

Optimalisasi Pencarian Rute Waktu-Terpendek ke Kantin dan Musholla di Kampus Ganesha ITB: Integrasi Algoritma Dijkstra dan *Binary Search*

Nyoman Ganadipa Narayana - 13522066¹

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13522066@std.stei.itb.ac.id

Abstrak— Makalah "Optimalisasi Pencarian Rute Waktu-Terpendek ke Kantin dan Musholla di Kampus Ganesha ITB" mengusulkan sebuah sistem berbasis integrasi algoritma Dijkstra dan *Binary Search* untuk membantu mahasiswa menemukan rute tercepat ke kantin dan musholla. Respons terhadap kebijakan kampus tanpa kendaraan, sistem ini meningkatkan efisiensi waktu dan mendukung keberlanjutan lingkungan di ITB. Studi ini membuktikan pentingnya pengetahuan algoritma dalam meningkatkan kualitas hidup mahasiswa di lingkungan kampus.

Kata kunci—Algoritma Dijkstra, *Binary Search*, Rute terpendek, Waktu-tempuh terpendek.

I. PENDAHULUAN

Institut Teknologi Bandung atau biasa disingkat menjadi ITB adalah salah satu perguruan tertinggi tertua di Indonesia, dan dikenal dengan prestasi akademik dan inovasinya. ITB berkonsep multikampus, termasuk di Jatinangor, Ganesha, dan Cirebon. Masing-masing kampus ini memiliki kekhasan dan kontribusi tersendiri dalam mendukung visi dan misi ITB sebagai pusat inovasi dan penelitian. ITB berkembang dengan menekankan bukan hanya pencapaian akademik, tetapi juga pada Pembangunan kampus berkelanjutan dan ramah lingkungan.

Kampus Ganesha ITB, dengan kebijakan larangan penggunaan kendaraan pribadi oleh mahasiswa di area kampus, memperlihatkan upaya serius dalam mengurangi jejak karbon dan meningkatkan keberlanjutan lingkungan. Kebijakan ini tidak hanya mencerminkan komitmen ITB terhadap kelestarian lingkungan, tetapi juga mendukung transformasi kampus menjadi ruang yang lebih hijau, sehat, dan kondusif untuk pejalan kaki. Dengan mengurangi kepadatan kendaraan, Kampus Ganesha menjadi contoh nyata dari upaya institusi pendidikan dalam mendorong gaya hidup ramah lingkungan dan berkelanjutan di kalangan generasi muda.

Kebijakan ini, meskipun bermanfaat bagi lingkungan, menghadirkan tantangan tersendiri bagi mahasiswa, terutama dalam mengatur waktu antara aktivitas akademik dan kegiatan lain seperti ibadah dan makan. Kegiatan akademik yang padat membuat mahasiswa harus seefisien mungkin menentukan tempat untuk beribadah dan tempat untuk membeli makanan. Untuk mengatasi tantangan ini, dibuatlah makalah ini yang memiliki tujuan sebagai solusi dari tantangan yang telah dideskripsikan: Optimalisasi Pencarian Rute Terdekat ke Kantin dan Musholla di Kampus Ganesha ITB.

II. LANDASAN TEORI

A. Graf

Dalam teori graf, graf adalah model atau representasi dari objek-objek diskrit yang menggunakan simpul (*vertex*) dan sisi (*edge*) sebagai penghubung antar simpul tersebut.

1. Graf Berarah dan Graf Tak Berarah

Berdasarkan orientasi arah pada sisi, graf dibedakan menjadi dua jenis utama: graf berarah dan graf tak berarah. Graf berarah, atau digraf, memiliki sisi yang menunjukkan arah dari satu simpul ke simpul lain, sering digambarkan dengan panah. Sementara itu, graf tak berarah tidak memiliki arah spesifik pada sisi-sisinya, sehingga hubungan antar simpulnya bersifat dua arah. Perbedaan ini penting karena menentukan bagaimana graf dapat digunakan dalam berbagai aplikasi, seperti dalam permodelan jaringan komunikasi atau jalur dalam sistem transportasi.

2. Graf Berbobot dan Graf Tak Berbobot

Graf berbobot dan graf tak berbobot adalah dua kategori lain dalam teori graf. Pada graf berbobot, setiap sisi memiliki nilai atau 'bobot' tertentu yang dapat mewakili jarak, biaya, atau parameter lainnya. Sebaliknya, graf tak berbobot tidak memiliki nilai tambahan pada sisinya, sehingga semua sisi dianggap memiliki bobot yang sama atau tidak relevan. Perbedaan ini sangat penting dalam aplikasi seperti pencarian rute terpendek, di mana bobot sisi mempengaruhi penentuan jalur yang dipilih.

3. *Adjacency lists*

Pemodelan bentuk graf dalam kode dapat menggunakan berbagai macam cara, salah satunya dengan menggunakan pemodelan dengan *adjacency lists*. Pemodelan ini menitikberatkan pada fakta bahwa setiap graf dapat dipandang dengan simpul-simpul yang memiliki sisi ke simpul lain. Dengan memandang graf seperti hal ini, Graf dapat dimodelkan sebagai *array of array of node* untuk graf tak berbobot atau *array of array of pair weight and node* untuk graf berbobot.

B. Algoritma Dijkstra

Algoritma Dijkstra adalah metode yang digunakan dalam pemrograman komputer untuk menemukan jalur terpendek antara satu titik dan semua titik lain dalam graf berbobot.

Metode ini ditemukan oleh ilmuwan komputer Belanda, Edsger Dijkstra, pada tahun 1956. Algoritma ini menggunakan prinsip bahwa jalur terpendek dari satu titik ke titik lainnya akan selalu lebih pendek atau sama dengan jalur yang melalui titik lain. Algoritma ini efektif digunakan dalam berbagai aplikasi, seperti dalam sistem GPS untuk menemukan rute terpendek atau dalam jaringan komputer untuk menentukan jalur terbaik. Namun, perlu dicatat bahwa algoritma ini menggunakan syarat bahwa tidak ada sisi yang berbobot negative di dalam graf.

C. Algoritma Binary Search

Algoritma Dijkstra adalah metode yang digunakan dalam pemrograman komputer untuk menemukan jalur terpendek antara satu titik dan semua titik lain dalam graf berbobot. Metode ini ditemukan oleh ilmuwan komputer Belanda, Edsger Dijkstra, pada tahun 1956. Algoritma ini menggunakan prinsip bahwa jalur terpendek dari satu titik ke titik lainnya akan selalu lebih pendek atau sama dengan jalur yang melalui titik lain. Algoritma ini efektif digunakan dalam berbagai aplikasi, seperti dalam sistem GPS untuk menemukan rute terpendek atau dalam jaringan komputer untuk menentukan jalur terbaik.

III. METODE

A. Representasi

Dalam upaya memecahkan masalah konkret pencarian rute waktu-terpendek di Kampus Ganesha ITB, makalah ini mengadopsi teori graf tak berarah dengan mengonversi lokasi penting seperti bangunan-bangunan yang sering dijumpai menjadi simpul dalam graf. Rute antar lokasi ini diwakili sebagai sisi graf, dengan integrasi Algoritma Dijkstra untuk menentukan jalur waktu-terpendek dan *Binary Search* untuk efisiensi waktu dalam pencarian. Pendekatan ini memungkinkan optimalisasi dalam menemukan rute tercepat berdasarkan waktu yang dibutuhkan.

B. Asumsi

Untuk menyederhanakan representasi konkret ke dalam terminologi graf lebih lanjut, dilakukan beberapa asumsi serta simplifikasi untuk mempermudah pengambilan dan pengolahan data. Beberapa asumsi yang dilakukan dinilai oleh penulis tidak akan berdampak besar dan tujuan dari asumsi ini murni untuk penyederhanaan tanpa mengurangi makna dari kondisi sesungguhnya secara signifikan. Beberapa asumsi tersebut ialah

1. Dalam konteks representasi bangunan sebagai simpul dalam teori graf, sebagian besar simpul diimplementasikan untuk mewakili suatu bangunan, tidak terbatas pada lokasi spesifiknya, seperti bagian timur, barat, atau selatan. Pendekatan ini memungkinkan representasi yang lebih universal dan tidak terikat pada orientasi geografis tertentu dari bangunan tersebut.
2. Jalur antara dua simpul yang berbeda jumlahnya terbatas, artinya mungkin saja memiliki jalur dari simpul A ke simpul B tetapi harus melewati beberapa simpul lain terlebih dahulu, padahal dalam kenyataannya, setidaknya di Kampus Ganesha ITB, selalu bisa memilih jalur dari bangunan A ke bangunan B tanpa melewati bangunan lain terlebih dahulu.
3. Bobot dari setiap sisi adalah waktu tempuh antara dua

simpul yang dikaitnya.

4. Kelajuan penulis selalu dianggap konstan saat pengukuran oleh penulis makalah untuk mengukur waktu yang dibutuhkan dari suatu simpul ke simpul lainnya.

C. Modifikasi Graf

Pada keadaan konkritnya, tidak semua bangunan dapat dikunjungi oleh mahasiswa, sehingga selain representasi dan asumsi yang telah dibuat dibutuhkan sedikit modifikasi lanjutan untuk merepresentasikan keadaan konkrit di Kampus Ganesha ITB dengan sebuah graf yang relatif sederhana.

Modifikasi yang dilakukan adalah dengan menganggap bahwa gedung tersebut antara hanya berupa tanah kosong (tidak ada kantin dan musholla) atau tidak dimasukkan ke dalam simpul pada graf yang dibuat.

D. Pengambilan Data

Dari ketiga bagian yang telah disebutkan: Representasi, Asumsi, serta Modifikasi Graf, maka akan dapat dibuat suatu graf yang berdasar pada keadaan konkrit bangunan-bangunan di Kampus Ganesha ITB secara relatif lebih sederhana.

Pengambilan data untuk panjang sisi-sisinya dilakukan dengan mengukur waktu-tempuh penulis untuk mencapai suatu lokasi dari lokasi lainnya menggunakan alat ukur stopwatch pada gawai penulis. Pengambilan data dilakukan hanya pada beberapa tempat dan lintasan yang relatif lebih sering digunakan. Sementara pengambilan data untuk mengetahui apakah suatu bangunan memiliki mushola, kantin, atau keduanya adalah dari arsip panitia tour AMI 2023 yang dapat dipercaya validitasnya.

IV. DATA DAN PENGOLAHAN

Peta Kampus Ganesha ITB yang semula dapat secara konkrit direpresentasikan dengan gambar 1, dapat dibuat graf yang



Gambar 1: Peta Kampus Ganesha ITB yang diambil dari potongan gambar pada <https://infografis.itb.ac.id/data-info-itb-2015-peta-kampus-ganesha-dan-jatinangor/peta-ganesha-jatinangor-p1/> yang diakses pada 9 Desember 2023.

mewakili bangunan-bangunan dan jalur-jalurnya tanpa mengganti kondisi aslinya.

Dengan itu, secara keseluruhan akan terdapat 53 simpul yang mewakili bangunan-bangunan tersebut yang secara detail terdapat pada rincian berikut.

No.	Nama Simpul	Representasi dari gedung
1	1	FTSP
2	2	Aula Barat
3	3	Teknik Sipil
4	4	Fisika
5	5	Ruang Serba Guna
6	6	FSRD
7	7	Aula Timur
8	8	LFM
9	9	Seni Murni, Desain
10	10	Labtek IX B
11	11	Labtek IX C
12	12	Labtek IX A
13	13	Teknik Lingkunga
14	14	Teknik Geodesi
15	15	GKUT
16	16	Labtek VII
17	17	Labtek VIII
18	18	Labtek V
19	19	Labtek VI
20	20	Lab Konversi
21	21	Lab Radar
22	22	GKUB
23	23	Labtek II
24	24	Labtek XI
25	25	Labtek X
26	26	Oktagon
27	27	TVST
28	28	Koordinator TPB
29	29	Gedung Kerjasama PLN – ITB
30	30	Labtek I
31	31	Kimia
32	32	Teknik Perminyakan
33	33	Labtek IV
34	34	Pusat Penelitian Energi
35	35	LAPI
36	36	Gedung Perpustakaan Pusat
37	37	Pusat Peneliti antar Universitas
38	38	Gedung Serba Guna
39	39	Labtek 3 (lantai 1)
40	40	Gedung SBM
41	41	CC Timur
42	42	Area parkir Sipil
43	43	Area parkir Seni Rupa
44	44	CC Barat bagian Timur
45	45	CIBE
46	46	Gedung TL bagian Barat
47	47	Labtek VII bagian Barat
48	48	Labtek VIII bagian Barat
49	49	Labtek V bagian Timur
50	50	Labtek VI bagian Timur
51	51	Labtek Biru
52	52	Labtek III (lantai 2)

53	53	CC Timur bagian Barat
----	----	-----------------------

Tabel 1: Tabel Simpul

Setelah membangun dan menamakan simpul-simpul yang dibutuhkan, ditentukan panjang sisi-sisi yang menghubungkan simpul-simpul tersebut, yaitu dengan mengukur waktu tempuh antara dua simpul yang terkoneksi. Dengan menggunakan asumsi yang telah dijelaskan sebelumnya, dicari 87 simpul yang dikatakan “cukup” untuk merepresentasikan keadaan konkrit ke dalam graf tak berarah. Hasil pengukuran oleh penulis didapatkan hasil panjang dari 87 sisi tersebut adalah sebagai berikut.

No.	Simpul 1	Simpul 2	Panjang sisi
1	1	2	81
2	1	6	102
3	1	41	148
4	2	3	75
5	2	5	140
6	2	7	120
7	2	42	82
8	3	4	84
9	3	42	102
10	4	5	55
11	4	18	70
12	4	20	52
13	5	18	90
14	5	44	33
15	6	7	72
16	6	43	155
17	7	8	22
18	7	9	56
19	8	9	25
20	9	10	40
21	9	12	65
22	10	11	71
23	11	12	63
24	11	13	90
25	12	46	40
26	13	14	72
27	13	15	95
28	13	46	50
29	14	15	45
30	15	16	123
31	15	17	107
32	15	30	131
33	15	31	114
34	16	17	31
35	16	29	62
36	16	47	71
37	17	41	81
38	17	46	59
39	17	48	63
40	18	19	35
41	18	20	50
42	18	22	84
43	18	49	70
44	19	22	85
45	19	24	80

46	19	50	70
47	20	21	62
48	20	22	62
49	21	22	50
50	22	23	100
51	22	24	122
52	22	42	210
53	22	45	122
54	23	24	85
55	23	51	95
56	23	39	105
57	24	51	30
58	25	51	30
59	26	27	62
60	26	38	80
61	26	51	74
62	26	28	48
63	27	29	48
64	27	51	74
65	28	29	42
66	28	30	68
67	29	30	68
68	30	31	72
69	30	32	128
70	31	32	68
71	32	33	121
72	33	34	71
73	34	35	41
74	34	36	148
75	36	37	105
76	37	38	50
77	38	40	97
78	38	52	108
79	39	40	102
80	41	46	65
81	41	53	32
82	42	45	163
83	44	53	42
84	44	49	63
85	47	48	32
86	47	50	32
87	49	50	35

Tabel 2: Tabel Sisi *Bidirectional*

Ingat juga bahwa setiap sisi pada tabel 2 merupakan sisi *bidirectional* karena graf yang dibuat merupakan graf tak berarah. Setelah mendefinisikan simpul-simpul dan mengukur Panjang masing-masing sisi, kita telah berhasil mengonstruksi graf yang kita inginkan sebagai representasi graf dari keadaan konkrit.

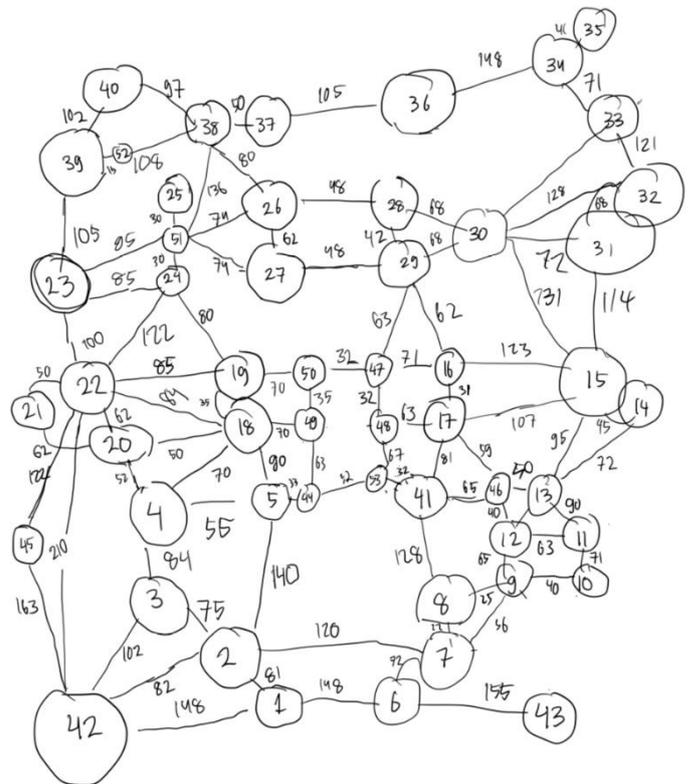
Selanjutnya, ditentukan simpul apa saja yang mewakili bangunan yang memiliki musholla, kantin, atau keduanya. Hasil penelusuran didapati data dengan rincian seperti pada tabel berikut.

No.	Keterangan	Simpul-simpul
1	Simpul memiliki musholla	3, 4, 5, 10, 11, 15, 16, 18, 22, 23, 24, 25, 30, 33, 36, 38, 41, 45, 52

2	Simpul memiliki kantin	12, 15, 18, 22, 31, 32, 39, 51
---	------------------------	--------------------------------

Tabel 3: Tabel Kumpulan simpul yang memiliki musholla atau kantin

Jika divisualisasikan, graf yang simpul-simpulnya telah didefinisikan pada tabel 1 dan sisi-sisinya telah diukur pada tabel 2, model graf tersebut akan tampak seperti pada gambar 2.



Gambar 2: Visualisasi graf dari definisi-definisinya yaitu tabel 1 dan tabel 2.

Setelah mendefinisikan graf yang dapat merepresentasikan kondisi konkrit peta Kampus Ganesha ITB dan mengumpulkan data simpul yang memiliki musholla ataupun kantin. Data yang dibutuhkan sudah cukup untuk dapat mengimplementasikan algoritma Dijkstra sebagai penentu rute terdekat dari simpul yang ditentukan ke simpul pemilik musholla terdekat dan simpul pemilik kantin terdekat. *pseudocode* dari algoritma Dijkstra yang diimplementasikan adalah sebagai berikut.

```

type Edge: array 0..1 of integer {0 menyatakan node
next dan 1 menyatakan bobotnya}

type AdjacencyList: array 0..173 of Edge

type Graph: array 0..52 of AdjacencyList

const musholla_nodes: array 0..18 of integer = ...
const canteen_nodes: array 0..7 of integer = ...

function is_in(target: integer, arr: array of
integer) -> boolean
{fungsi sudah terdefinisi}

```

```
function Dijkstra(g: Graph, start: integer) -> array
of 0..1 of integer {0 menyatakan musholla terdekat,
1 menyatakan kantin terdekat}
```

ALGORITMA

```
num_nodes <- lenth(g)
distances <- array 0..num_nodes-1 of integer

i traversal [0..num_nodes-1]
  distances[i] <- INF

distances[start] <- 0

priority_queue <- createPriorityQueue(0, start)

nearest_m <- -1
nearest_c <- -1
nearest_m_distance <- INF
nearest_c_distance <- INF

finish <- false
while (not isEmpty(priority_queue)
  and not finish) do
  elmt <- pop(priority_queue)
  current_distance <- elmt[0]
  current_node <- elmt[1]

  if (is_in(current_node, musholla_nodes)
    and
    current_distance < nearest_m_distance)
    then

      nearest_m <- current_node
      nearest_m_distance <- current_distance

  if (is_in(current_node, canteen_nodes)
    and
    current_distance < nearest_c_distance)
    then

      nearest_c <- current_node
      nearest_c_distance <- current_distance

  if (nearest_m /= -1 and nearest_c /= -1) then
    finish <- true
  else

    pointer <- g[0]
    count <- 0
    while (pointer != Nil) do
      nbor <- pointer[0]
      weight <- pointer[1]

      dst <- current_distance + weight
      if dst < distance[neighbor] then
        distance[neighbor] = dst
        push(priority_queue, dst, nbor)

      count <- count + 1
      pointer <- g[count]

  -> [nearest_m, nearest_c]
```

```
function is_in(target: integer, arr: array of
integer) -> boolean
```

KAMUS LOKAL

```
length: integer
low, high, mid : integer
found: boolean
```

ALGORITMA

```
length <- len(array)
low <- 0
high <- length - 1
mid <- (low + high) div 2

found <- false
while (low <= high and not found) do
  mid <- (low + high) div 2
  if (arr[mid] = target) then
    found <- true
  else if (arr[mid] < target) then
    low <- mid + 1
  else if (arr[mid] > target) then
    high <- mid - 1
-> found
```

Sekarang, program telah teroptimisasi dari yang sebelumnya *linear search* menjadi *binary search*. Mengetahui bahwa program telah selesai dibuat, dapat digunakanlah program yang dapat mengetahui lokasi musholla dan/atau kantin terdekat. Kemudian dilakukanlah sebagian kecil sample sebagai *testing* pada program dan berikut adalah tabel masukan, ekspektasi keluarannya, dan keluaran program.

No.	Simpul masukan	Eskpektasi keluaran	keluaran program
1	1	Musholla: 3	Musholla: 3
		Kantin: 12	Kantin: 12
2	2	Musholla: 3	Musholla: 3
		Kantin: 18	Kantin: 18
3	42	Musholla: 3	Musholla: 3
		Kantin: 22	Kantin: 22
4	33	Musholla: 33	Musholla: 33
		Kantin: 32	Kantin: 32

Tabel 4: Tabel hasil uji program pada sebagian kecil simpul.

Hasil dari *testing* sampel yang diuji menunjukkan bahwa ekspektasi dan keluaran program sepenuhnya sependapat dengan keadaan konkrit. Hasil dari program yang telah dibuat telah diimplementasikan menggunakan bahasa python, yang dapat diakses *source code*-nya melalui tautan di [sini](#), atau melalui tautan yang terletak pada bagian Lampiran.

V. KESIMPULAN

Hasil yang telah diuji untuk mencari *shortest time-based path* antara suatu simpul ke simpul terdekat yang memiliki musholla dan/atau kantin, didapat kesamaan antara keadaan konkrit dan keluaran program. Dengan fakta bahwa algoritma Dijkstra bekerja dengan baik untuk sisi-sisi yang nonnegatif seperti apa yang tertera pada landasan teori, algoritma Dijkstra bekerja dengan baik dalam mencari rute waktu-tempuh terpendek untuk

Menjalankan fungsi dijkstra di atas menggunakan fungsi *is_in* yang relatif *naive* (seperti *target in arr* pada python) akan menggunakan *linear search* karena bahasa-bahasa program tidak mengetahui karakteristik dari variable *arr* pada program. Padahal, *musholla_nodes* dan *canteen_nodes* adalah konstan yang dapat dibuat sedemikian sehingga menjadi konstan yang terurut. Jadi, untuk efisiensi program, digunakanlah algoritma *binary search* dan menjadikan variabel *musholla_nodes* dan *canteen_nodes* terurut. Sehingga, fungsi *is_in* yang diimplementasi adalah sebagai berikut.

mencari lokasi musholla dan/atau kantin terdekat.

VI. CATATAN KENDALA DAN EVALUASI

Dalam proses pengukuran untuk makalah ini, sering kali merasa jalur yang mungkin ditempuh terlalu banyak sehingga, di sisi lain, tenaga dan waktu pengukur terbatas, sehingga terkadang tidak konsisten kelanjutan berjalan penulis.

Selain itu, terdapat evaluasi yang dapat dijadikan pertimbangan dan koreksi dari proses penelitian ini, salah satunya adalah terkait pengambilan asumsi. Asumsi yang diambil seringkali tidak menggambarkan kondisi konkrit, salah satunya adalah sisi yang pada keadaan konkrit tetapi dengan alasan penyederhanaan dihilangkan. Evaluasi lainnya adalah perlu diperhatikan letak musholla apakah berada di bagian timur, barat, lantai 2, lantai 3, atau lainnya sehingga program dapat lebih presisi. Lebih lanjut, evaluasi untuk penulis kedepannya adalah sadari bahwa setiap data yang statis seperti simpul-simpul yang memiliki musholla dan/atau kantin selalu bisa dibuat search menggunakan struktur data hashtable yang menjadikan skema searching pada makalah ini menjadi $O(1)$ yang sebelumnya $O(\log n)$.

VII. PENUTUP

Pada akhir makalah ini, saya ingin menyampaikan rasa syukur saya kepada Tuhan Yang Maha Kuasa atas kelancaran dan keberhasilan penyelesaian makalah IF2120 Matematika Diskrit ini. Saya juga ingin mengungkapkan rasa terima kasih saya kepada Ibu Fariska Zakhralativa Ruskanda, S.T., M.T., yang bimbingan dan pengetahuannya sangat berharga dalam penerapan materi yang saya sajikan dalam makalah ini. Penghargaan juga saya tujukan kepada Bapak Dr. Ir. Rinaldi Munir, M.T. atas penyediaan materi pembelajaran yang inovatif selama kursus ini. Terima kasih tak terhingga juga saya ucapkan kepada keluarga, teman-teman, senior, dan seluruh anggota civitas akademika lainnya yang telah memberikan dukungan dan kontribusi dalam penyelesaian makalah ini. Saya berharap makalah ini memberikan manfaat nyata tidak hanya bagi mahasiswa ITB, tetapi juga bagi dosen dan tamu yang berkunjung ke Kampus Ganesha ITB.

REFERENSI

- [1] Laaksonen, Antti, "Competitive Programming Handbook." 2018, pp. 126, 32.
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2023-2024/19-Graf-Bagian1-2023.pdf>, diakses pada 9 Desember 17.49 WIB.
- [3] <https://infografis.itb.ac.id/data-info-itb-2015-peta-kampus-ganesha-dan-jatinangor/>, diakses pada 8 Desember 23.30 WIB.

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.

Bandung, 9 Desember 2023



Nyoman Ganadipa Narayana 13522066

KAMPUS INSTITUT TEKNOLOGI BANDUNG GANESHA

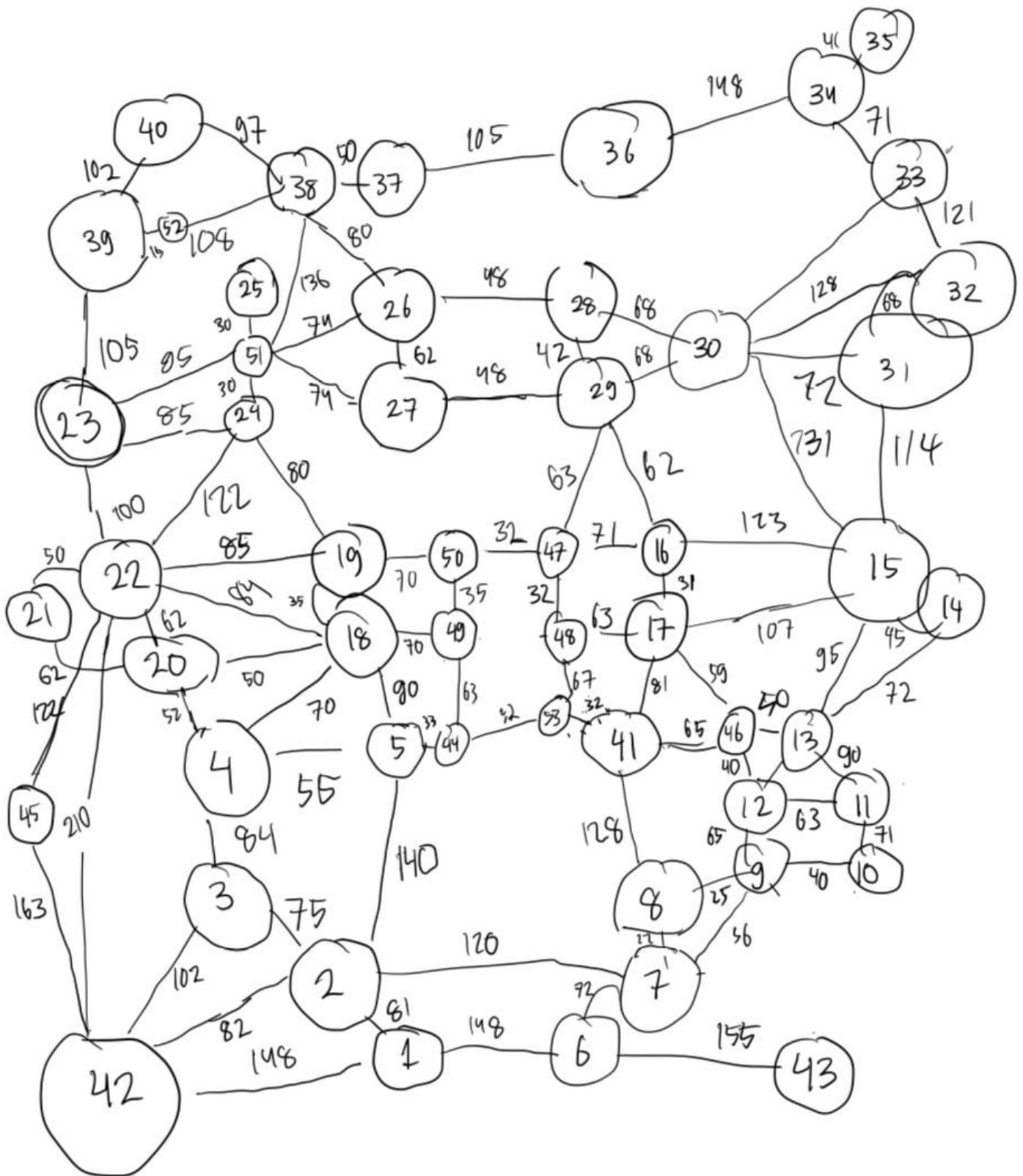
Keterangan

- 1 FTSP
- 2 Aula Barat
- 3 Teknik Sipil
- 4 Fisika
- 5 Ruang Serba Guna (RSG)
- 6 FSRD
- 7 Aula Timur
- 8 LFM/9009
- 9 Seni Murni, Desain
- 10 Labtek IX B - Teknik Arsitektur
- 11 Labtek IX C - Teknik Geodesi
Teknik Lingkungan
- 12 Labtek IX A - Teknik Planologi
- 13 Teknik Lingkungan
- 14 Teknik Geodesi
- 15 Gedung Kuliah Umum - Timur
- 16 Labtek VII - Farmasi, MKDU
- 17 Labtek VIII - FMIPA, Teknik Elektro,
UPT Bahasa
- 18 Labtek V - PIKSI, Teknik Informatika
- 19 Labtek VI - Teknik Fisika, Teknik Kelautan
Pusat Penelitian Kelautan
- 20 Lab Konversi
- 21 Lab Radar
- 22 Gedung Kuliah Umum - Barat
- 23 Labtek II - Teknik Mesin
Teknik Penerbangan
- 24 Labtek XI - Biologi, Geofisika, Meteorologi
- 25 Labtek X - Teknik Kimia, Teknik Mineral
- 26 Gedung Kuliah Umum - Oktagon
- 27 Gedung Kuliah Umum - TVST
- 28 Koordinator TPB
- 29 Gedung Kerjasama PLN - ITB
- 30 Labtek I - Teknik Geofisika, Lab Struktur
- 31 Kimia
- 32 Teknik Perminyakan
- 33 Labtek IV - FTM, Teknik Geologi
Teknik Pertambangan
- 34 Pusat Penelitian Energi
- 35 LAPI
- 36 Gedung Perpustakaan Pusat
- 37 Pusat Peneliti antar Universitas
- 38 Gedung Serba Guna (GSG)
- 39 Labtek III - FTI, Teknik Industri, Matematika
Astronomi, PPM
- 40 Pasca Sarjana, Pusat Peneliti Kepariwisataaan
- 41 Sarana Olahraga Ganesha (SARAGA)



LAMPIRAN

Model graf peta Kampus Ganesha ITB



LAMPIRAN

Tautan Kode Program

<https://github.com/ganadipa/Dijkstra-untuk-Kampus-Ganesha-ITB>