

TUGAS DATA STRUCTURE

COLLECTION, ARRAYS, AND LINKED STRUCTURE



Disusun Oleh :

Ricky Yohanes Wijaya (31120013)

Teknik Informatika (Kelas Malam)

**UNIVERSITAS WIDYA KARTIKA : University of Business Excellence for
Entrepreneurship in Indonesia**

Jl. Sutorejo Prima Utara II/1 Surabaya, Jawa Timur, Indonesia

Phone: +62 31 5922403 / 5926359, Fax: +62 31 5925790 | Email: info@widyakartika.ac.id

KATA PENGANTAR

Puji dan syukur kehadiran Tuhan Yang Maha Esa, atas kasih dan karunia-Nya, sehingga penulis dapat menyelesaikan penulisan tugas paper ini dengan baik. Paper ini disusun untuk memenuhi tugas mata kuliah Data Structure. Adapun judul paper ini "Collection, Arrays, and Linked Structure".

Penulis menyadari sepenuhnya bahwa penulisan tugas ini masih jauh dari kata sempurna, karena keterbatasan pengetahuan, pengalaman serta referensi yang penulis miliki. Untuk itu penulis mengharapkan saran dan kritik demi perbaikan pada masa-masa yang akan mendatang.

Surabaya, 28 April 2021

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN	
A. Latar Belakang	1
B. Rumusan Masalah.....	2
C. Tujuan	2
BAB II PEMBAHASAN	
A. Data Collection	3
B. Arrays.....	8
C. Linked List.....	11
BAB III PENUTUP	
A. Kesimpulan	14
B. Saran	15
DAFTAR PUSTAKA	16

BAB I

PENDAHULUAN

A. Latar Belakang

Dalam ilmu komputer , struktur data adalah suatu cara penyimpanan, penyusunan, dan pengaturan data dalam media penyimpanan computer sehingga data tersebut dapat digunakan secara efisien.(Ichan Taufik, 2017)

Struktur Data juga mengatur semua item data dengan tidak hanya mempertimbangkan elemen yang disimpan tetapi juga hubungan antar data satu sama lain.(Dr. Pradyumansinh Jadeja, 2015).

Secara mudah , suatu “Algoritma” + “Data Structure” = Program / Application.

Data sendiri berarti representasi (perwujudan) dari fakta di dunia nyata, dan fakta tersebut disimpan, direkam , atau direpresentasikan dalam bentuk tulisan, suara, gambar, sinyal, kode biner (bit), symbol, ataupun format – format lain.

Ada beberapa istilah dalam data dan struktur data , antara lain :

1. Tipe data :

Jenis / macam data di dalam suatu variable dalam bahasa pemrograman misalnya Integer , Float, Character (Sering disebut tipe data primitive) dan ada juga Array, List , File (Tipe data non-primitive)

2. Object :

Kumpulan elemen / domain.

3. Representasi data :

Wujud dari data tersebut , misal Boolean di representasikan dalam 0 dan 1 atau True and False.

4. Struktur data :

pengelompokan beberapa informasi yang terkait menjadi sebuah kesatuan.

Dalam Teknik pemrograman, Struktur data berarti tata letak data yang berisi kolom – kolom data , baik yang tampak oleh user ataupun yang hanya untuk keperluan pemrograman.(Bullinaria, 2019)

Kolom – kolom tersebut diistilahkan dengan record (catatan). Dengan sifatnya ini, Structure data sering digunakan untuk keperluan database misalnya , database keuangan, pengolah kata, spreadsheet, dan masih banyak lagi.

Tipe data dapat dikelompokkan menjadi 2 yaitu :

1. Simple data

- Data Tunggal : Integer , Boolean, Character, etc
- Data Majemuk : String

2. Structured data

- Data Tunggal : Array , Record.
- Data Majemuk :
 - Linear : Stack, Queue, List, Multilist
 - Non – linear : Binary Tree, Graph.

Pemakaian Struktur data yang tepat dalam proses pemrograman akan menghasilkan algoritma yang lebih jelas dan tepat , sehingga keseluruhan program menjadi lebih efisien dan sederhana.(Nisa'ul Hafidhoh, 2007)

Struktur data yang akan di bahas di paper ini adalah :

1. Collection

2. Arrays

3. Linked List

B. Rumusan Masalah

Rumusan masalah yang akan dibahas di paper ini adalah :

1. Apa itu Collection?
2. Apa itu Arrays?
3. Apa itu Linked Structure ?
4. Bagaimana contoh implementasi dari Collection, Arrays, dan Linked Structure?

C. Tujuan

1. Dapat memahami definisi dari Collection, Arrays, dan Linked Structure.
2. Dapat memahami struktur data dari Collection, Arrays, dan Linked Structure.
3. Dapat mengimplementasikan struktur data Collection, Arrays, dan Linked Structure.

BAB II

PEMBAHASAN

A. COLLECTION

Dalam Bahasa Pemrograman Python dikenal 4 tipe data collection, yaitu List, Tuple, Set, dan Dictionary.

- **List** dibuat dengan penanda kurung siku “[]”.
- **Tuple** dibuat dengan penanda kurung biasa“()” , bisa juga tanpa tanda kurung , tapi ini tidak direkomendasikan karena dapat menyebabkan kebingungan.
- **Set** dibuat dengan penanda kurung kurawal “{ }”.
- **Dictionary** dibuat dengan penanda kurung siku “[]”, tetapi dengan isi yang berbeda yang akan dibahas di segmennya nanti.

List adalah kumpulan atau urutan satu atau lebih item data. List adalah salah satu tipe data yang **paling banyak digunakan di Python dan sangat fleksibel**. Semua item dalam daftar tidak harus berjenis sama dan boleh bernilai sama (Allow duplicate). (Drake, 2014).

Contoh Implementasi :

```
list1 = ["apple", "banana", "cherry"]
list2 = ["abc", 34, True, 40, "male"]
# mengakses element dalam list
print(list1[0])
print(list2[3])
```

List memiliki beberapa built-in method dari python (sama dengan array) :

Method	Deskripsi
append()	Menambah element baru diakhir array
clear()	Menghapus semua element dari array
copy()	Menduplikat array
count()	Menghitung jumlah element dalam array
extend()	Menambahkan per element (iterable), di akhir array.
index()	Mengembalikan nilai index suatu element

insert()	Menyisipkan element di index spesifik
pop()	Menghapus element akhir atau index tertentu dari array
remove()	Menghapus element spesifik dari suatu array
reverse()	Membalik urutan array
sort()	Mengurutkan array

Tuple adalah tipe data yang sangat mirip dengan list , namun **sifatnya immutable (Unchangeable)**, sangat cocok untuk pemrograman fungsional – stateless.(Journaldev, 2020)

Contoh implementasi:

```

tuple1 = ("apple", "banana",
"cherry")
tuple2 = (1, 5, 7, 9, 3)
tuple3 = ("abc", 34, True, 40,
"male")
# mengakses element dalam tuple
print(tuple1[0])
print(tuple2[3])
print(tuple3[4])
tuple1.count()
tuple1.index(1)

```

Karena tidak banyak yang bisa dilakukan untuk mengedit Tuple, jadi Tuple hanya memiliki dua built-in fuction dari python:

Method	Deskripsi
count()	Menghitung jumlah element dalam tuple
index()	Mencari nilai spesifik dari element dalam tuple dan mengembalikan nilai berupa posisi dimana element tersebut ditemukan.

Set adalah kumpulan satu atau lebih item data yang **tidak punya urutan** . Data dalam set dapat ditambahkan atau dihapus, tapi tidak memungkinkan untuk dirubah atau melakukan slicing karena set **tidak punya indeks**. Untuk mengaksesnya membutuhkan method khusus.(Journaldev, 2020)

Contoh implementasi :

```
set1 = {"abc", 34, True, 40, "male"}

#Menambah elemen pada set

set1.add('name')

set1.add('age')
```

Set memiliki beberapa built-in function dari python :

Method	Deskripsi
add()	Menambah element dalam set
clear()	Menghapus semua element dari set
copy()	Membuat duplikat dari set
difference()	Mencari perbedaan antara dua set.
difference_update()	Menghapus perbedaan dari dua set.
discard()	Menghapus element spesifik dari set
intersection()	Mencari irisan / element yang sama dari dua set
intersection_update()	Menghapus irisan dari dua set
isdisjoint()	Mengecek apakah dua set memiliki irisan atau tidak.
issubset()	Mengecek apakah set lain memiliki semua element dari set ini.
issuperset()	Mengecek apakah set ini memiliki semua element dari set lain.
pop()	Menghapus suatu element dari set.(Biasanya di akhir set)
remove()	Menghapus element spesifik dari set
symmetric_difference()	Mencari element yang berbeda dari dua set
symmetric_difference_update()	Menyisipkan element yang berbeda dari dua set atau yang lain.
union()	Menggabungkan set

Dictionary adalah kumpulan satu atau lebih **pasangan keys dan value** dimana keysnya adalah suatu object jenis apapun yang dapat di-hash dan Valuenya referensi objek yang bisa bernilai apapun. Sama seperti Set, Dictionary juga dapat ditambahkan dan dihapus tapi tidak memungkinkan untuk melakukan slicing karena tidak memiliki indeks.

Pengaksesannya pun harus menggunakan keys nya.(Journaldev, 2020; Refsnes Data, 2020)

Contoh implementasi dictionary :

```
# Mendeklarasikan Dictionary

dict = {

    "id": 1948,

    "name": "Washer",

    "size": 3

}

# Mengakses keys Dictionary

print(dict[name])

dict.keys()
```

Dictionary memiliki beberapa built-in method dari python :

Method	Deskripsi
clear()	Menghapus semua element dalam dictionary
copy()	Menduplikat dictionary
fromkeys()	Menampilkan dictionary dengan keys dan value spesifik.
get()	Menampilkan value dari key spesifik
items()	Menampilkan list yang berisi pasangan keys dan values.
keys()	Menampilkan semua keys yang terdapat dalam dictionary
pop()	Menghapus element dengan key spesifik.
popitem()	Menghapus pasangan key dan value yang terakhir dimasukkan
setdefault()	Menampilkan value dari key spesifik, jika key tersebut tidak ditemukan , inisiasi key tersebut dan isi dengan value yang didefinisikan.
update()	Updates dictionary dengan pasangan key dan value spesifik
values()	Menampilkan semua value dari dictionary

Dari penjelasan dan method diatas , bisa diketahui sifat dan perbedaan masing – masing data collection. Beberapa sifat dan perbedaan dari tiap data collection dijelaskan di tabel berikut :

Sifat	List	Tuple	Set	Dictionary
Pendefinisian	Kurung Siku “[]”	Kurung Biasa “()”	Kurung Kurawal “{}”	Kurung Siku “{}”
Metode Pengaksesan	Diakses menggunakan indeks	Diakses menggunakan indeks	Diakses dengan method khusus karena set tidak memiliki indeks (sifatnya unordered)	Diakses menggunakan Keys
Keseragaman data	Dapat menampung berbagai tipe data.	Dapat menampung berbagai tipe data.	Dapat menampung berbagai tipe data , tetapi	Dapat menampung berbagai tipe data , dan
Allow Duplicates	Memperbolehkan data duplikasi	Memperbolehkan data duplikasi	Tidak memperbolehkan data duplikasi	Memperbolehkan data duplikasi pada valuenya saja . bukan pada keysnya
Mutable/ Imutable	Mutable	Imutable (Unchangeable)	Imutable (Unchangeable)	Mutable
Penggunaan	Imperative Programming (OOP) - Stateful	Fungsional Programming - Stateless	Fungsional Programming - Stateless	Imperative Programming (OOP) - Stateful

Source : (Drake, 2014; Python.org, n.d.)

Terlepas dari semua kelebihan dan kekurangannya , setiap tipe data memiliki kegunaannya masing – masing, seperti bahasa pemrograman , tidak ada tipe data yang 100% baik dan tepat untuk segala kondisi.(Syuhada, 2018)

Programmer diharapkan bisa menentukan sendiri data collection mana yang paling cocok digunakan untuk pembuatan programnya

B. ARRAYS

Array atau dalam Bahasa Indonesia sering disebut larik adalah suatu kumpulan data terstruktur yang berupa sejumlah data sejenis (memiliki jenis data yang sama) yang jumlahnya tetap dan diberi suatu nama tertentu.(Sihombing, 2020)

Sebenarnya cara secara konsep dan penggunaan semuanya sama dengan list , bahkan list dinilai lebih fleksibel daripada arrays. Tapi tidak ada salahnya kita mempelajari tentang array, karena kebanyakan Bahasa pemrograman yang lain tidak memiliki tipe data list seperti python, hanya array yang dikenali (Rismayani & Ardimansyah, 2015) Sama seperti list dan tipe data collection yang lain , Arrays dimulai dari indeks ke 0.

Dan ciri – ciri arrays yaitu : (FACEPrep.in, 2020; Umum, 2016)

1. Hanya dapat menampung satu jenis tipe data (sama).
2. Array di Bahasa pemrograman lain (selain python) harus didefinisikan dulu ukurannya karena perlu pengalokasian dulu di memory, jika array sudah penuh maka perlu inisiasi array baru dengan ukuran yang lebih besar.

Untuk cara pengaksesannya bisa menggunakan dua cara , mirip seperti list : (menggunakan Bahasa pemrograman python) :

1. Mengakses indeks nya secara langsung.

Contoh :

```
From array import *  
number = array ("i",[1,2,3,4,5])  
number[0] = 6  
number[1] = 7
```

2. Method khusus array / list.

Contoh:”

```
From array import *  
number = array ("i",[1,2,3,4,5])  
  
#menambah isi array  
number.append(8)  
  
#menyisipkan nilai dalam indeks tertentu array  
number.insert(2,9)  
  
#menghapus value tertentu dalam array  
Number.remove(5)
```

Beberapa built-in method untuk mengelola array dan list di python :

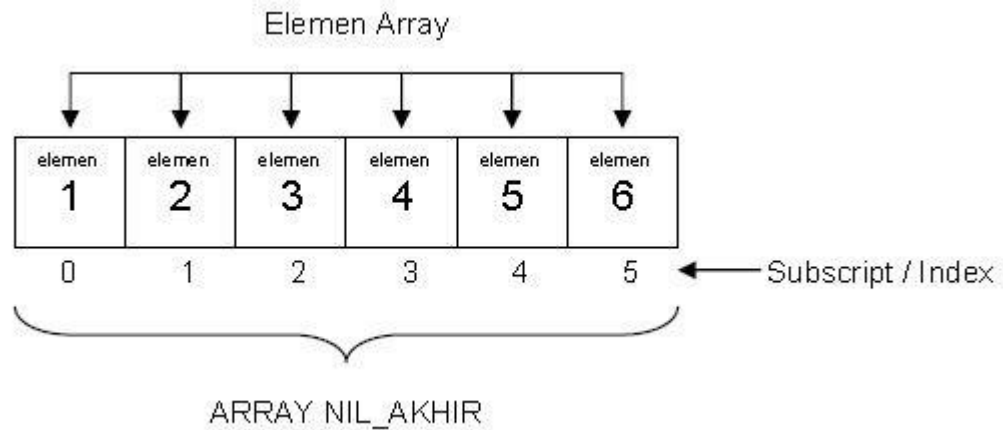
Method	Description
append()	Menambah element baru diakhir array
clear()	Menghapus semua element dari array
copy()	Menduplikat array
count()	Menghitung jumlah element dalam array
extend()	Menambahkan per element (iterable), di akhir array.
index()	Mengembalikan nilai index suatu element
insert()	Menyisipkan element di index spesifik
pop()	Menghapus element akhir atau index tertentu dari array
remove()	Menghapus element spesifik dari suatu array
reverse()	Membalik urutan array
sort()	Mengurutkan array

Perlu diperhatikan, di python , saat ingin menggunakan array perlu diimport dulu library nya karena array bukan struktur data bawaan dari Bahasa python

Method untuk array dan list di python sama , karena python mendefinisikan array juga setingkat list.

Array terbagi jadi array satu dimensi dan array multidimensi.

- Array berdimensi satu memiliki elemen berupa item data individual yang bisa diakses lewat indeksnya.



Source : (Hambali, 2020)

- Array multidimensi sering dianalogikan sebagai matriks karena memiliki baris dan kolom , singkatnya kita memiliki data array dalam array.

`Int A [3][4];`

Materi Dosen	Kolom 0	Kolom 1	Kolom 2	Kolom 3
Baris 0	A[0][0]	A[0][1]	A[0][2]	A[0][3]
Baris 1	A[1][0]	A[1][1]	A[1][2]	A[1][3]
Baris 2	A[2][0]	A[2][1]	A[2][2]	A[2][3]

Source : (MateriDosen, 2020)

C. LINKED LIST

Linked List adalah Salah satu bentuk struktur data yang berisi kumpulan data yang tersusun secara sekuensial, saling bersambungan, dinamis dan terbatas adalah senarai berkait (linked list). (Romanides, 1956)

Suatu senarai berkait (linked list) adalah suatu simpul (node) yang dikaitkan dengan simpul yang lain dalam suatu urutan tertentu. Suatu simpul dapat berbentuk suatu struktur atau class.(Umum, 2016)

Setiap node memiliki data dan referensi ke node selanjutnya (singly) dan atau ke node sebelumnya (doubly). Digunakan sebagai dasar implementasi Stacks dan Queue. Tidak ada unsur Random Access via indeks seperti Array dan setiap operasi biasanya perlu operasi sekuensial dari node awal.

Linked List bersifat dinamis secara ukuran, Alokasi penggunaan memory yang dibutuhkan pada run-time, Mudah di implementasikan.

Tetapi kekurangannya , Linked List lebih boros memory, pembacaan node hanya bisa melalui proses sekuensial dari awal, tidak seperti indeks di Array. Pada Singly Linked List, mustahil bisa melakukan traversal dari belakang ke depan, namun dapat disolusikan dengan Doubly Linked List.(Prasetio, 2018)

Di dalam Linked List pun ada beberapa Operasi yang perlu diketahui yaitu:

- Insert

Istilah Insert berarti menambahkan sebuah node baru ke dalam suatu linked list.

- IsEmpty

Fungsi ini menentukan apakah linked list kosong atau tidak.

- Find First

Fungsi ini mencari elemen pertama dari linked list.

- Find Next

Fungsi ini mencari elemen sesudah elemen yang ditunjuk now.

- Retrieve

Fungsi ini mengambil elemen yang ditunjuk oleh now. Elemen tersebut lalu dikembalikan oleh fungsi.

- Update

Fungsi ini mengubah elemen yang ditunjuk oleh now dengan isi dari sesuatu.

- Delete Now

Fungsi ini menghapus elemen yang ditunjuk oleh now. Jika yang dihapus adalah elemen pertama dari linked list (head), head akan berpindah ke elemen berikutnya.

- Delete Head

Fungsi ini menghapus elemen yang ditunjuk head. Head berpindah ke elemen sesudahnya.

- Clear

Fungsi ini menghapus linked list yang sudah ada. Fungsi ini wajib dilakukan bila anda ingin mengakhiri program yang menggunakan linked list. Jika anda melakukannya, data-data yang dialokasikan ke memori pada program sebelumnya akan tetap tertinggal di dalam memori

(Umum, 2015)

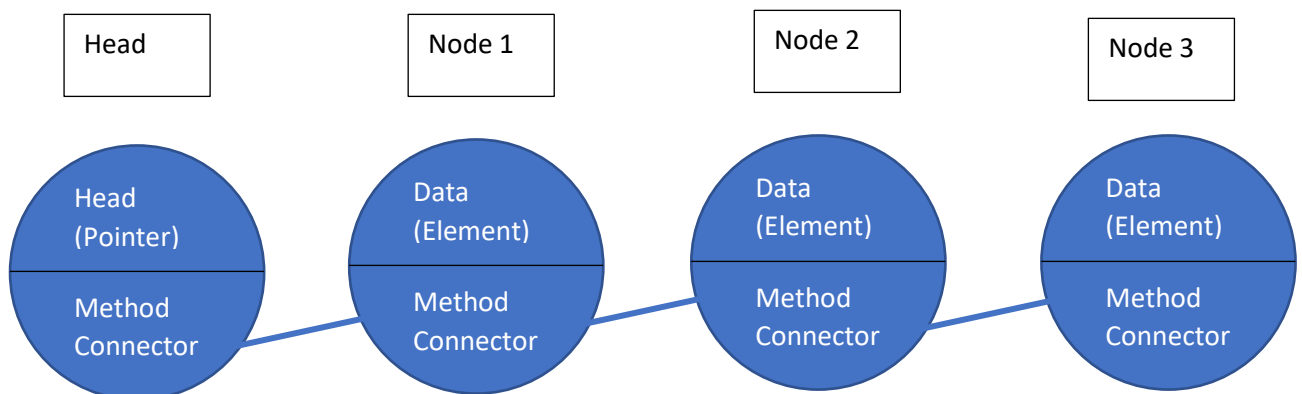
Konsep single linked list bisa dianalogikan sebagai kereta api. Head / pointer menunjuk pada lokomotifnya dan tiap gerbong dibelakangnya sebagai node – nodenya.

Kemudian tiap gerbongnya tentu memiliki isi sendiri – sendiri , baik penumpang, bahan bakar, logistic dan sebagainya. Ini dikategorikan sebagai data

Agar tiap gerbong dapat berhubungan satu sama lainnya , diperlukan penghubung (connector) yang berupa method penghubung.(Bullinaria, 2019)

Tiap node terbagi jadi 2 bagian yaitu element (data) dan link (connector)

Ilustrasi Linked List :



Berikut contoh source code untuk implementasi single linked list :

```
def main ():  
    myList = SinglyLinkedList()  
    myList.add_first(10)  
    myList.add_first(10)  
    myList.add_first(10)  
    print(str(myList))  
    print(len(myList))  
  
class SinglyLinkedList ():  
    class _Node:  
        def __init__(self, element, nextNode = None):  
            self.element = element  
            self.nextNode = nextNode  
    def __init__(self):  
        self._head = None  
        self._size = 0  
    def __str__(self):  
        result = ""  
        pointer = self._head  
        while pointer != None :  
            result = result + str(pointer.element) + " "  
            pointer = pointer.nextNode  
        return result  
    def add_first(self, element):  
        newNode = self._Node(element)  
        newNode.nextNode = self._head  
        self._head = newNode  
        self._size += 1  
    def __len__(self):  
        return self._size
```


BAB III

PENUTUP

A. KESIMPULAN

Struktur data adalah cara penyimpanan, penyusunan dan pengaturan data di dalam media penyimpanan komputer sehingga data tersebut dapat digunakan secara efisien.

Beberapa tipe data yang sering digunakan dalam penyusunan data tersebut adalah collection data type , yaitu List, Tuple, Set, Dictionary, Arrays, dan Linked List.(Dr. Pradyumansinh Jadeja, 2015; Ichan Taufik, 2017)

List paling sering / umum digunakan di Bahasa Python karena efisien dan mudah digunakan.

Set dan Tuple cocok untuk pemrograman fungsional.

Dictionary cocok untuk data yang membutuhkan kata kunci tertentu dan penanda / simbol tertentu.

Untuk arrays sendiri adalah tipe data khusus di python yang perlu diimport dulu , tetapi dalam penggunaannya sama seperti list. Array hanya bisa menampung satu jenis tipe data dengan Panjang yang didefinisikan (di Bahasa pemrograman lain)

Sedangkan linked list , seperti namanya adalah list yang saling terhubung.

Linked List adalah salah satu bentuk struktur data, berisi kumpulan data (node) yang tersusun secara sekuensial, saling sambung-menyambung, dinamis dan terbatas.(Romanides, 1956)

Konsepnya sering digunakan untuk Object Oriented Programming.(McMillan, 2007)

Tidak ada tipe data yang lebih bagus satu dengan yang lainnya, semuanya memiliki fungsi masing – masing yang disesuaikan dengan kebutuhan pembuatan program.

Sebagai programmer , harus memahami baik – baik konsep dan kegunaan masing – masing tipe data , dan tipe data mana yang paling efisien untuk digunakan dalam programnya

B. SARAN

Adapun saran-saran yang dapat disampaikan adalah :

1. Pada tiap kasus yang memerlukan data collection, arrays ataupun linked list, buatlah algoritma dan deklarasi struktur data yang efektif, pertimbangkan waktu proses dan alokasi memory yang dibutuhkan. Semakin kecil memory yang dibutuhkan , semakin cepat waktu proses source code tersebut , dan tentunya semakin efektif algoritma tersebut.
2. Memahami lebih detail lagi fungsi dan cara kerja tiap tipe data dan strukturnya.
3. Menggunakan software yang sesuai untuk keperluannya dan lebih canggih dalam pengimplementasian di masa mendatang.

DAFTAR PUSTAKA

- Bullinaria, J. (2019). Lecture Notes for Data Structures and Algorithms Revised each year, (March), 9–29.
- Dr. Pradyumansinh Jadeja. (2015). Introduction to Data Structure & Case Study : Dr . Appointment, (9879461848).
- Drake, J. (2014). Python Lists, 0–3. Retrieved April 28, 2021, from https://www.w3schools.com/python/python_lists.asp
- FACEPrep.in. (2020). Difference between Linked List and Arrays. Retrieved April 28, 2021, from <https://www.faceprep.in/data-structures/linked-list-vs-array/>
- HAMBALI, Y. (2020). ARRAY. Retrieved April 29, 2021, from <https://sites.google.com/site/arrayyusuphambaliif15d1053/>
- Ichan Taufik, S. M. (2017). MakalahStrukturDataKel4_docx.
- Journaldev. (2020). Python Data Types (With Complete List) - JournalDev. Retrieved April 28, 2021, from <https://www.journaldev.com/14036/python-data-types>
- MateriDosen. (2020). Array Dua Dimensi C++, Lengkap Contoh Program dan Penjelasan - Materi Dosen. Retrieved April 29, 2021, from <http://www.materidosen.com/2017/06/array-dua-dimensi-c-lengkap-contoh.html>
- McMillan, M. (2007). Data Structures and Algorithms Using C#. *Data Structures and Algorithms Using C#*.
- Nisa'ul Hafidhoh. (2007). Pendahuluan Struktur Data Tujuan.
- Prasetio, F. (2018). Struktur Data. Struktur data adalah cara penyimpanan... | by Fahmi Prasetio | Medium. Retrieved April 28, 2021, from <https://medium.com/@fahmiprasetiio/struktur-data-2de34750df89>
- Python.org. (n.d.). Collections - Container Datatypes. *Python.org*. Retrieved April 28, 2021, from <https://docs.python.org/id/3.8/library/collections.html?highlight=collection#module-collections>

- Refsnes Data. (2020). Python Data Types. *W3Schools*. Retrieved April 28, 2021, from <https://www.programiz.com/python-programming/variables-datatypes>
- Rismayani, & Ardimansyah. (2015). Mobile-Based Public Transportation Search Application for Makassar City Using Depth First Search Algorithm. *Jurnal Pekommas*, 18(3), 171–180.
- Romanides. (1956). No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title, 147004, 1–21.
- Sihombing, J. (2020). Penerapan Stack Dan Queue Pada Array Dan Linked List Dalam Java, (June), 15–24.
- Syuhada, M. Y. (2018). JOURNAL OF INFORMATICS AND Perbandingan Algoritma Greedy Search dan Algoritma Depth-First- Search pada Pencarian Kota dengan Graph Romania Problem Comparison of Greedy Search Algorithm and Depth-First-Search Algorithm on City Search with Romanian Graph Pr, 1(2), 58–60.
- Umum, T. I. (2015). Modul Praktikum Algoritma dan Struktur Data MODUL 6 SINGLE & DOUBLE LINKED LIST Modul Praktikum Algoritma dan Struktur Data, 1–23.
- Umum, T. I. (2016). Modul 6 Single & Double Linked List.