

## Program Konversi Mata Uang dengan Pecahan Terkecil Menggunakan Algoritma Greedy dan String Matching

*Jeremy Michael<sup>\*</sup>, Berliano, Geo Kendra, I Putu Yoga Sastrawan*

*Fakultas Teknik, Program Studi Teknik Informatika, Universitas Palangka Raya, Palangka Raya, Indonesia*

### INFORMASI ARTIKEL

Diterima Redaksi: 18 Mei 2025  
Revisi Akhir: 07 September 2025  
Diterbitkan *Online*: 18 September 2025

### KATA KUNCI

Konversi  
Pecahan  
Greedy  
String Matching

### KORESPONDENSI (\*)

Phone: +62 813-4658-5707  
E-mail: [2430105030010@mhs.upr.ac.id](mailto:2430105030010@mhs.upr.ac.id)

### A B S T R A K

Dalam proses konversi mata uang asing, seringkali didapati bahwa adanya sebuah pecahan mata uang. Karena hasil konversi tidak pasti langsung dalam bentuk bilangan yang bulat. Penggunaan algoritma greedy berperan dalam memecah nilai uang hingga mencapai nilai terkecil yang optimal, lalu algoritma string matching sendiri berperan dalam melakukan pencocokan terhadap mata uang yang sudah ditentukan dimana diantaranya berupa IDR, USD, EUR, SGD, GBP. Implementasi dari kedua algoritma dilakukan untuk memberikan solusi efisien dalam menyelesaikan permasalahan penukaran mata uang menjadi pecahan terkecil dengan meminimalkan jumlah lembar uang berdasar nilai mata uang dari beberapa negara yang diperlukan. Implementasi ini dapat membantu dalam proses penukaran mata uang bagi pengguna. Hasil dari penelitian ini dapat digunakan sebagai dasar untuk pengembangan sistem yang lebih kompleks dalam manajemen penukaran mata uang.

### PENDAHULUAN

Perkembangan teknologi informasi dan globalisasi yang semakin pesat telah mendorong peningkatan aktivitas transaksi keuangan lintas negara. Dalam konteks ini, konversi mata uang menjadi salah satu aspek penting yang harus dilakukan secara cepat, akurat, dan efisien. Proses konversi tidak hanya melibatkan perubahan nilai tukar dari satu mata uang ke mata uang lain, tetapi juga perlu menghasilkan pecahan terkecil dari mata uang tersebut agar memudahkan pengguna dalam melakukan pembayaran secara praktis. Namun, kendala yang sering dihadapi adalah variasi penamaan dan ejaan mata uang asing yang berbeda-beda, sehingga memerlukan mekanisme pengenalan yang fleksibel dan akurat.

Untuk mengatasi permasalahan tersebut, pada artikel ini mengembangkan sebuah program sederhana yang mampu melakukan konversi antara mata uang Rupiah dan beberapa mata uang asing utama, yaitu United States Dollar (USD), Euro (EUR), Singapore Dollar (SGD), dan British Pound Sterling (GBP). Program ini menggunakan algoritma greedy untuk menentukan pecahan terkecil secara optimal sehingga pengguna dapat memperoleh hasil konversi yang mudah digunakan dalam transaksi sehari-hari. Selain itu, algoritma string matching diterapkan untuk memecahkan dan melakukan pencocokan dalam mengenali nama mata uang asing dengan variasi penulisan yang berbeda, sehingga proses identifikasi menjadi lebih akurat dan efisien [1]. Dengan adanya program ini tidak sekedar memberikan efisiensi terhadap konversi terhadap mata uang, program ini hadir dengan memberikan kemudahan dalam penggunaan antarmuka yang sederhana dan dapat memberikan hasil yang tentunya mudah untuk dipahami oleh pengguna dengan berbagai macam latar belakang. Program ini juga tentunya dapat menjadi wadah edukasi finansial entah bagi pelajar dan masyarakat umum agar dapat membantu mereka dalam memahami bagaimana sistem pecahan mata uang itu bekerja, sekaligus untuk meningkatkan kesadaran akan literasi keuangan.

Penggunaan algoritma greedy dalam program ini memungkinkan pemilihan pecahan mata uang secara optimal dengan cara yang sederhana dan cepat, sesuai dengan kebutuhan pengguna yang menginginkan solusi praktis tanpa harus menggunakan sistem yang kompleks. Sedangkan algoritma string matching membantu dalam pencocokan nama mata uang asing, mengatasi masalah perbedaan ejaan atau format penulisan yang sering ditemukan di berbagai sumber data. Dengan menggabungkan kedua algoritma tersebut, program ini diharapkan dapat memberikan kemudahan bagi pengguna dalam melakukan konversi mata uang Rupiah ke USD, EUR, SGD, dan GBP, serta sebaliknya, dengan hasil pecahan terkecil yang sesuai.

Nilai tukar yang digunakan dalam program ini bersifat statis untuk menyederhanakan proses perhitungan dan fokus pada mekanisme pembagian pecahan serta pengenalan nama mata uang. Pecahan mata uang yang digunakan adalah pecahan resmi yang berlaku di masing-masing negara, sehingga hasil konversi dapat langsung diaplikasikan dalam transaksi nyata. Program sederhana ini diharapkan dapat menjadi alat bantu yang efektif bagi pengguna yang membutuhkan solusi cepat dan mudah dalam konversi mata uang, terutama bagi pelaku usaha kecil, pelajar, dan masyarakat umum yang tidak memerlukan aplikasi konversi yang rumit.

Dengan demikian, penelitian ini bertujuan untuk mengembangkan program konversi mata uang yang mengintegrasikan algoritma greedy dan string matching dalam sebuah aplikasi sederhana namun fungsional. Program ini diharapkan dapat memberikan kontribusi nyata dalam mempermudah proses konversi mata uang lintas negara dengan hasil pecahan terkecil yang optimal dan pengenalan nama mata uang asing yang akurat.

## TINJAUAN PUSTAKA

### *Konversi Mata Uang Asing*

Konversi adalah proses perubahan bentuk, satuan, atau sistem menjadi bentuk lain [2]. Dalam ekonomi global, konversi mata uang asing sangat penting untuk memfasilitasi perdagangan internasional, investasi lintas negara, dan transaksi keuangan antar negara [2]. Nilai tukar mata uang asing menjadi dasar dari proses konversi ini [2]. Pemecahan mata uang asing menghasilkan pecahan nominal yang berbeda untuk setiap mata uang, tetapi juga perbedaan signifikan dalam stabilitas nilai tukar [2]. Mata uang yang stabil, seperti Dolar AS, sering digunakan sebagai alat pembayaran internasional [2]. Sebaliknya, mata uang yang kurang stabil jarang digunakan dalam transaksi internasional [2]. Nilai tukar mata uang dapat ditentukan melalui sistem nilai tukar tetap, mengambang, atau mengambang terkendali.

Dalam sistem nilai tukar tetap, pemerintah atau bank sentral campur tangan untuk menjaga nilai mata uang domestik [3]. Sistem nilai tukar mengambang memungkinkan nilai mata uang berfluktuasi berdasarkan penawaran dan permintaan pasar [4], tetapi pemerintah dapat melakukan intervensi untuk menghindari gejolak nilai tukar [3]. Pemahaman tentang konversi mata uang asing dan faktor-faktor yang memengaruhinya sangat penting bagi pelaku ekonomi, investor, dan pembuat kebijakan di era globalisasi saat ini.

### *Algoritma Greedy*

Algoritma Greedy ialah pemecahan masalah yang melakukan pendekatan heuristik, yang dimana berguna untuk memecahkan suatu masalah optimasi dalam pembuatan serangkaian lokal yang lebih baik. Dalam penerapan algoritma ini sendiri, keputusan yang diambil tanpa mempertimbangkan konskuensi jangka panjang, dengan harapan bahwa memberikan pilihan lokal yang optimal menghasilkan solusi pada suatu permasalahan yang baik. Pendekatannya sendiri pun banyak diterapkan diberbagai bidang termasuk pemrogram, jaringan, dan analisis data.

Algoritma ini umumnya memerlukan kondisi tertentu dalam menjamin solusi yang diberikan dan diperoleh mendekati optimal. Salah satu prekripsi penting ialah sifat submodular dari fungsi tujuan yang akan digunakan dalam proses penentuan suatu keputusan. Contohnya sendiri ialah penelitian yang dilakukan pada referensi [5, 6]. Menunjukkan bagaimana algoritma greedy dapat digunakan secara efisien dalam suatu permasalahan set covering dan penentuan suatu fitur dengan hasil yang baik dalam hal kualitas solusi yang dihasilkan. Hasil analisis yang dihasilkan oleh penelitian tersebut menunjukkan pendekatan yang sangat optimal dalam komputasi yang efisien. Lalu penelitian yang dilakukan referensi [7, 8]. Menunjukkan implementasi algoritma Greedy pada aplikasi penukaran uang Rupiah terbukti memudahkan proses pencarian solusi optimal dan mempercepat waktu penukaran, baik untuk keperluan perbankan maupun transaksi ritel seperti di mini market. Dalam konteks pemecahan mata uang, terletak pada kemampuan algoritma greedy dalam menentukan kombinasi pecahan uang yang paling sedikit jumlahnya. Dimana algoritma memilih pecahan terbesar yang mampu digunakan tanpa melibihi nilai yang ditukar, sehingga menghasilkan solusi optimal dengan jumlah pecahan terkecil. Maka pendekatan akan sangat sesuai dalam sistem mata uang seperti Rupiah maupun mata uang asing utama yang berjenjang, memungkinkan algoritma greedy untuk memberikan solusi yang optimal dan mempercepat proses penukaran uang.

### ***Algoritma String Matching***

Algoritma string matching didalam dunia pemrograman sendiri merupakan algoritma fundamental yang memiliki peran yang krusial dalam memproses suatu teks dan menganalisis data, serta dapat digunakan dalam konsep pengkonversian mata uang asing. Konversi mata uang asing melibatkan pengenalan dan pengalihan simbol mata uang asing, nilai tukar, dan format numerik dan dapat diketahui juga bahwa string matching dimanfaatkan untuk melakukan pengekstrakan pada sebuah informasi yang relevan berasal dari sumber data tak terstruktur maupun semi terstruktur [9]. Garis besarnya algoritma string matching mengusahakan melakukan penelusuran terhadap suatu keberadaan string tertentu (pola) pada string dengan ukuran yang lebih (teks) [9]. Tahapan pada komparasi pola (string yang ditelusuri) dan teks (proses lokasi pencarian string) dan memberikan sebuah tanda pada pola yang cocok dengan teks [9].

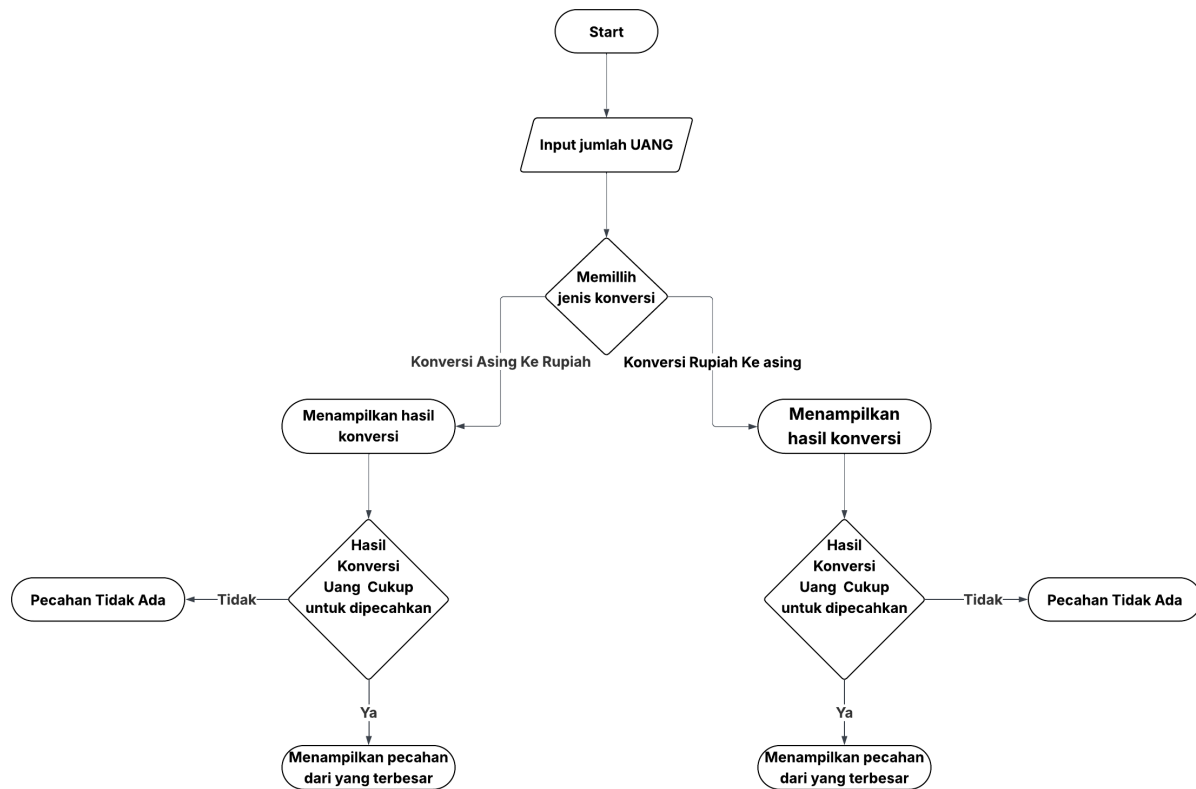
Dalam penggunaannya untuk konversi mata uang asing, algoritma dapat dipergunakan untuk mengenali nilai tukar dari sumber data keuangan dan memvalidasi input user [9]. Misalnya, algoritma string matching yang digunakan dalam melakukan pencarian nama mata uang tertentu (misal, USD, EUR, JPY) dan memvalidasi input user menyesuaikan dengan format mata uang yang diharapkan. Pada referensi [8], menunjukan bahwa algoritma string matching digunakan untuk mencari keberadaan nama "Indomaret" dalam daftar kota di Indonesia. Algoritma ini melakukan pencocokan pola dengan membandingkan karakter demi karakter secara sistematis, dan mencatat posisi kemunculan pola yang sesuai. Proses pencarian dapat dipercepat dengan praproses data, seperti mengubah teks menjadi huruf kecil atau menghilangkan karakter khusus. Dengan demikian dari penelitian yang dilakukan tersebut sama halnya dengan pencarian mata uang asing pada format nantinya menyesuaikan dengan data input. Keterkaitannya ialah terletak pada cara algoritma secara otomatis memproses dan mencocokkan input teks berupa nama atau simbol mata uang dengan berbagai variasi penulisan yang mungkin digunakan oleh pengguna. Memungkinkan sistem untuk mengenali dan memberikan validasi mata uang secara akurat meskipun terdapat perbedaan penulisan, atau variasi format yang umum ditemukan dalam input data.

### **METODOLOGI**

Pengembangan program konversi mata uang asing memanfaatkan algoritma string matching dan algoritma greedy. algoritma string matching digunakan untuk memvalidasi data mata uang. algoritma greedy dapat digunakan untuk mengoptimalkan proses pencarian dan ekstraksi informasi nilai tukar. Fokus penelitian memanfaatkan bagaimana kombinasi dari kedua algoritma mampu menghasilkan sistem yang optimal dalam memecahkan pecahan nominal mata uang asing yang terdapat.

Program konversi mata uang dikembangkan menggunakan bahasa pemrogram Python, yang dapat dikenal sebagai suatu bahasa serbaguna dan mudah diakses untuk pengembangan program. Dimana Python menyediakan berbagai macam jenis library dan modul yang mempermudah dalam pengembangan suatu program. Dalam hal ini, pengembangan menggunakan library Tkinter, yang merupakan library bawaan Python yang berfungsi dalam membangun GUI dekstop secara sederhana dan efisien yang memungkinkan dalam pembuatan tampilan interaktif dan ramah untuk pengguna.

**Diagram Alur Algoritma Greedy**

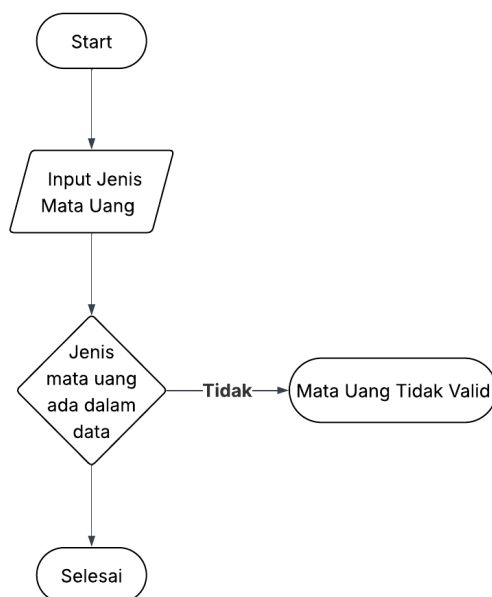


Gambar 1. Diagram Alur Algoritma Greedy

Gambar 1 yang tertera merupakan runtutan algoritma greedy bekerja dalam program untuk menguraikan nilai mata uang, algoritma ini selain digunakan untuk memecahkan mata uang yang di input oleh user juga mampu untuk menampilkan output banyaknya pecahan-pecahan yang terdapat pada mata uang yang telah dikonversi. Sebelum mengakses algoritma, user akan diminta untuk mengisi nominal jumlah uang dan diberi beberapa opsi apakah akan melakukan penukaran ke mata uang asing maupun sebaliknya sebelum memproses inputan dan memulai konversi. Terdapat beberapa pengkondisian yang dimana jika hasil konversi uang mampu untuk dipecahkan maka program dapat menampilkan program dan tidak dapat menampilkan pecahan jika hasil konversi tidak cukup dan otomatis mengosongkan bagian pecahan pada program nantinya dan sekedar menampilkan output berupa hasil konversinya saja.

**Diagram Alur Algoritma String Matching**

String matching akan berperan pada bagian utama program, sebelum program mengonversi user akan dimintai inputan berupa jenis mata uang yang ingin digunakan terlebih dahulu berdasar pada format mata uang yang telah ditentukan yaitu United States Dollar (USD), Euro (EUR), Singapore Dollar (SGD), dan British Pound Sterling (GBP).



Gambar 2. Diagram Alur Algoritma String Matching

Sebagaimana yang tertera pada gambar program akan menjalankan perintah memintah penginputan jenis mata uang asing, dan dilakukan beberapa pengkondisian dimana program jika mendeteksi adanya mata uang asing didalam data akan melanjutkan proses menuju algoritma greedy untuk melakukan pemecahan berdasar pada input jenis mata uang user. Sebaliknya jika pengkondisian inputan yang diberikan user pada program tidak relevan didalam data maka proses tidak dapat dilanjutkan, dan program secara otomatis memberikan peringatan untuk memasukan jenis mata uang yang sudah terdaftar di program.

## HASIL DAN PEMBAHASAN

Untuk program yang sudah diteliti dan dibuat, penulis disini menggunakan program python untuk pengimplementasian program konversi. Dipilihnya python dikarenakan kemudahan dalam penggunaan dan library yang melimpah. Penampilan untuk entry data akan menggunakan salah satu library yang sudah disediakan python berupa tkinter untuk menampilkannya dalam bentuk GUI (Graphical User Interface) untuk memudahkan entry data.

### *Algoritma String Matching Dalam Program*

Algoritma String Matching sederhana digunakan dalam proses pencarian mata uang asing yang akan digunakan. Pencocokan string penting untuk menemukan kemunculan sebuah pola dalam sebuah teks.

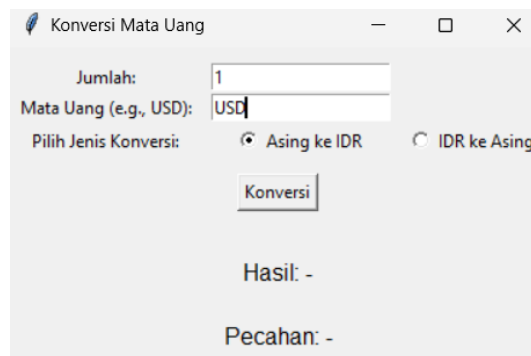
```

1 #Data Mata Uang
2 exchange_rates = {
3     "USD": 16000,
4     "EURO": 17000,
5     "SGD": 12000,
6     "GBP": 20000,
7 }
8 #Fungsi String Matching
9 def match_currency(input_currency):
10     input_currency = input_currency.upper()
11     if input_currency in exchange_rates:
12         return input_currency
13     return None
14
  
```

Gambar 3. Penerapan Algoritma String Matching

Dalam implementasi algoritma ini, string matching yang digunakan adalah yang sederhana dan tidak menggunakan teknik optimisasi. Fungsi string matching digunakan untuk memeriksa apakah mata uang yang dimasukkan oleh pengguna valid atau tidak. Secara umum, prosesnya dilakukan dengan cara mencocokkan (match) input mata uang pengguna dengan dictionary mata uang asing yang tersedia.

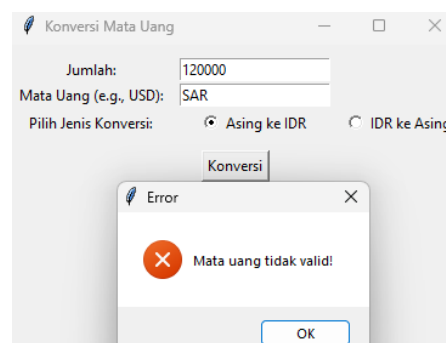
Program memeriksa apakah mata uang tersebut ada dalam kunci kamus. Jika ditemukan, mata uang dianggap valid dan proses konversi dapat dilanjutkan, sedangkan jika tidak ditemukan, program akan menampilkan pesan kesalahan bahwa mata uang tersebut tidak valid.



Gambar 4. Entry data dan jumlah uang

Seperti yang sudah dijelaskan, string matching yang diterapkan didalam program akan meminta input dari pengguna terlebih dahulu. Dimana akan disesuaikan dengan dictionary yang sudah ditentukan. Kemudian user akan diberi opsi apakah akan dikonversi ke asing maupun sebaliknya.

Keunggulan yang dimiliki dalam pengkonversian yang dilakukan ialah pencocokan pada dictionary yang sudah ditentukan, dimana mempermudah program dalam mengeksekusi dan menyesuaikan permintaan entry dari user untuk memilah jenis mata uang yang ingin digunakan.



Gambar 5. Entry data ketika tidak sesuai dengan isi kamus

Pada gambar tersebut merupakan pesan kesalahan yang ditampilkan program jika entry data dari user tidak sesuai dengan isi kamus string matching yang telah ditentukan.

### ***Algoritma Greedy Dalam Program***

Algoritma Greedy digunakan dalam proses pemecahan nilai uang, dimana program ini akan mencoba untuk menguraikan hasil konversi uang ke dalam pecahan-pecahan yang lebih kecil dengan cara yang efisien.

```

1 # Fungsi Greedy untuk memecah nilai uang
2 def greedy_currency_breakdown(amount, denominations):
3     breakdown = {}
4     for unit in sorted(denominations, reverse=True):
5         count = amount // unit
6         if count > 0:
7             breakdown[unit] = count
8             amount -= count * unit
9     return breakdown
10
11 # Fungsi untuk menampilkan hasil konversi dan pecahan uang
12 def show_conversion():
13     try:
14         amount = float(entry_amount.get())
15         currency = entry_currency.get().strip().upper()
16
17         if currency not in exchange_rates:
18             messagebox.showerror("Error", "Mata uang tidak valid!")
19             return
20
21         if amount <= 0:
22             messagebox.showerror("Error", "Masukkan jumlah yang lebih besar dari 0!")
23             return
24
25         if var_conversion_type.get() == 1: # Foreign to IDR
26             result = foreign_to_idr(amount, currency)
27             label_result.config(text=f"Hasil: {result:.2f} IDR")
28             breakdown = greedy_currency_breakdown(result, [100000, 50000, 20000, 10000, 5000, 2000, 1000])
29         else: # IDR to Foreign
30             result = idr_to_foreign(amount, currency)
31             label_result.config(text=f"Hasil: {result:.2f} {currency}")
32             breakdown = greedy_currency_breakdown(result, [100, 50, 20, 10, 5, 2, 1])
33
34         if breakdown:
35             breakdown_text = "\n".join([f"{k}: {v}" for k, v in breakdown.items()])
36             label_breakdown.config(text=f"Pecahan:\n{breakdown_text}")
37         else:
38             label_breakdown.config(text="Pecahan: Tidak ada")
39     except ValueError:
40         messagebox.showerror("Error", "Masukkan jumlah yang valid!")

```

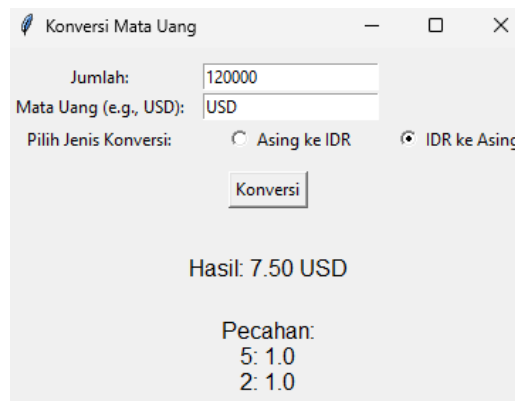
Gambar 6. Penerapan Algoritma Greedy

Pada algoritma ini, pada setiap langkah, program memilih pecahan terbesar yang dapat digunakan dari daftar denominasi yang ada untuk mengurangi jumlah uang yang tersisa. Pendekatan ini didasarkan pada prinsip Greedy, di mana solusi optimal dipilih secara lokal (memilih pecahan terbesar) tanpa mempertimbangkan solusi global di awal. Algoritma ini efektif ketika denominasi yang tersedia mengikuti aturan tertentu, seperti pecahan uang dalam sistem desimal (misalnya, 100, 50, 20, dll), yang memungkinkan pendekatan Greedy untuk menghasilkan solusi optimal.

Meskipun algoritma Greedy sering kali memberikan solusi optimal pada kasus seperti sistem mata uang desimal, penerapan algoritma ini bergantung pada struktur denominasi yang tersedia, di mana pecahan uang yang lebih besar akan lebih sering dipilih terlebih dahulu.

Gambar 7. Output hasil konversi mata uang menggunakan greedy

Dapat diketahui bahwa, entry data yang telah dimasukan oleh user dan disesuaikan sebelumnya menggunakan algoritma string matching maka program akan melanjutkannya dengan pemecahan menggunakan algoritma greedy. Dimana setelah user memilih opsi untuk mengonversi asing ke idr dan melakukan konversi, secara langsung program akan menampilkan pecahan-pecahan yang ada pada uang yang telah dikonversi tersebut.



Gambar 8. Output hasil konversi mata uang menggunakan greedy

Demikian juga konversi dalam bentuk mata uang asing dollar, program dapat menghitung pecahan yang dihasilkan berdasar pada hasil konversi yang telah didapatkan. Berlaku juga pada jenis mata uang asing jenis lainnya, yang membedakannya hanyalah pada nilai tukar mata uang yang berbeda pada masing-masing negara pada program, apakah pecahannya lebih banyak ataupun minim.

### ***Implementasi Algoritma Greedy dan String Matching Dalam Program Konversi Mata Uang***

Implementasi dari kedua algoritma didalam program secara efisien mampu bekerja dengan cepat dan memerlukan sumber daya yang relatif kecil, terutama data yang diproses berjumlah terbatas. Algoritma Greedy berperan dalam menghasilkan pecahan uang terkecil secara optimal melalui proses yang langsung dan tidak rumit. Sementara itu, algoritma String Matching memiliki kesederhanaan dalam penggunaannya dalam mengenali dan mencocokkan mata uang dari inputan pengguna. Namun, jika data yang diproses jauh lebih besar dan kompleks, penerapan algoritma pencarian yang lebih canggih seperti KMP atau Boyer-Moore akan lebih tepat karena memiliki kemampuan pencarian yang lebih cepat dan efisien dalam menangani volume data yang besar.

Pemilihan algoritma String Matching yang bersifat sederhana didasari pada kebutuhan mengenali sejumlah kecil mata uang dan variasi penulisan yang terbatas. Keunggulan dari algoritma ini terletak pada kecepatan dan kesederhanaannya, sehingga tidak memerlukan algoritma yang lebih rumit dan sumber daya yang besar. Pendekatan ini memastikan proses pencocokan mata uang berjalan secara cepat dan akurat, sehingga mendukung efisiensi serta kemudahan penggunaan dalam sistem konversi mata uang yang dikembangkan. Dengan demikian, algoritma sederhana ini sudah memadai untuk kebutuhan aplikasi yang bersifat praktis dan tidak berskala besar.

## **KESIMPULAN DAN SARAN**

Program konversi menggunakan perpaduan antara algoritma string matching dan greedy memiliki hasil yang optimal dalam melakukan pemecahan uang yang telah dikonversi. Dimulai dari string matching membuat efisiensi untuk melakukan pencarian pada jenis mata uang yang ingin dikonversi. Hanya dengan menggunakan algoritma yang sederhana dan memasukan jenis mata uang asing yang ada, program dapat mengeksekusi secara langsung. Namun keterbatasannya ialah pada pengisian kamus mata uang yang ingin dimasukan lebih banyak dari sebelumnya, memerlukan waktu lebih untuk mengisi mata uang yang ingin ditambahkan sehingga secara tidak langsung mengurangi efisiensi program itu sendiri. Sedangkan pada algoritma greedy dengan mudah melakukan pemecahan pada hasil konversi yang telah ditentukan oleh program, sehingga user mengetahui pecahan-pecahan yang terdapat pada hasil konversi. Keterbatasan pada penggunaannya sendiri, terdapat pada pemecahan mata uang asing dalam bentuk lain. Karena tidak spesifik memecahkan hingga ke bentuk lebih kecil. Contohnya seperti hasil daripada konversi mata uang rupiah ke dolar pada gambar sebelumnya, program hanya mengkalkulasi untuk bagian pecahan terbesarnya saja. Dengan demikian program ini diperuntukan bagi para pengguna yang membutuhkan suatu solusi sederhana dan cepat dalam melakukan konversi mata uang lintas negara dengan hasil pecahan terkecil yang optimal dan pengenalan nama mata uang asing yang akurat.

Jika program mampu dikembangkan lebih kompleks dari sebelumnya akan menghasilkan hasil yang jauh lebih optimal tentunya dan memaksimalkan fungsi daripada masing-masing algoritma yang digunakan, terlebih jika dapat dikembangkan dalam dunia perbankan akan menjadi solusi yang menjanjikan dalam mempermudah dan memaksimalkan kinerja. Pada bagian string matching jika menggunakan beberapa jenis mata uang tidak akan menjadi masalah dalam pengisian data mata uang asing yang ingin dikonversi, karena algoritma yang digunakan merupakan algoritma yang sangat sederhana maka mengurangi efisiensi pengisian jenis mata uangnya sendiri. Pada algoritma greedy optimalisasi pemecahan uang akan lebih baik jika dalam program dibuat mampu menghitung pecahan terkecil didalam beberapa mata uang asing yang terdapat didalam data.

**DAFTAR PUSTAKA**

- [1] C. Irawan, M. R. Pratama, and V. Wahanggara, "Perbandingan algoritma Boyer Moore dan Brute Force pada pencarian Kamus Besar Bahasa Indonesia berbasis Android," *Universitas Muhammadiyah Jember*, 2021.
- [2] A. N. Alifah, "PENGARUH INFLASI, TINGKAT SUKU BUNGA, NILAI TUKAR, PDB, BAGI HASIL TERHADAP DEPOSITO MUDHARABAH BANK SYARIAH BUKOPIN." 2021.
- [3] R. Hastuti, I. Irawan, and A. Hukom, "Pengaruh Inflasi, Nilai Tukar, Suku Bunga dan Produk Domestik Bruto terhadap Return Saham pada Perusahaan Manufaktur (The Effect of Inflation, Exchange Rate, Interest Rate and Gross Domestic Products on Stock Returns in Manufacturing Companies)." 2023.
- [4] D. Islamiyati and I. H. Hany, "Pengaruh Inflasi, Indeks Produksi Industri dan Kurs terhadap Penghimpunan Zakat, Infaq dan Sedekah," *Telaah Bisnis*, vol. 20, no. 2, p. 25, Mar. 2021, doi: 10.35917/tb.v20i2.167.
- [5] Adamo et al. "A Surprisal-Based Greedy Heuristic for the Set Covering Problem" *Algorithms* (2023) doi:10.3390/a16070321.
- [6] Wang et al. "Carousel Greedy Algorithms for Feature Selection in Linear Regression" *Algorithms* (2023) doi:10.3390/a16090447.
- [7] T. Apriliani, A. S. D. Putri, J. Feranita, dan M. P. Mentari, "Implementasi Algoritma Greedy Dalam Penukaran Uang Di Alfamart Di Kota Dan Algoritma String Matching Untuk Pencarian Cabang Alfamart Di Kota-Kota Di Indonesia," *SABER: Jurnal Teknik Informatika, Sains dan Ilmu Komunikasi*, vol. 2, no. 2, pp. 283-294, Apr. 2024.
- [8] M. A. Afandy, S. Praja, dan T. Andriansah, "Implementasi Algoritma Greedy dan String Matching Penukaran Uang Menjadi Koin, Pencarian Pola dalam Teks yang Berisi Nama Indomaret," *Jurnal Multimedia dan Teknologi Informasi (Jatilima)*, vol. 6, no. 02, pp. 123-133, 2024.
- [9] R. K. Pandey and S. Taruna, "Prevalent Exact String-Matching Algorithms in Natural Language Processing: A Review," *Journal of Physics Conference Series*, vol. 1854, no. 1. IOP Publishing, p. 12042, Apr. 01, 2021. doi: 10.1088/1742-6596/1854/1/012042.