**HargaSetara: Inovasi Smart Distribution Komoditas Pangan Pertanian dan Hortikultural Berbasis AI untuk Urban Food Resilience**

**🧠 Justifikasi Orisinalitas & Keunggulan Proyek HargaSetara**

**HargaSetara** adalah sistem cerdas yang mengklasifikasikan status surplus, defisit, atau netral suatu komoditas pangan di tingkat kabupaten/kota berdasarkan prediksi permintaan dan pasokan, dengan tujuan utama **menstabilkan harga pangan melalui redistribusi yang lebih tepat dan cepat**.

Berbeda dari sistem yang sudah ada seperti **SICA (ITB)** yang fokus pada kalender tanam berbasis iklim, atau sistem statistik seperti **Agmarknet (India)** dan **AMIS FAO**, HargaSetara **menggabungkan data harga harian, prediksi permintaan (berbasis populasi, hari libur, event khusus), dan pasokan (berbasis data cuaca, luas panen, dan produktivitas)** dalam satu kerangka model prediktif terpadu.

🔎 Tidak ada sistem di Indonesia—dan bahkan di luar negeri—yang:

* Memodelkan pasokan dan permintaan pangan **berbasis data granular** harian dan spasial,
* Menggunakan klasifikasi AI untuk mendeteksi daerah **berpotensi defisit atau surplus**,
* Memberikan **rekomendasi distribusi antar wilayah** untuk stabilisasi harga pangan.

Dengan cakupan lokal (kabupaten/kota), pendekatan prediktif berbasis AI, dan keluaran yang langsung actionable untuk pengambil kebijakan maupun pelaku distribusi pangan, HargaSetara mengisi celah kritis yang belum dijangkau oleh sistem eksisting, menjadikannya **solusi orisinal dan berdampak tinggi** dalam penguatan ketahanan pangan nasional.

## **🎯 TUJUAN AKHIR: Klasifikasi Surplus / Defisit / Netral**

Kebutuhannya:

* Prediksi **permintaan** (kg)
* Prediksi **pasokan** (kg)
* Surplus = pasokan > permintaan; Defisit = sebaliknya

**🔧 MASALAH: Tidak Ada Data Langsung Permintaan & Pasokan**

Namun kamu **punya**:

* ❗ **Harga harian** (indikator tidak langsung)
* ✅ Data cuaca (harian)
* ✅ Jumlah penduduk (per kab/kota)
* ❗ Event khusus dan hari libur (bisa di-generate)
* ❗ Luas panen dan produksi (tahunan, terbatas)

**✅ SOLUSI STRATEGIS: Tahapan Langkah Praktis**

**🧩 Langkah 1: Bangun Estimator Permintaan Konsumen**

Permintaan dipengaruhi oleh:

* Hari (libur atau bukan)
* Event (lebaran, natal, dst)
* Jumlah penduduk
* Hari-hari tertentu seperti awal/akhir bulan, akhir pekan
* Harga historis (jika murah → permintaan naik)

🔧 **Langkahmu:**

* Buat model regresi (RandomForest/XGBoost) untuk memprediksi permintaan\_kg (target semu) dari semua fitur di atas
* Gunakan pendekatan **semi-supervised**: misalnya pakai rerata konsumsi nasional → skala ke populasi tiap kabupaten

💡 Contoh:

Konsumsi rata-rata cabe rawit nasional: 1.5 kg/kapita/bulan  
→ Hariannya: 0.05 kg → *dikalikan penduduk kab/kota*

Kamu bisa gunakan itu sebagai pseudo-target awal, lalu refine via model.

**🧩 Langkah 2: Bangun Estimator Pasokan**

Pasokan dipengaruhi oleh:

* Cuaca (hujan, suhu, kelembaban → tanam atau tidak)
* Musim (rolling cuaca 30 hari → waktu panen?)
* Panen tahunan (kalibrasi volume pasokan ke kabupaten)
* Jumlah kelompok tani (*proxy* jika tidak historikal)

🔧 **Langkahmu:**

* Gunakan data luas panen + produksi → buat baseline pasokan bulanan/harian dengan distribusi
* Buat model untuk mendeteksi *musim panen* dari cuaca (misal: banyak hujan bulan ke-1 → panen bulan ke-4?)
* Bangun model pasokan harian berbasis rolling cuaca + kalender panen lokal

**🧩 Langkah 3: Bangun Label Klasifikasi Surplus–Defisit**

Gunakan rumus sederhana:

python

SalinEdit

if pasokan\_pred > permintaan\_pred + margin:

label = 'surplus'

elif permintaan\_pred > pasokan\_pred + margin:

label = 'defisit'

else:

label = 'netral'

margin bisa kamu tetapkan secara eksperimen (misal 10% gap dianggap netral).

**✳️ Perhatikan:**

**❗ Masalah Data Historis**

* Tidak perlu semua benar/komplet → kamu bisa *drop* atau *imputasi* dengan rata-rata provinsi
* **Rolling cuaca** & **jumlah penduduk** sangat powerful → manfaatkan maksimal

**❗ Validasi**

Kamu perlu:

* Lakukan validasi manual terbatas (misal bandingkan label vs harga riil: harga naik → defisit?)

**🚀 ROADMAP FINAL UNTUK PROYEK HARGASETARA**

| **Tahapan** | **Target** | **Tools/Model** |
| --- | --- | --- |
| 1. Prediksi Permintaan | kg/hari/kabkota | RandomForest / Rule-based |
| 2. Prediksi Pasokan | kg/hari/kabkota | Cuaca + model regresi panen |
| 3. Buat label surplus–defisit | dari prediksi | Rule (pasokan vs permintaan) |
| 4. Bangun model klasifikasi surplus–defisit | harga, cuaca, populasi, event | XGBoost / RF |
| 5. Dashboard interaktif | Visualisasi hasil prediksi | Streamlit / Dash |

**🔚 Kesimpulan & Saran**

* **Jangan tunggu data sempurna** → mulai dari *rule-based pseudo label*, refine dengan model
* **Gunakan kombinasi statistik + ML** untuk *permintaan* dan *pasokan*
* **Bangun satu pipeline komoditas dulu (misal: cabai rawit)** → baru replikasi ke lainnya

Kalau kamu mau, aku bisa bantu mulai dari:

* Generator data permintaan awal dari penduduk
* Model baseline prediksi pasokan dari rolling cuaca
* Pipeline integrasi semua komoditas

Mau mulai dari mana?

|  |  |  |
| --- | --- | --- |
| **Kolom** | **Tipe Data** | **Keterangan** |
| tanggal | datetime | Tanggal pencatatan |
| kota | string | Nama kabupaten/kota di Jawa Timur |
| harga\_rata2 | numeric | Harga eceran rata-rata cabai rawit merah (dari data konsumen) |
| pasokan | numeric | Total panen & distribusi masuk ke kota tersebut (jika ada data) |
| cuaca | kategorikal | Contoh: hujan, cerah, berawan, ekstrem, dll (bisa hasil scraping API BMKG) |
| event\_khusus | boolean/kateg. | Libur nasional, puasa, lebaran, dll |
| day\_of\_week | kategorikal | Hari dalam minggu, karena konsumsi bisa naik saat weekend |
| is\_hari\_libur | boolean | Apakah hari itu hari libur atau tidak |
| jumlah\_penduduk | numeric | Estimasi populasi (bisa disamakan per kota dan digunakan sebagai fitur statik) |
| permintaan\_cabai\_kg | numeric (label) | Label target (jumlah cabai dibeli/terjual/tersedia di pasar pada hari itu) |

KODE GENERATE PASOKAN  
  
import numpy as np

def simulasi\_pasokan(tanggal, total\_petani):

petani\_cabai = int(0.10 \* total\_petani)

panen\_hari\_ini = np.random.binomial(petani\_cabai, p=1/14) # 1 dari 14 hari

total\_kg = sum(np.random.randint(50, 150) for \_ in range(panen\_hari\_ini))

return total\_kg  
  
KODE GENERATE PERMINTAAN CABAI\_KG

import numpy as np

# Parameter

alpha = 0.05 # bobot pengaruh penduduk

beta = 2 # bobot pengaruh harga

gamma = 10 # bobot cuaca

delta = 1000 # tambahan permintaan jika event khusus

epsilon = 800 # tambahan permintaan jika hari libur

zeta = 500 # tambahan jika weekend

# Fungsi cuaca ke skor

def cuaca\_factor(cuaca):

return {

'cerah': 1.0,

'berawan': 0.8,

'hujan': 0.5,

'ekstrem': 0.2

}.get(cuaca, 0.8)

# Fungsi estimasi permintaan

def estimasi\_permintaan(penduduk, harga, cuaca, event\_khusus, is\_libur, day\_of\_week):

faktor\_cuaca = cuaca\_factor(cuaca)

noise = np.random.normal(0, 300)

permintaan = (

(alpha \* penduduk) -

(beta \* harga) +

(gamma \* faktor\_cuaca)

)

if event\_khusus:

permintaan += delta

if is\_libur:

permintaan += epsilon

if day\_of\_week in ['Sabtu', 'Minggu']:

permintaan += zeta

permintaan += noise

return max(0, round(permintaan)) # permintaan tidak boleh negatif