ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR – IS184853

PENGGUNAAN FAKTOR DEMOGRAFI PENGGUNA SOSIAL MEDIA UNTUK MENGKLASIFIKASIKAN AVERAGE REVENUE PER USER (ARPU) PENGGUNA DENGAN MENGGUNAKAN METODE RANDOM FOREST (STUDI KASUS PERUSAHAAN TELEKOMUNIKASI PT.XYZ)

SOCIAL MEDIA USER DEMOGRAPHY FACTORS USAGE FOR AVERAGE REVENUE PER USER (ARPU) CLASSIFICATION WITH RANDOM FOREST METHOD (CASE STUDY TELECOMUNICATION COMPANY PT.XYZ)

RACHMADDINTA HERPRADIPTO ADIANSYAH
NRP 05211540000133

Dosen Pembimbing
Edwin Riksakomara, S.Kom., MT.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

TUGAS AKHIR – IS184853

PENGGUNAAN FAKTOR DEMOGRAFI PENGGUNA SOSIAL MEDIA UNTUK MENGKLASIFIKASIKAN AVERAGE REVENUE PER USER (ARPU) PENGGUNA DENGAN MENGGUNAKAN METODE RANDOM FOREST (STUDI KASUS PERUSAHAAN TELEKOMUNIKASI PT.XYZ)

RACHMADDINTA HERPRADIPTO ADIANSYAH
05211540000133

Dosen Pembimbing
Edwin Riksakomara, S.Kom., MT.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019

FINAL PROJECT – IS184853

SOCIAL MEDIA USER DEMOGRAPHY FACTORS USAGE FOR AVERAGE REVENUE PER USER (ARPU) CLASSIFICATION WITH RANDOM FOREST METHOD (CASE STUDY TELECOMUNICATION COMPANY PT.XYZ)

RACHMADDINTA HERPRADIPTO ADIANSYAH

05211540000133

Supervisor
Edwin Riksakomara, S.Kom., MT.

INFORMATION SYSTEMS DEPARTMENT
Information and Communication Technology Faculty
Sepuluh Nopember Institut of Technology
Surabaya 2019

LEMBAR PENGESAHAN

PENGGUNAAN FAKTOR DEMOGRAFI PENGGUNA SOSIAL MEDIA UNTUK MENGKLASIFIKASIKAN AVERAGE REVENUE PER USER (ARPU) PENGGUNA DENGAN MENGGUNAKAN METODE RANDOM FOREST (STUDI KASUS PERUSAHAAN TELEKOMUNIKASI PT.XYZ)

TUGAS AKHIR
Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:
RACHMADDINTA HERPRADIPTO ADIANSYAH
NRP. 05211540000133

Surabaya, Juli 2019

KEPALA
DEPARTEMEN SISTEM INFORMASI

Mahendrawati ER, S.t., M.Sc., Ph.D
NIP. 19761011 200604 2 001

LEMBAR PERSETUJUAN

PENGGUNAAN FAKTOR DEMOGRAFI PENGGUNA SOSIAL MEDIA UNTUK MENGKLASIFIKASIKAN AVERAGE REVENUE PER USER (ARPU) PENGGUNA DENGAN MENGGUNAAN METODE RANDOM FOREST (STUDI KASUS PERUSAHAAN TELEKOMUNIKASI PT.XYZ)

TUGAS AKHIR
Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:
RACHMADDINTA HERPRADIPTO ADIANSYAH
NRP. 05211540000133

Disetujui Tim Penguji : Tanggal Ujian : Juli 2019
Periode Wisuda : September 2019

Edwin Riksakomara, S.Kom, M.T **(Pembimbing I)**

Ahmad Mukhlason, S.Kom, M.Sc, Ph.D **(Penguji I)**

Raras Tyasnurita, S.Kom, MBA, Ph.D **(Penguji II)**

PENGGUNAAN FAKTOR DEMOGRAFI PENGGUNA SOSIAL MEDIA UNTUK MENGKLASIFIKASIKAN AVERAGE REVENUE PER USER (ARPU) PENGGUNA DENGAN MENGGUNAKAN METODE RANDOM FOREST (STUDI KASUS PERUSAHAAN TELEKOMUNIKASI PT. XYZ)

Nama : **RACHMADDINTA HERPRADIPTO ADIANSYAH**
NRP : **05211540000133**
Departemen : **SISTEM INFORMASI FTIK-ITS**
Dosen Pembimbing : **Edwin Riksakomara, S.Kom., MT.**

ABSTRAK

Pelanggan merupakan aset penting bagi perusahaan. Pada divisi pemasaran perusahaan, pelanggan merupakan objek yang harus dikelola dengan baik. PT. XYZ sebagai salah satu perusahaan operator seluler mempunyai divisi yang dikhususkan untuk melakukan pemasaran yaitu divisi pemasaran digital. Salah satu kegiatan divisi pemasaran digital PT. XYZ adalah mengenali para pengguna layanan perusahaan yaitu telekomunikasi seluler. Untuk membantu divisi pemasaran dalam menyusun strategi pemasaran yang tepat, divisi pemasaran memerlukan pengelompokan penggunana layananya. Dalam hal ini pengelompokan average revenue per user (ARPU) pengguna berdasarkan data demografi pengguna yang mengakses sosial media melalui layanan PT. XYZ. Masalah dalam melakukan prediksi pengelompokan ARPU muncul karena tidak ada rumus pasti dari pola demografi pengguna dalam mengelompokan ARPU. Penelitian ini melakukan klasifikasi yang selanjutnya dapat digunakan untuk memprediksi kelompok ARPU pengguna baru (top usage, very high, high, medium, low, dan very low) berdasarkan demografinya. Dalam penelitian ini pengelompokan menggunakan metode klasifikasi random forest. Model dibangun menggunakan parameter yang dihasilkan dari proses

parameter tuning. Dari pemodelan yang telah dibangun berdasarkan data demografi pengguna sosial media didapatkan hasil akurasi sebesar 37.28% dan nilai rata-rata AUC sebesar 0.77. Berdasarkan performa random forest tidak dapat digunakan untuk melakukan klasifikasi arpu pengguna berdasarkan demografi karena tingkat akurasi yang sangat kecil.

Kata kunci: Penggalian data, Klasifikasi, Random forest, ARPU.

SOCIAL MEDIA USER DEMOGRAPHY FACTORS USAGE FOR AVERAGE REVENUE PER USER (ARPU) CLASSIFICATION WITH RANDOM FOREST METHOD (CASE STUDY TELECOMUNICATION COMPANY PT.XYZ)

Name : **RACHMADDINTA HERPRADIPTO ADIANSYAH**
NRP : **05211540000133**
Department : **SISTEM INFORMASI FTIK-ITS**
Supervisor : **Edwin Riksakomara, S.Kom., MT.**

ABSTRACT

Customer are an important assets for the company. In the company's marketing division, customers are objects that must be managed properly. PT. XYZ as a cellular operator company has a division dedicated to marketing operation called digital marketing division. One of the activities in the division is recognizing users of company services that is cellular telecommunications services. To help the marketing division in drawing up the right marketing strategy, the marketing division needs to group its users. The grouping in this case is by classify average revenue per user (ARPU) users based on demographic data of user who access social media through PT.XYZ telecommunications services. The prblem with predict ARPU classification arises because there is no exact formula for the user's demographic patterns in classifying ARPU. This research classifies which can then be ised to predict new user ARPU groups (top usage, very high, high, medium, low, and very low) based on their demographics. In this study, classification is using random forest method. Model built using parameters generated from parameter tuning process. From the model that has been built based on demographic of social media user data, the accuracy score is 37.28% and the average value of AUC is 0.77.

Keyword: Datamining, Classification, Random forest, ARPU.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada kehadirat Allah SWT atas segala rahmat dan berkat serta hidayahnya yang telah memberikan anugerah dan tuntunan kepada penulis sehingga penulis dapat menyelesaikan tugas akhir dengan judul "**PENGGUNAAN FAKTOR DEMOGRAFI PENGGUNA SOSIAL MEDIA UNTUK MENGKLASIFIKASIKAN AVERAGE REVENUE PER USER (ARPU) PENGGUNA DENGAN MENGGUNAKAN METODE RANDOM FOREST (STUDI KASUS PERUSAHAAN TELEKOMUNIKASI PT. XYZ)**" yang merupakan salah satu syarat kelulusan pada Departemen Sistem Informasi, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya. Penyusunan tugas akhir ini senantiasa mendapatkan dukungan dari berbagai pihak baik dalam bentuk doa, motivasi, semangat, kritik, saran dan berbagai bantuan lainnya. Untuk itu, secara khusus penulis akan menyampaikan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Allah SWT yang telah memberikan hidayah serta atas ijin-Nya pula saya mampu mengerjakan tugas akhir ini hingga selesai.
2. Segenap keluarga besar penulis, terutama kedua orang tua, Bapak Moch. Nurdin dan Ibu Agustina yang selalu senantiasa mendoakan, memberikan motivasi, dan kebutuhan materiil maupun non-materiil sehingga penulis mampu menyelesaikan pendidikan S1 ini dengan baik.
3. Ibu Mahendrawati ER, S.t., M.Sc., Ph.D., selaku Ketua Jurusan Sistem Informasi ITS, Bapak Nisfu Asrul Sani, S.Kom, M.Sc selaku KaProdi S1 Sistem Informasi ITS serta seluruh dosen pengajar beserta staf dan karyawan di Jurusan Sistem Informasi, FTIF ITS Surabaya selama penulis menjalani kuliah.

4. Bapak Edwin Riksakomara, S.Kom, MT. selaku dosen pembimbing yang telah meluangkan waktu untuk membimbing, memberi arahan, dan memberikan ilmu kepada penulis selama pengerjaan tugas akhir ini.
5. Bapak Ahmad Mukhlason, S.Kom, M.Sc, Ph.D dan Ibu Raras Tyasnurita, S.Kom., MBA, Ph.D selaku dosen penguji yang telah memberikan masukan dan saran kepada penulis selama pengerjaan tugas akhir ini.
6. Rekan-rekan divisi pemasaran digital perusahaan PT.XYZ Jakarta yang menerima penulis untuk melakukan penelitian dan memberikan arahan serta data yang berguna dalam proses pengerjaan tugas akhir ini.
7. Teman-teman se-angkatan Lannister khususnya Fariz, Eric, Oky, Rizky, dan Farhan yang telah memberikan motivasi dan masukan kepada penulis selama proses pengerjaan tugas akhir ini.
8. Serta semua pihak yang telah membantu dalam pengerjaan tugas akhir ini yang belum tersebut namanya.

Terima kasih atas segala bantuan, dukungan, serta doanya. Semoga Allah SWT senantiasa melimpahkan anugerah serta membalas kebaikan yang telah diberikan kepada penulis. Penulis pun menyadari bahwa tugas akhir ini memiliki kekurangan, oleh karena itu penulis mengharapkan saran dan masukan demi kebaikan penulis dan tugas akhir ini. Akhir kata, penulis berharap bahwa tugas akhir ini dapat memberikan kebermanfaatan.

Surabaya, Juni 2019
Penulis,

Rachmaddinta Herpradipto A

DAFTAR ISI

DAFTAR GAMBAR

Halaman ini sengaja dikosongkan

DAFTAR TABEL

DAFTAR SKRIP

Halaman ini sengaja dikosongkan

BAB I
PENDAHULUAN

Pada bab ini dijelaskan mengenai penelitian yang meliputi latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta relevansi pengerjaan tugas akhir ini.

1.1. Latar Belakang

Pelanggan merupakan salah satu aset penting perusahaan. PT. XYZ sebagai salah satu perusahaan telekomunikasi di bidang operator seluler biasa menggunakan kata pengguna sebagai pengganti kata pelanggan. Sebagai prospek sumber keuntungan bagi perusahaan, pengguna berperan penting akan keberlangsungan perusahaan. PT. XYZ mempunyai divisi yang berkaitan dengan pemasaran secara digital. Dalam kegiatan divisi pemasaran digital PT. XYZ, pengguna merupakan objek yang harus dikelola. Divisi pemasaran digital merupakan bagian yang berhubungan langsung dengan pengguna. Pemasaran bertujuan untuk menciptakan komunikasi dan memberikan nilai kepada pengguna dan mengelola hubungan pengguna sehingga dapat memberikan keuntungan bagi perusahaan [1]. Untuk mewujudkan tujuanya tentu divisi pemasaran harus mengenali para penggunanya terlebih dahulu. Data demografi pengguna dibuat untuk membantu perusahaan mempelajari para penggunanya. Demografi pengguna berisikan data personal pengguna, meliputi usia, jenis kelamin, status pernikahan, kota asal, dan lain-lain. PT. XYZ mengembangkan data demografi pengguna yang di integrasikan dengan data pengaksesan pengguna terhadap sosial media.

Salah satu pemasalahan yang ada pada divisi pemasaran digital PT. XYZ adalah dalam mengelompokan pengguna. Pengelompokan pengguna dapat dilakukan menggunakan beberapa indikator. Indikator yang dipakai oleh PT. XYZ dalam pengelompokan pelanggan adalah *average revenue per user* (ARPU). ARPU menjelaskan pendapatan perusahaan yang

dihasilkan per user atau pengguna. Setiap pengguna memberikan nilai pendapatan pada perusahaan yang berbeda-beda. Pengelompokan ARPU dilakukan dengan menggunakan data demografi pengguna sosial media. Proses pengelompokan menjadi rumit karena tidak adanya pola yang pasti dalam pengelompokan ARPU ini ditambah dengan jumlah pengguna yang sangat banyak. Berdasarkan sumber Siaran Pers no.112/HM/KOMINFO/05/2018 milik Kementrian Komunikasi dan Informatika Republik Indonesia atau Kominfo, jumlah pengguna operator di indonesia setelah registrasi per tanggal 30 April 2018 berjumlah 254.792.159 nomor pengguna. Jumlah pengguna operator terbagi dengan rincian 150 juta pengguna Telkomsel, 45 juta pengguna XL Axiata, 34 juta Indosat Ooredoo, 17 juta Tri Indonesia, 7 juta pengguna Smartfren dan sisa jumlah pengguna tidak disebutkan [2]. Penggunaan sosial media juga menjadi tren di masyarakat Indonesia. Berdasarkan data yang diterbitkan oleh *We are Social* dan *Hootsuite*, pengguna sosial media di Indonesia menyentuh angka 130 juta pengguna aktif pada tahun 2018 [3].

Penelitian ini dibuat untuk membantu bagian pemasaran PT. XYZ dalam mengelompokan pengguna berdasarkan ARPU pengguna. Permasalahan ini dapat dibantu menggunakan teknik penggalian data atau *datamining*. Penggalian data adalah teknik yang digunakan untuk mengeksplorasi dan menganalisis data yang secara kuantitas sangat besar untuk menemukan pola maupun aturan yang bermakna [4]. Teknik penggalian data yang dipakai dalam masalah ini adalah klasifikasi. Klasifikasi adalah salah satu metode *supervised learning* dimana setiap objek mempunyai label atau kelas yang sudah di tentukan sebelumnya. Metode klasifikasi dalam kasus ini digunakan untuk menentukan kelompok ARPU pengguna menggunakan atribut yang ada pada data demografi pengguna PT. XYZ.

Saat ini terdapat bermacam-macam metode klasifikasi, salah satunya adalah *Random Forest*. *Random Forest* adalah kombinasi dari kumpulan *decision tree* yang setiap *tree* atau pohon bergantung pada suatu nilai sample acak yang telah

didistribusikan secara sama rata pada setiap pohonnya [5]. *Random Forest* digunakan dalam penelitian ini karena merupakan solusi pengklasifikasian yang bersifat universal [6]. Keuntungan lainya dalam menggunakan metode *random forest* adalah metode ini dapat menangani *multiclass* data dengan jumlah data yang besar sehingga sesuai dengan masalah yang dialami pemasaran PT. XYZ [7].

Dengan adanya penelitian ini diharapkan akan membantu bagian pemasaran PT. XYZ dalam mengelompokan tingkat ARPU pengguna. Luaran dari penelitian ini berupa aplikasi pengklasifikasian berbasis *web* yang dibangun menggunakan *framework* Flask. Luaran penelitian berupa aplikasi diharapkan dapat mempermudah dan mempercepat proses pengklasifikasian pengguna PT. XYZ berbasis ARPU berdasarkan data demografi pengguna sosial media.

1.2. Rumusan Masalah

Berdasarkan penjelasan latar belakang sebelumnya, rumusan masalah pada penelitian ini adalah:

1) Bagaimana model klasifikasi menggunakan metode *random forest* pada pengklasifikasian pengguna berbasis ARPU berdasarkan demografi pengguna sosial media dapat memberikan rekomendasi klasifikasi pengguna perusahaan.
2) Bagaimana hasil pengukuran kinerja model menggunakan metode *random forest* berdasarkan hasil uji performa.
3) Bagaimana desain dan implementasi sistem menggunakan model klasifikasi *random forest* dapat mempermudah perusahaan untuk menjalankan model.

1.3. Batasan Masalah

Batasan permasalahan yang digunakan penelitian ini adalah:

1) Data yang digunakan merupakan data demografi pengguna sosial media pada bulan Desember 2017.

2) Data yang digunakan merupakan data pengguna di wilayah Kota Surabaya sebanyak 744.889 pengguna.
3) Aplikasi yang digunakan oleh pengguna adalah Instagram, Whatsapp, Line, Facebook, dan BBM.
4) Metode pengklasifikasian *random forest* dibangun menggunakan bahasa pemograman python. Perancangan sistem berbasis *web* menggunakan *framework* Flask.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1) Membangun model pengklasifikasian pengguna berbasis ARPU berdasarkan demografi pengguna sosial media menggunakan *random forest* guna memberikan rekomendasi klasifikasi pengguna perusahaan.
2) Mengidentifikasi kinerja model klasifikasi ARPU pengguna berdasarkan demografi pengguna sosial media menggunakan *random forest* melalui proses uji performa model.
3) Merancang dan mengimplementasikan aplikasi berbasis web menggunakan model klasifikasi *random forest* yang dapat membantu perusahaan dalam melakukan klasifikasi pengguna.

1.5. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat antara lain:

1) Bagi Peneliti

Mampu memahami dan mengimplementasikan model penggalian data klasifikasi menggunakan *random forest* dalam kasus nyata, yaitu pengklasifikasian pengguna berbasis ARPU berdasarkan demografi pengguna sosial media.

2) Bagi Perusahaan

Memberi rekomendasi bagi perusahan dalam melakukan pengklasifikasian pengguna berbasis ARPU berdasarkan demografi pengguna sosial media. Sehingga dari rekomendasi pengklasifikasian ARPU yang telah dibangun, perusahaan dapat memberikan perlakuan yang tepat pada setiap kelompok demografi pengguna.

1.6. Relevansi

Penelitian ini dibuat dalam rangka mengimplementasikan ilmu yang didapat pada perkuliahaan di Departemen Sistem Informasi ITS dan menjadi syarat kelulusan pada tahap sarjana. Penelitian ini mempunyai relevansi terhadap salah satu lab bidang minat yang ada di Departemen Sistem Informasi ITS yaitu bidang Rekayasa Data dan Intelegensia Bisnis (RDIB). Penelitian ini menggunakan proses penggalian data yaitu klasifikasi yang relevan dengan salah satu sub bidang keilmuan analitikal bisnis dan penggunaan random forest sebagai metode klasifikasi yang relevan dengan sub bidang keilmuan intelegensia sistem. Penelitian ini didukung oleh ilmu yang didapatkan pada perkuliahan sebelumnya khususnya pada mata kuliah seperti Statisitika, Sistem Cerdas, Penggalian Data, Analitikal Bisnis, dan Sistem Pendukung Keputusan.

Halaman ini sengaja dikosongkan

BAB II
TINJAUAN PUSTAKA

Pada bab ini dijelaskan mengenai studi literatur penelitian sebelumnya yang berkaitan dengan penelitian ini. Selain itu pada bab ini dijelaskan dasar teori yang digunakan dalam melaksanakan penelitian ini.

2.1. Penelitian Sebelumnya

Pada bagian ini dijelaskan mengenai penelitian sebelumnya yang berkaitan dengan penelitian ini. Penelitian-penelitian sebelumnya ini di jelaskan lebih rinci pada tabel 2.1.

Tabel 2.1 Penelitian sebelumnya

Penelitian 1	
Judul	Spectral Classification of Asteroids by Random Forest [8].
Penulis	Huang Chao, Ma Yue-Hua, Zhao Hai-bin, Lu Xiao-Ping.
Tahun	2017.
Deskripsi Umum	Penelitian ini bertujuan untuk mengklasifikasikan asteroid berdasarkan data *Moving Object Catalogue* (MOC) yang diperoleh dari hasil observasi *The Sloan Digital Sky Survey* (SDSS) menggunakan *random forest*. Pemodelan pada penelitian ini dilakukan dengan menggunakan *supervised* data (asteroid yang sudah dikelompokan) sejumlah 1120 asteroid untuk selanjutnya di implementasikan pada data asteroid yang belum dikelompokan dengan jumlah 48462 asteroid. Asteroid diklasifikasikan ke dalam delapan kelompok yaitu C, X, S, B, D, K, L, dan V. *Random forest* dibangun menggunakan parameter *n_trees* sebanyak 600 *trees*. Uji performa tidak dilakukan pada penelitian ini.

Keterkaitan	Penerapan metode klasifikasi pada penelitian sebelumnya ini menggunakan metode yang sama dengan penelitian ini, yaitu *random forest*. Sehingga dapat dijadikan referensi dalam penelitian ini.
Penelitian 2	
Judul	Customer Churn Prediction Using Improved Balanced Random Forests [9].
Penulis	Yaya Xie, Xiu Li, E.W.T Ngai, Weiyun Ying.
Tahun	2009.
Deskripsi Umum	Penelitian ini mengambil masalah yang terjadi pada perusahaan bank di China. Fokusan utama yang diangkat adalah dalam kasus *customer churn* atau keadaan dimana konsumen berhenti atau mengganti layanan yang disediakan perusahaan. Penelitian ini bertujuan untuk mengklasifikasikan konsumen apakah ada kemungkinan konsumen itu berhenti atau mengganti layanan yang disesiakan atau tidak (positif atau negative). Penelitian ini menggunakan metode *improved balanced random forest* (IBRF). Metode ini digunakan karena data yang digunakan tidak seimbang. Untuk meningkatkan akurasi digunakan lah metode IBRF dimana data akan di sampling sehingga antara *class* positif dan negatif menjadi seimbang. Hasil dari penelitian ini dapat dilihat dengan lebih tinggi nya akurasi metode ini 93.2%, dibandingkan random forest biasa maupun metode lainya seperti artificial *neural network* 78.1% dan *decision tree* 62.0%.
Keterkaitan	Penelitian ini sama-sama mengangkat masalah pengklasifikasian konsumen. Keterkaitan masalah yang diangkat menjadikan fitur/atribut dataset yang digunakan sama terutama di bagian *personal demographics*. Penggunan metode klasifikasi menggunakan prinsip metode yang sama

	yaitu *random forest*. Penggunaan *random forest akan* diadopsi pada penelitian tugas akhir ini.
Penelitian 3	
Judul	Analysis of User Behaviors by Mining Large Network Data Sets [10].
Penulis	Zhenhua Wang, Lai Tu, Zhe Guo, Laurence T. Yang, Benxiong Huang.
Tahun	2014.
Deskripsi Umum	Penelitian ini bertujuan untuk meneliti keterkaitan perilaku konsumen terhadap average revenue per unit (ARPU) atau rerataan pendapatan per unit. Menggunakan bigdata yang didapat dari salah satu perusahaan *mobile operator* di China. Penelitian ini mengklasterkan konsumen menjadi tiga klaster berdasarkan perilakunya menggunakan metode *Fuzzy c-Means* (FCM). Dari hasil klasterisasi maka akan dilakukan analisa keterkaitan perilaku terhadap ARPU. Pada penelitian ini ARPU dibagi menjadi tiga tingkatan yaitu *High ARPU*, *Medium ARPU*, dan *Low ARPU*. Hasil penelitian ini akan menjawab pertanyaan apakah konsumen dengan tingkat ARPU yang sama mempunyai perilaku yang sama atau mirip satu sama lain.
Keterkaitan	Penelitian ini melakukan pengklasifikasian berdasarkan perilaku konsumen. Dalam proses klasifikasi nya, penelitian ini menggunakan metode klasterisasi menggunakan Fuzzy c-Means (FCM). Pengangkatan masalah penelitian ini yaitu segmentasi konsumen yang menganalisis keterkaitan ARPU berkaitan dengan penelitian ini. Luaran penelitian berupa analisis lanjutan dari hasil klasterisasi berupa visualisasi menjadi referensi penelitian ini.
Penelitian 4	
Judul	Rancang Bangun Aplikasi untuk Klasifikasi Komentar Netizen pada Media Sosial

	Pemerintah Daerah di Indonesia Menggunakan Algoritma *Random Forest* [11].
Penulis	Muhammad Fikry Hazmi.
Tahun	2018.
Deskripsi Umum	Penelitian ini bertujuan untuk menggali informasi dari komentar-komentar yang ada pada media sosial pemerintah daerah Indonesia yaitu facebook, twitter, dan youtube. Data komentar ini didapat dengan melakukan proses *crawling*. Komentar-komentar yang didapat diklasifikasikan menggunakan *random forest* ke dalam beberapa kategori. Penelitian ini menghasilkan aplikasi yang dapat menjalankan proses klasifikasi dan memvisualisasikan hasil yang didapat secara *real-time*. Untuk membantu perancangan aplikasi, penelitian ini menggunakan Kafka sebagai data *pipeline* dan penggunaan Spark dalam proses *machine learning*.
Keterkaitan	Dalam penelitian ini, penggunaan *random forest* dalam pengklasifikasian dapat diadopsi dalam penelitian tugas akhir ini. *Random forest* yang digunakan penelitian ini untuk menglasifikasikan *multiclass* data. *Multiclass* menggunakan *random forest* ini dimana setiap *instance* akan diklasifikasikan tidak hanya ke dalam dua kelas yaitu positif atau negatif tetapi ke dalam beberapa kategori.

2.2. Dasar Teori

Pada bagian ini dijelaskan dasar teori yang menjadi landasan dalam pengerjaan penelitian tugas akhir ini.

2.2.1. Rerataan pendapatan per pengguna (ARPU)

Rerataan pendapatan per pengguna atau yang biasa disebutkan ARPU (*average revenues per unit)* adalah satuan metrik yang biasa digunakan oleh perusahaan telekomunikasi yang

membantu dalam bidang manajemen dan analitik [12]. ARPU sendiri adalah pendapatan yang dihasilkan oleh satu unit, dimana unit dalam perusahaan telekomunikasi adalah *user* atau pengguna. Dalam perhitungan ARPU, periode waktu harus ditentukan. Time period yang biasa dipakasi pada perhitungan ARPU adalah bulanan. ARPU dapat dihitung menggunakan rumus:

$$ARPU = \frac{Total\ N\ Revenues}{N\ of\ Unit}$$

Rumus ARPU menjelaskan bahwa ARPU didapatkan dari total pendapatan N dibagi jumlah N, dimana N adalah jumlah unit yang dipakai dalam perhitungan. Jika perhitungan ARPU untuk setiap satu unit maka ARPU didapatkan dari total pendapatan unit tersebut.

2.2.2. Praproses data

Data *preprocessing* atau praproses data, bisa juga disebut juga data *preparation* adalah salah satu tahapan penting dalam proses penggalian data. Praproses data adalah serangkaian aksi yang diterapkan pada data aktual sebelum data itu dipakai dalam proses selanjutnya dalam penggalian data. Secara umum Praproses data adalah proses transformasi T yang mentransformasikan data mentah X_{ik} menjadi sebuah data baru yang siap dipakai Y_{ij} [13].

$$Y_{ij} = T(X_{ik})$$

(i) Y_{ij} menyimpan nilai informasi yang didapat dari X_{ik}.
(ii) Y_{ij} setidaknya menghilangkan setidaknya satu permasalahan dari X_{ik}.
(iii) Y_{ij} lebih berguna dalam proses selanjutnya daripada X_{ik}. Dan dalam hubungan diatas,
   a) i = 1,….,n dimana n = jumlah objek,
   b) j = 1,….,m dimana m = jumlah atribut/fitur sebelum *data preprocessing*.

c) k = 1,….,o dimana o = jumlah atribut/fitur setelah *data preprocessing*.

Tujuan dari praproses data adalah mengolah data aktual yang mentah menjadi data yang reliabel dan siap digunakan dalam permodelan. Penggunaan data yang reliabel pada proses penggalian data akan mendapatkan hasil yang lebih berkualitas. Kebanyakan data aktual mentah masih "kotor" untuk dilakukan proses penggalian data. Kata kotor disini dapat diartikan data itu tidak komplit, tidak konsisten, *noise* atau terdistorsi, dan ada yang hilang dari data tersebut [14]. Ada banyak metode yang digunakan dalam praproses. Beberapa metode yang bisa dilakukan diantaranya *instance selection, outlier detection, handling missing feature values, data discretization, data normalization,* dan *feature selection* [15].

2.2.3. Klasifikasi

Klasifikasi merupakan proses membangun sebuah model yang mendeskripsikan dan membedakan kelas data sesuai atribut nya. Klasifikasi adalah salah satu metode *supervised learning* dimana setiap objek mempunyai label atau kelas yang sudah di tentukan sebelumnya. Kelas yang telah ditentukan sebelumnya bisa disebut juga kelas target atau kelas tujuan. Tujuan dari klasifikasi adalah untuk secara akurat dapat memprediksikan kelas target untuk setiap objek sesuai dengan atribut prediktor atau bisa disebut juga fitur [16].

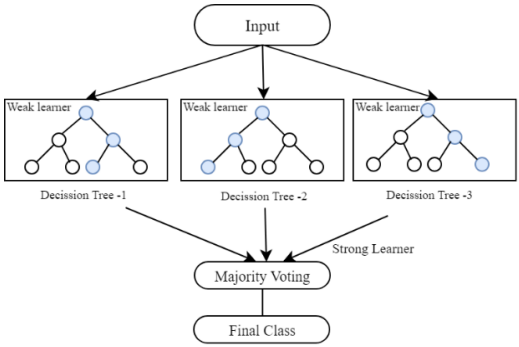
Hal pertama dalam membangun model klasifikasi adalah membagi dataset menjadi dua bagian yaitu *training set* dan *test set*. Saat melakukan fase latih, dimana suatu data yang mempunyai kelas atau label yang telah ditentukan sebelumnya atau bisa disebut juga *training set* dikenalkan kepada mesin atau model yang dibangun. Setiap objek pada *training set* data mempunyai atribut atau fitur dan kelas [17]. Algoritma model akan mencari model terbaik untuk mempelajari keterkaitan antara fitur dan kelas pada training set. Saat model terbaik telah terpilih maka akan dilakukan proses validasi model. Proses validasi dilakukan pada fase selanjutnya yaitu fase uji,

merupakan penerapan model klasifikasi terpilih pada dataset baru bisa disebut juga *test set.*

2.2.4. Random forest

Random forest adalah salah satu algoritma *machine learning* yang digunakan pada *supervised learning*. *Random forest* pertama kali diperkenalkan oleh Breiman pada 2001. Breiman menjelaskan bahwa *random forest* adalah kombinasi dari kumpulan pohon terstruktur $\{h(X, \odot_k), k = 1\}$ dimana $\{\odot_k\}$ adalah pohon yang bergantung pada suatu nilai sample acak yang telah didistribusikan secara sama rata pada setiap pohonya dan setiap pohonya memberikan suara unit untuk kelas paling populer di input X [18]. Sama halnya dengan cara kerja *decision tree*, setiap pohon bertugas untuk memprediksikan kelas data. Keputusan akhir prediksi diambil dari suara atau kelas terbanyak yang diprediksikan oleh pohon-pohonya. Diagram skematik dari algoritma *random forest* dijelaskan pada gambar 2.1. *Random forest* telah menjadi salah satu algoritma terbaik dan paling sukses dalam untuk tujuan yang universal [19].

Algoritma *randorm forest* dikenal sebagai algoritma yang berkerja dengan cara *ensemble* yang artinya adalah suatu kelompok terdiri dari model-model yang lemah yang bersatu saling berkombinasi untuk menjadi model yang *powerful*. Random forest merupakan metode *ensemble* yang menjalankan model menggunakan konsep *bagging (bootstrap agregating)*. Konsep dari bagging sendiri adalah melakukan sampling acak dengan pergantian *(random sampling with replacement)* pada training set dan disimpan menjadi training set baru atau disebut juga bootstrap [20]. Proses bagging ini dilakukan berulang sehingga menghasilkan banyak bootsrap training set. Bootstrap training set akan menjadi input (gambar 2.1) pada skema kerja *random forest.*



Gambar 2.1 Skema cara kerja random forest

Dalam membangun model menggunakan *random forest*, beberapa parameter harus didefinisikan terlebih dahulu seperti jumlah *decision tree* yang dibuat (Ntree), jumlah maksimum level setiap *decision tree* (Max_depth), dan jumlah variabel yang akan dipilih dan diuji untuk di pecah kepada pohon yang telah ditanam (Mtry). Model yang telah dihasilkan akan menghasilkan prediksi pada setiap Ntree yang dibangun dan prediksi akhir diputuskan sesuai dengan pemilihan suara terbanyak untuk klasifikasi dan rata-rata untuk regresi [21].

2.2.5. Uji performa

Uji performa dibutuhkan dalam proses validasi. Uji performa berguna untuk melihat performa klasifikasi yang telah dibangun pada fase latih [16]. Performa dari klasifikasi yang dihasilkan nantinya dapat dievaluasi. performa dari sutau klasifikasi dapat ditunjukan dari tabel kontingensi atau *confusion matrix* [20]. *Confusion matrix* adalah sebuah konsep dalam uji performa dimana berisi informasi tentang *actual class* dan *predicted class* yang dihasilkan klasifikasi [21]. Pada tabel 2.2 menunjukan konsep sederhana *confusion matrix* dari pengklasifikasian dua kelas (positif dan negatif) sehingga mempunyai dua dimensi. Dimensi pertama berisikan keadaan kelas sebenarnya *(aktual class)*. Dimensi lainya berisikan kelas yang diprediksikan oleh klasifikasi *(predicted class)*. Penerapan *confusion matrix* pada *multiclass* menggunakan konsep yang sama.

Tabel 2.2 Confusion matrix

	Aktual (+)	**Aktual (-)**
Predicted (+)	TP (True Positive)	FP (False Positive)
Predicted (-)	FN (False Negative)	TN (True Negative)

Dari tabel 2.2 didapatkan kinerja klasifikasi bisa dievaluasi dengan menggunakan persamaan dibawah ini:

a. Akurasi
Akurasi adalah rasio ketepatan prediksi pengklasifikasian dari seluruh atribut pengamatan.

$$Akurasi = \frac{TP + TN}{TP + TN + FN + FP}$$

Akurasi untuk multiclass data akan dihasilkan dari rata-rata akurasi di setiap kelasnya. Ada tiga metode untuk mengambil nilai akurasi akhir berdasarkan rata-rata setiap kelas yaitu:

i. Micro : rataan dihitung dari semua TP, FN, FP
ii. Macro : rataan dihitung dari setiap kelas nya yang ditotal dan didapatkan rataannya tanpa memperhatikan berat pada setiap kelas.
iii. Weighted : rataan dihitung dari setiap kelasnya yang ditotal dengan memperhatikan jumlahnya sehingga akan didapatkan berat pada setiap kelasnya.

b. Presisi
Presisi adalah rasio prediksi positive yang benar dibandingkan dengan dengan total prediksi positive.

$$Presisi = \frac{TP}{TP + FP}$$

c. Recall

Recall adalah rasio prediksi yang benar dibandingkan dengan kondisi aktual yang benar.

$$Akurasi = \frac{TP}{TP + FN}$$

d. F1 Score

F1 Score adalah rata-rata dari presisi dan recall.

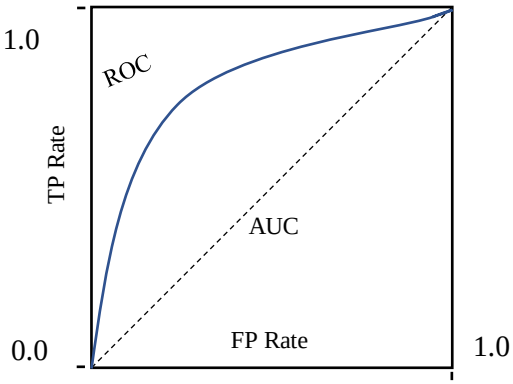
$$F1\ Score = \frac{2\ \times (Recall\ \times Presisi)}{(Recall + Presisi)}$$

Selain itu, uji performa dapat dilakukan dengan menggunakan grafik ROC *(Receiver operating characteristics)*. Grafik ROC akan memvisualisasikan bagaimana model dapat membedakan suatu kelas dengan kelas yang lain [22]. ROC menjelaskan performa model dalam membedakan dua kelas denga *treshold* yang berbeda-beda. Grafik ROC (gambar 2.2) merupakan grafik dua dimensi yang menampilkan hubungan dari FP *rate* sebagai sumbu horizontal dan TP *rate* sebagai sumbu vertikal. Dari *confusion matrix*, FP *rate* dapat dirumuskan dengan $\frac{FP}{FP+TN}$. Sementara rumus TP *rate* sama dengan recall $\frac{TP}{TP+FN}$.

Dari gambar 2.2, semakin lengkungan ROC mendekati titik (0,1) atau melengkung kearah pojok kanan atas maka performa model akan semakin baik, begitupun sebaliknya. Lengkungan ROC akan berpengaruh pada nilai AUC *(Area Under The Curve)*. AUC merupakan nilai area yang berada di bawah lengkungan ROC. Semakin besar nilai AUC maka dapat dikatakan performa model semakin baik. Diasumsikan nilai ROC dilambangkan dengan f, Maka AUC dirumuskan [23]

$$AUC = \int_0^1 f(FPRate)\ d\ FPRate$$

$$= 1 - \int_0^1 f^{-1}(TPRate)\ d\ TPRate$$



Gambar 2.2 Grafik ROC - AUC

Penggunaan ROC – AUC pada *multiclass* model dapat dilakukan dengan menggunakan pendekatan *one vs all*. ROC-AUC dibangun sejumlah N dengan N adalah jumlah kelas pada data. Jika data mempunyai tiga kelas yaitu x, y, dan z, maka akan dibuat tiga grafik ROC – AUC yang menjelaskan kelas x → (y, z), y → (x, z), dan z → (x, y).

2.2.6. Flask

Flask adalah *web framework* mikro yang ditulis menggunakan Python. Flask dibangun dengan dua ketergantungan. *Routing*, *debugging*, dan *web server gateway interface (WSGI)* yang disediakan oleh subsistem Werkzeug, sementara template pendukung disediakan oleh Jinja2. Kedua nya dibuat oleh pengembang yang sama dengan pengembang Flask [24]. Dalam penggunaanya, Flask sama sekali tidak memaksa pengguna untuk memakai salah satu alat pendukung seperti *library* maupun ekstensi dari *framework* ini. Alat pendukung disesuaikan dengan kepentingan penggunanya. Beberapa keunggulan dari flask adalah cukup besarnya komunitas pengguna yang menyediakan berbagai macam library flask.

Selain itu flask mempunyai banyak ekstensi yang dapat digunakan sesuai keperluan pengembangan.Ekstensi Flask adalah paket tambahan untuk mendunkung fungsionalitas dari

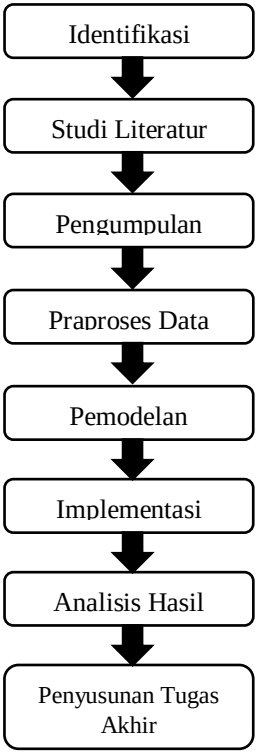
aplikasi [25]. Beberapa ekstensi yang sering digunakan secara umum diantaranya adalah Flask-couchDB, Flask-Mail, Flask-RESTful dan masih banyak lagi.

BAB III
METODOLOGI

Pada bab ini dijelaskan mengenai tahapan metodologi sistematis pengerjaan penelitian ini. Setiap tahapan akan diterangkan lebih rinci pada bab ini. Metodologi ini akan menjadi panduan dalam penyusunan penelitian ini.

3.1. Tahapan Pelaksanaan Tugas Akhir

Pada sub bab ini akan diuraikan alur metodologi yang akan dipakai pada penelitian ini. Alur metodologi dijelaskan pada gambar 3.1.



Gambar 3.1 Alur metodologi

3.2. Uraian Metodologi

Pada sub bab ini dijelaskan secara lebih rinci masing-masing alur metodologi yang dilakukan dala penyelesaian penelitian tugas akhir ini.

3.1.1. Identifikasi masalah

Tahapan ini dilakukan untuk menemukan masalah yang ada pada studi kasus perusahaan PT. XYZ. Objek penelitian ini adalah divisi pemasaran digital. Salah satu kegiatan divisi ini adalah melakukan pengelompokan pengguna. Pengelompokan pengguna bertujuan untuk membantu perusahaan dalam menentukan perlakuan yang tepat kepada setiap kelompoknya. Selain itu pengelompokan bisa digunakan dalam kegiatan kerjasama dengan perusahaan lainya yang memerlukan data dari perusahaan PT. XYZ. Beberapa pengelompokan berdasarkan klasifikasi maupun regresi sudah dilakukan oleh divisi ini seperti pengelompokan *lifestyle* pengguna dan *credit scoring*. Permasalahannya, divisi ini belum mempunyai suatu sistem klasifikasi untuk mengelompokan *average revenue per user* atau ARPU pengguna. Data yang menunjang sistem klasfikasi ini sudah siap namun masih mentah dan belum diolah. Masalah ini yang mendorong penelitian ini untuk dilakukan. Penelitian ini mengangkat masalah yang ada pada divisi pemasaran digital PT. XYZ untuk mengklasifikasikan tingkatan ARPU pengguna berdasarkan data demografi pengguna sosial media.

3.1.2. Studi literatur

Pada tahap ini dilakukan studi literatur terkait permasalahan yang diangkat pada penelitian ini. Tahapan ini bertujuan untuk mencari berbagai referensi yang bersumber dari buku, artikel, jurnal, maupun dokumen lainya yang berkaitan dengan penelitian ini. Studi literatur dilakukan untuk menambah ilmu dan memahami konsep dan dasar teori yang digunakan pada penelitian ini. Terkait dengan penelitian ini, studi literatur yang dilakukan difokuskan mengenai proses penggalian dengan teknik klasifikasi. Literatur selanjutnya akan berfokus pada

metode klasifikasi menggunakan *random forest*. Literatur klasifikasi mengenai ARPU lainya juga dapat dijadikan sebagai referensi penelitian ini.

3.1.3. Praproses data

Pada tahap ini data yang didapat diolah terlebih dahulu sehingga data menjadi siap pakai pada tahap selanjutnya yaitu permodelan. Data yang didapat dipraproses menggunakan python. Proses ini akan menyeleksi fitur-fitur yang ada sehingga sesuai dengan keperluan pemodelan. Setelah itu data akan di filter sesuai batasan masalah penelitian. Sebagai proses pembersihan data, *missing value* dan *unknown value* akan dihilangkan dalam proses ini.

Proses selanjutnya adalah proses *encoding non numeric* data menjadi *numeric* untuk keperluan pemodelan. Tahap ini merupakan salah satu bagian dari tahap praproses data. Tahapan ini bertujuan untuk merubah data mentah yang pada atributnya terdapat nilai data *non-numeric* akan diubah menjadi *numeric* untuk keperluan permodelan. Metode data encoding yang digunakan dalam penelitian ini adalah *label encoding* untuk data *non-numeric* yang mengandung tingkatan dan *one-hot encoding* pada data *non-numeric* yang tidak mempunyai tingkatan. Luaran dari proses ini adalah data siap pakai untuk proses pemodelan klasifikasi selanjutnya.

3.1.4. Pemodelan klasifikasi

Pada tahap ini dilakukan proses pemodelan klasifikasi menggunakan *random forest*. Secara garis besar tahapan ini dibagi menjadi tiga, yaitu fase latih model, fase uji model, dan uji performa. *K-fold cross validation* akan digunakan untuk melakukan validasi model antara data *train* dan data *test*. Data akan dibagi sejumlah K bagian yang setelahnya akan dilakukan fase latih pada K-1 bagian dan fase uji pada satu bagian tersisa. Iterasi dilakukan dengan bagian train dan test yang berganti sehingga model dibangun tervalidasi silang ke semua bagian dataset. Pemodelan *random forest* akan dibangun menggunakan python dengan bantuan *library Sklearn.*

3.1.4.1. Fase latih model

Pada tahapan ini, model klasifikasi *random forest* akan dibangun menggunakan data *train* dari *K-fold cross validation.* Tujuan dari tahap ini adalah memperkenalkan model dengan pola yang ada pada dataset. Parameter dari klasifikasi *random forest* ditentukan dalam tahap ini. Beberapa parameternya yang ditentukan seperti *n_estimators* yang mendefinisikan jumlah *tree* yang digunakan pada model.

3.1.4.2. Fase uji model

Setelah model dibangun pada tahapan sebelumnya, pada tahap ini model akan diuji menggunakan data *test* dari *K-fold cross validation* . Tujuan dari tahap ini adalah mencoba model yang telah dibangun dengan data *test* sehingga mengetahui apakah model berhasil mengklasifikasikan suatu *instance* dengan tepat (sesuai dengan label *instance*) atau tidak.

3.1.4.3. Uji performa

Dari fase uji model, hasil pengujian akan menghasilkan *confusion matrix* guna keperluan uji performa. Dari *confusion matrix* maka dapat dihitung nilai akurasi, presisi, recall, dan f1 score dari model klasifikasi yang telah dibangun beserta visualisasi kurva ROC - AUC.

3.1.5. Implementasi

Pada tahap ini dijelaskan bagaimana pembangunan model dilakukan dan akan diimplementasikan pada aplikasi berbasis web. Aplikasi web digunakan untuk memudahkan perusahaan dalam melakukan proses pengklasifikasian ARPU pengguna sosial media periode-periode selanjutnya. Aplikasi web akan menjadi media interaksi antara pengguna dengan sistem pemodelan klasifikasi. Salah sati fungsi dari aplikasi ini adalah menyediakan layanan untuk membangun model baru berdasarkan data yang diunggah. Selain itu, fungsi lainya adalah menyediakan tampilan untuk memasukan data. Memasukan data digunakan fungsi *import file CSV*. Fungsi lainya adalah melakukan visualisasi data yang telah dimasukan

beserta hasil pemodelan klasifikasi. Aplikasi web dibangun menggunakan *framework* Flask.

3.1.6. Analisis hasil

Pada tahapan ini akan dilakukan analisis dari hasil implementasi pemodelan klasifikasi. Analisis yang dilakukan berdasarkan visualisasi maupun hasil uji performa data yang digunakan. Tujuan dari analisis ini guna memberikan wawasan bagi perusahaan. Analisis hasil ini akan dijadikan rekomendasi untuk perusahaan.

3.1.7. Penyusunan tugas akhir

Pada tahap terakhir ini dilakukan penyusunan laporan penelitian dalam format tugas akhir yang telah ditetapkan oleh Departemen Sistem Informasi ITS. Laporan penelitian dibuat sebagai bentuk dokumentasi atas terlaksananya penelitian ini. Laporan tugas akhir ini mencakup:

b. Bab I Pendahuluan
   Pada bab ini dijelaskan mengenai penelitian yang meliputi latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, serta relevansi pengerjaan tugas akhir ini.
c. Bab II Tinjauan Pustaka
   Pada bab ini dijelaskan mengenai studi literatur penelitian sebelumnya yang berkaitan dengan penelitian ini. Selain itu pada bab ini dijelaskan dasar teori yang digunakan dalam melaksanakan penelitian ini.
d. Bab III Metodologi Penelitian
   Pada bab ini dijelaskan mengenai tahapan metodologi sistematis pengerjaan penelitian ini. Setiap tahapan akan diterangkan lebih rinci pada bab ini. Metodologi ini akan menjadi panduan dalam penyusunan penelitian ini.
e. Bab IV Perancangan
   Pada bab ini dijelaskan perancangan yang terkait dalam proses pembuatan penelitian ini. Perancangan

diperlukan sebagai panduan yang digunakan untuk implementasi sesuai metodologi yang dipakai.

f. Bab V Implementasi
   Pada bab ini dijelaskan mengenai hal-hal yang dilakukan terkait pengerjaan penelitian ini.
g. Bab VI Analisis Hasil dan Pembahasan
   Pada bab ini dijelaskan mengenai hasil pengerjaan yang diperoleh serta analisis dari hasil tersebut.
h. Bab VII Kesimpulan dan Saran
   Pada bab terakhir ini dijelaskan mengenai kesimpulan yang dapat diambil dari penelitian ini dan dijelaskan juga saran untuk penelitian kedepanya.

BAB IV
PERANCANGAN

Pada bab ini dijelaskan perancangan yang terkait dalam proses pembuatan penelitian ini. Perancangan diperlukan sebagai panduan yang digunakan untuk implementasi sesuai metodologi yang dipakai.

4.1. Pengumpulan Data

Dalam proses pengumpulan data, data yang didapat berbentuk *comma-separated values* atau csv. Dataset yang akan digunakan merupakan data demografi pengguna sosial media pada periode Desember 2017 yang berwilayah di Kota Surabaya. Dataset terdiri dari 13 kolom dengan 744,889 baris. Penjelasan secara rinci dataset diuraikan pada tabel 4.1. Domain dari setiap kolom atau atribut yang bersifat kategorikal dijelaskan pada tabel 4.1.

Tabel 4.1 Deskripsi dataset

No	Nama Kolom	Deskripsi	Domain
1	UUID	*Universal unique identifier* adalah Satuan ID unik pada setiap data.	-
2	MSISDN	*Mobile subscriber integrated services digital network number* atau dikenal dengan nomor layanan provider pengguna.	-
3	Perio Start	Periode data dimulai dalam satuan bulan	-

No	Nama Kolom	Deskripsi	Domain
4	Period End	Periode data berakhir dalam satuan bulan	-
5	Accessed_App	Aplikasi sosial media yang diakses	• BBM • Facebook • Instagram • Line • Whatsapp
6	Age	Usia pengguna	-
7	Gender	Jenis kelamin pengguna	• Male • Female
8	Education	Jenjang pendidikan terakhir pengguna	• Primary : Setara SD. • Secondary : Setara SMP/SMA. • University : Setara Sarjana/Diploma.
9	Marital Status	Status pernikahaan pengguna	• Single : Belum Menikah. • Married – no children : Menikah dan tidak punya anak. • Married – with children : Menikah dan punya anak
10	SES	*Socio economic status* atau status sosial berdasarkan ekonomi pengguna (pendapatan)	• A1 : > 11 juta • A2 : 7 – 11 juta • B1 : 4,25 – 7 juta • B2 : 2,8 – 4,25 juta • C1 : 1,48 – 2,8 juta • C2 : 1,0 – 1,48 juta • D : 0,6 – 1,0 juta • E : < 0,6 juta

No	Nama Kolom	Deskripsi	Domain
11	City	Kota asal pengguna	-
12	Status	Status keaktifan layanan elektronik pengguna	• Dormant : > 3 bulan terakhir aktif • Less Active : 1 – 3 bulan terakhir aktif • Active : < 1 bulan terakhir aktif
13	ARPU	*Average revenure per user* atau rataan pendapatan yang dihasilkan pengguna untuk perusahaan	• Very Low • Low • Medium • High • Very High • TOP Usage

4.2. Praproses Data

Praproses data dilakukan untuk mengolah data mentah yang didapatkan agar siap pakai dalam pemodelan klasifikasi. Beberapa langkah dilakukan dalam praproses data ini yaitu membersihkan data dari nilai *null*, *one-hot-encoding* dengan fungsi agregasi, *encoding* data dan pemilihan fitur.

4.2.1. Eksplorasi Data

Proses eklporasi data dilakukan untuk mempelajari dan memahami data. Pemahaman tentang data akan memudahkan dalam menentukan langkah praproses selanjutnya. Eksplorasi yang dilakukan adalah meninjau jumlah nilai *null* pada data, proporsi sebaran kelas data, dan korelasi antar atribut.

4.2.2. Pembersihan data dari nilai null

Proses ini bertujuan untuk membersihkan data dari nilai yang tidak diinginkan. Nilai yang tidak diinginkan terdiri dari nilai kosong atau *Nan* dan nilai tidak diketahui atau *unknown (UNK)*. *Record* atau *tuple* yang mempunyai nilai yang tidak diinginkan pada salah satu atributnya akan dihapus.

4.2.3. One-hot encoding dengan fungsi agregasi

Pada proses ini dilakukan pengelompokan data dengan menggunakan fungsi agregasi *group-by* berdasarkan msisdn dan atribut lainya. Pada setiap *record* akan dicek apakah mempunyai atribut yang sama namun mengakses aplikasi yang berbeda. *Record* yang mempunyai msisdn dan atribut yang sama akan di dikelompokan menjadi satu *record* dan aplikasi yang diakses akan diagregasikan kedalam langkah selanjutnya yaitu *one-hot-encoding*.

One-hot-encoding merupakan proses membuat atribut *non-numeric* yang bersifat kategorikal yang tidak mempunyai hirarki yang jelas antara setiap nilainya menjadi atribut baru. One-hot-encoding akan diimplementasikan pada atribut a*ccessed_app*. Proses one-hot-encoding dengan fungsi agregasi diilustrasikan pada gambar 4.1.

Gambar 4.1 Illustrasi one-hot-encoding

4.2.4. Encoding data

Pada proses ini data non-numerik yang bersifat kategorikal dan mempunyai hirarki yang jelas antara setiap nilainya akan diubah menjadi urutan angka. Proses ini dilakukan secara manual dengan memetakan nilai yang ada kepada nilai tujuan yaitu angka sehingga setelah proses ini dilakukan, atribut dapat dikenali oleh model. Atribut yang akan dilakukan proses

encoding ini adalah gender, education, marital_status, ses, status, dan kelas ARPU. Pemetaan dijelaskan pada tabel 4.2.

Tabel 4.2 Pemetaan label encoding

Atribut	**Value**	**Encoded Value**	**Atribut**	**Value**	**Encoded Value**
Gender	Female	0	Ses	B1	5
	Male	1		A2	6
Education	Primary	0		A1	7
	Secondary	1	Status	Dormant	0
	University	2		Less_active	1
Marital_status	Single	0		Active	2
	Married - no children	1	Arpu	Very Low	0
	Married - with children	2		Low	1
Ses	E	0		Medium	2
	D	1		High	3
	C2	2		Very High	4
	C1	3		TOP Usage	5
	B2	4			

4.2.5. Pemilihan fitur

Tahap terakhir dalam praproses data adalah pemilihan atribut. Pada tahap ini atribut yang ada akan di filter sesuai dengan kebutuhan model. Pemilihan dilakukan dengan mempertimbangkan relevansi dan korelasi antara atribut yang ada dengan kelas tujuan yang diinginkan. Fitur yang akan dipakai dalam pemodelan ini antara lain age, gender, education, marital_status, ses, status, dan accessed_app dengan kelas tujuan yaitu ARPU.

4.3. Pemodelan Klasifikasi

Proses ini secara garis besar dibagi menjadi empat tahap yaitu penentuan parameter terbaik, fase latih, fase uji, dan uji performa. Dalam tahap penentuan parameter, data akan dibagi menjadi data latih dan data uji dengan perbandingan 70% data latih dan 30% data uji. Untuk tahap selanjutnya model akan dilatih dan diuji menggunakan data latih dan data uji yang dihasilkan dar metode K-fold *cross validation*. Model klasifikasi *random forest* dibangun dengan menggunakan *library* dari *sklearn* yaitu *ensamble.RandomForestClassifier*.

4.3.1.Penentuan parameter terbaik

Tahapan ini dilakukan untuk mencari parameter terbaik atau ambang batas terbaik pada parameter dari algoritma random forest. Parameter yang akan diatur nilainya dijelaskan pada tabel 4.3.

Tabel 4.3 Daftar Parameter

No	Parameter	Deskripsi
1	n_estimator	Jumlah pohon yang dibangun
2	bootstrap	Penggunaan metode bootstrap saat memilih sample data untuk membangun setiap pohonnya
3	max_features	Metode penetapan maksimal fitur yang dipakai pada setiap pohonnya
4	max_depth	Kedalaman percabangan pada setiap pohonnya
5	min_samples_split	Minimal jumlah sampel untuk dipecah
6	min_samples_leaf	Minimal jumlah sampel pada setiap daun atau akhir dari cabang

Pencarian nilai dilakukan dengan mencoba semua pola dari nilai parameter yang telah didefinisikan sebelumnya. Setiap parameter akan diberi urutan nilai atau urutan metode yang akan dicoba semua pola yang ada untuk menemukan parameter dengan nilai akurasi terbaik. Percobaan diimplementasikan pada data latih yang dibagi menggunakan metode *cross-validation.* Penentuan parameter terbaik dilakukan

menggunakan *library* dari *sklearn* yaitu *model_selection.GridSearchCV*.

4.3.2. Fase latih model

Model akan dilatih menggunakan metode *k-fold cross validation* dengan nilai k = 5 dan data yang telah diacak berdasarkan barisnya. Fase latih akan menggunakan 4/5 bagian pada setiap iterasi nya. Proses ini dilakukan menggunakan *library* dari *sklearn* yaitu *model_selection.Kfold.*

4.3.3. Fase uji model

Model akan diuji menggunakan metode *k-fold cross validation* dengan nilai k = 5 dan data yang telah diacak berdasarkan barisnya. Fase uji akan menggunakan 1/5 bagian pada setiap iterasi nya. Proses ini dilakukan menggunakan *library* dari *sklearn* yaitu *model_selection.Kfold.*

4.3.4. Uji performa

Performa dari kinerja model akan dilihat dari hasil *multi-class confusion matrix. Confusion matrix* akan divisualisasikan dengan menggunakan grafik *heat map*. Dari laporan *confusion matrix* akan di hasilkan nilai *accuracy, preicsion, recall, f1-score* dan *support* dari setiap kelas. Nilai setiap metrik didapatkan dari tiga metode perhitungan rata-rata per kelasnya. Rataan *micro, macro* dan *weighted*. *Confusion matrix* dan laporan metrik lainya dilakukan dengan menggunakan *library sklearn.metrics*.

4.3.1. Perancangan Aplikasi

Model yang telah dibuat akan diimplementasikan pada aplikasi berbasis web. Aplikasi dibangun menggunakan kerangka kerja Flask. Aplikasi bertujuan untuk mempermudah perusahaan dalam memprediksi pengguna baru berdasarkan atribut demografinya. Secara garis besar, aplikasi dibagi menjadi dua modul yaitu modul *create* dan *load.*

4.4.1.Modul create

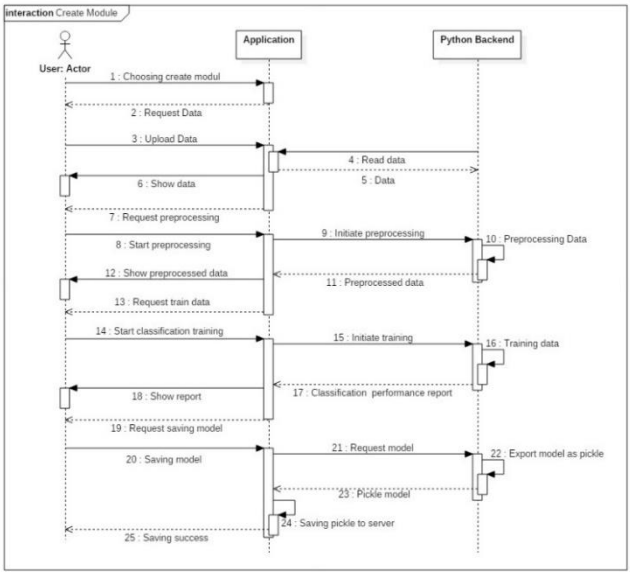
Modul ini berfungsi untuk membuat model klasifikasi baru berdasarkan data baru berbentuk csv. Dengan modul ini perusahaan dapat dengan mudah membangun model klasifikasi baru sesuai dengan data yang diunggah. Model klasifikasi yang dibangun menggunakan metode yang sama dengan model induk penelitian ini yaitu *random forest* dengan menggunakan parameter yang sama.

Secara umum pada modul ini pengguna aplikasi diminta untuk mengunggah data berbentuk csv yang selanjutnya akan dilakukan praproses data. Data yang sudah di praproses akan digunakan model dengan tahapan yang sama dengan tahapan pembuatan model induk. Aplikasi akan menampilkan visualisasi performa model baru. Tahapan terakhir, pengguna dapat menyimpan model tersebut ke dalam server sehingga dapat dipakai dalam prediksi klasifikasi ARPU dalam modul *load*. Tahapan pada modul ini digambarkan pada gambar 4.2 dengan menggunakan *sequence diagram.* Berikut adalah skenario beserta penjelasannya dari setiap tahapan pada *sequence diagram* pada gambar 4.2.

1) Pengguna memilih modul *create* **(Choosing create modul)**. Aplikasi akan meminta pengguna untuk mengupload data **(request data)**.
2) Pengguna mengunggah data csv **(Upload data)**. *Backend* akan membaca data tersebut dan menyimpan secara sementara **(read data)**.
3) Aplikasi akan menampilkan data yang telah diunggah **(Show data)** beserta pilihan untuk melakukan praproses data **(request preprocessing)**.
4) Pengguna memilih untuk melakukanpraproses data **(Start preprocessing)**. Aplikasi akan menginisasi *backend* **(initiate preprocessing)** untuk melakukan prarposes data **(preprocessing data)**.
5) Hasil data yang telah di praproses ditampilkan oleh aplikasi beserta penjelasan singkat langkah-langkah

praproses yang dilakukan **(Show preprocessed data)**. Aplikasi juga menampilkan pilihan untuk melakukan pembuatan model **(request train data)**.

6) Pengguna memilih melakukan pembuatan model klasifikasi **(Start classification training)**. Aplikasi akan menginisiasi *backend* **(initiate training)** untuk melatih model dengan data yang ada **(training data)**.
7) Aplikasi akan menampilkan laporan uji performa klasifikasi beserta visualisasinya **(show report)**. Aplikasi akan menampiklan pilihan untuk menyimpan model ke server **(request saving model)**.
8) Pengguna memilih untuk menyimpan model **(saving model)** sehingga aplikasi akan meminta model pada *backend* **(request model)**. *Backend* akan mengekspor model menjadi .pickle **(esxport model as pickle)** lalu memberikan **Pickle model** kepada aplikasi. Aplikasi akan menyimpan pickle di server **(saving pickle to server)** dan akan menotifikasi pengguna bahwa enyimpanan model berhasil **(saving success).**



Gambar 4.2 Modul create sequence diagram

Untuk *user interface*, fitur utama dari modul create adalah upload data, praproses data, train beserta hasil uji performa. Rancangan tampilan halaman utama dari modul create dapat dilihat pada gambar 4.3.

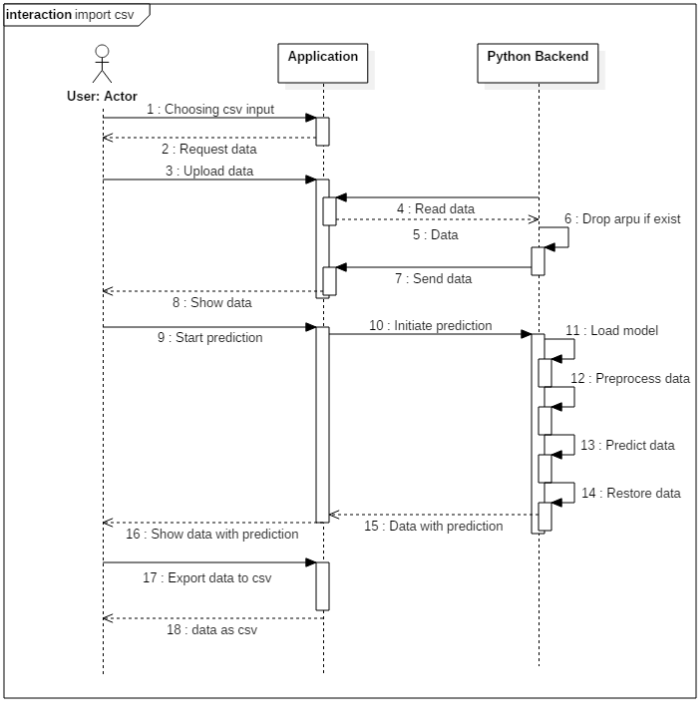
Gambar 4.3 Rancangan tampilan halaman utama modul create

4.4.2. Modul load

Modul ini berfungsi untuk memprediksi ARPU dari data baru. Ada dua pilihan input untuk dapat memprediksi ARPU yaitu input formulir dan upload csv. Pengguna aplikasi dapat memlih model apa yang diinginkan dalam menjalankan prediksi ini. Model *default* atau model induk yang telah dibuat pada tahap selanjutnya ataupun model yang dibuat dari aplikasi modul *create*. Dengan adanya modul ini pengguna aplikasi dapat dengan mudah untuk mendapatkan hasil ARPU berdasarkan atribut demografinya.

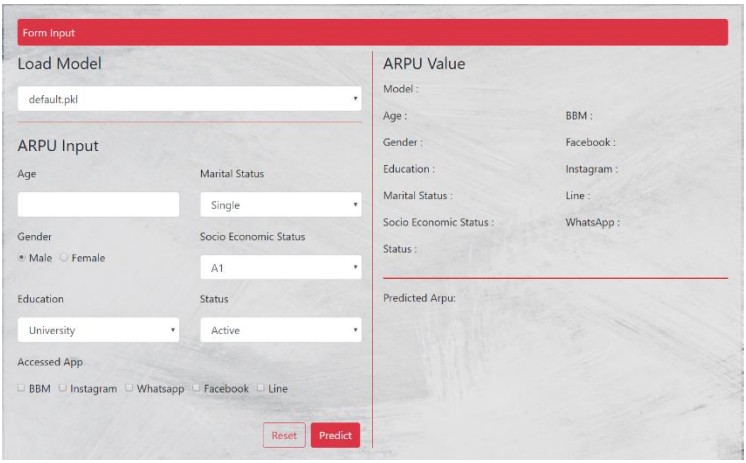
Dalam menggunakan sub-modul input formulir pengguna aplikasi akan disediakan beberapa isian formulir untuk mendefinisikan atribut yang menjadi prediktor untuk model klasifikasi. Setelah mendefinisikan atribut beserta memilih model klasifikasi yang ingin digunakan, hasil prediksi ARPU akan muncul.Tahapan pada input formulir ini digambarkan pada gambar 4.4 dengan menggunakan *sequence diagram*. Berikut adalah skenario beserta penjelasannya dari setiap tahapan pada *sequence diagram* pada gambar 4.4.

1) Pengguna memilih modul load dengan menggunakan formulir **(Choosing form input).** Aplikasi akan menampilan formulir **(show form)**.
2) Pengguna mengisi formulir **(Input form)**. Selanjutnya aplikasi akan mengirim data formulir ke *backend* **(send data)**.
3) *Backend* akan memuat model berdasarkan pilihan pada formulir **(load model)**. Model akan melakukan prediksi berdasarkan data formulir **(predict class)**.
4) *Backend* akan mengirim hasil kepada aplikasi **(send prediction result)** untuk ditampilkan kepada pengguna **(show prediction result)**.



Gambar 4.4 Input csv sequence diagram

User interface pada sub-modul formulir berisikan formulir beserta hasil pada tampilan yang sama. Rancangan tampilan sub-modul formulir dapat dilihat pada gambar 4.5.



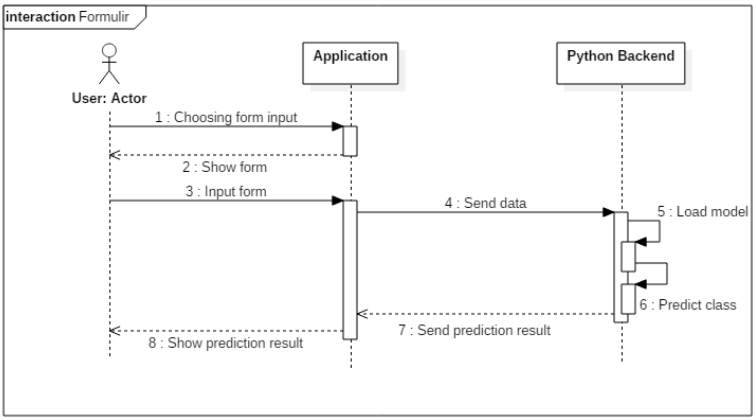
Gambar 4.5 Rancangan tampilan sub-modul input formulir

Selain formulir, pengguna aplikasi dapat memprediksi klasifikasi ARPU dengan mengunggah data berbentuk csv yang ada. Hasil prediksi akan dimasukan kedalam kolom baru pada csv sehingga pengguna aplikasi dapat melihat ARPU untuk semua *record*. Hasil dapat diekspor kedalam csv. Alur klasifikasi ARPU dari data csv dapat dilihat pada gambar 4.6. Berikut adalah skenario beserta penjelasannya dari setiap tahapan pada *sequence diagram* pada gambar 4.6.

1. Pengguna memilih sub-modul input csv **(choosing csv input)**. Aplikasi akan meminta pengguna mengunggah data berbentuk csv.
2. Pengguna mengunggah data csv **(upload data)**. *backend* akan membaca data tersebut. Jika sudah terdapat atribut ARPU maka *backend* akan membuang atribut tersebut **(drop arpu if exist)**. *Backend* akan

mengirimkan data **(send data)** untuk ditampilkan oleh aplikasi **(show data)**.

3. Pengguna memulai untuk memprediksi arpu pengguna dengan memlih model yang sudah ada **(start prediction)**. Aplikasi akan menginisasi *backend* untuk melakukan prediksi **(initiate prediction)**. *Backend* akan memuat model **(load model)**. Data dipraproses terlebih dahulu **(preprocess data)**. *Backend* akan memulai prediksi menggunakan model yang telah dimuat **(predict data)**. *Backend* akan mengembalikan data ke bentuk semula **(restore data)** dan mengirimnya kepada aplkasi **(Data with prediction)** untuk ditampilkan beserta visualisasinya **(show data with prediction)**.
4. Pengguna dapat mengekspor data yang telah diprediksi dalam bentuk csv **(export data to csv).** Data berbentuk akan tersimpan di sistem lokal **(data as csv)**.



Gambar 4.6 Input formulir sequence diagram

User interface pada sub-modul input csv akan menyediakan pilihan unggah berkas csv untuk diprediksi. Rancangan tampilan sub-modul input csv dapat dilihat pada gambar 4.7.

Gambar 4.7 Rancangan tampilan sub-modul input csv

BAB V
IMPLEMENTASI

Pada bab ini dijelaskan bagiamana proses pelaksanaan penelitian ini. Proses meliputi pembuatan model dan aplikasi berdasarkan rancangan yang sudah dijelaskan pada bab sebelumnya.

5.1. Lingkungan Implementasi

Lingkungan merupakan perangkat-perangkat yang dibutuhkan dalam melaksanakan proses implementasi penelitian tugas akhir ini. Adapun lingkungan yang dijelaskan pada sub bab ini antara lain perangkat keras yang digunakan selama proses implementasi berlangsung, perangkat lunak yang digunakan untuk menunjak proses implementasi, dan *library* yang dipakai dalam implementasi. Lingkungan implementasi akan dituangkan dalam tabel. Perangkat keras yang digunakan dijelaskan pada tabel 5.1. Perangkat lunak yang digunakan dijelaskan pada tabel 5.2. *Library* yang digunakan dijelaskan pada tabel 5.3.

Tabel 5.1 Lingkungan perangkat keras

No	Perangkat keras	Spesifikasi
1	Laptop	Acer Notebook Swift SF314-54G
2	Processor	Intel Core i5-8250U
3	Memory	DDR4 8192MB
4	Hardisk	1 TB Toshiba HDD

Tabel 5.2 Lingkungan perangkat lunak

No	Perangkat lunak	Penggunaan
1	Windows 10	Sistem Operasi
2	Python 3.7 64-bit	Bahasa pemograman
3	Anaconda Jupyter Notebook	Python IDE
4	Google Chrome	Web Browser
5	Visual Studio Code	Text Editor

Tabel 5.3 Lingkunan library

No	Library	Penggunaan
1	flask	Web service microframework
2	wtforms	Formulir validator dan render
3	flask_wtf	Integrator antara flask dengan wtforms
4	flask_uploads	Menangani pengunggahan berkas
5	werkzeug	Alat pengamanan nama berkas
6	pandas	Mengelola data berbentuk tabel
7	numpy	Operasi vektor dan matriks
8	pickle	Serialisasi objek Python
9	sklearn	Membangun model
10	scipy	Melakukan operasi matematika
11	json	JSON Flattened
12	plotly	Visualisasi data

Library yang dijelaskan pada tabel 5.3 menjadi *dependencies* aplikasi. Kumpulan *library* tersebut harus diinstall terlebih dahulu agar aplikasi dapat berjalan dengan baik. *Library* dapat diinstal dengan menggunakan pip install. Aplikasi dapat dijalankan dengan menggunakan skrip 5.1 untuk mengkonfigurasi flask mikro-framework sebagai *web service.* Aplikasi dijalankan pada port 5000. Menjalankan flask dan melakukan instalasi pip bisa dilakukan dengan menggunakan terminal dengan skrip 5.2.

```
# Configuration
app = Flask(__name__)

if __name__ == '__main__':
    app.run(debug=True, port=5000)
```

Skrip 5.1 Konfigurasi flask

```
# menginstall library depedencies aplikasi
Pip install library

# menjalankan aplikasi
Python app.py
```

Skrip 5.2 Terminal syntax

5.2. Praproses Data

Pada Tahap ini data akan diproses melalui beberapa langkah transformasi data sehingga luaran dataset dari tahap ini akan siap pakai untuk tahap berikutnya yaitu pemodelan. Praproses akan dilakukan menggunakan bahasa pemograman Python dengan bantuan beberapa library Pandas dan Numpy. Pada penelitian ini data akan dibaca dan disimpan menjadi *dataframe* dengan nama variabel df.

5.1.1. Eksplorasi Data

Eksplorasi data yang pertama adalah melihat tipe data dan nilai *null* pada setiap atributnya dengan menggunakan skrip 5.3. Tipe data dan jumlah data pada setiap atributnya dapat diketahui dengan menggunakan fitur yang ada pada tipe data dataframe yaitu info. Presentase didapatkan dengan membandingkan data *non-null* dengan total keseluruhan data.

```
def percentage(df):
    df.info()
    numerator = df.count()-1
    denomenator = df.shape[0]-1
    if type(numerator) == pd.core.series.Series:
        return
(numerator/denomenator*100).map('{:.1f}%'.format)
    elif type(numerator) == int or type(numerator) ==
float:
        return
'{:.1f}%'.format(float(numerator)/float(denomenator)*100)
    else:
        print("check type")

percentage(df)
```

Skrip 5.3 Melihat tipe data dan nilai null

Pengecekan distribusi kelas juga dilakukan pada tahap ini. Distribusi nilai dari kelas target yaitu ARPU dilakukan untuk melihat keseimbangan nilai kelas target. Melihat distribusi kelas didapatkan dengan menggunakan skrip 5.4. Fungsi dari skrip 5.4 akan menghasilkan tabel dataframe yang memperlihatkan jumlah beserta presentase dari setiap kelas yang ada.

```
def distribution(col):
    index = col.value_counts().index
    value = col.value_counts().values
    precentage = col.value_counts(normalize=True) * 100
    df = pd.DataFrame({'jumlah':value,
'presentase':precentage})
    return df

distribution(df['arpu'])
```

Skrip 5.4 Melihat distribusi kelas

Eksporasi data yang terakhir adalah melihat korelasi dari tiap atributnya. Atribut yang dipakai diantaranya adalah age, gender, education, marital_status, ses, status, dan ARPU. Pembangunan skrip dilakukan dengan menggunakan korelasi metode pearson. Hasil korelasi divisualisasikan dengan diagram heatmap. Skrip yang digunakan dalam melakukan perhitungan korelasi dan pembuatan visualisasi heatmap dilakukan dengan menggunakan skrip 5.5.

```
col=['age', 'gender', 'education', 'marital_status', 'ses',
'status', 'arpu']
def heatmap_corr(col, df):
    corr_df=df[col]
    cor= corr_df.corr(method='pearson')
    fig, ax =plt.subplots(figsize=(8, 6))
    plt.title("Correlation Plot")
   sns.heatmap(cor, mask=np.zeros_like(cor,dtype=np.bool),
   cmap=sns.diverging_palette(220, 10, as_cmap=True),
   square=True, ax=ax, linewidths=.5, annot=True)
    plt.show()
heatmap_corr(col, df)
```

Skrip 5.5 Diagram korelasi antar atribut

5.1.2. Pembersihan dari nilai null

Nilai dianggap kosong jika tidak bernilai atau *null* dan pada dataset ini nilai 'UNK' dianggap sebagai *unknown* sehingga dianggap sebagai tidak bernilai. Pembersihan nilai *null* dilakukan dengan cara membuang baris yang mengandung nilai *null*. Membuang nilai *unknown* atau pada kasus data ini bernilai UNK dilakukan dengan mengambil data dengan atribut gender selain nilai UNK. Pengecekan nilai UNK hanya dilakukan pada atribut gender dikarenakan nilai *null* maupun nilai UNK terfokuskan pada satu *record*, sehingga *record* yang cacat akan mempunyai nilai *null* atau UNK pada seluruh atributnya. Tahap pembersihan nilai null dilakukan dengan menggunakan skrip 5.6.

```
df = df.dropna(how='any',axis=0)
df = df[~df.gender.str.contains("UNK")]
```

Skrip 5.6 Membersihkan data dari nilai kosong

5.1.3. One-hot-encoding dengan fungsi agregasi

Setiap baris pada data merepresentasikan satu msisdn dengan atributnya yang mengakses satu aplikasi. Sehingga satu msisdn yang mengakses dua aplikasi akan direpresentasikan kedalam dua baris dengan atribut yang sama dan seterusnya. Untuk membedakan satu record dengan lainya makan terdapat kolom uuid sebagai identitas.

Proses ini akan membuat baris dengan msisdn dengan atribut yang sama akan menjadi satu baris. Kolom aplikasi yang diakses akan dilakukan proses one-hot-encoding sehingga setiap nilai aplikasi yaitu Whatsapp, Line, Instagram, Facebook, dan BBM akan dijadikan atribut baru yang jika diakses oleh msisdn tersebut maka akan bernilai satu (1) dan jika tidak akan bernilai nol (0). Proses ini dilakukan dengan menggunakan skrip 5.7.

```
df['accessed_app'] =
pd.Categorical(df['accessed_app'],
categories=['WhatsApp', 'Instagram', 'LINE',
'BBM-UCI ', 'Facebook'])

df = pd.get_dummies(df,
columns=['accessed_app']).groupby(['msisdn',
'period_start', 'period_end', 'age', 'gender',
'education', 'marital_status', 'ses','city',
'status', 'arpu'], as_index=False, sort=False).sum()
```

Skrip 5.7 One-hot-encoding dengan fungsi agregasi

5.1.4. Encoding data

Proses ini dilakukan untuk menjadikan data dengan tipe non-numerik menjadi data dengan tipe numerik. Hal ini dilakukan karena model pada *library* Sklearn pada python hanya mengenali data numerik. Proses ini dilakukan pada atribut yaitu *gender*, *education*, *marital_status*, *ses*, *status*, dan *arpu*. *Encoding* dilakukan secara manual dengan memetakan sendiri nilai fungsi dan tujuannya. Pemetaan secara manual memperhatikan tingkatan dari setiap atribut data. tahapan *encoding* dapat dilihat pada skrip 5.8.

```
encoding = {'gender':{ 'FEMALE':0, 'MALE':1},
'education':{'Primary':0,'Secondary':1,'University':2
},
'marital_status':{'Single':0,'Married - no
children':1,'Married - with children':2},
'ses':{'E':0,'D':1,'C2':2,'C1':3,'B2':4,'B1':5,'A2':6
,'A1':7},
'status':{'dormant':0,'less_active':1,'active':2},
'arpu':{'VERY LOW':0, 'LOW':1, 'MEDIUM':2, 'HIGH':3,
'VERY HIGH':4, 'TOP Usage':5}
 }

df.replace(encoding, inplace=True)
```

Skrip 5.8 Encoding Data

5.1.5. Pemilihan fitur

Fitur merupakan atirbut yang akan dijadikan bahan latih model atau disebutjuga prediktor. Tidak semua atribut yang ada dibutuhkan dalam membangun suatu model klasifikasi. Atribut yang akan digunakan sebagai fitur harus mempunyai arti bagi kelas tujuan yang pada kasus ini adalah ARPU.

Dari 13 atribut yang ada, atribut yang akan dijadikan sebagai fitur antara lain *age*, *gender*, *education*, *marital_status*, *ses* dan atribut baru yaitu aplikasi-aplikasi yang diakses yang diperoleh proses sebelumnya. Atribut yang lain akan dibuang dengan menggunakan skrip 5.9. Msisdn tidak perlukan karena merupakan identifier pada dataset. period_start, period_end dan city dibuang karena seluruh data pada dataset mempunyai nilai yang sama.

```
df = df.drop(['msisdn','period_start','period_end',
'city'], axis = 1)
```

Skrip 5.9 Membuang atribut yang tidak dibutuhkan

5.3. Pemodelan Klasifikasi

Proses ini dilakukan untuk membangun model klasifikasi metode *random forest* menggunakank data yang telah dipraproses sebelumnya. Sebelum mencari parameter *random forest* terbaik, data dipecah menggunakan skrip 5.10 menjadi dua bagian yaitu data latih dan data uji.

```
df = df.sample(frac=1).reset_index(drop=True)
length = len(list(df))
X = df.iloc[:,0:-1]
y = df.iloc[:,-1]

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.3)
```

Skrip 5.10 Membagi data menjadi data uji dan data latih

Sebagai langkah awal pada skrip 5.10, data akan diacak terlebih dahulu. Selanjutnya fitur dan kelas tujuan akan dipisah sebelum akhirnya menggunakan *library train_test_split* data akan dipisah dengan proporsi 0.7 bagian sebagai data latih dan 0.3 bagian menjadi data uji.

5.2.1. Penentuan parameter terbaik

Parameter terbaik ditentukan dengan mencoba semua kemungkinan nilai parameter dengan suatu set nilai yang telah didefinisikan. Pola nilai parameter terbaik ditentukan berdasarkan hasil akurasi tertinggi klasifikasi yang dicoba dengan menggunakan cross validation dengan tiga *fold* pada data latih. Penentuan dilakukan dengan menggunakan bantuan *library GridSearchCV* dengan melakukan beberapa iterasi. Setiap iterasi parameter akan disesuaikan agar mendapat nilai terbaik. Parameter lainya yang bukan menjadi variabel penyesuaian akan mengikuti nilai *default* dari *library*.

Penentuan parameter terbaik atau biasa disebut parameter tuning dilakukan dengan menggunakan skrip 5.11. Tingkatan nilai parameter yang dipakai dapat disesuaikan pada setiap iterasinya. Iterasi dilakukan untuk dapat menjangkau nilai parameter yang lebih optimal dari iterasi sebelumnya. Nilai setiap parameter pada iterasi pertama sebagai inisiasi didefinisikan dengan nilai pada tabel 5.4.

Tabel 5.4 Set nilai iterasi pertama

No	Parameter	Set Nilai	Nilai Terbaik
1	n_estimators	[30, 50, 100, 200]	100
2	bootstrap	[True, False]	False
3	max_features	[‘auto’, ‘sqrt’]	Auto
4	max_depth	[4, 6, 8]	8
5	min_samples_split	[2, 5, 10]	10
6	min_samples_leaf	[1, 2, 4]	4

```
param_grid = {
    'n_estimators': [30, 50, 100, 200],
    'bootstrap': [True, False],
    'max_features': ['auto', 'sqrt'],
    'max_depth' : [4,6,8],
    'min_samples_split' : [2, 5, 10],
    'min_samples_leaf' : [1, 2, 4]
}

rfc=RandomForestClassifier(random_state=27, n_jobs=3)

CV_rfc = GridSearchCV(estimator=rfc,
param_grid=param_grid, cv= 3)
CV_rfc.fit(X_train, y_train)
```

Skrip 5.11 Parameter tuning iterasi pertama

5.2.2. Fase latih model

Model akan dibangun menggunakan data yang sudah dipraproses sebelumnya. Data akan dilatih menggunakan metode *K-fold cross validation* dengan K = 5. Data latih akan mengambil 4/5 bagian pada setiap iterasinya. Metode ini dilakukan dengan menggunakan skrip 5.12 dengan bantuan library *Kfold. Classifier* yang digunakan merupakan *random forest* dengan parameter terbaik yang telah ditentukan pada tahap sebelumnya.

5.2.3. Fase uji model

Model yang telah dilatih akan diuji menggunakan 1/5 bagian sisa pada setiap iterasinya. Hasil prediksi pada data uji akan dicocokan dengan data aslinya sehingga menghasilkan skor akurasi yang disimpan kedalam *array list.* Selain skor, kelas asli dan kelas yang diprediksi akan dimasukan kedalam *array list* untuk keperluan uji performa seperti yang diperlihatkan pada skrip 5.12.

```
#Define Method
kf = KFold(n_splits=5, random_state=27,
shuffle=True)

scores = []
y_preds = []
y_trues = []
y_probs = []
for train_indices, test_indices in kf.split(X):
    # Perform Fold Validation
    rf1.fit(X.iloc[train_indices],
y.iloc[train_indices])
    y_pred = rf1.predict(X.iloc[test_indices])
    y_true = y.iloc[test_indices]
    score = rf1.score(X.iloc[test_indices],
y_true)
    y_prob =
rf1.predict_proba(X.iloc[test_indices])

    # Append to list
    scores.append(score)
    y_preds.append(y_pred)
    y_trues.append(y_true)
    y_probs.append(y_prob)

    print('iteration..')
```

Skrip 5.12 K-fold cross validation

5.2.4. Uji Performa

Uji performa dilakukan menggunakan *confusion matrix* dan laporan *precision*, *recall*, *f1-score*, dan *support* yang dihasilkan menggunakan bantuan *library sklearn.metrics*. Skrip 5.13 menjelaskan cara membuat *confusion matrix* berdasarkan perbandingan kelas prediksi dan aktual. Hasil *confusion matrix* divisualisasikan menggunakan bantuan plotly. Pada skrip 5.13 untuk membuat *confusion matrix*, daftar kelas yang dikenali oleh *classifier* dipanggil. Daftar kelas yang tadinya berupa

angka sesuai pemetaan pada proses sebelumnya akan dikonversikan menjadi label semula

```
indexes = rf_classifier.classes_
indexes = define_class(indexes)
indexes = asarray(indexes)
cp = cp(actual, pred, indexes)
makecm = makecm(actual, pred)

def asarray(lst):
    res = np.array(lst)
    return res

def define_class(lst):
    indexes = []
    for i, value in enumerate(lst):
        if value == 0:
            indexes.append('Very Low')
        elif value == 1:
            indexes.append('Low')
        elif value == 2:
            indexes.append('Medium')
        elif value == 3:
            indexes.append('High')
        elif value == 4:
            indexes.append('Very High')
        elif value == 5:
            indexes.append('Top Usage')
    return indexes

def makecm(x,y):
    cm = confusion_matrix(x, y)
    return cm
```

Skrip 5.13 Confusion matrix

Tingkat kepentingan dari fitur pembangun model dapat dihasilkan dengan menggunakan atribut dari *library random forest* yaitu *feature_importances_*. Tingkat kepentingan divisualisasikan menggunakan grafik batang yang mengindikasikan tingkat kepentingan seluruh fitur yang membangun model. Visualisasi tingkat kepentingan dibantu *library* plotly. Tingkat kepentingan fitur dijalankan menggunakan skrip 5.14

Selain menggunakan *confusion matrix*, uji performa juga dilakukan menggunakan grafik ROC-AUC. Grafik ROC-AUC dibuat menggunakan skrip 5.15 dengan bantuan Plotly. Penilaian lainya yaitu *accuracy*, *precision*, *recall*, *f1-score*, dan *support* dilakukan menggunakan skrip 5.16 dengan bantuan *library sklearn.metrics* berdasarkan perbandingan kelas prediksi dan aktual. Skor akurasi juga didapatkan dengan menggunakan metode rataan *micro*, *macro* dan *weighted*. Nilai yang dihasilkan akan dijadikan sebuah tabel laporan menggunakan Pandas dataframe.

```
def fi_plot(feature, y):
    feature = pd.Series(feature,
index=list(y)).sort_values(ascending=True)

    data = [go.Bar(x=feature, y=feature.index,
        orientation='h', marker={'color': feature,
'colorscale': 'Viridis'},
    )]
    layout = go.Layout(
        title='Feature Importance',
        paper_bgcolor='rgba(0,0,0,0)',
        plot_bgcolor='rgba(0,0,0,0)'
    )
    fig = go.Figure(data=data, layout=layout)

    graphJSON = json.dumps(fig,
cls=plotly.utils.PlotlyJSONEncoder)

    return graphJSON
```

Skrip 5.14 Tingkat kepentingan fitur

```
def roc_auc_plot(fpr, tpr, roc_auc, n_classes, indexes):
    # Plot all ROC curves
    lw = 2
    colors = cycle(['red', 'orange', 'yellow', 'green', 'blue',
'purple'])
    data = []
    trace1 = go.Scatter(x=fpr["micro"], y=tpr["micro"],
                        mode='lines',
                        line=dict(color='deeppink', width=lw,
dash='dot'),
                        name='micro-average ROC curve (area =
{0:0.2f})'
                        ''.format(roc_auc["micro"]))
    data.append(trace1)

    trace2 = go.Scatter(x=fpr["macro"], y=tpr["macro"],
                        mode='lines',
                        line=dict(color='navy', width=lw,
dash='dot'),
                        name='macro-average ROC curve (area =
{0:0.2f})'
                        ''.format(roc_auc["macro"]))
    data.append(trace2)

    colors = cycle(['red', 'darkorange', 'green',
                    'cornflowerblue', 'aqua', 'purple'])
    for i, color, index in zip(range(n_classes), colors, indexes):
        trace3 = go.Scatter(x=fpr[i], y=tpr[i],
                            mode='lines',
                            line=dict(color=color, width=lw),
                            name='ROC curve of class {0} (area =
{1:0.2f})'
                            ''.format(index, roc_auc[i]))
        data.append(trace3)

    trace4 = go.Scatter(x=[0, 1], y=[0, 1],
                        mode='lines',
                        line=dict(color='black', width=lw,
dash='dash'),
                        showlegend=False)
    data.append(trace4)

    layout = go.Layout(title='ROC - AUC',
                       xaxis=dict(title='False Positive Rate'),
                       yaxis=dict(title='True Positive Rate'),
                       paper_bgcolor='rgba(0,0,0,0)',
                       plot_bgcolor='rgba(0,0,0,0)')

    fig = go.Figure(data=data, layout=layout)
    graphJSON = json.dumps(fig, cls=plotly.utils.PlotlyJSONEncoder)

    return graphJSON
```

Skrip 5.15 Grafik ROC-AUC

```
def cp(true, pred, indexes, cm):
    clf_all = precision_recall_fscore_support(true, pred)
    clf_micro = precision_recall_fscore_support(true, pred,
average='micro')
    clf_weighted = precision_recall_fscore_support(
        true, pred, average='weighted')

    all_dict = {
        "precision": clf_all[0].round(2), "recall":
clf_all[1].round(2), "f1-score": clf_all[2].round(2), "support":
clf_all[3]
    }
    weighted_dict = {
        "precision": clf_weighted[0].round(2), "recall":
clf_weighted[1].round(2), "f1-score": clf_weighted[2].round(2),
"support": clf_weighted[3]
    }
    micro_dict = {
        "precision": clf_micro[0].round(2), "recall":
clf_micro[1].round(2), "f1-score": clf_micro[2].round(2), "support":
clf_micro[3]
    }

    out_df = pd.DataFrame(all_dict, index=indexes)
    avg_tot = (out_df.apply(lambda x: round(x.mean(), 2) if x.name
!=  "support" else round(x.sum(), 2)).to_frame().T)
    avg_tot.index = ["macro avg/total"]
    weighted_df = pd.DataFrame(weighted_dict, index=['weighted
avg/total'])
    micro_df = pd.DataFrame(micro_dict, index=['micro avg/total'])
    out_df = out_df.append(avg_tot)
    out_df = out_df.append(micro_df)
    out_df = out_df.append(weighted_df)
    out_df = out_df.fillna(method='ffill')

    # add accuracy
    apc = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
    apc = apc.diagonal().round(2)

    # avg
    macro = np.average(apc).round(2)
    micro = (cm.diagonal().sum() / cm.sum()).round(2)
    avg = np.asarray([macro, micro, '-'])

    accuracy = np.concatenate((apc, avg))

    out_df['accuracy'] = accuracy
    out_df = out_df[['accuracy','precision', 'recall', 'f1-score',
'support']]
    return out_df
```

Skrip 5.16 Laporan klasifikasi

BAB VI
HASIL DAN PEMBAHASAN

Pada bab ini dijelaskan hasil dari proses penelitian beserta pembahasannya. Hasil dan bahasan pada bab ini menyangkut hasil dari setiap proses maupun luaran dari penelitian ini.

6.1. Hasil Eksplorasi Data

langkah awal sebelum melakukan prarposes data adalah eksplorasi data. Eksplorasi data menghasilkan tipe data dari semua kolom atau atribut yang ada pada dataset. Jumlah data *null* juga diperlihatkan beserta presentasenya. Hasil eksplorasi data dapat dilihat pada tabel 6.1.

Tabel 6.1 Tipe data atribut dan presentase nilai non-null

No	Atribut	Tipe Data	Nilai non-Null	Presentase non-Null
1	UUID	Object	744889	100.0%
2	MSISDN	Object	719095	96.5%
3	Period_Start	Int32	744889	100.0%
4	Period_End	Int64	744889	100.0%
5	Accessed_App	Object	744889	100.0%
6	Age	Float64	689129	92.5%
7	Gender	Object	689129	92.5%
8	Education	Object	689129	92.5%
9	Marital_Status	Object	689129	92.5%
10	SES	Object	689129	92.5%
11	City	Object	744889	100.0%
12	Status	Object	744889	100.0%
13	ARPU	Object	737576	99.0%

Dari hasil eksplorasi data dapat dilihat bahwa dari total 13 atribut yang ada, sepuluh diantaranya merupakan tipe data *object*. *Object* merupakan salah satu tipe data pada dataframe yang dapat diartikan sebagai *string*. Dua atribut merupakan tipe data *Integer* yaitu period_start dan period_end. Satu atribut yaitu age mempunyai tipe data *float*. Dataset mempunyai nilai

null di beberapa atribut. Dari presentase yang ada pada hasil esplorasi data, nilai *null* yang ada pada setiap atributnya tidak melebihi 10% dari total data yang ada. Dari eksplorasi data juga dapat dilihat bahwa nilai *null* maupun nilai UNK yang terdapat pada dataset terfokuskan pada satu baris yang sama, sehingga baris yang cacat akan mempunyai nilai *null* atau UNK pada hampir seluruh atributnya yaitu age, gender, education, marital_sattus, dan ses.

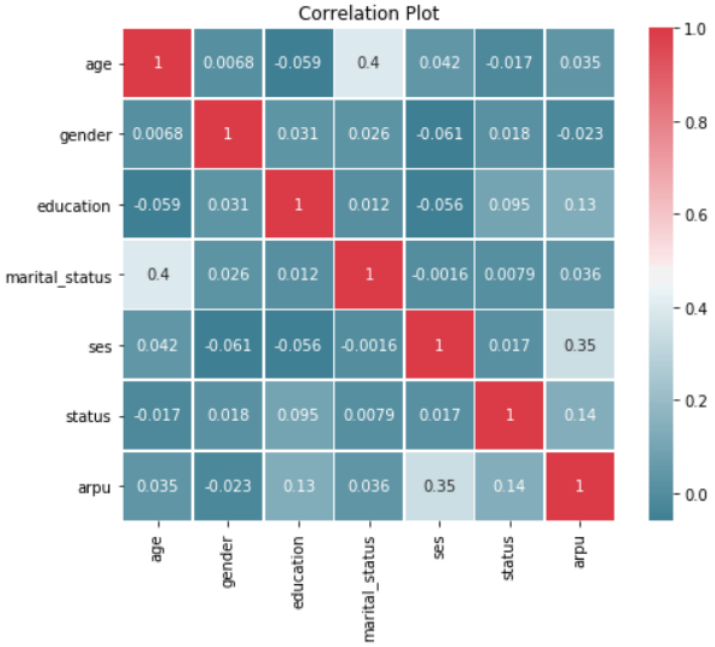
Eksplorasi data lainya dilakukan untuk melihat distribusi dari kelas tujuan klasifikasi yaitu ARPU. Hasil eksplorasi data mengenai distribusi kelas dapat dilihat pada tabel 6.2. Dapat dilihat bahwa dataset memiliki distribusi yang tidak merata. Kelas *high* dan *very high* memiliki jumlah presentase hampir 60% sedangkan untuk kelas low dan medium bahkan masing-masing mempunyai nilai presentase dibawah 10%. Dari hasil distribusi kelas dapat disimbulkan bahwa dataset mempunyai kelas yang tidak seimbang.

Tabel 6.2 Distribusi kelas

No	Nilai	Jumlah	Presentase
1	Very Low	89734	12.17 %
2	Low	22878	3.1 %
3	Medium	64498	8.74 %
4	High	223407	30.29 %
5	Very High	199392	27 %
6	Top Usage	137667	18.66 %

Eksplorasi data lainya dilakukan untuk mengukur nilai korelasi antar atribut data. Atribut-atribut pada data yang akan diuji korelasinya antara lain adalah age, gender, education, marital_status, ses, status, dan ARPU. Hasil uji korelasi antar atribut divisualisasikan menggunakan diagram *heatmap* yang dapat dilihat pada gambar 6.1. Pada fokusan klasifikasi ARPU, maka dari gambar 6.1 dapat dilihat korelasi antara atribut

ARPU dengan atribut lainya. Dari hasil uji korelasi didapat bahwa hanya SES yang mempunyai nilai korelasi yang paling signifikan dibanding atribut lain yaitu sebesar 0.35. dari gambar diagtam korelassi menunjukan bahwa atribut-atribut yang dipakai tidak mempunyai korelasi dengan atribut ARPU.



Gambar 6.1 Diagram heatmap uji korelasi atribut data

6.2. Hasil Praproses Data

Praproses data dilakukan dengan melalui tahapan yang telah dijelaskan pada bab sebelumnya. Tahapan-tahapan praproses data yaitu pembersihan dari nilai *null*, *one-hot encoding* dengan fungsi agregasi, *encoding* data dan terakhir adalah pemilihan fitur. Hasil dari praproses data merupakan dataset yang siap digunakan untuk pemodelan pada tahapan selanjutnya.

Tahap pertama dalam praproses data adalah pembersihan dari nilai *null*. Pembersihan nilai *null* dilakukan dengan metode *drop row*. Metode ini dilakukan karena jumlah *record* yang cacat kurang dari 10% dan dari hasil eksplorasi data didapat bahwa nilai *null* cenderung tidak menyebar melainkan terpusat pada atribut di baris yang sama. Karena kecenderungan terpusat inilah baris dengan nilai *null* tidak bisa dipakai karena hampir seluruh atributnya merupakan nilai *null*. Dari hasil implementasi pembersihan didapat baris yang cacat atau mempunyai nilai *null* berjumlah 56775. Setelah pembersihan ini baris dataset berkurang dari 744889 menjadi 688114.

Selain nilai *null*, nilai tidak diketahui atau *unknown* juga dikategorikan sebagai nilai yang cacat. Pada dataset ini nilai tidak diketahui akan berisikan *string* 'UNK'. Dari hasil ekslorasi data diketahui nilai UNK juga tidak menyebar melainkan terpusat pada baris yang sama. Baris yang mempunyai nilai UNK berjumlah 22133. Setelah pembersihan ini baris dataset berkurang dari 688114 menjadi 666027.

Tahapan selanjutnya pada praproses data adalah *one-hot encoding* dengan fungsi agregasi. Setelah melakukan agregasi, baris pada dataset berjumlah 339491. Tahapan selanjutnya adalah *encoding* secara manual sesuai *mapping* yang telah didefinisikan. Tahapan terakhir adalah pemilihan fitur. Tahap pemilihan fitur akan membuang kolom period_start, period_end, dan city karena mempunyai nilai yang sama.

Hasil dari prarposes data keseluruhan menghasilkan dataset yang siap pakai dalam pemodelan. Dataset hasil proses ini mempunyai baris berjumlah 339491 dengan kolom berjumlah 12. Kolom yang dipakai adalah untuk fitur pemodelan yaitu age, gender, education, marital_status, ses, whatsapp, instagram, line, bbm-uci, dan facebook dengan kelas yaitu ARPU. Hasil praproses data direpresentasikan pada tabel 6.3 yang memperlihatkan sampel data berjumlah lima baris.

Tabel 6.3 Potongan hasil praproses data

	Age	Gender	Education	Marital_status	SES	Status	Whatsapp	Instagram	Line	BBM-UCI	Facebook	ARPU
0	52.0	0	2	2	3	1	1	1	0	0	0	4
1	37.0	0	1	2	6	1	1	1	0	0	0	5
2	40.0	1	2	2	7	1	1	1	0	0	0	3
3	40.0	0	1	2	7	0	1	1	1	0	0	5
4	36.0	1	1	0	0	0	1	1	1	0	0	0
5	37.0	0	2	2	0	0	1	1	1	0	0	2
6	24.0	1	1	0	7	0	1	1	1	1	0	3
7	41.0	1	2	2	0	0	1	0	0	0	0	4
8	36.0	1	1	2	4	0	0	0	0	1	0	1
9	21	0	1	0	6	0	0	0	1	0	0	1
.	…..	…	…	…	…	…	…	…	…	…	…	…

6.3. Hasil Pemodelan Klasifikasi

Pemodelan klasifikasi dengan metode *random forest* dibuat dengan parameter berdasarkan hasil proses penentuan parameter terbaik menggunakan metode pencarian *GridSearchCV*. Performa model dapat dilihat pada hasil proses uji performa.

6.3.1. Hasil penentuan parameter terbaik

Penentuan parameter terbaik dilakukan dengan menggunakan bantuan *library GridSearchCV* dengan melakukan beberapa iterasi. Setiap iterasi parameter akan disesuaikan agar mendapat nilai terbaik. Parameter lainya yang bukan menjadi variabel penyesuaian akan mengikuti nilai *default* dari *library*.

Iterasi pertama menggunakan parameter yang diberikan set nilai sesuai dengan tabel 6.4. Kombinasi parameter yang dihasilkan pada percobaan penentuan parameter terbaik iterasi pertama sejumlah 432 kombinasi. Sepuluh kombinasi nilai parameter terbaik dari nilai yang sudah di tentukan pada tabel 6.4 dapat

dilihat pada tabel 6.5. Dari set nilai yang telah didefinisikan dihasilkan nilai parameter terbaik dengan skor akurasi sebesar 0.36999. Pada iterasi pertama, nilai terbaik pada parameter max_depth, min_sample_splot, dan min_sample_leaf merupakan batas atas dari set nilai yang diberikan, oleh karena itu iterasi selanjutnya dibutuhkan untuk menyesuaikan set nilai pada parameter tersebut.

Tabel 6.4 Set nilai penentuan parameter terbaik iterasi pertama

No	Parameter	Set Nilai
1	n_estimators	[30, 50, 100, 200]
2	bootstrap	[True, False]
3	max_features	[‘auto’, ‘sqrt’]
4	max_depth	[4, 6, 8]
5	min_samples_split	[2, 5, 10]
6	min_samples_leaf	[1, 2, 4]

Tabel 6.5 Hasil sepuluh kombinasi nilai parameter dengan akurasi terbaik pada iterasi pertama

No	bootstrap	Max depth	Max features	Min samples	Min samples split	N estimator	score
1	False	8	auto	4	10	100	0.369996142613
2	False	8	sqrt	4	10	100	0.369996142613
3	True	8	auto	2	2	50	0.369949853970
4	True	8	sqrt	2	2	50	0.369949853970
5	True	8	auto	2	5	50	0.369855072464
6	True	8	sqrt	2	5	50	0.369855072464
7	False	8	sqrt	4	10	30	0.369848459801
8	False	8	auto	4	10	30	0.369848459801
9	False	8	sqrt	1	5	50	0.369833030253
10	False	8	auto	1	5	50	0.369833030253
….	…..	…..	….	….	…..	….	……...

Iterasi kedua dilakukan untuk menyesuaikan set nilai yang diberikan untuk parameter max_depth, min_sample_split, dan min_sample_leaf. Percobaan iterasi kedua ini menggunakan parameter beserta nilai yang telah didefinisikan sesuai pada tabel 6.6. Hasil percobaan iterasi kedua dapat dilihat pada tabel 6.7. Percobaan kedua ini mempunyai 8 buah kombinasi nilai parameter. Dari hasil percobaan didapatkan nilai terbaik dengan skor akurasi naik menjadi 0.37137. Iterasi selanjutnya masih dibutuhkan melihat hasil nilai terbaik pada parameter max_depth merupakan batas atas dari set nilai yang diberikan.

Tabel 6.6 set nilai penentuan parameter terbaik iterasi kedua

No	Parameter	Set Nilai
1	max_depth	[8, 9]
2	min_samples_split	[10, 11]
3	min_samples_leaf	[4, 5]

Tabel 6.7 Hasil kombinasi nilai parameter dengan akurasi terbaik pada iterasi kedua

No	Max depth	Min sample leaf	Min sample split	score
1	9	4	10	0.371370907908
2	9	5	11	0.367988445796
3	9	5	10	0.362741305953
4	9	4	11	0.362297373325
5	8	5	10	0.360038151834
6	8	4	10	0.359967825873
7	8	4	11	0.359928267520
8	8	5	11	0.359809592461

Iterasi ketiga dilakukan untuk menyesuaikan set nilai yang diberikan untuk parameter max_depth. Set nilai parameter yang diberikan dapat dilihat pada tabel 6.8. Percobaan iterasi ketiga mempunyai 5 buah kombinasi nilai parameter. Hasil percobaan pada iterasi ketiga dapat dilihat pada tabel 6.9 dimana dari hasil percobaan didapatkan nilai terbaik dengan skor akurasi naik menjadi 0.37288 dengan parameter max_depth senilai 13.

Tabel 6.8 Hasil percobaan penentuan parameter terbaik iterasi ketiga

No	Parameter	Set Nilai	Nilai Terbaik
1	max_depth	[9, 13, 14, 15, 17]	13
Akurasi			**0.37288**

Tabel 6.9 Hasil kombinasi nilai parameter dengan akurasi terbaik pada iterasi ketiga

No	max depth	score
1	13	0.372884502619
2	15	0.372430746510
3	14	0.371957203137
4	17	0.371875983684
5	9	0.371370907908

Hasil percobaan iterasi ketiga menjadi iterasi terakhir. Dari ketiga iterasi yang dilakukan maka didapatkan nilai parameter terbaik pada parameter n_estimators, bootstrap, max_features, max_depth, min_samples_split, dan min_samples_leafs yang dapat dilihat pada tabel 6.10.

Tabel 6.10 Nilai parameter terbaik

No	Parameter	Nilai Terbaik
1	n_estimators	100
2	bootstrap	False
3	max_features	Auto
4	max_depth	13
5	min_samples_split	10
6	min_samples_leaf	4

Sesuai dengan parameter terbaik, pemodelan dibangun menggunakan jumlah pohon sebanyak 100. Setiap pohon melakukan proses klasifikasi menggunakan parameter-parameter lainya sesuai dengan parameter terbaik yang sudah ditentukan. Visualisasi Struktur pohon pada model ini dapat

dilihat pada gambar 6.2. gambar 6.2 merupakan potongan struktur tree pada pohon pertama model random forest.

Gambar 6.2 Struktur pohon klasifikasi random forest

6.3.2. Hasil uji performa

Uji performa dilakukan pada model *random forest* dengan menggunakan parameter terbaik. Model dibangun menggunakan metode k-fold *cross validation* dengan k=5. Pengacakan baris dilakukan dalam menjalankan *k-fold cross validation* (*shuffle = True)* dengan *random_state* = 27. Metode *k-fold* ini menghasilkan lima *fold* dengan setiap *fold* nya mempunyai 65377 baris. Setiap iterasi *cross-validation* akan menggunakan data latih sejumlah 261508 baris dan data uji sejumlah 65377 baris. Dari model yang telah dibangun, uji performa dilakukan menggunakan beberapa metode yaitu *confusion matrix* dan grafik ROC-AUC.

Nilai akurasi didapatkan dari rata-rata akurasi setiap iterasi *cross-validation.* Dari tabel 6.11 dapat dilihat bahwa iterasi dilakukan sebanyak lima kali dengan nilai akurasi yang tidak jauh berbeda. Akurasi mempunyai rentang nilai antara 0.37235 pada iterasi kedua sampai 0.37702 pada iterasi kelima dengan rata-rata nilai yaitu 0.37439.

Dalam kasus klasifikasi menggunakan data dengan kelas yang tidak merata, salah satu pendekatan untuk melihat akurasi secara jelas yaitu dengan mencari nilai akurasi pada setiap kelasnya. Nilai uji performa lainya yang dipakai adalah *accuracy*, *precision, recall,* dan *f1-score*. Nilai-nilai tersebut dihitung berdasarkan kelas yang ada. Nilai *support*

mengindikasikan jumlah setiap kelas yang ada pada dataset. Hasil dari perhitungan nilai-nilai tersebut akan dirata-ratakan dengan menggunakan tiga metode yaitu *macro, micro,* dan *weighted average*. Untuk nilai support maka baris *macro, micro,* dan *weighted average* mengindikasikan jumlah total kelas yang ada. Hasil pengujian nilai ini dapat dilihat pada tabel 6.12.

Tabel 6.11 Skor akurasi

Iterasi	**Skor**
1	0.37457
2	0.37235
3	0.37484
4	0.37316
5	0.37702
Avg	**0.37439**

Tabel 6.12 Accuracy, Precision, recall, f1-score, dan support

	Accuracy	**Precision**	**Recall**	**F1-score**	**Support**
Very Low	0.38	0.45	0.38	0.41	32509.0
Low	0.01	0.31	0.01	0.02	12665.0
Medium	0.03	0.27	0.02	0.05	30786.0
High	0.50	0.35	0.50	0.41	92131.0
Very High	0.51	0.38	0.52	0.44	92464.0
TOP Usage	0.23	0.39	0.23	0.29	66328.0
Macro avg/total	0.28	0.36	0.28	0.27	326883.0
Micro avg/total	0.37	0.37	0.37	0.37	326883.0
Weighted avg/total	-	0.37	0.37	0.34	326883.0

Dari tabel 6.9. dapat dilihat untuk nilai *precision* semua kelas mempunyai nilai yang kecil yaitu dibawah 0.5 dengan kelas Medium mempunyai nilai *precision* terendah yaitu 0.27. pada

kelas *low* dan *medium* mempunyai nilai recall dan f1-score yang sangat kecil. Nilai *f1-score* pada kelas *medium* hanya bernilai 0.05 dan kelas *low* hanya bernilai 0.02. Hal ini menunjukan bahwa kelas model kesulitan untuk secara akurat memprediksi kelas *low* dan *medium* dengan benar. Secara keseluruhan nilai *precision*, *recall* dan *f1-score* dapat dilihat pada bari rata-rata. Uji performa model menghasilkan nilai rata-rata f1-score dengan metode rataan *macro* yaitu 0.27, metode *micro* menghasilkan 0.37, dan metode *weighted* menghasilkan 0.34. Untuk melihat distribusi antara jumlah kelas aktual dengan prediksi dapat dilihat dengan *confusion matrix* pada tabel 6.10.

Pada tabel *confusion matrix* 6.12 sumbu x merupakan kelas prediksi dan sumbu y merupakan kelas aktual. Hasil *confusion matrix* menunjukan model sulit untuk memprediksi kelas low dan medium dengan benar. Model lebih mudah untuk memprediksi kelas very low, high, very high dan top usage. Kebanyakan kelas low dan medium terprediksi menjadi kelas high. Performa model dalam membedakan kelas yang ada dapat diukur menggunakan grafik ROC-AUC. Grafik ROC-AUC dapat dilihat pada gambar 6.3.

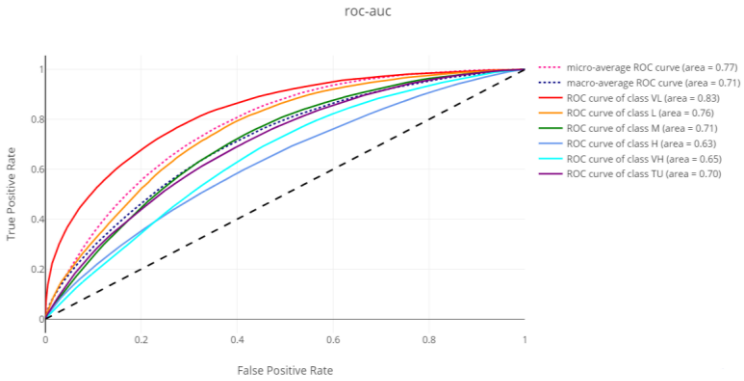
Tabel 6.13 Hasil uji performa confusion matrix

		Prediksi					
		Very Low	**Low**	**Medium**	**High**	**Very High**	**TOP Usage**
Aktual	**Very Low**	12497	47	430	13483	4484	1568
	Low	1863	99	353	6713	2455	1182
	Medium	3590	52	769	17560	6432	2838
	High	5337	40	501	46145	33009	7099
	Very High	2789	46	409	29748	47758	11714
	TOP Usage	1743	34	386	16932	32120	15113

Pada tabel *confusion matrix* 6.12 sumbu x merupakan kelas prediksi dan sumbu y merupakan kelas aktual. Hasil *confusion*

matrix menunjukan model sulit untuk memprediksi kelas low dan medium dengan benar. Model lebih mudah untuk memprediksi kelas very low, high, very high dan top usage. Kebanyakan kelas low dan medium terprediksi menjadi kelas high. Performa model dalam membedakan kelas yang ada dapat diukur menggunakan grafik ROC-AUC. Grafik ROC-AUC dapat dilihat pada gambar 6.3.

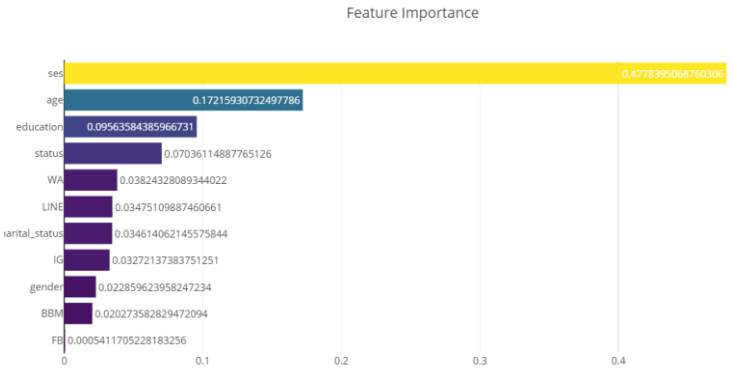
Grafik ROC-AUC pada gambar 6.3 menunjukan performa model dalam memberdakan kelas yang ada. Hasil AUC menunjukan model dapat membedakan kelas dengan akurat jika bernilai 1 dan tidak bisa membedakan dengan kelas lain jika bernilai 0.5. Hasil AUC model yang telah dibangun didapatkan model paling lemah untuk mengklasifikasikan kelas high (0.63) dan paling kuat untuk mengklasifikasikan kelas very low (0.83) dengan akurat. Rata-rata AUC menggunakan rataan mikro mendapatkan niai 0.77 sedangkan rata-rata AUC menggunakan rataan *macro* mendapatkan nilai 0.71.



Gambar 6.3 Grafik ROC-AUC

Hasil dari tingkat kepentingan fitur pembangun model divisualisasikan pada gambar 6.3. Kepentingan sebuah fitur ditentukan berdasarkan seberapa besar fitur tersebut mempengaruhi penurunan nilai *gini impurity* dari setiap pohon.

Pada Random forest maka nilai pengaruh fitur didapat pada rata-rata dari semua pohon yang dibangun [26]. Dari gambar tingkat kepentingan diperoleh bahwa fitur SES atau status ekonomi dan sosial pengguna sangat menonjol sehingga sangat berpengaruh pada kinerja model klasifikasi. Nilai kepentingan SES mencapai 0.47784 disusul oleh fitur Age dengan nilai 0.17216. Pengaruh aplikasi yang diakses oleh pengguna tidak berdampak secara signifikan dalam pembangunan model. Pembangunan model dengan menyaring penggunaan fitur tidak berpengaruh secara signifikan terhadap performa model.



Gambar 6.4 Tingkat kepentingan fitur

6.3.3. Perbandingan dengan metode lain

Pada sub bab ini, hasil dari pemodelan random forest yang telah digunakan untuk memodelkan klasifikasi demografi pengguna sosial media akan dibandingkan dengan metode lainya. Perbandingan dilakukan untuk mengetahui apakah metode lain dapat lebih baik memodelkan klasifikasi ARPU berdasarkan demografi pengguna sosial media atau lebih buruk. Beberapa metode lain yang dicoba untuk perbandingan ini antara lain *K-nearest neighbor* (KNN) dan *Naïve bayes*. Kedua metode tersebut dipakai karena merupakan salah satu metode yang sederhana dan cepat.

Parameter yang digunakan pada setiap metode nya menggunakan parameter *default* yang telah ditentukan pada *library* nya masing-masing. Metode pertama yaitu naïve bayes dilakukan dengan menggunakan 5-*fold cross validation*. Hasil dari pemodelan dengan metode naïve bayes dijelaskan pada tabel 6.14.

Tabel 6.14 Hasil performa model dengan metode Naive bayes

	Accuracy	**Precision**	**Recall**	**F1-score**	**Support**
Very Low	0.16	0.31	0.38	0.41	32339.0
Low	0.23	0.04	0.01	0.02	12609.0
Medium	0.00	0.00	0.00	0.00	30635.0
High	0.50	0.31	0.50	0.38	91614.0
Very High	0.36	0.36	0.36	0.36	91928.0
TOP Usage	0.02	0.35	0.02	0.03	65893.0
Macro avg/total	0.21	0.23	0.21	0.18	325018.0
Micro avg/total	0.27	0.27	0.27	0.27	325018.0
Weighted avg/total	-	0.29	0.27	0.24	325018.0

Metode kedua yang dipakai untuk membandingkan performa model ialah KNN. Sama halnya dengan percobaan menggunakan naïve bayes, Knn dibangun menggunakan 5-*fold cross validation*. Parameter yang digunakan ialah N_neighbors senilai 3. Dari hasil pemodelan ini didapatkan hasil perfora model yang dijelaskan pada tabel 6.15

Dari hasil ketiga pemodelan yang telah dilakukan yaitu *Random fores*t, *Naïve bayes*, dan *K-nearest neighbor*, didapatkan bahwa perbedaan rata-rata akurasi dari ketiga model tersebut dalam mengklasifikasikan ARPU berdasarkan demografi pengguna sosial media tidak signifikan. *Random forest* menjadi metode dengan performa terbaik dengan nilai akurasi 0.37 disusul dengan Knn dengan 0.31 dan terakhir *Naïve bayes* dengan skor

0.27. Ketiga model ini dapat disebutkan tidak dapat memodelkan klasifikasi ARPU berdasarkan demografi pengguna sosial media dengan baik dikarenakan nilai akurasi yang rendah. Perbandingan performa akurasi ketiga model berdasarkan penarikan nilai dengan *macro* dan *micro average* digambarkan pada gambar 6.5.

Tabel 6.15 Hasil performa model dengan menggunakan metode Knn

	Accuracy	**Precision**	**Recall**	**F1-score**	**Support**
Very Low	0.48	0.28	0.48	0.35	32339.0
Low	0.14	0.09	0.14	0.11	12609.0
Medium	0.20	0.16	0.20	0.18	30635.0
High	0.37	0.35	0.37	0.36	91614.0
Very High	0.30	0.39	0.30	0.34	91928.0
TOP Usage	0.23	0.34	0.23	0.27	65893.0
Macro avg/total	0.29	0.27	0.29	0.27	325018.0
Micro avg/total	0.31	0.31	0.31	0.31	325018.0
Weighted avg/total	-	0.32	0.31	0.31	325018.0

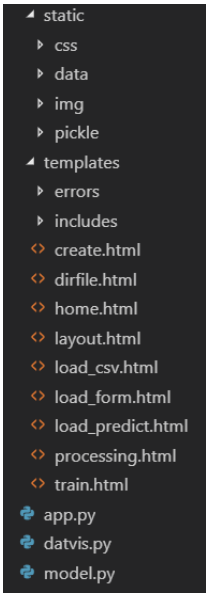
Gambar 6.5 Perbandingan nilai performa akurasi Random forest, Naive bayes, dan Knn

6.4. Aplikasi

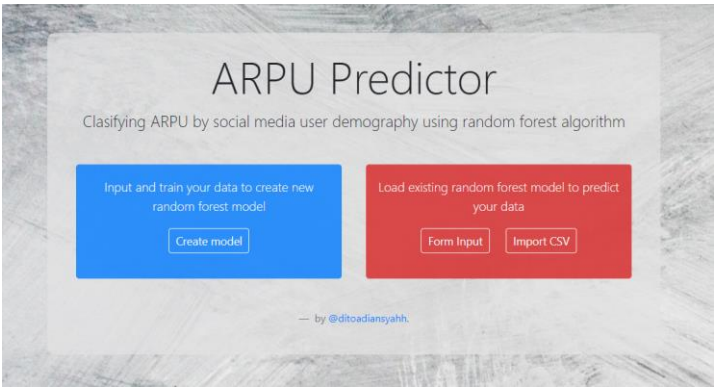
Aplikasi yang dibangun merupakan aplikasi berbasis web menggunakan micro framework yaitu Flask. Aplikasi Flask dibangun menggunakan bahasa pemograman Python. Tampilan dibangun menggunakan HTML dengan dibantu oleh Bootstrap dan CSS. Aplikasi dibagi menjadi dua modul yaitu modul *create* untuk membuat model baru dan modul *load* untuk memprediksi klasifikasi data menggunakan model yang sudah ada. Modul *load* dibagi menjadi dua sub-modul yaitu sub modul input formulir dan sub-modul input csv.

Struktur berkas aplikasi dibagi menjadi tiga bagian yaitu static, templates dan logic. Static berfungsi untuk menyimpan bermacam berkas yang menunjang aplikasi seperti CSS untuk memperindah tampilan, Data untuk menyimpan berkas csv yang diunggah, IMG yang menyimpan gambar-gambar yang ditampilkan di aplikasi, dan PICKLE yang berfungsi menyimpan model yang telah diekspor kebentuk .pickle. Templates berisikan tampilan-tampilan HTML. Tampilan-tampilan pada aplikasi berupa tampilan setiap halaman aplikasi, tampilan error, dan tampilan pendukung. Logic berisikan kerangka yang menjalankan aplikasi berbentuk Python. Logic pada aplikasi berisikan *controller*, *model*, dan visualisasi. Struktur berkas dapat dilihat pada gambar 6.6.

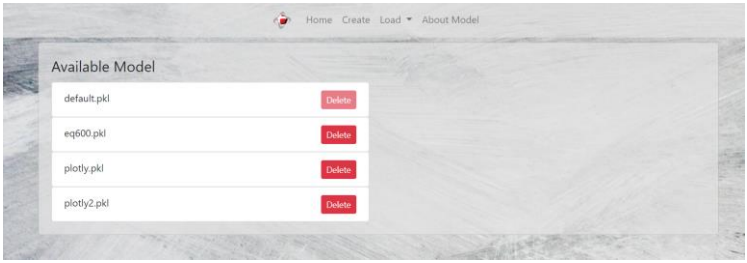
Tampilan utama pada aplikasi akan memberikan pengguna pilihan modul yang tersedia. Modul yang tersedia pada aplikasi berupa modul create dan modul load. Tampilan utama dapat dilihat pada gambar 6.7. Aplikasi akan menampilkan model-model yang tersedia di aplikasi. Model yang tersedia berupa model induk yang telah dibangun sebelumnya menggunakan seluruh dataset (default.pkl) maupun model baru yang dibangun dan disimpan di aplikasi. Model yang dibuat pada aplikasi dapat dihapus kecuali model induk. Tampilan model-model yang tersedia dapat dilihat pada gambar 6.8.



Gambar 6.6 Struktur berkas aplikasi



Gambar 6.7 Tampilan halaman utama

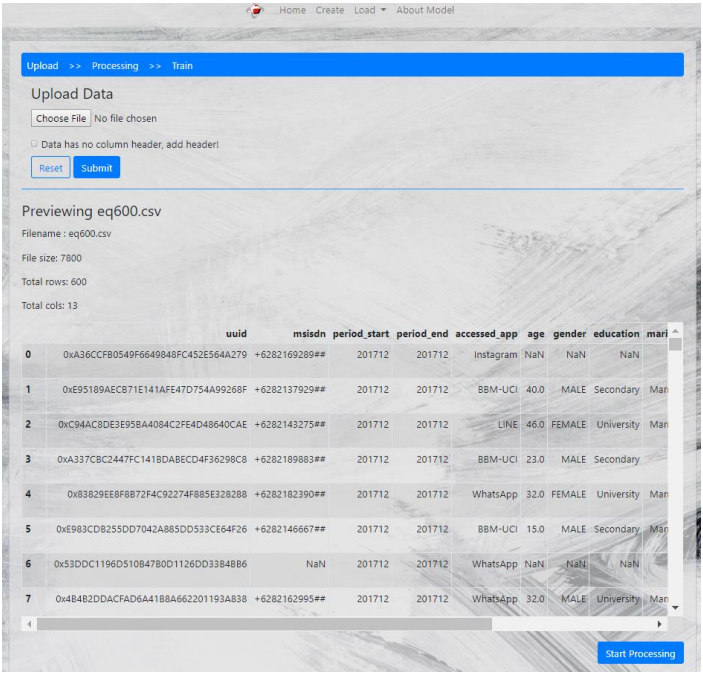


Gambar 6.8 Tampilan model yang tersedia

Proses pengujian aplikasi dilakukan dengan menggunakan data yang merupakan sampel dari data induk. Data sampel yang dipakai pada pengujian aplikasi ini berjumlah 600 baris dengan menyetarakan jumlah kelas yang ada. Hasil yang ditunjukan pada bagian ini merupakan hasil model yang dibangun menggunakan sampel data ini yang bernama eq600.pkl . Hasil prediksi klasifikasi pada modul *load* memakai model eq600.pkl. Pada input csv data yang dipakai merupakan data sampel 600 ini dengan nilai ARPU bawaan dibuang terlebih dahulu oleh aplikasi.

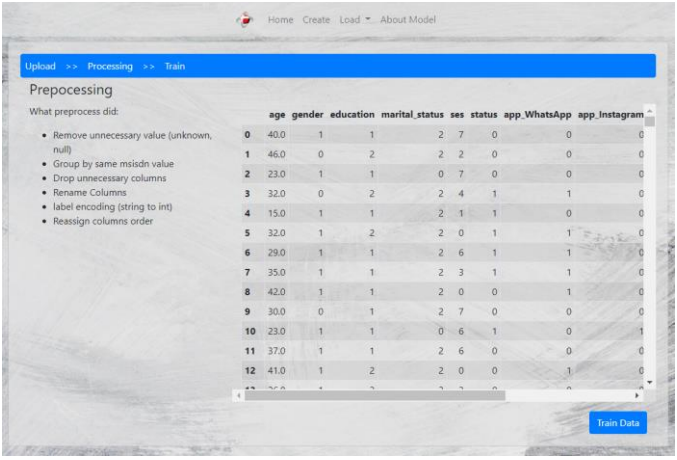
Modul *create* berfungsi untuk membuat modul baru akan meminta pengguna untuk mengungga berkas csv pada aplikasi. Tampilan awal modul *create* akan meminta pengguna untuk menggunggah data berbentuk berkas csv. Setelah mengunggah data, aplikasi akan menampilkan data tersebut dan memebrikan pilihan untuk melakukan proses selanjutnya yaitu praproses data. Tampilan awal mdul *create* dapat dilihat pada gambar 6.9.

Tampilan selanjutnya adalah tampilan hasil praproses beserta langkah langkah yang dilakukan. Tampilan hasil praproses dapat dilihat pada gambar 6.10. Hasil dari modul ini berupa nilai uji performa beserta visualisasinya. Hasil dari modul ini dapat dilihat pada gambar 6.11. Model yang telah dibangun disimpan sebagai pickle.



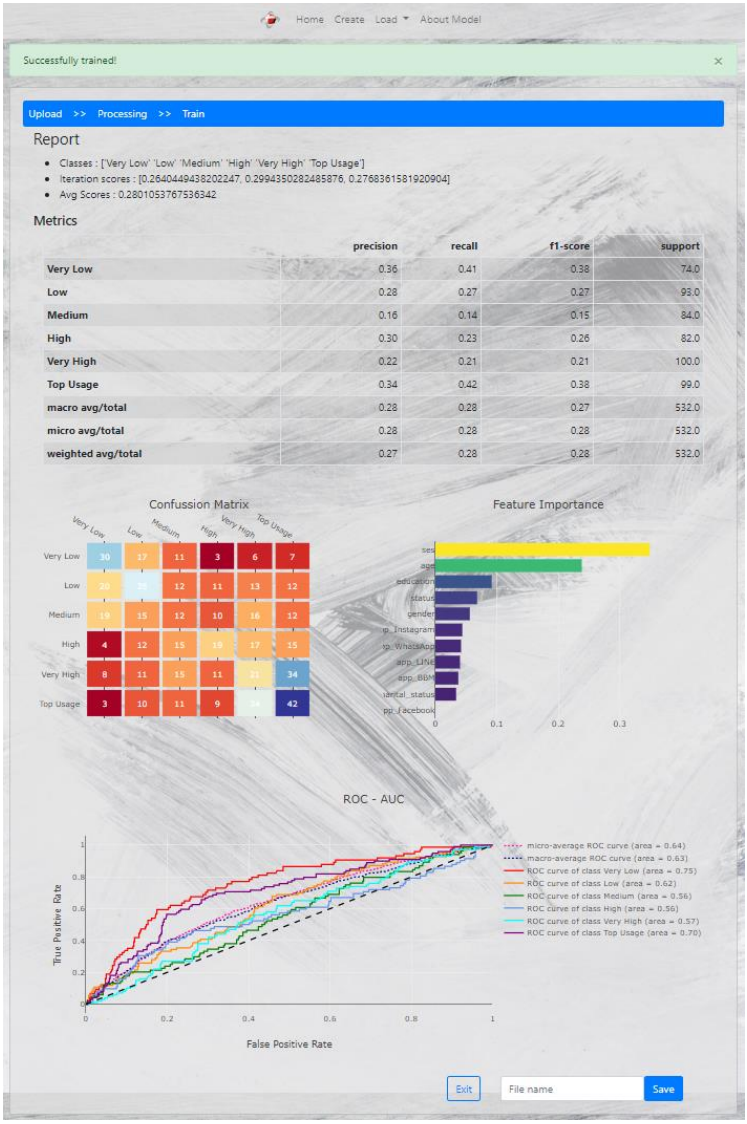
	uuid	msisdn	period_start	period_end	accessed_app	age	gender	education	mari
0	0xA36CCFB0549F6649848FC452E564A279	+6282169289##	201712	201712	Instagram	NaN	NaN	NaN	
1	0xE95189AECB71E141AFE47D754A99268F	+6282137929##	201712	201712	BBM-UCI	40.0	MALE	Secondary	Mar
2	0xC94AC8DE3E95BA4084C2FE4D48640CAE	+6282143275##	201712	201712	LINE	46.0	FEMALE	University	Mar
3	0xA337CBC2447FC141BDABECD4F36298C8	+6282189883##	201712	201712	BBM-UCI	23.0	MALE	Secondary	
4	0x83829EE8F8B72F4C92274F885E328288	+6282182390##	201712	201712	WhatsApp	32.0	FEMALE	University	Mar
5	0xE983CDB255DD7042A885DD533CE64F26	+6282146667##	201712	201712	BBM-UCI	15.0	MALE	Secondary	Mar
6	0x53DDC1196D510847B0D1126DD33B4BB6	NaN	201712	201712	WhatsApp	NaN	NaN	NaN	
7	0x4B4B2DDACFAD6A4188A662201193A838	+6282162995##	201712	201712	WhatsApp	32.0	MALE	University	Mar

Gambar 6.9 Tampilan awal modul create



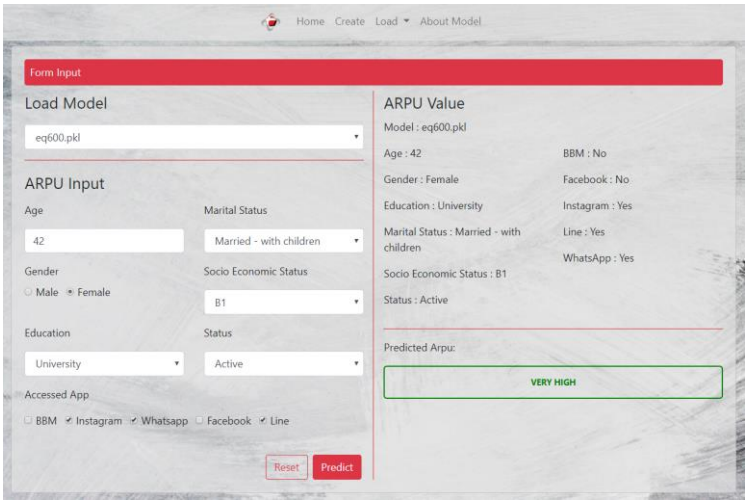
	age	gender	education	marital_status	ses	status	app_WhatsApp	app_Instagram
0	40.0	1	1	2	7	0	0	0
1	46.0	0	2	2	2	0	0	0
2	23.0	1	1	0	7	0	0	0
3	32.0	0	2	2	4	1	1	0
4	15.0	1	1	2	1	1	0	0
5	32.0	1	2	2	0	1	1	0
6	29.0	1	1	2	6	1	1	0
7	35.0	1	1	2	3	1	1	0
8	42.0	1	1	2	0	0	1	0
9	30.0	0	1	2	7	0	0	0
10	23.0	1	1	0	6	1	0	1
11	37.0	1	1	2	6	0	0	0
12	41.0	1	2	2	0	0	1	0

Gambar 6.10 Tampilan hasil praproses data



Gambar 6.11 Tampilan hasil modul create

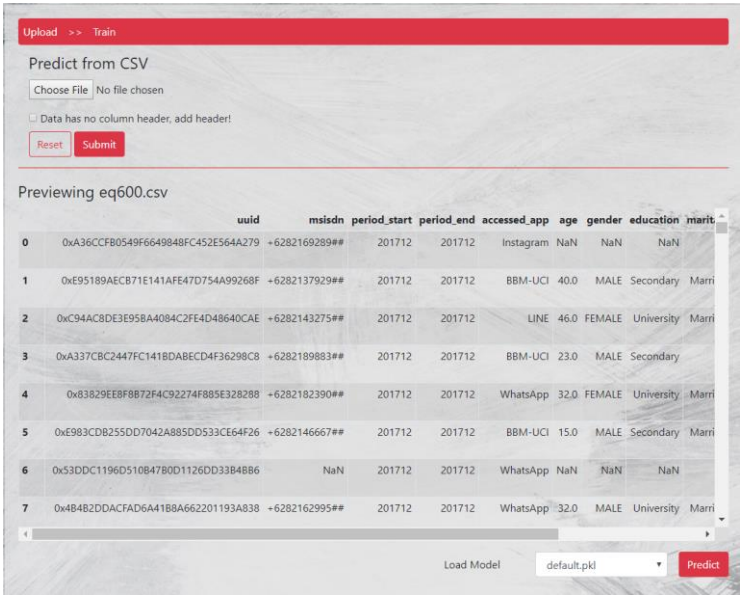
Modul *load* berfungsi untuk memprediksi klasifikasi data baru menggunakan model yang sudah ada. Pada sub-modul input formulir, pengguna akan diberikan formulir berisikan fitur-fitur demografi pengguna beserta aplikasi yang diakses untuk diisi. pengguna diharuskan memilih model yang ingin dipakai sebelum aplikasi akan melakukan prediksi klasifikasi sesuai data yang diisikan. Hasil prediksi klasifikasi akan ditampilkan pada halama yang sama. Tampilan sub-modul input formulir dapat dilihat pada gambar 6.12.



Gambar 6.12 Tampilan hasil prediksi klasifikasi sub-modul input formulir

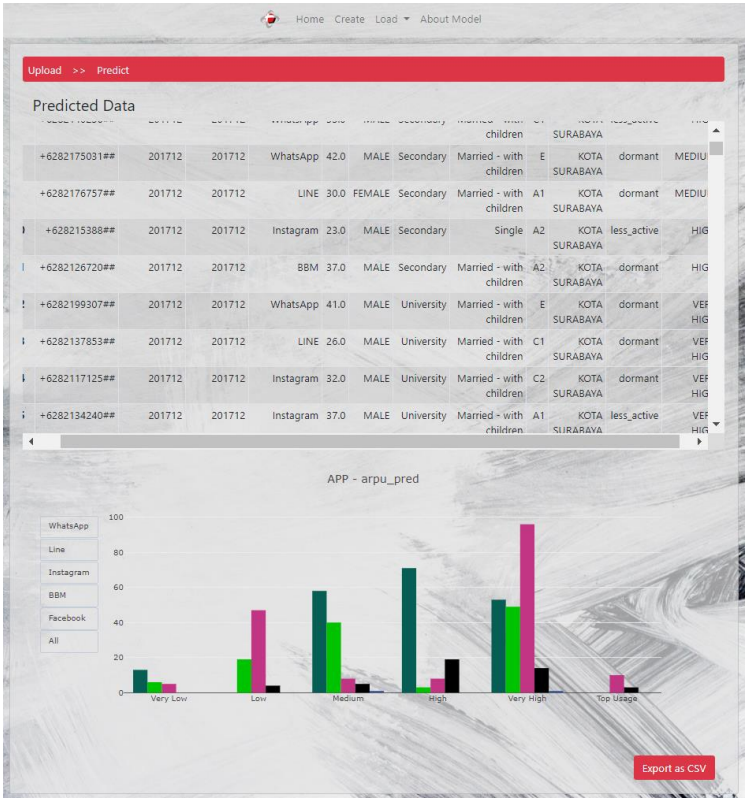
Pada sub-modul input csv, tampilan awal aplikasi akan meminta pengguna untuk mengunggah berkas csv pada aplikasi. Aplikasi akan memberikan pilihan model yang ingin digunakan dalam prediksi klasifikasi. Tampilan awal sub modul input csv ini ditampilkan pada gambar 6.13. Aplikasi akan melakukan prediksi klasifikasi pada berkas csv yang telah diunggah. Hasil prediksi akan ditampilkan berupa tabel beserta visualisasi hasil prediksi ARPU berdasarkan aplikasi yang diakses. Hasil dapat

diekspor kebentuk csv. Tampilan hasil dapat dilihat pada gambar 6.14.



Gambar 6.13 Tampilan awal modul load sub modul input csv

Aplikasi dibangun untuk memudahkan pengguna dalam hal ini PT.XYZ untuk membangun model baru pada data demografi pengguna sosial media yang baru, dan memprediksi klasifikasi ARPU pengguna pada data baru maupun dengan isian formulir. Data yang bisa dipakai pada aplikasi terbatas pada data dengan bentuk csv dan dengan fitur-fitur yang masih statis yaitu age, gender, education, marital status, ses, dan status. Jika data yang dipakai tidak sesuai dengan format yang ada, maka aplikasi meminta untuk mengunggah data kembali.



Gambar 6.14 Tampilan hasil prediksi klasifikasi sub-modul input csv

6.5. Uji Performa Aplikasi

Uji performa aplikasi dilakukan untuk melihat kinerja aplikasi berdasarkan dua aspek yaitu fungsional dan non-fungsional. Pada sub bab ini hasil yang diberikan akan menjawab apakah aplikasi dapat menjalankan dan dijalankan sesuai skenario yang diberikan. Hasil uji performa aplikasi akan dituangkan pada tabel dan akan memberikan hasil "Berhasil" jika berhasil menjalankan atau dijalankan sesuai skenario dan "Gagal" jika tidak berhasil menjalankan atau dijalankan sesuai skenario yang diberikan.

6.5.1. Uji fungsional

Uji fungsional yang dilakukan berupa menguji fungsi-fungsi yang ada pada aplikasi yang dijalankan sesuai skenario yang diberikan. Uji fungsional dilakukan pada kedua modul yaitu modul *create* dan modul *load*.

Uji fungsional aplikasi pada modul *create* dilakukan dengan menguji aplikasi mulai dari skenario menampilkan menu modul sampai dengan menyimpan model yang telah dibuat pada server. Hasil uji fungsional aplikasi modul *create* dapat dilihat pada tabel 6.16.

Tabel 6.16 Hasil uji performa fungsional aplikasi pada modul create

No	Skenario	Hasil
1	Menampilkan menu modul	Berhasil
2	Mengunggah data csv ke aplikasi	Berhasil
3	Mencegah data selain csv ke aplikasi	Berhasil
4	Menampilkan data yang telah diunggah	Berhasil
5	Melakukan prarproses data	Berhasil
6	Menampilkan hasil prarproses data	Berhasil
7	Melakukan pelatihan model klasifikasi pada data hasil praproses	Berhasil
8	Menampilkan nilai uji performa	Berhasil
9	Menampilkan visualisasi hasil *confusion matrix*	Berhasil
10	Menampilkan visualisasi hasil nilai kepentingan fitur	Berhasil
11	Menampilkan visualisasi hasil grafik ROC-AUC	Berhasil
12	Menyimpan model kedalam server	Berhasil

Uji fungsional aplikasi pada modul *load* dilakukan pada kedua sub modul yaitu sub modul input csv dan input formulir. Uji fungsional aplikasi pada sub modul input formulir dilakukan dengan menjalankan skenario pengisian formulir atribut demografi pengguna sosial media sampai menampilkan hasil prediksi klasifikasinya. Hasil uji performa fungsional aplikasi untuk sub modul input formulir dapat dilihat pada tabel 6.17.

Tabel 6.17 Hasil uji performa fungsional aplikasi pada modul load sub-modul input formulir

No	Skenario	Hasil
1	Menampilkan menu sub modul	Berhasil
2	Menampilkan pilihan model yang tersedia	Berhasil
3	Menampilkan formulir atribut demografi pengguna media sosial	Berhasil
4	Memvalidasi isi formulir yang dimasukan	Berhasil
5	Melakukan prediksi klasifikasi ARPU berdasarkan isi formulir	Berhasil
6	Menampilkan isi formulir	Berhasil
7	Menampilkan hasil prediksi klasifikasi ARPU	Berhasil

Uji fungsional aplikasi pada modul *load* berikutnya dilakukan pada sub modul input csv. Uji fungsional dilakukan dengan skenario mulai dari mengunggah data csv sampai dapat mengunduh hasil prediksi klasifikasi secara menyeluruh dalam bentuk csv. Hasil uji performa fungsional aplikasi untuk sub modul input csv dapat dilihad pada tabel 6.18.

Tabel 6.18 Hasil uji performa fungsional aplikasi pada modul load sub-modul input csv

No	Skenario	Hasil
1	Menampilkan menu sub modul	Berhasil
2	Mengunggah data csv ke aplikasi	Berhasil
3	Mencegah data selain csv ke aplikasi	Berhasil
4	Menampikan data yang telah diunggah	Berhasil
5	Menampilkan pilihan model yang tersedia	Berhasil
6	Melakukan praproses data	Berhasil
7	Menjalankan model untuk melakukan prediksi klasifikasi pada data hasil praproses	Berhasil
8	Mengembalikan data ke bentuk semula dan menambah atribut hasil prediksi klasifikasi ARPU	Berhasil
9	Menampilkan data hasil prediksi klasifikasi	Berhasil
10	Menyimpan hasil dalam bentuk csv	Berhasil

Uji fungsional yang dilakukan pada aplikasi menunjukan bahwa aplikasi dapat menjalankan semua skenario yang diberikan. Hal ini dapat dilihat pada tabel 6.16 dalam menjalankan uji fungsional pada modul *create*, tabel 6.17 dalam menjalankan uji fungsional pada modul *load* sub modul input csv, dan tabel 6.18 dalam menjalankan uji fungsional pada modul *load* sub modul input formulir.

6.5.2. Uji non-fungsional

Uji non-fungsional digunakan untuk menguji aplikasi pada hal-hal lain selain fungsi dari aplikasi. Hal-hal yang diuji dalam pengujian ini antara lain skalabilitas, kompatibilitas, dan realibilitas.

Uji non-fungsional skalabilitas dilakukan untuk menguji kemampuan aplikasi dalam menjalankan skenario dengan beban yang diberikan, dalam hal ini adalah ukuran data csv. Pengujian dilakukan dengan skenario aplikasi dijalankan dengan beberapa data dengan ukuran yang bervariasi. Pengujian dilakukan dalam menjalankan modul *create* dan modul *load* sub modul input csv. Hasil dari pengujian diterangkan pada tabel 6.19.

Hasil uji non fungsional skalabilitas dilakukan terhadap semua modul dan sub modul yang ada pada aplikasi. Hasil uji dapat dilihat pada tabel 6.14. Fokusan pada uji ini adalah pada skalah data yag digunggah ke aplikasi. Skala data yang dipakai dalam uji ini adalah data dengan 600 baris, 1000 baris, 10000 baris dan 100000 baris. Dari hasil uji ini dapat dilihat bahwa aplikasi dapat menjalankan seluruh percobaan yang diberikan walaupun skala data yang sangat besar membuat aplikasi memproses dengan waktu yang lebih lama.

Uji non-fungsional selanjutnya adalah reabilitas. Pengujian ini dilakukan untuk menguji tingkat kepercayaan konsistensi aplikasi. Hasil uji non-fungsional reabilitas diterangkan pada tabel 6.19.

Tabel 6.19 Hasil uji non-fungsional skalabilitas

No	Skenario	Hasil
Modul create		
1	Menjalankan modul *create* dengan data berukuran 600 baris.	Berhasil
2	Menjalankan modul *create* dengan data berukuran 1000 baris.	Berhasil
3	Menjalankan modul *create* dengan data berukuran 10000 baris.	Berhasil
4	Menjalankan modul *create* dengan data berukuran 100000 baris.	Berhasil
Modul load sub modul input csv		
5	Menjalankan modul *load* sub modul input csv dengan data berukuran 600 baris.	Berhasil
6	Menjalankan modul *load* sub modul input csv dengan data berukuran 1000 baris.	Berhasil
7	Menjalankan modul *load* sub modul input csv dengan data berukuran 10000 baris.	Berhasil
8	Menjalankan modul *load* sub modul input csv dengan data berukuran 100000 baris.	Berhasil

Hasil uji non fungsional kompabilitas dilakukan pada aplikasi dengan menjalankan aplikasi pada pada lingkungan yang berbeda. Dari hasil yang diperlihatkan tabel 6.20, aplikasi dapat dijalankan pada semua browser pecobaan yaitu Chrome, Mozilla, dan Internet Exploler. Dalam pecobaan sistem operasi aplikasi tidak ada masalah yang diperlihatkan baik windows maupun linux. Pada lingkup *hardware* atau perangkat keras, aplikasi dicoba dalam beberapa perangkat dengan *processor* yang berbeda. Processor yang digunakan dalam percobaan ini adalah inte i3, i5, i7, dan Amd 1600. Dari hasil uji ini tidak ada kendala yang dialami aplikasi. Kendala yang dialami pada aplikasi dalam melakukan uji ini terjadi saat uji kompabilitas versi Python. Aplikasi dibangun menggunakan Python 3.7 dan dapat berjalan baik pada percobaan menggunakan veri Python

3.6. Percobaan pada versi Python 2.7 mengalami masalah. Aplikasi tidak dapat dijalankan dengan baik pada versi Python yang lebih lama dengan muncul beberapa notifikasi masalah terjadi pada *framework* yang dipakai.

Tabel 6.20 Hasil uji non-fungsional kompatibilitas

No	Skenario	Hasil
Browser		
1	Aplikasi dijalankan pada *browser* Chrome	Berhasil
2	Aplikasi dijalankan pada *browser* Mozilla	Berhasil
3	Aplikasi dijalankan pada *browser* Internet Exploler	Berhasil
Sistem operasi		
4	Aplikasi dijalankan pada perangkat windows	Berhasil
5	Aplikasi dijalankan pada perangkat linux	Berhasil
Hardware		
6	Aplikasi dijalankan pada intel i5	Berhasil
7	Aplikasi dijalankan pada intel i3	Berhasil
8	Aplikasi dijalankan pada intel i7	Berhasil
9	Aplikasi dijalankan pada Amd 1600	Berhasil
Versi Python		
9	Aplikasi dijalankan dengan Python 3.6	Berhasil
10	Aplikasi dijalankan dengan Python 2.7	Gagal

Hasil uji non fungsional reabilitas yang dilakukan terhadap aplikasi dapat dilihat pada tabel 6.21. Aplikasi dapat berjalan di lokal tanpa terhubung jaringan internet namun tata letak tampilan menjadi rusak karena tidak terhubung dengan CSS maupun Bootstrap. Selain itu masalah yang terjadi tanpa ada hubungan dengan internet muncul pada aplikasi. Aplikasi tidak dapat menampilkan visualisasi data. Pada percobaan menjalankan aplikasi dan diputus hubungan ke server saat sedan berjalan menghasilkan masalah pada aplikasi. Saat

aplikasi berjalan aplikasi harus terhubung dengan server secara terus menerus.

Tabel 6.21 Hasil uji non-fungsional reabilitas

No	Skenario	Hasil
1	Aplikasi dijalankan tanpa terhubung jaringan internet	Berhasil
2	Aplikasi menampilkan visualisasi tanpa terhubung jaringan internet	Gagal
2	Aplikasi tetap berjalan saat hubungan ke server diputus	Gagal

6.6. Kesimpulan Percobaan

Percobaan model klasifikasi ARPU menggunakan metode random forest terhadap data demografi pengguna sosial media. Dataset demografi pengguna sosial media memiliki 744889 jumlah baris dan 14 jumlah kolom. Praproses data dilakukan untuk menghasilkan data siap pakai pada pemodelan. Praposes data menghasilkan dataset teragregasi yang berjumlah 339491 baris. Fitur-fitur yang dipakai pada pemodelan antara lain age, gender, education, marital_status, ses, whatsapp, instagram, line, bbm-uci, dan facebook dengan kelas yaitu ARPU.

Pemodelan dibangun menggunakan metode *k-fold cross validation* dengan k=5. Kinerja model dapat dilihat dari hasil uji performa yang dilakukan. Dari hasil percobaan yang dilakukan, didapatkan nilai akurasi yang sangat rendah yaitu 0.37439 atau 37.349%. Lemahnya model dalam melakukan klasifikasi juga dapat dilihat dari rendahnya nilai *precision, recall* dan *f1-score* terutama nilai pada kelas *low* dan *medium*. Rata-rata f1-score yang didapatkan dari uji performa model ini bernilai 0.27 meggunakan rataan mikro, 0.37 menggunakan rataan makro, dan 0.34 menggunakan rataan bobot atau *weighted* average.

Uji performa model menggunakan grafik ROC-AUC hanya memberikan nilai Rata-rata AUC menggunakan rataan mikro senilai 0.77 sedangkan rata-rata AUC menggunakan rataan

makro senilai 0.71. Kelas *low* menjadi satu-satunya kelas dengan nilai AUC melebihi 0.80. Kontradiksi terjadi pada nilai presisi, recall dan f1-score terhadap nilai ROC-AUC. Kelas *low* dan *medium* yang mempunyai nilai *recall* yang sangat kecil, tetapi pada nilai ROC-AUC bernilai lebih tinggi daripada kelas-kelas lainya kecuali kelas *low*. Hal ini disebabkan grafik ROC-AUC tidak sensitif terhadap data dengan kelas tidak seimbang. Data yang tidak seimbang terkadang dapat menghasilkan grafik ROC-AUC yang tidak akurat[27].

Model yang dibuat sebelumnya akan menjadi model induk pada implementasi sistem terhadap aplikasi berbasis web. Aplikasi dibangun dalam dua modul yaitu modul *create* modul *load*. Pada modul *create*, aplikasi dapat membuat model klasifikasi baru menggunakan parameter yang telah ditentukan daam proses penentuan parameter terbaik. Model yang dibuat berdasarkan data yang diunggah ke aplikasi. Model yang dibuat dapat disimpan pada aplikasi untuk melakukan prediksi klasifikasi pada modul *loadi*. Pada modul *load* aplikasi dapat melakukan prediksi klasifikasi dengan menggunakan model induk maupun model yang telah dibuat pada aplikasi. Prediksi aplikasi dapat dilakukan pada data yang diunggah ke aplikasi berbentuk berkas csv maupun data yang diisikan pada formulir.

Kesimpulan yang dapat ditarik dari proses percobaan pembuatan model klasifikasi ARPU menggunakan metode random forest terhadap data demografi pengguna sosial media ini bahwa model kurang dapat melakukan klasifikasi dengan akurat. Hasil uji performa yang dilakukan memberikan nilai yang rendah baik dari nilai akurasi, *precision*, *recall*, *f1-score* maupun nilai AUC.

BAB VII
KESIMPULAN DAN SARAN

Pada bab ini dijelaskan kesimpulan yang didapat dari pengerjaan penelitian tugas akhir ini dan saran yang dapat diberikan untuk penelitian selanjutnya.

7.1. Kesimpulan

Berdasarkan hasil penelitian tugas akhir ini, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1) Random forest tidak dapat melakukan klasifikasi ARPU berdasarkan tingkat akurasi yang sangat rendah. Hal ini kemungkinan besar disebabkan karena tingkat korelasi fitur dengan kelas yang rendah.
2) Hasil pengukuran kinerja model dengan menggunakan uji performa menghasilkan penilaian tingkat akurasi sebesar 0.37439 atau 37.349%. Nilai rataan mikro pada *precision* sebesar 0.37, recall sebesar 0.37, f1-score sebesar 0.37. Nilai rataan mikro AUC sebesar 0.77. %. Nilai rataan makro pada *precision* sebesar 0.36, recall sebesar 0.28, f1-score sebesar 0.27. Nilai rataan makro AUC sebesar 0.71.
3) Model terbaik klasifikasi pengguna berbasis ARPU berdasarkan demografi pengguna sosial media dibangun dengan menggunakan metode *random forest* dengan parameter n_estimator 100, bootstrap False, max_features Auto, max_depth 13, max_sample_split 10, max_sample_leaf 4. Model dibangun menggunakan metode *5-fold cross validation.*
4) Aplikasi berbasis web dibangun menggunakan *micro-framework* Flask. Aplikasi dibangun menggunakan bahasa pemograman Python dengan tampilan HTML yang dibantu oleh CSS dan Bootstrap.

7.2. Saran

Berdasarkan hasil penelitian tugas akhir ini, maka saran yang dapat diberikan untuk membantu penelitian selanjutnya adalah sebagai berikut:

1) Penelitian selanjutnya dapat menggunakan data demografi pengguna sosial media dengan rentang waktu yang lebih panjang yaitu lebih dari rentang waktu sebulan.
2) Penelitian selanjutnya dapat menggunakan data demografi pengguna sosial media dengan skala wilayah yang lebih luas, skala wilayah provinsi maupun skala wilayah nasional.
3) Penelitian selanjutnya dapat menggunakan data pengaksesan sosial media yang lebih banyak.
4) Penelitian selanjutnya dapat menggunakan metode pengklasifikasian lainya selain *random forest* pada Python.

DAFTAR PUSTAKA

[1] Philip Kotler and Kevin Lane Keller, *Marketing Management (13th edition)*, 13th ed. 2008.

[2] PDSI KOMINFO, "Siaran Pers No. 112/HM/KOMINFO/05/2018 Tentang Jumlah Pelanggan Telekomunikasi Seluler Prabayar Hasil Rekonsiliasi dan Berakhirnya Program Registrasi Ulang," *Website Resmi Kementerian Komunikasi dan Informatika RI*. [Online]. Available: https://kominfo.go.id:443/content/detail/13125/siaran-pers-no-112hmkominfo052018-tentang-jumlah-pelanggan-telekomunikasi-seluler-prabayar-hasil-rekonsiliasi-dan-berakhirnya-program-registrasi-ulang/0/siaran_pers. [Accessed: 28-Feb-2019].

[3] "Digital in 2018: World's internet users," *We Are Social*, 30-Jan-2018. [Online]. Available: https://wearesocial.com/blog/2018/01/global-digital-report-2018. [Accessed: 28-Feb-2019].

[4] Michael J. A. Berry and Gordon Linoff, *Data mining techniques: for marketing, sales, and customer relationship management*, 2nd ed. Indianapolis, Ind: Wiley Pub, 2004.

[5] Leo Breiman, "Random Forest," pp. 5–32, 2001.

[6] William Sullivan, *Machine Learning For Beginners Algorithms, Decision Tree & Random Forest Introduction*. .

[7] R. Genuer, J.-M. Poggi, C. Tuleau-Malot, and N. Villa-Vialaneix, "Random Forests for Big Data," *Big Data Res.*, vol. 9, pp. 28–46, Sep. 2017.

[8] Huang Chao, Ma Yue-hua, Zhao Hai-bin, and Lu Xiao-ping, "Spectral Classification of Asteroids by Random Forest," *Chin. Astron. Astrophys.*, vol. 41, no. 4, pp. 549–557, Oct. 2017.

[9] Yaya Xie, Xiu Li, E.W.T. Ngai, and Weiyun Ying, "Customer churn prediction using improved balanced

random forests," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5445–5449, Apr. 2009.

[10] Zhenhua Wang, Lai Tu, Zhe Guo, Laurence T. Yang, and Benxiong Huang, "Analysis of user behaviors by mining large network data sets," *Future Gener. Comput. Syst.*, vol. 37, pp. 429–437, Jul. 2014.

[11] Muhammad Fikry Hazmi, "Rancang Bangun Aplikasi untuk Klasifikasi Komentar Netizen pada Media Sosial Pemerinta Daerah di Indonesia Menggunakan Algoritma Random Forest," 2018.

[12] A. Famili, Wei Min Shen, Richard Weber, and Evangelos Simoudis, "Data preprocessing and intelligent data analysis," *Intell. Data Anal.*, vol. 1, no. 1, pp. 3–23, 1997.

[13] Suad A. Alsadi and Wesam S. Bhaya, "Review of Data Preprocessing Technique in Data Mining," 2017.

[14] S B Kotsiantis, D Kanellopoulos, and P E Pintelas, "Data Preprocessing for Supervised Leaning," vol. 1, no. 1, p. 7, 2006.

[15] G. Kesavaraj and S. Sukumaran, "A study on classification techniques in data mining," in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, Tiruchengode, 2013, pp. 1–7.

[16] Alfonso Urso, Antonino Fiannaca, Massimo La Rosa, Valentina Ravì, and Riccardo Rizzo, "Data Mining: Classification and Prediction," in *Encyclopedia of Bioinformatics and Computational Biology*, Elsevier, 2019, pp. 384–402.

[17] Yangming Zhou and Guoping Qiu, "Random forest for label ranking," *Expert Syst. Appl.*, vol. 112, pp. 99–109, Dec. 2018.

[18] Leo Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[19] A. Liaw and M. Wiener, "Classification and Regression by randomForest," vol. 2, p. 6, 2002.

[20] Peter Flach, *Data, Machine Learning: The Art and Science of Algorithms that Make Sense of Data*, 1st Edition. Cambridge University Press, 2012.

[21] Xinyang Deng, Qi Liu, Yong Deng, and Sankaran Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Inf. Sci.*, vol. 340–341, pp. 250–261, May 2016.

[22] Tom Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[23] Xiaoli Zhang, Xiongfei Li, Yuncong Feng, and Zhaojun Liu, "The use of ROC and AUC in the validation of objective image fusion evaluation metrics," *Signal Process.*, vol. 115, pp. 38–48, Oct. 2015.

[24] Miguel Grinberg, *Flask web development*, First edition. Sebastopol, CA: O'Reilly, 2014.

[25] "Extensions Registry | Flask (A Python Microframework)." [Online]. Available: http://flask.pocoo.org/extensions/. [Accessed: 28-Feb-2019].

[26] Carolin Strobl, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis, "Conditional variable importance for random forests," *BMC Bioinformatics*, vol. 9, no. 1, Dec. 2008.

[27] Jesse Davis and Mark Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proceedings of the 23rd international conference on Machine learning - ICML '06*, Pittsburgh, Pennsylvania, 2006, pp. 233–240.

Halaman ini sengaja dikosongkan

BIODATA PENULIS

Penulis lahir di Jakarta, 27 Juni 1997. Penulis bernama lengkap Rachmaddinta Herpradipto Adiansyah merupakan anak pertama dari empat bersaudara. Penulis telah menempuh pendidikan formal di SDIT Auliya Bintaro, SMPN 1 Kota Tangeran, SMAN Cahaya Madani Banten Boarding School, dan akhirnya menjadi mahasiswa Sistem Informasi Institut Teknologi Sepuluh Nopember Surabaya angkatan 2015 dengan NRP 05215 4000 0133. Selama menempuh masa perkuliahan penulis aktif dalam beberapa keanggotaan organisasi maupun kepanitiaan. Keanggotaan tersebut diantaranya UKM Musik ITS, bagian hubungan luar BEM Fakultas, bagian logistik *Information System Expo 2015*, dan koordinator bagian akomodasi transportasi *Information system expo* 2016. Selama masa perkuliahan penulis pernah menjalankan kegiatan magang selama tiga bulan di divisi *IT Support* Telkomsel. Penulis mengambil bidang minat Rekayasa Data dan Intelegensia Bisnis dengan fokusan pada penggalian data. Penulis dapat dihubungi melalui email di dito13adiansyah@gmail.com atau nomor whatsapp +6282125637631.

Halaman ini sengaja dikosongkan

LAMPIRAN A : DATA MENTAH DEMOGRAFI PENGGUNA SOSIAL MEDIA

Potongan data mentah demografi pengguna sosial media bulan desember 2017.

UUID	MSISDN	Period Start	Period End	Accessed app	Age	Gender	Education	Marital Status	SES	City	Status	ARPU
0x1CA0F9C5F832294F959F950E107CEBFF	+62821374017##	201712	201712	Instagram	15.0	FEMALE	Secondary	Married - with children	A1	SURABAYA	dormant	HIGH
0xFFADFD5B0848674B9ABE23F79AF68306	+62821893545##	201712	201712	LINE	15.0	FEMALE	Secondary	Married - with children	A1	SURABAYA	less_active	HIGH
0x39B64E58FE309A47BBD2B1AC21B7F50E	+62821754443##	201712	201712	BBM-UCI	15.0	FEMALE	Secondary	Married - with children	C2	SURABAYA	dormant	TOP Usage
0xFCEA117850177341B1233DCD70C28991	+62821510374##	201712	201712	WhatsApp	15.0	FEMALE	Secondary	Married - with children	C2	SURABAYA	less_active	TOP Usage

0xF0B210558313614F8F82FECC8AF567D1	+62821510374##	201712	201712	Instagram	15.0	FEMALE	Secondary	Married - with children	C2	SURABAYA	less_active	TOP Usage
0xA9DEF4B27D8BE542B29052724AE304E0	+62821243764##	201712	201712	Instagram	15.0	FEMALE	Secondary	Married - with children	E	SURABAYA	dormant	HIGH
0x18FF482AFE15AE4E990A1EE285D6D699	+62821243764##	201712	201712	WhatsApp	15.0	FEMALE	Secondary	Married - with children	E	SURABAYA	dormant	HIGH
0x9E0EE424A0ACED49A31779D81007889A	+62821243764##	201712	201712	LINE	15.0	FEMALE	Secondary	Married - with children	E	SURABAYA	dormant	HIGH
0x7790C534A871174694E94F3661AD2ADB	+62821345882##	201712	201712	LINE	15.0	FEMALE	Secondary	Single	A1	SURABAYA	dormant	HIGH
0x4A3E6CD963001443B55A06F2542C399C	+62821345882##	201712	201712	WhatsApp	15.0	FEMALE	Secondary	Single	A1	SURABAYA	dormant	HIGH
0xDB56F0B9F60A764990B2E21AEBDF03FE	+62821345882##	201712	201712	Instagram	15.0	FEMALE	Secondary	Single	A1	SURABAYA	dormant	HIGH

0x0B7826B8D7333D4385C4D498386E7CA8	+62821206856##	201712	201712	Instagram	15.0	FEMALE	Secondary	Single	A1	SURABAYA	dormant	VERY HIGH
0x303854CBBB74094BB8780B49254BC3F7	+62821206856##	201712	201712	LINE	15.0	FEMALE	Secondary	Single	A1	SURABAYA	dormant	VERY HIGH
0x40141292D77DDA47B6F1B3D91D655931	+62821206856##	201712	201712	WhatsApp	15.0	FEMALE	Secondary	Single	A1	SURABAYA	dormant	VERY HIGH
0xFE7BE151882AE64087D288B5778AF3BA	+62821206856##	201712	201712	Instagram	15.0	FEMALE	Secondary	Single	A1	SURABAYA	dormant	VERY HIGH
0xB0F2CF8B7A62F84EB935310E6E3BDF55	+62821339545##	201712	201712	LINE	15.0	FEMALE	Secondary	Single	A1	SURABAYA	less_active	VERY HIGH
0x3537852B387E074FBB9882C32BE0A820	+62821339545##	201712	201712	Instagram	15.0	FEMALE	Secondary	Single	A1	SURABAYA	less_active	VERY HIGH

0xB7EE4C8DBA01B041A185D4FCEB5389E1	+62821200456##	201712	201712	Instagram	15.0	FEMALE	Secondary	Single	A1	SURABAYA	less_active	VERY LOW
0xD9874165BB4DDA459ED841BB2C1C1356	+62821200456##	201712	201712	LINE	15.0	FEMALE	Secondary	Single	A1	SURABAYA	less_active	VERY LOW
0x2B01ABC805349C4B8C169C6958EFC354	+62821116826##	201712	201712	LINE	15.0	FEMALE	Secondary	Single	A2	SURABAYA	dormant	VERY HIGH
0xE29B813B44236149AD3F4E8CAC68152B	+62821116826##	201712	201712	Instagram	15.0	FEMALE	Secondary	Single	A2	SURABAYA	dormant	VERY HIGH
0xA1604041D4DB9B40AAAAC859F8BE160E	+62821569079##	201712	201712	WhatsApp	15.0	FEMALE	Secondary	Single	B2	SURABAYA	dormant	HIGH
……	…….	…	….	….	….	….	….	….	..	….	….	…..

LAMPIRAN B : DATA DEMOGRAFI PENGGUNA SOSIAL MEDIA

Potongan data demografi pengguna sosial media bulan desember 2017.

msisdn	Period start	Period end	age	gender	education	Marital status	ses	city	status	WA	IG	Line	BBM	FB	ARPU
+6282137401 7##	201712	201712	15.0	FEMALE	Secondary	Married - with children	A1	SURABAYA	dormant	0	1	0	0	0	HIGH
+6282189354 5##	201712	201712	15.0	FEMALE	Secondary	Married - with children	A1	SURABAYA	less_active	0	0	1	0	0	HIGH
+6282175444 3##	201712	201712	15.0	FEMALE	Secondary	Married - with children	C2	SURABAYA	dormant	0	0	0	1	0	TOP Usage

+62821510374##	201712	201712	15.0	FEMALE	Secondary	Married - with children	C2	SURABAYA	less_active	1	1	0	0	0	TOP Usage
+62821243764##	201712	201712	15.0	FEMALE	Secondary	Married - with children	E	SURABAYA	dormant	1	1	1	0	0	HIGH
+62821345882##	201712	201712	15.0	FEMALE	Secondary	Single	A1	SURABAYA	dormant	1	1	1	0	0	HIGH
+62821206856##	201712	201712	15.0	FEMALE	Secondary	Single	A1	SURABAYA	dormant	1	2	1	0	0	VERY HIGH
+62821339545##	201712	201712	15.0	FEMALE	Secondary	Single	A1	SURABAYA	less_active	0	1	1	0	0	VERY HIGH
+62821200456##	201712	201712	15.0	FEMALE	Secondary	Single	A1	SURABAYA	less_active	0	1	1	0	0	VERY LOW
+62821116826##	201712	201712	15.0	FEMALE	Secondary	Single	A2	SURABAYA	dormant	0	1	1	0	0	VERY HIGH
+62821569079##	201712	201712	15.0	FEMALE	Secondary	Single	B2	SURABAYA	dormant	1	0	0	0	0	HIGH
+62821406684##	201712	201712	15.0	FEMALE	Secondary	Single	B2	SURABAYA	dormant	1	1	1	0	0	MEDIUM

+628212233 37##	2017 12	2017 12	15. 0	FEMA LE	Second ary	Singl e	B 2	SURAB AYA	less_act ive	0	1	1	0	0	HIGH
+628212871 61##	2017 12	2017 12	15. 0	FEMA LE	Second ary	Singl e	C 1	SURAB AYA	dorman t	0	1	1	0	0	HIGH
+628218316 35##	2017 12	2017 12	15. 0	FEMA LE	Second ary	Singl e	C 1	SURAB AYA	less_act ive	1	0	0	0	0	HIGH
+628216636 18##	2017 12	2017 12	15. 0	FEMA LE	Second ary	Singl e	C 2	SURAB AYA	dorman t	1	1	0	0	0	VERY HIGH
+628211971 1##	2017 12	2017 12	15. 0	FEMA LE	Second ary	Singl e	C 2	SURAB AYA	less_act ive	1	1	1	0	0	VERY HIGH
+628212068 56##	2017 12	2017 12	15. 0	FEMA LE	Second ary	Singl e	C 2	SURAB AYA	less_act ive	0	0	1	0	0	VERY HIGH
+628212562 40##	2017 12	2017 12	15. 0	FEMA LE	Second ary	Singl e	E	SURAB AYA	dorman t	0	1	1	0	0	HIGH
+628215145 60##	2017 12	2017 12	15. 0	FEMA LE	Second ary	Singl e	E	SURAB AYA	dorman t	1	1	0	0	0	LOW
……	…	….	….	…	…	…	..	….	….	…	..	…	…	…	….