



Analisis dan Verifikasi Correctness Algoritma Perhitungan Total Pembayaran pada Sistem Checkout E-Commerce

Sofia Zahra¹, Sevta Triwana Simamora^{2*}, Andly Sofian Hasugian³, Adidtya Perdana⁴

^{1,2,3} Program Studi Ilmu Komputer, Universitas Negeri Medan

Email: ¹sofiazahra2005@gmail.com, ^{2*}sevtasimamora@gmail.com, ³dlyhasugian@gmail.com,
⁴adidtya@unimed.ac.id

Abstrak—Keandalan algoritma merupakan aspek krusial dalam pengembangan sistem komputasi, khususnya pada aplikasi transaksi digital seperti e-commerce. Penelitian ini bertujuan menganalisis dan memverifikasi correctness algoritma perhitungan total pembayaran pada sistem checkout e-commerce menggunakan pendekatan verifikasi formal. Implementasi dilakukan menggunakan bahasa pemrograman Python dan diuji menggunakan data transaksi nyata dari beberapa struk pembelian e-commerce. Analisis correctness meliputi pembuktian precondition, postcondition, loop invariant, partial correctness, dan termination. Hasil pengujian menunjukkan bahwa nilai total pembayaran yang dihasilkan algoritma sesuai dengan data transaksi pada seluruh kasus yang diuji tanpa ditemukan perbedaan. Keberadaan loop invariant pada proses perhitungan subtotal memastikan konsistensi hasil pada setiap iterasi. Penelitian ini memberikan kontribusi berupa pendekatan sistematis untuk memverifikasi kebenaran algoritma transaksi digital sebagai dasar pengembangan sistem pembayaran yang lebih andal dan terpercaya.

Kata kunci: kebenaran algoritma; verifikasi formal; e-commerce; checkout; kebenaran parsial; invarian perulangan

Abstract – *Algorithm reliability is a crucial aspect in the development of computational systems, particularly in digital transaction applications such as e-commerce. This study aims to analyze and verify the correctness of the algorithm used to calculate the total payment in an e-commerce checkout system using a formal verification approach. The implementation was carried out using the Python programming language and tested with real transaction data from several e-commerce purchase receipts. The correctness analysis includes the verification of preconditions, postconditions, loop invariants, partial correctness, and termination. The results show that the total payment values produced by the algorithm are consistent with the transaction data across all tested cases, with no discrepancies found. The presence of loop invariants in the subtotal calculation process ensures consistency of results at each iteration. This study contributes a systematic approach to verifying the correctness of digital transaction algorithms as a foundation for developing more reliable and trustworthy payment systems.*

Keywords: *algorithm correctness; formal verification; e-commerce; checkout; partial correctness; loop invariant*

1. PENDAHULUAN

Dalam pengembangan sistem komputasi modern, keandalan algoritma merupakan aspek penting untuk memastikan bahwa suatu sistem mampu menghasilkan keluaran yang benar sesuai dengan spesifikasi yang telah ditentukan. Algoritma yang tidak diverifikasi dengan baik berpotensi menghasilkan kesalahan perhitungan yang dapat berdampak pada kinerja maupun kepercayaan pengguna terhadap sistem tersebut. Oleh karena itu, diperlukan pendekatan sistematis untuk memastikan bahwa algoritma bekerja secara benar melalui proses verifikasi.

Verifikasi formal merupakan salah satu metode yang digunakan untuk membuktikan bahwa suatu algoritma atau program memenuhi sifat-sifat yang telah ditentukan berdasarkan model matematis. Pendekatan ini memungkinkan pengembang untuk menganalisis perilaku algoritma secara lebih mendalam sehingga dapat dibuktikan bahwa keluaran yang dihasilkan sesuai dengan spesifikasi yang diharapkan. Penelitian mengenai verifikasi formal menunjukkan bahwa metode ini dapat digunakan untuk memastikan kebenaran berbagai jenis algoritma, termasuk algoritma yang melibatkan struktur perulangan dan invarian matematis dalam proses komputasi (Shukla & Yadav, 2025).

Selain itu, penerapan verifikasi formal juga telah digunakan dalam pengembangan dan pembuktian kebenaran berbagai struktur data dan algoritma kompleks. Sebagai contoh, metode verifikasi mekanis telah diterapkan dalam konstruksi algoritma suffix array untuk memastikan



JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 12 Tahun 2026
ISSN 3025-0919 (media online)
Hal 3056-3065

bahwa algoritma yang dihasilkan benar secara matematis dan sesuai dengan spesifikasi yang telah ditentukan (Cheung et al., 2025). Pendekatan serupa juga digunakan dalam berbagai bidang lain seperti sistem komputasi awan, kontrak pintar (*smart contracts*), serta desain sistem operasi untuk memastikan bahwa sistem yang dikembangkan memiliki tingkat keandalan yang tinggi (Cheung et al., 2025).

Lebih lanjut, kajian mengenai verifikasi formal menunjukkan bahwa teknik ini semakin banyak digunakan untuk memastikan konsistensi logika dalam sistem perangkat lunak, termasuk pada aplikasi yang berkaitan dengan proses bisnis dan transaksi digital. Verifikasi formal memungkinkan pengembang untuk memeriksa apakah suatu sistem memenuhi batasan dan aturan bisnis yang telah ditetapkan sehingga dapat meminimalkan kesalahan dalam proses operasional (Stoica & Stoica, 2024). Selain itu, pendekatan verifikasi juga digunakan dalam berbagai model komputasi lainnya, seperti agen berbasis probabilistik dan jaringan saraf probabilistik, guna memastikan bahwa sistem yang dikembangkan tetap memenuhi spesifikasi yang telah dirancang (Nguyen & Rakio, 2023).

Dalam konteks aplikasi bisnis digital, seperti sistem checkout pada e-commerce, algoritma perhitungan transaksi memiliki peran penting dalam menentukan total pembayaran yang harus dilakukan oleh pengguna. Proses perhitungan ini biasanya melibatkan berbagai komponen seperti harga barang, jumlah barang, ongkos kirim, diskon pengiriman, diskon voucher, dan biaya layanan. Apabila algoritma perhitungan tidak dirancang dengan benar, maka hasil yang diperoleh dapat menimbulkan ketidaksesuaian nilai transaksi yang berpotensi merugikan pihak pengguna maupun penyedia layanan. Oleh karena itu, diperlukan analisis terhadap correctness algoritma untuk memastikan bahwa proses perhitungan transaksi dilakukan secara benar sesuai dengan model matematis yang telah ditentukan.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk menganalisis correctness algoritma pada sistem checkout e-commerce melalui proses verifikasi menggunakan data transaksi sebagai contoh pengujian. Analisis dilakukan dengan membandingkan keluaran algoritma dengan hasil perhitungan matematis yang seharusnya sehingga dapat dibuktikan apakah algoritma memenuhi sifat partial correctness, yaitu algoritma menghasilkan keluaran yang benar selama kondisi input yang diberikan memenuhi asumsi yang telah ditentukan.

Selain itu, proses verifikasi juga dapat memberikan pemahaman yang lebih mendalam mengenai bagaimana algoritma bekerja serta bagaimana metode formal dapat digunakan untuk menjamin keandalan sistem komputasi (Mahmoud et al., 2024).

2. METODE

Penelitian ini menggunakan pendekatan analisis algoritma untuk mengkaji kebenaran perhitungan total pembayaran pada sistem checkout e-commerce. Pendekatan ini dipilih karena fokus penelitian terletak pada bagaimana algoritma menghasilkan nilai keluaran yang sesuai dengan spesifikasi perhitungan yang telah ditentukan.

Tahapan penelitian dilakukan secara bertahap, dimulai dari pengumpulan data transaksi, perancangan model algoritma, implementasi algoritma dalam bentuk program, hingga analisis correctness algoritma. Seluruh tahapan tersebut dilakukan untuk memastikan bahwa algoritma yang dirancang tidak hanya berjalan secara teknis, tetapi juga benar secara logika dan matematis.

2.1. Data Penelitian

Data yang digunakan dalam penelitian ini berupa data transaksi yang diperoleh dari beberapa struk pembelian pada sistem e-commerce. Data tersebut mencakup komponen-komponen utama dalam proses checkout, seperti harga barang, jumlah barang, ongkos kirim, diskon pengiriman, diskon voucher toko, diskon voucher platform, serta biaya layanan.

Data transaksi digunakan sebagai bahan pengujian algoritma untuk melihat apakah algoritma mampu menghasilkan total pembayaran yang sesuai dengan nilai pada transaksi sebenarnya. Dengan menggunakan data nyata, hasil analisis diharapkan lebih representatif terhadap kondisi sistem e-commerce yang sesungguhnya.

Data transaksi tersebut digunakan sebagai input dalam proses pengujian algoritma yang dirancang. Melalui data ini, proses perhitungan total pembayaran dapat disimulasikan sehingga



algoritma yang dibuat dapat diuji apakah mampu menghasilkan hasil perhitungan yang sesuai dengan kondisi transaksi yang sebenarnya.

Tabel 2.1. Data Transaksi (bagian 1)

No	Subtotal	Ongkir	D.Kirim	D.Toko
1	205000	40000	40000	0
2	39519	14000	14000	60
3	99999	15000	15000	5000
4	59900	35000	35000	1000
5	149999	24100	24100	0
6	105211	37000	37000	3000
7	110000	24300	24300	100
8	81000	12000	12000	0
9	135500	50000	50000	0
10	136999	34600	34600	0

Tabel 3.2. Data Transaksi (bagian 2)

No	D. Platform	Biaya Layanan	Total
1	24601	2637	183036
2	1973	2000	39486
3	14250	2000	82749
4	8835	1900	51965
5	15000	2500	137499
6	15332	1000	87879
7	13188	2000	98712
8	0	2000	83000
9	16260	1994	121234
10	16440	2006	122565

Keterangan: D = Diskon

2.2. Model Algoritma

Algoritma perhitungan total pembayaran dirancang berdasarkan komponen-komponen transaksi yang terdapat pada sistem checkout e-commerce. Perhitungan dilakukan secara bertahap mulai dari subtotal barang hingga total pembayaran akhir.

Secara matematis, algoritma dapat dinyatakan sebagai berikut:

$$\text{Subtotal} = \sum (\text{harga} \times \text{jumlah})$$

$$\text{Ongkir}_{\text{akhir}} = \text{ongkir} - \text{diskon}_{\text{pengiriman}}$$

$$\text{Total} = \text{Subtotal} + \text{Ongkir}_{\text{akhir}} + \text{Biaya}_{\text{layanan}} - \text{Diskon}_{\text{toko}} - \text{Diskon}_{\text{platform}}$$

Model ini digunakan sebagai dasar dalam implementasi algoritma dan analisis correctness.



2.3. Implementasi Model Algoritma

Algoritma yang telah dirancang kemudian diimplementasikan menggunakan bahasa pemrograman Python. Implementasi dilakukan dalam bentuk program yang mampu menerima input berupa beberapa jenis barang serta komponen transaksi lainnya.

Program ini dirancang untuk menghitung subtotal melalui proses iterasi terhadap daftar barang yang dibeli. Selanjutnya program menghitung ongkos kirim setelah diskon, serta menghitung total pembayaran dengan mempertimbangkan seluruh komponen transaksi.

Pseudocode Algoritma

Algoritma perhitungan total pembayaran dalam sistem checkout dapat direpresentasikan dalam bentuk pseudocode sebagai berikut:

Input :

n, harga [i], jumlah [i], ongkir, diskon_pengiriman, diskon_toko, diskon_platform, biaya_layanan

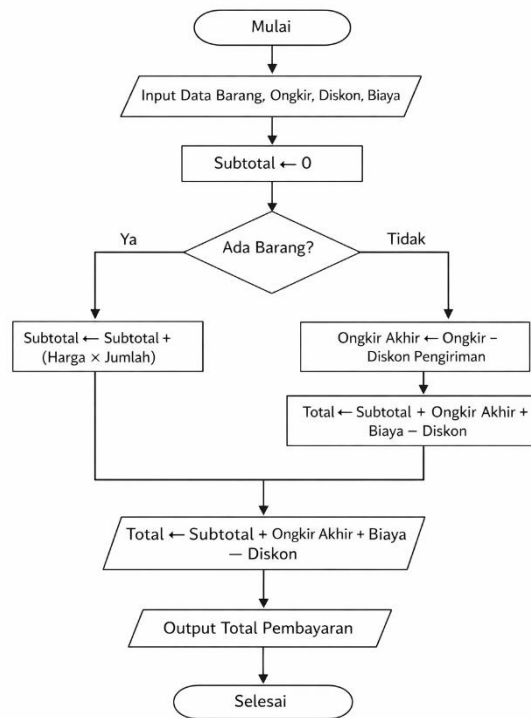
Output :

total_pembayaran

1. subtotal \leftarrow 0
2. for i \leftarrow 1 to n do
3. subtotal \leftarrow subtotal + (harga [i] \times jumlah [i])
4. endfor
5. ongkir_akhir \leftarrow ongkir - diskon_pengiriman
6. total_pembayaran \leftarrow subtotal + ongkir_akhir + biaya_layanan - diskon_toko - diskon_platform
7. // Verifikasi Precondition
8. if semua input \geq 0 then
9. status_precondition \leftarrow "Terpenuhi"
10. else
11. status_precondition \leftarrow "Tidak terpenuhi"
12. endif
13. //Verifikasi Postcondition
14. if total sesuai rumus then
15. status_postcondition \leftarrow "Terpenuhi"
16. else
17. status_postcondition \leftarrow "Tidak terpenuhi"
18. endif
19. return total_pembayaran

Flowchart Algoritma

Flowchart digunakan untuk memvisualisasikan alur kerja algoritma dalam menghitung total pembayaran pada sistem checkout.



Gambar 3.1. Flowchart Algoritma Perhitungan Total Pembayaran

Struktur Program

Program yang dikembangkan terdiri dari beberapa bagian utama, yaitu proses input data transaksi, proses perhitungan subtotal menggunakan perulangan, perhitungan ongkos kirim setelah diskon, serta perhitungan total pembayaran.

Cuplikan Program

Berikut merupakan sebagian kode program yang digunakan dalam perhitungan:

```
subtotal += harga * jumlah
```

Gambar 3.2. Kode perhitungan subtotal

Proses ini merupakan implementasi dari perhitungan subtotal menggunakan struktur iteratif, di mana setiap nilai harga barang dikombinasikan dengan jumlahnya, kemudian diakumulasikan ke dalam variabel subtotal. Mekanisme ini memastikan bahwa nilai subtotal selalu mencerminkan jumlah harga barang yang telah diproses pada setiap langkah perulangan.

Mekanisme akumulasi tersebut juga mendukung konsep loop invariant, di mana nilai subtotal pada setiap iterasi tetap merepresentasikan jumlah harga barang yang telah diproses.

```
ongkir_akhir = ongkir - diskon_pengiriman
```

```
total = subtotal + ongkir_akhir + biaya_layanan - diskon_toko - diskon_platform
```

Gambar 3.3. Kode perhitungan ongkos kirim dan perhitungan total pembayaran

Baris kode `ongkir_akhir = ongkir - diskon_pengiriman` digunakan untuk menghitung nilai ongkos kirim setelah dikurangi diskon pengiriman. Perhitungan ini bertujuan untuk menyesuaikan biaya pengiriman yang harus dibayarkan oleh pengguna berdasarkan potongan yang diberikan.



Dengan demikian, nilai ongkir yang digunakan dalam perhitungan total pembayaran merupakan nilai yang telah diperbarui sesuai dengan kondisi transaksi.

Baris kode total = subtotal + ongkir_akhir + biaya_layanan - diskon_toko - diskon_platform digunakan untuk menghitung total pembayaran akhir yang harus dibayarkan oleh pengguna. Perhitungan dilakukan dengan menjumlahkan subtotal barang, ongkos kirim akhir, serta biaya layanan, kemudian dikurangi dengan diskon yang berlaku, baik dari toko maupun platform. Rumus ini merepresentasikan model matematis yang digunakan dalam sistem checkout e-commerce.

Cuplikan Output Program

Menggunakan data 4

```
===== SISTEM CHECKOUT E-COMMERCE =====  
Masukkan jumlah jenis barang: 1
```

```
Barang ke- 1  
Harga barang: 59900  
Jumlah barang: 1
```

```
Subtotal barang: 59900.0  
Masukkan ongkir: 35000  
Masukkan diskon pengiriman: 35000  
Masukkan diskon voucher toko: 1000  
Masukkan diskon voucher platform: 8835  
Masukkan biaya layanan: 1900
```

```
===== RINCIAN PEMBAYARAN =====  
Subtotal : 59900.0  
Ongkir akhir : 0.0  
Biaya layanan : 1900.0  
Diskon toko : 1000.0  
Diskon platform : 8835.0  
-----  
Total pembayaran : 51965.0
```

```
===== VERIFIKASI CORRECTNESS =====  
Precondition terpenuhi  
Postcondition terpenuhi
```

Penjelasan Loop

Perhitungan subtotal dilakukan menggunakan struktur perulangan (loop) yang mengakumulasi nilai harga barang pada setiap iterasi. Setiap iterasi merepresentasikan satu jenis barang yang diproses dalam keranjang belanja.

Validasi Input

Dalam implementasi program, dilakukan validasi sederhana untuk memastikan bahwa seluruh nilai input tidak bernilai negatif. Hal ini bertujuan untuk mencegah terjadinya kesalahan perhitungan yang tidak logis.

2.4. Metode Analisis Data

Metode analisis yang digunakan dalam penelitian ini adalah analisis correctness algoritma. Analisis ini bertujuan untuk memastikan bahwa algoritma perhitungan total pembayaran pada sistem



checkout e-commerce mampu menghasilkan keluaran yang sesuai dengan spesifikasi yang telah ditentukan.

Pendekatan yang digunakan dalam analisis ini melibatkan beberapa komponen utama, yaitu precondition, postcondition, dan loop invariant. Precondition digunakan untuk mendefinisikan kondisi awal yang harus dipenuhi sebelum algoritma dijalankan, seperti nilai harga barang, jumlah barang, dan komponen transaksi lainnya yang tidak boleh bernilai negatif. Kondisi ini penting untuk menjamin bahwa proses perhitungan dilakukan pada data yang valid.

Selanjutnya, postcondition digunakan untuk memastikan bahwa hasil akhir yang diperoleh dari algoritma sesuai dengan model matematis yang telah dirancang. Dalam penelitian ini, nilai total pembayaran harus memenuhi hubungan antara subtotal, ongkos kirim setelah diskon, biaya layanan, serta potongan diskon yang diberikan.

Selain itu, analisis juga dilakukan terhadap loop invariant yang terdapat pada proses perhitungan subtotal. Loop invariant memastikan bahwa pada setiap iterasi, nilai subtotal yang dihasilkan selalu merepresentasikan jumlah harga barang yang telah diproses hingga iterasi tersebut. Dengan demikian, pada akhir proses perulangan, nilai subtotal yang diperoleh merupakan jumlah keseluruhan harga barang yang dibeli.

Untuk memperkuat analisis, dilakukan pembuktian partial correctness, yaitu memastikan bahwa algoritma menghasilkan keluaran yang benar apabila kondisi awal terpenuhi. Selain itu, dilakukan pula analisis termination untuk memastikan bahwa algoritma akan berhenti setelah seluruh proses perhitungan selesai dilakukan.

Melalui pendekatan tersebut, algoritma tidak hanya diuji dari sisi implementasi, tetapi juga dianalisis secara logis untuk memastikan bahwa setiap langkah perhitungan berjalan sesuai dengan prinsip kebenaran algoritma.

3. ANALISA DAN PEMBAHASAN

3.1. Hasil Implementasi

Algoritma perhitungan total pembayaran yang telah dirancang pada penelitian ini berhasil diimplementasikan dalam bentuk program menggunakan bahasa pemrograman Python. Implementasi ini dilakukan dengan mengacu pada model algoritma yang telah disusun sebelumnya, yang mencakup proses perhitungan subtotal, penyesuaian ongkos kirim setelah diskon, serta penggabungan seluruh komponen transaksi untuk memperoleh total pembayaran akhir.

Program yang dikembangkan mampu menerima input berupa beberapa jenis barang dengan harga dan jumlah yang berbeda. Setiap data barang diproses menggunakan struktur perulangan, sehingga nilai subtotal dihitung secara bertahap berdasarkan setiap item yang dimasukkan. Pendekatan ini memungkinkan sistem untuk menangani variasi jumlah barang secara fleksibel, sebagaimana yang terjadi pada transaksi e-commerce pada umumnya.

Setelah proses perhitungan subtotal selesai, program melanjutkan perhitungan pada komponen lain seperti ongkos kirim yang telah dikurangi diskon pengiriman, serta penambahan biaya layanan. Seluruh komponen tersebut kemudian dikombinasikan dengan pengurangan diskon voucher untuk menghasilkan nilai total pembayaran.

Berdasarkan hasil implementasi, program dapat dijalankan tanpa kendala dan mampu menghasilkan output yang sesuai dengan perhitungan yang diharapkan. Hal ini menunjukkan bahwa algoritma yang dirancang tidak hanya dapat diimplementasikan secara teknis, tetapi juga mampu merepresentasikan proses perhitungan transaksi secara logis.

3.2. Pengujian Algoritma

Pengujian algoritma dilakukan dengan menggunakan data transaksi yang diperoleh dari beberapa struktur pembelian pada sistem e-commerce. Data tersebut mencerminkan berbagai kondisi transaksi, mulai dari variasi jumlah barang, perbedaan harga, hingga kombinasi diskon yang digunakan. Dengan menggunakan data yang beragam, pengujian ini diharapkan dapat menunjukkan kemampuan algoritma dalam menangani berbagai kemungkinan kasus yang terjadi pada sistem checkout.

Tabel 4.1. Hasil pengujian algoritma perhitungan total pembayaran



No	Sub	OngA	Biaya	Diskon	Total
1	205000	0	2637	24601	183036
2	39519	0	2000	2033	39486
3	99999	0	2000	17250	82749
4	59900	0	1900	7935	51965
5	149999	0	2500	15000	137499
6	105211	0	1000	18332	87879
7	110000	0	2000	13388	98712
8	81000	0	2000	0	83000
9	135500	0	1994	16260	121234
10	136999	0	2006	16440	122565

Keterangan: Sub – Subtotal, OngA = Ongkir Akhir

Berdasarkan hasil pengujian tersebut, dapat dilihat bahwa nilai total pembayaran yang dihasilkan oleh algoritma sesuai dengan nilai pada masing-masing data transaksi. Tidak ditemukan perbedaan antara hasil perhitungan algoritma dengan nilai yang terdapat pada struk transaksi.

Hal ini menunjukkan bahwa algoritma mampu menghasilkan perhitungan yang konsisten dan akurat pada berbagai kondisi transaksi. Dengan demikian, algoritma dapat dikatakan memiliki tingkat keandalan yang baik dalam menangani proses perhitungan total pembayaran pada sistem checkout e-commerce.

3.3. Analisis Correctness Algoritma

Analisis correctness dilakukan untuk memastikan bahwa algoritma yang dirancang tidak hanya berjalan secara teknis, tetapi juga benar secara logika dan sesuai dengan spesifikasi yang telah ditentukan.

Precondition

Analisis correctness dilakukan untuk memastikan bahwa algoritma yang dirancang tidak hanya berjalan secara teknis, tetapi juga benar secara logika dan sesuai dengan spesifikasi yang telah ditentukan.

Loop Invariant

Loop invariant terdapat pada proses perhitungan subtotal yang menggunakan struktur perulangan. Pada setiap iterasi, nilai subtotal selalu merepresentasikan jumlah harga barang yang telah diproses hingga iterasi tersebut.

Dengan kata lain, sebelum iterasi berikutnya dilakukan, nilai subtotal sudah mencerminkan hasil perhitungan yang benar untuk data yang telah diproses sebelumnya. Ketika seluruh iterasi selesai, nilai subtotal yang dihasilkan merupakan jumlah keseluruhan harga barang yang dibeli.

Keberadaan loop invariant ini menunjukkan bahwa proses perhitungan dalam perulangan berlangsung secara konsisten dan tidak mengalami penyimpangan pada setiap langkahnya.

Postcondition

Postcondition dalam algoritma ini menyatakan bahwa nilai total pembayaran yang dihasilkan harus sesuai dengan model matematis yang telah dirancang. Nilai total diperoleh dari penjumlahan subtotal, ongkos kirim akhir, dan biaya layanan, kemudian dikurangi dengan diskon yang berlaku.



JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 12 Tahun 2026
ISSN 3025-0919 (media online)
Hal 3056-3065

Jika hasil perhitungan memenuhi hubungan tersebut, maka algoritma dapat dikatakan menghasilkan keluaran yang sesuai dengan spesifikasi yang telah ditentukan.

Partial Correctness

Berdasarkan hasil pengujian yang telah dilakukan, algoritma menghasilkan nilai total pembayaran yang sesuai dengan data transaksi pada seluruh kasus yang diuji. Hal ini menunjukkan bahwa algoritma memenuhi sifat partial correctness, yaitu mampu menghasilkan keluaran yang benar selama kondisi awal terpenuhi.

Termination

Algoritma menggunakan struktur perulangan yang bergantung pada jumlah barang yang dimasukkan. Karena jumlah iterasi bersifat terbatas, maka algoritma dipastikan akan berhenti setelah seluruh data barang diproses. Dengan demikian, algoritma memenuhi sifat termination.

3.4. Analisis Kesalahan

Meskipun algoritma telah dirancang dan diimplementasikan dengan baik, terdapat beberapa potensi kesalahan yang dapat terjadi dalam proses perhitungan. Salah satu kemungkinan yang dapat terjadi adalah nilai diskon yang melebihi subtotal atau ongkos kirim, yang dapat menyebabkan hasil perhitungan menjadi tidak sesuai.

Selain itu, kesalahan juga dapat disebabkan oleh input data yang tidak valid, seperti nilai negatif atau kesalahan dalam memasukkan jumlah barang. Kondisi ini dapat mempengaruhi hasil perhitungan dan menurunkan keakuratan sistem.

Oleh karena itu, diperlukan mekanisme validasi input untuk memastikan bahwa data yang digunakan dalam proses perhitungan berada dalam batas yang wajar dan sesuai dengan kondisi transaksi yang sebenarnya.

3.5. Implikasi

Keakuratan algoritma dalam menghitung total pembayaran memiliki peran yang sangat penting dalam sistem e-commerce. Algoritma yang mampu menghasilkan perhitungan secara konsisten dapat meningkatkan transparansi transaksi dan memperkuat kepercayaan pengguna terhadap sistem.

Selain itu, algoritma yang telah memenuhi sifat correctness dapat dijadikan sebagai dasar dalam pengembangan sistem yang lebih kompleks, seperti sistem pembayaran otomatis maupun integrasi dengan berbagai metode transaksi digital. Dengan demikian, hasil penelitian ini tidak hanya bermanfaat secara akademis, tetapi juga memiliki potensi untuk diterapkan dalam pengembangan sistem nyata.

4. KESIMPULAN

Berdasarkan hasil analisis dan pengujian yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut.

Pertama, algoritma perhitungan total pembayaran pada sistem checkout e-commerce berhasil diimplementasikan menggunakan bahasa pemrograman Python dan mampu menghasilkan nilai total pembayaran yang akurat dan konsisten pada seluruh 10 data transaksi yang diuji tanpa ditemukan perbedaan antara hasil algoritma dengan data transaksi nyata.

Kedua, hasil analisis correctness menunjukkan bahwa algoritma memenuhi precondition yang telah ditetapkan, yaitu seluruh nilai input yang digunakan dalam proses perhitungan bersifat non-negatif dan sesuai dengan kondisi transaksi yang valid. Loop invariant pada proses perhitungan subtotal juga terbukti terpenuhi, di mana nilai subtotal pada setiap iterasi selalu merepresentasikan jumlah harga barang yang telah diproses, sehingga memastikan konsistensi hasil perhitungan pada setiap langkah perulangan.

Ketiga, postcondition terpenuhi pada seluruh kasus pengujian yang dibuktikan dengan kesesuaian nilai total pembayaran yang dihasilkan algoritma terhadap model matematis yang telah dirancang. Algoritma terbukti memenuhi sifat partial correctness, yaitu mampu menghasilkan keluaran yang benar selama kondisi awal terpenuhi. Selain itu, algoritma juga memenuhi sifat



termination karena jumlah iterasi yang bersifat terbatas memastikan proses perhitungan akan selalu berhenti setelah seluruh data barang selesai diproses.

Secara keseluruhan, penelitian ini menunjukkan bahwa pendekatan verifikasi formal dapat diterapkan secara efektif untuk membuktikan correctness algoritma pada sistem transaksi digital, serta memberikan dasar yang kuat bagi pengembangan sistem pembayaran e-commerce yang lebih andal dan terpercaya.

REFERENCES

- A. M. Abdulaziz and K. Mehlhorn, "A formal correctness proof of Edmonds' blossom shrinking algorithm," *Journal of Automated Reasoning*, vol. 70, pp. 1–37, 2026.
- A. Bezza, R. Benaboud, T.-M. Maarouk, and R. Ameur-Boulifa, "Formal verification of the correctness and soundness of a Pomset-to-LTS transformation algorithm," *Journal of Communications Software and Systems*, vol. 21, no. 4, 2025.
- A. Jaszczak, "Partial correctness of an algorithm computing Lucas sequences," *Formalized Mathematics*, vol. 28, pp. 280–286, 2020.
- A. Muhammad et al., "Formal verification of transcompiled mobile applications using first-order logic," *Technologies*, vol. 13, pp. 1–19, 2025.
- C. Pilatte, "Unconditional correctness of recent quantum algorithms for factoring and computing discrete logarithms," *Forum of Mathematics*, pp. 1–22, 2026.
- C. Yang, S. Wu, and Q. Cao, "A formal framework for naturally specifying and verifying sequential algorithms," *arXiv*, 2025.
- F. Stoica and L. F. Stoica, "Formal verification of business constraints in workflow-based applications," *Information*, vol. 15, no. 12, 2024.
- G. Ernst, "A complete approach to loop verification with invariants and summaries," *arXiv*, 2021.
- H. N. Nguyen and A. Rakio, "Formal modelling and verification of probabilistic resource bounded agents," *Journal of Logic, Language and Information*, vol. 32, pp. 829–859, 2023.
- H. Wang and L. Song, "Study of Isabelle/HOL on formal algorithm analysis and code generation," *IEEE Access*, vol. 9, pp. 25002–25013, 2021.
- K. S. Namjoshi and L. D. Zuck, "Program correctness through self-certification," *Communications of the ACM*, vol. 68, no. 2, pp. 74–83, 2025.
- L. Cheung, A. Moffat, and C. Rizkallah, "Formally verified suffix array construction," *Journal of Automated Reasoning*, vol. 69, pp. 1–38, 2025.
- L. Nigro and F. Cicirelli, "Correctness verification of mutual exclusion algorithms by model checking," *Preprints.org*, 2024.
- L. Nigro and F. Cicirelli, "Formal verification of mutual exclusion algorithms in anonymous memory," *Applied Computing and Intelligence*, vol. 5, pp. 236–263, 2025.
- L. Nigro and F. Cicirelli, "Proving properties of Dekker's algorithm for mutual exclusion of N processes," *Algorithms*, vol. 18, no. 4, p. 226, 2025.
- L. S. K. G. Yaramolu, "Smart contracts in fintech: Revolutionizing financial transactions," *World Journal of Advanced Research and Reviews*, vol. 26, no. 1, pp. 4149–4159, 2025.
- M. A. S. Al-humaikani and L. B. A. Rahim, "A review on the verification approaches and tools used to verify the correctness of security algorithms and protocols," *International Journal of Advanced Computer Science and Application*, vol. 10, no. 6, 2019.
- N. Amit, S. Goldwasser, O. Paradise, and G. N. Rothblum, "Models that prove their own correctness," *arXiv*, 2025.
- R. Ait El Manssour, G. Kenison, M. Shirmohammadi, and A. Varonka, "Simple linear loops: Algebraic invariants and applications," *Proceedings of the ACM on Programming Languages*, vol. 9, pp. 745–771, 2025.
- R. Cohen and E. Feron, "Verification and validation of convex optimization algorithms for model predictive control," *Journal of Guidance, Control, and Dynamics*, vol. 7, pp. 78–83, 2020.
- Shukla and V. Yadav, "Formal modelling and verification of effective probabilistic neural networks for load balancing in cloud environment," *Discover Computing*, vol. 28, pp. 1–29, 2025.
- T. Ferariu, P. Wadler, and O. Melkonian, "Validity, liquidity, and fidelity: Formal verification for smart contracts in Cardano," *OpenAccess Series in Informatics (OASISs)*, vol. 6, pp. 1–21, 2025.
- Z. Sbai, "Model checking deep neural networks: Opportunities and challenges," *Frontiers in Computer Science*, vol. 7, 2025.
- Z. Wang, Y. Lan, X. He, and J. Lv, "A formal verification approach for Linux kernel designing," *Technologies*, vol. 12, no. 8, 2024.
- Z. YOU, C. ZHANG, H. SUN, and Z. ZUO, "Program derivation and mechanized verification of edit distance algorithm," *Wuhan University Journal of Natural Sciences*, vol. 30, no. 6, pp. 576–588, 2025.