DESAIN JARINGAN INTERNET *INDIHOME* DI PERUMAHAN MANGGAR PERMAI AMBULU MENGGUNAKAN ALGORITMA *K-MEANS* DAN KRUSKAL

(*Indihome Internet Network Design in The Manggar Permai Ambulu Housing Estate Using The K-Means and Kruskal Algorithm*)

Reggy Valentinnes Septa Jeniusa[1)], Kiswara Agung Santoso[2)], Kusbudiono[3)]

[1,2,3)]Jurusan Matematika, Fakultas MIPA, Universitas Jember,
Jl. Kalimantan 37, Jember, 68121, Indonesia
e-mail: reggyvalentines@gmail.com, kiswara.fmipa@unej.ac.id, kusbudiono@unej.ac.id

Abstract. Based on the survey results of the Association of Indonesian Internet Service Providers, internet users are increasing reaching 196.7 million people. This community need is the driving force for PT. Telkom to overcome the need for internet access, namely the presence of Indihome. The Indihome internet network is distributed to several houses through ODP. The grouping of houses on ODP is based on the closest distance using the k-means algorithm. Meanwhile, minimizing the use of Indihome internet cables using the Kruskal algorithm. The data used in this study are house coordinates data obtained from the Google Earth application. The data is processed using the k-means algorithm, then continued using the Kruskal algorithm. The results of the research are clustering in houses with ODP formed 16 clusters which are processed using the k-means algorithm and stop at the 5th iteration. The ODP used has capacities of 8 and 12. ODP 1, 2, 4, 5, 8, 9, 11, 12, 13, 14, 15, 16 have capacities of 8. While ODP 3, 6, 7, 10 has a capacity of 12. Fiber optic cable between ODP is processed using the Kruskal algorithm and produces a minimum weight of 0.006601782 or ± 735 meters.

Keywords: Indihome, K-Means Algorithm, Kruskal Algorithm

1. Pendahuluan

Internet adalah singkatan dari *Interconnected Networking* atau dapat diartikan sebagai rangkaian komputer yang terhubung di dalam beberapa rangkaian jaringan. Melalui internet semua orang dapat bertukar informasi tanpa harus bertatap muka, hal ini sangat mempermudah pekerjaan seseorang. Hasil survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) periode 2019 hingga kuartal II 2020 mencatat jumlah pengguna internet mencapai 196,7 juta jiwa. Jumlah ini meningkat 23,5 juta atau 8,9% dibandingkan pada tahun 2018. Kebutuhan masyarakat dalam komunikasi dan informasi menjadi pendorong bagi perusahaan jasa penyedia layanan informasi dan telekomunikasi. Salah satu penyedia jasa telekomunikasi yaitu PT. Telekomunikasi Indonesia (Telkom). Telkom mengatasi pemenuhan kebutuhan akses internet yaitu dengan salah satu produk *Indonesia Digital Home* (*Indihome*). Jaringan internet *Indihome* terdistribusi ke beberapa rumah melalui ODP. Kapasitas ODP dapat mencakup 8, 12, 16, 24, dan 48 rumah. Pengelompokan rumah pada ODP didasarkan pada jarak terdekat antara rumah dengan tiang ODP. Pengelompokan rumah ini dapat dilakukan menggunakan algoritma *k-means.*

Sedangkan meminimalkan penggunaan kabel internet *Indihome* antar ODP dapat menggunakan algoritma kruskal.

Bastian [1] menggunakan algoritma *K-Means* untuk *clustering* pada penyakit menular manusia. Hasilnya *cluster* 1 sampai 4 didominasi oleh penyakit ISPA dan diare, sedangkan *cluster* 5 dan 6 didominasi oleh penyakit diare dan tuberkulosis. Priyono [5] menggunakan algoritma kruskal untuk menentukan lintasan terpendek efektif *call salesman.* Hasil yang didapatkan yaitu selisih jarak tempuh lintasan *salesman* sebelum dan sesudah menggunakan algoritma kruskal sebesar 2.560 meter.

Graf

Budayasa [2] menjelaskan bahwa sebuah graf G terdiri dari dua himpunan yaitu himpunan berhingga tak kosong $V(G)$ dan himpunan berhingga (mungkin kosong) $E(G)$. Himpunan $V(G)$ disebut himpunan titik yang digambarkan dengan sebuah noktah dan himpunan $E(G)$ disebut himpunan sisi yang digambarkan dengan ruas garis. Setiap elemen dalam $E(G)$ merupakan pasangan tak berurutan dari himpunan $V(G)$. Titik-titik tersebut akan dihubungkan melalui kurva sederhana (ruas garis).

Algoritma *K-Means*

Hung [4] menjelaskan bahwa algoritma *k-means* merupakan pengelompokan data ke dalam *cluster* yang sudah ditentukan sebelumnya. Data yang digunakan berupa jarak dari tiap objek ke pusat *cluster* (*centroid*). Data ini dikelompokkan ke pusat *cluster* yang terdekat. Proses pada algoritma *K-means* butuh berulang-ulang untuk mendapatkan basis data *cluster*. Jumlah *cluster* awal sangat dibutuhkan sebagai masukan dan akan menghasilkan titik *centroid* akhir sebagai output. Pemilihan pola sebagai titik awal *centroid* dilakukan secara acak atau random. Selanjutnya akan dilakukan iterasi sebagai cara untuk mendapatkan pola titik akhir *centroid.*

Menurut Ediyanto [3] algoritma *K-Means* adalah sebagai berikut:

1. Menghitung jumlah data.
2. Menentukan jumlah *k-cluster.*
3. Menentukan *k-centroid* (titik pusat *cluster*) awal secara random.
4. Menghitung jarak setiap data ke masing-masing *centroid* menggunakan *Euclidian Distance*

$$d(x,y) = \sum_{i=1}^{n} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\ ; i,j = 1,2,\dots,n \qquad (2.1)$$

dengan x_i : data x ke i

x_j : data x ke j

y_i : data y ke i

y_j : data x ke j

n : banyaknya data

5. Menentukan jarak terpendek dari setiap data ke masing-masing *centroid.*
6. Menghitung rata-rata dari data pada setiap *cluster*, kemudian akan mendapatkan posisi *centroid* baru.
7. Langkah 3-6 diulangi sebanyak n kali, jika posisi *centroid* baru tidak sama.

Algoritma Kruskal

Menurut Tambunaan [6] algoritma kruskal adalah salah satu algoritma yang digunakan untuk mendapatkan *minimum spanning tree* (MST) dengan bobot terkecil. Pada algoritma kruskal, sisi-sisi dalam graf diurutkan terlebih dahulu berdasarkan bobotnya dari kecil ke besar. Sisi yang dimasukkan kedalam himpunan V adalah sisi graf G sedemikian sehingga V adalah sebuah simpul. Pada keadaan awal, garis-garis sudah diurutkan berdasarkan bobot membentuk graf. Graf tersebut dinamakan graf merentang. Sisi dari graf G ditambahkan ke V jika sisi tersebut tidak membentuk sikel. Algoritmanya yaitu:

1. Himpunan sisi dari graf G diurutkan dari bobot terkecil ke terbesar.
2. Membuat graf T dengan memasukkan 1 sisi terpendek dari graf G.
3. Diulangi (sebanyak sisi).
   a. Mengambil sisi selanjutnya dari graf G.
   b. Jika sisi itu tidak membuat sikel di graf T.
      1) Memasukkan sisi itu ke graf T
      2) Memasukkan simpul-simpul sisi itu ke graf T.

2. Metodologi

Data yang digunakan pada penelitian ini merupakan data koordinat rumah. Data ini didapatkan dari aplikasi *google earth.* Data tersebut diproses menggunakan algoritma *k-means,* kemudian dilanjutkan menggunakan algoritma kruskal.

Algoritma *k-means* dapat diuraikan sebagai berikut:

1. Menghitung jumlah data
   Data yang digunakan yaitu 111 koordinat rumah. Koordinat rumah digunakan sebagai data awal. Adapun data tersebut dapat dilihat pada Tabel 1.
2. Menentukan jumlah *k-cluster*
   Berdasarkan data koordinat 111 rumah, dibentuk *cluster* sebanyak 16. Hal ini didasarkan pada jumlah minimum kapasitas ODP yaitu 8.

Tabel 1. Data koordinat rumah

Rumah	Koordinat (°S)	Koordinat (°E)
1	8,344138	113,597260
2	8,343980	113,597273
3	8,343803	113,597256
4	8,343673	113,597246
5	8,343569	113,597269
6	8,343507	113,597231
7	8,343420	113,597226
8	8,343301	113,597161
9	8,343184	113,597137
10	8,343050	113,597128
11	8,342949	113,597100
12	8,342837	113,597101
13	8,342715	113,597077
14	8,342611	113,597077
15	8,342391	113,597064
16	8,342277	113,597091
17	8,342172	113,597102
18	8,342051	113,597127
19	8,341978	113,597149
20	8,341890	113,597174
21	8,341815	113,597192
22	8,341705	113,597207
23	8,341581	113,597231
24	8,341508	113,597274
25	8,341433	113,597289
26	8,341354	113,597293
27	8,341274	113,597307
28	8,343220	113,597322
29	8,343157	113,597308
30	8,343088	113,597300
31	8,343019	113,597289
32	8,343194	113,597412
33	8,343109	113,597406
34	8,343034	113,597395
35	8,342816	113,597262
36	8,342637	113,597229
37	8,342451	113,597256
38	8,342276	113,597268
39	8,342180	113,597279
40	8,342064	113,597316
41	8,342820	113,597377
42	8,342631	113,597364

Rumah	Koordinat (°S)	Koordinat (°E)
43	8,342458	113,597359
44	8,342298	113,597370
45	8,342137	113,597405
46	8,341984	113,597435
47	8,341574	113,597416
48	8,341422	113,597473
49	8,344192	113,597541
50	8,344069	113,597551
51	8,343881	113,597527
52	8,344208	113,597753
53	8,344031	113,597708
54	8,343900	113,597717
55	8,343643	113,597497
56	8,343482	113,597526
57	8,343610	113,597682
58	8,343611	113,597769
59	8,343458	113,597741
60	8,343427	113,597551
61	8,343277	113,597592
62	8,343155	113,597580
63	8,343183	113,597739
64	8,343098	113,597725
65	8,343017	113,597703
66	8,342832	113,597524
67	8,342616	113,597514
68	8,342429	113,597513
69	8,342867	113,597715
70	8,342747	113,597703
71	8,342491	113,597686
72	8,342387	113,597718
73	8,342290	113,597723
74	8,342195	113,597721
75	8,342084	113,597726
76	8,341904	113,597814
77	8,341637	113,597754
78	8,341432	113,597880
79	8,344274	113,597977
80	8,344027	113,598014
81	8,343915	113,597990
82	8,343735	113,598002
83	8,343626	113,597989
84	8,343444	113,597990
85	8,343435	113,598197

Rumah	Koordinat (°S)	Koordinat (°E)
86	8,343428	113,598310
87	8,343424	113,598401
88	8,343423	113,598493
89	8,343289	113,598006
90	8,343137	113,597971
91	8,342991	113,597993
92	8,343108	113,598171
93	8,342980	113,598167
94	8,342808	113,597935
95	8,342790	113,598022
96	8,342803	113,598165
97	8,342228	113,597968
98	8,342280	113,598079
99	8,342146	113,598098
100	8,343260	113,598332
101	8,343179	113,598343
102	8,343089	113,598373
103	8,342988	113,598419
104	8,343252	113,598471
105	8,342815	113,598359
106	8,342650	113,598341
107	8,342492	113,598328
108	8,342799	113,598556
109	8,342679	113,598516
110	8,342558	113,598525
111	8,343324	113,597704

3. Menentukan *k-centroid* secara *random*
   Data *k-centroid* yang digunakan yaitu koordinat ODP.
4. Menghitung jarak terpendek dari setiap data koordinat rumah ke masing-masing *centroid* menggunakan persamaan (1).
5. Menentukan jarak terpendek dari setiap data koordinat rumah ke masing-masing *centroid.*
   Hasil yang didapatkan pada langkah ini yaitu pengelompokan data.
6. Menghitung rata-rata dari data pada setiap *cluster*
   Data yang dihitung merupakan data jarak terpendek pada setiap *cluster*. Hasil rata-rata digunakan sebagai titik *centroid* baru untuk langkah selanjutnya.
7. Mengulangi langkah 3-6
   Jika titik *centroid* baru tidak sama dengan titik *centroid* sebelumnya, maka dilakukan perulangan.

Hasil akhir berupa koordinat ODP dari algoritma *k-means* akan digunakan sebagai data awal pada algoritma kruskal. Data ini dapat dilihat pada Tabel 2.

Tabel 2. Koordinat ODP

ODP	Koordinat (°S)	Koordinat (°E)
1	8,3435655	113,5973503
2	8,3418878	113,5971698
3	8,342634818	113,5972436
4	8,341657667	113,597816
5	8,341449429	113,597326
6	8,342201889	113,5973088
7	8,343121667	113,5972948
8	8,344012857	113,5978801
9	8,343454875	113,5978091
10	8,342947	113,597803
11	8,342262625	113,5978399
12	8,342566667	113,598398
13	8,342963833	113,5982757
14	8,343343	113,5983639
15	8,344052	113,5974304
16	8,342739	113,598536

Adapun langkah-langkah dari algoritma kruskal adalah sebagai berikut:

1. Himpunan sisi dari graf G (graf awal) diurutkan dari bobot terkecil ke terbesar. Bobot didapatkan dari jarak antar ODP yang dihitung menggunakan persamaan (1).
2. Membuat graf T dengan memasukkan 1 sisi terpendek dari graf G
3. Diulangi (sebanyak sisi), jika sisi yang diambil tidak membentuk sikel di graf T, maka sisi tersebut dimasukkan di graf T beserta dengan titik-titiknya.

3. Hasil dan Pembahasan

Data penelitian berupa koordinat rumah didapatkan melalui aplikasi *google earth.* Peta lokasi dapat dilihat pada Gambar 1.

Gambar 1. Peta lokasi penelitian

Algoritma *K-Means*

Metode pertama yang akan dibahas yaitu algoritma *k-means.* Jumlah data awal yang digunakan berjumlah 111 rumah. Kemudian dibentuk 16 *cluster* sebagai acuan untuk pengelompokan. Penentuan *centroid* (titik pusat *cluster*) dilakukan secara *random.* Koordinat ODP awal digunakan sebagai *centroid.* Koordinat ODP dapat dilihat pada Tabel 3.

Tabel 3. Koordinat ODP awal

ODP	Koordinat (°S)	Koordinat (°E)
1	8,343718	113,597458
2	8,341938	113,597251
3	8,342494	113,597132
4	8,341812	113,598024
5	8,341734	113,597629
6	8,342241	113,597516
7	8,342990	113,597541
8	8,343825	113,597865
9	8,343420	113,597921
10	8,342879	113,597885
11	8,342291	113,597909
12	8,342574	113,598280
13	8,342935	113,598235
14	8,343302	113,598231
15	8,3442	113,597409
16	8,342809	113,598666

Kemudian menghitung jarak sekaligus menentukan jarak terpendek setiap koordinat rumah ke masing-masing *centroid* (koordinat ODP). Perhitungan ini menggunakan rumus *Euclidian Distance* pada persamaan (1). Setelah mendapatkan jarak terpendek,

menghitung rata-rata setiap *cluster*. Hasil rata-rata ini digunakan sebagai titik *centroid* baru (koordinat ODP baru) dapat dilihat pada Tabel 4.

Tabel 4. Koordinat ODP baru

ODP	Koordinat (°S)	Koordinat (°E)
1	8,3436015	113,5974011
2	8,341942	113,5972212
3	8,342556444	113,5971798
4	8,341904	113,597814
5	8,34145425	113,5974608
6	8,3422365	113,5975005
7	8,34305595	113,5973987
8	8,343927571	113,5978824
9	8,3434282	113,597886
10	8,34289	113,5978898
11	8,342303667	113,5978787
12	8,342566667	113,598398
13	8,342963833	113,5982757
14	8,343343	113,5983639
15	8,3441174	113,5974756
16	8,342739	113,598536

Setelah mendapat koordinat ODP baru, data ini digunakan sebagai data awal untuk perulangan selanjutnya. Perulangan berhenti pada perulangan ke-5, karena koordinat ODP awal dan ODP baru sama. Koordinat ODP akhir dapat dilihat pada Tabel 5.

Tabel 5. Koordinat ODP akhir

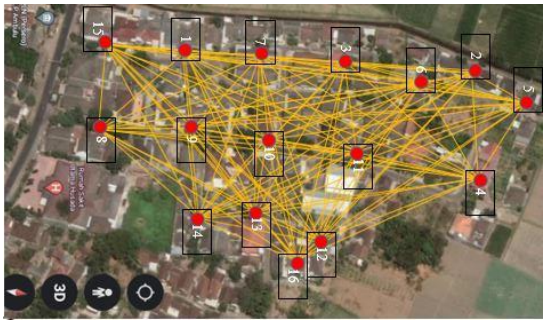
ODP	Koordinat (°S)	Koordinat (°E)
1	8,3435655	113,5973503
2	8,3418878	113,5971698
3	8,342634818	113,5972436
4	8,341657667	113,597816
5	8,341449429	113,597326
6	8,342201889	113,5973088
7	8,343121667	113,5972948
8	8,344012857	113,5978801
9	8,343454875	113,5978091
10	8,342947	113,597803
11	8,342262625	113,5978399
12	8,342566667	113,598398
13	8,342963833	113,5982757
14	8,343343	113,5983639
15	8,344052	113,5974304
16	8,342739	113,598536

Hasil *clustering* data koordinat rumah dengan koordinat ODP:

a) ODP 1 memiliki kapasitas 8, karena mendistribusikan kabel ke rumah 3,4,5,6,7,55,56,60.
b) ODP 2 memiliki kapasitas 8, karena mendistribusikan kabel ke rumah 18,19,20,21,22.
c) ODP 3 memiliki kapasitas 12, karena mendistribusikan kabel ke rumah 12,13,14,15,35,36,37,41,42,43,67.
d) ODP 4 memiliki kapasitas 8, karena mendistribusikan kabel ke rumah 76,77,78.
e) ODP 5 memiliki kapasitas 8, karena mendistribusikan kabel ke rumah 23,24,25,26,27,47,48.
f) ODP 6 memiliki kapasitas 12, karena mendistribusikan kabel ke rumah 16,17,38,39,40,44,45,46,68.
g) ODP 7 memiliki kapasitas 12, karena mendistribusikan kabel ke rumah 8,9,10,11,28,29,30,31,32,33,34,62.
h) ODP 8 memiliki kapasitas 8, karena mendistribusikan kabel ke rumah 52,53,54,79,80,81,82.
i) ODP 9 memiliki kapasitas 8, karena mendistribusikan kabel ke rumah 57,58,59,61,83,84,89,111.
j) ODP 10 memiliki kapasitas 12, karena mendistribusikan kabel ke rumah 63,64,65,66,69,70,90,91,94,95.
k) ODP 11 memiliki kapasitas 8, karena mendistribusikan kabel ke rumah 71,72,73,74,75,97,98,99.
l) ODP 12 memiliki kapasitas 8, karena mendistribusikan kabel ke rumah 106,107,110.
m) ODP 13 memiliki kapasitas 8, karena mendistribusikan kabel ke rumah 92,93,96,102,103,105.
n) ODP 14 memiliki kapasitas 8, karena mendistribusikan kabel ke rumah 85,86,87,88,100,101,104.
o) ODP 15 memiliki kapasitas 8, karena mendistribusikan kabel ke rumah 1,2,49,50,51.
p) ODP 16 memiliki kapasitas 8, karena mendistribusikan kabel ke rumah 108 dan 109.

Algoritma Kruskal

Data awal yang digunakan pada algoritma kruskal yaitu data koordinat ODP akhir dari algoritma *k-means.* Data ini dapat dilihat pada Tabel 5. Data koordinat tersebut dihitung jarak antar ODP menggunakan persamaan (1). Data jarak digunakan sebagai data bobot sisi pada graf terhubung awal yang dibentuk. Graf awal dapat dilihat pada Gambar 2.

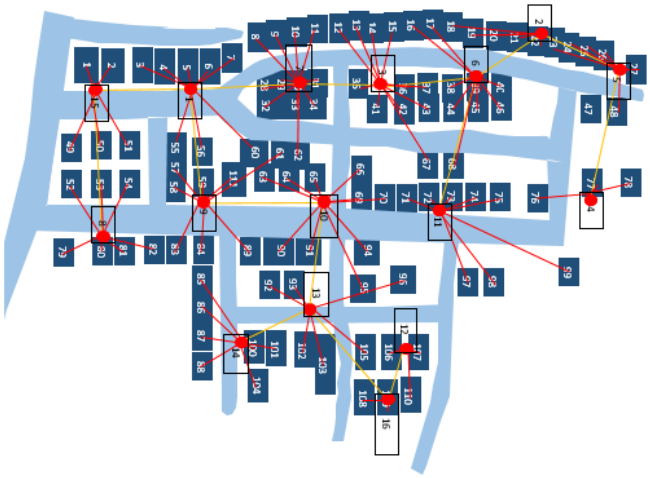


Gambar 2. Graf awal

Bobot setiap sisi diurutkan dari bobot terkecil hingga terbesar. Setelah itu membuat graf baru dengan memasukkan sisi dengan bobot terkecil. Langkah ini diulang sebanyak jumlah sisi. Jika sisi yang diambil tidak membentuk sikel pada graf baru, maka sisi tersebut dimasukkan di graf baru beserta dengan titik-titiknya. Hasil akhir berupa graf baru dapat dilihat pada Gambar 3.

Gambar 3. Hasil akhir graf

Berdasarkan proses algoritma *k-means* dan kruskal didapatkan hasil akhir berupa *clustering* rumah dan desain jaringan internet *indihome* di Perumahan Manggar Permai Ambulu seperti pada Gambar 4 berikut.



Gambar 4. Hasil Desain Jaringan Internet *Indihome*

Hasil yang didapatkan dari kedua algoritma di atas sudah sesuai dengan rumusan masalah dari penelitian ini, meskipun pada algoritma *k-means* jumlah distribusi tidak sesuai dengan jumlah kapasitas per ODP yang digunakan. Hal ini sudah disesuaikan dengan jarak terpendek agar mendapatkan panjang kabel minimum. Kabel yang digunakan oleh pelanggan *indihome* dari pihak telkom maksimal sepanjang 250 meter, jika melebihi itu akan dikenakan biaya tambahan. Jika jumlah distribusi mengikuti kapasitas ODP, maka tidak akan menghasilkan panjang kabel minimum dan akan ada biaya tambahan, karena terdapat jarak antar ODP yang melebihi jarak 250 meter.

4. Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa *clustering* pada rumah dengan ODP terbentuk 16 *cluster* yang diproses menggunakan algoritma *k-means* dan berhenti pada perulangan ke-5. ODP yang digunakan memiliki kapasitas 8 dan 12. ODP 1, 2, 4, 5, 8, 9, 11, 12, 13, 14, 15, dan ODP 16 memiliki kapasitas 8. Sedangkan ODP 3, 6, 7, dan ODP 10 memiliki kapasitas 12. Kabel fiber optik antar ODP diproses menggunakan algoritma kruskal dan menghasilkan bobot minimum yaitu 0,006601782 atau 735 meter.

Daftar Pustaka

[1] Bastian, A., H. Sujadi, dan G. Febrianto. (2018). Penerapan Algoritma K-means Clustering Analysis pada Penyakit Menular Manusia (Studi Kasus Kabupaten Majalengka). *Jurnal Sistem Informasi.* **14(1)**, 026-032.

[2] Budayasa, I. K. (2007). *Teori Graph dan Aplikasinya.* Surabaya: Unesa University Press.

[3] Ediyanto, M. N. Mara, dan N. Satyahadewi. (2013). Pengklasifikasian Karakteristik dengan Metode K-Means Cluster Analysis. *Buletin Ilmiah Matematika, Statistika, dan Terapannya (Bimaster).* **2(2)**, 133-136.

[4] Hung, M. Chuan, J. Wu, J. H. Chang, dan D. L. Yang. (2005). An Efficient K-Means Clustering Algorithm Using Simple Partitioning. *Journal of Information Science and Engineering.* **2(1)**, 1157-1177.

[5] Priyono, N. Muhassanah, & A. Winarni. (2020). Algoritma Kruskal Menentukan Lintasan Terpendek Efektif Call Salesman. *Fundamental Scientific Journal of Mathematic.* **1(2)**, 25-32.

[6] Tambunan, T. L. (2012). *Pengoptimalan Jaringan Distribusi Air dengan Menggunakan Algoritma Kruskal Pada Perusahaan Daerah Air Minum (PDAM) Tirtanadi Cabang Tuasan.* Universitas Negeri Medan: Jurusan Matematika.