

Penjelasan mengenai Import :

1. React :

Untuk menggunakan framework React di dalam aplikasi kita tentunya kita perlu mengimportnya.

1. useState dan useEffect :

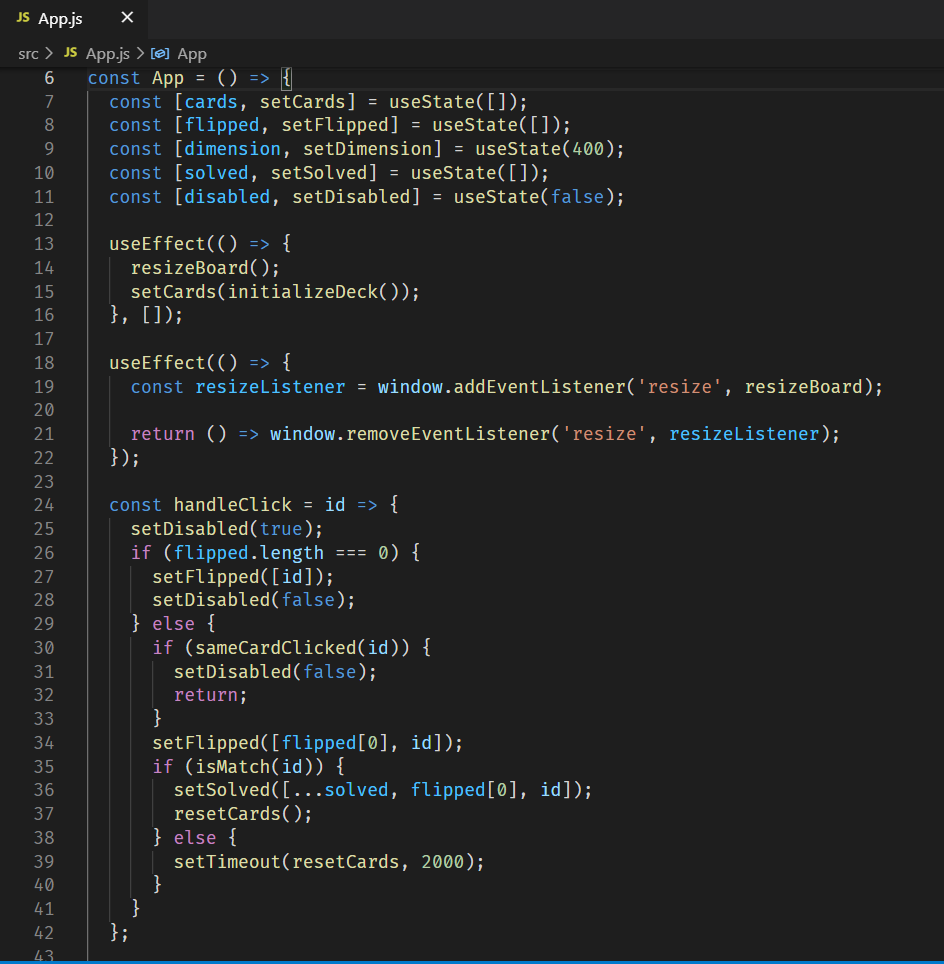
Hook yang kami digunakan dalam aplikasi kita, untuk penjelasan fungsi dan cara pemakainannya akan dijelaskan dibagian bawah.

1. App.css :

Custom css yang kita gunakan untuk mempercantik tampilan halaman Web kita.

1. Board dan initializeDeck :

Komponen yang kita buat secara sendiri untuk dipanggil didalam App.js



App adalah sebuah fungsi yang didalamnya berisikan :

1. useState :

Sebuah Hook yang berfungsi untuk mengubah value dari sebuah variable.

Untuk menggunakan useState adalah sebagai berikut :

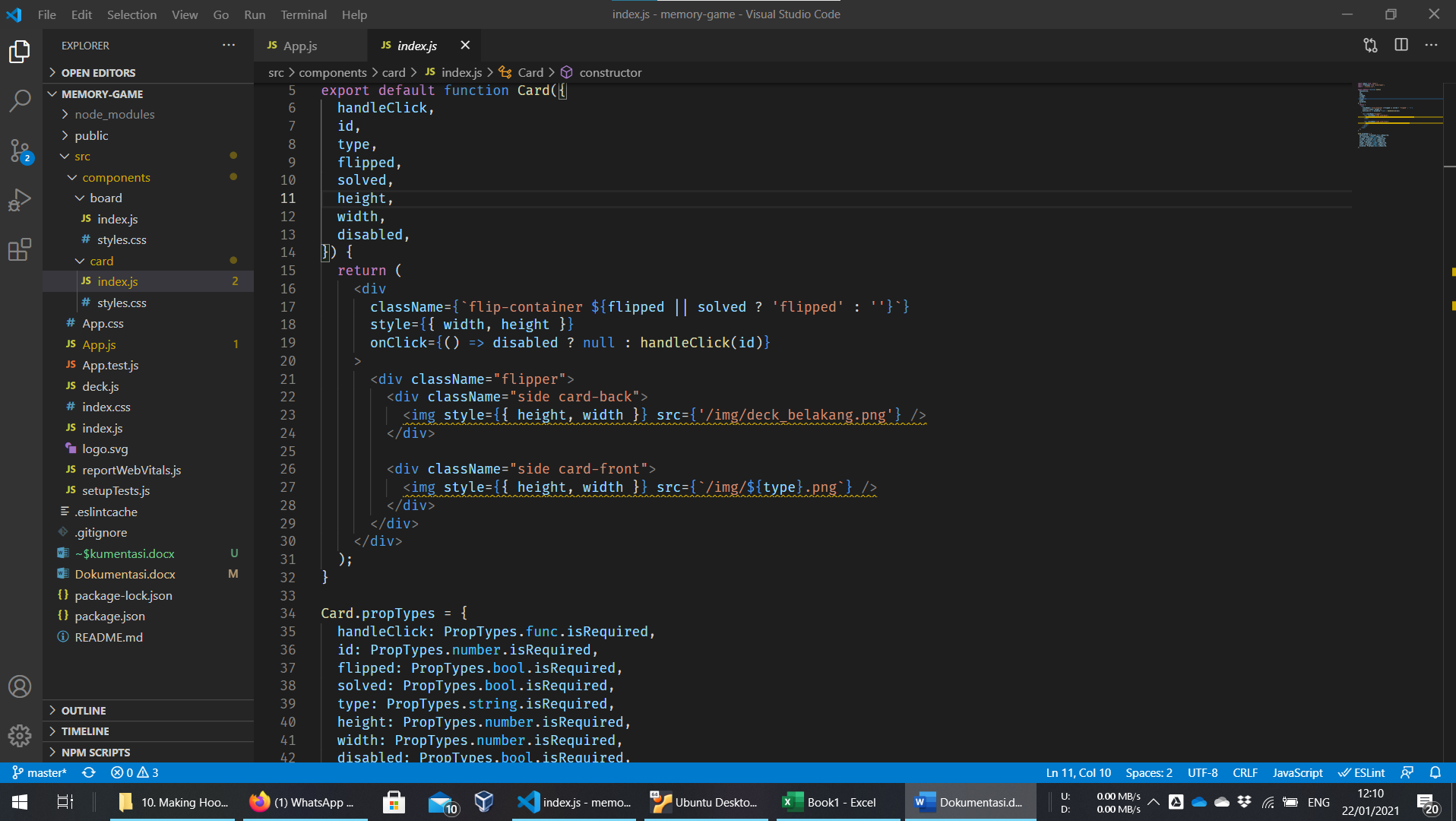
* **const [disabled, setDisabled] = useState(false);**
* **disabled** adalah sebuah variable penampung dari nilai yang akan diberikan.
* **setDisabled** adalah sebuah *function* yang berperan untuk mengset nilai ke dalam variable disabled.
* **useState(false)** adalah nilai awal atau dikenal initial value yang akan diset ke dalam variable disabled.

1. useEffect :

Sebuah Hook yang akan menjalankan fungsi lain dengan catatan :

* Tidak memiliki argument kedua
  + Fungsi di dalam useEffect ini akan dijalankan setiap saat ketika komponen di render
* Argument kedua adalah **[]**
  + Fungsi di dalam useEffect ini akan dijalankan saat pertama kali komponen siap dirender
* Argument kedua adalah [*value*]
  + Fungsi di dalam useEffect akan dijalakan hanya pertama kali komponen siap di render dan nilai *value* berubah

1. *Function* lainnya yang akan dijelaskan dalam Game Logic.

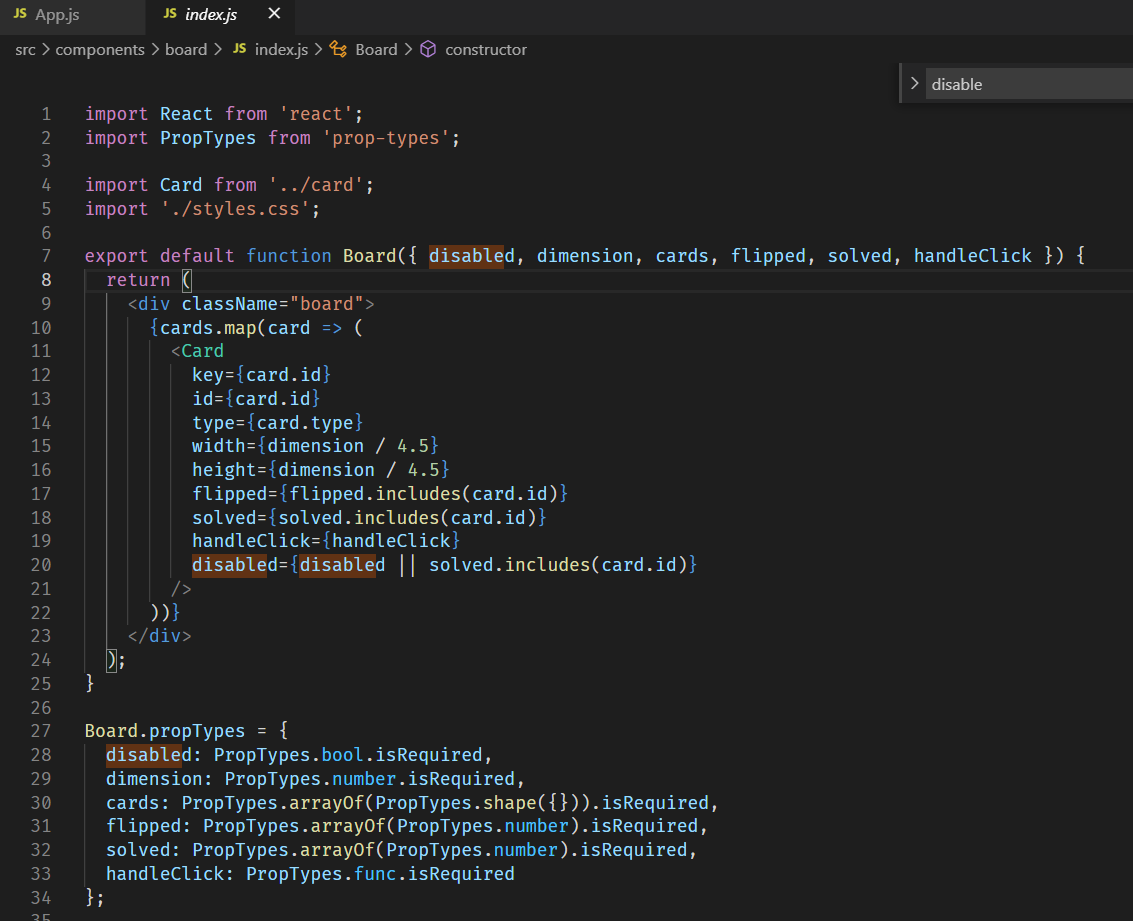


**Komponen Card**

Di dalam komponen Card terdapat 1 fungsi Card yang akan di export untuk digunakan di dalam komponen Board.

Dalam fungsi Card ini terdapat 8 prop yang akan digunakan untuk :

* **handleClick** adalah sebuah fungsi yang akan dijalankan ketika kartu tersebut di klik.
* **id** adalah prop yang bertipe data number yang berfungsi identitas kartu yang akan dikirim ke fungsi handleClick.
* **type** adalah prop bertipe data string yang digunakan untuk mengetahui nama kartu. Disini kita gunakan untuk memanggil nama file. Contoh kartu keyboard, maka kita memanggil file keyboard.png yang telah kami sediakan di dalam folder assets.
* **flipped** dan **solved** adalah prop yang bertipe data boolean yang digunakan untuk menentukan nama class dari div yang kita pake.
* **height** dan **width** adalah prop yang bertipe data number yang digunakan untuk menentukan ukuran lebar dan tinggi kartu.
* **disabled** adalah prop yang bertipe data Boolean yang berfungsi untuk menentukan apakah kartu tersebut masih bisa menjalankan fungsi handleClick. Kartu yang sedang dibuka / sudah dissolve tidak dapat menjalankan handleClick, sedangkan kartu yang lain bisa.

**Komponen Board**

Di dalam komponen Board terdapat 1 fungsi Board yang akan di export untuk digunakan di dalam App.js.

Dalam fungsi Board ini terdapat 6 prop yang akan digunakan untuk :

* **disabled** adalah prop bertipe data boolean hasil antara variable disabled di *or* kan dengan hasil cek card yang sudah di solved.
* **Dimension** adalah prop bertipe data number yang berasal dari state Dimension agar ukuran kartu dapat disesuaikan dengan ukuran layer pengguna (*responsive*).
* **Cards** adalah prop bertipe array yang berisi kumpulan kartu yang telah dirandom.
* **flipped** adalah prop bertipe data array yang berisi kumpulan kartu (2 kartu) yang sedang dibuka.
* **solved** adalah prop bertipe data array yang berisi kumpulan kartu yang sudah di solved.
* **handleClick** adalah prop bertipe data func yang digunakan sebagai perantara. Maksud perantara adalah memasingkan function handleClick yang berada di App.js ke komponen Card.

**Game Logic :**

**Disabled**: untuk mencegah user mengklik / membuka kartu-kartu yang ada selama proses pengecekan kartu yang dibuka.

**Solved**: untuk menyimpan kartu-kartu sepasang mana saja yang sudah ditemukan oleh user.

**Flipped**: untuk menyimpan kartu yang sedang dibuka oleh user menunggu kartu lain dibuka lagi oleh user.

**Logic**:

* Jika kartu yang dibuka adalah yang pertama, maka masukkan index kartu tersebut ke flipped
* Jika kartu yang dibuka adalah yang kedua:
  + Cek apakah kartu yang diklik adalah kartu yang sama dengan flipped, jika ya abaikan aksi tersebut
  + Jika berbeda, masukkan index kartu tersebut ke dalam flipped
    - Jika kedua kartu yang ada didalam flipped sama, maka masukkan index kedua kartu tersebut kedalam solved untuk membiarkan kedua kartu tersebut terus membuka
    - Jika kedua kartu berbeda, maka berikan delay 2 detik untuk melanjutkan permainan
    - Setelah pengecekan kedua kartu, hapus isi array flipped.