# LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN KE – 1

# ARRAY DALAM PYTHON



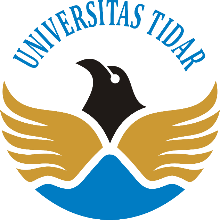
## DISUSUN OLEH :

Zidhan Arzaa Karim 23040506063

## JURUSAN TEKNOLOGI INORMASI FAKULTAS TEKNIK UNIVERSITAS TIDAR

**2023**

# PYTHON CLASSES AND OBJECTS



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Diisi Mahsiswa Praktikan** | | | | | | | | |
| Nama Praktikan | Zidhan Arzaaq Karim | | | | | | | |
| NPM | 2340506063 | | | | | | | |
| Rombel | Rombel 4 | | | | | | | |
| Judul Praktikum | Array dalam python | | | | | | | |
| Tanggal Praktikum | 6 Februari 2024 | | | | | | | |
| **Diisi Asisten Praktikum** | | | | | | | | |
| Tanggal Pengumpulan |  |  |  |  |  |  |  |  |
| Catatan |  | | | | | | | |

|  |  |  |
| --- | --- | --- |
| PENGESAHAN | | NILAI |
| Diperiksa oleh : | Disahkan oleh : |  |
| Asisten Praktikum | Dosen Pengampu |
|  |  |
| (Kurnadi) | (Imam Adinata) |

## PROGRAM STUDI S1 TEKNOLOGI INFORMASI FAKULTAS TEKNIK UNIVERSITAS TIDAR

**2023**

## Tujuan Praktikum

1. Memahami Konsep Array:
   * Memahami konsep dasar array sebagai struktur data yang menyimpan elemen-elemen dengan tipe data yang sama.
   * Mengetahui cara membuat array dan mengakses elemennya.
2. Operasi Dasar pada Array:
   * Melakukan operasi dasar seperti penambahan, pengurangan, perkalian, dan pembagian elemen array.
   * Menerapkan operasi perbandingan pada elemen array.
3. Pengindeksan dan Slicing:
   * Memahami konsep pengindeksan dan slicing untuk mengakses sebagian atau elemen tertentu dari array.
   * Praktek menggunakan pengindeksan dan slicing dalam konteks array.
4. Manipulasi Data dengan Array:
   * Menerapkan manipulasi data, seperti pengurutan dan pencarian, menggunakan array.
   * Menggunakan metode atau fungsi built-in untuk mempermudah manipulasi data.
5. Penggunaan Array dalam Konteks Aplikasi Spesifik:
   * Menggunakan array untuk memodelkan dan menyelesaikan masalah dalam konteks aplikasi spesifik.
6. Pemahaman Konsep Memori:
   * Memahami bagaimana elemen-elemen array disimpan dalam memori komputer.
   * Menjelaskan pengaruh ukuran array terhadap penggunaan memori.
7. Optimisasi Kode:
   * Mengoptimalkan penggunaan array untuk meningkatkan efisiensi dan kinerja kode.
   * Menerapkan strategi pengoptimalan, seperti penggunaan algoritma yang efisien.
8. Pengenalan Konsep List Comprehension:
   * Memahami konsep list comprehension untuk membuat array dengan sintaks yang lebih singkat dan ekspresif.

## Dasar Teori

Array adalah struktur data yang merupakan kumpulan data dengan tipe data yang sama. Saat mendeklarasikan sebuah array, komputer akan memberikan slot lokasi memori yang berurutan atau berdampingan antar elemen satu dengan yang lainnya. Sehingga jumlah slot memori atau elemen array bersifat tetap sesuai jumlah saat pendeklarasian. Array juga dapat diartikan sebagai sekumpulan pasangan indeks dan nilai. Nilai dalam elemen array memiliki tipe data yang sama. Indeks array merepresentasikan alamat memori elemen array. Indeks elemen array dimulai dari 0 (nol).

1.1 Membuat Array

Array dalam Python dapat dibuat dengan mengimpor modul array. Fungsi array (data\_type, value\_list) digunakan untuk membuat array dengan menentukan tipe data dan daftar nilai pada argumennya.

1.2 Menambahkan Elemen ke Array

Elemen dapat ditambahkan ke dalam Array dengan menggunakan fungsi bawaan insert(). Insert digunakan untuk memasukkan satu atau lebih elemen data ke dalam array. Berdasarkan kebutuhan, elemen baru dapat ditambahkan di awal, akhir, atau indeks tertentu dari array. append() juga digunakan untuk menambahkan nilai yang disebutkan dalam argumennya di akhir array.

1.3 Mengakses Elemen dari Array

Untuk mengakses elemen-elemen array, gunakan nomor indeks. Gunakan operator indeks [ ] untuk mengakses elemen dalam array. Indeks harus berupa bilangan bulat.

1.4 Menghapus Elemen Array

Elemen-elemen dapat dihapus dari array dengan menggunakan fungsi bawaan remove(), tetapi muncul kesalahan jika elemen tidak ada dalam set. Metode remove() hanya menghapus satu elemen pada satu waktu, untuk menghapus rentang elemen, iterator digunakan. Fungsi pop() juga dapat digunakan untuk menghapus dan mengembalikan elemen dari array, tetapi secara default hanya menghapus elemen terakhir dari array. Untuk menghapus elemen dari posisi tertentu dalam array, indeks elemen tersebut diberikan sebagai argumen untuk metode pop(). Metode remove pada List hanya akan menghapus kemunculan pertama elemen yang dicari.

1.4 Menghapus Elemen Array

Elemen-elemen dapat dihapus dari array dengan menggunakan fungsi bawaan remove(), tetapi muncul kesalahan jika elemen tidak ada dalam set. Metode remove() hanya menghapus satu elemen pada satu waktu, untuk menghapus rentang elemen, iterator digunakan. Fungsi pop() juga dapat digunakan untuk menghapus dan mengembalikan elemen dari array, tetapi secara default hanya menghapus elemen terakhir dari array. Untuk menghapus elemen dari posisi tertentu dalam array, indeks elemen tersebut diberikan sebagai argumen untuk metode pop(). Metode remove pada List hanya akan menghapus kemunculan pertama elemen yang dicari.

1.6 Mengubah Elemen dalam Array

Untuk mengubah elemen dalam array, kita hanya perlu menetapkan nilai baru ke indeks yang diinginkan untuk diubah.

1.7 Operasi pada Array Python

a. Menghitung Elemen dalam Array

Untuk menghitung elemen dalam sebuah array, kita perlu menggunakan metode

count.

1.8 Membalik Elemen dalam Array

Untuk membalik elemen-elemen dalam sebuah array, kita hanya perlu menggunakan metode reverse.

1.9 Extending Elemen dalam Array

Dalam Python, array digunakan untuk menyimpan beberapa nilai atau elemen dengan tipe data yang sama dalam satu variabel. Fungsi extend() digunakan untuk menambahkan item dari iterable ke bagian akhir array. Dengan kata lain, metode ini digunakan untuk menambahkan array nilai ke bagian akhir array yang ada atau yang diberikan.

Syntax dari metode extend() adalah sebagai berikut :

*list.extend(iterable)*

Melalui kode tersebut semua elemen dari iterable ditambahkan ke bagian akhir dari list1.

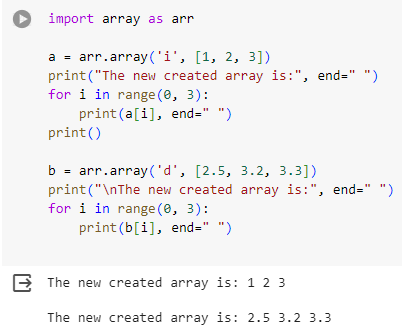
## Metode Praktikum

1. **Alat dan bahan**
   * PC
   * Google colab

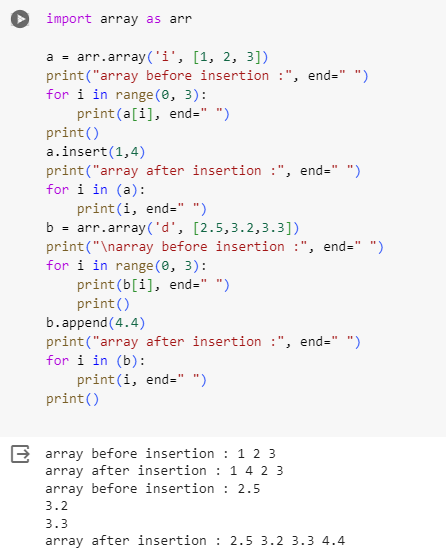
## Langkah kerja

1. Nyalakan pc yang akan digunakan
2. Buka modul yang telah diberikan dosen
3. Ikuti petunjuk yang ada pada modul
4. Kerjakan sesuai yang telah diperintahkan dalam modul

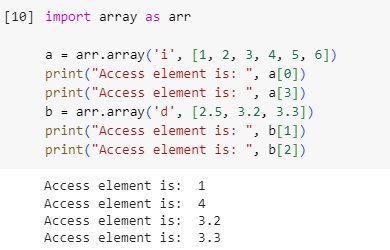
## Hasil dan Analisis



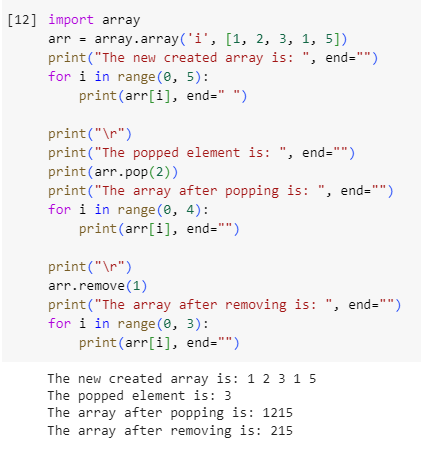
* + arr.array('i', [1, 2, 3]) membuat array dengan tipe data integer ('i') dan elemen-elemen [1, 2, 3].
  + Variabel a menjadi objek array yang menyimpan nilai [1, 2, 3].
  + Mencetak pesan "The new created array is:" tanpa baris baru.
  + Menggunakan loop for untuk mengakses dan mencetak setiap elemen array a.
  + Baris print() digunakan untuk menambahkan baris baru setelah mencetak elemen-elemen array a.
  + Mencetak pesan "The new created array is:" tanpa baris baru.
  + Menggunakan loop for untuk mengakses dan mencetak setiap elemen array a.
  + Baris print() digunakan untuk menambahkan baris baru setelah mencetak elemen-elemen array a.
  + Mencetak pesan "The new created array is:" tanpa baris baru.
  + Menggunakan loop for untuk mengakses dan mencetak setiap elemen array a.
  + Baris print() digunakan untuk menambahkan baris baru setelah mencetak elemen-elemen array a.



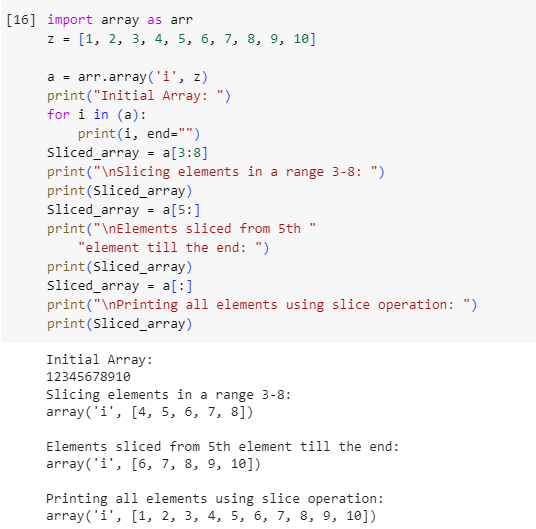
* + arr.array('i', [1, 2, 3]) membuat array dengan tipe data integer ('i') dan elemen-elemen [1, 2, 3].
  + Variabel a menjadi objek array yang menyimpan nilai [1, 2, 3].
  + Mencetak pesan "array before insertion:" tanpa baris baru.
  + Menggunakan loop for untuk mengakses dan mencetak setiap elemen array a.
  + Memanggil metode insert untuk menyisipkan nilai 4 ke indeks ke-1 pada array a.
  + Mencetak pesan "array after insertion:" tanpa baris baru.
  + Menggunakan loop for untuk mengakses dan mencetak setiap elemen array a setelah penyisipan.
  + arr.array('d', [2.5, 3.2, 3.3]) membuat array dengan tipe data float ('d') dan elemen-elemen [2.5, 3.2, 3.3].
  + Variabel b menjadi objek array yang menyimpan nilai [2.5, 3.2, 3.3].
  + Mencetak pesan "array before insertion:" dengan baris baru.
  + Menggunakan loop for untuk mengakses dan mencetak setiap elemen array b.
  + Memanggil metode append untuk menambahkan nilai 4.4 ke akhir array b.
  + Mencetak pesan "array after insertion:" tanpa baris baru.
  + Menggunakan loop for untuk mengakses dan mencetak setiap elemen array b setelah penyisipan.



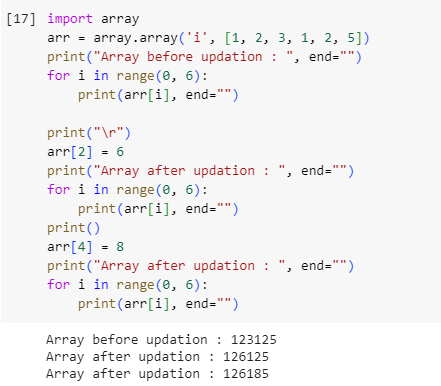
* + Membuat array a dengan tipe data integer ('i') dan elemen-elemen [1, 2, 3, 4, 5, 6].
  + Mengakses elemen array a pada indeks 0 dan 3 menggunakan operator indeks [].
  + Mencetak nilai elemen-elemen yang diakses.
  + Membuat array b dengan tipe data float ('d') dan elemen-elemen [2.5, 3.2, 3.3].
  + Mengakses elemen array b pada indeks 1 dan 2 menggunakan operator indeks [].
  + Mencetak nilai elemen-elemen yang diakses.



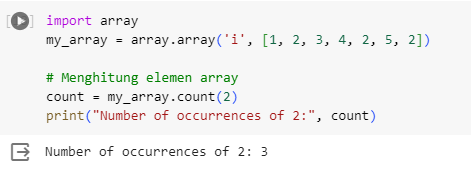
* + Membuat array arr dengan tipe data integer ('i') dan elemen-elemen [1, 2, 3, 1, 5].
  + Menggunakan loop for untuk mencetak elemen-elemen array arr.
  + Baris print("\r") digunakan untuk menambahkan baris baru setelah mencetak elemen-elemen.
  + Menggunakan metode pop(2) untuk menghapus elemen pada indeks ke-2 dari array arr.
  + Mencetak elemen yang dihapus dan array arr setelah penghapusan menggunakan loop for.
  + Menggunakan metode remove(1) untuk menghapus elemen pertama dengan nilai 1 dari array arr.
  + Mencetak array arr setelah penghapusan menggunakan loop for.



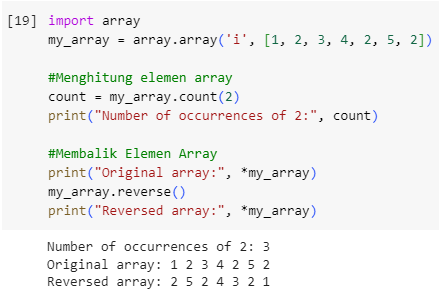
* + Membuat array a dengan tipe data integer ('i') dan elemen-elemen [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].
  + Menggunakan loop for untuk mencetak elemen-elemen array a.
  + Melakukan slicing pada array a dari indeks 3 hingga 8.
  + Mencetak hasil slicing dalam variabel Sliced\_array.
  + Melakukan slicing pada array a untuk mendapatkan seluruh array.
  + Mencetak hasil slicing dalam variabel Sliced\_array.



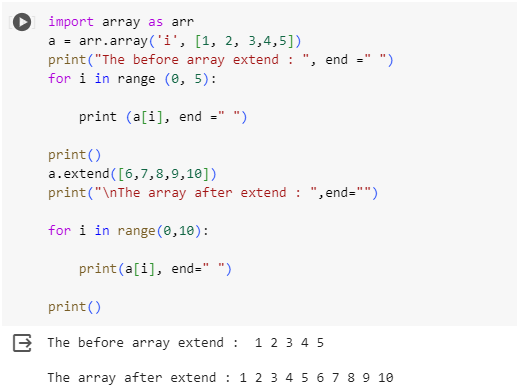
* + Membuat array arr dengan tipe data integer ('i') dan elemen-elemen [1, 2, 3, 1, 2, 5].
  + Menggunakan loop for untuk mencetak elemen-elemen array arr sebelum pembaruan.
  + Mengganti nilai elemen pada indeks ke-2 dengan nilai 6.
  + Mencetak elemen-elemen array arr setelah pembaruan menggunakan loop for.
  + Mengganti nilai elemen pada indeks ke-4 dengan nilai 8.
  + Mencetak elemen-elemen array arr setelah pembaruan menggunakan loop for.



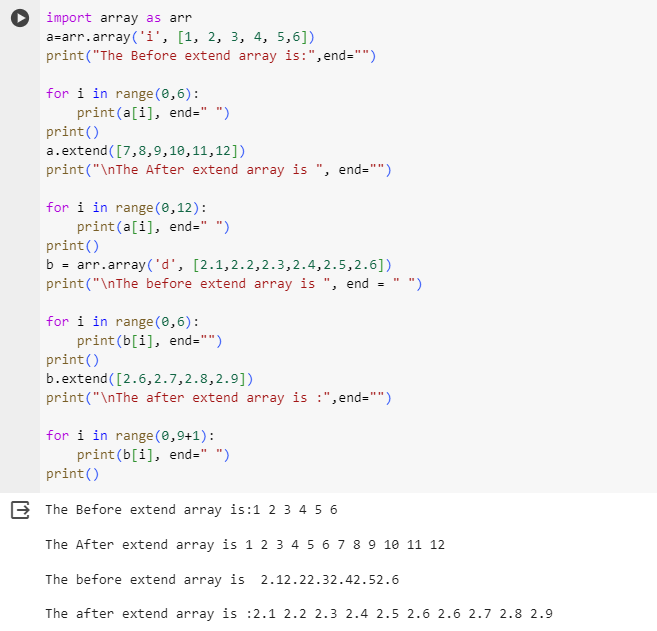
* + Membuat array my\_array dengan tipe data integer ('i') dan elemen-elemen [1, 2, 3, 4, 2, 5, 2].
  + Menggunakan metode count(2) untuk menghitung jumlah kemunculan elemen dengan nilai 2 dalam array my\_array.
  + Hasil perhitungan disimpan dalam variabel count.
  + Mencetak hasil perhitungan jumlah kemunculan elemen dengan nilai 2 dalam array my\_array.



* + Menggunakan metode count(2) untuk menghitung jumlah kemunculan elemen dengan nilai 2 dalam array my\_array.
  + Hasil perhitungan disimpan dalam variabel count.
  + Mencetak jumlah kemunculan elemen dengan nilai 2 dalam array.
  + Mencetak elemen-elemen array sebelum pembaruan.
  + Menggunakan metode reverse() untuk membalik urutan elemen-elemen dalam array my\_array.
  + Mencetak elemen-elemen array setelah pembalikan.



* + Membuat array a dengan tipe data integer ('i') dan elemen-elemen [1, 2, 3, 4, 5].
  + Menggunakan loop for untuk mencetak elemen-elemen array a sebelum dilakukan operasi extend.
  + Menggunakan metode extend([6, 7, 8, 9, 10]) untuk menambahkan elemen-elemen [6, 7, 8, 9, 10] ke akhir array a.
  + Mencetak elemen-elemen array a setelah dilakukan operasi extend.



* + Membuat array a dengan tipe data integer ('i') dan elemen-elemen [1, 2, 3, 4, 5, 6].
  + Menggunakan loop for untuk mencetak elemen-elemen array a sebelum dilakukan operasi extend.
  + Menggunakan metode extend([7, 8, 9, 10, 11, 12]) untuk menambahkan elemen-elemen baru ke akhir array a.
  + Mencetak elemen-elemen array a setelah dilakukan operasi extend.
  + Membuat array b dengan tipe data float ('d') dan elemen-elemen [2.1, 2.2, 2.3, 2.4, 2.5, 2.6].
  + Menggunakan loop for untuk mencetak elemen-elemen array b sebelum dilakukan operasi extend.
  + Menggunakan metode extend([2.6, 2.7, 2.8, 2.9]) untuk menambahkan elemen-elemen baru ke akhir array b.
  + Mencetak elemen-elemen array b setelah dilakukan operasi extend.

## Kesimpulan

Kesimpulan yang dapat diambil dari praktikum minggu ini dapat dinyatakan bahwa Penggunaan Modul array: Modul array pada Python menyediakan tipe data array yang efisien untuk menyimpan elemen dengan tipe data tertentu. Inisialisasi Array: Array dapat diinisialisasi dengan menggunakan modul array dan menyertakan tipe data elemen serta nilai-nilai awal. Akses Elemen Array: Elemen-elemen dalam array dapat diakses menggunakan indeks. Indeks array dimulai dari 0. Operasi Slicing: Slicing memungkinkan kita untuk mengakses sejumlah elemen dari array menggunakan notasi [start:stop]. Ini berguna untuk mendapatkan subset dari array. Operasi Pembaruan (Update): Elemen-elemen dalam array dapat diubah atau diperbarui menggunakan indeks. Operasi Penambahan (Extend): Metode extend dapat digunakan untuk menambahkan beberapa elemen ke akhir array. Operasi Penghapusan (Remove dan Pop): Metode remove digunakan untuk menghapus elemen dengan nilai tertentu, sedangkan metode pop digunakan untuk menghapus elemen pada indeks tertentu. Operasi Pembalikan (Reverse): Metode reverse digunakan untuk membalik urutan elemen dalam array. Menghitung Jumlah Kemunculan (Count): Metode count digunakan untuk menghitung berapa kali suatu elemen muncul dalam array. Kelebihan dan Keterbatasan: Array pada Python memberikan kelebihan dalam hal pengelolaan memori dan efisiensi akses elemen. Namun, array juga memiliki keterbatasan, seperti ketidakbisaan menambah atau mengurangi elemen setelah inisialisasi.

## Referensi

Ardiansah, Irfan. Permana, Hara, Ryan. *Memulai Python: Belajar Python Dari Nol.*

Cendekia press, Bandung: 2023.

Hidayah, Kharisma, Agung. Alam, Guntur, RG. Prihandoko. *Struktur Data Dengan Python.* Sonpedia Publishing Indonesia, Jambi: 2023.

Andini, Dwi, Titania. Setyorii, Arifin, Jaenal. Irsyadi, Eizzy, Abdulloh. Indahsari, Dewi, Rina. *Pelatihan Pemrograman Bahasa Python Pada Jurusan. Perangkat Lunak Dan Gim Smkn 12 Malang.* Malang: 2023. [http://dx.doi.org/10.26798/jpm.v2i2.880.](http://dx.doi.org/10.26798/jpm.v2i2.880)