

Penerapan Array Dalam Mengelola Data CPNS Untuk Menampilkan Data Pelamar

Ahmad Denil¹ Bob Valentino² Idris Putra³ Pedro Mario⁴

Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Medan, Kota Medan, Provinsi Sumatera Utara, Indonesia^{1,2,3,4}

Email: adenilsitepu@gmail.com¹ bobvalentino46@gmail.com² idrisputra76@gmail.com³ 08mariowr@gmail.com⁴

Abstrak

Penerapan array dalam mengelola data CPNS untuk menampilkan data pelamar merupakan topik yang relevan dalam pengembangan sistem informasi. Abstrak ini akan membahas tentang cara efektif menggunakan array untuk menyimpan, mengelola, dan menampilkan data pelamar CPNS. Dengan menggunakan array, proses penyimpanan dan pengelolaan data pelamar dapat dilakukan secara efisien dan akurat. Fokus utama adalah pada deklarasi array, penyimpanan data pelamar, dan penggunaan array untuk menampilkan data pelamar yang lulus atau gagal dalam tes SKD.

Kata Kunci: Deklarasi Array, Pengelolaan Data, Sistem Informasi, Data Pelamar

Abstract

The application of arrays in managing CPNS data to display applicant data is a relevant topic in information system development. This abstract will discuss how to effectively use arrays to store, manage and display CPNS applicant data. By using arrays, the process of storing and managing applicant data can be carried out efficiently and accurately. The main focus is on array declaration, storing applicant data, and using arrays to display data on applicants who passed or failed the SKD test.

Keywords: Array Declaration, Data Management, Information System, Applicant Data



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

PENDAHULUAN

Python memiliki berbagai jenis tipe data bawaan (built in data type). Sebelumnya, kita telah mempelajari terkait basic tipe data, seperti integer, float, boolean, dan string. Selanjutnya, kita juga akan belajar terkait tipe data di Python yang merepresentasikan struktur data array. Di dalam dunia pemrograman, kita mengenal berbagai jenis struktur data, seperti array, hash table, tree, graph, maps, dll. Berbagai jenis struktur data tersebut memiliki kelebihan dan kekurangannya masing-masing. Kami akan fokus membahas tentang data struktur array yang direpresentasikan menjadi sebuah jenis tipe data tertentu di Python. (Dede, 2022) Array merupakan jenis struktur data yang digunakan untuk menyimpan berbagai data dengan tipe yang sama dalam memori dan dapat diakses menggunakan indeks tertentu. Singkatnya, array merupakan kumpulan nilai (collections). Di Python, array diperlakukan sedikit berbeda dibandingkan bahasa pemrograman lain (diatur pada modul khusus array). Akan tetapi, Python memiliki jenis tipe data yang dapat direpresentasikan mirip seperti array, tetapi memiliki karakteristik dan keunikannya tersendiri. (Kurniawan, 2022) Pengelolaan data CPNS merupakan proses yang kompleks yang memerlukan sistematis dan efisien dalam menyimpan dan menampilkan data pelamar. Array merupakan struktur data yang cocok untuk mengelola data homogen seperti data pelamar CPNS. Array dapat diakses secara acak dan memungkinkan penyimpanan data dalam urutan yang berurutan, sehingga memudahkan proses pengelolaan dan pencarian data. Dalam konteks pengelolaan data CPNS, array dapat digunakan untuk menyimpan informasi pelamar seperti nama, alamat, tanggal lahir, dan hasil tes SKD. Dengan

demikian, array dapat membantu meningkatkan efisiensi dan akurasi dalam mengelola data pelamar CPNS (Kurniawan, 2022).

METODE PENELITIAN

Ide Awal dan pengembangan ide

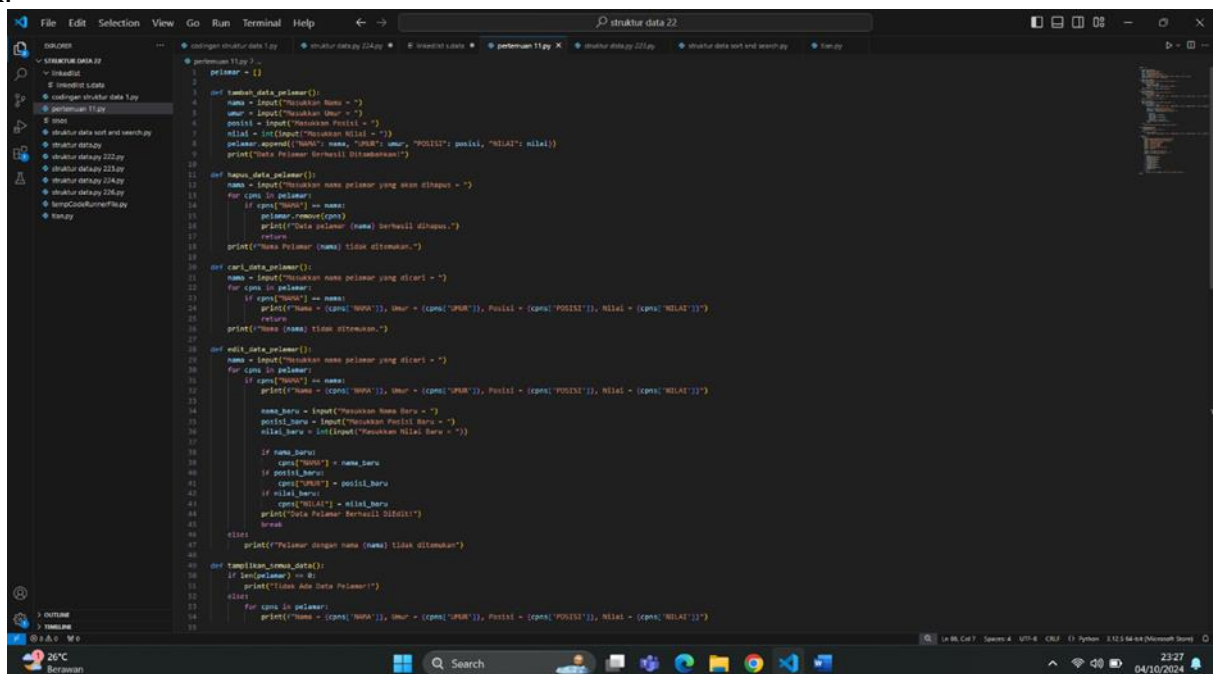
Kami mengembangkan sebuah program python yang menggunakan array untuk menyimpan, mengelola dan menampilkan data pelamar cpns. Program ini akan memungkinkan user dapat menginput data sendiri, mengetahui nilai skd peserta dalam tes tersebut. Lalu dengan ide tersebut dapat kami kembangkan lagi seperti mengklasifikasikan data pelamar misalnya, nama pelamar, umur pelamar, dan posisi yang dilamar. Klasifikasi tersebut dapat diambil berdasarkan data pelamar yang melamar cpns.

Teknik Pengumpulan Data

Untuk mendapatkan data mahasiswa seperti nilai skd, umur, dan posisi yang dilamar, kami berkomunikasi dengan mahasiswa yang pernah ikut tes CPNS untuk pengumpulan data yang akan dianalisis menggunakan program python nantinya. Hasil dari komunikasi yang telah kami diskusikan, mahasiswa yang pernah ikut tes CPNS dapat mengisi data seperti nama, umur, dan posisi yang diambil saat melamar CPNS.

Flowchart

Flowchart merupakan langkah – langkah penyelesaian masalah yang dituliskan dalam simbol-simbol tertentu. Flowchart adalah alur program yang dibuat mulai dari pembuka (start), isi sampai keluar program (output). Adapun untuk alur proses pada flowchart dibawah ini.



```
def tambah_data_pelamar():  
    nama = input("Masukkan Nama : ")  
    umur = input("Masukkan Umur : ")  
    posisi = input("Masukkan Posisi : ")  
    nilai = int(input("Masukkan Nilai : "))  
    pelamar.append({"nama": nama, "umur": umur, "posisi": posisi, "nilai": nilai})  
    print("Data Pelamar Berhasil Ditambahkan")  
  
def hapus_data_pelamar():  
    nama = input("Masukkan nama pelamar yang akan dihapus : ")  
    for cpns in pelamar:  
        if cpns["nama"] == nama:  
            pelamar.remove(cpns)  
            print("Data pelamar (nama) berhasil dihapus.")  
            return  
    print("Nama Pelamar (nama) tidak ditemukan.")  
  
def cari_data_pelamar():  
    nama = input("Masukkan nama pelamar yang dicari : ")  
    for cpns in pelamar:  
        if cpns["nama"] == nama:  
            print("Nama : (cpns['nama']), Umur : (cpns['umur']), Posisi : (cpns['posisi']), Nilai : (cpns['nilai'])")  
            return  
    print("Nama (nama) tidak ditemukan.")  
  
def edit_data_pelamar():  
    nama = input("Masukkan nama pelamar yang dicari : ")  
    for cpns in pelamar:  
        if cpns["nama"] == nama:  
            print("Nama : (cpns['nama']), Umur : (cpns['umur']), Posisi : (cpns['posisi']), Nilai : (cpns['nilai'])")  
            nama_baru = input("Masukkan Nama Baru : ")  
            posisi_baru = input("Masukkan Posisi Baru : ")  
            nilai_baru = int(input("Masukkan Nilai Baru : "))  
            if nama_baru:  
                cpns["nama"] = nama_baru  
            if posisi_baru:  
                cpns["posisi"] = posisi_baru  
            if nilai_baru:  
                cpns["nilai"] = nilai_baru  
            print("Data Pelamar Berhasil Diedit!")  
            break  
    else:  
        print("Pelamar dengan nama (nama) tidak ditemukan")  
  
def tampilkan_data_pelamar():  
    if len(pelamar) == 0:  
        print("Tidak Ada Data Pelamar!")  
    else:  
        for cpns in pelamar:  
            print("Nama : (cpns['nama']), Umur : (cpns['umur']), Posisi : (cpns['posisi']), Nilai : (cpns['nilai'])")
```

Gambar 1.

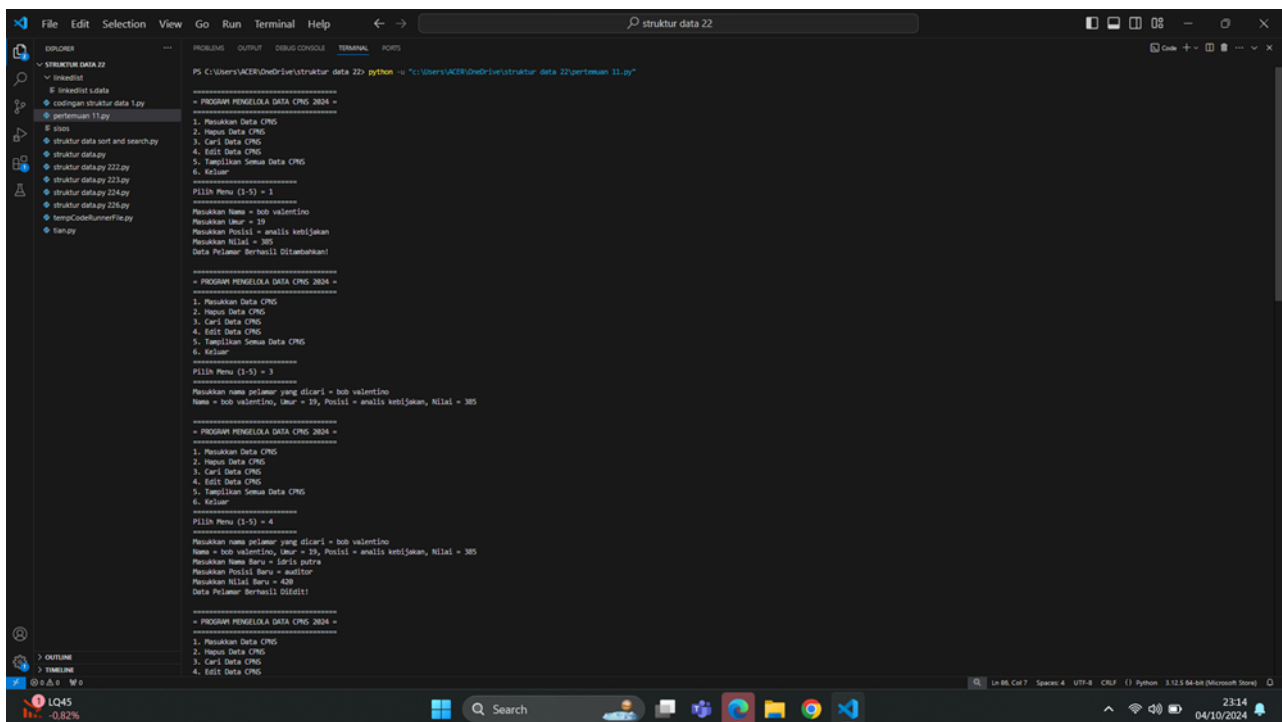
Codingan diatas berisi skrip Python yang tampaknya merupakan aplikasi dasar CRUD (Buat, Baca, Perbarui, Hapus) untuk mengelola daftar pelamar (pelamar). Berikut penjelasan singkat setiap bagian kodenya:

1. `pelamar = []`: Ini menginisialisasi daftar kosong yang disebut pelamar, yang akan menyimpan kamus data pelamar (seperti nama, usia, posisi, dan skor).

2. def tambah_data_pelamar(): Fungsi ini digunakan untuk menambah data pelamar baru. Pengguna diminta untuk memasukkan nama (nama), usia (umur), posisi (posisi), dan skor (nilai) pelamar. Data ditambahkan sebagai kamus ke daftar pelamar dengan kunci 'NAMA', 'UMUR', 'POSISI', dan 'NILAI'.
3. def hapus_data_pelamar(): Fungsi ini digunakan untuk menghapus data pelamar berdasarkan namanya. Program mencari daftar pelamar, dan jika nama ditemukan, pelamar tersebut akan dihapus. Jika pemohon tidak ditemukan, maka akan muncul pesan yang menunjukkan bahwa pemohon tidak ditemukan.
4. def cari_data_pelamar(): Fungsi ini mencari pelamar berdasarkan namanya. Ini menelusuri daftar, dan jika ditemukan kecocokan, ini akan menampilkan rincian pelamar (nama, usia, posisi, dan skor). Jika nama tidak ditemukan, pesan akan dicetak yang menunjukkan hal ini.
5. def edit_data_pelamar(): Fungsi ini memungkinkan pengguna untuk mengedit data pelamar. Pengguna diminta memasukkan nama pemohon yang akan diedit. Jika ditemukan, program akan meminta nilai baru (nama, posisi, dan skor). Nilai-nilai baru ini menggantikan nilai-nilai yang sudah ada. Jika nama tidak ditemukan, maka akan muncul pesan yang menunjukkan bahwa pemohon tidak ditemukan.
6. def tampilkan_semua_data(): Fungsi ini menampilkan seluruh data pelamar dalam daftar pelamar. Jika daftarnya kosong, ia akan mencetak pesan yang menunjukkan bahwa tidak ada data yang tersedia.

HASIL PENELITIAN DAN PEMBAHASAN

Setelah melakukan pengumpulan data lalu kami memasukkan data tersebut kedalam program python yang telah dibuat sebelumnya dengan memasukkan masing-masing data pelamar seperti nama, umur dan posisi yang dilamar. Berdasarkan hasil penelitian yang telah dilakukan, analisis pada implementasi data pelamar CPNS dengan python dapat menunjukkan bahwa data pelamar seperti nama, umur dan juga posisi yang dilamar telah keluar.



```
File Edit Selection View Go Run Terminal Help
struktur data 22
Python 3.12.5 Shell (Microsoft Store)

STRUKTUR DATA 22
  Inkeadlist
  Inkeadlist sdata
  Codingan struktur data 1.py
  peritemen 11.py
  klon
  struktur data sort and search.py
  struktur data.py
  struktur data.py 222.py
  struktur data.py 223.py
  struktur data.py 224.py
  struktur data.py 225.py
  tempCodeRunnerFile.py
  kany

Python 3.12.5 Shell (Microsoft Store)
PS C:\Users\ACER\OneDrive\struktur data 22> python -u "c:\Users\ACER\OneDrive\struktur data 22\peritemen 11.py"

=====
PROGRAM PENGELOLA DATA CPNS 2024 =
1. Masukkan Data CPNS
2. Hapus Data CPNS
3. Cari Data CPNS
4. Edit Data CPNS
5. Tampilkan Semua Data CPNS
6. Keluar

Pilih Menu (1-5) = 1

Masukkan Nama = bob valentino
Masukkan Umur = 19
Masukkan Posisi = analis kebijakan
Masukkan Nilai = 385
Data Pelamar Berhasil Ditambahkan!

=====
PROGRAM PENGELOLA DATA CPNS 2024 =
1. Masukkan Data CPNS
2. Hapus Data CPNS
3. Cari Data CPNS
4. Edit Data CPNS
5. Tampilkan Semua Data CPNS
6. Keluar

Pilih Menu (1-5) = 3

Masukkan nama pelamar yang dicari = bob valentino
Nama = bob valentino, Umur = 19, Posisi = analis kebijakan, Nilai = 385

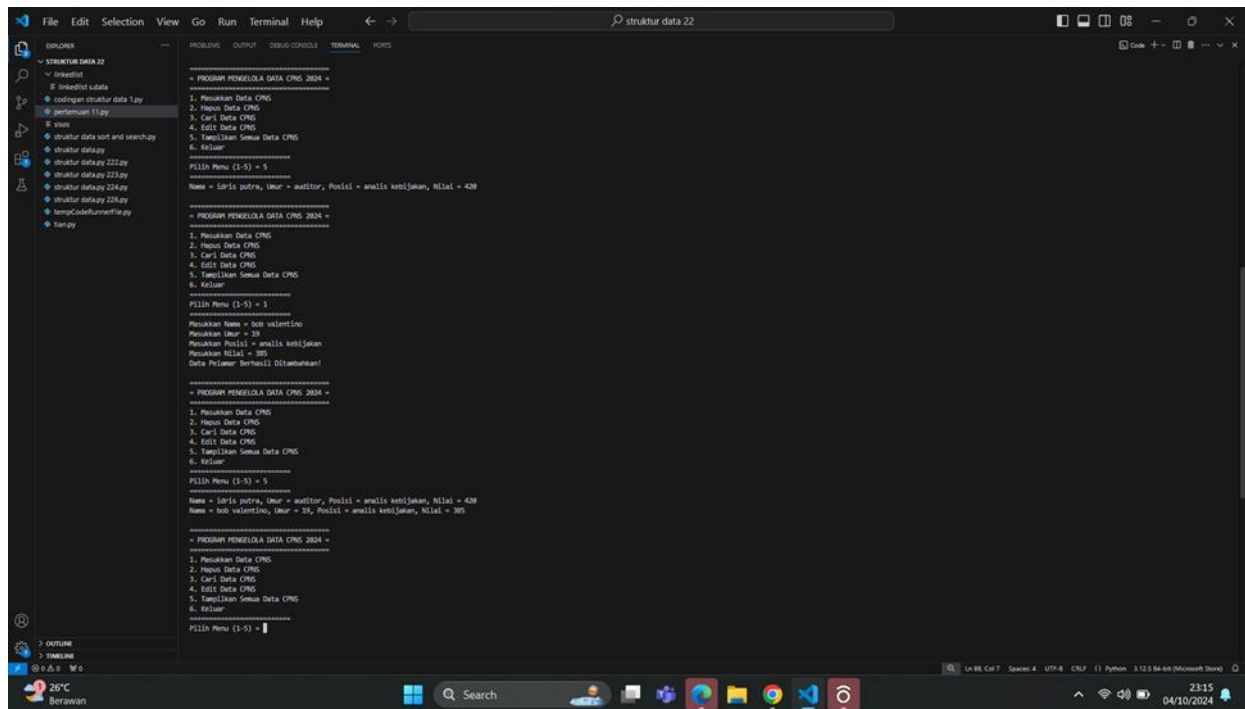
=====
PROGRAM PENGELOLA DATA CPNS 2024 =
1. Masukkan Data CPNS
2. Hapus Data CPNS
3. Cari Data CPNS
4. Edit Data CPNS
5. Tampilkan Semua Data CPNS
6. Keluar

Pilih Menu (1-5) = 4

Masukkan nama pelamar yang dicari = bob valentino
Nama = bob valentino, Umur = 19, Posisi = analis kebijakan, Nilai = 385
Masukkan Nama Baru = laris pifra
Masukkan Posisi Baru = auditor
Masukkan Nilai Baru = 428
Data Pelamar Berhasil Diditit!

=====
PROGRAM PENGELOLA DATA CPNS 2024 =
1. Masukkan Data CPNS
2. Hapus Data CPNS
3. Cari Data CPNS
4. Edit Data CPNS
```

Gambar 2. Tampilan Menu dan Data Mahasiswa



```
File Edit Selection View Go Run Terminal Help
struktur data 22
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
STRUKTUR DATA 22
  linkedlist
  linkedlist.sistem
  codingan struktur data 1.py
  permenum 11.py
  sios
  struktur data sort and search
  struktur data.py
  struktur data.py 222.py
  struktur data.py 224.py
  struktur data.py 225.py
  tempCodeRunnerFile.py
  test.py
=====
PROGRAM PENGELOLA DATA CPNS 2024.
=====
1. Masukkan Data CPNS
2. Hapus Data CPNS
3. Cari Data CPNS
4. Edit Data CPNS
5. Tampilkan Semua Data CPNS
6. Keluar
=====
Pilih Menu (1-5) : 5
=====
Name = Idris putra, Umur = auditor, Posisi = analis kebijakan, Nilai = 420
=====
PROGRAM PENGELOLA DATA CPNS 2024.
=====
1. Masukkan Data CPNS
2. Hapus Data CPNS
3. Cari Data CPNS
4. Edit Data CPNS
5. Tampilkan Semua Data CPNS
6. Keluar
=====
Pilih Menu (1-5) : 1
=====
Masukkan Nama = bob valentino
Masukkan Umur = 19
Masukkan Posisi = analis kebijakan
Masukkan Nilai = 385
Data Pelamar Berhasil Ditambahkan!
=====
PROGRAM PENGELOLA DATA CPNS 2024.
=====
1. Masukkan Data CPNS
2. Hapus Data CPNS
3. Cari Data CPNS
4. Edit Data CPNS
5. Tampilkan Semua Data CPNS
6. Keluar
=====
Pilih Menu (1-5) : 5
=====
Name = Idris putra, Umur = auditor, Posisi = analis kebijakan, Nilai = 420
Name = bob valentino, Umur = 19, Posisi = analis kebijakan, Nilai = 385
=====
PROGRAM PENGELOLA DATA CPNS 2024.
=====
1. Masukkan Data CPNS
2. Hapus Data CPNS
3. Cari Data CPNS
4. Edit Data CPNS
5. Tampilkan Semua Data CPNS
6. Keluar
=====
Pilih Menu (1-5) : 1
```

Gambar 3. Tampilan Data Mahasiswa

Dari data yang kami kumpulkan didapat bahwa 2 data mahasiswa yang kami input telah keluar yang dimana data pelamar atas nama bob valentino, umur 19, posisi analis kebijakan mendapatkan nilai skd sebesar 385, sedangkan data pelamar yang lainnya dengan nama idris putra, umur 19, posisi auditor, mendapatkan nilai skd sebesar 420. Dalam konteks data pelamar, array dapat digunakan untuk menyimpan atribut seperti nama, usia, dan nilai ujian, memungkinkan pencarian dan pengurutan data dengan algoritma sederhana seperti bubble sort atau quick sort¹⁴. Dengan struktur data ini, proses pengolahan dan analisis data pelamar menjadi lebih cepat dan terorganisir, mendukung keputusan seleksi yang lebih baik. (Kurniawan, 2022)

KESIMPULAN

Penerapan array dalam mengelola data CPNS (Calon Pegawai Negeri Sipil) terbukti sangat efektif dalam menyimpan dan mengolah informasi pelamar. Dengan menggunakan array, data dapat diorganisir secara sistematis, memungkinkan akses yang cepat dan efisien terhadap informasi penting seperti nama, usia, dan nilai ujian pelamar. Penggunaan array juga memfasilitasi proses pencarian, pengurutan, dan pemfilteran data, yang sangat penting dalam tahap seleksi. Selain itu, struktur data ini mendukung pengembangan aplikasi yang lebih kompleks untuk manajemen data pelamar di masa depan. Secara keseluruhan, penerapan array tidak hanya meningkatkan efisiensi pengelolaan data, tetapi juga membantu dalam pengambilan keputusan yang lebih baik dalam proses rekrutmen CPNS.

DAFTAR PUSTAKA

- Dede, K. (2022) Penerapan Sequeintal Search Untuk Pengelolaan Data. *Antivirus : Jurnal Ilmiah Teknik Informatika*, 15(1), 86 - 97
- Kurniawan, K. A. (2022). Penerapan Struktur Data Tree menggunakan Array Pada Bahasa Java menunjang Logika Cerdas. *Surakarta Informatic Journal*, 2022, 5.2: 27 – 34.