

# **Mahjong Analyzer**

Mini Projek

---

Muhammad Jilan Wicaksono

October 30, 2025

Universitas Indonesia

## Apa itu Mahjong?

---

- Permainan papan yang menggunakan  $(3 \times 9 + 7) \times 4$  tile.



- Dimainkan oleh 4 atau 3 pemain.
- Awalnya, setiap pemain mendapatkan *hand* berisi 13 tile.
- Tiap giliran, satu pemain mengambil satu tile dari dek (disebut dinding atau *wall*) untuk mendapatkan bentuk tertentu.



**Hand**

Each player starts with a hand of 13 Tiles

**Sequence**  
A set of 3 consecutive tiles

**Triplet**  
A set of 3 identical tiles

**Pair**  
A set of 2 identical tiles

- Satu hand terdiri dari 13 tile.
- Ada dua jenis kumpulan tile:
  - Pair: 2 tile yang sama.
  - Meld: 3 tile yang bisa berbentuk Sequence ataupun Triplet.

## Winning

A winning hand usually consists of 4 Sequences or Triplets + 1 Pair as well as a Yaku



- Winning hand (*Agari*): hand dengan 14 ubin ( $13 + 1$  setelah draw sebelum discard) yang memenuhi pola tertentu.
- Biasanya harus berupa 4 meld disertai dengan 1 pair.
- Jika suatu pemain mencapai agari, pemain itu dapat menghentikan ronde tersebut, mendapatkan skor sesuai dengan hand terakhirnya, lalu ronde baru dimulai.

## **Permasalahan**

---

Mudah dilihat bahwa contoh tadi memuat 4 meld dan 1 pair.

**Winning**

A winning hand usually consists of 4 Sequences or Triplets + 1 Pair as well as a Yaku



1 2 3 4 4 4 5 6 7 E E E S 南

Namun, terkadang ada hand yang sulit untuk dilihat, seperti



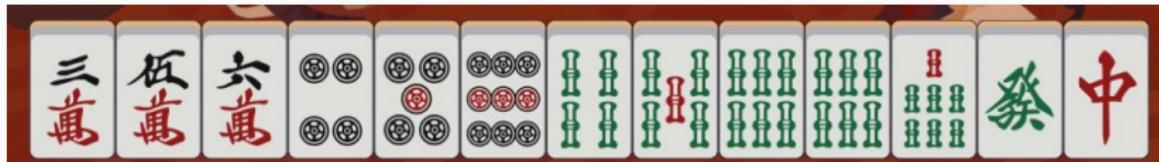
Akan dibuat suatu fungsi untuk memeriksa apakah suatu hand adalah agari.

Satu hal lain yang menarik untuk dihitung adalah seberapa jauh suatu hand dari agari. Hal ini disebut sebagai shanten number.

contoh 1 shanten



contoh lain



Yang terakhir adalah menghitung skor yang diterima dari agari



# Tujuan Mini Projek

- Membuat fungsi untuk mengecek tangan kemenangan, bertipe seperti

```
1     agariCheck :: Hand -> Bool
```

- Membuat fungsi untuk menghitung shanten number, atau seberapa jauh hand dari agari, bertipe seperti

```
1     shantenCalculator :: Hand -> Int
```

- Membuat fungsi untuk menghitung skor yang diperoleh dari agari, bertipe seperti

```
1     scoreCalculator :: Hand -> Maybe Int
```

## Target Progress

---

# Target Progres per Pekan i

- Target Pekan 1:
    - Menyusun tipe data untuk Suit, Tile, Hand, Meld, dan Agari.
    - Membuat fungsi untuk mencari pair dan meld dari suatu hand.
  - Target Pekan 2:
    - Membuat fungsi validasi tangan kemenangan untuk hand standar, yaitu 4 meld dan 1 pair.
    - Membuat fungsi validasi tangan kemenangan untuk hand tidak standar, yaitu Chii Toitsu (7 pair) dan Kokushi Musou (13 + 1 tile honor dan terminal).
- Fungsi bertipe seperti
- ```
agariCheck :: Hand -> Maybe Partition dengan
Just Partition menandakan hand yang valid beserta
partisinya, dan Nothing jika hand tidak valid.
```

## Target Progres per Pekan ii

- Target Pekan 3:
  - Mengidentifikasi semua kandidat set dalam suatu hand: Meld, pair, dan meld yang tidak lengkap.
  - Mengimplementasi fungsi untuk menghitung nilai shanten, seperti `calculateShanten :: Hand -> Int`.
  - Mengadaptasi algoritma untuk menangani hand yang tidak standar.

## Target Progres per Pekan iii

- Target Pekan 4:
  - Mendefinisikan tipe data untuk merepresentasikan semua Yaku yang mungkin.
  - Mengimplementasikan fungsi untuk mengecek yaku dari tangan, bertipe seperti `isTanyao :: Hand -> Bool`.
  - Membuat fungsi yang menerima sebuah hand dan mengembalikan daftar semua Yaku yang terpenuhi.
  - Membuat fungsi untuk menghitung *Fu* (poin minor) dari bentuk hand.
  - membuat fungsi untuk menghitung skor yang didapat berdasarkan Yaku (Han) dan Fu dari tangan.

## Target Progres per Pekan iv

- Target Pekan 5:
  - Melakukan testing terhadap fungsi yang telah dibuat.
  - Mulai membangun Command-Line Interface sederhana.
- Target Pekan 6:
  - Melanjutkan testing dan CLI, serta finalisasi.
  - Membuat dokumentasi.

## Aspek Fungsional

---

# Aspek Fungsional

- **Immutability & Pure Function**

Fungsi seperti `calculateShanten` selalu memberikan hasil yang sama untuk input yang sama.

- **Typeclass, ADT, & Pattern Matching**

Untuk memodelkan semua entitas dan aturan permainan secara type-safe.

- **Monad**

Mencakup State **Monad** untuk mengelola state internal, Reader **Monad** yang menyediakan konteks permainan, dan **IO Monad**.

- **Higher-Order Function**

Seperti **map**, **filter**, **foldl** yang digunakan untuk memproses Hand.