

Aplikasi Android Untuk Mencari Jalur Tercepat Pada Pengiriman Barang Dengan Algoritma Held-Karp

Alfin H^{1*}, Kartina Diah Kesuma Wardhani^{2**}

* Teknik Informatika, Politeknik Caltex Riau

alfin16ti@mahasiswa.pcr.ac.id¹, diah@pcr.ac.id²

Article Info

Article history:

Received 2020-07-01

Revised 2020-08-19

Accepted 2020-09-17

Keyword:

Pengiriman Barang,
Traveling Salesman Problem,
Algoritma Held-Karp.

ABSTRACT

The total number of shipments to be sent from 2017 with an average total of 18 million shipments per month until 2018 reaches 24 million per month, while the process the shipping path is still as described and guessing, so couriers often wrong in estimating the fastest route or forgetting to order delivery based on previous couriers which cause the courier to have to turn around and every time the courier turns back then the shipping process becomes ineffective in terms of time, distance and cost. Based on the results of a 2018 study, Held-Karp produced a better route than Iterative Deepening Search in finding a shipping route. Therefore an Android-based application is created and uses the Google Maps API to determine the time, distance, and cost required at each point of delivery, as well as using the Held-Karp algorithm to search for the fastest path based on time, distance and cost. Based on functionality testing the application is estimated to run correctly and per the expected function, then in cyclometric complexity testing, the Held-karp algorithm is categorized in well-written and structured code and high testability because there is no $v(G)$ above 10.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Seiring meningkatnya jumlah pengiriman barang yang harus dikirimkan dari tahun 2017 dengan total rata-rata 18 juta kiriman perbulan hingga tahun 2018 mencapai 24 juta perbulan[1], sedangkan proses penentuan jalur pengiriman barang masih ditentukan berdasarkan perkiraan dan menerkanerka jalur tercepat, sehingga kurir sering salah dalam memperkirakan jalur tercepat ataupun lupa urutan pengiriman berdasarkan perkiraan kurir sebelumnya yang menyebabkan kurir harus berputar balik dan setiap kali kurir berputar balik maka proses pengiriman menjadi tidak efektif dari segi waktu, jarak dan biaya.

Beberapa penelitian yang telah dilakukan diantaranya aplikasi pencarian jalur terpendek yang dirancang menggunakan Microsoft Visual Basic. Tujuan dari pembuatan aplikasi ini yaitu menerapkan algoritma A* dalam menyelesaikan permasalahan pencarian jalur teroptimal dalam mengirimkan barang dan membantu para supir atau kurir dalam menentukan jalur mana yang terpendek untuk mengantarkan barang. Proses penginputan data alamat pada aplikasi ini dilakukan secara manual dengan parameter asal,

tujuan dan jarak yang ditempuh [2].penelitian selanjutnya yaitu aplikasi pencarian jalur optimal berdasarkan jarak, waktu dan biaya. Proses penginputan data alamat pada aplikasi ini dilakukan secara manual pada *console* dengan parameter sesuai dengan satuan yang digunakan jarak/waktu/biaya [3]. Penelitian lainnya yaitu aplikasi pencarian jalur dengan 2 algoritma yang bertujuan untuk membandingkan kedua algoritma agar dapat diketahui kelebihan dan kekurangan dari tiap metode. Proses memasukkan data alamat pada aplikasi ini dilakukan secara manual dengan data yang diambil dari *google map*[6].

Pada penelitian ini akan menggunakan salah satu solusi Traveling Salesman Problem (TSP) dengan algoritma Held-Karp untuk mempercepat proses pengiriman. TSP adalah sebuah permasalahan dimana seorang salesman harus mengunjungi setiap kota dimana untuk setiap kota dikunjungi tepat sekali, kemudian dia harus mulai dan kembali ke kota asal [3]. Salesman dianalogikan sebagai kurir dan kota yang dikunjungi merupakan alamat barang yang di kirikan. Kurir akan mengirimkan barang dari kantor ke setiap alamat yang

hanya dikunjungi sekali dan kurir akan kembali ke kantor setelah pengiriman barang selesai dilakukan.

Algoritma Held-Karp atau yang sering disebut dengan algoritma Bellman Held-Karp [2] merupakan suatu algoritma Dynamic Programming yang diperkenalkan secara independen oleh Bellman, Held dan Karp pada tahun 1962 [4]. Algoritma Held-Karp umumnya digunakan untuk mengatasi permasalahan pencarian jalur terdekat, contohnya pada Traveling Salesman Problem (TSP) atau pada perjalanan wisata. Algoritma Held-Karp memiliki kompleksitas $O(2^n n^2)$ [3] dan dari hasil perbandingan *Iterative Deepening Search* dan *Held-Karp*, Held-Karp menghasilkan rute yang lebih baik dibandingkan dengan metode *Iterative Deepening Search* [6].

Dengan aplikasi ini pengguna dapat memasukkan alamat-alamat tujuan pengiriman yang kemudian akan dilakukan pengumpulan informasi peralamat dari Google Maps API sesuai dengan satuan yang digunakan. Lalu dengan adanya algoritma Held-Karp pada aplikasi ini, maka aplikasi akan menghasilkan jalur pengiriman tercepat dari segi waktu, jarak dan biaya. Harapan dari penelitian ini adalah mempercepat pengiriman barang dan memudahkan kurir dalam menentukan jalur pengiriman barang.

II. METODE PENELITIAN

Metode penelitian dimulai dari studi literatur, pemilihan algoritma dari hasil studi literatur, perancangan aplikasi sesuai hasil identifikasi permasalahan, dan pengujian hasil implementasi kepada pengguna. Berikut pengujian pada penelitian ini adalah sebagai berikut.

A. Pengujian Fungsionalitas

Pengujian *blackbox* yang digunakan pada sistem ini memfokuskan fungsionalitas dari aplikasi yang dibangun. Pengujian ini akan menguji penggunaan dari aplikasi, sehingga didapatkan apakah fungsi dari aplikasi tersebut telah berjalan dengan benar dan fungsi aplikasi telah sesuai dengan rancangan.

TABEL I
PENGUJIAN FUNGSIONALITAS

No	Pengujian	Penjelasan	Hasil yang diharapkan
1	Registrasi	Kurir melakukan pendaftaran akun.	Aplikasi akan mendaftarkan akun kurir.
2	Menambah alamat pengiriman	Kurir menambah alamat pengiriman	Aplikasi akan memasukan alamat pengiriman ke daftar alamat.
3	Menghapus alamat pengiriman	Kurir menghapus alamat pengiriman	Aplikasi akan menghapus alamat pengiriman dari daftar alamat.
4	Mencari Jalur Pengiriman	Kurir mencari jalur pengiriman tercepat.	Aplikasi akan menampilkan jalur pengiriman berdasarkan

No	Pengujian	Penjelasan	Hasil yang diharapkan
			daftar alamat dan satuan yang digunakan.
5	Melihat jalur pengiriman	Kurir melihat jalur pengiriman.	Aplikasi menampilkan jalur pengiriman dalam bentuk <i>Google Maps</i> .
6	Melihat riwayat jalur pengiriman	Admin / Kurir melihat daftar jalur pengiriman yang pernah dilakukan.	Aplikasi akan menampilkan daftar jalur pengiriman yang pernah dilakukan oleh kurir.
7	Menghapus akun Kurir	Admin menghapus akun kurir.	Aplikasi menghapus data kurir pada <i>database</i> .

B. Cyclomatic Complexity

Pengujian *whitebox* yang digunakan pada sistem ini memfokuskan struktur kontrol dari sebuah aplikasi yang dibangun secara prosedural. Pengujian ini akan menguji setiap baris coding dari algoritma *Held-Karp* apakah algoritma merupakan algoritma yang sederhana atau kompleks berdasarkan logika yang diterapkan.

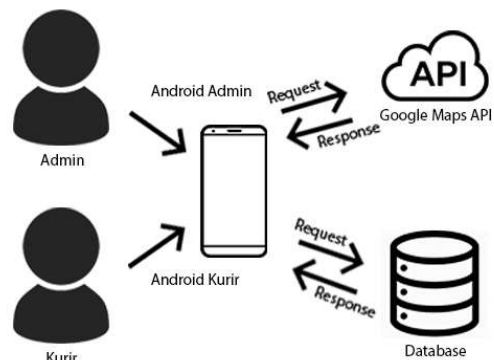
TABEL III
NILAI CYCLOMATIC COMPLEXITY V(G)

Complexity	Keterangan
1-10	Kode ditulis dengan baik dan terstruktur <i>High Testability</i> Biaya dan upaya rendah
11-20	Kode kompleks <i>Medium Testability</i> Biaya dan upaya sedang
21-40	Kode yang sangat kompleks <i>Low Testability</i> Biaya dan upaya tinggi
>41	Sama sekali tidak bisa diuji Biaya dan usaha yang sangat tinggi

III. ANALISIS DAN PERANCANGAN

A. Arsitektur Sistem

Arsitektur sistem di dalam rancang bangun aplikasi android untuk mencari jalur pengiriman tercepat adalah sebagai berikut.

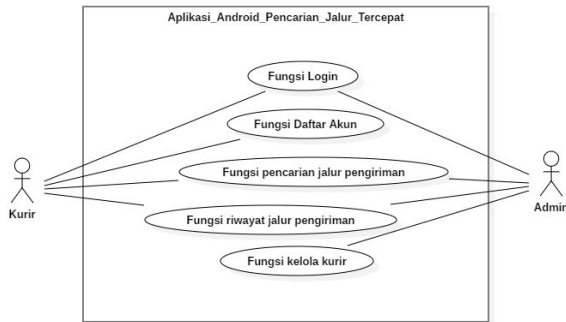


Gambar 1. Arsitektur Sistem

Pengguna dari aplikasi ini adalah Admin dan Kurir, aplikasi ini dibangun dengan menggunakan *Google Maps API* untuk mendapatkan data dan menggunakan *Database* untuk menyimpan data hasil pencarian jalur.

B. Use Case Diagram

Use Case Diagram berdasarkan kebutuhan sistem sesuai dengan aktornya adalah sebagai berikut:



Gambar 2. Use Case Diagram Kurir dan Admin

Gambar 2 menjelaskan bahwa pada aplikasi ini, pengguna dapat melakukan beberapa aktivitas seperti login, mendaftar akun, mencari jalur pengiriman, melihat jalur pengiriman dan melihat riwayat jalur pengiriman. Sedangkan aktivitas yang dapat dilakukan admin, yaitu login, melihat riwayat jalur pengiriman dan kelola daftar kurir.

C. Perhitungan algoritma dengan satuan waktu

Berikut adalah contoh perhitungan algoritma *Held-Karp* untuk mencari jalur pengiriman tercepat pada pengiriman dengan satuan waktu:

TABEL IIIII
DAFTAR ALAMAT DENGAN SATUAN WAKTU

(x,y)	1. JNE Warehouse	2.Elementary School 36	3. Toko buku gramedia	4. Jl. Lokomotif
1.JNE Warehouse	0	1153 detik	628 detik	1317 detik
2.Elementary School 36	974 detik	0	383 detik	363 detik
3.Toko buku gramedia	668 detik	490 detik	0	762 detik
4.Jl. Lokomotif	1322 detik	363 detik	731 detik	0

k = 0, null set:

Set \emptyset :

$$g(2, \emptyset) = c_{21} = 974$$

$$g(3, \emptyset) = c_{31} = 668$$

$$g(4, \emptyset) = c_{41} = 1322$$

k = 1, pertimbangkan set dari 1 elemen:

Set {2}:

$$g(3, \{2\}) = c_{32} + g(2, \emptyset) = c_{32} + c_{21} = 490 + 974 = 1464 \quad p(3, \{2\}) = 2$$

$$g(4, \{2\}) = c_{42} + g(2, \emptyset) = c_{42} + c_{21} = 363 + 974 = 1337 \quad p(4, \{2\}) = 2$$

Set {3}:

$$g(2, \{3\}) = c_{23} + g(3, \emptyset) = c_{23} + c_{31} = 383 + 668 = 1051 \quad p(2, \{3\}) = 3$$

$$g(4, \{3\}) = c_{43} + g(3, \emptyset) = c_{43} + c_{31} = 731 + 668 = 1399 \quad p(4, \{3\}) = 3$$

Set {4}:

$$g(2, \{4\}) = c_{24} + g(4, \emptyset) = c_{24} + c_{41} = 363 + 1322 = 1685 \quad p(2, \{4\}) = 4$$

$$g(3, \{4\}) = c_{34} + g(4, \emptyset) = c_{34} + c_{41} = 762 + 1322 = 2084 \quad p(3, \{4\}) = 4$$

k = 2, pertimbangkan set dari 2 elemen:

Set {2,3}:

$$g(4, \{2,3\}) = \min\{c_{42}+g(2, \{3\}), c_{43}+g(3, \{2\})\} \quad p(4, \{2,3\}) = 2$$

$$= \min\{363+1051, 731+1464\}$$

$$= \min\{1414, 2195\} = 1414$$

Set {2,4}:

$$g(3, \{2,4\}) = \min\{c_{32}+g(2, \{4\}), c_{34}+g(4, \{2\})\} \quad p(3, \{2,4\}) = 4$$

$$= \min\{490+1685, 762+1337\}$$

$$= \min\{2175, 2099\} = 2099$$

Set {3,4}:

$$g(2, \{3,4\}) = \min\{c_{23}+g(3, \{4\}), c_{24}+g(4, \{3\})\} \quad p(2, \{3,4\}) = 4$$

$$= \min\{383+2084, 363+1399\}$$

$$= \min\{2467, 1762\} = 1762$$

Panjang perjalanan yang optimal:

$$f = g(1, \{2,3,4\})$$

$$= \min\{c_{12} + g(2, \{3,4\}), c_{13} + g(3, \{2,4\}), c_{14} + g(4, \{2,3\})\}$$

$$= \min\{1153 + 1762, 628 + 2099, 1317 + 1414\}$$

$$= \min\{2915, 2727, 2731\} = 2727$$

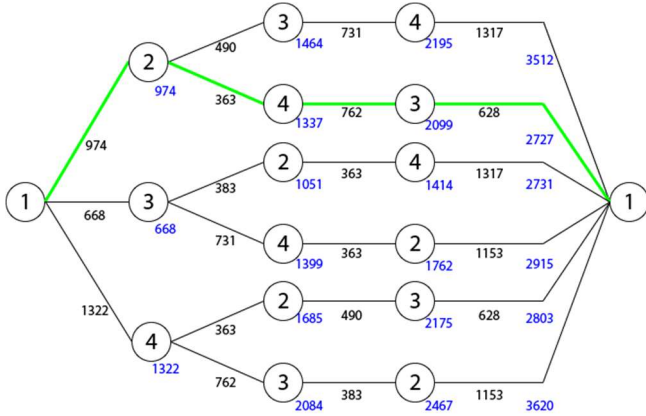
Penerus node 1: $p(1, \{2,3,4\}) = 3$

Penerus node 3: $p(3, \{2,4\}) = 4$

Penerus node 2: $p(4, \{2\}) = 2$

Rute yang optimal: $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1$

1. Kantor JNE \rightarrow 3. Toko Buku Gramedia-Pekanbaru \rightarrow 4. Jl Lokomotif \rightarrow 2.Elementary School 36 Pekanbaru \rightarrow 1. Kantor JNE



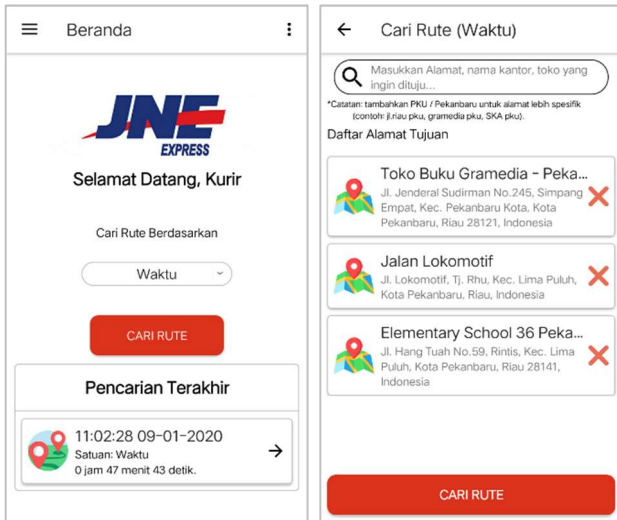
Gambar 3. Graf perhitungan dengan satuan waktu

Gambar 3 merupakan hasil pemetaan graf dari perhitungan satuan waktu dengan rute optimal 1-2-4-3-1 dan hasil 2727.

IV. HASIL DAN PEMBAHASAN

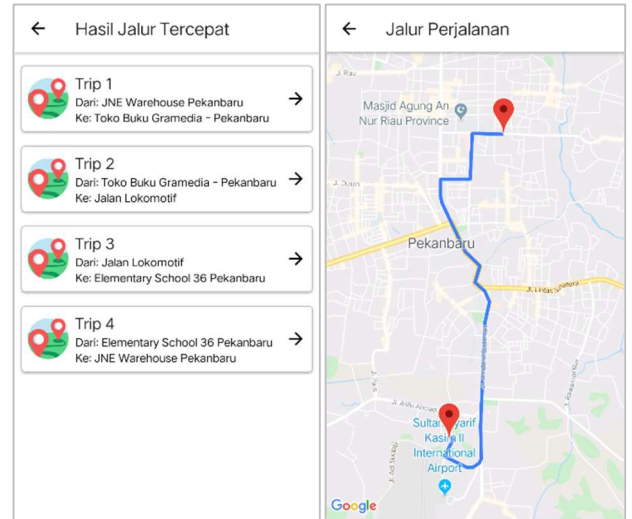
A. Hasil Perancangan

Berikut adalah beberapa halaman yang menjadi poin utama dalam Aplikasi Android Untuk Mencari Jalur Tercepat Pada Pengiriman Barang Dengan Algoritma Held-Karp. Pada sisi kurir terdapat halaman beranda, halaman untuk mencari jalur pengiriman, halaman hasil jalur tercepat dan tampilan map. Kemudian untuk admin terdapat halaman kelola daftar kurir.



Gambar 4. Tampilan halaman beranda (kurir) dan halaman input alamat

Gambar 4 merupakan halaman beranda untuk sisi kurir, pada halaman tersebut kurir dapat memilih satuan untuk jalur pengiriman tercepat berdasarkan waktu, jarak dan biaya. Tampilan halaman input alamat merupakan tampilan untuk menginputkan alamat tujuan pengiriman. Proses penginputan alamat langsung terintegrasi dengan Google Maps API, sehingga pengguna akan auto-correction atau rekomendasi untuk alamat yang diinputkan.



Gambar 5. Tampilan hasil jalur tercepat dan halaman jalur pengiriman.

Gambar 5 merupakan halaman hasil jalur pengiriman tercepat dari inputan halaman sebelumnya. Halaman jalur pengiriman menampilkan jalur pengiriman titik awal dan titik tujuan dengan Google Maps API.



Gambar 6. Tampilan halaman kelola daftar kurir

Gambar 6 merupakan halaman kelola kurir yang hanya dapat di akses oleh admin. Pada halaman ini admin dapat menghapus kurir yang sudah tidak aktif. Ketika admin menghapus akun kurir tertentu, maka akun tersebut tidak dapat login ataupun menggunakan fitur di aplikasi.

B. Hasil Pengujian dan Analisis

Aplikasi yang telah dibangun diujikan ke 5 pengguna selama 1 minggu, dan di dapatkan hasil pengujian fungsionalitas dengan data selama 1 minggu. Berikut data hasil pengujian selama 1 minggu:

TABEL IVV
DATA PEGUJIAN PENGGUNAAN APLIKASI

No	Tanggal	Kurir	Jumlah Alamat	Satuan	Hasil
1	07/01/2020	andi	4	biaya	Rp4.731
2	07/01/2020	latief	4	waktu	1 jam 4 menit 40 detik
3	07/01/2020	syahril	3	waktu	50 menit 35 detik
4	07/01/2020	andi	4	waktu	54 menit 56 detik
5	08/01/2020	syahril	6	biaya	Rp3.549
6	08/01/2020	andi	8	waktu	1 jam 6 menit 56 detik
7	08/01/2020	rolyz	9	jarak	22 Km
8	08/01/2020	syahril	6	waktu	46 menit 42 detik
9	09/01/2020	rolyz	9	biaya	Rp3.984
10	09/01/2020	syahril	7	biaya	Rp4.711
11	09/01/2020	latief	5	waktu	1 jam 4 menit 52 detik
12	09/01/2020	rolyz	9	waktu	47 menit 43 detik
13	09/01/2020	syahril	7	waktu	57 menit 27 detik
14	10/01/2020	ade	5	biaya	Rp3.576
15	10/01/2020	rolyz	9	biaya	Rp3.644
16	10/01/2020	rolyz	10	biaya	Rp3.666
17	10/01/2020	andi	5	biaya	Rp4.525
18	10/01/2020	rolyz	9	jarak	20 Km
19	10/01/2020	ade	5	jarak	20 Km
20	10/01/2020	rolyz	4	jarak	21 Km
21	10/01/2020	ade	5	waktu	45 menit 18 detik
22	10/01/2020	andi	5	waktu	51 menit 49 detik
23	11/01/2020	rolyz	4	biaya	Rp3.928
24	11/01/2020	latief	5	waktu	1 jam 5 menit 41 detik
25	11/01/2020	rolyz	4	jarak	22 Km
26	11/01/2020	latief	5	jarak	33 Km
27	11/01/2020	ade	5	waktu	44 menit 53 detik
28	11/01/2020	andi	5	waktu	48 menit 24 detik
29	12/01/2020	rolyz	3	biaya	Rp3.786
30	12/01/2020	latief	5	biaya	Rp8.409
31	12/01/2020	latief	5	waktu	1 jam 32 menit 35 detik
32	12/01/2020	rolyz	3	jarak	21 Km

33	12/01/2020	ade	4	jarak	24 Km
34	12/01/2020	rolyz	3	waktu	49 menit 14 detik
35	12/01/2020	andi	4	waktu	49 menit 23 detik
36	13/01/2020	rolyz	5	waktu	1 jam 4 menit 57 detik
37	13/01/2020	latief	3	waktu	1 jam 9 menit 47 detik
38	13/01/2020	syahril	8	jarak	19 Km
39	13/01/2020	rolyz	5	jarak	30 Km
40	13/01/2020	syahril	8	waktu	46 menit 6 detik
41	16/01/2020	rolyz	6	waktu	39 menit 18 detik
42	17/01/2020	rolyz	10	biaya	Rp4.760
43	17/01/2020	rolyz	10	waktu	33 menit 51 detik

Pada pengujian fungsionalitas ini terdapat 7 aktivitas yang diuji dengan data yang benar maupun salah. Berdasarkan pengujian fungsionalitas ini, didapatkan 7 dari 7 aktivitas yang diuji telah berjalan dengan semestinya.

TABEL V
PENGUJIAN CYCLOMATIC COMPLEXITY

No	Method	v(G)
1	.TravelingSalesmanHeldKarp.tourRoute(int[][])	10
2	.TravelingSalesmanHeldKarp.SetSizeComparator.c ompare(Set<Integer>,Set<Integer>)	1
3	.TravelingSalesmanHeldKarp.Index.hashCode()	2
4	.TravelingSalesmanHeldKarp.Index.equals(Object)	6
5	.TravelingSalesmanHeldKarp.Index.createIndex(int ,Set<Integer>)	1
6	.TravelingSalesmanHeldKarp.getTour(Map<Index, Integer>,int)	5
7	.TravelingSalesmanHeldKarp.getCost(Set<Integer> ,int,Map<Index, Integer>)	1
8	.TravelingSalesmanHeldKarp.generateCombination (int[],int,int,List<Set<Integer>>,int[])	3
9	.TravelingSalesmanHeldKarp.generateCombination (int)	2
10	.TravelingSalesmanHeldKarp.createSet(int[],int)	3

Berdasarkan hasil pengujian *cyclometric complexity* pada tabel III. Pada setiap *method* dalam algoritma *held-karp* yang ditulis tidak terdapat yang memiliki $V(G)$ lebih dari 10 maka algoritma *held-karp* pada aplikasi android untuk mencari jalur tercepat dapat dikategorikan dalam kode yang ditulis dengan baik dan terstruktur serta *high testability*.

V. KESIMPULAN

Berdasarkan penelitian yang telah dilaksanakan, didapatkan hasil dan kesimpulan bahwa aplikasi yang dibangun berjalan dengan benar dan sesuai dengan fungsi

yang diharapkan setelah dilakukan pengujian fungsionalitas, dan berdasarkan hasil pengujian *cyclomatic complexity*, aplikasi yang dibangun dikategorikan kode yang ditulis dengan baik dan terstruktur serta *high testability* karena tidak ada *method* dengan nilai $v(G)$ yang lebih dari 10.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak yang telah banyak memberikan bantuan dan dukungan yang tiada terhingga baik secara langsung maupun tidak langsung kepada Tuhan YME, Politeknik Caltex Riau dan Politeknik Negeri Batam

DAFTAR PUSTAKA

- [1] A. A. N. Hidayat, "2019, JNE Targetkan Pengiriman 1 Juta Paket Tiap Hari," 9 Desember 2018. [Online]. Available: <https://bisnis.tempo.co/read/1153712/2019-jne-targetkan-pengiriman-1-juta-paket-tiap-hari/full&view=ok>.
- [2] W. Alfarisi, "PENCARIAN JALUR TERPENDEK PENGIRIMA N BARANG," *Jurnal Riset Komputer (JURIKOM)*, pp. 90-95, 2016.
- [3] I. G. S. Rahayuda and N. P. Linda Santiari, "Penerapan Pemrograman Dinamis Pada Manajemen Pengiriman Produk Menggunakan Metode Held-Karp," *E-Proceedings KNS&ISTIKOM Bali*, pp. 513-518, 2017.
- [4] A. R. Amin, M. Ikhsan and L. Wibisono, *Traveling Salesman Problem*, Bandung: Institut Teknologi Bandung, 2012.
- [5] C. Chekuri, "Approximating the Held-Karp Bound for Metric TSP in Nearly-Linear Time," *58th Annual IEEE Symposium on Foundations of Computer Science*, pp. 789-800, 2017.
- [6] I. G. S. Rahayuda and N. P. Linda Santiari, "Implementasi dan Perbandingan Metode Iterative Deepening Search dan Held-Karp pada Manajemen Pengiriman Produk," *Open Access Journal Of Information System (OAJIS)*, pp. 131-149, 2018.
- [7] Megariza, "Penentuan Rute Belanja dengan TSP dan Algoritma Greedy," *Makalah IF3051 Strategi Algoritma*, 2011.
- [8] R. Munir, *Matematika Diskrit, Revisi Kelima.*, Bandung: Informatika., 2012.
- [9] I. ECMA, "'The JSON Data Interchange Format.'," *ECMA International 2st Edition(December):16.*, 2017.
- [10] "softwaretestingfundamentals," 2019. [Online]. Available: <http://softwaretestingfundamentals.com/black-box-testing/>.
- [11] R. M. Patelia and S. Vyas, "A Review and Analysis on Cyclomatic Complexity," *ORIENTAL JOURNAL OF COMPUTER SCIENCE & TECHNOLOGY*, pp. 382-384, 2014.