DETEKSI FRAUD PADA AKUN WIFI UNIVERSITAS ISLAM INDONESIA DENGAN METODE K-MEANS

Disusun Oleh:

N a m a : Dio Agus Nofrizal

NIM : 17523110

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2021

HALAMAN PENGESAHAN DOSEN PEMBIMBING

DETEKSI FRAUD PADA AKUN WIFI UNIVERSITAS ISLAM INDONESIA DENGAN METODE K-MEANS

TUGAS AKHIR

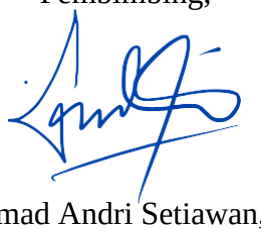
Disusun Oleh:

N a m a : Dio Agus Nofrizal

NIM : 17523110

Yogyakarta, 08 April 2021

Pembimbing,



(Dr. Mukhammad Andri Setiawan, S.T., M.Sc.)

HALAMAN PENGESAHAN DOSEN PENGUJI

DETEKSI FRAUD PADA AKUN WIFI UNIVERSITAS ISLAM INDONESIA DENGAN METODE K-MEANS

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 04 Mei 2021

Tim Penguji

Dr. Mukhammad Andri Setiawan, S.T., M.Sc.

Anggota 1

Lizda Iswari, S.T., M.Sc.

Anggota 2

Ahmad Fathan Hidayatullah, S.T., M.Sc.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Dio Agus Nofrizal

NIM : 17523110

Tugas akhir dengan judul:

DETEKSI FRAUD PADA AKUN WIFI UNIVERSITAS ISLAM INDONESIA DENGAN METODE K-MEANS

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 08 April 2021

(Dio Agus Nofrizal)

HALAMAN PERSEMBAHAN

Al-hamdu lillahi rabbil'alamin, puji syukur kepada Allah SWT yang telah melimpahkan rahmat dan karunia yang tidak terhingga. Shalawat dan salam senantiasa tercurah kepada Nabi Muhammad SAW yang menjadi cahaya bagi segala perbuatan mulia dan Insya Allah kita semua termasuk umat Nabi Muhammad SAW hingga akhir zaman.

Terima kasih yang tidak terhingga saya ucapkan kepada kedua orangtua saya, yang selalu memberi dukungan dan do'a, serta segala bentuk kasih sayang yang diberikan hingga saat ini.

Terima kasih saya ucapkan kepada dosen pembimbing saya, Bapak Dr. Mukhammad Andri Setiawan, S.T., M.Sc. yang telah meluangkan waktu, memberikan ilmu, dan motivasi selama membimbing saya.

Terima kasih kepada seluruh dosen Informatika UII yang sudah memberikan ilmu selama saya berkuliah di sini. Semoga segala dukungan yang telah diberikan menjadi amal jariah.

Terima kasih saya ucapkan kepada Shinta Dewi Kusumaningrum yang telah memberikan bantuan, waktu, dan dukungan selama penyusunan laporan ini. Teman-teman saya, Amiin Majiid Nugroho dan Dimastyo Muhaimin Arifin yang telah memberikan bantuan dan waktunya. Terima kasih juga saya ucapkan untuk seluruh teman-teman Informatika 2017 yang telah memberikan memori yang berkesan selama perkuliahan.

Terima kasih kepada seluruh pihak yang tidak bisa saya sebutkan satu persatu atas segala bentuk bantuan dan kebaikan yang telah diberikan.

HALAMAN MOTO

“Jika hidup tidak berjalan sesuai keinginanmu maka ingatlah bahwa Allah pasti punya jalan yang lebih baik untukmu”

“Masa depan adalah milik setiap orang yang menyiapkan hari ini”

“*I’d it this far and refused to give up because all my life I had always finished the race*” – Louis Zamperini

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh

Allhamdulillahirobbil'alamin, penulis memanjatkan puji syukur yang sebesar-besarnya kepada Allah SWT atas segala nikmat dan karunianya, sehingga penulis dapat menyelesaikan laporan tugas akhir yang berjudul "Deteksi Fraud Pada Akun Wifi Universitas Islam Indonesia Dengan Metode K-means" dengan lancar tanpa suatu hambatan yang berarti. Shalawat dan salam senantiasa tercurah kepada Nabi Muhammad SAW dan semoga kita semua mendapatkan syafa'atnya di akhir zaman.

Banyak pihak yang telah membantu dalam proses penyusunan laporan ini. Untuk itu, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua yang selalu memberikan dukungan dan do'a.
2. Bapak Dr. Mukhammad Andri Setiawan, S.T., M.Sc. selaku dosen pembimbing yang senantiasa memberikan waktu dan ilmunya.
3. Shinta Dewi Kusumaningrum yang telah memberikan bantuan selama proses penyusunan laporan.
4. Teman-teman informatika 2017.
5. Seluruh pihak yang telah membantu.

Penulis menyadari bahwa penelitian ini belum sempurna, untuk itu diharapkan penelitian ini bisa dikembangkan menjadi lebih baik lagi. Selain itu, penulis berharap penelitian ini dapat bermanfaat bagi penulis, BSI UII, maupun orang lain.

Yogyakarta, 08 April 2021

(Dio Agus Nofrizal)

SARI

Fraud menjadi sebuah masalah yang dapat merugikan orang lain sehingga harus dilakukan tindakan. *Fraud* terjadi ketika pengguna membagikan akunnya dengan orang lain untuk mengakses wifi UIIConnect tanpa memikirkan celah keamanan yang dapat membahayakan data pengguna. Terdapat dua faktor untuk mengetahui akun yang terindikasi melakukan *fraud*, yaitu faktor lokasi dan akses yang dilakukan ketika menggunakan wifi. Kedua faktor tersebut dapat diketahui dengan melihat *Acces Point* di mana pengguna tersebut terhubung dan akses apa yang dilakukan. Tujuan dari penelitian ini adalah membuat aplikasi yang dilengkapi dengan metode k-means untuk mengelompokkan akun-akun yang terindikasi melakukan *fraud* dan tidak sehingga dapat bermanfaat bagi Badan Sistem Informasi Universitas Islam Indonesia. Dalam proses mencapai tujuan dari penelitian, peneliti menggunakan enam tahap yaitu pengumpulan data, *pre-processing*, *dimension reduction*, *clustering*, implementasi dan pengujian. Selanjutnya akun yang terindikasi melakukan *fraud* akan dilakukan tindakan untuk keamanan data pengguna, sehingga pengguna harus berhati-hati dalam menggunakan akunnya.

Keywords—*fraud*, k-means, *principal component analysis*, *clustering*.

GLOSARIUM

Fraud	tindakan membagikan akun kepada orang lain.
PCA	metode yang digunakan untuk melakukan *dimension reduction*.
Clustering	penentuan kelompok berdasarkan kemiripan.
K-means	metode *clustering* data.
Elbow Method	metode penentuan jumlah *cluster* optimal.
Shiny App	*package* untuk implementasi *website* dalam R.
Data Internal	data yang dikumpulkan oleh BSI UII.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Universitas Islam Indonesia (UII) merupakan salah satu kampus yang memberikan fasilitas kepada mahasiswa, dosen, dan staf aktif di lingkungan UII dapat mengakses UIIConnect dengan menggunakan Akun UII masing-masing. Fasilitas ini didukung dengan pengembangan infrastruktur TI berskala *enterprise* yang diwujudkan dengan konsep UIIConnect yang sampai dengan hari ini telah memasang lebih dari 700 *Access Points* di seluruh gedung UII, dengan total *bandwidth* yang disediakan mencapai 3.7 Gbps, dengan akses setiap pengguna mencapai 125 Mbps (BSI UII, 2017). Adanya fasilitas tersebut beberapa pengguna biasanya membagikan akunnya baik dengan teman atau orang terdekatnya untuk mengakses UIIConnect. Kondisi ini sangat berbahaya karena dapat dideteksi sebagai *fraud*.

Fraud merupakan penipuan yang dilakukan secara sengaja dengan tujuan untuk mendapatkan keuntungan pribadi yang dapat menyebabkan kerugian bagi orang lain (Sorournejad, Zojaji, Atani, & Monadjemi, 2016). Orang yang melakukan kejahatan ini biasanya disebut *fraudster*. Ketika *fraudster* memiliki hak akses pada suatu *platform*, akan sangat mungkin orang tersebut dapat mengakses *platform* lainnya. Hal ini dikarenakan seseorang cenderung *login* di berbagai macam *platform* dengan akun yang sama. Selain itu, *fraudster* yang mendapatkan akses akun orang lain bisa saja akan terjadi penipuan yang berujung pada masalah finansial seperti menargetkan akun bank untuk transfer dana ke akun sendiri atau akun *eCommerce* dan melakukan pembelian palsu.

Banyak aktivitas di kampus UII yang menggunakan wifi UIIConnect untuk mengakses internet setiap harinya. Aktivitas seperti kegiatan belajar mengajar atau aktivitas lain yang dilakukan oleh staf, dosen maupun mahasiswa pasti membutuhkan akses internet dengan menggunakan wifi UII. Terlebih lagi UII memberikan masing-masing akun yang dapat terhubung ke UIIConnect hingga empat perangkat. Maka dari itu, sangat sulit mengidentifikasi akun yang melakukan *fraud* karena banyaknya pengguna yang terhubung dengan UIIConnect. Terdapat dua faktor untuk mengetahui akun yang terindikasi melakukan *fraud*, yaitu faktor lokasi dan akses yang dilakukan ketika menggunakan wifi. Kedua faktor tersebut dapat diketahui dengan melihat *Acces Point* di mana pengguna tersebut terhubung dan akses apa yang dilakukan. Dari permasalahan tersebut, dapat ditarik kesimpulan bahwa Badan Sistem

Informasi UII perlu memiliki suatu aplikasi untuk membantu dalam mengidentifikasi akun yang terdeteksi melakukan *fraud*. Oleh karena itu, akan dibuat aplikasi untuk mengetahui apakah akun yang menggunakan wifi UIIConnect melakukan *fraud* atau tidak.

Aplikasi dibuat menggunakan metode *k-means* untuk *clustering* yang bertujuan untuk mengelompokkan akun-akun yang terindikasi melakukan *fraud* dan tidak. *Clustering* adalah metode untuk menemukan pola kelompok, data yang memiliki kesamaan akan dikumpulkan menjadi satu kelompok, sedangkan untuk data yang memiliki perbedaan akan disatukan dalam kelompok yang berbeda (Sinaga & Yang, 2020). Aplikasi akan diimplementasi menggunakan Rstudio dengan bahasa pemrograman R dan *package* Shiny. Shiny adalah *open source R package* untuk membangun aplikasi website yang menyediakan kerangka kerja yang rapi. Shiny dapat membantu mengubah hasil analisis dari R menjadi aplikasi web yang interaktif dengan menggunakan HTML dan CSS sebagai pendukung (Beeley, 2013).

Diharapkan aplikasi yang dihasilkan dapat membantu Badan Sistem Informasi UII untuk mengetahui akun yang terindikasi melakukan *fraud* atau tidak. Selanjutnya Badan Sistem Informasi UII dapat melakukan tindakan terhadap akun tersebut berupa peringatan atau pemblokiran akun.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, rumusan masalah dalam penelitian ini adalah bagaimana mengembangkan aplikasi untuk melakukan pengelompokkan pengguna yang terindikasi melakukan *fraud* dan tidak?

1.3 Tujuan

Adapun tujuan dari penelitian ini adalah mengembangkan aplikasi untuk melakukan pengelompokkan pengguna yang terindikasi melakukan *fraud* dan tidak.

1.4 Manfaat

1.4.1 Manfaat bagi pengembang

a. Mendapatkan pengetahuan mengenai metode *k-means* serta penerapannya.
b. Mendapatkan pengetahuan mengenai pengembangan aplikasi menggunakan Shiny App.

1.4.2 Manfaat bagi Badan Sistem Informasi UII

Bagi Badan aplikasi informasi UII, hasil penelitian ini dapat membantu dalam mengidentifikasi akun yang terindikasi melakukan *fraud* dan tidak.

1.5 Batasan Masalah

Batasan masalah pada penelitian untuk deteksi *fraud* akun pada wifi UII ini mencakup beberapa hal berikut:

a. Aplikasi yang dihasilkan berupa hasil identifikasi akun terindikasi *fraud* dan tidak berdasarkan faktor lokasi pengguna dan akses yang dilakukan.
b. Data uji hanya berasal dari Badan Sistem Informasi UII.
c. Aplikasi menggunakan metode *k-means* yang diimplementasikan dalam bentuk *website* dengan menggunakan *framework* Shiny App.

1.6 Metode Penelitian

Metodologi dalam tugas akhir ini merupakan tahapan yang dilakukan agar penyusunan penelitian ini lebih terarah. Metodologi yang digunakan antara lain:

a. Pengumpulan Data

   Pengumpulan data dalam penelitian ini adalah data internal yang diambil dari *database* Badan Sistem Informasi UII. Data tersebut berupa file csv yang digunakan sebagai data untuk kebutuhan sistem.

b. *Pre-processing*

   Pre-processing dilakukan untuk mempersiapkan data dimulai dari data *reduction*, data *cleaning*, data *transformation*, dan data *integration.*

c. *Dimension Reduction*

   Dimension reduction dilakukan menggunakan *principal component analysis* yang bertujuan untuk meringankan proses komputasi dan meningkatkan performa *clustering*.

d. *Clustering*

   Setelah tahap *dimension reduction*, dilakukan proses *clustering* dengan metode *k-means*. Proses ini bertujuan untuk mengelompokkan akun-akun yang terindikasi *fraud* dan tidak.

e. Implementasi

Tahap selanjutnya adalah mengimplementasi pemodelan yang dibuat menggunakan *framework* Shiny App dari Rstudio.

f. Pengujian

Setelah aplikasi telah selesai dibuat, selanjutnya dilakukan pengujian menggunakan *black box*. Pengujian bertujuan untuk mengetahui kualitas dari aplikasi dan kesesuaian dengan tujuan.

1.7 Sistematika Penulisan

Penulisan dalam penelitian ini dibuat dengan terstruktur untuk mengetahui pembahasan-pembahasan yang ada pada setiap bab yang dibagi menjadi lima bab pembahasan. Berikut sistematika penulisan dalam laporan tugas akhir ini.

BAB I PENDAHULUAN

Bab ini berisi tentang pembahasan yang meliputi latar belakang penelitian, permasalahan yang diangkat, tujuan, manfaat, batasan masalah, metodologi penelitian, serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi teori, kajian pustaka, dan penelitian terkait yang diambil dari berbagai sumber.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang proses yang dilakukan dalam penelitian, mulai dari bagaimana mendapatkan data, *pre-processing*, *dimension reduction*, implementasi, hingga pengujian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi hasil dan pembahasan dari seluruh proses yang ada pada metodologi penelitian.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran berdasarkan penelitian yang telah dilakukan.

BAB II
LANDASAN TEORI

2.1 *Fraud*

Fraud merupakan penipuan yang dilakukan secara sengaja dengan tujuan untuk mendapatkan keuntungan pribadi yang dapat menyebabkan kerugian bagi orang lain (Zojaji, 2016). Ketika *fraudster* mendapatkan akses yang tidak diijinkan atau bahkan mengambil kendali akun milik orang lain, *fraudster* dapat melakukan penipuan, penyalahgunaan akun, pencurian identitas, dan lain-lain (Jialing Tao, 2018). *Fraud* juga dapat terjadi dalam berbagai bentuk yang dapat menghasilkan risiko yang berbeda. Misalnya orang yang menyalahgunakan sistem dengan membuat beberapa akun untuk memasukkan informasi palsu yang secara eksplisit dilarang dalam syarat dan ketentuan umum atau seseorang yang menyamar sebagai pengguna lain dengan akun yang dia dicuri (Ricardo, 2019).

Deteksi *fraud* mengacu pada proses menemukan keberadaan *fraud*. Deteksi *Fraud* dapat dilakukan melalui penggunaan kontrol internal yang dirancang dengan baik, pengawasan, dan pemantauan serta pencarian aktif untuk bukti potensi kecurangan. Investigasi *fraud* terjadi ketika indikator *fraud* menunjukkan bahwa telah terjadi tindakan penipuan dan memerlukan investigasi untuk menentukan sejauh mana kerugian dan identitas pelakunya (Riley, 2019).

2.2 *Principal Component Analysis*

Principal component analysis (PCA) berguna untuk mengurangi dimensi permasalahan menjadi lebih sederhana dengan cara mengidentifikasi sebagian kecil komponen utama dan secara efektif merangkum sebagian besar variasi data (Ait-Sahalia, Y., & Xiu, D, 2015). PCA menghitung kovarian matriks dari kumpulan data kemudian mencari *eigenvalue* dan *eigenvector*. Selanjutnya memilih beberapa *eigenvector* dari nilai *eigenvalue* yang lebih banyak untuk membentuk matriks transformasi sehingga dapat mengurangi dimensi kumpulan data (Ganesh R 2018). PCA mempertahankan *variance* sebanyak mungkin untuk menemukan variabel baru yang merupakan fungsi linier dari yang ada di kumpulan data asli. Variabel baru tersebut disebut *principal component* (PC) untuk menyelesaikan masalah *eigenvalue* dan *eigenvector* (M. Bi, 2016). Berikut adalah tahap-tahap yang dilakukan dalam algoritma PCA (Puteri Noraisya Primandari, 2018) antara lain:

a. Menghitung rata-rata semua sampel data dimana $x^{(i)}$ adalah sampel data dan m adalah kolom yang diperoleh dengan persamaan 2.1.

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} \quad (2.1)$$

b. Melakukan *adjusted* pada setiap data dengan persamaan 2.2.

$$x^{(i)} = x^{(i)} - \mu \quad (2.2)$$

c. Menghitung matriks kovarian (c) dimana $x^{(i)T}$ adalah *transpose* dari matriks $x^{(i)}$ yang diperoleh menggunakan persamaan 2.3.

$$C = \frac{1}{M} x^{(i)} x^{(i)T} \quad (2.3)$$

d. Menghitung *eigenvalue* dan *eigenvector* dengan menggunakan persamaan 2.4 dan 2.5.

$$C - \lambda I = 0 \quad (2.4)$$

$$(C - \lambda I)v = 0 \quad (2.5)$$

Keterangan:

C = Matriks kovarian λ = *eigenvalue*

I = Matriks identitas v = *eigenvector*

e. Menghitung nilai terbesar *eigenvalue* dan *eigenvector* yang berhubungan dan dipilih menjadi *Principal Component*. *Eigenvector* disusun dari yang terbesar ke yang terkecil seperti pada persamaan 2.6.

$$v = (eig1, eig2, eig3, \dots, eign) \quad (2.6)$$

f. Menghitung *Principal Component* dengan persamaan 2.7.

$$PC = x^{(i)}.v \quad (2.7)$$

g. Terakhir melakukan transformasi data untuk menghasilkan data PCA menggunakan persamaan 2.8.

$$PCA\ data = PC^{T}.x^{(i)T} \quad (2.8)$$

2.3 *Elbow Method*

Elbow method merupakan metode yang digunakan untuk menentukan jumlah *cluster* terbaik dengan cara melihat persentase perbandingan antara jumlah *cluster* yang akan membentuk siku pada suatu titik. Untuk mendapatkan perbandingan, dilakukan dengan cara menghitung SSE (*Sum of Square Error*) dari masing-masing nilai *cluster*. *Sum of Square Error* (SSE) pada persamaan 2.9 merupakan rumus untuk mengukur selisih data yang diperoleh dengan model prediksi yang sudah dilakukan sebelumnya (Nainggolan, Perangin-angin, Simarmata, & Tarigan, 2019).

$$SSE = \sum_{i=1}^{n} (d)^2 \quad (2.9)$$

Elbow method diterapkan dengan melihat grafik nilai k yang sesuai dengan posisi siku beserta SSE (*Sum of Square Error*) yang kurang dari satu. Hasil *cluster* k terbaik akan menjadi dasar untuk *clustering*. Semakin kecil nilai SSE dan semakin menurun grafik siku maka semakin baik hasil *cluster* (Syakur, Khotimah, Rochman, & Satoto, 2017).

2.4 *K-Means*

K-Means merupakan metode data *clustering* non-hirarki yang memisahkan data ke dalam satu atau lebih kelompok sehingga data akan dikumpulkan sesuai dengan karakteristik masing-masing (Darmi & Setiawan, 2016). Metode *K-Means* hanya bisa mengolah data dalam bentuk angka saja, maka dari itu untuk data yang berbentuk nominal harus diinisialisasikan ke dalam bentuk angka terlebih dahulu agar dapat terbaca (Sibuea, Sapta, Informasi, & Royal, 2017). Langkah-langkah yang dilakukan pada proses *clustering* dengan menggunakan *K-Means* (Syakur et al., 2017) antara lain:

1. Menentukan jumlah *cluster* K dan jumlah iterasi maksimum.
2. Melakukan proses inisialisasi titik tengah *cluster* K, kemudian melakukan *centroid count* fitur dengan persamaan 2.10.

   $$C_i = \frac{1}{M} \sum_{j=1}^{M} x_j \quad (2.10)$$

   Persamaan 2.10 dilakukan sebanyak p dimensi dari $i = 1$ sampai $i = p$
3. Menghubungkan data observasi ke *cluster* terdekat. Pengukuran jarak *Euclidean* dapat ditemukan menggunakan persamaan 2.11.

   $$d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \quad (2.11)$$

4. Melakukan realokasi data ke masing-masing kelompok berdasarkan perbandingan jarak antar data dengan masing-masing *centroid* grup menggunakan 2.12.

   $$a_{ij} \begin{cases} 1 & d = \min \{D(x_i, c_i)\} \\ 0 & \end{cases} \quad (2.12)$$

5. Menghitung ulang posisi titik tengah *cluster* yang merupakan nilai keanggotaan poin x_i ke pusat grup c_1, d adalah yang terpendek jarak dari data x_i ke grup K setelah dibandingkan, dan $c_1$adalah pusat grup ke 1. *The objective function* yang digunakan dalam metode ini didasarkan pada jarak dan nilai dari keanggotaan data dalam grup. *The objective function* menurut MacQueen (1967) bisa ditentukan menggunakan persamaan 2.13.

   $$J = \sum_{i=1}^{n} \sum_{l=1}^{k} a_{ic} D(x_i, c_i)^2 \quad (2.13)$$

n adalah jumlah data, k adalah jumlah kelompok, a_{i1} adalah nilai keanggotaan data titik x_i ke grup c_1 diikuti a memiliki nilai 0 atau 1. Jika data adalah anggota grup maka nilai $a_{i1} = 1$ dan jika tidak, nilai $a_{i1} = 0$.

6. Jika terjadi perubahan posisi titik tengah *cluster* atau jumlah iterasi kurang dari maksimum jumlah iterasi, kembali ke langkah 3. Jika tidak, hasil *clustering* dikembalikan.

2.5 Shiny App

Shiny App merupakan R *package* untuk mempermudah menyampaikan data kepada pengguna melalui *website*. Shiny hadir dengan berbagai macam *widget* untuk membangun antarmuka pengguna dengan cepat dan interaktif. Aplikasi Shiny memiliki gaya *default* yang rapi dan efektif sehingga Shiny sangat mudah untuk dikembangkan dan diintegrasikan dengan konten web menggunakan HTML dan CSS. JavaScript dan jQuery juga dapat digunakan untuk memperluas cakupan aplikasi Shiny (Beeley, 2013).

2.6 *Black Box Testing*

Dalam *black box testing*, pengujian dirancang dengan melakukan pemeriksaan nilai *input* atau *output* saja tanpa memiliki pengetahuan mengenai desain ataupun kode yg digunakan (MAll, 2014). *Black box testing* bertujuan untuk mengetahui spesifikasi *software* dari segi tampilan maupun fungsi-fungsi yang ada pada *software*. Menurut (Jacob & Prasanna, 2016) terdapat dua jenis *black box testing* antara lain:

a. *Equivalence Class Partitioning* (ECP)

   Dalam teknik *equivalence class partitioning*, seluruh rangkaian input pada *software* dibagi menjadi berbagai kelas. Setiap kelas terdapat *input condition specification* yang memiliki nilai valid atau tidak valid. Jika satu kondisi pada *input condition specification* bernilai valid, maka semua kondisi pada kelas itu akan berfungsi, dan begitu juga sebaliknya.

b. *Boundary Value Analysis* (BVA)

   Boundary value analysis merupakan teknik pengujian yang dilakukan terhadap batasan pada berbagai sub kelas. Jika nilai input sesuai dengan *range* yang ditetapkan maka nilai tersebut valid dan sebaliknya jika tidak sesuai dengan *range* yang ditetapkan maka nilai tersebut invalid.

2.7 Penelitian Terkait

Penelitian yang dilakukan sebelumnya memiliki kasus dan cara yang berbeda dalam mendeteksi *fraud*. Berikut penelitian terkait yang ditampilkan pada Tabel 2.1.

Tabel 2.1 Tabel Penelitian Terkait

No	Penelitian	Metode	Hasil
1	Towards Detecting Anomalous User Behavior in Online Social Networks (Viswanath et al., 2014)	Principal Component Analisys (PCA)	Deteksi dilakukan dengan metode *unsupervised learning* yaitu *Principal Component Analisys* (PCA) yang digunakan untuk membedakan perilaku pengguna normal dan tidak normal. PCA memodelkan perilaku pengguna normal secara akurat dan mengidentifikasi anomali secara signifikan. Hasil evaluasi pendekatan yang dilakukan mencapai tingkat deteksi lebih dari 66% dan mencakup lebih dari 94% perilaku buruk dengan positive false kurang dari 0,3% (Viswanath et al., 2014).
2	Anomaly detection model of user behavior based on principal component analysis (Bi, Xu, Wang, & Zhou, 2016)	Principal Component Analysis (PCA)	Penelitian tentang anomaly detection menggunakan metode PCA yang menghasilkan secara akurat dapat menggambarkan perilaku pengguna normal dan abnormal serta dapat meningkatkan efisiensi dan stabilitas (Bi, Xu, Wang, & Zhou, 2016).

3	Using Machine Learning to Detect Anomalies in Internet Browsing Pattern of Users (Paul & Medhe, 2019)	Perbandingan metode Gaussian Mixture Model (GMM), K-means dan Bayesian Gaussian Mixture Model (BGMM).	Penelitian ini menggunakan data jaringan aktivitas pengguna pada organisasi dan perusahaan yang disimpan sebagai log. Log ini akan digunakan sebagai fitur untuk melatih model dalam melakukan pengelompokan. GMM menghasilkan *false positive* paling sedikit sebesar 0.33% sedangkan *K-means* 21.77% dan BGMM 5.67% (Paul & Medhe, 2019).

Terdapat beberapa perbedaan dari penelitian-penelitian sebelumnya. Penelitian ini membahas mengenai anomaly pada penggunaan akun wifi UIIConnect atau terindikasi melakukan *fraud*. *Fraud* dapat diketahui berdasarkan faktor lokasi dan akses yang dilakukan pengguna. Aplikasi akan dilengkapi dengan metode *k-means* untuk melakukan pengelompokan akun-akun yang terindikasi melakukan *fraud* dan tidak.

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Data yang digunakan dalam penelitian ini merupakan data sekunder. Data sekunder adalah data yang sudah dikumpulkan sebelumnya oleh pihak lain, bukan didapatkan secara langsung dari objek penelitiannya. Data sekunder bisa didapatkan secara mudah dan cepat karena data yang sudah tersedia tersebut tinggal diambil dan dipakai sesuai dengan kebutuhan (Hasibuan, 2007). Dalam penelitian ini, data sekunder yang digunakan merupakan data internal yang dikumpulkan oleh Badan Sistem Informasi UII.

3.2 Studi Literatur

Studi literatur berisi tentang teori-teori dan bahan penelitian yang digunakan sebagai acuan dan landasan dalam penelitian. Studi literatur juga bertujuan untuk membantu peneliti dalam mencari tujuan dan mengetahui proses penelitian (Hasibuan, 2007). Informasi yang didapat dari tahap ini dapat dijadikan rujukan untuk memperkuat argumentasi-argumentasi yang ada dan sebagai pengetahuan yang diperlukan untuk mengembangkan aplikasi.

3.3 Indikator Keberhasilan

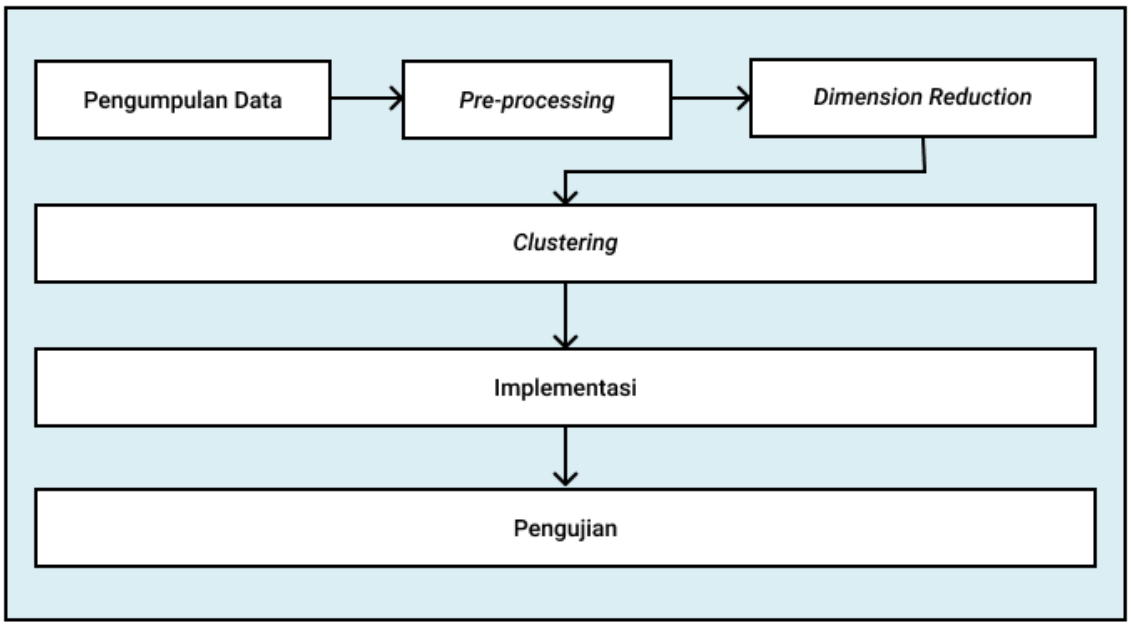
Indikator keberhasilan dari penelitian ini antara lain:

a. Aplikasi dapat memproses data yang diunggah.
b. Aplikasi dapat memberikan *output* berupa akun-akun yang terindikasi melakukan *fraud* dan tidak.

3.4 Metode Penelitian

Penelitian ini dilakukan untuk mencapai tujuan melalui proses yang terdapat pada Gambar 3.1. Penelitian dimulai dengan pengumpulan data yang sesuai dengan kebutuhan. Setelah itu, penelitian dilanjutkan dengan *pre-processing* yang bertujuan untuk mempersiapkan data dan dilanjutkan proses *dimension reduction* untuk mengoptimalkan proses *clustering*. Kemudian dilanjutkan dengan *clustering* yang bertujuan untuk mengelompokkan akun-akun yang terindikasi melakukan *fraud* atau tidak. Setelah selesai melakukan *clustering*, hasil tersebut akan diimplementasikan ke dalam aplikasi menggunakan Shiny App. Terakhir,

aplikasi akan diuji menggunakan *black box testing* untuk mengetahui kualitas dari aplikasi dan kesesuaian dengan tujuan.



Gambar 3.1 Metode Penelitian

3.4.1 Pengumpulan Data

Pertama, dilakukan pengumpulan data internal yang diambil dari *database* Badan Sistem Informasi UII. Data internal adalah data pribadi milik suatu organisasi yang memberikan gambaran mengenai situasi dan kondisi organisasi tersebut (Hasibuan, 2007). Data internal Badan Sistem Informasi UII berupa file csv yang digunakan sebagai data untuk kebutuhan sistem.

3.4.2 *Pre-Processing*

Tahap *pre-processing* digunakan untuk mempersiapkan data sebelum dilakukan tahap selanjutnya. Terdapat empat teknik *pre-processing* data yaitu data *cleaning*, data *integration*, data *reduction*, dan data *transformation*. Data *cleaning* digunakan untuk memperbaiki data yang inkonsisten dan menghilangkan *noise*. Data *integration* digunakan untuk menggabungkan data dari berbagai macam sumber. Data *reduction* digunakan untuk mengurangi ukuran data dengan cara menggabungkan, menghilangkan, atau mengelompokkan fitur. Data *transformation* bertujuan untuk meningkatkan akurasi dan efisiensi algoritma salah satunya dengan cara membuat data menjadi kisaran skala yang lebih kecil (Han, Kamber, & Pei, 2012).

Teknik *pre-processing* dengan data *reduction*, data *cleaning*, data *transformation*, dan data *integration* dilakukan dengan cara sebagai berikut:

a. Data *Reduction*

Pada tahap data *reduction* ini dilakukan pengambilan beberapa variabel yang akan digunakan untuk penelitian. Variabel yang digunakan pada masing-masing data akan mewakili faktor lokasi dan akses yang dilakukan pengguna ketika menggunakan wifi UIIConnect.

b. Data *Cleaning*

1. Mengganti format waktu

   Proses mengganti format waktu dilakukan dengan tujuan untuk menyingkronkan waktu pada data URL dan data *access point*. Hal ini dikarenakan kedua data memiliki format yang berbeda dalam melakukan *capture* waktu pada saat menggunakan wifi UIIConnect.

2. Menghapus karakter domain email

   Pada data URL dan data *access point* terdapat perbedaan format dalam melakukan *capture* nama akun pada saat menggunakan wifi UIIConnect. Untuk itu diperlukan proses untuk menyamakan format nama akun yaitu dengan cara menghapus karakter domain email.

3. Menghapus nilai *NA*

   Pada proses ini, dilakukan dengan mengelompokan aplikasi yang digunakan oleh masing-masing akun. Setelah itu, menghapus seluruh baris yang bernilai *NA*. Hal ini dikarenakan baris tersebut tidak dapat digunakan dalam penelitian.

c. Data *Transformation*

1. Memberikan label kemiripan aplikasi

   Setelah melakukan pengelompokkan aplikasi, selanjutnya diperlukan adanya pemberian label. Hal ini dilakukan untuk membedakan kemiripan aplikasi yang digunakan oleh masing-masing akun.

2. Menghitung persentase kemiripan aplikasi

   Selanjutnya persentase label pada setiap akun akan dihitung. Tahap ini dilakukan untuk mengetahui persentase kemiripan aplikasi yang digunakan untuk setiap IP yang menggunakan akun yang sama.

3. Menghitung jumlah IP

Pada tahap ini, dilakukan perhitungan jumlah IP yang digunakan oleh setiap akun. Berdasarkan jumlah IP ini, dapat diketahui jumlah perangkat yang digunakan pengguna ketika menggunakan wifi UIIConnect.

4. Menghitung jumlah *Access Point*

Selanjutnya, dilakukan perhitungan jumlah *Acces Point* yang digunakan oleh setiap pengguna. Jumlah *access point* tersebut dapat digunakan untuk mengetahui apakah pengguna berada di lokasi yang sama atau tidak.

d. Data *Integration*

Tahap terakhir dalam *pre-processing* adalah data *intergration* dengan melakukan *merge* data URL dan data *access point*. Proses ini dilakukan untuk menggabungkan kedua data tersebut berdasarkan nama akun pengguna pada masing-masing data.

3.4.3 *Dimension Reduction*

Dimension reduction merupakan proses pengurangan jumlah variabel acak atau atribut yang dipertimbangkan dengan cara mengubah atau memproyeksikan data asli menjadi lebih kecil (Han et al., 2012). Tahap ini menjabarkan tentang proses pengurangan dimensi variabel menggunakan metode *principal component analisys* yang bertujuan untuk mengoptimalkan proses *clustering*. Berikut adalah tahap-tahap yang dilakukan pada metode PCA:

a. Standardisasi Data

Standardisasi data bertujuan agar setiap variabel memiliki kontribusi yang sama. Berikut adalah hal-hal yang dilakukan pada standarisasi data:

1) Menghitung rata-rata menggunakan persamaan 3.1.

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} \quad (3.1)$$

2) Menghitung *center* dengan persamaan 3.2.

$$x^{(i)} = x^{(i)} - \mu \quad (3.2)$$

3) Menghitung kovarian matrix yang merupakan matrix M x M menggunakan persamaan 3.3.

$$C = \frac{1}{M} X' X'^{T} \quad (3.3)$$

b. Menghitung *Variance*

Variance meruapakan sebaran data yang ditangkap oleh masing-masing *principal component*. Berikut persamaan 3.4 untuk menghitung *variance*:

$$\frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)\,T}\,u\right)^2 = \frac{1}{m}\sum_{i=1}^{m} u^{T} x^{(i)} x^{(i)\,T}\mathrm{u} \qquad (3.4)$$

$$= u^{T}\left(\sum_{i=1}^{m} x^{(i)} x^{(i)\,T}\right)u$$

3.4.4 *Clustering*

Tahap ini menjabarkan tentang proses *clustering* pengguna dengan menggunakan metode *k-means*. Proses ini bertujuan untuk mengelompokkan akun-akun yang terindikasi melakukan *fraud* dan tidak. Sebelum melakukan proses *clustering*, dilakukan penentuan jumlah *cluster* dengan menggunakan *elbow method*.

3.4.5 Implementasi

Implementasi merupakan tahap pengembangan aplikasi berdasarkan pemodelan yang dilakukan. Implementasi menggunakan *framework* Shiny App dari Rstudio agar memudahkan untuk menerapkan hasil dari pemodelan yang telah dibuat sebelumnya.

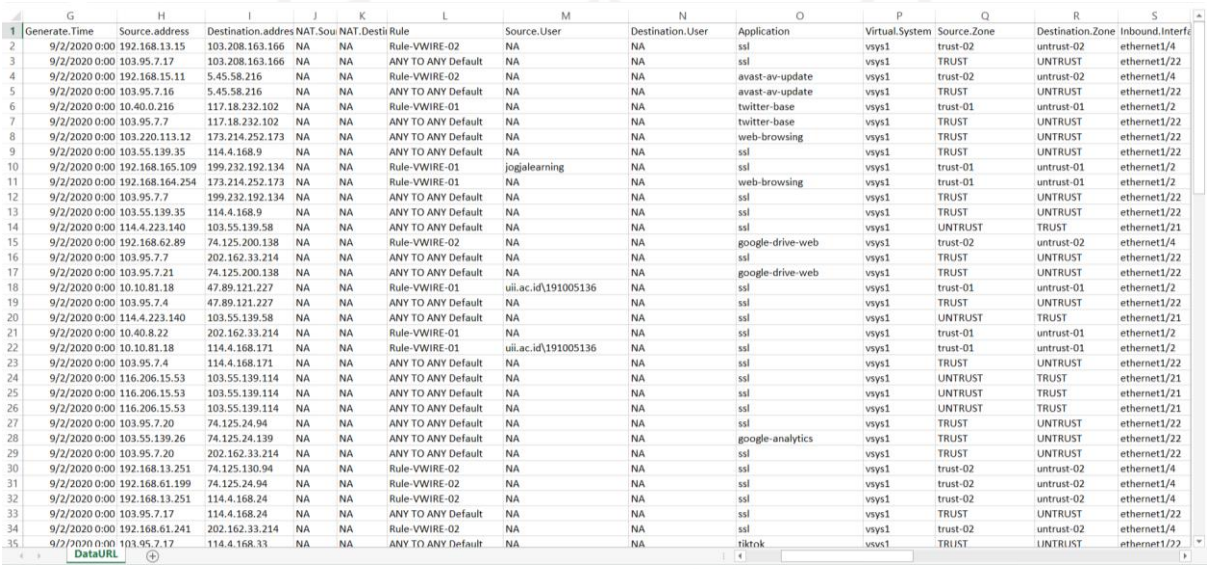
3.4.6 Pengujian

Pengujian merupakan tahap untuk mengetahui kualitas dan kinerja dari sistem apakah berfungsi sesuai dengan tujuan atau tidak. Pengujian dalam aplikasi ini menggunakan metode *black box testing* dengan teknik *equivalence class partitioning*. Teknik ini dilakukan dengan menggunakan skenario pengujian dan *test case* untuk memastikan *output* sesuai dengan yang diinginkan.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

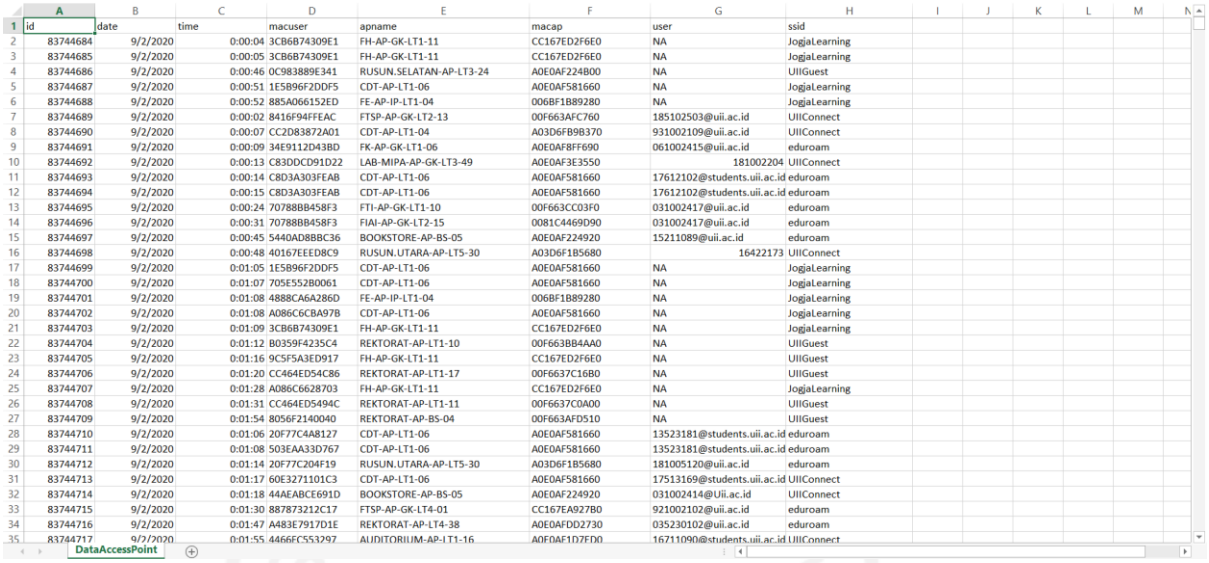
Data yang digunakan dalam penelitian ini berupa hasil *record* setiap akun yang menggunakan wifi UII yang telah dikumpulkan oleh Badan Sistem Informasi UII. Terdapat dua data yang digunakan pada penelitian yaitu data URL dan data *access point*. Gambar 4.1 memuat data URL yang berisi variabel "Receive.Time", "Source.user", "Source.address", "Application" dan lain-lain. Gambar 4.2 memuat data *access point* yang berisi "id", "date", "time", "macuser" dan lain-lain.



	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Generate.Time	Source.address	Destination.addres	NAT.Sou	NAT.Destii	Rule	Source.User	Destination.User	Application	Virtual.System	Source.Zone	Destination.Zone	Inbound.Interfa
2	9/2/2020 0:00	192.168.13.15	103.208.163.166	NA	NA	Rule-VWIRE-02	NA	NA	ssl	vsys1	trust-02	untrust-02	ethernet1/4
3	9/2/2020 0:00	103.95.7.17	103.208.163.166	NA	NA	ANY TO ANY Default	NA	NA	ssl	vsys1	TRUST	UNTRUST	ethernet1/22
4	9/2/2020 0:00	192.168.15.11	5.45.58.216	NA	NA	Rule-VWIRE-02	NA	NA	avast-av-update	vsys1	trust-02	untrust-02	ethernet1/4
5	9/2/2020 0:00	103.95.7.16	5.45.58.216	NA	NA	ANY TO ANY Default	NA	NA	avast-av-update	vsys1	TRUST	UNTRUST	ethernet1/22
6	9/2/2020 0:00	10.40.0.216	117.18.232.102	NA	NA	Rule-VWIRE-01	NA	NA	twitter-base	vsys1	trust-01	untrust-01	ethernet1/2
7	9/2/2020 0:00	103.95.7.7	117.18.232.102	NA	NA	ANY TO ANY Default	NA	NA	twitter-base	vsys1	TRUST	UNTRUST	ethernet1/22
8	9/2/2020 0:00	103.220.113.12	173.214.252.173	NA	NA	ANY TO ANY Default	NA	NA	web-browsing	vsys1	TRUST	UNTRUST	ethernet1/22
9	9/2/2020 0:00	103.55.139.35	114.4.168.9	NA	NA	ANY TO ANY Default	NA	NA	ssl	vsys1	TRUST	UNTRUST	ethernet1/22
10	9/2/2020 0:00	192.168.165.109	199.232.192.134	NA	NA	Rule-VWIRE-01	jogjalearning	NA	ssl	vsys1	trust-01	untrust-01	ethernet1/2
11	9/2/2020 0:00	192.168.164.254	173.214.252.173	NA	NA	Rule-VWIRE-01	NA	NA	web-browsing	vsys1	trust-01	untrust-01	ethernet1/2
12	9/2/2020 0:00	103.95.7.7	199.232.192.134	NA	NA	ANY TO ANY Default	NA	NA	ssl	vsys1	TRUST	UNTRUST	ethernet1/22
13	9/2/2020 0:00	103.55.139.35	114.4.168.9	NA	NA	ANY TO ANY Default	NA	NA	ssl	vsys1	TRUST	UNTRUST	ethernet1/22
14	9/2/2020 0:00	114.4.223.140	103.55.139.58	NA	NA	ANY TO ANY Default	NA	NA	ssl	vsys1	UNTRUST	TRUST	ethernet1/21
15	9/2/2020 0:00	192.168.62.89	74.125.200.138	NA	NA	Rule-VWIRE-02	NA	NA	google-drive-web	vsys1	trust-02	untrust-02	ethernet1/4
16	9/2/2020 0:00	103.95.7.7	202.162.33.214	NA	NA	ANY TO ANY Default	NA	NA	ssl	vsys1	TRUST	UNTRUST	ethernet1/22
17	9/2/2020 0:00	103.95.7.21	74.125.200.138	NA	NA	ANY TO ANY Default	NA	NA	google-drive-web	vsys1	TRUST	UNTRUST	ethernet1/22
18	9/2/2020 0:00	10.10.81.18	47.89.121.227	NA	NA	Rule-VWIRE-01	uii.ac.id\191005136	NA	ssl	vsys1	trust-01	untrust-01	ethernet1/2
19	9/2/2020 0:00	103.95.7.4	47.89.121.227	NA	NA	ANY TO ANY Default	NA	NA	ssl	vsys1	TRUST	UNTRUST	ethernet1/22
20	9/2/2020 0:00	114.4.223.140	103.55.139.58	NA	NA	ANY TO ANY Default	NA	NA	ssl	vsys1	UNTRUST	TRUST	ethernet1/21
21	9/2/2020 0:00	10.40.8.22	202.162.33.214	NA	NA	Rule-VWIRE-01	NA	NA	ssl	vsys1	trust-01	untrust-01	ethernet1/2
22	9/2/2020 0:00	10.10.81.18	114.4.168.171	NA	NA	Rule-VWIRE-01	uii.ac.id\191005136	NA	ssl	vsys1	trust-01	untrust-01	ethernet1/2
23	9/2/2020 0:00	103.95.7.4	114.4.168.171	NA	NA	ANY TO ANY Default	NA	NA	ssl	vsys1	TRUST	UNTRUST	ethernet1/22
24	9/2/2020 0:00	116.206.15.53	103.55.139.114	NA	NA	ANY TO ANY Default	NA	NA	ssl	vsys1	UNTRUST	TRUST	ethernet1/21
25	9/2/2020 0:00	116.206.15.53	103.55.139.114	NA	NA	ANY TO ANY Default	NA	NA	ssl	vsys1	UNTRUST	TRUST	ethernet1/21
26	9/2/2020 0:00	116.206.15.53	103.55.139.114	NA	NA	ANY TO ANY Default	NA	NA	ssl	vsys1	UNTRUST	TRUST	ethernet1/21
27	9/2/2020 0:00	103.95.7.20	74.125.24.94	NA	NA	ANY TO ANY Default	NA	NA	ssl	vsys1	TRUST	UNTRUST	ethernet1/22
28	9/2/2020 0:00	103.55.139.26	74.125.24.139	NA	NA	ANY TO ANY Default	NA	NA	google-analytics	vsys1	TRUST	UNTRUST	ethernet1/22
29	9/2/2020 0:00	103.95.7.20	202.162.33.214	NA	NA	ANY TO ANY Default	NA	NA	ssl	vsys1	TRUST	UNTRUST	ethernet1/22
30	9/2/2020 0:00	192.168.13.251	74.125.130.94	NA	NA	Rule-VWIRE-02	NA	NA	ssl	vsys1	trust-02	untrust-02	ethernet1/4
31	9/2/2020 0:00	192.168.61.199	74.125.24.94	NA	NA	Rule-VWIRE-02	NA	NA	ssl	vsys1	trust-02	untrust-02	ethernet1/4
32	9/2/2020 0:00	192.168.13.251	114.4.168.24	NA	NA	Rule-VWIRE-02	NA	NA	ssl	vsys1	trust-02	untrust-02	ethernet1/4
33	9/2/2020 0:00	103.95.7.17	114.4.168.24	NA	NA	ANY TO ANY Default	NA	NA	ssl	vsys1	TRUST	UNTRUST	ethernet1/22
34	9/2/2020 0:00	192.168.61.241	202.162.33.214	NA	NA	Rule-VWIRE-02	NA	NA	ssl	vsys1	trust-02	untrust-02	ethernet1/4
35	9/2/2020 0:00	103.95.7.17	114.4.168.33	NA	NA	ANY TO ANY Default	NA	NA	tiktok	vsys1	TRUST	UNTRUST	ethernet1/22

DataURL

Gambar 4.1 Data URL



id	date	time	macuser	apname	macap	user	ssid
83744684	9/2/2020	0:00:04	3CB6B74309E1	FH-AP-GK-LT1-11	CC167ED2F6E0	NA	JogjaLearning
83744685	9/2/2020	0:00:05	3CB6B74309E1	FH-AP-GK-LT1-11	CC167ED2F6E0	NA	JogjaLearning
83744686	9/2/2020	0:00:46	0C9833889E341	RUSUN.SELATAN-AP-LT3-24	A0E0AF224B00	NA	UIIGuest
83744687	9/2/2020	0:00:51	1E5B96F2DDF5	CDT-AP-LT1-06	A0E0AF581660	NA	JogjaLearning
83744688	9/2/2020	0:00:52	885A066152ED	FE-AP-IP-LT1-04	006BF1B89280	NA	JogjaLearning
83744689	9/2/2020	0:00:02	8416F94FFEAC	FTSP-AP-GK-LT2-13	00F663AFC760	185102503@uii.ac.id	UIIConnect
83744690	9/2/2020	0:00:07	CC2D83872A01	CDT-AP-LT1-04	A03D6FB9B370	931002109@uii.ac.id	UIIConnect
83744691	9/2/2020	0:00:09	34E9112D43BD	FK-AP-GK-LT1-06	A0E0AF8FF690	061002415@uii.ac.id	eduroam
83744692	9/2/2020	0:00:13	C83DDCD91D22	LAB-MIPA-AP-GK-LT3-49	A0E0AF3E3550	181002204	UIIConnect
83744693	9/2/2020	0:00:14	C8D3A303FEAB	CDT-AP-LT1-06	A0E0AF581660	17612102@students.uii.ac.id	eduroam
83744694	9/2/2020	0:00:15	C8D3A303FEAB	CDT-AP-LT1-06	A0E0AF581660	17612102@students.uii.ac.id	eduroam
83744695	9/2/2020	0:00:24	70788BB458F3	FTI-AP-GK-LT1-10	00F663CC03F0	031002417@uii.ac.id	eduroam
83744696	9/2/2020	0:00:31	70788BB458F3	FIAI-AP-GK-LT2-15	0081C4469D90	031002417@uii.ac.id	eduroam
83744697	9/2/2020	0:00:45	5440AD8BBC36	BOOKSTORE-AP-BS-05	A0E0AF224920	15211089@uii.ac.id	eduroam
83744698	9/2/2020	0:00:48	40167EEED8C9	RUSUN.UTARA-AP-LT5-30	A03D6F1B5680	16422173	UIIConnect
83744699	9/2/2020	0:01:05	1E5B96F2DDF5	CDT-AP-LT1-06	A0E0AF581660	NA	JogjaLearning
83744700	9/2/2020	0:01:07	705E552B0061	CDT-AP-LT1-06	A0E0AF581660	NA	JogjaLearning
83744701	9/2/2020	0:01:08	4888CA6A286D	FE-AP-IP-LT1-04	006BF1B89280	NA	JogjaLearning
83744702	9/2/2020	0:01:08	A086C6CBA97B	CDT-AP-LT1-06	A0E0AF581660	NA	JogjaLearning
83744703	9/2/2020	0:01:09	3CB6B74309E1	FH-AP-GK-LT1-11	CC167ED2F6E0	NA	JogjaLearning
83744704	9/2/2020	0:01:12	B0359F4235C4	REKTORAT-AP-LT1-10	00F663BB4AA0	NA	UIIGuest
83744705	9/2/2020	0:01:16	9C5F5A3ED917	FH-AP-GK-LT1-11	CC167ED2F6E0	NA	UIIGuest
83744706	9/2/2020	0:01:20	CC464ED54C86	REKTORAT-AP-LT1-17	00F6637C16B0	NA	UIIGuest
83744707	9/2/2020	0:01:28	A086C6628703	FH-AP-GK-LT1-11	CC167ED2F6E0	NA	JogjaLearning
83744708	9/2/2020	0:01:31	CC464ED5494C	REKTORAT-AP-LT1-11	00F6637C0A00	NA	UIIGuest
83744709	9/2/2020	0:01:54	8056F2140040	REKTORAT-AP-BS-04	00F663AFD510	NA	UIIGuest
83744710	9/2/2020	0:01:06	20F77C4A8127	CDT-AP-LT1-06	A0E0AF581660	13523181@students.uii.ac.id	eduroam
83744711	9/2/2020	0:01:08	503EAA33D767	CDT-AP-LT1-06	A0E0AF581660	13523181@students.uii.ac.id	eduroam
83744712	9/2/2020	0:01:14	20F77C204F19	RUSUN.UTARA-AP-LT5-30	A03D6F1B5680	181005120@uii.ac.id	eduroam
83744713	9/2/2020	0:01:17	60E3271101C3	CDT-AP-LT1-06	A0E0AF581660	17513169@students.uii.ac.id	UIIConnect
83744714	9/2/2020	0:01:18	44AEABCE691D	BOOKSTORE-AP-BS-05	A0E0AF224920	031002414@Uii.ac.id	UIIConnect
83744715	9/2/2020	0:01:30	887873212C17	FTSP-AP-GK-LT4-01	CC167EA927B0	921002102@uii.ac.id	eduroam
83744716	9/2/2020	0:01:47	A483E7917D1E	REKTORAT-AP-LT4-38	A0E0AFDD2730	035230102@uii.ac.id	eduroam
83744717	9/2/2020	0:01:55	4466FC553297	AUDITORIUM-AP-LT1-16	A0E0AF1D7ED0	16711090@students.uii.ac.id	UIIConnect

DataAccessPoint

Gambar 4.2 Data *Acess Point*

Data yang digunakan merupakan *file* dengan format csv. *File* tersebut kemudian dibaca dengan kode program yang bisa dilihat pada Gambar 4.3. Kode program tersebut memanggil data URL dan data *access point* pada lokasi *file* csv berada.

```
fraudData <- read.csv(file="D:/Skripsi/R/PAN-PA-5050-BSI-
UII_url_2020_09_03_last_calendar_day.csv",nrows=800000,na.strings=c("","
NA"))
test_extraction <- read.csv(D:/Skripsi/R/DataAccessPoint,
na.strings=c("","NA"))
```

Gambar 4.3 *Code* Membaca Data URL dan Data *Access Point*

4.2 *Pre-Processing*

4.2.1 Data *Reduction*

Data *reduction* dilakukan dengan mengambil beberapa variabel dari masing-masing data yang digunakan untuk penelitian. Pengambilan variabel dilakukan berdasarkan dua faktor yaitu faktor lokasi dan akses yang dilakukan ketika pengguna menggunakan wifi UIICOnnect. Faktor akses dapat diketahui pada data URL dengan mengambil variabel “Application” yang berisi aplikasi apa saja yang diakses oleh pengguna. Selain itu, diambil juga variabel lain seperti “Receive.Time”, “Source.User” untuk mengetahui akun yang digunakan, dan “Source.address” yang berisi alamat IP perangkat yang digunakan. Gambar 4.4 merupakan *code* untuk melakukan pengambilan variabel data URL, hasilnya ditunjukkan pada Gambar 4.5.

```
analisisData <- fraudData[c(2,13,8,15)]
```

Gambar 4.4 *Code* Pengambilan Variabel Pada Data Url

	Receive.Time	Source.User	Source.address	Application
1	2020/09/02 00:00:00	NA	192.168.13.15	ssl
2	2020/09/02 00:00:00	NA	103.95.7.17	ssl
3	2020/09/02 00:00:00	NA	192.168.15.11	avast-av-update
4	2020/09/02 00:00:00	NA	103.95.7.16	avast-av-update
5	2020/09/02 00:00:00	NA	10.40.0.216	twitter-base
6	2020/09/02 00:00:00	NA	103.95.7.7	twitter-base
7	2020/09/02 00:00:00	NA	103.220.113.12	web-browsing
8	2020/09/02 00:00:00	NA	103.55.139.35	ssl
9	2020/09/02 00:00:00	jogjalearning	192.168.165.109	ssl
10	2020/09/02 00:00:00	NA	192.168.164.254	web-browsing
11	2020/09/02 00:00:00	NA	103.95.7.7	ssl
12	2020/09/02 00:00:00	NA	103.55.139.35	ssl
13	2020/09/02 00:00:00	NA	114.4.223.140	ssl
14	2020/09/02 00:00:00	NA	192.168.62.89	google-drive-web
15	2020/09/02 00:00:00	NA	103.95.7.7	ssl
16	2020/09/02 00:00:00	NA	103.95.7.21	google-drive-web
17	2020/09/02 00:00:00	uii.ac.id\191005136	10.10.81.18	ssl
18	2020/09/02 00:00:00	NA	103.95.7.4	ssl
19	2020/09/02 00:00:00	NA	114.4.223.140	ssl
20	2020/09/02 00:00:00	NA	10.40.8.22	ssl

Gambar 4.5 Hasil Pengambilan Variabel Pada Data Url

Selanjutnya, untuk faktor lokasi dapat diketahui dari data *access point* dengan mengambil variabel "apname" yang berisi nama *access point* ketika pengguna menggunakan wifi UIIConnect. Selain itu, digunakan juga variabel "time" dan "user". Gambar 4.6 merupakan *code* untuk pengambilan variabel data *access point* beserta dengan hasilnya yang ditunjukkan pada Gambar 4.7.

```
test_extraction <- test_extraction[c(3,7,5)]
```

Gambar 4.6 *Code* Pengambilan Variabel Pada Data *Access Point*

	time	user	apname
1	00:00:04	NA	FH-AP-GK-LT1-11
2	00:00:05	NA	FH-AP-GK-LT1-11
3	00:00:46	NA	RUSUN.SELATAN-AP-LT3-24
4	00:00:51	NA	CDT-AP-LT1-06
5	00:00:52	NA	FE-AP-IP-LT1-04
6	00:00:02	185102503@uii.ac.id	FTSP-AP-GK-LT2-13
7	00:00:07	931002109@uii.ac.id	CDT-AP-LT1-04
8	00:00:09	061002415@uii.ac.id	FK-AP-GK-LT1-06
9	00:00:13	181002204	LAB-MIPA-AP-GK-LT3-49
10	00:00:14	17612102@students.uii.ac.id	CDT-AP-LT1-06
11	00:00:15	17612102@students.uii.ac.id	CDT-AP-LT1-06
12	00:00:24	031002417@uii.ac.id	FTI-AP-GK-LT1-10
13	00:00:31	031002417@uii.ac.id	FIAI-AP-GK-LT2-15
14	00:00:45	15211089@uii.ac.id	BOOKSTORE-AP-BS-05
15	00:00:48	16422173	RUSUN.UTARA-AP-LT5-30
16	00:01:05	NA	CDT-AP-LT1-06
17	00:01:07	NA	CDT-AP-LT1-06
18	00:01:08	NA	FE-AP-IP-LT1-04
19	00:01:08	NA	CDT-AP-LT1-06
20	00:01:09	NA	FH-AP-GK-LT1-11

Gambar 4.7 Hasil Pengambilan Variabel Pada Data *Access Point*

4.2.2 Data *Cleaning*

Tahap data *cleaning* digunakan untuk memperbaiki data dari format nilai yang tidak konsisten dan *noise* menjadi data yang dapat dianalisis. Berikut adalah data *cleaning* yang dilakukan dalam penelitian:

1. Menghapus tanggal dan waktu dalam menit dan detik

Proses menghapus tanggal dan waktu dilakukan pada data URL dan data *access point*. Pada data URL, format pada variabel "Receive.Time" terdapat tanggal dan durasi akses pengguna dalam jam, menit, dan detik. Selanjutnya diproses dengan cara menghapus durasi akses dalam format menit dan detik serta tanggal akses, sehingga hanya tersisa jam saja. Format waktu yang baru disimpan dalam variabel "time". Gambar 4.8 merupakan *code* untuk penghapusan karakter domain email data URL yang menghasilkan Gambar 4.9.

```
time1 <- format(as.POSIXct(strptime(analisisData1$Receive.Time,"%Y/%m/%d
  %H:%M:%S",tz="")) ,format = "%H")
analisisData1$time <- as.numeric(time1)
```

Gambar 4.8 *Code* Menghapus Menit, Detik, dan Tanggal Pada Data URL

	Receive.Time	Source.User	Source.address	Application	time
1	2020/09/02 00:00:00	NA	192.168.13.15	ssl	0
2	2020/09/02 00:00:00	NA	103.95.7.17	ssl	0
3	2020/09/02 00:00:00	NA	192.168.15.11	avast-av-update	0
4	2020/09/02 00:00:00	NA	103.95.7.16	avast-av-update	0
5	2020/09/02 00:00:00	NA	10.40.0.216	twitter-base	0
6	2020/09/02 00:00:00	NA	103.95.7.7	twitter-base	0
7	2020/09/02 00:00:00	NA	103.220.113.12	web-browsing	0
8	2020/09/02 00:00:00	NA	103.55.139.35	ssl	0
9	2020/09/02 00:00:00	jogjalearning	192.168.165.109	ssl	0
10	2020/09/02 00:00:00	NA	192.168.164.254	web-browsing	0
11	2020/09/02 00:00:00	NA	103.95.7.7	ssl	0
12	2020/09/02 00:00:00	NA	103.55.139.35	ssl	0
13	2020/09/02 00:00:00	NA	114.4.223.140	ssl	0
14	2020/09/02 00:00:00	NA	192.168.62.89	google-drive-web	0
15	2020/09/02 00:00:00	NA	103.95.7.7	ssl	0
16	2020/09/02 00:00:00	NA	103.95.7.21	google-drive-web	0
17	2020/09/02 00:00:00	uii.ac.id\191005136	10.10.81.18	ssl	0
18	2020/09/02 00:00:00	NA	103.95.7.4	ssl	0
19	2020/09/02 00:00:00	NA	114.4.223.140	ssl	0
20	2020/09/02 00:00:00	NA	10.40.8.22	ssl	0

Gambar 4.9 Hasil Menghapus Menit, Detik, dan Tanggal Pada Data URL

Dalam data *access point* pada variabel “time”, dilakukan penghapusan format durasi akses menit dan detik ke dalam format baru pada variabel “time1” yang hanya mencantumkan durasi akses dalam format jam saja. Gambar 4.10 merupakan hasil perubahan format pada data *access point* berdasarkan *code* pada Gambar 4.11.

```
time1 <- format(as.POSIXct(strptime(temp$time,"%H:%M:%S",tz=""))
  ,format = "%H")
temp$time1 <- as.numeric(time1)
```

Gambar 4.10 *Code* Menghapus Menit dan Detik Pada Data *Access Point*

	time	user	apname	time1
1	00:00:04	NA	FH-AP-GK-LT1-11	0
2	00:00:05	NA	FH-AP-GK-LT1-11	0
3	00:00:46	NA	RUSUN.SELATAN-AP-LT3-24	0
4	00:00:51	NA	CDT-AP-LT1-06	0
5	00:00:52	NA	FE-AP-IP-LT1-04	0
6	00:00:02	185102503@uii.ac.id	FTSP-AP-GK-LT2-13	0
7	00:00:07	931002109@uii.ac.id	CDT-AP-LT1-04	0
8	00:00:09	061002415@uii.ac.id	FK-AP-GK-LT1-06	0
9	00:00:13	181002204	LAB-MIPA-AP-GK-LT3-49	0
10	00:00:14	17612102@students.uii.ac.id	CDT-AP-LT1-06	0
11	00:00:15	17612102@students.uii.ac.id	CDT-AP-LT1-06	0
12	00:00:24	031002417@uii.ac.id	FTI-AP-GK-LT1-10	0
13	00:00:31	031002417@uii.ac.id	FIAI-AP-GK-LT2-15	0
14	00:00:45	15211089@uii.ac.id	BOOKSTORE-AP-BS-05	0
15	00:00:48	16422173	RUSUN.UTARA-AP-LT5-30	0
16	00:01:05	NA	CDT-AP-LT1-06	0
17	00:01:07	NA	CDT-AP-LT1-06	0
18	00:01:08	NA	FE-AP-IP-LT1-04	0
19	00:01:08	NA	CDT-AP-LT1-06	0
20	00:01:09	NA	FH-AP-GK-LT1-11	0

Gambar 4.11 Hasil Menghapus Menit dan Detik Pada Data *Access Point*

Proses menghapus tanggal dan waktu dalam menit dan detik yang hanya menyisakan format jam pada data URL dan data *access point* menghasilkan format waktu yang konsisten pada kedua data tersebut. Selanjutnya format tersebut dapat digunakan dalam menyinkronkan waktu data URL dan data *access point* sehingga kedua data memiliki *range* waktu yang sama.

2. Menghapus karakter domain email

Menghapus karakter domain email dilakukan pada data URL dan data *access point*. Pada data URL, karakter yang dihapus adalah karakter “uii\ac\id\” dan karakter setelah “@” seperti “@uii.ac.id”. Gambar 4.12 merupakan *code* untuk penghapusan karakter domain email data URL yang menghasilkan seperti pada Gambar 4.13.

```
library(tidyr)
library(dplyr)
library('stringr')

analisisData <- analisisData[c(2,3,4)]
analisisData <- analisisData %>% group_by(Source.User =
  str_remove(Source.User, "(@.+$)|(^.+\\\\)"))
```

Gambar 4.12 *Code* Menghapus Karakter Domain Email Pada Data URL

	Source.User	Source.address	Application
1	NA	192.168.13.15	ssl
2	NA	103.95.7.17	ssl
3	NA	192.168.15.11	avast-av-update
4	NA	103.95.7.16	avast-av-update
5	NA	10.40.0.216	twitter-base
6	NA	103.95.7.7	twitter-base
7	NA	103.220.113.12	web-browsing
8	NA	103.55.139.35	ssl
9	jogjalearning	192.168.165.109	ssl
10	NA	192.168.164.254	web-browsing
11	NA	103.95.7.7	ssl
12	NA	103.55.139.35	ssl
13	NA	114.4.223.140	ssl
14	NA	192.168.62.89	google-drive-web
15	NA	103.95.7.7	ssl
16	NA	103.95.7.21	google-drive-web
17	191005136	10.10.81.18	ssl
18	NA	103.95.7.4	ssl
19	NA	114.4.223.140	ssl
20	NA	10.40.8.22	ssl

Gambar 4.13 Hasil Penghapusan Karakter Domain Email Pada Data URL

Sedangkan pada data *access point* penghapusan karakter domain email hanya dilakukan pada karakter setelah "@" seperti "@students.uii.ac.id". Gambar 4.15 merupakan hasil penghapusan karakter pada data *access point* berdasarkan *code* pada Gambar 4.14.

```
temp<- test_extraction %>% select(user, apname)
temp <- temp %>% group_by(user = str_remove(user, '(@.+$)'))
```

Gambar 4.14 *Code* Penghapusan Karakter Domain Email Pada Data *Access Point*

	user	apname
1	NA	FH-AP-GK-LT1-11
2	NA	FH-AP-GK-LT1-11
3	NA	RUSUN.SELATAN-AP-LT3-24
4	NA	CDT-AP-LT1-06
5	NA	FE-AP-IP-LT1-04
6	185102503	FTSP-AP-GK-LT2-13
7	931002109	CDT-AP-LT1-04
8	061002415	FK-AP-GK-LT1-06
9	181002204	LAB-MIPA-AP-GK-LT3-49
10	17612102	CDT-AP-LT1-06
11	17612102	CDT-AP-LT1-06
12	031002417	FTI-AP-GK-LT1-10
13	031002417	FIAI-AP-GK-LT2-15
14	15211089	BOOKSTORE-AP-BS-05
15	16422173	RUSUN.UTARA-AP-LT5-30
16	NA	CDT-AP-LT1-06
17	NA	CDT-AP-LT1-06
18	NA	FE-AP-IP-LT1-04
19	NA	CDT-AP-LT1-06
20	NA	FH-AP-GK-LT1-11

Gambar 4.15 Hasil Penghapusan Karakter Domain Email Pada Data *Access Point*

Proses menghapus karakter domain email dengan memperbaiki penamaan akun pada data URL dan data *access point* menghasilkan penamaan akun yang konsisten pada kedua data tersebut.

3. Menghapus nilai NA

Tahap selanjutnya adalah mengelompokkan aplikasi yang digunakan oleh masing-masing akun dan menghapus baris yang bernilai *NA*. Gambar 4.16 merupakan *code* untuk mengelompokkan aplikasi dan menghapus nilai *NA* pada data URL, dan Gambar 4.17 merupakan hasilnya.

```
library(data.table)
#Menghapus data NA atau null
myData <- setDT(analisisData3)[, .(Freq = .N), by =
  .(Source.User,Source.address,Application)]
myData <- myData[rowSums(myData=="")!=ncol(myData), ]
```

Gambar 4.16 *Code* Pengelompokan Aplikasi dan Menghapus Nilai NA Pada Data URL

	Source.User	Source.address	Application	Freq
1	sutarno	10.10.81.200	google-base	124
2	sutarno	10.10.81.200	ssl	114
3	sutarno	10.10.81.200	facebook-base	36
4	sutarno	10.10.81.200	facebook-video	106
5	sutarno	10.10.81.200	whatsapp-base	8
6	sutarno	10.10.81.200	web-browsing	6
7	sutarno	10.10.81.200	youtube-base	14
8	staffit	10.60.24.3	chrome-remote-desktop	9
9	staffit	10.60.24.3	ssl	6
10	staffit	10.60.24.3	google-base	2
11	staffit	10.60.24.3	ms-update	1
12	staffit	10.60.24.3	web-browsing	1
13	rafifalkhusni.2017	10.10.83.98	web-browsing	3
14	rafifalkhusni.2017	10.10.83.98	google-base	3
15	proyek.fiai	10.40.11.32	ssl	1177
16	proyek.fiai	10.40.11.32	facebook-base	106
17	proyek.fiai	10.40.11.32	google-base	100
18	proyek.fiai	10.40.11.32	whatsapp-base	6
19	proyek.fiai	10.40.11.32	dns-over-tls	43
20	proyek.fiai	10.40.11.32	facebook-video	6

Gambar 4.17 Hasil Pengelompokan Aplikasi dan Menghapus Nilai *NA* Pada Data URL

Sedangkan pada data *access point* hanya menghapus baris yang bernilai *NA*. Gambar 4.18 menunjukkan *code* dan memberikan hasil penghapusan nilai *NA* pada data *access point* seperti pada Gambar 4.19.

```
temp <- temp %>% drop_na()
```

Gambar 4.18 *Code* Penghapusan Nilai NA Pada Data *Access Point*

	user	apname
1	185102503	FTSP-AP-GK-LT2-13
2	931002109	CDT-AP-LT1-04
3	061002415	FK-AP-GK-LT1-06
4	181002204	LAB-MIPA-AP-GK-LT3-49
5	17612102	CDT-AP-LT1-06
6	17612102	CDT-AP-LT1-06
7	031002417	FTI-AP-GK-LT1-10
8	031002417	FIAI-AP-GK-LT2-15
9	15211089	BOOKSTORE-AP-BS-05
10	16422173	RUSUN.UTARA-AP-LT5-30
11	13523181	CDT-AP-LT1-06
12	13523181	CDT-AP-LT1-06
13	181005120	RUSUN.UTARA-AP-LT5-30
14	17513169	CDT-AP-LT1-06
15	031002414	BOOKSTORE-AP-BS-05
16	921002102	FTSP-AP-GK-LT4-01
17	035230102	REKTORAT-AP-LT4-38
18	16711090	AUDITORIUM-AP-LT1-16
19	16513001	BOOKSTORE-AP-LT2-15
20	11410592	FH-AP-GK-LT3-43

Gambar 4.19 Hasil Penghapusan Nilai *NA* Pada Data *Access Point*

4.2.3 Data *Transformation*

Tahap data *transformation* dilakukan untuk menambah variabel baru yang akan mewakili faktor lokasi dan akses yang dilakukan saat menggunaan wifi dalam mendeteksi adanya *fraud*. Selain itu, data *transformation* juga digunakan untuk mengubah data ke dalam bentuk numerik sehingga data dapat digunakan dalam proses *clustering*. Adapun data transformation yang dilakukan pada penelitian ini antara lain:

1. Memberikan label kemiripan aplikasi

Pada tahap ini, setiap akun diberikan label pada variabel “Similar”. Pemberian label kemiripan aplikasi hanya dilakukan pada data URL. Akun diberikan label 1 jika terdapat kemiripan aplikasi dan label 0 jika terdapat ketidakmiripan aplikasi yang digunakan oleh masing-masing IP pada akun yang sama. Gambar 4.21 merupakan hasil dari pemberian label pada variabel “similar” berdasarkan *code* pada Gambar 4.20.

```
#Pemberian label aplikasi, 1 jika masing" perangkat mirip dan 0 jika
tidak mirip
labelAppUser <- myData %>% dplyr::group_by(Source.User, Application) %>%
  dplyr::mutate(c = n()) %>% dplyr::mutate(Similar = case_when(c > 1 ~
  1, TRUE ~ 0)) %>% dplyr::select(-c)
```

Gambar 4.20 *Code* Pemberian Label

	Source.User	Source.address	Application	Freq	Similar
310	191005137	10.10.81.207	facebook-base	172	0
311	191005137	10.10.81.207	instagram-base	10	0
312	191005137	10.10.81.207	facebook-video	10	0
313	191005137	10.10.81.207	taobao	5	0
314	191005137	10.10.81.207	alipay	4	0
315	191005137	10.10.81.207	google-analytics	2	0
316	191005137	10.10.81.207	whatsapp-base	2	0
317	191005137	10.10.81.207	naver-line	47	0
318	191005136	10.10.81.18	ssl	11	1
319	191005136	10.10.81.189	liveperson	1	0
320	191005136	10.10.81.189	gmail-base	1	0
321	191005136	10.10.81.189	ssl	33	1
322	191005136	10.10.81.189	web-browsing	2	1
323	191005136	10.10.81.18	web-browsing	1	1
324	191005136	10.10.81.18	google-base	1	1
325	191005136	10.10.81.189	google-base	3	1
326	191005136	10.10.81.189	google-play	2	0
327	191005136	10.10.81.189	linkedin-base	5	0
328	191005135	10.10.81.3	google-base	56	0
329	191005135	10.10.81.3	instagram-base	5	0

Gambar 4.21 Hasil Pemberian Label

2. Menghitung persentase label kemiripan aplikasi

Setelah memberikan label, selanjutnya akan dihitung persentase label 0 yang berada pada variabel “Similar0” dan persentase label 1 pada variabel “Similar1” untuk setiap akun. Variabel “Similar0” merupakan hasil dari perhitungan persentase ketidakmiripan aplikasi yang digunakan pengguna, sedangkan variabel “Similar1” merupakan persentase kemiripan aplikasi. *Code* perhitungan persentase ditunjukkan pada Gambar 4.22, sedangkan Gambar 4.23 merupakan hasil dari perhitungan persentase.

```
#Memberikan nilai persentase kemiripan app (label 0 dan 1) setiap
pengguna
similarAppUser <- labelAppUser[c(1,5)]
similarAppUser <- similarAppUser %>% dplyr::mutate(Similar=
  factor(Similar)) %>% dplyr::count(Source.User, Similar, .drop = FALSE,
  name = 'Percentage') %>% group_by(Source.User) %>%
  dplyr::mutate(Percentage = Percentage/sum(Percentage) * 100) %>%
  pivot_wider(names_from = Similar, values_from = Percentage,
  names_prefix = "Similar",)
```

Gambar 4.22 *Code* Perhitungan Persentase

	Source.User	Similar0	Similar1
1	001002407	100.000000	0.00000
2	001002425	100.000000	0.00000
3	001002433	100.000000	0.00000
4	001002437	100.000000	0.00000
5	011002421	100.000000	0.00000
6	011002425	100.000000	0.00000
7	011002428	100.000000	0.00000
8	011002444	100.000000	0.00000
9	021002406	100.000000	0.00000
10	021002408	100.000000	0.00000
11	021002425	100.000000	0.00000
12	021002429	100.000000	0.00000
13	031002414	100.000000	0.00000
14	031002415	100.000000	0.00000
15	031002417	100.000000	0.00000
16	035202403	100.000000	0.00000
17	035230102	100.000000	0.00000
18	041002419	100.000000	0.00000
19	041002421	100.000000	0.00000
20	041002423	33.333333	66.66667

Gambar 4.23 Hasil Perhitungan Persentase

3. Menghitung jumlah IP

Pada tahap ini, dilakukan perhitungan jumlah IP pada data URL yang digunakan oleh setiap pengguna. Variabel “freq” merupakan hasil dari perhitungan jumlah IP atau perangkat yang digunakan setiap akun. Gambar 4.24 menunjukkan *code* untuk menghasilkan perhitungan jumlah IP pada Gambar 4.25.

```
#Mengelompokkan IP yang digunakan masing-masing akun
analisisData2 <- analisisData[c(1,2)]
analisisData2 <- analisisData2 %>% group_by(Source.User =
  str_remove(Source.User, "(@.+$)|(^.+\\\\)"))
ipByUser <- setDT(analisisData2)[, .(Freq = .N), by =
  .(Source.User,Source.address)]
ipByUser <- ipByUser[rowSums(ipByUser=="")!=ncol(ipByUser), ]
#Menghitung jumlah IP
ipUser <- dplyr::count(ipByUser, Source.User)
ipUser <- ipUser %>% drop_na()
```

Gambar 4.24 *Code* Perhitungan IP

	user	freq
38	11511002	1
39	11511261	1
40	12512188	1
41	12525081	1
42	131002215	3
43	13313159	1
44	13511299	1
45	13523181	2
46	14311563	2
47	14321036	1
48	151002203	1
49	151002206	1
50	151002225	2
51	15211089	1
52	15313065	1
53	15313303	1
54	15320172	1
55	15322010	1
56	15423073	1
57	15511158	2

Gambar 4.25 Hasil Perhitungan IP

4. Menghitung jumlah *Access Point*

Pada tahap ini, dilakukan perhitungan jumlah *acces point* yang digunakan oleh setiap akun. Variabel “n” merupakan hasil dari perhitungan jumlah *access point* yang digunakan setiap akun. Gambar 4.27 merupakan hasil dari perhitungan jumlah *access point* yang digunakan oleh setiap akun berdasarkan *code* pada Gambar 4.26.

```
#Menghitung Jumlah AP
temp <- unique(temp)
dataAksesPoint <- dplyr::count(temp, user)
```

Gambar 4.26 *Code* Perhitungan Jumlah *Access*

	user	n
1	001002407	1
2	001002425	2
3	001002433	1
4	001002437	1
5	011002421	1
6	011002428	1
7	011002444	1
8	021002406	1
9	021002408	2
10	021002425	2
11	021002429	2
12	031002414	3
13	031002415	1
14	031002417	5
15	035202403	1
16	035230102	1
17	041002419	3
18	041002421	2
19	041002423	2
20	041002446	6

Gambar 4.27 Hasil Perhitungan Jumlah *Access Point*

4.2.4 Data *Integration*

Tahap terakhir pada *pre-processing* adalah data *integration* dengan melakukan *merge* jumlah IP pada data URL dan jumlah AP pada data *access point*. Data tersebut digabungkan berdasarkan variabel “user” pada data URL. Gambar 4.29 merupakan hasil penggabungan jumlah IP pada data URL dan jumlah AP pada data *access point* berdasarkan *code* pada Gambar 4.28.

```
#Menggabungkan jumlah IP pada data URL dan jumlah AP pada data akses
point
ipUser <- data.frame(user = ipUser$Source.User, JumlahIP = ipUser$n)
HasilGabungan <- merge(ipUser, dataAksesPoint, by='user', all.x = T,
  suffixes = c(".IP",".AP"))
```

Gambar 4.28 *Code* Penggabungan Data URL dan Data *Access Point*

	user	freq.IP	freq.AP
1	001002407	1	1
2	001002425	1	2
3	001002433	1	1
4	001002437	1	1
5	011002421	1	1
6	011002425	1	NA
7	011002428	1	1
8	011002444	1	1
9	021002406	1	1
10	021002408	1	2
11	021002425	1	2
12	021002429	1	2
13	031002414	1	3
14	031002415	1	1
15	031002417	1	5
16	035202403	1	1
17	035230102	1	1
18	041002419	1	3
19	041002421	1	2
20	041002423	2	2

Gambar 4.29 Hasil Penggabungan Data URL dan Data *Access Point*

Gambar 4.30 adalah *code* untuk Gambar 4.31 yang merupakan hasil akhir dari proses *pre-processing* yang menyajikan variabel “user”, “JumlahIP”, “JumlahAP” dan variabel “PersentaseKetidakmiripan”. Variabel “JumlahIP” dan “JumlahAP” dapat menggambarkan jumlah perangkat yang digunakan dan lokasi pengguna, sedangkan variabel “PersentaseKetidakmiripan” dapat menggambarkan akses yang dilakukan pada saat menggunakan wifi UIIConnect.

```
#Hasil preprocessing
hasilPreprocessing <- data.frame(User = HasilGabungan$user, JumlahIP =
  HasilGabungan$JumlahIP, JumlahAP = HasilGabungan$n,
  PersentaseKetidakmiripan = similarAppUser$Similar0)
i1 <- with(hasilPreprocessing, JumlahIP ==1 & PersentaseKetidakmiripan
  == 100)
hasilPreprocessing$PersentaseKetidakmiripan[i1] <- 0
hasilPreprocessing1 <- data.frame(hasilPreprocessing, row.names = 1)
hasilPreprocessing1 <- hasilPreprocessing1 %>% drop_na()
```

Gambar 4.30 *Code Pre-Processing*

	user	JumlahIP	JumlahAP	PersentaseKetidakmiripan
75	16711090	1	1	0.000000
76	16711107	1	1	0.000000
77	17213073	2	1	33.333333
78	17213076	2	1	75.000000
79	17311225	1	1	0.000000
80	17321069	1	1	0.000000
81	17410218	1	1	0.000000
82	17410529	1	1	0.000000
83	17410568	1	2	0.000000
84	17421205	1	1	0.000000
85	17422173	1	1	0.000000
86	17423135	2	2	71.428571
87	17512100	1	4	0.000000
88	17512170	2	4	72.413793
89	17513169	1	1	0.000000
90	17521052	1	1	0.000000
91	17521119	1	5	0.000000
92	17522231	1	1	0.000000
93	17523055	1	4	0.000000
94	17524030	2	1	50.000000

Gambar 4.31 Hasil *Pre-Processing*

4.3 *Dimension Reduction*

Pada tahap ini menggunakan fungsi **prcomp** untuk melakukan *dimension reduction* menggunakan PCA. Gambar 4.32 merupakan *code* untuk menampilkan Gambar 4.33 yang merupakan hasil dari nilai yang dikembalikan oleh fungsi tersebut. Fungsi **prcomp** akan mengembalikan nilai seperti *center*, *variance* dan lain-lain.

```
fraudDataResult1New <- fraudDataResult1[, 2:4]
pc <- prcomp(fraudDataResult1New,
             center = TRUE,
             scale. = TRUE)
```

Gambar 4.32 *Code* Fungsi PCA

```
> pc
Standard deviations (1, .., p=3):
[1] 1.3565408 0.9997019 0.4004910

Rotation (n x k) = (3 x 3):
                                PC1         PC2           PC3
JumlahIP                  0.7068517 -0.01915481  0.7071023720
JumlahAP                 -0.0266291 -0.99964528 -0.0004598927
PersentaseKetidakmiripan  0.7068604 -0.01850442 -0.7071110408
```

Gambar 4.33 Nilai Fungsi PCA

4.3.1 Standardisasi Data

Pada hasil *pre-processing*, masing-masing variabel memiliki *range* nilai yang berbeda dan cukup jauh. Maka dari itu perlu dilakukan standardisasi data dengan melakukan skala ulang sehingga menghasilkan nilai yang akan memiliki *impact* yang sama dan *comparable.* Berikut adalah hasil nilai-nilai yang dikembalikan oleh fungsi **prcomp**:

a. Nilai *center*

Hasil nilai *center* yang dikembalikan oleh fungsi PCA ditampilkan melalui *code* pada Gambar 4.34. Pada Gambar 4.35 menunjukkan variabel "JumlahAP" menghasilkan nilai *centering* (1.19200), "JumlahAP" (1.72800) dan "PersentaseKetidakmiripan" (9.20512).

```
pc$center
```

Gambar 4.34 *Code* Pemanggilan Nilai *Center*

```
  JumlahIP                 JumlahAP PersentaseKetidakmiripan
   1.19200                  1.72800                   9.20512
```

Gambar 4.35 Hasil Nilai *Center*

b. Nilai kovarian matrix

Hasil nilai kovarian matrix ditampilkan melalui code pada Gambar 4.36. Hasil pada Gambar 4.37 menampilkan hubungan masing-masing variabel dengan setiap *principal component.* Selanjutnya dapat dilihat hasil standardisasi data pada Gambar 4.39 dengan menggunakan *code* pada Gambar 4.38.

```
pc$rotation
```

Gambar 4.36 *Code* Pemanggilan Nilai Kovarian Matrix

	PC1	PC2	PC3
JumlahIP	0.7068517	-0.01915481	0.7071023720
JumlahAP	-0.0266291	-0.99964528	-0.0004598927
PersentaseKetidakmiripan	0.7068604	-0.01850442	-0.7071110408

Gambar 4.37 Hasil Nilai Kovarian Matrix

```
pc$x
```

Gambar 4.38 *Code* Standardisasi Data

	PC1	PC2	PC3
001002407	-0.5760803	0.4471305	-0.01205311
001002425	-0.5918685	-0.1455519	-0.01232577
001002433	-0.5760803	0.4471305	-0.01205311
001002437	-0.5760803	0.4471305	-0.01205311
011002421	-0.5760803	0.4471305	-0.01205311
011002428	-0.5760803	0.4471305	-0.01205311
011002444	-0.5760803	0.4471305	-0.01205311
021002406	-0.5760803	0.4471305	-0.01205311
021002408	-0.5918685	-0.1455519	-0.01232577
021002425	-0.5918685	-0.1455519	-0.01232577
021002429	-0.5918685	-0.1455519	-0.01232577
031002414	-0.6076567	-0.7382342	-0.01259844
031002415	-0.5760803	0.4471305	-0.01205311
031002417	-0.6392331	-1.9235990	-0.01314378
035202403	-0.5760803	0.4471305	-0.01205311
035230102	-0.5760803	0.4471305	-0.01205311
041002419	-0.6076567	-0.7382342	-0.01259844
041002421	-0.5918685	-0.1455519	-0.01232577

Gambar 4.39 Hasil Standardisasi Data

4.3.2 Nilai *Variance*

Pada Gambar 4.41 merupakan nilai *variance* setiap *principal component* berdasarkan *code* pada Gambar 4.40. *Principal component* (PC) merupakan dimensi baru yang merangkum banyaknya informasi yang ada pada data hasil *pre-processing* sebelumnya. Hasil *variance* yang didapat oleh “PC1” (0.6134), “PC2” (0.3331) dan “PC3” (0.05346).

```
summary(pc)
```

Gambar 4.40 *Code* Pemanggilan Nilai *Variance*

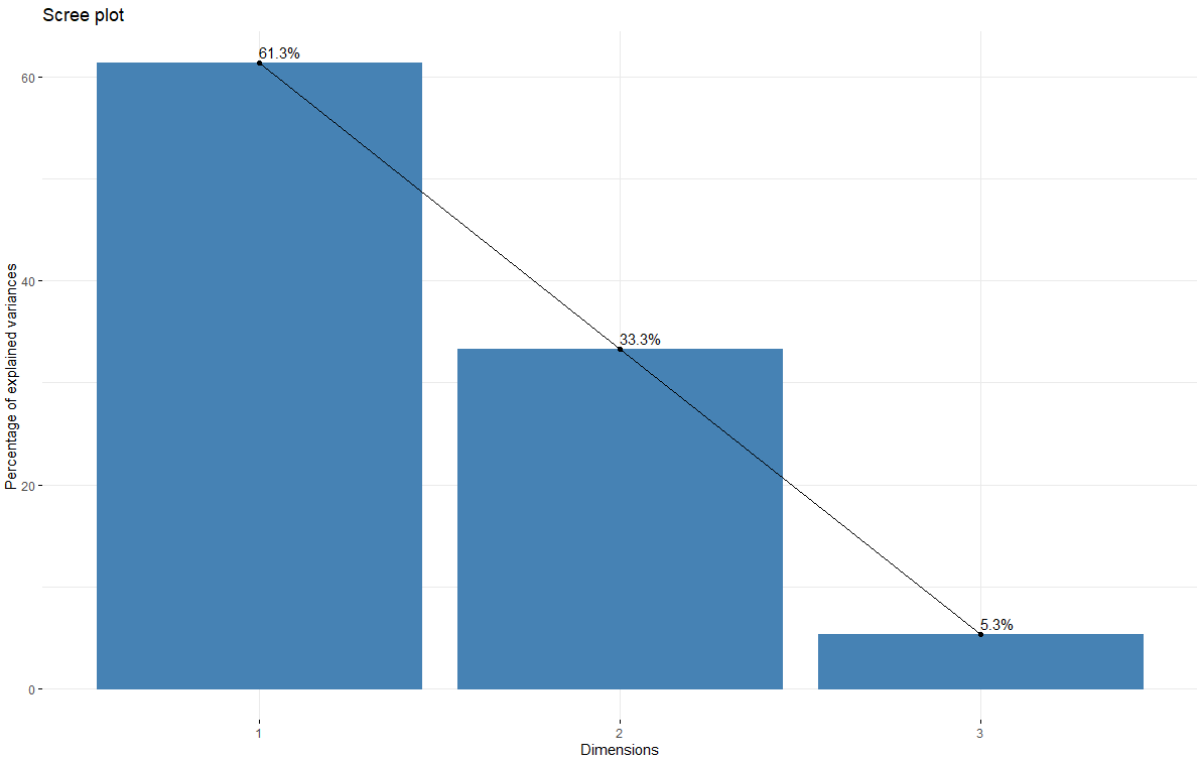
```
Importance of components:
                          PC1    PC2     PC3
Standard deviation     1.3565 0.9997 0.40049
Proportion of Variance 0.6134 0.3331 0.05346
Cumulative Proportion  0.6134 0.9465 1.00000
```

Gambar 4.41 Hasil Nilai *Variance*

Selanjutnya *variance* tersebut dapat dilakukan ploting untk melihat total *variance* yang dibawa oleh masing-masing *component* dengan *code* Gambar 4.42. Pada Gambar 4.43, PC1 merangkum informasi paling besar dengan nilai 61.3%, PC2 33.3% dan PC3 dengan nilai terkecil 5.3%. Dengan mengambil komponen PC1 dan PC2 dari *plot* tersebut maka didapatkan total informasi yang dirangkum dari data hasil *pre-processing* sebesar 94.6 %. Dengan hanya mengambil dua komponen tersebut penelitian ini berhasil melakukan *dimension reduction* sehingga proses komputasi menjadi lebih ringan karena tidak perlu melakukan komputasi semua komponen.

```
fviz_eig(pc, addlabels = TRUE)
```

Gambar 4.42 *Code* Plot *Variance*



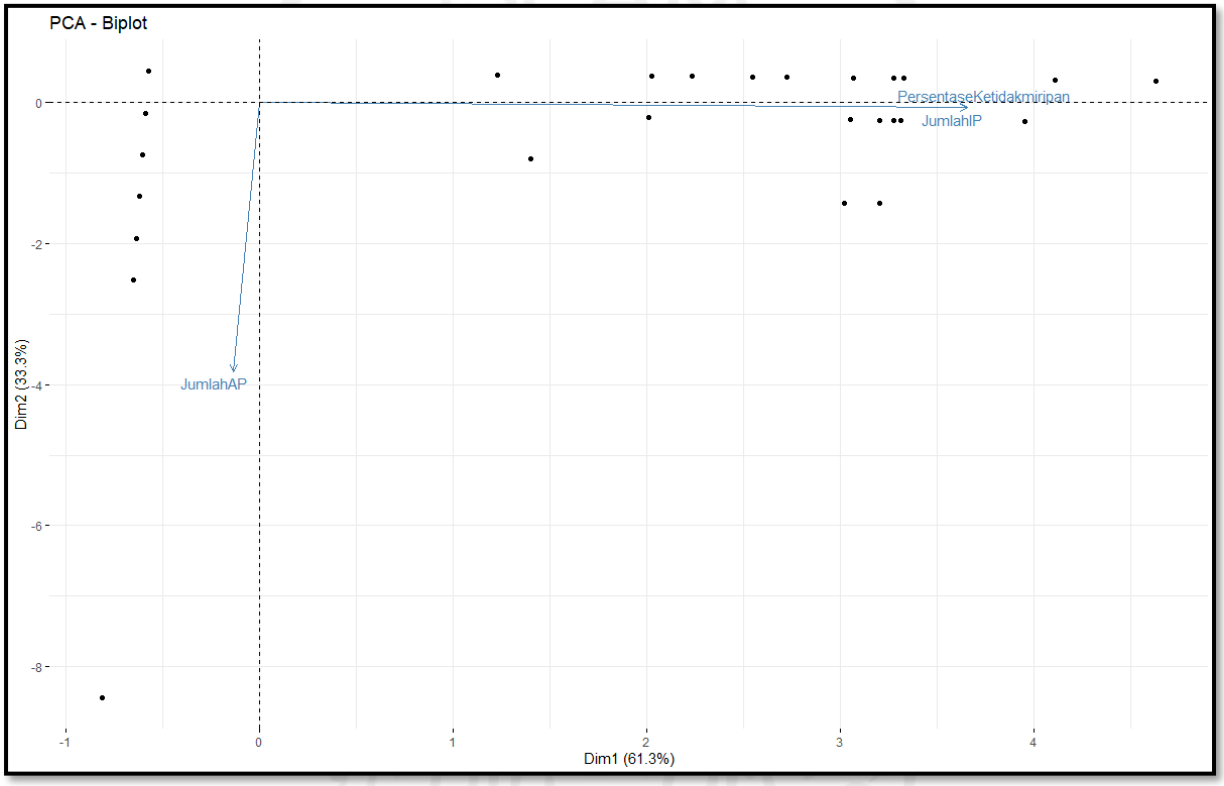
Gambar 4.43 Hasil Plot *Variance*

Selain itu, pada Gambar 4.45 dapat dilihat hasil *biplot* pada sumbu vertikal (Dim1) dan sumbu horizontal (Dim2). Kemudian pada *biplot* tersebut juga divisualisasikan setiap variabel

sebagai bentuk vektor. Dari plot tersebut, dapat dilihat arah vektor "JumlahIP" dan "PersentaseKetidakmiripan" cenderung horizontal seperti arah *principal component* yang kedua (Dim1). Hal ini mengindikasikan bahwa variabel "JumlahIP" dan "PersentaseKetidakmiripan" lebih banyak dijelaskan atau diwakili oleh *principal component* yang pertama. Sebaliknya, arah vektor "JumlahAP" lebih mendekati arah *principal component* yang kedua (Dim2). Hal ini mengindikasikan jika informasi yang dibawa oleh variabel "JumlahAP" lebih banyak diwakili oleh *principal component* yang kedua.

```
fviz_cluster(result,fraudDataResult2, frame = FALSE, geom = "point")
```

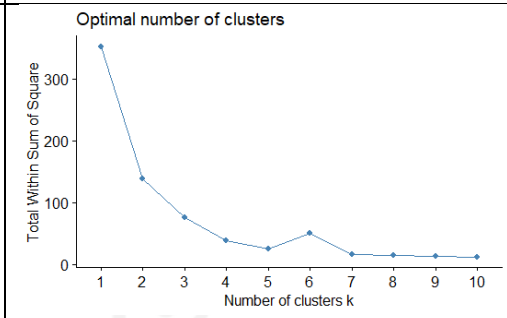
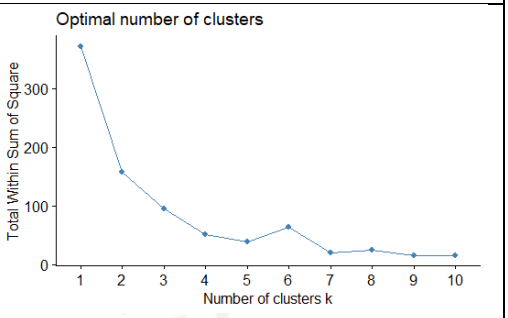
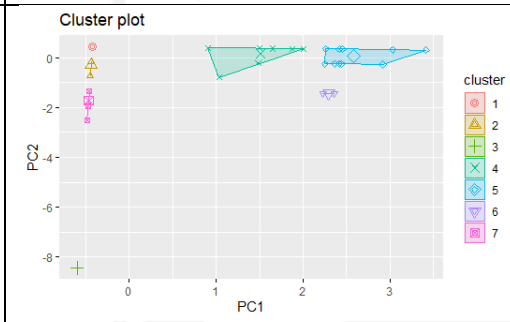
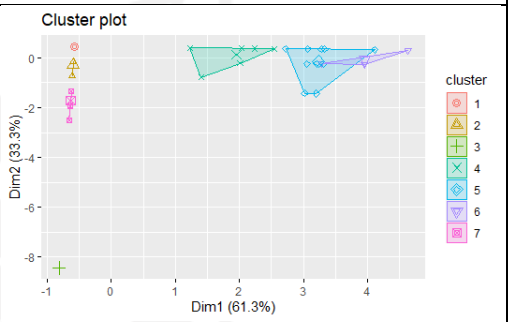
Gambar 4.44 *Code Biplot* PC1 dan PC2



Gambar 4.45 *Code* Hasil *Biplot* PC1 dan PC2

Sebelum melakukan *clustering*, dilakukan proses *dimension reduction* agar proses *clustering* menjadi lebih baik. Berikut adalah perbandingan *clustering* metode k-means menggunakan PCA dan tidak yang ditampilkan pada Tabel 4.1.

Tabel 4.1 Tabel Perbandingan K-means *Clustering* Menggunakan PCA atau Tidak

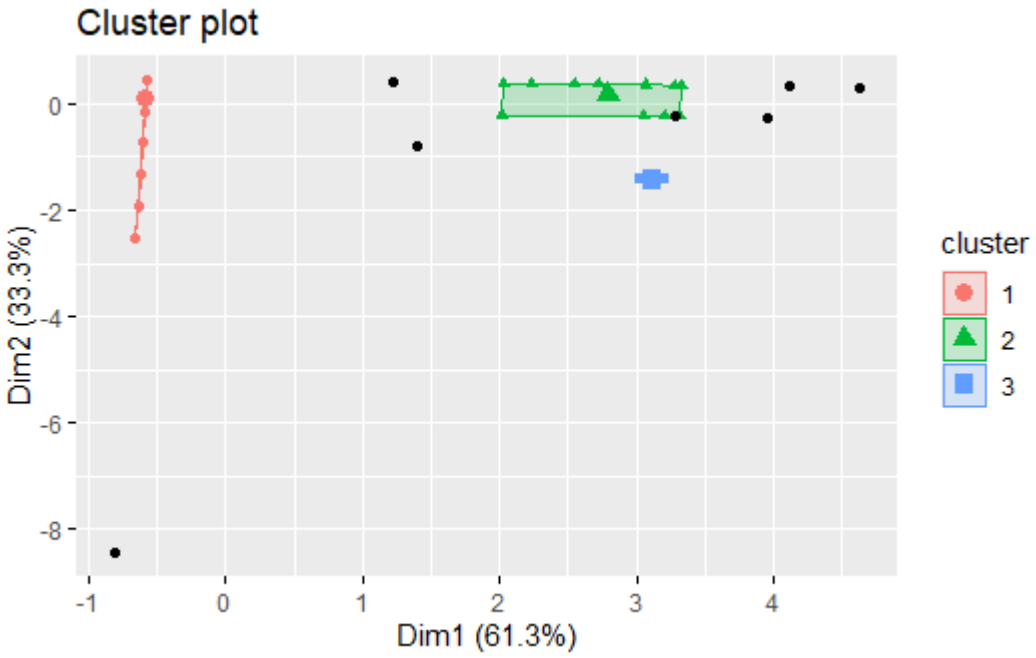
	K-means dan PCA	**K-means**
Elbow Method		
Hasil Plot		
Score SSE	10.19534	16.54752

Hasil perbandingan performa *clustering* metode k-means dengan cara menggunakan PCA dan tidak berdasarkan hasil dari *elbow method*, hasil plot, dan *score* SSE. Hasil *elbow method* kedua cara tersebut memiliki bentuk yang mirip dan jumlah *cluster* optimalnya sama yaitu 7. Jumlah *cluster* optimal tersebut dapat dilihat dari bentuk siku dan *score* SSE yang paling kecil. Selanjutnya hasil plot pada kedua cara tersebut memiliki bentuk plot yang berbeda. Pada metode k-means dan PCA, hasil *clustering* lebih baik karena setiap *cluster* memiliki karakteristik masing-masing. Sedangkan untuk cara yang hanya menggunakan metode k-means menghasilkan plot setiap *cluster* yang kurang baik karena terdapat *cluster* yang memiliki karakteristik anggota yang berbeda tetapi dijadikan dalam satu *cluster*. Berdasarkan hasil *score* SSE dengan jumlah *cluster* 7, kedua cara tersebut menghasilkan *score* yang berbeda. Penggunaan PCA sebelum *clustering* menghasilkan *score* SSE yang lebih baik dengan nilai 10.19534 sedangkan ketika tidak menggunakan PCA menghasilkan nilai 16.54752. Berdasarkan analisis pada Tabel 4.1 penelitian ini melakukan *dimension reduction* dengan menggunakan PCA sebelum melakukan *clustering* menggunakan k-means.

4.4 *Clustering*

4.4.1 *Density-based Spatial Clustering of Applications with Noise* (DBSCAN)

Sebelum melakukan *clustering* menggunakan metode k-means, penelitian ini juga pernah melakukan *clustering* menggunakan metode *Density-based Spatial Clustering of Applications with Noise* atau DBSCAN. Metode ini melakulan *clustering* berdasarkan *density* atau kepadatan data. Terdapat dua nilai parameter untuk melakukan *clustering* menggunakan DBSCAN yaitu *epsilon* yang merupakan jarak sebuah titik terhadap tetangganya dan *minpts* yang merupakan minimal jumlah tetangga di dalam ukuran *epsilon*. Hasil *cluster* plot menggunakan DBSCAN ditampilkan pada Gambar 4.46 dan hasil dari *clustering* ditunjukkan pada Gambar 4.47.



Gambar 4.46 Hasil Plot DBSCAN

	JumlahIP	JumlahAP	PersentaseKetidakmiripan	cluster
061002420	1	16	0.000000	0
061002422	2	1	40.000000	2
061002425	1	1	0.000000	1
074200505	2	1	55.555556	2
091002119	1	1	0.000000	1
097110403	1	1	0.000000	1
114100101	1	1	0.000000	1
11410592	2	4	66.666667	3
11511002	1	1	0.000000	1
11511261	1	1	0.000000	1
12512188	1	4	0.000000	1
12525081	1	1	0.000000	1
131002215	3	2	23.809524	0
13313159	1	2	0.000000	1
13511299	1	1	0.000000	1
13523181	2	1	73.333333	2
14311563	2	1	7.692308	0
14321036	1	1	0.000000	1
151002203	1	2	0.000000	1
151002206	1	1	0.000000	1

Gambar 4.47 Hasil *clustering* menggunakan DBSCAN

Dari plot pada Gambar 4.47 dapat dilihat hasil *clustering* menggunakan DBSCAN membentuk tiga *cluster* sedangkan kumpulan data yang tidak masuk kedalam tiga *cluster* tersebut masuk kedalam *cluster* 0 yang disebut *noise*. Selanjutnya dapat dilihat juga pada Gambar 4.47 bahwa ketiga *cluster* tidak menggambarkan karakteristik pengguna sehingga pada *cluster* yang sama terdapat anggota yang memiliki karakteristik yang berbeda.

Pada penelitian ini, metode DBSCAN tidak dilanjutkan dikarenakan tidak cocok dengan masalah yang diangkat penelitian ini. DBSCAN lebih cocok jika digunakan untuk mencari *noise* pada suatu data. *Noise* yang dicari dengan metode DBSCAN bukan karakteristik pengguna yang terindikasi melakukan *fraud* sehingga peneliti memutuskan untuk mengganti metode dalam melakukan *clustering*.

4.4.2 K-means

Pada *clustering* menggunakan k-means, langkah pertama yang dilakukan adalah menentukan jumlah *cluster* optimal menggunakan *elbow method* pada Gambar 4.48. Dari hasil *chart* pada Gambar 4.49 titik *cluster* yang membentu siku yaitu *cluster* 2, 5 dan 7. Selanjutnya dilakukan analisis lebih lanjut untuk mengetahui jumlah *cluster* yang optimal. Dapat dilihat pada Tabel 4.2 menampilkan perbandingan *score* SSE dan besaran persentase kemiripan pada masing-masing *cluster*. Untuk menentukan *cluster* yang terbaik, dapat dilihat dari *score* SSE terkecil dan persentase kemiripan setiap anggota pada *cluster* yang sama dengan nilai terbesar.

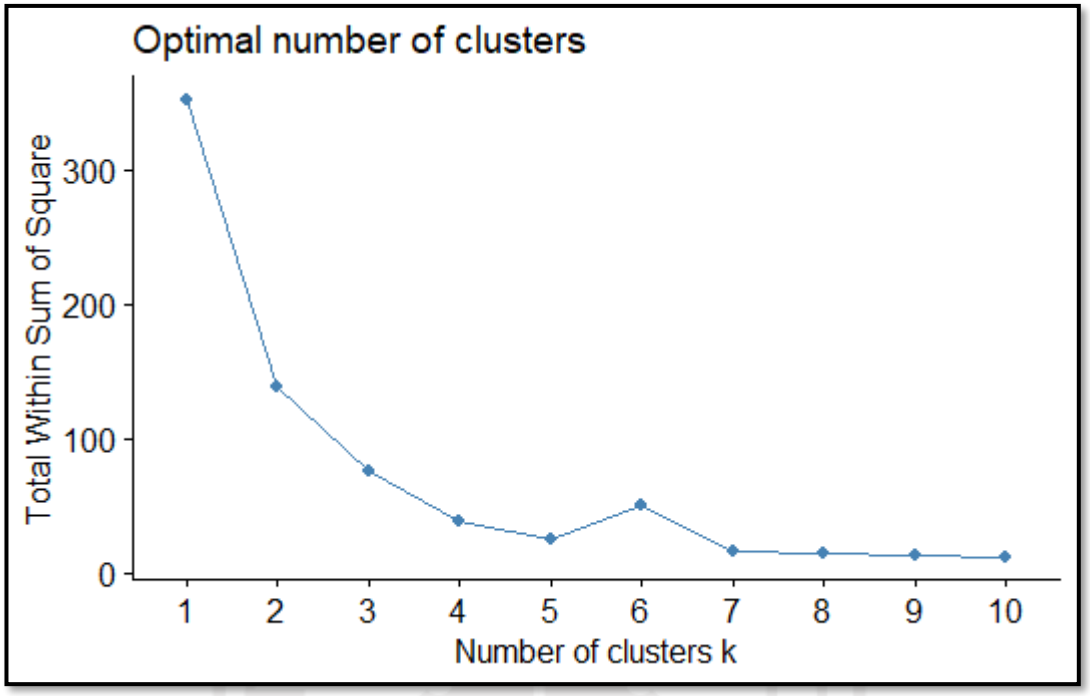
Tabel 4.2 Tabel Perbandingan *Score Cluster*

Cluster	***Score* SSE**	**Persentase Kemiripan**
2	138.2041	60.7%
5	21.25351	94%
7	10.19534	97.1%

```
library(FactoMineR)

# Elbow Method
set.seed(123)
pcData <- as.data.frame(pc$x)
fraudDataResult2 <- data.frame(PC1 = pcData$PC1, PC2 = pcData$PC2)
wss <- function(k) {
  kmeans(fraudDataResult2, k, nstart = 1000, iter.max = 1000
  )$tot.withinss
}
fviz_nbclust(fraudDataResult2, kmeans, method = "wss")
```

Gambar 4.48 *Code* Visualisasi Jumlah *Cluster* Optimal Menggunakan *Elbow Method*



Gambar 4.49 Hasil Visualisasi Jumlah *Cluster* Optimal Menggunakan *Elbow Method*

Berdasarkan hasil perbandingan pada Tabel 4.2 dapat diketahui *optimal cluster* menggunakan *elbow method* yaitu *cluster* 7. *Cluster* 7 dipilih karena *cluster* tersebut membentuk siku dan memiliki nilai SSE terkecil yaitu 10.19534. Selanjutnya proses *clustering* menggunakan metode k-means dengan *code* seperti pada Gambar 4.50. Selain itu, dapat dilihat pada Gambar 4.51 nilai yang dikembalikan fungsi k-means seperti *cluster means* masing-masing *cluster* untuk setiap variabel, *clustering vector*, dan persentase kemiripan setiap anggota pada *cluster* yang sama mencapai 97.1%.

```
result <- kmeans(fraudDataResult2, 7, nstart = 1000, iter.max = 1000)
fraudDataResult1$cluster <- factor(result$cluster)
```

Gambar 4.50 K-means *Clustering Code*

```
K-means clustering with 7 clusters of sizes 2, 70, 9, 1, 11, 24, 8

Cluster means:
         PC1         PC2
1  3.1117478 -1.43013548
2 -0.5760803  0.44713053
3 -0.6339703 -1.72603821
4 -0.8129032 -8.44310522
5  3.4995831  0.06901765
6 -0.5951577 -0.26902735
7  2.0511413  0.15450562

Clustering vector:
  [1] 2 6 2 2 2 2 2 2 6 6 6 6 2 3 2 2 6 6 7 3 3 2 2 7 3 6 2 4 7 2 7 2 2 2 1 2 2 3 2 5 6 2 5 7 2 6 2 7 6 2 2 2 2 2 5
 [56] 6 6 2 2 2 2 2 6 6 2 2 5 2 2 2 2 6 5 2 2 2 7 5 2 2 2 2 6 2 2 5 3 1 2 2 3 2 3 7 2 6 6 2 6 2 2 2 2 2 2 6 5 2 6 3
[111] 2 5 2 2 5 2 5 6 2 2 2 2 2 2 6

Within cluster sum of squares by cluster:
[1] 1.613910e-02 3.127710e-29 2.109130e+00 0.000000e+00 3.484730e+00 1.391440e+00 3.193901e+00
 (between_SS / total_SS =  97.1 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
```

Gambar 4.51 Nilai Fungsi K-means

	JumlahIP	JumlahAP	PersentaseKetidakmiripan	cluster
16422173	2	1	100.000000	2
15511158	2	2	75.000000	2
17213076	2	1	75.000000	2
13523181	2	1	73.333333	2
191005137	2	1	73.333333	2
17512170	2	4	72.413793	5
17423135	2	2	71.428571	2
18512044	2	2	66.666667	2
18524103	3	1	66.666667	2
191005133	2	1	66.666667	2
11410592	2	4	66.666667	5
074200505	2	1	55.555556	6
17524030	2	1	50.000000	6
16523222	3	2	45.454545	2
061002412	2	1	40.000000	6
061002422	2	1	40.000000	6
041002423	2	2	33.333333	6
17213073	2	1	33.333333	6
131002215	3	2	23.809524	2
151002225	2	3	14.285714	6

Gambar 4.52 Hasil *Clustering* Menggunakan K-means

Hasil *clustering* menggunakan metode k-means dengan jumlah 7 *cluster* ditunjukkan pada Gambar 4.52. Untuk menentukan kelompok pengguna yang terindikasi melakukan *fraud* dan tidak, dilakukan analisis setiap variabel pada semua *cluster*. Berikut adalah hasil dari analisis yang berisikan kesimpulan, analisis, dan alasannya yang ditampilkan pada Tabel 4.3.

Tabel 4.3 Tabel Hasil Analisis

Cluster	**Kesimpulan**	**Analisis**	**Alasan**
1	Kelompok 1 merupakan pengguna yang dalam satu hari cendrung menggunakan 1 perangkat, berada dalam 2-3 lokasi, dan persentase ketidakmiripan aplikasi yang digunakan diatas 0%.	Kelompok pengguna normal	Karena pengguna pada kelompok 1 hanya menggunakan 1 perangkat saja.
2	Kelompok 2 merupakan pengguna yang dalam satu hari cendrung menggunakan 2-3 perangkat, berada dalam 1-2 lokasi, dan persentase ketidakmiripan aplikasi yang digunakan diatas 23%-100%.	Kelompok pengguna terindikasi melakukan *fraud*	Karena pengguna pada kelompok 2 menggunakan cukup banyak perangkat yaitu 2-3, berada 1-2 lokasi dan persentase ketidakmiripan aplikasinya 23%-100%.
3	Kelompok 3 merupakan pengguna yang dalam satu hari cendrung hanya menggunakan 1 perangkat, berada dibanyak lokasi yaitu 16, dan persentase ketidakmiripan aplikasi yang digunakan 0%.	Kelompok pengguna normal	Walupun berada dibanyak lokasi, pengguna pada kelompok 3 hanya menggunakan 1 perangkat saja dan persentase ketidakmiripan aplikasi 0%.
4	Kelompok 4 merupakan pengguna yang dalam satu hari cendrung hanya menggunakan 1 perangkat, tetapi berada dibanyak lokasi yaitu 4-6, dan persentase ketidakmiripan aplikasi yang digunakan 0%.	Kelompok pengguna normal	Sama seperti kelompok 3, kelompok 4 juga berada dibanyak lokasi tetapi tidak sebanyak kelompok 3. Selain itu, pengguna pada kelompok 4 hanya menggunakan 1 perangkat saja dan persentase ketidakmiripan aplikasi 0%.
5	Kelompok 5 merupakan pengguna yang dalam satu hari cendrung menggunakan	Kelompok pengguna terindikasi	Karena pengguna pada kelompok 5 menggunakan 2 perangkat, berada di 4 lokasi

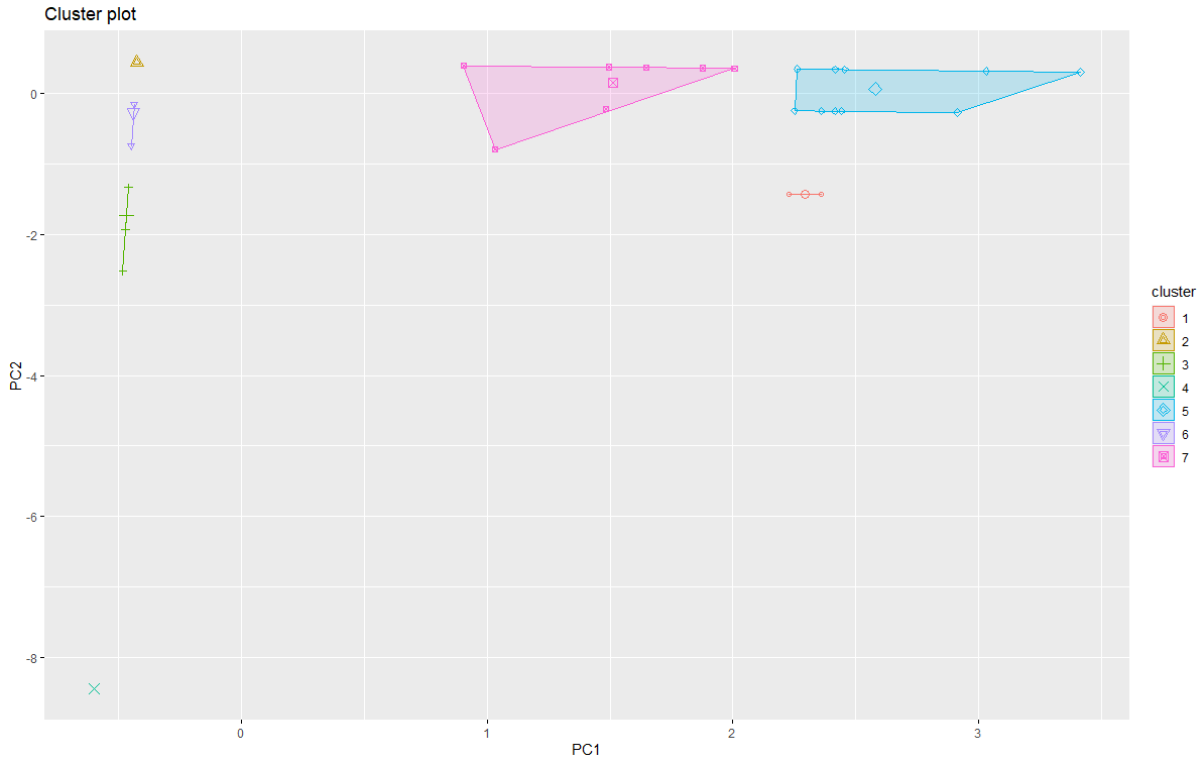
	2 perangkat, berada di 4 lokasi, dan persentase ketidakmiripan aplikasi yang digunakan tinggi 60%-72%.	melakukan *fraud*	dan persentase ketidakmiripan aplikasinya yang tinggi.
6	Kelompok 6 merupakan pengguna yang dalam satu hari cendrung menggunakan 2 perangkat, berada di 1 lokasi, dan persentase ketidakmiripan aplikasi yang digunakan hanya 40%.	Kelompok pengguna normal	Karena pengguna pada kelompok 6 hanya berada di 1 lokasi, menggunakan 2 perangkat dan persentase ketidakmiripan aplikasi yang digunakan hanya 40%.
7	Kelompok 7 merupakan pengguna yang dalam satu hari cendrung hanya menggunakan 1 perangkat, berada di 1 lokasi, dan persentase ketidakmiripan aplikasi yang digunakan 0%.	Kelompok pengguna normal	Sama seperti kelompok 1, kelompok 7 juga menggunakn 1 perangkat tetapi berada di 1 lokasi. Selain itu, persentase ketidakmiripan aplikasi 0%.

Dari hasil Tabel 4.3 dapat ditentukan *threshold* untuk mengetahui pengguna yang terindikasi melakukan *fraud* dan tidak. Pengguna yang terindikasi melakukan *fraud* adalah jika memenuhi kondisi pada saat menggunakan perangkat lebih dari satu, berada dibanyak lokasi, dan persentase ketidakmiripan aplikasi yang digunakan diatas 60%. Sedangkan untuk pengguna yang normal tidak memenuhi kondisi-kondisi tersebut.

Selanjutnya dapat dilihat hasil plot *clustering* menggunakan metode k-means dengan *code* pada Gambar 4.53 yang menggunakan **fviz_cluster**. Pada Gambar 4.54 menunjukkan hasil plot k-means dengan *principal component* yang merupakan hasil visualisasi dari Gambar 4.54.

```
fviz_cluster(result,fraudDataResult2, frame = FALSE, geom = "point")
```

Gambar 4.53 *Code* Plot K-means *Clustering*



Gambar 4.54 Hasil Plot K-means *Clustering*

4.5 Implementasi

4.5.1 *Library*

Pada Gambar 4.55 merupakan *library* yang digunakan untuk mengimplementasikan pemodelan ke dalam Shiny App. ***Library* shiny** merupakan *library* utama yang digunakan untuk membangun aplikasi Shiny dengan menggunakan R. ***Library* shinydashboard** dan **shinydashboardplus** merupakan *library* yang memudahkan pembuatan dashboard yang menarik. ***Library* tidyverse** dan **tidyr** yang berfungsi untuk memudahkan dalam proses analisis data. ***Library* plotly** digunakan untuk mempermudah proses visualisasi data. ***Library* data.table** dan **DT** digunakan untuk mengolah data beserta dengan tampilannya. ***Library* dplyr** merupakan *tool* yang memudahkan proses manipulasi data. ***Library* stringr** merupakan *tool* yang mempermudah proses manipulasi *string*. ***Library* Factominer** dan **factoextra** digunakan untuk visualisasi dari hasil data analisis.

```
#library
library(shiny)
library(shinydashboard)
library(shinydashboardPlus)
library(tidyverse)
library(plotly)
library(DT)
library(data.table)
library(tidyr)
library(dplyr)
library('stringr')
library(FactoMineR)
library(factoextra)
```

Gambar 4.55 *Library*

4.5.2 Contoh *Code*

Dalam pembuatan shiny app, terdapat *code* untuk UI dan server. Pada Gambar 4.56 menampilkan contoh *code* implementasi UI untuk menampilkan plot *cluster*. Fitur ini berada dalam **boxPlus** dengan menggunakan **plotOutput** yang digunakan untuk menampilkan plot *cluster*, dan **sliderInput** untuk menentukan jumlah *cluster*.

```
boxPlus(
    title = "Clustering Plot",
    closable = FALSE,
    enable_label = TRUE,
    label_status = "danger",
    status = "primary",
    solidHeader = FALSE,
    collapsible = TRUE,
    width = 12,
    sliderInput("kValue", label = "Jumlah Klaster",
                min = 1,
                max = 10,
                value = 1, step = 1),
    plotOutput("plotkmeans")
)
```

Gambar 4.56 Contoh Code UI

Selajutnya pada Gambar 4.57 merupakan contoh *code* server untuk menampilkan plot *cluster* dengan menggunakan **renderPlot**. Dalam *code* server fitur ini juga terdapat *code* pemodelan dari k-means dan PCA. Selain itu, untuk menampung nilai dari **sliderInput** dalam UI memerlukan *code* **input$kValue** di dalam servernya.

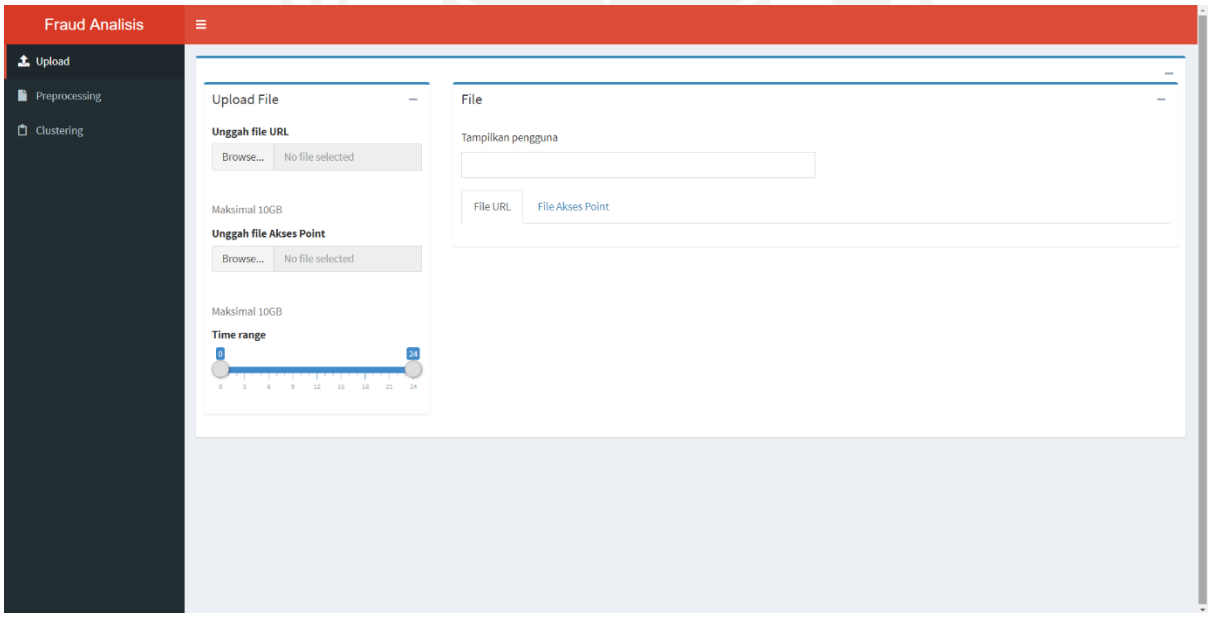
```
# PLOT KMEANS
output$plotkmeans <- renderPlot({
        if(is.null(data()))return()
        dbClustering <- data()
        pc <- prcomp(dbClustering,
                     center = TRUE,
                     scale. = TRUE)
        pcData <- as.data.frame(pc$x)
        dbClustering2 <- data.frame(PC1 = pcData$PC1, PC2 = pcData$PC2)
        result <- kmeans(dbClustering2, centers = input$kValue, nstart =
        1000, iter.max = 1000)
        fviz_cluster(result, dbClustering2, stand = FALSE, frame =
        FALSE, geom = "point")
})
```

Gambar 4.57 Contoh *Code* Server

4.5.3 Hasil

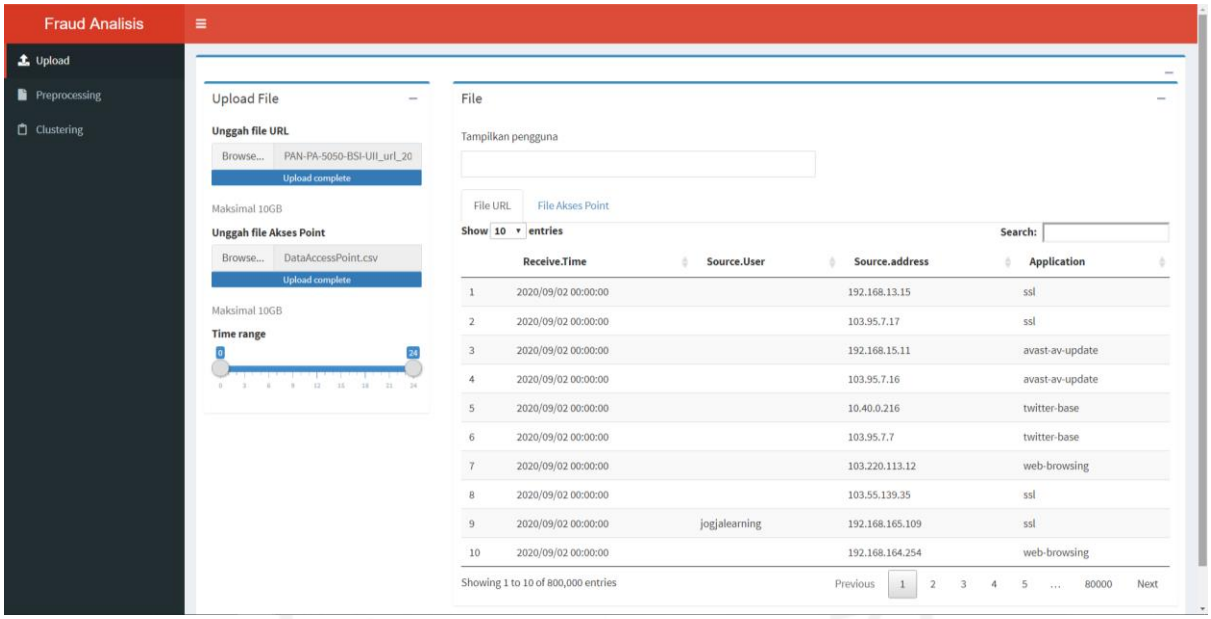
A. Halaman *Upload File*

Pada Gambar 4.58 merupakan tampilan *upload* file URL dan *access point*. Pengguna dapat mengupload *file* dengan cara menekan tombol "Browse" lalu memilih *file* yang akan digunakan dari komputer. Ketentuan *file* yang dapat diupload adalah *file* dengan format csv dan ukuran maksimal masing-masing file 10GB.



Gambar 4.58 Halaman *Upload File* 1

Setelah melakukan *upload file*, sistem menampilkan isi dari masing-masing *file* yang telah diupload dalam bentuk tabel seperti pada Gambar 4.59. Dari *file* yang diupload tersebut, tidak semua variabel ditampilkan, hanya beberapa variabel yang diperlukan saja. Pengguna dapat melihat isi *file* dan menggunakan fitur pencarian "tampilkan pengguna" berdasarkan isi dari variabel "Source.user".

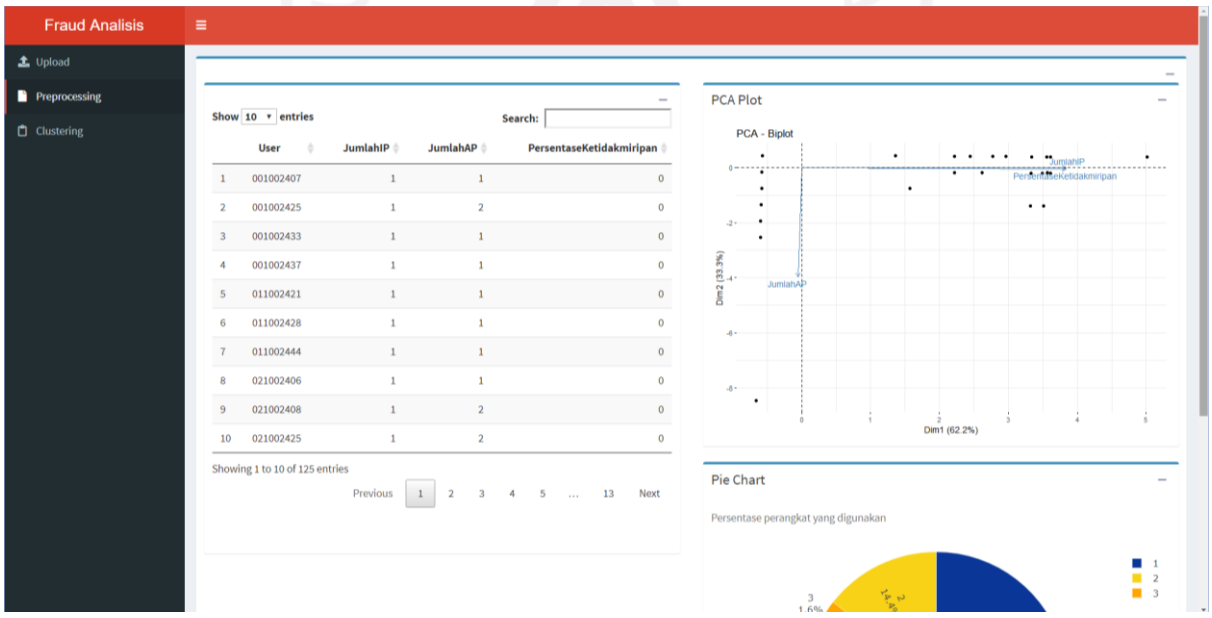


Gambar 4.59 Halaman Upload File 2

B. Halaman Hasil *Pre-processing*

Pada

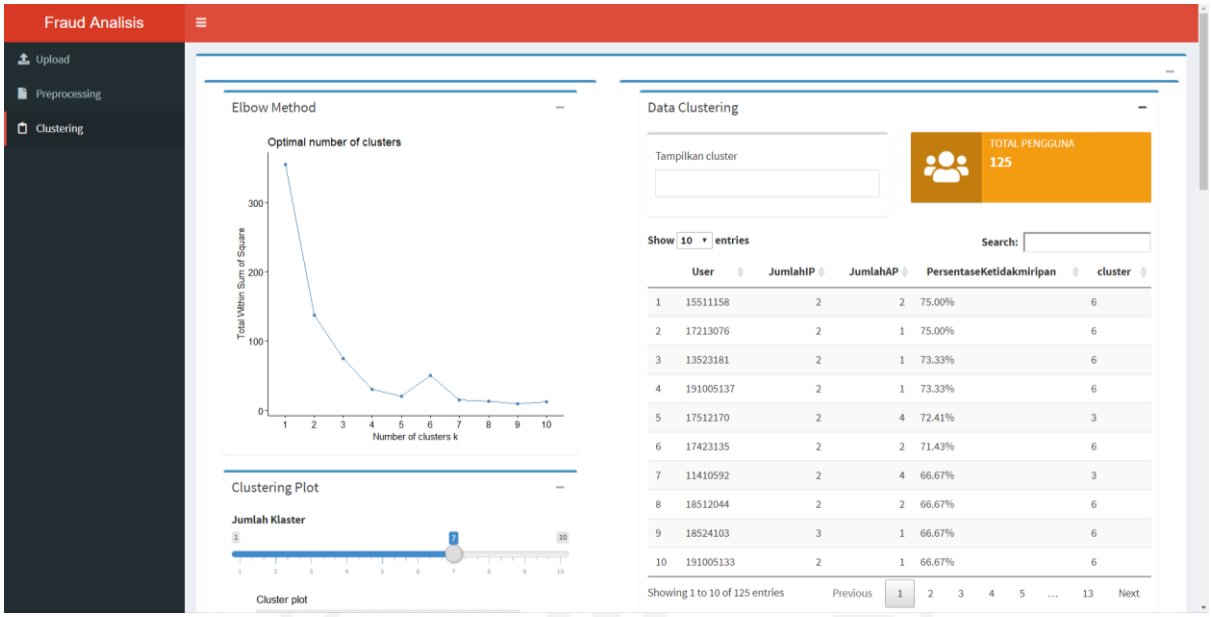
Gambar 4.60 terdapat halaman untuk menampilkan hasil dari *pre-processing* berdasarkan *file* yang sudah diupload sebelumnya. Dalam halaman ini terdapat tabel hasil *pre-processing* yang terdiri dari variabel “JumlahIP”, “JumlahAP” dan “persentaseKetidakmiripan”. Selain itu, terdapat visualisasi *biplot* PCA untuk melihat sebaran data dan visualisasi hubungan vektor setiap variabel dengan *principal component*. Selanjutnya terdapat juga *Pie Chart* yang menampilkan persentase perangkat yang digunakan.



Gambar 4.60 Halaman Hasil *Pre-processing*

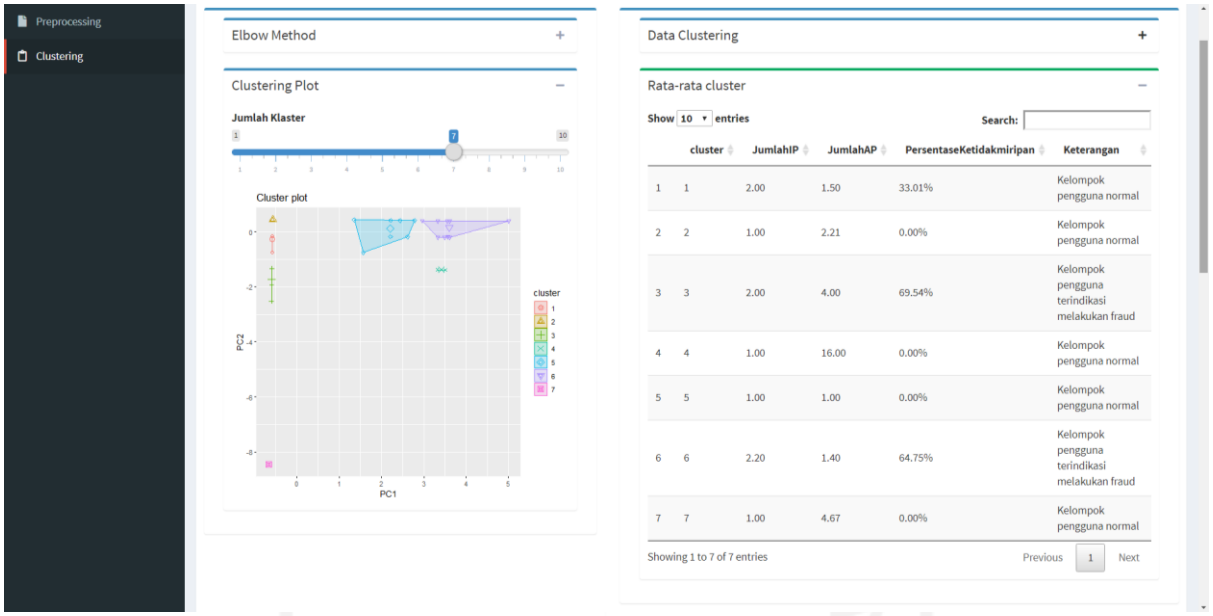
C. Halaman Hasil *Clustering*

Pada Gambar 4.61 merupakan halaman hasil *clustering* yang menampilkan plot dan tabel. Sebelum melakukan proses *clustering* pengguna melakukan analisis dengan melihat plot *elbow method* untuk menentukan jumlah *cluster*. Dari plot tersebut, *cluster* yang dipilih merupakan titik yang membentuk siku dan memiliki nilai SSE paling kecil. Setelah pengguna mengetahui jumlah *cluster* optimal, pengguna menetapkan jumlah *cluster* pada *slider* "Jumlah Klaster". Setelah menetapkan jumlah *cluster* maka aplikasi akan menampilkan tabel hasil *clustering* yang digunakan untuk melakukan analisis pengguna yang terindikasi melakukan *fraud*.



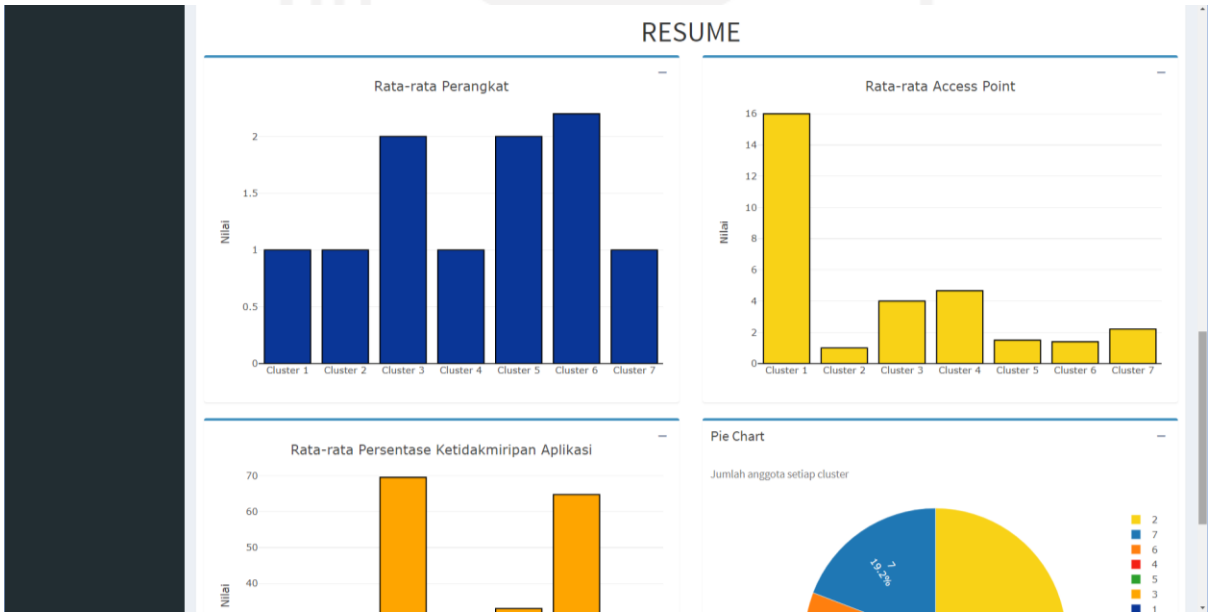
Gambar 4.61 Halaman Hasil *Clustering* 1

Selain itu, pada halaman *clustering* terdapat juga hasil plot k-means dan tabel nilai rata-rata setiap *cluster* pada Gambar 4.62. Pada plot k-means terdapat visualisasi 7 *cluster* dengan dimensi baru yaitu PC1 dan PC2.



Gambar 4.62 Halaman Hasil *Clustering* 2

Terakhir, Gambar 4.63 menampilkan tabel nilai rata-rata setiap *cluster* yang divisualisasikan dengan *bar chart* dan jumlah anggota setiap *cluster* divisualisasikan dengan *pie chart*.



Gambar 4.63 Halaman Hasil *Clustering* 3

4.6 Pengujian

Pengujian sistem menggunakan *black box testing* dilakukan dengan cara mencoba skenario terhadap sistem melalui 12 *test case*. Setelah itu, hasil dari pengujian akan dibandingkan dengan hasil yang diharapkan. Berikut adalah hasil pengujian sistem yang ditampilkan pada Tabel 4.4.

Tabel 4.4 Tabel Pengujian

No	Skenario	*Test Case*	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
A. Halaman *Upload*					
1	Mengunggah *file* URL	Menekan tombol "Browse"	Muncul status "Upload complete"	Sesuai Harapan	Valid
2	Mungunggah *file access point*	Menekan tombol "Browse"	Muncul status "Upload complete"	Sesuai Harapan	Valid
3	Mengunggah *file* URL (csv)	Menekan tombol "Browse" dan muncul status "Upload complete"	Muncul tabel *file* URL	Sesuai Harapan	Valid
4	Mungunggah *file access point* (csv)	Menekan tombol "Browse" dan muncul status "Upload complete"	Muncul tabel file *access point*	Sesuai Harapan	Valid
5	Mencari nama akun pengguna	Mengetik akun nama pengguna	Muncul data pengguna yang dicari	Sesuai Harapan	Valid
6	Menentukan jangka waktu	Menggeser *slider* "Jangka Waktu"	Muncul data sesuai waktu yang ditentukan	Sesuai Harapan	Valid
7	Tidak mengunggah *file*	Mengosongkan *file*	Tidak ada *error*	Sesuai Harapan	Valid
8	Mengunggah *file* dengan format selain csv	Menekan tombol "Browse" dan mengunggah *file* dengan format selain csv	Data tidak dapat diproses	Sesuai Harapan	Valid

B. Halaman Hasil *Pre-processing*					
1	Menampilkan hasil *pre-processing data*	Menekan tombol "Preprocessing" pada *sidebar*	Muncul tabel hasil *pre-processing*, plot PCA, dan *chart* persentase prengkat yang digunakan	Sesuai Harapan	Valid
C. Halaman Hasil *Clustering*					
1	Menampilkan hasil *clustering*	Menekan tombol "Clustering" pada *sidebar*	Muncul tabel plot *elbow method*, tabel nilai rata-rata variabel setiap *cluster*, hasil k-means *clustering*, dan tabel hasil k-means *clusetering*	Sesuai Harapan	Valid
2	Mencari nama akun pengguna	Mengetik akun nama pengguna	Muncul data pengguna yang dicari	Sesuai Harapan	Valid
3	Menentukan jumlah *cluster*	Menggeser *slider* "Jumlah Cluster"	Muncul data hasil *cluster*	Sesuai Harapan	Valid

Dari Tabel 4.4 hasil pengujian *balck box testing* pada seluruh skenario *test case* menunjukkan hasil yang sesuai harapan dengan menghasilkan pengujian sebesar 100%. Sebanyak 12 *test case* yang dilakukan pada aplikasi menghasilkan nilai yang valid.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari pembuatan aplikasi yang bertujuan untuk melakukan pengelompokkan pengguna yang terindikasi melakukan *fraud* dan tidak, dapat disimpulkan bahwa:

a. Penelitian ini telah berhasil dalam mengembangkan aplikasi untuk mengelompokkan pengguna yang terindikasi melakukan *fraud* dan tidak.
b. Hasil k-means *clustering* menggunakan PCA lebih baik daripada hanya menggunakan k-means saja. Hasil dari *dimension reduction* menggunakan *principal component analysis* mendapatkan total *variance* sebesar 94.6% dan hanya menggunakan dua *component principal* sehingga komputasi menjadi lebih baik.
c. Metode *clustering* yang digunakan adalah k-means dan penentuan jumlah *cluster* menggunakan *elbow method* yang menghasilkan persentase kemiripan setiap anggota pada *cluster* yang sama mencapai 97.1%.
d. Aplikasi berhasil diimplementasikan menggunakan shiny app.
e. Hasil pengujian aplikasi menggunakan *black box testing* menunjukan performa yang baik dengan 12 skenario *test case* dan menghasilkan pengujian sebesar 100%.

5.2 Saran

Peneliti menyadari bahwa penelitian ini belum sempurna. Maka dari itu, peneliti menyarankan perbaikan untuk penelitian selanjutnya berupa:

a. Menambah faktor penyebab pengguna melakukan *fraud* dengan cara menambah variabel pada data URL seperti variabel “category” yang digunakan untuk mengetahui kategori *website* yang dikunjungi.
b. Menggunakan metode lain dalam mencari optimal *cluster*.

DAFTAR PUSTAKA

Hendrik, Anjomshooa, A., & Tjoa, A. M. (2014). Towards Semantic Mashup Tools For Big Data Analysis. *Proceding of the Information & Communication Technology-EurAsia Conference 2014*, (pp. 100-145). Bali.

Setiawan, A. M. (2013). *Integrated Framework For Business Process Complexity Analysis*. Retrieved from ECIS 2013 Completed Research: http://aisel.aisnet.org/ecis2013_cr/49

Taufiq, H. (2015). *Argumentasi dan Validitas.* Yogyakarta: Darqin.

Wahid, F. (2014). The Antecedents And Impacts of a Green Eprocurement Infrastructure: Evidence From The Indonesian Public Sector. *International Journal of internet Protocol Technology, 7*(4), 210-218.

Zukhri, Z. (2014). *Algoritma Genetika: Metode Komputasi Evolusioner untuk Menyelesaikan Masalah Optimasi.* Yogyakarta: Andi Publisher.

LAMPIRAN