Journal Of Global Computer Science

Volume 2 ; Number 2 ; August 2026 ; Page 1-9
Doi : https://doi.org/10.59435/jgcs.v2i2.2026.52
Web : https://journal.padangtekno.com/index.php/jgcs
E-ISSN : 3032-372X

Implementasi Struktur Data Linked List Berbasis Python Dalam Pengelolaan Data Mahasiswa Dinamis Pada Sistem

Endang Pramita Nainggolan[1*], Juniar Cicilia Sitohang[2], Ekha Gabena Siahaan[3], Indra Gunawan[4]

STIKOM Tunas Bangsa
Program Studi Teknik Informatika, Sekolah Tinggi Ilmu Komputer Tunas Bangsa , Jl. Jendral Sudirman Blok A/B No. 1,2,3, Kota Pematang Siantar, Sumatera Utara 21143
Email: [1]mitanainggolan00@email.com, [2]juniarsitohang339@email.com, [3]ekhasiahaan@email.com

ABSTRAK

Perkembangan teknologi informasi memberikan pengaruh besar dalam pengelolaan data di berbagai bidang, termasuk pendidikan. Pengelolaan data mahasiswa memerlukan sistem yang efektif agar proses penyimpanan, pencarian, penambahan, dan penghapusan data dapat dilakukan dengan mudah. Salah satu struktur data yang dapat digunakan adalah *Linked List*. *Linked List* merupakan struktur data linear yang terdiri dari kumpulan node yang saling terhubung melalui pointer. Penelitian ini bertujuan untuk memahami implementasi *Linked List* pada pengelolaan data mahasiswa serta mengetahui kelebihan dan kekurangannya. Metode penelitian yang digunakan adalah metode deskriptif dengan teknik pengumpulan data melalui studi literatur dan observasi. Implementasi dilakukan menggunakan bahasa pemrograman Python dengan beberapa fitur utama seperti menambahkan data mahasiswa, menampilkan data, mencari data, dan menghapus data. Hasil penelitian menunjukkan bahwa penggunaan *Linked List* mempermudah pengelolaan data secara dinamis dibandingkan array, terutama dalam proses penambahan dan penghapusan data tanpa perlu menggeser elemen lain. Selain memiliki fleksibilitas tinggi, *Linked List* juga memiliki kelemahan yaitu membutuhkan memori tambahan untuk pointer dan proses pencarian data yang relatif lebih lambat. Dengan demikian, *Linked List* dapat menjadi salah satu solusi yang efektif dalam pengelolaan data mahasiswa pada sistem berbasis komputer.

Kata kunci: algoritma, Linked List, pengelolaan data mahasiswa, Python, struktur data.

PENDAHULUAN

Perkembangan teknologi informasi memberikan pengaruh yang sangat besar dalam berbagai bidang kehidupan, termasuk bidang pendidikan. Pemanfaatan teknologi dalam pengelolaan data akademik menjadi salah satu faktor penting untuk meningkatkan efektivitas dan efisiensi administrasi di lingkungan perguruan tinggi (Anto, 2017). Pengelolaan data mahasiswa yang meliputi identitas mahasiswa, nomor induk mahasiswa (NIM), program studi, semester, serta data akademik lainnya memerlukan sistem yang mampu menyimpan dan mengolah data secara cepat, tepat, dan terstruktur.

Dalam ilmu komputer, struktur data merupakan komponen penting yang digunakan untuk mengatur penyimpanan dan pengolahan data agar proses manipulasi data dapat dilakukan dengan lebih efisien (Karumanchi, 2016; Lee & Hubbard, 2015). Salah satu struktur data yang banyak digunakan adalah *Linked List* (Wijaya et al., 2018). *Linked List* merupakan struktur data linear yang terdiri dari kumpulan node, di mana setiap node memiliki data dan pointer yang menghubungkan node dengan node lainnya (Karumanchi, 2016).

Struktur data ini memungkinkan pengelolaan data dilakukan secara dinamis tanpa harus menentukan ukuran data di awal seperti pada array (Lee & Hubbard, 2015).

Penggunaan *Linked List* dalam pengelolaan data mahasiswa memiliki beberapa kelebihan. Proses penambahan dan penghapusan data dapat dilakukan dengan mudah tanpa perlu menggeser elemen lainnya (Sofianti et al., 2025). Selain itu, penggunaan memori menjadi lebih fleksibel karena alokasi memori dilakukan sesuai kebutuhan (Karumanchi, 2016). Struktur data ini sangat sesuai digunakan pada sistem yang memiliki perubahan data secara terus-menerus, seperti sistem pengelolaan data mahasiswa.

Beberapa penelitian sebelumnya menunjukkan bahwa *Linked List* dapat digunakan dalam berbagai sistem pengolahan data karena memiliki kemampuan manipulasi data yang baik (Wijaya et al., 2018; Mbejo et al., 2025). Struktur data ini sering diterapkan pada sistem antrian, manajemen data, dan aplikasi berbasis database sederhana. Penelitian oleh Sofianti et al. (2025) membahas implementasi struktur data array dan *Linked List* dalam pengelolaan data mahasiswa, sedangkan penelitian Fitra et al. (2024) menerapkan Python dalam pengolahan data mahasiswa menggunakan *Linked List*. Namun, penelitian mengenai implementasi *Linked List* pada pengelolaan data mahasiswa sederhana berbasis Python masih perlu dipahami lebih lanjut agar dapat diketahui cara kerja, kelebihan, dan kekurangannya dalam penerapan nyata.

Permasalahan yang sering terjadi dalam pengelolaan data mahasiswa adalah kesulitan dalam menambah, menghapus, dan mencari data secara efisien ketika jumlah data terus bertambah. Penggunaan array memiliki keterbatasan karena ukuran data harus ditentukan terlebih dahulu dan proses penghapusan data memerlukan pergeseran elemen (Rudianto et al., 2025). Oleh karena itu, diperlukan alternatif struktur data yang lebih fleksibel untuk mengatasi permasalahan tersebut.

Penelitian ini bertujuan untuk mengimplementasikan struktur data *Linked List* pada pengelolaan data mahasiswa menggunakan bahasa pemrograman Python (Fitra et al., 2024). Implementasi dilakukan melalui pembuatan program sederhana yang mampu melakukan proses penambahan, pencarian, penampilan, dan penghapusan data mahasiswa. Kebaruan penelitian ini terletak pada penerapan konsep *Linked List* dalam sistem pengelolaan data mahasiswa secara sederhana dan mudah dipahami sebagai media pembelajaran struktur data bagi mahasiswa informatika.

Hasil penelitian diharapkan dapat memberikan pemahaman mengenai penerapan *Linked List* dalam pengelolaan data mahasiswa serta menjadi referensi pembelajaran dalam memahami konsep struktur data dan algoritma pada bidang informatika (Lestanto & Bakrie, 2023; Santoso, 2024).

METODE

Tahapan Penelitian

Penelitian ini menggunakan metode deskriptif dengan pendekatan kualitatif. Metode deskriptif digunakan untuk menggambarkan secara sistematis implementasi struktur data *Linked List* dalam pengelolaan data mahasiswa. Pendekatan kualitatif dipilih karena penelitian lebih berfokus pada proses kerja sistem, alur pengolahan data, serta analisis terhadap kelebihan dan kekurangan *Linked List* dibandingkan struktur data lainnya (Karumanchi, 2016; Lee & Hubbard, 2015).

Tahapan penelitian dilakukan secara bertahap agar sistem yang dibangun sesuai dengan konsep yang telah dirancang. Penelitian dimulai dari identifikasi masalah, studi literatur, perancangan sistem, implementasi program, pengujian sistem, hingga analisis hasil penelitian. Tahapan tersebut dilakukan untuk memastikan sistem dapat berjalan sesuai dengan tujuan penelitian dan mampu mengelola data mahasiswa secara efektif.

Pada tahap identifikasi masalah dilakukan pengamatan terhadap proses pengelolaan data mahasiswa yang masih memiliki keterbatasan dalam proses penambahan, pencarian, dan penghapusan data. Selanjutnya dilakukan studi literatur dengan mengumpulkan referensi dari buku, jurnal ilmiah, dan sumber daring yang berkaitan dengan struktur data *Linked List* sebagai dasar teori penelitian (Lestanto & Bakrie, 2023).

Tahap perancangan sistem dilakukan dengan merancang struktur node dan fungsi-fungsi utama yang digunakan dalam pengelolaan data mahasiswa. Sistem dibangun menggunakan konsep *Singly Linked List* yang terdiri dari node-node saling terhubung menggunakan pointer. Setiap node menyimpan data mahasiswa berupa nama, NIM, program studi, semester, dan nilai.

Implementasi sistem dilakukan menggunakan bahasa pemrograman Python dengan menerapkan operasi utama pada *Linked List*, yaitu *insertion*, *traversal*, *search*, dan *deletion* (Fitra et al., 2024). Operasi *insertion*

digunakan untuk menambahkan data mahasiswa, *traversal* digunakan untuk menampilkan seluruh data mahasiswa, *search* digunakan untuk mencari data berdasarkan NIM, sedangkan *deletion* digunakan untuk menghapus data mahasiswa tertentu.

Pengujian sistem dilakukan menggunakan metode *Black Box Testing* untuk memastikan seluruh fungsi berjalan sesuai rancangan. Pengujian dilakukan pada fitur tambah data, tampil data, cari data, dan hapus data mahasiswa. Hasil pengujian digunakan sebagai dasar analisis terhadap efektivitas penerapan *Linked List* dalam pengelolaan data mahasiswa.

Tabel 1. Pengujian Operasi Sistem Linked List Pada Pengelolaan Data Mahasiswa

No	Fungsi Sistem	Skenario Pengujian	Input Data	Hasil yang Diharapkan	Hasil Pengujian
1	Tambah Data	Menambahkan data mahasiswa baru	Nama, NIM, Prodi, Semester, Nilai	Data tersimpan dalam Linked List	Berhasil
2	Tampil Data	Menampilkan seluruh data mahasiswa	-	Semua data tampil berurutan dari head	Berhasil
3	Cari Data	Mencari data berdasarkan NIM	NIM mahasiswa	Data ditemukan dan ditampilkan	Berhasil
4	Hapus Data	Menghapus data berdasarkan NIM	NIM mahasiswa	Data terhapus tanpa merusak struktur Linked List	Berhasil
5	Validasi Data Kosong	Menampilkan data saat list kosong	-	Sistem menampilkan pesan “data kosong”	Berhasil
6	Insert Banyak Data	Menambahkan beberapa data sekaligus	Beberapa data mahasiswa	Data tersimpan sesuai urutan input	Berhasil

Metode Penyelesaian Masalah

Metode penyelesaian masalah pada penelitian ini dilakukan dengan menerapkan struktur data *Singly Linked List* pada sistem pengelolaan data mahasiswa. Struktur data ini dipilih karena memiliki fleksibilitas dalam proses manipulasi data tanpa harus menentukan kapasitas data di awal seperti pada array (Rudianto et al., 2025).

Sistem yang dibangun terdiri dari beberapa operasi utama, yaitu penambahan data (*insertion*), penampilan data (*traversal*), pencarian data (*search*), dan penghapusan data (*deletion*). Setiap operasi dilakukan dengan memanfaatkan hubungan antar node melalui pointer sehingga data dapat dikelola secara dinamis.

Teknik pengumpulan data dilakukan melalui studi literatur dan observasi. Studi literatur dilakukan dengan mengumpulkan referensi yang berkaitan dengan struktur data *Linked List*, algoritma, dan implementasi Python dalam pengelolaan data mahasiswa. Observasi dilakukan dengan mengamati bagaimana struktur data diterapkan pada sistem pengelolaan data sederhana (Nopriandi, 2018).

Analisis data dilakukan secara deskriptif dengan membandingkan teori struktur data *Linked List* dengan hasil implementasi program (Wijaya et al., 2018; Sofianti et al., 2025). Fokus analisis meliputi efektivitas proses penambahan dan penghapusan data, performa pencarian data yang bersifat *sequential*, serta penggunaan memori yang dinamis karena setiap node dialokasikan secara terpisah. Hasil analisis digunakan untuk mengetahui apakah *Linked List* layak digunakan dalam sistem pengelolaan data mahasiswa sederhana.

HASIL DAN PEMBAHASAN

Hasil Implementasi Sistem

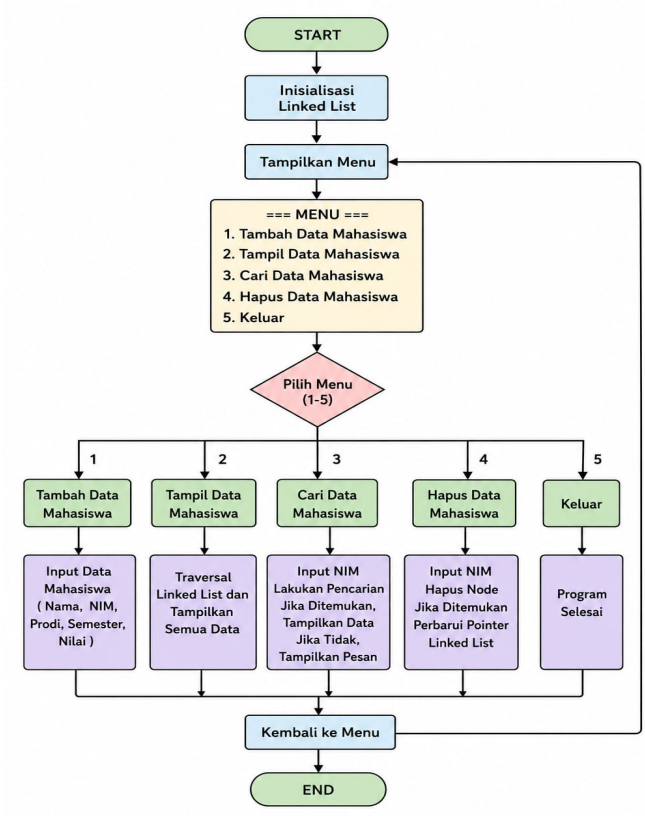
Hasil penelitian ini berupa sistem pengelolaan data mahasiswa menggunakan struktur data *Singly Linked List* berbasis Python (Fitra et al., 2024). Sistem yang dibangun mampu melakukan beberapa operasi utama, yaitu menambah data, menampilkan data, mencari data, dan menghapus data mahasiswa secara dinamis.

Setiap data mahasiswa disimpan dalam bentuk node yang berisi atribut nama, NIM, program studi, semester, dan nilai. Node-node tersebut saling terhubung menggunakan pointer *next* sehingga membentuk struktur *Linked List* (Karumanchi, 2016). Struktur ini memungkinkan data dikelola secara fleksibel tanpa perlu menentukan ukuran data di awal seperti pada array (Lee & Hubbard, 2015).

Penggunaan *Linked List* pada sistem ini memberikan kemudahan dalam proses manipulasi data, terutama pada proses penambahan dan penghapusan data mahasiswa. Sistem dapat menambahkan data baru tanpa perlu menggeser elemen lain sehingga proses pengolahan data menjadi lebih efisien (Sofianti et al., 2025).

Flowchart Sistem

Flowchart digunakan untuk menggambarkan alur kerja sistem sebelum program diimplementasikan ke dalam bahasa pemrograman Python (Lestanto & Bakrie, 2023). Flowchart membantu dalam memahami proses pengelolaan data mahasiswa mulai dari input data, proses penyimpanan data ke dalam *Linked List*, pencarian data, hingga penghapusan data mahasiswa.



Gambar 1. Flowchart Implementasi *Singly Linked List* pada Pengelolaan Data Mahasiswa

Implementasi Program

Implementasi program dilakukan menggunakan bahasa pemrograman Python dengan menerapkan konsep *Object Oriented Programming* (OOP) dan struktur data *Singly Linked List* (Fitra et al., 2024). Program terdiri dari class node dan class *Linked List* yang digunakan untuk mengelola data mahasiswa.
Beberapa fungsi utama yang diterapkan dalam program meliputi:

a. Fungsi tambah data mahasiswa ke dalam *Linked List*.
b. Fungsi menampilkan seluruh data mahasiswa.
c. Fungsi pencarian data berdasarkan NIM mahasiswa.
d. Fungsi penghapusan data mahasiswa berdasarkan NIM tertentu.

Implementasi ini menunjukkan bahwa *Linked List* mampu menangani pengelolaan data secara dinamis tanpa memerlukan alokasi memori berurutan seperti array (Wijaya et al., 2018). Selain itu, penggunaan Python mempermudah proses implementasi program karena memiliki sintaks yang sederhana dan mudah dipahami. Berikut implementasi Linked List pada pengelolaan data mahasiswa menggunakan bahasa pemrograman Python:

```python
class Node:
    def __init__(self, nama, nim, prodi, semester, nilai):
        self.nama = nama
        self.nim = nim
        self.prodi = prodi
        self.semester = semester
        self.nilai = nilai
        self.next = None


class LinkedList:
    def __init__(self):
        self.head = None

    def tambah(self, nama, nim, prodi, semester, nilai):
        baru = Node(nama, nim, prodi, semester, nilai)

        if self.head is None:
            self.head = baru
        else:
            temp = self.head
            while temp.next:
                temp = temp.next
            temp.next = baru

    def tampil(self):
        temp = self.head
        if temp is None:
            print("Data kosong")
            return

        while temp:
            print(temp.nama, temp.nim, temp.prodi, temp.semester, temp.nilai)
            temp = temp.next

    def cari(self, nim):
        temp = self.head
        while temp:
            if temp.nim == nim:
                print("Data ditemukan:", temp.nama)
                return
            temp = temp.next
        print("Data tidak ditemukan")
```

```python

    def hapus(self, nim):
        temp = self.head
        prev = None

        if temp and temp.nim == nim:
            self.head = temp.next
            print("Data berhasil dihapus")
            return

        while temp and temp.nim != nim:
            prev = temp
            temp = temp.next

        if temp is None:
            print("Data tidak ditemukan")
            return

        prev.next = temp.next
        print("Data berhasil dihapus")


ll = LinkedList()

while True:
    print("\n1. Tambah")
    print("2. Tampil")
    print("3. Cari")
    print("4. Hapus")
    print("5. Keluar")

    pilih = input("Pilih: ")

    if pilih == "1":
        nama = input("Nama: ")
        nim = input("NIM: ")
        prodi = input("Prodi: ")
        semester = input("Semester: ")
        nilai = input("Nilai: ")
        ll.tambah(nama, nim, prodi, semester, nilai)

    elif pilih == "2":
        ll.tampil()

    elif pilih == "3":
        nim = input("Cari NIM: ")
        ll.cari(nim)

    elif pilih == "4":
        nim = input("Hapus NIM: ")
        ll.hapus(nim)

    elif pilih == "5":
        break
```

Gambar 2. Implementasi Program Python *Linked List* Data Mahasiswa

Hasil Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan seluruh fungsi program berjalan sesuai dengan rancangan yang telah dibuat (Nopriandi, 2018). Pengujian dilakukan terhadap fitur tambah data, tampil data, cari data, dan hapus data mahasiswa.

Hasil pengujian menunjukkan bahwa seluruh proses pengolahan data dapat berjalan dengan baik tanpa terjadi kesalahan pada sistem. Data mahasiswa dapat ditambahkan ke dalam *Linked List*, ditampilkan secara berurutan, dicari berdasarkan NIM, serta dihapus sesuai input pengguna.

a. Pengujian tambah data berhasil menyimpan data mahasiswa ke dalam sistem.
b. Pengujian tampil data berhasil menampilkan seluruh data mahasiswa secara berurutan.
c. Pengujian pencarian data berhasil menemukan data mahasiswa berdasarkan NIM.
d. Pengujian penghapusan data berhasil menghapus data mahasiswa tanpa merusak struktur *Linked List*.

```
1. Tambah
2. Tampil
3. Cari
4. Hapus
5. Keluar
Pilih: 1
Nama       : Budi
NIM        : 250100123
Prodi      : Teknik Informatika
Semester   : 2
Nilai      : 90
```

Gambar 3. Output Penambahan Data Mahasiswa

```
1. Tambah
2. Tampil
3. Cari
4. Hapus
5. Keluar
Pilih: 2
Budi 250100123 Teknik Informatika 2 90
```

Gambar 4. Output Tampil Data Mahasiswa

```
1. Tambah
2. Tampil
3. Cari
4. Hapus
5. Keluar
Pilih: 3
Cari NIM: 250100123
Data ditemukan: Budi
```

Gambar 5. Output Pencarian Data Mahasiswa

```
1. Tambah
2. Tampil
3. Cari
4. Hapus
5. Keluar
Pilih: 4
Hapus NIM: 250100123
Data berhasil dihapus
```

Gambar 6. Output Penghapusan Data Mahasiswa

Pengujian Black Box Testing

Metode *Black Box Testing* digunakan untuk menguji fungsi sistem tanpa melihat struktur kode program (Lestanto & Bakrie, 2023). Pengujian dilakukan untuk mengetahui apakah sistem telah berjalan sesuai dengan kebutuhan fungsional yang telah dirancang.

Tabel 2 Hasil Pengujian Sistem Pengelolaan Data Mahasiswa

No	Pengujian	Input	Hasil yang Diharapkan	Hasil
1	Tambah data	Data mahasiswa	Data tersimpan	Valid
2	Tampil data	Menu tampil	Semua data muncul	Valid
3	Cari data	NIM 22002	Data ditemukan	Valid
4	Hapus data	NIM 22001	Data terhapus	Valid
5	Data kosong	Tampil data	Muncul pesan “Data kosong”	Valid

KESIMPULAN

Berdasarkan hasil penelitian dan implementasi sistem pengelolaan data mahasiswa menggunakan struktur data *Singly Linked List* berbasis Python, dapat disimpulkan bahwa struktur data *Linked List* efektif digunakan dalam pengelolaan data yang bersifat dinamis. Sistem yang dibangun mampu menjalankan operasi utama seperti penambahan data, penampilan data, pencarian data, dan penghapusan data mahasiswa dengan baik sesuai rancangan program. Hasil pengujian menggunakan metode *Black Box Testing* menunjukkan bahwa seluruh fungsi sistem berjalan dengan valid tanpa mengalami kesalahan sehingga kebutuhan fungsional sistem telah terpenuhi. Selain memiliki kelebihan dalam proses manipulasi data yang fleksibel, *Linked List* juga memiliki keterbatasan pada proses pencarian data yang masih bersifat *sequential* sehingga kurang efisien ketika jumlah data semakin besar. Dengan demikian, struktur data *Linked List* cocok digunakan pada sistem pengelolaan data mahasiswa sederhana hingga menengah yang membutuhkan fleksibilitas dalam pengolahan data.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak STIKOM Tunas Bangsa, dosen pembimbing, serta semua pihak yang telah memberikan dukungan, bantuan, dan arahan dalam proses penyusunan penelitian ini. Ucapan terima kasih juga disampaikan kepada rekan-rekan yang telah membantu dalam proses pengumpulan data, implementasi program, serta pengujian sistem sehingga penelitian mengenai implementasi struktur data *Linked List* dalam pengelolaan data mahasiswa ini dapat diselesaikan dengan baik. Penulis berharap penelitian ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan, khususnya pada bidang struktur data dan pemrograman.

DAFTAR PUSTAKA

[1]. City, W. J. (n.d.). *PERBANDINGAN ALGORITMA BOOSTING UNTUK VULKANIK*. *d*.
[2]. Kasus, S., Putusan, K., & Konstitusi, M. (2025). *Jurnal Sains Informatika Terapan (JSIT) Jurnal Sains Informatika Terapan (JSIT)*. (1), 187–201.
[3]. Lee, K. D., & Hubbard, S. (2015). *Undergraduate Topics in Computer Science Data Structures and Algorithms with Python*. http://link.springer.com/book/10.1007/978-3-319-13072-9
[4]. Lestanto, Y., & Bakrie, U. (2023). *Modul Panduan Struktur Data Menggunakan MATLAB*.
[5]. Lucky E. Santoso. (2024). Standard Template Library C++ Untuk Mengajarkan Struktur Data . *Jurnal FASILKOM*, *2*(2), 7–15.
[6]. Mbejo, M. T., Nopa, L. A., Putri, J. S., & Risky, M. (2025). *Jurnal Sains Informatika Terapan (JSIT) ANALISIS STRUKTUR DATA LINKED LIST DALAM PENGOLAHAN*. 441–444.
[7]. Narasimha Karumanchi. (2016). *Data Structures And Algorithmic Thinking With Python*. 470. https://www.amazon.com/Data-Structure-Algorithmic-Thinking-Python/dp/8192107590
[8]. Nopriandi, H. (2018). Perancangan Sistem Informasi Registrasi Mahasiswa. *Jurnal Teknologi Dan Open Source*, *1*(1), 73–79. https://doi.org/10.36378/jtos.v1i1.1
[9]. Parlante, B. N. (2002). Stantford C library. *Review Literature And Arts Of The Americas*.
[10]. Rizki Andrian Fitra, M., Afrrahman S. Effendi, A., & Ramadhani, F. (2024). 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)*, *9*(1), 51–58. https://doi.org/10.36040/jati.v9i1.12164
[11]. Rizq Daffa Jodi, M., & komputer, F. (n.d.). *komputer Algoritma dan Struktur data*.
[12]. Rudianto, Sapaatullah, A., Rakhim Setya Permana, B., & Darip, M. (2025). Implementasi Struktur Data Array dalam Sistem Perpustakaan Berbasis Web dengan Python Flask. *Buletin Ilmiah Informatika Teknologi*, *3*(2), 30–36. https://doi.org/10.58369/biit.v3i2.91
[13]. Sari, N. N. K., & Pranatawijaya, V. H. (2021). adminjti,+4.+Jurnal+Nova+141+-+151. *Sistem Informasi Mahasiswa Berprestasi Universitas Palangka Raya Berbasis Website*, *15*(2), 141–151.
[14]. Sofianti, H. A., Manullang, Y. V., Tampubolon, N. A., Naibaho, L. H., & Gunawan, I. (2025). Implementasi Struktur Data Array Dan Linked List Dalam Pengelolaan Data Mahasiswa. *Menulis: Jurnal Penelitian Nusantara*, *1*(6), 871–877. https://padangjurnal.web.id/index.php/menulis/article/view/417

[15]. Susandi, D., Karyaningsih, D., & Suryani, S. (2023). *Tradisional Berbasis Android*. *10*(1), 7–11.
[16]. Wijaya, H., Wardhono, W. S., & Arwani, I. (2018). Implementasi Linked List pada Interaksi Antar Marker Augmented Reality untuk Operand dan Operator Aritmetika. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, *2*(9), 3328–3332.