

# Perencanaan Rute UAV Ulang-Alik untuk Mengantisipasi Timbulnya Klaster Kerumunan COVID-19 dengan Pendekatan Program Dinamis

Leonard Matheus Wastupranata - 13519215

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: 13519215@std.stei.itb.ac.id

**Abstrak**—Kerumunan menjadi hal yang dianggap sepele oleh masyarakat karena merasa telah menjalankan protokol kesehatan dengan menggunakan masker. Kerumunan harus diminimalisasi agar penyebaran virus covid-19 tidak semakin tinggi. Makalah ini bertujuan untuk merencanakan rute UAV ulang-alik agar dapat menghampiri sebanyak mungkin lokasi yang berpotensi terjadi kerumunan sekaligus mengarah ke tujuan akhir rute terbang tanpa harus berputar-putar dahulu. Metode yang digunakan adalah perbandingan algoritma Greedy dan Program Dinamis dalam menentukan rute yang paling efektif. Simulasi terbang dilakukan menggunakan Software in The Loop (SITL) dan ArduPilot Mission Planner. Hasil yang didapatkan yakni Program Dinamis dapat mengunjungi 14 lokasi dari 18 pilihan lokasi yang ada, sedangkan dengan pendekatan algoritma Greedy, UAV hanya dapat mengunjungi 8 lokasi dari 18 pilihan lokasi yang ada. Kesimpulannya adalah Program Dinamis mampu memaksimalkan rute agar lokasi yang dikunjungi oleh UAV lebih banyak dan tentunya lebih baik dari Algoritma Greedy

**Kata Kunci**—UAV; Program Dinamis; Algoritma Greedy; COVID-19; Kerumunan

## I. PENDAHULUAN

Pandemi Covid-19 telah memberikan banyak perubahan dalam kehidupan dunia. Banyak tradisi dan kebudayaan yang harus menghadapi segala bentuk penyesuaian guna hidup berdampingan dengan virus Covid-19 yang penyebarannya tak kunjung selesai. Waktu terus berjalan dan kehidupan tidak boleh berhenti bergerak.

Fase dalam kehidupan manusia akan cepat berganti, terutama dalam garis kehidupan mahasiswa Institut Teknologi Bandung yang sedang menjalani masa perkuliahan. Rencana perkuliahan *hybrid* pun terus digaungkan demi memenuhi segala kekurangan dalam menjalani kuliah daring.

Akan tetapi, mahasiswa harus mengikuti protokol kesehatan yang telah dirancang ITB, terutama dalam hal meminimalisasi kerumunan. Sayangnya, tidak diketahui apakah ada seseorang di tengah kerumunan tersebut yang dinyatakan positif terdapat virus covid-19 dalam tubuhnya atau tidak. Untuk itu, diperlukan adanya monitoring dan penyemprotan desinfektan secara berkala pada jam-jam sibuk di tengah aktivitas kampus yang padat.

Penelitian mengenai pencegahan covid-19 sudah banyak dilakukan, salah satunya yaitu mencegah kerumunan dengan penyemprotan desinfektan. Efelina dkk (2021) [1] telah melakukan penelitian mengenai penggunaan drone untuk menyemprotkan desinfektan pada daerah pedesaan. Akan tetapi, kontrol drone yang dilakukan masih menggunakan cara manual dan masih kurang efektif apabila ternyata kerumunan tercipta dalam jumlah yang banyak. Oleh karena itu, diperlukan sebuah strategi algoritma yang bisa mengoptimalkan penyemprotan desinfektan secara berkala.

Pada makalah ini, diberikan sebuah perbandingan strategi jangkauan UAV untuk mencegah kerumunan dengan melakukan perbandingan antara algoritma greedy dengan program dinamis. Semakin banyak lokasi yang terjangkau secara berkala oleh UAV, kerumunan dapat cepat diminimalisasi. Akan tetapi, dikarenakan daya UAV yang terbatas, maka rute terbang akan dilakukan ulang-alik dari lokasi tertentu. Tugas UAV adalah mencari rute ulang-alik yang akan memaksimalkan lokasi kerumunan yang akan dikunjungi sekaligus mengarah kepada tujuan akhirnya.

Makalah ini akan terbagi menjadi lima bagian besar. Bagian Pendahuluan akan memaparkan latar belakang dari masalah yang timbul. Berikutnya, bagian Studi Literatur akan menjelaskan dasar-dasar strategi algoritma dalam melakukan eksperimen terbang. Bagian yang tak kalah pentingnya yakni metode, akan dibahas mengenai langkah-langkah penyelesaian masalah dengan pendekatan algoritma greedy dan program dinamis. Bagian Hasil dan Pembahasan akan memaparkan penjelasan mengenai simulasi terbang dan analisis strategi. Terakhir, bagian Kesimpulan akan menyatukan segala kerangka berpikir dan eksperimen yang telah dilakukan pada makalah ini.

## II. STUDI LITERATUR

### A. Algoritma Greedy

Algoritma greedy adalah algoritma yang memecahkan persoalan secara langkah per langkah (step by step) sedemikian sehingga, pada setiap langkah mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan (prinsip “take what you

can get now!”) dan “berharap” bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global.

Algoritma ini merupakan metode yang paling populer dan sederhana untuk memecahkan persoalan optimasi. Algoritma ini mengedepankan persoalan untuk mencari solusi optimal. Ada dua macam persoalan optimasi, yakni persoalan maksimasi dan minimasi. Adapun skema umum algoritma greedy adalah seperti yang terlihat di bawah ini. [2]

```

function greedy (C : himpunan_kandidat)- himpunan_solusi
{ Mengembalikan solusi dari persoalan optimasi dengan
algoritma greedy }
Deklarasi
x : kandidat
S : himpunan_solusi

Algoritma:
S ← {} { inialisasi S dengan kosong }
while (not SOLUSI(S)) and (C ≠ {}) do
x ← SELEKSI(C) { pilih sebuah kandidat dari C }
C ← C - {x} { buang x dari C karena sudah dipilih }
if LAYAK(S ∪ {x}) then { x memenuhi kelayakan untuk
dimasukkan ke dalam himpunan solusi }
S ← S ∪ {x} { masukkan x ke dalam himpunan solusi }
endif
endwhile
{SOLUSI(S) or C = {}}
if SOLUSI(S) then { solusi sudah lengkap }
return S
else
write('tidak ada solusi')
endif
    
```

Algoritma Greedy ini hampir sama dengan metode exhaustive search dan brute force, dimana Exhaustive search adalah teknik pencarian solusi secara brute force pada masalah yang melibatkan pencarian elemen dengan sifat khusus, biasanya di antara objek-objek kombinatorik seperti permutasi, kombinasi, atau himpunan bagian dari sebuah himpunan. Berdasarkan definisi ini, maka exhaustive search adalah brute force juga. Oleh karena itu, exhaustive search adalah salah satu implementasi dalam brute force dalam kasus pencarian. [3]

**B. Program Dinamis**

Program Dinamis (dynamic programming) adalah metode pemecahan masalah dengan cara menguraikan solusi menjadi sekumpulan tahapan (stage) sedemikian sehingga solusi persoalan dapat dipandang sebagai serangkaian keputusan yang saling berkaitan. Program dinamis digunakan untuk menyelesaikan persoalan-persoalan optimasi (maksimasi atau minimisasi).

Pada program dinamis, rangkaian keputusan yang optimal dibuat dengan menggunakan Prinsip Optimalitas. Menurut Prinsip Optimalitas, jika solusi total optimal, maka bagian solusi sampai tahap ke-k juga optimal. Prinsip optimalitas berarti bahwa jika kita bekerja dari tahap k ke tahap k + 1, kita

dapat menggunakan hasil optimal dari tahap k tanpa harus kembali ke tahap awal.

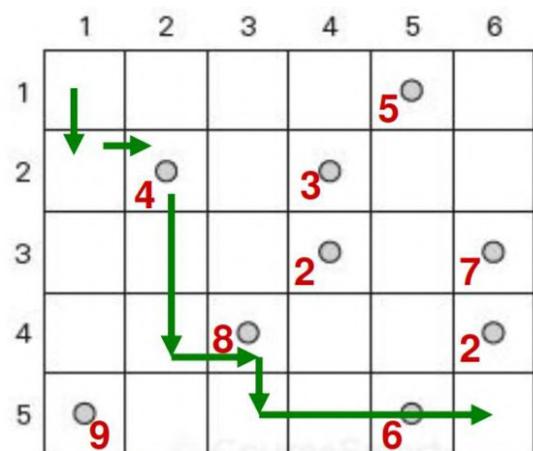
Karakteristik Persoalan dengan Program Dinamis adalah sebagai berikut:

1. Persoalan dapat dibagi menjadi beberapa tahap (stage), yang pada setiap tahap hanya diambil satu keputusan.
2. Masing-masing tahap terdiri dari sejumlah status (state) yang berhubungan dengan tahap tersebut. Secara umum, status merupakan bermacam kemungkinan masukan yang ada pada suatu tahap.
3. Hasil dari keputusan yang diambil pada setiap tahap ditransformasikan dari status yang bersangkutan ke status berikutnya pada tahap berikutnya.
4. Ongkos (cost) pada suatu tahap meningkat secara teratur (steadily) dengan bertambahnya jumlah tahapan.
5. Ongkos pada suatu tahap bergantung pada ongkos tahap-tahap yang sudah berjalan dan onkos dari tahap tersebut ke tahap berikutnya.
6. Adanya hubungan rekursif yang mengidentifikasi keputusan terbaik untuk setiap status pada tahap k memberikan keputusan terbaik untuk setiap status pada tahap k + 1.
7. Prinsip optimalitas berlaku pada persoalan tersebut.[4]

**C. Coin-collecting problem**

Pada permasalahan ini, beberapa koin ditempatkan dalam sel  $n \times m$  papan, tidak lebih dari satu koin per sel. Robot dimulai dari kiri atas papan sel. Misi dari permasalahan ini adalah robot harus mengumpulkan koin sebanyak mungkin dan finish di kanan bawah dari papan sel.

Pada setiap langkah, robot bisa bergerak baik satu sel ke kanan atau ke bawah dari lokasinya saat ini. Saat robot mengunjungi sel yang berisi koin, robot akan selalu mengambil koin tersebut. Robot harus menemukan jumlah maksimum koin yang dapat diambil dan menentukan jalur yang harus dilewati agar efisien.[5]



**Gambar 1.** Coin-collecting Problem

(Sumber:<http://www.jsuums.edu/nmeghanathan/files/2016/08/CSC323-Fall2016-Module-4-DynamicProgramming.pdf>)

### III. METODE

#### A. Persiapan Perencanaan Rute (Mission Planner)

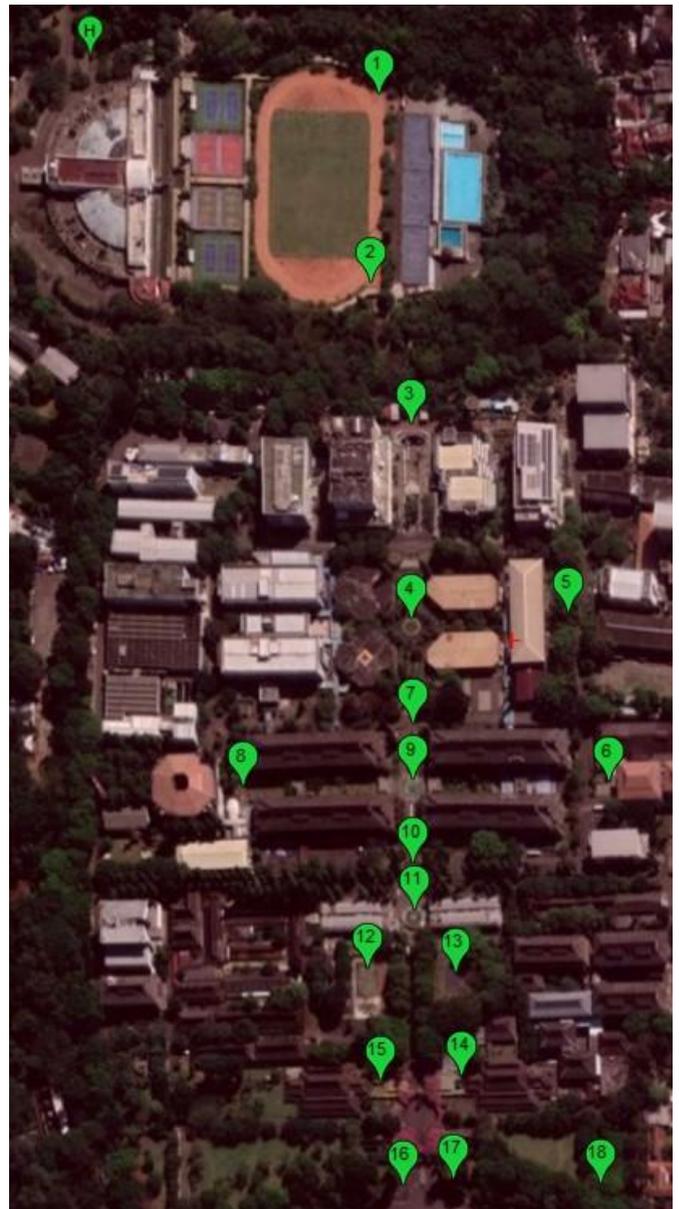
Terdapat beberapa titik potensi kerumunan yang diproyeksikan akan bermunculan di dalam kampus ITB Ganesha, lokasinya dapat dilihat pada tabel di bawah ini.

TABEL 1. LOKASI RAWAN KERUMUNAN DALAM LINGKUNGAN KAMPUS

Nomor Lokasi	Lokasi Spesifik		
	Nama Lokasi	Latitude	Longitude
HOME	Parkir Saraga	-6.885423	107.608179
1	Jam Digital Saraga	-6.8856415	107.6101238
2	Kantin Saraga	-6.8869463	107.6100782
3	Pintu Utara ITB Ganesha	-6.8878970	107.6103544
4	Bundaran Oktagon-TVST	-6.8892657	107.6103705
5	Selasar Labtek 1	-6.8892151	107.6114380
6	GKU Timur	-6.8903628	107.6117063
7	DPR	-6.8899660	107.6103625
8	GKU Barat	-6.8903920	107.6091984
9	Kolam Intel	-6.8903521	107.6103678
10	Tugu Soekarno	-6.8909166	107.6103759
11	CTim-CBar	-6.8912335	107.6103839
12	Lapangan Basket	-6.8916649	107.6100540
13	Lapangan Cinta	-6.8916702	107.6106602
14	Selasar Aula Timur	-6.8923972	107.6106924
15	Selasar Aula Barat	-6.8924291	107.6102981
16	Pintu Selatan ITB sisi barat	-6.8931294	107.6102981
17	Pintu Selatan ITB sisi timur	-6.8930895	107.6106495
18	Parkir Timur Umum	-6.8931294	107.6116392

Ada delapan belas lokasi yang dapat dikunjungi untuk dilakukan penyebaran disinfektan agar membantu pensterilan di saat ada kerumunan. Akan tetapi, karena pada malam hari dan pagi hari telah dilakukan penyemprotan secara menyeluruh, UAV akan terbang start dari HOME menuju titik terakhir yaitu Parkir Timur Umum dengan ketinggian relatif 100 meter terhadap tanah. Akan ditentukan rute penerbangan sehingga UAV dapat mengunjungi potensi kerumunan sebanyak mungkin, dengan bobot tiap lokasi bernilai sama (untuk kasus ini disimbolkan dengan angka 1).

Simulasi terbang akan dilakukan menggunakan Software in the Loop (SITL) dengan Pustaka Ardupilot dan Mission Planner versi 1.3.74. Seperti terlihat pada **Gambar 2**, tersebar lokasi potensi kerumunan yang terdapat pada **Tabel 1**. Karena UAV akan terbang dari titik HOME yang berada di barat laut dan menuju Simpul Tujuan Parkir Timur Umum ITB yang berada di tenggara, UAV akan condong melewati ITB Ganesha dengan arah diagonal.



**Gambar 2.** Lokasi Rawan Kerumunan dalam Lingkungan Kampus

#### B. Elemen pada Algoritma Greedy

Dalam menyelesaikan permasalahan ini, solusi akan diabstraksikan ke dalam elemen-elemen pada algoritma Greedy sebagai berikut:

##### 1) Himpunan kandidat, $C$

Berisi kandidat simpul yang akan dipilih pada setiap langkah, minimal berisi satu buah kandidat simpul. Simpul yang dipilih hanya boleh berada di kanan atau bawah dari simpul sebelumnya. Dalam hal ini, simpul yang akan dipilih akan menjadi representasi lokasi yang berpotensi menimbulkan kerumunan.

##### 2) Himpunan solusi, $S$

Simpul-simpul yang sudah terpilih sebelumnya. Dalam kasus ini,  $S$  merupakan lokasi yang telah dikunjungi dan telah dilakukan penyemprotan disinfektan.

### 3) Fungsi solusi

Fungsi ini akan memeriksa apakah simpul yang dipilih merupakan simpul goal (Parkir Timur ITB) atau bukan.

### 4) Fungsi seleksi (selection function)

Simpul yang dipilih merupakan simpul yang berada di sekitarnya (baik di kanan maupun bawah simpul sebelumnya) dengan jarak terdekat.

### 5) Fungsi kelayakan (feasible)

Memeriksa apakah simpul yang baru dipilih membuat pergerakan UAV condong bergerak ke tenggara (sebagai akibat dari pemilihan lokasi yang hanya berada di kanan atau bawah lokasi sebelumnya)

### 6) Fungsi obyektif

Didapatkan rute terbang UAV dengan lokasi yang akan dikunjungi sebanyak mungkin.

## C. Elemen Pengembangan pada Program Dinamis

### 1) Karakteristikkan struktur solusi optimal.

Misalkan  $F(i, j)$  adalah angka terbesar dari permasalahan abstraksi lokasi yang akan dikunjungi pada sel  $(i, j)$  pada baris ke- $i$  dan kolom ke- $j$ . Sel tersebut dapat dijangkau dari sel  $(i-1, j)$  hingga di atasnya atau dari sel  $(i, j-1)$  hingga di sekitar kirinya.

Tentunya, pada permasalahan ini, tidak ada baris lain di atasnya pada sel di baris pertama dan tidak ada kolom lain di sebelah kirinya pada kolom di sel pertama. Diasumsikan  $F(i-1, j)$  dan  $F(i, j-1)$  sama dengan angka 0 untuk lokasi yang tidak berpotensi menimbulkan kerumunan.

### 2) Definisikan secara rekursif nilai solusi optimal.

$$\begin{aligned} F(0, j) &= 0, 1 \leq j \leq m && \text{basis} \\ F(i, 0) &= 0, 1 \leq i \leq n && \text{basis} \\ F(i, j) &= \max\{F(i-1, j), F(i, j-1)\} + c_{ij} \end{aligned}$$

untuk  $1 \leq i \leq n, 1 \leq j \leq m$

**rekurens**

Keterangan:

$F(i, j)$ : merupakan nilai dari baris ke- $i$  dan kolom ke- $j$ , bisa saja merupakan sel yang akan dipilih selanjutnya.

$n$ : jumlah baris tabel penyelesaian

$m$ : jumlah kolom tabel penyelesaian

$c_{ij}$ : konstanta, pada kasus ini berfungsi sebagai boolean ada

tidaknya pin lokasi yang ada pada tabel (angka 0 apabila tidak ada lokasi yang terdefinisi, angka 1 apabila ada lokasi yang terdefinisi)

### 3) Hitung nilai solusi optimal secara maju.

Sebelum melakukan perhitungan mengenai solusi optimal, akan dibuat tabel yang berisi pembagian baris dan kolom dari ekstraksi koordinat masing-masing lokasi. Pembagian ini didasarkan pada aproksimasi untuk beberapa skala latitude dan longitude tertentu dan memaksimalkan ruang kosong agar dekomposisi sel menjadi lebih efektif.



Gambar 3. Dekomposisi tiap lokasi menjadi sel-sel kecil

Seperti terlihat pada Gambar 3, Terdapat  $5 \times 11$  sel yang tergabung pada suatu tabel representasi lokasi. Satu sel hanya maksimal diisi oleh satu pin lokasi saja, sehingga perhitungan tiap tahap nantinya tepat sasaran dan tidak akan mengalami bias yang berlebihan. Selain itu, posisi HOME dan simpul tujuan dari UAV juga akan tergabung dalam perhitungan pada saat terbang.

Dekomposisi di atas dapat diterjemahkan dalam bentuk tabel biner yang merepresentasikan lokasi yang akan dikunjungi sebanyak mungkin. Pada kasus ini, tiap pin akan diwakili oleh angka 0 atau 1 untuk merepresentasikan keberadaan lokasi kerumunan pada tabel.

TABEL 2. TABEL BINER REPRESENTASI LOKASI YANG AKAN DIKUNJUNGI

(i, j)	0	1	2	3	4
0	1	0	1	0	0
1	0	0	1	0	0
2	0	0	1	0	0
3	0	0	1	0	1
4	0	0	1	0	0
5	0	1	1	0	1
6	0	0	1	0	0
7	0	0	1	0	0
8	0	0	1	1	0
9	0	0	1	1	0
10	0	0	1	1	1

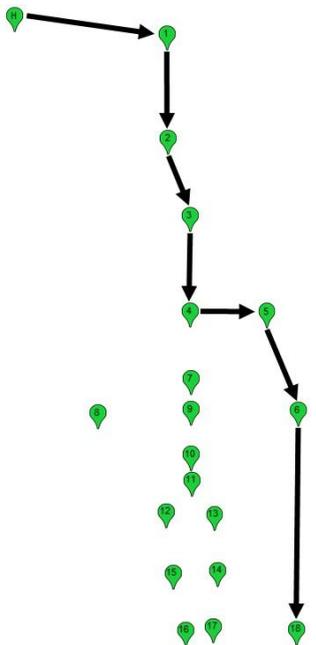
4) Rekonstruksi solusi optimal.

Solusi yang didapatkan pada tabel penyelesaian akan digunakan untuk merekonstruksi rute yang harus dilalui UAV untuk mendapatkan rute yang optimal.

D. Penyusunan Waypoint

1) Penyusunan dengan Algoritma Greedy

Karena prinsip algoritma Greedy adalah mencari rute terpendek terlebih dahulu yang memprioritaskan arah kanan atau bawahnya, pola pergerakan akan terlihat seperti berikut:



Gambar 4. Penyusunan Waypoint dengan Algoritma Greedy

Dengan demikian, akan dihasilkan rute dalam angka waypoint sebagai berikut: HOME → 1 → 2 → 3 → 4 → 5 → 6 → 18

2) Penyusunan dengan Program Dinamis

Perhitungan dilakukan untuk mendapatkan tempat yang akan menyempotkan disinfektan sebanyak mungkin. Tabel perhitungan solusi optimal dapat dilihat di bawah ini.

TABEL 3. TABEL PENGOLAHAN LOKASI YANG AKAN DIKUNJUNGI

(i, j)	0	1	2	3	4
0	1	1	2	2	2
1	1	1	3	3	3
2	1	1	4	4	4
3	1	1	5	5	5
4	1	1	6	6	6
5	1	2	7	7	7
6	1	2	8	8	8
7	1	2	9	9	9
8	1	2	10	10	11
9	1	2	11	12	12
10	1	2	12	13	14

Dengan adanya tabel pengolahan ini, solusi akan direkonstruksi dari simpul tujuan dengan mencari angka disekitar simpul setelahnya yang berselisih satu, dimana akan dicari simpul yang angkanya lebih kecil dari dirinya. Karena pergerakan diprioritaskan ke kanan terlebih dahulu, apabila simpul di atas dan kirinya bernilai sama, akan dicari konstruksi simpul di sebelah kiri dirinya terlebih dahulu.

Dengan demikian, akan dihasilkan rute dalam angka waypoint sebagai berikut: HOME → 1 → 2 → 3 → 4 → 7 → 9 → 10 → 11 → 12 → 15 → 14 → 17 → 18

E. Environment Pendukung Simulasi

Program akan dijalankan pada Sistem Operasi Ubuntu versi 18.04 karena sebagian besar aplikasi akan dijalankan pada lingkungan Linux. Untuk Visualisasi, digunakan aplikasi Gazebo (Gambar 5) sehingga perilaku terbang UAV dapat terlihat. UAV tersebut akan terbang mengikuti jarak yang ada pada GPS sehingga visualisasi waypoint yang dilewati dapat terlihat dengan baik.[6]



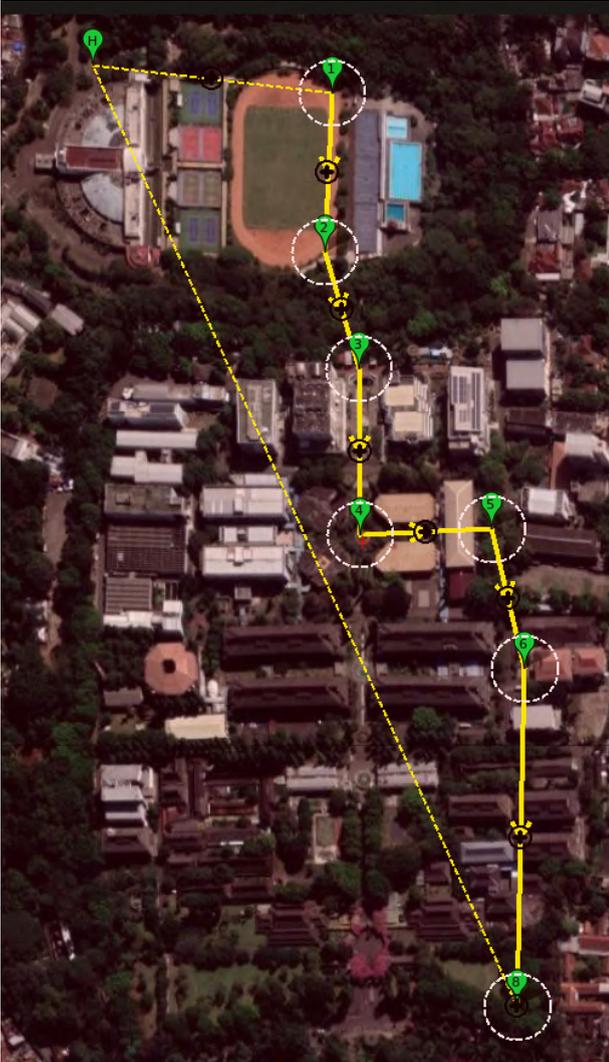
Gambar 5. Tampilan Drone lewat aplikasi Gazebo secara realtime

#### IV. HASIL DAN PEMBAHASAN

##### A. Hasil

Telah dilakukan simulasi terkait perilaku terbang UAV yang dirancang untuk menghampiri lokasi tempat berkerumun. Rencana Rute Terbang akan disusun sedemikian rupa oleh aplikasi sehingga nantinya UAV akan terbang sesuai algoritma yang telah disusun sebelumnya.

Eksperimen perencana rute terbang dengan Algoritma Greedy menghasilkan 8 buah waypoint yang termasuk HOME itu sendiri, penyusunan rute secara spesifik dapat dilihat pada **Gambar 6**.



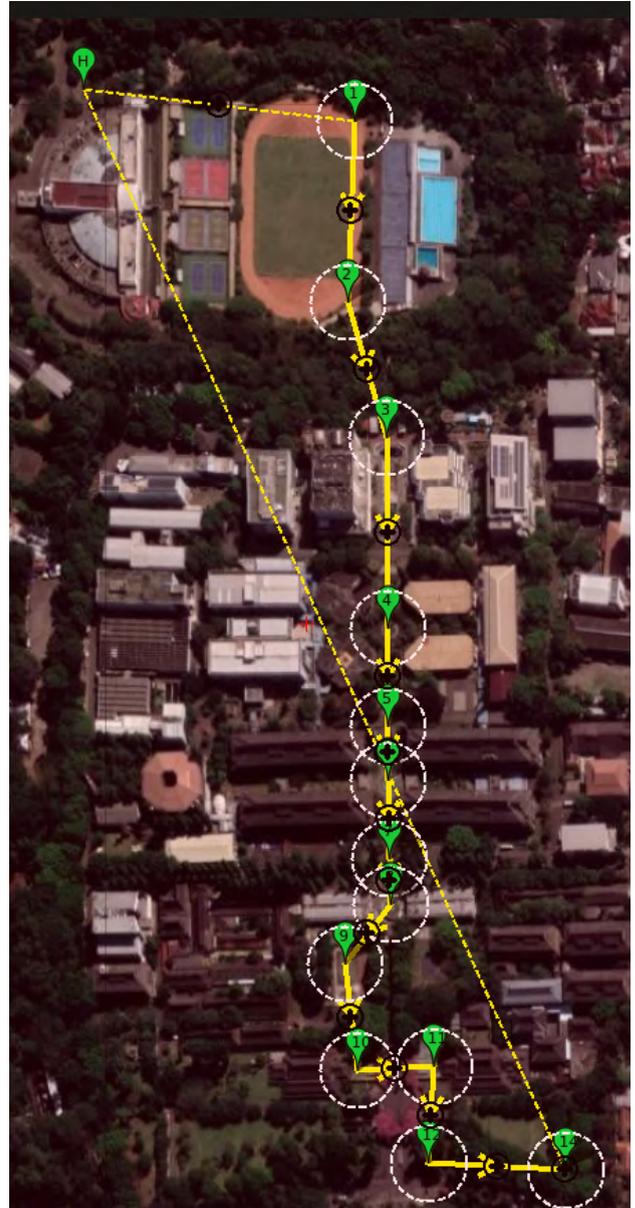
**Gambar 6.** Rute dengan pendekatan Algoritma Greedy

Dengan pendekatan ini, UAV dapat mengunjungi 6 lokasi yang berpotensi menimbulkan kerumunan, yakni Jam Digital Saraga, Kantin Saraga, Pintu Utara ITB Ganesha, Bundaran Oktagon-TVST, Selasar Labtek 1, dan GKU Timur.

Total waktu yang digunakan UAV untuk terbang mulai dari Take Off hingga sampai ke parkir Timur umum adalah 4 menit 30 detik. Baterai yang digunakan untuk melakukan sekali rute terbang adalah 9,2644 Volt dari kapasitas baterai 12,19 Volt.

Jarak Tempuh yang dilalui UAV start dari HOME hingga sampai ke Parkir Timur Umum ITB sejauh 1.205 meter.

Pada Eksperimen berikutnya, dilakukan Eksperimen menggunakan pendekatan Program Dinamis. Eksperimen ini menghasilkan 14 buah waypoint yang termasuk HOME itu sendiri, penyusunan rute secara spesifik dapat dilihat pada **Gambar 7**.



**Gambar 7.** Rute dengan pendekatan Program Dinamis

Dengan pendekatan ini, UAV dapat mengunjungi 12 lokasi yang berpotensi menimbulkan kerumunan, yakni Jam Digital Saraga, Kantin Saraga, Pintu Utara ITB Ganesha, Bundaran Oktagon-TVST, DPR, Kolam Intel, Tugu Soekarno, CTim-CBar, Lapangan Basket, Selasar Aula Barat, Selasar Aula Timur, dan Pintu Selatan ITB sisi timur.

Total waktu yang digunakan UAV untuk terbang mulai dari Take Off hingga sampai ke parkir Timur umum adalah 4 menit 30 detik. Baterai yang digunakan untuk melakukan sekali rute

terbang adalah 9,752 Volt dari kapasitas baterai 12,19 Volt. Jarak Tempuh yang dilalui UAV start dari HOME hingga sampai ke Parkir Timur Umum ITB sejauh 1.280 meter.

Waktu, jarak, dan besarnya baterai yang dipakai dihitung hanya untuk perjalanannya, belum termasuk waktu untuk menyemprotkan desinfektan di tengah kerumunan. Detail lebih lengkap mengenai hasil eksperimen dapat dilihat pada **Tabel 4**.

TABEL 4. HASIL EKSPERIMEN RUTE TERBANG UAV

No	Parameter	Algoritma	
		Greedy	Program Dinamis
1.	Jumlah Lokasi yang dikunjungi	6 Lokasi	14 Lokasi
2.	Total Waktu Terbang	4'30"	4'30"
3.	Total Jarak Terbang	1.205 m	1.280 m
4.	Baterai yang Terpakai	9,2644 V	9,752 V

### B. Pembahasan

Eksperimen dalam menentukan Rute yang Efektif untuk mengunjungi lokasi yang berpotensi menimbulkan kerumunan telah dilakukan. Hal yang menarik adalah jumlah lokasi yang dihampiri oleh UAV untuk algoritma yang berbeda.

Pada Algoritma Greedy, akan dicari simpul di sekitarnya yang memiliki jarak terdekat, tidak peduli apakah pada tahap di depan ternyata hasilnya tidak optimal. Akibatnya, pada waypoint nomor 4, kesalahan pengambilan keputusan ini turut mengurangi lokasi yang dapat dihampiri oleh UAV. Sebaliknya, pada Algoritma Program Dinamis, akan dicari optimasi simpul pada tahap tertentu yang bernilai maksimum. Tentunya pertimbangan ini didasarkan bukan hanya jarak terdekat, melainkan solusi optimum untuk keseluruhan kasus.

Pada tahap dekomposisi tiap lokasi dengan menggunakan Algoritma Program Dinamis, waypoint yang diambil terlihat berbelok ke kiri padahal batasan pada kasus ini tidak diperbolehkan. Hal ini terjadi lantaran program dinamis akan membagi setiap permasalahan menjadi sub-masalah yang lebih kecil sehingga untuk toleransi sekian derajat, longitude dan latitude yang ada pada gambar akan menjadi satu kolom yang padu dan sejajar.

Selanjutnya, untuk parameter perbandingan total waktu terbang, hal yang menarik adalah selisih waktu di antara keduanya sangat kecil, sehingga terlihat bahwa permasalahan optimasi yang ditekankan adalah seberapa banyak lokasi yang dapat dihampiri oleh UAV untuk mencegah potensi munculnya kerumunan di tengah-tengah kampus.

Total Jarak Terbang antara kedua eksperimen memiliki selisih 75 meter. Penentuan rute terbang dengan menggunakan algoritma greedy membuktikan bahwa persoalan optimasi jarak sangat diprioritaskan untuk lokasi di sekitarnya yang terdekat dengan lokasi sebelumnya. Eksperimen dengan pendekatan program dinamis menghasilkan total jarak terbang yang lebih jauh dikarenakan berkorespondensi dengan banyaknya lokasi yang sudah dikunjungi.

Baterai yang terpakai pada UAV untuk kedua eksperimen menunjukkan adanya perbedaan dengan selisih sekitar 0,5

Volt. Eksperimen dengan pendekatan Program Dinamis mengonsumsi lebih banyak baterai karena lokasi yang dikunjungi juga lebih banyak dibanding menggunakan algoritma Greedy. Artinya, untuk parameter baterai, nilainya berkorespondensi dengan parameter lokasi yang dikunjungi seperti terlihat pada **Tabel 4**.

### V. KESIMPULAN

Berkerumun atau berkumpul merupakan hal yang wajib dihindari karena penyebaran virus Covid-19 akan semakin tinggi apabila hal tersebut dianggap sepele. Dengan adanya hal ini, UAV dapat menjadi solusi untuk melakukan pemantauan dan penyemprotan desinfektan guna melakukan sterilisasi bagi potensi kerumunan tersebut. Untuk itu, dibutuhkan strategi yang efektif agar UAV dapat menjangkau banyak lokasi semaksimal mungkin dengan metode ulang-alik.

Semakin banyak lokasi yang akan menjadi potensi titik berkerumun, akan semakin banyak alternatif solusi UAV untuk menghampiri sebanyak mungkin titik-titik yang ada. Untuk itu, strategi algoritma penentuan rute UAV untuk mencegah titik berkumpul dilihat dari parameter banyak lokasi, jarak, waktu, dan daya baterai yang digunakan. Penentuan rute UAV untuk Pencegahan Titik Berkumpul lebih baik menggunakan Program Dinamis ketimbang Algoritma Greedy.

### VI. SARAN

Saran bagi peneliti selanjutnya adalah dapat dikembangkan lebih lanjut implementasi UAV dengan menggunakan Robotic Operating System (ROS) dan HITL (Hardware In The Loop). Selain itu, tabel dekomposisi Program Dinamis dapat dikembangkan dengan memberikan bobot untuk setiap intensitas kerumunan yang berbeda agar UAV dapat memprioritaskan kerumunan dengan intensitas tertinggi harus lebih dahulu dikunjungi dan dilakukan penyemprotan desinfektan.

Disarankan juga untuk melakukan perbandingan selain Algoritma Greedy, misalnya Algoritma Branch & Bound, Algoritma A\*, dan lain-lain. Hal ini diperlukan karena untuk kasus yang berbeda, diharapkan dapat ditemukan solusi terbaik agar penentuan rute UAV semaksimal mungkin mencapai lokasi kerumunan dan melakukan pencegahan secara efektif dan efisien.

VIDEO LINK AT YOUTUBE

<https://youtu.be/22X-2E10tM>

UCAPAN TERIMA KASIH

Pertama-tama, Saya mengucapkan syukur kepada Tuhan Yang Maha Esa karena dengan rahmatNya, Makalah berjudul "Perencanaan Rute UAV Ulang-Alik untuk Mengantisipasi Timbulnya Klaster Kerumunan COVID-19 dengan Pendekatan Program Dinamis" dapat terselesaikan dengan baik dan tidak ada hambatan sedikit pun. Terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, MT yang telah mengajar mata kuliah IF2211 Strategi Algoritma dan membimbing selama satu semester ini. Terima kasih terhadap segala pihak yang tidak dapat disebutkan satu persatu sehingga turut membantu terselesaikan pembuatan makalah ini.

## DAFTAR PUSTAKA

- [1] V. Efelina, S. Dampang, dan I. Maulana, "Penggunaan Drone untuk Penyemprotan Disinfektan dalam Pencegahan Covid-19 di Masa Pandemi (Studi Kasus di Desa Margasari)," Selaparang, vol. 4, no. April, hal. 368–373, 2021.
- [2] R. Munir, "Algoritma Greedy," no. Bagian 1, 2021, [Daring]. Tersedia pada:  
[http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Greedy-\(2018\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Greedy-(2018).pdf)
- [3] H. Sunandar dan Pristiwanto, "Optimalisasi Implementasi Algoritma Greedy dalam Fungsi Penukaran Mata Uang Rupiah," J. Tek. Inform. Unika St. Thomas, vol. 04, no. 02, hal. 193–201, 2019.
- [4] R. Munir, "Program Dinamis ( Dynamic Programming )," Progr. Stud. Tek. Inform. STEI-ITB, vol. 2021, hal. 1–57, 2021, [Daring]. Tersedia pada:  
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Program-Dinamis-2020-Bagian1.pdf>
- [5] N. Meghanathan, "Module 4 Dynamic Programming," hal. 6–11, 2015, [Daring]. Tersedia pada:  
<http://www.jsms.edu/nmeghanathan/files/2015/04/CSC323-Sp2015-Module-4-DynamicProgramming.pdf?x61976>.
- [6] L.M. Wastupranata, "Strategi Rute UAV untuk Pengantaran Obat-obatan bagi Penderita Covid-19 saat Isolasi Mandiri dengan Algoritma Cheapest Link & Sirkuit Hamilton," hal. 1–6, 2020, [Daring]. Tersedia pada:  
[https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Makalah/Makalah-Matdis-2020 \(208\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Makalah/Makalah-Matdis-2020 (208).pdf)

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Jakarta, 10 Mei 2021



Leonard Matheus Wastupranata - 13519215