

# Penyelesaian Solusi Pemilihan Bahan Pangan Ternak Berdasarkan Kandungan nutrisinya Menggunakan Algoritma Genetik

Nurul Amelia, Tegar Pandji Asmoro, Asep Irawan  
*School of Electrical Engineering*  
*Telkom University, Indonesia*

nurulamelia@student.telkomuniversity.ac.id, tegarasmoro@student.telkomuniversity.ac.id,  
airawan@student.telkomuniversity.ac.id

**Abstrak** – Berhubung pesatnya perkembangan dalam populasi manusia dan meningkatnya jumlah permintaan akan bahan pakan, dimana para petani harus memenuhi permintaan dan meningkatkan produksi ternak. Dalam penelitian ini, dengan menggunakan masalah yang dialami oleh seorang petani dalam meningkatkan sebuah produksi dengan harga yang minimal namun juga memberikan nutrisi yang paling optimal. Masalah ini sering kali dialami oleh seorang petani, dimana sebuah harga dari pakan ataupun bahan pakan yang akan mereka berikan pada ternak ternyata memiliki harga yang lebih mahal ataupun nutrisinya tidak sesuai dengan harga yang diberikannya. Dengan mencari nutrisi yang ideal bagi ternak ayam, peneliti menawarkan sebuah solusi dengan menggunakan algoritma genetika untuk mencari harga yang optimal namun juga dengan jumlah nutrisi yang maksimal. Tes dilakukan sebanyak 3 kali dengan total 2 generasi setiap tesnya. Tidak hanya secara program namun juga secara manual, dimana semua hasil yang didapatkan berbeda namun juga tentu saja tidak melebihi batasan yang diberikan dimana dalam IDR 25.000.

**Kata Kunci** : algoritma genetika, pakan, harga, petani.

## I. PENDAHULUAN

Ternak adalah hewan yang dipelihara sebagai sumber pangan, sumber bahan baku industri, jasa dan lainnya. Memiliki hewan ternak berarti harus memperhatikan pakan ternak yang diberikan. Pakan ternak adalah makanan atau asupan yang diberikan kepada hewan ternak. Pakan ternak merupakan faktor yang sangat penting dalam kegiatan budidaya di sektor ternak karena pakan ternak sangat menentukan keberhasilan dalam usaha ternak yang dilakukan. Faktor-faktor yang dipertimbangkan dalam pakan ternak biasanya terdiri dari 2 faktor yaitu faktor zat gizi dan faktor biaya. Pada kenyataannya peternak memilih faktor zat gizi yang berkualitas tetapi dengan biaya yang relatif lebih murah. Kandungan nutrisi yang memenuhi kebutuhan ternak dan penggunaan bahan pakan ternak yang murah dapat menguntungkan bagi peternak.

Pada kenyataannya peternak masih kesulitan untuk mencari gabungan antara pakan satu dan yang lainnya

dengan nutrisi yang memenuhi kebutuhan ternak tetapi dengan biaya yang murah. Dengan adanya permasalahan tersebut maka dapat ditemukan solusi melalui pemilihan bahan ternak dengan menggunakan algoritma genetika. Algoritma genetika ini bekerja dengan menemukan solusi optimal dari permasalahan yang mempunyai banyak solusi. Teknik ini akan melakukan pencarian dari banyak solusi yang ada sampai mendapatkan solusi yang terbaik sesuai dengan kriteria yang telah ditentukan sebelumnya atau yang disebut sebagai nilai *fitness*.

## II. RELATED WORK

Pemberian pakan sesuai kebutuhan nutrisi pada ternak ayam sangat disarankan karena dapat mempengaruhi kualitas ayam. Ada 5 nutrisi dasar yang harus terpenuhi dalam pakan ayam diantaranya yaitu protein, lemak, serat kasar, kalsium dan fosfor. Dalam proses pemenuhan kebutuhan nutrisi untuk pakan ternak ayam para peternak membutuhkan biaya yang cukup besar tiap harinya. Sehingga, peternak harus lebih pintar dalam membuat kombinasi pakan ayam yang sesuai agar memperoleh biaya yang minimum namun kebutuhan nutrisi ayam terpenuhi.

## III. METODE PENELITIAN

### A. Knapsack Problem

Knapsack Problem adalah suatu masalah dimana seseorang dihadapan dengan persoalan untuk memilih suatu benda dan lainnya tetapi dengan kapasitas atau jumlah yang terbatas (maksimum) atau sering disebut dengan masalah optimasi kombinatorial. Adapun untuk optimasi pemilihannya dimaksudkan agar dalam proses pemilihan menghasilkan keuntungan semaksimal mungkin.

Jenis-jenis *Knapsack Problem* :

- 0/1 Knapsack problem*  
Setiap barang hanya tersedia 1 unit, 0 artinya barang dibawa dan 1 artinya barang ditinggal (*take it or leave it*).
- Fractional Knapsack problem*  
Barang boleh dibawa sebagian saja. Versi *problem* ini apabila barang yang tersedia dapat dibagi-bagi misalnya gula, tepung dan sebagainya.
- Bounded Knapsack problem*

Setiap barang tersedia sebanyak N unit (jumlahnya terbatas).

d) *Unbounded Knapsack problem*

Setiap barang tersedia lebih dari 1 unit, jumlahnya tidak terbatas.

B. Algoritma Genetika

Menurut John Holland (1975) dari Universitas Michigan, Algoritma Genetika ini sebagai algoritma pencarian yang digunakan untuk memecahkan masalah optimasi kompleks yang sulit dilakukan dengan cara konvensional yang berbasiskan dasar alam seleksi dan genetika. Misalnya individu dalam satu populasi disebut kromosom, dan mereka yang mewakili solusi untuk masalah pada kromosom.

1) Populasi Inisial

Sebuah populasi inisial ditentukan secara acak sehingga solusi inisialnya dapat ditemukan. Beberapa kromosom menampilkan sebagai sebuah solusi yang diinginkan di dalam sebuah populasi, meskipun hal ini dapat mengurangi adanya kesempatan dalam mendapatkan solusi yang maksimum. Dalam kesempatan penelitian ini jumlah kromosomnya bergantung dalam jumlah bahan yang akan dibeli atau digunakan, dan setiap gen merupakan bahan-bahan atau barang yang sudah dimasukan sebelumnya.

$$P = K \times jD$$

informasi :

P = Populasi

K = Kromosom yang panjangnya sesuai dengan daftar nama pakan.

jD = Jumlah pakan.

Hasilnya, populasi dari setiap pakan akan selalu berbeda bergantung pada jumlah pakan yang dimasukan.

2) Evaluasi Kromosom

Setelah tahap Populasi Inisial langkah selanjutnya adalah tahap mengevaluasi kromosom dengan cara menghitung nilai kebugaran di setiap kromosom yang dihasilkan dengan menggunakan rumus:

$$fitness = \sum_{i=1}^n Bi.Xi$$

Dimana :

- n = panjang kromosom
- Bi = gen ke -i
- Xi = nilai (total harga) ke -i

Jadi dalam permasalahan knapsack ini, fungsi fitness ditentukan adanya tujuan yang ingin dicapai. Pada kasus ini, rencana yang akan dilakukan adalah menentukan sebuah item dengan nilai total paling signifikan dengan menggunakan fungsi maksimalisasi untuk mendapatkan hasil rekomendasi terbaik dan total harga kurang atau sama dengan total yang telah ditentukan.

3) Crossover

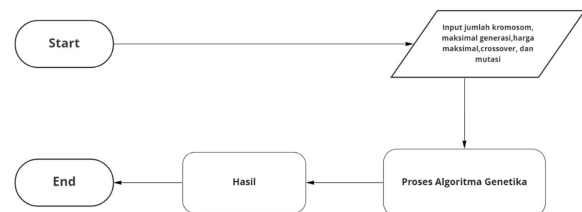
Crossover adalah operator dari algoritma genetika yang melibatkan dua induk untuk membentuk kromosom baru. Proses crossover dilakukan dengan menukarkan sebagian informasi pada kromosom induk pertama dengan informasi dari kromosom induk kedua.

4) Mutasi

Mutasi merupakan operator yang menukar nilai gen dengan nilai inversinya. Pada setiap individu mengalami mutasi gen dengan probabilitas mutasi yang ditentukan, mutasi ini dilakukan dengan memberikan nilai gen pada gen yang terpilih untuk dimutasikan. Dalam mutasi ini ada dua hal penting yang diperhatikan yaitu hasil implikasi dari fungsionalitas algoritma genetika yang terbaik dan tahap kritis, maka untuk memilih jenis metode yang tepat menggunakan fungsionalitas dan implikasi operasi dengan variabel sampel populasi yang diperlukan tergantung pada pengkodean dan masalah yang menjadi perhatian. Dalam mutasi ini memiliki banyak metode seleksi salah satu nya dengan menggunakan metode *roulette wheel*.

IV. GAMBARAN DAN DESAIN SISTEM

A. Desain Sistem



Gambar 1. Desain Sistem

Berdasarkan flowchart di atas langkah penyelesaian algoritma genetika setelah memulai program akan diperintahkan untuk menginputkan jumlah kromosom yang diinginkan, kemudian setelah itu jumlah maksimal generasi. Setelah proses tersebut dilakukan, proses selanjutnya adalah memasukkan harga maksimal (maksimasi-nya) sesuai dengan ketentuan yang ditentukan agar menghasilkan jumlah nutrisi yang memenuhi tetapi dengan biaya yang murah. Tahap selanjutnya adalah memasukkan nilai crossover serta mutasi dan proses algoritma genetika akan diproses. Pada tahap akhir, algoritma genetika akan menghasilkan output generasi akhir, fitness generasi akhir, parameter yang dioptimalkan serta rekomendasi bahan pangan yang akan dibeli.

B. Implementasi Algoritma Genetika

1) Aturan Rekomendasi Item

Aturan yang ada pada Tabel akan digunakan dalam proses Algoritma Genetika. Aturan dari rekomendasi bahan pakan, terlampir sebagai berikut :

TABEL I. ATURAN REKOMENDASI ITEM

No.	Aturan
1	Harga dari total hasil dari optimasi algoritma genetika tidak boleh melebihi atau sama dengan harga Rp 25.000
2	Hasil perhitungan algoritma mencari nilai <i>fitness</i> nutrisi dengan gabungan bahan pakan paling besar.
3	Id kategori dari bahan pangan ditentukan sesuai urutan pada nama, sehingga gabungan antara bahan pangan tidak berulang

## 2) Representasi Kromosom

Dalam hal ini, terdapat 7 nutrisi yang dapat menjadi pakan ayam, sebagai ditunjukkan pada tabel. Setiap nutrisi mempunyai harga yang berbeda-beda per-kg. Harga ini diperoleh berdasarkan hasil observasi dari internet.

TABEL II. DAFTAR NUTRISI PAKAN AYAM

No	Nama Pakan	Harga	Nilai Nutrisi
1	Jagung	3000	3250
2	Dedak	4000	1750
3	Ampas Kelapa	4000	3500
4	CGF	6000	4370
5	Polar Gandum	8000	2160
6	Pur Ayam	11000	3550
7	Ampas Tahu	13000	4240

## 3) Prosedur Inisialisasi

Inisialisasi populasi ditentukan secara acak, dimana terdapat 1 populasi yang terdiri dari 10 kromosom. Dalam 1 kromosom terdapat 7 gen didalamnya. Dibawah ini, tampilan inisialisasi populasi yang diperoleh :

```
Initial populasi:
[[1 1 1 1 1 0 1]
 [0 1 1 1 0 0 0]
 [0 0 1 0 1 0 1]
 [0 1 1 1 1 0 0]
 [1 1 0 1 1 1 1]
 [1 1 0 1 0 0 0]
 [0 0 0 1 1 0 0]
 [1 0 1 0 0 1 1]
 [0 0 1 1 0 1 0]
 [1 1 1 1 1 0 0]]
```

Gambar 2. Inisialisasi Populasi

## 4) Fitness Function

Dalam masalah knapsack, fungsi *fitness* ditentukan dengan menentukan tujuan yang ingin dicapai. Dalam hal ini, rencana yang akan dilakukan adalah menentukan item dengan nilai total paling signifikan dengan menggunakan fungsi maksimisasi untuk mendapatkan hasil rekomendasi terbaik. Total harga kurang atau sama dengan total anggaran, maka dapat dirumuskan sebagai berikut :

$$fitness = \sum_{i=1}^n Bi.Xi$$

Dimana :

- n = panjang kromosom
- Bi = gen ke -i
- Xi = nilai (total harga) ke -i

kromosom 1 = [ 1 1 1 1 0 1 ]

fitness =

$$((1 \times 3250) + (1 \times 1750) + (1 \times 3500) + (1 \times 4370) + (1 \times 2160) + (0 \times 3550) + (1 \times 4140)) = 19170$$

## 5) Selection

Pada penelitian ini kita menggunakan metode seleksi, yang dimana seleksi adalah sebuah metode dengan cara mencari suatu individu dari beberapa grup. Dengan membiarkan grup ini melakukan kompetisi satu dengan yang lain. Sehingga dari semua individu akan ditentukan individu dengan *fitness* skor tertinggi dan terbaik, dan akan dilanjutkan ke proses berikutnya. Proses ini digunakan tidak hanya dikarenakan tingkat efisien namun juga kemudahan dalam implementasinya.

Pada seleksi akan memilih sebuah *parent* dan dibagi dengan jumlah populasi secara acak serta melakukan pemilihan dengan nilai *fitness* tertinggi. Kromosom yang terpilih akan menjadi *parent*.

selection :

```
[[0. 1. 1. 1. 0. 1. 0.]
 [0. 1. 1. 1. 0. 1. 0.]
 [0. 1. 1. 1. 0. 1. 0.]
 [0. 1. 1. 1. 0. 1. 0.]
 [0. 1. 1. 1. 0. 1. 0.]]
```

Gambar 3. Selection

## 6) Crossover

Proses *crossover* dilakukan setelah mendapatkan *parent* dalam proses *selection*. *Crossover* adalah proses persilangan setiap *parent* yang akan menghasilkan keturunan atau kromosom baru.

Proses *crossover* tidak hanya dilakukan satu kali. Namun, dilakukan terus menerus hingga nilai random yang dimunculkan lebih kecil dari probabilitas *crossover*, dan probabilitas *crossover* yang digunakan 50% atau 0.50. Setelah proses *crossover* selesai, pemeriksaan setiap

kromosom akan dilakukan untuk memastikan bahwa berat kromosom tidak melebihi batas. Jika kromosom melebihi batas, maka nilai gen kromosom akan bernilai nol mulai dari gen terakhir setiap kromosom hingga bobotnya sama atau lebih ringan dari batas. Pada gambar 4 menunjukkan hasil *crossover*:

*crossover* :

```
[[0. 1. 1. 1. 0. 1. 0.]
 [0. 1. 1. 1. 0. 1. 0.]
 [0. 1. 1. 1. 0. 1. 0.]
 [0. 1. 1. 1. 0. 1. 0.]
 [0. 1. 1. 1. 0. 1. 0.]]
```

Gambar 4. *Crossover*

#### 7) Mutasi

Proses mutasi pada algoritma genetika ini hampir sama dengan proses pada *crossover*. Proses mutasi akan dilakukan secara terus menerus sampai nilai acak yang ada kurang dari nilai probabilitas mutasinya. Nilai probabilitas mutasi yang digunakan pada algoritma genetika adalah 70% atau 0.7. Proses mutasi ini akan mengoptimalkan nilai gen di dalam setiap kromosom yang ada.

Proses mutasi ini juga akan memeriksa apakah nilai kromosom sudah maksimal (tidak melebihi ambang batas). Jika nilai kromosom yang didapatkan melebihi batas ambang yang telah ditentukan, maka nilai dari gen setiap kromosom bakal ditinjau kembali dan akan diubah sampai dengan nilai yang didapatkan sama dengan atau kurang dari ambang batas yang telah ditentukan.

*mutasi* :

```
[[0. 1. 1. 1. 0. 1. 0.]
 [1. 1. 1. 1. 0. 1. 0.]
 [0. 0. 1. 1. 0. 1. 0.]
 [0. 1. 1. 1. 0. 0. 0.]
 [0. 0. 1. 1. 0. 1. 0.]]
```

Gambar 5. Mutasi

## V. PENGUJIAN DAN HASIL

### A) Algoritma Genetika

Pada setiap eksperimen, jumlah generasi dan ukuran populasi menggunakan nilai input yang sama, dimana disini peneliti menggunakan populasi dengan jumlah kromosom 10 dan jumlah generasinya 2. Sedangkan beberapa hal yang berbeda tiap percobaannya adalah probabilitas *crossover*-nya sebesar 0.5, 0.55, 0.6. Sedangkan probabilitas mutasinya 0.7. Untuk batasan harga yang akan digunakan dalam algoritma sama dengan IDR 25.000 pada setiap percobaannya. Hasil dari percobaan yang didapat disini tidak boleh melebihi atau sama dengan IDR 25.000.

Parameter yang digunakan dalam tes :

TABEL III ALGORITMA GENETIKA

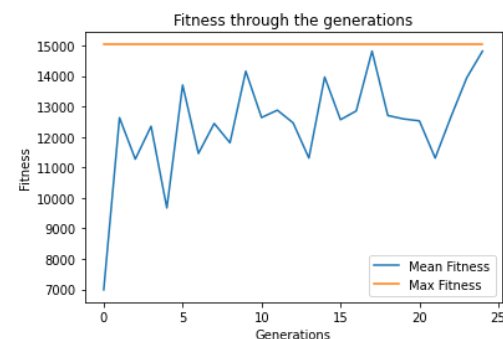
No	Parameter	Test 1	Test 2	Test 3
1	Kromosom	10	10	10
2	Jumlah Generasi	25	25	25
3	Crossover	0.5	0.55	0.6
4	Mutasi	0.7	0.7	0.7
5	Batasan Harga	Rp 25.000	Rp 25.000	Rp 25.000

Hasil Tes :

TABEL IV HASIL TES

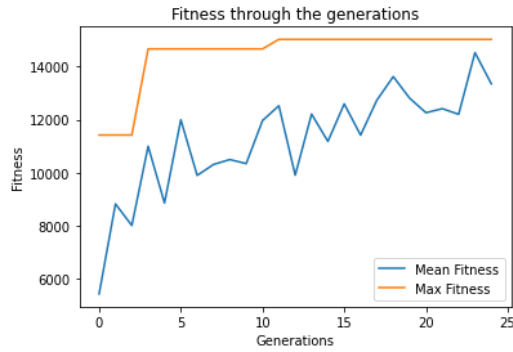
No	Hasil	Test 1	Test 2	Test 3
1	Hasil Harga yang Terpakai	Rp 25.000	Rp.25.000	Rp.25.000
2	Jumlah pakan yang Terpakai	5	5	5
3	Nilai Fitness Terbaik	15.030	15.030	15.030
4	Pakan Jagung	1	1	1
5	Pakan Dedak	1	1	1
6	Pakan Ampas Kelapa	1	1	1
7	Pakan CGF	1	1	1
8	Pakan Polar Gandum	1	1	1
9	Pakan Pur Ayam	0	0	0
10	Pakan Ampas Tahu	0	0	0

Setelah dilakukan 3 kali tes dengan 25 generasi, dapat diketahui bahwa memiliki grafik yang sama, sebagai berikut :



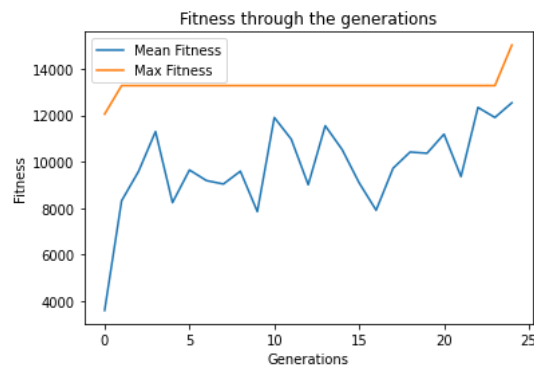
(25, 10)

Gambar 6. Grafik Test 1



(25, 10)

Gambar 7. Grafik Test 2



(25, 10)

Gambar 8. Grafik Test 3

Setelah dilakukan 3 kali test dengan jumlah generasi 25 dan crossover yang berbeda-beda tetapi dengan mutasi yang sama, dapat disimpulkan bahwa dari ke 3 test tersebut menghasilkan nilai fitness yang sama yaitu 15.030. Tetapi, untuk grafik yang dihasilkan mengalami perbedaan. Perbedaanya dapat dilihat pada gambar grafik 7,8 dan 9 pada nilai max fitness dan mean fitness.

#### B) Hasil Hitungan Manual

STEP - 7 - HASIL KROMOSOM GENERASI KE-2

K1	1	1	0	1	0	0	9570
K2	0	0	1	1	0	1	11920
K3	0	0	1	1	1	0	10030
K4	0	0	1	1	0	1	12120
K5	0	0	0	1	0	0	8620
K6	0	1	1	1	0	0	9620
K7	1	1	0	1	1	1	19330
K8	1	0	1	0	0	1	14570
K9	0	0	1	0	1	0	9910
K10	1	1	1	1	1	0	15030 ✓
Average							12000

Solusi K10 = Jagung, Dedek, Apas Kelapa, CGF, Polar Gandum.

Gambar 9. Hitungan Manual

## VI. KESIMPULAN

Setelah melakukan implementasi menggunakan algoritma genetika, untuk menentukan kebutuhan nutrisi tetapi dengan biaya murah (tidak melebihi biaya maksimal), maka dapat disimpulkan bahwa algoritma ini berjalan dengan baik. Untuk hasil solusi bahan pangan yang akan dibeli dari algoritma ini dapat berubah-ubah karena untuk inisialisasi populasi yang dihasilkan bersifat random. Tetapi, untuk solusi hasil tidak melebihi batas maksimal yang diinputkan.

## VII. DAFTAR PUSTAKA

- Bambang. 2019. "Pedoman Pentingnya Nutrisi Pakan Ternak", [http://ternak.blitarkab.go.id/2016/08/pedoman-pentingnya-nutrisi-pakan-ternak\\_22.html](http://ternak.blitarkab.go.id/2016/08/pedoman-pentingnya-nutrisi-pakan-ternak_22.html), diakses pada 02 Juli 2022.
- Disnak. 2013. "Nutrisi dan Pakan Ternak Sapi", [NUTRISI DAN PAKAN TERNAK SAPI - Informasi Dinas Peternakan Dan Kesehatan Hewan Provinsi Sumatera Barat](http://NUTRISI%20DAN%20PAKAN%20TERNAK%20SAPI%20-%20Informasi%20Dinas%20Peternakan%20Dan%20Kesehatan%20Hewan%20Provinsi%20Sumatera%20Barat), diakses pada 01 Juli 2022.
- Ervina. 2015. "Penentuan Komposisi Pakan Ternak Untuk Memenuhi Kebutuhan Nutrisi Ayam Petelur Dengan Biaya Minimum Menggunakan Algoritma Genetika", [Penentuan Komposisi Pakan Ternak Untuk Memenuhi Kebutuhan Nutrisi Ayam Petelur Dengan Biaya Minimum Menggunakan Algoritma Genetika - Brawijaya Knowledge Garden \(ub.ac.id\)](http://Penentuan%20Komposisi%20Pakan%20Ternak%20Untuk%20Memenuhi%20Kebutuhan%20Nutrisi%20Ayam%20Petelur%20Dengan%20Biaya%20Minimum%20Menggunakan%20Algoritma%20Genetika%20-%20Brawijaya%20Knowledge%20Garden%20(ub.ac.id)), diakses pada 1 Juli 2022.
- Id, Paralegal. 2014. "Ternak", <https://paralegal.id/pengertian/ternak/>, diakses 27 Juni 2022.
- Kartini, dkk. "Penyelesaian Knapsack problem menggunakan algoritma genetika", <https://www.neliti.com/id/publications/172385/penyelesaian-knapsack-problem-menggunakan-algoritma-genetika>, diakses pada 1 Juli 2022.
- Rosyid. "Algoritma Genetika", [http://rosyid.lecturer.pens.ac.id/materi%20AI/Minggu 10%20-%20Algoritma%20Genetika.pdf](http://rosyid.lecturer.pens.ac.id/materi%20AI/Minggu%2010%20-%20Algoritma%20Genetika.pdf), diakses pada 04 Juli 2022.
- Suharjito.2021."Algoritma Genetika dengan Python", <https://onlinelearning.binus.ac.id/computer-science/post/algoritma-genetika-dengan-python>, diakses pada 02 Juli 2022.
- Surya, dkk.. "Budgeting System In Event Management Application Using Web-Based Genetic Algorithms", <https://lms.telkomuniversity.ac.id/mod/resource/view.php?id=2139634>, diakses pada 26 Juni 2022.
- Sutiono. "Algoritma Knapsack Problem", <https://dosenit.com/kuliah-it/rpl/algoritma-knapsack>, diakses pada 29 Juni 2022.