

ANALISIS KECEPATAN METODE PENYORTIRAN DAN PEMROSESAN YANG DITERAPKAN PADA PROGRAM KERANJANG BELANJA

Adamsyach Prana Walidin, Fahra Pebiana Putri, Nazwar Farezi, Najwatul Khoiriah, Fanny Ramadhani

Ilmu Komputer, Universitas Negeri Medan

Jalan Willem Iskandar, Pasar V Medan Estate, Percut Sei Tuan, Deli Serdang, Indonesia

adamsyachpranawalidin@gmail.com

ABSTRAK

Kecepatan adalah salah satu aspek yang sangat diperlukan dalam teknologi saat ini. Kecepatan biasanya menjadi tolak ukur sebuah sistem atau perangkat, semakin cepat semakin baik. Efisiensi sebuah sistem atau perangkat ditentukan oleh kecepatan sistem itu sendiri ketika memproses atau mengolah data. Pada zaman yang mewajibkan tiap hal berjalan secara singkat dan praktis, diperlukan pengujian terhadap berbagai macam hal terutama pada bidang teknologi struktur data. Untuk mengetahui seberapa efisien teknik struktur data list dan dictionary maka dilakukan implementasi pada sebuah program keranjang belanja sederhana untuk membuktikan dan mendapatkan angka pasti dalam hal kecepatan. Dengan menggunakan metode eksperimental kuantitatif, program dikembangkan dalam bahasa python dengan GUI dari library tkinter dan modul penghitungan waktu dari library time. Diketahui bahwa penggunaan dan kombinasi list dan kamus pada program keranjang belanja relatif cepat dalam memproses data, dengan rentang waktu 0.00007 detik hingga 1.8 detik dalam memproses setiap variabel. Hal ini membuat list dan kamus relatif efisien dalam penggunaannya pada program shopping cart.

Kata kunci : Struktur Data, List, Dictionary, Kecepatan

1. PENDAHULUAN

Pengelolaan penggunaan struktur data merupakan sebuah hal krusial yang sangat dibutuhkan di dalam sektor manapun mulai dari pendidikan sampai perdagangan. Terutama pada sektor perdagangan, atau lebih dikenal dengan sebutan *e-commerce* sudah melekat menjadi kebutuhan sehari-hari beberapa kalangan. Dengan pentingnya bidang ini dalam kehidupan, dibutuhkan pengelolaan data yang cepat serta efisien dalam mengelola setiap aktivitas serta kondisi yang ada pada *e-commerce*.

Pada platform *e-commerce*, data yang ditentukan untuk suatu produk bisa sangat beragam. Misalnya, untuk laptop, pembeli bisa mendapatkan informasi spesifikasi laptop seperti Model Laptop, Merek, Tipe Prosesor, Kecepatan Prosesor, Turbo Boost Processor, Tipe Video Graphics Array (VGA), Kapasitas VGA, Ukuran Layar, Kapasitas Random Access Memory (RAM), Harddisk, Solid State Disk, Sistem Operasi, Sidik Jari, Kamera, dan Harga. Informasi yang didapat dari aplikasi tersebut mungkin masih belum membantu sebagian orang untuk memutuskan membeli laptop yang sesuai dengan kebutuhannya. Pembeli biasanya menginginkan laptop yang sesuai dengan spesifikasi dan anggaran yang mereka inginkan. Misalnya, kebutuhan desain grafis membutuhkan VGA dengan performa tinggi untuk mengoperasikan laptop secara optimal. Di sisi lain, untuk kebutuhan perkantoran, kapasitas hard disk yang besar dibutuhkan untuk menyimpan data dalam jumlah yang banyak. Oleh karena itu, algoritma pencarian dengan struktur data yang sesuai diperlukan untuk mencapai waktu pencarian dan penyortiran yang efisien. [1]

List dan Dictionary merupakan jenis struktur pengelolaan data yang populer di banyak kondisi dan situasi. Selain mengelola data, list dan dictionary

memiliki keterkaitan dengan adanya algoritma penyortiran yang membuat kedua hal ini tidak dapat dipisahkan.

List adalah struktur data linier yang digunakan untuk menyimpan elemen-elemen dalam urutan tertentu dan bersifat dinamis, memungkinkan penambahan, penghapusan, atau modifikasi elemen dengan mudah. Elemen dalam list dapat berupa berbagai jenis data dan diakses menggunakan indeks yang dimulai dari nol. List dilengkapi dengan operasi bawaan seperti *append*, *insert*, *remove*, dan *index* yang membuatnya serbaguna. Namun, untuk data dalam jumlah besar, performa list bisa kurang optimal dibandingkan struktur data lain seperti *set* atau *dictionary*. Oleh karena itu, meskipun fleksibel dan mudah digunakan, pemilihan list harus mempertimbangkan kebutuhan dan karakteristik data yang dikelola.

Dictionary adalah struktur data yang menyimpan pasangan *key-value*, di mana setiap *key* unik dan digunakan untuk mengakses nilai terkait. Tidak seperti list, elemen dalam dictionary diakses berdasarkan *key* dan tidak memiliki urutan tetap. Struktur ini sangat efisien untuk operasi pencarian, penyimpanan, dan penghapusan data dengan kompleksitas waktu rata-rata $O(1)$. Dictionary sering digunakan untuk memetakan data atau mengelola informasi berbasis identitas unik. Meskipun membutuhkan lebih banyak memori dan tidak cocok untuk pengolahan data berurutan, dictionary unggul dalam pengelolaan data yang memerlukan akses cepat dan fleksibilitas tinggi.

Penelitian ini menganalisis seberapa efisien dan cepat pengolahan data yang dikemas dalam bentuk program keranjang belanja menggunakan python dengan modul waktu disertai dengan *Graphical User Interface* (GUI) dari library tkinter.

2. TINJAUAN PUSTAKA

2.1. Konsep Dasar Struktur Data

Struktur data adalah metode untuk menyimpan atau merepresentasikan data di dalam komputer sehingga dapat digunakan secara efisien. Data sendiri merupakan representasi dari fakta di dunia nyata, yaitu informasi atau keterangan mengenai suatu kenyataan yang disimpan, direkam, atau diungkapkan dalam bentuk tulisan, suara, gambar, sinyal, atau simbol. [2]

2.2. List dan Dictionary

List adalah tipe data yang paling serbaguna yang terdapat pada bahasa Python, yang dapat ditulis sebagai daftar nilai yang dipisahkan koma (item) antara tanda kurung siku. Hal penting tentang daftar adalah item dalam list tidak boleh sama jenisnya. [3]

Dictionary adalah struktur data dalam Python yang digunakan untuk menyimpan data dalam bentuk pasangan kunci (key) dan nilai (value). Setiap kunci bersifat unik, sementara nilai dapat berupa tipe data apa pun. Dictionary dideklarasikan menggunakan tanda kurung kurawal {} dan elemen-elemen di dalamnya dipisahkan dengan tanda koma, dengan format "kunci": "nilai". Dictionary bersifat mutable, sehingga memungkinkan penambahan, pengubahan, dan penghapusan elemen setelah dideklarasikan. Contoh penggunaan meliputi pengaksesan nilai berdasarkan kunci, iterasi menggunakan perulangan, dan penghapusan elemen menggunakan metode seperti pop(). [4]

2.3. Visual Studio Code (VS Code)

Visual Studio Code adalah editor teks serbaguna dan ringan yang dikembangkan oleh Microsoft untuk berbagai platform, termasuk Linux, macOS, dan Windows. Editor ini secara native mendukung bahasa pemrograman seperti JavaScript, TypeScript, dan Node.js, serta memungkinkan penggunaan bahasa pemrograman lainnya seperti C++, C#, Python, Go, Java, dan PHP dengan bantuan ekstensi yang dapat diunduh melalui marketplace Visual Studio Code. [5]

Visual Studio Code dibangun menggunakan Electron, sebuah framework yang mengintegrasikan Chromium dan Node.js ke dalam aplikasi mandiri. Oleh karena itu, ekstensi di VS Code memiliki kesamaan dengan aplikasi Node.js, termasuk kemampuan untuk mengimpor dependensi dari *Node Package Manager* (npm), yang menawarkan lebih dari satu juta paket. Tidak seperti ekstensi peramban web yang beroperasi dalam lingkungan terisolasi (*sandbox*), ekstensi VS Code memiliki akses penuh untuk memodifikasi antarmuka pengguna, menjalankan perintah shell, membaca dan menulis file pengguna, membangun koneksi jaringan, serta membuat server yang dapat menerima koneksi masuk. [6]

2.4. Graphical User Interface (GUI)

Graphical User Interface (GUI) adalah antarmuka pengguna yang memungkinkan interaksi

dengan komputer atau perangkat melalui elemen visual seperti ikon, tombol, dan jendela. Desain GUI yang menarik, intuitif, dan mudah digunakan sangat penting untuk keberhasilan produk atau layanan berbasis perangkat lunak. GUI secara signifikan meningkatkan produktivitas dan efisiensi di berbagai industri, berkat kemudahan navigasi dan interaksi yang ditawarkannya. Selain itu, GUI memiliki dampak ekonomi yang besar pada produk digital, terlihat dari banyaknya aplikasi yang tersedia, seperti 2,56 juta di Google Play Store dan 1,85 juta di App Store, yang mengandalkan antarmuka grafis untuk menarik dan mempertahankan pengguna. [7]

2.5. Bahasa Pemrograman Python

Python adalah salah satu perangkat lunak yang semakin populer karena kemampuannya yang luas, termasuk analisis data, perhitungan statistik kompleks, visualisasi data, implementasi algoritma pembelajaran mesin, manipulasi data, dan penyelesaian berbagai tugas matematika. Python unggul dalam memberikan hasil yang lebih akurat dan efisien dibandingkan dengan metode manual. Dalam pembelajaran matematika, terutama pada materi Permukaan Ruang, Python dapat digunakan untuk memvisualisasikan grafik fungsi dua variabel, memberikan kemudahan dalam memahami dan mengajarkan konsep matematika yang kompleks secara lebih interaktif dan efektif. [8]

2.6. Penelitian Terdahulu

- Judul: Performa Algoritma Bubble Sort dan Quick Sort pada Framework Flutter dan Dart SDK (Studi Kasus Aplikasi E-Commerce)
- Penulis: Dendi Rizka Poetra, Nur Hayati
- Ringkasan: Penelitian ini membahas performa algoritma Bubble Sort dan Quick Sort dalam konteks aplikasi e-commerce berbasis framework Flutter dan bahasa pemrograman Dart. Data untuk pengujian diambil melalui web scraping dari Tokopedia dengan keyword "kacamata," menghasilkan 2000 data. Pengujian dilakukan dalam lima iterasi dengan panjang data yang bervariasi. Hasilnya menunjukkan bahwa *Quick Sort* jauh lebih cepat dibandingkan *Bubble Sort* dalam hal waktu eksekusi, meskipun keduanya menunjukkan akurasi pengurutan yang identik sebesar 100%. *Quick Sort* membutuhkan waktu 0,011126 detik untuk 2000 data, sementara *Bubble Sort* memakan waktu 0,315543 detik. Penelitian ini menyimpulkan bahwa *Quick Sort* lebih efisien untuk pengurutan data dalam skenario aplikasi besar seperti e-commerce. [9]
- Kesamaan: Penelitian ini melakukan uji kecepatan demi mendapat dan membandingkan metode mana yang lebih efisien dalam skenario e-commerce. Penelitian ini juga mencantumkan hasil kecepatan yang diperoleh.

- e. Perbedaan: Penelitian ini melakukan uji kecepatan dalam metode struktur data penyortiran sedangkan penelitian yang dilakukan menguji pengelolaan dalam struktur data itu sendiri (*List dan Dictionary*).

3. METODE PENELITIAN

Pada bagian ini memuat metode yang digunakan pada pembuatan skripsi. Penelitian ini menggunakan metode eksperimental kuantitatif untuk melakukan uji kecepatan terhadap struktur data *list* dan *dictionary* serta memahami proses dan implementasinya dalam bentuk keranjang belanja yang berada dalam lingkup *e-commerce*.

3.1. Identifikasi Masalah

Penelitian ini fokus dalam mengidentifikasi kecepatan pengelolaan data dari struktur data *list* dan *dictionary* yang dikemas pada program keranjang belanja sebagai bingkai implementasinya pada bidang *e-commerce*. Penelitian ini bertujuan untuk mengatasi masalah efisiensi dalam pengelolaan dan penggunaan berbagai macam struktur data yang akan memberi dampak terhadap algoritma dibawah nya. Masalah yang dapat diidentifikasi dalam proses pemaparan data mengenai kecepatan olah struktur data ini ialah :

- Luasnya potensi berbagai macam struktur data dalam penggunaannya sehingga tidak dapat diidentifikasi dalam ruang lingkup tertentu.
- Tidak ada estimasi perkiraan mentah terhadap kecepatan pengelolaan struktur data tanpa adanya algoritma sorting.

Maka dari itu, dengan adanya data kecepatan struktur data *list* dan *dictionary* ini, tepat dan tidak tepatnya digunakan dalam ruang lingkup luas dapat diketahui serta bisa menjadi titik acuan dalam pengembangan yang lebih lanjut.

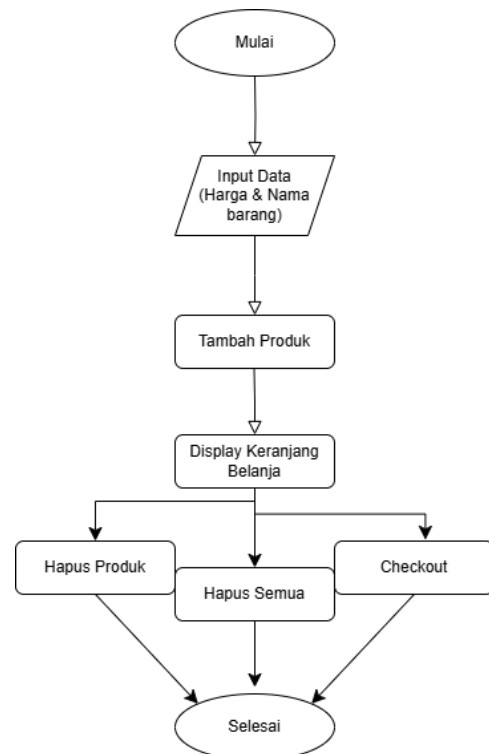
3.2. Perancangan Struktur Data List dan Dictionary

Berdasarkan identifikasi masalah, program keranjang belanja yang disusun menggunakan struktur data terkait dirancang untuk menampung data data dengan skema belanja online secara umum, namun dikarenakan berfokus pada pengolahan data, sampel yang digunakan juga dimasukkan secara manual. Dalam penelitian ini, *library python time* memainkan peran yang krusial sebagai pengukur kecepatan pengelolaan serta eksekusi dari struktur data terkait.

3.3. Desain Pembuatan Program Keranjang Belanja

Setelah dilakukan perancangan struktur data *list* dan *dictionary*, dilakukan proses perancangan serta pembuatan program keranjang belanja dengan menggunakan Bahasa pemrograman python pada VScode dengan *library time* dan *tkinter* sebagai acuan tampilan dan pengukuran.

Dalam proses desain dari program ini, didapat flowchart sebagai berikut.



Gambar 1. Flowchart Program

Pada bagian “mulai”, kode program akan mulai menginisiasi tampilan grafis untuk menampilkan GUI yang sudah dirancang. Setelah GUI muncul, kolom untuk melakukan input data yang terbagi atas dua tipe, yaitu nama barang dan harga, lalu tekan tombol tambah produk.

Setelah data sudah di input, akan ditampilkan pada display keranjang belanja lengkap dengan nama serta harga yang dipisahkan oleh “-“. Pada display ini, user dapat memilih tiga pilihan yang disediakan, yaitu “Hapus Produk”, “Hapus Semua”, “Checkout”. Pada tombol “Hapus Produk”, user hanya dapat menghapus satu produk data yang dipilih. Hal ini tidak berbeda jauh dengan tombol “Hapus Semua”, yang memiliki fungsi untuk menghapus setiap data yang berada pada display Keranjang Belanja. Tombol “Checkout” merupakan proses akhir untuk semua item atau data pada display tersebut yang akan menghasilkan total harga dari semua input data yang ada.

Setiap tahapan dari proses tersebut direkam dan diukur kecepatannya oleh *library time* yang akan ditampilkan pada terminal program tersebut.

3.4. Tahapan Implementasi Data

Program keranjang belanja yang digunakan untuk mengemas struktur data *list* dan *dictionary* pada penelitian ini mencakup tahap penginputan data secara manual agar dapat diukur kecepatannya oleh *library time*. Setiap data yang masuk dan keluar dalam ruang lingkup *list* dan *dictionary* akan diberi rincian waktu oleh *library time* pada terminal vscode dengan maksud tidak mengganggu GUI yang telah didesain. Berikut merupakan pembagian serta rincian dalam proses implementasi sampel data pada program:

3.4.1. Struktur Data List

List dalam penelitian ini berfungsi sebagai pengolah data saat sampel data di input kedalam program dengan dibagi pada output sebagai berikut:

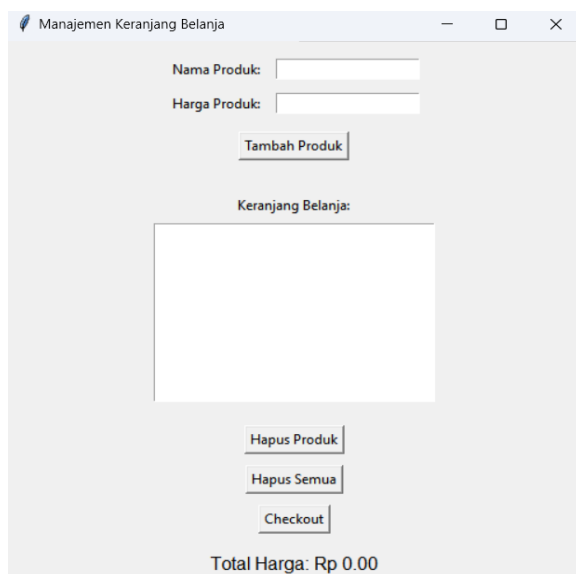
- Add_to_cart (item)**
Ini menandakan saat item sudah dimasukkan kedalam program.
- Add_to_cart (harga)**
Ini menandakan saat harga item sudah dimasukkan kedalam program.
- Update_display**
Ini menandakan jika semua proses sudah dialihkan pada *dictionary*.

3.4.2. Struktur Data Dictionary

Dictionary dalam penelitian ini berfungsi sebagai pengelola sampel data yang telah ditentukan sebelumnya oleh *list*, dengan bagina perintah sebagai berikut:

- Remove_from_cart**
Lingkup *dictionary* terbatas pada data yang telah diproses *list*, *Remove* bekerja pada proses hapus dan juga *checkout*.
- Update_cart_display**
Ini memberitahu user terhadap perubahan apa yang berubah pada daftar *dictionary*.

4. HASIL DAN PEMBAHASAN



Gambar 2. Tampilan GUI Program Keranjang Belanja

Hasil penelitian menunjukkan bahwa penggunaan struktur pengelolaan data, *list* dan *dictionary* memiliki waktu eksekusi yang relatif cepat dalam pengimplementasiannya pada program keranjang belanja. *List* dan *dictionary* ini juga memberi hasil input yang sesuai dengan program keranjang belanja pada umumnya.

User akan berinteraksi langsung terhadap GUI yang telah didesain untuk menjalankan proses pengukuran kecepatan. Ada beberapa tombol perintah seperti Tambah Produk, Hapus Produk, Hapus Semua,

dan checkout. Dapat diketahui jika susunan barang atau sampel yang tercantum pada kotak putih dapat diinteraksi secara langsung untuk mengakses pengelolaan data dari *dictionary*.

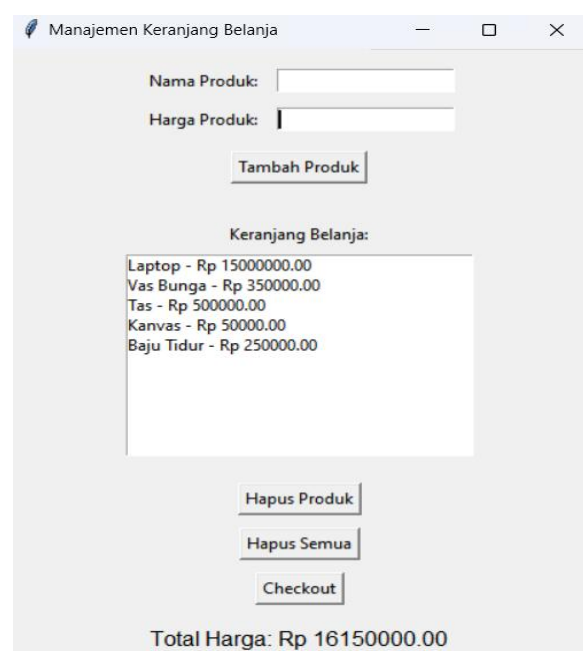
Library time dari pemrograman python akan mengukur kecepatan olah data dari *list* dan *dictionary* dalam satuan *second* atau detik pada terminal program di VScode.

Secara keseluruhan, program sederhana berbasis keranjang belanja yang didukung oleh *library* waktu python berhasil memberitahu analisis kecepatan setiap data input yang dimasukkan. Pada penerapannya, struktur data *list* bertindak sebagai pengelola data yang masuk sebelum disusun oleh struktur data *dictionary* pada kolom yang ada.

Tabel 1. Sampel Data yang Digunakan

No	Nama	Price
1	Laptop	15.000.000
2	Vas Bunga	350.000
3	Tas	500.000
4	Kanvas	50.000
5	Baju Tidur	250.000

Pada tabel diatas, disebutkan 5 sampel data yang digunakan sebagai data yang akan dikelola oleh *list* dan juga *dictionary*. Nama barang akan dimasukkan pada bagian nama produk beserta harga yang telah dicantumkan pada tabel di bagian Harga Produk. Dalam proses ini, *list* yang akan mengelola data masukan dari sampel sampel pada tabel diatas. User akan berinteraksi langsung dalam GUI yang telah didesain, proses saat input ini akan secara otomatis menjalankan *library time* yang digunakan untuk mengukur kecepatan, yang saat ini akan mengukur kecepatan dari struktur data *list*. Berikut merupakan tampilan GUI saat data sudah di input.



Gambar 3. Tampilan GUI saat sudah Input Data

Tabel 2. Tabel Waktu Eksekusi Struktur Data List

No	Waktu Eksekusi List		
	Add_to_cart (Item)	Add_to_cart (Harga)	Update_cart_display
1	0.00007 s	1.86145 s	0.00013 s
2	0.00014 s	0.00013 s	0.00009 s
3	0.00032 s	0.00024 s	0.00027 s
4	0.00016 s	0.00007 s	0.00011 s
5	0.00018 s	0.00016 s	0.00012 s

Tabel diatas menunjukkan hasil ukur kecepatan saat sampel data selesai dalam proses input, berurutan dengan tabel sampel data yang digunakan.

Tabel 3. Tabel Waktu Eksekusi Struktur Data Dictionary

No	Waktu Eksekusi Dictionary	
	Remove_from_Cart (Item)	Update_cart_display
1	0.00030 s	0.00024 s
2	0.00017s	0.000012 s
3	0.00013 s	0.00009 s
4	0.00013 s	0.00009 s
5	0.00019 s	0.00015 s

Tabel diatas didapat saat menghapus atau checkout data dari susunan yang telah didapat dalam GUI yang telah dikelola oleh struktur data Dictionary.

```
# fungsi pembungkus untuk mengukur waktu eksekusi
def measure_time(func):
    def wrapper(*args, **kwargs):
        start_time = time.perf_counter() # Mulai pengukuran waktu (float)
        result = func(*args, **kwargs) # Eksekusi fungsi
        end_time = time.perf_counter() # Akhiri pengukuran waktu (float)
        execution_time = end_time - start_time
        print(f"[INFO] Waktu eksekusi fungsi '{func.__name__}': {execution_time:.5f} detik")
        return result
    return wrapper
```

Gambar 4. Modul dari library time

Dengan menggunakan modul waktu dari python yang di implementasikan pada gambar diatas, dapat menghasilkan output perhitungan waktu yang dibutuhkan untuk mengolah data. Hasilnya akan ditampilkan pada terminal seperti gambar dibawah.

```
[INFO] Waktu eksekusi fungsi 'update_cart_display': 0.00012 detik
[INFO] Waktu eksekusi fungsi 'add_to_cart': 0.00015 detik
[INFO] Waktu eksekusi fungsi 'update_cart_display': 0.00011 detik
[INFO] Waktu eksekusi fungsi 'add_to_cart': 0.00017 detik
[INFO] Waktu eksekusi fungsi 'update_cart_display': 0.00024 detik
[INFO] Waktu eksekusi fungsi 'remove_from_cart': 0.00030 detik
[INFO] Waktu eksekusi fungsi 'update_cart_display': 0.00012 detik
[INFO] Waktu eksekusi fungsi 'remove_from_cart': 0.00017 detik
[INFO] Waktu eksekusi fungsi 'update_cart_display': 0.00009 detik
[INFO] Waktu eksekusi fungsi 'remove_from_cart': 0.00013 detik
[INFO] Waktu eksekusi fungsi 'update_cart_display': 0.00009 detik
[INFO] Waktu eksekusi fungsi 'remove_from_cart': 0.00013 detik
[INFO] Waktu eksekusi fungsi 'update_cart_display': 0.00015 detik
[INFO] Waktu eksekusi fungsi 'remove_from_cart': 0.00019 detik
```

Gambar 5. Hasil Perhitungan Modul Waktu

5. KESIMPULAN DAN SARAN

Kesimpulan yang didapat dari perhitungan dan analisa kecepatan struktur data list dan dictionary dalam bentuk shopping cart adalah setiap detik yang digunakan dalam waktu eksekusi mempengaruhi ukuran dan kompleksitas dari sebuah variabel yang akan dipaketkan dalam sebuah list atau dictionary. Dalam penerapannya pada python, lebih baik menggunakan library python seperti collection, heapq,

queue untuk memaksimalkan lebih banyak variasi dalam pengolahan data. Penggunaan dan kombinasi list dan dictionary pada shopping cart ini terbukti memiliki waktu kecepatan eksekusi yang cukup singkat dalam pengerjaannya, berkisar antara 0.00007 detik hingga 1.8 detik tergantung variabelnya. Pada penelitian di atas, waktu eksekusi terlama didapatkan oleh variabel harga laptop yaitu 15.000.000 dan waktu eksekusi terpendek didapatkan oleh variabel nama barang laptop.

Perlu diketahui bahwa pada penelitian ini, 5 item yang digunakan dapat ditambahkan dengan variabel lain seperti jumlah proses, aksesoris dan lain-lain untuk mendapatkan kombinasi yang lebih bervariasi. Penggunaan daftar dan kamus terbukti layak untuk digunakan dalam bidang ini. Dengan adanya penelitian ini, diharapkan dapat membantu penelitian yang sudah ada di masa yang akan datang. Area implementasi dan perluasan dari penelitian ini seperti perbandingan kecepatan dengan struktur data lain, pengembangan cara yang lebih efisien dalam bidang yang berbeda dan bervariasi, penerapan struktur data dalam algoritma pengurutan.

DAFTAR PUSTAKA

- [1] M. E. Al Rivan, "Perbandingan Kecepatan Gabungan Algoritma Utama Quick Sort dan Merge Sort dengan Algoritma Tambahan Insertion Sort, Bubble Sort dan Selection Sort," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 3, no. 2, pp. 319–321, Jul. 2017, doi:https://doi.org/10.28932/jutisi.v3i2.629.
- [2] D. Djayusman, I. Suhartini, F. N. Gunawan, N. T. Alhakim, and T. Gunawan A, "IMPLEMENTASI STRUKTUR DATA TREE PADA GAME SNAKE DENGAN C," *Jurnal Digit*, vol. 11, no. 1, p. 20, May 2021, doi:https://doi.org/10.51920/jd.v11i1.176.
- [3] Khoirudin, *Algoritma Dan Struktur Data Dengan Python 3*, 1st ed. Semarang: Universitas Semarang Press, 2019, pp. 34–41.
- [4] G. Prehandayana, W. Yahya, and H. Nurwasito, "Implementasi Struktur Data Dictionary Untuk Sistem Monitoring Perangkat Internet of Things," *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, vol. 2, no. 10, pp. 3467–3468, Oct. 2018, Accessed: Dec. 09, 2024. [Online]. Available: http://j-ptiik.ub.ac.id/
- [5] M. B. Nendya, B. Susanto, G. I. W. Tamtama, and T. J. Wijaya, "Desain Level Berbasis Storyboard Pada Perancangan Game Edukasi Augmented Reality Tap The Trash," *Fountain Informatics J.*, vol. 8, no. 1, pp. 1–6, 2023, doi: 10.21111/fij.v8i1.8836.
- [6] E. Lin, I. Koishybayev, T. Dunlap, W. Enck, and A. Kapravelos, "UntrustIDE: Exploiting Weaknesses in VS Code Extensions," *Proceedings 2024 Network and Distributed System Security Symposium*, pp. 1–2,

- 2024, doi: <https://doi.org/10.14722/ndss.2024.24073>.
- [7] E. Saputro, A. Syarief, and R. D. W. Utomo, "Graphical User Interface dalam Perspektif Pelindungan Desain Industri di Indonesia," *J. Seni Reka Ranc.*, vol. 6, no. 2, pp. 255–268, 2023, doi: 10.25105/jsrr.v6i2.17553.
- [8] Nurul Maulida Surbakti *et al.*, "Penggunaan Bahasa Pemrograman Python dalam Pembelajaran Kalkulus Fungsi Dua Variabel," *Algoritma*, vol. 2, no. 3, pp. 98–107, May 2024, doi: <https://doi.org/10.62383/algoritma.v2i3.67>.
- [7] D. R. Poetra and N. Hayati, "Performa Algoritma Bubble Sort Dan Quick Sort Pada Framework Flutter Dan Dart SDK(Studi Kasus Aplikasi E-Commerce)," *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, vol. 9, no. 2, pp. 806–816, Jun. 2022, doi: <https://doi.org/10.35957/jatisi.v9i2.1886>.