

# Pencarian Layout Keyboard dengan Travel Distance Terkecil Menggunakan Algoritma Genetik

I Nengah Oka Darmayasa<sup>a1</sup>, Ngurah Agus Sanjaya ER<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Pengetahuan Alam, Universitas Udayana  
Badung, Bali, Indonesia

<sup>1</sup>okadarmayasa10@gmail.com

<sup>2</sup>agus\_sanjaya@unud.ac.id

## Abstract

Mengetik merupakan kegiatan yang selalu dilakukan beberapa kalangan masyarakat. Salah satu parameter untuk mengetik dengan efisien adalah meminimalkan pergerakan jari untuk mengetik suatu huruf, yang disebut sebagai *travel distance*. Metode yang dapat digunakan untuk mencari *layout keyboard* dengan *travel distance* terkecil secara efisien adalah dengan men-*generate layout* menggunakan algoritma genetika. Dengan menggunakan algoritma genetika, setiap huruf pada *keyboard* akan berperan sebagai gen, *layout keyboard* berperan sebagai kromosom, dan kumpulan beberapa *layout keyboard* sebagai populasi. Setiap kromosom akan dilakukan kawin silang dengan kromosom lain dengan mengambil beberapa gen dari masing-masing kromosom lalu menggabungkannya menjadi kromosom baru. Cara yang digunakan untuk menentukan kromosom mana yang digunakan untuk melakukan kawin silang dapat menggunakan *travel distance* dari tiap kromosom. Semakin kecil *travel distance*, maka semakin besar kemungkinan kromosom tersebut untuk dikawinsilangkan. *Keyboard* yang paling banyak digunakan saat ini masihlah *keyboard* dengan *layout* bernama "QWERTY". Saat dibandingkan untuk mengetik data kalimat berbahasa Indonesia dari TED-Multilingual-Parallel-Corpus, *keyboard* QWERTY memiliki *travel distance* yang lebih besar, sekitar 9.311,735, jika dibandingkan dengan *layout keyboard* yang didapat dari metode algoritma genetika yang memiliki *travel distance* sebesar 5.747,246, sekitar 62% lebih besar.

**Keywords:** Genetic Algorithm, Keyboard Layout, Keyboard, Machine Learning, Algoritma Genetik

## 1. Introduction

Mengetik merupakan kegiatan yang semakin banyak dilakukan orang-orang terutama setelah adanya pandemi Covid-19 yang membuat kebanyakan orang beraktivitas secara daring [1]. Dengan bertambahnya waktu yang digunakan untuk beraktivitas secara daring, tentu akan meningkatkan kesempatan untuk mengetik [2]. Bagi orang-orang yang menghabiskan kebanyakan waktunya mengetik, mengetahui teknik baru untuk mengetik dengan lebih efisien tentunya merupakan sesuatu yang diinginkan. Salah satu cara yang dapat dilakukan untuk mengefisienkan kegiatan mengetik adalah dengan mengganti *layout* dari *keyboard* yang digunakan menjadi *layout* yang memiliki *travel distance* terkecil, yaitu sebuah skor yang menandakan panjangnya pergerakan jari untuk mengetik suatu teks.

Banyaknya karakter yang ada di dalam sebuah *keyboard* dapat menyebabkan ketidakefisienan proses mengetik apabila penempatan setiap karakter berada pada posisi yang tidak optimal. Misalkan saja huruf-huruf yang sering digunakan untuk mengetik diletakkan pada tempat yang sulit terjangkau akan menyebabkan performa mengetik yang lebih rendah dibandingkan apabila seandainya karakter-karakter tersebut diletakkan di tempat yang mudah dijangkau.

Pada penelitian [3], mencari *layout keyboard* yang ergonomik dengan melihat frekuensi munculnya 3.000 kata paling umum dalam bahasa Inggris menggunakan algoritma genetika. Pada penelitian [4], mencari *layout keyboard* untuk bahasa Portugis menggunakan algoritma genetika. Dari kedua penelitian tersebut, hasil yang didapatkan adalah *layout keyboard* QWERTY memiliki performa, dengan parameter tertentu pada masing-masing penelitian, yang lebih rendah kualitasnya dibandingkan dengan *layout keyboard* hasil pencariannya.

Penelitian [5] dan [6] merupakan penelitian yang mencari sebuah *layout keyboard* lalu membandingkan *travel distance*-nya terhadap *layout keyboard* QWERTY dan Dvorak. Metode yang digunakan pada [6] adalah dengan algoritma genetika. Penghitungan *travel distance* yang digunakan adalah dengan

menghitung Euclidean Distance dari posisi jari sekarang menuju posisi huruf yang jari tersebut perlu tekan. Dari percobaannya, didapat hasil sebuah *layout keyboard* yang 32,66% lebih baik dibanding *layout keyboard* QWERTY dan 15,78% lebih baik dibandingkan *layout keyboard* Dvorak.

## 2. Research Methods

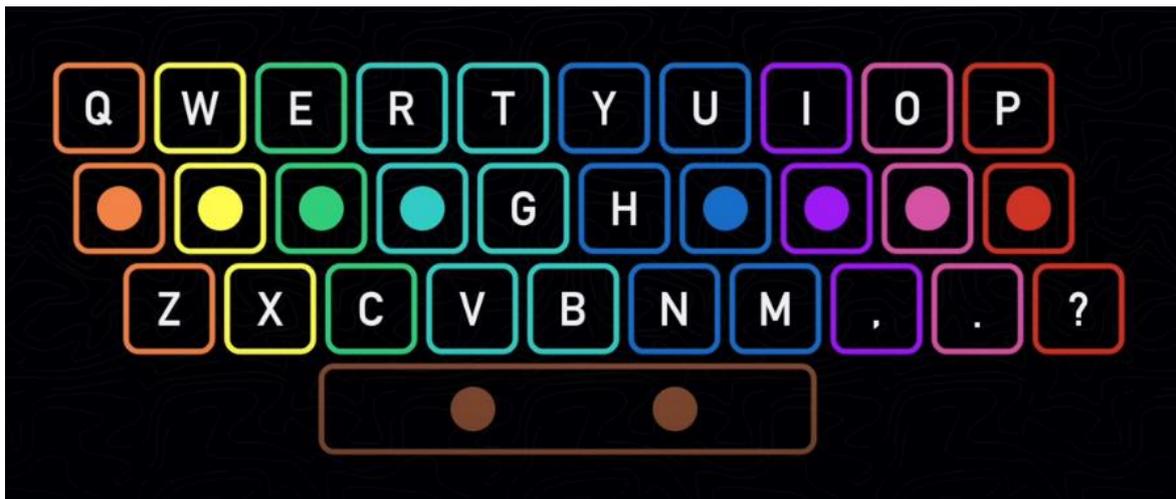
### 2.1 Identifikasi Masalah

Dalam penelitian ini, masalah yang ingin diselesaikan adalah mengenai pencarian sebuah *layout keyboard* yang cocok digunakan untuk mengetik kalimat-kalimat dalam bahasa Indonesia. Cara yang digunakan untuk mencari *layout keyboard* tersebut adalah dengan menggunakan algoritma genetika.

Mengetik 10 jari merupakan sebuah teknik mengetik dengan menggunakan 10 jari yang setiap jainya berperan menekan huruf tertentu. Pada *keyboard* QWERTY, setiap jari memiliki area jangkauannya masing-masing, yaitu:

- a. Kelingking kiri = q, a, z
- b. Manis kiri = w, s, x
- c. Tengah kiri = e, d, c
- d. Telunjuk kiri = r, f, v, t, g, b
- e. Jempol kiri = spasi
- f. Jempol kanan = spasi
- g. Telunjuk kanan = y, h, n, u, j, m
- h. Tengah kanan = i, k, ,
- i. Manis kanan = o, l, .
- j. Kelingking kanan = p, ;, /

Setiap jari ini memiliki posisi awalnya masing-masing, misalnya pada *layout* QWERTY, seperti jari kelingking kiri, manis kiri, tengah kiri, telunjuk kiri, jempol kiri, jempol kanan, telunjuk kanan, tengah kanan, manis kanan, dan kelingking kanan terletak ada huruf a, s, d, f, spasi, spasi, j, k, l, dan ; secara terurut. Artinya, saat ingin mengetik kata “makan”, maka jari-jari yang perlu bergerak adalah jari-jari yang area jangkauannya ada pada setiap huruf pada kata “makan”, yaitu jari telunjuk kanan untuk huruf “m” yang bergerak ke bawah, jari kelingking kiri untuk huruf “a” dengan tidak bergerak, jari tengah kanan untuk huruf “k” dengan tidak bergerak, dan jari telunjuk kanan untuk huruf “n” yang bergerak ke kiri karena sebelumnya sedang berada di huruf “m” yang posisinya di sebelah kanan huruf “n”.



**Gambar 1.** Area Jangkauan Tiap Jari Pada *Layout* QWERTY

Namun, tidak semua *layout keyboard* mengharuskan posisi tiap jari pada huruf tertentu. Yang dijadikan patokan adalah posisinya pada *layout* tersebut, bukan hurufnya. Jadi, pada dasarnya, setiap jari akan memiliki area jangkauan sesuai dengan warna-warna yang berbeda seperti pada gambar 1.

Setiap pergerakan akan dikenai bobot yang berbeda-beda tergantung gerakannya. Nantinya, jumlah dari semua bobot untuk mengetik suatu teks dengan *layout keyboard* tertentu tersebut yang akan dijadikan *fitness value* dalam algoritma genetika.

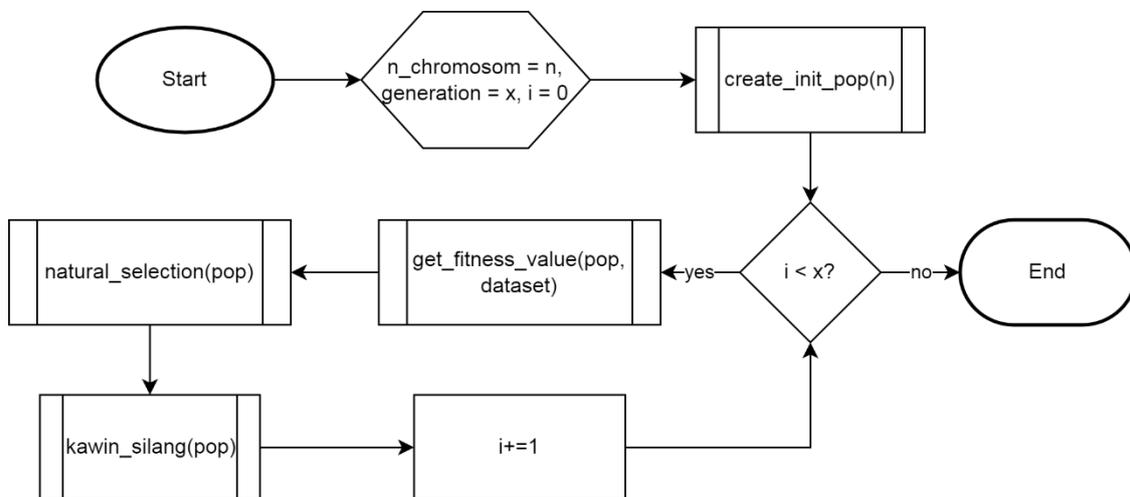
Algoritma genetika dapat digunakan untuk mencari *layout keyboard*. Dalam algoritma genetika, akan ada beberapa komponen, yaitu gen, kromosom, populasi, dan *fitness value*. Kumpulan gen akan membentuk sebuah kromosom. Dalam hal ini, gen merupakan setiap huruf pada *layout* dan kromosom merupakan *layout keyboard*. Kumpulan dari beberapa kromosom dinamakan sebagai populasi.

Sedangkan *fitness value* merupakan sebuah skor yang dimiliki oleh sebuah kromosom yang digunakan untuk menentukan kemungkinan kromosom tersebut untuk tetap “hidup” dalam rangkaian alur algoritma genetik ini. Dalam kasus ini, *fitness value* yang digunakan adalah *travel distance* dari setiap kromosom saat digunakan untuk mengetik sebuah teks tertentu.

## 2.2 Data Penelitian

Penelitian ini akan menggunakan data berupa kumpulan kalimat dalam bahasa Indonesia yang didapat dari TED-Multilingual-Parallel-Corpus. Dataset yang digunakan terdiri dari 2.000 kata.

## 2.3 Desain Penelitian



Gambar 2. Flowchart Algoritma Genetik

Penelitian akan dimulai dari pengambilan data. Data yang telah didapat lalu dilakukan *preprocessing* seperti menghilangkan beberapa karakter-karakter yang tidak penting seperti angka, “-”, “+”, “=”, “\_”, “[”, “]”, “{”, “}”, “\”, dan “|”. Setelah dilakukan *preprocessing*, maka selanjutnya adalah menentukan jumlah kromosom yang ada di dalam sebuah populasi dan juga menentukan banyak generasi. Selanjutnya adalah membuat populasi pertama secara *random* yang terdiri dari *n* kromosom. Lalu akan dilakukan pengulangan sebanyak generasi dengan pertama mencari *fitness value* tiap kromosom-nya dengan menggunakan data yang telah disiapkan sebelumnya. Setelah itu, 10% kromosom yang memiliki *fitness value* terkecil akan tetap berada di dalam populasi tersebut. Sedangkan, 90% kromosom sisanya akan dilakukan kawin silang dengan membagi gen dalam kromosom menjadi 2 bagian lalu menggabungkan masing-masing bagian dengan bagian lain dari kromosom yang lain. Dari proses ini akan menghasilkan kromosom baru. Hasil dari kromosom baru ini akan memiliki *trait* bagus yang didapat dari *parent*-nya, yaitu susunan huruf-huruf dalam *layout keyboard* yang berakibat pada *fitness value* dari kromosom-kromosom hasil kawin silang ini cenderung bagus, bahkan lebih bagus dari *parent*-nya. Selain kawin silang ini, pada penelitian ini juga menerapkan sebuah konsep mutasi, yaitu dalam persentase 10%, akan terjadi mutasi yang akan menukarkan satu pasang gen dari sebuah kromosom. Jumlah kromosom dalam suatu populasi akan selalu tetap. Karena dari kawin silang ini menghasilkan kromosom baru, maka ada kromosom yang perlu dibuang, yaitu kromosom yang tidak dikawinsilangkan yang akan digantikan posisinya dengan kromosom hasil kawin silang yang memiliki *fitness value* terbaik.

## 2.4 Algoritma Genetika Untuk Pencarian *Layout Keyboard*

Algoritma genetika dapat digunakan untuk melakukan pencarian *layout keyboard* yang efektif. Konsep utama yang ada pada algoritma ini adalah dengan melakukan kawin silang dan seleksi alam terhadap suatu komponen yang disebut sebagai kromosom [7]. Di dalam setiap kromosom terdiri dari beberapa bagian kecil yang dinamakan gen. Nantinya, setiap kromosom ini akan memiliki sebuah *fitness value*, yaitu sebuah skor yang akan menentukan kemungkinan kromosom ini untuk diseleksi. Yang dimaksud dengan seleksi adalah pengeluaran sebuah kromosom dari suatu kumpulan kromosom yang dinamakan populasi. Semakin kecil *fitness value* sebuah kromosom, maka semakin besar peluang kromosom tersebut untuk keluar dari sebuah populasi. Dari kromosom yang berhasil untuk hidup, akan dilakukan kawin silang antar kromosom tersebut. Perkawinan silang akan menggunakan beberapa gen yang diambil dari 2 kromosom, yang disebut *parent*, yang nantinya akan digabung untuk membentuk suatu kromosom baru, yang disebut *child*.

## 2.5 Gen

Gen merupakan bagian yang membentuk sebuah kromosom. Dalam teori evolusi, gen ini yang membawa *trait* yang dimiliki oleh sebuah spesies. Sama halnya pada kasus ini, gen akan membawa kualitas yang memungkinkannya untuk tetap berada di dalam populasi, yaitu memiliki kualitas untuk memiliki *fitness value* yang paling baik (kecil). Dalam kasus ini, yang menjadi gen adalah setiap huruf yang ada di dalam suatu *layout keyboard*.

## 2.6 Kromosom

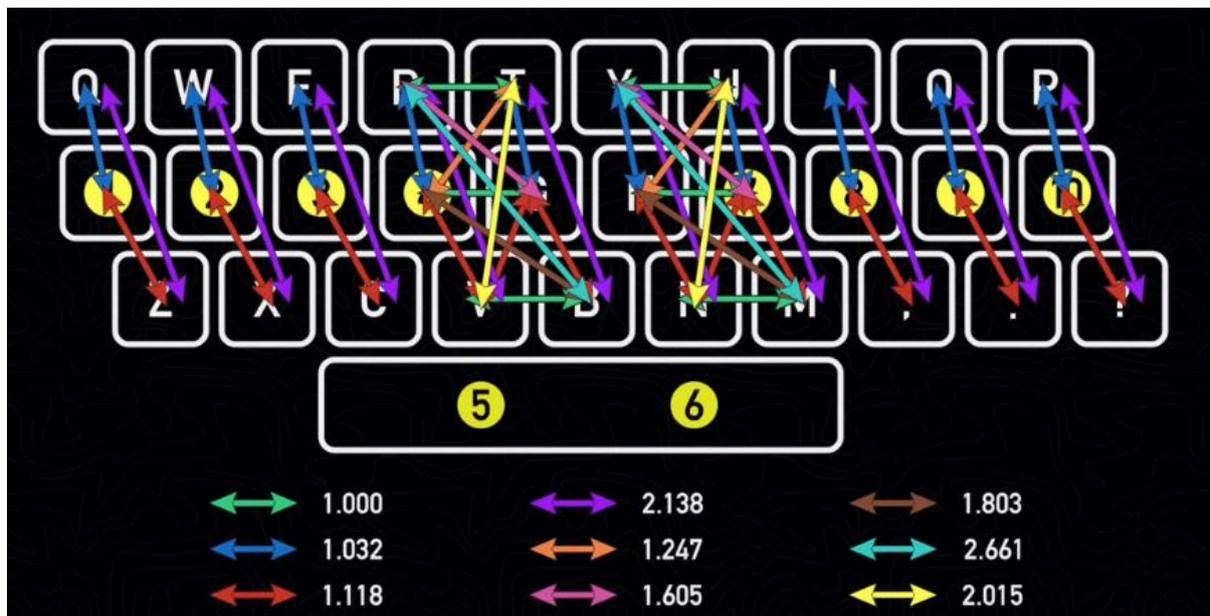
Kromosom merupakan kumpulan dari beberapa gen. Dalam kasus ini, yang menjadi kromosom dari sebuah *layout keyboard*. Nantinya, setiap kromosom ini akan dihitung *fitness value*-nya lalu dibandingkan dengan *fitness value* dari kromosom lain di dalam populasi yang sama. Setelah itu, kromosom yang memiliki *fitness value* yang baik akan dikawin silang dengan kromosom lain dengan membagi gennya menjadi 2 bagian. Hal ini akan membentuk kromosom baru dengan *trait* berkualitas baik yang didapat dari gen *parent*-nya.

## 2.7 Populasi

Populasi merupakan kumpulan kromosom. Jumlah kromosom di dalam suatu populasi akan selalu tetap. Hal ini akan memungkinkan terjadinya seleksi alam, yaitu keluarnya kromosom yang tidak mampu bersaing dari populasi dan digantikan dengan kromosom lain yang mampu bersaing. Indikator mampu bersaing yang digunakan adalah dengan membandingkan kromosom mana yang memiliki *fitness value* yang lebih baik. Semakin kecil *fitness value* maka semakin baik kualitasnya yang berujung pada semakin kecil kemungkinan kromosom tersebut untuk keluar dari populasi.

## 2.8 Perhitungan *fitness value*

*Fitness value* yang akan digunakan adalah besarnya bobot pergerakan jari tangan untuk mengetik dengan 10 jari. Untuk mengetik dengan 10 jari, setiap jari bertugas untuk menekan tombol-tombol tertentu. Karena jumlah tombol yang perlu ditekan lebih banyak dibandingkan dengan jumlah jari yang ada, maka tentunya jari-jari akan bergerak untuk menekan tombol-tombol yang menjadi area tugasnya.



Gambar 3. Bobot Pergerakan Tiap Jari

Posisi awal setiap jari ditandai pada lingkaran kuning. Setiap jari, kecuali jempol dan telunjuk, memiliki area jangkauan vertikal, yaitu ke atas, ke bawah, atau diam di tempat. Untuk jempol selalu bertugas untuk menekan tombol spasi. Sedangkan jari telunjuk memiliki area jangkauan yang lebih luas dibandingkan jari lainnya. Hal ini karena posisinya yang selalu berada di tengah dan juga yang paling fleksibel. Jari telunjuk bertugas untuk menekan tombol dengan area yang lebih luas yang menyebabkan jari telunjuk memiliki *range of motion* yang lebih beragam dibandingkan jari lainnya.

Gambar di atas merupakan bobot yang akan digunakan dalam setiap pergerakan jari. Baris paling bawah *keyboard* biasanya sedikit lebih ke kanan dibandingkan dengan baris paling atas relatif terhadap baris tengah, maka bobot untuk bergerak secara vertikal dari baris paling bawah akan memiliki bobot yang lebih besar dibandingkan dengan baris paling atas.

Misalkan  $l1, l2, l3, l4, l5$  merupakan jari kelingking kiri, manis kiri, tengah kiri, telunjuk kiri, dan jempol kiri, begitu juga  $k1, k2, k3, k4, k5$  untuk tangan kanan, maka untuk mencari *travel distance* sebuah *layout keyboard* saat digunakan untuk mengetik suatu teks F adalah seperti berikut:

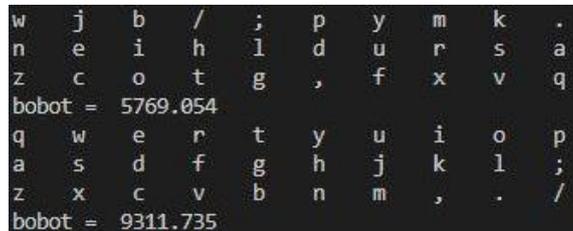
$$F = f(l1) + f(l2) + f(l3) + f(l4) + f(l5) + f(k1) + f(k2) + f(k3) + f(k4) + f(k5)$$

Dengan  $f$  merupakan fungsi yang menghitung jumlah dari bobot pergerakan setiap jari selama mengetik sebuah teks.

### 3. Result and Discussion

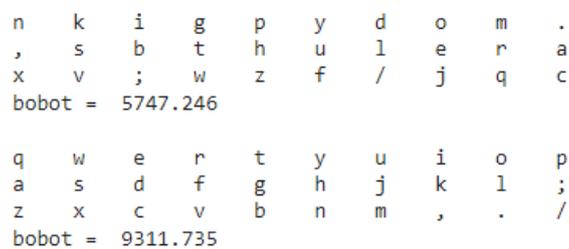
#### 3.1 Hasil

Dalam pengimplementasiannya, penelitian ini menggunakan populasi dengan jumlah kromosom sebanyak 10 dan iterasi generasinya sebanyak 10.000 kali. Percobaan ini dilakukan sebanyak 2 kali.



**Gambar 4.** Hasil Percobaan Pertama

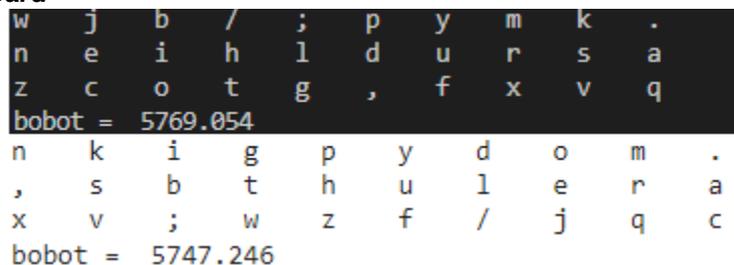
Seperti yang dapat dilihat pada gambar 3, didapatkan hasil sebuah *layout keyboard* yang memiliki bobot sekita 5.769,054 saat digunakan untuk mengetik teks yang ada dalam dataset yang digunakan. Sedangkan, di bawahnya merupaakn perbandingan pada *layout keyboard* QWERTY yang memiliki bobot sekitar 9.311,735 untuk mengetik dataset yang sama.



**Gambar 5.** Hasil Percobaan Kedua

Pada percobaan kedua, yang dapat dilihat pada gambar 4, terlihat hasil *layout keyboard* yang didapatkan memiliki perbedaan yang sangat sedikit dari segi bobot dibandingkan dengan hasil *layout keyboard* pada percobaan pertama. Hal ini karena dari algoritmanya sendiri bersifat konvergen, yaitu akan mengerucut ke suatu hasil tertentu, dalam hal ini *layout keyboard*.

#### 3.2 Layout Keyboard



**Gambar 6.** Layout Keyboard yang Dihasilkan

Gambar di atas merupakan 2 *layout keyboard* yang didapat menggunakan algoritma genetika.

#### 3.3 Analisis Layout Keyboard

Jika diperhatikan, kedua *layout keyboard* tersebut tidak memiliki kesamaan yang jelas terlihat, seperti huruf-huruf yang sama berada pada posisi yang sama. Hal ini terjadi karena alur dari program yang dibuat yang tidak melakukan pengembalian posisi jari ke posisi awal disaat mengetik. Artinya, dalam hasil tersebut beranggapan bahwa pengetikan setiap kata dilakukan terus menerus dengan posisi jari akan selalu berada di posisi di mana jari tersebut terakhir menekan sebuah tombol.

Selain itu, penyebab lain adalah karena indikator *fitness value* yang dipakai dalam penelitian ini hanya menggunakan faktor *travel distance*. Hal ini menyebabkan bahwa kedua *layout keyboard* tersebut saat digunakan untuk mengetik, akan memiliki *travel distance* yang lebih kecil dibandingkan dengan *layout keyboard* QWERTY.

### 3.4 Diskusi

Pada penelitian ini hanya berfokus pada pencarian *layout keyboard* yang memiliki *travel distance* terkecil untuk mengetik kalimat berbahasa Indonesia. Masih banyak faktor lain yang dapat digunakan untuk mengefisienkan proses mengetik, seperti bobot pada setiap jari yang berbeda tergantung dari seberapa mudah jari tersebut digerakkan, metode mengetik yang memanfaatkan penggunaan 2 tangan secara bergantian, atau bisa juga dengan meminimalkan *type jamming*, yaitu keadaan saat mengetik yang menyebabkan posisi jari tangan berada di posisi yang tabrakan atau posisi yang tidak nyaman. Faktor-faktor tersebut dapat dijadikan pertimbangan pada penelitian-penelitian selanjutnya.

### 4. Conclusion

Setelah didapat *layout keyboard* dengan menggunakan metode algoritma genetika, performa *layout keyboard* QWERTY dalam hal *travel distance* ternyata bernilai 62% lebih besar dibandingkan dengan *layout keyboard* yang dihasilkan. Hal ini berarti bahwa *layout keyboard* yang dihasilkan lebih efisien dalam hal *travel distance* dibandingkan dengan *layout* QWERTY.

### References

- [1] O. B. Adedoyin, "Covid-19 pandemic and online learning: the challenges and opportunities," *Interactive Learning Environments*, pp. 1-13, 2020.
- [2] S. Fung Lau dan C. B. H. Chan, "A special virtual keyboard for disabled computer users," *Proceedings of the International MultiConference of Engineers and Computer Scientists 2017, IMECS 2017*, pp. 539-544, 2017.
- [3] A. Onsorodi dan O. Korhan, "Application of a genetic algorithm to the keyboard layout problem," *PLOS ONE*, vol. 15, p. e0226611, 2020.
- [4] G. Pacheco, E. Palmeira dan K. Yamanaka, "Using Genetic Algorithms to Design an Optimized Keyboard Layout for Brazilian Portuguese," *Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC 2020)*, pp. 437-448, 2020.
- [5] M. Mortenson dan C. S. Sweeten, "Novel Keyboard Layout Optimization," 2020.
- [6] A. Fadel, I. Tuffaha, M. Al-Ayyoub dan Y. Jararwch, "QWERTY Keyboard? }.?BZQ is Better!," *2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, pp. 81-86, 2020.
- [7] M. Mitchell, *An Introduction to Genetic Algorithms*, Cambridge, MA, USA: MIT Press, 1998.