

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

Pada bab ini secara garis besar akan dijelaskan pengertian-pengertian dan konsep-konsep dasar yang akan digunakan dalam pengembangan sistem yang dibuat dalam tugas akhir ini.

#### **2.1. Tinjauan Pustaka**

Beberapa referensi yang digunakan untuk pembuatan Tugas Akhir ini akan disajikan dalam bentuk tabel perbandingan sebagai berikut :

1. Penelitian oleh Rina Pebriana dan Eni Suasri (2022) membahas penerapan rekonsiliasi dalam konteks keuangan daerah. Penelitian ini memberikan gambaran tentang pentingnya rekonsiliasi untuk mencegah kesalahan dalam pencatatan transaksi. Hal ini mendukung relevansi penelitian ini dalam menerapkan algoritma untuk mencocokkan transaksi debit dan kredit.
2. Penelitian Muhammad Arifin Fikri (2018) memberikan pandangan mengenai perbandingan performa algoritma Boyer-Moore dan Brute Force dalam menyelesaikan game puzzle Sudoku. Meskipun penelitian ini berada pada domain yang berbeda, hasilnya menunjukkan bahwa algoritma Brute Force memiliki waktu eksekusi yang lebih cepat, namun memerlukan jumlah iterasi yang lebih banyak dibandingkan Boyer-Moore. Hal ini menjadi referensi penting dalam memahami

efisiensi algoritma Brute Force, yang juga digunakan dalam penelitian ini.

3. Penelitian oleh Rizky Berlia Oktaviandi, dkk (2018) memperbandingkan algoritma genetika dan algoritma greedy dalam konteks pencarian rute terpendek. Penelitian ini menyoroti efisiensi algoritma greedy dalam menyelesaikan masalah optimasi dengan waktu dan biaya minimum. Oleh karena itu, penelitian ini relevan sebagai referensi untuk memahami bagaimana algoritma greedy dapat memberikan solusi optimal secara cepat, yang selaras dengan fokus penelitian ini dalam memanfaatkan algoritma greedy untuk data besar.
4. Penelitian Irwan Adi Pribadi, dkk (2021) menggunakan algoritma Brute Force untuk menghitung recall, precision, dan accuracy dalam konteks pencarian berbasis web. Penelitian ini memberikan wawasan tentang cara pengujian performa algoritma Brute Force, yang dapat diadaptasi dalam pengujian akurasi algoritma dalam penelitian ini.
5. Penelitian oleh Ibrahim Alkhwaja, dkk (2023) menyoroti efektivitas algoritma Brute Force dalam keamanan siber dengan optimasi pemrograman paralel. Studi ini menunjukkan bahwa Brute Force tetap relevan untuk menyelesaikan masalah meskipun memiliki waktu eksekusi yang lama, sehingga relevan untuk dijadikan pembanding dalam penelitian ini.
6. Penelitian oleh Sangeeta Simanjuntak (2025), berfokus pada komparasi algoritma Brute Force dan Greedy Pairing dalam konteks

rekonsiliasi transaksi debit dan kredit pada laporan keuangan. Penelitian ini bertujuan untuk menguji efektivitas dan efisiensi kedua algoritma tersebut dalam mencocokkan transaksi keuangan, serta memberikan solusi atas kendala yang sering dihadapi dalam penyelesaian akun suspense, masalah praktis yang sering diabaikan namun sangat penting dalam laporan keuangan.

Tabel 2. 1. Tabel Kajian Pustaka

<b>Peneliti</b>	<b>Algoritma yang Digunakan</b>	<b>Keterangan</b>
Pebriana, R., & Suasri, E. (2022)	Rekonsiliasi Kas Umum (Manual)	Rekonsiliasi untuk meminimalkan kesalahan pencatatan transaksi keuangan daerah.
Fikri, M.A.(2018)	Boyer-Moore, Brute Force	Brute Force lebih cepat, tetapi iterasi lebih banyak dibandingkan Boyer-Moore.
Oktaviandi, R. B., et al. (2018)	Genetika, Greedy Pairing	Greedy lebih efisien dalam waktu dan biaya untuk pencarian rute terpendek.
Pribadi, I. A., et al. (2021)	Brute Force	Brute Force digunakan untuk mengukur recall, precision, dan accuracy pada pencarian berbasis web.
Alkhwaja, I., et al. (2023)	Brute Force, Dictionary Attack	Brute Force lebih lambat tetapi efektif, dipercepat dengan pemrograman paralel.

## 2.2. Dasar Teori

### 2.2.1. Rekonsiliasi Transaksi

Rekonsiliasi transaksi merupakan proses penting dalam memastikan keakuratan data keuangan. Pengertian rekonsiliasi yang diatur dalam Peraturan Menteri Keuangan Nomor 59/PMK.06/2005 Pasal 1 Ayat 19 yaitu: “Rekonsiliasi

adalah proses penyocokan data transaksi keuangan yang diproses dengan beberapa sistem / subsistem yang berbeda berdasarkan dokumen sumber yang sama.”

Menurut Wilkinson dan Cerullo (2017), Rekonsiliasi adalah langkah krusial dalam pengelolaan laporan keuangan untuk memastikan keakuratan catatan dan mencegah terjadinya kesalahan dalam pencatatan. Proses rekonsiliasi melibatkan pencocokan antara transaksi debit dan kredit untuk memastikan keseimbangan laporan. Proses ini penting untuk menjaga integritas data keuangan dan mengidentifikasi kesalahan pencatatan atau transaksi yang belum terselesaikan (*suspense*).

### ***2.2.2. Suspense Account***

*Suspense account*, atau akun sementara, sering kali muncul ketika terdapat transaksi yang belum dipasangkan atau terjadi ketidaksesuaian antara dua sisi laporan keuangan (Jones & Jones, 2020).

Contoh kasus :

Bank Cabang B menerima pembayaran pinjaman dari nasabah X yang mempunyai pinjaman di Cabang A yang merupakan tempat lahir dan tempat pembuatan rekening dari nasabah X. Pembayaran yang diterima Bank Cabang B akan dimasukkan ke akun *Suspense* karena nomor rekening pinjaman dari nasabah X tidak terdaftar di portal pembukuan Bank Cabang B. Sehingga pembukuan yang dilakukan adalah :

**D.** Kas Kantor

**K.** *Suspense Account*

Pada cabang A, di tanggal bersamaan menerima uang masuk dari cabang B namun norek titipan Cabang B tidak terdaftar di Cabang A, maka di Cabang B pembukuan yang dilakukan adalah :

**D. Suspense**

**K. Pembayaran Pinjaman nasabah X**

### 2.2.3. Algoritma Greedy Pairing

Algoritma Greedy Pairing merupakan salah satu metode yang digunakan dalam pemecahan masalah pencocokan transaksi dengan mengutamakan pemilihan solusi terbaik pada setiap langkah, tanpa mempertimbangkan konsekuensi dari pilihan tersebut pada tahap berikutnya (Vishwanathan, 2020).

Prosedur Algoritma Greedy Pairing :

```

GREEDY-ACTIVITY-SELECTOR( $s, f, n$ )
1   $A = \{a_1\}$ 
2   $k = 1$ 
3  for  $m = 2$  to  $n$ 
4      if  $s[m] \geq f[k]$            // is  $a_m$  in  $S_k$ ?
5           $A = A \cup \{a_m\}$      // yes, so choose it
6           $k = m$                  // and continue from there
7  return  $A$ 

```

Gambar 2. 1. Struktur Algoritma Greedy Pairing (Sumber: Cormen, T.H. et al, 2022. **Introduction to Algorithms** (4th ed.). MIT Press).

Line 1 : Inisialisasi  $A = \{a_1\}$

Aktivitas pertama ( $a_1$ ) dipilih terlebih dahulu karena secara otomatis dianggap sebagai bagian dari solusi optimal.

Line 2 : Inisialisasi Indeks  $k = 1$

Variabel  $k$  melacak aktivitas terakhir yang dipilih dari solusi.

Line 3 : for  $m = 2$  to  $n$

Algoritma memeriksa setiap aktivitas dari aktivitas kedua ( $a_2$ ) hingga aktivitas terakhir ( $a_n$ ).

Line 4 : if  $s[m] \geq f[k]$

Aktivitas  $m$  hanya akan dipilih jika waktu mulai (start time) dari aktivitas  $m$  lebih besar atau sama dengan waktu selesai (finish time) dari aktivitas yang dipilih terakhir ( $k$ ). Ini memastikan bahwa aktivitas tidak tumpang tindih.

Line 5 :  $A = A \cup \{a_m\}$

Jika aktivitas  $m$  kompatibel, tambahkan ke dalam himpunan solusi  $A$ .

Line 6 :  $k = m$

Indeks  $k$  diperbarui ke aktivitas  $m$  yang baru saja dipilih, sehingga aktivitas berikutnya akan diperiksa terhadap aktivitas ini.

Line 7 : return  $A$

Himpunan  $A$  berisi semua aktivitas yang telah dipilih sesuai dengan aturan di atas.

#### 2.2.4. Algoritma Brute Force

Algoritma Brute Force adalah pendekatan komputasi yang mencoba semua kemungkinan solusi untuk menemukan hasil yang benar.

---

```

For each subset  $S$  of  $k$  nodes
  Check whether  $S$  constitutes an independent set
  If  $S$  is an independent set then
    Stop and declare success
  Endif
Endfor
If no  $k$ -node independent set was found then
  Declare failure
Endif

```

---

Gambar 2. 2. Pseudocode Algoritma Brute Force (Sumber: Kleinberg & Tardos, 2006. Algorithm Design. Pearson Education).

Logikanya adalah mencari independent set di sebuah graf. Sebuah independent set adalah himpunan simpul dalam graf di mana tidak ada dua simpul yang terhubung dengan sebuah sisi (edge). Tujuannya adalah menentukan apakah ada independent set dengan ukuran tetap  $k$  dalam graf.

Pendekatan brute force dilakukan dengan :

1. Memeriksa semua subset simpul yang berukuran  $k$ .
2. Mengecek apakah subset tersebut merupakan independent set.

Jika ditemukan subset  $k$ -simpul yang tidak saling terhubung, algoritma mengembalikan *success*. Jika tidak ada subset yang ditemukan, kembalikan *failure*.

Kompleksitas waktu utama berasal dari:

Jumlah semua subset  $k$ -elemen dari  $n$ -elemen (yaitu, simpul-simpul dalam graf

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Untuk setiap subset, perlu memeriksa apakah subset tersebut adalah independent set, yang melibatkan cek untuk semua pasangan simpul dalam subset tersebut ( $O(k^2)$ ).

Total waktu yang diperlukan adalah :

$$O\left(\binom{n}{k} \cdot k^2\right) = O\left(\frac{n^k}{k!} \cdot k^2\right)$$

Kompleksitas ini bersifat eksponensial karena faktor  $\binom{n}{k}$  tumbuh sangat cepat seiring dengan meningkatnya  $n$  dan  $k$ .

"Pendekatan brute force ini memiliki kompleksitas waktu eksponensial sebesar  $\mathcal{O}\left(\binom{n}{k} \cdot k^2\right)$  sehingga hanya cocok untuk graf kecil atau k kecil (Kleinberg & Tardos, 2006)."

### **2.2.5. Python**

Python adalah salah satu bahasa pemrograman tingkat tinggi yang populer di dunia akademik dan industri. Bahasa ini dirancang untuk kemudahan penggunaan dan memiliki sintaks yang sederhana sehingga cocok digunakan untuk berbagai aplikasi, termasuk pengolahan data, pengembangan algoritma, dan analisis statistik (Van Rossum, 2007). Python juga didukung oleh pustaka yang diperlukan dalam penelitian ini, seperti:

NumPy : Digunakan untuk perhitungan numerik yang efisien.

Pandas : Memudahkan manipulasi dan analisis data dalam bentuk tabel.

Itertools : Mendukung implementasi algoritma melalui fungsi kombinasi dan permutasi.

Dengan fleksibilitasnya, Python memungkinkan implementasi algoritma Brute Force dan Greedy Pairing secara efisien, memanfaatkan pustaka-pustaka tersebut untuk mempermudah proses rekonsiliasi transaksi.

### **2.2.6. Efektivitas Algoritma**

Efektivitas adalah ukuran sejauh mana sebuah sistem, proses, atau algoritma mampu mencapai tujuan yang telah ditentukan. Dalam konteks algoritma pencocokan transaksi, efektivitas diukur berdasarkan kemampuan



algoritma dalam menemukan semua pasangan transaksi yang sesuai. Robbins dan Coulter (2012) mendefinisikan efektivitas sebagai tingkat pencapaian hasil yang diinginkan dibandingkan dengan tujuan yang telah ditetapkan.

Dalam penelitian ini, efektivitas diukur menggunakan formula berikut:

$$Efektivitas (\%) = \frac{Jumlah\ pasangan\ ditemukan}{Jumlah\ pasangan\ diharapkan} \times 100$$

Formula ini mengacu pada prinsip bahwa efektivitas ditentukan oleh perbandingan antara output aktual dengan output ideal (Knuth, 1973).

### 2.2.7. Efisiensi Waktu Algoritma

Efisiensi mengukur seberapa optimal sumber daya yang digunakan untuk mencapai tujuan tertentu. Dalam konteks algoritma, sumber daya yang dimaksud meliputi waktu eksekusi dan jumlah pasangan yang berhasil ditemukan. Taylor (1911) mendefinisikan efisiensi sebagai perbandingan antara input yang digunakan dan output yang dihasilkan.

Efisiensi waktu dalam penelitian ini dirumuskan sebagai:

$$Efisiensi = \frac{Waktu\ Eksekusi}{Jumlah\ Pasangan\ yang\ Ditemukan}$$

Formula ini didasarkan pada teori kompleksitas algoritma yang dijelaskan oleh Cormen et al. (2009), di mana efisiensi algoritma dapat dianalisis berdasarkan waktu eksekusi untuk menyelesaikan masalah tertentu.