

Tugas Besar 1 IF3170 Intelelegensi Buatan Pencarian Solusi Penjadwalan Kelas Mingguan dengan Local Search

Dipersiapkan Oleh Tim Asisten Lab AI '22



Deadline:
Minggu, 19 Oktober 2025, 22.22 WIB

Revisi 1: 1 Oktober 2025, 10.41 WIB

Tujuan

Tugas Besar I pada kuliah IF3170 Intelelegensi Buatan bertujuan agar peserta kuliah mendapatkan wawasan tentang bagaimana cara mengimplementasikan dan mengevaluasi algoritma-algoritma local search untuk menyelesaikan masalah penjadwalan kelas mingguan. Pada tugas ini peserta akan merancang representasi solusi, mengimplementasikan beberapa varian algoritma local search, melakukan eksperimen, dan menganalisis hasilnya.

Spesifikasi

Tugas ini bertujuan untuk membuat jadwal kelas selama satu minggu untuk sekumpulan mata kuliah yang harus dipasangkan dengan waktu dan tempat (ruang/kelas).

Daftar Variabel

Entitas	Atribut	Contoh	Keterangan
Kelas Mata kuliah	Kode	IF3071_K01	Kode mata kuliah dan kode kelas digabung.
	Jumlah mahasiswa terdaftar	60	Jumlah mahasiswa 60 hanya untuk satu kelas saja (dalam kasus ini untuk K01 sesuai kode di atas).
	Jumlah SKS	3	
Waktu	Waktu mulai	[Senin, 11]	Jam dalam integer. 1 SKS = 1 jam.
	Waktu akhir	[Selasa, 13]	Struktur data gabungan harus list .
Ruangan	Kode ruangan	7609	Untuk ruangan yang tidak memiliki kode, gunakan nama ruangan sebagai kode (misalnya “multimedia”).
	Kuota ruangan	60	Ruangan tersebut dapat menampung 60 mahasiswa.
Mahasiswa	NIM	13523999	
	Daftar mata kuliah	[IF3071_K01, IF9999_K99]	Bentuk data bebas, bisa menggunakan dictionary, dua list, atau implementasi lain.
	Prioritas mata kuliah	[1,2]	Elemen harus unik (satu orang hanya memiliki 1 matkul untuk prioritas)

		tertentu).
--	--	------------

Ketentuan Representasi

- Struktur data yang digunakan untuk representasi bebas. Misal, menggunakan key-value seperti JSON.
- Representasi solusi (**state**) adalah pemetaan setiap Pertemuan Mata Kuliah ke Waktu dan Ruangan.
- Inisialisasi (**initial state**) random.
- Langkah neighbor (**move**) yang diperbolehkan tiap iterasi:
 1. menukar posisi dua pertemuan mata kuliah
 2. memindahkan pertemuan mata kuliah ke waktu dan ruangan yang kosong

*Pertemuan Mata Kuliah merupakan kombinasi dari Kelas Mata Kuliah, Waktu, dan Ruangan sesuai dengan jumlah SKS yang dimiliki. Misalnya IF3071_K01 dengan 3 sks memiliki 2 Pertemuan Mata Kuliah, kelas di jam 7-9 hari Senin di ruang 7609 dan kelas di jam 9-10 hari Selasa di ruang multimedia.

Contoh Input

```
{
  "kelas_mata_kuliah": [
    {
      "kode": "IF3071_K01",
      "jumlah_mahasiswa": 60,
      "skls": 3
    },
    {
      "kode": "IF3130_K01",
      "jumlah_mahasiswa": 45,
      "skls": 2
    },
    {
      "kode": "IF3110_K02",
      "jumlah_mahasiswa": 70,
      "skls": 3
    },
    {
      "kode": "IF3140_K01",
      "jumlah_mahasiswa": 55,
      "skls": 2
    }
  ],
  "ruangan": [
    {
      "kode": "7609",
      "jumlah_pertemuan": 2
    }
  ]
}
```

```
"kuota": 60
},
{
  "kode": "7606",
  "kuota": 80
},
{
  "kode": "multimedia",
  "kuota": 40
}
],
"mahasiswa": [
  {
    "nim": "13523601",
    "daftar_mk": ["IF3071_K01", "IF3130_K01"],
    "prioritas": [1, 2]
  },
  {
    "nim": "135236641",
    "daftar_mk": ["IF3110_K02", "IF3130_K01"],
    "prioritas": [1, 2]
  },
  {
    "nim": "13523669",
    "daftar_mk": ["IF3140_K01", "IF3071_K01"],
    "prioritas": [1, 2]
  },
  {
    "nim": "13523600",
    "daftar_mk": ["IF3110_K02"],
    "prioritas": [1]
  }
]
```

Contoh Output

Kode ruang: 7609

Jam	Senin	Selasa	Rabu	Kamis	Jumat
7				IF3140	
8	IF3130		IF3110		IF3130
9	IF3130	IF3140	IF3110		
10		IF3140	IF3110		
11					

12					
13					
14					
15					
16					
17					

*kolom jam di output merupakan jam mulai

*contoh di atas merupakan output untuk satu ruangan.

Objective Function

Objective function yang digunakan **dibebaskan** (minimal 1), namun tetap berpedoman pada acuan berikut:

1. **Jumlah Waktu yang bertabrakan untuk tiap Mahasiswa**

Misalnya, mahasiswa A memiliki 2 mata kuliah di jam 9-11 hari Selasa akan berkontribusi 2 poin ke state tersebut.

2. **Jumlah Pertemuan Beberapa Mata Kuliah yang bertabrakan di Ruangan dan**

Waktu tertentu dikali dengan bobot prioritas Mata Kuliah tiap peserta kelas

Misalnya, IF2230_K01 dan IF1221_K01 sama-sama dilaksanakan di ruangan 7606 pada jam 13 - 14 . Bobot prioritas matkul adalah sebagai berikut.

Prioritas	Bobot
1	1.75
2	1.5
3	1.25
dll	1

Peserta kelas IF2330_K01 adalah sebagai berikut.

```
{
  "nim": "13523601",
  "daftar_mk": ["IF2330_K01"],
  "prioritas": [1]
},
```

```
{  
    "nim": "135236641",  
    "daftar_mk": ["IF2330_K01"],  
    "prioritas": [2]  
}
```

Sedangkan peserta kelas IF1221_K01 adalah sebagai berikut.

```
{  
    "nim": "13523701",  
    "daftar_mk": ["IF1221_K01"],  
    "prioritas": [1]  
},  
{  
    "nim": "135237641",  
    "daftar_mk": ["IF1221_K01"],  
    "prioritas": [3]  
}
```

Maka state valuenya akan bertambah/berkurang (tergantung implementasi) sebanyak
 $1 \text{ jam} * (1.75 \text{ bobot prioritas 1} * 2 \text{ jumlah mahasiswa prioritas 1} + 1.5 \text{ bobot prioritas 2} * 1 \text{ jumlah mahasiswa prioritas 2} + 1.25 \text{ bobot prioritas 3} * 1 \text{ jumlah mahasiswa prioritas 3} + 1 \text{ bobot prioritas lainnya} * 0 \text{ jumlah mahasiswa prioritas lainnya}) = 6.25$

3. **Selisih Jumlah Mahasiswa di Pertemuan Mata Kuliah tertentu dengan Kuota Ruangan tersedia (dalam kasus Jumlah Mahasiswa > Kuota Ruangan)**

Misalnya terdapat 65 mahasiswa di ruangan 7609 (Kuota Ruangan: 60) di suatu Pertemuan Mata Kuliah **2** SKS akan berkontribusi 10 poin $((65 - 60) * 2)$ ke state tersebut.

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)
 - 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 eksperimentnya (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:

- Buatlah implementasi untuk **seluruh algoritma hill-climbing** yang ada.
- Tambahkan **jadwal dosen** sebagai komponen objective function (jadwal dosen dikreasikan sendiri).

Kelompok

Pembagian kelompok ditentukan sendiri oleh mahasiswa dengan mengisi [sheets kelompok IF3170 Inteligensi Artifisial](#) berikut ini dengan 1 kelompok terdiri dari **anggota sebanyak 3 orang**. Batas waktu pengisian kelompok adalah **Jumat, 3 Oktober 2025 pukul 22.22 WIB**. Setelah waktu yang ditentukan, mahasiswa yang belum mengisi sheets kelompok akan diacak.

QnA

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

Aturan

Terdapat beberapa hal yang harus diperhatikan dalam penggerjaan 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.

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.

- Batas akhir pengumpulan adalah hari **Minggu, 19 Oktober 2025 pukul 22.22 WIB**. Keterlambatan pengumpulan tugas akan menyebabkan pengurangan nilai.
- Repository baru boleh diubah ke public setelah **Minggu, 19 Oktober 2025 pukul 22.22 WIB**.
- Pengumpulan dilakukan oleh **NIM terkecil**.
- Pengumpulan hanya dilakukan **sekali** untuk tiap kelompok.