

# Implementasi sederhana Algoritma Greedy pada Penukaran Uang Pecahan

Ferli Andriansyah 120140018 (Author)

Program Studi Teknik Informatika

Jurusan Teknologi Produksi dan Industri

Institut Teknologi Sumatera, Jalan Terusan Ryacudu Lampung Selatan

E-mail: [ferli.120140018@student.its.ac.id](mailto:ferli.120140018@student.its.ac.id)

**Abstrak**—Uang adalah alat tukar yang kita gunakan untuk kita tukarkan untuk kegiatan ekonomi dengan nilai uang yang berbeda-beda. Masyarakat sendiri demi untuk memudahkan transaksi biasanya melakukan pemecahan uang dengan menukarkan uang ke bank atau ke penyedia jasa terdekat agar memudahkan saat sedang melakukan proses transaksi. Jurnal ini membahas permasalahan mencari jumlah minimum uang pecahan yang dihasilkan dari uang yang akan ditukarkan. Untuk perhitungan manual maka akan sangat sulit mencari jumlah minimum uang pecahan tersebut, oleh karena itu akan dibuat suatu program yang dapat mencari solusi dari masalah tersebut, yaitu dengan menggunakan algoritma greedy. Dengan menggunakan algoritma greedy tersebut kita dapat menyelesaikan proses pemecahan uang dengan cepat dan juga dapat lebih baik dalam memanfaatkan waktu dan tenaga.

**Kata kunci**—uang; greedy; optimal ;tukar

## I. PENDAHULUAN

Uang menjadi hal yang sangat penting dalam melakukan suatu transaksi, uang menjadi salah satu hal yang kita cari untuk memenuhi kebutuhan hidup. Uang dapat kita gunakan untuk membeli barang kebutuhan. Uang sendiri di Indonesia memiliki nominal nilai yang berbeda seperti pada nominal uang kertas, yaitu mulai dari yang terbesar Rp100.000, Rp75.000, Rp50.000, Rp20.000, Rp10.000, Rp5.000, Rp2.000, Rp1.000 dan beberapa nominal lainnya. Dalam kehidupan sehari-hari kita selalu ingin melakukan segala sesuatu dengan cepat dan efisien seperti pada saat kita ingin memecahkan uang agar mudah saat kita gunakan dalam bertransaksi dengan cara memecahkan uang ke tempat pemecahan uang yang tersedia seperti Bank ataupun penyedia jasa pemecahan uang yang ada.



Gambar 1.1 Nominal uang

Sumber: <https://www.merdeka.com/jatim/8-jenis-uang-berdasarkan-nilai-dan-kegunaannya-ketahui-agar-tak-keliru-klm.html>

Pemecahan uang secara manual menjadi salah satu masalah yang terjadi karena dari pemecahan uang secara manual memakan waktu dan tenaga yang ekstra dibandingkan dengan menggunakan suatu program untuk melakukan pemecahan uang tersebut. Pemecahan nominal kecil memang akan cepat saat dipecahkan tetapi bagaimana jika nominal uang yang dipecahkan berjumlah besar maka akan sangat memakan waktu dan tenaga juga tentunya. Hal tersebut dapat kita atasi dengan mengimplementasikan konsep dari algoritma greedy yang mana ini dapat membantu agar dapat lebih cepat dan efisien dalam melakukan pemecahan uang.

## II. TEORI DASAR

### A. Algoritma

Algoritma adalah suatu susunan atau langkah-langkah secara terurut yang sesuai logis dan sistematis dalam memecahkan suatu permasalahan untuk mendapatkan suatu hasil yang diinginkan. Algoritma sendiri tidak hanya berkaitan dengan dunia informatika tetapi juga hal yang kita lakukan sehari-hari saat menghadapi suatu permasalahan. Algoritma juga dapat diartikan suatu susunan yang logis secara urut dengan berdasarkan sistematika khusus yang digunakan dalam memecahkan suatu persoalan atau permasalahan tertentu. Secara umum, algoritma berguna agar bisa melakukan perhitungan, melakukan penalaran secara otomatis dengan membuat algoritma untuk menyelesaikan permasalahan tersebut, serta dapat juga untuk mengolah data didalam suatu komputer dengan menggunakan bantuan dari aplikasi / software tertentu.[1]

Algoritma sendiri memiliki tiga dasar utama yaitu :

- Algoritma sekuensial  
Algoritma sekuensial merupakan suatu algoritma yang langkah-langkahnya dilakukan secara urut dari awal hingga akhir dari atas hingga bawah.
- Algoritma perulangan atau looping algoritma  
Algoritma perulangan adalah suatu perintah untuk melakukan suatu aksi secara terus-menerus atau berulang-ulang.
- Algoritma percabangan(bersyarat)  
yaitu algoritma yang melakukan aksi jika syaratnya sudah terpenuhi.[2]

Algoritma pada dasarnya memiliki fungsi untuk memecahkan suatu permasalahan dengan berbagai macam manfaat atau kegunaannya seperti:

- Dapat membantu dalam menyederhanakan suatu program yang rumit.
- Dapat digunakan untuk menyelesaikan permasalahan yang memiliki pola sama secara berulang-ulang.
- Dapat membantu menyelesaikan persoalan secara sistematis dan juga secara logis.
- Dapat membuat program menjadi rapih dan lebih terstruktur.
- Mempermudah kesalahan yang terjadi karena langkah-langkahnya jelas.
- Membantu meminimalisir penulisan program secara berulang

### B. Algoritma greedy

Algoritma *greedy* adalah algoritma yang bekerja secara langkah demi langkah atau step-by step tanpa memperhatikan langkah-langkah yang kedepannya. Algoritma *greedy* memilih solusi optimum lokal, yang tidak selalu mengarah pada solusi global yang optimal, karena tidak mempertimbangkan seluruh data yang ada layaknya *exhaustive search* dari algoritma *Brute Force*. Algoritma *greedy* ini merupakan metode yang paling sering digunakan untuk masalah optimasi (optimization problems) untuk persoalan mencari solusi optimal.

Hanya terdapat dua macam persoalan optimasi untuk algoritma *greedy* yaitu :

1. Maksimasi atau maximization
2. Minimasi atau minimization

Solusi yang diberikan oleh pendekatan algoritma *greedy* tidak mempertimbangkan data dan pilihan untuk kemungkinan ke depan. Dalam beberapa kasus, algoritma *greedy* membuat keputusan yang pada saat itu memberikan solusi terbaik, tetapi dalam beberapa kasus lainnya mungkin tidak berlaku demikian.[3]

Pendekatan dengan algoritma *greedy* memiliki beberapa pengorbanan untuk menemukan solusi optimal pada setiap permasalahan. Dalam permasalahan *activity selection*, jika lebih banyak aktivitas dapat dilakukan sebelum menyelesaikan aktivitas yang dikerjakan saat ini, aktivitas ini dapat dilakukan dalam waktu yang sama. Alasan lainnya adalah untuk membagi masalah secara rekursif berdasarkan suatu kondisi, tanpa perlu menggabungkan semua solusi. Dalam permasalahan *activity selection*, langkah *recursive division* dicapai dengan memindai daftar item hanya sekali kemudian mempertimbangkan aktivitas tertentu.

Solusi optimal global dapat diperoleh dengan membuat solusi optimal lokal dari algoritma *greedy*. Solusi yang diberikan oleh algoritma *greedy* mungkin bergantung pada pilihan sebelumnya, tetapi tidak bergantung pada langkah ke depannya. Hal tersebut secara berulang membuat satu pilihan *greedy* kemudian mengurangi permasalahan yang diberikan sehingga terbagi menjadi permasalahan yang lebih kecil.

Keuntungan dari penggunaan algoritma *greedy* yaitu :

- Algoritma *greedy* mudah untuk diterapkan
- Memiliki kompleksitas waktu yang terbilang sedikit
- Dan algoritma *greedy* dapat digunakan untuk tujuan optimasi.

Oleh karena itu, Algoritma Greedy merupakan algoritma yang lazim untuk memecahkan persoalan optimasi meskipun hasilnya tidak selalu merupakan solusi yang optimum. Prinsip utama dari algoritma ini adalah mengambil sebanyak mungkin apa yang dapat diperoleh sekarang

### C. Optimasi

Optimasi secara umum adalah untuk memaksimalkan hasil dari suatu hal yang dikerjakan yang sedang dikelola. Oleh karena itu optimasi dapat dikatakan sebagai ilmu pengetahuan dan seni untuk tujuan yang ingin di maksimalkan kerja atau hasilnya. Secara matematis optimasi adalah bagaimana kita untuk mendapatkan harga kstrim baik dari segi maksimum atau minimum dari suatu fungsi-fungsi tertentu dengan faktor yang menjadi pembatasnya.



Gambar 2.1 Optimization

Sumber : <https://www.logique.co.id/blog/2018/03/06/optimasi-gambar-pada-website/>

Optimasi untuk menyelesaikan masalah merupakan suatu cara untuk kita mudah dalam mengambil keputusan sehingga didapatkan hasil penyelesaian dengan hasil yang optimal sesuai dengan kendala. Optimasi juga berarti sebagai proses untuk mendapatkan keadaan yang dapat menghasilkan nilai maksimum atau minimum dari suatu fungsi program.

### D. Metode Optimasi

Metode dari pencarian optimum biasa dikenal sebagai teknik mathematical programming dan biasa dipelajari sebagai bagian dari hasil riset operasi. Yang mana riset operasi ini sendiri memiliki keterkaitan dengan penerapan metode ilmiah dan teknik pengambilan dari keputusan penyelesaian terbaik atau paling optimal.

Perkembangan dari metode optimasi ini semakin meningkat dan terus mengalami kemajuan sampai sekarang ini. Hal ini dapat kita lihat dari semakin banyaknya metode optimasi yang ditemukan dan terus dapat menemukan solusi-solusi yang lebih optimal. Contoh dari metode optimasi yang populer dan sering dipakai contohnya seperti Integer Programming, Game Theory, Dynamic programming dan metode optimasi modern

Metode non-tradisional atau yang biasa disebut metode optimasi modern ini hadir sebagai metode yang baik dalam menyelesaikan masalah optimasi dan juga populer untuk menyelesaikan masalah optimasi yang kompleks. Metode yang termasuk seperti optimasi koloni semut, optimasi fuzzy, algoritma genetik, optimasi partikel swarm dan masih banyak yang lainnya.

#### E. Python

Python adalah bahasa pemrograman yang interpretatif dan multifungsi. Pada Python lebih menekankan pada desain bagus dan sederhana agar user lebih mudah membaca kode dan memahami sintaks-sintaks yang tersedia. Hal ini yang membuat bahasa pemrograman Python sendiri memiliki banyak peminat dan juga cocok untuk pemula ataupun juga untuk seorang yang sudah menguasai bahasa pemrograman lain.

Bahasa pemrograman ini pertama kali muncul pada tahun 1991, yang dirancang oleh Guido Van Rossum. Sampai saat ini bahasa pemrograman Python masih terus dikembangkan. Bahasa Python sendiri hampir mendukung semua sistem operasi yang tersedia untuk sekarang ini, seperti pada sistem operasi Linux yang hampir semua distronya menambahkan atau menyertakan Python didalamnya.



Gambar 2.2 Logo Python

Sumber : <https://www.petanikode.com/python-linux/>

Dengan kode-kode yang simpel dan mudah dipahami serta diimplementasikan, user atau pengguna dapat lebih fokus dan juga mengutamakan pengembangan dari aplikasi atau program yang sedang dibuat. Bukan hanya sibuk untuk mencari sintaks yang error. Karena memang bahasa pemrograman Python ini sangat sederhana dibandingkan bahasa pemrograman yang lainnya. Contohnya seperti untuk menampilkan "Hello world" kelayar tidak perlu ini dan itu cukup dengan print("Hello world") maka sudah ditampilkan di layar. [4]

Keunggulan dari Python yaitu :

1. Cepat dan efektif untuk digunakan
2. Mudah dipelajari khususnya untuk pemula yang baru ingin terjun ke dunia coding
3. Banyak digunakan di perusahaan besar
4. Sintaks-sintaks yang terdapat didalamnya mudah dipahami
5. Sangat dinamis dan dapat mudah untuk membaca kode karena tingkat keterbacaannya yang tinggi.
6. Dan juga bersifat open source sehingga kita dapat mengunduhnya secara gratis dan tidak perlu untuk membeli lisensi.

Tetapi bahasa pemrograman ini sendiri ini juga memiliki kekurangan yaitu :

1. Kurang populer untuk membuat sebuah mobile app
2. Memiliki banyak batasan desain dikarenakan Python diketik secara dinamis
3. Membutuhkan compiler yang membuat Python akan bekerja lebih lambat dibandingkan dengan bahasa pemrograman lainnya jika program yang dikerjakan berukuran besar.
4. Sedikit developer yang berpengalaman menggunakan bahasa Python.
5. Masih kurangnya dukungan dari multiprosesor sehingga dapat membatasi penulisan dari kode
6. Menghasilkan konsumsi memori yang cukup tinggi

### III. ANALISIS DAN PEMBAHASAN

#### A. fungsi

Fungsi pada python adalah berisi kumpulan perintah atau baris kode yang dikelompokkan menjadi satu bentuk kesatuan yang nantinya dapat dipanggil atau kita gunakan berulang kali sebuah fungsi dapat menerima parameter, dapat mengembalikan suatu nilai dan bisa dipanggil secara independen.

Dalam fungsi terdapat sintaks umum yang terbagi menjadi 4 bagian :

1. Kata kunci `def` menjadi pertanda bahwa kode program adalah fungsi
2. Nama fungsi yang kita buat
3. Parameter yang akan diterima oleh fungsi yang kita buat.
4. Dan blok kode fungsi yang di sana akan kita tulis perintah-perintah yang harus dilakukan oleh sebuah fungsi
5. Memiliki `()` diakhir nama variable

#### B. `split()`

`split()` dalam Python adalah alat yang sangat berguna untuk membagi atau memisahkan string menjadi potongan-potongan tergantung pada pembatas yang bisa berupa apa saja mulai dari karakter atau angka atau bahkan teks

#### C. `len()`

Fungsi dari `len()` pada python berguna untuk mengetahui ukuran atau panjang dari nilai dari variabel tersebut seperti objek string, list, tuple dan range. [5]

### IV. Penjelasan source code

Penjelasan source code :

- Program akan meminta input berupa jumlah nominal uang yang akan dipecahkan
- Dan kemudian terdapat fungsi `cari_nilaimin(nilai_uang)` yang akan dipanggil
- Dan didalam fungsi tersebut diminta untuk menginputkan nominal uang yang ingin dipecahkan
- Dan terdapat perulangan
- Kemudian terdapat `n = len(uang_dipecah)` untuk menghitung jumlah dari nilai\_uang yang diinputkan sebelumnya
- Dan terdapat sebuah list hasil untuk menampung nilai pecahan
- `i = n - 1` untuk menginisialisasi untuk `n` dikurangkan dengan 1
- Yang akan digunakan pada perulangan `while` dengan syarat `i >= 0`
- Dan terdapat lagi syarat bahwa jika nilai\_uang lebih besar atau sama dengan uang\_dipecah indeks ke `i` maka akan mengurangi dengan nilai uang\_dipecah ke indeks `i`
- Dan akan ditambahkan ke list hasil selumnya
- Kemudian terdapat perulangan untuk menampilkan hasil kelayar
- Selanjutnya akan menampilkan kelayar dan juga melakukan perulangan untuk menampilkan banyaknya uang yang dipecahkan sejumlah berapa lembar.

### V. IMPLEMENTASI

Dalam jurnal ini membahas mengenai penukaran uang pecahan dengan menggunakan konsep algoritma greedy dimana pada pengujian ini menggunakan python untuk menyelesaikan penukaran uang kepecahan. pada pengujian kali ini bahasa pemrograman yang digunakan adalah python.

Untuk mempermudah saat pemecahan uang maka dibuatlah suatu program untuk membantu memecahkan masalah tersebut agar dapat diselesaikan lebih cepat dengan mengimplementasikan algoritma greedy

Dimana mekanisme pada saat dijalankan program akan meminta untuk memasukkan nominal dari uang yang ingin dipecahkan kemudian dilanjutkan dengan menginputkan nominal uang pecahan yang diinginkan.

Dan kemudian akan menampilkan hasil dari sesudah dipecahkan dari nominal yang diinputkan sebelumnya dan menampilkan hasil berupa berapa banyak atau lembar uang yang dipecahkan

berikut adalah kode program :

```
def cari_nialiMin(nilai_uang):
    Uang_dipecah = input("Masukan Nominal
uang (Rp): ").split()
    Uang_dipecah = [int(i) for i in
Uang_dipecah]
    n = len(Uang_dipecah)
    hasil = []
    i = n - 1
    while(i >= 0):
        while (nilai_uang >=
Uang_dipecah[i]):
            nilai_uang -= Uang_dipecah[i]
            hasil.append(Uang_dipecah[i])
            i -= 1

    for i in range(len(hasil)):
        print(hasil[i], end=" ")

    print("\ndipecahkan dengan",end=" ")
    for i in Uang_dipecah :
        print(i, end="")
    print(" sebanyak :",len(hasil) ,"lembar")

nilai_uang = int(input("Masukan nominal yang
ingin di pecahkan : "))
cari_nialiMin(nilai_uang)
```

Berdasarkan dari program yang dibuat untuk memecahkan nominal uang maka akan dilakukan uji dari progrm tersebut pada saat dijalankan program akan meminta untuk memasukan nominal dari uang yang ingin dipecahkan kemudian dilanjut dengan menginputkan nominal uang pecahan yang diinginkan dan kemudian akan menampilkan hasil jumlah uang yang dipecahkan :

Berikut ini contoh ouput program uji tes pertama:

```
Masukan Nominal uang (Rp): 100000
Masukan nominal yang ingin di pecahkan (Rp) :20000
20000 20000 20000 20000 20000
dipecahkan dengan 20000 sebanyak : 5 lembar
```

dalam percobaan diatas , masukan nominal uang yang merupakan jumlah total uang yang ingin dipecahkan dan kemudian masukan nominal yang ingin dipecahkan dan akan menampilkan hasil dari jumlah uang pecahan yang didapat dari nominal yang dimasukan.

Uji tes kedua:

```
Masukan Nominal uang (Rp): 100000
Masukan nominal yang ingin di pecahkan (Rp) :2000
2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 20
00 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000
2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2
000 2000 2000 2000
dipecahkan dengan 2000 sebanyak : 50 lembar
```

Dari hasil percobaan kedua dengan nomnial uang yang lebih besar dan menghasilkan output tersebut.

## VI. UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kehadiran Tuhan Yang MahaEsa atas limpahan rahmat dan karunia oleh-Nya, sehingga makalah ini dapat diselesaikan dengan baik.Pada bagian akhir dari makalah ini, penulis inginmenyampaikan ucapan terima kasih setulus-tulusnya kepadadosen pengampu mata kuliah Strategi Algoritma yaitu ImamEkowicaksono, M. Si., dan Winda Yulita, M.Cs. yang telah memberi ilmu yang mendasari pembuatan makalah ini serta dukungan ketika proses pembuatan karya tulis ini.

Selain itu, penulis ingin menyampaikan rasa terima kasihkepada kedua orang tua penulis atas dukungan, doa, dan nasehat yang tak henti-hentinya diberikan kepada penulis sehingga makalah ini dapat diselesaikan dengan baik menggunakan kobaran semangat dan kekuatan yang disalurkan oleh pemberian dari kedua orang tua penulis.Penulis sepenuhnya sadar bahwa hasil karya tulis berupa makalah ini masih jauh dari standar kesempurnaan. Maka dari itu, penulis ingin menyampaikan permohonan maaf atas kesalahan yang ada pada hasil maupun proses dari penulisan makalah ini. Penulis sangat menghargai pemberian kritik dan saran yang membangun untuk penyempurnaan proses

penulisan karya ilmiah serupa yang dihadapi penulis di kemudian hari. Besar harapan penulis, semoga makalah

yang

telah ditulis ini dapat mendatangkan manfaat serta nilai yang positif bagi pihak pembaca yang membutuhkan.



## REFERENCES

- [1] Renaldi munir, “algoritma greedy” [Algoritma Greedy \(itb.ac.id\)](#) Diakses pada 28 Maret 2022.
- [2] Belajar bahasa pemrograman python “pendahuluan python <https://belajarpython.com/tutorial/apa-itu-python> Diakses pada 28 Maret 2022.
- [3] Programiz, “Learn C++ Programming”, <https://www.programiz.com/cpp-programming>. Diakses pada 28 Maret 2022.
- [4] Guru99, “C++ Programming: What is C++ | Learn Basic Concepts of C++”, <https://www.guru99.com/cpp-tutorial.html>. Diakses pada 28 Maret 2022.
- [5] phytont, “fungsi len python <https://kopiding.in/fungsi-len-pyth>. on/ Diakses 28 Maret 2022.
- [6] CPlusPlus, “Arrays”, <https://www.cplusplus.com/doc/tutorial/arrays/>. Diakses pada 28 Maret 2022.
- [7] GeeksforGeeks, “Vector in C++ STL”, <https://www.geeksforgeeks.org/vector-in-cpp-stl/>. Diakses pada 28 Maret 2022.
- [8]

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandar Lampung, 28 Maret 2022



Ferli Andriansyah 120140018