Explorer:** Link ke halaman verifikasi contract.
- **Penjelasan Singkat:** Penjelasan singkat mengenai pendekatan solusi yang Anda ambil (terutama bagaimana Anda menangani pembagian hadiah).