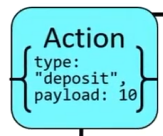
******Redux – State Management**

Ketika tombol Deposit diklik, **click event** akan masuk ke **Event Handler**. Event Handler mengirimkan sebuah **payload** ke **Store** (misalnya berupa state value). **Store** memeriksa **payload** tersebut untuk kemudian diolah oleh **reducer** yang sesuai. Misalnya payload yang dikirim adalah deposit, maka reducer untuk deposit yang akan dipanggil/dijalankan.



**1. Import:**

* **import { legacy\_createStore } from "redux";**
  + Mengambil fungsi **legacy\_createStore** dari pustaka Redux untuk membuat **store**.

**2. Reducer:**

* **const cartReducer = (state = { ... }, action) => { ... };**
  + Fungsi **reducer** bertugas mengelola perubahan **state** aplikasi.
  + Menerima dua argumen:
    - **state**: State saat ini (keranjang belanja dalam contoh ini).
    - **action**: Objek yang menjelaskan perubahan yang ingin dilakukan.
  + Mengembalikan **state** yang baru berdasarkan action yang diterima.

**3. State Awal:**

* **state = { cart: [{ id: 1, qty: 10 },] }**
  + Menentukan **state awal** keranjang belanja, berisi satu produk dengan **ID 1** dan **qty 10**.

**4. Menangani Action:**

* **switch (action.type) { ... }**
  + Memeriksa jenis **action** yang diterima.
  + **case "ADD\_TO\_CART": ...**
    - Jika **action** adalah **"ADD\_TO\_CART"**, tambahkan produk baru ke keranjang.
  + **return { ...state, cart: [...state.cart, action.payload] };**
    - * Mengembalikan **state** baru dengan produk yang ditambahkan.

**5. Store:**

* **const store = legacy\_createStore(cartReducer);**
  + Membuat **store Redux** menggunakan fungsi **legacy\_createStore** dan **reducer cartReducer**.

**6. Mengakses State Awal:**

* **console.log("ON CREATE STORE : ", store.getState());**
  + Menampilkan **state** awal keranjang belanja saat **store** dibuat.

**7. Berlangganan Perubahan State:**

* **store.subscribe(() => { ... });**
  + Mendaftarkan **callback** yang akan dijalankan setiap kali **state store** berubah.

**8. Dispatch Action:**

* **const action1 = { type: "ADD\_TO\_CART", payload: { id: 2, qty: 20 } };**
  + Membuat **action** untuk menambahkan produk baru dengan V dan **qty 20.**
* **store.dispatch(action1);**
  + Mengirim **action** ke **store** untuk mengubah **state**.

**9. Menampilkan State yang Berubah:**

* Callback dalam **store.subscribe** akan menampilkan **state** yang telah diperbarui setelah **action** di-**dispatch**.