REPOSITOR, Vol. 2, No. 12, Desember 2020, Pp. 1631-1638
ISSN : 2714-7975
E-ISSN : 2716-1382

Deteksi Botnet Pada Passive DNS Dengan Menggunakan Metode K Nearest Neighbor

Vinna Utami Putri[1], Eko Budi Cahyono[2], Yufis Azhar[3]
[1,2,3]Universitas Muhammadiyah Malang
e-mail: [1]vinnautamiputri@gmail.com, [2] ebcahyono@yahoo.com, [3] yufis.az@gmail.com

Abstrak

Teknologi internet di masa kini berkembang dengan pesat berbanding lurus dengan penggunanya yang juga semakin banyak. Salah satu kejahatan *software* yang berbahaya adalah *robot network (Botnet). Botnet* adalah sebuah *zombie* dalam jaringan dari jutaan perangkat yang tersambung ke internet, yang mana *bot* diinfeksi dengan malware yang khusus agar bisa dikendalikan oleh *cybercriminal* dari jarak jauh untuk memberikan serangan seperti mengirim email, mencuri informasi pribadi, dan meluncurkan serangan DDoS.

Pada penelitian ini penulis mengelompokan dan mengklasifikasikan dataset yang *botnets* dan normal pada *passive DNS* yang terdapat pada dataset CTU-13 dengan metode *k Nearest Neighbor* dan juga pengujian dengan mengunakan *confusion matrix* dengan nilai *precision*, *recall* dan *accuracy* dari *k-nearest neighbor* dari standart bahasa pemograman python dengan library sciketlearn disetiap kelas prediksi dan hasil yang dicapai cukup tinggi dengan nilai dari *uniform dengan nilai 76%* untuk *precission 86%* dan *recal*-nya 93,9% untuk *accuracy*. *Uniform ternormalisasi* dengan nilai *76%* untuk *precission 88%* dan *recal*-nya 83% untuk *accuracy*. Hasil *Distance* didapatkan nilai *100%* untuk *precission 85%* dan *recal*-nya 92% untuk *accuracy*. Hasil *Distance ternormalisasi 100%* untuk *precission 87%* dan *recal*-nya 93% untuk *accuracy*.
Kata kunci: *Network, Botnet, kNN.*

Abstract

. The present internet technology develops by leaps and bounds is directly proportional to its users which is also more and more. One of the crime of malicious software is a robot network (Botnet). A botnet is a network of millions of zombies in a device that is connected to the internet, which is where a bot infection with malware that specifically so that it can be controlled by the cybercriminal remotely to provide attack such as sending email, steal personal information, and launching DDoS attacks.

In this study the authors classify and classify the botnets and normal dataset on a passive DNS contained on dataset CTU-13 k Nearest Neighbor method and also testing using confusion matrix with values of precision, recall and the accuracy of k-nearest neighbor of the python programming language with the standard library sciketlearn every class predictions and results achieved high enough with the value of the uniform with a value of 76% to 86% and precission recal its 93.9% for accuracy. Uniform ternormalisasi with a value of 76% to 88% and precission recal 83% for its accuracy. The results obtained by the value of 100% Distance for precission 85% and 92% of his recal for accuracy. Ternormalisasi 100% Distance results for precission 87% and 93% of his recal for accuracy.

Keywords*:* ***Kata kunci:*** *Network, Botnet, kNN.*

1. Pendahuluan

Teknologi internet di masa kini berkembang dengan pesat berbanding lurus dengan penggunanya yang juga semakin banyak. Internet bukan hanya digunakan untuk sekedar bertukar segala macam informasi namun mulai juga digunakan untuk keperluan manusia untuk transaksi seperti finansial dan infrastruktur teknologi informasi dan komunikasi. Hal ini menyebabkan sangat banyak data yang berharga tersebar melalui jaringan di internet. Hal ini menjadi salah satu faktor seseorang yang ingin mencari celah keamanan di internet dan menyalah gunakannya dengan tindakan yang salah. Tindakan penyalahgunaan internet saat ini sudah berbeda dengan tindakan yang terdahulu dimana masa kini tindakan seperti menyerang server atau suatu perangkat yang ada di dalam jaringan, kebanyakan serangan yang dilakukan

pada saat ini adalah untuk mencari keuntungan finansial, tindakan tersebut sangat mengancam jutaan orang yang beraktivitas dalam penggunaan internet. [1] Kegiatan serangan internet lainnya adalah pencurian informasi pribadi, *spam mail* dan juga meluncurkan serangan *Distributed Denial of Service (DDoS)* [2].

Salah satu kejahatan *software* yang berbahaya adalah *robot network (Botnet). Botnet* adalah sebuah *zombie* dalam jaringan dari jutaan perangkat yang tersambung ke internet seperti *Personal Computer* (PC), *smartphone, tablet, router* dan *gadget* lainnya. Yang mana *bot* diinfeksi dengan malware yang khusus agar bisa dikendalikan oleh *cybercriminal* dari jarak jauh untuk memberikan serangan seperti mengirim email, mencuri informasi pribadi, dan meluncurkan serangan DDoS.[3]

Terdapat beberapa penelitian sebelumnya yang membahas mengenai pendeteksian botnet seperti pada thesis Pedro Marques da Luz pada tahun 2013/2014 dengan judul *Botnet Detection Using Passive DNS* kemudian penelitian yang dilakukan oleh Septian Gegas pada tahun 2013 dengan judul Identifikasi Botnets Melalui Pemantauan Group Activity Pada DNS Traffic yang dirancang untuk mendeteksi botnet yaitu dengan melakukan pemantauan aktivitas grup dalam query DNS.

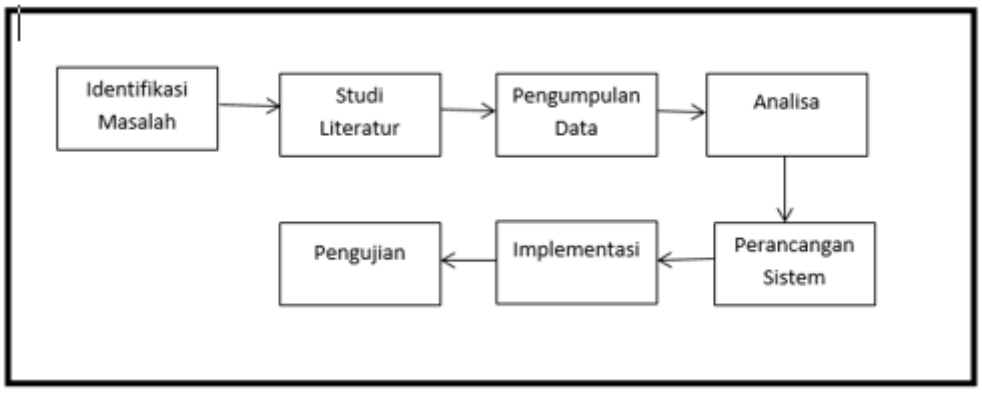
Perbedaan pembahasan yang dipaparkan oleh peneliti berupa pemantauannya di *passive DNS* yang berbeda dan metode yang berbeda dari penelitian yang dilakukan sebelumnya yaitu dengan melakukan pengujian *passive DNS* yang didapatkan dari dataset CTU-13 dengan metode *kNN* dan diuji dengan *K-Fold Cross Validation dan Compusion Metric.*

Terdapat tiga komponen agar sebuah botnet dapa berjalan yaitu *Botmaster, Command and Control* (C&C) server dan *bot*. Kegiatan untuk merekam atau mengendalikan suatu komputer yang terinfeksi yaitu dengan menggunakan C&C sebagai server pusat, mengendalikan C&C diperankan oleh *botmaster* itu sendiri, dan karena itulah sebuah serangan *botnet* dapat dilakukan dari jarak jauh.[3] Untuk mendeteksi apakah suatu komputer apakah terjangkit botnet maka akan dilakukan pendeteksian dini dengan memanfaatkan fitur *passive DNS* sehingga dapat diprediksi apakah didalam data tersebut mengandung botnet atau tidak.

Domain Name System (DNS) merupakan sebuah layanan yang digunakan untuk menerjemahkan alamat IP ke alamat domain dan juga menerjemahkan alamat domain ke alamat IP, agar lebih mudah diingat oleh ingatan manusia.

Passive DNS menurut [4] adalah sebuah database yang berisi data dari DNS itu sendiri seperti *DNS-client, DNS-server, domain name, time stamp, client IP query type (RR), query name, answer, TTL.* Dari katagori passive DNS tersebut sehingga dapat dianalisis mana DNS yang mengandung botnet dan mana DNS yang tidak mengandung botnet. Untuk mengidentifikasi *passive DNS* yang teridentifikasi *botnet* atau tidak peneliti meperlukan sebuah metode yaitu k-Nearest Neigbours (kNN) yang bertugas untuk mengklasifikasi data *passive DNS*.

2. Metode Penelitian



Gambar 1 Metodologi Penelitian

2.1 Identifikasi Masalah

Pada tahap ini merupakan langkah pertama yang dilakukan oleh penulis untuk menentukan topik dari penelitian dan mengidentifikasi dari permasalahan yang ada, identifikasi ini bermaksud untuk penegassan batasan-batasan dari penegasan batasan permasalahan agar penelitian ini sesuai dengan tujuan.

Internet bukan hanya digunakan untuk sekedar bertukar segala macam informasi namun mulai juga digunakan untuk keperluan manusia untuk transaksi seperti finansial dan infrastruktur teknologi informasi dan komunikasi. Hal ini menyebabkan sangat banyak data yang berharga tersebar melalui jaringan di internet. Hal ini menjadi salah satu faktor seseorang yang ingin mencari celah keamanan di internet dan menyalah gunakannya dengan tindakan yang salah. Tindakan penyalahgunaan internet saat ini sudah berbeda dengan tindakan yang terdahulu dimana masa kini tindakan seperti menyerang server atau suatu perangkat yang ada di dalam jaringan, kebanyakan serangan yang dilakukan pada saat ini adalah untuk mencari keuntungan finansial, tindakan tersebut sangat mengancam jutaan orang yang beraktivitas dalam penggunaan internet. [1] Kegiatan serangan internet lainnya adalah pencurian informasi pribadi, *spam mail* dan juga meluncurkan serangan *Distributed Denial of Service (DDoS)* [2]. Salah satu kejahatan *software* yang berbahaya adalah *robot network (Botnet). Botnet* adalah sebuah *zombie* dalam jaringan dari jutaan perangkat yang tersambung ke internet seperti *Personal Computer* (PC), *smartphone, tablet, router* dan *gadget* lainnya. Yang mana *bot* diinfeksi dengan malware yang khusus agar bisa dikendalikan oleh *cybercriminal* dari jarak jauh untuk memberikan serangan seperti mengirim email, mencuri informasi pribadi, dan meluncurkan serangan DDoS.[3]

Melihat dari permasalahan yang ada penulis berkeninginan untuk dapat mengelompokan, mengklasifikasikan dan mendapatkan pengujian yang tepat dari dataset yang mengandung botnets dan dataset normal yang mana dataset tersebut diambil dari CTU 13. Peneliti melakukan penelitian ini dengan menggunakan metode K Nearest Neighbor untuk mengklasifikasikan datasetnya dan juga Confusion Matrix untuk pengujiannya.

2.2 Studi Literatur

Pada tahap studi literatur ini penulis mengumpulkan dan mempelajari berbagai literatur baik dari jurnal, buku, maupun artikel penunjang yang berhubungan dengan botnet, DNS, Passive DNS, DNS Traffic, aktivitas di DNS, K Nearest Neighbor, Confusion Matrix, bahasa pemograman python dan lain-lain untuk mendapatkan pemahaman tentang apa yang akan dilakukan berdasarkan studi literatur terdahulu yang dapat menunjang dalam melakukan penelitian ini.

2.3 Pengumpulan Data

Pengumpulan data yang dilakukan oleh penulis berupa dataset yang didapatkan dari University Czech Republic sejak tahun 2011 yaitu data CTU 13. Pada data tersebut terdapat indikasi terjadinya infeksi botnet namun belum dapat diketahui karakteristik infeksi tersbut sehingga pada penelitian ini penulis ingin mengidentifikasikan dan mengklasifikasi botnet tersebut berdasarkan hasil seleksi fitur.

Data set CTU 13 merupakan sekumpulan traffic jaringan yang direkam oleh CTU University Czech Republic sejak tahun 2011. [5] Tujuan dari proses capture data ini adalah untuk menyediakan data set penelitian traffic jaringan yang mengandung botnet dan memberikan gambaran serta data yang digunakan untuk analisa perkembangan botnet.

2.4 Analisa

2.4.1 Analisa Data

Pada tahap ini penulis menganalisa data dari dataset CTU 13, dari 13 folder yang ada pada dataset tersebut data mana yang akan di jadikan data testing ataupun data training.
Dataset CTU-13 [5] bertujuan sebagai berikut :

a. Dataset bukan simulasi, dataset CTU-13 serupakan serangan yang sebenarnya.
b. Trafik yang aktual, artinya tidak dikondisikan seperti simulasi dan *capture* trafik inftastruktur berjalan sepeti biasanya.
c. Untuk perlabelan memiliki dasar dan evaluasi yang dibahas di [5] yang artinya host yang terinfeksi bot pada proses *capture* telah dilabeli sebagai trafic botnet.

2.4.2 Analisa Kebutuhan

Pada tahap ini akan dilakukan identifikasi analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa diperoleh dari penelitian, percobaan, konsultasi dengan pakar dan studi literatur.

Berdasarkan studi literatur yang berhubungan dengan Botnet dapat dijelaskan bahwa botnet merupakan suatu malware yang digunakan untuk mengontrol suatu komputer dan

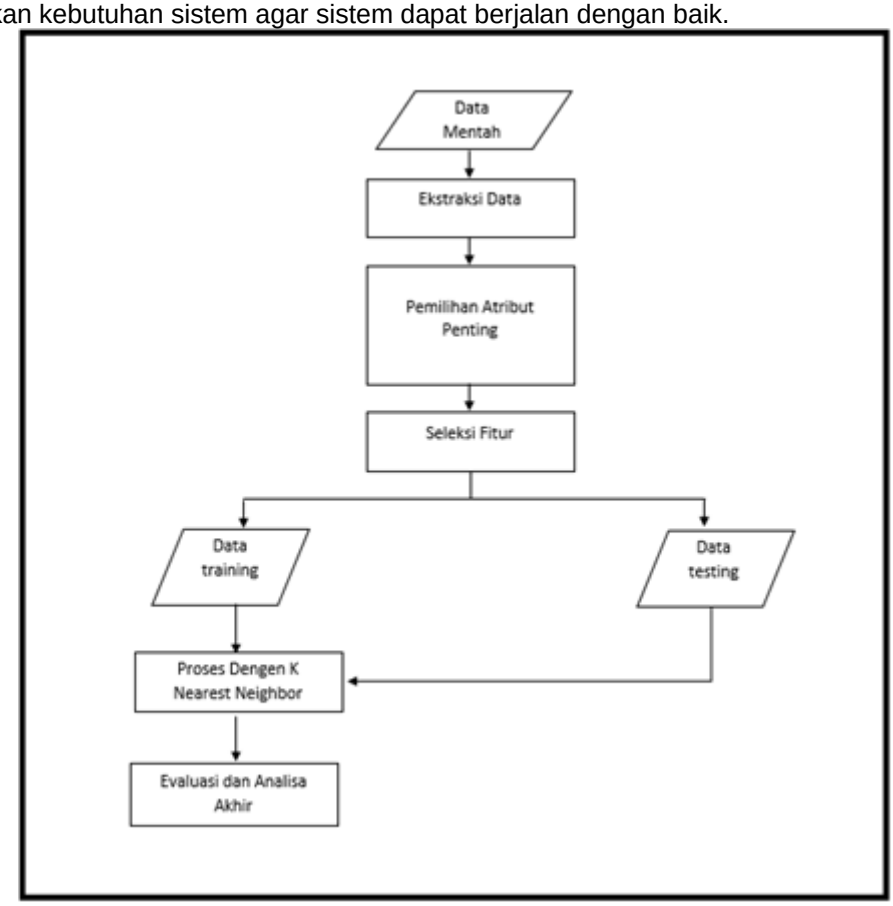
menunggu perintah dari Botmaster. Untuk menganalisa keberadaan lalu lintas data yang mengandung atau terindikasi data dari suatu Botnet maka peneliti akan menggunakan k-Nearest Neighbor guna memilah dan memberi klasifikasi dari paket data tersebut. Dengan ekstrasi Feature seperti TTL, Source Address, Destination Address serta beberapa parameter lain akan digunakn sebagai fitur yang diproses oleh kNN.

2.4.3 Analisa Masalah

Permasalahan yang dialami oleh pengelola jaringan dan pengamat sistem keamanan komputer adalah pesatnya perkembangan malware seperti botnet yang tersebar di internet yang tanpa sengaja diunduh oleh user dalam jaringan yang mereka kelola, sehingga dibutuhkan suatu langkah pengembangan sistem yang mampu mengklasifikasi *malware* dalam jaringan, pada penelitian ini dikhususkan botnet dan dari penelitian ini yang menggunakan data CTU 13 dan data tersebut harus diolah menjadi *data passive dns* setelah itu mengekstraksi data tersebut hingga melakukan pelabelan data yang bersifat normal atau data yang mengandung botnet sehingga dapat dikembangkan untuk sistem yang lebih bersifat aktif kedepannya.

2.5 Perancangan Sistem

Perancangan sistem akan menjelaskan proses kegiatan yang akan di terapkan dan menjeaskan kebutuhan sistem agar sistem dapat berjalan dengan baik.

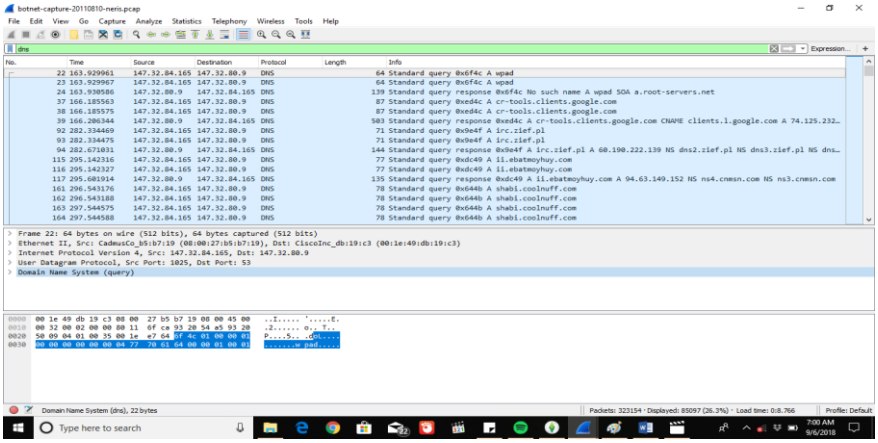


Gambar 2 flowchat program

2.5.1 Ekstraksi Data

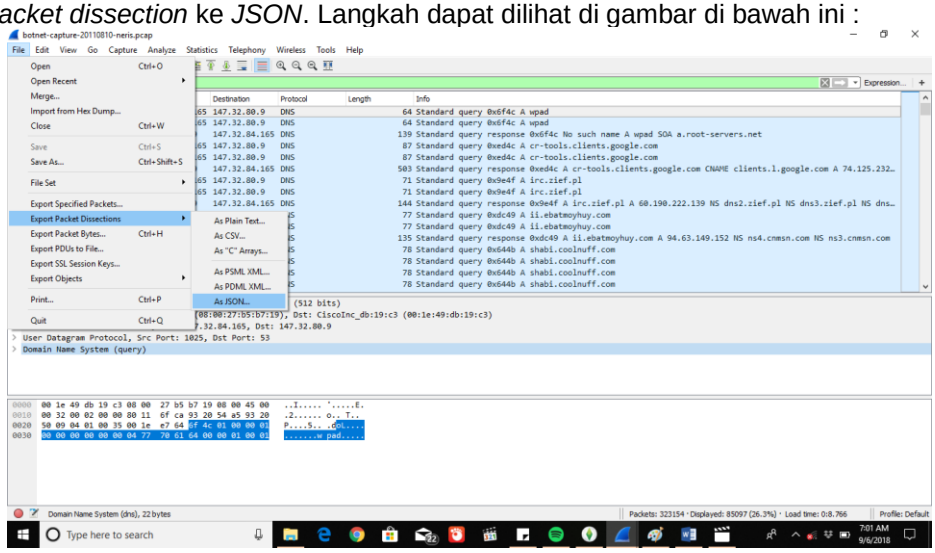
Proses dari ekstraksi fitur dimulai dari membaca file awal dataset (data mentah) ekstraksi fitur merupakan suatu pengambilan ciri atau fitur dari suatu bentuk yang nilainya akan dapat dianalisis untuk diproses.

Langkah selanjutnya mengubah format data *CTU-13* yang masih berformat *pcap file* diubah menjadi format *JSON*. Pertama *pcap file* dapat dibuka melalui aplikasi *Wireshark* yaitu dapat dilihat di gambar berikut :



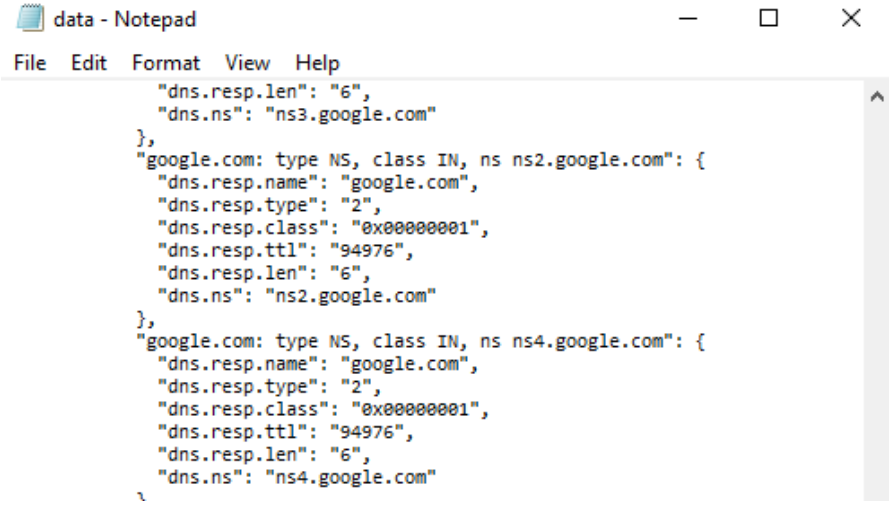
Gambar 3 Data yang masih berformat pcap file

Didalam aplikasi *wireshark* ini informasi yang terlihat hanya *Time*, *Source*, *Destination*, *Protocol*, *Length* dan *Info* sedangkan infromasi yang lainnya masih banyak tersembunyi agar bisa mengeluarkan informasi yang tersembunyi tersebut peneliti mengkonversi dari *pcap file* ke *json* melalui aplikas *wireshark* itu sendiri. Pertama dengan membuka *file* yang berformat *pcap file* pilih *export packet dissection* ke *JSON*. Langkah dapat dilihat di gambar di bawah ini :



Gambar 4 Data pcap file ke json

Dibawah ini gambar dari hasil perubahan format pcap files ke format json, adapun datanya yang berformat json dapat dibuka dengan menggunakan aplikasi notepad :



data - Notepad

File Edit Format View Help


```
    "dns.resp.len": "6",
    "dns.ns": "ns3.google.com"
  },
  "google.com: type NS, class IN, ns ns2.google.com": {
    "dns.resp.name": "google.com",
    "dns.resp.type": "2",
    "dns.resp.class": "0x00000001",
    "dns.resp.ttl": "94976",
    "dns.resp.len": "6",
    "dns.ns": "ns2.google.com"
  },
  "google.com: type NS, class IN, ns ns4.google.com": {
    "dns.resp.name": "google.com",
    "dns.resp.type": "2",
    "dns.resp.class": "0x00000001",
    "dns.resp.ttl": "94976",
    "dns.resp.len": "6",
    "dns.ns": "ns4.google.com"
```


Gambar 5 Data CTU-13 dalam format .*JSON*

2.5.2 Pemilihan Atribut

Pemilihan atribut penting dalam passive DNS adalah untuk mencari fitur-fitur yang hanya digunakan pada passive DNS untuk mendeteksi serangan-serangan yang terjadi, menurut penelitian dari Pedro dengan judul penelitiannya Botnet Detection Using Passive DNS dia mengambil delapan atribut untuk dapat mendeteksi botnet di passive DNS [4] diantaranya *DNS Client, DNS Server, Query Class, Time Stamp, Query Type (RR), Query (Domain Name), Answer, TTL (Time To Life)*

Tabel 1 Ektraksi Atribut

NO	Fitur	Keterangan Atribut
1	*DNS Source IP Client*	*IP address client* yang mengakses ke *server*
2	*DNS Destination IP Server*	*IP address server* yang dituju oleh *client*
3	*Query Class*	Kode yang menentukan kelas *query*
4	*Time Stamp*	Keterangan waktu pada suatu paket
5	*Query Type (RR)*	Kumpulan sumber informasi yang berhubungan dengan nama-nama domain.
6	*Query*	Perintah yang dilakukan oleh *client* saat *request* ke *server*
7	*Answer*	Jawaban sukses atau tidaknya *request*
8	*TTL (Time To Life)*	satuan lama pengiriman dan pemrosesan paket

Proses pemilihan atribut di dalam data dilakukan dengan proses coding python yang dapat dikerjakan dari beberapa sumber contohnya melalui *command prompt,* IDLE 64 bit, maupun di notepad. Running program dapat di running melalui cmd dan IDLE 64 bit. Running program melalui cmd cukup dengan menulis nama file dan path nya harus benar sesuai dengan file yang disimpan didalam folder tersebut. Running program melalui IDLE 64 bit mudah sekali cukup dengan memilih “Run” di bar utama.

Pemilihan atribut yang dibuat dan disimpan dengan nama file pyctu.py. databaru.json dan file pyctu.py harus disimpan di folder yang sama karena jika tidak disimpan di folder yang sama, file pyctu.py tidak bisa running. Seperti gambar dibawah ini.

2.5.3 Seleksi Fitur

Pemilihan fitur digunakan untuk menghilangkan fitur yang tidak relevan dan akan menyebabkan kekacauan. Penelitian ini akan menggunakan seleksi fitur karena jumlah fitur yang terlalu banyak dan hanya akan menggunakan fitur yang bermaanfaat dan berguna saja untuk penelitian agar hasilnya lebih akurat. [4][6] [7][8]

Tabel 2 Seleksi Fitur

Nama Fitur	Atribut yang Digunakan	Keterangan
F1	Nilai *Time to Live TTL*	Nilai *TTL* akan sangat berguna ketika dilakukan suatu evaluasi berdasarkan berapa lama suatu paket dapat beredar di dalam suatu jaringan, untuk mengetaui pola sebuah paket
F2	*DNS query answer*	seberapa banyak alamat IP yang digunakan pada suatu alamat domain, karena dari variabel ini kita dapat menduga karakteristik paket normal dan botnet
F3	*Time feature*	dalam fitur ini akan menjelaskan seberapa sering suatu paket melintas dalam jaringan dan

		pola yang muncul pada paket tersebut
F4	*IP Geolocate*	fitur berdasarkan lokasi suatu *ip address* pada alamat domain tertentu, akan menandakan ada atau tidaknya perubahan lokasi pada suatu *ip domain*
F5	*Autonomous Domain*	fitur berdasarkan seringnya pencarian dan pengalihan akses nama domain ke alamat *ip*

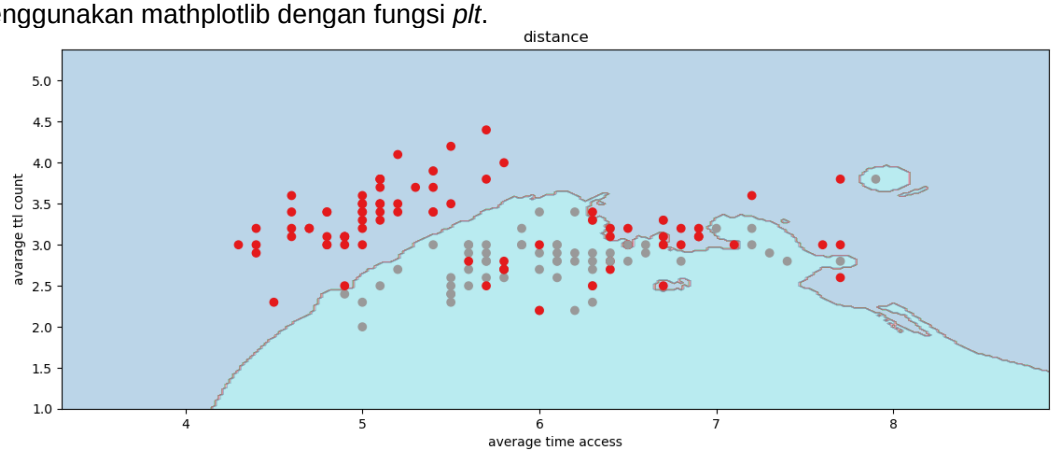
3. Hasil Penelitian dan Pembahasan

Pada penelitian ini penulis dapat mengelompokan dan mengklasifikasikan dataset yang *botnets* dan normal pada *passive DNS* yang terdapat pada dataset CTU-13 dengan metode *k Nearest Neighbor*

```
for weights in ['uniform','distance']:
    knn = neighbors.KNeighborsClassifier(15, weights =
weights)
    knn.fit(X,y)
    x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
    y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
    xx, yy = np.meshgrid(np.arange(x_min, x_max, .02),
                         np.arange(y_min, y_max, .02))
    Z = knn.predict(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)
    plt.figure(figsize=(15, 5))
    plt.contourf(xx, yy, Z, cmap=plt.cm.tab10, alpha=0.3)
    plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Set1)
    plt.xlabel('average time access')
    plt.ylabel('avarage ttl count')
```

Gambar 6 Source code kNN

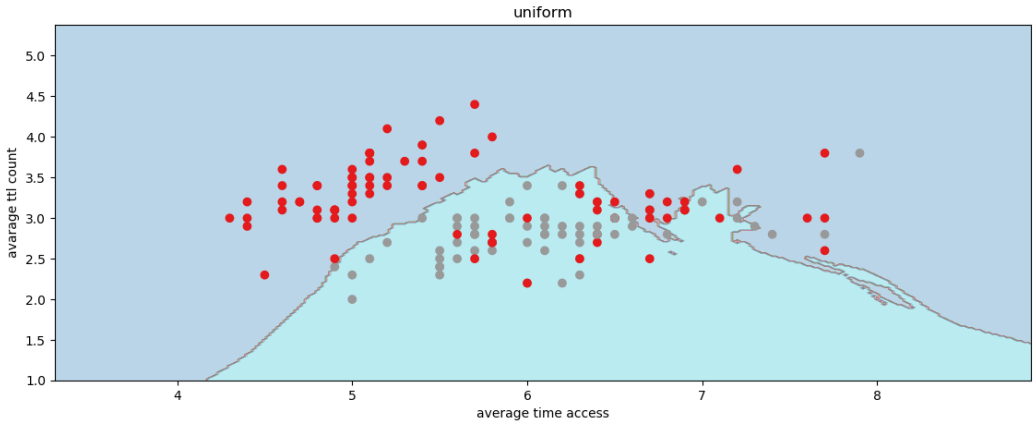
Pada Gambar 6 diatas merupakan bentuk *source code* untuk memproses matriks hasil ekstraksi data untuk kemudian diplot engan mnggunakan *matplotlib* pada *python* sehingga dapat diketahui hasil data berupa area. Cara kerja *source code* tersebut adalah, pada setiap model pembobotan yaitu "*uniform*" dan "*distance*" akan dilakukan suatu klasifikasi menggunakan *KNN* dengan nilai K=23 alasan nilai K sebesar lima belas adalah dari beberapa rujukan menggunakan plilihan nilai K dari 9,15,21 dan 23. Nilai X merupakan matriks data dan y merupakan label yang digunakan untuk membentuk area dan koordinat menggunakan *np.meshgrid* dan akan diplot menggunakan mathplotlib dengan fungsi *plt*.



Gambar 7 Bentuk Plot hasil KNN Distance

Hasil yang ditunjukan pada Gambar 7 dan 8 pada area berwana gelap dengan titik-titik merah merupakan indikasi dari paket yang bersifat botnet, pada domain tertentu dan area yang berwarna lebih terang dengan titik-titik abu-abu menandakan *traffic normal* sedangkan untuk tiitik-titik merah pada area lebih terang dapat dikatakan bukan botnet namun dikatakan botnet dan titik-titik abu-abu pada area yang lebih gelap merupakan botnet bukan dikatakan botnet. *K-*

nearest neighbor merupakan algoritma pemisahan data atau klasifikasi yang berdasar pada nilai jarak yang dihasilkan tiap set data di dalam suatu matriks, nilai jarak dari data-data tersebut menjadi dasar pengelompokan suatu label. Dengan menampilkan hasil klasifikasi dalam bentuk area seperti pada Gambar 7 dan 8 akan mempermudah dalam memahami suatu hasil dari proses pengolahan data.



Gambar 8 Bentuk Plot hasil KNN Uniform

4. Kesimpulan

Dari hasil penelitian yang telah penulis lakukan dapat disimpulkan penulis dapat mengelompokan dan mengklasifikasikan dataset yang *botnets* dan normal pada *passive DNS* yang terdapat pada dataset CTU-13 dengan metode *k Nearest Neighbor* dan bahasa pemograman *python* dan didapatkan hasil pengujian yang juga dilakukan dengan bahasa pemograman python dengan library sciketlearn disetiap kelas prediksi dan hasil yang dicapai cukup tinggi dengan nilai dari *uniform dengan nilai 76%* untuk *precission 86%* dan *recal*-nya 93,9% untuk *accuracy*. *Uniform ternormalisasi* dengan nilai *76%* untuk *precission 88%* dan *recal*-nya 83% untuk *accuracy*. Hasil *Distance* didapatkan nilai *100%* untuk *precission 85%* dan *recal*-nya 92% untuk *accuracy*. Hasil *Distance ternormalisasi 100%* untuk *precission 87%* dan *recal*-nya 93% untuk *accuracy*.

Refrensi

[1] S. Geges, W. Wibisono, and T. Ahmad, “Identifikasi Botnets Melalui Pemantauan Group activity Pada DNS Traffic,” *J. Tek. Pomits Vol. 2, No. 1, ISSN 2337-3539 (2301-9271 Print)*, vol. 2, no. 1, pp. 1–6, 2013.

[2] H. Lee, H. Choi, H. Lee, H. Lee, and H. Kim, “Botnet Detection by Monitoring Group Activities in DNS Traffic Botnet Detection by Monitoring Group Activities in DNS Traffic,” no. November, 2007.

[3] D. N. Fuadin, D. Pembimbing, P. Magister, D. T. Elektro, and F. T. Elektro, “DETEKSI BOTNET MENGGUNAKAN NAÏVE BAYES,” *Thesis Fuadin, Didin Nizarul*, 2017.

[4] P. da Pedro Marques Luz, “Botnet Detection Using Passive DNS,” *Thesis Ru.Nl*, p. 41, 2014.

[5] A. García, S., Grill, M., Stiborek, J. and Zunino, “An empirical comparison of botnet detection methods,” *J. Comput. Secur.*, pp. 100–123, 2014.

[6] J. Nazario and T. Holz, “As the Net Churns : Fast-Flux Botnet Observations Tracking Fast-Flux Domains,” 2008.

[7] M. D. Data and S. Features, “Mining DNS-related Data for Suspicious Features Tilman Frosch,” 2011.

[8] T. Frosch, K. Marc, T. Holz, G. Horst, and R. Bochum, “Predentifier : Detecting Botnet C & C Domains From Passive DNS Data Motivation : DNS Features of Botnet Domains,” pp. 1–14.