

# Aplikasi *Selection Sort* pada Pengurutan Raket Bulu Tangkis Ramah Pemula

Aldwin Hardi Swastia - 13520167

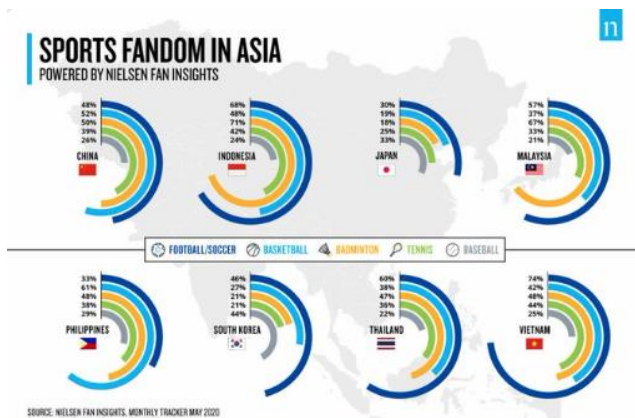
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
E-mail (gmail): 13520167@std.stei.itb.ac.id

**Abstrak**—Permainan olahraga yang semakin diminati oleh masyarakat membuat perlengkapan olahraga kian laris. Terdapat berbagai jenis olahraga, salah satunya adalah olahraga bola kecil. Biasanya olahraga bola kecil bergantung pada sebuah alat untuk memainkannya. Salah satunya adalah bulu tangkis, permainan yang memanfaatkan raket untuk memindahkan bola dari daerah diri ke daerah lawan. Raket menjadi hal yang sangat utama dalam permainan bulu tangkis. Terdapat banyak jenis raket, baik dari segi berat, fleksibilitas, serta keseimbangan dari raket bulu tangkis itu sendiri. Oleh karena itu, tujuan makalah ini adalah untuk menerapkan algoritma *selection sort* pada pengurutan raket-raket yang *beginner-friendly*.

**Keywords**—*brute force; sort; selection sort*

## I. PENDAHULUAN

Permainan bulu tangkis merupakan olahraga yang sangat diminati oleh masyarakat Indonesia. Hal ini dapat dilihat dari berbagai hal, seperti banyaknya pencapaian yang dicapai oleh pemain-pemain bulu tangkis asal Indonesia. Hal ini dapat dibuktikan pada survei olahraga di Asia Bernama Nielsen Sports. Sebanyak 71 persen masyarakat di Indonesia menyatakan suka pada bulu tangkis. Hal ini mengalahkan sepak bola sebagai olahraga yang cukup marak di Indonesia dengan persentase sebesar 68 persen menyukai sepak bola.



Gambar 1.1. Hasil Survei Olahraga di Asia  
Sumber: [1]

Permainan ini memerlukan sebuah raket untuk memindahkan bola dari daerah sendiri ke daerah lawan dengan

memukul bola yang biasa disebut *shuttlecock*. Saat ini, pandemi mulai menurun dan pemerintah telah melonggarkan berbagai jenis restriksi dalam berkegiatan. Hal ini membuat masyarakat mulai bermain bulu tangkis. Pemain terdiri atas pemain lama hingga pemain baru yang tertarik setelah menonton pertandingan-pertandingan bulu tangkis. Raket menjadi kebutuhan utama bagi seorang pemain bulu tangkis yang telah disesuaikan dengan kebutuhan pemain-pemain tersebut. Terdapat berbagai jenis raket untuk bulu tangkis yang dapat dibeli di pasaran. Jenis raket mulai dari berat, fleksibilitas, panjang *shaft*, *grip*, hingga keseimbangan dari raket itu sendiri. Tentu saja diperlukan banyak pengalaman dalam menggunakan raket sehingga tau spesifikasi raket yang diperlukan. Akan tetapi, hal ini membuat pemain-pemain pemula kesulitan untuk memilih raket yang sesuai karena banyaknya pilihan raket yang terdapat di pasar. Terkadang hal ini dapat diselesaikan dengan kenalan yang memiliki pengalaman cukup banyak di bulu tangkis. Di saat yang lain, banyak pemain pemula yang bingung dalam memilih raket pertamanya. Oleh karena itu, diperlukan sebuah pengurut raket-raket yang sesuai bagi para pemula.

Algoritma *selection sort* merupakan algoritma yang cukup dikenal dengan kompleksitas waktu  $O(n^2)$ . Algoritma ini termasuk ke dalam algoritma *brute force* karena melakukan penukaran terhadap setiap elemen yang dianggap lebih kecil nilainya. Hal ini membuat *selection sort* dapat menjadi solusi tercepat dalam mengatasi masalah ini walau akan membutuhkan waktu yang lama pada jumlah objek yang banyak.

## II. TEORI DASAR

### A. Brute Force

Algoritma *brute force* sendiri diimplementasikan dengan secara langsung melakukan iterasi terhadap masalah itu sendiri. Hal ini didasarkan oleh pernyataan persoalan serta konsep yang dilibatkan. Sebagai penjabar berikut merupakan karakteristik dari algoritma *brute force*:

1. Algoritma *brute force* adalah algoritma yang tidak mangkus dan cenderung mengeksekusi solusi secara keseluruhan tanpa kecuali sehingga waktu yang dibutuhkan juga lama

2. Algoritma *brute force* tidak cocok pada persoalan besar oleh karena waktu komputasi yang lama.

Kelebihan dari algoritma *brute force* sendiri antara lain:

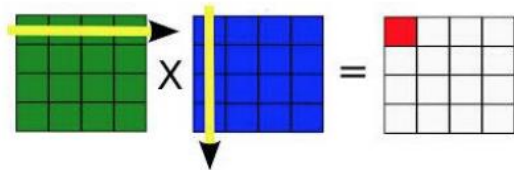
1. Algoritma yang sangat sederhana,
2. Dapat digunakan hampir di seluruh permasalahan,
3. langsung, dan
4. jelas caranya.

Kekurangan dari algoritma *brute force* antara lain:

1. Algoritma tidak mangkus, biasanya terdapat pendekatan algoritma lain yang jauh lebih ringkas,
2. Algoritma cenderung lambat maka tidak sesuai dengan masalah yang rumit,
3. Kurangnya kreativitas dari algoritma, banyak pembatasan yang dapat dilakukan sehingga algoritma tidak perlu melakukan pengecekan pada hal-hal yang pasti salah.

Banyak permasalahan yang memanfaatkan algoritma *brute force*, seperti mencari elemen terbesar/terkecil, pencarian beruntun (*sequential search*), menghitung  $a^n$ , menghitung  $n!$ , mengalikan dua buah matriks, uji bilangan prima, dan lain-lain. Dapat dibilang hampir semua persoalan dapat diselesaikan dengan *brute force* dengan kekurangan kompleksitas waktu yang cenderung lama. Sebagai contoh, isPrima yang merupakan algoritma penentu sebuah bilangan prima atau bukan dilakukan dengan cara melakukan traversal terhadap angka-angka dari 0 hingga angka yang diuji dan mengeceknya apakah angka yang diuji dapat dibagi dengan angka tersebut. Apabila dapat dibagi *loop* dihentikan dan menandakan bahwa angka yang diuji bukanlah prima. Dengan begitu, algoritma tersebut memiliki kasus terburuk sebesar  $n$  karena mengecek semua angka hingga angka itu sendiri.

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj} = \sum_{k=1}^n a_{ik}b_{kj}$$



Gambar 2.1. Contoh Algoritma *brute force* dalam penyelesaian perkalian matriks

Sumber: [2]

Pada permasalahan kali ini algoritma diimplementasikan dengan melakukan traversal terhadap *array* sebagai elemen pertama yang akan dibandingkan. Kemudian, traversal akan dilakukan untuk kedua kalinya untuk menentukan elemen kedua yang akan dibandingkan nantinya. Artinya untuk melakukan pengurutan pada suatu kumpulan objek, diperlukan *loop* sebanyak  $n \times n$  sehingga kompleksitas algoritma  $O(n^2)$ .

## B. Algoritma Sorting

Algoritma pengurutan merupakan algoritma yang ada untuk melakukan pengurutan terhadap kumpulan dari objek. Banyak pendekatan dari pengurutan itu juga, seperti *brute force* dan *divide and conquer*. Beberapa algoritma pengurutan antara lain:

1. *bubble sort*,
2. *selection sort*,
3. *insertion sort*,
4. *shell sort*,
5. *merge sort*,
6. *quick sort*,
7. *Heap sort* dan lain-lain.

| Algorithm                      | Time Complexity     |                     |                |
|--------------------------------|---------------------|---------------------|----------------|
|                                | Best                | Average             | Worst          |
| <a href="#">Selection Sort</a> | $\Omega(n^2)$       | $\theta(n^2)$       | $O(n^2)$       |
| <a href="#">Bubble Sort</a>    | $\Omega(n)$         | $\theta(n^2)$       | $O(n^2)$       |
| <a href="#">Insertion Sort</a> | $\Omega(n)$         | $\theta(n^2)$       | $O(n^2)$       |
| <a href="#">Heap Sort</a>      | $\Omega(n \log(n))$ | $\theta(n \log(n))$ | $O(n \log(n))$ |
| <a href="#">Quick Sort</a>     | $\Omega(n \log(n))$ | $\theta(n \log(n))$ | $O(n^2)$       |
| <a href="#">Merge Sort</a>     | $\Omega(n \log(n))$ | $\theta(n \log(n))$ | $O(n \log(n))$ |
| <a href="#">Bucket Sort</a>    | $\Omega(n+k)$       | $\theta(n+k)$       | $O(n^2)$       |
| <a href="#">Radix Sort</a>     | $\Omega(nk)$        | $\theta(nk)$        | $O(nk)$        |
| <a href="#">Count Sort</a>     | $\Omega(n+k)$       | $\theta(n+k)$       | $O(n+k)$       |

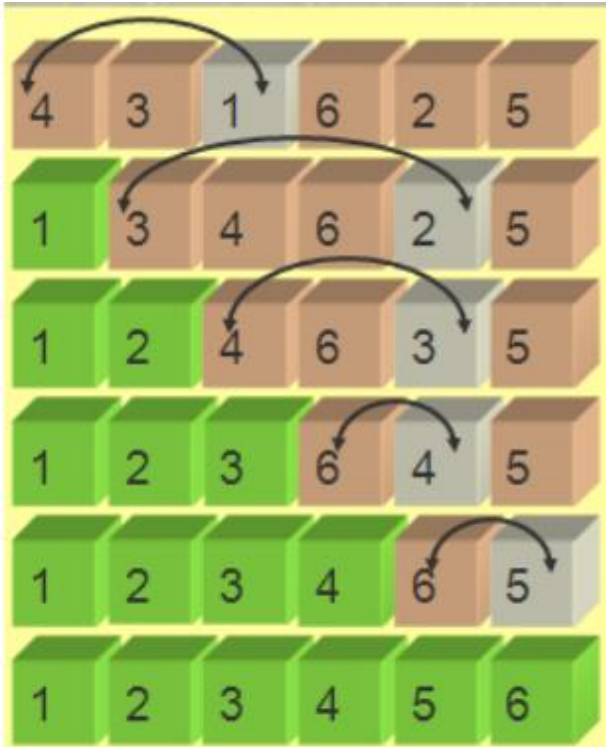
Gambar 2.2. Perbandingan kompleksitas waktu algoritma pengurutan

Sumber: [5]

Pada kali permasalahan kali ini, algoritma sort yang digunakan menggunakan pendekatan *brute force*, seperti *selection sort* dan *bubble sort*. Pada *selection sort* elemen akan dipilih untuk dilakukan pertukaran. Sedangkan *bubble sort*, pertukaran dilakukan setiap iterasi apabila elemen setelahnya lebih besar/kecil.

### C. Selection Sort

Algoritma *selection sort* merupakan salah satu algoritma pengurutan dengan pendekatan *brute force*. Cara kerja algoritma tersebut meliputi dua *loop* yang digunakan untuk melakukan perbandingan pada setiap elemen yang berada di setelahnya, hal ini dilakukan sebanyak  $n$  kali dan dibandingkan dengan  $n-1$  kali.



Gambar 2.3. Ilustrasi selection sort  
Sumber: [2]

Berikut merupakan langkah-langkah pengerjaan *selection sort*:

1. Mulai dari elemen pertama kemudian lakukan pilih elemen yang akan dibandingkan dengan memilih elemen-elemen yang ada di depannya
2. Lakukan pertukaran setiap elemen yang di depannya lebih besar dari elemen tersebut
3. Lakukan pemilihan hingga dicek elemen terakhir.
4. Ulangi langkah pertama pada elemen kedua hingga elemen terakhir

Seperti yang dapat dilihat dari langkah-langkah di atas, akan terjadi perbandingan sebanyak  $n(n-1)/2$  kali dan pertukaran sebanyak  $n-1$  kali. Oleh karena itu, kompleksitas waktu algoritma diukur dari jumlah perbandingan yaitu  $O(n^2)$ . Berikut merupakan pseudocode dari *selection sort*

**Procedure** SelectionSort (input/output  $s_1, s_2, \dots, s_n$  : integer)

**Deklarasi**

$i, j, \text{imin}, \text{temp}$  : integer

### Algoritma:

```
for i ← 1 to n – 1 do { jumlah pass sebanyak n – 1 }
    { cari elemen terkecil di dalam  $s[i], s[i+1], \dots, s[n]$  }
    imin ← i { elemen ke-i diasumsikan sebagai elemen
    terkecil sementara }
    for j ← i+1 to n do
        if  $s[j] < s[\text{imin}]$  then
            imin ← j
        endif
    endfor
    {pertukarkan  $s[\text{imin}]$  dengan  $s[i]$  }
    temp ←  $s[i]$ 
     $s[i] \leftarrow s[\text{imin}]$ 
     $s[\text{imin}] \leftarrow \text{temp}$ 
endfor
```

### D. Bulu Tangkis

Permainan bulu tangkis merupakan salah satu olahraga bola kecil yang pada permainannya bertujuan untuk mendaratkan bola pada daerah lawan. Permainan sangat disenangi oleh masyarakat Indonesia karena banyaknya tokoh-tokoh pebulu tangkis yang mendunia. Permainan ini dimainkan oleh dua orang untuk partai tunggal dan empat orang untuk partai ganda yang dibagi dalam dua tim. Masing-masing tim bertujuan untuk menjatuhkan bola (*shuttlecock*) pada daerah lawan dan mencegah *shuttlecock* untuk mendarat di daerah diri. Kedua tim akan bermain di satu lapangan yang dipisahkan oleh net sebagai batas daerah.



Gambar 2.4. Gambar permainan bulu tangkis  
Sumber: [3]

Adapun perlengkapan yang dibutuhkan dalam bermain bulu tangkis:

1. Raket sebagai pemukul bola,
2. Senar pemantul bola pada raket,
3. Kok sebagai bola, dan
4. Sepatu.

Selain itu, terdapat banyak teknik-teknik dasar dalam permainan badminton, seperti:

1. Teknik dasar bersikap
  - a. Berdiri tegak dengan kedua kaki menjaga keseimbangan tubuh
  - b. Kedua lutut ditekuk dan dibuka selebar bahu dengan kondisi rileks
  - c. Posisikan lengan raket di samping tubuh pada kondisi bebas bergerak
  - d. Mengikuti aturan
2. Teknik dasar memegang raket
  - a. Raket dicengkram dengan santai
  - b. Ibu jari dan telunjuk berbentuk huruf V pada pegangan raket
  - c. Pegang raket pada jari-jari.
3. Teknik dasar servis
  - a. Servis *forehand* rendah
  - b. Servis *forehand* tinggi
  - c. Servis *backhand*
  - d. Servis *flick*
4. Teknik dasar kaki
 

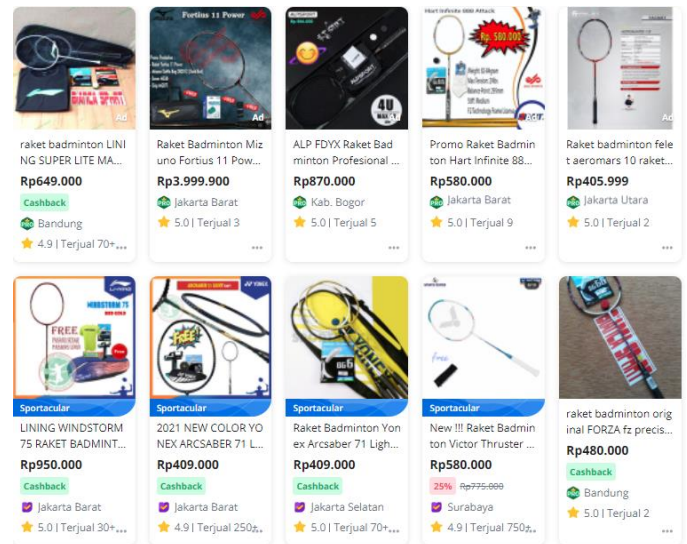
Gerakan kaki yang terlatih akan mempercepat pergerakan di lapangan yang akan menambah akurasi pada pukulan-pukulan selanjutnya
5. Teknik dasar *stroke*

Sikap saat mau memukul bola perlu diperhatikan karena dikap memukul bola yang baik akan memberikan pantulan bola yang baik juga.
6. Teknik dasar pukulan
  - a. *Overhead forehand*
  - b. *Overhead backhand*
  - c. *Underarm forehand*
  - d. *Underarm backhand*

Pemain pemula dapat mempraktikan hal di atas untuk bermain bulu tangkis langsung.

#### E. Raket

Pada permainan bulu tangkis, sebuah raket merupakan penentu arah dari pukulan sebuah *shuttlecock* sehingga diperlukan spesifikasi yang sesuai agar pemain dapat bermain dengan baik. Namun, terlalu banyak pilihan raket yang dapat dibeli di toko. Berbagai spesifikasi disediakan pada toko-toko olahraga baik di toko konvensional maupun online.



Gambar 2.5. Gambar banyaknya pilihan raket  
Sumber: Dokumen Pribadi

Pada gambar di atas dapat dilihat berbagai macam raket yang ada di toko. Raket-raket tersebut juga memiliki spesifikasi yang berbeda-beda juga. Hal inilah mengapa diperlukan sebuah pengurut untuk raket yang ada sehingga para pemula tidak kesulitan dalam memilih raket.

### III. PEMBAHASAN

#### A. Implementasi Program

Pada permasalahan pemilihan raket, algoritma sorting difokuskan pada berat raket terlebih dahulu. Hal ini karena pemain pemula masih belum terbiasa dalam mengayunkan raket yang lebih berat sehingga memiliki kemungkinan cedera yang tinggi. Kemudian, mengurutkannya berdasarkan pada fleksibilitas, keseimbangan, dan harga. Selain hal yang telah disebutkan, spesifikasi raket cenderung tidak terlalu berpengaruh pada pengalaman bermain seorang pemula.

Raket memiliki rentang berat dari 1u hingga 7u yang artinya semakin tinggi angkanya semakin ringan juga raket tersebut. Pemula akan lebih baik memilih raket yang lebih ringan. Setelah itu, terdapat 3 fleksibilitas yaitu *high*, *medium*, *low*. Raket yang memiliki fleksibilitas yang tinggi akan memaksimalkan pantulan yang dihasilkan oleh raket sehingga cocok untuk pemula. Sedangkan pada keseimbangan, Tingkat keseimbangan terdiri dari *balanced*, *head-heavy*, *head-light*. Raket yang seimbang artinya raket dapat digunakan hampir pada semua skenario. Selanjutnya, raket dengan kepala raket yang lebih berat lebih dianjurkan karena dengan lebih beratnya kepala, pukulan yang dihasilkan akan lebih keras.

Menimbang hal di atas, pendekatan solusi yang dilakukan dengan memberikan poin dalam tiap spesifikasinya dengan rentang 1-7 pada berat raket (semakin tinggi semakin baik), rentang 1-3 pada fleksibilitas raket (semakin fleksibel semakin baik), dan rentang 1-3 pada keseimbangan raket (*balanced*(3), *head-heavy*(2), *head-light*(1)). Apabila memiliki poin yang sama, akan diurutkan sesuai harga sehingga didapatkan raket yang murah sesuai dengan spesifikasi ramah pemula.

## B. Aplikasi Selection Sort pada Pengurutan Raket

Pada pengujian kali ini digunakan data *dummy* untuk menyelesaikan permasalahan raket pemula. Data *dummy* yang digunakan adalah sebagai berikut:

```
listOfRackets = [
    {'name': 'Racket 1', 'price': '$100', 'weight': '4', 'balance': 'balanced', 'flexibility': 'stiff', 'brand': 'Li-Ning'},
    {'name': 'Racket 2', 'price': '$200', 'weight': '5', 'balance': 'head-heavy', 'flexibility': 'medium', 'brand': 'Li-Ning'},
    {'name': 'Racket 3', 'price': '$300', 'weight': '6', 'balance': 'head-light', 'flexibility': 'high', 'brand': 'Li-Ning'},
    {'name': 'Racket 4', 'price': '$400', 'weight': '3', 'balance': 'balanced', 'flexibility': 'high', 'brand': 'Yonex'},
    {'name': 'Racket 5', 'price': '$500', 'weight': '3', 'balance': 'head-heavy', 'flexibility': 'medium', 'brand': 'Yonex'},
    {'name': 'Racket 6', 'price': '$600', 'weight': '1', 'balance': 'balanced', 'flexibility': 'medium', 'brand': 'Yonex'},
    {'name': 'Racket 7', 'price': '$700', 'weight': '2', 'balance': 'head-light', 'flexibility': 'high', 'brand': 'Victor'},
    {'name': 'Racket 8', 'price': '$800', 'weight': '3', 'balance': 'balanced', 'flexibility': 'stiff', 'brand': 'Victor'},
    {'name': 'Racket 9', 'price': '$900', 'weight': '4', 'balance': 'head-heavy', 'flexibility': 'stiff', 'brand': 'Victor'}
]
```

Gambar 3.1. Data dummy  
Sumber: Dokumen Pribadi

Sebagai penjas, berikut merupakan pseudocode algoritma yang digunakan

**Procedure** selectionSort (input/output listOfRackets: array of racket)

### Deklarasi

i, j, imin: integer

temp: racket

### Algoritma:

**for** i ← 1 to n – 1 **do** { jumlah pass sebanyak n – 1 }

imin ← i { elemen ke-i diasumsikan sebagai elemen terkecil sementara }

**for** j ← i+1 to n **do**

    point1 = pointCounter(listOfRackets[imin])

    point2 = pointCounter(listOfRackets[j])

**if** poin2 > poin1 **then**

        imin ← j

**else if** poin2 = poin1 **then**

**if** listOfRackets[imin]['price'] > listOfRackets[j]['price'] **then**

            imin ← j

**endif**

**endfor**

{pertukarkan s[imin] dengan s[i]}

temp ← listOfRackets [i]

listOfRackets [i] ← listOfRackets [imin]

listOfRackets [imin] ← temp

**endifor**

Pada pengujian kali ini, implementasi program menggunakan bahasa pemrograman Python sebagai platform eksekusi program. Hasil eksekusi program dalam mengurutkan raket sebagai berikut

```
The list of racket is:
{'name': 'Racket 1', 'price': '$100', 'weight': '4', 'balance': 'balanced', 'flexibility': 'stiff', 'brand': 'Li-Ning'}
{'name': 'Racket 2', 'price': '$200', 'weight': '5', 'balance': 'head-heavy', 'flexibility': 'medium', 'brand': 'Li-Ning'}
{'name': 'Racket 3', 'price': '$300', 'weight': '6', 'balance': 'head-light', 'flexibility': 'high', 'brand': 'Li-Ning'}
{'name': 'Racket 4', 'price': '$400', 'weight': '3', 'balance': 'balanced', 'flexibility': 'high', 'brand': 'Yonex'}
{'name': 'Racket 5', 'price': '$500', 'weight': '3', 'balance': 'head-heavy', 'flexibility': 'medium', 'brand': 'Yonex'}
{'name': 'Racket 6', 'price': '$600', 'weight': '1', 'balance': 'balanced', 'flexibility': 'medium', 'brand': 'Yonex'}
{'name': 'Racket 7', 'price': '$700', 'weight': '2', 'balance': 'head-light', 'flexibility': 'high', 'brand': 'Victor'}
{'name': 'Racket 8', 'price': '$800', 'weight': '3', 'balance': 'balanced', 'flexibility': 'stiff', 'brand': 'Victor'}
{'name': 'Racket 9', 'price': '$900', 'weight': '4', 'balance': 'head-heavy', 'flexibility': 'stiff', 'brand': 'Victor'}

The sorted list of racket is:
{'name': 'Racket 3', 'price': '$300', 'weight': '6', 'balance': 'head-light', 'flexibility': 'high', 'brand': 'Li-Ning'}
{'name': 'Racket 2', 'price': '$200', 'weight': '5', 'balance': 'head-heavy', 'flexibility': 'medium', 'brand': 'Li-Ning'}
{'name': 'Racket 4', 'price': '$400', 'weight': '3', 'balance': 'balanced', 'flexibility': 'high', 'brand': 'Yonex'}
{'name': 'Racket 1', 'price': '$100', 'weight': '4', 'balance': 'balanced', 'flexibility': 'stiff', 'brand': 'Li-Ning'}
{'name': 'Racket 5', 'price': '$500', 'weight': '3', 'balance': 'head-heavy', 'flexibility': 'medium', 'brand': 'Yonex'}
{'name': 'Racket 8', 'price': '$800', 'weight': '3', 'balance': 'balanced', 'flexibility': 'stiff', 'brand': 'Victor'}
{'name': 'Racket 9', 'price': '$900', 'weight': '4', 'balance': 'head-heavy', 'flexibility': 'stiff', 'brand': 'Victor'}
{'name': 'Racket 6', 'price': '$600', 'weight': '1', 'balance': 'balanced', 'flexibility': 'medium', 'brand': 'Yonex'}
{'name': 'Racket 7', 'price': '$700', 'weight': '2', 'balance': 'head-light', 'flexibility': 'high', 'brand': 'Victor'}
```

Gambar 3.2. Hasil eksekusi program  
Sumber: Dokumen Pribadi

Pada eksekusi program dapat dilihat bahwa raket berhasil diurutkan berdasarkan spesifikasi pemula. Terdapat beberapa poin penting yang perlu diperhatikan pada pengurutan di atas:

1. Raket yang memiliki berat di atas 4u akan menjadi raket yang lebih *preferable* untuk pemula karena ringan.
2. Pada raket yang memiliki berat di sekitar 3u dan 4u akan bergantung pada fleksibilitas dan keseimbangan raket tersebut hal ini juga sesuai dengan pemain-pemain yang umumnya menggunakan raket dengan berat 3u dan 4u.
3. Sedangkan pada raket dengan berat 1u dan 2u sudah pasti akan menjadi urutan terbawah akibat bobotnya yang terlalu berat. Bahkan pemain profesional tidak ada yang menggunakan raket berbobot 1u dan 2u.

Dari ketiga poin di atas, mencerminkan bahwa pengurutan relevan terhadap kebutuhan pemain bulu tangkis saat ini.

## IV. KESIMPULAN

Algoritma *selection sort* dapat digunakan untuk melakukan pengurutan terhadap kumpulan raket-raket yang ada berdasarkan kategori raket pemula. Pengurutan dilakukan dengan memberikan poin-poin pada setiap spesifikasi raket yang berpengaruh paling besar. Semakin ringan semakin baik, semakin fleksibel semakin baik, dan keseimbangan diurutkan berdasarkan seimbang, berat kepala, ringan kepala.

Harapannya fitur pengurutan pada raket dapat diimplementasikan pada situs penjualan raket ataupun aplikasi pencarian raket sehingga pemain-pemain pemula tidak kesulitan dalam memilih raket yang akan dibeli.

## V. UCAPAN TERIMA KASIH

Puji syukur pada Tuhan Yang Maha Esa karena atas berkatnya, penulis dapat menyelesaikan karya tulis dengan baik dan tepat waktu. Terima kasih kepada orang tua dan dosen pengampu Strategi Algoritma K02, Bu Ulfa, dalam membimbing penulis dengan ilmu yang telah diberikan selama kuliah. Selain itu, terima kasih kepada teman-teman yang telah mendukung penulis selama penulisan makalah ini.

## REFERENCES

- [1] <https://www.kompas.com/sports/read/2020/09/25/10000058/survei-nielsen-pastikan-badminton-jadi-olahraga-terpopuler-di-indonesia> diakses pada 23 Mei 2022 pukul 17.16.
- [2] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Brute-Force-\(2021\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Brute-Force-(2021)-Bag1.pdf) diakses pada 23 Mei 2022 pukul 17.43.
- [3] [https://asset.kompas.com/crops/osRoSjXDxjjOTeuEqGCpz2C\\_9y0=/0x0:900x600/750x500/data/photo/2021/07/27/60ff6c4e0a7ee.jpg](https://asset.kompas.com/crops/osRoSjXDxjjOTeuEqGCpz2C_9y0=/0x0:900x600/750x500/data/photo/2021/07/27/60ff6c4e0a7ee.jpg). Diakses pada 23 Mei pukul 20.43.
- [4] <https://kumparan.com/info-sport/cara-bermain-badminton-begini-teknik-yang-benar-1wkCAX4kUQB/full> diakses pada 23 Mei 2022 pukul 20.59.
- [5] [Time Complexities of all Sorting Algorithms - GeeksforGeeks](#) diakses pada 23 Mei 2022 pukul 22.40.

#### PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 23 Mei 2022



Aldwin Hardi Swastia 13520167