

Implementasi Pengelolaan Data Mahasiswa Menggunakan Pointer dan Sorting Berbasis Bahasa C++

Yehezkiel Revan Santoso¹, Aurelio Fico Gavriel²

^{1,2}Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Kota Kediri, Indonesia

Artikel Info

Kata kunci:

C++
pointer
sorting
pencarian
data mahasiswa

ABSTRAK

Penelitian ini bertujuan untuk membangun aplikasi sederhana pengelolaan data mahasiswa dengan memanfaatkan struktur data pointer dalam bahasa pemrograman C++. Fitur yang dibangun meliputi input data, pencarian berdasarkan NIM, pengurutan berdasarkan nilai, dan penilaian predikat nilai. Dalam proses implementasi, algoritma bubble sort digunakan untuk pengurutan nilai dari yang tertinggi ke terendah. Fungsi rekursif juga diterapkan sebagai bagian dari latihan logika pemrograman melalui perhitungan faktorial. Program ini ditujukan sebagai media pembelajaran pengolahan data struktural menggunakan pendekatan prosedural. Hasil dari pengujian menunjukkan bahwa aplikasi berjalan stabil dan mampu melakukan pencarian serta pengurutan data sesuai dengan kebutuhan. Diharapkan hasil ini dapat dijadikan acuan dalam pengembangan sistem informasi skala kecil berbasis bahasa pemrograman C++.

Penulis Korespondensi :

Yehezkiel Revan Santoso
Teknik Informatika
Fakultas Ilmu Komputer
Universitas Dian Nuswantoro, Kediri, 61462
Email: santosoyehezkiel95@gmail.com

1. PENDAHULUAN

Pengelolaan data merupakan salah satu aspek fundamental dalam proses pengembangan perangkat lunak modern. Seiring dengan pesatnya pertumbuhan teknologi informasi di era digital saat ini, kebutuhan akan sistem yang mampu mengelola data secara efektif, efisien, dan terstruktur menjadi semakin mendesak [1]. Tidak terkecuali dalam sektor pendidikan tinggi, yang merupakan institusi vital dalam pembangunan sumber daya manusia [2].

Perguruan tinggi dituntut untuk memiliki sistem informasi yang handal, terutama dalam hal pencatatan, pemrosesan, dan pelaporan data mahasiswa, yang jumlah dan kompleksitasnya terus meningkat setiap tahunnya [3], [4].

Dalam konteks pengembangan aplikasi dengan skala data kecil hingga menengah, penggunaan bahasa pemrograman C++ menjadi salah satu solusi yang sangat relevan. Bahasa C++ dikenal sebagai bahasa pemrograman yang memiliki kendali penuh terhadap pengelolaan memori, terutama dengan fitur pointer yang memungkinkan manipulasi data secara langsung pada tingkat memori [5], [6]. Selain itu, C++ juga menyediakan kemampuan untuk membuat struktur data yang kompleks melalui *struct*, *class*, dan *array*, yang sangat berguna dalam penyimpanan dan pengelolaan informasi seperti data mahasiswa [4], [7], [8].

Beberapa penelitian terdahulu telah menunjukkan potensi besar bahasa C++ dalam membangun sistem pengelolaan data. Penelitian oleh Turnip et al. (2021), misalnya, menjelaskan bagaimana penggunaan pointer dan array dinamis dapat digunakan untuk menangani koleksi data dengan ukuran yang bervariasi, sehingga aplikasi yang dikembangkan menjadi lebih fleksibel dan efisien [9]. Sementara itu, studi oleh Sari et al. (2022) menekankan efektivitas algoritma *bubble sort* sebagai metode pengurutan sederhana namun cukup efisien untuk jumlah data yang tidak terlalu besar [10]. Kombinasi dari kedua pendekatan ini memungkinkan terciptanya sistem pengelolaan data yang tidak hanya optimal dalam kinerja, tetapi juga mudah untuk dipahami dan diterapkan oleh kalangan pemula.

Melalui penelitian ini, dikembangkan sebuah aplikasi pengelolaan data mahasiswa berbasis C++. Aplikasi ini dirancang dengan beberapa fitur utama, yaitu: input data mahasiswa (NIM, nama, nilai), pengurutan data berdasarkan nilai atau nama, pencarian data mahasiswa berdasarkan NIM menggunakan algoritma pencarian (*searching*), pemberian predikat nilai secara otomatis, serta penerapan fungsi rekursif untuk menampilkan data atau menjalankan fungsi tertentu [11]. Fitur-fitur tersebut disusun dalam suatu kerangka logika pemrograman struktural yang mengintegrasikan penggunaan pointer, array, fungsi, dan kontrol alur program yang sistematis [12], [13], [14].

Tujuan utama dari pengembangan aplikasi ini bukan hanya menghasilkan sebuah sistem yang fungsional, tetapi juga memberikan kontribusi dalam pembelajaran dan pemahaman konsep dasar pemrograman bagi mahasiswa atau pemula [15], [16], [17]. Dengan pendekatan yang terstruktur dan berorientasi pada praktik langsung, diharapkan penelitian ini dapat menjadi referensi yang berguna dalam mata kuliah pemrograman dasar, algoritma dan struktur data, serta pengembangan perangkat lunak berskala kecil.

Selain itu, aplikasi ini juga mengedepankan prinsip keterbacaan kode (*code readability*) dan modularitas, sehingga setiap bagian dari program dapat dipahami secara terpisah maupun sebagai bagian dari keseluruhan sistem. Dengan demikian, aplikasi ini tidak hanya relevan sebagai alat bantu pengelolaan data, tetapi juga sebagai media pembelajaran untuk mengenalkan konsep penting seperti pengelolaan memori manual, manipulasi data terstruktur, serta penerapan logika pemrograman yang efisien dan efektif.

2. METODE

Studi ini menerapkan metode eksperimental rekayasa perangkat lunak, dengan pendekatan berorientasi prosedural, melalui pengembangan aplikasi berbasis bahasa pemrograman C++. Tujuan utama dari metode ini adalah untuk menunjukkan secara langsung bagaimana konsep-konsep dasar dalam pemrograman struktural dapat diimplementasikan dalam konteks dunia nyata, yaitu pengelolaan data mahasiswa.

Proses ini meliputi beberapa tahapan sistematis, dimulai dari perancangan struktur data, proses penginputan data, pengurutan nilai dengan algoritma yang sesuai, pencarian data mahasiswa berdasarkan kriteria tertentu, hingga penerapan fungsi-fungsi tambahan seperti predikat nilai dan rekursi. Berikut ini penjelasan tahapan-tahapan tersebut secara lebih rinci:

2.1 Struktur Data Mahasiswa

Struktur data menjadi dasar utama dalam pengelolaan informasi. Dalam hal ini, digunakan struktur struct dalam C++ untuk merepresentasikan entitas mahasiswa. Adapun atribut yang disimpan mencakup nama, NIM, dan nilai:

```
struct Mahasiswa {  
    string nama;  
    string nim;  
    float nilai;  
};
```

Gambar 1. Struktur Struct

Struktur ini dipilih karena efisien dalam penyimpanan data yang bersifat homogen dan berukuran tetap. Dengan struktur ini, data tiap mahasiswa dapat diakses dan dimanipulasi dengan lebih mudah serta memungkinkan pengelolaan data yang lebih terorganisir.

2.2 Input Data

Pengisian data mahasiswa dilakukan melalui fungsi inputMahasiswa(), yang bertugas menerima input dari pengguna secara interaktif. Data yang dimasukkan meliputi nama, NIM, dan nilai. Proses input dilakukan dalam bentuk perulangan berdasarkan jumlah mahasiswa yang akan didata.

```

void inputMahasiswa(Mahasiswa* mhs, int jumlah) {
    for (int i = 0; i < jumlah; i++) {
        cout << "\nMahasiswa ke-" << i + 1 << ":\n";
        cout << "Nama   : ";
        cin.ignore();
        getline(cin, mhs[i].nama);
        cout << "NIM    : ";
        getline(cin, mhs[i].nim);
        cout << "Nilai  : ";
        cin >> mhs[i].nilai;
    }
}

```

Gambar 2. Input Data Mahasiswa

Penggunaan fungsi `getline()` dimaksudkan agar input string seperti nama dan NIM yang mengandung spasi tetap dapat ditangani dengan baik. Sementara itu, penggunaan pointer dalam parameter memungkinkan fungsi ini untuk memodifikasi langsung data yang disimpan dalam array mahasiswa.

2.3 Sorting Bubble

Untuk mengurutkan data mahasiswa berdasarkan nilai, digunakan algoritma *Bubble Sort*. Algoritma ini dipilih karena mudah diimplementasikan dan cukup efisien untuk data berukuran kecil hingga menengah.

```

void bubbleSort(Mahasiswa* mhs, int jumlah) {
    for (int i = 0; i < jumlah - 1; i++) {
        for (int j = 0; j < jumlah - i - 1; j++) {
            if (mhs[j].nilai < mhs[j + 1].nilai) {
                swap(mhs[j], mhs[j + 1]);
            }
        }
    }
}

```

Gambar 3. Penggunaan Algoritma Bubble Sort

Pengurutan dilakukan dari nilai tertinggi ke terendah, dengan asumsi bahwa urutan ini memudahkan dalam menampilkan mahasiswa terbaik di posisi awal. Fungsi `swap()` digunakan untuk menukar posisi elemen dalam array.

2.4 Pencarian Data

Untuk menemukan mahasiswa tertentu berdasarkan NIM, digunakan teknik pencarian linear atau *sequential search*. Teknik ini dipilih karena implementasinya sederhana dan efektif untuk data dalam jumlah kecil, terutama ketika data belum terurut berdasarkan NIM.

```

int cariMahasiswa(Mahasiswa* mhs, int jumlah, string targetNIM) {
    for (int i = 0; i < jumlah; i++) {
        if (mhs[i].nim == targetNIM) {
            return i;
        }
    }
    return -1;
}

```

Gambar 4. Pencarian Data Mahasiswa

Jika NIM yang dicari ditemukan, fungsi akan mengembalikan indeks dari array mahasiswa yang bersesuaian. Jika tidak ditemukan, maka fungsi akan mengembalikan nilai -1 sebagai penanda ketidaksamaan.

2.5 Fungsi Predikat

Salah satu fitur penting dalam aplikasi ini adalah pemberian predikat terhadap nilai akhir mahasiswa. Predikat digunakan sebagai indikator pencapaian akademik, dengan skala huruf yang umum digunakan seperti A, B, C, D, dan E. Kriteria ini dapat membantu dalam menginterpretasikan

nilai numerik secara lebih komunikatif.

Berikut adalah implementasi fungsi `hitungPredikat()`:

```
string predikat(float nilai) {
    if (nilai >= 85) return "A";
    else if (nilai >= 70) return "B";
    else if (nilai >= 55) return "C";
    else if (nilai >= 40) return "D";
    else return "E";
}
```

Gambar 5. Fungsi Hitung Predikat

Dengan pendekatan ini, setiap nilai numerik mahasiswa dikonversi ke huruf berdasarkan rentang nilai. Fungsi ini akan dipanggil saat menampilkan data atau saat dibutuhkan analisis kualitas performa mahasiswa.

2.6 Fungsi Rekursi

Fungsi rekursif dalam aplikasi ini digunakan untuk menampilkan daftar mahasiswa secara berurutan. Penggunaan rekursi dimaksudkan untuk memperkenalkan konsep lanjutan kepada pemula, di mana fungsi memanggil dirinya sendiri sampai kondisi tertentu terpenuhi.

Berikut contoh implementasi fungsi rekursif untuk menampilkan data mahasiswa

```
int faktorial(int n) {
    if (n <= 1) return 1;
    return n * faktorial(n - 1);
}
```

Gambar 6. Fungsi Rekursi

Rekursi dalam konteks ini digunakan sebagai pendekatan alternatif dari perulangan biasa (for/while), yang dapat memperkaya cara berpikir mahasiswa terhadap pemecahan masalah dalam algoritma.

3. PEMBAHASAN HASIL

Aplikasi diuji coba dengan menggunakan sampel sebanyak 5 data mahasiswa. Setiap data berisi informasi nama, NIM, dan nilai. Setelah proses input selesai, program secara otomatis melakukan pengurutan berdasarkan nilai, menentukan predikat nilai, serta menampilkan hasilnya dalam format yang rapi. Selain itu, fungsi pencarian dan rekursif juga diuji sebagai bagian dari kelengkapan fitur utama.

3.1 Hasil Tampil Output

```
--- Data Mahasiswa ---  
Nama : Andi, NIM : 123, Nilai : 89, Predikat: A  
Nama : Budi, NIM : 125, Nilai : 76, Predikat: B  
Nama : Cici, NIM : 124, Nilai : 62, Predikat: C  
Faktorial dari jumlah mahasiswa (3) = 6
```

Gambar 7. Tampilan Output Data Mahasiswa

Dari hasil di atas, terlihat bahwa proses pengurutan berhasil menempatkan data berdasarkan urutan nilai tertinggi ke terendah, dan sistem mampu menampilkan predikat yang sesuai dengan ketentuan logika if- else.

3.2 Analisis

Analisis fungsional menunjukkan bahwa seluruh fitur inti bekerja dengan baik, di antaranya: (a) Sorting berjalan dengan lancar dan akurat, sesuai dengan urutan nilai menurun (*descending*); (b) Pencarian berhasil menemukan posisi data mahasiswa dengan menggunakan *sequential search*, meskipun belum seefisien binary search; (c) Predikat ditentukan secara tepat, mengikuti kondisi nilai menggunakan logika percabangan.; (d) Fungsi rekursif, seperti menampilkan data dan menghitung faktorial, berhasil dijalankan dengan benar untuk kasus jumlah data kecil.

Keberhasilan ini menunjukkan bahwa aplikasi memenuhi harapan sebagai alat bantu sederhana dalam mengelola data akademik mahasiswa.

3.3 Kelebihan

Aplikasi ini memiliki sejumlah keunggulan yang mendukung efektivitasnya sebagai media pembelajaran pemrograman. Salah satu keunggulan utamanya terletak pada struktur kode yang disusun secara modular dan ringkas, sehingga setiap fungsi memiliki tanggung jawab yang jelas dan terpisah. Pendekatan modular ini tidak hanya meningkatkan keterbacaan kode, tetapi juga memudahkan proses pemeliharaan dan pengembangan lebih lanjut. Dengan pembagian fungsi yang terstruktur, pengguna—khususnya mahasiswa atau pelajar—dapat lebih mudah memahami alur logika program serta keterkaitan antarbagian dalam sistem secara menyeluruh. Selain itu, aplikasi ini memiliki tingkat portabilitas yang tinggi karena dapat dijalankan pada berbagai compiler C++ seperti GCC, Code::Blocks, maupun Visual Studio. Fleksibilitas ini memungkinkan pengguna untuk mengimplementasikan dan menguji program pada beragam lingkungan pengembangan tanpa kendala kompatibilitas yang berarti. Dari sisi pedagogis, aplikasi ini sangat relevan digunakan dalam proses pembelajaran dasar-dasar pemrograman, terutama dalam memahami konsep pemrograman struktural, algoritma pengurutan (*sorting*), pencarian (*searching*), serta penggunaan pointer. Lebih jauh lagi, aplikasi ini memberikan representasi nyata mengenai bagaimana teori algoritma dan struktur data yang dipelajari di kelas dapat diimplementasikan dalam skenario pengelolaan data yang konkret, sehingga membantu menjembatani pemahaman antara aspek teoretis dan praktik pemrograman secara langsung.

3.4 Kekurangan

Meskipun aplikasi ini memiliki berbagai keunggulan dari sisi struktur dan fungsionalitas, masih terdapat beberapa keterbatasan yang perlu diperhatikan. Salah satu keterbatasan utama adalah belum adanya mekanisme penyimpanan data ke dalam file eksternal (*file I/O*), sehingga setiap kali program dijalankan ulang, seluruh data harus diinput kembali dari awal. Kondisi ini tentu membatasi keberlanjutan penggunaan aplikasi dalam jangka panjang, terutama apabila jumlah data yang dikelola semakin besar dan membutuhkan penyimpanan yang bersifat permanen. Selain itu, aplikasi ini juga belum dilengkapi dengan sistem validasi input yang memadai untuk meminimalkan kesalahan pengguna. Sebagai contoh, program belum mampu mendeteksi apabila pengguna memasukkan karakter huruf pada kolom yang seharusnya diisi angka, atau sebaliknya memasukkan angka pada bagian yang diperuntukkan bagi teks seperti nama. Ketiadaan validasi

ini berpotensi menimbulkan error saat proses eksekusi maupun menghasilkan data yang tidak konsisten. Di samping itu, aplikasi juga belum mendukung penggunaan tipe data karakter seperti `char` atau enumerasi (`enum`) untuk merepresentasikan atribut tertentu, misalnya jenis kelamin atau status mahasiswa, sehingga representasi data masih bersifat sederhana dan belum terstandarisasi secara optimal. Keterbatasan-keterbatasan tersebut pada dasarnya membuka peluang pengembangan lebih lanjut di masa mendatang. Dengan menambahkan fitur penyimpanan data, validasi input yang lebih komprehensif, serta dukungan tipe data yang lebih variatif, aplikasi ini berpotensi untuk dikembangkan menjadi sistem yang lebih robust dan siap digunakan dalam skala yang lebih besar maupun dalam konteks implementasi nyata di lingkungan institusi pendidikan.

4. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa aplikasi manajemen data mahasiswa berbasis C++ ini berhasil dikembangkan sesuai dengan tujuan penelitian. Aplikasi ini mampu mendukung pengguna dalam memahami serta menerapkan berbagai konsep dasar pemrograman, seperti pointer, fungsi, algoritma pengurutan (`sorting`), pencarian (`searching`), dan rekursi. Seluruh fitur yang dibangun dapat berjalan dengan baik dan mampu mengelola data mahasiswa secara sederhana namun sistematis. Selain itu, program juga menyajikan keluaran yang informatif, termasuk penentuan predikat nilai secara otomatis serta pemanfaatan fungsi rekursif pada proses tertentu sebagai bentuk implementasi konsep lanjutan dalam pemrograman. Hasil uji coba menunjukkan bahwa aplikasi ini sangat sesuai digunakan dalam konteks pembelajaran karena strukturnya yang jelas dan modular, sehingga memudahkan mahasiswa dalam memahami alur logika program. Struktur kode yang fleksibel juga memungkinkan pengembangan lebih lanjut untuk kebutuhan yang lebih kompleks, dan setiap fungsi terbukti bekerja sesuai ekspektasi pada pengolahan data berskala kecil hingga menengah. Untuk pengembangan di masa mendatang, aplikasi ini dapat ditingkatkan dengan menambahkan fitur penyimpanan data menggunakan file eksternal (`file I/O`), memperkuat validasi input agar lebih tahan terhadap kesalahan pengguna, serta mengembangkan antarmuka grafis yang lebih interaktif dan ramah pengguna. Dengan pengembangan tersebut, aplikasi ini diharapkan tidak hanya berfungsi sebagai media pembelajaran, tetapi juga berpotensi menjadi sistem informasi akademik sederhana yang dapat diterapkan dalam lingkup institusi pendidikan.

REFERENCES

- [1] M. Rizki Andrian Fitra, A. Afrahman S. Effendi, and F. Ramadhani, "IMPLEMENTASI PYTHON DALAM PENGOLAHAN DATA PRIBADI MAHASISWA ILMU KOMPUTER ANGKATAN 23 PADA UNIVERSITAS NEGERI MEDAN MENGGUNAKAN STRUKTUR DATA LINKED LIST," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 1, pp. 51–58, Dec. 2024, doi: 10.36040/jati.v9i1.12164.
- [2] R. Resti, R. A. Wati, S. Ma'Arif, and S. Syarifuddin, "Pemanfaatan Media Pembelajaran Berbasis Teknologi sebagai Alat Untuk Meningkatkan Kemampuan Literasi Digital Siswa Sekolah Dasar," *Al Madrasah Jurnal Pendidikan Madrasah Ibtidaiya*, vol. 8, no. 3, p. 1145, Jul. 2024, doi: 10.35931/am.v8i3.3563.
- [3] R. T. Caesarajmi, J. Raya, T. No, K. Gedong, P. Rebo, and J. Timur, "APLIKASI REKAPITULASI DATA MAHASISWA SEKOLAH VOKASI IPB BERBASIS WEB," *Jurnal Riset dan Aplikasi Mahasiswa Informatika (JRAMI)*, vol. 01, 2020.
- [4] I. Ramadhana and B. Sujatmiko, "PENGEMBANGAN APLIKASI KAMUS BAHASA PEMROGRAMAN C++ BERBASIS ANDROID UNTUK MENINGKATKAN KOMPETENSI KOGNITIF MATA KULIAH STRUKTUR DATA."
- [5] A. Hendra Saptadi and D. Windi Sari, "ANALISIS ALGORITMA INSERTION SORT, MERGE SORT DAN IMPLEMENTASINYA DALAM BAHASA PEMROGRAMAN C++," 2012.
- [6] M. Y. Arianti, N. Fitriani, D. Khairani, and S. Tree Adinda, "ANALISIS NILAI AKHIR RAPOR DENGAN PROGRAM C++ SMKISLAMIYAH SEIKAMAHII2021." [Online]. Available: <http://pkm.uika-bogor.ac.id/index.php/pkm-p/issue/archive>
- [7] R. Amalia and E. D. Putra, "PENGEMBANGAN MODUL MATA KULIAH ALGORITMA DAN PEMROGRAMAN (PEMROGRAMAN C++)".
- [8] I. Ice *et al.*, "PENGENALAN PEMROGRAMAN DASAR DUNIA KODING DENGAN C++."

- [9] T. N. Turnip, W. Suarez Lumbantobing, D. C. Sitorus, and F. L. Sianturi, "SOFTWARE WATERMARKING DINAMIS DENGAN ALGORITME COLLBERG-THOMBORSON DAN PARENT POINTER GRAF PADA APLIKASI ANDROID", doi: 10.25126/jtiik.202184500.
- [10] N. Sari, W. A. Gunawan, P. K. Sari, I. Zikri, and A. Syahputra, "Analisis Algoritma Bubble Sort Secara Ascending Dan Descending Serta Implementasinya Dengan Menggunakan Bahasa Pemrograman Java," *ADI Bisnis Digital Interdisiplin Jurnal*, vol. 3, no. 1, pp. 16–23, Jan. 2022, doi: 10.34306/abdi.v3i1.625.
- [11] B. Tahun, "MIFORTEKH (Jurnal Manajemen Informatika & Teknologi) Integrasi Algoritma Rekursif pada Pemrosesan Data Multilevel untuk Aplikasi Berbasis AI," vol. XX, [Online]. Available: <https://journal.stiestekom.ac.id/index.php/mifortekh>
- [12] M. Nugraha, L. Sakinah, R. A. Setiawan, and H. Mulyani, "RANCANG BANGUN SISTEM INFORMASI PENERIMAAN MAHASISWA BARU BERBASIS WEB DENGAN MENGGUNAKAN FRAMEWORK LARAVEL," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, Apr. 2024, doi: 10.23960/jitet.v12i2.4179.
- [13] J. Teknik Informatika, P. Negeri Tanah Laut Jl AYani Km, and P. Tanah Laut Kalimantan Selatan, "Aplikasi Pengelolaan Nilai Akademik Mahasiswa dan DPNA (Daftar Peserta dan Nilai Akhir) Herpendi," 2016.
- [14] S. Mauluddin, "SISTEM INFORMASI REKAPITULASI DAFTAR HADIR MAHASISWA (STUDI KASUS : PROGRAM STUDI MANAJEMEN INFORMATIKA)."
- [15] M. Nanak Zakaria, Y. Heru Prasetyo Isnomo, P. Studi Jaringan Telekomunikasi Digital, J. Teknik Elektro, and P. Negeri Malang, "Rancang Bangun Sistem Rekap Nilai Mahasiswa Prodi Jaringan Telekomunikasi Digital Politeknik Negeri Malang Menggunakan Jaringan Fiber Optik Berbasis Website".
- [16] E. I. Fenetiruma and J. Lahallo, "RANCANG BANGUN SISTEM PENDATAAN MAHASISWA DAN REKAP NILAI, PRESENSI PRAKTIKUM STIMIK SEPULUH NOPEMBER JAYAPURA BERBASIS WEB Designing Web-based Application of Students Identity, Value Recap, and Attendance of Practice Activities in STIMIK Sepuluh Nopember Jayapura," 2023.
- [17] A. Santoso, A. Irawan, and A. Irawan, "Perancangan Sistem Informasi Penjualan Alat Tulis pada Toko Nusantara Barokah Berbasis Java," *Jurnal Riset dan Aplikasi Mahasiswa Informatika (JRAMI)*, vol. 5, no. 1, pp. 125–132, Jan. 2024, doi: 10.30998/jrami.v5i1.8324.