



Introduction

- ✓ NPCs (Non-Playable Characters) are key agents in single-player games, simulating player interaction.
- ✓ Static environments present navigation challenges for NPCs.
- ✓ Pathfinding algorithms are needed to calculate optimal routes and avoid obstacles.
- ✓ This study focuses on comparing Dijkstra and A-Star algorithms within the Chaos Crossing game.



Research Question

Which algorithm Dijkstra or A-Star provides better performance for NPC pathfinding in static grid-based environments?

Method

Design Type

Quantitative comparative analysis

Scenarios

- 5 level designs with fixed static obstacles.
- 3 start-goal positions per level.

Metrics

- Route search time.
- Path distance.
- Memory usage (via System.GC.GetTotalMemory() on Unity Engine C#)

Data Collection

Results averaged per parameter across 15 tests per algorithm

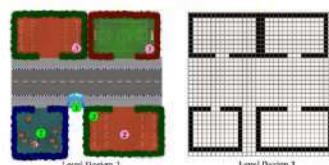
Comparative Analysis of Dijkstra and A-Star Algorithms for NPC Pathfinding in Chaos Crossing

Research Objectives

- ✓ To evaluate the effectiveness of Dijkstra and A-Star algorithms for NPC pathfinding.
- ✓ To measure performance based on route time, distance, and memory usage.
- ✓ To recommend the most suitable algorithm for grid-based static environments.

Result & Discussion

Design and Implementation



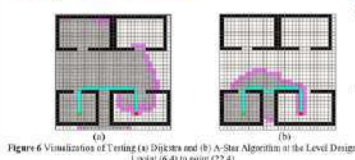
- Five static level designs were created with increasing complexity.
- Each level contains three NPC routes (start to goal).
- Implemented in Unity Engine using C#.
- NPCs use either Dijkstra or A-Star for movement on a black-and-white grid (black = obstacle, white = path).

Result of Testing

Points (x,y)		Time (s)		Distance (unit)		Memory (MB)	
Start	Destination	Dijkstra	A-Star	Dijkstra	A-Star	Dijkstra	A-Star
Level Design 1							
(13,8)	(28,24)	6.74	0.96	23	23	20.80	18.38
(6,4)	(22,4)	5.23	1.84	29	29	20.32	18.91
(17,8)	(13,24)	6.42	2.66	30	30	21.68	18.30

- 15 test cases per algorithm (5 levels × 3 positions).
- Parameters measured:
 - Route Search Time (seconds)
 - Path Distance (units)
 - Memory Usage (MB)

Analysis and Result



- ⌚ Search Time (avg):
 - **A-Star: 20.76 s** ✓
 - Faster due to heuristic-based search.
 - Dijkstra: 36.37 s
 - Slower as it explores all nodes equally.
- 📏 Path Distance (avg):
 - A-Star: 42.39 units
 - Slightly longer, but acceptable in practice.
 - **Dijkstra: 42.26 units** ✓
 - More precise shortest path.
- 💾 Memory Usage (avg):
 - **A-Star: 20.19 MB** ✓
 - More efficient, skips unnecessary nodes.
 - Dijkstra: 22.17 MB
 - Uses more memory due to full node exploration.

Conclusion

- ✓ A-Star performs faster and uses less memory, ideal for real-time pathfinding.
- ✓ Dijkstra yields slightly more accurate paths but at the cost of time and resources.
- ✓ Recommendation: Use A-Star for responsive NPC behavior in games with performance constraints; consider Dijkstra for offline or precision-critical tasks.

Game Demo



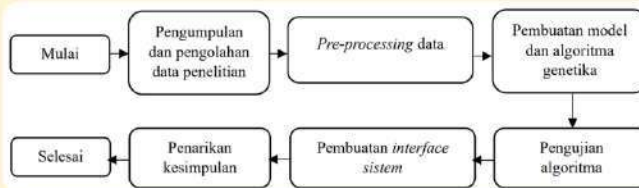
Full Paper



PENERAPAN ALGORITMA GENETIKA UNTUK OPTIMASI KOMPOSISI MAKANAN PADA INDIVIDU DENGAN PREDIABETES

1 PERMASALAHAN

Banyak penderita prediabetes kesulitan menentukan komposisi makanan yang seimbang. Ini disebabkan keterbatasan pilihan makanan, kompleksitas kebutuhan nutrisi, dan kurangnya pemahaman pola makan sehat.



2 METODE

Pengumpulan Data dari Tabel Komposisi Pangan Indonesia 2017

3 PENGUJIAN DAN HASIL ANALISA ALGORITMA

Menu Makanan Sebelum Algoritma Genetika

	Karbohidrat	Protein hewani	Protein nabati	Serat	Camilan
Makan pagi	Gembili ubi 100gr	Daging angsa 100gr	Tempe lamtoro 100gr	Labu waluh 100gr	Buah naga putih 100gr
Makan siang	Tapai singkong 100gr	Ikan belida 100gr	Tempe pasar 100gr	Jamur kuping segar 100gr	-
Makan malam	Lepok/Ubi rumput 100gr	Ikan mamar merah 100gr	Kacang bogor segar 100gr	Baligo 100gr	Pisang ketip segar 100gr

Menu Makanan Setelah Algoritma Genetika

	Karbohidrat	Protein hewani	Protein nabati	Serat	Camilan
Makan pagi	Bubur Manado 60gr	Ikan bader 90gr	Kacang kedelai 40gr	Kacang mekah 140gr	Mangga 200gr
Makan siang	Kerupuk kemplang 70gr	Ginjal sapi 100gr	Tahu goreng 150gr	Jantung pisang 40gr	-
Makan malam	Getuk singkong 60gr	Ikan sidat segar 60gr	Tauco 70gr	Jamur kuping kering 90gr	Pisang ayam 200gr

Hasil Akurasi Sebelum Algoritma Genetika

	Energi	Karbohidrat	Protein	Lemak	Serat
Target	2157 Kal	269.6 gr	107.8 gr	47.9 gr	35.9 gr
Aktual	1569 Kal	271.4 gr	77.7 gr	21.8 gr	51.3 gr
Akurasi target	72.7%	99.3%	72.1%	45.5%	57.1%
Normalisasi akurasi	14.6%	19.9%	14.4%	9.1%	11.4%
Total akurasi keseluruhan	69.4%				

Hasil Akurasi Setelah Algoritma Genetika

	Energi	Karbohidrat	Protein	Lemak	Serat
Target	2157 Kal	269.6 gr	107.8 gr	47.9 gr	35.9 gr
Aktual	2152 Kal	253.6 gr	100.5 gr	70.7 gr	65.4 gr
Akurasi target	99.7%	94%	93.2%	52.4%	17.8%
Normalisasi akurasi	19.9%	18.8%	18.6%	10.5%	3.56%
Total akurasi keseluruhan	71.4%				

Tampilan Halaman Awal Sistem



Tampilan Formulir Data Personal



Tampilan Hasil Optimasi Sistem



4 KESIMPULAN

Penggunaan algoritma genetika menghasilkan akurasi 71,4%, lebih tinggi 2% dibanding metode tanpa algoritma genetika (69,4%). Peningkatan akurasi 2% ini menunjukkan bahwa algoritma genetika mampu memberikan solusi yang lebih baik dalam optimasi komposisi makanan, yang berpotensi untuk membantu individu prediabetes dalam mengelola kondisi mereka dengan lebih efektif.

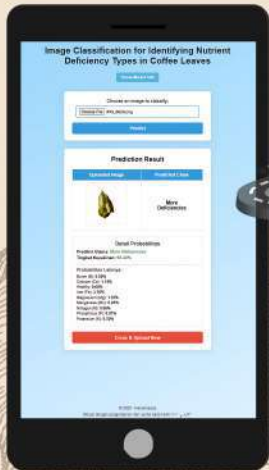


KLASIFIKASI KEKURANGAN NUTRISI DAUN KOPI DENGAN PENGGABUNGAN FITUR GOOGLNET-RESNET

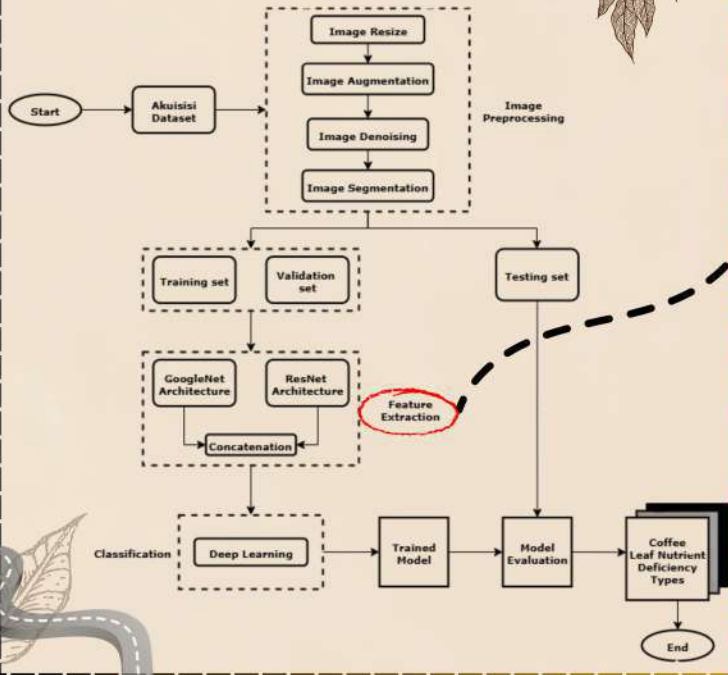


LATAR BELAKANG

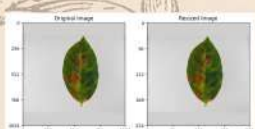
- Kekurangan nutrisi pada daun kopi menurunkan kualitas dan produktivitas, berdampak buruk pada pendapatan petani.
- Dibutuhkan sistem otomatis untuk identifikasi kekurangan nutrisi yang cepat dan akurat, karena metode manual bersifat subjektif dan lambat.
- Penelitian ini mengusulkan model deep learning dengan menggabungkan fitur dari arsitektur GoogLeNet dan ResNet untuk meningkatkan akurasi.



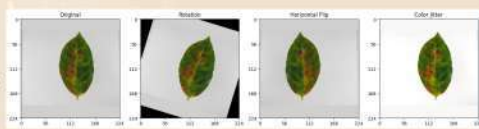
TAHAPAN PENELITIAN



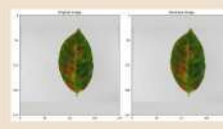
Demonstrasi Model Klasifikasi



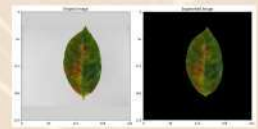
Ubah Ukuran Citra
(Ukuran diubah menjadi 224x224 untuk input)



Contoh Augmentasi Citra
(Rotation, Horizontal Flip, Color Jitter)



Citra Hasil Denoising
(Setelah Gaussian Blur)

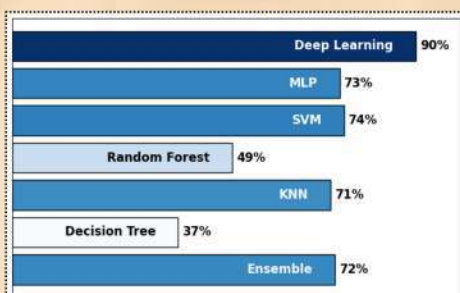


Citra Hasil Segmentasi
(Setelah Thresholding Otsu dan Operasi Morfologi)

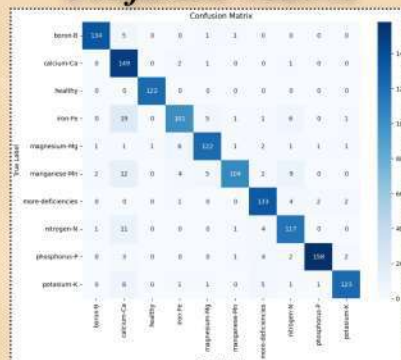
METODOLOGI

- **Dataset:** Menggunakan dataset CoLeaf yang berisi 1.006 gambar daun kopi Arabika dalam 10 kelas kekurangan nutrisi.
- **Preprocessing:** Resize (224x224), Augmentasi, Denoising dan Segmentasi.
- **Ekstraksi Fitur:** Menggabungkan (concatenation) fitur dari arsitektur GoogLeNet dan ResNet.
- **Klasifikasi:** Menggunakan model Deep Learning dengan head classifier Softmax.

Perbandingan Kinerja Model



Confusion Matrix



KESIMPULAN

- Penggabungan fitur dari GoogLeNet dan ResNet secara signifikan meningkatkan kinerja klasifikasi dibandingkan arsitektur tunggal.
- Model gabungan dengan pendekatan deep learning mencapai akurasi tertinggi sebesar 90%.
- Metode yang diusulkan efektif mengekstrak fitur visual yang lebih representatif, sehingga meningkatkan akurasi diagnosis otomatis.

Penerapan Genetic Algorithm Pada Optimasi Bahan MPASI Guna Memenuhi Gizi Bayi

Safa Auliya Hidayat

Pendahuluan

Gizi berperan penting dalam tumbuh kembang balita. Kekurangan gizi dapat menyebabkan komplikasi kesehatan dan stunting, yang di Indonesia mencapai 21,6% (SSGI Kemenkes 2022). Untuk mencukupi kebutuhan gizi, dibutuhkan kombinasi bahan makanan yang tepat. Optimasi bahan makanan bertujuan untuk menghasilkan formulasi MPASI yang sehat dan sesuai kebutuhan nutrisi. Penelitian ini menggunakan algoritma genetika untuk mengoptimalkan bahan MPASI sehingga menghasilkan komposisi yang variatif dan bernutrisi sesuai kebutuhan bayi.

Metodologi

Analisis Kebutuhan Gizi

- Menghitung kebutuhan kalori, protein, lemak, dan karbohidrat bayi sesuai usia dan BB.

Pengumpulan Data Bahan Makanan

- Mengambil data komposisi gizi dari Tabel Komposisi Pangan Indonesia (TKPI).

Penentuan Batasan Nutrisi

- Menentukan batas minimal dan maksimal setiap unsur gizi yang harus dipenuhi.

Implementasi Algoritma Genetika

- Inisialisasi populasi awal berupa kombinasi acak bahan makanan.
- Evaluasi fitness berdasarkan pemenuhan kebutuhan gizi.
- Proses seleksi, crossover, dan mutasi untuk menghasilkan kombinasi baru.
- Iterasi hingga ditemukan kombinasi optimal bahan makanan.

Analisis Hasil

- Menampilkan kombinasi bahan makanan terbaik yang memenuhi kebutuhan gizi bayi secara seimbang.

Hasil

Optimasi bahan makanan MPASI menggunakan Genetic Algorithm menunjukkan hasil yang menjanjikan. Berdasarkan pengujian parameter, didapat nilai optimal sebagai berikut

Testing Parameters	Optimal Value
Population Size	140
Generation Size	4500
Mr Size	0.4
Cr Size	1

Pengujian dilakukan pada kasus bayi A (11 bulan, berat 6.8 kg). Sistem menghitung kebutuhan nutrisi aktual dan membandingkannya dengan hasil optimasi bahan makanan. Akurasi sistem mencapai 99.1%, yang dihitung dari selisih kebutuhan nutrisi aktual dengan total nutrisi hasil perhitungan Genetic Algorithm.

Kesimpulan

Penelitian ini menunjukkan bahwa Genetic Algorithm efektif dalam mengoptimasi bahan MPASI sesuai kebutuhan gizi bayi. Sistem merekomendasikan bahan makanan berdasarkan data bayi, dengan hasil yang menunjukkan akurasi tinggi terhadap kebutuhan gizi aktual.

Rekomendasi:

- Perlu penelitian lanjutan untuk memperhatikan alergi makanan pada bayi.
- Juga disarankan untuk mempertimbangkan variasi dan kompatibilitas bahan makanan dalam pengembangan berikutnya.

Hasil Sistem

