1. Fetching Products

ProductNotifier Class

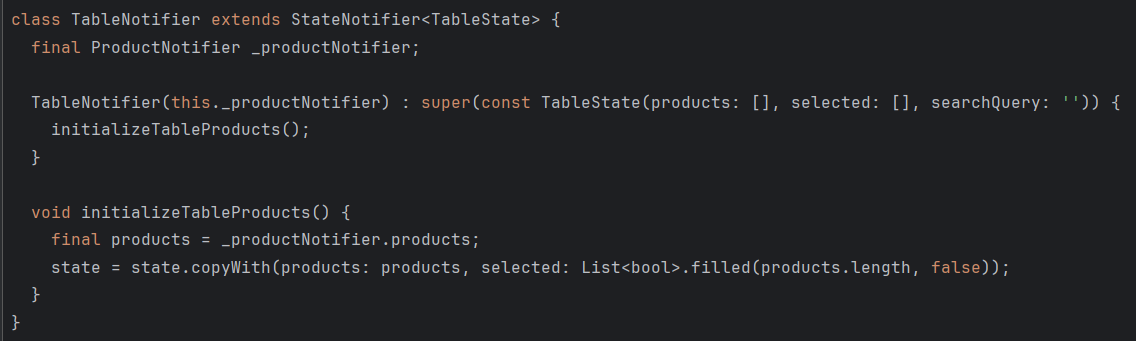
The ProductNotifier class bertanggung jawab untuk mengambil produk dari repository dan mengelola *state* dari produk tersebut.



* **ProductNotifier Class**: Mengelola *state* dari produk.
* **Metode fetchProducts**: Mengambil produk dari repository dan memperbarui *state*.
* **Getter products**: Mengembalikan daftar produk yang disimpan dalam *state*.
* **productNotifierProvider**: Menyediakan instance dari ProductNotifier.

2. Menginisialisasi Produk Tabel (*Initializing Table Products*)

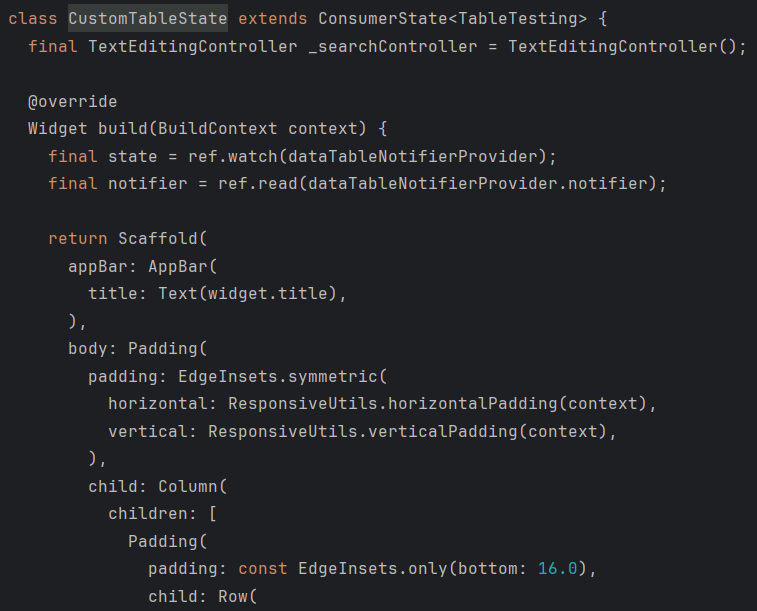
TableNotifier class menginisialisasi produk dalam tabel dengan fetching dari ProductNotifier.



* **TableNotifier Class**: Mengelola *state* dari tabel.
* **Metode initializeTableProducts**: Mengambil produk dari ProductNotifier dan memperbarui *state*.

3. Search Bar Input

CustomTableState class menangani input di *search bar* dan memperbarui *search query* di TableNotifier.

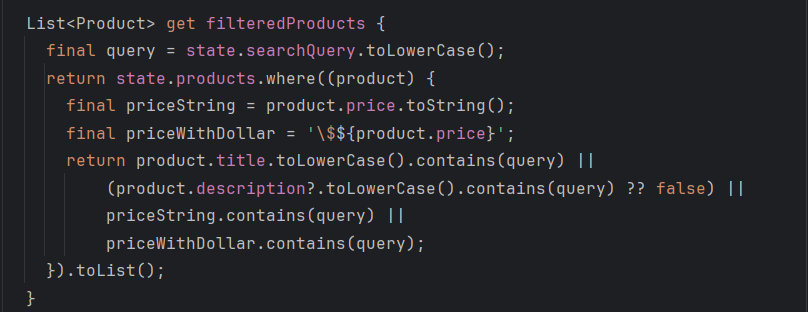




* **TextEditingController**: Mengelola input teks untuk *search bar*.
* **TextField**: Widget untuk *search bar*.
* **Callback onChanged**: Memperbarui *search query* di TableNotifier.

4. Memfilter Produk (*Filtering Products*)

TableNotifier class memfilter produk berdasarkan *search query*.



* List<Product> get filteredProducts {:

Line ini mendefinisikan *getter* bernama filteredProducts yang mengembalikan List<Product>

* final query = state.searchQuery.toLowerCase();:

Mengubah *search query* menjadi huruf kecil (lowercase) untuk perbandingan yang tidak case-insensitive dan menyimpannya dalam variabel query.

* return state.products.where((product) {:

Memfilter daftar produk menggunakan metode where, yang menerima sebuah fungsi yang mengembalikan true untuk produk yang cocok dengan *search query*.

* final priceString = product.price.toString();:

Mengubah harga produk menjadi string dan menyimpannya dalam variabel priceString.

* final priceWithDollar = '\$${product.price}';:

Membuat representasi string dari harga produk dengan tanda $ dan menyimpannya dalam variabel priceWithDollar.

* return product.title.toLowerCase().contains(query) ||:

Memeriksa apakah judul produk (yang telah diubah menjadi huruf kecil) mengandung *search query*.

* (product.description?.toLowerCase().contains(query) ?? false) ||:

Memeriksa apakah deskripsi produk (jika ada dan sudah diubah menjadi huruf kecil) mengandung *search query*. Jika deskripsi *null*, maka akan mengembalikan false.

* priceString.contains(query) ||:

Memeriksa apakah representasi string dari harga produk mengandung *search query*.

* priceWithDollar.contains(query);:

Memeriksa apakah representasi string dari harga produk dengan tanda $ mengandung *search query*.

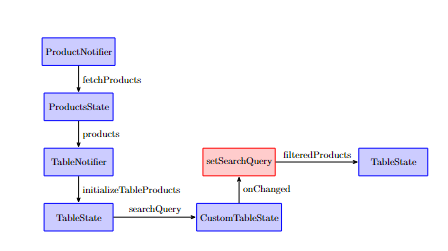
* }).toList();:

Mengkonversi produk yang sudah difilter menjadi daftar (*list*) dan mengembalikannya.

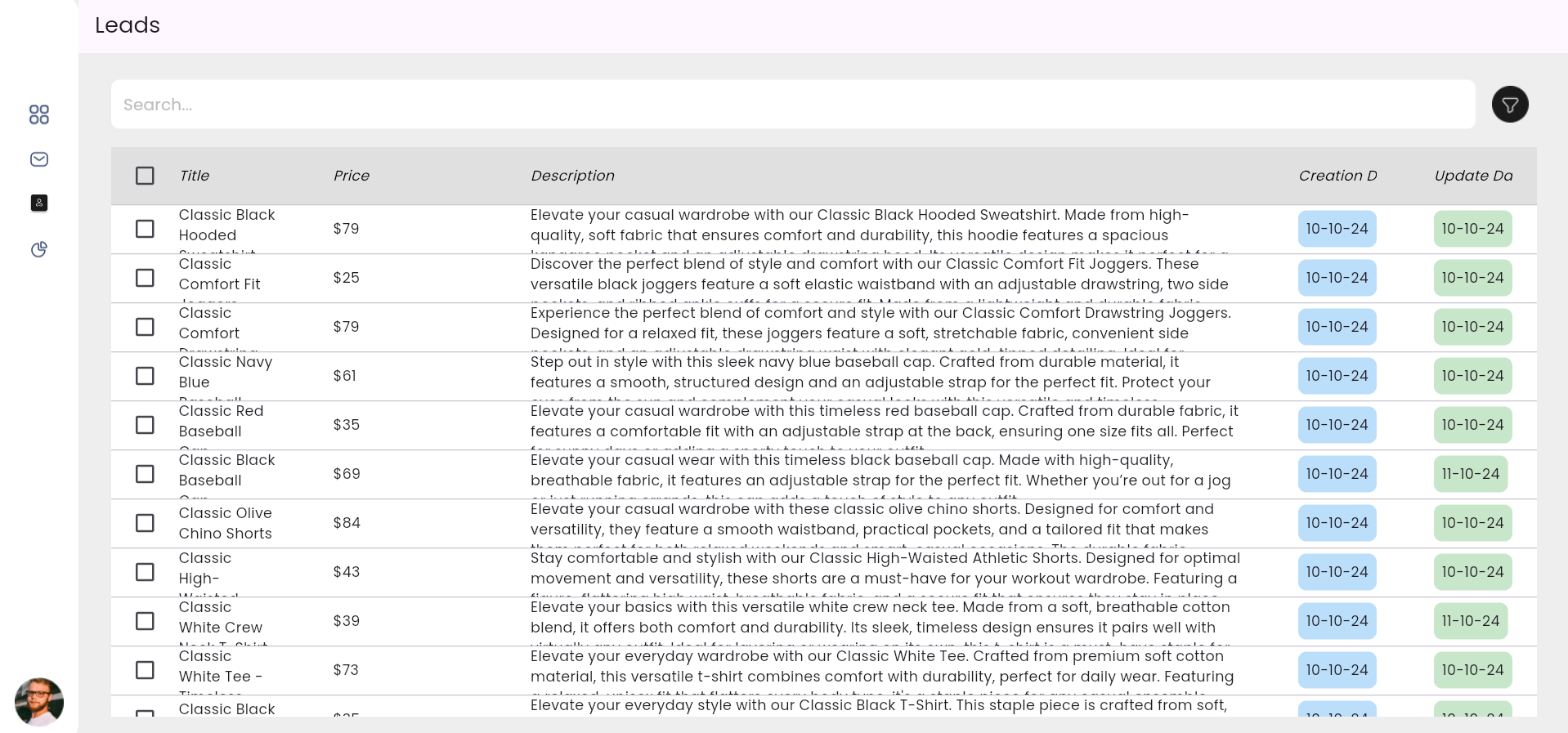


Summary

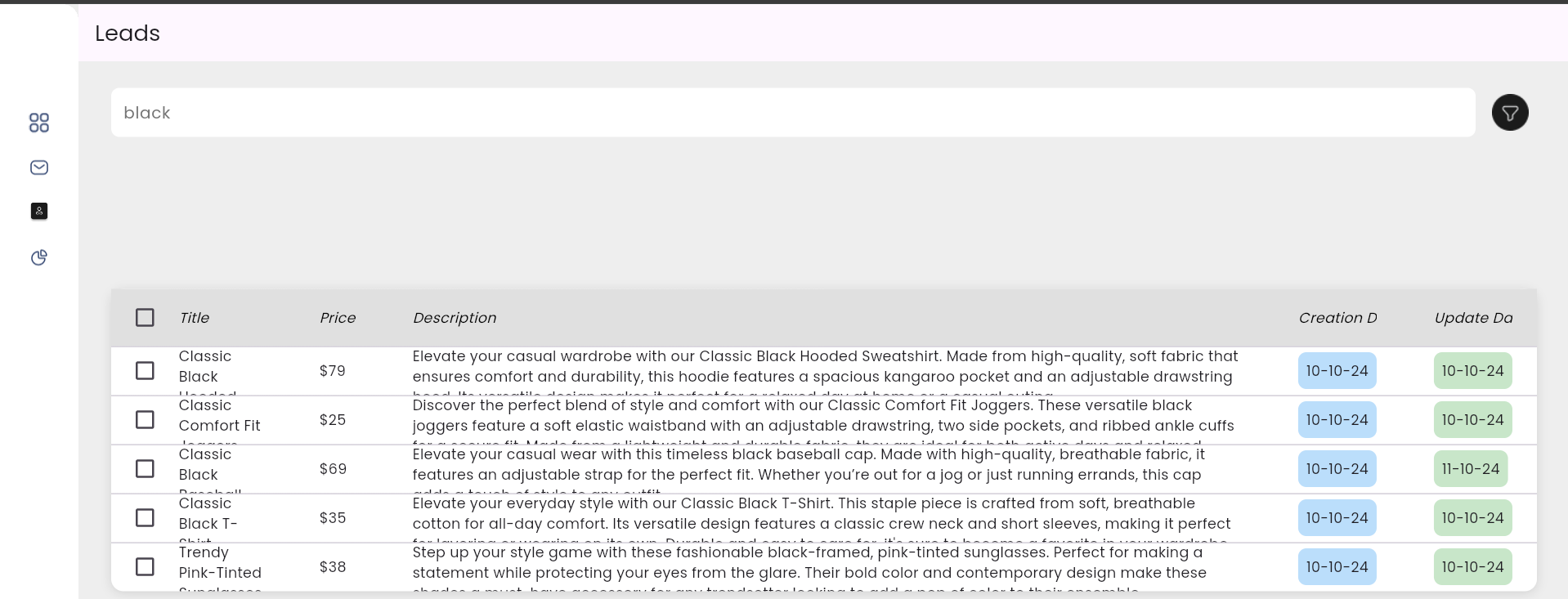
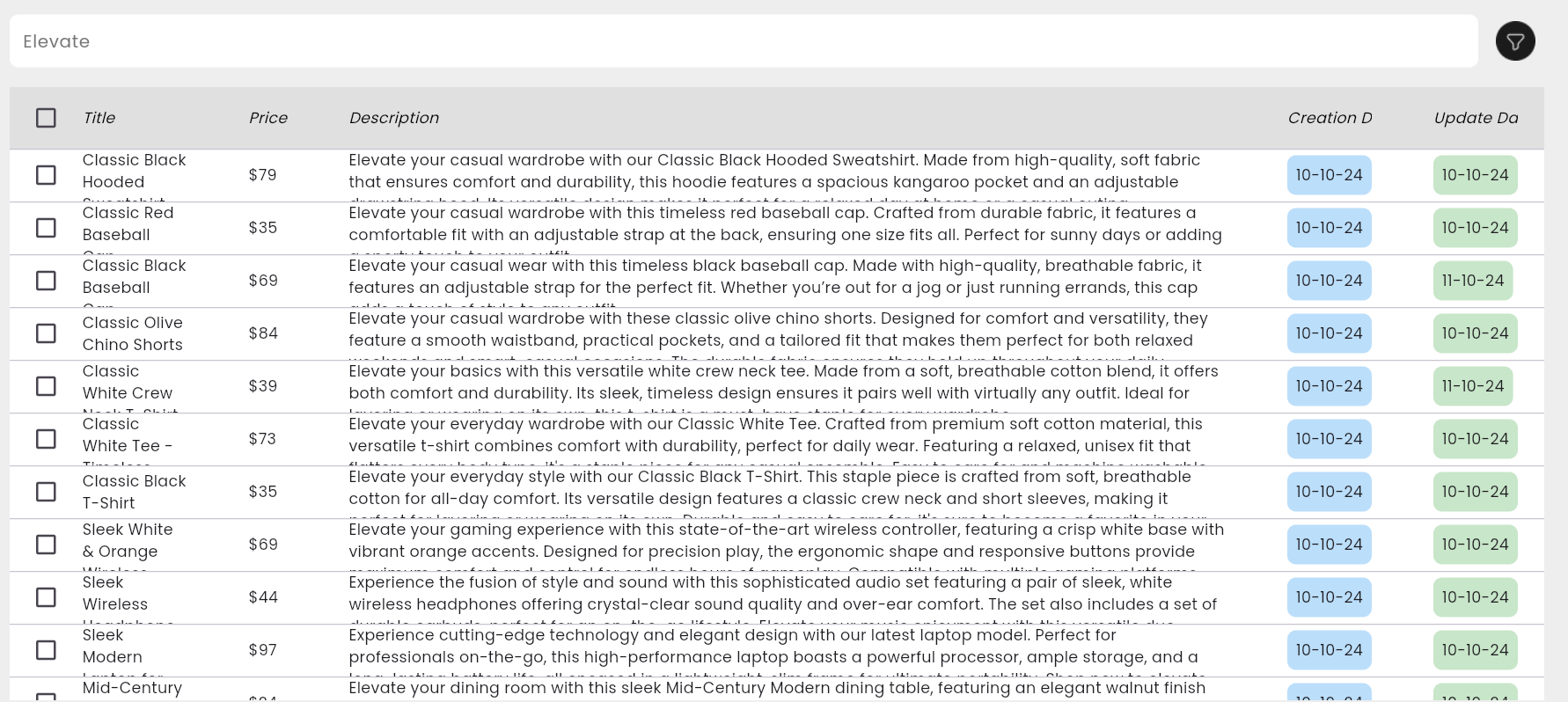
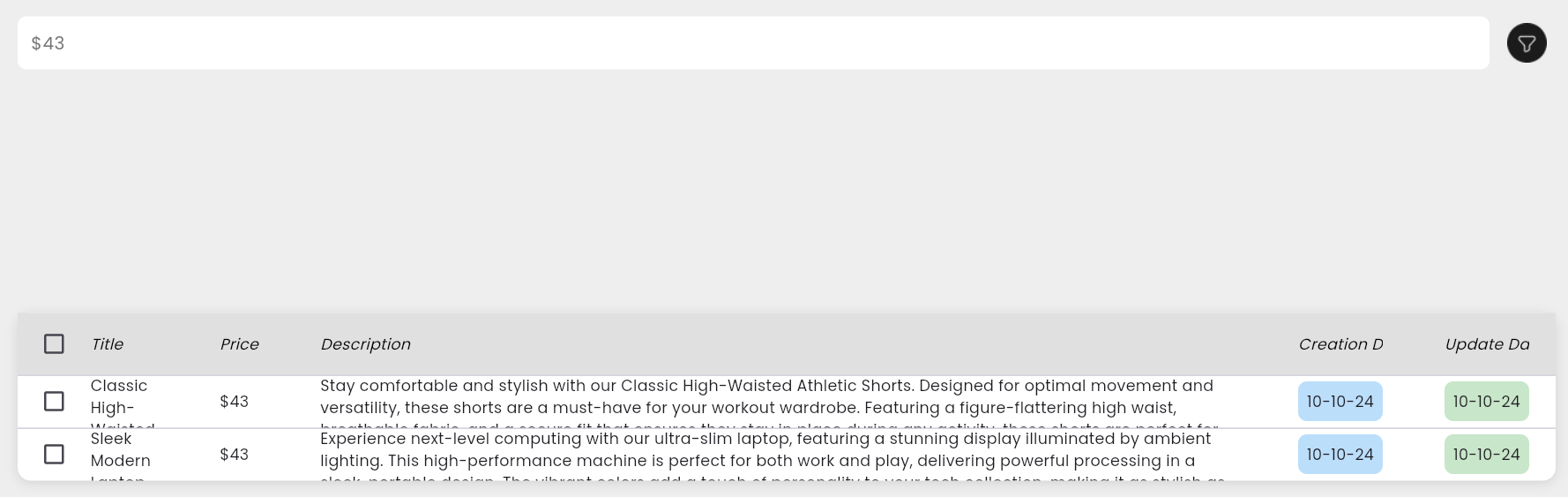
1. **Fetching Products**: The ProductNotifier mengambil produk dari repository dan memperbarui *state*-nya.
2. **Menginisialisasi Produk Tabel**: TableNotifier menginisialisasi *state*-nya dengan produk yang diambil oleh ProductNotifier.
3. **Input di Search Bar**: Kelas CustomTableState menangani input di *search bar* dan memperbarui *search query* di TableNotifier.
4. **Memfilter Produk**: TableNotifier memfilter produk berdasarkan *search query* dan menyediakan daftar produk yang sudah difilter.

  
  
**(*Flow Details*):**

* **ProductNotifier**: Mengambil produk dari repository.
* **ProductsState**: Menyimpan daftar produk yang diambil.
* **TableNotifier**: Menginisialisasi produk tabel menggunakan ProductsState.
* **TableState**: Menyimpan produk dan *search query*.
* **CustomTableState**: Mengelola input di *search bar*.
* **setSearchQuery**: Memperbarui *search query* di TableState.
* **filteredProducts**: Memfilter produk berdasarkan *search query*.

  
  
Hasil Uji :  
Typing di search bar menampilkan state data produk di memori tersebut yang di filter

berdasarkan isi dari Title, Description dan Price

1. Black (Title)  
   
2. Elevate (Description)  
   
3. $43 (Price dengan Dolar ‘$’ simbol)
4. 79 (Price tanpa Dolar ‘$’ simbol)

