

# TEST REPORT – Fitur Search, Filter, dan Sort Tokopedia

---

## 1. Judul Laporan

---

Test Report – Pengujian Fitur Search, Filter Harga, Filter Lokasi, dan Sort Termurah pada Sistem Pencarian Produk Tokopedia

## 2. Ringkasan Eksekutif

---

Laporan ini menyajikan hasil pengujian terhadap fitur pencarian produk, filter harga, filter lokasi, dan sorting harga termurah pada sistem pencarian mirip Tokopedia. Total terdapat 5 test case yang diuji menggunakan PyTest, dengan seluruh skenario berada pada kategori PASS. Pengujian dilakukan menggunakan pendekatan unit testing dengan mock API untuk memastikan setiap fungsi merespons input dengan benar tanpa bergantung pada koneksi eksternal.

## 3. Tujuan Pengujian

---

Tujuan dari pengujian ini adalah untuk memastikan bahwa seluruh fitur pencarian dan filter pada modul pencarian produk:

- Mengembalikan data produk sesuai keyword.
- Memberi pesan yang benar saat tidak ada produk ditemukan.
- Mengaplikasikan filter harga (min–max) dengan benar.
- Memfilter produk berdasarkan lokasi.
- Mengurutkan produk berdasarkan harga termurah.

Pengujian dilakukan untuk memvalidasi keandalan, konsistensi, dan ketepatan output pada setiap fungsi.

## 4. Lingkup Pengujian

---

Pengujian dilakukan terhadap file fungsi dan file test berikut:



## 4.1 File Fungsi

- search\_valid\_keyword.py
- search\_not\_found.py
- filter\_harga.py
- filter\_lokasi.py
- sort\_harga\_termurah.py

## 4.2 File Test

- test\_search\_valid\_keyword.py
- test\_search\_not\_found.py
- test\_filter\_harga.py
- test\_filter\_lokasi.py
- test\_sort\_harga\_termurah.py

**Total fitur diuji:** 5

**Total test case:** 5

## 5. Software dan Tools yang Digunakan

---

### 5.1 Bahasa & Library

- Python 3
- Requests
- PyTest
- PyTest-Mock / mocker

### 5.2 Tools Pengembangan

- Visual Studio Code / PyCharm
- Terminal / Command Prompt
- Virtual Environment (venv/conda)

## 6. Lingkungan Pengujian (Testing Environment)

---

- **Sistem Operasi:** Windows 11
- **Python Version:** 3.10
- **PyTest Version:** 7.x



- **Requests Version:** 2.x
- **CPU:** Standar Development Machine

Semua pengujian dilakukan pada lingkungan lokal tanpa koneksi langsung ke API Tokopedia melalui teknik mocking.

## 7. Metodologi Pengujian

---

**Pengujian dilakukan dengan pendekatan:**

### 7.1 Unit Testing

- Setiap modul diuji secara terpisah menggunakan PyTest.

### 7.2 Mocking API

**Menggunakan:**

- `mock.patch("module.requests.get")`

**untuk meniru respons API, sehingga:**

- pengujian tidak tergantung internet,
- hasil respons dapat dikontrol,
- validasi URL bisa dilakukan secara tepat.

### 7.3 Black-box Testing

- Fokus pada input dan output tanpa melihat implementasi internal.

## 8. Hasil Pengujian Per Fitur

---

### 8.1 Search Valid Keyword

- **Input:** "laptop"
- **Output:** Mengembalikan daftar produk.
- URL dipanggil dengan benar.
- **Status:** PASS

### 8.2 Search Product Not Found

- **Input:** keyword tidak valid
- **Output:** Products = [] + pesan "Produk tidak ditemukan"



- **Status:** PASS

### 8.3 Filter Harga

- **Input:** min\_price = 4.000.000, max\_price = 8.000.000
- URL menggabungkan parameter dengan benar.
- Produk berada dalam range harga.
- **Status:** PASS

### 8.4 Filter Lokasi

- **Input:** "Bandung"
- Sistem menampilkan produk dari lokasi tersebut.
- URL dipanggil dengan parameter &city=Bandung
- **Status:** PASS

### 8.5 Sort Harga Termurah

- **Input:** sort="termurah"
- Produk diurutkan ascending berdasarkan harga.
- URL mengandung sort=price\_asc
- **Status:** PASS

## 9. Rekapitulasi Test Case

NO	TEST CASE	FITUR	EXPECTED RESULT	ACTUAL RES-ULT	STATUS
1	TC-Search-Valid	Search	Produk tampil	Sesuai	PASS
2	TC-Search-NotFound	Search	Produk kosong + pesan	Sesuai	PASS
3	TC-Filter-Harga	Filter Harga	Produk sesuai range	Sesuai	PASS
4	TC-Filter-Lokasi	Filter Lokasi	Produk sesuai kota	Sesuai	PASS
5	TC-Sort-Termurah	Sorting Harga	Urutan ascending	Sesuai	PASS

## 11. Temuan (Findings)

---

Tidak ditemukan kegagalan maupun anomali pada semua skenario pengujian. Semua fungsi merespons input sesuai spesifikasi.