**Prosiding Seminar Nasional Sains dan Teknologi Seri 02**
**Fakultas Sains dan Teknologi, Universitas Terbuka**
**Vol. 1 No. 2 (2024)**
**e-ISSN: 3047-6569**

# IMPLEMENTASI ALGORITMA *SELECTION SORT* UNTUK PENGURUTAN DATA MAHASISWA DI PROGRAM STUDI TEKNIK INFORMATIKA POLITEKNIK NEGERI PONTIANAK

**Ferrosi Pratama, Dwiky Juniardi, Muhammad Fauzan Arif, Suharsono***
*Program Studi Teknik Informatika, Politeknik Negeri Pontianak, Pontianak*

**Penulis korespondensi: suhar2006@gmail.com*

## ABSTRAK

Pengelolaan data mahasiswa merupakan elemen penting dalam sistem informasi akademik di sebuah program studi. Dimana data dapat ditemukan kembali dengan mudah, cepat dan akurat. Mahasiswa terdiri dari nama, NIM, semester, status keaktifan, dan lain sebagainya. Pada jumlah data mahasiswa yang banyak tentunya akan mengalami kesulitan dalam mencari atau mengurutkan data mahasiswa sesuai urutan yang dibutuhkan. Penelitian ini bertujuan untuk menerapkan algoritma sorting yaitu selection sort dalam mengurutkan data mahasiswa berdasarkan nama, NIM, kelas, semester, dan status. Data yang digunakan adalah data mahasiswa semester 2, 4, dan 6 tahun akademik 2023/2024. Algoritma *selection sort* ini diimplementasikan menggunakan bahasa pemrograman Python untuk mengetahui waktu eksekusi dalam proses pengurutan. Hasil penelitian ini menunjukkan bahwa algoritma *selection sort* mampu mengurutkan data mahasiswa dengan benar dan konsisten berdasarkan kata kunci yang diberikan baik secara *ascending* maupun *descending*. Implementasi dengan visualisasi juga mempermudah pemahaman, sehingga bermanfaat untuk pengajaran algoritma *sorting* di lingkungan akademik. Penelitian ini menyediakan dasar untuk pengembangan lebih lanjut dalam pemilihan algoritma *sorting* yang sesuai dengan ukuran dan kebutuhan data.

**Kata kunci:** *selection sort*, algoritma, python, *sorting*, mahasiswa

## 1 PENDAHULUAN

Pengurutan data atau "*Sorting*" adalah suatu proses menyusun kembali data yang sebelumnya telah tersusun secara acak hingga tersusun secara teratur menurut aturan tertentu. Data ini biasanya bertipe numerik dan karakter (N. Sari *et al*., 2022). Ada banyak algoritma penggabungan dan penyortiran yang telah dikembangkan untuk memecahkan masalah penggabungan dan penyortiran data besar. Begitu juga algoritma penyortiran dan penggabungan yang lebih tua telah ditingkatkan untuk menurunkan waktu berjalan mereka dan meningkatkan kecepatan mereka untuk membuatnya lebih efisien. Pengurutan data memiliki peranan penting yang patut dipertimbangkan agar keseluruhan permasalahan (terutama mengenai pengolahan data) menjadi lebih mudah dan lebih cepat untuk diselesaikan (Rizki Saputra *et al*., 2021).

Pengurutan data mahasiswa adalah salah satu konsep fundamental dalam ilmu komputer yang memiliki aplikasi luas dalam pengelolaan informasi akademik. Pengurutan data yang efisien dan efektif dapat meningkatkan produktivitas dan akurasi dalam pengelolaan data. Algoritma *selection sort* memiliki kelebihan yang dapat mudah dipahami dan diimplementasikan, serta tidak memerlukan banyak ruang penyimpanan. Namun, kekurangan dari algoritma ini adalah bahwa ia memiliki kompleksitas waktu yang relatif lambat dibandingkan dengan algoritma lain (Aura D.E. & Otik, 2023).

Dalam kasus ini, data diurutkan secara *ascending* dan *descending* berdasarkan kriteria, seperti nama, NIM, kelas, semester, dan status. Algoritma yang digunakan untuk mengurutkan data adalah *selection sort*. Algoritma ini berfungsi dengan cara memilih elemen terkecil dari *array* dan memindahkannya ke posisi awal, kemudian memilih elemen terkecil berikutnya dan memindahkannya ke posisi berikutnya, dan seterusnya. Dengan menggunakan teknik optimasi yang tepat, kecepatan algoritma ini dapat ditingkatkan sehingga menjadi lebih cepat dan lebih efektif dalam mengurutkan data. Dalam konteks akademik, pengurutan data mahasiswa sangat penting untuk mempermudah berbagai aktivitas administratif, seperti pencarian data individual, dan pengelompokan mahasiswa berdasarkan semester atau kelas. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi pada pengelolaan informasi akademik yang lebih baik dan lebih efisien. Menurut (Yanti & Sita Eriana, 2024) *ascending* adalah pengurutan data yang dimulai dari nilai yang paling kecil ke nilai yang paling besar. Identifikasi elemen yang lebih kecil dari data dan menukarnya dengan elemen pertama, kemudian identifikasi elemen terkecil dari sisa data (setelah elemen pertama) dan menukarnya dengan elemen kedua, dan begitu seterusnya sampai seluruh data diurutkan. Sedangkan *descending* adalah dengan cara memindahkan nilai data yang paling besar ke posisi paling kiri, demikian seterusnya untuk nilai data yang besar berikutnya diletakkan di posisi sebelah kanan dari nilai terbesar sebelumnya yang sudah diletakkan di posisi sebelah kiri (Sunandar & Indrianto, 2020)

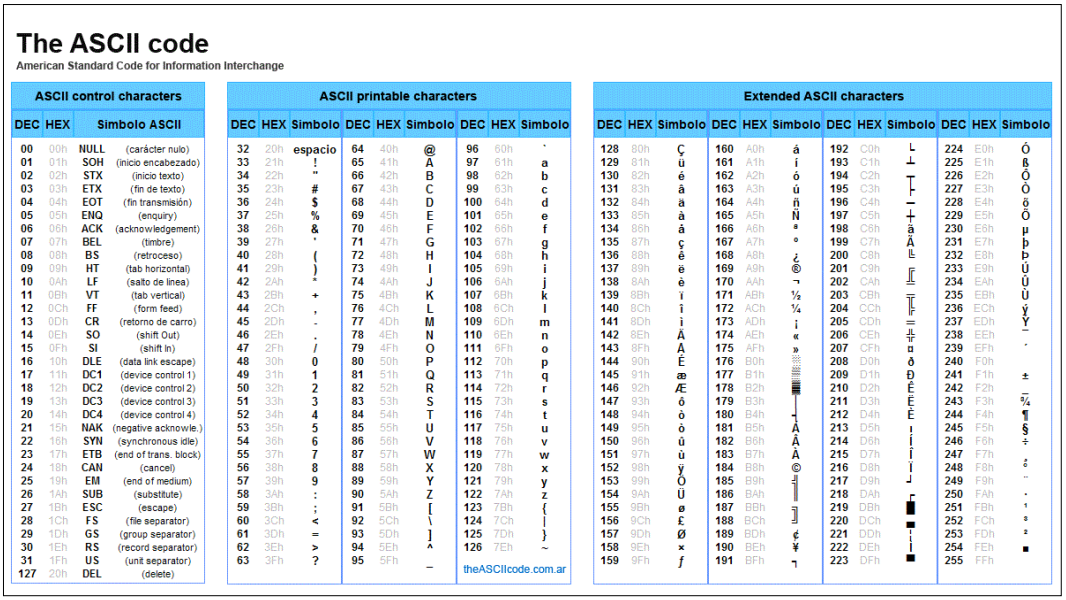
Metode *selection sort* adalah suatu metode *sorting* yang membandingkan elemen pertama dengan elemen berikutnya sampai elemen terakhir. Jika ditemukan elemen lain yang lebih kecil dari elemen pertama, maka posisinya akan disimpan dan langsung ditukar. Perbandingan dilakukan terus menerus hingga tidak ada lagi pertukaran data (Sandria *et al*., 2022). Cara kerja algoritma ini adalah dengan menggunakan teknik perulangan, di mana program secara berulang memilih bilangan dalam blok pertama dan membandingkannya dengan bilangan dalam blok kedua, ketiga, dan seterusnya. Jika ditemukan bilangan yang lebih kecil daripada bilangan dalam blok pertama, maka posisi *swap* diganti (I. P. Sari *et al*., 2023). *Selection sort* adalah algoritma pengurutan sederhana yang membuat beberapa lintasan untuk memilih *item* data. Algoritma ini dikenal karena sederhananya dan seringkali berhasil lebih baik daripada algoritma lain yang lebih sulit dalam beberapa situasi (Yovita *et al*., 2023). Menurut (Wisda, 2022), algoritma *selection sort* merupakan perbaikan dari algoritma *bubble sort* dengan mengurangi jumlah perbandingan.

## 2 METODE

Metode pada penelitian ini yaitu menggunakan algoritma *selection sort* terhadap dua kriteria pengurutan data. Pertama berdasarkan nama mahasiswa dan kedua berdasarkan Nomor Induk Mahasiswa (NIM). Kedua kriteria diurutkan secara *ascending* dari data terkecil ke data terbesar dan *descending* dari data terbesar ke data terkecil dengan menggunakan data *set* dalam bentuk *file* csv.

### 2.1 Algoritma *Selection Sort*

Pada data yang bertipe *char*, nilai data dikatakan lebih kecil atau lebih besar dari yang lain didasarkan pada urutan relatif (*collating sequence*) seperti dinyatakan dalam tabel ASCII pada **Gambar 1** (N. Sari *et al*., 2022).



The ASCII code

American Standard Code for Information Interchange

ASCII control characters

| DEC | HEX | Simbolo ASCII | |
|---|---|---|---|
| 00 | 00h | NULL | (carácter nulo) |
| 01 | 01h | SOH | (inicio encabezado) |
| 02 | 02h | STX | (inicio texto) |
| 03 | 03h | ETX | (fin de texto) |
| 04 | 04h | EOT | (fin transmisión) |
| 05 | 05h | ENQ | (enquiry) |
| 06 | 06h | ACK | (acknowledgement) |
| 07 | 07h | BEL | (timbre) |
| 08 | 08h | BS | (retroceso) |
| 09 | 09h | HT | (tab horizontal) |
| 10 | 0Ah | LF | (salto de línea) |
| 11 | 0Bh | VT | (tab vertical) |
| 12 | 0Ch | FF | (form feed) |
| 13 | 0Dh | CR | (retorno de carro) |
| 14 | 0Eh | SO | (shift Out) |
| 15 | 0Fh | SI | (shift In) |
| 16 | 10h | DLE | (data link escape) |
| 17 | 11h | DC1 | (device control 1) |
| 18 | 12h | DC2 | (device control 2) |
| 19 | 13h | DC3 | (device control 3) |
| 20 | 14h | DC4 | (device control 4) |
| 21 | 15h | NAK | (negative acknowle.) |
| 22 | 16h | SYN | (synchronous idle) |
| 23 | 17h | ETB | (end of trans. block) |
| 24 | 18h | CAN | (cancel) |
| 25 | 19h | EM | (end of medium) |
| 26 | 1Ah | SUB | (substitute) |
| 27 | 1Bh | ESC | (escape) |
| 28 | 1Ch | FS | (file separator) |
| 29 | 1Dh | GS | (group separator) |
| 30 | 1Eh | RS | (record separator) |
| 31 | 1Fh | US | (unit separator) |
| 127 | 20h | DEL | (delete) |

ASCII printable characters

| DEC | HEX | Simbolo | DEC | HEX | Simbolo | DEC | HEX | Simbolo |
|---|---|---|---|---|---|---|---|---|
| 32 | 20h | espacio | 64 | 40h | @ | 96 | 60h | ` |
| 33 | 21h | ! | 65 | 41h | A | 97 | 61h | a |
| 34 | 22h | " | 66 | 42h | B | 98 | 62h | b |
| 35 | 23h | # | 67 | 43h | C | 99 | 63h | c |
| 36 | 24h | $ | 68 | 44h | D | 100 | 64h | d |
| 37 | 25h | % | 69 | 45h | E | 101 | 65h | e |
| 38 | 26h | & | 70 | 46h | F | 102 | 66h | f |
| 39 | 27h | ' | 71 | 47h | G | 103 | 67h | g |
| 40 | 28h | ( | 72 | 48h | H | 104 | 68h | h |
| 41 | 29h | ) | 73 | 49h | I | 105 | 69h | i |
| 42 | 2Ah | * | 74 | 4Ah | J | 106 | 6Ah | j |
| 43 | 2Bh | + | 75 | 4Bh | K | 107 | 6Bh | k |
| 44 | 2Ch | , | 76 | 4Ch | L | 108 | 6Ch | l |
| 45 | 2Dh | - | 77 | 4Dh | M | 109 | 6Dh | m |
| 46 | 2Eh | . | 78 | 4Eh | N | 110 | 6Eh | n |
| 47 | 2Fh | / | 79 | 4Fh | O | 111 | 6Fh | o |
| 48 | 30h | 0 | 80 | 50h | P | 112 | 70h | p |
| 49 | 31h | 1 | 81 | 51h | Q | 113 | 71h | q |
| 50 | 32h | 2 | 82 | 52h | R | 114 | 72h | r |
| 51 | 33h | 3 | 83 | 53h | S | 115 | 73h | s |
| 52 | 34h | 4 | 84 | 54h | T | 116 | 74h | t |
| 53 | 35h | 5 | 85 | 55h | U | 117 | 75h | u |
| 54 | 36h | 6 | 86 | 56h | V | 118 | 76h | v |
| 55 | 37h | 7 | 87 | 57h | W | 119 | 77h | w |
| 56 | 38h | 8 | 88 | 58h | X | 120 | 78h | x |
| 57 | 39h | 9 | 89 | 59h | Y | 121 | 79h | y |
| 58 | 3Ah | : | 90 | 5Ah | Z | 122 | 7Ah | z |
| 59 | 3Bh | ; | 91 | 5Bh | [ | 123 | 7Bh | { |
| 60 | 3Ch | < | 92 | 5Ch | \ | 124 | 7Ch | \| |
| 61 | 3Dh | = | 93 | 5Dh | ] | 125 | 7Dh | } |
| 62 | 3Eh | > | 94 | 5Eh | ^ | 126 | 7Eh | ~ |
| 63 | 3Fh | ? | 95 | 5Fh | _ | | | |

theASCIIcode.com.ar

Extended ASCII characters

| DEC | HEX | Simbolo | DEC | HEX | Simbolo | DEC | HEX | Simbolo | DEC | HEX | Simbolo |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 128 | 80h | Ç | 160 | A0h | á | 192 | C0h | └ | 224 | E0h | Ó |
| 129 | 81h | ü | 161 | A1h | í | 193 | C1h | ┴ | 225 | E1h | ß |
| 130 | 82h | é | 162 | A2h | ó | 194 | C2h | ┬ | 226 | E2h | Ô |
| 131 | 83h | â | 163 | A3h | ú | 195 | C3h | ├ | 227 | E3h | Ò |
| 132 | 84h | ä | 164 | A4h | ñ | 196 | C4h | ─ | 228 | E4h | õ |
| 133 | 85h | à | 165 | A5h | Ñ | 197 | C5h | ┼ | 229 | E5h | Õ |
| 134 | 86h | å | 166 | A6h | ª | 198 | C6h | ã | 230 | E6h | µ |
| 135 | 87h | ç | 167 | A7h | º | 199 | C7h | Ã | 231 | E7h | þ |
| 136 | 88h | ê | 168 | A8h | ¿ | 200 | C8h | ╚ | 232 | E8h | Þ |
| 137 | 89h | ë | 169 | A9h | ® | 201 | C9h | ╔ | 233 | E9h | Ú |
| 138 | 8Ah | è | 170 | AAh | ¬ | 202 | CAh | ╩ | 234 | EAh | Û |
| 139 | 8Bh | ï | 171 | ABh | ½ | 203 | CBh | ╦ | 235 | EBh | Ù |
| 140 | 8Ch | î | 172 | ACh | ¼ | 204 | CCh | ╠ | 236 | ECh | ý |
| 141 | 8Dh | ì | 173 | ADh | ¡ | 205 | CDh | ═ | 237 | EDh | Ý |
| 142 | 8Eh | Ä | 174 | AEh | « | 206 | CEh | ╬ | 238 | EEh | ¯ |
| 143 | 8Fh | Å | 175 | AFh | » | 207 | CFh | ¤ | 239 | EFh | ´ |
| 144 | 90h | É | 176 | B0h | ░ | 208 | D0h | ð | 240 | F0h | |
| 145 | 91h | æ | 177 | B1h | ▒ | 209 | D1h | Ð | 241 | F1h | ± |
| 146 | 92h | Æ | 178 | B2h | ▓ | 210 | D2h | Ê | 242 | F2h | ‗ |
| 147 | 93h | ô | 179 | B3h | │ | 211 | D3h | Ë | 243 | F3h | ¾ |
| 148 | 94h | ö | 180 | B4h | ┤ | 212 | D4h | È | 244 | F4h | ¶ |
| 149 | 95h | ò | 181 | B5h | Á | 213 | D5h | ı | 245 | F5h | § |
| 150 | 96h | û | 182 | B6h | Â | 214 | D6h | Í | 246 | F6h | ÷ |
| 151 | 97h | ù | 183 | B7h | À | 215 | D7h | Î | 247 | F7h | ¸ |
| 152 | 98h | ÿ | 184 | B8h | © | 216 | D8h | Ï | 248 | F8h | ° |
| 153 | 99h | Ö | 185 | B9h | ╣ | 217 | D9h | ┘ | 249 | F9h | ¨ |
| 154 | 9Ah | Ü | 186 | BAh | ║ | 218 | DAh | ┌ | 250 | FAh | · |
| 155 | 9Bh | ø | 187 | BBh | ╗ | 219 | DBh | █ | 251 | FBh | ¹ |
| 156 | 9Ch | £ | 188 | BCh | ╝ | 220 | DCh | ▄ | 252 | FCh | ³ |
| 157 | 9Dh | Ø | 189 | BDh | ¢ | 221 | DDh | ¦ | 253 | FDh | ² |
| 158 | 9Eh | × | 190 | BEh | ¥ | 222 | DEh | Ì | 254 | FEh | ■ |
| 159 | 9Fh | ƒ | 191 | BFh | ┐ | 223 | DFh | ▀ | 255 | FFh | |

**Gambar 1.** ASCII
(Sumber: https://wawanlaksito.wordpress.com/)

## 2.2 Cara Kerja Algoritma *Selection Sort*

Metode pemilihan dan penyortiran data atau *selection sort* yang digunakan adalah *ascending order*, yaitu diurutkan dari nilai terendah sampai nilai tertinggi pada data mahasiswa dan *descending order*, yaitu diurutkan dari nilai tertinggi sampai nilai terendah pada data mahasiswa. Dalam hal ini, akademik dapat secara acak memasukkan data mahasiswa.
Konsep Algoritma *Selection Sort* ini bekerja sebagai berikut:

1. Mulai dengan larik data yang akan diurutkan.
2. Tentukan indeks pertama sebagai posisi awal untuk mencari elemen terkecil dalam larik. Inisialisasi indeks pertama dengan nilai 0.
3. Cari elemen terkecil dalam sisa larik yang belum diurutkan. Untuk melakukan ini, lakukan langkah-langkah berikut.
   a. Bandingkan elemen pada indeks pertama (posisi awal pencarian) dengan setiap elemen dalam sisa larik, mulai dari indeks pertama + 1 hingga indeks terakhir.
   b. Jika elemen yang dibandingkan lebih kecil dari elemen pada indeks pertama, catat indeks yang lebih kecil tersebut sebagai indeks terkecil.
   c. Setelah mencari seluruh elemen, tukar elemen terkecil dengan elemen pada indeks pertama.
4. Pindahkan indeks penanda ke elemen kedua dalam larik, dan lakukan langkah (c) kembali untuk mencari elemen terkecil dalam sisa larik yang belum diurutkan
5. Ulangi langkah-langkah di atas untuk sisa daftar dimulai dengan tempat kedua.

Merujuk pada (Lasriana & Gunaryati, 2022) algoritma ini digunakan untuk mengurutkan data secara *ascending* dan *descending* dimulai dari data awal sampai data ke (n - 1). Jika ada (n) data dan mulai dari data ke 0 sampai (n - 1) maka proses pengurutan datanya sebagai berikut.

1. Mencari data minimum pada urutan (j = 0) sampai (n – 1).
2. Jika ditemukan data minimum, lakukan pertukaran posisi dengan data di posisi (i).
3. Proses 1 dan 2 dilakukan berulang-ulang dengan (j = j + 1) sampai (j = n – 1).

Kita dapat mengklasifikasikan daftar secara efektif, dalam dua bagian, yang diurutkan, akan diperoleh dengan menyortir dari kiri ke kanan dan dilakukan pada awal dan akhir bagian daftar yang disortir elemen yang akan dilakukan penyortiran (Aura D.E. & Otik, 2023).

Berikut *pseudocode* algoritma *selection sort* dalam mengurutkan bilangan acak dari bilangan terkecil ke bilangan terbesar dengan metode *selection sort*.

Deklarasi:

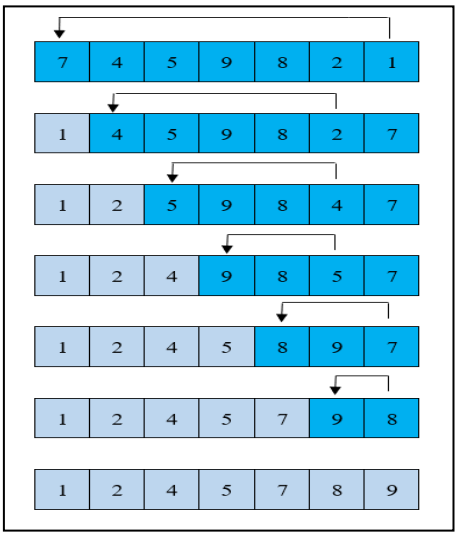
1. n = sebuah *integer* yang menyimpan panjang dari daftar {data}.
2. i = *integer* {Pencacah jumlah tahapan}.
3. j = *integer* {Pencacah untuk langkah pada tiap tahapan}.
4. min_idx = *integer* {Variabel sementara untuk pertukaran}.

Deskripsi:

```
SelectionSort (data, key):
    n = length of data
    for i from 0 to n-1:
        min_idx = ia
        for j from i + 1 to n-1:
            if data[min_idx][key] > data[j][key]:
                min_idx = j
        swap data[i] with data[min_idx]
```

### 1.1.1 Pengurutan Secara *Ascending*

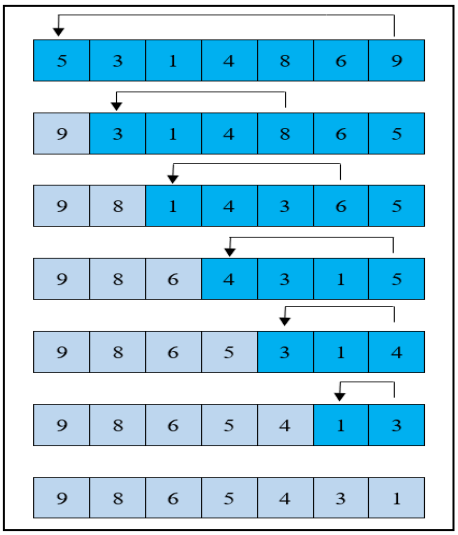
Contoh pada **Gambar 2**, terdapat 6 data acak yaitu 7, 4, 5, 9, 8, 2, dan 1 dapat diurutkan menjadi 1, 2, 4, 5, 7, 8, dan 9. Berikut adalah ilustrasi pengurutan data algoritma *selection sort* secara *ascending*:




**Gambar 2.** Ilustrasi *Selection sort Ascending*

### 1.1.2 Pengurutan Secara *Descending*

Contoh pada **Gambar 3**, terdapat 6 data acak yaitu 5, 3, 1, 4, 8, 6, dan 9 dapat diurutkan menjadi 9, 8, 6, 5, 4, 3, dan 1. Berikut adalah ilustrasi pengurutan data algoritma *selection sort* secara *descending*.




**Gambar 3.** Ilustrasi *Selection sort Descending*

## 3 HASIL DAN PEMBAHASAN

Implementasi algoritma *selection sort* dengan menggunakan bahasa pemrograman *Python* dengan menggunakan Visual Studio Code. Program dan *output* pengurutan data mahasiswa menggunakan *selection sort* secara *ascending* dan *descending* yang hanya dikelompokkan berdasarkan NIM mahasiswa dan nama. Semua pengurutan yang dilakukan dalam penelitian ini menggunakan data pada **Gambar 4**.

Data Awal

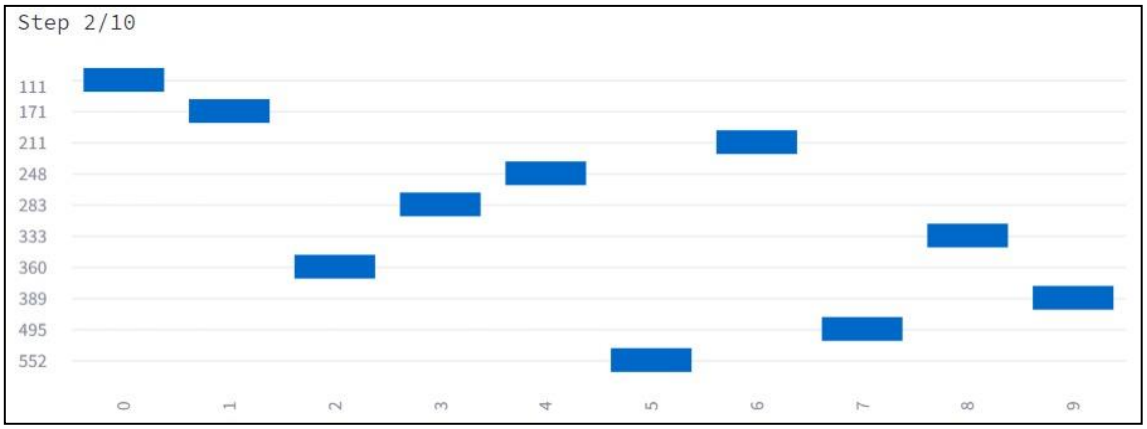
| | semester | kelas | nim | nama | status |
|---|---|---|---|---|---|
| 0 | 4 | B | 283 | MUHAMMAD DAFFA FEBRIYAN | Aktif |
| 1 | 4 | E | 211 | IRFAN SABRIAN FADHILLAH | Aktif |
| 2 | 4 | A | 360 | ERICK ERDIANSYAH | Aktif |
| 3 | 6 | D | 111 | NABILA | Aktif |
| 4 | 4 | D | 248 | MUHAMMAD SIGIT FIRMANSYAH | Aktif |
| 5 | 2 | E | 552 | FALIH PANGESTU | Aktif |
| 6 | 6 | C | 171 | MARIA UTRI | Aktif |
| 7 | 2 | B | 495 | REFI SULISTIAWATI | Tidak Aktif |
| 8 | 4 | E | 333 | HAFIZH FATHIN ADMAJA | Aktif |
| 9 | 4 | C | 389 | FAIZ TAUFIK | Aktif |

**Gambar 4.** Data Mahasiswa Acak

### 3.1 Hasil Pengurutan *Ascending*

3.1.1. Berdasarkan NIM

Pada pengujian algoritma *selection sort*, didapatkan rata-rata waktu pengurutan yang dilakukan sebanyak 5 kali yaitu 0,556 detik. Visualisasi salah satu proses pengurutan NIM mahasiswa dapat dilihat pada **Gambar 5**.




**Gambar 5.** Proses Visualisasi Pengurutan NIM Mahasiswa Secara *Ascending*

Pada **Gambar 5** menunjukkan proses pengurutan data mahassiwa berdasarkan NIM, pada visualisasi tersebut terlihat bahwa sedang berjalan *step* 2 dari 10 step pada proses pengurutan. Pada gambar tersebut terlihat tiga digit angka adalah digit terakhir dari NIM, angka 0 sampai 9 adalah nomor indeks dan bar berwarna biru adalah posisi awal data sebelum di urutkan. Pada saat proses pengurutan bar berwarna biru akan berpindah sesuai dengan tahapan algoritma *selection sort.* Hasil dari pengurutan ditampilakan dalam bentuk tabel seperti dapat dilihat pada **Gambar 6**.

**Sorted Data**

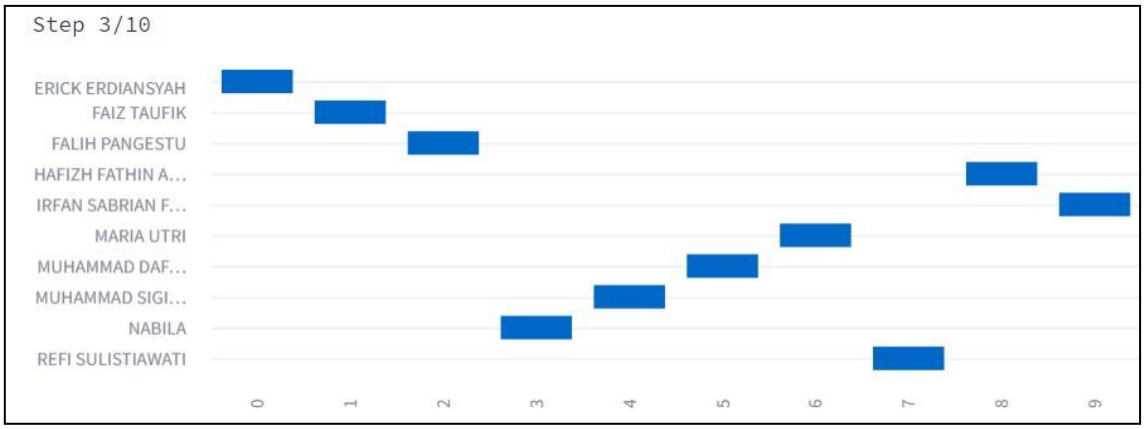
| | semester | kelas | nim | nama | status |
|---|---|---|---|---|---|
| 0 | 6 | D | 111 | NABILA | Aktif |
| 1 | 6 | C | 171 | MARIA UTRI | Aktif |
| 2 | 4 | E | 211 | IRFAN SABRIAN FADHILLAH | Aktif |
| 3 | 4 | D | 248 | MUHAMMAD SIGIT FIRMANSYAH | Aktif |
| 4 | 4 | B | 283 | MUHAMMAD DAFFA FEBRIYAN | Aktif |
| 5 | 4 | E | 333 | HAFIZH FATHIN ADMAJA | Aktif |
| 6 | 4 | A | 360 | ERICK ERDIANSYAH | Aktif |
| 7 | 4 | C | 389 | FAIZ TAUFIK | Aktif |
| 8 | 2 | B | 495 | REFI SULISTIAWATI | Tidak Aktif |
| 9 | 2 | E | 552 | FALIH PANGESTU | Aktif |

**Gambar 6.** Hasil Pengurutan NIM Mahasiswa Secara *Ascending*

Pada **Gambar 6** merupakan hasil pengurutan berdasarakan NIM yang diurutkan secara *ascending* dari NIM terkecil sampai ke NIM terbesar.

### 3.1.2. Berdasarkan Nama

Pada pengurutan berdasarkan nama, didapatkan rata-rata waktu yang diperlukan oleh algoritma *selection sort* adalah 0,5626 detik berdasarkan pengujian yang dilakukan sebanyak 5 kali. Visualilasi salah satu proses pengurutan nama secara *ascending* dapat dilihat pada **Gambar 7**.




**Gambar 7.** Proses Visualisasi Pengurutan Nama Acak Secara *Ascending*

**Gambar 7** memperlihatkan proses pengurutan data mahasiswa berdasarkan nama. Dalam visualisasi tersebut, proses pengurutan menunjukkan bahwa saat ini berada pada tahap ketiga dari sepuluh tahap keseluruhan. Gambar tersebut menampilkan nama mahasiswa, dengan angka 0 hingga 9 sebagai nomor indeks, serta batang berwarna biru yang menandakan posisi awal data sebelum diurutkan. Selama proses pengurutan, batang berwarna biru ini akan bergerak sesuai dengan langkah-langkah algoritma *selection sort*. Hasil akhir dari pengurutan ini disajikan dalam bentuk tabel seperti yang terlihat pada **Gambar 8.**

Sorted Data

| | semester | kelas | nim | nama | status |
|---|---|---|---|---|---|
| 0 | 4 | A | 360 | ERICK ERDIANSYAH | Aktif |
| 1 | 4 | C | 389 | FAIZ TAUFIK | Aktif |
| 2 | 2 | E | 552 | FALIH PANGESTU | Aktif |
| 3 | 4 | E | 333 | HAFIZH FATHIN ADMAJA | Aktif |
| 4 | 4 | E | 211 | IRFAN SABRIAN FADHILLAH | Aktif |
| 5 | 6 | C | 171 | MARIA UTRI | Aktif |
| 6 | 4 | B | 283 | MUHAMMAD DAFFA FEBRIYAN | Aktif |
| 7 | 4 | D | 248 | MUHAMMAD SIGIT FIRMANSYAH | Aktif |
| 8 | 6 | D | 111 | NABILA | Aktif |
| 9 | 2 | B | 495 | REFI SULISTIAWATI | Tidak Aktif |

**Gambar 8.** Hasil Pengurutan Nama Secara *Ascending*

Pada **Gambar 8** merupakan hasil pengurutan berdasarakan nama yang diurutkan secara *ascending* dari nama dengan awalan A sampai ke nama dengan awalan Z.

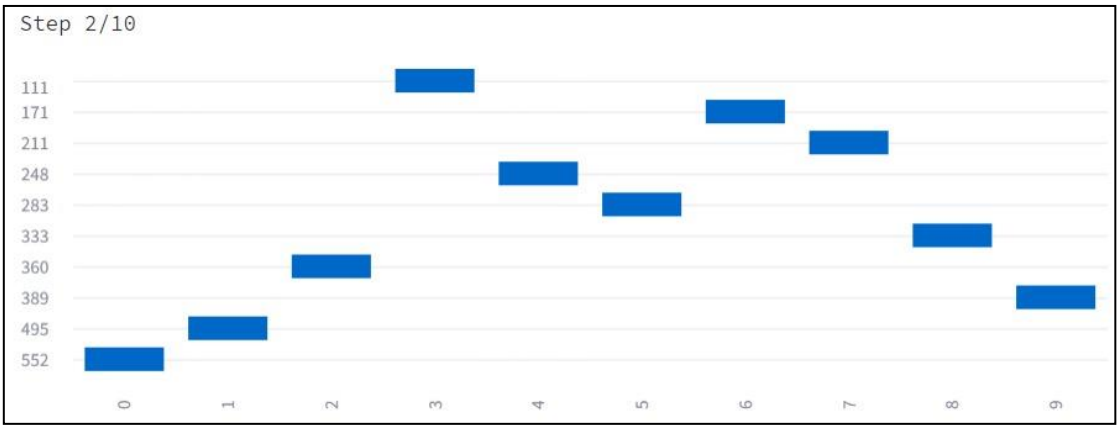
### 3.2 Hasil Pengurutan *Descending*

Jika pada *ascending* "`if int(data[min_idx][key]) > int(data[j][key])`", maka pada *descending* kebalikan dari *ascending* yaitu "`if int(data[min_idx][key]) < int(data[j][key])`".

Program dan *output* pengurutan data mahasiswa menggunakan *selection sort* secara *descending* yang hanya dikelompokkan berdasarkan NIM dan nama.

3.2.1. Berdasarkan NIM

Selama pengujian algoritma *selection sort*, ditemukan rata-rata waktu pengurutan yang dilakukan sebanyak 5 kali yaitu 0,5208 detik. **Gambar 9** menunjukkan salah satu proses pengurutan NIM mahasiswa.




**Gambar 9.** Proses Visualisasi Pengurutan NIM Secara *Descending*

Pengurutan data mahasiswa berdasarkan NIM divisualisasikan pada Gambar 9. Proses pengurutan menunjukkan bahwa saat ini berada pada tahap kedua dari total sepuluh tahap dalam visualisasi tersebut. Gambar menunjukkan tiga digit terakhir dari NIM, dengan angka yang terdiri dari 0 hingga 9, sebagai nomor indeks. Selain itu, batang berwarna biru menunjukkan posisi awal data sebelum diurutkan. Batang berwarna biru ini akan bergerak sesuai dengan langkah-langkah algoritma *selection sort* selama proses pengurutan. **Gambar 10** menunjukkan tabel hasil dari pengurutan ini.

Sorted Data

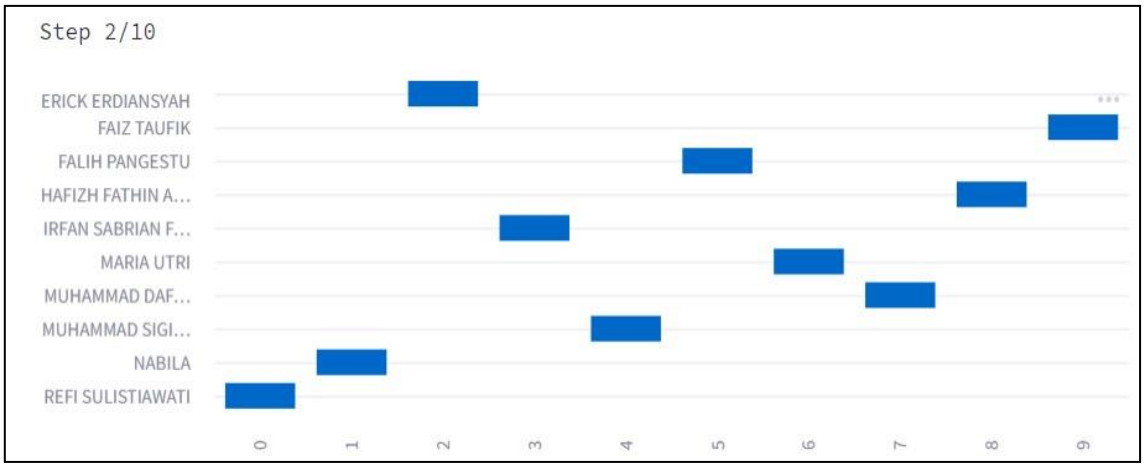
| | semester | kelas | nim | nama | status |
|---|---|---|---|---|---|
| 0 | 2 | E | 552 | FALIH PANGESTU | Aktif |
| 1 | 2 | B | 495 | REFI SULISTIAWATI | Tidak Aktif |
| 2 | 4 | C | 389 | FAIZ TAUFIK | Aktif |
| 3 | 4 | A | 360 | ERICK ERDIANSYAH | Aktif |
| 4 | 4 | E | 333 | HAFIZH FATHIN ADMAJA | Aktif |
| 5 | 4 | B | 283 | MUHAMMAD DAFFA FEBRIYAN | Aktif |
| 6 | 4 | D | 248 | MUHAMMAD SIGIT FIRMANSYAH | Aktif |
| 7 | 4 | E | 211 | IRFAN SABRIAN FADHILLAH | Aktif |
| 8 | 6 | C | 171 | MARIA UTRI | Aktif |
| 9 | 6 | D | 111 | NABILA | Aktif |

**Gambar 10.** Hasil Pengurutan NIM Secara *Descending*

Dapat dilihat pada **Gambar 10**, bahwa NIM mahasiswa telah diurutkan menjadi dari NIM yang terbesar sampai NIM yang terkecil.

3.2.2. Berdasarkan Nama

Algoritma *selection sort* menggunakan rata-rata waktu selama 0,5658 detik untuk pengurutan berdasarkan nama yang dilakukan sebanyak 5 kali. **Gambar 11** menunjukkan visualisasi salah satu proses pengurutan nama secara *descending*.




**Gambar 11.** Salah Satu Proses Visualisasi Pengurutan NIM Secara *Descending*

**Gambar 11** menampilkan proses pengurutan data mahasiswa berdasarkan nama. Visualisasi ini menunjukkan bahwa proses pengurutan berada pada tahap kedua dari sepuluh tahap keseluruhan. Gambar tersebut memperlihatkan nama mahasiswa, dengan angka 0 hingga 9 digunakan sebagai nomor indeks. Batang berwarna biru menunjukkan posisi awal data sebelum diurutkan dan akan bergerak sesuai dengan langkah-langkah algoritma *selection sort* selama proses pengurutan berlangsung. Hasil akhir dari proses pengurutan ini ditampilkan dalam bentuk tabel pada **Gambar 12**.

Sorted Data

| | semester | kelas | nim | nama | status |
|---|---|---|---|---|---|
| 0 | 4 | A | 360 | ERICK ERDIANSYAH | Aktif |
| 1 | 4 | C | 389 | FAIZ TAUFIK | Aktif |
| 2 | 2 | E | 552 | FALIH PANGESTU | Aktif |
| 3 | 4 | E | 333 | HAFIZH FATHIN ADMAJA | Aktif |
| 4 | 4 | E | 211 | IRFAN SABRIAN FADHILLAH | Aktif |
| 5 | 6 | C | 171 | MARIA UTRI | Aktif |
| 6 | 4 | B | 283 | MUHAMMAD DAFFA FEBRIYAN | Aktif |
| 7 | 4 | D | 248 | MUHAMMAD SIGIT FIRMANSYAH | Aktif |
| 8 | 6 | D | 111 | NABILA | Aktif |
| 9 | 2 | B | 495 | REFI SULISTIAWATI | Tidak Aktif |

**Gambar 12.** Hasil Pengurutan Nama Secara *Descending*

Pada **Gambar 12** merupakan hasil pengurutan berdasarakan nama yang diurutkan secara *descending* dari nama dengan awalan Z sampai ke nama dengan awalan A.

## 4 KESIMPULAN

Penelitian ini telah berhasil mengimplementasikan algoritma *selection sort* untuk mengurutkan data mahasiswa berdasarkan nama, NIM, kelas, semester, dan status menggunakan bahasa pemrograman Python. Temuan utama dari penelitian ini menunjukkan bahwa algoritma *selection sort* mampu mengurutkan data mahasiswa dengan benar dan konsisten, meskipun efisiensinya menurun pada data *set* yang lebih besar dibandingkan dengan algoritma *sorting*

lain. Pengurutan data mahasiswa secara efisien sangat penting dalam mengelola data akademik, memudahkan pencarian, penyusunan laporan, serta pengelolaan administrasi dan akademik di Program Studi D3 Teknik Informatika, Politeknik Negeri Pontianak. Penelitian ini menjawab hipotesis bahwa *Selection Sort* dapat diterapkan untuk pengurutan data mahasiswa dan mencapai tujuan penelitian untuk mengevaluasi kinerjanya.

## UCAPAN TERIMAKASIH

Saya mengucapkan terima kasih yang sebesar-besarnya kepada Program Studi D3 Teknik Informatika, Politeknik Negeri Pontianak atas dukungan dan izin yang diberikan dalam pelaksanaan penelitian ini. Terima kasih juga kepada pihak-pihak yang telah memberikan bantuan dalam pengumpulan data mahasiswa serta kepada dosen pembimbing yang telah memberikan bimbingan dan konsultasi yang berharga sepanjang penelitian ini. Dukungan dan bantuan dari semua pihak tersebut sangatlah berarti dan telah memungkinkan penelitian ini dapat diselesaikan dengan baik.

## DAFTAR PUSTAKA

Aura D.E., F., & Otik, D. (2023). Penggunaan Algoritma Selection Sort Untuk Menentukan Nilai Tertinggi Siswa. *JuSiTik : Jurnal Sistem Dan Teknologi Informasi Komunikasi*, *6*(2), 23–26. https://doi.org/10.32524/jusitik.v6i2.961

Lasriana, L., & Gunaryati, A. (2022). Sistem Informasi Apotek Berbasis Web Menggunakan Algoritma Sequential Search Dan Selection Sort. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *7*(2), 392–401. https://doi.org/10.29100/jipi.v7i2.2709

Rizki Saputra, D. Y., Andryana, S., & Sholihati, I. D. (2021). ANALISIS PERBANDINGAN ALGORITMA BUBBLE SORT DAN SELECTION SORT PADA SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN TEMPAT KOST BERBASIS IOS (IPHONE OPERATING SYSTEM). *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *6*(2), 318–324. https://doi.org/10.29100/jipi.v6i2.2015

Sandria, Y. A., Nurhayoto, M. R. A., Ramadhani, L., Harefa, R. S., & Syahputra, A. (2022). Penerapan Algoritma Selection Sort untuk Melakukan Pengurutan Data dalam Bahasa Pemrograman PHP. *Hello World Jurnal Ilmu Komputer*, *1*(4), 190–194. https://doi.org/10.56211/helloworld.v1i4.187

Sari, I. P., Ramadhani, F., & Sulaiman, O. K. (2023). Implementation of the Selection Sort Algorithm to Sort Data in PHP Programming Language. *Journal of Computer Science, Information Technology and Telecommunication Engineering*, *4*(1), 377–381. https://doi.org/10.30596/jcositte.v4i1.14362

Sari, N., Gunawan, W. A., Sari, P. K., Zikri, I., & Syahputra, A. (2022). Analisis Algoritma Bubble Sort Secara *Ascending* Dan *Descending* Serta Implementasinya Dengan Menggunakan Bahasa Pemrograman Java. *ADI Bisnis Digital Interdisiplin Jurnal*, *3*(1), 16–23. https://doi.org/10.34306/abdi.v3i1.625

Sunandar, E., & Indrianto, I. (2020). Implementasi Algoritma Bubble Sort Terhadap 2 Buah Model Varian Pengurutan Data Menggunakan Bahasa Program Java. *Petir*, *13*(2), 255–265. https://doi.org/10.33322/petir.v13i2.1008

Wisda, Y. (2022). IMPLEMENTASI METODE SELECTION SORT PADA SISTEM INFORMASI RETENSI DOKUMEN REKAM MEDIS DI KLINIK PKU MUHAMMADIYAH KARANGANOM, KLATEN. *Elkom : Jurnal Elektronika Dan Komputer*, *15*(1), 108–117. https://doi.org/10.51903/elkom.v15i1.789

Yanti, F., & Sita Eriana, E. (2024). *Algoritma Sorting Dan Searching Penerbit*. CV.Eureka Media Aksara.

Yovita, C., Nasution, K., & Haramaini, T. (2023). Penerapan Algoritma Naive Bayes dan

Selection Sort pada Penilaian Kuis di Aplikasi Pembelajaran Pemrograman Java dan PHP. *Hello World Jurnal Ilmu Komputer*, *2*(3), 120–128. https://doi.org/10.56211/helloworld.v2i3.278