

# **Tugas Besar 1 IF3070 Dasar Inteligensi Artifisial Pencarian Solusi Pengepakan Barang (*Bin Packing Problem*) dengan Local Search**

Dipersiapkan Oleh Tim Asisten Lab AI '22



Deadline:  
Kamis, 30 Oktober 2025, 22.22 WIB

## **Tujuan**

Tugas Besar I pada kuliah IF3070 Dasar Inteligensi Artifisial bertujuan agar peserta kuliah mendapatkan wawasan tentang bagaimana cara mengimplementasikan dan mengevaluasi algoritma-algoritma local search untuk menyelesaikan masalah optimasi pengepakan barang (*bin packing problem*). Pada tugas ini, peserta akan merancang representasi solusi, mengimplementasikan beberapa varian algoritma local search, melakukan eksperimen, dan menganalisis hasilnya.

# Spesifikasi

Tugas ini bertujuan untuk menempatkan sekumpulan barang dengan ukuran yang berbeda-beda ke dalam sejumlah kontainer dengan kapasitas yang sama, dengan tujuan menggunakan jumlah kontainer sesedikit mungkin.

## Daftar Variabel

Entitas	Atribut	Contoh	Keterangan
Barang	ID Barang	BRG001	Kode unik untuk setiap barang.
	Ukuran	25	Ukuran atau berat barang (dalam unit yang sama, misal: kg atau m³).
Kontainer	Kapasitas	100	Setiap kontainer memiliki kapasitas yang sama dan seragam.

## Ketentuan Representasi

- Struktur data yang digunakan untuk representasi bebas. Misal, menggunakan list of lists.
- Representasi **state** adalah alokasi setiap Barang ke salah satu Kontainer. Contoh representasi: sebuah list yang berisi beberapa list lain, di mana setiap list internal merepresentasikan satu kontainer dan berisi ID barang di dalamnya.  
Contoh: [['BRG001', 'BRG005'], ['BRG002', 'BRG003'], ['BRG004']] berarti ada 3 kontainer yang digunakan.
- Inisialisasi (**initial state**) **bebas**, bisa secara **acak** atau menggunakan **algoritma heuristik** sederhana (misal: First Fit), di mana barang-barang ditempatkan satu per satu ke dalam kontainer.
- **Move** yang diperbolehkan tiap iterasi:
  1. Memindahkan satu barang dari satu kontainer ke kontainer lain (yang sudah ada atau yang baru).
  2. Menukar dua barang dari dua kontainer yang berbeda.

## Contoh Input

```
{
  "kapasitas_kontaine": 100,
  "barang": [
    { "id": "BRG001", "ukuran": 40 },
    { "id": "BRG002", "ukuran": 55 },
    { "id": "BRG003", "ukuran": 25 },
    { "id": "BRG004", "ukuran": 60 },
    { "id": "BRG005", "ukuran": 30 },
    { "id": "BRG006", "ukuran": 45 },
    { "id": "BRG007", "ukuran": 50 }
  ]
}
```

## Contoh Output

Total Kontainer Digunakan: 4

1. Kontainer 1 (Total: 95/100):
  - BRG002 (55)
  - BRG006 (45)
2. Kontainer 2 (Total: 100/100):
  - BRG004 (60)
  - BRG001 (40)
3. Kontainer 3 (Total: 55/100):
  - BRG003 (25)
  - BRG005 (30)
4. Kontainer 4 (Total: 50/100):
  - BRG007 (50).

## Objective Function

Tujuan utama adalah **meminimalkan jumlah kontainer yang digunakan**.

Berikut adalah acuan untuk membuat fungsi objektif:

1. **Penalti Kapasitas Berlebih:** Setiap kontainer yang total ukuran barangnya melebihi kapasitas harus diberi penalti yang sangat besar. Ini untuk memastikan solusi akhir adalah solusi yang valid.
2. **Skor Berdasarkan Jumlah dan Kepadatan Kontainer:** Fungsi ini bertujuan untuk memprioritaskan *state* yang menggunakan lebih sedikit kontainer dan lebih padat.

3. Seberapa berpengaruh setiap penalti atau reward terhadap objective function keseluruhan akan dibebaskan. **Setiap konsiderasi besar pengaruh harus disertai alasan.**

## Cakupan

Anda akan diminta untuk mengimplementasikan rencana yang telah Anda buat pada Tugas Besar 1. Berikut merupakan hal-hal yang perlu dilakukan oleh setiap kelompok:

- Implementasikan 3 algoritma local search dengan rincian sebagai berikut:
  - Salah satu algoritma hill-climbing
  - Simulated Annealing
  - Genetic Algorithm
- Lakukan eksperimen dengan skema sebagai berikut:
  - Jalankan setiap algoritma sebanyak **3 kali**, kemudian catat beberapa hal berikut:
    - Berlaku untuk semua algoritma
      - State awal dan akhir
      - Nilai objective function akhir yang dicapai
      - Plot nilai objective function terhadap banyak iterasi yang telah dilewati
      - Durasi proses pencarian
    - Berlaku hanya untuk Steepest Ascent Hill-Climbing dan Stochastic Hill-Climbing
      - Banyak iterasi hingga proses pencarian berhenti
    - Berlaku hanya untuk Hill-Climbing with Sideways Move
      - Banyak iterasi hingga proses pencarian berhenti
      - Note: Tambahkan parameter maximum sideways move, dimana ketika banyak sideways move yang dilakukan sudah mencapai maksimum, pencarian dihentikan
    - Berlaku hanya untuk Random Restart Hill-Climbing
      - Banyak restart
      - Banyak iterasi per restart
      - Note: Tambahkan parameter maximum restart, dimana ketika banyak restart sudah mencapai maksimum, pencarian dihentikan.
    - Berlaku hanya untuk Simulated Annealing
      - Plot  $e^{-\frac{\Delta E}{T}}$  terhadap banyak iterasi yang telah dilewati
      - Frekuensi 'stuck' di local optima
  - Khusus untuk Genetic Algorithm, lakukan beberapa hal berikut:

- Terdapat 2 parameter yang dapat diubah, yaitu **jumlah populasi** dan **banyak iterasi**.
- Jadikan jumlah populasi sebagai kontrol, kemudian pilih **3 variasi** banyak iterasi yang berbeda. Jalankan program sebanyak masing-masing **3 kali** untuk setiap konfigurasi parameter.
- Jadikan banyak iterasi sebagai kontrol, kemudian pilih **3 variasi** jumlah populasi yang berbeda. Jalankan program sebanyak masing-masing **3 kali** untuk setiap konfigurasi parameter.
- Untuk setiap eksperimen, catat beberapa hal berikut:
  - State awal dan akhir
  - Nilai objective function akhir yang dicapai
  - Plot nilai objective function terhadap banyak iterasi yang telah dilewati (Cukup plot nilai objective function maksimum dan rata-rata dari populasi terhadap banyak iterasi yang telah dilewati. Jika sudah terlanjur membuat plot untuk tiap individu pada populasi tidak menjadi masalah, silahkan diberikan keterangan saja maksud plotnya apa)
  - Pengaruh nilai probabilitas mutasi (**coba minimal 2**)
  - Jumlah populasi
  - Banyak iterasi
  - Durasi proses pencarian
- Lakukan analisis terhadap hasil eksperimen, berikut merupakan beberapa pertanyaan yang dapat menjadi acuan untuk analisis yang Anda lakukan (Anda boleh menambahkan beberapa pertanyaan tambahan jika dirasa perlu untuk dijelaskan):
  - Seberapa dekat tiap-tiap algoritma bisa mendekati global optima dan mengapa hasilnya demikian?
  - Bagaimana perbandingan hasil pencarian tiap-tiap algoritma dengan algoritma local search yang lain?
  - Bagaimana perbandingan durasi proses pencarian tiap algoritma relatif terhadap algoritma lainnya?
  - Seberapa konsisten hasil akhir yang didapatkan dari tiap-tiap eksperimen yang dilakukan?
  - Bagaimana pengaruh banyak iterasi dan jumlah populasi terhadap hasil akhir pencarian pada Genetic Algorithm?
  - dst...
- Program yang dibuat harus bisa **memvisualisasikan** state awal, state akhir, dan juga hasil eksperimennya (sesuaikan informasi hasil eksperimen yang ditampilkan dengan ketentuan yang telah dijelaskan di poin sebelum ini).
- Cara visualisasi dibebaskan kepada masing-masing kelompok.
- Gunakan bahasa **Python**.

- Diperbolehkan untuk menggunakan heuristik yang Anda buat sendiri atau dari referensi lain untuk optimasi pencarian solusi, asalkan masih dalam lingkup local search. Jangan lupa jelaskan heuristik yang Anda pakai di laporan.

## Spesifikasi Bonus

Hanya kerjakan spesifikasi ini jika Anda sudah menyelesaikan **seluruh spesifikasi wajib**. Berikut merupakan beberapa spesifikasi bonus yang dapat Anda kerjakan:

1. **Buatlah implementasi untuk seluruh algoritma hill-climbing yang diajarkan.**
2. **Tambahkan Aturan Batasan Antar Barang (*Item Constraints*).** Modifikasi masalah dengan menambahkan batasan baru yang membuat pengepakan lebih kompleks. Tambahkan aturan ini sebagai komponen dalam *objective function* Anda (misalnya, dengan menambahkan penalti baru). Anda dipersilahkan untuk **membuat batasan sendiri**.

**Contoh** batasan yang bisa ditambahkan:

- a. **Barang Rapuh (Fragile Items):** Tambahkan atribut **rapuh** pada beberapa barang. Sebuah barang rapuh tidak boleh ditempatkan dalam satu kontainer bersama barang lain yang total ukurannya melebihi nilai tertentu (misal, 50). Situasi Ini mensimulasikan barang rapuh yang tidak boleh tertindih beban berat.
- b. **Barang Tidak Kompatibel (Incompatible Items):** Tambahkan atribut **tipe** pada setiap barang (contoh: 'makanan', 'kimia', 'pakaian'). Buat aturan bahwa barang dengan tipe tertentu tidak boleh berada dalam satu kontainer yang sama (misalnya, 'makanan' tidak boleh digabung dengan 'kimia').

## Kelompok

Pembagian kelompok ditentukan sendiri oleh mahasiswa dengan mengisi [sheets kelompok IF3070 Dasar Inteligensi Artifisial](#) berikut ini dengan 1 kelompok terdiri dari **anggota sebanyak 3 orang**. Batas waktu pengisian kelompok adalah **Minggu, 5 Oktober 2025 pukul 22.22 WIB**. Setelah waktu yang ditentukan, mahasiswa yang belum mengisi sheets kelompok akan dikelompokkan secara acak.

## QnA

Pertanyaan dapat ditanyakan pada [link QnA](#) berikut. Pastikan pertanyaan yang ditanyakan tidak berulang.

## Aturan

Terdapat beberapa hal yang harus diperhatikan dalam pengerjaan tugas ini, yakni:

1. Jika terdapat hal yang tidak dimengerti, silahkan ajukan pertanyaan kepada asisten melalui **link QnA** yang telah diberikan di atas. Pertanyaan yang diajukan secara personal ke asisten **tidak akan dijawab** untuk menghindari perbedaan informasi yang didapatkan oleh peserta kuliah.
2. Dilarang melakukan **plagiarisme, menggunakan AI dalam bentuk apapun untuk men-generate jawaban Anda, dan melakukan kerjasama antar kelompok**. Pelanggaran pada poin ini akan menyebabkan pemberian **nilai E** pada setiap anggota kelompok.
3. **Konsekuensi keterlambatan: -5 poin tiap 10 menit.**

## Deliverables

- Tugas dikumpulkan dalam bentuk link ke *repository* GitHub yang **minimal** berisi beberapa hal berikut (boleh ditambahkan jika dirasa perlu):
  - Folder **src**, digunakan untuk menyimpan source code
  - Folder **doc**, digunakan untuk menyimpan laporan dalam bentuk **.pdf** yang terdiri atas komponen berikut:
    - Cover
    - Deskripsi Persoalan
    - Pembahasan
      - Pemilihan objective function
      - Penjelasan implementasi algoritma local search (berisi deskripsi fungsi/kelas beserta source codenya)
      - Hasil eksperimen dan analisis (disertai dengan visualisasi dari program yang telah dibuat)
    - Kesimpulan dan Saran
    - Pembagian tugas tiap anggota kelompok
    - Referensi
  - **README.md**, yang berisi deskripsi singkat repository, cara setup dan run program, dan pembagian tugas tiap anggota kelompok.
- Pengumpulan dilakukan melalui **edunex** dan gunakan [form ini](#) jika (dan hanya jika) terjadi masalah pada edunex dalam periode **3 jam sebelum deadline**.
- Batas akhir pengumpulan adalah hari **Kamis, 30 Oktober 2025 pukul 22.22 WIB**. Keterlambatan pengumpulan tugas akan menyebabkan pengurangan nilai.
- Repository baru boleh diubah ke public setelah **Kamis, 30 Oktober 2025 pukul 22.22 WIB**.

## Spesifikasi Tugas Besar 1 IF3070 Dasar Inteligensi Artifisial 2025/2026

---

- Pengumpulan dilakukan oleh **NIM terkecil**.
- Pengumpulan hanya dilakukan **sekali** untuk tiap kelompok.